An investigation into developing an E-Voting decentralized application that operates through Blockchain

Syed Mohammad Shah Mostafa

001322473

Supervisor: Mustafizur Rahman

COMP1682 Final Year Project

A dissertation submitted in partial fulfilment of the University of Greenwich's

Program Title: BSc (Hons) COMPUTING

Due date: 28 November 2023

Word count: 36,263

Abstract

This thesis embarks on a transformative journey to explore the feasibility and potential of a blockchain-based voting system Dapp. The central aim is to optimize the system's performance while addressing critical concerns surrounding security and reliability in the context of electoral processes. The research endeavors to identify inventive solutions, introducing novel methods to mitigate vulnerabilities and streamline the voting experience within the robust framework of blockchain technology.

At its core, this thesis delves into the pioneering concept of "E-voting using Blockchain" as a groundbreaking solution to persistent challenges in electoral systems, including security lapses, transparency issues, and the imperative need for confidentiality. It provides a comprehensive examination of the technical underpinnings of blockchain-based voting, emphasizing its inherent decentralization and encryption attributes. The study illuminates the myriad benefits brought about by the integration of blockchain, including enhanced data security and fraud prevention.

Moreover, this study conscientiously addresses the multifaceted challenges, legal considerations, and ethical dilemmas that accompany the adoption of this revolutionary technology. Real-world case studies offer invaluable insights into practical applications. The goals of study have been achieved by creating a comprehensive web application based on HTML, JavaScript, CSS, ReactJS, Node JS, MySQL and solidarity using the agile DSDM methodology. The developed application has been tested and the results are satisfactory, achieving all the objectives of this study. In the future, it would be interesting to enhance the application for a larger scale use to examine its practicality. In essence, this research offers a visionary perspective on how blockchain technology can potentially revolutionize electronic voting, safeguarding the pillars of democracy i.e., transparency, privacy, and accessibility in the digital age.

.

Acknowledgement

I want to sincerely thank the project supervisors for their invaluable help and constant encouragement throughout the project's various phases.

Originality Statement

This report is offered as a partial requirement for the University of Greenwich's degree in computing. Unless otherwise noted and properly cited in the report, I hereby declare that I am the only author of this work.

Table of Contents

List of Tables

List of Figures

1 Introduction

The rapid growth of technology has opened new frontiers in a variety of disciplines, including the political process. This document goes into the novel area of Electronic Voting (E-voting) and the building of a secure and transparent voting system employing cutting-edge technologies. E-voting has the potential to completely transform the way we conduct elections, making them more accessible and efficient while also increasing security and trust. A solid technology stack was used to bring this vision to reality, combining agile development approaches such as Agile and DSDM, extensive testing processes such as user acceptability and functional testing, and scripting languages such as Solidity for smart contract development. This project takes advantage of the potential of decentralized applications within the Ethereum framework, employing tools such as Truffle and Ganache to maintain the integrity of the voting process. The MetaMask browser add-on is included to enhance user engagement, making E-voting a seamless experience for participants. This document sets out on a journey to demonstrate the modern-day potential of E-voting, emphasizing the importance of secure and accessible electoral systems.

1.1 Background

The current electoral systems in numerous countries confront persistent challenges concerning transparency, security, and public confidence in the accuracy of the electoral process (Ahmad et al. 2016). While traditional voting systems have made commendable progress in mitigating some of these concerns, apprehensions about data integrity and centralized control continue to linger (Nittari et al.2020). The traditional paper ballot voting system clearly lacks critical characteristics of a trustworthy election, such as transparency, accountability, and fairness, and is vulnerable to human errors and manipulations (Ahmad et al. 2016).

Blockchain technology has effectively addressed current electronic voting system issues by improving accessibility, reducing fraud, enhancing data security, and verifying election results. Its application in electronic voting is essential (Prashar et al. 2020). When blockchain technology was originally brought to the internet, it immediately acquired popularity because of its high level of system transparency (Abhishek et al. 2020). To increase the transparency, security, and overall credibility of electoral processes, this project aims to explore the viability and benefits of using a blockchain-based voting system.

Furthermore, using traditional transportation means exposes election results to a variety of risks, including potential attacks by political agitators, angry party members, or manipulation by unscrupulous authorities. These difficulties undercut the traditional paper ballot system's effectiveness and cast doubt on its long-term viability. This condition also provides an opportunity for the introduction of e-voting because electronic result compilation and sharing provides a more secure alternative (Ahmad et al. 2016).

This problem statement encapsulates the core components of the research theme and establishes a clear focal point for the ensuing report. It not only identifies the prevailing issues in contemporary electoral systems but also underscores the constraints of existing solutions, subsequently introducing the concept of leveraging blockchain technology as a potential panacea.

1.2 Aims

This project's main goal is to investigate whether a decentralized application (DApp) built around a blockchain-based voting system is even feasible. Our primary goal is to enhance the DApp's overall performance while conscientiously attending to dependability and security issues. We are dedicated to finding and applying novel solutions during this exploration, including fresh methods to strengthen security and speed up voting within the context of blockchain technology. Our ultimate objective is to improve the voting process by making it more effective and secure and by boosting the system's resistance to potential dangers.

1.3 Objectives

Objective One - Research

1.3.1 Research Current Voting Systems and Blockchain Technology

- Investigate the existing challenges and vulnerabilities in traditional voting systems. This includes understanding the common types of fraud, inefficiencies, and limitations of current systems.
- Analyze the potential benefits and drawbacks of implementing blockchain technology in the e-voting system. This involves understanding how blockchain can enhance security, transparency, and efficiency in voting.
- Study existing blockchain-based e-voting systems and their security mechanisms. This includes understanding how these systems work, what security measures they have in place, and how they handle common threats.

1.3.2 Research Blockchain Architecture

- Implement a secure and transparent blockchain architecture tailored for electronic voting. This involves understanding how to design a blockchain system that can handle the specific requirements of an e-voting system.

- Develop smart contracts using Solidity, Truffle, and JSON to manage the voting process. This includes understanding how to write smart contracts that can handle voting transactions, ensure accuracy and transparency, and prevent fraud.
- Implement the Proof of Authority mechanism for e-voting. This involves understanding how this consensus algorithm works and how it can be used to ensure security and fairness in an e-voting system.

1.3.3 Conduct Requirements Analysis

- Conduct a comparative analysis of different systems. This involves comparing different e-voting systems to understand their strengths and weaknesses.
- Conduct a MoSCoW analysis to prioritize functional requirements. This involves identifying the Must have, Should have, Could have, and Won't have requirements for the e-voting system.

1.3.4 Objective Two - Design

- Design an easy-to-use interface that enables voters to participate in the electronic voting process. This involves understanding user needs and preferences, designing a user-friendly interface, and ensuring accessibility for all users.

1.3.5 Objective Three - Development

- Develop an HTML-based e-voting system utilizing blockchain technology to facilitate online voting for voters. This involves coding the front-end using HTML, CSS, JavaScript, and ReactJS.
- Use MySQL for database management, JavaScript for server-side scripting, and Node.js for runtime environment in back-end development.

1.3.6 Objective Four - Testing

- Conduct extensive testing of the e-voting system to identify and address potential vulnerabilities. This includes performing security testing to identify

potential threats and vulnerabilities, functionality testing to ensure all features work as expected, and usability testing to ensure the system is easy to use.

- Perform testing on the MySQL database to ensure data integrity and security.

1.3.7 Objective Five - Deployment

- Execute a phased deployment of the e-voting system. This involves deploying the system in stages to monitor its performance and make necessary adjustments.

1.3.8 Objective Six - Future Exploration

- Explore opportunities for further improvement of the electronic voting system based on new technological developments and research insights. This includes staying updated with the latest research in blockchain technology and e-voting systems and incorporating relevant findings into future versions of the system.

1.4 Scope

The development of E-voting application's scope includes:

- Transparent election results reporting
- Secure electronic voting for qualified voters
- Strong data protection and process integrity safeguards
- Scalability to handle a high volume of users during elections
- User support and accessibility features
- Comprehensive end-user documentation
- Verification of national identification numbers
- Ensuring voter eligibility
- Preventing multiple or underage voting
- Preventing voting intimidation and fraud
- Simplifying voter registration procedures
- Improving voter education and information dissemination

1.5 Client Information

The clients for this application are individuals and groups participating in the voting process. The main end user groups include government and semi government institutions, private organizations, election officials, administrators, legal authorities and regulatory bodies, candidates, political parties, and voters.

1.6 Nature of Challenges

Being an academic research project in nature, all the tasks involved in the completion of this project need to be performed solely by myself, which is very challenging. From literature review to implementation and testing of the application, I must perform the tasks as well as to take all the roles and responsibilities such as project manager, designer, developer, analyst, tester, and researcher. Also, the time constraint is another challenge because the nature and scope of this project is difficult to achieve in the provided time. Testing the application in real world scenario is also not possible for this project thus leaving a validity gap in the project. Risk management is also difficult to tackle due to limitations of time, budget, and team resources.

1.7 Summary

This report will be divided into various chapters, each of which will give a summary of its contents:

- **Literature Review:** The research is summarized in Chapter 2, which includes information from academic sources on topics including blockchain, electronic voting, a comparison of traditional and electronic voting systems, the potential revolutionization of electronic voting through blockchain, and related topics.
- **Product Research:** Chapter 3 consists of a comparison of similar systems and their features and limitations.
- **Legal, Social, Ethical, and Professional Issues:** Legal, social, and environmental issues are discussed in Chapter 4 and how they may affect the project.
- **Development Methods and Tools:** Chapter 5 consists of development methods and tools where a comparison of different methodology followed by recommendation and justification.
- **Planning:** Chapter 6 consists of project planning and management planning
- **Feasibility Study:** Chapter 7 consists of feasibility study that contains technical, operational, and economic feasibility
- **Requirements Analysis & Specifications:** Chapter 8 consists of requirements analysis & specifications that contains rich diagram of traditional voting system, organizational structure, project stakeholder, and eliciting requirements
- **New system design and Architecture:** Chapter 9 consists of use case, ERD, architecture, class, and component diagram of e-voting system
- **Review of Technology:** Chapter 10 consists of all technologies details that are involved to develop the e-voting system
- **Development:** Chapter 11 consist of new developed system's modules
- **Testing:** Chapter 12 consists of all testing methods applied on the developed system

- **Deployment:** Chapter 13 consists of deployment criteria and training of users on specific modules
- **Evaluation:** Chapter 14 consists of product evaluation with non-functional requirements
- **Critical Appraisal:** Chapter 15 consists of Objectives met, objectives that partially met, and Objectives totally not met and son on.
- **Lessons Learned:** Chapter 16 consists of personal experiences felt during development of e-voting system
- **Conclusion:** Chapter 17 consists of document conclusion that contains project summary, project goals, achieved and unachieved goals, and personal experience
- **Reference:** Chapter 18 consists of all references of articles, website that helped to get data and information about e-voting system.

2 Literature Review

2.1 Introduction

The interaction of blockchain technology with electronic voting (or "e-voting") is thoroughly explored in this chapter. Our study begins with precise definitions of "Blockchain" and "E-voting" before contrasting e-voting with conventional voting procedures. The system requirements for blockchain-based electronic voting are also covered in detail, along with an evaluation of how well it might revolutionize the electoral process and a discussion of its substantial limitations and difficulties.

2.2 Strategy for literature Review

A systematic and all-encompassing approach is used when seeking literature on the crucial topics of "E-voting system," which include precise definitions of "Blockchain" and "E-voting," the interaction between blockchain technology and electronic voting (or "e-voting"), challenges, and blockchain requirements. The first step will be to do a thorough search of academic databases like IEEE Xplore, ACM Digital Library, and Google Scholar using key phrases like "Blockchain and E-voting," "Electronic Voting System," "Blockchain Definitions," and "Challenges in Blockchain-Based E-voting." These initial searches will be conducted with the goal of locating important books, academic papers, and research studies that offer fundamental definitions and insights into blockchain and electronic voting.

We will then perform a snowball search by looking at the citations and references in these found papers. This will make it possible for us to comprehend the historical context of ideas and hypotheses surrounding how blockchain technology and electronic voting interact. We will concentrate on literature addressing security issues, scalability issues, regulatory frameworks, and inclusion aspects to obtain insights into the unique difficulties offered by blockchain-based electronic voting. We will search through important academic publications, conference proceedings, and reports from recognized organizations for pertinent information.

2.3 Identifying a problem

A defective electoral system contributes significantly to unhealthy political competition among those aspiring for power, which frequently results in electoral violence. Several issues have persisted throughout the election season. Notable among these issues are the absence of some registered voters' names, intimidation against voters and exclusion, multiple and under-aged voting, ballot box robbery or destruction, and errors and manipulation in result calculation (Ahmad at al. 2016). The traditional electoral system has served as a stimulus for election-related violence, which has serious effects, such as eroding public trust and confidence in the democratic process. This violence is frequently caused by manipulative politicians and certain biased electoral officials who perpetrate electoral fraud by exploiting the vulnerabilities of the traditional voting system (Anjan et al. 2019).

Therefore, the demand for improved security, transparency, and trust in election processes is addressed by developing an e-voting system employing blockchain technology, which solves a significant issue present in traditional voting systems. Voter fraud, data manipulation, and disagreements over election results are just a few of the problems traditional voting systems have long struggled with. An e-voting system can create an incorruptible log of votes by utilizing the immutability, transparency, and decentralized consensus capabilities of blockchain, effectively eradicating the potential of tampering or manipulation (Gibson et al. 2016).

Additionally, blockchain's cryptographic protections can guarantee voter privacy while making it simple to confirm each vote's legitimacy. This invention aims to enhance the basis of democracy itself while also enhancing the credibility of elections and public confidence in the democratic system (Zhou et al. 2020).

2.4 What is Blockchain

Blockchain technology has gained a lot of attention in the modern software business ever since the launch and widespread acceptance of Bitcoin, the original cryptocurrency, in people's daily lives. When blockchain technology was originally brought to the internet, it immediately acquired popularity because to its high level of system transparency and became an active topic of research and study for its possible applications in a variety of other sectors. The core architectural structure of the cryptocurrency bitcoin is where blockchain technology originates Abhishek et al (2020).

As the name shows, a blockchain is a chain of interconnected blocks, every block is connected to the next via encryption. Two cryptographic keys—a private key and a public key—make up a blockchain. These secrets enable two parties to conduct successful transactions. Two keys are assigned to each individual and are used to develop a secure identity reference. This identity is used to control transactions and is known as a digital signature Kanika Garg et al (2019). Each block consists of transaction information, a timestamp, and the previous block's cryptographic hash value. Using distributed ledger technology, blockchain manages a peer-to-peer network that facilitates node communication and the validation of new blocks Aishwarya et al (2020). The blockchain’s structure is presented in Figure 2.1:

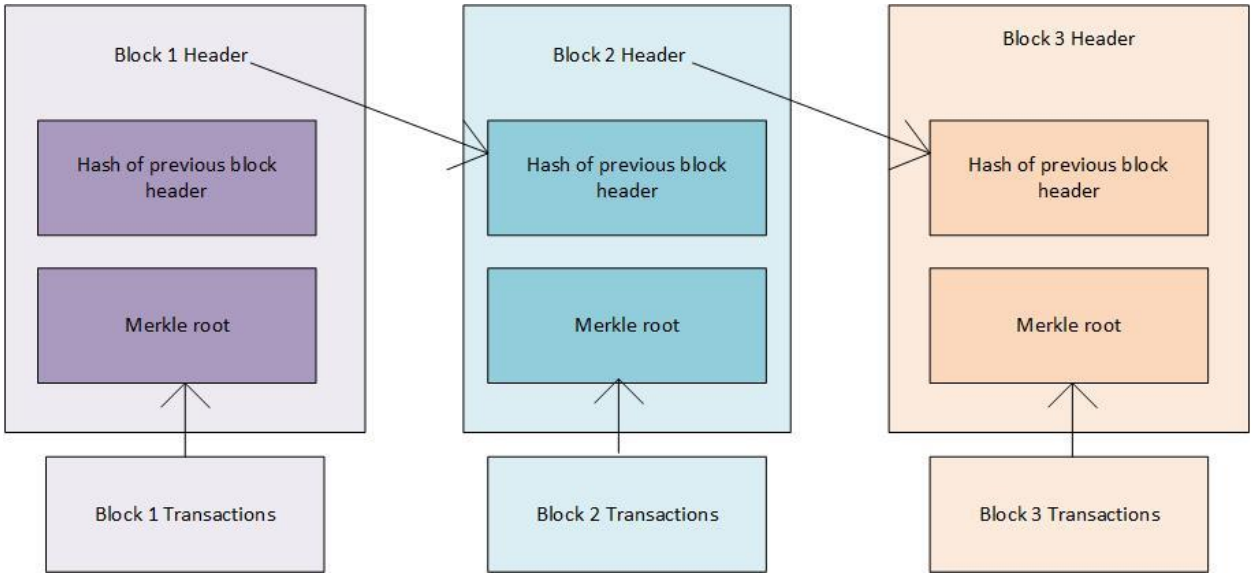


Figure 2.1Blockchain Structure

- Blocks: As illustrated in Figure 1, each chain comprises numerous blocks, with each block housing a collection of transactions that undergo hashing and encoding into a Merkle tree structure. A cryptographic hash of the block before it can be found in each block, creating a linked and chain-like structure. Each block incorporates elements such as digital signatures, timestamps, and other pertinent data A. G. Malvik et al (2016). The block is transmitted across all networks and, when the correct person uses his private key and matches it with the block using the block, the transaction is successful. It should be highlighted that the block does not contain individual identities. The three basic parts of each block are the data in a block, a 32-bit whole integer called the nonce, and a 256-bit number called the hash.
- Miners: The chain of blocks is expanded by miners because of the mining process. In a blockchain, each block has a unique nonce and hash, but it also has a relationship to the hash of the block that came before it. Therefore, mining a block could be challenging, particularly on long chains. It takes specialized software to solve the difficult arithmetic problem of finding a nonce that yields a hash. Finding the right combination requires almost 4 billion iterations because the nonce is just 32 bits long and the hash is 256 bits. All network nodes accept a change when a block is successfully mined
- Nodes: The chain cannot be owned by a single machine using blockchain technology. Instead, it is spread through nodes connected by a chain. A node is any form of device that stores copies and keeps a network going Abhishek et al (2020).

2.5 Key Elements of Blockchain Structure

Ass presented in the Figure 2.2; These are the main key elements of the Blockchain.

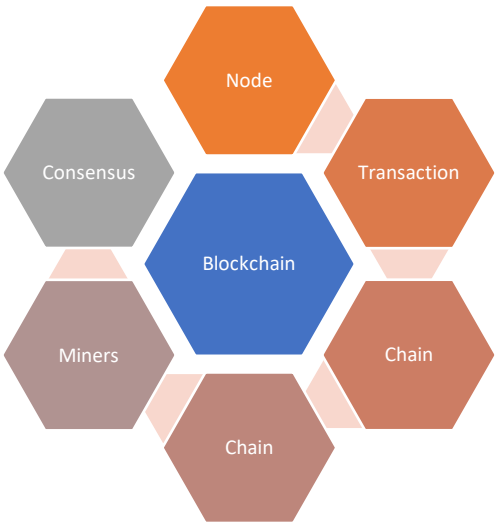


Figure 2.2Key Elements of Blockchain Architecture

2.6 Vital Attributes of Blockchain Structure

The many industries that use blockchain technology can benefit from the blockchain architecture. Figure 2.3 shows a variety of inherent traits, including the following:

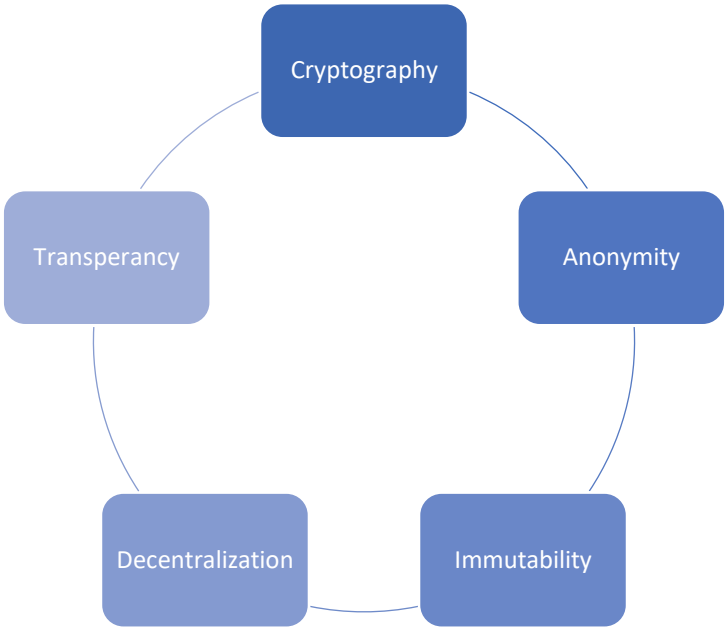


Figure 2.3Attributes of Blockchain Structure

1 Cryptography: To ensure accuracy between parties, blockchain transactions are validated and secured using cryptographic calculations.

2 Immutability: The blockchain offers a high level of data integrity because documents within it cannot be changed or removed.

3 Decentralization: The entire distributed database is accessible to all users of the blockchain network. A consensus mechanism is employed to take command of the system, as seen in the core procedure.

4 Anonymity: In public blockchain systems, players are recognized by addresses rather than user IDs, preserving anonymity.

5 Transparency: The tremendous computational resources needed to update the blockchain network make it resistant to manipulation, preserving data integrity Usama Jaffar et al (2021).

2.7 Electronic Voting (E-Voting)

Electronic voting, also referred to as "e-voting," is the casting and counting of ballots using electronic methods. E-voting can occur in a variety of ways, such as online voting on computers connected to the Internet or using freestanding electronic voting machines (EVMs). It consists of a range of Internet services, from providing tabulated results to permitting fully functional online voting with the use of common household electronics. The degree of automation varies, ranging from the simple electronic marking of a paper ballot to a whole system that incorporates vote input, recording, encryption, transfer to servers, and the consolidation and tabulation of election results J Paul et al (2016).

2.8 How Blockchain can revolutionize the E-voting system

By making voting open and accessible, reducing fraud, enhancing data security, and certifying election results, blockchain technology has addressed the shortcomings in the current electronic system. The application of blockchain technology in electronic voting is essential. Oliver et al (1996).

However, there are significant concerns associated with electronic voting. All cast votes may be modified and used against the voter if an electronic voting system is compromised. Despite its potential advantages, electronic voting has not yet been widely used on a national level due to these possible flaws. Blockchain technology currently offers a workable approach to reduce these dangers related to electronic voting Prashar et al. (2020).

Figure 2.4 illustrates the primary distinction between the two systems. In traditional voting systems, there exists a central authority responsible for casting votes. If someone wishes to alter or tamper with the record, they can do so easily, and there is no foolproof way to verify that record. Conversely, in a decentralized system, there is no central authority, and data is distributed across multiple nodes. Hacking all these nodes and altering the data is virtually impossible. Consequently, this approach safeguards against the manipulation or destruction of votes, ensuring the integrity of the voting process through cross-verification with other nodes.

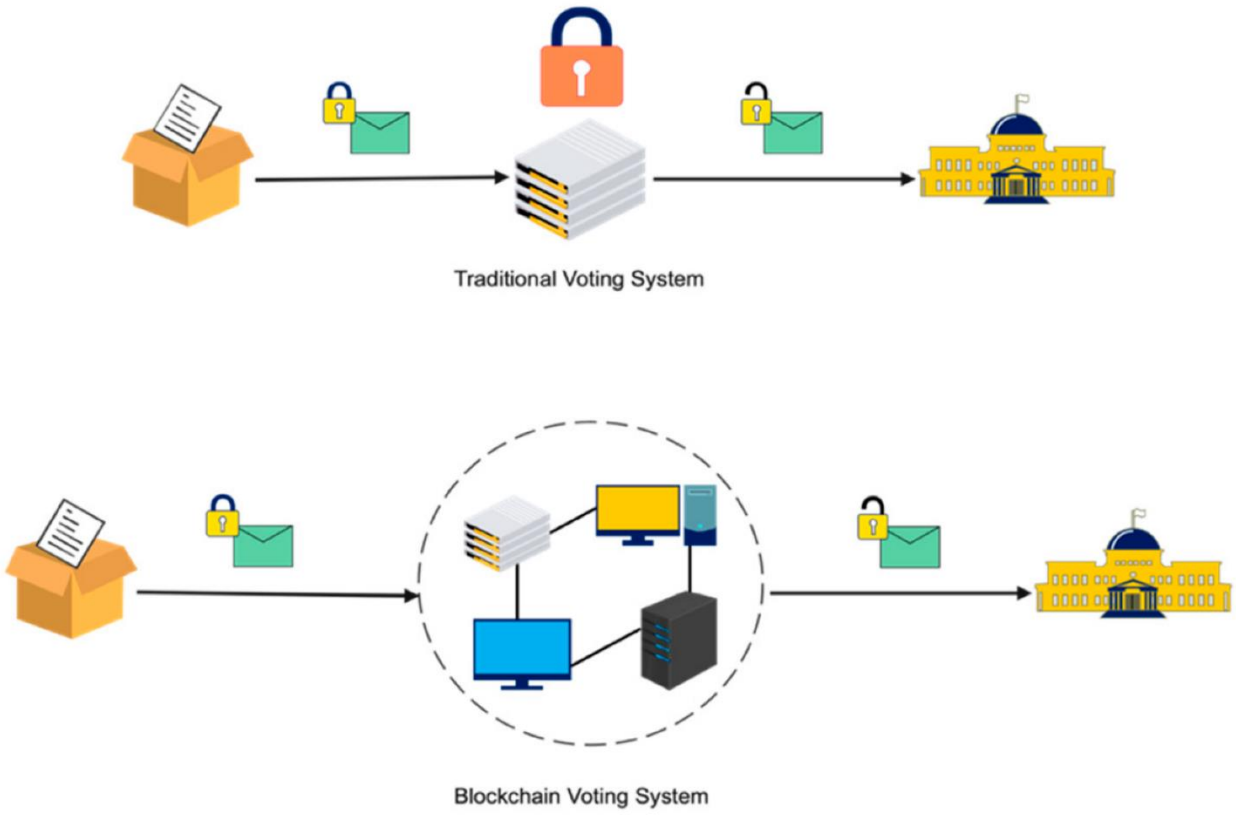


Figure 2.4Traditional VS E-voting Blockchain Voting System

When used properly, blockchain technology acts as a digital, decentralized, transparent, encrypted, and highly resistant to fraud and manipulation ledger. A Bitcoin e-voting system considerably reduces the vulnerability connected with e-voting and maintains the safeguard integrity of the voting steps because of its global nature. A blockchain-based electronic voting system requires a completely decentralized voting infrastructure to operate. Only when the internet voting process is independent from all other entities, including the government, can such a system be effective Imperial **(**M. 2021).

In conclusion, the integrity of elections relies on a widespread belief in the legitimacy of those in evidence. The assessment of the literature in this area and associated experiments offer a promising route for improving the administration and participation of the voting process. However, the development of blockchain has made a fresh approach to electronic voting possible.

2.9 Requirements for Voting System Security

The following criteria for electronic voting should be satisfied by appropriate electronic voting systems. The primary security criteria for electronic voting systems are shown in Figure 2.5.

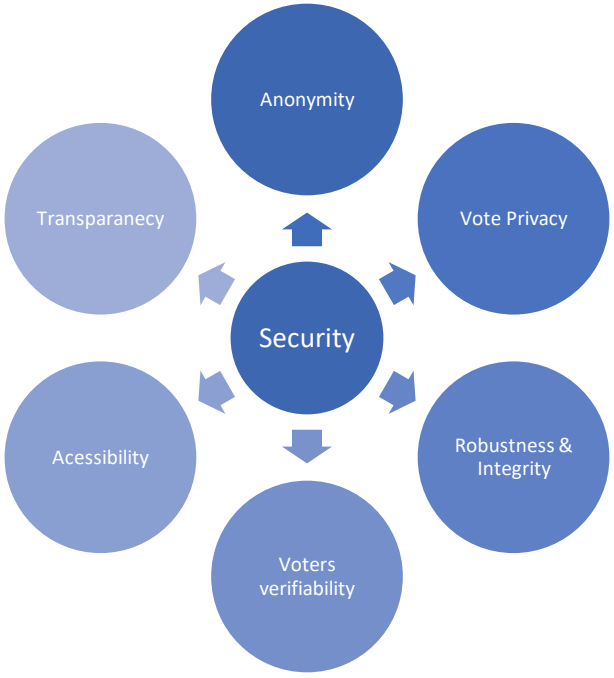


Figure 2.5 Requirements for Voting system Security

2.9.1 Anonymity

It is essential to protect voter turnout data from outside interpretation throughout the whole voting process. Within the voting system, there shouldn't be any disclosure of any connection between registered voters and voter identities Ma et al (2019).

2.9.2 Vote Privacy

The system makes use of the cryptographic capabilities of the blockchain to safeguard a voter's privacy. More specifically, as soon as a voter registers with the system, the blockchain generates a voter hash. The collision resistance quality of the cryptographic hash serves as the voter's unique identifier within the blockchain and protects it from misuse. As a result, voting is also non-trivially traceable, shielding the voter against coercion Augoye et al (2019).

2.9.3 Durability and Integrity

This criterion stops a substantial group of electors or representatives from interfering with the election. It ensures that voters who are duly registered will either abstain peacefully or influence others to cast their own genuine votes. Contesting election results on the grounds that one of the candidates did not fully fulfil their obligation is illegal for both citizens and officials to do Wei et al (2020).

2.9.4 Fairness and Openness

Prior to the revelation of the count, no one will be able to find out the specifics. It refrains from doing things like using predictions to influence the decisions of late voters or giving unfair advantages to individuals or organizations to be the first to know Poniszewska et al (2020).

2.9.5 Voters Verifiability

After correctly casting their vote, a user receives their unique transaction ID in the form of a cryptographic hash. A user can check the transaction ID to see if their vote was counted. This approach was developed to decrease threats when acting under duress, but also prevents users from viewing their votes.

The research that was done before shows how well the offered solution meets the requirements for electronic voting. Additionally, it highlights the significance of blockchain's distinguishing characteristics and their critical role in laying the groundwork for a successful e-voting system. Since blockchain technology can be used to establish a secure digital voting system, we believe the study presented here considerably expands on what is already known about this topic. Khan et al (2019).

2.9.5.1 Accessibility and Reassurance

It must ensure that every voter can get to the appropriate polling station, and that location must be open and simple for the voter to access. Only qualified voters should be allowed to participate in elections, and all ballots must be carefully tallied to ensure that elections are valid Dagher et al (2018).

2.10 Discussion on Problem Domain and Possible Solutions

2.10.1 Initial Discussion

Although it is not the only one, secure digital identity management is one of the most current and important technical challenges related to electronic voting systems. All prospective citizens must register with the electoral system prior to the elections. Their data needs to be very simple to process digitally. Additionally, any content pertaining to them must keep their identity fully a secret.

2.10.2 Problems in Current Systems and Their Solutions

The following are the drawbacks of an antiquated electronic voting system:

2.10.2.1 Casting a ballot without a photo ID

Allowing voters to cast ballots without being asked for a photo ID raises the risk of identity fraud and impersonation, which might compromise the integrity of the political process.

Solution: Establish stringent voter identification procedures that require voters to show a legitimate photo ID or submit to biometric verification procedures before casting a ballot. Make sure that everyone who is eligible to vote can understand and access these requirements.

2.10.2.2 Individualized voting processes

Each voter's particular, personalized voting procedure can complicate the overall voting process, making it difficult for election administrators to uphold consistency and prevent mistakes.

Solution: Create a uniform and user-friendly electronic voting interface that takes voters step-by-step through the procedure and is accessible to people with different technological backgrounds. Provide support for individuals who require it, such as language translation or voting options that are accessible to people with impairments.

2.10.2.3 The ballot's ability to be verified by the voter (and only the voter)

Maintaining privacy and security requires making sure that only the voter can validate their ballot; otherwise, there is a higher chance of coercion or vote-buying.

Solution: Rely on blockchain technology or another tamper-evident method to issue each voter a distinct, verifiable digital receipt or confirmation code after they cast their ballot. This preserves the identity of the voter while enabling them to confirm that their vote was appropriately recorded.

2.10.2.4 Astronomical setup costs

Older electronic voting systems frequently need expensive hardware and infrastructural upgrades, rendering them financially unviable.

Solution: Invest in the creation of an extensible, economical e-voting infrastructure that can be applied to numerous elections. To lower licensing and development costs, look into open-source options. Collaborate with public and private institutions to split infrastructure costs.

2.10.2.5 Expanding security concerns

Older voting systems are more susceptible to cyberattacks and hacking as technology develops, raising the possibility of election results and voter data being compromised.

Solution: Establish a thorough cybersecurity system that consists of encryption, multi-factor authentication, frequent security audits, and constant monitoring. Engage cybersecurity professionals to proactively find and fix any potential weaknesses.

2.10.2.6 A lack of confidence and transparency

Outdated systems might not provide the openness and auditability required to foster confidence in the electoral process, which could lead to decreased voter confidence and disagreements over election results.

Solution: Remedy Promote openness by increasing public access to all aspects of the voting process, including voter registration, candidate nomination, and vote tallying. Accessible real-time election information and outcomes. Use blockchain or comparable technology to guarantee the process's integrity.

2.10.2.7 Voting irregularities or delays resulting from remote or absentee voting

Voting delays or irregularities brought on by remote or absentee voting: Remote or absentee voting may provide difficulties with regard to the timely and accurate processing of votes, which may result in delays, irregularities, or questions about the fairness of the election (Lahane et al, 2020).

Solution: While maintaining strict security measures, implement user-friendly and secure remote voting options, such as online or mobile voting. Make sure that absentee voting procedures are simplified and that votes are collected and counted promptly. Utilize blockchain technology to safely manage distant voting.

2.10.3 Recommended Approach

The optimal approach to address the identified problems is to:

- Implement stringent voter identification requirements.
- Standardize and simplify the voting interface for user-friendliness.
- Incorporate blockchain technology for secure verification.
- Ensure cost-effective infrastructure for reliability.
- Prioritize cybersecurity with encryption and multi-factor authentication.
- Implement ongoing monitoring for security.
- Offer secure remote voting options.
- Streamline absentee voting procedures.
- Use blockchain to enhance security and efficiency.
- Reduce vote anomalies and delays associated with remote or absentee voting.
- Conduct public education efforts to build trust in the improved voting system.
- Ensure accessibility for all eligible voters.

3 Product Research

3.1 Review

This cutting-edge e-voting system was created in response to the drawbacks of conventional electronic voting systems to improve the validity, usability, and security of the electoral process. There are variety of e-voting systems available in the market, but each system comes with certain limitations. This chapter discusses the most popular available e-voting systems in terms of their usability, security, robustness,

reliability, transparency, and availability. The discussed e-voting platforms range in terms of these attributes. For example, the i-Voting system from Election runner out for its strong blockchain-based security, usability, and transparency, making it a good option for conducting digital elections. Swiss Post e-voting is safe and easy to use, but because of its limited availability, it is not widely used. Scytl's eVoting provides a reliable choice with an emphasis on security and robustness and a global presence. Despite controversy and security worries, Voatz strives to make voting accessible through mobile devices. For many jurisdictions, Smartmatic offers a dependable and transparent solution. To preserve security and confidence in the quickly changing environment, any e-voting system must undergo ongoing review and improvements. Our system uses a complex strategy to tackle major issues such voter identification, customized voting experiences, ballot verification, setup costs, security worries, transparency, and voting anomalies. We ensure that voters may confidently cast their votes by adopting strict voter identification mechanisms and a user-friendly interface, while blockchain technology ensures the verifiability of each vote without compromising anonymity. Additionally, this system prioritizes cybersecurity with encryption and constant monitoring while reducing setup costs through scalability and open-source technologies. Transparency measures are incorporated into every aspect of the voting process to foster trust.

3.1.1 Electionrunner

A platform called ElectionBuddy is committed to advancing open democracies all over the world. Within member-based organizations, their team of Election Experts focuses on creating and delivering user-friendly, safe, and affordable voting solutions for both voters and voting administrators. Three key service categories are available from ElectionBuddy: Elections, Meeting Votes, and Additional Services. They make a variety of elections possible while ensuring affordability, accessibility, and user-friendliness. Additionally, they offer meeting vote services that simplify voting procedures at corporate events. In order to improve their clients' election and voting experiences, they may also offer customization, assistance and training, integration, and consultancy. ElectionBuddy's overall goal is to enable organizations to hold free, secure, and effective elections while expanding access to democracy globally.

Website URL: https://app.electionrunner.com/login

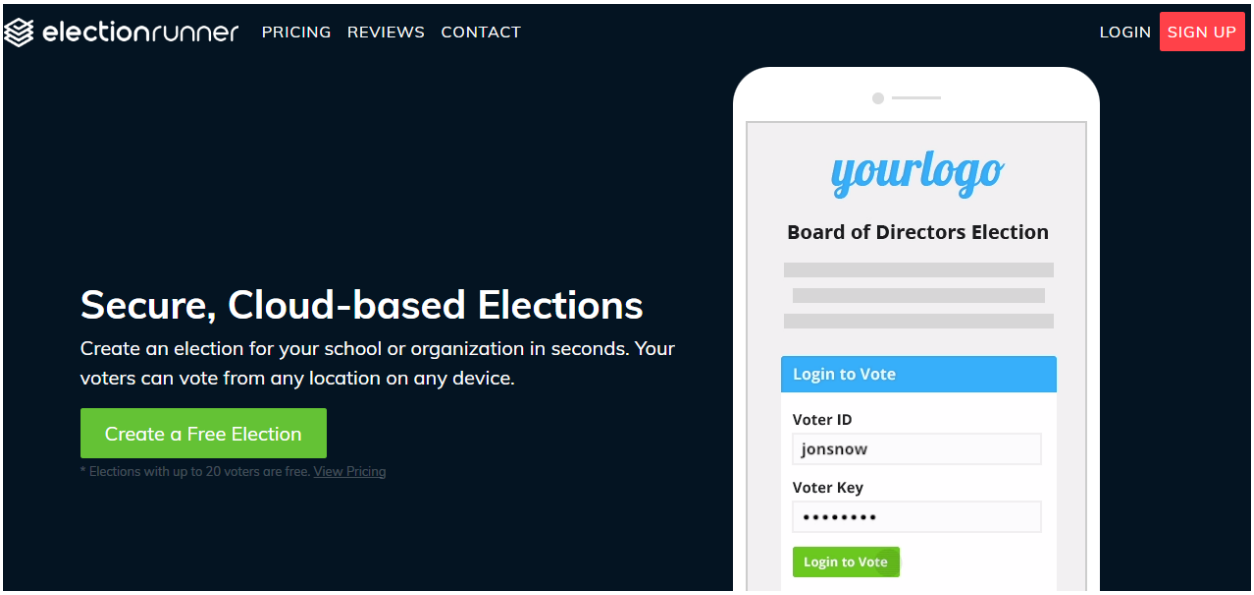


Figure 3.1 Electionrunner Homepage

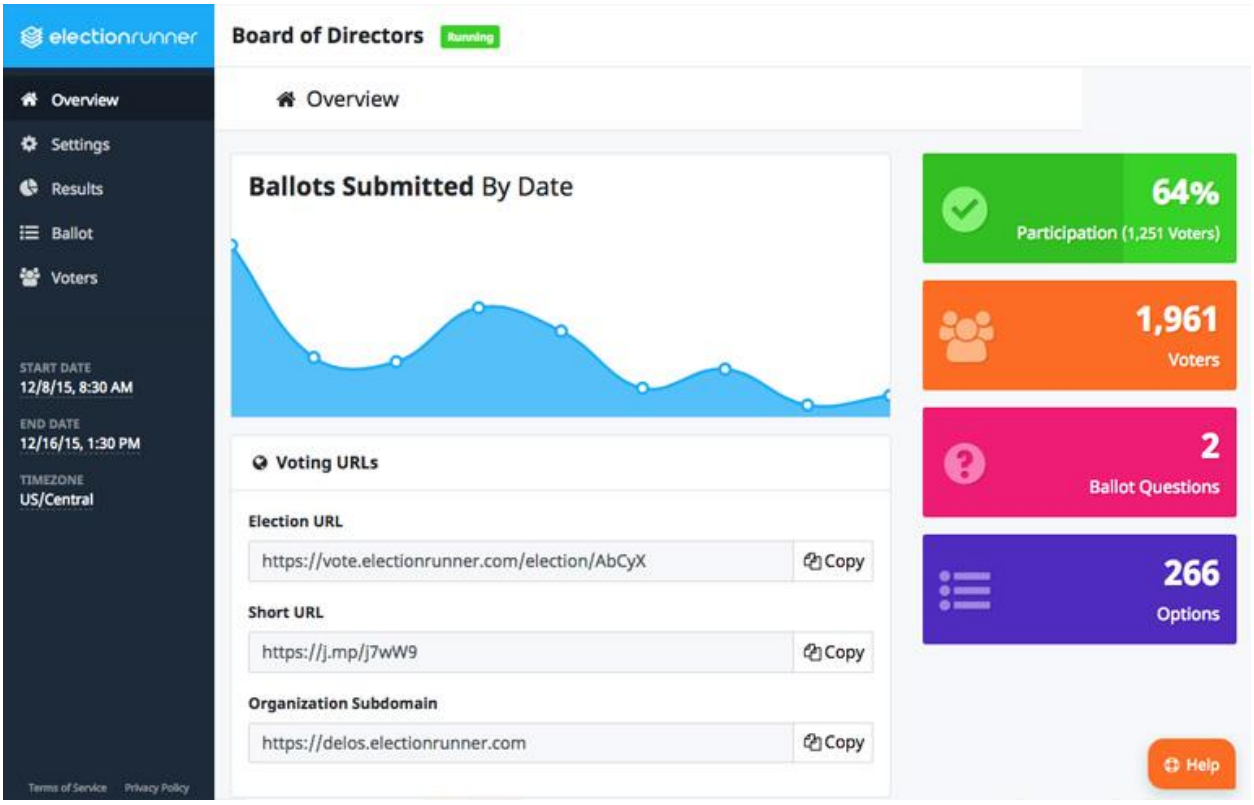


Figure 3.2 Electionrunner Overview

Best Features:

- Voter Registration

- Authentication
- Voting Interface
- Ballot Verification
- Multi-lingual Support
- Real time reporting
- Voter notification
- SMS & email notifications

Limitations

The ElectionRunner application has the following limitations:

- Limited Support for Offline Voting
- Possibly Vulnerabilities in Security
- Administrators' Learning Curve
- The Scalability of Large Elections is Limited
- Reliance on Internet Access
- Limitations on Older Device Compatibility
- Privacy concerns for voters
- Limited Options for In-Person Voting
- Possibility of Problems with Voter Authentication
- Limited Accessibility for Disabled Voter

3.1.2 Swiss Post e-voting

The Swiss Post e-voting system is distinguished by its emphasis on security, which is achieved by using cutting-edge cryptographic and encryption methods to guarantee the greatest levels of data security. With a user-friendly interface and accessibility features created to suit a wide spectrum of voters, it also prioritizes user-friendliness. In areas where it is available, the system's robustness is regularly assessed and upgraded, which increases its dependability. One drawback, though, is that it's only now available in a few cantons and regions of Switzerland. It is regarded as being less transparent than certain other e-voting methods in comparison. However, it essentially acts as a substitute for the traditional voting process for qualified Swiss voters, providing a safe and effective way to take part in elections.

Website URL: https://digital-solutions.post.ch/en/e-government/digitization-solutions/e-voting

Best Features:

- User friendly interface

- Security
- Authentication
- Robustness

Limitations:

- Limited availability
- Less transparency
- Security concerns

3.1.3 Scytl eVoting

The Scytl e-voting system under consideration places a strong focus on security, emphasizing end-to-end encryption and secure voting mechanisms to safeguard the integrity of the voting process. It boasts user-friendly interfaces for both voters and election authorities, ensuring accessibility and ease of use. Demonstrating robustness through its participation in numerous elections worldwide, this system has established itself as a reliable choice, subject to adoption in many nations. Its proven dependability and transparency, facilitated by verification processes and audit trails, contribute to its effectiveness in providing secure and efficient voting options while expediting election procedures.

Website URL: https://scytl.com

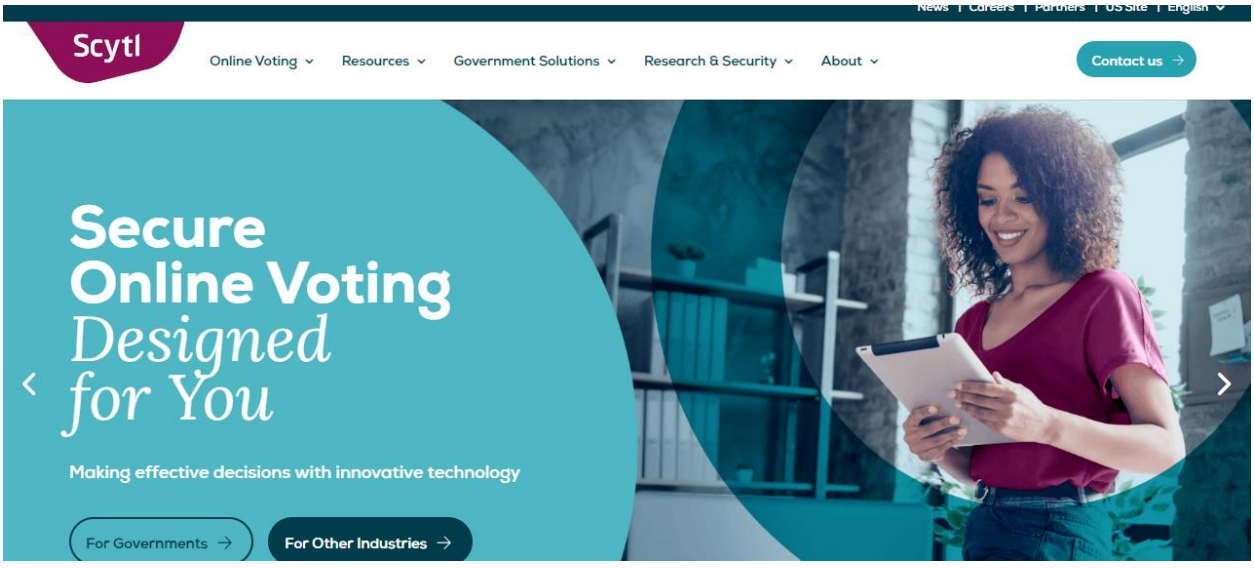


Figure 3.3 Scytl Homepage

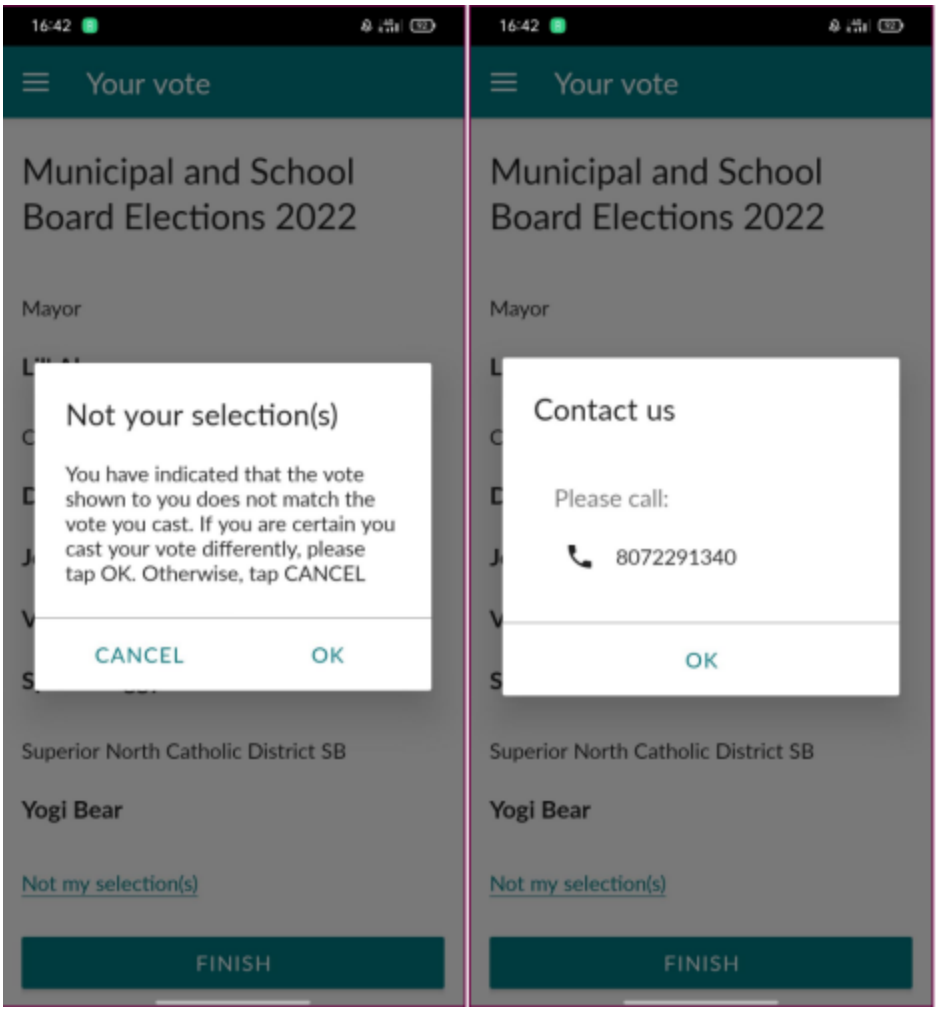


Figure 3.4 Vote Screen

Best Features:

- Authentication
- Ballot Design
- Election Scheduling
- Candidate management
- Election monitoring

Limitations

- Security concerns
- Trust and Skepticism
- Limited adoptions

3.1.4 Voatz

Voatz, a mobile voting app, has faced both scrutiny and controversy over its security measures. To maintain the security of the voting process, the organization places a high value on blockchain technology. In terms of usability, Voatz attempts to provide an accessible mobile voting app. However, the system's robustness has been a source of contention, with security worries. It is only available in a few areas, limiting its widespread use. Voatz's dependability has been rated variously, with varied assessments due in part to security concerns. Transparency is a concern, as audit accessibility and overall transparency have been called into doubt. Due to the platform's issues, questions have also been raised about its usefulness and popular adoption.

