CongressStats
for UTM CSCI 352 Spring 2018

Cody Robertson and Joshua Freeman

Abstract

CongressStats is a desktop application for Microsoft Windows that displays statistics such as attendance and voting records for U.S. senators and representatives. The application also displays descriptions and statuses of popular bills. Our application is for use by the average American citizen looking to learn more about how the U.S. government is representing him or her. Currently, we have fleshed out basic project goals as well as a short-term roadmap to implement the application's core features. In this paper, we document our design of CongressStats and the challenges we faced.

1. Introduction

In recent years, the United States has seen record interest in policy and legislation passed by its national government. The FCC's net neutrality policy, for example, reigned in nearly 22 million comments regarding citizens' positions on the issue [1]. Public trust in the government also remains near historic lows [2]. Observing these two trends has generated our inspiration for creating CongressStats, a desktop application for Microsoft Windows.

CongressStats fetches and arranges public data to allow users to view voting histories and attendance records for U.S. senators and representatives. The application also allows users to view descriptions and statuses of popular bills. By empowering U.S. citizens with this wealth of information, voters will be able to stay active in current legislature and be able to see for themselves whether or not they are truly being represented by their elected officials.

1.1. Background

In the United States, national and state governments are split into three branches: executive, legislative, judicial. The legislative branch, or Congress, is split into two groups: the House of Representatives and the Senate. Members of the House are referred to as both representatives and congressmen/congresswomen, which lead to confusion as both parties are technically members of Congress. Members of the Senate are referred to simply as Senators.

1.2. Challenges

An initial challenge for the project has been manually tracking down all public records in XML form hosted by various sites of the U.S. government and automate the fetching, storage, and retrieval of their contents. Once all data sources were successfully discovered, a simple Python script called at the start of each application instance was created to fetch source data and compare updated dates in order to determine if local data needs to be updated. Python proved to be a much better solution than C# to parsing XML due to its loose typing system.

2. Scope

Our goals for CongressStats are to fetch and automatically update biographies and attendance for U.S. senators and representatives. We also want to parse voting records for bills to generate and display statistics for each official. In addition to these statistics, we would like to display information on bills themselves, including bill descriptions, times/dates of voting, and voting results. The project will be complete if we can accomplish the above.

If time permits, we would like to add support for the same statistics for state governments that maintain adequate and public digital records, and we would like to add notifications for the following of specific bills. We would also like to display information about upcoming elections and election cycles.

2.1. Requirements

As part of fleshing out the scope of your requirements, you'll also need to keep in mind both your functional and non-functional requirements. These should be listed, and explained in detail as necessary. Use this area to explain how you gathered these requirements.

Use Case ID	Use Case Name	Primary Actor	Complexity	Priority
1	Find Members of Congress by Location of Representation	User	Low	1
2	Get Detailed Statistics for Members of Congress	User	Med	2
3	Get Detailed Information on Popular Bills	User	Med	2

TABLE 1. USE CASE TABLE

2.1.1. Functional.

- User needs to be able to navigate to all pages of the application.
- User needs to be able to locate relevant information tied to an inputted location.
- User needs to be able to easily find and view detailed statistics for all members of the United States national Congress.
- User needs to be able to easily find and view detailed information regarding all national bills for at least the previous year.

2.1.2. Non-Functional.

- Change in Members of Congress should be updated within 24 hours of change to http://www.house.gov.
- All bills from the current 115th U.S. Congress should be contained within the application, unless waiting on updated information to the bulk data repository.
- Change in Bill Status should be updated within a business week.

2.2. Use Cases

Use Case Number: 1

Use Case Name: Find Members of Congress by Location of Representation

Description: A user of the application is wanting to know what members of Congress represent him or her. This is the process of events the user will engage in in order to accomplish this task.

Below is the basic flow for the process:

1) User selected “Members of Congress” to navigate to the Congress Navigation page.
2) User types in his or her location into the search bar and chooses “search”.
3) User is navigated to a Congress Listing page that displays only the members of Congress who represent the inputted location.

Termination Outcome: The user now has a listing of his or her representing officials.

Use Case Number: 2

Use Case Name: Get Detailed Statistics for Members of Congress

Description: A user of the application is looking for detailed voting records and/or statistics for individual members of Congress. This is the process of events the user will engage in in order to accomplish this goal.

