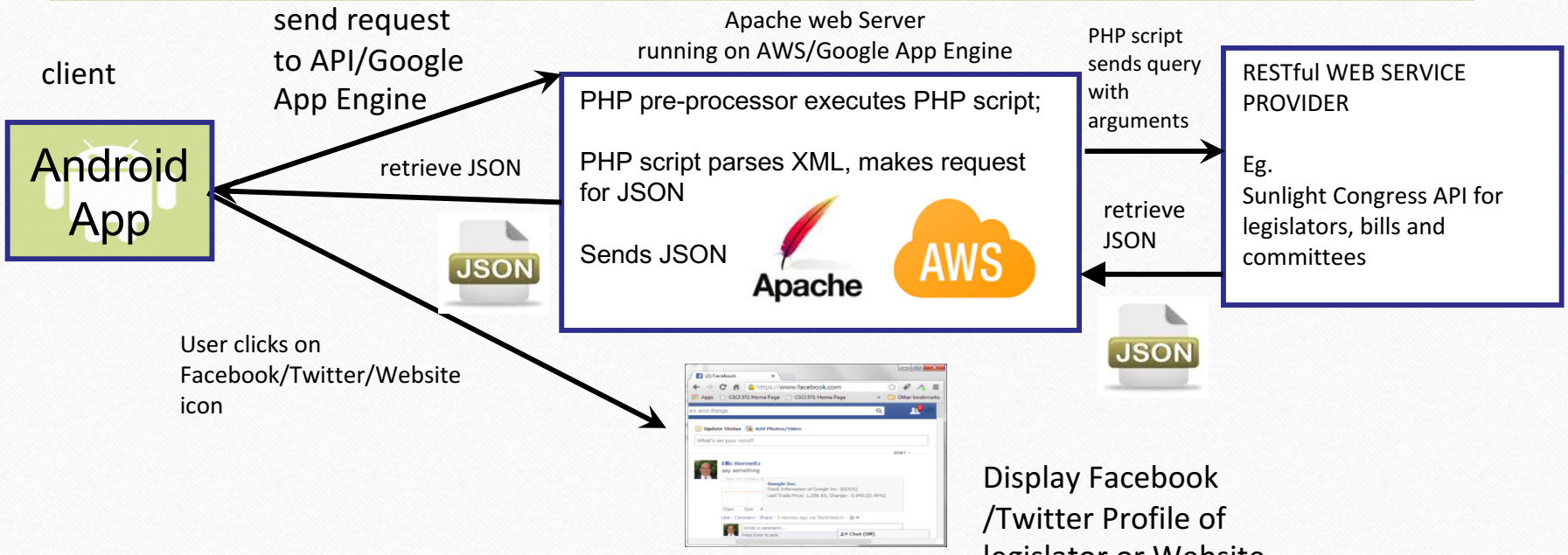
What is needed

- You will need to download and install Android Studio
  https://developer.android.com/sdk/index.html
- If you have a device, you can build and run the app on your android device(if not, just run it on the emulator)
- Note if you wish to run it on the emulator, you might have to go through installation which is an added component to the SDK

HW#9 Architecture Overview



HW9 Implementation

- You will mainly create 2 activities, 4 Fragment Activites and a Manifest file
- There will be other files too and you can create more as necessary
- **AndroidManifest.xml**
- **MainActivity.java** – routine that controls the entire process
  - Creates the initial screen
  - Initializes required variables for data sharing
  - Sets controls for Navigation Drawer
  - Initializes Navigation Drawer and its layout

HW9 Implementation

- **AboutMeActivity.java**
  - Displays student details
- **Fragment Activities**
  - The 4 fragment activities should be used to handle and display each of the 4 sections Legislators, Bills, Committees and Favorites as well as the various interactions.

AndroidManifest.xml File

Every application must have an AndroidManifest.xml file in its root directory. The manifest presents essential information about the application to the Android system. Among other things, the manifest does the following:

- It names the Java package for the application.
- It describes the components of the application — the activities, services, broadcast receivers, and content providers that the application is composed of.
- It names the classes that implement each of the components and publishes their capabilities.

AndroidManifest.xml File

See http://developer.android.com/guide/topics/manifest/manifest-intro.html.

Please note that the file is created by default on creation of a new Android project using Android Studio IDE.

UI Controls in Android

For the homework exercise, you can use the following widgets (not limited to):

- **TextView** (i.e., label)
  http://developer.android.com/reference/android/widget/TextView.html
- **EditText** (i.e., text field)
  http://developer.android.com/reference/android/widget/EditText.html
- **AutoCompleteTextView**
  http://developer.android.com/reference/android/widget/AutoCompleteTextView.html

UI Controls in Android (Contd.)

