Software Requirements Specification

BLOCKCHAIN VOTING SYSTEM

Version 1.0

Poornima (22103010)

Pranav Sharma (22103018)

Arnav Singh (22103020)

Gurmehar Singh (22103078)

Punjab Engineering College

23/09/2024

Table of Contents

Software Requirements for Blockchain-Based Voting System

1. Introduction

1.1 Purpose

The purpose of this document is to define the software requirements for a 13 Blockchain-Based Voting System. This system aims to address the challenges faced by traditional voting systems by leveraging blockchain technology to create a secure, transparent, and tamper-proof voting process.

1.2 Scope

This system will provide a comprehensive solution for conducting elections using blockchain technology. It will cover voter registration, secure voting, vote tallying, and result computation. The system is designed specifically to be applicable for corporate and organizational voting.

1.3 Definitions, Acronyms, and Abbreviations

- Blockchain: A decentralized, distributed ledger technology that records transactions across many computers.
- SHA-256: Secure Hash Algorithm 256-bit, a cryptographic hash function.
- Smart Contract: Self-executing contracts with the terms of the agreement directly written into code.

1.4 References

- IEEE Standard 830-1998 for Software Requirements Specifications
- Ethereum Blockchain Documentation
- Solidity Programming Language Documentation

1.5 Overview

The remainder of this document provides a general description of the product, including its functions, user characteristics, constraints, and dependencies. It then details the specific requirements for the system, including functional, performance, and design requirements.

2. Overall Description

2.1 Product Perspective

The Blockchain-Based Voting System is a standalone system that can integrate with existing electoral processes. It consists of several interconnected modules that work together to provide a secure and transparent voting experience.

2.2 Product Functions

The main functions of the system include: - Voter registration and authentication - Secure vote casting - Blockchain management for vote storage - Real-time vote tallying and result computation - Blockchain validation to ensure data integrity.

2.3 User Characteristics

The system will be used by: - Voters: Individuals participating in the election - Election officials: Administrators overseeing the election process - Observers: Entities monitoring the election for transparency.

Users should have basic digital literacy. The system will provide a user-friendly interface to accommodate users with varying levels of technical expertise.

2.4 Constraints

- The system must comply with relevant election laws and regulations.
- It must ensure voter privacy and prevent the possibility of linking votes to individual voters.
- The system should be scalable to handle elections of various sizes.

2.5 Assumptions and Dependencies

- Users have access to devices capable of connecting to the internet.
- A reliable internet connection is available for all users.
- The system depends on the security and reliability of the underlying blockchain technology.

3. Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

- Web-based interface for voter registration and voting
- Administrative interface for election officials

3.1.2 Hardware Interfaces

- The system should be compatible with standard computing devices (PCs, smartphones, tablets)

3.1.3 Software Interfaces

- Integration with existing voter databases for authentication
- Interaction with the Ethereum blockchain network

3.2 Functional Requirements

3.2.1 Voter Registration Module

- REQ-1: The system shall provide a secure registration process for eligible voters.
- REQ-2: Each registered voter shall be assigned a unique identification number.

3.2.2 Voting Module

- REQ-3: The system shall allow registered voters to cast their votes securely.
- REQ-4: Each vote shall be recorded as a transaction in a block.

3.2.3 Blockchain Module

- REQ-5: The system shall maintain a blockchain to store all votes.
- REQ-6: Each block in the blockchain shall contain vote data, a hash pointer to the previous block, and its own hash.

3.2.4 Validation Component

- REQ-7: The system shall regularly validate the integrity of the blockchain.
- REQ-8: Any tampering attempts shall be immediately detected and flagged.

3.2.5 Result Calculation Module

- REQ-9: The system shall provide real-time vote tallying.
- REQ-10: Final election results shall be computed based on the blockchain data.

3.3 Performance Requirements

- The system shall support a minimum of 100 concurrent users.
- Vote recording shall occur within 5 seconds of submission.
- The system shall be available 99% of the time during active voting periods.

3.4 Design Constraints

- The system shall be implemented using Python, JavaScript, and Solidity.

- Smart contracts shall be deployed on the Ethereum blockchain.
- The system shall use SHA-256 or an equivalent cryptographic hash function.

3.5 Software System Attributes

3.5.1 Reliability

- The system shall ensure the accurate recording and counting of all valid votes.

