

ITIS/ITCS 5180 Mobile Application Development

Homework 03

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#_HW03.zip**
7. **Failure to follow the above instructions will result in point deductions.**

Homework 03 (100 points)

We will be building 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/cities/{Country}/search?apikey={YOUR_API_KEY}&q={CITY_NAME}
2. **Current Conditions API**: http://dataservice.accuweather.com/currentconditions/v1/{CITY_UNIQUE_KEY}?apikey={YOUR_API_KEY}

CITY_UNIQUE_KEY can be retrieved from JSON returned by the Location API

3. **Forecast API**, 5 Day 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/cities/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>

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 the user didn't set current city details previously) and two EditTexts to enable users to give inputs for 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 1(d).

2. User can set current location by clicking on Set Current City button. An AlertDialog Should be displayed up on clicking the Set Current city button as in Figure 1(b).
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



Figure 1: App wireframes

message "Current City details saved". If no city details found then display the toast with message "City not found".

4. Get "LocalObservationDateTime", "WeatherText", "WeatherIcon", "Metric" in "Temperature" to display the current city forecast in Main Activity as in Figure 1(d).

5. 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.
6. **Changing the current city:** clicking on the city name should let the user change the current city.
 1. You can repeat Step 2 by preloading the current city, and country.
7. To search cities User should provide the City name and Country as inputs and press Search button. Pressing Search button should start an alert dialog to display the list of cities found by the keywords. Clicking on any one of the options should start the City Weather Activity, figure 1(d, e, and f).
8. Create a class City with the variables citykey, cityname, country, temperature, favorite.
9. This activity should also keep a global list of City objects for saving cities and display a list of previously saved cities.
10. Each item in the cities list should 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. The stars can be found in the Android resources.
11. If there are no stored cities in the list 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.
12. The List of saved cities should be displayed using RecyclerView.
13. Long press on any of the items in the RecyclerView should delete it from the global list.

Part 2: City Weather Activity (60 points)

This activity should display the detailed Weather Forecast for the searched City. The requirements are as follows:

1. When Search button is pressed in the main activity you should use the Location API to retrieve the cities that match the search keyword and display that using an alert dialog. Selecting a city from the dialog should start City Weather activity for that selected city.
2. If no city found, then display the Toast with message “City not found” and should return to main activity.
3. 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.
4. At the top of the layout, display the texts indicating the city, country and headline. See Figure 2.
5. Then you should display the forecast for current day including the one icon which indicates the weather at daytime, and another icon indicating the weather at night on that particular day.
6. Below that there should a horizontal RecyclerView containing 5 days weather forecasts summary, see Figure 2.
 1. Each item should be clickable. Clicking on any item should show the detailed weather for that day, updating the current screen’s views.
 2. Each item in the RecyclerView should display the Date, corresponding weather icon for that day at daytime.
 3. 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 “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”.
8. There are two buttons in this activity.

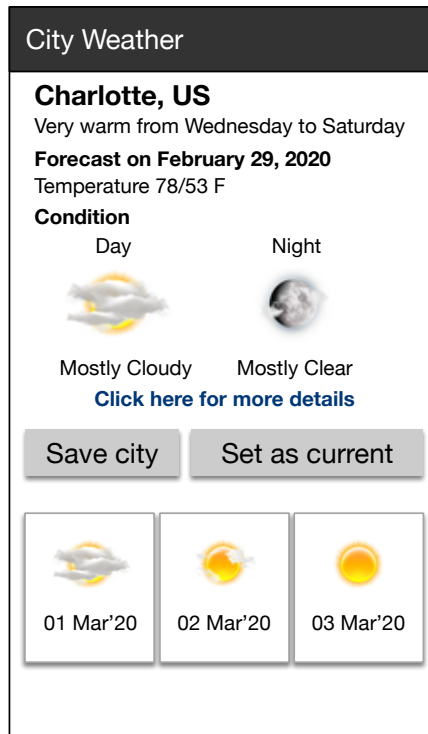


Figure 2, City Weather Activity

4. Save City: Pressing on this button should save the city's details (citykey, cityname, country, temperature, favorite) in the saved cities list. Subsequently return to the main activity.
 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. 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.
5. Set as current: Pressing this button should set the city as current city.
 1. If this city is marked as the current city, then update details and make a Toast displaying the message, "Current City Updated".
 2. If this city is not set as the current city, then set this city as the current city and make a Toast displaying the message, "Current City Saved".

Note:

1. Use SimpleDateFormat to parse the date and PrettyTime (<https://github.com/ocpsoft/prettytime>) library to display the last updated time.