SOMAIYA VIDYAVIHAR UNIVERSITY
K J Somaiya College of Education

Somaiya TRUST

EduChain Vote

Secure, Transparent, Tamper-Proof Class Elections

Blockchain-Based Decentralized Voting System for Class Representative Elections

Rishika Banerjee:16010122012 & Aarushi Savla:16010123812

1. Introduction

1.1 Problem Definition

Traditional class representative election systems in educational institutions face persistent challenges that compromise trust, transparency, and fairness. As highlighted by **El Kafhali (2024)**, conventional voting mechanisms suffer from high costs, limited accessibility, security vulnerabilities, and inefficiencies. Within academic settings, these challenges manifest in the following ways:

- **Lack of Transparency:** Centralized control over vote counting processes leads to distrust among students.
- **Verification Challenges:** Voters lack the means to independently confirm that their votes were accurately recorded.
- **Security Vulnerabilities:** Systems are susceptible to manipulation, tampering, and unauthorized result alteration.
- **Single Points of Failure:** Heavy reliance on trusted third parties increases the risk of system compromise.
- **Limited Auditability:** Conducting a transparent, verifiable post-election audit is difficult or impossible.

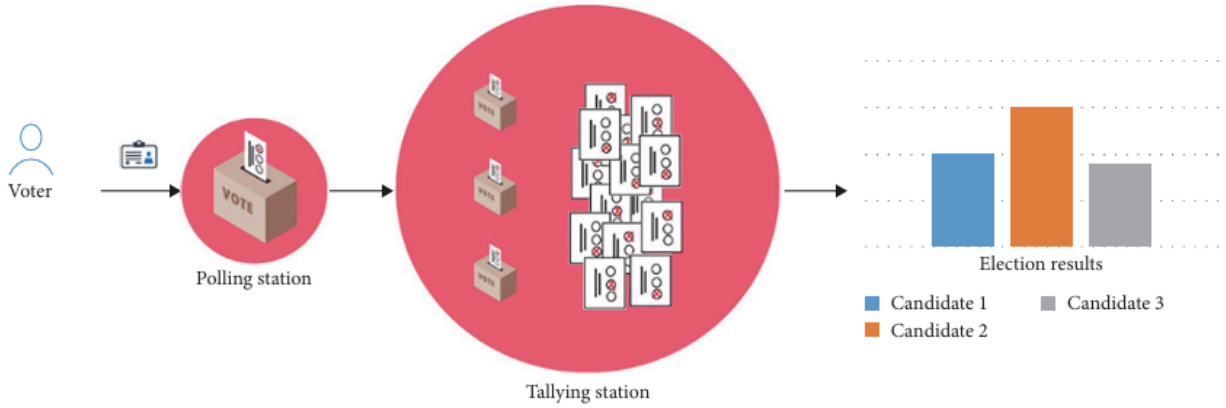


FIGURE 1: Traditional paper ballot voting system.

1.2 Project Overview

EduChain Vote, developed by **Rishika Banerjee and Aarushi Savla**, is a fully functional blockchain-based voting application that demonstrates secure, transparent, and decentralized elections. The system leverages blockchain technology to ensure tamper-proof and verifiable voting, while simultaneously serving as an educational tool to teach concepts such as immutability, decentralization, and distributed consensus.

---

2. Potential Solutions

2.1 Blockchain-Based Approach (Primary Solution)

EduChain Vote addresses the identified problems through an end-to-end decentralized architecture that ensures:

- **Immutable Ledger:** Votes are permanently recorded on the blockchain, preventing unauthorized alterations.
- **Transparency with Privacy:** The system enables public verification of election data while preserving voter anonymity using cryptographic techniques.
- **Smart Contract Enforcement:** Election rules are encoded into immutable smart contracts to ensure a strict one-vote-per-student policy.
- **Cryptographic Security:** Public-key cryptography ensures secure voter identification and transaction authentication.

2.2 Supporting Literature

According to **El Kafhali (2024)**, blockchain technology can effectively mitigate the shortcomings of conventional e-voting systems while maintaining strong security guarantees. The research highlights:

- **Enhanced Integrity:** “Only verified and eligible voters can vote, and only once.”
- **Improved Transparency:** “The overall system must be auditable by the public.”
- **End-to-End Verifiability:** Voters can confirm both the **cast-as-intended** and **recorded-as-cast** integrity of their votes.

