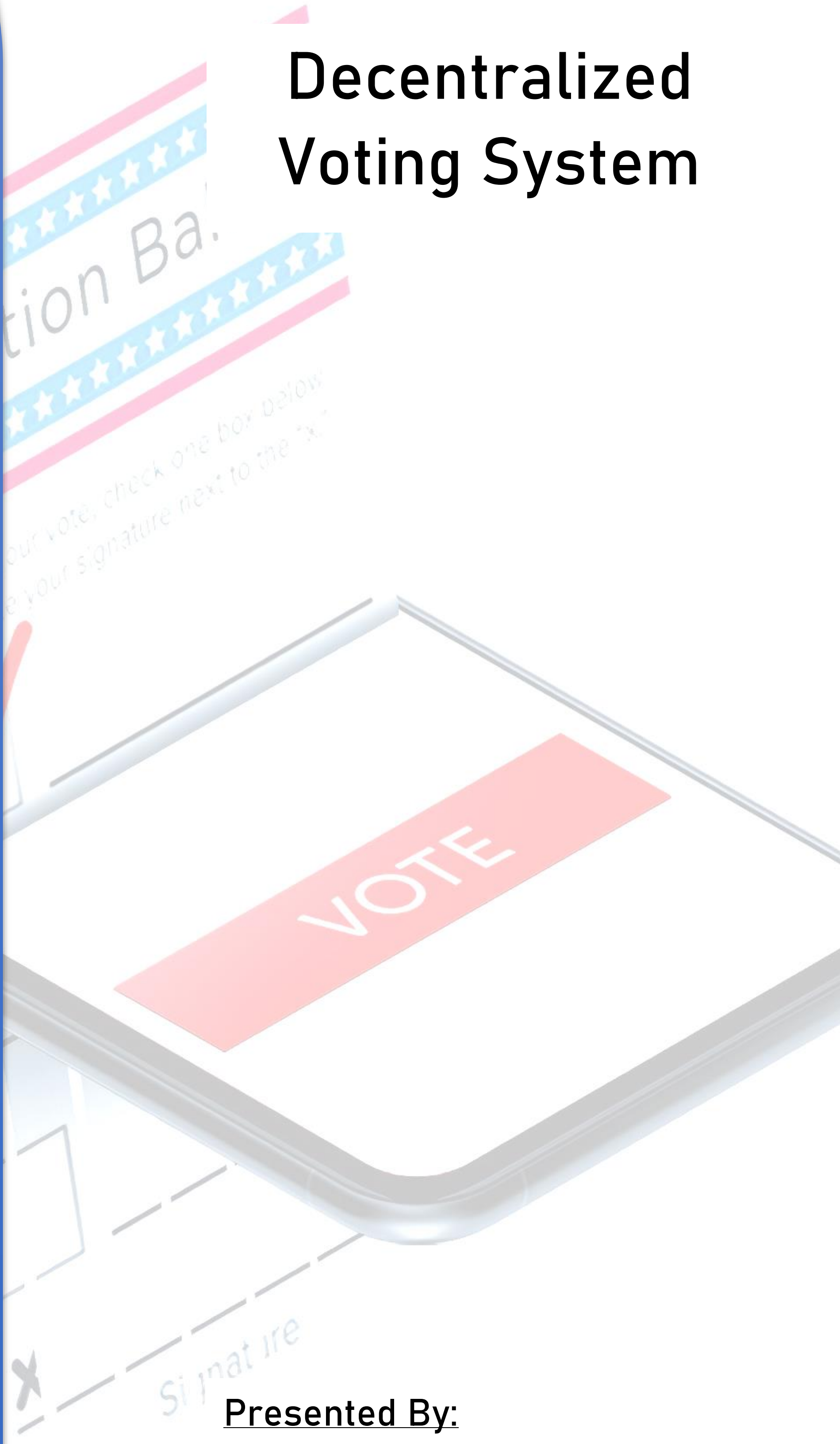
Decentralized Voting System



Presented By:

Team 2 - SANKALP KH

Election Ballot
To cast your vote, check one box below and write your signature next to the "x"
VOTE
Signature

ASSIGNMENT TITLE

DECENTRALISED E-VOTING SYSTEM

Submitted in partial fulfilment of the requirement of

PG DIPLOMA IN ADVANCED COMPUTING

BY

TEAM 2 – SANKALP, KH

Centre for Development of Advanced Computing

Kharghar/Juhu

Sept 2022

ABSTRACT

This Software Requirement System discusses about the study of implementation of decentralized e-voting system. In information security, computational trust is the generation of trusted authorities or user trust through cryptography. In centralised systems, security is typically based on the authenticated identity of external parties.

