

# **Homework 9: Congress Search Android App – A Mobile Phone Exercise**

## **1. Objectives**

- Become familiar with Android Studio, Android App development for Android.
- Build a good-looking Android app using the Android SDK.
- Build mobile user experience for Congress Search using the Sunlight Congress APIs

## **2. Background**

### **2.1 Android Studio**

Android Studio is the official IDE for Android application development, based on IntelliJ IDEA (<https://www.jetbrains.com/idea/>). On top of the capabilities you expect from IntelliJ, Android Studio offers:

- Flexible Gradle-based build system
- Build variants and multiple apk file generation
- Code templates to help you build common app features
- Rich layout editor with support for drag and drop theme editing
- Lint tools to catch performance, usability, version compatibility, and other problems
- ProGuard and app-signing capabilities
- Built-in support for Google Cloud Platform, making it easy to integrate Google Cloud Messaging and App Engine

The home page of the Android Studio is located at:

<http://developer.android.com/tools/studio/index.html>

### **2.2. Android**

Android is a mobile operating system initially developed by Android Inc., a firm purchased by Google in 2005. Android is based upon a modified version of the Linux kernel. As of December 2013, Android was the number 1 mobile OS, in unit sales, surpassing iOS, while iOS was still the most profitable platform.

The Android operating system software stack consists of Java applications running on a Java based object oriented application framework on top of Java core libraries running on the Dalvik virtual machine featuring JIT compilation.

The Official Android home page is located at:

<http://www.android.com/>

The Official Android Developer home page is located at:

<http://developer.android.com/index.html>

### **2.3 Sunlight Congress API**

The Sunlight Congress API is a live JSON API for the people and work of Congress, provided by the Sunlight Foundation. With the API you can:

- Look up members of Congress by location or by zip code.
- Obtain the official Twitter, YouTube, and Facebook accounts.
- Look up the daily work of Congress: bills, amendments, nominations.
- Get the live activity of Congress: past and future votes, floor activity, hearings.

In order to make requests to Sunlight Congress API, you need to get an API key:

- Go to <http://sunlightfoundation.com/api/>
- Click on “Get a Key!
- Fill the sign up form
- You will get a confirm email. Click on the link in the email.

Then login at the Sunlight Foundation website and go to your “Profile Settings”. You will see your API key string right above you name. For information about the Sunlight Congress API, please go to:

<https://sunlightlabs.github.io/congress/>

and

<http://tryit.sunlightfoundation.com/congress>

### **2.4 Amazon Web Services (AWS)**

AWS is Amazon’s implementation of cloud computing. Included in AWS is Amazon Elastic Compute Cloud (EC2), which delivers scalable, pay-as-you-go compute capacity in the cloud, and AWS Elastic Beanstalk, an even easier way to quickly deploy and manage applications in the AWS cloud. You simply upload your application, and Elastic Beanstalk automatically handles the deployment details of capacity provisioning, load balancing, auto-scaling, and application health monitoring. Elastic Beanstalk is built using familiar software stacks such as the Apache HTTP Server, PHP, and Python, Passenger for Ruby, IIS 7.5 for .NET, and Apache Tomcat for Java.

The Amazon Web Services homepage is available at: <http://aws.amazon.com/>

### **2.5 Google App Engine (GAE)**

Google App Engine applications are easy to create, easy to maintain, and easy to scale as your traffic and data storage needs change. With App Engine, there are no servers to maintain. You simply upload your application and it’s ready to go. App Engine applications automatically scale based on incoming traffic. Load balancing, micro services, authorization, SQL and NoSQL

databases, memcache, traffic splitting, logging, search, versioning, roll out and roll backs, and security scanning are all supported natively and are highly customizable.

To learn more about GAE support for PHP visit the page:

<https://cloud.google.com/appengine/docs/php/>

### 3. Prerequisites

This homework requires the use of the following components:

1. This homework requires the use of the following components:
  - A. Download and install Android Studio. You may use any other IDE other than Android Studio such as Eclipse, but you will be on your own if problems spring up.
  - B. First you need to install Java on your local machine. You can download JDK 8 from - <http://www.oracle.com/technetwork/java/javase/downloads/index.html>. For windows users, after installing the JDK, you need to add environment variables for JDK.
    - Properties -> Advanced -> Environment Variables -> System variables -> New Variable

Name: JAVA\_HOME, Variable Value: <Full path to the JDK>

- Typically, this full path looks something like C:\Program Files\Java\jdk1.8.0.

Then modify the PATH variable as follows on Microsoft Windows:

C:\WINDOWS\system32;C:\WINDOWS;C:\Program Files\Java\jdk1.8.0\bin

This path may vary depending on your installation.