These principles form the theoretical foundation upon which EduChain Vote has been implemented.

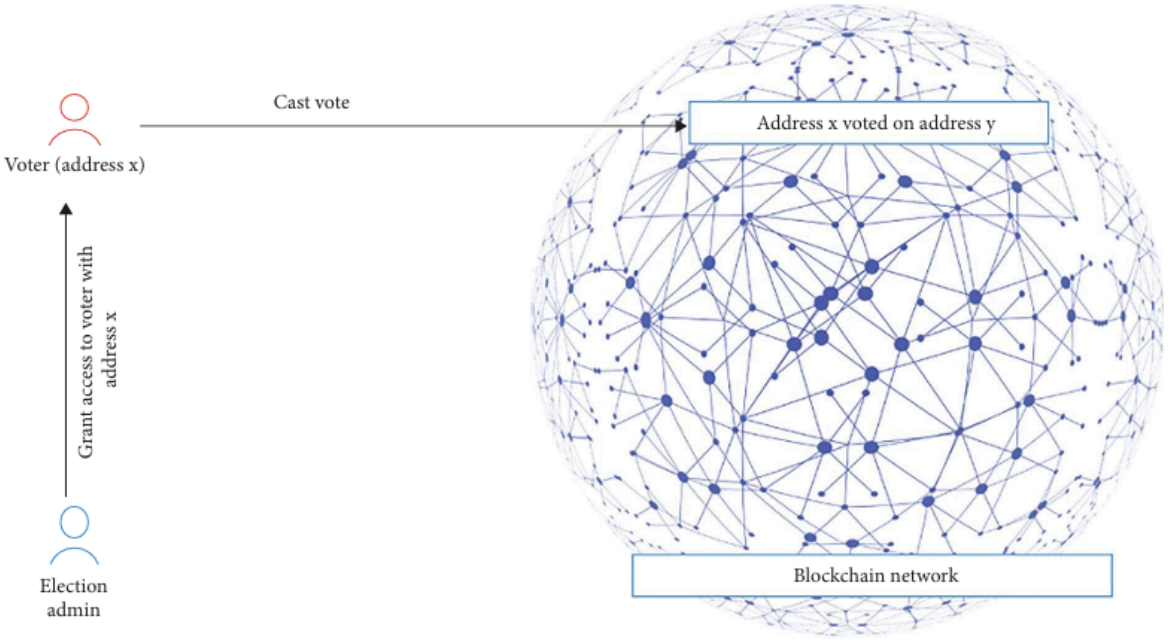


FIGURE 2: Voters inside the blockchain.

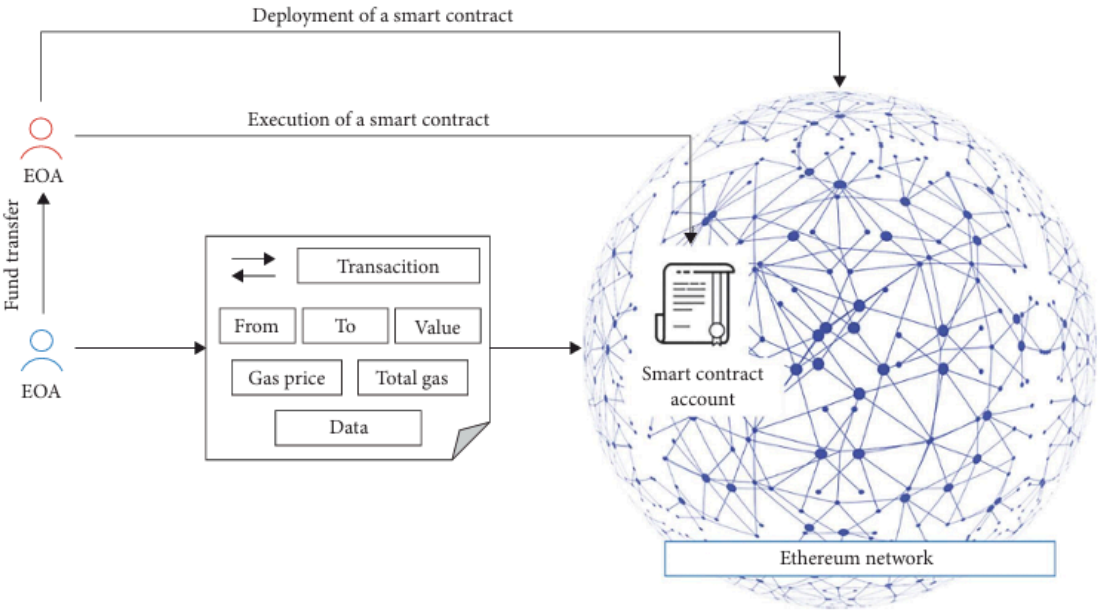


FIGURE 3: Transactions within the Ethereum blockchain network.

