KWAME NKRUMAH UNIVERSITY OF SCIENCE AND TECHNOLOGY

COLLEGE OF SCIENCE

DEPARTMENT OF COMPPUTER SCIENCE

A BLOCKCHAIN BASED ELECTRONIC VOTING SYSTEM

A PROJECT THESIS SUBMITTED BY:

ADAMS SUMAILA – 4621518

SAEED BASHAR - 4620618



September 1, 2022

Table of Contents

[DECLARATION iii](#_Toc116083958)

[DEDICATION iv](#_Toc116083959)

[ABSTRACT iv](#_Toc116083960)

[ACKNOWLEDGEMENT v](#_Toc116083961)

[CHAPTER ONE 1](#_Toc116083962)

[INTRODUCTION 1](#_Toc116083963)

[1.1. OVERVIEW 1](#_Toc116083964)

[1.2. PROBLEM DEFINITION 2](#_Toc116083965)

[1.3. PROBLEM MOTIVATION 3](#_Toc116083966)

[1.4. AIMS OF THE PROJECT 4](#_Toc116083967)

[1.5. OBJECTIVES OF THE PROJECT 4](#_Toc116083968)

[1.7. PROJECT TERMINOLOGIES 5](#_Toc116083969)

[1.9. PROJECT BENEFICIARIES 6](#_Toc116083970)

[1.10. PROJECT SCOPE 6](#_Toc116083971)

[1.12. ACADEMIC AND PRACTICAL RELEVANCE OF THE PROJECT 8](#_Toc116083972)

[1.12.1. Academic relevance 8](#_Toc116083973)

[1.12.2. Practical relevance 8](#_Toc116083974)

[1.14. STRUCTURE OF THE RESEARCH 9](#_Toc116083975)

[CHAPTER TWO 10](#_Toc116083976)

[2.1. REVIEW OF SIMILAR SYSTEMS 10](#_Toc116083977)

[2.2. CONCEPTTUAL SYSTEM DESIGN 11](#_Toc116083978)

[2.3. PROPOSED SYSTEM 11](#_Toc116083979)

[2.3.1. DEVELOPMENT TOOLS 12](#_Toc116083980)

[2.3.2. SOFTWARE FEATURES 12](#_Toc116083981)

[2.3.3. DESIRABLE FEATURES 13](#_Toc116083982)

[2.4. ARCHITECTURE OF THE PROSED SYSTEM 13](#_Toc116083983)

[2.5. COMPONENTS DESIGN 14](#_Toc116083984)

[2.5.1. The voter 14](#_Toc116083985)

[2.5.2. Registration Authority 15](#_Toc116083986)

[2.5.3. Board members 15](#_Toc116083987)

[2.5.4. Miner 15](#_Toc116083988)

[2.6. DEVELOPMENT TOOLS AND ENVIRONMENT 15](#_Toc116083989)

[2.7. CONSTRAINTS 16](#_Toc116083990)

[CHAPTER THREE 16](#_Toc116083991)

[3.1. REQUIREMENT SPECIFICATION OVERVIEW 16](#_Toc116083992)

[1.2. FUNCTIONAL AND NON-FUNCTION REQUIREMENT 17](#_Toc116083993)

[1.2.1. Functional requirement 17](#_Toc116083994)

[1.2.2. Non-Functional Requirement 18](#_Toc116083995)

[1.3. USER AND SYSTEM REQUIREMENT 18](#_Toc116083996)

[1.3.1. User Requirement 18](#_Toc116083997)

[1.3.2. System Requirement 19](#_Toc116083998)

[1.4. UML DIAGRAMS 20](#_Toc116083999)

[1.4.1. CLASS DIAGRAMS. 20](#_Toc116084000)

[1.4.2. USE CASE DIAGRAMS 22](#_Toc116084001)

[1.4.3. SEQUENCE DIAGRAMS 27](#_Toc116084002)

[CHAPTER FOUR 29](#_Toc116084003)

[4.1. METHODOLOGY 29](#_Toc116084004)

[4.1.1 BOEHM’S SPIRAL MODEL DESIGN 30](#_Toc116084005)

[4.1.2. WATERFALL MODEL 38](#_Toc116084006)

[4.1.3. INCREMENTAL MODEL 40](#_Toc116084007)

[4.2.1 USER INTERFACE DESIGN 41](#_Toc116084008)

[4.2.2 DATABASE DESIGN 42](#_Toc116084009)

[CHAPTER FIVE 44](#_Toc116084010)

[5.1. IMPLEMENTATION OF THE SYSTEM 44](#_Toc116084011)

[5.1.1. Setting Up a Node(s) 44](#_Toc116084012)

[5.1.2. Integrating A DApp On the Blockchain (E-VOTE APPLICATION) 45](#_Toc116084013)

[SYSTEM IMPLEMENTATION AND TESTING 49](#_Toc116084014)

[5.2. TESTING 49](#_Toc116084015)

[5.2.1. TESTING RESULT I 49](#_Toc116084016)

[5.2.2. TESTING RESULT II 49](#_Toc116084017)

[CONCLUSION 50](#_Toc116084018)

[REFERENCES 51](#_Toc116084019)

# DECLARATION

We hereby declare that, except for specific references which have been duly acknowledged, this work is the result of our own research and it has not been submitted either in part or whole for any other degree elsewhere.

We also declare that we have under supervision undertaken the work herein submitted

DATE NAME SIGNATURE

………………………….. …………………………………. ……………………………

DATE NAME SIGNATURE

………………………….. …………………………………. ……………………………

I have declared that I have supervised the work herein submitted and confirmed that the students have my permission to present it for assessment.

…………………………. …………………………………… ………………………………

DATE DR. EMMANUEL AHENE SIGNATURE

# DEDICATION

We will like to dedicate this work to Dr. Emmanuel Ahene for his immense support throughout the entire time in developing this project. He has been of great assistance.

# ABSTRACT

Modern culture is increasingly embracing the practice of online voting. It has a significant chance of lowering administrative expenses and raising participation rates. Voters can cast their ballots from any location with an Internet connection, doing away with the need to print ballots or set up polling places. Online voting solutions are nevertheless considered with great caution because they pose new risks despite these advantages. Voting manipulation on a massive scale may be possible due to a single vulnerability. When utilized in elections, electronic voting systems must be reliable, precise, secure, and practical. However, potential issues with computerized voting methods can restrict implementation. Blockchain technology was developed to address these problems, and it now provides decentralized nodes for electronic voting. Electronic voting systems are created using blockchain technology primarily because to the benefits of end-to-end verification. With distributed, non-repudiation, and security protection features, this technology is an excellent replacement for conventional electronic voting systems. The article that comes next provides an overview of blockchain-based electronic voting systems. The major purpose of this analysis was to assess the current status of blockchain-based voting research and online voting systems and any related issues to anticipate future advancements. This study offers an introduction to the basic structure and properties of the blockchain in relation to electronic voting as well as a conceptual description of the anticipated blockchain-based electronic voting application. This investigation led to the discovery that some of the problems now plaguing election systems may be resolved by blockchain technologies. On the other side, privacy protection and transaction speed are the problems with blockchain applications that are most frequently raised. The security of remote participation must be practical for a blockchain-based electronic voting system to be scalable, and transaction speed needs to be addressed. These issues led to the conclusion that the current frameworks needed to be modified before being applied to voting systems.

Keywords: electronic voting; security; blockchain-based electronic voting; privacy; blockchain technology; voting; trust

# ACKNOWLEDGEMENT

To God be the glory for seeing us through a successful completion of this research work and the whole BSc Computer Science course with the Kwame Nkrumah University of Science and Technology Kumasi. Our utmost appreciation goes to our supervisor Dr. Emmanuel Ahene for his relentless supervision, constructive criticisms, useful suggestions, and encouragements and not forgetting the entire lecturers of the Department of Computer Science.

Finally, our heartfelt thanks go to everyone who in one way or the other help us to complete this research work successfully.

# LIST OF FIGURES

[Figure5. 1 generating key for node 44](file:///C:\Users\IUPAC\Documents\project-doc-backups\FINAL-YEAR-PROJECT-DOCUMENTATION-bakup-1.docx#_Toc116087769)

[Figure5. 2 Login to as node 44](file:///C:\Users\IUPAC\Documents\project-doc-backups\FINAL-YEAR-PROJECT-DOCUMENTATION-bakup-1.docx#_Toc116087770)

[Figure5. 3 show connecting to another node 45](file:///C:\Users\IUPAC\Documents\project-doc-backups\FINAL-YEAR-PROJECT-DOCUMENTATION-bakup-1.docx#_Toc116087771)

[Figure5. 4 node transactions 45](file:///C:\Users\IUPAC\Documents\project-doc-backups\FINAL-YEAR-PROJECT-DOCUMENTATION-bakup-1.docx#_Toc116087772)

[Figure5. 5 node dashboard 45](file:///C:\Users\IUPAC\Documents\project-doc-backups\FINAL-YEAR-PROJECT-DOCUMENTATION-bakup-1.docx#_Toc116087773)

[Figure5. 6 RA view of the candidates sent by the board 46](file:///C:\Users\IUPAC\Documents\project-doc-backups\FINAL-YEAR-PROJECT-DOCUMENTATION-bakup-1.docx#_Toc116087774)

[Figure5. 7 RA dashboard overview 46](file:///C:\Users\IUPAC\Documents\project-doc-backups\FINAL-YEAR-PROJECT-DOCUMENTATION-bakup-1.docx#_Toc116087775)

[Figure5. 8 Board dashboard 47](file:///C:\Users\IUPAC\Documents\project-doc-backups\FINAL-YEAR-PROJECT-DOCUMENTATION-bakup-1.docx#_Toc116087776)

[Figure5. 9 Board dashboard 47](file:///C:\Users\IUPAC\Documents\project-doc-backups\FINAL-YEAR-PROJECT-DOCUMENTATION-bakup-1.docx#_Toc116087777)

[Figure5. 10 Board approved contract state as deployed by RA 47](file:///C:\Users\IUPAC\Documents\project-doc-backups\FINAL-YEAR-PROJECT-DOCUMENTATION-bakup-1.docx#_Toc116087778)

[Figure5. 11 Board interface overview 47](file:///C:\Users\IUPAC\Documents\project-doc-backups\FINAL-YEAR-PROJECT-DOCUMENTATION-bakup-1.docx#_Toc116087779)

[Figure5. 12 Voter tries to vote but election period is over 49](file:///C:\Users\IUPAC\Documents\project-doc-backups\FINAL-YEAR-PROJECT-DOCUMENTATION-bakup-1.docx#_Toc116087780)

