**Android Weather Application**

# Executive Summary

This project is based on Weather Application which is an essential meteorological application that involves the capacity to forecast weather patterns and conditions for a specific location. Weather application can indicate not only the temperature of a certain place, but also the exact time of sunrise and sunset, rainfall, humidity levels, etc. This application is built using Android Studio based on Java programming language, where an API interacts with the weather report available through a weather service provider which is retrieved in JSON file format, upon which the data is parsed to make it convenient for a naive user to understand. The application serves the functionalities such as Displaying accurate Weather, GPS location-based weather display, Weather mood (sunny/ windy/ rainy), Temperature range through-out the day, Latitude & Longitude, Humidity percentage, Sunrise and sunset time, Pressure, Wind speed.

The purpose of developing this application was to fetch the current weather state of a particular location and with this application, it becomes easier to identify the temperature, this method makes it easy to produce reports for weather forecasts. There's less risk of system failure.

The program has attained a stable level, but there are still progress to be made. The system is operating at a high degree of productivity and its benefit is recognized by both the work and consumers involved with the system. In future this system can be implemented to all platforms.

**CONTENTS Page**

**No.**

#### Acknowledgement

[Executive Summary](#_TOC_250004) 4

Table of Contents 5

[List of Figures](#_TOC_250003) 6

Abbreviations 7

1. [INTRODUCTION](#_TOC_250000) 8
   1. Objective 8
   2. Motivation 8
2. PROJECT DESCRIPTION AND GOALS 9
3. LITERATURE SURVEY 10
4. TECHNICAL SPECIFICATION 12
5. **DESIGN APPROACH AND DETAILS 13**
   1. Modelling Diagrams 13
   2. System Architecture 15
   3. Sample User Interface 16
   4. Design Layouts 17
6. **PROJECT IMPLEMENTATION & DEMONSTRATION 19**
7. **RESULT & DISCUSSION 24**
8. **SUMMARY 26**
9. **REFERENCES 27**

|  |
| --- |
|  |
|  |
| . |

**List of Figures**

|  |  |  |  |
| --- | --- | --- | --- |
| **Figure No.** |  | **Title** | **Page No.** |
| 1  2  3.1  3.2  3.3  4.1  4.2  5  6 | Use Case Diagram  Activity Diagram  0 lvl DFD  1 lvl DFD  2 lvl DFD  System Architecture  UI System Architecture  Sample UI diagrams  Design Layouts |  | 13  13  14  14  14  15  15  16  18 |

## List of Abbreviations

JSON Java Script Object Notation

API Application Programing Interface

UI User Interface

GPS Global Positioning System

HTTP Hyper Text Transfer Protocol

XML Extensible Markup Language

SOAP Simple Object Access Protocol

hPa Hecto Pascal

### INTRODUCTION

* 1. OBJECTIVE

Weather forecasting is an important meteorological application, it is the ability to predict the weather patterns and the weather conditions that could be projected at a certain place. The criteria of the climatic situation are based on the temperature, wind, humidity, rainfall and data set scale.

The primary objective of this application is to display the weather condition whenever the user desires. It shall provide accurate information with respect to time. Furthermore, it should also allow the user to check the weather condition of various locations by searching and also determine the current location of the user through GPS and provide accurate weather details for the same.

The purpose of developing this application was to fetch the current weather state of a particular location and with this application, it becomes easier to identify the temperature, this method makes it easy to produce reports for weather forecasts. There's less risk of system failure.

* 1. MOTIVATION

Nowadays it has become one of the world's most technically and technologically challenging issues. With small measurements and historical knowledge, forecasting methods are threatened by dynamic weather phenomena. There are several parameters for weather conditions that are difficult to enumerate and quantify. Increasing communication systems growth has made it possible for weather prediction expert systems to incorporate and exchange resources and hybrid systems have evolved.

This Forecasting is important in the agriculture to measure the precipitation of rainfall and identify the ideal weather for a particular crop. On a daily basis, people decide their attire on the basis of forecasting. If any unusual changes in weather are determined, then warning signals are sent to protect lives.

