

# ***Software Engineering Software Requirements Specification (SRS) Document***

**Currently In Congress**

**02/20/2024**

**V1.0**

**By: Garrett Emerich, Mitch Haws, Victor Telles, and Ivan Ramos  
Candelero**

**Our words and actions will reflect Academic Integrity.  
We will not cheat or lie or steal in academic matters.  
We will promote integrity in the UNCG community.**

# Table of Contents

1.	Introduction	4
1.1.	Purpose	4
1.2.	Document Conventions	4
1.3.	Definitions, Acronyms, and Abbreviations	4
1.4.	Intended Audience	5
1.5.	Project Scope	5
1.6.	Technology Challenges	5
1.7.	References	5
2.	General Description	6
2.1.	Product Features	6
2.2.	User Class and Characteristics	6
2.3.	Operating Environment	6
2.4.	Constraints	6
2.5.	Assumptions and Dependencies	6
3.	Functional Requirements	7
3.1.	Primary	7
3.2.	Secondary	7
3.3.	Use-Case Model	7
3.3.1.	Use-Case Model Diagram	7
3.3.2.	Use-Case Model Descriptions	7
3.3.2.1.	Actor: Congressional Representative (Garrett Emerich)	7
3.3.2.2.	Actor: Admin (Victor Telles)	8
3.3.2.3.	Actor: Basic Users (Mitch Haws)	8
3.3.2.4.	Actor: Moderator (Ivan Ramos Candelero)	8
3.3.3.	Use-Case Model Scenarios	8
3.3.3.1.	Actor: Congressional Representative (Garrett Emerich)	8
3.3.3.2.	Actor: Admin (Victor Telles)	8
3.3.3.3.	Actor: Basic Users (Mitch Haws)	9
3.3.3.4.	Actor: Moderator (Ivan Ramos Candelero)	9
4.	Technical Requirements	10
4.1.	Interface Requirements	10
4.1.1.	User Interfaces	10
4.1.2.	Hardware Interfaces	10
4.1.3.	Communications Interfaces	10
4.1.4.	Software Interfaces	10

5.	Non-Functional Requirements	11
5.1.	Performance Requirements	11
5.2.	Safety Requirements	11
5.3.	Security Requirements	11
5.4.	Software Quality Attributes	11
5.4.1.	Availability	11
5.4.2.	Correctness	11
5.4.3.	Maintainability	11
5.4.4.	Reusability	11
5.4.5.	Portability	11
5.5.	Process Requirements	11
5.5.1.	Development Process Used	11
5.5.2.	Time Constraints	11
5.5.3.	Cost and Delivery Date	11
5.6.	Other Requirements	11
6.	Design Documents	12
6.1.	Software Architecture	12
6.2.	High-Level Database Schema	12
6.3.	Software Design	12
6.3.1.	State Machine Diagram: Actor Name (Responsible Team Member)	12
6.3.2.	State Machine Diagram: Actor Name (Responsible Team Member)	12
6.3.3.	State Machine Diagram: Actor Name (Responsible Team Member)	12
6.4.	UML Class Diagram	12
7.	Scenario	12
7.1.	Brief Written Scenario with Screenshots	12

# 1. Introduction

## 1.1. Purpose

Currently In Congress (CIC) aims to provide current information to the general public regarding the day-to-day proceedings of the current (117th) sitting U.S. Congress. Our goal is to combat misinformation in the 21st-century news landscape by offering transparent and unbiased coverage. We will also facilitate direct interaction between users and Congressmen, allowing Congressmen to gauge public opinion about their decisions. As we approach an important general election year, we believe this project is particularly relevant and essential for fostering informed civic participation.

## 1.2. Document Conventions

The purpose of this Software Requirements Document (SRD) is to describe the user-view and developer-view requirements for the Currently In Congress Web Discussion Board. User-oriented requirements describe the system from the user's perspective. In it, we will detail both user-oriented and developer-oriented requirements. From the user's perspective, the document will provide insights into how the system is intended to operate, highlighting the variety of user types the system caters to. Meanwhile, from a developer's viewpoint, it will delve into the technical specifics of the system, covering aspects such as functionality, data management, storage/access protocols, and other critical requirements. This comprehensive approach ensures a clear understanding of the system's objectives and functionalities for all involved parties.

## 1.3. Definitions, Acronyms, and Abbreviations

Java	A programming language originally developed by James Gosling at Sun Microsystems. We will be using this language to build the Restaurant Manager.
.HTML	Hypertext Markup Language. This is the code that will be used to structure and design the web application and its content.
SpringBoot	An open-source Java-based framework used to create a micro Service. This will be used to create and run our application.
MVC	Model-View-Controller. This is the architectural pattern that will be used to implement our system.
Spring Web	Will be used to build our web application by using Spring MVC. This is one of the dependencies of our system.
Thymeleaf	A modern server-side Java template engine for our web environment. This is one of the dependencies of our system.
API	Application Programming Interface. This will be used to get up-to-date congressional information and display it for users to easily access and comment on.

## 1.4. Intended Audience

This document is intended to outline the basic use cases for all users (U.S. Voters) in the first 2 sections. The remaining sections are intended to outline the design and functionality that will need to be implemented for our development team to refer back to as needed.