Rigid authentication mechanisms, such as Public Key Infrastructures (PKIs)[1] or Kerberos[2] have allowed this model to be extended to distributed systems within a few closely collaborating domains or within a single administrative domain. During recent years, computer science has moved from centralised systems to distributed computing. This evolution has several implications for security models, policies and mechanisms needed to protect users' information and resources in an increasingly interconnected computing infrastructure.

CHAPTER 1

1.1 INTRODUCTION:

Voting is the civic duty of the people in a democracy, unfortunately, the archaic methods of voting still haunt us in the 21st century. Long queues, inconvenient drives, and the perceived insignificance of a single vote deter many from participating. We need something that can provide the infrastructure for a voting system that is efficient, secure, mobile, and traceable. Thankfully, blockchain technology enables us to facilitate a voting infrastructure that can modernize voting the way we see fit.

1.2 DOCUMENT PURPOSE:

Online Voting System is a system which enables all citizens to cast their vote online. The purpose is to increase the voting percentage across the country, as in the present system people have to visit the booth to cast their vote and those people who live out of their home town are not able to cast vote during the elections. So due to this the voting percentage across the country is very less. Through this software those people who live out of their home town will also be able to cast their votes as this system is online and this software is on a blockchain platform so it’s secure, immutable and cannot be manipulated.

1.3 PRODUCT SCOPE:

In the 21st century the archaic methods of voting worry us, therefore, building a simple web application is not preferable for this purpose because the votes can be changed, the voting rules can be changed and the trust of the voters is lost. For these reasons we are revolutionizing the voting process. An e-voting dap is not only secure from corruption, but also provides strong resistance to hacking and other cybercrimes. It can be built within the given time frame and budget. The most important factor of this application is that not even the programmers can alter the votes once submitted by voters. It is easy to log into, feasible and easy to use, transparent, audible – the core problems of the prevailing voting systems.

Intended Audience and Document Overview:

This SRS is an easy guide for the developers (us), the professor who can act as our client and the project managers. It will help in building the application with the flow and also provide a clear vision of the application before we actually start making the application. The SRS has been divided into different chapters and further into sub-sections to make it easy for each audience to develop an understanding of the details and aims of the software. Chapter 1 is typically related for the client and the developers to understand the development stages of the application. It provides the general overview and puts on path on what do we have to achieve. Chapter 2 is related to the overall description of the application which provides the different interfaces, and the platforms to be used in app development. Hence it concerns to the developers and project managers but somehow the clit is also involved because the application developed should be compatible with their systems. Other remaining chapters relate to the development side which includes the details of requirements, the look pf the application (interfaces), safety and security constraints

Definitions, Acronyms and Abbreviations:

The SRS is designed in a way that it is comprehensive to whom it concerns. Here is the little detail of the few words used in the SRS
1. Dapp Decentralized application.
2. DLT Distributed Ledger Technology
3. Ledger an open, distributed file that can record transactions between two parties efficiently and in a verifiable and permanent way.
4. ETH ether currency.

Document Conventions:

The document strictly follows the IEEE standard format and no other formatting is done in the document.

