

A Major Project Final Report on
**Digital Voting System based on
Blockchain Technology**

Submitted in Partial Fulfillment of the Requirements for
The Degree of **Bachelor of Engineering in Software Engineering**
Under Pokhara University

Submitted by:
Binod Adhikari, 171746
Dipesh Regmi, 171716

Under the supervision of
Mr. Bishal Trital

Date:
13 September 2022



Department of Software Engineering
NEPAL COLLEGE OF
INFORMATION TECHNOLOGY

Balkumari, Lalitpur, Nepal

Acknowledgement

We take this opportunity to express our deepest and sincere gratitude to our supervisor **Mr. Bishal Trital** for his insightful advice, motivating suggestions, invaluable guidance, help and support in successful completion of this project and also for his constant encouragement and advice throughout the project.

We express our deep gratitude to our head of department **Mr. Birendra Bista** for his regular support, cooperation and coordination.

We are also grateful to our professor **Dr. Roshan Chitrakar** for providing us assistance in various stages during the course of our project.

Last but not the least, we would like to thank our teachers and colleagues who have been knowingly or unknowingly part of this project with their views during the entire development time.

Abstract

In digital electronic voting, security is always the biggest anxiety. In the existing system, the EVM system used, when compared to the conventional paper ballot system, EVM reduces the time for casting vote and result announcement. But still having many issues there is the risk that the election authorities are able to change or remove the vote, therefore chances for violating its secrecy. The entire system may be rigged by any third party. Blockchain technology provides secure electronic voting platform, which is a decentralized, peer-to-peer transaction ledger, which enables every vote that is casted to be considered as an individual transaction which creates a transparent and secure environment for elections. The users will be able to cast their votes only once and will be able to view the total votes casted in real time without having the permission to edit the same after the election gets over. These votes will be counted and the results will then be announced. This work is achieved by solving the issues of digital voting systems and a mechanism to boost the number of voters and their trust in the electoral process.

Keywords

digital voting system, blockchain, peer-peer transaction, distributed system, hashing

Table of Contents

Acknowledgement	i
Abstract	ii
List of Figures	iv
List of Tables	v
1. Introduction	v
1.1 Problem Statement	1
1.2 Objectives	2
1.3 Significance of the Study	2
1.4 Scope and Limitations	3
2. Literature Review	4
2.1 Related Works	4
2.2 Technologies Used	6
3. Methodology	11
3.1 Software Development Life Cycle	11
3.2 Software Requirement Specifications	14
3.2.1 Generic Voting Principles	14
3.2.2 Voting System Design Criteria	14
4. System Design	16
4.1 System Architecture	16
4.2 Use Case Diagram	18
4.3 Activity Diagram	19
5. Time Scheduling	21
6. Testing	22
6.1 Testing Table	22
6.2 List of Testing	23
7. Results and Discussion	25
8. Conclusion	26
9. Further Works/Recommendations	27
10. References	28

List of Figures

Figure i: Different blocks in Blockchain	6
Figure ii: Sample Image of Elon Musk	8
Figure iii: List of Encodings of Figure ii	8
Figure iv: Incremental Model	11
Figure v: System Architecture	16
Figure vi: Use-Case Diagram of a System	18
Figure vii: Activity Diagram of voter's registration	19
Figure viii: Activity Diagram of voting process	20
Figure ix: Gantt Chart	21
Figure x: Response of TC_1	23
Figure xi: Response of TC_2	23
Figure xii: Response of TC_3	24
Figure xiii: Response of TC_4	24

List of Tables

Table i: Time Scheduling	21
Table ii: Test Cases	22

1. Introduction

Voting is an important part of choosing a rightful candidate through the democratic process and hence is a crucial process for any country. Among different vote collection methods, a paper-based method is widely adopted. In this method, the voter selects the preferred candidate or party by marking the corresponding symbol or letter. In the context of Nepal too, the paper-based method is practiced for the process of election. With the advancement of technology all around the world, isn't it high time that we modernize the voting process? The use of machines at the polling centers or casting vote directly from home via the internet can digitize the voting process. It would be wonderful if the voting is carried out through a platform that is secure enough and provides transparency into the result of the election. The places like Estonia, Norway, and Australia have already conducted an online voting process. There are several challenges that we have to face to create this online platform. Digitizing the voting process may upsurge different security issues. The weakness in the architecture of the system can lead to manipulation of the election result. Vote fraud due to attacks from the intruders may manipulate the result or may result in the loss of the data which is totally unacceptable in a democratic process like the election and the election process cannot be rerun. One of the best potential solutions is to use blockchain technology.

1.1 Problem Statement

Electronic Voting is not a new concept, in fact, it was first-ever introduced in 1986. David Shagum was the first to introduce the first-ever electronic voting system which was based on public-key cryptography [1]. Keeping no connection between voters and ballots, a secure country election was conducted. Electronic voting machines have been viewed as vulnerable, based on physical security concerns. These machines can be sabotaged by anyone who has physical access to it, thus affecting all votes cast on the machine. The major vulnerability is the centralized database where election data are stored. This vulnerability could be a single point of failure. However, integrating blockchain technology on the aforementioned machine can take the entire voting system to a whole new level. The blockchain-based election was held for the first time in Sierra Leone, a country in Western Africa. Blockchain is a distributed, immutable public ledger introduced in 2008 by Satoshi Nakamoto in the form of the creation of the first cryptocurrency, called Bitcoin. The Bitcoin blockchain uses a decentralized public ledger with POW (Proof-of-Work) consensus protocol [2]. The chain of the blocks in the blockchain is cloned, cryptographically signed, and verified publicly at every transaction so

that no one can meddle with the data once written on the blockchain [3]. Due to its immutable property and decentralized architecture, it can carry out elections in a more transparent and secure way. Blockchain Digital Voting System comprising of blockchain where votes cast is recorded, python scripts handling overall election processes, private nodes providing consensus, and the voter portal accessible to the voters can guarantee the security, integrity, and transparency of the casted votes including the anonymity of the voters avoiding the single point of failure as compared to the traditional centralized voting system. The script can be taken as an election, the transaction as a ballot paper, and the blockchain as a ballot box. This blockchain is accessible to the general public to make the election more transparent.

1.2 Objectives

The voting system that is hereby conceived must satisfy the following requirements:

1. To improve the existing voting system using Blockchain technology as a secure transaction database.
2. To digitize the overall voting system.
3. To ensure voting systems should be tamper-proof and decentralized.

1.3 Significance of the Study

Through the comprehensive exploration of digital voting based on blockchain, it introduces a different means of democratic changes to the previous system. The decentralized system will provide a proof of concept where a voter can vote by keeping their privacy i.e. a voted person or a system will have no knowledge of the voter's identity.

Previously digital voting systems were not taken into consideration as they possess a huge risk since they are centralized but introducing a concept of blockchain technology will have high benefits as the chance of tampering with the vote count will be significantly zero.

Being a tamper-proof system even while keeping the privacy of the voters will be a huge turnover in this digital era where almost all of the systems are digital except for some like voting which doesn't earn public trust.

Especially, this system will benefit the following:

Government: This system will allow the government to conduct the election without needing to spend a lot of resources on paper ballots or the security of the system.

Public: The process of digital voting will be easy and secure i.e. the public can vote from digital machines available at the election venue of their respective locations.

