

The image features a teal ribbon banner with white text that reads "ANDROID APPLICATION DEVELOPMENT PROJECT REPORT". The banner is positioned diagonally across the upper portion of the image. Below the banner, there is a white background area containing several colorful icons related to software development and technology, such as a magnifying glass, a lightbulb, gears, and a smartphone. The overall design is clean and professional, with a focus on the text of the banner.

# Android Application of Weather Forecast



**35°F  
Sunny**

Yangzhou, China



**24°F  
Rainy**

Dhaka, Bangladesh



# ABSTRACT

Since the 19th century, mankind has struggled to predict the weather of a particular area. Weather forecasting is a science using which people can predict the weather of an area for a particular location and time.

Over the years, scientists have discovered weather patterns using factors like humidity, winds, clouds, temperature, sea level etc. and now the weather predictions are more efficient than before due to the use of advance technologies and algorithms.

Even though there are many advance techniques and technologies, the web apps are very complicated for everyone to use. Especially for the illiterate people. Many of the farmers depend on the weather, which plays a big role in determining the future of their crops.

Many times they are not able to use technologies accordingly. Hence I decided to make an android application which would show the main details like probability of rain, temperature, humidity etc, just on clicking the app, no other steps required. The app would only require a stable internet connection and GPS.



# CONTENTS

i. Introduction	01-02
ii. Problem Statement	03-04
iii. Purpose and Scope	05
iv. Objectives	06-07
v. Requirement Analysis	08-11
vi. Use Case Diagram	12
vii. Class Diagram	13
viii. Activity Diagram	14-15
ix. System Architecture	16-22
x. Methodology	23-30
xi. System Implementation and Module Analysis	31-41
xii. Testing and Validation	42-48
xiii. Conclusion	49-51



# INTRODUCTION

Weather forecasting is the application of science and technology to predict the conditions of the atmosphere for a given location and time. Human beings have attempted to predict the weather informally for millennia and formally since the 19th century. 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.

Once a human-only endeavour based mainly upon changes in barometric pressure, current weather conditions, and sky condition or cloud cover, weather forecasting now relies on computer-based models that take many atmospheric factors into account. Human input is still required to pick the best possible forecast model to base the forecast upon, which involves pattern recognition skills, tele-connections, knowledge of model performance, and knowledge of model biases. The inaccuracy of forecasting is due to the chaotic nature of the atmosphere, the massive computational power required to solve the equations that describe the atmosphere, the error involved in measuring the initial conditions, and an incomplete understanding of atmospheric processes. Hence, forecasts become less accurate as the difference between current time and the time for which the forecast is being made (the range of the forecast) increases. The use of ensembles and model consensus help narrow the error and pick the most likely outcome.

There are a variety of end uses to weather forecasts. Weather warnings are important forecasts because they are used to protect life and property. Forecasts based on temperature and precipitation are important to agriculture, and therefore to traders within commodity markets. Temperature forecasts are used by utility companies to estimate demand over coming days. On an everyday basis, people use weather forecasts to determine what to wear on a given day. Since outdoor activities are severely curtailed by heavy rain, snow and wind chill, forecasts can be used to plan activities around these events, and to plan ahead and survive them. In 2009, the US spent \$5.1 billion on weather forecasting.

## PROBLEM STATEMENT

Weather forecasts are used by both private and government industries to plan a wide range of daily activities, protect life or properties. Some of the natural disasters which cause a lot of damage to human life on earth are high wind, flood, cyclone, smoke, fire and fog. Presently, severe advisory and alerts are determined by MOSDAC (Meteorological and Oceanographic Satellite Data Archival Centre).

The biggest problem many people face in this world is that many people are not able to use complex apps, they just need to click and know the weather at a particular time. Not all the smart phones have the features to have a google widget to show the weather. Hence this app will just require the user to click on it and it would show the current weather details to the person.

All that you would require would be a stable internet connection as well as GPS, which are found in almost all the phones now a day.

The data of the weather will be fetched by an open source website and it would be displayed on the screen of the phone in a user friendly way, so that every person could view it without many confusing steps. In the later versions of this app, we could also add some more features like multiple languages to view the weather, but for now it would be seen only in English.

It is important to first understand why people need to find the weather information and where do they turn up to for finding this information. The use and understanding of weather is interrelated and they both affect each other. Lazo et al, did a study on sources and personal interpretation of weather, where he looked at the locations for which people obtain forecasts, individual perception of the important characteristics of a forecast, and the decisions made from the gathered information.

87% times, people use weather forecasts for the city or area in which they live. 72% times, people use weather forecast to stay informed about the weather. Other times, they use weather apps to plan future events.

Respondents most often looked at forecasts for areas within close proximity to their own residence (cities in their region rather than in other states, territories and countries). Location, timing, probability, and type of precipitation along with forecast temperatures are seen as most valuable to users.

# PURPOSE AND SCOPE

## Purpose:

The purpose of developing weather app is to fetch the data in the need of taking information about weather worldwide. Another purpose for developing this software is to generate the report automatically at the end of the session or in the between of the session.

## Scope:

The scope of the project is the system on which the software is installed, i.e. the project is developed as a desktop application, and it will work for a particular institute or organization. But later on the project can be modified to operate it online. The intention of developing weather app is to fetch the data in the need of taking information about weather worldwide. Another purpose for developing this software is to generate the report automatically at the end of the session or in the between of the session or in the between of the session as they require. This project is basically a desktop application which means 3 self-contained software runs on which it has been installed under the user control.

## OBJECTIVES

There are various areas where weather forecast is used. The aviation industry is sensitive to weather. Hence, accurate weather reports are required to manage and control the air traffic. Farmers rely on weather conditions to manage their work throughout the day.

Forestry department requires information regarding wind, rain, and humidity in order to control the wild fires. Electricity department also relies on this to predict the demand. Other commercial companies also pay for weather forecasts so they can increase their profits or avoid large losses.

The main objective of this app is to show weather of a region in a user friendly manner. So that it is easy to be viewed by majority of the people. The scope of this app will not be too broad. On the contrary, it will be narrowed down to few functions which are mainly needed. The main functions involve temperature and probability of rain.

The use of weather forecast has wide importance in our lives. They are important because they can help us to be prepared in case of a weather apocalypse, and help to protect life and property.

Forecasts based on rain and temperature helps us in agriculture. Since outdoor activities are severely curtailed by heavy rain, snow and wind chill, forecasts can help us to plan activities

around these events and the plan ahead and survive them. The US spends billions every year on weather forecasting.

The purpose of this app is to fetch data in need of taking information about the weather worldwide. Another purpose is to generate report and view it on the screen of the mobile phones. The app will be developed in **Android Studio** in **Java** language, and the data will be collected from an open source website.

The project can be used by any person for checking their city or local weather. App checks the weather it gives the details of the temperature ,wind speed, condition, pressure, humidity and tells us about the sunset and sunrise time and gives the maximum and minimum temperature of the day it is updated in every four hours.

- **Simple Interface:** It has the simple interface where you can easily type your city and get the details of the temperature and other necessary details.
- **Light Application:** This is a very light app which will not take more space after you install in your mobile.
- **Easy Accessible:** It is easily accessible you can easily access your city just by typing the name of your city in which you live in it will give you the temperature and all details about the weather. All the cities are available in the world.

# **REQUIREMENT ANALYSIS**

Analysis can be defined as breaking up of any whole so as to find out their nature, function etc. It defines design as to make preliminary sketches of; to sketch a pattern or outline for plan. To plan and carry out especially by artistic arrangement or in a skilful way. System analysis and design can be characterized as a set of techniques and processes, a community of interests, a culture and an intellectual orientation.

## **Identification Of Need:**

System analysis is a process of gathering and interpreting facts, diagnosing problems and the information to recommend improvements on the system. It is a problem solving activity that requires intensive communication between the system users and system developers. System analysis or study is an important phase of any system development process. The system is studied to the minutest detail and analyzed. The system analyst plays the role of the interrogator and dwells deep into the working of the present system. The System is viewed as a whole and the input to the system are identified. The outputs from the organization are traced to the various processes. System analysis is concerned with becoming aware of the problem, identifying the relevant and Decisional variables, analysis and synthesizing the various factors and determining an optimal or at least a satisfactory solution or program of action.

A detailed study of the process must be made by various techniques like interviews, questionnaires etc. The data collected by these sources must be scrutinized to arrive to a conclusion. The conclusion is an understanding of how the system functions.

This system is called the existing system. Now the existing system is subjected to close study and problem area are identified. The designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces. The solutions are given as proposals. The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is presented to the user for an endorsement by the user. The proposal is reviewed on user request and suitable changes are made. This is loop that ends as soon as the user is satisfied with proposal.

