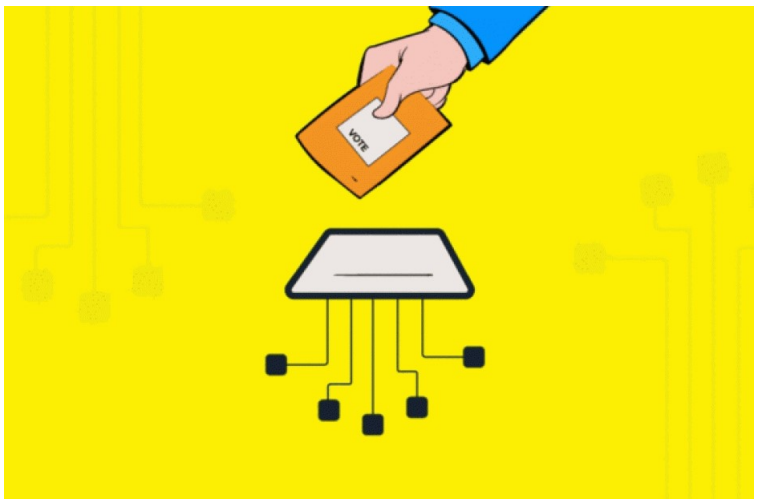
Department of Computer Science & Engineering
The University of Texas at Arlington

Detailed Design Specification
CSE 4317: Senior Design II
Fall 2022



The ChainVoters
Blockchain Voting

Ahmed Harun
Abdulla Sakallah
Seth Rodgers
Kelechi Egubuty

Revision History

Revision	Date	Author(s)	Description
0.1	10.05.2022	AS	document creation
0.2	10.18.2022	AS	completed application settings and poll results layers
1.0	10.20.2022	KC	completed onboarding and front end layers
1.0	10.20.2022	AH	completed blockchain and database layers
1.0	10.20.2022	SR	completed vote and poll manager layers

CONTENTS

List of Figures

List of Tables

1 Introduction

This project is a voting application that ensures votes transparency and privacy by using the Blockchain's functionality. The project is designed to match the needs of a business or government agency that must ensure voters' anonymity and transparency to help make decisions based on poll results.

2 System Overview

The voting application consists of eight main layers that communicate across the application to provide the key requirements that a user can perform. The frontend layer controls how the data is displayed or prompted and builds the interface for the user, using the onboarding, poll manager, vote, application settings, and poll results layers, whereas the blockchain and the application database are responsible to store and verify the validity of the data.

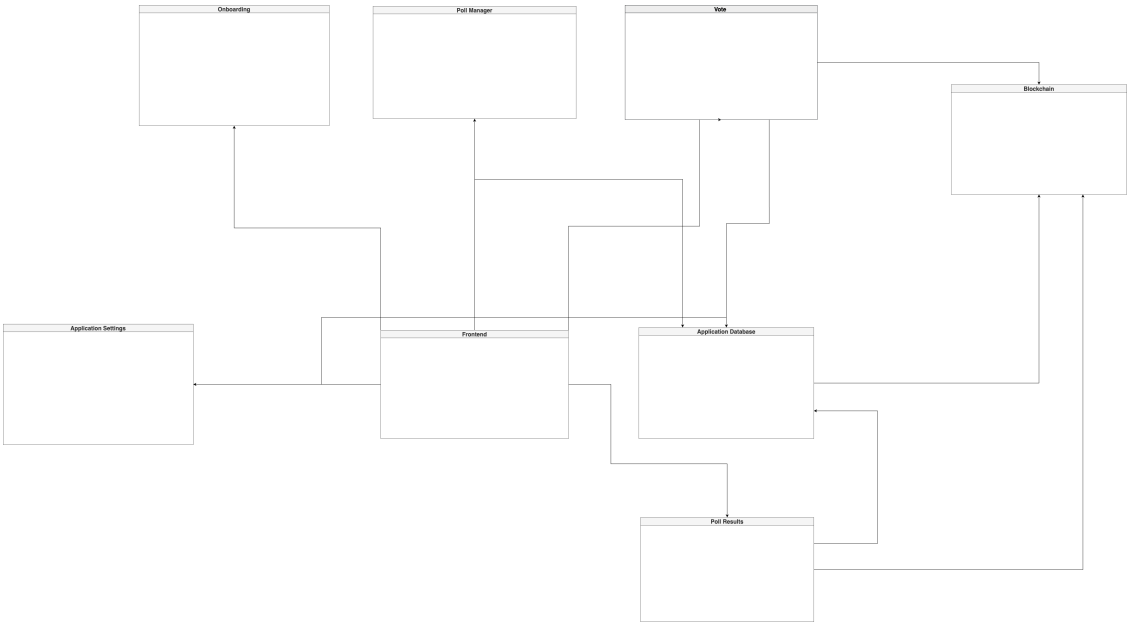


Figure 1: System Architecture

3 Poll Results Layer

The poll results layer is responsible to display the user's poll information by having them encapsulated as a collector. If a user wishes to retrieve or update values to them, they can. This layer also performs a database query that gets validated by blockchain access to display the poll's results.

3.1 Layer Hardware

mySQL server which contains information about each poll and user.

3.2 Layer Operating System

Can run on Windows 10 or Windows 11, mac OS, Linux, IOS, and Android.

3.3 Layer Software Dependencies

- JavaScript framework React v 18.2.0
- JavaScript framework Hardhat v 2.9.9
- mySQL v 2.18.1
- npm v 8.19.1
- node js v 16.13.2

3.4 Poll Information Collector

This subsystem is responsible for collecting the poll's information entered by the user in a form and saved as an object to be stored in the mySQL server.

Figure 2: Poll information Subsystem diagram

3.4.1 Subsystem Hardware

The mySQL server is needed to store the data entered by the user.

3.4.2 Subsystem Operating System

No operating system is needed.

3.4.3 Subsystem Software Dependencies

- JavaScript framework React v 18.2.0.
- mySQL v 2.18.1
- npm v 8.19.1
- node js v 16.13.2

3.4.4 Subsystem Programming Languages

- JavaScript ES6
- CSS v 4.15
- HTML5
- SQL

3.4.5 Subsystem Data Structures

Information about each poll is stored as an object that contains poll attributes and is sent to mySQL database.

3.4.6 Subsystem Data Processing

Prompting the user to enter the data using a given form.

3.5 Database Query

This subsystem handles the information that was stored inside the object from the form that the user filled out to each specific poll by storing it inside the database or retrieving it to verify the poll information.

