Cs3300
Assignment 4
Anushervon Rakhmatov

Table of Contents

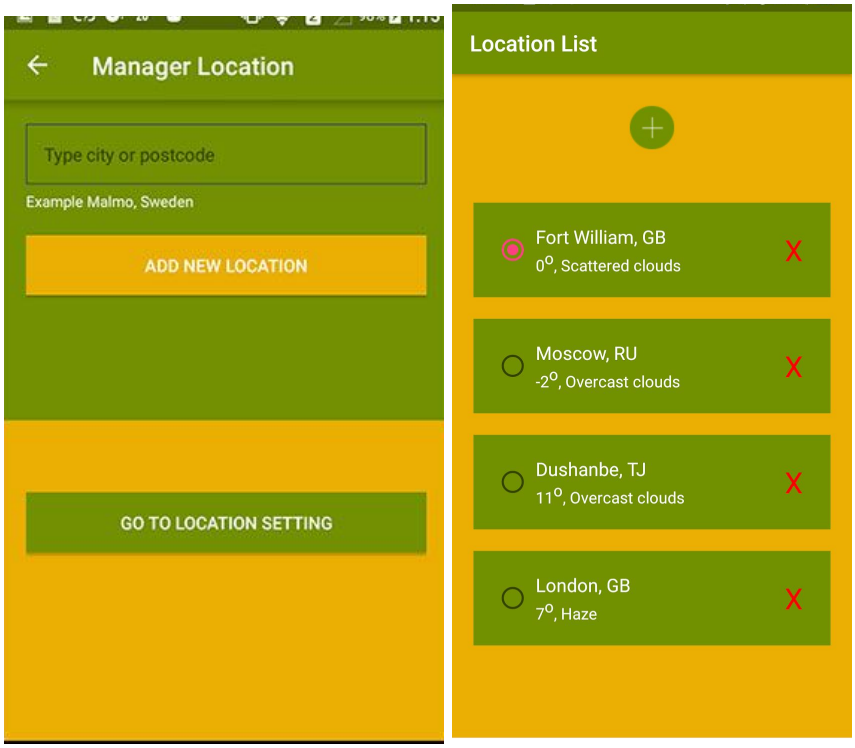
1. Project

Student name: Anushervon Rakhmatov
Student ID: 201356375
Project: Improved Clima(Clima PLUS)
Android Studio Version 3.2.1

How to run:
1.Download the project folder.
2.Open the Project through Android studio
3. Build and Run the app.

2. Improved Clima Design

- At least four locations can be compared simultaneously



- A user can add a new location by pressing “Add new location”
- After adding 3 locations, a user will click on “Go to location setting” in order to see all 4 locations
- To set the desirable location as current location, a user needs to click on the radio button next to the location desired.
- To delete one of four locations, a user can click on the cross icon.

- Your most recently used locations will be used by the app until there is a change
    - A location can be updated or changed by indicating lat/lon or by city name
    - A location can be updated to be the current location
    - persistence of locations between power cycles of the device is a desirable feature, but may be simulated for this project submission

- The current location remains persistent even after the restart or app reset.
- As mentioned in the previous point, a user can set one of four location to be the current location by clicking on the radio-button.

3. UX for improved Clima

The design pattern that I am using in this app resembles the combination of list menu pattern and dashboard pattern.

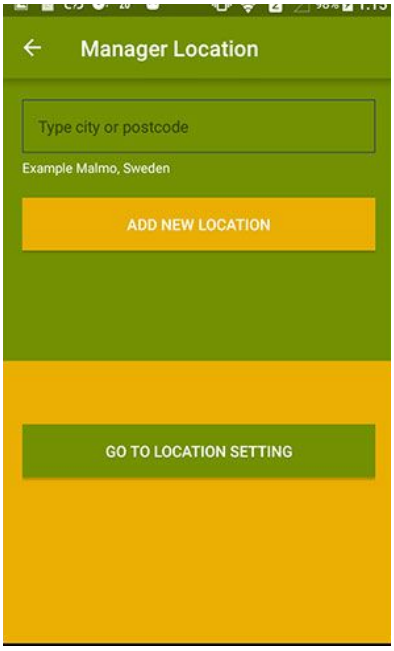
In the “add location” section, I have added an *Error Prevention* technique which prevents the users from moving to the next step of adding location. If the user clicks on ”Add new location” button without inputting data in the text box, then the toast message pops up notifying the user that he/she needs to input location first. This function clearly represents a error prevention technique of UX and provides an *Informative Feedback* which is another pillar of UX.

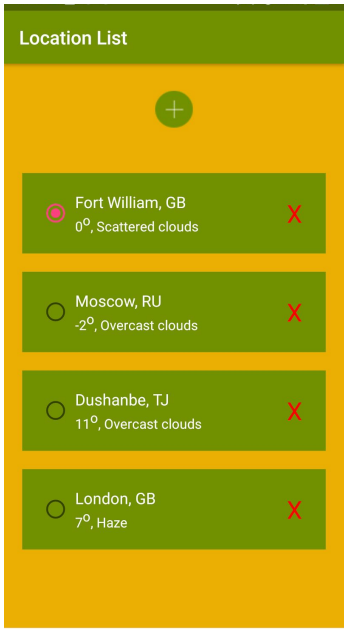
Another aspect of design that I want to motivate is the “Add Button” which represents the addition of additional location and can be found in the Main Weather layout and List of location layout. This button have an appearance of green circle with a “plus” sign inside, which in most cases represents the addition of the action in many applications. This button represents the UX pillar called *Form follows the Function*, since it performs the addition of an item. The same UX principle has been used in the List of Locations layout where the selectedd radio-button next to location represents that location has been selected and the “red cross” on the right side of the each location means that it can be “closed/deleted”, which makes it intuitive for a user.