References and Acknowledgments:

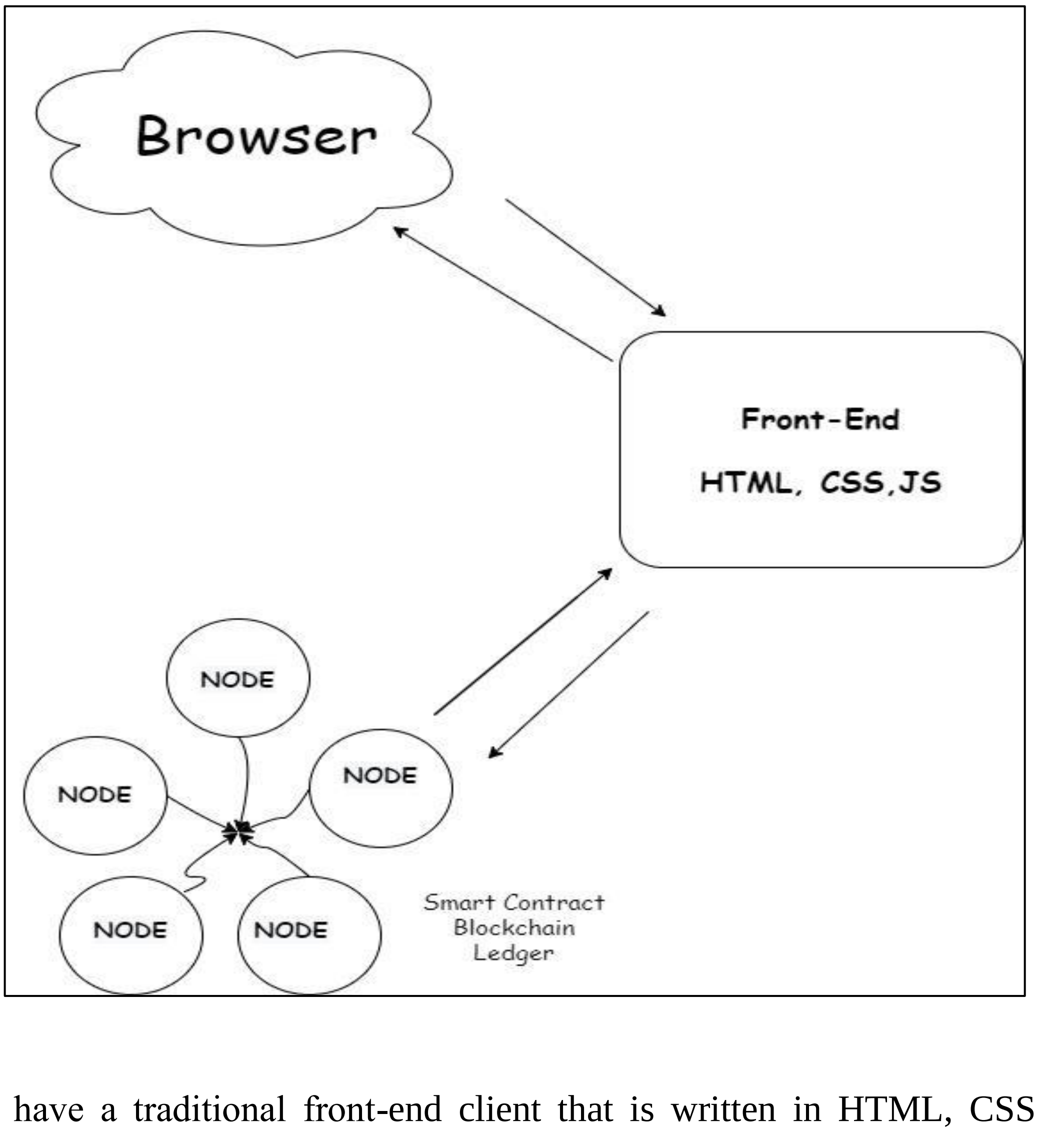
1. IEEE Standards Description: 830-1998
2. Writing Effective Use Cases - Alistair Cockburn (1998)
3. Software Engineering –Robert Pressman
4. Blockchain Database System Concepts by Mastering Etherium.

CHAPTER 2) OVERALL DESCRIPTION

2.1 Product Perspective:

The product is made with the perspective that it will replace the prevailing voting system, although it can be first be applied to small areas for testing and after being successful it might be implemented at a larger scale.

