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

		
Imani Allen	Morgan Stafford	Dariel Alejos

Table of Contents

1. Introduction

Represent Me is a mobile app built with React Native. Its main purpose is to gather voting history from recent/all votes taken within the House of Representatives and Senate, using the congress.gov API. Using an API, we will be able to obtain data from the website to check what bill was voted for. Using search methods, the users will use the API to compare the lists and display the list of who voted for or opposed the bill.

On the front end of things, the main page will allow the user to search for representatives and bills. Each representative or bill will have a profile with some important information and links to find out more. Individual users can also have a profile and account; however, the app is fully functional without signing in. We will be storing user information (name, email, password, etc.) for login-in purposes that will be sent to a database to cross-reference when it is time to log in to the website.

Data will be divided by members and assigned unique identifiers (Zip codes) so that the end user can easily look them up. Representatives will also have how often they have voted yay, nay, or abstained listed, but an overall stance should be determined by the user via links to each member's respective website if they exist. All this content will be aggregated and presented on an app where users can create an account to track new votes, receive push notifications, and save specific representative's pages.

1.1. Document Description

This document is set to describe the clear, but flexible software design of the RepersentMe App according to the requirements. It will address the constraints and dependencies that can negatively impact the development process and our strategies to overcome these challenges. Upon reading this document, the team's priorities, goals, concerns, and strategies should be understood.

1.1.1. System Overview

The front end of the software is user focused. Having a User Interface (UI) that is clean and intuitive, yet informative is the top priority. This will be accomplished by categorizing the data that is retrieved. A user's ability to have an account and therefore some aspects of personalization is a main feature of the app. On the backend a database will be required to store user information and likely handle the categorization of bills. The backend has some flexibility as to the system organization until main features are complete.

2. Design Considerations

This section describes many of the issues which need to be addressed or resolved before attempting to devise a complete design solution.

2.1. Assumptions and Dependencies

Considering the teams inexperience with React Native and app development, there are features dependent on first completing other basic functionalities such as:

- The Search function on the main page may contain limited functionality (i.e. presidential candidates only)
- User Sign in and profile
- Congress.gov API
- Database speed and storage
- Possible and/or probable changes in functionality

2.2. General Constraints

As this product is to be completed by April, there is a major time constraint, but there are other constraints as well such as:

- Any limitations placed on the API by congress.gov
- Time required implement the most important aspects of the app
- Availability or volatility of resources
- Memory and other capacity limitations (specifically of our database)
- Verification and validation requirements (testing)
- Other means of addressing quality goals
- Other requirements described in the requirements specification

