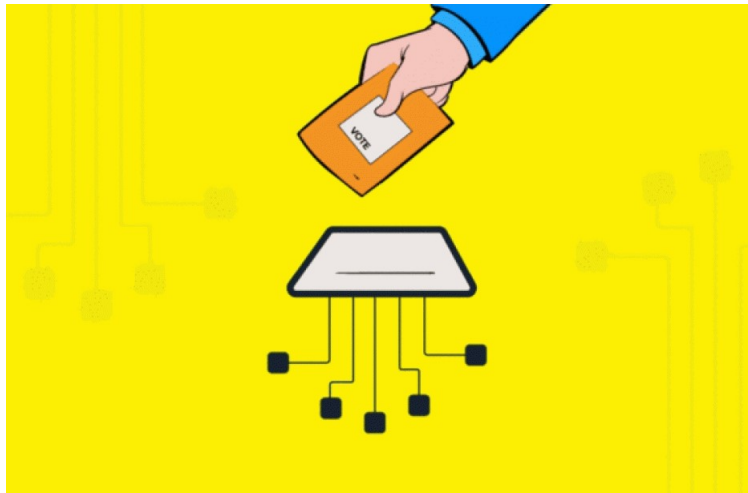


**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING  
THE UNIVERSITY OF TEXAS AT ARLINGTON**

**PROJECT CHARTER  
CSE 4316: SENIOR DESIGN I  
SUMMER 2022**



**THE CHAINVOTERS  
BLOCKCHAIN VOTING**

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

## REVISION HISTORY

| Revision | Date      | Author(s)      | Description       |
|----------|-----------|----------------|-------------------|
| 0.1      | 6.25.2022 | AH             | document creation |
| 0.2      | 7.01.2022 | AH, AS         | complete draft    |
| 1.0      | 7.12.2022 | AH, AS, SE, KE | official release  |

# CONTENTS

|   |           |
|---|-----------|
| <b>1 Problem Statement</b>                          | <b>6</b>  |
| <b>2 Methodology</b>                                | <b>6</b>  |
| <b>3 Value Proposition</b>                          | <b>6</b>  |
| <b>4 Development Milestones</b>                     | <b>6</b>  |
| <b>5 Background</b>                                 | <b>7</b>  |
| <b>6 Related Work</b>                               | <b>7</b>  |
| <b>7 System Overview</b>                            | <b>8</b>  |
| <b>8 Roles &amp; Responsibilities</b>               | <b>9</b>  |
| <b>9 Cost Proposal</b>                              | <b>9</b>  |
| 9.1 Preliminary Budget . . . . .                    | 9         |
| 9.2 Current & Pending Support . . . . .             | 9         |
| <b>10 Facilities &amp; Equipment</b>                | <b>10</b> |
| <b>11 Assumptions</b>                               | <b>10</b> |
| <b>12 Constraints</b>                               | <b>11</b> |
| <b>13 Risks</b>                                     | <b>11</b> |
| <b>14 Documentation &amp; Reporting</b>             | <b>11</b> |
| 14.1 Major Documentation Deliverables . . . . .     | 11        |
| 14.1.1 Project Charter . . . . .                    | 11        |
| 14.1.2 System Requirements Specification . . . . .  | 12        |
| 14.1.3 Architectural Design Specification . . . . . | 12        |
| 14.1.4 Detailed Design Specification . . . . .      | 12        |
| 14.2 Recurring Sprint Items . . . . .               | 12        |
| 14.2.1 Product Backlog . . . . .                    | 12        |
| 14.2.2 Sprint Planning . . . . .                    | 12        |
| 14.2.3 Sprint Goal . . . . .                        | 12        |
| 14.2.4 Sprint Backlog . . . . .                     | 12        |
| 14.2.5 Task Breakdown . . . . .                     | 12        |
| 14.2.6 Sprint Burn Down Charts . . . . .            | 12        |
| 14.2.7 Sprint Retrospective . . . . .               | 12        |
| 14.2.8 Individual Status Reports . . . . .          | 13        |
| 14.2.9 Engineering Notebooks . . . . .              | 13        |
| 14.3 Closeout Materials . . . . .                   | 13        |
| 14.3.1 System Prototype . . . . .                   | 13        |
| 14.3.2 Project Poster . . . . .                     | 13        |
| 14.3.3 Web Page . . . . .                           | 13        |

|  |    |
|--|----|
| 14.3.4 Demo Video . . . . .                | 13 |
| 14.3.5 Source Code . . . . .               | 14 |
| 14.3.6 Source Code Documentation . . . . . | 14 |
| 14.3.7 Installation Scripts . . . . .      | 14 |
| 14.3.8 User Manual . . . . .               | 14 |