Below is the basic flow for the process:

1) User selected “Members of Congress” to navigate to the Congress Navigation page.
2) User selects either “Browse House” or “Browse Senate” to view a filtered list of members of Congress that matches his/her selection.
3) The user may choose additional filters on this page to further limit the displayed members of Congress.
4) The user may the . select any displayed member of Congress

Termination Outcome: Finally, the user views a detailed information page about the chosen member.

Use Case Number: 3

Use Case Name: Get Detailed Information on Popular Bills

Description: A user of the application is looking for detailed voting records and/or descriptions of popular bills. This is the process of events the user will engage in in order to accomplish this goal.

Below is the basic flow for the process:

1) The user selected “Popular Bills” to navigate to the Popular Bills page.
2) The user can now optionally filter all listed bills by name before selected one of interest from the “New”, “In-Vote”, or “Results” list.

Termination Outcome: The user is redirected to a Bill Profile page which details a description as well as voting records and statuses in House, Senate, or Executive review for the chosen bill.

2.3. Interface Mockups

Users begin with the main menu (Figure 1). For use cases 1 and 2, users will select “Members of Congress” and proceed to the Congress Navigation screen (Figure 2). Once the user has searched by location, selected “Browse Representatives”, or selected “Browse Senators” the user will be redirected to a Congress Listing page (Figure 3). This page details all relevant members of Congress to the filters chosen on the previous and current pages. Once a user has selected any of the members of Congress on the page, they will be directed to a Congressional Profile page, seen in figure 4, detailing a basic description, statistics, and voting records for the selected member of Congress.

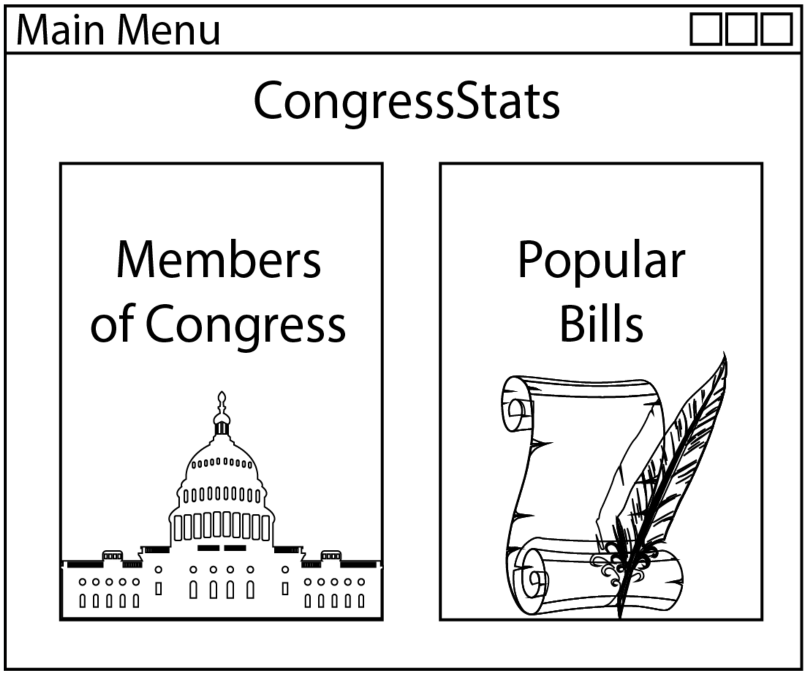


Figure 1. The Starting Screen for CongressStats

For use case 3, users will select “Popular Bills” from the main menu 1 and proceed to the Popular Bills screen (Figure 5). This screen displays bills in ordered lists that display newly introduced bills, bills that are currently in the voting process, and bills that have completed the voting process. When a bill is selected from any of the lists, the user is redirected to a Bill Profile page, seen in figure 6, which contains a short description of the bill itself and its status in House, Senate, and Presidential review.