Website URL: https://voatz.com

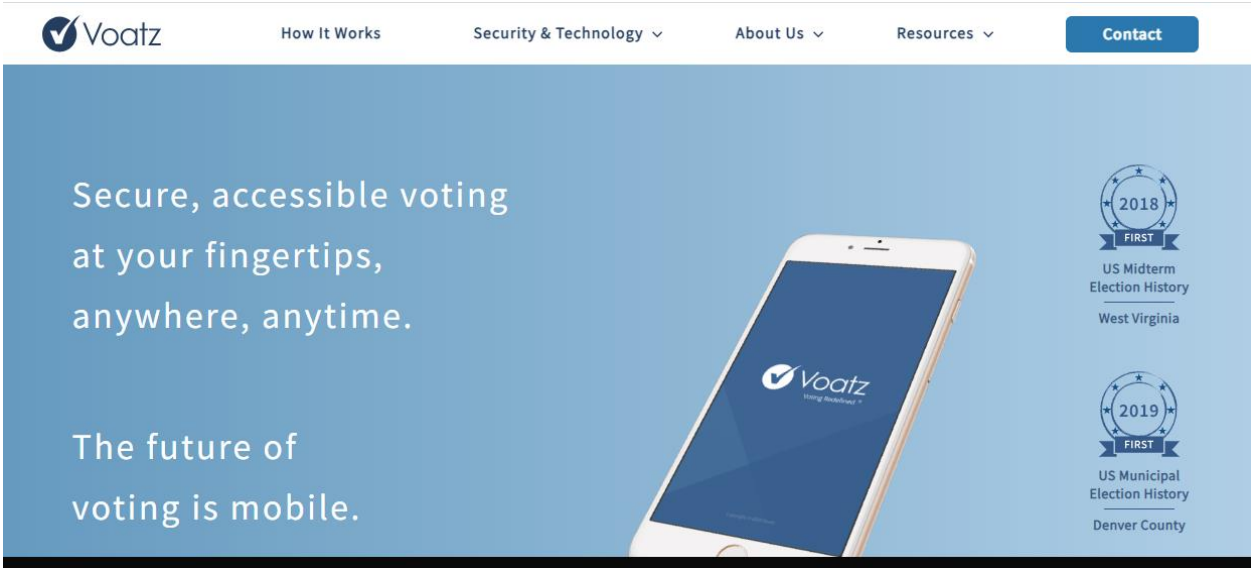


Figure 3.5 Voatz Homepage

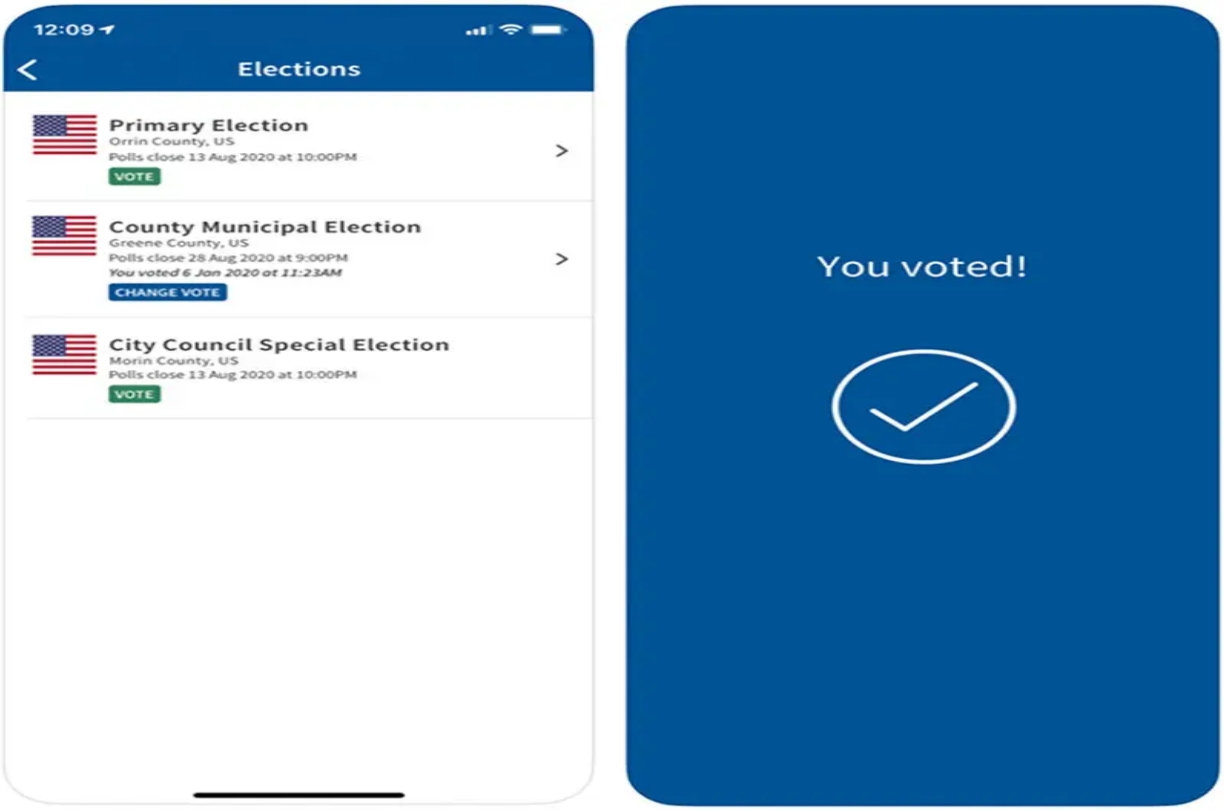


Figure 3.6 List of elections

Best Features:

- Identity verification
- Real time Result reporting
- Multi language
- Voter notifications

Limitations

- Limited availability
- Dependability
- Transparency and Auditability
- Privacy concerns

3.1.5 Smartmatic

The Smartmatic e-voting system in question prioritizes security, with a focus on audits and secure encryption mechanisms to verify the voting process's integrity. It provides intuitive user interfaces for both administrators and voters, improving usability. Demonstrating its robustness, the system has been

successfully used in elections across multiple nations. With availability in numerous countries where it has been adopted, it has proven to be a reliable choice for digital voting. The system's overall trustworthiness is supported by its usage in various elections, and it offers transparency through verification tools and comprehensive audit records. In terms of effectiveness, it provides a secure and efficient way to conduct elections and expedite the voting procedure.

Website URL: https://www.smartmatic.com

Figure 3.7 Smartmatic Homepage

Best Features:

- Election Continuity
- Manual Elections
- Voter Management
- Poll Worker Support
- e-Voting
- e-Counting
- Remote Voting
- Election Management Platform
- Project Management & Services
- Vote Anywhere
- Mobile Application

Limitations:

- Complexity
- Privacy Concerns
- Dependability

3.2 Model for Weighted Scoring

Table 1 Model for Weighted Scoring

Criteria	Weight	ElectionRunner	Swiss Post	Scytl	Voatz	Smartmatic
Security	20%	18%	15%	17%	18%	18%
User Friendliness	15%	14%	10%	10%	10%	11%
Robustness	15%	12%	13%	11%	12%	13%
Availability	15%	13%	9%	8%	8%	8%
Transparency	20%	18%	17%	18%	18%	17%
Effectiveness	15%	12%	11%	11%	12%	11%
Total	100%	87%	75%	75%	78%	70%

To establish their total performance, we evaluated them using important criteria and weighed each category. Security, user-friendliness, robustness, availability, transparency, and efficacy are among the characteristics, with varied weights reflecting their relative importance. Swiss Post and ElectionRunner stand out as serious competitors after receiving high marks for security and transparency. They have gained a great deal of trust for their dependability and resilience and provide user-friendly interfaces. However, ElectionRunner has a little advantage because of a higher user-friendliness rating. Despite displaying robustness and efficacy, Scytl trails a little behind due to lower ratings in usability and accessibility. Despite emphasizing user-friendliness, Voatz has serious security issues that lower its overall rating. The effectiveness and robustness of Smartmatic, on the other hand, are behind, which has an impact on its total score.

3.3 Recommended Approach

- **Implement blockchain-based security** measures like those emphasized by Swiss Post and Voatz to prioritize vote integrity. This ensures that votes are securely recorded and cannot be tampered with.
- **User-Friendly Interface:** an intuitive and accessible user interface should be created like ElectionRunner's. This will allow voters with diverse degrees of computer literacy to participate, encouraging inclusivity.
- **Comprehensive Testing:** Like Scytl's approach, thoroughly test the system for dependability and robustness. This will entail assessing the system's performance under various scenarios to discover and address potential vulnerabilities.
- **Transparency and auditing:** Prioritize transparency through using Swiss Post and Scytl's transparency features. Implement comprehensive auditing measures and make election data publicly available to ensure an open and transparent process.
- **Ongoing Security Evaluations:** Using the Voatz model, continuously evaluate and upgrade the system's security protections. This will aid in dealing with evolving security threats and weaknesses.
- **Focus on accessibility features** comparable to those provided by ElectionRunner. This will ensure that all voters, including those with disabilities, can use the system.
- **Privacy Protection:** Include strong privacy safeguards, addressing concerns regarding voter data gathering and storage, as evidenced in the instance of Smartmatic.

The final e-voting system can maximize security, accessibility, transparency, and efficacy by integrating these best practices and resolving the specific shortcomings discovered in the analysed systems. It will serve as a model for holding secure and inclusive digital elections that suit the changing needs of modern democracy.

4 Legal, Social, Ethical, and Professional Issues

4.1 Professional Issues and Considerations

In the pursuit of this project, it is imperative to give due consideration to several professional considerations. Ensuring the project is conducted in a professional manner entails assessing the various implications it may encounter during development and when in use by real-world users and the public.

4.2 Discrimination

The developed product must adhere to principles of non-discrimination, refraining from supporting bias or prejudice towards any individual or social group based on attributes such as race, religion, ethnicity, nationality, gender, or any other defining characteristic. Furthermore, the product should abstain from employing any language, visuals, symbols, or content that might cause offense to any culture, group, or individual through any means or channels (Augoye et al. 2019).

4.3 Security and Data Protection

While executing this project, it is crucial to give utmost attention to security and data protection aspects. Given the product's nature, it will involve the storage and retrieval of user data for its proper functioning. Therefore, it is of paramount importance to ensure a high level of transparency is provided to the users. They should have a clear understanding of the type of data and information being stored, as well as the measures in place to secure it. Users must be explicitly informed that their data will not be utilized for purposes to which they have not given their consent.

Ensuring the protection of end-user data is of utmost importance in the development of this product. The system must be built to ensure that no private or sensitive information can be accessed without the user's explicit authorization (Khan et al. 2019).

4.4 Software Licensing

Various open-source hardware, software, and frameworks will be used in this project. According to the terms of the license agreements, it is essential to provide due credit to the software's developers while using these licenses. You can be sure that this project will fully adhere to the use of licensed and legal software (Augoye et al. 2019).

4.5 Environmental Impact

Adoption of e-voting systems utilizing blockchain technology, while promising increased efficiency and security in electoral processes, also raises concerns about its environmental impact. The computational power required for blockchain networks to validate transactions and secure data through complex cryptographic algorithms can be energy intensive. This energy consumption primarily stems from the

process of mining or validating new blocks on the blockchain, which often involves power-hungry hardware and data centers. This increased energy use, which comes primarily from fossil fuel sources, can increase carbon emissions, and worsen the environmental impact of blockchain-based electronic voting systems. Moreover, as these systems scale to accommodate larger populations or more frequent elections, the environmental consequences may become more pronounced.

Efforts are underway to develop eco-friendly consensus mechanisms and optimize energy usage within blockchain networks. These initiatives aim to mitigate the environmental impact of blockchain technology. However, it remains essential for stakeholders to carefully assess and manage the ecological consequences of implementing e-voting systems to ensure that the advantages of enhanced security and transparency are not outweighed by environmental drawbacks (Khan et al. 2019).

4.6 Legal Impact

The legal impact of the product encompasses adherence to a comprehensive framework of laws and regulations to ensure its lawful operation and safeguard the rights and interests of all stakeholders. The product falls under the jurisdiction of British and European law and commits to strict compliance with all relevant statutes and directives established by these authorities. In particular, the product includes a software component that processes user data, necessitating strict adherence to the Data Protection Act 1998 and the General Data Protection Regulation (GDPR).

Furthermore, the product recognizes its responsibility to consider and comply with other pertinent regulations and information authorities, including but not limited to sector-specific regulations and data protection authorities. The product will actively cooperate with requests and inquiries from these authorities wherever possible to maintain transparency, accountability, and the highest legal standards (Nittari and Khuman 2020).

By diligently adhering to the legal framework outlined by British and European authorities and demonstrating a commitment to data protection and regulatory compliance, the product aims to operate within the bounds of the law, fostering trust and legal integrity among users and stakeholders alike (Snyder et al. 2019).

4.7 Social Impact

Applications for electronic voting have a substantial and varied social influence. By improving voting accessibility, convenience, and inclusivity, these technologies have the potential to strengthen democratic processes. E-voting software enables voters to submit their ballots electronically, increasing voter turnout, especially among younger and more tech-savvy demographics, and making voting easier for people with mobility or transportation issues. E-voting can also ease the administrative strain on election officials, which could result in faster and more effective election results. To make sure that the societal impact is

beneficial and that the democratic process' integrity is maintained, issues like security, privacy, and the digital gap need to be carefully addressed (Snyder et al. 2019).

5 Development Method and Tools

There was a wise saying that goes, “If we start with a well-thought-out plan, half the job is done." This is also true for software development. A solid strategy acts as a guiding star, leading to successful outcomes. varied approaches have been developed over time in the field of software development to accommodate varied project parameters such as size, complexity, and deadlines. In the instance of designing and deploying the "E-voting" application, we'll first go over some of the software industry's standard procedures. Following that, we'll lay out a basic strategy for developing and implementing the 'E-voting' application.

5.1 Importance of Development Methodology

Adhering to a development framework or methodology is quite advantageous because it acts as an organized guide for producing a fully functional application. Without such a methodology, accomplishing successful development becomes extremely difficult. According to Hannah et al. (2019), 25% of all software projects fail due to either not following any development methodology or employing an incorrect one. This wastes a lot of time and frequently results in poorly developed initiatives. A development methodology focuses on the actions that occur during the development process, the development cycle itself, and the result. In essence, it aids in the effective use of critical tools, methodologies, and resources, as well as the well-structured management of the development team, ultimately leading to success.

5.2 Comparison of Various Methodologies

Several methods of development have arisen over time in the dynamic arena of the software business, each with its own set of principles and practices. The efficacy of these approaches varies greatly from one software project to the next, based on criteria such as the type of software, its size, and its inherent nature. In this section, we'll look deeper into these various techniques, throwing light on their distinguishing features and how they affect the development process. Understanding these approaches allows us to better tailor our approach to the unique objectives and problems of each software project, resulting in more successful and efficient solutions.

5.2.1 Waterfall Methodology

The waterfall model epitomizes the traditional approach to software development, as it is a well-established and simple methodology. This approach follows a linear and sequential path, finishing each development activity one by one without overlap, making it a well-organized and time-tested strategy in software engineering (Kramer 2018). Waterfall project management divides projects into phases based on tasks such as requirement analysis, design, implementation, testing, deployment, and maintenance. Each

step is completed before moving on to the next, resulting in a systematic and sequential approach to project development (Fowler and M 2004).

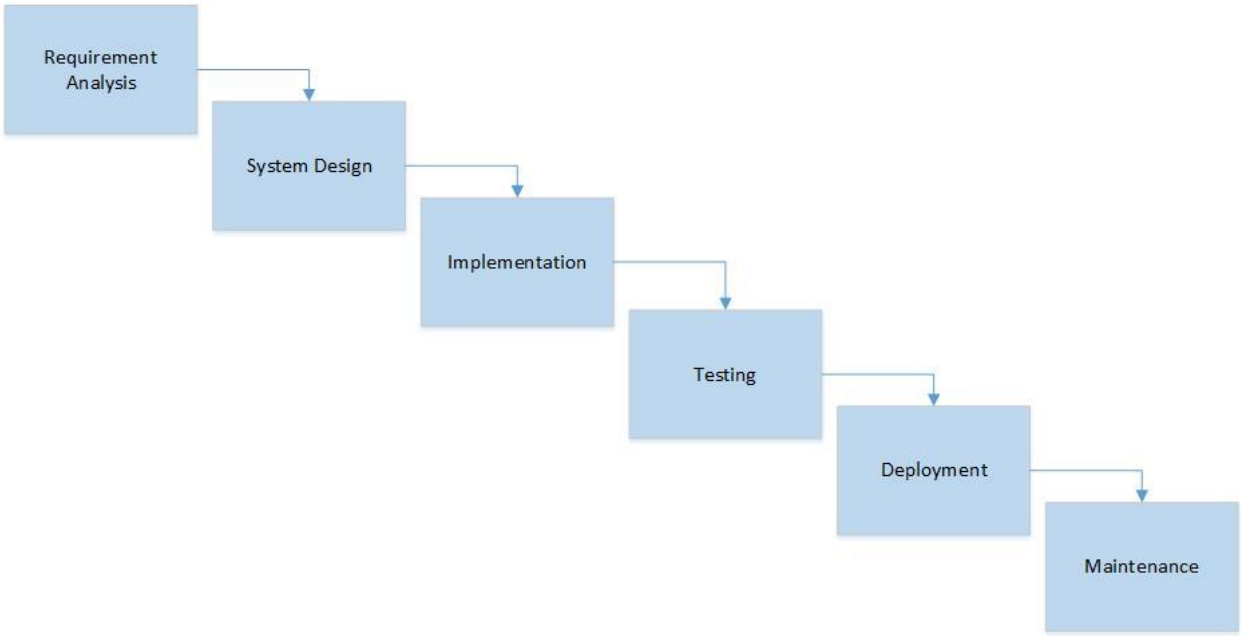


Figure 5.1 Waterfall Process Model

Every approach/methodology has benefits and drawbacks, and the waterfall method is no exception. Here, we'll go over the approach's Pros and Cons:

Pros:

- **Clear Project Planning:** Waterfall's sequential nature necessitates significant planning ahead of time. This can aid in the definition of defined project objectives, scope, and requirements, limiting the possibility of scope creep.
- **Stable Requirements:** Waterfall can be highly effective when needs are well-defined and unlikely to change dramatically during the project, as it does not accommodate frequent modifications.
- **Documented Progress:** Each step generates clear and thorough documentation, allowing stakeholders to follow progress and understand the project's status more easily (Kramer 2018).
- **Timeline and Budget Predictability:** Because the project has been thoroughly planned, the timeline and budget can be projected more reliably, providing stakeholders with a predictable roadmap
- Because of these interdependencies, each element is dependent on the completion of the previous one, making it simple for novice developers to understand (Adenowo, et al. 2013).

Cons:

- **Inflexibility to Change**: Waterfall is extremely sequential, and after a phase is completed, it is difficult and costly to go back and adjust. This can be an issue if needs change or mistakes are identified late in the project.
- **Limited Stakeholder Involvement:** Stakeholder feedback is frequently obtained late in the process, during testing or after deployment, which can result in misconceptions or unhappiness with the final product (Easan 2016).
- **Long Delivery Time:** Because Waterfall is a linear process, the final product is not supplied until all steps are completed, which can result in lengthy delivery delays, particularly for large and complicated projects.
- **Requirement Risk Uncertainty:** It presupposes that all requirements are known and can be described ahead of time, which may not be the case, resulting in scope modifications and project delays.
- **Difficulty in Handling complicated Projects:** Waterfall is less appropriate for complicated projects where needs are not well-defined or if rapid iterations and frequent adjustments are required (Kramer 2018).

5.2.2 Scrum Methodology

Scrum, invented by Jeff Sutherland in 1993, is a project management framework that adheres to the Agile methodology's principles. It is primarily intended as a responsive framework for software projects and to support the leadership of product or application creation. Unlike previous methodologies that take a step-by-step approach, Scrum emphasizes a more flexible and holistic approach to product development in which the development team collaborates as a cohesive unit to achieve common goals. Scrum is a multifaceted method that considers several elements that can influence the final output of a project (Permana and P.A.G 2015).

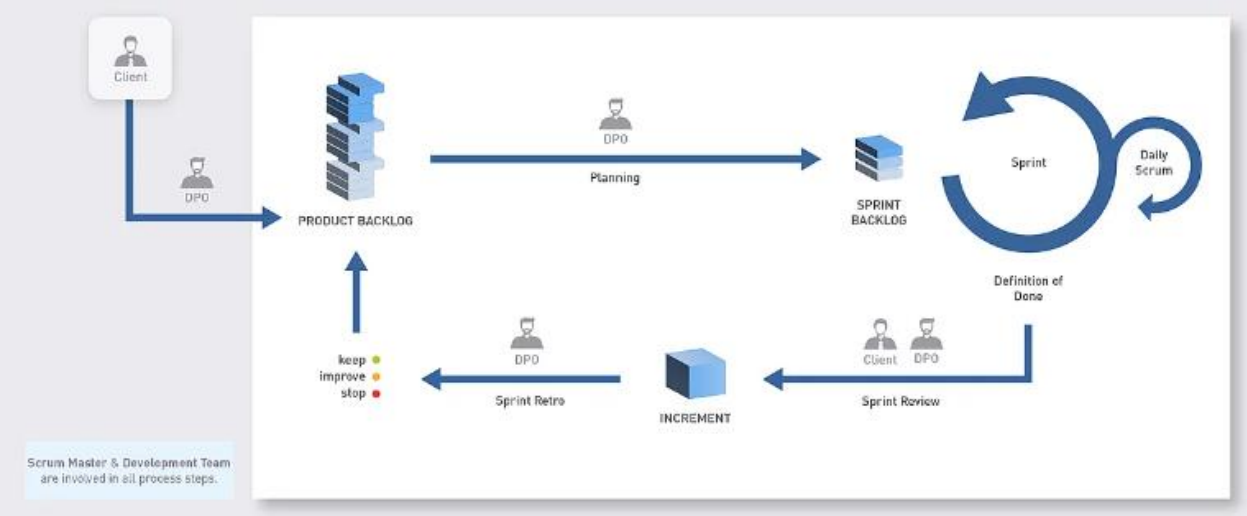


Figure 5.2 Scrum Model

Pros:

- **Simple to Implement:** Because it is a sequential model, it is simple to implement.
- **Minimal Resources:** The resources needed to implement this approach are modest.
- **Proper documentation** is used to ensure the quality of the development.
- **Customer-Centric:** Scrum emphasizes tight collaboration with stakeholders and regular feedback, ensuring that the product meets the needs and expectations of the customer.
- **Support for Change:** Scrum manages shifting requirements and priorities efficiently, making it ideal for projects with evolvable requirements (Mahalakshmi 2013).

Cons:

- **Inflexibility to Changes in Requirements:** If a client requests changes to the project requirements, they may not be accommodated or implemented due to the fixed scope established at the start.
- **Scope Freeze:** Projects that use this strategy typically have a fixed scope. This means that the customer's initial needs are considered contractual, and revisions are usually discouraged or difficult to implement.
- **Estimated project timetable:** The project schedule, including expected delivery time, is often estimated upfront. This might be troublesome if unanticipated problems or modifications develop during the project, causing delays or difficulties in fulfilling the initial dates (Mahalakshmi 2013)

5.2.3 Agile Methodology

Agile software development is a software development technique that embodies the ideals of being fast, portable, adaptable, and reactive. The term "Agile" refers to a break from standard software development

process paradigms, offering a fresh and innovative approach to project management. Kent Beck and a team of 16 colleagues pioneered the notion of Agile software development. They defined Agile Software Development as a process that entails actively producing software while also assisting others in doing so. This method emphasizes teamwork and hands-on learning (Permana and P.A.G 2015).

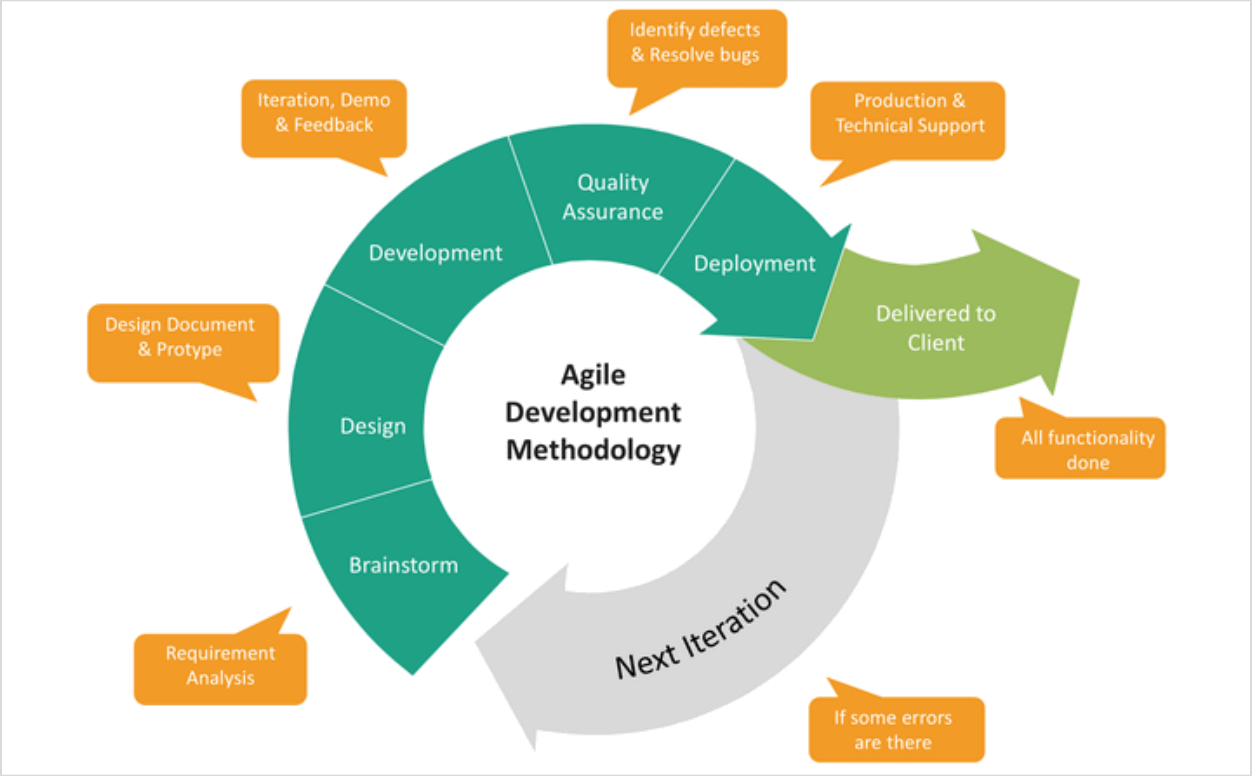


Figure 5.3 Agile Development Methodology

Pros:

- **Managing requirement changes**: Customers are actively involved in the development process during the planning phase, allowing them direct influence over project activities. This results in requirements that accurately represent end-users' current demands.
- **Early Fault Detection**: In contrast to plan-driven process models, testing is performed throughout each iteration, allowing for the early detection and rectification of faults before they become more severe. constant testing also gives constant feedback, which leads to continual improvement in succeeding revisions of code.
- **Enhanced Performance**: Daily standup meetings provide a forum for exchanging useful insights and continuously refining processes. Simplifying complex tasks with simple narrative and designs develops a sense of collaboration. As team members gain trust among themselves, improved communication improves information exchange, empowers self-organizing teams, and boosts team

morale. This increased collaboration boosts total team productivity, resulting in a higher Return on Investment than the sum of individual efforts.

- **Iterative and Incremental Delivery**: The delivery strategy for the project entails breaking it down into small, functional releases or increments. This is done to efficiently control risk while also obtaining early input from consumers and end users. These smaller releases adhere to a strict timetable, with iterations spanning one to four weeks. Initial plans, requirements, designs, code, and tests are established and then incrementally revised to suit project changes.

Cons:

- Long testing periods and insufficient test coverage.
- Project managers must coordinate and communicate extensively with several teams.
- It is not suitable for large projects because it requires multiple iterations to acquire the needed functionality.
- Individual, tiny features may be given too much attention.
- The potential cost of using agile methodologies on large-scale projects may be too great when contrasted to more profitable and leaner ventures.
- Because effective agile methodology application is strongly reliant on excellent teamwork, project managers must be intimately involved in team dynamics (Kumar and G 2012).

5.2.4 DSDM Methodology

The Dynamic Systems Development Method (DSDM) is a project management methodology that encapsulates a plethora of contemporary project management knowledge. DSDM emerged from the software development sector over time as a result of the convergence of software development practices, process engineering principles, and the demands of business development projects. As a result, DSDM has evolved into a versatile framework that can be used to solve a wide range of complicated problem-solving tasks other than software development. Its versatility, as well as its emphasis on teamwork and iterative processes, make it a powerful tool for managing diverse types of projects in today's fast-paced business climate (Delima, et al. 2018).

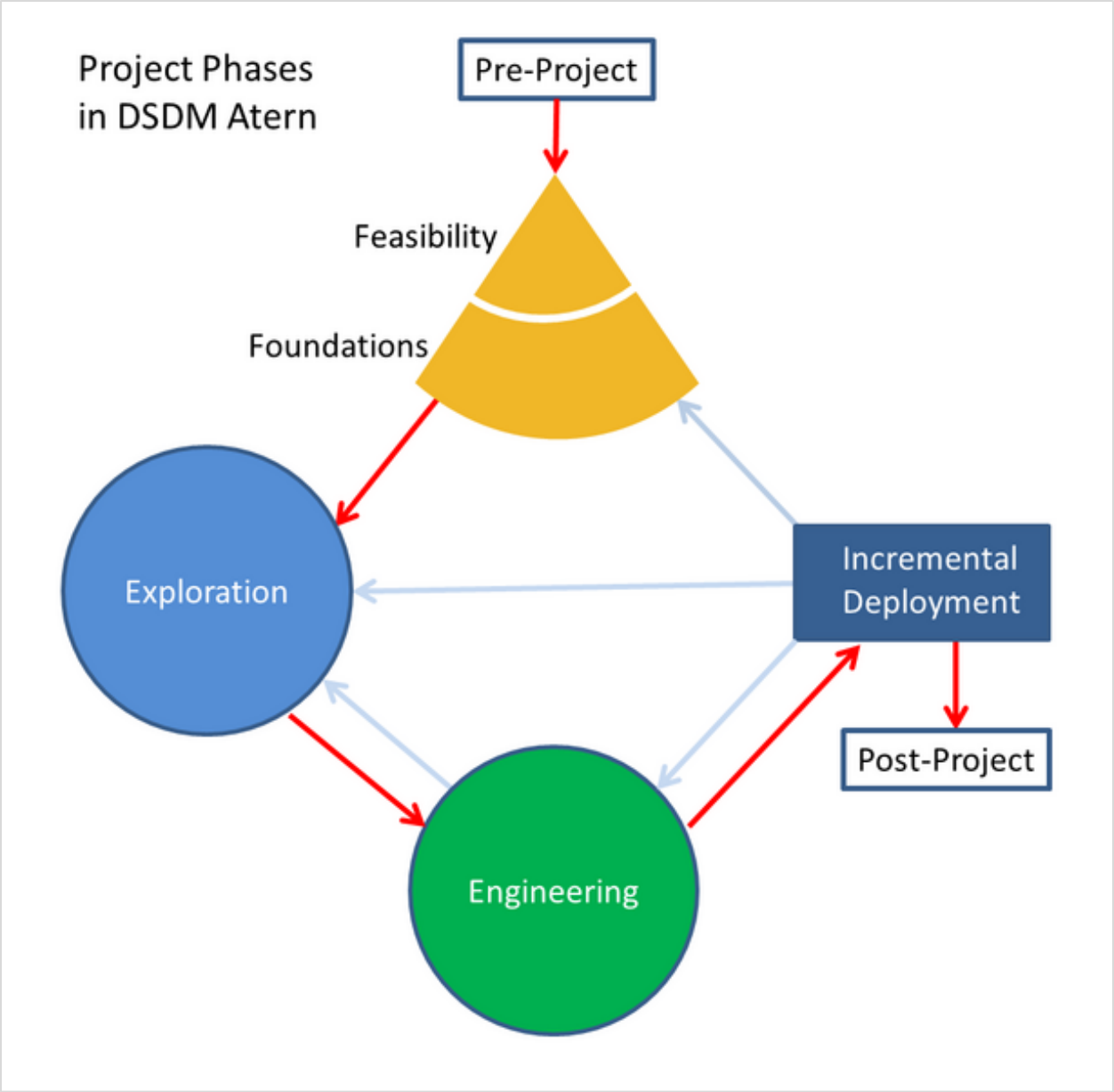


Figure 5.4 DSDM Methodology

Pros:

- **Early Prerequisite Prioritization:** The project starts by prioritizing needs, which allows for an early focus on the most critical features and goals.
- **Iterative Development:** DSDM encourages consistent iterative development, allowing for incremental improvement and adaptability throughout the project.
- **Improved User Experience:** Involving users in the development process ensures that the end product closely matches the user's wants and expectations, resulting in a better user experience.

- **Smooth Deployment:** DSDM emphasizes an organized approach to deployment, which reduces the likelihood of deployment errors and ensures a smoother transition to production.
- **Risk Mitigation:** By regularly reviewing and confirming requirements, DSDM reduces the risk of developing the incorrect project and aids in the maintenance of project alignment with objectives and stakeholder needs.

Cons:

- **Not Suitable for Small Projects:** The structured approach and emphasis on collaboration of DSDM may be extremely difficult and time-consuming for small-scale projects, making it unsuitable in such cases.
- **Beginners' steep learning curve**: due to the complexities and collaborative nature of DSDM, which may necessitate additional training and onboarding activities.
- **Cost Escalation:** Due to Inadequate Planning: If the project lacks thorough planning and management, costs within the DSDM framework can climb, as it relies on disciplined planning and prioritization to effectively control expenses.

5.3 Justification for Recommendation

The development approaches discussed above are largely intended for team-based project management. However, coursework frequently necessitates the completion of specific projects. As a result, we can use components from these techniques to develop a tailored strategy to managing solo coursework tasks. Agile and, more specifically, Scrum, are particularly effective approaches in this environment. Scrum's split of development into sprints, each with its own set of tasks, allows for the project to be broken down into manageable micro-projects that can be done independently. This method has advantages, particularly in terms of testing. DSDM can also help with work prioritization. For the development of this project, I want to use a combination of "Dynamic System Design Model for Multiple Agile" (DSDM with Agile).

5.3.1 Rationale for chosen DSDM method with Agile

For various reasons, the adoption of the Dynamic Systems Development Method (DSDM) in conjunction with Agile methodology for designing an e-voting system based on blockchain is well-justified. To begin, the e-voting system is a sophisticated and vital piece of software, and DSDM's emphasis on handling complex problem-solving tasks is well aligned with the nature of this project. The key principles of DSDM, such as early need prioritization and iterative development, allow us to constantly enhance and adjust the system in response to changing needs and prospective changes in voting legislation. The inclusion of Agile practices within DSDM improves adaptability even further. E-voting criteria can change because of changing security concerns, regulatory requirements, or user feedback. Agile principles

enable us to respond to these changes quickly, ensuring that the system stays secure, transparent, and user-friendly throughout its development lifecycle.

5.3.2 Phases of Selected Methodology

Initial Study:

The Initial Study phase plays a vital role in this project since it entails an in-depth investigation of the project's purpose. It also includes a thorough assessment and comparison of previous research and solutions in the same topic. This phase attempts to determine the uniqueness and necessity of the proposed project in reference to the present research and solution landscape.

Requirements Analysis:

In the requirement analysis phase, we will meticulously prioritize all the identified requirements. Using the Moscow analysis technique, these requirements will be categorized into four distinct groups: 'Must Have,' 'Should Have,' 'Could Have,' and 'Won't Have.' This categorization will enable us to assign importance levels to each requirement and establish appropriate timeframes or time boxes for their implementation.

Design:

During the design phase, we will sketch up the necessary features and build the software interface.

Implementation:

During the implementation phase, we will code the entire software based on the accepted design.

Testing:

During the testing phase, we will rigorously evaluate the software's functionality and performance to ensure it meets the specified requirements and functions as intended.

Evaluation:

The application will be given to the product owner for review in the final stage of evaluation. If the owner is happy with the outcomes, the following phase will begin. If any problems or concerns surface, the development process will return to the analysis phase for additional modification.

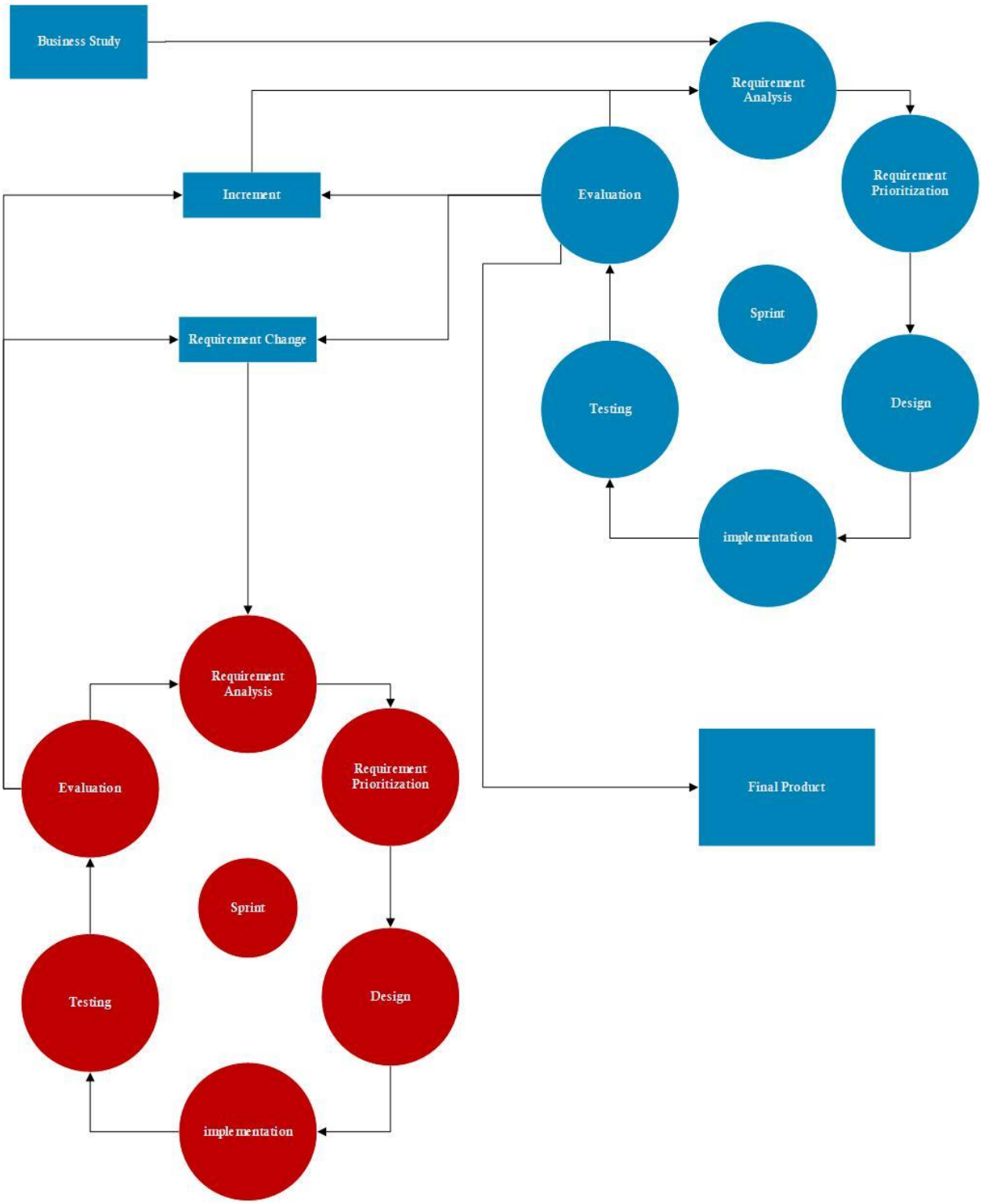


Figure 5.5 DSDM with Agile Framework

5.4 Requirements Gathering Strategies

To design the 'E-voting' application effectively, it is critical to set clear project requirements that are linked with the end purpose. Setting a clear aim guarantees that the project meets its primary objectives. Given that this project is part of academic education, a single developer's scope for significant requirement collecting study is limited. On a smaller scale, though, numerous data collection approaches were used. I'll give you a quick rundown of these procedures down below.

Feedback Mechanism:

Setting up a solid feedback mechanism is critical for requirements collection. This entails establishing channels via which stakeholders and end-users can contribute continual input and insights as the project advances. This continuous communication guarantees that changing needs and preferences are captured, allowing for requirement refinement and validation throughout the project's lifecycle. It's a fluid procedure that keeps the project on track with shifting expectations.

Prototyping:

Prototyping is a helpful method for providing stakeholders with a practical grasp of the proposed system. This entails quickly creating rudimentary, low-fidelity prototypes or mockups. These visual representations provide stakeholders with a real visualization of the system, assisting in the clarification and refinement of requirements. It's a good technique to bridge the gap between abstract concepts and concrete expectations.

Workshops:

Interactive workshops with key stakeholders are an effective approach for gathering requirements. These collaborative sessions may include activities such as brainstorming, prioritization of requirements, and consensus building. Workshops encourage active participation by bringing together varied viewpoints and experience, generating a deeper knowledge of project objectives, and ensuring that requirements are well-defined and aligned with the broader project vision.

Interview:

Conducting interviews was one of the most effective approaches used for this project's application development to acquire data relevant to user wants. Interviews were done with local farmers and people who were interested in using the application. Furthermore, experienced automation experts were interviewed to gain significant insights and suggestions for improving the project.

Document Analysis:

Document analysis was essential in developing the project strategy and establishing requirements. Extensive reading and study of academic papers and scientific research articles were undertaken to guarantee that the project was based on past successful work. This method not only helped discover gaps in the existing literature, but it also avoided past efforts from being duplicated.

5.5 Justification of chosen Technical Tools

After the requirements specification process was completed, many tools were used to successfully manage these needs. Among these tools, the following are the most important:

UML Use Case Diagram:

A use case diagram is useful in assisting a developer or programmer in understanding how end users interact with the system. It aids in the description of the links between user roles and various system modules, as well as the specification of the scope of their access.

Class Diagram:

A class diagram is a graphical representation of the many classes and their relationships in a software system. It provides a high-level overview of the system's structure, explaining how different classes interact and communicate to offer the desired functionality. This diagram is essential for developing and comprehending the software's architecture,

Architecture Diagram:

An architecture diagram is a graphical representation of a software system's high-level structure. It depicts the system's primary components or modules, how they interact, and how data flows between them. Architecture diagrams are used to present stakeholders with a big-picture view of the system's architecture and to assist them comprehend the overall organization and functionality of the system. When developing complicated systems, they are an essential tool for software architects and developers.

IDE/Text Editor (Vs Code)

Visual Studio Code (VS Code) was a critical tool in the creation of the E-voting system. Its adaptability and broad range of plugins aided in effective coding, debugging, and version control, hence expediting the software development process. VS Code's collaborative capabilities also promoted good teamwork among developers.

Entity Relationship Diagram

The Entity Relationship Diagram (ERD) was critical in modelling the database structure for the electronic voting system. It gave a visual depiction of the system's data structure, assisting in the definition of

various data entity associations. The ERD aided in the development of a strong and well-organized database, assuring data integrity and efficient data retrieval.

Sequence Diagram

Sequence diagrams proved indispensable in illustrating the dynamic interactions between system components and users during the voting process. They helped visualize the flow of actions, highlighting key decision points and error handling procedures, ultimately enhancing the system's usability and user experience.

Component Diagram

Component diagrams were instrumental in breaking down the E-voting system into its constituent parts or modules. This visual representation allowed for a clear understanding of the system's architecture and the dependencies between different components. Component diagrams facilitated effective code modularization and maintenance, ensuring scalability and system robustness.

Database:

This project exclusively utilizes the MySQL database system.

5.6 Principle of DSDM

5.6.1 Principal No. 1 Focus on the Business Needs

"Focus on the Business Need" is emphasized as being of utmost importance in project development in the first principle of the Dynamic Systems Development Method (DSDM). Constant communication and input between the development team and the company are essential for analyzing requirements and ensuring alignment with the business's key priorities in the context of modernizing and securing the electronic voting system. As a useful tool for prioritization, the MoSCoW technique (Must have, should have, could have, Won't have) enables the project to remain in line with the most important business needs while allowing additional functionalities depending on their relative relevance. This collaborative and iterative approach guarantees that the project stays closely connected with the business objectives throughout its development lifespan, finally producing a solution that caters to the organization's particular requirements.

5.6.2 Principal No.2 Deliver on Time

The DSDM framework's second principle, "Deliver on Time," is crucial to the creation of an electronic voting system. Elections adhere to stringent timetables and deadlines; thus, it is crucial to guarantee prompt delivery. The voting process’s integrity can be jeopardized, and the public's confidence can be damaged, if certain dates are not met. To follow this approach, the project team must set up precise

milestones, keep a careful eye on development, and rank work items according to their urgency. They must also keep open lines of communication with stakeholders so that any problems can be resolved quickly. The e-voting system is made ready and operational far in advance of any scheduled election thanks to this proactive strategy, guaranteeing that voters have a simple and secure voting process.

5.6.3 Principal No. 3 Collaboration

The foundation of a project's success is collaborative cooperation, which is reinforced by constant client input and feedback. This method offers several significant benefits. First off, it fosters a deeper understanding of the project's aims and anticipated results among team members, ensuring that everyone is on board with the big picture objectives. Second, it greatly quickens the pace of the process. Teams can overcome obstacles and accomplish goals more quickly and effectively by pooling their resources and experience. A sense of shared ownership and responsibility for the project's success is also fostered by this collaborative process between the project's designers and the clients. Project leadership that is effective is essential to achieving these benefits. A project manager must foster a "one-team" culture that promotes teamwork and cooperation while also giving team members the freedom to make informed decisions. This all-encompassing strategy not only promotes project quality but also increases customer satisfaction by producing results that are in line with their expectations and vision.

5.6.4 Principal No. 4 Do not Compromise on Quality

This principle emphasizes the need to decide on the project's target level of quality early in the planning process. The goal of every project activity should be to reach and sustain this predetermined level of quality. From the very beginning of the project, the project manager oversees defining the required quality standard and making sure that quality is maintained throughout the whole project lifecycle. They should carry out early and continuing testing utilizing a variety of techniques, including functionality, usability, and performance testing, to achieve this.

5.6.5 Principal No. 5 Develop in Iteration

Like principle 4, principle 5 is focused on an iterative development methodology that incorporates continuous development and testing. As the project is being built iteratively, the Project Manager is crucial in ensuring that it moves in the right path. They must regularly check to see if the system being developed is in line with the goals of the project. To maximize the project's production quality, the project's development should be modified as necessary. Experimentation and continuous project evolution can lead to this optimization.

5.6.6 Principal No. 6 Build Incrementally from Firm Foundations

The direct advantages for the firm are emphasized in principle six. It emphasizes how quicker project delivery to the client is made possible through incremental development. Because of the early delivery,

developers may better understand the scope of the business and match possible improvements to operational requirements. Before beginning development, it is essential to establish a significant amount of design in order to achieve this. Additionally, throughout this phase, developers should put more emphasis on describing what they want to produce than on the specifics of how to accomplish it.

5.6.7 Principal No. 7 Communicate Continuously & Clearly

To ensure a successful business model, DSDM Atern uses strategies and principles to improve communication among development team members. Effective communication is essential to project success.

Project managers can sustain this idea by carrying out actions that encourage active team communication. This involves fostering in-person encounters and setting up workshops. Additionally, it is crucial to keep both formal and informal lines of communication open with stakeholders. This can be accomplished by maintaining thorough documentation, such as meeting agendas.

This notion can be upheld by DSDM teams in a variety of ways, including by supporting daily stand-up meetings that encourage casual conversations and idea exchange among team members. Facilitated workshops are also quite helpful for increasing stakeholders' comprehension and fostering requirement discussions.

5.6.8 Principal No 8 Demonstrate Control

The team should continuously maintain control of the project, according to the final principle. To achieve this control, project progress must be proactively monitored, managed, and always demonstrated. The proper levels of formality should be used in daily operations to maintain this control and that processes are faithfully followed. Furthermore, this approach should guide the efficient management of contracts and other company requirements.

6 Planning

Without a well-thought-out plan, a goal is nothing more than a faraway fantasy. The secret to great success is diligent planning. The development process is more likely to succeed if the entire project is meticulously planned from the start. The "E-voting" application was also meticulously built to assure its success. We will look at the comprehensive project plan, test plan, risk management plan, change of requirements management plan, and quality assurance management plan in this section.

6.1 Project Plan

The planning for the project itself is the most important of all the plans in this project. It is the pivot for tracking the project's progress. This section will go into the "E-voting" project's planning process.

