

ITIS/ITCS 4180/5180 Mobile Application Development

Homework 8

Basic Instructions

1. In every file submitted you MUST place the following comments:
 - Assignment #.
 - File Name.
 - Full name of all students in your group.
2. Each group should submit only one assignment. Only the group leader is supposed to submit the assignment on behalf of all the other group members.
3. Your assignment will be graded for functional requirements and efficiency of your submitted solution. You will lose points if your code is not efficient or does unnecessary processing or blocks the UI threads.
4. Please download the support files provided with this assignment and use them when implementing your project
5. Export your Android project and create zip file which includes all the project folder and any required libraries.
6. The file name should follow the following format: **Group#_HW08.zip**
7. **Failure to follow the above instructions will result in point deductions.**

Homework 8 (100 points)

In this assignment you will get familiar with Firebase operations, RecyclerView and PreferenceActivity. You will build a weather app with a different look and features. You will learn how to use Preference Activity to add settings or preferences. You will also learn how to build RecyclerView and display items in a list using it.

Initial Setup and API Description

You should use Accuweather API (<http://developer.accuweather.com>) for loading the weather information. The API of interest is the 5 days of daily forecast API (<http://developer.accuweather.com/accuweather-forecast-api/apis/get/forecasts/v1/daily/5day/%7BlocationKey%7D>) which is based on the Unique Key for a City which can be acquired by Location API. You need to create an account in order to create an API Key. Follow the steps given below:

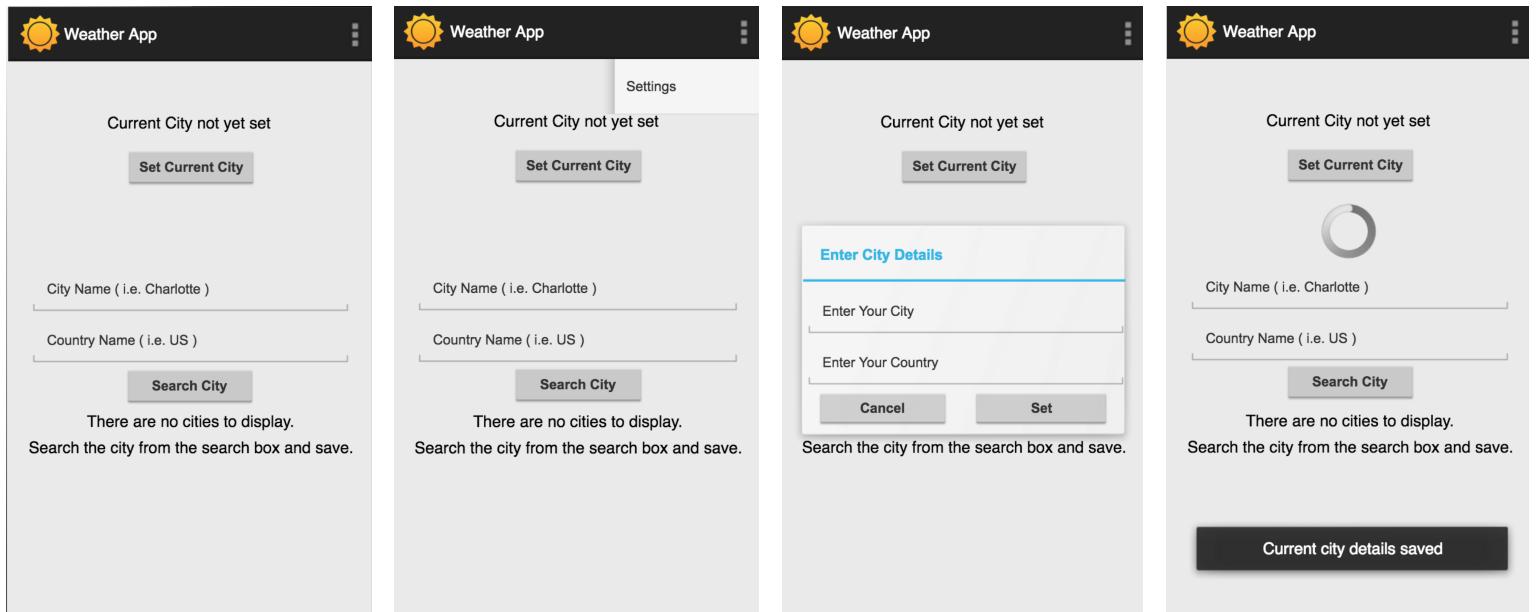
1. Go to <http://developer.accuweather.com> and Sign Up.
2. Fill up required details to create your account
3. You will receive an email confirming your account's been created.
4. Sign in with your credentials.
5. Go to My Apps and add an app to Generate a Key for your future use, or use the default key they provide.

