Software Requirement Specification

For

Real Time Air Quality Monitoring & Weather Forecasting System

Project Mentor:
Ms. Shalini Singhal
Assistant Professor (IT)

Submitted By:
Rishabh Dev Singh (19ESKIT077)
Sakshi Agarwal (19ESKIT083)
Vinit Kumar Shah (19ESKIT098)
Yashwardhan Gaur (19ESKIT099)

Submitted To:
Ms. Sanju Choudhary
Associate Professor (IT)

Department of Information Technology
Swami Keshvanand Institute of Technology, Management and Gramothan, Jaipur

Table of Contents

1. Introduction

1.1 Purpose

The main purpose of this document is to layout the requirement of this system. The document will describe how the application will collect and display local area weather and analyze weather forecast and also display the air quality index of the currently selected location. This document serves as a layout for the user interface, software and hardware dependencies. The SRS will serve as a reference for the development team for design, implementation and verification phases; the SRS is also an agreement between the client and the development teams regarding the functionality of the finished product.

1.2 Document Conventions

Bold-faced text has been used to emphasize section and sub-section headings. Highlighting is to point out words in the glossary and italicized text is used to label and recognize diagrams.

1.3 Intended Audience and Reading Suggestions

This project is a prototype for the Real Time Air Quality Monitoring & Weather Forecasting System and it is restricted within the college premises. This has been implemented under the guidance of college professors. This is for all those who will participate in describing, making, developing and working of the application, such as stakeholders, developers, and end users, those who might be interested in advertising and marketing of the product. The supervising team who will lead the project, and those who will validate it. It can be used in daily routine by all kind of users, either they be students or office workers, shop owners or successful business man, any person with plans and routines that might be effected by sudden change in forecast.

1.4 Project Scope

Application will collect weather data, analyze forecast readings, display output and help the users with their day, per conditions. It will display wind speed, humidity, temperature, day type and how the rest of the day will be like. System shows hourly and daily weather change, calculate weekly output and help the user with his wear according to the weather; give look ahead for the upcoming forecast change, and allergies that users might get. System also streams live map in iteration with weather change. User can send weather report to various social media platforms on their accounts, also send images and reports as personal messages. The application can automatically detect user's location, while user can also manually set or find locations to detect its weather, and get live updates through notifications. The product also provides the functionality to set weather units according to the user's preference, also allows the user to set severe weather alerts. It will cover global news of natural disasters and phenomenon. Hence, it helps the people relax, as they can easily know how the rest of the day's weather would be, and can set plans accordingly.

1.5 References

- IEEE 830-1998 standard for writing SRS document.
- IEEE Std 1233, 1998 Edition, IEEE Guide for Developing System Requirements Specifications

2. Overall Description

2.1 Product Perspective

The product is an application that provides commercial weather forecasting services worldwide. It will utilize ideas passed on Indian Meteorological Department (mausam.imd.gov.in) and the Weather Channel (weather.com/en-IN/).

Both of these sites use cameras to show live feeds of the climate in different areas.

- **System Interface:**

The Air Quality Monitoring & Weather Forecasting System is designed to provide users with real-time air quality information and accurate weather forecasts. The system utilizes the Android SDK 5.1 as the foundation for its user interface, ensuring compatibility with a wide range of Android devices. Through an intuitive and user-friendly interface, users can easily access and interact with the system's functionalities.

- **Operations:**

The system offers a seamless experience to users, allowing them to view current weather conditions, including temperature, humidity, wind speed, and atmospheric pressure. Additionally, it provides accurate air quality data, such as pollutant levels, air quality index (AQI), and associated health recommendations.

Users can perform various operations within the system, such as selecting their location to retrieve localized weather and air quality data. They can also set personalized preferences, including notification settings for specific weather conditions or air quality thresholds.

2.2 Product Functions

The application will gather and show the nearby climate information from the live video feed. It will likewise investigate the climate hypothesis to enable clients to analyze weather forecast. The application will likewise have a connection to a climate site where it will show the present temperatures alongside a 5-day forecast.