3. Project Timeline

1) Requirements
   - January 11 - Began Forming Project Requirements
   - February 1 - Finalized Project Requirements
2) Design
   - February 12 - Formalized Use Cases and Initial Interface Mockups
   - February 13 - Presented Proposal to Class
   - February 15 - Negotiated Initial Application Structure
3) Implementation
   - March 10 - Create All Initial Windows and Programmatically Fetch All Congress XML Files
   - March 17 - Parse and Store Congress Data in Access Database, Create UI Placeholders, and Add Navigation Between Pages
   - March 24 - Programmatically Generate and Add Controller Code for Congress Listing and Popular Bills Dropdowns from Model

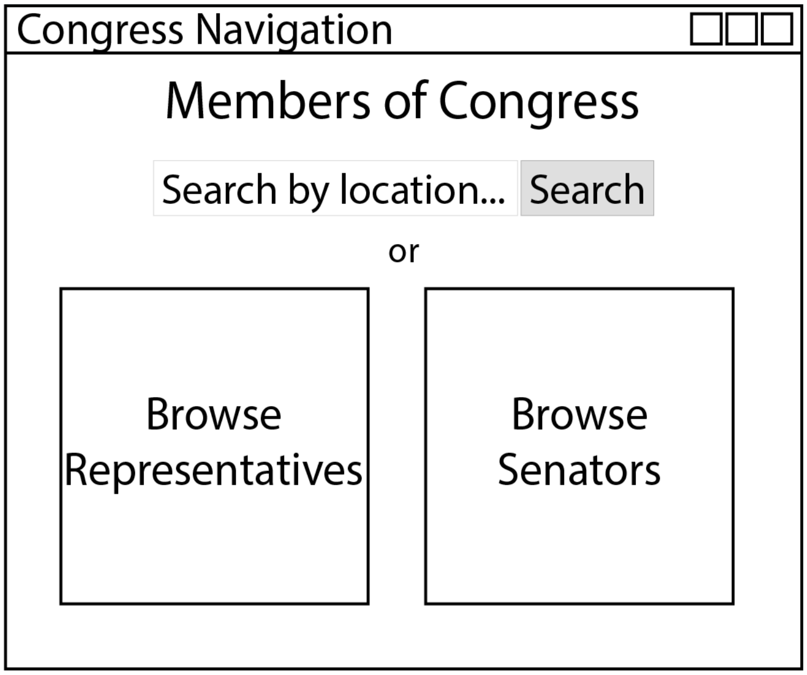


Figure 2. The Congress Navigation Screen for Searching for Members of Congress

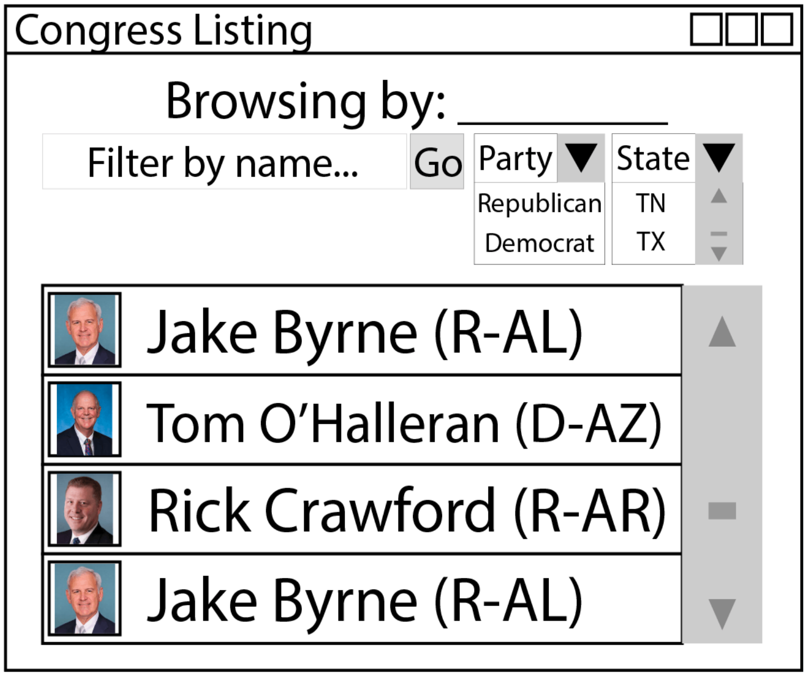
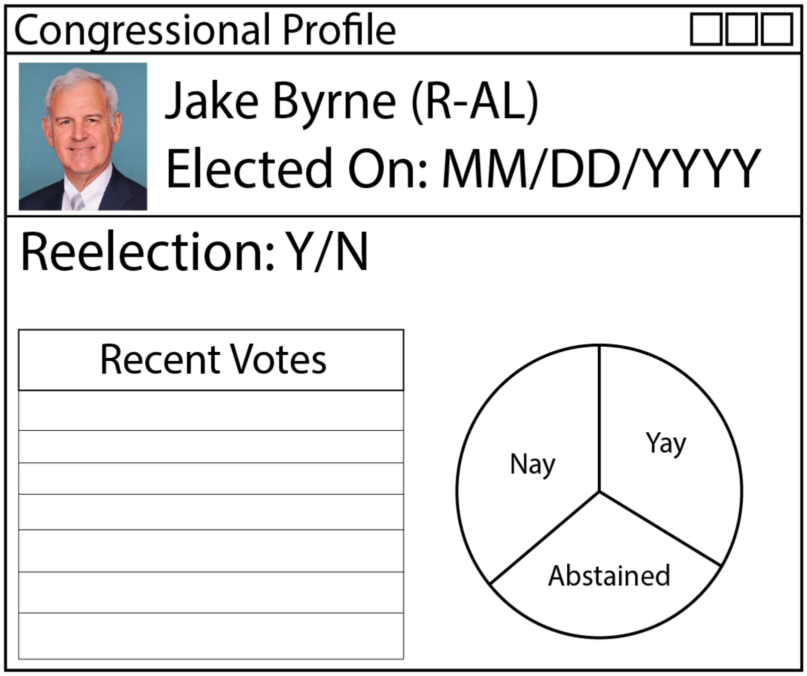