## 1.5. Project Scope

The software project aims to address the challenge of misinformation within the contemporary news environment by delivering transparent and impartial coverage of the U.S. Congress's day-to-day proceedings to all American citizens. Additionally, it facilitates direct interaction between constituents and their congressional representatives, enabling constituents to offer feedback directly to their representatives. These objectives are in line with our overarching objectives of providing clear information during a pivotal general election year and fostering active civic engagement.

The benefits of the project to voters include:

- **Access to Clear and Unbiased Information:** The project provides voters with transparent and unbiased insights into congressional activities amidst the complexities of the 24-hour news cycle.
- **Enhanced Voter Knowledge and Participation:** By increasing awareness and understanding of congressional proceedings, the project aims to boost voter knowledge and encourage higher turnout in the 2024 general election.
- **Direct Communication with Representatives:** Through a direct line of communication, constituents can offer feedback to their representatives, ensuring their voices are heard and promoting representation in Congress.

## 1.6. Technology Challenges

N/A as of 2/20/2024

## 1.7. References

N/A as of 2/20/2024

## **2. General Description**

### **2.1. Product Features**

The product features include the ability to provide a live feed of congressional proceedings such as voting and debates, as well as a large repository of information on congressional activities, legislation, and members. The purpose of this is to keep users informed in real-time and to provide information on past events leading up to recent ones. Another feature is the forums/discussion boards, where users can discuss and comment on congressional matters. The discussion board functionality helps users engage with the community and the public on legislative issues.

### **2.2. User Class and Characteristics**

Our website application is aimed at the general public, where interest levels in politics and congressional activities may vary widely. Users may have minimal to moderate knowledge of how the U.S. Congress works. Access to the website is simplified by providing understandable information on congressional proceedings, legislation, and how they affect the general public and personal interests.

### **2.3. Operating Environment**

This application is going to be a web-based platform and will be accessible through any web browser. This ensures access to the application on many devices without the need to download additional software.

### **2.4. Constraints**

Currently, design constraints we will be facing will include the selection of the database for storage as well as designing the back-end API to facilitate user interactions with the applications.

### **2.5. Assumptions and Dependencies**

Our software will be dependent on Spring Web and Thymeleaf in order to create and implement an MVC architecture. The application will depend on the use of the ProPublica Congress API (<https://projects.propublica.org/api-docs/congress-api/>) to gather information that will be displayed in the application, this includes bill data, member data, voting records, and congressional committees. The application will also depend on a back-end API that our team will be developing in order to facilitate user interfacing including signing up, logging in/out, writing/reading comments, and voting on polls.

### 3. Functional Requirements

#### 3.1. Primary

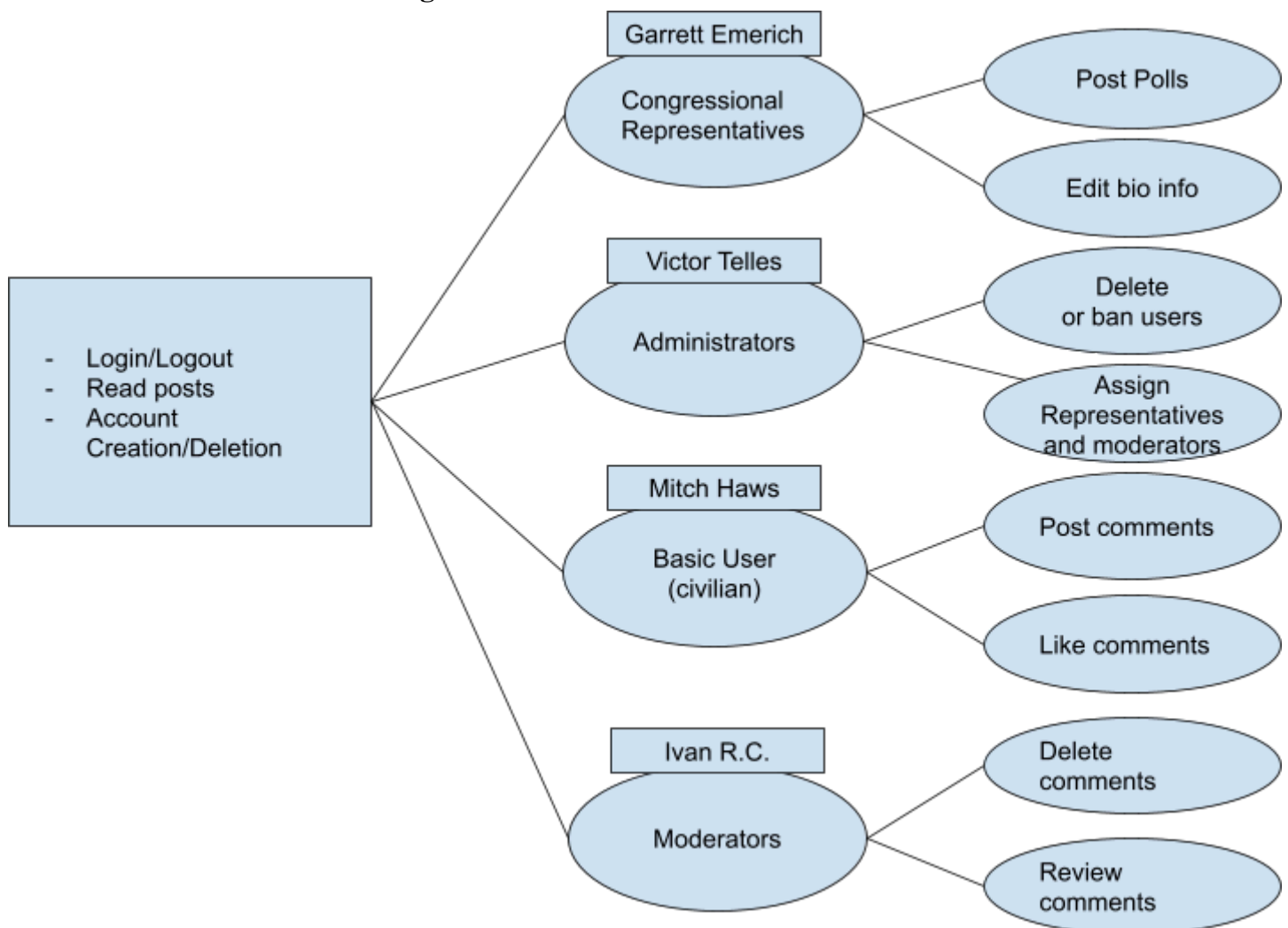
- FR0: The system will allow the user to like, reply to, and post comments.
- FR1: The system will allow the user to supervise comments in which they can review comments that violate the code of conduct with the ability to delete those that do so.
- FR2: The system will allow the user to create/edit their biography in their profile
- FR3: The system will allow the user the ability to create and post polls.
- FR4: The system will allow the user to assign moderators and ban/delete users for not following the code of conduct.
- 