- Smart Dissipation System
- Evaluation Option For Users
- Economical
- Supervision And Review
- Providing Information On Forecasted Weather & Agro-Met Advisory Services
- User interaction programs -Conducted at State, District levels

2.3 User Classes and Characteristics

The intended group for the application won't require special expertise or any special experience at all. They only require fundamental computing skills and fundamental knowledge of how to use Internet web pages.

- **Current Weather**
  a) This will show us the current weather.
  b) It is displayed on the main screen.
- **Hourly Weather**
  a) By swiping right one time we can see hourly weather.
  b) Further info can be displayed by clicking on any hour.
- **Daily Weather**
  a) User can view weekly weather by swiping right
  b) We can click on any day of the week for further information.
- **What to Wear**
  a) User will be able to know how to dress accordingly
- **Allergies**
  a) User will be updated about the dust and dander.
  b) User will be able to look at the weekly rate of dust.
- **Fahrenheit**
  a) User can change the temperature into Celsius or Fahrenheit.
- **Multiple forecasts**
  a) Users can see forecast of multiple places at a time.

2.4 Operating Environment

The product is a weather update application. We can download it on any android/apple device. The application's forecasts and warning services are based on weather information derived from numerous sources, including weather observations and data gathered by the National Weather

Service and meteorological organizations. This application is linked with Google maps so it can find our current location tell us about the weather, temperature etc. Since this is a weather forecast application, graphics don't matter much but still the graphics are quite good.

Frontend: Flutter, React Native or React based front-end Frameworks.

Backend: Firebase or Supabase for real time data management, or some back-end Framework.

APIs and Services: Open Weather & AQI APIs and webhooks for real time forecasting data and analytics.

2.5 Design and Implementation Constraints

As we know that the product is a weather forecast application so the major design constraint is the mobile platform. This application is more concerned with their accuracy about the prediction. And being one of the very accurate sites. Graphics are not one of the top priorities. As the application is actually designed for mobiles so the resolution for the product and the screen size limitation will be a big design consideration. Therefore to design an interface that will attract the user eye and is easy to navigate will be a tough job. As the mobiles have limited processing speed and storage memory so these things should be under consideration while implementation process. The product is meant to be quick even when a large amount of users are interacting with it, so each of the functionality should be design and implemented while promising the efficiency of the product.

2.6 User Documentation

For user documentation and information, please consult section 3: External Interface Requirements.

2.7 Assumptions and Dependencies

Our product will depend on some dependencies. There are two types of dependencies (internal and external).

- Internal Dependencies

The product will be only depending upon hardware which is the only internal dependency that we have to care about.

- External Dependencies

  Some of the features of the application will rely on some of the hardware components of the android mobile systems. First the app will be depending upon the memory of the android mobile systems. Therefore the product installation will be depending upon available storage memory in the android mobiles. It also depends on the location. We can manually put in our location to check weather. But we turn on our location services, things will be way easier.

3. System Design

Now we have to design our system before implementing it in such a way that it can execute all the work we want without any loss of data. For this purpose, we have different kind of diagrams to represent the system interaction flow the data storage.

3.1 System Analysis

The system analysis phase is considered to be one of the most important phases in the system development life cycle. It is immensely important that the software developer make through study of the existing system. Thorough study of the system is made and need i.e. features that are critical to system success and users wants (i.e. features that would be good but not essential) are brought out. The study will enable the developer to know the intricacies of the existing system.

3.2 Use Case Diagram

A use case diagram at its simplest is a representation of a user's interaction with the framework that demonstrates the relationship between the user and the different use cases in which the user is involved. A use case diagram can recognize the different types of users of a system and the different use cases and will frequently be accompanied by other types of diagrams as well.

The following figure is a use case diagram which illustrates the interactions between the user and various functionalities that are present in the application and how the user accesses the information.