3.5.2 Security

- The system shall implement encryption to protect voter data and votes.
- It shall prevent double voting and unauthorized access.

3.5.3 Maintainability

- The system shall be modular to allow for easy updates and improvements.

3.5.4 Portability

- The system shall be accessible via web browsers and mobile devices across different platforms.

3.6 Other Requirements

- The system shall comply with data protection regulations and election laws.
- It shall provide an audit trail for post-election verification.

4. Appendices

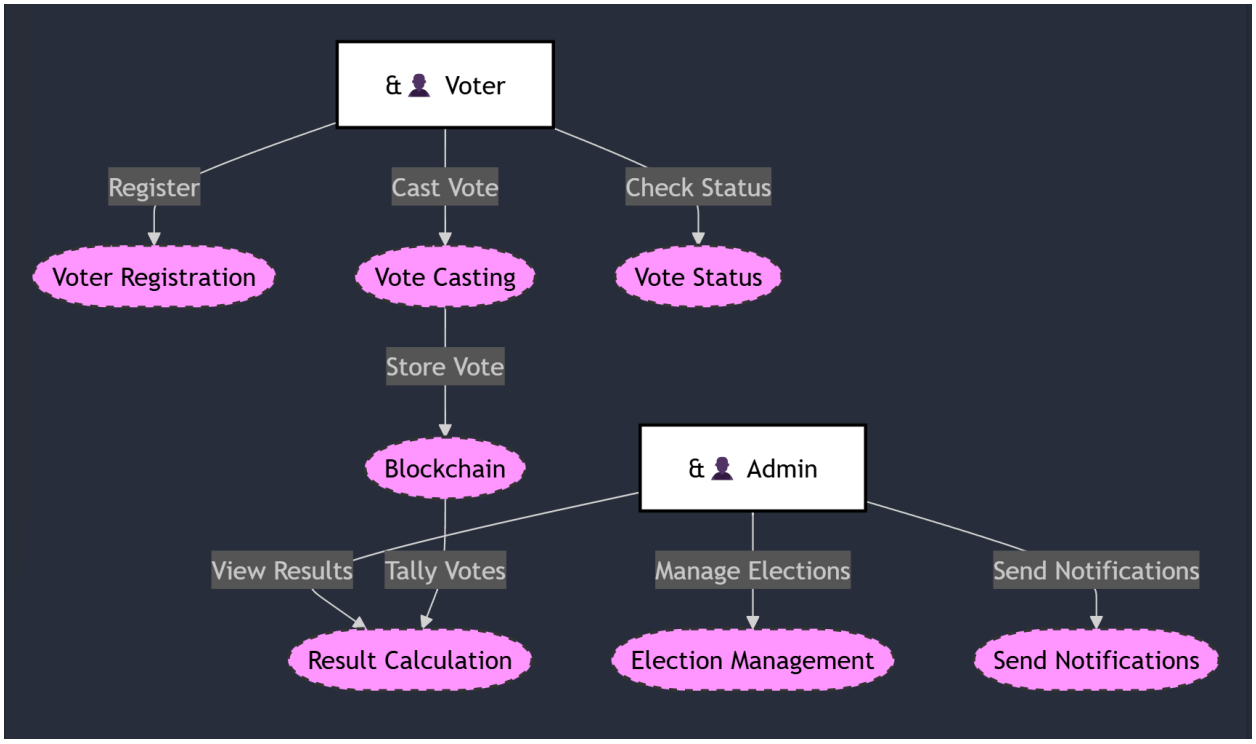
4.1 Glossary

- *Blockchain*: A decentralized, distributed ledger technology that records transactions across many computers in a way that ensures security, transparency, and immutability.
- *Smart Contract*: Self-executing contracts with the terms of the agreement directly written into code, which automatically enforce and execute the terms when predetermined conditions are met.
- *Cryptographic Hash Function*: A mathematical algorithm that maps data of arbitrary size to a bit string of a fixed size, designed to be one-way and collision-resistant.
- *SHA-256*: Secure Hash Algorithm 256-bit, a specific cryptographic hash function commonly used in blockchain technology.
- *Node*: A computer or device connected to the blockchain network that participates in maintaining and validating the blockchain.
- *Block*: A unit of data in the blockchain that contains a set of transactions (in this case, votes) and other relevant information, including a reference to the previous block.
- *Immutability*: The quality of being unchangeable, a key feature of blockchain technology that prevents tampering with recorded data.
- *Decentralization*: The distribution of control and decision-making across a network, rather than being concentrated in a single authority.

