

(After 4 enter spaces)

**Project ID- 00**

**Project Advisor: Name of Project Adviser**

**Submitted By**

Name of Student ID of Student

Name of Student ID of Student

**University of Education**

Inner Title Cover Specimen

(After 3 enter spaces)

**Type Your Project Title Here**

(14 pt. Bold, Capitalized Each word, Align Centre)

(After 6 enter Spaces)

#### BS in Information Technology-20XX MSc in Information Technology-20XX

(14 pt. Bold, Capitalized Each word, Align Centre)

(After 6 enter times)

A project submitted in partial fulfillment of the requirements for the award of the degree of

BS in information Technology /

MSc in Information Technology

(After 11 enter times)

**NAME OF DIVISION/CAMPUS UNIVERSITY OF EDUCATION LAHORE**

November 2018

© Copyright A B creed, 2018

“I hereby declare that I have read this project documentation and in my opinion this project is sufficient in terms of scope and quality for the award of the degree of BS in Information Technology/MSc in Information Technology.”

|  |  |
| --- | --- |
| \_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Project Primary Supervisor  Name: Write name of Project advisor here  Designation: Write designation of Project Advisor here  University of Education. | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Project Examiner  Name: Write name of Project advisor here  Designation: Write designation of Project Advisor here  University of Education. |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| **DECLARATION**   I declare that this project title entitled “**write your project title here**” is the result of my own research and development except as cited in the references. This project has not been accepted for any degree and is not concurrently submitted in candidate for any other degree. At any time if my statement is found to be incorrect even afterwards of BS in Information Technology/MSc in Information Technology, the university has the right to withdraw my BS in Information Technology/MSc in Information Technology degree. |

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Month date, year

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Month date, year

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Month date, year

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Month date, year

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# **PLAGIARISM UNDERTAKEN**

I solemnly declare that project work presented in this documentation entitles **“Name of your project**” is solely my work with no significant contribution from any other person. Small contribution/help wherever taken has been acknowledged and that complete project has been written by me.

I understand that zero tolerance policy of the HEC and University of Education, Lahore towards plagiarism. Therefore, we as an author of the above titled project declare that no portion of my project documentation and any material used as reference is properly referred/ cited.

I undertake that of I am found guilty of any formal plagiarism in the above titled project even after award of BS/MSc degree, the University reserve the rights to withdraw/revoke my BS/MSc degree and that HEC and the University has the right to publish my name on the HEC/University Website on which names of students are place who submitted plagiarized projects.

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

NAME IN CAPITAL LETTERS

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Month date, year

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

NAME IN CAPITAL LETTERS

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Month date, year

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

NAME IN CAPITAL LETTERS

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Month date, year

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

NAME IN CAPITAL LETTERS

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Month date, year

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| **CERTIFICATE OF APPROVAL**   This is to certify that the project work presented in this documentation entitled, “**write name of your project”**, was conducted by **“name of member 1”, “name of member 2”, “name of member 3”, “name of member 4”**, under the supervision of **“write your supervisor name”**. No part of this project has been submitted anywhere else for any degree. This project is submitted to the **“name of your campus, University of Education”** is partial fulfillment of the requirements of the degree of BS in Information  Technology/MSc in Information Technology. |

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

NAME IN CAPITAL LETTERS

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Month date, year

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

NAME IN CAPITAL LETTERS

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Month date, year

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

NAME IN CAPITAL LETTERS

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Month date, year

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

NAME IN CAPITAL LETTERS

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Month date, year

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Project Primary Supervisor

Name: Write name of Project advisor here

Designation: Write designation of Project Advisor here

University of Education.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Project Examiner

Name: Write name of Project Examiner here

Designation: Write designation of Project Examiner here University of Education.

# **OFFICE OF CONTROLLER OF EXAMINATION NOTIFICATION**

No: \_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_\_

It is notified for the nomination of all the concerned that Mr./Ms. (name of the student-1), (name of the student-2), (name of the student-1), (name of the student-1) BS/MSc student of Name of the department of University of Education has completed all the requirements for the award of BS/MSc Degree in the discipline of Information Technology as per detail given hereunder:

**BS in Information Technology/ Cumulative Result**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **MSC in Information Technology** | | **Credit Hours:\_\_\_\_\_\_\_\_\_\_\_** | | | **Cumulative**  **Grade Point**  **Average**  **(CGPA)** |
| **Registration No** | **Complete Name** | **Course work** | **Project** | **Total** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Project Title: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name of Supervisor: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Signed by**

**Controller of Examination**

**CC:**

1.abcd

2. xyz3

# **ACKNOWLEDGEMENT**

We truly acknowledge the cooperation and help make by Name of **Acknowledger**, **Designation** of **Address of Organization**. He has been a constant source of guidance throughout the course of this project. We would also like to thank **Acknowledger** from **Designation**, **Address of Organization** for his help and guidance throughout this project. We are also thankful to our friends and families whose silent support led us to complete our project.

