VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belagavi-590018

ACTIVITY ON

Writing

"SOFTWARE REQUIREMENT SPECIFICATION"

In

V SEMESTER

As a Part of Software Engineering and Project Management Course- BCS501

Submitted By

SUMIT KUMAR	**1HK22CS166**
SK SAMINOOR HUSSIAN	**1HK22CS157**
VIVEK SINGH	**1HK22CS186**
VISHWA PATIL	**1HK22CS185**

Under the guidance of
Prof. Khallikkunaisa
Associate Professor
Dept. of CSE, HKBKCE

Department of Computer Science and Engineering
2024-2025

Accredited by NBA

HKBK COLLEGEOF ENGINEERING
22/1, Nagawara , Bengaluru – 5600 045.
E-mail: info@hkbk.edu.in, URL: www.hkbk.edu.in

ACTIVITY

Software Engineering and Project Management-BCS501

Software Requirements Specification - (SRS) Document

1.Introduction

Purpose:

The Software Requirements Specification (SRS) document is essential for guiding the development of the "BLOCKCHAIN VOTING SYSTEM" . It defines the scope, features, and constraints of the system to ensure the project progresses systematically from concept to implementation.

Title:

BLOCKCHAIN VOTING SYSTEM

Scope:

- The scope of the software project involves creating a secure and transparent voting system that leverages blockchain technology and smart contracts. The system will:
- Ensure the secure collection, classification, and verification of votes in real-time through a decentralized blockchain network.
- Enable election administrators, policymakers, and voters to conduct and participate in elections with confidence, ensuring transparency and tamper-proof processes.
- Provide functionalities such as voter authentication, immutable vote storage, and automated vote tallying using smart contracts.

The scope excludes features unrelated to blockchain voting, such as broader election campaign management tools, physical vote collection systems, and processes outside the technical boundaries of blockchain technology.

2.Audience

- **Primary Users**: Election commissions, polling officers, and voters directly participating in the voting process.
- **Secondary Users**: Government agencies, policymakers, and organizations interested in electoral data and results analysis.
- **Stakeholders**: System administrators, blockchain developers, election organizers, and external security auditors.

3.Overall Description

Product Features

- **Real-time Vote Collection**: Securely collecting votes from registered users in real-time through blockchain-enabled platforms.
- **Automatic Voter Validation and Record Maintenance**: Leveraging smart contracts and blockchain to validate voter identities and maintain immutable records of votes.
- **Transparent Results Visualization**: Providing transparent, interactive dashboards to display voting trends and outcomes while maintaining anonymity.
- **Tamper-Proof Voting Ledger**: Using blockchain technology to ensure the integrity and immutability of all votes cast.
- **Integration with External Systems**: APIs for data sharing with election monitoring organizations
- and analytics platforms.

Operating Environment

- Hardware: Decentralized cloud-based servers, high-performance computing nodes, and distributed ledger systems to support blockchain operations.
- Software: Blockchain frameworks (e.g., Ethereum, Hyperledger Fabric), programming languages like Python or Solidity, smart contract deployment tools, and databases (e.g., PostgreSQL, MongoDB) for auxiliary data storage.
- Network: Secure and scalable peer-to-peer network architecture with cryptographic protocols to enable real-time, tamper-proof vote recording and validation.

Design and Implementation Constraints

- **Data Quality**: Ensuring high-quality and structured voter data is essential for accurate voter validation and secure vote processing.
- **Language Diversity**: The system must support multilingual interfaces and datasets to accommodate voters from diverse linguistic backgrounds.
- **Scalability**: The blockchain network should be scalable to handle increasing numbers of voters and transactions during large-scale elections.
- **Privacy & Regulatory Compliance**: Adhering to data privacy laws (e.g., GDPR, CCPA) to ensure voter information and voting data remain secure and compliant with regional regulations.

- **Third-Party Dependencies**: Integrating blockchain frameworks, APIs, or external services may introduce constraints related to licensing, performance, and interoperability.

4.Specific Requirements

Functional Requirements

Data Collection & Ingestion:

- **Fetch voter data from verified databases, such as government electoral rolls.**
- Extract and validate metadata (e.g., voter ID, eligibility status, and region) using secure protocols.

