A Project Report on

Weather Forecasting Application

Submitted by

P Charitha   R170348
Y Neeraja   R170349
P Anoosha   R170356

Submitted to

IIIT RK Valley
Idupulapaya,Vempalli,YSR Kadapa
Andhra Pradesh,India PIN 516330.

Under the guidance of

Ms. M.HimaBindu
Assistant Professor

as a part of
Partial fulfillment of the degree of Bachelor of Technology in
Computer Science and Engineering.

September 2022.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
RAJIV GANDHI UNIVERSITY OF KNOWLEDGE
TECHNOLOGIES ,RK VALLEY ,
SEPTEMBER 2022.

CERTIFICATE

This is to certify that the report entitled “**Weather Forecast Application**” submitted by Y Neeraja (R170349), P Charitha (R170348), P Anoosha (R170356) in partial fulfillment of the requirements for the award of Bachelor of Technology in Computer Science and Engineering is a bonafide work carried out by them under my supervision and guidance.

The report has not been submitted previously in part or in full to this or any other University or Institution for the award of any degree or diploma.

M.HimaBindu,
Project Internal Guide,
Computer Science and Engineering,
R.K Valley, RGUKT.

P .Harinadha,
Head of the Departmet,
Computer ScienceandEngineering,
R.K.Valley, RGUKT.

Acknowledgement

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose constant guidance and encouragement crown all the efforts success.

I am extremely grateful to our respected Director,Prof. K. SANDHYA RANI for fostering an excellent academic climate in our institution.

I also express my sincere gratitude to our respected Head of the Department Mr.P HARINADHA for his encouragement, overall guidance in viewing this project a good asset and effort in bringing out this project.

I would like to convey thanks to our guide at college Miss.M HIMABINDHU for her guidance, encouragement, co-operation and kindness during the entire duration of the course and academics.

My sincere thanks to all the members who helped me directly and indirectly in the completion of project work. I express my profound gratitude to all our friends and family members for their encouragement.

Table of Contents

Abstract

The goal of this paper is to find out what are the steps in developing an application, the difficulty of it and future improvements. With this occasion, we choose to develop a weather application. A weather application is a very handy one, with daily usability and can be constantly updated to fit certain needs. Many people now rely on data directly from their smartphones and the weather forecast is something indispensable to have. The most commonly used development environment is Android Studio, the all in one developing tool which ensures full control on the application development.

The development of the user friendly application exploited using using the Android platform Android is an open-source software stack created for mobile phones and other devices and is built on top of Linux kernel and GNU software. The software stack of the Android runs Java applications using Java core libraries. Each instance of Java application runs on its own Virtual Machine called Dalvik . Android is freely available to zmanufacturers for customization, there are no fixed hardware and software configurations .However, Android itself supports features. The following listed are some of them which used in the project Uses SQLite, a lightweight relational database,for data storage.

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.

Over the past few years, Weather Report applications have become an extremely hyped thing for mobile users from all over the world. And, it is no wonder to learn that with the passage of time, their popularity and demand continue to enhance. Some of the most popular and downloaded Weather Report applications include Google Weather, Yahoo Weather App and so on.The versatile weather application is a user friendly application in which it helps the user to interact with it easily and users don’t have to face any difficulties while using this application as it was developed as a user friendly application .

Purpose

The purpose of forecasting is to prepare people and businesses from losing money and to enhance human comfort. Good forecasts pay off economically not by generating large sums of money but rather by preventing the loss of a large sum of money. Some examples of forecasts saving money include: Warning a car dealership of the likeliness of damaging hail, Warning the farmer of a potential damaging freeze.. On an individual basis, good forecasts help people plan their day more efficiently and keep them more comfortable. Tools such as having an umbrella and being properly dressed for the temperatures that day enhance human comfort. The public is most concerned with precipitation characteristics, temperature trends, very humid/dry air, sunshine hours, wind-chill/heat index, severe weather and air quality.

There are numerous products available that allow for real-time “Weather” over the internet. The purpose of this project is to implement a java based chat application that will allow users with an internet connection to engage in private and public conversations. The development of this projectcentered on the development of environment protocol that would allow the applications to properly. This is a project to create a weather application with third party server and a client to enable the clients to know the weather conditions all over the world. This project is to simulate multicast report. In a case of multicasting user can know the predictions of climate.

Scope

Weather Report application is going to be the text communication software, it will be able to communicate between two users using point to point communication. Companies would like to have communication software where in they can communicate instantly within their organization. The fact that software uses an external network setup within the organization makes it very secure from outside attacks. In the future the application can be made more effective by adding these services like Extending this application by providing graphs. provides the following facilities:

Provide Weather Information: we can know the climate behaviour all over the world with a single click by providing the name of the place.

