Homework 9
Android App for Weather Forecast

1. Objectives

- ➢ Become familiar with Android Studio, Android App development and Facebook SDK for Android.
- ➢ Build a good-looking Android app using the Android SDK.
- ➢ Add social networking features using the Facebook SDK.
- ➢ Add map features using the AERIS weather SDK.

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. Android dominates the world mobile operating system market share by a large margin.

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 (for versions <= Android 4.4 KitKat) and the ART runtime from Android 5.0. 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
You can start learning android at https://developer.android.com/training/index.html.

2.3 Facebook

Facebook is a social networking service launched in February 2004, owned and operated by Facebook Inc. Users can add friends and send them messages, and update their personal profiles to notify friends about themselves and what they are doing. Users can additionally post news feeds to their profiles, and these feeds may include images, besides text messages.

The Facebook homepage is available at: http://www.facebook.com. Facebook provides developers with an application-programming interface, called the Facebook Platform.

2.4 AERIS Weather

AERIS weather overlays allow you to overlay weather information onto any interactive mapping framework, such as Google Maps, Apple Maps and Mapbox. In order to load the proper tiles into your mapping application and since our tiles are time-specific, there are a number of steps you will need to follow.

Read more at
http://www.aerisweather.com/support/docs/toolkits/aeris-android-sdk/getting-started/weather-maps/

3. Prerequisites

This homework requires the use of the following components:

1. 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. 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.

2. You also need to create a Facebook Developer application as you did for your homework 8. Follow the following steps to get started:

- **Download SDK**: Download the latest Facebook Android SDK
  Link: https://developers.facebook.com/docs/android
- Instructions to **import in Android Studio**:
  https://developers.facebook.com/docs/android/getting-started
- Create a **new app on facebook** developer:
  https://developers.facebook.com/apps/
- Specify **App Info** related to the HW9 android application you are developing.
- **Key Hashes**: Specify Android key hash for the development environment using the commands mentioned.
- Track App Installs and App Opens: Not required.
- Next Steps: Utilize **Login** (optional) and **Share** tutorials to achieve the functionality required for the exercise.

Note: In your Facebook application settings, you should go to the "Status & Review" section and choose "Yes" for the question "*Do you want to make this app and all its live features available to the general public*?" as you did for homework 8.

3. You also need to install AERIS Map library in your app.

- Create an **Aeris account**: http://www.aerisweather.com/signup
- **Register the application** to get keys: http://www.aerisweather.com/account/apps
- Generate SHA1 in your system and use it to create Google Maps API key. Use GoogleMaps API key in your AndroidManifest.xml.
  - **SHA1 Creation** details:
    http://stackoverflow.com/questions/27609442/how-to-get-the-sha1-fingerprint-certificate-in-android-studio-for-debug-mode
  - **Google Maps API key creation**:
    https://developers.google.com/maps/documentation/android-api/
- Import Aeris Map library as a module in your android app.
  - **Aeris Map SDK** Link: https://github.com/aerisweather/AerisAndroidSDK It contains SDK as well as examples. **Follow its readme for installation** and usage.

Also see: http://www.aerisweather.com/support/docs/toolkits/aeris-android-sdk/getting-started/weather-maps/

4. Description of the Exercise

In this exercise, you will implement an Android Mobile application (hereby the "app") that ports the Homework 8 web interface to a mobile interface. The app does the following sequence of actions:

4.1. Initial Form

Create an Android Activity that takes the "Street"(**mandatory**), "City"(**mandatory**), "State"(**mandatory**) and "Degree"(**mandatory**) as input. The "Street" and "City" fields are text fields (*called EditText in Android*), the "State" field is a dropdown menu selector (*called Spinner in Android*) and "Degree" field is a radio button group (*called RadioGroup with RadioButton in Android*). For the labels you can use *TextView in Android*. The form should also contain Search and Clear Buttons. On the bottom-right corner you must display "Powered by:" with Forecast icon as in HW8. On click of Forecast icon, the forecast site "**http://forecast.io**" must open in the web browser. Sample screenshot for form activity is shown in Figure 1.