Vote Classification & Validation:

- **Automatically validate votes based on eligibility criteria (e.g., age, citizenship).**
- **Use blockchain smart contracts to ensure real-time vote classification and validation.**

Data Visualization & Analysis:

- **Provide interactive, real-time visualizations of voter turnout and election results using charts and graphs.**

Vote Integrity Verification:

- **Calculate and display the validity of votes using cryptographic proofs for voter authentication.**

External Data Integration:

- **Integrate third-party APIs for voter registration and identity verification to enhance data accuracy.**

Non-functional Requirements

Performance:

- The system must process and record a large number of votes in real time with minimal latency.

Security:

- Implement strong data encryption, access control, and privacy protection mechanisms to safeguard voter data and voting integrity.

Scalability:

- The system should efficiently handle increasing numbers of voters and transactions without performance degradation.

Usability:

- Provide intuitive and user-friendly interfaces for voters and election administrators.

External Interface Requirements:

- **APIs**: Integrate with external APIs for voter registration and verification services.
- **Data Formats**: Support JSON, XML, and custom formats for vote and voter metadata exchange.

Quality Assurance and Testing Requirements:

- **Test Cases**: Ensure the accuracy of voter data, vote validation, and election result calculations.
- **Acceptance Criteria**: Verify the integrity of votes, real-time processing, and user interface satisfaction.

5.Use Cases

Use Case 1: Voter Registration

- **Use Case Name**: Voter Registration
- **Description**: The system registers eligible voters and stores their information securely on the blockchain.
- **Actors**: Voter, election administrators, blockchain system.
- **Preconditions**: The voter must provide valid identification and meet eligibility criteria.

Main Flow:

1. Voter submits personal details and identification documents.
2. System verifies the voter's eligibility using external verification services.
3. Upon successful validation, the voter's data is encrypted and stored on the blockchain.

- **Alternate Flows**: If the voter's eligibility cannot be confirmed, the system provides an error message and suggests resubmission of valid documentation.
- **Postconditions**: The voter is successfully registered and authorized to participate in the election.

Use Case 2: Vote Visualization & Analysis

- **Use Case Name**: Visualization
- **Description**: The system generates interactive visualizations of voting trends and outcomes based on recorded data.
- **Actors**: System users (election administrators, policymakers, public observers).
- **Preconditions**: The system must have processed and stored all voting data.

Main Flow:

1. Users select an election or voting period for analysis.
2. The system generates interactive visualizations, such as bar charts, pie charts, and heatmaps, to represent voting patterns and turnout.
3. Users explore voting trends and outcomes through the visual tools provided.

- **Alternate Flows**: If no voting data is available for the selected criteria, the system prompts the user to modify their query.

- **Postconditions**: Users gain insights into voter turnout and election results

