Arsalan Ishtiaq

**01-134182-013**

Sarah Farooq

**01-134182-050**

# D-Voting

**Bachelors of Science in Computer Science**

Supervisor: Ms. Maryam Bibi

Department of Computer Science
Bahria University, Islamabad

**June 2022**

# Certificate

We accept the work contained in the report titled **"D-Voting"**, written by **Arsalan Ishtiaq** and **Sarah Farooq** as a confirmation to the required standard for the partial fulfillment of the degree of Bachelor of Science in Computer Science.

## Approved By:

**Supervisor:** Maryam Bibi

________________________________________

**Internal Examiner:**

________________________________________

**External Examiner:**

________________________________________

**Project Cordinator:** Dr. Moazam Ali

________________________________________

# Abstract

Online voting systems were thought of as the solution to the problems of the traditional voting system but the problem still lingered because even in the online voting systems, there is a single authority that makes the decisions hence, leading to a single point of failure. As the technology is enhancing day by day, individuals are becoming more aware of the ways elections are rigged and the traditional voting system, as we all know, gets rigged or either the ballot boxes are burnt hence, putting a question mark on the credibility and authenticity of the whole voting system. These issues result in trust problems like corruption, fraud, fake votes, etc. A decentralized voting system represents all the steps involved such as showing the list of potential political parties eligible for voting along with their candidates, smart contract, a list of voters, and real-time democratic results. In decentralized voting, a smart contract is set up and not a single entity has the authority to make decisions. Blockchain is an emerging technology that is very safe and we have used it to overcome the problems of the traditional voting system by providing all the voters to see the real-time democratic process and since the votes are counted autonomously, so the chance of human error is also eliminated. We are using Ethereum Blockchain to make the most out of it. It brings the power of permission-based blockchain to enable reliable, safe, and quick applications. Our project will help solve the issues of traditional and online voting systems as it provides security, is non-hackable as one needs to win a consensus mechanism to make any change in the smart contract, maintains the anonymity of the voters and the counting of votes happens automatically.

# Acknowledgements

In the name of ALLAH the Most Gracious and the Most Merciful. We pay our primary gratitude to Almighty ALLAH for giving us courage, strength to conduct this Final Year Project and complete it. We would like to thank our supervisor Ma'am Maryam Bibi who showed trust in us, gave us chance to work on this project. My friends, who guided us, helped us to complete this work. Certainly it was not possible for us to successfully complete this project on time without their timely guidance and tips. We would like to thank Mr. Syed Hassan tanvir and Ms. Saima Jawad for supporting us in doing this project. We would also like to thank our families for bearing my academic engagements throughout our BS program. We thank our parents for praying and supporting, our siblings for helping and guiding us in essential points.

Arsalan Ishtiaq, Sarah Farooq
Bahria University Islamabad, Pakistan

July 14, 2022

*”When you change your thoughts, remember to also change your world.”*

Norman Vincent Peale

# Contents

# List of Figures

# List of Tables

# Acronyms and Abbreviations

| | |
|---|---|
| **API** | Application Programming Interface |
| **ASP** | Active Server Pages |
| **Baas** | Blockchain As A Service |
| **CNIC** | Computerized National Identity Card |
| **CSS** | Cascading Style Sheets |
| **EC** | Electronic Commission |
| **ER** | Entity Relationship |
| **ETH** | Ethereum |
| **EVM** | Electronic Voting Machine |
| **GUI** | Graphical User Interface |
| **HTML** | Hypertext Markup Language |
| **ID** | Identification |
| **ZoE** | Zcash and Ethereum |

# Chapter 1

# Introduction

## 1.1 Overview

Blockchain is an incorruptible, immutable digital ledger that can be programmed to practically record everything significant or valuable. It is a digital ledger of transactions that is divided and shared evenly across numerous networks of computer systems through the use of a peer-to-peer (P2P) network. Without the use of a central server, data can be shared directly among nodes in a P2P network. A consensus mechanism is used to update the data in the blockchain and this specific attribute of blockchain gives it the power of decentralization.

The cryptography technique is used to secure the transactions as well as the blocks. In the blockchain, each bock is connected to the next block using the hash of the previous block. The first block of the blockchain is called "Genesis". The only way to add data to the blockchain is by following the time-sequential order. Every transaction that is made is recorded on the block and when the capacity of the block is reached, the block is added to the ledger called the blockchain. A unique hash code is given to every block and every transaction made, so if someone attempts to make any change to the data of the transaction, the hash code of that transaction automatically changes which makes the chain invalid.

Blockchain has attracted many applications towards it because of its security, robustness, and great potential. Voting problems can be solved using blockchain. With the passage of time, people's faith in the traditional voting system has been decreased, and to restore their faith, there is a need for an innovative system. The proposed system will not only allow people to cast votes from within Pakistan but also from anywhere in the world because of its decentralized nature. The system would include some mechanism related to the security and the anonymity of the voter which helps us in ensuring that it is rigging-free. There is a need of making people believe that their votes matter. Right now, there is no easy way to check whether the vote cast by the person is authentic, cast, or counted rightly. The complex system of casting votes to choose the leader is not transparent to the voters.

To address the problem of the unsecured voting system, a blockchain-based solution is designed which empowers the voters while showing them the live statistics of the votes. In the designed system, a smart contract is used to manage the transactions and communication among all the network nodes. All the nodes of the system are able to verify the transactions that are stored in the decentralized system database. The introduction of the private blockchain allows all the network nodes to encrypt their confidential information however, there are some significant participants who have an additional authority such as the owner of the smart contract. This system will collect and manage the votes, not only the votes of people living within the country but also the votes of overseas Pakistanis. The main purpose of the system is to create a safe, robust, and trustworthy system that will not only restore the faith of people in voting but also allows the overseas Pakistanis to cast vote while sitting anywhere in the World. It also helps to encourage the development of advanced business strategies based on blockchain and IoT.

## 1.2 Problem Description

With the improvements in technology every day and emerging technologies of the Internet, improvements are being made to the voting systems, but almost all the advanced systems are centralised which do not solve the major issue of voters trust, false votes, and rigging. Centralised systems have a single point of failure meaning that it is controlled by a single authority which makes them more vulnerable to hacking and attacks and that's the biggest drawback as it can lead to the whole system crash. In traditional voting system, there is no way to authenticate if the person has casted vote only once. Similarly, the votes are counted manually which means there is a chance of human error and the votes can easily be tampered. Since the traditional system is centralised, the central authority could be biased which could lead to the selection of an undeserving candidate. Blockchain technology provides us an efficient solution for secure voting. It helps us in maintaining the anonymity of the voter as only a hash is generated when the vote is casted by the person. If decentralized voting is implemented successfully, then all the problems of the traditional voting can be solved.

## 1.3 Project Objectives

Our project's goal is to design and develop a system that uses blockchain technology, features, and characteristics to conduct elections that are fair while maintaining the anonymity of the voters and showing real-time democratic results. A web application will be developed which would provide a user-friendly interface to the end-users. In order to overcome the problems of the traditional voting system, blockchain technology is quite helpful. The proposed system will allow the development of a decentralized electronic voting system on blockchain.

## 1.4 Project Scope

The project scope is a declaration of what work is and isn't included in the project. A proper scope declaration minimizes the risk of project overruns and unexpected turbulence greatly. Our project aimed at building a web application using Blockchain Technology which will help provide a set of protocols that allow voters to cast votes while the election commission is responsible for creating elections and adding candidates and voters.

# Chapter 2

# Literature Review

In this section, we have discussed the contributions and research papers of different researchers who have provided and discussed their findings and theoretical solutions using blockchain to introduce decentralized voting. With the increasing threat of security and rigging; it has become almost impossible to ensure fair elections. Using blockchain technology, secure and fair elections can be ensured. We have categorized these research works into centralized and decentralized system approaches.

## 2.1 A Political Election

An election is a procedure through which the general public selects candidates for public office. Typically, the selected candidate will represent the public's needs through the voting process. A central voting authority is in charge of overseeing, organizing, validating, and announcing the election results in a centralized manner. The voting system can be organized in a variety of ways, depending on the voting authority chosen by each organization or country. There are three types of voting systems in general which are mentioned below:

- Ballot-based voting systems.
- Electronic-based voting systems.
- Online voting systems

The voting systems are centralised and can be summarised as follows:

### 2.1.1 A Ballot-based Voting System

A ballot system is the most prevalent centralized voting mechanism. Every authenticated person who is approved by the voting authority is considered a voter in a ballot system. A ballot, which is usually a little piece of paper, would be provided to each voter. After that, the voter would choose one of the options before casting their ballot in a polling station box.
After the voting session has ended, the voting authorities will normally transport the boxes containing the voters' votes to a public location designated for the vote

counting. The ballot boxes would be opened, each ballot is examined for accuracy, and the votes would be classified publicly. Stakeholders in the voting process are frequently invited to see the vote tallying process to help validate it.
However, this form of the voting process has many disadvantages and some of them are listed below

- It is costly to prepare a paper-based voting ballot.
- Printing out all of the ballots, on the other hand, becomes an expensive and time-consuming task.
- If the voting authority is biased and there are no stakeholders checking the tallying process, the procedure becomes time-consuming, error-prone, and simple to tamper.

### 2.1.2 An Electronic-based Voting System

One option for improving paper-based ballot systems is to use electronic voting systems. Electronic voting machines, often known as electronic ballot machines or EVMs, are used in electronic voting systems. Even private networks or Internet services, such as the transmission of tabulated results after voting results, are possible with EVM systems. In nature, the EVM would have specific criteria, such as high levels of security, privacy, accuracy, verifiability, accessibility, and scalability.
Even after providing many benefits, EVMs still face a few challenges as mentioned below:

- The high cost of EVM machines is one of the main reasons for their sluggish acceptance in voting systems.
- Many EVMs have been poorly implemented, and there have been reports of voting fraud on the platform. This should not be interpreted as a criticism of EVMs in general, as voter fraud is typically caused by poorly implemented security systems behind the EVM machine.

### 2.1.3 Online Voting Systems

The ability to vote online, via online network systems such as the Internet or private networks set up by the voting authority, is one type of electronic voting system. The ballots are completed online due to the online nature of the voting systems, which eliminates the costly requirements of ballot preparation inherent in traditional voting systems. The votes would be routed to a central server for tallying, making the procedure easy, and efficient.

Although the online voting system provides many benefits; a growing number of technical concerns must be addressed by the voting authority such as the system should be able to authenticate and allowing multiple users to vote at the same time without experiencing severe delays or issues. Similarly, the security components of online voting systems are exceedingly complicated, with numerous points of attack and vulnerability. An attacker only needs to exploit one of these points to alter vote results or even invalidate parts or all of the voting.

## 2.2 Centralized Systems

The effectiveness of the democratic process i.e., voting depends a lot on the methodology used. If different stages of voting can be tracked, only then we will be able to monitor the real-time democratic process. There is an existing voting system called Israeli E-Voting Scheme. They came up with an idea to save paper and to make the process of voting transparent. The components of the voting station are shown in Figure 1.

Each voting station has a few elements. At first, a voting terminal, which is used by the voter to cast the vote. The vote is stored on the contact-based smart card connected with the terminal. Secondly, there is a verification terminal that a voter can use anytime to verify his/her vote. Thirdly, there are blank ballots that are cryptographically paired with verification terminals. Fourthly, there is a voting booth, where voters go to cast vote privately. At last, there is a ballot box, where contactless smartcards are collected and counted by the administration. The local elections committee consisting of several trustworthy people is always present in the stations to oversee the voting process and they also have the responsibility to cross-check the voter's identity. However, the proposed voting system didn't last for long as its credibility was compromised by a relay attack [1].

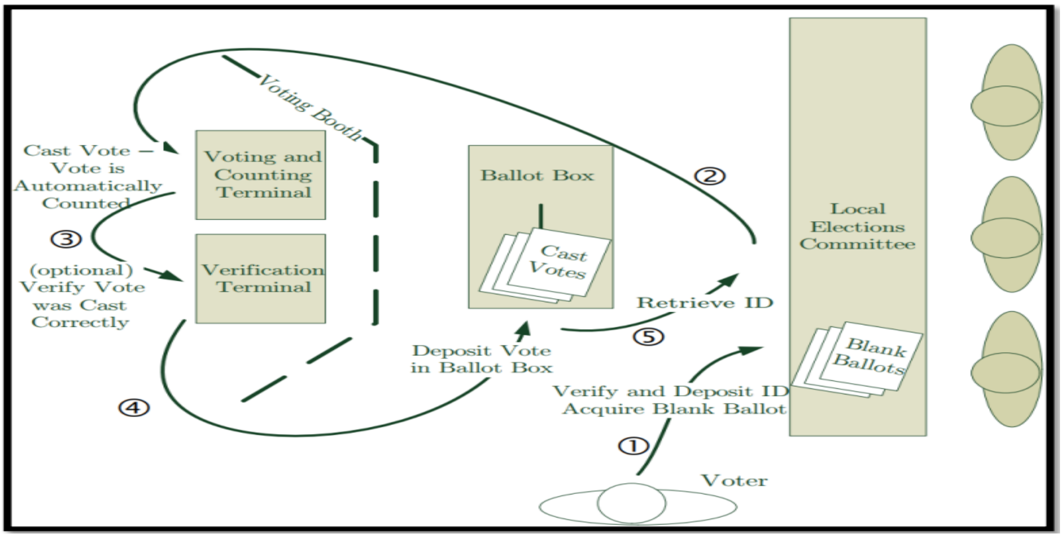

Voting Booth
Cast Vote – Vote is Automatically Counted
Voting and Counting Terminal
③
(optional) Verify Vote was Cast Correctly
Verification Terminal
④
Ballot Box
Cast Votes
Deposit Vote in Ballot Box
②
Retrieve ID
⑤
Verify and Deposit ID Acquire Blank Ballot
①
Voter
Local Elections Committee
Blank Ballots

Figure 2.1: Israeli E-Voting Overview

There is another system devised for centralized voting. The devised system consists of a computerized national identity card (CNIC), QR code scanner, biometric recognition system, voting machine, and the central database as shown in Figure 2. A QR code scanner is used to scan the QR code printed on the CNIC or identification card of the voter. In this way, the authenticity of the voter will be verified, then a biometric recognition system, which in this case is eye retina scanning, is used to identify the person. Once the person is verified, the electronic voting machine (EVM) will be activated. The voter can then vote for the candidate of his/her choice with the help of this machine. All of the data is kept on SQL Server, and the required constraints can be applied to assure the voting's integrity [2].

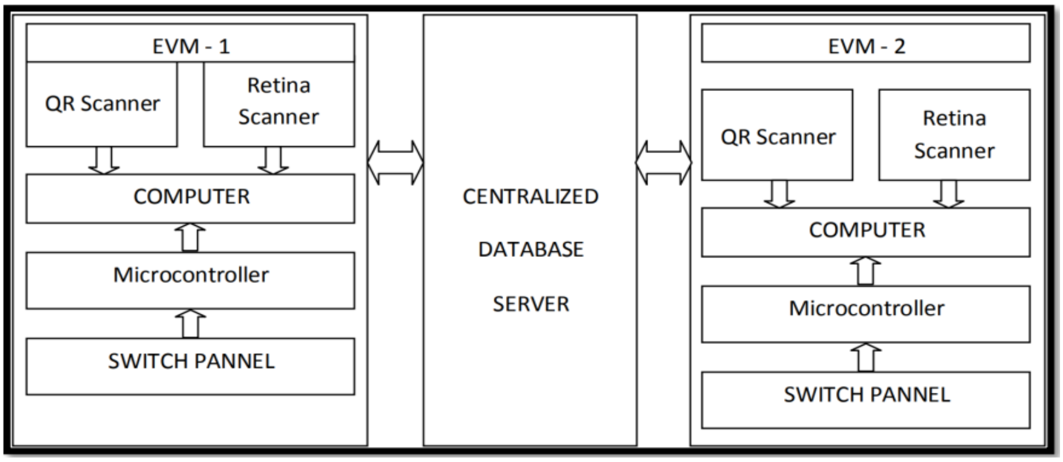


Figure 2.2: E-Voting using centralized database

## 2.3 Blockchain: Background

A blockchain is made up of a sequence of cryptographically linked blocks. Each block includes information and data. The Genesis block is the very first block in the chain. Except for the Genesis block, which is the first block in the chain, each block in the chain has data, hash, and the hash of the previous block. Cryptography is a technology that restricts access to the content of messages sent and received to just the sender and receiver. The secret key is used to secure/encrypt the data. When the sender transmits a message to the receiver, he or she includes the secret key in the message. The communication is then decrypted by the receiver with the same secret key[3].

Originally, blockchain was intended to be used to assist Bitcoin transaction processing. Bitcoin's fundamental goal is to provide a decentralized financial system in which each transaction between ledgers can be completed without relying on a single server to validate transactions. Rather than depending on a central bank, other ledgers in the Bitcoin ecosystem will attempt to verify the transaction's legitimacy on their own. The fact that each transaction within the Bitcoin ecosystem is depicted as a Block is why the algorithm is called a blockchain.
Blockchain is a distributed database that utilizes a peer-to-peer network. The consensus process adds the material to the distributed database. When a new block is formed, it is distributed to all network users. Each node verifies the block and

then adds it to its own blockchain once it is completely verified. This is how nodes come to an agreement on which blocks to add and which blocks to leave out. In a peer-peer network, all peers communicate directly without the need for a central authority. Blockchain is also sometimes referred to as distributed ledger which means data once stored in a block cannot be changed or deleted. It is an append-only ledger. “A ledger is a written or computerized record of all the transactions.”[4].

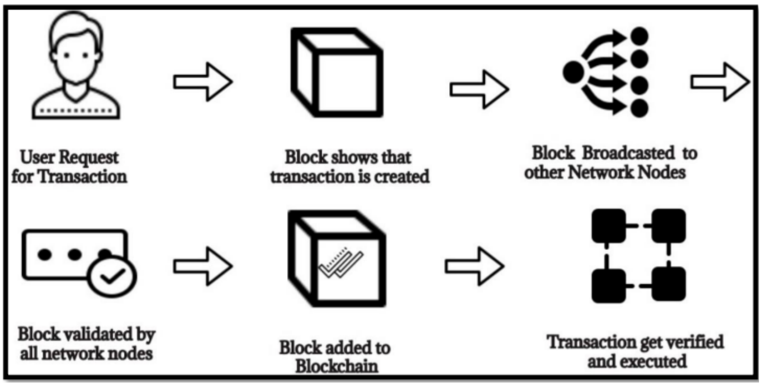

User Request for Transaction
Block shows that transaction is created
Block Broadcasted to other Network Nodes
Block validated by all network nodes
Block added to Blockchain
Transaction get verified and executed

Figure 2.3: Explanation of Blockchain Modules

## 2.4 Web 3.0

The introduction of blockchain technology brings in a new age of the web, which we refer to as Web 3.0. The early days of the Internet were dominated by information delivered via static web pages with no opportunity for interaction. It was primarily created by information portals with flat material that allowed visitors to ”just” read but not submit any comments, reviews, or feedback. The capabilities to communicate, share information, add content, and trade data define Web 2.0, or the second stage of the World Wide Web’s evolution. Then comes the third generation of the Internet called Web 3.0 which is built on decentralization, with no one point of control or profit centre. The blockchain allows value to be transferred without the need for a profit centre or monopolistic service providers. While social media allowed users to share information, it concentrated control in the hands of a few private actors (creating digital oligarchies with social media companies, peer-to-peer ride sharing, and peer-to-peer hospitality networks), and blockchain technologies allow for the creation of decentralized networks with no centralized points of control.

