**A Project/Dissertation Report**
**on**

# Decentralized Block-Chain Voting

***Submitted in partial fulfillment***
***of the requirement for the***
***award of the degree of***

B.Tech SCSE


**UNDER THE SUPERVISION**
**OF**
**Mr. ARJUN KP**
**Assistant Professor**


# Submitted By

**PRASHANT SINGH - 19SCSE1010674**
**PRIYADARSHI SHAKYA - 19SCSE1180091**

**SCHOOL OF COMPUTING SCIENCE AND ENGINEERING**
**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING GALGOTIAS UNIVERSITY, GREATER NOIDA**
**INDIA,**

**October,**
**2021**

## SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
## GALGOTIAS UNIVERSITY, GREATER NOIDA

# CANDIDATE'S DECLARATION

I/We hereby certify that the work which is being presented in the thesis/project/dissertation, entitled **"CAPS…."** in partial fulfillment of the requirements for the award of the B.Tech submitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of month, Year to Month and Year, under the supervision of Name… Designation, Department of Computer Science and Engineering/Computer Application and Information and Science, of School of Computing Science and Engineering , Galgotias University, Greater Noida

The matter presented in the thesis/project/dissertation has not been submitted by me/us for the award of any other degree of this or any other place.

Prashant Singh, 19SCSE1010674

Priyadarshi Shakya, 19SCSE1180091

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Mr. Arjun KP

Assistant Professor

# CERTIFICATE

The Final Thesis/Project/ Dissertation Viva-Voce examination of

has been held on _____ and his/her work is recommended for the award of B. Tech.

**Signature of Examiner(s)** **Signature of Supervisor(s)**

**Signature of Project Coordinator** **Signature of Dean**

Date: December, 2021

Place: Greater Noida

# Abstract

Voting is the fundamental right for every nation. An Electronic Voting (E-Voting) system is a voting system in which the election process is notated, saved, stored, and processed digitally, which makes the voting management task better than the traditional paper-based method. Blockchain is offering new opportunities to develop new types of digital services. While research on the topic is still emerging, it has mostly focused on the technical and legal issues instead of taking advantage of this novel concept and creating advanced digital services. Blockchain-enabled e-voting (BEV) could reduce voter fraud and increase voter access. Eligible voters cast a ballot anonymously using a computer or smartphone. BEV uses an encrypted key and tamper-proof personal IDs. Electronic credibility services have become an integral part of the information space. With the reliable implementation of basic services as an electronic signature and electronic authentication, it is possible to build more complex systems that rely on them, particularly the electronic voting system. In this project, the concept of developing an electronic voting system using blockchain technology is implemented. The two-level architecture provides a secure voting process without redundancy of existing (not based on blockchain) systems. The blockchain-based voting project has two modules to make the whole project integrated and work along. One will be the Election Commission who will be responsible for creating elections, adding registered parties and candidates contesting for the election added under the smart contracts. The other end will be the voter's module where each individual can cast a vote for their respective Assembly Constituency and the vote will be registered on the blockchain to make it tamper proof.

**List Of Tables**

**List Of Figures**

## Acronyms

| | |
|---|---|
| EVM | Electronic Voting Machine |
| BEV | Block-Chain Enabled e-voting |
| DRE | Direct Recording Electronics |
| NPV | Net Present Value |
| ROI | Return On Investment |
| TEA | Techno-Economic Assessment |
| FDD | Feature-Driven Development |
| DSDM | Dynamic Systems Development Method |
| PDCA | Plan, Design, Check, Adjust |
| ADCT | Analysis, Design, Code, Test |
| ER | Entity Relationship |

# Table Of Contents

# 1. INTRODUCTION

## i) Overview

Modern democracies are built upon traditional ballot or electronic voting (e-voting). In recent years, devices known as EVMs are hugely criticized due to irregular reports of the election results. There have been many questions regarding the design and internal architecture of these devices and how it might be susceptible to attacks. This paper has analyzed different techniques of tampering the EVMs. Online-voting is pushed as a potential solution to attract the young citizens and the non-resident of the country. For a robust online election scheme, a number of functional and security requirements are to be met such as transparency, accuracy, auditability, data privacy, etc.

We have worked the following ideas by having the two different sets of modules: election commission and the voter(s). The Election Commission creates elections and adds registered candidates along with the parties for contesting the election. Using an election's REST API hosted on Ethereum's Blockchain, the details are shown at the front-end of the voter for casting the vote. Then, while polling the vote is stored on our blockchain framework of which the Election Commission fetches the vote count. The limitation which we have faced due to not using the traditional way of smart contracts is that the blockchain framework which we have coded cannot run on the main net as it needs to be hosted and a separate web3 provider have to be used for interacting with it and not having a public API of voter ID creates a drawback of not having authentication of a voter.

The most important factor of this application is to integrate the blockchain framework with both the modules for seamless voting.

## ii) Objectives

The objectives for developing the project are as follows:

• To improve the existing online voting system using Blockchain technology.

• To reduce the workload of setting up an election booth and conducting elections in physical form.

• Non-Resident Indian can cast their votes as it is totally online.

• We are supposed to learn the concept of Blockchain and how it can be utilized to work on different sectors.

## 2. Literature Review

In this paper, it has highlighted the major problem in voting security where in the 2016 US Presidential Elections, EVM's were likely to be intercepted and votes were tampered. The study found that this old voting equipment is not only more prone to failures and crashes but is also notoriously easy to hack and tamper with. In this study by Ayed, Ahmed, et al., it has been proposed an electronic voting system based on the Blockchain technology. The system is decentralized and does not rely on trust. Any registered voter will have the ability to vote using any device connected to the Internet. The Blockchain will be publicly verifiable and distributed in a way that no one will be able to corrupt it. Rifa and Budi have come to a conclusion that the use of hash values in recording the voting results of each polling station linked to each other makes this recording system more secure and the use of digital signatures makes the system more reliable. The use of the sequence proposed in the blockchain creation process in this system considers that in an electoral system not required for mining as in the Bitcoin system because the voter data and numbers are clear and are not allowed to select more than once, the proposed sequence ensures that all nodes Which is legally connected and can avoid collision in transportation. Bin, Joseph, et al., has come to a conclusion that the current blockchain voting system cannot provide the comprehensive security features, and most of them are platform dependent, we have proposed a blockchain based voting system that the voters' privacy and voting correctness are guaranteed by homomorphic encryption, linkable ring signature, and PoKs between the voter and blockchain.