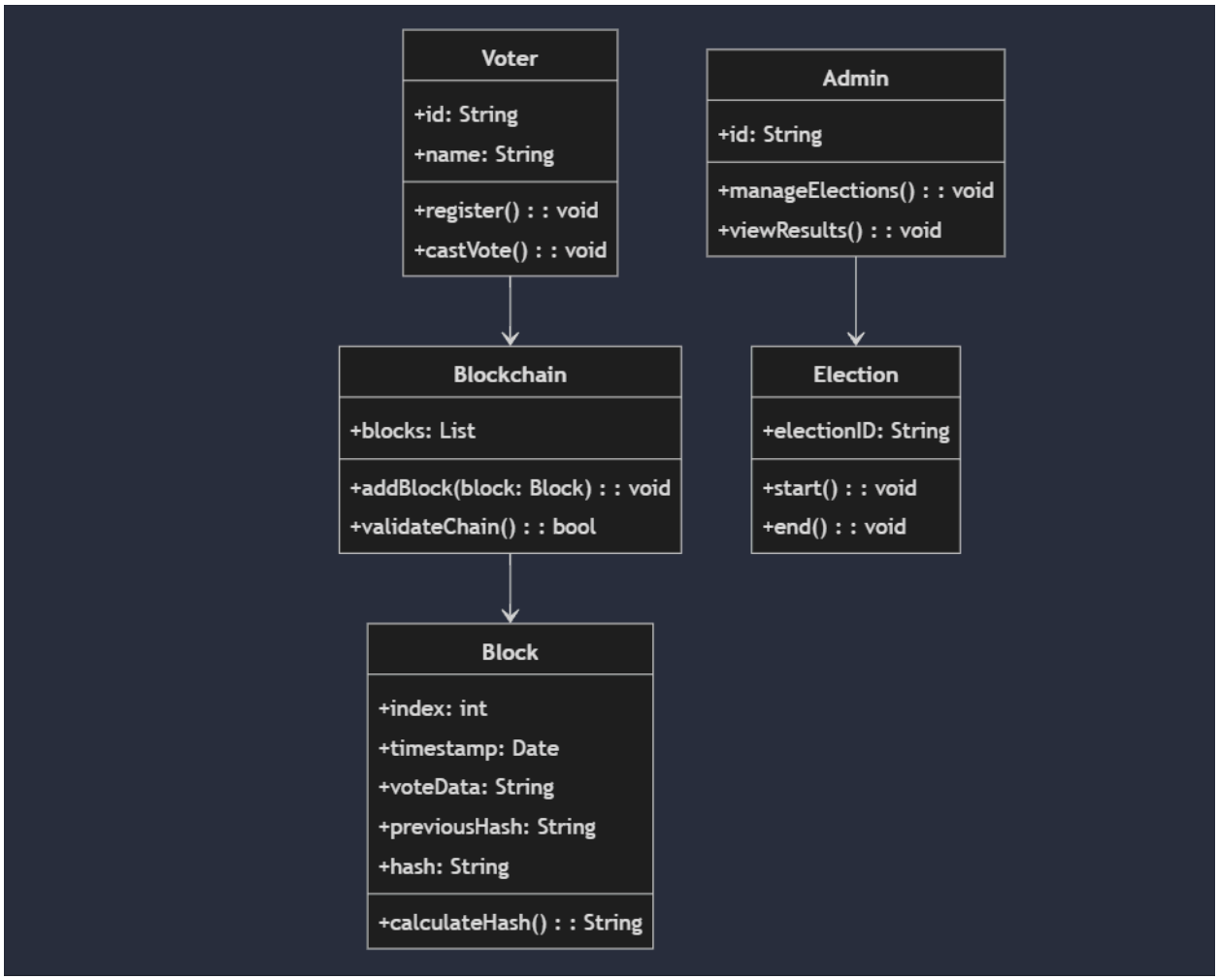
- *Voter Authentication*: The process of verifying the identity and eligibility of a voter before allowing them to cast a vote.
- *Tallying*: The process of counting votes and determining the final results of an election.
- *Transparency*: The quality of being open and visible to all participants, allowing for verification and trust in the voting process.

