

Weather 24/7
Software Requirement Analysis Document

Submitted by:

SH-01, Tahmid Mosaddek

FH-97, Kazi Shadman Sakib

Submitted to:

Dr. Saifuddin Md. Tareeq
Professor & Chairperson
Computer Science and Engineering, University of Dhaka

Dr. Sarker Tanveer Ahmed
Assistant Professor
Computer Science and Engineering, University of Dhaka

TABLE OF CONTENTS

1. Introduction	3
1.1 Purpose of the system	3
1.2 Scope of the system	3
1.3 Objective and success criteria of the project	3
1.4 Definitions and Acronyms and abbreviations	4
1.5 References	5
1.6 Overview	6
2. Overall Description	6
2.1 Product Perspective	6
2.2 Product Function Detail	6
2.3 User Profiles	7
2.4 Constraint	7
2.5 Assumption and Dependencies	7
3. Proposed System	8
3.1 Overview	8
3.2 Functional Requirement	8
3.2.1 Requirement 1	8
3.2.2 Requirement 2	8
3.2.3 Requirement 3	9
3.2.4 Requirement 4	9
3.2.5 Requirement 5	9
3.2.6 Requirement 6	10
3.2.7 Requirement 7	10
3.3 Non Functional Requirement	10
3.3.1 Usability	10
3.3.2 Reliability	11
3.3.3 Performance	11
3.3.4 Supportability	12
3.3.5 Implementation	12
3.3.6 Scalability	12
3.3.7 Security	13
3.3.8 Maintainability	13
3.3.9 Testability	13

3.4	System Models	14
3.4.1	Scenarios	14
3.4.1.1	Scenario 1	14
3.4.1.2	Scenario 2	14
3.4.1.3	Scenario 3	15
3.4.1.4	Scenario 4	15
3.4.1.5	Scenario 5	16
3.4.1.6	Scenario 6	16
3.4.2	Use cases	16
3.4.2.1	Use Case 1	16
3.4.2.2	Use Case 2	17
3.4.2.3	Use Case 3	18
3.4.2.4	Use Case 4	18
3.4.2.5	Use Case 5	19
3.4.2.6	Use Case 6	19
3.4.3	Use case model	20
3.4.4	Dynamic model	21
3.4.4.1	Sequence Diagram	21
3.4.4.2	Activity Diagram	21
3.4.4.3	State Diagram	22
3.4.5	User Interface	22
3.4.5.1	User Interface	22
3.4.5.2	Software Interface	25
3.4.5.3	Hardware Interface	26
4.	Supporting Information	26

Description of Contents

1. Introduction

1.1 Purpose of the system

Accurate weather forecasts have been an ever-increasing demand for ages. For which we always needed to know about the weather parameters such as temperature, humidity, precipitation as quickly and as accurately as possible. Weather 24/7 forecasts all these parameters and their effects. This software application notifies the user with in-detailed information about daily weather. Any user can use this software to know about the weather at the beginning of the day and take action accordingly. Big enterprises like Airline companies and agricultural industries can also benefit from using this application. From scheduling an arrival or departure time of a plane to applying fertilizer at the right moment according to the weather prediction, the possibilities of the use of this application are limitless.

1.2 Scope of the system

The main objective of this system is to provide its daily user with accurate information about the weather. This will be done by scraping data from popular weather websites using the help of APIs. So this software system will be able to provide the weather information of almost any region of the world. So, any user from any part of the world can use this system to know his local weather or search the location of a place to know the weather of a distant place.

1.3 Objective and success criteria of the project

The main objective of this application is to provide its user with accurate data and predictions of the weather. Also, present an interactive and easy-to-use interface for the users. This application aims to notify its users about bad weather beforehand so that they can take the necessary steps.

To ensure the success of this software project, all the fundamental software engineering activities will be followed. First, the requirements and the architecture of this software project will be analyzed, where every possible scenario will be checked. To represent the functionality of the system from the

user's point of view, Use-case diagrams will be designed. And, to describe the Software System, the Dynamic model of the software will be designed to describe the system's reaction to external events. After this, the wireframe of User-Interface, Software Interface, and Hardware Interface will be designed to make sure the software project is user friendly, has efficient responsiveness and processing time. Implementation of the software project will be done accordingly with the help of the designed wireframes. To get user-specific local weather data, we will search for APIs that suit the demand of this project. Then, the implemented features will be tested to ensure the quality of each smaller unit. This software should deliver the required functionality to the user and should be efficient, dependable, and usable.