#### 3.2. Secondary

- Account verification to recognize the type of account the user has and that the user resides in the United States Of America.

#### 3.3. Use-Case Model

##### 3.3.1. Use-Case Model Diagram



##### 3.3.2. Use-Case Model Descriptions

###### 3.3.2.1. Actor: Congressional Representative (Garrett Emerich)

- **Post Poll:** The Congressional Representative can create and post polls in which the basic user can respond to the post voting, leaving a comment.
- **Edit Biography Information :** The Congressional Representative can edit their biography in which they can put their information for voters to see their political values.

#### 3.3.2.2. Actor: Admin (Victor Telles)

- **Assign Mods and Representatives:** The admin can assign users to be moderators to protect and moderate the website from actions that violate the code of conduct. Administrators will also be responsible for verifying and assigning users as Congressional Representatives.
- **Delete Users:** Admins can delete/ban users that violate the code of conduct. Admins will review these reports that were produced by users .

#### 3.3.2.3. Actor: Basic Users (Civilians) (Mitch Haws)

- **Post/Reply to Comments:** Basics users can post and reply comments to the website.
- **Like Posts and Comments:** The user can interact to post/liking comments on posts that are posted in the website.

#### 3.3.2.4. Actor: Moderator (Ivan Ramos Candelero)

- **Delete Comments:** The user can delete comments that report is found because it violates the code of conduct of the website.
- **Review Comments:** The user can review comments that were reported by other users. They can determine if the comments follow the code of conduct.

### 3.3.3. Use-Case Model Scenarios

#### 3.3.3.1. Actor: Congressional Representative (Garrett)

- **Use-Case Name:** Post Polls
  - **Initial Assumption:** Congressional representative's have access to the web app and the congressional representative role.
  - **Normal:** The representative will be able to post a poll with a question, multiple options for answers, a start date/time, and end date/time.
  - **What Can Go Wrong:** The representative could attempt to post a poll with an invalid time frame, the application should detect this and not allow it to be posted.
  - **Other Activities:** N/A
  - **System State on Completion:** The poll gets posted to the representatives page and users can respond as they wish.
- **Use-Case Name:** Edit Biography
  - **Initial Assumption:** Congressional representative's have access to the web app and the congressional representative role.
  - **Normal:** The representative can edit their biography with their personal information and whatever information they would like to share about their political standing/goals.
  - **What Can Go Wrong:** N/A
  - **Other Activities:** N/A
  - **System State on Completion:** The representative's biography is updated to show the changes they made and available to be seen by other users.

#### 3.3.3.2. Actor: Admin (Victor Telles)

- **Use-Case Name:** Assign Moderators
  - **Initial Assumption:** Admin will assign users as moderators to inspect actions of users.
  - **Normal:** The assigned moderators will check that there is no violation in the code of conduct such as inappropriate language or anything vulgar offending other users.
  - **What Can Go Wrong:** Information can be lost due to the removal of a post or replies before or after the offense was made until the moderator takes care of it.
  - **Other Activities:** N/A



- **System State on Completion:** Case number of violation will be sent to Admin to not accept the user back into the system.
- **Use-Case Name:** Assigning Congressional Representatives
  - **Initial Assumption:** Assigns representatives to the database and removes the ones going out on term.
  - **Normal:** Will be adding new representatives and deleting previous while keeping comments and information from when that candidate was in term
  - **What Can Go Wrong:** N/A
  - **Other Activities:** N/A
  - **System State on Completion:** There will be a confirmation number for each individual member along with term start and end.