[Figure5. 13 Voter select his choice of candidate to vote for. 49](file:///C:\Users\IUPAC\Documents\project-doc-backups\FINAL-YEAR-PROJECT-DOCUMENTATION-bakup-1.docx#_Toc116087781)

[Figure5. 14 Voter try to double vote 49](file:///C:\Users\IUPAC\Documents\project-doc-backups\FINAL-YEAR-PROJECT-DOCUMENTATION-bakup-1.docx#_Toc116087782)

[Figure5. 15 Voter login page 49](file:///C:\Users\IUPAC\Documents\project-doc-backups\FINAL-YEAR-PROJECT-DOCUMENTATION-bakup-1.docx#_Toc116087783)

[Figure5. 17 Results analysis from a mobile phone 50](file:///C:\Users\IUPAC\Documents\project-doc-backups\FINAL-YEAR-PROJECT-DOCUMENTATION-bakup-1.docx#_Toc116087784)

[Figure5. 16 Results analysis from desktop computer 50](file:///C:\Users\IUPAC\Documents\project-doc-backups\FINAL-YEAR-PROJECT-DOCUMENTATION-bakup-1.docx#_Toc116087785)

[Table 4. 1 database table for the RA 43](#_Toc116088903)

[Table 4. 2 database table for the Miner 43](#_Toc116088904)

[Table 4. 3 database table for the Board 44](file:///C:\Users\IUPAC\Documents\project-doc-backups\FINAL-YEAR-PROJECT-DOCUMENTATION-bakup-1.docx#_Toc116088905)

# CHAPTER ONE

# INTRODUCTION

## OVERVIEW

Not only is electoral integrity crucial for democracies, but it is also crucial for state voter confidence and responsibility. Methods of political voting are essential in this regard. From a political perspective, electronic voting technology can increase voter turnout, trust, and enthusiasm for the electoral process. Elections have long been a social issue due to their effectiveness as a democratic decision-making process. Citizens are becoming more conscious of the importance of the voting system as more people vote in actual elections (Liu, 2017). Judges decide who will represent them in political and business governance via the voting system. Voters use the democratic system to choose their representatives. (Racsko, Yaga, Mell, Roby, & Scarfone, 2019). The degree of trust that people have in the electoral process largely determines how effective such a method will be. A well-known trend is the development of legislative institutions to represent public opinion. These political entities include constituencies and student unions, among others. Voting has evolved into the main method for expressing the will of the populace by making a decision from among the options they provide. People's confidence in the selection made by majority vote increased as a result of the traditional or paper-based polling approach. It has aided in making the democratic procedure and election system valuable for choosing representatives and governments that are more democratic. Out of over 200 countries, 167 have democracy in 2018, the remainder are either fully defective or hybrid. (Economist, 2020), (Cullen & Houghton, 2000). Since the commencement of the voting system, the secret voting model has been utilized to increase trust in democratic institutions. It is crucial to prevent a decline in voting confidence. According to a recent study, the traditional voting process was not entirely hygienic, raising concerns about issues like fairness, equality, and the public's will. (Schinckus, 2020) measured and comprehended in the context of government (Shahzad & Crowcroft, 2019). Globally, engineers have developed innovative voting procedures that guarantee electoral integrity while providing some protection against corruption. The advent of technology led to the development of new computerized voting procedures. (Kim, et al., 2020), which are crucial and have presented serious difficulties for the democratic system. Compared to manual polling, electronic voting increases election dependability. Compared to the traditional voting process, it has improved the process' efficiency and integrity. (Hang & Kim, 2019). Electronic voting is frequently used in many choices due to its adaptability, ease of use, and low cost compared to conventional elections. (Chang, et al., 2020). Despite this, the inherent fairness, privacy, secrecy, anonymity, and transparency of the voting process are limited by the risk of excessive authority and manipulated facts. A issue for a transparent voting process in and of itself is that most procedures are now centralized, licensed by the crucial authority, managed, quantified, and monitored via an electronic voting system. The electronic voting protocols, on the other hand, have a single controller that manages the entire voting process. (Wang, Sun, He, Pang, & Lu, 2018). Due to the election commission's dishonesty, this process causes incorrect selections, which are challenging to correct using current techniques. Modern electronic voting methods can be utilized to go around the central authority via the decentralized network. A decentralized node for electronic or online voting is provided by blockchain technology. Due to the benefits of end-to-end verification, distributed ledger technologies like blockchain have recently been employed to create electronic voting systems (Ometov, et al., 2020). With characteristics like decentralization, non-repudiation, and security protection, blockchain offers an intriguing alternative to traditional electronic voting methods. Both boardroom and public votes are conducted there (Gao, Zheng, Guo, Jing, & Hu, 2019). A blockchain is a growing list of blocks connected by cryptographic connections, which at first was just a chain of blocks. A hash, date, and transaction information from the preceding block are all included in each block. The blockchain was developed to withstand data loss. In this field, researchers are attempting to take advantage of advantages like transparency, secrecy, and non-repudiation that are crucial for voting applications (Hakak, Khan, Gilkar, Imran, & Guizani, 2020). Voting is a new phase of blockchain technology. The use of blockchain for electronic voting applications has recently drawn attention to initiatives to use the technology to safeguard and correct elections. (Xiao, Wang, Wang, & Wang, 2019).

## PROBLEM DEFINITION

We have severe concerns regarding the widespread use of voting machines that only use electronic means to record and count votes and do not give voters with paper ballots that may be used to confirm their choices. We oppose using these devices to conduct our elections for three main reasons.;

* Software errors are unequivocally unavoidable.
* It becomes if not impossible but difficult to perform meaningful recounts.
* Offers greater opportunities for fraud to permeate the process.
* Inconveniences faced by voters having to be in long queues and staying at the polling centers for long hours.

Consider this scenario, a voter marks the appropriate candidates he or she wish to vote for, reviews and give command to cast the vote. Due to software problem the system records the ballot incorrectly or not at all. The voter leaves the boot and at the end of the day, the poll worker prints out the ballot images. The voter’s votes are incorrectly tallied and the printed ballot images is incorrect, but this error goes undetected because the voter is not there to view the print version.

The researchers, decided to solve these problems by building an electronic voting system based on blockchain, which provide security and transparency as it is based on cryptography and transparency.

## PROBLEM MOTIVATION

With regard to elections, blockchain technology addressed issues with the current system by making the voting process transparent and easy to use, preventing fraudulent voting, enhancing data security, and verifying the results. The adoption of the electronic voting process in the blockchain is crucial (Xiao, Wang, Wang, & Wang, 2019). Electronic voting does, however, come with some serious security issues, such as the possibility of vote manipulation and abuse if a voting system is infiltrated. Considering all of its potential benefits, electronic voting has not yet been widely used at the national level. Blockchain technology offers a practical way to get around the dangers of computerized voting today. The ability to vote is centralized in traditional voting systems. No one is aware of how to validate that document, so if someone wants to alter or change it, they can do it swiftly. Since the data are kept among numerous nodes, there is no one point of authority. All nodes cannot be compromised in order to alter the data. As a result, it is impossible to do away with the votes and effectively tally them with other nodes.

The blockchain is a digital, decentralized, encrypted, transparent ledger that can survive fraud and manipulation if the technology is applied properly. A Bitcoin electronic voting system lowers the dangers associated with electronic voting and enables tamper-proof voting due to the distributed structure of the blockchain. A fully distributed voting infrastructure is required for a blockchain-based electronic voting system. Blockchain-based electronic voting will only be successful in scenarios in which no single entity, not even the government, has complete control over the online voting system (Imperial, 2021). In conclusion, a strong conviction in the legitimacy of the power exercised by individuals in positions of authority is a prerequisite for free and fair elections. Voting can be made more effective in terms of administration and participation by using the literature review for this field of study and other similar experiments, which may be viewed as a useful starting point. But the concept of leveraging blockchain provided a fresh approach to electronic voting.

## AIMS OF THE PROJECT

The main aim of the project is to form a framework for a voting model to ensure transparency and fairness in electronic voting system.

Additionally, the researchers seek to achieve the following aims after a successful implementation of the system.

* Speed up election results declarations.
* Reduce cost of paying staffs to count ballots.
* Improved accessibility.
* Improve ballots integrity.
* Eliminate elections malpractices.
* Enhanced voters trust in electronic voting.

## OBJECTIVES OF THE PROJECT

The researchers after critically thinking, decided that a very robust and secure electronic voting system can be achieve only if the following objectives are supported by the system;

Decentralization: - Ensure that voting results and process is not centralized to only one particular authority.

Anonymity: - That the voters is completely protected and his right to security is preserved

Transparency: - To facilitate free and fair election devoid of any election malpractices.

Provenance: - Ensures traceability and verification of votes

Security: - Ensures that the system is secured and difficult to hack.

Hence, the researchers seek to achieve the above stated objectives by building electronic voting system based on blockchain. Since, blockchain is essentially a database that run on peer-to-peer network where encrypted blocks consisting of data are chained together by hashes. It is the best choice because it offers to us many of the aims of the system we intend to building.

* 1. JUSTIFICATION OF THE PROJECT

Significant changes to existing voting technologies may produce intended or unintended side-effects on election process, voter behaviour and participation.

Current voting processes be it electronic means or manual paper and ballot, are ridden with many security flaws, the issues are due to undeniably software errors, human activities and prone to hackers.

In order, to bring a nearly lasting solutions to these problems, we proposed an electronic voting technology that is built on blockchain. We believe blockchain is the best remedy to the many adverse behaviour of the current voting technologies.

Blockchain technology is enabled with decentralization, immutability, anonymity, transparency and cryptography.

Integrating these features in a voting system, improves fairness, trust, integrity and post-election rioting.

## PROJECT TERMINOLOGIES

Registration Authority (RA): - An entity or node that initiates the election process, oversees the eligibility of voters, registration of candidates, provision of digitally signed ballots to voters.

Board: - Nodes who together register their party, edit party details and sends the information to RA.

Voter: - These are node that perform the action of voting in the TruVote

Miner: - Nodes that update state of the chain which is maintained by the blockchain network, ensures that integrity of transaction before accepting it.

Blockchain: - Is an append-only data structure maintained by the miners. This consist of chain blocks which are cryptographically linked together to ensure immutability of the structure.

* 1. SCOPE OF THE PROJECT

The Scope of this work shall be limited to governmental and non-governmental organizations and social group with at least ten peoples and not exceeding hundred thousand people.

