ElectMe!

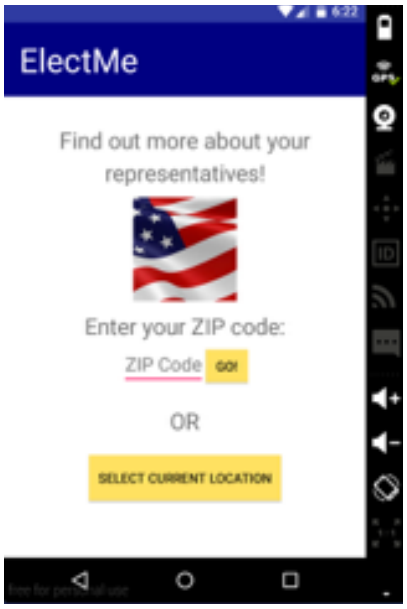
Represent:

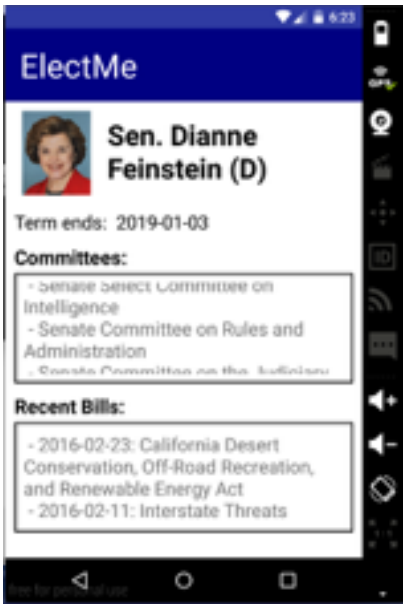
Informational Legislative App for Voters and Constituents

Video Link: https://youtu.be/foLRX_Dp8iQ

GitHub Repository Link:
https://github.com/cs160-sp16/prog-02-represent-vucourtney

How Does ElectMe Work? [Phone Functionality]





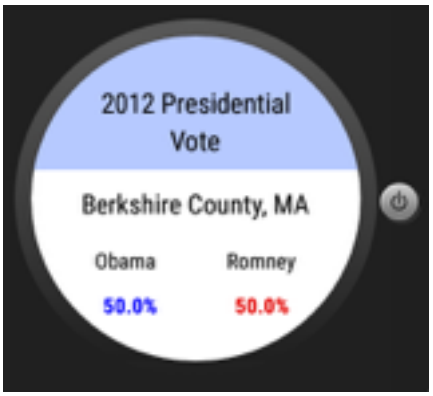
Main Screen (Top Left)
The user can input a Zip Code and tap "Go!", or the user can tap "Select current location," which retrieves the phone's current location using Google's Location API. Using the Google Geocoding API and Sunlight API with this location information, the app generates information that will appear in the congressional view.

Congressional View (Top Center)
Provides a section for each legislator who represents the given location (supports Zip Code ambiguity and can display up to 7 resulting legislators). Each section (color coded based on party) displays the legislator's photo, name, political party, email and website links, and a button to view his or her Twitter timeline. As indicated at the top, clicking a photo will load a detailed view for that legislator.

Detailed View (Top Right)
Loads a detailed view for a legislator (with photo, name, and party).
Using the Sunlight API, the app also displays the end date of the legislator's term as well as the committees he or she serves on and the recent bills he or she has sponsored (with the date of introduction of the bill).

Tweet Screen (Bottom Right)
Clicking the "View latest tweet" button in the congressional view loads the corresponding legislator's Twitter timeline (using Fabric to access Twitter's API). Again displays the legislator's name and party.

How Does ElectMe Work? [Watch Functionality]



