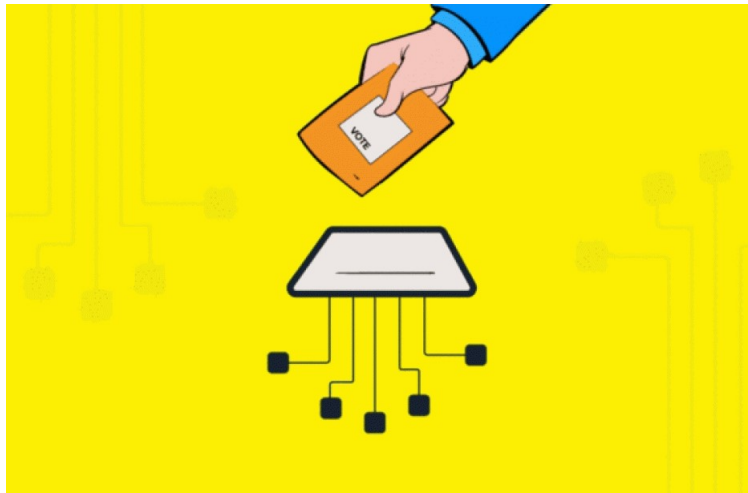


**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
THE UNIVERSITY OF TEXAS AT ARLINGTON**

**SYSTEM REQUIREMENTS SPECIFICATION
CSE 4316: SENIOR DESIGN I
SUMMER 2022**



**THE CHAINVOTERS
BLOCKCHAIN VOTING**

**AHMED HARUN
ABDULLA SAKALLAH
SETH RODGERS
KELECHI EGUBUTY**

REVISION HISTORY

Revision	Date	Author(s)	Description
0.1	7.15.2022	AH	document creation
0.2	7.23.2015	AH, AS, SR, KE	complete draft
1.0	7.31.2022	AH, AS, SR, KE	official release

CONTENTS

1	Product Concept	6
1.1	Purpose and Use	6
1.2	Intended Audience	6
2	Product Description	7
2.1	Features & Functions	7
2.2	External Inputs & Outputs	7
2.3	Product Interfaces	8
3	Customer Requirements	14
3.1	Landing Page	14
3.2	Login and Registration	14
3.3	Registration Confirmation Email	15
3.4	Poll Creation	16
3.5	Joining a Poll and Voting	16
3.6	Poll Invitation Methods	17
3.7	Poll Results	18
3.8	Poll Update	18
3.9	Poll Deletion	19
4	Packaging Requirements	20
4.1	Compilation on Git/GitHub	20
4.2	Availability in Google Play/App Store	20
5	Performance Requirements	21
5.1	Cross Platform	21
5.2	Real-Time Firebase Database	21
5.3	Real-Time Tracking	22
5.4	Responsitivity	22
6	Safety Requirements	23
6.1	Laboratory equipment lockout/tagout (LOTO) procedures	23
6.2	National Electric Code (NEC) wiring compliance	23
6.3	RIA robotic manipulator safety standards	23
6.4	Smart Contracts	24
6.5	Cloud Backup	24
6.6	Authentication	25
6.7	Data Protection	25
6.8	Authorization	26
7	Maintenance & Support Requirements	27
7.1	Obtain User-Data	27
7.2	Document Code	27
7.3	Blockchain Management	28
7.4	User Submitted Reports	28
7.5	Troubleshooting Guide	29

8 Other Requirements	30
8.1 Error Handling	30
8.2 Usability Standards	30
8.3 Accessibility Standards	31
8.4 Feasibility Standards	32
8.5 Extensibility	32
9 Future Items	34
9.1 Two-Factor Authentication	34
9.2 Extension of User Information	34
9.3 User Success Rate	35
9.4 Notifications to Remind Users to Participate	35
9.5 Poll Time Extension Request	36

LIST OF FIGURES

1	UML System Diagram	6
2	Landing Page	8
3	Registration and Verification Pages	9
4	Login Page	10
5	Password Recovery Page	11
6	Operational Pages	11
7	Operational Pages	12
8	Poll Creation Pages	12
9	Poll Voting Pages	13

1 PRODUCT CONCEPT

This section describes the purpose and usage of the Blockvote application and its intended audience. It will briefly describe the motivation for the application, the benefit it has for the stakeholders and the use cases for the application.

1.1 PURPOSE AND USE

The voting system currently is in a substandard condition where people are greatly dissatisfied. The existing voting environment greatly contributes to higher stress levels and other negative effects that can impact the voter's decision making ability. In this day and age where all of our needs are at our fingertips, the ability to cast our votes for important topics should be no different. Our application serves to address the aforementioned issues by allowing the voter to cast their vote from any location in the world safely and securely. It will be used for important rulings such as elections and corporate decisions that require anonymity and accuracy. Users will have various capabilities such as creating polls, joining polls, casting votes and even see a detailed evaluation of the results of a poll. All of these features will result in the creation of a voting system that allows the user to focus more on the voting matter rather than the inconveniences they will face in their journey to the ballot.

1.2 INTENDED AUDIENCE

The intended target audience is any entity whether governmental or business that requires the ability to make decisions while maintaining privacy of the participants. It is clear that the application has a niche market and it would not appeal to the general public. Attributes as such privacy, security and accuracy don't appeal to the typical consumer in the same way that it would entice governmental and business officials and that is the exact reason why our application has a narrow audience despite its effectiveness for all voting and decision making matters.

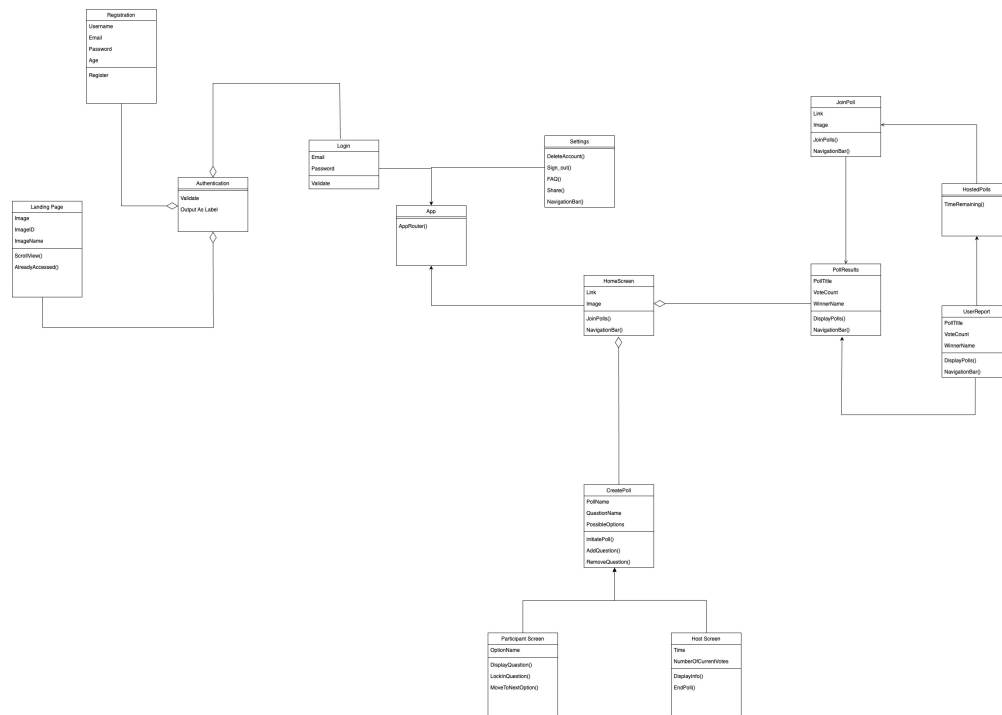


Figure 1: UML System Diagram

2 PRODUCT DESCRIPTION

This section is intended to provide the reader with an overview of the blockchain application's primary features and functionalities, its external inputs and numerous mock-ups will be provided to showcase the different interfaces of the application. In addition, the reader will be provided with details of the various interactions that will be provided to each user in the application.