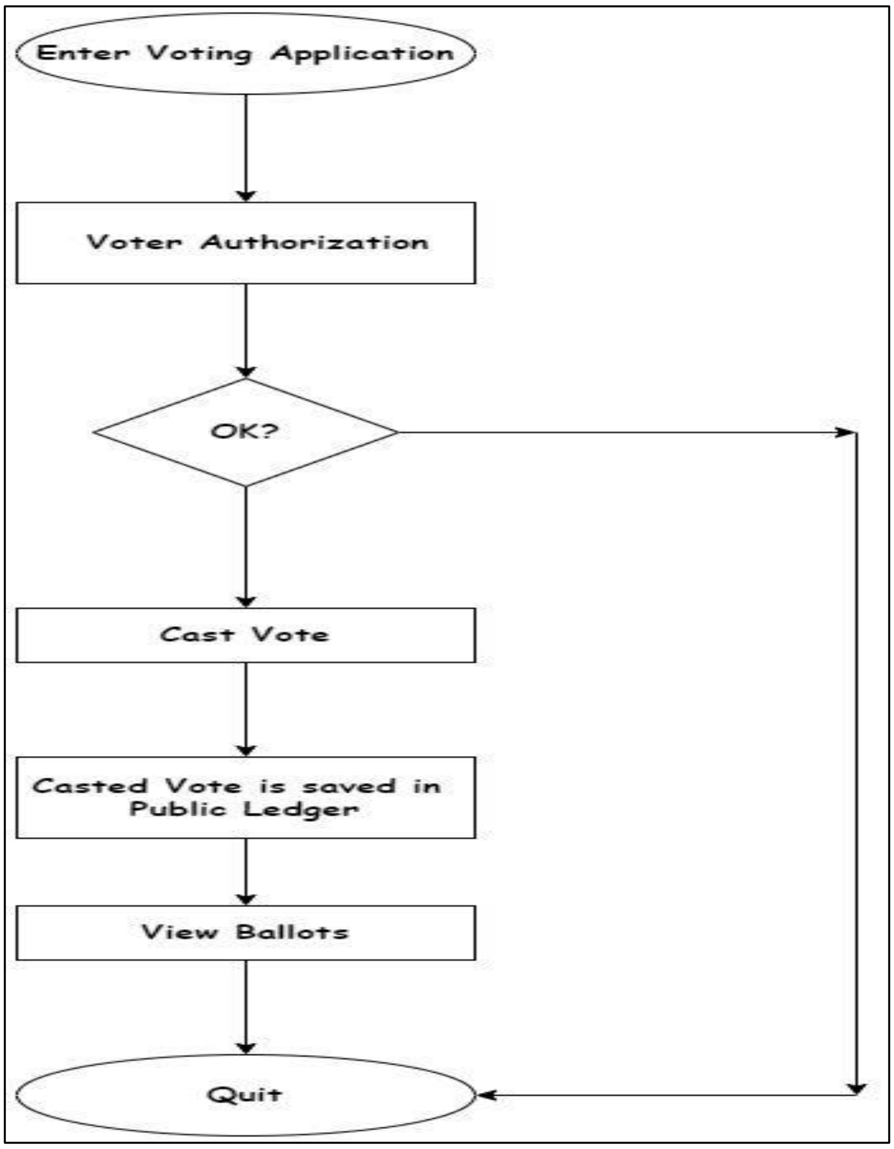
Now let’s take a quick look at the structure of the Dapp we’re building.



We’ll have a traditional front-end client that is written in HTML, CSS, and JavaScript. Instead of talking to a back-end server, this client will connect to a local Ethereum blockchain that we’ll install. We’ll code all the business logic about our dapp in an Election smart contract with the Solidity programming language. We’ll deploy this smart contract to our local Ethereum blockchain (database), and allow accounts to start voting.

2.2 Product Functionality:

From the voter's perspective, the system is used to help them cast their votes and after the elections are over, allow them to view the results, which are automatically posted on the same site after the election duration is over. The beneficial factor of using this application is that the voters have assurity that their votes will never get manipulated in any circumstances and they can ultimately be a part of fair elections.