4.2 Analysis Models

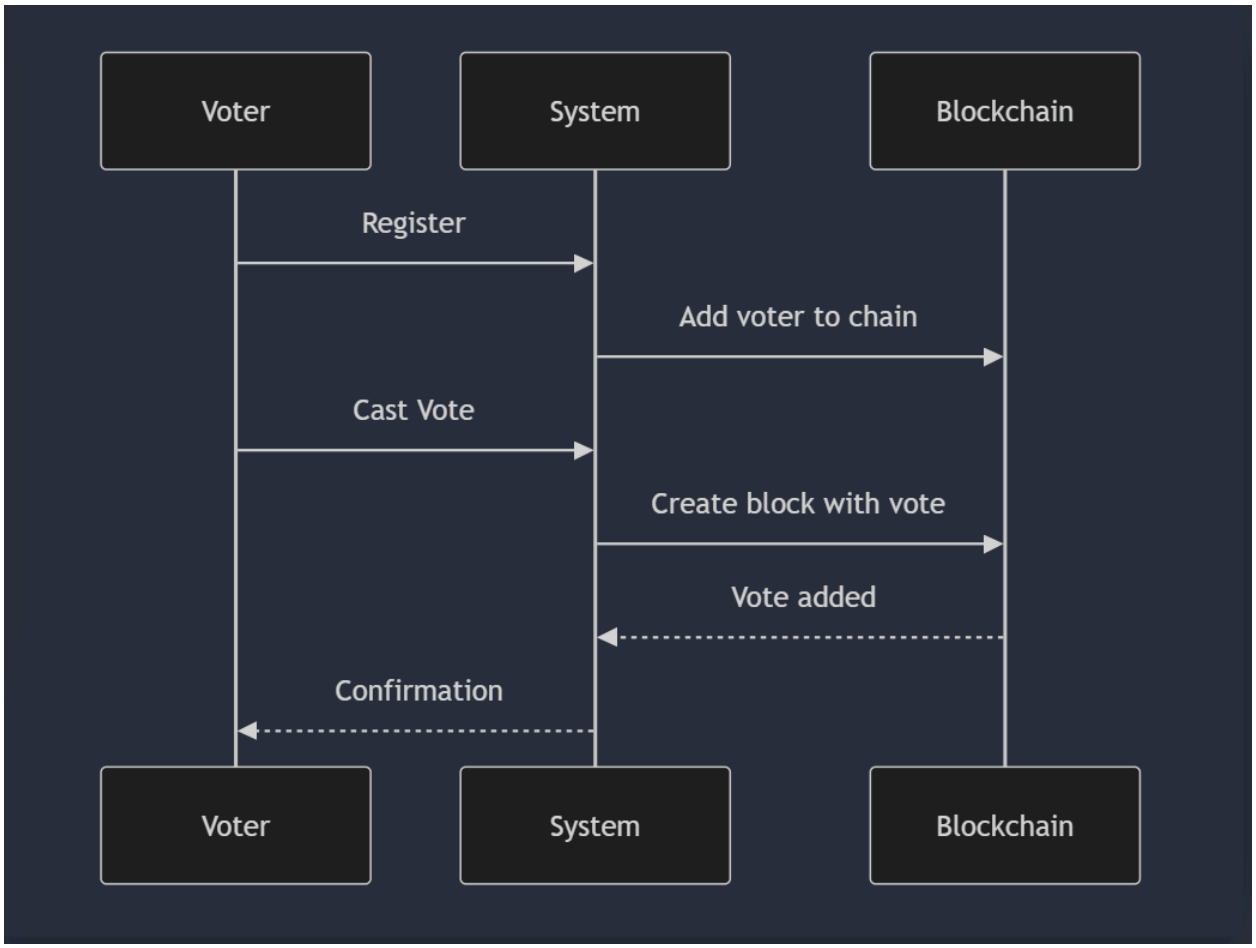
4.2.1 Use Case Diagram



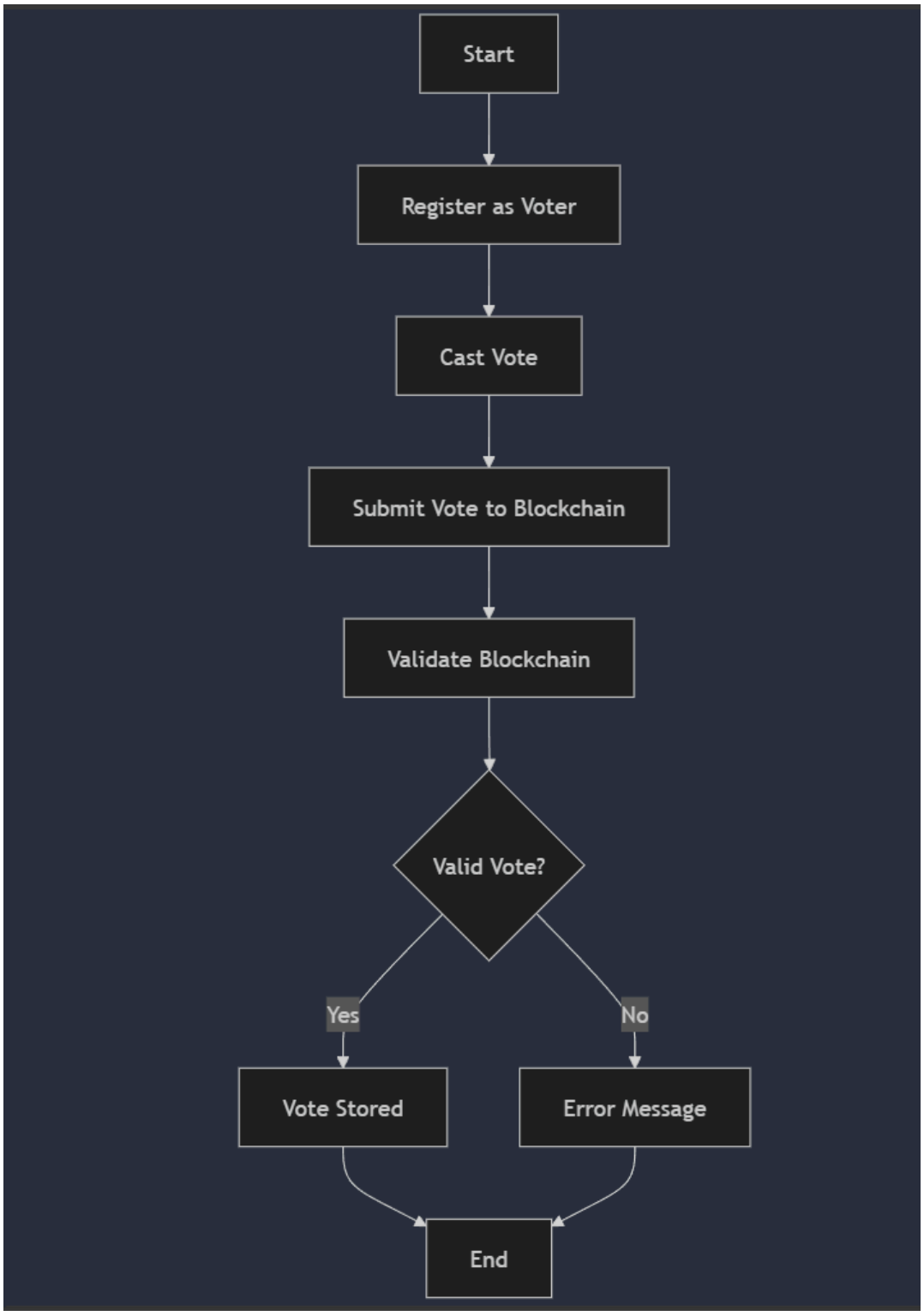
4.2.2 Class Diagram



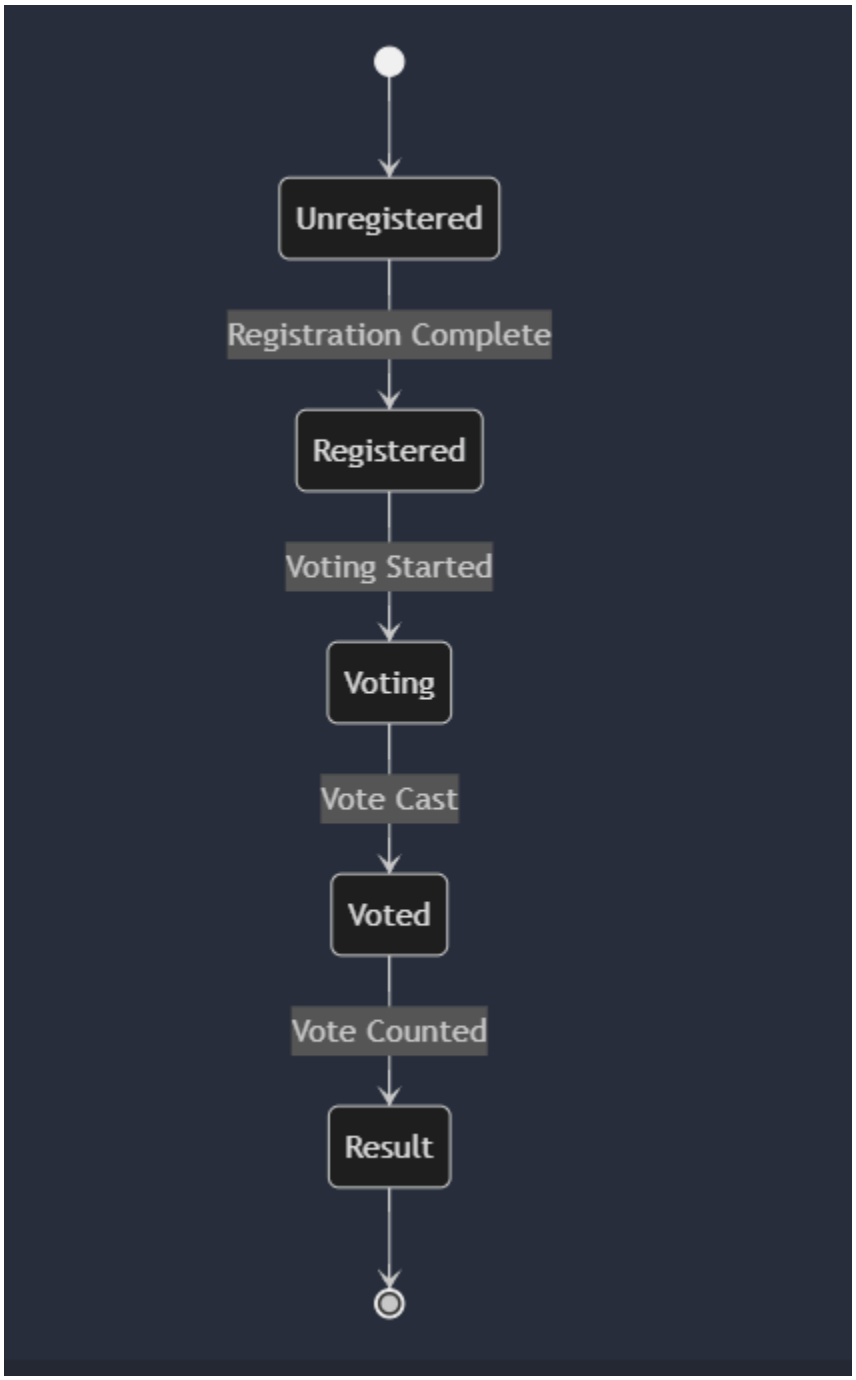
4.2.3 Sequence Diagram



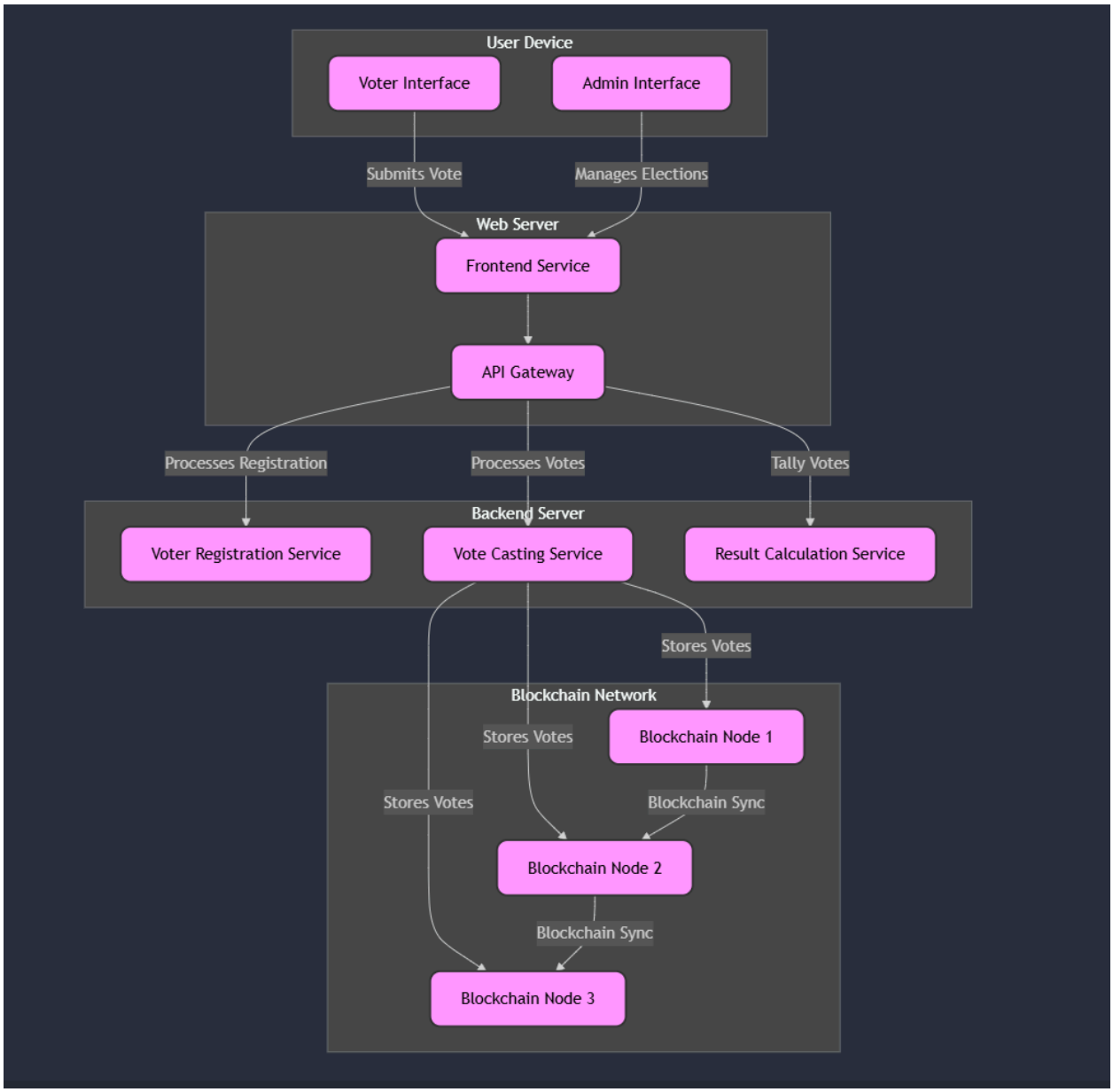
4.2.4 Activity Diagram



4.2.4 Status Diagram



4.2.5 Deployment Diagram



4.3 Issues List

- *Vote Counting Method Selection:* The choice of vote counting method may affect election fairness and accuracy; consider alternative systems and conduct thorough research.