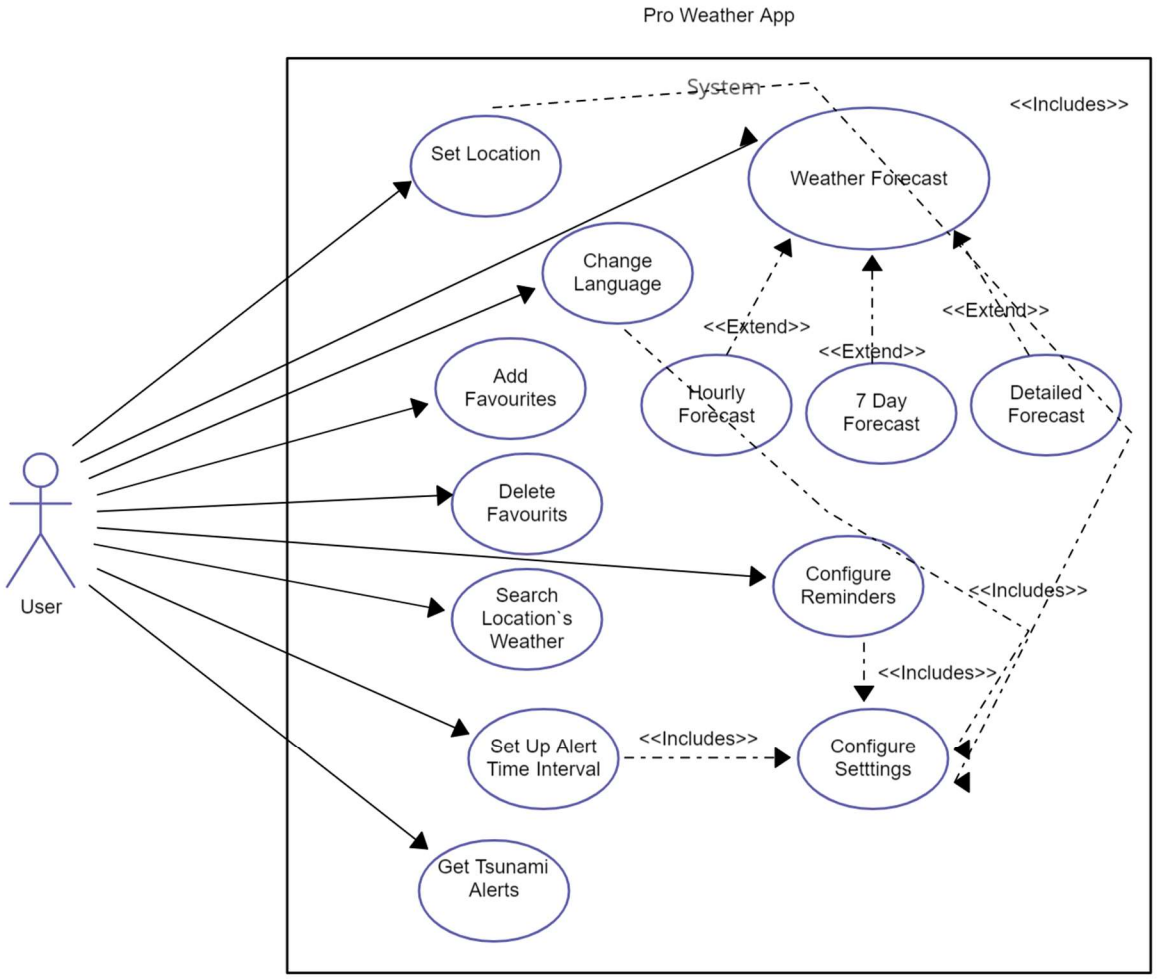

Fig. - 1 Use Case Diagram

3.3 Class Diagram

A class diagram is an outline of the relationships and source code dependencies among classes in the Unified Modeling Language (UML). In this context, a class characterizes the methods and variables in an object, which is a particular entity in a program or the unit of code representing to that entity.

Classes of weather application are:

1) Admin: Manages all the services and users

2) Member: Manages all the operations of the member of the application.

3) Guest: Manages all the operations of the user accessing the application in guest mode

4) Weather: Track all the details about the weather conditions and forecasts within the given dates.