### **Feasibility Study:**

Feasibility analysis begins once the goals are defined. It starts by generating broad possible solutions, which are possible to give an indication of what the new system should look like. This is where creativity and imagination are used. Analysts must think up new ways of doing things and generate new ideas. There is no need to go into the detailed system operation yet. The solution should provide enough information to make reasonable estimates about project cost and give users an indication of how the new system will fit into the organization. It is important not to exert considerable effort at this stage only to find out that the project is not worthwhile or that there is a need significantly change the original goal. Feasibility of a new system means ensuring that the new system, which we are going to implement, is efficient and affordable. There are various types of feasibility to be determined. They are-

- **Technical feasibility:** The technical requirement for the system is economic and it does not use any other additional Hardware and software. Technical evaluation must also assess whether the existing systems can be upgraded to use the new technology and whether the organization has the expertise to use it. Install all upgrades framework into the .Net package supported windows based application. this application depends on Microsoft office and intranet service database. Enter their attendance and generate report to excel sheet.
- **Economically Feasibility:** Development of this application is highly economically feasible. The only thing to be done is making an environment with an effective supervision. It is cost effective in the sense that has eliminated the paper work completely. The system is also time effective because the calculations are automated which are made at the end of the month or as per the user requirement.
- **Operational Feasibility:** The system working is quite easy to use and learn due to its simple but attractive interface. User requires no special training for operating the system. Technical performance include issues such as determining whether the system can provide the right information for the Department personnel student details, and whether the system can be organized so that it always delivers this information at the right place and on time using intranet services. Acceptance revolves around the current system and its personnel

## **HARDWARE REQUIREMENT FOR APP INSTALLATION**

- Processor 1.8 GHz
- Ram: 1 GB
- Free space required on phone: 2mb
- Android Smartphone

## **SOFTWARE REQUIREMENTS FOR APP DEVELOPMENT**

- Java Development kit
- Android Studio
- Android SDK Manager
- Android AVD Manager

## **HARDWARE REQUIREMENT FOR APP DEVELOPMENT**

- Standard computer with at least i3 processor
- Standard computer with 4GB of RAM
- Standard computer with 100GB of free space
- Active Internet Connectivity with good bandwidth

# USE CASE DIAGRAM

In UML, use-case diagrams model the behavior of a system and help to capture the requirements of the system. Use-case diagrams describe the high-level functions and scope of a system. These diagrams also identify the interactions between the system and its actors.

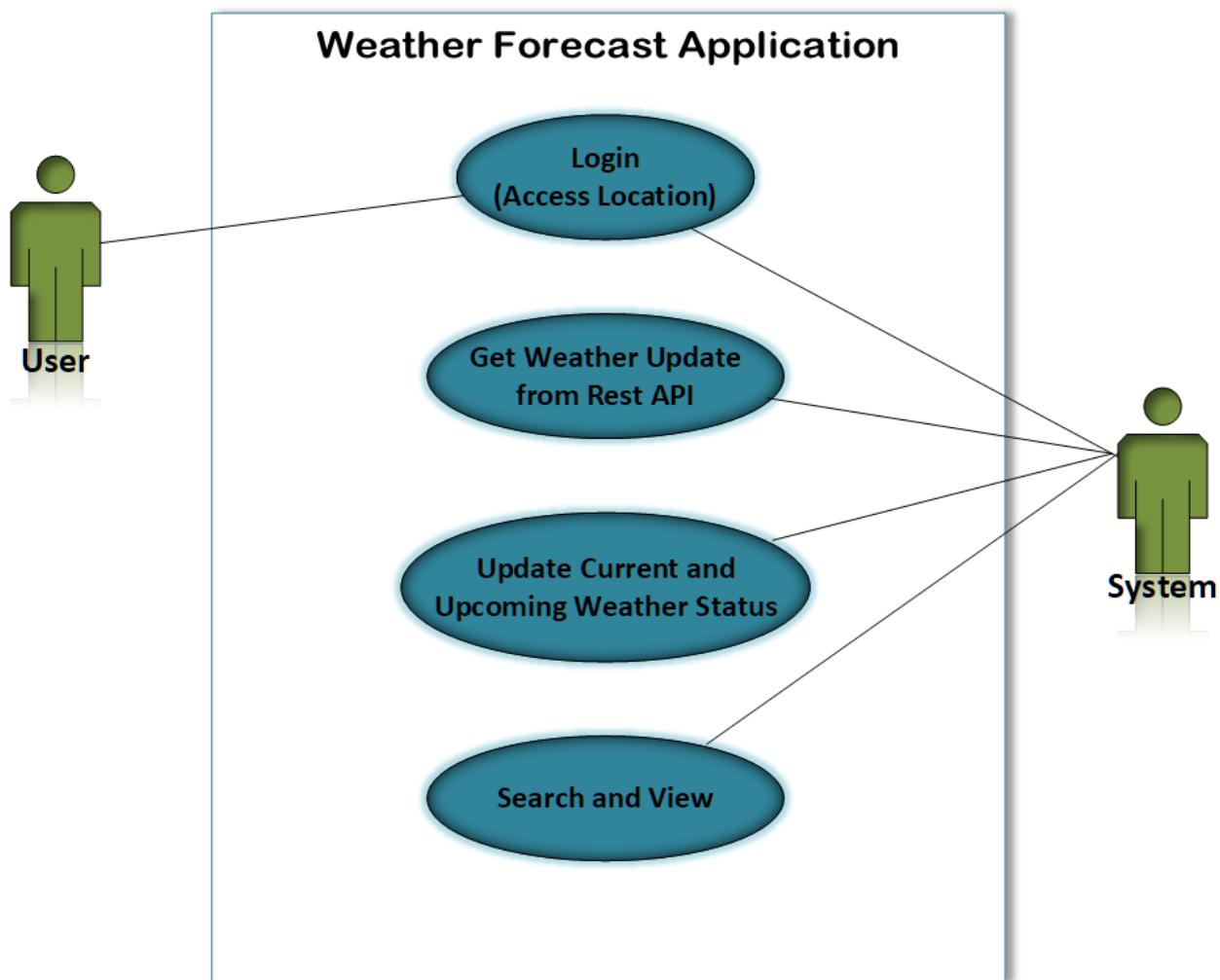


Figure: Use Case Diagram

# CLASS DIAGRAM

The basic class diagram can be seen below. The user requires having a smart phone with a data plan for accessing internet and GPS should be on because the app selects the city through it. The user then requires to install the weather app on his smart phone.