2.3 Users and Characteristics:

The three main users of this system would be;

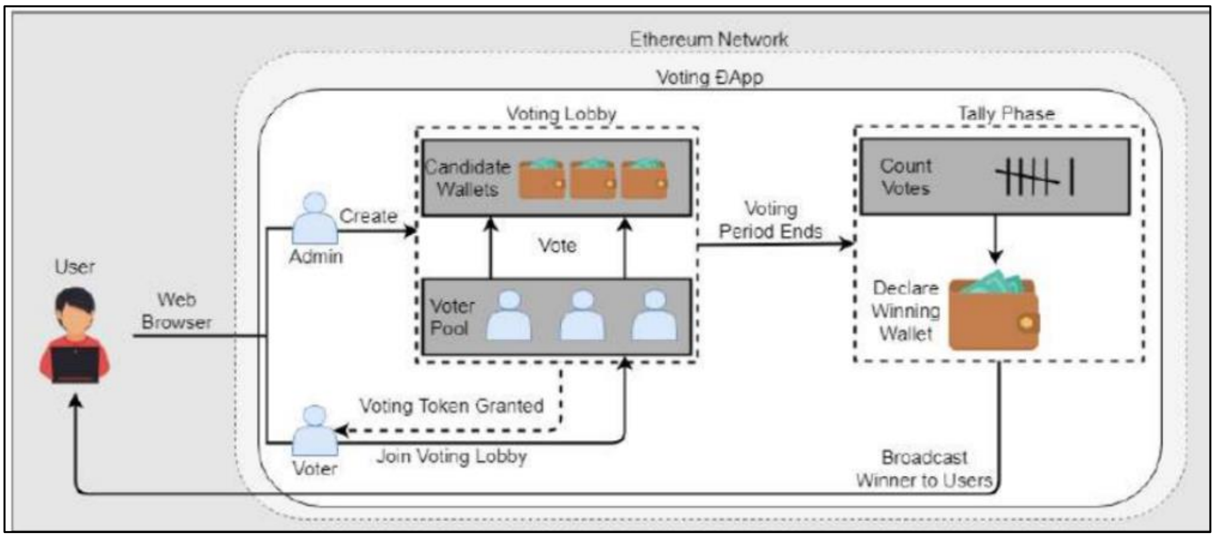
ADMINSTRATOR: Creates the voting lobby, sets lobby to public or private, adds candidates to lobby, and sets the date and time the voting period will end. Administrators can vote during the voting period.

VOTER: Joins lobby of choice with associated Meta Mask wallet, receives voting token upon registration, votes before time ends.

CANDIDATES: Added to lobby by administrator by wallet address, tokens are received by this address during voting phases, candidates can vote. However other stakeholders could be a maintenance person, a development team toprovide updated versions to meet the changing requirements with time and also to make the application compatible with existing common machines. A technical expert would be required to troubleshoot the bugs that might occur when the system and been deployed and implemented.

2.4 Operating Environment:

The application can be run on any computer (machine), since it is a web application, it requires an internet connection. The application is supported by both Windows and Linux. No other hardware component is required other than a computer to access the application.
The diagram below shows the major components of the overall system, subsystem interconnections, and external interface.



2.5 Design and Implementation Constraints:

The application has been designed for a large-scale public use therefore it was required to keep the design and implementation constrains as low as possible however the few of them could be:

- The customer's organization that is responsible to maintain the project must be familiar with how blockchain works.
- Since the app is build using Solidity programming language, to provide further updates the language is a major constraint.
- The budget could be a major constraint, for example if the customer wants more options and is low on budget then it becomes a major design constraint.
- Any changes in the design demanded by the client in latter stages of application development could bring great implementation troubles.
- GUI is in English only.
- User should have basic knowledge of computer.

2.6 Assumptions and Dependencies:

- The end user knows English and have basic knowledge of computer.
- Roles and tasks are pre-defined.
- The voting results will be managed and calculated by the Blockchain.
- Administrator is created in the system already.

CHAPTER 3) SPECIFIC REQUIREMENTS

3.1 External Interface Requirements:

3.1.1 User Interfaces:

- Voters: The citizens of the country who are eligible for casting vote.

- Register for Online Voting System : Those who already have voter id, they will register themselves for online voting system and they will use their voter id's their user's name and separate password will be used for secure authentication.

- Cast vote: The citizens will cast their votes for their favourite candidates online through a secure system.

- View own details: The voters will view their own details which they filled up at the time of their registrations.

3.1.2 Hardware Interfaces:

- The only interfaces are through a computer system.
- The operating system can be Windows/Linux or MAC.