For the homework exercise, you can use the following widgets (not limited to):

- **Button** http://developer.android.com/reference/android/widget/Button.html
- **RadioButton** http://developer.android.com/reference/android/widget/RadioButton.html
- **ImageView** http://developer.android.com/reference/android/widget/ImageView.html
- **Switch** http://developer.android.com/reference/android/widget/Switch.html
- **ProgressBar** http://developer.android.com/reference/android/widget/ProgressBar.html

UI Controls in Android (Contd.)

- ListView http://developer.android.com/reference/android/widget/ListView.html
- TabLayout http://developer.android.com/reference/android/support/design/widget/TabLayout.html
- ViewPager http://developer.android.com/reference/android/support/v4/view/ViewPager.html
- WebView http://developer.android.com/reference/android/webkit/WebView.html

UI Controls in Android (Contd.)

- **ImageSwitcher** (It is useful to animate an Image on screen)
  http://developer.android.com/reference/android/widget/ImageSwitcher.html
- **TextSwitcher** (It is useful to animate a label on screen)

  http://developer.android.com/reference/android/widget/TextSwitcher.html
- **TableLayout**
  http://developer.android.com/reference/android/widget/TableLayout.html

UI Controls in Android (Contd.)

- **TableRow** http://developer.android.com/reference/android/widget/TableRow.html
- **RelativeLayout**
  http://developer.android.com/reference/android/widget/RelativeLayout.html
- **LinearLayout** (It arranges "components" in vertical or horizontal order, via orientation attribute.)
  http://developer.android.com/reference/android/widget/LinearLayout.html
- **ScrollView** http://developer.android.com/reference/android/widget/ScrollView.html

RelativeLayout

RelativeLayout lets you position your component base on the nearby (relative or sibling) component's position. You can use "above, below, left and right" to arrange the component position.

LinearLayout

LinearLayout is a common layout that arranges "component" in vertical or horizontal order, via *orientation* attribute

MainActivity.java

onCreate does the following

•Initialize data sharing variables - https://developer.android.com/reference/android/content/SharedPreferences.html

•Render and setup the initial screen with Legislator Section and its three tabs

•Register and setup the Navigation Drawer and load its menu components with the required text and icons

•Setup toggle for opening and closing the Navigation Drawer

Fragment Activities

- Each Fragment activity will extend the Fragment class
- Used to render and setup the initial screen for each section with the required tabs
- Use TabHost for the tabs in each section(https://developer.android.com/reference/android/widget/TabHost.html)
- Each child fragment within the parent Fragment will implement gathering data and display of the ListView along with OnItemClickListener for the ListView
- Use AsyncTask to make calls and gather data from the API or Web service(will be explained next)

Fragment Activities(contd…)

- **onItemClickListener does the following:**
- Create new Intent for displaying details of that row(Legislator details, Bill Details or Committee Details)
- Add the Legislator object as an extra in the intent, bundle it with the intent
- Start the intent to display details
- Use AsyncTask to make calls and gather data from the API or Web service(will be explained next)

AsyncTask within sub class of Fragments

AsyncTask is an abstract class provided by Android which helps us to use the UI thread properly. This class allows us to perform long/background operations and show its result on the UI thread without having to manipulate threads.

AsyncTask has four steps:

doInBackground: Code performing long running operation goes in this method. When onClick method is executed on click of button, it calls execute method which accepts parameters and automatically calls doInBackground method with the parameters passed.

AsyncTask (Contd.)

onPostExecute: This method is called after doInBackground method completes processing. Result from doInBackground is passed to this method.

onPreExecute: This method is called before doInBackground method is called.

onProgressUpdate: This method is invoked by calling publishProgress anytime from doInBackground call this method.

DisplayActivity

- You will use different display activities for the displaying details on clicking the ListView row
- A total of three display activities to customize details screen for the three sections
- This activity will be responsible for getting the object for that row and implementing various listeners or displaying data for different fields in that screen. This will be done in the **onCreate** function of that activity.

AboutActivity.java

- ImageView can be used for student image.
- TextView can be utilized for student name and student id.

Other notes

- Take a look at https://developer.android.com/guide/topics/ui/layout/listview.html
- The above link also has links to Adapters which are classes that can be customized to bundle details or data with a list view row
- The icons required may be taken from the ones used in HW8 and for star filled and empty star you can get the nearest match

Android Libraries

You may find the below libraries useful to implement the features:

- Google Gson : https://github.com/google/gson
- Fast Image Loading and Caching: http://square.github.io/picasso/
- Apache Commons Lang : https://commons.apache.org/proper/commons-lang/

Android Libraries

You may find the below libraries useful to implement the features:

- Nhaarman library for ListView animations : https://github.com/nhaarman/ListViewAnimations
- Chrisbanes library for zoomable ImageView : https://github.com/chrisbanes/PhotoView