2.1 FEATURES & FUNCTIONS

The application will contain many features that are intended to make the voting process as easy and robust as possible. One of the most important features of the application is the authentication feature. For an application that contains important information about the user's opinions and views on sensitive topics, it is clear that it should verify the user's identity. These authentication pages will consist of registration pages where the application will set up the user's account based on the provided information and the login page which is how the user gains access to their account. In addition, the application will provide poll administrators with certain privileges such as seeing the number of voters that have already voted, set constraints on the availability of the poll and also the ability to edit and even terminate the poll. Another important feature is the ability to join polls, for now the poll administrator will be given the options of distributing a link to the poll however they wish or they can upload a CSV/Excel file which contains information on the voters and the application will automatically send those voters a link. Accuracy is an important component of this application, our application ensures that each person is only given access to polls to which they are registered and limits their number of votes to just one per poll in order to maintain the integrity of the poll results. There are many other features which will be included in the application however the ones mentioned above are the most critical ones for the functionality of the application.

2.2 EXTERNAL INPUTS & OUTPUTS

Components	Input from user/external system	output for consumption
Registration	Full name, Email , Password, Email confirmation code	Creation of a brand new account for the newly registered user
Login	Email and Password	Opens up the user's account and presents them with the home screen
Account Recovery	Email	The user will be sent an email which will include a link that allows them to reset their password
Poll Creation	Poll Title, Poll start date, Poll end date, Poll question title, Poll choices,	A brand new poll will be created and it will become active at the designated start date and close on end date. The poll creator will also be given a link to the poll which will allow him to distribute the poll for the intended participants.
Poll Voting	Response	After the users submit their responses to the users they will immediately be given a badge to validate their vote. Later when the poll closes they will be able to see the results and a break down of the poll response.