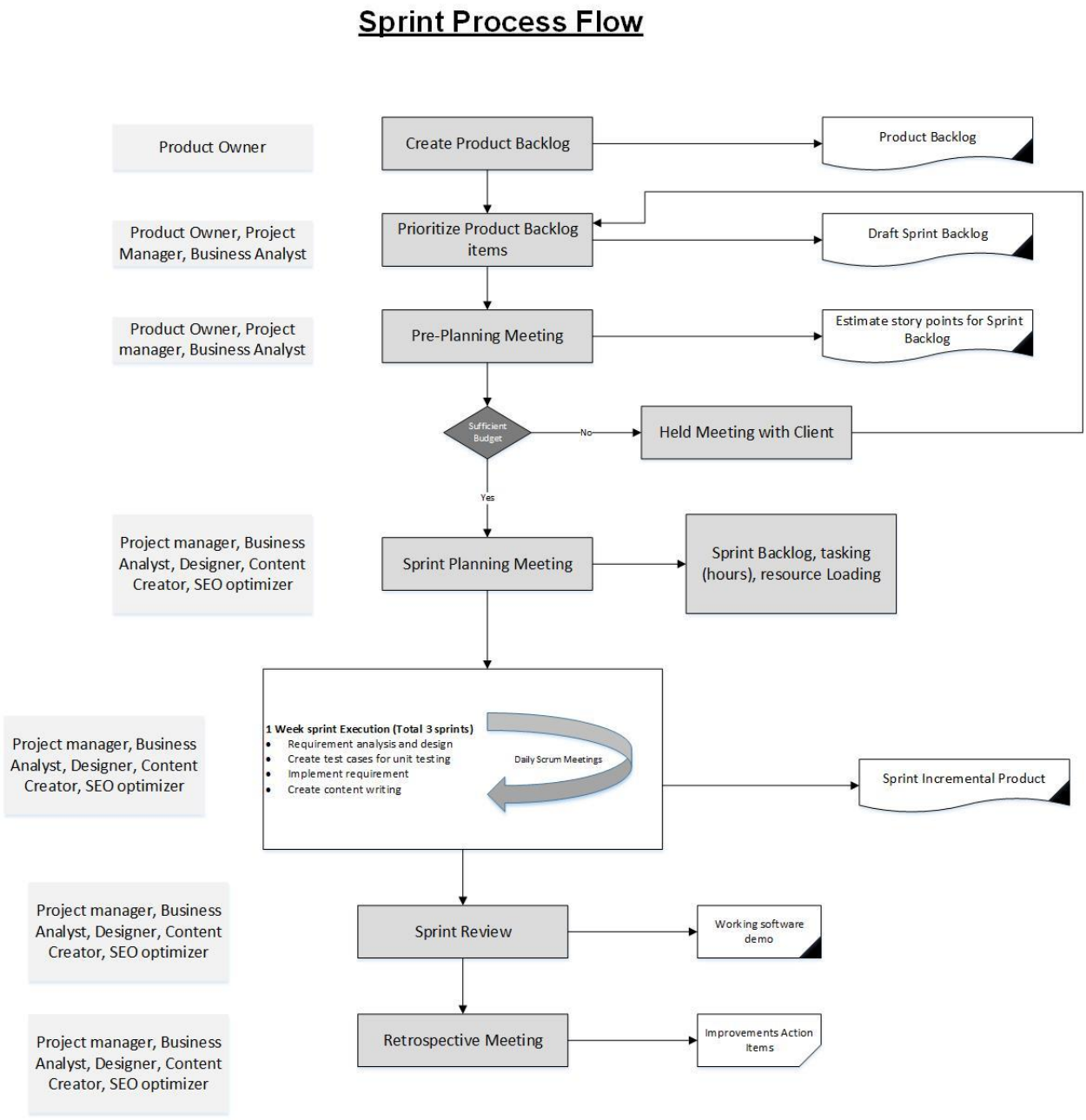


Figure 6.1 Sprint Process Flow

6.1.1 Management Plan

	Task Mode	Task Name	Duration	Start	Finish	Prede	Resou Name
1		**General Research**	**5 days**	**Thu 09-02-23**	**Thu 16-02-23**		
2		Literature Review	4 days	Wed 08-02-23	Mon 13-02-23		Mostafa
3		Product Research	3 days	Mon 13-02-23	Wed 15-02-23		
4		**Critical Analysis**	**8 days**	**Fri 17-02-23**	**Tue 28-02-23**		
5		Rich Picture	2 days	Fri 17-02-23	Mon 20-02-23		
6		Methology	2 days	Tue 21-02-23	Wed 22-02-23		
7		Root Definition	2 days	Thu 23-02-23	Fri 24-02-23		
8		Conceptual Modeal	2 days	Mon 27-02-23	Tue 28-02-23		
9		**Algorithm research**	**13 days**	**Wed 01-03-23**	**Fri 17-03-23**	**4**	
10		Statstical Analysis	4 days	Wed 01-03-23	Mon 06-03-23		
11		Result Analysis	4 days	Tue 07-03-23	Fri 10-03-23		
12		Time Complexity	4 days	Mon 13-03-23	Thu 16-03-23		
13		Analysis and Results	1 day	Fri 17-03-23	Fri 17-03-23		
14		**Design**	**8 days**	**Mon 20-03-23**	**Wed 29-03-23**	**13**	
15		Sketch	3 days	Mon 20-03-23	Wed 22-03-23		
16		Low Fidelty	2 days	Thu 23-03-23	Fri 24-03-23		
17		Mid fidelty	3 days	Mon 27-03-23	Wed 29-03-23		
18		**Development**	**27 days**	**Thu 30-03-23**	**Fri 05-05-23**	**14**	
19		Database	1 day	Thu 30-03-23	Thu 30-03-23		
20		Front End Development	10 days	Fri 31-03-23	Thu 13-04-23		
21		Back End Development	10 days	Fri 14-04-23	Thu 27-04-23		
22		API Integration	7 days	Thu 27-04-23	Fri 05-05-23		
23		**Testing**	**8 days**	**Tue 09-05-23**	**Thu 18-05-23**		
24		Test Plan	2 days	Tue 09-05-23	Wed 10-05-23	21	
25		Test Cases	5 days	Thu 11-05-23	Wed 17-05-23	21	
26		Test Report	1 day	Thu 18-05-23	Thu 18-05-23	21	
27		**Evaluation**	**11 days**	**Fri 19-05-23**	**Fri 02-06-23**		
28		Theoritical Description	4 days	Fri 19-05-23	Wed 24-05-23		
29		Experiment	7 days	Thu 25-05-23	Fri 02-06-23		
30		Terminologies	1 day	Thu 01-06-23	Thu 01-06-23		
31		Programming Language	1 day	Fri 02-06-23	Fri 02-06-23		
32		**Project Report**	**17 days**	**Mon 05-06-23**	**Tue 27-06-23**		
33		Complete Project	38 days	Fri 05-05-23	Tue 27-06-23		

Figure 6.2 Project Management Plan

6.1.2 Work Break Down Structure

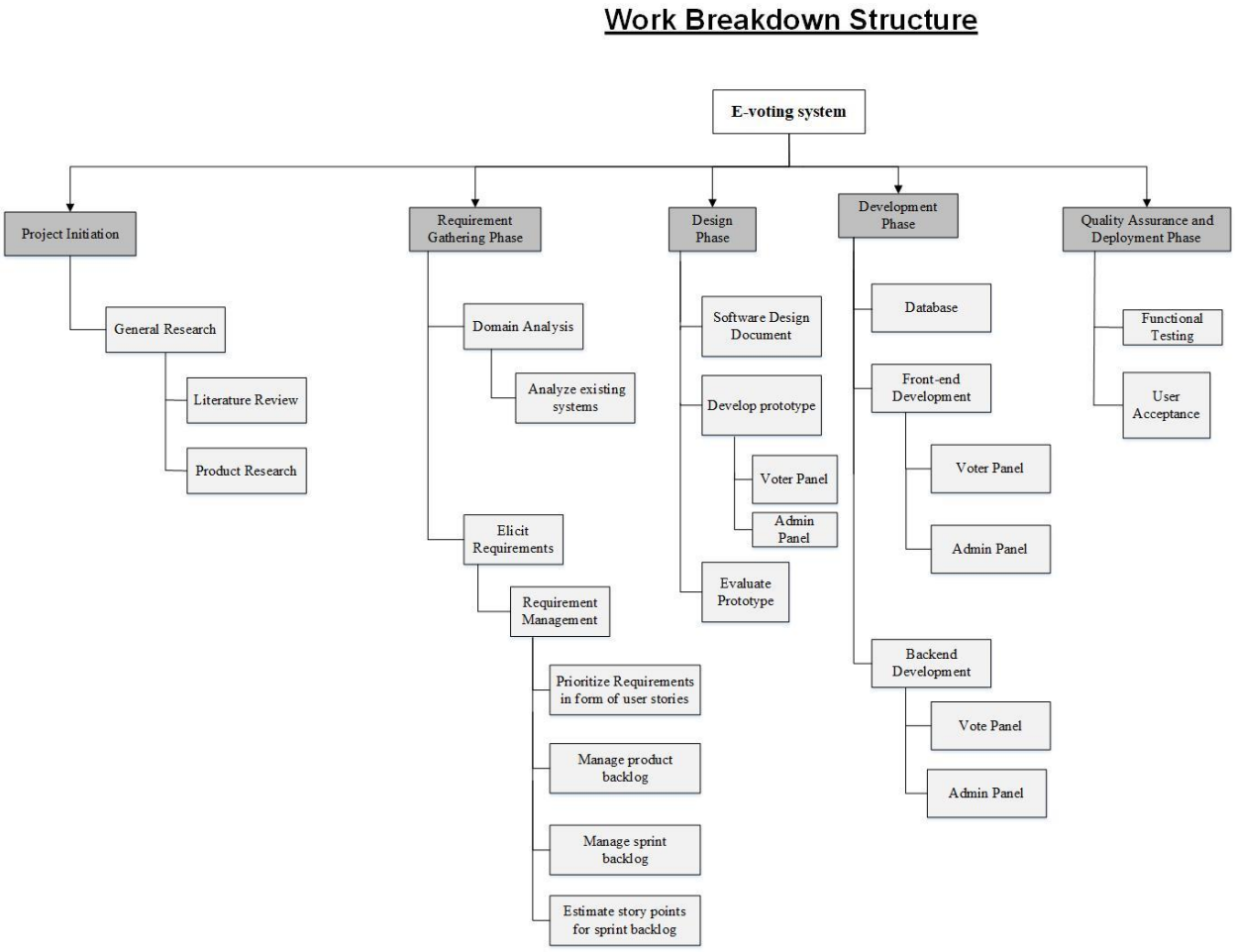


Figure 6.3 Work Break Down Structure

6.1.3 Allocation of Resources

We will build a detailed blueprint defining the distribution of resources for various specific responsibilities within the E-voting development team in this section. Analysts, designers, developers, testers, and users are examples of these positions. The following is the structure of the resource allocation table:

Table 2 Allocation of Resources

ID	Task Name	Duration	Resource Name
1	E-Voting	99 Days	Analyst, User, Designer, Developer, Tester
2	Project Proposal	3	Analyst

3	Introduction	1	Analyst
4	Initial Study	1	Analyst
5	Literature Review	4	Analyst
6	Methodology	2	Analyst
7	Project Plan	3	Analyst, Designer, Developer, User
8	Risk Management	2	Analyst, Developer
9	Change Management	2	Analyst, Developer
10	Feasibility Study	3	Analyst, Developer
11	Foundation	4	Analyst, Designer, Developer, User
12	Exploration	3	Analyst, Designer, Developer, User
13	Engineering	7	Analyst, Designer, Developer
14	Designing	8	Designer, User
15	Development & Deployments	27	Analyst, Developer, Tester
16	Implementation	2	Analyst, Developer, Tester
17	Evaluation	11	Analyst, Developer, Tester
18	Conclusion	1	Analyst

6.1.4 Time Allocation/Time Boxing

	Task Name	Resource Name
1	**E-Voting**	
2	**Time Box 1**	Analyst, Designer, Developer, Tester, QA, User
3	Analysis	Analyst , User
4	Design	Design, User
5	Development	Developer
6	Testing	Tester, User
7	Quality Assurance	QA
8	Increment and Documentation	Analyst
9	**Time Box 2**	Analyst, Designer, Developer, Tester, QA, User
10	Analyst	Analyst, User
11	Design	Design, User
12	Development	Developer
13	Testing	Tester, User
14	Quality Assurance	QA
15	Increment and Documentation	Analyst
16	**Time Box 3**	Analyst, Designer, Developer, Tester, QA, User
17	Analyst	Analyst, User
18	Design	Design, User
19	Development	Developer
20	Testing	Tester, User
21	Quality Assurance	QA
22	Increment and Documentation	Analyst
23	**Time Box 4**	Analyst, Designer, Developer, Tester, QA, User
24	Analyst	Analyst, User
25	Design	Design, User
26	Development	Developer
27	Testing	Tester, User
28	Quality Assurance	QA
29	Increment and Documentation	Analyst
30	**Time Box 5**	Analyst, Designer, Developer, Tester, QA, User
31	Analyst	Analyst
32	Design	Design, User
33	Development	Developer
34	Testing	Tester, User
35	Quality Assurance	QA
36	Increment and Documentation	Analyst
37	**Time Box 6**	Analyst, Designer, Developer, Tester, QA, User
38	Analyst	Analyst
39	Design	Design, User
40	Development	Developer
41	Testing	Tester, User
42	Quality Assurance	QA
43	Increment and Documentation	Analyst
44	**Time Box 7**	Analyst, Designer, Developer, Tester, QA, User
45	Analyst	Analyst
46	Design	Design, User
47	Development	Developer
48	Testing	Tester, User
49	Quality Assurance	QA
50	Increment and Documentation	Analyst

Figure 6.4 Time Allocation

6.1.5 Activity Network Diagram

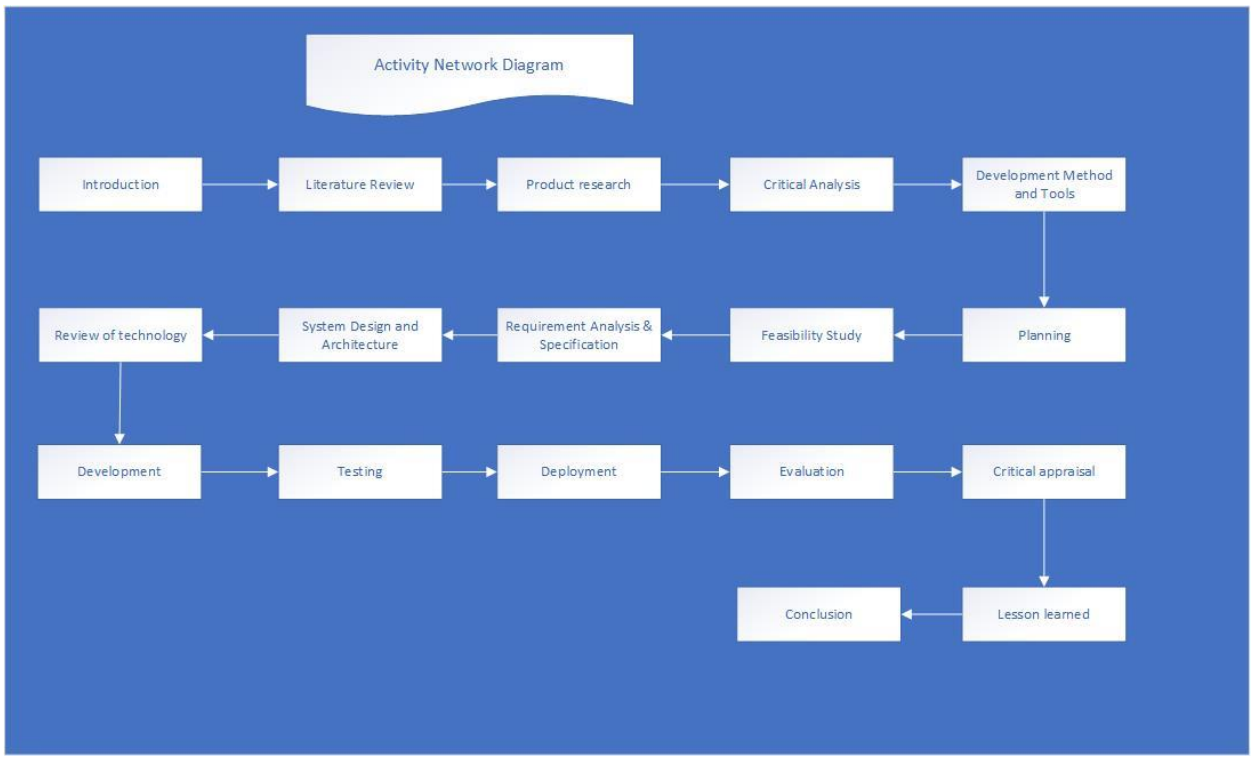


Figure 6.5 Activity Network Diagram

6.1.6 Critical Path

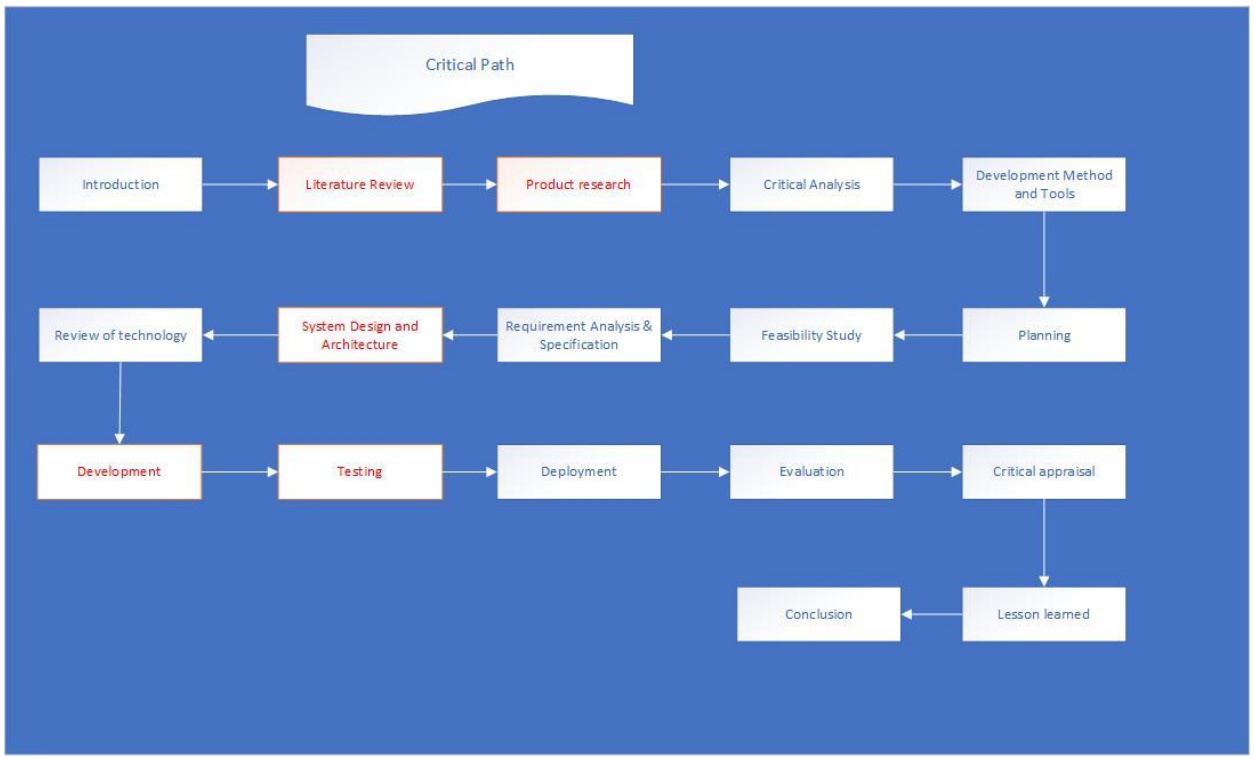


Figure 6.6 Critical Path

6.1.7 Gantt Chart

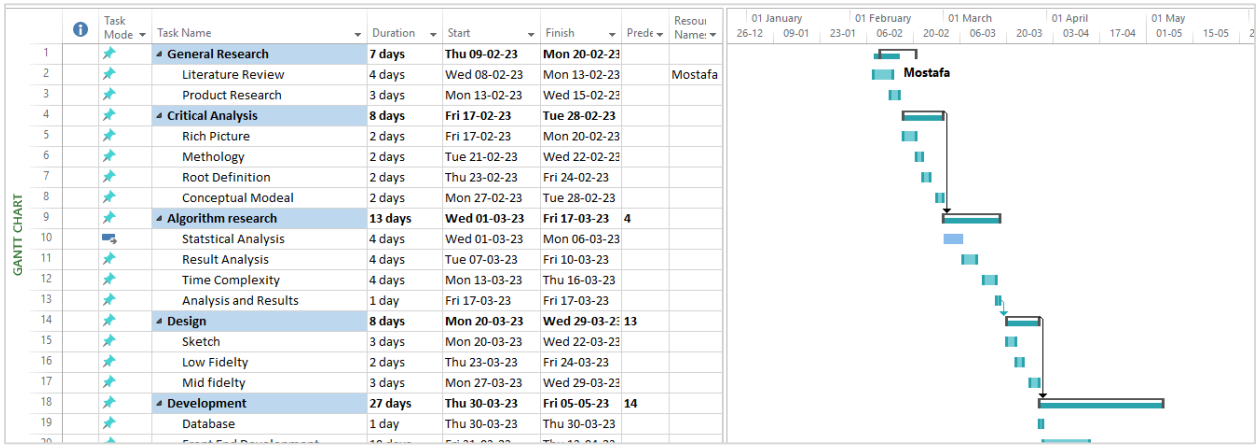


Figure 6.7 Gantt Chart (Part 1)

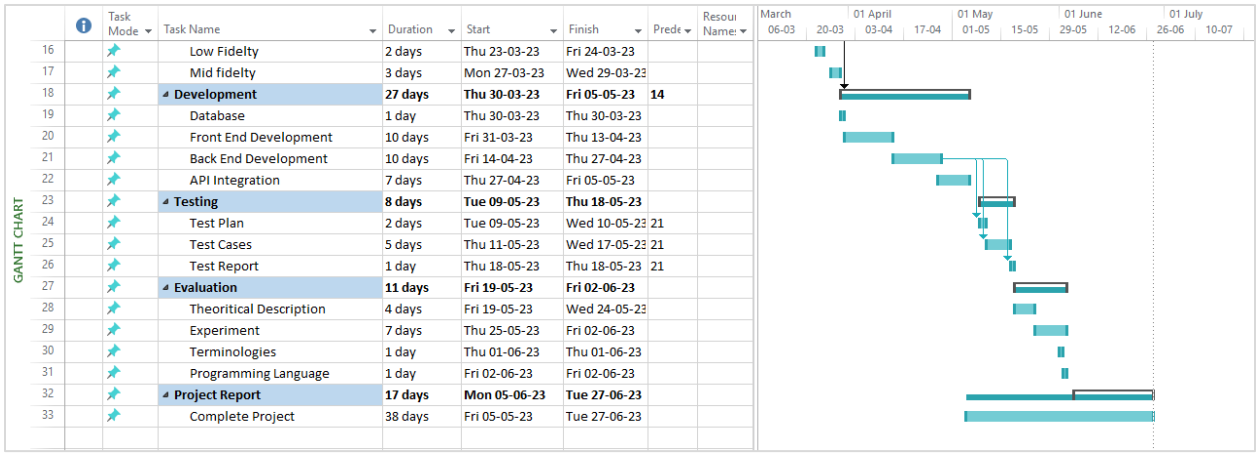


Figure 6.8 Gantt Chart (Part 2)

6.2 Test Plan

Technology is essential in today's world for creating numerous machines that make living easier. Software, however, frequently has a variety of faults and might not work as intended. Testing is therefore essential to maintain control and guarantee software that is error-free. It painstakingly finds every flaw, bringing the software into compliance with the stated requirements. Testing turns out to be quite cost-effective, preventing any faults later. Early bug fixes not only save money but also improve the quality of the final product after testing. The following techniques are utilized in this project to test all our test cases for the application, which uses Mocha as the testing framework:

6.2.1 Required Test Techniques

❖ **Functional Testing**

Software testing must include functional testing, which focuses on determining if a software program correctly executes its functions in accordance with the stated functional requirements. It guarantees that the software performs as anticipated and complies with all operational criteria. Functional Testing is the project's main strategy, and it includes the following crucial elements.

- **Analysis of Overall Functionality:** The primary objective of functional testing is to evaluate the application's overall functionality. The features and functions of the application will be tested to ensure they operate as intended.
- **Alignment Verification:** The testing procedure seeks to verify that the application complies with the listed functional requirements. This indicates that the software must satisfy the standards and demands specified in the requirements documents for the project.

- **Utilization of Testing Tools:** A range of testing tools and frameworks are used to conduct Functional Testing efficiently. Cypress and Selenium tools are frequently used for functional testing as well as for automating web application testing.
    I. **Selenium:** An open-source framework for automating web browsers is called Selenium. It supports a variety of programming languages and enables web application scripting by testers. It is frequently employed for functional testing of web applications.
    II. **Cypress:** Another open-source testing framework made exclusively for web apps is called Cypress. It is renowned for its quick execution and user-friendliness and provides a variety of tools for end-to-end testing.

❖ **Security Testing:**

Security testing is a complex process in software deployment. It is performed to secure confidential information like user credentials, payment information, personal documents. It recognizes and prevents the system from being attacked. Like it assures that only users who have registered themselves and have valid credentials can only log in the system and can perform task that they have permission for. It assures the user to trust on system that their data is stored securely. It reduces risks involved with cyber threats such as ransomware, phishing, data breach, denial of service attack. Identify risk and address in early stage of development, the chances of security breach are reduced. Any security violation can demolish trust of users on a system. It ensures system maintainability, availability, preventing risks that can disturb system service and result in system outage.

Figure 6.9 Principles of Security Testing

❖ **Unit Testing:**

It is mandatory to carry out unit testing on the system for many causes. Performing testing on a single unit to check if it is executing as according to expectations or not. First, it discloses the bugs and defects at an early stage in a single component or unit. Finding defects at early stage facilitates reducing these problems to become enormous or more complex and save system from high-cost issues at high-rise. Moreover, it stimulates developers to write cleaner and more precise code. Developers are encouraged to

follow the most suitable practices of designing and coding to take code quality to the next level by performing unit testing. Additionally, unit test also informs as in the form of document how every component integrates with other to build a function. It gives assurance to the intended stakeholders that the unit is performing as expected. Refactoring code becomes easier by performing unit testing because developers know if any defect occurs unit testing will catch this.

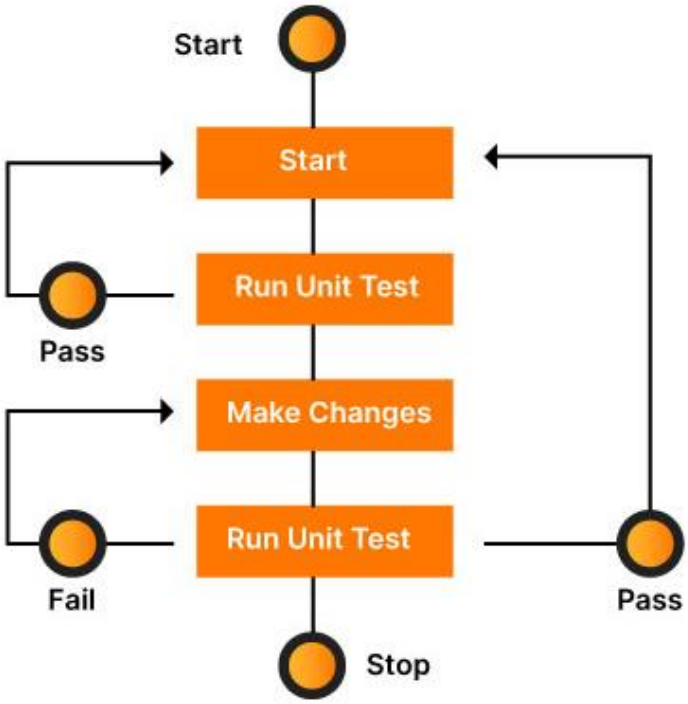


Figure 6.10 Process of Unit Testing

❖ **Integration Testing:**

Integration testing is the complex stage in the software development lifecycle. Firstly, it encounters issues raised when two or more components integrate with each other in system to become a function or functionality. In integration testing before evaluating the code of entire system, code of each integrated module is checked first. Normally, when components are isolated and unit test has been performed successfully, they may raise defects and issues on communication on integration. In integration testing bugs like miscommunication, inconsistency is discovered. Integration testing also handles errors in exceptional cases like incorrect input. Integration testing assures that the control flow is consistent between two or more components that the product is working as expected. Integration testing ensures that the requirements are fulfilled, and the product is developed according to requirements. By disclosing risks at an early stage, it prevents defects from becoming more costly in future.

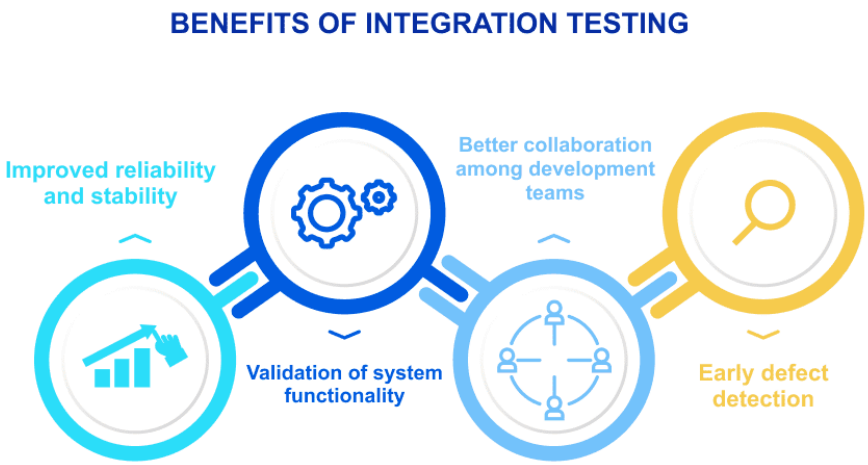


Figure 6.11 Benefits of Integration Testing

❖ **User Acceptance Testing:**

In acceptance testing, a user examines and validates the work of another person to ascertain whether it serves the intended goal. Its main objective is to provide the user confidence that the programmed is truly appropriate for the use for which it is intended. Acceptance testing essentially functions as a software product's validation procedure. Although the end-user or client normally conducts the evaluation, other stakeholders may also be involved. The several types of acceptance tests include operational acceptance tests, regulatory acceptance tests, contract acceptance tests, and user acceptance tests (which include internal alpha tests and external beta tests). Assurance-giving is the main goal of user acceptance testing. Testing of this kind can be used for presentation, demonstration, probing, usability evaluation, and validation, among other things (Pallavi Pandit and Swati Tahiliani 2015).

Tools used in UAT

User Acceptance Testing (UAT) can be performed using a variety of technologies, as described in reference [22]. Notably, acceptance test cases are created using acceptance criteria by tools like Cucumber, Jira, Explorer, and RSpec.

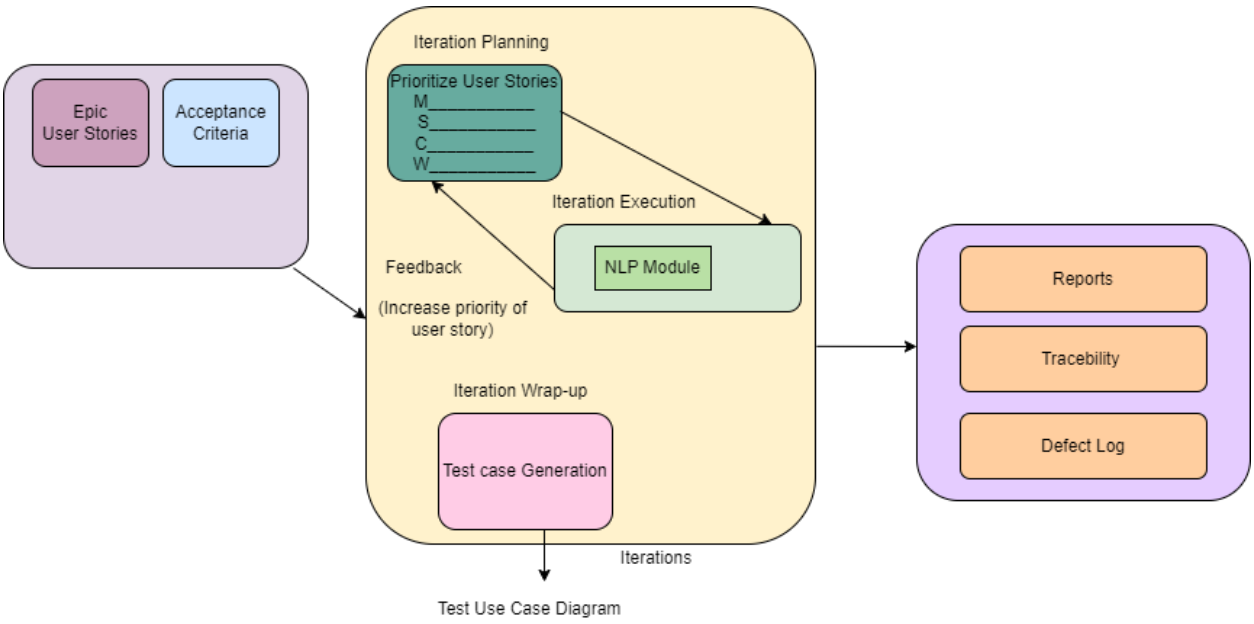


Figure 6.12 Agile UAT

6.2.2 Testing Environments

The testing environment for the "E-voting" system will include the following components:

Required Test Environment:

- ✓ One PC with compatibility for Windows, Ubuntu, or macOS.
- ✓ A minimum of three Android devices with varying screen sizes and configurations.
- ✓ One iPhone for testing iOS compatibility.
- ✓ Various web browsers such as Google Chrome, Firefox, Microsoft Edge, Safari, and Opera to ensure cross-browser compatibility.
- ✓ Access to the database and server resources for thorough testing and validation

6.2.3 What Must Be Tested

To ensure the development of a very successful software system, it is critical to rigorously perform all the previously described test plans. These plans will be carried out for each individual unit of each aspect of the system. Once this granular testing is completed, the entire system will be thoroughly tested as all the features are merged.

Following that, an incremental development strategy will be used, with rigorous testing performed after each step. Non-functional testing will be performed after the functional tests are completed. This meticulous testing approach is required to ensure the system's dependability, performance, and overall quality.

6.2.4 Risk Management Plan

Since the 17th century, the term "risk" has been used to describe rushing into danger or facing barriers such as rocks. In current times, the notion of risk is widely used in a variety of sectors, and it is frequently expressed using phrases such as "hazard," "threat," "challenge," or "uncertainty." Risks have a dual nature in the architecture, engineering, and construction (AEC) business, indicating both the possibility of unwanted hazards and their potential effects. When dealing with complicated projects in this industry, this comprehensive awareness of risk is critical (Zou et al. 2017). Risk is an unavoidable component of our life, whether we work in a company, industry, or on any endeavors. Risk-taking is an unavoidable component of the route to progress and growth. We may reap the benefits of achievement only if we overcome these risk factors. In the case of 'E-voting,' the same principles apply. There are risks, and they must be recognized and managed efficiently. It is critical to understand risk management to navigate these hazards. In the following sections, we will look at how to manage the risks connected with voting online (e-voting) and how it can be done efficiently.

Figure 6.13 Risk Management Plan Process

6.2.5 Identify Risk

Several types of risks may arise throughout the development process of the "E-voting" application. All potential risks will be recognized at an early stage of the development process in this phase risk management method.

Here are listed below:

Technical Risks:

- **Security Vulnerabilities:** The risk of security breaches, hacking, or cyberattacks jeopardizing the voting system's integrity.
- **Software Bugs and Glitches**: Development faults or software bugs that may affect the application's reliability and functionality.
- **Compatibility issues:** Incompatibility with multiple devices, operating systems, or web browsers, resulting in voter usability issues.
- **Data Loss:** The risk of losing or corrupting data, particularly during data transit or storage procedures.
- **Scalability Issues:** During peak voting hours, the application may not scale efficiently to support many concurrent users.

Internal Risks:

- **Inadequate competence or abilities** within the development team to address complicated technical obstacles or security risks.
- **Limited cash, time, or human resources** allocated to the project, which may have an impact on development quality.
- **Poor project planning,** communication, or collaboration resulting in delays or scope creep are examples of project management issues.
- **Inadequate testing techniques** may fail to uncover key flaws before deployment.

Environmental Risks:

- **Natural disasters**: Events such as earthquakes, hurricanes, or floods that potentially affect data centre or infrastructure operations.
- **Power Outages:** Interruptions in electrical supply that may influence the application's availability and reliability.
- **Pandemics or Health Crises**: Events such as pandemics may have an influence on the ability to hold elections or develop the application as intended.
- **Social Unrest** that could interrupt the voting process or project timeline is referred to as social turbulence.

External Risks:

- **Changes in election regulations** or legal requirements that may demand adjustments to the application.
- **Political Interference**: External influence or political pressure on the development process that may jeopardize the system's integrity.

- **Risks connected with vendor** or third-party vendors: Reliance on third-party suppliers for certain components or services introduces risks related with their performance and security (Tupa et al. 2017).

6.2.6 Analyze Risks

Following the identification of potential risks, risk analysis is a vital stage. After identifying these risks, it is critical to analyze them based on characteristics such as their likelihood of occurrence, frequency of occurrence, and potential impact. This analysis is critical because it provides us with useful insights that allow us to make informed decisions and take necessary steps to effectively manage and reduce these risks.

Table 3 Risk Analysis

Risk	Risk Specification	Likelihood	Frequency
Technical Risks	Arise due to security breaches	Likely	1-2
	Arise due to incompatibility with various devices, operating systems	Likely	1-2
Internal Risks	Insufficient expertise or skill	Likely	1-2
	Limited budget	Likely	1-2
	Poor project planning	Likely	1-2
External Risks	Changes in electronic regulations	Rare	0-2
	Political stress	Rare	0-2
Environmental Risks	Arise Due to Natural Disasters	Rare	0-2
	Interruptions in electrical supply	Unlikely	5-6
	social unrest	Likely	1-2

6.2.7 Prioritize Risks

Table 4 Prioritization of Risks

Risk	Risk Specification	Likelihood	Frequency	Prioritization
Technical Risks	Arise due to security breaches	Likely	1-2	High
	Arise due to incompatibility with various devices, operating systems	Likely	1-2	High
Internal Risks	Insufficient expertise or skill	Likely	1-2	High
	Limited budget	Likely	1-2	Medium
	Poor project planning	Likely	1-2	Medium
External Risks	Changes in electoral regulations	Rare	0-2	High
	Political pressure	Rare	0-2	Low
Environmental Risks	Arise Due to Natural Disasters	Rare	0-2	Medium
	Interruptions in electrical supply	Unlikely	5-6	High

	Political or social unrest	Likely	1-2	Medium

6.3 Change Management Plan

The use of DSDM (Dynamic Systems Development Method) and Agile methodologies in the development of the 'E-voting' system is an appropriate approach, especially given the inherent dynamism of software development and the frequent need for adjustments and enhancements to requirements. These approaches excel in their adaptability and flexibility to handle changes in the middle of a project. The MoSCoW prioritization approach in DSDM is useful for categorizing requirements as "Must have," "Should have," "Could have," or "Won't have." This facilitates the management of changing priorities and ensures that the most important features are addressed first, while less important ones can be added as needed. Furthermore, Agile techniques, including DSDM, promote an incremental development approach, which is frequently organized into sprints. This method enables the release of functional software in short cycles, making it amenable to tolerating mid-cycle changes. New requirements can be smoothly integrated into the following sprint, ensuring that the system stays versatile and responsive to changing demands throughout its development lifecycle.

6.3.1 Factors can cause changes

- New Legal or Regulatory Requirements
- Stakeholder Feedback
- Voter Demands
- Technical issues
- Vendor or Technology change

6.3.2 Key Change Decision Makers

This is an academic research project in which a single person will serve as developer, designer, tester, analyst, business visionary, and user. As a result, this person will be crucial in making vital decisions throughout the project.

6.4 Quality Management Plan

A project can only be considered successful if specific quality requirements are met. Quality management is critical in ensuring that the project not only satisfies these criteria, but also maintains usability, stability, and consistency of performance throughout time. Quality management's key goal is to ensure long-term sustainability while functioning in a short time frame.

6.4.1 Using Riles to Maintain Quality

To maintain the quality of the "E-voting" system, the following rules were followed.

Software Quality Assurance (SQA)

Software Quality Assurance (SQA) for an E-voting system in accordance with ISO 9000 standards is critical to ensuring that the system fulfils the highest quality and reliability requirements, promoting trust in the electoral process. ISO 9000, a widely accepted standard model for quality management, provides a solid framework for the SQA of E-voting systems. ISO 9000 emphasizes the significance of a well-defined quality management system (QMS) first and foremost. This includes the adoption of explicit quality objectives and policies unique to the software development process in the context of an E-voting system. It also involves the documenting of standard operating procedures, as well as precise blueprints for the system's design, development, and testing phases.

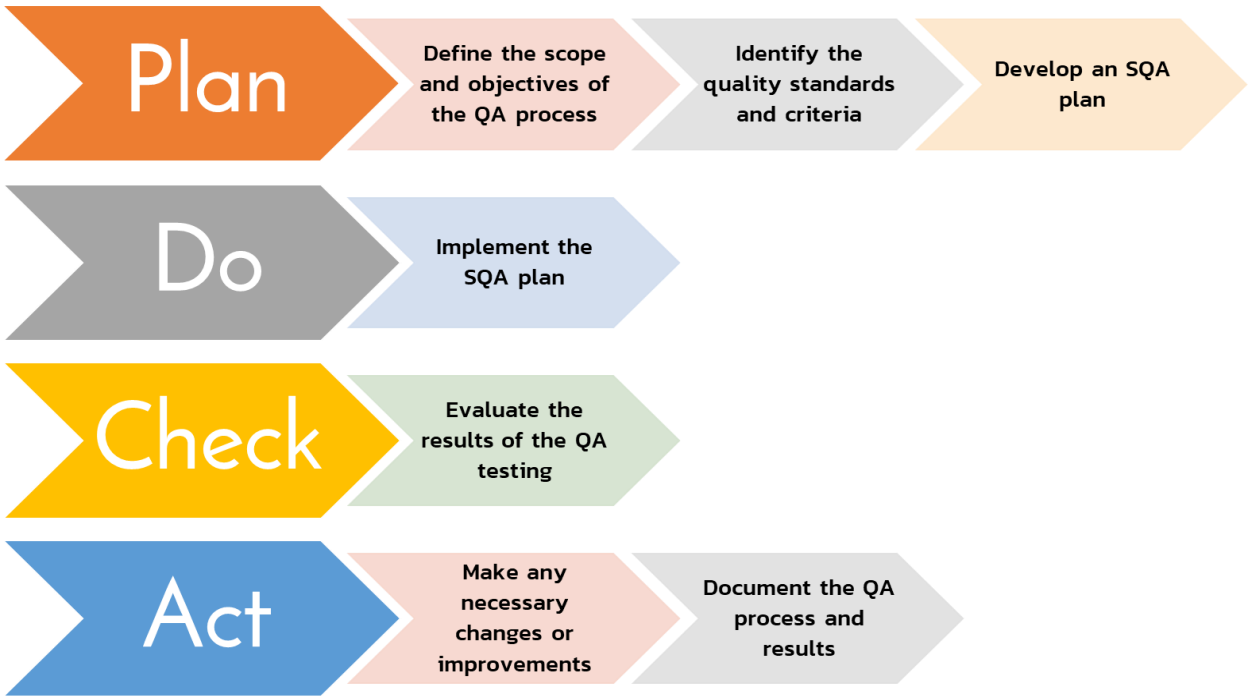


Figure 6.14 SQA Process

Software Quality Control

Several activities will be carried out to ensure the specified quality of the 'E-voting' application. These will be carried out iteratively throughout the development life cycle, with the tasks unfolding as follows:

Table 5 Software Quality Control (SQC)

Review	Testing
Requirement Review	Unit Testing
Code Review	Integration
Design Review	Functional Testing
Deployment Plan Review	System and Security Testing
Test Plan Review	Acceptance Testing
Test Cases Review	User Acceptance Testing

6.4.2 Quality Plan and Measuring

A quality plan for an electronic voting system is required to ensure that the system is dependable, secure, and achieves the required performance standards. The plan defines the processes and procedures that must be followed to achieve and maintain the intended level of quality throughout the development and operation of the system. A quality plan aids development by providing a scale for measuring system quality.

The study's questions revolve around determining the quality of a solution, particularly in the context of software development. These questions aid in the evaluation of many aspects of the development process and final product. Each question is interpreted as follows:

- **Is business engagement sufficient?** The level of interaction and collaboration between the development team and the business or client stakeholders is addressed in this question. It determines whether the development team understands and meets the business needs efficiently.
- **Is the development team truly in charge?** This question investigates if the development team has the authority and autonomy to make project decisions. Teams that are empowered are frequently more efficient and capable of producing high-quality solutions.
- **Is the life cycle observed?** This question asks whether the project is following its software development life cycle. It determines whether the established processes and phases are being followed correctly.

- **Are items being produced?** This inquiry is concerned with the tangible results of the development process. It assesses if the development team consistently produces deliverables on time in accordance with project milestones.
- **Is reviewer feedback being considered?** This question emphasizes the significance of feedback integration. It evaluates whether feedback from stakeholders or peer reviewers is actively considered and incorporated into the development process.
- **Is it feasible to reverse development?** This question determines whether there is a system in place for dealing with problems or making changes when they arise. It assesses the project's adaptability and ability to reverse or alter decisions as needed.
- **Is the priority order prioritized?** This question is about task and feature prioritization. It investigates whether the development team is focused on high-priority items to handle the most crucial components of the project first.
- Are the time constraints carefully adhered to? This question analyses the project's adherence to time limitations. It assesses whether timeboxing procedures are being strictly followed in order to keep the project on track.

7 Feasibility Study

A feasibility study is preliminary research carried out to examine the viability of a proposed project or venture. Its goal is to give an unbiased assessment that considers all aspects of the project, including technical, economic, financial, legal, and environmental concerns. This research helps to determine whether the initiative is viable and desirable. A feasibility study's four key components are technical feasibility, financial feasibility, market feasibility, and operational feasibility. These factors interact to decide whether a project is viable and should be pursued.

7.1 Technical Feasibility

The technological feasibility of developing an e-voting application is vital to its viability. It assesses whether the project can be completed successfully in terms of technology. Conducting a feasibility analysis based on the capability of a single developer is a reasonable way for determining the viability of a software development project. The following elements should be considered:

- Examine the developer's technical abilities, such as programming languages, development tools, and knowledge of technology. Determine whether the developer have the necessary knowledge to complete the project on his or her own.
- Analyze the e-voting application's project complexity. Consider the number of features, the security needs, scalability, and integration with external systems. Determine whether the developer can handle the project's complexities.
- Determine the resources that a single developer has access to, such as hardware, software licenses, and development environments. Ascertain that the developer has access to the necessary tools and resources for productive development.

7.2 Market Feasibility

An e-voting application's commercial feasibility is crucial in establishing its viability. It comprises analyzing demand, potential user base, competition, and market dynamics. When doing market feasibility research for an e-voting application, the following criteria must be considered:

- Examine whether electronic voting solutions are required in the chosen region or jurisdiction. Determine whether there is a recognized need for more accessible and efficient voting procedures.
- Determine the e-voting application's intended audience. This includes eligible voters, election officials, political parties, and other stakeholders.
- Keep up to date on new developments in e-voting technology and voter preferences. Consider the use of mobile voting, blockchain integration, and remote voting possibilities.

7.3 Operational Feasibility

The operational practicality of an e-voting application is a significant factor in determining its viability. It is concerned with whether the proposed system can be successfully deployed and incorporated into the existing operating environment. The following are the most important factors to consider when conducting operational feasibility for an e-voting application:

- Examine whether the e-voting application can interface easily with existing election management systems, registration databases, and associated infrastructure.
- Determine whether the essential resources, such as hardware, software, staff, and financing, are available to support the application's development, implementation, and maintenance.

7.4 Financial Feasibility

Financial feasibility is an important factor in determining the viability of an e-voting solution. It entails assessing the project's financial features to evaluate whether it is economically viable and justifiable. The following are the major elements to examine when determining the financial feasibility of an e-voting application:

7.4.1 Hardware List

Here is a list of hardware and software requirements that may be required for an academic project involving the development of an e-voting application by a single person:

- A trustworthy computer with enough processing power, memory, and storage space to support software development duties.
- Keyboard, Mouse, and other Accessories
- Mobile devices (smartphones and tablets) with different operating systems (iOS and Android) for testing mobile compatibility.
- AMD RYZEN 7 (3rd Generation) Desktop Processor
- Desktop RAM: 64 GB (G-Skill DDR4, 2666 MHz) Desktop GPU: AMD Radeon R7 240 (4 GB)

7.4.2 Software List

- Operating System: Windows 10
- Microsoft Office 2016
- Microsoft Project 2013
- Microsoft Visio 2013
- Adobe Photoshop 2019
- Adobe XD 2019
- Microsoft Office 365 Online

- Draw io
- PDF Reader

Hardware Cost

Table 6 Hardware Cost

Hardware	Cost (Dollars)
Personal Computer	$700
UPS: Power Pac (1200 Volt)	$500
Keyboard	$500
Mouse and other accessories	$1500
Router	$500
Laptop Graphic card	$300
RAM	$1200
Switches, firewalls, and load balancers.	$3000
Laptop Processor: Hp Core i7	$3500
Total Cost	**$11700**

7.4.3 Software Cost

Table 7 Software Cost

Software	Cost
Operating System Windows 10	$1500
Microsoft Office 2013	Free
Microsoft Project 2013	Free
Microsoft Visio 2013	Free
Adobe Photoshop 2019	$500
Microsoft Office 365 Online	Free

Draw.io	Free
Adobe Reader	Free
PDF Reader	Free
Arduino IDE	Free
Total Cost	**$2000**

7.4.4 Hosting Cost

Table 8 Hosting Cost

Service Name	Package Name	Rent	Total
Name Cheap	Production	$ 2250	$2700 year
Play store	One Time	$3000	$3000
Total Cost			**$3000**

7.4.5 Additional Cost

In addition to the upfront costs of hardware, software, and hosting, there are several continuing charges to consider when budgeting for an e-voting solution. These are some examples:

Table 9 Additional Cost

Service Name	Duration	Rent/month	Total
Internet Connectivity Cost	6 Months	$ 800	$4800
Poer Cost	6 Months	$ 1000	$6000
Total Cost			**$10800**

7.4.6 Cost Benefit Analysis

The application's development is projected to incur significant upfront costs. Nonetheless, these costs are expected to fall with time, and the solution's profitability is expected to rise significantly. We give a three-year cost-benefit analysis for this project in the following section, demonstrating how the investment is projected to yield growing returns.

