BLOCKCHAIN BASED SMART CONTRACT MODEL FOR REAL ESTATE MANAGEMENT

ALEX KIBET

A Research Thesis submitted to the Institute of Postgraduate Studies for Partial Fulfillment of the Requirements for Master of Science in Information Technology of Kabarak University.

KABARAK UNIVERSITY

NOVEMBER 2019

DECLARATION

I hereby declare that this research thesis is my original work and has not been submitted to any other university or college for the purpose of examination or academic award. Any information is given in this entire thesis and all the relevant sources are quoted and acknowledged accordingly

Signature ………………………………… Date …………………………….

Alex Kibet
GMI/NE/0221/01/18

RECOMMENDATION

The research Thesis entitled **‘Blockchain Based Smart Contract Model for Real Estate Management’** written by Alex Kibet is presented to the Institute of postgraduate studies and research of Kabarak University. We have reviewed this thesis and recommended it be accepted in partial fulfillment of the requirements for the Degree of Master of Science in Information Technology.

Signature ……………………………………… Date …………………………….
Prof. Simon Karume
Department of Computing and Informatics
Laikipia University.

Signature ……………………………………… Date …………………………….
Dr. Moses M. Thiga
Department of Computer Science
Kabarak University

COPYRIGHT

ACKNOWLEDGMENTS

Firstly, I wish to I praise God for the gift of life, and provision without which I would not have made it this far in my academic pursuits. Indeed, the ability to read and comprehend is a gift that only you God give. All honor, praise, and glory are to God. Secondly, I would like to express my sincere gratitude to my supervisors Prof. Simon Karume and Dr. Moses Thiga for the constant support during my studies. Their patience, inspiration, immense knowledge sharing, wise counsel and input during my research have made the writing of this thesis possible. As my teachers and mentors, they have taught me more than I could ever give them credit from here. They have shown me, by their example, what a good scientist (and person) should be. Besides my supervisors, I would like to thank all of those whom I have had the pleasure to work with during this and other related academic aspects. I am grateful to my classmates Komen, Jared, Leah, And Tony for their not only insightful comments and encouragement but also for the hard question which incented me to widen my research from various perspectives. Nobody has been more important to me in the pursuit of my studies than the members of my family. I would like to thank my parents, whose love and guidance are with me in whatever I pursue. They are the ultimate role models. Most importantly, I would wish to thank the entire Kabarak university for the opportunity to study in Kabarak and all the support that they accorded me during my studies.

DEDICATION

I dedicate this research thesis to my parents who have always loved me unconditionally and whose good examples have taught me to work hard for the things that I aspire to achieve and greatly taught me that even the biggest assignment can be accomplished if it is done a single step at a time. It is also dedicated to my friends Makupi Dan, David Mbugua, Njoroge Maina, Joe Karume, Damian Kipkoech, Dylan Chirchir, Mercy Masungo and Jared Nyakamba who have always been a constant source of support and encouragement during the challenges of my academic life. Also to my brothers and Aunts who am truly grateful for having in my life. This work is also dedicated to all those who might find it useful.

ABSTRACT

Industries worldwide are facing a technological revolution where there is a need for technologies that can speed up business processes with more safety and transparency. The real estate industry is among many other industries that would benefit from such technologies due to the magnitude of financial transactions handled. The current management strategy employed by real estate owners depends on a number of intermediaries, including brokers, agents and banking service providers. This strategy of operation results to inadequacies within the real estate industry that cause problems such as lack of transparency, high transaction costs, personal biases, tax evasion or under taxation, landlord versus tenant conflicts and slow transaction processes. This thesis examines the potential of implementation of blockchain-based smart contract technology in the real estate industry and how it might resolve the inefficiencies within the industry. Blockchain is a new and emerging technology with the potential for implementation in various industries. Previous research in blockchain technology has concentrated on its potential application in digital currency. In this study, the researcher endeavored to design, develop and validate a blockchain-based smart contract model for management of real estate property that would address the weaknesses of the existing management models and potentially reduce the housing cost by elimination middlemen in the management process. To achieve the study objectives, a proof of concept (POC) methodology was adopted to prove the concept of resolving the weaknesses of the existing models for real estate property management using a blockchain-based smart contract model. The proof of concept was carried out in four steps namely planning, designing, implementation and improvement (PDIOI). The POC focused on coming up with a Solidity Smart Contract Based on Ethereum Model for Housing Industry available at grandmullah.github.io/real-estate. To demonstrate blockchain transactions the researcher created a cryptocurrency “EToken” that would be used as a medium of exchange. The researcher also acquired test ethers from Ethereum Rinkeby faucet for paying transaction validation fee. The smart contract model was able to transfer ETokens from tenant account to the landlord account. All the transactions were recorded in blockchain distributed ledgers upon successful validation, and the entire Ethereum network users would view allowing transparency and integrity of transactions. As a generalisation and extension into other domains, the researcher noted that a blockchain-based model is important in any organisation that values historical transactions such as organisation which provides financial service. As a recommendation for further study, research can be extended to creating a completely blockchain-based version of electronic real estate transaction records, including land registry, or real estate ownership transfers.

Keywords: *Blockchain, Ethereum network, smart contract, Real Estate, consensus algorithm.*

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

ASD -	Adaptive software development
BTP -	Bit-property
DLT -	distributed ledger technology
EVM -	Ethereum Virtual Machine
GBP –	Great British Pounds
GDP -	Gross domestic product
KARA -	Kenya Alliance Residents Association
KBS-	Kenya Bureau of Statistics
KRA-	Kenya Revenue Authority
KTA -	Kenya tenants Association
NACOSTI-	National Commission for Science Innovation and Technology
PDIOI-	Plan, Design, Implement, Operate, Improve
POC -	Proof of concept
POS –	point of sale
UTAK -	Urban Tenants Association of Kenya

OPERATIONAL DEFINITION OF TERMS

Blockchain A blockchain is a digitized, decentralized, public ledger of all cryptocurrency transactions. Constantly growing as ‘completed’ blocks (the most recent transactions) are recorded and added to it in chronological order, it allows industry participants to keep track of digital currency transactions without central recordkeeping. Each node (a computer connected to the network) gets a copy of the blockchain, which is downloaded automatically (Fahmy, 2018).

Ethereum Network Ethereum is a decentralized platform that runs smart contracts: applications that run exactly as programmed without any possibility of downtime, censorship, and fraud or third-party interference (Hjalmarsson *et al*, 2018).

Ethers Ethers are the integral element of the Ethereum blockchain network that acts as the network's fuel, keeping it agile and functional. While many believe that ether is the native digital currency of Ethereum, it acts as a medium of incentive or form of payment for the network participants to execute their requested operations on the network (Tran, 2018).

Model Is a program that runs on a computer that creates a prototypical, or simulation, of a real-world feature, phenomenon or event (Finley, 2017).

Smart Contract Smart contracts are self-executing contracts with the terms of the agreement between buyer and seller being directly written into lines of code. The code and the agreements contained therein exist across a distributed, decentralized blockchain network. Smart contracts permit trusted transactions and agreements to be carried out among disparate, anonymous parties without the need for a central authority, legal system, or external enforcement mechanism. They render transactions traceable, transparent, and irreversible (Preston *et al*, 2018).

Solidity Solidity is a contact-oriented, high-level language for implementing smart contracts. It is designed to target the Ethereum Virtual Machine (EVM) (Grishchenko *et al*, 2018).

CHAPTER ONE
INTRODUCTION

1.0 Introduction

This chapter provides a background of the fundamental concepts and problems informing this study specifically on the frictions in Claiming of deposit, intermediaries, Enforcement of tenants' rights and obligations, Conflict between tenants and landlord overpayment of rent or cases of Unexplained Rent Increments. It further continues to outline the research objectives, listing the research questions, state the research problem, defines the scope, assumptions, significance and the contribution of the study.

1.1 Background of the Study

Technological revolution worldwide is rapidly changing the way of doing business. This has resulted in business entities reengineering their business processes and improving the efficiency of operations. According to Tapscott (2016), technology not only facilitates efficient and effective operation with minimal manpower and helps to reduce the overall operational cost but also speeds up the different processes and bring more safety and transparency into the industry. The real estate industry is equally affected. The importance of real estate industry emanates from the fact that it deals with housing which is an essential service that is highly important for the overall well-being of a country. Ansar (2018) argues that real estate problems often affect and creates challenges which in turn could lead to economic meltdowns not only on the micro but also on a macro level. The real estate challenges are in many cases linked to inefficiencies within the market. Inefficiencies such as illiquid market, transparency issues, high transaction costs, personal biases, intermediaries and slow transaction processes (Franke & Krahnen, 2017).

The overall cost of renting a house depends mainly on the market structure and the management strategy employed by real estate owners. The current management strategy is such that part-time landlords choose to work with middlemen to bring their rental properties to market (Pattillo, 2017). Most large rental corporations often choose to work with freelance middlemen to market their properties, vet potential tenants, and take care of maintenance

(Ziemann, 2017). This increases the overall rental fee without contributing any significant value to the final consumer (Tenant). In the real estate market, the tenant is often the least well-informed and with the least wealth player (Scott, 2018). In combination, this leads to a power imbalance to Tenant's disadvantage. Though, strict laws designed to protect tenant's rights exist, property owners and middlemen often find a way around them (Chordia, 2018). From this, cases of unexplained rent increment or unequal rental fee for tenants on the same rental house is common.

The most fundamental transaction in the housing market is the rental contract. Property owners and tenants agree that the tenant will be allowed to live in and use a certain allocation of space within the building (Layard, 2018). For this, the tenant pays the property owner (landlord or landlady) a certain sum of money in an agreed-upon period of time. In most countries, rent is paid fortnightly or monthly, but shorter and longer payment intervals are possible too (Hills, 2017). Currently, rent can be paid via bank transfer, credit card, cash, or in any other form that owner or agents and tenant agree upon. This means of rental payment and management strategy has led to rental fee increment and high level of bureaucracy where cases of minimum balance requirement, transaction costs, early notifications, and middlemen are common.

According to Social Networks and the Success of Market research conducted by Crowston *et al*, (2015) real estate transactions presently depend on a number of intermediaries, including brokers, dealers, agents, banks service providers, etc. In the short term, sharing contracts and approvals in real time will reduce delays caused by mailing and delivery. Indeed, Goldman (2018) estimated that blockchain technologies could lead to a yearly savings of 2-4 billion USD in the real estate market alone. It would also eliminate the need for parties to reconcile documents, as all parties maintain an identical, immutable copy of transaction records. In addition, several times consuming, expensive functions can be replaced with blockchain and smart contracts. Payments of rent, deposits, and fees could be automated. Escrow accounts could be restructured around smart contracts and multisig wallets (Veuger & Jan 2018). The same infrastructure could be harnessed for other transactions that occasionally require resolution by a neutral party, such as disputes over rent deposits. In the longer term, blockchain-based registries could allow peer-to-peer asset transfers or payment of rental fee, reducing transaction times from months or weeks to minutes. Transaction costs could come down from thousands of dollars per sale or lease to a modest service fee. The ease and security of transactions could also permit the efficient unbundling of property rights. A landlord could lease a house quickly and cheaply.

It takes on average a whole day to finish a bank transaction if queues are long (Flink, 2017). Rental payments move from tenants account to agent's account that after deducting his/her commission sends to the real estate owner. This process is time-consuming and inefficient. Crowston & Wigand, (2016) argue that a real estate transaction usually involves three parties; Agent, Landlord, and Tenant which in turn results to transaction costs of constituting up to 10% of the total rental fee according to (Lantmäteriet *et al,* 2016) (see figure 1).

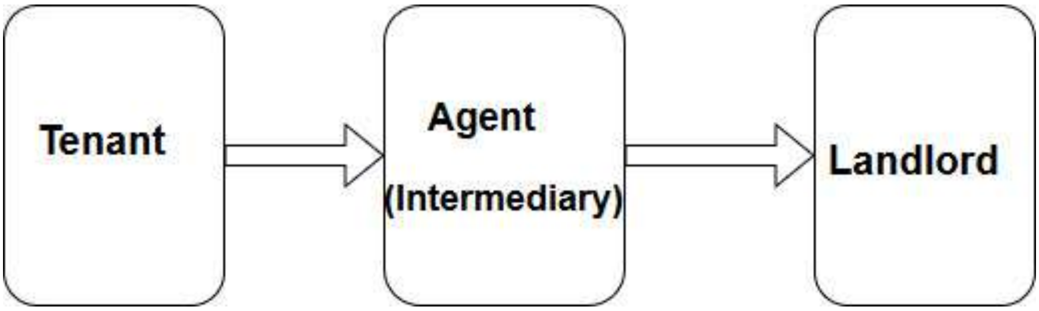


Figure 1present real estate rent transaction process (Corluka & Lindh, 2017)

Lantmäteriet *et al,* (2016) argues that this system is slow due to several reasons; the major one is due to the repeated process of validating the information. Many of the documents are signed on paper and need to be sending from agent to Landlord. The validation of the documents needs to be done through manual processes. Due to the number of signed papers, mistakes occur and needs to be corrected. Chances of fraud and corruption are possible. All these combined make the process slow and inefficient due to the time-consuming process, high transaction costs, and frauds.

Considering that having an affordable home is vital for people (Tatjana *et al,* 2014). There is a need to look at the current housing market structure and management strategy for more improvement. Though some tenants are contented with their current rental house, landlord or letting agent, others, including those on little incomes are undergoing problems with a substantial impact on their lives (Yates *et al,* 2017). Although policymakers are still working to mitigate the immediate effects of the dramatic downturn in the developing housing market, they also must prepare for the challenges that lie ahead (Stone, 2010). According to Kintrea (2014), high friction transactions are hardwired into the structure of modern real estate markets. As a result, the current infrastructure in the sector is slow, expensive, and brittle. For the rental payment, transaction costs can constitute up to 10% of the total rental fee. Entire industries have emerged for the sole purpose of capitalizing on the inefficiencies that exist around transactions. The situation in less developed markets is time and again even more cumbersome (Felson, 2017).

With blockchain technology, many of these inefficient activities could be reduced or excluded from the process (see figure 2).

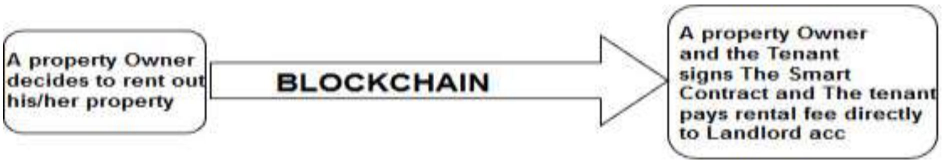

A property Owner decides to rent out his/her property
BLOCKCHAIN
A property Owner and the Tenant signs The Smart Contract and The tenant pays rental fee directly to Landlord acc

Figure 2: A transaction processes based on blockchain technology (Barkham et al, 2018)

Lantmäteriet and others (2016) argue further that blockchain technology is a much more secure technology to use compare with todays. In this new technology fraud and lost documents cannot occur or are minimized and the whole transaction process could be significantly reduced. Most governments raise income through tax to enable them to finance development projects that are meant to improve the economy of a given country or the region. In most countries like Kenya, Rwanda, and Tanzania, house rental taxation has been classified as economic. In developing nations, statistics indicate that less than half of property owners and developers comply with rental income tax requirements (Sirr *et al*, 2017). Regardless of sanctions like penalties, armed monitoring, routine audits and fines, lack of documentation to proof exact amount of rental income to base on when computing payable tax is a limitation to revenue collections (Karanja, 2015). According to Mwangi (2014), one of the main tax evasion reasons in personal income generating is lack of correct reference information such as rent income details, which tend to lead taxpayers to evade tax. It’s therefore important to put in place a mechanism that will help in ensuring that all landlords provide correct information to compute taxes (Ayuba, Saad &Ariffin, 2016). This would improve the current situation where landlords file tax based on paper based records.

Blockchain is a new and emerging technology on the market. It is an information technology with multiple classes of application for example, any form of asset registry, exchange or inventory (Swan, 2015). Blockchain is a decentralized transaction technology first developed for Bitcoin cryptocurrency (Yli-Huumo *et al*, 2016), but every area of economics, finance, and money could be included in the technology. The technology uses ledgers (a distributed database architecture) to register any kind of information (Pinna & Ruttenberg, 2016) and has the potential to shorten the transaction time in any kind of transaction and make systems more transparent and reliable.

1.2 Statement of the Problem

The current management strategy employed by real estate owners depends on a number of intermediaries, including brokers, agents and banking service providers. The process of renting a house involves a great deal of bureaucracy and estate intermediary fees. When a property is let, the agreed terms such as rental duration and monthly payments due are laid out in tenancy agreements; a predominantly paper-based and hand-signed form of contract. This has led to slow, expensive, and brittle transactions with transparency and integrity issues. Problems with the current system involve the excess of paper documents required, lengthy bank transfer times, and high estate agent fees. Research proves that it can take up to a day when transferring funds between different banks or longer for larger amounts, which can incur extra charges. Furthermore, the current management strategy has a potential of transaction records manipulation such that the government tax authorities might not have clear or true transaction records for auditing and filing taxes. This warrants or permits tax evasion or under filling of tax as well as having false information for government tax administrations, and tax advisory organizations for making tax-related decisions. While there is a wide range of technologies that have been employed for real estate and housing market, there is a need for technologies to allow Immutable and Autonomous payment of a rental fee in a transparent, conflict-free way while avoiding the services of a middleman.

1.3 Purpose of the Study

The primary concern of the study was to come up with a blockchain based smart contract model to reduce operational costs while managing a rental house and consequently lowering the rental fee. This would be realized by reducing the current great deal of bureaucracy around real estate property management.

1.4 Objectives of the study

i) To explore the weakness of existing models used in real estate property management.
ii) To design a blockchain based smart contract model for real estate property management.
iii) To develop the blockchain based smart contract prototype for real estate property management.
iv) To validate and verify the implemented smart contract model prototype

1.5 Research Questions

The research seeks to answer the following questions;

i) What are the existing models used in managing real estate property?
ii) How can a model based on a blockchain smart contract for the real estate industry be designed?
iii) How can blockchain based smart contract prototype for the real estate industry are implemented?
iv) How can the implemented blockchain based smart contract model prototype be validated and verified?

1.6 Research Contributions

The output of this study includes the following deliverables;

i) A report of the existing real estate property management models.

ii) Model design for the real estate industry based on blockchain smart contract.

iii) Prototype of a blockchain based smart contract implementation model to assist real estate crews.

iv) Validation and Verification report on the functionality of the real estate blockchain smart contract model prototype.

1.7 Justification of the Study

Frictions in a process are hardwired into the market structure, for the rental fee, transaction costs can constitute up to 10% of the total rental fee (Swan & Melanie, 2015). Entire industries have emerged for the sole drive of getting the most out of the inefficiencies that exist around transactions. The situation in less advanced markets is time and again even more cumbersome. According to Reiss (2012), Real estate transactions currently rests on a number of intermediaries, including brokers, dealers, agents, and banks service providers, etc. The unstructured nature of transactions leads to an exasperating situation by increasing the cost of doing business. The use of smart contract explored in this research reduces transaction cost and reduces the chaotic business in housing market industry. Accordingly, reducing transaction cost in the long run saves millions of money lost in expensive procedural transactions. At the higher levels, blockchain based tamper proof transaction records showed in the developed model can be used by the government tax authorities for auditing and filing taxes. This will reduce the cases of tax evasion as well as helping government tax administrations, and tax advisory organizations in making tax-related decisions.

1.8 Scope of the Study

This research primarily focused on developing a blockchain based smart contract model for the real estate industry to allow Immutable payment of a rental fee in a transparent, conflict-free way while avoiding the services of a middleman. A review of existing models for managing real estate property was done. A blockchain based smart contract model was designed, implemented and validated to prove the potential viability of blockchain based application in real estate as a solution to issues mentioned in this study. Two rental agents were involved in the evaluation to determine the model viability and practical potential of smart contract in real estate property management. The entire research used PDIOI research design by Huawei and the system of smart contracts was designed based on a five-type model by Monax. The model was developed with the lite server, truffle framework, nodeJS and web3 JSON RPC(s) in interactive functions. The model was designed to ensure security and immutable traceability of transaction records. The model was tested on Ethereum test networks to reduce the research cost and ensure a safe environment for testing and experimenting. It is worth highlighting, however, that no security or privacy liabilities outside of the blockchain have been resolved with this implementation and that blockchain based smart contract model for real estate is not a substitute for real estate agents. It can only be used to ensure transparency and traceability of the rental fee.

1.9 Limitations of the Study

Although the general realization by developing blockchain based smart contract model for real estate industry in this study there were, some limitations which included:

i) Limited test ethers: faucet gives out single ether for testing in 18 hours. This slowed the development process of the model.

ii) The Criminal Connection**:** Since its launch, Bitcoin has long been associated with the shadowy dealings of the black market and the dark web. Because this is the first interaction of the public with blockchain technology, this connection has persisted with Bitcoin, altcoins, and the technology underlying it as well. This resulted to model resistance during testing and evaluation.

1.10 Assumptions of the Study

i) The study assumed that the model developed can be scaled to other domains with regard to the emerging decentralized economic system that is cryptocurrencies.

ii) The study also assumed that the government will implement a policy to guide and issue a legal tender to cryptocurrencies.

CHAPTER TWO

LITERATURE REVIEW

2.0 Introduction

This section presents existing real estate industry management models and comprehensive overview of blockchain technology, followed by an in-depth look at Ethereum network and Smart contract. The conceptual frameworks and model implementation overview for the study will also be presented and discussed.

2.1 Real Estate Management

Real estate management involves putting in place procedures to ensure the smooth running of a real estate property (Cooke *et al.*, 2018). In tenant-focused property management, firms are committed to creating a safe and enjoyable experience for those in their communities. They do this by offering the highest quality properties at competitive pricing and by forging sincere relationships with their tenants. Precisely, their goal is to create positive experiences for tenants. The major areas of responsibility include; Administration and Risk Management, Tenant and Occupancy. However, in the digital world, “Real estate management” involves the use of technology to manage real estate property. The real estate transactions records are saved and managed in a standard format by assurances needed to complete transactions (Ullah *et al.,* 2018).

Real estate is fundamental to everything we do; we need real estate property to live in, work in, and for leisure. It is not only an important commercial and industrial resource but is also home to many national and international investment funds. According to Hugo (2018), today, more people live in cities than in rural areas, and urbanization continues to grow around the world. This increases the magnitude of financial transactions and transparency challenges. Therefore, a technology to manage such transaction records is needed. This thesis explored blockchain technology to aid in reducing such challenges.

2.2 Existing Models Used In Real Estate Property Management

To appreciate the importance of technology products, there needs to be a reason for the technology to exist. Therefore, to justify the need for a new blockchain smart contract, the existing real-world alternative needs to exhibit inefficiencies in their use cases. That is why blockchain technology is rapidly hitting the mainstream and according to a World Economic Forum survey of executives and ICT experts, 57.9% of the respondents believe that 10% of the global GDP information will be stored on blockchain technology by 2025 (Kuznetsov *et al*, 2018). For that case Industry players now realize that blockchain-based smart contracts can play a much larger role in real estate dealings, potentially transforming core real estate operations such as property transactions (Wong, 2018). The following are existing smart contract systems for real estate management that have been analyzed by the researcher.

2.2.1 E-Resident Platform in Kenya

KCB Bank Kenya Limited in partnership with E-Resident Limited designed E-resident Platform to aid in the management of rental payment for the real estate market (Daniels, 2018). E-Resident has been publicized to offer a host of benefits such as security, convenience and cut collection cost for landlords and agents. The benefits apply to tenants too who will not have to deal with agents or queue at the bank to make payments. The platform also makes tracking of payments easier since the money is sent from any of the money transfer platforms straight to the bank. Accounting functions such as reconciliation make it easier for agents and property owners to see in real-time the amount paid and the amount due at any given time. This innovation was crafted to cut across multiple payment channels such as credit and debit card, mobile money transfer as well as bank deposits. This model also designed to ensure customer’s right to privacy and right to access information with lawful restrictions therein as provided for in the Constitution of Kenya.

2.2.2 SPENN Mobile-based Platform

SPENN is a mobile-based platform for banking application that was designed to enable Rwandans to make a range of financial transactions (bizimungu, 2018). SPENN was developed by Block-bonds, a Norwegian financial technology company. The application allows smartphone users to transact and pay for goods and services. Unlike many other blockchain technology platforms, SPENN uses a different model whereby it removes the need for cash by digitalizing national currencies. As a user, after downloading and registering with the app, you immediately have a digital account and the ability to transact and interact with others in the financial system. Using blockchain technology, Block-bonds is bringing ease in transacting as SPENN account holders are able to keep track of the transactions and funds on their accounts. The mobile banking platform doesn’t only target the unbanked but also the banked to ultimately bring efficiency and effectiveness in the way people transact and make payments in the country.

2.2.3 Deed-coin Agent Location

Deed-coin platform allows searching for a real estate agent. It replaces the way you find your next real estate agent. Instead of finding next agent on google for higher commission, customers access deed-coin platform, input their property information, and link up with their local Deed agent for a 1% commission. Deed customers use DEED to decide the commission or rebate before linking with the Deed agent (Arruñada, 2018). All discounts are reflected on the standard HUD forms during a property closing and all homes are listed or purchased through standard MLS. Customers have access to a groundbreaking technology dashboard allowing 24/7 access to an agent or licensed transaction assistant. DEED tokens will access a better real estate network that is already usable in 130+ cities and all 50 states. This streamlines real estate to run on 1%, allowing future commissions to be set by the free market based on the price of DEED.

2.2.4 The Bit-property Platform.

Bit-property is a decentralized real estate platform powered by Ethereum smart contracts that aim at giving users a seamless, quick, and low-cost way to own and trade real estate revenue streams (Bitproperty, 2017). It has two types of token: the BTP platform token and individual property asset tokens. BTP tokens represent the inherent value of the platform and provide holders with income from transaction fees within the platform in proportion to the amount they own. BTP tokens also pay income from the energy revenues of Bit-property's solar farm. Asset tokens will be issued for each property listed on the platform. The number of tokens issued against a specific property represents the value of that property. Token holders get income dependent on the performance of the asset and the stake that they hold.

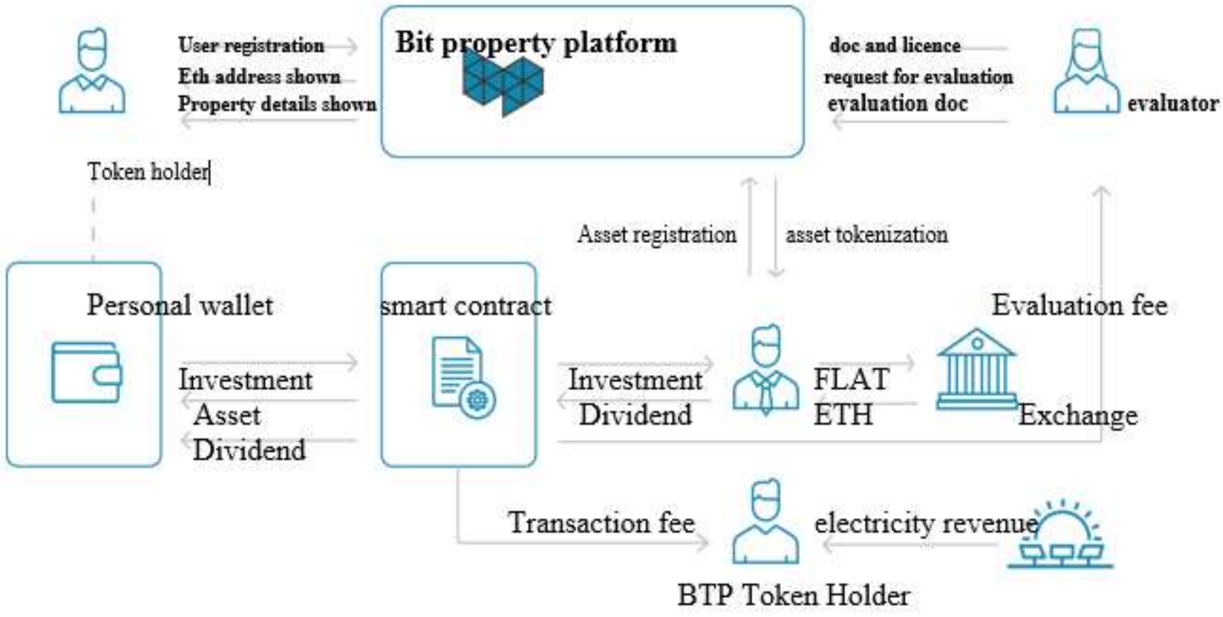


Figure 3 Bitproperty Platform (Bitproperty, 2017)

2.3 Design of Blockchain Based Smart Contract Models

The constituent parts of a blockchain based model include Blockchain, distributed ledger, consensus algorithm, smart contracts, and cryptocurrency. These are discussed in this section

2.3.1 Blockchain Overview

Blockchain is technology originally developed as the accounting method for the virtual currency Bitcoin (Eskandari *et al.*, 2018). Blockchain uses distributed ledger technology (DLT) and is appearing in a variety of commercial applications today (Lipton, 2018). Currently, the technology is primarily used to verify transactions, within digital currencies though it is possible to digitize code and insert practically any document into the blockchain. Doing so creates an indelible record that cannot be changed. Furthermore, the record's authenticity can be verified by the entire community using the blockchain instead of a single centralized authority. The sustained high level of robust security demonstrated by public crypto-currencies has shown to the world that this new wave of blockchain technologies can provide efficiencies and intangible technological benefits very similar to what the internet has done (Freund & Stanko, 2018).

Within the blockchain is a block that is the 'current' part of a blockchain, which records some or all of the recent transactions (Ozercan *et al.*, 2018). Once completed, a block goes into the blockchain as a permanent database. Each time a block gets completed, a new one is generated. There is a countless number of such blocks in the blockchain, connected to each other (like links in a chain) in proper linear, chronological order. 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. Blockchain transaction works broadly as follows:

i) **Transaction Initiation**: One party (the sender) creates a transaction and transmits it to the network.
ii) **Transaction Authentication:** The nodes (computers and users) of the peer network receive the message and authenticate its validity by decrypting the digital signature.

- **iii) Block Creation:** Pending transactions are put together in an updated version of the ledger, called a block, by one of the nodes in the network.
- **iv) Block Validation:** The validator nodes of the network receive the proposed block and work to validate it through an iterative process that requires consensus from the majority of the network.
- **v) Block Chaining:** If all transactions are validated, the new block is "chained" into the blockchain, and the new current state of the ledger is broadcast to the network.

Blockchain contains sequence of blocks, which holds a complete list of transaction records like conventional public ledger (Lee, 2015). The figure below demonstrates an example of a blockchain. Each block points to the immediately previous block via a reference that is essentially a hash value of the previous block called parent block. It is worth noting that uncle blocks (children of the block's ancestors) hashes would also be stored in Ethereum blockchain (Buterin, 2014). The first block of a blockchain is called genesis block which has no parent block.

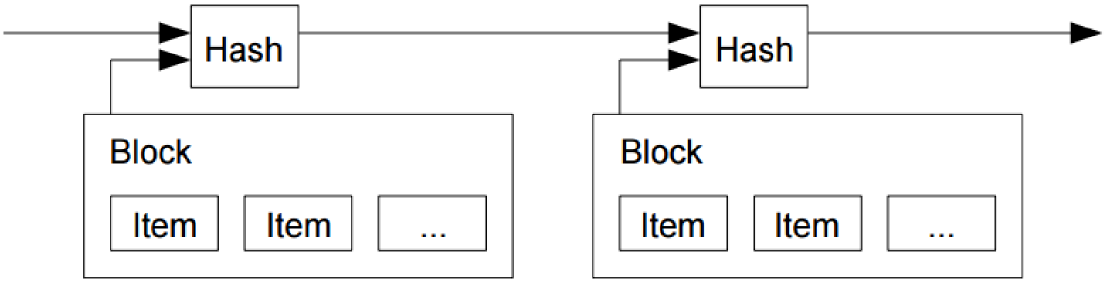


Figure 4 an example of blockchain which consists of a continuous sequence of blocks (Cachin, 2016)

The block body is composed of a transaction counter and transactions. The maximum number of transactions that a block can contain depends on the block size and the size of each transaction. Blockchain uses an asymmetric cryptography mechanism to validate the authentication of transactions (Aitzhan & Svetinovic, 2018). A digital signature based on asymmetric cryptography is used in an untrustworthy environment.

To carry out blockchain transaction, a digital currency called cryptocurrency is involved. A cryptocurrency is a digital or virtual currency designed to work as a medium of exchange in a blockchain based transaction (Zheng *et al*, 2016). It uses cryptography to secure and verify transactions as well as to control the creation of new units of a particular cryptocurrency (Barber *et al*, 2012). Essentially, cryptocurrencies are limited entries in a database that no one can change unless specific conditions are fulfilled. Presently accepted digital currencies include Litecoin, Ripple, Ethereum and Bitcoin (Atzori, 2015). In this research, Ethereum ether is used for mining and digital Ethereum ETokens for real estate transactions.

All transactions carried out in blockchain infrastructure is recorded in a public distributed ledger. A distributed ledger is a type of database that is shared, replicated, and synchronized among the members of a network (Cohen *et al*, 2017). The distributed ledger records the transactions, such as the exchange of assets or data, among the participants in the network. Participants in the network govern and agree by consensus on the updates to the records in the ledger (Robles *et al*, 2018). No central, third-party mediator, such as a financial institution or clearinghouse, is involved. Every record in the distributed ledger has a timestamp and unique cryptographic signature, thus making the ledger an auditable history of all transactions in the network. In short, blockchain can be described as a network of computers, each having an identical copy of the database (distributed) and changing its state (records) by common agreement based on pure mathematics, with no need for any central server or agent to entrust.

Before a blockchain transaction is recorded to a distributed ledger, a transactions validation using consensus is performed. Consensus is the mechanism by which the nodes agree whether a transaction and block are valid or not. As mentioned above, transactions need to be validated by the nodes. Nodes are parties with the ability to vote on the validity of a transaction (ACM Digital Library, 2006).This often requires processes to agree on some data value that is needed during computation. How to reach a consensus in a distributed environment is a challenge. Blockchains are inherently decentralized systems which consist of different actors who act depending on their incentives and on the information that is available to them (Wright & Filippi, 2015). Whenever a new transaction gets broadcasted to the network, nodes have the option to include that transaction to the copy of their ledger or to ignore it. When the majority of the actors which comprise the network decide on a single acceptable state, a ‘consensus’ is achieved. For cryptocurrency exchanges, smart contracts and distributed ledger based on blockchain to be led effectively, they should be affirmed by the blockchain. These affirmations depend on what is alluded to as consensus mechanisms. These mechanisms empower the system to continue working regardless of whether some of its members are coming up short. Below presents several common methods to reach a consensus in the blockchain distributed network.

i) Proof of Work

This is a protocol that has the main goal of preventing cyber-attacks such as a distributed denial-of-service attack (DDoS) which has the purpose of exhausting the resources of a computer system by sending multiple fake requests (Tahir et al, 2018). Proof of work is a requirement to define an expensive computer calculation, also called mining that needs to be performed in order to create a new group of trustless transactions on a blockchain distributed ledger. The mining process verifies the legitimacy of a transaction, or avoiding the double-spending and also to create new digital currency by rewarding miners for performing the previous task. Anytime a transaction using POW algorithm is set, Transactions are bundled together into a block and miners verify that transactions within each block are legitimate (Bergstra & Burgess, 2018). To do so, miners should solve a mathematical puzzle known as proof-of-work problem and a reward is given to the first miner who solves each blocks problem (Miraz, & Donald, 2018). The verified transactions are stored in the public blockchain as a new block creating chain of blocks (Blockchain). Although this algorithm reduces fraud, it is power intensive thus inefficient for large production networks.

ii) POS (Proof of Stake)

Proof of stake is an energy-saving alternative to PoW. Miners in PoS have to prove the ownership of the amount of currency. It is believed that people with more currencies would be less likely to attack the network (Caubet, 2018). The selection based on the account balance is quite unfair because the single richest person is bound to be dominant in the network. As a result, many solutions are proposed with the combination of the stake size to decide which one to forge the next block.

iii) Kraft

Zheng (2016) proposed this as a new consensus method to ensure that a block is generated in a relatively stable speed. It is known that high blocks generation rate compromise network's security. So the GHOST (Greedy Heaviest-Observed Sub-Tree) chain selection rule is proposed to solve this problem. Instead of the longest branch scheme, GHOST weights the branches and miners could choose the better one to follow.

iv) Peer-To-Peer Consensus Algorithm

Chepurnoy (2016) presented this consensus algorithm for peer-to-peer blockchain systems where anyone who provides non-interactive proofs of retrieving the past state snapshots based on agreed to generate the block. In such a protocol, miners only have to store old block headers instead of full blocks.

