IS 2901 – Software Development Project

Interim Report

Online Voting System

Code Crew

Index No	Name
225110A	A.M.A.D. Weerasinghe
225008T	W.H.P. Anuththara
225087G	R.A.D.P. Ranasinghe
225015L	J.G.J.M.R.K Bandara
225083P	G.D. Punchihewa

Supervised by:

Ms. W.M.R.M. Wijesuriya
Prof. T.C. Sandanayake

Client:

WSO2

20 Palm Grove,

Colombo 00300,

Sri Lanka.

Faculty of Information Technology

University of Moratuwa

2025

Declaration

We declare that this report is our own work and has not been submitted in any form for another degree or diploma at any university or other institution of tertiary education. Information derived from the published or unpublished work of others has been acknowledged in the text and a list of references is given.

Names of Students	Signatures of Students
A.M.A.D. Weerasinghe	
W.H.P. Anuththara	
R.A.D.P. Ranasinghe	
J.G.J.M.R.K Bandara	
G.D. Punchihewa	

07.02.2024
Date:

Supervised by:

Names of Supervisors	Signatures of Supervisors
Ms. W.M.R.M. Wijesuriya	
	Date:
Prof. T.C. Sandanayake	
	Date:

Abstract

Elections are a fundamental part of democracy, ensuring that citizens can express their political choices. However, traditional election systems face challenges such as inefficiencies, high costs, security risks, and limited accessibility. Manual voting processes often result in delays, long queues, and vulnerability to fraudulent activities.

To address these issues, the Online Election System is designed as a hybrid voting model, where voters must visit election centers to cast their votes electronically, enabling real-time vote counting and enhancing election security. This approach maintains the integrity of physical elections while leveraging digital technologies for faster and more accurate results.

The system is developed using Ballerina for backend services, Next.js and React.js for the user interface, and PostgreSQL for secure database management. JWT authentication ensures that only eligible voters can participate, while AES encryption protects sensitive voter data. The application is deployed on cloud platforms like AWS or Azure, ensuring scalability, reliability, and security. Additional technologies like Docker and Kubernetes are used for efficient containerized deployment and system management.

This report outlines the problem statement, system architecture, key functionalities, and security mechanisms implemented in the Online Election System. By integrating modern technologies with traditional voting methods, this solution aims to enhance election efficiency, ensure voter security, and increase transparency, making the electoral process more reliable and accessible.

Table of Contents

List of Figures/Tables

Chapter 1 Introduction

1.1 Introduction

Elections are fundamental to democracy, ensuring that citizens can choose their representatives and express their political preferences. However, traditional election systems often face significant challenges, including inefficiencies in manual vote counting, security risks, accessibility issues, and the potential for fraud. These limitations can lead to delays in results, high operational costs, and reduced voter participation, particularly in remote areas.

To address these concerns, our Online Election System introduces a hybrid voting model that combines digital and physical voting processes. While voters must visit an election center to cast their votes, the system enables secure online voting and real-time result calculation, enhancing both efficiency and transparency. The system incorporates advanced security measures such as encryption, authentication protocols, and audit trails to prevent tampering and unauthorized access.

The system will be used by voters, administrators, candidates, and auditors. Voters will authenticate themselves before casting their votes, while administrators manage voter and candidate registrations and configure elections. Candidates can view election results, and auditors ensure system integrity.

System Overview

Input	Voter registration details, authentication credentials, vote selections.
Process	Secure authentication, encrypted vote submission, vote storage, real-time result processing.
Output	Verified voter list, successful vote casting confirmation, real-time election results.
Technology Stack	Web-based application with secure APIs, PostgreSQL for database management, encryption techniques, and an audit logging system to ensure reliability.
Key Features	Hybrid voting model, real-time result calculation, fraud prevention mechanisms, and data security compliance.

1.2 Problem in Brief

The efficiency and integrity of elections are fundamental to upholding democracy, yet traditional election systems often face significant challenges. Manual processes are time-consuming and prone to errors, leading to delays in result announcements. Also, it leads to long queues and overcrowding at polling centers, which result in extended waiting times, increased voter frustration, and traffic congestion. Additionally, ensuring voter privacy and security remains a persistent issue, with risks of tampering, fraud, and multiple registrations undermining the credibility of the electoral process.

With the growing reliance on technology in all aspects of life, the absence of a secure and transparent digital voting platform exacerbates these challenges. Modern elections require tools that can handle large-scale participation efficiently while maintaining trust and transparency.

The importance of solving these issues lies in safeguarding democracy itself. An inefficient or unreliable election system erodes public trust and could lead to disputes, and reduced participation. Addressing these challenges with a robust, secure, and accessible solution is crucial to ensure the fairness and accuracy of elections while fostering confidence among voters, candidates, and stakeholders. This project aims to fill the gap by introducing a technologically advanced system that addresses these critical concerns and modernizes the electoral process.

1.3 Aim and Objectives

1.3.1. Aim

This project aims to develop a secure and transparent Online Election System that addresses the inefficiencies and vulnerabilities of traditional election processes using modern technologies like encryption, and real-time analytics.

1.3.2. Objectives

- To design a secure voter registration system that ensures privacy, prevents duplicate registrations, and verifies voter eligibility.
- To provide administrative tools for configuring elections, managing candidate profiles, and defining timelines.

- To implement a secure and tamper-proof vote-casting mechanism to maintain voter anonymity and integrity.
- To enable real-time result calculation with advanced analytics based on geographical and other metrics.
- To maintain comprehensive audit trails and security logs for ensuring accountability and detecting unauthorized activities.

1.4 Proposed Solution

The proposed Online Election System is a secure and user-friendly platform to conduct elections, ensuring transparency, efficiency, and voter privacy. It integrates technologies like secure APIs for vote security, encryption for data privacy, and real-time analytics for accurate reporting. The system accommodates various election types seamlessly.

1. Voter Registration
    - Secure API for voter registration using credentials like NIC, name, and email.
    - Encrypted storage to ensure data privacy and prevent unauthorized access.
    - Verification to prevent duplicate or false registrations.
2. Election Configuration
    - Admin tools for election setup, including timelines and candidate profiles.
    - Support for different election types to suit various formats.
3. Vote Casting
    - Secure voting mechanism with JWT authentication to allow one vote per voter.
    - Tamper-proof and anonymous vote storage.
    - Encrypted data handling to ensure privacy.
4. Result Calculation and Reporting
    - Real-time result calculation.
    - Dashboard to display results with different filters.
5. Auditing and Security
    - Comprehensive audit trails to track voting events.
    - Tamper-proof vote storage and security logs for accountability.