1. Mr. Furqan
2. Mr. Akram

Date:

March 11, 2016

# **ABSTRACT**

A good abstract explains in a few lines what was actually done and why the work was important. The maximum number of words should not be more than 1000 words. The abstract page may include the following:

* Background
* Objectives/aim of the study
* Research methodologies
* Findings
* Conclusions
* Implications
* Limitations

Contents

[**1** INTRODUCTION: 3](#_Toc197346201)

[**1.1** **PROBLEM DESCRIPTION:** 3](#_Toc197346202)

[**1.2** **WEATHER WISE GOAL AND OBJECTIVES:** 4](#_Toc197346203)

[1.1.1 Addressing the Needs of Rural and Remote Areas 4](#_Toc197346204)

[1.1.2 Real-Time Alerts for Severe Weather Conditions 4](#_Toc197346205)

[1.1.3 Enhancing Safety and Responsiveness 4](#_Toc197346206)

[1.1.4 Designed for Both Urban and Rural Users 4](#_Toc197346207)

[**1.3** **OBJECTIVES:** 5](#_Toc197346208)

[**1.4** **WEATHER WISE: CONCEPT AND VISION** 5](#_Toc197346209)

[**1.3.3 FOCUS ON LOCATION-SPECIFIC PREDICTIONS:** 6](#_Toc197346210)

[REAL-TIME UPDATES: 8](#_Toc197346211)

[**1** **RESEARCH AND REQUIRMENT GATHERING:** 11](#_Toc197346212)

[**1.1** **APP AND WEB DEVELOPMENT:** 11](#_Toc197346213)

[Waterfall Model (Traditional Approach) 14](#_Toc197346214)

[Key Features of the Agile Model: 17](#_Toc197346215)

[Phases of the Agile Model: 18](#_Toc197346216)

[Set up the OpenWeatherMap API: 21](#_Toc197346217)

[Include the Google Maps API: 21](#_Toc197346218)

[The application must retrieve and display current temperature and weather conditions.: 21](#_Toc197346219)

[Save Location Settings Locally: 21](#_Toc197346220)

[Create Home, Forecast, and Settings Screens: 22](#_Toc197346221)

[Create Functions for 5-Day Forecast and Current Weather: 22](#_Toc197346222)

[Use Jetpack Compose for Real-Time UI Updates: 22](#_Toc197346223)

[Use Tailwind CSS for Web Styling : 23](#_Toc197346224)

[Deliverable: A Working Weather Wise App: 23](#_Toc197346225)

[Deliverables 25](#_Toc197346226)

[Implementation Steps 26](#_Toc197346227)

[Deliverable 26](#_Toc197346228)

[Deliverable 28](#_Toc197346229)

[**1** **SOFTWARE REQUIREMENTS SPECIFICATIONS (SRS)** 38](#_Toc197346230)

[**1.1** **Stakeholder Overview** 38](#_Toc197346231)

[1.1 Primary Users 38](#_Toc197346232)

[1.2 Development Team 38](#_Toc197346233)

[1.3 Academic Supervisor 38](#_Toc197346234)

[1.4 External Service Providers 38](#_Toc197346235)

[**1.2** **2. DOMAIN CONTEXT:** 39](#_Toc197346236)

[**1. Meteorological Information Systems** 39](#_Toc197346237)

[**2. Mobile Application Development** 39](#_Toc197346238)

[**3. Web-Based User Interface** 39](#_Toc197346239)

[**4. Integration with External APIs** 39](#_Toc197346240)

[**5. Real-Time Data Processing and User Interaction** 40](#_Toc197346241)

[**1.3** **3. FUNCTIONAL REQUIREMENTS:** 40](#_Toc197346242)

[1.1.1 3.1 Live Weather Display 40](#_Toc197346243)

[3.2 Multi-Day Forecast 40](#_Toc197346244)

[3.3 Critical Weather Notifications 40](#_Toc197346245)

[3.4 Support Without Internet 41](#_Toc197346246)