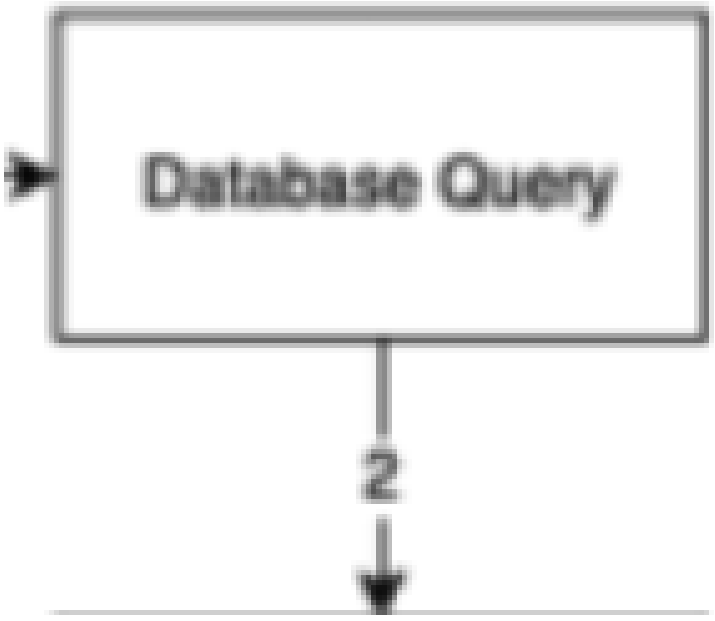


Figure 3: Database Query Subsystem Diagram

3.5.1 Subsystem Hardware

The mySQL server is needed to perform CRUD operations per user or system request.

3.5.2 Subsystem Operating System

No operating system is needed.

3.5.3 Subsystem Software Dependencies

- JavaScript framework React v 18.2.0.
- mySQL v 2.18.1
- npm v 8.19.1
- node js v 16.13.2

3.5.4 Subsystem Programming Languages

- JavaScript ES6
- SQL

3.5.5 Subsystem Data Structures

CRUD operations are performed using queries and stored inside objects to be able to perform actions on this data or display it to the user.

3.5.6 Subsystem Data Processing

CRUD operations are performed to match the functionality of the operation given or set by the user.

3.6 Blockchain Access

This subsystem handles the information that was retrieved by the database query and verifies that each vote is valid.

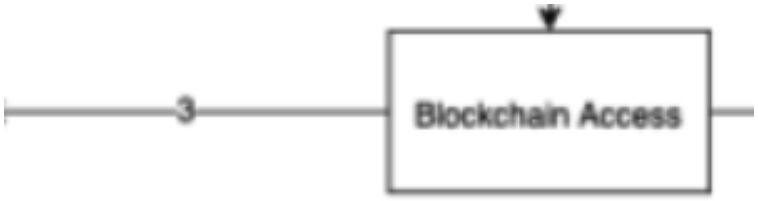


Figure 4: Blockchain Access Subsystem Diagram

3.6.1 Subsystem Hardware

The mySQL server is needed to retrieve the poll information.

3.6.2 Subsystem Operating System

No operating system is needed.

3.6.3 Subsystem Software Dependencies

- JavaScript framework React v 18.2.0.
- mySQL v 2.18.1
- npm v 8.19.1
- node js v 16.13.2
- JavaScript framework Hardhat v 2.9.9

3.6.4 Subsystem Programming Languages

- JavaScript ES6
- Solidity v 0.8.4
- SQL

3.6.5 Subsystem Data Structures

The data is not being stored but it is using a list of poll objects that are retrieved from the database.

3.6.6 Subsystem Data Processing

The smart contract checks for each poll object in the list of poll objects if they are valid or not.

3.7 Display Results

This subsystem displays the results of each validated poll to the user after the poll was stored in the database and verified on the blockchain.

Figure 5: Display Results Subsystem Diagram

3.7.1 Subsystem Hardware

The mySQL server is needed to perform CRUD operations per user or system request.

3.7.2 Subsystem Operating System

No operating system is needed.

3.7.3 Subsystem Software Dependencies

- JavaScript framework React v 18.2.0.
- mySQL v 2.18.1
- npm v 8.19.1
- node js v 16.13.2

3.7.4 Subsystem Programming Languages

- JavaScript ES6
- HTML5
- CSS v 4.15
- SQL

3.7.5 Subsystem Data Structures

CRUD operations are performed using queries and stored inside objects to be able to perform actions on this data or display it to the user.

3.7.6 Subsystem Data Processing

CRUD operations are performed to match the functionality of the operation given or set by the user.

4 Poll Manager Layer

The poll manager layer is responsible for the creation and deletion of polls. The user can input commands to create or delete polls as they wish. Depending on the command, the application will update the database for the corresponding action taken.

4.1 Layer Hardware

Node JS Frontend

4.2 Layer Operating System

Can run on Windows 10 or Windows 11, mac OS, Linux, IOS, and Android.

4.3 Layer Software Dependencies

- JavaScript framework React V 18.2.0.
- Firebase v 9.
- mySQL v 2.18.1
- npm v 8.19.1
- node js v 16.13.2

4.4 User Command

This subsystem handles the commands that are given by the user to perform the expected functionalities of these commands.

Figure 6: User Command subsystem diagram

4.4.1 Subsystem Hardware

An interface is needed to be able to perform these tasks by using a computer or a smartphone.

4.4.2 Subsystem Operating System

No operating system is needed.

4.4.3 Subsystem Software Dependencies

- JavaScript framework React v 18.2.0.
- npm v 8.19.1
- node js v 16.13.2

4.4.4 Subsystem Programming Languages

- JavaScript ES6
- HTML5
- CSS v 4.15

4.4.5 Subsystem Data Structures

The user command doesn't get stored inside any data structure; however, depending on the functionality it has, it will have its anonymous function that is responsible to handle the command.

4.4.6 Subsystem Data Processing

The user clicks any option that they would like to select from the list of options in the poll manager section and the information and command will be sent to the specific subsystem to do the expected task.

4.5 Poll Creation

This subsystem handles the creation of polls in the application when the user inputs this command.

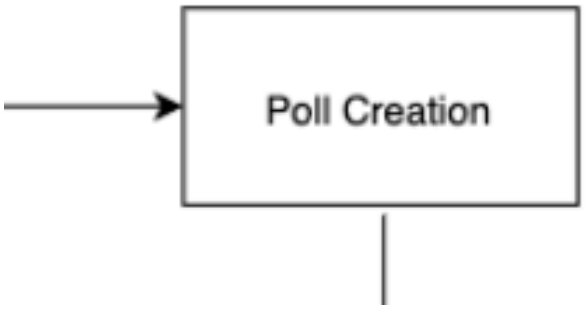


Figure 7: Poll Creation subsystem diagram

4.5.1 Subsystem Hardware

Interacting with the SQL Server

4.5.2 Subsystem Operating System

No operating system is needed.

4.5.3 Subsystem Software Dependencies

- JavaScript framework React v 18.2.0.
- Firebase v 9.
- npm v 8.19.1
- node js v 16.13.2

4.5.4 Subsystem Programming Languages

- JavaScript ES6
- HTML5
- CSS v 4.15

4.5.5 Subsystem Data Structures

No advanced data structures will be used, just the flat data schema that holds the information on the poll being created.

4.5.6 Subsystem Data Processing

After a user clicks on 'create poll', the data will be stored based on the inputs given by the user. These inputs will be processed from the given input fields to describe the poll and it's features, the data will be processed and sent to the SQL server.

4.6 Poll Deletion

This subsystem handles the deletion of polls in the application when the user inputs this command.