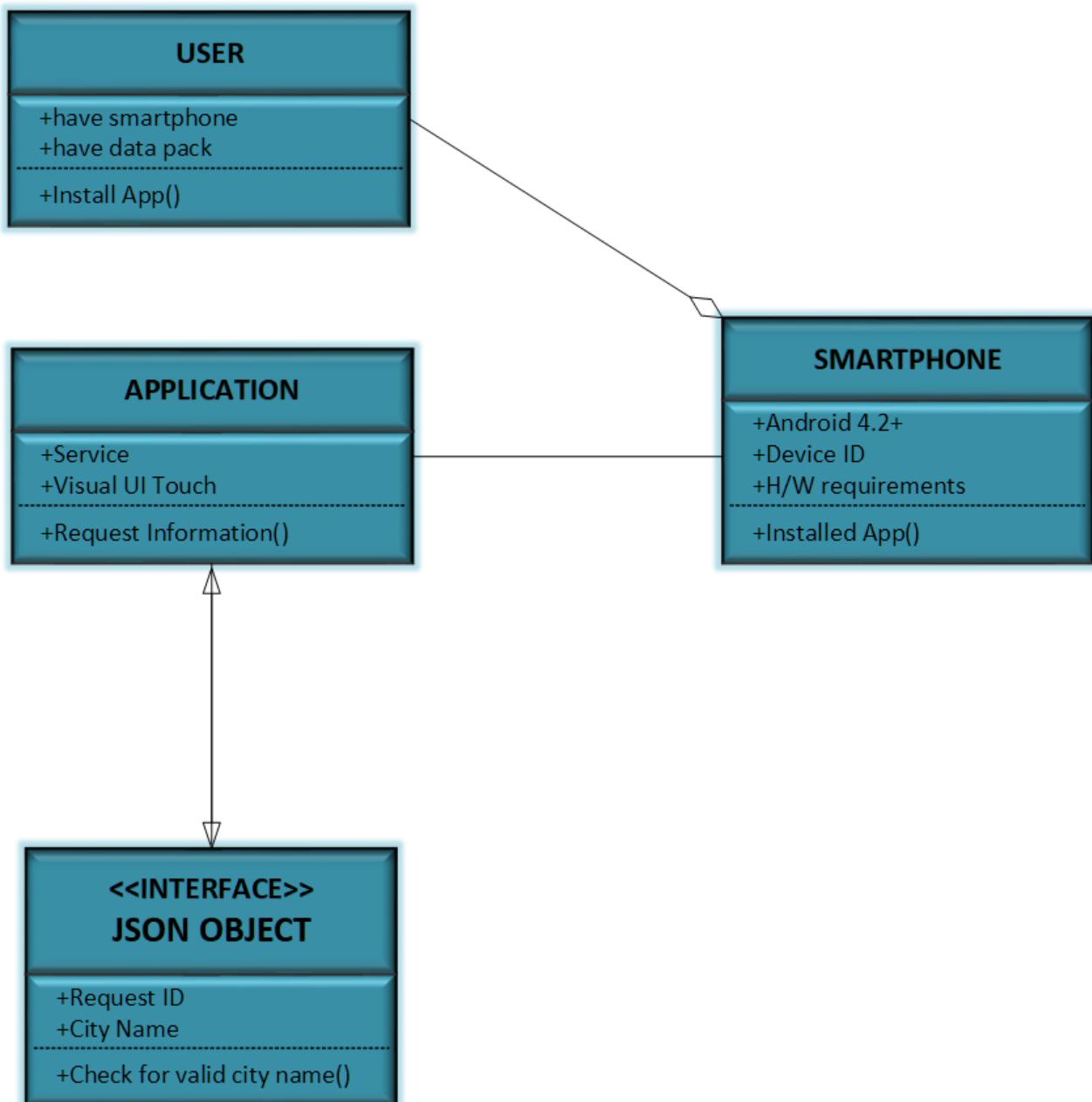


Figure: Class Diagram

# ACTIVITY DIAGRAM

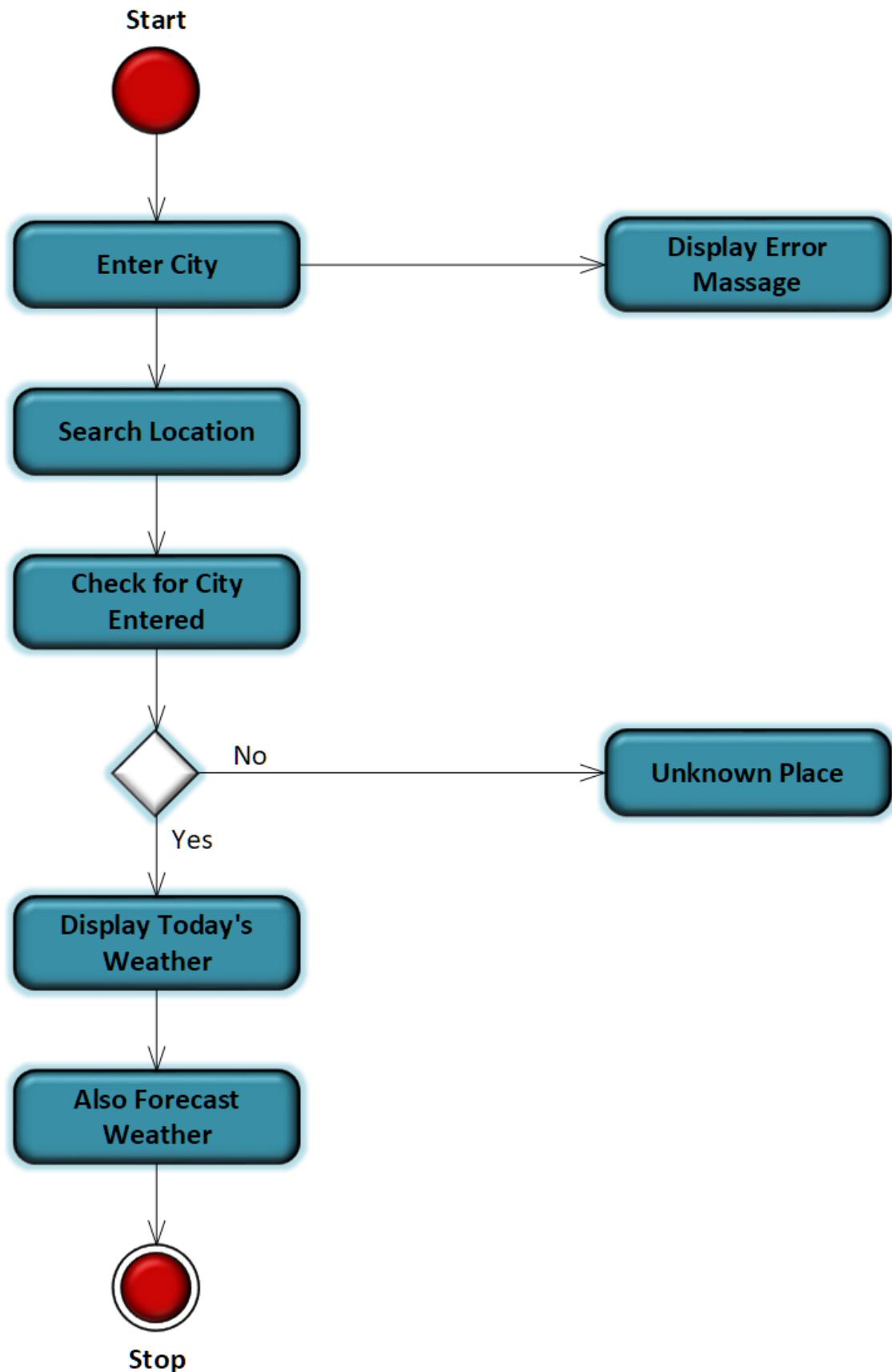


Figure: Activity Diagram

In the app, first of all, the GPS will help to select the city, where the weather has to be checked. If the location is a valid location, then we will proceed else, an error message will be displayed on the screen, saying invalid City.

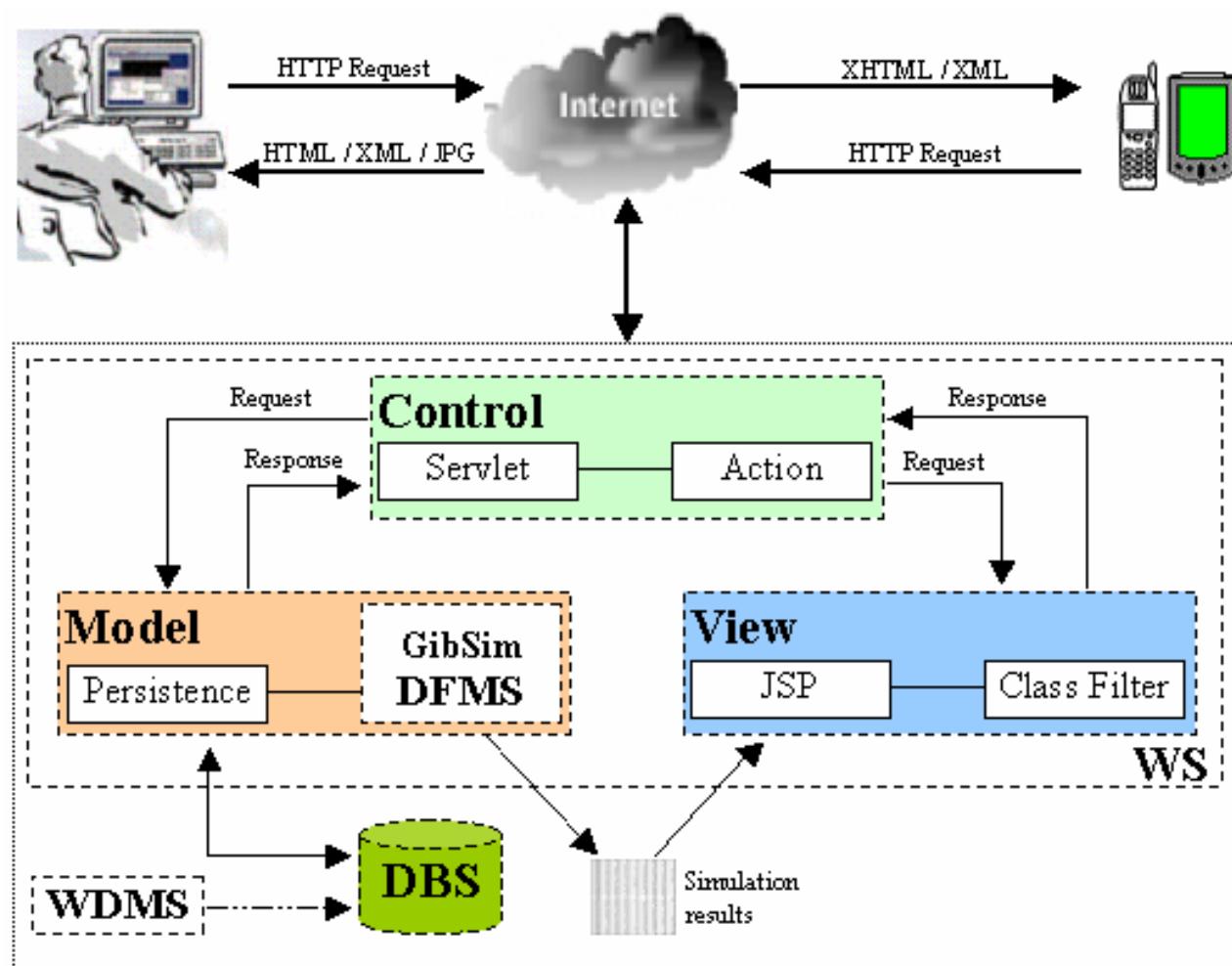
After the city is verified as a correct location the app will fetch data from the open source website and we will be able to access the weather location.

# SYSTEM ARCHITECTURE

Location-based Weather App is implemented on android platform, which is an open source operating system for mobile devices such as smart phones and tablet computers. Android Studio was developed by Open Handset Alliance, led by Google. Android Studio allows the developers, wireless operators, and handset manufacturer to make new applications and products at lower cost.

Many companies like, Samsung, HTC, and a lot of other companies are using this open source operating system for their smart phones with a very considerable price which is the cause of availability of smart-phones to almost every person. Android has its own language for developing location and internet based application. But in this project we will be using **Java language** to create a mobile weather app.

The proposed weather app is based on Service Orient Architecture (SOA) where we have thin client like Android Phones. Services are self contained and communicate using JSON and XML messages. Services are cross platform, asynchronous, reliable and secured. Once we have a web service in remote server with centralized database then we can use the same web service for different clients such as Android, iPhone, Blackberry, Windows phone, and Symbian phones. Different clients can use the common web service to save or retrieve the data.



The architecture used for developing the Android application is shown in above Figure. Components of architecture while developing a client-server application, the communication between client and database server becomes imperative. Therefore, it is important to select appropriate web service, data exchange protocol, data exchange format and Mobile positioning technologies while developing client-server application.

There are mainly four components in the architecture.

