October 2023

SYSTEMS DESIGN DOCUMENT

E-Voting System

by Bit System Legion

Table of Contents

Introduction

The purpose of this documentation is to give an overview of how the application works. It includes the system design, system requirements and interfaces.

Purpose and scope

The purpose of this System Design Document is to describe the creation of our application and confirm that the design conforms with the requirements stated in our project requirements. You will receive a general overview of our database design, security, software, and hardware herein.

Project Executive Summary

The overall basis of our project is to conduct and provide a comprehensive analysis of the Blockchain-Based Voting System which is designed to transform the way elections are conducted by leveraging blockchain technology. This innovative system ensures the voting process's integrity, security, and transparency, providing a trusted and tamper-proof platform for citizens to exercise their voting rights.

The goal is to create a reliable and secure voting platform based on blockchain technology to guard against fraud, tampering, and illegal access to the election process. By offering a simple, remote voting process, we hope to increase voter accessibility and involvement. Establishing an open and verifiable voting process will increase public confidence in democratic elections, Make the counting of votes more efficient and error-proof to produce timely and accurate election results. Enable safe and anonymous voting while preserving the privacy of each voter's preferences.

Key Features:

a. Blockchain Infrastructure:

- Implement a decentralized blockchain network to record and store votes securely.
- Utilize cryptographic techniques for immutability and tamper resistance

b . Voter Authentication:

- Employ robust identity verification methods to ensure only eligible voters participate.
- Enable multi-factor authentication for added security.

c. Digital Ballots:

- Create digital ballots that replicate the traditional paper ballot format.
- Include accessibility features for all voters, including those with disabilities.

d. End-to-End Encryption:

- Encrypt all data transmission and storage to protect voter privacy.
- Utilize public and private key pairs to secure individual votes.

e. Transparent Voting Process:

- Record every vote on the blockchain, allowing voters to verify their choices.
- Enable real-time monitoring by election officials and stakeholders.

f. Immutable Audit Trail:

- Maintain a comprehensive and immutable audit trail for transparency and accountability.
- Allow for post-election audits to verify results.

g. Decentralized Consensus:

- Utilize a consensus mechanism (e.g., Proof of Stake or Proof of Work) to validate and record votes.
- Prevent any single entity from controlling the voting process.

Security Measures:

Implement advanced cybersecurity protocols to protect against hacking and unauthorized access. Conduct regular security audits and penetration

testing to identify and address vulnerabilities. Employ zero-knowledge proofs or similar techniques to protect voter anonymity.

User Education and Support:
Provide extensive user education resources to ensure voters can use the blockchain-based system confidently. Offer a support system for technical assistance during the voting process.

Legal and Regulatory Compliance:
Collaborate with election authorities to ensure compliance with election laws and regulations.
Seek the necessary approvals and endorsements for implementing blockchain in elections.

Pilot Program:
Initiate a pilot program to test the system's functionality and gather feedback from users. Make improvements based on pilot program results and address any identified issues.

Scaling and Deployment:
Gradually introduce the blockchain voting system in local elections before scaling to regional and national elections. Continuously refine the system based on user experiences and evolving security standards.