## PROJECT BENEFICIARIES

The project is intended to benefit Government of the Nation by reducing cost, ensuring trustworthiness in the election process and prevent.

## PROJECT SCOPE

This System is built as a decentralized peer – to – peer blockchain currency network to provide the features needed as a blockchain system, and then pushes it further as a platform for decentralized applications. However, due to the limitless existence of distributed applications, the scope of the project shall be limited to the implementation of blockchain based electronic-voting system.

As a result, the project shall cover the following;

Implementation of blockchain for use by every other decentralized application.

* 1. PROJECT LIMITATIONS

Although, this technology comes packed with security features suited for an electronic voting system, it is however, undeniably true that there are several limitations that prevent the technology from being fully implemented and explored. We enumerate these limitations below;

* Scalability and processing overheads

For a small number of users, the blockchain works well. However,

If the network is us for large -scale elections, the number of users will increase and the cost will increase.

Time consumption to consume transactions. Scalability issues have worsened due to an increase in the number of nodes in the blockchain network.

In the election situation, the scalability of the system is already an important issue. The integration of electronic voting also affects the scalability of the system based on the blockchain.

* User identity

Blockchain uses pseudonyms as usernames. This strategy does not provide  
Complete privacy and secrecy. Transactions are public, so user identities can  
discovered by examining and analyzing the blocks. Blockchain technology doesn't is not suitable for national elections.

* Immatureness

Blockchain is a revolutionary technology that symbolizes a complete shift to a decentralized network. It has the potential to revolutionize businesses in terms of strategy, structure, processes, and culture. The current implementation of blockchain is not without flaws. The technology is presently useless, and there is little public or professional understanding about it, making it impossible to evaluate its future potential. All present technical issues in blockchain adoption are usually caused by the technology’s immaturity [94].

* Energy efficiency

Blockchain incorporates energy-intensive processes such as protocols, consensus, peer-to-peer communication, and asymmetrical encryption. Appropriate energy-efficient consensus methods are a need for blockchain-based electronic voting.

* Acceptability

While blockchain excels at delivering accuracy and security, people’s confidence and trust are critical components of effective blockchain electronic voting [95]. The intricacy of blockchain may make it difficult for people to accept blockchain-based electronic voting, and it can be a significant barrier to ultimately adopting blockchain-based electronic voting in general public acceptance [96]. A big marketing campaign needed for this purpose to provide awareness to people about the benefits of blockchain voting systems, so that it will be easy for them to accept this new technology.

* Political Leaders’ Resistance

Central authorities, such as election authorities and government agencies, will be shifted away from electronic voting based on blockchain. As a result, political leaders who have profited from the existing election process are likely to oppose the technology because blockchain will empower social resistance through decentralized autonomous organizations.

## ACADEMIC AND PRACTICAL RELEVANCE OF THE PROJECT

In this section we describe the project relevance at two level.

* The academic aspect
* The practical aspect

### Academic relevance

Blockchain is a relatively new technology and still in the revolutionary state. The researchers seek to use this project to achieve the following academic relevance.

1. Introduce the concept of blockchain to students: - Because blockchain is a new technology it lacks mass adoption, acceptance and use. We bring the technology to create the awareness of students about the security advantages that can be leverage by the adoption of blockchain not only for voting but also, in other applications that require maximum security to operate.
2. Create the awareness for more research in the technology: - Not only to introduce students to the technology, but also, to bring to light the need for more research on this technology.

Generally, blockchain technology has not receive widespread research and hence the technology is at its infancy, making it inefficient in implementation.

### Practical relevance

The practical relevance of this work is to reduce the extend of irregularities and fraud to which public elections are usually marred with by leveraging the securities features of blockchain network.

* 1. PROJECT ACTIVITY AND PLANNING SCHEDULES.

In each stage of the project, we outline various activities that are to be completed according defined times lines set against each activity.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Week1** | **Week2** | **Week3** | **Week4** | **Week5** | **Week6** | **Week7** | **Week8** | **Week9** | **Week10** | **Week11** | **Week12** |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |

Table1. Project Schedule

## STRUCTURE OF THE RESEARCH

This report is structured to ease understanding and an easy walkthrough of the project.

Chapter one of this project, introduce the purpose of the study, problem statement, aims, specific objectives, justification, motivation, scope of the project, beneficiaries, academic and practical relevance of the project, the project schedules and deliverables.

Chapter two talks about literature review, here, we review related system that are aid us in understanding and building of our system and duly acknowledge them. Then we discussed our proposed system, the development tools, software features, desirable feature and constraints.

In chapter three, the requirements of the system are enumerated, we discuss the functional, non- functional, system and user requirements of the system.

Then we discuss the various UML diagram after the requirements of the system, we intent to give the reader a complete mental view of the system.

We take the reader, the various methods employed in developing the system, we discuss the Boehm’s spiral model, waterfall and incremental software developmental processes.

Chapter five is about implementation, testing, conclusion and references.

# CHAPTER TWO

## REVIEW OF SIMILAR SYSTEMS

This chapter reviews similar systems from which the researcher ideas stems-up, it looks at related literature, the proposed system, development tools, features and constraints.

1. (S Sekar, 2020) proposed a decentralized e-voting system using blockchain for elections as shown in the system consist of three modules, User validation using biometric information that is hashed by - Message-digest version 5 (MD5) algorithm to verify the user. Dynamic ballot loading relied on the residence location of citizens and loaded in the ballot. After casting their vote in which a vote ID is given as acknowledgment to the voter. The voters must transfer their public key to the Election Authority (EA) that is charge of creating the eligible votes and paying the vote fees for the address of bitcoin that is created automatically within the backend. Also, results are published by the EA that has its own bitcoin address. The voter should register in Registration Authority (RA), also the candidate should register in RA with his information in order to create an ID for that candidate. The voting fees for the address of bitcoin for the voters is zero as soon as the voter cast their vote, so there is no chance to repeat a vote more than once.
2. Decentralized, Transparent, Trustless Voting on the Ethereum Blockchain: (Aicha Fatrah) in his paper entitled “Decentralized, Transparent, Trustless Voting on the Ethereum Blockchain” discusses two types of ongoing issues with E-Voting solutions. First, the capability of anyone to tally the results from the smart contract before having all the votes casted, and second, the anonymity of the votes since public keys can be associated with the recorded votes. In this paper, the author presents the implementation of a voting system as a smart contract running on Ethereum that uses threshold keys and linkable ring signatures. Nevertheless, this solution again includes a registration phase, and voters rely on a Centralized Authority to register their public key for casting a vote.
3. (Fatrah, 2019), present the first implementation of a decentralized and self-tallying internet voting protocol. This system is able to retain maximum voting privacy by using the Ethereum Blockchain. Initially crafted by Kiayias and Yung, and later improved by Hao et al., this protocol pioneered the needlessness for a trusted authority in the processes of calculating the tally and protecting the anonymity of a voter. The Open Vote Network is a self-tallying protocol in which the privacy of a vote is controlled by its corresponding voter. Thus, it follows that only a full collusion amongst all the voters would allow for a single vote to be exposed. The execution of this protocol is enforced by the same consensus mechanism that safeguards the Ethereum Blockchain. Their Schnorr non-interactive ZKP and 1 out of 2 ZKP functions to create and verify ZKPs on an Ethereum smart contract have been adopted by this work.

## CONCEPTTUAL SYSTEM DESIGN

The conceptual framework as seen in the figure below is divided into two (2) layers which are the front end and back end that was also adapted.

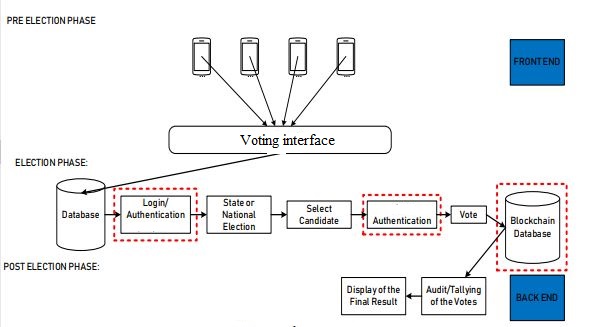


Figure 2. 1 Conceptual view of the proposed system

## PROPOSED SYSTEM

We proposed a blockchain-based e-voting system, which meets the essential requirement of e – voting process. All votes in the blockchain cryptographically linked block by block.

The voter can vote for any candidate in accordance with the list of candidate present.

The blockchain-based e-voting system can be applied to a variety of voting situation and other distributed application.

Although, blockchain is a secure technology, it uses ECC public key cryptography, which is not secure to quantum computer attacks, thus, blockchain with countermeasures to quantum computer attacks is a future research topic in this area.

### DEVELOPMENT TOOLS

Because the system is intended to be used to elect leaders who will be entrusted with the right and mandate to make decisions that can affect the day-to-day operation of an entire group, society or country, and invariably has strong baring to the living condition of the citizens, we carefully, selected more recent and tested tools that integrates with the current technological innovation.

Among the various programming languages that exit, we decided to use python for the backend and the blockchain network. Currently python ranks among to top ten most popular and fastest-growing language. It is interpreted, high level, general-purpose, and object-oriented scripting language, which makes easier to learn and faster in execution.

On the front end, we use HTML, CSS, JavaScript, these are by far the most well-known front-end languages that integrate across browsers and will give minimal display error with older browsers.

For instance, approximately 96.34 percent of browsers support the new features of JavaScript ECMAScript 6 with features including Promises, Modules, Classes, Template literals, Arrow function, Let and Const, Default Parameters, Generators, Destructuring, Assignment, Rest and Spread, Map/Sets and WeakMap/WeakSet and many more. Whiles over 97.09 percent of browsers support current features of HTML and CSS.

### SOFTWARE FEATURES

The system was build taking advantages of the many security features blockchain technology provides. We clearly designed the system to achieve the following features.

### DESIRABLE FEATURES

**Security and transparency**: - the ballots are maintained in a blockchain-wide distributed ballot ledger. Because our proposed blockchain is open to the public, anyone can join a peer-to-peer network to keep up with the blockchain’s data. Any effort to modify the ballot ledger without the consent of all node administrators is detectable and impossible because the chain is public and synchronized throughout the network

Any interested party can check the outcome in real time using a public blockchain ledger, removing the possibility result manipulation and computation.

**Elimination of ballot staffing**: - In the electoral process, ballot staffing is a major concern. In our suggested approach, each voter goes through a verification procedure in which their information is compared to that of voters in the database. All of the IDs that have been confirmed and successfully cast a vote are stored in the database. When the database receives a new verification request, it checks to see if the requested ID has been tagged. As a result, any attempt to vote multiple times from the same ID will be rejected. As a result, the suggested methodology eliminates the need for ballot workers.