# 3. SYSTEM ANALYSIS

## 3.1 Identification of Need

Identification of need is a process of determining what and how an end-user would expect a product to perform after the deployment at production level. There Are also nontechnical needs of an end-user or a business client which reflects the users' perception of the product and not the actual technical workaround, but they are closely related to the technical need at times. By implementing a needs identification system, the organization helps to ensure the proper allocation of assets to different project within the organization.

### Identifying Problems

Identifying potential problems before the start of a project can save the organization significant amounts of time and money. Problem analysis is one of the most critical stages of project planning because this stage helps to guide all subsequent analysis and decision-making. If the project does not advance past this stage with solutions that the organization can implement, the project should not go forward in its current form.

### Observations

The needs for a project are identified after the organization makes observations about the project. Observations are often subjective and therefore someone with expertise about the proposed project should help to make observations. A good observer can identify the needs of the project by answering key questions about the project. If the observations take into consideration the project itself and the outcome of the project, the observations should meet all of the needs of the project.

## Gathering Information

Observation and gathering information represent two processes. Observations highlight what is needed. On the other hand, gathering information highlights the processes needed to execute the proposed project. Both observations and the actual gathering of information should include comments from the group that ultimately will benefit from the completed project.

## Objectives and Opportunities

Once the organization has analyzed the needs and identified the objectives, the organization needs to allocate funds to capitalize the project. By successfully identifying the needs, an organization can begin to allocate resources to pay for the project. Additionally, a business needs to consider the potential future cash flow of the project. This allows the business to analyze potential cost savings to minimize costs and maximize the efficiency of the project.

### 3.1.1 Existing System

In India, before 2004 there was a paper-based voting system. This is called the ballot Paper system. Voters had to go to the polling booth and cast their vote by marking on seal in front of the symbol of a candidate for which they wanted to cast their votes on ballot paper. Results were announced by counting the votes. The maximum vote gainer was declared as the winner. India has population more than 120 crores the ballot paper voting is not much reliable, time consuming and very difficult to count the vote and there are also problems like replacement of ballot paper boxes with duplicate, damage of ballot paper, marking stamp seal for more than one candidate hence there is a strong need to overcome these problems. In order to overcome these problems Electronic Voting Machines Were introduced. Electronic Voting Machine (EVM's) mainly consists of two components:

1. Control Unit: It stores and assembles votes, used by poll workers.

2. Ballot Unit: It is placed in the election booth and is used by the voters.

Both the units are connected via 5m cable and one end of the cable is permanently fixed to the ballot unit. The control unit has a battery pack inside, which motorizes the system. The ballot unit has 16 candidate buttons and the unused buttons are covered with a plastic masking tab inside the unit. An additional ballot unit can be connected when there are more than 16 candidates. The additional ballot unit can be connected to a port on the underside of the first ballot unit. EVM's are internationally known as DRE's (Direct recording Electronic). EVM's have been used in India since the general elections of 2004, when ballots were completely out of trend. They have been used in all the assembly polls and general elections of 2009. By using EVM's, Votes are correctly recorded and there is no problem in counting, scalability, Accuracy, fast declaration of results and robustness of the system. Main Problem lies in authentication, the person who is voting may not be the legitimate person. Other problems like capturing of booths by political parties, casting of votes by underage people and fraud voting may occur. A person is provided with the voter id card as a proof of identity, issued by the Indian government. Lot of problems are seen in voter id cards like name misprinting, missing of name, no clear photo on photo id card, etc.

### 3.1.2 Proposed System

Several studies have been done on using computer technologies to improve elections. These studies talk about the risks of adopting electronic voting system, because of the software challenges, insider threats, network vulnerabilities, and the challenges of auditing.

We've proposed to design the existing online voting system which is integrated with the Blockchain technology. The proposed system has the following advantages as compared to the existing system:

• Users' can vote from anywhere in the world until they possess a citizenship of the country.

• The voting is stored in the Blockchain which makes it tamper proof.

• As there's no standing in queue for casting vote it will save a lot of time and reduce the workload.

## 3.2 Preliminary Investigation

The main aim of the preliminary investigation is to identify the problem. First, the need for the new or the enhanced system is established. Only after the recognition of need, then the proposed system is compared and then further analysis is possible. At this stage, we had to perceive the problem and opportunities. The existing system was studied and found out that there were few areas where we can integrate with other technology to make the system better than the existing system. It was analyzed that such a proposed system would be possible to develop with given and it might turn out to be the feasible solution.

In this project, the biggest challenge was to integrate the existing online voting system with the designed blockchain framework and on further development levels we encountered various unit level problems such as the model for the Election Commission to create votes and store the necessary details of candidates along with the election details. In the later part of this document, we have come up with the features which can be added to our software to make it better than the initial deployment.