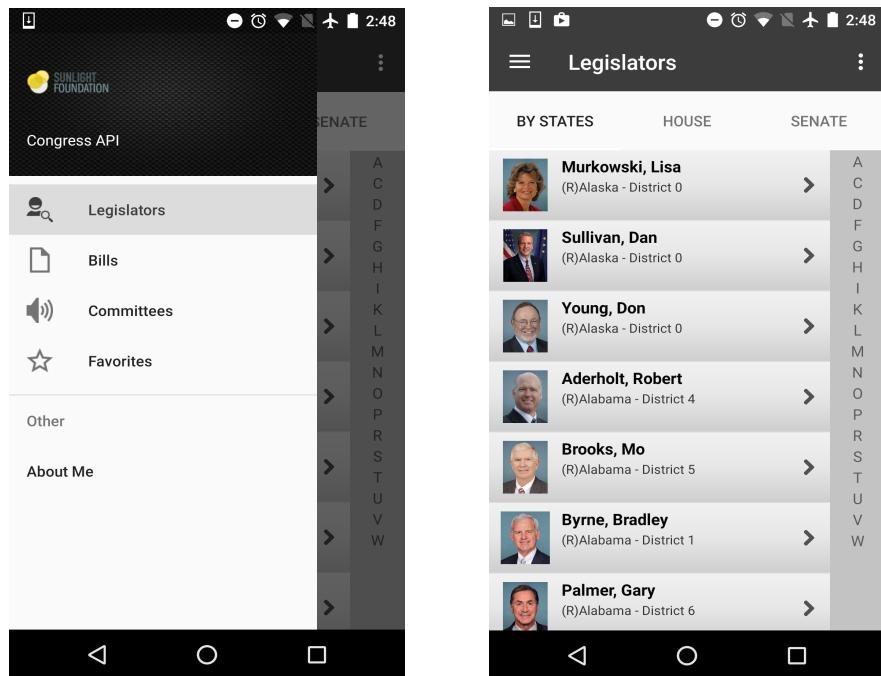
- Note: The PATH environment variable is a series of directories separated by semicolons (;) and is not case-sensitive. Microsoft Windows looks for programs in the PATH directories in order, from left to right. You should only have one bin directory for a JDK in the path at a time. Those following the first instance are ignored. If you are not sure where to add the path, add it to the right of the value of the PATH variable. The new path takes effect in each new command window you open after setting the PATH variable.
- Reboot your computer and type “java -version” in the terminal to see whether your JDK has been installed correctly.

Set up the Android Studio environment so that you can run any sample android app on your phone/tablet/virtual device from it. Then you can start with this homework app. You will need to enable “Developer Options” and “USB debugging” if you are using an actual device. There are endless resources a simple search away on how to setup your Android Studio.

## 4. High Level Design

In this exercise, you will develop an Android Mobile application, which will have following functionality:

There will be a slide out-menu which will provide access to the different screens such as Legislators, Bills, Committees and Favorites. The content displayed in each of these sections will be similar to Homework 8 but we will go into details later on. The initial screen will be defaulted to Legislators as shown below.



**Figure 1. Initial Screens**

On successful loading, we would get the legislator details using our PHP script hosted on Amazon Web Services/Google App Engine, which would return the result in JSON format. We would display the legislator details in a ListView component in the 'By States' tab. Again, the legislator screen is defaulted to the 'By States' tab. It has two other tabs 'House' and 'Senate' which displays the legislator details according to their Chamber.

## 5. Implementation

### 5.1 Legislator Section

You must replicate the Legislators Screen, as shown in Figure 1. The tab names remain the same as in Homework 8.

The interface consists of the following:

- A 'ListView' component displaying the list of legislators

- A ‘TabHost’ component for hosting screens and data for different tabs
- A ‘LinearLayout’ component to host the side-index on every tab for easy access based on state or last name

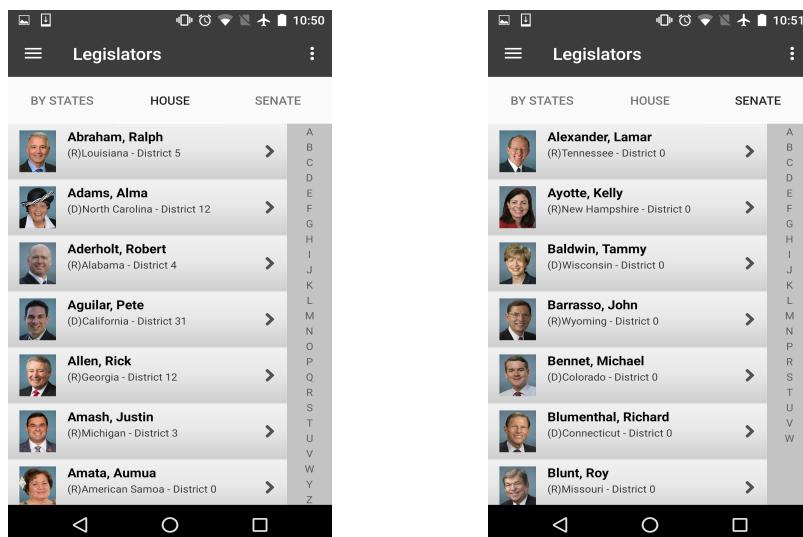
Every element in the list view should contain:

Field	Description
<b>Thumbnail</b>	Photo of the legislator (Can be retrieved based on the bio-guide id from <a href="https://theunitedstates.io/images/congress/original/bioguide_id.jpg">https://theunitedstates.io/images/congress/original/bioguide_id.jpg</a> )
<b>Party</b>	Republican or Democrat – Should be in format (R) for Republican and (D) for Democrat
<b>State</b>	The state to which the legislator belongs to – Should be full name of state
<b>District</b>	The district number within the state which he/she represents
<b>Arrow Button</b>	Arrow Image to guide user for viewing details

Each row in the list view should be clickable which will take you to the legislator details screen (explained later). Below is an explanation of the required details in every tab.

- **‘By States’:** Ordered list of legislators **based on state** which they represent. Order the states ascending from A to Z based on the first letter of state. **Index should be created based on state name.**
- **‘House’:** Order legislators **based on their last name** in ascending order. This tab should contain only **legislators from chamber ‘house’**. **Index should be created based on last name.**
- **‘Senate’:** Order legislators **based on their last name** in ascending order. This tab should contain only **legislators from chamber ‘senate’**. **Index should be created based on last name.**

Below are screenshots for the House and Senate tabs respectively.



**Note:** Each tab should be darkened or bolded on selection in the way that the active tab should be dark or bold faced and other tab names are greyed out.

### 5.1.1 Legislator Details:

On clicking a specific row in the Legislator Section, the Legislator details should be displayed in a new screen as shown below. The displayed screen will have the following clickable icons and a set of fields as indicated below.

The fields on this page are described below. The photo and party details are center aligned and each of the labels are left aligned whereas the details are right aligned. On clicking the **back arrow button** on this details page, **the app should go back to tab in the section where the row click was made. This can be any of the ‘By States’, ‘House’ or ‘Senate’ tabs in the Legislators section or even within the Legislators tab in the Favorites section.**