#### 3.3.3.3. Actor: Basic Users (Civilians) (Mitch Haws)

- **Use-Case Name:** Post/Reply Comments
  - **Initial Assumption:** Civilian has access to the web app and the basic user role.
  - **Normal:** Civilian users can post comments and reply to other posted comments.
  - **What Can Go Wrong:** The civilian may attempt to reply to a comment that has been deleted. The system should throw up an error when this occurs.
  - **Other Activities:** N/A
  - **System State on Completion:** Comments will appear under the original post/poll for all users to see.
- **Use-Case Name:** Comment Interaction
  - **Initial Assumption:** Civilian has access to the web app and the basic user role.
  - **Normal:** Civilian users can interact (like) comments.
  - **What Can Go Wrong:** N/A
  - **Other Activities:** N/A
  - **System State on Completion:** Number of likes will appear next to comment or post for all users to see.

#### 3.3.3.4. Actor: Moderator (Ivan R.C.)

- **Use-Case Name:** Reviewing Comments
  - **Initial Assumption:** Moderators have access to the web app and the moderator role.
  - **Normal:** Moderators can review reported comments which violate the code of conduct.
  - **What Can Go Wrong:** The reported comments do not appear on the moderator's "Reported Comment" inbox.
  - **Other Activities:** The moderators can reject the comment review, leaving no change to the original comment.
  - **System State on Completion:** The moderator can access the inbox in which the moderators can review the reported comments.
- **Use-Case Name:** Delete Comments
  - **Initial Assumption:** Moderators have access to the web app and the moderator role. The moderators have access to the "Reported Comment" inbox in which they can see the comments that were reported by users.
  - **Normal:** The moderators can delete comments that violate the code of conduct.
  - **What Can Go Wrong:** The moderators can have a problem deleting the comment which can give an error to the moderators.
  - **Other Activities:** N/A

- **System State on Completion:** The moderators have buttons below the comment to decide if the comment needs to be deleted or not to delete the comment.

## **4. Technical Requirements**

### **4.1. Interface Requirements**

#### **4.1.1. User Interfaces**

The home screen will feature a clean, minimalistic design with a navigation bar at the top, including links to key sections such as "Current Proceedings", "Legislation", "Your Representatives", "Discussion Forum", and "Educational Resources". Each page will have associated data displayed and allow for further user interaction.

#### **4.1.2. Hardware Interfaces**

The SpringBoot application will need to run on a server to keep the application online and running at all times, the web application will interact with this server as the Controller. The web application will run on any hardware device that has access to the internet, the ability to display webpages, and the ability to interact with web pages. This includes, but is not limited to, smartphones, tablets, desktop computers, and laptops.

#### **4.1.3. Communications Interfaces**

Each user's browser must be able to communicate over the internet to the server hosting the Controller over HTTP/DNS. The controller will also need to be able to communicate with the API.

#### **4.1.4. Software Interfaces**

We will also use Spring Boot with Java to connect the frontend to the backend. The frontend will be built using SpringBoot with Thymeleaf. The database management system has not been selected as of 02/20/2024.

## **5. Non-Functional Requirements**

### **5.1. Performance Requirements**

- NFR0(R): The system will allow the user to like, reply to, and post comments within 5 seconds.
- NFR1(R): The system will allow the user to review comments for violating the code of conduct within 30 seconds of a comment being posted.
- NFR2(R): The system will allow the user to access their biography and make updates within 5 minutes.
- NFR3(R): The system will allow the user the ability to create and post polls within 5 minutes,
- NFR4(R): The system will allow the user to assign moderators and ban/delete users for not following the code of conduct within 2 minutes.
- 

### **5.2. Safety Requirements**

- N/A

### **5.3. Security Requirements**

- NFR5(R): The system will only be accessible by users who have completed the sign-up process.
- NFR6(R): The system will only allow verified Congressional members to register as such.

### **5.4. Software Quality Attributes**

#### **5.4.1. Availability**

The application should stay online, with minimal downtime for updates, to allow users to access it any time.

#### **5.4.2. Correctness**

The application will only allow for parent posts to be made directly from the ProPublica Congress API to ensure correctness and integrity.

#### **5.4.3. Maintainability**

Administrators and Moderators should be able to maintain users/comments as well as approve congressional representatives.

#### **5.4.4. Reusability**

The application will post up-to-date data provided by the ProPublica Congress API to ensure users can re-use the application as Congressional proceedings occur.

#### **5.4.5. Portability**

The application will run on any device that has access to the internet, the ability to display webpages, and the ability to interact with web pages. This includes, but is not limited to, smartphones, tablets, desktop computers, and laptops.

### **5.5. Process Requirements**

#### **5.5.1. Development Process Used**

The application will be developed using the waterfall model.