---

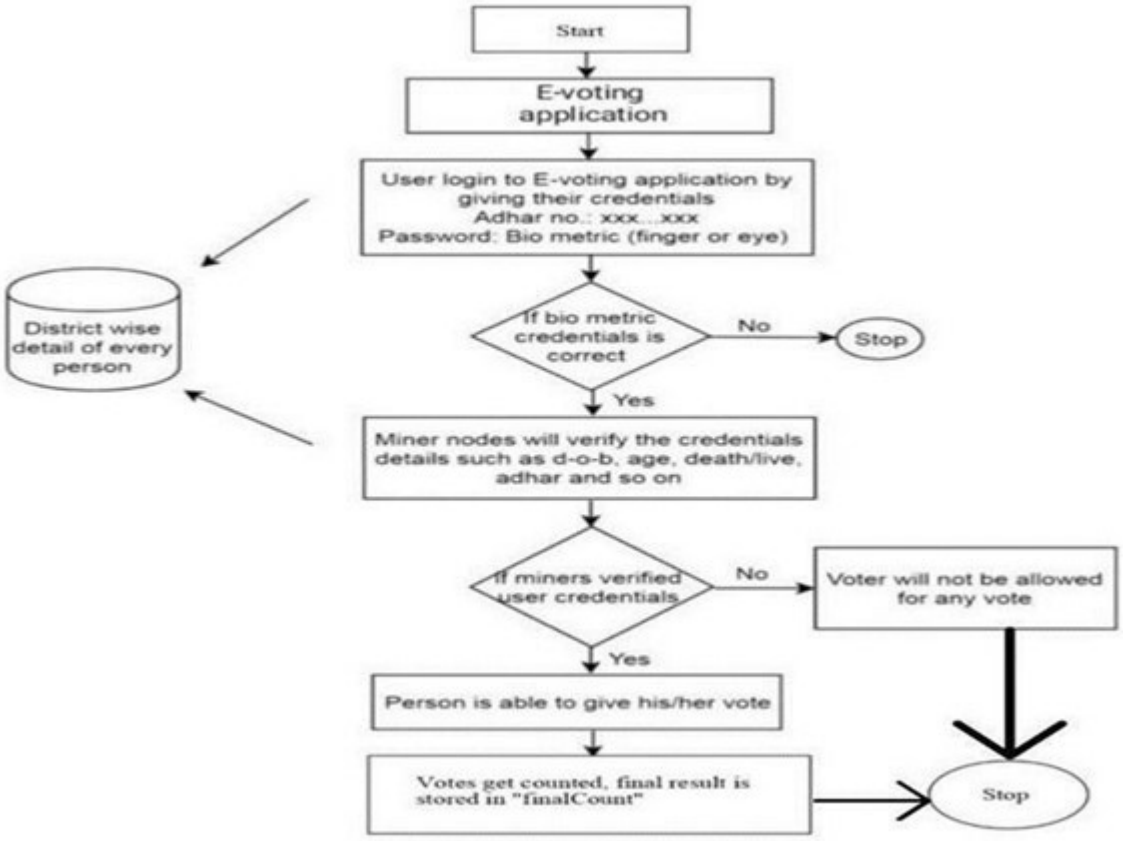
Use Case 3: Vote Counting

- **Use Case Name**: Vote Counting
- **Description**: The system automatically counts the votes in a secure, transparent manner using blockchain technology.
- **Actors**: Election officials, System
- **Preconditions**: All votes have been cast and recorded on the blockchain.

Main Flow:

1. Election officials request the vote count.
2. The system retrieves and verifies all recorded votes from the blockchain.
3. The system calculates the total votes for each candidate or option.
4. The system displays the final results.

- **Alternate Flows**: If there are discrepancies or missing votes, the system alerts the officials and requests further verification.
- **Postconditions**: The vote counting is completed, and the results are accurately displayed and stored on the blockchain.