In addition I’ve used some third party libraries in order to smoothen the look and the operation of the app.
The main libraries that helped to improve the UX of the Clima app are:
1. sdp-android:1.0.3 – for uniform dimension across different screen sizes
2. circleview:1.3 – display text inside a circle with coloured stroke

4. Prototype


St.Johns,Canada
Friday, 23 November
-6
Partly Cloud
WIND
23 km/h
HUMIDITY
78%


Manager Location
Type city or postcode
Example Malmo, Sweden
ADD NEW LOCATION
GO TO LOCATION SETTING


Location List
Fort William, GB
0°, Scattered clouds
Moscow, RU
-2°, Overcast clouds
Dushanbe, TJ
11°, Overcast clouds
London, GB
7°, Haze


Input field must be filled
?123
English

5. UX testing

Conduct a cognitive walkthrough using your prototype, and report your results. Follow the structure of a cognitive walkthrough as discussed in lecture. Make sure to include all the elements of cognitive walkthrough, including a description of your materials, procedure, setup, user profile, task description, your record of observations, notes and your conclusions. Also include a separate section in your conclusions to discuss the specific UX design aspect you presented in section 3.

The Improved Clima app that I named "Clima PLUS" consist of 4 main layouts:

1. A splash intro screen
2. Main Weather Activity Page
3. Add New Location Page – use to search for weather information about any location
4. List Location Page – that has a list of all the added locations.

We have decomposed the task into 11 main steps:

1. Launch the app.
2. Allow use your location.
3. Press add new location button(green button with cross).
4. Type in the city and country name.
5. Click add new location.
6. On a location list layout the click on a round green button to add another location.
7. Repeat step 5 three more times with different city and country name.
8. Click on "Go to location setting button".
9. Change the default location by clicking on a radio button next to one of the locations.
10. Delete one of the locations by clicking the X button on right side of each location box.
11. Add Another location by clicking on the green button on the top of the screen.

1. Launch the app.	Observations	UX/UI comments
Representation	Had to find the app before launching	State of system is clear from the context of the apps
Attention	All apps icon are standard	
Evaluation	See above	
Goal	Get the right app	
Intention	Go to the apps	
Specification	Click on the app	
Execution	Icons big, should be no exec problem	
2. Check "Allow your location"	**Observations**	**UX/UI comments**
Representation	It looks like a standard android notification	This is a familiar interaction for android devices, many apps do pop up similar notifications.
Attention	It pops up immediately after the app launch so there is no problem	
Evaluation	See above	
Goal	To allow the app use the current location	
Intention	To give the app access to the location usage	
Specification	Check the "Allow the use of location" and click "ok".	

Execution	Not hard to execute, only 2 clicks.	
3. Press add new location button	**Observations**	**UX/UI comments**
Representation	The button maybe a bit too small but there is nothing else resembling the button, so it is still okay.	Perhaps adding a descriptive text under the button would be helpful
Attention	The user will see an interface with current location weather, and go for the only green button to add another location.	
Evaluation	There is no explicit text under the button, so the user will need to rely on visuals.	
Goal	The goal is to add another location	
Intention	The user will try to find something to click on	
Specification	Press the round green button on a yellow background	
Execution	That is only button on the screen so no problem	
4. Type in the city and country name.	**Observations**	**UX/UI comments**
Representation	Instructional text inside the text box is useful	The prediction text functions proves to be useful.
Attention	The text box is big enough to see it	The example below the text box proves to be useful as well.
Evaluation	The typing cursor and popping cursor makes it clear	

Goal Intention Specification Execution	Define what location needs to be added Type City and Country name as shown in the example. As long as you follow the format of example, there should be no issues.	
5.Click to add new location.	**Observations**	**UX/UI comments**
Representation Attention Evaluation Goal Intention Specification Execution	The button is yellow colored which makes it easy to see it. The layout of the screen changes to another one Due to the above, the user should understand that the location has been added To add an additional location to be displayed The intention is to queue the location that has been typed Easy to notice and press since the button is big enough and visible	- Good color contrast on that button - Easy to notice the green button on a yellow background
6. In the location list layout press on a green button	**Observations**	**UX/UI comments**
Representation Attention	The button looks exactly the same as the button on the main layout. Easy to notice, as it looks there are no there buttons	Same button as in other layouts, which makes it very intuitive.

Evaluation Goal Intention Specification Execution	The button looks the same as other buttons, it is intuitive. The user will see one of the locations added, and would want to add another one. The user will try to add another location Click on a round green button with a cross inside it Easy to execute, as it requires only 1 click.	
7. Repeat step 5-6 three more times with different city and country name.	**Observations**	**UX/UI comments**
Representation Attention Evaluation Goal Intention Specification Execution	Same as in step 5-6. Same as in step 5-6. Same as in step 5-6. To have 4 location displayed at the same time. To add 3 additional locations. Repeat steps 5 to 6 three times. Easy to execute as it has been done once.	-The layout changes between the steps, which makes it clear and intuitive. -Same as in step 5 to 6.
8. Click on "Go to location setting button".	**Observations**	**UX/UI comments**

Representation Attention Evaluation Goal Intention Specification Execution	Located in a Location Manager layout, a green button on a yellow background Since the color of button is different from background color it is easy to interpret. The button is labeled so there is no problem with interpretation. Go to the location settings to delete or set the default location. Click on the "go to location setting button" Click on a green button that is labeled as "go to location setting button" The button is big enough to notice and click.	-Very distinctive button that makes it easy to notice.
9. Change the default location by clicking on a radio button next to one of the locations.	**Observations**	**UX/UI comments**
Representation Attention Evaluation Goal Intention	Located on the left side of the each location. Red-colored, can't be toggled. You don't really know what it does unless you have used the app once To set the location to be the current location.	This is a familiar interaction for android users.

Specification Execution	To set the location to be the default one Click on the radio button next to the location that you want to be a default one. Simple to execute as you just click on it.	
10.Delete one of the locations by clicking the X button on right side of each location box.	**Observations**	**UX/UI comments**
Representation Attention Evaluation Goal Intention Specification Execution	It looks like a cross on a right side of each location tab. It is easy to interpret the action as for android devices X represent close/delete It is easy to understand the system state as if the Red Cross is clicked the location will disappear. To delete the selected location. To remove the location from the list. Click on the red cross located on the right side of each location. It is intuitive and easy to execute.	Good choice of the icon and easy to execute.
11.Add Another location by clicking on the green button on the top of the screen.	**Observations**	**UX/UI comments**

Representation Attention Evaluation Goal Intention Specification Execution	Same as in step 5-6. Same as in step 5-6. This step is necessary to ensure that there 4 locations after deleting one in a previous step. To have 4 location displayed at the same time. To add 3 additional locations. Repeat steps 5 to 6 three times. Easy to execute as it has been done once.	-The layout changes between the steps, which makes it clear and intuitive. -Same as in step 5 to 6.