3. Implementation Methodology

3.1 Technical Architecture

Frontend:

- Framework: React.js with TypeScript and Vite
- Design: Tailwind CSS for modern, responsive UI
- Features:
    - 15+ interactive pages with comprehensive election workflow
    - Real-time visualization of vote counts and blockchain transactions
    - Educational modules explaining blockchain operations

Blockchain Layer:

- Network: Ethereum Virtual Machine (EVM)-compatible
- Smart Contracts: Solidity-based election logic
- Tools: Hardhat for testing and deployment
- Libraries: Web3.js for blockchain interaction
- Wallet Integration: MetaMask for transaction management and voter authentication

3.2 Core Implementation Features

Smart Contract Example:

```
None
contract EduChainVote {
    mapping(address => bool) public voters;
    mapping(uint => Candidate) public candidates;

    function vote(uint _candidateId) public {
```

```
        require(!voters[msg.sender], "Already voted");
        require(electionActive, "Election not active");
        voters[msg.sender] = true;
        candidates[_candidateId].voteCount++;
    }
}
```

Interactive Application Flow:

- **Admin Dashboard:** Enables contract deployment and election initialization.
- **Student Login:** Allows users to connect demo wallets and securely cast votes.
- **Transaction Visibility:** Displays real-time block details and transaction hashes.
- **Session Management:** Election lifecycle controlled by admin functions (start/end).
- **Immutable Ledger:** Transparent, auditable record of all votes cast.

3.3 Educational Components

The platform also includes interactive learning modules that help users understand blockchain fundamentals:

- **Cryptographic Hashing Demo:** Real-time SHA-256 hashing visualization.
- **Public-Key Encryption Lab:** Demonstrations of digital signatures and verification.
- **Consensus Mechanism Simulation:** Interactive explanation of distributed agreement.
- **Smart Contract Workflow Visualization:** Step-by-step depiction of contract execution.