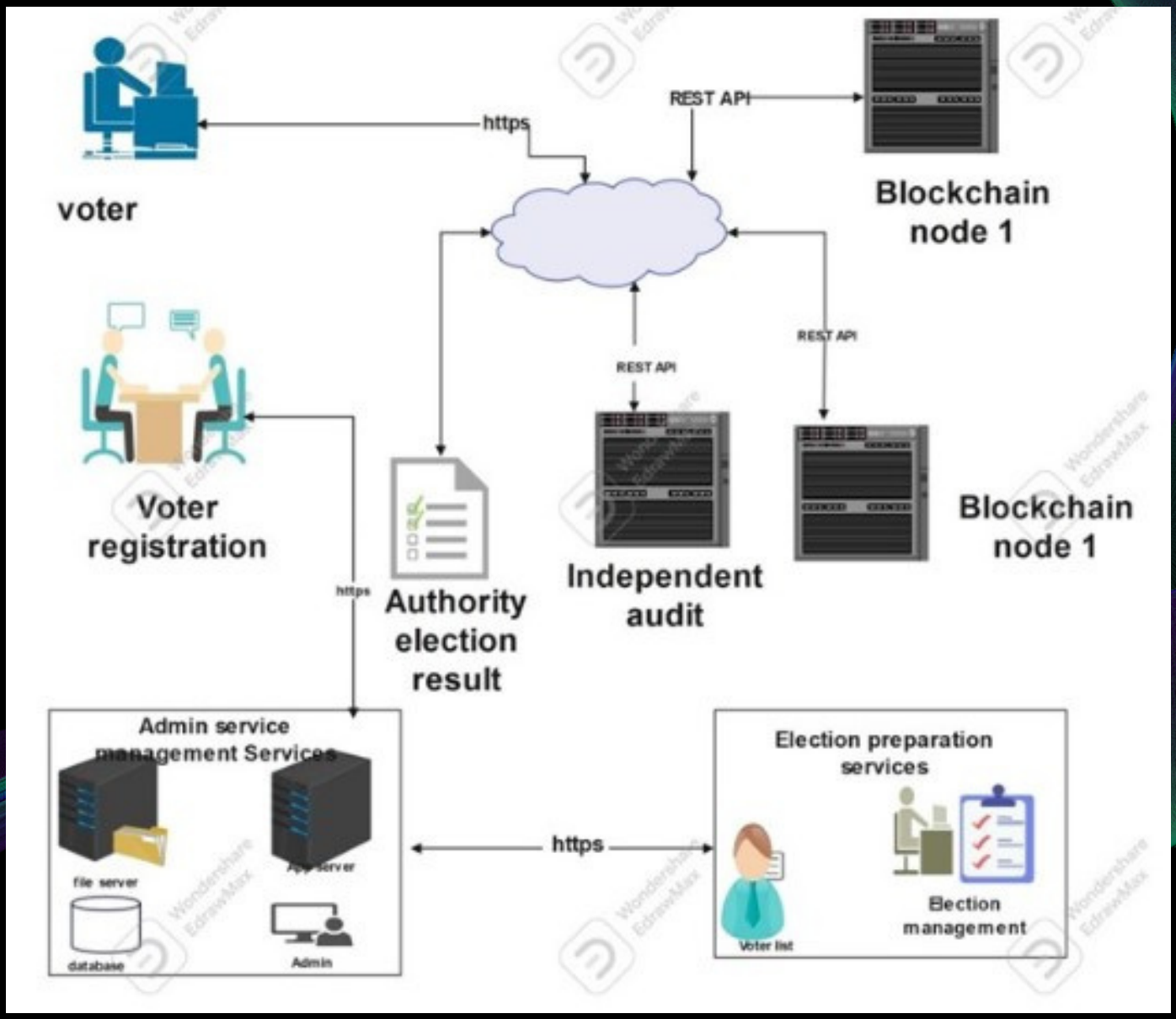
In conclusion, The Blockchain-Based Voting System project offers a game-changing innovation in the field of election procedures. By utilizing blockchain technology, it assures voting is secure, transparent, and accessible, ultimately improving the basis of democracy and restoring faith in the democratic process.