### PROJECT DESCRIPTION AND GOALS

The weather app has been created using Android Studios, a powerful tool for building Android applications, it has all the requirements needed by an application designer. This app is capable of detecting the user’s location using CoreLocation, and provides the accurate weather report as provided by openweathermap.org.

The Application Programming Interface or API interacts with the weather report available on openweathermap.org, this can be accessed via an API key, the API key is unique to a user and cannot be used by someone else.

This API key is called by the app and is identified by the openweathermap servers, which provide the current weather data. The weather report can be accessed either for the users location as per his/her request or the user can input a particular city name and the respective city’s weather report is fetched using the API. Furthermore, this app parses the JSON files to a readable format.

The application is simple to use and user-friendly application. With not many complex features a Naïve user can also simply navigate through the application and get the expected output easily. The application by itself will obtain the users location using GPS location or the user can manually enter the desired location in the text area and search for the weather at that particular location.

The application serves the functionalities

* Display accurate Weather
* GPS location-based weather display
* Weather mood (sunny/ windy/ rainy)
* Temperature range through-out the day

Additional functionalities

* + Latitude & Longitude
  + Humidity percentage
  + Sunrise and sunset time
  + Pressure
  + Wind speed

1. **LITERATURE SURVEY**

This literature review explores different weather information sources and addresses the changing nature of weather communication and access to information. In addition, the rapid increase of smartphone and mobile device technologies is discussed, followed by perceptions of trust of weather forecasts and the inclusion of probabilistic knowledge. Finally, the introduction of conceptual frameworks relating to the adoption of technological technology provides the basis for research into how society adopts new technologies, such as Mobile Weather Applications.

**The Trend in Mobile Technology**

* Younger generations, constantly seek information and expect to have immediate results. The added value of convenience is certainly a motivating factor in what options and sources they choose (Oblinger and Oblinger, 2005). Students value convenience over many other factors and therefore turn to their smartphones and mobile devices to quickly access information (Bomhold, 2013).
* Cellular phones and mobile devices are ubiquitous in modern society, and their day-to- day functions are becoming increasingly important for cellphone owners and consumers of information. A 2011 Pew Research Center study found that 95% of the “millennial” generation (ages 18-34) and 85% of all adults own cellular phones. College students, who align mostly with the millennial generation, have the highest rate of cell phone use compared to any other generation. In terms of how cell phone owners use their phone, 76% take photos, and 72% use text messaging capabilities (Zickuhr, 2011). For mobile internet use, only 38% of all adults have accessed the internet on their mobile devices, while the number of millennials who use the internet on their phones was much higher at 63% (Zickuhr, 2011). Many Americans who use the internet on their phones are seeking weather information as well (Handmark, 2010).
* Cellular devices currently consist of smartphones, defined as “a cellphone that includes additional software functions as email or an internet browser” (smartphone, 2016). Smartphone use has soared in recent years (Zickuhr, 2011; Dahlstrom et al., 2012; Smith, 2015) with research in 2012 indicating that 62% of undergraduate college students own a smartphone, up from 55% in the previous year (Dahlstrom et al., 2012). A more recent study of smartphone use in the United States found that 64% of American adults own a smartphone, much higher than the 35% of adults reported in 2011 (Smith, 2015). Across all age spectrums, the use of smartphones for text messaging, internet access, calling, and emailing ranked highest as reasons people use their devices. Smartphones are often used for many other practical purposes too, including online banking and real estate searches (Smith, 2015).
* In terms of how people use their different mobile apps throughout the day, news applications are most popular in the morning as are weather applications, suggesting that people seek information to plan their day based on weather conditions (Böhmer et al., 2011). Overall use of mobile devices is at its lowest around 5:00AM and peaks around 6:00PM. Further, the average amount of time spent on any particular app was less than one minute for all users at all times (Böhmer et al., 2011).