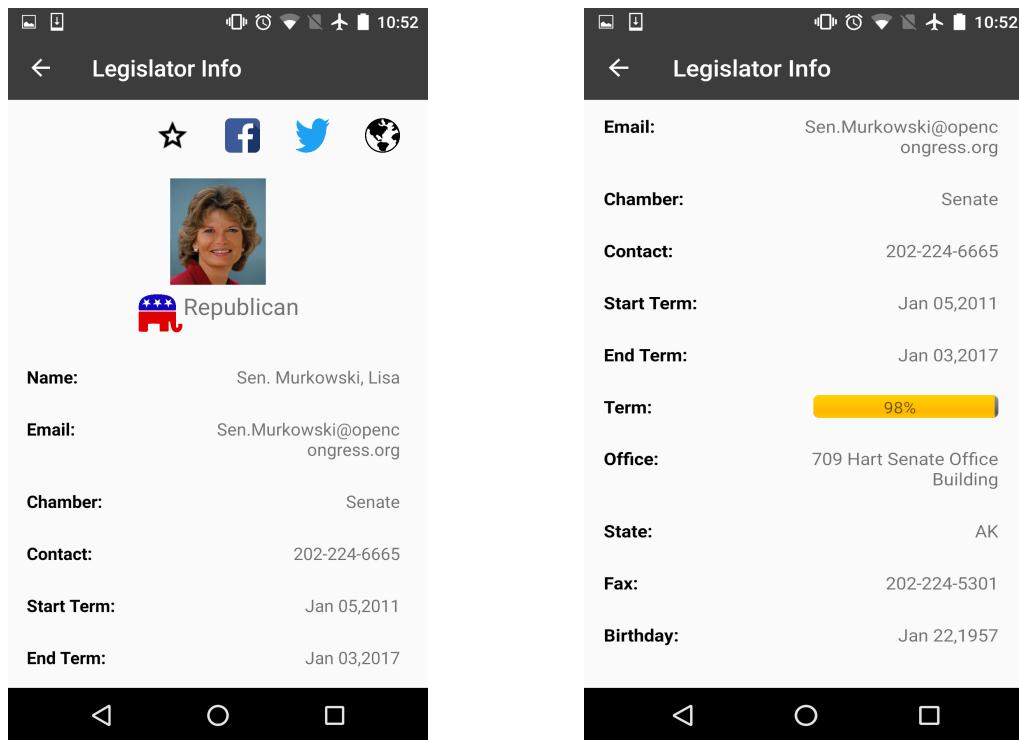


Figure 2: Legislator Details

#### Icons:

- **Plain star** – Clickable star icon used to add to favorites (explained later on)
- **Facebook icon** – On click of this icon there should be a browser opened displaying the legislator's Facebook page
- **Twitter icon** – On click of this icon there should be a browser opened displaying the legislator's Twitter page
- **Earth icon** – On click of this icon there should be a browser opened displaying the legislator's webpage.

**Note:** If the legislator does not have a webpage, Facebook or Twitter page an error message should be displayed as a toast message.  
(<https://developer.android.com/guide/topics/ui/notifiers/toasts.html>)