Provide Pictorial Representation: where a user can easily figure out the condtions of the environment.

Provide Search tool: System provides a search option where a user can know the weather information all over the world with a single click.

Simple and interactive GUI: System facilitates simple and interactive Graphical User Interface for the user while handling the system. With the development of weather report and Mobile terminal, especially the release of Android smart phone platform has injected new vitality to the mobile space. Android is an open sourcing mobile operating system based on Linux which is a completely open and integrated platform for mobile devices.

Android platform consists of the operating system, middleware and user interface and application software. Bluetooth technology is a mature short-range wireless communi cation technology. The working frequency band of Bluetooth does not need a license around the globe. The advantage of Bluetooth technology is reflected in the low price, easy to control and non-visual distance limitations.

5.Android App Development

Mobile app development is a process for building mobile applications that run on mobile devices. These applications can either be pre-installed or downloaded and installed by the user later. They use the network capabilities of the device to work computing resources remotely. Hence, the mobile app development process requires creating software that can be installed on the device, and enabling backend services for data access through APIs, and testing the application on target devices.

To develop scalable mobile apps, you also need to consider screen sizes, hardware requirements, and many other aspects of the app development process. With an increasing number of jobs in the mobile app development industry, it is essential that the process is well defined and understood by entrepreneurs, startups, and especially developers.

5.1 Mobile App Development Lifecycle 2021

There are over 3.5 Billion Software Users worldwide, so there is no doubt that the industry is healthy and thriving. Stats are growing steadily, without any indications of slowing down. And studies show that an average American checks their phone at least once every twelve minutes, and over 10% of these people check their phone about every four minutes. There are some more statistics to keep in mind.

5.2 Mobile App Development Platforms

The two most important mobile app platforms are iOS from Apple Inc. and Android from Google. iOS is Apple’s proprietary mobile operating system built specifically for iPhones. Android, however, runs on mobile devices manufactured by various OEMs, including Google.

While there are many similarities between the two, however different software development kits (SDKs) are used for different platforms. Apple uses iOS exclusively for their own devices, while Google has made Android available for other companies that meet specific requirements. Developers have built over 1.5 million applications for both platforms to date.

6.Requirement Specification

Platform requirements:

Hardware/Software	Hardware/Software Component	Specification or Version
Hardware	Processer	i3
	RAM	4GB
	Hard disk	64GB
Software	Android Os	From Android 7
	Android Studio	3

7.Analysis and Design

a. Literature search and review: We will study about Android studio and the data fetching using the API that used in the weather application

b.Analysis and modeling: Based on our gathered knowledge base, we will analyze the processes and make a prototype of the application.

c.navigation and UI design: We will design application layout and application flow.

d.Implementation: From prototype we will start to integrate the modules together and we will finish all the features enlisted.The program must be proper functioning and error free.

e.Testing and debugging: Testing and debugging will be a challenging job for us, as we will let some people use the app and note suggestions from them.Also,we always need to emphasize on the comprehensiveness of our data.

f.Work on final report and documentation: We worked on the final report and software documentations .A proper functioning copy of the software(written on CD/DVD preferably) shall be provided at the end of the assigned time period.We also kept track of our limitations and enlisted some unimplemented features that we intend to work on in the near future.