2.3 PRODUCT INTERFACES

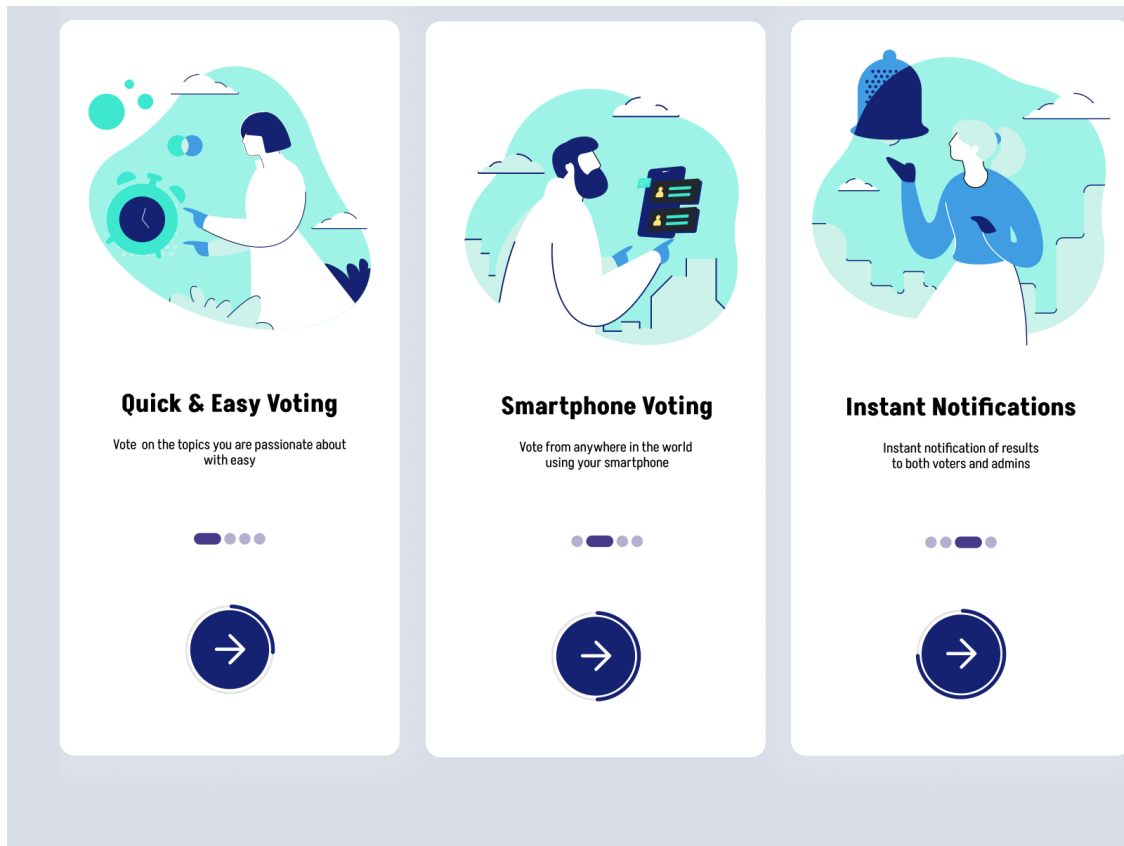


Figure 2: Landing Page

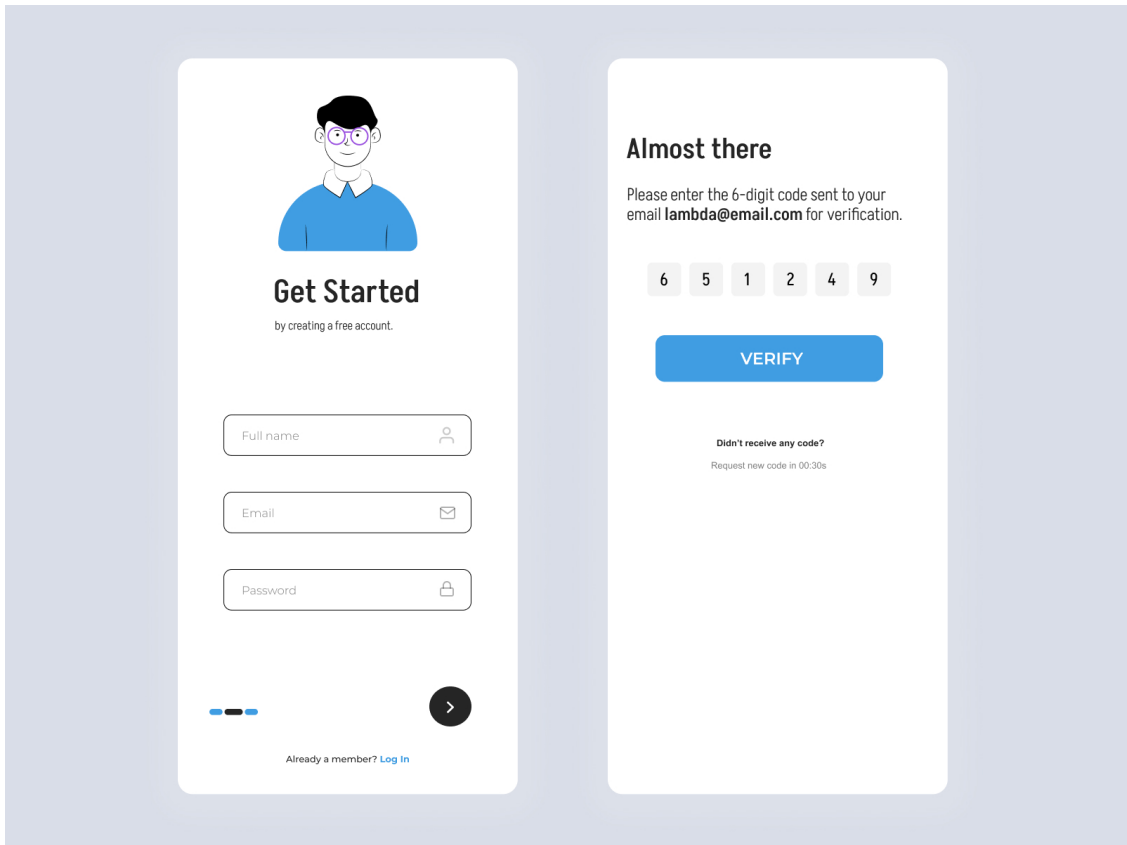


Figure 3: Registration and Verification Pages

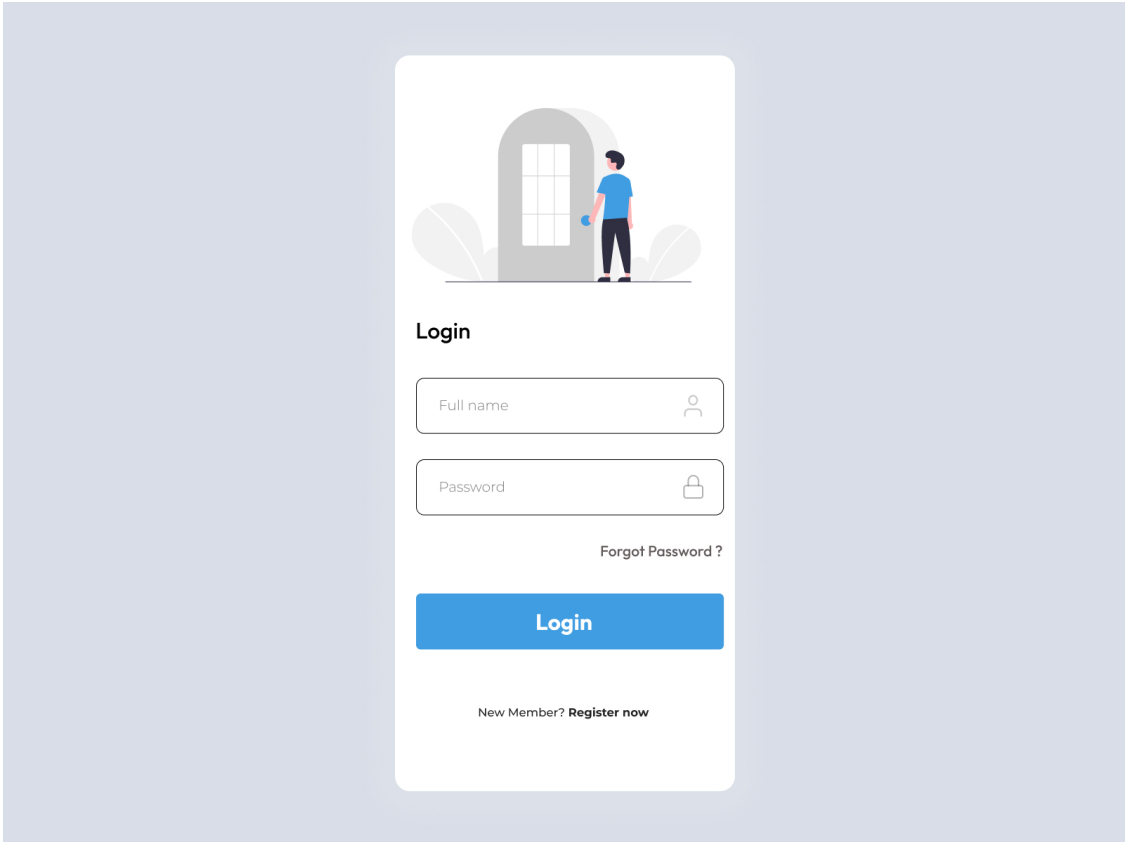


Figure 4: Login Page

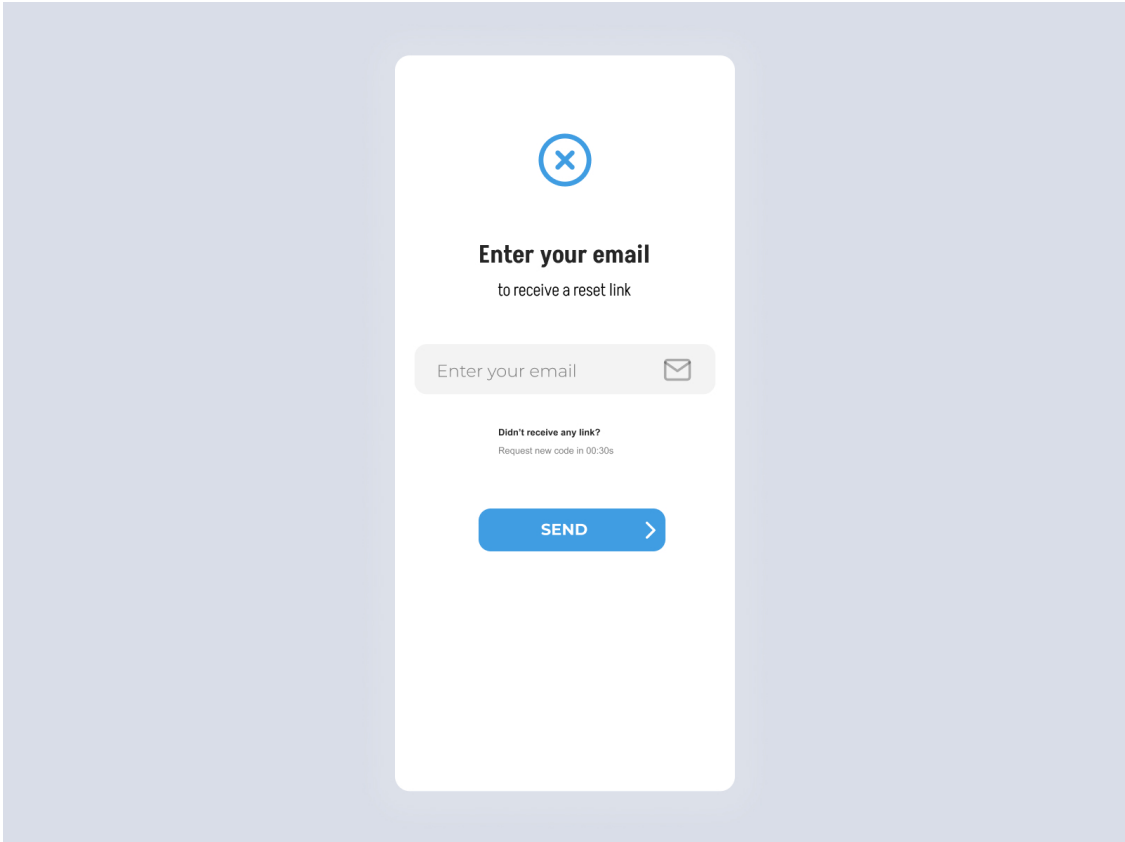


Figure 5: Password Recovery Page

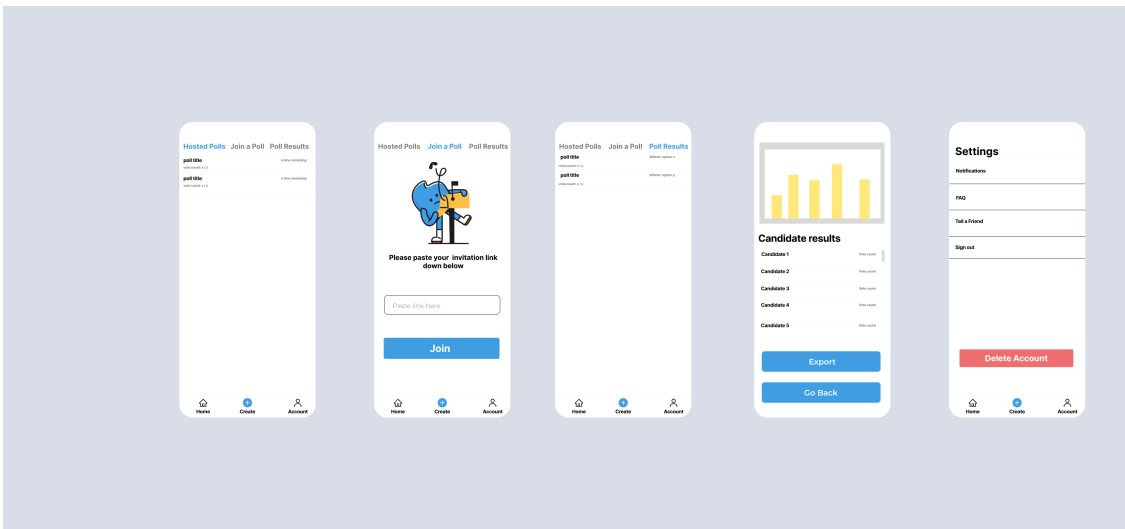


Figure 6: Operational Pages

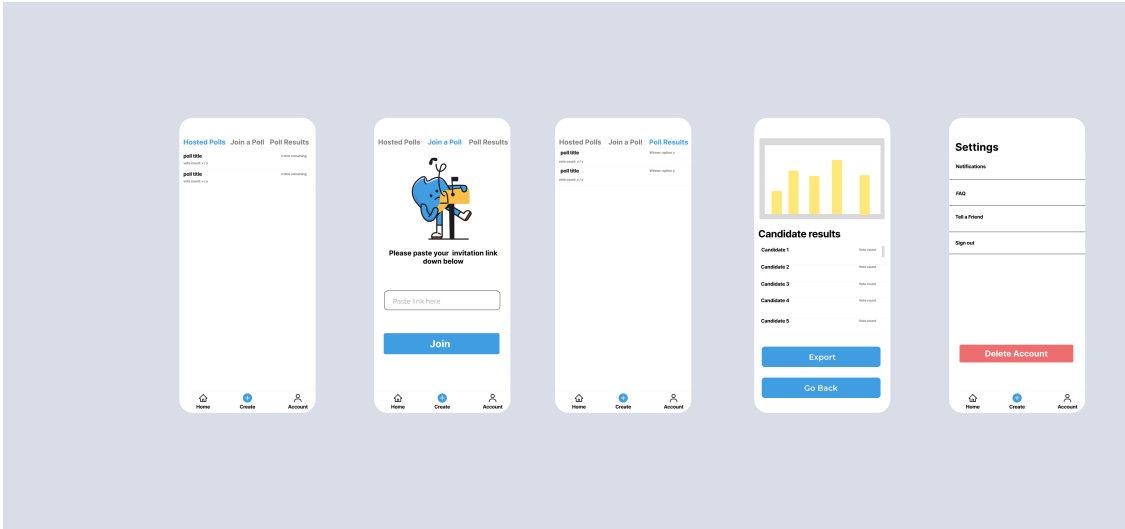


Figure 7: Operational Pages

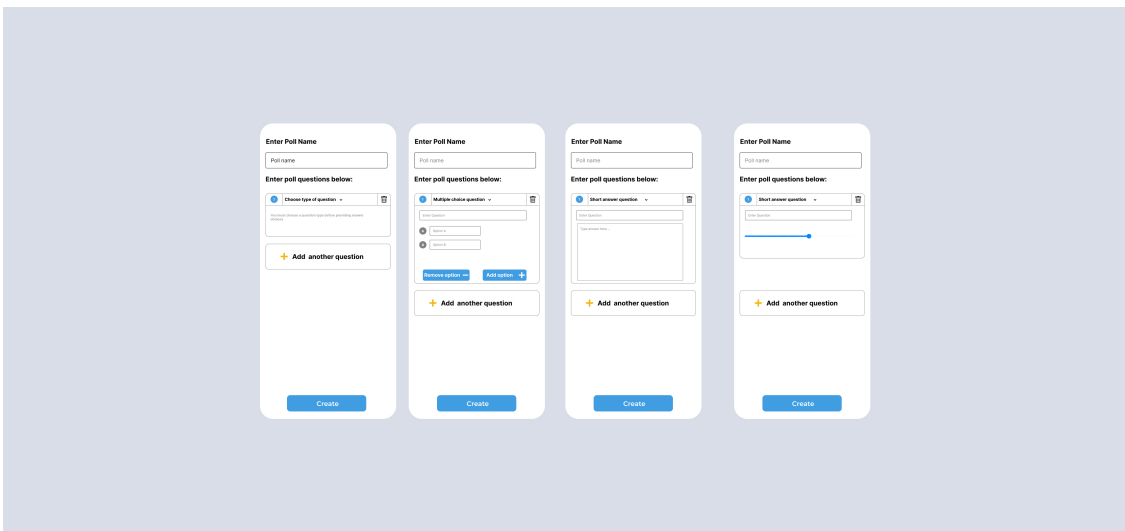


Figure 8: Poll Creation Pages

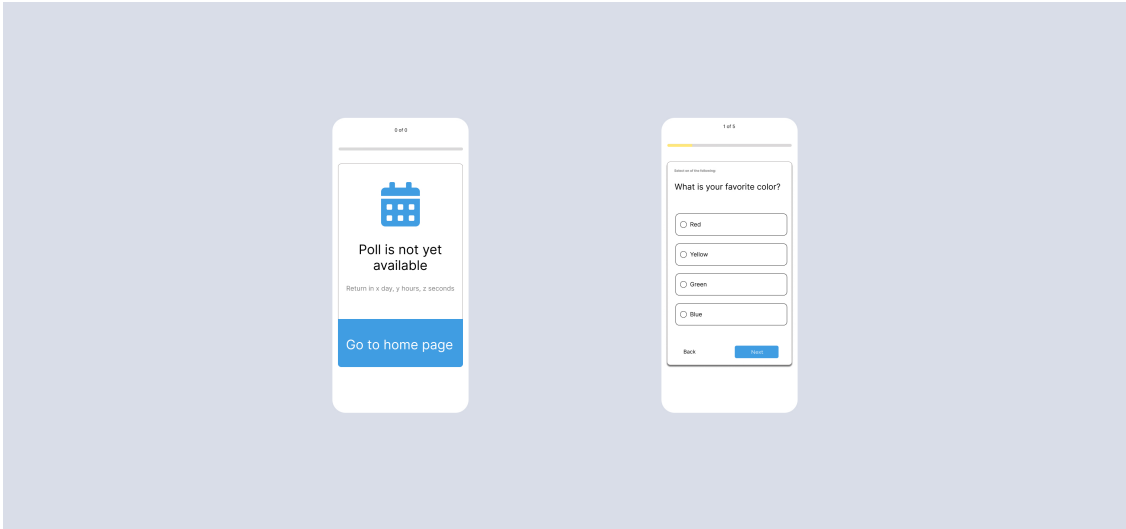


Figure 9: Poll Voting Pages

3 CUSTOMER REQUIREMENTS

The following requirements will showcase all of the necessary functionalities and features for the Block-Vote application. These functionalities and features will range from basic aesthetic requirements to critical functionalities that are necessary for the application to operate properly. By the end of this section it will be clear to the reader what components are necessary for the application and what the final product should look like.

3.1 LANDING PAGE

3.1.1 DESCRIPTION

- Quick overview of what the application does
- Some benefits of using the application as opposed to voting in the traditional manner

3.1.2 SOURCE

The origin of this feature is from a discussion that the team members had together and agreed upon the fact that this feature is critical to the functionality of the application.

3.1.3 CONSTRAINTS

- There must be an image describing the benefit being illustrated
- A title must accompany the illustration and be placed below the illustration
- Lastly a brief sentence describing the title should be placed below the title

3.1.4 STANDARDS

- A standard the team members came up with is that the landing page must be a single page which has a carousel banner that allows the user to scroll through the different product descriptions
- The color used must be navy blue (#152272) and bright teal(#CCF2EA)

3.1.5 PRIORITY