## 3.3 Feasibility Study

A feasibility study is a high-level capsule version of the entire system analysis and design process. The study begins by classifying the problem definition. The purpose of feasibility study is not to solve the problem, but to determine whether the problem is worth solving. It is a preliminary study which is conducted before the real development of the project commences not keeping the factor of project's success. It creates a roadmap of what are the possible solutions if we choose a certain path. The feasibility study concentrates on the following areas:

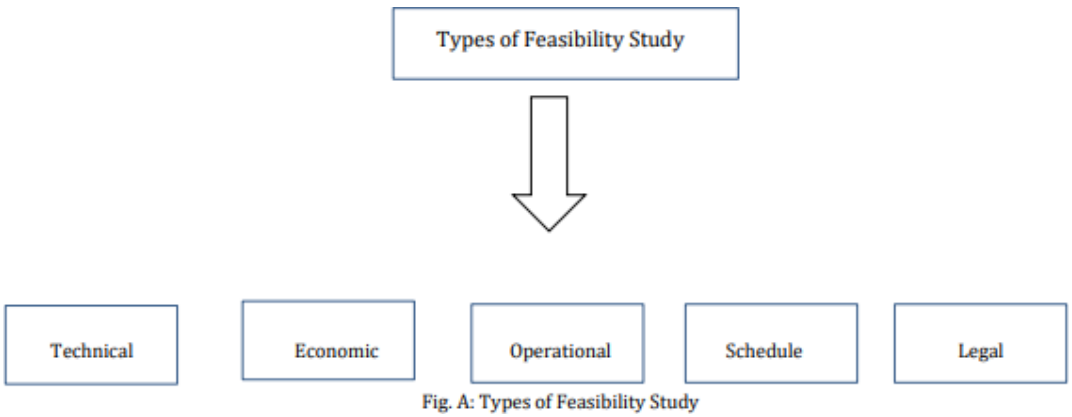



Fig. A: Types of Feasibility Study

### 3.3.1 Technical Feasibility

Evaluating the technical feasibility study is the trickiest part of a feasibility study. This is because, at this point in time, not too many detailed designs of the system, making it difficult to assess issues like performance, costs on (on account of the kind of technology to be deployed) etc. A number of issues have to be considered while doing a technical analysis. Understand the different technologies involved in the proposed system before commencing the project we have to be very clear about what are the technologies that are to be required for the development of the new system. Overall, this study needs to demonstrate that the proposed system which needs to be developed is technically feasible.

This requires:

• An outline of the requirements,

• A possible system design,

• Possible choices of software to be used or developed,

• Estimates on number of users, data, etc.

### 3.3.2 Economic Feasibility

The economic feasibility study evaluates the cost of the software development against the ultimate income or benefits from the developed system. There must be scopes for profit after the successful Completion of the project. The life cycle of an engineering project or product consists of several stages, namely: (i) Planning and design; (ii) Development; (iii) Operation and maintenance. It should be performed to identify the financial risk associated with the project.

Various techniques like net present value (NPV), payback period, return on investment (ROI) are employed. Techno-Economic Assessment (TEA) is a cost-benefit comparison using different methods. These assessments are used for tasks such as:

• Evaluate the economic feasibility of a project.

• Investigate cash flows over the lifetime of the project.

• Evaluate the likelihood of different technology scales and applications.

• Compare the economic quality of different technology applications providing the same service.

### 3.3.3 Operational Feasibility

The operational feasibility study focuses on the degree to which the proposed development project fits in with the existing business environment and objectives with regard to development schedule, delivery date, corporate culture, and existing business processes. It is also the measure of how well the solution will work in the organization after it is deployed. As we are dealing with blockchain voting system, which indirectly targets the country's or state's election process protocol, so there will be a detailed comparison between these two to check

which one dominates the other. It is also the measure of how people will feel about the project as will people be accustomed to use this in a proper way or it will be too complex to deal with.

There are two aspects of operational feasibility to be considered:

• Is the problem worth solving?

• How do the end user (voters in this case) and management (Election Commission) feel in this case?

### 3.3.4 Schedule Feasibility

It means that the project can be implemented in an acceptable time frame. When assessing schedule feasibility, a systems analyst must consider the interaction between time and costs. For example, speeding up a project schedule might make a project feasible, but much more expensive.

Other issues that relate to schedule feasibility include the following:

• Can the company control the factors that affect schedule feasibility?

• Has management established a firm timetable for the project?

• What conditions must be satisfied during the development of the system?

• Will an accelerated schedule pose any risks? If so, are the risks acceptable?

• Will project management techniques be available to coordinate and control the project?

• Will a project manager be appointed? It is also the likelihood that timeframes can be met and that this is adequate to meet organization's needs.

### 3.3.5 Legal Feasibility

It determines whether the proposed system conflicts with the legal requirements, in this case as we didn't try to execute anything on the public domain, hence this project is legally feasible.

It is important that the project is following the requirements needed to start a project including certificates, copyrights, business insurance, tax number, health and safety measures and many more. There are some things to consider in legal feasibility study including ethical issues and some social issues. These issues are privacy and accountability. In this project, everything is designed keeping in mind all the legal terms and no real-world data or privacy has been breached for any person of this country to use as a sample voter to implement this application.

## 3.4 Project Planning

Project Planning is the most essential thing in developing a project. It sets out the phases, activities and task needed to deliver a project. The timeframes required to deliver the project, along with the resources and milestones are also shown on the project plan.

Initially, the project scope is defined and the appropriate methods for completing the project are determined. Following this step, the durations for the various tasks necessary to complete the work are listed and grouped into a work breakdown structure. Project planning is often used to organize different areas of a project, including project plans, workloads and the management of teams and individuals. The logical dependencies between tasks are defined using an activity network diagram that enables identification of the critical path. Project planning is inherently uncertain as it must be done before the project is actually started. Therefore, the duration of the tasks is often estimated through a weighted average of optimistic, normal, and pessimistic cases. The critical chain method adds "buffers"; in the planning to anticipate potential delays in project execution. Float or slack time in the schedule can be calculated using project management software. Then the necessary resources can be estimated and costs for each activity can be allocated to each resource, giving the total project cost. At this stage, the project schedule may be optimized to achieve the appropriate balance between resource usage and project duration to comply with the project objectives. Once established and agreed, the project schedule becomes what is known as the baseline schedule. Progress will be measured against the baseline schedule throughout the life of the project. Analyzing progress compared to the baseline schedule is known as earned value management.

A project plan is a model of the process that the project team intends to follow to realize the project objectives. It brings together a number of important aspects of this process including its scope, timing and associated risks. The project plan can be viewed as a type of "contract" between the project team members and the reviewers. It defines the process by which objectives will be achieved, and the responsibilities in carrying out this process. It also underpins a number of other key project management functions including estimating and forecasting, options analysis and decision-making, and performance monitoring and control.