Required API Calls:

1. **Location API** : http://dataservice.accuweather.com/locations/v1/{COUNTRY_CODE}/search?apikey={YOUR_API_KEY}&q={CITY_NAME}
2. **Current Forecast for Current city** : http://dataservice.accuweather.com/currentconditions/v1/{CITY_UNIQUE_KEY}?apikey={YOUR_API_KEY}
3. **5Day Forecast for search cities** : http://dataservice.accuweather.com/forecasts/v1/daily/5day/{CITY_UNIQUE_KEY}?apikey={YOUR_API_KEY}
4. **Weather ICON** : http://developer.accuweather.com/sites/default/files/{Image_ID}-s.png

Example API Calls:

1. http://dataservice.accuweather.com/locations/v1/US/search?apikey={YOUR_API_KEY}&q=Charlotte
2. http://dataservice.accuweather.com/currentconditions/v1/349818?apikey={YOUR_API_KEY}
3. http://dataservice.accuweather.com/forecasts/v1/daily/5day/349818?apikey={YOUR_API_KEY}
4. <http://developer.accuweather.com/sites/default/files/01-s.png>



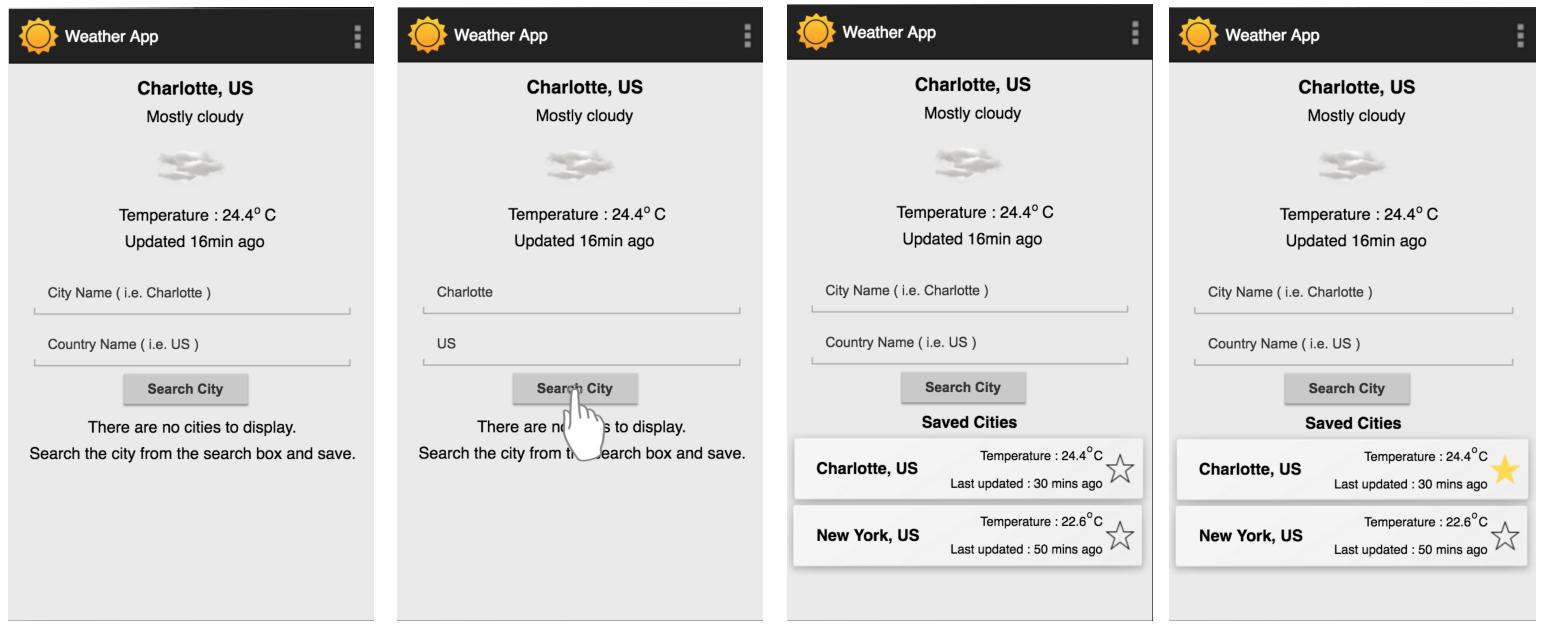
(a) Main Activity

(b) Main activity with menu

(c) AlertDialog to set current city

(d) Loading current city details

Figure 1: Main Activity Wireframes



(a) Main Activity with Current city details

(b) Main activity with city search

(c) Main Activity with to saved current city

(d) Main Activity with saved cities and favorites

Figure 2: Main Activity Wireframes

Part 1: Main Activity (40 points)

The activity UI should match the UI presented in Figures 1 and 2. The requirements are given below.