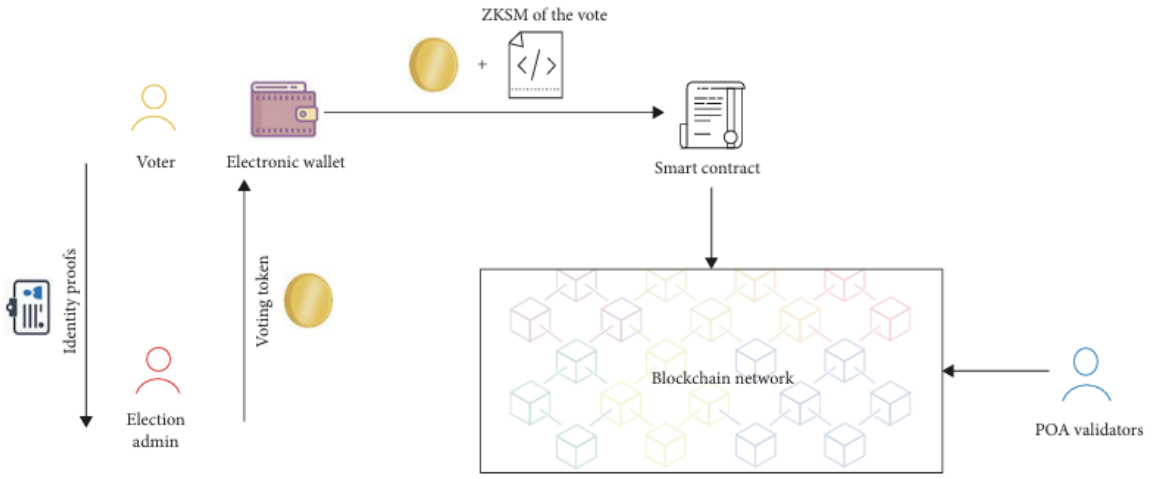
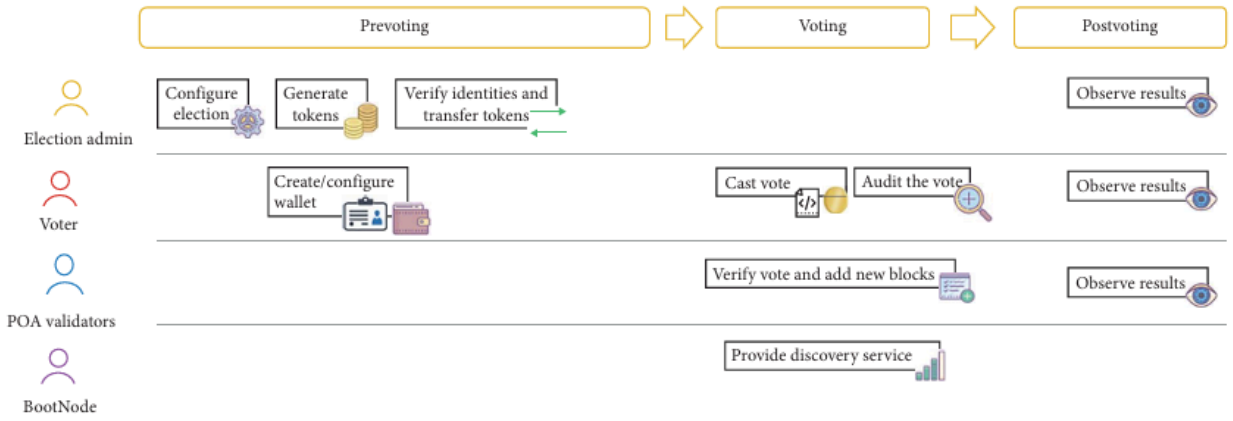
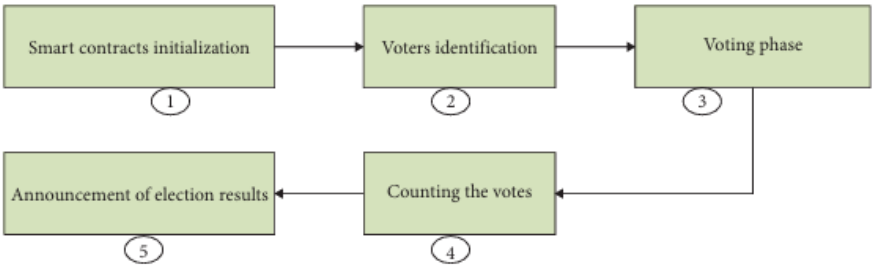
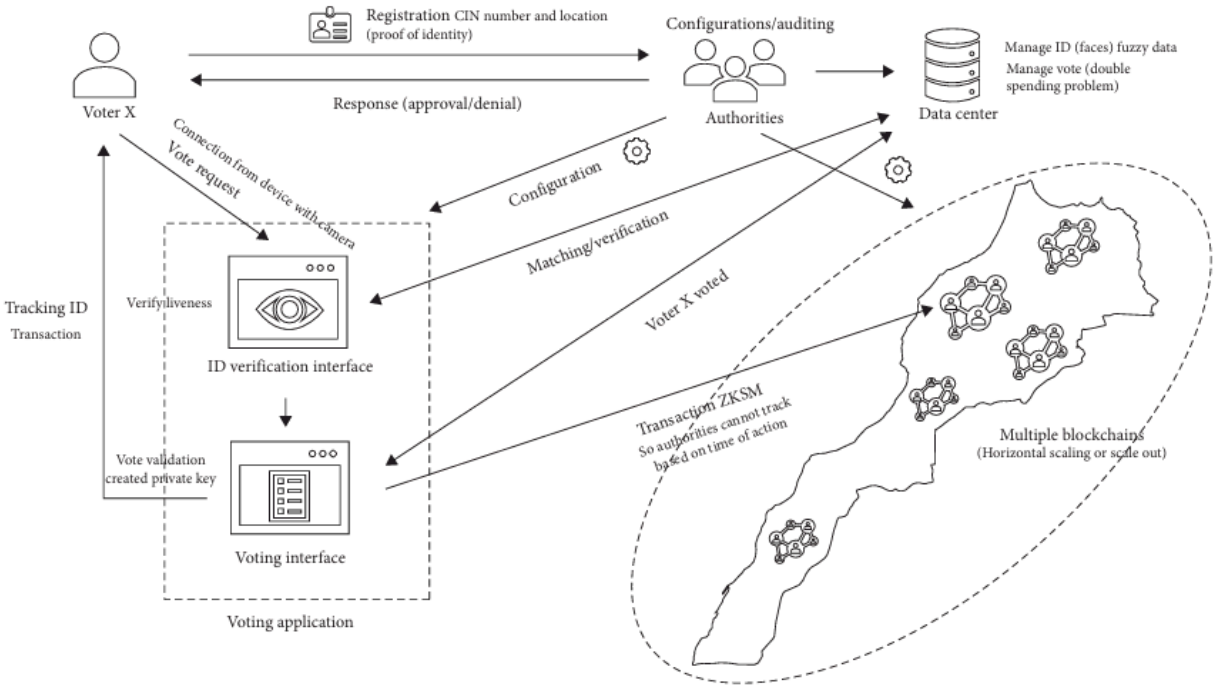


