

CHAPTER 1

INTRODUCTION

Weather forecasting is the application of science and technology to predict the conditions of the atmosphere for a given location and time. Weather forecasts are made by collecting quantitative data about the current state of the atmosphere at a given place and using meteorology to project how the atmosphere will change. The role of Technology has been remarkable in the field of weather forecasting. Weather data is not only necessary for researchers or scientists, ordinary people can be benefitted from it as well. People nowadays are feeling the necessity of weather data as well. There are a variety of weather mobile apps in Google Play and the App store. Those apps have great features and functionalities to satisfy users. However, only a few of them have friendly user interface and human centered interactions, which means that a lot of them might be difficult to be navigated even though they provide enough functionalities. It is not convenient for new users. Therefore, we would like to do improvements on weather mobile apps. It is basically for Apple smart phones and tablets. For centuries, people have tried to forecast the weather informally, and officially since the nineteenth century. Weather forecasting, which used to be done by hand and was focused mostly on variations in barometric pressure, existing weather patterns, and sky state or cloud cover, is now done using computer-based models that account for a variety of atmospheric variables. Weather predictions are created by gathering objective data about the actual condition of the atmosphere at a certain location and using meteorology to predict how the weather will behave in the future. Human feedback is also required to choose the best possible forecast model on which to base the forecast. Weather forecasting is a part of the economy; for example, the United States spent \$5.1 billion on weather forecasting in 2009, with gains expected to be six times that amount. Since we know the weather forecast, let us take a look at the importance of weather forecasting pdf and the different methods used to forecast.

CHAPTER 2

REQUIREMENT SPECIFICATION

2.1 Specific requirements

A Software requirements definition is an abstract description of the service which the system should provide, and the constraints and which system must operate. it should only specify the external behavior of the system.

2.1.1 User Requirement

- Easy to understand and should be simple.
- The built-in functions should be utilized to maximum extent.

2.1.2 Hardware Constraints

- Processor : intel
- RAM : 512MB
- Hard Disk : 20GB(approx)

2.1.3 Software Constraints

- Operating System : Windows 10/2000/XP/Vista/UBUNTU
- Language : XML,JAVA
- Compiler : Android studio

CHAPTER 3

METHODOLOGY/WORKING

3.1 Persistence forecasting:

- Persistence forecasting is the easiest method of forecasting which assumes a continuation of the present. It relies upon today's conditions to forecast the weather when it is steady state, such as during the summer season in the tropics. This method of forecasting strongly depends upon the presence of a stagnant weather pattern. It can be useful in both short-range forecasts and long-range forecasts.
- Persistence forecasts are used by local forecasters in determining such events as the time of the arrival of a thunderstorm that is moving toward their region. Persistence forecasts do not account for changes that might occur in the intensity or in the path of a weather system, and they do not predict the formation. Because of these limitations and the rapidity with which weather system change in most geographical regions, persistence forecasts break down after twelve hours, or a day at most.

3.2 Climatology forecasting:

- Whereas persistence forecasting is most accurate over short periods (before factors for change have had time to operate), the best estimate of the weather a long time ahead is the average value of past measurements there at that time of day and year
- Climatology forecast relies on the observation that weather for a particular day at a location does not change much from one year to the next. As a result, a long term average of weather on a certain day or month should be a good guess as the weather for that day or month. The most obvious climatology forecast in this part of the world (Nigeria) is, "Cold in December, warm in July (the popular July break)". One does not need to be a meteorologist to make that forecast.

3.3 Nowcasting:

- The forecasting of the weather within the next six hours is often referred to as nowcasting. In this time range, it is possible to forecast smaller features such as individual showers and thunderstorms with reasonable accuracy, as well as other features too small to be resolved by a computer model.
- A human given the latest radar, satellite and observational data will be able to make a better analysis of the small scale features present and so will be able to make a more accurate forecast for the following few hours. Severe weather is typically short-lived (less than two hours) and, due to its mesoscale character (less than one hundred kilometers), it affects local/regional areas necessitating site-specific forecasts. Included in this category are thunderstorms, gust fronts, tornadoes, high winds especially along coasts, over lakes and mountains, heavy snow and freezing precipitation. The development of radar networks, new instruments and high speed communication links has provided a means of issuing warnings of such phenomena.