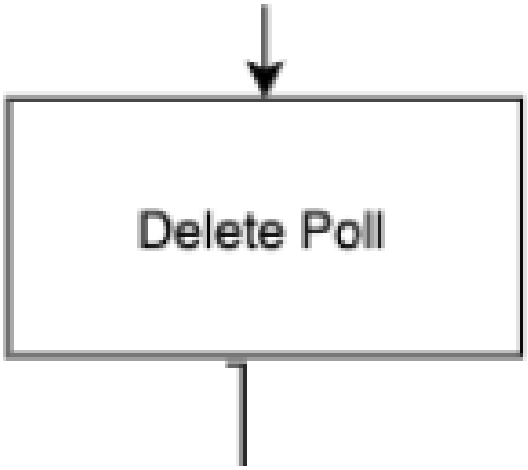


Figure 8: Delete Poll subsystem diagram

4.6.1 Subsystem Hardware

Interacting with the SQL Server

4.6.2 Subsystem Operating System

No operating system is needed.

4.6.3 Subsystem Software Dependencies

- JavaScript framework React v 18.2.0.
- Firebase v 9.
- npm v 8.19.1
- node js v 16.13.2

4.6.4 Subsystem Programming Languages

- JavaScript ES6
- HTML5
- CSS v 4.15

4.6.5 Subsystem Data Structures

No advanced data structures will be used, just storing the name of the poll to be deleted as a string.

4.6.6 Subsystem Data Processing

After a user clicks on 'delete poll', the data will be stored based on the inputs given by the user. These inputs will be processed from the given input fields to determine which poll gets deleted, and this data will be sent to the SQL server to delete the poll.

5 Vote Layer

The vote layer is responsible for handling and processing vote inputs from the users. The system handles vote validation when a user attempts to cast a vote, and once the vote is validated, the system also handles registering the vote.

5.1 Layer Hardware

Node JS Frontend in combination with the SQL Server, and also the user's smart wallet.

5.2 Layer Operating System

Can run on Windows 10 or Windows 11, mac OS, Linux, IOS, and Android.

5.3 Layer Software Dependencies

- JavaScript framework React v 18.2.0.
- Firebase v 9.
- mySQL v 2.18.1
- npm v 8.19.1
- node js v 16.13.2

5.4 Vote Validation

When a user casts a vote, this is the system that validates that the user is valid and the vote is valid for the poll they wish to participate in.

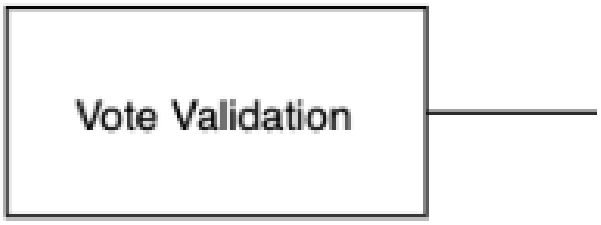


Figure 9: Vote Validation subsystem diagram

5.4.1 Subsystem Hardware

An interface is needed to be able to perform these tasks by using a computer or a smartphone.

5.4.2 Subsystem Operating System

No operating system is needed.

5.4.3 Subsystem Software Dependencies

- JavaScript framework React v 18.2.0.
- npm v 8.19.1
- node js v 16.13.2

5.4.4 Subsystem Programming Languages

- JavaScript ES6
- HTML5
- CSS v 4.15

5.4.5 Subsystem Data Structures

No complex data structure used, the data will be stored in a flat schema including the user's info and their vote.

5.4.6 Subsystem Data Processing

The data is gathered from the input fields on the page and processed and compared with data from the SQL server to validate the vote.

5.5 Vote Registration

This subsystem handles the registration of votes once they are deemed valid.

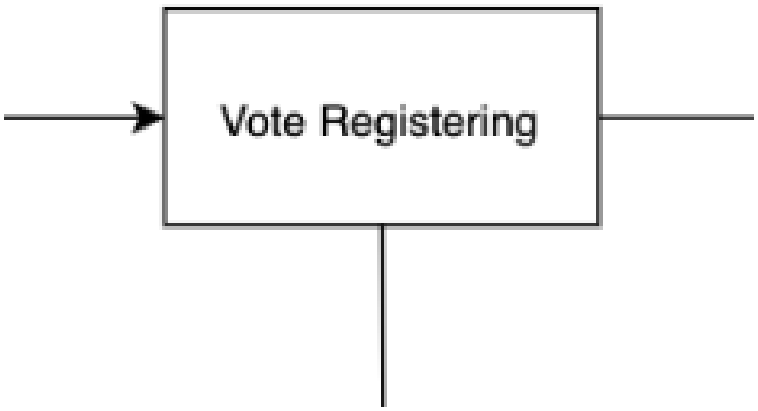


Figure 10: Vote registering subsystem diagram

5.5.1 Subsystem Hardware

Interacting with the SQL Server and the user's smart wallet through Ether.js

5.5.2 Subsystem Operating System

No operating system is needed.

5.5.3 Subsystem Software Dependencies

- JavaScript framework React v 18.2.0.
- Firebase v 9.
- npm v 8.19.1
- node js v 16.13.2

5.5.4 Subsystem Programming Languages

- JavaScript ES6
- HTML5
- CSS v 4.15

5.5.5 Subsystem Data Structures

No complex data structure used, the data will be stored in a flat schema including the user's info and their vote.

5.5.6 Subsystem Data Processing

The data is gathered from the input fields on the page and processed and sent the SQL server and also the user's smart wallet to register the vote on the blockchain.

6 Application Settings Layer

The application settings layer is responsible to make the application more interactive with the user. A user can access any of the sign-out, FAQ, delete account, or/and notifications functionalities. The layer allows users to sign out or delete their account, once they use these two functionalities they have to either log in/register again to access other functionalities in the layer.

6.1 Layer Hardware

mySQL server which contains information about each poll and user.

6.2 Layer Operating System

Can run on Windows 10 or Windows 11, mac OS, Linux, IOS, and Android.

6.3 Layer Software Dependencies

- JavaScript framework React V 18.2.0.
- Firebase V 9.
- mySQL V 2.18.1
- npm v 8.19.1
- node js v 16.13.2

6.4 User Command

This subsystem handles the commands that are given by the user to perform the expected functionalities of these commands.

Figure 11: User Command subsystem diagram

6.4.1 Subsystem Hardware

An interface is needed to be able to perform these tasks by using a computer or a smartphone.

6.4.2 Subsystem Operating System

No operating system is needed.

6.4.3 Subsystem Software Dependencies

- JavaScript framework React v 18.2.0.
- npm v 8.19.1
- node js v 16.13.2

6.4.4 Subsystem Programming Languages

- JavaScript ES6
- HTML5
- CSS v 4.15

6.4.5 Subsystem Data Structures

The user command doesn't get stored inside any data structure; however, depending on the functionality it has, it will have its anonymous function that is responsible to handle the command.

6.4.6 Subsystem Data Processing

The user clicks any option that they would like to select from the list of options in the application settings and the information and command will be sent to the specific subsystem to do the expected task.