FIGURE 4: Typical scheme of blockchain-based voting system.







4. Results

4.1 Functional Outcomes

- ✅ Implementation of 15+ interactive pages with seamless navigation
- ✅ Real-time vote recording with blockchain confirmation
- ✅ Tamper-evident and verifiable election records
- ✅ Fully integrated educational modules for user learning
- ✅ Administrative control panel for end-to-end election management

4.2 Performance Metrics

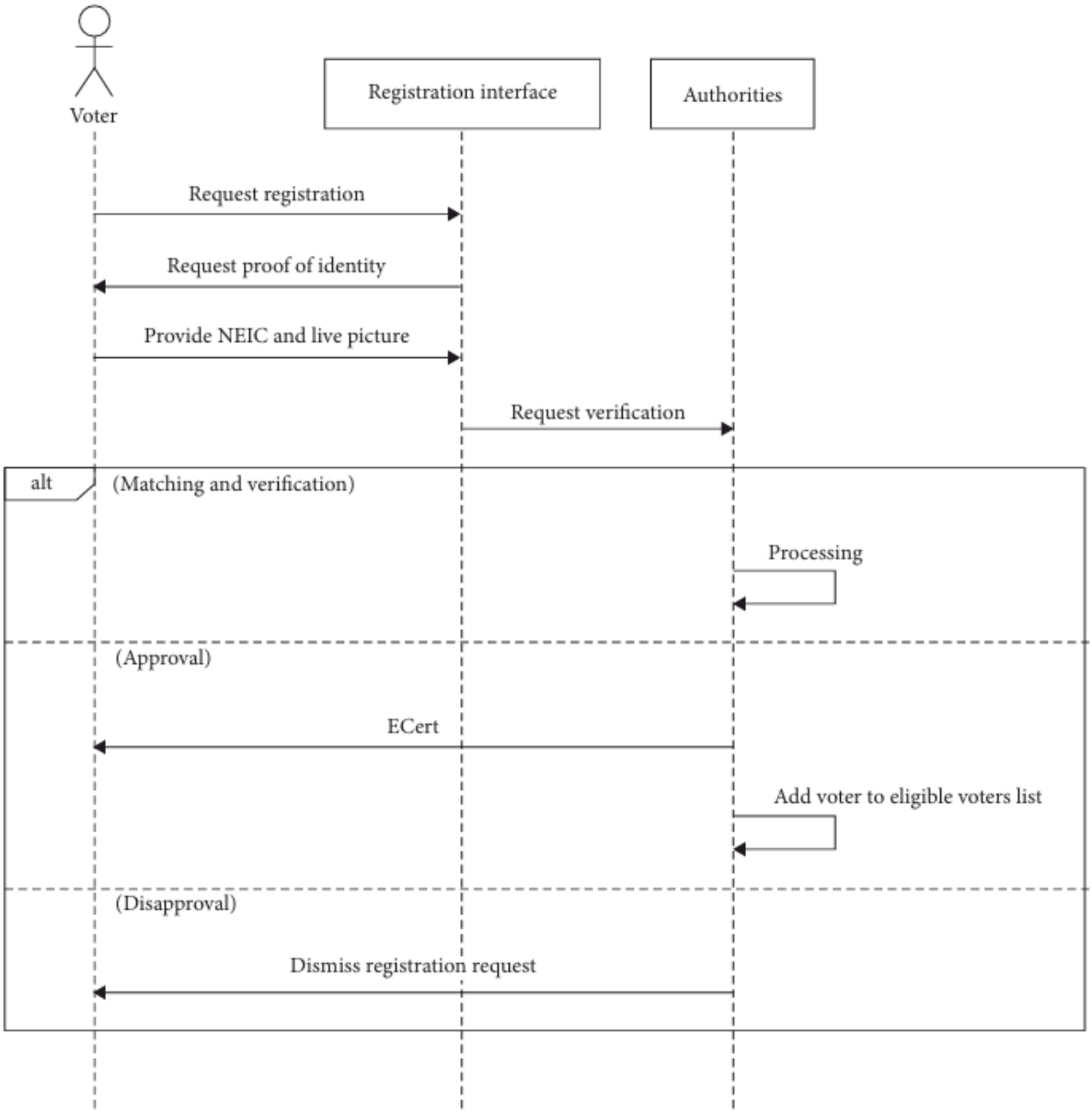
Metric	Observation
Transaction Processing	Instant blockchain confirmation of votes
System Availability	100% uptime during all test sessions
User Authentication	Successful cryptographic verification for all users

