

**A SECURE TITLE DEED REGISTRATION MODEL FOR LAND
TRANSACTIONS IN KENYA BASED ON BLOCKCHAIN**

EVANS KIMUTAI CHESANG

**A Thesis Submitted to the Institute of Postgraduate Studies in Partial Fulfillment of
the Requirements for the Award of Doctor of Philosophy in Information
Technology Security and Audit**

KABARAK UNIVERSITY

DECEMBER, 2025

DEDICATION

1. I do declare that:

- i. This thesis is my work and to the best of my knowledge, it has not been presented for the award of a degree in any university or college.
- ii. This work has not incorporated material from other works or a paraphrase of such material without due and appropriate acknowledgement.
- iii. This work has been subjected to processes of anti-plagiarism and has met the Kabarak University 15% similarity index threshold.

2. I do understand that issues of academic integrity are paramount, and therefore I may be suspended or expelled from the University, or my degree may be recalled for academic dishonesty, or any other related academic malpractices.

Signed: _____

Date: _____

Evans Kimutai Chesang

GDS/M/0175/01/17

RECOMMENDATION

To the Institute of Postgraduate Studies;

The thesis entitled “**A Secure Title Deed Registration Model for Land Transactions in Kenya Based on Blockchain.**” written by **Evans Kimutai Chesang** is presented to the Institute of Postgraduate Studies of Kabarak University. We have reviewed the thesis and recommend it to be accepted in partial fulfillment of the requirements for the award of Doctor of Philosophy in Information Technology Security and Audit.

Signed:_____

Date:_____

Dr. Nelson Masese

Department of Computer Science and Information Technology

Kabarak University

Signed:_____

Date:_____

Prof. Peter Rugiri

Department of Mathematics and Actuarial Science

Kabarak University

COPYRIGHT

©2025

Evans Kimutai Chesang

All rights reserved. No part of this thesis may be reproduced or stored in any retrieval system or transmitted in any form or any means be it electronically or physically including photocopying or recording without the prior written permission of the author or Kabarak University on that behalf.

ACKNOWLEDGEMENTS

This is to acknowledge the Almighty God of all creation for His provision throughout my studies at Kabarak University. Foremost, I am grateful to the Government of the Republic of Kenya for making it possible to have a peaceful country for all her citizens.

I acknowledge Kabarak University, the Chancellor, Management, Staff and students for upholding the mission and the vision of Kabarak University at all times. Their contribution and continued support has created a wonderful conducive learning environment for my academic progression. Secondly, I would wish to register my honest appreciation to my Kabarak University supervisors, Dr. Nelson Masese and Prof. Peter Rugiri as well as the Institute of Postgraduate Studies at Kabarak University, following their sustained guidance, input and advice throughout the entire research journey. Their mentorship and support has made the study a success. My sincere gratitude goes to the Ministry of Education of the Republic of Kenya, through their agency- Higher Education Loans Board (HELB) for granting me a scholarship for my studies.

Finally, thank you my family, colleagues and friends for the sincere prayers, best wishes and moral support throughout my studies.

Thank you very much and God bless you.

DEDICATION

I dedicate this thesis to all scholars seeking to use research innovations in Information Technology to provide sustainable solutions that address inequalities in the society.

ABSTRACT

Land is a basic need that every individual aspires to own. To have control over Land is associated with economic power which is the basis for both social as well as political influence in Kenya. However, there exist persistent challenges, usually attributed to the processes involved in Land registration and Land transactions for Landownership. The non-existence of a universal National Land Information Management System (NLIMS) in Kenya has become the genesis of occasioned perennial conflicts relating to the proof of Land ownership burden on any land dealership. Fraudulent Land registration and transaction activities (Land grabbing) are reportedly being executed almost on a daily basis in Kenya and more often, these kind of transactions result in an endless protracted emotive Land cases in Courts. It is in the view of the aforementioned challenges that the study embarked on a mission of designing a secure title deed registration model for Land transactions in Kenya based on Blockchain, a secure technology with the potential of simplifying the execution of authentic Land registration and transaction processes across the Lands registry in Kenya. To achieve its research objective, the study examined the integration of Land transactions into Blockchain with the aim of creating a secure and transparent environment that guarantees the much needed confidentiality and integrity of Land transactions. Design Science research methodology was adopted by the research to provide an overall scope in the model design and development requirements while the Proof of Concept (PoC) research methodology was used to design and develop the essential modules for the model. The research employed subjective sampling to select users to perform a criteria-based model evaluation approach against the evaluation metrics (persistence, time-stamping, transparency and decentralization) set out by the study. Likert scale tool was adopted by the study to capture the user feedback during the model evaluation process. Results were presented in form of a descriptive summary of the users' feedback. From the results, the study established that the model could effectively execute immutable, time-stamped, transparent and decentralized land registration and land transaction processes that can be used as the basis upon which an authentic proof of land ownership burden can be founded. The study recommends that future research may focus on the prospect of integrating Blockchain with Land taxation systems as well as the adoption of crypto currency as a legal tender to facilitate the execution of smart contracts for properties in Kenya.

Keywords: *NLMIS/ Ardhisasa, Title Deed Registration, Secure Model, Land Transactions, Blockchain*

TABLE OF CONTENTS

DEDICATION	ii
RECOMMENDATION.....	iii
COPYRIGHT.....	iv
ACKNOWLEDGEMENTS	v
DEDICATION	vi
ABSTRACT.....	vii
TABLE OF CONTENTS	viii
LIST OF TABLES.....	xii
LIST OF FIGURES.....	xii
ABBREVIATIONS AND ACRONYMS.....	xv
OPERATIONAL DEFINITION OF TERMS.....	xvi
CHAPTER ONE	1
1.1 Introduction.....	1
1.2 Background Information of the Study	1
1.2.1 Land in Kenya	1
1.2.2 Challenges Facing Land Administration and Land Transactions in Kenya.....	2
1.3 Blockchain Technology	3
1.4 Statement of the Problem.....	4
1.5 Objectives of the Study	5
1.5.1 Overall Objective	5
1.5.2 Specific Objectives.....	5
1.6 Research Questions	6
1.7 Significance of the Study	6
1.8 Scope of the Study	7
1.9 Justification for the study.....	7
1.9 Limitations of the Study.....	8
CHAPTER TWO	9
LITERATURE REVIEW	9
2.1 Introduction.....	9
2.2 The Registration and Titling of Land in Kenya	9
2.3 The Registry of Title Deeds	10
2.3.1 The Torrens System	11

2.4 The Classification of Land in Kenya	12
2.5 The Land Laws in Kenya	13
2.6 The Challenge of Land Administration in Kenya.....	14
2.8 The Land Transaction Process in Kenya.....	15
2.9 The Title Deed Transfer Process in Kenya	16
2.9.1 The Categories of Land Title Deeds in Kenya.....	18
2.10 The Status of the Lands Registry in Kenya Today	19
2.11 The National Land Information Management System.....	21
2.11.1 The Ardhisasa Platform.....	22
2.11.2 Ardhisasa Implementation Issues.....	26
2.12 The Concept of Blockchain Technology	27
2.12.1 The Blockchain Architecture	30
2.12.2 The Transaction execution Process in Blockchain.....	32
2.12.3 The Technical Capabilities of Blockchain Technology	35
2.12.4 Blockchain andthe Land Registry	36
2.12.5 The Integrity of Blockchain Transactions.....	38
2.13 The Features of a Blockchain-Based Model	38
2.14 The Blockchain Transaction authenticity using the Hash Function Algorithm... 42	
2.15 Potential Industry Application Areas of Blockchain	42
2.16 Secure Execution of Land Transactions in Blockchain	48
2.16.1 Recording a Land Transaction in Blockchain.....	49
2.16.2 Validatinga Land Transaction in Blockchain.....	49
2.16.3 Updating a Public Ledger in Blockchain	50
2.17 Factom Blockchain Model for Honduran Land Registry System.....	51
2.17.1 Transaction Recording in the Factom Model.....	53
2.17.2 Validating transactions in the Factom Model	55
2.17.3 Updating a Public Ledger and Authenticating Transactions in the Factom Model.....	56
2.17.4 Inadequacies of the Factom Model	57
2.17.5 Authenticity of the Factom Model	57
2.17.6 Information Security Concerns in the Factom Model.....	58
2.18 Key management in the Factom Model	61
2.19 Digital Record-keeping and Crypto Currency in the Factom Model.....	61
2.19.1 Long-Term Digital Preservation	63

2.19.2 Preservation and Availability of Original Records	64
2.19.3 Digital Signatures	65
2.20 Blockchain and the Legal System	66
2.21 Conceptual Framework	69
CHAPTER THREE.....	72
RESEARCH DESIGN AND METHODOLOGY	72
3.1 Introduction	72
3.2 Research Design.....	72
3.2.1 Design Science and Proof of Concept Research Methodologies	73
3.2.2 Thematic Literature Review	73
3.2.3 Proof of Concept Approach	74
3.3 Proof of Concept Methodology.....	74
3.4 Proof of Concept: Model Design Stages.....	75
3.4.1 The Concept Stage	76
3.4.2 The Planning Stage	76
3.4.3 The Product Design Stage	76
3.4.4 The Design Verification Stage	77
3.4.5 The Design Validation Stage	77
3.4.6 The Design Transfer Stage.....	77
3.5 Benefits of the PoC Research Methodology	77
3.6 Limitations of Proof of Concept Methodology.....	78
3.7 Criteria-Based Evaluation Approach	78
3.8 Ethical Consideration	80
CHAPTER FOUR	81
MODEL DESIGN AND EVALUATION	81
4.1 Introduction	81
4.2 Model Design.....	81
4.2.1 Design Requirements	82
4.2.2 The Software Specifications	83
4.2.3 Expected Functional Requirements.....	85
4.2.4 Expected Non-Functional requirements.....	86
4.2.5 The User Interface Design	87
4.3 User Accounts	89
4.3.1 Land Seller Account.....	90

4.3.2 Land Buyer Account	91
4.4 Title Deed Registration	92
4.4.1 The Deed Registration Form Requirements.....	93
4.4.2 Title Deed Processing	95
4.4.3 Title Deeds Authentication	96
4.5 Land Transactions	98
4.6 Model Evaluation.....	99
4.6.1 Design Outputs.....	100
4.6.2 Model Evaluation Metrics.....	104
4.6.3 Evaluation Criteria	106
4.7 Results and Discussions	118
CHAPTER FIVE	122
SUMMARY, CONCLUSIONS AND RECOMMENDATION	122
5.1 Introduction.....	122
5.2 Conclusions.....	122
5.2.1 Challenges of Land registrations and Land transactions in Kenya	123
5.2.2 Requirements for the Model Development	123
5.2.3 Model Demonstration and Evaluation	124
5.3 Recommendations	124
5.3.1 Policy Recommendations	125
5.4 Recommendations for Future Research	126
REFERENCES	128
APPENDICES.....	138
Appendix I: Model Source Codes.....	138
Appendix II: Model Evaluation Tool (The Likert Scale Tool).....	339
Appendix III: Letter from Institute of Postgraduate Studies	341
Appendix IV: NACOSTI Reserach Authorization Letter	342
Appendix V: NACOSTI Reserach Permit.....	343
Appendix VI: Evidence of Conference Participation.....	344
Appendix VII: List of Publications	345

LIST OF TABLES

Table 1: A Summary Table of the Research Gaps Established by the Study	67
Table 2: Assignment of the Model’s evaluation Key Criteria	119
Table 3: Assessment of the current status of land registration and land transaction processes in the lands registry in Kenya against the Four Key Areas of the model evaluation used in the study	120
Table 4: A Summary of the Users’ autonomous evaluation of the secure title deed registration model for land transactions in Kenya based on Blockchain against the Four Key areas of model evaluation criteria adopted by the study	121

LIST OF FIGURES

Figure 1: Ardhisasa Platform.....	22
Figure 2: Ardhisasa Land Rent Payment Procedure.....	24
Figure 3: The Ardhisasa Platform	27
Figure 4: Microsoft Blockchain Architecture.....	31
Figure 5: Blockchain Transaction.....	34
Figure 6: Potential Industry Application areas of Blockchain	43
Figure 7: The conceptual Framework.....	69
Figure 8: Summary of Variables	71
Figure 9: The Proof of Concept IT Project Stages	76
Figure 10: Components of the Functional Requirements of the Model	85
Figure 11: Component of the Non-Functional Requirement of the Model	87
Figure 12: The Model's User Interface Registration Form	88
Figure 13: The Model's Sign Up Form	89
Figure 14: Land Buyer / Seller Accounts	90
Figure 15: A Land Seller A/C loaded with ETH.....	91
Figure 16: A Land Buyer A/C loaded with ETH.....	92
Figure 17: Land Seller A/C and Land Buyer A/C in Metamask	93
Figure 18: Title Deed Registration Form	95
Figure 19: Title Deed Registration BC030KBO578002 initiated from Land Seller A/C.....	96
Figure 20: Land Transaction Confirmation in Metamask	97
Figure 21: A Land transaction for registered title deed BC030KBO578002 in the Designed Model.....	99
Figure 22: Output Summary Report of All Land transaction Entries executed in the Model.....	101
Figure 23: The Land Buyer A/C in Metamask with 0.6886.....	101
Figure 24: The List and the Status of Land in the designed Model.....	102
Figure 25: Status of Land with Title Deed number BC030KBO578002	102
Figure 26: Cost of Land Transaction in Blockchain.....	103
Figure 27: Title Deed ownership transfer in the model, a Case of Title Deed number BC030KBO578002.....	104
Figure 28: Critical features of Blockchain Technology	106
Figure 29: Immutable Entries of Land Transactions Recorded in the Designed Model	109

Figure 30: A time-stamped Land transaction record whose title deed number is BC030KBO578002 in the model.....	110
Figure 31: All users can see the Land whose title deed number is BC030KBO578002 as available for sell in the Model.....	111
Figure 32: A Land sale transaction record executed in the model is available in the entire Blockain.....	111
Figure 33: The Block transaction details from the model can be extracted from the entire peer to Peer Network	112
Figure 34: Land whose title deed number is BC030KBO578002 has been bought by a buyer in the Model	112
Figure 35: A Land Purchase Transaction Executed in the model is recorded in the entire Blockain network.....	113
Figure 36: The Block transaction details extracted from the entire peer to peer Network	113
Figure 37: A Distributed Ledger of a Land sale transaction record in the entire Blockain. Component source codes in Appendix 1(G)	115
Figure 38: A Title Deed registration form for a Land Transaction	116
Figure 39: Entry of the title deed registration number BC030KBO578002 as Captured the Model	117
Figure 40: Land Transaction Authentication in the Model	118

ABBREVIATIONS AND ACROYNMS

AES–CMA	Advanced Encryption Standard – Cipher- Based Message Authentication Code
BCT	Blockchain Technology
CIA	Confidentiality Integrity Availability
CoK	Constitution of Kenya
DLT	Distributed Ledger Technology
GIS	Geographical Information System
GLA	Government Land Act
IA	Information Assurance
IBM	International Business Machines
ISO	International Organization of Standards
ITPA	Indian Transfer of Property Act
LCB	Land Control Board
LRA	Land Registration Act
LTA	Land Titles Act
MitMA	Man –in – the –Middle Attack
MoL	Ministry of Lands
NACOSTI	National Commission for Science, Technology and Innovation
NLC	National Land Commission
NLMIS	National Land Management Information System
NLP	National Land Policy
PKI	Public Key Infrastructure
RLA	Registered Land Act
RTA	Registration of Titles Act
SLCB	Special Land Control Board
USD	United States Dollar

OPERATIONAL DEFINITION OF TERMS

Land Transaction: This refers to the process of buying or selling Land such that title deed information changes from the seller to the buyer.

Title Registration: This is a system through which Land ownership is established by issuance of an official certificate having the profile details of the person owning it.

Ethereum: This is an open source software platform based on Blockchain technology that enables developers to build and deploy decentralized application in a network of peer to peer nodes.

Model: This is a conceptual representation of a system whose purpose is to help designers understand the user requirements of an actual system.

Blockchain: This is an electronic ledger of digital records, events, or transactions that are hashed cryptographically, authenticated, and controlled through a distributed or shared network of participants using a group consensus protocol.

Persistence: This is where transaction records are stored permanently as an immutable ledger containing history of transactions. No alteration or deletion of data, hence data or records cannot be corrupted.

Time- Stamping of Records: All transaction entries in the Blockchain are digitally recorded at the time of execution. This enhances the traceability of blocks.

Transparency: All transaction records are accessible to the network participants. They can therefore execute transaction entries based on consent granted. This enhances transparency.

Decentralization: This allows transactions to be executed directly by the users involved without the need of a third party. This enhances the integrity of the transactions executed as well as the integrity of the Blockchain system.

CHAPTER ONE

1.1 Introduction

The chapter begins by discussing the background information on the genesis of the registry of title deeds as a system for land administration and management. The challenges of Land transactions for Land ownership in Kenya are also highlighted as well an explanation on how an emerging Blockchain technology can be used to address the current challenges of Land ownership in the country. The chapter proceeds to present a brief discussion on the problem statement of the research, its general and specific objectives along with the research questions. Lastly, the Chapter highlights the research scope, assumptions, significance and its expected outcomes.

1.2 Background Information of the Study

1.2.1 Land in Kenya

United Kingdom introduced a registry of title deeds scheme in the year 1708 as an efficiency system for land administration and management. The system gained popularity and it was later on adopted and implemented by authorities across several European countries (Nishio, 1998).

Kenya, being a former colony of the United Kingdom, adopted a similar scheme in the management and administration of land resources. Over the years, the registry of title deeds system largely remained manual. Title deeds and other related land transaction records were managed manually and as a result, these records became highly susceptible to all forms of manipulations. The ripple effect of such a system paved the way for a compromised, inefficient, time consuming, unpredictable, inflated and ineffective land administration and management processes that relied on a centrally managed registry of title deeds by the Ministry of Lands in Kenya (Gillies, 2019).

To address these challenges, the Government of Kenya in the year 2007, through the parent Ministry of Lands, made an attempt to introduce the automation of Land records and transactions by developing and deploying a National Land Information Management System (NLIMS) based on big data technology. The objective was to implement a system that has the capability to hold immense and varied data sets on Land ownership and Land transactions. This decision to automate Land records and Land transactions was informed by the need for the Ministry of Lands to comply with the provisions of the country's National Land Policy which was launched in 2009 (Kwanya, 2014).

In Kenya, the decisions on how public land can be used or its management remains a contentious issue. In the public domain, the decisions seem to be generally influenced by political considerations since policies governing land acquisition and usage seem to be implemented outside the established management procedures by Government agencies (NLC, 2014).

1.2.2 Challenges Facing Land Administration and Land Transactions in Kenya

The Government of Kenya in 2021 launched an integrated National Land Information Management System (NLIMS) called Ardhisasa, jointly developed by the Ministry of Lands and The National Lands Commission (NLC, 2021).

The objective of Ardhisasa is to digitize land records, an ongoing process in the land's registry in Nairobi. However, the process of digitizing the records is hampered by an array of challenges in the Land's registry. The challenges relate to matters to do with overlapping institutional responsibilities, absence of proper documentation of Land records, missing Land records in the main registry, illegitimate and non-procedural Land transfers, more than double allocation of the same Land with multiple title deeds issued to it as well as corruption in Government (MoL, 2021).

All these challenges may not be exhaustively addressed by the NLIMS. The inadequacies in Land transactions, difficulties in proof of Land ownership that end up in endless court proceedings, lack of accountability and transparency in the issuance of Land title deeds among many other challenges relating to the administration and management of Land in Kenya seem to persist (Unwomen, 2021).

1.3 Blockchain Technology

Blockchain is a decentralized solution which does not require any third party organization in the middle. A block refers to the current part of a Blockchain, which records some or all of the most recent transactions. Once completed, a block goes into the Blockchain as a permanent database. Every block contains a hash of the previous block. The Blockchain has complete information about different user addresses and their balances right from the genesis block to the most recently completed block (Kshetri, 2017).

Blockchain Technology (BCT) uses the public key infrastructure (PKI) mechanism. In PKI, the user has one pair of public and private keys. The public key is used in the address of the user wallet, and the private key is for the authentication of the user. All the participating nodes store information about all recorded transactions of the network and confirm the correctness of each new transaction made by using previous blocks. The new blocks are linked to previous blocks and all the blocks are aligned in one continuous chain. This chain of blocks is the public ledger technique known as Blockchain (Wiley, 2004).

Blockchain allows the creation of a digital ledger of transactions which is shared across a peer to peer network. Seemingly similar to the distinct elements of a trade finance transaction, Blockchain has the power to synchronize records of several parties across

the supply chain, allowing the exchange of data values without the need for a central authority (Wright, 2015). These attributes make the Blockchain technology a more secure, reliable, transparent and appropriate for the execution of Land transactions (Kshetri, 2017), compared to the existing largely manual Land's registry in Kenya or the NLIMS which is still under deployment.

1.4 Statement of the Problem

A Kenyan investor or a citizen or anyone else interested in a land transaction for land ownership has to engage in a lengthy process involving different entities or agencies at any given stage and at any given time. Sometimes the Land buyer or seller is subjected to very long queues at various stages, endless paperwork, delays occasioned by absence of registry officials or missing documents during the process of executing a Land transaction. With Such kind of experience a Land buyer or seller may get frustrated and opt for an alternative means and ways of executing the land transaction, for example the use of a middleman (a land broker or a lawyer) handle the extensive Land transaction process.

More often, the middlemen, driven by personal interests and motivated by the need for a marginal financial gain out of the entire transaction process, complicate the land ownership procedure due to lack of transparency. The outcome of such a land transaction would become a potential genesis of interminable land conflict in the country. Existing gaps in laws governing land ownership, unethical civil servants and Land politics coupled with inefficiencies in the largely manual nature of both the Land registry as well as the non-universal nature of the NLIMS have become a recipe for unlawful Land transaction activities across Kenya (MoL, 2021).

To address these endless challenges experienced in Land transactions for Land ownership, the study investigated how Blockchain technology can provide the much

needed authentic account pertaining proof of Land ownership as a result of a successful execution of a Land transaction. Given that the Blockchain is a digital database distributed across a network of computers, Land transaction records are protected by codes using cryptography, making the records to be temper-proof/non-immutable in nature. This does not only increase security of Land transaction records but it also goes a long way in fighting corruption by distributing the maintenance of Landrecords and related Land transaction information to all parties involved, rather than to a central registry(Kshetri, 2017). By allowing participants to see who owns what Land and who sells what Land, Blockchain enhances verification and transparency on Land transactions (Njoroge, 2019).

1.5 Objectives of the Study

1.5.1 Overall Objective

The overall objective of the study is to design and evaluate a secure title deed registration model for Land transactions in Kenya based on Blockchain.

1.5.2 Specific Objectives

These are the specific research objectives of the study:

- i. To review the challenges of Land transactions for Land ownership in Kenya.
- ii. To design a secure title deed registration model for land transactions in Kenya Based on Blockchain.
- iii. To demonstrate how the designed model integrates Blockchain to provide a secure platform for title deed registration for land transactions.
- iv. To evaluate the output of the designed model against the evaluation metrics set out by the study with the view of recommending it for future implementation in Kenya.

1.6 Research Questions

The study seeks to address the following research questions:

- i. What challenges exist in Land transactions for Land ownership in Kenya?
- ii. How can a secure title deed registration model for Land transactions in Kenya based on Blockchain be designed?
- iii. How can a secure title deed registration model for land transactions in Kenya based on Blockchain be demonstrated?
- iv. What requirements are needed for a successful evaluation of the designed model as well as the expected outcomes to inform the model's recommendation for future implementation in Kenya?

1.7 Significance of the Study

The Land's registry in Kenya is largely a manual collection of Land records. The existing NLIMS/Ardhisasa database is centralized and therefore making it an easy target of manipulation. There is need for a secure, transparent and a decentralized model to be integrated with the existing NLIMS. With respect to that, a secure title deed registration model for Land transactions in Kenya based on Blockchain becomes necessary. Blockchain is decentralized and distributed digital ledger that can keep a record of all Land transactions across many computers. The Land transaction records cannot be altered retroactively without the alteration of all subsequent blocks in the entire network, thus providing the much needed transparency and security of Land records.

The study similarly contributes to the body of knowledge and may therefore be used by both current and future scholars and researchers as a reference literature in areas relating to the use and application of Blockchain technology in various Industry sectors.

1.8 Scope of the Study

The primary focus of the study was to design a secure title deed registration model for land transactions in Kenya based on Blockchain. This was to demonstrate how a land title deed can securely be registered in the model and used by a land buyer/seller to execute a secure Land transaction as a block that can be shared in the entire Blockchain network. The study adopted four critical benefits of Blockchain namely: Persistence, Time-stamping, Transparency and Decentralization to establish that Land transaction records executed in the designed model demonstrate those blockchain features and therefore the scope of evaluating the model can be limited to that.

1.9 Justification for the study

In Kenya, fraudulent Land transactions emerge almost on a daily basis. Land administration has been complicated by corruption, gaps in the Land laws and inefficiencies in the current NLIMS. Integrating Blockchain in Land transactions for Land ownership provides a platform with enormous benefits to the integrity of the Land registry in Kenya. Blockchain uses cryptography to secure Land transaction records across the decentralized database thus preserving the authenticity and the integrity of Land transaction records. The cryptographic controls provided by the model guarantees confidentiality, integrity and availability of Land transaction. The concept of a transparent, decentralized public ledger could easily apply to Land transactions for Land ownership, where the Land registry serves as a database of all property rights and historical transactions. This ensures that all transactions related to Land in Kenya are highly secured.

1.9 Limitations of the Study

The research was limited to the design and evaluation of a secure title deed registration model in Kenya based on Blockchain using Design Science and the Proof of Concept methodologies. For secure Land transactions to be successfully executed in the model, Ether is required, a scarce crypto currency that must be acquired at a cost. This makes the entire study an expensive undertaking. It's likely that this may impede the full exploitation of Blockchain in both current and future research. Time and privacy constraints were also encountered during the study.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

This chapter presents an overview discussion on the literature review relevant to the study. Areas covered included the registration and titling of Land in Kenya, the NLMIS and the challenges of Land management and Land transaction processes in the Kenyan context. Discussions on Blockchain technology as a solution for secure, efficient and effective execution of Land transaction is presented as part of the literature reviewed by the research. A brief presentation on the case of Factom Blockchain model for the Hondurian Land registry is also made. The chapter concludes by briefly discussing the research gap and the conceptual framework that informed the study.

2.2 The Registration and Titling of Land in Kenya

In Kenya, Land management used to be under customary laws whereby Land ownership was managed by the native communities and no single person could claim ownership to Land. But things changed with the arrival of the colonial masters in the country who were of the view that customary tenure arrangements practiced by the native people in African was not consistent with development and modernization as was the case in their homeLand. They then sought to establish a tenure system which only accorded recognition to Land rights secured by individual freehold title as a formal arrangement of Land ownership (Okoth-Ogendo, 1991).

The colonial government viewed African customary Land tenure as an impediment to greater agricultural production and proper Land use practices and as a result they introduced various statutes to govern Land administration and registration of title deeds in Kenya (Hanstad, 1998).

Land ownership in Kenya is greatly significant. It is a critical factor of production which is the backbone to the country's economy due to the fact that Kenya's economy relies on Agriculture and its related activities. Therefore, Lack of Land will practically have a ripple effect on the economy of the Kenyan households or individuals. As a consequence, every individual dreams to own a piece of Land by whatever means. However, there exist persistent challenges, usually attributed to the processes involved in Land transactions that lead to Land registration and title registrations for ownership (Makoro, 2015).

2.3 The Registry of Title Deeds

The Registry of Title Deeds system was introduced in England in the year 1708. This system was implemented in several European countries to prevent double selling of Land. A deed registration system meant that the deed itself, being a document which described an isolated transaction, was registered. However, the deed was not in itself proof of the legal rights of the involved parties and consequently, it was not evidence of its legality. Thus, before any Land dealing is sealed, the apparent owner must trace his ownership back to a good root of title. The deeds registration system was hailed as having had the advantage of providing a maximum degree of elasticity as almost every conceivable right or claim to a right in Land could be registered thus achieving public participation (Feder, 1998).

The system was associated with a major disadvantage in that no inquiry is made into the authenticity of the deed either as to form or as to content. It was left to the prospective purchaser of Land to investigate all deeds in the chain of the titles back to the grant title before concluding his purchase. Other weaknesses with the deed system were that, Lands were not properly surveyed and demarcated, giving rise to inaccurate plans or maps which often created conflicts among Land owners. Moreover, registration was based on

the deed and not on the Land leading to multiple registration of the same piece of Land as there was no system to detect multiple registrations in the registration process (Williamson & Ting, 2001).

2.3.1 The Torrens System

The Torrens system was first introduced in Australia, in 1858, by Sir Robert Torrens who was then the Register General of the Province of South Australia. He believed that a Land register should show the actual state of ownership, rather than the mere evidence of ownership. Shortly after Torrens introduced the concept of title registration in Australia, a similar modified version of the Torrens system was adopted in England. The system ensures security of title under a system of State guarantee whereby the register itself provides proof of title to the Land (Zevenbergen, 2014).

According to Taylor (2008), the Torrens system is said to encompass three principles which are: The mirror principle under which the register of title is a mirror reflecting accurately and completely the current facts that are material to title. With certain inevitable exception, the title is free from all adverse burdens, rights and qualifications unless they are mentioned in the register. Under the curtain principle, the register is regarded as the sole source of information and intending purchasers need not concern themselves with trusts and equities which lie behind the curtain. Finally, the insurance principle explains that if the register does not give the correct reflection of the title and a flaw appears, anyone who thereby suffers loss must be indemnified by the government (Taylor, 2008).

Registration of title was preferable to recording of deeds, since the former aimed at presenting the prospective purchaser or mortgagee the net result of all the previous dealings with the given property, while the latter presented the dealings themselves

before such prospective purchaser who was left to investigate for himself (Steudler, 2004).

In registration of title system, the register is a conclusive evidence of previous dealings over the Land, while in registration of deeds only the figures were given and the party in interest was left to work out for the final answer to himself and at his own risk (Griggs, 2016).

2.4 The Classification of Land in Kenya

Land in Kenya is divided into three major categories under the Lands Act of 2012:

- i. **Public Land:** Under Article 62 of the Constitution of Kenya (CoK, 2010), public Land contains alienated Land, Land engaged by a State system, Land passed on to the State, Land lacking recognized beneficiaries, substances, ecosystems, reservations, national parks, areas for collecting water, sea, rivers and lakes, Land among the highest and lowest water marks, and any Land that is not classified as private or society land. The National Land Commission is charged with the administration of public lands.
- ii. **Community Land:** As stated in Article 63 of the Kenyan Constitution (CoK, 2010), social property is land that is legitimately held in the hands of category officials, land that is legitimately assigned to the members of a particular community, and any property that has been stated as society land by an Act of the legislature. The law approved under the Constitution will eventually govern community property ownership. However, the necessary law has not yet been adopted, and the founding document provides five years for its passage.

- iii. **Private Land:** Land established under freehold appointment, Property held over leased ownership, and any other Land proclaimed private by an Act of Parliament falls into the abovementioned groups. It is also feasible to convert Land between these types.

2.5 The Land Laws in Kenya

Before the enactment of the Land Registration Act 2012 (LRA), the registration of title to Land in Kenya was characterized by multiple statutes such as the Government Lands Act (GLA), Land Titles Act (LTA), Registration of Titles Act (RTA) and Registered Land Act (RLA). The array of Land laws has led to a complex registration regime often leading to complex Land transaction processes, duplication of titles deeds and fraudulent dealings. The LTA and the GLA were a deed registration system and were characterized with many problems including the tattered and ineligible registers, missing records and a complex registration regime. The RTA and RLA were also faced with problems such as the duplicity of titles, fraudulent titles and multiple boundary disputes that usually end up in violence that lead to loss of lives (CoK, 2010).

In an attempt to address these challenges in Land and title registration, the Land Registration Act, 2012 (LRA) was enacted to revise, consolidate and rationalize the registration of titles to Land to give effect to the principles and objects of devolved government in the processes of Land registration. The intention is to provide a simplified, secure and harmonized registration regime. Whereas the LRA may have succeeded in harmonizing registration statutes by recognition of the registers and titles under the repealed statutes, one of the questions that arise is whether this harmonization has enhanced efficiency, transparency and accountability in Land and title registration process in Kenya (Wanjala, 2000).

2.6 The Challenge of Land Administration in Kenya

Since independence, successive Kenyan governments have been trying to address Land issues. Unfortunately Land has continued to remain an emotive issue that emerges in almost all civic and economic developments in the nation. As at today, very few influential people own expansive Lands while the majority of the citizens live as squatters, even in their very own ancestral Land. The citizens are unable to enjoy the socio-economic benefits that come with Land ownership in a country whose economic backbone is agriculture. Such scenarios have in time and again generated ferocious cycles of relentless struggles for Land leading to endless tribal conflicts, loss of lives as well as property in many parts of Kenya. The Government of Kenya together with other stakeholders have made several efforts to address the Land issue through mechanism such as Land legislation, demarcation, rationalization and issuance of Land titles to the Landless with little allocation of resources going into the development of a National Land Information Management System (Mulaku, 2005).

The situation in Kenya is that we have many Land laws in place, some of which are incompatible or overlapping. This has resulted in a complex Land management and administration system with lots of disparities in the Lands registers. The non-existence of an efficient, transparent and secure NLIMS has contributed to numerous Land related challenges such as deterioration in Land quality, squatting, landlessness, disinheritance of some native groups and individuals, underutilization of land, abandonment of agricultural Land, tenure insecurity and communal conflicts relating to Land ownership. Land impostors and brokers masquerading as genuine land owners have taken an advantage of gaps in land administration to engage in illegal Land transactions across the Kenya (MoL 2009).

Without land, households and individuals become economically vulnerable. Subsequently, more Kenyans work extra hard on a daily basis to acquire land by any means possible including engaging in criminal undertakings such as falsifications or unauthorized modification of Land records and transactions. To address these challenges, the Government of Kenya in 2007 resolved to automate all Land records and transactions by developing and deploying a National Land Management Information System (NLIMS). The NLIMS is a big data system capable of holding and supporting vast and diverse data sets on Land ownership and transactions (MoL, 2007).

2.8 The Land Transaction Process in Kenya

In Kenya, due diligence must be followed during Land transactions to avoid human conflicts arising out of errors committed during a Land transaction or title registration process. Muli (2017) explains that prior to the buying or selling of Land in Kenya, an investor must always ensure that they familiarize themselves with the following laid down procedures provided for by the Ministry of Lands:

- i. **The Land Searches:** Due diligence requires that an individual who wants to buy or sell land in any part of Kenya must conduct a land search. The process involves the physical querying and searching of land records at the Registry of Deeds in the Ministry of Land or at the County level to establish the ownership of land in question before the execution of a land transaction.
- ii. **The payment of Land Rates:** This is a levy applied to individuals owning land in Kenya. Annually, land owners are legally obliged to remit a levy referred to as a land rate paid to the county government from where the land is located. The rates vary from one county to another.

- iii. **Mapping:** Land mapping is usually done by a surveyor. It's therefore recommended that a land buyer or seller acquires the land map showing the physical location and measurements of the land.
- iv. **Land Verification:** Once a land buyer or seller obtains the land maps, a visit to the physical location of the land is necessary for the sole purpose of confirming the existence of the land alongside the demarcations.
- v. **Land Transaction Agreement:** The agreement refers to a manual document drafted by a lawyer to be signed by all land transaction participants.
- vi. **The County Land Control Boards:** This is a stakeholder's forum comprising the government officials and the members of the community who get involved in giving the final consent involving a land transaction between individuals by ensuring that the transactions are genuine and transparent.

2.9 The Title Deed Transfer Process in Kenya

According to the Laws of Kenya, The Land Act, 2012 under Section 7 outlines seven ways with which an individual or investor may use to acquire a title deed for a given piece of Land located anywhere in the country. The seven ways are explained below:

- i. **Allocation:** This is where public Land is transferred by the government to individuals usually for a specified time and for a specific use with stated conditions. Allocation can be through a public auction, tender, drawing of lots or balloting or by confining allocation to a disadvantaged group. For Land to be allocated, it must first be planned, surveyed and serviced.
- ii. **Land adjudication:** This is when rights and interests to Land are ascertained and recorded in areas of community Land (formally known as the trust Land areas).

- iii. **Compulsory acquisition:** This is the acquisition of Land by the government for a public purpose but subject to fair and prompt compensation. Publication of the intention to acquire is done through the Kenya Gazette and County Gazette.
- iv. **Prescription:** This is also known as adverse possession and occurs when one gains title to real property through occupation of Land without opposition for 12 years. It involves a court process and the adverse possessor must prove that she actually occupied the Land continuously without interruption for the above specified time.
- v. **Settlement Scheme:** This is when the government provides access to Land for squatters, persons displaced by natural courses like floods, development projects, conservation or internal conflicts by use of the Land Settlement Fund administered by the National Land Commission. Formerly it was known as Settlement Fund Trustees.
- vi. **Transmissions:** This is when ownership passes to another party whose powers over the Land are restricted as in cases of death, bankruptcy and on liquidation of a company. In case of death, a personal representative will be registered as the proprietor of the Land, for bankruptcy a trustee becomes the registered proprietor while a liquidator shall be registered as the proprietor in cases of liquidation of companies, all by order of the court.
- vii. **Transfers:** This is the transfer of rights to Land through sale. A transfer is done after acquiring consent from the relevant County Land Management Boards in (all 47 counties). Stamp duty of 2 per cent of the value of the Land is payable to the government in townships and rural areas while a duty of 4 per cent is

charged on Land in municipalities and cities. The transfer document with the original title are then booked for registration in favor of the buyer.

2.9.1 The Categories of Land Title Deeds in Kenya

According to Investopedia.com (2018), a title is a legal document that supports an individual's right to ownership and possession of Land that can be recognized as being owned or belonging to a particular person or entity. It is a document that indicates recognition of ownership and it may be obtained as a result of purchase, descent or grant.

In Kenya, there exist different types of title deeds as per the various types of Land Acts that have been in existence in the laws of the Land. They include the following:

- i. **Certificate of Lease:** This is a title under the Registered Lands Act Cap 300 (repealed) for leasehold Land. Leasehold is a form of Land tenure where a lessee (occupant) holds rights to Land for a specific period and subject to conditions imposed on Land rights by the lessor (Landowner). The period may be for example, 99 years.
- ii. **Absolute Title:** This is a title under the Registered Lands Act Cap 300 (repealed) for freehold Land. This is a form of ownership whereby the Land owner has the maximum rights in terms of period of ownership and use of the Land.
- iii. **Sectional Title:** This is a title for a unit within a building, for example a flat. It emanates from the Sectional Properties Act of 1987 and the title is deemed to be registered under the Registered Lands Act Cap 300. It is important to note that the Land Registration Act, 2012 in Section 26 consolidates the above several titles into the Certificate of Title. The aim is to rationalize and reduce confusion brought about by the previous Land Acts, which resulted in

various types of titles existing concurrently, causing disparities in the Land registry.

2.10 The Status of the Lands Registry in Kenya Today

The current status of the Lands registry in Kenya is generally manual in nature. Subsequent Governments in the Republic of Kenya have all long been working towards the digitization of the Lands registry with the view of improving on the efficiency and the effectiveness of service delivery in the Ministry of Lands. This commitment to change has been explicitly mentioned in two previous National Development Plans, and it remains a significant objective outlined in the proposed National Land Policy Declaration of the year 2007 (MoL, 2018).

According to the Ministry of Lands (2018), the existing manual registration system is encountering several difficulties, which are listed as follows:

- i. Land information is stored in manual format. Data can be found at a variety of facilities in a variety of entities and regions. Access to information requires an extended period and is usually given less efficiently. The Ministry of Lands considers the scenario to be confusing and time-demanding.
- ii. According to the Ministry of Land, Housing, and Urban Planning, the outdated manual process has encountered issues to do with Land title documents. It has made it simpler for fraudsters to obtain genuine property data from Government facilities.
- iii. Documents preserved at several government offices around the country are duplicated.
- iv. Manipulating information from different sources to improve research and the discovery of unexpected linkages across collections of data is challenging.

- v. Citizens, alongside other property owners have raised the need for land information. Since there is inadequate documentation, this need causes even more challenges with accessing information.

In the opinion of the Ministry of Lands (2018), the non-existence of a universal National Land Information Management System has resulted in numerous unresolved issues involving Land administration in Kenya. These challenges are highlighted as follows:

- i. Inadequacy in the organization and coordination of land usage to promote sustainable development.
- ii. The instability of land ownership rights, particularly impacting vulnerable segments of society like squatters, women and children.
- iii. Unequal availability of land resources, specifically disadvantaging indigenous populations and minority communities.
- iv. Poor disaster mitigation planning; ineffective preparation for emergencies.
- v. Lack of efficient monitoring for environmental matters; inadequate supervision of ecological concerns.
- vi. Insufficient backing for a robust credit market, particularly concerning mortgages.
- vii. Poor governance of Government owned Land regarding allocation and development.
- viii. Ineffective monitoring of squatter activity on government land and informal settlement development.
- ix. Non-existence of equitable and comprehensive recording of informal and user rights on land leads to uncertainty and potential conflicts.
- x. Absence of a secure Land management data system that provides an accurate appraisal and evaluation of land over revenue.

2.11 The National Land Information Management System

A well-functioning property management system necessitates technology that allows for the transfer of property-related data among government organizations. Especially the administrative links present among those keeping Land files, which include the database of titles and the individuals who perform Land surveys (Dale, 1988).

Cadastral schemes are employed to give data on numerous purposes of Land in society. Information regarding land sales and ownership should be readily available and accessible for public examination. The main objectives of a comprehensive land rights registration system are as follows: (1) to provide easily accessible land maps for investigation whenever required, (2) to preserve accurate records that contain details of all officially recognized entitlements and active customs associated with the Land (3) to establish the present owners or organisms with whom the rights in question are invested, and (4) to make it easier the rapid and precise recognition of particular records in the maps and records co-responsible for the authorization. In essence, the system should aim to increase the accessibility of evidence, ensure transparency in legal matters relating to land transactions and enable precise verification of property rights and ownership details (FAO, 1995).

The accuracy and swiftness with which alterations are documented and reported will determine whether a LMIS succeeds or fails. Therefore, any modifications that influence records in the files, especially alterations to the Land transaction records, the terms under which the Land transactions are executed as well as the ownership rights should be captured in the Land's registry of title deeds database (Wanjala, 2000).

2.11.1 The Ardhisasa Platform

According to the Kenyan Constitution (2010), every citizen has a right to property ownership and by extension, the right to acquire and own Land in any part of the Kenya on condition that there is compliance to the legal provisions for Land acquisition, including the Land Control Act. Due diligence must be followed to the end while performing a Land transaction and Muriuki (2019) explains that a complete Land transaction procedure involves the following stages.

Figure 1

Ardhisasa Platform



i. **Identification of Land for Purchase**

This is the first stage in a Land transaction procedure whereby the potential buyer, either on his/her own or through an agency firm, identifies a suitable Land to purchase from a seller. Land Transaction stages.

ii. **Land Search at the Ministry of Land Registry**

Once the prospective buyer has identified the Land that meets his/her desired standards, the next stage is to carry out a Land search at the Lands Registry. This is done by obtaining a photocopy of the original Title Deed from the seller to facilitate the process. Currently, an online platform referred to as Ardhisasa, allows Citizens, stakeholders and interested parties to interact with Land information held and processes undertaken by

Government (Ardhisasa, 2022). Alternatively, a physical search can be done which is usually a process that may take up to two hours to get search results upon filing a search application form and payment of **Kes.500** search fee. A valid search should not be more than six months old. The search results should indicate the Land details such as the registered owner, Land size as well as any caveats registered against the title deed.

iii. **The County Office Land Search**

The rates are normally factored into the buying price of the Land on sale. In Nairobi County, one has to pay **Kes.10, 000** to get a certificate of clearance. However, this may vary from one county to another. Payment of Land rates is a legal duty of a Land seller and it is mandatory for them to clear any pending rates and penalties, if any, before putting up their Land for sale (Ministry of Lands, 2021).

iv. **Land Rent Clearance Certificate**

This is the fourth stage whereby the seller's Attorney has to obtain a Land rent clearance certificate from Ministry of Lands, Kenya with some costs. This process may usually take up to 19 days in some counties. However, on the Ardhisasa platform, the process is more efficient as shown in Figure 2.

Figure 2

Ardhisasa Land Rent Payment Procedure

The graphic is a rectangular box with a brown header and footer. The header contains the title 'How To Pay Land Rent on Ardhisasa' in white text, the Kenyan coat of arms, and the Ardhisasa logo with the slogan 'Shamba Lako, Hati Safi!'. The main body is white with a list of 13 numbered steps. Step 1: Go to ardhisasa.lands.go.ke. Step 2: Register an account to begin transacting. If you are a registered user, simply log in to your account and transact. Step 3: Go to Services. Step 4: Under Land Administration, click on Land Rent. Step 5: Then click on Pay Land Rent. Step 6: Enter Title Number on the Search section using the following formats: Registration Unit/Registration Section/Parcel Number; - For Nairobi Blocks: Nairobi/Block12/345 or Block No/Parcel Number (for example: 12/345) - For Mombasa Parcels: Mainland North (or South)/Roman Number/Parcel Number (for example MN/IV/123) - For other towns and municipalities: Kiambu/MunicipalityBlock1/234 Step 7: After searching for the parcel, pending payments will be listed Step 8: Click on the View button to see the invoice details, which includes the land rent owed and penalty, if any. Step 9: Click on Pay to make the payment Step 10: Choose a Payment method (Co-operative Bank, M-Pesa or M-Pesa Express) and follow the instructions given to make the payment. Step 11: You will receive a payment Confirmation message on your phone. Step 12: Click on Confirm on the Ardhisasa platform to complete the transaction. Step 13: The payments will reflect on your account instantly! A yellow highlighted box contains a NOTE: If you have several pending land rent payments, you will be required to pay each one of them separately, since each has a unique invoice number. You will ONLY receive a Clearance Certificate once you have settled all pending payments. Step 14: Download the Clearance Certificate or Print it directly from the site. Below the steps, there is contact information: Twitter: @Lands_Kenya | Facebook: Ministry of Lands and Physical Planning | www.ardhisasa.lands.go.ke | Call: +254 773 641 455 | +254 732 682 593 | +254 743 922 876

v. **The Land Maps**

Two Land maps should be obtained at this stage. One map has an overview of the Land while the other is the Land map drawn to scale. The two Land maps are acquired from Ministry of Lands, Kenya or from a surveyor at a fee of **Kes.300** each. Once the maps have been obtained, the Land buyer and his/her surveyor arrange for a physical visit to the location of the Land for verification and marking of the Land boundaries using. This stage brings to an end the strenuous due diligence process involving a Land transaction in Kenya.

vi. **Price Offers and Negotiations**

Upon the buyer being satisfied with the details of the Land based on information obtained from previous stages, he/she may request his/her attorney to prepare a purchase offer for the Land. Details of the purchase offer should capture information about the buyer and seller, Land description, the proposed buying price and preferred mode of payment.

vii. **The Sale Agreement**

To facilitate the Land transaction, the Land seller's Attorney is required to prepare a sale agreement that captures information about the buyer and seller, Land price and the preferred mode of payment. This sale agreement is supposed to be signed off between the buyer and the seller in the presence of their Attorneys.

viii. **Clearance from the Land Control Board (LCB)**

Land Control Board (LCB) is a forum comprising of community leaders and elders and the County Commissioner. The mandate of the LCB is to certify that the Land transaction is transparent.

ix. **The Transfer Consent**

Once the Land transaction forms have been signed off by both parties, the Land buyer with all the required documents may visit the offices of the National Land Commission (NLC) for an application of consent to transfer. This can as well be done through the Ardhisasa platform; however the online platform may not apply in some Land transactions with no title deeds.

x. **Stamp Duty Valuation and Payment**

The Land valuation forms are filled by the Land seller. The forms are then supplied to the buyer who will in turn use them to apply for valuation of the Land by a Kenyan Government valuer (Ministry of Lands, 2020). The generated valuation report is then used to compute the required taxes for the Land – popularly known as stamp duty. The stamp duty is payable to the Commissioner of Domestic Taxes at 4% of the total Land value for Land in areas classified as urban and at 2% for Land in areas classified as rural. As of January 2021, the Government of Kenya amended Section 10A of the Stamp

Duty Act, Cap 480 to allow private valuers to undertake valuation for stamp duty (KRA, 2021).

xi. **Transfer Registration**

Upon payment of the stamp duty and settlement of the Land transactions, the stamped documents are then taken to the Lands registry for verification and transfer of ownership. This stage will then mark the end of the Land transaction procedure and the said Land will be considered to have been legally transferred from the seller to the buyer.

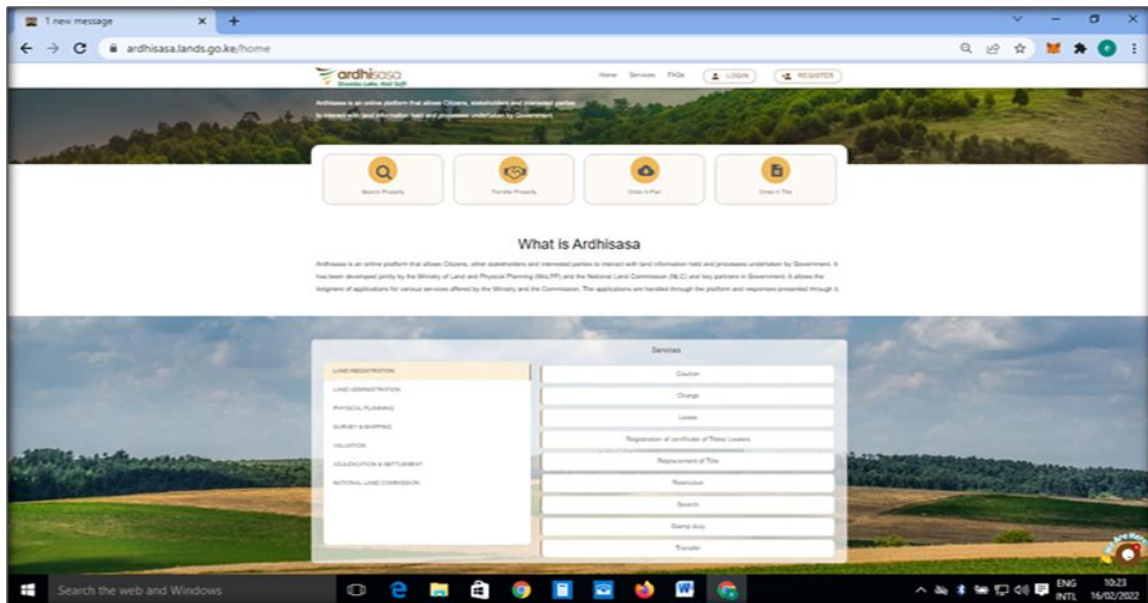
2.11.2 Ardhisasa Implementation Issues

On the 27th of April, 2021, the Government of Kenya launched its National Land Information Management System (NLIMS) known as *ArdhiSasa*. According to the Ministry of Lands (2021), the NLIMS was designed to enhance the security of Land records, improve accessibility and cut down the cost of Land transactions, boost service delivery and eliminate paper records that breed and catalyze corruption and fraud.

They pointed out that Poor record management and bureaucratic processes are some of the hindrances to doing business in Kenya. Therefore, the NLIMS objective was to First-track search, registration, valuation, transfer and issuance of titles in order to facilitate and accelerate investment and development of Land as capital. They went ahead to explain that **ArdhiSasa** will improve public service delivery, foil fraud and stop further deterioration of paper records. It will also improve access, storage and retrieval of Land records (Ministry of Lands, 2022).

Figure 3

The Ardhisasa Platform



However, according to the Ministry of Lands (2021), the Ardhisasa system is currently live only in the Nairobi Registry and plans are at an advanced stage towards its implementation in the whole country by end of the year 2022.

2.12 The Concept of Blockchain Technology

Blockchain technology has become significantly a disruptive concept, attracting in an equal measure the industry practitioners and business executives from various sectors of the global economy. It is synonymous with a publicly available distributed ledger and presents a unique decentralized data management strategy that safeguards the integrity of various transactions. The technology was initially known for powering crypto currency, but its revolution since its development in 2008 has sparked increased interest and acceptance simply because of its ability to generate confidence among members in a distributed system, hence bypassing the need for an external middleman (Swan, 2011).

Blockchain is a decentralized method used for issuing and transferring money. It operates a public ledger containing a record of all executed transactions, free from any third-party control. The immutable state of transaction records executed in Blockchain is a critical advantage of the technology. Once a transaction has been successfully executed in it, the record about that particular transaction cannot be altered or deleted upon approval by all participating nodes in the peer to peer network. This attribute enhances data integrity and security. Additionally, the technology's versatility enables it to be applied in various sectors of industry such the facilitation of digital smart contracts and peer to peer sharing of data in cloud services. These benefits of Blockchain technology is the driving force behind its adoption in diverse services and applications (O'Reilly, 2015).

In contrast to conventional transactions which typically involve centralized monitoring through a third party entity, Blockchain transactions operate in a decentralized manner. In conventional systems, the responsible entity verifies the feasibility and occurrence of the transactions. Physical assets like real money possess intricate security features and cannot exist in two places simultaneously, but digital data can be easily copied or intercepted.

Therefore, an intermediary such as a bank is required for digital payments or to digitize products like software licenses or music files. Notaries and public authorities often need to verify and register various interactions. This secondary relationship between the two primary organizations may become expensive and time consuming, and it constitutes one single point of vulnerability in centralized systems (Bertino & Sandhu, 2005).

Rather than relying on a single authorized ledger to confirm the occurrence of a transaction, the concept of a shared ledger involves replicating its contents across thousands of nodes. Trust shifts towards the consensus of multiple copies, with most

ledgers prevailing over corrupted or manipulated information. Consequently, central authentication of ledgers is optional. However, while data decentralization is feasible, an overarching entity must still determine which information should be rightfully stored. From a technical standpoint, it is crucial to establish a systemic truth to update the entire network (Mainelli & Milne, 2016).

Antonopoulos (2015) explains that decentralization relies on replication and a distributed ledger is the functional infrastructure among equipotent participants. Only in the Whitepaper in 2008 did a practical concept emerge to establish trust between unknown participants. Blockchain is a ground breaking technology that successfully constructs a fully public and unrestricted distributed ledger. Its ability to build confidence between unknown parties is predicated on the confidentiality of data and the structure, generation and distribution of information. This technology results from decades of research and is supported by four essential components:

- i. A network of peers to peer is the cornerstone of a publicly distributed ledger's database design.
- ii. To ensure the confidentiality of payments between anonymous accounts, transaction semantics employs encryption and digital signatures.
- iii. Data fixity is achieved by securely linking data blocks within a chain in consecutive order and cryptographically sealing and protecting each block individually.
- iv. A global selection is enabled via an algorithm, allowing users to form a consensus on the present state of the network's operations and reconcile the shared database.

While "Blockchain" and "distributed ledger" are sometimes used collectively, it is crucial to clarify that a shared ledger method does not rely simply on implementing Blockchain technology. Today, advancements in Blockchain technology allow for various distributed database solutions that can store an expanding list of transaction records. These solutions can digitally maintain exchanges of assets, physical objects or funds. The system ensures that everyone confirms every transfer as the data records undergo consistent validation. As a result, the need for third parties or intermediaries is eliminated. While Blockchain contains information about all past transactions, it guarantees complete data integrity (Kelly & Williams, 2016).