1.5 Structure of the Report

Chapter 1 - Introduces the project background, problems, objectives, and proposed solution.

Chapter 2 - Reviews existing election systems and their limitations.

Chapter 3 - Describes the system's technological approach.

Chapter 4 - Provides system analysis, diagrams, and architecture.

Chapter 5 - Explains the implementation and security mechanisms.

1.6 Summary

This chapter introduced the project's motivation, problems, objectives, and proposed solution. The next chapter presents a literature review comparing existing election systems and discussing their drawbacks.

Chapter 2 Literature Review

2.1 Introduction

Elections play a vital role in democratic governance, ensuring citizens have a voice in decision-making. Traditional paper-based voting methods, while widely used, pose several challenges, such as inefficiencies in vote counting, security risks, and accessibility barriers. To address these issues, many countries have experimented with online election systems, leveraging modern technology to enhance election integrity, voter accessibility, and result accuracy.

This chapter presents a literature survey on online voting systems used in different countries, highlighting their security mechanisms, strengths, and limitations. We compare these approaches with our Online Election System, emphasizing how our system addresses key challenges while ensuring security and efficiency.

2.2 Global Approaches to Online Voting

Several countries have explored or implemented online election systems, integrating advanced security technologies to prevent fraud and unauthorized access. Below is a summary of key approaches:

United Kingdom

- Implementation: Limited trials in local elections; not adopted nationwide due to security concerns.
- Security Features: Encryption, Two-Factor Authentication (2FA), Audit Trails, Centralized Servers.
- Limitations: Concerns over cyber threats prevented national-scale implementation.

India

- Implementation: Small-scale trials for municipal elections; not nationwide.
- Security Features: Blockchain-based voting, Aadhaar Biometric Verification, End-to-End Encryption.
- Limitations: Digital divide and biometric authentication challenges.

United States

- Implementation: Limited to overseas and military voters in states like West Virginia and Utah.
- Security Features: Blockchain, Digital Signatures, Encryption, Tokenization, Audit Mechanisms.
- Limitations: Risk of cyberattacks and inconsistent adoption across states.

South Korea

- Implementation: "K-Voting" system used for non-government and local elections.
- Security Features: Digital Certificates, Encryption, Tamper-Proof Records, Audit Mechanisms.
- Limitations: Usage mainly limited to internal and municipal-level elections.

France

- Implementation: Online voting allowed for overseas citizens in legislative elections.
- Security Features: Two-Factor Authentication (2FA), Public Key Infrastructure (PKI), Data Encryption, Regular Penetration Testing.
- Limitations: Vulnerability concerns have led to periodic system evaluations.

Switzerland

- Implementation: Swiss Post and Scytl developed online voting for overseas citizens.
- Security Features: End-to-End Encryption, Blockchain Pilots, Universal Verifiability, Cryptographic Proofs.
- Limitations: Regular penetration tests are required to maintain system integrity.

Australia

- Implementation: State-level systems (e.g., iVote in New South Wales) for remote voters.
- Security Features: Encryption Keys, Authentication Tokens, End-to-End Verifiability, Digital Signatures.
- Limitations: Dependence on digital infrastructure availability.

Canada

- Implementation: Used in some municipal elections; not yet adopted at the federal level.
- Security Features: End-to-End Encryption, Two-Factor Authentication (2FA), Blockchain Trials, Voter Verification.
- Limitations: Security concerns and lack of national implementation.

2.3 Comparison of Global Online Voting Systems

The table below summarizes the security technologies used by different countries in online voting:

Country	**Security Technologies Used**
United Kingdom	Encryption, Two-Factor Authentication, Audit Trails
India	Blockchain, Aadhaar Biometric Verification, End-to-End Encryption
United States	Blockchain, Digital Signatures, Encryption, Tokenization, Audit Mechanisms
South Korea	Digital Certificates, Encryption, Tamper-Proof Records, Audit Mechanisms
France	Two-Factor Authentication, PKI, Data Encryption, Penetration Testing
Switzerland	End-to-End Encryption, Blockchain, Universal Verifiability, Cryptographic Proofs
Australia	Encryption Keys, Authentication Tokens, End-to-End Verifiability, Digital Signatures
Canada	End-to-End Encryption, Two-Factor Authentication, Blockchain, Voter Verification

2.4 Summary

This chapter provided a literature review of global online election systems, their security technologies, and the challenges faced in their implementation. While several countries have tested or adopted online voting in different capacities, concerns regarding security risks, voter authentication, and infrastructure reliability remain common obstacles.

Our Online Election System integrates proven technologies like secure authentication (JWT), encrypted vote storage (AES), and cloud scalability to ensure a reliable and transparent voting process. The next chapter will discuss the system approach, including the design architecture and technological choices that make our system a viable and secure election solution.

Chapter 3 System Aproach

3.1 Introduction

The Online Election System is designed to address inefficiencies and security risks in traditional voting methods by integrating modern technologies that ensure secure, scalable, and efficient election management. This chapter explains the approach taken to develop the system, detailing the technologies used and how they are applied to solve key election challenges. The system incorporates a cloud-based architecture, secure authentication, encrypted data storage, and real-time result calculation to ensure fairness and transparency.

3.2 System Overview

The proposed system consists of four key components:

- User Interface (Frontend) - Developed with React.js and Next.js for a responsive and user-friendly experience.
- Backend API - Built using Ballerina, ensuring secure and scalable request handling.
- Database Management - Uses PostgreSQL to store voter and election data securely.
- Security & Infrastructure - Implements JWT authentication, AES encryption, Docker, Kubernetes, and cloud hosting (AWS/Azure) for security and scalability.

3.2 System Components and Technologies

3.3.1 Frontend – React.js and Next.js

The frontend is built using React.js and Next.js, offering:

- A dynamic and responsive UI for voters, administrators, and auditors.
- Server-Side Rendering (SSR) in Next.js to improve performance and load times.
- Real-time updates for displaying election results.

3.3.2 Backend – Ballerina

The backend is developed using Ballerina, chosen for:

- API-First Development – Provides RESTful API endpoints for frontend integration.
- Scalability & Flexibility – Manages high-traffic loads efficiently.
- Service Integration – Connects with authentication, database, and logging services securely.

3.3.3 Database – PostgreSQL

PostgreSQL is used for data storage due to:

- ACID Compliance – Ensures reliability and consistency in vote recording.
- Scalability – Handles large datasets for nationwide elections.
- Security – Supports encryption and access control mechanisms.