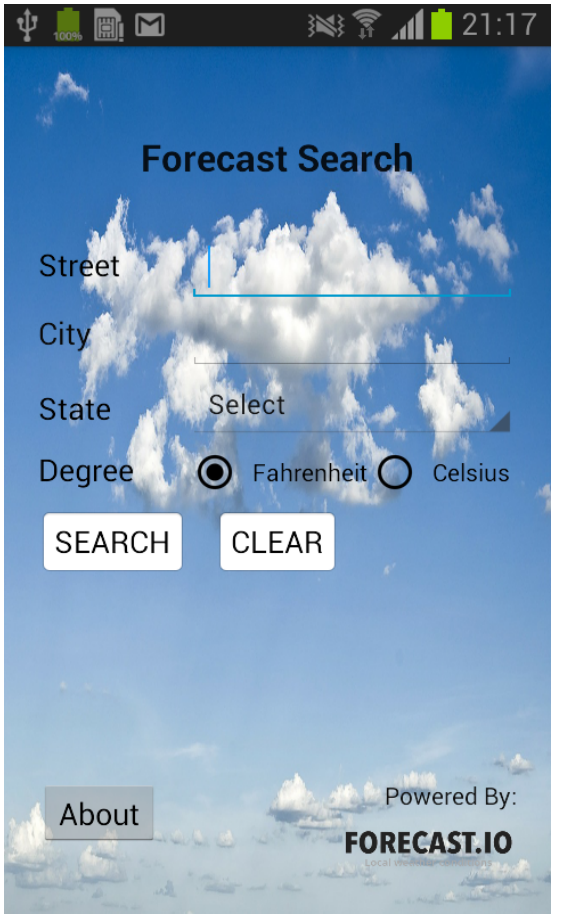

Figure 1: The Form

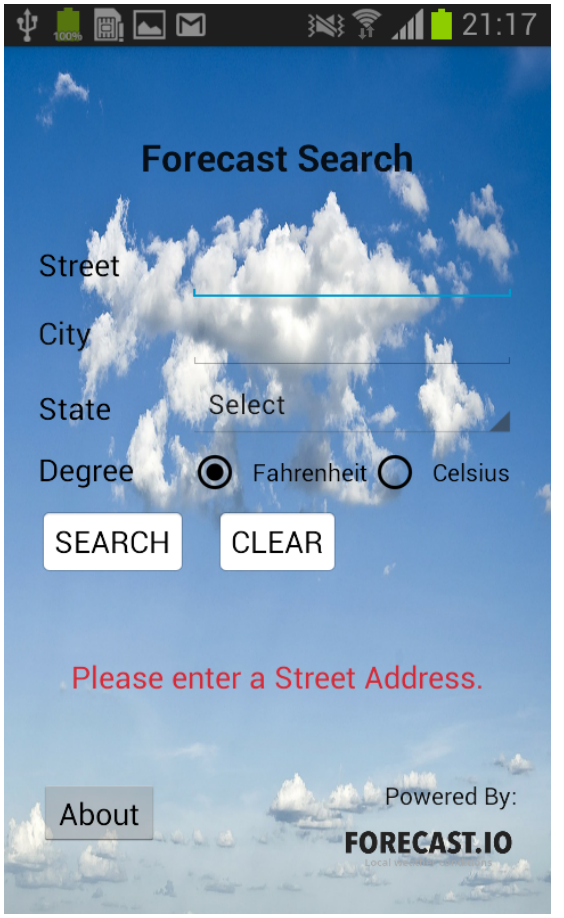

Figure 2: Error handling in the form

When the "Search" button is tapped, the search form is validated first. If the values provided in the form are valid, the app makes a request to the PHP script located on your AWS account (similar to hw8) and retrieve the JSON result. Another activity should be displayed to show the results, which is a scrollable list of items described below.

On clicking on the "Clear" button, the fields should be reset such that the text fields are empty, the State spinner must have the default value and degree must be set to Fahrenheit. Also any previous error messages must be cleared.

You may feel free to use any background image for your form activity. The default is the one from HW8.

4.1.1 Error handling

In the first activity, all errors must be shown at the bottom as shown in figure 2.

a. **Street Empty Field:** Please enter a Street.
b. **City Empty Field:** Please enter a City.
c. **State Empty Field:** Please select a State.