## 2.5 Smart Contracts

There should be a defined set of rules and logic upon which transactions should take place. Those sets of rules and logic are called smart contracts. Smart contracts are once written cannot be altered. If one wishes to update the smart contract, he/she must create an updated version of the previous contract and execute it. Smart contracts are stored in blockchains and are triggered automatically. The use of smart contracts significantly reduces the cost as it does not include the involvement of third parties. Smart contracts are deterministic in nature which means that the output is known for each input. This property ensures that each node gets the same input for consensus[5].

## 2.6 Decentralized Systems

We have seen the changing trend in which people feel reluctant to obey centralized authority due to factors like biasness, distrust, dishonesty, etc., and they feel an appeal towards the concept of decentralization. The traditional way of voting requires a lot of trust in the central authority and as discussed in the above section, the centralized system also revolves around the centralized authority. The centralized approaches may lead to the selection of undeserving candidates due to the centralized authority's dishonesty, and hacking of the centralized devices. The decentralized network has the potential to be used as a modern electronic voting technique to overcome the central authority. A decentralized approach helps us in seeing the democratic process in real-time, where blockchain is used for maintaining transparency in the data flow between voters and the capacity of data management. This approach will help us to choose the most deserving candidate [6].

BlockVote is a decentralized, blockchain-based system for electronic voting[7]. The system follows the three main steps of any voting process i.e., poll creation, voting, and result tallying. The anonymity of the voter is maintained throughout the voting process and results are announced once the voting period is over. The system used two protocols for implementation is using the Ethereum framework and using Hyperledger framework. In conclusion, the author has compared the two protocols and did an analysis of both technical and management aspects of adopting either protocol. The author has also suggested solutions to improve the system.

## 2.7 Related Searches

We have done research on conducting elections online with the use of blockchain technology. We came to know about the different methods used to conduct blockchain-based voting. We have summed up our findings in the following table which contains the title, tools and technology, consensus algorithm, limitations, and the solutions of the particular research paper.

| Title | Tools and Technology | Concensus Algorithm | Limitations | Solution |
|---|---|---|---|---|
| Conceptualization of a Blockchain-Based Voting Ecosystem in Estonia[8]. | Any Electronic device with access to the Internet. | - | I-Voting is just a supplement to the ballot paper. Voting turnout remained low at around 60 percent. | Estonia government provided its citizen with options to either choose to vote online via the internet or ballot paper. |
| On Trade-offs of Applying Block Chains for Electronic Voting Bulletin Boards[9]. | Public blockchains i.e., Ethereum and Bitcoin. | Proof of Work | The need for exception handling. The proposed ledgers provide no guarantees of transaction acceptance and performance limitations. | Proposed proposals make use of public permissionless economically incentivized blockchains, mostly either Bitcoin or Ethereum as these are the favorite choices in most of the blockchain-based voting proposals. |
| E-Voting with Blockchain: An E-Voting Protocol with Decentralization and Voter Privacy[10]. | Ethereum Blockchain | Proof of Work | There is still room for advancement in fundamental blockchain technology research in terms of functionality and support for sophisticated applications. | The blockchain is presented as a transparent vote box in a new prospective electronic voting technology. |

Table 2.1: Blockchain based voting papers (1)

| Title | Tools and Technology | Concensus Algorithm | Limitations | Solution |
|---|---|---|---|---|
| An E-voting Protocol Based on Blockchain [11]. | Integration of blockchain paradigm into proposed e-voting protocol. | - | It does not guarantee data privacy or neutrality. | They presented an e-voting technology based on blockchain and blind signatures. |
| The future of E-voting [12]. | Integration of Zcash and Ethereum (ZoE). | Zero-knowledge proof[13]. | The security protocol verification is inadequate. | The voting system's underlying technology is a payment method that provides transaction anonymity, a feature that has yet to be seen in blockchain protocols. |
| What if blockchain technology revolutionized voting [14]? | - | - | To deploy blockchain in national elections, various areas of European law would have to be followed. This may go against the blockchain's properties. | They propose that BEV would transfer power and trust away from central players like electoral officials, and instead promote the formation of a tech-enabled community consensus. |

Table 2.2: Blockchain based voting papers (2)

| Title | Tools and Technology | Concensus Algorithm | Limitations | Solution |
|---|---|---|---|---|
| Follow My Vote[15]. | Bitcoin Blockchain | Proof of Work. | Scalability. | It's a group that proposed a blockchain-based secure online voting network with the ability to audit ballot boxes and track election results in real time. |
| A Security Analysis of Voatz, the First Internet Voting Application Used in U.S. Federal Elections [16]. | Hyperledger. | - | A passive network adversary can learn about a user's vote, and an active one can respond by disrupting transmission. | To provide end-to-end encrypted and voter-verifiable ballots, Voatz uses a combination of a permissioned blockchain, biometrics, and hardware-backed keystores. |
| Polys online voting system [17]. | Exonum blockchain. | Practical Byzantine Fault Tolerance (PBFT). | Does not support write- in voting. | Polys goal was to help local governments, state governments, and other organisations save time and money by allowing them to focus on gathering and preparing proposals. |

Table 2.3: Blockchain based voting papers (3)

## 2.8 Conclusion

The research paper titled BlockVote[7] discussed earlier is the base of our project. The findings of the research paper showed that the suggested system may be utilized to conduct a poll while keeping the results secure and limiting the result tallying time to a minimum in both of their implementations. In our proposed system, we introduce an election mode in addition to the voting mode used in BlockVote. In the election mode phase, Election Commission will have the authority to start elections. Therefore, the system is intended to respond in a fair amount of time. The voter should be able to import his or her Election Commission-provided wallet in a matter of seconds while keeping network stability in mind. The system's performance varies depending on two modes that is: voting mode and election mode.

# Chapter 3

# Software Requirement Specification

In this chapter, we have discussed the requirements of our proposed project. Along with the requirements, we will also talk about the existing systems that are centralized and tell how our proposed system is different from them. This chapter will brief about existing systems, proposed systems, functional requirements as well as non-functional requirements.

## 3.1 Existing System

Electronic Democracy is a combination of electronic and democracy, which consists of the use of the Internet to enhance the democratic process within a democratic state. E-democracy aims at making the process of voting more accessible to the public so that more individuals can take part in policy decision-making. This approach will result in the development of smart policies, increasing transparency and accountability. It is a relatively new concept that has surfaced just recently and has been making rounds among the public mainly through social media. In this century, we have seen a drastic shift to online activities with everything gradually shifting online. Similarly, public participation in policy-making through cyberspace is defining a new form of democracy. Electronic voting combines the democratic process with technology to make the process more convenient for the voters. Electronic voting can be carried out in two of the following ways i.e., either people can vote from their homes using the internet or they can vote at a polling station.

### 3.1.1 E-voting at the polling station

This approach closely resembles the traditional voting process except that computer is used instead of the ballot paper. These computers have their private network and server. The process is centralized as the government monitors all the coordinated activities.

## 3.2 E-voting online

This advanced approach will allow voters to vote from their homes. All they will be needing is access to the internet. Being said that the Internet is viewed as the most suitable option to carry out the democratic process. The issue of electronic voting is global, and many states have tried to come up with solutions to tackle this problem. A few states and the approaches that they used are discussed below:

### 3.2.1 Brazil

This country tops the list when it comes to electronic voting because they were the first one to conduct elections online in 1990. Similarly, the elections of 1998 mark one of the largest electronic elections in the history of mankind. In this way, Brazil successfully managed to elect its President, 27 Senators, 27 Governors, and many other officials [18].

### 3.2.2 New Zealand

New Zealand tried to solve a problem concerning the democratic process which is overseas residents. We know that a significant number of people work abroad, and they miss the democratic process. Therefore, New Zealand devised a system where overseas could vote. In this way, registered overseas voted online at the parliamentary election in the Netherlands in November 2006 [19].

### 3.2.3 Australia

Australia came up with a unique and one-of-a-kind electronic voting system in 2001 and then in 2004. The system used barcodes to authenticate the voters and personal computers were used as the voting terminals. Voting terminals were linked to a private server in each polling station using a secured local area network so that there is no room for vote manipulation. They avoided the use of a public network like the Internet, and it resulted in a successful voting [20].

### 3.2.4 Pakistan

To date, no electronic voting system has been implemented in Pakistan to carry out the process of voting. There is no doubt that Pakistan is in dire need of a proper voting system that could be more tech-oriented rather than having human interactions all the time. With the increasing population and corruption, the traditional voting system has resulted in disputes and this approach doesn't look promising for the future either.

## 3.3 Proposed System

The proposed system is a blockchain-based system where voters can vote and see the democratic process in real-time. In this system, a decentralized approach is used to store data and information related to all network attributes. Once the data is stored on the blockchain, it cannot be changed. Blockchain is an immutable database meaning no one can manipulate the data. Blockchain also provides transparency. Hence, the blockchain-based voting system which cannot be tempered is much faster and more secure.

### 3.3.1 Product Perspective

The system consists of two parts. The EC will utilize one for general purposes, such as viewing candidates and their registered parties, viewing voter lists, and starting and ending elections. The other part uses web browsers such as Mozilla Firefox, Internet Explorer, Google Chrome, etc. On election day, voters should import their Ethereum wallets and be validated as a result. The voters will use the given interface to cast their votes. After the voters have cast their votes, the results will be updated according to the smart contract.

### 3.3.2 Product Features

Product features are the characteristics or properties of a product that provide value to end consumers and set it apart from competing products on the market. Some of the features of our project are mentioned below.

**Eligibility:**

Only eligible users are allowed to vote, according to this property meaning those who have received certification from the Election Commission.

**Privacy:**

One of the most fundamental features of democratic process is privacy. The anonymity of the voter is maintained throughout the process.

**Verifiable:**

According to this property, everyone involved in the voting process should be able to verify the results. This increases election transparency. A voter should also be able to check whether or not his or her vote was counted.

**Immutability:**

The voter's decision should be final. Once the vote has been cast, no one can change the vote.

### 3.3.3 Constraints, Assumptions and Dependencies

EC authenticates the system and provides the ETH wallet address and private key, allowing voters to vote from anywhere. The most important aspects of our blockchain voting method are security and anonymity. We can list the following assumptions and dependencies for the system's proper operation:

#### Metamask Browser Extension

Users can manage their accounts and keys in a variety of methods, including hardware wallets, while keeping them separate from the site context.

#### Ganache

It's a personal blockchain for developing Ethereum and Corda distributed applications quickly.

#### Truffle

A world-class development environment, testing framework, and asset pipeline for blockchains based on the Ethereum Virtual Machine (EVM), with the goal of making life easier for developers.

#### NodeJs

It's a JavaScript runtime based on the V8 JavaScript engine in Chrome.

### 3.3.4 Software Requirements

The table 3.1 enlists the software required to develop the project.

| Software | Type | Version |
|---|---|---|
| Ganache | Ethereum Blockchain Server | 2.4.0 |
| Metamask | Ethereum Wallet | 7.7.9 |
| Truffle | Development framework for ETH | 5.1.31 |
| Node | JavaScript Runtime | 12.17.0 |
| Visual Studio Code | Integrated development environment | 1.46 |
| Remix | Solidity's IDE | 0.10.1 |
| Remix | Solidity's IDE | 0.10.1 |
| Windows 10 | Operating System | 1809 |

Table 3.1: View ledgers and polling results

*For our blockchain-based voting system, functional and non-functional requirements are specified below:*

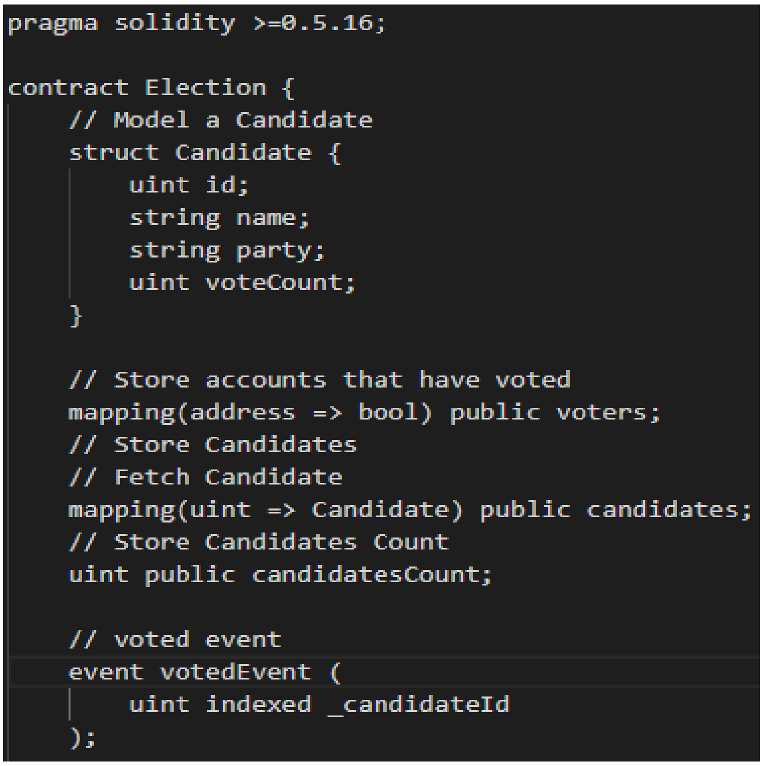
### 3.3.5 Functional Requirements

These requirements include the set of activities that are necessary for the system to perform. Below is the list of requirements that the developer must fulfill. According to our decentralized system, functional requirements are as follows:

**Smart Contract Developer**

- Smart contract developer should have the list of candidates and the list of voters in advance.
- The developer should include the necessary entities in the smart contract and deploy it on blockchain.

The pseudo code of the smart contract is shown in Figure 3.1.



```
pragma solidity >=0.5.16;

contract Election {
    // Model a Candidate
    struct Candidate {
        uint id;
        string name;
        string party;
        uint voteCount;
    }

    // Store accounts that have voted
    mapping(address => bool) public voters;
    // Store Candidates
    // Fetch Candidate
    mapping(uint => Candidate) public candidates;
    // Store Candidates Count
    uint public candidatesCount;

    // voted event
    event votedEvent (
        uint indexed _candidateId
    );
```

Figure 3.1: Pseudo code of smart contract

**End Users**

- Users should be able to log into their accounts with the help of unique address provided to them.
- Users should be able to vote.
- Users should be able to see real time democratic process.
- Users should be able to view their voting status.

### 3.3.6 Non-Functional Requirements

These requirements are not concerned with the functionality of the system. They impose restrictions on the product. One non-functional requirement can give birth to the number of functional requirements. According to our decentralized system, functional requirements are as follows:

- **Performance**

  The system should be able to reply to the user quickly. Similarly, exception handling is performed so that system does not crash and gives an alternative route to the user.

- **Anonymity**

  Only the voter's public key, which has been previously hashed, is broadcast on the blockchain. Due to the fingerprint's hash, which is essentially the binary values of the coordinates, no one, other than the voter, will be able to identify any voters within the blockchain.

- **Privacy**

  The likelihood of voter manipulation and coercion by dishonest supporters is decreased because voters do not know when to cast their ballots. Only the randomly generated time against each group in the code that was performed is used to clock the voters' votes. As a result, voters cannot be threatened or blackmailed by manipulators or members of a particular party.

- **Transparency**

  A distributed, open ledger known as a blockchain makes every transaction and activity accessible for peer validation, review, and visibility. This prevents fraudulent acts from happening in secret while keeping everything in plain sight. The transparency of blockchain allows for the fairness and accuracy to be achieved.

- **Serviceability**

  Once the voting has started, the voting event should trigger so that the voters are able to cast their votes.

- **Consistency**

  The application should be able to carry out multiple transactions at once.

## 3.4 Use Cases

Uses cases shows visually, how user will interact with the system. Following are the uses cases that will depict how the voter will interact with decentralized web application.

### 3.4.1 Voters and candidates' verification

The figure 3.2 and table 3.2 shows the verification phase of the voters and the candidates who are taking part in the elections.

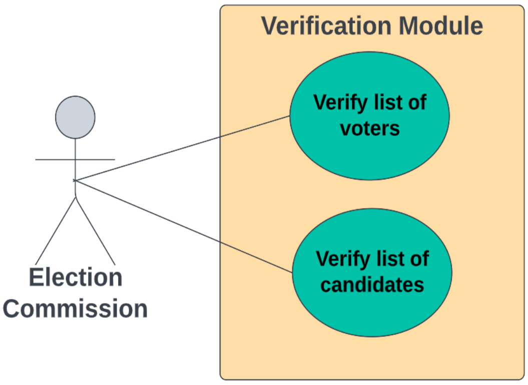


Figure 3.2: Outline of verification module

| | |
|---|---|
| **Use Case ID** | UC 1 |
| **Use Case Name** | Voters and candidates' verification |
| **Actor** | Government Official/List of voters and list of candidate's providers. |
| **Pre-Condition** | Candidates would have to register their interest in order to be added to the candidate registry list, and voting authorities would have to check the candidate's qualifications before adding them to the list. Similarly, the voting authorities would compile a list of potential voters and be in charge of verifying those voters before they cast their ballots. |
| **Post-Condition** | The list of candidates and voters will be passed to the use case UC 2. |
| **Success Scenario** | Successful creation of lists. |
| **Details** | Because of the differences in voting authority and the inability to include the process inside the architecture, the system would not handle the process of declaring interest and checking the qualifications of candidates before approval. |

Table 3.2: Verification

### 3.4.2 Smart Contract Deployment

The figure 3.3 and table 3.3 explains the interaction of smart contract developer with the blockchain.

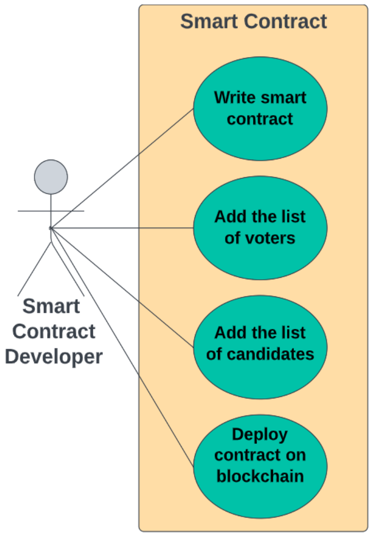


Figure 3.3: Outline of smart contract module

| | |
|---|---|
| **Use Case ID** | UC 2 |
| **Use Case Name** | Smart Contract Deployment. |
| **Actor** | Smart Contract Developer. |
| **Pre-Condition** | The smart contract should have a list of voters and candidates in advance. |
| **Post-Condition** | The smart contract is deployed on the Ethereum blockchain. Its copy is sent to all the nodes. |
| **Success Scenario** | Successful deployment of smart contract on Ethereum blockchain. |

Table 3.3: Smart contract deployment

### 3.4.3 Login

The figure 3.4 and table 3.4 explains that the user is verified against the list of voters that is provided in the smart contract.