Future researchers: This system will open up a new way for future researchers, they can use this technology to further develop it or use it on different platforms.

1.4 Scope and Limitations

The scopes and limitations of our digital voting system based on blockchain are as follows:

Scope:

The scope of this project is to digitize the overall voting system with the help of blockchain technology for more accurate and convenient organization of election systems. Our system provides the following scope:

1. This system can be implemented on various platforms where the election needs to be held for a different purpose whether it be for government election or for choosing a representative.
2. Voters will have the opportunity to vote quickly and conveniently.
3. Votes can be counted in real time.
4. Voting can be completed in a very short period of time with better accuracy.

Limitations:

Although our project provides reliability by providing an alternative to the old existing paper based voting system, we have some limitations, which are to be taken into consideration. Some of the limitations of our projects are:

1. The system will act as a blueprint for the voting and election process. This will not cover all the systems that we can possibly use.
2. An admin with the highest level of privileges can see the details of the voting phase including who voted for whom.
3. Difficulties in implementation in rural areas.

2. Literature Review

2.1 Related Works

Several articles have been published in the recent era that highlighted the security and privacy issues of blockchain-based electronic voting systems. This review reflects the comparison of selected electronic voting schemes based on blockchain.

The open vote network (OVN) was presented by [4], which is the first deployment of a transparent and self-tallying internet voting protocol with total user privacy by using Ethereum. In OVN, the voting size was limited to 50–60 electors by the framework. The OVN is unable to stop fraudulent miners from corrupting the system. A fraudulent voter may also circumvent the voting process by sending an invalid vote. The protocol does nothing to guarantee the resistance to violence, and the electoral administrator wants to trust [5,6].

Furthermore, since solidity does not support elliptic curve cryptography, they used an external library to do the computation [7]. After the library was added, the voting contract became too big to be stored on the blockchain. Since it has occurred throughout the history of the Bitcoin network, OVN is susceptible to a denial-of-service attack [8]. Table 3 shows the main comparison of selected electronic voting schemes based on blockchain.

Lai et al. [9] suggested a decentralized anonymous transparent electronic voting system (DATE) requiring a minimal degree of confidence between participants. They think that for large-scale electronic elections, the current DATE voting method is appropriate. Unfortunately, their proposed system is not strong enough to secure from DoS attacks because there was no third-party authority on the scheme responsible for auditing the vote after the election process. This system is suitable only for small scales because of the limitation of the platform. Although using Ring Signature keeps the privacy of individual voters, it is hard to manage and coordinate several signer entities. They also use PoW consensus, which has significant drawbacks such as energy consumption: the “supercomputers” of miners monitor a million computations a second, which is happening worldwide. Because this arrangement requires high computational power, it is expensive and energy-consuming.

Shahzad et al. [10] proposed the BSJC proof of completeness as a reliable electronic voting method. They used a process model to describe the whole system’s structure. On a smaller scale, it also attempted to address anonymity, privacy, and security problems in the election. However, many additional problems have been highlighted. The proof of labor, for example,

is a mathematically vast and challenging job that requires a tremendous amount of energy to complete. Another problem is the participation of a third party since there is a significant risk of data tampering, leakage, and unfair tabulated results, all of which may impact end-to-end verification. On a large scale, generating and sealing the block may cause the polling process to be delayed [11].

Khan, K.M. [12] has proposed block-based e-voting architecture (BEA) that conducted strict experimentation with permission and permissionless blockchain architectures through different scenarios involving voting population, block size, block generation rate, and block transaction speed. Their experiments also uncovered fascinating findings on how these parameters influence the overall scalability and reliability of the electronic voting model, including interchanges between different parameters and protection and performance measures inside the organization alone. In their scheme, the electoral process requires the generation of voter addresses and candidate addresses. These addresses are then used to cast votes from voters to candidates. The mining group updates the ledger of the main blockchain to keep track of votes cast and the status of the vote. The voting status remains unconfirmed until a miner updates the main ledger. The vote is then cast using the voting machine at the polling station.

However, in this model, there are some flaws found. There is no regulatory authority to restrict invalid voters from casting a vote, and it is not secure from quantum attack. Their model is not accurate and did not care about voters' integrity. Moreover, their scheme uses Distributed consensus in which testimonies (data and facts) can be organized into cartels because fewer people keep the network active, and a "51%" attack becomes easier to organize.

This attack is potentially more concentrated and did not discuss scalability and delays in electronic voting, which are the main concerns about the blockchain voting system. They have used the Multichain framework, a private blockchain derived from Bitcoin, which is unsuitable for the nationwide voting process. As the authors mentioned, their system is efficient for small and medium-sized voting environments only.

2.2 Technologies Used

The tools and techniques that we have deployed in our system are explained below.

1. Blockchain

Blockchain is a system of recording information in a way that makes it difficult or impossible to change, hack, or cheat the system. A blockchain is essentially a digital ledger of transactions that is duplicated and distributed across the entire network of computer systems on the blockchain. Each block in the chain contains a number of transactions, and every time a new transaction occurs on the blockchain, a record of that transaction is added to every participant's ledger. The decentralized database managed by multiple participants is known as Distributed Ledger Technology (DLT).

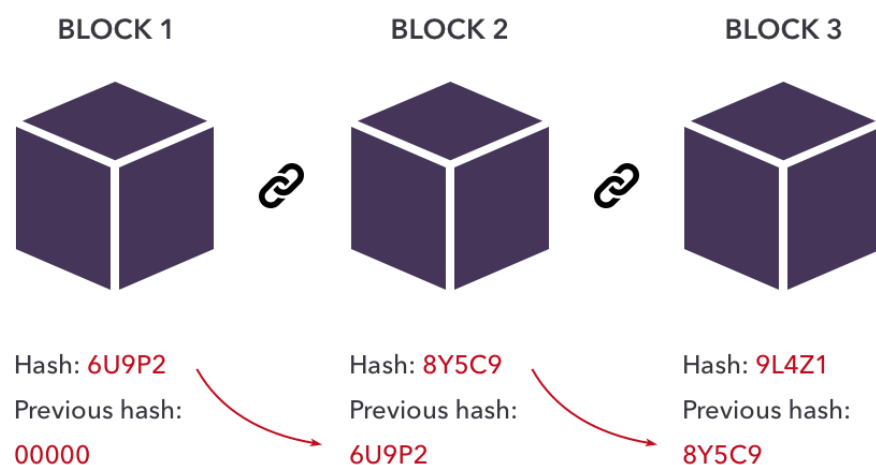


Figure i: Different blocks in Blockchain

Blockchain is a type of DLT in which transactions are recorded with an immutable cryptographic signature called a hash. This means if one block in one chain was changed, it would be immediately apparent it had been tampered with. If hackers wanted to corrupt a blockchain system, they would have to change every block in the chain, across all of the distributed versions of the chain.

2. Proof of Work

Proof of work (PoW) is a form of adding new blocks of transactions to a cryptocurrency's blockchain. The work, in this case, is generating a hash (a long string of characters) that matches the target hash for the current block. Proof of work is a technique used by cryptocurrencies to verify the accuracy of new transactions that are added to a blockchain. The

decentralized networks used by cryptocurrencies and other defi applications lack any central governing authority, so they employ proof of work to ensure the integrity of new data.

