

**PolitiMap:
User Stories, Use Cases, and Requirements
Version 1.1**

Matthew Goodrich

October 6, 2016

Contents

1	User Stories	3
1.1	Informing New Voters	3
1.2	Customer Value	3
1.3	The Bigger Picture	3
2	Unique Use Case	3
2.1	Viewing Federal Bills	3
3	Functional Requirements	5
3.1	Frequent Request Cost	5
3.2	Network Refresh	5
3.3	Default Bill Sorting	5
4	Non-Functional Requirements	5
4.1	Quick Loading	5
4.2	Well-Developed iOS Application	5
4.3	Current Information	5

1 User Stories

1.1 Informing New Voters

As a new voter, I want to be able to view previously-passed bills, so I can understand how politics have evolved and develop context for future bills.

1.2 Customer Value

As a busy voter, I want to be able to quickly view the status of new bills, so I can stay updated with politics, without investing too much of my valuable time.

1.3 The Bigger Picture

As an out-of-state student, I am affected by politics in multiple areas, meaning I want to be able to browse only federal bills and representatives, so I can invest my time well and only view relevant information.

2 Unique Use Case

2.1 Viewing Federal Bills

Use Case ID:	1
Use Case Name:	View Federal Bills
Created By:	Matthew Goodrich
Last Updated By:	Matthew Goodrich
Date Created:	October 6, 2016
Date Last Updated:	October 6, 2016
Actors:	The Application User; The Swift iOS Application; The Backend Server
Description:	The application shall display a list of federal bills and further information about each upon selection.
Preconditions:	<ol style="list-style-type: none">1. The device is connected to the internet.2. The backend is loaded with the organized data.3. The device has the application downloaded, updated, and open to the home screen.4. The backend server is running, whether it be an AWS component or EC2 instance.

Postconditions:	<ol style="list-style-type: none"> 1. A list of federal bills will be shown and likely scrolled-down, or the details of a specific bill will be showing. 2. A request will have been made to the AWS component or EC2 instance. 3. TBD - The federal bills may be cached in the user's device.
Normal Flow:	<ol style="list-style-type: none"> 1. The user chooses to view the list of bills relating to their local, state, and federal governments. 2. A GET request is made to the web server while the user sees a loading animation. 3. The user selects to filter the list of bills to show only federal bills. 4. The user scrolls through the bills and notices an interesting one. 5. The interesting bill is selected and the user reads related information, including a summary, associated representatives, and links to related external documentation.
Alternative Flows:	<ol style="list-style-type: none"> 1. The user selects a federal bills list instead of filtering a combined list. 2. The bills are loaded upon opening the application and only refreshed manually. 3. The bill information does not include associated representatives. 4. The bill shows dynamic data, such as votes and comments.
Exceptions:	<ol style="list-style-type: none"> 1. Unable to load the bills through the lack of an internet connection.
Includes:	<ol style="list-style-type: none"> 1. An external library to make the GET request.
Priority:	High - One of the core functions of the application.
Frequency of Use:	Depends on the number of users and other variables.
Business Rules:	None
Special Requirements:	<ol style="list-style-type: none"> 1. The application requests the data asynchronously. 2. The device is an iPhone 5 or newer. 3. Specific bill information is loaded with a GET request after a bill is selected.

Assumptions::	<ol style="list-style-type: none"> 1. Viewable federal bills exist in the backend. 2. Representatives have been linked to associated bills. 3. Documentation links have been added to associated bills.
Notes and Issues:	None

3 Functional Requirements

3.1 Frequent Request Cost

The system shall store the frequently accessed data, such as the list of federal bills, in AWS S3, to minimize the cost to \$.004 per 10,000 requests.

3.2 Network Refresh

The system shall refresh the list of federal bills from GET requests through a network connection.

3.3 Default Bill Sorting

The system shall default to sorting bills by the timestamp of their last update in content.

4 Non-Functional Requirements

4.1 Quick Loading

The system shall load the list of bills in under two seconds.

4.2 Well-Developed iOS Application

The system shall have zero memory leaks.

4.3 Current Information

Federal bills shall be updated within one day of receiving new information.