3.3 Methodology Chart

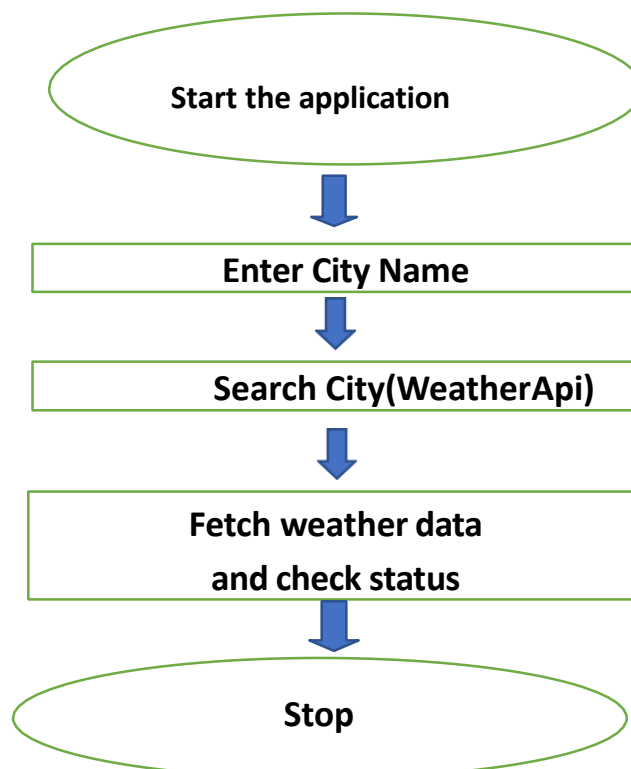


Figure 1.0 Methodology chart draft

CHAPTER 4

IMPLEMENTATION

MainActivity.java

```
package com.example.weatherapp;

import androidx.appcompat.app.AppCompatActivity;

import android.location.Location;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import com.luckycatlabs.sunrisesunset.SunriseSunsetCalculator;
import com.squareup.picasso.Picasso;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Locale;
```

```
public class MainActivity extends AppCompatActivity {  
    EditText editText;  
    Button button;  
    ImageView imageView;  
    TextView temptv, time, country, city_nam, max_temp, min_temp;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        editText = findViewById(R.id.editTextTextPersonName);  
        button = findViewById(R.id.button);  
        imageView = findViewById(R.id.imageView);  
        temptv = findViewById(R.id.textView3);  
        time = findViewById(R.id.textView2);  
        country = findViewById(R.id.country);  
        city_nam = findViewById(R.id.city_nam);  
        max_temp = findViewById(R.id.temp_max);  
        min_temp = findViewById(R.id.min_temp);  
  
        button.setOnClickListener(new View.OnClickListener()  
        {  
            @Override  
            public void onClick(View v)  
            {  
                FindWeather();  
            }  
        });  
    }  
    public void FindWeather()  
    {  
        final String city = editText.getText().toString();
```

```
Stringurl="http://api.openweathermap.org/data/2.5/weather?q="+city+"&appid=462f445106adc1d21494341838c10019&units=metric";
```

```
StringRequest stringRequest = new StringRequest(Request.Method.GET,url,  
    new Response.Listener<String>() {
```

```
@Override
```

```
public void onResponse(String response) {
```

```
    try {
```

```
        //find temperature
```

```
        JSONObject jsonObject = new JSONObject(response);
```

```
        JSONObject object = jsonObject.getJSONObject("main");
```

```
        double temp = object.getDouble("temp");
```

```
        temptv.setText("Temperature\n"+temp+"°C");
```

```
        //find country
```

```
        JSONObject object8 = jsonObject.getJSONObject("sys");
```

```
        String count = object8.getString("country");
```

```
        country.setText(count+" :");
```

```
        //find city
```

```
        String city = jsonObject.getString("name");
```

```
        city_nam.setText(city);
```

```
        //find icon
```

```
        JSONArray jsonArray = jsonObject.getJSONArray("weather");
```

```
        JSONObject obj = jsonArray.getJSONObject(0);
```

```
        String icon = obj.getString("icon");
```

```
Picasso.get().load("http://openweathermap.org/img/wn/"+icon+"@2x.png").into(imageView);
```

```
        //find date & time
```

```
        Calendar calendar = Calendar.getInstance();
```

```
        SimpleDateFormat std = new SimpleDateFormat("HH:mm a \nE, MMM dd yyyy");
```

```
        String date = std.format(calendar.getTime());
```

```
time.setText(date);

        //find min temperature
        JSONObject object10 = jsonObject.getJSONObject("main");
        double mintemp = object10.getDouble("temp_min");
        min_temp.setText("Min Temp\n"+mintemp+" °C");

        //find max temperature
        JSONObject object12 = jsonObject.getJSONObject("main");
        double maxtemp = object12.getDouble("temp_max");
        max_temp.setText("Max Temp\n"+maxtemp+" °C");
    } catch (JSONException e) {
        e.printStackTrace();
    }

}

}, new Response.ErrorListener() {
    @Override
    public void onErrorResponse(VolleyError error) {

        Toast.makeText(MainActivity.this,error.getLocalizedMessage(),Toast.LENGTH_SHORT).show();
    }
});

    RequestQueue requestQueue = Volley.newRequestQueue(MainActivity.this);
    requestQueue.add(stringRequest);
}

}
```