**Decreasing election cost**: - The proposed strategy involves fewer workers and removes intermediaries and papers, lowering election costs dramatically. Also, reduce some unavoidable election expenditures, such as documents printing and transportation

**Protect the environment:** - The recommended architecture will add to the protection of our ecosystem. The entire process is digital, and we will be able to preserve some trees from being cut. Since the process purely paperless and does not require any printing of ballot papers.

**Reduce political confusing:** - The proposed model ensures decentralization. As a result, no political party has any influence over the electoral process. Because the entire voting process is public and auditable, there is no risk of political chaos because of a rigged election.

## ARCHITECTURE OF THE PROSED SYSTEM

The system composed of four components, which collectively linked together to achieve the objectives of the project.

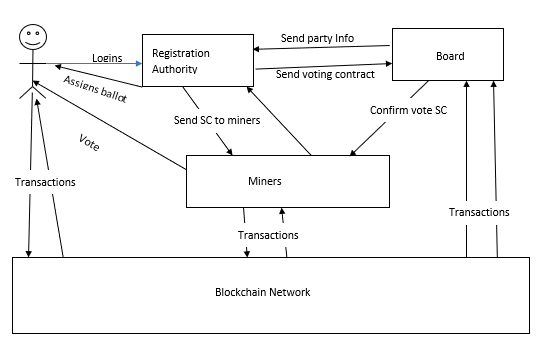
The architecture of the system is built and gives a general overview of the organization of the system. The figure below shows the architectural design of the proposed system. 

Figure 2. 2 Architectural view of the proposed system

## COMPONENTS DESIGN

The system shall have the following components based on the architectural design;

* Voter
* Registration authority
* Board member
* Miners

### The voter

Based on the design of the system, the voter class will be able to initiate transaction with the blockchain network directly, they can will also need to login in order to vote.

In logging in the voter first interact with the voter database, a container with list of eligible voters, who are part of an organization and can be identified with unique username and password.

In order to prevent double voting and ballot forgery, the authentication process goes through the registration authority, this process authenticate the unique verifies the unique id of the voter and also decrypted the request message using the public key of the voter before an encrypted ballot is sent to the voter.

On receipt of a ballot the public key of the registration authority available to the voter is used to decrypted the message and the ballot information is then displayed for the voter the vote.

### Registration Authority

The registration authority, control the most of the election processes. This class set authenticate valid users and prevent ballot forgery as well as double voting. The following are function of the registration authority

* Authenticate valid users
* Prevent double voting
* Generate and send ballot to valid voters
* Configure election and deploy the election contract
* Initiate the start of the of the election

### Board members

This class is associated with consensus, the board members are upload the party information of their respective parties and send the information to the registration authority.

The message send the RA is encrypted using RSA, this is because the board knows the public key of the RA and vice versa.

The board also, approves and the election contract deployed by the RA to start the election by double checking if the information provided by the RA is correct before voting begin.

In this way each party is represented and agrees to the state of the election contract.

### Miner

The miner class perform a number of activities including;

* Updating the state of the chain on the blockchain network
* Send and/or receive transaction
* Solves complex algorithm in order to update the state of the blockchain

## DEVELOPMENT TOOLS AND ENVIRONMENT

The system will build by using the following tools;

* Font-end development languages for web thus Html, CSS and JavaScript
* Back-end will be build using python and flask

The environment for this purpose is Visual Studio Code editor

Python is a very popular language, used in building application ranging from mobile to web. It has a set of rich libraries that ease development process and supported and improved by very wide range of programmers worldwide.

Visual Studio Code editor is most suitable for this purpose because it support various file extension, .py, .ipny, .c, etc. this enables us to build both the font end and the backend using the same code editor.

Secondly, it wide range of extension that make development process easy with the help of intellisence, the IDE is just right for this development.

## CONSTRAINTS

Whiles our proposed system provided the takes advantages of the many security features offered by blockchain technology such as integrity, anonymity and non-repudiation which are critical for a voting application. It comes with some constraints we outline below.

Performance level with respect to attributes including time required to download mine to min a transaction into the block of the longest blockchain, processing rate for number of transactions, time required to download and run full copy of the system from scratch in order to participate in the process of transactions validation has been relatively slow.

Also, the throughput of the system does not only depend upon the capacity of infrastructure such as hashing power and memory but also on the transaction being mined.

For our system, proposed system large number of transactions are expected to be recorded in the blockchain concurrently. Therefore, if the rate of incoming transaction to the unconfirmed pool of transactions does not match the rate of confirmation of transaction to the blocks by the miners. Hence significant overhead as well as delays in transaction confirmation time are expected.

Lastly, an in-depth investigation is required to identify and assess challenges with respect to scalability for wider application domains.

# CHAPTER THREE

## REQUIREMENT SPECIFICATION OVERVIEW

The system will have the properties of ballot privacy, individual verifiability, eligibility, anonymity, non-repudiation and immutability.

In general, the system allows voters and candidates to be registered and take part in an election process. The voter can vote in the system and anyone can verify the result of voting. The administration can be broken into two authorities, the Registration authority and the board members.

## FUNCTIONAL AND NON-FUNCTION REQUIREMENT

The system shall have the following function and non-functional requirements.

### Functional requirement

Req 1.1 - The system must be composed of Miners, Registration authority, Board members and voters.

* + - 1. Registration Authority.

Req 1.1: - The system must ensure that only eligible voters take part of the election.

Req 1.2: - The system must make sure each board members provides his/her list of candidates.

Req 1.3: - The system should be able to initiate the election process and specify the start and end time of the election process.

Req 1.4: - The system must be able to provide a signed digital ballot for voters.

* + - 1. Voter

Req 2.1: - The system must allow eligible voters to be able to login and access election information.

Req 2.2: - The system must require that voter’s login with their credentials.

Req 2.3: - The system must be able to display real-time voting results on request.

Req 2.3: - The system should only allow access to the election ballot after verification of the voters’ identity.

Req 2.6: - The system should allow access to the election ballot within the election time period only.

#### Board Member

Req 5.1. The system must be able to accept the information of the party’s candidates.

Req 5.2. The system must be able to approve the election contract deployed by the RA.

Req 5.3. The system must be able to view the election result after the election process.

Req 5.4. The system must submit information of its candidates and its public key to the RA.

### Non-Functional Requirement

Req 6.1. The response time of voting page must be less than 300ms.

Req 6.2 The response time of public API must be less than 300ms.

Req 6.3. The response time of voting page must be less than 300ms.

Req 6.4. The system could use third party API to make the transaction on the blockchain.

Req 6.5. The tallying system should be stable enough to show the correct result.

Req 6.6. The voting system should be stable enough to broadcast the commitment to the blockchain.

Req 6.7. The system should use a relational database to store the information of the voting item, Bitcoin addresses candidates and voter.

Req 6.8. The system should be tested with the different browser and OS system.

Req 6.9. The user interfaces should provide friendly user experience.

## USER AND SYSTEM REQUIREMENT

The system among other user and system requirements is expected to satisfy the following use and system requirements.

### User Requirement

Req 1.1. User must be able to log into the voting page anytime to check voting results as long as the voting is contract is on the network.

Req 1.2. Users can make transactions using their various system.

Req 1.3. System should be able to determine voters / users’ eligibility.

Req 1.4. Should ensure that each voter is able to cast vote only ones.

Req 1.6. Voters should not be able to access partial results in real-time before the voting ends.

Req 1.7. The System should be able to uphold voters’ anonymity.

Req 1.8. Users accessing the system as voters shall not be able to participate in mining transactions

Req 1.9. Users must be able to verify election outcome once the election contract expires.

### System Requirement

Req 2.1. The System should provide a voting page that will enable valid users to log into the system and cast their votes only if voting contract is started.

Req 2.2. The System shall be assigned to each voter a ballot and display the list of registered aspirants on the voting page to enable user make their choice on who to vote for.

Req 2.3. The System shall make the electoral public by distributing the voting contract to every node participating in mining the votes casts.

Req 2.4. The system shall provide to the use based on query the voting result either on vote contract on, before or after vote contract expiration.

Req 2.5. The system shall not restrict the number of miners at whatsoever.

Req 2.6. The system shall provide for miners the ability to send transactions to specific individual on the network.

Req 2.7. The system shall assign signed ballot to each user to prevent double voting.

Req 2.8. The System shall update it database in real-time the status of valid voters who have casted their votes.

Req 2.9. The System shall provide the list of aspirants who are candidate in the voting page with option to vote for any candidate of voters’ choice.

Req 2.10. The System shall ensure that for any portfolio only a single selection is allowed.

Req 2.11. The system shall deactivate the option of selecting another candidate once a selection has already been made.

Req 2.12. The system shall query the transaction and display the current state of the voting contract on user request.

Req2.13. The system shall only accept votes in the validity period of the voting contract.

Req 2.14. Voting contract shall expire only after the time limit set for the voting exercise by the registration authority has expired.

Req 2.15. The system shall only run the contract within the time period specified by the registration authority.

Req 2.16. The system shall accept any other distributed application to interface with it.

Req2.17. The system shall provide the back bone on which other distributed application shall run.

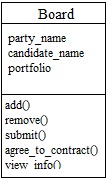
## UML DIAGRAMS

For the sake of clarity and widespread understanding of the system modelled diagrams, we employ the standards of the Unified Modelling Language.

All diagrams here, are drawn in accordance to the standards implemented in the Unified Modelling Language (UML).

### CLASS DIAGRAMS.

We model the class diagram starting from the very discrete level and gradually built upon each object and their relationship to finally achieve the complete model of the whole system class diagram.

Figure 3. 1 UML diagram for board Figure 3. 2 UML diagram for voter class

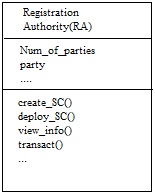
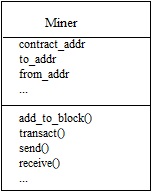


Figure 3. 3 UML class diagram for RA Figure 3. 4 UML class diagram for miner

Registration Authority

Voter

\*

1

Figure 3. 5 Relationship between the voter and the RA

Figure 3. 6 Relation between the board and RA

\*

1

Registration Authority

Board

### USE CASE DIAGRAMS

We try to model the user interactions with the system in this case. We use the standard UML modelling to graphically depicts the user interactions with the system at various operational stages.