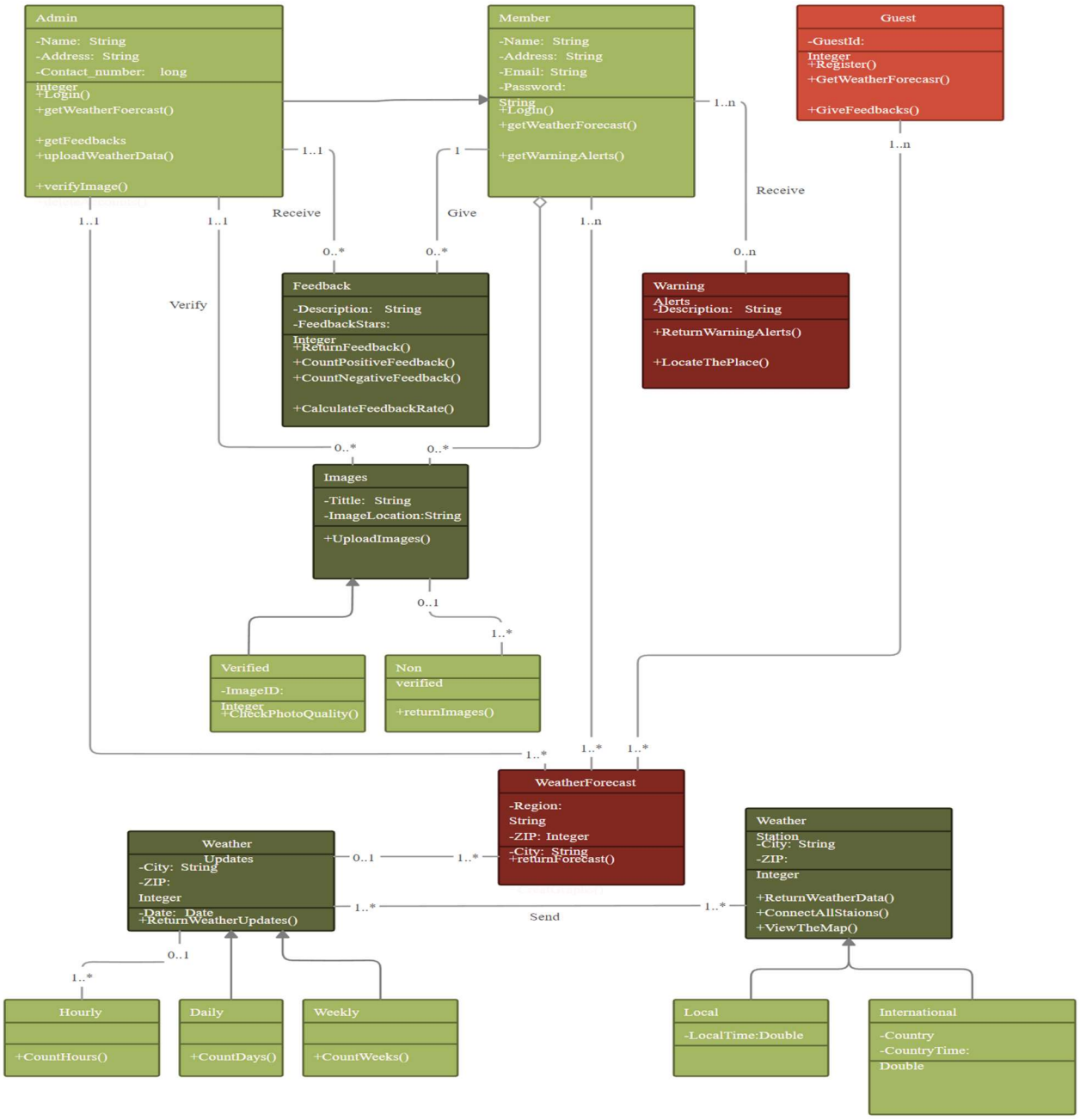

Fig. -2 Class Diagram

3.4 Sequence Diagram

A Sequence diagram is a connection chart that shows how procedures work with each other and in what request. It is a build of a Message Sequence Chart. A sequence diagram shows object collaborations orchestrated in time arrangement. It portrays the articles and classes included in the situation and the succession of messages traded between the items expected to do the situation's usefulness. Arrangement graphs are normally connected with use case acknowledge in the Logical View of the framework a work in progress. Succession outlines are now and then called occasion charts or occasion situations.