4.1.2 About Activity

On clicking "About" button, a new Activity should start, this activity should display the Student image, Name and Student ID, as shown in Figure 3.

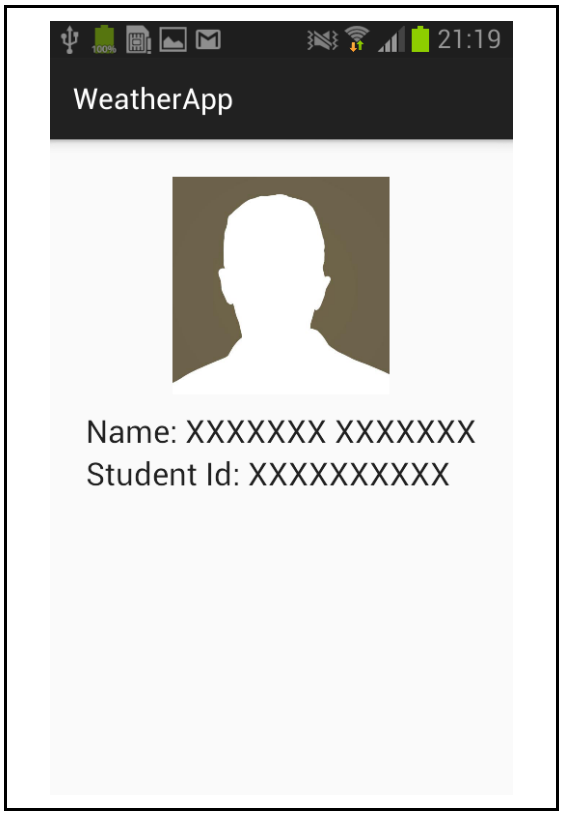

Figure 3: About Activity

4.2. Result Display

This activity should have a **scrollable** view detailing the current weather details for the given input. For the given input, its image, summary, temperature, low temperature and high temperature must be shown (similar to HW8). Further below that current weather details must be shown as in Figure 4 below. If any of the field is unavailable, you should show "N/A" instead. It also contains three buttons, for MORE DETAILS, VIEW MAP and Facebook Share.