#### **5.5.2. Time Constraints**

The application must be completed by April 30 and Documentation must be completed by May 1

#### **5.5.3. Cost and Delivery Date**

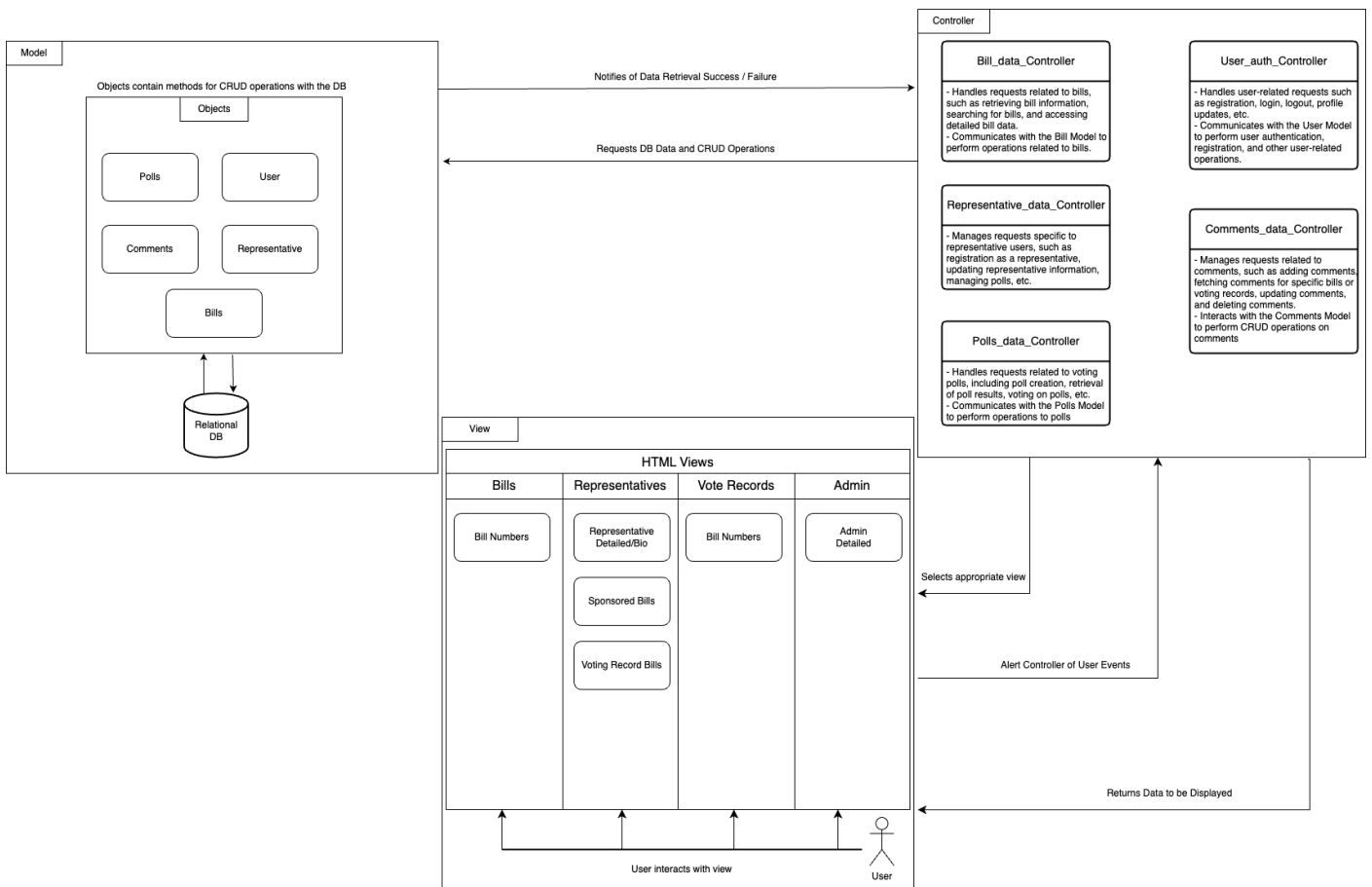
The application delivery date is 4/30/2024, the application should have zero associated costs.