2.12.1 The Blockchain Architecture

In response to the demand from enterprise customers, including major players in the Financial Services industry, Microsoft developed an open platform for recording transactions in digital, distributed ledgers. Due to the use of cryptography, these transactions are considered tamper-proof (Microsoft 2016). Microsoft has initiated Project Bletchley to facilitate customer adoption, which addresses various architectural challenges customers face. Some of the themes Microsoft is focusing on include:

- i. Platform openness is essential.
- ii. Integration of features like identity, key management, privacy, security, operations management, and interoperability is necessary.
- iii. Performance, scale, support, and stability are critical.
- iv. Consortium Blockchains, limited to consortium members for contract execution, are ideal.

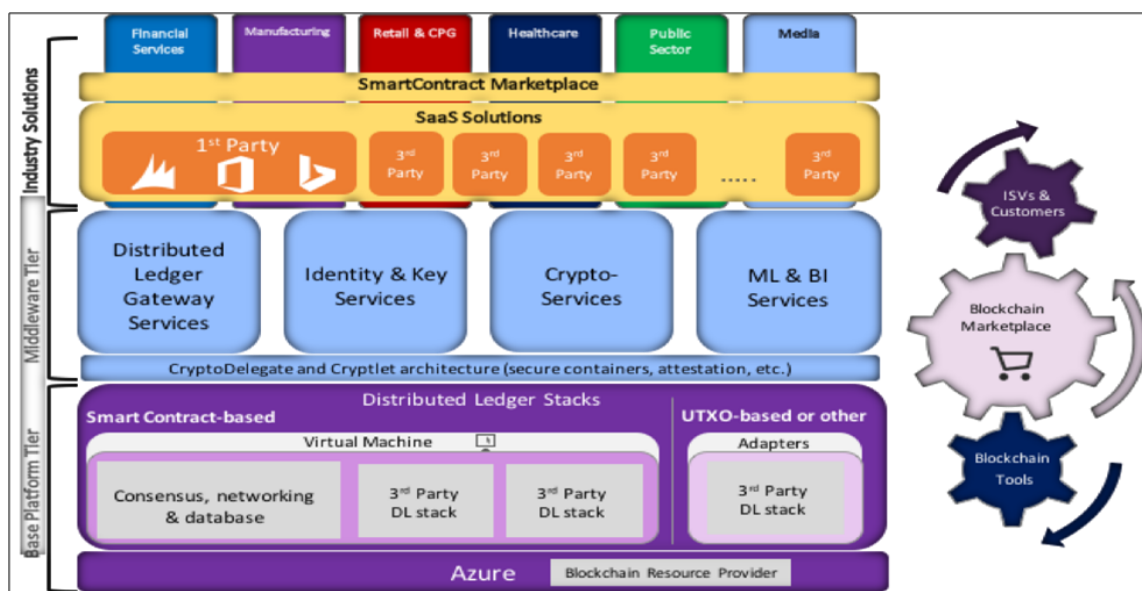
Blockchain middleware and crypt lets are two essential components of Project Bletchley.

Blockchain technology will provide crucial basic features such as:

- i. Identity and Certificate Services utilize Azure Active Directory and Key Vault for authentication, authorization, key issuance, storage access, and lifecycle management capabilities.
- ii. Encryption Services enable partial payload or field-level encryption for Blockchain transactions, ensuring data visibility only to intended counterparties.
- iii. Blockchain Gateway Services facilitate communication between interrelated distributed ledgers using inter-ledger-like services.
- iv. Data Services provide off-chain data, referenced by public keys on distributed file systems (IPFS, Storj), and analytical services like machine learning and reporting for stakeholders and regulators.
- v. Management and operations software enables organizations to provision and administer distributed ledgers.

Figure 4

Microsoft Blockchain Architecture



2.12.2 The Transaction execution Process in Blockchain

According to Kent Weare (2016), a blockchain is a distributed database that connects subsequent blocks of transactions over an internet of communicating nodes. He continues with his explanation in as follows:

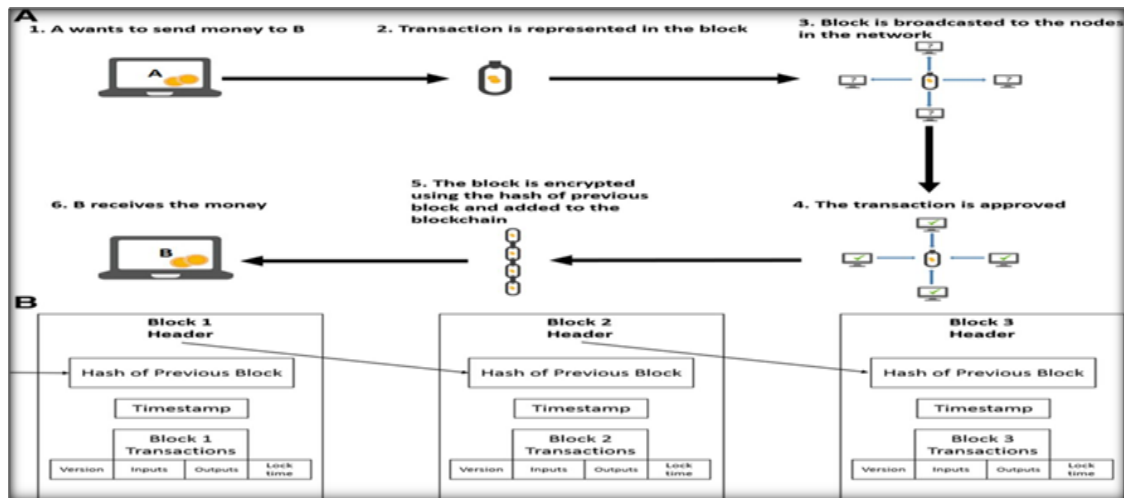
- i. Anyone who aims to transact any property through a public or private network must have internet access. The ability to do so is provided through a program that bridges the consumer and the digital ledger. The computer's application, often known as a wallet, might be deployed on a gadget or accessible through an internet browser. The Blockchain wallet might be utilized for sending or receiving digital currencies, according to how it is configured. Some wallets enable direct transactions without needing a third-party intermediary, but others are managed by outside organizations that hold users' digital currency on behalf of the user.
- ii. Users frequently must set up Blockchain applications on their mobile devices to get involved in transaction validation through consensus. They can use this program to add entries to the ledger, keep an entire copy of the ledger, and maintain it coordinated with other copies. Public Blockchains allow anyone to install the software and have a copy of the entire ledger, enabling direct transactions within the network without third-party restrictions. On the other hand, permission Blockchains are controlled by a centralized authority that decides who can access and participate in the consensus. The records in a Blockchain, often called blocks, are securely linked together using cryptographic techniques, ensuring they cannot be tampered with. Unlike traditional electronic records, once activity on the Blockchain has been documented and timestamped, it's permanent and cannot be altered, edited, or erased.

- iii. A Blockchain transaction contains essential information, such as the details of the transfer, the involved parties, and metadata. It also includes a unique cryptographic hash and a digital fingerprint to verify the transaction's integrity. If someone tries to change the transaction content, the resulting Hash will no longer match the original, and the Blockchain software will detect the discrepancy.
- iv. All parties involved must agree on adding a new transaction record to the Blockchain. Other nodes in the network only verify the parties' capacity to participate in the transaction. The transaction gets finished and posted to the Blockchain if everyone involved consents to it and the nodes validate their capacity.
- v. Every computer in the network constantly and mathematically verifies that its representation of the Blockchain resembles every other copy on the internet. The version operating on a large percentage of personal computers is regarded as legitimate. Therefore, attempting to hack the records would require taking control of over half of the network's computers, which is virtually impossible for public Blockchains like Ethereum that run on thousands of computers. To destroy the ledger, one must delete every copy of it worldwide. Process.

Figure 5 illustrates the step-by-step execution of a transaction within Blockchain A, denoted by steps 1 to 6. Concurrently, the blocks forming the entire Blockchain during the transaction process are represented by Blockchain B.

Figure 5

Blockchain Transaction



Compared to typical transaction processing systems, the strategy described in figure 5 above has significant practical advantages. When an exchange is initiated in the traditional system, a processing company verifies the available money, and the funds are authorized and given to the merchant after several days. Blockchain's fundamental goal as an electronic record is to completely remove these middlemen by generating digital confidence, resulting in enhanced handling of transactions. Once digitally certified, the infrastructure in a Blockchain ecosystem validates interactions, secures past transactions, and facilitates direct transfer of assets between parties.

Blockchains that operate are protocols for encryption that enable a distributed system of computers to keep a common record of transactions without requiring total trust between nodes. Each Blockchain database is made up of a time-stamped sequence of transactions that are validated using the protocol's stated consensus process. This process ensures that the data recorded on the Blockchain is trustworthy as long as the majority of the network's participants validate the resulting blocks or chains in accordance with the regulations of governance. As a result, payment data is continuously reproduced across

the entire system, protecting the Blockchain's integrity and trustworthiness (Newman, 2017).

Because Blockchain uses distributed consensus methods, all nodes in the distributed system have an exact duplicate of the data recorded within the Blockchain. This guarantees that all information stored on Blockchain leaves an auditable record with a verifiable electronic signature. The ledger's data develop widespread and permanent, forming a dependable "transaction cloud," making losing information virtually unattainable and every attempt at manipulation exceedingly costly for participants. As a result, this method removes the hazards of only one point of failure and disparities in information fragmentation among participants. Cryptography plays a pivotal role in enhancing data security by employing diverse techniques. Individuals can cater to their privacy and transparency requirements by utilizing distinct consensus mechanisms outlined by the protocol, along with public and private key pairs. Within a Blockchain framework, data protection operates at the granularity of individual data elements rather than on a collective level, and entry to data is governed by appropriate permissions established within the protocol (Hanson & Staples, 2017).

2.12.3 The Technical Capabilities of Blockchain Technology

The distributed ledger serves as an independent digital money payment mechanism that uses the Blockchain. It doesn't require oversight from groups or governments. The network continuously grows with an increasing number of transfers and users. Furthermore, frequent conversions between digital and traditional currencies, like EUR and USD, occur in currency exchange markets. As a result, Blockchain has attracted significant attention from diverse communities and currently stands as the most successful digital currency utilizing this technology (Martinelli, 2015).

According to Burgess (2015), Blockchain technology is fast gaining popularity considering its numerous potential applications in various domains. Three key characteristics can be derived based on Blockchain's technical capabilities:

- i. Firstly, efficiency can be improved across multiple areas, including administration, verification, approval, transactions and settlements. Blockchain allows for the automated handling and disbursement of routine tasks, reducing time, errors, expenses, and waste while enhancing resource distribution.
- ii. Secondly, Blockchain offers greater transparency through its ability to provide complete records and maintain an immutable distributed public ledger. Many firms operate in secrecy due to a lack of knowledge, norms, teamwork, administration, and confidence in authority. Blockchain, on the other hand, improves visibility by exposing faults, theft, and misinformation. It disseminates knowledge about policies and procedures, enhancing transparency and allowing for more precise assessment and monitoring.
- iii. Finally, by distributing authority and distributing power, Blockchain promotes freedom. These self-government frameworks create trust among all parties and facilitate successful agreement, resulting in a more acceptable and transparent performance. Despite technological hurdles and limits, the integrative examination of Blockchain applications provides a fascinating and expanding topic for additional research.

2.12.4 Blockchain and the Land Registry

The numerous advantages offered by Blockchain have positioned it as the future of Land registries. One key benefit is its potential to significantly diminish property fraud, which has become a pressing issue for Land Registry. By assigning a unique code to each

property and linking it to an owner held smart key, fraud prevention is strengthened. Besides, adopting Blockchain for property transactions could lead to more efficient property management, enabling real time information review and reducing the need for ongoing management time (Donegan, 2016).

Initially created for virtual currency accounting, Blockchain employs Distributed Ledger Technology (DLT). Presently, it mainly verifies transactions within digital currencies, but it also allows the digitization of codes and the insertion of various documents into the Blockchain, generating immutable records. This technology enables the creation of permanent records that remain unchangeable. Fundamentally, record authenticity can be confirmed by the entire community through the Blockchain, eliminating the need for a single centralized authority (Wright, 2015).

Blockchain stands out as a significant technological innovation, distinct from others due to its lack of regulation by any central authority. Users themselves control and validate transactions, enabling direct payments between individuals for goods and services, eliminating the need for intermediaries. Once a transaction is completed, it is publicly recorded in blocks, eventually forming the Blockchain. Other users within the network verify and relay the transaction. Mining, a process where new blocks are appended to the Blockchain approximately every 10 minutes, supports this system (Kshetri, 2017).

When a new computer enters the system, it obtains a copy of the full Blockchain database, which is disseminated and distributed between all nodes in the network. This copy contains records of every executed transaction, serving as indisputable evidence. Consequently, the Blockchain can provide historical information, such as the value associated with a specific address in the past. Access to the entire Blockchain is possible by examining the information stored within the blocks (Friis, 2017).

2.12.5 The Integrity of Blockchain Transactions

The digital ledger technology has been developed to generate permanent operations, meaning that they cannot be removed or altered once documented. The security of adding new blocks to the Blockchain is ensured through cryptographic techniques, which make the system tamper proof. This technology allows data to be stored in blocks, and only intended users with the appropriate access rights can read and view the data without being able to modify it, thus ensuring data integrity. Each participant in the Blockchain network is assigned a unique pair of secret private and public keys, which serve as their visible identifier. These keys are linked cryptographically, allowing identification in only one direction using the private key. To access another participant's identity and pertinent information on the Blockchain, a private encryption key equivalent to their publicly available key has to be present (Kshetri, 2017).

2.13 The Features of a Blockchain-Based Model

Blockchain, as a developing disruptive innovation, has the potential to alter several sectors by improving the framework of legislation, protection, transparency, and efficacy in how the sector could function. Among the fundamental features of Blockchain, according to O'Dair and Beaven (2017), are:

- i. **Decentralized systems:** The concept of decentralized systems simplifies asset storage by enabling network access over the Internet. Assets can encompass tokens, contracts, chain-of-evidence, or property registry documents. With decentralized technology, asset owners retain direct control through their private key, directly linked to the asset. They can transfer the asset to anyone whenever desired. Blockchain technology is perceived as a powerful tool for decentralizing the web and can significantly change industries like banking and finance. Embracing this technology gives individuals direct control over their assets,

eliminating the need for costly transaction intermediaries. As banks prepare for this transformative shift, they are developing services to adapt to the forthcoming changes. The underlying powerful idea is that trust is placed not in a single entity like a government, bank, or multinational company but in the network itself. Assets such as contracts, evidence documents, identity information, and financial assets can all be stored within this decentralized system. There are numerous causes for independence.

- i. Empowered users: Decentralized systems grant users complete control over their information and transactions.
- ii. Fault tolerance: Decentralized systems are less prone to accidental failures because they rely on numerous separate components.
- iii. Durability and attack resistance: Blockchain's lack of central control enhances its ability to withstand malicious attacks, making it more expensive and challenging to attack or manipulate.
- iv. Free from scams: Users in decentralized systems find engaging in activities that harm others for personal gain challenging.
- v. Removing third-party risks: Decentralized technology enables direct user exchanges, eliminating the need for intermediaries and associated risks.
- vi. Increased transaction rate: Despite conventional financial interactions, which frequently require longer clearance intervals, blockchain-based transactions are able to be completed in moments and at any point.
- vii. Lower transaction costs: By eliminating third-party intermediaries and overhead costs, Blockchains have the potential to reduce transaction fees substantially.

- viii. **Transparency:** Because modifications are accessible to all events, public Blockchains guarantee openness, and all activities are permanent, which means they are unable to be changed or erased.
- ix. The decentralized nature of Blockchain guarantees comprehensive, coherent, timely, precise, and accessible data for a broad audience.

The Internet revolutionized our lives through decentralized access and information flow. With Blockchain technology, a paradigm shift is underway, potentially transforming industries and enhancing democracy, security, transparency, and efficiency. The wide-ranging implications of Blockchain technology on society and businesses are profound.

- ii. **A distributed ledger:** Blockchain functions as a public ledger, containing information about all participants and executed digital transactions. Each transaction is recorded in a block, which becomes part of the Blockchain once completed. Blocks are added sequentially, each containing the Hash of the previous one. Every user in the network has an identical copy of the ledger and can validate transactions with encrypted transactions added to their respective copies. The ledger is updated across the network in minutes or seconds when changes occur. Through cryptographic keys and signatures, participants maintain the security and accuracy of assets. This results in a highly efficient and secure transaction processing method, an immutable online ledger for recording transactions.
- iii. **Tamper-free:** Don and Alex Tapscott (2016) defined Blockchain as a secure digital record that has the ability of documenting monetary transactions and almost everything of value, not only financial dealings. The previous features elucidate how Blockchain offers a tamper-free environment for network participants. The

network automatically checks and updates itself regularly, creating a robust, self-reviewing system. This ensures the following:

- i. Data is embedded within the network, ensuring public accessibility and promoting transparency.
 - ii. Tampering with block data would necessitate substantial computing power to surpass the entire network, making corruption challenging.
 - iii. Each block is linked and verified, eliminating any single point of failure.
 - iv. The Blockchain is comprised of networks of purported computing nodes.
- iv. **Minting:** There are different techniques for generating currency, among them the most prevalent of which is mining, but additional techniques are also being developed. Proof of work is a protocol used to prove engagement in significant computational work, solve puzzles for mining, and provide proof of work by encrypting data in the block creation process. Miners are financially rewarded for their efforts and compete to add the latest block on top of the Blockchain. Other nodes verify the added block when additional blocks are added on top. Proof of stake is another method used in mining, where the creator of a new block is chosen based on their wealth, eliminating the need for absolute power and hardware.

This lowers the overall cost of the mining process and incentivizes stakeholders to be responsible for the currency's system. In Blockchain, the nodes form a robust network where each node acts as an administrator. Every node has the motivation to participate in the network. They compete with each other to solve computational puzzles and have the opportunity to win digital currencies. These

nodes are known as miners because they search for current transactions that safeguard the authenticity of most recent transactions (Kelly & Williams, 2016).

2.14 The Blockchain Transaction authenticity using the Hash Function Algorithm.

Many firms place a premium on data and record truthfulness, particularly when working with confidential data, resources, or agreements. A hash is generated for each file to protect this data, serving as a unique fingerprint through an algorithm that converts data into a fixed-length output. Each block in the Blockchain consists of a header and a transaction part. The header includes the previous block's Hash, referencing the previous link in the chain, and the Hash of the current transaction, connecting it to the next block when added. The distributed and updated nature of the Blockchain allows other locations to maintain it if any copy is compromised, making it immutable. By comparing the Hash of the data, the authenticity can be verified, ensuring an independently verifiable system (Kelly & Williams, 2016).

2.15 Potential Industry Application Areas of Blockchain

Blockchain can be likened to a collection of technologies, much like bricks used in construction. Just as different bricks can be combined in various ways to create different structures, Blockchain technologies offer diverse possibilities for application. As bricks have numerous use cases, so does Blockchain technology (Pinteres, 2017).

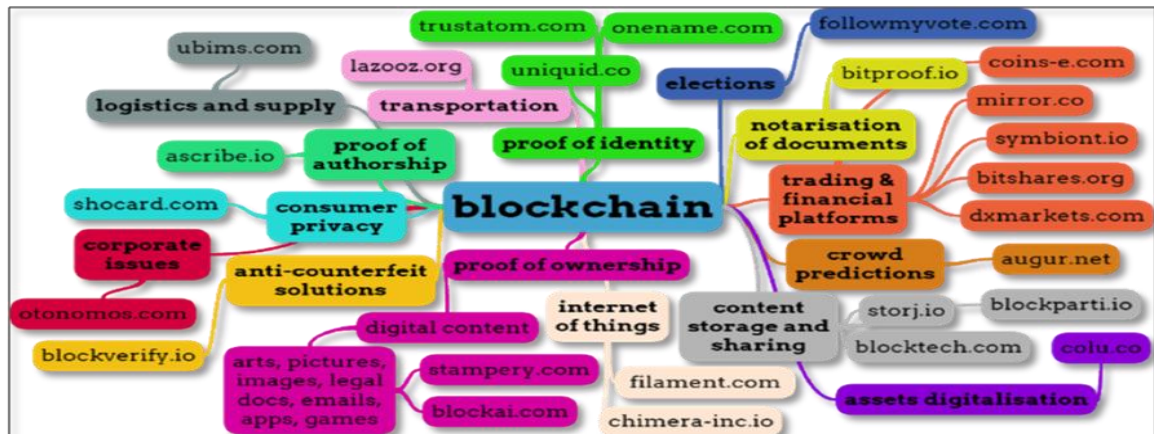
A report by Moody (2017) titled "Credit Strategy - Blockchain Technology: Robust, Cost-effective Applications Key to Unlocking Blockchain's Potential Credit Benefits" identified approximately 120 ongoing Blockchain use cases being evaluated by various companies. These applications span various industries, including finance, capital markets, healthcare, energy, government taxation, investments, partnerships, internal

projects, and industry collaborations. For instance, Blockchain can be implemented in capital markets to streamline ledger reconciliation among members.

Moody's selected the top 25 use cases from a pool of 120. Cloud storage, intelligent agreements, social media, anti-counterfeiting, oversight, electronic verification, distribution chain executives, art and property verification, predictive sectors, and Internet of Things (IoT) applications are some significant Blockchain technology scenarios.

Figure 6

Potential Industry Application areas of Blockchain



The core purpose of Blockchain is to facilitate bookkeeping through distributed ledger technology. Its main advantages include removing the need for intermediaries, user-friendliness, secure transaction verification, enhanced security, cost-effectiveness, transparency, and decentralization. The most robust aspect of Blockchain lies in its immutability, ensuring that data, once recorded, cannot be modified or tampered with. According to financial professionals, blockchain can greatly improve efficiency and safety in various conventional procedures and structured enterprises (Pinteres, 2017).

According to Wood (2016), several use cases are gaining momentum with adopting blockchain technology.

These include digitizing paperwork and binding agreements, secure communication of information tracking in the Internet of Things programs, determining confidence and security in custody-based and deposit user accounts, developing dispersed cloud storage facilities, expanding medical treatment employing the treatment of patients and management of documents. Other use cases include making it possible for voter identification and digital polling, verification of the ownership of assets and preventing counterfeiting of digital records. Blockchain technology enables internet users to produce value and validate digital data, enabling a wide range of commercial applications in these fields.

- i. Land title registration is a critical area where property titles are vulnerable to fraud and require significant administrative labor and expenses. Several countries are now exploring blockchain-based solutions for land registries. Honduras pioneered a blockchain initiative for this purpose in 2015, though the project's current progress remains to be determined.
- ii. Intelligent agreements are made achievable by distributed ledgers, which enable the construction of agreements that execute dynamically when certain criteria are met. The digital currency Ethereum, a blockchain that uses open-source technology, was created to support this capability. Smart contracts, in their current state of expansion, can handle very simple tasks. For instance, in the banking industry, a derivative agreement can be immediately compensated when a particular standard is fulfilled due to the automation afforded by the blockchain system and Bitcoin.
- iii. Governance can be significantly enhanced through distributed database technology, which ensures complete transparency and public accessibility of results in elections or polls. By employing Ethereum-based smart contracts, the

process can be automated and streamlined. An application like Boardroom facilitates decision-making within organizations on the Blockchain, resulting in fully transparent and verifiable company governance, especially when dealing with digital assets, equity, or information.

- iv. Supply chain auditing is becoming more crucial as consumers seek assurance regarding the authenticity of ethical claims made by companies about their products. Distributed ledgers offer a convenient solution to verify the genuine origins of the items we purchase. By utilizing blockchain-based time stamping, information like the date and location of ethical diamonds can be recorded with a unique product number, providing transparency and ensuring the credibility of the product's backstory.
- v. Decentralizing file storage on the Internet offers significant advantages by distributing data across the network and protecting files from potential hacking or loss. The Inter Planetary File System (IPFS) provides a clear vision of how a distributed web could function, similar to how BitTorrent moves data around the Internet. IPFS eliminates the dependence on centralized client-server relationships, which is common in the current web structure. An internet composed of fully decentralized websites has the potential to significantly enhance file transfer and streaming speeds, alleviating the strain on the overloaded content-delivery systems of the present web. This improvement is convenient and necessary to optimize online file sharing and access.
- vi. Prediction markets leverage the collective wisdom of crowds to make predictions about event probabilities, which are highly accurate. By aggregating opinions, these markets can mitigate unexamined biases that may distort individual judgment. Blockchain technology plays a significant role in these prediction

markets, ensuring transparency and trust in the outcomes. Augur, a prediction market application still in Beta, allows users to make share offerings on real-world events. Participants can earn money by investing in correct predictions, with higher payouts based on the number of shares purchased for the right outcome. It only requires a small financial commitment, making it accessible to anyone who wants to create a market based on a predicted outcome and collect half of the generated transaction fees. This "wisdom of the crowd" technology is expected to find additional applications in the future beyond its current use in prediction markets.

- vii. Intellectual property protection has become a significant concern due to the ease with which digital information can be copied and distributed on the Internet. While users enjoy access to a vast amount of free content, copyright holders often need help maintaining control over their creative works and facing financial losses. However, smart contracts offer a promising solution to this problem. They can protect intellectual property and systematize the internet-based purchase of one's original works, removing the possibility of illicit file duplication and resale.
- viii. The Internet of Things (IoT) refers to the network-controlled administration of certain electronic gadgets, such as measuring the environmental conditions of the air in a place of storage. Intelligent agreements allow for the centralization of remote system administration in the Internet of Things. This automation is accomplished through programs, sensors, and social media, allowing seamless data interchange among objects and procedures. As a consequence, the effectiveness of the system improves, and cost control improves.

- ix. Identity management on the Internet is a critical aspect that requires improvement. Verifying one's identity is essential for secure online financial transactions. However, existing remedies for security risks in web commerce could be better. Distributed ledgers offer enhanced solutions for identity verification and the potential to digitize personal documents. Secure identity is equally vital for online interactions, especially in the sharing economy. Maintaining a good reputation becomes crucial for conducting successful transactions online.
- x. Anti-money laundering (AML) and know-your-customer (KYC) practices hold great promise for integration with blockchain technology. Presently, financial institutions undergo time-consuming and multi-step processes for each new customer. By leveraging Blockchain, KYC costs could be minimized through cross-institution client verification while enhancing monitoring and analysis effectiveness.
- xi. Currently, individuals provide their data to social media platforms like Facebook in exchange for using their services for free. However, in the future, users may gain control over the data generated from online activities and have the opportunity to sell it. With the ease of distributing data in small fractions, Bitcoin will likely be the preferred currency for such transactions.
- xii. Blockchain technology offers significant advantages for stock trading, particularly in terms of efficiency in share settlement. By using Blockchain for peer-to-peer transactions, trade confirmations can occur almost instantly, unlike the current three-day clearance process. This could eliminate the need for stock trading intermediaries such as clearing houses, auditors, and custodians.

xiii. E-voting has faced challenges related to transparency and security when conducting online electoral polling. However, blockchain technology offers a solution that allows voters to verify the successful transfer of their votes while maintaining anonymity. In 2014, the Liberal Alliance, a Danish political party, was the first to use blockchain-based voting. Estonia and Norway also tested electronic voting platforms for local and national elections, but fewer transparent procedures may have aided their success. Despite these initial obstacles, blockchain technologies have the potential to raise accountability to the highest possible level as organizations and processes grow increasingly transparent.

2.16 Secure Execution of Land Transactions in Blockchain

In terms of land database, Blockchain technology creates a publicly accessible distributed database that keeps track of each property's payment records, allowing confirmation land ownership at any given point in time. This distributed database is copied on various computers worldwide, making it public (The Economist, 2015).

Previously Blockchain technology had a doubtful reputation and only a few individuals would trust the technology. However, in the recent past, the technology has outlived the notion and it is now gaining recognition as a viable solution for recordkeeping challenges that require a reliable public ledger. The technology is now being used in financial transaction records, land registries, civil registration, and various public registration systems. Integrating "smart contracts," which automate contractual transaction rules, further enhances its potential in these areas (Szabo, 2005).

According to Almgren and Stengard (2015) the U.S state of Vermont has enacted legislation to investigate the prospect of establishing an expectation of authenticity for computerized facts and documents that use the Blockchain system.

2.16.1 Recording a Land Transaction in Blockchain

Blockchain technology is based on a digital chain of hashes generated by digital signatures. These digital signatures employ asymmetric cryptography with private and public keys to prove the legitimacy of an electronic communication or document. A valid electronic signature ensures that the message was transmitted by an identifiable sender (authenticity), that the sender is prohibited from transmitting it (non-repudiation), and that the content remained unmodified during transmission (integrity). Digital signatures are widely used in software transportation, money transactions, and other cases where recognizing forgery or manipulation is critical (Wood, 2016).

Each participant involved in a blockchain transaction digitally signs an encrypted version of the preceding transaction, along with the publicly available key of the next recipient, using their private key. These signed components are then appended to the end of the hash chains. The recipient, such as a payee, can verify the signature; a process often accomplished using the Digital Signature Algorithm with an Elliptic Curve, specifically the secp256k1 curve. The secp256k1 private keys consist of 256 bits of randomly generated data, and the corresponding public key is derived through computational manipulation. This procedure eliminates the requirement for a single governing body (Certificate Authority) to develop and store public keys, which is a common component of the infrastructure for public key encryption (Nakamoto, 2009).

2.16.2 Validating a Land Transaction in Blockchain

To prevent asset duplication, transactions are dispersed over an internet of nodes to establish agreement on their order and authenticity. Nodes within the system organize transactions into blocks, which are subsequently encrypted and assigned timestamps. To address this, a timestamp server is introduced as a remedy. It produces a hash of various objects and openly disseminates it to demonstrate the existence of certain data at a

specific time. Subsequent blocks in the sequence include the previous block's hash, forming a chain where each following timestamp strengthens the preceding ones. This process safeguards the transactional consistency and sequential arrangement of transactions within the Blockchain (Nakamoto, 2009).

The Hash cash verification of functionality uses three parameters to accomplish its objectives: an authorization rope, an acknowledgment (a generated or pseudo-random quantity used in verification for avoiding replay attempts), and a countdown timer. The header of the block information structure encodes the service string. It contains critical information such as the version number subject content, the Hash of the previous block, the essential password of the Merkle structure of each transaction in the interference with the current time and the evidence of work complexity factor. Each device in the distributed system searches for a specific amount that when encrypted, starts with a sufficient number of zero bits. This strategy safeguards the Blockchain's security and dependability by making the production of new activities technologically and labor intensive (Wiki, 2022).

2.16.3 Updating a Public Ledger in Blockchain

When a computer discovers the evidence of a place of employment, it communicates the freshly formed block to all of the network's nodes. All transactions in the block must be valid to allow the transaction to be acknowledged. Nodes demonstrate their consent to the use of the block by constructing the following transaction in the Blockchain, referencing the hash of the acknowledged block. The network's nodes function on an agreement structure, which means they collaborate with minimum coordination. They are not required to be identified individually and can freely enter and exit the network. This autonomous and self-governing nature protects the blockchain network's integrity and stability (Nakamoto, 2009).

Blockchain supporters say that confidence is strengthened among everybody involved since there is no opportunity for abuse by a controlling node, as there is with systems that rely on just one reliable outsider that may be corrupted or act maliciously. This decentralized system has integrity as long the truthful nodes together control a greater amount of CPU power than any collection of working together attacker nodes because every attempt to change a previous block requires the intruder to rewrite the evidence of work for the current block and all the following ones while exceeding the combined work performed by the truthful node locations. The distributed ledger network's dependence on the agreement and computing resources makes it resistant to prospective attacks (Wild, 2015).

2.17 Factom Blockchain Model for Honduran Land Registry System

An article from The Economist (2015) highlighted the challenges faced by Hondurans due to weaknesses in the country's Land registration system. The story of Mariana Catalina Izaguirre exemplifies the difficulties many people encounter in Tegucigalpa, the capital of Honduras. Despite living in her house for three decades and possessing an official title to the Land, another person was registered as the owner in the country's Property Institute records. This situation led to the wrongful eviction of Ms. Izaguirre after someone else convinced a judge to issue an eviction order. Eventually, after a lengthy legal process, Ms. Izaguirre's house was demolished. Unfortunately, Honduras is not alone in this circumstance; numerous towns and countries worldwide suffer from poorly managed or compromised land records and land information systems for management. Individuals experience severe insecurity and injustice due to a lack of protected ownership rights. Additionally, it hampers the ability to use houses or Land as collateral for investment and job creation (The Economist, 2015). The adoption of

Blockchain technology in the Land registry system aims to address these issues and provide a more secure and reliable recordkeeping system for land ownership.

Honduras enacted new property legislation in 2004 in order to establish an extensive regulatory structure and modernize the handling of land. This comprised a number of governmental changes, such as the creation of a new national system for property administration known as SINAP (Sistema Nacional de Administracion de la Propriedad), and the development of the technological infrastructure sections for the nation's unified registry system, SURE (Government of Honduras, 2004).

Despite these attempts, the mechanisms have proven vulnerable to manipulation and theft. According to one source, the nation's database had been compromised, encouraging officials to take advantage of the structure and purchase beachfront homes. This raised questions regarding the dependability of the Honduran Registry of Lands data and the nation's continuous validity of land titles. The flaws in the Land Registry process emphasized the requirement for a safer and open alternative, prompting the most recent investigation and adoption of Blockchain technology (Chavez-Dreyfuss, 2015). Honduras adopted Blockchain technology in the management of their Land registry system with the aim of addressing these issues by creating a more transparent, efficient, and secure process for recording Land ownership and Land transactions (USAID, 2010). The Factom Blockchain model developed and deployed in the Hondurian Lands Registry system (Snow, 2014). The Factom model relied on a unique data structure specific to Factom, which included multiple layers for organizing and verifying the recorded information such as:

- i. Chains: Groups of related entries within a specific application.
- ii. Entry Block Layer: Contains references to entries.
- iii. Entries: The actual raw data of an application or a hash of its private data.

- iv. Directory Layer: Manages the Merkle Foundations of entering components. Every directories block has a combination of microscopic networks of entrance ChainIDs.

2.17.1 Transaction Recording in the Factom Model

According to Snow (2015), the following were the procedures for keeping track of Land transactions in the Factom Blockchain Model:

- i. At the start of each processing period, all nodes clear their process lists, which are lists of chains created by the federated nodes during their designated time. Creating a list is critical in propagating node-specific judgments across the entire network.
- ii. To begin an exchange, the user makes an entrance payment with a public key connected to entry credits referred to as Factoids - the digital currency that Factom employs. Users must have publicly accessible keys to sign Factoid interactions, entry obligations, and chain responsibilities. If users want to vote in the system, they must first construct an identity and then connect the public keys containing admission credits with that person's identity. To put anchors into the Blockchain, Factom servers must have a username and password with verified keys.
- iii. Each node takes payment according to the publicly available key used for the data entry. The particular node communicates payment approval, and the individual receives it and publishes the application's data.
- iv. Factom groups the data entries across chains and connects them with a ChainID. ChainIDs are generated from link designations, which can be obtained from spoken language and have meaningful value in connection with documentation.

This allows for speedier retrieval than if every record were consolidated into just one record. Authentic transactions and associated ChainIDs are kept apart. At the protocol level, chains with various ChainIDs are not dependent on one another, which means that a program capturing a chain can disregard any additional chains. Still, Factom cannot verify trustworthiness to the same stage as the Blockchain because the most current record does not contain all previous records in a digital signature but simply the previous transactions associated with that particular chain in concern.

- v. The component broadcasts a transaction confirmation, which includes the process line index of the acceptance, the encryption key of the information entering (which is associated to the payment), and the sequence hash of the node's current task list thus far. The remaining nodes shift their focus to the node's operations list, validate its contents, and shift their focus to the appropriate entry node corresponding to the ChainID.
- vi. Each minute, every node is used to verify the procedure list elevation reveal the predictable personal key, and disclose the serial number hash corresponding to the process block (which matches the final item on the procedure list). The minute's directory block is produced by collecting all of the entry entries specified by each node. As a result, each node has all of the entry blocks of data, directory blocks of data, and articles.
- vii. The randomized confidential keys are gathered and concatenated to form a seed file, which is subsequently utilized to distribute the ChainIDs throughout the nodes in the network in the next round.

2.17.2 Validating transactions in the Factom Model

Factom Model required users and applications on the client side to validate transactions. Real estate and land transactions involve intricate and diverse rules across different countries. Local jurisdictions may impose specific requirements based on the buyer's status, such as being a foreigner, farmer, or part-time resident. Properties can be categorized based on location, price, or architecture, each with rules for validating smart contracts. In this context, relying solely on a cryptographic signature may not be sufficient to ensure the authenticity of ownership transfers. Instead of immediately authenticating the funds being transferred to one another, Factom model provided a mechanism for recording the entire transaction procedure (Snow, 2014).

The Factom solution aligns with the concept of reliability as defined by ISO 15,489, focusing specifically on ensuring documentary reliability. This dependability is related to the legitimacy of the user and the authentication of the transaction by the Land registration application. Two sorts of nodes are involved in the process: Factom nodes and Auditing nodes. Factom nodes receive records, group them into categories such as blocks, and organize them. It becomes irreversible by anchoring the entry order into the Blockchain after 10 minutes. Auditing records is an independent process that can be carried out in two ways within Factom. A "thin client" chooses an Audit node it believes in the "with trust" strategy. The Audit node subsequently provide individually cryptographically verified submissions, certifying that the information provided has passed all essential tests and meets the requisite requirements. Also, the "trustless auditing" approach works in the same way that the Blockchain does. Factom's mechanisms ensure documentary reliability throughout the transaction process, offering a robust and secure solution for recording and validating data downstream of users and Land registration applications (Snow 2015).

2.17.3 Updating a Public Ledger and Authenticating Transactions in the Factom Model

Factom uses a consensus approach to enable efficient and orderly recording of Submissions without central administration or the necessity for participants' public identities. To guarantee effective implementation, the protocol includes cryptographic safeguards. A user voting mechanism is used to eliminate undesirable individuals from power when their inappropriate conduct cannot be identified in real-time by the Factom protocol. Factom works well as long as most participants are honest and follow the protocol. The system's overall effectiveness may suffer due to growing duplicity among system actors. Nevertheless, Factom ensures that non-compliant actors will eventually be replaced with new ones. The records are saved in the Federated and Audit nodes, and services need this information to make informed decisions about adding new entries. Because they have the knowledge required as a consequence of becoming fully functioning nodes, these nodes can deliver data as a service. Additionally, partial nodes only share specific data relevant to their application within the Factom data set. Factom ensures the integrity and reliability of recorded entries by using this consensus approach and cryptographic measures (Snow, 2015). The following is a description on how Factom updated the distributed public record:

- i. Create the root directory block header, which includes the Merkle Core and the prior directories block's sequential serial.
- ii. The directory block is further constructed by including the ChainIDs of all the chains that have had entries in the last 10 minutes. These ChainIDs are sorted in ascending order, ensuring a systematic arrangement of the chains within the directory block.

- iii. The serial hash of the directory block is calculated, and most Federated nodes provide their signatures for verification. These signatures are then appended to the directory block, ensuring that the directory block remains secure and tamper-proof.
- iv. A Merkle Root is formed by combining the directory block and its signatures into a single cryptographic hash value, ensuring data integrity and efficiency in the verification process.
- v. Node #1, assigned for the previous minute, adds the Merkle Root to the Blockchain using one of their addresses.
- vi. Reset or empty all the process lists.

2.17.4 Inadequacies of the Factom Model

As previously mentioned, the Factom solution for Honduras fails to deal with the issue of record integrity. This aspect is still outside of the realm of what Factom officially provides. As a result, despite the company's claims, it's difficult to see how the existing Factom system will alleviate Hondurans' recordkeeping problems. Hondurans must still register their land possession with the Property Institute, an impartial and trusted third party charged with registering land titles in the SURE Network. This procedure occurs outside the technological structure of a solution such as Factom. Furthermore, regardless of recorded applications, inaccurate or unauthorized entries can be generated in the underlying Property registration database before the documents are delivered as Submissions to a Factom server (Snow, 2015). As a result, assuring the reliability of records remains a significant restriction of the Factom Blockchain model.

2.17.5 Authenticity of the Factom Model

The core promise of the Factom solution and Blockchain technology lies in maintaining authenticity, particularly the integrity of records. However, the effectiveness of this

technology in preserving record authenticity is dependent on its vulnerability to faults and security breaches. Although a thorough evaluation of information assurance and security risks is not possible without in-depth familiarity with the exact system architecture of the Honduran Land registry and would necessitate separate investigation (Eskandari, 2015).

2.17.6 Information Security Concerns in the Factom Model

2.17.6.1 The Man-in-the-Middle Attack

The Factom solution for the Honduran Land registry faces potential vulnerability to Man-in-the-Middle Attacks (MitMA) at two critical junctures. The primary weakness occurs when a fresh Land registration is entered into the platform, particularly if the data being transferred is not encrypted before reaching the Factom answer. At this stage, hackers can silently monitor and modify exchanges between parties, jeopardizing the information's integrity and authenticity. The Factom approach roots the Merkle Root that holds the certified Factom catalog block in the Blockchain, an additional source of weakness. As miners do not conduct audits to verify transaction validity, attackers can insert invalid transactions designed to appear genuine, further jeopardizing the system's integrity. The probability of a MitMA occurring in this context is particularly high, given the prevalent system hacking in Honduras and the lack of encryption during data transmission between systems (Eskandari, 2015).

The unencrypted passage of data amplifies the risk of unauthorized access and manipulation by malicious actors. Addressing these vulnerabilities is crucial to ensuring the security and reliability of the proposed Honduran Land registry solution. Implementing robust encryption measures during data transmission and incorporating additional security protocols can enhance protection against MitMA and safeguard the authenticity of Land registration records. Furthermore, establishing stringent auditing

mechanisms for Blockchain transactions can help mitigate the risk of fraudulent entries and enhance the overall trustworthiness of the system (Luvanda, 2014).

2.17.6.2 SYN Flood attacks

As described by Wiki (2015), an SYN Flood attack is a type of Denial-of-Service (DoS) attack aiming to incapacitate a target system. The attacker bombards the system with numerous rapid SYN requests to deplete the server resources and render it unresponsive to legitimate traffic. When a server wishes to initiate communication with another server, it sends an SYN (synchronize) message to establish a connection. Subsequently, a handshake process ensues, wherein both servers acknowledge each other. During an SYN Flood attack, the targeted server cannot complete the handshake process due to the continuous influx of new SYN requests. As a result, the server becomes overwhelmed, as its resources are flooded with requests, leading to unresponsiveness. While Blockchain technology has implemented various measures to prevent DoS attacks, including SYN Flood attacks, it remains challenging to eliminate such threats, especially in solutions that rely heavily on broadcasting communications through a public network.

The decentralized nature of Blockchain and its reliance on public networks can leave it susceptible to potential DoS attacks. Although protective measures are in place, attackers constantly evolve their techniques, making it challenging to rule out the possibility of SYN Flood attacks or other denial-of-service attacks on Blockchain networks (wiki, 2015). Continuous monitoring, prompt response to suspicious activities, and ongoing security enhancements safeguard Blockchain systems from potential threats.

2.17.6.3 Sybil attack

A Sybil attack involves an adversary overwhelming a Blockchain mesh network by introducing numerous controlled nodes. This manipulation increases the likelihood of genuine nodes mainly connecting to nodes controlled by the attacker. This attack enables the adversary to obstruct or decline the relay of legitimate blocks and transactions, and it may even result in the disconnection of entry registration communications from the network. Moreover, the attacker gains the capability to relay only the blocks they have generated selectively. As the concentration of miners within the network increases, the probability of such an attack becomes more likely (wiki, 2015). Essentially, a Sybil attack undermines the integrity of the Blockchain network by exploiting the network's trust mechanism and its dependence on the quantity of participating nodes in the system.

2.17.6.4 Timing Errors and Attacks

Each block produced in the distributed ledger comprises information about the transaction and a timestamp, which aids in the regulation of block production and establishes the chronological order of transactions, preventing double-spending issues. In a Land registry system context, time stamping is crucial for distinguishing between authentic and inauthentic Land registry records. Nodes calculate timestamps based on the median time of their peers obtained during the connection process. All nodes must have properly functional network time counters to avoid timestamp mistakes. Attackers, on the other hand, may alter an appliance's networking time tracker by linking as many peer to peer nodes and supplying erroneous dates and times. A bigger concentration of miners, similar to the attacks on Sybil, improves the risk of this assault. Ensuring accurate and secure times tamping is essential to maintain the integrity and reliability of the Blockchain system (Culubas, 2011).

2.18 Key management in the Factom Model

Effective key management is critical in cryptographic systems like the Factom model, encompassing key generation, exchange, storage, use, and replacement. This task presents practical challenges as users must ensure secure accessibility, protection against theft, and resilience to loss for potentially millions of keys. Key effectiveness in management is achieved through network regulations, user instruction, administrative communication, and departmental contacts. On the other hand, solving the difficulties of key leadership constitutes an unanswered topic (Eskandari, 2015).

The complexities of managing cryptographic keys create vulnerabilities for the loss or theft of private keys within systems like Factom. For instance, private keys are often stored in files or databases on a node's local storage in these systems, making them susceptible to unauthorized access by any application with access to the user's application folder. Sharing the application folder unintentionally, using peer to peer file sharing, remote backups, or shared network drives, can expose these keys to potential attackers. Additionally, risks such as physical theft, equipment malfunctions caused by natural disasters, power problems, wartime events, or accidental deletion further contribute to the potential loss of keys (Eskandari, 2015). To guarantee the security and reliability of cryptographic systems, users must be proactive in protecting private keys and adopting strong key management practices.

2.19 Digital Record-keeping and Crypto Currency in the Factom Model

According to the approved Factom options for the Honduran Land registry, Factoids, and Original Land entries can only be submitted to the Factom Blockchain platform by using Factoids to acquire entry credits, which are then used to publish entries.

While the current setup works effectively when the Honduran government controls the Factom nodes and Factoids provide as the technique's dedicated currency, the scenario changes when the nodes are managed by another organization that can influence the Factoid currency-to-entry acknowledge the cost of return (Chavez-Dreyfuss, 2015). Aside from the possibility of currency exchange tampering, fluctuations in currency rates of conversion as a result of varying volumes of trading and consumer demand present currency exposure to radiation, which could prove challenging for the Honduran government to keep track of and may jeopardize the approach's financial viability over the long term.

Developing countries like Honduras already possess vulnerable economies, making them susceptible to external economic shocks. Introducing crypto currencies for recordkeeping in such economies raises concerns about exposing them to higher risks of these shocks, requiring further research and careful deliberation. Despite these restrictions and concerns, there is universal agreement that Blockchain-based open-source databases perform exceptionally well at confirming transactions such as cash transfers or commodity sales. By eliminating the need for financial intermediaries, this technology overcomes inefficiencies, allowing individuals to transfer assets to others at low costs instantly and without the typical documentation requirements (Government of Canada, 2015).

Blockchain could be beneficial in developing countries, particularly where a large segment of the general population, including those with less disposable income, owns mobile phones but lacks access to financial services. Money transfer recordkeeping in such cases is typically short-term, whereas Land titles require long-term retention. In invalid transactions, losses are relatively minimal for individual money transfers. However, they could be significant and have broader societal

consequences in the case of Land transfers, considering the public good nature of Land. As a result, solutions based on the Blockchain may be best suited for use cases requiring short retention periods and little evidence (Government of Canada, 2015). Those with greater retention periods and higher evidential standards, on the other hand, could be less appropriate for these kinds of requests.

Progress in implementing Blockchain-based financial transaction processing is advancing quickly, with significant developments occurring in different countries. For instance, the Australian Stock Exchange (ASX) has collaborated with Digital Asset Holdings, a US company, to integrate Blockchain technology for securities trade settlement. Unlike the public Blockchain employed in the Honduran Land registry case, the ASX's system will utilize a private network, allowing only authorized parties to participate. This intimate network approach could indicate a possible direction for future Blockchain-based recordkeeping solutions, addressing some of the mentioned risks. The efforts in Australia and Honduras and research conducted in the US State of Vermont and by the Canadian Government underscore the increasing adoption of Blockchain technology. According to Australian Stock Exchange (2016), this highlights the importance for records professionals to acquaint themselves with this technology to effectively respond to its utilization proposals.

2.19.1 Long-Term Digital Preservation

When confronted with for a long time socio-economic or historical relevance documentation, which includes Honduran land registers, electronic preservation is critical. While some may view it as a legacy concern to be addressed later, there is a growing consensus that digital preservation, similar to IT security, should be considered early in the process. Lessons from past experiences, like the Mortgage

Electronic Registration System (MERS) in the USA, have shown the importance of proactive digital preservation. MERS was designed to replace the manual property filing procedures (Kreiger, 2013). However, it was discovered following the financial crisis that it had been responsible for widespread Land title difficulties.

2.19.2 Preservation and Availability of Original Records

ISO 14,721 (2012a) underlines the importance of having a Trusted Digital Repository with technological, regulations, and organizational capacities for various record-related procedures. The Factom system, on the other hand, needs to include these features, creating worries regarding the retention of original data for comparing Blockchain updates. Factom's, for a long time, entry authentication is based on the retention of an exact duplicate of the initial information that was that is utilized for encrypting and signatures that are digital. Because the Blockchain's digitally signed version cannot be altered or manipulated for generating the initial documentation, keeping originals is critical for re-hashing and verification (Eskandari, 2015).

The particular elements that comprise the Factom Blockchain technological framework must be retained accompanying the original documentation to allow evaluation against Blockchain transactions and maintain that they are genuine if the initial records are not explicitly stated. Even tiny changes to any portion of the underlying electronic document, which include bit corruption or electronic preservation methods, or differences in capturing techniques or how they're implemented utilizing different elements, could make future record identification unachievable. Moreover, managing and preserving these protocols and components, particularly across the extensive distributed network of nodes in different jurisdictions, may be entirely beyond the control of the Honduran government

(Eskandari, 2015). This presents a significant challenge in realizing the proposition of digital preservation in practice.

2.19.3 Digital Signatures

Factom and Blockchain technologies depend on digitally signed documents that combine a hash digest of communications with encryption. Handling encryption necessitates dealing with key management, which for Factom and Blockchain, entails maintaining confidential keys for a prolonged time. Users must also retain the file format specifications to ensure continued access to these keys. Another issue is the increased collision likelihood over time, which occurs when various entries yield an identical message sequence. Although meant to reduce the chance of collisions, digest algorithms are prone to collisions, according to Rosenthal (2005). However, several of the hypotheses assumed in the algorithm design may not hold in electronic preservation situations. For instance, the algorithm typically assumes random input strings of bits, which may not be true for digital preservation scenarios (Rosenthal, 2005).

Blockchain technology seeks to safeguard against theft, but the methods of encryption employed in digital signatures used to generate message digests are susceptible to becoming vulnerable as time progresses. Some commonly employed techniques have been compromised in previous years, and the encryption powering the electronic signature is also a source of worry. In an internet-based conservation system that verifies against prior communication digests, you must substitute the digest mechanism before the present one becomes hacked or inadequate.

To accomplish this, the system has to initially audit the current digest to ensure it is genuine before constructing a digest utilizing the substitution technique, which can

then be attached to the product's recorded digest collection. However, this procedure necessitates using an impartial third party and raises questions regarding the reliability and accuracy of "recomputed" breaks down as a foundation for authenticating original entries. Such issues may require resolution in a court of law, demanding a system of law competent to make these kinds of decisions (Rosenthal, 2005). Addressing the evolving vulnerability of encryption algorithms and ensuring digital records' long-term integrity and authenticity pose significant challenges in digital preservation. A well-established and trusted mechanism must be in place to manage algorithm updates and maintain the integrity of preserved data throughout its lifecycle.

2.20 Blockchain and the Legal System

Introducing a Blockchain-based Land Registry system does not inherently change the fundamental nature of the existing registry system, whether it operates on deeds or Torren's system. Blockchain technology does not automatically convert a deeds system into a title system or vice versa; it will retain its original characteristics. For instance, the Registrar will not start issuing titles in a deeds system utilizing Blockchain. On the other hand, if a title system employs Blockchain for title transfers, the titles will be securely transferred through the technology, ensuring their preservation. However, Blockchain does not enhance the legal certainty regarding the content and legal implications of the initial block. If there is already uncertainty about the title holder, implementing Blockchain will not address this issue. Similarly, to traditional systems, the accuracy and comprehensiveness of Land Registers can be upgraded by documenting additional transactions, potentially incorporating titles in following blocks, or uploading fresh transactions in the initial block. This process helps maintain accurate and up-to-date Land Registers,

providing an overview of the current state of the registry, akin to a conventional Land Registry system (Aas, 2014).

Table 1

A Summary Table of the Research Gaps Established by the Study

Author and Year	Research Gap
Ministry of Lands (2021). Retrieved 10 December 2021, from https://Lands.go.ke/national-Land-information-management-system-nlms .	Non–existence of a Universal NLIMS in Kenya. Land Transactions records are still largely managed and stored manually.
Njoroge, T. G. (2019). <i>Use of Blockchain to Digitize Land Records and Track Land Transfer Transactions in Kenya</i> . University of Nairobi.	More simulations of Blockchain required to ascertain its integrity in solving land transaction issues.
Snow, P., Deery, B., Kirby, P. and Johnston, D. (2015). “ <i>Factom ledger by consensus</i> ”, available at: www.factom.org .	Inaccurate or Unauthorized entries can be generated in the underlying Property registration database before the documents are delivered as Submissions to a Factom server.
Mulaku G.C. (2005). <i>Land Information Management in Kenya. An Integrated Approach</i> .	No transparency of land transaction records since they are centralized.
Kariuki, J. W. O., Karugu, W. N., & Opiyo, M. M. O. (2018). Challenges facing digitization projects in kenya: case of implementation of national Land information management system. <i>International Journal of Technology and Systems</i> , 3(1), 23-42.	The Lands registry in Kenya is generally administered manually. Land transactions involve third parties who can alter them.
Makoro, D. (2015). <i>Use of Land management information System (NLIMS)</i> . A case study of ministry of Land, housing and urban development, Nairobi.	Most land transactions lack integrity since individuals not authorized can access them.
Muriuki, R. W. (2019). <i>A Tool for tracking Land transactional activities: case of Land registry in Kenya</i> (Doctoral dissertation, Strathmore University).	It is difficult to trace the history involving a land transaction since several land registries exist.
Home, R. (2021). <i>History and Prospects for African Land Governance: Institutions, Technology and ‘Land Rights for All’</i> . <i>Land</i> , 10(3), 292.	Most land transactions lack integrity since they individuals not authorized can access them.
Griggs, L. (2016). <i>The doctrinal coherence of the Torrens system of Land registration in Australia: evolution or revolution?</i>	Traceability of land transaction is difficult.
H.W.O. Okoth-Ogendo. (1991). <i>Tenants of the Crown: Evolution of Agrarian Law & Institutions in Kenya</i> .	It is difficult to trace the history involving a land transaction since several land registries exist
Gaston J. R. (1979). <i>Land Issue in Kenya Politics</i> . Ufahamu, Vol. 9, No. 2, pp. 30 – 58.	No transparency of land transaction records since they are centralized.
Feder, G., & Nishio, A. (1998). <i>The benefits of Land registration and titling: economic and social perspectives</i> . <i>Land use policy</i> , 15(1), 25-43.	Land transactions involve third parties who can alter them.

According to the Ministry of Lands (2021), the NLIMS was designed to enhance the security of Land records, improve accessibility and cut down the cost of Land transactions, boost service delivery and eliminate paper records that breed and catalyze corruption and fraud. They pointed out that Poor record management and bureaucratic processes are some of the hindrances to doing business in Kenya. Therefore, the NLIMS objective was to First-track search, registration, valuation, transfer and issuance of titles in order to facilitate and accelerate investment and development of Land as capital. They went ahead to explain that **ArdhiSasa** will improve public service delivery, foil fraud and stop further deterioration of paper records. It will also improve access, storage and retrieval of Land records. However, the Ardhisasa system is currently live only in the Nairobi Registry and plans are at an advanced stage towards its implementation in the whole country in the near future.