**The Weather Enterprise and Smartphone Technology:**

* The private sector of the weather enterprise has taken advantage of the growing use of mobile apps. Various companies and organizations have introduced some of the most well- known weather apps used by Americans today (Nagle, 2014). In 2007, The Weather Channel became the first major weather corporation to launch its own app aimed at consumers who sought to easily retrieve forecasts in the palm of their hand (The Weather Channel, 2013). Since then, a multitude of companies has joined the mobile technology market, creating their own MWAs. Accuweather launched its app in 2008 (Accuweather, no date), and Weather Underground released its mobile app in 2011 (Weather Underground, 2011). While there are many MWAs available to consumers, each with its own unique offerings, some apps use data that is sourced from the same organization. For instance, The Weather Company, provides information for many MWAs available to consumers, including The Weather Channel MWA, Apple Weather, and some local television station MWAs.
* In addition to disseminating weather forecasts and emergency weather alerts to the public, meteorologists are capitalizing on smartphone technology in a creative way, developing an application that crowdsources data that is then ingested into weather technology. The mPING weather app allows citizen meteorologists to provide real-time surface observations of winter time precipitation events. The interface of the application was designed to facilitate the reporting of winter weather events, and the information is intended to ground-truth and verify radar data (Elmore et al., 2014).

1. **TECHNICAL SPECIFICATION**

ANDROID STUDIO:

Based on IntelliJ IDEA, Android Studio is the official Integrated Development Environment (IDE) for Android app development. Android Studio, in addition to IntelliJ's strong code editor and developer tools, includes additional capabilities that improve your efficiency when developing Android apps, such as:

* Gradle-based build system with a lot of flexibility
* An emulator with a lot of features.
* A unified development environment for all Android devices
* Fill out an application Changes to running app's code and resources without having to restart it
* To assist construct common app features and import sample code.
* Tools and frameworks for extensive testing

The weather app has been created using Android Studios, a powerful tool for building Android applications, it has all the requirements needed by an application designer. This app is capable of detecting the user’s location using CoreLocation, and provides the accurate weather report as provided by openweathermap.org.

The Application Programming Interface or API interacts with the weather report available on openweathermap.org, this can be accessed via an API key, the API key is unique to a user and cannot be used by someone else.

This API key is called by the app and is identified by the openweathermap servers, which provide the current weather data. The weather report can be accessed either for the users location as per his/her request or the user can input a particular city name and the respective city’s weather report is fetched using the API. Furthermore, this app parses the JSON files to a readable format.

1. **DESIGN DIAGRAMS-** 
   1. Modelling Diagrams

Use Case Diagram:

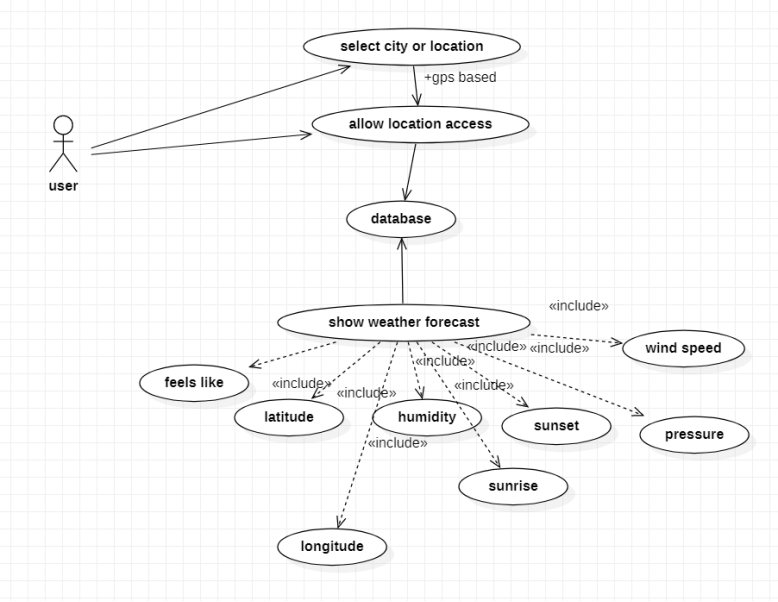


Figure 1: Use case diagram with user as an actor and various use cases

Activity Diagram:

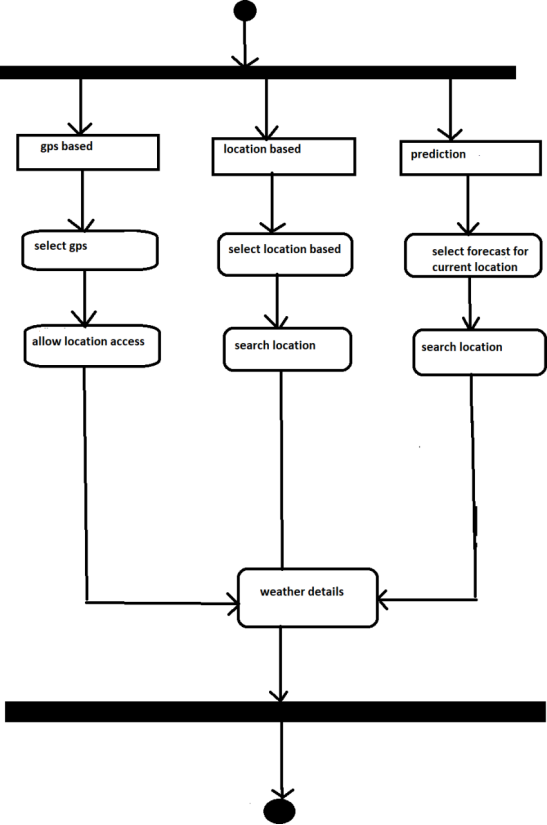


Figure 2: Activity Diagram with GPS based, location based, prediction based modes being displayed

Data Flow Diagram:

0 Level DFD



Figure 3.1: A 0 Level Data Flow Diagram

Level 1 DFD

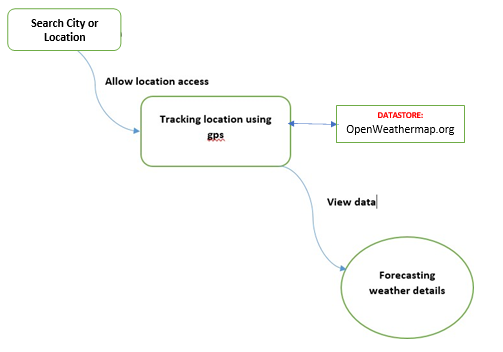


Figure 3.2: A Level 1 Data Flow Diagram

Level 2 DFD

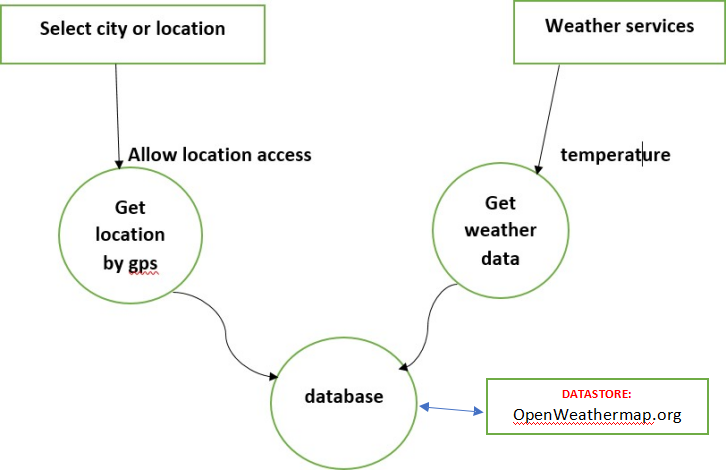


Figure 3.3: A Level 2 Data Flow Diagram

* 1. System Architecture

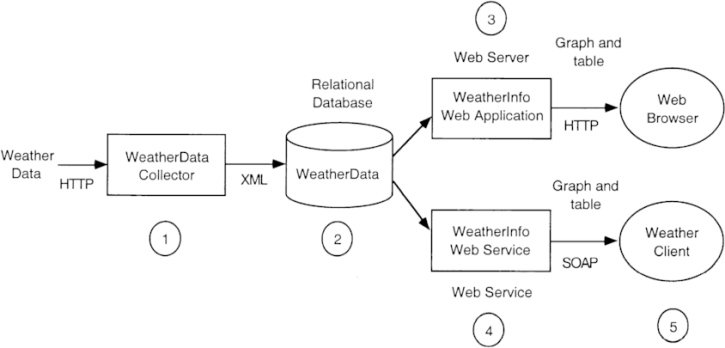
****

Figure 4.1: System architecture of Weather application where weather data is being displayed onto weather client and browser through various protocols.