### **5.6. Other Requirements**

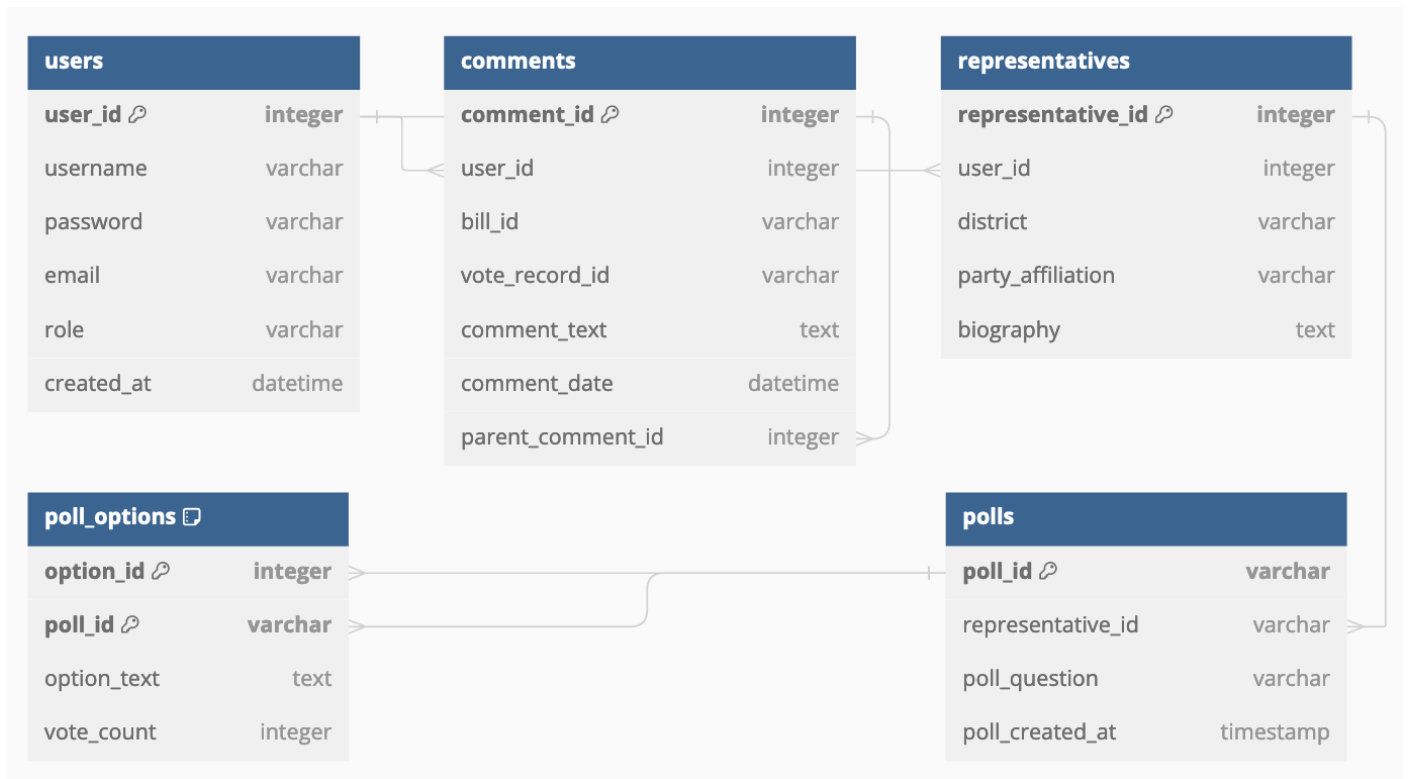
N/A

## **6. Design Documents**

### **6.1. Software Architecture**

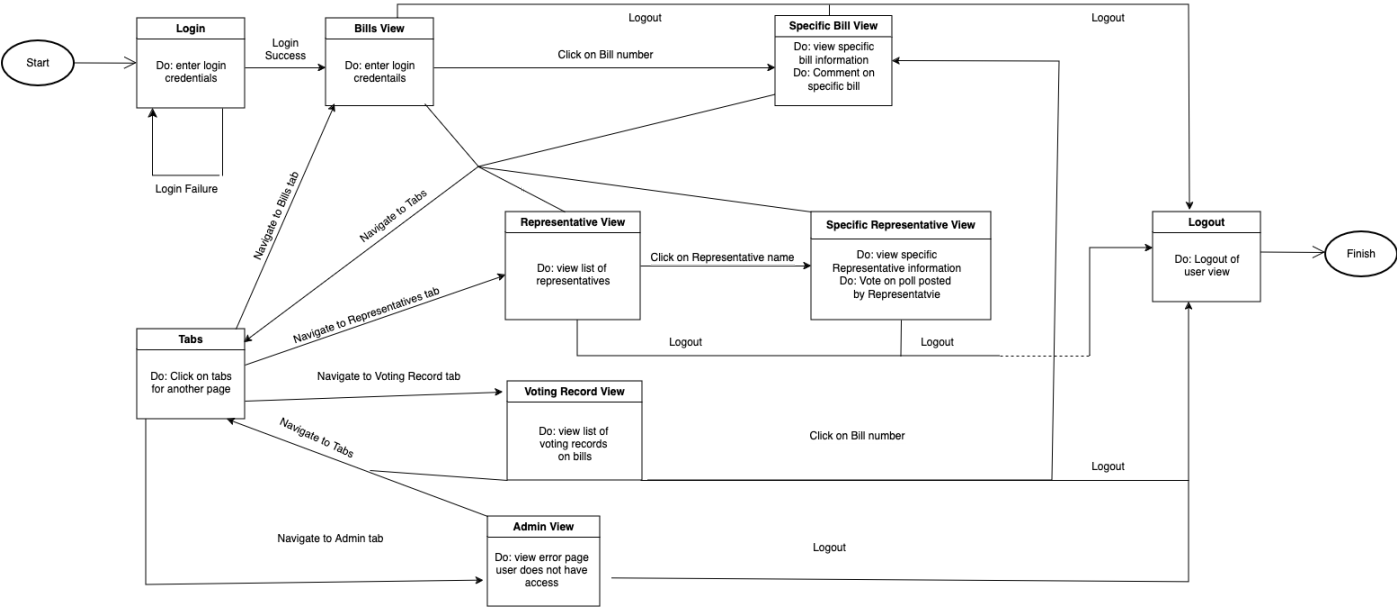