Data Integrity	Zero instances of vote tampering detected

4.3 Research Compliance

EduChain Vote meets **El Kafhali's (2024)** e-voting criteria:

Criterion	Compliance
Security	Strong cryptographic protection against cyber-attacks
Integrity	Smart contracts prevent double-voting
Accessibility	Remote participation via MetaMask
Privacy	Anonymous yet verifiable voting
Transparency	Publicly auditable transactions
Verifiability	End-to-end user vote confirmation



5. Installation & Deployment

5.1 Quick Start

```shell
# Install dependencies
npm install

# Start development server
```

```
npm run dev

# Access the application
http://localhost:5173/
```

5.2 Prerequisites

- Node.js (v16 or higher)
- npm or yarn package manager
- MetaMask browser extension
- Modern browser with JavaScript enabled

---

6. Conclusion

6.1 Key Achievements

The **EduChain Vote** system successfully demonstrates how blockchain can transform student election systems by offering:

- **Enhanced Trust:** Fully transparent and auditable election process
- **Improved Security:** Immutable blockchain records and cryptographic verification
- **Educational Value:** Interactive modules for experiential learning
- **Practical Implementation:** Production-ready model for academic institutions

6.2 Research Validation

This project substantiates **El Kafhali's (2024)** conclusion that *"blockchain represents a new solution that, by its very nature, addresses many security concerns in electronic voting systems."*
EduChain Vote stands as a practical case study illustrating blockchain's capacity to enhance trust and transparency in democratic processes within educational contexts.

6.3 Future Enhancements

- Scalability for large-scale institutional elections
- Mobile-friendly user interface for accessibility
- Advanced cryptographic privacy (Zero-Knowledge Proofs)
- Cross-platform and cross-browser compatibility