There are mainly three types of Blockchains that have emerged after Bitcoin introduced Blockchain to the world: public blockchain, private blockchain and consortium blockchain (Guo & Liang, 2016).

i) **Public blockchain**- all records in this blockchain are visible to the public and everyone could take part in the consensus process. Here no one is in charge and anyone can participate in reading/writing/auditing the blockchain (Sato & Matsuo, 2017). These types of blockchain are open and transparent hence anyone can review anything at a given point of time on a public blockchain. For example Bitcoin, Litecoin and ethereum (Gervais *et al*, 2016).

ii) **Private blockchain**- This is a private property owned by an individual or an organization. Only those nodes that come from one specific organization would be allowed to join the consensus process (Rouhani & Deters, 2017). Unlike public blockchain, this chain has an in charge who looks after important things such as selectively giving access to read or vice versa. Here the consensus is achieved on the whims of the central in-charge who can give mining rights to anyone or not give at all. This makes it centralized again where various rights are exercised and vested in a central trusted party but yet it is cryptographically secured from the company's point of view and more cost-effective for them. For Example Bank-chain.

iii) **Consortium blockchain**- This blockchain tries to remove the sole autonomy which gets vested in just one entity by using private blockchains. It ensures that only a group of pre-selected nodes would participate in the consensus process of a consortium blockchain. Such that instead of one in charge, there would be more than one in charge. Fundamentally, a group of companies or representative individuals come together and make decisions for the best benefit of the entire network (Li *et al*, 2018). Such groups are named consortiums or a federation.

Table 1 Comparisons among public blockchain, consortium blockchain and private blockchain (Lin & Liao, 2017)

Feature	**Private chain**	**Federation chain**	**Public chain**
Efficiency	High	High	Low
Consensus determination	One organization	A selected set of nodes	All miners
Reading permissions	Could be public or restricted	Could be public or restricted	Public
Immutability	Could be tampered	Could be tampered	Nearly impossible to tamper
Centralization	Yes	Partial	No
Consensus process	Permissioned	Permissioned	Permission less

The above discussed blockchain types exhibit the following common features;

a) **Decentralization-** In most conventional centralized transaction systems, the transaction requires to be authorized through the central trusted parties (such as the central bank) inevitably causing the cost and the performance hold-ups at the central servers. Inversely, a transaction in the blockchain network can be conducted between any two peers (P2P) without the authentication by the central agency (Tosh, 2017). Therefore, blockchain technology is effective tool for decentralizing the web. And it does possess the power to bring massive changes in the industries (Swan, 2015).

b) **Immutability**-Since each of the transactions spreading across the network needs to be confirmed and recorded in blocks distributed in the whole network, it is nearly impossible to tamper. Additionally, each broadcasted block would be validated by other nodes and transactions would be checked. So any falsification could be detected easily.

c) **Anonymity**-Each user can interact with the blockchain network with a generated address. Further, a user could generate many addresses to avoid identity exposure. There is no longer any central party keeping users' private information. This mechanism preserves a certain amount of privacy on the transactions included in the blockchain. Note that blockchain cannot guarantee the perfect privacy preservation due to the intrinsic constraint that are yet to be discussed.

d) **Auditability-** Since each of the transactions on the blockchain is validated and recorded with a timestamp, users can easily verify and trace the previous records through accessing any node in the distributed network (Zheng et al, 2016). This improves the traceability and the transparency of the data stored in the blockchain

2.3.2 Smart Contracts

Smart contracts are self-executing programs with terms of the agreement between two or more parties being directly written into lines of code that runs on the blockchain network (Atzei, 2017). The code and the agreements contained therein exist across a distributed, decentralized blockchain network (Luu *et al*, 2016). Kosba and others (2016), argues that contracts permit trusted transactions and agreements to be carried out among different, anonymous parties without the need for a central trusted party, legal system, or external enforcement mechanism. They render transactions traceable, transparent, and irreversible. Smart contracts help to exchange currency, possessions, stocks, lease of assets or whatever of value in a clear, conflict-free manner while avoiding the services of a middleman. As shown in the following figures, the difference between traditional contracts and the present smart contract can be seen. The traditional middlemen is replaced by auto-executable contract code.

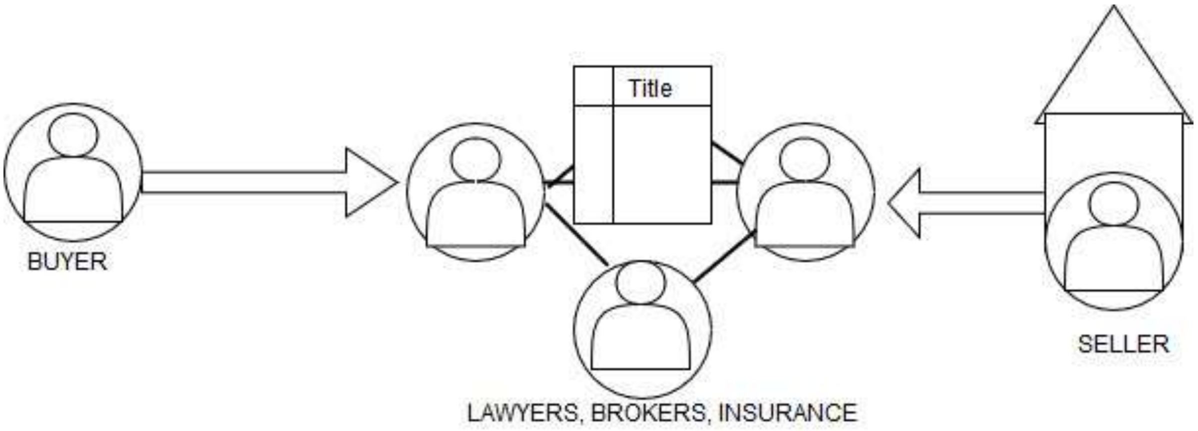


Figure 5 A Simple visualization of traditional contract having intermediaries (CB Insights, 2017)

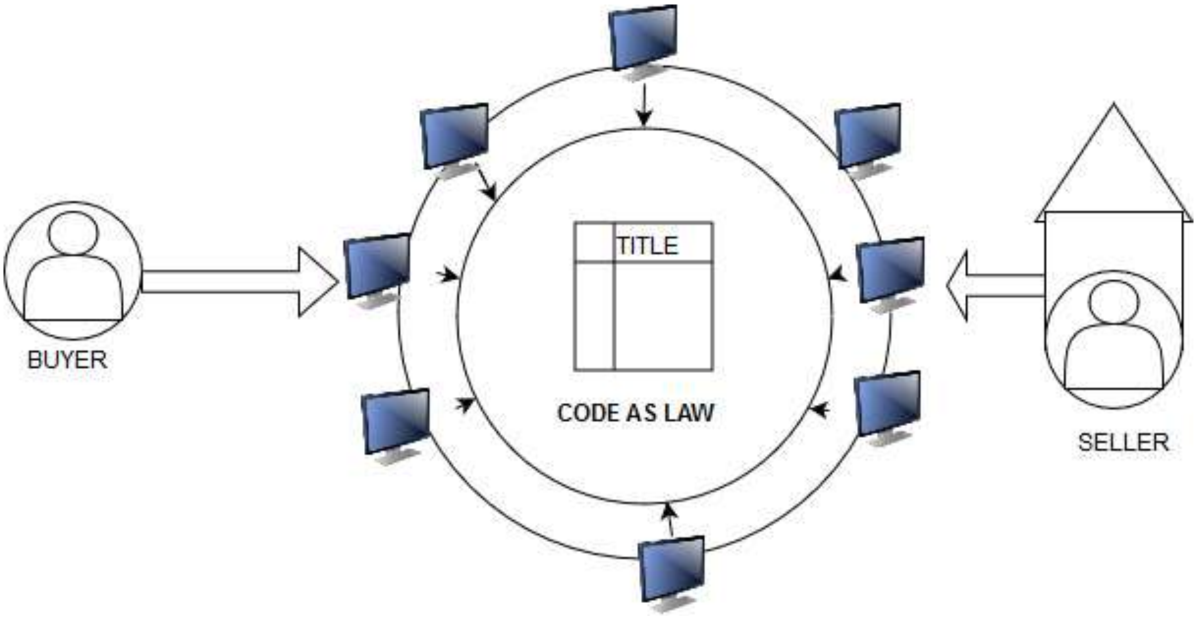


Figure 6 an example of working smart contract (CB Insights, 2017)

i) Ethereum Based Solidity Smart Contract

Ethereum smart contract used in this study is an example of a blockchain based smart contract. Ethereum is an open software platform based on blockchain technology that enables developers to build and deploy decentralized applications (Atzei *et al*, 2017). Ethereum blockchain miners work to earn Ether, a type of crypto token that fuels the network (Zhang *et al*, 2016). Ether is also used by application developers to pay for transaction fees and services on the Ethereum network (Bartoletti & Pompianu, 2017). According to Khan and others (2018), Ethereum Based Solidity smart contract is an autonomous solidity code that resides on the Ethereum blockchain. The code is implemented using solidity programming language. The currency of the system is ether which is used to pay for the transaction fees and the computation involved when the smart contracts are run (Pãnescu & Manta, 2018). The units of ether conversion are in the table below:

Table 2 Ether Unit Conversion (Pãnescu & Manta, 2018)

ether	**Other units**
1	288.31 USD
1	wei
1012	szabo
1015	finney
018	ether

Ethereum network supports two types of accounts: externally owned accounts and contract accounts. Externally owned accounts are owned by individuals and are controlled by their associated private key, whereas contract accounts are controlled by their contract code. Both types of account are referred to by their 20-byte address and have an ether balance associated with them (Vitalik, 2017). The accounts also store a nonce, to prevent double spending, as well as a storage field (Surcouf *et al*, 2018). Externally owned accounts are able to send transactions whereas contract accounts can only send messages. Only contract accounts can have corresponding code which is executed upon receiving a transaction or message (Cronie & Nolan, 2018).

2.4 Research Gap

From the literature, the researcher has noted that there is a gap in the existing models explored in section 2.1. The discussion below seeks to elaborate on these gaps;

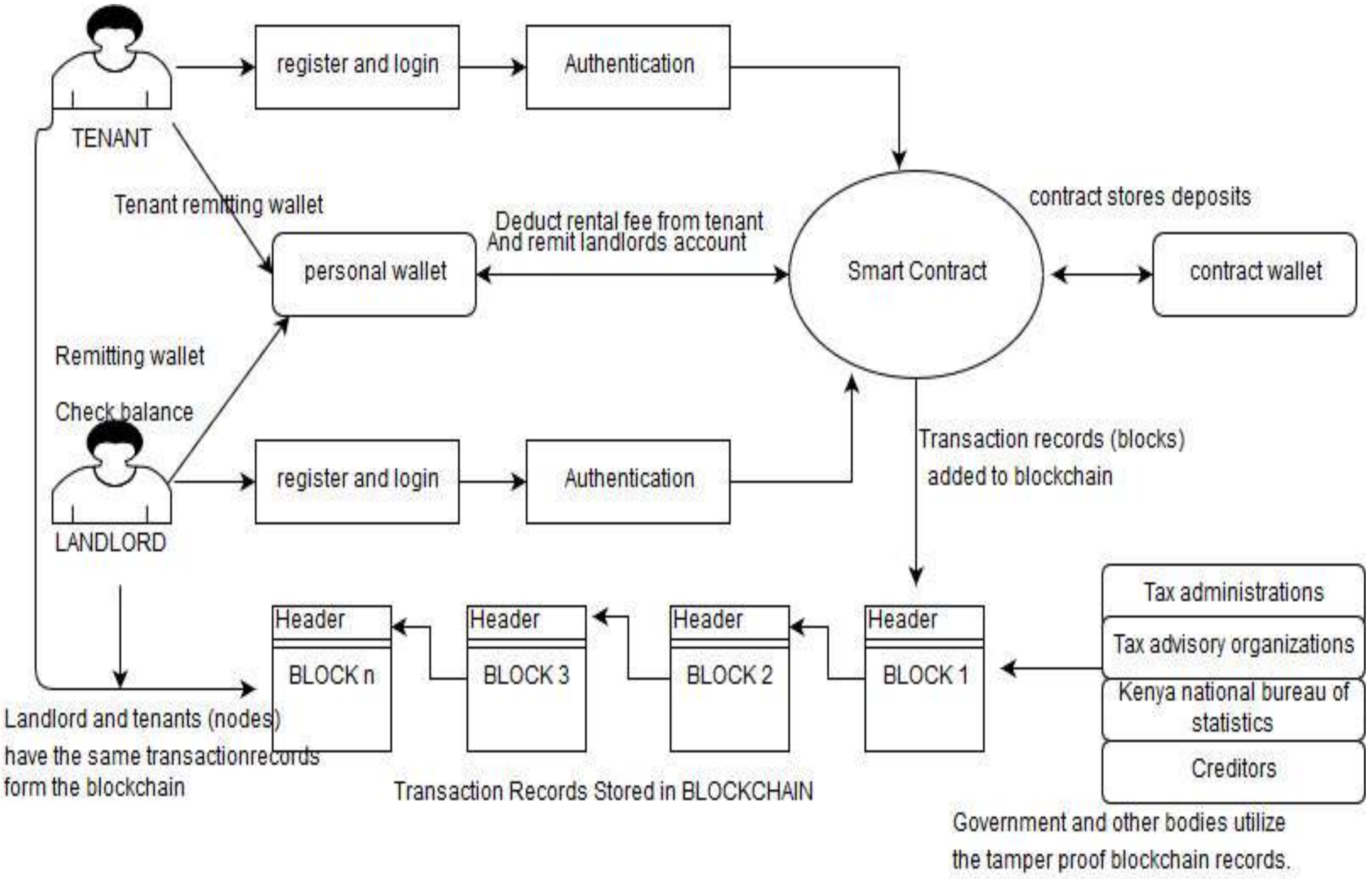
E-Resident in Kenya is a web-based platform that focuses only on creating convenience to tenants who will not have to deal with agents or queue at the bank to make rental payments. This platform aside from creating convenience, it does not address all of the above mentioned real estate property management Conundrum. The Bit-property smart contract platform aside from being decentralized smart contract platform that aims at giving users a seamless, quick, and low-cost way to own and trade real estate revenue streams, it does not address issues surrounding renting a rental property.

On the other hand, SPENN is a blockchain mobile-based platform for banking application that only enables users to make a range of financial transactions including rental fees payment. This model tries to removes the need for cash by digitalizing national currencies as an effort to reduce corruption within any chain of distribution; however, it does not entirely troubleshoot the real estate issues stated in section 4.1.1of this research. For the case of DEED'S platform, it focuses on reducing challenges in finding real estate agent and the decision on the best commission. Therefore it does not meet the design requirements for a blockchain based model and the overall objective of this study.

Although the existing models examined seek to improve real estate management, not much on reducing frictions in Claiming of deposit, eliminating intermediaries, Enforcement of tenants' rights and obligations, reducing Conflict between tenants and landlord over late payment of rent or reducing cases of Unexplained Rent Increments which are the core issues highlighted in the SDGs and the big four agenda. The management approach in the real estate industry specifically housing market need to consider Ethereum Based Solidity Smart Contract Model for Housing Marketbest practices. This study therefore, aims to bridge the gap by designing Smart Contract Model for Housing Marketwith a more comprehensive approach that reduces the above-mentioned issues.

2.5 Conceptual Framework of the Study

The diagram below shows the conceptual framework showing interaction between involved parties.



TENANT
register and login
Authentication
Tenant remitting wallet
personal wallet
Deduct rental fee from tenant
And remit landlords account
Smart Contract
contract stores deposits
contract wallet
Remitting wallet
Check balance
LANDLORD
register and login
Authentication
Transaction records (blocks)
added to blockchain
Header
BLOCK n
Header
BLOCK 3
Header
BLOCK 2
Header
BLOCK 1
Tax administrations
Tax advisory organizations
Kenya national bureau of statistics
Creditors
Landlord and tenants (nodes)
have the same transactionrecords
form the blockchain
Transaction Records Stored in BLOCKCHAIN
Government and other bodies utilize
the tamper proof blockchain records.

Figure 7 conceptual framework

Ethereum Based Solidity Smart Contract Model for Housing Market is therefore designed as a solution to the challenges facing the housing market industry. The model will have the following modules; A User Registration module for both the tenant and the landlord, User login and authentication module that will be regulating access to only authorized users, Smart contract to automate transaction and create immutable transaction blocks of records, personal wallet to store individual ethers and contract walled to store deposits, and a chain of transaction blocks. The model must be very robust as it will be holding cash, must not show any authentication details, due to occupancies being private matters, and must have an intuitive and simple user interface for non-technical/Amateurish users. The tenant will be paying the deposit in ether to the smart contract which is storing the logic for the model, therefore not subject to long bank processing times and high fees in the case of international payments. The funds are locked in until the end of the tenancy when the landlord can decide the appropriate deduction amount. An arbitrator makes the final decision for the withdrawal split in the case of a dispute. In addition to the deposit being paid in ether, using smart contracts facilitates the rent installments to also be paid to the landlord in ether. The smart contract enforces logic on these payments, such as preventing the rent from being overpaid or transferred after the tenancy is complete. This automatic payment will involve the user allowing program access to their account in order to transfer rent on their behalf.

2.6 Prototype Implementation Overview

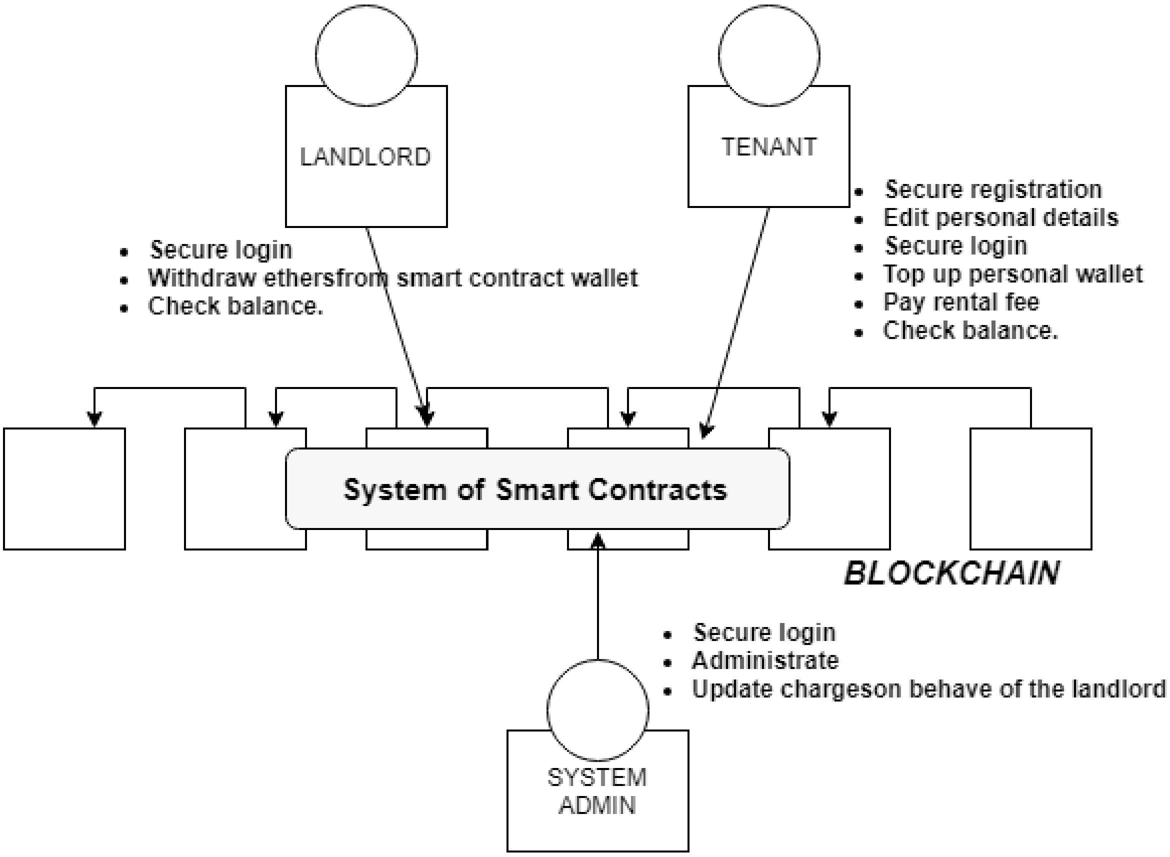


Figure 8 Prototype implementation overview with different users and their interactions with the blockchain and system of smart contracts which exist on the blockchain Source: (Researcher, 2019)

To sum up, this chapter explored existing literature on the existing models used in real estate property management. It furthermore explained the theories underpinning blockchain as a public ledger of executed transactions and finally presented a blockchain based smart contract design for real estate property management.

CHAPTER THREE

RESEARCH DESIGN AND METHODOLOGY

3.0 Introduction

This chapter presents the research methodology adopted in this study. According to Kumar, research methodology is a systematic way to solve a problem. Its aim is to give the work plan of research. To achieve the study objectives a proof of concept (POC) methodology was adopted to prove the concept of resolving the weaknesses of the existing models for real estate property management using a blockchain based smart contract model. The proof of concept was carried out in five steps namely planning, designing, implementation, operation and improvement (PDIOI). The POC focused on coming up with a Solidity Smart Contract Based on Ethereum Network.

3.1 Proof of Concept

Proof of concept (PoC) is a realization of a certain method or idea in order to demonstrate its feasibility or a demonstration in principle with the aim of verifying that some concept or theory has practical potential (Cedarbaum, 2018). The use of proof of concept approach in this thesis gave an opportunity to demonstrate and verify the practical potential of smart contract in real estate management. The model was meant to be cheaper and less complex than the system it models as proposed in (Gordon, 2014), therefore this allowed the researcher to better understand and utilize the model to perform experiments that would not be better carried out in a real environment because of cost implications (Rezaee, 2018).

PoC is characterized with the aim of avoiding spending a lot of money (resources) on something that is not viable market-wise or technology-wise. One can get tangible proof that his/her idea is worthwhile and that will be difficult to argue against other project stakeholders. This allows with high probability to determine the value of the project even before the beginning of the development process.

3.1.1 The Proof of Concept Procedure Stages

A Proof of Concept (POC) can either be a Prototype without code or an MVP (Minimum Viable Product) with a minimal feature set with a week zero. As pointed in planning in section 3.2 this research used proof-of-concept plan to address how the proposed model will support organizational goals, objectives and other upcoming business requirements. This involves clearly defining criteria for success, documentation for how the proof of concept will be carried out, an evaluation component; and a proposal for how to move forward should the POC prove to be successful. This study by use of focus group defined user requirements and evaluation criteria via user stories as shown in chapter 4.

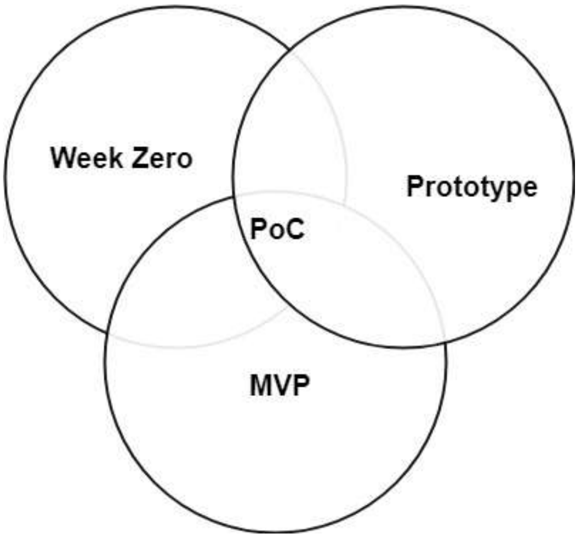


Figure 9 the proof of concept procedure stages (Oliveira *et al*, 2015)

During Week Zero, this research did a literature review to prove the need for Blockchain based solutions for real estate management as presented in chapter 2 and part of background and justification of this research in chapter 1. It showed how blockchain smart contracts would improve manageability or efficiency. Thereafter, it demonstrated how to map the goals/problems to a possible solution as shown in chapter four. The prototype was purposefully extremely rudimentary as the researcher fully planned on not reusing any of the prototypes in this state in later stages. In general this stage (week Zero) gave a clear theoretical postulate ready to be tested.

In this research again, the prototype is needed to affirm and demonstrate the feasibility and practical potential of the proposed idea. This informed and enabled the researcher and other decision-makers to determine how best to develop the product when it moves into full production for a final, market-ready item as pointed in (Segura, 2018).

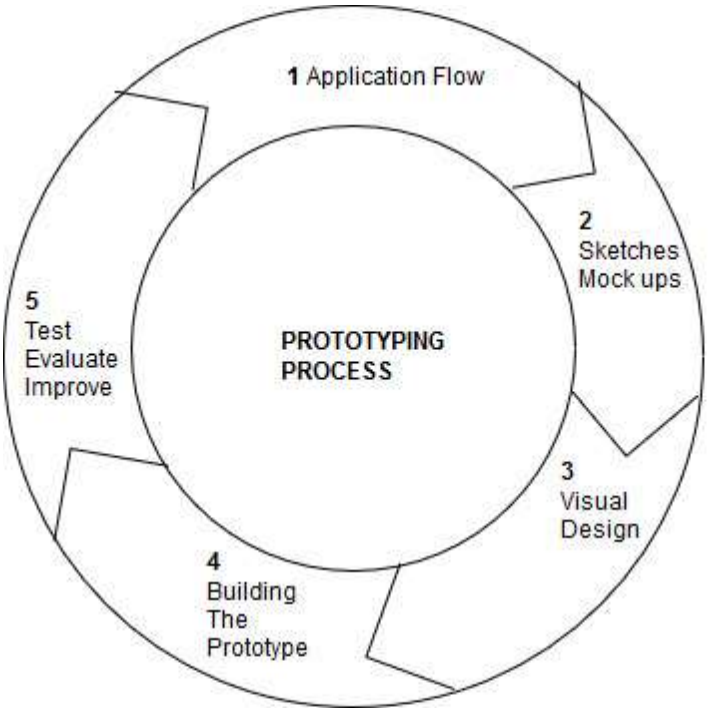


Figure 10 Prototyping process (inquvent, 2017)

Similarly, a minimum viable product (MVP) came after successful prototyping of the proof of concept (Schmidts, 2018). In this stage, the idea was to create a product with a minimum set of features that were viable enough to test the research fundamental assumptions. The MVP was used to prove the proposed models postulate with minimal resources. To prove that the model designed is an achievable solution, a PoC has been built. Chapter 4 contains a description of how this PoC has been implemented and a discussion on the design considerations. Furthermore, descriptions of the use cases that are defined for the PoC are given. The PoC has been created and tested. The results of these model and tests are used to validate the model and PoC. This process is described in Chapter 4.

3.2 PDIOI Approach

The Approach chosen for this research was Planning, Design, Implementation, Operation, and Improvement. PDIOI is Huawei methodology for Implementing Enterprise Engineering Project (IEEP). This methodology was ultimate as it aided in improving agility, Speeding processes, and Reducing and addressing complexities during the PoC artifact implementation. As shown in this figure 11 below, PDIOI lifecycle involves five closely related phases.

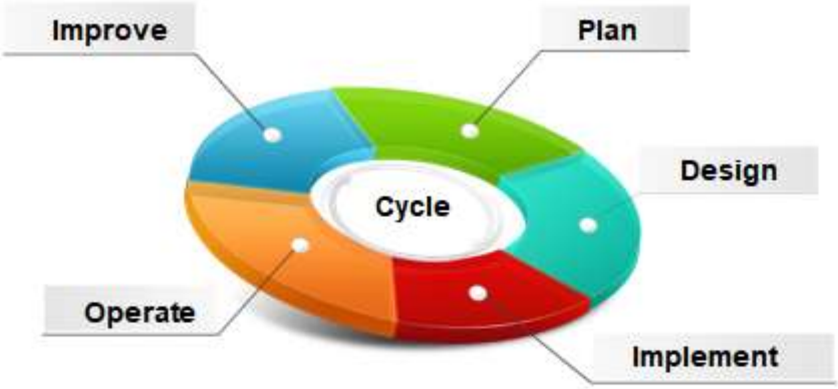


Figure 11 Plan, Design, Implement, Operate, and Improve source (Huawei, 2019)

i) Planning

Planning is the initial phase of the proposed model. The model's background, requirements and objectives analysis are done as well as determining the technological roadmap as presented from section 4.2. It reviewed the proposed model from a macro view and aims to set a framework for the entire research and determines the general orientation, which directly influences model achievements. It also determines the model scope, which is the prerequisite for budgets and resources allocation. It defines responsibilities for matching the research and relevant systems respectively. Guiding principles for the research design are clarified providing guidance and basis for future work. In general, this phase grasps the general background of the research, ensuring a sound exterior environment for smooth development.

ii) Design

In the design phase, designs based on the model requirements and guidelines specified in the planning phase are implemented through technological methods. Design phase follows the modular design principle and hierarchical architecture is used in each module as shown in section 3.2.1. It determines device selection, technological roadmap, model functions, and performance specifications. The starting-up activities for the model and technology selection were done in this stage as shown in section 4.2.3.

iii) Implementation

During the implementation stage, the model was built and the additional components were incorporated according to the design specifications, Supporting infrastructure and governance as shown in section 4.2 and 4-3.

iv) Operation

Throughout the operate phase, the researcher proactively monitored the working and vital signs of the model to improve service quality; reduce disruptions; mitigate outages; and maintain high availability, reliability, and security. It provided an efficient framework and operational tools to respond to problems, avoid costly downtime and interruption. It also allowed upgrades, moves, additions, and changes while effectively reducing model operation defects.

v) Improve

The purpose of the model optimization phase was to improve model performances, security, and user experience. This enabled the developed model to meet service requirements. Optimization includes hardware optimization, software optimization, expansion, and technology update.

3.2.1 Smart Contracts Design

Every non-trivial Dapp will require more than one contract to work well. According to Beck (2016), there is no way to write a secure and scalable smart contract back-end without distributing the data and logic over multiple. Therefore, the designed architecture of the model as system of smart contracts is based on the design principle of having different types of contracts to perform different classes of tasks. To classify the contracts, a model called "The Five Types Model" is used (Monax, 2017), although not all of the five models are actually used in this PoC. This model divides contracts into Database contracts, Controller contracts, Contract managing contracts, Application logic contracts and Utility contracts.

i) **Database contracts**

These Contracts are used only as data storage. The only logic they need is functions that allow other contracts to write to, update and get data, and some simple way of checking caller permissions (whatever those permissions may be).

ii) **Controller Contracts**

One step up in the layer of abstraction is a contract for controlling database contracts. These contracts operate on storage contracts. In a flexible system, both controllers and databases can be replaced by other, similar contracts that share the same public API (although this is not always needed). Controllers can be advanced, and could, for example, do batch reads/writes, or read from and write to multiple different databases instead of just one. They can also act on multiple database contracts.

iii) **Contract Managing Contracts (CMCs)**

These contracts are needed to control and manage the actions and existence of other contracts. Their main task is to keep track of all the contracts/components of the system, handle the communication between contracts and other components, and to make modular design easier. Keeping this functionality separate from normal business logic should be considered good practice, and has a number of positive effects on the system (as we will see later).

iv) **Application logic contracts (ALCs)**

Any contract that is implementing application-specific code tasks through controllers is an application logic contract. Generally speaking, if the contract utilizes controllers and other contracts to perform application-specific tasks it's an ALC.

v) **Utility contracts**

These types of contracts usually perform a specific task and can be called by other contracts without restrictions. It involves some small, generic functions that can be outsourced into utility contracts that are highly specialized. It could be a contract that hashes strings using some algorithm, provide random numbers, or other things. They normally don't need a lot of storage, and often have few or no dependencies.

Based on the above-proposed monax model and the PoC functional requirement, the model design proposes the following contracts.

Table 3 the contracts in the PoC of this study are the following Source: (Researcher, 2019)

EToken	A contract that creates tokens that are used in the PoC. The created Etokens are used for gas prices and rental fee.
Interface	A composite contract that manages the RPC and links all other contracts.
Migrations	Migrations are JavaScript files that help to deploy contracts to the Ethereum network. These files are responsible for staging deployment tasks, and they're written under the assumption that the deployment needs will change over time.
TenantInfoDb	A database contract to store information about tenants. The information stored for each tenant is personal information and Addresses. A Tenant-struct is created for each Tenant to contain the corresponding data as well as a next and a prev-attribute so that a doubly-linked list can be used to iterate over them.
CMC	The contract-managing-contract is simply named CMC and contains a collection of the different contracts. All other contracts must be connected to the CMC or inherit from the class CmcEnabled.
CmcEnabled	Base class for contracts that are used in a CMC system
Tenant	Application logic contract for handling requests from Tenants such as retrieving making a payment, changing consent-level for a certain house description-Admin tuple, etc
Landlord	Application logic contract for handling requests from Landlords. These include adding new room features, payment or confirming that a certain feature has been posted
ContractProvider	Interface for getting contracts from Cmc.

3.2.2 Model Design Constructs

The real estate model was subdivided to modules. The sections were organized according to functions to be achieved. The model functional sections are as listed below;

i) **The Smart Contract -** To write an Ethereum smart contract Solidity programming language was used. It's a general-purpose programming language that uses a class (*contract*) and methods that define it. Solidity main purpose is to send and receive digital tokens as well as storing states.

ii) **Ethereum Wallets-** To fully demonstrate the research objectives, three types of wallets on Ethereum were created: for the tenant, for the landlord, and for the contract respectively.

iii) **Dapp** - The front end users are using Dapp to connect with the blockchain via the Smart Contract (Front End → Smart Contract → Blockchain). Truffle development framework for Ethereum was used.

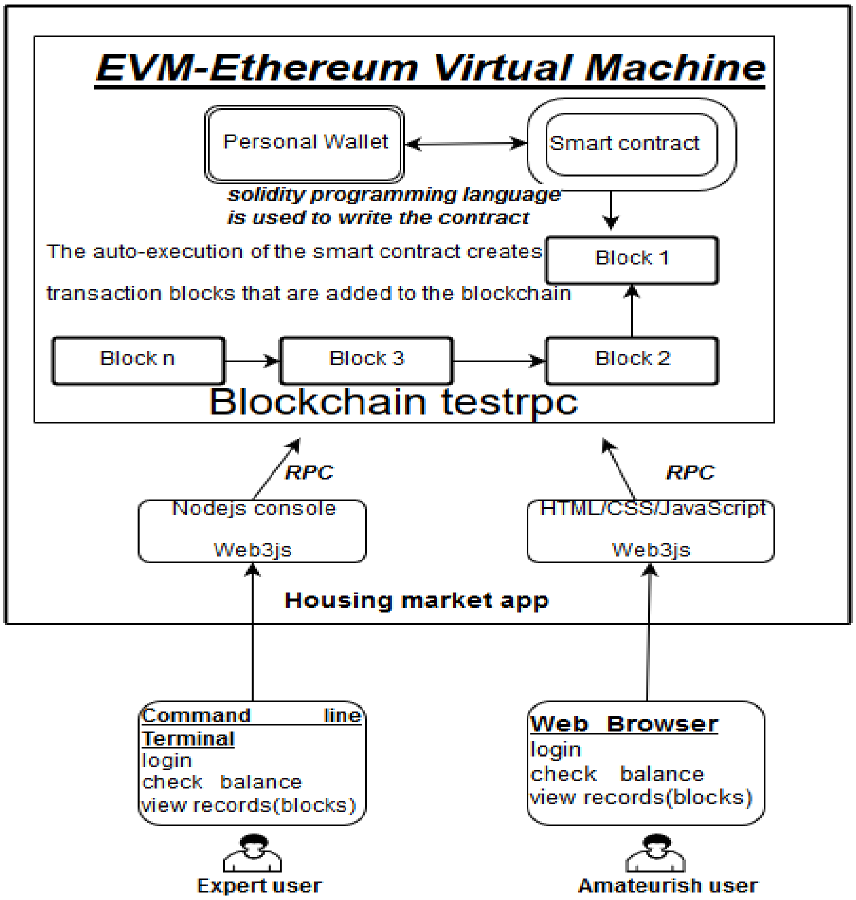


Figure 12 Model Development

3.2.3 Overview of Smart Contracts Model Design

To manage the model through its entire life cycle, from setting up a structure to drawing up detailed plans, to executing and completing the project the following overview of the system of smart contracts for the Real Estate Management PoC figure was of help. It describes what the system is and what it does. Although, based on visualization technique by (Brown, 2016) not all smart contracts are included in the PoC for sake of clarity and relevance.