- 1. Web Services:** The Web services maintain client and service provider communications through protocols where client makes a request and service providers provide the responses.

- 2. Data Exchange Format:** Data exchange format defines the data structure for communication between the requester and the service provider to formulate requests and responses in a simple way. Examples could be XML, JSON and KML.
- 3. Data Exchange Protocols:** For establishing connection between client and server in Mobile application, it is important to decide the best way to exchange information between client and server for minimizing the limitation of mobile devices such as response time, network traffic and resource utilization. Examples could be Sockets and HTTP.
- 4. Mobile Positioning Technologies:** Location based services are the mobile services in which the user's location information is used to provide a service. The location information consists of latitude, longitude and altitude generated by any given positioning technique. Examples could be Cell-ID, GPS and Wi-Fi based techniques.

### **Flow of Architecture:**

Here, we explain the data flow of the architecture, presented in the figure. Android device fetches the input such as user's current location (latitude-longitude) using different mobile positioning technologies such as Global Positioning System, WiFi Positioning System, and Cell-id Positioning System. Android device then creates URL with run-time parameters (latitude-longitude) and calls the RESTful web service. Device uses HTTP to open the URL and awaits response. RESTful web service is responsible for querying database, retrieving results, converting them into JSON format and sending JSON response by using HTTP protocol.

Mobile device receives JSON response and parses JSON objects. For system perspective of Weather Application, we have maintained a central server which is a database containing wide range of information of weather stations and information related to weather parameters around all over World. We have also employed haversine formula for calculating the distances between user's current location and the nearest the weather stations. Therefore nearest weather station can be fetched by using Haversine formula and weather details for the particular weather station will be provided to the user.

The formula is presented in equations (1) to (3):

$$d = R \cdot c \quad \dots(1)$$

Here,  $R$  = earth's radius (mean radius = 6,371km)

$$c = 2 \cdot a \tan2(\sqrt{a}, \sqrt{1-a}) \quad \dots(2)$$

$$\Delta\text{lat} = \text{lat}_2 - \text{lat}_1 \quad \Delta\text{long} = \text{long}_2 - \text{long}_1$$

$$a = \sin^2(\Delta\varphi/2) + \cos \varphi_1 \cdot \cos \varphi_2 \cdot \sin^2(\Delta\lambda/2) \quad \dots(3)$$

Where “d” is the distance between two places considering as two points. “lat1”, “lat2” stand for latitudes, “long1”, “long2” stand for longitudes of two points and “ $\Delta\text{lat}$ ” stands for difference in latitude of two points and “ $\Delta\text{long}$ ” stands for difference in longitude of two points.

## **The forecasting process:**

Forecasting weather requires three steps:

- Observation and Analysis
- Extrapolation to find the future state of atmosphere
- Prediction of particular variables

One qualitative extrapolation technique is to assume that weather features will continue to move as they have been moving. In some cases the third step simply consists of noting the results of extrapolation, but actual prediction usually involves efforts beyond this.

The tools that people can use depend on the range of forecast that is for how much in the future we intend to forecast the weather.

Short range forecasts also known as nowcasts; extend up to 12 hours ahead. Daily range forecasts are valid for 1 to 2 days ahead. Medium range can show weather up to a week. And finally, short term climate forecasts can show weather such as one-month and three-month average forecasts.

It was earlier mentioned, as time increases the probability of correctness of prediction of weather decreases. This is because; we cannot predict changes in weather pattern in the atmosphere which are frequently changing with time.

Hence, mentioning that the results will be perfect is always a useless thing to mention. Now we will discuss the three steps for weather forecasting.

## **Algorithm Used:**

OpenWeatherMap uses the VANE Geospatial Data Science platform for collecting, processing, and distributing information about our planet through easy to use tool and APIS.

VANE platform permits processing and distributing of weather data point every second. Number of servers processes more than 10,000calls a second. This is possible due to cutting edge Big Data technologies that are being applied to the VANE platform.

It provides:

- Current weather data for more than 200,000cities
- Current weather data for any location (lat/lon)
- Daily forecast for 16 days, forecast for 5 days with 3 hour interval
- This day in history
- Weather maps (Precipitations, Clouds, Pressure, Temperature, Wind, Weather stations)
- Analytics
- Raw data from weather stations

VANE platform collects processed weather forecast data. Calculation of the global forecasts require incredible computing power. They mostly use data from NOAA GFS model and Environment Canada.

These models are global and they see their mission in not a prediction of specific weather situation at a certain point of land but in determining of the total atmospheric dynamics on the whole planet. They have a large grid size - about 50 km and longer ranged forecast 5-7 days. Then, they calculate more detailed models for specific regions. Then they combine data of multi-scaled forecasts, from the average and global to local and more accurate. That is why OWM interactive maps are very convenient. You can use global forecasts for large scale and you can get more and more detailed data just by zooming.

# METHODOLOGY

Basically, every android user will be able to access this app.

**Step 1:** Since the app will be made in Android Studio, we will first need to make a java project and add the libraries like Picasso and gson.

Gson is a Java library that can be used to convert Java Objects into their JSON representation. It can also be used to convert a JSON string to an equivalent Java object.

Picasso is one of the most popular library for android. It is very simple and powerful library for image downloading and caching.

**Step 2:** The data of the current weather report is fetched from an open source weather website. So we need to create a java class so that we could connect the application to the weather application. This is done using the website **openweathermap**. I have obtained the data of the website by fetching the link and the key of the website. Hence, to fetch the current data we require a stable internet connection.

**Step 3:** Using the key we have obtained the data which will give us information like amount of clouds, maximum temperature, minimum temperature, pressure, precipitation etc. So we need to create separate classes in a package so that we can fetch these details separately.

**Step 4:** Then we have to make a main class where we will display the City name, the last updated time, description of the weather in one word, the amount of humidity, the current time and the temperature in Celsius.

**Step 5:** In the final step we have to make changes in the display of the app that will be reflected when we run it.

## **Technologies Used**

### **Android:**

Android is a mobile operating system developed by Google, based on a modified version of the Linux kernel and other open source software and designed primarily for touchscreen mobile devices such as smartphones and tablets. In addition, Google has further developed Android TV for televisions, Android Auto for cars, and Wear OS for wrist watches, each with a specialized user interface. Variants of Android are also used on game consoles, digital cameras, PCs and other electronics.

The source code for Android is available under free and open source software licenses. Google publishes most of the code under the Apache License version 2.0 and the rest, Linux kernel changes, under the GNU General Public License version 2.

### **Why to use android?**

1. It is open-source.
2. Anyone can customize the Android Platform.
3. There are a lot of mobile applications that can be chosen by the consumer.
4. It provides many interesting features like weather details, opening screen, live RSS (Really Simple Syndication) feeds etc.

It provides support for messaging services(SMS and MMS), web browser, storage (SQLite), connectivity (GSM, CDMA, Blue Tooth, Wi-Fi etc.), media, handset layout etc.

## **Features of Android:**

- Beautiful UI: Android OS basic screen provides a beautiful and intuitive user interface.
- Connectivity: It provides the connectivity with GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE, NFC and WiMAX.
- Storage: SQLite, a lightweight relational database, is used for data storage purposes.
- Media support: H.263, H.264, MPEG-4 SP, AMR, AMR-WB, AAC, HE-AAC, AAC 5.1, MP3, MIDI, Ogg Vorbis, WAV, JPEG, PNG, GIF, and BMP.
- Open Source: Since it is open source it has a larger community support if any one gets into any problem there is a big community to support it.
- Multi-tasking: User can jump from one task to another and same time various application can run simultaneously.
- Multi-Language: supports single direction and bi-directional text.

## **Android Applications:**

Android applications are usually developed in the Java language using the Android Software Development Kit. Once developed, Android applications can be packaged easily and sold out either through a store such as Google Play, SlideME, Opera Mobile Store, Mobango, F-droid and the Amazon Appstore. Android powers hundreds of millions of mobile devices in more than 190 countries around the world. It's the largest installed base of any mobile platform and growing fast. Every day more than 1 million new Android devices are activated worldwide.

## Android Architecture:

Android operating system is a stack of software components which is roughly divided into five sections and four main layers as shown below in the architecture diagram.