Mainactivity.xml

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView 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">
<androidx.constraintlayout.widget.ConstraintLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/backd"
    android:scrollbarSize="10dp"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/textView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="#7A090909"
        android:gravity="center"
        android:padding="15dp"
        android:text="@string/weather_app"
        android:textColor="#FFFFFF"
        android:textSize="30sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <EditText
        android:id="@+id/editTextTextPersonName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginStart="15dp"
```

```
android:layout_marginLeft="15dp"
android:layout_marginTop="30dp"
android:layout_marginEnd="15dp"
android:layout_marginRight="15dp"
android:ems="10"
android:hint="Enter City Name"
android:inputType="textPersonName"
android:paddingLeft="15dp"
android:textColor="#FFFFFF"
android:textColorHint="#FD9CFC"
android:textSize="20sp"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/textView" />
```

<Button

```
android:id="@+id/button"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginStart="15dp"
android:layout_marginLeft="15dp"
android:layout_marginTop="15dp"
android:layout_marginEnd="15dp"
android:layout_marginRight="15dp"
android:background="#95000000"
android:padding="10dp"
android:text="Search Weather"
android:textColor="#FBFBFB"
android:textSize="20dp"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/editTextTextPersonName" />
```

<androidx.constraintlayout.widget.ConstraintLayout

```
android:id="@+id/constraintLayout"
```

```
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginStart="15dp"
android:layout_marginLeft="15dp"
android:layout_marginTop="30dp"
android:layout_marginEnd="15dp"
android:layout_marginRight="15dp"
android:background="#6B0C0C0C"
android:scrollbarSize="4dp"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/button">
```

```
<TextView
```

```
    android:id="@+id/textView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_marginLeft="16dp"
    android:layout_marginTop="10dp"
    android:text="Date / Time"
    android:textColor="#FAF7F7"
    android:textSize="20sp"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/country" />
```

```
<ImageView
```

```
    android:id="@+id/imageView"
    android:layout_width="150dp"
    android:layout_height="150dp"
    android:layout_marginTop="1dp"
    android:layout_marginEnd="15dp"
    android:layout_marginRight="15dp"
    android:src="@drawable/logo"
    app:layout_constraintEnd_toEndOf="parent"
```

```
app:layout_constraintTop_toTopOf="parent" />
```

```
<TextView
```

```
    android:id="@+id/country"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="15dp"
    android:layout_marginLeft="15dp"
    android:layout_marginTop="30dp"
    android:text="Country"
    android:textColor="#FFFEFE"
    android:textSize="20sp"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

```
<TextView
```

```
    android:id="@+id/city_nam"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="15dp"
    android:layout_marginLeft="15dp"
    android:layout_marginTop="30dp"
    android:text="City"
    android:textColor="#FFFFFF"
    android:textSize="20sp"
    app:layout_constraintStart_toEndOf="@+id/country"
    app:layout_constraintTop_toTopOf="parent" />
```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

```
<androidx.constraintlayout.widget.ConstraintLayout
```

```
    android:id="@+id/constraintLayout2"
    android:layout_width="match_parent"
    android:layout_height="280dp"
    android:layout_marginStart="15dp"
```

```
android:layout_marginLeft="15dp"
android:layout_marginTop="60dp"
android:layout_marginEnd="15dp"
android:layout_marginRight="15dp"
android:background="#6B0C0C0C"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.0"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/constraintLayout">
```

```
<TextView
```

```
    android:id="@+id/min_temp"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="15dp"
    android:layout_marginLeft="15dp"
    android:layout_marginBottom="45dp"
    android:text="Min Temperature"
    android:textColor="#FFFFFF"
    android:textSize="20sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintStart_toStartOf="parent" />
```

```
<TextView
```

```
    android:id="@+id/textView3"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginStart="15dp"
    android:layout_marginLeft="15dp"
    android:layout_marginTop="20dp"
    android:layout_marginEnd="15dp"
    android:layout_marginRight="15dp"
    android:text="Temperature"
    android:textColor="#FAF9F9"
    android:textSize="50sp"
```

```
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />
```

```
<TextView
    android:id="@+id/temp_max"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginEnd="15dp"
    android:layout_marginRight="15dp"
    android:layout_marginBottom="45dp"
    android:text="Max Temperature"
    android:textColor="#FFFFFF"
    android:textSize="20sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent" />
```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

```
<androidx.constraintlayout.widget.ConstraintLayout
    android:id="@+id/constraintLayout3"
    android:layout_width="match_parent"
    android:layout_height="150dp"
    android:layout_marginStart="15dp"
    android:layout_marginLeft="15dp"
    android:layout_marginTop="60dp"
    android:layout_marginEnd="15dp"
    android:layout_marginRight="15dp"
    android:background="#6B0C0C0C"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/constraintLayout2">
```