The Factom model for Honduras fails to deal with the issue of record integrity. This aspect is still outside of the realm of what Factom officially provides. As a result, despite the company's claims, it's difficult to see how the existing Factom system will alleviate Hondurans' recordkeeping problems. Hondurans must still register their land possession with the Property Institute, an impartial and trusted third party charged with registering land titles in the SURE Network. This procedure occurs outside the technological structure of a solution such as Factom. Furthermore, regardless of recorded applications, inaccurate or unauthorized entries can be generated in the underlying Property registration database before the documents are delivered as Submissions to a Factom server (Snow, 2015). As a result, assuring the reliability of records remains a significant restriction of the Factom Blockchain model.

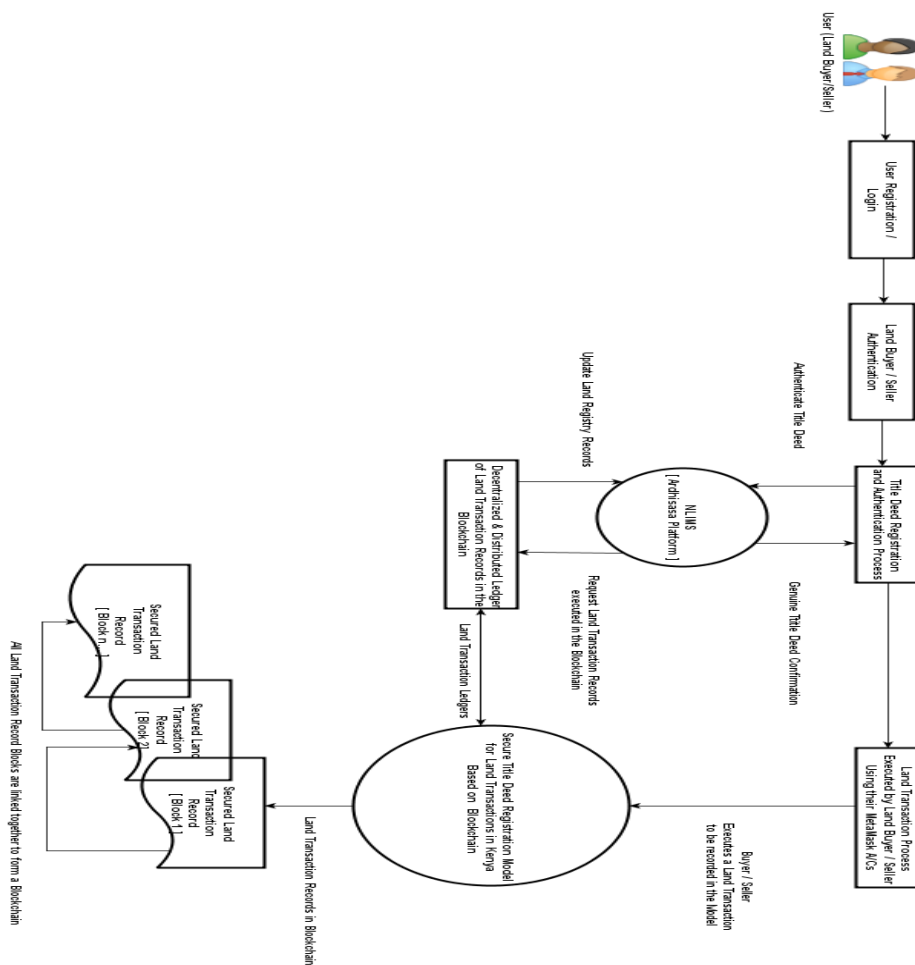
Gaps identified in the Ardhisasa platform and the Factom Model points out to the need for a more secure, transparent and efficient platform for the execution of Land

transaction records in Kenya. The Secure Title Registration Model for Land Transactions in Kenya based on Blockchain presents an alternative platform through which Persistent, Transparent, Temper-Proof and Decentralized Land Transactions records can be executed.

2.21 Conceptual Framework

Figure 7

Conceptual Framework



Source: Author, 2025

For the study to achieve its objective a conceptual framework shown in Figure 7 has been used to demonstrate the stages and the processes through which an individual willing to sell or buy a piece of Land must pass through. The framework is based on a Blockchain distributed ledger technology that guarantees a secure title deed registration

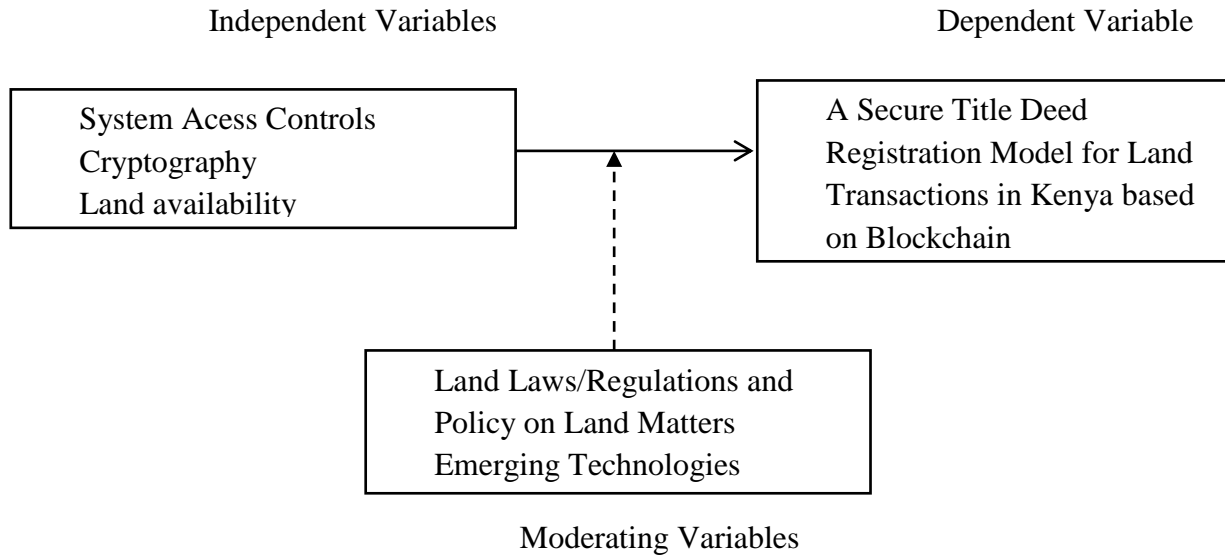
model for Land transactions. Upon the authentication of user credentials, the model allows him or her to execute a Land transaction either as a seller or a buyer. The conceptual framework informed the architectural design of the model which is anchored on Ethereum. In this case, our Ethereum is decentralized Blockchain whose function is to establish a peer to peer network that securely executes and verifies an application code referred as Land registry smart contract. The Smart contract enables Land buyers and sellers to execute Land transactions amongst themselves without a trusted central authority or an intermediary/Land broker. Once a Land has been bought or sold, a transaction record is generated. All transaction records are time stamped, immutable and they can be verified by anyone in the network.

User created Ethereum accounts are used to send and receive transaction records which come along with a cost referred to as Ether. Several transaction records are generated out of the Land registry smart contract and updated as a distributed ledger in the entire peer to peer network. Every transaction is treated as a Block and each Block is linked to the next Block using a cryptographic hash value. This forms an effective tamperproof chain that has all the benefits of using Blockchain in Land transactions. With each block having been updated and secured in the Blockchain, the Kenyan Government agency, for example the Ministry of Lands, or any other interested party, can have access rights to Land transaction records in the Blockchain for purposes of making informed decisions on matters related to Land transactions and management.

A summary of the independent variables, dependent and moderating variables for the model are highlighted below.

Figure 8

Summary of Variables



CHAPTER THREE

RESEARCH DESIGN AND METHODOLOGY

3.1 Introduction

In this chapter, the study presents the design and the scientific methodology that was employed during the inception, development and evaluation of the designed model. In order for the research to achieve its specific objectives and to effectively address the research questions, the study used the proof of concept methodology in the design and development of the specific modules required for successful evaluation the secure title deed registration model for land transactions in Kenya based on Blockchain. A brief discussion on the thematic literature, model evaluation approach, sampling technique and criteria, user feedback tool is also presented in this chapter. The model performance was evaluated by comparing its output against the expected results informed by the research objectives and the research questions. Finally, the chapter presents the ethical considerations that were taken into account by the study as well as the research authorization that was obtained from the National Commission for Science, Technology and Innovation (NACOSTI).

3.2 Research Design

According to Robert Yin (2013), a research methodology refers to a scientific research approach for solving a research problem systematically. Hevner *et al.* (2004) explains that the objective of research design is to provide new knowledge that can be used by researchers and professionals from different disciplines to design solutions to research problems in their fields.

3.2.1 Design Science and Proof of Concept Research Methodologies

The study adopted a blended approach in selecting the research design for model development and evaluation. The Design Science research methodology approach was adopted by the study in an attempt to effectively address the overall main objective of developing a secure title deed registration model for Land transactions in Kenya based on Blockchain. In Kenya, for example, the National Land Information Management System (NLIMS) platform was developed as a result of design science research methodology (Makoro, 2015).

For the study to address the specific objective of designing the specific modules required for the model as well as evaluating them, the Proof of Concept (PoC) research methodology was adopted to define the scope and ascertain the feasibility of actualizing the prototype model within the stipulated timelines.

3.2.2 Thematic Literature Review

The first specific objective of the study was to:

- i. To review the challenges of Land transactions for Land ownership in Kenya.

For the study to achieve objective one above, a thematic literature review methodology was used to:

- ii. Foremost, examine and identify the current challenges experienced by Land Buyers / Sellers in the process of executing Land transactions for Land ownership in Kenya.
- iii. Secondly, to examine and identify gaps and vulnerabilities existing in Kenya's Lands Registry and by extension, the National Land Information Management System, in the management of Land transactions and Land records. Current and relevant literature relating to Land management in Kenya as well as Blockchain

technology was reviewed. The Literature reviewed informed the design and the evaluation of the secure title deed registration model for land transactions in Kenya based on Blockchain.

3.2.3 Proof of Concept Approach

The second, the third and the fourth specific objectives of the study were:

- i. To design a secure title deed registration model for land transactions in Kenya Based on Blockchain.
- ii. To demonstrate how the designed model integrates Blockchain to provide a secure platform for title deed registration for land transactions.
- iii. To evaluate the output of the designed model against the evaluation metrics set out by the study with the view of recommending it for future implementation in Kenya.

In order for the study to adequately address the above specific objectives, a Proof of Concept methodology approach was used as part of the research methodology that informed the design of the specific modules required for prototyping and evaluation. The underlying architecture of the Secure Title Deed Registration Model for Land Transactions in Kenya based on Blockchain was founded on Ethereum (a decentralized Blockchain that has smart contract functionality in it), to provide support for the creation and distribution of Land transaction records as decentralized ledger across the peer to peer network.

3.3 Proof of Concept Methodology

To address the specific of objectives of the study, Proof of Concept methodology was adopted for the design and evaluation of the designed model for a secure title deed registration model for Land transactions in Kenya based on Blockchain. According to

McPherson *et.al* (2023), Proof of Concept (PoC) is an act of determining whether software can be developed in the real world, what technologies should be used in development, and whether the software is likely to be adopted by its intended users. Therefore, adopting the Proof of Concept approach is of great significance because of the following reasons:

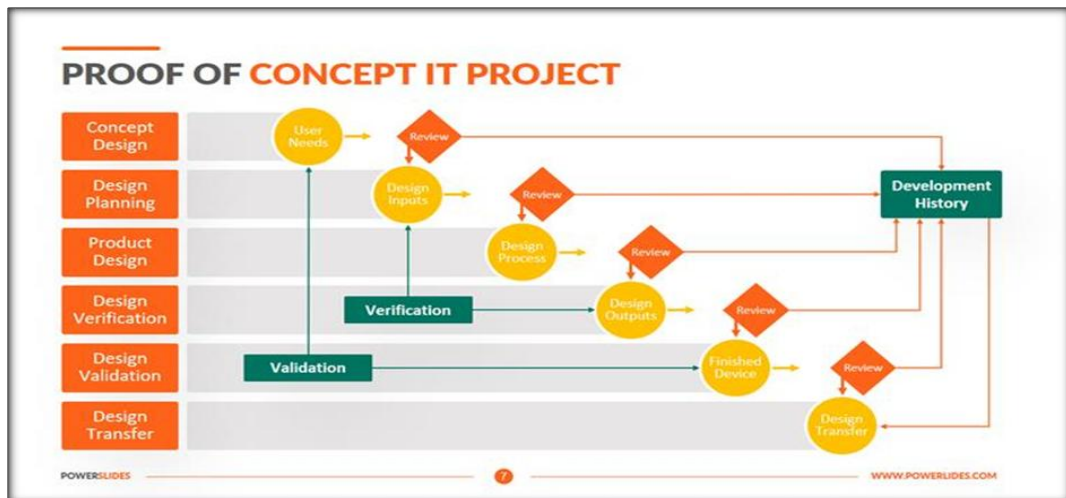
- i. **Validity** – Always ensure that your idea is feasible. For example, can the model be built within the budget? How about the demand for it in the market? What would the return on investment be?
- ii. **Funding.** If you plan on sourcing investors to back your project, you're going to need to show them that you've thought your idea through to the end. Creating an early roadmap on how to monetize your software idea will make a strong statement in creating investor confidence in your end product.
- iii. **Focus Group Evaluation.** Most start-ups waste resources such as time and money in a product that they consider to be useful to users but only to go back and modify the same upon getting user feedback that their idea did not meet its expectation. Taking a proof of concept and evaluate it with your potential user base will help you get early feedback on what people actually want. You can then streamline your idea's development, thus saving you time and money spent on solving the wrong problems.

3.4 Proof of Concept: Model Design Stages

The study adopted the Proof of concept research methodology in the design and evaluation of the model. Six design model stages were considered as shown in Figure 9.

Figure 9

The Proof of Concept IT Project Stages



3.4.1 The Concept Stage

This is the first stage in the proof of concept research methodology whereby the user requirements were taken into consideration as part of what the model was expected to address. The first specific objective of the study is to review the existing challenges related to Land transactions in Kenya. The outcome of this objective informed the user needs that were deliberated at the concept design stage.

3.4.2 The Planning Stage

The second objective of the study is to design a model for evaluation. The model requirements and specifications including design inputs were considered at this stage. The model requirements must accommodate all user requirements that were discussed accepted during the concept design stage.

3.4.3 The Product Design Stage

This is the stage whereby the model design processes were developed based on the model requirements and guidelines specified in the planning phase. The model functionalities were designed based on the user and model requirements.

3.4.4 The Design Verification Stage

For the research objectives to be achieved, the model was evaluated against its expected output. This was done at this verification stage whereby the model inputs were verified against its outputs to ascertain that the model address the objectives of the study.

3.4.5 The Design Validation Stage

This was basically the last stage in the development of a secure title deed registration model for Land transaction in Kenya based on Blockchain. The design validation stage involved evaluating the overall suitability of the resultant model designed for the study. This was to ensure that user requirements at the concept design stage were addressed by the final model that has just been designed as the finished product.

3.4.6 The Design Transfer Stage

The design transfer stage is the last of the Proof of Concept research methodology that was discussed in the PoC Methodology. This stage involves the deployment of the model to its application area for use. The study recommends that the model may be adopted and deployed by the Government of Kenya, based on the outcome of model evaluation, to address the challenges currently experienced by Land buyers/sellers while executing transactions. The study also recommends that future upgrades and maintenance shall be done on need basis upon request by the user or as a result of emerging application technologies relevant to the Industry.

3.5 Benefits of the PoC Research Methodology

Clay *et.al* (2021) highlighted some benefits of using Proof of Concept (PoC) research methodology as follows:

- i. Proof of concept is an important step of the product development process because it forms the base for a product prototype.

- ii. Reduction in the overall risk of project failure by identifying gaps that will harm their chances of succeeding.
- iii. Act as a tool to forecast the feasibility of the projects.
- iv. The ability for the customer to visualize the look-and-feel of the solution early on therefore providing a chance to assess design decisions early in the process.

3.6 Limitations of Proof of Concept Methodology

According to *Toschi et.al* (2021) project details provides descriptive information for the project development, there could be still chances to get varied from the project flow which may give different outcomes. Proof of the concept sometimes appears as a barrier. It can limit your creativity as it imposes some guidelines to be followed. It adds another level of scrutiny. Moreover, to conduct such activities, funds are required which is another indirect expense. To carry out this pre-prototype experiment it takes a lot of time as well.

To address these limitations, the study adopted a descriptive evaluation criteria based on Persistence, Time-Stamping, Transparency and Decentralization of Land transaction records as the critical features of Blockchain technology as discussed in Chapter Four.

3.7 Criteria-Based Evaluation Approach

According to *Goldkuhl et.al* (2003), the Criteria-based evaluation approach takes into consideration the designed model's user interfaces as the point of focus during the interaction between a user and the system. This therefore acts as the basis for the evaluation alongside a set of predefined criteria. The study adopted the Criteria-based evaluation approach in order to ascertain whether the designed model exhaustively addressed its specific research objectives and research questions. Subjective sampling was adopted for model evaluation.

The study randomly selected a group of 10 experts to carry out the exercise of evaluating the designed secure title deed registration model for Landtransactions in Kenya based on Blockchain. To achieve a reliable user feedback on the designed model based on the set of predefined criteria outlined in this Chapter of the study, a Likert scale tool was subjected to the users. A sample of the Likert scale tool adopted by the study is presented in the appendix section. The random group of 10 sampled experts used for evaluating the model were distributed as follows:

- i. Four (4) Surveyors from the Ministry of Roads, Housing and Urban Development – Department of Kenya Highways Authority, Nakuru County Office.
- ii. Three (3) Ministry of Lands officers – Lands Registry, Nakuru County Office.
- iii. Three (3) Ministry of Lands officers – Lands Registry, Nakuru County, Rongai Sub-County Office.

The designed model in the study is based on Ethereum Blockchain and it adopted the AES-CMAC algorithm as a plugin to provide an additional layer of security to the Land transactions being executed in model. This prevents unauthorized access and tampering of data on the Blockchain. By using smart contracts, the execution of AES-CMAC can be automated, making the process more efficient and secure. Metamask is used interact with the smart contract, making it easier for users to execute the AES-CMAC algorithm on the Blockchain (Mathur & Nandini, 2023).

The model performance was then evaluated by comparing its outputs against the expected results. Key benefits of Blockchain technology include an immutable history of transactional records, such that no one can ever doubt the authenticity of the records. The transactional records are permanently linked in the system in a way that no one can ever tamper with or perform a forgery on them. The records can be seen by any party, at any

given time, from anywhere in the World. Chapter Four of the study presents four critical features of Blockchain technology that formed the basis of model performance evaluation, listed as: Persistence, Time-Stamping of Records, Traceability and Decentralization (Balakrishnan, 2021).

3.8 Ethical Consideration

According to Fleming (2018), ethical considerations in research refer to those principles that guide research designs and practices. The principles include voluntary participation, anonymity, informed consent and confidentiality among others. For the study to achieve its overall objective, all the ethical considerations' principles, as required by the regulating authorities in the education sector in Kenya were taken into consideration. A formal letter of introduction from the Institute of Postgraduate Studies - Kabarak University, was obtained for the purpose of the study. This is presented in the Appendix section of the study.

The Model evaluation was performed by users. They were informed about the purpose of the study and their role during evaluation. Their honest feedback and opinion about the status of the Lands registry in Kenya and their experience with the designed model in the study shall always be treated with utmost confidentiality.

As a prerequisite for conducting research in the Republic of Kenya, express permission and authority must be obtained from the National Commission for Science, Technology and Innovation (NACOSTI). The study complied with these requirements. This is presented in the Appendix section as well.

CHAPTER FOUR

MODEL DESIGN AND EVALUATION

4.1 Introduction

This chapter presents a discussion on the Model design and Evaluation criteria. The first specific objective of the study was achieved through a thematic literature review done on Chapter Two of the study. The last two specific objectives of the study were to design a model for evaluation. Design Science research methodology was adopted to provide a roadmap into the model design and development while Proof of Concept research methodology was used for the development of specific modules for evaluation based on the six stages of the methodology as discussed earlier in Chapter Three. Chapter Four presents a detailed discussion on the user and software requirements for the model, its expected functional and non-functional requirements, the evaluation metrics used and finally the evaluation of the output of the designed model to ascertain whether it meets the specific research objectives of the study and that it exhaustively addresses the research questions in the study.

4.2 Model Design

The Proof of Concept research methodology was adopted by the study in order to address the specific research objectives and its research questions as well. In this section, the study specifically focused on answering the last three research questions presented in Chapter One, that is:

- i. How can a secure title deed registration model for Land transactions in Kenya based on Blockchain be designed?
- ii. How can a secure title deed registration model for land transactions in Kenya based on Blockchain be demonstrated?

- iii. What requirements are needed for a successful evaluation of the designed model as well as the expected outcomes to inform the model's recommendation for future implementation in Kenya?

In Chapter Three, the study outlined six stages of the Proof of Concept methodology. By adopting the six stages, the above research questions were adequately addressed by taking into account the user needs as well as the design inputs of the model. The study used the designed model to demonstrate how a secure title deed registration can be used to execute a Land transaction on Blockchain. As discussed earlier in Chapter Two under the conceptual framework, the model is designed to interact with two users at any given point in time. The two users are a Land buyer and a Land seller, who must be registered and authenticated into the system.

4.2.1 Design Requirements

This section provides a detailed discussion on the requirements for designing a secure title deed registration model for Land transactions in Kenya based on Blockchain. Here, the study considered the first three stages of the Proof of Concept research methodology approach in order to address the model design requirements as follows:

- i. **Concept Design Stage**

At this stage, user inputs and requirements were taken into consideration as part of designing a model for the execution of secure Land transactions.

- ii. **Design Planning Stage**

Here the model inputs including its requirements and specifications including design inputs and outputs were taken into account and aligned with all the user input requirements at the concept design stage.

iii. **Product Design Stage**

At this stage, the model design processes were developed by integrating the user inputs as well as the model design inputs and requirements. Model functionalities were also considered at this stage.

4.2.2 The Software Specifications

To effectively design and develop a secure title registration model for land transactions in Kenya based on Blockchain, the following software specifications were considered by the study:

i. **React.JS**

This is a declarative, efficient and flexible JavaScript Library for building reusable User Interface components. It is an Open Source component-based frontend library responsible for the view layer of an application. It was initially developed and maintained by Facebook in the year 2013, and then later on, Facebook used it in its products such as WhatsApp & Instagram. The goal of React.js is to develop User Interface for improving the speed of apps. It utilizes a virtual JavaScript object known as the Document Object Model (DOM). DOM is an Application Programming Interface (API) for both HTML and XML documents that define the logical structure of documents, how to access documents and how the documents can be manipulated (Thakkar, 2020). The specific components that were used by the study were as follows:

- a. **App** – This is the component that was used to handle routing from one page to another for example the homepage, the sign up page and the register page.
- b. **Main** – This component handles the frontend interaction between the designed model and the smart contract.

- c. **Navbar** – The component handles the display of user Metadata (personal account information) and allows the user to log out.
- d. **Dashboard** – This is the component for handling all the logic involving interaction with Metamask accounts.
- e. **Login** – This component is for handling Logins.
- f. **Signup** – It's the component for handling the Sign ups

ii. **Solidity**

This is an object-oriented programming language that is used to write and execute Smart Contracts that can be implemented on different Blockchain platforms for example Ethereum (Liu *et.al* 2020).

iii. **Node.JS**

Pradel et al. (2021) explains that Node.JS is a cross platform JavaScript runtime environment. It is open source. It contains a default Node Package Manager (NPM) which is used to handle the installation of dependencies like the libraries that are required for the application to function.

iv. **Metamask**

This is a software Crypto currency wallet that is used to interact with the Ethereum Blockchain. Metamask enables a user to have access to their Ethereum wallet via a browser extension or a mobile application, which can then be used to interact with decentralized applications (Shakya, 2022). Each and every time a user makes a new transaction that is to be added to Ethereum, the user is charged a certain amount of Gas/Ether as the transaction cost. According to Arroyo et al. (2021), an Ether/Gas refers to the unit for measuring the amount of computational effort necessary for the execution of a specific task in an Ethereum network. The Land seller and the buyer are also

required to pay a certain amount of money in order to acquire a piece of Land. All these transaction are handled by the Metamask wallet. The Dashboard component of the model handles the connection with Metamask.

v. **Firestore**

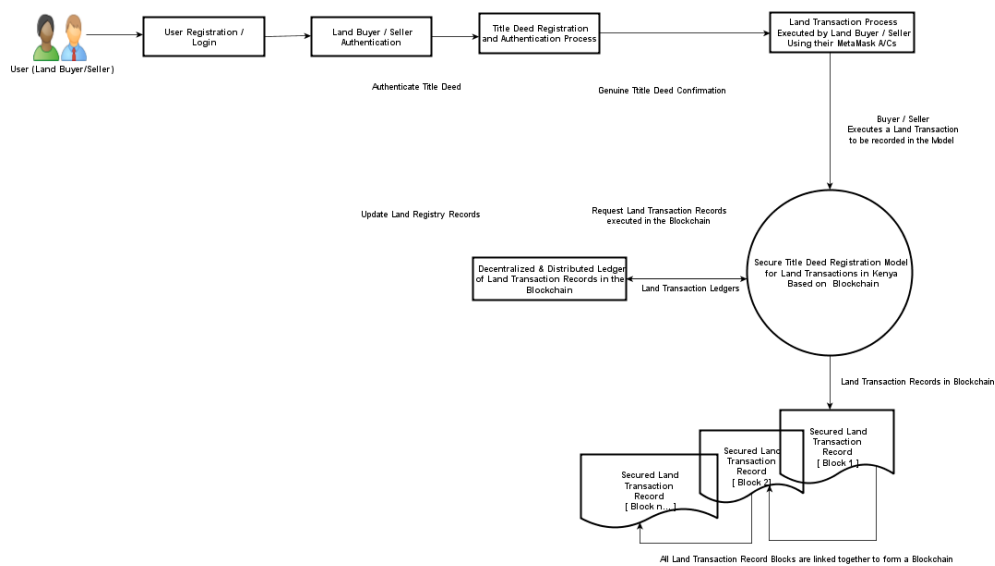
The research used Firestore as a real time database for the model.

4.2.3 Expected Functional Requirements

To complete a land transaction, two users are required to interact with the model; a land buyer and a land seller. Upon account creation, a user is expected to login to the model and proceed to initiate a land transaction either as a land seller or a land buyer. A successful land transaction record is then generated by the model and a copy is posted on the peer to peer network as a block on which future records of the same land transaction record shall be chained together to form a Blockchain of land transaction records. Figure 10 below demonstrates the components of the expected functional requirements of the model.

Figure 10

Components of the Functional Requirements of the Model



A summary of the components of the functional requirements are provided as follows:

- i. User – This refers to either a land buyer or a land seller
- ii. User registration/Login – This component allows the user to register by providing his/her credentials.
- iii. User Authentication component – This allows authenticated user access to the model.
- iv. User wallet component – This component allows a user to create an Ethereum account used to facilitate land transaction on the Blockchain network using Ether.
- v. Land Registry Model based on Blockchain – This refers to the model component that allows the user to buy and sell land using his/her account where land transaction records are executed as smart contracts that generate Blocks
- vi. Land Transaction Records – This component refers to the records of sold/bought land generated as Blocks and linked together to form Blockchain.
- vii. Peer to Peer contract Wallet – The component refers to the entire land transaction records decentralized and distributed amongst all the participants in the network.

The study considered all the above functional requirements of the model in order to overcome some of the limitations of the Proof of Concept methodology discussed earlier in Chapter Three.

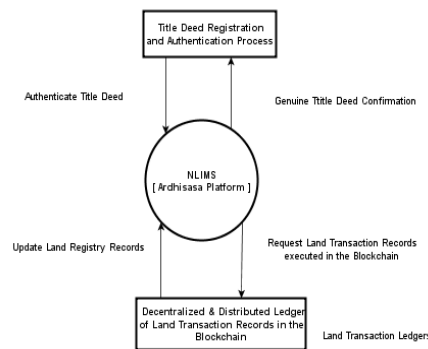
4.2.4 Expected Non-Functional requirements

Chapter Two presented a conceptual framework that guided the design of the model in the study. However, one component of the model is expected to be a non-functional requirement, which is the National Land Information Management System (Ardhisasa Platform). This was informed by financial, time and privacy constraints that were

experienced during the entire research period. Figure 11 below demonstrates the component of the expected non-functional requirement of the model.

Figure 11

Component of the Non-Functional Requirement of the Model



The summary of the component of the non-functional requirement of the model is explained below:

- i. The National Land Information Management System (Arthisasa Platform) – This was to perform two functionalities; one is to authenticate the validity of Title deeds being held in the Lands registry in Kenya during the title registration process for land transactions processes to be executed in Blockchain. Secondly, it is to update all land transaction records executed in Blockchain to allow an upto date Lands registry in Kenya.

During the design stage, this component was considered to be non-functional since only two users were considered as the principal transaction owners in the model. The NLIMS (Arthisasa Platform) could not be used due to the sensitivity and confidentiality nature of land records in it.

4.2.5 The User Interface Design

User Interface Design refers to developing the required components and elements that enables a user to interact with the model. The process of developing the components

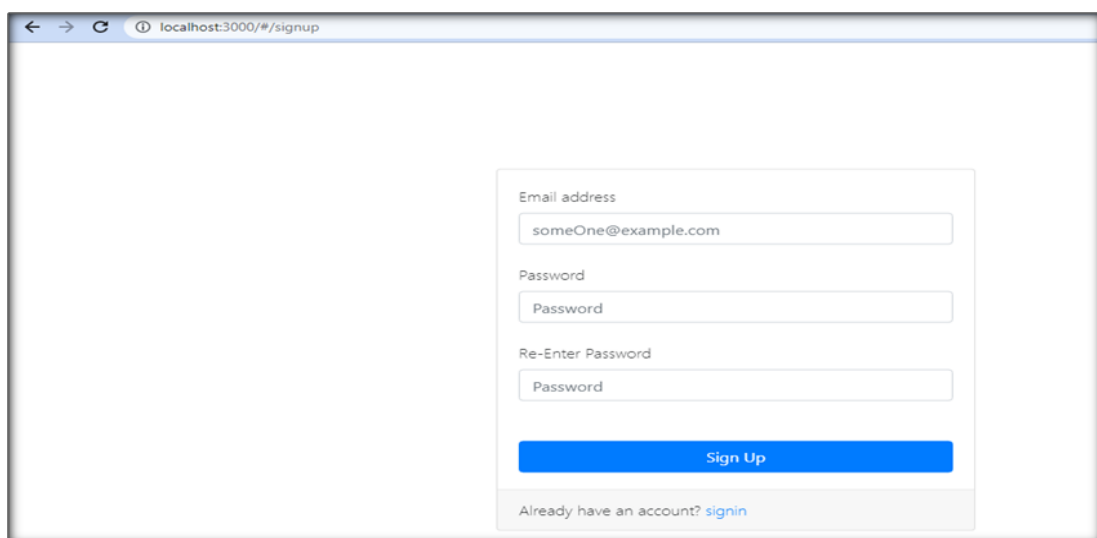
should take into consideration user activities that are expected to be performed on the model. The elements of the user interface should be easily accessible, easy to understand and use (Ruiz, 2021). The conceptual framework in Chapter Two of the study, explained that a user interacts with the model either through a registration or login procedure.

4.2.5.1 User Registration and Login

The Land seller and buyer must provide their credentials to the model in order for them to interact with the system. This was done by providing the user with two options. The first option was to allow a new user to sign up into the model by providing their relevant information. In the study, a sign up interface was designed and developed. This requires a Land buyer or seller to provide a valid email address and a strong password for account creation in the model. Upon completion of the registration process using valid credentials, user is required to re-enter his or her password and proceed to sign up into the model. The user interface sign up form for user registration is shown in Figure 12 and the component source codes are in Appendix 1(A).

Figure 12

The Model's User Interface Registration Form



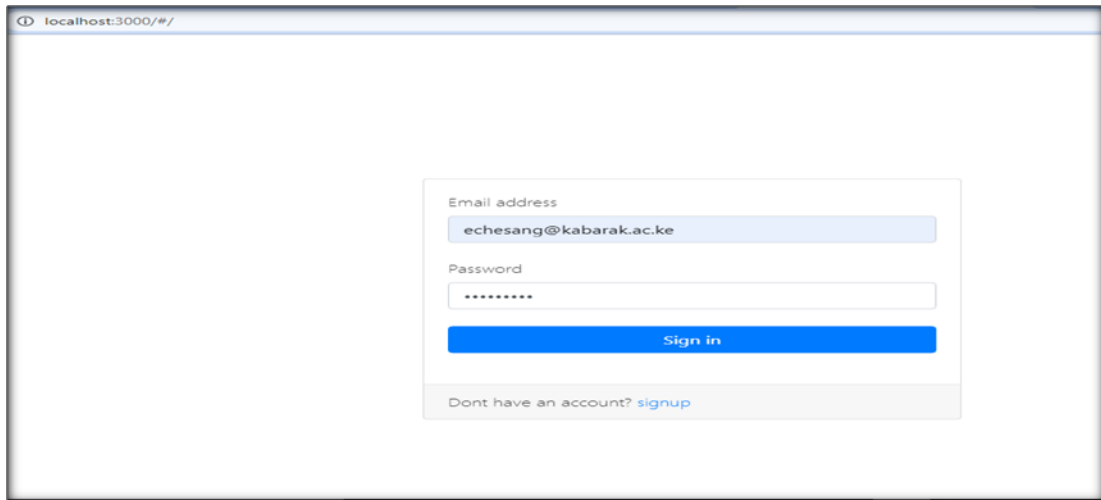
The image shows a web browser window with the address bar displaying 'localhost:3000/#/signup'. The main content area contains a registration form with the following elements:

- An 'Email address' input field containing the text 'someOne@example.com'.
- A 'Password' input field containing the text 'Password'.
- A 'Re-Enter Password' input field containing the text 'Password'.
- A prominent blue button labeled 'Sign Up'.
- A link at the bottom that reads 'Already have an account? [signin](#)'.

With valid credentials supplied, the Land buyer or seller proceeds to sign up into the model for account creation. The created account allowed a user to perform model login. The sign up form is shown in Figure 13 and the component source code is in Appendix 1(A).

Figure 13

The Model's Sign Up Form

A screenshot of a web browser window showing a sign-up form. The browser's address bar displays 'localhost:3000/#/'. The form is centered on the page and contains the following elements: an 'Email address' label above a text input field containing 'echesang@kabarak.ac.ke'; a 'Password' label above a text input field filled with asterisks; a blue 'Sign in' button; and a link that says 'Dont have an account? [signup](#)'.

4.2.5.2 User Authentication

The objective of authentication is to prove the user's identity. It is therefore the process through which a user who is requesting for an access to a resource in a system can be identified through their username and password (Sanni, 2022). For the model to achieve authentication, a Land seller and buyer were required to sign up into the model using the correct email address and password for their account. Any invalid sign up credentials were not accepted by the model.

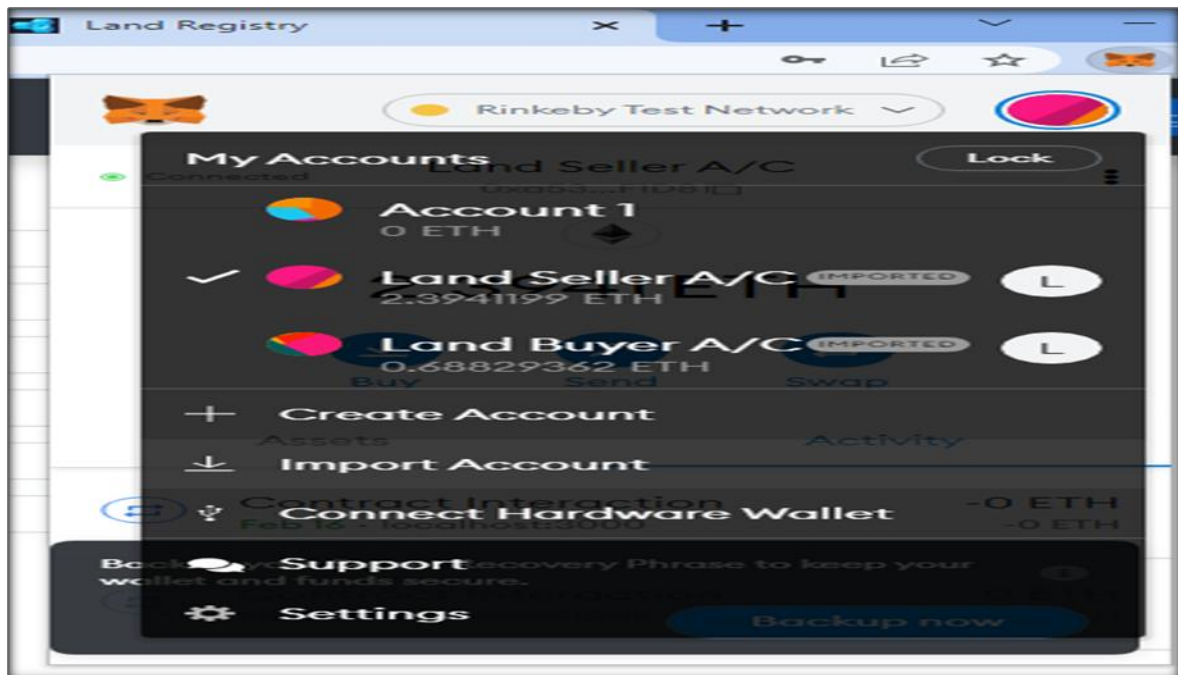
4.3 User Accounts

Previously in Chapter Four, the study provided the software specification requirements for the design and evaluation of a secure title deed registration model for Land transactions in Kenya based on Blockchain. The study utilized **Metamask** to create the

essential crypto wallet account that enables a user to execute a successful transaction in the designed model. Metamask is integrated into the designed model. The Land buyer/seller accounts must have ether in them to enable land transaction to be executed. Upon successful authentication into the system, a user has to be identified as either a Land buyer or a Land seller in their Metamask accounts. Therefore, two accounts were created, one for a Land buyer and another for a Land seller as shown in Figure 14 and its component source code in Appendix 1(B):

Figure 14

Land Buyer / Seller Accounts

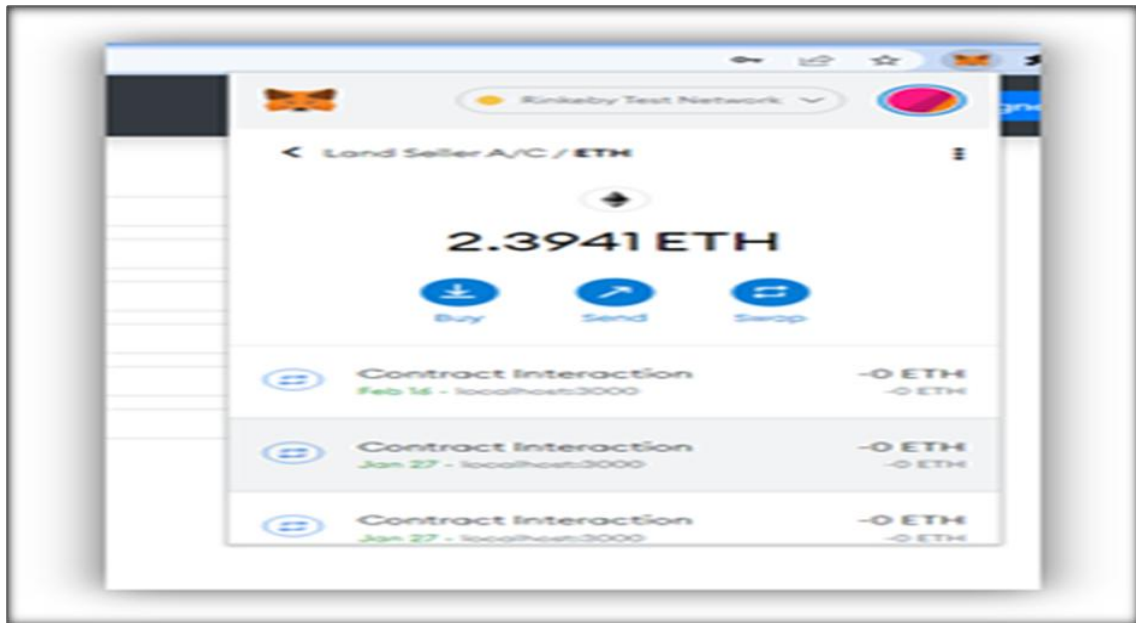


4.3.1 Land Seller Account

In the study, the designed model was used to execute Land transactions entered by the Land seller/buyer. Ethereum Blockchain was used in the model since its features supported the execution of the Land transactions as smart contracts. This was achieved by the creation of a Land Seller A/C in the Metamask as shown in Figure 15 and its source code on Appendix 1(B):

Figure 15

A Land Seller A/C loaded with ETH



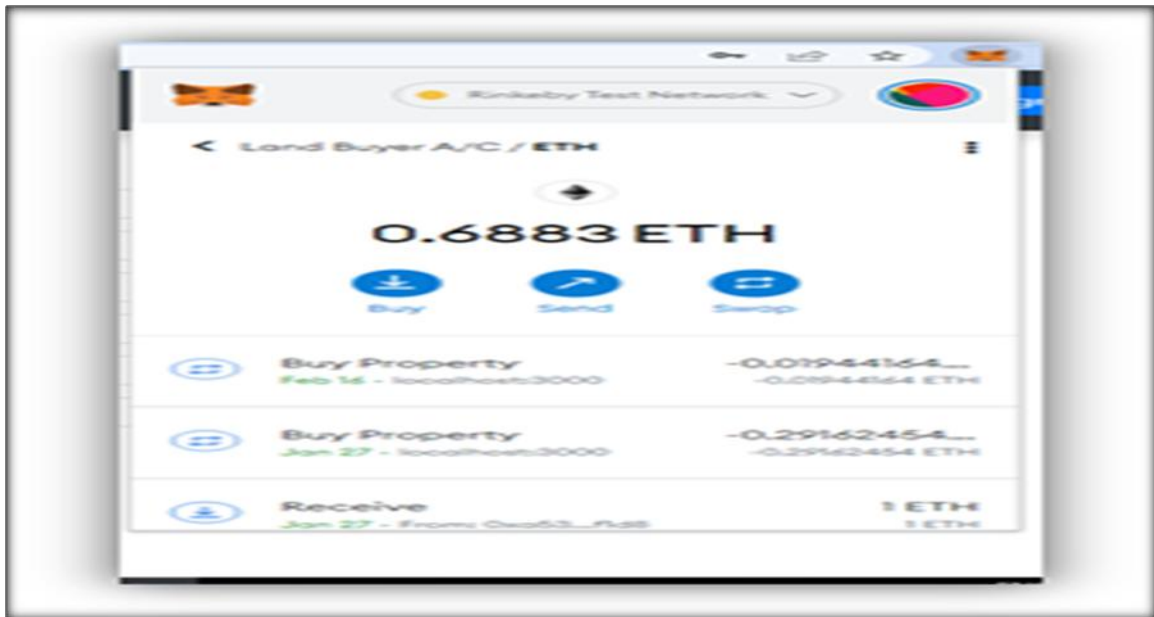
The objective of creating this account was to allow a Land seller initiate the process of selling his/her Land in the model. The created Land Seller A/C has to be loaded with Ether (ETH) that is required by the Land seller to execute a Land transaction as a smart contract in the designed model. As seen in Figure 14 above, the Land Seller A/C is loaded with **2.3941ETH**.

4.3.2 Land Buyer Account

This was the second account created in the Metamask. The objective for the creation of this account, Land Buyer A/C, was to demonstrate how a Land buyer can initiate the process of acquiring Land in the designed model. Figure 16 demonstrates how the Land Buyer A/C was created in the designed model. Its component source code is in Appendix 1(B).

Figure 16

A Land Buyer A/C loaded with ETH



The created Land Buyer A/C was loaded with Ether (ETH) that was utilized by the Land buyer to execute a Land transaction as a smart contract in the designed model. In Figure 16 above, the Land Buyer A/C was loaded with **0.6886 ETH**.

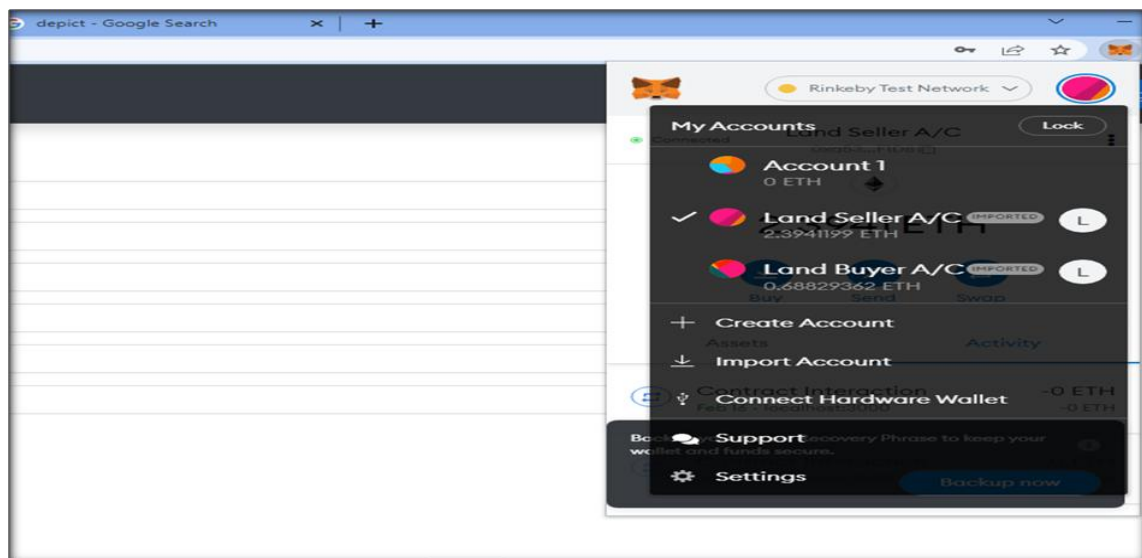
4.4 Title Deed Registration

Metamask is used to create cryptocurrency wallet that acts as a gateway to a Blockchain platform such as Ethereum (Prasad, 2021). The study adopted the same approach whereby the Metamask was used to provide a gateway to the designed model user accounts. The Metamask was configured with custom Remote Procedure Call (RPC) that was provided by the evaluating network and it enabled the Land buyer/Seller to have access to their Ethereum wallets using a browser. The browser is then used to interact with decentralized applications (Shakya, 2022). A user must be in possession of Ether/Gas in their specific user accounts for purposes of executing a transaction as mentioned earlier in this Chapter. A successful transaction executed in the Blockchain cost the user some amount of Ether/Gas. It is therefore recommended that a user must

always be economical while executing land transactions in the designed model to avoid running out of Ether/Gas (Arroyo, 2021). For the study to achieve its specific research objectives, a user was assigned an account in the Metamask, either as a Land Seller or a Land Buyer as shown in Figure 17 and Component source code in Appendix 1(D):

Figure 17

Land Seller A/C and Land Buyer A/C in Metamask



While executing a Land transaction, the designed model allows a user to select an appropriate account from the Metamask on which the Land transaction is to be executed. In the study, only two accounts were created in Metamask. One account for the person selling Land, referred to as Land Seller A/C and the other account belongs to the person buying Land, referred to as Land Buyer A/C.

4.4.1 The Deed Registration Form Requirements

In Chapter Two of the study, the conceptual framework depicts a secure title deed registration model for land transactions in Kenya based on Blockchain. To demonstrate the working of the designed model towards achieving the specific objectives set out in the study, a user interface form was developed in the designed model. The objective of

developing the user interface form was to allow a user, in that case, a Land buyer/seller to register their Land title deeds in the model to facilitate Land transactions. For the designed model to address its specific objectives as outlined in Chapter One of the study, a Land buyer/seller was required to provide the following Land title deed registration details:

- i. **The Size of Land** – Land sizes are usually determined by International Land measurement standards (Kavanagh, 2018). The Land size used in the model was in Hectares.
- ii. **The Location of the Land** - The model considered the description of the physical address of where the Land is located in Kenya.
- iii. **The Land Holder's Name** – The Model requires that the name of the person owning the Land must be provided.
- iv. **The Land Registry Number** – This is the Land's title deed registration number.
- v. **Land Holder's ID** – This is the National Identification Number of the Land buyer/seller as provided for by the Government of Kenya.
- vi. **The Land Price** – This the cost of buying/selling Land. The transaction costs are in Kenyan Currency.

The above Land buyer/seller details were captured in the designed model as shown in user's form in Figure 18. Appendix 1(C) contains the component source codes.

Figure 18

Title Deed Registration Form

The screenshot displays a web browser window with the address bar showing 'localhost:3000/#/home'. The page content includes a dark header with 'Welcome Back' on the left and a blue 'Signout' button on the right. The main content area is titled 'Register Property' and contains a form with the following fields: 'Size in Hectares', 'Location', 'Holder Name', 'Land Registry Number', 'Holder ID', and 'Price'. A dark 'Add Item' button is positioned below the 'Price' field.

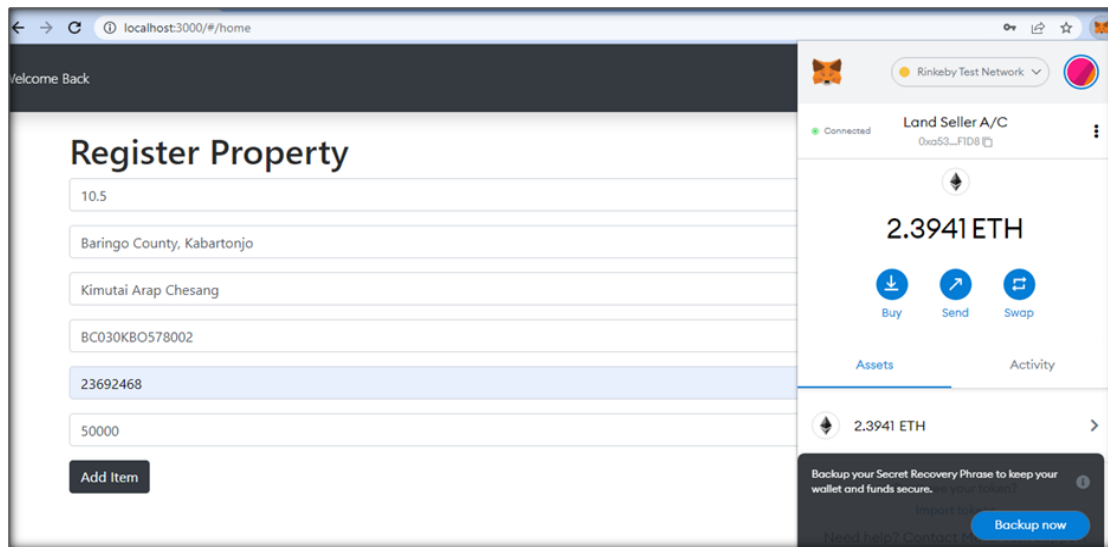
4.4.2 Title Deed Processing

Earlier in the beginning of this section of the study in Chapter Four, several software specifications required for developing the model were highlighted, Solidity being one of them. Solidity is the dominating objected oriented programming language that is used to implement smart contracts for Ethereum Blockchain (Zdun, 2018). According to IBM (2022) a smart contracts refers to programs stored in a Blockchain that run when predetermined conditions are met and they are used to automate the execution of an agreement such that all participants can be immediately certain of the outcome without any intermediary's involvement.

To execute a Land transaction from the designed model, a user must identify himself/herself either as a Land buyer/seller. This process begins by selecting the appropriate account from Metamask and then proceeds to provide the accurate title deed registration details as demonstrated in Figure 19 and the Appendix 1(D) contains the component source codes.

Figure 19

Title Deed Registration BC030KBO578002 initiated from Land Seller A/C



In the designed model, a Land seller initiates the processing of selling Land by providing the title registration details; The size of land to be sold, the physical location of the land, the official name of the land owner as it appears in their national ID, the official title deed number as it appears in the Lands registry, the national ID number of the person executing the transaction and lastly the price of the land be transacted. All these details should be captured in the title deed registration form shown in Figure 16 above. As earlier explained in Chapter Two of the study, the verification, authentication and confirmation of the land ownership details are obtained from the National Land Information Management System (NLIMS) which is currently the Ardhisasa platform in Kenya.

4.4.3 Title Deeds Authentication

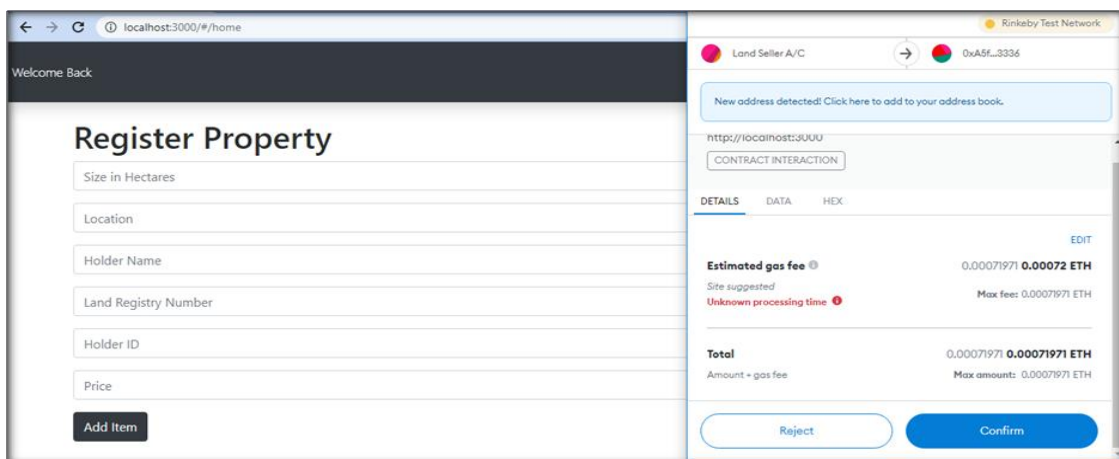
The NLIMS/Ardhisasa platform's function is to confirm that all title deeds used for the execution of land transactions in the designed model are authentic. However, due to privacy constraints, this functionality was not fully integrated into the model as a functional requirement. To simulate the functionality of the NLIMS/Ardhisasa platform,

the study established an authentication instrument in the designed model. The authentication mechanism was developed in such manner that once a Title deed, for example a title deed registration number **BC030KBO578002** is generated and captured in the model for purposes of executing a land transaction, the model shall not allow the exact title deed number to be entered as a new title deed registration into the system.

The function of the **Add Item** button in the designed model is to upload the title deed registration details into the model's smart contract for execution. Using the example in Figure 16 above, a seller's Land whose title deed registration number is **BC030KBO578002** was uploaded into the smart contract upon the seller clicking the Add Item button. As discussed earlier in the study, every transaction that is executed in this Ethereum Blockchain model comes with a cost. Therefore, upon clicking the Add Item button, Metamask must confirm that the seller has enough Ether/ETH to meet the cost of executing the Land transaction in the model. Figure 20 demonstrates how a Land seller whose title deed registration number **BC030KBO578002** was able to effectively perform a Land transaction as prompted by the Metamask. Appendix 1(E) contains the component source codes.