1.4 Definitions and Acronyms and abbreviation

- **UI:** Shorts for User Interface.
- **UX:** Shorts for User Experience.
- **API:** An application programming interface (API) is a set of programming codes that queries data, parse responses, and sends instructions between one software platform and another.
- **JAVA:** Java is an Object-Oriented Programming Language.
- **XML:** Extensible Markup Language (XML) is a markup language and file format for storing, transmitting, and reconstructing arbitrary data. It defines a set of rules for encoding documents in a format that is both human-readable and machine-readable.
- **JSON:** Javascript Object Notation is an open standard file format and data interchange format that uses human-readable text to store and transmit data objects consisting of attributes.
- **Parse JSON:** Parse JSON is the process to parse the JSON file format and use its contained data for the Software System.
- **Unit Testing:** Unit testing is a software testing method by which individual units of source code—sets of one or more computer program modules

together with associated control data, usage procedures, and operating procedures—are tested to determine whether they are fit for use.

- **GitHub:** GitHub is a website for developers and programmers to collaboratively work on code. The primary benefit of GitHub is its version control system, which allows for seamless collaboration without compromising the integrity of the original project.
- **IDE:** An integrated development environment (IDE) is a software application that provides comprehensive facilities to computer programmers for software development.
- **Geolocation Service:** Geolocation Service is the process of determining the location of a computer, phone, or other network-based devices. This inferred location is based on geographical measurements of latitude and longitude to narrow down the location to city, zip code, street, and even address.
- **Target SDK version:** The target SDK version is the version an application was targeted to run on.

1.5 References

List of external websites used to fetch weather data from:

1. <https://www.accuweather.com>
2. <https://www.windy.com>
3. <https://openweathermap.org>

List of materials used to write this document:

1. <https://personal.utdallas.edu/~chung/RE/IEEE830-1993.pdf>
2. Object-Oriented Software Engineering Using UML, Patterns, and Java(Third Edition).

List of online drawing tools to used to draw diagrams presented in this document:

1. <https://online.visual-paradigm.com>
2. <https://www.figma.com>

1.6 Overview

1. **Introduction:** This section provides a general overview of this Requirement Analysis Document (RAD). It tells us about the purpose and scope of the project very briefly, defines the abbreviations and some terms used in this RAD, and mentions the references.
2. **Overall Description:** This section describes the perspective and functionalities of the project. It provides us with a basic overview of the entire project as well as the constraints, assumptions, and dependencies that come with it.
3. **Proposed System:** This section points out the functional and non-functional requirements of the project. All the necessary models and diagrams such as the Use Case model, Sequence diagram, Activity diagram, and state-chart diagram have been depicted.
4. **Supporting Information:** This section is for the supporting information that might be necessary.

2. Overall Description

2.1. Product Perspective:

The user of this product can know the current weather of his location along with the prediction of the next 30 days. He can also get the weather of a remote place by inserting the name of the place in the location search bar.

The product also aims to alert the user if there is going to be extreme or hazardous weather today by sending a notification.

However, this product is not going to be self-contained as it requires data from external web servers to present to the user.

2.2. Product Function Detail

- Simple UI where the user can insert the location of a place in the search bar and learn about the weather of that place.
- Provide the user with accurate information about current weather.
- Provide the user with the weather prediction of upcoming days.
- Notify the user early on if there is going to be bad weather outside.

2.3. User Profile

Right now we are not planning to add any user profile. In the future, we may try to implement this feature, a separate account for separate users.

2.4. Constraints

2.4.1. Dependency on external web servers

The application will show information based on the data fetched by APIs from different websites. So, if the web servers are down, the application will show out-of-date information or may not show any information at all.

2.4.2. Location

This application requires the user to insert a valid name of a location in the location search bar. If an invalid location is inserted, the application will keep showing the previous data.

2.5. Assumptions and dependencies

2.5.1. Assumptions:

- The device is assumed to be connected to the internet to fetch data from external web servers.
- The users of this application are assumed to use the English language to insert the location of a place.
- The users are assumed to understand the English language, as every piece of information will be shown in this language.

2.5.2. Dependencies:

- Geolocation API to get access to the user's location.
- Weather APIs to fetch data from external web servers.