This is a high priority to requirement due to the uniqueness of the application. Most people assume blockchain technology is only for cryptocurrency so a page describing the usage of blockchain in the context of electronic voting is important.

3.2 LOGIN AND REGISTRATION

3.2.1 DESCRIPTION

The user must register for the app in order to either create or join a poll. The reasoning behind this requirement comes from the fact that generally if an organization requires a service on one occasion then it is likely that it will need it on many other occasions as well. Therefore the user must register and provide the following information:

- Full name
- Age
- Net ID
- Email
- Password

3.2.2 SOURCE

The origin of this feature is from a discussion that the team members had together and agreed upon the fact that this feature is critical to the functionality of the application.

3.2.3 CONSTRAINTS

- The application must require as little information as possible when registering a user. The reasoning behind this is to ensure that the identity of the user is safeguarded.
- The application must use the following colors:
 - White (#FFFFFF)
 - Navy blue (#152272)
 - Bright Teal (#CCF2EA)

3.2.4 STANDARDS

This functionality does not have a standard because it is not heavy on technicality and is straightforward in terms of usage and implementation.

3.2.5 PRIORITY

This is a High priority feature due to its importance for the app developers and the improvements that it contributes to the overall user experience.

3.3 REGISTRATION CONFIRMATION EMAIL

3.3.1 DESCRIPTION

Once the user has successfully registered into the app, they must confirm their identity in order for the app to know that the account belongs to a real user. This will be done by sending a confirmation code to the provided email and the user must then enter that code into the confirmation page. If the user fails to provide the correct code then the account will not be created.

3.3.2 SOURCE

The origin of this feature is from a discussion that the team members had together and agreed upon the fact that this feature is critical to the functionality of the application.