Risk Management

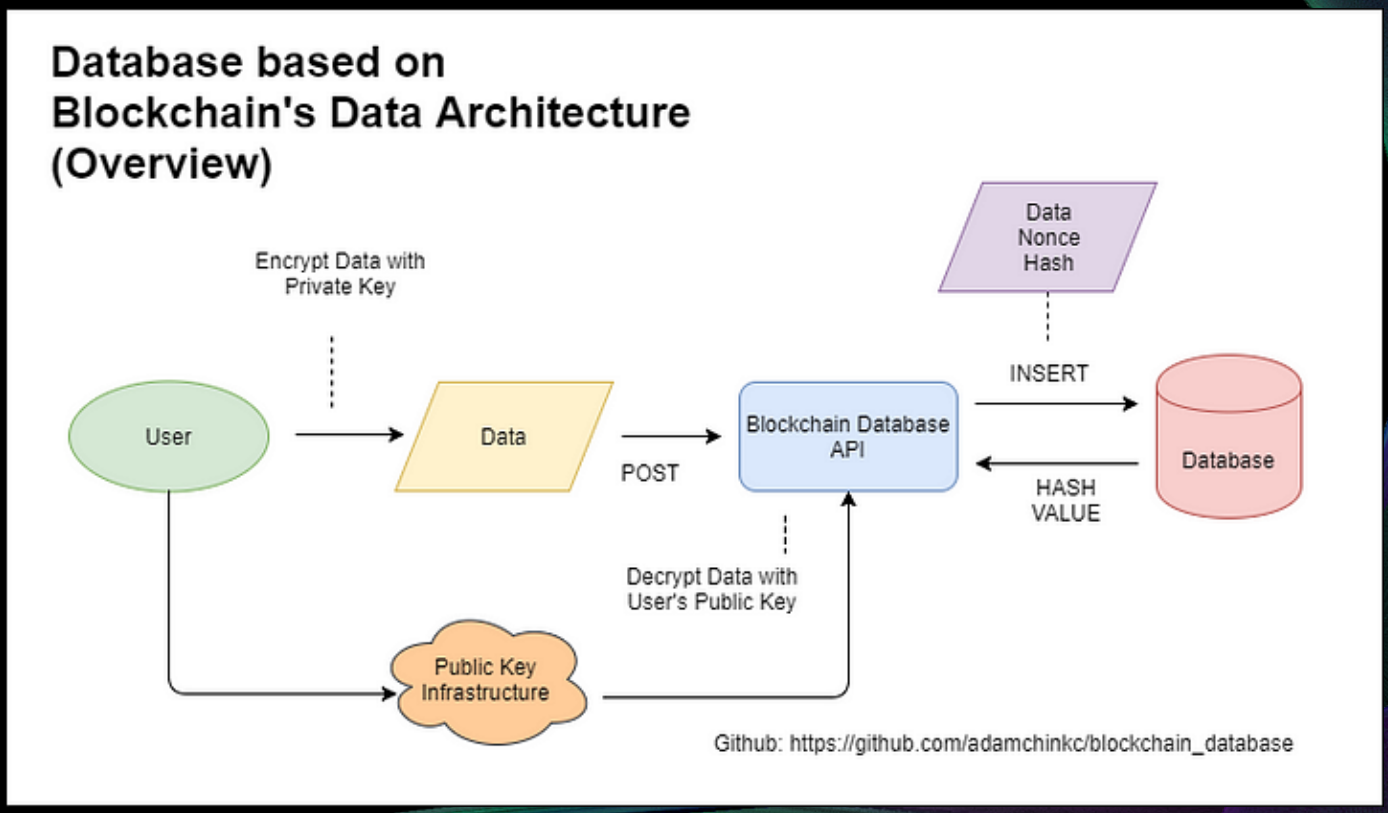
Risk	Impact	Likelihood of Occurrence	Risk Level	Mitigation plan	Responsibilities
Data Integrity Risks	Tampering with the blockchain data could compromise the integrity of the election results and lead to inaccurate outcomes.	MEDIUM	HIGH	Implement strong cryptographic algorithms to ensure data immutability and use consensus mechanisms like Proof of Work (PoW) or Proof of Stake (PoS) to prevent unauthorized modifications.	Backend Developers
Smart Contract Vulnerabilities	Smart contracts, which govern the voting process, may contain coding errors or vulnerabilities that could be exploited by malicious actors.	LOW	LOW	Implement strong cryptographic algorithms to ensure data immutability and use consensus mechanisms like Proof of Work (PoW) or Proof of Stake (PoS) to prevent unauthorized modifications.	Front-end developers
DDoS Attacks on Blockchain Network	A Distributed Denial of Service (DDoS) attack on the blockchain network hosting the electoral system could disrupt the voting process.	HIGH	HIGH	Utilize a decentralized blockchain network to reduce the impact of DDoS attacks and implement DDoS protection measures at the network level.	Back-end Developers(System Analyst)