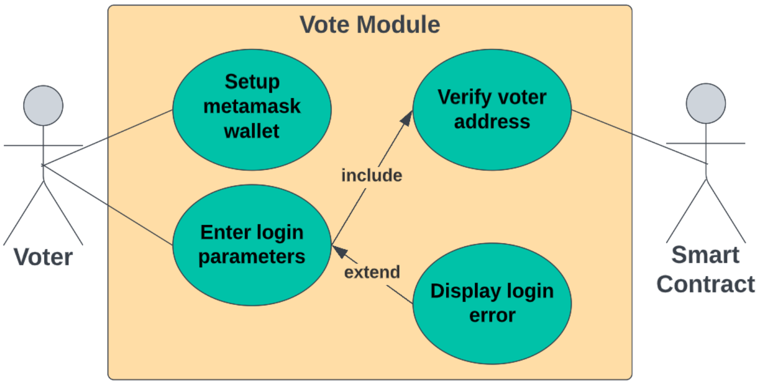


Figure 3.4: Outline of login module

| | |
|---|---|
| **Use Case ID** | UC 3 |
| **Use Case Name** | Login. |
| **Actor** | Voters. |
| **Pre-Condition** | Every voter must have a account on metamask wallet. |
| **Post-Condition** | Getting ready to vote and accessing the smart contract functionality. |
| **Success Scenario** | Successful verification. |

Table 3.4: Voter login

### 3.4.4 Vote Casting

The figure 3.5 and table 3.5 explains the vote casting procedure such that the vote is casted and submitted after paying a small fee through meta mask.

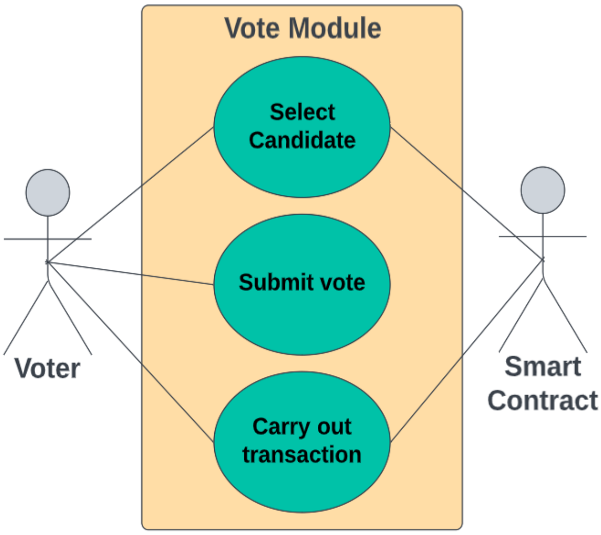


Figure 3.5: Outline of cast vote module

| | |
|---|---|
| **Use Case ID** | UC 4 |
| **Use Case Name** | Vote Casting. |
| **Actor** | Voters. |
| **Pre-Condition** | A metamask wallet and successful verification. |
| **Post-Condition** | The vote is added to the ledger. |
| **Success Scenario** | Transaction is carried out successfully and the vote is added to the ledger successfully. |

Table 3.5: Vote casting

### 3.4.5 View ledger and polling results

The figure 3.6 and table 3.6 shows that the voters are able to see the real-time democratic process.

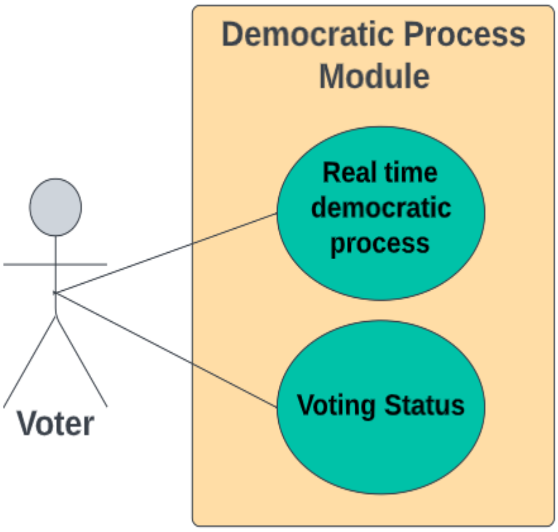


Figure 3.6: Outline of view ledgers module

| | |
|---|---|
| **Use Case ID** | UC 5 |
| **Use Case Name** | View ledger and polling results. |
| **Actor** | Voters. |
| **Pre-Condition** | The vote has been cast by the voter. |
| **Post-Condition** | View transactions by different wallets. |
| **Success Scenario** | Successful declaration of a winner after the poll has ended. All the users should be able to view the results. |

Table 3.6: View ledgers and polling results

### 3.4.6 D-voting overview

The figure 3.7 and table 3.7 shows all the sub modules of democratic process from start to end.

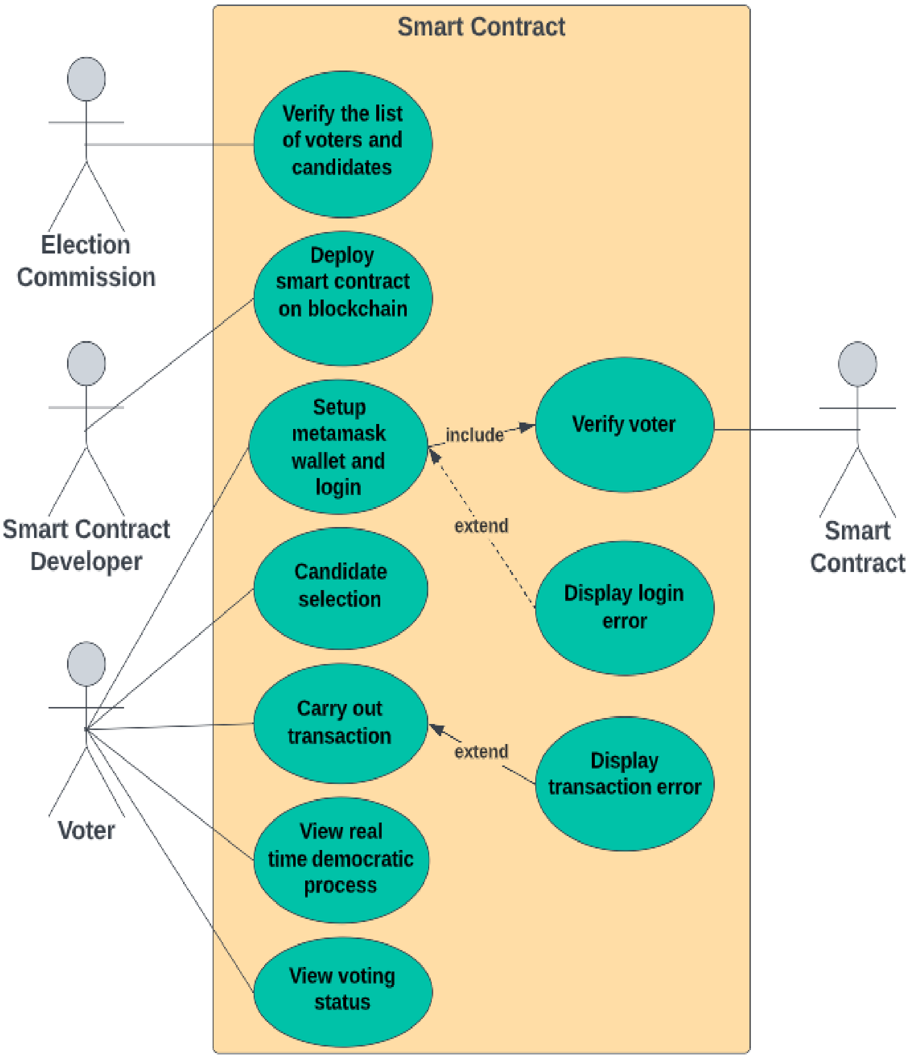


Figure 3.7: Outline of application's modules

| | |
|---|---|
| **Use Case ID** | UC 6 |
| **Use Case Name** | Overview |
| **Actor** | Voters, Government Official, Smart Contract Developer, Smart Contract |
| **Pre-Condition** | Eligible voters with access to the internet and metamask. |
| **Post-Condition** | List of candidates with their votes will be displayed to voters at the end of voting. |
| **Success Scenario** | All the modules of democratic process are carried out successfully. |

Table 3.7: Application description

# Chapter 4

# System Design

In this chapter, we have presented the development phase of our blockchain-based decentralized voting. The aim of this chapter is to understand the basic flow of our application so that we can turn our proposed solution into reality. This chapter contains the design goals, suggested application's abstract view, and conceptual model. We have also included specifications of our system components, modules, and data to fulfil the requirements stated in the previous section. The details of the chapter are discussed below:

## 4.1 Design Goals

Design goals are key properties of the system to be optimized that may have an impact on the overall design. The distinction between system design and requirements is thin. Design goals are properties that the designers seek to make "as excellent as possible" without precise criteria for acceptability, whereas requirements include particular values that must be reached in order for the product to be acceptable to the client.

## 4.2 High Level Design

The boundaries of a software system are described and shown using a context diagram, also known as a level 0 data-flow diagram. It tracks the flow of data between the system and external entities. The figure 4.1 depicts the entire software system as a single process. To conduct fair elections, we have proposed a decentralised system to ensure rigging free elections. A smart contract will be deployed on blockchain. The eligible voters are issued with an address through which they'll be able to conduct votes. When a person casts a vote, a new block is added to the blockchain which ensures transparency.

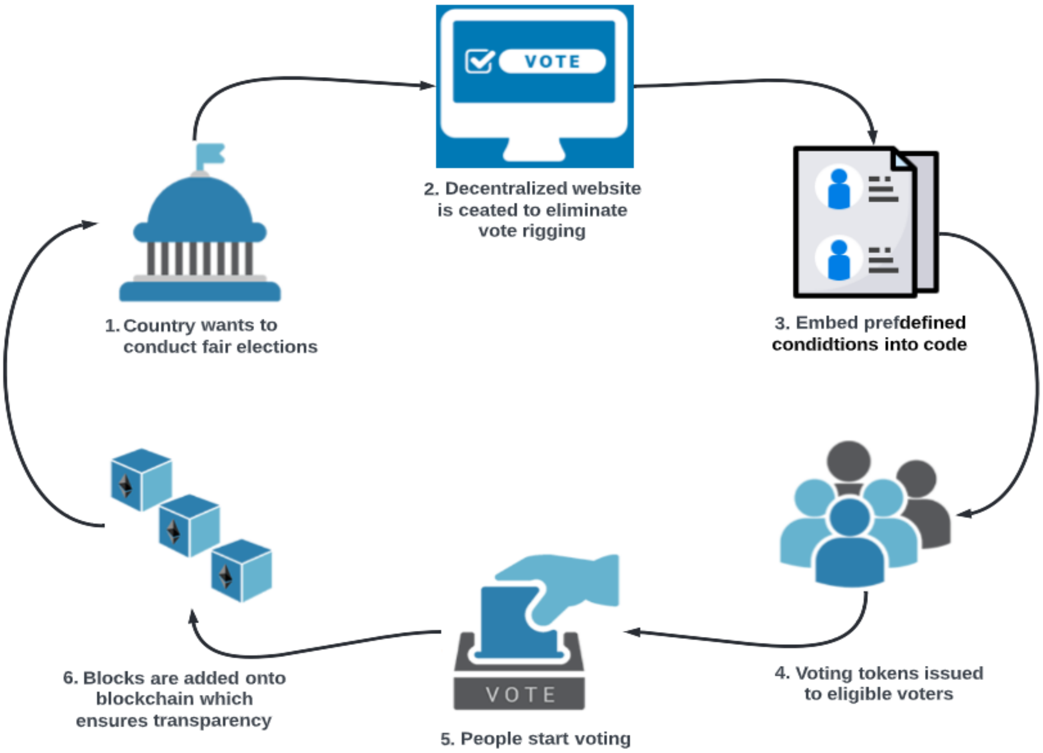

VOTE
2. Decentralized website is ceated to eliminate vote rigging
1. Country wants to conduct fair elections
3. Embed prefdefined condidtions into code
4. Voting tokens issued to eligible voters
VOTE
5. People start voting
6. Blocks are added onto blockchain which ensures transparency

Figure 4.1: Modules of Decentralized Voting

### 4.2.1 Context Diagram

D-voting is a web application with which the voter will interact through a metamask. The figure 4.2 illustrates that the user will log in and then will be authenticated. If the user is successfully authenticated, they will be able to see the list of candidates that are taking part in elections. The user will cast a vote for the candidate of their liking. After the vote has been cast, a transaction will be carried out. The transaction will be accepted when a fee in Ethers will be paid. Every transaction that will be made will be recorded in the ledger and the results can also be viewed. Another component of this diagram is a smart contract which will have the list of candidates and the voters in advance so that the voters can be authenticated as well as the candidates who are mentioned in the list are the ones receiving the votes.

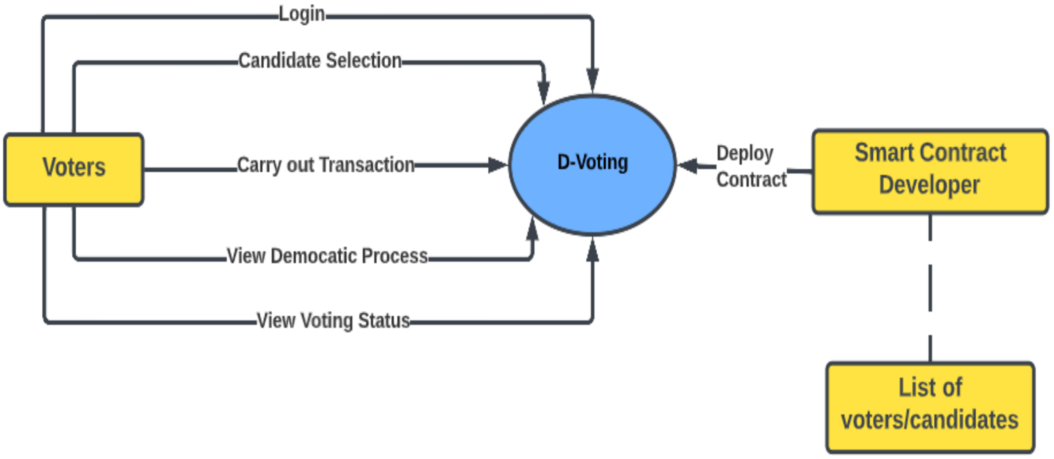


Figure 4.2: External components interacting with the system

## 4.3 System Activity

The activity diagram is a behavioural diagram, which means it depicts the system's behaviour. An activity diagram depicts the control flow from a start point to an endpoint, as well as the multiple decision pathways that exist while the activity is being conducted. When a user casts a vote, a transaction is initiated. The vote is broadcasted to the P2P network so it can be validated by all the blocks in the network. Once the transaction is validated by blocks, the new transaction is added as a new block to the block hain network. After the transaction is complete and validated, the results are updated as shown in figure 4.3.

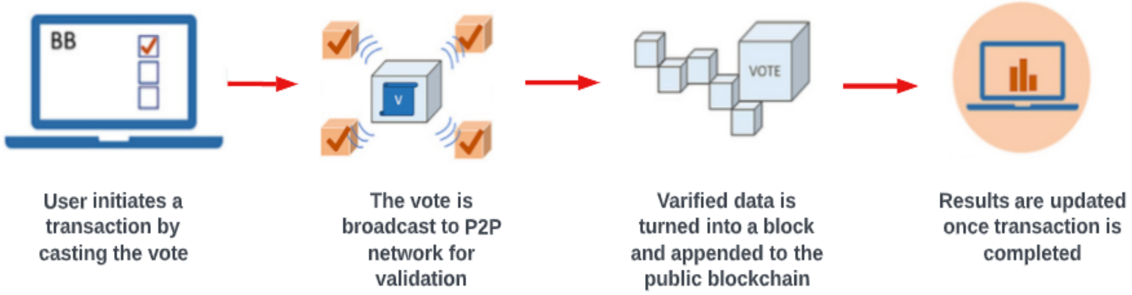

BB
V
VOTE
User initiates a transaction by casting the vote
The vote is broadcast to P2P network for validation
Varified data is turned into a block and appended to the public blockchain
Results are updated once transaction is completed

Figure 4.3: Flow of control in the system

### 4.3.1 Activity Diagram

The activity will start once Admin triggers any one of the two events, i.e., start election and end election. Initially, Admin needs to start the election. After that, the voters will be authenticated by checking if a hash value exists against the voters' names. If the hash value already exists, the voting button will be disabled and the voters will not be allowed to vote. If the hash value against the voter's name does not exist, the voting button will stay enabled and the voters will be able to cast the vote. Voters vote by selecting the appropriate candidate at first. If the voters do not have the fee that has to be paid to cast a vote or do not have an active internet connection, their vote will be nullified. However, if the voter has adequate fees in his/her wallet, then a transaction will be carried out. Once the transaction exits successfully, the candidate's vote count will be increased by one. This action will also update the status of the voter on the digital vote card. If Admin chooses to end the election, then the election will be ended such that no voter will be allowed to vote and the voting button will be disabled as illustrated in figure 4.4.