3. Proposed System

3.1 Overview

Proposed Weather 24/7 will serve its users simply by providing the users with daily and future 30 days weather information. It is a weather-based android system where the user will get all sorts of weather information. From basic to detailed weather information will be provided to the users. Users can also search for location-specific weather information. The system will notify its users about bad weather and also predict the next 30 days of weather information. Users can change the unit of weather parameters according to their needs. As weather information can be useful throughout the day for everyone, reaching out to everybody with accurate weather information is the first priority.

3.2 Functional Requirement

3.2.1 Requirement 1

ID	01
Name	Location Specific Weather Search Bar
Description	RegularUser can search a specific location to know about that area's weather information.
Priority	High

3.2.2 Requirement 2

ID	02
Name	Predicted Weather
Description	RegularUser gets the predicted weather information for the next 30 days.
Priority	High

3.2.3 Requirement 3

ID	03
Name	Detailed weather information
Description	RegularUser can know about today's detailed weather information.
Priority	High

3.2.4 Requirement 4

ID	04
Name	Changing parameters
Description	RegularUser can change the weather parameters according to his needs.
Priority	High

3.2.5 Requirement 5

ID	05
Name	Add to the home screen
Description	RegularUser can add the basic weather information to their home screen.
Priority	Medium

3.2.6 Requirement 6

ID	06
Name	If an invalid location is given by the user
Description	If the RegularUser inputs an invalid location to the system, it will show the user that it can not find the invalid desired location.
Priority	High

3.2.7 Requirement 7

ID	07
Name	If device location is turned off
Description	If the device of the RegularUser is turned off, the system will detect it and will ask the user to turn on the location, thus the RegularUser will get a button to turn on their device location.
Priority	High

3.3 Non Functional requirement

3.3.1 Usability

Weather 24/7 android system will show local, national, and global daily weather forecasts in the simplest way possible. Users will get to enjoy the simple user interface while having the best user experience. The system will auto-detect all its needs and run thereby. User location will be detected by the android system within 1 second as an input. In the location search bar case, the location provided by the user should also be acknowledged within 1 second as an input. Predicted next 30 days weather information is shown with a calendar to simplify the user

experience.

The android system will be as simple and user-friendly as possible. The user experience of the system will be great as the user interface will be more informative about the weather information and faster to detect the change in weather.

3.3.2 Reliability

Weather 24/7 will consistently provide the user with reliable information about the weather. As this application fetches information from popular weather websites using the help of APIs, the room for error is very little. The application can reliably show accurate information about the weather as long as those web servers are running.

When no location is specified, the location provided by the Geolocation API is used in this application to show the weather data of the user's location. Since the location given by Geolocation API is known to be consistent, the user requirement to get the weather information of a place should also be reliable.

As the weather APIs provide weather information in user-acceptable time, the responsiveness of the system can also be said to be reliable.

3.3.3 Performance

The performance of the android system depends on the APIs used in this system. As the weather information would be fetched from the APIs, the system should store the fetched data for the case if there is no internet available on the user's device. Along with this data, the previously specified location of the user also should be stored in case the device location is also turned off in the user device. If there is no internet available on the user's device, the system should show the stored weather information within 1 second time. A system crash should not result in data loss of the weather information of the last acknowledged location. As the

user gets a notification whenever there is bad weather incoming, the alert should be given to the user within 1 second of time whenever the system acknowledges the bad weather.

3.3.4 Supportability

As this is a 3rd Year Undergraduate Software Engineering project, There will be no dedicated support team to provide real-time maintenance service for the android system. But, as the software engineers of this android system will open a Github repository and work using the git environment to develop the android system, there will be options given in the system to give feedback and suggestions for more features using the Github issues. Users can open an issue using the system's Github link. The engineers will try their best to fix and/or add more features identified by the users.

3.3.5 Implementation

The backend of the system will be built using Java. If the user does not provide any location, we will take the user's location provided by the Geolocation API. Otherwise, the location given by the user will be used. This location information will be sent to the selected weather APIs which will collect weather data of that location from previously mentioned web servers.

The user interface of this application will be designed using XML. And Android Studio will be used as an IDE to develop this android application.

3.3.6 Scalability

The system will be built keeping in mind that it will have the capability to adapt to any environment it goes through.