The essential elements of a project plan are:

• Scope statement

• Schedule

• Requirements

• Quality criteria

• Project resources

• Communications Plan

## Scope statement

It is a statement of what work is included within the project, and what is not. A good scope statement significantly reduces risk of project overruns and unexpected turbulence. In this project, the scope statement is as follows:

“This project is for the creation of an online election system using Blockchain technology. There will be a website for the Election Commission and for the voters. The user interface will be designed as part of the project which will contain necessary details at both the end”.

## Schedule

The project schedule communicates to all stakeholders what the expected arrival time will be, and serves to keep the project manager’s hands on the throttle throughout the project. Since projects are by definition temporary endeavors with a defined beginning and end, the exact location of that end date is a primary consideration for most projects.

The details of this project’s schedule will be discussed later under Project Scheduling.

## Requirements

All projects have requirements which are drafted at the beginning as per client’s needs. In this project, the requirements are such as the module of creating elections, adding candidates contesting the elections. Detailed discussion of the requirements is discussed under Requirement Specifications.

## Quality Criteria

It is one of the essential elements of project planning as if a software is not inspected properly and then deployed to the market it might cause few problems which will then create pressure among the maintenance. The quality criteria should be identified in the project plan, including pass/fail requirements, as well as the methods used to ensure the quality criteria will be met.

## Project Resources

Resources often require the most planning and coordination throughout the project’s execution. That’s because they arrive late, require unexpected maintenance, don’t meet specifications, or any other host of issues that can trip up a project. Resources are the technology stack which will be used in developing the software. Details about this is discussed at the later part of this documentation.

## 3.5 Project Scheduling

It requires us to follow some carefully laid-out steps, in order for the schedule to take shape. It is an organized method of presenting information on when activities need to be started, and how long activities are planned to be completed.

There are basic principles for project scheduling, such as follows:

**• Defined responsibilities**

- Every task that is scheduled is assigned to a specific team member.

**• Defined outcomes**

- Every task that is scheduled should have a defined outcome for software projects such as a work product.

**• Define milestones**

- Every task or group of tasks should be associated with a project milestone.
- A milestone is accomplished when one or more work products has been reviewed and then approved by the team leader.

PERT and GANTT Chart were developed to represent the project schedule and track the different tasks.

## 3.6 Software Requirement Specification

### 3.6.1 Introduction

This document describes the structural properties and software requirements of the Online Election System using Blockchain Technology.

#### 3.6.1.1 Problem Definition

Manual voting system has been deployed for many years in our country. However, in many parts of our country people cannot attend the voting because of several reasons. To illustrate, sometimes people may not be in their own registration region and due to this fact, they cannot fulfill their voting duties. In order to solve these problems, there is a need of online election voting system with this keeping in mind that EVM votes tampering issues are also encountered, so this online election system will be integrated with Blockchain Technology to make it tamper proof.

#### 3.6.1.1 Purpose

The purpose of this document is to make the functional and non-functional requirements of the Online Election System using Blockchain Technology easy to comprehend. It also serves the purpose of making the functionality clear to end users.

### 3.6.1 Scope

This SRS document applies to the initial version (release 1.0) of the "Online Election System using Blockchain Technology" software package. This document describes the modeling and the requirement analysis of the system. The main aim of the system is to provide a set of protocols that allow voters to cast votes while the election commission is responsible for creating elections and adding candidates.

### 3.6.2 Functional Requirements

#### 3.6.2.1 Software Requirements

| 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 |
| Windows 10 | Operating System | 1809 |

### 3.6.3 Non-functional Requirements

### 3.6.3.1 Performance Requirements

The system is expected to have a reasonable short time of response. The voter should be able to import his/her wallet provided by the Election Commission within a few seconds keeping in mind the condition of network stability. The system's performance is different according to its modes:

**(i) Election Mode:** In this phase, the expected time to deploy the smart contracts totally depends upon the miners connected to the blockchain and the amount of GAS we decide to sign off the transaction to mark as validated one but as we are working locally, it is just a matter of half a minute or so.

**(ii)** Voting Mode: In this phase, the system will be responding within seconds as we don't have to sign off transaction just to fetch the list of candidates for the elections but depending on the network

stability and web3 connection the above performance might be delayed. Next, after casting the vote it might take a minute or two to sign off the transaction depending upon the miners and GAS limit.

### 3.6.3.1 Security Requirements

- The data transaction between client and the blockchain server must be done over https to avoid mixed content attack.
- The re-entrancy on a single function has to be minimized while deploying the smart contract.
- To address the integer overflow error, the idea of counting the votes have been done within a specific event responsible for it.

### 3.6.3.2 Reliability

- In Election Mode: The system needs to be maintained from time to time as if the smart contract which is to be deployed encounters any bugs, it needs to be fixed to prevent votes miscalculation and transaction error handling.
- In Voting Mode: As the maintenance part is in the Election Mode, if there's any error in web3 connection the interoperability status might change otherwise the system will work flawlessly all the time.

### 3.6.3.3 Usability

- The system will have a minimal and simple User Interface.
- To guide the users for the first time using it, there will be guidance related to the usage of the system.

## 3.7 Software Engineering Paradigm

This project uses an iterative model approach using Agile methodologies. Let's discuss this in detail. Agile methods of software development are most commonly described as iterative and incremental development. The iterative strategy is the cornerstone of Agile practices, most prominent of which are SCRUM, DSDM, and FDD. The general idea is to split the development of the software into sequences of repeated cycles (iterations). Each iteration is issued a fixed-length of time known as a timebox. A single timebox typically lasts 2-4 weeks.

The ADCT (Analysis, Design, Code, Test) wheel is more technically referred to as the PDCA (Plan, Design, Check, Adjust) cycle. The team implements the PDCA cycle on each iteration separately in the following manner:

**• P (Plan) – Iteration Planning**

In this event, the team collaborates to discuss the objectives for the next iteration. It also summarizes the work done and determines the team backlog required for the next iteration.