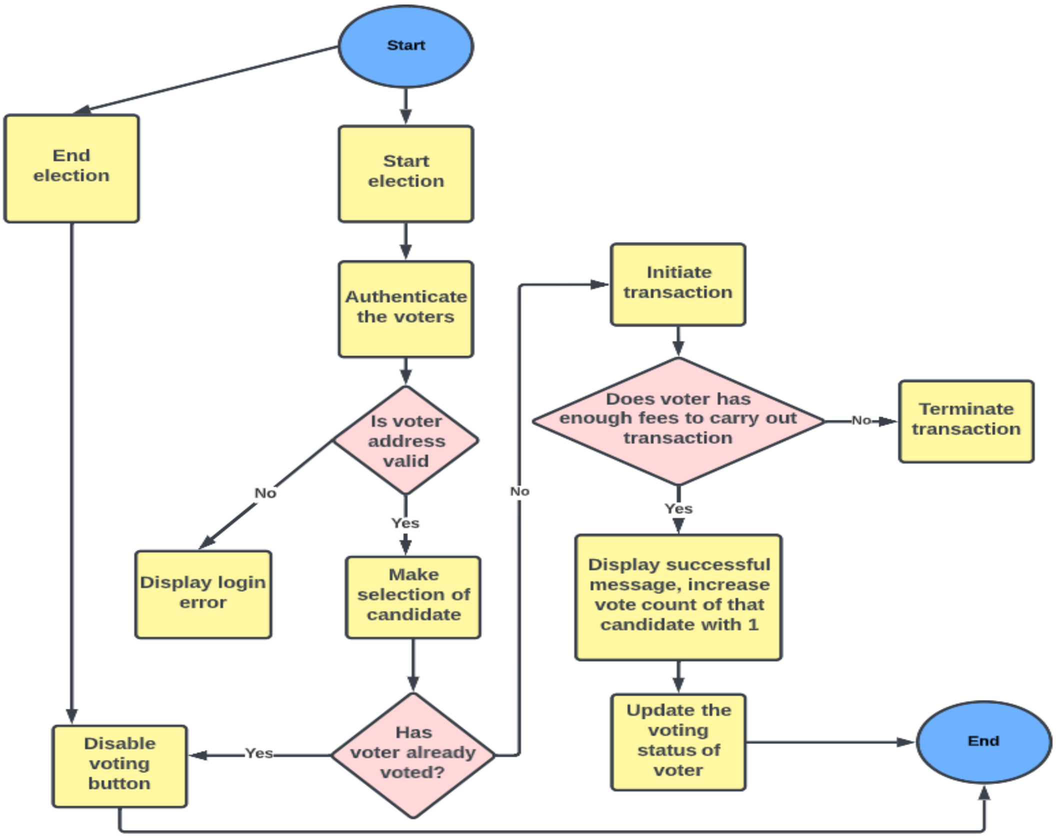


Figure 4.4: Process flows of proposed system

## 4.4 System Events

A sequence diagram, also known as an event diagram, simply displays the order in which the components interact with each other. Sequence diagram shows in which order and how the system component interact with each other.

### 4.4.1 Login Sequence Diagram

The figure 4.5 shows that the user interacts with the D-voting component. The user has a meta mask wallet. Without this wallet, the vote cannot be cast. The user will enter the login credentials and they will be verified by the smart contract. If the credentials are verified, the user will be successfully logged in or else an error message will be displayed.

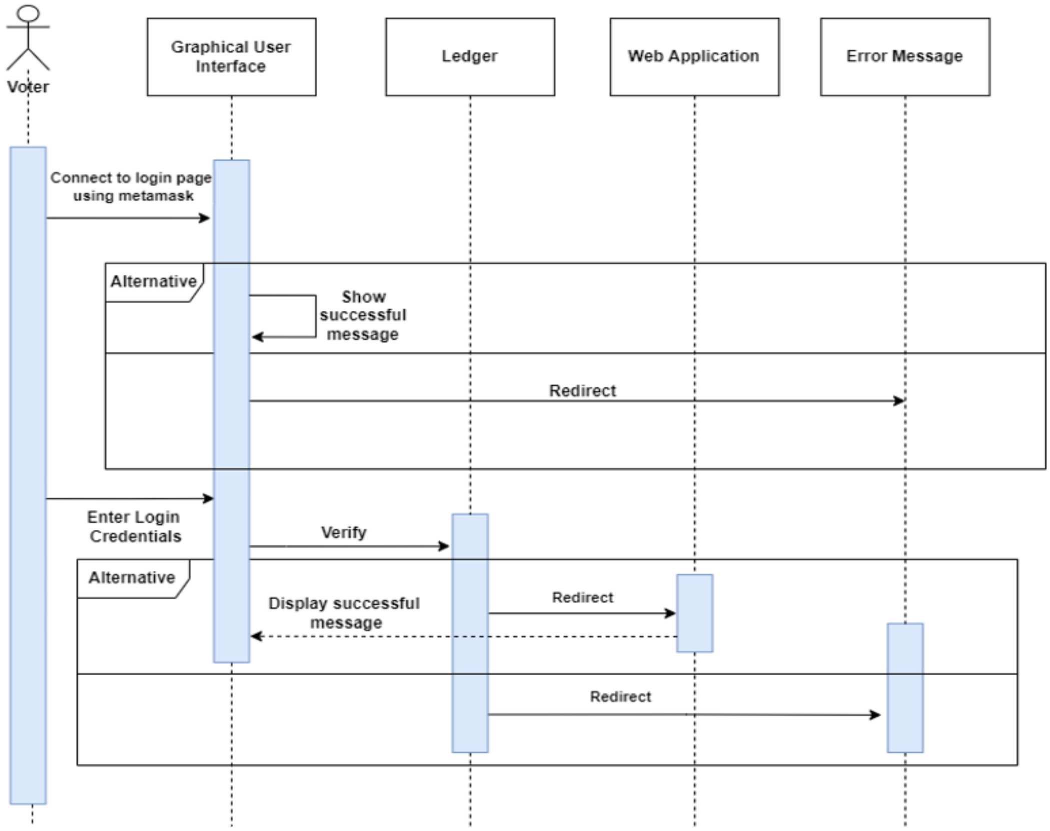


Figure 4.5: Sequence of Voter Login Module

### 4.4.2 Vote Sequence Diagram

The figure 4.6 depicts that the user will select the candidate, cast the vote and will be directed back to the web application. Basically in this process, the metamask will reappear and a transaction will be made. If the transaction will be successful, in the backend the vote count of the candidate will be increased by 1 after every successful transaction. At the user end after every successful transaction, the user will receive a notification that would say: “Your vote has been cast successfully.” After successfully casting vote, the vote will be counted. Similarly, if the meta mask transaction is unsuccessful, the vote cast will be unsuccessful and no data will be updated on the blockchain and the user will receive a notification saying that the vote is unsuccessful. The vote will not be counted.

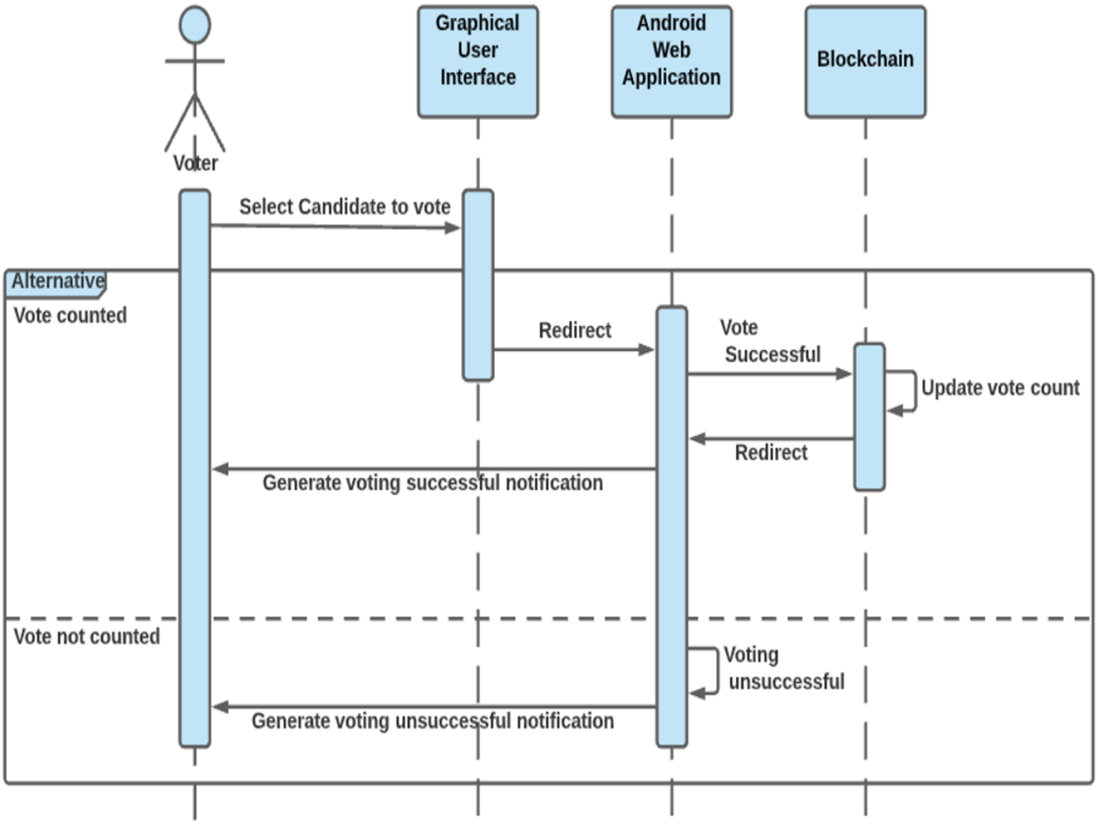


Figure 4.6: Sequence of Vote Casting Module

### 4.4.3 Result Sequence Diagram

The figure 4.7 depicts that after the specified voting period is over, the voter will click on the show result button on the application. This click will initiate a process at the backend. The list of candidates along with the number of votes they have received will be fetched from the blockchain and will be displayed on the user interface.

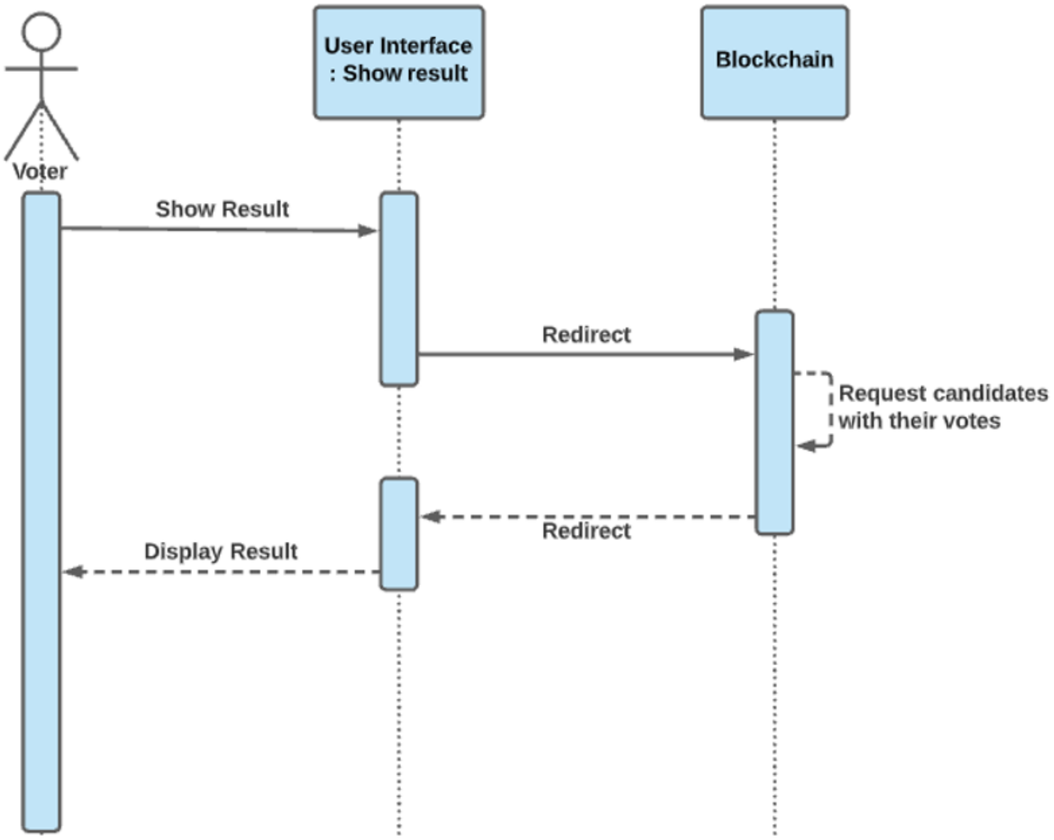


Figure 4.7: Sequence of View Results Module

## 4.5 Entity Relationship Diagram

An entity-relationship (ER) diagram, also known as an entity-relationship model, depicts the connections between entities. The most typical application of ER diagrams is to arrange data within databases or information systems.

Six typical symbols are used in a conceptual ER diagram.

- Entities are crucial data-representing objects or concepts. These entities, also known as strong entities or parent entities, are frequently dependent on weak entities.

- An entity's attributes are its characteristics (i.e., many-to-many or one-to-one).

- Relationships are the connections between things.

- Weak entities rely on the presence of another entity.

- Attributes with several values are known as multi-valued attributes.

- The links between a weak entity and its parent are known as weak relationships.

In this ER diagram, we have explained the 5 relationships that occur in our proposed system. The relationships are that the voters can cast votes to the candidates, voters register to election commission, candidates register to election commission and candidates receive votes. It is illustrated in figure 4.8.

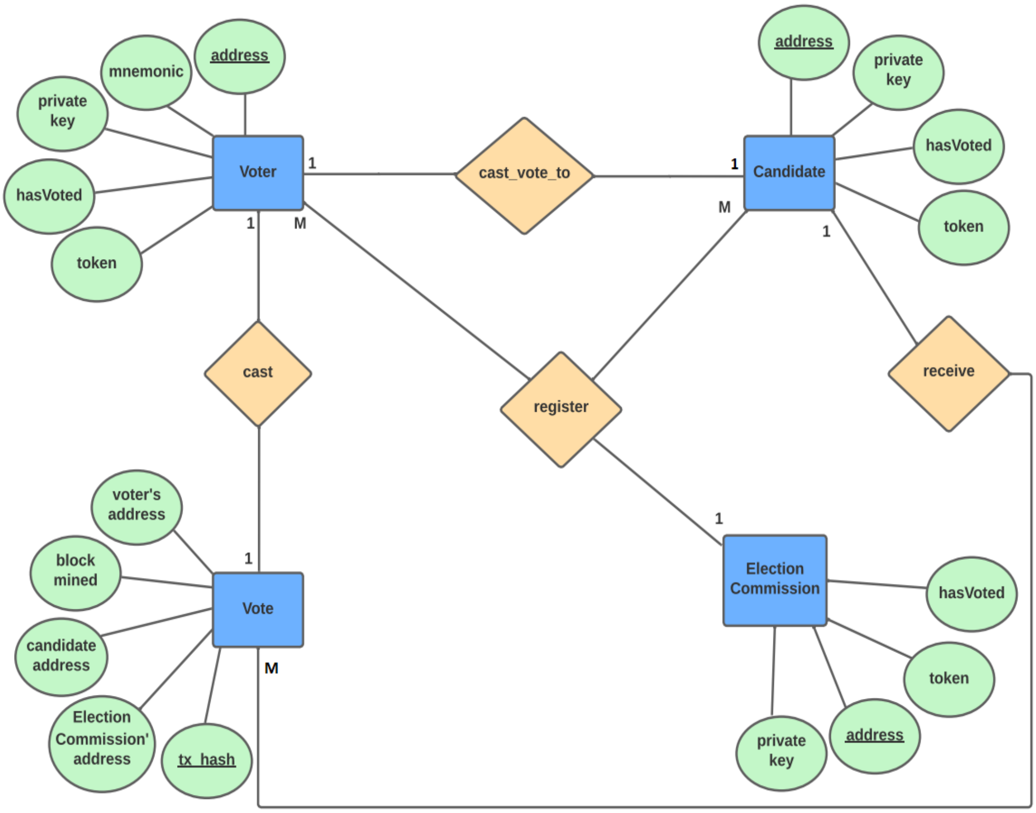


Figure 4.8: ERD illustrating Entity Relationship Model

## 4.6 PCAM Methodology

PCAM stands for partitioning, communication, agglomeration and mapping. It is used for methodically designing simple parallel algorithms and identifying design problems that impact efficiency or scalability. Most programming problems have multiple solutions. Existing sequential algorithms may not always provide the optimal answer. The design technique we describe is designed to encourage an experimental approach to design in which machine-independent issues like concurrency are tackled early in the design process and machine-specific features of the design are left until later.

### 4.6.1 PCAM Design

Foster's methodology structures the design process as four distinct stages:

- Partitioning.

- Communication,
- Agglomeration.
- Mapping.

### 4.6.2 Partitioning

Partitioning is dividing the computation and the data into tasks. Therefore, we divided the tasks into smaller chunks as much as possible. The tasks are as below:

- The poll organizer starts the voting.
- The poll organizer predefines the list of candidates.
- The poll organizer predefines the list of voters.
- The poll organizer creates a smart contract.
- Deployment of smart contract on Ethereum blockchain.
- Voters interaction with Web 3.0.
- Validation of voters by the system.
- Voters initiate the poll by casting a vote.
- Selection of candidates by voters.
- One block is responsible for one transaction within the blockchain therefore a new block is mined for every new voter.
- No more blocks are mined after the election is ended by the poll organizer.
- Data stored in the most recent block contains the final result.

The figure 4.9 explains the partitioning of proposed system according to PCAM Methodology.

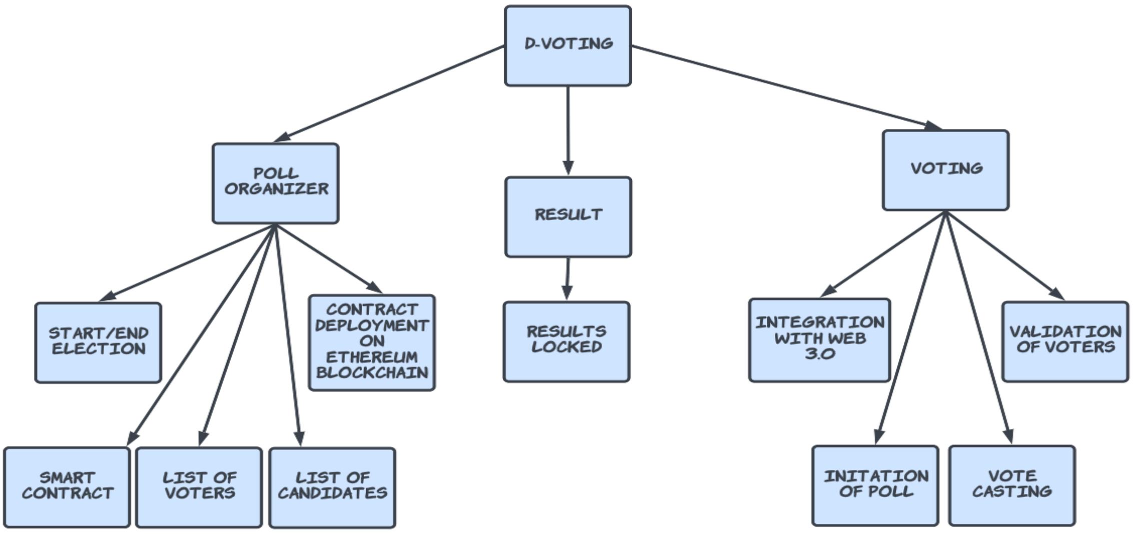


Figure 4.9: Functional decomposition of processes

### 4.6.3 Communication

It considers the plan for inter-process communication necessary for the parallel program. We need to identify the necessary communication between the fine-grained tasks to perform the necessary computation. For functional decomposition, this task is often relatively straightforward.

In our project, we believe that the process of voting can be parallelized because it involves inter-process communication to some extent. Each voter has a copy of the distributed ledger, such that all the changes will be visible to every other voter in real-time. The process of voting can be done concurrently as shown in figure 4.10.