The following figure shows the sequence diagram shows the data sequence of the application.

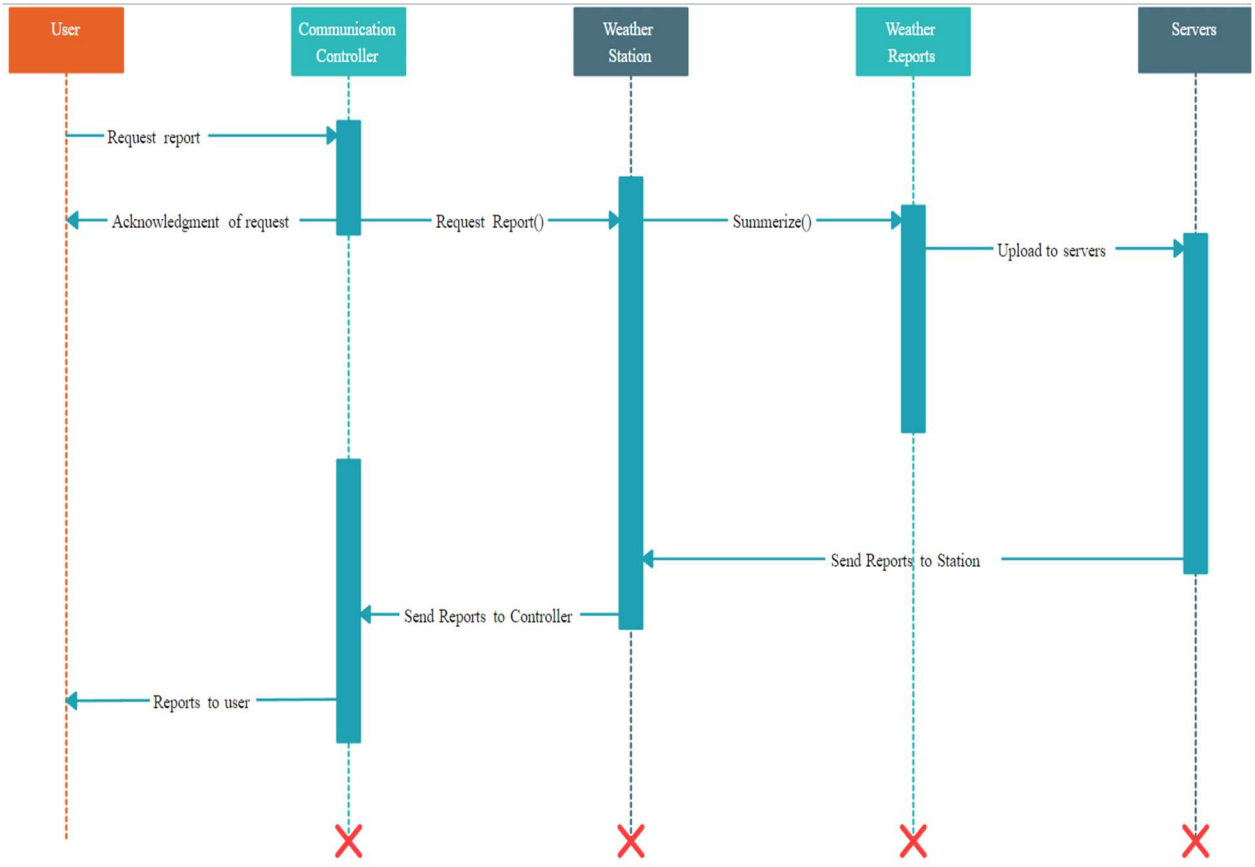

Fig. - 3 Sequence Diagram

3.5 Activity Diagram

Activities diagrams are graphical representations of work processes of stepwise exercises and actions with backing for decision, emphasis and simultaneousness. In the Unified Modeling Language, movement outlines are planned to display both computational and authoritative procedures (i.e., workflows). Activity charts demonstrate the general stream of control.