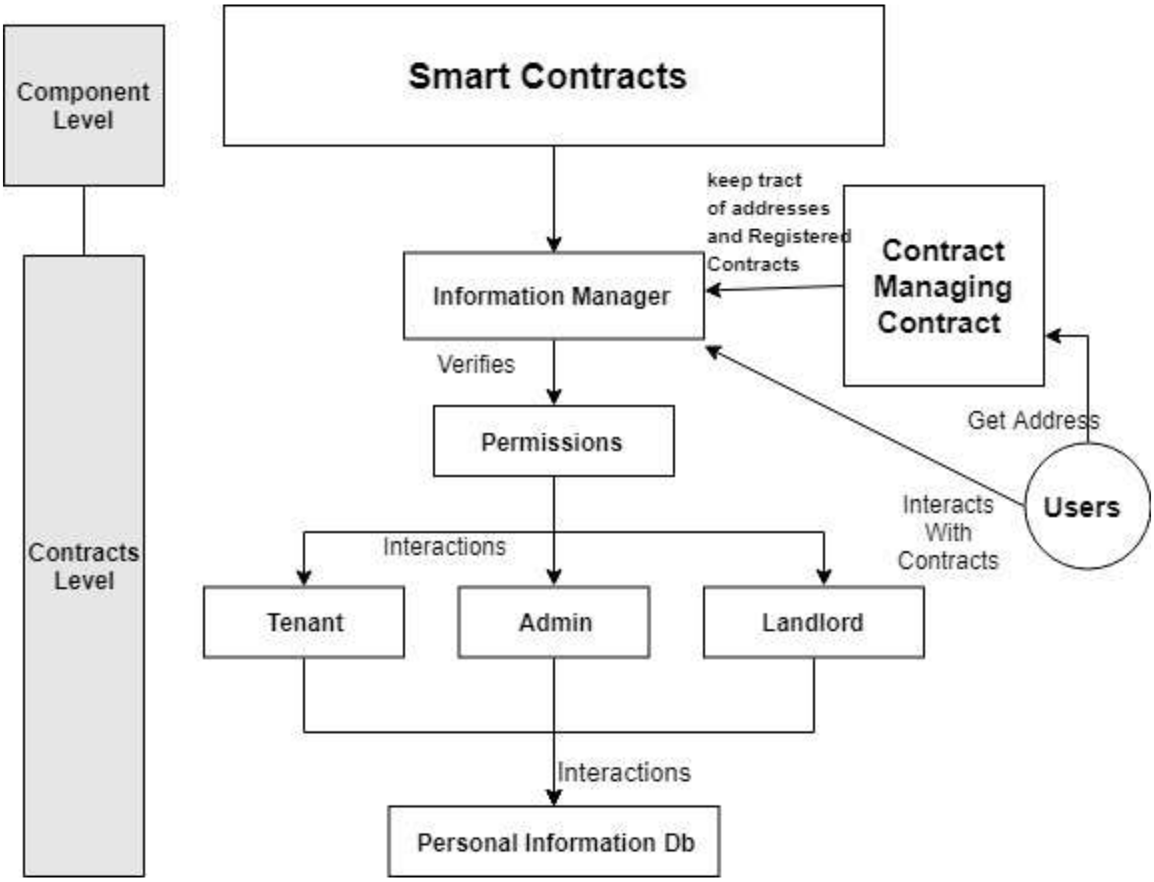


Figure 13 Overview of smart contract design Source: (Researcher, 2019)

3.3 Model Evaluation

As pointed by Davidson (2018), the model evaluation approach was based on the idea that "the purposes, goals or targets" of a research are determined at the start and the evaluation process will be used to establish whether the goals have actually been achieved and, if not, why not questions are used to amend missed functionalities. The developed model was evaluated using Goal-based evaluation technical whose main objective is to establish whether the prototype meets the set technical objectives (Fahmideh *et al*, 2018). The model evaluation details are presented in chapter 4.

3.3.1 Description of Evaluation Criteria

The artifact (model) was evaluated according to the criteria: "functionality, completeness, consistency, accuracy, performance, reliability, usability, fit with the organization, and other relevant quality attributes" (Hevner *et al*, 2004). However, given the novelty of blockchain technology, the scope of the thesis and the extension of the PoC, the artifact was evaluated according to all criteria. Abdu (2011) says a descriptive evaluation can be used only in the case where the technology is especially innovative and other methods may not be feasible. The Blockchain based Model for real estate fits those criteria and was therefore evaluated using two methods from the descriptive evaluation theory.

i) **Informed Argument** use information from the knowledge base for example other research to build a convincing argument for the artifact's utility.
ii) **Scenarios** a detailed scenario around the artifact was constructed to demonstrate its utility.

The verification criterion was based on specification and description stated in section 4.1.1 and table therein and validated based on model functions defined in section 4.2.1. This was pointed out in the section 4.4.1-2 "Fulfillment of evaluation criteria".

3.3.2: Selection Criteria for Model Testing

In selecting the agencies for model testing, purposive sampling was adopted to ensure that only those agencies that use the IT service were included in the study to test the adequacy of the model. Purposive sampling (also known as judgment, selective or subjective sampling) is a sampling technique in which researcher relies on his or her own judgment when choosing members of population to participate in the study (Wiklund & Jansson, 2019). This study used purposive sampling to ensure that all those agencies that have adopted IT-based service are used as the sampling frame of the study. To realize this, a list of all licensed property managers was obtained from the department of license and revenue of county government of Nakuru.

3.4 Ethical Consideration

During the requirement gathering and model testing, several ethical considerations adhered. The researcher obtained information that further the purpose of this study towards coming up with a model to manage real estate. The ethical considerations were in line with regulating authorities' requirements. This was done by having an introductory letter from the Institute of postgraduate and research of Kabarak University and also, a permit obtained from the National Commission for Science Innovation and Technology (NACOSTI). The information that was obtained from the respondents was and will be treated with strict confidentiality.

3.5 Conclusion

The above section described the research approach and the methodology that was used in validating the viability of the proposed idea and focused on achieving the objectives of this study and adequately answering the research questions. It described PDIOI (Planning, Design, Implementation, Operation, and Improvement) and POC (proof of concept) approach.

CHAPTER FOUR

DATA ANALYSIS, PRESENTATION AND DISCUSSION

4.0 Introduction

This chapter presents the results of the study as per the research objectives presented in section 1.3. A report of the existing real estate property management models, a prototype of a blockchain based smart contract implementation model to assist real estate crews, and verification report on the functionality of the real estate blockchain smart contract model prototype is presented.

4.1 Existing Models Used in Managing Real Estate Property

This section presents results of objective one of the study which involved exploration of the weakness of existing models used in real estate property management. The Existing real estate management models reviewed in section 2.1 of this study included E-Resident, SPENN, Deed-coin and The Bit-property. E-resident Platform was designed by KCB Bank Kenya Limited in partnership with E-Resident Limited and has been publicized to offer a host of benefits such as security, convenience and cut collection cost for landlords and agents. The platform also makes tracking of payments easier since the money is sent from any of the money transfer platforms straight to the bank. This model also designed to ensure customer’s right to privacy and right to access information with lawful restrictions therein as provided for in the Constitution of Kenya. SPENN is a mobile-based platform for banking application that was designed to enable Rwandans to make a range of financial transactions. The application allows smartphone users to transact and pay for goods and services. Deed-coin platform allows searching for a real estate agent. Bit-property is a decentralized real estate platform powered by Ethereum smart contracts that aim at giving users a seamless, quick, and low-cost way to own and trade real estate revenue streams.

4.1.1 Weaknesses of the Existing Models

To appreciate the importance of technology product, there needs to be a reason for the technology to exist. Therefore, to justify the need for a new blockchain smart contract for real estate, the existing real-world alternative needs to exhibit inefficiencies in their use cases. The set criteria for the existing models evaluation include; transaction transparency and immutability, decentralization (distributed ledger), presence intermediaries, implementation of rights and obligations, potential to reduce overall rental fee, fractions in claiming deposits conflicts over late payment, possibility of unexplained rent increments and possibility to enforce lease agreement. The table below presents the key of assessment criteria:

Table 4 Key of Assessment Criteria

S/N	Assessment Criteria	Traceability
i)	Transparency	A.
ii)	Immutability	B.
iii)	Decentralization	C.
iv)	Presence Intermediaries	D.
v)	Implementation of Rights and Obligations	E.
vi)	Potential To Reduce Overall Rental Fee	F.
vii)	Fractions in Claiming Deposits	G.
viii)	Conflicts Over Late Payment	H.
ix)	Possibility of Unexplained Rent Increments	I.
x)	Possibility To Enforce Lease Agreement	J.

Table 5 Existing Models Assessment

Model	A	B	C	D	E	F	G	H	I	J
E-Resident	✗	✗	✗	✓	✓	✗	✓	✓	✓	✓
Bit-property	✓	✓	✓	✗	✓	✗	✗	✗	✗	✓
SPENN	✓	✓	✓	✓	✗	✗	✓	✓	✓	✗
EED'S	✓	✓	✓	✗	✓	✓	✗	✓	✓	✗

The E-Resident platform in Kenya is a web-based platform that focuses only on creating convenience to tenants who will not have to deal with agents or queue at the bank to make rental payments. It does not allow transaction transparency, immutability or decentralization (distributed ledger). Its operation also encourages the need for trusted intermediaries such as the banks and agents thus increasing the rental fee. Furthermore, the model does not prevent the cases of unexplained rent increment or allow enforcement of lease agreement, rights and obligations. The Bit-property smart contract platform aside from being decentralized smart contract platform that aims at giving users a seamless, quick, and low-cost way to own and trade real estate revenue streams, it does not address issues surrounding renting a property.

SPENN is a blockchain mobile-based platform for banking application that enables users to make a range of financial transactions. This model does not address property management requirements such agreements, to authorize occupants' rights or obligations. DEED'S platform focuses on reducing challenges in finding real estate agent and the decision on the best commission. It does not deal with real estate property management.

4.2 Design of Blockchain Based Smart Contract Model for Real Estate

In this section, the design of the blockchain based smart contract model for real estate is presented. This addresses the attainment of research objective two of the study. First, the three different user archetypes are described along with user stories in order to provide functional specifications for the PoC and the model objectives. Further descriptions of the PoC such as quality attributes and how to set the system of smart contracts up with a blockchain are also given. Thereafter a schematic of the prototypical interactions on the blockchain is shown along with the interactions between the smart contracts. The suggested solution to the described problem uses the decentralized, trust-less and immutable properties of blockchain technology as well as permissions in the smart contracts. To be noted is, however, that no security or privacy liabilities outside of the blockchain have been resolved with this implementation. Some of the larger off-chain issues are mentioned in the model discussions.

4.2.1 Defining Model Functional Requirements

The implementation outlined in this thesis is limited to three archetypical users: Admin, Tenant, and Landlord. In order to describe the various functional requirements that users have on the application, user stories from focused group were written and are shown in the Table below. The different user types are thereafter defined more in detail. An effort was made to simplify the user stories and requirements to the bare minimum, while still keeping the PoC at a viable level of usability and security.

Table 6 User stories defining functional requirements and guiding the development of the model PoC Source: (Researcher, 2019)

As a . . .	I want to. . .	Traceability
admin	Identify myself in a cryptographically secure manner upon accessing Tenant information on the blockchain, so that no unauthorized entities can access it.	1.1
	Administrate (setup and maintaining accounts) the system.	1.2
Landlord	Be able to see records of my tenants so that I can know the number and compute expected pay and when to expect payment.	2.1
	Verify the tenant-account identity to know who has paid and who has not.	2.2
	Be able to see how much rent has been collected, so that I can plan for service, bills, and further investment.	2.3
	Set and Adjust the rent level to attract tenants to the property.	2.4
	Be able to Collect Rent from smart contract account each month and strictly enforcing late fees.	2.5
	Identify myself in a cryptographically secure manner upon accessing landlord information on the blockchain, so that no unauthorized entities can access it.	2.6
Tenant	Be able to create an account through registration.	3.1
	Identify myself in a cryptographically secure manner upon accessing my personal information on the blockchain.	3.2
	Be able to see and update my personal information upon registration.	3.3
	Be able to share information on my account with Landlord and Admin on the blockchain.	3.4
	Be able to pay my rental fee using any form of cryptocurrency. bills for gas, electricity, and telephone if this was agreed with the landlord	3.5
	Be able to check my account balance (ether wallet or token wallet).	3.6
	Be able to see my current payment status and expected date of the next payment.	3.7

Towards realizing the concept, the model was implemented using open source tools as shown in section 4.2.4. The model was developed with lite server, truffle framework, nodeJS, NPM, and web3 JSON RPC(s) in interactive functions. The above user stories can be summarized in the table below to give the model functional overview.

Table 7 Model Functional Overview Source: (Researcher, 2019)

The Model Functionality	**How to realize**
1. User registration	All users are required to register with the model prior to accessing any of the model functionality. This is to capture the user personal information and authentication details.
2. User authentication and security	The proposed model ensured that all users are registered before allowing them to access any of the system functions. The registered persons are required to log in using their account addresses and passwords.
3. User de-registration	Anytime a tenancy contract is terminated, the tenant details are removed from an active account to a repository.
4. Rental fee payment	The model allows rental payment by use of Ethereum. This PoC uses a created crypto tokens to allow unlimited testing.
5.Transaction records management	The model allows recording of the transaction on the blockchain
6. Report generation	The model enables users to generate reports regarding transactions.

4.2.2 Users Overview and Interactions with Blockchain

The participating individuals in the blockchain based model for real estate management PoC are composed of; Tenants, admin, Landlord, and hosting service providers (EVM, Github, and Metamask).

a) **Tenants** are assumed to be private persons or group that rents and occupies a house, an office, or the like, from another (landlord) for a period of time; lessee.

b) **The landlord** is assumed to be a person or group who owns a house, apartment, condominium or real estate which is rented or leased to an individual or business, who is called a tenant (also a lessee or renter).

c) **System administrator, or sysadmin,** is a person who is responsible for the upkeep, configuration, and reliable operation of computer systems; especially multi-user computers, such as servers. The system administrator seeks to ensure that the uptime, performance, resources, and security of the computers they manage meet the needs of the users, without exceeding a set budget when doing so.

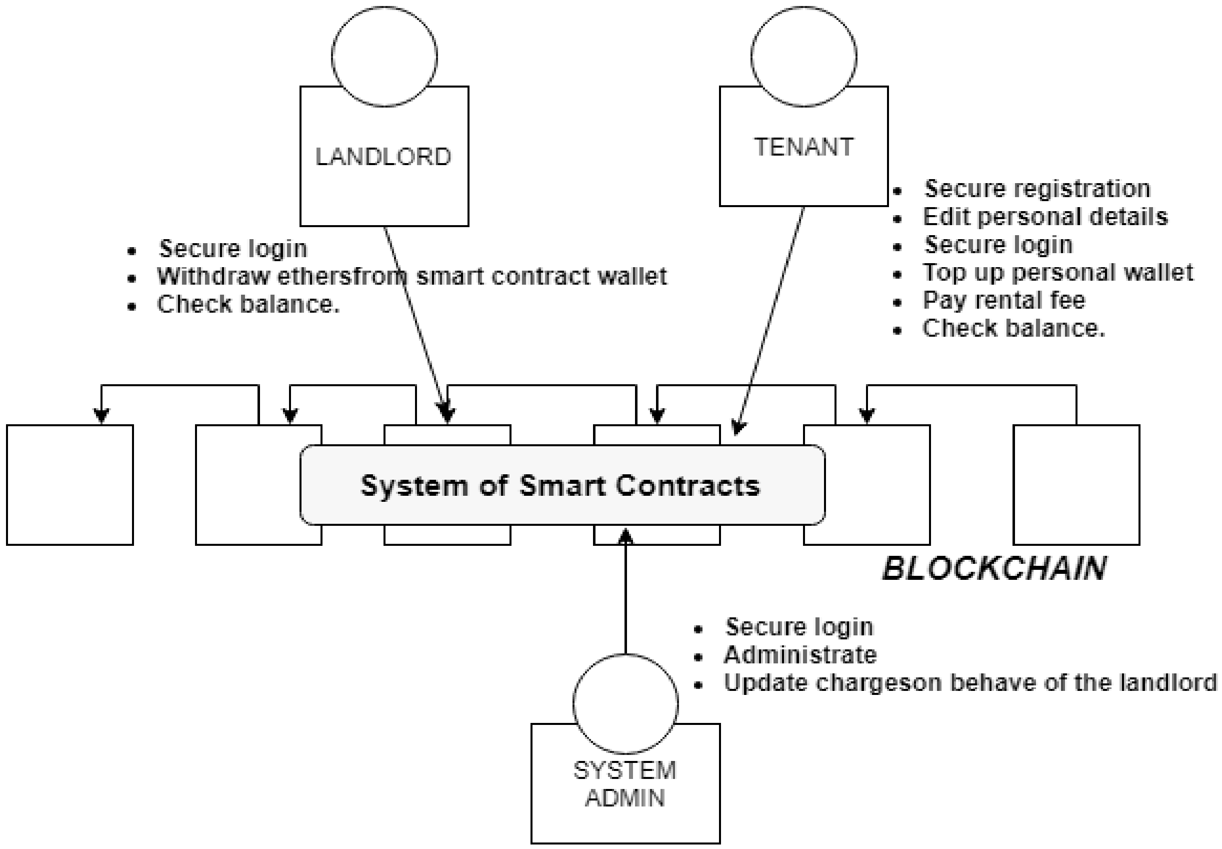


Figure 14 Overview of different users and their interactions with the blockchain and the model as a system of smart contracts

The requirements are described in Figure 14, where the different users are shown interacting with the blockchain, on which the smart contracts reside. In item next to them are the actions they need to be able to perform. There are some requirements that apply to the general system and not just to one user specifically. Some of them are described in part by the user stories, but for the sake of exhaustiveness and application to users not in the system, they are explicitly written below.

R1. It must be impossible for a non-registered person to join the model network.

R2. Only those permitted to should be allowed to connect to the network.

R3. There must be immutable traceability built into the model, where it is possible to see:

i) All rental fee transactions captured in the blockchain (in form of transaction Blocks).
ii) Account balance before and after the transaction.
iii) Old transactions.

Immutable traceability means that there must be a history of changes made to the rental fee, payment and that it must be made very difficult, if not impossible, to alter it post ex.

R4. Smart contracts must be exchangeable without the need to remove the entire model or change addresses to contracts with which humans interact.

The design of the final PoC was based on the requirements and user stories mentioned above. It would also be guided by the core objective of this research.

4.2.3 The Model Starting-Up Activities

To start bare truffle project the “truffle init” is ran from the command line in the project directory. Once this operation is completed, the project structure is created with the following items: Contracts Directory for Solidity contracts, migrations Directory for scriptable deployment files, Test Directory for test files for testing your application and contracts, and truffle.config.js a Truffle configuration file. The sequence of activities is as shown in the figure below where the validators are configured, the blockchain is started and transactions can begin.

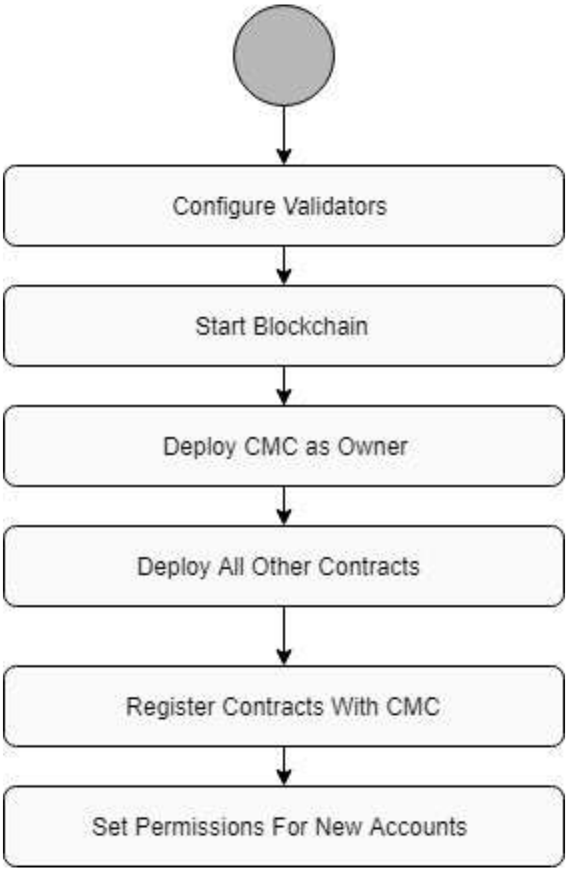

Configure Validators
Start Blockchain
Deploy CMC as Owner
Deploy All Other Contracts
Register Contracts With CMC
Set Permissions For New Accounts

Figure 15 Diagram of how the starting-up activities for the model work
Source: (Researcher, 2019)

4.2.4 Truffle Framework Installation

The PoC model isn’t complete with just the smart contracts. There is also a need for blockchain, as well as a structure for handling keys. Additionally, there needs to be a type of either distributed or centralized consensus established, on who is allowed to join the network as a Tenant or Landlord. The details of the consensus algorithms and the nature rationale for the selected blockchain are explained in Section 2.2.Truffle platform was used due to its efficient consensus mechanism compared to the available alternative (parity). Thus issues around appropriate consensus mechanism are solved. Truffle is a development environment, testing framework and asset pipeline for blockchains using the Ethereum Virtual Machine (EVM), aiming to make life as a developer easier (Ahmad, 2017). It has become one of the most widely used IDEs and framework in the Ethereum community. Developers use it to build and deploy Dapp for testing purposes with many features that make it more attractive to users with a Web 3.0 development background. It has Built-in smart contract compilation, linking, deployment and binary management with Mocha and Chai Automated contract testing. It is also known to be Configurable build pipeline with support for custom build processes and Scriptable deployment & migrations framework. The following steps describe how to install and run truffle.

Truffle Framework Environment Setup

To use truffle framework, the following discussed dependencies and software must be configured. These include NodeJS, Web3, Node packet manager NPM, and lite server.

a) NodeJS v8.9.4 or later

Node.js is an open-source, cross-platform JavaScript run-time environment that executes JavaScript code outside of a browser. Both the JavaScript and Node.js run on the V8 JavaScript runtime engine. This engine takes the JavaScript code and converts it into faster machine code. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient.

b) The Node Package Manager (npm)

npm is a package manager for the JavaScript programming language written entirely in JavaScript and was developed by Isaac Z. It is the default package manager for the JavaScript runtime environment Node.js. It consists of a command line client, also called npm, and an online database of public and paid-for private packages called the npm registry. The registry is accessed via the client, and the available packages can be browsed and searched via the npm website. The package manager and the registry are managed by npm.

c) Lite Server

Lite-server is a lightweight development web server with support for Single Page Apps (SPAs). Lite-server is a simple customized wrapper around Browser Sync to make it easy to serve Single Page Applications (SPAs). It is Lightweight development only node server that serves a web app, opens it in the browser, refreshes when html or JavaScript change, injects CSS changes using sockets and has a fallback page when a route is not found. Browser Sync is super-fast lightweight development server. It serves the static content, detects changes, refreshes the browser, and offers many customizations. Lite-server uses Browser Sync and allows for configuration overrides via a local bs-config.json or bs-config.js file in a project.

d) Metamask

Metamask is a browser plugin that allows users to make Ethereum transactions through regular websites. It is a bridge that allows you to visit the distributed web of tomorrow in your browser (MetaMask, 2018). It allows running of an Ethereum Dapps on browser without running a full Ethereum node. This facilitates the adoption of Ethereum because it bridges the gap between the user interfaces for Ethereum, for example, Mist browsers or Dapps and the regular web browser such as Chrome or Firefox. According to metamask (2018), MetaMask injects a JavaScript library called web3.js into the namespace of each page your browser loads. Furthermore, MetaMask allows users to specify which Ethereum node to send these requests to. The ability to send requests to nodes outside of the user’s computers is important because it means that people can use Ethereum without having to download a node consisting of 10+GB blockchain on to their computers. The MetaMask add-on can be installed in Chrome, Firefox, Opera, and the new Brave browser (Coin, 2018). To use MetaMask, this research installs it on the Chrome browser.

e) Web3

Web3.js is a collection of libraries which allow interaction with a local or remote Ethereum node, using an HTTP or RPC connection. It is for the Ethereum blockchain and smart contracts. The whisper protocol to communicate p2p and broadcast, swarm protocol, the decentralized file storage and has web3-utils helper functions for Dapp development. Web3.js talks to The Ethereum Blockchain with JSON RPC (Remote Procedure Call) protocol. Web3.js allows requests to an individual Ethereum node with JSON RPC in order to read and write data to the network as shown below.

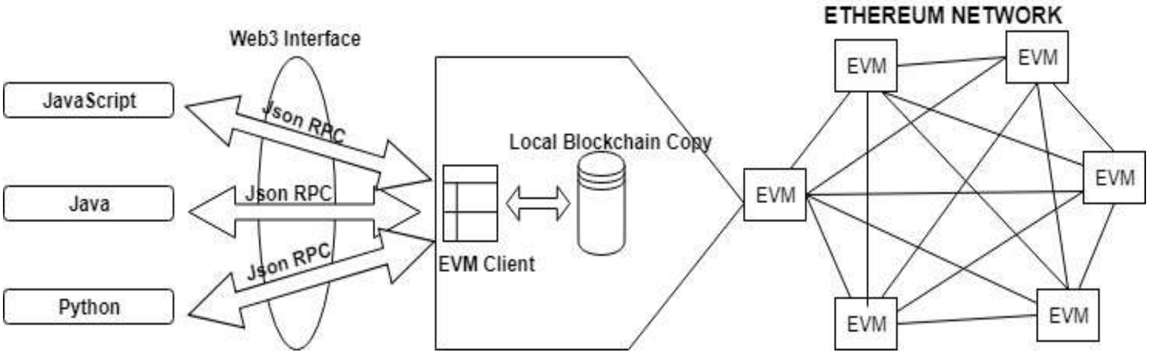


Figure 16 interaction with a local or remote ethereum node, using web3 interface*Source:* (Researcher, 2019)

4.3 Development of the Blockchain Based Smart Contract Model for Real Estate

This section demonstrates the attainment of research objective three of the study which required the researcher to develop a blockchain based smart contract model for real estate. The entire code of the developed model is presented in appendix I.

4.3.1 Development of User Registration and Authentication

To register means creating a new user account, that is a record in the database describing how you will prove your identity (Potterton *et al*, 2018). This ensures that model Participants can sign up in a convenient manner and save time that would have otherwise been used in manually sorting out illegible or inaccurate forms. To authenticate you means providing evidence that your identity matches the one described in your user account (Schmidt *et al*, 2016).

i) **User Registration**

The registration process is the entry point into the model and provides for the two types of system users, namely; tenant and the landlord. Registered users normally provide some credentials (such as Address and password) to the system in order to prove their identity. This is done during logging in process. Systems intended for use by the real estate crews and would often allow any user to register simply by selecting a register function and providing these credentials for the first time. Registered users are granted privileges beyond those granted to unregistered users. During the registration process, tenants must register by providing their details and set an eight character password. The user registration process is outlined in Figure 25 as shown below. The registration logic is performed by the function register as shown in listing 1 below.

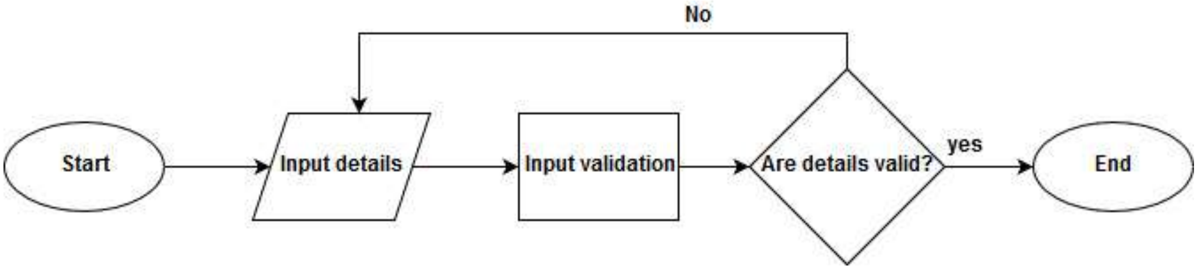


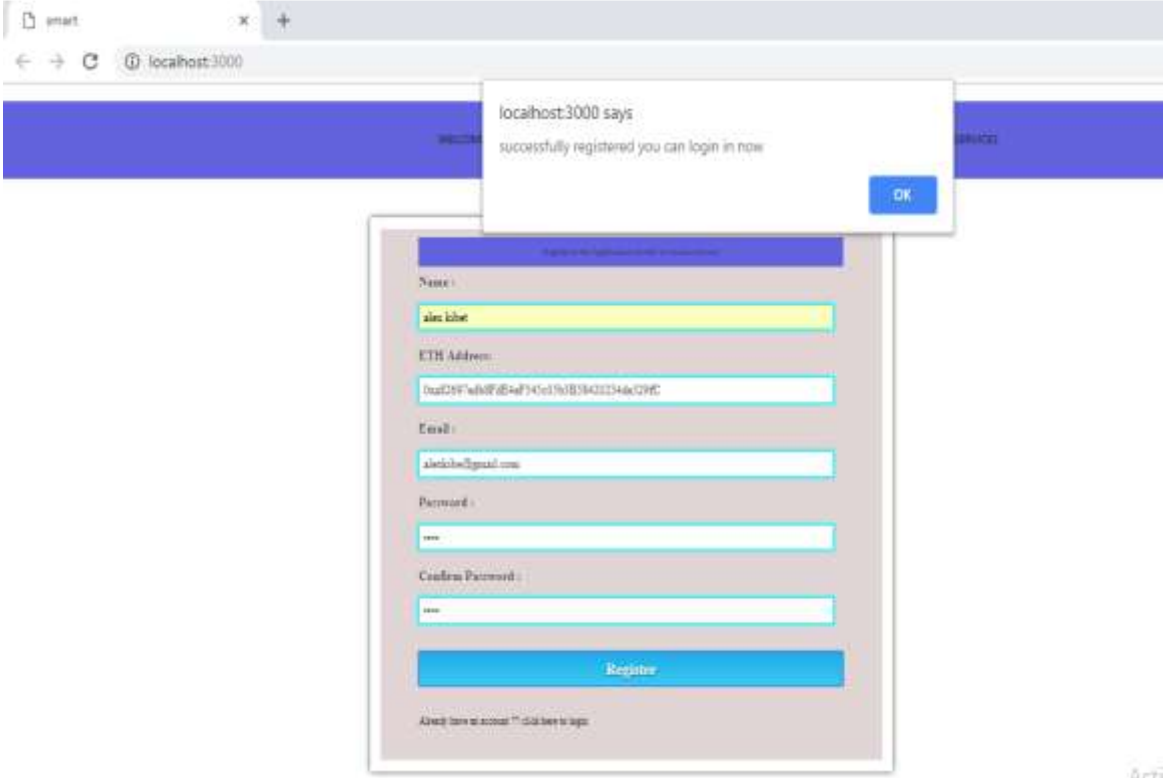
Figure 17 user registration Process Flowcharts Source: (Researcher, 2019)

List 1 Registration Function Source: (Researcher, 2019)

```
function register(address _user, string memory _password,string memory _email, string memory _name)public returns(bool success)
{
    address[] storage registed = registeredUsers;
    User storage user = Users[msg.sender];
    registed.push(msg.sender);
    require(!user.registered);
    require(msg.sender == _user);
    user.password = _password;
    user.name = _name;
```

```
        user.email = _email;
        user.registered = true;
        return true;
    }
```

In this research, the implementation of the PoC ensures that the user registration details are stored in a smart contract (TenantInfoDb). TenantInfoDb is a database contract to store information about tenants. The information stored for each tenant is personal information and Addresses. A Tenant-struct is created for each Tenant to contain the corresponding data as well as a next and a prev-attribute so that a doubly-linked list can be used to iterate over them. The transaction between the user and the smart contract during registration is shown in figure 23 below.



localhost:3000 says

successfully registered you can login in now

Name : alex kibet

ETH Address: [illegible]

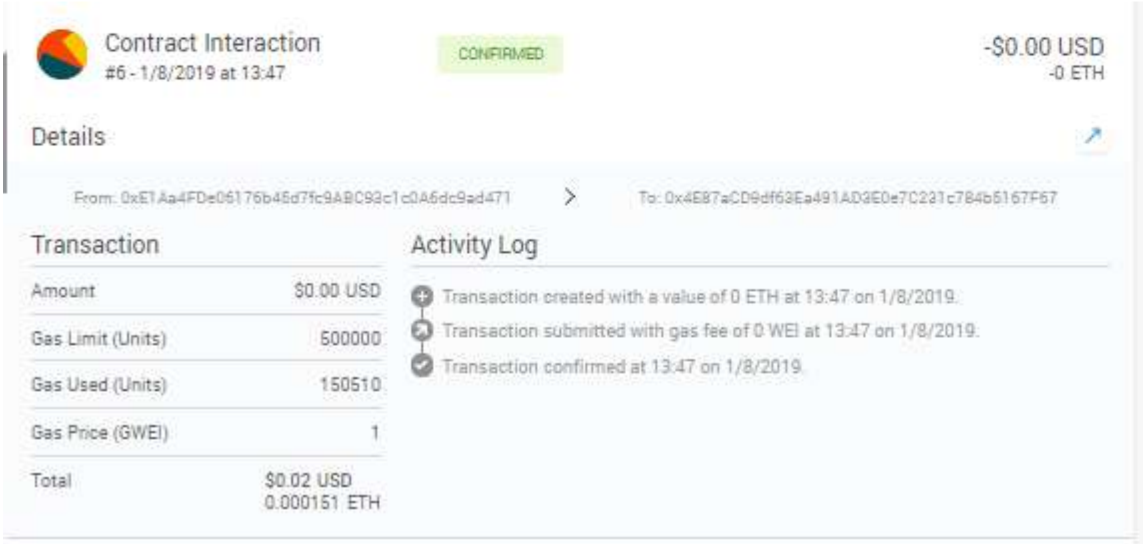
Email : alexkibet@gmail.com

Password : ****

Confirm Password : ****

Figure 18 User Registration Form with a positive acknowledgement of a successful registration Source: (Researcher, 2019)

The successful transaction creates a blockchain record and activity log as shown in figure 28 below.



Contract Interaction
#6 - 1/8/2019 at 13:47

CONFIRMED

-$0.00 USD
-0 ETH

Details

From: 0xE1Aa4FDe06176b45d7fc9ABC93c1c0A6dc9ad471 > To: 0x4E87aCD9df63Ea491AD3E0e7C231c784b5167F67

Transaction

Amount	$0.00 USD
Gas Limit (Units)	500000
Gas Used (Units)	150510
Gas Price (GWEI)	1
Total	$0.02 USD 0.000151 ETH

Activity Log

- Transaction created with a value of 0 ETH at 13:47 on 1/8/2019.
- Transaction submitted with gas fee of 0 WEI at 13:47 on 1/8/2019.
- Transaction confirmed at 13:47 on 1/8/2019.

Figure 19 transaction history and activity log Source: (Researcher, 2019)

ii) **User Authentication**

The user authentication process provides a mechanism of providing evidence that users identity matches the one described in the user account. This ensures that the system users are identified in a cryptographically secure manner so that no unauthorized entities can access it the network. The solidity code implemented to allow authentication is shown in the listing 2 below and the login form is shown in figure 30. The authentication process is as shown in figure 25 below.