View transactions

**Registration Authority**

Create Smart Contract

Register parties

Deploy Smart contract

Assigned ballot

Submit vote

View transactions

Submit ballots

Request for ballots

**VOTER**

Figure 3. 7 use case showing RA actions

Figure 3. 8 Use case showing the role of the voter

View transactions

Confirm contract status

transact

Submit party details

Edit party details

**Board**

Add Party

View transactions

Transact

Vote

Update chain

Mine

View chain status

**Miner**

Figure 3. 9 Use case depicting the role of the board party member

Figure 3. 10 Use case for the board

Voter

RA

Send ballot

Request ballot

Send PK

Submit ballot

Figure 3. 11 use case for voter interaction with registration authority

View transactions

Transaction

Relay PK

Board

Miner

Figure 3. 12 use case for miner and the board

View transaction

Deploys SC

Veify SC

Create SC

Submit party info

Edit party details

PK Relay

board

RA

Figure 3. 13 use case for board and registration authority relationship

3.4.3 ACTIVITY DIAGRAM

We model the activities each user can perform using the system. We use the standard UML modelling to graphically depicts depict these various scenarios.

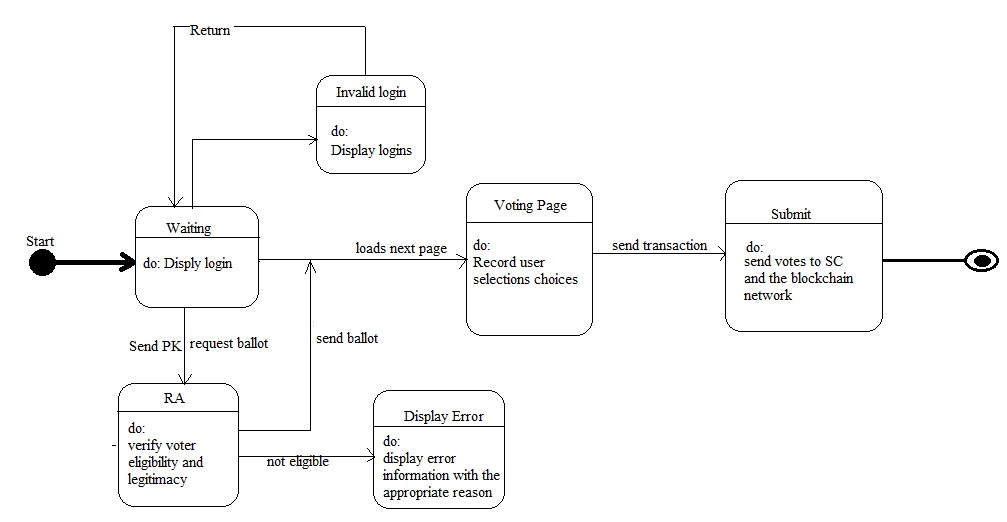


Figure 3. 14 show an activity diagram for a voter

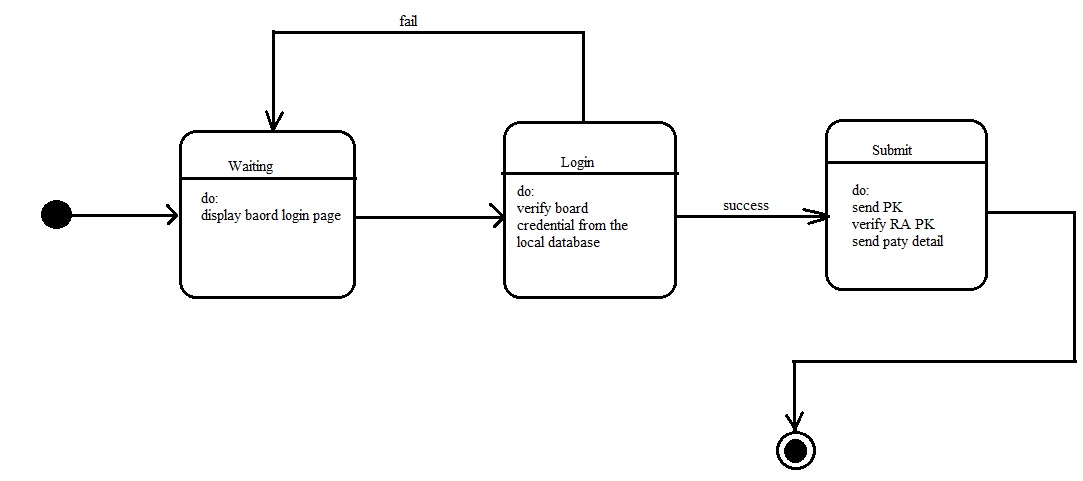


Figure 3. 15 shows the login activity diagram for the board

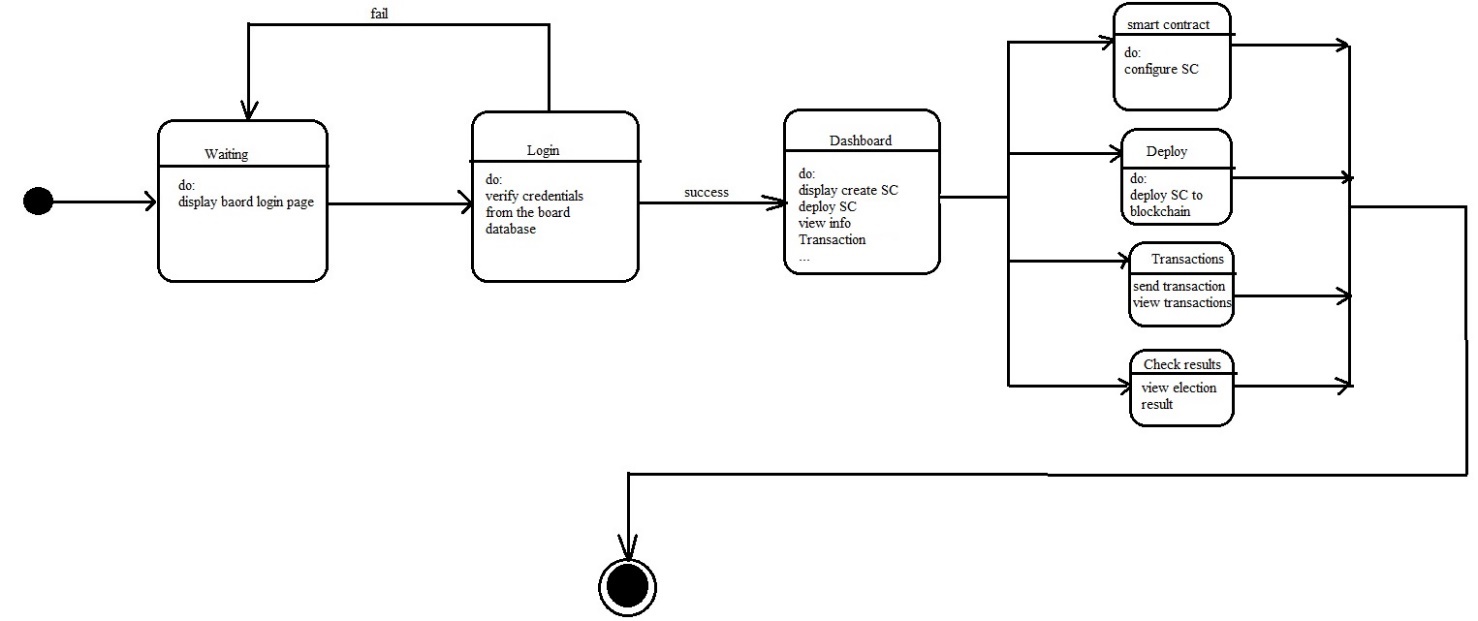


Figure 3. 16 shows the login activity diagram of the miner.

### SEQUENCE DIAGRAMS

We decided to use sequence diagrams to model the interaction and exchange of messages among the various unit class. This we have chosen is to provide a good visualization the system activities in its runtime scenarios.

We intend to paint, in the reader’s mind eye, how the system will behave and help him/her discover the responsibilities a class may have in executing each section of the system and how it may be modelled from the readers’ point of view.