## LIST OF FIGURES

|   |  |    |
|---|--|----|
| 1 | Example sprint burn down chart . . . . . | 13 |
|---|--|----|

## **1 PROBLEM STATEMENT**

As the United States is in the midst of a political uproar resulting from numerous controversial decisions, it has become vital for U.S. citizens to vote and make their voice heard. However, up until now the voter turnout each year has been on a downward trajectory due to numerous causes. Some of the causes for low voter turnout include the inconvenience of the voting process, the validity of the results and the sentiment that one's vote will not make a difference. This is problematic because this can in the worst case lead to the issues that exist in communities and cities not to be addressed despite their severity.

## **2 METHODOLOGY**

We will build a cross platform application that will allow users to register to vote during the designated time period. This application will solve the issues in our current voting system by allowing the user to vote from the comfort of their own home or anywhere else in the world, the user will be able to see exactly how the votes were distributed to the various candidates and overall will foster a sense of legitimacy. With these proposed solutions, our application will increase the participation of people in the voting process, which will of course be beneficial for the greater society.

## **3 VALUE PROPOSITION**

The value of this application lies in its ability to host safe and secure elections and polls. Government entities can host their elections on the application and they would benefit greatly from the security and efficiency of the application. The grueling process of having to count the votes in order to ensure that the rightful candidate is given the position will be avoided and that is extremely beneficial for both the voters and the government. For the general person such as a school teacher or a store manager it can also help in determining important decision by collecting people's opinions and returning concise and easy to understand results. For the students who are working on the application, they become exposed to technologies that will soon become prevalent and highly sought after by employees. Overall, all stakeholders will benefit in their own unique way which makes this application even more necessary to build.

## **4 DEVELOPMENT MILESTONES**

- Project Charter first draft - July 2022
- System Requirements Specification - July 2022
- Architectural Design Specification - August 2022
- Demonstration of landing page - August 2022
- Detailed Design Specification - September 2022
- Demonstration of Login and Registration Pages - August 2022
- Demonstration of homepage - September 2022
- CoE Innovation Day poster presentation - October 2022
- Demonstration of poll creation page - October 2022
- Demonstration of vote placement page - October 2022
- Demonstration of poll result page - November 2022
- Final Project Demonstration - December 2022

## 5 BACKGROUND

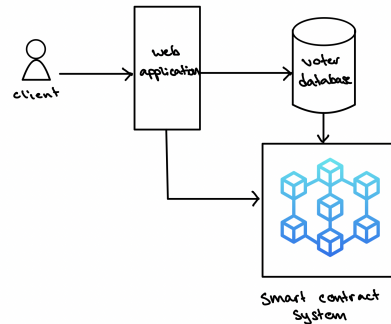
Currently the voting system that is in place requires all citizens to register prior to actually voting and when it is time to vote they either have to physically go to a voting location or through absentee voting. The current system does not always make it easy for racial minorities and disabled individuals to vote, in fact it requires significantly more effort for them to vote. People who fit in the categories mentioned above often don't get the opportunity to vote and as a result their voices are rarely heard. For the general voter, their issues are inconvenience and dissatisfaction with the current process. For example in the 2016 U.S. elections it was revealed that Russian hackers were caught interfering with the election by accessing the computers of individuals involved in the election. This conspiracy has led to many people questioning validity of the elections and has caused some to not even want to participate. For a while the U.S. has been facing issues with getting people into the voting booths but situations such as the 2016 elections exacerbated the problem. As of 2022 the u.s. government is under a lot of pressure due to recent decisions such as the Roe v. Wade and people will be paying special attention to the results of the upcoming elections. This means that the voting system in place must be robust, secure and trustworthy. Our application will be exactly that. The blockchain voting app will rely on blockchain technology to ensure that the voters' data is not tampered. The general voters, disabled voters and others will all have equal opportunities to vote due the fact that the application can be launched on either their phone or computer. The votes can also be easily audited for correctness while maintaining the anonymity of the voters allowing the system to be extremely reliable and efficient. The transparent nature of blockchain technology allows the poll/election administrators to easily check the validity of each vote eliminating the possibility of voter fraud from occurring during the election. This application will solve all the issues mentioned earlier by providing the user with an intuitive and easy interface that uses blockchain technology as the application database and it will be accessible for all users.