6.5 FAQ

This subsystem handle displays the answers to frequent questions asked by the users.

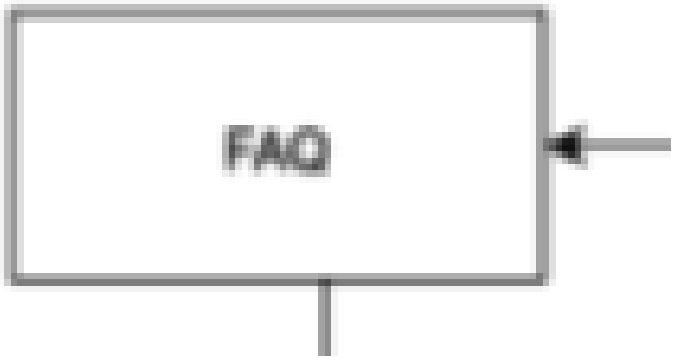


Figure 12: FAQ subsystem diagram

6.5.1 Subsystem Hardware

No subsystem hardware is needed.

6.5.2 Subsystem Operating System

No operating system is needed.

6.5.3 Subsystem Software Dependencies

- JavaScript framework React v 18.2.0.
- npm v 8.19.1
- node js v 16.13.2

6.5.4 Subsystem Programming Languages

- JavaScript ES6
- HTML5
- CSS v 4.15

6.5.5 Subsystem Data Structures

No data structure is used.

6.5.6 Subsystem Data Processing

Information is displayed on the screen and styled using HTML/CSS/JavaScript.

6.6 Sign Out

This subsystem handles logging the user out of the application, by setting the login token to false.

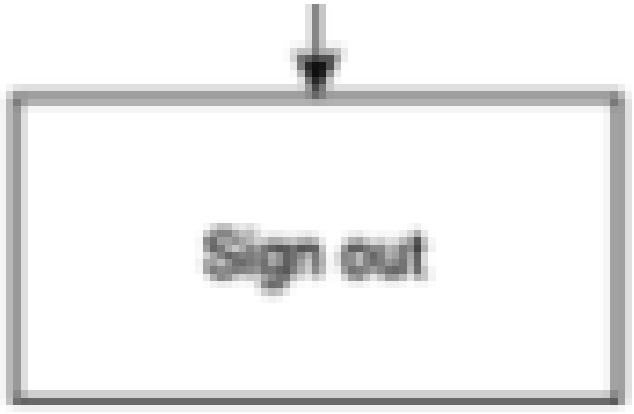


Figure 13: Sign Out subsystem diagram

6.6.1 Subsystem Hardware

Google Firebase server is needed to switch the login token to false.

6.6.2 Subsystem Operating System

No operating system is needed.

6.6.3 Subsystem Software Dependencies

- JavaScript framework React v 18.2.0.
- Firebase v 9.
- npm v 8.19.1
- node js v 16.13.2

6.6.4 Subsystem Programming Languages

- JavaScript ES6
- HTML5
- CSS v 4.15

6.6.5 Subsystem Data Structures

a boolean token contains the sign-out status for a user.

6.6.6 Subsystem Data Processing

After a user clicks on the sign-out option, the user will log out from their account and will not have any privileges to see which data they had stored in their accounts unless they sign back in.

6.7 Delete Account

This subsystem handles account deletion operation which makes the user's data no longer available or accessible by any user with any privileges.

Figure 14: User Command subsystem diagram

6.7.1 Subsystem Hardware

The mySQL server is needed to perform account the deletion operation.

6.7.2 Subsystem Operating System

No operating system is needed.

6.7.3 Subsystem Software Dependencies

- JavaScript framework React v 18.2.0.
- mySQL V 2.18.1
- npm v 8.19.1
- node js v 16.13.2

6.7.4 Subsystem Programming Languages

- JavaScript ES6
- HTML5
- CSS v 4.15
- SQL

6.7.5 Subsystem Data Structures

Account data is being stored inside an object to be compared with users in the database.

6.7.6 Subsystem Data Processing

Remove a row that matches a primary key for a user.

6.8 Notification

This subsystem handles the notifications about each vote that each user has participated in, to tell them when would the poll be expired after they vote.

Figure 15: Notification subsystem diagram

6.8.1 Subsystem Hardware

mySQL server which contains information about each poll and user.

6.8.2 Subsystem Operating System

No operating system is needed.

6.8.3 Subsystem Software Dependencies

- JavaScript framework React v 18.2.0.
- npm v 8.19.1
- node js v 16.13.2
- mySQL V 2.18.1

6.8.4 Subsystem Programming Languages

- JavaScript ES6
- SQL

6.8.5 Subsystem Data Structures

The notification will not be stored in a data structure.

6.8.6 Subsystem Data Processing

A query will be made to retrieve the ending time of a poll and will be outputted to the screen as a notification.

7 Frontend Layer

The front-end layer is responsible for creating a connection between the user and the application by providing a friendly interface for the user to be able to perform operations in it. The first functionality from the front-end layer is that it takes the user inside the application from an on-boarding screen. Then it directs the user to a poll interface to choose and decide which operations they want to create/join and update the database by their selection. After that, users will be directed to a voting interface to validate and update their votes concurrently. The user will have access to their poll results to review their results from previous polls. The user will also be able to access their account settings if they want to sign out or edit their information. The application router connects all these functionalities together by abstracted pages.

7.1 Layer Hardware

Web app created with Node Js

7.2 Layer Operating System

- Windows
- MAC OS
- IOS
- android

7.3 Layer Software Dependencies

- JavaScript framework React v 18.2.0
- JavaScript framework Hardhat v 2.9.9
- npm v 8.19.1
- node js v 16.13.2

7.4 User On-boarding

Here the user is introduced to the web app and is given instructions on how the app works as well as some other functionalities

7.4.1 Subsystem Hardware

Node JS

7.4.2 Subsystem Operating System

- Windows
- MAC OS
- IOS
- android

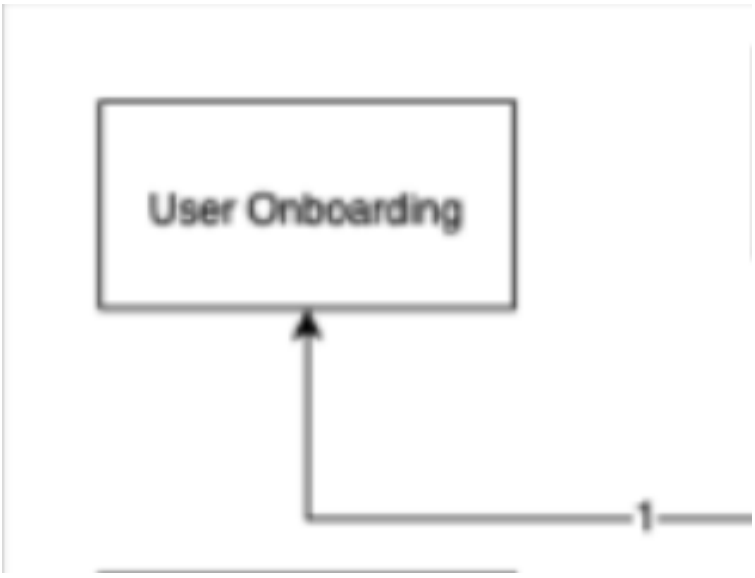


Figure 16: User On-boarding Subsystem diagram

7.4.3 Subsystem Software Dependencies

- JavaScript framework React v 18.2.0.
- npm v 8.19.1
- node js v 16.13.2

7.4.4 Subsystem Programming Languages

- JavaScript ES6
- CSS v 4.15
- HTML5

7.4.5 Subsystem Data Structures

N/A

7.4.6 Subsystem Data Processing

Prompting users to click on specific buttons on the app to show how it works

7.5 Poll Interface

This subsystem handles the creation of polls based on parameters provided by the user