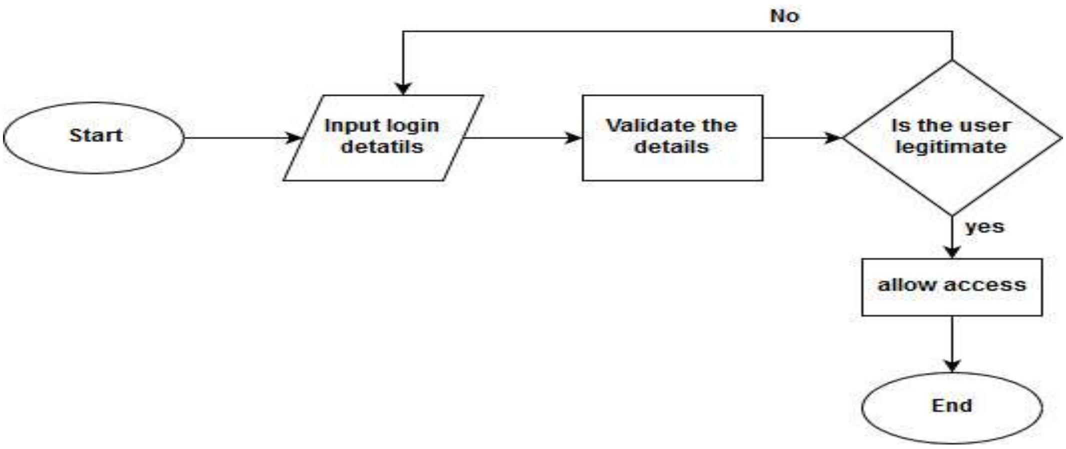


Figure 20 User authentication Flowchart Source: (Researcher, 2019)

List 2 User Authentication Solidity Code Source: (Researcher, 2019)

```
function login (string memory _password) public view returns (bool )
        {
                User storage user = Users[msg.sender];
                Require (user. registered);
                return  (keccak256  (abi.encodePacked  (user.password))  ==     keccak256(
                abi. encodePacked (_password)));


        }
```

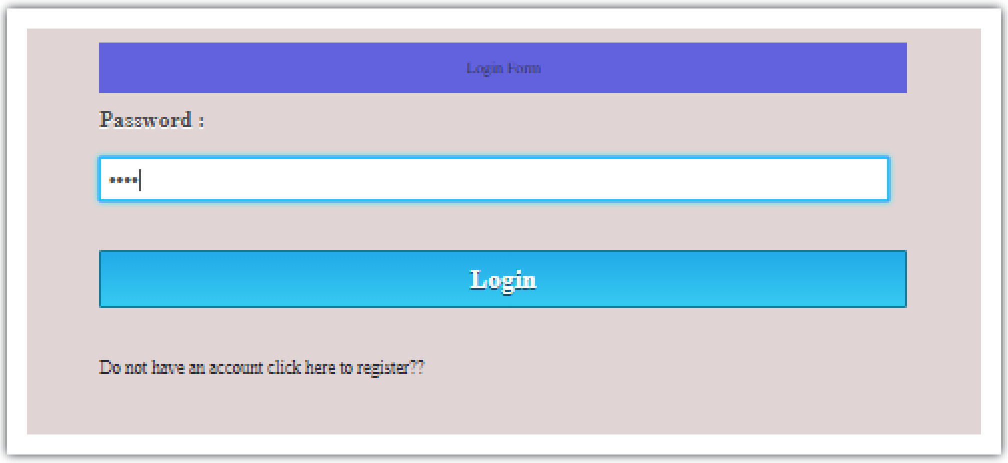


Figure 21 login form Source: (Researcher, 2019)

4.3.2 Development of the Rental Payment Module

The rental fee is a payment made for the temporary use of a good, facility, equipment, service or property owned or provided by another (Lovelock & Patterson, 2015). It is used in this research to mean payment of fixed amount fixed by contract, made by a tenant at specified intervals in return for the right to occupy or use the property of another. A gross lease is when the tenant pays a flat rental amount and the landlord pays for all property charges regularly incurred by the ownership (McCabe, 2019). In this research, it is assumed that the tenants are paying a gross lease. The solidity code implemented to allow rental fee payment is shown in the listing 3 below and the payment form is shown in figure 30. The payment process is as shown in figure 25 below.

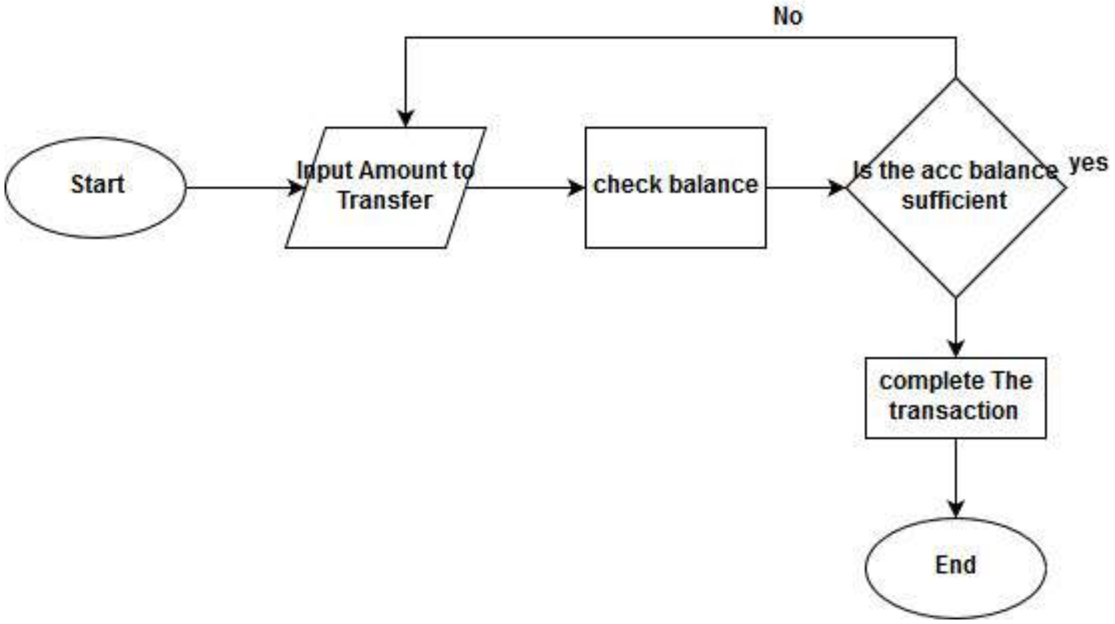
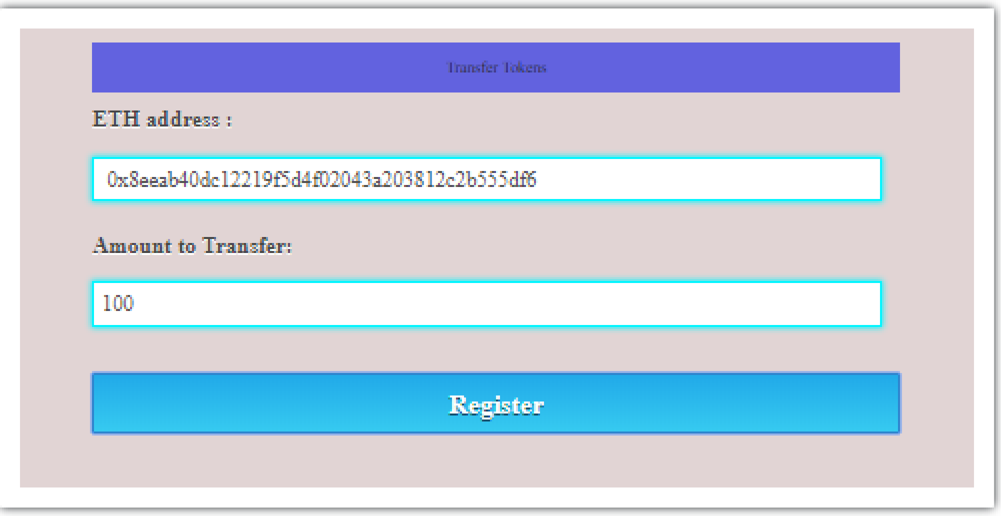


Figure 22 the rental payment DFD Source: (Researcher, 2019)

List 3 Rental Fee Payment Function Source: (Researcher, 2019)

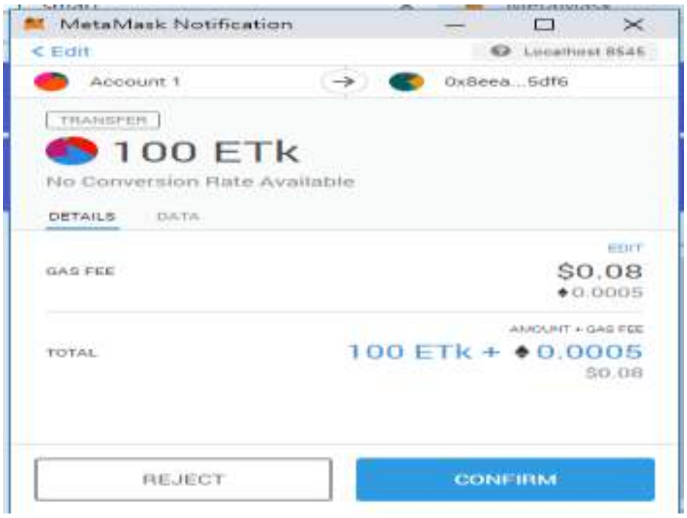
```
function rentRoom (bytes32  _housename, uint256  _roomnumber, uint256 _amount)
public returns (bool success)
{
        Require ( _amount> = roomPrice);
        EtokenContract.approve (address (this), _amount);
        EtokenContract .transferFrom (msg.sender,address (this) ,roomPrice);
        HouseContract. updateRentRoom (_housename, _roomnumber );
        return true;
}
```



ETH address : 0x8eeab40dc12219f5d4f02043a203812c2b555df6

Amount to Transfer: 100

Figure 23 a tenant initiating transaction Source: (Researcher, 2019)



100 ETk

No Conversion Rate Available

GAS FEE $0.08 ♦0.0005

AMOUNT + GAS FEE

TOTAL 100 ETk + ♦0.0005 $0.08

Figure 24 MetaMask requesting for authorization to complete transaction Source: (Researcher, 2019)

4.3.3 Development of the Balance Enquiry Module

An account balance inquiry is a review of any type of account, whether it be a depository account or credit account. The inquiry can refer to past records, payments or other specific transactions, or any other entries relating to the account. Account inquiry may typically be initiated when an individual seeks to take on new debt, particularly in conjunction with making a substantial purchase such as the acquisition of a property or when the person involved has received some payment. Account inquiries might also be made to ensure that payments that are owed have cleared and were delivered on time. Before the individual is approved for the new transaction, many people would wish to check their account balance. In most cases, this is a step in assessing the overall potential of settling a debt or purchasing power of an individual at a given time given an account. In the PoC, an account holder may initiate an account balance inquiry themselves, particularly if they are about or after carrying out a transaction. For instance, if a landlord would wish to check if he/she has received rental payment or a tenant checking whether a transaction was completed successfully.

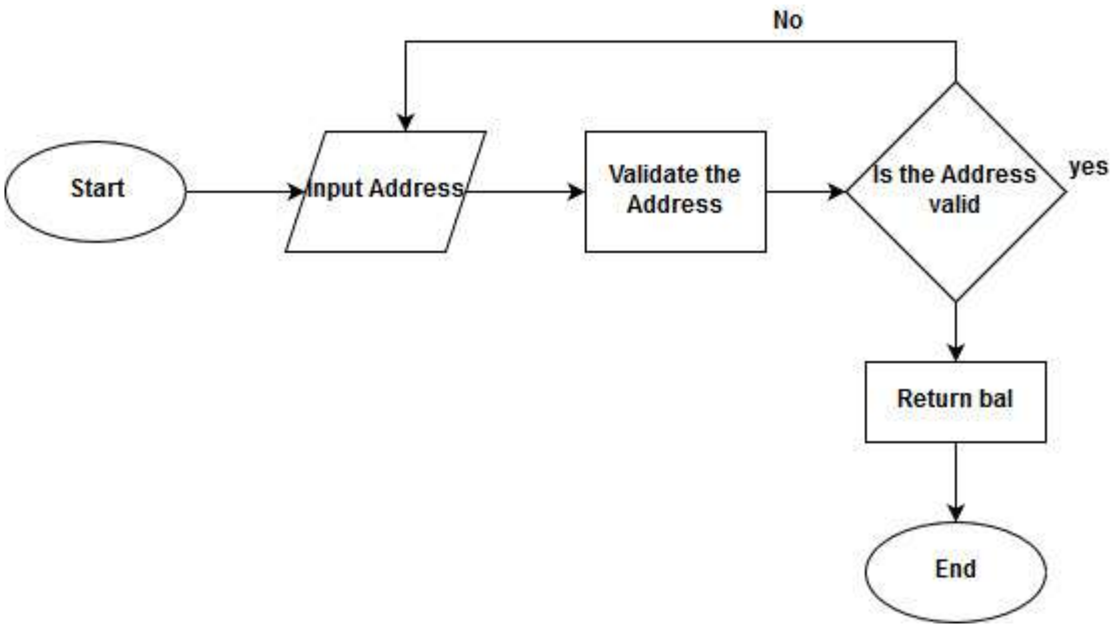
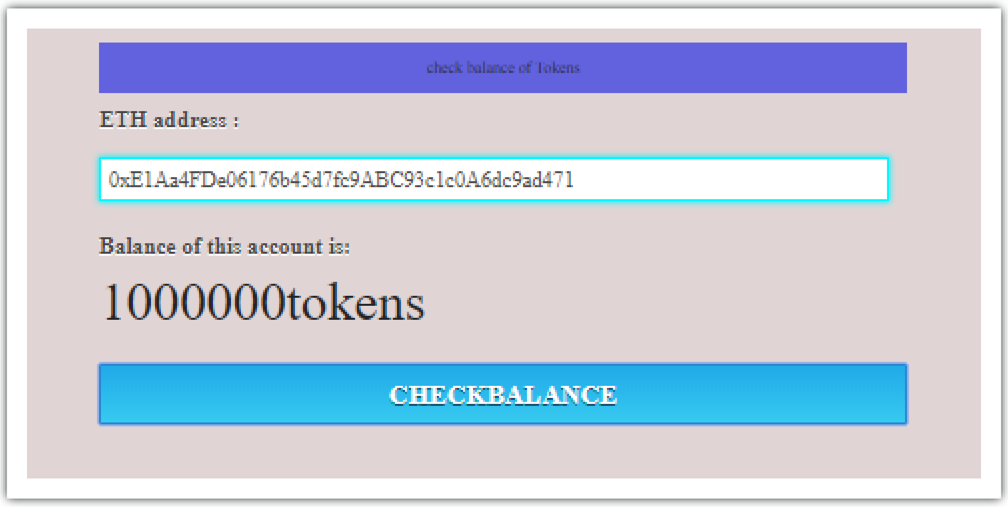


Figure 25 Balance Enquiry DFD Source: (Researcher, 2019)



check balance of Tokens

ETH address :

0xE1Aa4FDe06176b45d7fc9ABC93c1c0A6dc9ad471

Balance of this account is:

1000000tokens

Figure 26 checking balance Source: (Researcher, 2019)

4.3.4 Deployment of Smart Contracts to Blockchain Network.

Before a project launches on the Ethereum blockchain, it is deployed to an Ethereum Test Network (“testnet”), which simulates Ethereum. Testnet is a place to test smart contracts solutions. Basically, it’s a clone of the Ethereum network that allows you to deploy and test your smart contracts without paying real fees. This gives developers a chance to kick the tires before real assets are involved. Ether and tokens on a testnet are easy to obtain and carry no real-world value. There are three testnets currently in use, and each behaves similarly to the production blockchain (where real Ether and tokens reside) including kovan, Rinkeby and ropsten. The researcher used Rinkeby test network due to its efficient consensus mechanism. Rinkeby is a proof-of-authority based blockchain, started by the Geth team. Test ethers in Rinkeby can’t be mined but rather it is requested from the faucet. The convenient and easy way of getting test ethers and efficient consensus algorithm justifies the reason for its selection.

To deploy a project, Rinkeby node containing all the time transactions is downloaded. Upon successful download, MetaMask can be used to send Ether and tokens. Switching from the Main Ethereum Network to Rinkeby is also facilitated (Saadat, 2019). From MetaMask, one should see his/her balances and transaction history update, to reflect the selected network. It worth noting that Ethereum addresses & private keys that work on Ethereum works on each testnet. Therefore, one has to be extremely careful not to send Ether or tokens on the

Ethereum main-net to a testnet address. Upon successful deployment, the blockchain-based smart contracts model for real estate was demonstrated by executing PoC on a Rinkeby testnet based Transaction. This was done upon successful deployment and of the PoC project. To successfully demonstrate this transaction process the following steps were followed.

Step I

Initiate and authorize a transaction inside the deployed project as shown in figure 43 below. On a successful transaction, The MetaMask Bridge records transaction history as shown in figure 44.

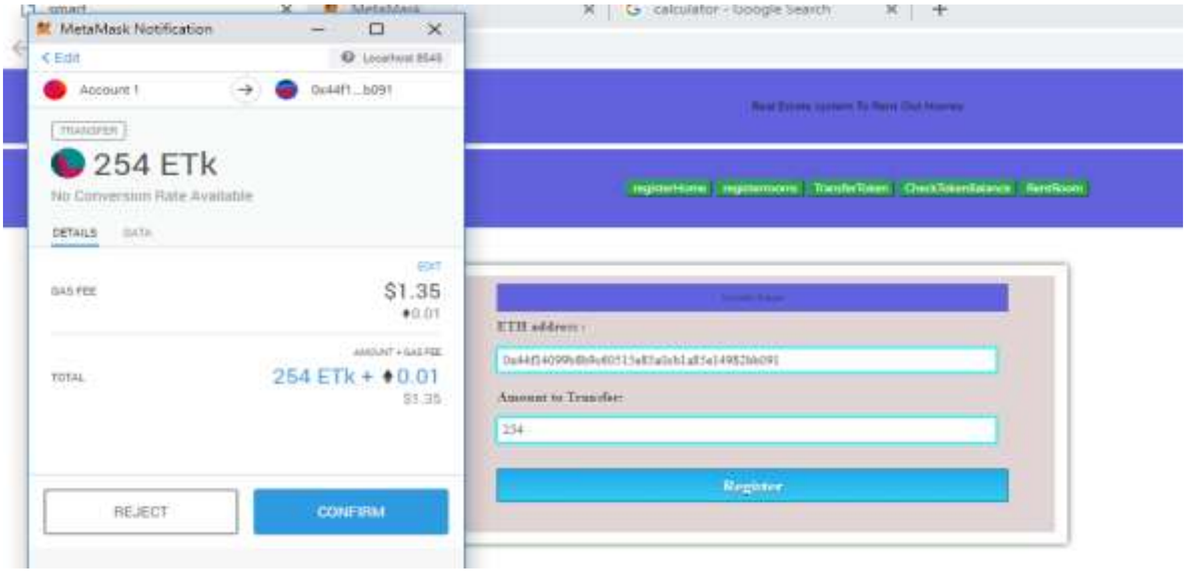


Figure 27 initiated transaction inside the deployed project Source: (Researcher, 2019)

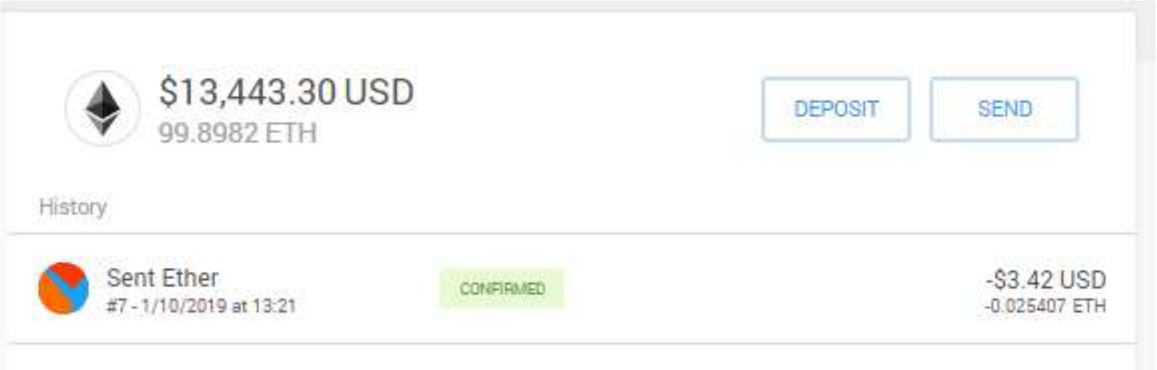
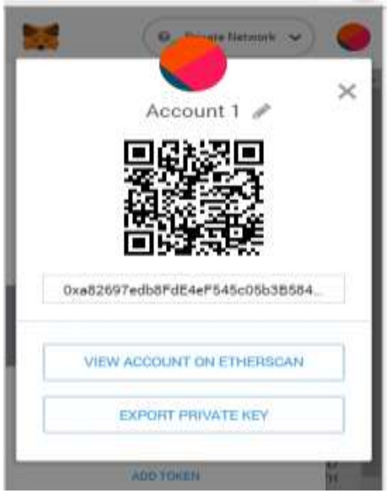


Figure 28 transaction was executed successfully Source: (Researcher, 2019)

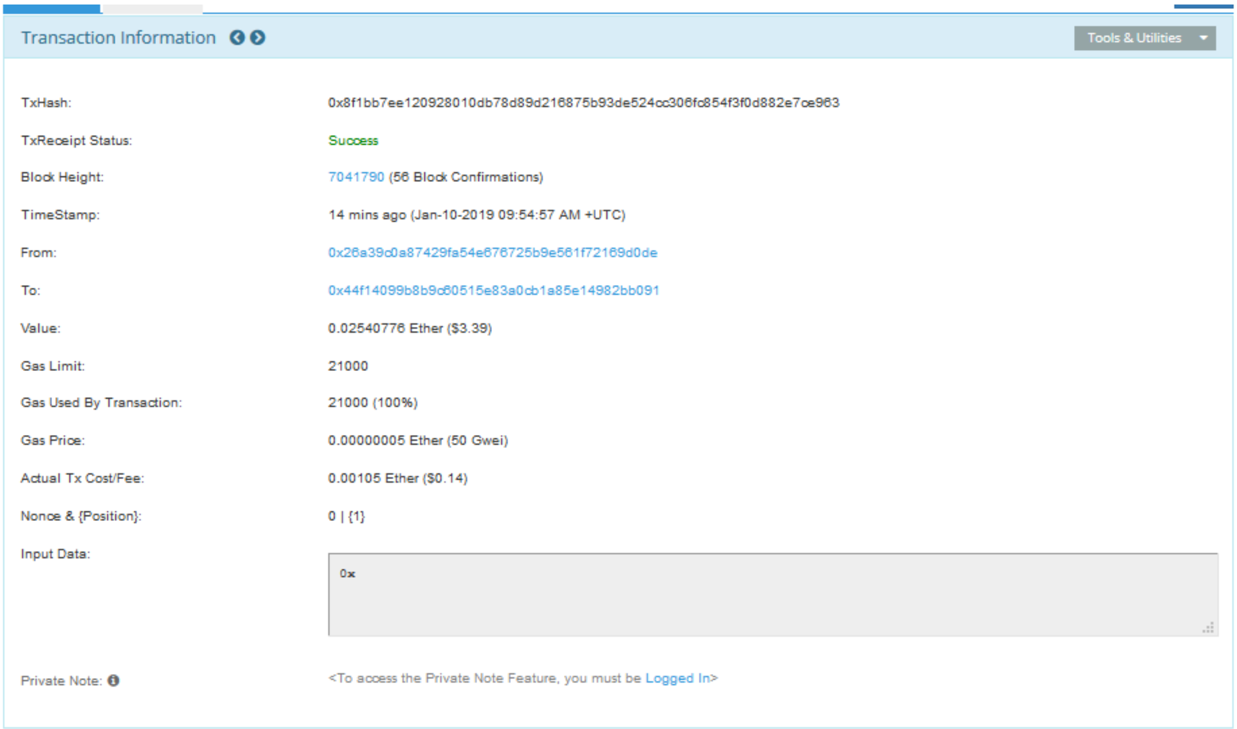
Step II

Open the account transaction details on Etherscan block explore. Etherscan is a BlockExplorer for the Ethereum Blockchain. A BlockExplorer is basically a search engine that allows users to easily lookup confirm and validate transactions that have taken place on the Ethereum Blockchain. It gives a better way to examine block header information as explained in section 2.2. It further allows exploration and appreciation of blockchain architecture as well as proving the realization of general objectives in this study (using blockchain to store transaction records). The following subsequent three figures show how to open Etherscan block explorer, transaction block body and header respectively.



0xa82697edb8FdE4eF545c05b3B584...

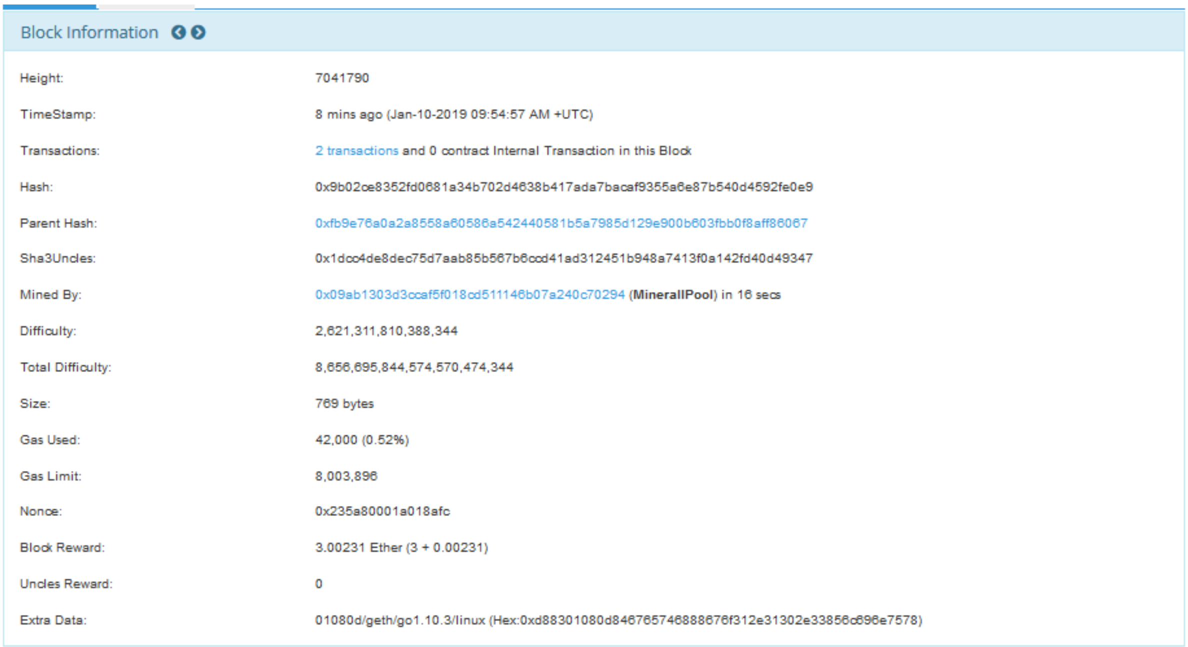
Figure 29 Opening Etherscan block explore Source: (Researcher, 2019)



Transaction Information

TxHash:	0x8f1bb7ee120928010db78d89d216875b93de524cc306fc854f3f0d882e7ce963
TxReceipt Status:	Success
Block Height:	7041790 (56 Block Confirmations)
TimeStamp:	14 mins ago (Jan-10-2019 09:54:57 AM +UTC)
From:	0x26a39c0a87429fa54e676725b9e561f72169d0de
To:	0x44f14099b8b9c60515e83a0cb1a85e14982bb091
Value:	0.02540776 Ether ($3.39)
Gas Limit:	21000
Gas Used By Transaction:	21000 (100%)
Gas Price:	0.00000005 Ether (50 Gwei)
Actual Tx Cost/Fee:	0.00105 Ether ($0.14)
Nonce & {Position}:	0 | {1}
Input Data:	0x
Private Note:	<To access the Private Note Feature, you must be Logged In>

Figure 30 transaction Block body Source: (Researcher, 2019)



Block Information

Height:	7041790
TimeStamp:	8 mins ago (Jan-10-2019 09:54:57 AM +UTC)
Transactions:	2 transactions and 0 contract Internal Transaction in this Block
Hash:	0x9b02ce8352fd0681a34b702d4638b417ada7bacaf9355a6e87b540d4592fe0e9
Parent Hash:	0xfb9e76a0a2a8558a60586a542440581b5a7985d129e900b603fbb0f8aff86067
Sha3Uncles:	0x1dcc4de8dec75d7aab85b567b6ccd41ad312451b948a7413f0a142fd40d49347
Mined By:	0x09ab1303d3ccaf5f018cd511146b07a240c70294 (**MinerallPool**) in 16 secs
Difficulty:	2,621,311,810,388,344
Total Difficulty:	8,656,695,844,574,570,474,344
Size:	769 bytes
Gas Used:	42,000 (0.52%)
Gas Limit:	8,003,896
Nonce:	0x235a80001a018afc
Block Reward:	3.00231 Ether (3 + 0.00231)
Uncles Reward:	0
Extra Data:	01080d/geth/go1.10.3/linux (Hex:0xd88301080d846765746888676f312e31302e33856c696e7578)

Figure 31 transaction Block header information Source: (Researcher, 2019)

4.4 Model Prototype Validation and Verification

In this section the results of research objective four which involved validation and verification of the implemented smart contract model prototype are presented. The prototype presented in section 4.3 was evaluated in two steps which are, Verification to ascertain how the model overcomes the weaknesses of the existing models and validation to ascertain how the model fulfills the evaluation criteria based on user stories.

4.4.1 Verification to Ascertain How the Model Overcomes the Weaknesses of the Existing Models

Nakuru country had 18 registered real estate agents at the time of this research. From those agents, only two were using IT-based service. Therefore, this research used the two housing agents in the model Verification to ascertain how the weaknesses of existing models were addressed. They were allowed to interact with the model for real estate management and the report is presented below. The tenants were able to register on the housing market platform. Also, interacted with the model such that tenants bought ethers and E-Tokens from their agents. Discussed below are sample reports from agency A and agency B.

i) Sample Report of Agent "A"

As shown in the table below, it can be noticed that agent A registered ten tenants into the platform and all of them paid the rental fee using ETokens equal to the value of their rental fee. The blocks of transactions can be trailed from when the agency initiated the exchange process and transferred the amounts needed by the tenant. Tough this research focused on the rental fee only, it worth noting that other transactions including registration, signing agreements, information update, or removal is captured in the blockchain.

Table 8 Sample Report of Agent “A”

Txhash	Blockno	Timestamp	DateTime	From	To	Quantity
0x4f9e909e96416d9d4e2cd18460941df2bd54ba0c46dd8eebb8a792301e8aed80	5070574	1550842150	2/26/2019 13:29	0x32cd37a01a6286058507d1f018c8cb8676a90aaf	0x13e7b6833e25e19b9d19ba3153bb95c173afdd34	20000
0xa4df7c7bd246a9e5de7bfd6bac29e9a5cadffa3a6da2dd8ac4a22faeced772ca	5102742	1551259083	2/27/2019 9:18	0xc6e532bb14d444a4af14614f300453609018c96c	0x32cd37a01a6286058507d1f018c8cb8676a90aaf	3000
0x8f2a22c99ab613f27201062f08c8251ba2669a3dd5dd83e5119bd7dd41559479	5137983	1551718836	2/29/2019 5:00	0x0289bd742d56e94dd04f7ba09185c5a139b35b2f	0x32cd37a01a6286058507d1f018c8cb8676a90aaf	2300
0x315d74a791457efec5c9a176a30e983f4f3d15176064de335648a806f0f03316	5234443	1552992765	3/4/2019 17:00	0x13e7b6833e25e19b9d19ba3153bb95c173afdd34	0x32cd37a01a6286058507d1f018c8cb8676a90aaf	1700
0xdd787bd3a83f166289fcd3b4b91b8e02e9fa2e0d2895b36e5b93336ace5b02a1	5235275	1553003197	3/7/2019 13:46	0x13e7b6833e25e19b9d19ba3153bb95c173afdd34	0x32cd37a01a6286058507d1f018c8cb8676a90aaf	1100
0xa799b13f9150e48c8bf8c85e09c0eed0a72c85ee0b8243865ed1d1dcefbd5217	5235278	1553003214	3/9/2019 13:46	0x13e7b6833e25e19b9d19ba3153bb95c173afdd34	0x32cd37a01a6286058507d1f018c8cb8676a90aaf	2000
0x4e4b945584bb863b3a648b70e007130db4f214294f72fe2f8d39322f94c979b2	5235449	1553005518	3/10/2019 14:25	0x13e7b6833e25e19b9d19ba3153bb95c173afdd34	0x32cd37a01a6286058507d1f018c8cb8676a90aaf	2760
0x693fe0148757bdc6977d728a8711b8ec8814fc6e9593a343681c55a67bf69ac6	5710843	1559395986	3/11/2019 13:33	0xebd3872ea4d1d85a03a581127b872c99ebbab000	0x32cd37a01a6286058507d1f018c8cb8676a90aaf	1900
0xda1da60d42cc73963a9d4d251e2a798fb707c7ca1e61eb93f622c051894e1e23	5716807	1559478661	6/12/2019 12:31	0xbb84b6d71c6c6626809bc1b971c9a1d9b2eb5000	0x32cd37a01a6286058507d1f018c8cb8676a90aaf	1000
0xec2b91a3e98bb1b313f26bbc41b0fa53d8ad23995778c70ccfc00d7690316c28	5730858	1563214002	6/12/2019 17:31	0x861b8b7c649d9717673f53dd8f02fc43c6fee09d	0x32cd37a01a6286058507d1f018c8cb8676a90aaf	1200

ii) Sample Report Of Agent “B”

As shown in the table below, it can be noticed that agent B registered 15 tenants into the platform and all of them paid rental fee using ETokens equal to the value of their rental fee. The blocks of transactions can be trailed from when the agency initiated the exchange process and transferred the amounts needed by tenant. Tough this research focused on the rental fee only, it worth noting that other transactions including registration, signing agreements, information update, or removal is captured in the blockchain.