3. Face Recognition

For our facial verification process, we have used the `face_recognition` python library. Face recognition algorithms can extract features from a face image, namely positions of forehead, eyes, nose, mouth, chin, jaws.

Face Landmarks: There are 68 specific points (called landmarks) that exist on every face.

Face Encodings: This is the 128 encoding feature vector from a pretrained network over millions of images.

Following are the basic steps which are used while creating this library.

1. Encode a picture using the HOG algorithm to create a simplified version of the image. Using this simplified image, find the part of the image that most looks like a generic HOG encoding of a face.
2. Figure out the pose of the face by finding the main landmarks in the face. Once we find those landmarks, use them to warp the image so that the eyes and mouth are centered.
3. Pass the centered face image through a neural network that knows how to measure features of the face. Save those 128 measurements.
4. Looking at all the faces we've measured in the past, see which person has the closest measurements to our face's measurements. That's our match!

The main reason behind using this face recognition feature is to provide additional security to our system so that only eligible and correct voters can participate in the voting process. Here, we simply input the image from our computer or get a direct image from our webcam. Then, the function we have created in our system will give us the output of the list of 128 encoding features which we mentioned earlier.



Figure ii: Sample picture of Elon Musk

```
[ -3.22140977e-02  1.32757962e-01  1.03660554e-01 -1.54176261e-03
-1.39290333e-01  4.59651649e-02 -6.04024976e-02 -1.88459799e-01
 1.57444119e-01 -4.63126823e-02  2.79171348e-01 -9.72657353e-02
-2.23405048e-01 -2.79332399e-02 -7.91804120e-03  1.09720878e-01
-1.88225403e-01 -1.01629436e-01  2.52081733e-03 -9.77773517e-02
 1.17015883e-01  6.97332993e-02  2.44141910e-02 -3.97718474e-02
-1.52565643e-01 -3.36885393e-01 -9.99340191e-02 -9.76220816e-02
 1.34754360e-01 -7.23698810e-02  1.56760253e-02  8.13452993e-03
-1.72818035e-01 -9.05525759e-02  4.11255248e-02  1.61343589e-02
-7.83793181e-02 -8.10466185e-02  1.79392233e-01 -5.83891533e-02
-1.90243706e-01 -1.92637518e-02  6.23493567e-02  1.71955809e-01
 1.12117536e-01  6.75343871e-02  2.97434032e-02 -7.59051293e-02
 4.62464914e-02 -2.05647394e-01  7.79409632e-02  1.61274910e-01
 4.53124307e-02  5.15399352e-02  1.08617954e-01 -1.37279317e-01
-5.84023967e-02  1.94541261e-01 -1.43956348e-01  9.32636112e-02
 1.27859026e-01 -8.58973265e-02 -5.54560823e-03 -4.30783778e-02
 1.62602231e-01  6.66275620e-02 -3.03007625e-02 -1.64515674e-01
 1.53465450e-01 -5.18120006e-02 -1.26958504e-01  1.17648147e-01
-9.81646404e-02 -1.55776948e-01 -2.56620944e-01  6.43276796e-03
 4.81631786e-01  8.70086811e-03 -1.54412910e-01  1.35882143e-02
-1.79607943e-02 -5.72739774e-03  1.32233039e-01  2.50677876e-02
 3.43443900e-02 -6.87113628e-02 -1.01266056e-01  6.55891839e-03
 1.83830023e-01 -7.84002990e-02 -1.90505795e-02  2.14571089e-01
-5.15913367e-02  3.06133069e-02  3.60265523e-02  1.12820342e-01
-8.16151351e-02  4.58311066e-02 -7.87415951e-02 -6.85654581e-02
 1.12734072e-01 -1.40151875e-02 -9.59167443e-03  1.30267650e-01
-1.90005273e-01  1.45173281e-01  3.23660416e-03 -1.25595555e-03
 3.55093144e-02 -6.63941875e-02 -2.56190076e-02  3.24442014e-02
 2.02211469e-01 -3.23678315e-01  1.97198913e-01  1.58962533e-01
-5.61595410e-02  1.41424313e-01 -3.00422776e-04  3.29012424e-02
 2.84861196e-02 -8.13870132e-02 -2.41067737e-01 -1.19648576e-01
 6.41226163e-03 -4.47309539e-02 -5.49955368e-02  2.69954074e-02 ]
```

Figure iii: List of Encodings of Figure ii

We match these encodings to the already available encodings to us using the compare faces function. If the result obtained after comparing the encodings is more than the tolerance level of 0.6 units, we consider the input image is permitted to vote in the election.

4. FastAPI

FastAPI is a Web framework for developing RESTful APIs in Python. FastAPI is based on Pydantic and type hints to validate, serialize, and deserialize data, and automatically auto-generate OpenAPI documents.

It fully supports asynchronous programming and can run with Uvicorn and Gunicorn. To improve developer-friendliness, editor support was considered since the earliest days of the project.

5. Hashing

Hashing is the process of converting a given key into another value. A hash function is used to generate the new value according to a mathematical algorithm. The result of a hash function is known as a hash value or simply, a hash.

We have used the Python SHA512 hashing algorithm in our project for hashing the block's address such that a small change in one block will reflect the change in other blocks. SHA stands for Secure Hash Algorithms. These are a set of cryptographic hash functions. These functions can be used for various applications like passwords, etc. The hashlib module of Python is used to implement a common interface to many different secure hash and message digest algorithms.

6. SQLite

SQLite is a C-language library that implements a small, fast, self-contained, high-reliability, full-featured, SQL database engine. SQLite is the most used database engine in the world.

Although we have used Blockchain for storing the transactions, we are using SQLite to manage the user's personal data and information.

7. OpenCV

OpenCV is the huge open-source library for computer vision, machine learning, and image processing and now it plays a major role in real-time operation which is very important in today's systems. By using it, one can process images and videos to identify objects, faces, or even handwriting of a human. We have used this library for a face recognition system.

8. Swagger UI

Swagger UI is a collection of HTML, JavaScript, and CSS assets that dynamically generate beautiful documentation from a Swagger-compliant API. Swagger UI allows the development team to visualize and interact with the API's resources without having any of the implementation logic in place.

9. QR code

A Quick Response code is a two-dimensional pictographic code used for its fast readability and comparatively large storage capacity. The code consists of black modules arranged in a square pattern on a white background. The information encoded can be made up of any kind of data (e.g., binary, alphanumeric, or Kanji symbols).

3. Methodology

3.1 Software Development Life Cycle

The life cycle model we have used for this project is the Incremental Model. Incremental Model is one of the most adopted models of software development process where the software requirement is broken down into many standalone modules in the software development life cycle. Once the modules are split then incremental development will be carried out in steps covering all the analysis, designing, implementation, carrying out all the required testing or verification and maintenance. In incremental models, each iteration stage is developed and hence each stage will be going through requirements, design, coding and finally the testing modules of the software development life cycle. Functionality developed in each stage will be added on the previously developed functionality and this repeats until the software is fully developed. At each incremental stage there will be a thorough review basing on which the decision on the next stage will be taken out.