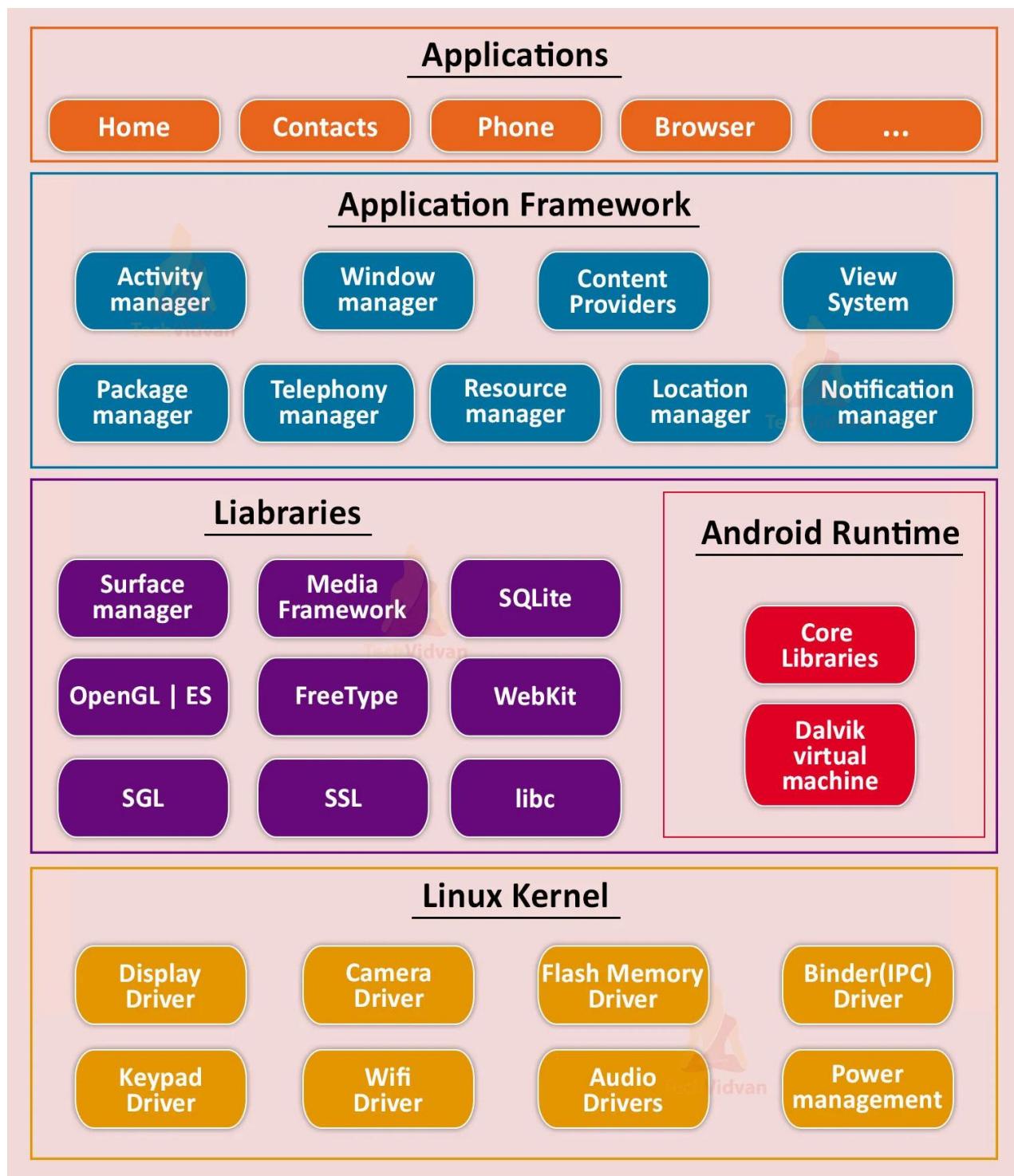


Figure: Android Architecture

Android software stack is categorized into five parts:

- **Linux kernel:** At the bottom of the layers is Linux - Linux 3.6 with approximately 115 patches. This provides a level of abstraction between the device hardware and it contains all the essential hardware drivers like camera, keypad, display etc. Also, the kernel handles all the things that Linux is really good at such as networking and a vast array of device drivers, which take the pain out of interfacing to peripheral hardware.
- **Libraries:** On top of Linux kernel there is a set of libraries including open-source Web browser engine WebKit, well known library libc, SQLite database which is a useful repository for storage and sharing of application data, libraries to play and record audio and video, SSL libraries responsible for Internet security etc.
- **Android Libraries:** This category encompasses those Java-based libraries that are specific to Android development. Examples of libraries in this category include the application framework libraries in addition to those that facilitate user interface building, graphics drawing and database access.
- **Android Runtime:** This is the third section of the architecture and available on the second layer from the bottom. This section provides a key component called Dalvik Virtual Machine which is a kind of Java Virtual Machine specially designed and optimized for Android.

The Dalvik VM makes use of Linux core features like memory management and multi-threading, which is intrinsic in the Java language. The Dalvik VM enables every Android application to run in its own process, with its own instance of the Dalvik virtual machine.

The Android runtime also provides a set of core libraries which enable Android application developers to write Android applications using standard Java programming language.

- **Application Framework:** The Application Framework layer provides many higher-level services to applications in the form of Java classes. Application developers are allowed to make use of these services in their applications.
- **Applications:** You will find all the Android application at the top layer. You will write your application to be installed on this layer only. Examples of such applications are Contacts Books, Browser, Games etc.

## **Application Components:**

Application components are the essential building blocks of an Android application. These components are loosely coupled by the application manifest file `AndroidManifest.xml` that describes each component of the application and how they interact.

- **Activities:** An activity represents a single screen with a user interface, in-short Activity performs actions on the screen.
- **Services:** A service is a component that runs in the background to perform long-running operations.
- **Broadcast Receivers:** Broadcast Receivers simply respond to broadcast messages from other applications or from the system.
- **Content Providers:** A content provider component supplies data from one application to others on request. Such requests are handled by the methods of the `ContentResolver` class. The data may be stored in the file system, the database or somewhere else entirely.

A content provider is implemented as a subclass of ContentProvider class and must implement a standard set of APIs that enable other applications to perform transactions.

### **Additional Components:**

There are additional components which will be used in the construction of above mentioned entities, their logic, and wiring between them. These components are –

1. **Fragments:** Represents a portion of user interface in an Activity.
2. **Views:** UI elements that are drawn on-screen including buttons, lists forms etc.
3. **Layouts:** View hierarchies that control screen format and appearance of the views.
4. **Intents:** Messages wiring components together.
5. **Resources:** External elements, such as strings, constants and drawable pictures.
6. **Manifest:** Configuration file for the application.

### **OpenWeatherMap API( Backend):**

OpenWeatherMap Api is here for fetching the weather of cities it is used as backend for the app to fetch the weather of different cities it is simple and clear API it is free api which is provided and the weather is updated in every four hours.

It provides the following features:

- Current conditions and forecast for 200,000+ cities and any geo location.
- Historical data
- Simple and clear API
- Interactive weather maps and satellite maps
- Raw data from 40,000+ weather stations
- 24/7 tech support via our Support Center

# SYSTEM IMPLEMENTATION AND MODULE ANALYSIS

## User Interface Design:

The design of user interfaces for machines and software, such as computers, home appliances, mobile devices and other electronic devices, with the focus on maximizing the user experience. The goal of user interface design is to make the user's interaction as simple and efficient as possible, in terms of accomplishing user goals (user-centered design).

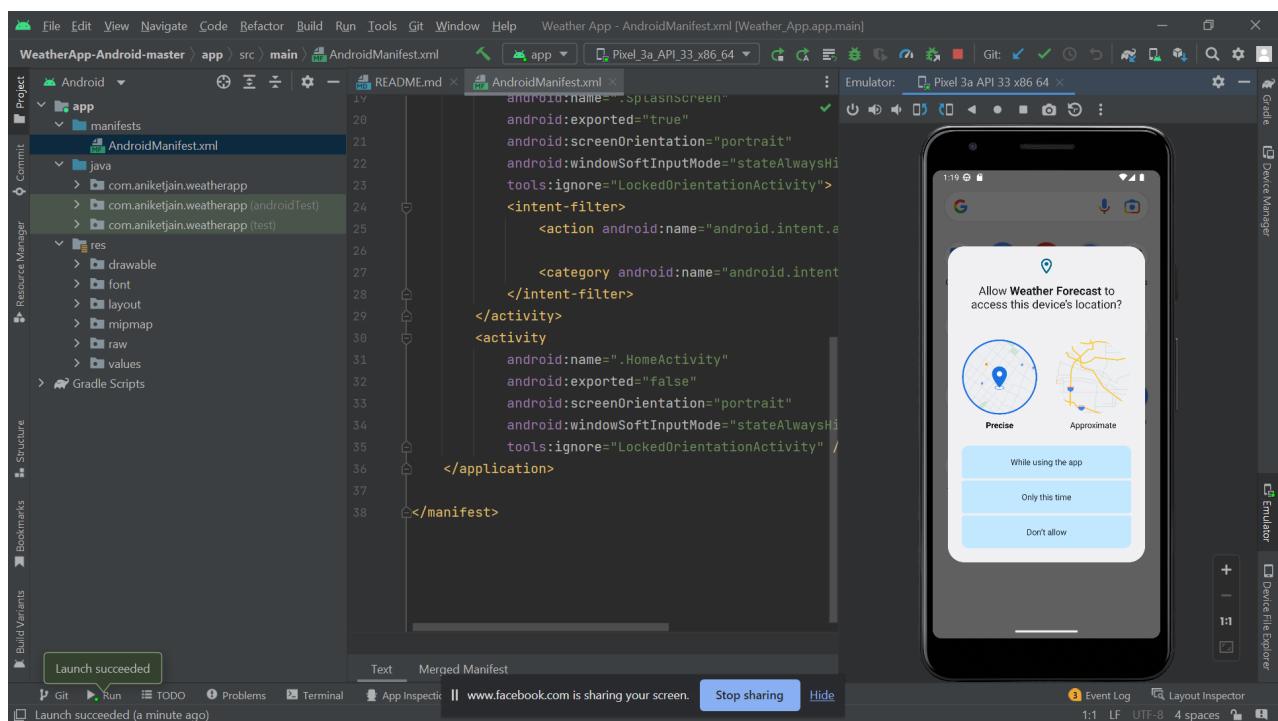
Good user interface design facilitates finishing the task at hand without drawing unnecessary attention to it. Graphic design and typography are utilized to support its usability, influencing how the user performs certain interactions and improving the aesthetic appeal of the design; design aesthetics may enhance or detract from the ability of users to use the functions of the interface. The design process must balance technical functionality and visual elements (e.g., mental model) to create a system that is not only operational but also usable and adaptable to changing user needs.