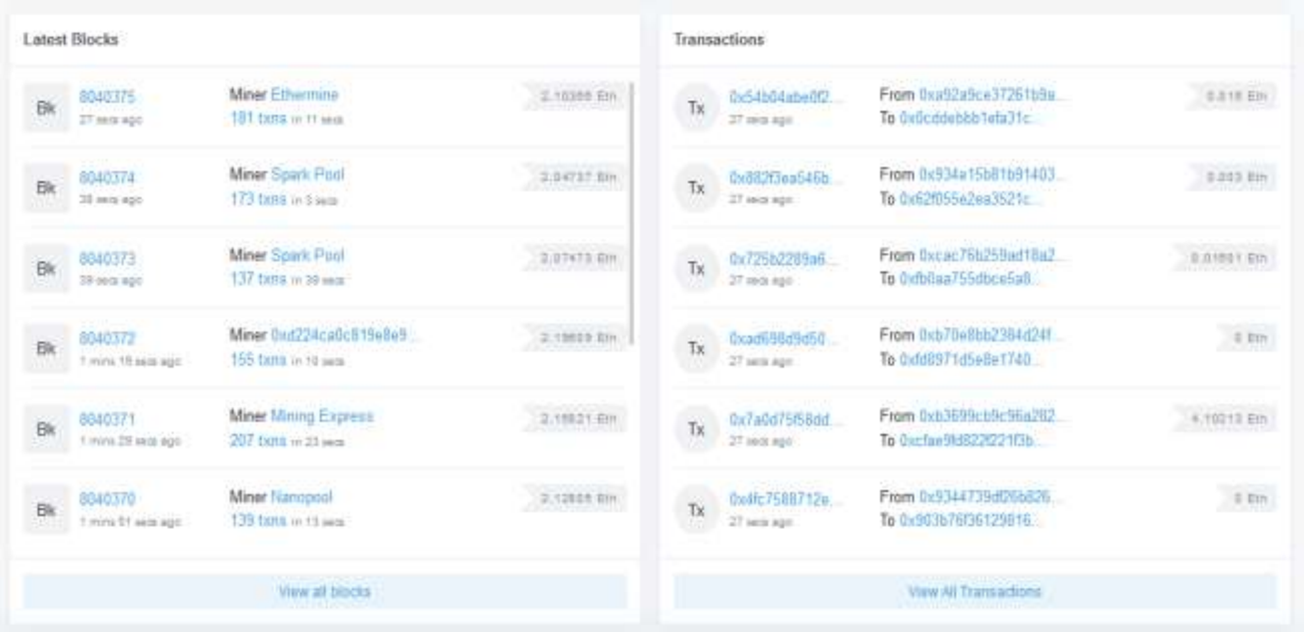
Table 9 Sample Report Of Agent "B"

Txhash	Blockno	Timestamp	DateTime	From	To	Quantity
0x718d928d0fd11e0aa208f68790fd6cc5fae209d02f948e7d28ee34b30c9ddd37	5710830	1550747144	2/22/2019 12:33	0x236ad9817ae3d957532e279469eb86f1dd673d8	0x13e7b6833e25e19b9d19ba3153bb95c173afd	2500
0xde802d9d1551c1034a33229de120b7ec8f439b09ff34532321fff6e34c6dbb62	5710830	1550747180	2/23/2019 12:34	0x08e743af812a37751c1cfc0fd61f6b997eec0c6c	0x13e7b6833e25e19b9d19ba3153bb95c173afd	2500
0x22762b88615c52f8fc00b19bbd8edbf4d1ccfb4ba2a1a32bf2301c6b0ad8a35b	5710830	1550747237	2/23/2019 12:35	0x64404296855fef130c1f215a3dbf9d940b332607	0x13e7b6833e25e19b9d19ba3153bb95c173afd	2500
0xca5532d2db4b4c52a0167172e2762dd2a3a0e37b9d81131397229e83d8258244	5710830	1550747399	2/25/2019 12:36	0x3f4eb593904495a346df4bc7c80a55a519aa4295	0x13e7b6833e25e19b9d19ba3153bb95c173afd	2500
0x493dd0186d383d43c515b2a752b2ea16d88c446bf910070610a1792bb9825b07	5710830	1550747710	2/26/2019 13:00	0x3f4eb593904495a346df4bc7c80a55a519aa4295	0x13e7b6833e25e19b9d19ba3153bb95c173afd	2500
0x5241679488b904366221c183f26aa531e6129e0a047e479037f26b4f612e10ec	5710830	1550748107	2/27/2019 13:20	0x08e743af812a37751c1cfc0fd61f6b997eec0c6c	0x13e7b6833e25e19b9d19ba3153bb95c173afd	2500
0xf849bdc366a4a85700f703b541ef323d15b4c450b43e4d0766dbc510429916	5710830	1550748160	2/28/2019 13:25	0xcc2acae48f60a324c8ae0fa2c991b8e741d72652	0x13e7b6833e25e19b9d19ba3153bb95c173afd	2500
0xbfdede075bdf7966d01952e0a0c55dedc7e030b4b60c332e8494eaf230339097	5710830	1550748166	2/28/2019 13:29	0x13e7b6833e25e19d19ba3153bb95c173afdd34	0x13e7b6833e25e19b9d19ba3153bb95c173afd	2500
0x34b7190f2f17db1057ae3c1b8aa35c6d0db804043c376a5c14efb60235705d76	5710830	1550748223	3/1/2019 9:18	0xbb84b6d71c6c662609bc1b971c9a1d9b2eb5000	0x13e7b6833e25e19b9d19ba3153bb95c173afd	2500
0xbd25f6fba8b002c5b3f272d82cf0f682794e490d2176f1785727251c3cf40921	5710830	1550748285	3/4/2019 17:00	0x861b8b7c649d9717673f53dd8f02fc43c6fee09d	0x13e7b6833e25e19b9d19ba3153bb95c173afd	2500
0xe9c2d81af3a12f6a76ac798ef19f92be377d21229375c479849a3c121c26dccc	5710830	1550748308	3/9/2019 13:43	0xfe96912711c1fc37168df6e84e0290339f3b2141	0x13e7b6833e25e19b9d19ba3153bb95c173afd	2500
0x2ddabe39c7479e4b988f51231dfbecfa59e3e840606db0a3aeb797348d9c74e5	5710830	1550838175	3/9/2019 20:37	0x69670b0c1b10073812415dd474804bb32b3aeca	0x13e7b6833e25e19b9d19ba3153bb95c173afd	2500
0x6f24353cc3b6cb4ac3f2e356067d9e5e50bc1924dc11bdd85879fd6f4156aa90	5710830	1550838784	3/10/2019 20:37	0x69670b0c1b10073912415dd474804bb32b3aeca	0x13e7b6833e25e19b9d19ba3153bb95c173afd	2400

From the above test data, it can be seen that the two agents test data shows blockchain consisting of a series of blocks that are used to store information related to transactions that occur on a blockchain network. A block (row) contains a unique header, and each such block is identified by its block Txhash (transaction hash) and other information /blockchain parameters as discussed in section 2.3. Though not all the block header or body information is represented here, it can adequately prove blockchain transaction. Based on the assessment criteria presented in Table 10 (Key of Assessment Criteria), the following describe the fulfillment of these criteria.

a) Transparency

Fulfilled by blockchain infrastructure both in the component and the contract layer where transactions are recorded in a distributed ledger thus improving Transparency. Public verifiability allows anyone to verify correct system working. This also ensures a situation in which transaction activities are done in an open way without the need for a trusted party. The following figure presents a BlockExplorer that allows users to easily lookup, confirm and validate transactions that have taken place on the Ethereum Blockchain. This promotes transparency and transaction verifiability.



Latest Blocks

Bk 8040375 27 secs ago
Miner Ethermine
181 txns in 11 secs
2.10388 Eth

Bk 8040374 38 secs ago
Miner Spark Pool
173 txns in 5 secs
2.04737 Eth

Bk 8040373 39 secs ago
Miner Spark Pool
137 txns in 39 secs
2.07473 Eth

Bk 8040372 1 mins 18 secs ago
Miner 0xd224ca0c819e8e9...
155 txns in 10 secs
2.19839 Eth

Bk 8040371 1 mins 29 secs ago
Miner Mining Express
207 txns in 23 secs
2.15821 Eth

Bk 8040370 1 mins 51 secs ago
Miner Nanopool
139 txns in 13 secs
2.12888 Eth

View all blocks

Transactions

Tx 0x54b04abe0f2... 27 secs ago
From 0xa92a9ce37261b9a...
To 0x0cddebbb1efa31c...
0.818 Eth

Tx 0x882f3ea546b... 27 secs ago
From 0x934a15b81b91403...
To 0x62f055e2ea3521c...
0.243 Eth

Tx 0x725b2289a6... 27 secs ago
From 0xcac76b259ad18a2...
To 0xfb8aa755dbce5a8...
0.01801 Eth

Tx 0xad698d9d50... 27 secs ago
From 0xb70e8bb2384d24f...
To 0xfd8971d5e8e1740...
0 Eth

Tx 0x7a0d75f58dd... 27 secs ago
From 0xb3699cb9c96a282...
To 0xcfae9fd822f221f3b...
4.10213 Eth

Tx 0x4fc7588712e... 27 secs ago
From 0x9344739df26b826...
To 0x903b76f36129816...
0 Eth

View All Transactions

Figure 32 a BlockExplorer

b) Immutability

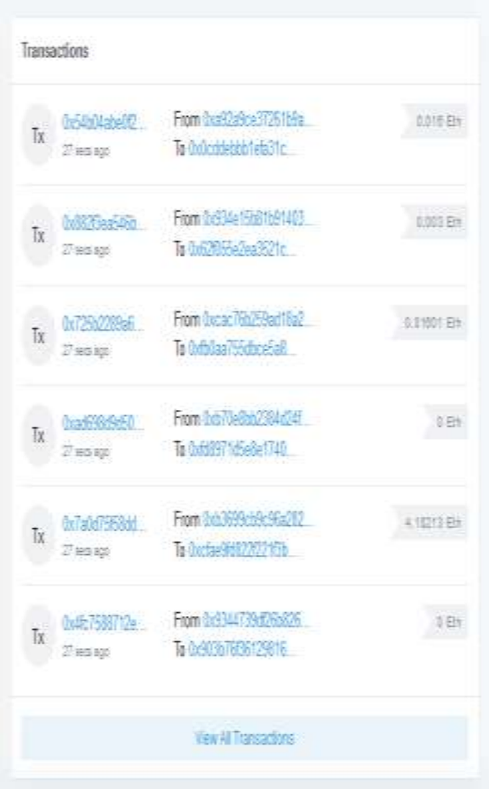
As a part of a standard mining exercise, a block header is hashed repeatedly by miners by altering the nonce value to create Txhash (transaction hash) for transaction block identification as shown in the first column of tables 8-9. Through this exercise, they create a proof of work, which helps miners get rewarded for their contributions to keep the blockchain system running and ensuring consistency state. Upon successful transaction validation, it is spread across the network with block number (Blockno), Timestamp and the Transaction harsh value to ensure that it is nearly impossible to tamper. Additionally, each broadcasted block would be validated by other nodes and transactions would be checked. So any falsification could be detected easily.

Table 11 Blockchain immutability attributes

Txhash	Blockno	Timestamp	DateTime
0x4f9e909e96416d9d4e2cd18460941df2bd54a0c46dd8eebb8a792301e8aed80	5070574	1550842150	2/26/2019 13:29
0xa4df7c7bd246a9e5de7bfd6bac29e9a5cadffa3a6da2dd8ac4a22faeced772ca	5102742	1551259083	2/27/2019 9:18
0x8f2a22c99ab613f27201062f08c8251ba2669a3dd5dd83e5119bd7dd41559479	5137983	1551718836	2/29/2019 5:00
0x315d74a791457efec5c9a176a30e983f4f3d15176064de335648a806f0f03316	5234443	1552992765	3/4/2019 17:00
0xdd787bd3a83f166289fcd3b4b91b8e02e9fa2e0d2895b36e5b93336ace5b02a1	5235275	1553003197	3/7/2019 13:46
0xa799b13f9150e48c8bf8c85e09c0eed0a72c85ee0b8243865ed1d1dcefbd5217	5235278	1553003214	3/9/2019 13:46
0x4e4b945584bb863b3a648b70e007130db4f214294f72fe2f8d39322f94c979b2	5235449	1553005518	3/10/2019 14:25
0x693fe0148757bdc6977d728a8711b8ec8814fc6e9593a343681c55a67bf69ac6	5710843	1559395986	3/11/2019 13:33
0xda1da60d42cc73963a9d4d251e2a798fb707c7ca1e61eb93f622c051894e1e23	5716807	1559478661	6/12/2019 12:31
0xec2b91a3e98bb1b313f26bbc41b0fa53d8ad23995778c70ccfc00d7690316c28	5730858	1563214002	6/12/2019 17:31

c) Decentralization

It can be confirmed that the housing transactions are recorded in a distributed ledger thus improving Transparency. Public verifiability allows anyone to verify correct system working. Everything is archived and authorized in a decentralized way ensuring the integrity and reliable record of transactions as shown in the figure below. These records can be used by various parties (government, investors, KBS, KRA, etc).



Transactions

Tx 0x54b04abe0f2... 27 secs ago From 0xa92a9ce37261b9a... To 0x0cddebbb1efa31c... 0.016 Eth

Tx 0x8829eaE46b... 27 secs ago From 0x934e15b81b91403... To 0x62f855e2ea3521c... 0.003 Eth

Tx 0x725b2289a6... 27 secs ago From 0xcac76b259ad16a2... To 0xfb0aa755dbce5a8... 0.01601 Eth

Tx 0xad698d9e60... 27 secs ago From 0xb70e8bb2384d24f... To 0xfd8971d5e8e1740... 0 Eth

Tx 0x7a0d7958dd... 27 secs ago From 0xb3699cb9c96a282... To 0xcfae9f48222215b... 4.18213 Eth

Tx 0x4E7588712e... 27 secs ago From 0x9344739df26b826... To 0x903b76836129816... 0 Eth

View All Transactions

Figure 33 Decentralization of transaction records

d) Potential To Reduce Overall Rental Fee

Blockchain smart contract models are trustless and the traditional middlemen are replaced by auto-executable contract code. This reduces both transaction complexity and the overall rental fee.

e) Presence Intermediaries

As shown in table 10 below, the developed model allowed direct payment of rental fee from the tenant's account to landlords. Compared to present management strategies, this model promotes disintermediation, therefore, reducing transaction costs. The traditional middlemen are replaced by auto-executable contract code.

Table 10 the model promoting disintermediation

From	To	Quantity
0x236ad9817ae3d957532e279469eb86f1dd673d8	0x13e7b6833e25e19b9d19ba3153bb95c173afd	2500
0x08e743af812a37751c1cfc0fd61f6b997eec0c6c	0x13e7b6833e25e19b9d19ba3153bb95c173afd	2500
0x64404296855fef130c1f215a3dbf9d940b332607	0x13e7b6833e25e19b9d19ba3153bb95c173afd	2500
0x3f4eb593904495a346df4bc7c80a55a519aa4295	0x13e7b6833e25e19b9d19ba3153bb95c173afd	2500
0x3f4eb593904495a346df4bc7c80a55a519aa4295	0x13e7b6833e25e19b9d19ba3153bb95c173afd	2500
0x08e743af812a37751c1cfc0fd61f6b997eec0c6c	0x13e7b6833e25e19b9d19ba3153bb95c173afd	2500
0xcc2acae48f60a324c8ae0fa2c991b8e741d72652	0x13e7b6833e25e19b9d19ba3153bb95c173afd	2500
0x13e7b6833e25e19d19ba3153bb95c173afdd34	0x13e7b6833e25e19b9d19ba3153bb95c173afd	2500
0xbb84b6d71c6c662609bc1b971c9a1d9b2eb5000	0x13e7b6833e25e19b9d19ba3153bb95c173afd	2500
0x861b8b7c649d9717673f53dd8f02fc43c6fee09d	0x13e7b6833e25e19b9d19ba3153bb95c173afd	2500
0xfe96912711c1fc37168df6e84e0290339f3b2141	0x13e7b6833e25e19b9d19ba3153bb95c173afd	2500
0x69670b0c1b10073812415dd474804bb32b3aeca	0x13e7b6833e25e19b9d19ba3153bb95c173afd	2500
0x69670b0c1b10073912415dd474804bb32b3aeca	0x13e7b6833e25e19b9d19ba3153bb95c173afd	2400
0xfe96912711c1fc37168df6e84e0290339f3b2141	0x13e7b6833e25e19b9d19ba3153bb95c173afd	2200
0x69670b0c1b10073912415dd474804bb32b3aeca	0x13e7b6833e25e19b9d19ba3153bb95c173afd	2100

f) Fractions in Claiming Deposits

The model allows Payment of regular rent fee from tenant to landlords account and the rental deposit is paid to smart contract account where it is stored until the end of the contract period. This will reduce cases of delays, frictions, and evasion of refunding deposits.

g) Conflicts Over Late Payment

The developed model allows Immutable and Autonomous payment of a rental fee in a transparent, conflict-free way while avoiding the services of a middleman. The designed smart contract is able to transfer ETokens from tenant account to landlord account and print the transaction records to the blockchain Ethereum network.

h) Possibility To Enforce Lease Agreement

Before a successful creation of an account through registration, a new Tenant is required to sign and agree to lease terms and conditions. Therefore, allowing enforcement of the Lease Agreement and avoids cases of Unexplained Rent Increments.

4.4.2 Validation to Ascertain How the Model Fulfills the Evaluation Criteria Based On User Stories

In this section, the artifact presented in section 4.3 is evaluated using a descriptive evaluation. Additionally, manual, functional testing was carried out on the system using the online compiler provided by the Ethereum foundation (https://ethereum.github.io/browser-solidity/). Development of the smart contracts was done in solidity.

a) **Description of evaluation criteria**

An IT artifact can be evaluated according to the criteria: "functionality, completeness, consistency, accuracy, performance, reliability, usability, fit with the organization, and other relevant quality attributes." (Hevner *et al*, 2004). However, given the novelty of blockchain technology, the scope of the thesis and the extension of the PoC, the artifact cannot be evaluated according to all criteria. Hevner and others (2004), argues that descriptive evaluation can be used only in the case where the technology is especially innovative and other methods may not be feasible. The real estate property management model fits those criteria and will, therefore, be evaluated using two methods from the descriptive evaluation theory as mentioned in section 3.3.1.

b) **Fulfillment of evaluation criteria**

The real estate property management PoC fulfills all the functional criteria shown in Table 3 in section 4.2.1. A motivation for how each of the user stories is satisfied is exposed in Table 12. For details on the code, the reader is directed to the appendix I where the full source code can be found. Besides the functional criteria, additional requirements were defined to cover non-functional aspects of the PoC. These are evaluated in argumentation based on the theory seen in Chapter 2 of this thesis. This section is subdivided based on the requirements deduced in section 4.2.2. It starts with analyzing the potential security and privacy exploits as a way of evaluating the model requirements

i) **R1: It must be impossible for a non-registered person to join the model network.**

The most important non-functional requirement on the PoC is the security of the transaction data as stated in requirement R1. The details of how this is designed and some reasoning regarding in the PoC is found in Section 4.2.5. However, something which is not covered by the solution proposed is the publicity of the transaction records. Hypothetically, should the model be implemented for only a small amount of people, and assuming an attacker could know who those persons were, it could be possible to match blockchain address with a physical identity. For example, it is not terribly difficult to, based on demographics, transaction statistics and some social hacking to find out where someone is leaving, room number or how much he/she is paying per payment interval. On the other hand, if there are a very large amount of Tenants, the data is also valuable. Perhaps not as valuable for malicious attackers as for data scientists, real estate companies or insurances, data is the new gold and should perhaps not be given away so easily. But it still does not violate requirement R1.

Another consideration that needs to be made is that of what takes place before logging on to the blockchain client. Should the IP address of a user be traceable to an account address, then all privacy claims would be flawed. Therefore, great care must be taken when building the surrounding infrastructure, as it cannot be assumed that a large number of people will use Tor. 1For more details on secure(ish) browsing, see https://torproject.org/

Table 12 Evaluation of User Story Acceptance Based On Real Estate Property Management Poc Source: (Researcher, 2019)

User	Traceability	Motivation for fulfillment
Admin	1.1	Fulfilled by the infrastructure of the blockchain and permission levels, both in the component layer and in the contract layer. (See figure 21 in section 4.3.1)
	1.2	Fulfilled by the privileges granted (setup and maintaining account and Maintaining system)
Landlord	2.1	Fulfilled by the infrastructure of the blockchain and permission levels, both in the component layer and in the contract layer. (See figure 18-19 in section 4.3.1)
	2.2	Fulfilled by the infrastructure of the blockchain that allows distributed ledger.
	2.3	Fulfilled by the functionality that the landlord is able to see how much rent has collected by calling checkBall() function.
	2.4	Through sysadmin, rent level can be Set and Adjusted to attract tenants to the property.
	2.5	The rental fee collected in the smart contract account can be transferred to Landlords (personal account) by calling SendFromTo() function.
	2.6	Fulfilled by the infrastructure of the blockchain and permission levels, both in the component layer and in the contract layer.
Tenant	3.1	A new Tenant is able to create an account through registration and the system auto-generates the Tenants address.
	3.2	Fulfilled by the infrastructure of the blockchain and permission levels, both in the component layer and in the contract layer.
	3.3	The Tenant can see and update their personal information upon registration by calling getInfo() and setInfo() functions respectively.
	3.4	The Tenant pays the rental fee and other bills eg gas, electricity and telephone if this was agreed with the landlord using tokens by calling SendFromTo() function.
	3.5	The Tenant can check his/her account balance (ether wallet or token wallet) by calling getbal() function and also by login into metamask Chrome browser extension.
	3.6	Fulfilled by the blockchain architecture

ii) **R2. Only those permitted to should be allowed to connect to the network.**

There are at least two layers of permissions in the PoC, because it is a permission blockchain, and because the smart contracts contain logic which is independent from the blockchain layer. One could also imagine the application of any number of other security measures a user would have to go through to be granted access and permissions on the blockchain. Thereby the requirements R2 and R1 are satisfied.

iii) **R3. There must be immutable traceability built into the model, where it is possible to See all rental fee transactions captured in the blockchain, check for account balance before and after the transaction and be able to see old transactions.**

Because the implementation used blockchain, and as long as more than two-thirds of the validators on that blockchain are benevolent, the record of the events stated in the requirements; transaction process, registration, signing contracts, can be considered very safely and immutably stored. Additional explicit logging of events can be built into the smart contracts, perhaps triggering a completely separate system to create redundancy in the keeping of records. This satisfies requirement R3.

iv) **R4. Smart contracts must be exchangeable without the need to remove the entire model or change addresses to contracts with which humans interact.**

The smart contracts are built in a modular fashion and with a specific contract managing contract. This means that any updates to be made to the system of smart contracts simply need to make a function call to the CMC. The users will not experience any changes, or the applications communicating with the blockchain application will not experience any changes since the address of the interface contract, the InfoManager, remains the same.

4.5 Conclusion

Given the considerations taken to functionality, completeness, and performance in terms of security, Immutability and privacy the artifact and the reasoning behind it is a strong argument for the usage of such an application in the real estate industry.

CHAPTER FIVE

CONCLUSIONS AND RECOMMENDATIONS

5.0 Introduction

In this chapter, a proof-of-concept model was built to function as an electronic blockchain based real estate management plan in a completely decentralized way. In order to achieve a peer-to-peer network secure enough to store transaction records and personal information, a system of smart contracts was developed using a Solidity programming language in combination with public blockchain architecture and common cryptographic tools. The resulting artifact was then evaluated according to established design research criteria and found to fulfill all the necessary requirements. Although the evaluation criteria were fulfilled, it is important to take notice that no claims on the security outside of the PoC can be made, based on this thesis as indicated in the scope of the study. Below are the conclusion, further areas of study and recommendations of this study.

5.1 Summary

This study was guided by the inefficiencies and weaknesses in the current management strategies in the real estate industry. Blockchain technology is explored in this research as a remedy. Due to its architecture, each rental transaction records are bundles into a block containing a unique header, and each such block is identified by its block Txhash (transaction hash) and other information /blockchain parameters as discussed in chapter 2. As a part of a standard mining exercise, a block header is hashed repeatedly by miners by altering the nonce value. Through this exercise, they attempt to create a proof of work, which helps miners get rewarded for their contributions to keep the blockchain system running and ensuring consistency state. It can be confirmed that the housing transactions are recorded in a distributed ledger thus improving Transparency. Public verifiability allows anyone to verify correct system working. Everything is archived and authorized in a decentralized way ensuring the integrity and reliable record of transactions. These records can be used by various parties (government, investors, KBS, KRA, etc). Payment of rent fee from tenant direct to landlord’s account also reduces transaction costs confirming that this model satisfies the user requirements and the research objectives.

This study set out to achieve four objectives and have all been answered by the concepts summarized in the thesis, as well as the architecture of the application and the reasoning around it. The conclusion with regard to each of the objectives is presented in this section and they are shortly summarized below;

5.1.1 To Explore The Weakness Of Existing Models Used In Real Estate Property Management.

With regard to this objective, the study sought to explore existing models used in managing real estate property and investigate their weaknesses. This study analyzed four models including E-Resident, SPENN, Deed-coin, and the Bit-property. To Achieve this, The study set out an evaluation criteria as shown in Table 4 which included; transaction transparency and immutability, decentralization (distributed ledger), presence intermediaries, implementation of rights and obligations, potential to reduce overall rental fee, fractions in claiming deposits conflicts over late payment, possibility of unexplained rent increments and possibility to enforce lease agreement. Based on the evaluation, this research found that none of the existing models fully satisfied the set criteria and there was a need for a new model. The E-Resident platform in Kenya is a web-based platform and does not allow transaction transparency, immutability or decentralization (distributed ledger). Its operation also encourages the need for trusted intermediaries such as the banks and agents thus increasing the rental fee. Furthermore, the model does not prevent the cases of unexplained rent increment or allow enforcement of lease agreement, rights, and obligations. The Bit-property smart contract platform does not address issues surrounding renting a property. SPENN platform does not address property management requirements such agreements, to authorize occupants' rights or obligations while DEED'S platform focuses on reducing challenges in finding real estate agent and the decision on the best commission and anything to do estate property management is not covered.

5.1.2 To Design Blockchain Based Smart Contract Model for Real Estate Property Management.

With regard to objective two, the study sought to design a blockchain based smart contract model for the real estate industry. A design based on the model requirements and guidelines specified in the focused groups were implemented through technological methods. To write a secure and scalable smart contract back-end, the designed model as a system of smart contracts was based on the design principle of having different types of contracts to perform different classes of tasks. To classify the contracts, a model called "The Five Types Model" proposed by monax was used. This model divides contracts into Database contracts, Controller contracts, Contract managing contracts, Application logic contracts and Utility contracts (Monax, 2018). Based on this, the model and the PoC functional requirement, the model design resulted in the following contracts; EToken contract for creating model test cryptocurrency, Interface contract to for Dapp interaction, Migrations contract for deployment, TenantInfoDb contract for storing tenant details, Contract managing Contracts, and the contractProvider contracts to coordinate testnet interactions. The design outlined is limited to three archetypical users: Admin, Tenant, and Landlord. In order to describe the various functional requirements that users have on the model, user stories from a focused group were written and are shown in Table 6 in section 4.2.1. The user stories were simplified to the bare minimum requirements, while still keeping the PoC at a viable level of usability and security. The Model Starting-Up Activities was also presented as shown in section 4.2.3.

5.1.3 To Develop The Blockchain Based Smart Contract Prototype For Real Estate Property Management.

Business firms and other organizations rely on information systems to carry out and manage their operations, interact with their customers and suppliers, and compete in the marketplace. With regard to this study, objective three sought to implement a blockchain based smart contract prototype for the real estate industry to ensure transparency, the immutability of transaction records and troubleshoot existing challenges in managing real estate property. The developed User Registration and Authentication module were to ensure that model Participants can sign up and log in in a convenient. A rental payment Module was developed to allow payment of rental fee via cryptocurrency and to distribute transaction records through the Ethereum network. The solidity code implemented to allow rental fee payment is shown in listing 3 in section 4.3.2. To ensure convenience and efficient working of this model, a complementary module namely balance inquiry was also developed to allow the review of the user account. For instance, if a landlord would wish to check if he/she has received rental payment or a tenant checking whether a transaction was completed successfully. The developed model was deployed to an Ethereum Test Network (“testnet”), which simulates Ethereum for model testing and evaluation. The deployment steps and procedures are described in section 4.4.4. The entire code of the developed model is presented in Appendix I.

5.1.4 To Validate and Verify the Implemented Smart Contract Model Prototype

With regard to objective four, the study sought to validated and verify the implemented blockchain based smart contract model prototype. The prototype presented in objective three was evaluated in two steps which was to, Verification to ascertain how the model overcomes the weaknesses of the existing models and validation to ascertain how the model fulfills the evaluation criteria based on user stories. Two housing agents were involved in the model Verification were they were allowed to interact with the model for real estate management as shown in section 4.4.1. The tenants were able to register on the housing market platform. They interacted with the model such that tenants bought ethers and E-Tokens from their agents and paid the rental fee via the model. Transaction records recorded in the distributed ledger as shown in table 8-9 and the model working was evaluated. The model was verified based on the set evaluation criteria for the existing models evaluation which included; transaction transparency and immutability, decentralization (distributed ledger), presence intermediaries, implementation of rights and obligations, potential to reduce overall rental fee, fractions in claiming deposits conflicts over late payment, possibility of unexplained rent increments and possibility to enforce lease agreement as shown in the Key of Assessment Criteria in section 4.1.1 table 4.

The model was validated using descriptive evaluation based on the model functional requirements derived from user stories as shown in table 6 of section 4.2.1. The model fulfillment of validation criteria is presented in section 4.4.2 where the motivation for how each of the user stories is satisfied is exposed in Table 12. For details on the code, the reader is directed to the appendix I where the full source code can be found. Besides the functional criteria, additional requirements were defined to cover non-functional aspects of the PoC. These were evaluated in argumentation based on the theory seen in Chapter 2 of this thesis.

5.2 conclusions

The objective of this study was to solve a very specific problem in the real estate management sector, however, one cannot fail to realize the potential cross-over effects it could have on other industries. It is essentially a model which lets users register information, allow cryptocurrency transaction securely and then in a highly controlled manner share it with distinctive partners on a distributed network. Consider a group of banks and perhaps even other regulated entities who are highly dependent on information about their customers and transactions. This is basically every company or organization which provides financial service, or an internet provider, or mobile phone operator. All of them need to know personal information on their customers or their historical transactions, by law. Now let us apply the concept of the PoC to this situation. A customer registers or carries and encrypts information on the blockchain using slightly modified smart contracts. When a bank or other organization needs to have access to that information, they simply make a request to the customer who can reject or accept the request. What makes it different from a normal centralized server is that there is immutable traceability, and more importantly, customers could have the power and right to question the usage of personal information or transaction history by corporations. Selling of information would have to be consensual on a much clearer level than it is in most systems today. Another area of application could be in a large corporate setting, where information sharing or transaction history is essential but very difficult when dealing with secret or sensitive data.

5.3 Challenges Experienced During the Study

i) Changes and updates: frequent update of solidity environment, dependencies, and development tools without backward compatibility was also a big challenge.

ii) The Criminal Connection**:** Since its launch, Bitcoin has long been associated with the shadowy dealings of the black market and the dark web. Because this is the first interaction of the public with blockchain technology, this connection has persisted with Bitcoin, altcoins, and the technology underlying it as well. During this study, the users were associating the proposed model to these issues.

5.4 Recommendations

The Government should move towards creating a legal framework for Ethereum and other digital currencies. Although many governments as shown in the literature are now considering launching their own Bitcoin-like cryptocurrency, the process should include sensitizing citizens and financial organizations. This will reduce the resistance of the real estate management model based on blockchain smart contracts among potential users.

5.5 Recommendation for Further Research

Blockchain technology is constantly evolving, both in the private and public sector. Much progress was made during the six months in which this thesis was written, and several research community and committee has begun to form. The introduction of blockchain technology-related topics at research conferences and the launch of blockchain focused peer-reviewed research papers are critical efforts in encouraging further research. There is no doubt that much more technological development is needed, both in academia and innovation coming from the industry. The benefits and problems it solves need to be communicated and an understanding for blockchain outside of crypto-currencies needs to be spread. In the context of this thesis, there are areas of interest which could be further developed. For example, introducing functionality for automatically controlling transaction records against a database of known persons or unintended interactions or creating a completely blockchain-based version of electronic real estate transaction records, including land registry, real estate ownership transfers, etc. There is also a large potential for fundamental research within distributed computing based on blockchain technology, specifically consensus algorithms and lightweight protocols for Internet-of-Things applications. With regards to the emerging decentralized economic system that is cryptocurrencies, there are many important questions related to game theory. Proof-of-Work solves the consensus problem at an increasingly high computational cost in the Bitcoin system, it would be a very important finding should that problem be solved in a provably secure manner, without the energy cost and the drawbacks of Proof-of-Stake.

REFERENCES

Aalbers, M. B. (2016). *The financialization of home and the mortgage market crisis. In The Financialization of Housing* (pp. 40-63).Routledge.

Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J. (2017). *Agile software development methods: Review and analysis.* arXiv preprint arXiv:1709.08439.

ACM Digital Library. (2006). *ACM Transactions on Programming Languages and Systems* (Vol. 28). Association for Computing Machinery.

Ahmad, A. (2017). *Integration of IoT devices via a blockchain-based decentralized application* (Master's thesis).

Aitzhan, N. Z., & Svetinovic, D. (2018). Security and privacy in decentralized energy trading through multi-signatures, blockchain and anonymous messaging streams. *IEEE Transactions on Dependable and Secure Computing,* 15(5), 840-852.

Alharby, M., & van Moorsel, A. (2017). Blockchain-based smart contracts: *A systematic mapping study*. arXiv preprint arXiv:1710.06372.

Ali, M. (2017). *Trust-to-trust design of a new Interne*t (Doctoral dissertation, Princeton University).

Alles, M., Brennan, G., Kogan, A., & Vasarhelyi, M. A. (2018). Continuous monitoring of business process controls: A pilot implementation of a continuous auditing system at Siemens. In Continuous Auditing: *Theory and Application* (pp. 219-246). Emerald Publishing Limited.

Alonso, W. (2017).A theory of the urban land market. *Paper delivered at the*.

Anderson, G., & Arsenault, N. (2005).Fundamentals of educational research.Routledge.

Androulaki, E., Cachin, C., De Caro, A., Kind, A., & Osborne, M. (2017, January).Cryptography and protocols in hyperledger fabric. *In Real-World Cryptography Conference.*

Ang, J. S. (2018). Toward a Corporate Finance Theory for the Entrepreneurial Firm. *FSU College of Law, Public Law Research Paper*, (872).

Arruñada, B. (2018). Blockchain's struggle to deliver impersonal exchange.*Minn. JL Sci. & Tech*., 19, 55.

Ansar, A., Flyvbjerg, B., Budzier, A., & Lunn, D. (2016). Does infrastructure investment lead to economic growth or economic fragility? *Evidence from China. Oxford Review of Economic Policy*, 32(3), 360-390.

Atzei, N., Bartoletti, M., & Cimoli, T. (2017, April). A survey of attacks on ethereum smart contracts (sok). In *International Conference on Principles of Security and Trust* (pp. 164-186). Springer, Berlin, Heidelberg.

Atzori, M. (2015). Blockchain technology and decentralized governance: Is the state still necessary?. *Available at SSRN 2709713*.

Bach, L. M., Mihaljevic, B., & Zagar, M. (2018, May). Comparative analysis of blockchain consensus algorithms. In *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)* (pp. 1545-1550). IEEE.

Balchin, P., & Rhoden, M. (2019). *Housing policy: an introduction*. Routledge.

Bandiera, O. (2007). Contract duration and investment incentives: evidence from land tenancy agreements. *Journal of the European Economic Association*, 5(5), 953-986.

Barber, S., Boyen, X., Shi, E., & Uzun, E. (2012, February). Bitter to better—how to make bitcoin a better currency. In *International Conference on Financial Cryptography and Data Security* (pp. 399-414). Springer, Berlin, Heidelberg.

Barkham, R., Bokhari, S., & Saiz, A. (2018). Urban Big Data: City Management and Real Estate Markets. *GovLab Digest: New York, NY, USA.*

Bartoletti, M., & Pompianu, L. (2017, April). An empirical analysis of smart contracts: platforms, applications, and design patterns. In *International Conference on Financial Cryptography and Data Security* (pp. 494-509).Springer, Cham.

Beck, R., Czepluch, J. S., Lollike, N., & Malone, S. (2016, May).Blockchain-the Gateway to Trust-Free Cryptographic Transactions.In *ECIS* (p. ResearchPaper153).

Been, V. (2017). What More Do We Need to Know About How to Prevent and Mitigate Displacement of Low-and Moderate-Income Households from Gentrifying Neighborhoods.

Belle, I. (2017). The architecture, engineering and construction industry and blockchain technology. *Digital Culture*, *2017*, 279-284.

Bergstra, J. A., & Burgess, M. (2018). Blockchain Technology and its Applications *A Promise Theory* view-V0. 11.

bizimungu, j. (2018, june 5). First blockchain based platform unveiled in Rwanda .Retrieved from the newTimes: https://www.newtimes.co.rw/business/first-blockchain-based-platform-unveiled-rwand.

Bitproperty. (2017, june 1). *Retrieved from The Bitproperty:* https://btptoken.io/en/

bizimungu, j. (2018, june 5). First blockchain based platform unveiled in Rwanda . Retrieved from the newTimes: https://www.newtimes.co.rw/business/first-blockchain-based-platform-unveiled-rwanda

Brown, S. (2016). Visualise, document and explore your software architecture – software architecture for developers. Leanpub. Retrieved from https ://leanpub .com/visualising-software-architecture

Cachin, C., & Vukolić, M. (2017). Blockchains consensus protocols in the wild. *arXiv preprint arXiv:1707.01873.*

Calvaresi, D., Appoggetti, K., Lustrissimi, L., Marinoni, M., Sernani, P., Dragoni, A. F., & Schumacher, M. (2018). Multi-Agent Systems' Negotiation Protocols for Cyber-Physical Systems: Results from a Systematic Literature Review. In *ICAART* (1) (pp. 224-235).

Catalini, C., & Gans, J. S. (2016). *Some simple economics of the blockchain* (No. w22952).National Bureau of Economic Research.

CB Insights. (2017, December 17). what is Ethereum. Retrieved from CBINSIGHTS: https://www.cbinsights.com/research/what-is-ethereum/.

Cedarbaum, J. M. (2018). Elephants, Parkinson's Disease, and Proof- of- Concept Clinical Trials. *Movement Disorders*, *33*(5), 697-700.

Chalaemwongwan, N., & Kurutach, W. (2018, January). State of the art and challenges facing consensus protocols on blockchain. In *2018 International Conference on Information Networking (ICOIN)* (pp. 957-962). IEEE.

Chapman, W. (2018).*Engineering modeling and design*.Routledge.

Chepurnoy, A., Larangeira, M., & Ojiganov, A. (2016). A prunable blockchain consensus protocol based on non-interactive proofs of past states retrievability. *arXiv preprint. arXiv preprint arXiv:1603.07926.*

Chenard, J. R., Manasse, P. E., Davis III, P. R., & Taylor, B. L. (2018). U.S. Patent Application No. 15/843,478.

Chiu, D. K. W. (2012). Web 2.0 environmental scanning and adaptive decision support for business mergers and acquisitions. MIS Quarterly, 36 (4), 1239-1268.

Christidis, K., & Devetsikiotis, M. (2016).Blockchains and smart contracts for the internet of things. Ieee Access, 4, 2292-2303.