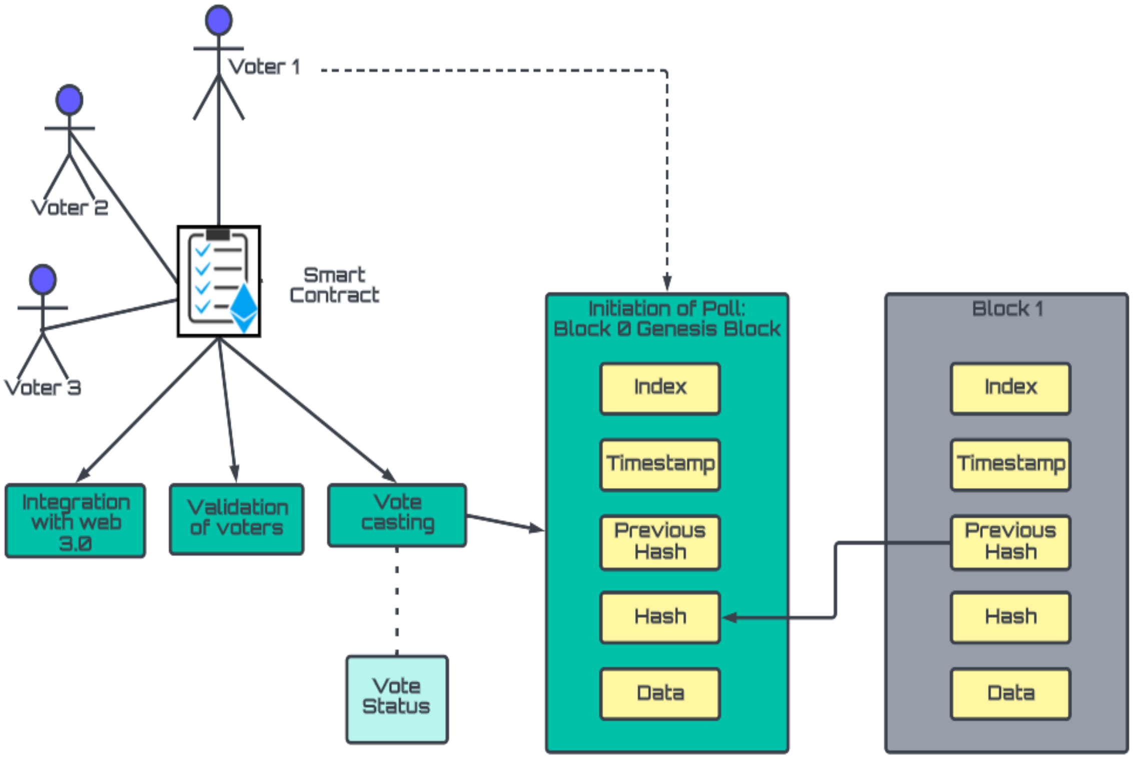


Figure 4.10: Communication between components of voting event

### 4.6.4 Agglomeration

Agglomeration is required to achieve data locality and good performance. It combines the many fine-grained tasks from partitioning into fewer coarse-grained tasks of larger size. It is shown in figure 4.11.

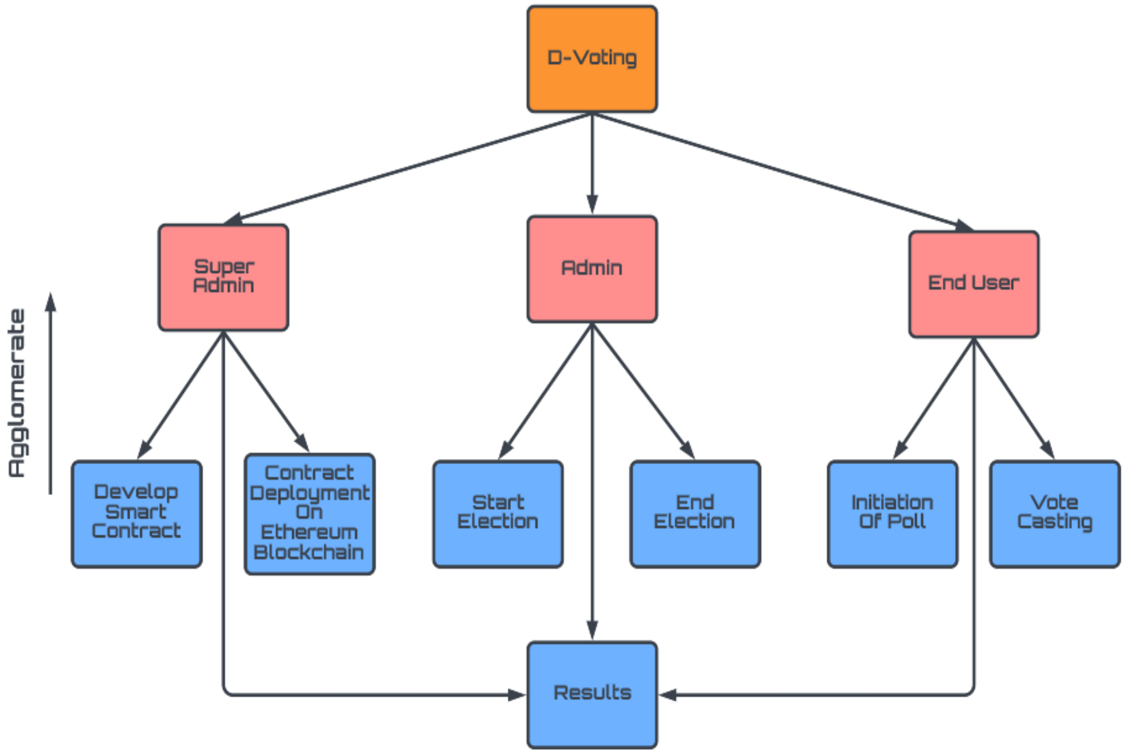


Figure 4.11: Agglomeration of processes

### 4.6.5 Replication of Data

By nature, blockchain supports this feature. Significant savings in communication can be made by replicating data. Once a smart contract is sent to all the nodes, we are reducing the dependency of nodes on one another. Newly added blocks to the chain have information about the previous block.

### 4.6.6 Mapping

The goal of mapping is to minimize total execution time. In general, we want to map tasks to achieve good load balancing. For problems involving functional decomposition, load balancing can be a very significant challenge. Complicated load balance algorithms often must be employed for this purpose. To do that, we must focus on making the code parallel first. Unfortunately, at this stage, we do not have the code related to our FYP.

Cloud computing allows nodes to operate across clusters of machines, thereby enabling increased transaction speed which supports parallel processing across the entire network. Blockchain-as-a-service (BaaS) refers to third-party cloud-based infrastructure and management for companies building and operating blockchain apps.

### 4.6.7 Conclusion

The above-proposed technique is our take on devising a parallel solution for our final year project. The project can be parallelized using this approach; better and more efficient results can be generated in a lesser period.

# Chapter 5

# System Implementation

In system implementation, all stages of deploying our logical designs into a physical environment using defined methods and methodology are included. During this phase, we develop a system design to meet the system's requirements as well as the decentralized voting mechanism. In this chapter, we'll go over all the tools, techniques, and procedures we used to implement prototypes that were developed based on system requirements during the system architecture phase.

## 5.1 System Architecture

Each network participant monitors, authorizes, and updates new entries in the blockchain architecture's distributed network. Blockchain's layered architecture is divided into five tiers which are the network layer, transaction layer, blockchain layer, trust layer, and application layer.

Inter-node communication is handled by the network layer, which is also known as the P2P layer. The data structure of a blockchain is described as a linked list of blocks in which transactions are arranged. The blockchain's data structure is made up of two basic components: pointers and a linked list. A linked list is a collection of linked blocks that include data as well as pointers to previous blocks. Transactions are digitally signed to ensure the security and integrity of the data contained in the blockchain. To sign transactions, a private key is utilized, and anyone holding the public key can verify the signer. Information modification is detected via the digital signature. The consensus layer is in charge of validating, ordering, and ensuring that everyone is in agreement. The application layer includes smart contracts, chain code, and decentralized apps (DApps). It includes scripts, application programming interfaces (APIs), user interfaces, and frameworks.

In a decentralized voting web application, a decentralized website provides an interface for the end-users to vote in the application. We have designed a user-friendly graphical user interface. Node JavaScript (Node.js) is used for the development of the GUI. We have used different images, styles, and buttons to make our GUI attractive and interactive. We used a local blockchain server maintained by Ganache.

The figure 5.1 shows the general view of different layers of blockchain.

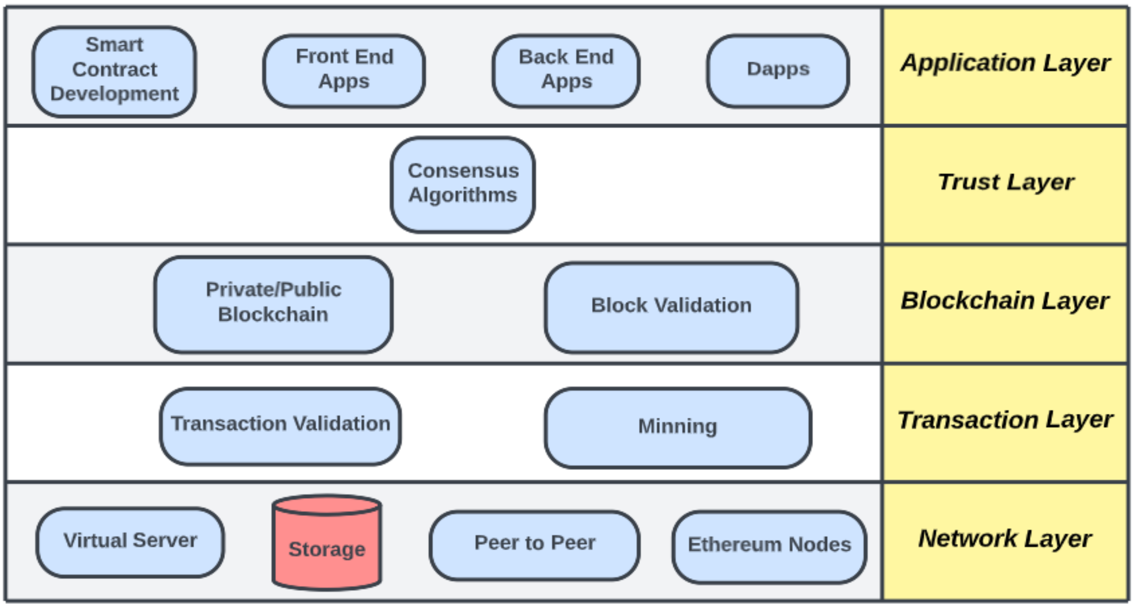


Figure 5.1: Five layers of system architecture

### 5.1.1 Proposed System Work flow

The proposed system development is compatible with a web application. The basic flow of our system is shown in Figure 5.2. Key activities of the system are mentioned below:

- Smart contract developer creates a smart contract and deploys it on Ethereum blockchain.
- Every functionality of the democratic process is written in solidarity language inside a smart contract.
- Voters interact with the application through metamask.
- Authorized voters can access the voting functionality of the smart contract and vote for their favorite candidate.
- To vote, a transaction must be carried out. Once that transaction is made successfully, the candidate's vote count will be updated.
- After the voting period is over, results will be displayed.

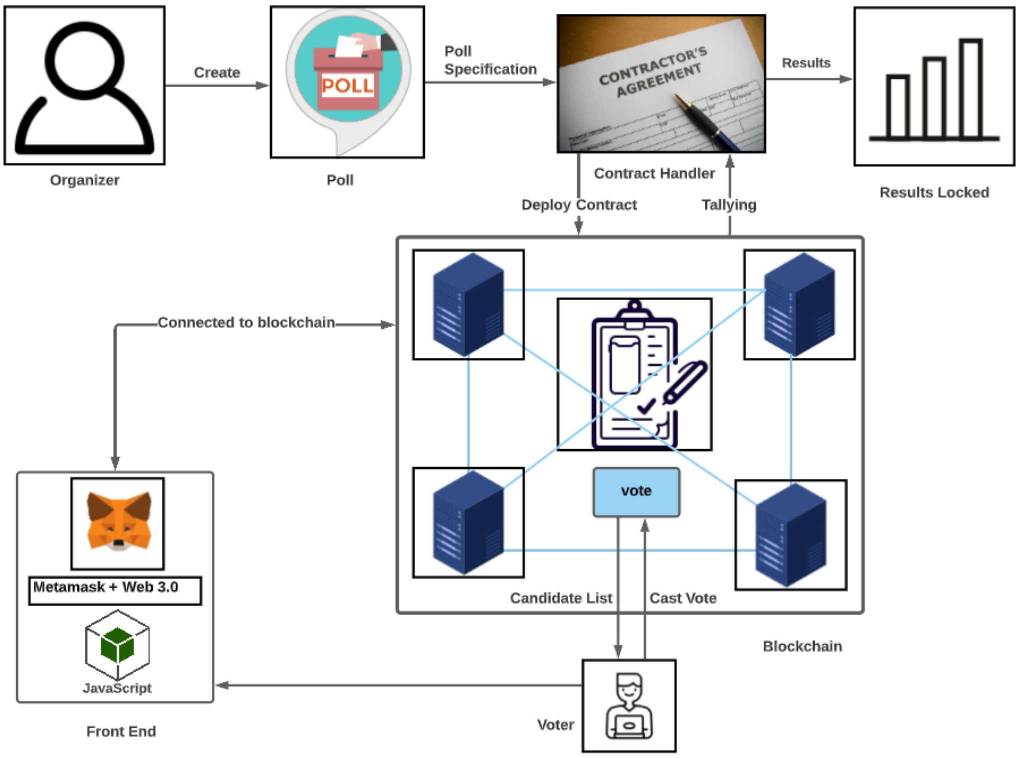


Figure 5.2: System architecture

## 5.2 System Modularization

The project has been separated into multiple components, with separate modules for each functionality. Any software consists of multiple systems, each of which has several sub-systems, each of which contains its own sub-systems. As a result, building a comprehensive system in one go that includes all essential capabilities is a time-consuming task that might result in several errors due to its huge size.
If the partitioned modules are separately solvable, modifiable, and compliant an effective modular design can be created. The project modules are as follows:

- **Election Commission:** In this module, the smart contract developer will be in charge of setting up the smart contract, registering candidates and parties provided by the Election Commission, and deploying contracts onto the blockchain.

- **Election Test:** This is the module for testing our smart contract, in which we use the Mocha Framework to unit test our application.

- **Voter Module:** Voters who have been given a personal ETH wallet will use the Metamask extension to import their wallets into the voting site and cast their vote in this module.

## 5.3 Ethereum Blockchain

One of the most well-known cryptocurrency ecosystems is Ethereum. Unlike Bitcoin, Ethereum allows every Ethereum wallet owner to install their own smart contract onto the Ethereum network to support their own business logic. To implement BlockVOTE on Ethereum, we must use the Ethereum Foundation's Truffle Framework for smart contract development.

Ethereum requires that a smart contract be written in the Solidity programming language. After the contract code is generated, it is compiled using the Truffle Framework's Solidity compiler tool. The Truffle Framework also provides a contract migration tool, which is used to deploy the built contract onto the Ethereum network. In our prototype, each voter must have their own voting application installed on their machine. The voting machine can be a computer, a smartphone, or a tablet that can connect to the Internet. The application interacts with the deployed contract using Web 3.0 and the Truffle JavaScript library.

Despite the fact that voting should be free, a user must spend some Ether (the Ethereum ecosystem's currency) in order to generate a new block, according to Ethereum platform specifications. The quantity of Ether required by a user is determined by the size of data contained within the created block. As a result, before casting a vote in our Ethereum-based voting program, each user must first connect their Ethereum wallet to the voting application. According to the automatic cost calculation on the Ethereum network, the cost per vote in the BlockVOTE application is 0.000652 ETH per vote.

## 5.4 Interface of the Application

To develop the decentralized voting application, we have used the Node JavaScript environment. Below is the list of the frameworks and tools which are used to develop web interfaces.

### 5.4.1 jQuery

jQuery is a JavaScript library that makes interactions between JavaScript code and HTML elements easier and more consistent. Websites may be interactive and dynamic using JavaScript, and jQuery is a utility that aids in this process. Some features of the jQuery are as follows:

- jQuery wraps a number of typical activities that require a lot of lines of JavaScript code into methods that can be called with only a single line of code.

- There are a lot of JavaScript libraries available but over 70 percent of the world's biggest companies use jQuery such as Google, Microsoft, IBM, Netflix, etc.

- The jQuery team is well-versed in cross-browser challenges, and this expertise has been included in the jQuery library. All major browsers will run jQuery in the same way.

### 5.4.2 BootStrap

Bootstrap is a free front-end framework that offers HTML and CSS-based design templates for typography, forms, buttons, tables, navigation, modals, picture carousels, and other features, as well as optional JavaScript plugins. Bootstrap also allows us to quickly construct responsive designs. Some of the features of the bootstrap are as follows:

- All current browsers are compatible with Bootstrap (Chrome, Firefox, Internet Explorer, Edge, Safari, and Opera).
- Bootstrap offers a vibrant community and extensive documentation, including live examples and templates.

### 5.4.3 HTML

The acronym HTML stands for Hypertext Markup Language. HTML is a markup language that is used to produce electronic texts (called pages) that are displayed on the Internet. Each page contains several hyperlinks, which are links to other pages. Every web page in our decentralized application was created using a single HTML version. Some features of HTML are as follows:

- HTML coding guarantees that text and images in your browser are formatted correctly. A browser would not be able to display text as elements or load images or other elements without HTML.
- One of the advantages of HTML is that it can incorporate programs written in a scripting language such as JavaScript, which affects the behavior and content of web pages.

### 5.4.4 REST API

Representational State Transfer (REST) is an architectural paradigm that favours basic HTTP calls over more complicated options like SOAP for inter-machine communication. When we use REST, our calls will be message-based, and the HTTP standard will be used to express these messages. A RESTful API sends a representation of the resource's state to the requester or endpoint when a client request is made. This data, or representation, is sent via HTTP in JSON format.

The Election Commission sets up elections and adds registered candidates along with the parties to contest the election. Election's rest API hosted on Ethereum blockchain is used to display election specifications on the frontend. GET, POST, PUT, and DELETE are the HTTP request methods used by the rest of the API. REST API is based on the server-client model, which allows the server and client to run and evolve independently.

### 5.4.5 Microsoft Visual Studio Code

Visual Studio Code is a stripped-down version of Microsoft's official programming environment that focuses solely on the code editor. Its cross-platform nature allows it to support a wide range of programming languages and syntaxes.