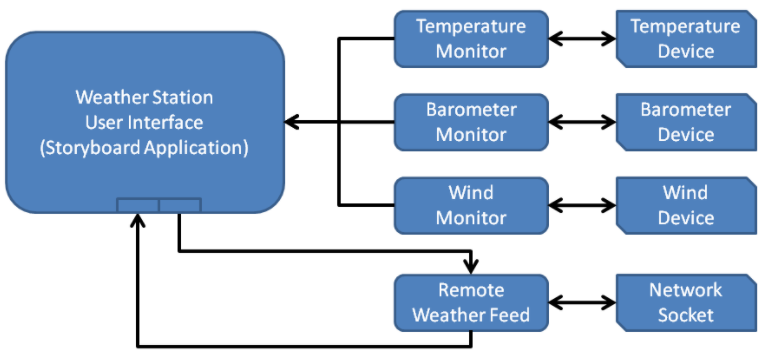
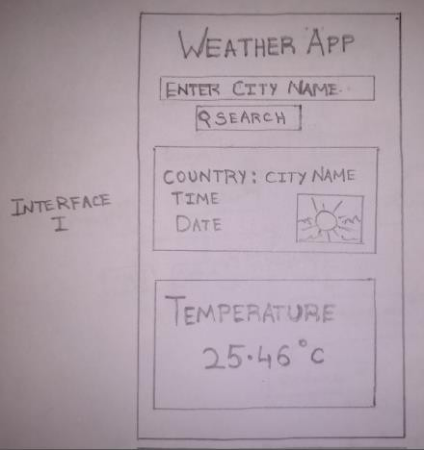
****

Figure 4.2: System architecture of Weather application User Interface where various weather parameters are obtained from the database and is being displayed.

* 1. Sample User Interface



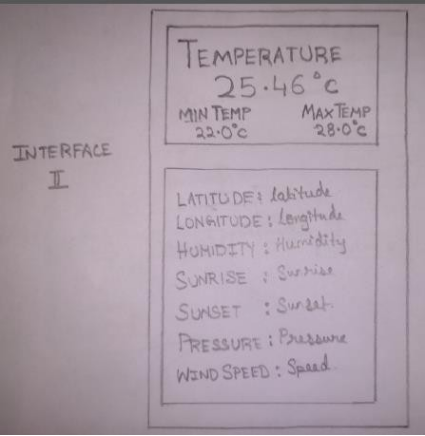
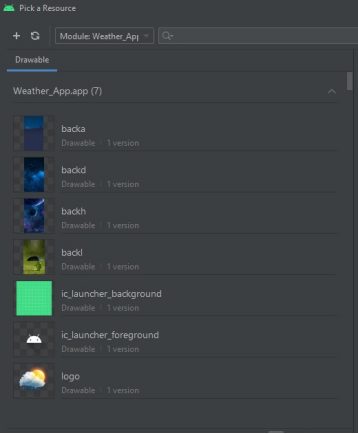
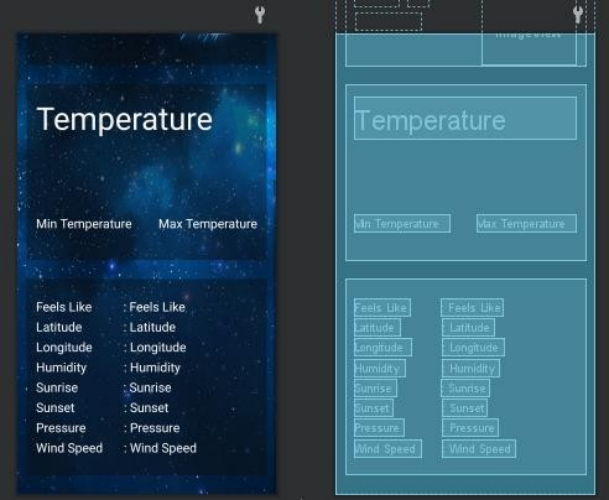