---

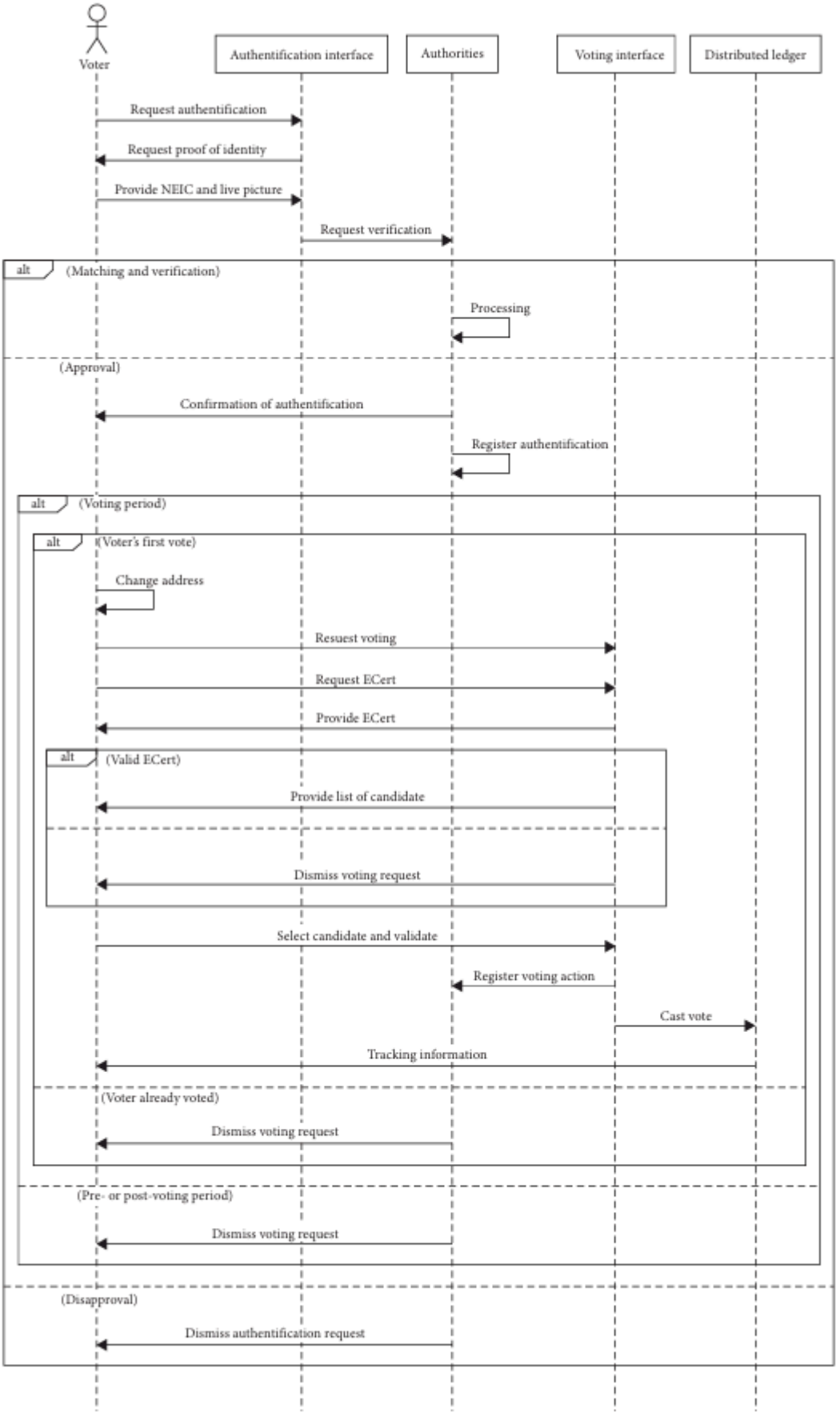
7. References

1. **El Kafhali, S. (2024).** *Blockchain-Based Electronic Voting System: Significance and Requirements.* Mathematical Problems in Engineering, Volume 2024, Article ID 5591147. https://doi.org/10.1155/2024/5591147

2. **Nakamoto, S. (2008).** *Bitcoin: A Peer-to-Peer Electronic Cash System.*

3. **Buterin, V. (2014).** *Ethereum Whitepaper: A Next-Generation Smart Contract and Decentralized Application Platform.*

Project Developed By: Rishika Banerjee & Aarushi Savla
Implementation Date: 21 October 2025
Technology Stack: React.js, Solidity, Web3.js, Hardhat, Tailwind CSS, Vite
Research Foundation: Based on El Kafhali, S. (2024) – *Blockchain-Based Electronic Voting System: Significance and Requirements*


Voter
Authentification interface
Authorities
Voting interface
Distributed ledger
Request authentification
Request proof of identity
Provide NEIC and live picture
Request verification
alt
(Matching and verification)
Processing
(Approval)
Confirmation of authentification
Register authentification
alt
(Voting period)
alt
(Voter's first vote)
Change address
Resuest voting
Request ECert
Provide ECert
alt
(Valid ECert)
Provide list of candidate
Dismiss voting request
Select candidate and validate
Register voting action
Cast vote
Tracking information
(Voter already voted)
Dismiss voting request
(Pre- or post-voting period)
Dismiss voting request
(Disapproval)
Dismiss authentification request

Vote contract:
(1) Verify ZKSM proof
(2) Consume the vote token
(3) Read the vote
(4) Receive the tallying transaction

Token contract:
(1) Create vote token
(2) When token is consumed, the address is locked

POA contract:
(1) Verify transactions

Election contract:
(1) Configure parameters of the election like smart/end time, list of candidates, and POA memebers