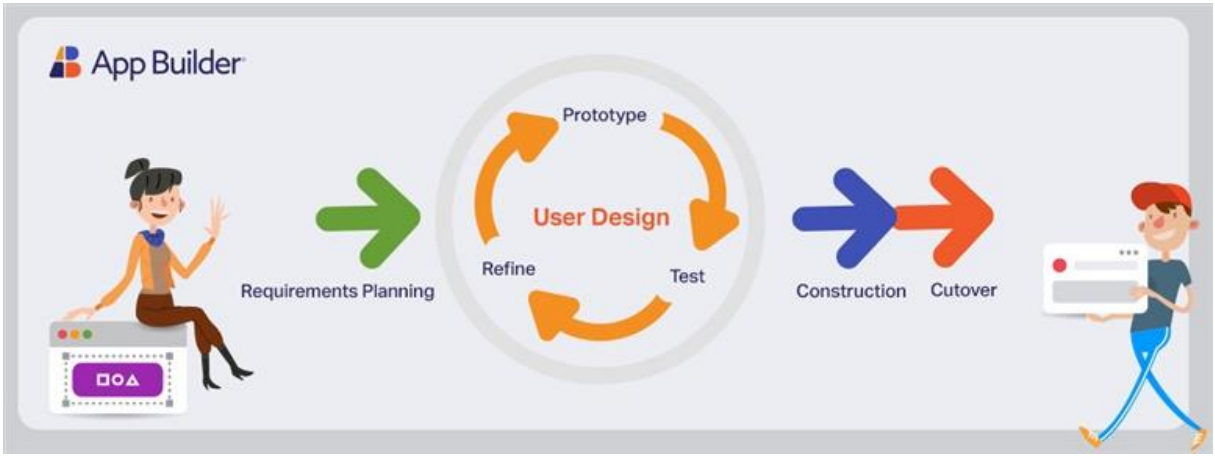
2.3. Goals and Guidelines

The goal of our app is to inform voters, but in addition draw attention towards the candidate's political actions rather than party affiliation. The general population tends to use party affiliation as a heuristic judgment when it comes to voting. The app should enable the public to become less reliant on that method. The design guidelines below help complete this goal.

- The KISS principle ("Keep it simple stupid!")
- Emphasis clean and easily understood UI with most important information
- People like convenience (Limit the need for external links)
- App should be intuitive, don't over complicate it

2.4. Development Methods

Our development approach will likely be the Rapid Application Development (RAD) Model. Each feature that's created will immediately go into testing and be continually refined according to the guidelines, until that phase of features has been completed. Below is a diagram that illustrates the RAD model.



https://www.infragistics.com/community/blogs/b/jason_beres/posts/rapid-application-development

This model is preferable for this product because of the user design phase. As the continuous testing and refining of the product is likely to create the most user-friendly product. The requirements planning is being completed in requirements document and this one.

3. Architectural Strategies

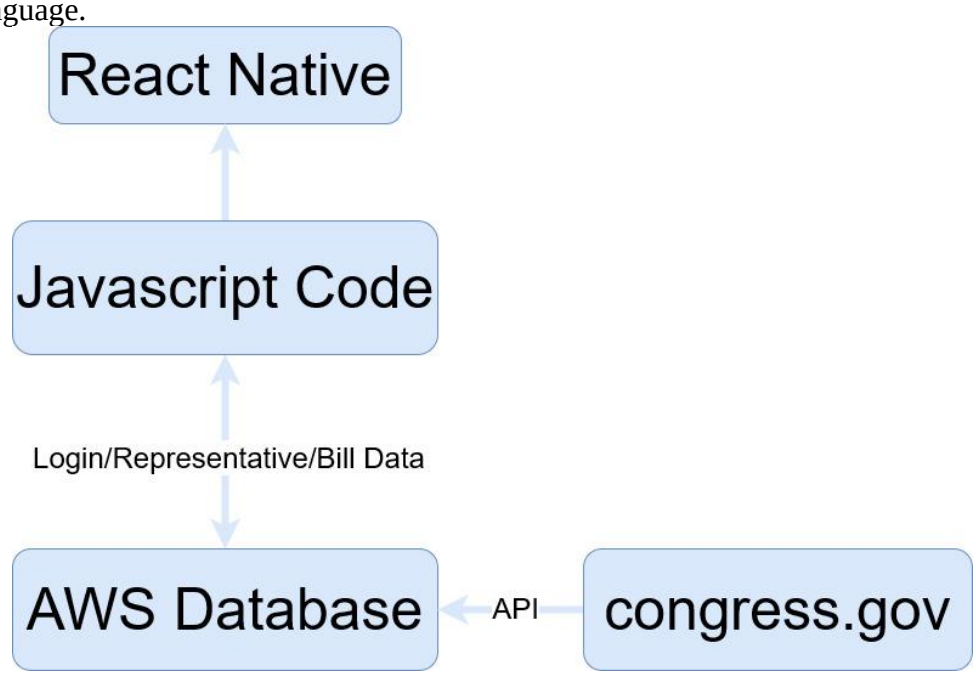
Initially, the app is being developed for Android, however, it is being built in React Native and JavaScript, so it's also compatible with iOS. The first version of the app will be limited, focusing only on presidential candidates and bill searches. Eventually, this will expand to more in-depth Bill profile pages and information for members of Congress and the House of Representatives. The app's furthest improvements include personalization and expanding to state and local levels. Many of these will likely require our database to be expanded as the current scale is minimal.