3.3.3 CONSTRAINTS

- Correct code must be entered in order for the account to be created
- Code is only valid within a certain time period e.g. 1 min after the email is sent
- Code must be a 6 digit unique numerical value

3.3.4 STANDARDS

- The email will be sent immediately after completing registration and user will be guided to the email confirmation page
- User can not use the application until the account is confirmed
- Once the user enters the correct code they will be directed to the home page

3.3.5 PRIORITY

The is a low priority functionality, although it is important to make sure all registered accounts are real accounts held by real people, this check does not contribute heavily to the functionality of the app.

3.4 POLL CREATION

3.4.1 DESCRIPTION

Each user will have the ability to login to their account and instantly create a new poll for their desired occasion. The user will be presented with a blue(#409DE2) button with the title of start a new poll. After clicking on a form will pop up which will require the user to enter the following information:

- Poll Title
- Start Date
- End Date
- Question of the Poll
- Choices

3.4.2 SOURCE

The origin of this feature is from a discussion that the team members had together and agreed upon the fact that this feature is critical to the functionality of the application.

3.4.3 CONSTRAINTS

- The title of the poll must not exceed 30 characters
- End date can't be before start date
- The number of choices must be at least 2 for the poll to be valid if the poll is multiple choice

3.4.4 STANDARDS

- After the poll is created, the poll admin will be provided with a link to the poll
- The number of steps to create a poll will be minimal in order to remove the tedious nature of polls.
- After the user creates the poll, the application should momentarily load before directing the user back to their home page.

3.4.5 PRIORITY

This is a critical functionality of the application and is necessary for the application to achieve its intended purpose.

3.5 JOINING A POLL AND VOTING

3.5.1 DESCRIPTION

The user will be given the ability to join a poll through the usage of a link distributed by the poll creator. As soon as the voter accepts the invitation through the usage of the link it will be displayed in his/her feed. If the user decides to reject the invitation then they can click on the reject button as opposed to the accept button. If the user accepts the invitation, then they can vote on the option that fits their views and click submit. This will bring the voting process to an end and the user's vote will be registered into the database.

3.5.2 SOURCE

The origin of this feature is from a discussion that the team members had together and agreed upon the fact that this feature is critical to the functionality of the application.

3.5.3 CONSTRAINTS

- The voter can join the poll only once afterwards the link is invalid
- The voter can also vote on each poll once

3.5.4 STANDARDS

The main standard set by the team members is that when registering the user's votes, we must make sure that their identity is not revealed to other voters of the poll

3.5.5 PRIORITY

This is a high priority functionality because it is how the application achieves collecting votes and determining the results of the poll.

3.6 POLL INVITATION METHODS

3.6.1 DESCRIPTION

It is clear that the users must have the ability to invite people to a poll but the methodology in which they can do so is still unclear. It is the developers' task to ensure that the poll invitation process is smooth and that the available options are plentiful. The options that will exist for the users are as follows:

- Invite by distributing a link which will be associated with the created poll
- Invite users by importing an excel file containing information pertaining the people who should be invited
- Invite users by importing an CSV file containing information pertaining the people who should be invited
- enter voter information manually

3.6.2 SOURCE

The origin of this feature is from a discussion that the team members had together and agreed upon the fact that this feature is critical to the functionality of the application.

3.6.3 CONSTRAINTS

- The provided CSV file must contain emails of the voters' of the poll
- The provided Excel file must contain emails of the voters' of the poll
- The generated link must only work for individuals whose account email is within the list of registered users
- User entered emails must also be valid in order to be added to the list of registered voters

3.6.4 STANDARDS

- After the creation of the poll, the application will suggest the user to pick one of the invitation options
- Invite users by importing an excel file containing information pertaining the people who should be invited
- Invite users by importing an CSV file containing information pertaining the people who should be invited
- If the provided information is incorrect or is not a valid format then the application must generate an appropriate error.

3.6.5 PRIORITY

This is a high priority functionalities because without these options being provided to the user, there would be no way to share the poll with the intended audience.

3.7 POLL RESULTS

3.7.1 DESCRIPTION

If the user clicks on a poll whose results are in, they will see the result in another screen. In the result screen the user will see a graph in the middle of the page to showcase how the votes were distributed and also a list of the top three choices will be placed directly below chart along with their vote count. In addition, the user will have the option to export the data from the poll if they want to further analyze the results.

3.7.2 SOURCE

The origin of this feature is from a discussion that the team members had together and agreed upon the fact that this feature is critical to the functionality of the application.

3.7.3 CONSTRAINTS

If the user decides to export the results of the data, the data must not include information that could clearly identify the voters.

3.7.4 STANDARDS

The poll results must be exportable in a .csv standard

3.7.5 PRIORITY

This is a high priority functionality because it is the only way in which the users will know the result of the poll. If not included the voting process that the user went through would become useless.

3.8 POLL UPDATE

3.8.1 DESCRIPTION

As developers we don't expect our users to be perfect or free from error, everyone is capable of making a mistake and the application must accommodate this human flaw. In the case a user is accidentally gives a wrong name to the poll or forgets to add a question to the poll, the application must allow the user to go back into the poll and edit it.

3.8.2 SOURCE

The origin of this feature is from a discussion that the team members had together and agreed upon the fact that this feature is critical to the functionality of the application.

3.8.3 CONSTRAINTS

All modifications to the poll must be done prior to the poll becoming active otherwise some users might vote with incorrect information or not get to vote making the results of the polls inaccurate.

3.8.4 STANDARDS

- User must press on the poll for a few second to see the update option
- Before saving the changes user must verify the changes to the poll
- If user does not make any changes then previous version in the database should not be modified

3.8.5 PRIORITY

The is a high priority functionality because it allows the users to be in control of the whole process similar to conducting an in person poll.

3.9 POLL DELETION

3.9.1 DESCRIPTION

In the case of a poll not being what the user wanted or being out of date, it is critical for the user to be able to remove this poll from their screen and only have relevant polls on their main screen. This results in the user playing the role of a manager or administrator in the application and that is extremely important for user experience.

3.9.2 SOURCE

The origin of this feature is from a discussion that the team members had together and agreed upon the fact that this feature is critical to the functionality of the application.

3.9.3 CONSTRAINTS

- Users can only delete polls in which they are the administrator
- Active polls can not be deleted, poll administrator must wait until the poll period has ended

3.9.4 STANDARDS

- User must press on the poll for a few second to see the delete option
- Confirm if the user want to delete the poll before permanently removing it

3.9.5 PRIORITY

The is a high priority functionality because it allows the users to be in control of the whole process and it greatly improves the user's experience.

4 PACKAGING REQUIREMENTS

The application will be available on different platforms for regularity purposes. Compilation using Git/GitHub will allow agencies and companies that might be interested to sponsor the application and help developing and extending the functionalities of the application. The deployment on the App-Store the Play-Store only focuses on usage of the application with it's current functionality and will be available to use for free.

4.1 COMPILATION ON GIT/GITHUB

4.1.1 DESCRIPTION

The user will be able to access the application as an uploaded source code on GitHub and cloning it on their device to use it. The user will just have to run the executable file to run the application on a live server.

4.1.2 SOURCE

It was discussed within fellow team members to provide flexibility for all users on various platforms.

4.1.3 CONSTRAINTS

The user will have to install nodeJS and JavaScript dependencies in order to run the application. The user will might need to have a simulator/emulator if their web browser does not connect automatically.

4.1.4 STANDARDS

The user will be provided with all instructions to download and run the application on their computer within the GitHub page in a README.md file.

4.1.5 PRIORITY

This is a high priority because it will make the application accessible to all users and sides that are interested in it, and when presenting the project as software it will be more professional.

4.2 AVAILABILITY IN GOOGLE PLAY/APP STORE

4.2.1 DESCRIPTION

The application will be deployed to an online market to provide the main purpose of the application and to expand the popularity of it. Users will be able to find the application in either Google Play or the App Store.

4.2.2 SOURCE

It was discussed within fellow team members to provide flexibility for all users on various platforms.

4.2.3 CONSTRAINTS

- The application will be free on both platforms.
- To download the application, the phone should be compatible with application.

4.2.4 STANDARDS

- Users using Apple products will be able to find the application in the App Store.
- Users using Android products will able to find the application in the Google Play store.

4.2.5 PRIORITY

This is a low priority however it will make the application more accessible to users with ease, without worrying about installing extra dependencies or run commands.