The user was a male psychology student, age 27, who is an active android user. The user was instructed to find the Clima Plus app, add 4 additional locations and set one of the locations to be a default one. It took around 20 seconds to find the app as it the icon blended very well among other apps. After clicking on the app the user was greeted with a main intro layout which introduces the app. After few seconds the app requested User to allow the access to the GPS, which user responded to by “Allow” option. After doing the screen changed to a weather layout where the toast message “Response good ” appeared indicating that the default location has been accessed. After few seconds of hesitation, the user clicked on a green “Add button” and was redirected to “Location Manager” layout. The user pressed “Add new location” button by mistake and got an appropriate toast message that stated that “Input must be filled”. After seeing the toast message, the user followed the example shown under the text box and entered the “Ottawa, Canada” and clicked on *Add New Location* button. Doing so changed the layout to the List of locations, where the user could see the box with a location, temperature and the weather description. There was not back button so the user clicked on the the green *add* button

which took him to the previous layout. The user repeat his previous actions 3 time until he could no longer add more locations to the limit set. After that the user tried to figure out how to change the location, and it took him about a minute to realize that the radio button next to each location is what allows you to set the default location. After setting the default location to Ottawa, the user attempted to return the main layout but had some troubles to do so due to the lack of “back” button in the *Location List layout.* However, based on his previous actions, he figured out that if he presses the green *Add* button, it will take him to the Location Manager layout where he found a *Back* button that took him to the main layout. Finally the user was asked to restart the phone to the the persistence of the location after the device reset. The test was successful as location was still showing “Ottawa”.

In conclusion, the test helpful to identify certain flaws in the UI such as a lack of back button and lack of text descriptions under the add button. There was also a certain bias since the tester was familiar with android devices hences he anticipated certain responses that are common among the android apps. The UX improvements that are implemented helped to maintain the minimalistic look and make the flow between the states of the app much smoother compared the original design.

6. Android Studio Project information and Code Listing

The project name:ImprovedClima
Andrioid Studio Version: 3.2.1.

Main Scripts used to implement Clima:

AndroidManifest.xml
build.gradle
AddLocationActivity.java
ListLocationActivity.java
MainActivity.java
WeatherActivity.java
Additional Assets located in “res” directory

- **AndroidManifest.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.ImprovedClima.androidweatherapp">

    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />

    <application
        android:name=".helpers.CustomApplication"
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="Clima PLUS"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".WeatherActivity" />
        <activity
            android:name=".AddLocationActivity"
            android:parentActivityName=".WeatherActivity">
        </activity>
        <activity android:name=".ListLocationActivity">
        </activity>
    </application>

</manifest>
```

- **build.gradle**

- To make the app better, I've added some third-party libraries such as:

1. sdp-android: – for uniform dimension across different screen sizes
2. circleview: – display text inside a circle with coloured stroke
3. gson:– for parsing Json to plain Java object
4. volley:library: – for android network call
5. sqliteassethelper – sqlite database helper classes
6. joda-time – working with date and time

```
android {
    compileSdkVersion 27
    buildToolsVersion '28.0.3'

    defaultConfig {
        applicationId "com.improvedClima.androidweatherapp"
        minSdkVersion 16
        targetSdkVersion 27
        versionCode 1
        versionName "1.0"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    implementation fileTree(include: ['*.jar'], dir: 'libs')
    testImplementation 'junit:junit:4.12'
    implementation 'com.intuit.sdp:sdp-android:1.0.3'
    implementation 'com.github.pavlospt:circleview:1.3'
    implementation 'com.google.code.gson:gson:2.6.1'
    implementation 'com.mcxiaoke.volley:library:1.0.19'
    implementation 'com.readystatesoftware.sqliteasset:sqliteassethelper:+'
    implementation 'com.google.guava:guava:19.0'
    implementation 'joda-time:joda-time:2.9.4'
    implementation 'com.android.support:appcompat-v7:27.1.1'
    implementation 'com.android.support:recyclerview-v7:27.1.1'
}
```

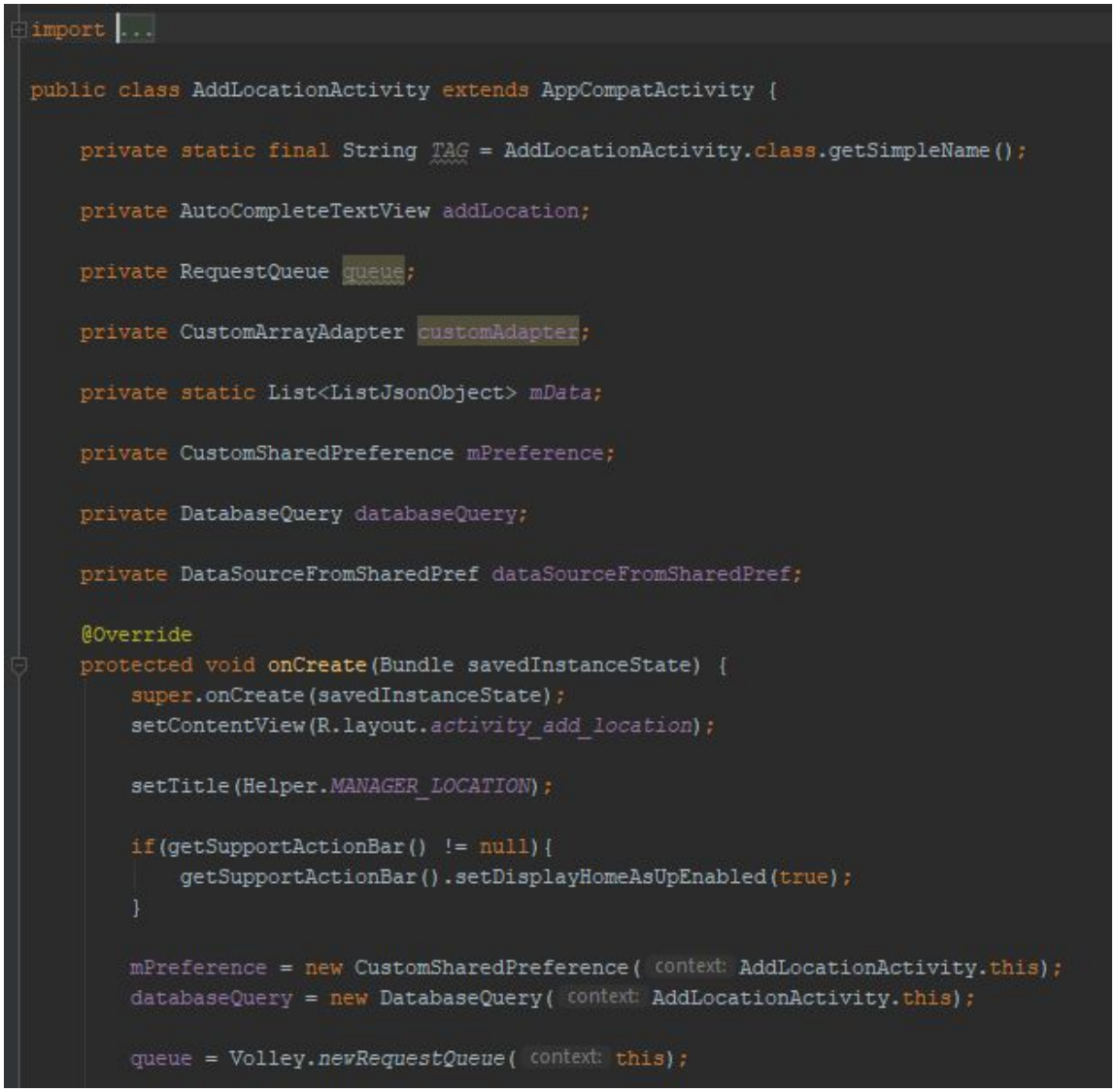
- **Strings.xml**