Privacy and Confidentiality	The transparent nature of blockchain can compromise voter privacy if not adequately addressed.	LOW	HIGH	Utilize zero-knowledge proofs or other privacy-preserving techniques to ensure that individual voting choices remain confidential while still maintaining the integrity of the overall election process	Back-end developers
Insider Threats	Insiders with privileged access to the blockchain-based electoral system could potentially manipulate data or interfere with the voting process.	HIGH	HIGH	Implement strict access controls and regular monitoring of system administrators' activities to detect and prevent any unauthorized actions.	Back-end developers(Data analyst)
Blockchain Network Consensus	Legal and regulatory frameworks surrounding elections may not be fully compatible with blockchain-based systems.	LOW	MEDIUM	Select a widely accepted and secure consensus algorithm, and conduct thorough testing and validation of the network's performance before deploying the electoral system.	Backend developers

System Architecture



Database Architecture



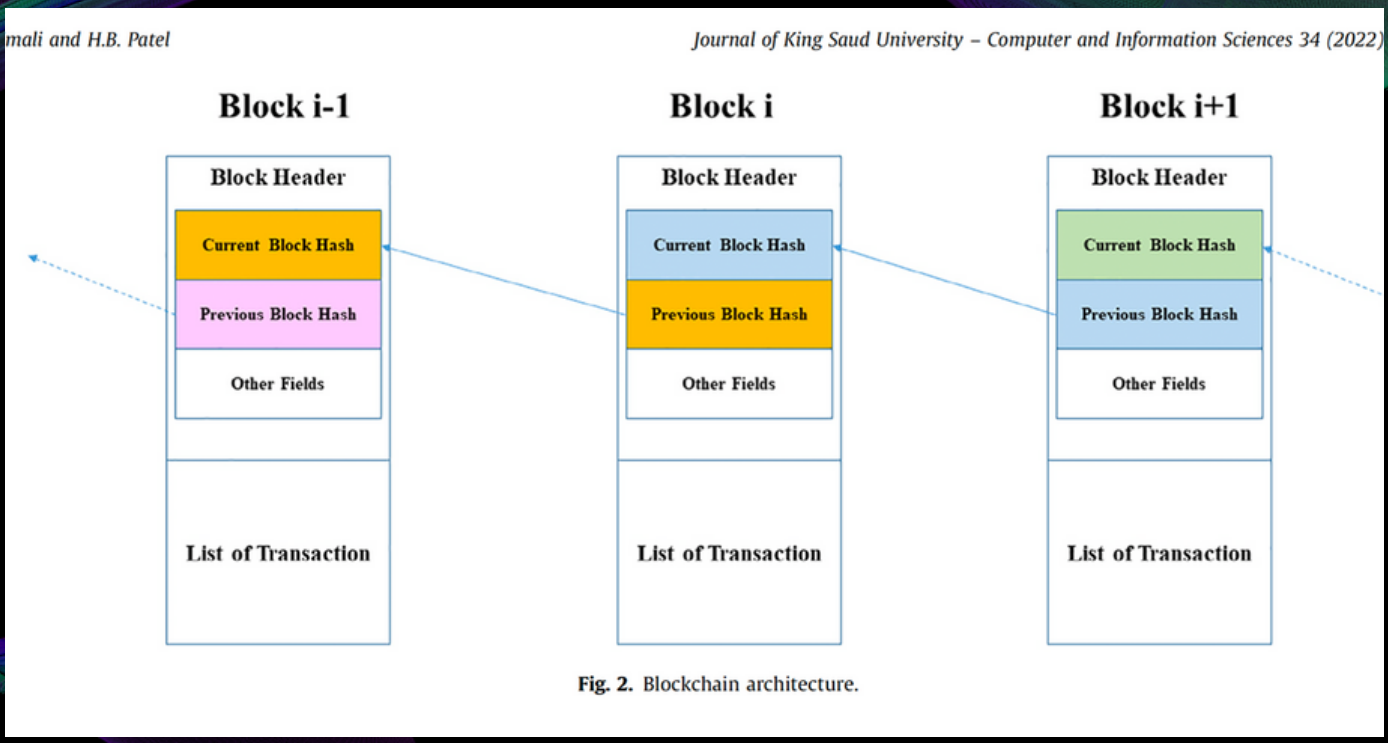
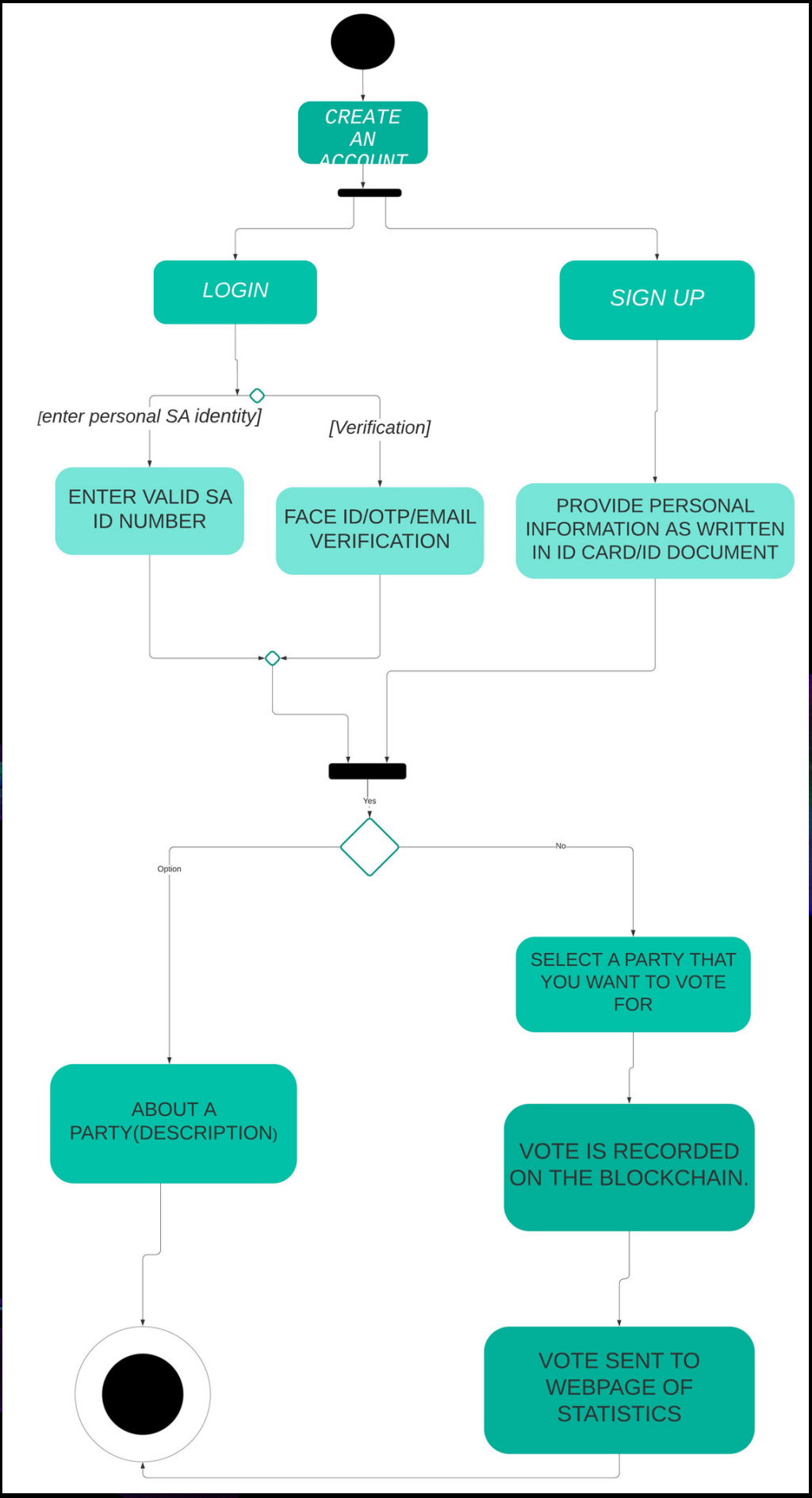