<TextView

```
    android:id="@+id/textView3"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginStart="15dp"
    android:layout_marginLeft="15dp"
    android:layout_marginTop="20dp"
    android:layout_marginEnd="15dp"
    android:layout_marginRight="15dp"
    android:text="Mobile Application Development Project By      Xavier Dias"
    android:textColor="#FAF9F9"
    android:textSize="30sp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

</androidx.constraintlayout.widget.ConstraintLayout>

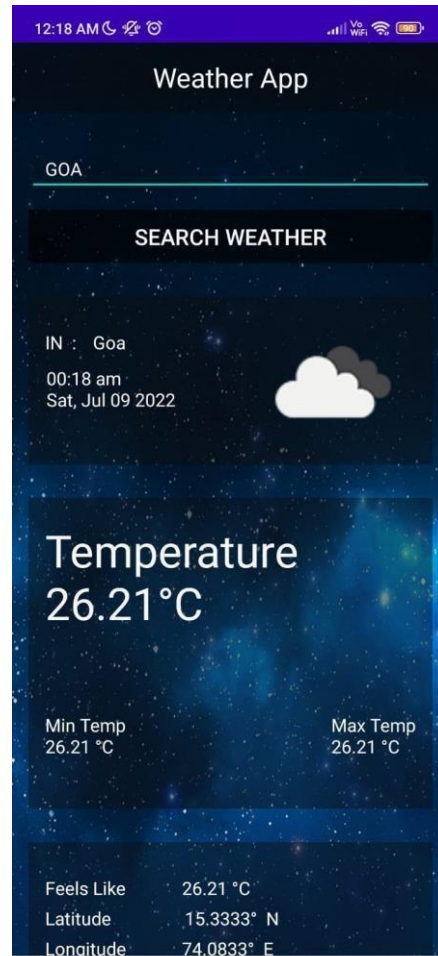
</androidx.constraintlayout.widget.ConstraintLayout>

</ScrollView>

CHAPTER 5

SNAPSHOTS

5.1 App design snapshots



5.1 App design snapshots



CHAPTER 7

CONCLUSION

Weather forecasting is a complex and challenging science that depends on the efficient interplay of weather observation, data analysis by meteorologists and computers, and rapid communication systems. Meteorologists have achieved a very respectable level of skill for shortrange weather forecasting. Further improvement is expected with denser surface and upper air observational networks, more precise numerical models of the atmosphere, larger and faster computers and more are to be realized. However, continued international cooperation is essential, for the atmosphere is a continuous fluid that knows no political boundaries. So far, the accuracy of long range forecasting has been minimal, but the short range forecasting has been of immense help and advantage to the world at large today.

REFERENCES

[1] **International Journal of Engineering Research & Technology** by : **Iseh. A. J. and Woma.** Vol. 2 Issue 12.December - 2018 (IJERT) ISSN: 2278-0181 (Weather Forecasting Models, Methods and Applications). By **T. Y.** Issue 12, December - 2020 (IJERT).

[2] **M. M. u. Hussain, B. Avci, G. Trajcevski and P. Scheuermann, "Incorporating Weather Updates for Public Transportation Users of Recommendation Systems".** Issue 2016 17th IEEE International Conference on Mobile Data Management (MDM),

[3] **K. Ramar and T. T. Mirnalinee, "A semantic web for weather forecasting systems"** Issue 2014 International Conference on Recent Trends in Information Technology,