[Users can access weather data using the offline mode after it fetches new updates from the server even when the application detects no internet connection. The app function depends on local storage to preserve critical data for operating normally when networks are unavailable. Users can check essential weather data no matter what their current internet status is because the application maintains key weather information display. This proves beneficial in isolated locations or crisis situations where network connectivity might be reduced. 41](#_Toc197346247)

[3.5 Location-Based Data 41](#_Toc197346248)

[3.6 Simple User Experience 41](#_Toc197346249)

[**1.4** **4. NON-FUNCTIONAL REQUIREMENTS:** 41](#_Toc197346250)

[1.1.1 4.1 Speed and Responsiveness 42](#_Toc197346251)

[1.1.2 4.2 Data Accuracy 42](#_Toc197346252)

[1.1.3 4.3 Ease of Use 42](#_Toc197346253)

[1.1.4 4.4 System Maintainability 42](#_Toc197346254)

[1.1.5 4.5 Future Scalability 42](#_Toc197346255)

[1.1.6 4.6 Secure Data Usage 42](#_Toc197346256)

[1.1.7 4.7 Accessibility and Inclusivity 43](#_Toc197346257)

[Summary: 47](#_Toc197346258)

[Flow: 51](#_Toc197346259)

[Weather Wise – ERD Description 55](#_Toc197346260)

[Chapter 6: Testing 59](#_Toc197346261)

[6.1 Test Scenario 59](#_Toc197346262)

[Users access the application by supplying their credentials before successfully entering the system to see the dashboard. A successful protocol ensures proper authentication together with restricted dashboard entry until valid login occurs. 59](#_Toc197346263)

[6.2 Test Plan 59](#_Toc197346264)

[6.3 Definition of Test Cases 60](#_Toc197346265)

[6.5 Test Case Results 61](#_Toc197346266)

# **Chapter 1: Gathering & Analyzing Information**

# INTRODUCTION:

Humans throughout history have relied on weather to perform diverse tasks because weather conditions simultaneously affect how they organize social events, operate transportation systems and tend their crops. People have depended on weather patterns in ancient times to make vital decisions regarding crop production schedules and their plans for essential gatherings together with travel arrangements. Present-day society requires exact and prompt weather forecast services because increasing urbanization and elaborate infrastructure meet sudden unpredictable climate changes.

The Weather Wise App/ Web addresses the identified challenges using advanced data tools combined with customization features for its users. The app delivers exact real-time weather predictions which are location-specific.

Weather Wise App/ Web employs contemporary technology and meteorological models along with real-time data examination for delivering the most current and applicable weather information to users. This forecasting tool functions as a dependable accurate tool that benefits commuters checking rain conditions and farmers designing irrigation systems while event management avoids weather-based disturbances.

The main target of Weather Wise App/ Web focuses on delivering climate data through an accessible and functional platform which users can easily navigate. The application delivers weather information only as one of its functions. Weather prediction systems nowadays supply general information which lacks accuracy for particular areas or vital choices. Our effort targets the information gap by delivering the most suitable and accurate weather data to consumers. Weather Wise App/ Web serves as an advancement solution for weather forecasting because it utilizes state-of-the-art data analytics and customization features.The app delivers detailed information to help users create better decisions in situations involving climate preparation and weather-sensitive businesses and daily scheduling. The intended project delivers accurate weather data to users who need this information to stay organized and motivated about their plans even when weather conditions change.

## **PROBLEM DESCRIPTION:**

Weather forecasting becomes essential for maintaining safety during daily life and making informed choices because climate predictions are getting harder to forecast. Weather predictions enable people to schedule their activities, prepare for potential risks then reduce destructive impacts of unexpected weather events.

Most weather apps fail to deliver precise and local updates that users need. General forecasts fail to recognize unique conditions which exist in different areas. Users become more prone to unexpected weather shifts since imprecise forecasts increase the risk levels for both travel activities and outdoor engagement along with emergency preparedness.

The system failure becomes more dangerous because severe weather warning systems are nonexistent. People who lack timely warning notifications miss the chance to prepare effectively against weather threats consisting of storm conditions and both heavy rainfall and intense temperature extremes.The platform offers localized weather predictions alongside live statistics in order to revolutionize operational weather forecasting. Users receive accurate relevant information through the method while also benefiting from simple safety-based alerts that maintain awareness and readiness status. Weather forecasting becomes essential for adapting to the future because climate along with weather patterns show increasing unpredictability.