## 6.2. High-Level Database Schema

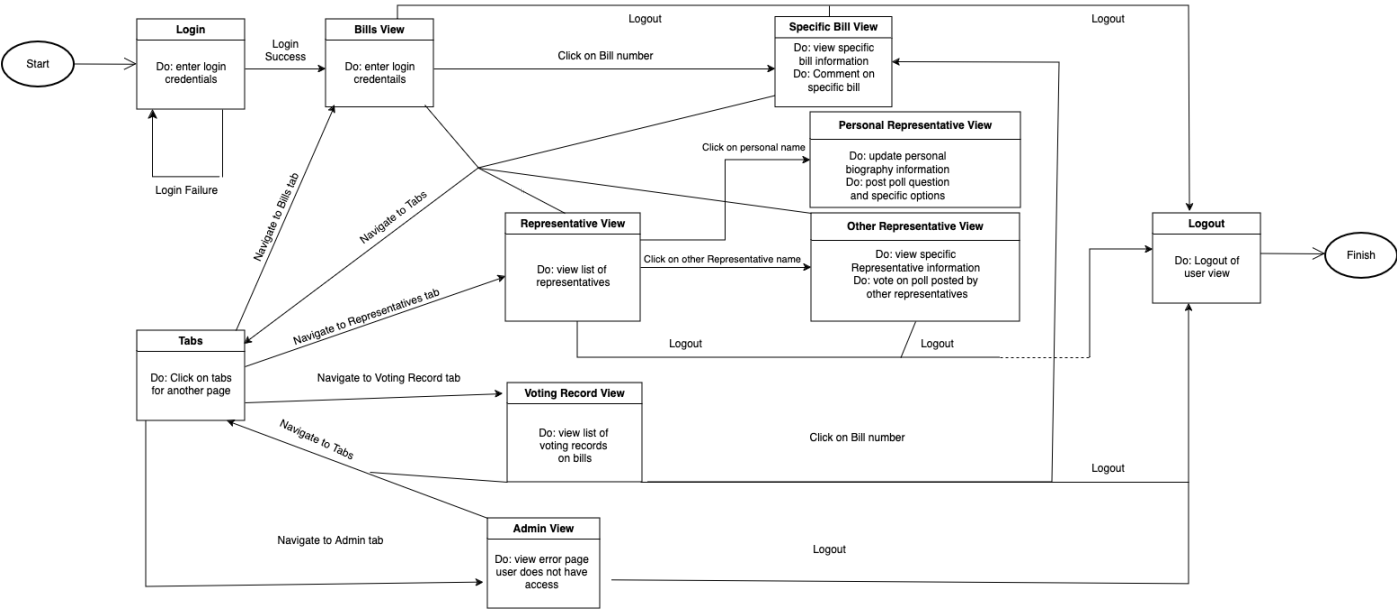


6.3. Software Design

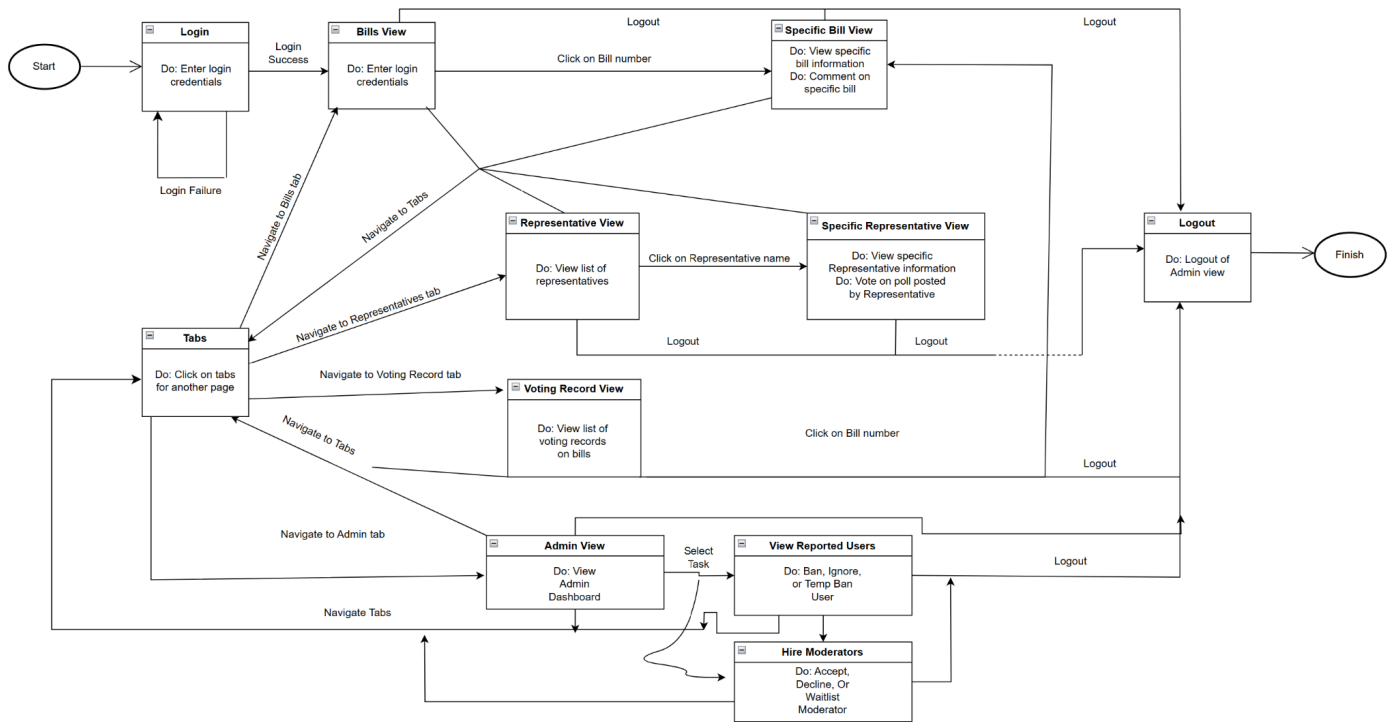
6.3.1. State Machine Diagram: User (Mitch Haws)