3.3.4 Authentication – JWT (JSON Web Tokens)

JWT ensures that only eligible voters can access the system by:

- Providing secure login sessions without needing server-side storage.
- Preventing unauthorized access through token validation.

3.3.5 Security – AES Encryption

Sensitive voter data is encrypted using AES (Advanced Encryption Standard):

- Data at Rest Protection – Encrypts database-stored voter and election data.
- Data in Transit Security – Ensures safe communication between system components.

3.3.6 Deployment – Docker & Kubernetes

- Docker: Packages the application into containers for easy deployment.
- Kubernetes: Manages containerized applications for load balancing and fault tolerance.

3.3.7 Cloud Hosting – AWS/Azure

The system is hosted on AWS or Azure, ensuring:

- High Availability – Reliable uptime for election day operations.
- Scalability – Expands resources as demand increases.
- Disaster Recovery – Data is backed up securely.

3.4 System Workflow

- Voter Registration – Users sign up and verify eligibility.
- Vote Casting – Voters log in, authenticate, and securely cast their vote at the election center.
- Vote Storage – Votes are encrypted and stored in PostgreSQL.
- Result Calculation – The system processes and displays real-time results.
- Audit and Monitoring – Admins and auditors review logs for transparency.

3.5 Summary

This chapter detailed the technologies and architecture of the Online Election System, demonstrating how the system ensures security, scalability, and efficiency in managing elections. The next chapter will focus on System Analysis and Design, including use case diagrams and system workflows.

Chapter 4 Analysis and Design

4.1 Introduction

This chapter details the analysis and design of the Online Election System. It includes the Software Requirement Specification (SRS) that outlines both functional and non-functional requirements. Additionally, it provides various design models such as Use Case Diagram, Class Diagram, Activity Diagrams, Sequence Diagrams, and Entity-Relationship Diagram to illustrate the system's workflow, structure, and data relationships. These models ensure a clear understanding of how different components of the system interact and contribute to a secure and efficient election process.

Table 4.1.1 *Software Requirement Specification Details*

Functional and Non-Functional Requirements

Functional Requirements	Non – Functional Requirements
Voter Registration • Users must provide valid identification details. • The system must verify user identity via OTP/email. • The system should reject duplicate registrations.	Performance Requirements • The system must handle up to 100,000 concurrent users without performance degradation. • Voter authentication and vote submission should be processed within 2 seconds. • Database queries should be optimized to retrieve results within 1 second for election results and voter verification. • The system should support load balancing and auto-scaling to accommodate peak election periods.
Election Configuration • The system must allow admins to configure elections. • Elections must prevent duplicate candidate entries.	Safety Requirements • The system must ensure data redundancy and regular backups to prevent data loss. • A failover mechanism should be in place to guarantee continuous operation during unexpected downtimes. • System integrity checks should be performed periodically to identify data corruption or inconsistencies.

	• User session timeouts should be implemented to prevent unauthorized access to voter data.
Vote Casting • Voters must be authenticated before voting. • System must prevent duplicate votes.	Security Requirements • All data transmissions must be encrypted using TLS 1.2 or higher. • Voter credentials must be securely stored using bcrypt hashing. • Role-Based Access Control (RBAC) must be enforced to restrict user privileges. • Multi-Factor Authentication (MFA) should be implemented for administrative accounts. • Anomaly detection and intrusion prevention systems should be integrated to monitor for suspicious activities.
Result Calculation and Reporting • System must accurately calculate election results. • Results must be securely stored.	Software Quality Attributes • Reliability: The system should maintain at least 99.99% uptime. • Usability: The UI should be intuitive and accessible, complying with WCAG 2.1 guidelines. • Maintainability: The system should follow modular coding practices to simplify updates and patches. • Scalability: Cloud-based infrastructure should allow horizontal scaling to accommodate varying loads. • Interoperability: The system should support integration with external identity verification services.
Auditing and Security • The system must log all critical actions. • Multi-layer security must prevent unauthorized access.	Business Rules • Only verified users are allowed to register as voters. • Election administrators must finalize configurations before the election begins. • A voter may only cast one vote per election.

	• Votes cannot be altered or removed once submitted. • Election results should be verified by auditors before being published.
	Other Requirements • The system must comply with local and international election laws. • Multi-language support should be provided to accommodate diverse voter demographics. • The application should be deployable in both on-premise and cloud environments. • A comprehensive logging system should be in place for auditing and compliance tracking. • Regular security audits and penetration tests should be conducted to ensure the system remains secure.