Figure 5: Expected User Interface diagram (hand drawn)

* 1. Design Layouts
* Applications front end user interface has been designed in a way that its simple and very direct to use. Without having to navigate deep into the application, the UI is designed in way that the user only has to enter the name of the location which he wants to search for.
* A simple scroll down also allows the user to navigate further into the UI allowing him to view specifics of the weather in that particular location.
* The User Interface is designed using the objects provided in the object library of Android Studio. Android Studio offers a variety of objects to choose between, as desired by the user.





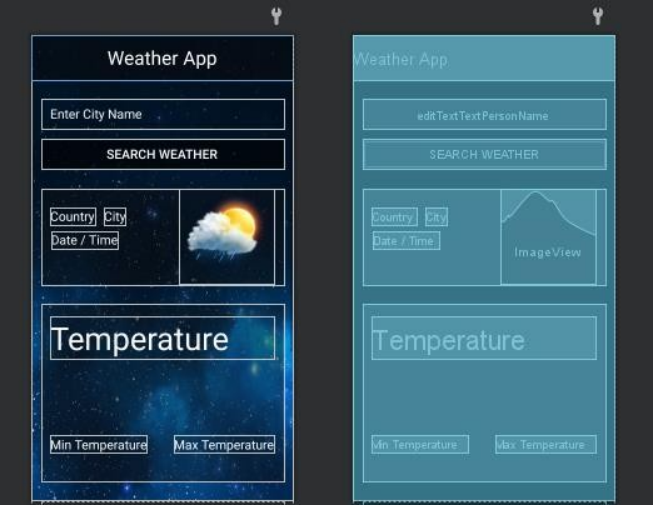


Figure 6: Various Design layouts, templates, background images used in the application

1. **PROJECT IMPLEMENTATION & DEMONSTRATION**

Application Manifest

<?xml version="1.0" encoding="utf-8"?>

<manifest xmlns:android="<http://schemas.android.com/apk/res/android>" package="com.example.weatherapp">

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

<application

android:allowBackup="true" android:usesCleartextTraffic="true" android:icon="@mipmap/ic\_launcher" android:label="@string/app\_name" android:roundIcon="@mipmap/ic\_launcher\_round" 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>

</application>

</manifest>

App Launcher

<?xml version="1.0" encoding="utf-8"?>

<adaptive-icon xmlns:android="<http://schemas.android.com/apk/res/android>">

<background android:drawable="@drawable/ic\_launcher\_background" />

<foreground android:drawable="@drawable/ic\_launcher\_foreground" />

</adaptive-icon>

Value Resources

* Styles

<resources>

<!-- Base application theme. -->

<style name="AppTheme" parent="Theme.AppCompat.Light.NoActionBar">

<!-- Customize your theme here. -->

<item name="colorPrimary">@color/colorPrimary</item>

<item name="colorPrimaryDark">@color/colorPrimaryDark</item>

<item name="colorAccent">@color/colorAccent</item>

</style> </resources>

* String

<resources>

<string name="app\_name">Weather App</string>

<string name="weather\_app">Weather App</string>

</resources>

* Colors

<?xml version="1.0" encoding="utf-8"?>

<resources>

<color name="colorPrimary">#6200EE</color>

<color name="colorPrimaryDark">#3700B3</color>

<color name="colorAccent">#03DAC5</color>

</resources>

MainActivity

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;

**AppCompatActivity**

public class MainActivity extends AppCompatActivity { EditText editText;

Button button; ImageView imageView;

TextView temptv, time, longitude, latitude, humidity, sunrise, sunset, pressure, wind, country, city\_nam, max\_temp, min\_temp, feels;

@Override

protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.*activity\_main*);