7.1 Usecase Diagram

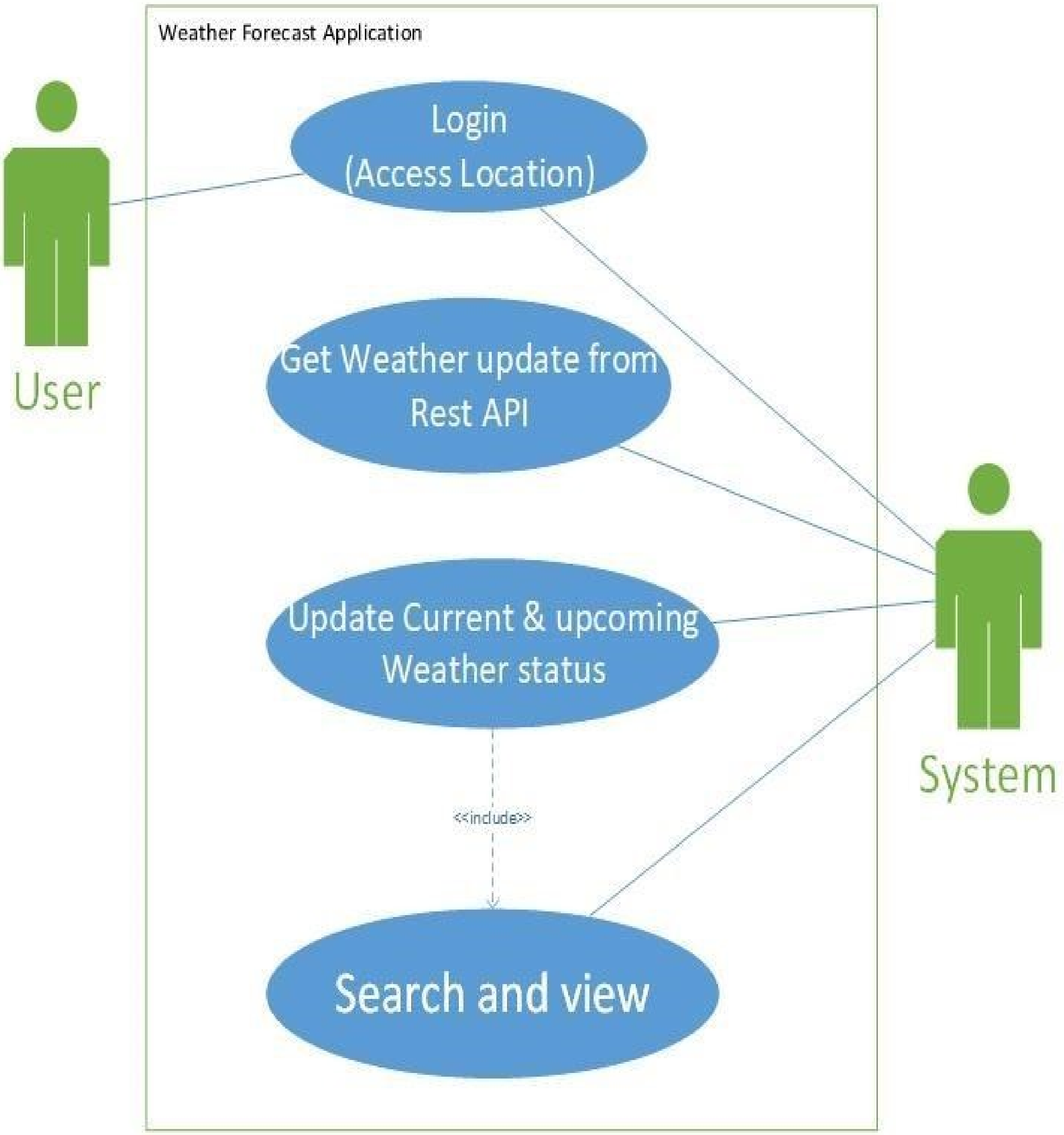
Use case diagrams model behavior within a system and helps the developers understand of what the user require. The stick man represents what’s called an actor.

Use case diagram can be useful for getting an overall view of the system and clarifying that can do and more importantly what they can’t do.

Use case diagram consists of use cases and actors and shows the interaction between the use case and actors.

- The purpose is to show the interactions between the use case and actor.
- To represent the system requirements from user’s perspective.
- An actor could be the end-user of the system or an external system.

USECASE DIAGRAM: A Use case is a description of set of sequence of actions.Graphically it is rendered as an ellipse with solid line including only its name.Use case diagram is a behavioral diagram that shows a set of use cases and actors and their relationship.It is an association between the use cases and actors. An actor represents a real-world object. Primary Actor–Sender,Secondary Actor Receiver.


Weather Forecast Application
User
Login
(Access Location)
Get Weather update from
Rest API
Update Current & upcoming
Weather status
<<include>>
Search and view
System

7.2 ER DIAGRAM

There is no standard for representing data objects in ER diagrams. Each modeling methodology uses its own notation. The original notation used by Chen is widely used in academics texts and journals but rarely seen in either CASE tools or publications by non-academics. Today, there are a number of notations used; among the more common are Bachman, crow's foot, and IDEFIX.