Fig. 2. Blockchain architecture.

Dataflow Diagram



System Integrity Controls

- To fortify the system, we will implement an interface where a user can upload a picture of them and their ID to verify if the user is really who they say they are on the mobile app.
- The app also sends an email verification as another measure of ensuring the system's integrity.
- An OTP will also be sent to the registered number when signing up the user

External Interfaces

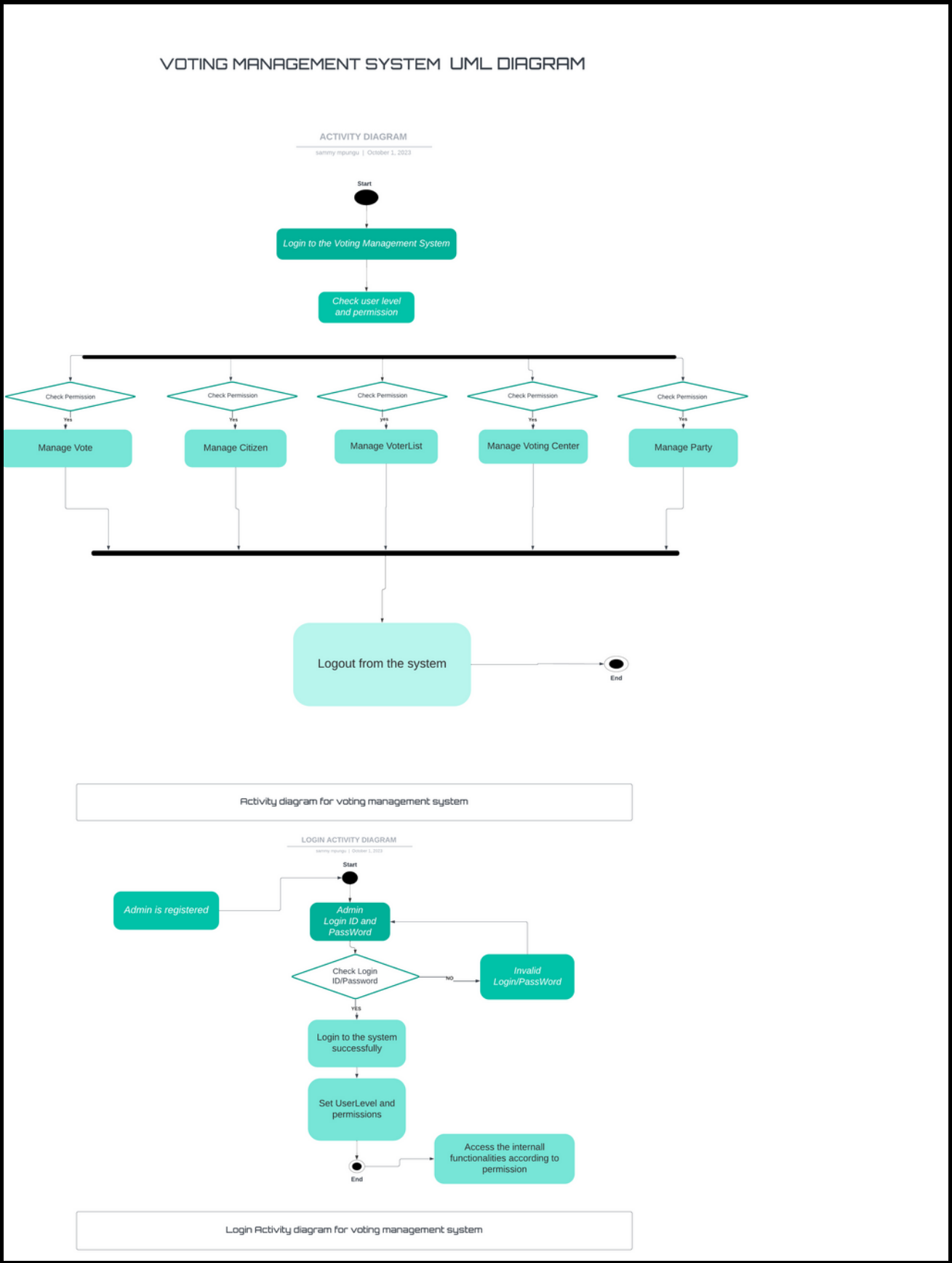
- The external interface we have implemented is the Vote Count Statistics webpage which we have created for our stakeholders, the IEC, to keep track of all the statistics when it is time to count the votes in each category.

