**CMSC234 Project 5A**

**Derya O. Kurin**

**Algorithm:**

**Main Activity:**

Create private data fields for:

Add City Button

EditText cityName

ListView to hold the city names

ArrayList<String> for the adapter to hold the ListView

PROCESS onCreate:

Bind the view widgets

Register a click listener on add City button

On add it will update the adapter view to show the new city item

For ListView Items, set onLongClickListener and remove the long clicked item from the list

If savedInstanceState is null

Then beginTransaction on Fragment Manager for Weather Fragment

ENDPROCESS

Add a menu with the inflate method on menuInflater

PROCESS changeCity: void

Param: String city

Create an instance object for WeatherFragment

Call changeCity with the city argument and

Set the CityPreference as the arg city

ENDPROCESS

**Weather Fragment:**

**(This fragment is shown on activity main layout file)**

PROCESS: OnCreateView

Bundle SavedInstanceState

Create fields for holding the weather information fetched from the API service

OnCreate: get the city from the shared preferences file which is implemented in City Preferences

P.S: I used sharedPreferences to be able to freely get the city selected in different actitivies.

**CityPreference:**

Store the selected city in the shared preferences file for some freedom between activities. With this the data will be saved during the app runs.

**RemoteFetch:**

In the query string, the city's name is passed and the results should be returned in the imperial system for the Fahrenheit result. For Celsius the unit system is metric.

HttpURLConnection class is used to make the remote request. The OpenWeatherMap API expects the API key in an HTTP header named “x-api-key”. In setRequestProperty method this value is used.

BufferedReader is to read the API's response into a StringBuffer. Once a complete response is received, this response is converted into a JSONObject object.

Cod is a field in the API, and it should be 200 if the request becomes successful. This cod is used to check if the request was successful.

**Screen Shots:**

MainActivity has a ListView and also a Weather Fragment which is responsible for getting the weather details from the Open Weather API service.

The list items can be added by clicking on **Add City button**.

I did not use Remove city or Select city to change the weather information.

Instead of using these buttons I used **on click event listener that was registered on the list view items for selecting the city** to show, and **on long click listener for removing** the city from the list view.

Calling notifyDataSetChanged() method on the adapter view helped me to update the list view items dynamically on adding and removing cities.

A screenshot of a cell phone

Description automatically generated

The city Lisbon is removed by a long click on that city:

A screenshot of a cell phone

Description automatically generated

Clicking on Istanbul changes the city that is shown in the weather fragment for showing the weather details:

A screenshot of a cell phone

Description automatically generated

**Learning Experience**

I realized that once the connection with an API service is established, retrieving data from that service is not very difficult if the documentation is studied well.

I had some difficulties in setting the connection with the API service to get the data, since I was missing one step which was essential, using BufferedReader to create an InputStream on the connection object ( HttpURLConnection) Having solved that issue, I could get the data from the API service.

If I had more time, I would try to change the background image depending on the weather situation. For instance, I could get the description data and in a switch case, could set the background image accordingly. For example, for rainy weather a grey rainy background image or for fog, an image with fog could have been used.