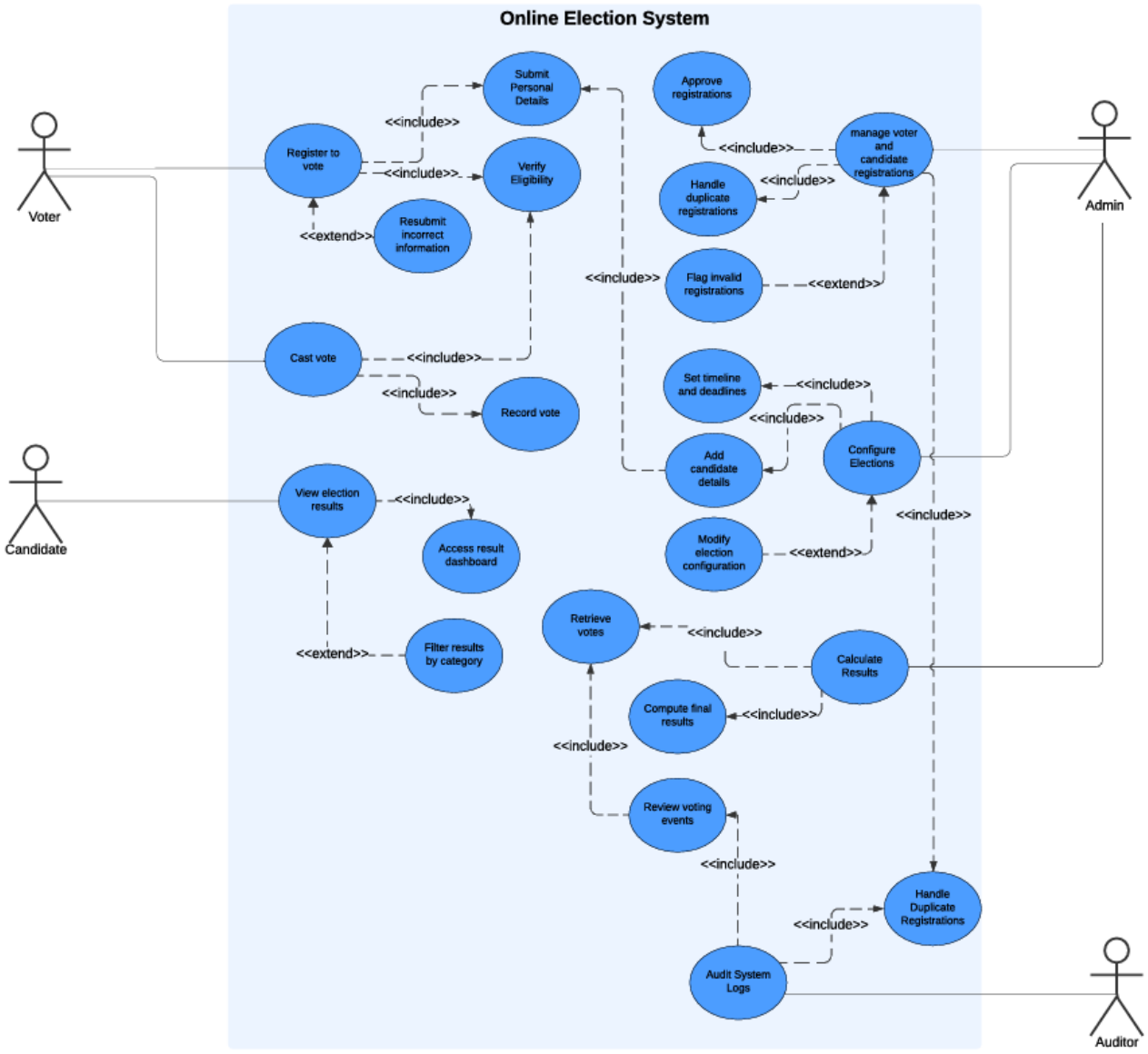

Figure 4.1.1 *Use Case Diagram*

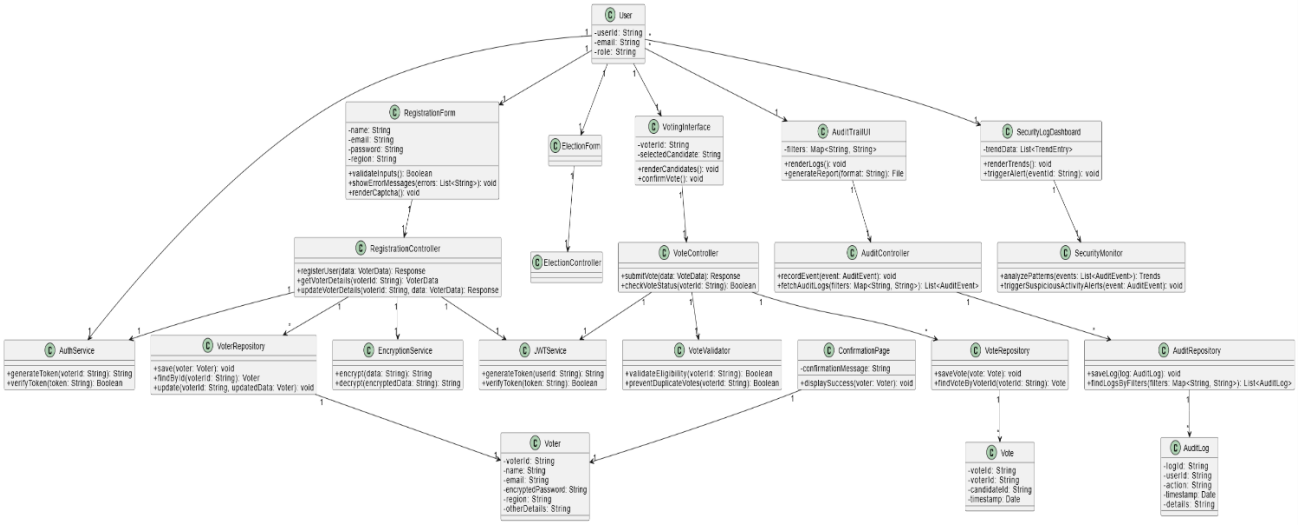

Figure 4.1.2 *Class Diagram*

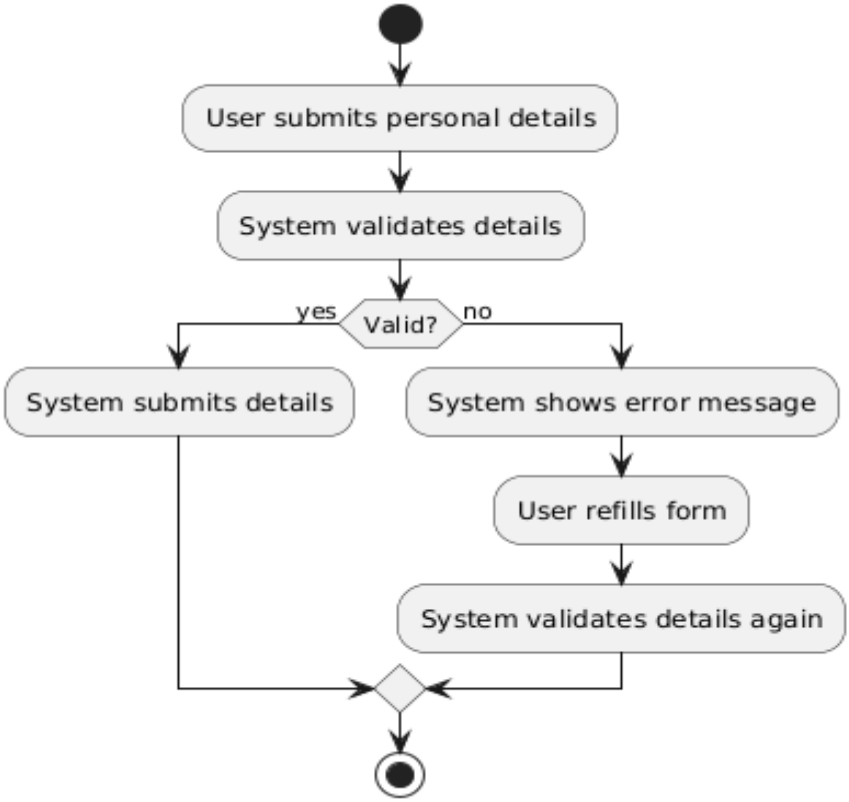

Figure 4.1.3 *Activity Diagram*

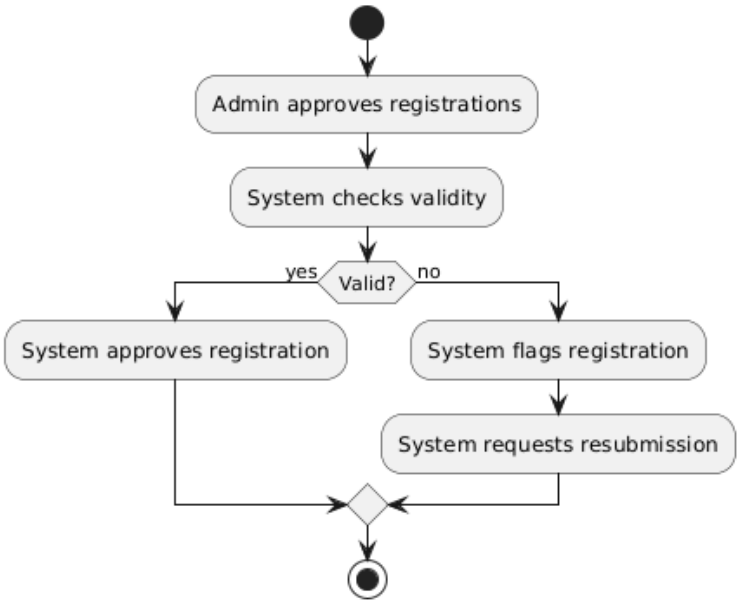

Figure 4.1.3 *Activity Diagram*

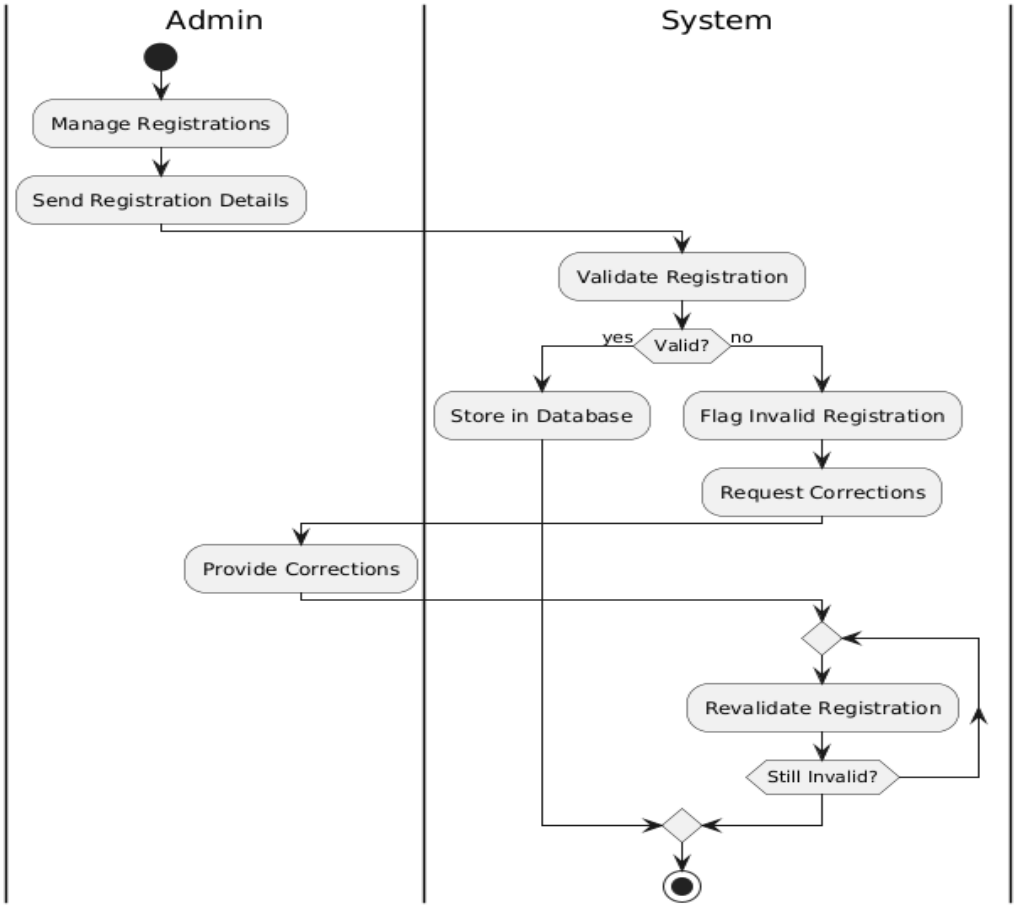

Figure 4.1.3 *Activity Diagram*

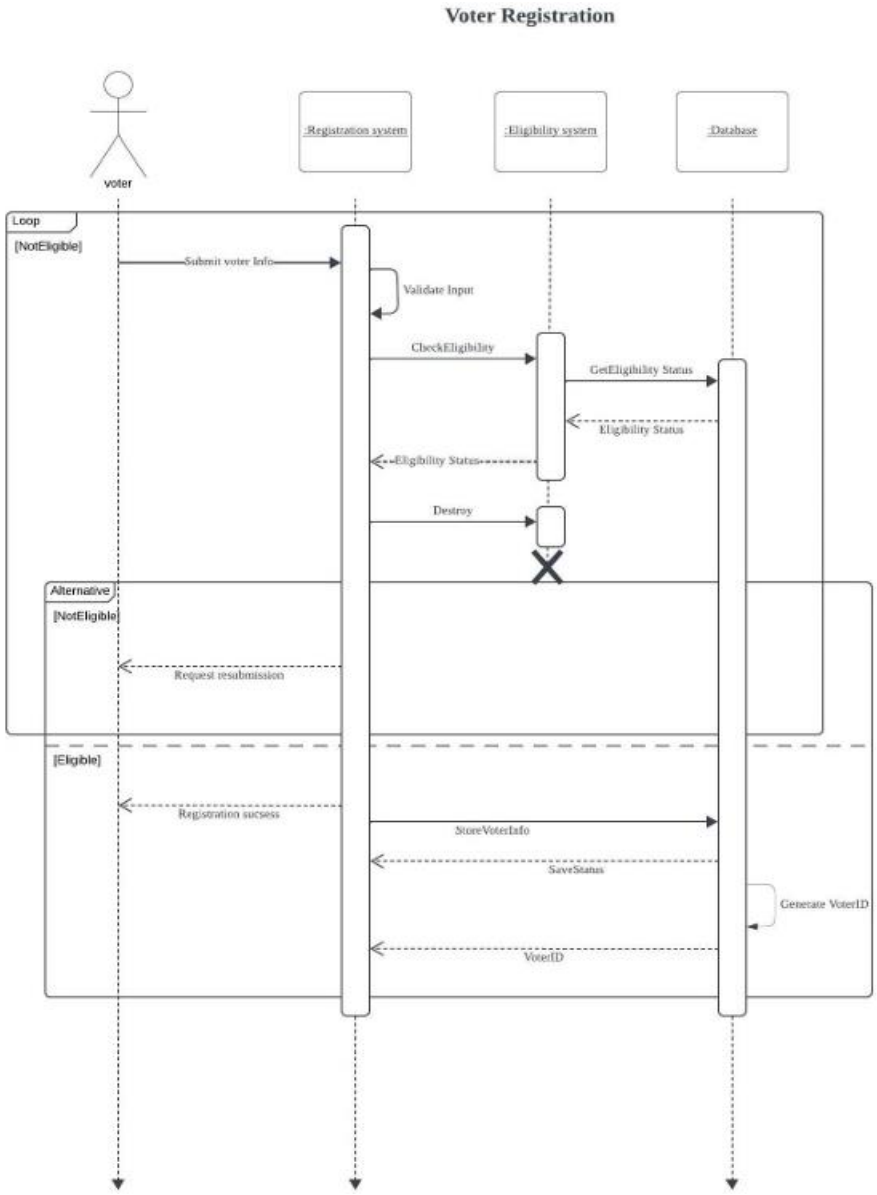

Figure 4.1.4 *Sequence Diagram*

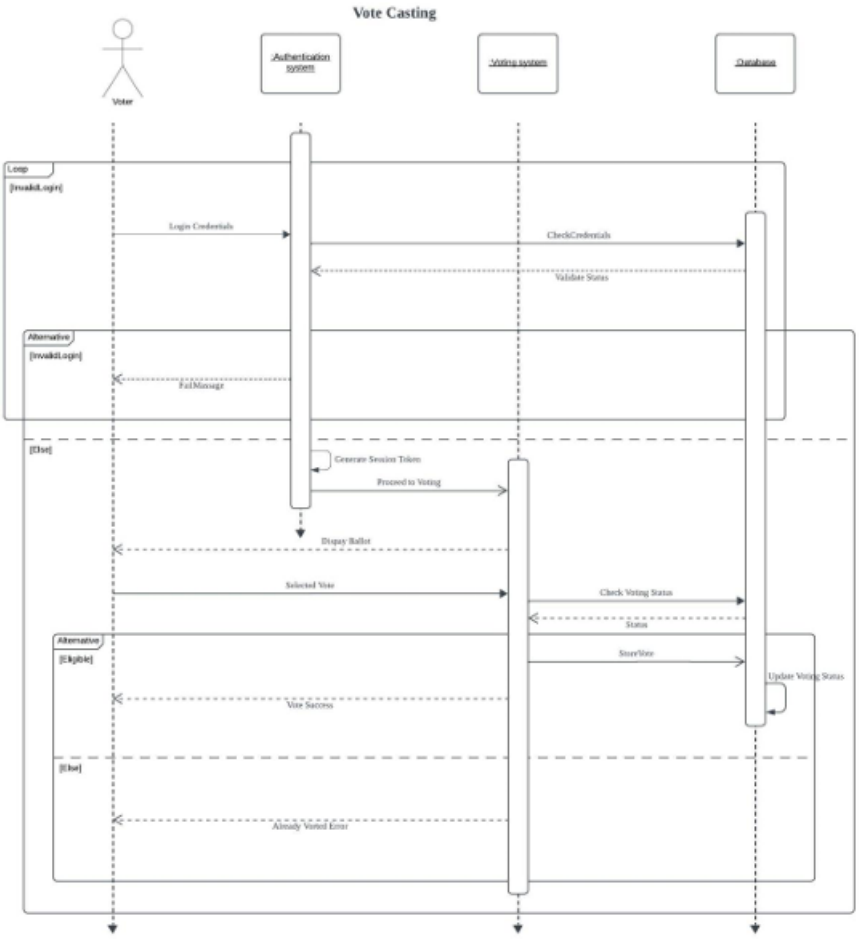

Figure 4.1.4 *Sequence Diagram*

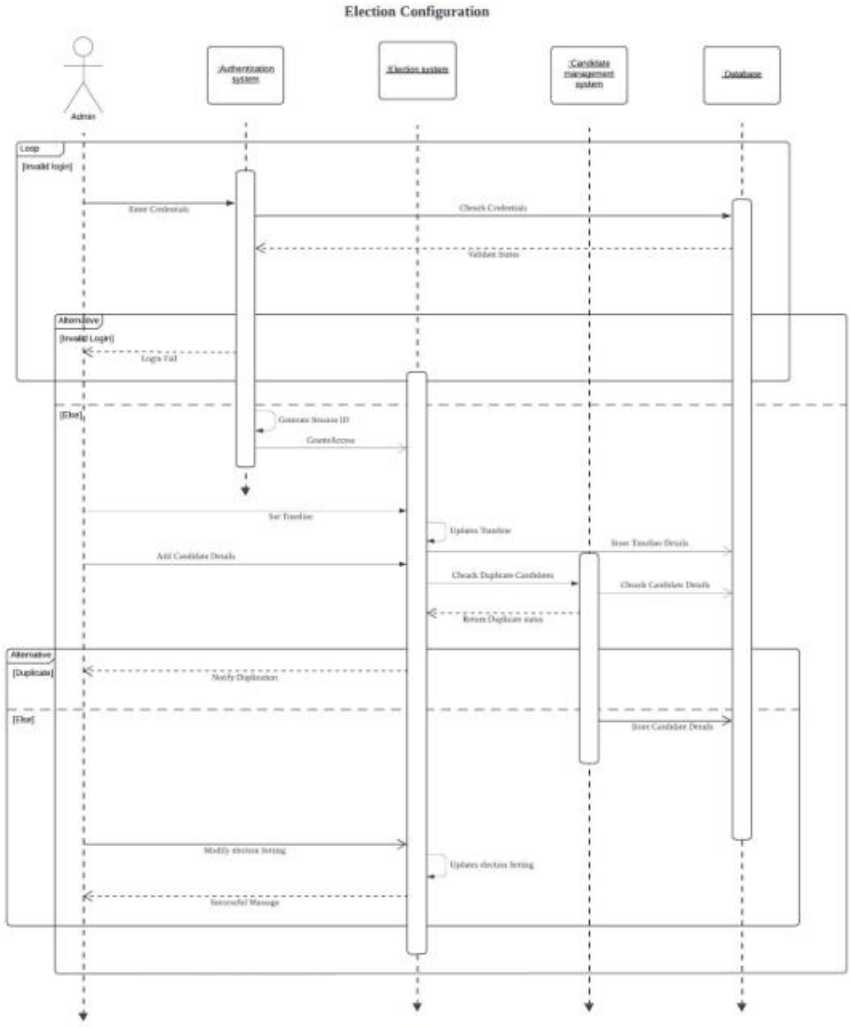

Figure 4.1.4 *Sequence Diagram*

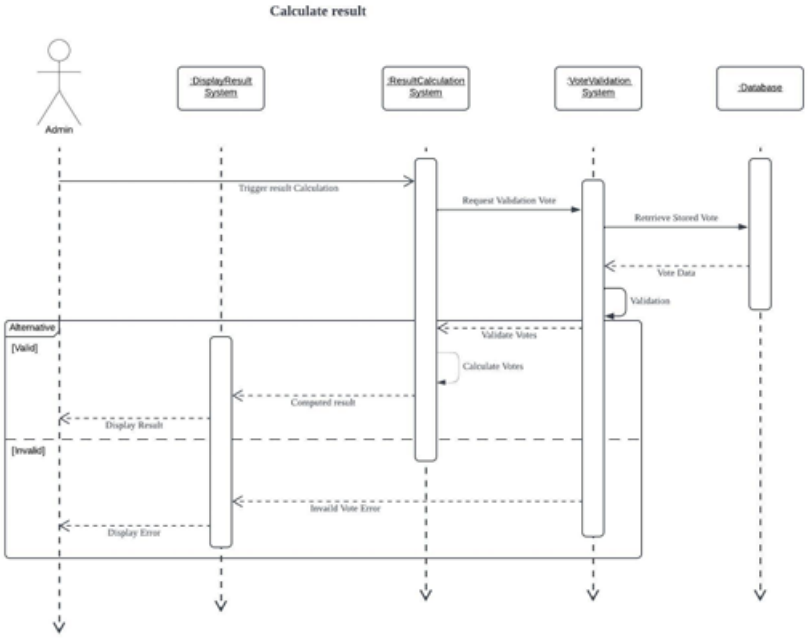

Figure 4.1.4 *Sequence Diagram*

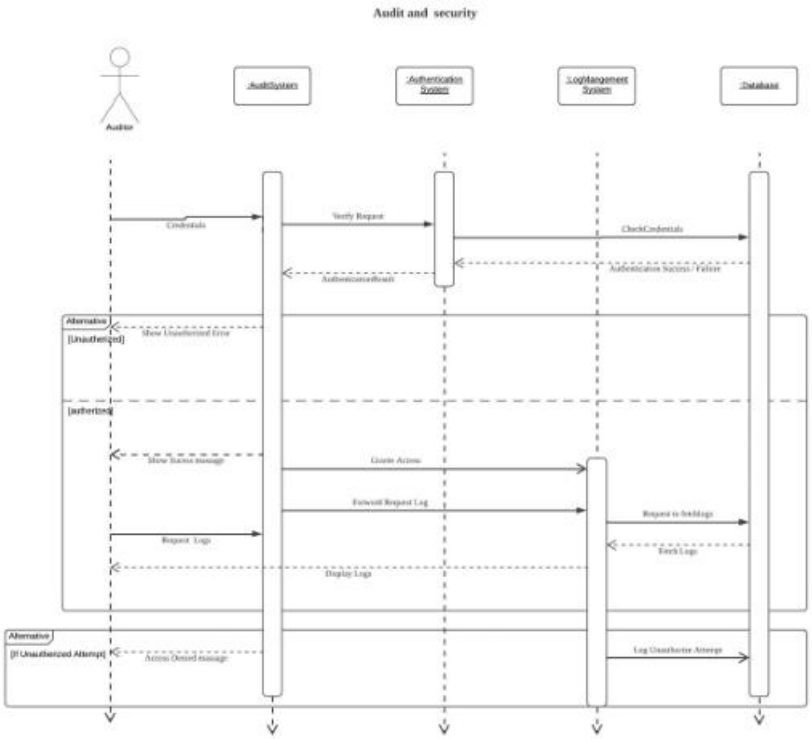

Figure 4.1.4 *Sequence Diagram*

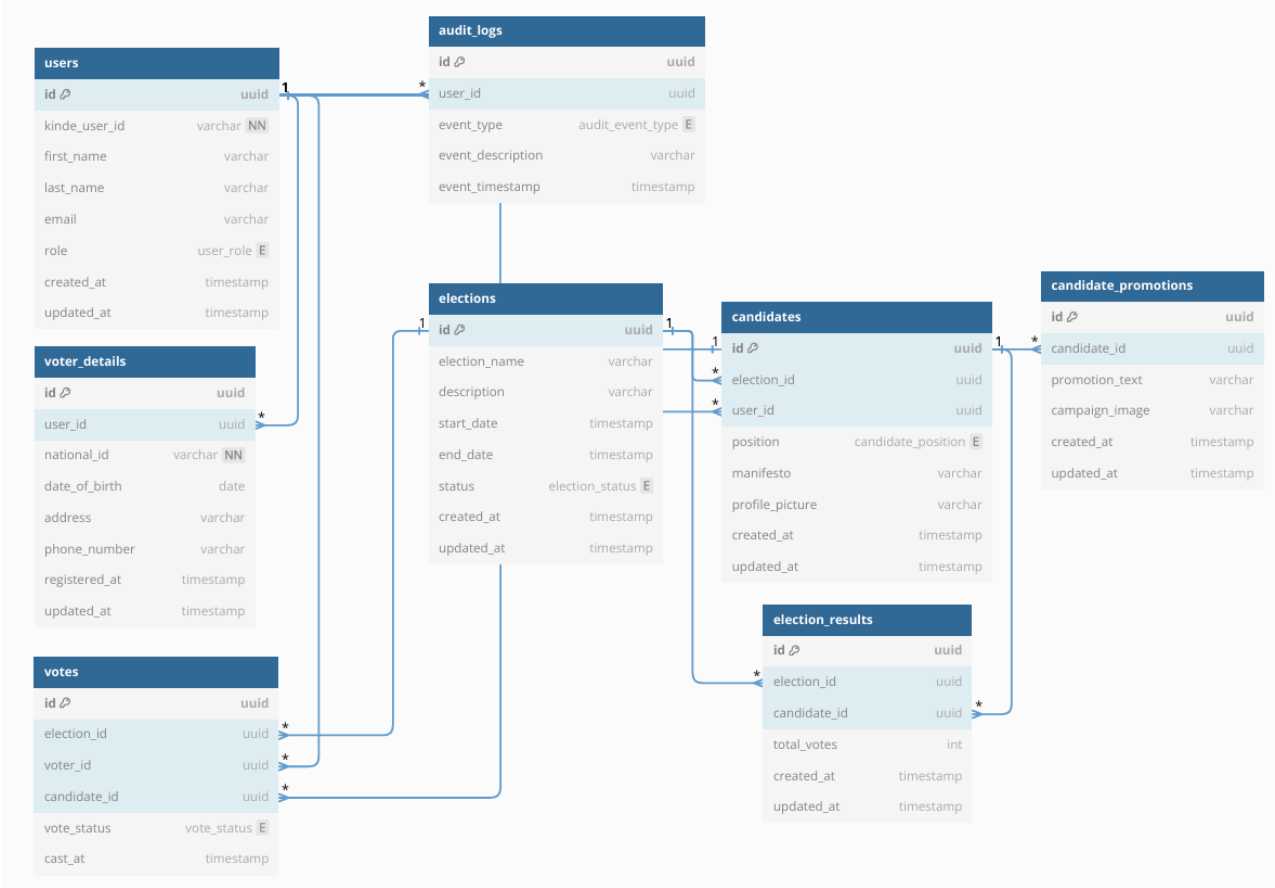

Figure 4.1.5 *ER Diagram*

Chapter 5 Implementation

5.1 Introduction

This chapter details the implementation progress of the Online Election System up to the time of report submission. It covers software components, hardware requirements, system architecture, and key code segments that contribute to the system's functionality. The chapter also provides the Figma link for the UI/UX design and references to the UML diagrams from the previous chapter to illustrate the system's design and workflow.