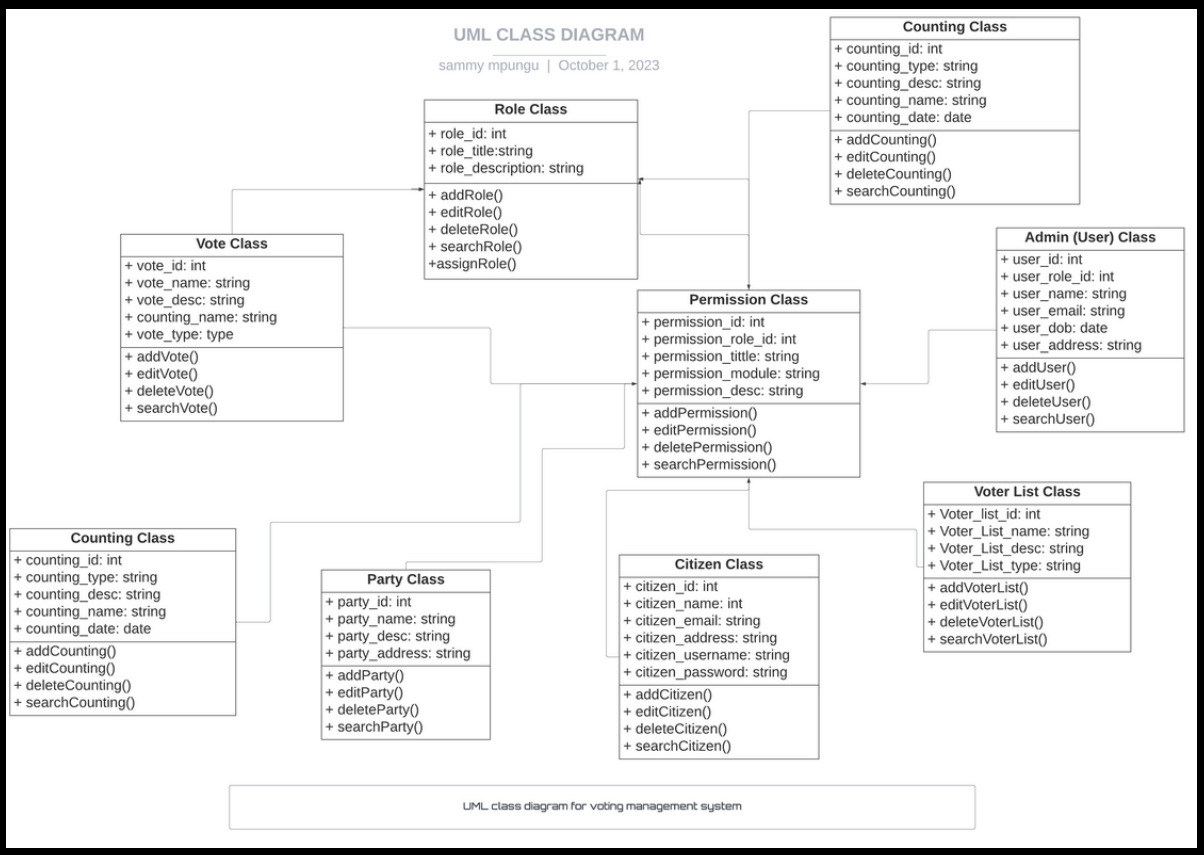
Design Constraints

Technical Difficultes

- We thought of using fingerprint sensors as authentication but could not implement this because we realized it wouldn't be inclusive of our target market. Not all mobile devices have the fingerprint sensor hence we decided to not go further with it.

Appendix A


VOTING MANAGEMENT SYSTEM UML DIAGRAM
ACTIVITY DIAGRAM
Start
Login to the Voting Management System
Check user level and permission
Check Permission
Check Permission
Check Permission
Check Permission
Check Permission
Yes
Manage Vote
Manage Citizen
Manage VoterList
Manage Voting Center
Manage Party
Logout from the system
End
Activity diagram for voting management system
LOGIN ACTIVITY DIAGRAM
Start
Admin is registered
Admin Login ID and PassWord
Check Login ID/Password
NO
Invalid Login/PassWord
YES
Login to the system successfully
Set UserLevel and permissions
Access the internall functionalities according to permission
End
Login Activity diagram for voting management system


UML CLASS DIAGRAM
sammy mpungu | October 1, 2023
Counting Class
+ counting_id: int
+ counting_type: string
+ counting_desc: string
+ counting_name: string
+ counting_date: date
+ addCounting()
+ editCounting()
+ deleteCounting()
+ searchCounting()
Role Class
+ role_id: int
+ role_title:string
+ role_description: string
+ addRole()
+ editRole()
+ deleteRole()
+ searchRole()