Chordia, T., & Miao, B. (2018). Market Efficiency in Real Time: *Evidence from Low Latency Activity around Earnings Announcements.*

Cohen, L. R., Samuelson, L., & Katz, H. (2017). How securitization can benefit from blockchain technology. Journal of Structured Finance, 23(2), 51.

Cooke, H., Appel-Meulenbroek, R., & Arentze, T. (2019). Is the much discussed agility of corporate real estate visible in practice? An empirical study of the relationship between business metrics and surplus property. International Journal of Strategic Property Management, 23(4), 227-243.

Corluka, D., & Lindh, U. (2017). Blockchain: a new technology that will transform the real estate market.

Cusumano MA. The bitcoin ecosystem.Communications of the ACM. 2014 Sep 23;57(10):22-4.

Cronie, H., & Nolan, J. (2018).U.S. Patent Application No. 15/819,518.

Crosby, M., Pattanayak, P., Verma, S., & Kalyanaraman, V. (2016). Blockchain technology: Beyond bitcoin. Applied Innovation, 2, 6-10.

Crosby, M., Nachiappan, Pattanayak, P., Verma, S. & Kalyanaraman, V. (2016). BlockChain Technology: Beyond Bitcoin. Applied Innovation Review, 2, 6-19

Crowston, K., Sawyer, S., & Wigand, R. (2015). Social Networks and the Success of Market Intermediaries: Evidence From the US Residential Real Estate Industry. The Information Society, 31(5), 361-378.

Data Flair. (2018, jan 1). Features Of Blockchain. Retrieved from Data Flair: https://data-flair.training/blogs/features-of-blockchain/

Daniels, M. (2018, 8 20).e-Resident To Provide Better Rental Collection and Management For Estate Owners. Retrieved from kenya tech news: https://www.kachwanya.com/2018/08/20/e-resident-to-provide-better-rental-collection-and-management-for-estate-owners/

Davidson, E. J. (2018). Evaluation methodology basics: The nuts and bolts of sound evaluation. Sage Publications.

DDEX. (2018, 12 25). Installing a digital wallet: MetaMask . Retrieved from DDEX: https://support.ddex.io/hc/en-us/articles/115004408534-Installing-a-digital-wallet-MetaMask

Deng, Q., & Ji, S. (2018). A Review of Design Science Research in Information Systems: Concept, Process, Outcome, and Evaluation. Pacific Asia Journal of the Association for Information Systems, 10(1).

Dinh, T. T. A., Liu, R., Zhang, M., Chen, G., Ooi, B. C., & Wang, J. (2018). Untangling blockchain: A data processing view of blockchain systems. IEEE Transactions on Knowledge and Data Engineering, 30(7), 1366-1385.

Dossey, A. (2019).A Beginner's Guide: POC vs. MVP vs. Prototype. Retrieved From https://clearbridgemobile.com/beginners-guide-poc-vs-mvp-vs-prototype/

Egbertsen, W., Hardeman, G., van den Hoven, M., van der Kolk, G., & van Rijsewijk, A. (2016). Replacing paper contracts with Ethereum smart contracts

Epstein, R. A. (2017). The Role of Exit Rights: What the Theory of the Firm Says about the Conduct of Brexit Negotiations. Cardozo L. Rev., 39, 825.

Eskandari, S., Clark, J., Barrera, D., & Stobert, E. (2018).A first look at the usability of bitcoin key management.arXiv preprint arXiv:1802.04351.

Eyal, I., & Sirer, E. G. (2018). Majority is not enough: Bitcoin mining is vulnerable. Communications of the ACM, 61(7), 95-102.

F. Tschorsch and B. Scheuermann, “Bitcoin and beyond: A technical survey on decentralized digital currencies,” IEEE Communications Surveys Tutorials, vol. 18, no. 3, pp. 2084–2123, 2016.

Fahmideh, M., & Beydoun, G. (2018).Reusing empirical knowledge during cloud computing adoption. Journal of Systems and Software, 138, 124-157.

Fahmy, S. F. (2018). Blockchain and its uses.

Fanning, Kurt, and David P. Centers."Blockchain and its coming impact on financial services." Journal of Corporate Accounting & Finance 27.5 (2016): 53-57.

Felson, M. (2017). Rural settlement and land use. Routledge.

Fernàndez-València, R., Caubet, J., & Vila, A. (2018).Cryptography Working Group Introduction to Blockchain Technology.

Finley, S. (2017). Multi-Disciplinary Probabilistic Design with High Fidelity 3-Dimensional Computer Modeling and Simulation (Doctoral dissertation, Clarkson University).

Franke, G., & Krahnen, J. P. (2017).Sme funding without banks?on the interplay of banks and markets (No. 44). White Paper Series.

Freund, A., & Stanko, D. (2018). The Wolf and the Caribou: Coexistence of Decentralized Economies and Competitive Markets. Journal of Risk and Financial Management, 11(2), 26.

Forstater, M. (2017).Beneficial openness?Weighing the costs and benefits of financial transparency.CMI Working Paper.

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).ACM.

Gipp, B., Meuschke, N., & Gernandt, A. (2015).Decentralized trusted timestamping using the crypto currency bitcoin. arXiv preprint arXiv:1502.04015.

Godfrey-Welch, D., Lagrois, R., Law, J., & Anderwald, R. S. (2018).Blockchain in Payment Card Systems.SMU Data Science Review, 1(1), 3.

Gomber, P., Kauffman, R. J., Parker, C., & Weber, B. W. (2018). On the Fintech Revolution: Interpreting the Forces of Innovation, Disruption, and Transformation in Financial Services. Journal of Management Information Systems, 35(1), 220-265.

Goodbrand, P. T., & Hiller, H. H. (2018). Unauthorized secondary suites and renters: a life course perspective. Journal of Housing and the Built Environment, 33(2), 263-279.

Gordon, N. (2014). Flexible pedagogies: Technology-enhanced learning. From the report series Flexible Pedagogies: Preparing for the Future. The Higher Education Academy, January. Online at: http://www.heacademy.ac.uk /resources/detail/flexiblelearning/ flexiblepedagogies/tech_enhanced_learning/main_report (accessed 20 June 2014).

Glaeser, E., Huang, W., Ma, Y., & Shleifer, A. (2017). A real estate boom with Chinese characteristics.Journal of Economic Perspectives, 31(1), 93-116.

Gregor, S., & Hevner, A. R. (2013). Positioning and presenting design science research for maximum impact. MIS quarterly, 37(2).

Grishchenko, I., Maffei, M., & Schneidewind, C. (2018, July).Foundations and Tools for the Static Analysis of Ethereum smart contracts. In International Conference on Computer Aided Verification (pp. 51-78). Springer, Cham.

Grosse, R. (2017). The global financial crisis—Market misconduct and regulation from a behavioral view. Research in International Business and Finance, 41, 387-398.

Guo, Y., & Liang, C. (2016).Blockchain application and outlook in the banking industry. Financial Innovation, 2(1), 24.

Haarmann, S., Batoulis, K., Nikaj, A., & Weske, M. (2018, June).DMN Decision Execution on the Ethereum Blockchain.In International Conference on Advanced Information Systems Engineering (pp. 327-341).Springer, Cham.

Hancock, D. R., & Algozzine, B. (2016). Doing case study research: A practical guide for beginning researchers.Teachers College Press.

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. MIS Quarterly.

Hills, J. (2017). Good Times, Bad Times (revised Edition): The Welfare Myth of Them and Us. Policy Press.

Hjalmarsson, F. P., Hreioarsson, G. K., Hamdaqa, M., & Hjalmtysson, G. (2018, July).Blockchain-Based E-Voting System.In 2018 IEEE 11th International Conference on Cloud Computing (CLOUD) (pp. 983-986).IEEE.

Hooper, M. (2018, February 22). blockchain. Retrieved from IBM: https://www.ibm.com/blogs/blockchain/2018/02/top-five-blockchain-benefits-transforming-your-industry/

Hugo, G. (2017). New forms of urbanization: beyond the urban-rural dichotomy. Routledge.

Hunt, G. D., & Koved, L. (2018). U.S. Patent Application No. 15/632,522.

Hussein, A. F., ArunKumar, N., Ramirez-Gonzalez, G., Abdulhay, E., Tavares, J. M. R., & de Albuquerque, V. H. C. (2018). A Medical Records Managing and Securing Blockchain Based System Supported by a Genetic Algorithm and Discrete Wavelet Transform. Cognitive Systems Research.

inquvent. (2017, 12 21). poc approach. Retrieved from inquvent: http://www.inquvent.com/poc-approach.html

Jacobs, R. (2017). Bitcoin, Ethereum, and Blockchain: Surprising Insights from 200+ Podcast Interviews of Industry Insiders. Speakeasy Marketing, Inc.

Jyoti, K. M., Singh, M. T., & Saharavat, P. (2018).Analysis the Strength of Agile Methodologies in Software Development.

Kasemsap, K. (2018). The role of information system within enterprise architecture and their impact on business performance. In Global Business Expansion: Concepts, Methodologies, Tools, and Applications (pp. 1078-1102). IGI Global.

Klein, A. (2018). Understanding the Ins and Outs of Financial Services and Products is a Daunting and Difficult Task: An Intern’s Reflections of Financial Services and Products over 11 Months.

Khan, N., Lahmadi, A., Francois, J., & State, R. (2018, April). Towards a management plane for smart contracts: Ethereum case study. In NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium.IEEE.

Kisubi, A. T., & Kisubi, A. T. (2012). A Comparative Study Of Us Inner-City And Kenyan Rural And Slum Schools. Contemporary Voices from the Margin: African Educators on African and American Education, 277.

Kokoris-Kogias, E., Jovanovic, P., Gasser, L., Gailly, N., & Ford, B. (2017). OmniLedger: A Secure, Scale-Out, Decentralized Ledger. IACR Cryptology ePrint Archive, 2017, 406.

Kosba, A., Miller, A., Shi, E., Wen, Z., & Papamanthou, C. (2016, May). Hawk: The blockchain model of cryptography and privacy-preserving smart contracts. In 2016 IEEE symposium on security and privacy (SP) (pp. 839-858).IEEE.

Koulu, R. (2016). Blockchains and online dispute resolution: smart contracts as an alternative to enforcement. SCRIPTed, 13, 40.

Kumar, R. (2019). Research methodology: A step-by-step guide for beginners. Sage Publications Limited.

Kuo, T. T., & Ohno-Machado, L. (2018). ModelChain: Decentralized Privacy-Preserving Healthcare Predictive Modeling Framework on Private Blockchain Networks. arXiv preprint arXiv:1802.01746.

Kuznetsov, S. V., Rastova, Y. I., & Sushcheva, N. V. (2018). The role of public companies in creating a platform for economic growth in Saint Petersburg. Baltic Region, 10(1), 37-55.

Lantmäteriet, Telia, Chromaway, Kairos Future. 2016. Framtidens husköp I blockkedjan. https://www.lantmateriet.se [Received 2017-01-25].

Layard, A. (2018). 26. Property and planning law in England: facilitating and countering gentrification. Handbook of Gentrification Studies, 444.

Leung, C. K. Y., & Tse, C. Y. (2017).Flipping in the housing market. Journal of Economic Dynamics and Control, 76, 232-263.

Levy, K. E. (2017). Book-smart, not street-smart: blockchain-based smart contracts and the social workings of law. Engaging Science, Technology, and Society, 3, 1-15.

Li, X., Jiang, P., Chen, T., Luo, X., & Wen, Q. (2017).A survey on the security of blockchain systems. Future Generation Computer Systems.

Lin, I. C., & Liao, T. C. (2017). A Survey of Blockchain Security Issues and Challenges. IJ Network Security, 19(5), 653-659.

Lipton, A. (2018). Blockchains and distributed ledgers in retrospective and perspective. The Journal of Risk Finance, 19(1), 4-25.

López-Pintado, O., García-Bañuelos, L., Dumas, M., Weber, I., & Ponomarev, A. (2018). CATERPILLAR: A Business Process Execution Engine on the Ethereum Blockchain. arXiv preprint arXiv:1808.03517.

López-Pintado, O., García-Bañuelos, L., Dumas, M., & Weber, I. (2017, September). Caterpillar: A blockchain-based business process management system. In Proceedings of the BPM Demo Track and BPM Dissertation Award co-located with 15th International Conference on Business Process Modeling (BPM 2017), Barcelona, Spain.

Lovelock, C., & Patterson, P. (2015).Services marketing.Pearson Australia.

Lundahl, M. (2015).The Haitian Economy (Routledge Revivals): Man, Land and Markets. Routledge.

Luu, L., Chu, D. H., Olickel, H., Saxena, P., & Hobor, A. (2016, October).Making smart contracts smarter.In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (pp. 254-269).ACM.

Mankoff, J., Onafuwa, D., Early, K., Vyas, N., & Kamath, V. (2018, June).Understanding the Needs of Prospective Tenants.In Proceedings of the 1st ACM SIGCAS Conference on Computing and Sustainable Societies (p. 36).ACM.

Mao, D., Hao, Z., Wang, F., & Li, H. (2018). Innovative Blockchain-Based Approach for Sustainable and Credible Environment in Food Trade: A Case Study in Shandong Province, China. Sustainability, 10(9), 3149.

McLaren, T. S., Head, M. M., Yuan, Y. and Chan, Y. E., (2011). A multilevel model for measuring fit between a firm's competitive strategies and information systems capabilities. 35(4), 909-930.

Medina, J., Ahmed, W. W., Kumar, S., Botey, M., Herrero, R., & Staliunas, K. (2018).Spatiotemporal stabilization proof of concept of Broad Area Semiconductor laser sources. arXiv preprint arXiv:1802.06256.

Merriam, S. B., & Grenier, R. S. (Eds.). (2019). Qualitative research in practice: Examples for discussion and analysis. Jossey-Bass.

MetaMask. (2018, 7 6). MetaMask. Retrieved from MetaMask: https://metamask.io/

Metke, A. R., & Ekl, R. L. (2010).Security technology for smart grid networks. IEEE Transactions on Smart Grid, 1(1), 99-107.

Miraz, M. H., & Donald, D. C. (2018).Application of Blockchain in Booking and Registration Systems of Securities Exchanges. arXiv preprint arXiv:1806.09687.

Molina-Jimenez, C., Sfyrakis, I., Solaiman, E., Ng, I., Wong, M. W., Chun, A., & Crowcroft, J. (2018).Implementation of Smart Contracts Using Hybrid Architectures with On-and Off-Blockchain Components.arXiv preprint arXiv:1808.00093.

monax. (2017, 12 16). the five type model. Retrieved from solidity explainer: https://monax.io/learn/smart_contracts/.

Moreno, A. (2018). The Problems of Expanding Landlord-Tenant Law in Minnesota Through Use of Legal Fiction. Mitchell Hamline Law Journal of Public Policy and Practice, 39(1), 5.

Mubaslat, J. S. (2018). Demonstrating the Functionality and Efficacy of Blockchain-based System in Healthcare Using Simulation Tools (Doctoral dissertation, Wright State University).

McCabe, M. J. (2019). Driving Investment in High-Performance Commercial Buildings.In Sustainable Real Estate (pp. 273-311). Palgrave Macmillan, Cham.

Mwangi, P. N. (2014). Factors Influencing Tax Compliance among Small and Medium Enterprises in Nairobi’s Industrial Area, Kenya. Unpublished MA Project. University of Nairobi, Kenya

Nickerson, R.C., Varshney U., Muntermann J. (2013), A method for taxonomy development and its application in information systems, European Journal of Information Systems, 22 (3), 336-359.

Nightingale, P. (2018). The Place of Credit and Coin in the Medieval English Economy.In Enterprise, Money and Credit in England before the Black Death 1285–1349 (pp. 1-26). Palgrave Macmillan, Cham.

NRI, “Survey on blockchain technologies and related services,” Tech.Rep., 2015. [Online]. Available: http://www.meti.go.jp/english/press/2016/pdf/0531 01f.pdf

Noppen, A. V. (2012). The ABC’s of Affordable Housing in Kenya. Nairobi: acumen fund.

Reiss, David. "The role of the Fannie Mae/Freddie Mac Duopoly in the American housing market." Journal of Financial Regulation and Compliance 17.3 (2009): 336-348.

Robles, T., Bordel, B., Alcarria, R., & Sánchez-de-Rivera, D. (2018).Blockchain Technologies for Private Data Management in AmI Environments.In Multidisciplinary Digital Publishing Institute Proceedings (Vol. 2, No. 19, p. 1230).

O'Leary, D. E. (2018). Open Information Enterprise Transactions: Business Intelligence and Wash and Spoof Transactions in Blockchain and Social Commerce. Intelligent Systems in Accounting, Finance and Management, 25(3), 148-158.

Oliveira, T. C., Gakidou, E., Vos, T., Higashi, H., & Murray, C. (2015). A systematic approach to produce robust, comparable and timely cost-effectiveness estimates for a set of interventions: proof of concept in two low-income countries. Annals of Global Health, 1(81), 64-65.

Ozercan, H. I., Ileri, A. M., Ayday, E., & Alkan, C. (2018).Realizing the potential of blockchain technologies in genomics.Genome research, 28(9), 1255-1263.

Pareliussen, J. K., André, C., Bourrousse, H., & Koen, V. (2017).Income, wealth and equal opportunities in Sweden.

Pattillo, M. (2017).Black on the block. In Racial Structure and Radical Politics in the African Diaspora (pp. 21-52). Routledge.

Piazzesi, M. & Schneider, M (2016). Housing and Macroeconomics. Stanford & NBER. Retrieved from https://web.stanford.edu/~piazzesi/housingandmacroeconomics.pdf

Potterton, L., Agirre, J., Ballard, C., Cowtan, K., Dodson, E., Evans, P. R., ...& Lebedev, A. (2018). CCP4i2: the new graphical user interface to the CCP4 program suite. Acta Crystallographica Section D, 74(2), 68-84.

Prat, N., Comyn-Wattiau,I. & Akoka, J. (2016). Artefact evaluation in information systems design-science research – a holistic view. Retrieved from https://cedric.cnam.fr/fichiers/art_3208.pdf

Preston, J. D., Preston, D. A., Vance, T. M., Simpson, B. C., Madakson, P. A., & Rieger, W. R. (2018). U.S. Patent Application No. 15/675,697.

Saadat, M. N., Rahman, S. A. H. S. A., Nassr, R. M., & Zuhiri, M. F. (2019, February). Blockchain based crowdfunding systems in Malaysian Perspective. In Proceedings of the 2019 11th International Conference on Computer and Automation Engineering (pp. 57-61). ACM.

Schmidt, A. U., Kuntze, N., & Kasper, M. (2016).U.S. Patent No. 9,253,588. Washington, DC: U.S. Patent and Trademark Office.

Schumpeter, J. A. (2017). Theory of economic development.Routledge.

Scott, P. (2018). Friends in high places: Government-industry relations in public sector house-building during Britain’s tower block era. Business History, 1-21.

Sheu, C., Chae, B., & Yang, C. L. (2004). National differences and ERP implementation: issues and challenges. Omega, 32(5), 361-371.

Sirr, L., Hegarty, O., & Reynolds, M. M. (2017). Submission to Oireachtas Committee on Housing, Community, Planning and Local Government: Addressing the issue of vacant housing, derelict sites and underused spaces.

Sompolinsky, Y., & Zohar, A. (2018). Bitcoin's underlying incentives. Communications of the ACM, 61(3), 46-53.

Surcouf, A. J. M., Latapie, H. M., Fenoglio, E., & Friel, J. T. (2018).U.S. Patent Application No. 15/222,890.

Stone, Michael. poverty Shelter: New ideas on housing affordability. Temple University Press, 2010.

Storey, D. J. (2016).Understanding the small business sector.Routledge.

Swan, Melanie. Blockchain: Blueprint for a new economy. " O'Reilly Media, Inc.", 2015.

Szabo, I. (2017). The architecture, engineering and construction industry and blockchain technology. Digital Culture, 279-284.

Tahir, M., Li, M., Ayoub, N., Shehzaib, U., & Wagan, A. (2018). A Novel DDoS Floods Detection and Testing Approaches for Network Traffic based on Linux Techniques. International Journal of Advanced Computer Science And Applications, 9(2), 341-357.

Tapscott D. & Tapscott, A. (2016). The Impact of the Blockchain Goes Beyond Financial Services. Retrieved from https://hbr.org/2016/05/the-impact-of-the-blockchain-goes-beyond-financial-services

Tatjana, M. B., Vognild, L. K., Krogstad, T., Borch, N., Østengen, G., Bratvold, A., & Risberg, M. J. (2008).An easy to use and affordable home-based personal eHealth system for chronic disease management based on free open source software.eHealth Beyond the Horizon–Get IT There, 83.

Taylor, D. (2018). An Analysis of Bitcoin and the Proof of Work Protocols Energy Consumption, Growth, Impact and Sustainability.

Thomas, M. (2018).*Blockchain technology: addressing the risk of digital assets exchange* (Doctoral dissertation, Stellenbosch: Stellenbosch University).

Titman, S. (2017). Does Ownership Structure Matter? *European Financial Management*, 23(3), 357-375.

Tran, B., & Tran, H. (2018).U.S. Patent Application No. 10/022,613.

Tschorsch, F., & Scheuermann, B. (2016). Bitcoin and beyond: A technical survey on decentralized digital currencies. *IEEE Communications Surveys & Tutorials*, 18(3), 2084-2123.

Uddin, N. (2018). Assessing urban sustainability of slum settlements in Bangladesh: Evidence from Chittagong city.*Journal of Urban Management*, 7(1), 32-42.

Ullah, F., Sepasgozar, S., & Wang, C. (2018). A Systematic Review of Smart Real Estate Technology: Drivers of, and Barriers to, the Use of Digital Disruptive Technologies and Online Platforms. *Sustainability*, 10(9), 3142.

Van Aken, J. E., & Berends, H. (2018).*Problem solving in organizations*.Cambridge University Press.

Vasarhelyi, M. A., & Halper, F. B. (2018).The continuous audit of online systems. In Continuous Auditing: *Theory and Application* (pp. 87-104). Emerald Publishing Limited.

Vigna, P., & Casey, M. J. (2019). *The Truth Machine: The Blockchain and the Future of Everything*. Picador.

Vitalik Buterin. (2017, Jan. 17th) Ethereum white paper. https:// github.com/ethereum /wiki /wiki /White-Paper.

Warren, W., & Bandeali, A. (2017).0x: An open protocol for decentralized exchange on the Ethereum blockchain. URl: https://github. com/0xProject/whitepaper.

Wiklund, F., & Jansson, S. (2019). Employee Loyalty and the Factors Affecting It: A qualitative study comparing people with different working experience on their view of employee loyalty.

Wong, M. (2018). CMBS and Conflicts of Interest: Evidence from Ownership Changes for Servicers. *The Journal of Finance*, 73(5), 2425-2458.

Wright, A., & De Filippi, P. (2015). Decentralized blockchain technology and the rise of lex cryptographia. *Available at SSRN 2580664*.

Yates, Judith, et al. "Housing affordability: a 21st century problem." *National research venture 3* (2017).

Yee, O. S., Sagadevan, S., & Malim, N. H. A. H. (2018). Credit Card Fraud Detection Using Machine Learning As Data Mining Technique. *Journal of Telecommunication, Electronic and Computer Engineering* (JTEC), 10(1-4), 23-27.

Zhang, F., Cecchetti, E., Croman, K., Juels, A., & Shi, E. (2016, October). Town crier: An authenticated data feed for smart contracts. *In Proceedings of the 2016 aCM sIGSAC conference on computer and communications security* (pp. 270-282).ACM.

Zheng, Z., Xie, S., Dai, H. N., & Wang, H. (2016). Blockchain challenges and opportunities: A survey. *Work Pap.–2016.*

Zheng, Z., Xie, S., Dai, H., Chen, X., & Wang, H. (2017, June). An overview of blockchain technology: *Architecture, consensus, and future trends. In Big Data* (BigData Congress), 2017 IEEE International Congress on (pp. 557-564). IEEE.

Ziemann, V. (2017).Inclusive labour Markets in the digital era.

Zook, M. A., & Blankenship, J. (2018).New spaces of disruption?*The failures of Bitcoin and the rhetorical power of algorithmic governance.* Geoforum, 96, 248-255.

Zyskind, G., & Nathan, O. (2015, May). Decentralizing privacy: Using blockchain to protect personal data. *In Security and Privacy Workshops* (SPW), 2015 IEEE (pp. 180-184). IEEE.

APPENDIX I: SYSTEM SOURCE CODE

Front end

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta http-equiv="X-UA-Compatible" content="ie=edge">
<link rel="shortcut icon" href="#" >
<link rel="stylesheet" href="./css/bootstrap.min.css">
<link rel="stylesheet" href="./css/main.css">
<title>smart </title>
</head>

<body>
<div id="header" >
<br>
<h2> WELCOME TO THIS PLATFORM GOOD TO SEE YOU !!PLEASE LOGIN OR
REGISTERTO CONTINUE ENJOYING THE SERVICES
</h2>
</div>
<div id="homehead" style="display:none" ><h2><button class="btn-default btn-success"
id="homereg">registerHome</button>
<button class="btn-default btn-success" id="roomreg">registerrooms</button>
<button class="btn-default btn-success" id="TransferToken">TransferToken</button>
<button class="btn-default btn-success" id="checkbalance">CheckTokenBalance</button>
<button class="btn-default btn-success" id="rentroom">RentRoom</button>
</h2></div>
<div id="main1" class="container"style="display:none">
<div class="main"  >
<form class="form" onSubmit="App.Registration();return false;" >
```

```html
<h2>Register to the Application Inorder to Access services</h2>
<label>Name :</label>
<input type="text" name="dname" id="name">
<label>ETH Address:</label>
<input type="text" name="dname" id="ethadd">
<label>Email :</label>
<input type="text" name="demail" id="email">
<label>Password :</label>
<input type="password" name="password" id="password">
<label>Confirm Password :</label>
<input type="password" name="cpassword" id="cpassword">
<input type="submit" name="register" id="register" value="Register">
</form><br><br>
<div id="loginclick">
		Already have an account ??<span>click here to login</span>
</div>
</div>
</div>

<div class="container" style="display:none" id="loginmain">
<div class="main">
<form class="form" onSubmit="App.Login(); return false;" >
<h2> Login Form </h2>
<label>Password :</label>
<input type="password" name="password" id="pass">
<input type="submit" name="login" id="login" value="Login">
</form><br><br>
<div id="registerclick" >Do not have an  account click here to register??</div>
</div>
</div>
<div class="container"style="display:none" id="houseregistration">
<div class="main">
<form class="form"onSubmit="App.registerHouse(); return false;">
```

```
<h2>Register your houses</h2>
<label>House Name :</label>
<input type="text" name="Hname" id="hname">
<label> House Location:</label>
<input type="text" name="location" id="location">
<label>Owner Name :</label>
<input type="text" name="owner" id="owner">
<label>Company owning :</label>
<input type="text" name="company" id="company">
<input type="submit" name="register" id="register1" value="Register">
</form>
</div>
</div>
<div class="container"style="display:none">
<div class="main">
<form class="form">
<h2>Register your Rooms</h2>
<label>House Name :</label>
<input type="text" name="Hname" id="Hname">
<label> Room number:</label>
<input type="text" name="rnumber" id="rnumber">
<label>Room facilities :</label>
<input type="text" name="facilities" id="facilities">
<label>Room Name:</label>
<input type="text" name="roomn" id="roomname">
<input type="submit" name="register" id="register2" value="Register">
</form>
</div>
</div>
<div class="container" style="display:none" id="tokensform">
<div class="main">
<form class="form"onSubmit="App.TransferToken(); return false;">
<h2>Transfer Tokens</h2>
```

```
<label>ETH address  :</label>
<input type="text" name="to" id="to">
<label> Amount to Transfer:</label>
<input type="text" name="tokenamount" id="tokenamount">
<input type="submit" name="register" id="register3" value="Register">
</form>
</div>
</div>
<div class="container" style="display:none" id="balanceform">
<div class="main">
<form class="form" onSubmit="App.CheckBalance(); return false;">
<h2> check balance  of  Tokens</h2>
<label>ETH address :</label>
<input type="text" name="accountof" id="accountof">
<label> Balance of this account is:</label>
<h1 id="accountofbalance"></h1>
<input type="submit" name="register" id="register4" value="CHECKBALANCE">
</form>
</div>
</div>

<script src="./js/jquery-3.3.1.min.js"></script>
<script src="./js/bootstrap.min.js"></script>
<script src="./js/web3.js"></script>
<script src="./js/truffle-contract.js"></script>
<script src="./js/app.js"></script>

</body>
</html>
```

EToken smart Contracts

```
pragma solidity ^0.5.0;
```

```
contract EToken {
    string public name = "EToken";
    string public symbol = "ETk";
    uint256 public totalSupply ;

    mapping(address =>uint256)public balanceOf;

    mapping(address=>mapping(address=>uint256))public allowance;

    event Transfer(
        address indexed _from,
        address indexed _to,
        uint256 _value
    );

    event Approval(
        address indexed _spender,
        address indexed _owner,
        uint256 _value
    );

    constructor (uint256 _initialSupply) public {
      balanceOf[msg.sender] = _initialSupply;
      totalSupply = _initialSupply;
    }

    function TotalSupply() public view returns (uint256 totalsupply) {
        return totalSupply ;
    }

    function BalanceOf(address _owner)public view  returns (uint256 balance) {
        return balanceOf[_owner];
    }

    function transfer(address _to,uint256 _value)public  returns(bool success) {
        require(balanceOf[msg.sender]>= _value);
        balanceOf[msg.sender] -= _value;
        balanceOf[_to] += _value;

        emit Transfer(msg.sender,_to,_value);
        return true;
    }

    function approve(address _spender, uint256 _value) public returns(bool success)  {
        require(balanceOf[msg.sender] >= _value);
        allowance[msg.sender][_spender] = _value;

        emit Approval(_spender,msg.sender,_value);
        return true;
    }

    function transferFrom(address _from,address _to,uint256 _value ) public returns (bool
success) {
        require(balanceOf[_from]>=_value);
        require(allowance[_from][msg.sender] >=_value);
        balanceOf[_from] -=_value;
        balanceOf[_to] +=_value;
        allowance[_from][msg.sender] -= _value;

        emit Transfer(_from,_to,_value);

        return true;
    }

}
```

Composite contract for User Registration, Login, Rental payment and CmCcontyracts.

```
pragma solidity ^0.5.0;
import './EToken.sol';
import './Houses.sol';

contract Interface {

    address[] public registeredUsers;
    Houses public HouseContract;
    EToken public EtokenContract;
    uint256 public roomPrice;

    struct User {
        string name;
        string email;
        string password;
        bool  registered;
    }

    mapping(address=>User)public Users;
    constructor (Houses _HouseContract ,EToken _EtokenContract,uint256 _price) public{
        HouseContract = _HouseContract;
        EtokenContract =  _EtokenContract;
        roomPrice = _price;
    }



    function register(address _user, string memory _password,string memory _email, string
memory _name)public returns(bool success) {
        address[] storage registed = registeredUsers;
        User storage user = Users[msg.sender];
        registed.push(msg.sender);
        require(!user.registered);
        require(msg.sender == _user);

        user.password = _password;
        user.name = _name;
        user.email = _email;
        user.registered = true;
        return true;
    }

    function login(string memory _password)public view returns(bool ) {
        User storage user = Users[msg.sender];
        require(user.registered);
         return          (keccak256(abi.encodePacked(user.password))          ==
keccak256(abi.encodePacked(_password)));



     }
     function  rentRoom(bytes32  _housename,uint256  _roomnumber,  uint256  _amount)public
returns(bool success){
         require(_amount>=roomPrice);
         EtokenContract.approve(address(this),_amount);
         EtokenContract.transferFrom(msg.sender,address(this),roomPrice);
         HouseContract.updateRentRoom(_housename,_roomnumber);
         return true;
     }
}
```

Truffle Migration Contracts

```
pragma solidity>=0.4.21 <0.6.0;

contract Migrations {
  address public owner;
  uint public last_completed_migration;

  constructor() public {
    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);
  }
}
```

Rental house management contracts Contracts

```
pragma solidity ^0.5.0;

contract Houses {
    struct House {
        address owneradd;
        string location;
        string owner;
        string company;
        uint256[] rooms;
    }
    bytes32[] public  houseNames;
     mapping(bytes32=>House)public registeredHouses;
     mapping(bytes32=>mapping(uint256=>Rooms)) public registeredRooms;

    struct Rooms {
        bool registered;
        string facilities;
        string name;
        bool booked;
    }

    function registerHouse(bytes32  _housename,string memory _location, string memory
_owner,string memory _company) public returns(bool success) {
            bytes32[] storage housen = houseNames;
            House storage house = registeredHouses[_housename];
            house.owneradd = msg.sender;
            house.location = _location;
            house.owner = _owner;
            house.company = _company;
            housen.push(_housename);
            return true;

    }

    function registerRoom( bytes32  _housename, uint256 _roomnumber,string memory
_facilities,string memory _name)public returns(bool success) {
        Rooms storage room = registeredRooms[_housename][_roomnumber];
        House storage house = registeredHouses[_housename];
        require(house.owneradd == msg.sender);
        room.facilities = _facilities;
        room.name = _name;
        room.registered = true;
        house.rooms.push(_roomnumber);
        return true;
    }
    function updateRentRoom(bytes32  _housename, uint256 _roomnumber)public {
         Rooms storage room = registeredRooms[_housename][_roomnumber];
         room.booked = true;
    }
    function checkOut(bytes32  _housename, uint256 _roomnumber)public {
        Rooms storage room = registeredRooms[_housename][_roomnumber];
         room.booked = false;

    }
// function  registeredhouses()public view returns(bytes32) {
//        for (uint256 i = 1; i < houseNames.length; i++) {
//                return houseNames[i];
//        }
// }
}
```

Smart Contract Tests (Web3 files).

Etoken ontract Testsfile (Web3 files).