```
<resources>
    <string name="app_name">Clima PLUS</string>
    <string name="city_country">St.Johns,Canada</string>
    <string name="date_today">Friday, 23 November</string>
    <string name="current_temperature">-6</string>
    <string name="weather_information">Partly Cloud</string>
    <string name="temp_information">-6, partly cloudy</string>
    <string name="wind">WIND</string>
    <string name="wind_speed">23 km/h</string>
    <string name="humidity">HUMIDITY</string>
    <string name="humidity_rate">78%</string>
    <string name="day_of_week">Mon</string>
    <string name="location_hints">Type city or postcode</string>
    <string name="location_example">Example Malmo, Sweden</string>
    <string name="add_new_location">Add new location</string>
    <string name="stored_location">You can only store 5 locations</string>
    <string name="permission_notice">This permission is important but you can as well set your location on Location page</string>
    <string name="go_to_location">Go to location setting</string>
</resources>
```

- **colors.xml**

```
<resources>
    <color name="colorPrimary">#729101</color>
    <color name="colorPrimaryDark">#4f6404</color>
    <color name="colorAccent">#FF4081</color>
    <color name="colorBlack">#000000</color>
    <color name="colorWhite">#ffffff</color>
    <string name="delete_text">X</string>
    <color name="colorBackground">#eaaf02</color>
    <color name="colorBottomBackground">#729101</colo
    <color name="colorCircleStroke">#df9502</color>
    <color name="colorSubTitle">#38455B</color>
</resources>
```

- **addLocationActivity.java** : The AddLocationActivity class contains an EditText box where users can search for a location and save it if they want weather information about the location. It also give you location text suggestions about locations.



```
import ...

public class AddLocationActivity extends AppCompatActivity {

    private static final String TAG = AddLocationActivity.class.getSimpleName();

    private AutoCompleteTextView addLocation;

    private RequestQueue queue;

    private CustomArrayAdapter customAdapter;

    private static List<ListJsonObject> mData;

    private CustomSharedPreference mPreference;

    private DatabaseQuery databaseQuery;

    private DataSourceFromSharedPref dataSourceFromSharedPref;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_add_location);

        setTitle(Helper.MANAGER_LOCATION);

        if(getSupportActionBar() != null){
            getSupportActionBar().setDisplayHomeAsUpEnabled(true);
        }

        mPreference = new CustomSharedPreference( context: AddLocationActivity.this);
        databaseQuery = new DatabaseQuery( context: AddLocationActivity.this);

        queue = Volley.newRequestQueue( context: this);
```

```
            if(null != result){
                mData = result;
                customAdapter = new CustomArrayAdapter( context: AddLocationActivity.this, R.layout.city_list, mData);
                addLocation.setAdapter(customAdapter);
                addLocation.setThreshold(1);
            }
        }

        @Override
        protected List<ListJsonObject> doInBackground(Void... voids) { return mergeStoredData(); }
    }

    @Override
    protected void onResume() {
        super.onResume();
        if(null == mData){
            dataSourceFromSharedPref = new DataSourceFromSharedPref();
            dataSourceFromSharedPref.execute();
        }
    }

    @Override
    protected void onDestroy() {
        super.onDestroy();
        if(null != null){
            dataSourceFromSharedPref.cancel( mayInterruptIfRunning: true);
        }
    }

    private List<ListJsonObject> mergeStoredData(){
        List<ListJsonObject> firstObject = mPreference.getAllDataObject(Helper.STORED_DATA_FIRST);
        List<ListJsonObject> secondObject = mPreference.getAllDataObject(Helper.STORED_DATA_SECOND);
        return Lists.newArrayList(Iterables.concat(firstObject, secondObject));
    }
}
```

```
    addLocation = (AutoCompleteTextView) findViewById(R.id.new_location);

    Button goToLocationButton = (Button)findViewById(R.id.go_to_location_button);
    goToLocationButton.setOnClickListener((view) -> {
            Intent listLocationIntent = new Intent( packageContext: AddLocationActivity.this, ListLocationActivity.class);
            startActivity(listLocationIntent);
    });

    final Button addLocationButton = (Button)findViewById(R.id.add_location_button);
    addLocationButton.setOnClickListener((view) -> {
            String enteredLocation = addLocation.getText().toString();
            if (TextUtils.isEmpty(enteredLocation)) {
                Toast.makeText( context: AddLocationActivity.this, Helper.LOCATION_ERROR_MESSAGE, Toast.LENGTH_LONG).show();
                return;
            }
            // add this to the database
            int numOfLocationsStored = databaseQuery.countAllStoredLocations();
            Toast.makeText( context: AddLocationActivity.this,  text: "Total count " + numOfLocationsStored, Toast.LENGTH_LONG).show();
            if(numOfLocationsStored <= 3){
                databaseQuery.insertNewLocation(enteredLocation);
            }else{
                Toast.makeText( context: AddLocationActivity.this, "You can only store 5 locations", Toast.LENGTH_LONG).show();
            }