Figure 20

Land Transaction Confirmation in Metamask



4.5 Land Transactions

The designed model ensures that only authentic title deeds are accepted and captured into the system as secured records of land transactions. The system allows a user to view the status of a land transaction as either land on offer for sale or on offer to be bought. The system user (land buyer or a land seller) is also in a position to see:

- i. The size of the land.
- ii. The location of the land.
- iii. The land owner details which is now encrypted for security purposes.
- iv. The title deed registration number.
- v. The price of the land
- vi. The land transaction date
- vii. The land transaction time
- viii. The status of the land as either available for sale or already sold.

Upon the confirmation of a Land transaction by a land seller, the designed model will then deploy the transaction as a smart contract in the Lands registry, from where a potential Land buyer may express his/her interest in the acquisition of the said Land.

Figure 21 demonstrate how Land registered with title deed number ***BC030KBO578002*** was uploaded into the secure title deed registration model for Land transactions in Kenya based on Blockchain. Appendix 1(E) contains the sources codes for the component.

Figure 21

A Land transaction for registered title deed BC030KBO578002 in the Designed Model

#	Size	Location	Land Owner Address	Land Registry Number	Price	Transaction Date	Transaction Time	State
1	0.1 ha	Kiambu	0xa53e44a65837b8Ab219945e244627be0c3E27F1D8	A12232GH3424895	87546 kshs	Thu Jan 27 2022	12:11:34 GMT+0300 (East Africa Time)	Buy
2	0.25 ha	London	0xa53e44a65837b8Ab219945e244627be0c3E27F1D8	LR0252022	1500000 kshs	Thu Jan 27 2022	17:20:58 GMT+0300 (East Africa Time)	Buy
3	0.1 ha	Kabarnet	0x56Ebb65bed04866Fc3d3A8A38DDF40850F089E64	LR0012022	150000 kshs	Thu Jan 27 2022	17:28:29 GMT+0300 (East Africa Time)	Sold
4	02 ha	Kabarak	0x56Ebb65bed04866Fc3d3A8A38DDF40850F089E64	v2024	10000 kshs	Wed Feb 16 2022	13:52:09 GMT+0300 (East Africa Time)	Sold
5	10.5 ha	Baringo County, Kabartonjo	0xa53e44a65837b8Ab219945e244627be0c3E27F1D8	BC030KBO578002	50000 kshs	Sat Feb 19 2022	11:21:36 GMT+0300 (East Africa Time)	Buy

As seen in Figure 21 above, a successful Land transaction was executed as smart contract in the designed model based on the ethereum Blockchain. The land registration details as well as the transaction details provided by the Land seller were captured in the smart contract. Anyone in the peer-to-peer distributed network is now able to view the status of the Land in the distributed ledger as either sold or can be bought.

4.6 Model Evaluation

This study adopted the Proof of Concept (PoC) research methodology in the design and development of the specific modules required for model in line with the specific objective one and two in Chapter One. Chapter Three presented a discussion on the PoC development stages essential for actualizing the design of a secure title deed registration model for Land transactions in Kenya based on Blockchain. The Six design stages that were considered by the study in the PoC research methodology were: the Concept Design stage, Design Planning Stage, Product Design stage, Design Verification stage, Design Validation stage and lastly the Design transfer respectively. In Chapter Four, under the Design Requirements section, the study adopted the first three stages of the

PoC in the development of the model's user requirements, design requirements and its functional requirements. The study adopted the last three stages to demonstrate the design outputs, perform verifications on Land transaction being executed as a solidity smart contract in the model and performing model validations as well.

Upon completion of the designed secure title registrations model for land transactions in Kenya based on Blockchain. Evaluation was necessary to determine that the model fundamentally addressed the research objectives of the study.

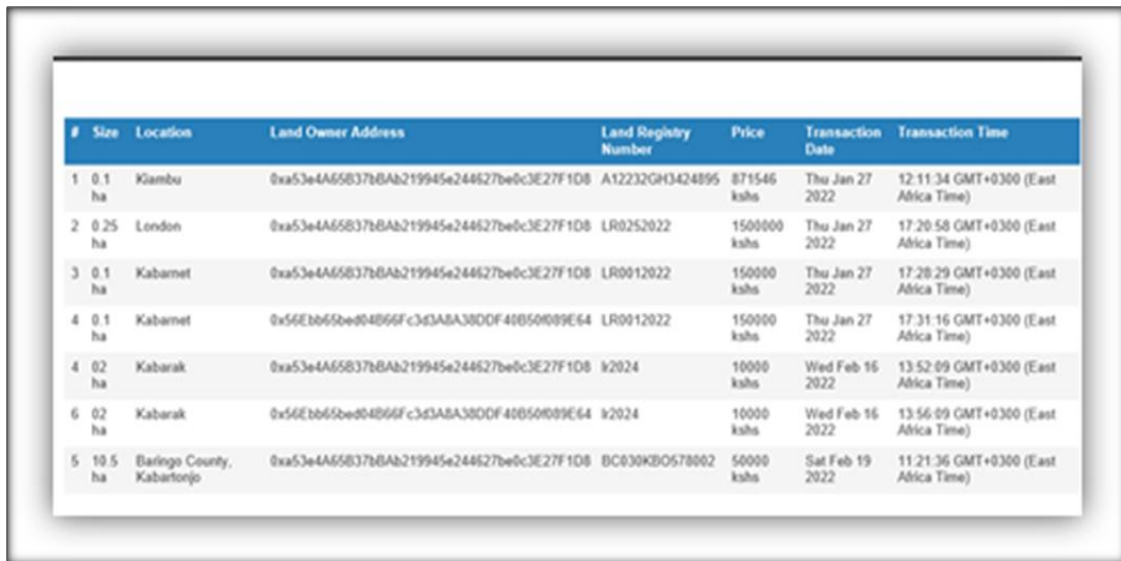
4.6.1 Design Outputs

In the previous sections of this Chapter, the study explained how a Land transaction was executed as smart contract in the designed model. The model was able to generate an output of all land transaction entries executed in the system by the user, in this case a land seller. The land transaction entries are presented as a list of smart contracts generated by the model. A case in particular is an entry of a smart contract indicated as transaction #5, with its title deed registration number indicated as *BC030KBO578002* in the list of all the land transaction entries executed in the model.

As discussed earlier, that transaction was initiated by a user who was selling a piece of Land. Once the selling process was completed, the model was able to generate an output in the form of a summary report on all land transaction entries in the system including transaction #5 as shown in Figure 22 and source codes in Appendix 1(E)

Figure 22

Output Summary Report of All Land transaction Entries executed in the Model



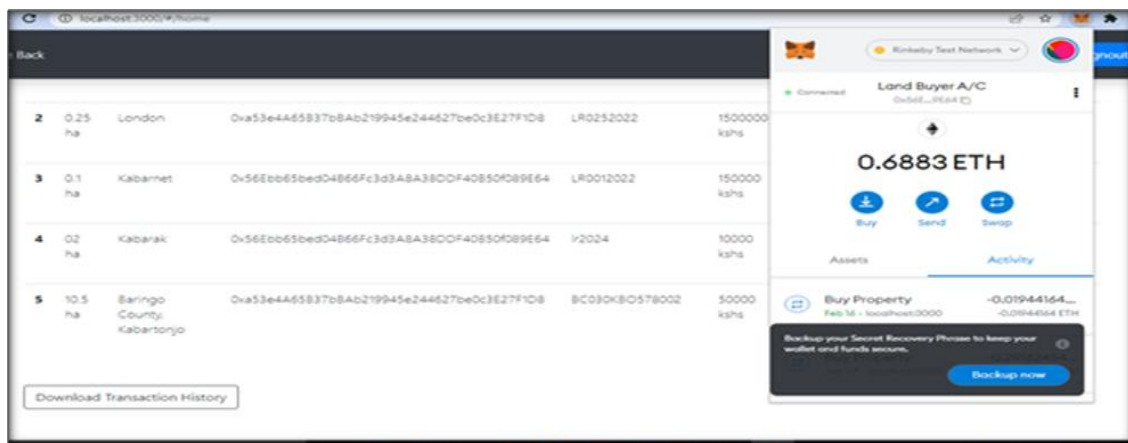
#	Size	Location	Land Owner Address	Land Registry Number	Price	Transaction Date	Transaction Time
1	0.1 ha	Kiambu	0xa53e4a65837b8Ab219945e244627be0c3E27F1D8	A12232GH3424895	871546 kshs	Thu Jan 27 2022	12:11:34 GMT+0300 (East Africa Time)
2	0.25 ha	London	0xa53e4a65837b8Ab219945e244627be0c3E27F1D8	LR0252022	1500000 kshs	Thu Jan 27 2022	17:20:58 GMT+0300 (East Africa Time)
3	0.1 ha	Kabarnet	0xa53e4a65837b8Ab219945e244627be0c3E27F1D8	LR0012022	150000 kshs	Thu Jan 27 2022	17:28:29 GMT+0300 (East Africa Time)
4	0.1 ha	Kabarnet	0x56Ebb65bed04866Fc3d3ABA380DF40B50F089E64	LR0012022	150000 kshs	Thu Jan 27 2022	17:31:16 GMT+0300 (East Africa Time)
4	0.2 ha	Kabarak	0xa53e4a65837b8Ab219945e244627be0c3E27F1D8	92024	10000 kshs	Wed Feb 16 2022	13:52:09 GMT+0300 (East Africa Time)
6	0.2 ha	Kabarak	0x56Ebb65bed04866Fc3d3ABA380DF40B50F089E64	92024	10000 kshs	Wed Feb 16 2022	13:56:09 GMT+0300 (East Africa Time)
5	10.5 ha	Baringo County, Kabartonjo	0xa53e4a65837b8Ab219945e244627be0c3E27F1D8	BC030KBO578002	50000 kshs	Sat Feb 19 2022	11:21:36 GMT+0300 (East Africa Time)

Now that we have managed to have a seller execute a successful Land transaction in the designed model, it is equally important for the study to demonstrate how a member of the peer to peer distributed network may acquire the same piece of Land.

The model was designed in such a way that it could not allow the same Land seller to purchase the same piece of Land he/she has just put up for sale. For a Land buyer to acquire Land on sale, he/she is required to utilize their appropriate Metamask account as demonstrated in Figure 23.

Figure 23

The Land Buyer A/C in Metamask with 0.6886



Given that the Land buyer has enough ETH to meet the cost of transaction execution in the designed model, he/she may now proceed to initiate the process of acquiring a piece of Land from the listed properties for sale available in the system. This is demonstrated in the designed model as Figure 24 and source codes in Appendix 1(E):

Figure 24

The List and the Status of Land in the designed Model

#	Size	Location	Land Owner Address	Land Registry Number	Price	Transaction Date	Transaction Time	State
1	0.1 ha	Kambu	0xa53e4465837b8Ab219945e244627be0c3e27f1D8	A12232GH3424895	871546 kshs	Thu Jan 27 2022	12:11:34 GMT+0300 (East Africa Time)	Buy
2	0.25 ha	London	0xa53e4465837b8Ab219945e244627be0c3e27f1D8	LR0252022	1500000 kshs	Thu Jan 27 2022	17:20:58 GMT+0300 (East Africa Time)	Buy
3	0.1 ha	Kabarnet	0x56Ebb65be04866Fc3d3A8A38DDF40B50C089E64	LR0012022	150000 kshs	Thu Jan 27 2022	17:28:29 GMT+0300 (East Africa Time)	Sold
4	02 ha	Kabarak	0x56Ebb65be04866Fc3d3A8A38DDF40B50C089E64	lr2024	10000 kshs	Wed Feb 16 2022	13:52:09 GMT+0300 (East Africa Time)	Sold
5	10.5 ha	Baringo County, Kabartonjo	0xa53e4465837b8Ab219945e244627be0c3e27f1D8	BC030KBO578002	50000 kshs	Sat Feb 19 2022	11:21:36 GMT+0300 (East Africa Time)	Buy

To demonstrate how a Land buyer transaction is executed in the model, the study selected property #5, whose title deed registration number is **BC030KBO578002**. The status of the Land for sale is shown in Figure 25. Once a buyer selects the appropriate account in the Metamask, he/she goes ahead to identify the Land for purchase and then click the **Buy** button. Source codes displayed in Appendix 1(E).

Figure 25

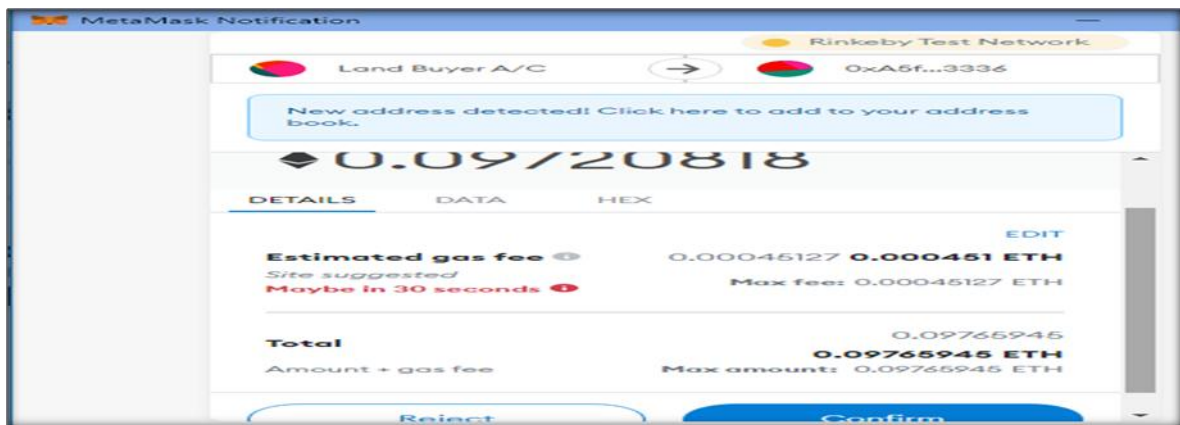
Status of Land with Title Deed number BC030KBO578002

#	Size	Location	Land Owner Address	Land Registry Number	Price	Transaction Date	Transaction Time	State
1	0.1 ha	Kambu	0xa53e4465837b8Ab219945e244627be0c3e27f1D8	A12232GH3424895	871546 kshs	Thu Jan 27 2022	12:11:34 GMT+0300 (East Africa Time)	Buy
2	0.25 ha	London	0xa53e4465837b8Ab219945e244627be0c3e27f1D8	LR0252022	1500000 kshs	Thu Jan 27 2022	17:20:58 GMT+0300 (East Africa Time)	Buy
3	0.1 ha	Kabarnet	0x56Ebb65be04866Fc3d3A8A38DDF40B50C089E64	LR0012022	150000 kshs	Thu Jan 27 2022	17:28:29 GMT+0300 (East Africa Time)	Sell
4	02 ha	Kabarak	0x56Ebb65be04866Fc3d3A8A38DDF40B50C089E64	lr2024	10000 kshs	Wed Feb 16 2022	13:52:09 GMT+0300 (East Africa Time)	Sell
5	10.5 ha	Baringo County, Kabartonjo	0xa53e4465837b8Ab219945e244627be0c3e27f1D8	BC030KBO578002	50000 kshs	Sat Feb 19 2022	11:21:36 GMT+0300 (East Africa Time)	Buy

Upon clicking the Buy button, the Land buyer is prompted by the Metamask to confirm the amount of ETH required as the cost for the execution of the smart contract in the distributed network. This is demonstrated in Figure 26 and Source Code in Appendix 1(E).

Figure 26

Cost of Land Transaction in Blockchain



The confirmation of the transaction cost triggers the smart contract execution in the model. That means the Land buyer has successfully executed the Land transaction in the Ethereum Blockchain and that the property has successfully changed hands. This can be confirmed by downloading the summary reports containing all transaction history of the Land transaction records including the transaction entry whose title deed registration number is **BC030KBO578002**. Figure 26 shows a summary report on transaction history of smart contracts of Land transactions executed in the model and source codes in Appendix 1(E).

From the above transaction history for title deed number **BC030KBO578002** in Figure 27, it worth noting that the Land owner address has changed from the initial Land seller, whose address was **0xa53e4A65B37bBAb219945e244627be0c3E27F1D8**, to the address of the buyer, which is indicated as

0x56Ebb65bed04B66Fc3d3A8A38DDF40B50f089E64 in the model. This is now the current owner and holder of the Land with title deed number **BC030KBO578002**.

Figure 27

Title Deed ownership transfer in the model, a Case of Title Deed number BC030KBO578002

#	Size	Location	Land Owner Address	Land Registry Number	Price	Transaction Date	Transaction Time
1	0.1 ha	Kiambu	0xa53e4A65B37bBAAb219945e244627be0c3E27F1D8	A12232GH3424895	871546 kshs	Thu Jan 27 2022	12:11:34 GMT+0300 (East Africa Time)
2	0.25 ha	London	0xa53e4A65B37bBAAb219945e244627be0c3E27F1D8	LR0252022	1500000 kshs	Thu Jan 27 2022	17:20:58 GMT+0300 (East Africa Time)
3	0.1 ha	Kabarnet	0xa53e4A65B37bBAAb219945e244627be0c3E27F1D8	LR0012022	150000 kshs	Thu Jan 27 2022	17:28:29 GMT+0300 (East Africa Time)
4	0.1 ha	Kabarnet	0x56Ebb65bed04B66Fc3d3A8A38DDF40B50f089E64	LR0012022	150000 kshs	Thu Jan 27 2022	17:31:16 GMT+0300 (East Africa Time)
4	02 ha	Kabarak	0xa53e4A65B37bBAAb219945e244627be0c3E27F1D8	lr2024	10000 kshs	Wed Feb 16 2022	13:52:09 GMT+0300 (East Africa Time)
6	02 ha	Kabarak	0x56Ebb65bed04B66Fc3d3A8A38DDF40B50f089E64	lr2024	10000 kshs	Wed Feb 16 2022	13:56:09 GMT+0300 (East Africa Time)
5	10.5 ha	Baringo County, Kabartonjo	0xa53e4A65B37bBAAb219945e244627be0c3E27F1D8	BC030KBO578002	50000 kshs	Sat Feb 19 2022	11:21:36 GMT+0300 (East Africa Time)
8	10.5 ha	Baringo County, Kabartonjo	0x56Ebb65bed04B66Fc3d3A8A38DDF40B50f089E64	BC030KBO578002	50000 kshs	Sat Feb 19 2022	15:11:56 GMT+0300 (East Africa Time)

4.6.2 Model Evaluation Metrics

In Chapter One of the study, the research specific objectives were presented. The last Specific objective was to evaluate the designed secure title registration model for Land transactions in Kenya based on Blockchain, with the view of recommending it for future implementation in Kenya. The previous sections of Chapter Four explained how the model was designed. In this section, the study presents how the model was evaluated. Evaluation refers to the assessment of the designed model to ascertain that its output provides the solution to challenges highlighted in the problem statement of the study (Singh, 2020).

The design outputs of the model presented in Chapter Four demonstrated how a Land buyer/seller executed a Land transactions that created a trustworthy platform whereby the buyer or the seller is the owner of land transaction records generated in the system. According to Manjunath (2019), Blockchain occurs in three stages of model development. The stages are: the pre-construction stage whereby data is collected, stored

and protected; second stage is the construction stage where the collected data is authenticated and secured against breach; third stage is the construction stage where all final changes are recorded in the Blockchain. Therefore, all the processes involved in the three stages culminate into a distributed ledger in the peer to peer network, with each block in the network having an identical copy of the ledger (Swan 2015).

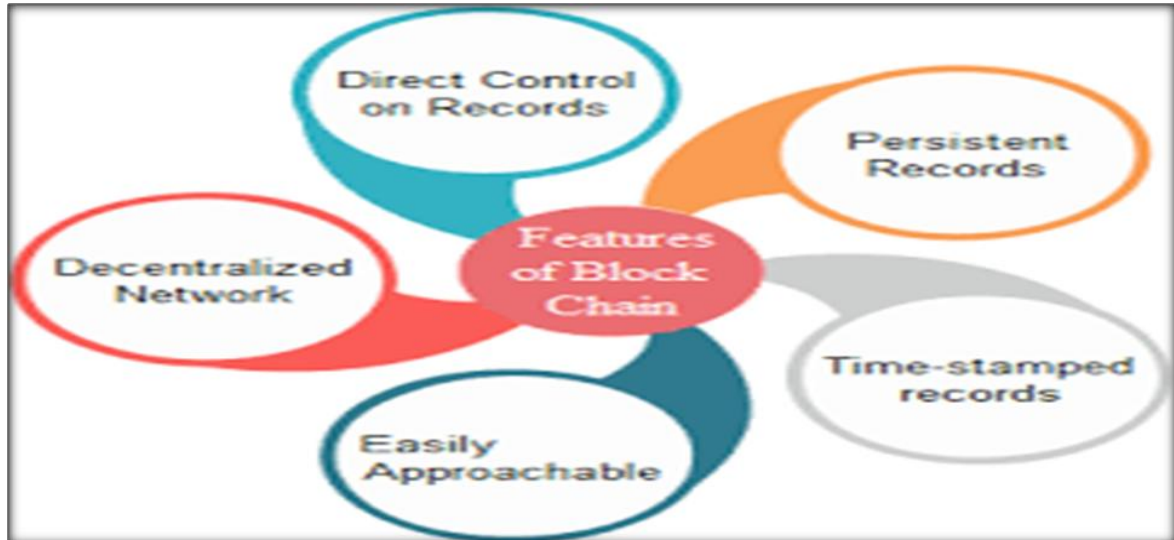
That provides an assurance that in the event of compromise or damage in the block, it can always be restored since blocks will always obtain complete copies of the database from the systems in the entire network. According to Underwood (2016), Blockchain has emerged to be a preferred potential technology for executing smart contracts due to the following critical features:

- i. **Persistence** – This is where transaction records are stored permanently as an immutable ledger containing history of transactions. No alteration or deletion of data, hence data or records cannot be corrupted.
- ii. **Time- Stamping of Records** - All transaction entries in the Blockchain are digitally recorded at the time of execution. This enhances the traceability of blocks.
- iii. **Transparency** – All transaction records are accessible to the network participants. They can therefore execute transaction entries based on consent granted. This enhances transparency.
- iv. **Decentralization** – This allows transactions to be executed directly by the users involved without the need of a third party. This enhances the integrity of the transactions executed as well as the integrity of the Blockchain system.

These features are summarized and presented in Figure 28:

Figure 28

Critical features of Blockchain Technology



To achieve its specific objectives and for the research questions to be fully addressed, specifically the third research question, the study adopted the above critical features of Blockchain technology as the basis that informed the evaluation metrics for the designed secure title deed registration model for Land transactions in Kenya based on Blockchain.

4.6.3 Evaluation Criteria

As discussed earlier in Chapter Three of this study, a criteria-based evaluation approach was adopted in the process of evaluating the output of the designed model in line with the research objectives of the study. The study applied a subjective sampling technique to randomly pick a group of 10 experts drawn from the Ministry of Lands in Kenya as users or evaluators of the designed model. These experts, almost on a daily basis interact with land documents relating to land transaction records or land administration documents in the Lands registry.

The study therefore engaged the group 10 experts by allowing them to interact with the designed model by registering into the system and executing land transactions as either a land buyer or a land seller. For the study to obtain a reliable user feedback on the designed model based on the set of predefined criteria outlined earlier in Chapter Three, a Likert scale tool was subjected to the users as they interact with the model.

The group of 10 experts sampled by the study to evaluate the designed model were drawn from the Ministry of Land in Kenya as follows:

- i. They were Four (4) Surveyors seconded from the Ministry of Land in Kenya to the Ministry of Roads, Housing and Urban Development – Department of Kenya Highways Authority, Nakuru County Office.
- ii. They were Three (3) Ministry of Lands officers – stationed in the Ministry of Lands Registry, Nakuru County Office.
- iii. They were Three (3) Ministry of Lands officers – stationed in the Ministry of Lands Registry, Nakuru County, and Rongai Sub-County Office.

The evaluation of the model was done by comparing its outputs against the expected results. As discussed earlier in this Chapter, the expected outputs of the designed model are entries of all land transaction records executed in the model by its users, who were either selling or buying land. Since the land transaction entries were executed in a Blockchain environment, the entries should contain all the critical features earlier listed in the study that formed the basis informing the evaluation metrics used by the study. These are listed as follows:

- i. Model Evaluation based on Persistence of Land Transaction Records – All land transaction entries should be indisputable.
- ii. Model Evaluation based on Time-Stamping of Land Transaction Records – All land transaction entries should contain the execution date and time clock.
- iii. Model Evaluation based on Transparency of Land Transaction Records – All land transaction entries should be available to the land buyers/sellers.
- iv. Model Evaluation based on Decentralization of Land Transaction Records – All land transaction entries should be executed by the land buyers/sellers without the help of a third party in the system.

4.6.2.1 Model Evaluation based on Persistence of Land Transaction Records

The designed model, as previously discussed in this Chapter allowed a Land buyer/seller to execute a Land transaction as a smart contract entry that was recorded as a block in the system. The model was therefore able to capture a Land transaction entry and recorded the same as permanent block of a Land transaction record in the Blockchain. Here, the study made a reference to a previous Land transaction executed in the designed model by a Land Seller with a registered title deed number *BC030KBO578002*. Initially the seller's address was captured as *0xa53e4A65B37bBAb219945e244627be0c3E27F1D8*. The new Land owner's address is *0x56Ebb65bed04B66Fc3d3A8A38DDF40B50f089E64*. This block of transaction is captured permanently in the model as an immutable ledger containing the history of Land transactions. Appendix 1(F) contains the component source codes.

Figure 29

Immutable Entries of Land Transactions Recorded in the Designed Model

#	Size	Location	Land Owner Address	Land Registry Number	Price	Transaction Date	Transaction Time
1	0.1 ha	Kiambu	0xa53e4A65B37bBAb219945e244627be0c3E27F1D8	A12232GH3424895	871546 kshs	Thu Jan 27 2022	12:11:34 GMT+0300 (East Africa Time)
2	0.25 ha	London	0xa53e4A65B37bBAb219945e244627be0c3E27F1D8	LR0252022	1500000 kshs	Thu Jan 27 2022	17:20:58 GMT+0300 (East Africa Time)
3	0.1 ha	Kabarnet	0xa53e4A65B37bBAb219945e244627be0c3E27F1D8	LR0012022	150000 kshs	Thu Jan 27 2022	17:28:29 GMT+0300 (East Africa Time)
4	0.1 ha	Kabarnet	0x56Ebb65bed04B66Fc3d3A8A38DDF40B50f089E64	LR0012022	150000 kshs	Thu Jan 27 2022	17:31:16 GMT+0300 (East Africa Time)
4	02 ha	Kabarak	0xa53e4A65B37bBAb219945e244627be0c3E27F1D8	lr2024	10000 kshs	Wed Feb 16 2022	13:52:09 GMT+0300 (East Africa Time)
6	02 ha	Kabarak	0x56Ebb65bed04B66Fc3d3A8A38DDF40B50f089E64	lr2024	10000 kshs	Wed Feb 16 2022	13:56:09 GMT+0300 (East Africa Time)
5	10.5 ha	Baringo County, Kabartonjo	0xa53e4A65B37bBAb219945e244627be0c3E27F1D8	BC030KBO578002	50000 kshs	Sat Feb 19 2022	11:21:36 GMT+0300 (East Africa Time)
8	10.5 ha	Baringo County, Kabartonjo	0x56Ebb65bed04B66Fc3d3A8A38DDF40B50f089E64	BC030KBO578002	50000 kshs	Sat Feb 19 2022	15:11:56 GMT+0300 (East Africa Time)

This demonstrated that the designed model was able to provide the required persistence of Land transactions. This would go a long way in addressing the challenges of the existing system, which is prone to deletion or alteration of Land transaction records in the Lands registry in Kenya.

4.6.2.2 Model Evaluation based on Time-Stamping of Land Transaction Records

All Blockchain transaction entries are digitally recorded and time-stamped at the time of execution. Therefore, a Land transaction executed by a buyer/seller in the designed model should easily be traced to the transaction date as well as its execution time. In the designed model, this was achieved as indicated in time-stamped entries of all Land transaction records in the system. For example, the land transaction entry #5 for a land registered under the title deed number **BC030KBO578002**, contains its execution date and the time clock as shown in Figure 30 and source codes in Appendix 1(F).

Figure 30

A time-stamped Land transaction record whose title deed number is BC030KBO578002 in the model

#	Size	Location	Land Owner Address	Land Registry Number	Price	Transaction Date	Transaction Time
1	0.1 ha	Kiambu	0xa53e4A65B37bBAb219945e244627be0c3E27F1D8	A12232GH3424895	871546 kshs	Thu Jan 27 2022	12:11:34 GMT+0300 (East Africa Time)
2	0.25 ha	London	0xa53e4A65B37bBAb219945e244627be0c3E27F1D8	LR0252022	1500000 kshs	Thu Jan 27 2022	17:20:58 GMT+0300 (East Africa Time)
3	0.1 ha	Kabarnet	0xa53e4A65B37bBAb219945e244627be0c3E27F1D8	LR0012022	150000 kshs	Thu Jan 27 2022	17:28:29 GMT+0300 (East Africa Time)
4	0.1 ha	Kabarnet	0x56Ebb65bed04B66Fc3d3A8A38DDF40B50f089E64	LR0012022	150000 kshs	Thu Jan 27 2022	17:31:16 GMT+0300 (East Africa Time)
4	02 ha	Kabarak	0xa53e4A65B37bBAb219945e244627be0c3E27F1D8	lr2024	10000 kshs	Wed Feb 16 2022	13:52:09 GMT+0300 (East Africa Time)
6	02 ha	Kabarak	0x56Ebb65bed04B66Fc3d3A8A38DDF40B50f089E64	lr2024	10000 kshs	Wed Feb 16 2022	13:56:09 GMT+0300 (East Africa Time)
5	10.5 ha	Baringo County, Kabartonjo	0xa53e4A65B37bBAb219945e244627be0c3E27F1D8	BC030KBO578002	50000 kshs	Sat Feb 19 2022	11:21:36 GMT+0300 (East Africa Time)
8	10.5 ha	Baringo County, Kabartonjo	0x56Ebb65bed04B66Fc3d3A8A38DDF40B50f089E64	BC030KBO578002	50000 kshs	Sat Feb 19 2022	15:11:56 GMT+0300 (East Africa Time)

4.6.2.3 Model Evaluation based on Transparency of Land Transaction Records

Blockchain generates a fully auditable and valid ledger of transactions accessible to all the network participants. As a result, the participants may engage in execution of transactions in their networks. This is the critical feature of Blockchain for achieving transparency. In the designed model, a Land seller was able register the Land details in the system.

This information was made available to other users in the network (in the case of the study, a Land buyer) who managed to go ahead and purchase the Land. This is demonstrated in Figure 31 and source codes in Appendix 1(F):

Figure 31

All users can see the Land whose title deed number is BC030KBO578002 as available for sell in the Model

#	Size	Location	Land Owner Address	Land Registry Number	Price	Transaction Date	Transaction Time	State
1	0.1 ha	Kimbu	0xa53e4a65b37b8Ab219945e244627be0c3E27F1D8	A12232GH3424895	871546 kshs	Thu Jan 27 2022	12:11:34 GMT+0300 (East Africa Time)	Buy
2	0.25 ha	London	0xa53e4a65b37b8Ab219945e244627be0c3E27F1D8	LR0252022	1500000 kshs	Thu Jan 27 2022	17:20:58 GMT+0300 (East Africa Time)	Buy
3	0.1 ha	Kabarnet	0x56Ebb65bed04866fc3d3A8A38D0F40850F089E64	LR0012022	150000 kshs	Thu Jan 27 2022	17:28:29 GMT+0300 (East Africa Time)	Sold
4	02 ha	Kabarak	0x56Ebb65bed04866fc3d3A8A38D0F40850F089E64	V2024	10000 kshs	Wed Feb 16 2022	13:52:09 GMT+0300 (East Africa Time)	Sold
5	10.5 ha	Baringo County, Kabartonjo	0xa53e4a65b37b8Ab219945e244627be0c3E27F1D8	BC030KBO578002	50000 kshs	Sat Feb 19 2022	11:21:36 GMT+0300 (East Africa Time)	Buy

On the peer to peer distributed blockchain network, the transaction record for that particular seller whose address is `0xa53e4A65B37bBAb219945e244627be0c3E27F1D8` and whose block **10194215** is available for every user to see in the entire network. This is shown in Figure 32 and source codes in Appendix 1(G):

Figure 32

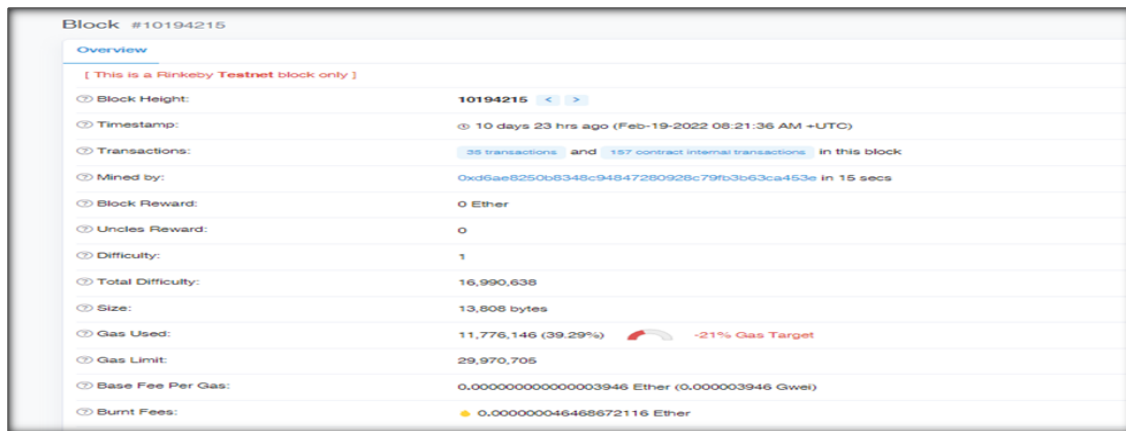
A Land sale transaction record executed in the model is available in the entire Blockchain

Txn Hash	Method	Block	Age	From	To	Value	Txn Fee
0xf0705e3cf376b13f130...	0xa91e647	10194215	10 days 23 hrs ago	0xa53e4a65b37bbab219...	OUT 0xa5f3e2d96e77999a8a...	0 Ether	0.000479806452

The Block details of that particular transaction as seen in the entire Blockchain distributed network is shown in Figure 33 and source codes in Appendix 1(G):

Figure 33

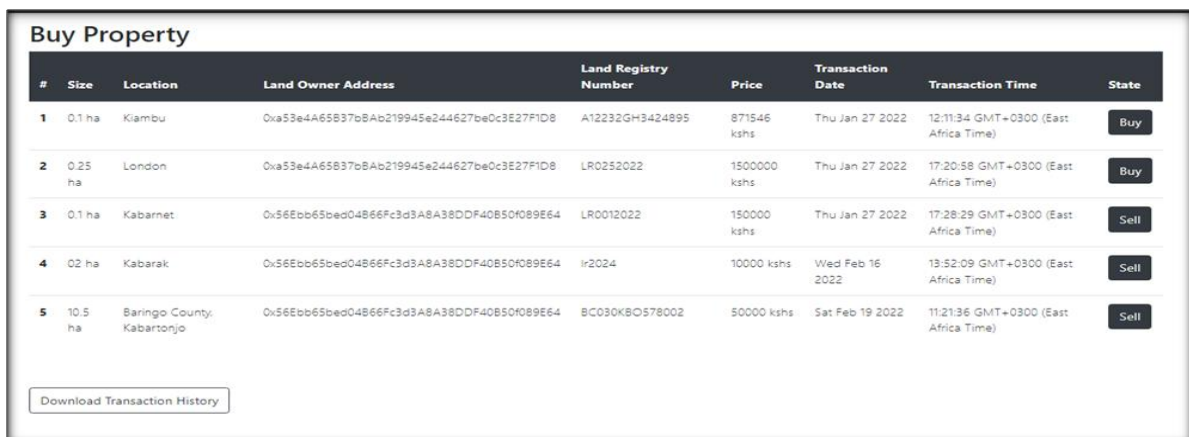
The Block transaction details from the model can be extracted from the entire peer to Peer Network



All the information was availed in the entire peer to peer network for everyone to see. Therefore, the designed model was able to achieve the much needed transparency in Land transactions. The Land buyer in the designed model buys Land and gets a new Land owner address assigned as **0x56Ebb65bed04B66Fc3d3A8A38DDF40B50f089E64**. This is shown in Figure 34 and source codes in Appendix 1(F). But the land title deed number **BC030KBO578002** remains unchanged/ immutable throughout the entire period of execution of the land transactions.

Figure 34

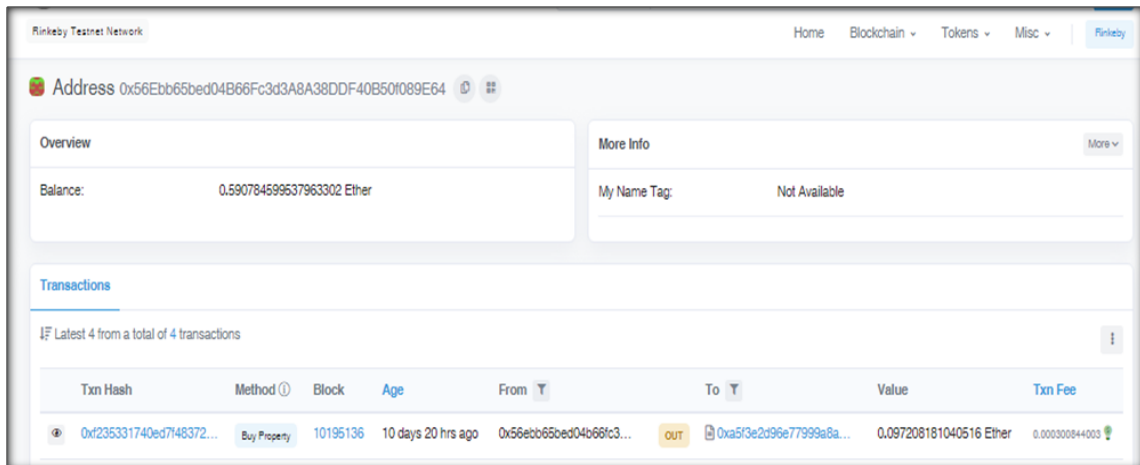
Land whose title deed number is BC030KBO578002 has been bought by a buyer in the Model



In the entire Blockchain distributed network, the transaction was recorded as block and assigned the block number **10195136** as shown in Figure 35 and source codes in Appendix 1(G):

Figure 35

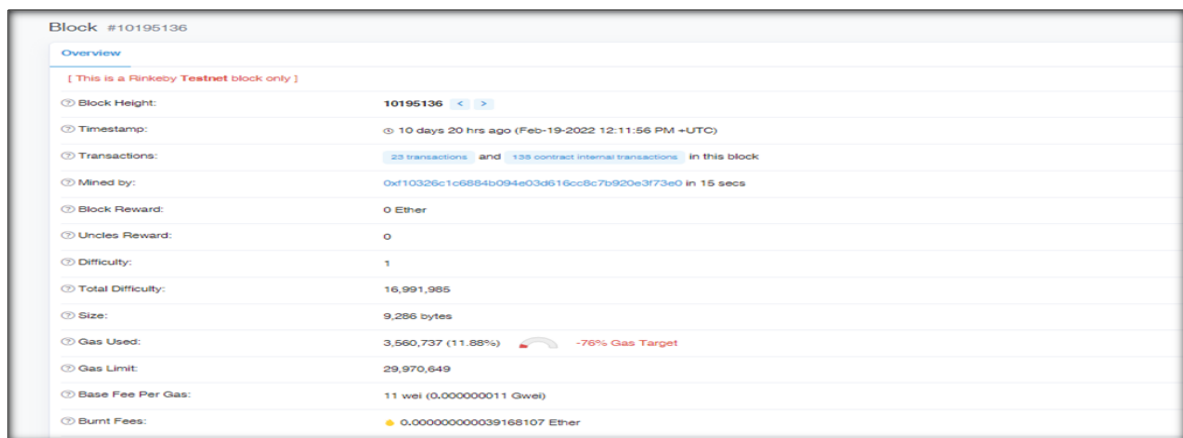
A Land Purchase Transaction Executed in the model is recorded in the entire Blockchain network



A successful purchase of Land by the Land buyer is recorded as a distributed ledger in the entire Blockchain network. The Land Buyer block was registered as **10195136**. This is shown in Figure 36 and source codes in Appendix 1(G):

Figure 36

The Block transaction details extracted from the entire peer to peer Network



From the discussions above, the study concludes that the designed model successfully executed Land transactions as smart contracts that generated auditable and valid ledger of land transaction records accessible to all participants in the Blockchain network. This can be used to enforcing non-repudiation in the system.

4.6.2.4 Model Evaluation based on Decentralization of Land Transaction Records

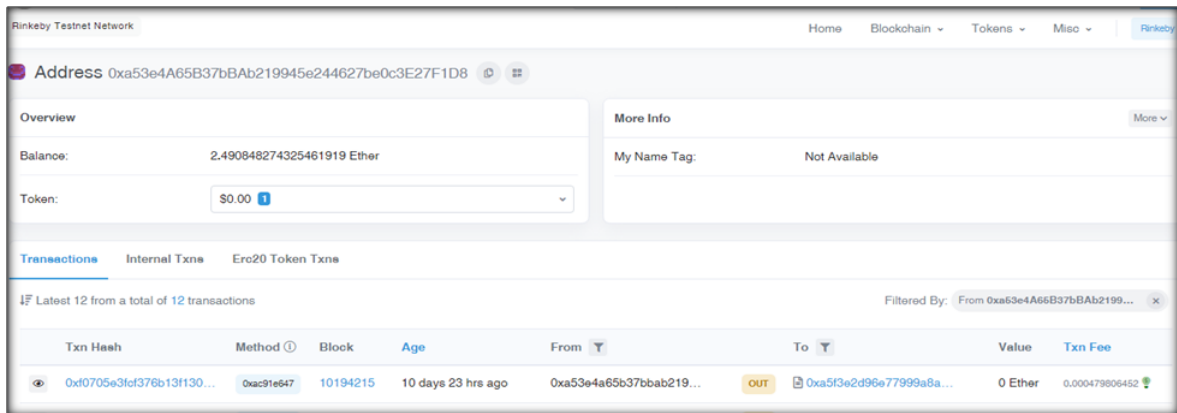
Amazon.com (2022), defines decentralization as the “transfer of control and decision-making from a centralized entity (individuals, organizations or groups) to a distributed network”. The objective of decentralization is to:

- i. To provide a trustless environment where no one trusts the other.
- ii. To optimize on resource distribution for consistency and better performance.
- iii. To minimize point of weakness in the system as a result of corruption
- iv. To improve on data reconciliation since all users have access to real-time shared view of records.

The designed secure title deed registration model for Land transaction in Kenya based on Blockchain was evaluated based on this critical feature. From the previous discussions on the model evaluation in this Chapter Four, Land transactions were executed successfully and independently by the users (Land Buyer/Seller) without the involvement of a third party. All transactions executed in model were distributed as a ledger in the entire Blockchain. Therefore, the designed model was able to demonstrate this critical feature of Blockchain.

Figure 37

A Distributed Ledger of a Land sale transaction record in the entire Blockchain. Component source codes in Appendix 1(G)



4.6.2.5 Best Case Scenario

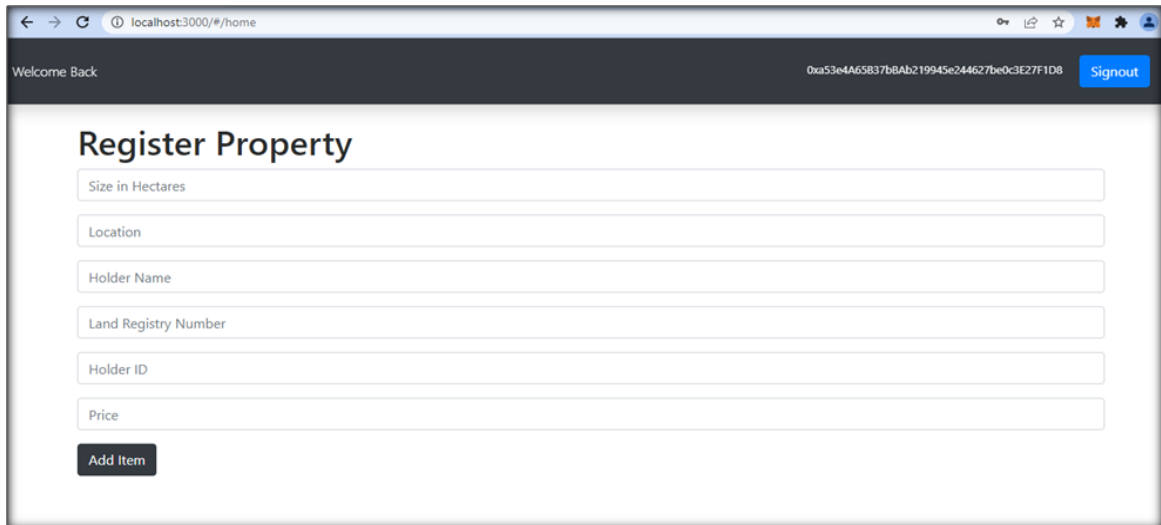
The designed model in the study allows a Land buyer/seller to register a Title Deed for the purpose of a land transaction. The buyer/seller is required to complete the registration process by providing the following details:

- i. The availability of the size of Land to be transacted expressed in form of Hectares.
- ii. The Physical Location of the Land to be transacted.
- iii. The Full Name of the individual who owns the Land.
- iv. The Title Deed Number, which is referred to as the Land Registry Number.
- v. The National Identity Card number of the individual under whose land is registered.
- vi. The value of the land expressed as its Price in Kenyan Currency.

This is demonstrated in the registration form shown in Figure 38 and source code in Appendix 1 (D):

Figure 38

A Title Deed registration form for a Land Transaction

The image shows a web browser window with the address bar displaying 'localhost:3000/#/home'. The page content includes a header with 'Welcome Back' on the left and a user ID '0xa53e4a65837b8ab219945e244627be0c3e27f1d8' and a 'Signout' button on the right. The main heading is 'Register Property'. Below this, there are six input fields: 'Size in Hectares', 'Location', 'Holder Name', 'Land Registry Number', 'Holder ID', and 'Price'. At the bottom left of the form area is a dark 'Add Item' button.

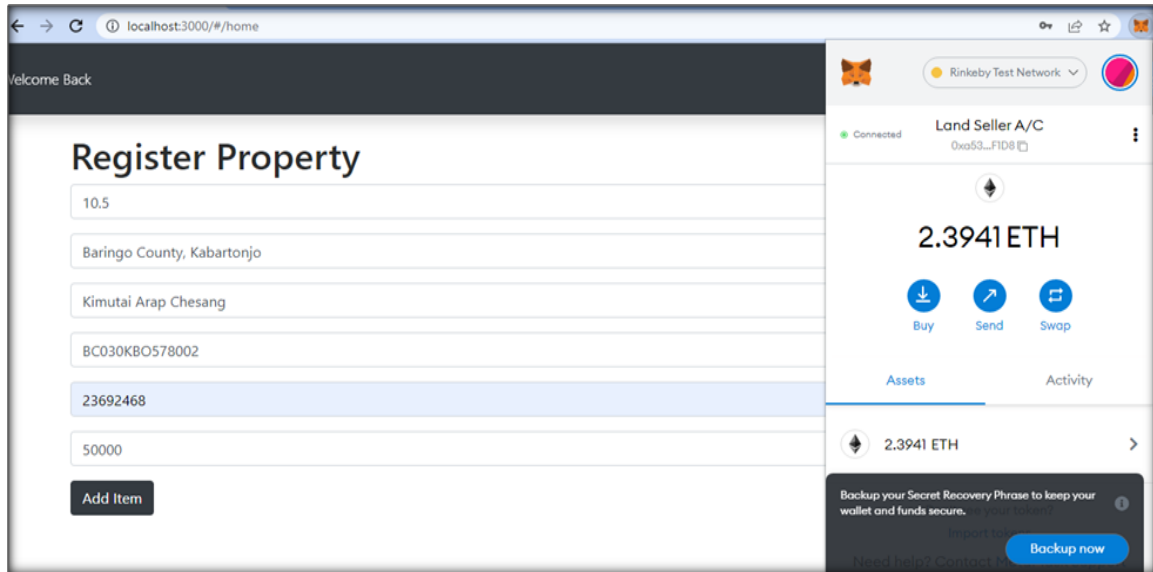
The title deed registration process requires the land buyer/seller to submit the land registry number in an alphanumeric format as follows: **BC030KBO578002**. For the purpose of the study, the above title deed registration number can be interpreted in the following order of sequencing:

- i. **BC** – This stands for Baringo County whose county code in Kenya is identified as code **030**. This is the County where the land is physically located.
- ii. **KBO** – This stands for Kabartonjo. It's the township within the County where the land is physically located.
- iii. **578002** – This is the actual land allotment number assigned to the particular parcel of land.

Figure 39 shows a filled up registration form in the designed model with the Land title deed registration Number **BC030KBO578002**. The source codes are available in Appendix 1(D):

Figure 39

Entry of the title deed registration number BC030KBO578002 as Captured the Model



Once the title deed registration number has been captured in the designed model during the title deed registration process, the user involved in the land transaction may proceed to **add item** into the system either as a land to be bought or land to be sold. As demonstrated in Figure 38 above, the title deed registration number **BC030KBO578002** shall remain throughout the entire lifetime of all land transaction entries executed in the model, as the only valid and authentic Title Deed registration number for that particular piece of land and can be referenced at any given time. The designed model rejects a new land transaction entry whose title deed registration number has already been captured in the system. A user executing such a transaction shall automatically be prompted of the same by the model.

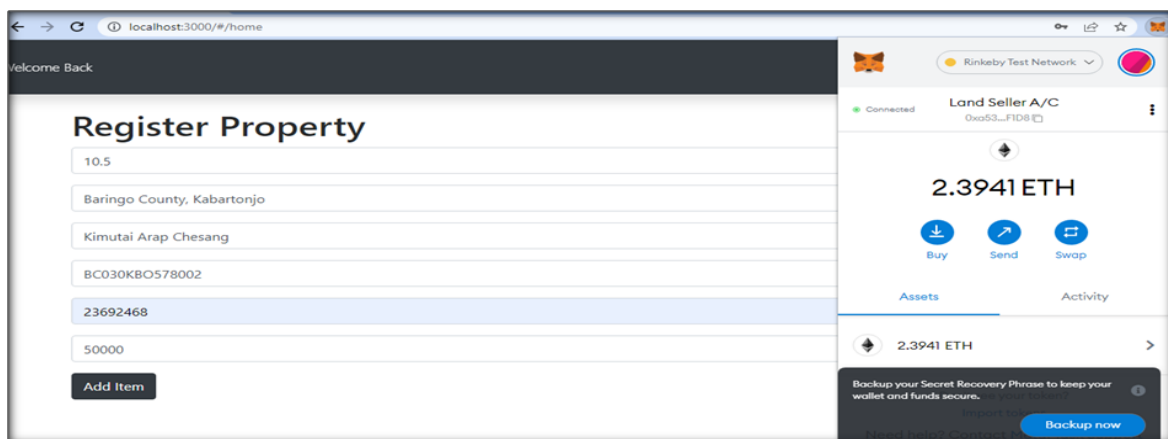
4.6.2.6 Worst Case Scenario

An individual buying or selling land is required to register the land title deed once. This involves the submission of the individual's profile as well as the land information into the registration form as demonstrated in Figure 40. The next step is to add a land record into the database of the model. Once this is done, then it means that no other land

transaction record having an exact Land registration number for example **BC030KBO578002** can be added into the Land Registry. However, in the case of a title deed registration number obtained fraudulently, the designed model does not have the authenticity to detect such kind of land transactions since the functionality of authenticating the transactions is left to the NLIMS/Ardhisasa platform that houses the live Lands registry in Kenya. The source codes are contained in Appendix 1(D):

Figure 40

Land Transaction Authentication in the Model



4.7 Results and Discussions

Muriuki (2019) highlighted six stages involved in land registrations and transactions that a potential land buyer/seller must follow as a due diligence process for title deed acquisition towards land ownership in Kenya. The first stage is the preliminary land identification and informal due diligence; second stage is to perform land search in the Land's Registry; third stage is to sign up land sale agreement between parties involved in the transaction; the fourth stage is the transfer of the land documents; fifth is the stamp transfer and lastly, the sixth stage involves the registration of the land title transfer. The entire process is extremely tedious and time consuming. It gets even more complicated since different individuals are involved at every stage alongside service charges incurred at every stage of the process.

In order to achieve its specific objectives, the study went ahead to establish and adopt criteria for evaluating the designed model, previously highlighted in parts of Chapters Three and Four. This was based on the following four key areas of model evaluation in Appendix 2:

- i. Persistence
- ii. Time-stamping
- iii. Transparency
- iv. Decentralization

The above four key areas of model evaluation were each assigned an evaluation key as shown in the summarized Table 2.

Table 2

Assignment of the Model's evaluation Key Criteria

Four Key Areas of Model Evaluation	Evaluation Key
Persistence	1
Time-stamping	2
Transparency	3
Decentralization	4

The evaluation keys shown in Table 2 above were adopted by the study for the purpose of identifying the key areas of focus during the model evaluation. Each area is assigned a key which does not necessarily represent the weights carried by each of the fourkey areas of focus for evaluation.

Based on the above criteria, the study made an assessment of the current status of land registration and land transaction processes presently employed at the Lands registry in Kenya .The study observed that land registration and transaction processes were largely

conducted manually. Based on the four key areas of model evaluation, a summary of the current status in the lands registry is provided in Table 3.

Table 3

Assessment of the current status of land registration and land transaction processes in the lands registry in Kenya against the Four Key Areas of the model evaluation used in the study

Title Registration	Land Transactions
1	X
2	X
3	X
4	X

Based on the assessment criteria in Table 3 above, the study observed that the current land registration and Land transaction processes employed by the lands registry in Kenya is extremely vulnerable to exploitation. The study highlighted most of the vulnerabilities in Chapters One and Two.

The largely manual nature of operations in the lands registry does not guarantee the persistence, time-stamping, transparency and the decentralization of land transactions compared to the designed secure title deed registration model for land transactions in Kenya based on Blockchain.

The users' feedback obtained from the autonomous evaluation of the designed model based on the Four Key areas of model evaluation adopted by the study were captured using a Likert scale tool. The summary of the same is presented in Table 4 below:

Table 4

A Summary of the Users' autonomous evaluation of the secure title deed registration model for land transactions in Kenya based on Blockchain against the Four Key areas of model evaluation criteria adopted by the study

User	Four Key Areas of Model Evaluation	Assessment Key	Feedback
Surveyors	Persistence	1	The Model demonstrated persistence in that Land transaction records were stored permanently in the form of immutable ledger containing all the history of land registrations and land transactions ever executed in it.
Land's Registry Officers	Time-stamping	2	The Model confirmed that all land registration and transaction entries were digitally recorded, dated and time stamped during execution, meaning all the transaction blocks are traceable.
Surveyors	Transparency	3	The Model revealed that all land registration and transaction records were accessible to the entire network participants, thus guarantee that all entries are transparent to all land buyers and sellers involved.
Land's Registry Officers	Decentralization	4	The Model proved that land registration and transactions processes can be executed directly by the users involved without the need of a third party. This feature enhances the integrity of land transactions in the entire peer to peer distributed network.

Table 4 above provides a summary of the reflections of the selected users upon evaluating the secure title deed registration model for land transactions in Kenya based on Blockchain. From the users' feedback, the study concludes that the specific objectives and the research questions in Chapter One were sufficiently addressed by the designed model.

CHAPTER FIVE

SUMMARY, CONCLUSIONS AND RECOMMENDATION

5.1 Introduction

Chapter Five presents a summary of the research. The study considered the research questions as the basis for drawing the summary and conclusions. The chapter also presents a brief discussion on its recommendations based on the outcomes of the study, the areas to be considered for future research as well as the contributions of the research to the body of knowledge.