3.3.7 Security

The system will use the user's device location using the Geolocation API that returns a location and accuracy radius based on information about cell towers and WiFi nodes that the mobile client can detect. The system will use this information only when the application is being used.

Communications between the application and the APIs are done over HTTPS using the POST method. As both the request and response messages will be encrypted, the user's location will not be leaked if someone eavesdrops on the data packets.

3.3.8 Maintainability

The software engineers of this android system will open a Github repository and work using the git environment to develop the android system, there will be options given in the system to give feedback and suggestions for more features using the Github issues. Users can open an issue using the system's Github link. The engineers will try their best to fix and/or add more features identified by the users.

3.3.9 Testability

We will conduct unit testing and debugging to ensure the maximum reliability of our system.

3.4 System Models

3.4.1 Scenarios

3.4.1.1 Scenario 1

Scenario name	Azwad just wants to know the temperature of the current day.
Participating actors	Azwad: RegularUser
Scenario Description	<ol style="list-style-type: none">1. Azwad was feeling a little bit cold. So he wanted to know about the current temperature of the place he lives in. He opens his android device and searches for Weather 24/7. And he opens up Weather 24/7.2. Azwad finds out the temperature outside of his home is 07.15 °C from the basic information provided by Weather 24/7 on the homepage.

3.4.1.2 Scenario 2

Scenario name	Fawwaz wants to go out today for lunch.
Participating actors	Fawwaz: RegularUser
Scenario Description	<ol style="list-style-type: none">1. Fawwaz wants to go out today for lunch with his friends. So before leaving his home, he wanted to know the detailed information about the weather today. Fawwaz opens his android device and searches for Weather 24/7. And he opens up Weather 24/7.2. Fawwaz goes to the detailed weather information of today's activity page.3. Fawwaz finds out that today at 2 PM it will be raining, thus he takes his umbrella with him to lunch, to avoid the raindrops on him.

3.4.1.3 Scenario 3

Scenario name	Jawad wants to know the weather information of Seattle, Washington.
Participating actors	Jawad: RegularUser
Scenario Description	<ol style="list-style-type: none">1. Jawad wants to know the weather information of Seattle, Washington today. So, he opens his android device and searches for Weather 24/7. And opens up Weather 24/7.2. Jawad clicks on the Search bar on the homepage of Weather 24/7. A keyboard appears before him to write a location name. Jawad writes Seattle, Washington, and clicks on the location name.3. The system automatically changes all the weather information according to the current weather of Seattle, Washington.4. Jawad finds out the information he required about the weather in Seattle, Washington.

3.4.1.4 Scenario 4

Scenario name	Sakib wants to plan a beach party.
Participating actors	Sakib: RegularUser
Scenario Description	<ol style="list-style-type: none">1. Sakib wants to plan a beach party for his friends on a sunny Friday. He wants to know about the weather information for next Friday, which comes after 4 days. Sakib opens his android device and searches for Weather 24/7. And he opens up Weather 24/7.2. Sakib goes to the Weather Prediction activity page of Weather 24/7.3. Sakib clicks on the next Friday in the calendar.4. Sakib now knows that next Friday will be sunny.

3.4.1.5 Scenario 5

Scenario name	Tahmid gets notifications about today's weather.
Participating actors	Tahmid: RegularUser Weather 24/7: System
Scenario Description	<ol style="list-style-type: none">1. Tahmid is ready to leave for his office today. Just before he leaves his home, Tahmid gets a notification/alert from Weather 24/7 that today at 10.15 AM it is going to start raining.2. Thus, Tahmid leaves for his office with an umbrella.

3.4.1.6 Scenario 6

Scenario name	Abdullah wants to change the weather parameter.
Participating actors	Abdullah: RegularUser
Scenario Description	<ol style="list-style-type: none">1. Abdullah wants to see his temperature units in Fahrenheit. Thus he opens his android device and searches for Weather 24/7. And opens up Weather 24/7.2. Abdullah goes to the Settings option in Weather 24/7.3. Abdullah searches for temperature parameters in the Settings. And changes the temperature unit to Fahrenheit.

3.4.2 Use cases

3.4.2.1 Use case 1

Use case name	EnterWeatherHomePage
Participating actors	Initiated by the RegularUser Communicates with Weather 24/7
Flow of events	<ol style="list-style-type: none">1. RegularUser press the application icon on their device.2. Weather 24/7 presents basic information about current weather.3. The user gets access to the location search bar.