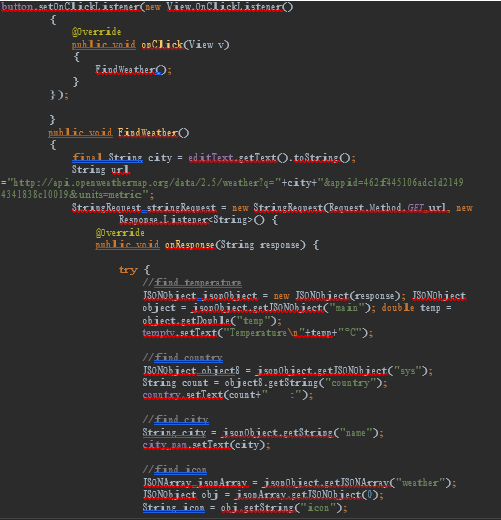
editText = findViewById(R.id.*editTextTextPersonName*); button = findViewById(R.id.*button*);

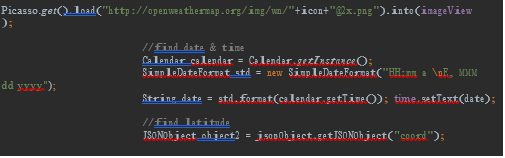
imageView = findViewById(R.id.*imageView*); temptv = findViewById(R.id.*textView3*); time = findViewById(R.id.*textView2*);

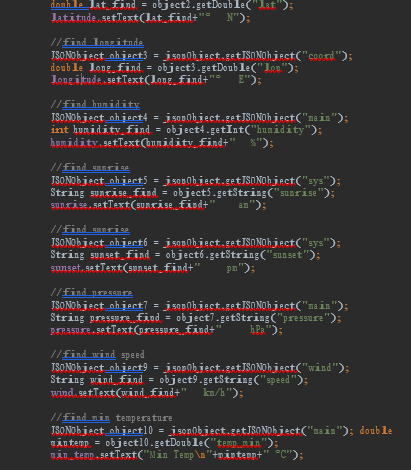
longitude = findViewById(R.id.*longitude*); latitude = findViewById(R.id.*latitude*); humidity = findViewById(R.id.*humidity*); sunrise = findViewById(R.id.*sunrise*); sunset = findViewById(R.id.*sunset*); pressure = findViewById(R.id.*pressure*); wind = findViewById(R.id.*wind*);

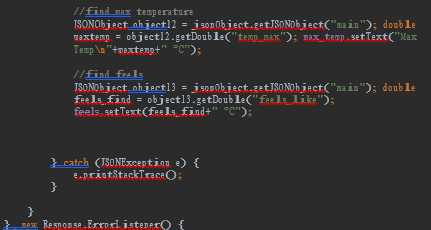
country = findViewById(R.id.*country*); city\_nam = findViewById(R.id.*city\_nam*); max\_temp = findViewById(R.id.*temp\_max*); min\_temp = findViewById(R.id.*min\_temp*); feels = findViewById(R.id.*feels*);