Figure 3. The Congress Listing Screen for Selecting Individual Members of Congress to View More Information About

- March 31 - Programmatically Generate Bill and Congress Profile Pages from Model

4) Verification

- April 11 - Verify Both Software Development Patterns Have Been Adhered to and All Use Cases are Fully Functional
- April 12 - Demo Project in Class



Congressional Profile

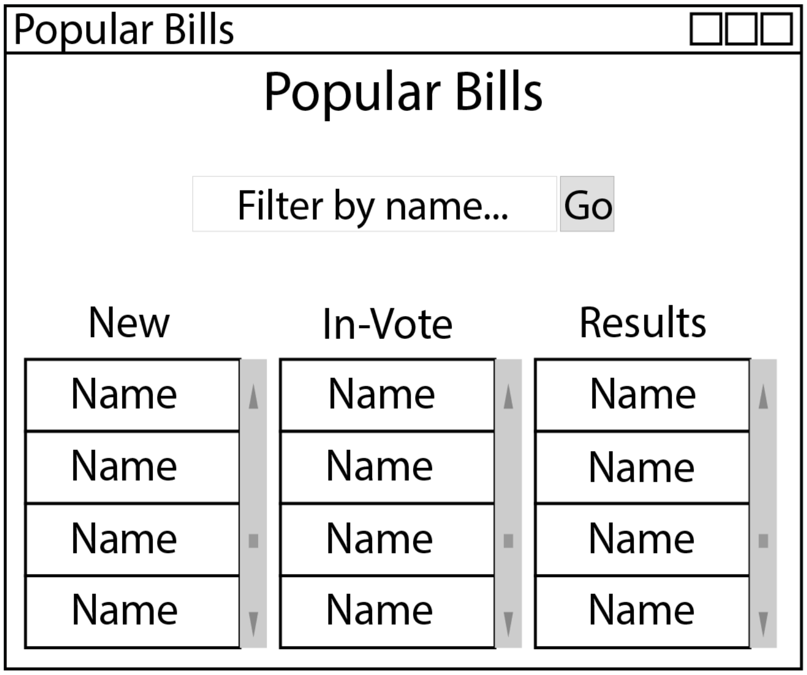
Jake Byrne (R-AL)

Elected On: MM/DD/YYYY

Reelection: Y/N

Recent Votes

Figure 4. The Congress Profile Screen for Viewing Basic Information, Voting Records, and Statistics about an Individual Member of Congress



Popular Bills

Popular Bills

New	In-Vote	Results
Name	Name	Name
Name	Name	Name
Name	Name	Name
Name	Name	Name

Figure 5. The Popular Bills Screen for Displaying Bills that are Newly Introduced, in the Voting Process, or have Recently Completed the Voting Process

- April 21 - Finalize Application and Report
- April 27 - Final Class Presentation

Parsing the government's XML data into the database set us back from our original estimates. We were never able to fully store all data provided by xml.house.gov. This led to the loss of functionality in generating statistics for each bill and congressional member. In addition, the factory classes for generating profile pages were not added until April 26. These setbacks led to the lack of some of our key features in our final project version.