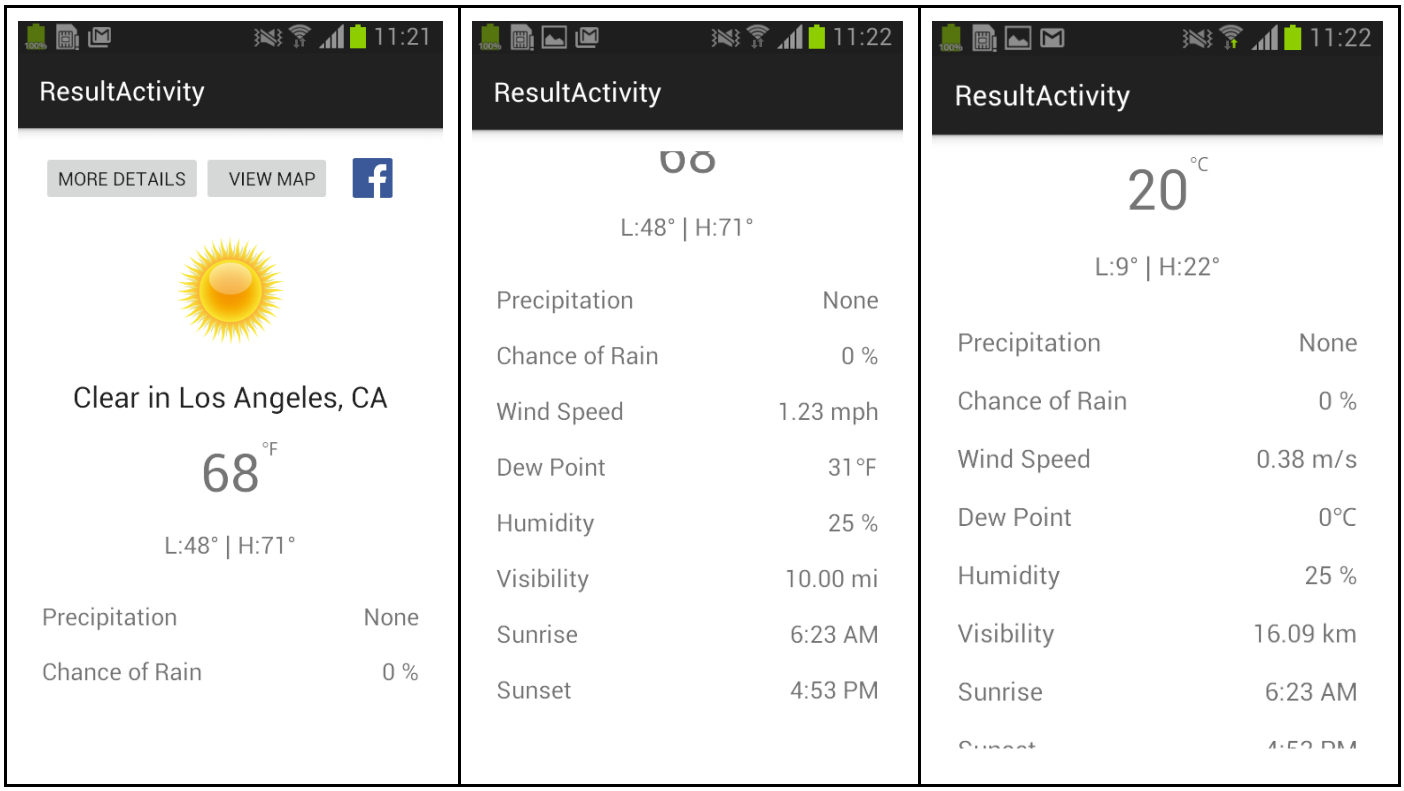

Figure 4: Results.

Tapping on the **MORE DETAILS button** must invoke the "Details Activity" described in section 4.3. Tapping on the **VIEW MAP button** should direct to the "Map Activity" described in section 4.4. Tapping on the **Facebook Share button** should invoke the facebook share process.

4.2.1 Facebook

Clicking on the facebook button on the Result Activity must allow users to login and post weather details on their own Facebook pages. The pop up window that first appears should also have an option to cancel the action, along with posting on Facebook. The information posted to Facebook must contain the title as "Current Weather in [City], [State]" where City and State should be taken from the initial form, weather summary, temperature, image and link to the Forecast.io home page (similar to HW8). The facebook post is shown in Figure 5.