The following figure show how the data is sent and collected in the central server using an activity diagram.

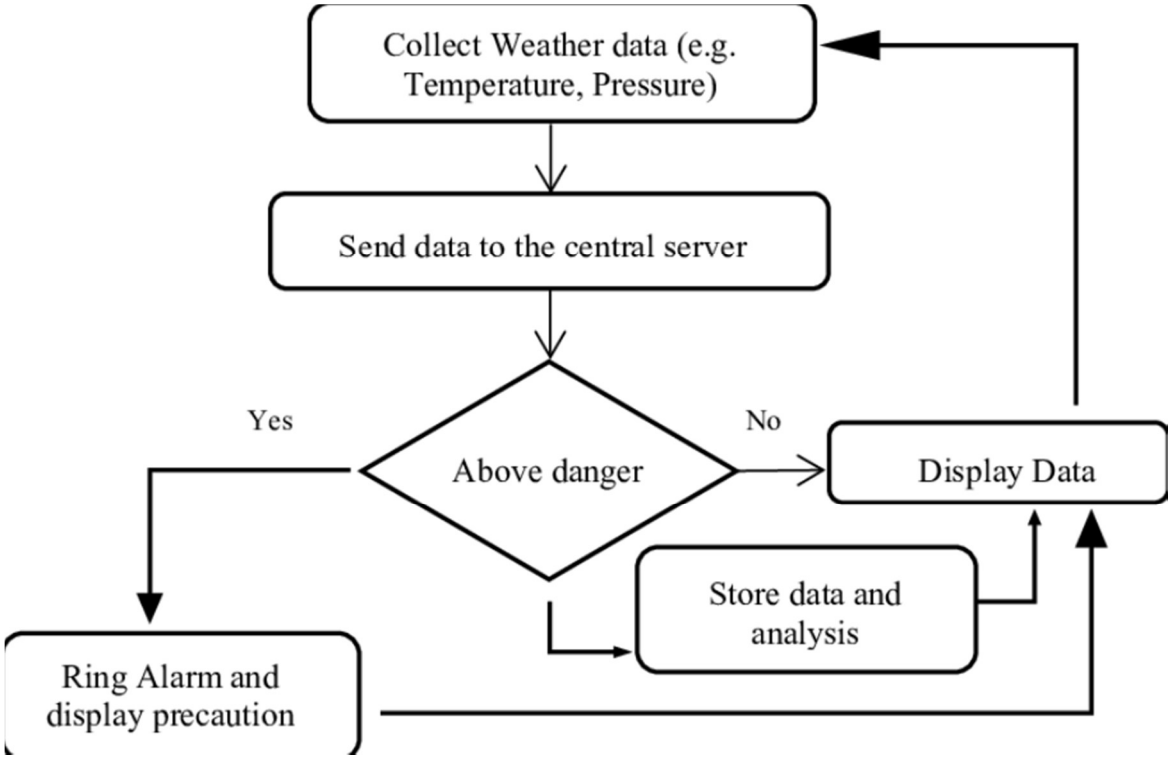

Fig. - 4 Activity Diagram

4. External Interface Requirements

4.1 User Interfaces

This system will provide a graphical user interface. The layout of the system will be a homepage of the app with a header on the top which displays the name of the website, under that there will be a navigation bar to go to related pages and have buttons to login and to go for help, under the navigation bar we have the main body which displays the live weather stream on one half of the page and displaying forecast on the other half, and at the end of the page there will be a footer which displays the contact information.

4.2 Hardware Interfaces

The physical GPS is managed by the GPS application in the mobile phone and the hardware connection to the database server is managed by the underlying operating system on the weathering recording system and the web server.