1. The activity should initially display Set Current City button (If user didn't set current city details previously) and two EditTexts to enable users to give inputs of City name and Country name (Figure 1) along with a Search button. If user set current city

- details previously, get the current weather details by using Current forecast API and load them in main activity as in figure 2(a).
2. User can set current location by clicking on Set Current City button. AlertDialog Should be displayed up on clicking the Set Current city button as in Figure 1(c).
 3. After Providing City details and clicking on set button, by using location API, get the first city object in JSON response and get the unique key for the City with key named "Key", and display Toast with message "Current City details saved". If no city details found then display the toast with message "City not found".
 4. Store the Key, City name and Country in shared preferences and get the current weather details of the city by using Current forecast API.
 5. Get "LocalObservationDateTime", "WeatherText", "WeatherIcon", "Metric" in "Temperature" to display the current city forecast in Main Activity as in Figure 2(a).
 6. To display the weather icon use the URL given above and for all icons with ids<10 append 0 as prefix and get the image.
 7. To search cities User should provide the City name and Country as inputs and press Search button. Pressing Search button should start the CityWeather Activity.
 8. Create the Firebase database with table Cities with columns: citykey, cityname, country, temperature, favorite. This table should be used to store the saved cities.
 9. This activity should also include a list of previously saved cities in Firebase database. Each item must display the City name, Country name, Temperature when saved, update date and a star button. The star button will act as the defining sign for the favorites. Initially the start should be grey, when the user taps the grey star it should mark the stored city as a favorite and should change the star to gold. Use "star_gray.png" and "star_gold.png" from the Resources.
 10. If there are no stored cities in the database then show the TextView displaying, "There are no cities to display. Search the city from the search box and save." as shown in Figure 1.
 11. The List of saved cities should be displayed using RecyclerView (<https://developer.android.com/reference/android/support/v7/widget/RecyclerView.html>). You should use LinearLayoutManager to display the list as a vertical scrolling list. You can find how to use RecyclerView here: <https://guides.codepath.com/android/using-the-recyclerview>.
 12. There should be one Menu button in the Main Activity. The menu item is "Settings." Clicking on the Settings menu item should start a PreferenceActivity (Figure 3).
 13. In the preference activity you should design Preference for temperature units and Current City. User can change the temperature from Celsius to Fahrenheit or vice versa. User can also change their current city from preferences by using alert dialog provided.
 14. Figure 2(b) shows the changed temperature from Celsius to Fahrenheit. You must maintain the preferences in shared preferences and use them in the rest of the activities.
 1. The details of PreferenceActivity will be found at : <https://developer.android.com/guide/topics/ui/settings.html>

15. A Toast message should confirm that the “Temperature Unit has been changed to °C to °F” or vice versa.
16. If User has current city already, then load the details in AlertDialog and change the button title as Update as in Figure 3(d). Otherwise just provide the AlertDialog to set. Repeat the steps 3 and 4.
17. If a user press a gray star, then you should mark the City as favorite and change the favorite field in database as TRUE. You should change the color of the star button to Golden. Finally, you should put the favorite cities upwards at the top of the RecyclerView (Figure 3).
18. Long press on any of the items in the RecyclerView should delete it from both the list and database.