Visual Studio Code supports highlights, auto-indents, snippets, and auto-complete in a variety of languages, including HTML, CSS, JavaScript, several varieties of C, JSON, Java, SQL, PHO, Ruby, Visual Basic, and many others. Beyond support for Git repositories and the ability to open several file iterations in one window, the environment isn't fancy and focuses solely on giving flexibility and simplicity to promote interoperability across the platforms supplied. Some of the features of Microsoft Visual Studio Code are as follows:

- The fact that VS Code is almost entirely open source is a huge plus. This feature helps to increase community engagement.
- Visual Studio Code is built using Electron which is a framework for generating desktop programs using JavaScript utilizing Chromium and Node.js.

### 5.4.6 Node.js

Node.js is a programming language that is used to create networking and server-side applications. We create JavaScript apps in Node.js, an open-source, cross-platform run-time environment. On Windows, Linux, and OS X, such a program can run within its run time. The following are some of Node.js's features:

- Its libraries are asynchronous since servers can't wait for API responses; instead, they call APIs and move on with Node.js. Previous API responses are handled by the event mechanism.
- Its libraries execute code quickly thanks to Google Chrome's V8 JavaScript Engine.
- It uses a single-threaded approach, but due to its event system, servers become far more scalable than standard servers that handle requests with a restricted number of threads.
- Data chunks are returned by the Node.js application, therefore there is no data buffering.

## 5.5 Environment Language Used

### 5.5.1 Javascript

JavaScript is an interpreted client-side scripting language that enables web designers to include code in their pages. JavaScript is the most popular programming language today, and it is typically embedded in HTML or ASP files and runs immediately from the web page. JavaScript can execute more complicated functions than

HTML, such as printing the time and date, building a calendar, and other things.

The following are some of the benefits of using JavaScript:

- We can validate user input before sending the page to the server, resulting in less server interaction. This reduces server traffic, resulting in lower server strain.
- Visitors get immediate feedback instead of having to wait for a page reload to discover if they've forgotten something.
- You may make interfaces that respond when the user moves their mouse over them or activates them with the keyboard.
- To provide a Rich Interface to our site users, we utilised JavaScript to incorporate features like drag-and-drop components and sliders.

## 5.6 Coding Standardization

The system design as discussed in chapter 4 is converted into code, which is a machine-readable format. It essentially converts a human-readable format to a machine-readable format. The efficiency of our code that has been transformed from the system design is referred to as coding standardization. The efficiency is mostly determined by:

- Readability:

  The code should be readable, with suitable indentation and space, so that all modules' contents are evident.

- Portability:

  The code is portable enough to run on a variety of platforms provided that all of the required dependencies are installed.

- Easily Debug:

  As far as possible, the coding should be error-free.

## 5.7 Conclusion

Initially, we tried to implement our proposed project using React Native, but we couldn't get the desire results. Afterwards we decide to make our website using JavaScript and its libraries. The usage of local blockchain maintained by Ganache saved us a lot of time. Even after the completion of the project, we cannot deploy it on Ethereum main-net because it costs a lot. Therefore, we are using personal blockchain. Other benefit that local blockchain provided us was that it was quick to response. Similarly, jQuery helped us to use built-in functions. We used those functions at certain occasions to fulfill our tasks. We used bootstrap and CSS for styling. We found it easier to make web pages using Html and CSS as lot of related work is available on the internet and we were quick to get our hands on whatever was required. We have tried to include the list of voters and candidates within the domain of blockchain but we couldn't do it in time. We will try to extend the domain of blockchain in the future.

# Chapter 6

# System Testing and Evaluation

This chapter will go over the testing and assessment of our decentralized voting application. For testing our application, we used a variety of methods. The main purpose of our application is to provide rigging free elections and to provide the ability for our voters to cast votes from anywhere in the World. We have used a personal blockchain called "ganache" for this purpose. This online application provides a user-friendly and adaptable environment in which voters can vote, check their vote status, and examine the voting tally at any time. The testing and evaluation process here comprises the testing of the components, functionalities, and modules that we constructed throughout the development phase. We can also check the progress of our application's design in this phase by comparing these modules and components to the software requirement phase's specifications.

## 6.1 Graphical User Interface Testing

To test the interface of our web application, we have used the graphical user interface technique i.e., manual testing. We have used several images in our web application. Therefore, we needed to make sure that they are completely visible on different browsers. We did so by using the ".jpg" type images as they were compatible with different browsers. Moreover, the ".jpg" type images can be easily modified to different widths and heights. We used buttons to carry out the important functionalities of the application as well as to navigate to different pages. Other actions that we performed to ensure specifications are as below.

- We have tested the working of all the application's buttons, icons, menus, and QR code.
- We used dialogue/pop-up messages to make sure that messages are displayed accurately.
- We aligned the text according to the pages and images.

We believe that graphical user interface testing meets the design specification goals and increases the reliability of the application.

## 6.2 Usability Testing

Usability testing helped us in improving customer satisfaction because "Decentralized Voting" focuses on reliability and customer satisfaction. Voters need to perform only one action to vote. Keeping that in mind, we knew that the user interaction with the application will be very limited. Therefore, we designed our application in a way that users with minimum technical knowledge can also use it with ease. The core functions of our application i.e., starting election and casting vote are very crucial to the democratic process so we tested them multiple times. To our satisfaction, the core functions work properly.

## 6.3 Compatibility Testing

To perform compatibility testing, we used cross-browser testing also known as browser testing. This type of testing helps us in discovering if our application works properly across the different browsers such as Google Chrome, Microsoft Edge, and Mozilla Firefox. Similarly, they should work properly on browsers on different devices such as laptops, iPhones, Androids, etc.

## 6.4 Application Performance Testing

The democratic process should be carried out as soon as possible to make sure that all voters get an opportunity to vote as well as the vote count is locked within 24 hours. To meet these requirements, the performance of our application should be fast.

To perform application performance testing, we used the endurance testing technique. This technique helped us to evaluate if transactions extended the specified amount of time. The vote count on two screens i.e., Admin Screen and Cast Vote screen is updated immediately after the vote is cast. The dialogue boxes waste no time to appear once crucial functions are invoked. This testing technique ensures that our application performance is good.

## 6.5 Exception Handling

Exception handling is performed to deal with runtime errors. In our case, these are the errors that might arise due to the wrong actions of the voters. We handled the possible exceptions in our applications with the following exception handling techniques:

- User cannot cast vote until or less he/she has imported the right account with adequate fees.
- Once the user has cast the vote, the voting button disappears and the message "You have already voted!" appears in its place.
- The status of the user vote is updated to "Yes" once he/she casts the vote.

## 6.6 Test Cases

A test case is a set of scenarios used to assess a specific feature of a software product to see if it meets the business requirements.

### 6.6.1 Smart Contract Deployment

| | |
|---|---|
| Test Case | TC1. |
| Test Function | Smart Contract Deployment. |
| Initial State | The creation of Migrations.sol, truffle-config.js in the source folder, and the set-up of metamask. |
| Test Setup | Two contracts should be deployed on the ganache local network. |
| Input | Provide correct values in truffle-config.js. |
| Expected Output | Display of successful deployment message on the console. |
| Actual Output | The message is displayed on the terminal and the contract is successfully deployed. |
| Status | Pass. |

Table 6.1: Smart Contract Deployment

### 6.6.2 Smart Contract Erroneous Deployment

| | |
|---|---|
| Test Case | TC2. |
| Test Function | Smart Contract Erroneous Deployment. |
| Initial State | The creation of Migrations.sol, truffle-config.js in the source folder, and the metamask wallet is set up. |
| Test Setup | Two contracts should not be deployed on the ganache local network. |
| Input | Provide incorrect values in truffle-config.js or the metamask wallet. |
| Expected Output | Display of unsuccessful deployment message on the console. |
| Actual Output | The message is displayed on the terminal and the contract is not deployed. |
| Status | Pass. |

Table 6.2: Smart Contract Erroneous Deployment

### 6.6.3 Browser Syncing Configuration

| | |
|---|---|
| Test Case | TC3. |
| Test Function | Browser Sync Configuration. |
| Initial State | Contracts are deployed on the local network. |
| Test Setup | The browser should be opened. |
| Input | Type “run npm dev” on the console to launch the application. |
| Expected Output | The browser should be opened to show the initial functionalities of the application. |
| Actual Output | The browser opened with a proper display of the main screen. |
| Status | Pass. |

Table 6.3: Browser Syncing Configuration

### 6.6.4 Browser Desynced Configuration

| | |
|---|---|
| Test Case | TC4. |
| Test Function | Browser Desynced Configuration. |
| Initial State | Contracts are deployed on the local network. |
| Test Setup | The browser should not be opened. |
| Input | Run npm run dev on the console to launch the application. |
| Expected Output | The browser should not be opened. |
| Actual Output | The browser has not opened, and the main screen has not been displayed. |
| Status | Pass. |

Table 6.4: Browser Desynced Configuration

### 6.6.5 Start Election

| | |
|---|---|
| Test Case | TC5. |
| Test Function | Start Election. |
| Initial State | Admin Screen should be on display. |
| Test Setup | The browser should not be opened. |
| Input | Click on the “Start Election” button. |
| Expected Output | The voting button is enabled, and users can vote. |
| Actual Output | A dialog box pops up which displays the successful message. |
| Status | Pass. |

Table 6.5: Start Election

### 6.6.6 Cast Vote Successfully

| | |
|---|---|
| Test Case | TC6. |
| Test Function | Cast Vote Successfully. |
| Initial State | Contracts are deployed on the local network; elections are started by the admin such that the voting button is enabled. |
| Test Setup | Users can vote from the provided entities. |
| Input | Select any candidate and vote. While performing the transaction, provide adequate gas fees to carry out the transaction. |
| Expected Output | The vote has been cast successfully and the vote count of the selected candidate has been increased by one. |
| Actual Output | The metamask wallet pops up to carry out the transaction. Afterward, a dialog box pops up which displays the successful message. The vote of the selected candidate has been increased by one. |
| Status | Pass. |

Table 6.6: Cast Vote Successfully

### 6.6.7 Cast Vote Unsuccessfully

| | |
|---|---|
| Test Case | TC7. |
| Test Function | Cast Vote Unsuccessfully. |
| Initial State | Contracts are deployed on the local network; elections are started by the admin such that the voting button is enabled. |
| Test Setup | Users can vote from the provided entities. |
| Input | Select any candidate and vote. While performing the transaction, provide inadequate gas fees to carry out the transaction. |
| Expected Output | The vote couldn't be cast, and the vote count of the selected candidate remains the same. |
| Actual Output | The metamask wallet pops up to carry out the transaction. The transaction fails due to inadequate fees. The vote of the selected candidate remained the same. |
| Status | Pass. |

Table 6.7: Cast Vote Unsuccessfully

### 6.6.8 End Election

| | |
|---|---|
| Test Case | TC8. |
| Test Function | End Election. |
| Initial State | Contracts are deployed on the local network. |
| Test Setup | Admin Screen should be on display. |
| Input | Click on the “End Election” button. |
| Expected Output | The voting button is disabled, and users cannot vote. |
| Actual Output | A dialog box pops up which displays the successful message, and the voting button is disabled. |
| Status | Pass. |

Table 6.8: End Election

### 6.6.9 Digital Vote Card

| | |
|---|---|
| Test Case | TC9. |
| Test Function | Digital Voting Card. |
| Initial State | Contracts are deployed on the local network. |
| Test Setup | The digital voting card screen should be on display. |
| Input | None. |
| Expected Output | If the user has cast the vote, then display yes. Otherwise display no. |
| Actual Output | An address is on display belonging to a particular user. Status is displayed as yes/no depending on the earlier actions of the user. |
| Status | Pass. |

Table 6.9: Digital Vote Card

# Chapter 7

# Conclusion

To summarise, we designed our Final Year Project, D-Voting, with the goal of providing convenience to the people of Pakistan, as well as Pakistanis living abroad, by bringing this new technology into our country. We have put a lot of effort and time into developing this application and report. We concentrated on learning the new technology and using it to complete our Final Year Project.

Trusted elections are essential for a strong democracy, and citizens should have faith in the electoral system. Traditional paper-based elections, on the other hand, are untrustworthy. In today's society, the idea of adapting the decentralized voting technique is to make the political process reliable, faster, and easier. Making the voting process inexpensive and quick normalizes it in the eyes of voters, lowers a power barrier between the voter and the elected official, and puts pressure on the elected official. It also allows for a more direct form of democracy by letting people voice their preferences on specific bills and initiatives.

This project is a blockchain-based decentralized voting system that uses smart contracts to allow secure and rigging-free elections while maintaining voter anonymity. It includes a description of the system's architecture, design, and security analysis.

## 7.1 Improvements for future

We successfully designed and launched our decentralized voting application, but there are some features that can be added later to enhance the system's functionality. The following are examples of such characteristics:

- To be able to conduct more than one type of election.
- Include voters' authentication within the domain of blockchain.
- To be able to add more candidates.

# Appendix A

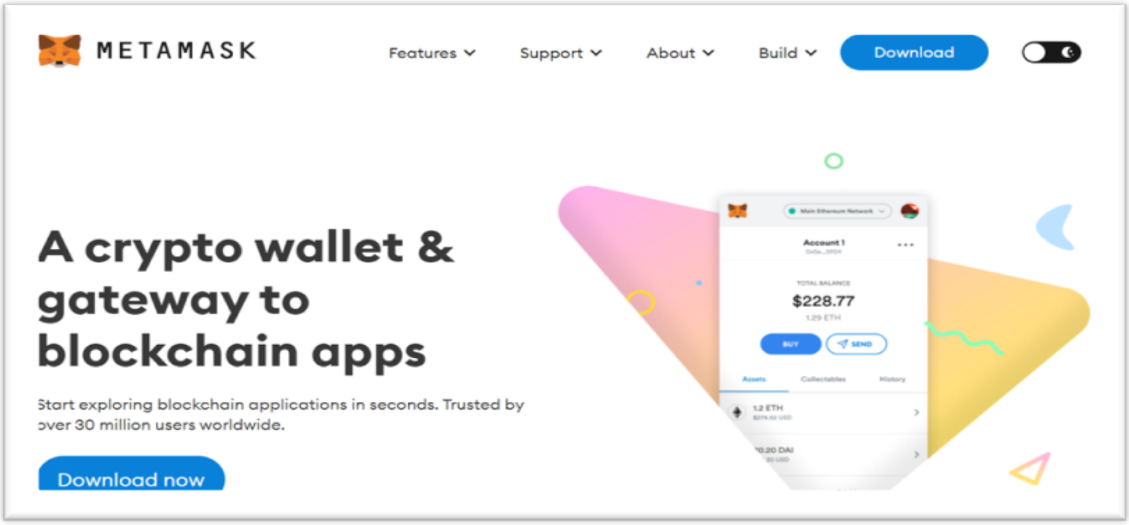
# User Manual

# Introduction

A user guide, often known as a user handbook, is a document that instructs users on how to use a product, service, or application.

## A.1 Metamask Installation

The very first thing that users need to do is to install metamask. It is a browser extension that acts as an Ethereum wallet and can be installed just like any other extension.



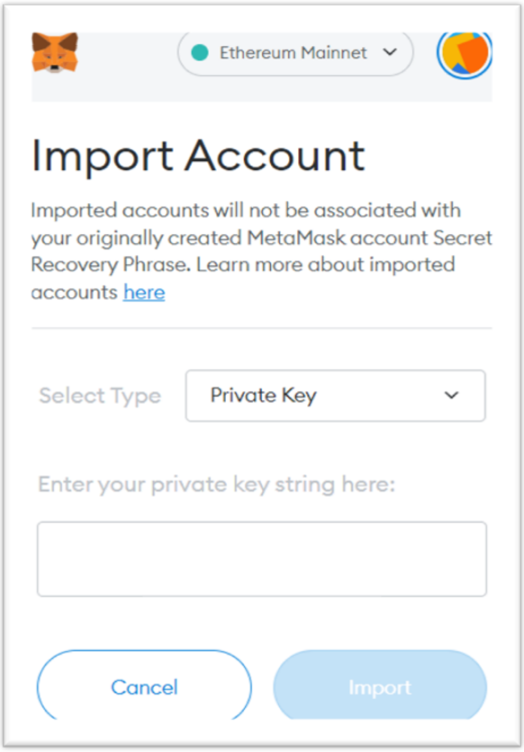
A crypto wallet & gateway to blockchain apps

Start exploring blockchain applications in seconds. Trusted by over 30 million users worldwide.

Figure A.1: Metamask Installation

## A.2 Import Account

Once metamask is installed, users need to import their accounts in the wallets. The account ID is provided by Election Commission.



Import Account

Imported accounts will not be associated with your originally created MetaMask account Secret Recovery Phrase. Learn more about imported accounts here

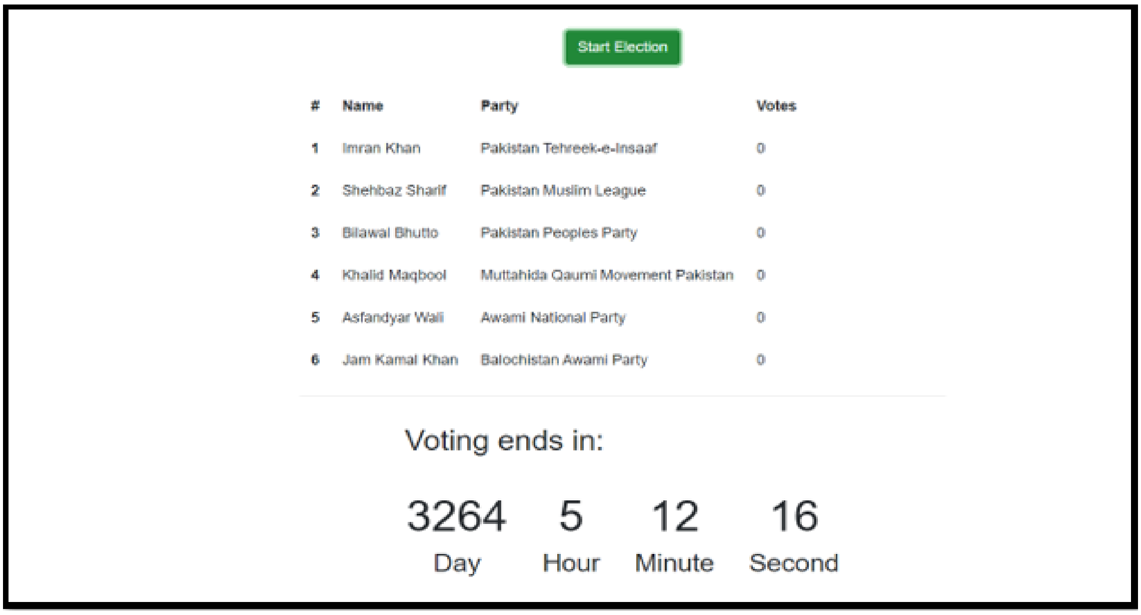
Select Type: Private Key

Enter your private key string here:

Figure A.2: Import Account

## A.3 Admin Page

The admin page has two buttons i.e. start election and stop election along with the list of all the candidates. Only admin can start the elections and see these buttons while the users can only see the list of candidates.



| # | Name | Party | Votes |
|---|---|---|---|
| 1 | Imran Khan | Pakistan Tehreek-e-Insaaf | 0 |
| 2 | Shehbaz Sharif | Pakistan Muslim League | 0 |
| 3 | Bilawal Bhutto | Pakistan Peoples Party | 0 |
| 4 | Khalid Maqbool | Muttahida Qaumi Movement Pakistan | 0 |
| 5 | Asfandyar Wali | Awami National Party | 0 |
| 6 | Jam Kamal Khan | Balochistan Awami Party | 0 |

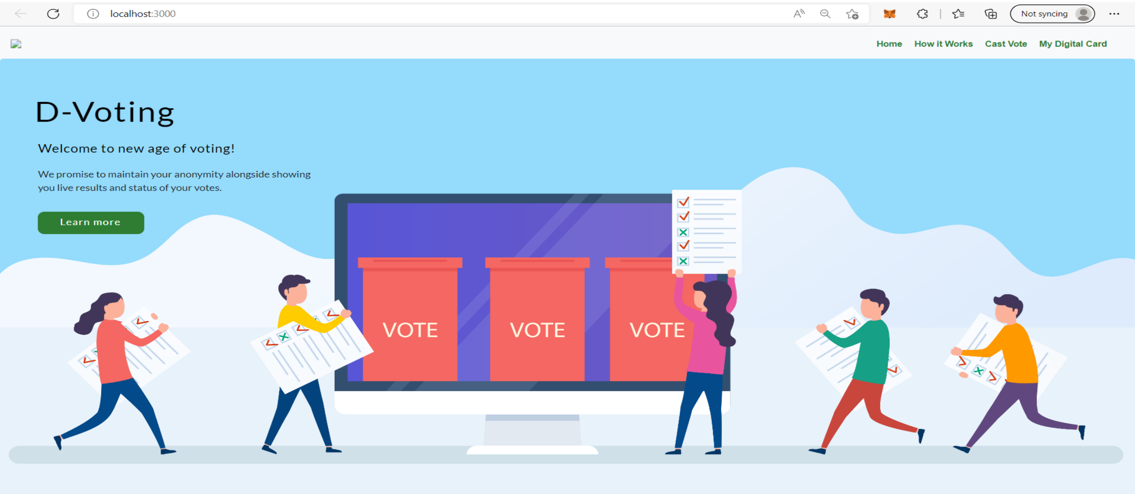
Voting ends in:

3264 Day 5 Hour 12 Minute 16 Second

Figure A.3: Admin Page

## A.4 Home Page

When a URL is loaded, the first page that voters see is the Home page.