5.2 Conclusions

The overall objective of the study was to design and evaluate a secure title deed registration model for Land transactions in Kenya based on Blockchain. Therefore the study drew the conclusions based on the overall objective by considering the research questions that were highlighted in Chapter One as follows:

- i. What challenges exist in Land transactions for Land ownership in Kenya?
- ii. How can a secure title deed registration model for Land transactions in Kenya based on Blockchain be designed?
- iii. How can a secure title deed registration model for land transactions in Kenya based on Blockchain be demonstrated?
- iv. What requirements are needed for a successful evaluation of the designed model as well as the expected outcomes to inform the model's recommendation for future implementation in Kenya?

5.2.1 Challenges of Land registrations and Land transactions in Kenya

To address the first research question on the challenges ass, the study reviewed various literatures with the objective of understanding the root cause of the persistent challenges relating to Land registrations and Land transactions in Kenya. The research established that the Lands registry in Kenya is largely managed manually. The research also established that the National Land Information Management System, NLIMS/Ardhisasa platform is not fully implemented.

The process of digitizing the Lands registry is a slow process that may take a little longer to come into its full implementation while some other issues relating to Land disputes date back to pre –colonial era in Kenya. Much of the Land in Kenya has no title deed and therefore any Land registrations and Land transactions carried out on such Land is not legally binding to the parties involved. In view of the challenges, the study designed a secure title deed registration model for Land transactions in Kenya based on Blockchain as a solution towards addressing the menace in the Lands registry in Kenya.

5.2.2 Requirements for the Model Development

To address the second research question on the model development requirements, the study adopted the Design Science and the Proof of Concept research methodologies as discussed in Chapter Three. Chapter Four of the study presented the model design and evaluation criteria. Several software requirements for the design of the model were presented. Functional and non-functional requirements for the model were highlighted as well.

Both the software and the user requirements for the model development provided a clear roadmap on which a secure title deed registration model for land transaction in Kenya based on Blockchain was successfully designed for evaluation.

5.2.3 Model Demonstration and Evaluation

The designed secure title deed registration model for Land transactions in Kenya based on Blockchain was evaluated. Evaluation was done as the formal process of ascertaining whether the overall objective of the study was achieved. Users evaluated the model against the four critical features of Blockchain that were adopted by the study. The critical features were: Persistence, Time-stamping, Transparency and Decentralization of land registration and transaction records. Based on the evaluation criteria set out by the study, the evaluation process was successful and the results were presented.

The outcomes demonstrated that the study was able to successfully design and evaluate a Secure Title Deed Registration Model for Land Transactions in Kenya Based on Blockchain that integrated all the critical features of Blockchain technology.

5.3 Recommendations

The research recommends that it is feasible to integrate the NLIMS/Ardhisasa platform into a model like the one designed and evaluated in the study. The Secure Title Deed Registration Model for Land Transactions in Kenya Based on Blockchain demonstrated that the land registration and the land transaction processes can securely be executed as smart contracts in Blockchain. The user feedback based on the evaluation of the designed model in the study confirmed that land registration and transaction records were persistent, time-stamped, transparent and decentralized, providing an authentic platform through which the proof of land ownership can be based. This could go a long way in addressing the menace presently associated with the operations at the lands registry in Kenya.

The benefits that come along with the integration of Blockchain technology in the NLIMS/Ardhisasa platform for the sole purpose of Land administration and management

in Kenya is enormous, perennial and worth investing. The study therefore recommends that the Government of Kenya, through the Ministry of Lands adopts the model as a solution towards the Land inequalities in Kenya.

5.3.1 Policy Recommendations

The research recommends that the Government of Kenya, through Parliament and its relevant agencies like the Ministry of Lands, National Lands Commission, National Treasury and the Ministry of ICT develops a Cybersecurity and Crypto currency Policy Framework towards the adoption of Blockchain technology in Land administration and management in Kenya. The policy should address emerging issues such as the adoption of Crypto currency as a legal tender by the Central Bank of Kenya to facilitate Land registrations and transactions executed in Blockchain. This would go a long way in the establishment of a National Cybersecurity and Critical Infrastructure Protection Agency charged with the responsibility of promoting collaborations and partnerships in both public and private sector on matters relating to perceived or real Cyberspace and Physical threats to Critical Infrastructure in Kenya.

The study illustrates best the contributions made by the research as a consideration of the specific objectives that were highlighted in Chapter One as follows:

- i. To review the challenges of Land transactions for Land ownership in Kenya.
- ii. To design a secure title deed registration model for land transactions in Kenya Based on Blockchain.
- iii. To demonstrate how the designed model integrates Blockchain to provide a secure platform for title deed registration for land transactions.
- iv. To evaluate the output of the designed model against the evaluation metrics set out by the study with the view of recommending it for future implementation in Kenya.

The literature review conducted by the study is presented in Chapter Two. The outcome of the research demonstrates that numerous challenges presently exist in the Lands registry in Kenya. Gaps were identified in the existing land registration and land transactions processes towards land ownership in Kenya. In The same Chapter, the study identified Blockchain as the much needed technological solutions for addressing the existing gaps in the Lands registry. As a result, the study established that Blockchain technology offers a better solution in the administration and management of land in Kenya. Future research in the field may find it useful to incorporate the knowledge in the literature review to inform the scope of their research interests.

The model design and evaluation was presented in Chapter Four of the study. User requirements and system requirements were discussed. The study identified the functional and non-functional requirements for the model. From the designed model, the study demonstrated that it was feasible to design a secure title deed registration model for land transactions based on Blockchain. The model provided a framework on which similar models that integrate Blockchain technology can be built upon.

The user feedback based on the evaluation of the designed model in the study confirmed that land registration and transaction records were persistent, time-stamped, transparent and decentralized, providing an authentic platform through which the proof of land ownership can be based. This could go a long way in addressing the menace presently associated with the operations at the lands registry in Kenya.

5.4 Recommendations for Future Research

In Chapter Four, the study highlighted both the expected functional and non-functional requirements of the model. The non-functional requirement of the model was intended to integrate NLIMS/Ardhisasa Platform for purposes of authenticating the title deeds held

in the Lands registry as pointed out in the conceptual framework, a requirement that the research never accomplished due to financial, time and privacy constraints that were experienced during the entire research period. The NLIMS/Ardhisasa platform could not be integrated due to the sensitivity and confidentiality nature of land records in the Lands registry. This is an area that future research may strive to achieve.

The study did not incorporate the use of Google Maps location based approach due to time constraints. It is therefore a recommendation of this study that future research may consider the integration of Google Maps in model to assist potential Land buyers / sellers to virtually view the land location and its coordinates in the Map of Kenya as well its environs.

The research adopted a Criteria-based model evaluation approach using Four Key areas of model evaluation that formed the basis of the evaluation metrics for the designed model. To enhance innovations in the industry and advance economic prosperity in developing countries like Kenya, the study recommends that future research in the field may seek to adopt an alternative approach in the design and evaluation of models that integrate Blockchain technology.

The study recommends that future research may also focus on the prospect of integrating Blockchain with Land taxation systems as well as the adoption of crypto currency as a legal tender to facilitate the execution of smart contracts for properties in Kenya.

REFERENCES

- Aas, K. F. (2014). Bordered penalty: Precarious membership and abnormal justice. *Punishment & Society*, 16(5), 520-541.
- Almgren, H. and Stengard, M. (2015). "How to maintain authenticity and integrity of electronic information without utilizing electronic certificates?", INFUTURE2015: e-Institutions – Openness, Accessibility, and Preservation, Zagreb, 11-13 November.
- Antonopoulos, A.M. (2015). *Mastering Bitcoin: Unlocking Digital Cryptocurrencies*. 1st Edition, O'Reilly Media: Sebastopol.
- Ardhisasa. (2021). Retrieved 4 October 2021, from <https://ardhisasa.Lands.go.ke/home> ARMA International (2013), Generally-Accepted Recordkeeping Principles, available at: www.arna.org/docs/bookstore/.
- Australian Stock Exchange (2016). ASX Selects Digital Assets Holdings.
- Balakrishnan, D. (2021). *A Brief Introduction to Hydrachain*, a Popular Distributed Ledger System. Retrieved 8 January 2021, From <https://www.opensourceforu.com/2020/08/a-brief-introduction-to-hydrachain-a-popular-distributed-ledger-system>.
- Beck, T.H.L, Demirgüç-Kunt, A., Maksimovic, V., (2004). Bank competition and access to finance. *Journal of Money, Credit and Banking*.
- Berlin. Hofmann, E., Kotzab, H., (2010). A supply chain-oriented approach of working capital management. *J Bus Logistics* 31(2),305–330
- Bertino, E. and Sandhu, R. (2005). Database Security: Concepts, Approaches, and Challenges. In: *IEEE Transactions on dependable and secure computing*, 1(1), 2-19.
- Bitcoin.org (2015). "Developer guide", available at: <https://bitcoin.org/en/developer-guide>
- Bitcoinwiki (2015a). "Block hashing algorithm", available at: https://en.bitcoin.it/wiki/Block_hashing_algorithm
- Bitcoinwiki (2015b). "Weaknesses". Blockchain references.
- Blockchain. (2021). Retrieved 15 December 2021, from <https://rinkeby.etherscan.io>
- Boulding, K. E. (1956). General systems theory—the skeleton of science. *Management science*, 2(3), 197-208.
- Bowersox, D.J., Closs, D.J., Stank, T.P., (2003). How to master cross-enterprise collaboration. *Supply Chain Management Review* 7 (4), 18–27.
- Burgess, K. (2015). The Promise of Bitcoin and the Blockchain Bretton Woods 2015 Working Paper, Consumers Research.
- Cao, M. and Qingyu, Z. (2010). Supply chain collaboration: Impact on collaborative advantage and firm performance. *Journal of Operations Management* 29(3), 163–180.
- Carter, R.C., Rogers, D.S. and Choi, T.Y., (2015). Toward the Theory of the Supply Chain. *Journal of Supply Chain Management*, 51(2), 2015.

- Caukwell, R. A. (1977). *Cadastral Survey in Kenya and its role in the development of the Country*. Master of Science thesis, Department of Surveying and Photogrammetry, University of Nairobi, Kenya
- CCSDS (2012). “*Reference Model for an Open Archival Information System (OAIS): Recommended Practice Issue 2*”, Consultative Committee for Space Data Systems, .
- Chavez-Dreyfuss, G. (2015). “*Honduras to build Land title registry using bitcoin technology*”, Reuters.
- Christidis K, Devetsikiotis M (2016.). *Blockchains and smart contracts for the internet of things*. IEEE Access 4:2292–2303
- Cohen, B. (2015). “*Vermont considering Blockchain tech for state records, smart contracts*”, The Coin Telegraph.
- Coindesk (2015). Who is Satoshi Nakamoto?.
- Collier, R. (2009). Proof-Of-Concept Projects Proving Productive. *Canadian Medical Association Journal*, 181(12), E278-E278. Doi: 10.1503/Cmaj.109-3092
- Constitution of Kenya 2010. (2020). Retrieved 12 December 2020, from <https://statelaw.go.ke/>
- Croman K, Decker C, Eyal I, Gencer AE, Juels A, Kosba AE, Miller A, Saxena P, Shi E, Siler EG, Song D, Wattenhofer R (2016). *On scaling decentralized Blockchains - (a position paper)*. In: Financial cryptography and data security - FC 2016 international workshops, BITCOIN, VOTING, and WAHC, pp 106–125
- Cronholm, S., & Goldkuhl, G. (2003). Strategies for information systems evaluation-six generic types. *Electronic Journal of Information Systems Evaluation*, 6(2), 65-74.
- Culubas (2011). Timejacking & Bitcoin.
- Dale, P. and McLaughlin, J. (1988). *Land Information Management*. Clarendon Press. The programs in geomatics at Laval University. Canadian Institute of Education.
- De Boer, R., van Bergen, M., and Steeman, M. A. (2015). Supply Chain Finance, its Practical Relevance and Strategic Value. *The Supply Chain Finance Essential Knowledge Series*. Zwolle.
- DLM Forum Foundation (2010). *Modular Requirements for Records Systems*. Publications Office of the European Union, Brussels.
- Don & Alex Tapscott, authors Blockchain Revolution (2016)
- Dorri A, Kanhere SS, Jurdak R (2016). *Blockchain in internet of things: challenges and solutions*.
- Due Diligence Procedures in Land Transactions in Kenya. (2018). Retrieved 4 August 2018, from <https://www.Landcommission.go.ke>.
- Duranti, L. (1990), “Diplomatics: new uses for an old science (Part 111)”, *Archivaria*, 30, 4-20.
- Duranti, L. (1997), “The archival bond”, *Archives and Museum Informatics*, 11, 213-218.
- Duranti, L. (2005). *The Long-Term Preservation of Authentic Electronic Records: Findings of the InterPARES Project*, Archilab, San Miniato.

- Duranti, L. and Rogers, C. (2012). "Trust in digital records: An increasingly cloudy legal area", *Computer Law & Security Report* 28(5)522-531.
- Eskandari, S., Barrera, D., Stobert, E. and Clark, J. (2015). "A first look at the usability of bitcoin key management", USEC 2015, San Diego, CA.
- Exploring the Types of Titles. (2020). Retrieved 25 February 2020, from <https://www.investopedia.com/terms/t/title.asp>
- Fairchild, A., (2005). Intelligent matching: integrating efficiencies in the financial supply chain. *Supply Chain Management*, 10(3/4): 244-249.
- Faqir-Rhazoui, Y., Ariza-Garzón, M. J., Arroyo, J., & Hassan, S. (2021, May). *Effect of the gas price surges on user activity in the daos of the ethereum blockchain*. In Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems (pp. 1-7).
- Feder, G., & Nishio, A. (1998). The benefits of Land registration and titling: economic and social perspectives. *Land Use Policy*, 15(1), 25-43.
- Fellenz, M.R., Augustenborg, C., Brady, M., Greene, J.,(2009). Requirements for an Evolving Model of Supply Chain Finance: A Technology and Service Providers Perspective. *Communications of the IBIMA. Volume 10*: 1943-7765.
- Findlay, C. (2015), "Decentralised and inviolate: the Blockchain and its uses for digital archives", Recordkeeping Roundtable.
- Fleming, J. (2018). Recognizing and resolving the challenges of being an insider researcher in work-integrated learning [Special Issue]. *International Journal of Work-Integrated Learning*, 19(3),311-320.
- Forte P, Romano D, Schmid G (2015). Beyond bitcoin - part I: a critical look at Blockchain-based systems. IACR Cryptology ePrint Archive 2015:1164
- Forte P, Romano D, Schmid G (2016). *Beyond bitcoin - part II*: Blockchain-based systems without mining. IACR Cryptology ePrint Archive 2016:747
- Franco, E. (2015). "Inside the Chinese Bitcoin Mine That's Grossing \$1.5M a Month", Motherboard, available at: <http://motherboard.vice.com/read/chinas-biggest-secret-bitcoin-mine> (accessed 3 December 2017).
- Frankel, R., Bolumole, A.Y., Eltantawy R.A., Paulraj A., Gundlach G., (2008.) The Domain and scope of SCM's foundational disciplines. *Insights and Issues to Advance Research*. 29(1),1–30.
- Garay JA (2015). Blockchain-based consensus (keynote). In: *19th international Conference on Principles of Distributed Systems*, OPODIS 2015, pp 5:1–5:1
- Gaston J. R. (1979) *Land Issue in Kenya Politics*. *Ufahamu*, Vol. 9, No. 2, pp. 30 – 58.
- Gelsomino, L. U. C. A., Mangiaracina, R., Perego, A., & Tumino, A. (2016). Supply chain finance: modelling a dynamic discounting programme. *Journal of Advanced Management Science*, 4(4), 283-291.
- Gervais, A., Karame, G. O., Wüst, K., Glykantzis, V., Ritzdorf, H., & Capkun, S. (2016, October). On the security and performance of proof of work blockchains. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security* (pp. 3-16).

- Giaretta, D. (2011). *Advanced digital preservation (pp. I-XXII)*. Berlin: Springer.
- Giarretta, D., Matthews, B., Bicarregui, L. and Guercio, M. (2009). "Significant properties, authenticity, provenance, representation information and OAIIS", iPres 2009: *The Sixth International Conference on Preservation of Digital Objects*, San Francisco, CA, pp. 67-73, available at: http://escholarship.org/uc/cdl_ipres09
- Goldsack, J. C., Dowling, A. V., Samuelson, D., Patrick-Lake, B., & Clay, I. (2021). Evaluation, acceptance, and qualification of digital measures: from proof of concept to endpoint. *Digital Biomarkers*, 5(1), 53-64.
- Government of Canada Standing Senate Committee on Banking, Trade and Finance (2015), "Digital currency: you can't flip this coin.
- Government of Honduras (2004). "Ley de Propiedad (Property Law).
- Griggs, L. (2016). *The doctrinal coherence of the Torrens system of Land registration in Australia: evolution or revolution?*
- Guadamuz A, Marsden C (2015). *Blockchains and bitcoin: regulatory responses to cryptocurrencies*. First Monday 20(12)
- Gupta S, Dutta K(2011). *Modeling financial supply chain*. Eur J Oper Res 211(1):47–56
- H.W.O. Okoth-Ogendo (1991). *Tenants of the Crown: Evolution of Agrarian Law & Institutions in Kenya*.
- Hamilton, M. (2020). Blockchain Distributed Ledger Technology: An Introduction And Focus On Smart Contracts. *Journal of Corporate Accounting & Finance*, 31(2), 7-12. Doi: 10.1002/Jcaf.22421
- Hanson, R.T., Staples, M. (2017). *Distributed Ledgers, Scenarios for the Australian economy over the coming decades*. Canberra. Commonwealth Scientific and Industrial Research Organisation.
- Hanstad T. (1998), "Designing Land Registration Systems for Developing Countries," *American University International Law Review* 13, No. 3
- Herrera-Joancomart J, Lupu E, Posegga J, Aldini A, Martinelli F, (2015). *Data Privacy Management, Autonomous Spontaneous Security, and Security Assurance*. vol. 8872 of Lecture Notes in Computer Science. Springer International Publishing; 2015. p. 3–16. Available from: http://dx.doi.org/10.1007/978-3-319-17016-9_1.
- Hofmann, E. (2005). Supply chain finance: some conceptual insights. *Beiträge Zu Beschaffung Und Logistik*, 16(1), 203-214.
- Hofmann, E. (2011). *Supply chain finance solutions*. Springer.
- Home, R. (2021). History and Prospects for African Land Governance: Institutions, Technology and 'Land Rights for All'. *Land*, 10(3), 292.
- Hyvärinen, H., Risius, M., & Friis, G. (2017). A Blockchain-Based Approach Towards Overcoming Financial Fraud in Public Sector Services. *Business & Information Systems Engineering*, 59(6), 441-456.
- Institution of Surveyors of Kenya (2000). *Land Reforms in Kenya The Institution of Surveyors of Kenya perspective*.

- International Council on Archives (2007). International Standard for Describing Functions, 1st ed., Committee on Best Practices and Standards, Dresden, 2-4 May, *International Council on Archives*, Paris.
- InterPARES (2015). Terminology database.
- ISO/IEC (2001). ISO 15489-1:2001 – *Information and Documentation – Records Management –Part I: General*, ISO, Geneva.
- ISO/IEC (2012a). ISO 14721: 2012– *Space Data and Information Transfer Systems – Open Archival Information System (OAIS) – Reference Model*, ISO, Geneva.
- ISO/IEC (2012b), ISO 16363: 2012– *Space Data and Information Transfer Systems – Audit and Certification of Trustworthy Digital Repositories*, ISO, Geneva.
- ISO/IEC (2013), ISO 27001: 2013 – *Information Security Management*, ISO, Geneva.
- Jiao, J., Kan, S., Lin, S. W., Sanan, D., Liu, Y., & Sun, J. (2020, May). Semantic understanding of smart contracts: Executable operational semantics of solidity. In *2020 IEEE Symposium on Security and Privacy (SP)* (pp. 1695-1712). IEEE.
- John Wiley & Sons, Inc; (2004). *Public Key Infrastructure (PKI)*. Available from: <http://dx.doi.org/10.1002>.
- Johnston, T. (2014). *Ethereum contracts as legal contracts, Silicon Valley Ethereum* .
- Jones, T., & Richey, R. (2000). *Rapid prototyping methodology in action: A developmental study*. ETR&D, 48(2), 63-80. <http://dx.doi.org/10.1007/bf02313401>.
- Karame G (2016) *.On the security and scalability of bitcoin's Blockchain*. In: Proceedings of the 2016 ACM SIGSAC conference on computer and communications security, Vienna, Austria, October 24–28, 2016, pp 1861–1862
- Kariuki, J. W. O., Karugu, W. N., & Opiyo, M. M. O. (2018). Challenges facing digitization projects in Kenya: case of implementation of national Land information management system. *International Journal of Technology and Systems*, 3(1), 23-42.
- Kavanagh, J. (2018). Setting global standards. *Land Journal*, 10-11.
- Kelly, J., Williams, A. (2016). *Forty Big Banks Evaluate Blockchain-Based Bond Trading System*.
- Kenya, L. O. (2013). *The constitution of Kenya: 2010*. Chief Registrar of the Judiciary.
- Kitchenham B, Charters(S. 2007). *Guidelines for performing Systematic Literature Reviews in Software Engineering*.
- Klapper, L., (2006). *The role of factoring for financing small and medium enterprises*. *Journal of Banking and Finance* 30, 3111-3130
- Kogan Page. Wuttke, D.A., Blome, C., Heese, H.S., Protopappa-Sieke, M., (2016). Supply chain finance: Optimal introduction and adoption decisions. *International Journal of Production Economics*. 178,72-81.
- Kombo, D. K., & Tromp, D. L. (2006). *Proposal and thesis writing: An introduction*. Nairobi: Paulines Publications Africa, 10-45.
- Kreiger, D. (2013). Wall Street's Mortgage Fraud Scandal .

- Kshetri, N. (2017). Blockchain's roles in strengthening cybersecurity and protecting privacy. *Telecommunications Policy*, 41(10), 1027-1038.
- Kshetri, N. (2017). *Can Blockchain Strengthen the Internet of Things?*. IT Professional, 19(4), 68-72.
- Levy, J. (2014). *I love the Blockchain, just not bitcoin*, CoinDesk.
- Luvanda, A., Kimani, D., & Kimwele, D. (2014). Identifying threats associated with man-in-the-middle attacks during communication between a mobile device and the back end server in mobile banking applications. *IOSRJCE*, 16(2), 35-42. <http://dx.doi.org/10.9790/0661-16293542>.
- Mathur, K., & Nandini, K. (2023). Ethereum Blockchain using AES-CMAC. In *ITM Web of Conferences* (Vol. 54, p. 03001). EDP Sciences.
- Mainelli, M. and Milne, A. (2016). The impact and potential of blockchain on the securities transaction lifecycle. In: *SWIFT Institute Working Paper No. 2015-007*.
- Mak, B. (2014). "Authenticity", in *Duranti, L. and Franks, P. (Eds), Encyclopedia of Archival Science*, Rowman & Littlefield, New York, NY.
- Makoro, D. (2015). *Use of Land management information system (NLIMS)*. A case study of ministry of Land, housing and urban development, Nairobi.
- Manjunath, P., Herrmann, M., & Sen, H. (2019). Implementation of blockchain data obfuscation. In *innovation in medicine and healthcare systems, and multimedia* (pp. 545-553). Springer, Singapore.
- Marca, D.A., and C.L. McGowan. (1988). *SADT: Structured analysis and design technique*. McGraw-Hill Book Co., Inc.: New York, NY.
- Mentzer, J.T., DeWitt, W., Keebler, J.S., Min, S., Nix, N.W., Smith, C.D., Zacharia, Z.G., (2001). *Defining supply chain management*. Journal of Business Logistics 22 (2), 1–25
- Meredith, J., (1993). Theory Building through Conceptual Methods, *International Journal of Operations and Production Management*, 13(5), 3 – 11.
- Merriam Webster (2015). Trust Merriam Webster Online Dictionary.
- Miller, A., Juels, A., Shi, E., Parno, B. and Katz, J. (2014). "Permacoin: repurposing bitcoin work for data preservation", and *IEEE Symposium on Security Privacy*, IEEE Press, New York, NY.
- Ministry of Lands (1969). *Recent Land Reforms in Kenya*. In *Land Reform in East Africa*, edited by James Obol-Ochola, Milton Obote Foundation, Kampala, pp. 233 – 254.
- Ministry of Lands (2007). Draft National Land Policy.
- Ministry of Lands (2008). *Department of Surveys: Achievements of Land Adjudication Surveys*.
- Ministry of Lands (2009). *Ardhi bulletins*.
- Mulaku G.C. (2005). Land information management in Kenya. An integrated approach.
- Mulaku, G.C. and McLaughlin, J. (1996). Concepts for Improving Property Mapping in Kenya. *South African Journal of Surveying and Mapping*, 23, Part 4, April 1996, pp. 211 - 216.

- Munari, F., & Toschi, L. (2021). The impact of public funding on science valorization: an analysis of the ERC Proof-of-Concept Programme. *Research Policy*, 50(6), 104211.
- Muriuki, R. W. (2019). *A Tool for tracking Land transactional activities: case of Land registry in Kenya* (Doctoral dissertation, Strathmore University).
- Mwathane, I. (2007). Transparency in Land Administration.
- McPherson, J., Kennedy, C., Slobogean, G., Hilsden, R., & Talbot, M. (2023). Augmented-reality telementoring for leg fasciotomy: a proof-of-concept study. *BMJ Mil Health*, 169(4), 355-358.
- Mwenda N.J. (2001). *Spatial Information in Land tenure reform with special reference to Kenya.FIGURE/habitat/ISK International conference, 2-5 October2001, Nairobi Kenya.*
- Nakamoto S. Bitcoin 2008). *A peer-to-peer electronic cash system*. Consulted. 2008;1(2012):28.
- National Land Commission Act, 2012.
- National Land Information Management System (NLIMS) - *Ministry of Lands and Physical Planning*. (2021). Retrieved 10 December 2021, from <https://Land.s.go.ke/national-land-information-management-system-nlms>.
- Newman, P. (2017). *The Blockchain in the IoT Report*. How distributed ledgers enhance the IoT through better visibility and create trust. BI Intelligence Report.
- Nienhuis, J.J., Corte, M., Lycklama, D., (2013). Real-time financing: Extending e-invoicing to real-time SME financing. *Journal of Payments Strategy & Systems*. 7(3), 232-245.
- Nivethini, P., Meena, S., Krithikaa, V., & Prethija, G. (2019). Data security using blockchain technology. *International Journal of Advanced Networking and Applications*, 279-282.
- Njoroge, T. G. (2019). *Use of Blockchain to Digitize Land Records and Track Land Transfer Transactions in Kenya* (Doctoral dissertation, University of Nairobi).
- O'Dair, M., & Beaven, Z. (2017). *The networked record industry: How Blockchain technology could transform the record industry*. *Strategic Change*, 26(5), 471-480. <http://dx.doi.org/10.1002>.
- O'Reilly Media, Inc.(2015). *Blockchain: Blueprint for a New Economy*
- OCLC/RLG Working Group on Preservation Metadata (2002). A Metadata Framework to Support the Preservation of Digital Objects.
- Omran, Y., (2016). Inclusive Supply Chain Finance approach : *Integrated Supply Chain Finance solution with digitalization*. White paper, Fraunhofer IML.
- Onalo, P.L. (1986). Land law and conveyancing in Kenya. General printers Ltd, Nairobi, Kenya.
- Onguny, P., & Gillies, T. (2019). *Land Conflict in Kenya: A Comprehensive Overview of Literature*. *Les Cahiers D'afrique De Lest*, (53). doi: 10.4000/eastafrica.879
- Palia, D., Sopranzetti, B.J, (2004). Securitizing Account Receivable. *Review of Quantitative Finance and Accounting*, 22: 29-38.

- Pauly, A., Mathews, D. M., & Prasad, J. C. (2021). *Data Protection Using Blockchain Technology*. In Conference Secretariat (P. 115).
- Petersz GW, Panayiy E (2015). *Understanding modern banking ledgers through Blockchain technologies: future of transaction processing and smart contracts on the internet of money*. SSRN 2692487(1)
- Pfohl, H.C., Gomm, M., (2009). Supply Chain Finance: optimizing financial flows in supply chains. *Logistics Research*, 1(3),149-161.
- Polak, P., Sirpal, R., Hamdan, M., (2012). Post-crisis emerging role of the treasurer. In: *European Journal of Scientific Research* 86 (3), 319-339
- Popa, V., (2013). The financial supply chain management: A new solution for supply chain resilience. *Amfiteatru Economic*, 15(33),140-153.
- Pramatari, K., (2007). *Collaborative supply chain practices and evolving technological approaches*. *Supply Chain Management: An International Journal*, 12(3),210 - 220
- Public Land Management. (2019). Retrieved 4 March 2022, from [https://www . Landcommission.go.ke](https://www.Landcommission.go.ke).
- Raval, S., (2016). *Decentralized Applications: Harnessing Bitcoins Blockchain Technology*. First ed. O'Reilly. Sebastopol. Seifert, R.W., Seifert, D., 2011. Financing the chain. *International Commerce Review*, 10 (1):32– 44.
- Rogers, C. (2015),. “*Virtual authenticity: authenticity of digital records from theory to practice*”, unpublished PhD dissertation, University of British Columbia.
- Rosenthal, D.S.H., Robertson, T., Lipkis, T., Reich, V. and Morabito, S. (2005). “*Requirements for digital preservation systems*”, D-Lib Magazine, Vol. 11 No. 11.
- Ruiz, J., Serral, E., & Snoeck, M. (2021). Unifying functional User Interface design principles. *International Journal of Human–Computer Interaction*, 37(1), 47-67.
- Sanni, M. I., & Apriliasari, D. (2021). *Blockchain Technology Application: Authentication System in Digital Education*. Aptisi Transactions on Technopreneurship (ATT), 3(2), 37-48.
- Shakya, S., & Kapoor, V. (2022). A Decentralized Polling System Using Ethereum Technology. *Journal of Information Technology Management*, 14(Security and Resource Management challenges for Internet of Things), 1-8.
- Shibanda, G.G. (2006). *Management of government information in Kenya*.
- Simbizi, M. C. D., Bennett, R. M., & Zevenbergen, J. (2014). Land tenure security: Revisiting and refining the concept for Sub-Saharan Africa's rural poor. *Land use policy*, 36, 231-238.
- Singh, A., Parizi, R. M., Zhang, Q., Choo, K. K. R., & Dehghantaha, A. (2020). Blockchain smart contracts formalization: Approaches and challenges to address vulnerabilities. *Computers & Security*, 88, 101654.
- Singh, P. (2020, December). Role of blockchain technology in digitization of Land records in indian scenario. In IOP Conference Series: *Earth and Environmental Science* 614(1), 012055). IOP Publishing.

- Snow, P., Deery, B., Kirby, P. and Johnston, D. (2015), “*Factom ledger by consensus*”, available at: www.factom.org.
- Snow, P., Deery, B., Lu, J., Johnston, D. and Kirby, P. (2014), “*Factom: business processes secured by immutable audit trails on the Blockchain*”, available at: www.factom.org .
- Stamp Duty- KRA. (2020). Retrieved 10 June 2020, from <https://www.kra.go.ke/en/>
- State of Vermont, United States Act 51 (2015), *An Act Relating to Promoting Economic Development: State of Vermont, United States Act 51*.
- Stuedler, D. (2004). *A framework for the evaluation of Land administration systems* (Doctoral dissertation).
- Swan, M. (2015). *Blockchain: Blueprint for a new economy*. " O'Reilly Media, Inc."
- Syagga, P.M. (1994). *Real estate valuation handbook with special reference to Kenya*. Nairobi university press.
- Sydow, A., Sunny, S. A., & Coffman, C. D. (2020). Leveraging blockchain’s potential—The paradox of centrally legitimate, decentralized solutions to institutional challenges in Kenya. *Journal of Business Venturing Insights*, 14, e00170.
- Szabo, N. (2005). *Secure Property Titles with Owner Authority*.
- Taylor, G. (2008). Is the Torrens System German?. *The Journal of Legal History*, 29(2), 253-285.
- Templar, S., Findlay, C., Hofmann, E., (2016). *Financing the End-to-end Supply Chain: A Reference Guide to Supply Chain Finance*. First ed.
- Thakkar, M. (2020). *Introducing React. js. In Building React Apps with Server-Side Rendering (pp. 41-91)*. Apress, Berkeley, CA.
- The Economist (2015). “*The Great Chain of being sure about things; Blockchain.*”
- UN and International federation of surveyors (1999) *Land tenure and cadastral infrastructures for sustainable development*. 18-22 October 1999, Bathurst Australia.
- Underwood, S. (2016). *Blockchain beyond bitcoin*. *Communications of the ACM*, 59(11), 15-17.
- UNHabitat (2007). *Experts to discuss transparency in Land administration*. Retrieved February 10, 2010 from <http://www.glt.net/en/newspage/experts-to-discuss-transparency-in-Land-administration.html>
- USAID (2010). *USAID Country Profile: Property Rights and Resource Governance – Honduras*, available at: www.usaidLandtenure.net.
- Valley rifts: building peace out of Kenya's tribal conflicts. (2021). Retrieved 30 December 2021, from <https://africa.unwomen.org/en/news-and-events/stories/2021/09/valley-rifts-building-peace-out-of-kenyas-tribal-conflicts>.
- Vasilakis, N., Staicu, C. A., Ntousakis, G., Kallas, K., Karel, B., DeHon, A., & Pradel, M. (2021, November). Preventing dynamic library compromise on node. js via rwx-based privilege reduction. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security* (pp. 1821-1838).

- Von Alan, R. H., March, S. T., Park, J., & Ram, S. (2004). *Design science in information systems research*. MIS quarterly, 28(1), 75-105.
- Vukolic M (2015). The quest for scalable Blockchain fabric: proof-of-work vs. BFT replication. In: Open problems in network security - IFIP WG 11.4 international workshop, inetsec 2015, pp 112
- Wahome, P. T. O. (2017). *Analysis of Forest and Land Legislations and Resource Use Conflict in the Eastern Mau, Nakuru County, Kenya* (Doctoral dissertation, The Open University of Tanzania).
- Wayumba,G.O.(2009) *An Evaluation of the Performance of the Cadastral System in Kenya and a Strategy for its Modernization*.PhD thesis presented in October, 2009 at the school of engineering, University of Nairobi, Kenya.
- What are smart contracts on blockchain? | IBM. (2020). Retrieved 4 January 2020, from <https://www.ibm.com/topics/smart-contracts>.
- What is Firebase?. (2019). Retrieved 20 November 2019, from <https://www.educative.io/edpresso/what-is-firebase>.
- Wild, J., Arnold, M. and Stafford, P. (2015), “*Technology: banks seek the key to Blockchain*”, Financial Times, available at: World Bank (2015). Honduras Overview, available at: www.worldbank.org/en/country/honduras/overview.
- Williamson, I., & Ting, L. (2001). *Land administration and cadastral trends—a framework for re-engineering*. Computers, Environment and Urban Systems, 25(4-5), 339-366.
- Wohrer, M., & Zdun, U. (2018, March). *Smart contracts: security patterns in the ethereum ecosystem and solidity*. In 2018 International Workshop on Blockchain Oriented Software Engineering (IWBOSE) (pp. 2-8). IEEE.
- Wood, G. (2016). *Ethereum: A secure decentralised generalised transaction ledger* (EIP-150 revision). Online] Available: <http://gawwood.com/paper>.
- World Bank report (2006) *Land Administration in Africa. Searching for Land Tenure Security*. May 22-26, 2006, Nairobi Kenya.
- Wright, A., & De Filippi, P. (2015). *Decentralized Blockchain technology and the rise of lex cryptographia*.
- Wuttke, D., Blome, C., Henke, M.(2013). Focusing the Financial Flow of Supply Chains: An Empirical Investigation of Financial Supply Chain Management. *International Journal of Production Economics*, 145(2), 773-789.
- Yeo, G. (2013) “Trust and context in cyberspace”, *Archives and Records* 34(2)214-234.
- Yin, R. K. (2013). *Case study research: Design and methods*. Sage publications.
- Zhang, C. and Dhaliwal, J. (2009). An investigation of resource-based and institutional theoretic factors in technology adoption for operations and supply chain management : *International Journal of Production Economics*, 120(1), 252–269.
- Zyskind G, Nathan O, PentLand A (2015). *Decentralizing privacy: using Blockchain to protect personal data*.

APPENDICES

Appendix 1: Model Source Codes

Appendix 1 (A):The App component Configuration Code

```
import React, { Component } from 'react';
import { HashRouter, Switch, Route , Redirect } from 'react-router-dom';
import firebase from 'firebase/compat/app';
import config from '../config'
import Login from "../features/User/Login";
//import Signup from "../features/User/Login";
import Dashboard from "../features/User/Dashboard";
import 'firebase/compat/auth';
import 'bootstrap/dist/css/bootstrap.css';
import signup from "../features/User/signup";
firebase.initializeApp(config);
const ProtectedRoute = ({ uid,component: Component, ...rest }) => {
  console.log(uid)
  if (uid) {
    return (
<Route {...rest} render={
      props =><Component {...rest} {...props} />
    } />
    )
  }else {
    return (<Redirect to = '/' />)
  }
}
class App extends Component {
  state = {
    User: null,
    Loaded : 'false'
  }
  componentDidMount = async() => {
```

```

        await firebase.auth().onAuthStateChanged(user => this.setState({User : user
, Loaded : true}))
    }
    render() {
        if (this.state.Loaded === true) {
            return (
<HashRouter>
<Switch>
<Route uid ={this.state.User} exact path='/' component={() =><Login uid =
{this.state.User}/>} />
<Route uid={this.state.User} exact path='/signup' component={signup} />
<ProtectedRoute uid ={this.state.User} exact path = "/home" component =
{Dashboard}/>
</Switch>
</HashRouter>
);
        } else {
            return (
<p>....LOADING</p>
)
        }
    }
}
export default App;

```

The Main component Configuration Code

```

import React, { Component } from 'react';
import 'bootstrap/dist/css/bootstrap.min.css';
import jsPDF from 'jspdf'
import 'jspdf-autotable'
class Main extends Component {
    generatePDF = () => {
        const doc = new jsPDF('l', 'pt')

```

```

    doc.autoTable({ html: '#my-table' })
    doc.save('table.pdf');
  }
  render() {
    return (
<div style={{margin: "50px"}}>
<h1>Register Property</h1>
<form onSubmit={(event) => {
    event.preventDefault()
    const name = this.name.value
    const location= this.location.value
    const holder_name = this.holder_name.value
    const lr_no = this.lr_no.value
    const holder_id = this.holder_id.value
    const eths = this.price.value / 514360
    const price = window.web3.utils.toWei(eths.toString(), 'Ether')
    this.props.registerLand(name, location, holder_name, lr_no,
holder_id, price)
  }}>
<div className="form-group mr-sm-2">
<input
      id="productName"
      type="text"
      ref={(input) => { this.name = input }}
      className="form-control"
      placeholder="Size in Hectares"
      required />
</div>
<div className="form-group mr-sm-2">
<input
      id="productLocation"
      type="text"
      ref={(input) => { this.location = input }}
      className="form-control"

```

```

                placeholder="Location"
                required />
</div>
<div className="form-group mr-sm-2">
<input
                id="holder_name"
                type="text"
                ref={(input) => { this.holder_name = input }}
                className="form-control"
                placeholder="Holder Name"
                required />
</div>
<div className="form-group mr-sm-2">
<input
                id="regNumber"
                type="text"
                ref={(input) => { this.lr_no = input }}
                className="form-control"
                placeholder="Land Registry Number"
                required />
</div>
<div className="form-group mr-sm-2">
<input
                id="holder_id"
                type="text"
                ref={(input) => { this.holder_id = input }}
                className="form-control"
                placeholder="Holder ID"
                required />
</div>
<div className="form-group mr-sm-2">
<input
                id="price"
                type="text"

```

```

        ref={(input) => { this.price = input }}
        className="form-control"
        placeholder="Price"
        required />
</div>
<button type="submit" className="btn btn-dark">Add Item</button>
</form>
<p></p>
<div style={{ margin: "100px" }}></div>
<h2>Buy Property</h2>
<table className="table" >
  <thead class="thead-dark">
    <tr>
      <th scope="col">#</th>
      <th scope="col">Size</th>
      <th scope="col">Location</th>
      <th scope="col">Land Owner Address</th>
      <th scope="col">Land Registry Number</th>
      <th scope="col">Price</th>
      <th scope="col">Transaction Date</th>
      <th scope="col">Transaction Time</th>
      <th scope="col">State</th>
    </tr>
  </thead>
  <tbody id="productList">
    { this.props.properties.map((property, key) => {
      return(
        <tr key={key}>
          <th scope="row">{property.id.toString()}</th>
          <td>{property.name.toString()} ha</td>
          <td>{property.location}</td>
          <td>{property._addr}</td>
          <td>{property.lr_no}</td>

```

```

<td>{window.web3.utils.fromWei(property.price.toString() , 'Ether')*514360}
kshs</td>
<td>{new Date(property.timestamp * 1000).toDaevaluatering()}</td>
<td>{new Date(property.timestamp * 1000).toTimeString()}</td>
<td>
      { !property.sold
        ? <button
          className="btn btn-dark"
          name={property.id}
          value={property.price}
          onClick={(event) => {
            this.props.purchaseLand(event.target.name,
event.target.value)
          }}
        >
          Buy
        </button>
        :[
          (this.props.account === property._addr
            ? <button
              className="btn btn-dark"
              name={property.id}
              onClick={(event) => {
                this.props.sellProperty(event.target.name)
              }}
            >
              Sell
            </button>
          : <span>Sold</span>
          ),
        <div key='1'></div>
      ]
    }

```

```

</td>
</tr>
        )
    }}
</tbody>
</table>
<table className="table" id="my-table" style={{ display: "none" }}>
<thead className="thead-dark">
<tr>
<th scope="col">#</th>
<th scope="col">Size</th>
<th scope="col">Location</th>
<th scope="col">Land Owner Address</th>
<th scope="col">Land Registry Number</th>
<th scope="col">Price</th>
<th scope="col">Transaction Date</th>
<th scope="col">Transaction Time</th>
</tr>
</thead>
<tbody id="productList">
        {this.props.transactions.map((property, key) => {
            return (
<tr key={key}>
<th scope="row">{property.id.toString()}</th>
<td>{property.name.toString()} ha</td>
<td>{property.location}</td>
<td>{property._addr}</td>
<td>{property.lr_no}</td>
<td>{window.web3.utils.fromWei(property.price.toString(), 'Ether')} * 514360}
kshs</td>
<td>{new Date(property.timestamp * 1000).toDaevaluatering()}</td>
<td>{new Date(property.timestamp * 1000).toTimeString()}</td>
</tr>

```

```

        )
    }}
</tbody>
</table>
<div>
<button
    style={{ margin: "50px 0 0 0" }}
    className="btn btn-outline-dark"
    onClick={this.generatePDF}
    type="primary">Download Transaction History
</button>
</div>
</div>
    );
}
}
export default Main;

```

The Navbar Component Configuration Code

```

import React, { Component } from 'react';
// import { withRouter } from 'react-router-dom';
import { Button } from 'react-bootstrap';
import firebase from 'firebase/compat/app';
import 'bootstrap/dist/css/bootstrap.css';
class Navbar extends Component {
  signin = () => {
    firebase.auth().signIn().then(() => {
      this.setState({
        user:null
      })
      this.props.history.push("/");
    }).catch(function(error) {
      // An error happened.
    });
  };
}

```

```

    }
    render() {
      return (
<nav
      className="navbar navbar-dark fixed-top bg-dark flex-md-nowrap p-0
shadow"
      style={{padding: "20px"}}
    >
<a className="text-white" href="/" style={{padding: "0 0 0 5px"}}>
      Welcome Back
</a>
<ul className="navbar-nav px-3">
<li className="nav-item text-nowrap d-none d-sm-none d-sm-block">
<small className="text-white">
<span id="account">{this.props.account}</span>
<Button
  onClick
  =
  {this.signout}
  style={{margin:
  "20px
20px"}}>Signout</Button>
</small>
</li>
</ul>
</nav>
      );
    }
  }
}
export default Navbar;

```

The Dashboard Component Configuration Code

```

import React, { Component } from 'react';
import Navbar from "../../components/Navbar";
import Main from "../../components/Main";
import Web3 from 'web3'
import Registry from '../../abis/Registry.json'
import '../../components/App.css';
import 'bootstrap/dist/css/bootstrap.min.css';

```

```

class Dashboard extends Component {
  constructor(props) {
    super(props)
    this.state = {
      account: "",
      propCount: 0,
      transactionCount: 0,
      properties: [],
      transactions: [],
      loading: true
    }

    this.registerLand = this.registerLand.bind(this);
    this.purchaseLand = this.purchaseLand.bind(this);
    this.sellProperty = this.sellProperty.bind(this);
  }
  async componentWillMount() {
    await this.loadWeb3()
    await this.loadBlockchainData()
  }
  // Detects the presence of ethereum web provider
  async loadWeb3() {
    if (window.ethereum) {
      window.web3 = new Web3(window.ethereum)
      await window.ethereum.enable()
    }
    else if (window.web3) {
      window.web3 = new Web3(window.web3.currentProvider)
    }
    else {
      window.alert('Non-Ethereum browser detected. You should consider
trying MetaMask!')
    }
  }
}

```

```

// fetches accounts/nodes from blockchain
async loadBlockchainData() {
  const web3 = window.web3
  // Load account
  const accounts = await web3.eth.getAccounts()
  this.setState({ account: accounts[0] })
  const networkId = await web3.eth.net.getId()
  const networkData = Registry.networks[networkId]
  if(networkData) {
    const registry = web3.eth.Contract(Registry.abi, networkData.address)
    this.setState({ registry })
    const transactionCount = await registry.methods.transactionCount().call()
    const propCount = await registry.methods.propCount().call()
    this.setState({ propCount })
    this.setState({ transactionCount })
    for (let i = 1; i <= propCount; i++) {
      const property = await registry.methods.properties(i).call()
      this.setState({
        properties: [...this.state.properties, property]
      })
    }
    for (let i = 1; i <= transactionCount; i++) {
      const transaction = await registry.methods.transactions(i).call()
      this.setState({
        transactions: [...this.state.transactions, transaction]
      })
    }
    this.setState({ loading: false })
  } else {
    window.alert('Marketplace contract not deployed to detected network.')
  }
}
refresh(){
  window.location.reload()
}

```

```

    }
    registerLand(name, location, holder_name, lr_no, holder_id, price){
      this.setState({ loading: true})
      this.state.registry.methods.regProperty(name, location, holder_name, lr_no,
holder_id, price)
        .send({ from: this.state.account })
        .once('receipt', (receipt) => {
          this.setState({ loading: false })
        })
      setTimeout(this.refresh, 30000);
    }
    purchaseLand(id, price) {
      this.setState({ loading: true })
      this.state.registry.methods.buyProperty(id)
        .send({ from: this.state.account, value: price })
        .once('receipt', (receipt) => {
          this.setState({ loading: false })
        })
      setTimeout(this.refresh, 30000);
    }
    sellProperty(id) {
      this.setState({ loading: true })
      this.state.registry.methods.approve(id)
        .send({ from: this.state.account })
        .once('receipt', (receipt) => {
          this.setState({ loading: false })
        })
      setTimeout(this.refresh, 30000);
    }
    render() {
      return (
<div>
<Navbar account={this.state.account} />
<div className="container-fluid mt-5">

```

```

<main role="main" className="col-lg-12 d-flex">
  { this.state.loading
    ? <div id="loader" className="text-center"><p
className="text-center">Loading...</p></div>
    : <Main
      account={this.state.account}
      transactions={this.state.transactions}
      properties={this.state.properties}
      sellProperty={this.sellProperty}
      purchaseLand={this.purchaseLand}
      registerLand={this.registerLand} />
    }
</main>
</div>
</div>
);
}
}
export default Dashboard;

```

The Login Component Code

```

import React, {useState, useEffect} from 'react'
import { Card } from 'react-bootstrap'
import { Link, withRouter } from 'react-router-dom'
import firebase from 'firebase/compat/app';
import 'bootstrap/dist/css/bootstrap.css';
function Login(props) {
  const [Email, setEmail] = useState("");
  const [Password, setPassword] = useState("");
  const login = (e) => {
    e.preventDefault();
    firebase.auth().signInWithEmailAndPassword(Email, Password)
      .then(res => {
        localStorage.setItem( "isAuth" , "true" )

```

```

        props.history.push({
            pathname: '/home',
            state: { ...props , ...res }
        });
    })
    .catch(err => console.log(err))
}
useEffect(() => {
    if (props.location.state !== undefined) {
        setEmail(props.location.state.Email);
        setPassword(props.location.state.Password);
    }
}, [props.location.state])
useEffect(() => {
    console.log("inside useEffect", props.uid);
    if (props.uid) {
        props.history.push({
            pathname: '/home',
        });
    }
}, [props.uid])
return (
<div>
< Card style = {
    {
        width:"450px",
        position: 'fixed',
        top: '5%',
        left: '50%',
        transform: 'translate(-50% , 50%)'
    }
} >
<Card.Body>
<form onSubmit= {login} >

```

```

<div className="form-outline mb-4">
  <label      className="form-label"      htmlFor="form1Example1">Email
  address</label>
  <input      type="email"      id="form1Example1"
  placeholder="someOne@example.com"  className="form-control"  value =
  {Email}
      onChange = { e => setEmail(e.target.value)}
      autoComplete = "on"/>
</div>
<div className="form-outline mb-4">
  <label className="form-label" htmlFor="form1Example2">Password</label>
  <input  type="password"  id="form1Example2"  className="form-control"
  placeholder = "Password"
      value = {Password}
      onChange = { e => setPassword(e.target.value)}
      autoComplete = "off"/>
</div>
<button  type="submit"  text="Login"  className="btn  btn-primary  btn-
block">Sign in</button>
</form>
<br/>
</Card.Body>
<Card.Footer>
<span>Dont have an account? <Link to ="/signup">signup</Link></span>
</Card.Footer>
</Card>
</div>
)
}
export default withRouter(Login);

```

The SignUp Component Code

```

import React, {useState, useEffect} from 'react'
import { Card } from 'react-bootstrap'

```

```

import { Link, withRouter } from 'react-router-dom'
import firebase from 'firebase/compat/app';
import 'bootstrap/dist/css/bootstrap.css';
function Signup(props) {
  const [Email, setEmail] = useState("");
  const [Password, setPassword] = useState("");
  const [VerifyPass, setVerifyPass] = useState("");
  const [IsValid, setIsValid] = useState(true);
  const [IsMatch, setIsMatch] = useState(true);
  const Signup = (e) => {
    e.preventDefault();
    if (Password === VerifyPass) {
      firebase.auth().createUserWithEmailAndPassword(Email,
Password).then(function (user) {
        alert('User created, please login')
        props.history.push({
          pathname: '/',
          state: {
            Email,
            Password
          }
        })
      }).catch(function (error) {
        var errorMessage = error.message;
        console.log(errorMessage)
      });
    }else {
      setIsMatch(false);
      console.log(IsMatch)
    }
  }
  useEffect(() => {
    console.log("inside useEffect", props.uid);
    if (props.uid) {

```

```

        props.history.push({
          pathname: '/home',
        });
      }
    }, [props.uid])
    return (
<div>
< Card style = {
  {
    width:"450px",
    position: 'fixed',
    top: '-10%',
    left: '50%',
    transform: 'translate(-50% , 50%)'
  }
} >
<Card.Body>
<form onSubmit ={Signup} >
<div className="form-outline mb-4">
<label      className="form-label"      htmlFor="form1Example1">Email
address</label>
<input      type="email"      id="form1Example1"
placeholder="someOne@example.com"  className="form-control"  value  =
{Email}
      onChange ={ e => setEmail(e.target.value)}
      autoComplete = "off"/>
</div>
<div className="form-outline mb-4">
<label className="form-label" htmlFor="form1Example2">Password</label>
<input
      type="password"
      id="form1Example2"

```

```

        className="form-control"
        placeholder = "Password"
        autoComplete = "off"
        value = {Password}
        onChange = {e => {
            setPassword(e.target.value);
            if (!e.target.value.match(/^(?=.*{8,})(?=.*\d)(?=.*[a-
z])(?=.*[A-Z])(?=.*[@#$%&]).*$$/)) {
                setIsValid(false)
            } else {
                setIsValid(true)
            }
        }}
    />
    {
        IsValid ===false ?
<p
        style={{
            fontSize : '0.7rem',
            color : 'red'
        }}
        >Password must contain 1 Upper case, one special character, one number and
        atleast 8 character </p>
        : <p></p>
    }
</div>
<div className="form-outline mb-4">
<label className="form-label" htmlFor="form1Example2">Re-Enter
Password</label>
<input
        id="form1Example2"
        className="form-control"
        type="password"
        placeholder = "Password"

```

```

        autoComplete = "off"
        value = {VerifyPass}
        onChange = {e => {
            setVerifyPass(e.target.value);
            if (Password === e.target.value) {
                setIsMatch(true);
            } else {
                setIsMatch(false)
            }
        }}
    />
<br />
    {
        IsMatch === false ?
    <p
        style={{
            fontSize : '0.7rem',
            color : 'red'
        }}
    >Password dosen't match</p>
    : <p></p>
    }
</div>
<button type="submit" text="Login" className="btn btn-primary btn-
block">Sign Up</button>
</form>
</Card.Body>
<Card.Footer>
<span>Already have an account? <Link to="/">signin</Link></span>
</Card.Footer>
</Card>
</div>
)
}

```

```
export default withRouter(Signup);
```

Solidity Migration Code

```
pragma solidity ^0.8.0;
//SPDX-License-Identifier: UNLICENSED
contract Migrations {
    address public owner;
    uint public last_completed_migration;
    constructor(){
        owner = msg.sender;
    }
    modifier restricted() {
        if (msg.sender == owner) _;
    }
    function setCompleted(uint completed) public restricted {
        last_completed_migration = completed;
    }
    function upgrade(address new_address) public restricted {
        Migrations upgraded = Migrations(new_address);
        upgraded.setCompleted(last_completed_migration);
    }
}
```

Appendix 1(B): Solidity Land Transactions Code

```
pragma solidity ^0.8.0;
import "@openzeppelin/contracts/access/Ownable.sol";
//SPDX-License-Identifier: UNLICENSED
contract Registry is Ownable{
    uint public propCount = 0;
    uint holderCount = 0;
    uint public transactionCount = 0;
    //struct that holds property owner details
    struct holder {
        string name;
```

```

    string tax_pin;
    string email;
    uint id_no;
    uint contact;
    address owner;
    bool isExist;
}
//struct that holds property details
struct property {
    uint id;
    string name;
    string location;
    string holder_name;
    string lr_no;
    uint holder_id;
    address _addr;
    uint price;
    bool sold;
    uint timestamp;
}
// array for storing properties
mapping(uint => property) public properties;
mapping(uint => property) public transactions;
//array for storing holders
holder [] public holders;
//event that listens to addition of a new property into the properties array
event NewProperty(uint propertyId, string name, string location, string
holder_name, string lr_no);
event LogApprovedForTransaction(uint propertyId);
event LogSold(uint propertyId);
// a lookup of how many properties a holder has
mapping (address => uint) public holderToPropertyCount;
// a lookup of property owners by their property id
mapping (uint => address) public propertyToHolder;