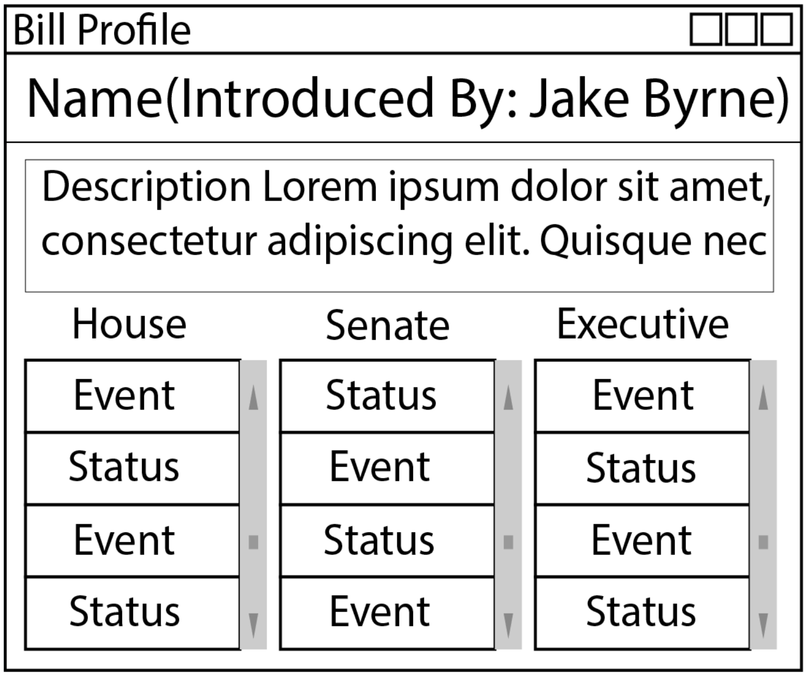

Bill Profile
Name(Introduced By: Jake Byrne)
Description Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque nec
House
Senate
Executive
Event
Status
Event
Status
Status
Event
Status
Event
Event
Status
Event
Status

Figure 6. The Congress Profile Screen for Viewing Basic Information, Voting Records, and Statistics about an Individual Member of Congress

4. Project Structure

This application will have two main pieces. The first will be a script that periodically runs to fetch new XML files from various government sites and parses and stores new data in a application's Access database. The second will be the application that users interact with and displays all stored data from the database in a user-friendly manner.

Currently, each window presented in our previously presented UI markups is created a WPF page consisting of both xaml and xaml.cs files. A grid within our MainWindow contains a navigation manager that sets the grid's contents to the currently viewed page.

Our profile pages are generated from factories. Each UI element displayed on these pages is generated through the use of a decorator. We gather the identifier of the bill or congressional member the user selects and pass it to the constructor of each page factory. The factory is then responsible for gathering necessary data from the database as well as formatting and displaying the data.

4.1. UML Outline

A UML of the programmatically generated pages in our project can be seen in Figure 7. The diagram visualizes how we are implementing the Factory Method design and decorator design patterns into our project through creation of the various pages in our application.

4.2. Design Patterns Used

We have implemented two design patterns in this application. The first of which is the Factory Method pattern. We are creating window factory classes that will generate needed components for the profile pages of our project. We are also using the Decorator pattern. We feature the use of a decorator for each UI element displayed in the profile pages. Each decorator is responsible for styling itself and setting its content appropriately.

5. Results

The application features all bills and members of the 115th U.S. Congress. The current version of our application is not set up to update stored bill status data on start-up. By executing a custom bash script that executes a wget command, unzips the downloaded file, and executes the custom python script for parsing and updating the data stored in the database.