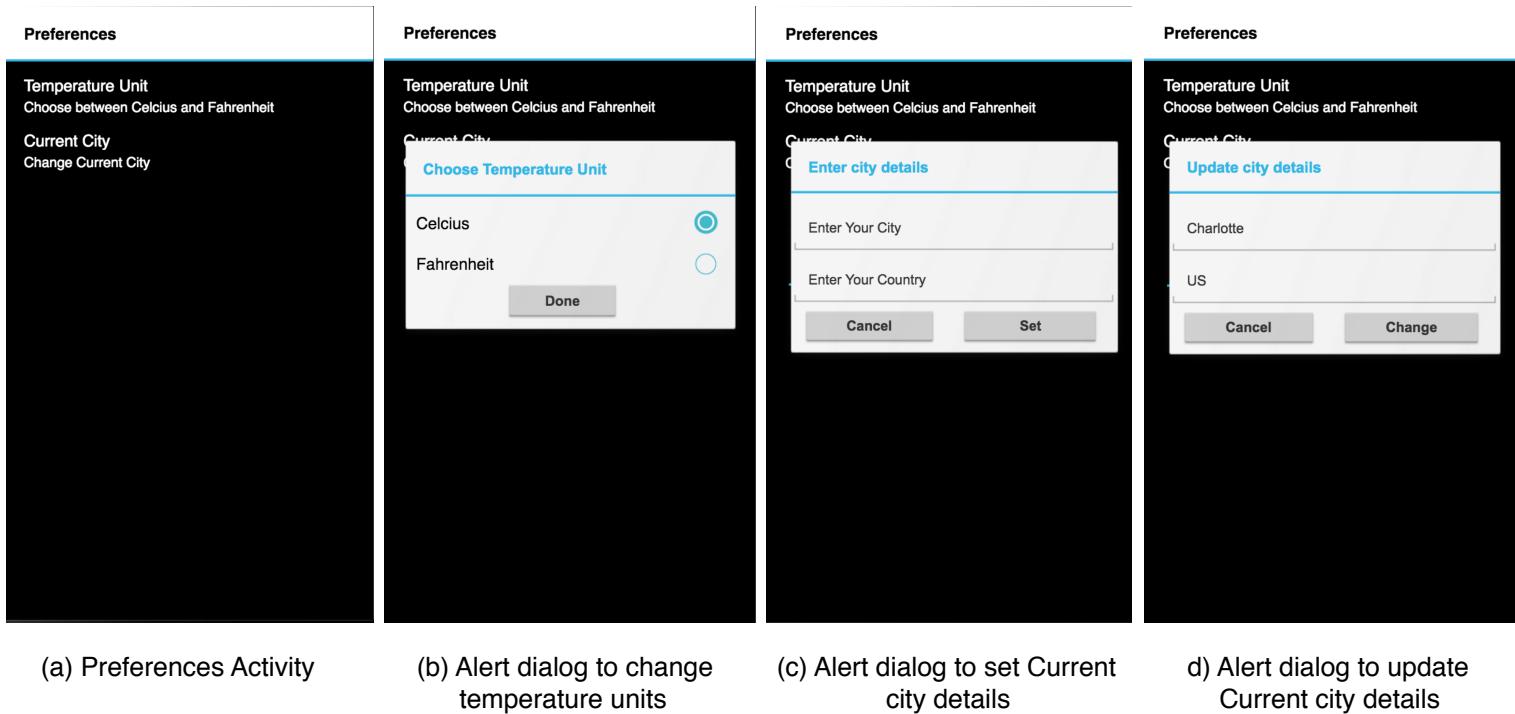


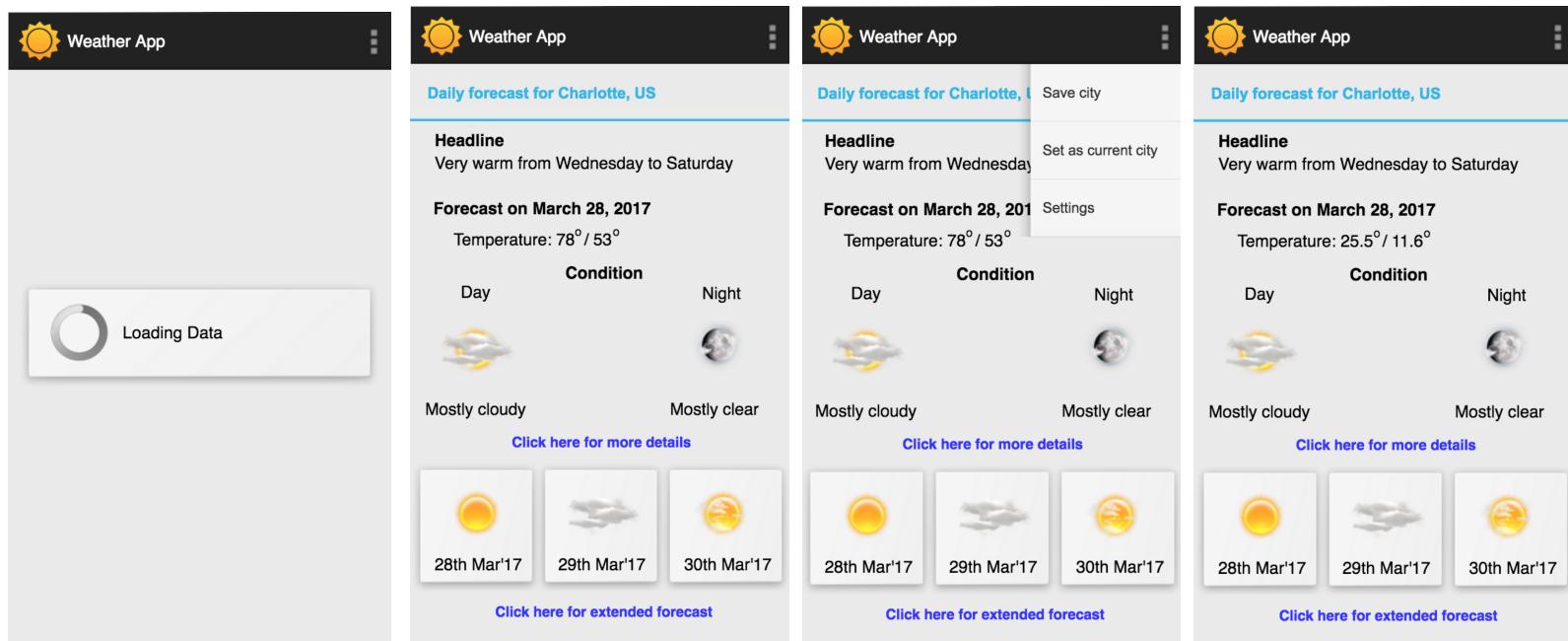
Figure 3, Preferences Activity Wireframes