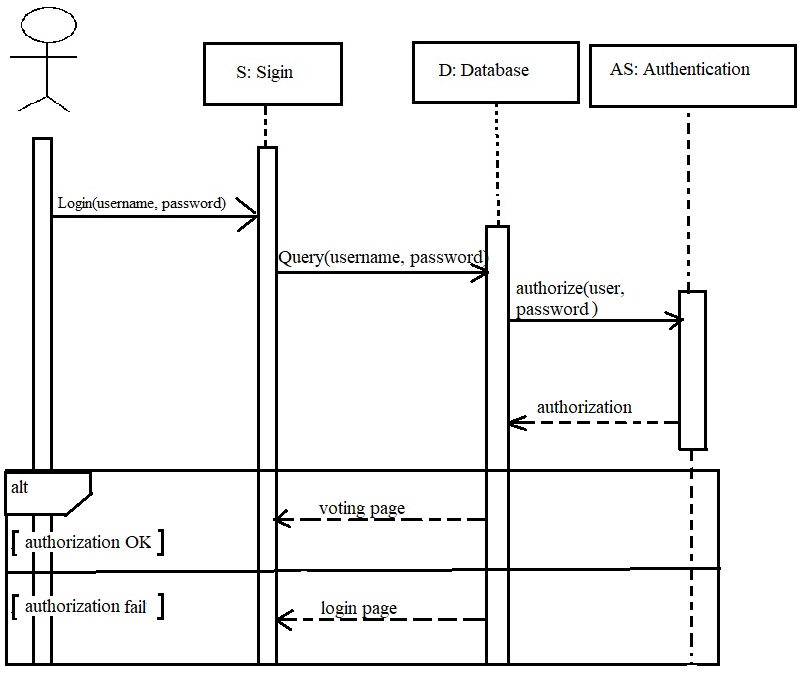


Figure 3. 17 Shows the sequence diagram for the voter login process

Voter

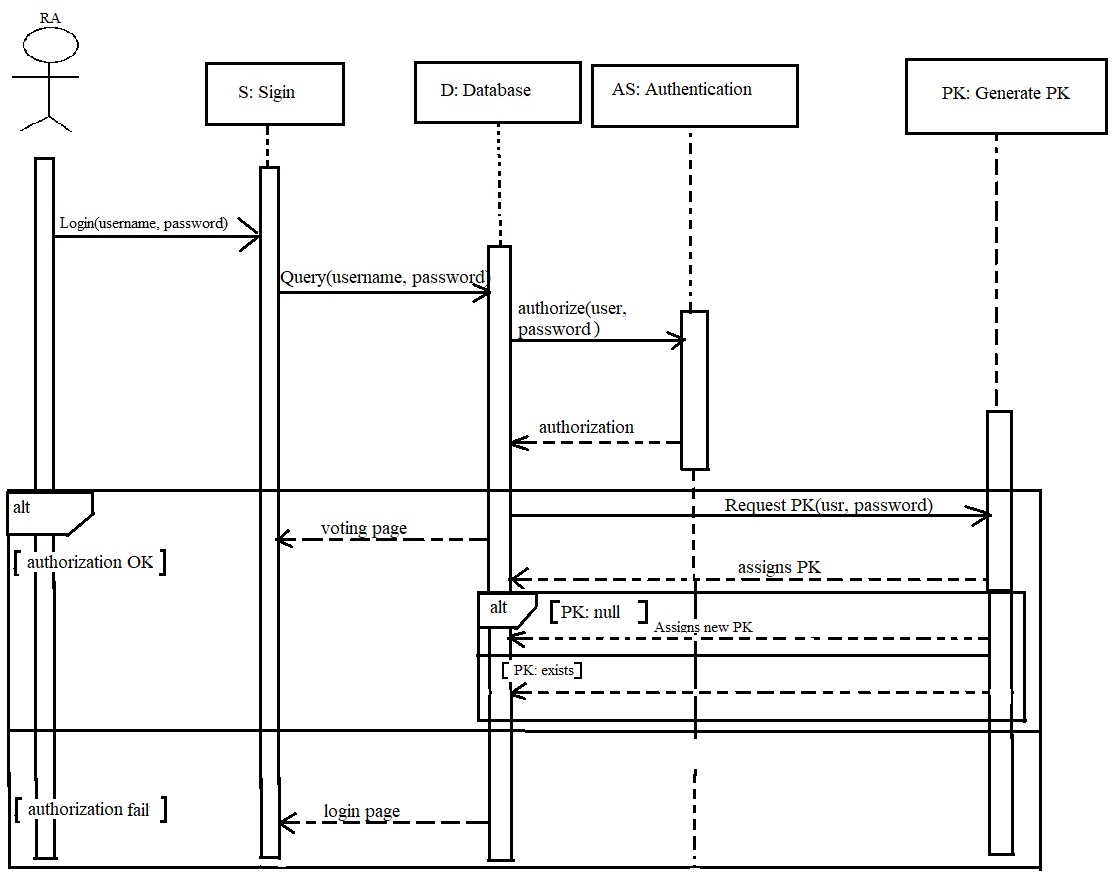


Figure 3. 18 RA sequence diagram for miner login

# CHAPTER FOUR

## METHODOLOGY

This study originates from a need to design a more secure and practical e-voting system, since it has become a popular topic in the area of industry and information security. Blockchain is based on distributed ledger technology and invented by Satoshi Nakamoto in 2008. Blockchain is a growing list of blocks. Each block except the first block stores its previous block’s hash value. It synchronizes the ledgers replicated among multiple nodes by using community validation, which is adopted to serve as the public transaction ledger of the crypto-currency Bitcoin.

We present techniques to exploit blockchain to improve the security of e-voting. Compared with the original blockchain, the improvements are as follows:

1. We design a synchronized model of voting records based on DLT to avoid forgery of votes.

2. We design a user credential model based on SHA256 to provide for authentication and non-repudiation.

3. We design an authority system, we labeled Registration Authority, responsible for elections management before voting starts

4. We design an entity labeled Board, responsible for providing details of their various parties to the Registration Authority and also verifying the RA deployed contract before voting begins.

5. The miners also a separate entity consists of individual nodes on the network and add transactions to the blockchain. The transactions contain the state of the contract at any particular moment.

Due to the distributed nature of the system, we reviewed the system based on the following software development methodologies in order to the readers understanding of the project.

* Boehm’s spiral model
* Waterfall model design
* Incremental model design

### 4.1.1 BOEHM’S SPIRAL MODEL DESIGN

We step through the system development by carefully crafting a suitable activity at each stage of the development life cycle.

In accordance with Boehm’s spiral design model, we perceived the system development by stepping through the following stages.

#### 4.1.1.1 Determination of objectives.

Although stating the exact requirement for a software of this kind from start is however impossible, we have carefully reviewed other system in chapter two of this work, in order to clearly define the objectives our system. The system will be of no use if we replicate an existing system without any additional functionality or improvement.

The review activities include searching online, questionnaire and interviews.

##### **Searching online**

The first step to seeking information on this very topic was doing a google search on already existing electronic voting system based on blockchain. The functionality they offer, the type of security, robustness and availability.

##### **Questionnaire**

Because the system is intended to be deployed on the local Ghanaian system, we design questionnaire and to learn how people opinion on electronic voting and comfortable are likely to be in using the system.

The data was amazing, we found that individuals between the ages of 16 – 45 were the most respondents who seems to be very comfortable with electronic voting.

##### **Interviews**

During the course of reviewing various system, we had interactions with other informed lecturers to seek their views concerning our proposed system.

#### 4.1.1.2. dentification and resolving risks

Our aim is to develop a robust system by trial and error, as such we begin with implementing various prototype of the system. In order to have a working implementation of the system, we first test using a quickly develop prototypes.

##### Prototype 1 (Rapid prototyping/ throwaway)

In this model, we model the following functionality of the system. We choose this approach to ensure that we validated the system functionalities and reduce cost in the long run, various protype were developed to arrive at the operational prototype. We choose to describe a detailed process method of the initial prototype here.

As error in the later phase of the development process is most daunting and cost effective.

* **Voter login page:** - The page shall be simple and provides inputs field of username and password. This is to authenticate various users. The user credentials are check against a database to ensure that the user is who he/ says he is.

For the purpose of the voting process an ID is generated using SHA256 algorithm automatically, to uniquely identify the voter in the system, this cannot be change but the voter can decide to change his or her username or password.

Along with the ID are two keys public and private keys using RSA which are used to send, receivedand signed transactions on the blockchain network.

* **Registration Authority**: - shall consists of the following functionalities;

1. Login page: contains an input field for username and password, which are check against a local database first. The user is automatically directed to generate key pair using the RSA algorithm, if that is his first-time login in else, he must have generated them already and in this case, he is redirected to the dashboard.
2. Configure election: - basically set up election time and deploys the contract for voting to start.
3. Organize the details send to it by the various party board members.
4. Assigned two addresses, one for transaction on the network and the other for indicating that the deployed contract is from him. So that the other board members can approved and give their consent for voting to start.

* Board: - shall consist of party executive who collectively define their party parameters and send to the registration authority. Functions the board include;

1. Add portfolio
2. Add candidates
3. Edit party details
4. Approved deployed contract from the RA upon being satisfied with the state.

* Miners: - Are powerful nodes that solve complex algorithm in order to add transactions to the blockchain. These transactions consist of the state of the deployed contract by the RA.

The functionalities of these node shall include;

1. Login
2. Mine
3. Send / receive transaction
4. Have public and private keys for both the electronic voting and validating transactions.

#### Development and test

The development process was carefully caried out through incremental approach. We develop each version of the software in small bits until the whole process was complete. This phase was divided into the following activities

* + **Requirement**

In collection the requirement we allocated a maximum of one week for this activity since, it involves seeking opinion from people. Some people become uncomfortable when they are approach with certain type of questions.

We collected the requirement using the procedures outline in requirement plan.

* + **Draft**

From the prototyping in we draft a more robust and detailed system. This activity also took one week to complete, since it involving planning and carefully thinking in order to avoid any loophole that may exist in the final deployment of the system.

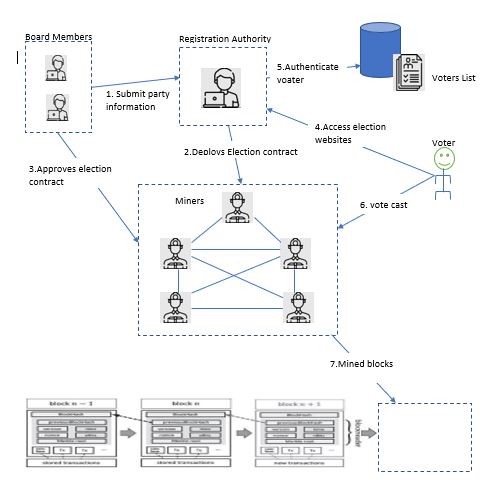


Figure 4. 1 Network diagram for the system

* + Detailed design

From the prototyping, the detailed designed took another one week to complete. The following designed were considered.

1. User interface planning
2. Backend designing
3. Database design
4. Blockchain network design

Each of the design has been gone through series of testing and prototyping before finalizing on a particular design.

Because, the software development process method was iterative, user feedback for each stage was paramount and was integrated into the design.

* + Coding

The next three to four weeks, was dedicated to coding the project. We worked on each section of the code independently.

The language choice was python, we choose this because python has large support library base and is currently one of the programming languages of interest that can be used to implement almost everything right from web application to mobile application.

The coding was divided into five stages;

1. The voting class consists of other functions such as the login function, Key-gen function, ID-gen function, submitting-function.

Below is a code snippet from VS Code for the login routine

def login():

    if request.method == 'GET':

        return render\_template('login.html')

    else:

        uname = request.form.get('username')

        pword = request.form.get('password')

                response = requests.post('http://127.0.0.1:7000/user-login', json={'username': uname, 'password': pword})

        data = response.json()

        if data['status'] == True:

            if 'message' not in data:

                response = requests.post('http://127.0.0.1:7000/user-get-info', json={'id': data['id']}).json()

                if response['status'] == True:

                    if 'message' not in response:

                        keys = gen\_key()

                        session['public\_key'] = keys['public\_key']

                        session['private\_key'] = keys['private\_key']

                        global voter\_img

                        for x in response['election\_info']['candidates']:

                            voter\_img[x['name'].replace(' ', '')] = x['img'] if x['img'] != "" or x['img'] != None else ""

                            x['img'] = ''

                        session['ELECTION\_INFO'] = response['election\_info']

                        session['ELECTION\_ADDR'] = response['election\_addr']

                        session['MINER\_NODES'] = response['miner\_nodes']

                        session['SIGNATURE'] = response['signature']

                        session['uid'] = response['user\_id']

                        session['constituency'] = response['constituency']

                        session['isLoggedIn'] = True

                        session['uname'] = uname

                        global hasVoted

                        g.hasVoted = True

                        return redirect('/index')

                    else:

                        return redirect('/voted')

            else:

                return redirect('/voted')

        else:

            t\_data = {'message': data['message']}

            return render\_template('not-eligible.html', data=t\_data)

1. The Registration Authority: - Also consists of other sub routines, such
   * + - Configuring the election
       - Deploying the contract to start the election

Below is a code snippet from VS Code for the login to generate key pair

* def generate():
* if 'u\_name' in session:
* uname = session['u\_name']
* pword = session['password']
* return\_data = db.get\_registrar((uname, pword))
* if len(return\_data) != 0:
* return\_data = return\_data[0]
* if (return\_data[4] == None or return\_data[4] == "") and (return\_data[5] == None or return\_data[5] == ""):
* keys = generate\_key()
* session['private\_key'] = keys['private\_key']
* session['public\_key'] = keys['public\_key']
* status = db.update\_registrar\_info((keys['public\_key'], keys['private\_key'], session['u\_name'], session['password']))
* if status:
* return jsonify({
* 'private\_key': format\_key\_for\_api(keys['private\_key'], 'priv'),
* 'public\_key': format\_key\_for\_api(keys['public\_key'])
* })
* else:
* return jsonify({'message': 'Error generating keys!!'})
* else:
* return jsonify({
* 'private\_key': format\_key\_for\_api(return\_data[4], 'priv'),
* 'public\_key': format\_key\_for\_api(return\_data[5]})
* else:
* return redirect('/register')
* else:
* return redirect('/login')

1. Board class: - handles the following routines
   * + - Adding party details
       - Submitting party details to RA
       - Agreeing to the state of the contract as deployed by the RA.