Table 10 Approximate Cost

ID	Cost Area	1st Year	2nd Year	3rd Year
1	Hardware	$11700		
2	Software	$2000	500	500
3	Domain & Hosting	$2500	2500	2500
4	Others	1000	1000	1000
5	Employee Salary	4000	4500	5000
Total		$17600	$8500	$9000

7.5 Analysis of the System's Major Organizational Changes

Major organizational changes, particularly when using an active server-based automation architecture, can be difficult to implement. These adjustments are not only technically difficult, but they also have a substantial societal impact. Due to the complexities required in setting up and fine-tuning the architecture, the transfer to such a system frequently necessitates a significant time investment. This complexity is exacerbated by the project's engagement of various stakeholders and parties, each with their own set of interests and opinions. As a result, the organization's social dynamics and interactions become critical variables to examine. Despite the hurdles, the development of an e-voting system represents a chance to not only streamline the electoral process but also introduce benefits to overall corporate administration. By adopting this technology revolution, the organization may improve efficiency, transparency, and data management, leading to more effective and responsive decision-making processes in the long run.

7.5.1 Change in Business Structure

Market dynamics can have a significant impact on a company's corporate structure. While adapting to these changes might provide several benefits, it is not without obstacles. For example, implementing an active e-voting management system is a substantial change that a corporation may contemplate. On the plus side, such a system can increase transparency, decision-making processes, and general governance. It

could provide an opportunity to interact with a digitally sophisticated customer base while also adapting to changing market trends. However, transitioning to such a system can be difficult due to the need for significant technological expenditures, potential resistance from conventional stakeholders, and the need for extensive training and assistance. As a result, while changes in organizational structure might provide promising benefits, careful planning and management are required to offset potential disadvantages and ensure a smooth and effective transition.

7.5.2 Modifications to Business Operating Procedures

Implementing an e-voting system causes a substantial shift in company working methods, especially in the context of election processes. Traditional paper-based voting methods are being replaced by digital technologies that are more efficient, accurate, and accessible. The reform affects many things, including voter registration, ballot casting, and result tabulation. Digital databases and online registration methods are replacing manual registration as the standard. Voters can vote online from anywhere with an internet connection, transforming the voting experience. The counting and verification operations are automated, decreasing error. This shift, however, is not without hurdles since it demands robust cybersecurity measures to ensure the system's integrity. Furthermore, employees must be trained to run and maintain the e-voting infrastructure. Despite the necessary adaptations, the introduction of an e-voting system promises to improve the democratic process, streamline operations, and boost accessibility, ultimately leading to a more efficient and inclusive electoral system.

7.5.3 Change in Business Policy

The implementation of an e-voting system involves a significant shift in company strategies, particularly those concerning the election process. These policy changes are important to the successful use of digital voting methods. The development of comprehensive cybersecurity policies and methods to ensure the integrity and confidentiality of voter data and election results is one of the most significant changes. To address the gathering and storage of voter information in digital formats, data privacy policies must be revised. Furthermore, voter registration and authentication policies should be changed to support online registration and verification processes. Furthermore, policies governing the duties and obligations of election officials, voters, and the e-voting system itself are critical. This involves auditing procedures and guaranteeing transparency throughout the electoral process. Policies for business continuity must also be updated to meet contingencies in the case of system outages or cyber threats.

7.5.4 Change in Staff Attitude

The transition to an e-voting system necessitates a considerable shift in employee attitudes inside the organization in charge of overseeing the electoral process. The transition from old, paper-based voting systems to digital voting platforms necessitates a new attitude and approach. Staff workers, especially

election officials and support personnel, must embrace technology and learn how to use it. They must recognize the potential benefits of electronic voting, such as enhanced efficiency, accuracy, and accessibility, and comprehend how this change might improve the democratic process.

Staff must also adjust to new duties and responsibilities, such as supervising the digital infrastructure, maintaining cybersecurity, and assisting voters via electronic systems. This transition can be difficult because it may necessitate further training and the development of digital literacy skills. Furthermore, an openness to change and a readiness to welcome innovation are required. The attitude of the personnel towards the deployment of an e-voting system can have a considerable impact on the success of the transition. Staff workers with the correct mindset can help the system run smoothly and create voter faith in new technology, ultimately improving the election process.

7.6 SWOT Analysis

SWOT analysis is an acronym that stands for assessing a company's strengths, weaknesses, opportunities, and threats. The following table summarizes the aspects connected to the 'E-voting' application's SWOT analysis:

Table 11 SWOT Analysis

Strengths	Weaknesses	Opportunities	Threats
Secure and verified access for voters	If the system's security is compromised, it could undermine the entire voting process.	The system could capitalize on the automatic report generation feature to streamline administrative processes.	The system could be vulnerable to cyberattacks or data breaches, compromising the integrity of elections.
Voters can fluently cast their votes,	Internet access is a prerequisite for using the system, which may limit accessibility for some users.	By allowing multiple constituencies, the system can facilitate elections across different regions, potentially increasing its adoption.	Adhering to local and national election laws and regulations can be complex and may pose legal challenges.
The system displays functionalities	The system's various features, including	Opportunities exist to educate users on the	Some voters and stakeholders may resist

according to user roles, enhancing usability and security	admin functions, could make it complex to manage and use.	benefits and proper usage of the system.	the transition from traditional voting methods to an electronic system.
Voters can view published results, promoting translucency and trust in the electoral process	Developing and maintaining the system may require significant resources	The system could integrate with other civic services or databases to streamline voter registration and verification.	Technical glitches or system failures during elections could erode trust in the system.

8 Requirements Analysis and Specifications

In this section, we will review the specifications of the 'E-voting' project and identify the ones that will be implemented to ensure the best project outcome.

8.1 Rich Picture

A Rich Picture is a powerful way for delving into difficult problems, truly understanding them, and visually expressing them through diagrams. It's a useful tool for learning about different parts of a scenario, whether it's a business process, a problem, or a system. The method starts with drawing up a visual depiction of the scenario, which is often unstructured and rough. This preliminary representation may comprise people, objects, procedures, and interactions, conveying the core of the scenario. The important thing to remember here is that it does not have to be a polished or formal diagram; rather, it should be a free-flowing, creative picture.

Individuals can debate and explore many facets of a problem more easily when they visualize it. It can expose hidden intricacies, interdependencies, and even opposing opinions that might not be obvious in text or vocal communication.

8.1.1 Rich Diagram

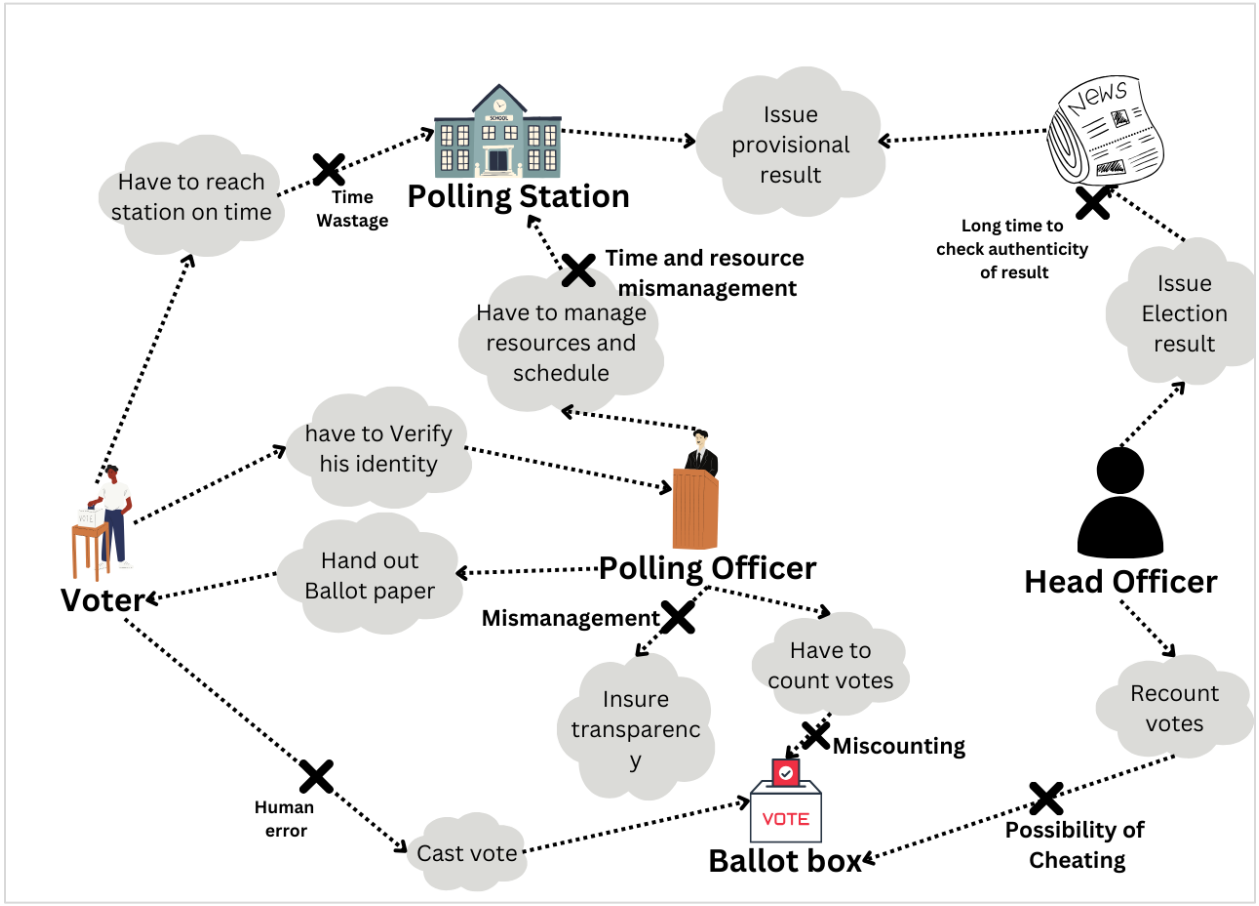


Figure 8.1 Rich Picture Diagram of current voting system

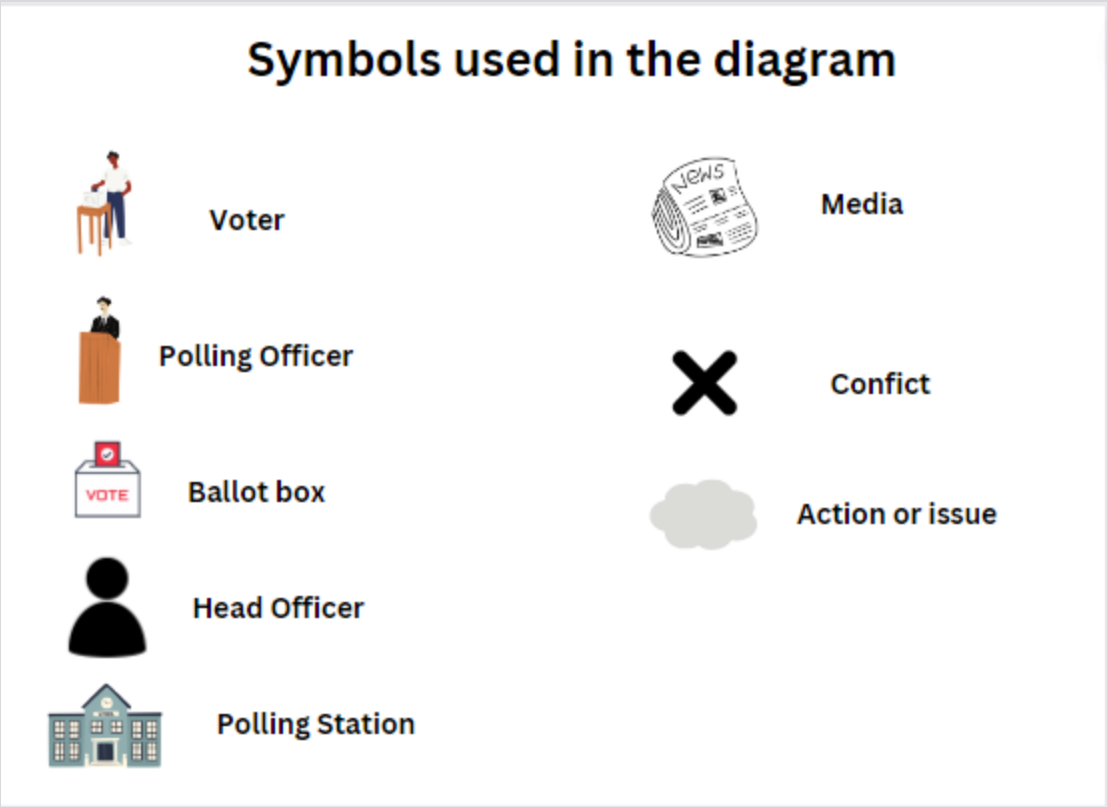


Figure 8.2 Symbols in Diagram

8.1.2 Description

Generally, to Perform proper elections there are many things that should be looked after. The Election Commission must arrange Proper polling stations, polling officer, Security officers, and other resources necessary for performing the elections. The diagram illustrates evident mismanagement within different facets of the traditional voting system. Polling officers face challenges in coordinating with the Head Office to meet resource requirements, leading to issues in time and resource management. The requirement for voters to travel to polling stations, often located at a distance, results in time wastage. Moreover, the simultaneous arrival of many people at polling stations makes maintaining order challenging. Human errors during the counting process contribute to miscounts, and there's a risk of cheating in the traditional approach. The manual counting of votes is time-consuming, and the generation of an authentic result for media dissemination takes considerable time. To address these challenges, a unified system can streamline the entire process. With voters able to cast their votes from their mobile or computer, eliminating the need for physical presence at polling stations, the system can automatically tally votes, minimizing the risk of miscounts. The application-driven approach eliminates the need for polling officers, security personnel, and other resources, simplifying the election management process and reducing costs.

8.1.3 Key Issues

The Current election system is not manual due to which there are many issues faced during election.

- Time Wastage of voters because they must go to polling stations in distant areas and then waiting in lanes for their turn.
- Time and resource management challenges for the polling officers.
- The manual counting of votes is time-consuming and there may be miscounting in it.
- There may be difficulty in maintaining the decorum of polling station due to large number of people.
- Delays in generating authentic results due to manual counting and recounting at the head office.
- High cost of salaries of polling officers, security officers, head officers and other resources.
- Lack of efficiency in traditional voting system.
- Overseas voters cannot vote in the traditional voting system.
- There are risks of cheating and fraudulent activities in the traditional voting approach.

8.1.4 Conflicted Areas

There remain some conflicted areas in the traditional voting process. Some of them are given below:

- There is mismanagement of resources and responsibilities by the Polling officer.
- There is also time wastage of voters and election officers due to manual processes.
- People who live abroad are not able to vote because of no online system availability.
- Risks of cheating and fraudulent activities in the traditional voting approach.
- Voters must carefully vote to avoid wasting their votes due to human error.
- There are conflicts over the legitimacy of election results.
- It takes a lot of time to compile and issue authentic results.
- There is disagreement among authorities over polling station’s locations.
- There are disputes over accuracy and fairness of the counting process.
- There may arise problems of voters’ disputes in the polling stations.
- Discrepancies in voter registration and identification

8.2 Organizational Structure

An e-voting system's organizational structure often includes several stakeholders and organizations responsible for various components of the election process. An election management organization, which can be a government agency, an independent commission, or a corporate corporation entrusted with organizing and administering elections, governs an e-voting system at its core. This body oversees the entire election process, which includes voter registration, candidate nomination, ballot production, and results tabulation.

8.3 Project Stakeholders

Individuals within a corporation who play critical roles in the development and implementation of a software project are known as stakeholders. Several stakeholders will be involved in the 'E-voting' initiative, which intends to automate voting processes, including:

Owner

The ultimate owner of the project provides financing for its development.

Project Manager

In charge of ensuring that the software development process runs well.

Chief Accountant

In charge of making project-related investment decisions.

Authority

Oversees all E-voting application procedures

System User

Individuals who utilize the application, provide input, and interact with the framework

8.4 Business/Specific Requirements' Elicitation

A variety of methods and techniques were employed to gather project specifications from diverse users. These methods and techniques include:

- Interviews
- Document Analysis
- Feedback Mechanism
- Prototyping
- Workshops

8.4.1 Interview Analysis

The analysis of E-voting project interviews yielded useful insights and information. Several critical aspects were determined through conversations with local farmers, interested persons, and experienced automation experts. Users emphasized the importance of an accessible and user-friendly interface to guarantee that the application is usable by a wide range of individuals, including those inexperienced with technology. Security and data protection were high priorities, emphasizing the significance of strong encryption and authentication measures. Furthermore, interviewees made useful suggestions for

improvement, which will be considered as the project progresses. Overall, the interviews were critical in establishing the project's specs and ensuring that they aligned with user needs.

8.4.2 Document Analysis phase Summary

The document analysis phase summary for the E-voting project reveals important insights. This phase entailed doing a thorough evaluation of academic publications and scientific research articles relevant to the project's topic. The purpose was to identify existing research work and solutions, as well as any literature gaps, to prevent duplicating earlier efforts. The relevance of the project's uniqueness and its ability to address deficiencies in the old paper-based voting system was emphasized in the document analysis. The initiative intends to improve openness, accessibility, and data security in the electoral process by combining blockchain technology and secure electronic voting. This phase laid the groundwork for the project's research and development.

8.4.3 Feedback Mechanism

The project made use of a well-structured feedback process, which allowed users and stakeholders to submit regular feedback and suggestions throughout the project's growth. This iterative process enabled the modification of project parameters in response to changing demands and input, ensuring that the result effectively fulfilled user expectations.

8.4.4 Prototyping

The project used prototypes to bridge the gap between conceptual concepts and tangible requirements. To provide stakeholders with a visual depiction of the envisioned system, low-fidelity mockups and prototypes were quickly generated. This technique was extremely useful in clarifying and validating project specs since it gave stakeholders a clear knowledge of how the system would function and appear.

8.4.5 Workshops

Interactive workshops involving key stakeholders played a pivotal role in gathering project specifications. These collaborative sessions served as platforms for brainstorming, requirements prioritization, and consensus building. By bringing together a diverse range of perspectives and expertise, workshops facilitated in-depth discussions and ensured that project specifications were comprehensively defined and aligned with the project's overall vision.

8.5 Identification and Description of a specific Problem Area

Several concerns have been uncovered after analyzing the current voting system management process. These are some of the issues:

Voter Registration Inefficiency

The existing voter registration process is frequently manual and time-consuming, resulting in mistakes in voter lists and the exclusion of qualified voters.

Paper Ballots and human Counting

Relying on paper ballots and human vote counting systems can result in errors, delays, and disagreements over election outcomes.

Limited Accessibility

Traditional voting procedures may not be accessible to all citizens, especially those with disabilities or living in rural places.

Voter Identification Difficulties

Ensuring accurate voter identification and preventing fraudulent voting can be difficult in the absence of adequate identity verification measures.

Security Vulnerabilities

Traditional voting methods are vulnerable to a variety of security issues, including ballot tampering, voter impersonation, and cyberattacks.

Openness

A lack of openness in the voting process, such as the inability to verify votes and follow ballot processing, can erode public faith.

High Costs

Due to the printing of ballots, the hiring of poll workers, and the physical infrastructure needs, organizing and maintaining elections via traditional techniques can be costly.

Voter Turnout

Increasing voter turnout and engagement remains an issue, with some eligible citizens choosing not to vote owing to inconvenience.

Manual Data Entry Errors

When transferring voter information from registration records to the voting system, data entry errors can lead to discrepancies and conflicts

8.6 Solutions

Here are some potential solutions to the stated voting system management issues:

Voter Registration Made Simple

- Implement an online voter registration system to increase accuracy and streamline the registration process.
- To confirm the legitimacy of voter registrations, use data verification procedures such as cross-referencing with national ID databases.

Paperless voting and automatic tally

- Transition to electronic voting (e-voting) systems, which allow voters to cast their ballots electronically rather than on paper ballots.
- Automated vote counting methods should be used to assure accuracy and speed up results.

Enhancement of Accessibility

- Make voting alternatives for people with disabilities more accessible, such as electronic ballots with screen readers and tactile interfaces.
- To reach voters in remote or underserved locations, set up mobile voting stations.

Strong Voter Identification

- Use secure voter identification mechanisms such as biometric authentication (for example, fingerprint or face recognition) or smart card-based systems.
- To avoid fraudulent voting, conduct identification verification checks at polling places.

Security Procedures

- To safeguard the e-voting system, use blockchain technology to ensure transparency, immutability, and resistance to tampering.
- To ensure the integrity of the voting process, use encryption, multi-factor authentication, and intrusion detection technologies.

Verification and transparency

- Allow voters to confirm their ballots using a safe and anonymous verification system.
- Keep a public record of all transactions (votes) for auditing purposes, allowing for independent verification of results.

Cost-cutting measures

- By using e-voting technology, you can save money on paper ballots, printing, and human operations.
- Investigate cost-sharing arrangements with technology suppliers or public-private partnerships.

Voter Participation

- Voter education and awareness efforts should be implemented to boost participation and emphasize the importance of voting.
- To meet varied voter preferences, provide multiple voting choices such as early voting, mail-in voting, and electronic voting.

Automation of data entry

- Integrate voter registration and voting technologies to automate data transfer and reduce errors caused by human data entry.
- Implement data validation tests to detect and repair errors in real time.

8.7 System Requirements with Specifications

The characteristics of the project have been selected after a thorough assessment of the market. Various requirement collection strategies were used to gather specific project requirements. It is vital to highlight that among the specified needs, two separate categories exist, as follows:

8.7.1 Functional Requirements

The components of the application that directly involve user interaction and the use of its visible features are referred to as functional requirements. The following are the functional requirements of the 'E-voting' system:

- Voter’s authentication and authorization
- Voters can vote in elections
- Voter can view published results
- Admin can log in through email & password
- Admin can add elections
- Admin can add candidates to election
- Admin can add constituencies
- Admin can generate election report
- Admin can view published report
- Generate automatic election reports
- The system should display the system’s functionalities according to their roles
- They can access the system using the internet

8.7.2 Non-Functional Requirements

A non-functional requirement is a specification that describes a system's operating capabilities and constraints, hence improving its functionality. These aspects can include things like speed, security, and dependability. The non-functional needs of the 'E-voting' system are listed below:

- To protect voter information, election outcomes, and system configurations, the system must use encryption and access controls.
- To prevent unauthorized access, both voters and administrators must have reliable authentication procedures in place.
- During the busiest voting periods, the system must manage many concurrent users while preserving performance and responsiveness.
- The system shall have high availability to ensure that voters should be able to access the e-voting system without experiencing any substantial disruption during elections
- Voters shouldn't encounter delays when dealing with the system because they should respond promptly.

8.8 Requirements' Prioritization Through MoSCoW Analysis

RQ_1: Voters' authentication and verification

RQ_2: Admins' authentication and verification

RQ_3: Creating up elections

RQ_4: Adding of voters/candidate in election

RQ_5: Adding of constituencies

RQ_6: Voters cast vote

RQ_7: Display results

RQ_8: Generation of election report

RQ_9: Admin can view published report

RQ_10: Candidate can view published report

RQ_11: Log out

RQ_12: Mobile app for voters to cast their votes

RQ_13: Multiple Language support

RQ_14: Voter Feedback Mechanism

RQ_15: Voice Voting

RQ_16: Offline Voting

RQ_17: Customize User Interface

8.8.1 Must have Requirements

Table 12 Must have Requirements

ID	Requirements
RQ_1	Voters' authentication and verification
RQ_2	Admins' authentication and verification
RQ_3	Creating up elections
RQ_4	Adding of voters/candidate in election
RQ_5	Adding of constituencies
RQ_6	Voters cast vote

8.8.2 Should Have

Table 13 Should Have Requirements

ID	Requirements
RQ_7	Display Results
RQ_8	Generation of election report

RQ_9	Admin can view results
RQ-10	Voters can view results
RQ_11	Log out

8.8.3 Could Have

Table 14 Could Have Requirements

ID	Requirements
RQ_12	Mobile app for voters to cast their votes
RQ_13	Multiple Language support
RQ_14	Voter Feedback Mechanism

8.8.4 Won’t Have

Table 15 Won't Have Requirements

ID	Requirements
RQ_15	Voice Voting
RQ_16	Offline Voting
RQ_17	Customize User Interface

8.9 Requirements Catalogue

The offered set of need specifications contains both functional and non-functional specifications. These specifications were carefully chosen to build this system. We will construct a catalogue of these specifications in this section.

Table 16 Voters' Authentication and Authorization

Source: Voter	**Sign off: Voter**	**Priority: Must Have**	**Requirements ID: RQ_1**
Functional Requirements: Voters' authentication and verification Voter should be able to log in to the system and authenticate and approve themselves.			
Non-Functional Requirement (s)			
Description	Target Value	Acceptable Range	Comments
The amount of login attempts per day should be restricted.	35 Per Day	50 per day	Users will be able to see who else is using the gadget.

Table 17 Admins' Authentication

Source: Admin	**Sign off: Admin**	**Priority: Must Have**	**Requirements ID: RQ_2**
Functional Requirements: Admins' authentication and verification Admin should be able to log in to the system			
Non-Functional Requirement(s)			
Description	Target Value	Acceptable Range	Comments
The amount of login attempts per day should be restricted	35 Per Day	50 per day	Admin will be able to see who else is using the gadget

Table 18 Creating up Elections

Source: Admin	**Sign off: Admin**	**Priority: Must Have**	**Requirements ID: RQ_3**
Functional Requirements: Creating up Elections The admin shall be able to create election			
Non-Functional Requirement(s)			
Description	Target Value	Acceptable Range	Comments
Online Mode	100 per day	120 per day	N/A

Table 19 Adding voters/candidate in elections

Source: Admin	**Sign off: Admin**	**Priority: Must Have**	**Requirements ID: RQ_4**
Functional Requirements: Adding voters/candidate in elections The system shall be able to provide the facility to admin to add candidate in elections			
Non-Functional Requirements			
Description	Target Value	Acceptable Range	Comments
Online Mode	500 per day	600 per day	N/A

Table 20 Adding of constituencies

Source: Admin	**Sign off: Admin**	**Priority: Must Have**	**Requirements ID: RQ_5**
Functional Requirements: Adding of constituencies			

The system shall be able to provide the facility to admin to add constituencies			
Non-Functional Requirement(s)			
Description	Target Value	Acceptable Range	Comments
Online Mode	2	2	N/A

Table 21 voters cast Votes

Source: Voter	**Sign off: Voter**	**Priority: Must Have**	**Requirements ID: RQ_6**
Functional Requirements: Voters cast their votes The system shall be able to provide the facility to voters to cast their votes in election			
Non-Functional Requirement (s)			
Description	Target Value	Acceptable Range	Comments
Online Mode	1	1	1 vote in one election

Table 22 Display Results

Source: Admin	**Sign off: Admin**	**Priority: Should Have**	**Requirements ID: RQ_7**
Functional Requirements: Display Results The system shall be able to provide the facility to admin to display results of election			
Non-Functional Requirements			
Description	Target Value	Acceptable Range	Comments

Online Mode	500 Results Per day	600 per day	N/A

Table 23 Generation of Result Report

Source: Admin	**Sign off: Admin**	**Priority: Should Have**	**Requirements ID: RQ_8**
Functional Requirements: Generation of Reports The system shall be able to provide the facility to admin to generate report of results			
Non-Functional Requirement(s)			
Description	Target Value	Acceptable Range	Comments
Online Mode	500 per day	600 per day	N/A

9 New System Architecture and Design

In this section, we will delve into the system architecture of the 'E-voting' application and its design process. Various types of diagrams were generated to facilitate the construction and deployment of this architecture. In the upcoming section, we will discuss these diagrams in the context of the e-voting application.

9.1 Use Case Diagram

The e-voting system's use case diagram, shown in Figure 7, illustrates the system's numerous interactions and features from the viewpoints of its two main actors, the "Voter" and the "Administrator." Beginning with the "Login" action, which permits safe access to the system through authentication with specified credentials, the "Voter" actor is endowed with various essential use cases. The "View Ballots" use case thereafter enables the Voter to browse the ballots that are available for upcoming elections, while the "Vote" use case enables the Voter to choose a ballot, cast a vote, and confirm that the vote was submitted. The "View Results" use case allows the voter to examine the published election results after participating in the voting process, promoting transparency and trust in the result.

The "Administrator" actor, on the other hand, possesses a collection of use cases that are exclusive to system administration. The Administrator begins their journey with the "Login" use case, much like the Voter does. They get access to the system via email and password authentication. The "Manage Elections" use case perfectly captures the Administrator's crucial job, giving them the power to add, alter, or cancel elections as needed.

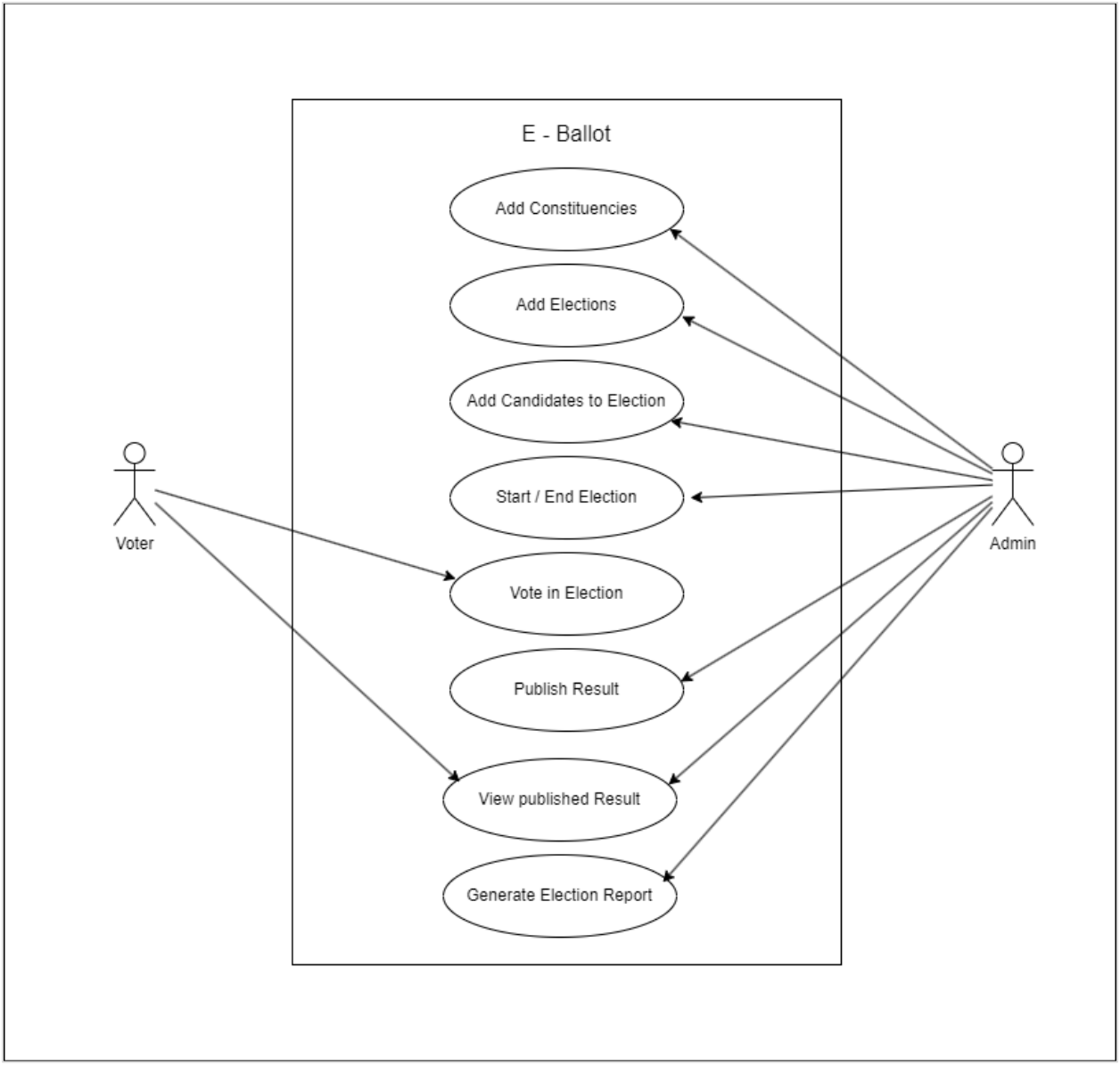


Figure 9.1 Use Case Diagram

9.2 Use Case Documentation

Table 24 Use Case Documentation

9.3 Use Case ID	9.4 Primary Actor	9.5 Use Case
UC_1	Voter	Vote in Election
UC_2	Voter Admin	View published result

UC_3	Admin	Add constituencies
UC_4	Admin	Add Election
UC_5	Admin	Add Candidates to Election
UC_6	Admin	Start / End Election
UC_7	Admin	Publish Result
UC_8	Admin	Generate Election Report

9.5.1 Use Case Scenarios

Table 25 UC_1

Use Case ID:	**UC_1**
Use Case Name:	**Vote In Election**
Actors:	Voter
Pre-conditions:	• An election is currently open and available for the voter to participate in. • The voter has not already cast their vote in the current election.
Normal Flow:	• the voter sees a list of available elections in which he/she is eligible to vote. • Once a selection has been made, the system displays a list of candidates/options to be chosen. • voter has to assess the list and choose. • The voter confirms their selection. • The voter's vote for their chosen candidate is recorded by the system. • Such changes include the voter's voting status in the current election. • The message is confirming the successful casting of the vote, and it is displayed by the system.

Alternative Paths:	• When a voter tries to vote on a closed or unavailable election, the system sends an error message that indicates that the selected election is not open for voting. • The system realizes that the voter has already voted in this current election, thus a message comes up informing the voter that he/she had already voted. The vote is not counted two times.
Post-Conditions	• The voter's vote has been recorded for the selected candidate. • The voter's status is updated to reflect that they have participated in the current election

Table 26 UC_2

Use Case ID:	**UC_2**
Use Case Name:	**Vote Published Result**
Actors:	Voter, Admin
Pre-conditions:	• At least one election has concluded, and the results have been uploaded by the admin. • The user has the appropriate role and permissions to view election results.
Normal Flow (Voter)	• This is when the voter chooses which election, he or she wants to view the results of. • The system shows election results – names of candidates/options and their respective vote count. • The Voter Reviews the Results Of The Election.

Normal Flow (Admin)	• The system authenticates the admin's credentials. • The admin is presented with administrative options such as viewing published election results. • The admin clicks the option to see the outcome of the election. • The list displayed by the system includes elections completed but with results made available. • Admin chooses the election whose outcome he wants to see. • Results of the election are shown on the screen, displaying the names of candidates or options and the number of votes each has.
Alternative Paths:	• there are no elections with published results available, the system informs the user that there are currently no election results to view.
Post-Conditions	• The voter or admin has successfully viewed the election results for the selected election.

Table 27 UC_3

Use Case ID:	**UC_3**
Use Case Name:	**Add constituencies**
Actors:	Admin
Pre-conditions:	• The admin is successfully logged into the E-voting system. • The admin has permissions to add and manage constituencies
Normal Flow (Admin)	• The admin logs on the E-voting system using the provided access credentials. • The system checks the admin's login details. • The admin has the ability to manage constituencies in this system.

	• The admin chooses the add constituency option. • The admin enters a form displayed by the system with such information as the name and location of the new constituency. • The admin writes in the required information in the form. • The admin fills it and submits it to add the new constituency. • This system verifies information provided by the admin and acknowledges the addition of the new constituency. • The system adds the new constituency to the list of constituencies.
Alternative Paths:	• The admin re-enters the credentials when system displays an error message • The system displays an error message that the user doesn't have permission to perform this action
Post-Conditions	• The admin has successfully added a new constituency to the system. • The new constituency is available for use in upcoming elections.

Table 28 UC_4

Use Case ID:	**UC_4**
Use Case Name:	**Add Election**
Actors:	Admin
Pre-conditions:	• The admin is logged into the system • The admin must have permission to add and manage election • Constituencies must have been already added into the system
	• The admin logs into the system • The system validates the admin's login information. • The system displays the admin witg options, including

Normal Flow (Admin)	the ability to manage elections. • The admin selects the option to create a new election. • The system displays a form for the admin to enter the details of the new election • The admin enters the details in the form and submit it • The system validates the information provided by the admin and displays a success message
Alternative Paths:	• System display an error message if the user don't have permission to perform this action
Post-Conditions	• The election added into the system

Table 29 UC_5

Use Case ID:	**UC_5**
Use Case Name:	**Add Candidate to Election**
Actors:	Admin
Pre-conditions:	• The admin is successfully logged into the E-voting system. • The admin must have permissions to manage candidates and elections. • One election has been created in the system.
Normal Flow (Admin)	• The admin logs into the system. • The system validates the admin's login information. • The system displays the admin with administrative options. • The admin selects the option to add a candidate to an election. • The system displays a list of available elections. • The admin selects the election to which they want to add a candidate.

	• The system displays a form for the admin to enter the candidate's details • The admin enters the details in the form with. • The admin submits the form to add the candidate to the selected election. • The system displays the confirmation message
Alternative Paths:	• the system display an error message if the user does not have the permissions to perform this action.
Post-Conditions	• The admin has successfully added the candidate in the election • The candidate is associated in the election will be able to receive votes from the voters

Table 30 UC_6

Use Case ID:	**UC_6**
Use Case Name:	**Start / End Election**
Actors:	Admin
Pre-conditions:	• The admin must be logged into the system • The admin has the permission to perform this action • One election has been created in the system.
Normal Flow (Start an Election)	• The admin accesses the E-voting system and logs in using their provided credentials. • The system validates the login information. • The system displays the admin with administrative options. • The admin clicks the option to start an election. • The system displays a list of available elections that have not yet started. • The admin clicks on the election they want to start. • The system updates the status of the election to "In

	Progress”
Normal Flow (END an Election)	• The admin logs into the system • The system validates the admin's login credentials. • The system presents the admin with administrative options. • The admin clicks the option to end an election. • The system displays a list of elections that are currently in progress. • The admin clicks on the end option to end the election. • The system updates the status of the selected election to "Completed
Alternative Paths:	• if a user with limited administrative permissions attempts to access the election management function during steps 3-4 of the normal flow, the system displays an error message indicating that the user does not have the necessary permissions to perform this action. • The system shall display the message there are no election in the system
Post-Conditions	• The voters can cast their vote as election is in progress • The admin has permission to manage this election • The status of election is updated to mark as complete, and no one can cast the vote • The election results can be viewed by both admin and voter

Table 31 UC_7

Use Case ID:	**UC_7**
Use Case Name:	**Publish Results**
Actors:	Admin
Pre-conditions:	• The admin is successfully logged into the E-voting system. • The admin has the appropriate permissions to manage elections and publish results.

	• At least one election has been completed and marked as "Completed" in the system.
Normal Flow (Start an Election)	• The admin accesses the E-voting system and logs in using their provided credentials. • The system validates the admin's login information. • The system presents the admin with administrative options, including the ability to manage elections. • The admin selects the option to publish election results. • The system displays a list of completed elections that are eligible for result publication. • The admin selects the election for which they want to publish results. • The admin confirms their decision to publish the results. • The system generates official election reports summarizing the results. • The system makes the election results available for viewing by both voters and the admin. • The system updates the status of the election to indicate that the results have been published.
Alternative Paths:	• If a user with limited administrative permissions attempts to access the election results publication of the normal flow, the system displays an error message indicating that the user does not have the necessary permissions to perform this action. • If, there are no completed elections eligible for result publication, the system informs the admin that there are currently no elections with results to publish. • If, the system encounters an error while generating the official election reports, it displays an error message and prompts the admin to try again later or contact system support.
Post-Conditions	• The election results for the selected election are published and available for viewing. • Voters and the admin can access and view the official

	election reports. • The election's status is updated to "Results Published.

Table 32 UC_8

Use Case ID:	**UC_8**
Use Case Name:	**Generate Election Report**
Actors:	Admin
Pre-conditions:	• The admin is successfully logged into the E-voting system. • The admin has the appropriate permissions to manage elections and generate reports. • At least one election has been completed and marked as "Completed" in the system.
Normal Flow (Start an Election)	• The admin accesses the E-voting system and logs in using their provided credentials. • The system validates the admin's login information. • The system presents the admin with administrative options, including the ability to manage elections and generate reports. • The admin selects the option to generate an election report. • The system displays a list of completed elections for which reports can be generated. • The admin selects the election for which they want to generate a report. • The admin specifies the type of report they want to generate (e.g., a summary report, a detailed report). • The admin confirms their selection to generate the report. • The system generates the requested election report. • The system makes the generated report available for download or viewing by the admin.

	• The admin can download or review the election report as needed.
Alternative Paths:	• If a user with limited administrative permissions attempts to access the election report generation function the normal flow, the system displays an error message indicating that the user does not have the necessary permissions to perform this action. • If, there are no completed elections eligible for report generation, the system informs the admin that there are currently no elections with results for which a report can be generated. • If, the system encounters an error while generating the requested election report, it displays an error message and prompts the admin to try again later or contact system support.
Post-Conditions	• The admin has successfully generated the requested election report. • The election report is available for download or viewing by the admin.

9.6 Class Diagram

The e-voting system's class diagram summarizes the essential classes that are essential to implementing the system's primary operations while adhering to the stated functional criteria. They are "Voter," "Constituency," "Administrator," "Election," and "Candidate."

The "Voter" class, which embodies attributes and procedures relevant to voter registration, authentication, and voting activity, represents system users. To vote and view election results, this class communicates with the "Election" class. It also communicates with the "Constituency" class, which establishes a voter's eligibility depending on the constituency in which they have registered.

Regions or locations where elections are held are represented by the "Constituency" class. It might have characteristics like the name of the constituency and its boundaries. In order to identify the constituency to which a voter belongs and to guarantee that voters can only participate in elections for their assigned region, the "Voter" class is coupled with the "Constituency" class.

The "Administrator" class represents the qualities and practices of system administrators who oversee overseeing elections, candidates, and other system settings. The system's functionality can be managed by administrators, who can also add or remove "Candidate" instances and manage "Election" instances.

The "Election" class, which has properties that describe specific election events like their names, dates, and statuses (such continuing or completed), is at the core of the electronic voting system. While the "Voter" class uses "Election" to cast votes and view election results, the "Administrator" class uses it to set up and run elections.

Candidates for political office are represented by the "Candidate" class in the context of an election. It contains information about the candidate's name, political party, and platform. This class is used by administrators to add, update, or remove candidate instances inside elections.

Overall, this class diagram shows the crucial connections and interactions between the key players in the electronic voting system as shown in Figure 9.2 below.

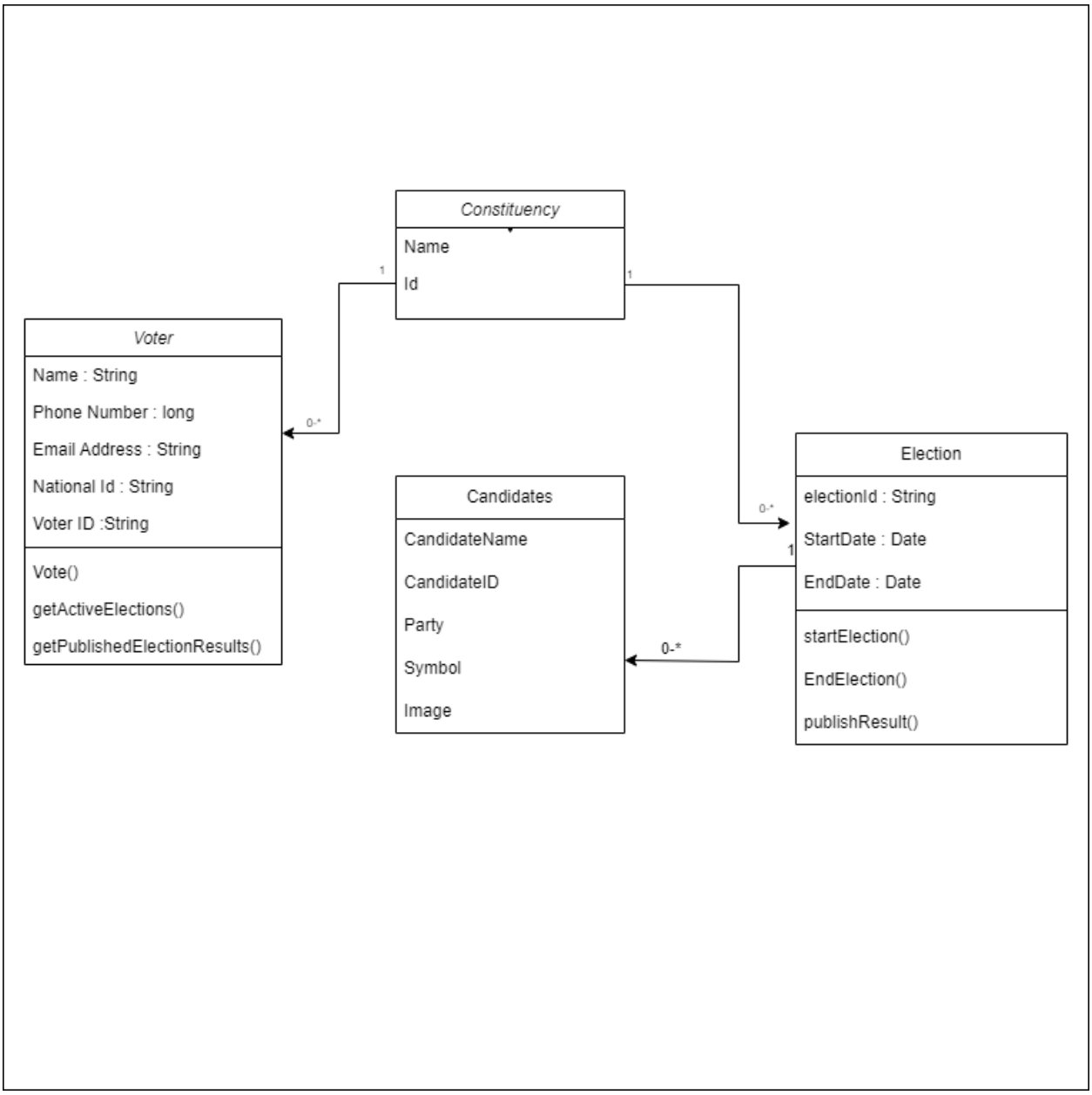


Figure 9.2 Class Diagram

9.7 Architecture Diagram

In architecture as shown in Figure 9.3, The User Layer, System Layer, and Blockchain Layer are the three main architectural pillars that support the e-voting system. Each of these levels is essential to how the system functions.

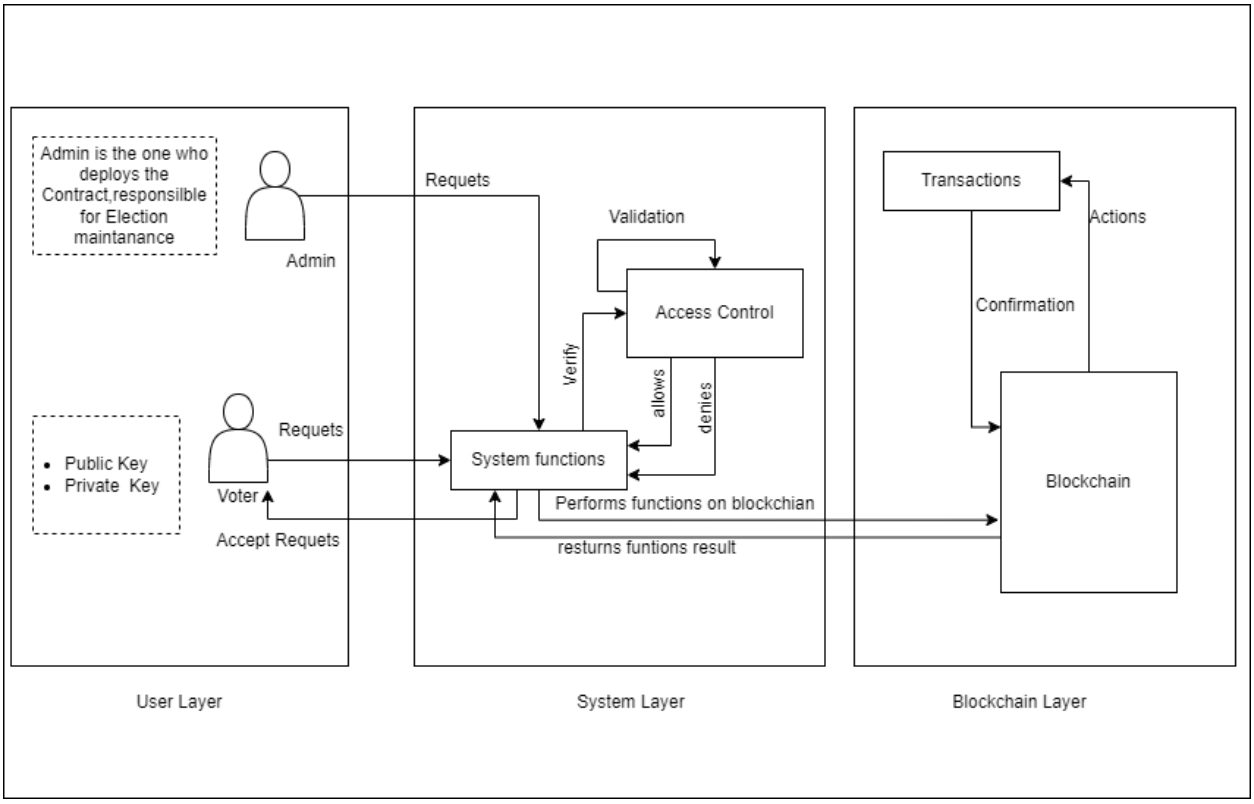


Figure 9.3 Architecture Diagram

9.7.1 User Layer

The "Admin" and the "Voter" are the two main characters included in the User Layer, which is at the top of the architecture. Both a public key and a private key, which are utilized for safe authentication and cryptographic activities, are in the possession of the "Voter". On the other hand, the "Admin" oversees implementing smart contracts, managing elections, and upkeep of the platform. Through a user interface designed specifically for their responsibilities, both users communicate with the system.