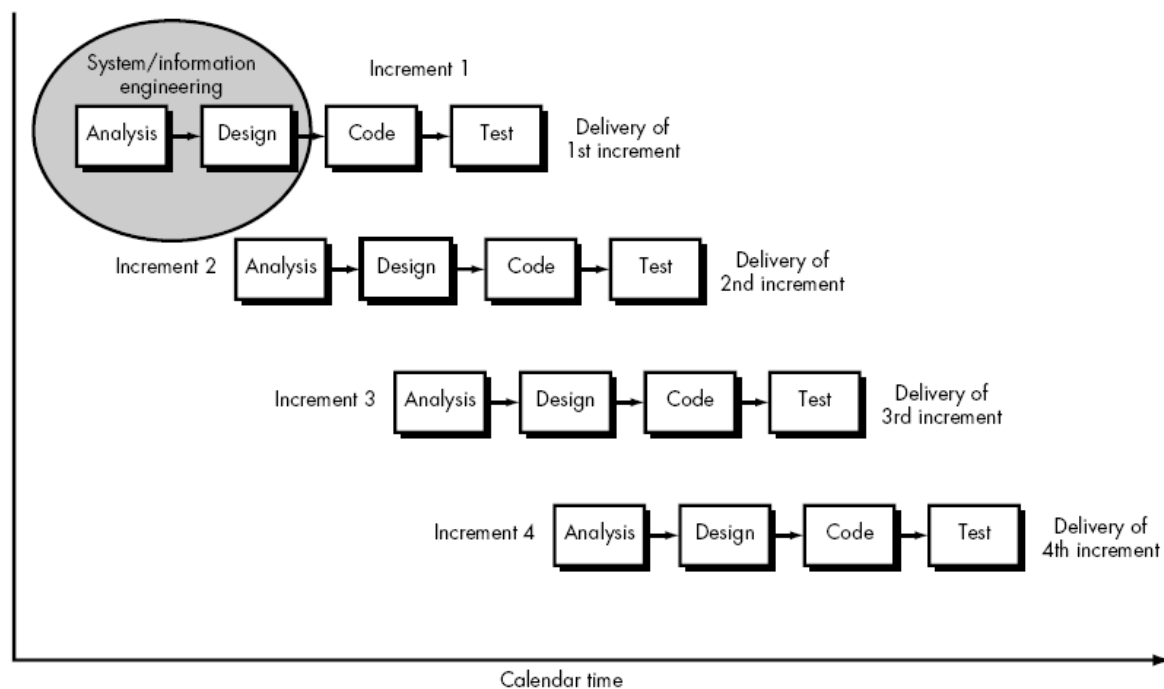


Figure iv: Incremental Model

We can see that at each stage of incremental development we are going through analysis, design, code and test phase and by doing this we are making sure that the various incremental stages are compatible and cumulatively helping in achieving the required objectives from the software.

The phases of the incremental model are described briefly as below.

Requirement Analysis: Complete analysis is performed on the requirement and how to make sure that this requirement will be compatible to previously developed.

Design: Once the requirement for this particular increment is understood and clear, then design will be drafted on how to implement and archive this requirement.

Code: Now the coding is performed in accordance to achieve the purpose of the requirements. All the coding standards will be followed without any defaults and unnecessary hard codes.

Test: This is the last in the incremental phase where aggressive testing is performed on the developed code and defects are reported and resolved.

We have completed this project in four increments which are shown below.

Increment 1: API development for registration of both voters and candidates.

Increment 2: Proper implementation of Blockchain logic in our system.

Increment 3: Vote transaction APIs creation with the help of Blockchain principles.

Increment 4: Development of vote counting API and webcam based face detection feature.

Our project which implements the incremental model comprises four increments which are discussed below.

Increment 1: API development for registration of both voters and candidates.

We need to first register the voters and candidates before we start the election process. So for that we have created separate APIs for both voters and candidates using the python based FastAPI web framework. Here, for voters, we give the correct registration details including facial scan as an input. And as an output, this API will generate a secret 64 bit key for each voter which is then encoded into a QR code for convenience. Similarly for candidates, the API takes similar information in addition to the position they are nominated for. The candidates are then provided with a unique identification number.

Increment 2: Proper implementation of Blockchain logic in our system.

For the blockchain principles, we have created a class called Blockchain, which includes various functions defined for different tasks like block creation, initialization, getting previous block, implementation of proof of work, hashing, validation of block, block mining and so on.

Increment 3: Vote transaction APIs creation with the help of Blockchain principles.

The key and special feature of our project is implementing the Blockchain principles during the transaction of each and every vote. Here, we utilize the already built principle in increment 2, to cast the vote in a reliable, secure, fast and efficient manner. For this, we have created an API which takes input as receiver id indicating the candidates, the voters face and the QR code given to the voter while registration. After receiving these inputs, our API checks for eligibility of voters with the help of the secret key encoded in QR code and the facial recognition algorithm used in the API. After passing the eligibility criteria, the voter will then send the unit amount of vote to the respective candidates. This transaction is then stored into the unconfirmed transaction list until the block is mined successfully. Then all the transactions are removed from the unconfirmed list.

Similarly, making the transaction decentralized is also one of the main objectives of our project. So for making our system decentralized, we have to run the system in multiple nodes with almost the same specifications in each of them. The transactions are carried out in multiple connected nodes. Each transaction will form a chain of blocks in multiple nodes. So tampering in any of the nodes will reflect in all other nodes, which makes our voting system tamper proof and secure.

After successfully completing a transaction in various nodes, we have to call a function which replaces the current chain with the longest one so that all the transaction upto the last one are included in the confirmed status.

Increment 4: Development of vote counting API and webcam based face detection feature.

After completing the transaction, the number of transactions is to be counted for each of the candidates. To achieve this functionality, we get the longest chain from the nodes and append the number of transactions to the respective candidates. This will then give us the output of the amount of votes a candidate received in the election.

Another task done in this increment was to develop a mechanism to capture the real time facial structure of the voters and candidates during the registration process and voting process. This function takes the face as input from a webcam and passes it to implement the face recognition algorithm, which will be further described in another section.

3.2 Software Requirement Specifications

3.2.1 Generic Voting Principles

1. Only eligible persons can vote.
2. No person gets to vote more than once.
3. The vote is secret.
4. Each (correctly cast) vote gets counted.
5. The voters trust that their vote is counted.

3.2.2 Voting System Design Criteria

Democratic: A system is considered to be “democratic” if only eligible voters are allowed to vote (eligibility) and if each eligible voter can only cast a single vote (reusability). An additional characteristic is that no one should be allowed to duplicate anyone else’s vote.

Accuracy: Correctness of the system. Election systems should record the votes correctly. The announced result should match the actual outcome of the election.

Reliability: No reasonably sized coalition of voters or authorities (either benign or malicious) may disrupt the election. This includes allowing abstention of registered voters, without causing problems or allowing other entities to cast legitimate votes on their behalf, as well as preventing misbehavior of voters and authorities from invalidating the election outcome by claiming that some other actor of the system failed to properly execute its part.

Robustness implies that security should also be provided against external threats and attacks, e.g. denial of service attacks.

Integrity: Votes should not be able to be modified without detection.

Verifiability: Should be possible to verify that votes are correctly counted for in the final tally. Results can be found to agree on the election result by comparing election data with other holders of election data or by checking whether an individual vote has been properly cast.