The code below is the login routine snippet from vs code for the board

def login():

    if request.method == 'GET':

        return render\_template('login.html')

    else:

        uname = request.form.get('username')

        pword = request.form.get('password')

        return\_data = db.signin((uname, pword))

        if len(return\_data) != 0:

            return\_data = return\_data[0]

        if len(return\_data) != 0:

            session['name'] = return\_data[0]

            session['email'] = return\_data[1]

            session['u\_name'] = return\_data[2]

            session['password'] = return\_data[3]

            session['private\_key'] = return\_data[4]

            session['public\_key'] = return\_data[5]

            if session['public\_key'] == None or session['public\_key'] == "":

                return redirect('/key-generate')

            else:

                return redirect('/home')

1. Miner class: - handles the following sub-routines
   * + - Updating the state of the chain in the blockchain network
       - Sending and receiving transaction

Below is a code snippet of the mining routing from vs code editor

def start\_miner(self, addr, send\_nodes):

        # Start a mining process on a new thread since

        self.miner\_thread = ThreadWithReturnValue(target=miner.mine, name='MinerThread', args=(self,))

        self.miner\_thread.start()

        mined\_block = self.miner\_thread.join()

        if mined\_block != None:

            mining\_reward\_tx = {

                'from\_addr': None,

                'to\_addr': addr,

                'value': self.reward,

                'gas': 0,

                'args': [],

                'timestamp': time()

            self.add\_transaction(mining\_reward\_tx, send\_nodes)

        return mined\_block

1. The Blockchain network class: - this class handles the connection among nodes in the network and allows for transparency, distribution, and communication among nodes.

Below shows a snippet from vs code for the genesis block routine

def create\_genesis\_block(self):

        t = dt.timestamp(dt(2022, 1, 1))

        block = Block(0, t, [self.initial\_transaction], '', self.difficulty)

        db.add\_to\_mined\_tx(0, self.initial\_transaction)

        self.transactions = list(filter(lambda tx: tx.tx\_hash != self.initial\_transaction.tx\_hash, self.transactions))

        block.prev\_hash = '0' \* 64

        block.mine\_block(self.difficulty)

        block.set\_block()

        self.chain.append(block)

        db.add\_block(block)

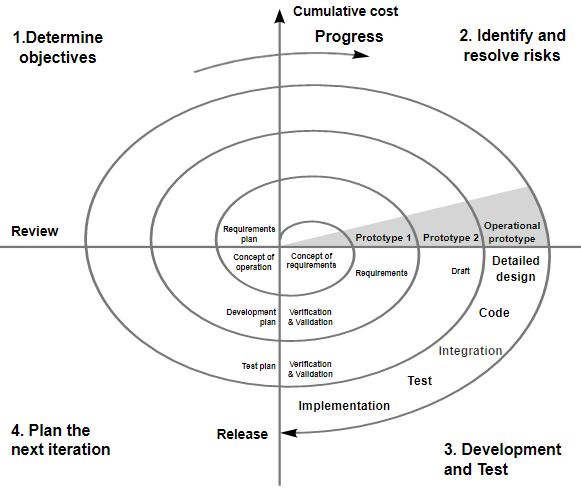


Figure 4. 2 shows an overview of the Boehm’s Spiral process model used.

## WATERFALL MODEL

In order to approach the development from the waterfall model perspective, it is paramount to clearly defined our requirements before any development process can begin as the waterfall Model illustrates the software development process in a linear sequential flow. This means that any phase in the development process begins only if the previous phase is complete. In this waterfall model, the phases do not overlap.

Under this model we discuss the development process in the following format

* Requirements gathering and analysis
* System design
* Implementation
* Integration and testing
* Deployment of the system
* Maintenance

#### 4.1.2.1. Requirement gathering and analysis

The requirement gathering phase was divided into two; requirement collection and analysis.

In the requirement gathering phase, we employed various method in order to gather requirement that will make the system an integral part of the electioneering process and usable to a wide range of people. Below are the methods employed

* Interview
* Online searching
* Questionnaires

#### 4.1.2.2. System design

The design phase was also categorized into three phases;

* System architecture design
* Component design
* Interfaces design

The system architecture was design to meet user and system requirement, we bring various entities together to model the real-world voting process but making it very easy for the voter as well as ensuring anonymity, voidance of vote forgery, transparency and so on.

We planned the architecture based on the data collected from the intended users and stakeholders of the system requirements.

In the next stage, we defined various components of the system. This following are the various components and the designed process are discussed in the previous chapter of this work.

1. Voting class
2. RA class
3. Miner class
4. Board class
5. Network class

To ensure that the interfaces are agile, we designed a GUI for each interface. This gives user the power to decide on when and how to interact with who and on which interface.

#### Implementation

To ensure an efficient implementation of the system, we run and tested the various units of the system individually to make sure that everything work fine.

The various component tested are discussed in the Boehm’s model in previous chapter.

#### Integration and testing

All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.

In the testing process 100 students were made to vote against two parties. Deployed online by the system 2 people representing the board members.

We also, use another computer we tag registration authority and this node was responsible for deploying the contract for voting to start.

The results were amazing and the analysis is as below

Figure 4.2 show the waterfall approach to chain vote

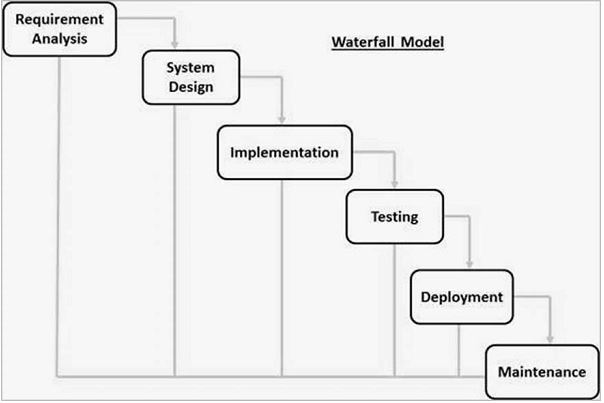


Figure 4. 3 Waterfall development process model

### INCREMENTAL MODEL

This process of the software design was most suitable for our system, since the requirement of the system usually changes and considering the size of the system. With the incremental model the user requirements were divided into multiple standalone modules of the software development cycle. In this model, each module goes through the requirements, design, implementation and testing phases. Every subsequent release of the module adds function to the previous release.

We treat each component as a standalone system ensures that its’ requirement, design, implementation and testing phases are achieved before moving to the next component.

In this model, the various component of the software; The RA, Board, Miner, Voter and Network are treated as if there were separate standalone software on their own.

Advantages of the Incremental model

* Errors are easy to be recognized.
* Easier to test and debug
* More flexible.
* Simple to manage risk because it handled during its iteration.
* The Client gets important functionality early.

Disadvantages

* Need for good planning
* Total Cost is high.
* Well defined module interfaces are needed

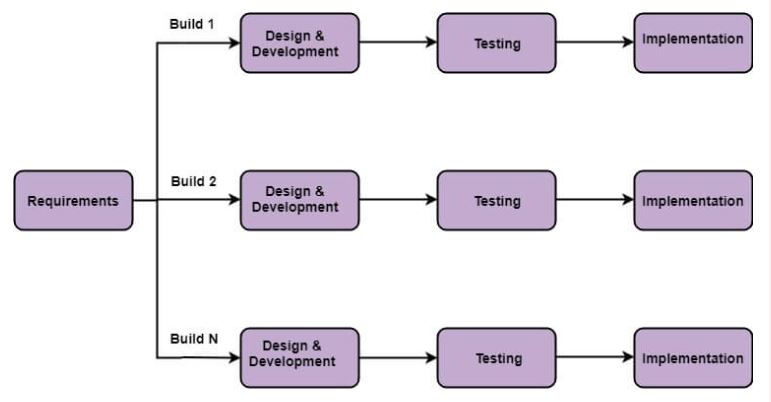


Figure 4. 4 Incremental process development model

### 4.2.1 USER INTERFACE DESIGN

The user interfaces were all developed to be agile and flexible, we made them simple to understand and a novice can master a walkthrough of the system upon a few practices and interactions with the system.

With the user requirement understood the user interface design was planned in the following stages.

The system has implemented the following user interfaces as per it functions written using HMTL, CSS and JavaScript.

* 1. Voter login interface
  2. Registration Authority page
  3. Miners’ interface
  4. Board interface

For the sake of simplicity, the user interfaces are made with minimal graphic contents but containing relevant GUI elements to direct the user to a successfully completion of the various task that can be performed in each interface.

### 4.2.2 DATABASE DESIGN

The system shall basically run on a shared database of the KNUST students. The database was designed using MySQL lite.

We carefully planned the databases to meet the data needs of the system. The logic behind this application is such that most of its runtime processes, such as preventing double voting are implemented with the support of the blockchain network, because of this unique feature the database of every component in the system is distributed to other components and everyone sees the state of the database at runtime and ensures that illegal modifications of the database is detected by the blockchain network.