Entry condition	The RegularUser is logged into Weather 24/7.
Exit conditions	<ul style="list-style-type: none"> • The RegularUser has received the required information about today's weather, OR • As the RegularUser has searched for a specific location in the location search bar, RegularUser has received the required weather information of that specific area.
Quality requirements	The RegularUser instantly gets accurate information about the current weather.

3.4.2.2 Use case 2

Use case name	LearnTodaysDetailedWeather
Participating actors	Initiated by the RegularUser Communicates with Weather 24/7
Flow of events	<ol style="list-style-type: none"> 1. The user swipes left after entering the application 2. A new page appears with a different interface containing detailed information about today's weather. 3. Weather 24/7 presents the user with current temperature, average temperature, humidity, wind flow, dew point, pressure, UV index, Visibility, hourly precipitation percentage along with the time when it is going to hit the maximum and minimum temperature of the day and sunrise time, sunset time
Entry condition	The RegularUser is logged into Weather 24/7 and swipes left to go to Today's detailed weather activity.
Exit conditions	The RegularUser has received the required information about today's detailed weather.
Quality requirements	The RegularUser instantly gets accurate information about today's detailed weather.

3.4.2.3 Use case 3

Use case name	UseLocationSearchBar
Participating actors	Initiated by the RegularUser Communicates with Weather 24/7
Flow of events	<ol style="list-style-type: none">1. RegularUser presses the search bar and a keyboard pops up.2. RegularUser enters the name of the location that he is willing to know the weather off in the search bar using the keyboard.3. The application presents information about the weather of that location on all of its pages.
Entry condition	The RegularUser is logged into Weather 24/7 and presses the search bar on top.
Exit conditions	The RegularUser has received the required information about the weather of that specific area which the RegularUser has searched for, on all of its pages.
Quality requirements	The RegularUser instantly gets accurate information about the weather of that specific area which the RegularUser has searched for, on all of its pages.

3.4.2.4 Use case 4

Use case name	ChangeSettings
Participating actors	Initiated by the RegularUser Communicates with Weather 24/7
Flow of events	<ol style="list-style-type: none">1. The user presses the “cog” icon at the top right corner of the screen.2. A settings interface appears where the user can select the theme and choose the unit in which the data will be shown.
Entry condition	The RegularUser is logged into Weather 24/7 and presses the “cog” icon at the top right corner of the screen.
Exit conditions	The RegularUser has successfully changed the required settings.