```

```

// a lookup of validatorDetails by their addresses
mapping (string => bool) public regToAvailable;
mapping (address => property) public publicProperty;
mapping (address => holder) public holderdetails;
// function that registers holders
function regHolders(string memory name, string memory tax_pin, string
memory email,uint id_no, uint contact) public {
    holderCount++;
    holders.push(holder(name,tax_pin,email, id_no,contact,msg.sender,true));
holderdetails[msg.sender]=holder(name,tax_pin,email,id_no,contact,msg.sender,t
rue);
}
//function that returns holder details
function getHolders() public view returns (holder[] memory){
    return holders;
}
//function that registers property by their owners
function regProperty(string memory name, string memory location, string
memory holder_name, string memory lr_no, uint holder_id,uint price) public {
    //push each property to the properties array
    require(!regToAvailable[lr_no]);
    bool pd = false;
    // add property count
    propCount++;
    transactionCount++;
    uint timestamp = block.timestamp;
    properties[propCount] = property(propCount, name, location, holder_name,
lr_no, holder_id, msg.sender,price,pd, timestamp);
    transactions[transactionCount] = property(propCount, name, location,
holder_name, lr_no, holder_id, msg.sender,price,pd, timestamp);
    regToAvailable[lr_no] = true;
    holderToPropertyCount[msg.sender]++;
    emit NewProperty(propCount, name, location, holder_name, lr_no);
}

```

```

// function that gets all properties by their owners address
function getPropertyByHolder(address _holder) external view returns(uint[]
memory) {
    uint[] memory result = new uint[] (holderToPropertyCount[_holder]);
    uint counter = 0;
    //loops through the holders array checking each holder and the properties
they own and adds a counter for each property a holder has
    for (uint i = 0; i < holders.length; i++) {
        if(propertyToHolder[i] == _holder) {
            result[counter] = i;
            counter++;
        }
    }
    return result;
}

function buyProperty(uint _propertyId) public payable {
    property memory _land= properties[_propertyId];
    address _seller = _land._addr;
    // Require that the land is for sale
    require(!_land.sold);
    // Require that the buyer is not the seller
    require(_seller != msg.sender);
    // Ensure price is correct
    require(msg.value >= _land.price);
    // Update the product
    _land._addr = msg.sender;
    _land.sold = true;
    properties[_propertyId] = _land;
    // Send money
    payable(_seller).transfer(msg.value);
    bool pd = false;
    // Transfer property ownership
    _land.timestamp = block.timestamp;
    transactionCount++;
}

```

```

        transactions[transactionCount] = property(transactionCount, _land.name,
        _land.location, _land.holder_name, _land.lr_no, _land.holder_id,
msg.sender,_land.price,pd, block.timestamp);
        emit LogSold(_propertyId);
    }
    function sellProperty (uint _propertyId) public view{
        property memory _land= properties[_propertyId];
        // require that the seller is the land owner
        require(msg.sender == _land._addr);
        _land.sold = false;
    }
}

```

Appendix 1(C):Land Registry Confirmation and Transaction Component Code

```

import React, {useState, useEffect} from 'react'
import { Card } from 'react-bootstrap'
import { Link, withRouter } from 'react-router-dom'
import firebase from 'firebase/compat/app';
import 'bootstrap/dist/css/bootstrap.css';
function Signup(props) {
const [Email, setEmail] = useState("");
const [Password, setPassword] = useState("");
const [VerifyPass, setVerifyPass] = useState("");
const [IsValid, setIsValid] = useState(true);
const [IsMatch, setIsMatch] = useState(true);
const Signup = (e) => {
e.preventDefault();
if (Password === VerifyPass) {
firebase.auth().createUserWithEmailAndPassword(Email, Password).then(function
(user) {
alert('User created, please login')
props.history.push({
pathname: '/',
state: {
Email,

```

```

Password
}
})
}).catch(function (error) {
var errorMessage = error.message;
console.log(errorMessage)
});
}else {
setIsMatch(false);
console.log(IsMatch)
}
}
useEffect(() => {
console.log("inside useEffect", props.uid);
if (props.uid) {
props.history.push({
pathname: '/home',
});
}
}, [props.uid])
return (
<div>
< Card style = {
{
width:"450px",
position: 'fixed',
top: '-10%',
left: '50%',
transform: 'translate(-50% , 50%)'
}
} >
<Card.Body>
<form onSubmit ={Signup} >
<div className="form-outline mb-4">

```

```

<label className="form-label" htmlFor="form1Example1">Email address</label>
<input type="email" id="form1Example1" placeholder="someOne@example.com"
className="form-control" value = {Email}
onChange = { e => setEmail(e.target.value)}
autoComplete = "off"/>
</div>
<div className="form-outline mb-4">
<label className="form-label" htmlFor="form1Example2">Password</label>
<input
type="password"
id="form1Example2"
className="form-control"
placeholder = "Password"
autoComplete = "off"
value = {Password}
onChange = {e => {
setPassword(e.target.value);
if (!e.target.value.match(/^(?=.*{8,})(?=.*\d)(?=.*[a-z])(?=.*[A-
Z])(?=.*[@#$%&]).*$/) {
setIsValid(false)
} else {
setIsValid(true)
}
}}
/>
{
IsValid ===false ?
<p
style ={{
fontSize : '0.7rem',
color : 'red'
}}
>Password must contain 1 Upper case, one special character, one number and atleast 8
character </p>

```

```

: <p></p>
}
</div>
<div className="form-outline mb-4">
<label className="form-label" htmlFor="form1Example2">Re-Enter Password</label>
<input
id="form1Example2"
className="form-control"
type="password"
placeholder = "Password"
autoComplete = "off"
value = { VerifyPass }
onChange = { e => {
setVerifyPass(e.target.value);
if (Password === e.target.value) {
setIsMatch(true);
} else {
setIsMatch(false)
}
}}
/>
<br />
{
IsMatch ===false ?
<p
style ={{
fontSize : '0.7rem',
color : 'red'
}}
>Password dosen't match</p>
: <p></p>
}

</div>

```

```

<button type="submit" text="Login" className="btn btn-primary btn-block">Sign
Up</button>
</form>
</Card.Body>
<Card.Footer>
<span>Already have an account? <Link to="/">signin</Link></span>
</Card.Footer>
</Card>
</div>
)
}
export default withRouter(Signup);
import React, {useState, useEffect} from 'react'
import { Card } from 'react-bootstrap'
import { Link, withRouter } from 'react-router-dom'
import firebase from 'firebase/compat/app';
import 'bootstrap/dist/css/bootstrap.css';
function Login(props) {
const [Email, setEmail] = useState("");
const [Password, setPassword] = useState("");
const login = (e) => {
e.preventDefault();
firebase.auth().signInWithEmailAndPassword(Email, Password)
.then(res => {
localStorage.setItem( "isAuth" , "true" )
props.history.push({
pathname: '/home',
state: { ...props , ...res }
});
})
.catch(err => console.log(err))
}
useEffect(() => {

```

```

if (props.location.state !== undefined) {
  setEmail(props.location.state.Email);
  setPassword(props.location.state.Password);
}
}, [props.location.state])
useEffect(() => {
  console.log("inside useEffect", props.uid);
  if (props.uid) {
    props.history.push({
      pathname: '/home',
    });
  }
}, [props.uid])
return (
  <div>
    < Card style = {
      {
        width:"450px",
        position: 'fixed',
        top: '5%',
        left: '50%',
        transform: 'translate(-50% , 50%)'
      }
    } >
    <Card.Body>
    <form onSubmit= {login} >
    <div className="form-outline mb-4">
    <label className="form-label" htmlFor="form1Example1">Email address</label>
    <input type="email" id="form1Example1" placeholder="someone@example.com"
      className="form-control" value = {Email}
      onChange ={ e => setEmail(e.target.value)}
      autoComplete = "on"/>
    </div>
    <div className="form-outline mb-4">

```

```

<label className="form-label" htmlFor="form1Example2">Password</label>
<input type="password" id="form1Example2" className="form-control" placeholder =
"Password"
value = {Password}
onChange = { e => setPassword(e.target.value)}
autoComplete = "off"/>

```

```

</div>
<button type="submit" text="Login" className="btn btn-primary btn-block">Sign
in</button>
</form>
<br/>

```

```

</Card.Body>
<Card.Footer>
<span>Dont have an account? <Link to ="/signup">signup</Link></span>
</Card.Footer>
</Card>
</div>
)
}

```

```

export default withRouter(Login);
pragma solidity ^0.8.17;
import "./Ownable.sol";
//SPDX-License-Identifier: UNLICENSED
contract Registry is Ownable{
    constructor() Ownable(msg.sender){}
    uint public propCount = 0;
    uint holderCount = 0;
    uint public transactionCount = 0;
    // struct for property owner details
    struct holder {
        string name;
        string tax_pin;
    }
}

```

```

    string email;
    uint id_no;
    uint contact;
    address owner;
    bool isExist;
}
//struct for property details
struct property {
    uint id;
    string name;
    string location;
    string holder_name;
    string lr_no;
    uint holder_id;
    address _addr;
    uint price;
    bool sold;
    uint timestamp;
}
// stores properties
mapping(uint => property) public properties;
mapping(uint => property) public transactions;
// stores holders
holder [] public holders;
// event that listens to addition of a new property into the properties array
event NewProperty(uint propertyId, string name, string location, string holder_name,
string lr_no);
event LogApprovedForTransaction(uint propertyId);
event LogSold(uint propertyId);
// a lookup of how many properties a holder has
mapping (address => uint) public holderToPropertyCount;
// a lookup of property owners by their property id
mapping (uint => address) public propertyToHolder;
// a lookup of validatorDetails by their addresses

```

```

mapping (string => bool) public regToAvailable;
mapping (address => property) public publicProperty;
mapping (address => holder) public holderdetails;
// registers holders
function regHolders(string memory name, string memory tax_pin, string memory
email,uint id_no, uint contact) public {
    holderCount++;
    holders.push(holder(name,tax_pin,email, id_no,contact,msg.sender,true));
    holderdetails[msg.sender]=holder(name,tax_pin,email,id_no,contact,msg.sender,true);
}
// returns holder details
function getHolders() public view returns (holder[] memory){
    return holders;
}

```

Appendix 1(D):Land registers property by Seller

```

function regProperty(string memory name, string memory location, string memory
holder_name, string memory lr_no, uint holder_id,uint price) public {
    //push each property to the properties array
    require(!regToAvailable[lr_no]);
    bool pd = true;
    // add property count
    propCount++;
    transactionCount++;
    uint timestamp = block.timestamp;
    properties[propCount] = property(propCount, name, location, holder_name, lr_no,
holder_id, msg.sender,price,pd, timestamp);
    transactions[transactionCount] = property(propCount, name, location, holder_name,
lr_no, holder_id, msg.sender,price,pd, timestamp);
    regToAvailable[lr_no] = true;
    holderToPropertyCount[msg.sender]++;
    emit NewProperty(propCount, name, location, holder_name, lr_no);
}
// function that gets all properties by their owners address

```

```

function getPropertyByHolder(address _holder) external view returns(uint[] memory)
{
    uint[] memory result = new uint[] (holderToPropertyCount[_holder]);
    uint counter = 0;
    //loops through the holders array checking each holder and the properties they own
and adds a counter for each property a holder has

    for (uint i = 0; i < holders.length; i++) {
        if(propertyToHolder[i] == _holder) {
            result[counter] = i;
            counter++;
        }
    }
    return result;
}

function buyProperty(uint _propertyId) public payable {
    property memory _land= properties[_propertyId];
    address _seller = _land._addr;
    // Require that the land is for sale
    require(!_land.sold);
    // Require that the buyer is not the seller
    require(_seller != msg.sender);
    // Ensure price is correct
    require(msg.value >= _land.price);
    // Update the product
    _land._addr = msg.sender;
    _land.sold = true;
    properties[_propertyId] = _land;
    // Send money
    payable(_seller).transfer(msg.value);
    bool pd = false;
    // Transfer property ownership
    _land.timestamp = block.timestamp;

```

```

        transactionCount++;
        transactions[transactionCount] = property(transactionCount, _land.name,
        _land.location, _land.holder_name, _land.lr_no, _land.holder_id,
msg.sender, _land.price, pd, block.timestamp);
        emit LogSold(_propertyId);
    }
function sellProperty (uint _propertyId) public {
    property memory _land= properties[_propertyId];
    // require that the seller is the land owner
    require(msg.sender == _land._addr);
    // change the owner of the land to the blockchain address
    _land.sold = false;
    properties[_propertyId] = _land;
}
}

// Land registers property by Seller
function regProperty(string memory name, string memory location, string memory
holder_name, string memory lr_no, uint holder_id, uint price) public {
    //push each property to the properties array
    require(!regToAvailable[lr_no]);
    bool pd = true;
    // add property count
    propCount++;
    transactionCount++;
    uint timestamp = block.timestamp;
    properties[propCount] = property(propCount, name, location, holder_name, lr_no,
holder_id, msg.sender, price, pd, timestamp);
    transactions[transactionCount] = property(propCount, name, location, holder_name,
lr_no, holder_id, msg.sender, price, pd, timestamp);
    regToAvailable[lr_no] = true;
    holderToPropertyCount[msg.sender]++;
    emit NewProperty(propCount, name, location, holder_name, lr_no);
}

// function that gets all properties by their owners address

```

```

function getPropertyByHolder(address _holder) external view returns(uint[] memory)
{
    uint[] memory result = new uint[] (holderToPropertyCount[_holder]);
    uint counter = 0;
    //loops through the holders array checking each holder and the properties they own
and adds a counter for each property a holder has
    for (uint i = 0; i < holders.length; i++) {
        if(propertyToHolder[i] == _holder) {
            result[counter] = i;
            counter++;
        }
    }
    return result;
}

function buyProperty(uint _propertyId) public payable {
    property memory _land= properties[_propertyId];
    address _seller = _land._addr;
    // Require that the land is for sale
    require(!_land.sold);
    // Require that the buyer is not the seller
    require(_seller != msg.sender);
    // Ensure price is correct
    require(msg.value >= _land.price);
    // Update the product
    _land._addr = msg.sender;
    _land.sold = true;
    properties[_propertyId] = _land;
    // Send money
    payable(_seller).transfer(msg.value);
    bool pd = false;
    // Transfer property ownership
    _land.timestamp = block.timestamp;
    transactionCount++;
}

```

```

        transactions[transactionCount] = property(transactionCount, _land.name,
        _land.location, _land.holder_name, _land.lr_no, _land.holder_id,
msg.sender, _land.price, pd, block.timestamp);
        emit LogSold(_propertyId);
    }
    function sellProperty (uint _propertyId) public {
        property memory _land= properties[_propertyId];
        // require that the seller is the land owner
        require(msg.sender == _land._addr);
        // change the owner of the land to the blockchain address
        _land.sold = false;
        properties[_propertyId] = _land;
    }
}

```

Appendix 1(E):Land registers property by Buyer

```

import React, { Component } from 'react';
import Navbar from "../components/Navbar";
import Main from "../components/Main";
import Web3 from 'web3'
import Registry from '../abis/Registry.json'
import '../components/App.css';
import 'bootstrap/dist/css/bootstrap.min.css';
class Dashboard extends Component {
    constructor(props) {
        super(props)
        this.state = {
            account: "",
            propCount: 0,
            transactionCount: 0,
            properties: [],
            transactions: [],
            loading: true
        }
    }
}

```

```

    this.registerLand = this.registerLand.bind(this);
    this.purchaseLand = this.purchaseLand.bind(this);
    this.sellProperty = this.sellProperty.bind(this);
  }
  async componentWillMount() {
    await this.loadWeb3()
    await this.loadBlockchainData()
  }

  // Detects the presence of ethereum web provider
  async loadWeb3() {
    if (window.ethereum) {
      window.web3 = new Web3(window.ethereum)
      await window.ethereum.enable()
    }
    else if (window.web3) {
      window.web3 = new Web3(window.web3.currentProvider)
    }
    else {
      window.alert('Non-Ethereum browser detected. You should consider trying
MetaMask!')
    }
  }

  // fetches accounts/nodes from blockchain
  async loadBlockchainData() {
    const web3 = window.web3
    // Load account
    const accounts = await web3.eth.getAccounts()
    this.setState({ account: accounts[0] })
    const networkId = await web3.eth.net.getId()
    const networkData = Registry.networks[networkId]
    if(networkData) {
      const registry = web3.eth.Contract(Registry.abi, networkData.address)
      this.setState({ registry })
    }
  }

```

```

const transactionCount = await registry.methods.transactionCount().call()
const propCount = await registry.methods.propCount().call()
this.setState({propCount})
this.setState({transactionCount})
for (let i = 1; i <= propCount; i++) {
  const property = await registry.methods.properties(i).call()
  this.setState({
    properties: [...this.state.properties, property]
  })
}
for (let i = 1; i <= transactionCount; i++) {
  const transaction = await registry.methods.transactions(i).call()
  this.setState({
    transactions: [...this.state.transactions, transaction]
  })
}
this.setState({ loading: false })
} else {
  window.alert('Marketplace contract not deployed to detected network.')
}
}
refresh(){
  window.location.reload()
}
registerLand(name, location, holder_name, lr_no, holder_id, price){
  this.setState({ loading: true })
  this.state.registry.methods.regProperty(name, location, holder_name, lr_no,
holder_id, price)
    .send({from: this.state.account})

    .once('receipt', (receipt) => {
      this.setState({ loading: false })
    })
}

```

```

        setTimeout(this.refresh, 30000);
    }
    purchaseLand(id, price) {
        this.setState({ loading: true })
        this.state.registry.methods.buyProperty(id)
            .send({ from: this.state.account, value: price })
            .once('receipt', (receipt) => {
                this.setState({ loading: false })
            })
        setTimeout(this.refresh, 30000);
    }
    sellProperty(id) {
        this.setState({ loading: true })
        this.state.registry.methods.approve(id)
            .send({ from: this.state.account })
            .once('receipt', (receipt) => {
                this.setState({ loading: false })
            })
        setTimeout(this.refresh, 30000);
    }
    render() {
        return (
<div>
<Navbar account={this.state.account} />
<div className="container-fluid mt-5">
<main role="main" className="col-lg-12 d-flex">
            { this.state.loading
                ? <div id="loader" className="text-center"><p className="text-
center">Loading...</p></div>
                : <Main
                    account={this.state.account}
                    transactions={this.state.transactions}
                    properties={this.state.properties}

```

```

        sellProperty={ this.sellProperty}
        purchaseLand={ this.purchaseLand}
        registerLand={ this.registerLand} />
    }
</main>
</div>
</div>
    );
}
}
export default Dashboard;
pragma solidity ^0.8.0;
import './registry.sol';
//SPDX-License-Identifier: UNLICENSED
contract PropertyOwnership is Registry {
    mapping (address => uint) public ownerToToken;
    // modifier ensures only the owner of the property approves
    modifier onlyOwnerOf(uint _propertyId) {
        require(msg.sender == propertyToHolder[_propertyId]);
        _;
    }
}
pragma solidity ^0.8.0;
//SPDX-License-Identifier: UNLICENSED
contract Migrations {
    address public owner;
    uint public last_completed_migration;
    constructor(){
        owner = msg.sender;
    }
    modifier restricted() {
        if (msg.sender == owner) _;
    }
    function setCompleted(uint completed) public restricted {

```

```

    last_completed_migration = completed;
}
function upgrade(address new_address) public restricted {
    Migrations upgraded = Migrations(new_address);
    upgraded.setCompleted(last_completed_migration);
}
}
import { Context } from "./Context.sol";
/**
 * @dev Contract module which provides a basic access control mechanism, where
 * there is an account (an owner) that can be granted exclusive access to
 * specific functions.
 *
 * The initial owner is set to the address provided by the deployer. This can
 * later be changed with {transferOwnership}.
 *
 * This module is used through inheritance. It will make available the modifier
 * `onlyOwner`, which can be applied to your functions to restrict their use to
 * the owner.
 */
abstract contract Ownable is Context {
    address private _owner;
    /**
     * @dev The caller account is not authorized to perform an operation.
     */
    error OwnableUnauthorizedAccount(address account);
    /**
     * @dev The owner is not a valid owner account. (eg. `address(0)`)
     */
    error OwnableInvalidOwner(address owner);
    event OwnershipTransferred(address indexed previousOwner, address indexed
newOwner);
    /**

```

* @dev Initializes the contract setting the address provided by the deployer as the initial owner.

```
*/  
constructor(address initialOwner) {  
    _transferOwnership(initialOwner);  
}
```

```
/**
```

* @dev Throws if called by any account other than the owner.

```
*/
```

```
modifier onlyOwner() {  
    _checkOwner();  
    _;  
}
```

```
/**
```

* @dev Returns the address of the current owner.

```
*/
```

```
function owner() public view virtual returns (address) {  
    return _owner;  
}
```

```
/**
```

* @dev Throws if the sender is not the owner.

```
*/
```

```
function _checkOwner() internal view virtual {  
    if (owner() != _msgSender()) {  
        revert OwnableUnauthorizedAccount(_msgSender());  
    }  
}
```

```
/**
```

* @dev Leaves the contract without owner. It will not be possible to call

* `onlyOwner` functions. Can only be called by the current owner.

```
*
```

* NOTE: Renouncing ownership will leave the contract without an owner,

* thereby disabling any functionality that is only available to the owner.

```
*/
```

```

function renounceOwnership() public virtual onlyOwner {
    _transferOwnership(address(0));
}
/**
 * @dev Transfers ownership of the contract to a new account (`newOwner`).
 * Can only be called by the current owner.
 */
function transferOwnership(address newOwner) public virtual onlyOwner {
    if (newOwner == address(0)) {
        revert OwnableInvalidOwner(address(0));
    }
    _transferOwnership(newOwner);
}
/**
 * @dev Transfers ownership of the contract to a new account (`newOwner`).
 * Internal function without access restriction.
 */
function _transferOwnership(address newOwner) internal virtual {
    address oldOwner = _owner;
    _owner = newOwner;
    emit OwnershipTransferred(oldOwner, newOwner);
}
}

```

Appendix 1(F):The Smart Contract Addresses

```

{
  "contractName": "Address",
  },
  "id": 1378,
  "license": "MIT",
  "nodeType": "SourceUnit",
  "nodes": [
    {
      "id": 1082,

```

```

"literals": [
  "solidity",
  "^",
  "0.8",
  ".0"
],
"nodeType": "PragmaDirective",
"src": "33:23:5"
},
{
  "abstract": false,
  "baseContracts": [],
  "canonicalName": "Address",
  "contractDependencies": [],
  "contractKind": "library",
  "documentation": {
    "id": 1083,
    "nodeType": "StructuredDocumentation",
    "src": "58:67:5",
    "text": " @dev Collection of functions related to the address type"
  },
  "fullyImplemented": true,
  "id": 1377,
  "linearizedBaseContracts": [
    1377
  ],
  "name": "Address",
  "nameLocation": "134:7:5",
  "nodeType": "ContractDefinition",
  "nodes": [
    {
      "body": {
        "id": 1099,
        "nodeType": "Block",

```

```

"src": "784:311:5",
"statements": [
  {
    "assignments": [
      1092
    ],
    "declarations": [
      {
        "constant": false,
        "id": 1092,
        "mutability": "mutable",
        "name": "size",
        "nameLocation": "989:4:5",
        "nodeType": "VariableDeclaration",
        "scope": 1099,
        "src": "981:12:5",
        "stateVariable": false,
        "storageLocation": "default",
        "typeDescriptions": {
          "typeIdentifier": "t_uint256",
          "typeString": "uint256"
        },
      },
    ],
    "declarations": [
      {
        "constant": false,
        "id": 1278,
        "mutability": "mutable",
        "name": "success",
        "nameLocation": "5857:7:5",
        "nodeType": "VariableDeclaration",
        "scope": 1292,
        "src": "5852:12:5",
        "stateVariable": false,
        "storageLocation": "default",

```

```

"typeDescriptions": {
  "typeIdentifier": "t_bool",
  "typeString": "bool"
},
"typeName": {
  "id": 1277,
  "name": "bool",
  "nodeType": "ElementaryTypeName",
  "src": "5852:4:5",
  "typeDescriptions": {
    "typeIdentifier": "t_bool",
    "typeString": "bool"
  }
},
"visibility": "internal"
},
{
  "constant": false,
  "id": 1280,
  "mutability": "mutable",
  "name": "returndata",
  "nameLocation": "5879:10:5",
  "nodeType": "VariableDeclaration",
  "scope": 1292,
  "src": "5866:23:5",
  "stateVariable": false,
  "storageLocation": "memory",
  "typeDescriptions": {
    "typeIdentifier": "t_bytes_memory_ptr",
    "typeString": "bytes"
  }
},
"typeName": {

```

```

    "id": 1279,
    "name": "bytes",
    "nodeType": "ElementaryTypeName",
    "src": "5866:5:5",
    "typeDescriptions": {
      "typeIdentifier": "t_bytes_storage_ptr",
      "typeString": "bytes"
    }
  },
  "visibility": "internal"
}
],
"initialValue": {
  "arguments": [
    {
      "id": 1283,
      "name": "data",
      "nodeType": "Identifier",
      "overloadedDeclarations": [],
      "referencedDeclaration": 1263,
      "src": "5911:4:5",
      "typeDescriptions": {
        "typeIdentifier": "t_bytes_memory_ptr",
        "typeString": "bytes memory"
      }
    }
  ]
},
"expression": {
  "argumentTypes": [
    {
      "typeIdentifier": "t_bytes_memory_ptr",
      "typeString": "bytes memory"
    }
  ]
}

```

```

    ],
    "expression": {
      "id": 1281,
      "name": "target",
      "nodeType": "Identifier",
      "overloadedDeclarations": [],
      "referencedDeclaration": 1261,
      "src": "5893:6:5",
      "typeDescriptions": {
        "typeIdentifier": "t_address",
        "typeString": "address"
      }
    },
    "id": 1282,
    "isConstant": false,
    "isLValue": false,
    "isPure": false,
    "IValueRequested": false,
    "memberName": "staticcall",
    "nodeType": "MemberAccess",
    "src": "5893:17:5",
    "typeDescriptions": {
      "typeIdentifier":
"t_function_barestaticcall_view$_t_bytes_memory_ptr_$returns$_t_bool_$_t_bytes_me
memory_ptr_$",
      "typeString": "function (bytes memory) view returns (bool,bytes
memory)"
    }
  },
  "id": 1284,
  "isConstant": false,
  "isLValue": false,
  "isPure": false,
  "kind": "functionCall",

```

```

    "IValueRequested": false,
    "names": [],
    "nodeType": "FunctionCall",
    "src": "5893:23:5",
    "tryCall": false,
    "typeDescriptions": {
      "typeIdentifier": "t_tuple$_t_bool_$_t_bytes_memory_ptr_$",
      "typeString": "tuple(bool,bytes memory)"
    }
  },
  "nodeType": "VariableDeclarationStatement",
  "src": "5851:65:5"
},
{
  "expression": {
    "arguments": [
      {
        "id": 1287,
        "name": "success",
        "nodeType": "Identifier",
        "overloadedDeclarations": [],
        "referencedDeclaration": 1278,
        "src": "5950:7:5",
        "typeDescriptions": {
          "typeIdentifier": "t_bool",
          "typeString": "bool"
        }
      }
    ],
    "nodeType": "FunctionCall",
    "src": "5950:7:5"
  },
  "id": 1288,
  "name": "returndata",
  "nodeType": "Identifier",
  "overloadedDeclarations": [],
  "referencedDeclaration": 1280,
  "src": "5950:7:5"
}

```

```

"src": "5959:10:5",
"typeDescriptions": {
  "typeIdentifier": "t_bytes_memory_ptr",
  "typeString": "bytes memory"
}
},
{
  "id": 1289,
  "name": "errorMessage",
  "nodeType": "Identifier",
  "overloadedDeclarations": [],
  "referencedDeclaration": 1265,
  "src": "5971:12:5",
  "typeDescriptions": {
    "typeIdentifier": "t_string_memory_ptr",
    "typeString": "string memory"
  }
}
],
"expression": {
  "argumentTypes": [
    {
      "typeIdentifier": "t_bool",
      "typeString": "bool"
    },
    {
      "typeIdentifier": "t_bytes_memory_ptr",
      "typeString": "bytes memory"
    },
    {
      "typeIdentifier": "t_string_memory_ptr",
      "typeString": "string memory"
    }
  ]
}

```

```

    ],
    "id": 1286,
    "name": "verifyCallResult",
    "nodeType": "Identifier",
    "overloadedDeclarations": [],
    "referencedDeclaration": 1376,
    "src": "5933:16:5",
    "typeDescriptions": {
      "typeIdentifier":
{
  "constant": false,
  "id": 1313,
  "mutability": "mutable",
  "name": "target",
  "nameLocation": "6601:6:5",
  "nodeType": "VariableDeclaration",
  "scope": 1345,
  "src": "6593:14:5",
  "stateVariable": false,
  "storageLocation": "default",
  "typeDescriptions": {
    "typeIdentifier": "t_address",
    "typeString": "address"
  },
  "typeName": {
    "id": 1312,
    "name": "address",
    "nodeType": "ElementaryTypeName",
    "src": "6593:7:5",
    "stateMutability": "nonpayable",
    "typeDescriptions": {
      "typeIdentifier": "t_address",
      "typeString": "address"
    }
  }
}

```

```

    },
    "visibility": "internal"
  },
  {
    "constant": false,
    "id": 1315,
    "mutability": "mutable",
    "name": "data",
    "nameLocation": "6630:4:5",
    "nodeType": "VariableDeclaration",
    "scope": 1345,
    "src": "6617:17:5",
    "stateVariable": false,
    "storageLocation": "memory",
    "typeDescriptions": {
      "typeIdentifier": "t_bytes_memory_ptr",
      "typeString": "bytes"
    },
    "typeName": {
      "id": 1314,
      "name": "bytes",
      "nodeType": "ElementaryTypeName",
      "src": "6617:5:5",
      "typeDescriptions": {
        "typeIdentifier": "t_bytes_storage_ptr",
        "typeString": "bytes"
      }
    },
    "visibility": "internal"
  },
  {
    "constant": false,
    "id": 1317,
    "mutability": "mutable",

```

```

    "name": "errorMessage",
    "nameLocation": "6658:12:5",
    "nodeType": "VariableDeclaration",
    "scope": 1345,
    "src": "6644:26:5",
    "stateVariable": false,
    "storageLocation": "memory",
    "typeDescriptions": {
      "typeIdentifier": "t_string_memory_ptr",
      "typeString": "string"
    },
    "typeName": {
      "id": 1316,
      "name": "string",
      "nodeType": "ElementaryTypeName",
      "src": "6644:6:5",
      "typeDescriptions": {
        "typeIdentifier": "t_string_storage_ptr",
        "typeString": "string"
      }
    },
    "visibility": "internal"
  }
],
"src": "6583:93:5"
},
"returnParameters": {
  "id": 1321,
  "nodeType": "ParameterList",
  "parameters": [
    {
      "constant": false,
      "id": 1320,
      "mutability": "mutable",

```

```

    "name": "",
    "nameLocation": "-1:-1:-1",
    "nodeType": "VariableDeclaration",
    "scope": 1345,
    "src": "6695:12:5",
    "stateVariable": false,
    "storageLocation": "memory",
    "typeDescriptions": {
      "typeIdentifier": "t_bytes_memory_ptr",
      "typeString": "bytes"
    },
    "typeName": {
      "id": 1319,
      "name": "bytes",
      "nodeType": "ElementaryTypeName",
      "src": "6695:5:5",
      "typeDescriptions": {
        "typeIdentifier": "t_bytes_storage_ptr",
        "typeString": "bytes"
      }
    },
    "visibility": "internal"
  }
],
"src": "6694:14:5"
},
"scope": 1377,
"src": "6554:387:5",
"stateMutability": "nonpayable",
"virtual": false,
"visibility": "internal"
},
{
  "body": {

```

```

"id": 1375,
"nodeType": "Block",
"src": "7321:532:5",
"statements": [
  {
    "condition": {
      "id": 1357,
      "name": "success",
      "nodeType": "Identifier",
      "overloadedDeclarations": [],
      "referencedDeclaration": 1348,
      "src": "7335:7:5",
      "typeDescriptions": {
        "typeIdentifier": "t_bool",
        "typeString": "bool"
      }
    }
  },
  "falseBody": {
    "id": 1373,
    "nodeType": "Block",
    "src": "7392:455:5",
    "statements": [
      {
        "condition": {
          "commonType": {
            "typeIdentifier": "t_uint256",
            "typeString": "uint256"
          },
        },
        "id": 1364,
        "isConstant": false,
        "isLValue": false,
        "isPure": false,
        "IValueRequested": false,
        "leftExpression": {

```

```

"expression": {
  "id": 1361,
  "name": "returndata",
  "nodeType": "Identifier",
  "overloadedDeclarations": [],
  "referencedDeclaration": 1350,
  "src": "7476:10:5",
  "typeDescriptions": {
    "typeIdentifier": "t_bytes_memory_ptr",
    "typeString": "bytes memory"
  }
},
"id": 1362,
"isConstant": false,
"isLValue": false,
"isPure": false,
"IValueRequested": false,
"memberName": "length",
"nodeType": "MemberAccess",
"src": "7476:17:5",
"typeDescriptions": {
  "typeIdentifier": "t_uint256",
  "typeString": "uint256"
}
},
"nodeType": "BinaryOperation",
"operator": ">",
"rightExpression": {
  "hexValue": "30",
  "id": 1363,
  "isConstant": false,
  "isLValue": false,
  "isPure": true,
  "kind": "number",

```

```

    "IValueRequested": false,
    "nodeType": "Literal",
    "src": "7496:1:5",
    "typeDescriptions": {
      "typeIdentifier": "t_rational_0_by_1",
      "typeString": "int_const 0"
    },
    "value": "0"
  },
  "src": "7476:21:5",
  "typeDescriptions": {
    "typeIdentifier": "t_bool",
    "typeString": "bool"
  }
},
"falseBody": {
  "id": 1371,
  "nodeType": "Block",
  "src": "7784:53:5",
  "statements": [
    {
      "expression": {
        "arguments": [
          {
            "id": 1368,
            "name": "errorMessage",
            "nodeType": "Identifier",
            "overloadedDeclarations": [],
            "referencedDeclaration": 1352,
            "src": "7809:12:5",
            "typeDescriptions": {
              "typeIdentifier": "t_string_memory_ptr",
              "typeString": "string memory"
            }
          }
        ]
      }
    }
  ]
}

```

```

    }
  ],
  "expression": {
    "argumentTypes": [
      {
        "typeIdentifier": "t_string_memory_ptr",
        "typeString": "string memory"
      }
    ],
    "id": 1367,
    "name": "revert",
    "nodeType": "Identifier",
    "overloadedDeclarations": [
      4294967277,
      4294967277
    ],
    "referencedDeclaration": 4294967277,
    "src": "7802:6:5",
    "typeDescriptions": {
      "typeIdentifier":
        "t_function_revert_pure$t_string_memory_ptr_$returns__$",
      "typeString": "function (string memory) pure"
    }
  },
  "id": 1369,
  "isConstant": false,
  "isLValue": false,
  "isPure": false,
  "kind": "functionCall",
  "IValueRequested": false,
  "names": [],
  "nodeType": "FunctionCall",
  "src": "7802:20:5",
  "tryCall": false,

```

```

    "typeDescriptions": {
      "typeIdentifier": "t_tuple$__$",
      "typeString": "tuple()"
    }
  },
  "id": 1370,
  "nodeType": "ExpressionStatement",
  "src": "7802:20:5"
}
]
},
"id": 1372,
"nodeType": "IfStatement",
"src": "7472:365:5",
"trueBody": {
  "id": 1366,
  "nodeType": "Block",
  "src": "7499:279:5",
  "statements": [
    {
      "AST": {
        "nodeType": "YulBlock",
        "src": "7619:145:5",
        "statements": [
          {
            "nodeType": "YulVariableDeclaration",
            "src": "7641:40:5",
            "value": {
              "arguments": [
                {
                  "name": "returndata",
                  "nodeType": "YulIdentifier",
                  "src": "7670:10:5"
                }
              ]
            }
          }
        ]
      }
    }
  ]
}

```

```

],
"functionName": {
  "name": "mload",
  "nodeType": "YulIdentifier",
  "src": "7664:5:5"
},
"nodeType": "YulFunctionCall",
"src": "7664:17:5"
},
"variables": [
  {
    "name": "returndata_size",
    "nodeType": "YulTypedName",
    "src": "7645:15:5",
    "type": ""
  }
]
},
{
"expression": {
"arguments": [
  {
"arguments": [
  {
    "kind": "number",
    "nodeType": "YulLiteral",
    "src": "7713:2:5",
    "type": "",
    "value": "32"
  },
  {
    "name": "returndata",
    "nodeType": "YulIdentifier",
    "src": "7717:10:5"

```

```

    }
  ],
  "functionName": {
    "name": "add",
    "nodeType": "YulIdentifier",
    "src": "7709:3:5"
  },
  "nodeType": "YulFunctionCall",
  "src": "7709:19:5"
},
{
  "name": "returndata_size",
  "nodeType": "YulIdentifier",
  "src": "7730:15:5"
}
],
"functionName": {
  "name": "revert",
  "nodeType": "YulIdentifier",
  "src": "7702:6:5"
},
"nodeType": "YulFunctionCall",
"src": "7702:44:5"
},
"nodeType": "YulExpressionStatement",
"src": "7702:44:5"
}
]
},
"evmVersion": "london",
"externalReferences": [
  {
    "declaration": 1350,
    "isOffset": false,

```

```

        "isSlot": false,
        "src": "7670:10:5",
        "valueSize": 1
    },
    {
        "declaration": 1350,
        "isOffset": false,
        "isSlot": false,
        "src": "7717:10:5",
        "valueSize": 1
    }
],
    "id": 1365,
    "nodeType": "InlineAssembly",
    "src": "7610:154:5"
}
]
}
}
]
},
    "id": 1374,
    "nodeType": "IfStatement",
    "src": "7331:516:5",
    "trueBody": {
        "id": 1360,
        "nodeType": "Block",
        "src": "7344:42:5",
        "statements": [
            {
                "expression": {
                    "id": 1358,
                    "name": "returndata",
                    "nodeType": "Identifier",

```

```

    "overloadedDeclarations": [],
    "referencedDeclaration": 1350,
    "src": "7365:10:5",
    "typeDescriptions": {
      "typeIdentifier": "t_bytes_memory_ptr",
      "typeString": "bytes memory"
    }
  },
  "functionReturnParameters": 1356,
  "id": 1359,
  "nodeType": "Return",
  "src": "7358:17:5"
}
]
}
}
]
},
"documentation": {
  "id": 1346,
  "nodeType": "StructuredDocumentation",
  "src": "6947:209:5",
  "text": " @dev Tool to verifies that a low level call was successful, and revert if
it wasn't, either by bubbling the\n revert reason using the provided one.\n _Available
since v4.3._"
},
"id": 1376,
"implemented": true,
"kind": "function",
"modifiers": [],
"name": "verifyCallResult",
"nameLocation": "7170:16:5",
"nodeType": "FunctionDefinition",
"parameters": {

```

```

"id": 1353,
"nodeType": "ParameterList",
"parameters": [
  {
    "constant": false,
    "id": 1348,
    "mutability": "mutable",
    "name": "success",
    "nameLocation": "7201:7:5",
    "nodeType": "VariableDeclaration",
    "scope": 1376,
    "src": "7196:12:5",
    "stateVariable": false,
    "storageLocation": "default",
    "typeDescriptions": {
      "typeIdentifier": "t_bool",
      "typeString": "bool"
    },
    "typeName": {
      "id": 1347,
      "name": "bool",
      "nodeType": "ElementaryTypeName",
      "src": "7196:4:5",
      "typeDescriptions": {
        "typeIdentifier": "t_bool",
        "typeString": "bool"
      }
    },
    "visibility": "internal"
  },
  {
    "constant": false,
    "id": 1350,
    "mutability": "mutable",

```

```

"name": "returndata",
"nameLocation": "7231:10:5",
"nodeType": "VariableDeclaration",
"scope": 1376,
"src": "7218:23:5",
"stateVariable": false,
"storageLocation": "memory",
"typeDescriptions": {
  "typeIdentifier": "t_bytes_memory_ptr",
  "typeString": "bytes"
},
"typeName": {
  "id": 1349,
  "name": "bytes",
  "nodeType": "ElementaryTypeName",
  "src": "7218:5:5",
  "typeDescriptions": {
    "typeIdentifier": "t_bytes_storage_ptr",
    "typeString": "bytes"
  }
},
"visibility": "internal"
},
{
  "constant": false,
  "id": 1352,
  "mutability": "mutable",
  "name": "errorMessage",
  "nameLocation": "7265:12:5",
  "nodeType": "VariableDeclaration",
  "scope": 1376,
  "src": "7251:26:5",
  "stateVariable": false,
  "storageLocation": "memory",

```

```

"typeDescriptions": {
  "typeIdentifier": "t_string_memory_ptr",
  "typeString": "string"
},
"typeName": {
  "id": 1351,
  "name": "string",
  "nodeType": "ElementaryTypeName",
  "src": "7251:6:5",
  "typeDescriptions": {
    "typeIdentifier": "t_string_storage_ptr",
    "typeString": "string"
  }
},
"visibility": "internal"
}
],
"src": "7186:97:5"
},
"returnParameters": {
  "id": 1356,
  "nodeType": "ParameterList",
  "parameters": [
    {
      "constant": false,
      "id": 1355,
      "mutability": "mutable",
      "name": "",
      "nameLocation": "-1:-1:-1",
      "nodeType": "VariableDeclaration",
      "scope": 1376,
      "src": "7307:12:5",
      "stateVariable": false,
      "storageLocation": "memory",

```

```

    "typeDescriptions": {
      "typeIdentifier": "t_bytes_memory_ptr",
      "typeString": "bytes"
    },
    "typeName": {
      "id": 1354,
      "name": "bytes",
      "nodeType": "ElementaryTypeName",
      "src": "7307:5:5",
      "typeDescriptions": {
        "typeIdentifier": "t_bytes_storage_ptr",
        "typeString": "bytes"
      }
    },
    "visibility": "internal"
  }
],
"src": "7306:14:5"
},
"scope": 1377,
"src": "7161:692:5",
"stateMutability": "pure",
"virtual": false,
"visibility": "internal"
}
],
"scope": 1378,
"src": "126:7729:5",
"usedErrors": []
}
],
"src": "33:7823:5"
},
"legacyAST": {

```

```

"absolutePath": "@openzeppelin/contracts/utils/Address.sol",
"exportedSymbols": {
  "Address": [
    1377
  ]
},
"id": 1378,
"license": "MIT",
"nodeType": "SourceUnit",
"nodes": [
  {
    "id": 1082,
    "literals": [
      "solidity",
      "^",
      "0.8",
      ".0"
    ],
    "nodeType": "PragmaDirective",
    "src": "33:23:5"
  },
  {
    "abstract": false,
    "baseContracts": [],
    "canonicalName": "Address",
    "contractDependencies": [],
    "contractKind": "library",
    "documentation": {
      "id": 1083,
      "nodeType": "StructuredDocumentation",
      "src": "58:67:5",
      "text": " @dev Collection of functions related to the address type"
    },
    "fullyImplemented": true,

```

```

"id": 1377,
"linearizedBaseContracts": [
  1377
],
"name": "Address",
"nameLocation": "134:7:5",
"nodeType": "ContractDefinition",
"nodes": [
  {
    "body": {
      "id": 1099,
      "nodeType": "Block",
      "src": "784:311:5",
      "statements": [
        {
          "assignments": [
            1092
          ],
          "declarations": [
            {
              "constant": false,
              "id": 1092,
              "mutability": "mutable",
              "name": "size",
              "nameLocation": "989:4:5",
              "nodeType": "VariableDeclaration",
              "scope": 1099,
              "src": "981:12:5",
              "stateVariable": false,
              "storageLocation": "default",
              "typeDescriptions": {
                "typeIdentifier": "t_uint256",
                "typeString": "uint256"
              }
            }
          ],
          "statements": [
            1092
          ]
        }
      ]
    }
  }
]

```

```

"typeName": {
  "id": 1091,
  "name": "uint256",
  "nodeType": "ElementaryTypeName",
  "src": "981:7:5",
  "typeDescriptions": {
    "typeIdentifier": "t_uint256",
    "typeString": "uint256"
  }
},
"visibility": "internal"
}
],
"id": 1093,
"nodeType": "VariableDeclarationStatement",
"src": "981:12:5"
},
{
  "AST": {
    "nodeType": "YulBlock",
    "src": "1012:52:5",
    "statements": [
      {
        "nodeType": "YulAssignment",
        "src": "1026:28:5",
        "value": {
          "arguments": [
            {
              "name": "account",
              "nodeType": "YulIdentifier",
              "src": "1046:7:5"
            }
          ],
          "nodeType": "YulIdentifier",
          "src": "1046:7:5"
        }
      ],
      "functionName": {

```

```

        "name": "extcodesize",
        "nodeType": "YulIdentifier",
        "src": "1034:11:5"
    },
    "nodeType": "YulFunctionCall",
    "src": "1034:20:5"
},
"variableNames": [
    {
        "name": "size",
        "nodeType": "YulIdentifier",
        "src": "1026:4:5"
    }
]
}
]
},
"evmVersion": "london",
"externalReferences": [
    {
        "declaration": 1086,
        "isOffset": false,
        "isSlot": false,
        "src": "1046:7:5",
        "valueSize": 1
    },
    {
        "declaration": 1092,
        "isOffset": false,
        "isSlot": false,
        "src": "1026:4:5",
        "valueSize": 1
    }
]
],

```

```

    "id": 1094,
    "nodeType": "InlineAssembly",
    "src": "1003:61:5"
  },
  {
    "expression": {
      "commonType": {
        "typeIdentifier": "t_uint256",
        "typeString": "uint256"
      },
      "id": 1097,
      "isConstant": false,
      "isLValue": false,
      "isPure": false,
      "IValueRequested": false,
      "leftExpression": {
        "id": 1095,
        "name": "size",
        "nodeType": "Identifier",
        "overloadedDeclarations": [],
        "referencedDeclaration": 1092,
        "src": "1080:4:5",
        "typeDescriptions": {
          "typeIdentifier": "t_uint256",
          "typeString": "uint256"
        }
      },
      "nodeType": "BinaryOperation",
      "operator": ">",
      "rightExpression": {
        "hexValue": "30",
        "id": 1096,
        "isConstant": false,
        "isLValue": false,

```

```

    "isPure": true,
    "kind": "number",
    "IValueRequested": false,
    "nodeType": "Literal",
    "src": "1087:1:5",
    "typeDescriptions": {
      "typeIdentifier": "t_rational_0_by_1",
      "typeString": "int_const 0"
    },
    "value": "0"
  },
  "src": "1080:8:5",
  "typeDescriptions": {
    "typeIdentifier": "t_bool",
    "typeString": "bool"
  }
},
"functionReturnParameters": 1090,
"id": 1098,
"nodeType": "Return",
"src": "1073:15:5"
}
]
},

```

```

"documentation": {
  "id": 1084,
  "nodeType": "StructuredDocumentation",
  "src": "148:565:5",

```

"text": " @dev Returns true if `account` is a contract.\n [IMPORTANT]\n\n====\n It is unsafe to assume that an address for which this function returns\n false is an externally-owned account (EOA) and not a contract.\n Among others, `isContract` will\n return false for the following\n types of addresses:\n - an externally-owned account\n - a contract in construction\n - an address where a contract will be created\n - an address\n where a contract lived, but was destroyed\n===="

```

},
"id": 1100,
"implemented": true,
"kind": "function",
"modifiers": [],
"name": "isContract",
"nameLocation": "727:10:5",
"nodeType": "FunctionDefinition",
"parameters": {
  "id": 1087,
  "nodeType": "ParameterList",
  "parameters": [
    {
      "constant": false,
      "id": 1086,
      "mutability": "mutable",
      "name": "account",
      "nameLocation": "746:7:5",
      "nodeType": "VariableDeclaration",
      "scope": 1100,
      "src": "738:15:5",
      "stateVariable": false,
      "storageLocation": "default",
      "typeDescriptions": {
        "typeIdentifier": "t_address",
        "typeString": "address"
      },
    },
  ],
  "typeName": {
    "id": 1085,
    "name": "address",
    "nodeType": "ElementaryTypeName",
    "src": "738:7:5",
    "stateMutability": "nonpayable",
    "typeDescriptions": {

```

```

        "typeIdentifier": "t_address",
        "typeString": "address"
    }
},
    "visibility": "internal"
}
],
"src": "737:17:5"
},
"returnParameters": {
    "id": 1090,
    "nodeType": "ParameterList",
    "parameters": [
        {
            "constant": false,
            "id": 1089,
            "mutability": "mutable",
            "name": "",
            "nameLocation": "-1:-1:-1",
            "nodeType": "VariableDeclaration",
            "scope": 1100,
            "src": "778:4:5",
            "stateVariable": false,
            "storageLocation": "default",
            "typeDescriptions": {
                "typeIdentifier": "t_bool",
                "typeString": "bool"
            },
        },
        "typeName": {
            "id": 1088,
            "name": "bool",
            "nodeType": "ElementaryTypeName",
            "src": "778:4:5",
            "typeDescriptions": {

```

```

        "typeIdentifier": "t_bool",
        "typeString": "bool"
    }
},
    "visibility": "internal"
}
],
"src": "777:6:5"
},
"scope": 1377,
"src": "718:377:5",
"stateMutability": "view",
"virtual": false,
"visibility": "internal"
},
{
    "body": {
        "id": 1133,
        "nodeType": "Block",
        "src": "2083:241:5",
        "statements": [
            {
                "expression": {
                    "arguments": [
                        {
                            "commonType": {
                                "typeIdentifier": "t_uint256",
                                "typeString": "uint256"
                            },
                            "id": 1115,
                            "isConstant": false,
                            "isLValue": false,
                            "isPure": false,
                            "IValueRequested": false,

```

```

"leftExpression": {
  "expression": {
    "arguments": [
      {
        "id": 1111,
        "name": "this",
        "nodeType": "Identifier",
        "overloadedDeclarations": [],
        "referencedDeclaration": 4294967268,
        "src": "2109:4:5",
        "typeDescriptions": {
          "typeIdentifier": "t_contract$_Address_$1377",
          "typeString": "library Address"
        }
      }
    ],
    "expression": {
      "argumentTypes": [
        {
          "typeIdentifier": "t_contract$_Address_$1377",
          "typeString": "library Address"
        }
      ],
      "id": 1110,
      "isConstant": false,
      "isLValue": false,
      "isPure": true,
      "IValueRequested": false,
      "nodeType": "ElementaryTypeNameExpression",
      "src": "2101:7:5",
      "typeDescriptions": {
        "typeIdentifier": "t_type$_t_address_$",
        "typeString": "type(address)"
      }
    },

```

```

"typeName": {
  "id": 1109,
  "name": "address",
  "nodeType": "ElementaryTypeName",
  "src": "2101:7:5",
  "typeDescriptions": {}
}
},
"id": 1112,
"isConstant": false,
"isLValue": false,
"isPure": false,
"kind": "typeConversion",
"IValueRequested": false,
"names": [],
"nodeType": "FunctionCall",
"src": "2101:13:5",
"tryCall": false,
"typeDescriptions": {
  "typeIdentifier": "t_address",
  "typeString": "address"
}
},
"id": 1113,
"isConstant": false,
"isLValue": false,
"isPure": false,
"IValueRequested": false,
"memberName": "balance",
"nodeType": "MemberAccess",
"src": "2101:21:5",
"typeDescriptions": {
  "typeIdentifier": "t_uint256",
  "typeString": "uint256"
}

```

```

    }
  },
  "nodeType": "BinaryOperation",
  "operator": ">=",
  "rightExpression": {
    "id": 1114,
    "name": "amount",
    "nodeType": "Identifier",
    "overloadedDeclarations": [],
    "referencedDeclaration": 1105,
    "src": "2126:6:5",
    "typeDescriptions": {
      "typeIdentifier": "t_uint256",
      "typeString": "uint256"
    }
  },
  "src": "2101:31:5",
  "typeDescriptions": {
    "typeIdentifier": "t_bool",
    "typeString": "bool"
  }
},
{
  "hexValue":
"416464726573733a20696e737566666696369656e742062616c616e6365",
  "id": 1116,
  "isConstant": false,
  "isLValue": false,
  "isPure": true,
  "kind": "string",
  "lValueRequested": false,
  "nodeType": "Literal",
  "src": "2134:31:5",
  "typeDescriptions": {

```

```

        "typeIdentifier":
"t_stringliteral_5597a22abd0ef5332f8053862eb236db7590f17e2b93a53f63a103becfb56
1f9",
        "typeString": "literal_string \"Address: insufficient balance\""
    },
    "value": "Address: insufficient balance"
}
],
"expression": {
    "argumentTypes": [
        {
            "typeIdentifier": "t_bool",
            "typeString": "bool"
        },
        {
            "typeIdentifier":
"t_stringliteral_5597a22abd0ef5332f8053862eb236db7590f17e2b93a53f63a103becfb56
1f9",
            "typeString": "literal_string \"Address: insufficient balance\""
        }
    ],
    "id": 1108,
    "name": "require",
    "nodeType": "Identifier",
    "overloadedDeclarations": [
        4294967278,
        4294967278
    ],
    "referencedDeclaration": 4294967278,
    "src": "2093:7:5",
    "typeDescriptions": {
        "typeIdentifier":
"t_function_pure__$t_bool__$t_string_memory_ptr__$returns__$__$",
        "typeString": "function (bool,string memory) pure"
    }
}

```

```

    }
  },
  "id": 1117,
  "isConstant": false,
  "isLValue": false,
  "isPure": false,
  "kind": "functionCall",
  "IValueRequested": false,
  "names": [],
  "nodeType": "FunctionCall",
  "src": "2093:73:5",
  "tryCall": false,
  "typeDescriptions": {
    "typeIdentifier": "t_tuple$__$",
    "typeString": "tuple()"
  }
},
"id": 1118,
"nodeType": "ExpressionStatement",
"src": "2093:73:5"
},
{
  "assignments": [
    1120,
    null
  ],
  "declarations": [
    {
      "constant": false,
      "id": 1120,
      "mutability": "mutable",
      "name": "success",
      "nameLocation": "2183:7:5",
      "nodeType": "VariableDeclaration",

```

```

"scope": 1133,
"src": "2178:12:5",
"stateVariable": false,
"storageLocation": "default",
"typeDescriptions": {
  "typeIdentifier": "t_bool",
  "typeString": "bool"
},
"typeName": {
  "id": 1119,
  "name": "bool",
  "nodeType": "ElementaryTypeName",
  "src": "2178:4:5",
  "typeDescriptions": {
    "typeIdentifier": "t_bool",
    "typeString": "bool"
  }
},
"visibility": "internal"
},
null
],
"id": 1127,
"initialValue": {
  "arguments": [
    {
      "hexValue": "",
      "id": 1125,
      "isConstant": false,
      "isLValue": false,
      "isPure": true,
      "kind": "string",
      "lValueRequested": false,
      "nodeType": "Literal",

```

```

        "src": "2226:2:5",
        "typeDescriptions": {
          "typeIdentifier":
"t_stringliteral_c5d2460186f7233c927e7db2dcc703c0e500b653ca82273b7bfad8045d85
a470",
          "typeString": "literal_string \"\""
        },
        "value": ""
      }
    ],
    "expression": {
      "argumentTypes": [
        {
          "typeIdentifier":
"t_stringliteral_c5d2460186f7233c927e7db2dcc703c0e500b653ca82273b7bfad8045d85
a470",
          "typeString": "literal_string \"\""
        }
      ],
      "expression": {
        "argumentTypes": [
          {
            "typeIdentifier":
"t_stringliteral_c5d2460186f7233c927e7db2dcc703c0e500b653ca82273b7bfad8045d85
a470",
            "typeString": "literal_string \"\""
          }
        ],
        "id": 1121,
        "name": "recipient",
        "nodeType": "Identifier",
        "overloadedDeclarations": [],
        "referencedDeclaration": 1103,