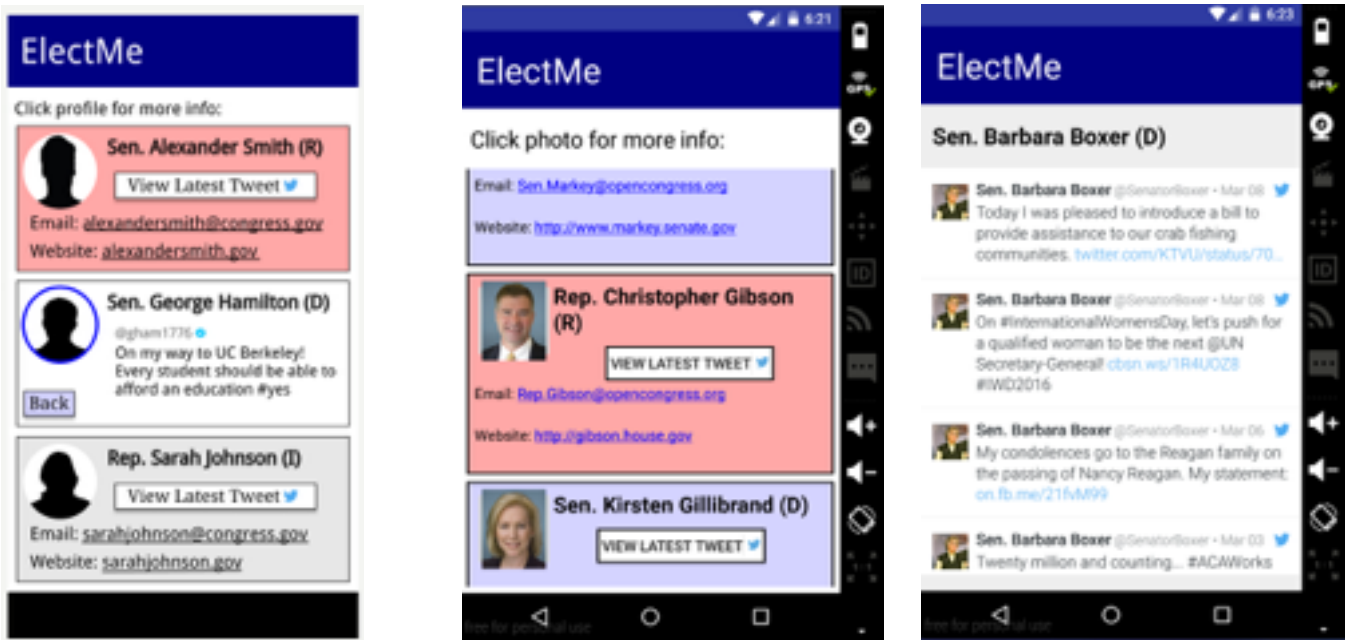
Main Screen (Left)
After the user inputs a location (either via Zip Code or GPS) on the phone and the congressional view loads, the app also starts on the smartwatch. The top part of the watch has a button that leads to 2012 Presidential Vote outcomes for the pertinent county. The bottom features a card for each legislator that was generated, which the user can scroll through by swiping left or right (an indicator for the number of cards appears when swiping or loading, as is visible on the shake screen). Each card features the legislator's name and political party with a flag icon.

2012 Presidential Vote Screen (Center)
If the "2012 Election" button is tapped from the main view, the app retrieves data on the county in which the current location is located. It displays the name of the county and state in addition to the percent of the vote earned by Obama and Romney in the 2012 presidential race.

Watch Shake [Accelerometer] (Right)
If the watch is shaken, the accelerometer will recognize it and signal the phone. The user is notified of the process by a Toast that indicates that the watch has been shaken. Given this signal, the app will generate a random valid location in the United States and use it to display an updated congressional view on the phone and an updated main view on the watch with new legislators and 2012 voting data.

Design Iteration [From Wireframes to Application]

Congressional View Transition to Tweet View



(Before) (After)

On the phone, the main design change I made was in the way to integrate recent legislators' tweets. In my wireframe, clicking the "View latest tweet" button would essentially flip the section for that legislator. Instead of displaying his or her email and website links, it would instead embed his or her latest tweet with a back button to return to the links. When I imagined myself using the app, it made more sense to design this a bit differently, such that clicking the "View latest tweet" button would display a new screen with many recent tweets. This felt less cluttered, would eliminate the need for a back button, and would allow the user to focus on the legislator that was selected. The newly iterated Tweet screen (the rightmost screenshot) still displays the legislator's name and political party, and each Tweet displays the legislator's Twitter photo, so no information was lost with this change.

Watch Main View and Transition to 2012 Vote View

(Before) (After)

When constructing wireframes, I envisioned that the user could see the 2012 presidential vote for the pertinent county after swiping right through all the legislators for the pertinent location (hence the screen in the middle with a notification that all the legislators have been viewed and a button to see the 2012 vote). In creating the app, I maintained the concept of cards that can be swiped through, each of which features a legislator and his or her political party. However, my design in the wireframe jeopardizes user control and freedom—if a user just wants to see the 2012 vote results without looking through all legislators, he or she may become frustrated. This led to a new iteration in which the "2012 Election" button is static across all legislator cards, so the user can see this screen at any time.

Watch Shake Indicator

(Before) (After)

To address the design heuristic of visibility of system status, I wanted to have an indicator that notified the user if the watch had been shaken and legislator information for a random location was being generated. This gives concise verification and feedback: the app recognizes that the user just shook the watch. My first concept was a new screen with this notification, but a new iteration was to simply display a Toast that displays "Shaken." On a smartwatch with very little screen space, simplicity is usually preferable, so displaying a temporary new view seemed excessive. The Toast serves the same purpose in a faster, less confusing manner.