**• D (Design) – Iteration Execution**

This is the 'do' step where the development of the software, its design and coding takes place. If it's a second or third iteration, then functionality testing is also conducted. The team collects user stories and prepares for the next step, that is the Iteration Review.

**• C (Check) – Iteration Review**

Also known as the 'check' step, Iteration Review is carried out with the Product Owner. The team shows the tested deliverable to the Product Owner, who then reviews the completed work and ascertains whether all criteria have been met.

**• A (Adjust) – Iteration Retrospect**

In this event, the team evaluates the entire process of the iteration from the first step. It essentially works on any improvements that are gathered in previous iterations. New problems are identified along with their causes. Before the team starts the next cycle again, team backlog is refined for future reference. The iterations are repeated for optimizations and improvisations and, the lessons learned from previous cycles are applied in the next cycle. Until a fully functional software is ready to hit the market.

Agile methodologies have the following advantages over other methods:

**Customer Involvement** – Agile Iterative development encourages user contribution. After each iterative cycle, customer feedback is obtained, and the product is then subjected to necessary changes based on that feedback. This aspect brings adaptability into the project's framework.

**Favors Evolution** – The planning in the Agile Iterative development process is a continuous feat, that allows space for evolving ideas, instead of extensive planning that only precedes execution and testing in Waterfall.

**Risk Assessment** – Agile iteration allows risk identification and mitigation early on in the development to avoid speed bumps later down the timeline.

**Rapid Delivery** – The work is divided into small cycles, allowing team members to dedicate their focus and deliver on time. Moreover, testing is conducted simultaneously in coding and design in every iteration, which greatly reduces the time needed to achieve completion.

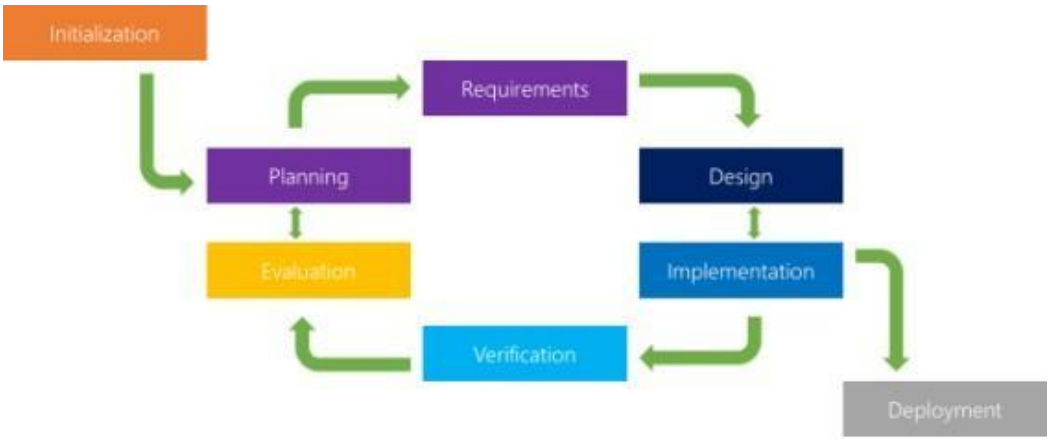



*Figure 1: Iterative Model of Software Development*

## 3.7.1 Process

- **Planning & Requirements:** As with most any development project, the first step is to go through an initial planning stage to map out the specification documents, establish software or hardware requirements, and generally prepare for the upcoming stages of the cycle.

- **Analysis & Design**: Once planning is complete, an analysis is performed to nail down the appropriate business logic, database models, and the like that will be required at this stage in the project. The design stage also occurs here, establishing any technical requirements (languages, data layers, services, etc.) that will be utilized in order to meet the needs of the analysis stage.

- **Implementation:** With the planning and analysis out of the way, the actual implementation and coding process can now begin. All planning, specification, and design docs up to this point are coded and implemented into this initial iteration of the project.

- **Testing**: Once this current build iteration has been coded and implemented, the next step is to go through a series of testing procedures to identify and locate any potential bugs or issues that have cropped up.

- **Evaluation**: Once all prior stages have been completed, it is time for a thorough evaluation of development up to this stage. This allows the entire team, as well as clients or other outside parties, to examine where the project is at, where it needs to be, what can or should change, and so on.