## 6 RELATED WORK

As of now, there are a few applications that utilize the blockchain in order to conduct votes. One example is Voatz [3] which is a company that is headquartered in Boston, Massachusetts. The company claims to offer "cutting-edge technology so people can vote securely from anywhere and administrators can manage elections with ease". The issue with their implementation of the idea is that they are too narrow scoped. The application is tailored for government elections and is not designed for other usages. Voatz is a commercial application that is cross platform as well but due to it lack of flexibility the application is different from ours. In addition Voatz android application was found to contain numerous vulnerabilities that put the voters at risk. Due to these reasons the application is disqualified and is no longer considered as a viable option. Another company that takes advantage of the blockchain technology is Votem [4]. Votem is a for-profit organization that makes the same claim as Voatz [3] by stating that their "mission is to expand access, fortify, and restore trust in elections while decreasing the cost to Election Management Bodies around the world" Although there are no existing indication that the application contains any bugs, this application is also too narrow to serve the purpose of our application. Democracy.Earth [1] is a non-profit foundation that has created a decentralized application for voting and enabling democracy to thrive. It is a UBI application that uses ethereum which our application does not use. Followmyvote is another commercial application [2] which allows secure end to end voting through the usage of blockchain. Followmyvote suffers from difficulty of usage due by having the user enter a lot of information which is similar to physically voting and that disqualifies it from contention. Lastly VoxVote [5] is an application that is extremely close to ours in the sense that it is a polling application that is multi-purposed; however, this is a commercial application which requires users to buy products and packages in order to use their services. Our application is free to use and does not require any in app purchases.

## 7 SYSTEM OVERVIEW

We will implement a solution to the problem using a smart contract system to create a voting platform that can not be manipulated by outside forces.



| User                 |
|----------------------|
| + email : String     |
| + password: String   |
| + SSN : String       |
| + DOB : String       |
| + log In ()          |
| + place Vote ()      |
| + change Password () |

| Election               |
|------------------------|
| + vote Choices: array  |
| + vote Count : Integer |
| + voter List: array    |
| + registerVote ()      |
| + get Count ()         |
| + get Results ()       |

When the client access the application they will be redirected to a landing page which shows quick information about the purpose and importance of the application. Each client must access their information by initially creating an account and signing into the application using their credentials. They will be prompted to enter their email and password to sign in or they can sign in using their google account. However, they will need to register first in order to login. If the client forgot their password, they will be directed to a page that sends them an authentication email and allow them to change their password into a new one. After authentication, The user will be able to select between viewing their history or creating/joining a poll. At the time of creation of a ballot, a smart contract system will be created using blockchain technology which provides a high encryption system. These implementation will not be visible to the client because of their complexities; however, the user will only see the text fields to enter information about the ballot such as, name, time limit, or the maximum capacity of participants. The voter's information will be linked directly to each smart contract through Firebase. The data will be used in tracking and monitoring the votes, which are smart contracts. Using blockchain, smart contracts will be generated and will represent a voting session. On the other hand, if a user joins a poll, a smart contract will not be generated since it already exists, so any new participant in the ballot will be added to the same contract to prevent fraud. In addition, if the client wanted to review the previous votes that they participated in, they will be able to because the application keeps track of every ballot as a smart contract, so the polls will be listed along with the results of the ballot. The client will also be able to export this information from the smart contract system to an exportable form to use this data in different application.