            Intent listLocationIntent = new Intent( packageContext: AddLocationActivity.this, ListLocationActivity.class);
            startActivity(listLocationIntent);
    });
}

private class DataSourceFromSharedPref extends AsyncTask<Void, Void, List<ListJsonObject>> {

    protected void onProgressUpdate() {
    }

    protected void onPostExecute(List<ListJsonObject> result) {
        if(null != result){
```

- **Activity_add_location.xml** : the corresponding layout for the above script

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="@color/colorBackground"
    tools:context="com.inducesmile.androidweatherapp.AddLocationActivity">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="5"
        android:padding="38.40dp"
        android:background="@color/colorPrimary"
        android:orientation="vertical">

        <AutoCompleteTextView
            android:id="@+id/new_location"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="Type city or postcode"
            android:inputType="textPersonName"
            android:maxLines="1"
            android:padding="16.80dp"
            android:textSize="16.80dp"
            android:background="@drawable/bottom_border"
            android:textColor="@color/colorWhite"
            android:singleLine="true" />
```

```
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textSize="17.00dp"
            android:text="Example Malmo, Sweden"
            android:layout_marginTop="5.20dp"
            android:textColor="@color/colorWhite"/>

        <Button
            android:id="@+id/add_location_button"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="38.40dp"
            android:textColor="@color/colorWhite"
            android:text="Add new location"
            android:background="@color/colorBackground"
            android:padding="16.80dp"/>

    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="5"
        android:padding="38.40dp"
        android:orientation="vertical">

        <Button
            android:id="@+id/go_to_location_button"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="35.20dp"
            android:layout_gravity="center"
            android:textColor="@color/colorWhite"
            android:text="Go to location setting"
            android:background="@color/colorPrimary"
```

- **ListLocationActivity.java:** In this page, we list all the locations we have stored. We can delete or select each location at a time.

```
public class ListLocationActivity extends AppCompatActivity {

    private static final String TAG = ListLocationActivity.class.getSimpleName();

    private DatabaseQuery query;

    private List<DatabaseLocationObject> allLocations;

    private LocationObject locationObject;

    private LocationMapObject locationMapObject;

    private RequestQueue queue;

    private List<LocationObject> allData;

    private LocationAdapter locationAdapter;

    private RecyclerView locationRecyclerView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_list_location);

        setTitle(Helper.LOCATION_LIST);

        queue = Volley.newRequestQueue( context: ListLocationActivity.this);
        allData = new ArrayList<LocationObject>();

        query = new DatabaseQuery( context: ListLocationActivity.this);
        allLocations = query.getStoredDataLocations();

        if(null != allLocations){
            for(int i = 0; i < allLocations.size(); i++){
                // make volley network call here
                System.out.println("Response printing " + allLocations.get(i).getLocation());
```

```
    Toast.makeText( context: ListLocationActivity.this,  text: "Count number of locations " + allLocations.size(), Toast.LENGTH_LONG).show();

    ImageButton addLocation = (ImageButton) findViewById(R.id.add_location);
    addLocation.setOnClickListener((view) -> {
            Intent addLocationIntent = new Intent( packageContext: ListLocationActivity.this, AddLocationActivity.class);
            startActivity(addLocationIntent);
    });

    LinearLayoutManager linearLayoutManager = new LinearLayoutManager( context: ListLocationActivity.this);
    locationRecyclerView = (RecyclerView) findViewById(R.id.location_list);
    locationRecyclerView.setLayoutManager(linearLayoutManager);
}

private void requestJsonObject(final DatabaseLocationObject paramValue){
    String url ="http://api.openweathermap.org/data/2.5/weather?q="+paramValue.getLocation()+"&APPID=d18ad530460091cdecd58e1a86e79054"+Helper.API_KEY+"
    StringRequest stringRequest = new StringRequest(Request.Method.GET, url, (response) -> {
            Log.d(TAG,  msg: "Response " + response);
            GsonBuilder builder = new GsonBuilder();
            Gson gson = builder.create();
            locationMapObject = gson.fromJson(response, LocationMapObject.class);
            if (null == locationMapObject) {
                Toast.makeText(getApplicationContext(),  text: "Nothing was returned", Toast.LENGTH_LONG).show();
            } else {
                int rowId = paramValue.getId();
                Long tempVal = Math.round(Math.floor(Double.parseDouble(locationMapObject.getMain().getTemp())));
                String city = locationMapObject.getName() + ", " + locationMapObject.getSys().getCountry();
                String weatherInfo = String.valueOf(tempVal) + "<sup>o</sup>, " + Helper.capitalizeFirstLetter(locationMapObject.getWeather().get(0).ge
                allData.add(new LocationObject(rowId, city, weatherInfo));

                locationAdapter = new LocationAdapter( context: ListLocationActivity.this, allData);
                locationRecyclerView.setAdapter(locationAdapter);
            }
    }, (error) -> { Log.d(TAG,  msg: "Error " + error.getMessage()); });
    queue.add(stringRequest);
}
```

- **Activitylist_location.xml**: The corresponding layout for the above class

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/colorBackground"
    tools:context="com.inducesmile.androidweatherapp.ListLocationActivity">

    <ImageButton
        android:id="@+id/add_location"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="@android:color/transparent"
        android:layout_centerHorizontal="true"
        android:layout_alignParentTop="true"
        android:elevation="5.20dp"
        android:layout_margin="33.60dp"
        android:src="@drawable/cross"/>

    <android.support.v7.widget.RecyclerView
        android:id="@+id/location_list"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_below="@+id/add_location"
        android:layout_centerHorizontal="true"
        android:orientation="vertical"
        android:scrollbars="none"/>

</RelativeLayout>
```

- **WeatherActivity.java:** This is the main weather class that display all Weather information. Create a WeatherActivity.java file and add the code below to the file.