Interface design is involved in a wide range of projects from computer systems, to cars, to commercial planes; all of these projects involve much of the same basic human interactions yet also require some unique skills and knowledge. As a result, designers tend to specialize in certain types of projects and have skills centered on their expertise, whether that be software design user research, web design or industrial design.

## Implementation and Testing:

A software system test plan is a document that describes the objectives, scope, approach and focus of software testing effort. The process of preparing a test plan is a usual way to think the efforts needed to validate the acceptability of a software product. The complete document will help people outside the test group understand the "WHY" and "HOW" product validation. It should be thorough enough to be useful but not so thorough that no one outside the test group will read it.

Now, we will implement the application in Android Studio and show the App screenshot below-



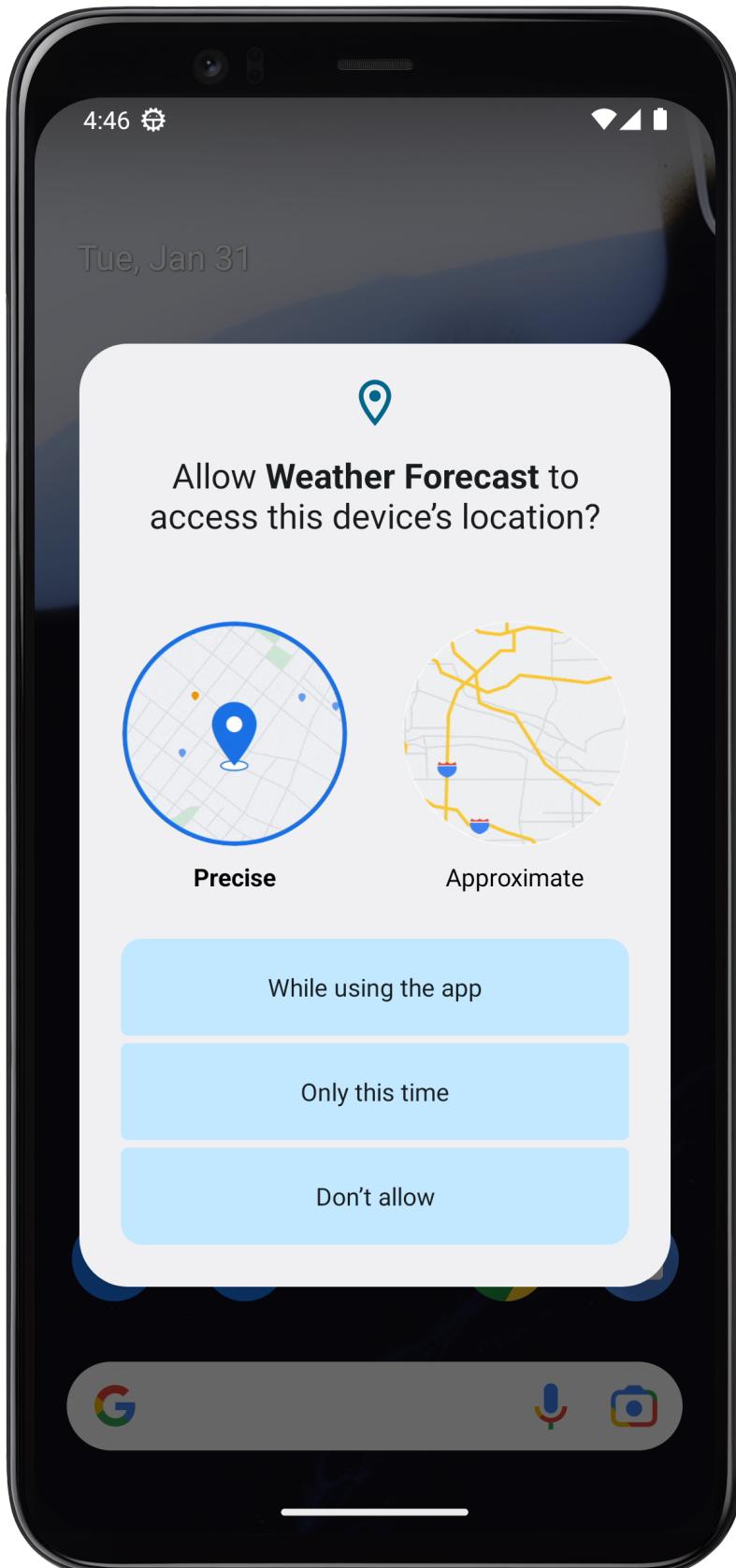
At first when we open the app it ask for access to mobile location. We need to provide access to avail the temperature of our current location automatically.

## **App Screenshot:**

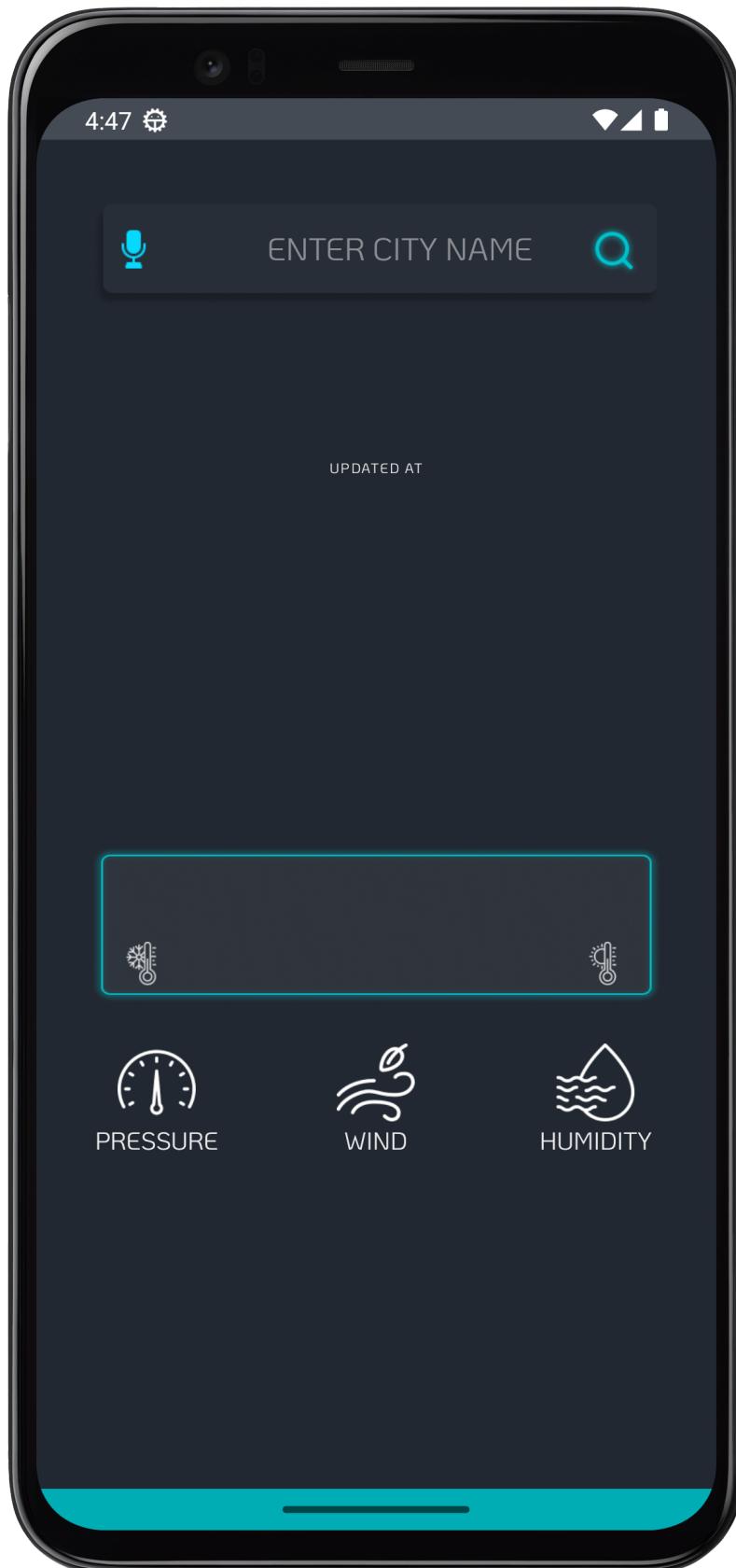
- 1) Application icon at the time of starting the **Weather Forecast** app in android phone.



## 2) Permission access page of app in android phone.



3) Home page of app in android phone.