## 8 ROLES & RESPONSIBILITIES

The stakeholders of the project include the project members and Professor Conly. The main point of contact from the sponsor will be Ahmed, however, each of the team members is able to contact the sponsor as needed. The different roles are as given:

**Product owner:** This is the person who is the key stakeholder and lead user for project blockchain voting. They set a clear direction for the project. Their main job includes communicating our project vision to our sponsor and our consumers as well as prioritizing which items on the product backlog are most important and should be completed first.

**Scrum master:** The scrum master is the project member who guarantees that each member is following the plan as agreed upon. The scrum master is in charge of the meetings (spring retrospective, daily 15 min meetings, etc) and works with the product owner on the prioritizing of items on the product backlog.

**Development team member:** The development team is the group of people who are responsible for making sure each sprint is completed on time.

The different team members and their responsibilities are:

- Ahmed Harun - Product owner, Development team member
- Seth Rodgers - Development team member
- Kelechi Egbuta - Development team member
- Abdulla Sakallah - Scrum master, development team member

For our project blockchain voting, we plan on changing the product owner and scrum master as needed. This is because each project member has a different vision and idea that they would like to see carried out and looking at the project in its entirety from different lenses can prove to be beneficial to the overall project. To make sure that the new scrum master/product owner does not come in blind, we will ensure that every member of the team is a part of every major decision taken so when we eventually change the responsibilities, the handoff process is not nearly as troubling.

## 9 COST PROPOSAL

We will be provided with the support of the CSE department as the department will be our main support. The current fees required for the application to be assembled as a connected interactive application would mainly be due to the implementation of Firebase since it is the major expense that the application requires. Since the application is made to be maintained on a big scale, it will need to be able to authenticate and store data for many users which is why we are investing in implementing Firebase in the app.

### 9.1 PRELIMINARY BUDGET

### 9.2 CURRENT & PENDING SUPPORT

Our main and only funding source for this project would be the CSE department. They are funding us with an amount of \$ 800. There are not any additional sponsors for this project.

| Component description | The Estimated Percentage of a Component |
|-----------------------|---|
| Firestore services    | 50%                                     |
| AWS                   | 5%                                      |

Table 1: Overview of Component Budgeting

## 10 FACILITIES & EQUIPMENT

This project only deals with an intangible system infrastructure so we will not need a physical testing ground space. Also, since it is a project involving a software product our need for physical facilities and equipment to use is minimal. However, we need some essentials that will help us to present and communicate with one another better. We will need a development space that is equipped with whiteboards, dry markers and erasers to brainstorm and present our ideas efficiently. Also, we will need workstations that have 2 monitors for each team member as it will be difficult to keep track of our thought processes using only 1 screen. We will need a strong and reliable WiFi connection to be able to compile our code and push it to GitHub or researching possible methodologies that might help us to build the application. We need a projector to discuss our goals in the weekly meeting for sprint reviews, and evaluate built components as a team, from a user point of view. We will also need iMacs to work on the application since we will be using simulators to debug our code which uses a lot of memory, and not all devices can handle this intensity. On each one of these iMacs, there needs to be a VScode code editor and all the JavaScript dependencies installed on it to prevent dependency issues upon transferring data.

## 11 ASSUMPTIONS

An assumption is a belief of what you assume to be true in the future. You make assumptions based on your knowledge, experience or the information available on hand. These are anticipated events or circumstances that are expected to occur during your project's life cycle.

Assumptions are supposed to be true but do not necessarily end up being true. Sometimes they may turn out to be false, which can affect your project significantly. They add risks to the project because they may or may not be true. For example, if you are working on an outdoor unmanned vehicle, are you assuming that testing space will be available when needed? Are you relying on an external team or contractor to provide a certain subsystem on time? If you are working at a customer facility or deploying on their computing infrastructure, are you assuming you will be granted physical access or network credentials?

This section should contain a list of at least 5 of the most critical assumptions related to your project. For example:

The following list contains critical assumptions related to the implementation and testing of the project.

- We will be able to access and utilize a blockchain library like solidity that will work with our project without major issues.
- We will be able to simulate enough users to replicate a real election.
- We will be able to efficiently store the data somehow for the smart contracts to be able to use it.
- We will be able to use existing verification systems (like government ID or SSN) to validate identity of users.