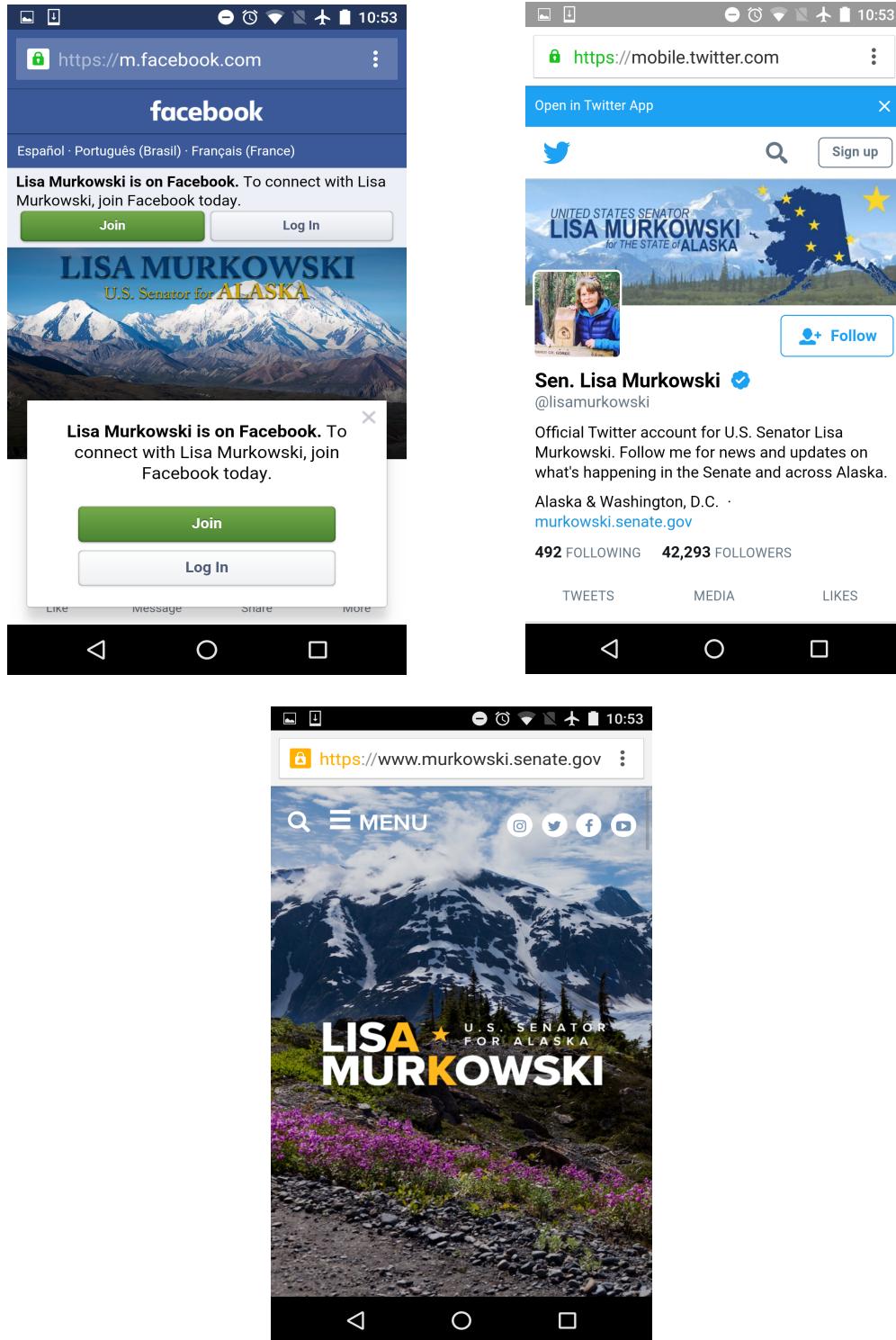


Table Field	Description
<b>Photo</b>	Displays the photo of the legislator. You should use the base link: <a href="https://theunitedstates.io/images/congress/original">https://theunitedstates.io/images/congress/original</a> and the "bioguide_id" to create the image URL.
<b>Party</b>	The party of the legislator. Format should be: <i>Party image, Party name</i> .
<b>Name</b>	Displays the name of the legislator. The format should be <i>Title, Last Name, First Name</i> .
<b>Email</b>	The email address of the legislator. The email should be a link, click on the link to write a new email to this address.
<b>Chamber</b>	Displays the chamber of the legislator. Format should be <i>Chamber: chamber name</i> .
<b>Contact</b>	The phone number of the legislator
<b>Start Term</b>	The start term of the legislator. Format of date should be: <i>Month DD, YYYY</i>
<b>End Term</b>	The end term of the legislator. Format of date should be: <i>Month DD, YYYY</i>
<b>Term</b>	A progress bar with percentage showing the term progress. The progress percentage should be $(\text{now} - \text{start}) / (\text{end} - \text{start}) * 100$
<b>Office</b>	The office of the legislator
<b>State</b>	The state the legislator belongs to
<b>Fax</b>	The fax number
<b>Birthday</b>	The birthday of the legislator. Format of date should be: <i>Month DD, YYYY</i>

**Note:** If any of the values are not present, your application should display “N.A” or “None”. It should not crash or show blank.

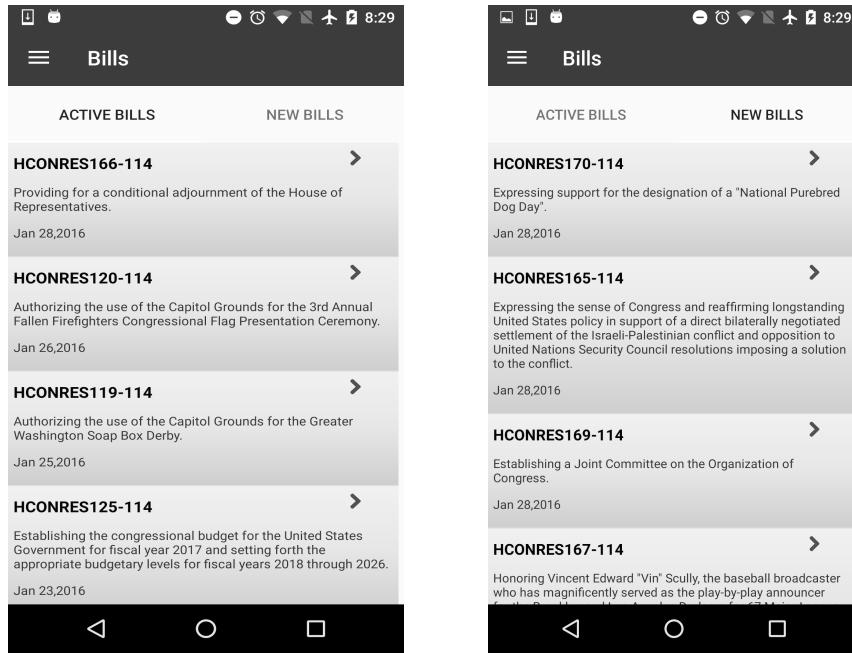
## 5.2 Bills Section

You must replicate the Bills Screen, as shown in Figure 3 below. The tab names remain the same as in Homework 8 and all the bills should be loaded.

The two tabs in this page are:

- ‘**Active Bills**’: Displays the active bills in descending order with newer bills at the **top**. Sort by the introduced date.
- ‘**New Bills**’: Displays new bills which may or may not be active. These are also sorted by introduced date in descending order.

There is no index for this screen and just the ListView which is clickable.



**Figure 3: Bills Section**

Every entry in the bills list should contain the following:

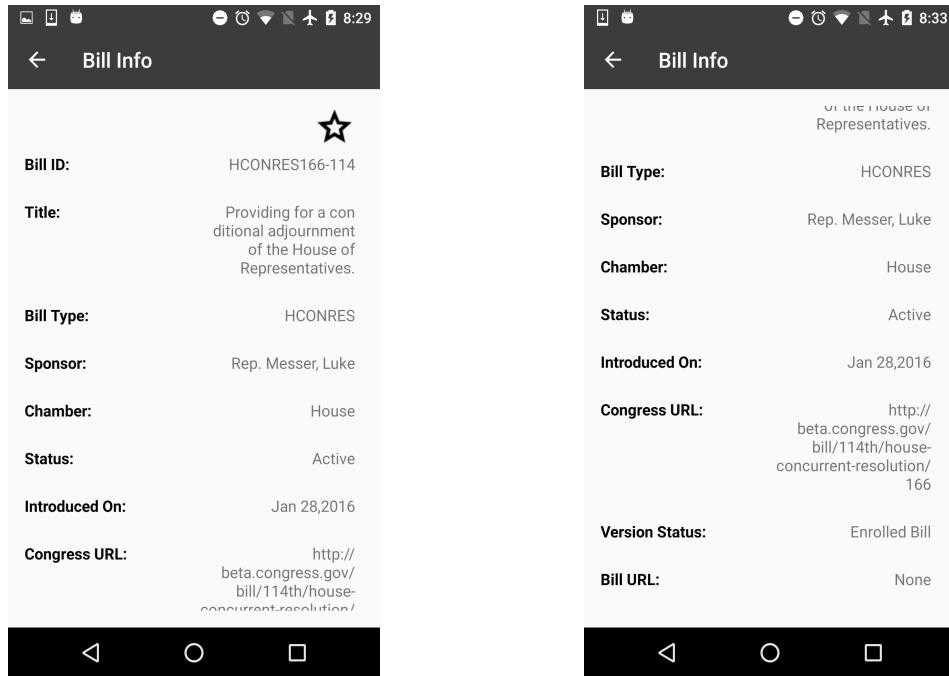
Field	Description
<b>Bill ID</b>	Displays the bill id of the bill
<b>Short title</b>	Displays the short title of the bill, if short title is null then display the long title
<b>Introduced On</b>	Displays the date on which the bill was introduced in Month DD,YYYY format
<b>Arrow Button</b>	Arrow Image to guide user for viewing details

Note: You have to highlight the active tab text in black and grey out the other tab

### 5.3 Bill Details

On clicking a specific row in the Bills Section, the Bill details should be displayed in a new screen as shown below. The displayed screen will have the **one clickable icon for adding to favorite** and a set of fields as indicated below.

On clicking the **back arrow button** on this details page, **the app should go back to tab in the section** where the row click was made. This can be any of the 'Active Bills or 'New Bills' tabs in the Bills section or even within the Bills tab in the Favorites section.



**Figure 4: Bill Details**

The following details would be displayed:

Table Field	Description
<b>Bill ID</b>	ID of the Bill
<b>Bill Type</b>	Type of the Bill
<b>Sponsor</b>	Sponsor of the Bill. Format should be: Title, Last Name, First Name
<b>Chamber</b>	Chamber of the Bill. Start with the image of the Chamber. Then the name of the chamber.
<b>Introduced</b>	The date that the bill was introduced. Date format should be: <i>Month DD,YYYY</i>
<b>Sponsor</b>	Sponsor of the Bill. Format should be: Title, Last Name, First Name
<b>Status</b>	The status should be “Active” or “New”, depends on the “active” field in JSON response.
<b>Introduced On</b>	The date that the bill was introduced. Date format should be: <i>Month DD,YYYY</i>
<b>Congress URL</b>	The link of congress URL
<b>Version Status</b>	The “version name” field in “last version” field of JSON response
<b>Bill URL</b>	The link to the pdf version of the bill

**Table Field**

**Description**

<b>Bill ID</b>	ID of the Bill
<b>Bill Type</b>	Type of the Bill
<b>Sponsor</b>	Sponsor of the Bill. Format should be: Title, Last Name, First Name
<b>Chamber</b>	Chamber of the Bill. Start with the image of the Chamber. Then the name of the chamber.
<b>Introduced</b>	The date that the bill was introduced. Date format should be: <i>Month DD,YYYY</i>
<b>Sponsor</b>	Sponsor of the Bill. Format should be: Title, Last Name, First Name
<b>Status</b>	The status should be “Active” or “New”, depends on the “active” field in JSON response.
<b>Introduced On</b>	The date that the bill was introduced. Date format should be: <i>Month DD,YYYY</i>
<b>Congress URL</b>	The link of congress URL
<b>Version Status</b>	The “version name” field in “last version” field of JSON response
<b>Bill URL</b>	The link to the pdf version of the bill

**Note:** If any of the values are not present, your application should display “N.A” or “None”. It should not crash or show blank.