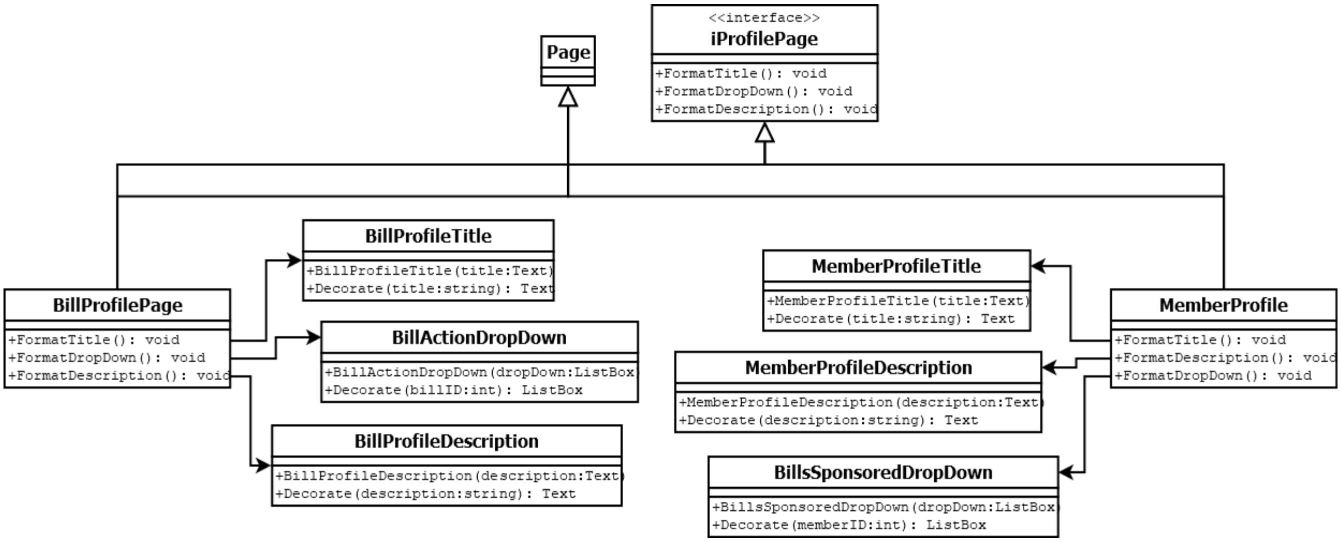

Page
<<interface>>
iProfilePage
+FormatTitle(): void
+FormatDropDown(): void
+FormatDescription(): void
BillProfilePage
+FormatTitle(): void
+FormatDropDown(): void
+FormatDescription(): void
BillProfileTitle
+BillProfileTitle(title:Text)
+Decorate(title:string): Text
BillActionDropDown
+BillActionDropDown(dropDown:ListBox)
+Decorate(billID:int): ListBox
BillProfileDescription
+BillProfileDescription(description:Text)
+Decorate(description:string): Text
MemberProfileTitle
+MemberProfileTitle(title:Text)
+Decorate(title:string): Text
MemberProfile
+FormatTitle(): void
+FormatDescription(): void
+FormatDropDown(): void
MemberProfileDescription
+MemberProfileDescription(description:Text)
+Decorate(description:string): Text
BillsSponsoredDropDown
+BillsSponsoredDropDown(dropDown:ListBox)
+Decorate(memberID:int): ListBox

Figure 7. Basic Project UML

Our application continues to function even without a network connection since a local database of bill and member information is maintained. Accessing member profile pages is currently functional. Since we are missing tables from our database that have not been properly parsed, viewing the bill listing page and custom statistics for both bills and members are not currently functional.

5.1. Future Work

Currently, the most pressing work is to finish implementing the remaining use cases by creating the remaining tables of our database. Afterward, we will be able to finish implementing the remaining features such as automating our python script to update bill information on application start.

The ability to show data from the 113th and 114th U.S. Congresses would also be useful to add in the future. If this feature were to be implemented, however, congress member listings would need to be updated in order to display and select members of prior Congresses. Also, the ability to detect changes in the current congress and update member data and bill status locations in an automated manner would provided lasting functionality and updates to the application with minimal required actions performed by the developers.

References

[1] J. L. Kent and A. Newcomb, “The fcc just stopped regulating equal access to the internet,” Dec 2017.

[2] P. Bell, “Public trust in government: 1958-2017,” May 2017.