```

var EToken = artifacts.require("EToken");


contract ("EToken", function(accounts){
   var Instance ;
```

```
 it("sets state variables ",function(){
    return EToken.deployed().then(function(instance){
        Instance = instance;
      return Instance.name();
    }).then(function(name){
        assert.equal(name ,"EToken","etoken has correct name");
      return Instance.symbol();
    }).then(function(symbol){
        assert.equal(symbol,"ETk","sets token symbol to ETk");

    });
 });
  it("sets total supply and allocates them  to token holder",function(){
     return EToken.deployed().then(function(instance){
         Instance=instance;
         return Instance.TotalSupply();
     }).then(function(deployedamount){
         assert.equal(deployedamount,"1000000","sets initialsupply to a million ");
         return Instance.BalanceOf(accounts[0]);
     }).then(function(tokenholderbalance){
         assert(tokenholderbalance.toNumber(),"1000000","sets the token holders amount to a million ");
     });
  });
  it("returns total supply of tokens ",function(){
      return EToken.deployed().then(function(instance){
          Instance = instance;
          return Instance.TotalSupply();
      }).then(function(totalsupply){
          assert.equal(totalsupply,"1000000","total supply is a million ");
      });
  });
  it("return tokens of a  particular account ", function(){
      return EToken.deployed().then(function(instance){
          Instance = instance;
          return Instance.BalanceOf(accounts[2]);
      }).then(function(balance){
          assert.equal(balance.toNumber(),"0" ,"returns zero balance for account 2 since  no
token has been transfered to it");
      });
  });
   it("should be allow token holders transfer tokens",function(){
       return EToken.deployed().then(function(instance){
            Instance = instance;
            return Instance.transfer(accounts[3],10000000,{from:accounts[2]});
       }).then(assert.fail).catch(function(error){
            assert(error.message.indexOf("revert")>=0,"error message must contain revert if
try to transfer more than what is in account");
            return Instance.transfer(accounts[2],1000,{from:accounts[0]});
       }).then(function(receipts){
             assert.equal(receipts.logs.length, 1,"has got only one log");
             assert.equal(receipts.logs[0].event,"Transfer","trigger transfer event");
             assert.equal(receipts.logs[0].args._value,  1000, " has argument to which token is
being sent to");
             assert.equal(receipts.logs[0].args._from, accounts[0], " has argument to which
token is being sent to");
             assert.equal(receipts.logs[0].args._to, accounts[2], " has argument to which token
is being sent to");
            return Instance.BalanceOf(accounts[2]);
       }).then(function(balance){
            assert.equal(balance.toNumber(),"1000","shows it has added account 2 with the
tokens");
            return Instance.BalanceOf(accounts[0]);
       }).then(function(balance){
            assert.equal(balance.toNumber(),"999000","should have  deducted ammount from
th eacccout");
```

```
         return Instance.transfer.call(accounts[2],1000,{from: accounts[0]});
    }).then(function(success){
         assert.equal(success,true,"returns succeess on transfer");
    });
  });
  it(" sets  spender to withdraw fromm their account",function(){
    return EToken.deployed().then(function(instance){
      Instance = instance;
      return Instance.approve.call(accounts[4],1000,{from:accounts[0]});
    }).then(function(success){
       assert.equal(success,true,  "allowed to withdraw 1000 tokens ");
       return Instance.approve(accounts[3],1000,{from:accounts[0]});
    }).then(function(receipts){
        return Instance.allowance(accounts[0],accounts[3]);
    }).then(function(allowed){
       assert.equal(allowed.toNumber(),1000,"allowed to withdraw 100 tokens ");
       return Instance.approve(accounts[2],10000000,{from:accounts[5]});
    }).then(assert.fail).catch(function(error){
       assert(error.message.indexOf("revert")>=0,"should revert because there is no
enough token in the account");
       return Instance.approve(accounts[3],100000,{from:accounts[0]});
    }).then(function(receipts){
       assert.equal(receipts.logs.length, 1, "has got only one log");
       assert.equal(receipts.logs[0].event, "Approval", " triggers  approval event");
      assert.equal(receipts.logs[0].args._value, 100000, " has argument value");
    });

  });

  it ("should allow delegated withdrawal ",function(){
    return EToken.deployed().then(function(instance){
       Instance = instance;
       return Instance.transfer(accounts[5],500,{from:accounts[0]});
    }).then(function(receipt){
       return Instance.approve(accounts[4],400,{from:accounts[5]});
    }).then(function(receipt){
       return Instance.transferFrom(accounts[5],accounts[3],600,{from:accounts[4]});
    }).then(assert.fail).catch(function(error){
       assert(error.message.indexOf("revert")>=0,"should revert if no enough token in account");
       return Instance.transferFrom(accounts[5],accounts[3],600,{from:accounts[4]});
    }).then(assert.fail).catch(function(error){
       assert(error.message.indexOf("revert")>=0,"should revertt if try to transfer is not enough than allocated");
        return Instance.transferFrom(accounts[5],accounts[3],300,{from:accounts[4]});
    }).then(function(receipts){
       assert.equal(receipts.logs.length, 1, "has got only one log");
       assert.equal(receipts.logs[0].event, "Transfer", " triggers  approval event");
       assert.equal(receipts.logs[0].args._value, 300, " has argument value");
       assert.equal(receipts.logs[0].args._from, accounts[5], " has argument to which token is being sent from");
       assert.equal(receipts.logs[0].args._to, accounts[3], " has argument to which token is being sent to");
       return Instance.balanceOf(accounts[5]);
    }).then(function(balance){
       assert.equal(balance.toNumber(),"200","should have deducted");
       return Instance.BalanceOf(accounts[3]);
    }).then(function(balance){
       assert(balance.toNumber(),"300","should have transfered token");
       return Instance.transferFrom.call(accounts[5],accounts[3],50,{from:accounts[4]});
    }).then(function(success){
       assert.equal(success,true,"should return true on complete transaction");
       return Instance.allowance(accounts[5],accounts[4]);
    }).then(function(allowed){
       assert.equal(allowed.toNumber(),"100","should deduct the amout spender can still deduct");
```

```
        });
    });
});
```

House management contract Test file

```
var Houses = artifacts.require("./Houses.sol");
var Web3 = require('web3');
web3 = new Web3(new Web3.providers.HttpProvider("http://localhost:8545"));

contract ("Houses",function(accounts){
    it("allows users to register homes",function(){
        return Houses.deployed().then( function(instance){
            Instance = instance;
             gateway = web3.fromAscii("gateway");
            return Instance.registerHouse(gateway, "kericho", "collins", "abno", {
                from: accounts[0]
            });
        }).then(function(re){
            return Instance.registeredHouses(gateway);
        }).then(function(house){
            assert.equal(house.company,"abno","assert abno has been registered");
            assert.strictEqual(house.owneradd,accounts[0], "assert abno has been registered");
            return Instance.houseNames(0);
        }).then(function(length){
            assert.equal(length,
0x6761746577617900000000000000000000000000000000000000000000000000,        "1
should be added to arry");
        });

    });
    it("allows users to register rooms",function(){
        return Houses.deployed().then(function(instance){
            instance = Instance;
```

```
             return Instance.registerHouse(gateway, "kericho", "collins", "abno", {
                 from: accounts[0]
             });
        }).then(function(re){
             return Instance.registerRoom(gateway, 1, "self contained", "room one", {
                 from: accounts[0]
             });
        }).then(function(re){
            return Instance.registeredRooms(gateway,1);
        }).then(function(receipts){
            assert.strictEqual(receipts.facilities,"self contained");
            assert.strictEqual(receipts.name, "room one");
             return Instance.registeredHouses(gateway);
        }).then(function(re){
// assert.strictEqual(re.rooms[0],1);
        });


    });
});
```

Composite contract for User registration, authentication and rental payment Test File

```

var Interface = artifacts.require("./Interface.sol");


contract ("Interface",function(accounts){
    it("have address of the other contracts",function(){
        return Interface.deployed().then(function(instance){
               Instance = instance;
            return Instance.HouseContract();
        }).then(function(re){
            assert.notStrictEqual(re,0x0);
            return Instance.EtokenContract();
        }).then(function(re){
              assert.notStrictEqual(re, 0x0);
         });
     });
      it("allows users to register",function(){
           return Interface.deployed().then(function (instance) {
             Instance =  instance;
             return
Instance.register(accounts[1],"shiti","collins@gmail.com","collins",{from:accounts[2]});
           }).then(assert.fail).catch(function(error){
              assert(error.message.indexOf("revert")>=0,"should revert since accounts are not
the same");
              return Instance.register(accounts[1], "shiti", "collins@gmail.com", "collins", {
from: accounts[1] });
           }).then(function(receipts){
              return Instance.Users(accounts[1]);
           }).then(function(receipt){
              assert.equal(receipt.name,"collins","should save the anme collins to accounts 1");
              assert.equal(receipt.email, "collins@gmail.com", "should save the anme collins
to accounts 1");
              assert.equal(receipt.registered, true, "should save the anme collins to accounts
1");
              assert.equal(receipt.password, "shiti", "should save the anme collins to accounts
1");
               return Instance.register(accounts[5], "shiti", "collins@gmail.com", "collins", {
               from: accounts[5]
               });
               }).then(function (receipts) {
               return Instance.registeredUsers(1);
            }).then(function(re){
              assert.equal(re, accounts[5], "should have pushed accounts1");
           });
      });
     it("allows users to login",function(){
           return Interface.deployed().then(function(instance){
              Instance= instance;
              return Instance.register(accounts[1], "shiti", "collins@gmail.com", "collins", {
from: accounts[1] });
           }).then(function(receipt){
              return Instance.login("shiti",{from:accounts[3]});
           }).then(assert.fail).catch(function(error){
              assert(error.message.indexOf("revert")>=0,"shold revert when user is not
registered");
            return Instance.register(accounts[3], "shiti", "collins@gmail.com", "collins", {
               from: accounts[3]
            });
           }).then(function(re){
              return Instance.login("shiti", {
                 from: accounts[3]
              });
           }).then(function(re){
              assert.equal(re,true);
           });
     });
});
```

Truffle Migration Files.

.

```
var EToken =artifacts.require("./EToken.sol");
// var Registration = artifacts.require("./registration.sol");
 var Houses = artifacts.require("./Houses.sol");
 var Interface = artifacts.require("./Interface.sol");

module.exports = function(deployer){
// deployer.deploy(EToken,1000000);
// deployer.deploy(Interface);
// deployer.deploy(Houses);
     deployer.deploy(EToken,1000000).then(function(){
        return deployer.deploy(Houses);
    }).then(function(){
        return deployer.deploy(Interface, Houses.address,EToken.address,20);
    });
};
```

Web3 app file

```
App = {
 web3Provider:"null",
 contracts:{},
 accounts:"0x0",
 loggedin:false,

 init: function() {
      console.log("app.initialized");
      return App.initweb3();
 },

    initweb3: function() {
        if (typeof web3 !== 'undefined') {
            App.web3Provider = web3.currentProvider;
            web3 = new Web3(App.web3Provider);
        } else {
// set the provider you want from Web3.providers
            App.web3Provider = new Web3.providers.HttpProvider("http://localhost:8545");
            web3 = new Web3(App.web3Provider);
        }
        return App.initcontracts();
    },
    initcontracts: function(){
       $.getJSON("EToken.json",
           function (etoken) {
           App.contracts.EToken = TruffleContract(etoken);
           App.contracts.EToken.setProvider(App.web3Provider);
            }).done(function(){
         $.getJSON("Houses.json",
             function (house) {
              App.contracts.Houses = TruffleContract(house);
              App.contracts.Houses.setProvider(App.web3Provider);
              App.contracts.Houses.deployed().then(function(instance){
                console.log(instance.address);
              });
             });
        }).done(function(){
            $.getJSON("interface.json",
                function (interface) {
                    App.contracts.Interface = TruffleContract(interface);
                    App.contracts.Interface.setProvider(App.web3Provider);
                    return App.Render();
                });
        });
    },
    Render:function(){
```

```
    web3.eth.getCoinbase(function(err,res){
        if(err === null){
            App.accounts = res;
        }
    });

},
Registration: function(){
    var Name = $("#name").val();
    var Ethadd = $("#ethadd").val();
    var Email = $("#email").val();
    var Password = $("#password").val();

    App.contracts.Interface.deployed().then(function(instance){
        Instance = instance;
        return Instance.register(Ethadd, Password, Email, Name, {
            from: App.accounts,
            gas: 500000
        });
    }).then(function(receipts){
        alert("successfully registered you can login in now");
        $("#loginmain").show();
        $("#main1").hide();
    });
},
Login:function(){

    var loginpassword = $("#pass").val();
    App.contracts.Interface.deployed().then(function(instance){
        Instance = instance;
        return Instance.login.call(loginpassword,{from:App.accounts,gas:500000});
    }).then(function(receipts){
        console.log(receipts);
        if(receipts == true){
           $("#loginmain").hide();
           $("#header").html("<h2>Real Estate system To Rent Out Homes</h2>");
           $("#homehead").show();
        }else{
            alert("wrong password");
        }
    }
    );
},
registerHouse: function(){
    var Hname = $("#hname").val();
    var Location = $("#location").val();
    var Owner = $("#owner").val();
    var Company = $("#company").val();

    var HOname = web3.fromAscii(Hname);
    console.log(HOname);
    App.contracts.Houses.deployed().then(function(instance){
        Instance = instance;
        return Instance.registerHouse(HOname, Location, Owner, Company, {
            from: App.accounts,
            gas: 500000
        });
    }).then(function(re){
        console.log(re);
    });
},
registerRoom:function(){
    var HName = $("#Hname").val();
    var Rnumber = $("#rnumber").val();
    var Facilities =$("#facilities").val();
    var Room = $("#roomname").val();
    App.contracts.Houses.deployed().then(function(instance){
      Instance = instance;
      return
Instance.registerRoom(HName,Rnumber,Facilities,Room,{from:App.accounts,gas:500000})
;
    }).then(function(re){
      console.log(re);
    });


  },
  TransferToken: function(){
    var to = $("#to").val();
    var tokenamount = $("#tokenamount").val();
    App.contracts.EToken.deployed().then(function(instance){
      Instance = instance;
      return Instance.transfer(to,tokenamount,{from:App.accounts,gas:500000});
    }).then(function(receipts){
      console.log(receipts);
    });
  },
  CheckBalance: function(){
    var accountbal = $("#accountof").val();
    App.contracts.EToken.deployed().then(function(instance){
      Instance = instance;
      return Instance.BalanceOf(accountbal,{from:App.accounts,gas:500000});
    }).then(function(re){
      balance = re.toNumber();
      console.log(balance);
      $("#accountofbalance").html(balance + "tokens");
    });
  }
```

```
};
$(document).ready(function(){
   App.init();
   $("#main1").show();
   $("#loginclick").click(function(){
        $("#loginmain").show();
         $("#main1").hide();
   });
   $("#homereg").click(function(){
        $("#houseregistration").show();
   });
   $("#TransferToken").click(function () {
       $("#tokensform").show();
   });
    $("#checkbalance").click(function () {
        $("#balanceform").show();
    });

});
```

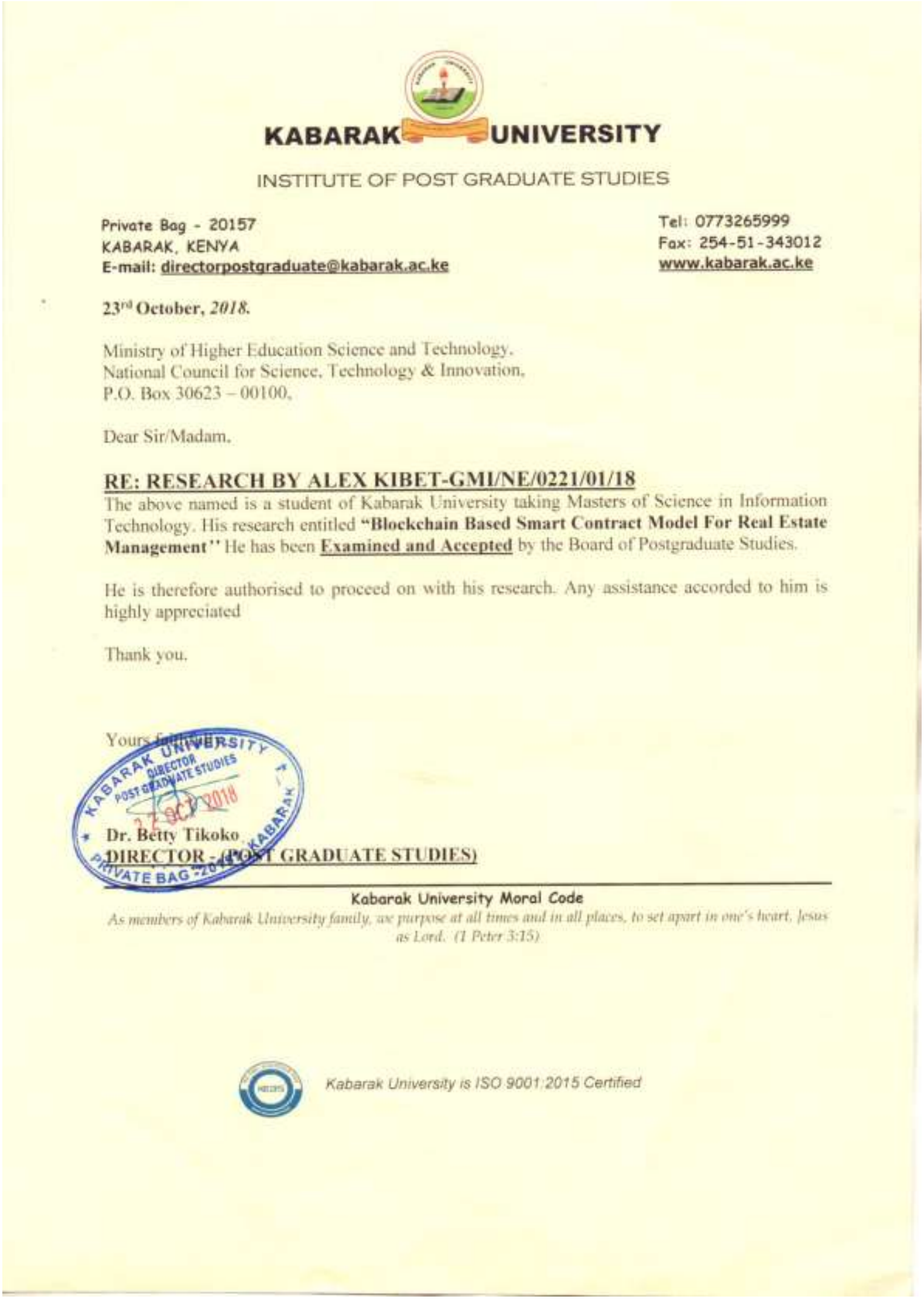
CSS File

```
h2{
text-align: center;
font-size: 90%;
padding: 2%;
background-color: rgb(98, 98, 223);
}
hr{
margin-bottom: 30px;
}
div.container{
```

```
width: 50%;
height: 610px;
margin:50px auto;
font-family: 'Droid Serif', serif;
position:relative;
}
div.main{
width: 75%;
float:left;
padding: 10px 55px 40px;
background-color:rgba(117, 57, 57, 0.219);
border: 15px solid white;
box-shadow: 0 0 10px;
border-radius: 2px;
font-size: 13px;
}
input[type=text],[type=password] {
width: 97.7%;
height: 34px;
padding-left: 5px;
margin-bottom: 20px;
margin-top: 8px;
box-shadow: 0 0 5px #00F5FF;
border: 2px solid #00F5FF;
color: #4f4f4f;
font-size: 16px;
}
label{
color: #464646;
text-shadow: 0 1px 0 #fff;
font-size: 17px;
font-weight: bold;
}
```