All notational styles represent entities as rectangular boxes and relationships as lines connecting boxes. Each style uses a special set of symbols to represent the cardinality of a connection. The notation used in this document is from Martin. The symbols used for the basic ER constructs are:

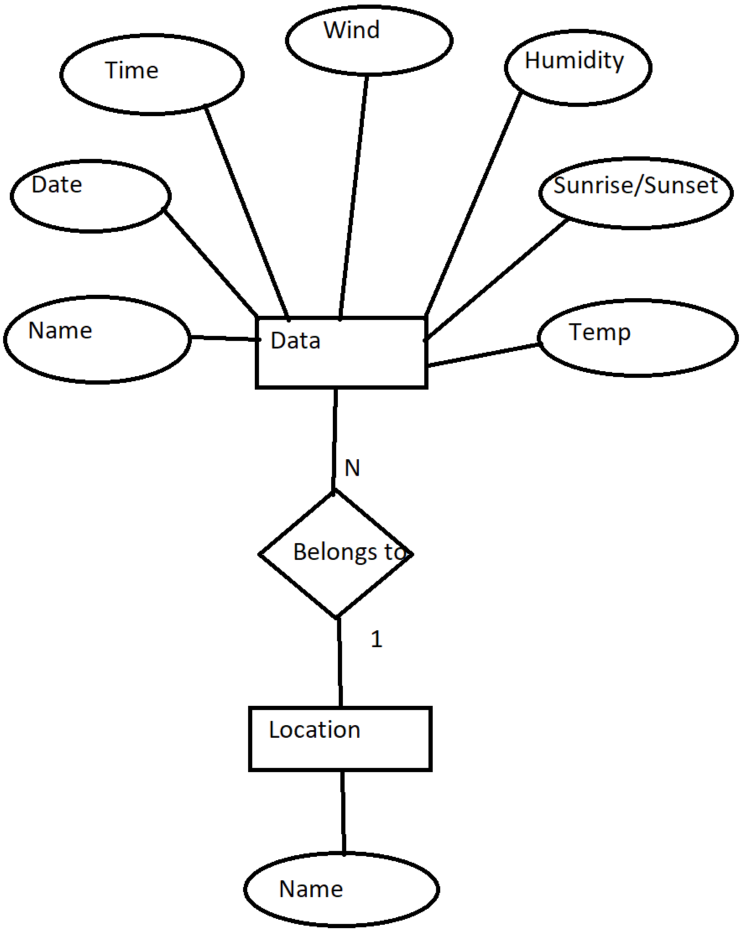
- **Entities** are represented by labeled rectangles. The label is the name of the entity. Entity names should be singular nouns.

- **Relationships** are represented by a solid line connecting two entities.The name of the relationship is written above the line. Relationship names should be verbs.

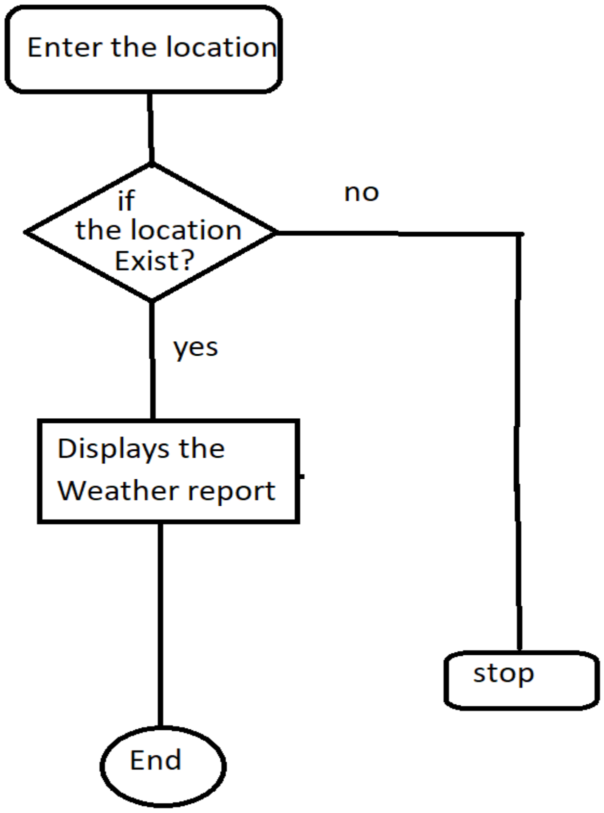
- **Attributes**,when included,are listed inside the entity rectangle.Attributes which are identifiers are underlined. Attribute names should be singular nouns.

- **Cardinality** of many is represented by a line ending in a crow's foot. If the crow's foot is omitted, the cardinality is one.

Existence is represented by placing a circle or a perpendicular bar on the line. Mandatory existence is shown by the bar (looks like a 1) next to the entity for an instance is required. Optional existence is shown by placing a circle next to the entity that is optional.


Wind
Time
Humidity
Date
Sunrise/Sunset
Name
Data
Temp
N
Belongs to
1
Location
Name

7.3 UI DIAGRAM



8.TOOLS

Android Studio:

Android is a complete set of software for mobile devices such as tablet computers, notebooks, smartphones, electronic book readers, set-top boxes etc.