```

```

"src": "2196:9:5",
"typeDescriptions": {
  "typeIdentifier": "t_address_payable",
  "typeString": "address payable"
}
},
"id": 1122,
"isConstant": false,
"isLValue": false,
"isPure": false,
"IValueRequested": false,
"memberName": "call",
"nodeType": "MemberAccess",
"src": "2196:14:5",
"typeDescriptions": {
  "typeIdentifier":
"t_function_barecall_payable$t_bytes_memory_ptr_returns$t_bool$t_bytes_mem
ory_ptr_$",
  "typeString": "function (bytes memory) payable returns (bool,bytes
memory)"
}
},
"id": 1124,
"isConstant": false,
"isLValue": false,
"isPure": false,
"IValueRequested": false,
"names": [
  "value"
],
"nodeType": "FunctionCallOptions",
"options": [
  {
    "id": 1123,

```

```

        "name": "amount",
        "nodeType": "Identifier",
        "overloadedDeclarations": [],
        "referencedDeclaration": 1105,
        "src": "2218:6:5",
        "typeDescriptions": {
          "typeIdentifier": "t_uint256",
          "typeString": "uint256"
        }
      },
      "src": "2196:29:5",
      "typeDescriptions": {
        "typeIdentifier":
"t_function_barecall_payable$_t_bytes_memory_ptr_$returns$_t_bool_$_t_bytes_memory_ptr_$value",
        "typeString": "function (bytes memory) payable returns (bool,bytes memory)"
      }
    },
    "id": 1126,
    "isConstant": false,
    "isLValue": false,
    "isPure": false,
    "kind": "functionCall",
    "IValueRequested": false,
    "names": [],
    "nodeType": "FunctionCall",
    "src": "2196:33:5",
    "tryCall": false,
    "typeDescriptions": {
      "typeIdentifier": "t_tuple$_t_bool_$_t_bytes_memory_ptr_$",
      "typeString": "tuple(bool,bytes memory)"
    }
  }

```

```

    },
    "nodeType": "VariableDeclarationStatement",
    "src": "2177:52:5"
  },
  {
    "expression": {
      "arguments": [
        {
          "id": 1129,
          "name": "success",
          "nodeType": "Identifier",
          "overloadedDeclarations": [],
          "referencedDeclaration": 1120,
          "src": "2247:7:5",
          "typeDescriptions": {
            "typeIdentifier": "t_bool",
            "typeString": "bool"
          }
        }
      ],
      "type": "bool"
    },
    "nodeType": "ExpressionStatement",
    "src": "2247:7:5"
  },
  {
    "hexValue":
"416464726573733a20756e616226c6520746f2073656e642076616c75652c20726563697
069656e74206d61792068617665207265766572746564",
    "id": 1130,
    "isConstant": false,
    "isLValue": false,
    "isPure": true,
    "kind": "string",
    "IValueRequested": false,
    "nodeType": "Literal",
    "src": "2256:60:5",
    "typeDescriptions": {

```

```

        "typeIdentifier":
"t_stringliteral_51ddaa38748c0a1144620fb5bfe8edab31ea437571ad591a7734bbfd0429a
eae",
        "typeString": "literal_string \"Address: unable to send value, recipient
may have reverted\""
    },
    "value": "Address: unable to send value, recipient may have reverted"
}
],
"expression": {
    "argumentTypes": [
        {
            "typeIdentifier": "t_bool",
            "typeString": "bool"
        },
        {
            "typeIdentifier":
"t_stringliteral_51ddaa38748c0a1144620fb5bfe8edab31ea437571ad591a7734bbfd0429a
eae",
            "typeString": "literal_string \"Address: unable to send value, recipient
may have reverted\""
        }
    ],
    "id": 1128,
    "name": "require",
    "nodeType": "Identifier",
    "overloadedDeclarations": [
        4294967278,
        4294967278
    ],
    "referencedDeclaration": 4294967278,
    "src": "2239:7:5",
    "typeDescriptions": {
        "typeIdentifier":

```

```

    "t_function_internal_nonpayable$_t_address_$_t_bytes_memory_ptr_$_t_uint256_$_t_s
tring_memory_ptr_$returns$_t_bytes_memory_ptr_$",
        "typeString": "function (address,bytes memory,uint256,string memory)
returns (bytes memory)"
    }
},
    "id": 1188,
    "isConstant": false,
    "isLValue": false,
    "isPure": false,
    "kind": "functionCall",
    "IValueRequested": false,
    "names": [],
    "nodeType": "FunctionCall",
    "src": "4206:87:5",
    "tryCall": false,
    "typeDescriptions": {
        "typeIdentifier": "t_bytes_memory_ptr",
        "typeString": "bytes memory"
    }
},
    "functionReturnParameters": 1182,
    "id": 1189,
    "nodeType": "Return",
    "src": "4199:94:5"
}
]
},
"documentation": {
    "id": 1172,
    "nodeType": "StructuredDocumentation",
    "src": "3690:351:5",
    "text": " @dev Same as {xref-Address-functionCall-address-bytes-
}[^functionCall],\n but also transferring `value` wei to `target`.\n Requirements:\n - the

```

calling contract must have an ETH balance of at least `value`.\n - the called Solidity function must be `payable`.\n _Available since v3.1._"

```
},
"id": 1191,
"implemented": true,
"kind": "function",
"modifiers": [],
"name": "functionCallWithValue",
"nameLocation": "4055:21:5",
"nodeType": "FunctionDefinition",
"parameters": {
  "id": 1179,
  "nodeType": "ParameterList",
  "parameters": [
    {
      "constant": false,
      "id": 1174,
      "mutability": "mutable",
      "name": "target",
      "nameLocation": "4094:6:5",
      "nodeType": "VariableDeclaration",
      "scope": 1191,
      "src": "4086:14:5",
      "stateVariable": false,
      "storageLocation": "default",
      "typeDescriptions": {
        "typeIdentifier": "t_address",
        "typeString": "address"
      },
    },
  ],
  "typeName": {
    "id": 1173,
    "name": "address",
    "nodeType": "ElementaryTypeName",
    "src": "4086:7:5",
```

```

"stateMutability": "nonpayable",
"typeDescriptions": {
  "typeIdentifier": "t_address",
  "typeString": "address"
}
},
"visibility": "internal"
},
{
"constant": false,
"id": 1176,
"mutability": "mutable",
"name": "data",
"nameLocation": "4123:4:5",
"nodeType": "VariableDeclaration",
"scope": 1191,
"src": "4110:17:5",
"stateVariable": false,
"storageLocation": "memory",
"typeDescriptions": {
  "typeIdentifier": "t_bytes_memory_ptr",
  "typeString": "bytes"
},
"typeName": {
  "id": 1175,
  "name": "bytes",
  "nodeType": "ElementaryTypeName",
  "src": "4110:5:5",
  "typeDescriptions": {
    "typeIdentifier": "t_bytes_storage_ptr",
    "typeString": "bytes"
  }
},
"visibility": "internal"

```

```

    },
    {
      "constant": false,
      "id": 1178,
      "mutability": "mutable",
      "name": "value",
      "nameLocation": "4145:5:5",
      "nodeType": "VariableDeclaration",
      "scope": 1191,
      "src": "4137:13:5",
      "stateVariable": false,
      "storageLocation": "default",
      "typeDescriptions": {
        "typeIdentifier": "t_uint256",
        "typeString": "uint256"
      },
      "typeName": {
        "id": 1177,
        "name": "uint256",
        "nodeType": "ElementaryTypeName",
        "src": "4137:7:5",
        "typeDescriptions": {
          "typeIdentifier": "t_uint256",
          "typeString": "uint256"
        }
      },
      "visibility": "internal"
    }
  ],
  "src": "4076:80:5"
},
"returnParameters": {
  "id": 1182,
  "nodeType": "ParameterList",

```

```

"parameters": [
  {
    "constant": false,
    "id": 1181,
    "mutability": "mutable",
    "name": "",
    "nameLocation": "-1:-1:-1",
    "nodeType": "VariableDeclaration",
    "scope": 1191,
    "src": "4175:12:5",
    "stateVariable": false,
    "storageLocation": "memory",
    "typeDescriptions": {
      "typeIdentifier": "t_bytes_memory_ptr",
      "typeString": "bytes"
    },
    "typeName": {
      "id": 1180,
      "name": "bytes",
      "nodeType": "ElementaryTypeName",
      "src": "4175:5:5",
      "typeDescriptions": {
        "typeIdentifier": "t_bytes_storage_ptr",
        "typeString": "bytes"
      }
    },
    "visibility": "internal"
  }
],
"src": "4174:14:5"
},
"scope": 1377,
"src": "4046:254:5",
"stateMutability": "nonpayable",

```

```

"virtual": false,
"visibility": "internal"
},
{
"body": {
" id": 1240,
"nodeType": "Block",
"src": "4727:320:5",
"statements": [
{
"expression": {
"arguments": [
{
"commonType": {
" typeIdentifier": "t_uint256",
" typeString": "uint256"
},
" id": 1212,
"isConstant": false,
"isLValue": false,
"isPure": false,
"lValueRequested": false,
"leftExpression": {
"expression": {
"arguments": [
{
" id": 1208,
"name": "this",
"nodeType": "Identifier",
"overloadedDeclarations": [],
"referencedDeclaration": 4294967268,
"src": "4753:4:5",
" typeDescriptions": {
" typeIdentifier": "t_contract$_Address_$1377",

```

```

        "typeString": "library Address"
    }
}
],
"expression": {
    "argumentTypes": [
        {
            "typeIdentifier": "t_contract$_Address_$1377",
            "typeString": "library Address"
        }
    ],
    "id": 1207,
    "isConstant": false,
    "isLValue": false,
    "isPure": true,
    "IValueRequested": false,
    "nodeType": "ElementaryTypeNameExpression",
    "src": "4745:7:5",
    "typeDescriptions": {
        "typeIdentifier": "t_type$_t_address_$",
        "typeString": "type(address)"
    },
    "typeName": {
        "id": 1206,
        "name": "address",
        "nodeType": "ElementaryTypeName",
        "src": "4745:7:5",
        "typeDescriptions": {}
    }
},
"id": 1209,
"isConstant": false,
"isLValue": false,
"isPure": false,

```

```

"kind": "typeConversion",
"IValueRequested": false,
"names": [],
"nodeType": "FunctionCall",
"src": "4745:13:5",
"tryCall": false,
"typeDescriptions": {
  "typeIdentifier": "t_address",
  "typeString": "address"
}
},
"id": 1210,
"isConstant": false,
"isLValue": false,
"isPure": false,
"IValueRequested": false,
"memberName": "balance",
"nodeType": "MemberAccess",
"src": "4745:21:5",
"typeDescriptions": {
  "typeIdentifier": "t_uint256",
  "typeString": "uint256"
}
},
"nodeType": "BinaryOperation",
"operator": ">=",
"rightExpression": {
  "id": 1211,
  "name": "value",
  "nodeType": "Identifier",
  "overloadedDeclarations": [],
  "referencedDeclaration": 1198,
  "src": "4770:5:5",
  "typeDescriptions": {

```

```

        "typeIdentifier": "t_uint256",
        "typeString": "uint256"
    }
},
"src": "4745:30:5",
"typeDescriptions": {
    "typeIdentifier": "t_bool",
    "typeString": "bool"
}
},
{
    "hexValue":
"416464726573733a20696e737566666696369656e742062616c616e6365206666f7220636
16c6c",
    "id": 1213,
    "isConstant": false,
    "isLValue": false,
    "isPure": true,
    "kind": "string",
    "IValueRequested": false,
    "nodeType": "Literal",
    "src": "4777:40:5",
    "typeDescriptions": {
        "typeIdentifier":
"t_stringliteral_565f1a77334fc4792800921178c71e4521acffab18ff9e7885b49377ee80ab
4c",
        "typeString": "literal_string \"Address: insufficient balance for call\""
    },
    "value": "Address: insufficient balance for call"
}
],
"expression": {
    "argumentTypes": [
        {

```

```

        "typeIdentifier": "t_bool",
        "typeString": "bool"
    },
    {
        "typeIdentifier":
"t_stringliteral_565f1a77334fc4792800921178c71e4521acffab18ff9e7885b49377ee80ab
4c",
        "typeString": "literal_string \"Address: insufficient balance for call\""
    }
],
"id": 1205,
"name": "require",
"nodeType": "Identifier",
"overloadedDeclarations": [
    4294967278,
    4294967278
],
"referencedDeclaration": 4294967278,
"src": "4737:7:5",
"typeDescriptions": {
    "typeIdentifier":
"t_function_require_pure$_t_bool$_t_string_memory_ptr_$returns$__$",
    "typeString": "function (bool,string memory) pure"
}
},
"id": 1214,
"isConstant": false,
"isLValue": false,
"isPure": false,
"kind": "functionCall",
"lValueRequested": false,
"names": [],
"nodeType": "FunctionCall",
"src": "4737:81:5",

```

```

    "tryCall": false,
    "typeDescriptions": {
      "typeIdentifier": "t_tuple$__$",
      "typeString": "tuple()"
    }
  },
  "id": 1215,
  "nodeType": "ExpressionStatement",
  "src": "4737:81:5"
},
{
  "expression": {
    "arguments": [
      {
        "arguments": [
          {
            "id": 1218,
            "name": "target",
            "nodeType": "Identifier",
            "overloadedDeclarations": [],
            "referencedDeclaration": 1194,
            "src": "4847:6:5",
            "typeDescriptions": {
              "typeIdentifier": "t_address",
              "typeString": "address"
            }
          }
        ]
      }
    ],
    "expression": {
      "argumentTypes": [
        {
          "typeIdentifier": "t_address",
          "typeString": "address"
        }
      ]
    }
  }
}

```

```

    ],
    "id": 1217,
    "name": "isContract",
    "nodeType": "Identifier",
    "overloadedDeclarations": [],
    "referencedDeclaration": 1100,
    "src": "4836:10:5",
    "typeDescriptions": {
      "typeIdentifier":
"t_function_internal_view$_t_address_$returns$_t_bool_$",
      "typeString": "function (address) view returns (bool)"
    }
  },
  "id": 1219,
  "isConstant": false,
  "isLValue": false,
  "isPure": false,
  "kind": "functionCall",
  "IValueRequested": false,
  "names": [],
  "nodeType": "FunctionCall",
  "src": "4836:18:5",
  "tryCall": false,
  "typeDescriptions": {
    "typeIdentifier": "t_bool",
    "typeString": "bool"
  }
},
{

```

```

    "hexValue":
functionbuyProperty(uint _propertyId) public payable {
  property memory _land= properties[_propertyId];
  address _seller = _land._addr;
  // Require that the land is for sale

```

```

require(!_land.sold);
    // Require that the buyer is not the seller
require(_seller != msg.sender);
    // Ensure price is correct
require(msg.value>= _land.price);
    // Update the product
    _land._addr = msg.sender;
    _land.sold = true;
properties[_propertyId] = _land;
    // Send money
payable(_seller).transfer(msg.value);
boolpd = false;
    // Transfer property ownership
    _land.timestamp = block.timestamp;
transactionCount++;
transactions[transactionCount] = property(transactionCount, _land.name, _land.location,
_land.holder_name, _land.lr_no, _land.holder_id, msg.sender, _land.price, pd,
block.timestamp);
emitLogSold(_propertyId);
    }

functionsellProperty (uint _propertyId) public {
property memory _land= properties[_propertyId];
    // require that the seller is the land owner
require(msg.sender == _land._addr);
    // change the owner of the land to the blockchain address
    _land.sold = false;

properties[_propertyId] = _land;
    }
}
}

this.registerLand = this.registerLand.bind(this);
this.purchaseLand = this.purchaseLand.bind(this);

```

```

this.sellProperty = this.sellProperty.bind(this);
    }

    asynccomponentWillMount() {
    await this.loadWeb3()
    awaitthis.loadBlockchainData()
    }

    // Detects the presence of ethereum web provider
    async loadWeb3() {
    if (window.ethereum) {
        window.web3 = new Web3(window.ethereum)
    awaitwindow.ethereum.enable()
    }
    else if (window.web3) {
        window.web3 = new Web3(window.web3.currentProvider)
    }
    else {
    window.alert('Non-Ethereum browser detected. You should consider trying MetaMask!')
    }
    }

    // fetches accounts/nodes from blockchain
    asyncloadBlockchainData() {
    const web3 = window.web3
        // Load account
    const accounts = await web3.eth.getAccounts()
    this.setState({ account: accounts[0] })
    constnetworkId = await web3.eth.net.getId()
    constnetworkData = Registry.networks[networkId]
    if(networkData) {
    const registry = web3.eth.Contract(Registry.abi, networkData.address)
    this.setState({ registry })
    consttransactionCount = await registry.methods.transactionCount().call()
    constpropCount = await registry.methods.propCount().call()

```

```

this.setState({propCount})
this.setState({transactionCount})
for (let i = 1; i<= propCount; i++) {
const property = await registry.methods.properties(i).call()
this.setState({
properties: [...this.state.properties, property]
})
}
},
{id": 1284,
"isConstant": false,
"isLValue": false,
"isPure": false,
"kind": "functionCall",
"IValueRequested": false,
"names": [],
"nodeType": "FunctionCall",
"src": "5893:23:5",
"tryCall": false,
"typeDescriptions": {
"typeIdentifier": "t_tuple$t_bool$t_bytes_memory_ptr_$",
"typeString": "tuple(bool,bytes memory)"
}
},
"nodeType": "VariableDeclarationStatement",
"src": "5851:65:5"
},
{
"expression": {
"arguments": [
{
"id": 1287,
"name": "success",
"nodeType": "Identifier",

```

```

"overloadedDeclarations": [],
"referencedDeclaration": 1278,
"src": "5950:7:5",
"typeDescriptions": {
  "typeIdentifier": "t_bool",
  "typeString": "bool"
}
},
{
  "id": 1288,
  "name": "returndata",
  "nodeType": "Identifier",
  "overloadedDeclarations": [],
  "referencedDeclaration": 1280,
  "src": "5959:10:5",
  "typeDescriptions": {
    "typeIdentifier": "t_bytes_memory_ptr",
    "typeString": "bytes memory"
  }
},
{
  "id": 1289,
  "name": "errorMessage",
  "nodeType": "Identifier",
  "overloadedDeclarations": [],
  "referencedDeclaration": 1265,
  "src": "5971:12:5",
  "typeDescriptions": {
    "typeIdentifier": "t_string_memory_ptr",
    "typeString": "string memory"
  }
}
],
"expression": {

```

```

"argumentTypes": [
  {
    "typeIdentifier": "t_bool",
    "typeString": "bool"
  },
  {
    "typeIdentifier": "t_bytes_memory_ptr",
    "typeString": "bytes memory"
  },
  {
    "typeIdentifier": "t_string_memory_ptr",
    "typeString": "string memory"
  }
],
"id": 1286,
"name": "verifyCallResult",
"nodeType": "Identifier",
"overloadedDeclarations": [],
"referencedDeclaration": 1376,
"src": "5933:16:5",
"typeDescriptions": {
  "typeIdentifier":
{
  "constant": false,
  "id": 1313,
  "mutability": "mutable",
  "name": "target",
  "nameLocation": "6601:6:5",
  "nodeType": "VariableDeclaration",
  "scope": 1345,
  "src": "6593:14:5",
  "stateVariable": false,
  "storageLocation": "default",
  "typeDescriptions": {

```

```

    "typeIdentifier": "t_address",
    "typeString": "address"
  },
  "typeName": {
    "id": 1312,
    "name": "address",
    "nodeType": "ElementaryTypeName",
    "src": "6593:7:5",
    "stateMutability": "nonpayable",
    "typeDescriptions": {
      "typeIdentifier": "t_address",
      "typeString": "address"
    }
  },
  "visibility": "internal"
},
{
  "constant": false,
  "id": 1315,
  "mutability": "mutable",
  "name": "data",
  "nameLocation": "6630:4:5",
  "nodeType": "VariableDeclaration",
  "scope": 1345,
  "src": "6617:17:5",
  "stateVariable": false,
  "storageLocation": "memory",
  "typeDescriptions": {
    "typeIdentifier": "t_bytes_memory_ptr",
    "typeString": "bytes"
  },
  "typeName": {
    "id": 1314,
    "name": "bytes",

```

```

"nodeType": "ElementaryTypeName",
"src": "6617:5:5",
"typeDescriptions": {
  "typeIdentifier": "t_bytes_storage_ptr",
  "typeString": "bytes"
}
},
"visibility": "internal"
},
{
  "constant": false,
  "id": 1317,
  "mutability": "mutable",
  "name": "errorMessage",
  "nameLocation": "6658:12:5",
  "nodeType": "VariableDeclaration",
  "scope": 1345,
  "src": "6644:26:5",
  "stateVariable": false,
  "storageLocation": "memory",
  "typeDescriptions": {
    "typeIdentifier": "t_string_memory_ptr",
    "typeString": "string"
  },
  "typeName": {
    "id": 1316,
    "name": "string",
    "nodeType": "ElementaryTypeName",
    "src": "6644:6:5",
    "typeDescriptions": {
      "typeIdentifier": "t_string_storage_ptr",
      "typeString": "string"
    }
  },
},

```

```

    "visibility": "internal"
  }
],
"src": "6583:93:5"
},
"returnParameters": {
  "id": 1321,
  "nodeType": "ParameterList",
  "parameters": [
    {
      "constant": false,
      "id": 1320,
      "mutability": "mutable",
      "name": "",
      "nameLocation": "-1:-1:-1",
      "nodeType": "VariableDeclaration",
      "scope": 1345,
      "src": "6695:12:5",
      "stateVariable": false,
      "storageLocation": "memory",
      "typeDescriptions": {
        "typeIdentifier": "t_bytes_memory_ptr",
        "typeString": "bytes"
      },
    },
    {
      "id": 1319,
      "name": "bytes",
      "nodeType": "ElementaryTypeName",
      "src": "6695:5:5",
      "typeDescriptions": {
        "typeIdentifier": "t_bytes_storage_ptr",
        "typeString": "bytes"
      },
    }
  ],
}

```

```

    "visibility": "internal"
  }
],
"src": "6694:14:5"
},
"scope": 1377,
"src": "6554:387:5",
"stateMutability": "nonpayable",
"virtual": false,
"visibility": "internal"
},
{
  "body": {
    "id": 1375,
    "nodeType": "Block",
    "src": "7321:532:5",
    "statements": [
      {
        "condition": {
          "id": 1357,
          "name": "success",
          "nodeType": "Identifier",
          "overloadedDeclarations": [],
          "referencedDeclaration": 1348,
          "src": "7335:7:5",
          "typeDescriptions": {
            "typeIdentifier": "t_bool",
            "typeString": "bool"
          }
        }
      },
      {
        "falseBody": {
          "id": 1373,
          "nodeType": "Block",
          "src": "7392:455:5",

```

```

"statements": [
  {
    "condition": {
      "commonType": {
        "typeIdentifier": "t_uint256",
        "typeString": "uint256"
      },
    },
    "id": 1350,
    "isConstant": false,
    "isLValue": false,
    "isPure": false,
    "IValueRequested": false,
    "leftExpression": {
      "expression": {
        "id": 1361,
        "name": "returndata",
        "nodeType": "Identifier",
        "overloadedDeclarations": [],
        "referencedDeclaration": 1350,
        "src": "7476:10:5",
        "typeDescriptions": {
          "typeIdentifier": "t_bytes_memory_ptr",
          "typeString": "bytes memory"
        }
      }
    },
  },
  "id": 1352,
  "isConstant": false,
  "isLValue": false,
  "isPure": false,
  "IValueRequested": false,
  "memberName": "length",
  "nodeType": "MemberAccess",
  "src": "7476:17:5",
  "typeDescriptions": {

```

```

    "typeIdentifier": "t_uint256",
    "typeString": "uint256"
  }
},
"nodeType": "BinaryOperation",
"operator": ">",
"rightExpression": {
  "hexValue": "30",
  "id": 1363,
  "isConstant": false,
  "isLValue": false,
  "isPure": true,
  "kind": "number",
  "IValueRequested": false,
  "nodeType": "Literal",
  "src": "7496:1:5",
  "typeDescriptions": {
    "typeIdentifier": "t_rational_0_by_1",
    "typeString": "int_const 0"
  },
  "value": "0"
},
"src": "7476:21:5",
"typeDescriptions": {
  "typeIdentifier": "t_bool",
  "typeString": "bool"
}
},
>falseBody": {
  "id": 1371,
  "nodeType": "Block",
  "src": "7784:53:5",
  "statements": [
    {

```

```

"expression": {
  "arguments": [
    {
      "id": 1368,
      "name": "errorMessage",
      "nodeType": "Identifier",
      "overloadedDeclarations": [],
      "referencedDeclaration": 1352,
      "src": "7809:12:5",
      "typeDescriptions": {
        "typeIdentifier": "t_string_memory_ptr",
        "typeString": "string memory"
      }
    }
  ],
  "expression": {
    "argumentTypes": [
      {
        "typeIdentifier": "t_string_memory_ptr",
        "typeString": "string memory"
      }
    ],
    "id": 1352,
    "name": "revert",
    "nodeType": "Identifier",
    "overloadedDeclarations": [
      4294967277,
      4294967277
    ],
    "referencedDeclaration": 4294967277,
    "src": "7802:6:5",
    "typeDescriptions": {
      "typeIdentifier":
"416464726573733a2063616c6c20746f206e6f6e2d636f6e7472616374",

```

```

    "id": 1220,
    "isConstant": false,
    "isLValue": false,
    "isPure": true,
    "kind": "string",
    "IValueRequested": false,
    "nodeType": "Literal",
    "src": "4856:31:5",
    "typeDescriptions": {
      "typeIdentifier":
"t_stringliteral_cc2e4e38850b7c0a3e942cfed89b71c77302df25bcb2ec297a0c4ff9ff6b90
ad",
      "typeString": "literal_string \"Address: call to non-contract\""
    },
    "value": "Address: call to non-contract"
  }
],
"expression": {
  "argumentTypes": [
    {
      "typeIdentifier": "t_bool",
      "typeString": "bool"
    },
    {
      "typeIdentifier":
"t_stringliteral_cc2e4e38850b7c0a3e942cfed89b71c77302df25bcb2ec297a0c4ff9ff6b90
ad",
      "typeString": "literal_string \"Address: call to non-contract\""
    }
  ],
  "id": 1216,
  "name": "require",
  "nodeType": "Identifier",
  "overloadedDeclarations": [

```

```

    4294967278,
    4294967278
  ],
  "referencedDeclaration": 4294967278,
  "src": "4828:7:5",
  "typeDescriptions": {
    "typeIdentifier":
"t_function_require_pure$_t_bool_$_t_string_memory_ptr_$returns$__$",
    "typeString": "function (bool,string memory) pure"
  }
},
  "id": 1221,
  "isConstant": false,
  "isLValue": false,
  "isPure": false,
  "kind": "functionCall",
  "IValueRequested": false,
  "names": [],
  "nodeType": "FunctionCall",
  "src": "4828:60:5",
  "tryCall": false,
  "typeDescriptions": {
    "typeIdentifier": "t_tuple$__$",
    "typeString": "tuple()"
  }
},
  "id": 1222,
  "nodeType": "ExpressionStatement",
  "src": "4828:60:5"
},
{
  "assignments": [
    1224,
    1226

```

```

],
"declarations": [
  {
    "constant": false,
    "id": 1224,
    "mutability": "mutable",
    "name": "success",
    "nameLocation": "4905:7:5",
    "nodeType": "VariableDeclaration",
    "scope": 1240,
    "src": "4900:12:5",
    "stateVariable": false,
    "storageLocation": "default",
    "typeDescriptions": {
      "typeIdentifier": "t_bool",
      "typeString": "bool"
    },
    "typeName": {
      "id": 1223,
      "name": "bool",
      "nodeType": "ElementaryTypeName",
      "src": "4900:4:5",
      "typeDescriptions": {
        "typeIdentifier": "t_bool",
        "typeString": "bool"
      }
    },
    "visibility": "internal"
  },
  {
    "constant": false,
    "id": 1226,
    "mutability": "mutable",
    "name": "returndata",

```

```

"nameLocation": "4927:10:5",
"nodeType": "VariableDeclaration",
"scope": 1240,
"src": "4914:23:5",
"stateVariable": false,
"storageLocation": "memory",
"typeDescriptions": {
  "typeIdentifier": "t_bytes_memory_ptr",
  "typeString": "bytes"
},
"typeName": {
  "id": 1225,
  "name": "bytes",
  "nodeType": "ElementaryTypeName",
  "src": "4914:5:5",
  "typeDescriptions": {
    "typeIdentifier": "t_bytes_storage_ptr",
    "typeString": "bytes"
  }
},
"visibility": "internal"
}
],
"id": 1233,
"initialValue": {
  "arguments": [
    {
      "id": 1231,
      "name": "data",
      "nodeType": "Identifier",
      "overloadedDeclarations": [],
      "referencedDeclaration": 1196,
      "src": "4967:4:5",
      "typeDescriptions": {

```

```

    "typeIdentifier": "t_bytes_memory_ptr",
    "typeString": "bytes memory"
  }
}
],
"expression": {
  "argumentTypes": [
    {
      "typeIdentifier": "t_bytes_memory_ptr",
      "typeString": "bytes memory"
    }
  ],
"expression": {
  "argumentTypes": [
    {
      "typeIdentifier": "t_bytes_memory_ptr",
      "typeString": "bytes memory"
    }
  ],
"expression": {
  "id": 1227,
  "name": "target",
  "nodeType": "Identifier",
  "overloadedDeclarations": [],
  "referencedDeclaration": 1194,
  "src": "4941:6:5",
  "typeDescriptions": {
    "typeIdentifier": "t_address",
    "typeString": "address"
  }
},
"expression": {
  "id": 1228,
  "isConstant": false,

```

```

    "isLValue": false,
    "isPure": false,
    "IValueRequested": false,
    "memberName": "call",
    "nodeType": "MemberAccess",
    "src": "4941:11:5",
    "typeDescriptions": {
      "typeIdentifier":
"t_function_barecall_payable$_t_bytes_memory_ptr_$returns$_t_bool_$_t_bytes_memory_ptr_$",
      "typeString": "function (bytes memory) payable returns (bool,bytes
memory)"
    }
  },
  "id": 1230,
  "isConstant": false,
  "isLValue": false,
  "isPure": false,
  "IValueRequested": false,
  "names": [
    "value"
  ],
  "nodeType": "FunctionCallOptions",
  "options": [
    {
      "id": 1229,
      "name": "value",
      "nodeType": "Identifier",
      "overloadedDeclarations": [],
      "referencedDeclaration": 1198,
      "src": "4960:5:5",
      "typeDescriptions": {
        "typeIdentifier": "t_uint256",
        "typeString": "uint256"
      }
    }
  ]
}

```

```

    }
  }
],
"src": "4941:25:5",
"typeDescriptions": {
  "typeIdentifier":
"t_function_barecall_payable$t_bytes_memory_ptr_$returns$t_bool_$t_bytes_memory_ptr_$value",
  "typeString": "function (bytes memory) payable returns (bool,bytes memory)"
}
},
"id": 1232,
"isConstant": false,
"isLValue": false,
"isPure": false,
"kind": "functionCall",
"IValueRequested": false,
"names": [],
"nodeType": "FunctionCall",
"src": "4941:31:5",
"tryCall": false,
"typeDescriptions": {
  "typeIdentifier": "t_tuple$t_bool_$t_bytes_memory_ptr_$",
  "typeString": "tuple(bool,bytes memory)"
}
},
"nodeType": "VariableDeclarationStatement",
"src": "4899:73:5"
},
{
  "expression": {
    "arguments": [
      {

```

```

    "id": 1235,
    "name": "success",
    "nodeType": "Identifier",
    "overloadedDeclarations": [],
    "referencedDeclaration": 1224,
    "src": "5006:7:5",
    "typeDescriptions": {
      "typeIdentifier": "t_bool",
      "typeString": "bool"
    }
  },
  {
    "id": 1236,
    "name": "returndata",
    "nodeType": "Identifier",
    "overloadedDeclarations": [],
    "referencedDeclaration": 1226,
    "src": "5015:10:5",
    "typeDescriptions": {
      "typeIdentifier": "t_bytes_memory_ptr",
      "typeString": "bytes memory"
    }
  },
  {
    "id": 1237,
    "name": "errorMessage",
    "nodeType": "Identifier",
    "overloadedDeclarations": [],
    "referencedDeclaration": 1200,
    "src": "5027:12:5",
    "typeDescriptions": {
      "typeIdentifier": "t_string_memory_ptr",
      "typeString": "string memory"
    }
  }

```

```

    }
  ],
  "expression": {
    "argumentTypes": [
      {
        "typeIdentifier": "t_bool",
        "typeString": "bool"
      },
      {
        "typeIdentifier": "t_bytes_memory_ptr",
        "typeString": "bytes memory"
      },
      {
        "typeIdentifier": "t_string_memory_ptr",
        "typeString": "string memory"
      }
    ],
    "id": 1234,
    "name": "verifyCallResult",
    "nodeType": "Identifier",
    "overloadedDeclarations": [],
    "referencedDeclaration": 1376,
    "src": "4989:16:5",
    "typeDescriptions": {
      "typeIdentifier":
"t_function_internal_pure$t_bool_$_t_bytes_memory_ptr_$_t_string_memory_ptr_$ret
urns_$_t_bytes_memory_ptr_$",
      "typeString": "function (bool,bytes memory,string memory) pure returns
(bytes memory)"
    }
  },
  "id": 1238,
  "isConstant": false,
  "isLValue": false,

```

```

    "isPure": false,
    "kind": "functionCall",
    "IValueRequested": false,
    "names": [],
    "nodeType": "FunctionCall",
    "src": "4989:51:5",
    "tryCall": false,
    "typeDescriptions": {
      "typeIdentifier": "t_bytes_memory_ptr",
      "typeString": "bytes memory"
    }
  },
  "functionReturnParameters": 1204,
  "id": 1239,
  "nodeType": "Return",
  "src": "4982:58:5"
}
]
},
"documentation": {
  "id": 1192,
  "nodeType": "StructuredDocumentation",
  "src": "4306:237:5",
  "text": " @dev Same as {xref-Address-functionCallWithValue-address-bytes-uint256-}[`functionCallWithValue`], but\n with `errorMessage` as a fallback revert reason when `target` reverts.\n _Available since v3.1._"
},
"id": 1241,
"implemented": true,
"kind": "function",
"modifiers": [],
"name": "functionCallWithValue",
"nameLocation": "4557:21:5",
"nodeType": "FunctionDefinition",

```

```

"parameters": {
  "id": 1201,
  "nodeType": "ParameterList",
  "parameters": [
    {
      "constant": false,
      "id": 1194,
      "mutability": "mutable",
      "name": "target",
      "nameLocation": "4596:6:5",
      "nodeType": "VariableDeclaration",
      "scope": 1241,
      "src": "4588:14:5",
      "stateVariable": false,
      "storageLocation": "default",
      "typeDescriptions": {
        "typeIdentifier": "t_address",
        "typeString": "address"
      },
      "typeName": {
        "id": 1193,
        "name": "address",
        "nodeType": "ElementaryTypeName",
        "src": "4588:7:5",
        "stateMutability": "nonpayable",
        "typeDescriptions": {
          "typeIdentifier": "t_address",
          "typeString": "address"
        }
      },
      "visibility": "internal"
    },
    {
      "constant": false,

```

```

    "id": 1196,
    "mutability": "mutable",
    "name": "data",
    "nameLocation": "4625:4:5",
    "nodeType": "VariableDeclaration",
    "scope": 1241,
    "src": "4612:17:5",
    "stateVariable": false,
    "storageLocation": "memory",
    "typeDescriptions": {
      "typeIdentifier": "t_bytes_memory_ptr",
      "typeString": "bytes"
    },
    "typeName": {
      "id": 1195,
      "name": "bytes",
      "nodeType": "ElementaryTypeName",
      "src": "4612:5:5",
      "typeDescriptions": {
        "typeIdentifier": "t_bytes_storage_ptr",
        "typeString": "bytes"
      }
    },
    "visibility": "internal"
  },
  {
    "constant": false,
    "id": 1198,
    "mutability": "mutable",
    "name": "value",
    "nameLocation": "4647:5:5",
    "nodeType": "VariableDeclaration",
    "scope": 1241,
    "src": "4639:13:5",

```

```

"stateVariable": false,
"storageLocation": "default",
"typeDescriptions": {
  "typeIdentifier": "t_uint256",
  "typeString": "uint256"
},
"typeName": {
  "id": 1197,
  "name": "uint256",
  "nodeType": "ElementaryTypeName",
  "src": "4639:7:5",
  "typeDescriptions": {
    "typeIdentifier": "t_uint256",
    "typeString": "uint256"
  }
},
"visibility": "internal"
},
{
  "constant": false,
  "id": 1200,
  "mutability": "mutable",
  "name": "errorMessage",
  "nameLocation": "4676:12:5",
  "nodeType": "VariableDeclaration",
  "scope": 1241,
  "src": "4662:26:5",
  "stateVariable": false,
  "storageLocation": "memory",
  "typeDescriptions": {
    "typeIdentifier": "t_string_memory_ptr",
    "typeString": "string"
  },
  "typeName": {

```

```

    "id": 1199,
    "name": "string",
    "nodeType": "ElementaryTypeName",
    "src": "4662:6:5",
    "typeDescriptions": {
      "typeIdentifier": "t_string_storage_ptr",
      "typeString": "string"
    }
  },
  "visibility": "internal"
}
],
"src": "4578:116:5"
},
"returnParameters": {
  "id": 1204,
  "nodeType": "ParameterList",
  "parameters": [
    {
      "constant": false,
      "id": 1203,
      "mutability": "mutable",
      "name": "",
      "nameLocation": "-1:-1:-1",
      "nodeType": "VariableDeclaration",
      "scope": 1241,
      "src": "4713:12:5",
      "stateVariable": false,
      "storageLocation": "memory",
      "typeDescriptions": {
        "typeIdentifier": "t_bytes_memory_ptr",
        "typeString": "bytes"
      }
    },
    "typeName": {

```

```

    "id": 1202,
    "name": "bytes",
    "nodeType": "ElementaryTypeName",
    "src": "4713:5:5",
    "typeDescriptions": {
      "typeIdentifier": "t_bytes_storage_ptr",
      "typeString": "bytes"
    }
  },
  "visibility": "internal"
}
],
"src": "4712:14:5"
},
"scope": 1377,
"src": "4548:499:5",
"stateMutability": "nonpayable",
"virtual": false,
"visibility": "internal"
},
{
  "body": {
    "id": 1257,
    "nodeType": "Block",
    "src": "5324:97:5",
    "statements": [
      {
        "expression": {
          "arguments": [
            {
              "id": 1252,
              "name": "target",
              "nodeType": "Identifier",
              "overloadedDeclarations": [],

```

```

"referencedDeclaration": 1244,
"src": "5360:6:5",
"typeDescriptions": {
  "typeIdentifier": "t_address",
  "typeString": "address"
}
},
{
  "id": 1253,
  "name": "data",
  "nodeType": "Identifier",
  "overloadedDeclarations": [],
  "referencedDeclaration": 1246,
  "src": "5368:4:5",
  "typeDescriptions": {
    "typeIdentifier": "t_bytes_memory_ptr",
    "typeString": "bytes memory"
  }
},
{
  "hexValue":
"416464726573733a206c6f772d6c6576656c207374617469632063616c6c206661696c6
564",
  "id": 1254,
  "isConstant": false,
  "isLValue": false,
  "isPure": true,
  "kind": "string",
  "IValueRequested": false,
  "nodeType": "Literal",
  "src": "5374:39:5",
  "typeDescriptions": {

```

```

        "typeIdentifier":
"t_stringliteral_90ec82aa826a536a4cbfae44ecfa384680faa9a4b77344bce96aa761ad904d
f0",
        "typeString": "literal_string \"Address: low-level static call failed\""
    },
    "value": "Address: low-level static call failed"
}
],
"expression": {
    "argumentTypes": [
        {
            "typeIdentifier": "t_address",
            "typeString": "address"
        },
        {
            "typeIdentifier": "t_bytes_memory_ptr",
            "typeString": "bytes memory"
        },
        {
            "typeIdentifier":
"t_stringliteral_90ec82aa826a536a4cbfae44ecfa384680faa9a4b77344bce96aa761ad904d
f0",
            "typeString": "literal_string \"Address: low-level static call failed\""
        }
    ],
    "id": 1251,
    "name": "functionStaticCall",
    "nodeType": "Identifier",
    "overloadedDeclarations": [
        1258,
        1293
    ],
    "referencedDeclaration": 1293,
    "src": "5341:18:5",

```

```

        "typeDescriptions": {
            "typeIdentifier":
"t_function_internal_view$t_address_$t_bytes_memory_ptr_$t_string_memory_ptr_
$returnst_bytes_memory_ptr_$",
            "typeString": "function (address,bytes memory,string memory) view
returns (bytes memory)"
        }
    },
    "id": 1255,
    "isConstant": false,
    "isLValue": false,
    "isPure": false,
    "kind": "functionCall",
    "IValueRequested": false,
    "names": [],
    "nodeType": "FunctionCall",
    "src": "5341:73:5",
    "tryCall": false,
    "typeDescriptions": {
        "typeIdentifier": "t_bytes_memory_ptr",
        "typeString": "bytes memory"
    }
},
"functionReturnParameters": 1250,
"id": 1256,
"nodeType": "Return",
"src": "5334:80:5"
}
]
},
"documentation": {
    "id": 1242,
    "nodeType": "StructuredDocumentation",
    "src": "5053:166:5",

```

```

    "text": " @dev Same as {xref-Address-functionCall-address-bytes-
}[\`functionCall`],\n but performing a static call.\n _Available since v3.3._"
  },
  "id": 1258,
  "implemented": true,
  "kind": "function",
  "modifiers": [],
  "name": "functionStaticCall",
  "nameLocation": "5233:18:5",
  "nodeType": "FunctionDefinition",
  "parameters": {
    "id": 1247,
    "nodeType": "ParameterList",
    "parameters": [
      {
        "constant": false,
        "id": 1244,
        "mutability": "mutable",
        "name": "target",
        "nameLocation": "5260:6:5",
        "nodeType": "VariableDeclaration",
        "scope": 1258,
        "src": "5252:14:5",
        "stateVariable": false,
        "storageLocation": "default",
        "typeDescriptions": {
          "typeIdentifier": "t_address",
          "typeString": "address"
        },
      },
    ],
    "typeName": {
      "id": 1243,
      "name": "address",
      "nodeType": "ElementaryTypeName",
      "src": "5252:7:5",

```

```

"stateMutability": "nonpayable",
"typeDescriptions": {
  "typeIdentifier": "t_address",
  "typeString": "address"
}
},
"visibility": "internal"
},
{
  "constant": false,
  "id": 1246,
  "mutability": "mutable",
  "name": "data",
  "nameLocation": "5281:4:5",
  "nodeType": "VariableDeclaration",
  "scope": 1258,
  "src": "5268:17:5",
  "stateVariable": false,
  "storageLocation": "memory",
  "typeDescriptions": {
    "typeIdentifier": "t_bytes_memory_ptr",
    "typeString": "bytes"
  },
},
"typeName": {
  "id": 1245,
  "name": "bytes",
  "nodeType": "ElementaryTypeName",
  "src": "5268:5:5",
  "typeDescriptions": {
    "typeIdentifier": "t_bytes_storage_ptr",
    "typeString": "bytes"
  }
},
"visibility": "internal"

```

```

    }
  ],
  "src": "5251:35:5"
},
"returnParameters": {
  "id": 1250,
  "nodeType": "ParameterList",
  "parameters": [
    {
      "constant": false,
      "id": 1249,
      "mutability": "mutable",
      "name": "",
      "nameLocation": "-1:-1:-1",
      "nodeType": "VariableDeclaration",
      "scope": 1258,
      "src": "5310:12:5",
      "stateVariable": false,
      "storageLocation": "memory",
      "typeDescriptions": {
        "typeIdentifier": "t_bytes_memory_ptr",
        "typeString": "bytes"
      },
    },
    {
      "type": "ElementaryTypeName",
      "id": 1248,
      "name": "bytes",
      "nodeType": "ElementaryTypeName",
      "src": "5310:5:5",
      "typeDescriptions": {
        "typeIdentifier": "t_bytes_storage_ptr",
        "typeString": "bytes"
      },
    }
  ],
  "visibility": "internal"
}

```

```

    }
  ],
  "src": "5309:14:5"
},
"scope": 1377,
"src": "5224:197:5",
"stateMutability": "view",
"virtual": false,
"visibility": "internal"
},
{
  "body": {
    "id": 1292,
    "nodeType": "Block",
    "src": "5763:228:5",
    "statements": [
      {
        "expression": {
          "arguments": [
            {
              "arguments": [
                {
                  "id": 1272,
                  "name": "target",
                  "nodeType": "Identifier",
                  "overloadedDeclarations": [],
                  "referencedDeclaration": 1261,
                  "src": "5792:6:5",
                  "typeDescriptions": {
                    "typeIdentifier": "t_address",
                    "typeString": "address"
                  }
                }
              ]
            }
          ]
        }
      }
    ]
  },
  ],

```

```

"expression": {
  "argumentTypes": [
    {
      "typeIdentifier": "t_address",
      "typeString": "address"
    }
  ],
  "id": 1271,
  "name": "isContract",
  "nodeType": "Identifier",
  "overloadedDeclarations": [],
  "referencedDeclaration": 1100,
  "src": "5781:10:5",
  "typeDescriptions": {
    "typeIdentifier":
"t_function_internal_view$t_address_$returns$t_bool_$",
    "typeString": "function (address) view returns (bool)"
  }
},
"id": 1273,
"isConstant": false,
"isLValue": false,
"isPure": false,
"kind": "functionCall",
"IValueRequested": false,
"names": [],
"nodeType": "FunctionCall",
"src": "5781:18:5",
"tryCall": false,
"typeDescriptions": {
  "typeIdentifier": "t_bool",
  "typeString": "bool"
}
},

```

```

    {
      "hexValue":
"416464726573733a207374617469632063616c6c20746f206e6f6e2d636f6e7472616374
",
      "id": 1274,
      "isConstant": false,
      "isLValue": false,
      "isPure": true,
      "kind": "string",
      "IValueRequested": false,
      "nodeType": "Literal",
      "src": "5801:38:5",
      "typeDescriptions": {
        "typeIdentifier":
"t_stringliteral_c79cc78e4f16ce3933a42b84c73868f93bb4a59c031a0acf576679de98c60
8a9",
        "typeString": "literal_string \"Address: static call to non-contract\""
      },
      "value": "Address: static call to non-contract"
    }
  ],
  "expression": {
    "argumentTypes": [
      {
        "typeIdentifier": "t_bool",
        "typeString": "bool"
      },
      {
        "typeIdentifier":
"t_stringliteral_c79cc78e4f16ce3933a42b84c73868f93bb4a59c031a0acf576679de98c60
8a9",
        "typeString": "literal_string \"Address: static call to non-contract\""
      }
    ],

```

```

    "id": 1270,
    "name": "require",
    "nodeType": "Identifier",
    "overloadedDeclarations": [
      4294967278,
      4294967278
    ],
    "referencedDeclaration": 4294967278,
    "src": "5773:7:5",
    "typeDescriptions": {
      "typeIdentifier":
"t_function_require_pure$_t_bool_$_t_string_memory_ptr_$returns$__$",
      "typeString": "function (bool,string memory) pure"
    }
  },
  "id": 1275,
  "isConstant": false,
  "isLValue": false,
  "isPure": false,
  "kind": "functionCall",
  "IValueRequested": false,
  "names": [],
  "nodeType": "FunctionCall",
  "src": "5773:67:5",
  "tryCall": false,
  "typeDescriptions": {
    "typeIdentifier": "t_tuple$__$",
    "typeString": "tuple()"
  }
},
  "id": 1276,
  "nodeType": "ExpressionStatement",
  "src": "5773:67:5"
},

```

```

{
  "assignments": [
    1278,
    1280
  ],
  "declarations": [
    {
      "constant": false,
      "id": 1278,
      "mutability": "mutable",
      "name": "success",
      "nameLocation": "5857:7:5",
      "nodeType": "VariableDeclaration",
      "scope": 1292,
      "src": "5852:12:5",
      "stateVariable": false,
      "storageLocation": "default",
      "typeDescriptions": {
        "typeIdentifier": "t_bool",
        "typeString": "bool"
      },
      "typeName": {
        "id": 1277,
        "name": "bool",
        "nodeType": "ElementaryTypeName",
        "src": "5852:4:5",
        "typeDescriptions": {
          "typeIdentifier": "t_bool",
          "typeString": "bool"
        }
      },
      "visibility": "internal"
    },
  ],
}

```

```

    "constant": false,
    "id": 1280,
    "mutability": "mutable",
    "name": "returndata",
    "nameLocation": "5879:10:5",
    "nodeType": "VariableDeclaration",
    "scope": 1292,
    "src": "5866:23:5",
    "stateVariable": false,
    "storageLocation": "memory",
    "typeDescriptions": {
      "typeIdentifier": "t_bytes_memory_ptr",
      "typeString": "bytes"
    },
    "typeName": {
      "id": 1279,
      "name": "bytes",
      "nodeType": "ElementaryTypeName",
      "src": "5866:5:5",
      "typeDescriptions": {
        "typeIdentifier": "t_bytes_storage_ptr",
        "typeString": "bytes"
      }
    },
    "visibility": "internal"
  }
],
  "id": 1285,
  "initialValue": {
    "arguments": [
      {
        "id": 1283,
        "name": "data",
        "nodeType": "Identifier",

```

```

"overloadedDeclarations": [],
"referencedDeclaration": 1263,
"src": "5911:4:5",
"typeDescriptions": {
  "typeIdentifier": "t_bytes_memory_ptr",
  "typeString": "bytes memory"
}
},
],
"expression": {
  "argumentTypes": [
    {
      "typeIdentifier": "t_bytes_memory_ptr",
      "typeString": "bytes memory"
    }
  ],
},
"expression": {
  "id": 1281,
  "name": "target",
  "nodeType": "Identifier",
  "overloadedDeclarations": [],
  "referencedDeclaration": 1261,
  "src": "5893:6:5",
  "typeDescriptions": {
    "typeIdentifier": "t_address",
    "typeString": "address"
  }
},
},
"constant": false,
"lValue": false,
"pure": false,
"returnValueRequested": false,
"memberName": "staticcall",

```

```

        "nodeType": "MemberAccess",
        "src": "5893:17:5",
        "typeDescriptions": {
            "typeIdentifier":
"t_function_barestaticcall_view$_t_bytes_memory_ptr_$returns$_t_bool_$_t_bytes_me
memory_ptr_$",
            "typeString": "function (bytes memory) view returns (bool,bytes
memory)"
        }
    },
    "id": 1284,
    "isConstant": false,
    "isLValue": false,
    "isPure": false,
    "kind": "functionCall",
    "IValueRequested": false,
    "names": [],
    "nodeType": "FunctionCall",
    "src": "5893:23:5",
    "tryCall": false,
    "typeDescriptions": {
        "typeIdentifier": "t_tuple$_t_bool_$_t_bytes_memory_ptr_$",
        "typeString": "tuple(bool,bytes memory)"
    }
},
"nodeType": "VariableDeclarationStatement",
"src": "5851:65:5"
},
{
    "expression": {
        "arguments": [
            {
                "id": 1287,
                "name": "success",

```

```

    "nodeType": "Identifier",
    "overloadedDeclarations": [],
    "referencedDeclaration": 1278,
    "src": "5950:7:5",
    "typeDescriptions": {
      "typeIdentifier": "t_bool",
      "typeString": "bool"
    }
  },
  {
    "id": 1288,
    "name": "returndata",
    "nodeType": "Identifier",
    "overloadedDeclarations": [],
    "referencedDeclaration": 1280,
    "src": "5959:10:5",
    "typeDescriptions": {
      "typeIdentifier": "t_bytes_memory_ptr",
      "typeString": "bytes memory"
    }
  },
  {
    "id": 1289,
    "name": "errorMessage",
    "nodeType": "Identifier",
    "overloadedDeclarations": [],
    "referencedDeclaration": 1265,
    "src": "5971:12:5",
    "typeDescriptions": {
      "typeIdentifier": "t_string_memory_ptr",
      "typeString": "string memory"
    }
  }
],

```

```

"expression": {
  "argumentTypes": [
    {
      "typeIdentifier": "t_bool",
      "typeString": "bool"
    },
    {
      "typeIdentifier": "t_bytes_memory_ptr",
      "typeString": "bytes memory"
    },
    {
      "typeIdentifier": "t_string_memory_ptr",
      "typeString": "string memory"
    }
  ],
  "id": 1286,
  "name": "verifyCallResult",
  "nodeType": "Identifier",
  "overloadedDeclarations": [],
  "referencedDeclaration": 1376,
  "src": "5933:16:5",
  "typeDescriptions": {
    "typeIdentifier":
      "t_function_internal_pure$t_bool_$t_bytes_memory_ptr_$t_string_memory_ptr_$ret
      urns$t_bytes_memory_ptr_$",
    "typeString": "function (bool,bytes memory,string memory) pure returns
      (bytes memory)"
  }
},
"id": 1290,
"isConstant": false,
"isLValue": false,
"isPure": false,
"kind": "functionCall",

```

```

    "IValueRequested": false,
    "names": [],
    "nodeType": "FunctionCall",
    "src": "5933:51:5",
    "tryCall": false,
    "typeDescriptions": {
      "typeIdentifier": "t_bytes_memory_ptr",
      "typeString": "bytes memory"
    }
  },
  "functionReturnParameters": 1269,
  "id": 1291,
  "nodeType": "Return",
  "src": "5926:58:5"
}
]
},
"documentation": {
  "id": 1259,
  "nodeType": "StructuredDocumentation",
  "src": "5427:173:5",
  "text": " @dev Same as {xref-Address-functionCall-address-bytes-string-
}[\`functionCall`],\n but performing a static call.\n _Available since v3.3._"
},
"id": 1293,
"implemented": true,
"kind": "function",
"modifiers": [],
"name": "functionStaticCall",
"nameLocation": "5614:18:5",
"nodeType": "FunctionDefinition",
"parameters": {
  "id": 1266,
  "nodeType": "ParameterList",

```

```

"parameters": [
  {
    "constant": false,
    "id": 1261,
    "mutability": "mutable",
    "name": "target",
    "nameLocation": "5650:6:5",
    "nodeType": "VariableDeclaration",
    "scope": 1293,
    "src": "5642:14:5",
    "stateVariable": false,
    "storageLocation": "default",
    "typeDescriptions": {
      "typeIdentifier": "t_address",
      "typeString": "address"
    },
    "typeName": {
      "id": 1260,
      "name": "address",
      "nodeType": "ElementaryTypeName",
      "src": "5642:7:5",
      "stateMutability": "nonpayable",
      "typeDescriptions": {
        "typeIdentifier": "t_address",
        "typeString": "address"
      }
    },
    "visibility": "internal"
  },
  {
    "constant": false,
    "id": 1263,
    "mutability": "mutable",
    "name": "data",

```

```

"nameLocation": "5679:4:5",
"nodeType": "VariableDeclaration",
"scope": 1293,
"src": "5666:17:5",
"stateVariable": false,
"storageLocation": "memory",
"typeDescriptions": {
  "typeIdentifier": "t_bytes_memory_ptr",
  "typeString": "bytes"
},
"typeName": {
  "id": 1262,
  "name": "bytes",
  "nodeType": "ElementaryTypeName",
  "src": "5666:5:5",
  "typeDescriptions": {
    "typeIdentifier": "t_bytes_storage_ptr",
    "typeString": "bytes"
  }
},
"visibility": "internal"
},
{
  "constant": false,
  "id": 1265,
  "mutability": "mutable",
  "name": "errorMessage",
  "nameLocation": "5707:12:5",
  "nodeType": "VariableDeclaration",
  "scope": 1293,
  "src": "5693:26:5",
  "stateVariable": false,
  "storageLocation": "memory",
  "typeDescriptions": {