```
#register,
#login,
#register1,
#register2,
#register3,
#register4 {
font-size: 20px;
margin-top: 15px;
background: linear-gradient(#22abe9 5%, #36caf0 100%);
border: 1px solid #0F799E;
padding: 7px 35px;
color: white;
text-shadow: 0px 1px 0px #13506D;
font-weight: bold;
border-radius: 2px;
cursor: pointer;
width: 100%;
}
#register:hover,{
background: linear-gradient(#36caf0 5%, #22abe9 100%);
}
```

APPENDIX II: UNIVERSITY RESEARCH AUTHORIZATION



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 ALEX KIBET-GMI/NE/0221/01/18

The above named is a student of Kabarak University taking Masters of Science in Information Technology. His research entitled **"Blockchain Based Smart Contract Model For Real Estate Management''** 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,

KABARAK UNIVERSITY
DIRECTOR
POST GRADUATE STUDIES
22 OCT 2018
PRIVATE BAG - 20157 KABARAK

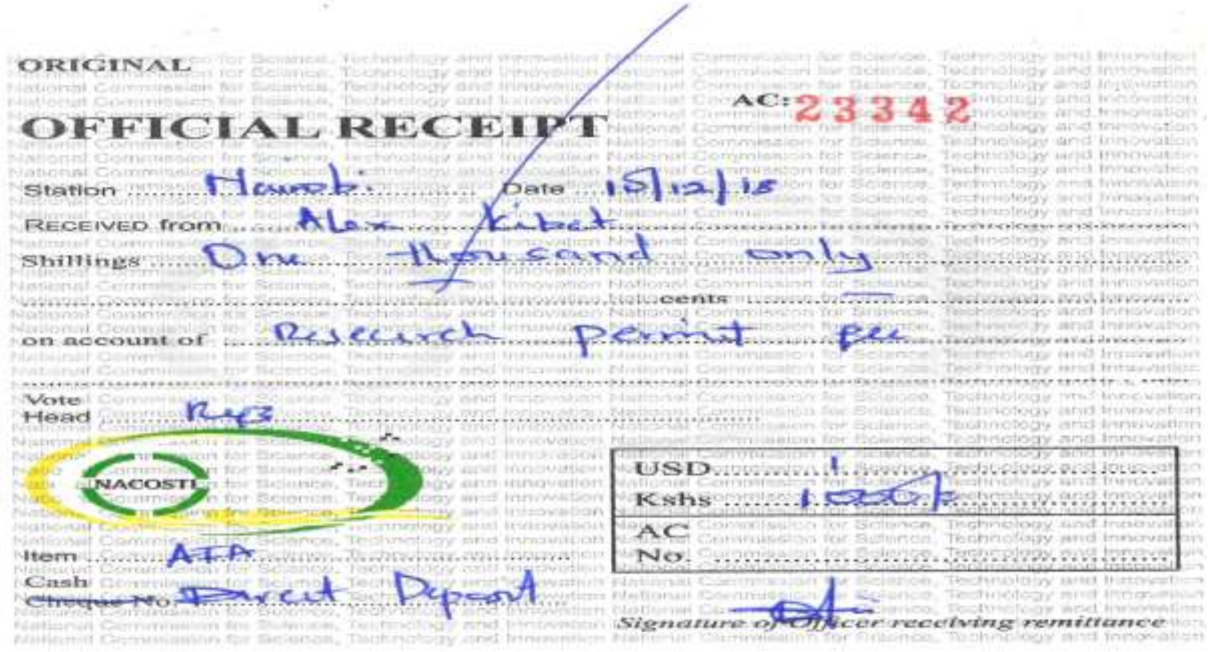
Dr. Betty Tikoko
DIRECTOR - (POST GRADUATE STUDIES)

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, Jesus as Lord. (1 Peter 3:15)

Kabarak University is ISO 9001:2015 Certified

APPENDIX III: OFFICIAL RECIEPT FOR PAYMENT OF RESEARCH PERMIT



ORIGINAL

OFFICIAL RECEIPT

AC: 23342

Station ....Nairobi.............. Date 15/12/15

RECEIVED from ....Alex .... Kibet..................................

Shillings ....One ....thousand .... only..................................

.... cents ....

on account of ....Research .... permit .... fee..................................

....

Vote Head ....Rg3..................................

NACOSTI

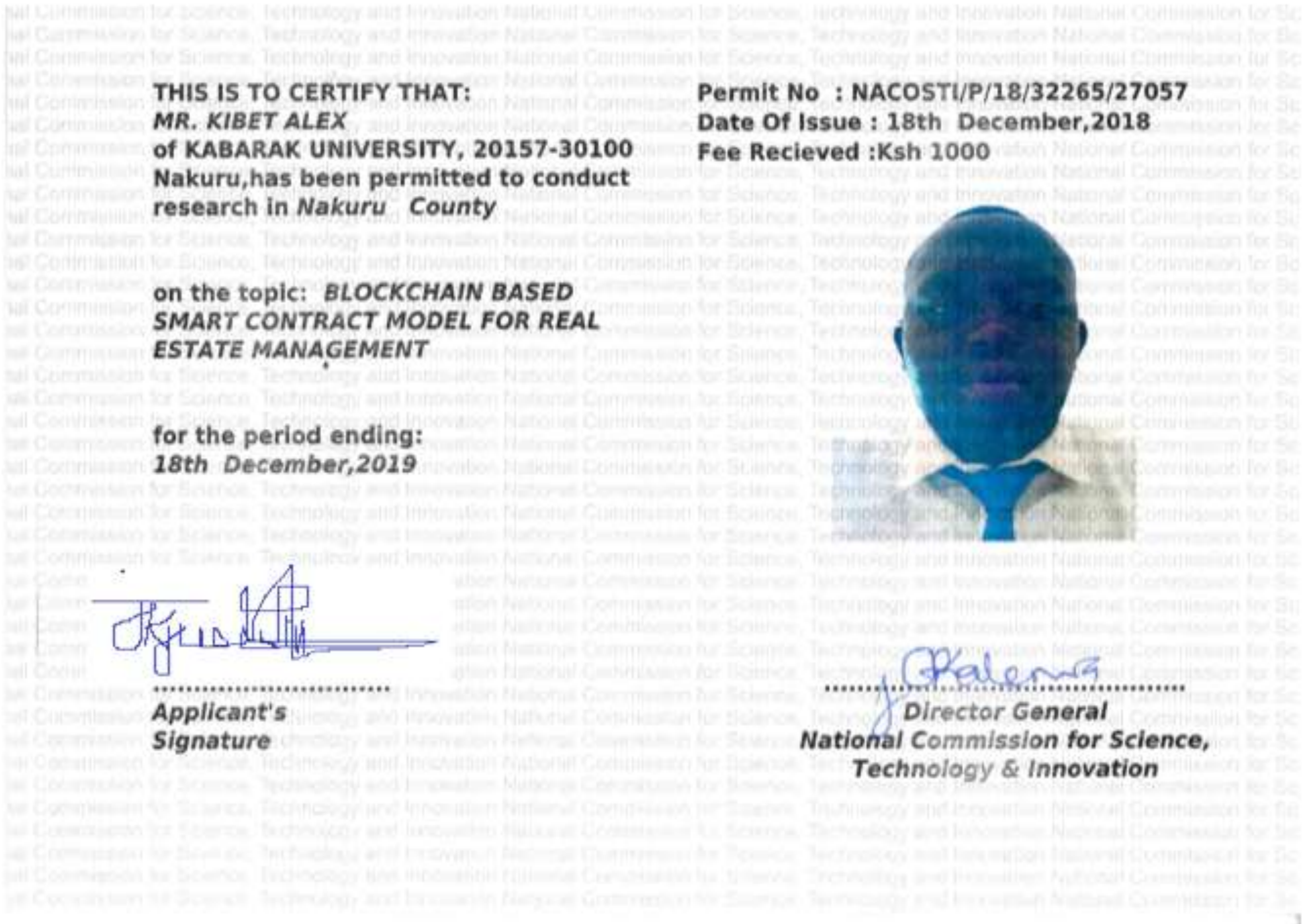
USD	....!..................
Kshs	....1,000/-..................
AC No.	....

Item ....AIA..................................

Cash ~~Cheque No.~~ ....Direct Deposit..................

Signature of Officer receiving remittance

APPENDIX IV: RESEARCH PERMIT



THIS IS TO CERTIFY THAT:
MR. KIBET ALEX
of KABARAK UNIVERSITY, 20157-30100
Nakuru,has been permitted to conduct
research in *Nakuru County*

on the topic: *BLOCKCHAIN BASED SMART CONTRACT MODEL FOR REAL ESTATE MANAGEMENT*

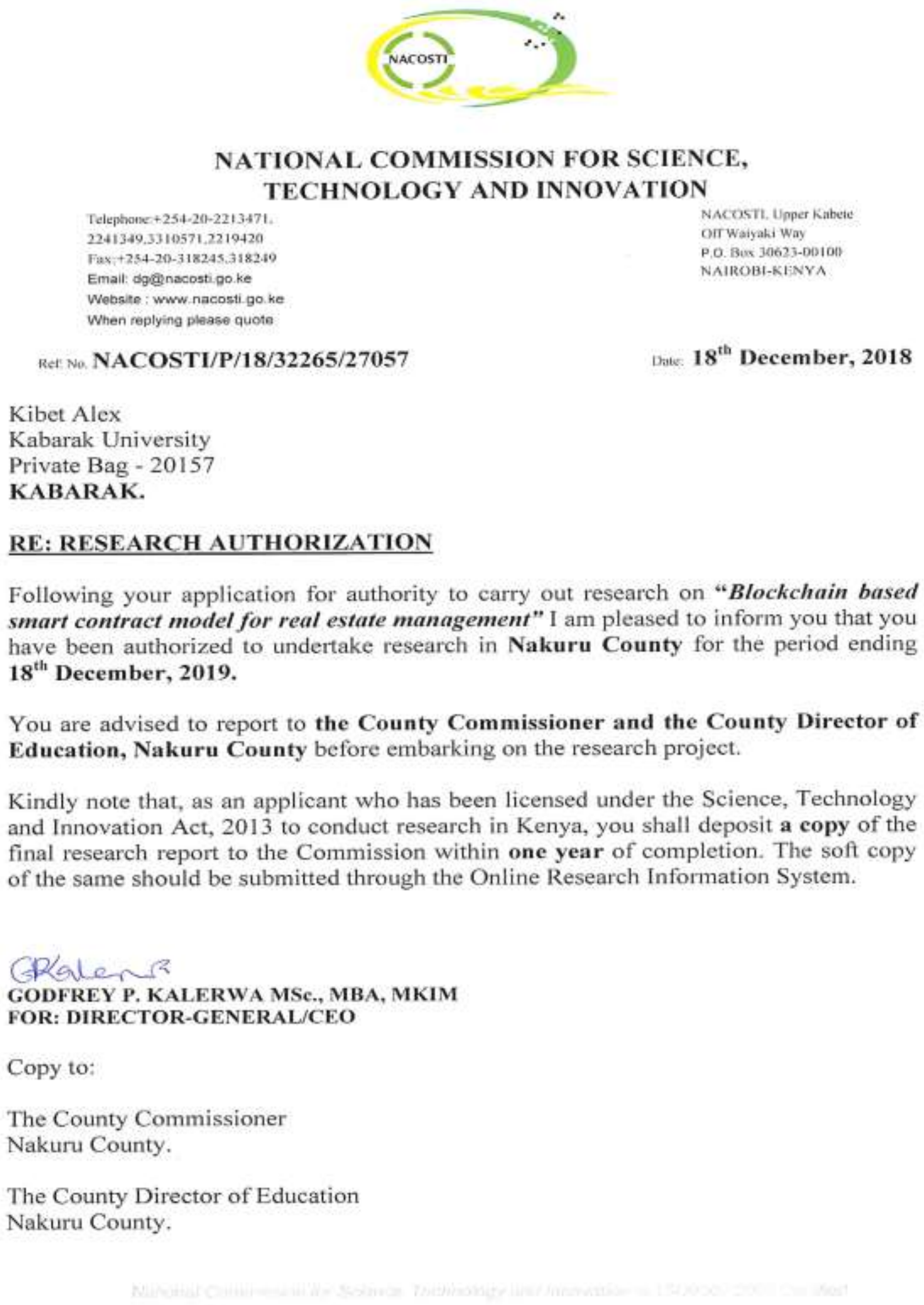
for the period ending:
18th December,2019

Permit No : NACOSTI/P/18/32265/27057
Date Of Issue : 18th December,2018
Fee Recieved :Ksh 1000

....
Applicant's Signature

....
Director General
National Commission for Science, Technology & Innovation

APPENDIX V: NACOSTI RESEARCH PERMIT



NACOSTI

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/18/32265/27057**

Date: **18th December, 2018**

Kibet Alex
Kabarak University
Private Bag - 20157
KABARAK.

RE: RESEARCH AUTHORIZATION

Following your application for authority to carry out research on **"*Blockchain based smart contract model for real estate management*"** I am pleased to inform you that you have been authorized to undertake research in **Nakuru County** for the period ending **18th December, 2019.**

You are advised to report to **the County Commissioner and the County Director of Education, Nakuru County** 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

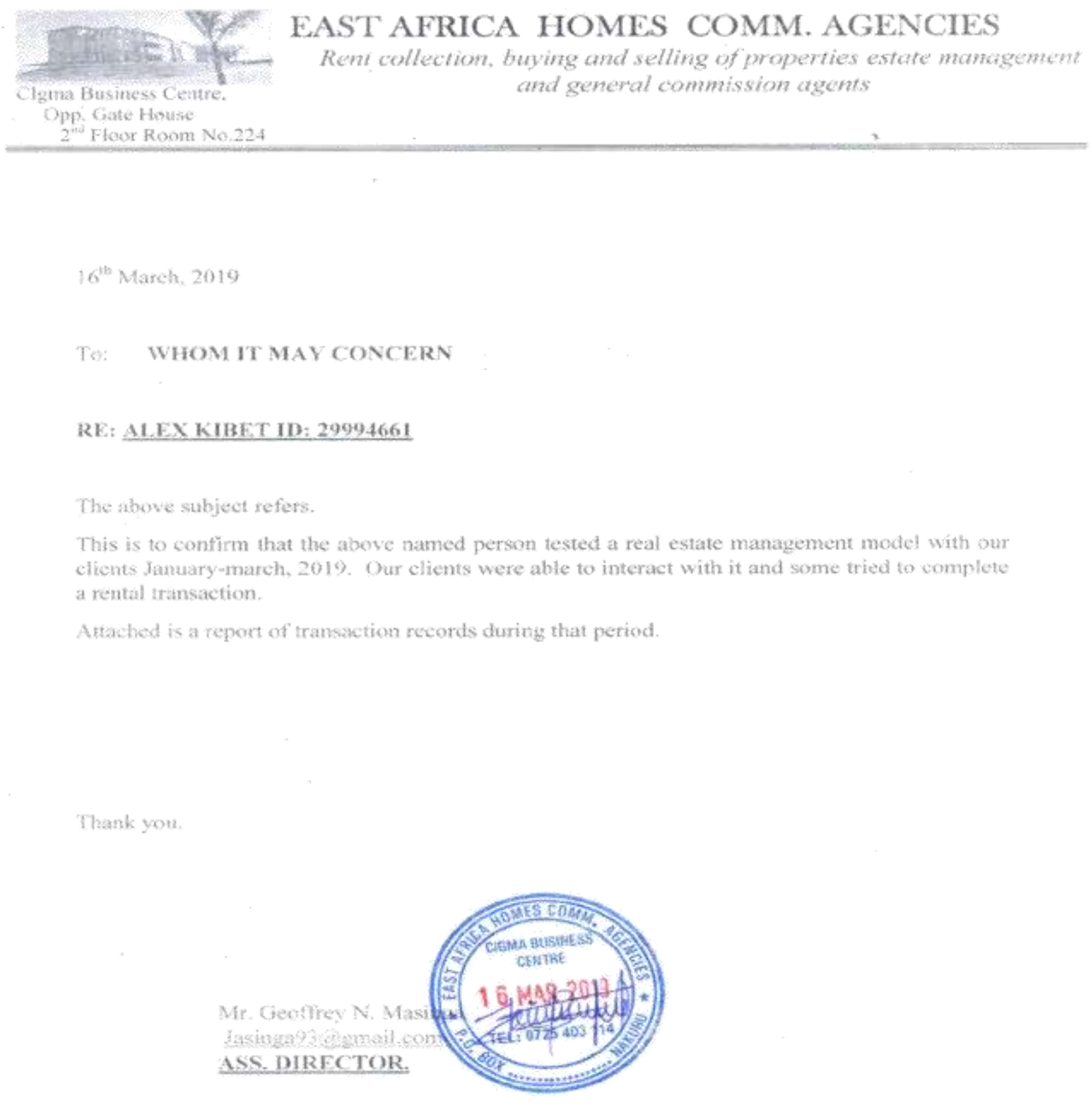
Copy to:

The County Commissioner
Nakuru County.

The County Director of Education
Nakuru County.

National Commission for Science, Technology and Innovation is ISO9001:2008 Certified

APPENDIX VI: SAMPLE REPORT OF AGENT “A”



EAST AFRICA HOMES COMM. AGENCIES

Rent collection, buying and selling of properties estate management and general commission agents

Cigma Business Centre,
Opp. Gate House
2nd Floor Room No.224

16th March, 2019

To: WHOM IT MAY CONCERN

RE: ALEX KIBET ID: 29994661

The above subject refers.

This is to confirm that the above named person tested a real estate management model with our clients January-march, 2019. Our clients were able to interact with it and some tried to complete a rental transaction.

Attached is a report of transaction records during that period.

Thank you.

EAST AFRICA HOMES COMM. AGENCIES
CIGMA BUSINESS CENTRE
16 MAR 2019
TEL: 0725 403 114
P.O. BOX .... NAKURU

Mr. Geoffrey N. Masi
Jasinga93@gmail.com
ASS. DIRECTOR.

Txhash	Blockno	UnixTimestamp
0x4f9e909e96416d9d4e2cd18460941df2bd54 ba0c46dd8eebb8a792301e8aed80	5070574	1550842150
0xa4df7c7bd246a9e5de7bfd6bac29e9a5cadffa3a6da2dd8ac4a22faeced772ca	5102742	1551259083
0x8f2a22c99ab613f27201062f08c8251ba2669a3dd5dd83e5119bd7dd41559479	5137983	1551718836
0x315d74a791457efec5c9a176a30e983f4f3d15176064de335648a806f0f03316	5234443	1552992765
0xdd787bd3a83f166289fcd3b4b91b8e02e9fa2e0d2895b36e5b93336ace5b02a1	5235275	1553003197
0xa799b13f9150e48c8bf8c85e09c0eed0a72c85ee0b8243865ed1d1dcefbd5217	5235278	1553003214
0x4e4b945584bb863b3a648b70e007130db4f214294f72fe2f8d39322f94c979b2	5235449	1553005518
0x693fe0148757bdc6977d728a8711b8ec8814fc6e9593a343681c55a67bf69ac6	5710843	1559395986
0xda1da60d42cc73963a9d4d251e2a798fb707c7ca1e61eb93f622c051894e1e23	5716807	1559478661
0xec2b91a3e98bb1b313f26bbc41b0fa53d8ad23995778c70ccfc00d7690316c28	5730858	1563214002
0x2b1848f7e0bc35ec4a37150e360c165dd57fe6738e66bdabbdf4ab5c35389143	4997940	1549886803
0x76959f4e1313c2c8dd0290ab9bf4ab6e4053bfeeee101a94544f457043973950	4997945	1549886834
0xd0ba1bd76826f77cf04884a4ab69d1ba739f0f5e920b69f51142662aa3597c61	5050576	1550585585
0x0b7e8ae782ae974fce8b2f2a59e96046e76aee39540132f64a641536bc7966dc	5050583	1550585675
0x399afc847bfd656dd2b577a9946ce7359ed627c7a244e5eaf2b1bf74a4ff4eeb	5050586	1550585725
0x4b696950b0e7ddbabf5690117d0b7f766c1f4c4d5864d85c265e258a4dcba392	5050593	1550585829
0xff966566bc2d9459c457c5380fc08864e734639739b2df75327728d80c966bf5	5050603	1550585932
0xe846bab400c32b24fafd3ea7dee171dd6127b02e732e570bdefb793b1d6d712e	5050610	1550585979
0xc9f1edf319fabd79df74036d19ab768aea4e018ad9ced536f1e9daf539ba4c3c	5051329	1550594411
0xa12b67ff2b1c931cc23bdb2cfb878edac9275a9450e28212e2cdcb69a6891807	5062185	1550731366
0xa5ab67809547e34e733495c03016258cf8e98cfa8ae231a35a258c315601c522	5062215	1550731775
0x90f8bff5b0c55d19352c7b9b938531dd3d3224d6d8846cfb3c1df7696ac3230c	5062228	1550731914
0x3b2aefc6b679cd3e15c7daf0e6a0da40baf6ef3ac18f5ff307e151a8f4d84058	5062243	1550732195
0x86f2698ed7e5d28e81707240ca1981d11e0413f655f5875500731d50faa4a55e	5062243	1550732195
0x8ba5e31e58b0813e2c1170a16d71328181803715c9a9f7160173da2b74a7f92c	5062458	1550735344

ST AFRICA HOMES COMM.
CIGMA BUSINESS
16 MAR 2019

DateTime	From
2/26/2019 13:29	0x32cd37a01a6286058507d1f018c8cb8676a90aaf
2/27/2019 9:18	0xc6e532bb14d444a4af14614f300453609018c96c
2/29/2019 5:00	0x0289bd742d56e94dd04f7ba09185c5a139b35b2f
3/4/2019 17:00	0x13e7b6833e25e19b9d19ba3153bb95c173afdd34
3/7/2019 13:46	0x13e7b6833e25e19b9d19ba3153bb95c173afdd34
3/9/2019 13:46	0x13e7b6833e25e19b9d19ba3153bb95c173afdd34
3/10/2019 14:25	0x13e7b6833e25e19b9d19ba3153bb95c173afdd34
3/11/2019 13:33	0xebd3872ea4d1d85a03a581127b872c99ebbab000
6/12/2019 12:31	0xbb84b6d71c6c6626809bc1b971c9a1d9b2eb5000
6/12/2019 17:31	0x861b8b7c649d9717673f53dd8f02fc43c6fee09d
2/11/2019 12:06	0x32cd37a01a6286058507d1f018c8cb8676a90aaf
2/11/2019 12:07	0x32cd37a01a6286058507d1f018c8cb8676a90aaf
2/19/2019 14:13	0x32cd37a01a6286058507d1f018c8cb8676a90aaf
2/19/2019 14:14	0x32cd37a01a6286058507d1f018c8cb8676a90aaf
2/19/2019 14:15	0x32cd37a01a6286058507d1f018c8cb8676a90aaf
2/19/2019 14:17	0x32cd37a01a6286058507d1f018c8cb8676a90aaf
2/19/2019 14:18	0x32cd37a01a6286058507d1f018c8cb8676a90aaf
2/19/2019 14:19	0x32cd37a01a6286058507d1f018c8cb8676a90aaf
2/19/2019 16:40	0x32cd37a01a6286058507d1f018c8cb8676a90aaf
2/21/2019 6:42	0x32cd37a01a6286058507d1f018c8cb8676a90aaf
2/21/2019 6:49	0x32cd37a01a6286058507d1f018c8cb8676a90aaf
2/21/2019 6:51	0x32cd37a01a6286058507d1f018c8cb8676a90aaf
2/21/2019 6:56	0x32cd37a01a6286058507d1f018c8cb8676a90aaf
2/21/2019 6:56	0x32cd37a01a6286058507d1f018c8cb8676a90aaf
2/21/2019 7:47	0x32cd37a01a6286058507d1f018c8cb8676a90aaf
2/21/2019 7:49	0x32cd37a01a6286058507d1f018c8cb8676a90aaf

2/19/2019 14:15

T AFRICA HOMES COMM.
CIGMA BUSINESS
[illegible] 2019

To

0x13e7b6833e25e19b9d19ba3153bb95c173afdd34
0x32cd37a01a6286058507d1f018c8cb8676a90aaf
0x32cd37a01a6286058507d1f018c8cb8676a90aaf
0x32cd37a01a6286058507d1f018c8cb8676a90aaf
0x32cd37a01a6286058507d1f018c8cb8676a90aaf
0x32cd37a01a6286058507d1f018c8cb8676a90aaf
0x32cd37a01a6286058507d1f018c8cb8676a90aaf
0x32cd37a01a6286058507d1f018c8cb8676a90aaf
0x32cd37a01a6286058507d1f018c8cb8676a90aaf
0x32cd37a01a6286058507d1f018c8cb8676a90aaf

0xd7f21fc9413c1f70d220492c73cfa9721f30d608

0x6bd735f0057adb5d90968c4c331125258fedc62e

0x6bd735f0057adb5d90968c4c331125258fedc62e
0x885037b35420b889a79e5201a9166ecfcab022d7
0x885037b35420b889a79e5201a9166ecfcab022d7
0x3c836abfa6bf05cd02f8133e0d9cd1d24e18f918
0x3c836abfa6bf05cd02f8133e0d9cd1d24e18f918
0x3bf4cbe3193750f1c1b53867fd903e54a605077b
0x3bf4cbe3193750f1c1b53867fd903e54a605077b
0x885037b35420b889a79e5201a9166ecfcab022d7
0xe2a7906f66a98ef28b0331c7aa6b76bc0211f260

ContractAddress

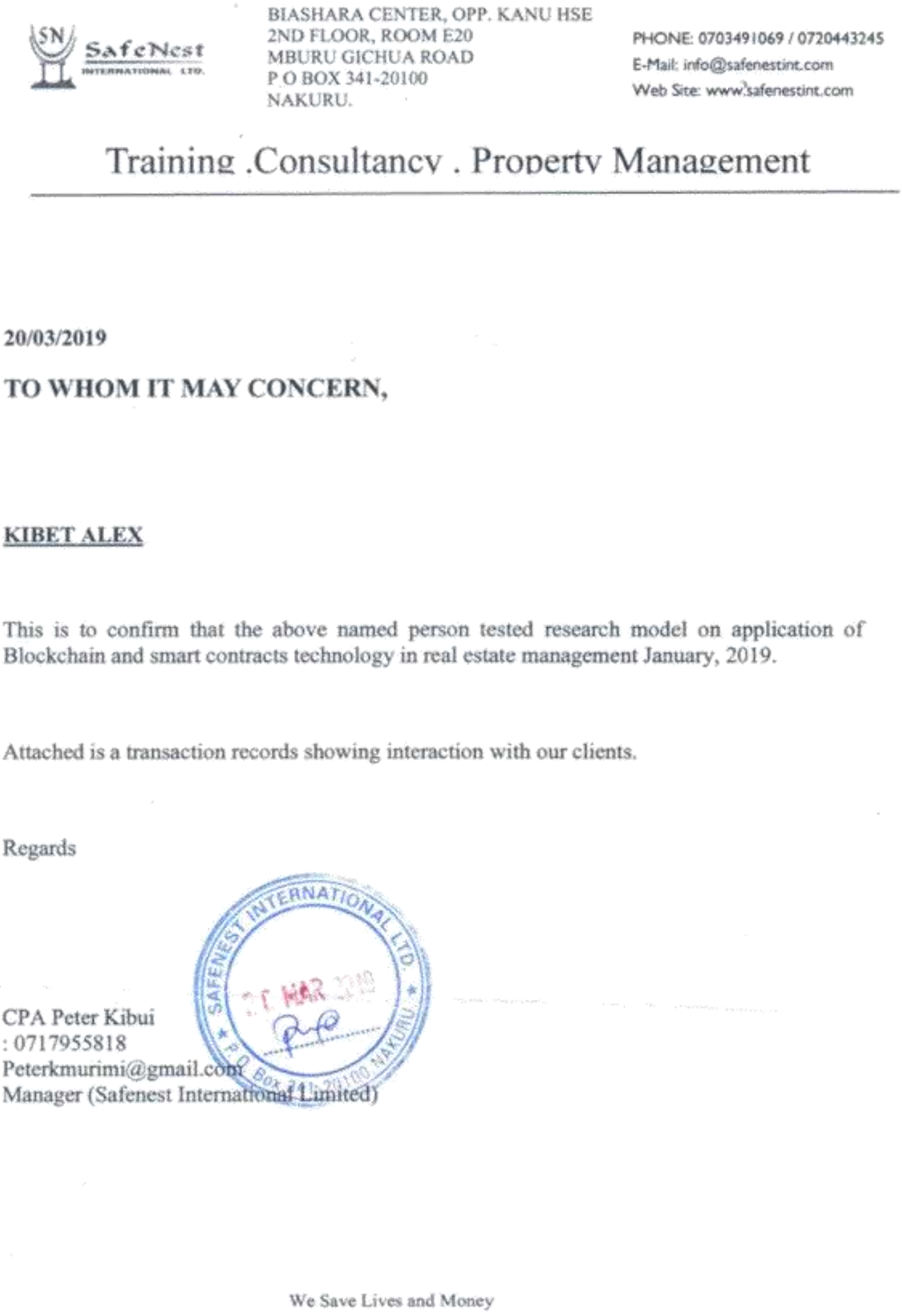
0xd9f1f9fc8d649b23937918f3d764306a49026010
0x6bd735f0057adb5d90968c4c331125258fedc62e
0x01d5fb6a42b54c86ed12ff35e371893cd0686f15
0x59b4755d38d2be6d05b62137e1d3eb9a239ebd5f
0x5c8af4241ef3b8b8785b33a50c899e27bce554a5
0x236ad9817ae3d957532ea279469eb86f1dd673d8
0x8ec2941f8da0c52a3b412619eae588d3675b26bf
0x64404296855fef130c1f215a3dbf9d940b332607
0x08e743af812a37751c1cfc0fd61f6b997eec0c6c
0xca1c21b3450e2d48e0714484a066cc4d2212976a
0x01d5fb6a42b54c86ed12ff35e371893cd0686f15

0x6bd735f0057adb5d90968c4c331125258fedc62e

0xd9f1f9fc8d649b23937918f3d764306a49026010
0x3c836abfa6bf05cd02f8133e0d9cd1d24e18f918
0x885037b35420b889a79e5201a9166ecfcab022d7

Value/Qua	CurrentVal	TxnFee(ETI	TxnFee(USD)	Historical $	Status	ErrCode
20000	0	0.000021	0			
3000	0	0.000021	0			
2300	0	0.000021	0			
1700	0	0.000021	0			
1100	0	0.000021	0			
2000	0	0.000021	0			
2760	0	0.005698	0			
1900	0	0.000841	0			
1000	0	0.022169	0			
1200	0	0.026165	0			
220	0	0.024733	0			
100	0	0.000541	0			
100	0	0.005698	0			
0	0	0.000841	0			
55	0	0.022169	0			
31	0	0.026165	0			
0	0	0.025462	0			
0	0	0.000541	0			
0	0	0.00015	0			
18	0	5.3E-05	0		Error(0)	Reverted
0	0	0.000195	0			
0	0	0.000127	0			
0	0	0.00021	0			
0	0	0.00021	0			
11	0	5.32E-05	0		Error(0)	Reverted
0	0	0.00021	0			

APPENDIX VI: SAMPLE REPORT OF AGENT "B"



SN SafeNest INTERNATIONAL LTD.

BIASHARA CENTER, OPP. KANU HSE
2ND FLOOR, ROOM E20
MBURU GICHUA ROAD
P.O BOX 341-20100
NAKURU.

PHONE: 0703491069 / 0720443245
E-Mail: info@safenestint.com
Web Site: www.safenestint.com

Training .Consultancy . Property Management

20/03/2019

TO WHOM IT MAY CONCERN,

KIBET ALEX

This is to confirm that the above named person tested research model on application of Blockchain and smart contracts technology in real estate management January, 2019.

Attached is a transaction records showing interaction with our clients.

Regards

SAFENEST INTERNATIONAL LTD. ★ P.O. Box 341-20100 NAKURU ★
20 MAR 2019

CPA Peter Kibui
: 0717955818
Peterkmurimi@gmail.com
Manager (Safenest International Limited)

We Save Lives and Money

Txhash	Blockno	UnixTimestam	DateTime
0x718d928d0fd11e0aa208f68790fd6cc5fae209d02f948e7d28ee34b30c9ddd37	5710830	1550747144	2/22/2019 12:33
0xde802d9d1551c1034a33229de120b7ec8f439b09ff34532321fff6e34c6dbb62	5710836	1550747180	2/23/2019 12:34
0x22762b88615c52f8fc00b19bbd8edbf4d1ccfb4ba2a1a32bf2301c6b0ad8a35b	5710838	1550747237	2/23/2019 12:35
0xca5532d2db4b4c52a0167172e2762dd2a3a0e37b9d81131397229e83d8258244	5710850	1550747399	2/25/2019 12:36
0x493dd0186d383d43c515b2a752b2ea16d88c446bf910070610a1792bb9825b07	5710853	1550747710	2/26/2019 13:00
0x5241679488b904366221c183f26aa531e6129e0a047e479037f26b4f612e10ec	5710855	1550748107	2/27/2019 13:20
0xf849bdc366a4a85700f703b541ef323d15b4c450b43e4d0766dbc510429916	5710859	1550748160	2/28/2019 13:25
0xbfdede075bdf7966d01952e0a0c55dedc7e030b4b60c332e8494eaf230339097	5710864	1550748166	2/28/2019 13:29
0x34b7190f2f17db1057ae3c1b8aa35c6d0db804043c376a5c14efb60235705d76	5710869	1550748223	3/1/2019 9:18
0xbd25f6fba8b002c5b3f272d82cf0f682794e490d2176f1785727251c3cf40921	5710877	1550748285	3/4/2019 17:00
0xe9c2d81af3a12f6a76ac798ef19f92be377d21229375c479849a3c121c26dccc	5710879	1550748308	3/9/2019 13:43
0x2ddabe39c7479e4b988f51231dfbecfa59e3e840606db0a3aeb797348d9c74e5	5710881	1550838175	3/9/2019 20:37
0x6f24353cc3b6cb4ac3f2e356067d9e5e50bc1924dc11bdd85879fd6f4156aa90	5710888	1550838784	3/10/2019 20:37
	5050583	1550585675	2/19/2019 14:14
0x0b7e8ae782ae974fce8b2f2a59e96046e76aee39540132f64a641536bc7966dc	5050586	1550585725	2/19/2019 14:15
0x399afc847bfd656dd2b577a9946ce7359ed627c7a244e5eaf2b1bf74a4ff4eeb	5050593	1550585829	2/19/2019 14:17
0x4b696950b0e7ddbabf5690117d0b7f766c1f4c4d5864d85c265e258a4dcba392	5050603	1550585932	2/19/2019 14:18
0xff966566bc2d9459c457c5380fc08864e734639739b2df75327728d80c966bf5	5050610	1550585979	2/19/2019 14:19
0xe846bab400c32b24fafd3ea7dee171dd6127b02e732e570bdefb793b1d6d712e	5051329	1550594411	2/19/2019 16:40
0xc9f1edf319fabd79df74036d19ab768aea4e018ad9ced536f1e9daf539ba4c3c	5062185	1550731366	2/21/2019 6:42
0xa12b67ff2b1c931cc23bdb2cfb878edac9275a9450e28212e2cdcb69a6891807	5062215	1550731775	2/21/2019 6:49
0xa5ab67809547e34e733495c03016258cf8e98cfa8ae231a35a258c315601c522	5062228	1550731914	2/21/2019 6:51
0x90f8bff5b0c55d19352c7b9b938531dd3d3224d6d8846cfb3c1df7696ac3230c	5062243	1550732195	2/21/2019 6:56
0x3b2aefc6b679cd3e15c7daf0e6a0da40baf6ef3ac18f5ff307e151a8f4d84058	5062243	1550732195	2/21/2019 6:56
0x86f2698ed7e5d28e81707240ca1981d11e0413f655f5875500731d50faa4a55e	5062452	1550735222	2/21/2019 7:47
0x8ba5e31e58b0813e2c1170a16d71328181803715c9a9f7160173da2b74a7f92c	5062458	1550735344	2/21/2019 7:49
0x1a4b4ff9b4f9c4f29dd8bb2e9bb9b64ce307cd13a1ccd6f8077fc243cda531bd	5062532	1550736444	2/21/2019 8:07
0xfc5107fbf6b65dd4bdb471e8c314b89cd6bb535a016b5f71f58f9cfcada97d49	5062538	1550736564	2/21/2019 8:09
0x5826b3fd8f908db8182c21bb954a4d0cd65d87629b29359fa729623b7139d7aa	5062553	1550736852	2/21/2019 8:14
0xdd1ddb313ac96db6a10da1034950c9c5a444f00260aa88ee69b38393335c5c7e	5062603	1550737476	2/21/2019 8:24
0xc2b9779faa163998ed6401adf4853556233cc865ca46925eab0ea8c96e95593b	5062647	1550738300	2/21/2019 8:38
0xc04b9b465c6fc3bad628886153690e0ce000e8d4eb820cfa30e29434eb7c3e04	5062654	1550738345	2/21/2019 8:39
0xf42834b27946b14d2924a5e6a24a1e3843ae934aaa7ed14a87a486b4175e6f64	5062662	1550738399	2/21/2019 8:39

0xad82e3eaaa866743aa94ff4084bf33c5ce70f495709588ad9516ecc055627895	5062671	1550738554	2/21/2019 8:42
0xf22de077c153f73d6b2f4fc798e78f09940cfc091e84cc5267984e66546018f5	5062679	1550738641	2/21/2019 8:44
0x22541257ae8fa06eefc21a477b2c9353fa4b02791fc5c375ba35d8c3cb4c5420	5062688	1550738746	2/21/2019 8:45
0xf70fd26abfd1bba87ad89afa5e03ebbae92183c5974858d8a3ef85a70ba7cc1e	5062769	1550739881	2/21/2019 9:04
0xb5dca90b338a21ea3197b18b62e6e2f5d3a04076809e30ef2adb838176a17a4b	5062834	1550740793	2/21/2019 9:19
0x171e8833a5bac8165c0f0b55d28cbb2913d4e9fd4e57b3a140a93b6d673a3fc7	5063267	1550747100	2/21/2019 11:05
0x5b1fc80dd6cc86cdd1a466cc68c1e90f7af66a4d0823e37464009334bbb27472	5063270	1550747119	2/21/2019 11:05
0xcf5614e8e6c995ad61a467793321ddefa5dd7672b92ec8531da295c2cfb536dc	5063271	1550747144	2/21/2019 11:05
0x5718c6e8768d4ec8dcb5acc8ccad773a6538b4751d8c1e97498d6bc5c8531d38	5063276	1550747180	2/21/2019 11:06
0x4e99d59ea0b81a6ce2d387edff248188fd6493e7119268b66ac4cb33534f7c7b	5063281	1550747237	2/21/2019 11:07
0x29b5d3277afca8d26dd36967985abd131fafb75a489818d80ad5fe6f55a2036e	5063286	1550747399	2/21/2019 11:09
0xab092954a46b63e69e44f977487f4e410bac30b94912cbfe191b320deb13666d	5063315	1550747710	2/21/2019 11:15
0x15f61f4a5336341f21bde9e4a2e81287f93bf1bff350c32de6db19fe2f6ebd78	5063358	1550748107	2/21/2019 11:21
0x9406e7a5b18026cd33aa44d90a81b64c6534f87e030bc57394f2e6130e53d63a	5063362	1550748160	2/21/2019 11:22
0x41c9f27d7f29b0f16bbfd9f6dcad26e27737bf01356c8a3f02c5b6ed828463d6	5063363	1550748166	2/21/2019 11:22
0x3f8a09bd719d8da29fb17336b91e7f83054f548c406486fafb4be52db4470684	5063366	1550748223	2/21/2019 11:23
0xccf0dde220c69b626497d0fe8f69dc4041262866d8ec5a8b235626a7150a9cd3	5063372	1550748285	2/21/2019 11:24
0x0aacd0221db5603133b2838c89aaee2cab2ebb5bfc637433b09b0c4ac973e960	5063376	1550748308	2/21/2019 11:25
0x20bc648e6baa87332a9b3e2beaa800563db936f67295288032c0431d21d76a46	5070178	1550838175	2/22/2019 12:22
0x86a6df8ee54bd9e46d0eb67f688ebddd14e61eee56dbdd63548b37bb0c73d29d	5070242	1550838784	2/22/2019 12:33
0x9d9b97e3a7928a25d102cc4efc6a2d3dbd6898d71faebf4f9c11ab31db41078c	5070247	1550838818	2/22/2019 12:33
0x1b474872d0639b6ce35ffbe71b638d46a521c8905eba17c2d8534656033788f3	5070248	1550838819	2/22/2019 12:33
0xaffb8f108c6097aa38e6857cc23a31748b859845b9616f54705dfe314d36ee5f	5070254	1550838873	2/22/2019 12:34
0x2ba6d56ffe29523f239d6f8aadf1bf8766763ccfc2e40185de140131b8049e7e	5070260	1550838916	2/22/2019 12:35
0x335fb51698b0d2aaf15df00ea4656d7829fd66eaf3084860387c7cd5a117b111	5070269	1550839010	2/22/2019 12:36

SAFENEST INTERNATIONAL LTD.
20 MAR 2019
* P.O. Box [illegible] 20100 NAKURU *

From	To	ContractAddress	Value/Quantity(tl
0x236ad9817ae3d957532e279469eb86f1dd673d8	0x13e7b6833e25e19b9d19ba3153bb95c173afd		2500
0x08e743af812a37751c1cfc0fd61f6b997eec0c6c	0x13e7b6833e25e19b9d19ba3153bb95c173afd		2500
0x64404296855fef130c1f215a3dbf9d940b332607	0x13e7b6833e25e19b9d19ba3153bb95c173afd		2500
0x3f4eb593904495a346df4bc7c80a55a519aa4295	0x13e7b6833e25e19b9d19ba3153bb95c173afd		2500
0x3f4eb593904495a346df4bc7c80a55a519aa4295	0x13e7b6833e25e19b9d19ba3153bb95c173afd		2500
0x08e743af812a37751c1cfc0fd61f6b997eec0c6c	0x13e7b6833e25e19b9d19ba3153bb95c173afd		2500
0xcc2acae48f60a324c8ae0fa2c991b8e741d72652	0x13e7b6833e25e19b9d19ba3153bb95c173afd	0xd7f21fc9413c1f70d220492c73cfa9721f30d	2500
0x13e7b6833e25e19d19ba3153bb95c173afdd34	0x13e7b6833e25e19b9d19ba3153bb95c173afd		2500
0xbb84b6d71c6c662609bc1b971c9a1d9b2eb5000	0x13e7b6833e25e19b9d19ba3153bb95c173afd	0xdcccbafa732f7184530e7d7c18f81583de0f0	2500
0x861b8b7c649d9717673f53dd8f02fc43c6fee09d	0x13e7b6833e25e19b9d19ba3153bb95c173afd	0xf307164e5e5c1cf4c6392f7606d8de652b1f	2500
0xfe96912711c1fc37168df6e84e0290339f3b2141	0x13e7b6833e25e19b9d19ba3153bb95c173afd	0x01d5fb6a42b54c86ed12ff35e371893cd068	2500
0x69670b0c1b10073812415dd474804bb32b3aeca	0x13e7b6833e25e19b9d19ba3153bb95c173afd		2500
0x69670b0c1b10073912415dd474804bb32b3aeca	0x13e7b6833e25e19b9d19ba3153bb95c173afd	0x6bd735f0057adb5d90968c4c331125258fec	2400
0x32cd37a01a6286058507d1f018c8cb8676a90aaf	0x6bd735f0057adb5d90968c4c331125258fedc62e		333
0x32cd37a01a6286058507d1f018c8cb8676a90aaf		0xd9f1f9fc8d649b23937918f3d764306a4902	30
0x32cd37a01a6286058507d1f018c8cb8676a90aaf		0x3c836abfa6bf05cd02f8133e0d9cd1d24e18	500
0x32cd37a01a6286058507d1f018c8cb8676a90aaf		0x885037b35420b889a79e5201a9166ecfcab	100
0x32cd37a01a6286058507d1f018c8cb8676a90aaf	0x6bd735f0057adb5d90968c4c331125258fedc62e		2400
0x32cd37a01a6286058507d1f018c8cb8676a90aaf	0x885037b35420b889a79e5201a9166ecfcab022d7		2500
0x32cd37a01a6286058507d1f018c8cb8676a90aaf	0x885037b35420b889a79e5201a9166ecfcab022d7		25
0x32cd37a01a6286058507d1f018c8cb8676a90aaf	0x3c836abfa6bf05cd02f8133e0d9cd1d24e18f918		2400
0x32cd37a01a6286058507d1f018c8cb8676a90aaf	0x3c836abfa6bf05cd02f8133e0d9cd1d24e18f918		20000
0x32cd37a01a6286058507d1f018c8cb8676a90aaf	0x3bf4cbe3193750f1c1b53867fd903e54a605077b		60
0x32cd37a01a6286058507d1f018c8cb8676a90aaf	0x3bf4cbe3193750f1c1b53867fd903e54a605077b		35
0x32cd37a01a6286058507d1f018c8cb8676a90aaf	0x885037b35420b889a79e5201a9166ecfcab022d7		19
0x32cd37a01a6286058507d1f018c8cb8676a90aaf	0xe2a7906f66a98ef28b0331c7aa6b76bc0211f260		44
0x32cd37a01a6286058507d1f018c8cb8676a90aaf		0xd32e99835c38ce44bb1b1a40376a9adc1c1	70
0x32cd37a01a6286058507d1f018c8cb8676a90aaf	0xd32e99835c38ce44bb1b1a40376a9adc1c1b2229		100
0x32cd37a01a6286058507d1f018c8cb8676a90aaf		0x59b4755d38d2be6d05b62137e1d3eb9a23	100
0x32cd37a01a6286058507d1f018c8cb8676a90aaf		0x5c8af4241ef3b8b8785b33a50c899e27bce5	100
0x32cd37a01a6286058507d1f018c8cb8676a90aaf		0x236ad9817ae3d957532ea279469eb86f1dd	100
0x32cd37a01a6286058507d1f018c8cb8676a90aaf	0x236ad9817ae3d957532ea279469eb86f1dd673d8		100
0x32cd37a01a6286058507d1f018c8cb8676a90aaf		0x8ec2941f8da0c52a3b412619eae588d3675	0

SAFENEST INTERNATIONAL LTD
MAR

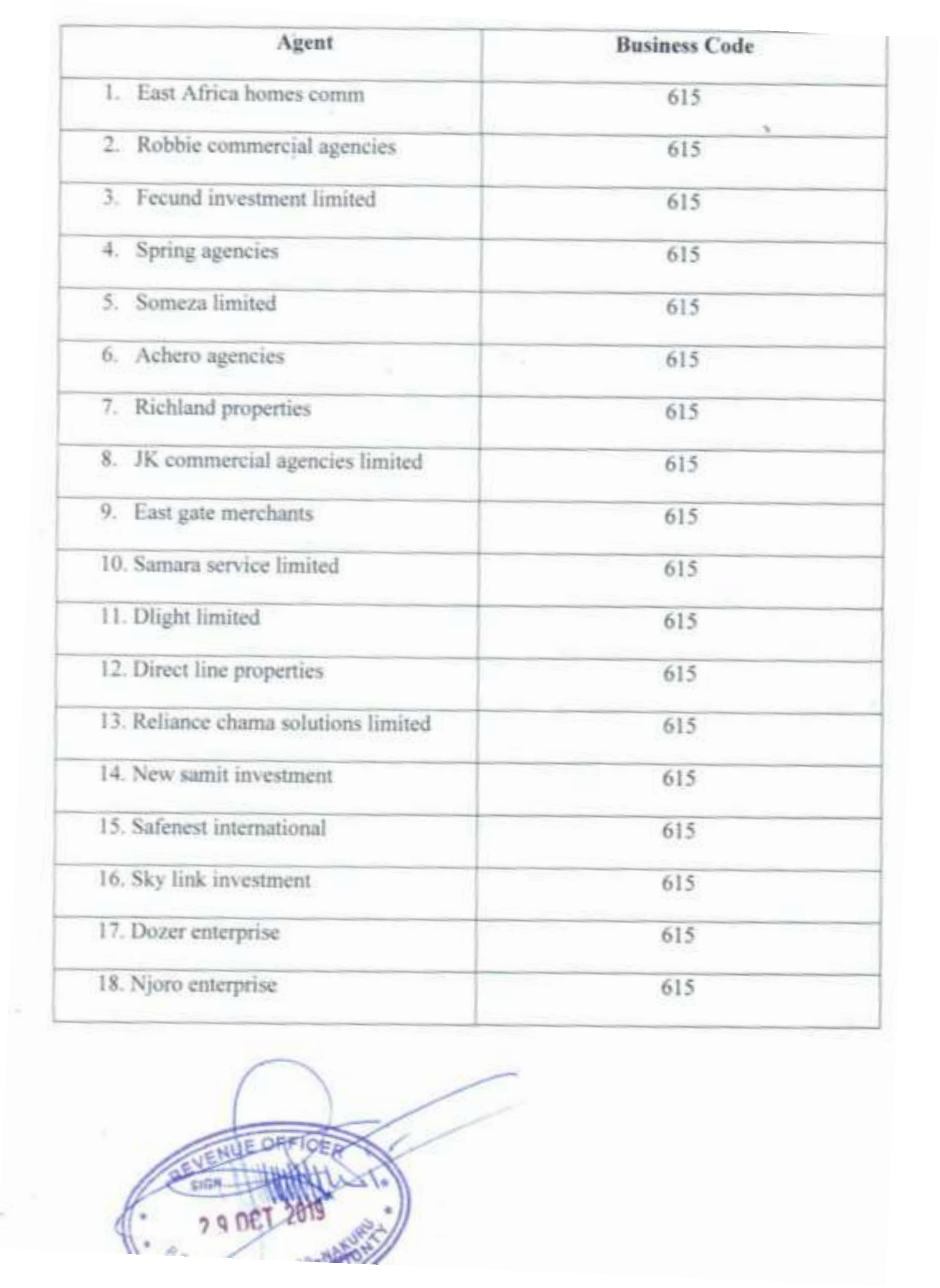
0x32cd37a01a6286058507d1f018c8cb8676a90aaf		0x64404296855fef130c1f215a3dbf9d940b33	0
0x32cd37a01a6286058507d1f018c8cb8676a90aaf		0x08e743af812a37751c1cfc0fd61f6b997eec0	0
0x32cd37a01a6286058507d1f018c8cb8676a90aaf	0x236ad9817ae3d957532ea279469eb86f1dd673d8		0
0x32cd37a01a6286058507d1f018c8cb8676a90aaf	0x08e743af812a37751c1cfc0fd61f6b997eec0c6c		0
0x32cd37a01a6286058507d1f018c8cb8676a90aaf	0x64404296855fef130c1f215a3dbf9d940b332607		0
0x32cd37a01a6286058507d1f018c8cb8676a90aaf		0x3f4eb593904495a346df4bc7c80a55a519aa	0
0x32cd37a01a6286058507d1f018c8cb8676a90aaf	0x3f4eb593904495a346df4bc7c80a55a519aa4295		0
0x32cd37a01a6286058507d1f018c8cb8676a90aaf		0x23f0d36ec08526282d60e6885baa1b70dd0	0
0x32cd37a01a6286058507d1f018c8cb8676a90aaf		0x9bfe27d702cc1b414a436dbb8018820cca4	0
0x32cd37a01a6286058507d1f018c8cb8676a90aaf		0x287ad9d012b1b8242e402e9de5fcda77710	250
0x32cd37a01a6286058507d1f018c8cb8676a90aaf	0x3f4eb593904495a346df4bc7c80a55a519aa4295		99
0x32cd37a01a6286058507d1f018c8cb8676a90aaf	0x08e743af812a37751c1cfc0fd61f6b997eec0c6c		700
0x32cd37a01a6286058507d1f018c8cb8676a90aaf		0xcc2acae48f60a324c8ae0fa2c991b8e741d7	600
0x32cd37a01a6286058507d1f018c8cb8676a90aaf	0xcc2acae48f60a324c8ae0fa2c991b8e741d72652		0
0x32cd37a01a6286058507d1f018c8cb8676a90aaf		0x81022bc319c961b1b79c83015ac97c816dd	0
0x32cd37a01a6286058507d1f018c8cb8676a90aaf		0x13e2345b06afe084a751376c54b3598a1e4	0
0x32cd37a01a6286058507d1f018c8cb8676a90aaf		0xec4b3c663de42c210ccf47bd45461418c735	0
0x32cd37a01a6286058507d1f018c8cb8676a90aaf	0xcc2acae48f60a324c8ae0fa2c991b8e741d72652		0
0x32cd37a01a6286058507d1f018c8cb8676a90aaf	0x13e7b6833e25e19b9d19ba3153bb95c173afdd34		0
0x32cd37a01a6286058507d1f018c8cb8676a90aaf		0xca1c21b3450e2d48e0714484a066cc4d221	0
0x32cd37a01a6286058507d1f018c8cb8676a90aaf	0xca1c21b3450e2d48e0714484a066cc4d2212976a		0
0x32cd37a01a6286058507d1f018c8cb8676a90aaf		0xfe96912711c1fc37168df6e84e0290339f3b	0
0x32cd37a01a6286058507d1f018c8cb8676a90aaf		0x25fdef81097bd56932135f7740b4c19f4d23	0
0x32cd37a01a6286058507d1f018c8cb8676a90aaf		0xb10f3cc0af774cd8ec92d85e98d0b2e86e41	0
0x32cd37a01a6286058507d1f018c8cb8676a90aaf	0xca1c21b3450e2d48e0714484a066cc4d2212976a		0

CurrentValue @	TxnFee(ETH)	TxnFee(USD)	Historical $Pr	Status	ErrCode
0	0.000021	0			
0	0.000021	0			
0	0.000021	0			
0	0.000021	0			
0	0.000021	0			
0	0.000021	0			
0	0.00569816	0			
0	0.00084068	0			
0	0.02216852	0			
0	0.02616538	0			
0	0.02473284	0			
0	0.00054068	0			
0	0.00569816	0			
0	0.00084068	0			
0	0.02216852	0			
0	0.02616538	0			
0	0.02546232	0			
0	0.00054068	0			
0	0.00015	0			
0	5.2999E-05	0		Error(0)	Reverted
0	0.00019477	0			
0	0.00012729	0			
0	0.00021	0			
0	0.00021	0			
0	5.3191E-05	0		Error(0)	Reverted
0	0.00021	0			
0	0.00569816	0			
0	0.00084068	0			
0	0.02216852	0			
0	0.02216852	0			
0	0.00569816	0			
0	0.00084068	0			
0	0.02216852	0			

SAFENEST INTERNATIONAL LTD. ★ NAKURU
MAR 2019

0	0.02616538	0		
0	0.02957116	0		
0	0.00054068	0		
0	0.00013525	0		
0	0.00019445	0		
0	0.00569816	0		
0	0.00084068	0		
0	0.02216852	0		
0	0.02616538	0		
0	0.02957116	0		
0	0.00054068	0		
0	5.3255E-05	0	Error(0)	Reverted
0	0.00569816	0		
0	0.00084068	0		
0	0.02216852	0		
0	0.02616538	0		
0	0.02957116	0		
0	0.00054068	0		
0	0.00021	0		
0	0.00569816	0		
0	0.00084068	0		
0	0.02216852	0		
0	0.02616538	0		
0	0.02957116	0		
0	0.00054068	0		

APPENDIX VII: LICENSED AGENCIES FOR SAMPLE FRAME



Agent	Business Code
1. East Africa homes comm	615
2. Robbie commercial agencies	615
3. Fecund investment limited	615
4. Spring agencies	615
5. Someza limited	615
6. Achero agencies	615
7. Richland properties	615
8. JK commercial agencies limited	615
9. East gate merchants	615
10. Samara service limited	615
11. Dlight limited	615
12. Direct line properties	615
13. Reliance chama solutions limited	615
14. New samit investment	615
15. Safenest international	615
16. Sky link investment	615
17. Dozer enterprise	615
18. Njoro enterprise	615

REVENUE OFFICER
SIGN
29 OCT 2019
NAKURU COUNTY