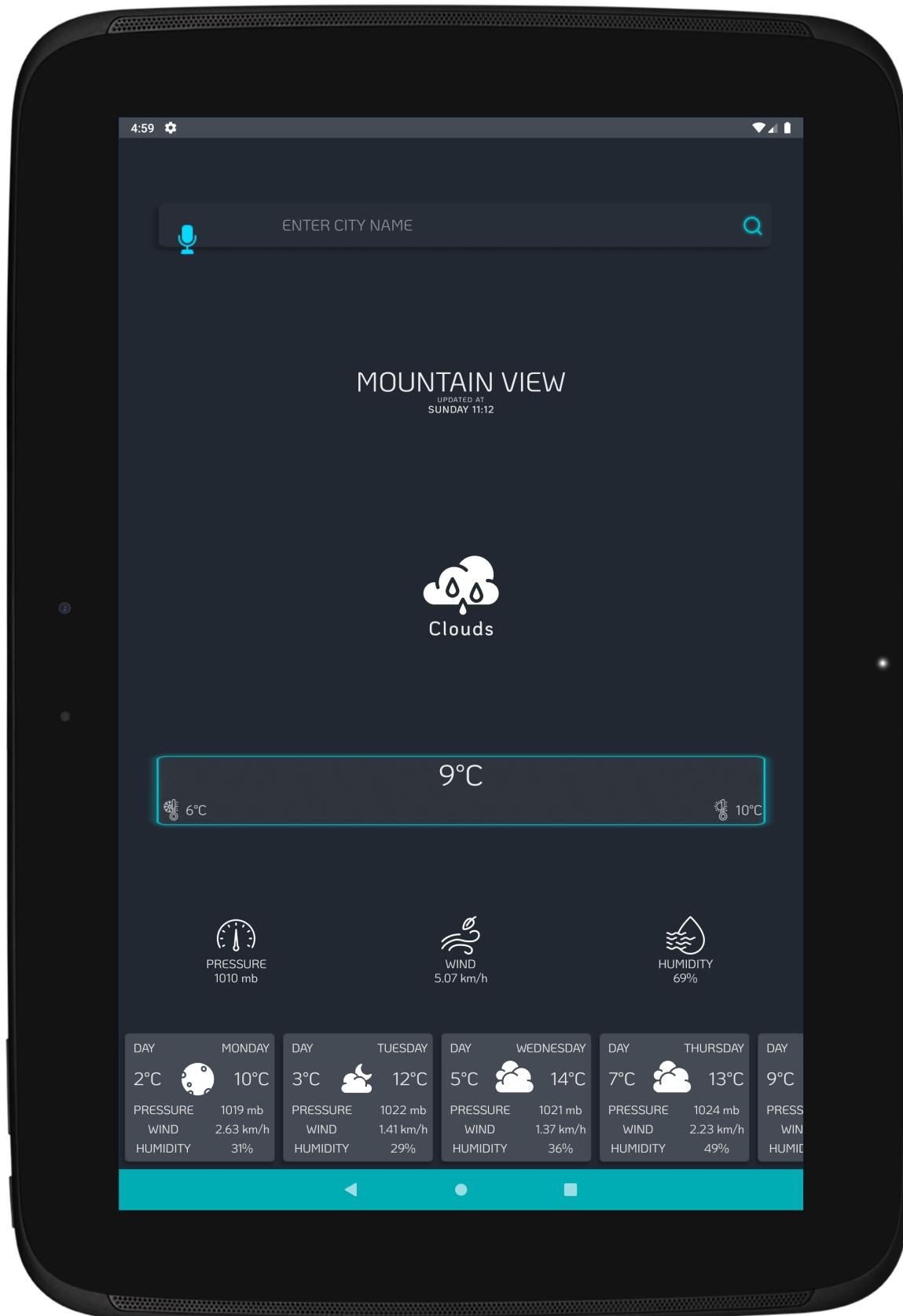


Here the data field are empty because our Emulator Device is not connected with sensor so it can not detect the current location.

#### 4) Temperature showing page of app in android phone.

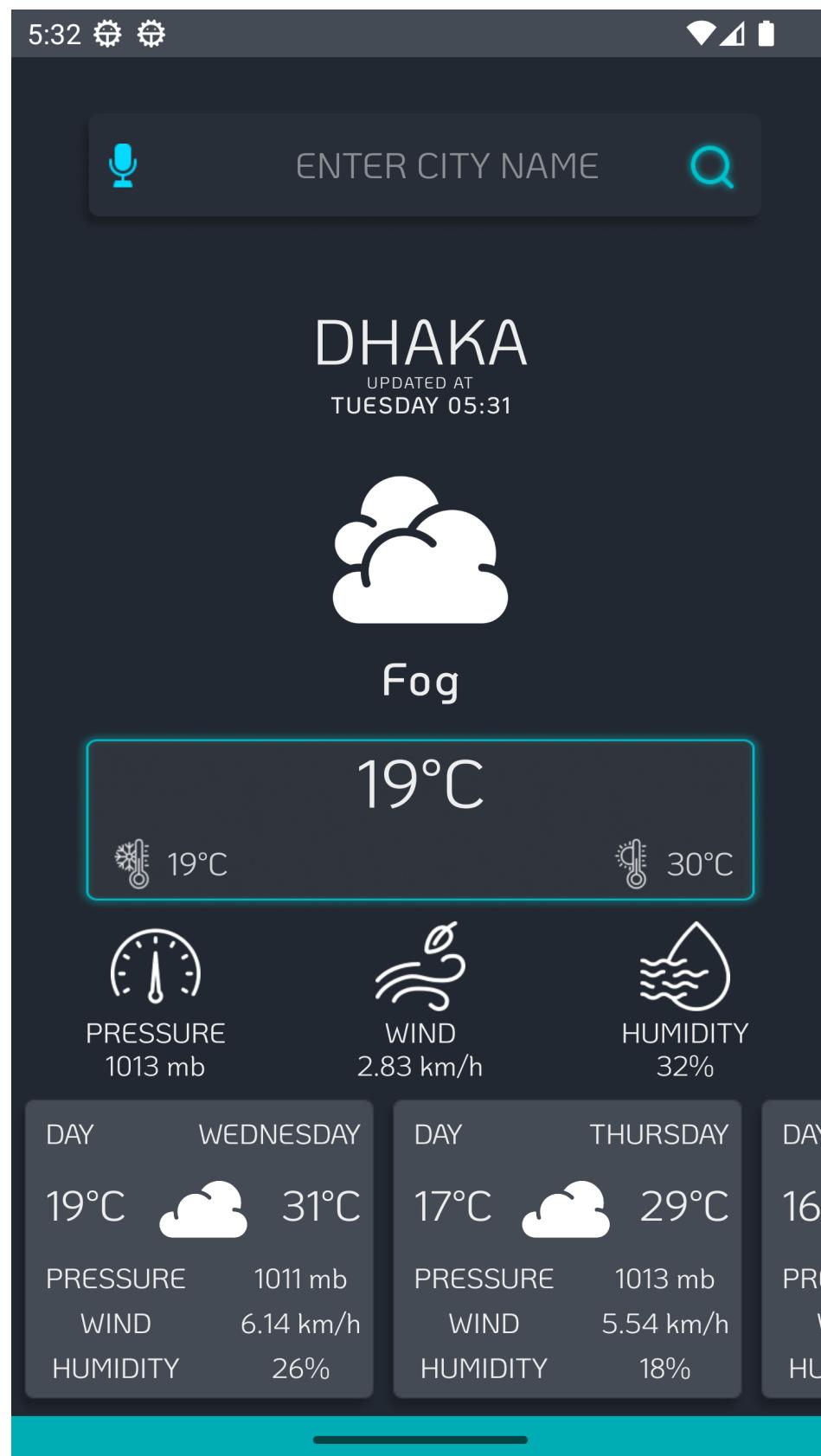


## 5) Home page of app in tab.



Here initially forecast of Mountain View is shown. Because our tab AVD is connected with sensor and this emulator's default location is Mountain View.

## 6) Tempareture page of app in android phone.



## PERFORMANCE ANALYSIS

Forecasting is more accurate than ever, but still there is wide variety in the mobile apps available in play store. So, the biggest question lies, what is the difference and which one is the best.

Foundation of modern forecasting involves collecting huge amount of data from different sources that were previously mentioned. Gaps of data are filled using a process called extrapolation. Then, using several equations, in a supercomputer, which can perform a thousand trillion calculations per second, weather, is predicted.

But the main difference comes from the main variable and the variables relying on it and the fact that different people consider different parameters as the main variable. Eg: Some might take temperature, rain or wind conditions as the main variable. Every time the prediction of weather will be slightly different from the other one. Eric Floehr, founder of ForecastWatch, a company that analyses performance of weather apps says that different forecasters perform better on different measures, longer or shorter timeframes or in a certain geographical regions.

Hence, we can understand, due to the difference in selection of the main variable, some apps are able to show one parameter of weather better than the other one. One application may be better for temperature averages and highs, probability of precipitation and wind speed; the other might be better for low temperature predictions.

To know everything about the weather, you will need to model every single particle in the atmosphere and all the interactions between them, which is not theoretically possible, because the computer doing the modelling would produce so much heat and become part of the system, and then need modelling. So no weather forecast will ever be perfect.

### **Analysis of System Developed:**

Now we will be discussing the functions of each class and package that we created in this project.

#### **Common:**

1. Created package common and a class in the package also called common.
2. This will help to hold data sharing.
3. Taken API key from an open source website, called OpenWeatherMaps, which is a London based company which shares data regarding the weather to developers.
4. API\_LINK is link API of OpenWeatherMap. And we will create a function to create a functional link to the API path.
5. Create a function to convert unit time stamp to date type with format “HH:mm”.
6. A third function will be used to get a link image from OpenWeatherMap.
7. Finally, a fourth function will be created to get Date with format “dd MMMM yyyy HH:mm”.

## **Helper:**

1. New package helper with a class package.
2. A function will be created to make a request to OpenWeatherMap's API and get return result.

## **Model:**

1. Look at the json string from OpenWeatherMap's API and create a model package for it.
2. Consider all the parameters and then create separate classes for each under the same package, also create constructors in each class with getters and setters respectively.