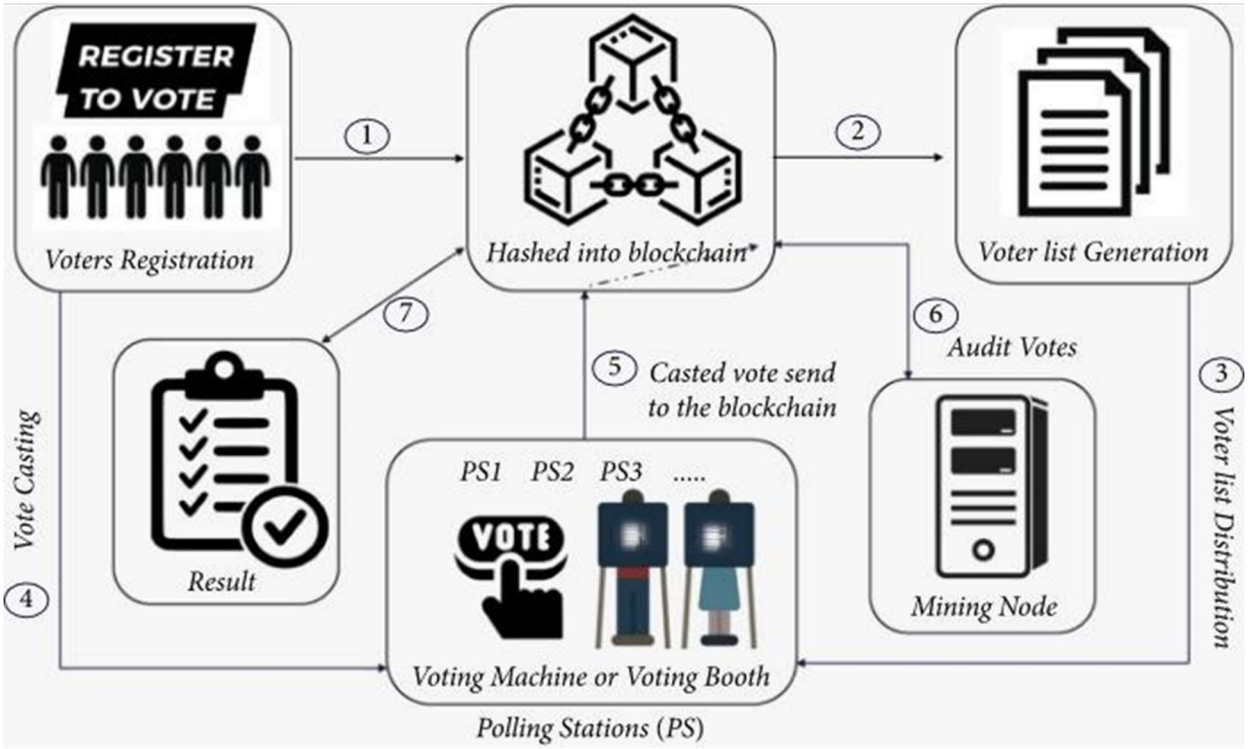
USE CASE DIAGRAM

6.System Architecture

- **Components:**

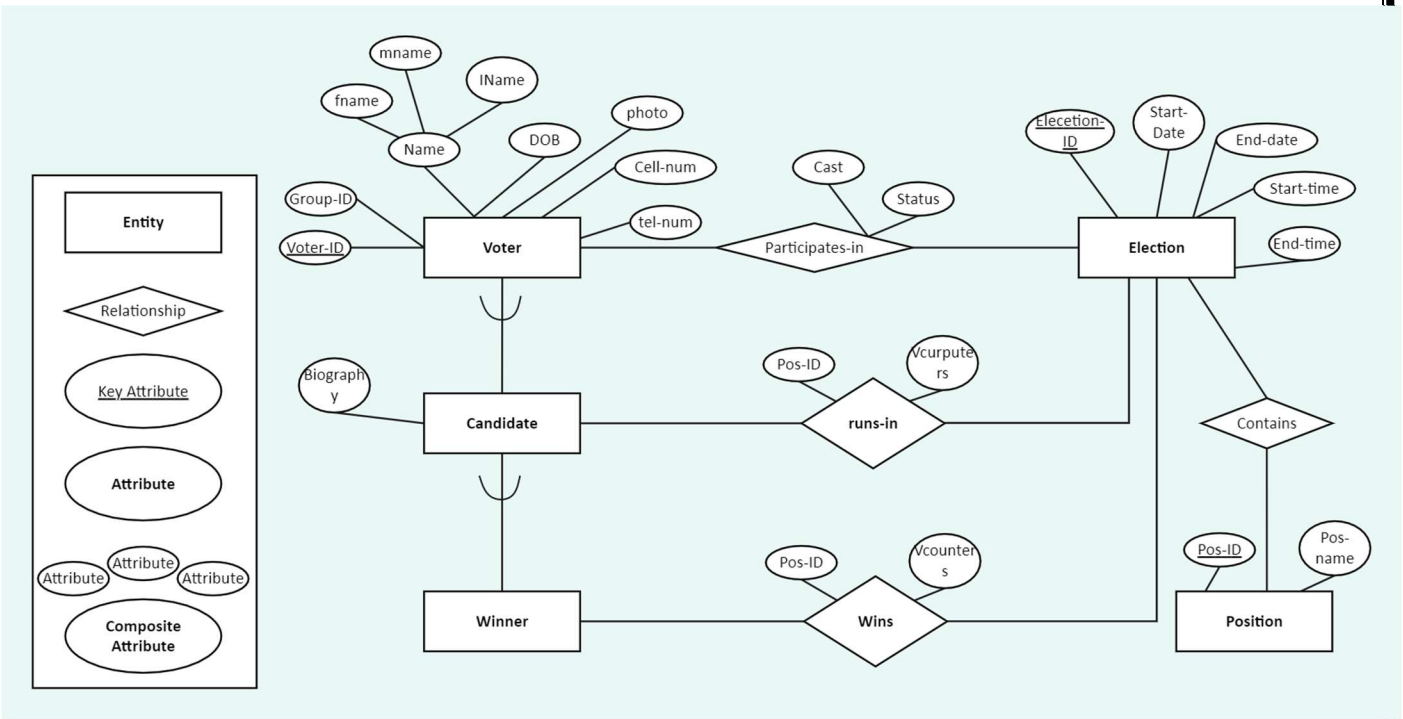
- **Blockchain Network Module**: Manages the distributed ledger for voting transactions.
- **Voter Authentication Module**: Verifies voter identities using blockchain-based credentials (e.g., biometric data, smart IDs).
- **Vote Casting Module**: Facilitates secure vote casting and records votes on the blockchain.
- **Vote Verification Module**: Ensures the integrity of voting results by accessing the blockchain for validation.
- **Database**: Stores voter registration data, election data, and blockchain logs for auditing and retrieval.
- **User Interface**: Provides easy access to voting, result verification, and authentication processes for users and election officials.