5 PERFORMANCE REQUIREMENTS

This section sets constraints on how should the application operate. The application will have various requirements to maintain overall performance and uniqueness among other applications. The application is cross platform to make the application feasible when accessing from a device. Also, it's synced with Firebase database which allows for easy retrieval and changes for data. The application is also responsive to allow users to participate in real-time activities between a host and a voter.

5.1 CROSS PLATFORM

5.1.1 DESCRIPTION

The system will have cross platform functionality using the native components that React-Native provides for the application. The expected behavior for the application will be the same in both platforms with one code.

5.1.2 SOURCE

[3] Documentation provides how React Native is used to create native applications with its functional components.

5.1.3 CONSTRAINTS

Dependencies must match across platforms since packages do not support both IOS and Android devices, because they are built specifically for these devices.

5.1.4 STANDARDS

- The application will run on both platforms.
- The application will not need any other component to connect react native with the application.

5.1.5 PRIORITY

This is a high priority functionality because the application is expected to be accessible through both platforms and React Native provides this service to the users without worrying about installing any extra services.

5.2 REAL-TIME FIREBASE DATABASE

5.2.1 DESCRIPTION

The system is expected to verify and validate the user's information when attempting to log in. Also, it is expected from the system to be able to store the user's information in the Firebase database when the user enters their information at registering.

5.2.2 SOURCE

The origin of this feature is from a discussion that the team members had together and agreed upon the fact that this feature is critical since Firebase will allow the application to obtain and store the data efficiently.

5.2.3 CONSTRAINTS

The database must retrieve and update data for only the requested user.

5.2.4 STANDARDS

- Prevent any conflicts with the user's data by validating each user's information at registering through the application.

- Prevent any unauthorized access to the application by verifying the user's information when logging in to the application.

5.2.5 PRIORITY

This is a high priority functionality because this functionality will be the backbone where the data of all users will be stored and processed throughout the application to retrieve and update their data respectively.

5.3 REAL-TIME TRACKING

5.3.1 DESCRIPTION

The host should have the ability to monitor the ballot to track how many votes have been placed and how many still have to decide. Voters will be able to join ballots at the same time without any conflicts and place their votes, and update their current status' in real-time.

5.3.2 SOURCE

The origin of this feature is from a discussion that the team members had together and agreed upon the fact that this feature is critical since real time retrieval for data is mandatory in this application.

5.3.3 CONSTRAINTS

No delay should be encountered by the participants or the host when creating or joining the ballot.

5.3.4 STANDARDS

Hosts and participants status of voting should be synced in real-time.

5.3.5 PRIORITY

This functionality is a high priority since without it, the votes will lose their transparency and will not be effective.

5.4 RESPONSITIVITY

5.4.1 DESCRIPTION

The application should behave according to each component expected behavior. The user must not deal with any kind of difficulties when trying to click or swipe through the application. The colors of the interface should indicate to the user which components are accessible and which are not.

5.4.2 SOURCE

The origin of this feature is from a discussion that the team members had together and agreed upon the fact that this feature is critical since the responsivity of the application will enhance the overall experience of the user and will attract more users to it.

5.4.3 CONSTRAINTS

Outdated operating systems might have buffering because it could be incompatible with the application software which might affect the overall responsivity of the application.

5.4.4 STANDARDS

UI/UX design should create a layer of abstraction for all complexities in the application and make it easy and responsive for users to access/use it.

5.4.5 PRIORITY

This functionality is a high priority since the system should be responsive. Votes will be placed in real-time and any delay that might happen could affect the accuracy of the data.

6 SAFETY REQUIREMENTS

The most critical part of this section is ensuring the users' safety by deploying high security measures in the application. This will be done by setting security constraints to make sure the right people have access, people can only access the information they need, and making sure the consumer's information is not at risk of being accessed by a non-authorized 3rd party.

6.1 LABORATORY EQUIPMENT LOCKOUT/TAGOUT (LOTO) PROCEDURES

6.1.1 DESCRIPTION

Any fabrication equipment provided used in the development of the project shall be used in accordance with OSHA standard LOTO procedures. Locks and tags are installed on all equipment items that present use hazards, and ONLY the course instructor or designated teaching assistants may remove a lock. All locks will be immediately replaced once the equipment is no longer in use.

6.1.2 SOURCE

CSE Senior Design laboratory policy

6.1.3 CONSTRAINTS

Equipment usage, due to lock removal policies, will be limited to availability of the course instructor and designed teaching assistants.

6.1.4 STANDARDS

Occupational Safety and Health Standards 1910.147 - The control of hazardous energy (lockout/tagout).

6.1.5 PRIORITY

Critical

6.2 NATIONAL ELECTRIC CODE (NEC) WIRING COMPLIANCE

6.2.1 DESCRIPTION

Any electrical wiring must be completed in compliance with all requirements specified in the National Electric Code. This includes wire runs, insulation, grounding, enclosures, over-current protection, and all other specifications.

6.2.2 SOURCE

CSE Senior Design laboratory policy

6.2.3 CONSTRAINTS

High voltage power sources, as defined in NFPA 70, will be avoided as much as possible in order to minimize potential hazards.

6.2.4 STANDARDS

NFPA 70

6.2.5 PRIORITY

Critical

6.3 RIA ROBOTIC MANIPULATOR SAFETY STANDARDS

6.3.1 DESCRIPTION

Robotic manipulators, if used, will either housed in a compliant lockout cell with all required safety interlocks, or certified as a "collaborative" unit from the manufacturer.

6.3.2 SOURCE

CSE Senior Design laboratory policy

6.3.3 CONSTRAINTS

Collaborative robotic manipulators will be preferred over non-collaborative units in order to minimize potential hazards. Sourcing and use of any required safety interlock mechanisms will be the responsibility of the engineering team.

6.3.4 STANDARDS

ANSI/RIA R15.06-2012 American National Standard for Industrial Robots and Robot Systems, RIA TR15.606-2016 Collaborative Robots

6.3.5 PRIORITY

Critical

6.4 SMART CONTRACTS

6.4.1 DESCRIPTION

This functionality will generate a layer of abstraction to encrypt the votes and prevent them from getting accessed by unauthorized third parties to prevent fraud. These contracts will be coded in solidity to be stored on the Blockchain database for the safety of the users.

6.4.2 SOURCE

The origin of this feature is from a discussion that the team members had together and agreed upon the fact that this feature is critical since the main purpose of the application is to encrypt each ballot and prevent any attempt to manipulate these ballots.

6.4.3 CONSTRAINTS

When generating a smart contract, it will not be removed and cannot be modified from the Blockchain due to its nature.

6.4.4 STANDARDS

Each smart contract will be created to represent each ballot created respectively.

6.4.5 PRIORITY

This safety requirement is the core of the application since without using smart contracts, the ability to encrypt the ballots and make it secured will not be efficient and will allow hackers to break into the voting system and manipulate the users' data at ease.

6.5 CLOUD BACKUP

6.5.1 DESCRIPTION

Storing the users' data on Real-Time Firebase will provide the application with ability to store the data on the cloud to be able to back up the users' data in case something happens to the original system. This will prevent any loss of data and will eliminate any additional cost of trying to retrieve the data if it got lost.

6.5.2 SOURCE

The origin of this feature is from a discussion that the team members had together and agreed upon the fact that this feature is critical because users' will feel more safe when using the application.

6.5.3 CONSTRAINTS

There will be a payment for how much cloud space the application would need for users to use it.

6.5.4 STANDARDS

Data will automatically be backed up to the cloud when users create an account using the Firebase authentication system.

6.5.5 PRIORITY

This requirement is important because it will make assured that their data is always accessible and will never be lost.

6.6 AUTHENTICATION

6.6.1 DESCRIPTION

Authentication is essential to ensure the user and the administration are allowed to be in the voting app. When a vote is brought up, we want to make sure that the people who vote are allowed to vote to uphold the integrity of the poll.

6.6.2 SOURCE

This was an essential safety requirement that was agreed upon by every member of the team.

6.6.3 CONSTRAINTS

One constraint with the authentication method is that it must be adaptable to Blockchain and must be able to be easily updated.

6.6.4 STANDARDS

[2] NCQA standards.

6.6.5 PRIORITY

This is the most important feature of our voting app in order to ensure the uprightness of the app.