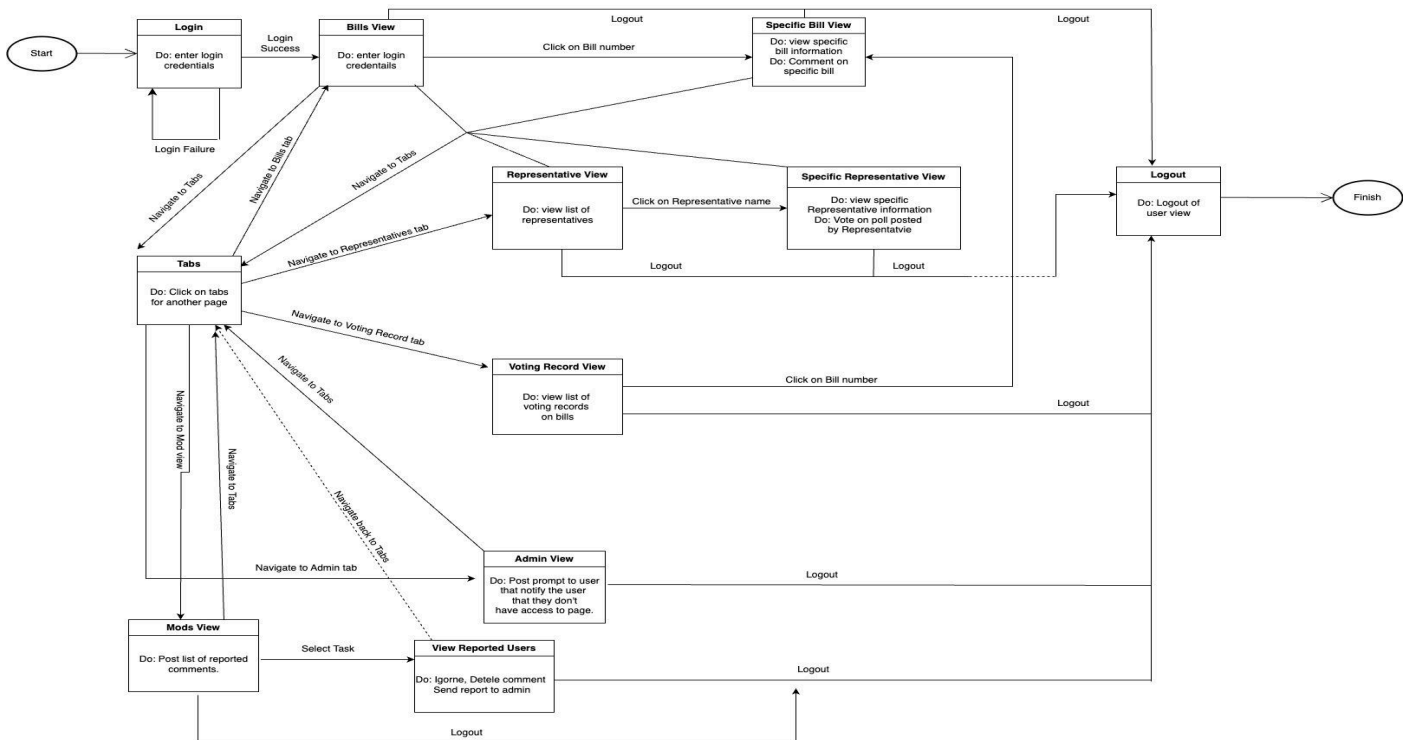
6.3.2. State Machine Diagram: Representative (Garrett Emerich)



### 6.3.3. State Machine Diagram: Admin (Victor Telles)



### 6.3.4. State Machine Diagram: Moderator (Ivan R.C.)



6.4. UML Class Diagram











## **7. Scenario**

### **7.1. Brief Written Scenario with Screenshots**