It contains a **linux-based Operating System**, **middleware** and **key mobile applications**.

It can be thought of as a mobile operating system. But it is not limited to mobile only.

It is currently used in various devices such as mobiles, tablets, televisions etc.

Android Studio is a new and fully integrated development environment, which has been recently launched by Google for the Android operating system. It has been designed to provide new tools for app development and to provide an alternative to Eclipse, currently the most widely used IDE.

When you begin a new project in Android studio, the project's structure will appear with almost all the files held within the SDK directory, this switch to a Gradle based management system offers an even greater flexibility to the build process.

Android Studio allows you to see any visual changes you make to your app in real-time, and you can also see how it will look on a number of different Android devices, each with different configurations and resolutions, simultaneously.

Another feature in Android Studio are the new tools for the packing and labelling of code. These let you keep on top of your project when dealing with large amounts of code. The programme also uses a drag & drop system to move the components throughout the user interface.

Android studio offers

A robust and straight forward development environment.
An easy way to test performance on other types of device.
Wizards and templates for common elements found in all Android programming.

A full-featured editor with lots of extra tools to speed up the development of your applications.

To install Android Studio, it is necessary to have Android's Software Developer Kit (SDK), along with Java Developer Kit (JDK), included in this pack.

Open Weathermap API:

OpenWeatherMap is an online service,owned by OpenWeatherLtd,that provides global weather data via API,including current weatherdata,forecasts,nowcasts and historical weather data for any geographical location.

Using an API:

- visit http://openweathermap.org/appid and click SignUp
- Create a new account by entering the desired information,then click Create Account
- click API keys to view your existing keys,as well as to generate new ones(if needed)
- Now we have a valid API key,we can use it to create the application to display current weather information.
- To display the current weather,we use the variable q with the city in which we wish the current weather data.