```

```

    "typeIdentifier": "t_string_memory_ptr",
    "typeString": "string"
  },
  "typeName": {
    "id": 1264,
    "name": "string",
    "nodeType": "ElementaryTypeName",
    "src": "5693:6:5",
    "typeDescriptions": {
      "typeIdentifier": "t_string_storage_ptr",
      "typeString": "string"
    }
  },
  "visibility": "internal"
}
],
"src": "5632:93:5"
},
"returnParameters": {
  "id": 1269,
  "nodeType": "ParameterList",
  "parameters": [
    {
      "constant": false,
      "id": 1268,
      "mutability": "mutable",
      "name": "",
      "nameLocation": "-1:-1:-1",
      "nodeType": "VariableDeclaration",
      "scope": 1293,
      "src": "5749:12:5",
      "stateVariable": false,
      "storageLocation": "memory",
      "typeDescriptions": {

```

```

        "typeIdentifier": "t_bytes_memory_ptr",
        "typeString": "bytes"
    },
    "typeName": {
        "id": 1267,
        "name": "bytes",
        "nodeType": "ElementaryTypeName",
        "src": "5749:5:5",
        "typeDescriptions": {
            "typeIdentifier": "t_bytes_storage_ptr",
            "typeString": "bytes"
        }
    },
    "visibility": "internal"
}
],
"src": "5748:14:5"
},
"scope": 1377,
"src": "5605:386:5",
"stateMutability": "view",
"virtual": false,
"visibility": "internal"
},
{
    "body": {
        "id": 1309,
        "nodeType": "Block",
        "src": "6267:101:5",
        "statements": [
            {
                "expression": {
                    "arguments": [
                        {

```

```

    "id": 1304,
    "name": "target",
    "nodeType": "Identifier",
    "overloadedDeclarations": [],
    "referencedDeclaration": 1296,
    "src": "6305:6:5",
    "typeDescriptions": {
      "typeIdentifier": "t_address",
      "typeString": "address"
    }
  },
  {
    "id": 1305,
    "name": "data",
    "nodeType": "Identifier",
    "overloadedDeclarations": [],
    "referencedDeclaration": 1298,
    "src": "6313:4:5",
    "typeDescriptions": {
      "typeIdentifier": "t_bytes_memory_ptr",
      "typeString": "bytes memory"
    }
  },
  {
    "hexValue":
"416464726573733a206c6f772d6c6576656c20646556c65676174652063616c6c2066616
96c6564",
    "id": 1306,
    "isConstant": false,
    "isLValue": false,
    "isPure": true,
    "kind": "string",
    "IValueRequested": false,
    "nodeType": "Literal",

```

```

        "src": "6319:41:5",
        "typeDescriptions": {
            "typeIdentifier":
"t_stringliteral_9fdcd12e4b726339b32a442b0a448365d5d85c96b2d2cff917b4f66c63110
398",
            "typeString": "literal_string \"Address: low-level delegate call failed\""
        },
        "value": "Address: low-level delegate call failed"
    }
],
"expression": {
    "argumentTypes": [
        {
            "typeIdentifier": "t_address",
            "typeString": "address"
        },
        {
            "typeIdentifier": "t_bytes_memory_ptr",
            "typeString": "bytes memory"
        }
    ]
}
}

```

Appendix 1(G):Land Ownership Transfer Component Code

```

{
    "contractName": "PropertyOwnership",
    "abi": [
        {
            "anonymous": false,
            "inputs": [
                {
                    "indexed": false,
                    "internalType": "uint256",
                    "name": "propertyId",
                    "type": "uint256"
                }
            ]
        }
    ]
}

```

```

    }
  ],
  "name": "LogApprovedForTransaction",
  "type": "event"
},
{
  "anonymous": false,
  "inputs": [
    {
      "indexed": false,
      "internalType": "uint256",
      "name": "propertyId",
      "type": "uint256"
    }
  ],
  "name": "LogSold",
  "type": "event"
},
{
  "anonymous": false,
  "inputs": [
    {
      "indexed": false,
      "internalType": "uint256",
      "name": "propertyId",
      "type": "uint256"
    }
  ],
  {
    "indexed": false,
    "internalType": "string",
    "name": "name",
    "type": "string"
  },
  {

```

```

    "indexed": false,
    "internalType": "string",
    "name": "location",
    "type": "string"
  },
  {
    "indexed": false,
    "internalType": "string",
    "name": "holder_name",
    "type": "string"
  },
  {
    "indexed": false,
    "internalType": "string",
    "name": "lr_no",
    "type": "string"
  }
],
"name": "NewProperty",
"type": "event"
},
{
  "anonymous": false,
  "inputs": [
    {
      "indexed": true,
      "internalType": "address",
      "name": "previousOwner",
      "type": "address"
    },
    {
      "indexed": true,
      "internalType": "address",
      "name": "newOwner",

```

```

    "type": "address"
  }
],
"name": "OwnershipTransferred",
"type": "event"
},
{
  "inputs": [
    {
      "internalType": "uint256",
      "name": "_propertyId",
      "type": "uint256"
    }
  ],
  "name": "buyProperty",
  "outputs": [],
  "stateMutability": "payable",
  "type": "function"
},
{
  "inputs": [],
  "name": "getHolders",
  "outputs": [
    {
      "components": [
        {
          "internalType": "string",
          "name": "name",
          "type": "string"
        },
        {
          "internalType": "string",
          "name": "tax_pin",
          "type": "string"
        }
      ]
    }
  ]
}

```

```

    },
    {
      "internalType": "string",
      "name": "email",
      "type": "string"
    },
    {
      "internalType": "uint256",
      "name": "id_no",
      "type": "uint256"
    },
    {
      "internalType": "uint256",
      "name": "contact",
      "type": "uint256"
    },
    {
      "internalType": "address",
      "name": "owner",
      "type": "address"
    },
    {
      "internalType": "bool",
      "name": "isExist",
      "type": "bool"
    }
  ],
  "internalType": "struct Registry.holder[]",
  "name": "",
  "type": "tuple[]"
}
],
"stateMutability": "view",
"type": "function"

```

```

},
{
  "inputs": [
    {
      "internalType": "address",
      "name": "_holder",
      "type": "address"
    }
  ],
  "name": "getPropertyByHolder",
  "outputs": [
    {
      "internalType": "uint256[]",
      "name": "",
      "type": "uint256[]"
    }
  ],
  "stateMutability": "view",
  "type": "function"
},
{
  "inputs": [
    {
      "internalType": "address",
      "name": "",
      "type": "address"
    }
  ],
  "name": "holderToPropertyCount",
  "outputs": [
    {
      "internalType": "uint256",
      "name": "",
      "type": "uint256"
    }
  ]
}

```

```

    }
  ],
  "stateMutability": "view",
  "type": "function"
},
{
  "inputs": [
    {
      "internalType": "address",
      "name": "",
      "type": "address"
    }
  ],
  "name": "holderdetails",
  "outputs": [
    {
      "internalType": "string",
      "name": "name",
      "type": "string"
    },
    {
      "internalType": "string",
      "name": "tax_pin",
      "type": "string"
    },
    {
      "internalType": "string",
      "name": "email",
      "type": "string"
    },
    {
      "internalType": "uint256",
      "name": "id_no",
      "type": "uint256"
    }
  ]
}

```

```

    },
    {
      "internalType": "uint256",
      "name": "contact",
      "type": "uint256"
    },
    {
      "internalType": "address",
      "name": "owner",
      "type": "address"
    },
    {
      "internalType": "bool",
      "name": "isExist",
      "type": "bool"
    }
  ],
  "stateMutability": "view",
  "type": "function"
},
{
  "inputs": [
    {
      "internalType": "uint256",
      "name": "",
      "type": "uint256"
    }
  ],
  "name": "holders",
  "outputs": [
    {
      "internalType": "string",
      "name": "name",
      "type": "string"
    }
  ]
}

```

```
},
{
  "internalType": "string",
  "name": "tax_pin",
  "type": "string"
},
{
  "internalType": "string",
  "name": "email",
  "type": "string"
},
{
  "internalType": "uint256",
  "name": "id_no",
  "type": "uint256"
},
{
  "internalType": "uint256",
  "name": "contact",
  "type": "uint256"
},
{
  "internalType": "address",
  "name": "owner",
  "type": "address"
},
{
  "internalType": "bool",
  "name": "isExist",
  "type": "bool"
}
],
```

```
{
```

```

"contractName": "Ownable",
"abi": [
  {
    "anonymous": false,
    "inputs": [
      {
        "indexed": true,
        "internalType": "address",
        "name": "previousOwner",
        "type": "address"
      },
      {
        "indexed": true,
        "internalType": "address",
        "name": "newOwner",
        "type": "address"
      }
    ],
    "name": "OwnershipTransferred",
    "type": "event"
  },
  {
    "inputs": [],
    "name": "owner",
    "outputs": [
      {
        "internalType": "address",
        "name": "",
        "type": "address"
      }
    ],
    "stateMutability": "view",
    "type": "function"
  },

```

```

{
  "inputs": [],
  "name": "renounceOwnership",
  "outputs": [],
  "stateMutability": "nonpayable",
  "type": "function"
},
{
  "inputs": [
    {
      "internalType": "address",
      "name": "newOwner",
      "type": "address"
    }
  ],
  "name": "transferOwnership",
  "outputs": [],
  "stateMutability": "nonpayable",
  "type": "function"
}
],
"metadata":
  "{\n  \"compiler\": {\n    \"version\": \"0.8.9+commit.e5eed63a\"\n  },\n  \"language\": \"Solidity\",\n  \"output\": {\n    \"abi\": [\n      {\n        \"anonymous\": false,\n        \"inputs\": [\n          {\n            \"indexed\": true,\n            \"internalType\": \"address\",\n            \"name\": \"previousOwner\",\n            \"type\": \"address\"\n          },\n          {\n            \"indexed\": true,\n            \"internalType\": \"address\",\n            \"name\": \"newOwner\",\n            \"type\": \"address\"\n          }\n        ],\n        \"name\": \"OwnershipTransferred\",\n        \"type\": \"event\"\n      },\n      {\n        \"inputs\": [],\n        \"name\": \"owner\",\n        \"outputs\": [\n          {\n            \"internalType\": \"address\",\n            \"name\": \"\",\n            \"type\": \"address\"\n          }\n        ],\n        \"stateMutability\": \"view\",\n        \"type\": \"function\"\n      },\n      {\n        \"inputs\": [],\n        \"name\": \"renounceOwnership\",\n        \"outputs\"
      }\n    ]\n  }\n}\n",
  "nodeType": "StructuredDocumentation",
  "src": "1269:331:0",
  "text": " @dev Leaves the contract without owner. It will not be possible to call\n`onlyOwner` functions anymore. Can only be called by the current owner.\nNOTE:"

```

Renouncing ownership will leave the contract without an owner,\n thereby removing any functionality that is only available to the owner."

```
},
"functionSelector": "715018a6",
"id": 60,
"implemented": true,
"kind": "function",
"modifiers": [
{
"id": 50,
"kind": "modifierInvocation",
"modifierName": {
"id": 49,
"name": "onlyOwner",
"nodeType": "IdentifierPath",
"referencedDeclaration": 46,
"src": "1649:9:0"
},
"nodeType": "ModifierInvocation",
"src": "1649:9:0"
}
],
"name": "renounceOwnership",
"nameLocation": "1614:17:0",
"nodeType": "FunctionDefinition",
"parameters": {
"id": 48,
"nodeType": "ParameterList",
"parameters": [],
"src": "1631:2:0"
},
"returnParameters": {
"id": 51,
"nodeType": "ParameterList",
```

```

"parameters": [],
"src": "1659:0:0"
},
"scope": 103,
"src": "1605:92:0",
"stateMutability": "nonpayable",
"virtual": true,
"visibility": "public"
},
{
"body": {
"id": 82,
"nodeType": "Block",
"src": "1916:119:0",
"statements": [
{
"expression": {
"arguments": [
{
"commonType": {
"typeIdentifier": "t_address",
"typeString": "address"
},
"id": 74,
"isConstant": false,
"isLValue": false,
"isPure": false,
"IValueRequested": false,
"leftExpression": {
"id": 69,
"name": "newOwner",
"nodeType": "Identifier",
"overloadedDeclarations": [],
"referencedDeclaration": 63,

```

```

"src": "1934:8:0",
"typeDescriptions": {
"typeIdentifier": "t_address",
"typeString": "address"
}
},
"nodeType": "BinaryOperation",
"operator": "!=",
"rightExpression": {
"arguments": [
{
"hexValue": "30",
"id": 72,
"isConstant": false,
"isLValue": false,
"isPure": true,
"kind": "number",
"IValueRequested": false,
"nodeType": "Literal",
"src": "1954:1:0",
"typeDescriptions": {
"typeIdentifier": "t_rational_0_by_1",
"typeString": "int_const 0"
}
},
"value": "0"
}
],
"expression": {
"argumentTypes": [
{
"typeIdentifier": "t_rational_0_by_1",
"typeString": "int_const 0"
}
]
},

```

```

"id": 71,
"isConstant": false,
"isLValue": false,
"isPure": true,
"IValueRequested": false,
"nodeType": "ElementaryTypeNameExpression",
"src": "1946:7:0",
"typeDescriptions": {
  "typeIdentifier": "t_type$_t_address_$",
  "typeString": "type(address)"
},
"typeName": {
  "id": 70,
  "name": "address",
  "nodeType": "ElementaryTypeName",
  "src": "1946:7:0",
  "typeDescriptions": {}
},
},
"id": 73,
"isConstant": false,
"isLValue": false,
"isPure": true,
"kind": "typeConversion",
"IValueRequested": false,
"names": [],
"nodeType": "FunctionCall",
"src": "1946:10:0",
"tryCall": false,
"typeDescriptions": {
  "typeIdentifier": "t_address",
  "typeString": "address"
},
},

```

```

"src": "1934:22:0",
"typeDescriptions": {
  "typeIdentifier": "t_bool",
  "typeString": "bool"
}
},
{
  "hexValue":
"4f776e61626c653a206e6577206f776e657220697320746865207a65726f206164647265
7373",
  "id": 75,
  "isConstant": false,
  "isLValue": false,
  "isPure": true,
  "kind": "string",
  "IValueRequested": false,
  "nodeType": "Literal",
  "src": "1958:40:0",
  "typeDescriptions": {
    "typeIdentifier":
"t_stringliteral_245f15ff17f551913a7a18385165551503906a406f905ac1c2437281a7cd0
cfe",
    "typeString": "literal_string \"Ownable: new owner is the zero address\""
  },
  "value": "Ownable: new owner is the zero address"
}
],
"expression": {
  "argumentTypes": [
    {
      "typeIdentifier": "t_bool",
      "typeString": "bool"
    },
  ],
  {

```

```

"typeIdentifier":
  "t_stringliteral_245f15ff17f551913a7a18385165551503906a406f905ac1c2437281a7cd0
  cfe",
"typeString": "literal_string \"Ownable: new owner is the zero address\""
}
],
"id": 68,
"name": "require",
"nodeType": "Identifier",
"overloadedDeclarations": [
  4294967278,
  4294967278
],
"referencedDeclaration": 4294967278,
"src": "1926:7:0",
"typeDescriptions": {
  "typeIdentifier":
    "t_function_require_pure$_t_bool_$_t_string_memory_ptr_$returns$__$",
    "typeString": "function (bool,string memory) pure"
  }
},
"id": 76,
"isConstant": false,
"isLValue": false,
"isPure": false,
"kind": "functionCall",
"IValueRequested": false,
"names": [],
"nodeType": "FunctionCall",
"src": "1926:73:0",
"tryCall": false,
"typeDescriptions": {
  "typeIdentifier": "t_tuple$__$",
  "typeString": "tuple()"
}

```



```

"src": "2009:9:0",
"typeDescriptions": {
"typeIdentifier": "t_function_internal_nonpayable$_t_address_$returns$__$",
"typeString": "function (address)"
}
},
"id": 80,
"isConstant": false,
"isLValue": false,
"isPure": false,
"kind": "functionCall",
"IValueRequested": false,
"names": [],
"nodeType": "FunctionCall",
"src": "2009:19:0",
"tryCall": false,
"typeDescriptions": {
"typeIdentifier": "t_tuple$__$",
"typeString": "tuple()"
}
},
"id": 81,
"nodeType": "ExpressionStatement",
"src": "2009:19:0"
}
],
},
"documentation": {
"id": 61,
"nodeType": "StructuredDocumentation",
"src": "1703:138:0",
"text": " @dev Transfers ownership of the contract to a new account (^newOwner`).\n
Can only be called by the current owner."
},

```

```

"functionSelector": "f2fde38b",
"id": 83,
"implemented": true,
"kind": "function",
"modifiers": [
  {
    "id": 66,
    "kind": "modifierInvocation",
    "modifierName": {
      "id": 65,
      "name": "onlyOwner",
      "nodeType": "IdentifierPath",
      "referencedDeclaration": 46,
      "src": "1906:9:0"
    },
    "nodeType": "ModifierInvocation",
    "src": "1906:9:0"
  }
],
"name": "transferOwnership",
"nameLocation": "1855:17:0",
"nodeType": "FunctionDefinition",
"parameters": {
  "id": 64,
  "nodeType": "ParameterList",
  "parameters": [
    {
      "constant": false,
      "id": 63,
      "mutability": "mutable",
      "name": "newOwner",
      "nameLocation": "1881:8:0",
      "nodeType": "VariableDeclaration",
      "scope": 83,

```

```

"src": "1873:16:0",
"stateVariable": false,
"storageLocation": "default",
"typeDescriptions": {
"typeIdentifier": "t_address",
"typeString": "address"
},
"typeName": {
"id": 62,
"name": "address",
"nodeType": "ElementaryTypeName",
"src": "1873:7:0",
"stateMutability": "nonpayable",
"typeDescriptions": {
"typeIdentifier": "t_address",
"typeString": "address"
}
},
"visibility": "internal"
},
"src": "1872:18:0"
},
"returnParameters": {
"id": 67,
"nodeType": "ParameterList",
"parameters": [],
"src": "1916:0:0"
},
"scope": 103,
"src": "1846:189:0",
"stateMutability": "nonpayable",
"virtual": true,
"visibility": "public"

```

```

},
{
  "body": {
    "id": 101,
    "nodeType": "Block",
    "src": "2086:124:0",
    "statements": [
      {
        "assignments": [
          89
        ],
        "declarations": [
          {
            "constant": false,
            "id": 89,
            "mutability": "mutable",
            "name": "oldOwner",
            "nameLocation": "2104:8:0",
            "nodeType": "VariableDeclaration",
            "scope": 101,
            "src": "2096:16:0",
            "stateVariable": false,
            "storageLocation": "default",
            "typeDescriptions": {
              "typeIdentifier": "t_address",
              "typeString": "address"
            },
            "typeName": {
              "id": 88,
              "name": "address",
              "nodeType": "ElementaryTypeName",
              "src": "2096:7:0",
              "stateMutability": "nonpayable",
              "typeDescriptions": {

```

```

"typeIdentifier": "t_address",
"typeString": "address"
}
},
"visibility": "internal"
}
],
"id": 91,
"initialValue": {
"id": 90,
"name": "_owner",
"nodeType": "Identifier",
"overloadedDeclarations": [],
"referencedDeclaration": 7,
"src": "2115:6:0",
"typeDescriptions": {
"typeIdentifier": "t_address",
"typeString": "address"
}
},
"nodeType": "VariableDeclarationStatement",
"src": "2096:25:0"
},
{
"expression": {
"id": 94,
"isConstant": false,
"isLValue": false,
"isPure": false,
"IValueRequested": false,
"leftHandSide": {
"id": 92,
"name": "_owner",
"nodeType": "Identifier",

```

```

"overloadedDeclarations": [],
"referencedDeclaration": 7,
"src": "2131:6:0",
"typeDescriptions": {
"typeIdentifier": "t_address",
"typeString": "address"
}
},
"nodeType": "Assignment",
"operator": "=",
"rightHandSide": {
"id": 93,
"name": "newOwner",
"nodeType": "Identifier",
"overloadedDeclarations": [],
"referencedDeclaration": 85,
"src": "2140:8:0",
"typeDescriptions": {
"typeIdentifier": "t_address",
"typeString": "address"
}
},
"src": "2131:17:0",
"typeDescriptions": {
"typeIdentifier": "t_address",
"typeString": "address"
}
},
"id": 95,
"nodeType": "ExpressionStatement",
"src": "2131:17:0"
},
{
"eventCall": {

```

```

"arguments": [
  {
    "id": 97,
    "name": "oldOwner",
    "nodeType": "Identifier",
    "overloadedDeclarations": [],
    "referencedDeclaration": 89,
    "src": "2184:8:0",
    "typeDescriptions": {
      "typeIdentifier": "t_address",
      "typeString": "address"
    }
  },
  {
    "id": 98,
    "name": "newOwner",
    "nodeType": "Identifier",
    "overloadedDeclarations": [],
    "referencedDeclaration": 85,
    "src": "2194:8:0",
    "typeDescriptions": {
      "typeIdentifier": "t_address",
      "typeString": "address"
    }
  }
],
"expression": {
  "argumentTypes": [
    {
      "typeIdentifier": "t_address",
      "typeString": "address"
    },
    {
      "typeIdentifier": "t_address",

```

```

"typeString": "address"
}
],
"id": 96,
"name": "OwnershipTransferred",
"nodeType": "Identifier",
"overloadedDeclarations": [],
"referencedDeclaration": 13,
"src": "2163:20:0",
"typeDescriptions": {
"typeIdentifier": "t_function_event_nonpayable$_t_address_$_t_address_$returns$__$",
"typeString": "function (address,address)"
}
},
"id": 99,
"isConstant": false,
"isLValue": false,
"isPure": false,
"kind": "functionCall",
"IValueRequested": false,
"names": [],
"nodeType": "FunctionCall",
"src": "2163:40:0",
"tryCall": false,
"typeDescriptions": {
"typeIdentifier": "t_tuple$__$",
"typeString": "tuple()"
}
},
"id": 100,
"nodeType": "EmitStatement",
"src": "2158:45:0"
}
]

```

```

},
"id": 102,
"implemented": true,
"kind": "function",
"modifiers": [],
"name": "_setOwner",
"nameLocation": "2050:9:0",
"nodeType": "FunctionDefinition",
"parameters": {
  "id": 86,
  "nodeType": "ParameterList",
  "parameters": [
    {
      "constant": false,
      "id": 85,
      "mutability": "mutable",
      "name": "newOwner",
      "nameLocation": "2068:8:0",
      "nodeType": "VariableDeclaration",
      "scope": 102,
      "src": "2060:16:0",
      "stateVariable": false,
      "storageLocation": "default",
      "typeDescriptions": {
        "typeIdentifier": "t_address",
        "typeString": "address"
      },
      "typeName": {
        "id": 84,
        "name": "address",
        "nodeType": "ElementaryTypeName",
        "src": "2060:7:0",
        "stateMutability": "nonpayable",
        "typeDescriptions": {

```

```

"typeIdentifier": "t_address",
"typeString": "address"
}
},
"visibility": "internal"
}
],
"src": "2059:18:0"
},
"returnParameters": {
"id": 87,
"nodeType": "ParameterList",
"parameters": [],
"src": "2086:0:0"
},
"scope": 103,
"src": "2041:169:0",
"stateMutability": "nonpayable",
"virtual": false,
"visibility": "private"
}
],
"scope": 104,
"src": "585:1627:0",
"usedErrors": []
}
],
"src": "33:2180:0"
},
"legacyAST": {
"absolutePath": "@openzeppelin/contracts/access/Ownable.sol",
"exportedSymbols": {
"Context": [

```

```

],
"Ownable": [
  103
]
},
"id": 104,
"license": "MIT",
"nodeType": "SourceUnit",
"nodes": [
  {
    "id": 1,
    "literals": [
      "solidity",
      "^",
      "0.8",
      ".0"
    ],
    "nodeType": "PragmaDirective",
    "src": "33:23:0"
  },
  {
    "absolutePath": "@openzeppelin/contracts/utils/Context.sol",
    "file": "../utils/Context.sol",
    "id": 2,
    "nameLocation": "-1:-1:-1",
    "nodeType": "ImportDirective",
    "scope": 104,
    "sourceUnit": 126,
    "src": "58:30:0",
    "symbolAliases": [],
    "unitAlias": ""
  },
  {
    "abstract": true,

```

```

"baseContracts": [
  {
    "baseName": {
      "id": 4,
      "name": "Context",
      "nodeType": "IdentifierPath",
      "referencedDeclaration": 125,
      "src": "614:7:0"
    },
    "id": 5,
    "nodeType": "InheritanceSpecifier",
    "src": "614:7:0"
  }
],
"canonicalName": "Ownable",
"contractDependencies": [],
"contractKind": "contract",
"documentation": {
  "id": 3,
  "nodeType": "StructuredDocumentation",
  "src": "90:494:0",
  "text": " @dev Contract module which provides a basic access control mechanism,
where
there is an account (an owner) that can be granted exclusive access to
specific
functions.
By default, the owner account will be the one that deploys the contract.
This
can later be changed with {transferOwnership}.
This module is used through
inheritance. It will make available the modifier
`onlyOwner`, which can be applied to
your functions to restrict their use to
the owner."
},
"fullyImplemented": true,
"id": 103,
"linearizedBaseContracts": [
  103,
  125
],

```

```

"name": "Ownable",
"nameLocation": "603:7:0",
"nodeType": "ContractDefinition",
"nodes": [
{
"constant": false,
"id": 7,
"mutability": "mutable",
"name": "_owner",
"nameLocation": "644:6:0",
"nodeType": "VariableDeclaration",
"scope": 103,
"src": "628:22:0",
"stateVariable": true,
"storageLocation": "default",
"typeDescriptions": {
"typeIdentifier": "t_address",
"typeString": "address"
},
"typeName": {
"id": 6,
"name": "address",
"nodeType": "ElementaryTypeName",
"src": "628:7:0",
"stateMutability": "nonpayable",
"typeDescriptions": {
"typeIdentifier": "t_address",
"typeString": "address"
}
},
"visibility": "private"
},
{
"anonymous": false,

```

```

"id": 13,
"name": "OwnershipTransferred",
"nameLocation": "663:20:0",
"nodeType": "EventDefinition",
"parameters": {
  "id": 12,
  "nodeType": "ParameterList",
  "parameters": [
    {
      "constant": false,
      "id": 9,
      "indexed": true,
      "mutability": "mutable",
      "name": "previousOwner",
      "nameLocation": "700:13:0",
      "nodeType": "VariableDeclaration",
      "scope": 13,
      "src": "684:29:0",
      "stateVariable": false,
      "storageLocation": "default",
      "typeDescriptions": {
        "typeIdentifier": "t_address",
        "typeString": "address"
      },
      "typeName": {
        "id": 8,
        "name": "address",
        "nodeType": "ElementaryTypeName",
        "src": "684:7:0",
        "stateMutability": "nonpayable",
        "typeDescriptions": {
          "typeIdentifier": "t_address",
          "typeString": "address"
        }
      },
      "visibility": "internal"
    }
  ]
}

```

```

},
{
  "constant": false,
  "id": 11,
  "indexed": true,
  "mutability": "mutable",
  "name": "newOwner",
  "nameLocation": "731:8:0",
  "nodeType": "VariableDeclaration",
  "scope": 13,
  "src": "715:24:0",
  "stateVariable": false,
  "storageLocation": "default",
  "typeDescriptions": {
    "typeIdentifier": "t_address",
    "typeString": "address"
  },
  "typeName": {
    "id": 10,
    "name": "address",
    "nodeType": "ElementaryTypeName",
    "src": "715:7:0",
    "stateMutability": "nonpayable",
    "typeDescriptions": {
      "typeIdentifier": "t_address",
      "typeString": "address"
    }
  },
  "visibility": "internal"
}
],
"src": "683:57:0"
},
"src": "657:84:0"

```

```

},
{
  "body": {
    "id": 22,
    "nodeType": "Block",
    "src": "857:40:0",
    "statements": [
      {
        "expression": {
          "arguments": [
            {
              "arguments": [],
              "expression": {
                "argumentTypes": [],
                "id": 18,
                "name": "_msgSender",
                "nodeType": "Identifier",
                "overloadedDeclarations": [],
                "referencedDeclaration": 115,
                "src": "877:10:0",
                "typeDescriptions": {
                  "typeIdentifier": "t_function_internal_view__$returns$t_address_$",
                  "typeString": "function () view returns (address)"
                }
              }
            }
          ],
          "id": 19,
          "isConstant": false,
          "isLValue": false,
          "isPure": false,
          "kind": "functionCall",
          "lValueRequested": false,
          "names": [],
          "nodeType": "FunctionCall",
          "src": "877:12:0",

```

```

"tryCall": false,
"typeDescriptions": {
  "typeIdentifier": "t_address",
  "typeString": "address"
}
},
],
"expression": {
  "argumentTypes": [
    {
      "typeIdentifier": "t_address",
      "typeString": "address"
    }
  ],
  "id": 17,
  "name": "_setOwner",
  "nodeType": "Identifier",
  "overloadedDeclarations": [],
  "referencedDeclaration": 102,
  "src": "867:9:0",
  "typeDescriptions": {
    "typeIdentifier": "t_function_internal_nonpayable$_t_address_$returns__$",
    "typeString": "function (address)"
  }
},
},
"expression": {
  "id": 20,
  "isConstant": false,
  "isLValue": false,
  "isPure": false,
  "kind": "functionCall",
  "lValueRequested": false,
  "names": [],
  "nodeType": "FunctionCall",
  "src": "867:23:0",

```

```

"tryCall": false,
"typeDescriptions": {
  "typeIdentifier": "t_tuple$__$",
  "typeString": "tuple()"
}
},
"id": 21,
"nodeType": "ExpressionStatement",
"src": "867:23:0"
}
],
},
"documentation": {
  "id": 14,
  "nodeType": "StructuredDocumentation",
  "src": "747:91:0",
  "text": " @dev Initializes the contract setting the deployer as the initial owner."
},
"id": 23,
"implemented": true,
"kind": "constructor",
"modifiers": [],
"name": "",
"nameLocation": "-1:-1:-1",
"nodeType": "FunctionDefinition",
"parameters": {
  "id": 15,
  "nodeType": "ParameterList",
  "parameters": [],
  "src": "854:2:0"
},
"returnParameters": {
  "id": 16,
  "nodeType": "ParameterList",

```

```

"parameters": [],
"src": "857:0:0"
},
"scope": 103,
"src": "843:54:0",
"stateMutability": "nonpayable",
"virtual": false,
"visibility": "internal"
},
{
"body": {
"id": 31,
"nodeType": "Block",
"src": "1028:30:0",
"statements": [
{
"expression": {
"id": 29,
"name": "_owner",
"nodeType": "Identifier",
"overloadedDeclarations": [],
"referencedDeclaration": 7,
"src": "1045:6:0",
"typeDescriptions": {
"typeIdentifier": "t_address",
"typeString": "address"
}
}
},
"functionReturnParameters": 28,
"id": 30,
"nodeType": "Return",
"src": "1038:13:0"
}
]

```

```

},
"documentation": {
  "id": 24,
  "nodeType": "StructuredDocumentation",
  "src": "903:65:0",
  "text": " @dev Returns the address of the current owner."
},
"functionSelector": "8da5cb5b",
"id": 32,
"implemented": true,
"kind": "function",
"modifiers": [],
"name": "owner",
"nameLocation": "982:5:0",
"nodeType": "FunctionDefinition",
"parameters": {
  "id": 25,
  "nodeType": "ParameterList",
  "parameters": [],
  "src": "987:2:0"
},
"returnParameters": {
  "id": 28,
  "nodeType": "ParameterList",
  "parameters": [
    {
      "constant": false,
      "id": 27,
      "mutability": "mutable",
      "name": "",
      "nameLocation": "-1:-1:-1",
      "nodeType": "VariableDeclaration",
      "scope": 32,
      "src": "1019:7:0",

```

```

"stateVariable": false,
"storageLocation": "default",
"typeDescriptions": {
"typeIdentifier": "t_address",
"typeString": "address"
},
"typeName": {
"id": 26,
"name": "address",
"nodeType": "ElementaryTypeName",
"src": "1019:7:0",
"stateMutability": "nonpayable",
"typeDescriptions": {
"typeIdentifier": "t_address",
"typeString": "address"
}
},
"visibility": "internal"
}
],
"src": "1018:9:0"
},
"scope": 103,
"src": "973:85:0",
"stateMutability": "view",
"virtual": true,
"visibility": "public"
},
{
"body": {
"id": 45,
"nodeType": "Block",
"src": "1167:96:0",
"statements": [

```

```

{
  "expression": {
    "arguments": [
      {
        "commonType": {
          "typeIdentifier": "t_address",
          "typeString": "address"
        },
        "id": 40,
        "isConstant": false,
        "isLValue": false,
        "isPure": false,
        "IValueRequested": false,
        "leftExpression": {
          "arguments": [],
          "expression": {
            "argumentTypes": [],
            "id": 36,
            "name": "owner",
            "nodeType": "Identifier",
            "overloadedDeclarations": [],
            "referencedDeclaration": 32,
            "src": "1185:5:0",
            "typeDescriptions": {
              "typeIdentifier": "t_function_internal_view__$returns$t_address_$",
              "typeString": "function () view returns (address)"
            }
          },
          "id": 37,
          "isConstant": false,
          "isLValue": false,
          "isPure": false,
          "kind": "functionCall",
          "IValueRequested": false,

```

```

"names": [],
"nodeType": "FunctionCall",
"src": "1185:7:0",
"tryCall": false,
"typeDescriptions": {
"typeIdentifier": "t_address",
"typeString": "address"
}
},
"nodeType": "BinaryOperation",
"operator": "==",
"rightExpression": {
"arguments": [],
"expression": {
"argumentTypes": [],
"id": 38,
"name": "_msgSender",
"nodeType": "Identifier",
"overloadedDeclarations": [],
"referencedDeclaration": 115,
"src": "1196:10:0",
"typeDescriptions": {
"typeIdentifier": "t_function_internal_view__$returns$t_address_$",
"typeString": "function () view returns (address)"
}
}
},
"id": 39,
"isConstant": false,
"isLValue": false,
"isPure": false,
"kind": "functionCall",
"IValueRequested": false,
"names": [],
"nodeType": "FunctionCall",

```

```

"src": "1196:12:0",
"tryCall": false,
"typeDescriptions": {
"typeIdentifier": "t_address",
"typeString": "address"
}
},
"src": "1185:23:0",
"typeDescriptions": {
"typeIdentifier": "t_bool",
"typeString": "bool"
}
},
{
"hexValue":
"4f776e61626c653a2063616c6c6572206973206e6f7420746865206f776e6572",
"id": 41,
"isConstant": false,
"isLValue": false,
"isPure": true,
"kind": "string",
"lValueRequested": false,
"nodeType": "Literal",
"src": "1210:34:0",
"typeDescriptions": {
"typeIdentifier":
"t_stringliteral_9924ebdf1add33d25d4ef888e16131f0a5687b0580a36c21b5c301a6c462e
ffe",
"typeString": "literal_string \"Ownable: caller is not the owner\""
},
"value": "Ownable: caller is not the owner"
}
],
"expression": {

```

```

"argumentTypes": [
  {
    "typeIdentifier": "t_bool",
    "typeString": "bool"
  },
  {
    "typeIdentifier":
    "t_stringliteral_9924ebdf1add33d25d4ef888e16131f0a5687b0580a36c21b5c301a6c462e
ffe",
    "typeString": "literal_string \"Ownable: caller is not the owner\""
  }
],
"id": 35,
"name": "require",
"nodeType": "Identifier",
"overloadedDeclarations": [
  4294967278,
  4294967278
],
"referencedDeclaration": 4294967278,
"src": "1177:7:0",
"typeDescriptions": {
  "typeIdentifier":
  "t_function_require_pure$t_bool_$t_string_memory_ptr_$returns__$",
  "typeString": "function (bool,string memory) pure"
}
},
"id": 42,
"isConstant": false,
"isLValue": false,
"isPure": false,
"kind": "functionCall",
"IValueRequested": false,
"names": [],

```

```

"nodeType": "FunctionCall",
"src": "1177:68:0",
"tryCall": false,
"typeDescriptions": {
"typeIdentifier": "t_tuple$__$",
"typeString": "tuple()"
}
},
"id": 43,
"nodeType": "ExpressionStatement",
"src": "1177:68:0"
},
{
"id": 44,
"nodeType": "PlaceholderStatement",
"src": "1255:1:0"
}
]
},
"documentation": {
"id": 33,
"nodeType": "StructuredDocumentation",
"src": "1064:77:0",
"text": " @dev Throws if called by any account other than the owner."
},
"id": 46,
"name": "onlyOwner",
"nameLocation": "1155:9:0",
"nodeType": "ModifierDefinition",
"parameters": {
"id": 34,
"nodeType": "ParameterList",
"parameters": [],
"src": "1164:2:0"

```

```

},
"src": "1146:117:0",
"virtual": false,
"visibility": "internal"
},
{
"body": {
"id": 59,
"nodeType": "Block",
"src": "1659:38:0",
"statements": [
{
"expression": {
"arguments": [
{
"arguments": [
{
"hexValue": "30",
"id": 55,
"isConstant": false,
"isLValue": false,
"isPure": true,
"kind": "number",
"lValueRequested": false,
"nodeType": "Literal",
"src": "1687:1:0",
"typeDescriptions": {
"typeIdentifier": "t_rational_0_by_1",
"typeString": "int_const 0"
},
},
"value": "0"
}
],
},
},
],
"expression": {

```

```

"argumentTypes": [
  {
    "typeIdentifier": "t_rational_0_by_1",
    "typeString": "int_const 0"
  }
],
"id": 54,
"isConstant": false,
"isLValue": false,
"isPure": true,
"lValueRequested": false,
"nodeType": "ElementaryTypeNameExpression",
"src": "1679:7:0",
"typeDescriptions": {
  "typeIdentifier": "t_type$_t_address_$",
  "typeString": "type(address)"
},
"typeName": {
  "id": 53,
  "name": "address",
  "nodeType": "ElementaryTypeName",
  "src": "1679:7:0",
  "typeDescriptions": {}
},
},
"id": 56,
"isConstant": false,
"isLValue": false,
"isPure": true,
"kind": "typeConversion",
"lValueRequested": false,
"names": [],
"nodeType": "FunctionCall",
"src": "1679:10:0",

```

```

"tryCall": false,
"typeDescriptions": {
  "typeIdentifier": "t_address",
  "typeString": "address"
}
},
"expression": {
  "argumentTypes": [
    {
      "typeIdentifier": "t_address",
      "typeString": "address"
    }
  ],
  "id": 52,
  "name": "_setOwner",
  "nodeType": "Identifier",
  "overloadedDeclarations": [],
  "referencedDeclaration": 102,
  "src": "1669:9:0",
  "typeDescriptions": {
    "typeIdentifier": "t_function_internal_nonpayable$_t_address_$returns__$",
    "typeString": "function (address)"
  }
},
"kind": "functionCall",
"IValueRequested": false,
"names": [],
"nodeType": "FunctionCall",
"src": "1669:21:0",

```

```

"tryCall": false,
"typeDescriptions": {
  "typeIdentifier": "t_tuple$__$",
  "typeString": "tuple()"
}
},
"id": 58,
"nodeType": "ExpressionStatement",
"src": "1669:21:0"
}
],
},
"documentation": {
  "id": 47,
  "nodeType": "StructuredDocumentation",
  "src": "1269:331:0",
  "text": " @dev Leaves the contract without owner. It will not be possible to call\n`onlyOwner` functions anymore. Can only be called by the current owner.\nNOTE: Renouncing ownership will leave the contract without an owner,\nthereby removing any functionality that is only available to the owner."
},
"functionSelector": "715018a6",
"id": 60,
"implemented": true,
"kind": "function",
"modifiers": [
  {
    "id": 50,
    "kind": "modifierInvocation",
    "modifierName": {
      "id": 49,
      "name": "onlyOwner",
      "nodeType": "IdentifierPath",
      "referencedDeclaration": 46,

```

```

"src": "1649:9:0"
},
"nodeType": "ModifierInvocation",
"src": "1649:9:0"
}
],
"name": "renounceOwnership",
"nameLocation": "1614:17:0",
"nodeType": "FunctionDefinition",
"parameters": {
"id": 48,
"nodeType": "ParameterList",
"parameters": [],
"src": "1631:2:0"
},
"returnParameters": {
"id": 51,
"nodeType": "ParameterList",
"parameters": [],
"src": "1659:0:0"
},
"scope": 103,
"src": "1605:92:0",
"stateMutability": "nonpayable",
"virtual": true,
"visibility": "public"
},
{
"body": {
"id": 82,
"nodeType": "Block",
"src": "1916:119:0",
"statements": [
{

```

```

"expression": {
  "arguments": [
    {
      "commonType": {
        "typeIdentifier": "t_address",
        "typeString": "address"
      },
      "id": 74,
      "isConstant": false,
      "isLValue": false,
      "isPure": false,
      "IValueRequested": false,
      "leftExpression": {
        "id": 69,
        "name": "newOwner",
        "nodeType": "Identifier",
        "overloadedDeclarations": [],
        "referencedDeclaration": 63,
        "src": "1934:8:0",
        "typeDescriptions": {
          "typeIdentifier": "t_address",
          "typeString": "address"
        }
      },
      "nodeType": "BinaryOperation",
      "operator": "!=",
      "rightExpression": {
        "arguments": [
          {
            "hexValue": "30",
            "id": 72,
            "isConstant": false,
            "isLValue": false,
            "isPure": true,

```

```

"kind": "number",
"lValueRequested": false,
"nodeType": "Literal",
"src": "1954:1:0",
"typeDescriptions": {
  "typeIdentifier": "t_rational_0_by_1",
  "typeString": "int_const 0"
},
"value": "0"
},
],
"expression": {
  "argumentTypes": [
    {
      "typeIdentifier": "t_rational_0_by_1",
      "typeString": "int_const 0"
    }
  ],
  "id": 71,
  "isConstant": false,
  "isLValue": false,
  "isPure": true,
  "lValueRequested": false,
  "nodeType": "ElementaryTypeNameExpression",
  "src": "1946:7:0",
  "typeDescriptions": {
    "typeIdentifier": "t_type$_t_address_$",
    "typeString": "type(address)"
  },
  "typeName": {
    "id": 70,
    "name": "address",
    "nodeType": "ElementaryTypeName",
    "src": "1946:7:0",

```

```

"typeDescriptions": {}
}
},
"id": 73,
"isConstant": false,
"isLValue": false,
"isPure": true,
"kind": "typeConversion",
"IValueRequested": false,
"names": [],
"nodeType": "FunctionCall",
"src": "1946:10:0",
"tryCall": false,
"typeDescriptions": {
"typeIdentifier": "t_address",
"typeString": "address"
}
},
"src": "1934:22:0",
"typeDescriptions": {
"typeIdentifier": "t_bool",
"typeString": "bool"
}
},
{
"devdoc": {
"details": "Contract module which provides a basic access control mechanism, where
there is an account (an owner) that can be granted exclusive access to specific functions.
By default, the owner account will be the one that deploys the contract. This can later be
changed with {transferOwnership}. This module is used through inheritance. It will
make available the modifier `onlyOwner`, which can be applied to your functions to
restrict their use to the owner.",
"kind": "dev",
"methods": {

```

```

"constructor": {
"details": "Initializes the contract setting the deployer as the initial owner."
},
"owner()": {
"details": "Returns the address of the current owner."
},
"renounceOwnership()": {
"details": "Leaves the contract without owner. It will not be possible to call `onlyOwner`
functions anymore. Can only be called by the current owner. NOTE: Renouncing
ownership will leave the contract without an owner, thereby removing any functionality
that is only available to the owner."
},
"transferOwnership(address)": {
"details": "Transfers ownership of the contract to a new account (`newOwner`). Can only
be called by the current owner."
}
},
"version": 1
},
"userdoc": {
"kind": "user",
"methods": {},
"version": 1
}
}

```

Appendix II: Model Evaluation Tool (The Likert Scale Tool)

Area of Model Evaluation	Assessment Criteria	Strongly Disagree	Disagree	Not Sure	Agree	Strongly Agree
Persistence	<ul style="list-style-type: none"> The model to offer Protection from any modifications or attacks is well maintained? 					
	<ul style="list-style-type: none"> How does the model store digital ledgers in the system? 					
	<ul style="list-style-type: none"> When adding a transaction does every node need to check its validity? 					
	<ul style="list-style-type: none"> Once the transaction blocks get added on the ledger, can it just be changed by anyone? 					
Time-stamping of Records	<ul style="list-style-type: none"> Does the Model secure encrypted data at the source in a way that can't be altered? 					
	<ul style="list-style-type: none"> The protocol used maintains consensus (tracks activities) on the Blockchain model? 					
	<p>The model is reliable and secure, in that activities cannot be changed or removed once published to the ledger?</p>					
Transparency	<ul style="list-style-type: none"> Does the model facilitate a trust less setting in that every member in the network has a duplicate or precisely the same information as a disseminated record? 	339				

	The Model tracks every change on the Blockchain, makes it visible to make it more reliable?					
	<ul style="list-style-type: none"> The model improves data recovery by ensuring each party has a timely and shared impression of the data? 					
	<ul style="list-style-type: none"> The profile of every participant is transparent and every change on every block model is viewable by all members? 					
Decentralization	<ul style="list-style-type: none"> Does the model rely on a central point of control? 					
	<ul style="list-style-type: none"> Due to its decentralized nature, can the model survive any malicious attack? 					
	<ul style="list-style-type: none"> Does the model rely on third party companies to facilitate transactions? 					
	<ul style="list-style-type: none"> The decentralized nature of Technology, does the model allow a transparent profile of every participant. 					

Appendix III: Letter from Institute of Postgraduate Studies



KABARAK UNIVERSITY

INSTITUTE OF POST GRADUATE STUDIES

Private Bag - 20157
KABARAK, KENYA
E-mail: directorpostgraduate@kabarak.ac.ke

Tel: 0773265999
Fax: 254-51-343012
www.kabarak.ac.ke

23rd October, 2018.

Ministry of Higher Education Science and Technology,
National Council for Science, Technology & Innovation,
P.O. Box 30623 - 00100.

Dear Sir/Madam,

RE: RESEARCH BY EVANS KIMUTAI CHESANG-GDS/M/0175/01/17

The above named is a student of Kabarak University taking Doctor of Philosophy in Information Technology Security and Audit. His research entitled "A Secure Title Deed Registration Model for Land Transactions in Kenya Based on Blockchain." He has been **Examined and Accepted** by the Board of Postgraduate Studies.

He is therefore authorised to proceed on with his research. Any assistance accorded to him is highly appreciated

Thank you.

Yours faithfully,



Dr. Betty Elikoro
DIRECTOR (POST GRADUATE STUDIES)

Kabarak University Moral Code

*As members of Kabarak University family, we purpose at all times and in all places, to set apart in one's heart, focus
on God. (1 Peter 3:15)*



Kabarak University is ISO 9001:2015 Certified

Appendix IV: NACOSTI Reserach Authorization Letter



NATIONAL COMMISSION FOR SCIENCE, TECHNOLOGY AND INNOVATION

Telephone:+254-20-2213471,
2241349,3310571,2219420
Fax:+254-20-318245,318249
Email: dg@nacosti.go.ke
Website : www.nacosti.go.ke
When replying please quote

NACOSTI, Upper Kabete
Off Waiyaki Way
P.O. Box 30623-00100
NAIROBI-KENYA

Ref: No. **NACOSTI/P/18/45524/26608**

Date: **13th December, 2018**

Evans Kimutai Chesang
Kabarak University
Private Bag - 20157
KABARAK.

RE: RESEARCH AUTHORIZATION

Following your application for authority to carry out research on *“A secure title deed registration model for land transactions in Kenya based on blockchain”* I am pleased to inform you that you have been authorized to undertake research in **Nairobi and Nakuru Counties** for the period ending **26th November, 2019**.

You are advised to report to **the County Commissioners and the County Directors of Education, Nairobi and Nakuru Counties** before embarking on the research project.

Kindly note that, as an applicant who has been licensed under the Science, Technology and Innovation Act, 2013 to conduct research in Kenya, you shall deposit a **copy** of the final research report to the Commission within **one year** of completion. The soft copy of the same should be submitted through the Online Research Information System.

**GODFREY P. KALERWA MSc., MBA, MKIM
FOR: DIRECTOR-GENERAL/CEO**

Copy to:

The County Commissioner
Nairobi County.

The County Director of Education
Nairobi County.


Appendix V: NACOSTI Reserach Permit


THIS IS TO CERTIFY THAT:
MR. EVANS KIMUTAI CHESANG
of KABARAK UNIVERSITY, 0-20157
Kabarak, has been permitted to conduct
research in Nairobi, Nakuru Counties


Permit No : NACOSTI/P/18/45524/26608
Date Of Issue : 13th December,2018
Fee Recieved :Ksh 2000

on the topic: A SECURE TITLE DEED
REGISTRATION MODEL FOR LAND
TRANSACTIONS IN KENYA BASED ON
BLOCKCHAIN

for the period ending:
26th November,2019


Applicant's
Signature


Director General
National Commission for Science,
Technology & Innovation




THE SCIENCE, TECHNOLOGY AND INNOVATION ACT, 2013


The Grant of Research Licenses is guided by the Science, Technology and Innovation (Research Licensing) Regulations, 2014.

CONDITIONS

1. The License is valid for the proposed research, location and specified period.
2. The License and any rights thereunder are non-transferable.
3. The Licensee shall inform the County Governor before commencement of the research.
4. Excavation, filming and collection of specimens are subject to further necessary clearance from relevant Government Agencies.
5. The License does not give authority to transfer research materials.
6. NACOSTI may monitor and evaluate the licensed research project.
7. The Licensee shall submit one hard copy and upload a soft copy of their final report within one year of completion of the research.
8. NACOSTI reserves the right to modify the conditions of the License including cancellation without prior notice.

National Commission for Science, Technology and innovation
P.O. Box 30623 - 00100, Nairobi, Kenya
TEL: 020 400 7000, 0713 788787, 0735 404245
Email: dg@nacosti.go.ke, registry@nacosti.go.ke
Website: www.nacosti.go.ke


REPUBLIC OF KENYA


National Commission for Science,
Technology and Innovation
RESEARCH LICENSE

Serial No.A 22402
CONDITIONS: see back page

Appendix VI: Evidence of Conference Participation



ST. PAUL'S UNIVERSITY

Certificate of Participation

This is to Certify that

Evans Kimutai Chesang

Presented a paper titled:

**Towards a Secure Title Deed Registration Model for
Land Transactions in Kenya Based on Blockchain.**

**At the St. Paul's University Annual International Research
Conference (SPUAIRC), held at St. Paul's University on
22nd – 25th October, 2024**

**Conference Theme: Leveraging Research, Technology and
Innovation In The New Economy For Sustainable Development**

**Dr John Muhoho
Conference Chair**

**Prof. Abraham K. Waithima
Deputy Vice-Chancellor
Academic Affairs**

Appendix VII: List of Publications



IMPLEMENTING A SECURE TITLE DEED REGISTRATION MODEL FOR LAND TRANSACTIONS IN BLOCKCHAIN

Authors

Evans K. Chesang ⁽¹⁾ ; Nelson Masese ⁽²⁾ ; Peter Rugiri ⁽³⁾

Main author email: echesang@kabarak.ac.ke

(1.2.3) Kabarak University, Kenya.

Cite this article in APA

Chesang, E. K., Masese, N., & Rugiri, P. (2024). Implementing a secure title deed registration model for land transactions in blockchain. *Journal of computer science and technology*, 2(1), 32-40. <https://doi.org/10.51317/jcst.v2i1.618>



A publication of Editon Consortium Publishing (online)

Article history

Received: 24.08.2024

Accepted: 23.09.2024

Published: 24.10.2024

Scan this QR to read the paper online



Copyright: ©2024 by the author(s). This article is an open access article distributed under the license of the Creative Commons Attribution (CC BY NC SA) and their terms and conditions.



Abstract


The purpose of this study was to implement a secure title deed registration model for land transactions in blockchain. The non-existence of a universal National Land Information Management System in Kenya has generated inconsistencies in land transactions, thus creating a need for a secure title deed registration model for land transactions. The study adopted a proof of concept (PoC) methodology to demonstrate the concept of using blockchain technology to resolve existing challenges currently experienced in land transactions. The outcome of the PoC methodology was the design of a model that was later evaluated for implementation. The design architecture of the model was based on Ethereum, an open-source, decentralised blockchain platform used to establish a peer-to-peer network that securely executes and verifies an application code referred to as a smart contract that enables network users to execute land transactions amongst themselves with no trusted central authority. Selected users evaluated the designed model against the metrics set out by the study. Upon model evaluation, its results provided a trail of non-immutable and verifiable land transaction records visible to all the parties involved. The land transaction records were securely distributed across the peer-to-peer network, providing participants with visibility of all the transaction data. Therefore, the model implemented in the study demonstrated that blockchain technology has an immense potential of executing secure and transparent land transactions across Kenya. This study recommends that future research should explore implementation strategies and regulatory frameworks for integrating blockchain technology into land registration.

Key terms: Blockchain, decentralisation, immutable ledger, lands registry, transparency.



Issue no: 1 | Vol no: 1 | October 2024: 13-20

Designing a secure title deed registration model for land transactions in Kenya based on blockchain

Evans K. Chesang⁽¹⁾ 

Nelson Masese⁽²⁾ 

Peter Rugiri⁽³⁾ 

(1,2,3) Kabarak University, Kenya.

Main author email: echesang@kabarak.ac.ke

Article History
Received: 2024-08-23
Accepted: 2024-09-24
Published: 2024-10-25

Cite this article in APA

Chesang, K. E., Masese, N., & Rugiri, P. (2024). Designing a secure title deed registration model for land transactions in Kenya based on blockchain. *Editon consortium journal of engineering and computer science*, 1(1), 13-20. <https://doi.org/10.51317/ecjecs.v1i1.539>

Abstract

The purpose of this study was to design a secure title deed registration model for land transactions in Kenya based on Blockchain. Blockchain technology creates a publicly accessible distributed database that keeps track of all land transaction records, allowing confirmation of land ownership at any given point in time with the aim of creating a secure and transparent environment that guarantees the much-needed confidentiality and integrity of land transactions. Design Science research methodology was adopted by the research to provide an overall scope in the model design and development requirements, while the Proof of Concept (PoC) research methodology was used to design and develop the essential modules for the model. The research employed subjective sampling to select users to perform a criteria-based model evaluation approach against the evaluation metrics (persistence, time stamping, transparency and decentralisation) set out by the research. Results were presented in the form of a descriptive summary of the users' feedback. From the results, the study established that the model could effectively execute immutable, time-stamped, transparent and decentralised land registration and land transaction processes that can be used as authentic proof of land ownership. The study concluded that the proposed model enhances transparency and security, contributing to a more reliable system that fosters public confidence. The study recommends that future research may focus on the integration of Blockchain with land taxation systems as well as the adoption of cryptocurrency as a legal tender to facilitate the execution of smart contracts for properties in Kenya.

Key words: Blockchain technology, NLMIS/Ardhisasa, title deed registration, secure model, land transactions.



This article is distributed under the license of a [Creative Commons Attribution-Non Commercial-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-nc-sa/4.0/). It is permitted to be used, reproduced and distributed in line with Editon Consortium Publishing guidelines. The work without further permission provided the original work is attributed as specified on the SAGE and Open Access pages