9.7.2 System Layer

The System Layer serves as a bridge between users and the underlying blockchain underneath the User Layer. Voters' and administrators' demands are received and handled by it. These demands frequently concern procedures like candidate selection, ballot access verification, and voter authentication. Assuring the validity of voting operations, confirming eligibility, and prohibiting unauthorized access to the blockchain layer are all crucial functions of the system layer. The System Layer starts the necessary Blockchain Layer processes after validating a user request.

9.7.3 Blockchain Layer

The e-voting system's base is the Blockchain Layer, which carries out the essential functions linked to voting and election management. Each action taken by a voter or administrator is represented as a

transaction in the system, which is built on a blockchain technology infrastructure. These transactions are processed by the blockchain layer, which also uses cryptographic techniques to confirm their authenticity and permanently stores them on the blockchain. The Blockchain Layer confirms the outcomes back to the blockchain after the action has been properly completed, assuring the accuracy and transparency of the voting process. Additionally, the Blockchain Layer oversees providing function results to the System Layer for further user communication, which may include vote confirmation or election outcome data.

In conclusion, blockchain technology is used in the well-structured architecture of the e-voting system to guarantee the security, transparency, and integrity of the election process. It enables a dependable and trustworthy electronic voting experience for both voters and administrators by integrating user interactions, system validations, and blockchain-based transactions.

9.8 Database Design

9.8.1 Entity Relationship Diagram

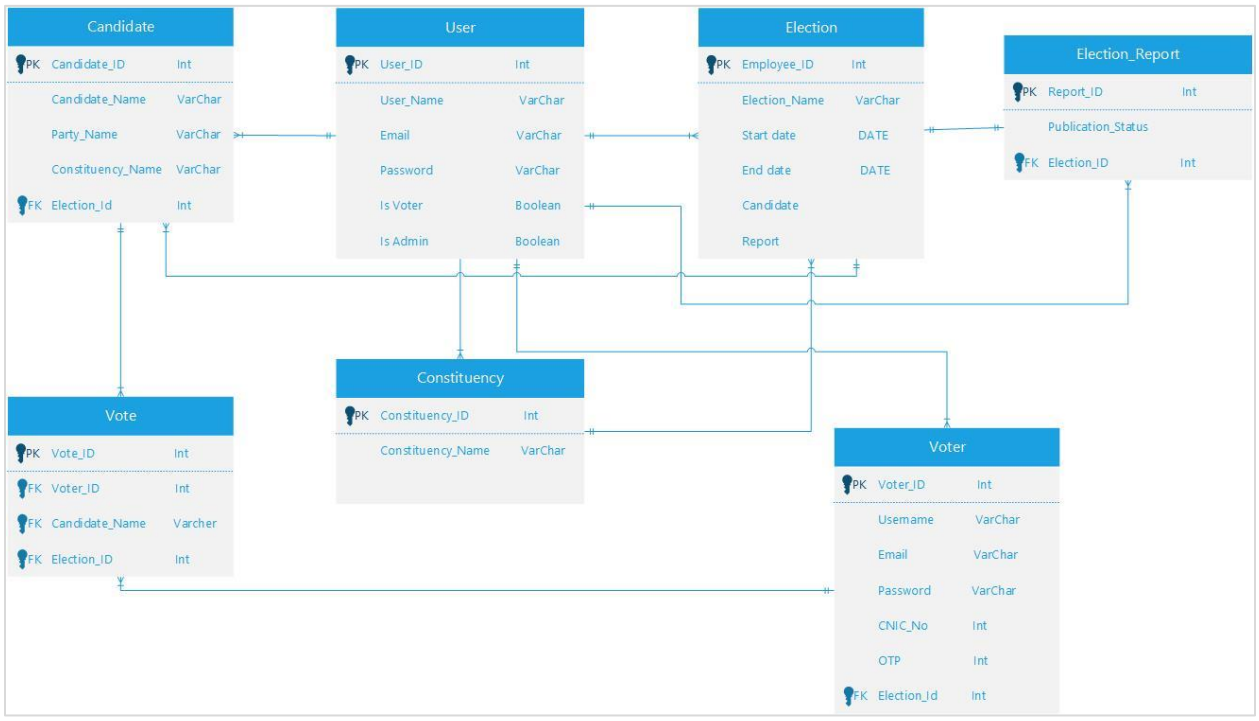


Figure 9.4 Entity Relationship Diagram

9.8.2 Database Mapping

Table 33 Database Mapping

User:	User ID, Username, Email, Password, Is Voter, Is Admin
Candidate:	Candidate ID, Candidate Name, Party Name, Constituency Name, Election ID
Vote:	Vote ID, Voter ID, Candidate Name, Election ID
Constituency:	Constituency ID, Constituency Name
Election:	Election ID, Election Name, Start Date, Ending Date, Candidate, Report
Voter:	Voter ID, Username, Email, Password, CNIC_No, OTP, Election ID
Election Report:	Report ID, Publication status, Election ID

9.9 Sequence Diagrams

9.9.1 Sign up (Voter)

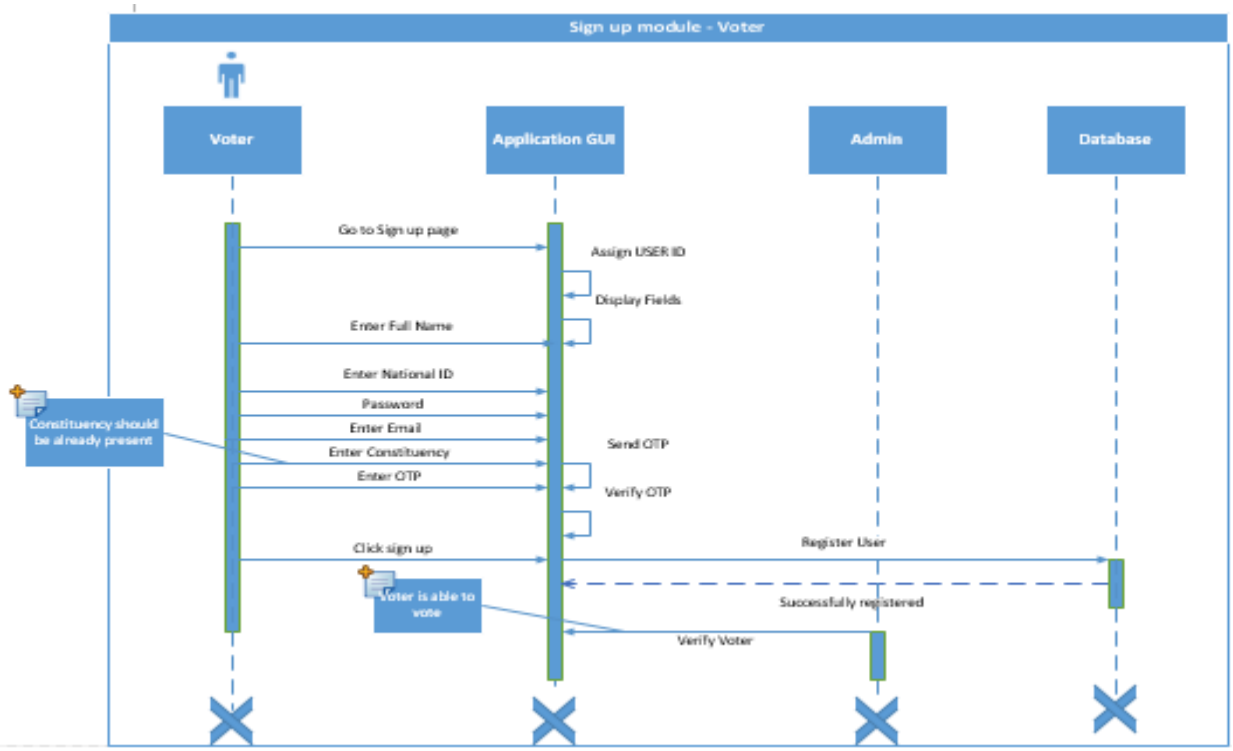


Figure 9.5 Sign up

9.9.2 Add Candidate (Admin)

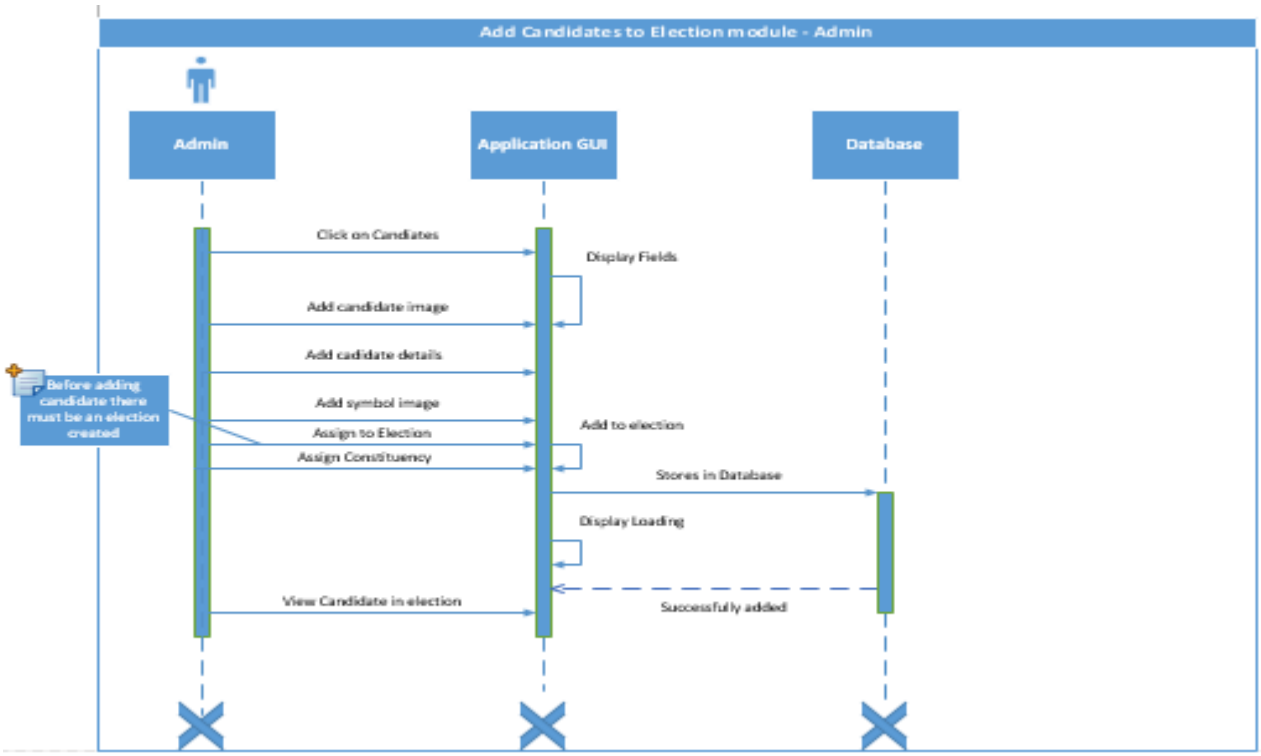


Figure 9.6 Add Candidate

9.9.3 Add Constituencies (Admin)

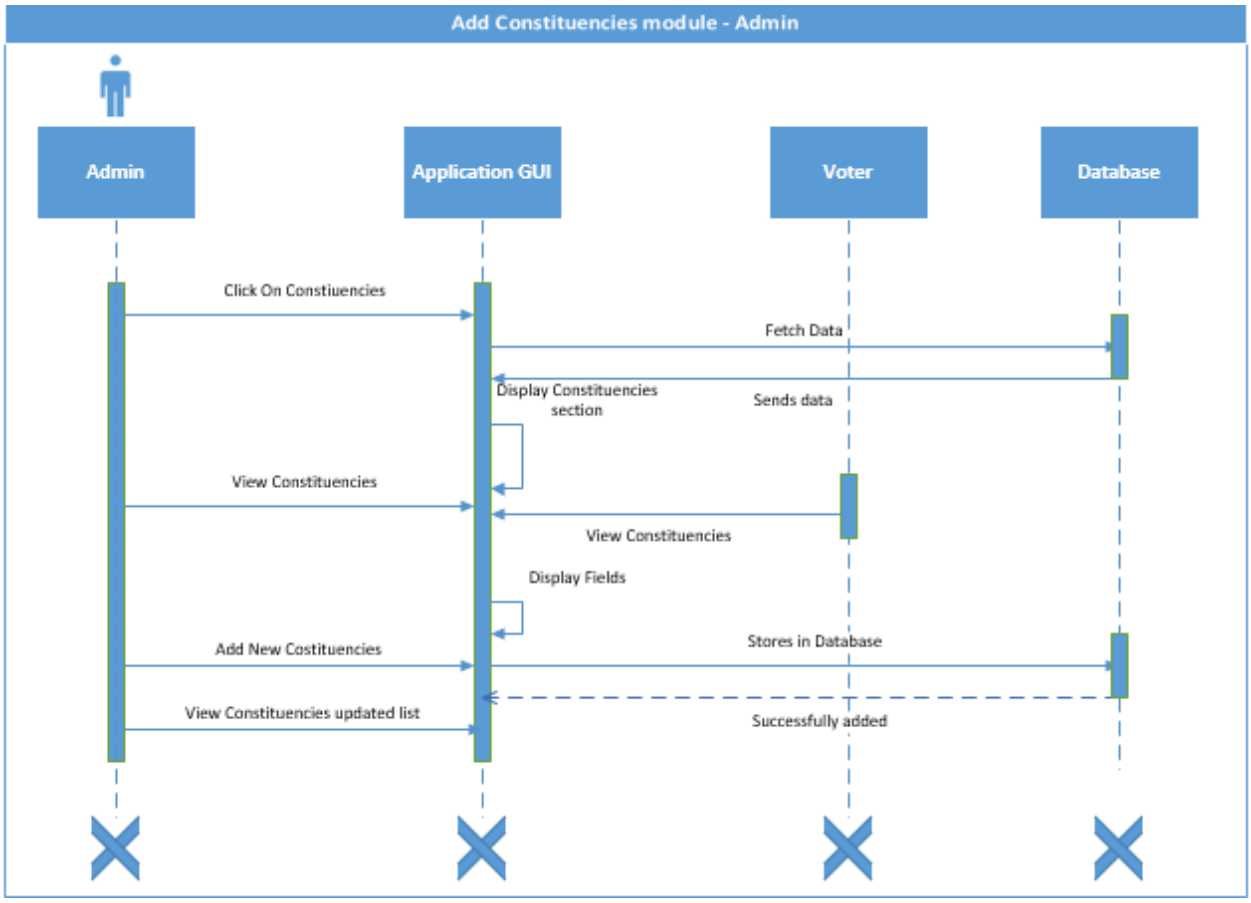


Figure 9.7 Add Constituencies

9.9.4 Add Election (Admin)

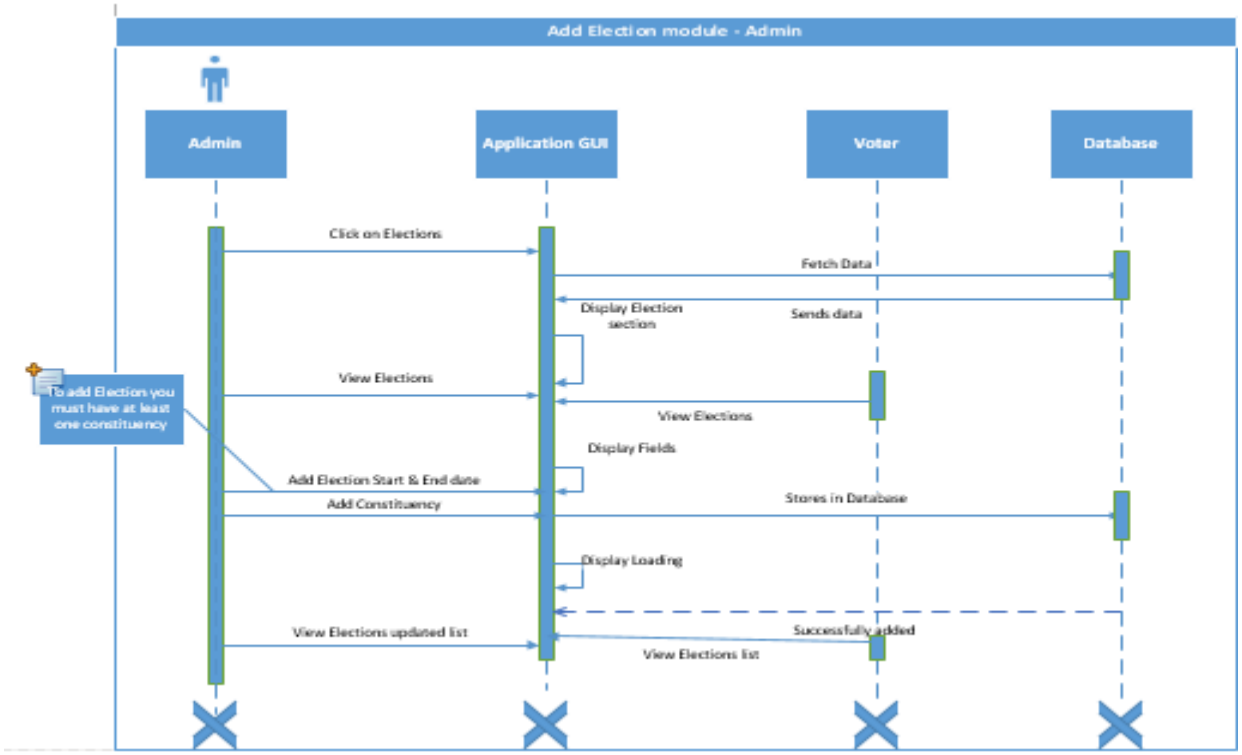


Figure 9.8 Add Election

9.9.5 Generate Election Report

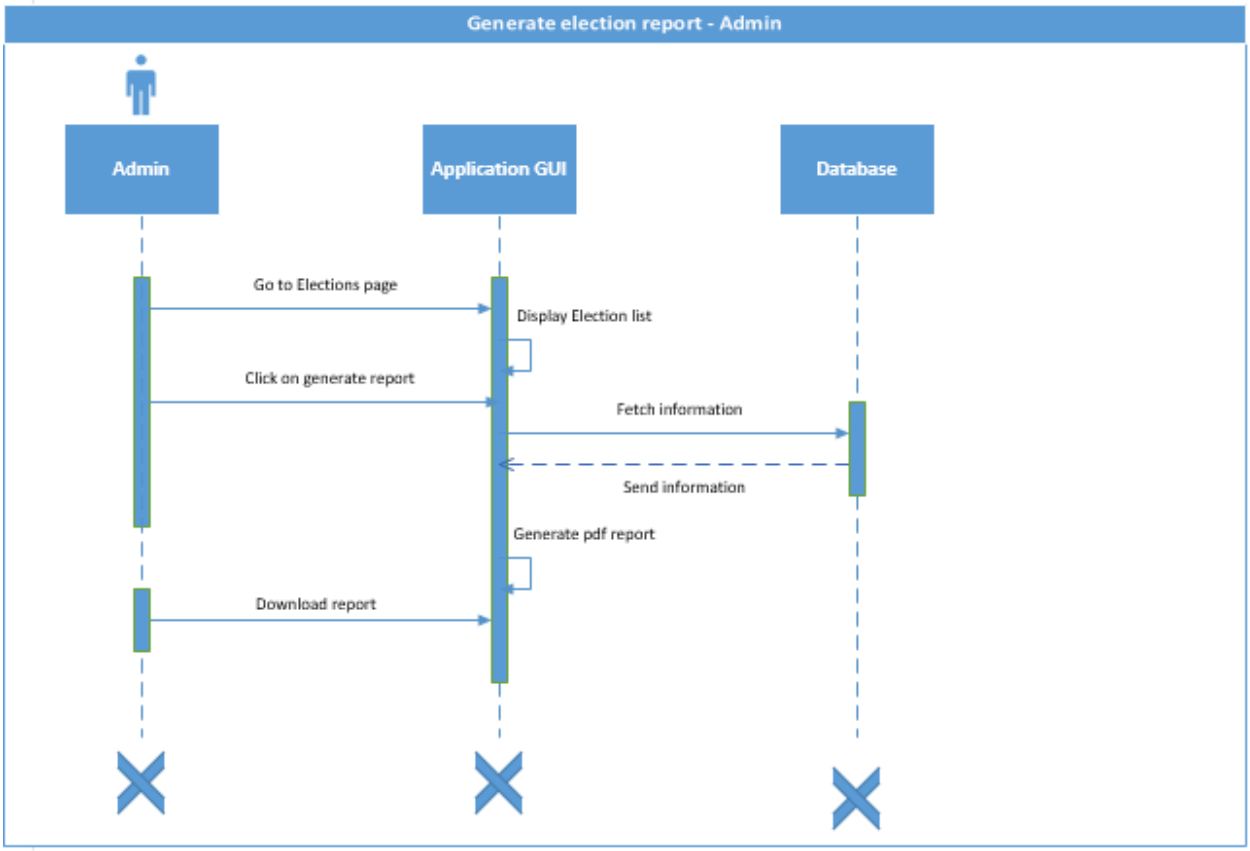


Figure 9.9 Generate Election Report

9.9.6 Publish Result (Admin)

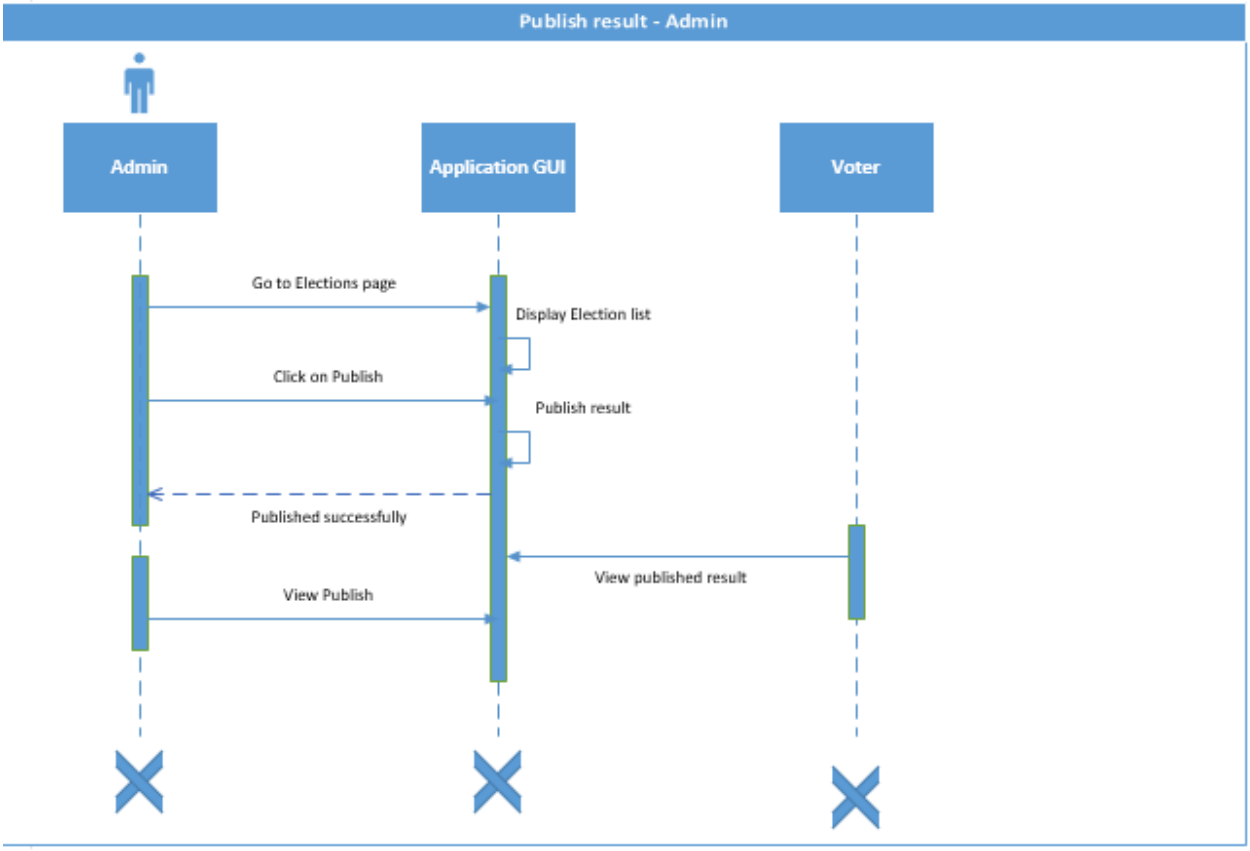
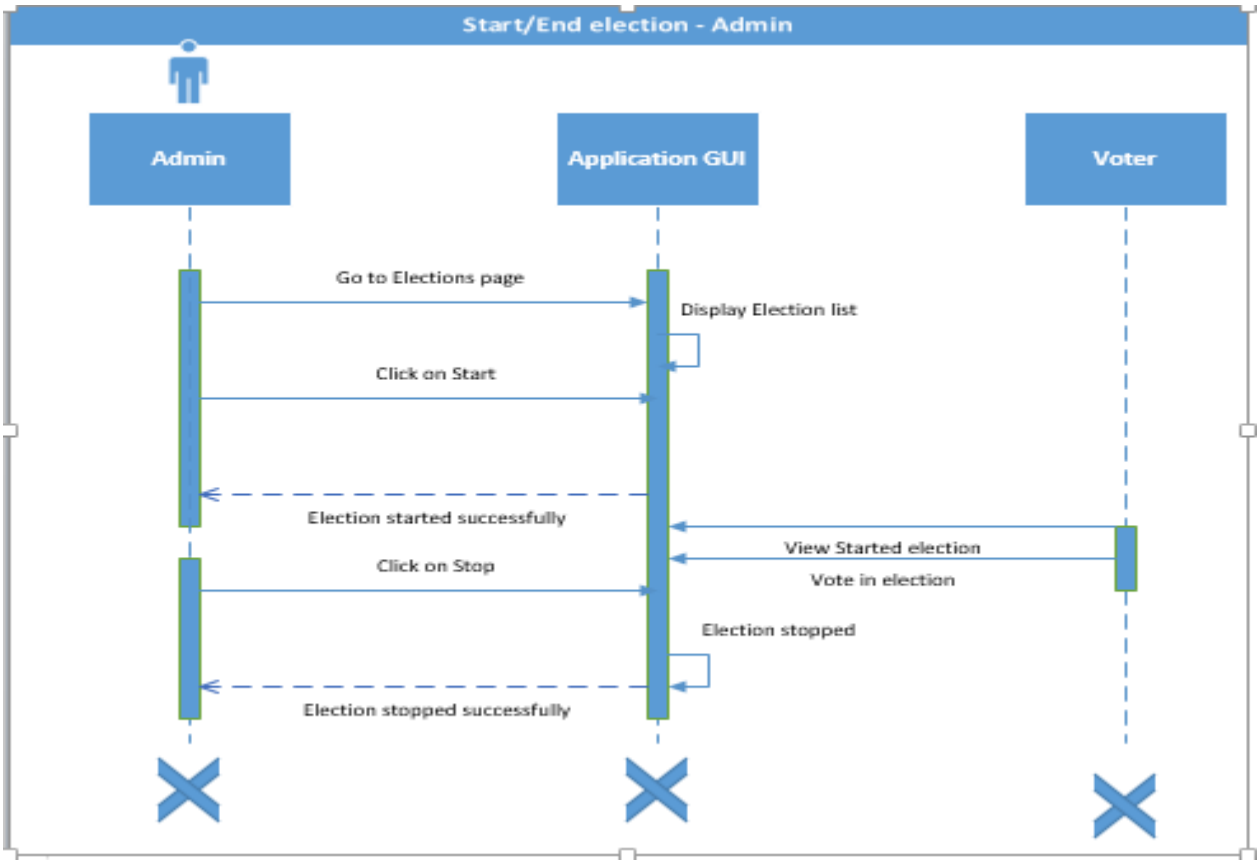
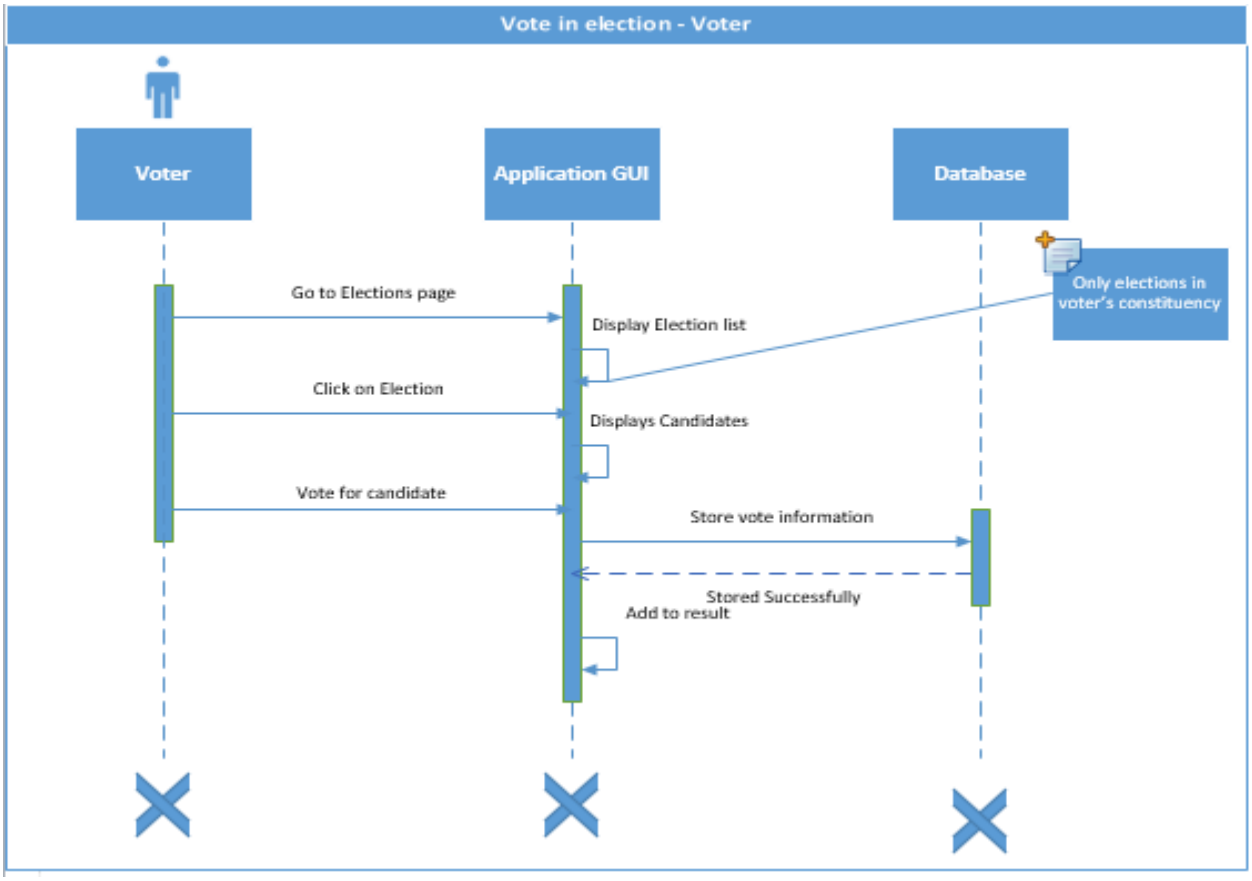