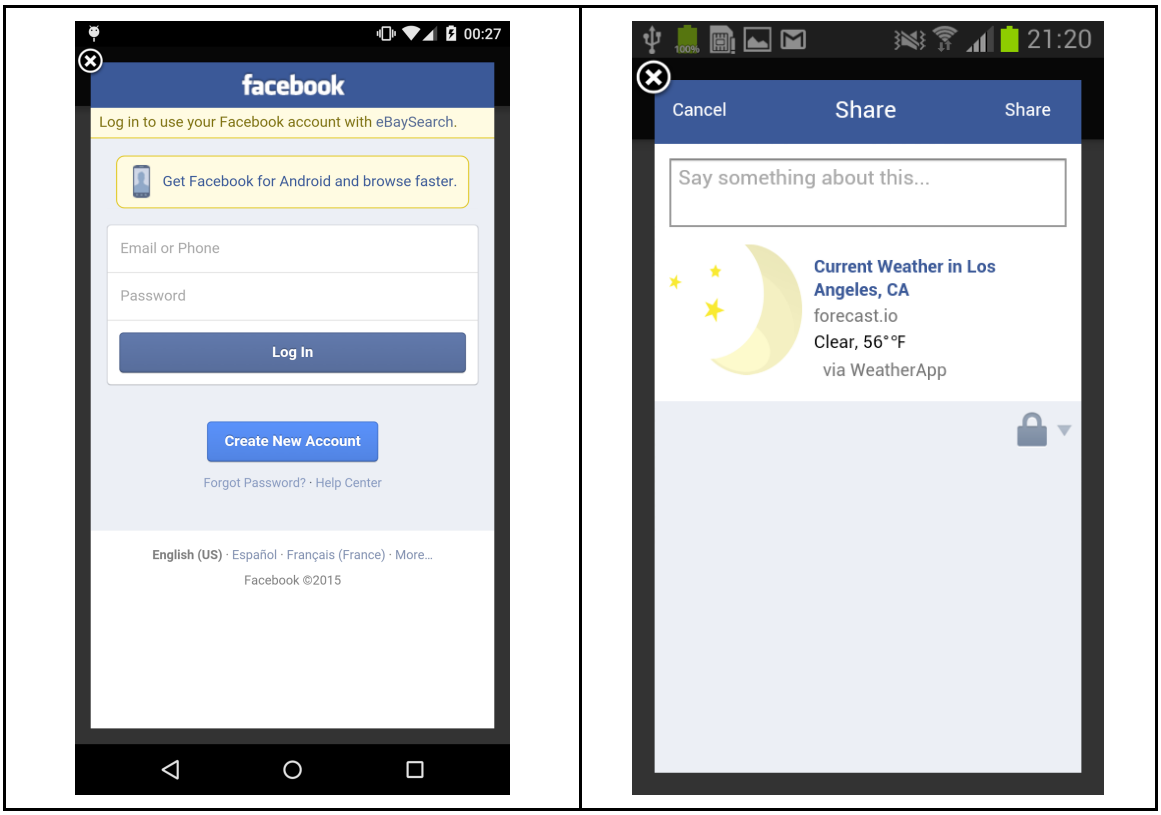

Figure 5: Facebook Post

On successful posting (or failure), appropriate short "**Toast**" messages should be shown at the bottom 'Facebook Post Successful' for successful post or 'Post Cancelled' for cancelled post as shown in Figure 6.