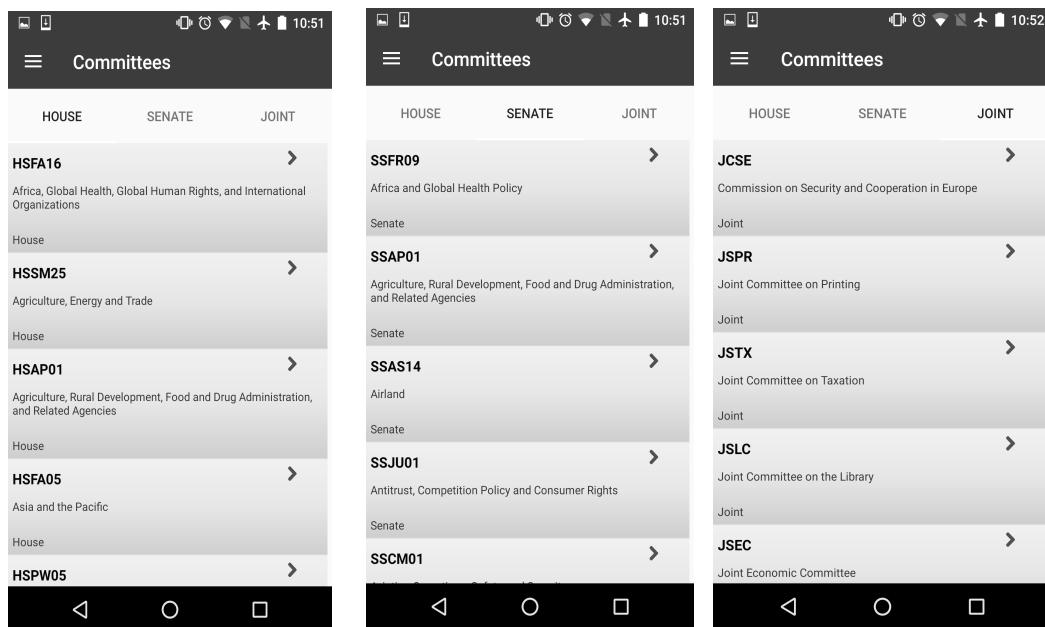
## 5.4 Committees Section

You must replicate the Committees Screen, as shown in Figure 5 below. The tab names remain the same as in Homework 8 and all the committees should be loaded.

The three tabs in this page are:

- ‘**House**’: Displays the house committees in ascending order of committee names.
- ‘**Senate**’: Displays the senate committees in ascending order of committee names.
- ‘**Joint**’: Displays the joint committees in ascending order of committee names.

There is no index for this screen and just the ListView which is clickable.



**Figure 5: Committees Section**

Every entry in the committee list should contain the following:

Field	Description
<b>Committee ID</b>	Displays the committee id of the bill
<b>Committee Name</b>	Displays the name of the committee
<b>Chamber</b>	Displays the chamber name of the committee
<b>Arrow Button</b>	Arrow Image to guide user for viewing details

Note: You have to highlight the active tab text in black and grey out the other tab

## 5.5 Committee Details

On clicking a specific row in the Committees Section, the Committee details should be displayed in a new screen as shown below. The displayed screen will have the **one clickable icon** for adding to favorite and a set of fields as indicated below.

On clicking the **back arrow button** on this details page, **the app should go back to tab in the section** where the row click was made. This can be any of the 'Joint', 'House' or 'Senate' tabs in the Committees section or even within the Committees tab in the Favorites section.

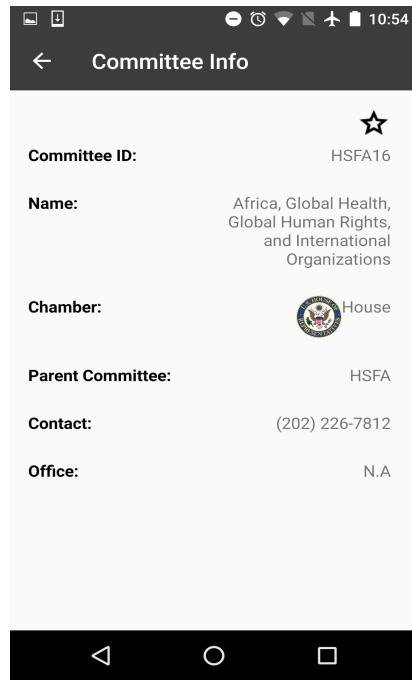


Figure 6: Committee Details

The details have to be in a tabular format. Each of the entry in the list should display the following details:

Table Field	Description
Committee ID	ID of the Committee
Name	Name of the Committee
Chamber	Chamber of the Committee. Start with the image of the Chamber. Then the name of the chamber.
Parent Committee	Parent Committee of the Committee. If not available, leave empty
Contact	Phone of the Committee
Office	Office of the Committee. If not available, display N.A

**Note:** If any of the values are not present, your application should display "N.A" or "None". It should not crash or show blank.