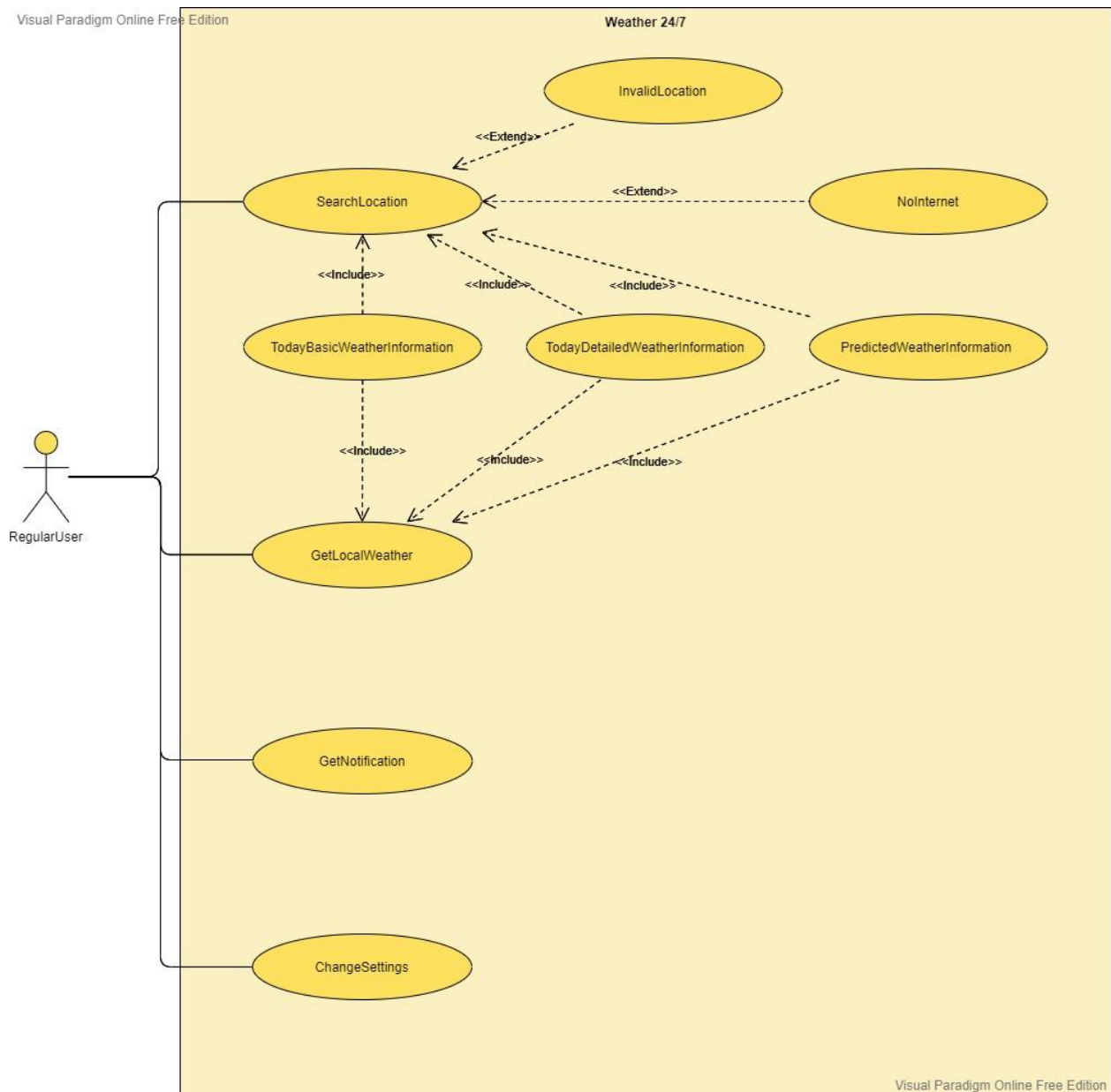
3.4.2.5 Use case 5

Use case name	GetWeatherPrediction
Participating actors	Initiated by the RegularUser Communicates with Weather 24/7
Flow of events	<ol style="list-style-type: none">1. The user swipes right on the home page.2. The application presents a calendar with the weather prediction of the upcoming 30 days.
Entry condition	The RegularUser is logged into Weather 24/7 and swipes right to go to the next 30 days of weather prediction activity.
Exit conditions	The RegularUser has received the required information about the predicted weather for the next 30 days.
Quality requirements	The RegularUser instantly gets predicted information about the weather of the next 30 days.

3.4.2.6 Use case 6

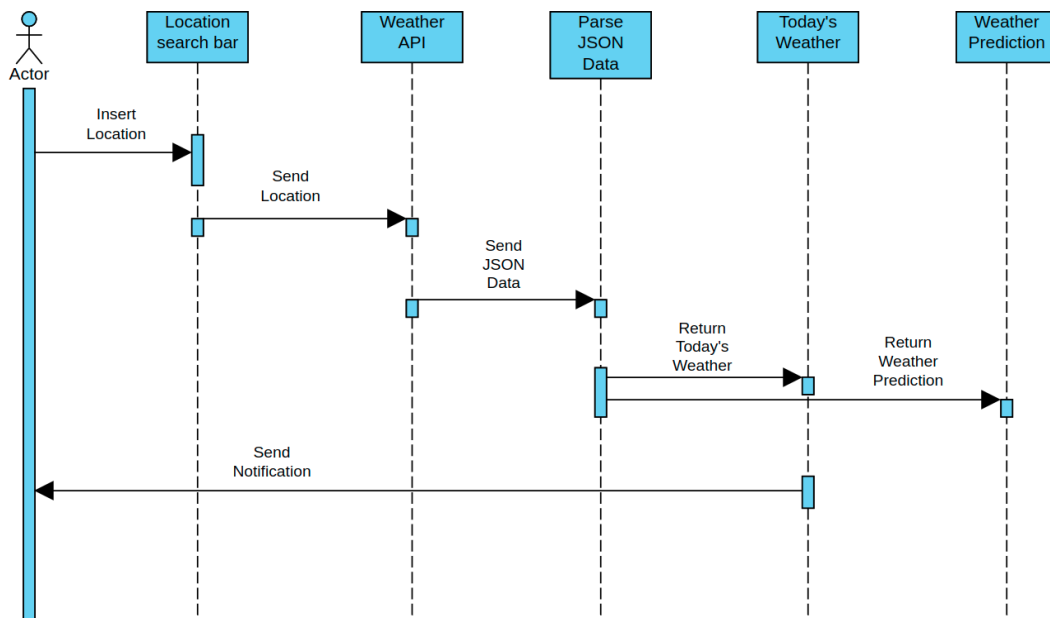
Use case name	GetNotified
Participating actors	Initiated by the Weather 24/7 Communicates with RegularUser
Flow of events	<ol style="list-style-type: none">1. The application notifies the RegularUser with basic information about today's weather.2. The application sends an emergency notification if there is an unprecedented rain or storm.
Entry condition	The RegularUser unlocks his device.
Exit conditions	RegularUser receives the notification.
Quality requirements	RegularUser gets notified in time.