4. System Architecture

React Native will be used to build the UI of the application. The source code will be written in JavaScript and will communicate with the database on the backend. This database will be from Amazon AWS and will contain user login, bill, and representative information. The database will communicate with a congress.gov API and the source code conducting the API calls. Login will be handled by the source code on the backend but communicate with the same AWS database to verify users.

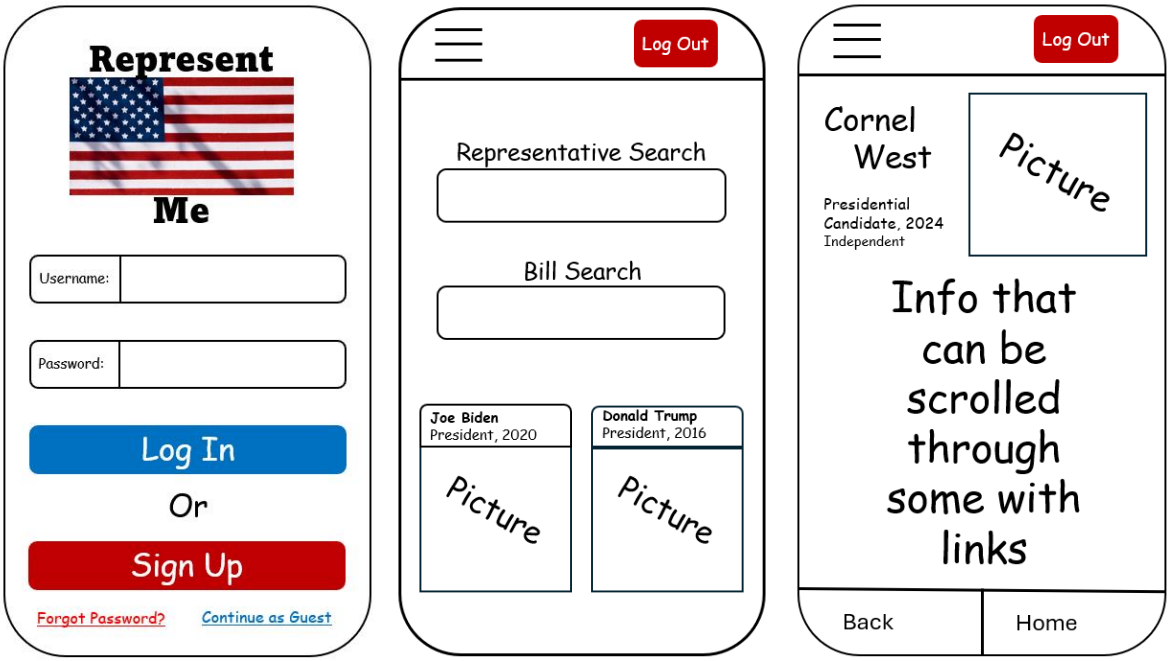
React Native was chosen for flexibility and ease of use, Amazon AWS was chosen for its availability and ease of integration. Congress.gov was chosen for its available information

and API key availability. JavaScript was chosen because of the developer's familiarity with the language.



5. Detailed System Design

The app will have many features, however, below are the first three pages that will be developed. From left to right there is the login screen, main page, and representative profile.



The Login Screen will be the first screen that pops up. Consisting of 4 buttons, 2 text input fields, and the Represent Me Logo. The app is going to have a red, white, and blue theme.

After logging in, the main home page appears. In the top left corner, there will be a hamburger menu and in the top right corner a login/logout button. The button will be red for logout and blue for login. Front and center are 2 search bars on for representatives the other for bills. Lastly, Will be “mini profiles” for 2 prominent representatives; these act as links to their full representative profiles.

The Representative profile page is the same at the top as the main page. The first elements in the Profile are a photo, their name, current or past position with start dates, and last their political affiliation. Below that will be a variety of information that can be scrolled through such as past positions held, if any, and votes by category, etc.

5.1. Resources

A database is necessary for storing user account information. The data gathered for the bills from the API may need to go into the database to be properly categorized and stored.

6. Conclusion

The design of the RepresentMe app and the development process have their challenges, however, using the RAD method we can achieve our goals. Making an app with React Native that has the flexibility to be both Android and iOS compatible as well as expand in the future.