

INDY-6 Represent Me
Software Requirements Specification
CS 4850 – Section 01/02 – Spring 2024
Feb 13, 2024
Sharon Perry

		
Imani Allen	Morgan Stafford	Dariel Alejos

Table of Contents

1.0	Introduction	3
1.1	Overview	3
1.2	Project Goals	3
1.4	Assumptions.....	3
2.0	Design Constraints	4
3.0	Functional Requirements.....	4
4.0	Non-Functional Requirements.....	4
5.0	External Interface Requirements	5
5.1	User Interface Requirements.....	5
5.3	Software Interface Requirements.....	5

1.0 Introduction

1.1 Overview

Hundreds of bills are proposed in the United States, and it is up to the legislature to determine which bills and policies are passed and which are dropped. The people who make up the legislature of the United States are elected by citizens of the United States and their sole purpose is to represent the ideals and political stances of the people. However, because there are hundreds of bills and policies suggested each day and voted on, the average American citizen does not know what their specific representative stands for. Most Americans know the general stances of the major political parties but that is not enough to make an informed vote that accurately represents their own political stances.

RepresentMe is an app developed by American citizens for American citizens. RepresentMe organizes all US representatives based on the policies and bills they may or may not support. By sorting the representatives by their stances rather than their political party, it eliminates bias and helps voters form informed votes based on people who also do the same things that they do.

1.2 Project Goals

This app aims to organize all the voting data of the proposed bills and policies and display the results in an easy-to-understand platform that allows users to quickly reference and know what bills their representative voted for or rejected. This app will also allow users to get to know their potential representatives with the representative profile page that highlights their political parties, the state they represent (if applicable), and all the policies they have rejected or supported.

1.4 Assumptions

Technical Assumptions: The software assumes that the user has a compatible operating system with the latest android updates.

Operational Assumptions: It is assumed that the users have basic computer literacy thus no training module or user tutorial will be included with this application.

Business Assumptions: It also assumed that users would have internet connection so that cloud-based services can be used with this app.

Business Assumption: This app relies on an API provided by a US government-owned website. So, this website must be working for the app to provide real-time updates and up to date for the information displayed on the app to be correct and relevant.

2.0 Design Constraints

Resource Constraint: This application must be completed with free resources and by May 2024, reducing the number of features that could be implemented for this app.

Technical Constraints: All information is provided by the API provided by the government site so our code and app must be combatable to be able to get and post the API information.

Performance Constraint: This app must support multiple users at a time and be able to perform concurrent requests in an adequate amount of time under normal operating conditions.

Operational Constraints: The app must support multiple user roles and have a working user authentication and authorization system.

External Constraints: This app handles minimal user data, but it must be secure and comply with data protection laws.

3.0 Functional Requirements

This app requires a dynamic navigation bar that links users to numerous pages within the app and allows them to return to the home page. Each button on the navigation bar must accurately link to the page corresponding to the title of the button.

Each page must load all the required information and images and the page must reflect what was determined in the design documents and by the UX designer.

The applicant must be able to sign up and log in. The sign-up will ask users for data stored and referenced for the login in interaction. The data entered for login must accurately match the data saved for login to be successful. If the login data is incorrect, the application must not allow the user to enter the site and must request different information logged in.

The application must accurately make API calls that gather and organize data as needed and the application must be able to connect the data retrieved by the API to the pages on the app. The information must be displayed as intended by the code and by the design document.

4.0 Non-Functional Requirements

Security: This app must protect all user data collected and adhere to user data protection laws

Compatibility: This application must be compatibility will all current systems of android devices and must be able to update and match any update with android operating systems.

Reliability and Availability: The user must access this application and receive accurate information throughout the day.

Performance: Multiple users should be able to use this app concurrently and send requests concurrently and get responses under 10 seconds on an average day.

Usability: The UI should be user friendly meaning that users should easily be able to navigate through the app and find intended information.

5.0 External Interface Requirements

5.1 User Interface Requirements

The first page upon opening the app will be a login screen, followed by the home page upon successful login or continuing as a guest. The home page defaults to two search boxes (Bill/Representative Search). Searching via the Bill box Tapping on a bill opens to that bill's page of for and against votes, totaled at the top as "X% For / X% Against / X% Abstain," with a list of each representative and their respective votes below: "Yay," "Nay," or "Abstain." On this page there will also be a link that opens the user's default browser to the bill's page on the congress.gov domain. Tapping on a representative will bring you to their profile, which will have their voting history alongside each bill they were present for. The bill's name will have "Yay," "Nay," or "Abstain" beside it. Tapping on these bills will take the user to that bill's page.

On the left-hand side, there will be a hamburger icon that can be tapped to open a context menu containing three options, "Bills," "Representatives," and "User Profile." Bills will take the user to the default main page, with all bills listed. Representatives will take the user to a page listing every representative. Tapping any representative will take the user to their respective page. Opening the user profile will bring up a page containing the users' name, username, city, and state.

5.3 Software Interface Requirements

The software will employ the use of a public API from congress.gov, as well as contain external links to the website for each bill presented. This will require interacting with the user's default browser.