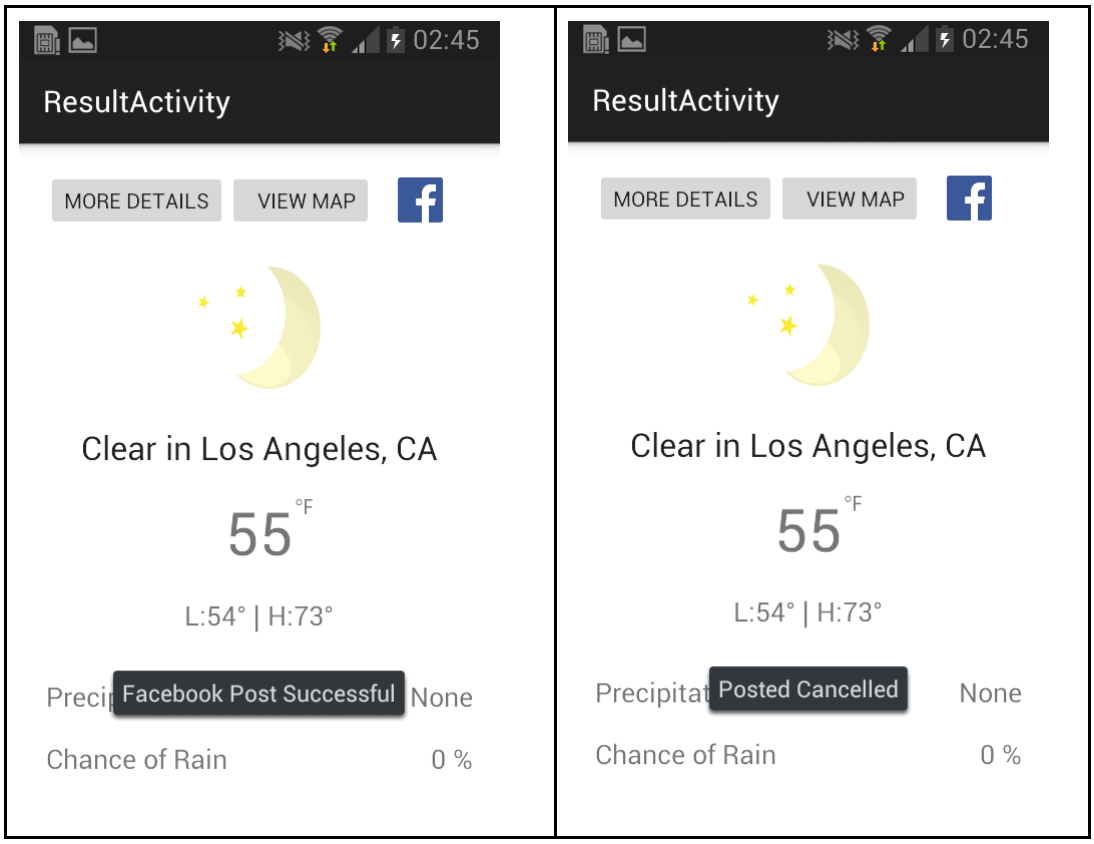

Figure 6: Facebook Toast message

4.3. Details Activity

When the MORE DETAILS button is tapped the DetailsActivity should start. This has two **tabs**, first NEXT 24 HOURS and second NEXT 7 DAYS.

4.3.1 Next 24 hours

When the DetailsActivity starts, this tab should be active and should show the hour, summary-icon and temperature in a tabular format for the next 24 hour **(from index 0 of json data)** as shown in Figure 7. Further at the bottom a **+ button** must be added. On click of this button, next 24 hours results should also be populated below. You can simply hide the + button and utilize that row itself to be the first row for the extra 24 hours (so that the entire data is rendered without any row break from previous data).