## 3.8 Data Models and Description

### 3.8.1 Sequence Diagram

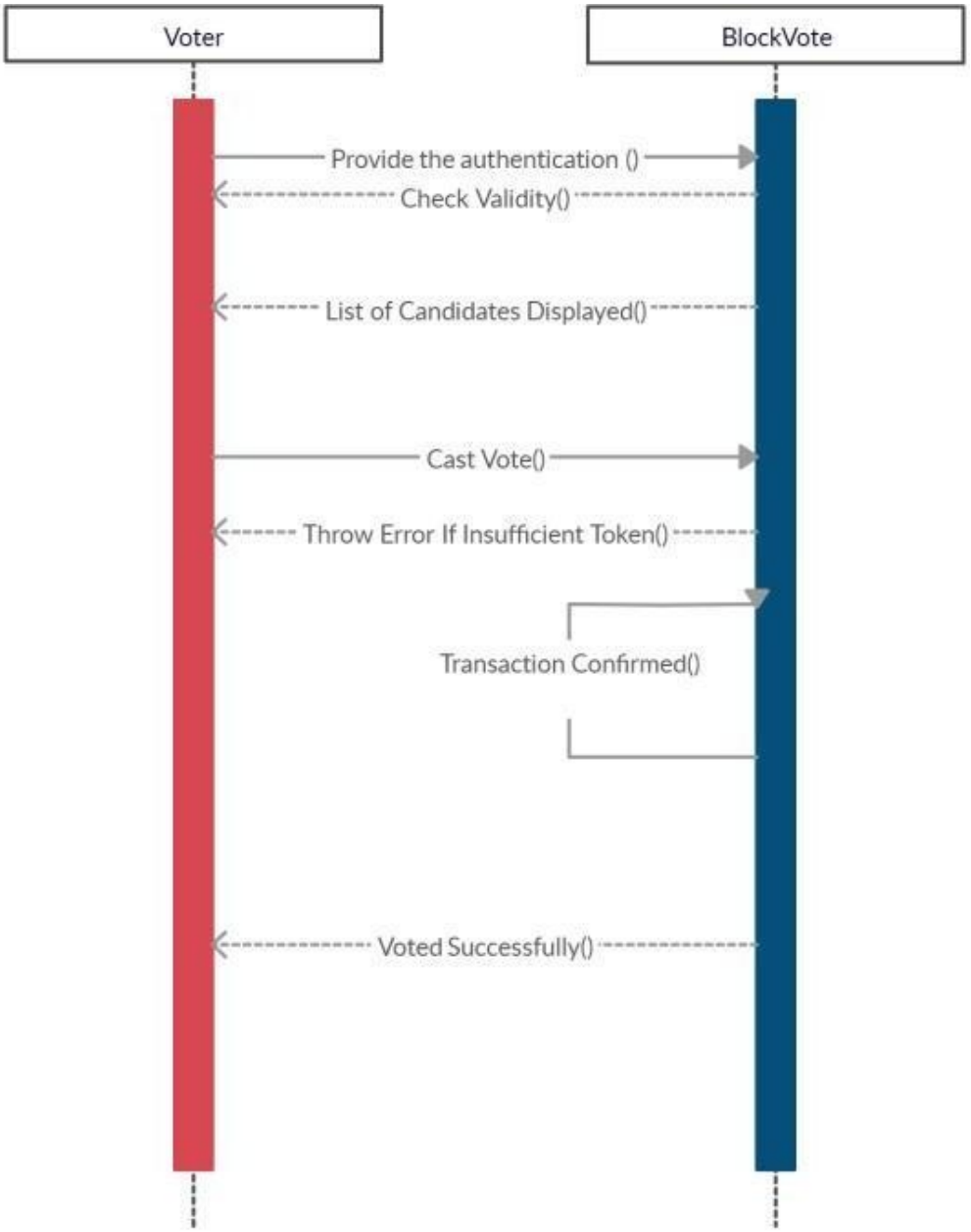
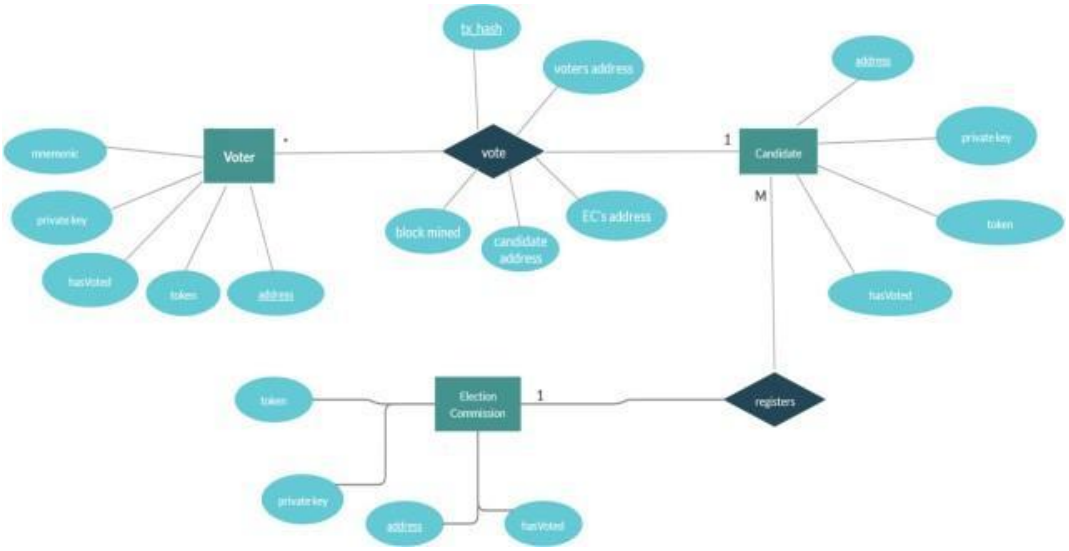
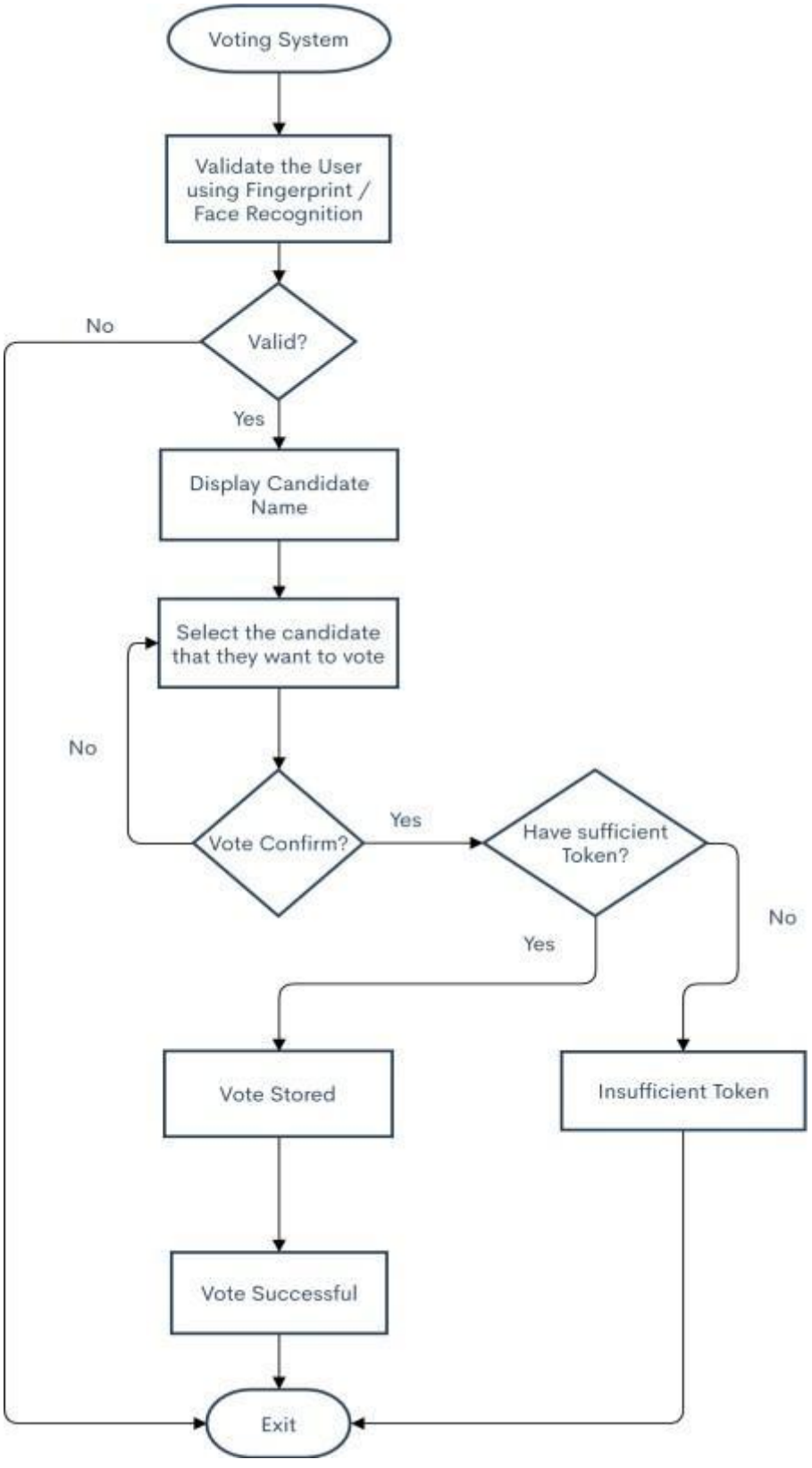