Auditability: There should be reliable and demonstrably authentic election records.

Secrecy: No one should be able to determine how or whom any individual voted.

Non-coercibility: Voters should not be able to prove how they voted. An incoercible scheme does not allow the voters to convince any other participant (e.g. a coercer) on what they have voted for.

Fairness: Should ensure that no one can learn the outcome of the election before the announcement of the tally. Therefore, acts like influencing the decision of late voters by announcing an estimate, or providing a significant but unequal advantage (being the first to know) to specific people or groups, are prevented.

Flexibility: Equipment should allow for a variety of ballot question formats.

Convenience: Voters should be able to cast votes with minimal equipment and skills.

Certiability: Systems should be testable against essential criteria.

Transparency: Voters should be able to possess a general understanding of the whole process.

Cost-effectiveness: Systems should be affordable and efficient.

4. System Design

This field contains the detailed design and architecture of the system and the associated UML diagrams.

4.1 System Architecture

This system proposes a design of new prototype architecture on digital voting based on Blockchain technology.

The following figure illustrates the system architecture.

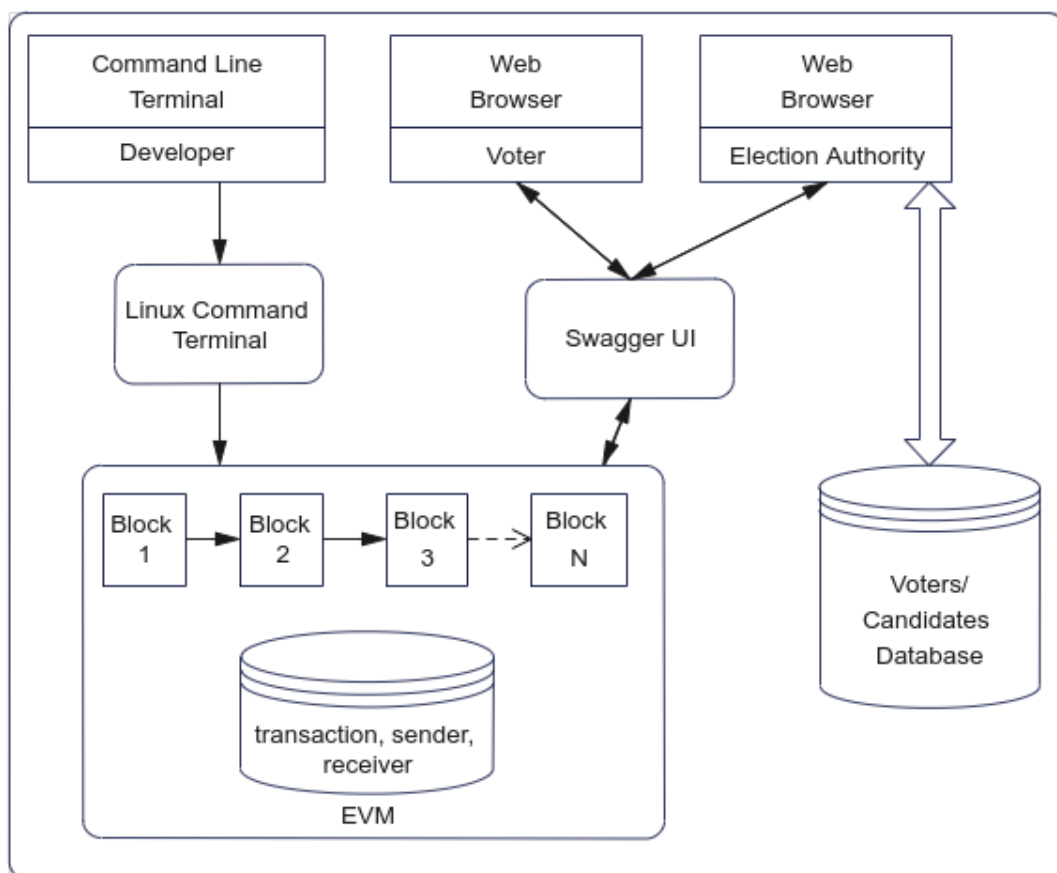


Figure v: System Architecture

The system architecture is the conceptual design of following important components:

1. Blockchain

The various blocks created during the voting transactions are the key components of our system which make our voting process secure and tamper-proof. Number of blocks depends upon the number of transactions done in different nodes. These blocks are implemented in EVM which stores important details of sender, receiver and transaction.

2. Actors

The main actors of our system are Developer, Voter and Election Authority. Developer uses mostly the command line terminal for controlling the overall system. While Voters and Election Authorities interact with our Swagger UI from the supporting web browsers.

3. Terminal and Swagger UI

The developer mostly makes use of linux terminal for operating overall workings of the system. But, the voters and election authorities, who are usually non technical actors, interact with the BLockchain via a UI implemented using Swagger UI.

4. Database

Although the transactions are stored in Blockchain, the personal details of voters and candidates are stored in normal databases. This avoids the huge load on Blockchain to make it a little faster.

4.2 Use Case Diagram

Use case diagram shows the relation between actors, scenarios and system boundary to reach the user goal. There are basically four actors in our system: Voter, Candidate, Election Authority and Admin. The use case diagram for our project is as below.