- Android
- iOS
- Desktop

4.3 Software Interfaces

The software interfaces that will be used are Android Studio and Postman. Android Studio will be used for the coding of the app and for buttons.Postman will be used to maintain the database to store the forecast. So that the weather can be stored and users can view them as they want and to give the overly month weather condition.

4.4 Communication Interfaces

The communication between the different parts of the system is important since they depend on each Other .There is two interfaces that our system will interact with. The first interface is the Weather Channel website to provide the forecast. The second interface is the camera that the live weather feed will come from. Also there are social networks such as face book, twitter that help

communication between the clients or costumers and the workers at the station for more detail or clarification about the service provides by the system.

5. System Features

Following are the functionalities of the application that are must for the system to work properly, to fulfil its purpose, and provide a desirable output.

5.1 User Location and Welcome

When the application is installed, user presses the icon and run it. The application will display a logo screen, and afterwards ask permission to locate the user automatically. If granted, the application will find the user location using the device's GPS, otherwise, the user can manually input his location by entering his desired city/state name.

5.1.1 Description and Priority

For the user to get the area weather, it is must that he let the application detect his location or provide a location manually, so it can give the output of the desired area. This feature has the top priority without this app won't run.

5.1.2 Stimulus/Response Sequences

Step1: User will launch the application.

Step2: Application asks for permission to access location.

Step3: User shall permit the application to access the location, or enter a location manually.

Step4: Application saves the location of the device.

5.1.3 Functional Requirements

REQ-1: Application shall ask to access the location.

5.2 Hourly Weather Update

This application shows us weather on hourly basis. When we press the icon and run this app the screen that appears shows the current weather. But if we swipe the main screen one time we will be able to see weather by every hour of the day. And if we click on any hourly weather it shows us all the weatherly updates like rain in millimeters, dew point, snow, wind speed, humidity, ice, wind gusts, UV index and cloud cover. On the same screen there are two more icons one for to swipe left for previous hour update and one for to swipe right for next hour update.

5.2.1 Description and Priority

For the user to get hourly weather they shall swipe the screen to the right one time. So that they can view hourly weather of current day.

5.2.2 Stimulus/Response Sequences

Step 1: user shall swipe right one time

Step 2: user might click on the current hour for further information.

5.2.3 Functional Requirements

REQ-1: user shall launch the app

REQ-2: user shall swipe right one time

REQ-3: user shall click on the current hourly weather

5.3 Current Weather

This application shows the current weather. When we press the icon and the app runs. The screen that appears is of current weather. It shows us weather in words with degrees outside. Also there is pictorial view at the right side of the screen. So that we can easily know what type of weather it is. There is small icon of an arrow on the right side that shows us details of current weather.

Following are the details:

- Humidity
- UV Index

- Wind Speed
- Cloud Cover
- Wind Direction
- Wind Gust
- Dew Point
- Pressure
- Visibility

5.3.1 Description and Priority

For the user to get instant weather update they will simply open this app. So that they can view current weather of the day. This feature's priority is high because this is like main screen of the app.

5.3.2 Stimulus/Response Sequences

Step 1: user might click on current weather for further information.

5.3.3 Functional Requirements

REQ-1: user shall click the app icon

REQ-2: user shall click on the current weather for further information.

5.4 Allergies

This feature shows expected allergies of the current day. For all the people who need precautions. It is a very useful app for them. It tells us about the dust and dander rate whether it is high or low. It has an option of weekly outlook. By clicking on it tells us about weekly dust rate.

5.4.1 Description and Priority

User can get precautions regarding allergies through this app. They can view rate of allergies by swiping down. This feature's priority is medium.

5.4.2 Stimulus/Response Sequences

Step 1: user will scroll down

Step 2: if user wishes to see weekly outlook they can click on it.

5.4.3 Functional Requirements

REQ-1: user shall click the app icon

REQ-2: user shall scroll down

REQ-3: user shall click on weekly outlook.

5.5 Appropriate Clothing

This app helps us select appropriate clothing according to the weather. It shows us when we open the app but we can also enable it, so it can notify us on daily basis. It mainly gives us reminder to wear a jacket or take an umbrella when going outside. When we open the app. This reminder is on the top of the screen.