```
public class WeatherActivity extends AppCompatActivity implements LocationListener {

    private static final String TAG = WeatherActivity.class.getSimpleName();

    private RecyclerView recyclerView;

    private RecyclerViewAdapter recyclerViewAdapter;

    private TextView cityCountry;

    private TextView currentDate;

    private ImageView weatherImage;

    private CircleView circleTitle;

    private TextView windResult;

    private TextView humidityResult;

    private RequestQueue queue;

    private LocationMapObject locationMapObject;

    private LocationManager locationManager;

    private Location location;

    private final int REQUEST_LOCATION = 200;

    private CustomSharedPreference sharedPreference;

    private String isLocationSaved;

    private DatabaseQuery query;

    private String apiUrl;

    private FiveDaysForecast fiveDaysForecast;
```

```
        }else{
            // make API call with city name
            String storedCityName = sharedPreference.getLocationInPreference();
            //String storedCityName = "Enugu";
            System.out.println("Stored city " + storedCityName);
            String[] city = storedCityName.split( regex: ",");
            if(!TextUtils.isEmpty(city[0])){
                System.out.println("Stored city " + city[0]);
                String url ="http://api.openweathermap.org/data/2.5/weather?q="+city[0]+"&APPID=d18ad53046009lcdecd58e1a86e79054"+Helper.API
                makeJsonObject(url);
            }
        }
    }

    ImageButton addLocation = (ImageButton) findViewById(R.id.add_location);
    addLocation.setOnClickListener((view) -> {
            Intent addLocationIntent = new Intent( packageContext: WeatherActivity.this, AddLocationActivity.class);
            startActivity(addLocationIntent);
    });

    GridLayoutManager gridLayoutManager = new GridLayoutManager( context: WeatherActivity.this,  spanCount: 4);

    recyclerView = (RecyclerView)findViewById(R.id.weather_daily_list);
    recyclerView.setLayoutManager(gridLayoutManager);
    recyclerView.setHasFixedSize(true);



rivate void makeJsonObject(final String apiUrl){
    StringRequest stringRequest = new StringRequest(Request.Method.GET, apiUrl, (response) -> {
            Log.d(TAG,  msg: "Response " + response);
            GsonBuilder builder = new GsonBuilder();
            Gson gson = builder.create();
            locationMapObject = gson.fromJson(response, LocationMapObject.class);
            if (null == locationMapObject) {
                Toast.makeText(getApplicationContext(),  text: "Nothing was returned", Toast.LENGTH_LONG).show();
            } else {
                Toast.makeText(getApplicationContext(),  text: "Response Good", Toast.LENGTH_LONG).show();

                String city = locationMapObject.getName() + ", " + locationMapObject.getSys().getCountry();
```

```
Override
rotected void onCreate(Bundle savedInstanceState) {
    requestWindowFeature(Window.FEATURE_NO_TITLE);
    getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN, WindowManager.LayoutParams.FLAG_FULLSCREEN);
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_weather);

    ActionBar actionBar = getSupportActionBar();
    if(actionBar != null){
        actionBar.hide();
    }

    queue = Volley.newRequestQueue( context: this);
    query = new DatabaseQuery( context: WeatherActivity.this);
    sharedPreference = new CustomSharedPreference( context: WeatherActivity.this);
    isLocationSaved = sharedPreference.getLocationInPreference();

    cityCountry = (TextView)findViewById(R.id.city_country);
    currentDate = (TextView)findViewById(R.id.current_date);
    weatherImage = (ImageView)findViewById(R.id.weather_icon);
    circleTitle = (CircleView)findViewById(R.id.weather_result);
    windResult = (TextView)findViewById(R.id.wind_result);
    humidityResult = (TextView)findViewById(R.id.humidity_result);

    locationManager = (LocationManager) getSystemService(Service.LOCATION_SERVICE);
    if (ActivityCompat.checkSelfPermission( context: this, Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED && ActivityCompat.che
        ActivityCompat.requestPermissions( activity: WeatherActivity.this, new String[]{Manifest.permission.ACCESS_COARSE_LOCATION, Manifest.permission.ACCESS_F
    } else {
        if(isLocationSaved.equals("")){
            // make API call with longitude and latitude
            locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER,  minTime: 100,  minDistance: 2,  listener: this);
            if (locationManager != null) {
                location = locationManager.getLastKnownLocation(LocationManager.NETWORK_PROVIDER);
                apiUrl = "http://api.openweathermap.org/data/2.5/weather?lat="+location.getLatitude()+"&lon="+location.getLongitude()+"&APPID=d18ad5304600911
                makeJsonObject(apiUrl);
            }
```

```
                }
            }
        }else{
            Toast.makeText( context: WeatherActivity.this, "This permission is important but you can as well set yo...", Toast.LENGTH_LONG).show();
        }
    }
}

@Override
public void onLocationChanged(Location location) { this.location = location; }

@Override
public void onStatusChanged(String s, int i, Bundle bundle) {

}

@Override
public void onProviderEnabled(String s) {

}

@Override
public void onProviderDisabled(String provider) {
    if (provider.equals(LocationManager.GPS_PROVIDER)) {
        showGPSDisabledAlertToUser();
    }
}

private void showGPSDisabledAlertToUser() {
    AlertDialog.Builder alertDialogBuilder = new AlertDialog.Builder( context: this);
    alertDialogBuilder.setMessage("GPS is disabled in your device. Would you like to enable it?")
            .setCancelable(false)
            .setPositiveButton( text: "Goto Settings Page To Enable GPS", (dialog, id) -> {
                    Intent callGPSSettingIntent = new Intent(android.provider.Settings.ACTION_LOCATION_SOURCE_SETTINGS);
                    startActivity(callGPSSettingIntent);
            });
    alertDialogBuilder.setNegativeButton( text: "Cancel", (dialog, id) -> {
            dialog.cancel();
    });
```

```
            String todayDate = getTodayDateInStringFormat();
            Long tempVal = Math.round(Math.floor(Double.parseDouble(locationMapObject.getMain().getTemp())));
            String weatherTemp = String.valueOf(tempVal) + "°";
            String weatherDescription = Helper.capitalizeFirstLetter(locationMapObject.getWeather().get(0).getDescription());
            String windSpeed = locationMapObject.getWind().getSpeed();
            String humidityValue = locationMapObject.getMain().getHumudity();