7.5.1 Subsystem Hardware

MySQL server, Node JS

7.5.2 Subsystem Operating System

- Windows
- MAC OS

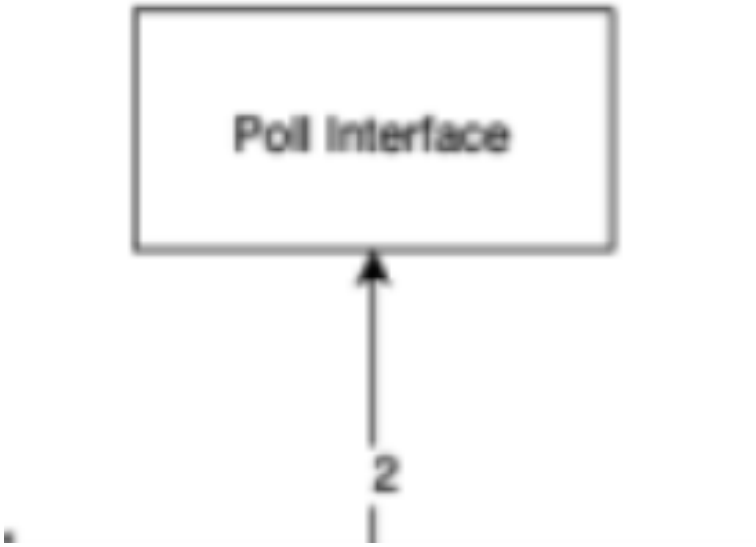


Figure 17: Poll Interface Subsystem Diagram

- IOS
- android

7.5.3 Subsystem Software Dependencies

- JavaScript framework React v 18.2.0.
- MySQL v 2.18.1
- npm v 8.19.1
- node js v 16.13.2

7.5.4 Subsystem Programming Languages

- HTML5
- CSS v 4.15
- JavaScript ES6
- SQL

7.5.5 Subsystem Data Structures

CRUD operations are performed using queries and stored inside objects to be able to perform actions on this data or display it to the user.

7.5.6 Subsystem Data Processing

CRUD operations are performed to create polls for the user

7.6 Voting Interface

This subsystem handles the information that was retrieved by the database query and verifies that each vote is valid.

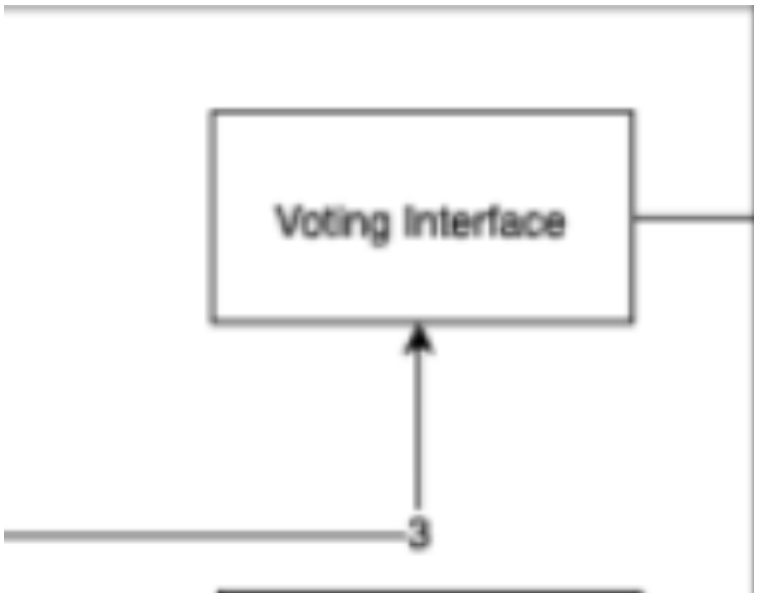


Figure 18: Voting Interface Subsystem Diagram

7.6.1 Subsystem Hardware

MySQL Server

7.6.2 Subsystem Operating System

- Windows
- MAC OS
- IOS
- android

7.6.3 Subsystem Software Dependencies

- JavaScript framework React v 18.2.0.
- MySQL v 2.18.1
- npm v 8.19.1
- node js v 16.13.2
- JavaScript framework Hardhat v 2.9.9

7.6.4 Subsystem Programming Languages

- JavaScript ES6
- SQL
- HTML5
- CSS v 4.15

7.6.5 Subsystem Data Structures

Data received from votes are saved as objects and stored in the SQL database

7.6.6 Subsystem Data Processing

The smart contract makes sure that a user has not previously cast a vote before allowing them to vote and sending the vote to the database

7.7 Poll Result Interface

This subsystem allows users to view the outcome of any polls they've created. It also allows users the ability to see who voted

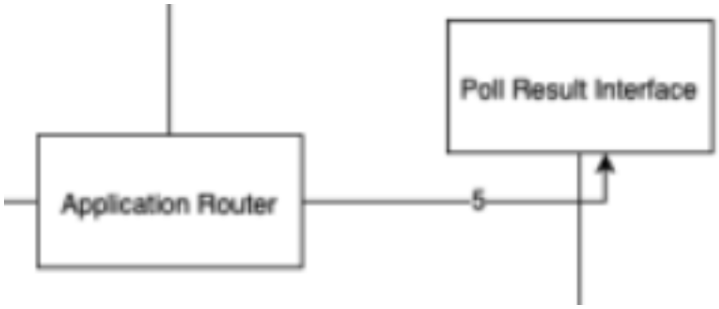


Figure 19: Poll Result Interface Subsystem Diagram

7.7.1 Subsystem Hardware

MySQL server Node JS

7.7.2 Subsystem Operating System

- Windows
- MAC OS
- IOS
- android

7.7.3 Subsystem Software Dependencies

- JavaScript framework React v 18.2.0.
- mySQL v 2.18.1
- npm v 8.19.1
- node js v 16.13.2

7.7.4 Subsystem Programming Languages

- JavaScript ES6
- HTML5
- CSS v 4.15
- SQL

7.7.5 Subsystem Data Structures

CRUD operations are performed using queries and stored inside objects to be able to perform actions on this data or display it to the user.

7.7.6 Subsystem Data Processing

CRUD operations are performed to match the functionality of the operation given or set by the user.