6.7 DATA PROTECTION

6.7.1 DESCRIPTION

Data protection is essential to make sure our customers feel secure using our app. We need to make sure that when there's a request and eventually a response that the information has not been compromised.

6.7.2 SOURCE

The source of the requirement is team member Kelechi Egbuta.

6.7.3 CONSTRAINTS

The method of data protection must be secure to ensure the safety of all the consumers' information. It must also enclose a disclosure for which information is going to be needed to register for the app.

6.7.4 STANDARDS

[1] Data protection act of 2018.

6.7.5 PRIORITY

Data protection is of high priority because we want to prevent these data from getting lost, stolen or fabricated.

6.8 AUTHORIZATION

6.8.1 DESCRIPTION

This requirement deals with validating users' information in order to give access to the application. It will also be used to verify what parts of the application the user would have access to, when using the application.

6.8.2 SOURCE

We were given this requirement by the customer to make sure that an average user is not able to manipulate a poll.

6.8.3 CONSTRAINTS

A layer of abstraction will be created to prevent the user to see the complexity of dealing with the back-end connection of the poll. It includes editing the poll, being able to see who voted, monitoring the number of placed votes and the results before it is publicly published, etc.

6.8.4 STANDARDS

[2] NCQA standards.

6.8.5 PRIORITY

This requirement is considered a high priority due to how critical users' data are. Since any data leak might threaten the privacy of users.

7 MAINTENANCE & SUPPORT REQUIREMENTS

This section focuses on constant development and improvement for the application by good design and project blue prints. To ensure our application's system maintainability, one need to create documentation for the written code and design a reliable schema design to be able to manage and administrate the database efficiently. Also, users' feedback have a critical role in the development process since we will be making changes according to their experience.

7.1 OBTAIN USER-DATA

7.1.1 DESCRIPTION

This requirement will help us understand what areas of the application is causing distress for the users. Obtaining data and analyzing reviews about the application will help with dealing with bugs and technical issues in the application.

7.1.2 SOURCE

The origin of this feature is from a discussion that the team members had together and agreed upon the fact that this feature is critical to the functionality of the application.

7.1.3 CONSTRAINTS

- The user-data must be hidden from the user's outreach since it will be sensitive data and any leak could cause vulnerability issues in the application.
- Optional data collection.

7.1.4 STANDARDS

Analyzing the data to fix the issues that users are dealing with in the application and improve overall performance of the application.

7.1.5 PRIORITY

This is an optional requirement since it will be costly to maintain and analyze user's data, and it will also require some time to fix these bugs.

7.2 DOCUMENT CODE

7.2.1 DESCRIPTION

A detailed written description of each component's expected functionality in the application. It must be written professionally without any grammar issues. It will be used as a reference when developers need to fix the errors that exist in the application.

7.2.2 SOURCE

The origin of this feature is from a discussion that the team members had together and agreed upon the fact that this feature is critical to the functionality of the application.

7.2.3 CONSTRAINTS

The writing and naming conventions must stay consistent across the application.

7.2.4 STANDARDS

It should contain the detailed description of the source code. It will help developers to have a reference whenever they want to fix something since it is impossible to remember everything that was implemented in the application.

7.2.5 PRIORITY

This functionality is important because it will help developers distribute their work efficiently without worrying about overall contribution and relying on their memory to remember code to debug.

7.3 BLOCKCHAIN MANAGEMENT

7.3.1 DESCRIPTION

Depending on what Blockchain platform we decide to build our voting platform on, the platform specifications could always change down the road. If for some reason our smart contract code is no longer compatible with the Blockchain, we need to update it.

7.3.2 SOURCE

The origin of this feature is from a discussion that the team members had together and agreed upon the fact that this feature is critical to the functionality of the application.

7.3.3 CONSTRAINTS

In the case that updates need to be made so that our smart contracts are still secure, updates need to be made promptly. Incompatible code could lead to vulnerabilities in our smart contracts and leave the program open to exploitation.

7.3.4 STANDARDS

Monitor the Blockchain platform we use for updates or any other changes that might need to be made and make these updates promptly.

7.3.5 PRIORITY

This is a medium priority requirement since it will be essential to maintaining the security and validity of our program and accuracy of the software.

7.4 USER SUBMITTED REPORTS

7.4.1 DESCRIPTION

The application will allow users to report any problems or inconveniences found in the program via a form that they could fill out. The application maintenance team will then ensure that the issue is resolved. Users can submit these if the application crashes, in which case a pop-up will appear to invite the user to report the event, or they can submit their complaints by clicking on the support page.

7.4.2 SOURCE

The origin of this feature is from a discussion that the team members had together and agreed upon the fact that this feature would be a great addition to the application.

7.4.3 CONSTRAINTS

The users will be limited on how many reports that they can submit based on how many that are already being processed. For example, if a user submits three reports they should wait for at least one of those three to be resolved prior to submitting another one. Too many reports could lead to some reports not being solved therefore there must be an adequate limit on bug reporting.

7.4.4 STANDARDS

- The user will always be able to report issues in the application in the help tab.
- The maintenance team must resolve each report within one week or provide update to the user.

7.4.5 PRIORITY

This is a low priority requirement which will be implemented in the case where the higher priority and medium priority requirements of the applications are completed.

7.5 TROUBLESHOOTING GUIDE

7.5.1 DESCRIPTION

In the case where the user is in a situation where they can not wait for help from the maintenance team they can follow the troubleshooting guideline will provide detailed information on how to solve certain issue. Users can also contribute by adding their solutions to the guide if a solution is not already present. The troubleshooting guide will be placed in the help tab.

7.5.2 SOURCE

The origin of this feature is from a discussion that the team members had together and agreed upon the fact that this feature would be a great addition to the application.

7.5.3 CONSTRAINTS

- The guide must be consistently updated to ensure correctness.
- User contributions must be validated prior to incorporation into the guide.

7.5.4 STANDARDS

- The users will be provided with an online document containing the guide.
- The users will be provided with a submission form for a new addition into the guide.
- Users will be notified on the status of their submission for the guide, whether it was accepted or rejected.

7.5.5 PRIORITY

This is a low priority feature which will be completed at the latest stages of the application.

8 OTHER REQUIREMENTS

The Blockchain application is more than just casting votes and collecting data provided by the user. In order to meet the expectations of our customers and users, the application must meet various other standards in addition to the ones listed above. The following requirements will be implemented, which will considerably improve the overall experience of our application.

8.1 ERROR HANDLING

8.1.1 DESCRIPTION

The application must go through several test cases before deployment to verify that the application will be able to handle any situation in case the operation was successful or not. Detecting any connectivity or compatibility issues from the user's end and showing it to them should be handled to prevent any delay that might happen during voting.

8.1.2 SOURCE

The origin of this feature is from a discussion that the team members had together and agreed upon the fact that this feature is critical to the functionality of the application.

8.1.3 CONSTRAINTS

Workflow diagrams and test cases must be performed to see if the expected functionality of the application is operational without any issues.

8.1.4 STANDARDS

The application must prompt to the user in case a connectivity or compatibility issue occurred.

8.1.5 PRIORITY

It is very important to handle any edge cases since the application should have a deterministic behaviour to avoid ambiguity and provide the best overall experience to all users using the application.

8.2 USABILITY STANDARDS

8.2.1 DESCRIPTION

The application must adhere to the usability standards of UX in order for the application function smoothly and improve the experience of the user. Jakob Nielsen's ten heuristic laws will be utilized extensively to ensure that the design of the application meets the usability standards.

8.2.2 SOURCE

The origin of this idea is from Ahmed who has taken a course where these laws were used and he believes that the incorporation of these techniques will improve the overall design of the application.

8.2.3 CONSTRAINTS

- Requires at least two team members to perform the evaluation.
- Developers must understand the customer's vision of the application or discuss before hand.
- Constant communication with the customer about the progression of the application design to make sure that the app appeals to the customer.

8.2.4 STANDARDS

- Conduct the evaluation prior to the implementation to ensure that customer and developer are on the same page.
- All evaluations must be conducted separately/remotely from one another to ensure that each evaluation brings out various methods of improvements.
- After the evaluation if the customer is dissatisfied with the design, then add the suggested improvements to the design and again show the customer the new design. Repeat until the customer is satisfied.