Part 2: CityWeather Activity (60 points)

Clicking on Search button from the Main Activity should start CityWeather Activity. This activity should display the detailed Weather Forecast for the searched City. The requirements are as follows:

1. When Search button is pressed you should start a Progress dialogue showing parsing progress as shown in Figure 4 (a).
2. Location API call should be triggered using the provided user input. The API call should return the data for searched city. By using the key got from the Location, Trigger the 5 day forecast API. API should return 5 day forecast data with one headline.
3. If no city found, then display the Toast with message “City not found” and should return to main activity.

4. Parse the data using JSON Parser. You will find one headline for current city and you will find 5 day forecast items each having min, max temperatures and forecast for day, night times.
5. At the top of the layout, display the text indicating the headline and the city & country as “Daily Forecast for {City},{Country}” (Figure 4 (b)).
6. Then you should display the forecast for current day and a Horizontal RecyclerView of items, each including the icon which indicates the Weather in daytime on that particular day. The RecyclerView should show 3 items on screen at once, and should be scrollable horizontally to display the remaining items, see Figure 4(b)).
 - a. Each item should be clickable. Clicking on any item should show the detailed weather for that day on top of that RecyclerView, see Figure .
 - b. Each item in the RecyclerView should display the Date, corresponding weather icon for that day in daytime.
 - c. You need to load the weather icon image using Weather ICON link mentioned above. In JSON, you will get the Symbol serial number in the “DailyForecasts” array, an attribute called “Icon” in “Day” contains it.
7. Clicking on RecyclerView item, should display the weather details as mentioned in Figure
8. Clicking on “Click for more details” text should take you to web browser by using Implicit intents. Please use the url in each DailyForecast object in DailyForecasts Array with the key “MobileLink”.
9. Extended forecast will provide you the forecast for next 90 days. Clicking on “Click for extended forecast” text should take you to web browser by using Implicit intents. Please use the url in each HeadLine object with the key “MobileLink”.



(a) City Weather Activity while loading

(b) City Weather Activity

(c) City Weather Activity with menu

(d) City Weather Activity with changed temperature units

Figure 4, City Weather Activity Wireframes

10. There are three Menu buttons in this activity.
1. Save City: Pressing on this button should save the city's details (citykey, cityname, country, temperature, favorite) into Firebase realtime database (Figure 4 (c)).
 1. If the city has been previously saved, then simply update the stored temperature to reflect the new temperature. A Toast should display the message, "City Updated".
 2. If the city has not been previously saved, then save the new city and set the favorite flag to false. In case of temperature, always use celsius unit to save into database. A Toast should display the message, "City Saved".
 3. Upon returning to the Main Activity the list should show the newly added city as a saved city.
 2. Set as Current City: Pressing this button should set the city as current city.
 1. If current city has been saved, then update details in shared preferences and a Toast should display the message, "Current City Updated".
 2. If current city has not been saved, then set details in shared preferences and a Toast should display the message, "Current City Saved".
 3. Settings: Pressing on this button will do exactly same tasks with Preference Activity as Main Activity to change the temperature from Celsius to Fahrenheit or vice versa. User can also change their current city from preferences by using alert dialog provided. See wireframes for preferences activity in Figure 3.

Note:

1. Use SimpleDateFormat to parse the date and Pretty Time to display last updated time.
2. By default Make "Celsius" as the temperature unit and upon changing the Temperature unit from Settings, In all activities temperature should change based on the temperature units from preferences.
3. UI should strictly match the wireframes.