            //save location in database
            if(apiUrl.contains("lat")){
                query.insertNewLocation(locationMapObject.getName());
            }
            // populate View data
            cityCountry.setText(Html.fromHtml(city));
            currentDate.setText(Html.fromHtml(todayDate));
            circleTitle.setTitleText(Html.fromHtml(weatherTemp).toString());
            circleTitle.setSubtitleText(Html.fromHtml(weatherDescription).toString());
            windResult.setText(Html.fromHtml(windSpeed) + " km/h");
            humidityResult.setText(Html.fromHtml(humidityValue) + " %");

            fiveDaysApiJsonObjectCall(locationMapObject.getName());
        }
    }, (error) -> { Log.d(TAG, msg: "Error " + error.getMessage()); });
    queue.add(stringRequest);


Override
ublic void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {
    if (requestCode == REQUEST_LOCATION) {
        if (grantResults[0] == PackageManager.PERMISSION_GRANTED && grantResults[1] == PackageManager.PERMISSION_GRANTED) {

            if (ActivityCompat.checkSelfPermission( context: this, Manifest.permission.ACCESS_FINE_LOCATION) == PackageManager.PERMISSION_GRANTED && Activit
                //make api call
                locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER,  minTime: 100,  minDistance: 2,  listener: this);
                if (locationManager != null) {
                    location = locationManager.getLastKnownLocation(LocationManager.NETWORK_PROVIDER);
                    apiUrl = "http://api.openweathermap.org/data/2.5/weather?lat="+location.getLatitude()+"&lon="+location.getLongitude()+"&APPID=d18ad530
                    makeJsonObject(apiUrl);
                }else{
                    apiUrl = "http://api.openweathermap.org/data/2.5/weather?lat=51.5074&lon=0.1278&APPID=d18ad530460091cdecd58e1a86e79054"+Helper.API_KEY
                    makeJsonObject(apiUrl);
```

```
        AlertDialog alert = alertDialogBuilder.create();
        alert.show();
    }

    private String getTodayDateInStringFormat(){
        Calendar c = Calendar.getInstance();
        SimpleDateFormat df = new SimpleDateFormat( pattern: "E, d MMMM", Locale.getDefault());
        return df.format(c.getTime());
    }

    private void fiveDaysApiJsonObjectCall(String city){
        String apiUrl = "http://api.openweathermap.org/data/2.5/forecast?q="+city+ "&APPID=d18ad530460091cdecd58e1a86e79054"+Helper.API_KEY+"&units=metric";
        final List<WeatherObject> daysOfTheWeek = new ArrayList<>();
        StringRequest stringRequest = new StringRequest(Request.Method.GET, apiUrl, (response) -> {
                Log.d(TAG, msg: "Response 5 days" + response);
                GsonBuilder builder = new GsonBuilder();
                Gson gson = builder.create();
                Forecast forecast = gson.fromJson(response, Forecast.class);
                if (null == forecast) {
                    Toast.makeText(getApplicationContext(), text: "Nothing was returned", Toast.LENGTH_LONG).show();
                } else {
                    Toast.makeText(getApplicationContext(), text: "Response Good", Toast.LENGTH_LONG).show();

                    int[] everyday = new int[]{0,0,0,0,0,0,0};

                    List<FiveWeathers> weatherInfo = forecast.getList();
                    if(null != weatherInfo){
                    for(int i = 0; i < weatherInfo.size(); i++){
                        String time = weatherInfo.get(i).getDt_txt();
                        String shortDay = convertTimeToDay(time);
                        String temp = weatherInfo.get(i).getMain().getTemp();
                        String tempMin = weatherInfo.get(i).getMain().getTemp_min();

                        if(convertTimeToDay(time).equals("Mon") && everyday[0] < 1){
                            daysOfTheWeek.add(new WeatherObject(shortDay, R.drawable.small_weather_icon, temp, tempMin));
                            everyday[0] = 1;
                        }
                        if(convertTimeToDay(time).equals("Tue") && everyday[1] < 1){
                            daysOfTheWeek.add(new WeatherObject(shortDay, R.drawable.small_weather_icon, temp, tempMin));
                            everyday[1] = 1;
```

```
                    }
                }
            }else{
                Toast.makeText( context: WeatherActivity.this, "This permission is important but you can as well set yo...", Toast.LENGTH_LONG).show();
            }
        }
    }

    @Override
    public void onLocationChanged(Location location) { this.location = location; }

    @Override
    public void onStatusChanged(String s, int i, Bundle bundle) {

    }

    @Override
    public void onProviderEnabled(String s) {

    }

    @Override
    public void onProviderDisabled(String provider) {
        if (provider.equals(LocationManager.GPS_PROVIDER)) {
            showGPSDisabledAlertToUser();
        }
    }

    private void showGPSDisabledAlertToUser() {
        AlertDialog.Builder alertDialogBuilder = new AlertDialog.Builder( context: this);
        alertDialogBuilder.setMessage("GPS is disabled in your device. Would you like to enable it?")
                .setCancelable(false)
                .setPositiveButton( text: "Goto Settings Page To Enable GPS", (dialog, id) -> {
                        Intent callGPSSettingIntent = new Intent(android.provider.Settings.ACTION_LOCATION_SOURCE_SETTINGS);
                        startActivity(callGPSSettingIntent);
                });
        alertDialogBuilder.setNegativeButton( text: "Cancel", (dialog, id) -> {
                dialog.cancel();
        });
```