5.5.1 Description and Priority

This is for user to wear a jacket or take an umbrella when going outside. It has comparatively low priority.

5.5.2 Stimulus/Response Sequences

Step 1: user can see on the top of the screen
Step 2: user can enable it for notification.

5.5.3 Functional Requirements

REQ-1: user shall click the app icon

REQ-2: user shall scroll down

REQ-3: user shall enable it

5.6 Global News

This application keeps us updated on the global news.

5.6.1 Description and Priority

This is for user to be updated about the global weatherly news. It has medium priority.

5.6.2 Stimulus/Response Sequences

Step 1: user will have to swipe right 4 times.

Step 2: user can view any news he wishes to watch.

5.6.3 Functional Requirements

REQ-1: user shall click the app icon.

REQ-2: user shall swipe right 4 times scroll.

REQ-3: user shall click on any news he wants to see.

5.7 Settings

Settings enable user to manipulate context according to their ease. Through settings we can change units of the wind gust, dew point, pressure, visibility etc. we can change date and time. Settings give us authority to decide if we want this app to notify us about clothes, weather and news. We can change themes. We can go through terms and conditions and privacy.

5.7.1 Description and Priority

Settings enable us to make changes according to our ease. We can change the notification settings. This feature has high priority.

5.7.2 Stimulus/Response Sequences

Step 1: click the settings.

Step 2: make any desired changes.

Step 3: save the changes.

5.7.3 Functional Requirements

REQ-1: user shall launch app

REQ-2: user shall click on three dots

REQ-3: user shall click on settings.

5.8 Fahrenheit

This application enables user to converts the temperatures according to their ease. They can view it in Fahrenheit as well as Celsius degree. If a company is using it for business purposes they might require temperature in Fahrenheit. This function will then come in handy.

5.8.1 Description and Priority

It is for user to view weathers of multiple places. This feature has a high priority since no one restricts themselves to one place only.

5.8.2 Stimulus/Response Sequences

Step 1: user will see three dots on the top right of the screen.

Step 2: user will choose the option.

5.8.3 Functional Requirements

REQ-1: user shall click the app icon.

REQ-2: user shall click the three dots.

REQ-3: user shall click the option ‘change to Fahrenheit'.

6. Other Nonfunctional Requirements

6.1 Non-functional requirement

Nonfunctional requirements specify whether a particular task will be performed or not. Some of the no non-functional requirements are listed below:

6.1.1 Speed

Speed is actually much more complicated than it seems. Speed is one of the nonfunctional requirements that must be considered. The speed may lag because of many reasons. Some of the common reason for slow load is due to high resolution un-optimized images, complex order entry process, use of many API calls to render, unclean code etc. All of these issues must be solved.

6.1.2 Security and privacy

Since this application use most sensitive data of the user. The system must be protected from vulnerabilities. The authentication must be secured from data breach. To protect the system from code manipulation, the software must be coded with much complexity. The privacy must be protected. Policies, terms and conditions must be able to update from the admin side.

6.2 Software Quality Attributes

- **Availability**-the item should be available to user.

- **Correctness**-the list of the items to be recommended should be correct according to the details provided by the user.

- **Usability**-the recommended items should satisfy a maximum number of user needs.

- **Maintainability**-admin should maintain the database and store in updated form.

- **Portability**- the application should be portable to android, iOS and Windows.

- **Reliability**- the system should give 98% correct search results out of 1000 searches during testing.

- **Extendibility**-the application should be easy to extend, code should be written in such a way that it favors implementation of new functions.

References

- Object Oriented Modeling and Design with UML-Michael Blaha, James Rambaugh.
- Software Engineering, Seventh Edition, Ian Sommerville.
- **Accuweather** - https://www.accuweather.com/
- **Wikipedia** - https://en.wikipedia.org/wiki/OpenWeatherMap
- **Android Studio** - https://developer.android.com/
- **JAVA** - https://www.java.com/en/
- **Android SDK** - https://developer.android.com/tools/releases/platform-tools