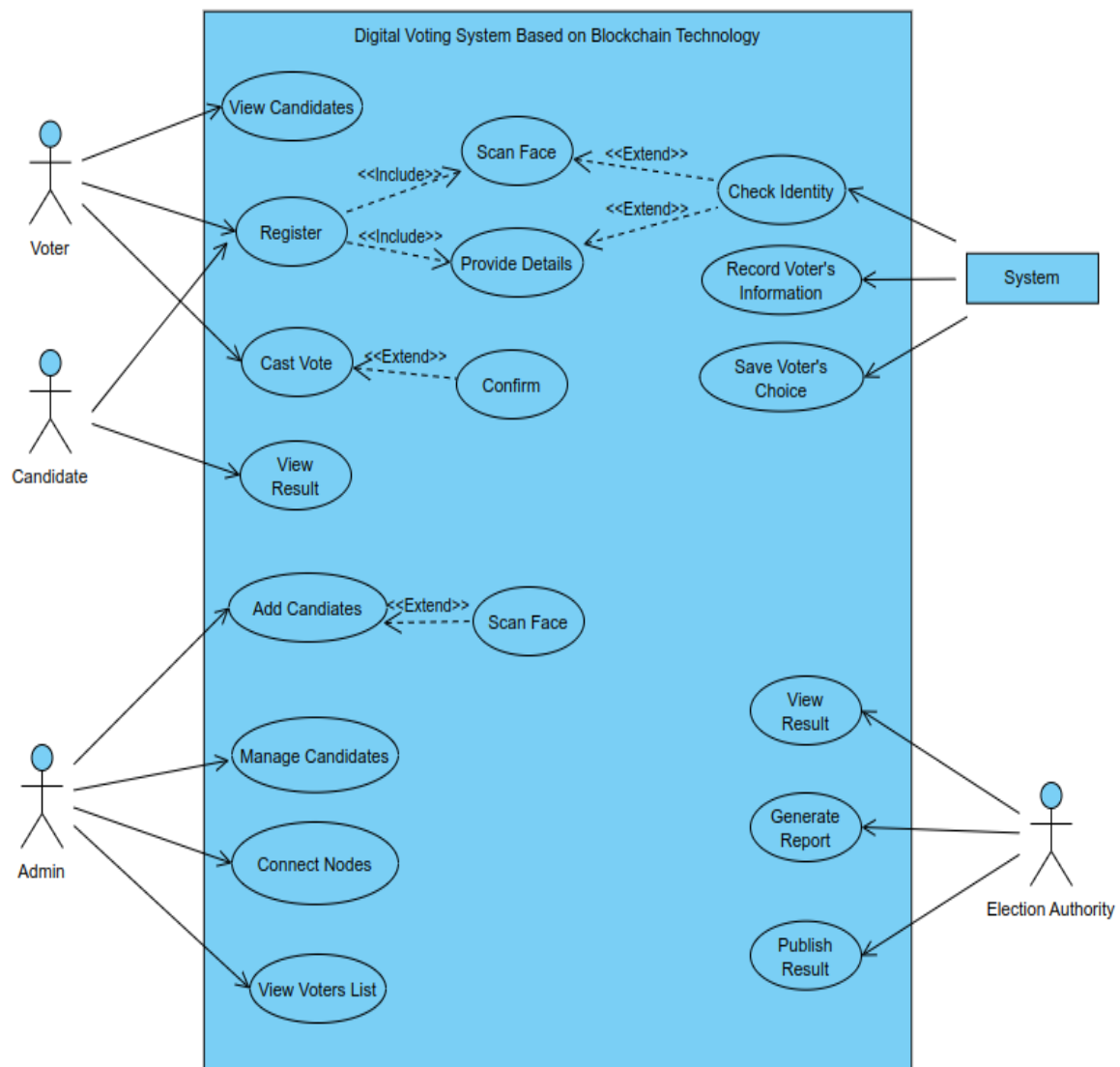


Figure vi: Use-Case Diagram of a System

4.3 Activity Diagram

Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent.

The activity diagrams for our project are as follows.