Figure 9.10 Publish Result

9.9.7 Start/End Election (Module)



9.9.8 Vote in Election (Voter)



9.10 Component Diagram

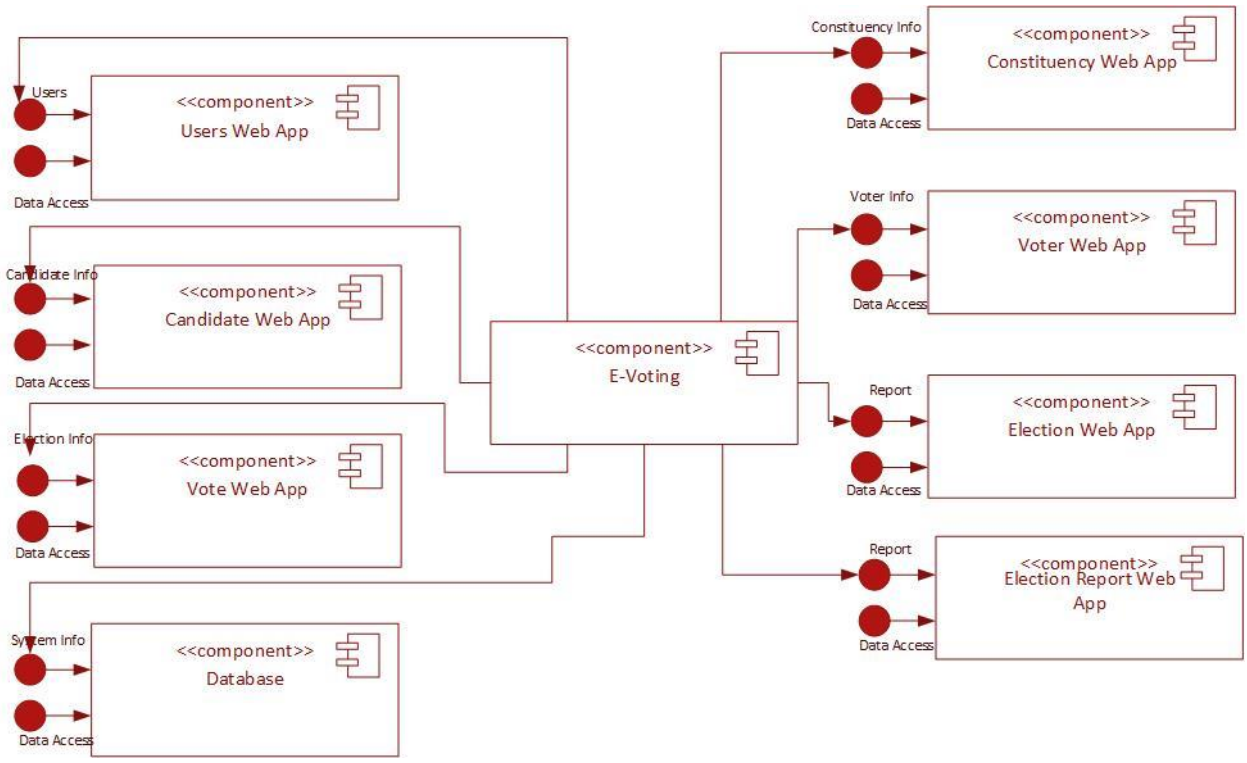


Figure 9.11Component Diagram

9.11 Deployment Diagram

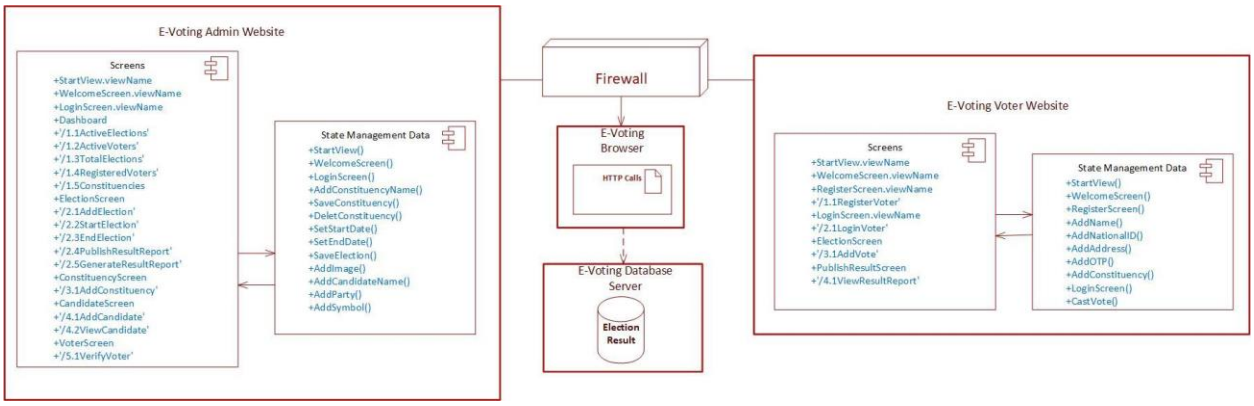


Figure 9.12 Deployment Diagram

9.12 Low Fidelity Wireframe for Prototype

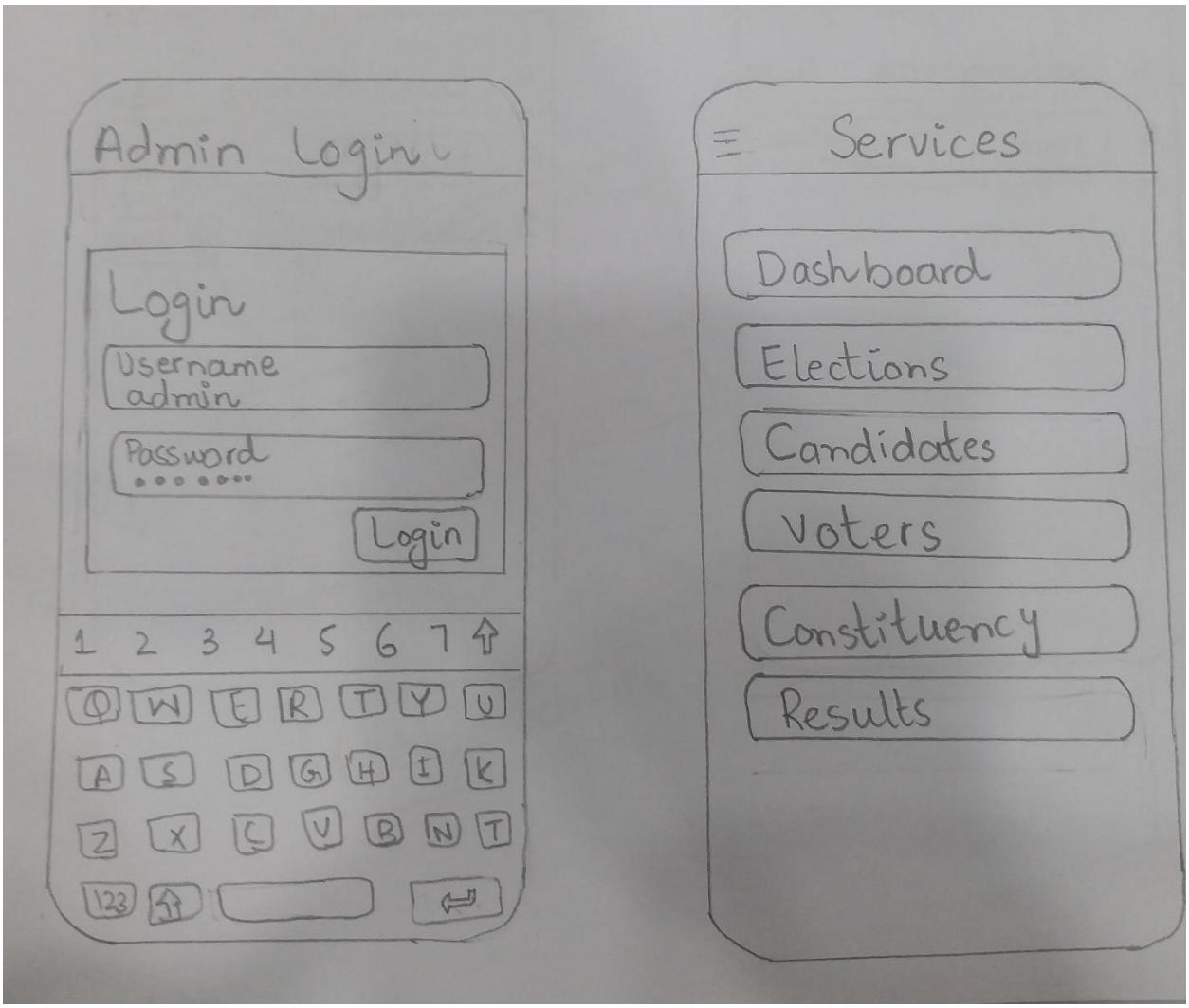


Figure 9.13 Admin Log in

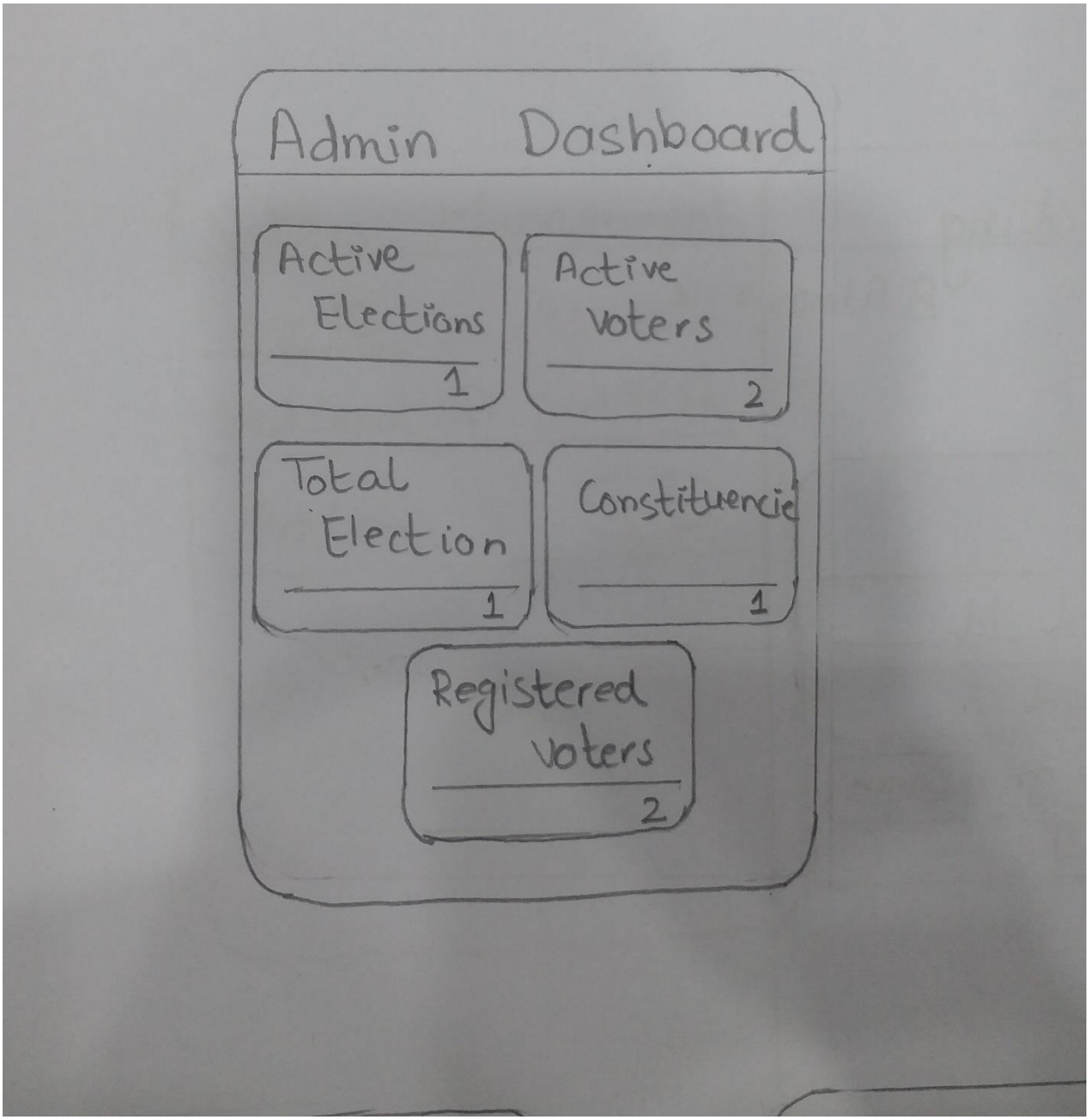


Figure 9.14 Admin Dashboard

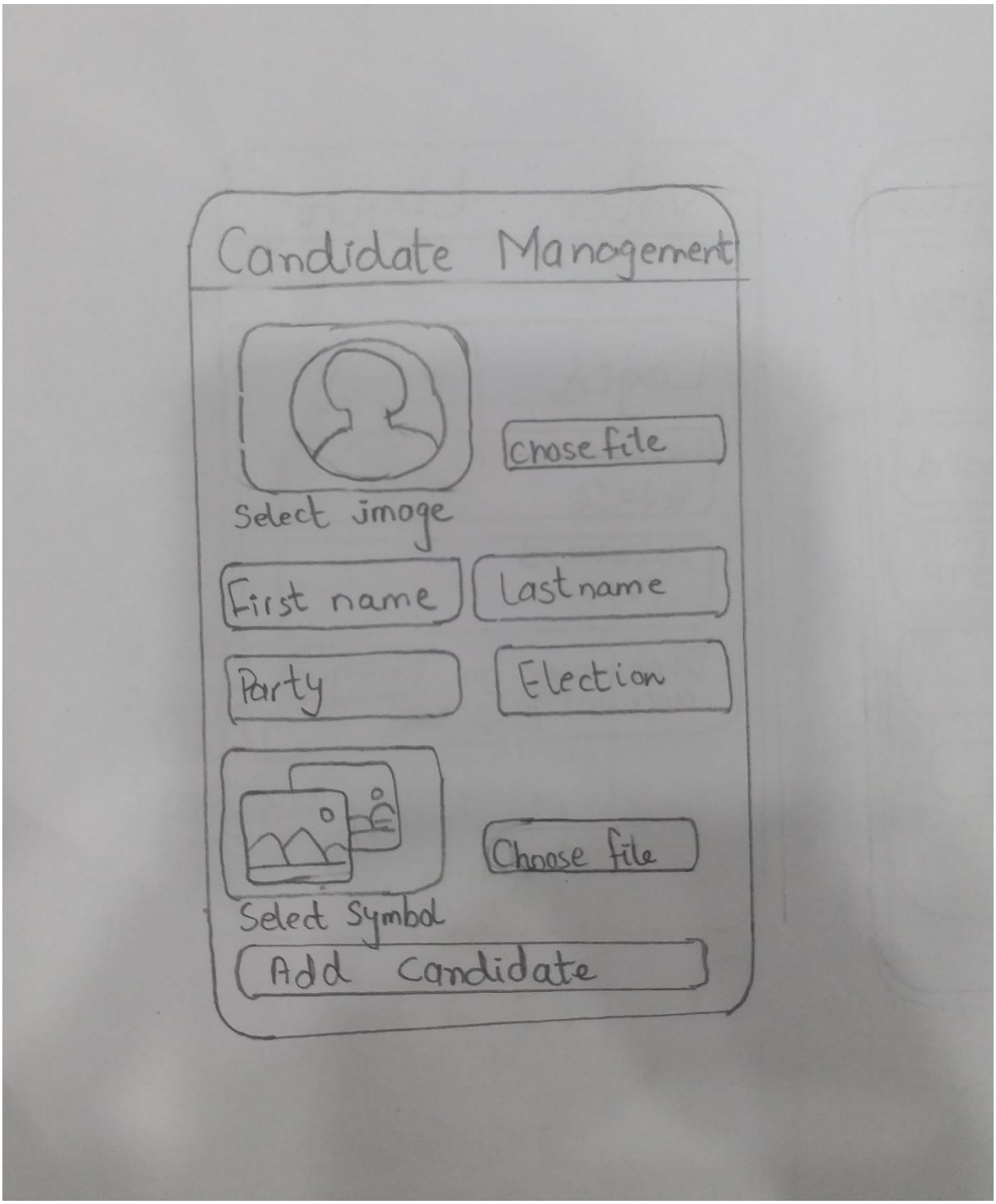


Figure 9.15 Candidate Management

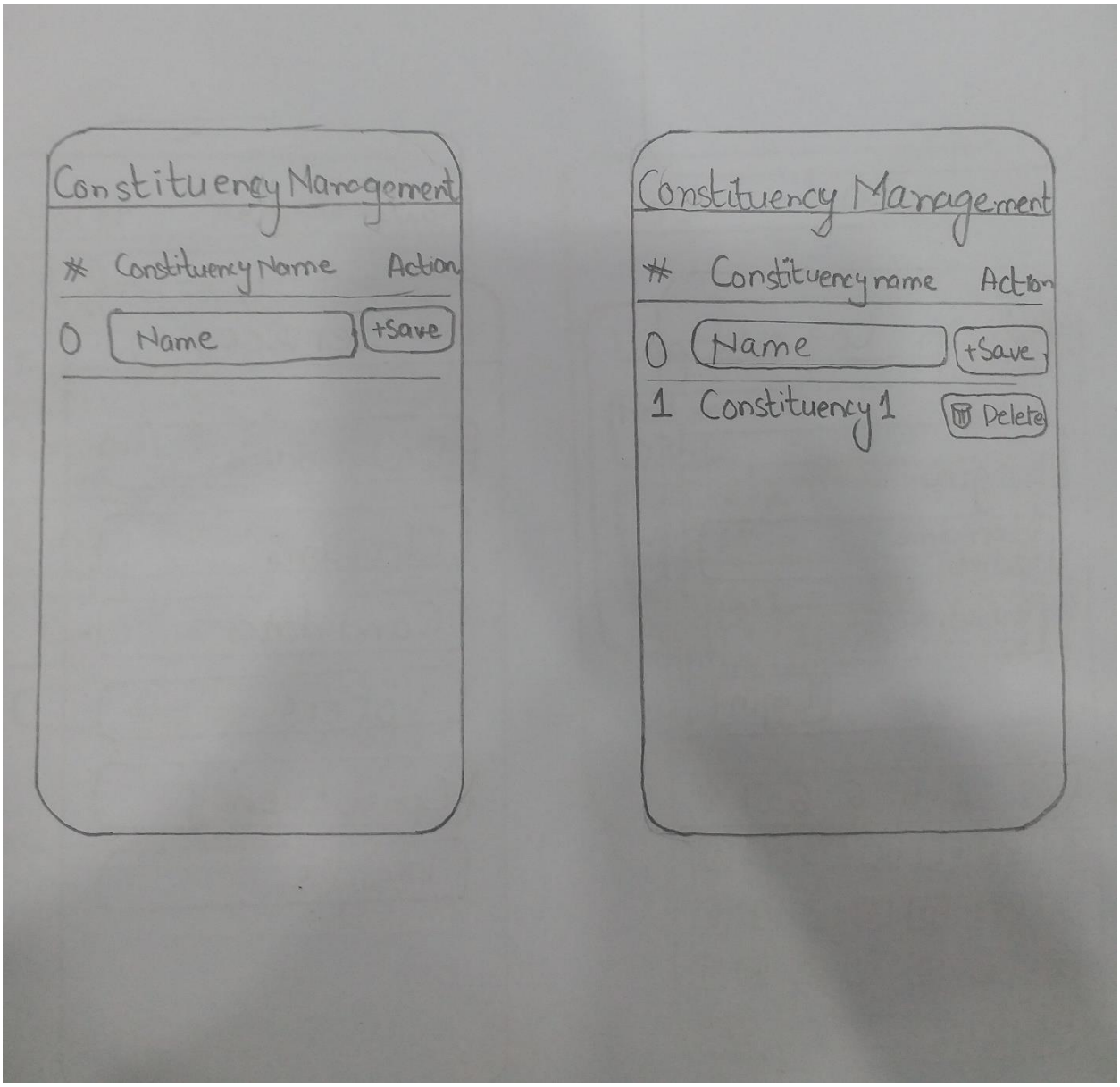


Figure 9.16 Constituency Management

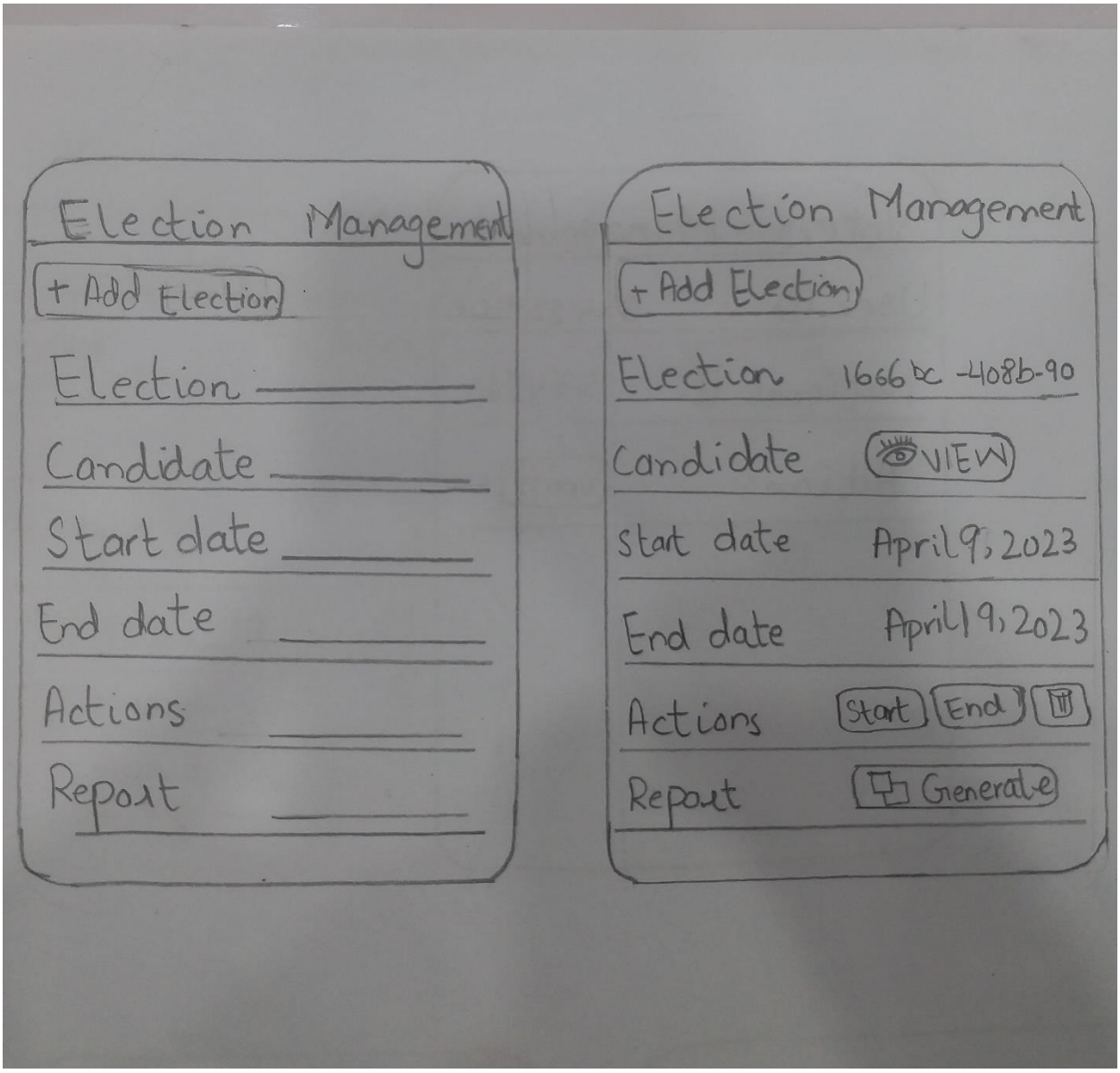


Figure 9.17 Election Management

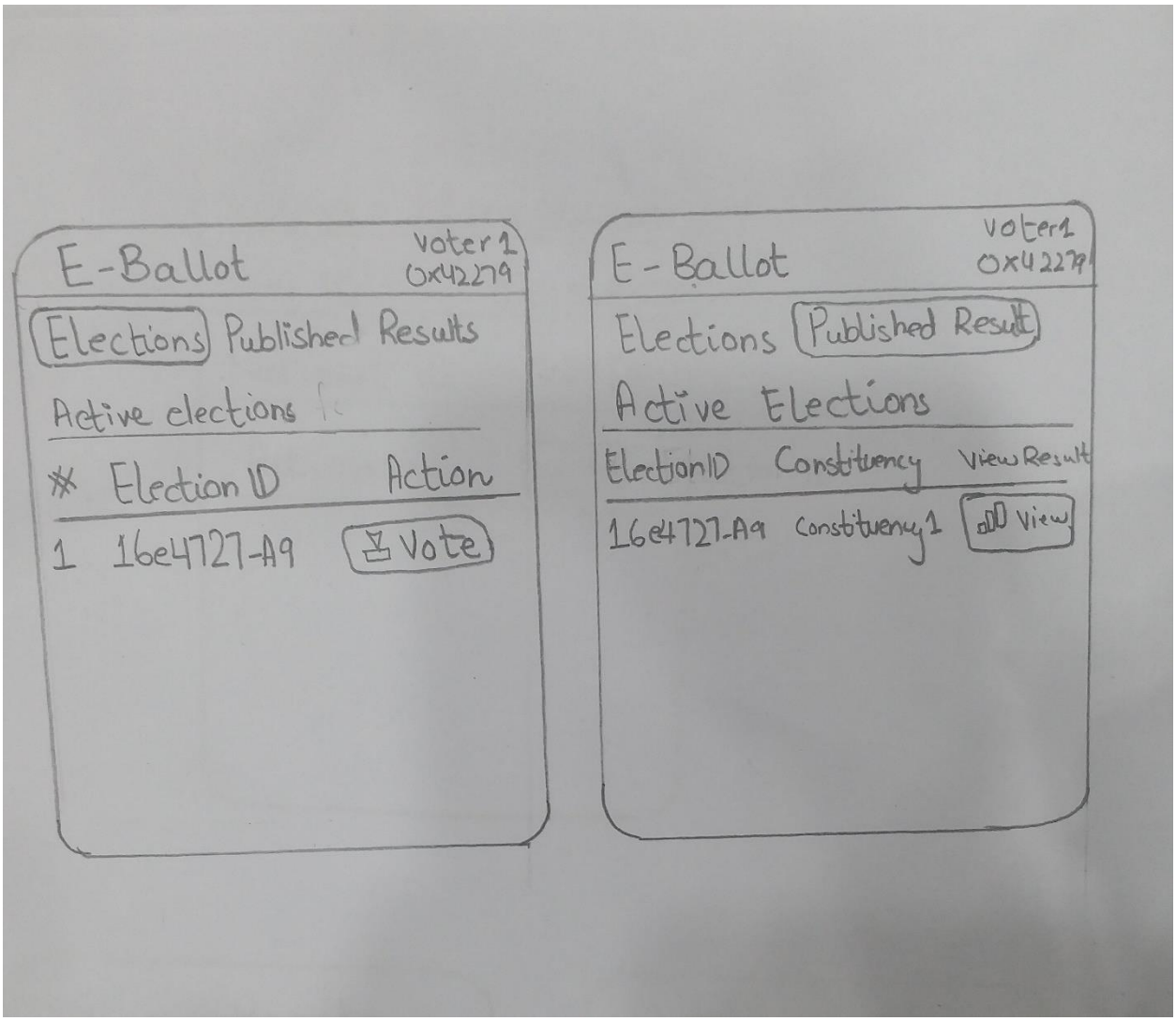


Figure 9.18 Vote Case and View Result Page

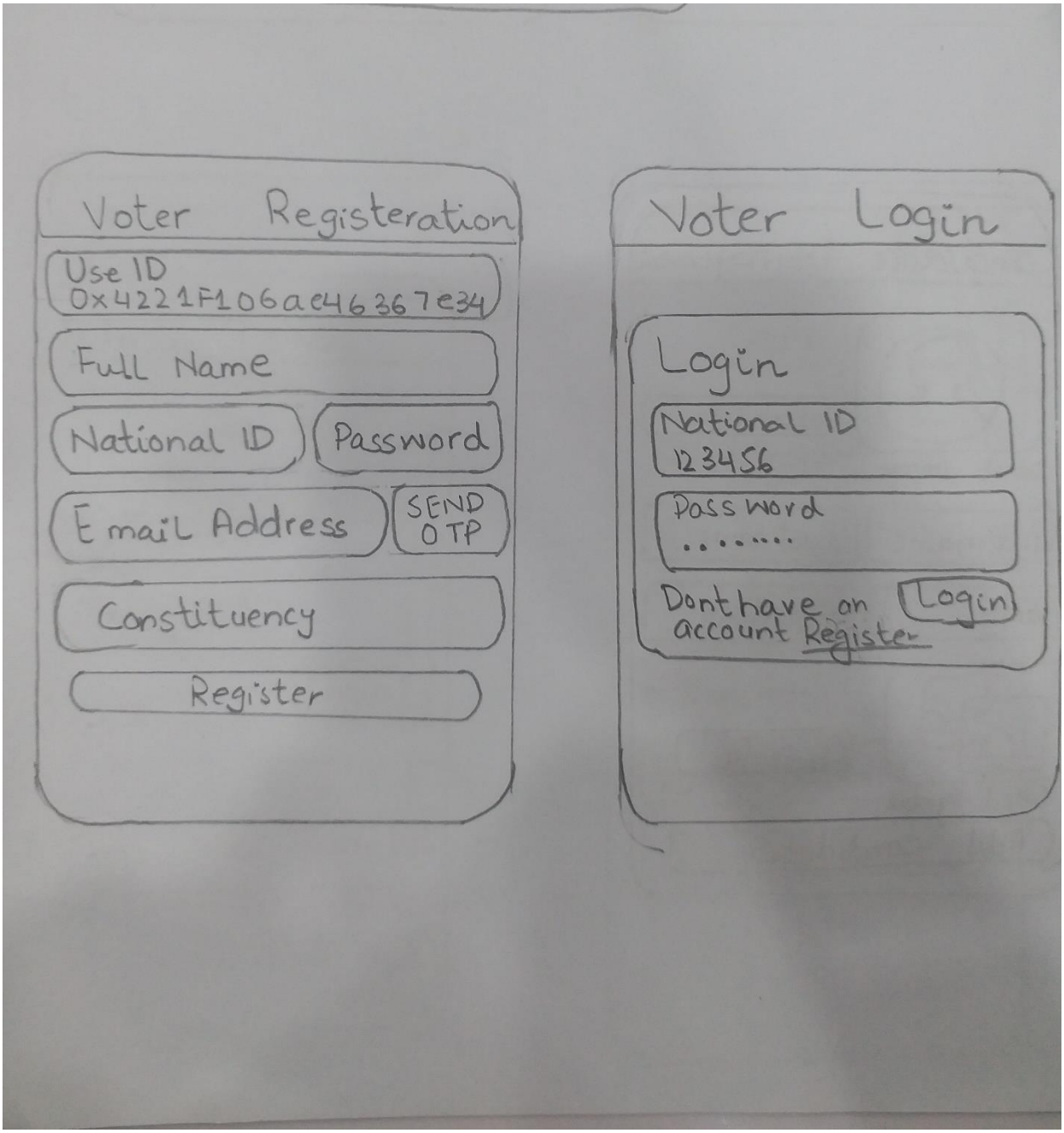


Figure 9.19 Voter Registration and log in page

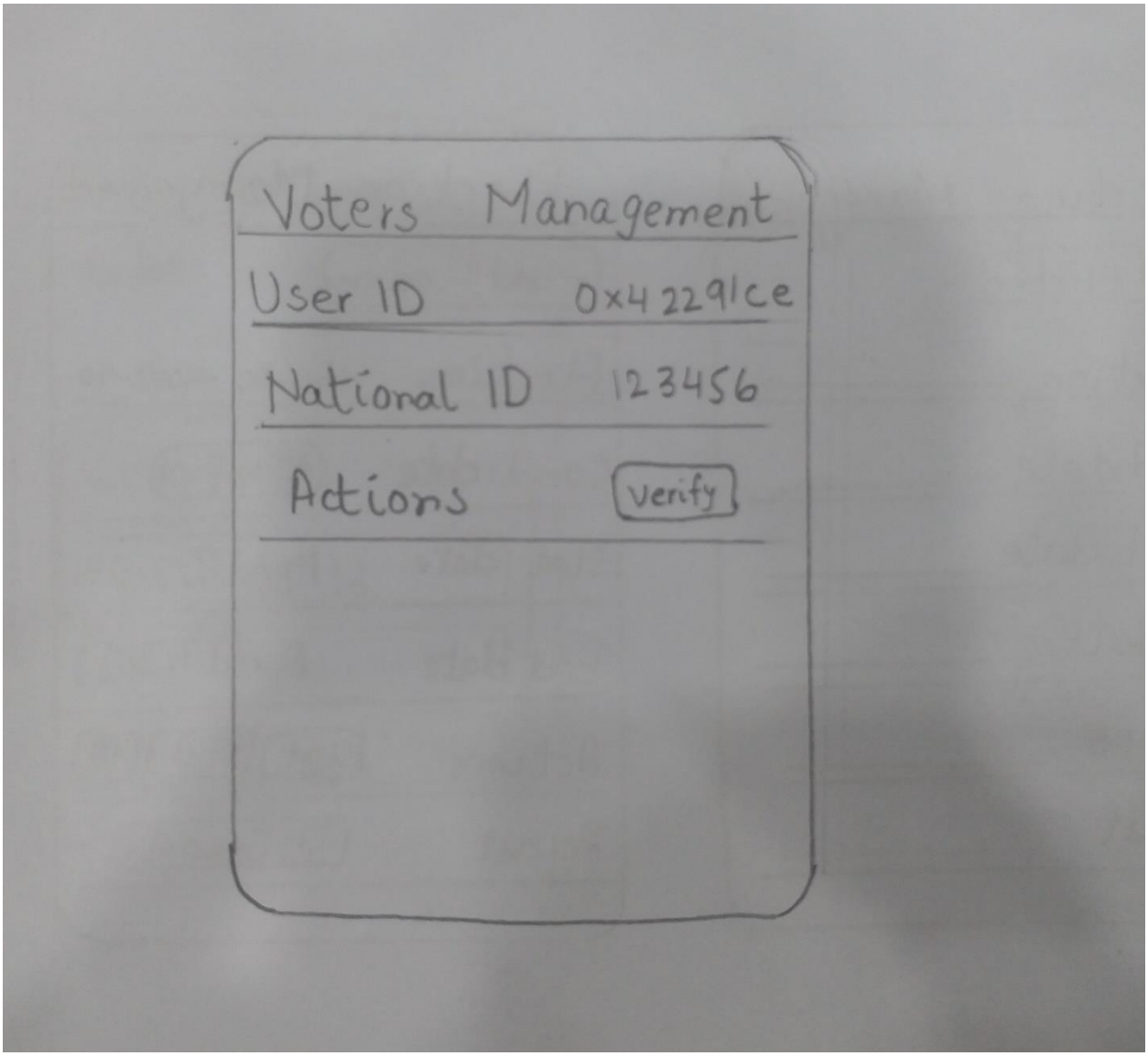


Figure 9.20 Voters Management

10 Review of Technology

There are numerous technologies available to us for developing software in the market. Furthermore, frameworks can be implemented in a variety of ways, depending on public accessibility and application use. Before deciding on a specific technology for framework development, it's critical to consider how the application may maximize its benefits for the intended consumers. The E-voting system attempts to improve election management and administration by utilizing blockchain technology. However, to properly achieve this goal, the E-voting system will be designed as a user-friendly RESTful backend API supplemented by a web application.

10.1 Algorithm Development

An algorithm created specifically to function in a distributed system is known as a distributed algorithm. A group of independent computers that don't share memory make up a distributed system. Each CPU in this system has its own memory, and communication between them is made possible by communication networks. These networks' internal communication is accomplished by processes operating on different machines talking to one other. It's important to keep in mind that many distributed system algorithms may require a coordinator to carry out tasks needed by other processes.

10.1.1 The Ring Algorithm

This approach is intended for systems that are logically or physically organized as rings. It is assumed in this method that communication between processes is one-way, allowing each process to communicate exclusively with the process immediately to its right. An active list, which comprises the priority numbers of all running processes in the system, serves as the algorithm's main data structure.

All system processes are grouped together in a logical ring using this approach, which is frequently assigned based on the numerical order of network addresses. The ring's communication is unidirectional; thus, messages can only move in one of two directions: clockwise or anticlockwise.

The token must be generated again if it is lost, which is a significant drawback of this approach. It can be difficult to find a token that has been lost. The token might not always be lost if it goes unreceived for a long time; instead, it might be being used by another operation.

10.1.2 Algorithm

Process P1 starts the election process by establishing a new, empty active list when it notices a coordinator failure. Then, it adds the number 1 to its own active list and sends an election message to its neighbor on the right.

Process P2 can react in one of three ways when it gets an election message from processes to its left:

I. P2 adds the number 2 to its active list and transmits the message to its neighbor on the right if the received message does not already contain the number 1 in its active list. In essence, this step raises the priority level in the active list.

II. A new active list with the numbers 1 and 2 is created if this is the first election message that P2 has either received or transmitted. Then, it broadcasts two messages on the election, one with the number 1 and the other with the number 2.

Process P2's active list now includes the numbers of all the other processes that are currently running in the system if it receives its own election message with the number 1. In this situation, process P2 can select the process with the greatest priority number from the list as the new coordinator or leader of the distributed system.

10.2 Technologies Used for Development

10.2.1 Front-End Technologies

HTML

A markup language called HTML is used to organize the information and visual appearance of online pages. HTML is largely used in the context of an e-voting system to provide the user interface that voters interact with. It outlines the format, forms, buttons, and other components required to gather voter input. HTML forms with elements make it easier to gather information and voter preferences. JavaScript can also be added to HTML to improve client-side interactivity and validation. But it's important to stress that more than just HTML is needed for an electronic voting system. Strong security safeguards, a database for data management, server-side programming for processing and securely storing votes, and extensive testing to assure the system's functionality, security, and dependability are all components of a robust system. Strong security safeguards, a database for data management, server-side programming for processing and securely storing votes, and extensive testing to assure the system's functionality, security, and dependability are all components of a robust system. Although HTML is an essential part of the frontend, an effective and secure e-voting system is created by combining several other technologies.

Features of HTML

- **Structure:** With HTML, you may specify the fundamental elements of a web page, such as headings, paragraphs, lists, and more. To mark up various content segments, it uses tags.
- **Hyperlinks**: The a> (anchor) element in HTML enables you to link to other web pages, documents, or resources both inside and outside of your website.

- **Images & Media:** Using elements like **"img," "audio,"** and **"video,"** you can include images and multimedia information like audio and video.
- **Forms:** To create interactive forms that users can fill out and submit, HTML includes form components like form>, input>, textarea>, and button>. For purposes like contact forms or search boxes, this is crucial for gathering user data.
- **Semantic Elements:** HTML5 introduced semantic elements such as "header," "nav," "main," "article," "section," and "footer." These aspects offer a more thorough and organized manner to categorize the various elements of a web page, making it simpler for search engines and auxiliary aids to comprehend the material.
- **Lists:** To structure text in a list format, HTML supports ordered lists (ol>), unordered lists (ul>), and definition lists (dl>)

CSS

The visual display and layout of HTML texts are determined by CSS, a key language in web development. Developers can use selectors to choose HTML components and then customize them using attributes and values in CSS. The appearance of web content is managed by stylesheets that are composed of selectors and style declarations. CSS uses inheritance and cascading principles to resolve conflicts and spread styles throughout the structure of the document. Responsive design is made possible via media queries, enabling designs to adjust to various screen sizes and devices. Pseudo-classes and pseudo-elements are also included in CSS to target states or elements. It can be used to support modular and maintainable code by being incorporated into HTML documents as inline styles, internal stylesheets, or external stylesheets linked to the HTML. For more effective development, CSS preprocessors like Sass enrich CSS with variables and mixins. Overall, CSS is crucial for determining how web pages look and for guaranteeing a consistent and pleasing user experience.

JavaScript

One of the key technologies used to create content for the World Wide Web is JavaScript. All contemporary web browsers support it natively, therefore using it doesn't require any additional plugins. It is commonly used in the creation of websites. In addition to prototype-based programming with first-class functions, imperative programming, and functional programming, JavaScript is a flexible language that supports a number of other programming paradigms. It offers an Application Programming Interface (API) for dealing with operations on text, arrays, dates, regular expressions, and fundamental DOM manipulation. However, it relies on the host environment to provide APIs for functions like network operations, storage, and complex graphics.

Despite having a similar name, syntax, and some standard libraries to Java, the two languages are unique and have very different designs. The creation of other programming languages like Self and Scheme had an impact on JavaScript.

JavaScript is also utilized in non-Web settings such desktop widgets, site-specific browsers, and PDF documents. The use of JavaScript for server-side Web applications has grown in popularity because of the development of new, faster JavaScript virtual machines (VMs) and the platforms that are based on them. JavaScript has historically been implemented on the client side as an interpreted language, but more current browsers do just-in-time compilation. JavaScript is also used by programmers to create desktop and mobile applications as well as video games.

ReactJS

A JavaScript package called ReactJS is used to create a variety of reusable UI components. It produces user interface (UI) elements that can display dynamic data. React is frequently used as the V in MVC. React provides a more straightforward programming model and higher speed. Node may be used to build React on the server, while React Native can be used to charge native apps. One-way reactive data flow, which React implements, is less boilerplate and simpler to understand than conventional data binding (Hasio et al. 2017).

10.2.2 Frameworks

10.2.2.1 Truffle

A development framework for Ethereum called Truffle makes it easier to create and use smart contracts. The backend of blockchain applications can be managed more easily because of the tools it offers for creating, testing, and deploying Solidity smart contracts Benny et al (2020).

10.2.2.2 Json

JSON (JavaScript Object Notation) is a popular data format that is renowned for being straightforward and readable. Its key-value pair format makes it simple to express structured data. JSON uses double quotes for keys, is like JavaScript object literals, and supports a wide range of data types, including strings, numbers, Booleans, arrays, and objects. It acts as a standard for all data interchange, making it simple to transfer information across systems, save data in files, or configure settings. JSON is frequently used for APIs, data storage, and configuration files in web development and other fields where data needs to be both machine-readable and understandable to developers due to its developer-friendly structure.

10.2.2.3 Vite

Vite is a cutting-edge building tool and development server made to improve the process of creating web apps. It is distinguished by its outstanding performance, providing quick startup times during

development thanks to the support for native ES modules. This tool offers built-in Hot Module Replacement (HMR) for seamless modifications to code and styles without complete page reloads and supports many JavaScript frameworks in addition to Vue.js. Vite is a well-liked option for web developers looking for a quicker and more productive workflow when creating modern web applications because of its effective development server and simplified build process.

10.2.3 Backend Technologies

MySQL

An important part of the data administration and storage process for several web applications and software systems is played by MySQL, an open-source relational database management system (RDBMS). MySQL is a popular database management system that is renowned for its dependability, performance, and simplicity of use. It uses SQL (Structured Query Language) to communicate with the database and enables users to add, edit, delete, and read data. Because it supports a variety of storage engines, each with unique features, MySQL may be used for a variety of use cases. Due to its dependability and wide-ranging community support, MySQL continues to be a popular option for powering content management systems, e-commerce platforms, and data-driven applications. This has facilitated its continuing development and wide adoption in the software development industry.

Node

Node.js, sometimes known as Node, is an open-source runtime environment that permits the execution of JavaScript on the server. It is based on the incredibly quick V8 JavaScript engine. Node.js is favored because of how well it handles several concurrent connections and I/O activities thanks to its non-blocking and asynchronous nature. As a result, it is frequently used to create scalable web applications, real-time systems, and APIs. By allowing programmers to utilize JavaScript on both the server and client sides, Node.js promotes code reuse. It is a popular option in contemporary online and application development thanks to its enormous ecosystem of libraries and packages, which are handled by npm (Node Package Manager).

10.2.4 Contract

Solidity

A high-level programming language called Solidity was created expressly for creating smart contracts for use on blockchain systems like Ethereum. It offers a simple, safe method for producing self-executing contracts with predetermined terms and conditions. The Ethereum Virtual Machine (EVM) runs Solidity code, which can be used to automate a variety of procedures, including token production, crowdfunding, and decentralized apps (DApps). A flexible option for blockchain development, it has robust typing and

enables inheritance, libraries, and complicated data structures. However, since flaws or vulnerabilities can pose serious financial concerns, it's imperative to build Solidity programming with security in mind. As blockchain technology gets traction for a variety of applications outside of cryptocurrencies, Solidity's popularity is rising Benny et al (2020).

Ganache

Ethereum has a personal blockchain emulator called Ganache. To test and develop blockchain applications, it enables developers to build a local blockchain environment. To test smart contracts and applications, we can set up a private Ethereum blockchain network using Ganache.

In conclusion, e-voting applications will construct local blockchains using Ganache and Truffle, smart contracts using Solidity, user interfaces using ReactJS, and server-side logic using Node.js. Figure illustrates how these technologies work together to provide a safe and transparent e-voting system on the Ethereum blockchain.

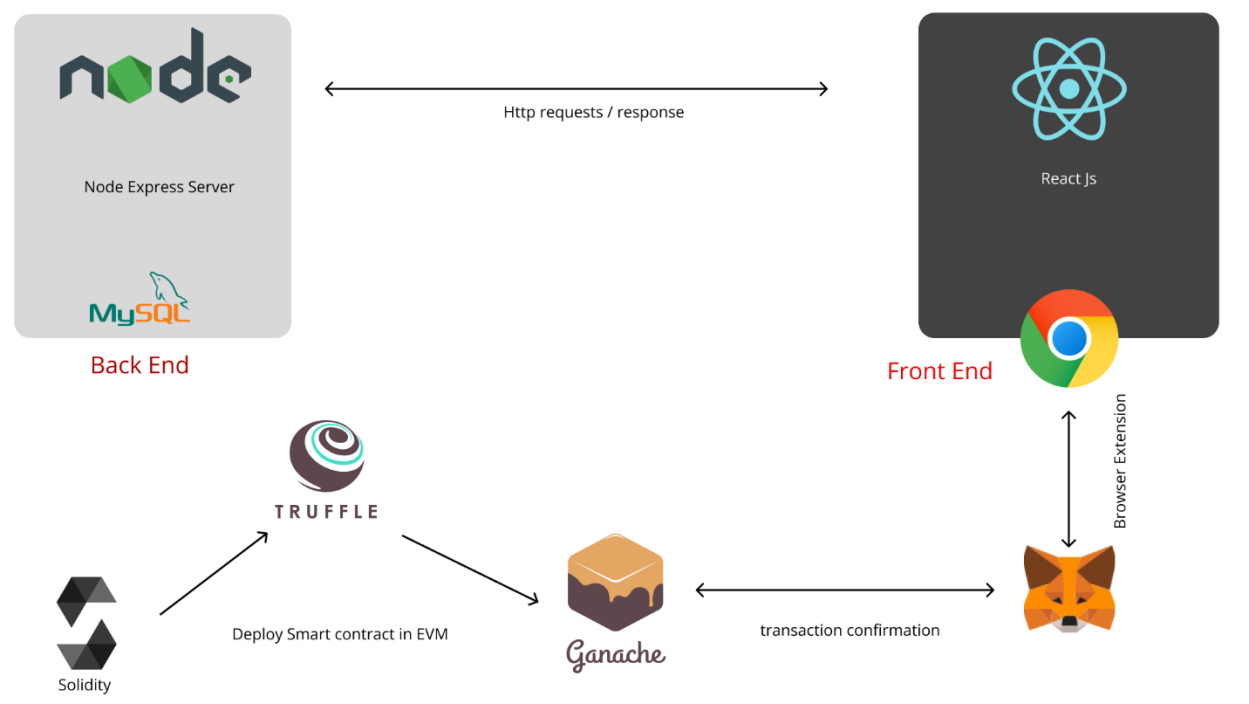


Figure 10.1 High Level System Flow

11 Development

This section will go over the most recent device modules and their high-fidelity prototypes. We will also deconstruct the time boxes connected with these modules and address any potential obstacles or concerns that may arise because of the deconstruction. Finally, we will investigate alternative techniques and tactics for addressing these difficulties and keeping the project on schedule.

11.1 New system Modules

The latest E-voting application system comprises the following modules:

User Module:

- Users have the capability to register himself
- Users have the capability to log in through the framework

Dashboard Module:

- View active elections
- View active voters
- View total elections
- View registered voters
- View constituencies

Election Module:

- Admin will add start date of election
- Admin will add end date of election
- Admin will start election
- Admin will end election
- Voter will cast election

Candidate Module:

- Admin will add candidate by uploading profile picture of candidate
- Admin will add candidates by adding first name, last name, party, election, and candidate symbol.

Constituency Module:

- Admin will add constituencies by adding constituency name

Results Module:

- Admin will generate result report
- Voter will view results

Log out Module:

- Admin will log out the system
- Voter will log out the system

11.2 Prototypes with High Precision for New Modules

Here are a set of high-fidelity sample screens for each module of the e-voting system.

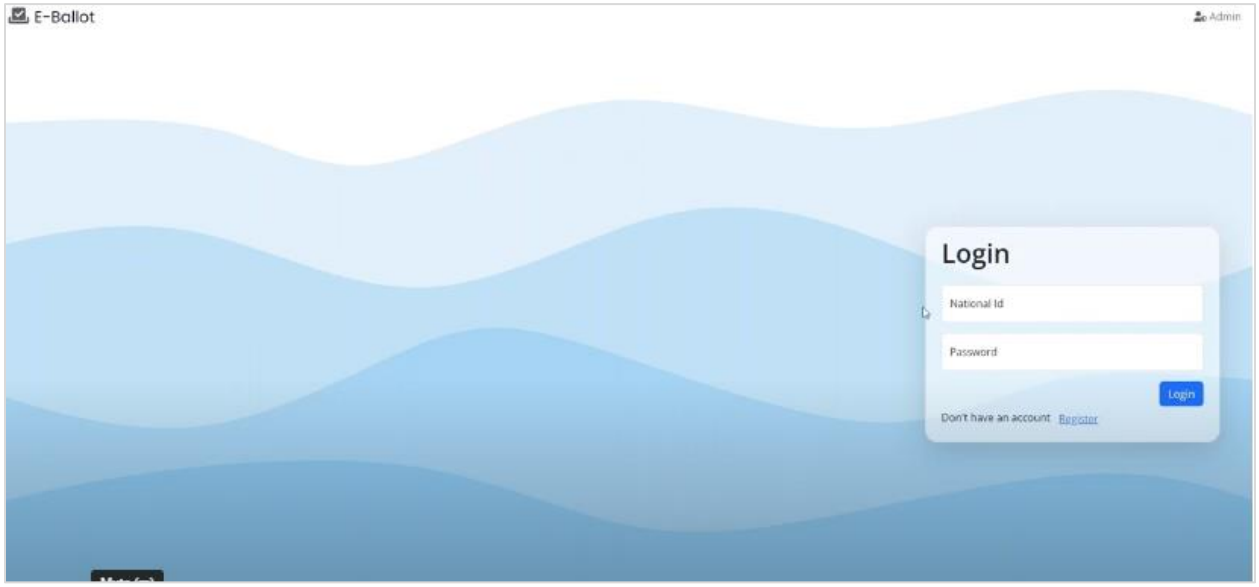


Figure 11.1 User Log in

Figure 11.2 Menu and Services

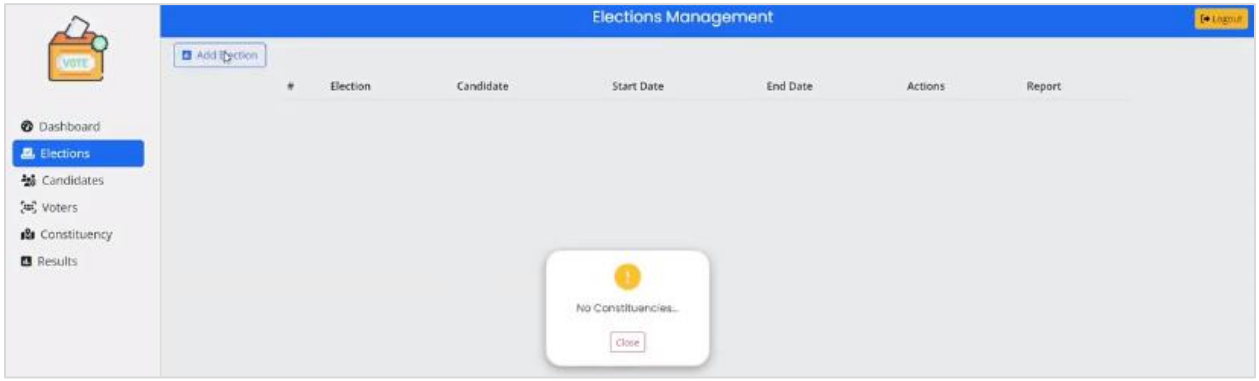


Figure 11.3 Election Management Module

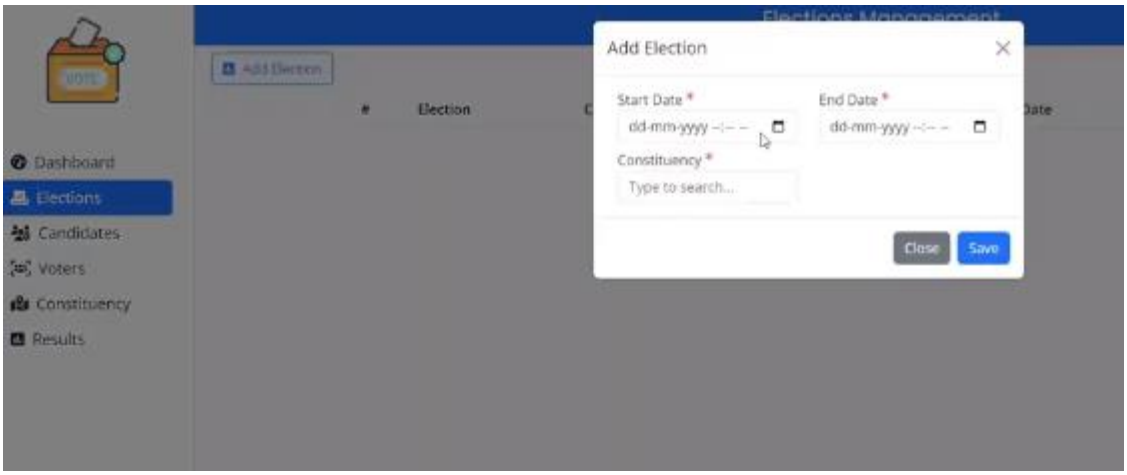


Figure 11.4 Add Election

Figure 11.5 Add Constituency Module

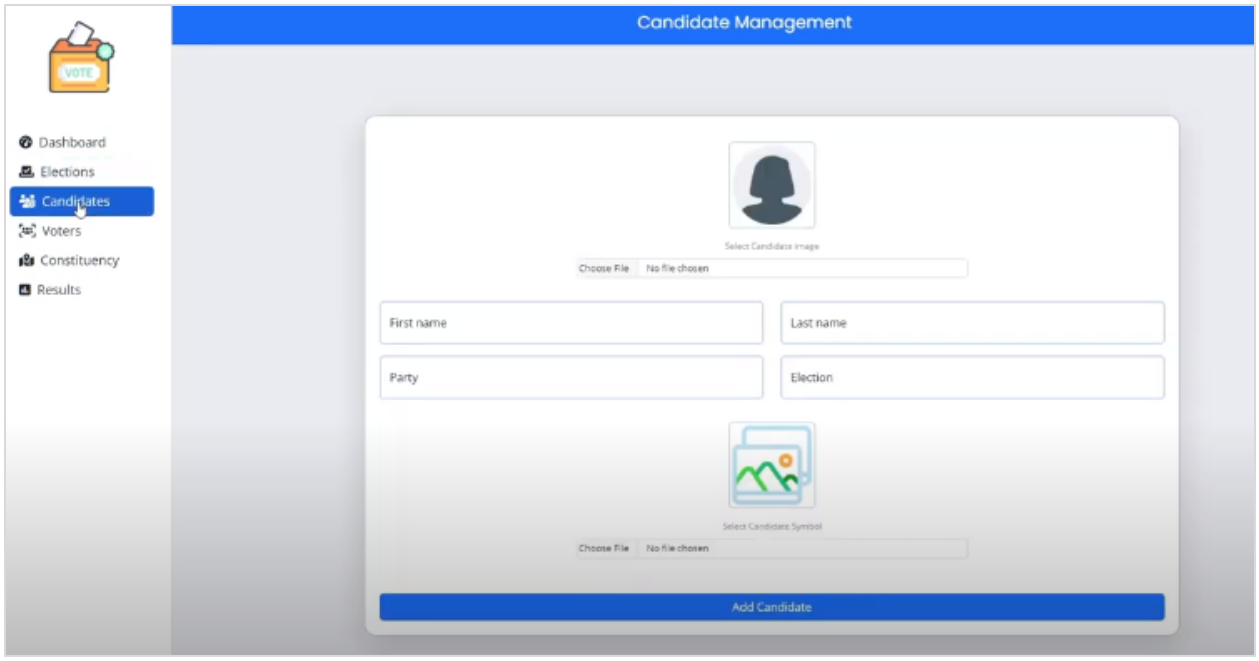


Figure 11.6 Add Candidate Module

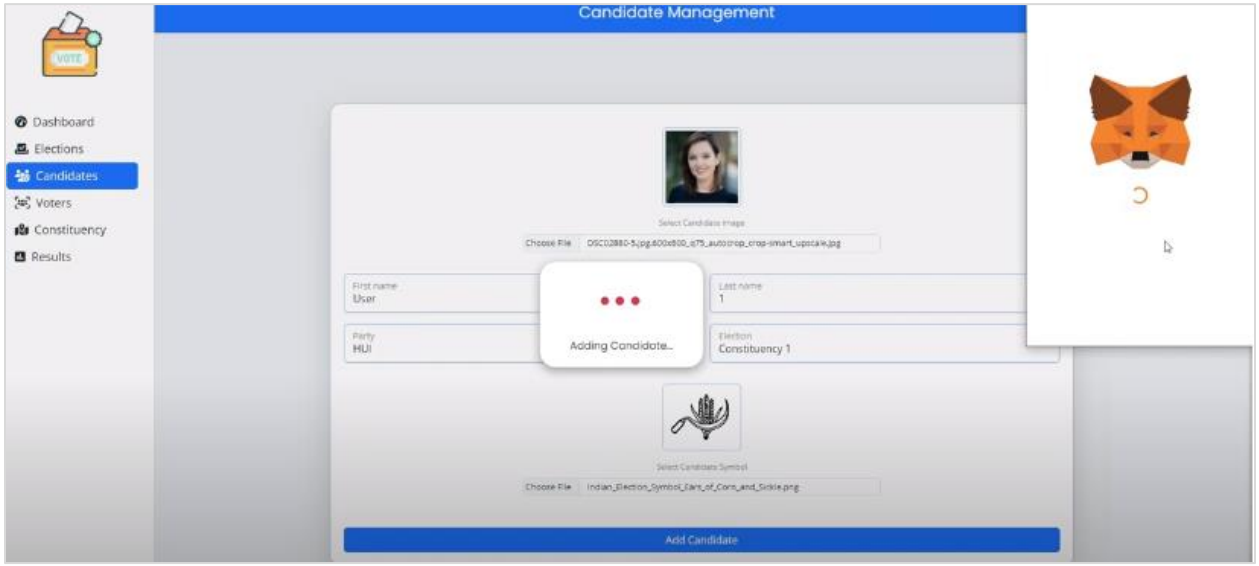


Figure 11.7 Process of adding Candidate

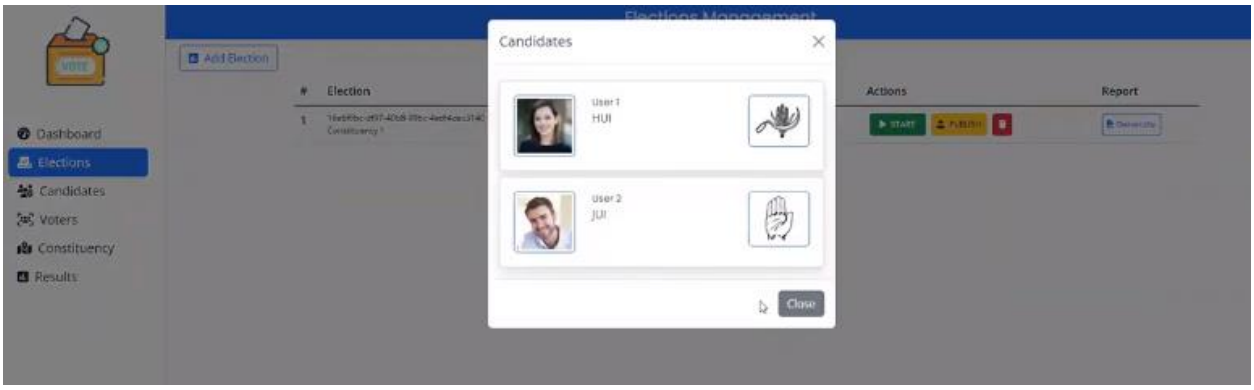


Figure 11.8 View Election Candidate

Figure 11.9 Start Election

Figure 11.10 Admin Dashboard Analytics

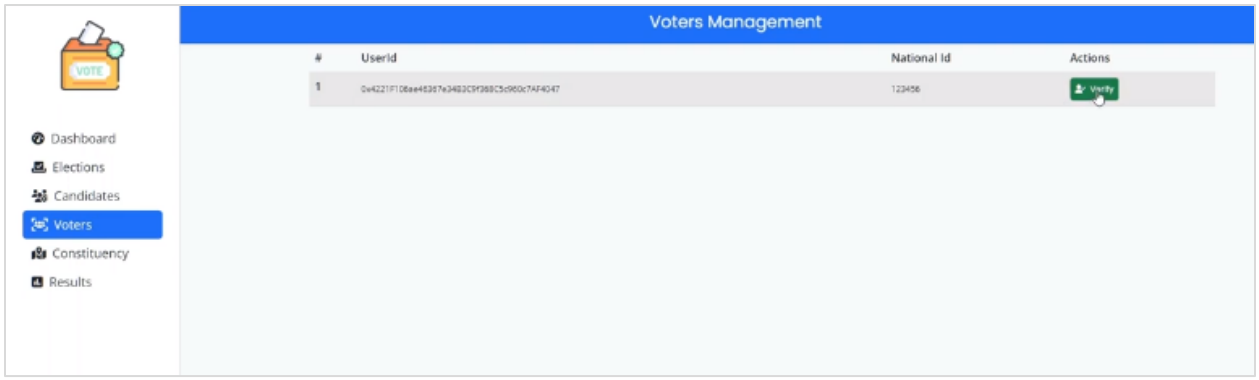


Figure 11.11 Verify User/Voter

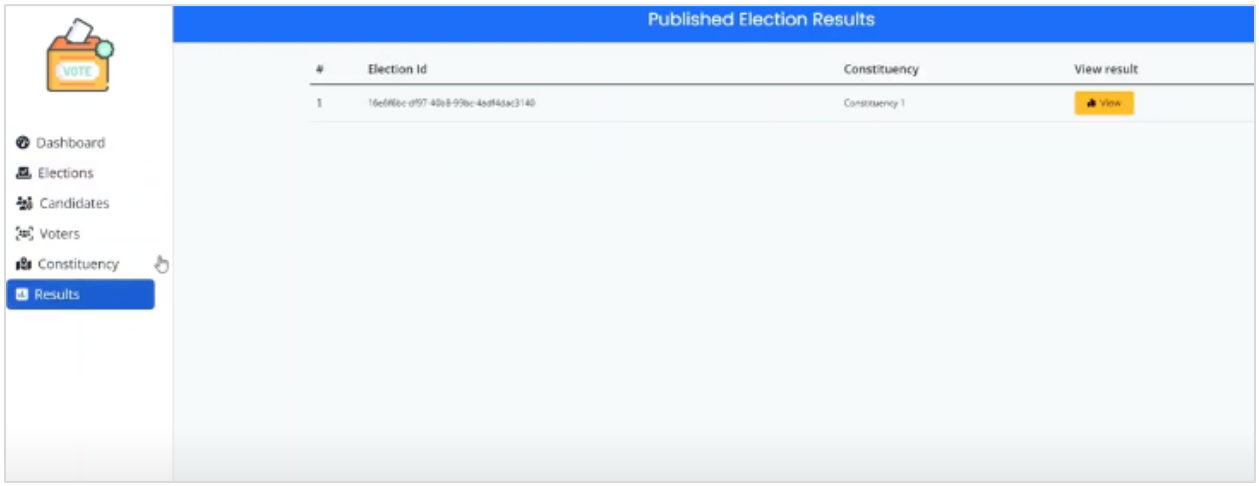


Figure 11.12 Publish Result

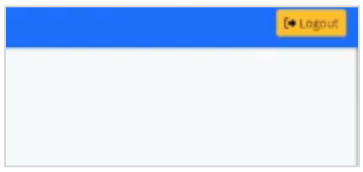


Figure 11.13 Admin Log out

Figure 11.14 Voter Registration

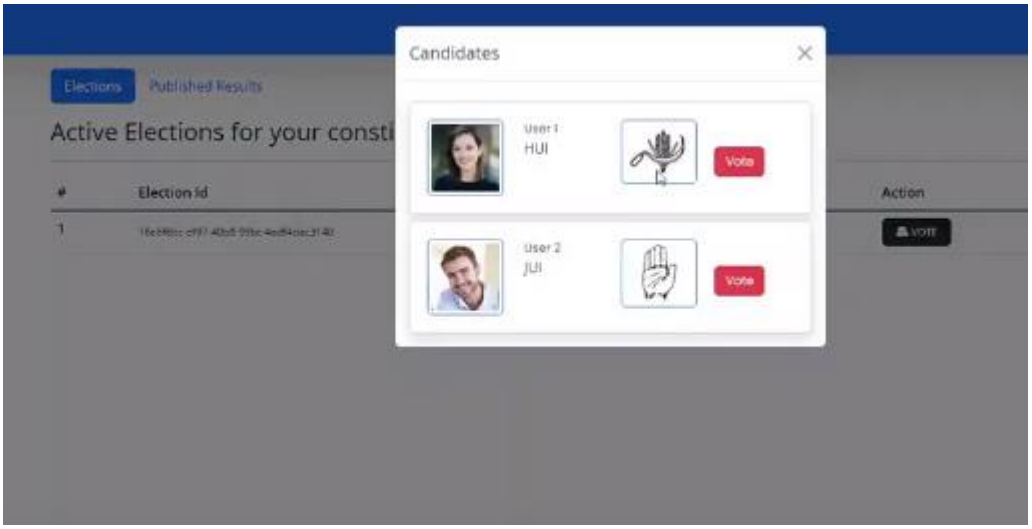


Figure 11.15 View candidates in his/her Constituency

Figure 11.16 Election Results

11.3 Time Boxing Module Breakdown

The creation of an e-voting system is a difficult and time-consuming task. The project has been meticulously segmented into numerous time boxes, each with a specialized focus on a distinct feature of the system, to assure its successful completion. These time constraints are critical for effectively managing the project, as software development is a gradual process. The approach begins with gathering the specs required to kickstart the project. As the project progressed, an iterative method was used to modify and improve the product. The feature prioritization followed a logical process, beginning with must-have functionality, then should-have, and ultimately could-have features. This methodical approach ensures that the project proceeds methodically and that critical elements are addressed first.