8 On boarding Layer

This layer handles the first interaction with the application. It does not consist of the voting functionality of the application; however, it is a major layer that needs to exist to verify the identity of the user. This layer is responsible for the authentication operations handled by the database and performs any necessary recoveries for accounts.

8.1 Layer Hardware

There is no hardware layer involved in this operation layer.

8.2 Layer Operating System

- Windows
- MAC OS
- IOS
- android

8.3 Layer Software Dependencies

- JavaScript framework React v 18.2.0
- JavaScript framework Hardhat v 2.9.9
- npm v 8.19.1
- node js v 16.13.2
- MySQL Server v 2.18.1
- Firebase v 9

8.4 Registration

The registration process is where users are able to create their accounts. The information entered is double-checked with the Firebase database to make sure it is unique.

Figure 20: User On-boarding Subsystem diagram

8.4.1 Subsystem Hardware

Firebase cloud servers.

8.4.2 Subsystem Operating System

- Windows
- MAC OS
- IOS
- android

8.4.3 Subsystem Software Dependencies

- JavaScript framework React v 18.2.0.
- npm v 8.19.1
- node js v 16.13.2
- Firebase v 9

8.4.4 Subsystem Programming Languages

- JavaScript ES6
- CSS v 4.15
- HTML5

8.4.5 Subsystem Data Structures

The data collected is cross-checked with a Firebase database before being saved in the database

8.4.6 Subsystem Data Processing

SHA1 hash function to generate hash values.

8.5 Login

This subsystem handles how users are able to enter the app

8.5.1 Subsystem Hardware

Firebase cloud servers.

8.5.2 Subsystem Operating System

- Windows
- MAC OS
- IOS
- android

8.5.3 Subsystem Software Dependencies

- JavaScript framework React v 18.2.0.
- MySQL v 2.18.1
- npm v 8.19.1
- node js v 16.13.2
- Firebase v 9

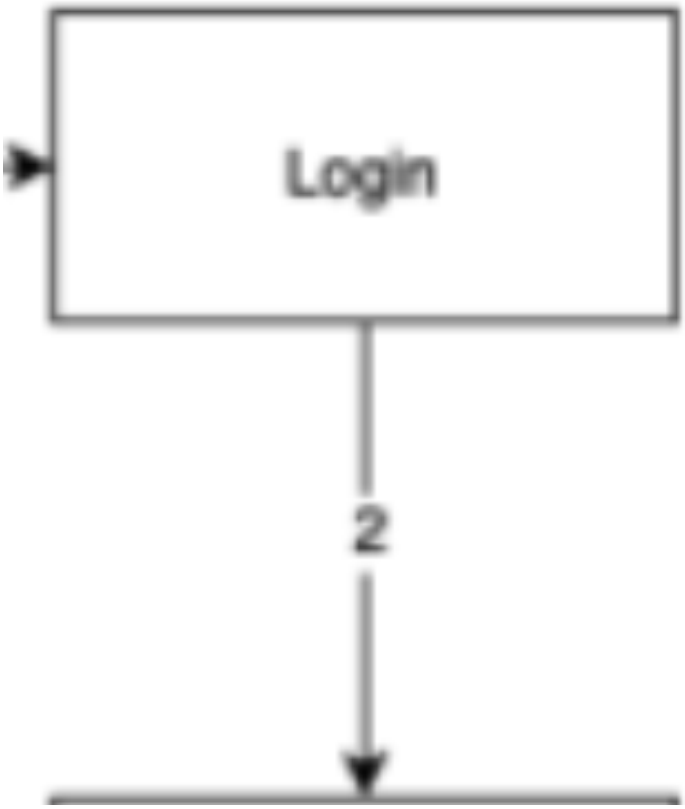


Figure 21: Poll Interface Subsystem Diagram

8.5.4 Subsystem Programming Languages

- HTML5
- CSS v 4.15
- JavaScript ES6
- SQL

8.5.5 Subsystem Data Structures

CRUD operations are performed to make sure the user has an existing account

8.5.6 Subsystem Data Processing

CRUD operations are performed to log in the user

8.6 Voting Interface

This subsystem handles the information that was retrieved by the database query and verifies that each vote is valid.

8.6.1 Subsystem Hardware

MySQL Server

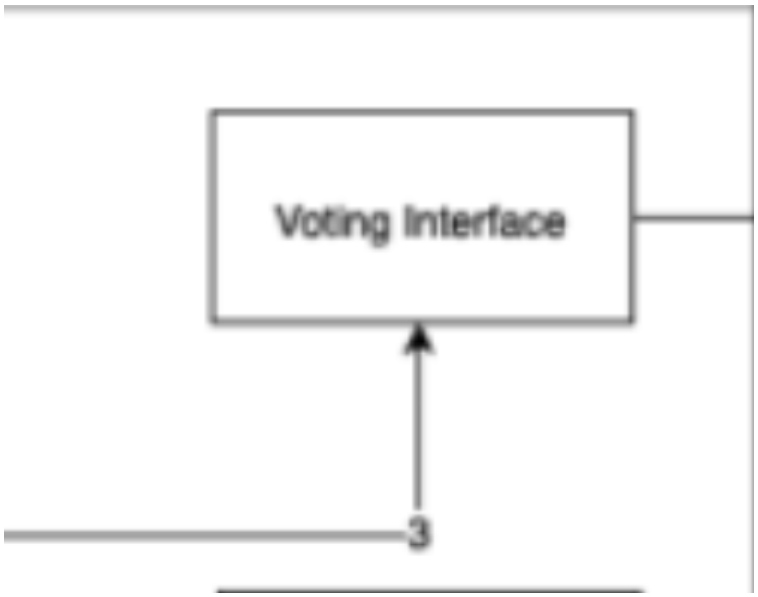


Figure 22: Voting Interface Subsystem Diagram

8.6.2 Subsystem Operating System

- Windows
- MAC OS
- IOS
- android

8.6.3 Subsystem Software Dependencies

- JavaScript framework React v 18.2.0.
- MySQL v 2.18.1
- npm v 8.19.1
- node js v 16.13.2
- JavaScript framework Hardhat v 2.9.9

8.6.4 Subsystem Programming Languages

- JavaScript ES6
- SQL
- HTML5
- CSS v 4.15

8.6.5 Subsystem Data Structures

Data received is cross-checked with an existing database, if it is a match the user can utilize the app

8.6.6 Subsystem Data Processing

SHA1 hash function to generate hash values.

8.7 Account Authentication

To guarantee the security of the web app, 2FA authentication is required after a successful login attempt. The user can use a phone number or email to pass the authentication

Figure 23: Poll Result Interface Subsystem Diagram

8.7.1 Subsystem Hardware

Firebase cloud servers.