9.Implementation

<u>MainActivity.java:</u>

```
package com.example.whetherapp;
import android.os.AsyncTask;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;
import androidx.appcompat.app.AppCompatActivity;
import org.json.JSONException;
import org.json.JSONObject;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Locale;
public class MainActivity extends AppCompatActivity {
        String CITY1;
        TextView addressT, updated_atT, statusT, tempT, temp_minTxt,temp_maxT,
        sunriseT,sunsetT, windT, pressureT, humidityT;
        EditText CITY;
        @Override
        protected void onCreate(Bundle savedInstanceState) {
                super.onCreate(savedInstanceState);
                setContentView(R.layout.activity_main);
                findViewById(R.id.img).setVisibility(View.GONE);
                findViewById(R.id.presd).setVisibility(View.GONE);
                findViewById(R.id.sund).setVisibility(View.GONE);
                findViewById(R.id.sunsd).setVisibility(View.GONE);
                findViewById(R.id.wnd).setVisibility(View.GONE);
                findViewById(R.id.humd).setVisibility(View.GONE);
                CITY=findViewById(R.id.city);
        }
        public void run(View view){
                CITY1 = CITY.getText().toString();
                new weatherTask().execute();
        }

        class weatherTask extends AsyncTask<String, Void, String> {
                @Override
                protected void onPreExecute() {
                        super.onPreExecute();
                        findViewById(R.id.loader).setVisibility(View.VISIBLE);
                        findViewById(R.id.mainContainer).setVisibility(View.GONE);
                        findViewById(R.id.errorText).setVisibility(View.GONE);
                        findViewById(R.id.city).setVisibility(View.GONE);
                        findViewById(R.id.button2).setVisibility(View.GONE);
                }


                protected String doInBackground(String args[]) {
                        String res=
                        GetData.excuteGet("https://api.openweathermap.org/data/2.5/weather17?
                        q="+CITY1+"&units=metric&appid=f91e3e848030d175083a637505314
                        d61");
                        return res;
                }
                @Override
                protected void onPostExecute(String result) {
                addressT = findViewById(R.id.address);
                updated_atT = findViewById(R.id.updated_at);
                statusT = findViewById(R.id.status);
                tempT = findViewById(R.id.temp);
                temp_minTxt = findViewById(R.id.temp_min);
                temp_maxT = findViewById(R.id.temp_max);
                sunriseT = findViewById(R.id.sunrise);
                sunsetT = findViewById(R.id.sunset);
                windT = findViewById(R.id.wind);
                pressureT = findViewById(R.id.pressure);
                humidityT = findViewById(R.id.humidity);
                try {
                        JSONObject jsonObj = new JSONObject(result);
                        JSONObject main = jsonObj.getJSONObject("main");
                        JSONObject sys = jsonObj.getJSONObject("sys");
                        JSONObject wind = jsonObj.getJSONObject("wind");
                        JSONObject weather =jsonObj.getJSONArray("weather").getJSONObject(0);

                        Long updatedAt = jsonObj.getLong("dt");
                        String updatedAtText = "Updated at: " + new SimpleDateFormat
                        ("dd/MM/yyyy hh:mm a",Locale.ENGLISH).format(new Date(updatedAt *
                        1000));
                        String temp = main.getString("temp") + "°C";
                        String tempMin = "Min Temp: " + main.getString("temp_min") + "°C";
                        String tempMax = "Max Temp: " + main.getString("temp_max") + "°C";
                        String pressure = main.getString("pressure");
                        String humidity = main.getString("humidity");


                        Long sunrise = sys.getLong("sunrise");
                        Long sunset = sys.getLong("sunset");
                        String windSpeed = wind.getString("speed");
                        String weatherDescription = weather.getString("description");
            String address = jsonObj.getString("name") + ", " +sys.getString("country");
                        addressT.setText(address);
                        updated_atT.setText(updatedAtText);
                        statusT.setText(weatherDescription.toUpperCase());
                        tempT.setText(temp);
                        temp_minTxt.setText(tempMin);


                        temp_maxT.setText(tempMax);
                        sunriseT.setText (new SimpleDateFormat
                        ("hh:mma",Locale.ENGLISH).format(new Date(sunrise *
                        1000)));

                        sunsetT.setText(new SimpleDateFormat("hh:mm a",
                        Locale.ENGLISH).format(new Date(sunset * 1000)));
                        windT.setText(windSpeed);
                        pressureT.setText(pressure);
                        humidityT.setText(humidity);
                        findViewById(R.id.loader).setVisibility(View.GONE);
                        findViewById(R.id.mainContainer).setVisibility(View.VISIBLE);
                        findViewById(R.id.presd).setVisibility(View.VISIBLE);
                        findViewById(R.id.sund).setVisibility(View.VISIBLE);
                        findViewById(R.id.sunsd).setVisibility(View.VISIBLE);
                        findViewById(R.id.wnd).setVisibility(View.VISIBLE);
                        findViewById(R.id.humd).setVisibility(View.VISIBLE);
                        findViewById(R.id.img).setVisibility(View.VISIBLE);
                }
                catch (JSONException e) {
                        findViewById(R.id.loader).setVisibility(View.GONE);
                        findViewById(R.id.errorText).setVisibility(View.VISIBLE);
                }
                catch (NullPointerException e1) {
                        return;
                }
            }
            }
```

GetData.java:

```
package com.example.whetherapp;

import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;

public class GetData {
    public static String excuteGet(String targetURL) {
        URL url;
        HttpURLConnection connection = null;
        try {
            url = new URL(targetURL);
            connection = (HttpURLConnection) url.openConnection();
            connection.setRequestMethod("GET");
            InputStream is;
            int status = connection.getResponseCode();
            if (status != HttpURLConnection.HTTP_OK)
```

```
                is = connection.getErrorStream();
            else
                is = connection.getInputStream();
            BufferedReader rd = new BufferedReader(new InputStreamReader(is));
            String line;
            StringBuffer response = new StringBuffer();
            while ((line = rd.readLine()) != null) {
                response.append(line);
                response.append('\r');
            }
            rd.close();
            return response.toString();
        }
        catch (Exception e) {
            e.printStackTrace();
            return null;
        }
        finally {
            if (connection != null) {
            connection.disconnect();
        }
    }
}
```

10.Testing the Developed App

After successfully developing an application, it is necessary that the quality of the application is ensured to be on-point. Quality assurance is a crucial phase in the mobile application development process as it determines the reliability, stability, and usability of the developed application. In order to ensure an all-inclusive testing process, there are a number of aspects that need to be addressed by following a complete testing cycle subjective to each application.

Testing can be broadly classified into two categories, manual testing, and automated testing. It depends on the type of application whether it needs manual testing or automated testing can give accurate results.