Table 34 Time Box for every Module

Time Box	Modules
Time Box 1	User Module
Time Box 2	Dashboard Module
Time Box 3	Election Module
Time Box 4	Candidate Module

Time Box 5	Constituency Module
Time Box 6	Result Module
Time Box 7	Log out Module

	Task Mode	Task Name	Duration	Start	Finish	Resource Names
1		**E-Voting System**	**90 days**	**Wed 22-02-23**	**Tue 27-06-23**	
2		Feasibility Study	7 days	Wed 22-02-23	Thu 02-03-23	Analyst
3		**Time Box 1**	**10 days**	**Fri 03-03-23**	**Thu 16-03-23**	**Analyst, Designer, Developer, Tester, User**
4		Analysis	1 day	Fri 03-03-23	Fri 03-03-23	Analyst
5		Design	2 days	Mon 06-03-23	Tue 07-03-23	Designer
6		Development	6 days	Wed 08-03-23	Wed 15-03-23	Developer
7		Testing	1 day	Thu 16-03-23	Thu 16-03-23	Tester, User
8		**Time Box 2**	**12 days**	**Fri 17-03-23**	**Mon 03-04-23**	**Analyst, Designer, Developer, User, Tester**
9		Analysis	2 days	Mon 20-03-23	Tue 21-03-23	User,Analyst
10		Design	2 days	Wed 22-03-23	Thu 23-03-23	Designer
11		Development	6 days	Thu 23-03-23	Thu 30-03-23	Developer, User
12		Testing	2 days	Fri 31-03-23	Mon 03-04-23	Tester, User
13		**Time Box 3**	**16 days**	**Mon 03-04-23**	**Mon 24-04-23**	**Designer, Developer, Tester, User,Analyst**
14		Analysis	2 days	Mon 03-04-23	Tue 04-04-23	Analyst
15		Design	3 days	Fri 07-04-23	Tue 11-04-23	Designer
16		Development	7 days	Wed 12-04-23	Thu 20-04-23	Developer, User
17		Testing	2 days	Fri 21-04-23	Mon 24-04-23	Tester, User
18		**Time Box 4**	**19 days**	**Tue 25-04-23**	**Fri 19-05-23**	**Designer, Developer, Tester, User,Analyst**
19		Analysis	2 days	Tue 25-04-23	Wed 26-04-23	User,Analyst

Figure 11.17 Timeline for the Development Process (part 1)

20		Design	3 days	Thu 27-04-23	Mon 01-05-23	Designer
21		Development	7 days	Tue 02-05-23	Wed 10-05-23	Developer
22		Testing	3 days	Wed 17-05-23	Fri 19-05-23	Tester
23		**Time Box 5**	**10 days**	**Thu 18-05-23**	**Wed 31-05-23**	**Designer, Developer, Tester, User,Analyst**
24		Analysis	1 day	Thu 18-05-23	Thu 18-05-23	Analyst
25		Design	2 days	Fri 19-05-23	Mon 22-05-23	Designer, User
26		Development	6 days	Tue 23-05-23	Tue 30-05-23	Developer
27		Testing	1 day	Wed 31-05-23	Wed 31-05-23	Tester, User
28		**Time Box 6**	**8 days**	**Thu 01-06-23**	**Mon 12-06-23**	**Designer, Developer, Tester, User,Analyst**
29		Analysis	1 day	Thu 01-06-23	Thu 01-06-23	Analyst
30		Design	5 days	Fri 02-06-23	Thu 08-06-23	Designer, User
31		Development	1 day	Fri 09-06-23	Fri 09-06-23	Developer
32		Testing	1 day	Mon 12-06-23	Mon 12-06-23	Tester, User
33		**Time Box 7**	**8 days**	**Tue 13-06-23**	**Thu 22-06-23**	**Designer, Developer, Tester, User,Analyst**
34		Analysis	1 day	Tue 13-06-23	Tue 13-06-23	User,Analyst
35		Design	2 days	Wed 14-06-23	Thu 15-06-23	Designer
36		Development	4 days	Fri 16-06-23	Wed 21-06-23	Developer
37		Testing	1 day	Thu 22-06-23	Thu 22-06-23	Tester

Figure 11.18 Timeline for the Development Process (Part 2)

11.4 Possible Breakdown of a Problem in a Complex Module

Because the solution's elements were complicated and linked, an iterative development strategy was required. These characteristics were so interconnected that developing them all at once was impracticable. As a result, the development process included numerous revisions, each of which improved the functionality more. In the following section, we'll look at Time Box 3 and 4, are difficult features that went through multiple revisions before reaching their intended degree of perfection.

11.4.1 Iteration 1

- Basic requirements for creating elections
- Data for training and testing were generated.
- A better algorithm for this scenario was discovered

```
DataTypes = require("sequelize");

module.exports = (sequelize, Sequelize) => {
    return sequelize.define("election", {
        electionId: {
            type: DataTypes.STRING,
        }, startDate: {
            type: DataTypes.DATE,
        }, endDate: {
            type: DataTypes.DATE,
        }, status: {
            type: DataTypes.BOOLEAN,
        }, result: {
            type: DataTypes.BOOLEAN,
        }
    });
};
```

11.4.2 Iteration 2

- Assign roles through smart contracts
- Data for training and testing were generated.
- A better algorithm for this scenario was discovered

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.11;

/**
 * @title Roles
 * @dev Library for managing addresses assigned to a Role.
 */
library Roles {
    struct Role {
        mapping (address => bool) bearer;
    }

    /**
     * @dev Give an account access to this role.
     */
    function add(Role storage role, address account) internal {
        require(!has(role, account), "Roles: account already has role");
        role.bearer[account] = true;
    }

    /**
     * @dev Remove an account's access to this role.
     */
    function remove(Role storage role, address account) internal {
        require(has(role, account), "Roles: account does not have role");
        role.bearer[account] = false;
    }

    /**
     * @dev Check if an account has this role.
     * @return bool
     */
    function has(Role storage role, address account) internal view returns (bool) {
        require(account != address(0), "Roles: account is the zero address");
        return role.bearer[account];
    }
}
```

11.4.3 Iteration 3

- Adding Candidate in election
- Get requirements for get candidates
- Data for training and testing were generated.
- A better algorithm for this scenario was discovered

```
function addCandidate(string memory id, string memory name, string memory eId) public {
    require(adminRole.has(msg.sender), "only admin can add Candidates");
    Candidate storage candidate = CandidatesMap[id];
    candidate.id = id;
    candidate.name = name;
    candidate.voteCount = 0;

    Election storage election = ElectionsMap[eId];
    election.candidates.push(candidate);

    candidates.push(candidate);
}

function getCandidates() public view returns (Candidate[] memory){
    return candidates;
}

function getCandidateVotes(string memory id) public view returns (uint){
    return CandidatesMap[id].voteCount;
}
```

```
DataTypes = require("sequelize");