3.4.3 Use case model

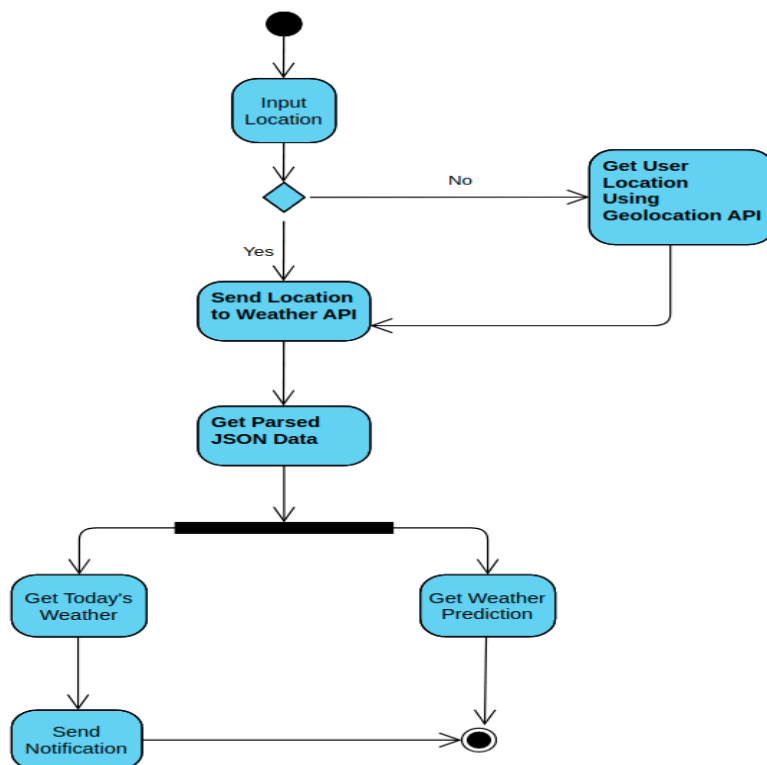


3.4.4 Dynamic model

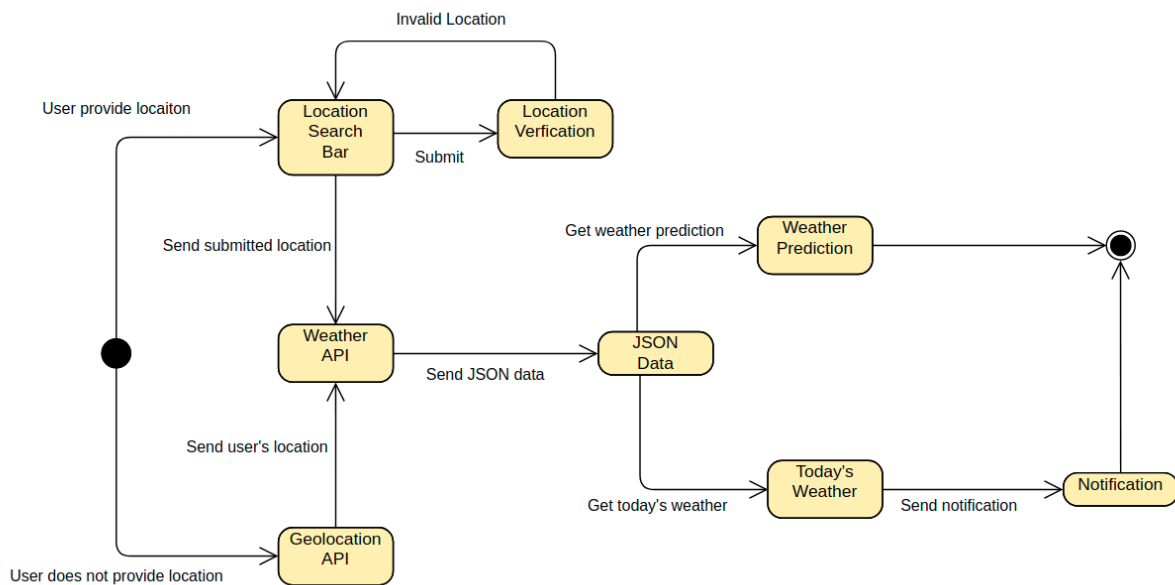
3.4.4.1 Sequence Diagram



3.4.4.2 Activity Diagram



3.4.4.3 State Diagram



3.4.5 User Interface

3.4.5.1 User Interface

- **Home Page:** This is the first activity that a user will face when they open the android software. The Home Page will contain all the basic information of today's weather such as temperature now, how it feels like outside, highest temperature today, and the lowest temperature today along with today's date and time. The Home Page will also contain a search bar so that a user can search a specific location manually and change the weather information according to the location provided.

Our main UI design for the "Home" activity page will be inspired by the below design.



Resource: <https://www.pinterest.com/pin/481955597631160956/>

- **In Detailed Weather Page:** This activity is confronted by a user when the user swipes left after entering the application. This interface contains detailed information about today's weather, such as current temperature, average temperature, humidity, wind flow, dew point, pressure, UV index, Visibility, hourly precipitation percentage along with the time when it is going to hit the maximum and minimum temperature of the day and sunrise time, sunset time. Our main UI design for the "In Detailed Weather" activity page will be inspired by the below design.



Resource: <https://www.pinterest.com/pin/257338566178682259/>

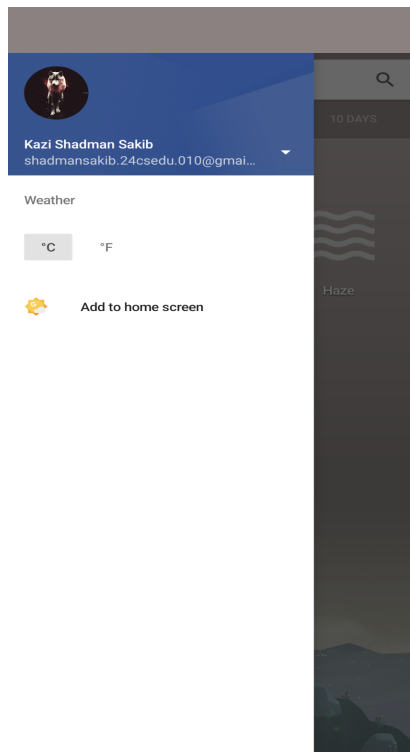
- **Weather Prediction Page:** This activity will appear when the user swipes right after entering the application. This interface contains a calendar with the weather prediction of the upcoming 30 days.

Our main UI design for the "Weather Prediction" activity page will be inspired by the below design.



Resource: <https://www.pinterest.com/pin/5488830786508615/>

- **Settings Page:** The settings page will appear when the user presses the “cog” icon at the top right corner of the screen. This interface is where the user can select the theme and choose the unit in which the weather data will be shown. Our main UI design for the "Settings" activity page will be inspired by the below design.



Resource: Phone screenshot of Google Weather App.

3.4.5.2 Software Interface

- This system will mainly be an android application. Java The required API level for this system is 23+. The minimum target SDK version for this system is 30. Geolocation API and Weather API will be used to fetch weather data according to the user device location. When the user location is detected/manually inserted the location data will be added to the API “/location” path and a request to fetch the weather data will be sent by the system to the API

server. After that, the API server will respond with the weather information that is needed.

3.4.5.3 Hardware Interface

- This android application should be compatible with any android device which has android version 6.0 and up. To run this android application smoothly it should have a working and stable internet connection in order to fetch weather information of a specific location.

4. Supporting Information

Currently, we do not have any supporting information that needs to be documented. If we encounter such necessary information, we will update it as soon as possible.