Below is a list of the tables for each component and description:

Registration Authority

|  |  |
| --- | --- |
| Table | Description |
| Election | Keeps information about the election process |
| Parliament | Holds the parameter of the parliament |
| Party | Holds information about party details |
| President | Keeps information about the presidential candidates |
| Registrar | Keeps information about elections authority |
| Voters | Keeps information about the voters |
| Verification | Use to show that users are authenticated by the RA as eligible voters. |

Table 4. database table for the RA

Miner

|  |  |
| --- | --- |
| Table | Description |
| Chain | Maintains the state of the chain, in the blockchain network. |
| Contract | Hold information about the contract and status |
| Profiles | Used to hold username and password of miners |
| Connected nodes | Details the information of node that are connected to the network |
| Mined transactions | Transactions that are successfully mined and added to the blockchain |
| Non – mined transactions | Transaction that are in the pending list and not mined yet. |

Table 4. database table for the Miner

Board

|  |  |
| --- | --- |
| Tables | Description |
| Parliament | Holds information of the parliamentary candidates |
| President | Holds information of the presidential candidates |
| Board | This keeps the board profiles |

Table 4. database table for the Board

# CHAPTER FIVE

## IMPLEMENTATION OF THE SYSTEM

### Setting Up a Node(s)

1. To begin, you first need to import **queries.sql** file in the database folder at the root folder from **sqlite** browser and create the database needed to keep track of ledger and other resources needed by every node on the network.
2. While you are in the root folder, run the flask application by executing application with this command “py app.py” from the command line.
3. The default port for the flask application is `4000`, you can specify a different one when running the app using the new `--port NEW\_PORT` command.
4. Access the Graphic User Interface, and create a wallet by signing up to create a profile and get a private and public key to be used for your transactions on the network.
5. You can set up a new node by making a copy of the code and repeating the steps above.
6. Note that, the nodes on the network use `SOCKET` connection to connect to themselves. The port used by each node for this, is always `1000` more than the one used by the flask app. So, to connect to a different node, you would need the IP address and the port (the one used for the socket connection) of the node.

Figure5. 1 generating key for node

Figure5. 2 Login to as node

1. Once a network of two or more nodes is set up successfully, basic functionality like mining, transacting, verifications of transactions, etc. of any node on any peer-to-peer blockchain network could be achieved.

Figure5. 5 node dashboard

Figure5. 6 node transactions

### Integrating A DApp On the Blockchain (E-VOTE APPLICATION)

Besides being used as a `Crypto Currency` system, this blockchain protocol also serves a platform for decentralize applications. It uses the concept of `smart Contract` to achieve this. This blockchain uses the stateful protocol. State is like a result created by the blockchain where you can get accounts’ information like balance or used timestamps.

This blockchain protocol follows an account-based model, which is better for smart contracts and is also used by Ethereum, while Bitcoin and many other networks use the UTxO (unspent transactions’ output) model. This model works like a key-value database, with the key being an address, and the value is its information. Every time a new transaction is submitted, the state will be changed according to the information in the transactions.

This Blockchain protocol provides APIs where transactions from such decentralize applications could submit transactions to the network and get some information about those contracts.

Any Decentralize application used in this project can be found in the `Applications folder` in the root directory.

#### Electronic Voting System Used In this Project

Per the decentralize application being integrated with the blockchain, you can add other nodes to the network besides the actual `Miner Nodes' who are helping to establish the network.

In this our application of voting, we introduced two other different type of nodes, Registration Authority and Board Member nodes.

###### Registration Authority (RA)

This entity is like an Election Commissioner. This entity is responsible for starting the whole election process. He creates and initialize some information like the election id, election addresses, etc. at the start of the whole process.

Figure5. 7 RA view of the candidates sent by the board

Figure5. 8 RA dashboard overview

###### Board Member

There is always more than one board member, each indicating a representative from the various political parties involved in the election. This entity is responsible for submitting information about his/ her party information to the registration authority.

Of course, besides the stated nodes in this application, there are also voters who are engaged in this whole election process.

Figure5. Board dashboard

Figure5. Board dashboard

Figure 5.9: Board added candidate

Figure5. Board approved contract state as deployed by RA

Figure5. Board interface overview

###### STEP BY STEP ACTIVITIES IN THE VOTING PROCESS

Access the `Application folder` and initialize the databases of the RA and board using the queries.sql of

each of them by importing file using sqlite browser. After initializing the database for the RA and each of the board you want to be involved, you run the flask application for each of them and create the profile and get public and private key for each entity. You then run the application to be used by the voters.

STEPS

1. The RA initiate the election from the graphic interface, to get the address which you would be used as the public key for the election smart contract when deployed to the blockchain network.
2. Each board member representing various political parties acquire information for each candidate for the various portfolio and submit that information together with the public key of that board to the RA.
3. After all board members have submitted their candidate’s information to the RA. The RA creates the election smart contract with the information provided by each of the board. In this application, there are only two portfolios, 'President and Parliament (for each constituency). We assumed only two constituencies for demonstration purposes. In our case, the smart contract was prewritten using the python programming language. It can be found in the RA directory.
4. The RA configure the election information, like the start and end date of the election including the time.
5. The RA then deploys (sends it as a transaction to any miner node) it to the network.
6. No Ballot paper (transaction from a voter to the smart contract) would be accepted by the smart contract unless the Board Members approve (transaction from BM to SC indicating approval) the smart contract.
7. After all the board members have approved, the election has officially begun.
8. To vote, a user has to login to be verified since only eligible voters are required to take part of the election. The RA maintains a list of voters who are eligible to vote.
9. A voter logs in, gets verified and the RA generates a signed ballot paper (contains information about the candidates) for the voter. Each voter has a unique ballot signed by the RA which would be verify by the smart contract on submitted to prevent the forgery of ballot papers by the unlawful entities.
10. Voters are supposed to vote within the allocated time, since the system does not accept any transaction to a smart contract that has expired.
11. Once the election is over, the RA, the board members and the voters can view the election the election results

Figure5. Voter tries to vote but election period is over

Figure5. Voter select his choice of candidate to vote for.

Figure5. Voter try to double vote

Figure5. Voter login page

## SYSTEM IMPLEMENTATION AND TESTING

The system will have the properties of ballot transparency, privacy, individual verifiability, eligibility and etc.

The administration can be broken into two authorities, the registration authority and the board member.

### TESTING

A total of hundred people were made to vote against four constituencies consisting of eight total parliamentary members and two presidential candidates.

We tested the system by assigning parliamentary candidates for each of these constituencies, namely: Ejisu, Asokwa, Bekwai and Juaben.

This gives a total of eight parliamentary candidate and two presidential candidates making a total of ten candidates.

### TESTING RESULT I

The accuracy of the system was 98%, this was due to some individuals not being able to vote before the election period expired.

### TESTING RESULT II

We notice an accuracy of 100%, when we modified the time

and make sure everyone was able to cast their vote

Figure5. Results analysis from a mobile phone

Figure5. Results analysis from desktop computer

# CONCLUSION

The goal of this research is to develop an electronic voting system based on blockchain technology. We start by enumerating the advantages of the blockchain and how we can leverage the unique advantages in the context of electronic voting.

We presented the blockchain concept, followed by existing system that we built on this technology. Then a set of deficiencies in electronic voting systems are identified and addressed, current solutions for blockchain-based electronic voting systems.

After implementation and testing, we can conclude that all for objectives of the system were achieved with a 100% accuracy.

Despite numerous advantages we can leverage, there have been many study gaps in electronic voting that need to be addressed in future studies. Scalability attacks, lack of transparency, reliance on untrustworthy systems, and resistance to compulsion are all potential drawbacks that must be addressed.

# REFERENCES

Aicha Fatrah, S. E. (n.d.). *Transparent Blockchain-Based Voting System: Guide to Massive Deployments.*

Chang, V., Baudier, P., Zhang, H., Xu, Q., Zhang, J., & Arami, M. (2020). How Blockchain can impact financial services–The overview. *Technol. Forecast. Soc. Chang.*, 158, 120166.

Cullen, R., & Houghton, C. D. (2000). y online: An assessment of New Zealand government web sites. *Gov. Inf. Q*, 243–267.

Economist, T. ( 2020, January 18). *EIU Democracy Index*. Retrieved from https://infographics.economist.com/2018/DemocracyIndex/

Fatrah, A. E. (2019). Proof of concept blockchain based voting system. *In: Proceedings of the 4th International Conference on Big Data and Internet of Things* (pp. 1–5). BDIoT.

Gao, S., Zheng, D., Guo, R., Jing, C., & Hu, C. (2019). An Anti-Quantum E-Voting Protocol in Blockchain with Audit Function. *IEEE Access*, 7, 115304–115316.

Hakak, S., Khan, W., Gilkar, G., Imran, M., & Guizani, N. (2020). Securing smart cities through blockchain technology: Architecture. *IEEE Netw*, 34, 8–14.

Hang, L., & Kim, D.-H. (2019). Design and implementation of an integrated iot blockchain platform for sensing data integrity. *Sensors*, 19, 2228.

Imperial, M. (2021). An Enquiry into the Vision of Blockchain-Powered E-Voting Start-Ups. *The Democracy to come*, (pp. 4, 17).

Kim, T., Ochoa, J., Faika, T., Mantooth, A., Di, J., Li, Q., & Lee, Y. (2020). An overview of cyber-physical security of battery management. *IEEE J. Emerg. Sel. Top. Power Electron*.

Liu, Y. (2017). An E-voting Protocol Based on Blockchain. p. 1024.

Ometov, A., Bardinova, Y., Afanasyeva, A., Masek, P., Zhidanov, K., Vanurin, S., . . . Komarov, M. (2020). An Overview on Blockchain for Smartphones: State-of-the-Art, Consensus, Implementation, Challenges and Future. *EEE Access*, 8, 103994–104015.

Racsko, P., Yaga, D., Mell, P., Roby, N., & Scarfone, K. (2019). .Blockchain and Democracy.

S Sekar, C. V. (2020). Decentralized e-voting system using Blockchain. 312-324.

Schinckus, C. T. (2020). An overview of the sustainability of blockchain technology. *Energy Res. Soc. Sci.*, 69.

Shahzad, B., & Crowcroft, J. (2019, 7). Trustworthy Electronic Voting Using Adjusted Blockchain Technology. *IEEE Access*, pp. 24477–24488.

Wang, B., Sun, J., He, Y., Pang, D., & Lu, N. (2018). Large-scale election based on blockchain. 129, 234–237.

Xiao, S., Wang, X., Wang, W., & Wang, H. (2019). *Survey on Blockchain-Based Electronic Voting. In Proceedings of the International.* Oita, Japan,.