8.2.5 PRIORITY

The priority of this high because it is important for the application to be usable and enjoyable for our users. Typically if an application is difficult to use or confusing to the user, they never return to the app and that is something we would not like to happen.

8.3 ACCESSIBILITY STANDARDS

8.3.1 DESCRIPTION

As our application is geared towards the general public, all people should have a great experience when using the application. This means that we must also accommodate for people with disabilities in our application to guarantee that the application give equal access to all users.

8.3.2 SOURCE

The origin of this idea is from Ahmed who has taken a course where the importance of accessibility were greatly emphasized and he therefore believes that the incorporation of these techniques will improve the overall design of the application.

8.3.3 CONSTRAINTS

- The developers must depend on third party resources for conducting automated tests
- The developers must find time to manually conduct their own examination in order to find an missed accessibility violations

8.3.4 STANDARDS

- Accessibility tests must be ran manually by the developers prior to the implementation of the design and post implementation using software.
- Each team member must evaluate the usability of the application separately in order to get a wide range of opinions.
- Developers must improve the accessibility of each screen prior to moving on to a different screen of the application.

8.3.5 PRIORITY

The priority of this high because it is important our application performs a very important task where everyone's participation is important therefore it is critical that we appeal to as many people as possible.

8.4 FEASIBILITY STANDARDS

8.4.1 DESCRIPTION

The Blockchain voting application must be usable on either a mobile application or on the web. In case users are low on available storage on their smartphone, the web application will be the other available option for them.

8.4.2 SOURCE

The origin of this idea comes from the fact that we believed that having multiple methods of using the application was important and our experience with the JavaScript library react-native made the idea even more reasonable and worth implementing.

8.4.3 CONSTRAINTS

- Both the web and mobile application must be developed using react-native to cut down on time.
- Developers must be able to learn new and useful libraries in order to keep up the pace for the application.
- User will require sufficient internet connection in order for either the phone or the web application to perform properly.

8.4.4 STANDARDS

- It must be checked that each screen works properly on both mobile and web.
- Each team member must update teammates on if issues arise on either platforms to ensure that others don't find themselves facing similar issues.

8.4.5 PRIORITY

The priority of this high and by the nature of using react-native it will be done simultaneously.

8.5 EXTENSIBILITY

8.5.1 DESCRIPTION

The application must be implemented in a manner in which it can be easily extended in case of new features and improvements. The developers must make use of customizable components that would act as templates for future users who are working on the application.

8.5.2 SOURCE

We as a group came to the conclusion that to make our individual components easier it would be more efficient and helpful that we create custom components and with the usage of react-native and JavaScript it would be very easy to do.

8.5.3 CONSTRAINTS

- All repeatable components must be customizable for future usage.
- Developers must ensure that custom components can work with various data types.

8.5.4 STANDARDS

- All custom components must be added to a single folder which other developers can access
- Custom components must be documented to indicate that the component is extendable

8.5.5 PRIORITY

The priority of this is medium however it is very useful if all team members implement their code in an extendible manner.

9 FUTURE ITEMS

This section covers the requirements that would enhance our application but would take longer than the time we were given to implement. All these requirements would improve the overall performance, support and maintenance, and other requirements that would improve our overall design.

9.1 TWO-FACTOR AUTHENTICATION

9.1.1 DESCRIPTION

In order to further protect the identity of our users, two-factor authentication would be greatly useful. We would allow users to directly receive either text messages containing a code in which they would use to verify their identity on the application or use one of the popular applications like Google's Authenticator to generate a code which would serve the same purpose.

9.1.2 SOURCE

The group members discussed some of the ways that the application can become even more secure and it has been proven that two-factor authentication is a great option.

9.1.3 CONSTRAINTS

- User must enter the generated code within 60 sec.
- Users must be either willing to download a third party application or provide phone number.
- If user provides phone number for authentication the application must not save it.

9.1.4 STANDARDS

- Generated codes are 6 digit numeric values.
- Generated codes must renew every 60 seconds.

9.1.5 PRIORITY

This is a low priority feature which most likely won't be implemented due to time constraints. However, if the project is completed early then it is highly likely that it will become part of the application.

9.2 EXTENSION OF USER INFORMATION

9.2.1 DESCRIPTION

In addition to the current information that we collect from the user during the registration phase, in the future we would like to collect more information such as gender, race, location, etc. for data analysis. The application as a result would be able to provide more information about the results of each poll which extends the knowledge of both the voters and administrators on the topic of the poll.

9.2.2 SOURCE

The idea came from a team discussion and we came to the conclusion that in addition to just knowing the result of the poll, it would be great to provide extra information to the user which would allow them to filter the results of the poll based on various criteria.

9.2.3 CONSTRAINTS

- Identity critical information must not be collected from the user in under any circumstance.
- If user decides to change their registered information then they must verify their identity.

- User must be provided with the option to skip providing the extra data.
- User contact information must never be collected

9.2.4 STANDARDS

- The type of information collected from all users must be uniform.
- The extra information must be either in a drop menu or be displayed in a way that indicates its optional.

9.2.5 PRIORITY

This is a low priority feature due its lack of effect on the overall application design however it is a feature that increases the quality of the application and its utility to the users.

9.3 USER SUCCESS RATE

9.3.1 DESCRIPTION

Similar to politicians with their legislation pass rate, we believe it would be useful for voters to know their pass/success rate as well. It would provide great insight to them in the sense that they would know if their views are aligned with their peers/co-workers and where they stand on the spectrum.

9.3.2 SOURCE

This was a team effort in the sense that the entire group contributed to the creation of the idea. We all believe that this feature would be extremely beneficial for the users of our application.

9.3.3 CONSTRAINTS

- The user will only be able to view the option they voted for and the winning option in order to protect other user information.
- After a year users will not be able to view past polls, in other words the poll will be discarded after a certain time period.

9.3.4 STANDARDS

- The success rate page must contain a pie chart or a graph to visually demonstrate the success rate.
- Each poll in which the user's vote won should be highlighted green while others are marked red. This will also enhance the visual aspects of the application.
- Users should have the option to filter the results based on the date or similar topics.

9.3.5 PRIORITY

This is a low priority and it is unlikely to be implemented within the project time frame.

9.4 NOTIFICATIONS TO REMIND USERS TO PARTICIPATE

9.4.1 DESCRIPTION

People are often busy and voting just happens to be one of those things that people often forget about. In order to make sure that the users don't miss the voting window they will be notified by the application within adjustable time frames that they can set or can be set for them if they choose to turn on notification. This can also be extended to when the result of the poll is released so they can be notified of the poll result as well.

9.4.2 SOURCE

This was a team effort in the sense that the entire group contributed to the creation of the idea. We understood the need for this feature from our experience with assignments and due dates in university and we saw that there are similarities between the two.

9.4.3 CONSTRAINTS

- User must receive one notification per provided time periods.
- Notification message must include poll name and remaining time period.
- User must have the ability to turn off the notification in the poll page.

9.4.4 STANDARDS

- Notification must follow push notification standard.
- Notification must show application logo so that the user is aware of what to do next.

9.4.5 PRIORITY

This is a low priority and it is unlikely to be implemented within the project time frame.

9.5 POLL TIME EXTENSION REQUEST

9.5.1 DESCRIPTION

Similar to the notification feature, this feature is meant help out the voters in the voting process. If the user is registered for a poll and they find themselves in a situation where they won't be able to vote, they should have the option of submitting a poll extension request where the poll administrator can choose to either grant the request or not.

9.5.2 SOURCE

This was a team effort in the sense that the entire group contributed to the creation of the idea. This feature is an offspring of the notification feature or are identical in the sense that their aim is to ensure that all registered voters are able to vote.

9.5.3 CONSTRAINTS

- Users must be limited on how often they use this feature and if they exceed the allotted amount then they should not be able to request the extension.
- For the reasoning of the extension the user must provide a brief explanation that does not exceed 100 characters.

9.5.4 STANDARDS

- Poll extension form must include a brief explanation of the reason for extension.
- Administrators should be able to view extension requests in a tab that allows them to view all requested forms.
- For administrators the forms should be sorted by which polls the request is made from.

9.5.5 PRIORITY

This is a low priority and it is unlikely to be implemented within the project time frame.

REFERENCES

- [1] UK government. The Data Protection Act.
- [2] NCQA. National Committee for Quality Assurance.
- [3] reactnative.dev. React Native.