Any application must pass through a myriad of testing methods to come up with a perfect application. Some major testing methods that are a must-do for all the mobile applications are:

Functional Testing

- Installation & initialization of the application on all the distribution channels
- Business features and functionality testing
- Testing fields, parameters, and user feedback fields
- Testing any possible interruptions
- Testing necessary device resources
- Testing possible updates for each distribution channel

Performance Testing

- Volume testing to check the app's performance under a high volume of data
- Load testing to check the speed of the app when subjected to normal and extreme load
- Stability testing to see whether the app behaves normally under all conditions
- Testing the response time of your application in all conditions

System Testing

The goal of the system testing process was to determine all faults in our project The program was subjected to a set of test inputs and many explanations were made and based on these explanations it will be decided whether the program behaves as expected or not. Our Project went through two levels of testing

1.Unit testing

2.Integration

Unit Testing:

Unit testing is commenced when a unit has been created and effectively reviewed .In order to test a single module we need to provide a completeenvironment i.e. besides the section we would require The procedures belonging to other units that the unit under test calls Non local data structures that module accesses .A procedure to call the functions of the unit under test with appropriate parameters

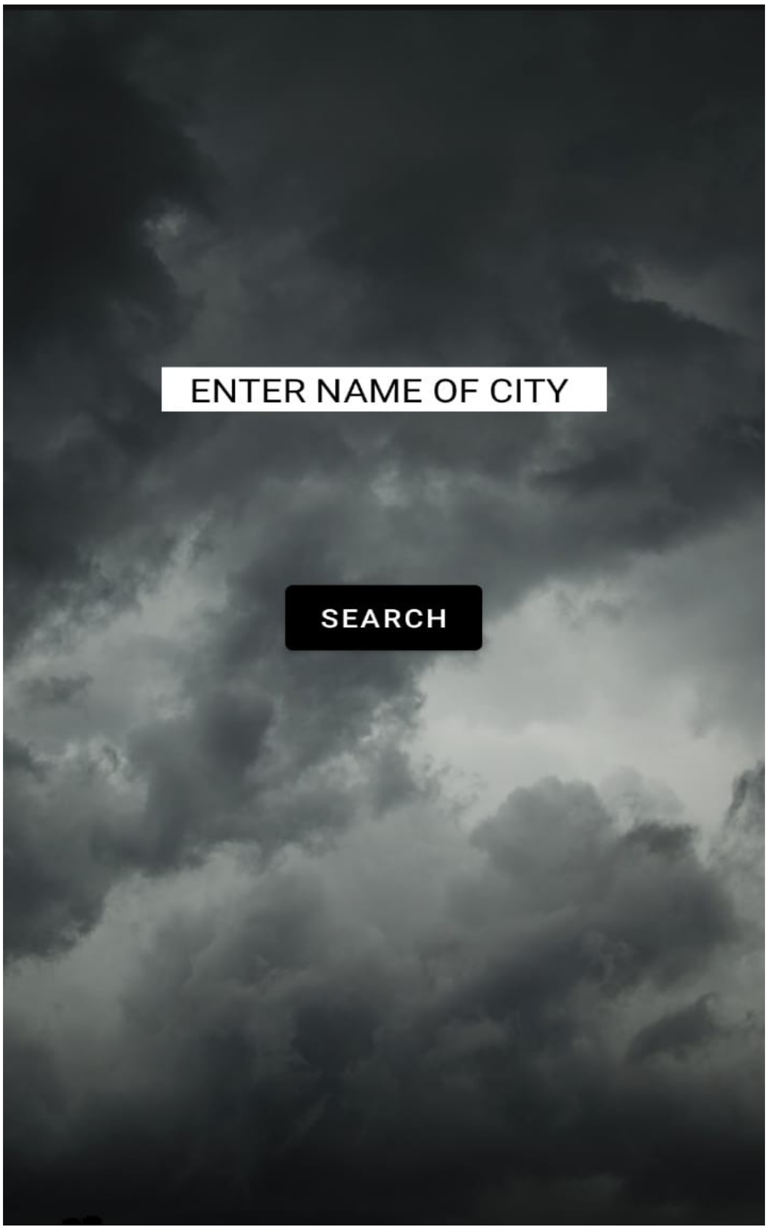
- Testing Home page-This form is used for entering the location name in the application. In this form we enter the location name if it exist data report page will open otherwise if location is wrong it will get redirected back to the home page.
- Report Generation: generate report from the main database.