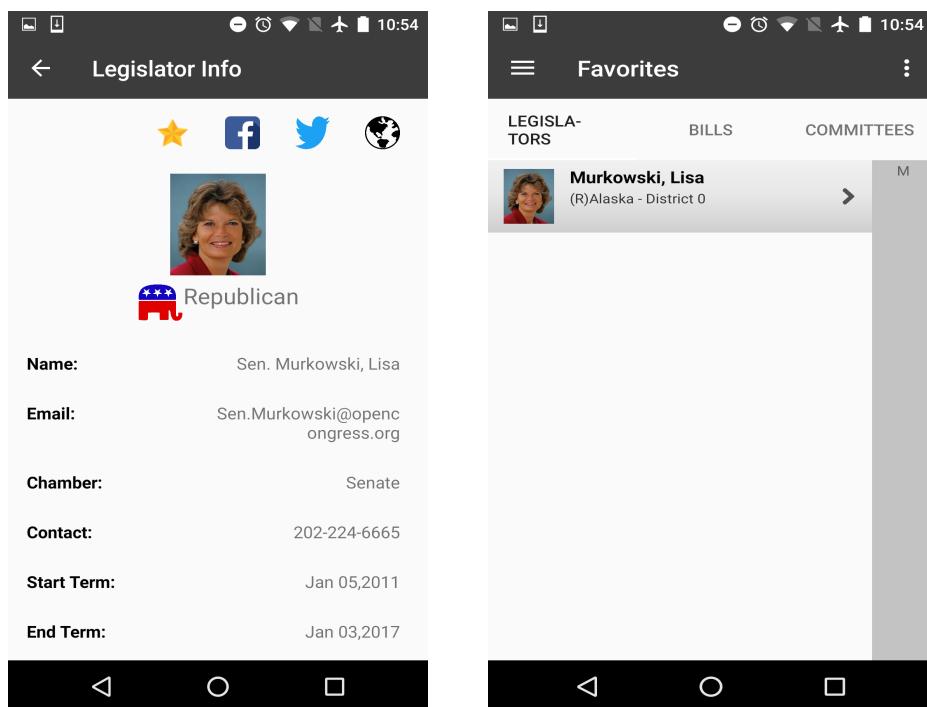
## 5.6 Favorites Section

You must replicate the Favorites Screen, as shown in Figure 5 below. The tab names remain the same as in Homework 8 and **should remain empty initially**.

When the user marks a particular item as favorite, it should appear in the corresponding tab for Legislators, Bills or Committees. Further, when the user unmarks the same item from favorites, the item should disappear. This is the traditional favorite/un-favorite functionality.

When an item is marked as favorite, the star icon which we have described in the different sections should change to yellow/golden color to indicate that it has been marked. Similarly, when a favorite item is unmarked, the golden color should disappear and icon comes back to regular.

You will see this in the demo video.



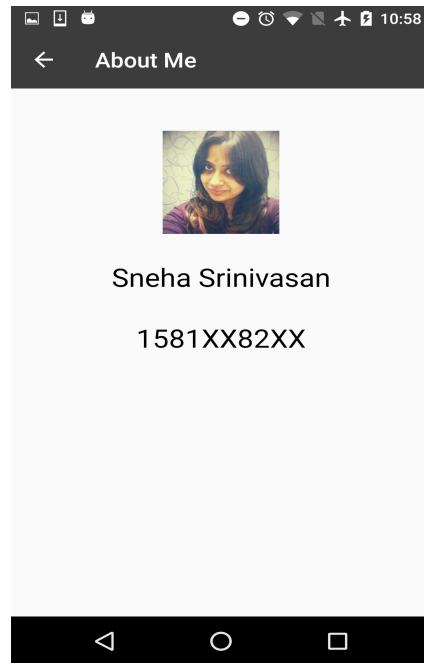
**Figure 8. Favorites**

**Note:** The rest of the details for Legislators, bills and committees are same as the previous sections in terms of what should be displayed in the ListView for each section as well as on click of the row item.

## 5.7 About me

You will also have to implement an additional menu item in the Navigation bar that links to a display screen which displays information about you.

On clicking the **back arrow button** on this details page, **the app should go back to the active tab in the section where the row click was made**. For example: If you click About me from the Bills Section and you are currently on New Bills, it should go back to New Bills on the Bills section. It should look as below:



**Figure 9. About Me**

## 6. Implementation Hints

See the HW9 Android Clues file.

## 7. Material You Need to Submit

Unlike other exercises, you will have to “demo” your submission “in person” during a special grading session. Details and logistics for the demo will be provided in class, in the Announcement page and in Piazza.

You should also ZIP your Android source directory and SUBMIT the resulting ZIP file. Make sure that the source path does not include the .apk binary file.