- *Blockchain Scalability:* Increased voter numbers may slow transaction processing; implement efficient consensus algorithms and layer-2 scaling solutions.
- *Voter Privacy:* Protecting voter anonymity is crucial to prevent intimidation; use zero-knowledge proofs or cryptographic techniques.
- *Smart Contract Vulnerabilities:* Bugs in smart contracts could lead to vote manipulation; perform code audits and use formal verification.
- *Voter Authentication:* Weak identity verification can lead to fraud; implement multi-factor authentication and decentralized identity solutions.
- *Accessibility:* Design the system for all eligible voters, including those with disabilities, to avoid reduced turnout; follow accessibility guidelines.
- *Network Attacks:* The system may face DDoS attacks; employ robust security measures and have contingency plans for offline voting.
- *Key Management:* Securely manage cryptographic keys to prevent unauthorized access; implement recovery mechanisms and educate users.
- *Regulatory Compliance:* Ensure the system complies with election laws and data protection regulations to avoid legal issues; engage legal experts early.
- *Auditability:* Provide transparency and verifiability in the voting process to build trust; implement public audit logs for independent verification.
- *System Updates and Maintenance:* Updating the system should not disrupt elections; design for modularity and establish careful change management.
- *Interoperability:* The system must integrate with existing electoral processes; develop clear APIs and adhere to data standards.
- *Long-term Data Storage:* Securely store voting records over time to prevent data loss; implement redundant storage and regular integrity checks.