```
                    }
                    if(convertTimeToDay(time).equals("Fri") && everyday[4] < 1){
                        daysOfTheWeek.add(new WeatherObject(shortDay, R.drawable.small_weather_icon, temp, tempMin));
                        everyday[4] = 1;
                    }
                    if(convertTimeToDay(time).equals("Sat") && everyday[5] < 1){
                        daysOfTheWeek.add(new WeatherObject(shortDay, R.drawable.small_weather_icon, temp, tempMin));
                        everyday[5] = 1;
                    }
                    if(convertTimeToDay(time).equals("Sun") && everyday[6] < 1){
                        daysOfTheWeek.add(new WeatherObject(shortDay, R.drawable.small_weather_icon, temp, tempMin));
                        everyday[6] = 1;
                    }
                    recyclerViewAdapter = new RecyclerViewAdapter( context: WeatherActivity.this, daysOfTheWeek);
                    recyclerView.setAdapter(recyclerViewAdapter);
                }
            }
        }
    }, (error) -> { Log.d(TAG,  msg: "Error " + error.getMessage()); });
    queue.add(stringRequest);
}

private String convertTimeToDay(String time){
    SimpleDateFormat format = new SimpleDateFormat( pattern: "yyyy-MM-dd HH:mm:SSSS", Locale.getDefault());
    String days = "";
    try {
        Date date = format.parse(time);
        System.out.println("Our time " + date);
        Calendar calendar = Calendar.getInstance();
        calendar.setTime(date);
        days = calendar.getDisplayName(Calendar.DAY_OF_WEEK, Calendar.SHORT, Locale.getDefault());
        System.out.println("Our time " + days);
    } catch (ParseException e) {
        e.printStackTrace();
    }
    return days;
}
```

- **Activity_weather.xml** - contains a layout for the above script

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="@color/colorBackground"
    tools:context="com.inducesmile.androidweatherapp.WeatherActivity">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="8"
        android:orientation="vertical">

        <TextView
            android:id="@+id/city_country"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center_horizontal"
            android:text="St.Johns,Canada"
            android:textSize="33.60dp"
            android:layout_marginTop="38.40dp"
            android:textColor="@color/colorWhite"/>

        <TextView
            android:id="@+id/current_date"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center_horizontal"
            android:text="Friday, 23 November"
            android:textSize="28.00dp"
            android:layout_marginTop="19.20dp"
            android:textColor="@color/colorWhite"/>

        <ImageView
            android:id="@+id/weather_icon"
            android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:contentDescription="Android Weather App"
        android:src="@drawable/sun"
        android:layout_marginTop="33.60dp"/>

    <com.github.pavlospt.CircleView
        android:id="@+id/weather_result"
        android:layout_width="165.00dp"
        android:layout_height="165.00dp"
        android:layout_marginTop="19.20dp"
        app:cv_titleSubtitleSpace="40"
        app:cv_fillColor="@color/colorBackground"
        app:cv_strokeColorValue="@color/colorCircleStroke"
        app:cv_backgroundColorValue="@color/colorCircleStroke"
        app:cv_titleColor="@color/colorWhite"
        app:cv_titleSize="95.00dp"
        app:cv_titleText="-6"
        app:cv_subtitleSize="16.80dp"
        app:cv_subtitleText="Partly Cloud"
        android:layout_gravity="center_horizontal"/>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:weightSum="3"
        android:layout_marginTop="16.80dp"
        android:layout_gravity="center_horizontal">

        <LinearLayout
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:orientation="vertical"
            android:paddingLeft="38.40dp"
            android:layout_weight="1">

            <TextView
                android:layout_width="wrap_content"
```

```
            android:textColor="@color/colorSubTitle"
            android:textSize="28.00dp"
            android:textStyle="bold"
            android:text="WIND"/>

        <TextView
            android:id="@+id/wind_result"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textColor="@color/colorWhite"
            android:textSize="28.00dp"
            android:layout_marginTop="5.20dp"
            android:text="23 km/h"/>

    </LinearLayout>

    <LinearLayout
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:layout_weight="1">

        <ImageButton
            android:id="@+id/add_location"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center_horizontal"
            android:background="@android:color/transparent"
            android:elevation="5.20dp"
            android:layout_marginTop="16.80dp"
            android:src="@drawable/cross"/>

    </LinearLayout>

    <LinearLayout
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:orientation="vertical"
```
adfa

```
                android:textSize="28.00dp"
                android:layout_gravity="right"
                android:textStyle="bold"
                android:text="HUMIDITY"/>

            <TextView
                android:id="@+id/humidity_result"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:textColor="@color/colorWhite"
                android:layout_gravity="center_horizontal"
                android:textSize="28.00dp"
                android:layout_marginTop="5.20dp"
                android:text="78%"/>

        </LinearLayout>

    </LinearLayout>

/LinearLayout>

LinearLayout
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="2"
    android:background="@color/colorBottomBackground"
    android:orientation="vertical">

    <android.support.v7.widget.RecyclerView
        android:id="@+id/weather_daily_list"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_gravity="center_horizontal"
        android:orientation="vertical"
        android:scrollbars="none"/>

/LinearLayout>

earLayout>
```

- For the adapter classes, we will be using the following layouts.
  City_LIST.XML

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <TextView
        android:id="@+id/text_suggestion"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textColor="@color/colorBlack"
        android:padding="@dimen/_16sdp"
        android:text="@string/app_name"/>

</LinearLayout>
```

LOCATION_LIST.XML

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="38.40dp"
    android:layout_marginLeft="38.40dp"
    android:layout_marginRight="38.40dp"
    android:padding="38.40dp"
    android:background="@color/colorPrimary"
    android:orientation="horizontal">

    <LinearLayout
        android:id="@+id/radio_button_wrapper"
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:orientation="vertical">

        <RadioButton
            android:id="@+id/radio_button"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_gravity="center" />

    </LinearLayout>

    <LinearLayout
        android:id="@+id/location_wrapper"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="8"
        android:layout_marginLeft="19.20dp"
        android:orientation="vertical">

        <TextView
            android:id="@+id/city_location"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textSize="28.00dp"
```

- WEATHER_DAILY_LIST.XML

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:paddingLeft="@dimen/_16sdp"
    android:paddingRight="@dimen/_16sdp"
    android:paddingBottom="@dimen/_4sdp"
    android:paddingTop="@dimen/_4sdp"
    android:orientation="vertical">

    <TextView
        android:id="@+id/day_of_week"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:textSize="@dimen/_12sdp"
        android:text="@string/day_of_week"
        android:textColor="@color/colorWhite"/>

    <ImageView
        android:id="@+id/weather_icon"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:contentDescription="@string/app_name"
        android:src="@drawable/small_weather_icon"
        android:layout_marginTop="@dimen/_4sdp"
        android:layout_gravity="center_horizontal"/>

    <TextView
        android:id="@+id/weather_result"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:layout_marginTop="@dimen/_4sdp"
        android:textSize="@dimen/_11sdp"
        android:text="@string/current_temperature"
        android:textColor="@color/colorWhite"/>
```