3.1.3 Software Interfaces:

The poll server runs on http server that is enabled to handle server pages (eg. Truffle and Ganache. In order to run the setup software, the environment needs to have a NPM running on it). It uses a blockchain database to keep track of the polls, which it connects through standard.

3.1.4 Communications Interfaces:

- A router connected through an internet.
- Communication interfaces include HTTP server hosted on Ganache Truffle Suite.
- Ganache is communicating with Meta Mask.
- Meta Mask is communicating with Ethereum Blockchain.

3.2 Functional Requirements:

3.2.1 Authorize Actor:

This use case is the starting point for any interaction with the information system. It is a general use case specialized for organizers, i.e., "Authorise organiser" and users i.e., "Authorise user".

3.2.2 Manage Electors:

The fundamental assumption of this system uses case is that almost all citizens above a certain age should be able to participate in the election procedure. It is practically infeasible for election organisers to manually enter all electors into the system. The system should thus be able to import an electronic list of electors.

3.2.3 Manage Parties:

This system use case is purely operational and is not directly linked to any business use case of our voting model.

3.2.4 Manage Candidates:

This system will manage the candidates participating for the elections.

3.2.5 Preview Ballots:

This use case provides the ability to anyone to preview the electronic ballots for any election.

3.2.6 Provide Party Info:

This use case can be available either before the voting procedure or during the actual voting.

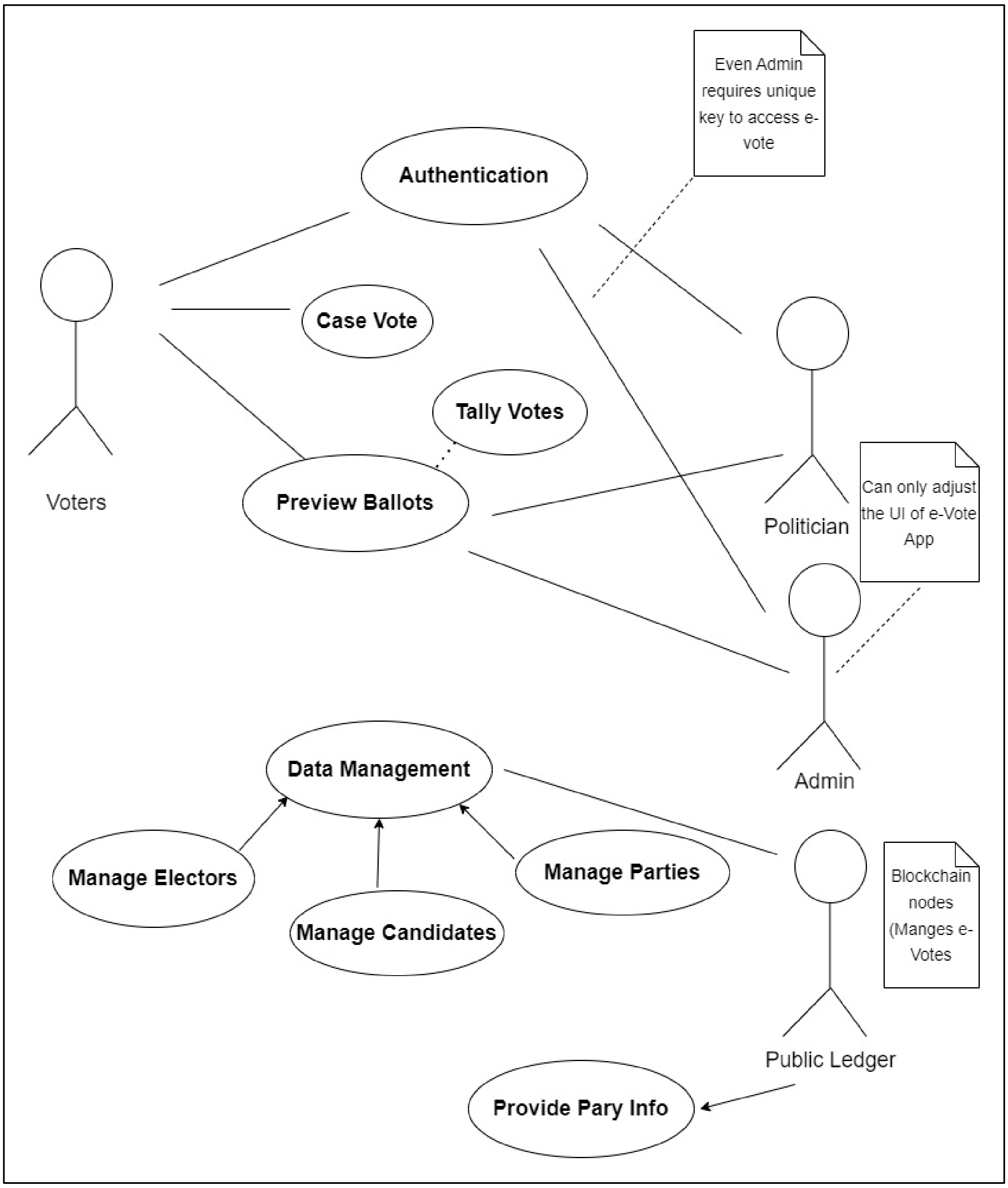
3.2.7 Cast Vote:

This is an important use case as it is the crux of our decentralized E-voting system as it poses the major part of the problems with regards to security.

3.2.8 Tally Votes:

This system uses case deals with the final tally calculation.

3.3 Behaviour Requirements:



CHAPTER 4: OTHER NON-FUNCTIONAL REQUIREMENTS

4.1 Performance Requirements:

1. Response time of e-vote should be less than 5 seconds most of the time. Response time refers to the time that the user should wait for before getting a response from the system after querying it.
2. E-vote shall show no visible deterioration.
3. E-voting system server will be down for 2 hours after every 12 hours of continuous run-time for maintenance and shall only be up and running during voting periods.
4. Votes will be updated on to the platform within 10 seconds of entering and conforming the vote and a conformation shall be provided within the next minute.
5. E-vote shall support up to a million voters at a time.

4.2 Safety and Security Requirements:

Safety:

1. Votes must only be placed once by each voter.
2. Votes may not be altered in any way, shape or form.
3. Voting rules are to be immutable, even to the admin of E-vote.

Security:

Data is decentralized

Since our E-voting system is based on blockchain, it provides an exceptional level of security since the technology is decentralized in nature and therefore does not rely on one central point of control. It is a digital ledger of transactions with every device having a complete copy of the data. A lack of a single authority makes the E-voting system fairer and considerably more secure. Instead of depending on a central authority to securely transact with other users, our E-voting system utilizes innovative consensus protocols across a network of nodes, to validate votes and record them in a manner that is incorruptible. Since the data is saved on multiple devices, it is extremely secured even if one or two devices malfunction.

Unfeasibly hard to hack
Since the data is decentralized and distributed ledgers across peer-to-peer networks are continuously updated and kept in sync. Each 'node' is connected to all the other 'nodes' before and after it. While hackers can break into traditional networks and find all the data in a single repository and exfiltrate it or corrupt it, the blockchain makes this unfeasibly hard.

4.3 Software Quality Attributes:

4.3.1 Availability:

E-voting system server will be down for 2 hours after every 12 hours of continuous run-time for maintenance and shall only be up and running during voting periods.

4.3.2 Correctness:

E-vote will perform as per the previously mentioned functional and non-functional requirements correctly and accurately.4.3.3 Reusability Yes, the component of our E-voting system can be used for other block-chain applications (Dapps).

4.3.4 Testability:

E-vote has low testability since testing requires admin to purchase Etherium gas through which a unique key is to be generated and then the admin can access E-vote and perform tests.

4.3.5 Portability:

E-vote is highly portable since it is an online application, therefore any device can access E-vote using a browser and an internet connection.

4.3.6 Reliability:

Since E-vote is based on Blockchain, it is highly reliable because full copies of the block- chain ledger are maintained by all active nodes. Thus, if one node goes offline, the ledger is still readily available to all other participants in the network and thereof lacks a single point of failure.

4.3.7 Usability:

Our E-vote online app is very user-friendly, with a simple and easy to use user-interface so that our voters may easily place and check the votes and have a satisfactory level experience with E-vote.