The system is implemented as a hybrid election system, where voters authenticate and cast their votes digitally at election centers, ensuring real-time result calculation while maintaining traditional election integrity. The implementation focuses on security, scalability, and efficiency by utilizing modern web technologies and best practices in software development.

5.2 Software and Hardware Requirements

5.2.1 Software Requirements

- Frontend: Next.js, React.js
- Backend: Ballerina
- Database: PostgreSQL
- Authentication: JWT (JSON Web Tokens)
- Encryption: AES (Advanced Encryption Standard)
- Containerization & Deployment: Docker, Kubernetes, AWS/Azure Cloud Hosting

5.2.2 Hardware Requirements

- Server Requirements: Cloud-based infrastructure (AWS/Azure)
- Client Requirements: Internet-enabled devices (PCs, tablets, or kiosks at election centers)
- Storage: Scalable cloud-based database storage

5.3 Implementation Details by Module

The system implementation is divided into key functional modules, each handling a specific aspect of the election process. The following sections describe the implementation details, including flowcharts, algorithms, pseudo code, and significant code snippets for major modules.

5.3.1 Voter Registration Module

- Flow: Voters register using NIC, email, and credentials. Data is encrypted and stored securely.
- Algorithm: Verifies voter eligibility and prevents duplicate registrations.
- Code Snippet: JWT-based authentication for secure user registration.

5.3.2 Vote Casting Module

- Flow: Voter authentication → Vote submission → Secure storage → Confirmation.
- Algorithm: Ensures each voter can cast only one vote per election.
- Code Snippet: Encrypted vote storage using AES.

5.3.3 Real-Time Result Calculation Module

- Flow: Votes are counted and aggregated dynamically.
- Algorithm: Retrieves, decrypts, and tallies votes securely.
- Code Snippet: SQL query for vote aggregation using PostgreSQL.

5.3.4 Admin & Auditor Module

- Flow: Admins manage election setup, voter approvals, and result audits.
- Algorithm: Logs system activities and flags suspicious events.
- Code Snippet: Admin dashboard implementation in Next.js.

5.4 UI/UX Design

The system's user interface is designed for an intuitive and seamless voting experience. The Figma link to the UI designs is provided below:

https://www.figma.com/design/ONxAM8spz4Q33kpCNbUXDt/Online-Election-System?node-id=0-1&t=HCma5KPE3FE0nvMr-1

5.5 Summary