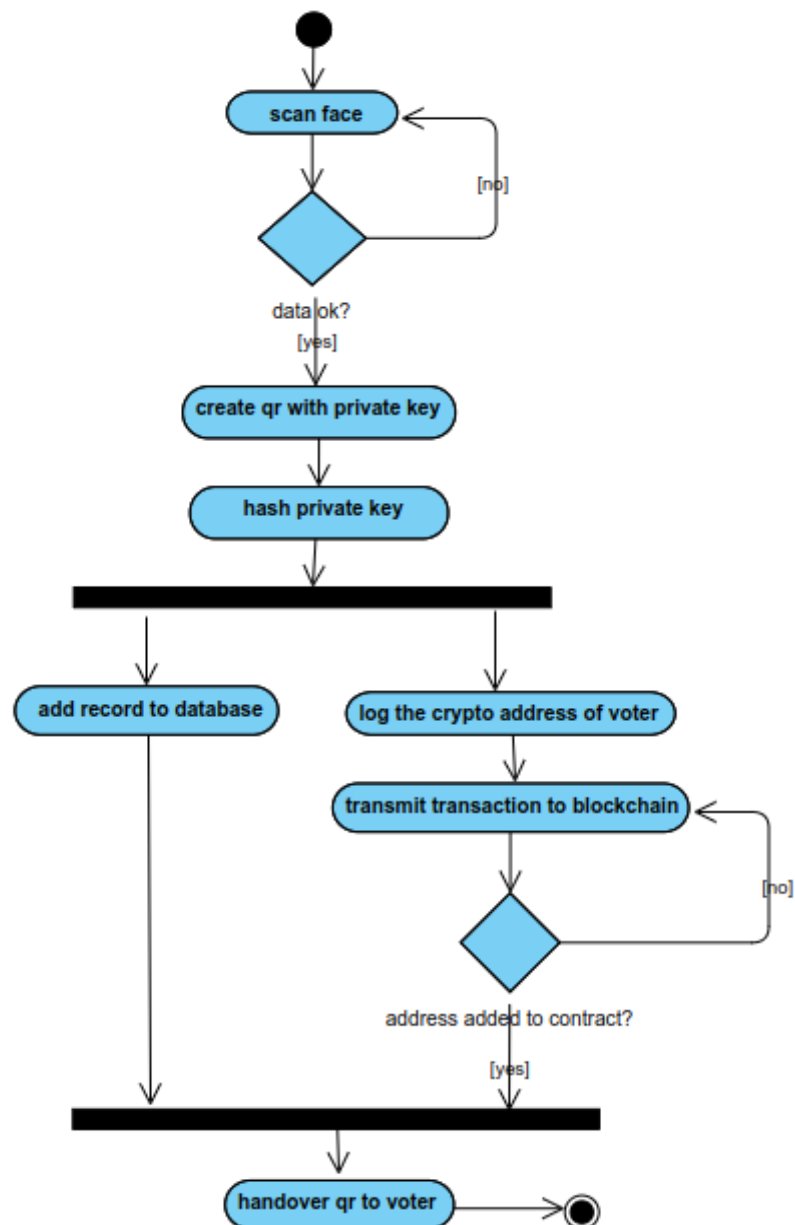


Figure vii: Activity diagram of voters' registration

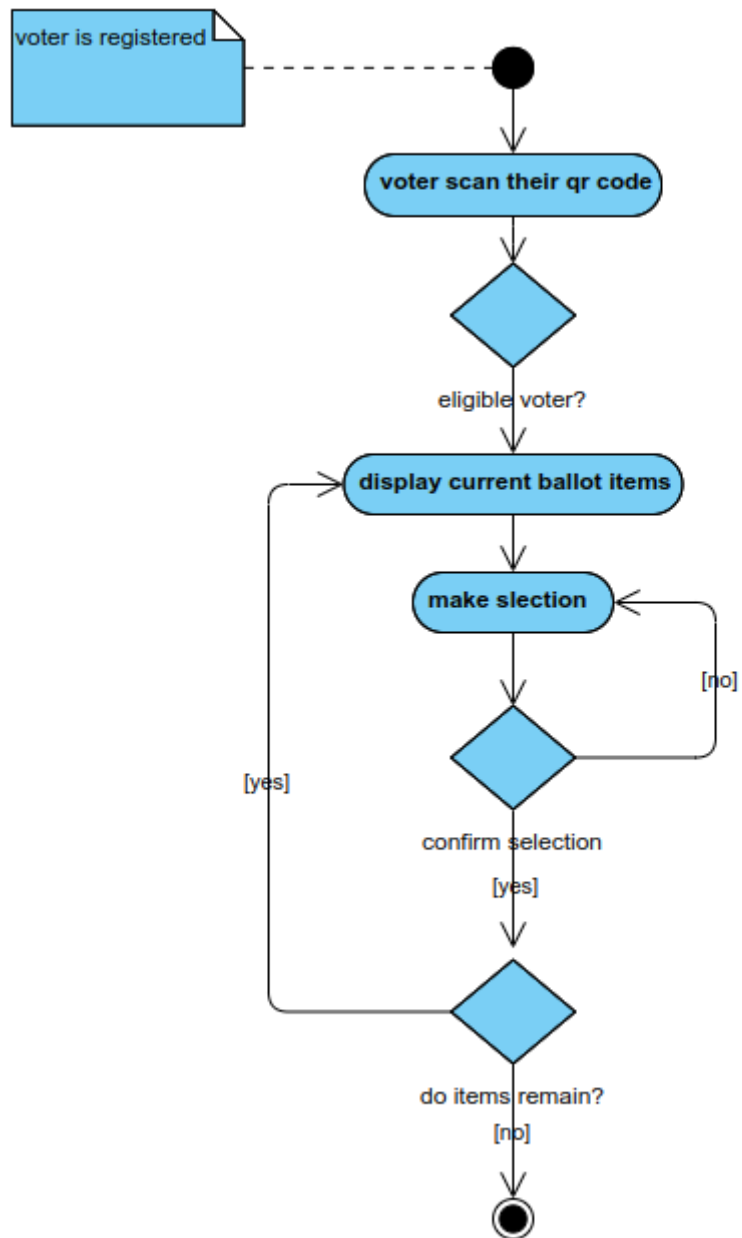


Figure viii: Activity diagram for voting process

5. Time Scheduling

The project has been followed as per requirements and time constraints involved in the table and chart below.

ID	Task Name	Start	Finish	Start on Day	Duration(days)
1	Project Planning	6/23/2022	6/28/2022	0	5
2	Analysis of a System	6/28/2022	7/5/2022	5	7
3	Project Requirement Specifications	7/1/2022	7/10/2022	8	9
4	Increment 1	7/10/2022	7/25/2022	17	15
5	User Registration API	7/10/2022	7/16/2022	17	6
6	Design Basic Diagrams	7/16/2022	7/21/2022	23	5
7	Testing/Evaluation	7/22/2022	7/25/2022	29	3
8	Increment 2	7/26/2022	8/10/2022	33	15
9	Blockchain Development	7/26/2022	8/5/2022	33	10
10	Testing/Evaluation	8/6/2022	8/10/2022	44	4
11	Increment 3	8/11/2022	8/27/2022	49	16
12	Transaction API	8/11/2022	8/19/2022	49	8
13	Blockchain Deployment	8/20/2022	8/23/2022	58	3
14	Testing/Evaluation	8/24/2022	8/27/2022	62	3
15	Increment 4	8/28/2022	9/15/2022	66	18
16	Vote Count API	8/28/2022	9/1/2022	66	4
17	Webcam Face Detection	9/2/2022	9/7/2022	71	5
18	Diagrams	9/8/2022	9/12/2022	77	4
19	Testing/Evaluation	9/12/2022	9/15/2022	81	3

Table i: Time Scheduling

Gantt Chart

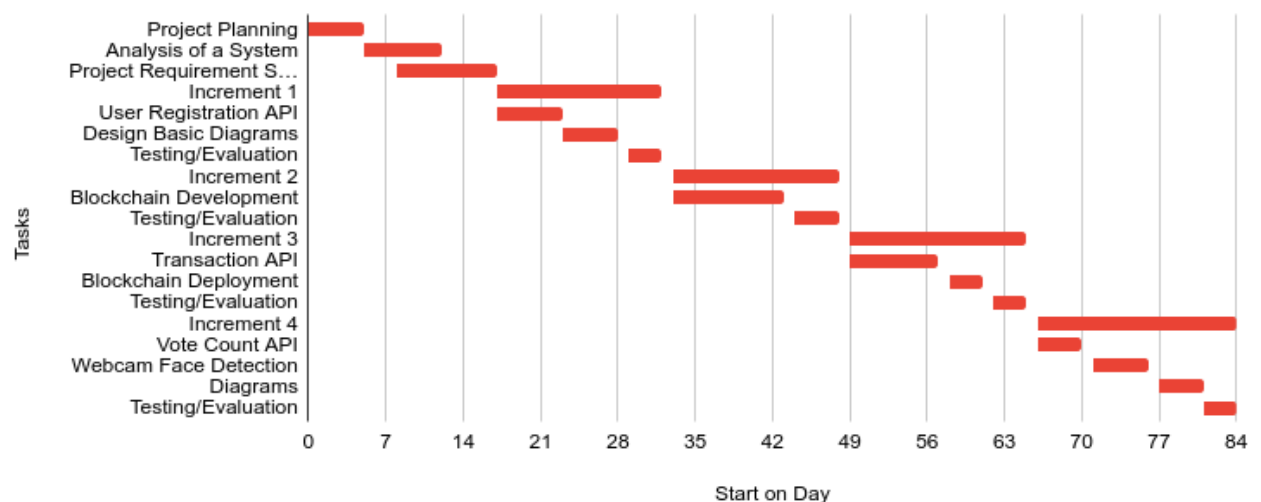


Figure ix: Gantt Chart

6. Testing

Testing is very important as it is necessary to determine whether our work is correct or not. So, we have created a test plan in which our system will be tested with various test cases. The system is tested for the normal condition.

6.1 Testing Table

Test Case #	Test Case Description	Pre-condition	Test Data	Expected Result	Actual Result	Pass/Fail
TC_1	Verify if a user can vote or not.	Voters must be registered previously.	receiver_id=2 current_image=index.jpg qr_code=elon_downloadqr.png	This transaction will be added to Block 2.	A vote has been casted.	Pass
TC_2	Verify if a previously voted voter can vote or not.	Voter must be registered previously And the vote from the same account must be casted previously.	receiver_id=1 current_image=index.jpg qr_code=elon_downloadqr.png	Response: Already Voted	Already Voted	Pass
TC_3	Verify if a voter can vote for the candidate that is not registered.	Voter must be registered previously	receiver_id=5 current_image=index.jpg qr_code=elon_downloadqr.png	Response: Candidate not available.	Candidate not available.	Pass
TC_4	Verify if the voter can vote if the face doesn't match.	Voter must be registered previously	receiver_id=2 current_image=jack_maa.jpg qr_code=elon_downloadqr.png	Response: Invalid accounts	Invalid Accounts	Pass

Table ii: Test Cases

6.2 List of Testing

All testings are given below.

1. TC_1

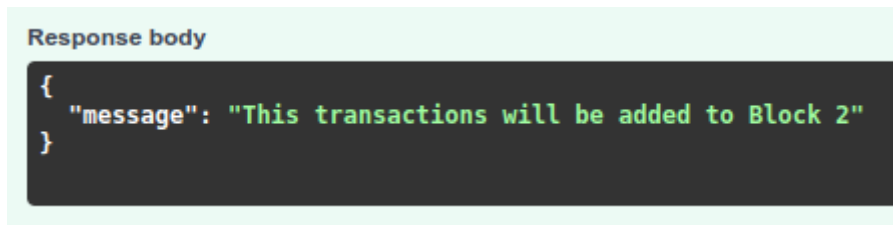
Purpose: To verify if a previously voted voter can vote or not.

Pre-Condition: Voters must be registered previously.

Test Data: receiver_id = 2, current_image = index.jpg, qr_code = elon_downloadqr.png

Expected Output: This transaction will be added to Block 2.