- We will have access to proper documentation and tutorials to be able to learn the smart contract development.

## 12 CONSTRAINTS

Constraints are limitations imposed on the project, such as the limitation of cost, schedule, or resources, and you have to work within the boundaries restricted by these constraints. All projects have constraints, which are defined and identified at the beginning of the project.

Constraints are outside of your control. They are imposed upon you by your client, organization, government regulations, availability of resources, etc. Occasionally, identified constraints turn out to be false. This is often beneficial to the development team, since it removes items that could potentially affect progress.

This section should contain a list of at least 5 of the most critical constraints related to your project. For example:

The following list contains key constraints related to the implementation and testing of the project.

- All voter data must remain private and not exposed by the application.
- The smart contracts must be written in a way that the voting system cannot be exploited.
- The voting system should be able to support elections of different sizes and criteria.
- The voting system should allow all users to only vote once per election and have a way to monitor this.
- The application should allow multiple elections to take place and different users to vote in each election they are apart of.

## 13 RISKS

This section should contain a list of at least 5 of the most critical risks related to your project. Additionally, the probability of occurrence, size of loss, and risk exposure should be listed. For size of loss, express units as the number of days by which the project schedule would be delayed. For risk exposure, multiply the size of loss by the probability of occurrence to obtain the exposure in days. For example:

The following high-level risk census contains identified project risks with the highest exposure. Mitigation strategies will be discussed in future planning sessions.

| Risk description   | Probability | Loss (days) | Exposure (days) |
|--|-------------|-------------|-----------------|
| Flaw in our smart contracts that break the voting system       | 0.50        | 18          | 9               |
| Blockchain library not compatible with technologies being used | 0.40        | 20          | 8               |
| Error/Loss in database for voters                              | 0.70        | 10          | 7               |
| Issues with users voting multiple times                        | 0.60        | 20          | 12              |
| Issues managing voters and what elections they belong to       | 0.30        | 15          | 4.5             |

Table 2: Overview of highest exposure project risks

## 14 DOCUMENTATION & REPORTING

### 14.1 MAJOR DOCUMENTATION DELIVERABLES

#### 14.1.1 PROJECT CHARTER

We will update this document and each sprint as needed and other times when big changes or decisions are made as a group. Initial July 12 2022 Final December 2022

### **14.1.2 SYSTEM REQUIREMENTS SPECIFICATION**

We will update this document as a group each sprint as needed and other times when big changes or decisions are made as a group. The bulk of the document will be created in July. Initial July 29 2022 Final December 2022

### **14.1.3 ARCHITECTURAL DESIGN SPECIFICATION**

We will update this document as a group each sprint as needed and other times when big changes or decisions are made as a group. The bulk of the document will be created this summer. Initial August 10 2022 Final December 2022

### **14.1.4 DETAILED DESIGN SPECIFICATION**

We will update this document as a group as needed when big changes or decisions are made as a group. The bulk of the document will be created this Fall once we have the rough draft of our design specifications. Initial September 2022 Final December 2022

## **14.2 RECURRING SPRINT ITEMS**

### **14.2.1 PRODUCT BACKLOG**

We will pick items as a group in order from the SRS based on a sequential design process, building initial features first and adding features one by one. Each sprint we will decide as a group which items get added to the sprint backlog. We will use google docs to maintain and share the document.

### **14.2.2 SPRINT PLANNING**

Each sprint will be planned as a group based on our top priorities for that sprint. There will be 3 sprints in SD1 and 3 sprints in SD2, so 6 total.

### **14.2.3 SPRINT GOAL**

We will decide each sprint goal as a group, and we will take the perspective as the customer to ensure we are working on the right tasks and making appropriate progress.

### **14.2.4 SPRINT BACKLOG**

We will decide as a group which items make it into the sprint backlog. We will maintain this with a simple google doc.

### **14.2.5 TASK BREAKDOWN**

We will individually claim small tasks while deciding as a group who works on which core sections of the project. Each person will document their time personally.