This chapter presented the implementation progress of the Online Election System, detailing the software and hardware requirements, key system modules, and algorithms used in the system. The next steps include further testing, deployment, and security enhancements to ensure a robust and scalable election system. The following chapter will discuss system testing and evaluation to validate performance and security.

• References

1. Herrnson, P. S., Bederson, B. B., Lee, B., Francia, P. L., & Niemi, R. G. (2005). Voters' Evaluations of Electronic Voting Systems. American Politics Research, 33(4), 705-729. https://gvpt.umd.edu/sites/gvpt.umd.edu/files/pubs/Herrnson%20et%20al%20APR%20Evals%20of%20Electronic%20Voting.pdf?utm_source=chatgpt.com

2. Oostveen, A. M., & Van den Besselaar, P. (2004). E-voting and Media Effects: An Exploratory Study. Electronic Voting in Europe-Technology, Law, Politics and Society, 47-55. https://www.lse.ac.uk/media%40lse/research/EMTEL/Conference/papers/Oostveen.pdf?utm_source=chatgpt.com

3. Ballerina Programming Language. (n.d.). Documentation. Retrieved from https://ballerina.io/learn/by-example/documentation/?utm_source=chatgpt.com

4. Abu-Shanab, E., Knight, M., & Refai, H. (2010). E-voting Systems: A Tool for E-democracy. Management Research and Practice, 2(3), 264-274. https://www.researchgate.net/publication/227490182_E-voting_systems_A_tool_for_e-democracy?utm_source=chatgpt.com

5. Spanos, A., & Kantzavelou, I. (2023). A Blockchain-based Electronic Voting System: EtherVote. arXiv preprintarXiv:2307.10726. https://arxiv.org/abs/2307.10726?utm_source=chatgpt.com

• Appendix A Individual Contribution to the Project

This section outlines the individual contributions of each team member to the development of the Online Election System. Each member played a significant role in different aspects of the project, including research, design, documentation, and implementation. Their combined efforts ensured a well-structured and functional system. The contributions are as follows:

Name	**Index Number**	**Contribution**
A.M.A.D. Weerasinghe	225110A	Class diagram drawing Figma UI designing Tech stack research
W.H.P. Anuththara	225008T	Preparing the literature review Use Case diagram drawing Preparing the SRS
R.A.D.P. Ranasinghe	225087G	Problem defining Activity diagram drawing Figma UI designing
J.G.J.M.R.K Bandara	2255015L	Preparing the literature review Sequence diagram drawing Preparing the SRS
G.D. Punchihewa	225083P	ER diagram drawing Figma UI designing Tech stack research