*Figure 2: Sequence Diagram*

### 3.8.2 Entity Relationship Diagram



### 3.8.3 Flowchart



# 4. CODING

The above system design is translated into a machine-readable form which is termed as coding. It is basically translating the human readable format to a machine friendly one. The code generation step performs this task. The following points are considered while converting the system design into coding:

• Are the initializations correct?
• Are the data types properly assigned?
• Is memory leak being dealt with?
• Does it comply with the coding standard?

## 4.1 Coding Standardization

Coding Standardization Coding Standardization basically the efficiency of our code which has been converted from the system design. The efficiency primarily depends upon:

• Readability: The code should be readable with proper indentation and spacing to make the contents clear of all the modules.

• Portability: The code is portable enough as it will work on various platform given all the necessary dependencies are installed.

• Debug Easily: The coding should be error-free as much as possible

## 4.2 Source Code

```
pragma solidity >=0.5.16;

//name of the Contract

contract Migrations {
  address public owner;
  uint public last_completed_migration;

  modifier restricted() {
    if (msg.sender == owner) _;
  }

//assigning the sender of the transaction to be owner
  constructor () public{
    owner = msg.sender;
  }

//Setting up the migration for the first time to be deployed on the Blockchain
  function setCompleted(uint completed) public{
    last_completed_migration = completed;
  }

//On necessary changes, upgrade function is triggered
  function upgrade(address new_address) public {
    Migrations upgraded = Migrations(new_address);
    upgraded.setCompleted(last_completed_migration);
  }
}
```

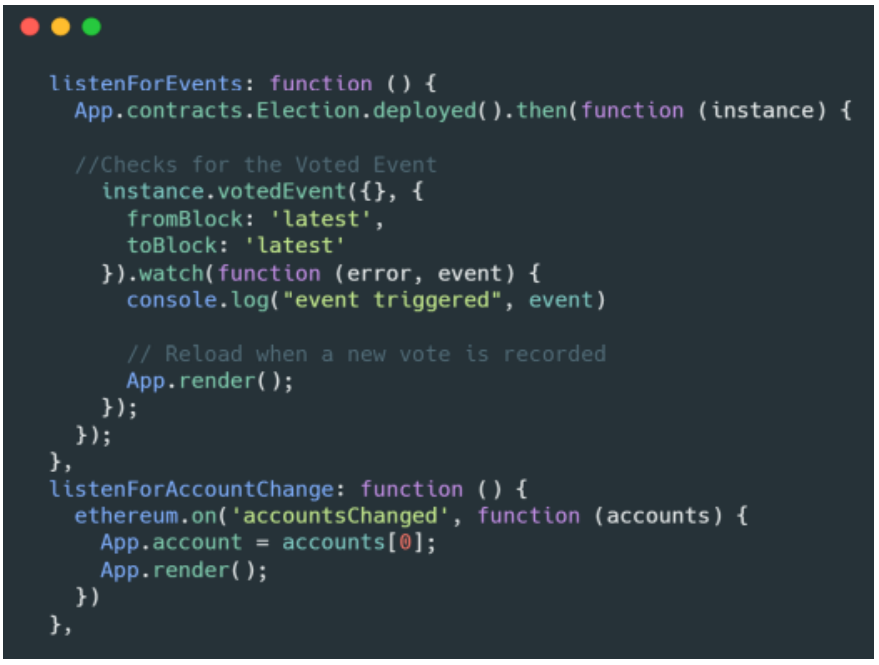
Migration Smart Contract

```solidity
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
    );

    //Adding Election Candidates along with the parties
    constructor () public {
        addCandidate("Raju Bista","Bharatiya Janata Party");
        addCandidate("Sankar Malakar","Indian National Congress");
        addCandidate("Saman Pathak","Communist Party Of India (Marxist)");
        addCandidate("Amar Singh Rai","All India Trinamool Congress");
        addCandidate("Sudip Mandal","Bahujan Samaj Party");
        addCandidate("NOTA","None of the above");
    }

    //Function to trigger the adding candidates
    function addCandidate (string memory name,string memory party) private {
        candidatesCount ++;
        candidates[candidatesCount] = Candidate(candidatesCount, name,party, 0);
    }

    function vote (uint _candidateId) public {
        // require that they haven't voted before
        require(!voters[msg.sender]);

        // require a valid candidate
        require(_candidateId > 0 && _candidateId <= candidatesCount);

        // record that voter has voted
        voters[msg.sender] = true;

        // update candidate vote Count
        candidates[_candidateId].voteCount ++;

        // trigger voted event
        emit votedEvent(_candidateId);
    }
}
```