- **Architectural Diagram:**

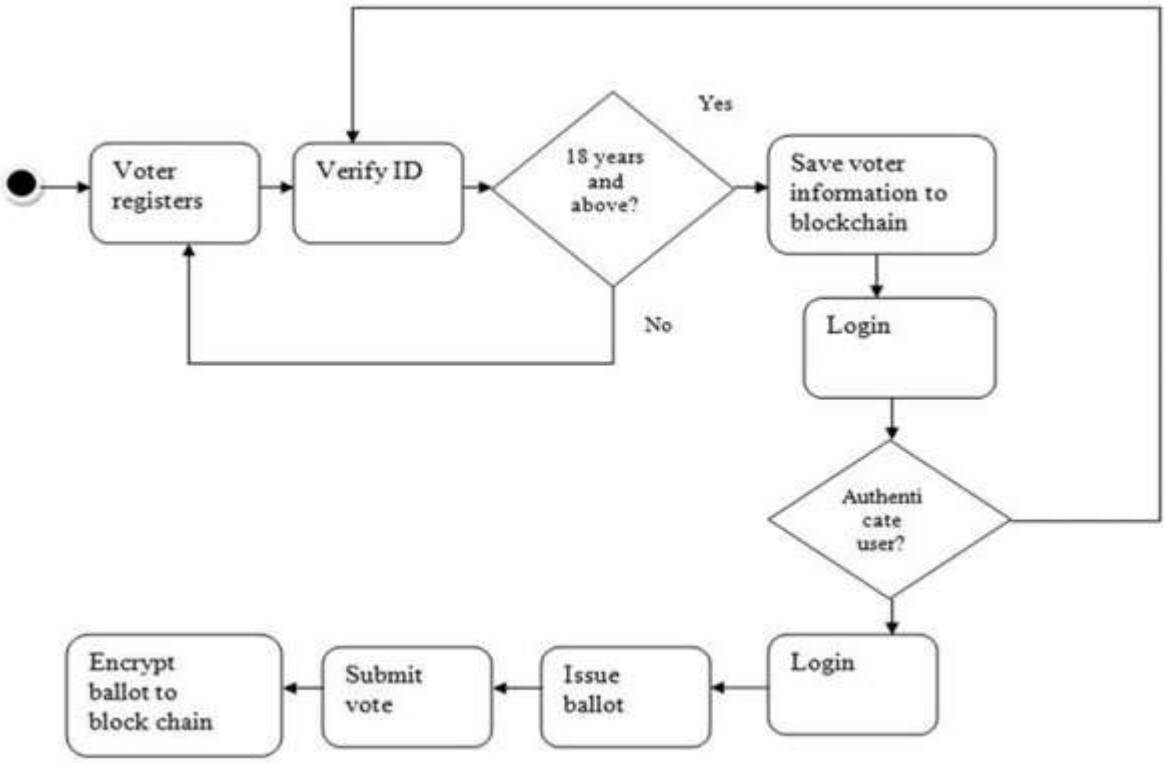


7.Data Models

- **Entity-Relationship Diagram (ERD):**



Data Flow Diagram:



8.External Interfaces

APIs:

Blockchain network APIs for interacting with the distributed ledger.
○ Voter identity verification APIs for authentication.
○ Election result APIs for retrieving and verifying voting outcomes.

Data Formats:

○ JSON, XML, CSV for data exchange.

Protocols:

○ REST APIs for vote casting, authentication, and result retrieval.

9.Quality Attributes

Performance:

○ System should process voting data in real-time with minimal latency.

Security:

○ Implement data encryption, access control, and compliance with election security laws.

Scalability:

○ The system must handle increasing voter registrations, vote casting, and result verification efficiently.

Usability:

○ Provide intuitive interfaces and tools for voters, election officials, and administrators.

10. Appendics

a. Glossary:

Blockchain: A distributed ledger technology used for securely recording transactions.

Voter Authentication: The process of verifying the identity of a voter.

Smart Contracts: Self-executing contracts with the terms of the agreement directly written into code.

Distributed Ledger: A database that is consensually shared, replicated, and synchronized across multiple sites.

b. References:

i. List of research papers, tools, and APIs used.
ii. Diagrams: Additional diagrams like sequence diagrams, state diagrams, etc.

11.Review and Approval:

Faculty In-charge	TEAM MEMBERS	
Prof. Khallikkunaisa	SUMIT KUMAR	1HK22CS166
	SK SAMINOOR HUSSIAN	1HK22CS157
	VIVEK SINGH	1HK22CS186
	VISHWA PATIL	1HK22CS185