Integration Testing:

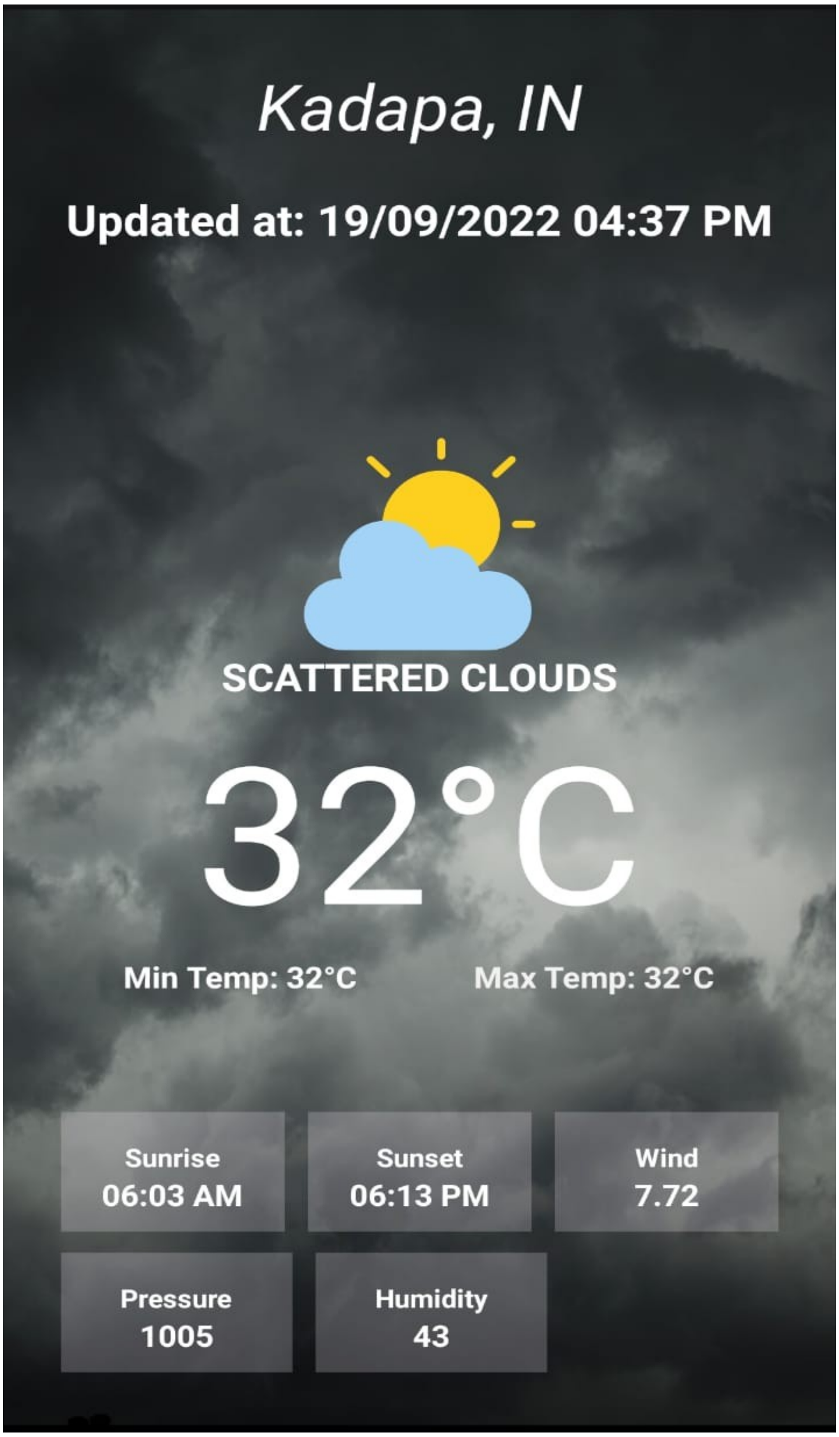
In the Integration testing we test various combination of the project module by providing the input.The primary objective is to test the module interfaces in order to confirm that no errors are occurring when one module invokes the other module.

11.OUTPUT

<u>HomePage:</u>



Weather report in the Entered place:



Kadapa, IN

Updated at: 19/09/2022 04:37 PM

SCATTERED CLOUDS

32°C

Min Temp: 32°C

Max Temp: 32°C

Sunrise
06:03 AM

Sunset
06:13 PM

Wind
7.72

Pressure
1005

Humidity
43

12.Conclusion

Finally Using this application The development of weather report happend we fetched the data of multiple elements in weather forecast.By this people can easily find about the weather in a particular place so its easy to plan everything according to them.

Weather plays a major role in our daily life, and without the meteorologist and forecaster we would have difficulty planning our daily activities. As we can see, the weather is not a simple subject like we may have been thinking. The study of weather phenomenon requires the use of science, math, and different types of equipment and technology and data. Even with all these equipment, data, and observation tools, the weather continues to be a topic to study because it is constantly changing. Meteorologist and forecasters predict the weather and its possible changes, but in reality, weather is still unpredictable.