Election Smart Contract

```
initContract: function () {
  $.getJSON("Election.json", function (election) {

    // Instantiate a new truffle contract from the artifact
    App.contracts.Election = TruffleContract(election);

    // Connect provider to interact with contract
    App.contracts.Election.setProvider(App.web3Provider);

    //invokes listen for Events
    App.listenForEvents();
    App.listenForAccountChange();

    return App.render();
  });
},
```

Initialization of Smart Contract



```
listenForEvents: function () {
  App.contracts.Election.deployed().then(function (instance) {

  //Checks for the Voted Event
    instance.votedEvent({}, {
      fromBlock: 'latest',
      toBlock: 'latest'
    }).watch(function (error, event) {
      console.log("event triggered", event)

      // Reload when a new vote is recorded
      App.render();
    });
  });
},
listenForAccountChange: function () {
  ethereum.on('accountsChanged', function (accounts) {
    App.account = accounts[0];
    App.render();
  })
},
```

Trigger voted events

```
castVote: function () {
    var candidateId = $('#candidatesSelect').val();
    App.contracts.Election.deployed().then(function (instance) {
      return instance.vote(candidateId, { from: App.account });
    }).then(function (result) {
      // Wait for votes to update
      $("#content").hide();
      $("#loader").show();
      alert("Thanks for voting")
    }).catch(function (err) {
      console.error(err);
    });
  }
```

CastVote: Function to vote

```
App.contracts.Election.deployed().then(function (instance) {
      electionInstance = instance;
      return electionInstance.candidatesCount();
    }).then(function (candidatesCount) {
      var candidatesResults = $("#candidatesResults");
      candidatesResults.empty();

      var candidatesSelect = $('#candidatesSelect');
      candidatesSelect.empty();

      for (var i = 1; i <= candidatesCount; i++) {
        electionInstance.candidates(i).then(function (candidate) {
          var id = candidate[0];
          var name = candidate[1];
          var voteCount = candidate[3];
          var party = candidate[2];
          // Render candidate Result
          var candidateTemplate =
            `<tr><th>${id}</th><td>${name}</td><td>${party}</td><td>${voteCount}</td></tr>`;
          candidatesResults.append(candidateTemplate);

          // Render candidate ballot option
          var candidateOption = `<option value="${id}"> ${name} (${party}) </option>`
          candidatesSelect.append(candidateOption);
        });
```

Front-End integration of the Election

# 5. Conclusion

Democracies depend on trusted elections and citizens should trust the election system for a strong democracy. However traditional paper-based elections do not provide trustworthiness. The idea of adapting digital voting systems to make the public electoral process cheaper, faster and easier, is a compelling one in modern society. Making the electoral process cheap and quick, normalizes it in the eyes of the voters, removes a certain power barrier between the voter and the elected official and puts a certain amount of pressure on the elected official. It also opens the door for a more direct form of democracy, allowing voters to express their will on individual bills and propositions. This project has been developed into a blockchain-based electronic voting system that utilizes smart contracts to enable secure and cost-efficient elections while guaranteeing voters privacy. It outlines the systems architecture, the design, and a security analysis of the system. In the next build of this application, it has been proposed to create separate client designs for various roles such as one for election commission and one for candidates registered to a certain party with the existing voting client design. Also, the current versions lack authentication as we don't have access to the current Aadhar or Voter SDK to integrate in our application. Also, it is planned that in the next build notification prompt will be given on the day of voting to all the voters to cast their vote so that the voter turnout is maximum for that electionDemocracies depend on trusted elections and citizens should trust the election system for a strong democracy. However traditional paper-based elections do not provide trustworthiness. The idea of adapting digital voting systems to make the public electoral process cheaper, faster and easier, is a compelling one in modern society. Making the electoral process cheap and quick, normalizes it in the eyes of the voters, removes a certain power barrier between the voter and the elected official and puts a certain amount of pressure on the elected official. It also opens the door for a more direct form of democracy, allowing voters to express their will on individual bills and propositions. This project has been developed into a blockchain-based electronic voting system that utilizes smart contracts to enable secure and cost-efficient elections while guaranteeing voters privacy. It outlines the systems architecture, the design, and a security analysis of the system. In the next build of this application, it has been proposed to create separate client designs for various roles such as one for election commission and one for candidates registered to a certain party with the existing voting client design. Also, the current versions lack authentication as we don't have access to the current Aadhar or Voter SDK to integrate in our application. Also, it is planned that in the next build notification prompt will be given on the day of voting to all the voters to cast their vote so that the voter turnout is maximum for that election.

# 6. References

[1] Wolchok, Scott, et al. "Security analysis of India's electronic voting machines." Proceedings of the 17th ACM conference on Computer and communications security. ACM, 2010.

[2] Ohlin, Jens David. "Did Russian cyber interference in the 2016 election violate international law." Tex. L. Rev. 95 (2016): 1579.

[3] Ayed, Ahmed Ben. "A conceptual secure blockchain-based electronic voting system." International Journal of Network Security & Its Applications 9.3 (2017): 01-09.

[4] Hanifatunnisa, Rifa, and Budi Rahardjo. "Blockchain based e-voting recording system design." 2017 11th International Conference on Telecommunication Systems Services and Applications (TSSA). IEEE, 2017.

[5] Yu, Bin, et al. "Platform-independent secure blockchain-based voting system." International Conference on Information Security. Springer, Cham, 2018.