Status: Pass

A screenshot of a REST client's response body. It shows a JSON object with a single key "message" and a value "This transactions will be added to Block 2". The text is displayed in a dark-themed editor with syntax highlighting.

```
Response body
{
  "message": "This transactions will be added to Block 2"
}
```

Figure x: Response of TC_1

2. TC_2

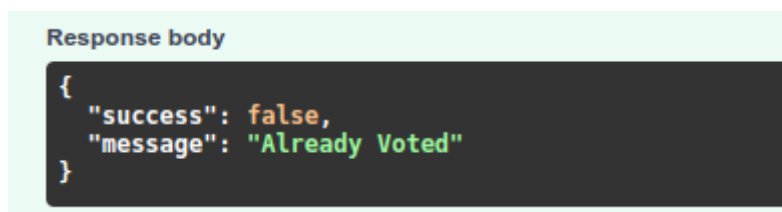
Purpose: To verify if a user can vote or not.

Pre-Condition: Voters must be registered previously. And the vote from the same account must be casted previously.

Test Data: receiver_id = 1, current_image = index.jpg, qr_code = elon_downloadqr.png

Expected Output: Already Voted.

Status: Pass

A screenshot of a REST client's response body. It shows a JSON object with two keys: "success" with a value of "false" and "message" with a value of "Already Voted". The text is displayed in a dark-themed editor with syntax highlighting.

```
Response body
{
  "success": false,
  "message": "Already Voted"
}
```

Figure xi: Response of TC_2

3. TC_3

Purpose: Verify if a voter can vote for the candidate that is not registered.

Pre-Condition: Voters must be registered previously.

Test Data: receiver_id = 5, current_image = index.jpg, qr_code = elon_downloadqr.png

Expected Output: Candidate not available.

Status: Pass

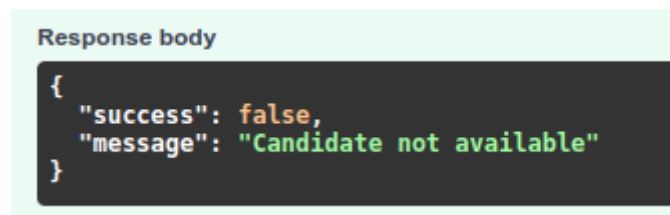


Figure xii: Response of TC_3

4. TC_4

Purpose: Verify if the voter can vote if the face doesn't match.

Pre-Condition: Voters must be registered previously.

Test Data: receiver_id = 2, current_image = jack_maa.jpg, qr_code = elon_downloadqr.png

Expected Output: Invalid accounts.

Status: Pass

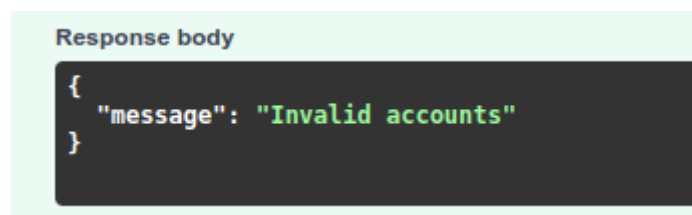


Figure xiii: Response of TC_4

7. Results and Discussion

The work on the described concept can be utilized in the development of a fully functional voting system over a blockchain network. With the immutability property and decentralized architecture of blockchain, properly implemented in the case of digital voting platforms, the wrangling around the voting process can be lessened tremendously. On the one hand, the inclusion of cryptography in the core architecture of blockchain the critical information of the voter and candidates results in the maintenance of anonymity while on the other hand, the public distributed ledger can be viewed by anyone on the system to verify the aftermath of the election.

A blockchain-based digital voting system has been implemented that utilizes our python scripts to enable secure and cost-effective elections while ensuring voters' privacy. The system architecture and the design have been outlined. Compared to the naïve electronic voting system, it has been shown that blockchain technology has tremendous potential for democratic countries to advance from pen and paper schemes to a more cost-effective and time-efficient election scheme and offer new possibilities of transparency. The major stages of the system are Voter Registration, Voting, and Voting confirmation.

8. Conclusion

The concept of digital voting systems to make the electoral process cheaper, faster, transparent and reliable, is fascinating in modern society. Each vote is important and should be registered as one vote can determine the fate of the election. Besides, the paper voting system results in multiple invalid votes. The digital voting system can also be one of the viable solutions to the problem of declining interest among the youth to participate in the election. Hence, to maintain fairness, privacy and verifiability in elections, blockchain has been implemented as the potential solution.

9. Further Works/Recommendations

In this section, we discuss some possible further improvements when applying the digital voting protocol in special elections and scenarios.

For now, we have created only APIs which work as the blueprint for the overall election system. As we are trying to develop the skeleton of our digital voting system which should be run on EVM machines, these APIs are quite enough to implement on those devices. What we can do further is make a UI front-end so that it can simulate the EVM machine like design in our web browser.

Also, we can add later the more authentic details of the users with more validations from proper authorities.

10. References

1. Schaupp LC and Carter L (2005), E-voting: from apathy to adoption, *Journal of Enterprise Information Management*, 18(5): 586–601.
2. Bashir I (2017), *Mastering Blockchain*, Packt Publishing, Mumbai, India.
3. Chase J (2017), *Distributed Systems, Failures and Consensus*, Duke University, USA
4. McCorry, P.; Shahandashti, S.F.; Hao, F. A smart contract for boardroom voting with maximum voter privacy. In *Proceedings of the International Conference on Financial Cryptography and Data Security*, Sliema, Malta, 3–7 April 2017.
5. Zhang, S.; Wang, L.; Xiong, H. Chaintegrity: Blockchain-enabled large-scale e-voting system with robustness and universal verifiability. *Int. J. Inf. Secur.* 2019, 19, 323–341.
6. Chaieb, M.; Koscina, M.; Yousfi, S.; Lafourcade, P.; Robbana, R. DABSTERS: Distributed Authorities Using Blind Signature to Effect Robust Security in E-Voting. Available online: <https://hal.archives-ouvertes.fr/hal-02145809/document> (accessed on 28 July 2020).
7. Woda, M.; Huzaini, Z. A Proposal to Use Elliptical Curves to Secure the Block in E-voting System Based on Blockchain Mechanism. In *Proceedings of the International Conference on Dependability and Complex Systems*, Wrocław, Poland, 28 June–2 July 2021.
8. Hjálmarsson, F.P.; Hreiðarsson, G.K.; Hamdaq, M.; Hjálmtýsson, G. Blockchain-based e-voting system. In *Proceedings of the 2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*, San Francisco, CA, USA, 2–7 July 2018.
9. Lai, W.J.; Hsieh, Y.C.; Hsueh, C.W.; Wu, J.L. Date: A decentralized, anonymous, and transparent e-voting system. In *Proceedings of the 2018 1st IEEE International Conference on Hot Information-Centric Networking (HotICN)*, Shenzhen, China, 15–17 August 2018.
10. Shahzad, B.; Crowcroft, J. Trustworthy Electronic Voting Using Adjusted Blockchain Technology. *IEEE Access* 2019, 7, 24477–24488.
11. Gao, S.; Zheng, D.; Guo, R.; Jing, C.; Hu, C. An Anti-Quantum E-Voting Protocol in Blockchain with Audit Function. *IEEE Access* 2019, 7, 115304–115316.
12. Khan K.M., Arshad J., Khan M.M. Investigating performance constraints for blockchain based secure e-voting system. *Future Gener. Comput.Syst.* 2020;105:13–26. doi: 10.1016/j.future.2019.11.005.