FindWeather



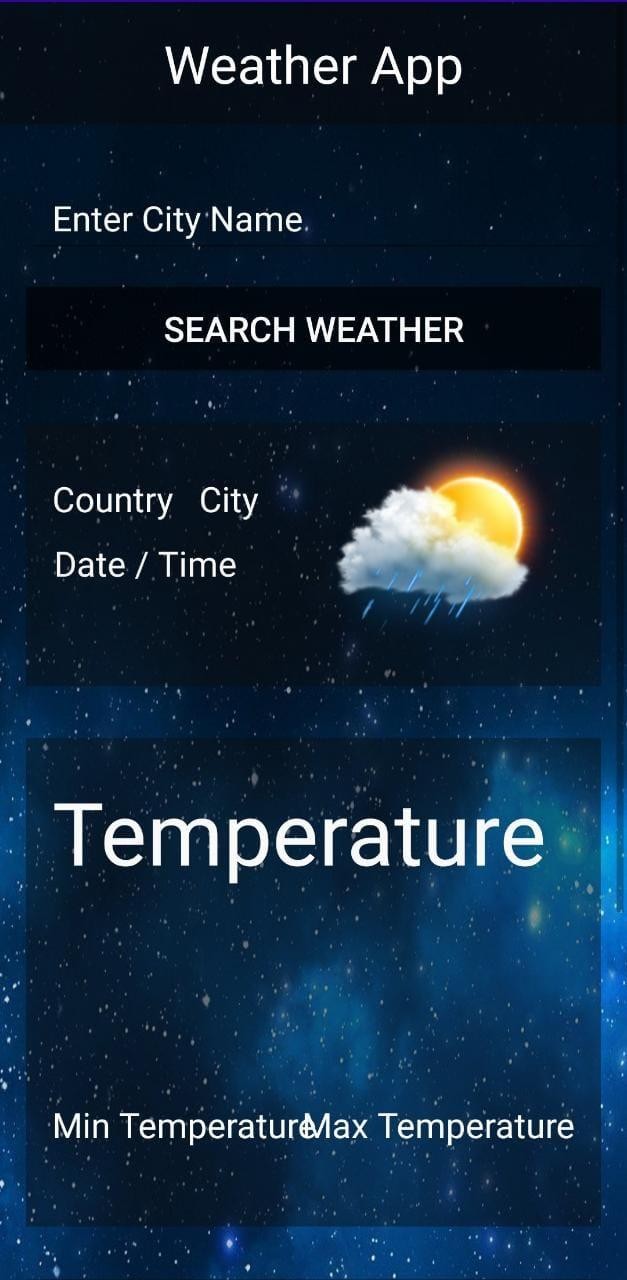








1. **RESULT & DISCUSSION**





* Applications front end user interface has been designed in a way that its simple and very direct to use. Without having to navigate deep into the application, the UI is designed in way that the user only has to enter the name of the location which he wants to search for.
* A simple scroll down also allows the user to navigate further into the UI allowing him to view specifics of the weather in that particular location.
* The User Interface is designed using the objects provided in the object library of Android Studio. Android Studio offers a variety of objects to choose between, as desired by the user.
* The purpose of developing this application was to fetch the current weather state of a particular location and with this application, it becomes easier to identify the temperature, this method makes it easy to produce reports for weather forecasts. There's less risk of system failure.
* The program has attained a stable level, but there are still progress to be made. The system is operating at a high degree of productivity and its benefit is recognized by both the work and consumers involved with the system. In future this system can be implemented to all platforms.

1. **SUMMARY**

Weather forecasting is an important meteorological application, it is the ability to predict the weather patterns and the weather conditions that could be projected at a certain place. The criteria of the climatic situation are based on the temperature, wind, humidity, rainfall and data set scales.

The purpose of developing this application was to fetch the current weather state of a particular location and with this application, it becomes easier to identify the temperature, this method makes it easy to produce reports for weather forecasts. There's less risk of system failure. The program has attained a stable level, but there are still progress to be made. The system is operating at a high degree of productivity and its benefit is recognized by both the work and consumers involved with the system. This system has can be implemented on various other platforms.

Though this application has met with the desired output, it still has some improvements to be done. It will have a result table, to show detailed prediction, and a report log to store past fetched weather details. This app will be designed in such a way that it would be capable of predicting weather report with its own algorithm and data mining techniques. It will also have widgets for easy accessibility and complete custom user-interface set-up making it fun, efficient and interactive session for users. This system will also be able to set-up real-time communication with API and provide report accurate to a second.

## References

1. openweathermap.org provides APIs with minute by minute forecasting and data on each point of globe. Accuweather. (no date).
2. Accuweather Timeline. Retrieved from <http://mediakit.accuweather.com/> timeline.
3. Diddi, A., & LaRose, R. (2006). Getting Hooked on News: Uses and Gratifications and the Formation of News Habits among College Students in an Internet Environment. Journal of Broadcasting & Electronic Media, 50(2), 193-210. doi:10.1207/s15506878jobem5002\_2.
4. Hickey, W. (2015, April 20). Where People Go To Check The Weather. Retrieved from <http://fivethirtyeight.com/datalab/weather-forecast-newsapp-habits/>
5. Joslyn, S., & Savelli, S. (2010). Communicating Forecast Uncertainty: Public Perception of Weather Forecast Uncertainty. Meteorological Applications, 17(2), 180-195. doi:10.1002/met.190