8.7.2 Subsystem Operating System

- Windows
- MAC OS
- IOS
- android

8.7.3 Subsystem Software Dependencies

- JavaScript framework React v 18.2.0.
- MySQL v 2.18.1
- npm v 8.19.1
- node js v 16.13.2

8.7.4 Subsystem Programming Languages

- JavaScript ES6
- HTML5
- CSS v 4.15
- SQL

8.7.5 Subsystem Data Structures

After cross-checking that the login is successful, an OTP is sent to either the phone number or email of the user to verify the login attempt

8.7.6 Subsystem Data Processing

TOTP (Time-based one-time password)

8.8 Account Recovery

This subsystem provides users with the ability to recover their account in case of forgotten passwords, someone accessing it without consent, etc.

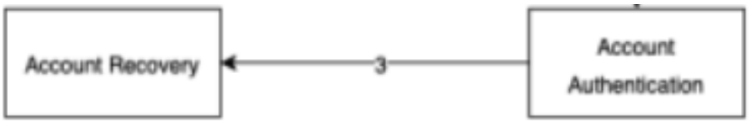


Figure 24: Poll Result Interface Subsystem Diagram

8.8.1 Subsystem Hardware

Firebase cloud servers.

8.8.2 Subsystem Operating System

- Windows
- MAC OS
- IOS
- android

8.8.3 Subsystem Software Dependencies

- JavaScript framework React v 18.2.0.
- MySQL v 2.18.1
- npm v 8.19.1
- node js v 16.13.2
- Firebase v 9

8.8.4 Subsystem Programming Languages

- JavaScript ES6
- HTML5
- CSS v 4.15
- SQL

8.8.5 Subsystem Data Structures

When requested, an OTP is sent to either the phone number or email for the user to complete the recovery attempt

8.8.6 Subsystem Data Processing

TOTP (Time-based one-time password)

9 Blockchain Layer

The blockchain layer is the heart of the application, it is critical for the security of the application and the transparency of the voting process. This layer will include smart contracts, libraries used for integrating blockchain technology into the application and the features that will be provided to the users as a result of having blockchain as the stabilizer of the application.

9.1 Layer Hardware

Due to this application being a web application using web frameworks and libraries, most layers will not include any hardware implementation, However there is one exception. blockchain functions in a distributed network where multiple computers perform calculations and verification, therefore several servers have to be already running in order for smart contracts to be uploaded into the blockchain. Fortunately this is already done for us so we have nothing to worry about.

9.2 Layer Operating System

- Windows
- MAC OS
- IOS
- android

9.3 Layer Software Dependencies

This layer has several dependencies that are required to be installed prior to the application being ran. This dependencies include:

- Ether v 5.7.2: which allows the creation of smart contracts
- Node.js v 16.13.2: allows for queries to be made to the blockchain and is the backbone of the application
- Hardhat v 2.12.0: provides numerous useful APIs for the developers which optimizes blockchain queries

9.4 Blockchain Addition

Blockchain addition is an essential component of the Blockchain system. It allows the user to create polls, cast votes and overall make the necessary changes in any given poll.

9.4.1 Subsystem Hardware

This subsystem requires no hardware with the exception of being able to communicate to the blockchain network in order to make transactions.

9.4.2 Subsystem Operating System

The subsystem has no operating system requirement that must be met, it works for all operating system and just requires internet connection

9.4.3 Subsystem Software Dependencies

The dependencies include for this subsystem include:

- Connection of the user's smart wallet to the application
- Sufficient ether funds to make the transaction

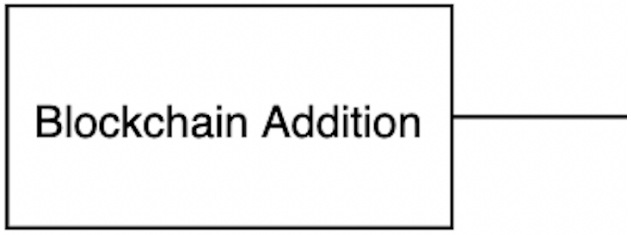


Figure 25: Blockchain Addition subsystem diagram

9.4.4 Subsystem Programming Languages

The programming language used for this component is Node.js for connecting the front end of the application to the blockchain and Solidity for uploading changes to the blockchain.

9.4.5 Subsystem Data Structures

The smart contract being created in the background is a class that contains multiple variables which correspond to the information about the poll and the registered users. The votes are essentially are stored in a key value pair where the key is the public key of the user and value is the vote they made in that specific poll.

9.4.6 Subsystem Data Processing

As soon as the user performs an action that changes the state of the blockchain, the application uses APIs provided by hardhat to propagate the data from the front end to the blockchain in order for the transaction to be processed properly. There is no specific data structure or unique algorithm being used besides having to detect when the user desires to add new information into the blockchain and processing that in the background.

9.5 Vote Storage

This component is responsible for storing the smart contracts and user votes in the blockchain which would allow fast retrieval of the necessary information whenever the user desires.

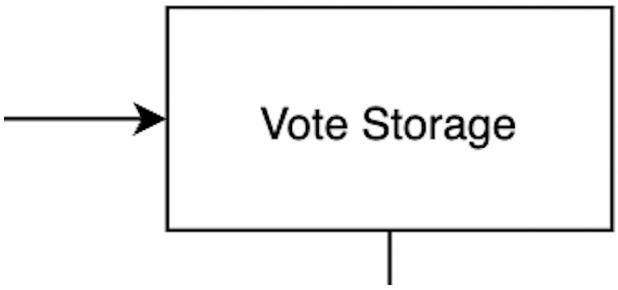


Figure 26: Vote Storage subsystem diagram

9.5.1 Subsystem Hardware

This subsystem requires no hardware with the exception of being able to communicate to the blockchain network in order to make transactions.

9.5.2 Subsystem Operating System

The subsystem has no operating system requirement that must be met, it works for all operating system and just requires internet connection

9.5.3 Subsystem Software Dependencies

This component only requires a connection of the user's smart wallet to the application

9.5.4 Subsystem Programming Languages

The programming language used for this component is Node.js for connecting the front end of the application to the blockchain and Solidity for uploading changes to the blockchain.

9.5.5 Subsystem Data Structures

The smart contract being created in the background is a class that contains multiple variables which correspond to the information about the poll and the registered users. The votes are essentially are stored in a key value pair where the key is the public key of the user and value is the vote they made in that specific poll. This allows the user to retrieve the data quickly without sifting through millions of records and it is a free transaction, meaning it does not cost any ether.

9.5.6 Subsystem Data Processing

As soon as the user decides to cast a vote, the application in the background will create a transaction which will result in the user's wallet to showcase the cost and time for the transaction. Once the user accepts the application will store the data into the blockchain using a key value pair which will ease retrieval in the next step

9.6 Vote Retrieval

Blockchain Retrieval is the last component which makes the entire system function. The other two components were responsible for creating polls and for storing votes, now we can retrieve votes from the blockchain.

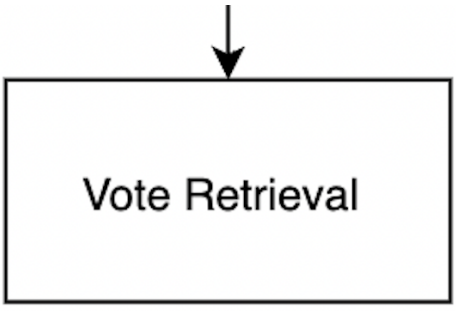


Figure 27: Vote Retrieval subsystem diagram

9.6.1 Subsystem Hardware

This subsystem requires no hardware with the exception of being able to communicate to the blockchain network in order to make transactions.