+assignRole()
Vote Class
+ vote_id: int
+ vote_name: string
+ vote_desc: string
+ counting_name: string
+ vote_type: type
+ addVote()
+ editVote()
+ deleteVote()
+ searchVote()
Admin (User) Class
+ user_id: int
+ user_role_id: int
+ user_name: string
+ user_email: string
+ user_dob: date
+ user_address: string
+ addUser()
+ editUser()
+ deleteUser()
+ searchUser()
Permission Class
+ permission_id: int
+ permission_role_id: int
+ permission_tittle: string
+ permission_module: string
+ permission_desc: string
+ addPermission()
+ editPermission()
+ deletePermission()
+ searchPermission()
Voter List Class
+ Voter_list_id: int
+ Voter_List_name: string
+ Voter_List_desc: string
+ Voter_List_type: string
+ addVoterList()
+ editVoterList()
+ deleteVoterList()
+ searchVoterList()
Counting Class
+ counting_id: int
+ counting_type: string
+ counting_desc: string
+ counting_name: string
+ counting_date: date
+ addCounting()
+ editCounting()
+ deleteCounting()
+ searchCounting()
Party Class
+ party_id: int
+ party_name: string
+ party_desc: string
+ party_address: string
+ addParty()
+ editParty()
+ deleteParty()
+ searchParty()
Citizen Class
+ citizen_id: int
+ citizen_name: int
+ citizen_email: string
+ citizen_address: string
+ citizen_username: string
+ citizen_password: string
+ addCitizen()
+ editCitizen()
+ deleteCitizen()
+ searchCitizen()
UML class diagram for voting management system