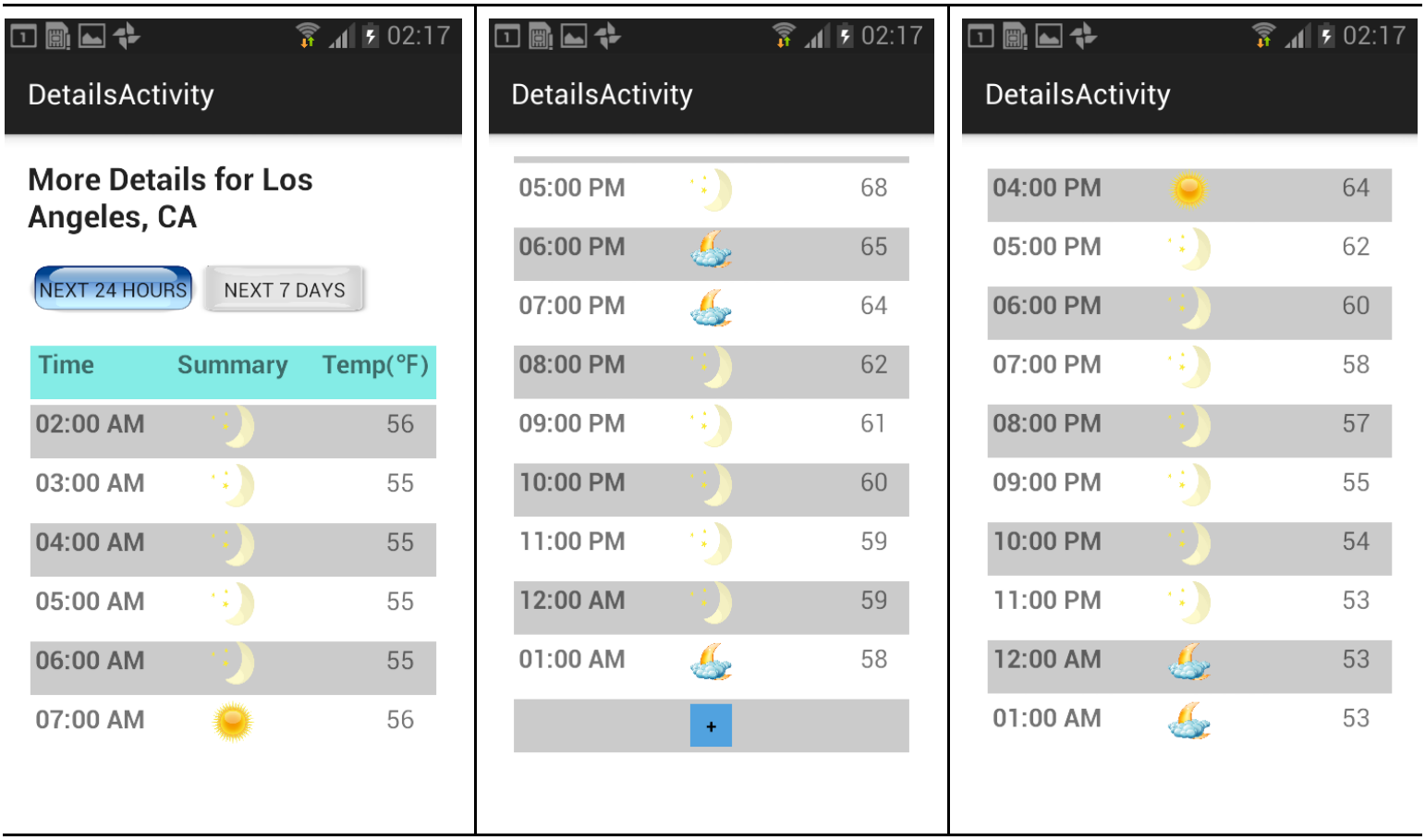

Figure 7: Details Activity: NEXT 24 HOURS tab.

You must show the details of the each hour similar to hw8. There is no bootstrap to help you with this, so you should use Android views to create the tabs, handle tab functionality, highlight the current tab button, display the current tab details and fade the non-current tabs. The entire Activity, with the tabs(inclusive) must be scrollable so that one can see the full details. For any unavailable fields, use "N/A".

4.3.1 Next 7 days

When the NEXT 7 DAYS button is tapped the second tab should open and display the next 7 days weather forecast **(from index 1 of JSON data)** as shown in Figure 8. The day and date should be displayed on the top-left, with the appropriate icon on the right. The Minimum and Maximum temperature should be displayed below. It should be scrollable so that all the seven days details can be viewed.