9.6.2 Subsystem Operating System

The subsystem has no operating system requirement that must be met, it works for all operating system and just requires internet connection

9.6.3 Subsystem Software Dependencies

This component only requires a connection of the user's smart wallet to the application

9.6.4 Subsystem Programming Languages

The programming language used for this component is Node.js for connecting the front end of the application to the blockchain and Solidity for uploading changes to the blockchain.

9.6.5 Subsystem Data Structures

The smart contract being created in the background is a class that contains multiple variables which correspond to the information about the poll and the registered users. The votes are essentially are stored in a key value pair where the key is the public key of the user and value is the vote they made in that specific poll.

9.6.6 Subsystem Data Processing

Whenever the user decides to retrieve poll voting information, the application will in the background retrieve the data from the key value pair in the voting smart contract and provide the necessary information to the user free of charge.

10 Application Database Layer

The database layer is an optimization layer that allows the application to be scalable and provides the users with a better overall user experience from the perspective of quicker loading time and faster queries. In addition, this layer maintains information not related to the polls and votes such as information related to user preferences and settings, which makes this multi-functional layer that is critical for the application.

10.1 Layer Hardware

There is no hardware requirement for this layer but it is required that there is a server running in order for the queries to be processed. Fortunately, this is very simple and it is always running as soon as the application starts.

10.2 Layer Operating System

The database can work with a variety of operating systems which is one of the benefits of using a database management system such as MySql.

10.3 Layer Software Dependencies

- node js v 16.13.2: for making the queries from the front-end possible
- mySQL v 2.18.1: The database management system required for the queries to be processed
- Axios v 1.1.3: the networking library that connects the front-end to the MySql server running in the back end

10.4 User relation

The user relation will maintain information that will be used to ensure a smooth experience for the user which will include wallet public key, username and password for the account. General information that will not expose the identity of the user.

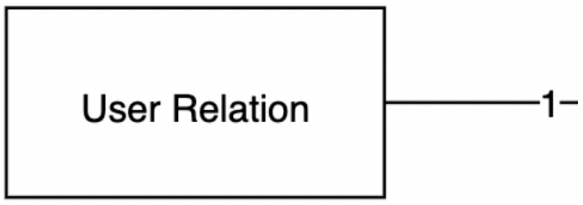


Figure 28: User Relation subsystem diagram

10.4.1 Subsystem Hardware

There is no hardware component to this subsystem. It is just a database relation that stores user information which is received from the front-end of the application.

10.4.2 Subsystem Operating System

No operating systems requirement as the DBMS works on all widely used operating systems.

10.4.3 Subsystem Software Dependencies

- node js v 16.13.2
- mySQL v 2.18.1
- Axios v 1.1.3

10.4.4 Subsystem Programming Languages

The programming languages used will include :

- node js v 16.13.2: for passing front-end information to the database in order for it to be stored
- SQL: For the retrieval and storage of user information from the back-end to the front-end

10.4.5 Subsystem Data Structures

The relation will include the following information in each query made to it:

- Public Key
- User name
- Pass word

10.4.6 Subsystem Data Processing

The information from the front-end and the back-end will essentially be passed back and forth using Axios which is a networking library that uses HTTP. The information will be decoupled when received or coupled and sent into the database for storage.

10.5 Registered Voters Relation

This relation will include information about the registered voters for each poll which will allow the poll administrator to get information such as who has voted already, who is registered for the poll, etc. This information is critical for each poll, in that it establishes a head count and it is mandatory for any voting system.

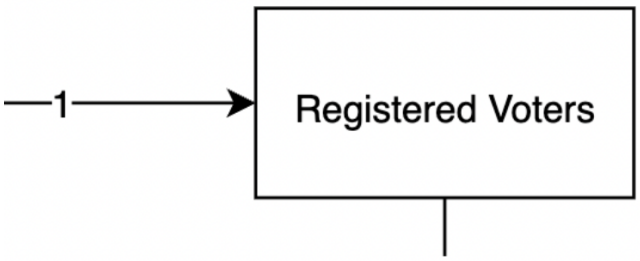


Figure 29: Registered Voters Relation subsystem diagram

10.5.1 Subsystem Hardware

There is no hardware component to this subsystem. It is just a database relation that stores voter information which is received from the front-end of the application.

10.5.2 Subsystem Operating System

No operating systems requirement as the DBMS works on all widely used operating systems.

10.5.3 Subsystem Software Dependencies

- node js v 16.13.2
- mySQL v 2.18.1
- Axios v 1.1.3

10.5.4 Subsystem Programming Languages

The programming languages used will include :

- node js v 16.13.2 : for passing front-end information to the database in order for it to be stored
- SQL : For the retrieval and storage of user information from the back-end to the front-end

10.5.5 Subsystem Data Structures

The relation will include the following information in each query made to it:

- Poll Address
- Public Key
- Registration Statuss

10.5.6 Subsystem Data Processing

The information from the front-end and the back-end will essentially be passed back and forth using Axios which is a networking library that uses HTTP. The information will be decoupled when received or coupled and sent into the database for storage.

10.6 Polls

This relation will contain information about all polls that have been created, for the purpose of quicker retrievals of important information when needed. As a result this relation will make possible a lot of convenient operations such as allowing a user to find out the number of polls they have participated in or created and even read about extra details pertaining to the poll.

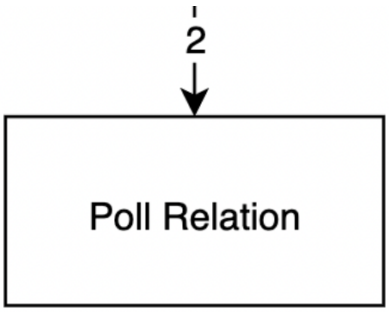


Figure 30: Poll Relation subsystem diagram

10.6.1 Subsystem Hardware

There is no hardware component to this subsystem. It is just a database relation that stores voter information which is received from the front-end of the application.

10.6.2 Subsystem Operating System

No operating systems requirement as the DBMS works on all widely used operating systems.

10.6.3 Subsystem Software Dependencies

- node js v 16.13.2
- mySQL v 2.18.1
- Axios v 1.1.3

10.6.4 Subsystem Programming Languages

The programming languages used will include :

- node js v 16.13.2: for passing front-end information to the database in order for it to be stored
- SQL: For the retrieval and storage of user information from the back-end to the front-end

10.6.5 Subsystem Data Structures

The relation will include the following information in each query made to it:

- Poll Address
- Poll Name
- Poll Owner
- Poll Start Date
- Poll End Date
- Poll Status

10.6.6 Subsystem Data Processing

The information from the front-end and the back-end will essentially be passed back and forth using Axios which is a networking library that uses HTTP. The information will be decoupled when received or coupled and sent into the database for storage.

11 APPENDIX A

REFERENCES