## **Main Activity:**

1. Implement Location Listener to get current position of our device.
2. Add runtime permission request for our app.
3. Override OnPause() and OnResume().
4. Create inner class extend Async Task.
5. Use Gson and Type to parse Json to class.
6. Set value for controls.

# **TESTING AND VALIDATION**

## **Introduction:**

Testing is the process of running a system with the intention of finding errors. Testing enhances the integrity of a system by detecting deviations in design and errors in the system. Testing aims at detecting error-prone areas. This helps in the prevention of errors in a system. Testing also adds value to the product by conforming to the user requirements.

The main purpose of testing is to detect errors and error prone areas in a system. Testing must be through well planned. A partially tested system is to detect errors and error prone areas in a system. Testing must be through well planned. A partially tested system is as bad as an untested system. And the price of an untested and under tested system is high.

## **Objectives of Testing:**

The objective our test plan is to find and report as many bugs as possible to improve the integrity of our program. Although exhaustive testing is not possible, we will exercise a broad range of tests to achieve our goal. Our user interface to utilize these functions is designed to be user-friendly and provide easy manipulation of the tree. The application will only be used as a demonstration tool, but we would like to ensure that it could be run from a variety of platforms with little impact on performance or usability.

## **Process Overview:**

The following represents the overall flow of the testing process: Identify the requirements to be tested. All test cases shall be derived using the current Program Specification. Identify which particular test(s) will be used to test each module. Review the test data and test cases to ensure that the unit has been thoroughly verified and that the test data and test cases are adequate to verify proper operation of the unit. Identify the expected results for each test.

Document the test case configuration, test data, and expected results. Document the test data, test cases, and test configuration used during the testing process. This information shall be submitted via the Unit/System Test Report (STR). Successful unit testing is required before the unit is eligible for component integration/system testing.

Unsuccessful testing requires a Bug Report Form to be generated. This document shall describe the test case, the problem encountered, its possible cause, and the sequence of events that led to the problem. It shall be used as a basis for later technical analysis.

## **Test Cases:**

Test case is an object for execution for other modules in the architecture does not represent any interaction by itself. A test case is a set of sequential steps to execute a test operating on a set of predefined inputs to produce certain expected outputs. There are two types of test cases:- manual and automated. A manual test

case is executed manually while an automated test case is executed using automation.

In system testing, test data should cover the possible values of each parameter based on the requirements. Since testing every value is impractical, a few values should be chosen from each equivalence class. An equivalence class is a set of values that should all be treated the same.

Ideally, test cases that check error conditions are written separately from the functional test cases and should have steps to verify the error messages and logs. Realistically, if functional test cases are not yet written, it is ok for testers to check for error conditions when performing normal functional test cases. It should be clear which test data, if any is expected to trigger errors.

### **Testing Steps:**

A strategy for software testing must accommodate low-level tests that are necessary to verify that a small source code segment has been correctly implemented as well as high-level tests that validate major system functions against customer requirements. A strategy must provide guidance for the practitioner and a set of milestones for the manager. Because the steps of the test strategy occur at a time when deadline pressure begins to rise, progress must be measurable and problems must surface as early as possible. Following testing techniques are well known and the same strategy is adopted during this project testing

## **Unit Testing:**

Unit testing focuses verification effort on the smallest unit of software design- the software component or module. The unit test is white-box oriented. The unit testing implemented in every module of Weather Application. By giving correct manual input to the system, the data's are stored in database and retrieved. If you want required module to access input or gets the output from the End user. any error will accrued the time will provide handler to show what type of error will accrued.

## **Integration Testing:**

Data can be lost across an interface. One module can have an adverse effect on another, sub functions, when combined, may not be linked in desired manner in major functions. Integration testing is a systematic approach for constructing the program structure, while at the same time conducting test to uncover errors associated within the interface.

## **Performance Testing:**

Performance testing is designed to test the run-time performance of software within the context of an integrated system. Performance testing occurs throughout all steps in the testing process. Even at the unit level, the performance of an individual module may be assessed as white-box tests are conducted. This project reduce attendance table, codes. It will generate report fast. No have extra time or waiting of results .entered correct data will show result few millisecond. Just used only low memory of our system. Automatically do not getting access at software. Get user permission and access to other applications.

## **Validation:**

At the culmination of the integration testing, Software is completely assembled as a package. Interfacing errors have been uncovered and corrected and a final series of software test begin in validation testing. Validation testing can be defined in many ways, but a simple definition is that the validation succeeds when the software functions in a manner that is expected by the customer. After validation test has been conducted, one of the three possible conditions exists.

- a) The function or performance characteristics confirm to specification and are accepted.
- b) A deviation from specification is uncovered and a deficiency lists is created.
- c) Proposed system under consideration has been tested by using validation test and found to be working satisfactory.

## **White Box Testing:**

In white box testing, the UI is bypassed. Inputs and outputs are tested directly at the code level and the results are compared against specifications. This form of testing ignores the function of the program under test and will focus only on its code and the structure of that code. Test case designers shall generate cases that not only cause each condition to take on all possible values at least once, but that cause each such condition to be executed at least once. To ensure this happens, we will be applying Branch Testing. Because the functionality of the program is relatively simple, this method will be feasible to apply.

## **Black Box Testing:**

Black box testing typically involves running through every possible input to verify that it results in the right outputs using the software as an end-user would. We have decided to perform Equivalence Partitioning and Boundary Value Analysis testing on our application.

## **System Testing:**

The goals of system testing are to detect faults that can only be exposed by testing the entire integrated system or some major part of it. Generally, system testing is mainly concerned with areas such as performance, security, validation, load/stress, and configuration sensitivity.

## **Output Testing:**

After performing the validation testing, the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in a specific format. The output format on the screen is found to be correct. The format was designed in the system design time according to the user needs. For the hardcopy also; the output comes as per the specified requirements by the user. Hence output testing did not result in any correction for the system.

## **User Acceptance Testing:**

User acceptance of a system is the key factor for the success of any system. The system under consideration is tested for the user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes

whenever required. This is done in regard to the following point:

- a) Input Screen Design.
- b) Output Screen Design.
- c) Format of reports and other outputs.

### **Integration Testing:**

Software testing is always used in association with verification and validation. In the testing phase of this project our aim is to find the answer to following two questions.

Whether the software matches with the specification (i.e. process base) to verify the product. Whether this software in one client what wants (i.e. product base) to validate the product.

Unit testing and integration testing has been carried out to find the answer to above questions. In unit testing each individual module was test to find any unexpected behavior if exists. Later all the module was integrated and flat file was generated.

### **Functional Testing:**

These are the points concerned during the stress test:

Nominal input: character is in putted in the place of digits and the system has to flash the message "Data error".

Boundary value analysis: exhaustive test cases have designed to create an output report that produces the maximum (and minimum) allowable number of table entries.

# CONCLUSION

Climate forecasting is the utilization of science and innovation to foresee the states of the environment for a given area and time. Individuals have endeavoured to anticipate the climate casually for centuries and officially since the nineteenth century. Climate forecasts are made by gathering quantitative information about the current condition of the environment at a given spot and utilizing meteorology to extend how the air will change.

There are various areas where weather forecast is used. The aviation industry is sensitive to weather, hence, accurate weather reports are required to manage and control the air traffic. Farmers rely on weather conditions to manage their work throughout the day. Forestry department requires information regarding wind, rain and humidity in order to control the wild fires. Electricity department also relies on this to predict the demand. Other commercial companies also pay for weather forecasts so they can increase their profits or avoid large losses.

By this system weather forecasting report generation becomes easy. Less chances of malfunctioning are there. The system has reached a steady state but still improvements are to be made. The system is operated at a high level of efficiency and all the work and user associated with the system understand its advantage. It was intended to solve as requirement specification. In future this system can be implemented to all over the world and will be designed for cross platform.

## **Limitations:**

- a)Unpaid APIs provide incomplete services. Many details cannot be fetched
- b)Often tuples of upcoming days remain empty once again due to free APIs
- c)The GMS API ( Google Manual Search) is actually keyword based that might only provide data of few discrete locations. The data might not be precise and continuous.
- d)Language diversity could have been implemented. Multilingual apps make it easy for users worldwide.

## **Unimplemented features:**

- a)We intend to provide more detailed tips based on the age, gender, region and health conditions(Dust allergy, heatstroke tendency etc.) of the user.
- b)FAQ section based on detailed data might be quite handy. It will save the user's efforts and make it more convenient.
- c)Using paid APIs to fetch more details might make the application more comprehensive and appealing. For example wind speed, precipitation & sea level values can be used to provide more intellectual tips.
- d)Mobility of the traveler, their start & end points of the journey, their route of traveling(road, waterway, aerial) shall be strongly focused. The application needs to be an aid for traveling users.
- e)Recording User inputs to understand the user preferences and providing them necessary notifications needs to be kept in mind.
- f)Maybe someday in the near future, we will use AIs for more precision and accuracy.

## **Bibliography:**

1. <https://www.javatpoint.com/>
2. <https://www.tutorialspoint.com/>
3. <https://www.github.com/>
4. <https://openweathermap.org/technology>
5. [https://en.wikipedia.org/wiki/Android\\_\(operating\\_system\)](https://en.wikipedia.org/wiki/Android_(operating_system))



Student Name: MEJBA RAOHA BIN (李一含)

Student ID: 208801135

Department: Software Engineering

Teacher Name: ZHU Xinfeng, Ph.D.

Yangzhou University

Date: 30th January 2023