D-Voting

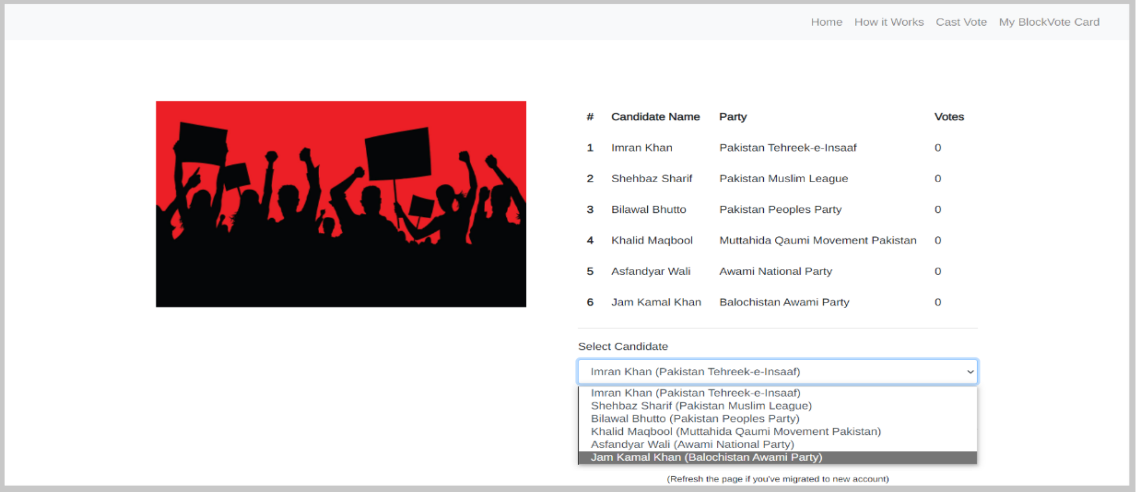
Welcome to new age of voting!

We promise to maintain your anonymity alongside showing you live results and status of your votes.

Figure A.4: Home Page

## A.5 Candidate Selection

This page shows the list of candidates to the voters from which they can select the candidate to whom they want to cast vote.



| # | Candidate Name | Party | Votes |
|---|---|---|---|
| 1 | Imran Khan | Pakistan Tehreek-e-Insaaf | 0 |
| 2 | Shehbaz Sharif | Pakistan Muslim League | 0 |
| 3 | Bilawal Bhutto | Pakistan Peoples Party | 0 |
| 4 | Khalid Maqbool | Muttahida Qaumi Movement Pakistan | 0 |
| 5 | Asfandyar Wali | Awami National Party | 0 |
| 6 | Jam Kamal Khan | Balochistan Awami Party | 0 |

Select Candidate

Imran Khan (Pakistan Tehreek-e-Insaaf)

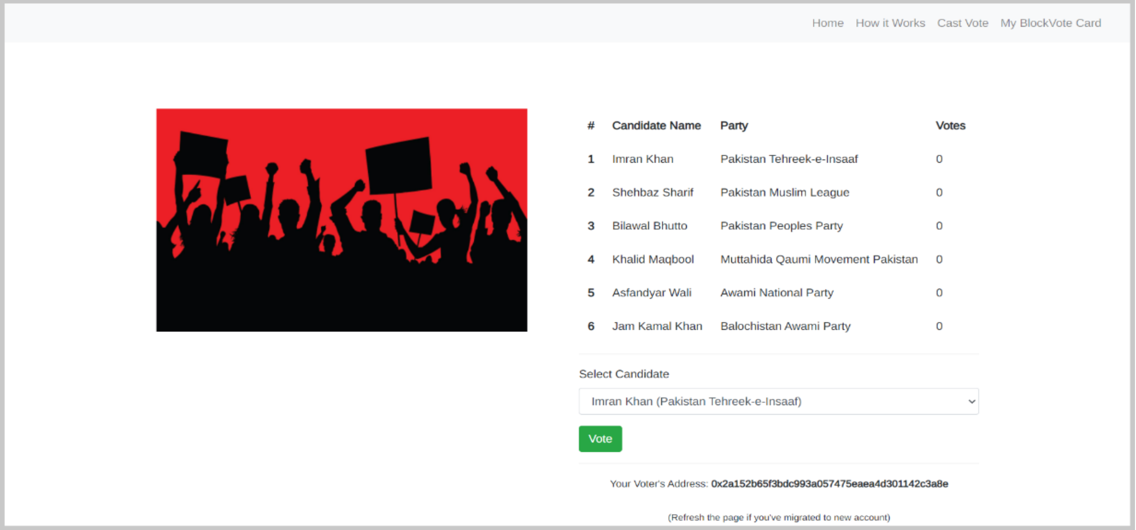
Imran Khan (Pakistan Tehreek-e-Insaaf)
Shehbaz Sharif (Pakistan Muslim League)
Bilawal Bhutto (Pakistan Peoples Party)
Khalid Maqbool (Muttahida Qaumi Movement Pakistan)
Asfandyar Wali (Awami National Party)
Jam Kamal Khan (Balochistan Awami Party)

(Refresh the page if you've migrated to new account)

Figure A.5: Candidate Selection

## A.6 Case Vote Page

The voters can cast vote through the following page.



| # | Candidate Name | Party | Votes |
|---|---|---|---|
| 1 | Imran Khan | Pakistan Tehreek-e-Insaaf | 0 |
| 2 | Shehbaz Sharif | Pakistan Muslim League | 0 |
| 3 | Bilawal Bhutto | Pakistan Peoples Party | 0 |
| 4 | Khalid Maqbool | Muttahida Qaumi Movement Pakistan | 0 |
| 5 | Asfandyar Wali | Awami National Party | 0 |
| 6 | Jam Kamal Khan | Balochistan Awami Party | 0 |

Select Candidate

Imran Khan (Pakistan Tehreek-e-Insaaf)

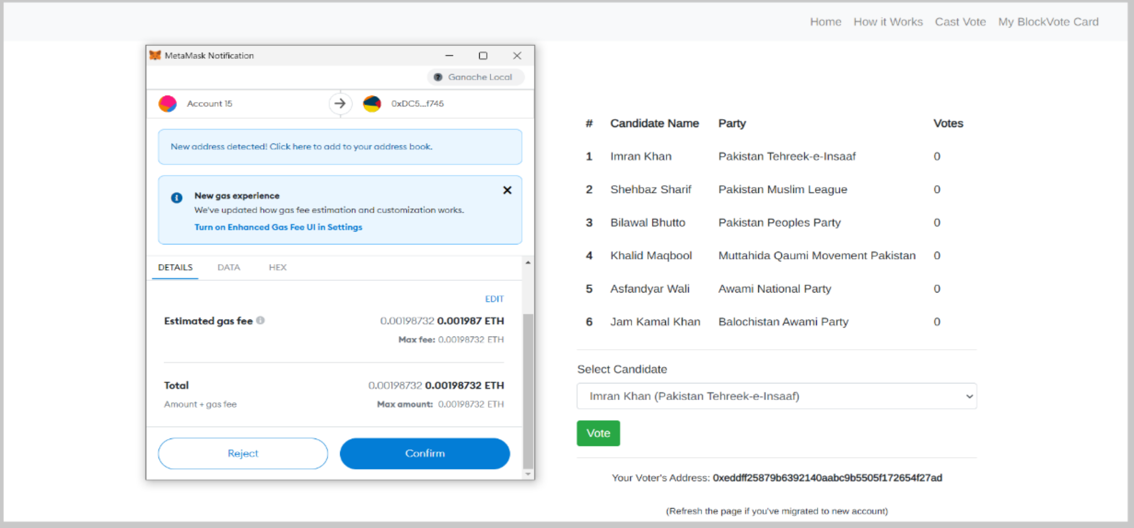
Your Voter's Address: 0x2a152b65f3bdc993a057475eaea4d301142c3a8e

(Refresh the page if you've migrated to new account)

Figure A.6: Cast Vote Page

## A.7 Transaction

Transaction will be carried out through a meta mask if the voter has an adequate gas fee.



MetaMask Notification

Ganache Local

Account 15 → 0xDC5...f745

New address detected! Click here to add to your address book.

**New gas experience**
We've updated how gas fee estimation and customization works.
**Turn on Enhanced Gas Fee UI in Settings**

**Estimated gas fee** 0.00198732 **0.001987 ETH**
**Max fee:** 0.00198732 ETH

**Total** 0.00198732 **0.00198732 ETH**
Amount + gas fee **Max amount:** 0.00198732 ETH

| # | Candidate Name | Party | Votes |
|---|---|---|---|
| 1 | Imran Khan | Pakistan Tehreek-e-Insaaf | 0 |
| 2 | Shehbaz Sharif | Pakistan Muslim League | 0 |
| 3 | Bilawal Bhutto | Pakistan Peoples Party | 0 |
| 4 | Khalid Maqbool | Muttahida Qaumi Movement Pakistan | 0 |
| 5 | Asfandyar Wali | Awami National Party | 0 |
| 6 | Jam Kamal Khan | Balochistan Awami Party | 0 |

Select Candidate: Imran Khan (Pakistan Tehreek-e-Insaaf)

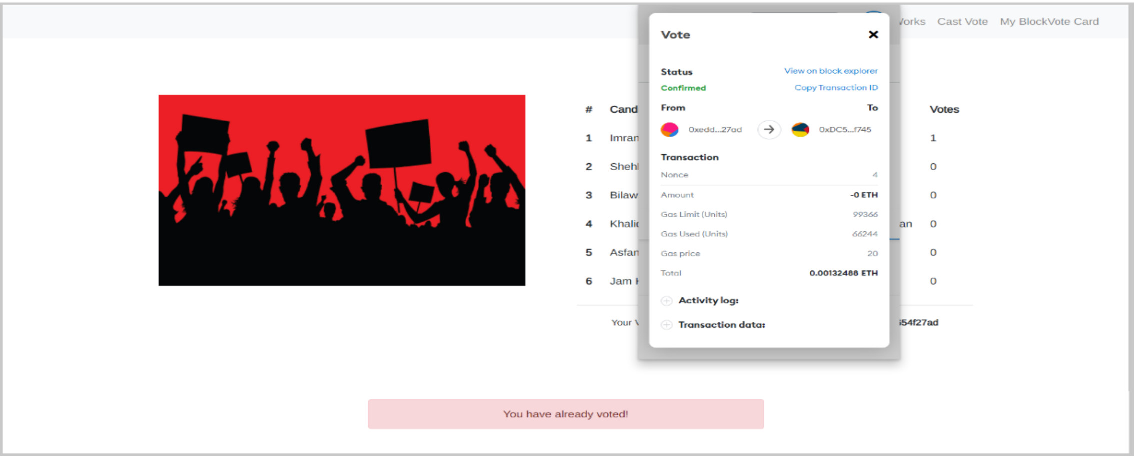
Your Voter's Address: **0xeddff25879b6392140aabc9b5505f172654f27ad**

(Refresh the page if you've migrated to new account)

Figure A.7: Carrying Transaction

## A.8 Transaction Confirmation by Miners

The transaction will be successful if voter has adequate fee in their account and the transaction history can be retrieved through the meta mask.



**Vote**

**Status** View on block explorer
Confirmed Copy Transaction ID

**From** 0xedd...27ad → **To** 0xDC5...f745

**Transaction**

| | |
|---|---|
| Nonce | 4 |
| Amount | **-0 ETH** |
| Gas Limit (Units) | 99366 |
| Gas Used (Units) | 66244 |
| Gas price | 20 |
| Total | **0.00132488 ETH** |

⊕ **Activity log:**

⊕ **Transaction data:**

| # | Cand | | Votes |
|---|---|---|---|
| 1 | Imran | | 1 |
| 2 | Shehl | | 0 |
| 3 | Bilaw | | 0 |
| 4 | Khalic | an | 0 |
| 5 | Asfan | | 0 |
| 6 | Jam I | | 0 |

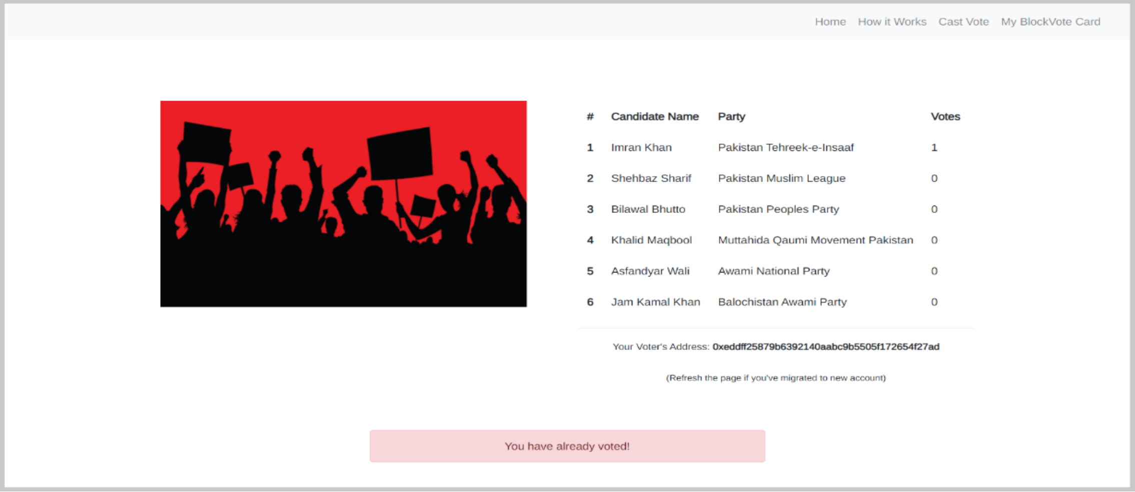
Your \ ... i54f27ad

You have already voted!

Figure A.8: Transaction Confirmed by Miners

## A.9 Already Voted Prompt

The vote cast button will be disabled if the voter has already cast the vote.



| # | Candidate Name | Party | Votes |
|---|---|---|---|
| 1 | Imran Khan | Pakistan Tehreek-e-Insaaf | 1 |
| 2 | Shehbaz Sharif | Pakistan Muslim League | 0 |
| 3 | Bilawal Bhutto | Pakistan Peoples Party | 0 |
| 4 | Khalid Maqbool | Muttahida Qaumi Movement Pakistan | 0 |
| 5 | Asfandyar Wali | Awami National Party | 0 |
| 6 | Jam Kamal Khan | Balochistan Awami Party | 0 |

Your Voter's Address: **0xeddff25879b6392140aabc9b5505f172654f27ad**

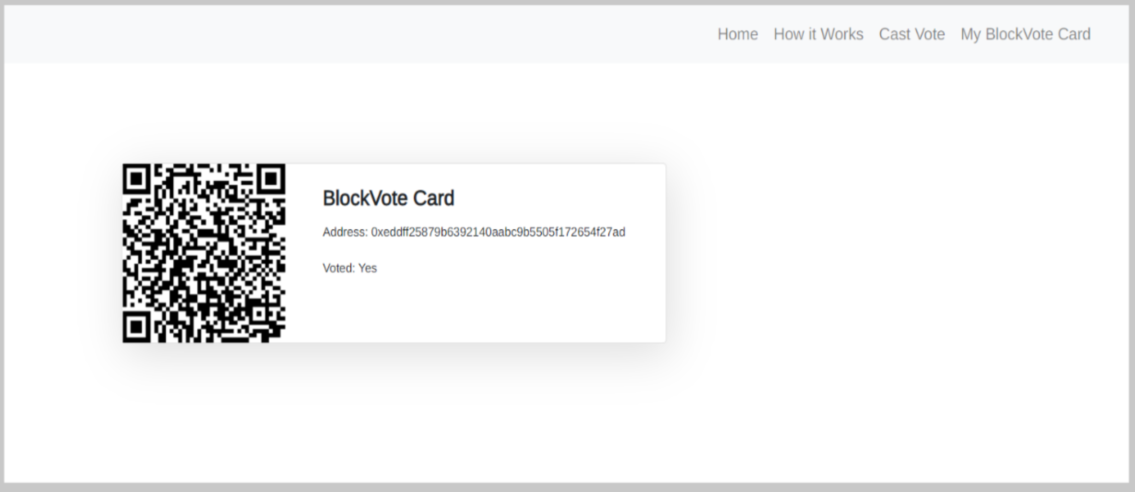
(Refresh the page if you've migrated to new account)

You have already voted!

Figure A.9: Already Voted Prompt

## A.10 Digital Vote Card

The voter can check the status of their vote either their vote was cast or not.