module.exports = (sequelize, Sequelize) => {
    return sequelize.define("candidate", {
        fName: {
            type: DataTypes.STRING,
        }, lName: {
            type: DataTypes.STRING,
        }, party: {
            type: DataTypes.STRING,
        }, candidateImage: {
            type: DataTypes.STRING,
        }, candidateSymbol: {
            type: DataTypes.STRING,
        },electionId :{
            type: DataTypes.STRING,
        }
    });
};
```

11.5 Alternative Approaches That Could Work

Multiple iterations were necessary to construct a sophisticated function because of adhering to the DSDM guideline of "enough architecture up front." This function may have been produced in a single phase in a traditional software development methodology, such as the waterfall process, where the complete system

is defined first and then the function is built on top of that design. This traditional approach, however, is time-consuming because it entails creating the complete system at once. In contrast, the agile approach (DSDM with Agile) was used, which allowed for iterative improvements, beginning with a tiny unit, and progressively increasing to incorporate the complete feature. Because of its efficiency and adaptability, this iterative technique was chosen.

12 Testing

Technology is essential in today's world for creating numerous machines that make living easier. Software, however, frequently has a variety of faults and might not work as intended. Testing is therefore essential to maintain control and guarantee software that is error-free. It painstakingly finds every flaw, bringing the software into compliance with the stated requirements. Testing turns out to be quite cost-effective, preventing any faults later. Early bug fixes not only save money but also improve the quality of the final product after testing. The following techniques are utilized in this project to test all our test cases for the application, which uses Mocha as the testing framework

12.1 Test Case

Table 35 Test Case

SR. NO	Test Name	Test Criteria
1	Functional Test	Admin Login with valid credentials.
		Add constituency with a valid constituency name.
		Add election with a valid election name.
		Add candidate with valid credentials.
		View admin dashboard details.
		Voter register with valid credentials.
		Voter login after admin verification.
		Voter cast vote in active election.
		Generate accurate and complete report of election result.
		Destroy admin’s session upon logout.
		View the result of an election that has been published by an admin.

		Destroy the voter's session upon logout.
2	User Acceptance Test	Admin Login with valid credentials.
		Add constituency with a valid constituency name.
		Add election with a valid election name.
		Add candidate with valid credentials.
		View admin dashboard details.
		Voter register with valid credentials.
		Voter login after admin verification.
		Voter cast vote in active election.
		Generate accurate and complete report of election result.
		Destroy admin's session upon logout.
		View the result of an election that has been published by an admin.
		Destroy the voter's session upon logout.

Figure 12.1 Test Case

12.2 Test Execution

12.2.1 Functional Testing

Table 36 FT_01

Test Case ID	FT_01	**Test type**	Functional
Description	Admin Login	**Test priority**	High

Pre-requisite		Admin must have verified credentials.		Tester		Syed Mohammad Shah Mostafa		
Test execution Steps:								
No.	**Action**	**Inputs**	**Input type**	**Expected Output**	**Actual Output**	**Device**	**Result**	**Comment**
1.	Fill the fields	Username, password	Alphanumeric	Logged In	Logged In	Windows	Pass	Successfully Admin logged in

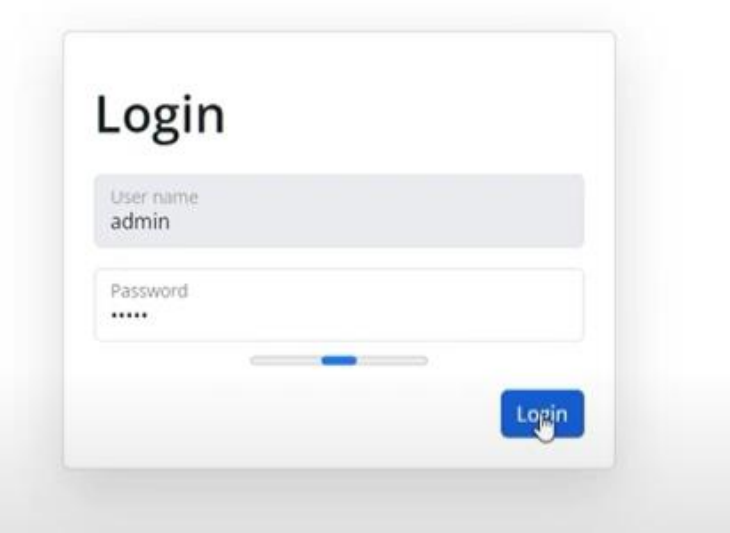


Table 37 FT_02

Test Case ID	FT_02	**Test type**	Functional
Description	Add constituency	**Test priority**	High
Pre-requisite	Logged in user	**Tester**	Syed Mohammad Shah Mustafa

Test execution Steps:

No.	**Action**	**Inputs**	**Input type**	**Expected Output**	**Actual Output**	**Device**	**Result**	**Comment**
1.	Fill the field	Add constituency name	Alphanumeric	Constituency added	Constituency added	Windows	Pass	Successful

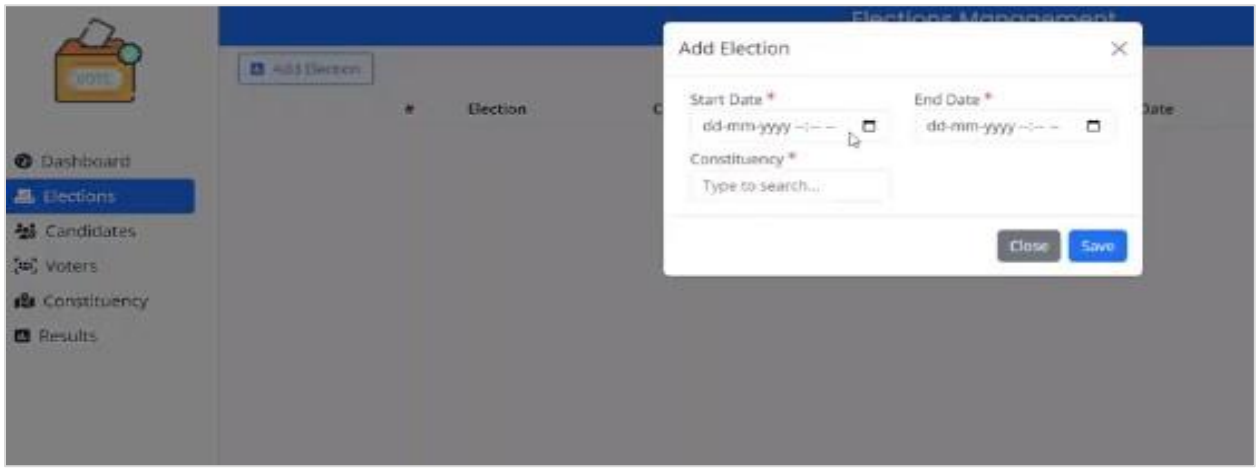
Table 38 FT_03

Test Case ID	FT_03	**Test type**	Functional
Description	Add election	**Test priority**	High
Pre-requisite	Constituency should be added	**Tester**	Syed Mohammad Shah Mustafa

Test execution Steps:

No.	**Action**	**Inputs**	**Input type**	**Expected Output**	**Actual Output**	**Device**	**Result**	**Comment**

1.	Click on add election button	Set start date, set end date, add constituency name	Alphanu meric	Election added	Election added	Windows	Pass	Successful



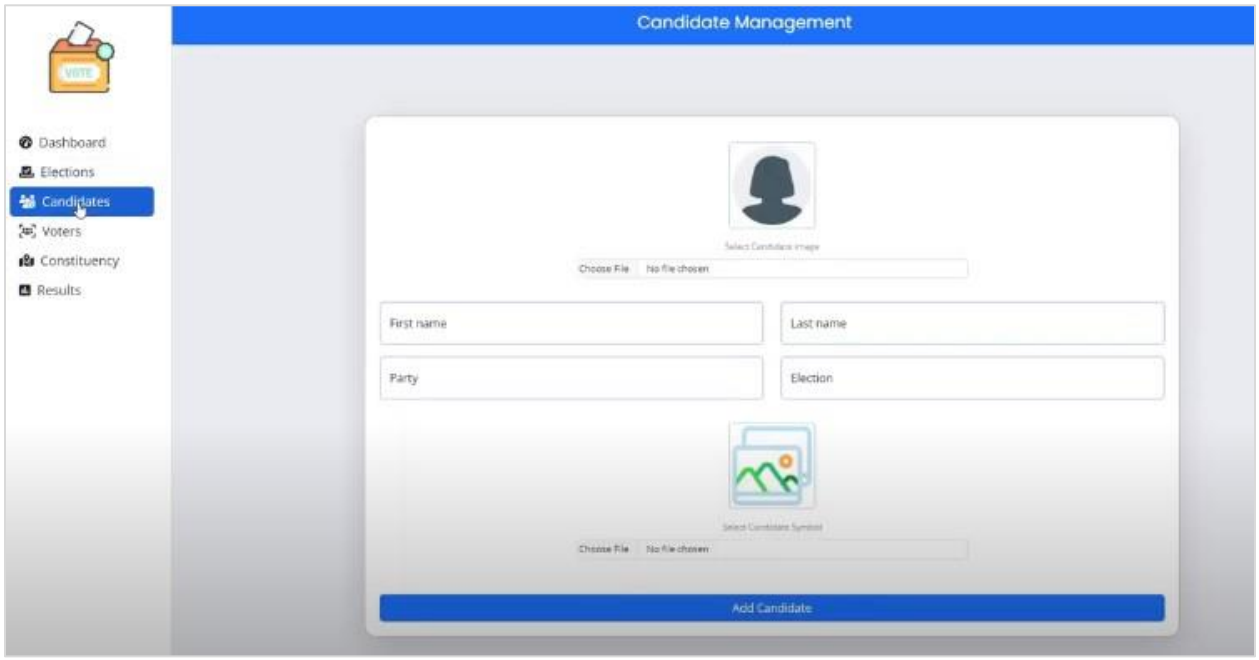


Table 39 FT_04

Test Case ID	FT_04	**Test type**	Functional

Description	Add candidate	Test priority	High
Pre-requisite	Election should be added	Tester	Syed Mohammad Shah Mustafa

Test Execution Steps								
No.	Action	Inputs	Input type	Expected Output	Actual Output	Device	Result	Comment
1.	Add required fields	Profile picture, first name, last name, party, election, candidate symbol	Alphanumeric	Candidate added	Candidate added	Windows	Pass	Successfully candidate added

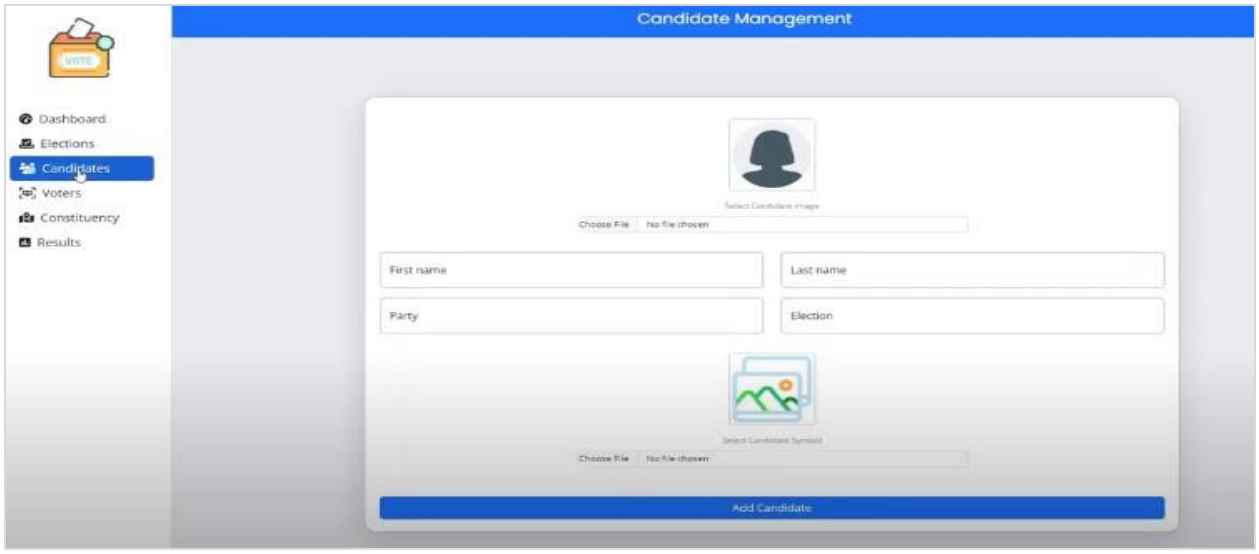


Table 40 FT_05

Test Case ID	FT_05	Test type	Functional

Description	View admin dashboard	**Test priority**	High
Pre-requisite	Admin must be logged in	**Tester**	Syed Mohammad Shah Mustafa

Test execution Steps:

No.	Action	Inputs	Input type	Expected Output	Actual Output	Device	Result	Comment
1.	Click on dashboard button from side menu			Dashboard viewed	Dashboard viewed	Windows	Pass	Successfully viewed dashboard

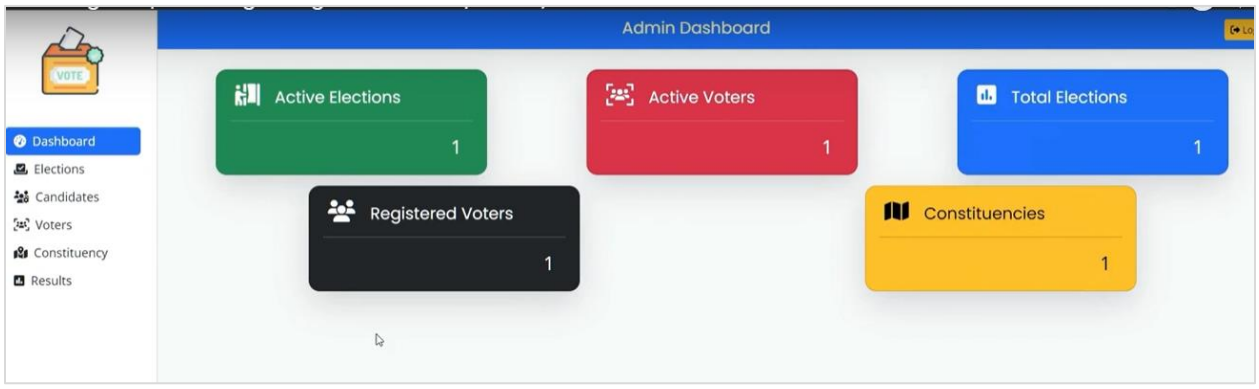


Table 41 FT_06

Test Case ID	FT_06	Test type	Functional
Description	Voter registration	**Test priority**	High
Pre-requisite	Valid email and national	**Tester**	Syed Mohammad Shah Mostafa

	ID							
Test execution Steps:								
No.	**Action**	**Inputs**	**Input type**	**Expected Output**	**Actual Output**	**Device**	**Result**	**Comment**
1.	Click on register button	Full name, national ID, email, password, OTP, constituency	Alphanumeric	Voter registered	Voter registered	Windows	Pass	Successfully voter is registered

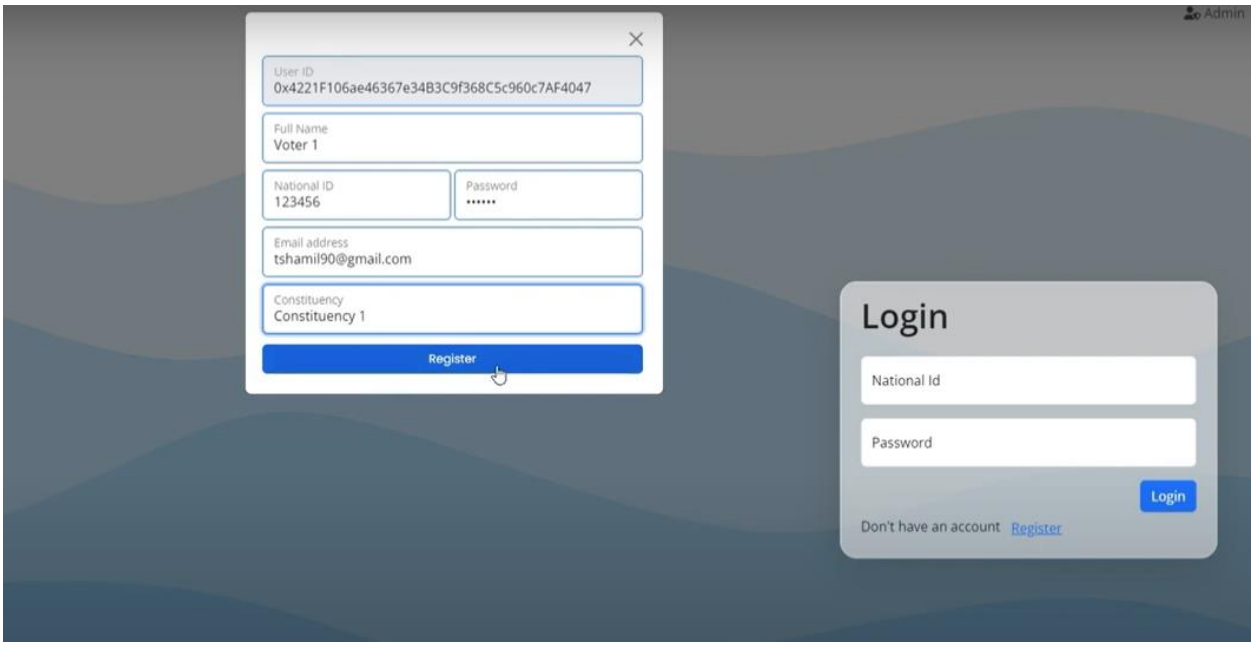


Table 42 FT_07

Test Case ID	FT_07	**Test type**	Functional
Description	Voter Login	**Test priority**	High

Pre-requisite	User ID must be verified by Admin	Tester	Syed Mohammad Shah Mustafa

Test Execution Steps								
No.	**Action**	**Inputs**	**Input type**	**Expected Output**	**Actual Output**	**Device**	**Result**	**Comment**
1.	Fill the fields	National ID, password	Alphanumeric	Logged In	Logged In	Windows	Pass	Successfully user logged in

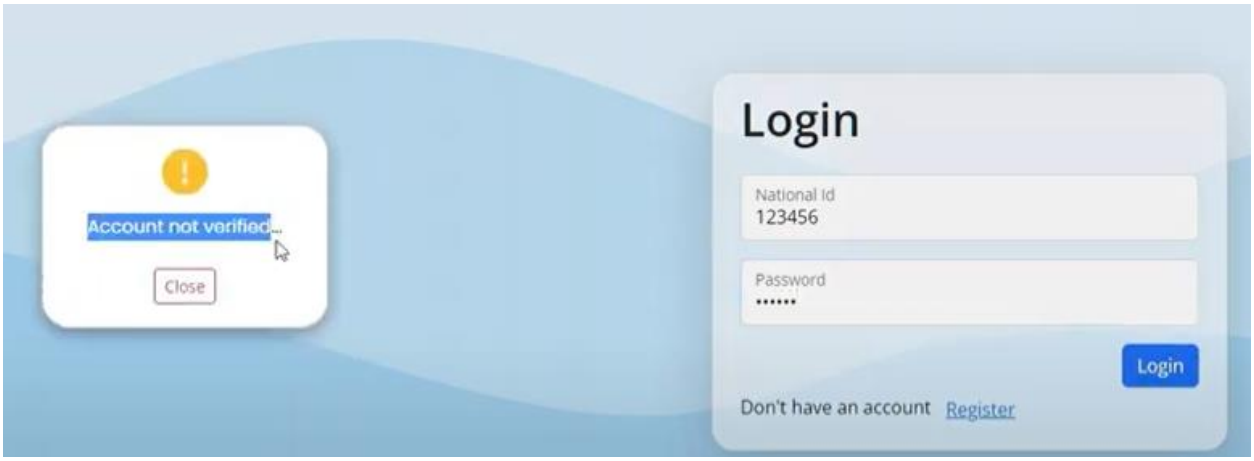


Table 43 FT_08

Test Case ID	FT_08	Test type	Functional
Description	Cast election	Test priority	High
Pre-requisite	Election must be in active state.	Tester	Syed Mohammad Shah Mustafa

Test execution Steps:								
No.	**Action**	**Inputs**	**Input type**	**Expected Output**	**Actual Output**	**Device**	**Result**	**Comment**

1.	Click on vote button	Select candidate		Vote casted	Vote casted	Windows	Pass	Successfully added the vote

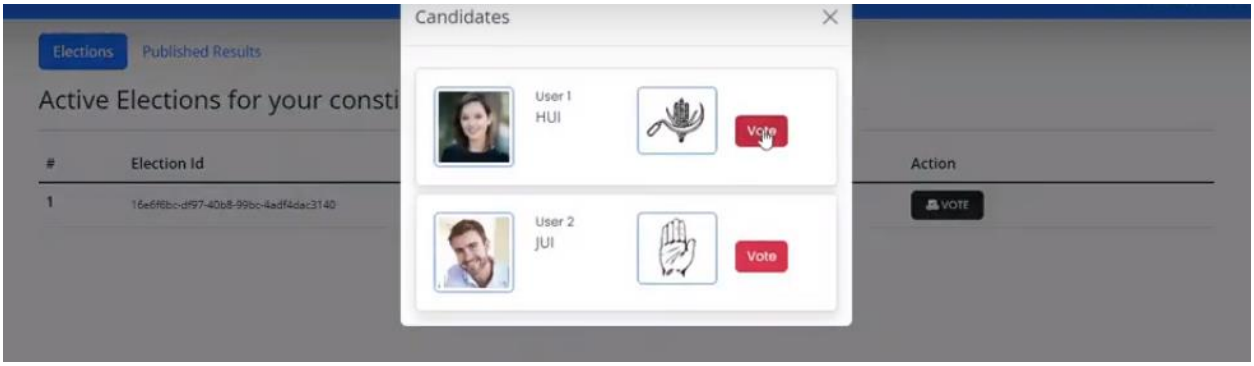


Table 44 FT_09

Test Case ID	FT_09			**Test type**		Functional		
Description	Generate result report			**Test priority**		High		
Pre-requisite	Logged in admin			**Tester**		Syed Mohammad Shah Mustafa		
Test execution Steps:								
No.	**Action**	**Inputs**	**Input type**	**Expected Output**	**Actual Output**	**Device**	**Result**	**Comment**
1.	Click on generate button			Report generated	Report generated	Windows	Pass	Successful

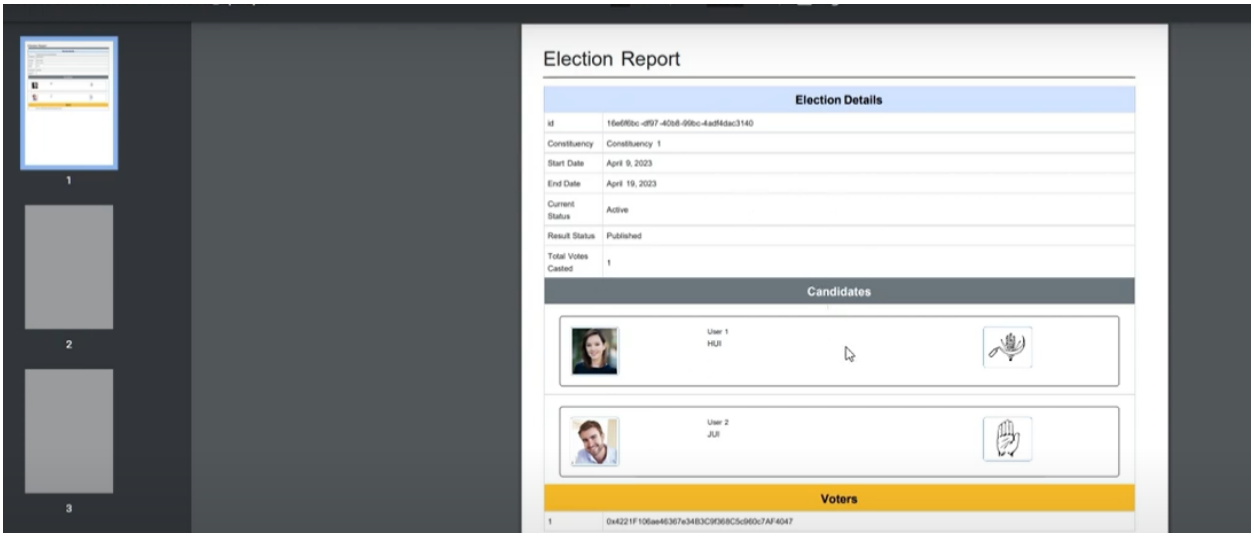


Table 45 FT_10

Test Case ID	FT_10	**Test type**	Functional
Description	Admin Logout	**Test priority**	High
Pre-requisite	Admin must be logged in	**Tester**	Syed Mohammad Shah Mustafa
Test execution Steps:			

No.	**Action**	**Inputs**	**Input type**	**Expected Output**	**Actual Output**	**Device**	**Result**	**Comment**
1.	Click on logout button			Logged out	Logged out	Windows	Pass	Successfully logged out

Table 46 FT_11

Test Case ID		FT_11		Test type			Functional	
Description		View published result		Test priority			High	
Pre-requisite		Report must be published by admin		Tester			Syed Mohammad Shah Mustafa	
Test Execution Steps								
No.	**Action**	**Inputs**	**Input type**	**Expected Output**	**Actual Output**	**Device**	**Result**	**Comment**
1.	Click on view button			Report viewed	Report viewed		Pass	Successful

12.2.2 User Acceptance Testing

Table 47 UAT_01

Test Case Id		UAT_01		Test Type			User Acceptance
Description		Registration Functionality		Test Priority			High
Pre-requisite		User on registration page		Tester			User_01
Test Execution Steps:							
No.	**Business Requiremen**	**Acceptanc e Criteria**	**Test Data**	**Test Scenario**	**Expected Outcome**	**Actual outcome**	**Status Fail/Pas**

	t						s
1.	User should be able to register him/herself before logging in.	User can fill out signup form and register his account.	**Name:** ABC **National ID:** 16102-1212-213 **Pass:** user123 **Email:** ab1@gmail.com **Constituency:** xyz **OTP:** 6754	1. Go to the website. 2. Enter Name, National id, constituency and password. 3. Click on the Register button. 4. Verify OTP.	User Registered.	Voter Registered.	Pass
Comments	The system should be able to send OTP to user's email. No issues found.						

Table 48 UAT_02

Test Case Id	UAT_02		**Test Type**		User Acceptance	
Description	Login Functionality		**Test Priority**		High	
Pre-requisite	User on verified		**Tester**		User_01	
Test Execution Steps:						
No.	**Business Requirement**	**Acceptance Criteria**	**Test Data**	**Test Scenario**	**Expected Outcome**	**Actual outcome**	**Status Fail/Pass**
1.	User should be able to Login with valid credentials	User can enter national Id and password and Log-in.	**National ID:** 16102-1212-213 **Password:** user123	1. Go to the website. 2. Enter National id and password. 3. Click on the login	User login Successfully.	User Logs in Successfully.	Pass

				button.			
Comments:	No Issues Found.						

Table 49 UAT_03

Test Case Id		UAT_03		**Test Type**		User Acceptance	
Description		Voting functionality		**Test Priority**		High	
Pre-requisite		User is verified		**Tester**		User_01	
Test Execution Steps:							
No.	**Business Requirement**	**Acceptance Criteria**	**Test Data**	**Test Scenario**	**Expected Outcome**	**Actual outcome**	**Status Fail/Pass**
1.	User should be able to Vote for his favorite Candidate in the election.	User can cast vote for a candidate in a started election.		1. Login to website. 2. Click on election. 3. Select a candidate. 4. Click vote.	Voted Successfully.	Voted Successfully.	Pass
Comments:	No Issues Found.						

Table 50 UAT_04

Test Case Id		UAT_04		**Test Type**		User Acceptance	
Description		View published result functionality		**Test Priority**		High	
Pre-requisite		User is logged in		**Tester**		User_01	
Test Execution Steps:							
No.	**Business Requirement**	**Acceptance Criteria**	**Test Data**	**Test Scenario**	**Expected Outcome**	**Actual outcome**	**Status Fail/Pass**
1.	User should be able to View Published result.	User can view election result in		1. Go to Publish result section.	Election result is displayed.	Election result is displayed	Pass

		publish result page.		2. View election result.			
Comments:	No Issues Found. The election result should be published by the admin.						

Table 51 UAT_05

Test Case Id	UAT_05			**Test Type**		User Acceptance	
Description	Login Functionality			**Test Priority**		High	
Pre-requisite	Admin on login page			**Tester**		Admin_01	
Test Execution Steps:							
No.	**Business Requirement**	**Acceptance Criteria**	**Test Data**	**Test Scenario**	**Expected Outcome**	**Actual outcome**	**Status Fail/Pass**
1.	Admin should be able to Login with valid credentials	Admin can enter national Id and password and Log-in.	**Username:** admin **Password:** admin123	1. Go to the website. 2. Enters Username and password. 3. Click on the login button.	Logged in Successfully.	Logged in Successfully. Display Admin Dashboard.	Pass
Comments:	No Issues Found.						

Table 52 UAT_06

Test Case Id	UAT_06	**Test Type**	User Acceptance
Description	Add election Functionality	**Test Priority**	High
Pre-requisite	There must be at least one constituency.	**Tester**	Admin_01
Test Execution Steps:			

No.	Business Requirement	Acceptance Criteria	Test Data	Test Scenario	Expected Outcome	Actual outcome	Status Fail/Pass
1.	Admin should be able to Add new Elections.	Admin can assign start date, end date and constituency.	**Start date:** 10/11/23 **End date:** 15/11/23 **Constituency:** constituency_1	1. Log in to website. 2. Go to election page. 3. Click on the add election button. 4. enter start date, end date and constituency. 5. Click on Save.	Successfully add election.	Show new added election in election's list.	Pass
Comments:	No Issues Found.						

Table 53 UAT_07

Test Case Id	UAT_07	**Test Type**	User Acceptance
Description	Add constituency Functionality	**Test Priority**	High
Pre-requisite	Admin is logged in.	**Tester**	Admin_01

Test Execution Steps:

No.	Business Requirement	Acceptance Criteria	Test Data	Test Scenario	Expected Outcome	Actual outcome	Status Fail/Pass
1.	Admin should be able to Add new	Admin can assign start date, end date and	**Constituency:** constituency_1	1. Go to election page. 2. Enter	Successfully add Constituency.	Show new added Constituency in list.	Pass

	Constituency.	constituency.		constituency name. 3. Click on Save.			
Comments:	No Issues Found. Admin can see constituency when adding election.						

Table 54UAT_08

Test Case Id	UAT_08		**Test Type**		User Acceptance	
Description	Add candidate Functionality		**Test Priority**		High	
Pre-requisite	Admin is logged in.		**Tester**		Admin_01	
Test Execution Steps:						
No.	**Business Requirement**	**Acceptance Criteria**	**Test Data**	**Test Scenario**	**Expected Outcome**	**Actual outcome**	**Status Fail/Pass**
1.	Admin should be able to Add new Candidates.	Admin can upload candidate's picture and symbol, he enters his name, party and constituency.	**Name:** bob **Party:** BJP **Picture:** file1 **Symbol:** file2 **Constituency :** con_1	1. Go to Candidate page. 2. Enter Name, party, constituency name and also add profile picture and symbol image. 3. Click on Add Candidate.	Candidate added successfully.	Candidate added.	Pass
Comments:	No Issues Found. Admin can see candidate when adding election.						

Table 55 UAT_09

Test Case Id		UAT_09		**Test Type**		User Acceptance	
Description		Generate report Functionality		**Test Priority**		High	
Pre-requisite		Admin is on election page.		**Tester**		Admin_01	
Test Execution Steps:							
No.	**Business Requirement**	**Acceptance Criteria**	**Test Data**	**Test Scenario**	**Expected Outcome**	**Actual outcome**	**Status Fail/Pass**
1.	Admin should be able to generate election report.	Admin can generate election's report in pdf format.		1. Click on generate button.	Generate election report successfully.	Election Report is downloaded in pdf form.	Pass
Comments:	No Issues Found.						

Table 56 UAT_10

Test Case Id		UAT_10		**Test Type**		User Acceptance	
Description		Publish result Functionality		**Test Priority**		High	
Pre-requisite		Admin is on election page.		**Tester**		Admin_01	
Test Execution Steps:							
No.	**Business Requirement**	**Acceptance Criteria**	**Test Data**	**Test Scenario**	**Expected Outcome**	**Actual outcome**	**Status Fail/Pass**
1.	Admin should be able to publish election result.	Admin can generate election's report in pdf format.		1. Click on Publish button.	Publish election result successfully.	Published Election result is displayed in result section.	Pass
Comments:	No Issues Found. Voter can view published result.						

Table 57 UAT_11

Test Case Id	UAT_11	**Test Type**	User Acceptance
Description	Start/End election Functionality	**Test Priority**	High
Pre-requisite	Admin is on election page.	**Tester**	Admin_01

Test Execution Steps:							
No.	**Business Requirement**	**Acceptance Criteria**	**Test Data**	**Test Scenario**	**Expected Outcome**	**Actual outcome**	**Status Fail/Pass**
1.	Admin should be able to start and end election.	Admin can Start election and stop it after ending time.		1. Click on Start button to start election. 2. Click on stop to end election	Election started successfully. Election stopped successfully.	Election started. Election stopped.	Pass

Comments:	No Issues Found. Voter can participate in started election. Voter cannot vote in stopped election.

12.2.3 Integration Testing Execution

Table 58 IT_01

Test Case ID	IT_01	**Test type**	Integration
Description	Candidate association with election	**Test priority**	High
Pre-requisite	Election should be added	**Tester**	Syed Mohammad Shah Mostafa

Test execution Steps:

No.	Action	Inputs	Input type	Expected Output	Actual Output	Device	Result	Comment
1.	Add required fields	Profile picture, first name, last name, party, election, candidate symbol	Alphanumeric	Candidate added in election	Candidate added in election	Windows	Pass	Successfully candidate added in election

Table 59 IT_02

Test Case ID	IT_02	Test type	Integration
Description	Voters association with constituency	**Test priority**	High
Pre-requisite	Constituency should be added	**Tester**	Syed Mohammad Shah Mostafa
Test execution Steps:			

No.	Action	Inputs	Input type	Expected Output	Actual Output	Device	Result	Comment
1.	Fill field	Add constituency name	Alphanumeric	Voter registered in specific constituency	Voter registered in specific constituency	Windows	Pass	Successful

12.2.4 Security Testing

Table 60 ST_01

Test Case ID	ST_01	**Test type**	Security
Description	Voter Login after validation	**Test priority**	High
Pre-requisite	Must be validated by admin	**Tester**	Syed Mohammad Shah Mustafa

Test Execution Steps

No.	Action	Inputs	Input type	Expected Output	Actual Output	Device	Result	Comment
1.	Fill the fields	National ID, password	Alphanumeric	Logged in after validation	Logged in after validation	Windows	Pass	Successful

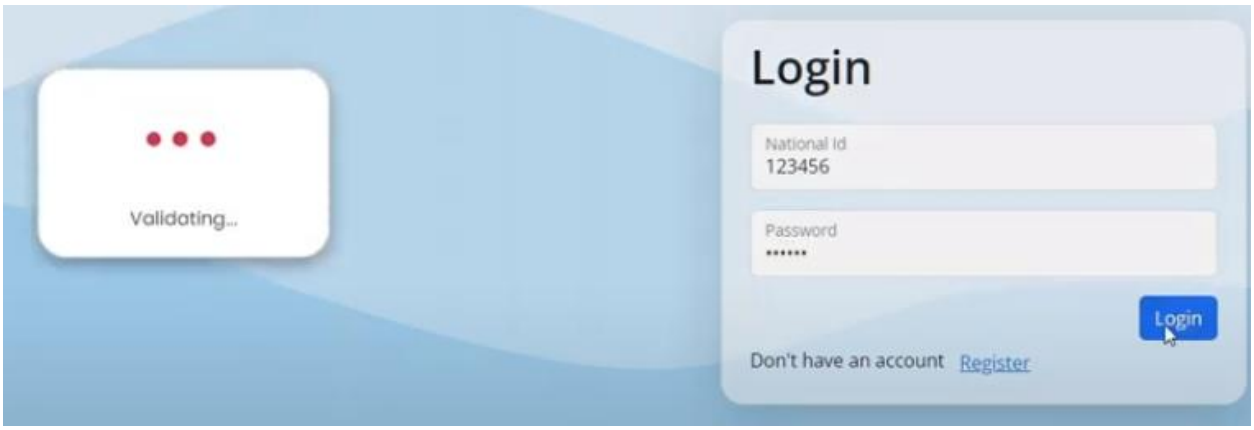


Table 61 ST_02

Test Case ID		ST_02		Test type			Security	
Description		Login with invalid credentials		Test priority			High	
Pre-requisite		Must not be registered with invalid credentials		Tester			Syed Mohammad Shah Mustafa	
Test Execution Steps								
No.	Action	Inputs	Input type	Expected Output	Actual Output	Device	Result	Comment
1.	Enter invalid credentials	National ID, password	Alphanumeric	Invalid User	Invalid User	Windows	Pass	Successful

12.2.5 Unit Testing

Table 62 UT_01

Test Case ID	UT_01	Test type	Unit
Description	Winner's name and total number of votes cast.	Test priority	High
Pre-requisite	Constituency should be added	Tester	Syed Mohammad Shah Mostafa

Test execution Steps:								
No.	**Action**	**Inputs**	**Input type**	**Expected Output**	**Actual Output**	**Device**	**Result**	**Comment**
1.	Click on "View" button			Viewed correct information	Viewed correct information	Windows	Pass	Successful

Table 63 UT_02

Test Case ID	UT_02	**Test type**	Unit
Description	Vote cast	**Test priority**	High
Pre-requisite	Election must be in active state	**Tester**	Syed Mohammad Shah Mostafa
Test execution Steps:			
No.	**Action**	**Inputs**	**Input type**	**Expected**	**Actual Output**	**Device**	**Result**	**Comment**

				Output				
1.	Click on vote button	Select candidate		Vote casted	Vote casted	Windows	Pass	Successfully added the vote

13 Deployment

The procedure through which the necessary authorities and stakeholders receive the blockchain-based electronic voting system is known as deployment. This key phase permits the system's smooth transfer into usage, empowering voters, and admin to actively participate in the voting process. This section will cover the "E-Voting Using Blockchain" framework's implementation requirements, procedure, training, and data migration (Curran k. 2018).

13.1 Deployment Criteria

The "E-Voting Using Blockchain", which is created with ease of use and accessibility in mind, must be adopted by users interested in building a reliable E-Voting system based on blockchain technology. Since it provides a crucial social service, it is always open and prepared to respond to user requests.

- Understanding the complexity of the election process is one of the key requirements for successful implementation of this E-Voting system.
- Recognize the challenges involved in creating and implementing an electronic voting system based on blockchain.
- Ensure that data is suitably translated and formatted to meet the necessary requirements
- Make sure that information and technology are seamlessly transferred from voting systems to the blockchain-based platform

13.2 Successful Deployment steps

The application must follow the steps below to properly implement a blockchain based E-Voting system that can handle a significant amount of user requests:

- Create a thorough plan for putting the E-Voting system in place.
- Create a well-thought-out system design to guarantee the system's usability and accessibility.
- Carefully assemble the entire E-Voting program.
- Carefully test the E-Voting technology to confirm its security and dependability.
- Give voters and administrator thorough training.
- Implement the E-Voting system in accordance with a planned and well-defined framework.
- Commit to keeping the E-Voting system up to date and doing daily maintenance.

13.3 Deployment Method

Following a systematic implementation methodology is crucial to the effective deployment of an E-voting system (Gagan Gurung's 2020) "seven-step process" offers an architecture for deployment. This procedure goes like this:

13.3.1 Planning:

Create a thorough software implementation plan for the deployment of the E-Voting system. A well-planned deployment is necessary for organization.

13.3.2 Analysis:

Analyze the procedural complications that come with computerized voting. Examine the voting system's specific criteria, such as security, openness, and accessibility.

13.3.3 Design:

Develop a well-thought-out and strategic design for the electronic voting system that successfully uses blockchain technology. Make sure to incorporate security, usability, and transparency into the system's design.

13.3.4 Development:

Using precise and industry-best blockchain development techniques, build the whole E-voting application. The building of a secure blockchain network, the design and testing of smart contracts, and the creation of a user-friendly voting interface are all part of this phase.

13.3.5 Testing:

Test the system thoroughly to find any weaknesses and fix them. To guarantee the dependability and security of the E-Voting system, rigorous testing is essential.

13.3.6 Deployment:

The E-Voting system is now prepared to go live after all the prior processes, including planning, analysis, design, development, testing, and training, have been successfully accomplished. The system will therefore be turned on, enabling voters to use it during elections.

13.3.7 Maintenance:

After the E-Voting system has been deployed, this step is a continuing step. The system needs constant upkeep and updates to maintain its dependability and efficiency.

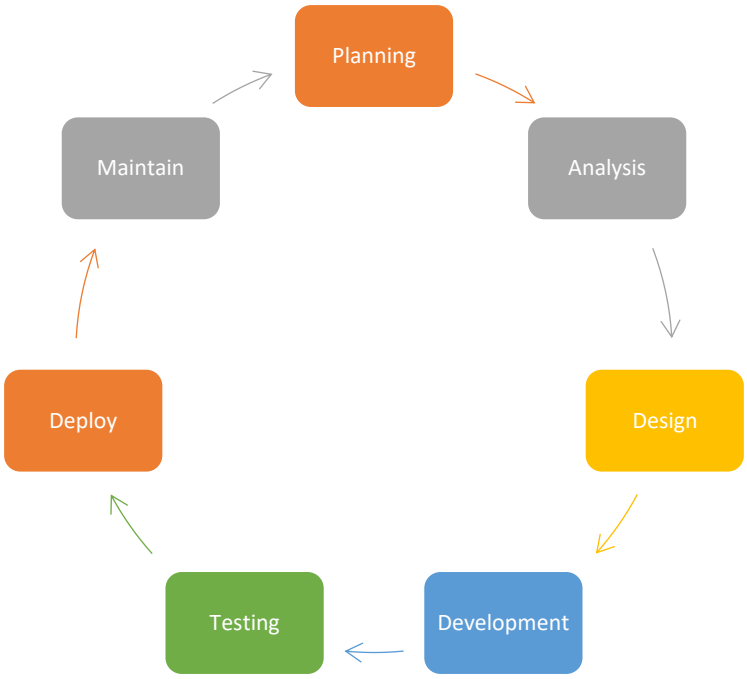


Figure 13.1 The Seven steps Process

13.4 Training

Effective control and maintenance procedures must be made while the E-Voting system is still being developed (Evin Curran1, 2018). Users' training is crucial to helping them to understand the system. Good training is necessary to address user issues and improve system openness. The goal of this training program is to enable a complete understanding of the E-Voting system. Voters and admin will participate in this training session.

The following is a summary of a successful training program designed to give thorough education and guarantee of a positive E-Voting experience:

Table 64 Plan of Training

ID	User	Module	Duration (Min)	Comment
1	General Users	Register, Login	30	General users will understand the process
2	Admin	Dashboard	20	Admin will be able to

				utilize dashboard
3	Admin	Add election, candidates, constituency	60	Admin will understand the process
4	Admin	Generate result report	15	Admin will be able to publish and generate report
5	Voter	View result report	15	Voter will be able to view report
6	General Users	Logout	15	General users will understand the process

13.5 Data Migration

Traditional voting techniques frequently relied on manual procedures for voter identification and fraud detection prior to the creation of the modern E-voting system. Traditionally, voter IDs had to be carefully verified.

Unlike such systems, our cutting-edge E-Voting program uses technologies, like blockchain, to automatically identify and detect qualified voters.

However, in order to maximize the functionality of the newly designed system, all existing manual records and data must be converted to the fully digitalized E-Voting system. Any paper-based records must be manually entered, and any existing digital data must go through a systematic database migration process

14 Evaluation

The evaluation procedure serves as vital for examining the solution and comprehending the system's qualities and results. This phase collects and analyses data from the developed system to make informed judgements about how to improve its performance, dependability, and overall effectiveness.

14.1 Product Evaluation

This evaluation procedure will be useful in determining the dependability and performance of the developed product. It will be used to determine whether all project criteria are effectively contributing to the achievement of the project's objectives. Furthermore, this evaluation will be critical in finding any hardware malfunctions or software faults and enabling their resolution. Furthermore, it will aid in identifying places where system enhancements may be required, should such requirements arise. The next section discusses the product evaluation of the e-voting application.

14.1.1 Usability

A complete set of usability requirements has been devised to prioritize accessibility, user-friendliness, efficiency, and overall user satisfaction in evaluating the e-voting system. To be accessible for all voters, including those with impairments, the system must adhere to accessibility guidelines by providing alternative input methods and screen reader compatibility. The user interface (UI) should be simple and consistent across devices, allowing for easy navigation and interactions. Voter authentication techniques that are efficient and safe should be implemented, while minimizing additional complexity that may deter users. Ballot design should be user-friendly and error-free, allowing for simple selection processes. To improve the voter experience, the system should provide clear error notifications and rapid feedback for problem rectification. It should also prioritize security awareness, educate voters on security measures, and maintain stable performance to avoid technological outages. Multilingual support and translation options should be available to satisfy a wide range of user preferences. Regular usability testing and feedback collecting should be an ongoing process to identify and address any usability issues as soon as possible, ensuring that the system remains user-centric and efficient.

Figure 14.1 User Friendly Icons

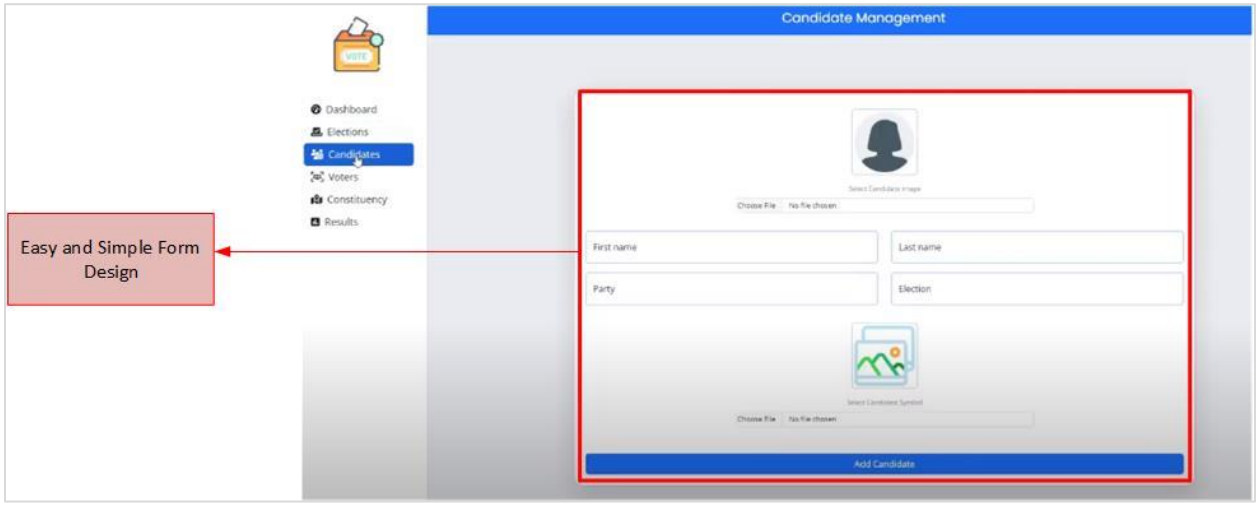


Figure 14.2 Easy and Simple Form Design

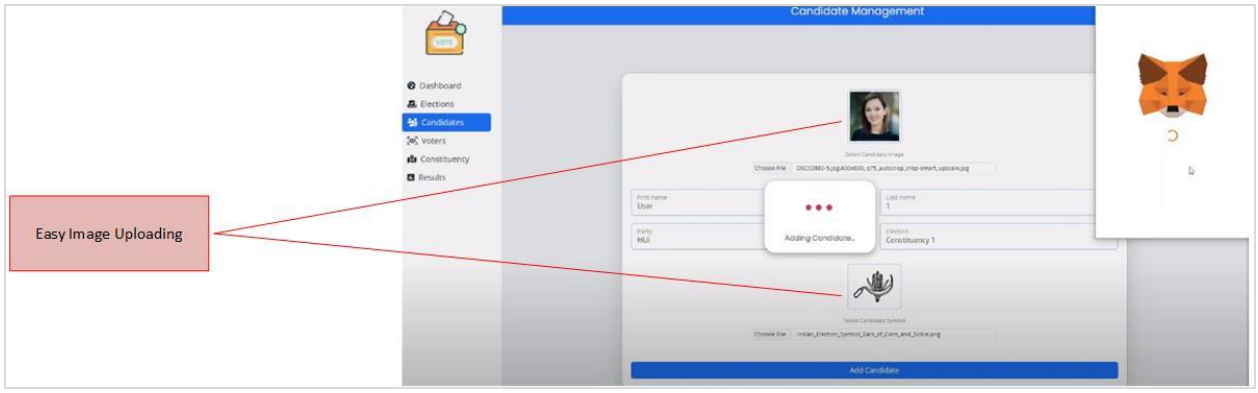


Figure 14.3Easy Image Uploading

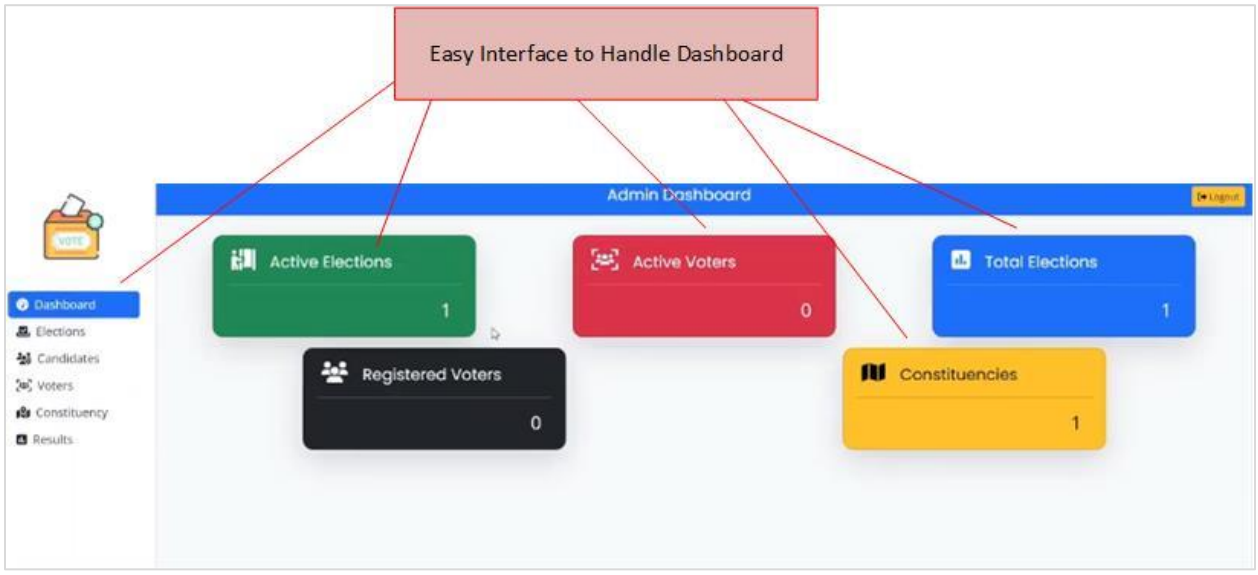


Figure 14.4 Easy Interface to Handle Dashboard

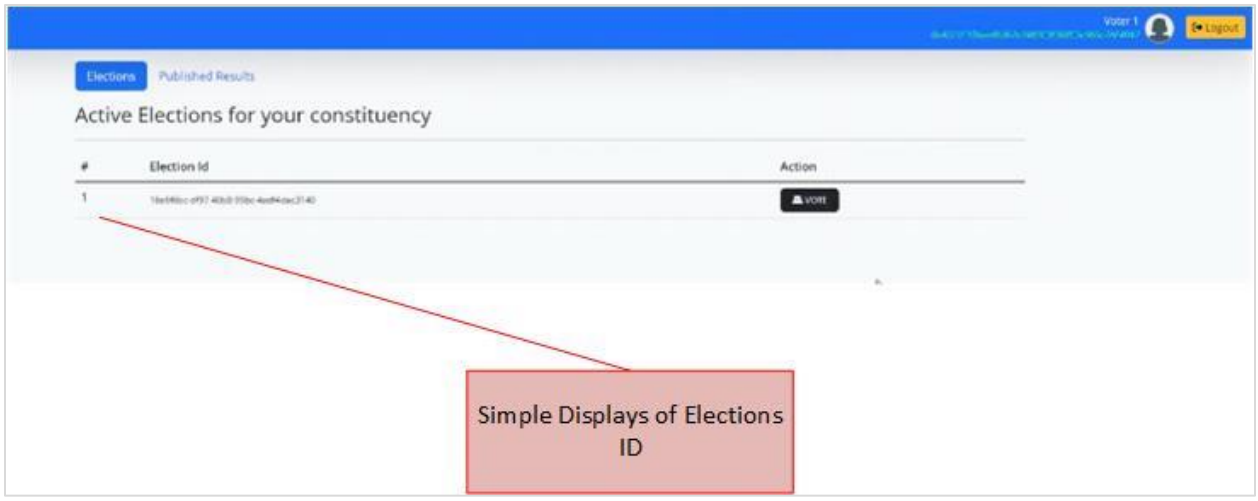


Figure 14.5 Simple Display of Elections

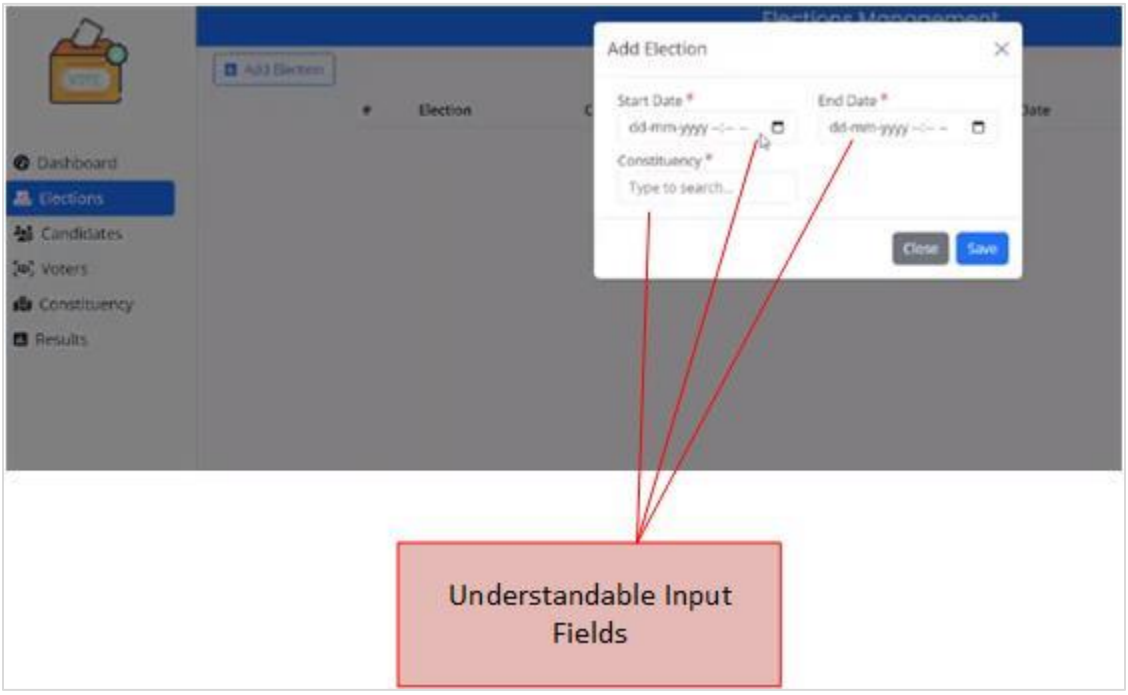


Figure 14.6 Understandable Input fields

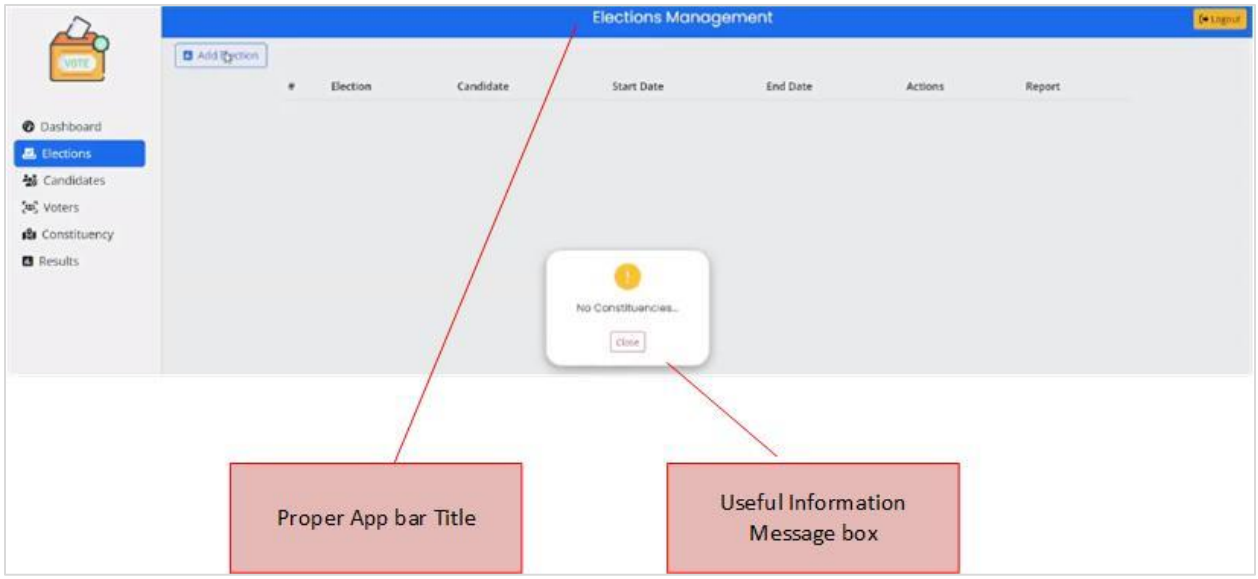


Figure 14.7 Proper App Bar Title

14.1.2 Sustainability

In the context of an e-voting system, sustainability refers to a holistic strategy to assure the system's long-term viability, ecological responsibility, and societal influence. Several essential elements must be addressed to attain sustainability:

Environmental Sustainability:

The e-voting system should have a low environmental impact. This includes lowering energy usage and supporting environmentally friendly practices in system design, operation, and maintenance. Environmental sustainability requires the use of energy-efficient hardware, data centers powered by renewable energy sources, and ethical disposal of electronic components.

Longevity and scalability:

A long-term e-voting system should be built to be adaptable to changing technology and electoral needs. Scalability should be supported to meet expanding voter demographics and developing election processes. Regular upgrades and system improvements should ensure its relevance in the future.

User Training and Education:

Maintaining an electronic voting system necessitates regular user training and education. Voters, election officials, and administrators should be taught about system usage, security best practices, and electoral procedures on a regular basis to increase their trust in the system and participation.

Transparency and Open Standards

Sustainability is inextricably linked to transparency. Implementing open standards and practices that enable independent audits and public inspection strengthens system confidence. The publishing of system documentation and code can improve transparency even more.

14.1.3 Maintainability

Regular device maintenance is required to ensure optimal performance throughout the device's lifecycle. This practice is critical for measuring the application's efficiency and the necessity for upgrades. Multiple tests are required to maintain consistency and ensure that the system operates properly.

15 Critical Appraisal

In this section, we will examine the e-voting application's features and functionalities, which allow it to accomplish its intended goals and objectives. We will present explanation for each of the system's objectives throughout this discussion, analyzing which have been reached satisfactorily and which may require additional attention. We will also determine which features have been addressed and efficiently used, as well as any that may still require implementation. Importantly, we will critically examine the established system's strengths and shortcomings, providing a thorough study of its overall performance and capabilities.

15.1 Achieved Objectives

A set of stated objectives was developed from the beginning of the development of this application. However, due to the nature of one-man development and changing project dynamics, not all of these goals were met. It is worth noting that the majority of the application's objectives have been met with success. The following goals have been met and realized:

- **Administration Account:** The system successfully incorporates an administration account, allowing authorized staff to conveniently manage and oversee the voting process.
- **Voter Registration:** The application enables for the registration of eligible voters, making it easier for people to participate in the election process.
- **Secure Authorization:** Strict security procedures are in place to provide secure authorization, protecting the voting system's integrity from unauthorized access.
- **Election Voting:** The system efficiently facilitates electronic voting, giving voters with an accessible and efficient way to participate in elections.
- **Election Results:** The system precisely tabulates and manages election results, assuring the outcome's transparency and integrity.
- **Transparency:** A fundamental goal that has been realized is transparency, with measures in place to ensure that the entire voting process is accessible and accountable, fostering trust in the system.

15.2 Objectives those were Partially achieved

Throughout the system implementation, the following objectives were only partially achieved:

- **Blockchain Technology:** While blockchain technology was used for openness and security, it is possible that some components of the system's blockchain implementation were not completely utilized or optimized for transparency and auditability.
- **Business Statistic Analytics**

- **Integration with External Systems:** The goal of smooth integration with external systems, such as voter registration databases or identity verification services, may have encountered difficulties, resulting in partial integration or data exchange constraints.

In complicated projects such as e-voting systems, partial attainment of specific objectives during system installation is relatively unusual. It is critical to understand the causes behind these partial successes and to identify viable solutions. Here's an examination of the stated goals, the likely causes for their partial fulfilment, and potential actions:

Reason and possible course of action:

Blockchain Technology: Due to the complexity of blockchain systems, resource constraints, or a requirement for further knowledge in blockchain deployment, partial use of blockchain technology for transparency and auditability may occur. In order to address this:

- Conduct an in-depth examination of the present blockchain implementation to find areas that can be optimized for more transparency.
- Invest in training or consulting with blockchain professionals to fully harness the technology's transparency and security features.
- To enable full auditability, blockchain initiatives should be updated and aligned with best practices on a regular basis.

Business Statistic Analytics: A lack of suitable data analytics tools or skills may be ascribed to the partial attainment of business statistic analytics aims. To improve the analytics of corporate statistics:

- Invest in advanced data analytics tools and software to process and analyze voting data better effectively.
- Provide training to staff working in data analytics to ensure they can fully utilize available tools.
- To streamline the analytics process, work with data experts or data science professionals.

Integration with External Systems: Integration difficulties with external systems, such as voter registration databases or identity verification services, may have stemmed from data format discrepancies, security concerns, or technological compatibility issues. To enhance integration:

- Conduct a thorough evaluation of the integration process, finding any bottlenecks or problems.
- Collaborate with third-party system vendors to develop standardized data interchange formats and protocols.

- Enhance security processes and methods to ensure the safe and seamless exchange of data between systems.

15.3 Objectives Totally not achieved

The subsequent goals were not attained:

- E-ballot for separate categories
    - E-ballot for schools
    - E-ballot for Business
    - E-ballot for essentials
- Anonymous & weighted voting options
- Product knowledge base
- Email support

Reason and possible course of action:

E-Ballot for Separate Categories: Due to the difficulties of designing and operating separate voting systems for each category, the goal of deploying e-ballots for unique categories, such as schools, businesses, and critical services, may not have been fully realized. To solve this, one option is to streamline the process by designing a more versatile and modular e-voting system that allows for the formation of separate categories inside a single platform. This would increase the system's versatility and usefulness, allowing for efficient voting in a variety of areas.

Anonymous and weighted voting choices: Achieving anonymous and weighted voting alternatives may have been difficult due to user identification and complex voting algorithms. To remedy this, one option is to refine the system's design to include robust anonymity features and advanced algorithms for weighted voting. To maintain voter privacy and proper vote weighing, specialized software development and security measures would be required.

Product Knowledge Base: Due to limits in content development or user involvement, the goal of developing a comprehensive product knowledge base may not have been fully realized. A possible course of action would be to actively expand and update the knowledge base with useful product information and FAQs. Furthermore, encouraging user participation through user-generated material or community contributions can increase the depth and usefulness of the knowledge base.

Email Support: If email support goals were not completely completed, it could be due to response time delays or resource restrictions. Streamlining the support process, allocating specialized support workers, and adopting systems for quick ticket monitoring and resolution are all activities that should be taken to improve email support. A service-level agreement (SLA) could also be explored to assure fast responses.

15.4 Features those were Touched

The following features are being touched:

- **Voter Registration:** The system contains a voter registration feature that allows eligible individuals to register and verify their eligibility to vote.
- **Secure Authentication:** The system is protected by strong security mechanisms that enable secure authentication.
- **Electronic Voting:** The application effectively enables electronic voting by allowing voters to cast ballots online via a user-friendly and straightforward interface.
- **Election Results Management:** The system includes capabilities for accurately handling and tabulating election results, assuring the outcome's integrity and transparency

15.5 Features those were partially touched

The following features are being partially touched:

- Real time result reporting
- User Authentication

Reason and possible action:

Reason: Partial user authentication implementation may be owing to problems in confirming user identities, security concerns, or user management issues.

Actions that could be taken:

- For enhanced security, strengthen user authentication processes by introducing multi-factor authentication (MFA).
- Review and improve security methods to ensure that user data is adequately protected.
- Invest in user management solutions that can verify and manage user identities and access permissions efficiently.
- For seamless user authentication, provide user-friendly interfaces and clear instructions.

15.6 Features those were not touched

Features those were not touched

- Custom voting method calculations
- Phone touchpad & paper voting
- Single-sign-on
- E-ballot for separate categories

  - E-ballot for schools
  - E-ballot for Business
  - E-ballot for essentials
- Anonymous & weighted voting options
- Product knowledge base
- Email support

Reason and Possible Action:

Due to several issues, several essential functionalities were not included in the e-voting application. These include the lack of custom voting method computations, phone touchpad and paper voting options, and single sign-on functionality. The reasons for these absences range from a lack of resources to a preference for computerized voting techniques. To close these gaps, the system may require extensive development efforts, such as the creation of unique interfaces for alternative voting techniques and the integration of external identity providers for single sign-on. Furthermore, features such as distinct e-ballots for schools, businesses, and necessities, anonymous and weighted voting alternatives, a product knowledge base, and email assistance were not included. To make these features a reality, there must be a concentrated focus on understanding unique requirements, technological complexity, and resource allocation. By identifying these areas for improvement, the system can be improved to deliver a more thorough and user-friendly voting experience.

15.7 Strength of the system

An e-voting system, often known as electronic voting, has the potential to improve the election process by making it more efficient, accessible, and transparent. Here are some of an e-voting system's primary advantages:

- **Accessibility:** E-voting technologies can make voting easier for a wider group of voters, including individuals with impairments and those who may have trouble getting to regular polling places.
- **Ease of use:** Electronic voting can simplify the voting process by minimizing long lines and waiting periods, encouraging better voter turnout and involvement.
- **Accuracy:** Electronic voting systems can reduce human errors in vote counting and result tabulation, resulting in more accurate election results.
- **Remote Voting:** E-voting technologies provide remote voting, allowing citizens who are travelling or residing overseas to participate in elections.
- **Quick Turnaround:** Because votes are transmitted electronically, the time it takes to finalize and announce election results is reduced.

- **Scalability:** Because e-voting systems can accommodate a high number of voters, they are appropriate for elections at all levels, from municipal to national.
- **Environmental Impact:** Electronic voting eliminates the need for paper ballots, resulting in less paper waste and a lower environmental footprint.

15.8 Weakness of the System

- **Concerns about security:** E-voting systems are subject to cyberattacks and hacking, which could jeopardize the confidentiality and integrity of votes and election outcomes.
- **Digital Divide:** Because not all citizens have equal access to technology or the internet, there is concern that certain populations will be excluded from the voting process.
- **Complexity:** E-voting systems can be difficult to build and manage, necessitating technical and cybersecurity knowledge.
- **Costs:** The initial setup and continuous maintenance of e-voting systems can be costly, requiring a significant investment.
- **Challenges with User Experience:** Users, particularly those who are not tech-savvy, may encounter usability challenges that limit their ability to vote efficiently.
- **Reliability:** Technical difficulties, network failures, or system issues might cause voting to be disrupted.

15.9 Future Development

The following are possible future developments:

- Voter notifications
- Real-time results
- Anonymous & weighted voting options
- Product knowledge base
- Email support
- Single-sign-on
- Certified results report
- Custom Layout Options
- Results Export (PDF)
- Results Export (Row Data)

16 Lessons Learned

As a lone developer, creating an e-voting system provided several serious challenges, ranging from the intricacy of the technology involved to the sheer size of the project. These hurdles necessitated ingenuity, a dedication to constant learning, and an unyielding desire to see the project through to completion.

16.1 What Have I Learned

Because this was an academic assignment, I assumed total responsibility for its completion. As a result, this endeavor has been a wonderful learning experience. It drove me to learn the abilities needed to do study on topics that necessitate technical growth.

As a result, I had to develop clear business objectives for my project, with a dual focus - first, evaluating challenges related to my own country, and then broadening the scope to include global aspects. To accomplish this, I painstakingly examined recent news from many sources to obtain a thorough grasp. After deciding on the issue of designing a system for 'automatic management of the voting system,' my path took me into the area of thorough research. To do so, I diligently collected raw data before doing a thorough analysis of previous academic research publications. This was a deliberate decision to avoid duplicating procedures that had already been verified in the context of internationally recognized investigations. Throughout this process, I actively attempted to discover gaps in the existing body of knowledge and investigated ways for filling these gaps.

Following the rigorous definition of all project features, the critical issue of picking the most compatible technology for the application occurred. This was critical to ensuring extensive community adoption and attracting a large user base. Furthermore, creative methods were required to facilitate application development. The framework for the entire development process had to be created, including essential components such as the test plan, risk management strategy, change management plan, quality management plan, and development tool and methodology selection. A strong understanding of software development processes, with a special emphasis on DSDM (Dynamic Systems Development Method) and Agile approaches, was required.

I discovered myself digging into the art of threat assessment and honing my risk-mitigation talents. Recognizing areas for improvement became a vital part of my approach, and I learnt to discern and analyze product quality. In this difficult process, I combined multiple methodologies and used fresh strategies to create a one-of-a-kind and bespoke development process that smoothly corresponded with the project's requirements.

Upon the completion of the initial preparatory phase, I began the path of learning the necessary skills for gathering project specifications. I dedicated myself to knowing the relevance and importance of many visual tools to comprehend the intricate working process and workflow intricacies of my project. These included understanding "rich pictures" for holistic representation, building and interpreting use case diagrams, structuring entity-relationship diagrams (ERD), depicting component diagrams, deployment diagrams, class diagrams, and sequencing events using sequential diagrams.

Furthermore, I improved my capacity to analyze and prioritize criteria depending on the project's overall goals. To ensure efficient project management, distinct time frames with prioritized specifications were established. I stepped into the area of user interface design to provide users with an upgraded and seamless experience, focusing on increasing the visual and interactive features of the application. Through hands-on experience, I realized the importance of operation diagrams, component diagrams, and deployment diagrams in the development process and overall project success.

Subsequently, my journey delved into the intricacies of iteratively creating functions within the project framework. I grasped the significance of not hesitating to pause the planning process when faced with the need for a more profound comprehension of a particular technological process. It was during these moments that I learned how to effectively engage with the community for assistance and where to explore potential solutions, leveraging the collective wisdom of experts and peers.

My education took me into the world of testing procedures. I gained a thorough awareness of various testing procedures and the complexities of precisely carrying them out. Equally important was the capacity to keep precise records of these tests, ensuring thorough documentation of the procedure. When tests produced unfavorable findings, I learnt the art of methodically diagnosing and correcting mistakes and flaws, which required patience and precision.

In one extremely difficult case, the login authorization had unexpectedly expired, and two critical live functionalities had been dormant, practically at the eleventh hour. In the midst of this dire circumstance, I had no choice but to act quickly, working tirelessly to remedy these concerns. This experience not only broadened my knowledge but also improved my problem-solving abilities and adaptability.

One of the most important skills I learned on this journey was how to produce detailed software technical documentation. This ability proved essential for effectively describing the complexities of the project, allowing for successful communication and knowledge sharing.

Finally, I learnt the art of self-evaluation, which is essential in any effort. I learned to evaluate my work objectively and make informed decisions regarding its success. This capacity to critically analyze and reflect on my work has been important in my pursuit of excellence and ongoing improvement.

16.2 What Have Problems I Faced?

The development of an e-voting system/application is a sophisticated and comprehensive endeavor that is not without its difficulties. Several major challenges arose during my tour, putting my problem-solving abilities and determination to the test. One of the most pressing issues was the question of security. It is critical to ensure the integrity and secrecy of the voting process and voter data, and the ever-changing landscape of cyber threats presents a substantial obstacle. A constant concern was the implementation of strong security mechanisms and encryption to safeguard the system from future attackers.

Another significant challenge was the requirement for inclusion. E-voting systems must cater to a varied user base, and ensuring accessibility for everybody, including people with impairments or restricted access to technology, was a difficult assignment. It was a difficult undertaking to bridge the digital divide and make the system user-friendly for all demographics.

Technical difficulties were another common issue. The complexities of software development, as well as the selection of acceptable technologies to achieve the project's objectives, necessitated a thorough understanding of the subject. Furthermore, during development, the need to fix technological problems and traverse unforeseen system issues was a regular challenge.

Furthermore, the legal and regulatory environment posed its own set of obstacles. E-voting systems must comply with a tangle of legal and regulatory standards that differ by jurisdiction. Navigating this complicated system and ensuring compliance demanded close attention to detail.

Another degree of complexity was added by resource limits, particularly financial and human resources. The initial setup and continuous maintenance of e-voting systems may be costly, and keeping the project under budget was a constant issue.

Finally, establishing and maintaining public faith in the system has been a constant problem. Voters must have confidence that their ballots are cast and counted correctly, and that the system is free of manipulation. Addressing concerns about openness and accountability was an ongoing process.

The unexpected failure of my server was a traumatic event that occurred at a vital moment in the project. With the clock ticking, I found myself in a race against the clock to answer the problem. The server went down for over two hours, which was a terrifying experience, especially because it happened at the eleventh hour. During this time, the pressure was palpable as I worked to identify and solve the problem as quickly as possible.

Another difficult obstacle I faced was having to perform all the mandatory tests on my own. This task was physically and mentally tiring, which was aggravated by the time restrictions. The urgency of the problem

added to the frustration, since it required rigorous testing, fault diagnosis, and adjustments all within a short timeline.

In overcoming these obstacles, I not only obtained a thorough awareness of the complexities of e-voting systems, but I also improved my problem-solving, technical, and regulatory abilities. The experience highlighted the necessity of tenacity and a dedication to quality in establishing a secure, inclusive, and dependable e-voting system. It was a testament to the dedication and perseverance required in the development of a robust and reliable e-voting system.

16.3 What Solutions Occurred?

In response to the difficulties, I experienced when developing the e-voting system, I created a number of ways to overcome these obstacles. I took a proactive approach to dealing with my server's unexpected failure. I not only sought quick remedies, but also took preventative precautions by updating the infrastructure of my computer. In addition, I handled two internet connections at the same time to reduce the danger of internet disconnections and ensure uninterrupted work. An exhaustive assessment of research articles was conducted to satisfy the demand for comprehensive research to navigate numerous technological challenges and to inform the development process. I sifted through a plethora of research paper references, painstakingly collected, and analyzed closely related pieces, and gleaned useful insights from the academic literature. This method enabled me to tap into the pooled expertise of specialists and provide a solid basis for the project. I investigated the study and practical application of currently widespread voting automation systems in different areas, including government, educational institutions, and important services, to better grasp the complexities of voting automation systems. This deep dive gave me significant insights into how these systems work and inspired novel methods for my project.

Recognizing the significance of good time management, the project was separated into distinct time boxes. This method ensured that the production process followed precise schedules and milestones, improving efficiency and output. In negotiating the challenges of e-voting system development, these solutions highlighted the importance of early preparation, substantial research, and a systematic approach to project management. They were a monument to the resilience and adaptation required to effectively handle and overcome hurdles in this endeavor.

17 Conclusion

17.1 Project Summary

The development of the "E-voting" application represents a big step forward in the modernization of the election process. With this digital platform in place, it is expected that all election management and automation tasks will be smoothly linked into this revolutionary system. The "E-voting" system has been

created with security, accessibility, and transparency in mind, delivering an efficient and user-friendly voting experience. It has the potential to increase voter participation, boost public trust, and contribute to a more ecologically responsible political process by complying with stringent usability, security, and sustainability criteria. As the digital era shapes the future of elections, this system serves as a sign of progress, highlighting the dedication to fair and inclusive democracy. It is evidence of the potential of technology to revolutionize the way we engage in the fundamental process of governance.

Recognizing the limits of an academic project with a limited schedule and a single developer is critical in understanding the background of the "E-voting" application's development. While the project may have defects and inadequacies, it is crucial to recognize the devotion and hard effort that went into delivering a high-quality solution given the available criteria and resources. Despite its shortcomings, this project is a good learning experience, demonstrating the constraints and complexities of establishing a digital platform for a critical civic function like voting. It emphasizes the commitment to continual improvement and the potential for future enhancements to solve the highlighted shortcomings, ultimately contributing to the advancement of digital democracy.

17.2 Project Goals

The "E-voting" project has many purposes, all of which contribute to the larger goal of modernizing and improving the election process. The fundamental goal is to create a safe, accessible, and efficient electronic voting system that empowers voters and election officials while encouraging confidence and openness. This involves safeguarding the confidentiality and integrity of votes, allowing voters of all abilities to participate, and offering a simple and user-friendly interface. The initiative also aims to reduce environmental impact by employing responsible technologies and adhering to legal and regulatory regulations. Furthermore, building public trust and engagement is a key goal, as is ensuring the system's long-term viability and adaptation. The project demonstrates a commitment to strengthening democracy and electoral practices by leveraging technology's potential to enable fair and inclusive elections.

17.2.1 Achieved Goals

- Develop complete E-voting system
- Develop Administration Module
- Develop Voter Module
- Develop user registration process through smart contracts using blockchain
- Develop Election system
- Develop Candidate module with complete add candidate simple and easy to use form
- Develop images uploading modules
- Develop Generate report system

- Develop start/End election date and time

17.2.2 Unachieved Goals

- Reset password for Voters
- User profile Management
- Live Chat box
- Voter Dashboard analytics
- Change Password
- Homepage/Landing page (Blogs, FAQs, Privacy policy, terms, and conditions)

17.3 Project Success

The effective completion of the "E-voting" project is a great accomplishment that reflects the attention and effort put into its development. The project has met its primary objectives by producing a robust e-voting system that includes an Administration Module, Voter Module, and user registration procedure supported by smart contracts enabled by blockchain technology. The Election system has been created successfully, allowing for the effective management of electoral operations. With its user-friendly candidate addition form, the Candidate Module provides a streamlined experience for anyone looking to participate in elections. The addition of picture uploading modules expands the system's functionality. Notably, the Generate Report method delivers useful insights and openness, which helps to build public trust. Finally, the project's ability to set and manage election start/end dates and times adds an important degree of control. Despite hurdles, the attainment of these aims highlights the potential of technology to revolutionize the voting process by enhancing accessibility, security, and efficiency. The completion of the project demonstrates the dedication to advancing the democratic process and the possibility of future improvements.

17.4 What has been accomplished in the documentation?

Comprehensive analysis, strategic planning, diagrammatic representations, exhaustive research, implementation, rigorous testing, and diligent evaluation have all gone into the project documentation. This extensive procedure resulted in the creation of a solid architecture for the 'E-voting' application. Please see the following section for a quick description of the documentation phases:

- We carefully analyzed earlier study efforts in the Literature Review section. This in-depth study enabled us to identify knowledge gaps and design appropriate solutions, thereby improving the overall quality of this project.

- We went into five pre-existing solutions actively used in chicken farm automation, as well as two map integration systems, in the Product Research area. We used a weighted scoring approach to assess their appropriateness.
- In the Legal, Social, Ethical, and Professional sections, I investigated the diverse implications of the e-voting application on society.
- Furthermore, I've shared insights into several approaches used in app development, as well as the specific strategy adopted for this application. Furthermore, I've expanded on the methodologies used to acquire the application's specs as well as the resources used in its development.
- I developed and discussed various plans in the Planning section. A project plan, a test plan, a risk management strategy, and a change management plan are all included.
- I investigated the project's technological feasibility, operational feasibility, and economic feasibility in the Feasibility Study phase.
- I also investigated the possible adjustments. I thoroughly depicted the organizational structure, project stakeholders, company requirements, system requirements in various formats, their respective priorities, and produced requirements catalogue in the Requirements Analysis and Specification section.
- I've included a variety of critical diagrams in the New System Design and architectural section, including a use case diagram, an Entity-Relationship Diagram (ERD), a class diagram, a sequence diagram, a component diagram, and an architectural diagram.
- The Development part delves into the e-voting system modules in depth, including high-fidelity prototypes, time constraints for requirements, and an analysis of their complexity.
- In the Testing section, various types of testing plans and their execution logs were presented. Additionally, the Deployment section delves into the strategies and preparations for a successful deployment of the system.
- The Evaluation section focused on assessing the developed product in terms of usability, sustainability, and maintainability.
- In the Lessons Learned section, I shared the valuable insights I've gained, the challenges I've faced, and the solutions I've discovered throughout the project.

17.5 My Personal Experience

Creating an e-voting system using blockchain technology was surely a difficult journey. Countless obstacles appeared to obstruct the path to project completion, leading me to consider quitting it at times. However, my family and friends' unfailing support and encouragement acted as a wellspring of inspiration, propelling me forward. To further the project, I needed to enter the thrilling domain of cutting-edge technologies that define our digital age. Immersion in research papers became a habit, a rich source of knowledge that helped define the project. As the challenges and roadblocks faded, they were

replaced by a great sense of accomplishment at the project's victorious finish. This was a remarkable experience, a monument to endurance and resilience. The success of the project, guided by the mercy of Almighty Allah, was dependent on my talents and determination, providing an opportunity to not only discover complicated problems but also analyze them and design multiple solutions. It turned into a hard test of my problem-solving abilities, ultimately strengthening my ability to overcome obstacles.

In reflection, the experience I got while working on this project will be imprinted in my memory for the rest of my life as a key milestone in my personal and professional development.

17.6 Implications for the Future

This project has significant long-term implications. The knowledge and skills I've acquired during this project will undoubtedly propel me towards achieving even greater goals in the future. In practical applications, this experience will prove invaluable for both academic research and the tech industry. I'm eager to continue improving both myself and this project so that, one day, this endeavor can truly revolutionize the transition from traditional e-voting systems to electronic ones.

18 Bibliography and References

18.1 Book Used

- Mastering Blockchain by Imran Bashir
- Blockchain Blueprint for New Economy By Melanie Swan
- HTML & CSS: Design and Build Web Sites
- Javascript By McGrawhill Publication

18.2 References

1. Yadav, A.S. (2020) “E-Voting using Blockchain Technology”, International Journal of Engineering Research, Volume 9(07), DOI: 10.17577/IJERTV9IS070183.
2. Garg, K., Saraswat, P., Bisht, S., Aggarwal, S. K., Kothuri, S. K., & Gupta, S. (2019) 'A Comparative Analysis on E-Voting System Using Blockchain', Title of the Journal (italicized), Volume(Number), Page Range. Publisher: IEEE.
3. Indapwar, A., Chandak, M., & Jain, A. (2020) 'E-Voting system using Blockchain technology', International Journal of Advanced Trends in Computer Science and Engineering, 9(3), May-June, pp. 2775-2779.
4. Murugesan, M., Pavithra, S., Preakshanashree, S., & Praveen Kumar, S. (2020) 'Rendering Untampered E-Votes Using Blockchain Technology', In Emerging Trends in Computing and Expert Technology, pp. 703-711. Publisher: [Publisher Name]. DOI: 10.1007/978-3-030-32150-5_69.
5. Jafar, U., Ab Aziz, M. J., & Shukur, Z. (2021) 'Blockchain for Electronic Voting System—Review and Open Research Challenges', Sensors, 21(17), 5874. DOI: 10.3390/s21175874.
6. Gibson, J. P., Krimmer, R., Teague, V., & Pomares, J. (2016) 'A review of E-voting: the past, present and future', Annales des Telecommunications/Annals of Telecommunications, 71(7-8), pp. 279-286. DOI: 10.1007/s12243-016-0525-8.
7. Oliver, J. E. (1996) 'The effects of eligibility restrictions and party activity on absentee voting and overall turnout', American Journal of Political Science, 40, pp. 498–513.
8. Prashar, D., Jha, N., Jha, S., Joshi, G., & Seo, C. (2020) 'Integrating IoT and blockchain for ensuring road safety: An unconventional approach', Sensors, 20, 3296.
9. M. The Democracy to Come? An Enquiry into the Vision of Blockchain-Powered E-Voting Start-Ups. Front. Blockchain 2021, 4, 17. [Google Scholar] [CrossRef].
10. Al-Maaitah, S., Qatawneh, M., & Quzmar, A. (2021) 'E-Voting System Based on Blockchain Technology: A Survey', in Proceedings of the 2021 International Conference [Title of the Conference], Publisher: IEEE.

11. Ma, X., Zhou, J., Yang, X., & Liu, G. (2020) 'A Blockchain Voting System Based on the Feedback Mechanism and Wilson Score', Information, 11, 552. [Google Scholar] [CrossRef].
12. Augoye, V., & Tomlinson, A. (2020) 'Analysis of Electronic Voting Schemes in the Real World',
13. Anjan, S., & Sequeira, J. P. (2019) 'Blockchain Based E-Voting System for India Using UIDAI's Aadhaar', Journal of Computer Science Engineering and Software Testing, Volume 5, Issue 3, e-ISSN: 2581-6969.
14. Khan, K. M., Arshad, J., & Khan, M. M. (2019) 'Secure Digital Voting System based on Blockchain Technology', 1 NED University of Engineering and Technology, Pakistan; 2 University of West London, UK.
15. Wei, P., Wang, D., Zhao, Y., Tyagi, S.K.S., & Kumar, N. (2020) 'Blockchain data-based cloud data integrity protection mechanism', Future Generation Computer Systems, 102, 902–911.
16. Poniszewska-Marańda, A., Pawlak, M., & Guziur, J. (2020) 'Auditable blockchain voting system—the blockchain technology toward the electronic voting process', International Journal of Web Grid Services, 16, 1–21. [Google Scholar]
17. Dagher, G.G., Marella, P.B., Milojkovic, M., & Mohler, J. (2018) 'Broncovote: Secure Voting System Using Ethereum's Blockchain', in Proceedings of the 4th International Conference on Information Systems Security and Privacy, Funchal, Madeira, Portugal, 22–24 January 2018. [Google Scholar].
18. Lahane, A. A., Patel, J., Pathan, T., & Potdar, P. (2020) 'Blockchain technology based e-voting system', ITM Web of Conferences 32, 03001. DOI: 10.1051/itmconf/20203203001. ICACC-2020. Juhu Versova Link Rd, behind HDFC Bank, Gharkul Society, Bharat Nagar, Versova, Andheri West, Mumbai, Maharashtra 400053.
19. Benny, A., Kumar, A. A., Basit, A., Cherian, B., & Kharat, A. (2020) 'Blockchain based E-voting System', Department of Computer Engineering, PCE, Navi Mumbai, India - 410206.
20. Hsiao, J-H., Tso, R., Chen, C-M., & Wu, M-E. (2017) 'Decentralized E-Voting Systems Based on the Blockchain Technology', in Advances in Computer Science and Ubiquitous Computing Conference, Lecture Notes in Electrical Engineering, Volume 474, First Online: 20 December 2017.
21. Pandit, P., & Tahiliani, S. (2015) 'AgileUAT: A Framework for User Acceptance Testing based on User Stories and Acceptance Criteria', International Journal of Computer Applications, Volume 120, No. 10, June 2015, Page 16.
22. www.testdriven.com/tag/acceptance_testing_tools_post_tag
23. Dranidis D., Tigka K. Writing Use Cases in XML
24. Snyder, H. (2019) 'Literature review as a research methodology: An overview and guidelines', Journal of Business Research, Volume 104, Pages 333-339.
25. Fowler, M. (2004) 'UML Distilled: A Brief Guide to the Standard Object Modeling Language', Pearson Education, Inc., Boston.

26. Adenowo, A.A.A., & Adenowo, B.A. (2013) 'Software Engineering Methodologies: A Review of the Waterfall Model and Object-Oriented Approach', International Journal of Scientific & Engineering Research, Volume 4, Issue 7, July-2013, ISSN 2229-5518.
27. Eason, O.K. (2016) 'Information Systems Development Methodologies Transitions: An Analysis of Waterfall to Agile Methodology', Honors Theses and Capstones, University of New Hampshire, Spring 2016.
28. Here's a Harvard reference for the provided journal article:
29. Kramer, M. (2018) 'Best Practices in Systems Development Lifecycle: An Analysis Based on the Waterfall Model', Review of Business & Finance Studies, Vol. 9, No. 1, pp. 77-84. ISSN: 2150-3338 (print), ISSN: 2156.
30. Permana, P.A.G. (2015) 'Scrum Method Implementation in a Software Development Project Management', International Journal of Advanced Computer Science and Applications (IJACSA), Vol. 6, No. 9, p. 198.
31. Mahalakshmi, M., & Sundararajan, M. (2013) 'Traditional SDLC Vs Scrum Methodology – A Comparative Study', International Journal of Emerging Technology and Advanced Engineering, Volume 3, Issue 6, June 2013, p. 192.
32. Kumar, G., & Bhatia, P.K. (2012) 'Impact of Agile Methodology on Software Development Process', International Journal of Computer Technology and Electronics Engineering (IJCTEE), Volume 2, Issue 4, August 2012.
33. Stapleton, J. 'DSDM: Dynamic Systems Development Method - The Method in Practice'.
34. Delima, R., Santoso, H.B., Andriyanto, N., & Wibowo, A. (2018) 'Development of Purchasing Module for Agriculture E-Commerce using Dynamic System Development Model', International Journal of Advanced Computer Science and Applications (IJACSA), Vol. 9, No. 10, p. 86.
35. Nittari, G., Khuman, R., Baldoni, S., Pallotta, G., Battineni, G., Sirignano, A., Amenta, F., & Ricci, G. (2020) 'Telemedicine Practice: Review of the Current Ethical and Legal Challenges', Telemedicine and e-Health, Vol. 26, No. 12. DOI: 10.1089/tmj.2019.0158.
36. Ahmad, S., Abdullah, S.A.J.B., & Arshad, R.B. (2016) 'Issues and Challenges of Transition to e-Voting Technology in Nigeria', Public Policy and Administration Research, Vol. 5, No. 4. ISSN 2224-5731
37. Zou, Y., Kiviniemi, A., & Jones, S.W. (2017) 'A review of risk management through BIM and BIM-related technologies', Safety Science, Volume 97, Pages 88-98. Elsevier.
38. Tupa, J., Simota, J., & Steiner, F. (2017) 'Aspects of risk management implementation for Industry 4.0', 27th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM2017), 27-30 June 2017,

39. Curran, K. (2018) 'E-Voting on the Blockchain', School of Computing, Engineering and Intelligent Systems, Ulster University, UK. Correspondence: kj.curran@ulster.ac.uk. Published: 14 September 2018. Received: 3 September 2018. Accepted: 11 September 2018.