### **14.2.6 SPRINT BURN DOWN CHARTS**

We will rotate the responsibility of the burn-down charts for different sprints, and the person making the chart will collect hours from each user to sum them up and make the chart. We will use google sheets to make the charts.

### **14.2.7 SPRINT RETROSPECTIVE**

The sprint retrospective will be handled over discord or in-person(if available). This discussion will happen after the sprint presentation. The documentation will include what we are still working on, stopped working on, and what we are set to start working on. The sprint retrospective must happen no longer than a week after the spring presentations.

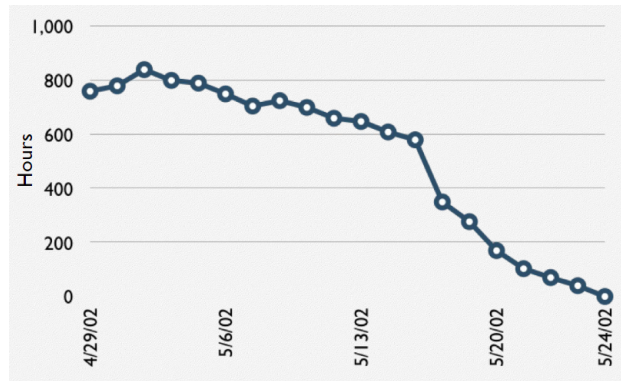


Figure 1: Example sprint burn down chart

### 14.2.8 INDIVIDUAL STATUS REPORTS

The status reports will contain the spring backlog, product backlog, burndown charts, and changes we have implemented from the original plan. Each individual will do a peer review and it will be reported after each spring presentation.

### 14.2.9 ENGINEERING NOTEBOOKS

The engineering notebook will be updated on a weekly basis by each individual team member. The minimum amount of pages that must be completed for each interval is 1 page and each interval is 1 week. Team members will keep each other accountable by ensuring the work is done before it goes into the engineering notebook. Each team member can have a family member or friend sign off as a witness for each ENB page

## 14.3 CLOSEOUT MATERIALS

### 14.3.1 SYSTEM PROTOTYPE

The final system prototype will be demonstrated in December of 2022. It will include the landing page, login/registration page, Page to create/start polls, page to join a poll, poll verification, show polls as badges, Page to place votes, and page to end a poll. There will be a prototype acceptance test with our customer that will occur in person on-site. Nothing will be demonstrated off-site.

### 14.3.2 PROJECT POSTER

The poster will include images of all the different pages available on the app. It will include some documentation of the application and ways to install the application. The poster will be 36 X 48 with 12-inch panels on the left and right and 24-inch panels in the middle.

### 14.3.3 WEB PAGE

The web page will include explanation of the utility of the application and a quick demonstration video of how to use it. It will include information on how to report bugs and also ways to get in touch with the developers. This web page will not public and it will be delivered in December. As for updating the web page it will be provided at closeout meaning no updates will be provided throughout the process.

### 14.3.4 DEMO VIDEO

The video will be 3-5 minutes and include a demo of a user logging in, creating an account and voting, and also we will show what the results of what an election look like.

#### **14.3.5 SOURCE CODE**

Our source code will be maintained through GitHub. We will maintain our changes with branches for each major task. We will not have the project be open source and when we turn the code over to the customer, they can simply download the source code repository with the information on how to run the program.

#### **14.3.6 SOURCE CODE DOCUMENTATION**

We will create a README.md with all of the information required to understand the program, how to build and run it, any inputs/outputs, etc. We won't use any generation tools, and our final documentation format will be PDF.

#### **14.3.7 INSTALLATION SCRIPTS**

The user will be provided with the source code and instructions on how to install the application. The combination of those two should be more than enough to install the application.

#### **14.3.8 USER MANUAL**

The user will not require a user manual however they will be given step by step instructions on how the application should be used, similar to a guide and also a video explanation will accompany that just in case a video is easier to watch than to read for the user.

## REFERENCES

- [1] Democracy.Earth. Universal Basic Income.
- [2] followmyvote. Secure Dapp.
- [3] Voatz. Elections, simplified.
- [4] Votem. Elections made easy.
- [5] VoxVote. Easy Mobile Voting for any Speaker or Teacher.