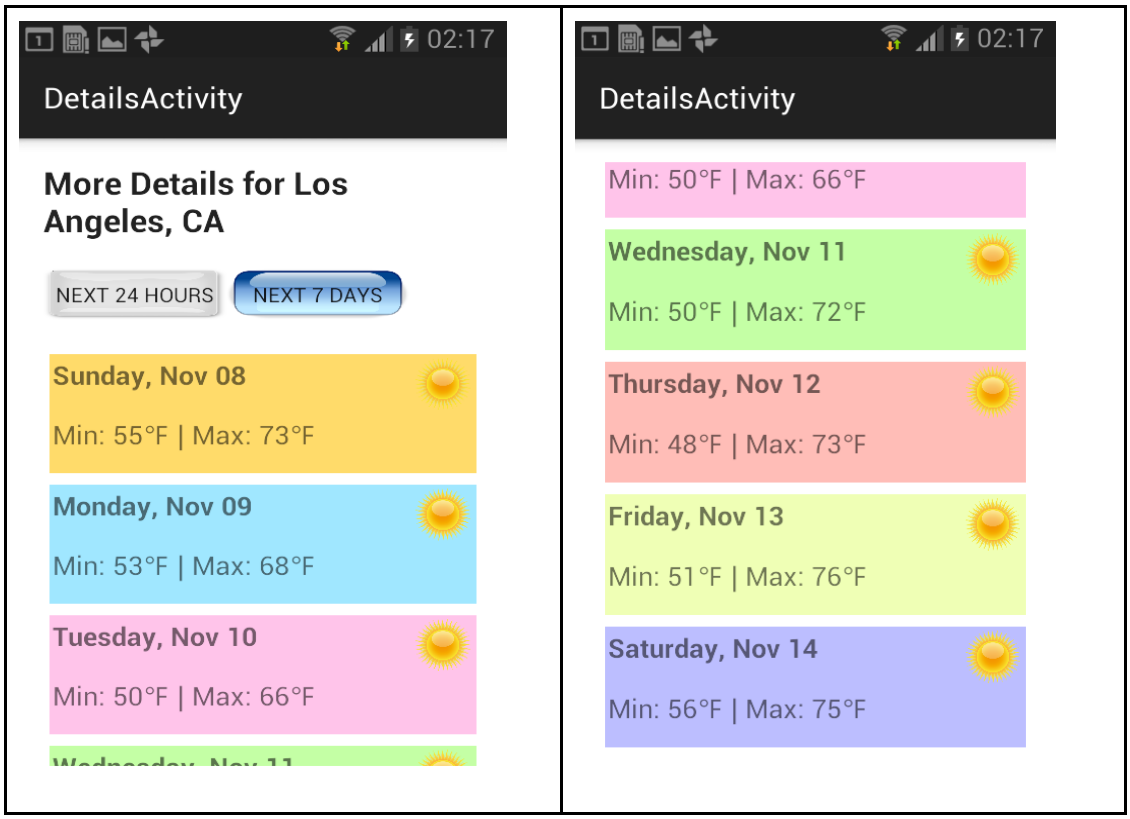

Figure 8: Details Activity: NEXT 7 DAYS tab.

4.4. Map Activity

When the VIEW MAP button is tapped on the Results Activity, Map Activity should start (Figure 9). It utilizes AERIS Weather Map for current input data. The layer used is AerisTile.RADSAT. Some implementation related hints related to Maps are in the next section. In the figure 9, the city browsed was Sterling, CO.

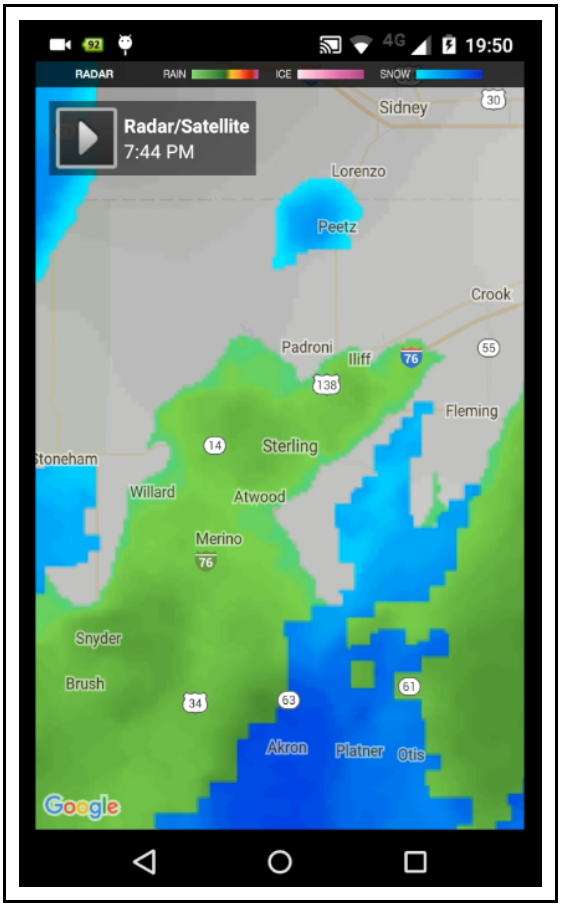

Figure 9: Map Activity

5. Implementation Hints

- **Facebook** offers multiple APIs for you to upload or request data. In this HW implementation, you need to use 2 Facebook APIs (Login and Share Dialog). You may also use any other facebook API if you desire so.
- You need to use **AsyncTask** class to fetch the JSON data/item images in background in order to avoid any "YourApp is not responding" error.
- Using **Relative Layouts** as containers for views will be very helpful, since you can hide/unhide an entire layout view which will hide/unhide all its children views. It is also good to position elements relative to their parent views.
- **Android "Toast"** messages can be very helpful in debugging an app.

Aeris map related hints:

- Create a Fragment extending **MapViewFragment** tied with a layout containing **AerisMapView** as an element.

- In the Fragment initialize **AerisEngine** before creating AerisMapView. Use AerisMap client secret key and package name of the android app to intitialize the AerisEngine.
- Set Location information and Layers in AerisMapView.
- Create an Activity with **FrameLayout** as a holder for fragment and add the fragment in it using **FragmentManager**.

6. Materials to submit

Unlike other exercises, for this homework 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, the class website, and Piazza.

You must 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.