BlockVote Card

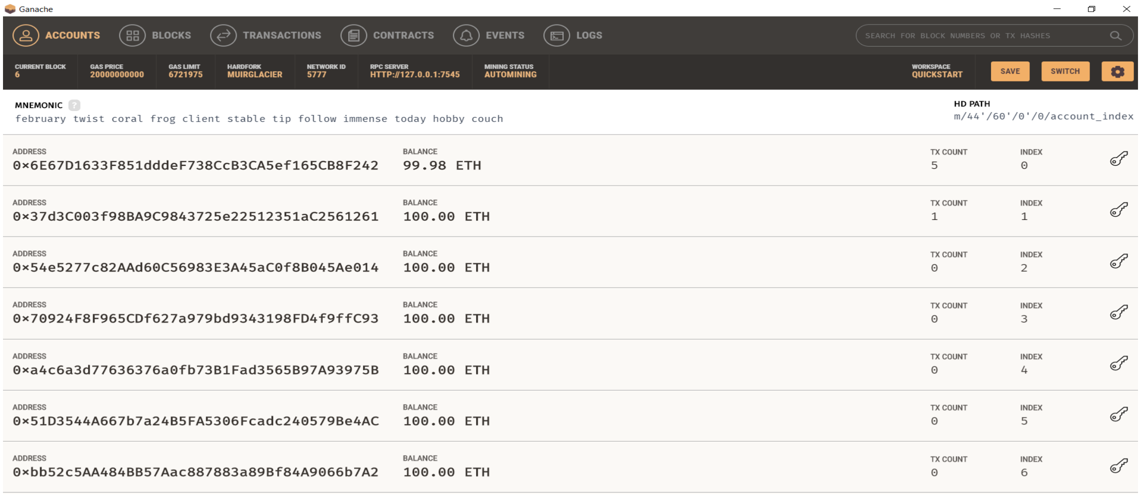
Address: 0xeddff25879b6392140aabc9b5505f172654f27ad

Voted: Yes

Figure A.10: Digital Vote Card Page

## A.11 Smart Contract Owner Account

Below is the blockchain back-end server which consists of all the accounts that could be used to vote. Any account that is not present in this list will not be allowed to the vote. In other words, it serves the purpose of containing the list of voters.



| CURRENT BLOCK | GAS PRICE | GAS LIMIT | HARDFORK | NETWORK ID | RPC SERVER | MINING STATUS | WORKSPACE |
|---|---|---|---|---|---|---|---|
| 6 | 20000000000 | 6721975 | MUIRGLACIER | 5777 | HTTP://127.0.0.1:7545 | AUTOMINING | QUICKSTART |

MNEMONIC
february twist coral frog client stable tip follow immense today hobby couch

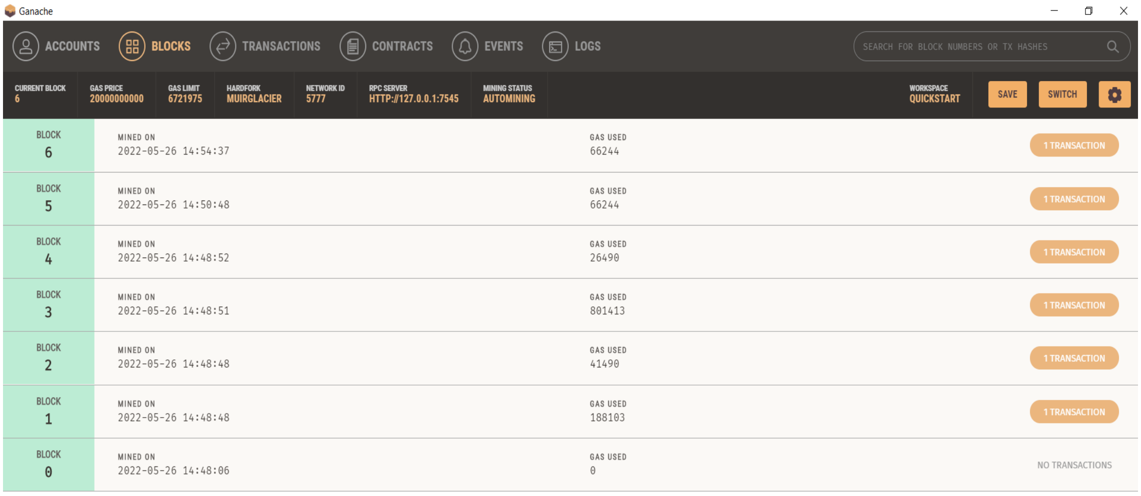
HD PATH
m/44'/60'/0'/0/account_index

| ADDRESS | BALANCE | TX COUNT | INDEX |
|---|---|---|---|
| 0×6E67D1633F851dddeF738CcB3CA5ef165CB8F242 | 99.98 ETH | 5 | 0 |
| 0×37d3C003f98BA9C9843725e22512351aC2561261 | 100.00 ETH | 1 | 1 |
| 0×54e5277c82AAd60C56983E3A45aC0f8B045Ae014 | 100.00 ETH | 0 | 2 |
| 0×70924F8F965CDf627a979bd9343198FD4f9ffC93 | 100.00 ETH | 0 | 3 |
| 0×a4c6a3d77636376a0fb73B1Fad3565B97A93975B | 100.00 ETH | 0 | 4 |
| 0×51D3544A667b7a24B5FA5306Fcadc240579Be4AC | 100.00 ETH | 0 | 5 |
| 0×bb52c5AA484BB57Aac887883a89Bf84A9066b7A2 | 100.00 ETH | 0 | 6 |

Figure A.11: Smart Contract Owner Account

## A.12 Blocks Mined After Transactions

Admin can view the the blocks as they are mined. Blocks are mined at each transactions.



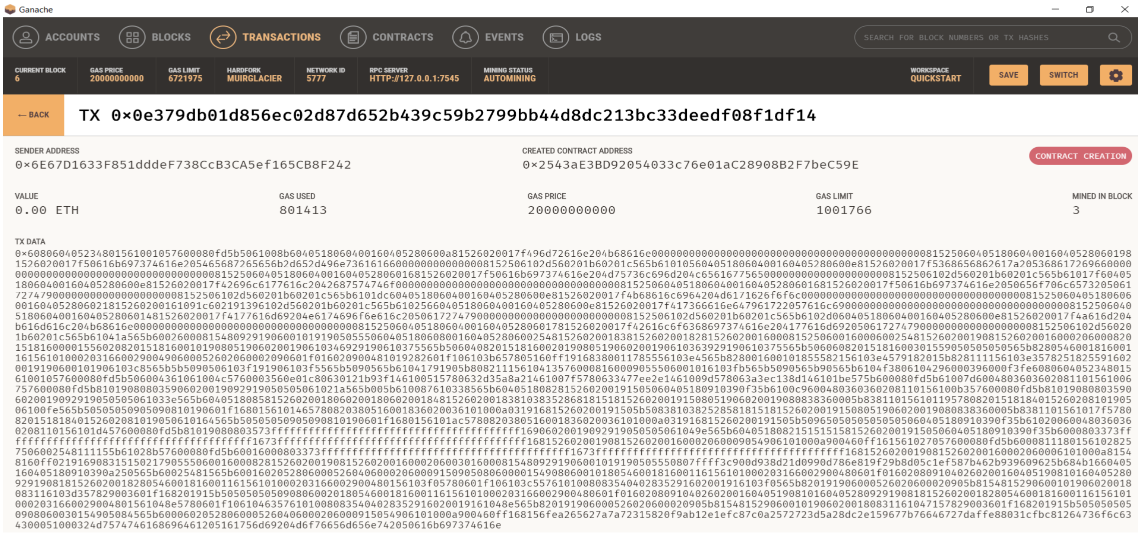
| CURRENT BLOCK | GAS PRICE | GAS LIMIT | HARDFORK | NETWORK ID | RPC SERVER | MINING STATUS | WORKSPACE |
|---|---|---|---|---|---|---|---|
| 6 | 20000000000 | 6721975 | MUIRGLACIER | 5777 | HTTP://127.0.0.1:7545 | AUTOMINING | QUICKSTART |

| BLOCK | MINED ON | GAS USED | |
|---|---|---|---|
| 6 | 2022-05-26 14:54:37 | 66244 | 1 TRANSACTION |
| 5 | 2022-05-26 14:50:48 | 66244 | 1 TRANSACTION |
| 4 | 2022-05-26 14:48:52 | 26490 | 1 TRANSACTION |
| 3 | 2022-05-26 14:48:51 | 801413 | 1 TRANSACTION |
| 2 | 2022-05-26 14:48:48 | 41490 | 1 TRANSACTION |
| 1 | 2022-05-26 14:48:48 | 188103 | 1 TRANSACTION |
| 0 | 2022-05-26 14:48:06 | 0 | NO TRANSACTIONS |

Figure A.12: Blocks Mined After Transactions

## A.13 Contact Creation Transaction

Admin can view the initial contract and its transaction on the blockchain through the following page.



CURRENT BLOCK 6
GAS PRICE 20000000000
GAS LIMIT 6721975
HARDFORK MUIRGLACIER
NETWORK ID 5777
RPC SERVER HTTP://127.0.0.1:7545
MINING STATUS AUTOMINING
WORKSPACE QUICKSTART

TX 0×0e379db01d856ec02d87d652b439c59b2799bb44d8dc213bc33deedf08f1df14

SENDER ADDRESS
0×6E67D1633F851dddeF738CcB3CA5ef165CB8F242

CREATED CONTRACT ADDRESS
0×2543aE3BD92054033c76e01aC28908B2F7beC59E

VALUE
0.00 ETH

GAS USED
801413

GAS PRICE
20000000000

GAS LIMIT
1001766

MINED IN BLOCK
3

TX DATA

```
0×608060405234801561001057600080fd5b5061008b6040518060400160405280600a81526020017f496d72616e204b68616e0000000000000000000000000000000000000000000081525060405180604001604052806019 8
1526020017f50616b697374616e2054656872656b2d652d496e73616600000000000000008152506102d560201b60201c565b6101056040518060400160405280600e81526020017f5368656862617a205368617269660000
000000000000000000000000000000008152506040518060400160405280601681526020017f50616b697374616e204d75736c696d204c65616775650000000000000000000008152506102d560201b60201c565b61017f60405
18060400160405280600e81526020017f42696c6177616c2042687574746f0000000000000000000000000000000000008152506040518060400160405280601681526020017f50616b697374616e2050656f706c6573205061
727479000000000000000000008152506102d560201b60201c565b6101dc6040518060400160405280600e81526020017f4b68616c6964204d6171626f6f6c00000000000000000000000000000000000081525060405180606
001604052806021815260200161091c602191396102d560201b60201c565b6102566040518060400160405280600e81526020017f417366616e64796172205761 6c690000000000000000000000000000000000008152506040
518060400160405280601481526020017f4177616d69204e6174696f6e616c2050617274790000000000000000000000008152506102d560201b60201c565b6102d06040518060400160405280600e81526020017f4a616d204
b616d616c204b68616e0000000000000000000000000000000000008152506040518060400160405280601781526020017f42616c6f6368697374616e204177616d692050617274790000000000000000008152506102d56020
1b60201c565b61041a565b6002600081548092919060010191905055506040518060800160405280600254815260200183815260200182815260200160008152506001600060025481526020019081526020016000206000820
15181600001556020820151816001019080519060200190610346929190610375565b5060408201518160020190805190602001906103639291906103 75565b5060608201518160030155 9050505050565b8280546001816001
161561010002031660029004906000526020600020906 01f016020900481019282601f106103b657805160ff19168380011785556103e4565b828001600101855582156103e4579182015b828111156103e3578251825591602
0019190600101906103c8565b5b5090506103f191906103f5565b5090565b61041791905b808211156104135760008160009055506001016103fb565b5090565b90565b6104f380610429600039600 0f3fe6080604052348015
61001057600080fd5b506004361061004c5760003560e01c80630121b93f146100515780632d35a8a21461007f5780633477ee2e1461009d578063a3ec138d146101be575b600080fd5b61007d6004803603602081101561006
757600080fd5b81019080803590602001909291905050506102 1a565b005b610087610338565b6040518082815260200191505060405180910390f35b6100c96004803603602081101561 00b357600080fd5b81019080803590
6020019092919050505061033e565b604051808581526020018060200180602001848152602001838103835286818151815260200191508051906020019080838360005b8381101561011957808201518184015260208101905
06100fe565b5050505090509081019060 1f16801561014657808203805160018360200361010 00a031916815260200191505b50838103825285818151815260200191508051906020019080838360005b838110156101 7f5780
8201518184015260208101905061016 4565b50505050905090810190601f1680156101ac5780820380516001836020036101000a031916815260200191505b5096505050505050506040518091039 0f35b61020060048036036
0208110156101d457600080fd5b81019080803573ffffffffffffffffffffffffffffffffffffffff16906020019092919050505061049e565b6040518082151515158152602001915050604051809103 90f35b6000803373ff
ffffffffffffffffffffffffffffffffffffff1673ffffffffffffffffffffffffffffffffffffffff16815260200190815260200160002060009054906101000a900460ff16156102705760 0080fd5b6000811180156102825
7506002548111155b61028b57600080fd5b600160008033 73ffffffffffffffffffffffffffffffffffffffff1673ffffffffffffffffffffffffffffffffffffffff16815260200190815260200160002060006101000a8154
8160ff021916908315150217905550600160008281526020019081526020016000206003016000815480929190600101919050555080 7ffff3c900d938d21d0990d786e819f29b8d05c1ef587b462b939609625b684b1660405
160405180910390a250565b600254815 65b600160205280600052604060002060009150905080600001549080600101805460018160011615610100020316600290048 0601f0160208091040260200160405190810160405280
929190818152602001828054600181600116156101000203166002900480156103f05780601f106103c557610100808354040283529160200191 6103f0565b8201919060005260206000209 05b8154815290600101906020018
083116103d357829003601f168201915b50505050509080600201805460018160011615610100020316600290048060 1f016020809104026020016040519081016040528092919081815260200182805460018160011615 6101
000203166002900480156104 8e5780601f1061046357610100808354040283529160200191 61048e565b820191906000526020600020905b81548152906001019060200180831161047157829003601f168201915b505050505
0908060030154905084565b6000602052806000526040600020600091505490610100 0a900460ff168156fea265627a7a72315820f9ab12e1efc87c0a2572723d5a28dc2e159677b76646727daffe88031cfbc81264736f6c63
4300051000324d75747461686964612051617 56d69204d6f76656d656e742050616b697374616e
```

Figure A.13: Contract Creation Transaction

# Bibliography

[1] Avishai Wool Yossef Oren. Rfid-based electronic voting: What could possibly go wrong? Tel-Aviv University, Ramat Aviv 69978, Israel. https://www.eng.tau.ac.il/~yash/evoting-relay-rfid2010.pdf.

[2] Jehangir Arshad. Design and implementation of a software-based advance electronic voting machine using automatic registration and fingerprint identification. Research Square. https://assets.researchsquare.com/files/rs-762430/v1/08977eab-0cd8-47fb-b66d-7d19742cd2ca.pdf?c=1631887724.

[3] Cryptography definition. Kaspersky. https://www.kaspersky.com/resource-center/definitions/what-is-cryptography.

[4] Peer-to-peer. From Wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/Peer-to-peer.

[5] Hong-Ning Dai Weili Chen Xiangping Chen Jian Weng MuhammadImran Zibin Zheng, Shaoan Xie. An overview on smart contracts: Challenges, advances and platforms. Elsevier. https://www.sciencedirect.com/science/article/abs/pii/S0167739X19316280?via%3Dihub.

[6] Zarina Shukur Uzma Jafar, Mohd Juzaiddin Ab Aziz. Blockchain for electronic voting system—review and open research challenges. Sensors. https://www.mdpi.com/1424-8220/21/17/5874.

[7] Pisal Setthawong Chinnapong Angsuchotmetee. Blockvote : An architecture of a blockchain-based electronic voting system. Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI) Association. https://www.mdpi.com/1424-8220/21/17/5874.

[8] Rahul Puniani. Conceptualization of a blockchain based voting ecosystem in estonia. UNIVERSITY OF TARTU, 2017-2019. https://core.ac.uk/download/pdf/237084264.pdf.

[9] Janno Siim Jan Willemson Sven Heiberg, Ivo Kubjas. On trade-offs of applying block chains for electronic voting bulletin boards. UNIVERSITY OF TARTU. https://media.voog.com/0000/0042/1115/files/evotingUT.pdf.

[10] Raja Naeem Akram Konstantinos Markantonakis Freya Sheer Hardwick, Apostolos Gioulis. E-voting with blockchain: An e-voting protocol with decentralisation and voter privacy. page 7. 2015. https://arxiv.org/pdf/1805.10258.pdf.

[11] Yi Liu and Qi Wang. An e-voting protocol based on blockchain. Southern University of Science and Technology, Shenzhen, China. https://eprint.iacr.org/2017/1043.pdf.

[12] Pavel Tarasov and Hitesh Tewari. The future of e-voting. School of Computer Science and Statistics, Trinity College Dublin, University of Dublin, Ireland. http://www.iadisportal.org/ijcsis/papers/2017210210.pdf.

[13] Taylor Hornby Nathan Wilcox Daira Hopwood, Sean Bowe†. Zcash protocol specication. Jubjub bird image credit: Peter Newell 1902; Daira Hopwood 2018. https://zips.z.cash/protocol/protocol.pdf.

[14] Philip Boucher. What if blockchain technology revolutionised voting? European Parliamentary Research Service. https://www.europarl.europa.eu/RegData/etudes/ATAG/2016/581918/EPRS_ATA(2016)581918_EN.pdf.

[15] https://followmyvote.com/survey-research/electoral-integrity-worldwide/.

[16] Daniel Weitzner Michael A. Specter, James Koppel. The ballot is busted before the blockchain: A security analysis of voatz, the first internet voting application used in u.s. federal elections. USENIX. https://www.usenix.org/system/files/sec20-specter.pdf.

[17] Polys online voting system. Kaspersky. https://cdn.polys.me/Whitepaper/7262_WP_Polys_En_WEB_7.pdf.

[18] Chrisanthi Avgerou. Explaining trust in it-mediated elections: A case study of e-voting in brazil. London School of Economics. https://aisel.aisnet.org/cgi/viewcontent.cgi?article=1653&context=jais.

[19] Julienne Molineaux. A working paper on online voting (internet voting, evoting) in new zealand. Auckland University of Technology. https://thepolicyobservatory.aut.ac.nz/__data/assets/pdf_file/0006/233088/A-working-paper-on-internet-voting-in-New-Zealand-December-2018.pdf.

[20] Jordi Barrat Ardita Driza Maurer. E-voting case law. Routledge Taylor and Francis Group. https://books.google.com.pk/books?hl=en&lr=&id=MLC1CwAAQBAJ&oi=fnd&pg=PP1&dq=australia+e-voting+2004&ots=suUf1VYYqK&sig=E7TS0pH46zY8gpPK6GAes6oFZS4&redir_esc=y#v=onepage&q=australia%20e-voting%202004&f=false.