Project Weather Wise pursues two main purposes above basic weather reporting by making climate information easier to understand and more accessible to users. The project seeks to establish a new research standard in weather forecasting while moving from general forecast accuracy towards specific location prediction accuracy which enables people to build their plans with greater certainty.

## **WEATHER WISE GOAL AND OBJECTIVES:**

Weather Wise exists as the main goal to build an application that provides timely accurate safety-oriented weather prediction features. This application has introduced beneficial practices for precise weather forecasting by offering timely updates and trustworthy weather information needed to help users in urban and rural areas..

### Addressing the Needs of Rural and Remote Areas

The platform specifically provides service to populations settled in distant rural locations.

Rural areas experience problems with obtaining timely accurate information through modern application services.

Weather Wise enables distant residents to obtain exact and fresh weather forecasts immediately.

### Real-Time Alerts for Severe Weather Conditions

The system generates immediate weather warnings which include storm alerts accompanying rain alerts as well as snowstorm alerts and hurricane warnings. The combination of inaccurate predictions with delayed alerts makes multiple applications unsafe for public protection..

### Enhancing Safety and Responsiveness

The application promotes user security by delivering real-time data which triggers instant warning notifications. Real forecast accuracy eliminates all gaps from traditional weather prediction methods since it links every predicted point accurately.

### Designed for Both Urban and Rural Users

* Weather predictions through this application deliver precise city-level and remote area-level estimates to every user.
* Sets a new standard for accuracy and reliability in weather prediction.

Multiple widely available weather applications such as AccuWeather, Yahoo Weather and The Weather Channel fail to deliver precise location details that negatively affects their users who live in low-population areas. Modern weather applications supply users in lesser populated locations with inaccurate and slow predictions because they do not account for small-scale regional weather patterns. Weather Wise supplies local weather updates that offer accurate forecasting for rural residents at the same level as urban residents

## **OBJECTIVES:**

**Theory of location demonstrates how exact weather data Weather Wise delivers to specific users. Users achieve precise forecasting because Weather Wise unites meteorological algorithms with real-time data streams to help track weather alterations. Weather updates from the platform provide fast accuracy to users because these updates help them plan daily activities which require temperature reports alongside precipitation forecasts and severe warning alerts.**

**The application provides entire five-day weather forecasts which users can use for securing their activities by seeing upcoming weather predictions. Faultless weather predictions which include temperature reports with precipitation rates and dangerous weather alerts lead users to stay informed about weather fluctuations. Through this feature users can generate sound travel and outback activity plans together with daily schedules because they see current and upcoming weather conditions for potential plan adjustments. Users obtain exact information through the system to ensure their safety and comfort throughout their decision-making process in different environments.**

**Complete meteorological information must integrate wind direction with surface pressure data along with humidity measurement and wind speed evaluation to create superior user relations. Users achieve better atmospheric knowledge with this feature because it delivers everything they need to know about accurate weather measurements. The application helps users create event schedules as they monitor environmental patterns to support better current situation decision-making. Users benefit from the application's advanced capabilities that let them properly handle any kind of weather condition they encounter whether in daily life or during travel or outside events.**

**Users can retrieve necessary weather data even when they lack internet connectivity thus making the app dependable in connection-limited as well as unconnected situations. Users maintain steady access to both their stored weather predictions and temperature histories and severe weather warnings as backup information for emergencies. Real-time data interruptions are significantly addressed in the weather application because rural visitors and travelers together with hikers receive substantial advantages from this feature.**

## **WEATHER WISE: CONCEPT AND VISION**

**Mission Statement of Weather Wise**

The service delivers to users current precise weather information at their present location to provide accurate forecasts for their specific location. The application benefits from current technological models that supply detailed weather information which helps users make important life decisions. User protection receives dedicated attention through Weather Wise because it provides alerts that notify users about all dangerous weather elements such as storms and floods and severe temperature threats. The tool generates specific recommendations and support users by identifying essential steps together with optimal travel periods that help people handle unexpected weather events while reducing potential risks. Precise safety-focused emergency readiness instructions of Weather Wise establish this system as a critical daily tool in addition to serving as an important readiness system for emergencies.

**How Weather Wise Stands Out**

The application achieves its predictions through advanced technology tools which adapt to specific locations. The application offers exact weather information for specific locations through the combination of contemporary forecasting methods with field-based current data sources. Weather Wise delivers personalized reports based on your specific geographical location to provide maximum precision during emergencies better than other smartphone weather applications. Intuitive, user-centric interface for better accessibility

**FOCUS ON LOCATION-SPECIFIC PREDICTIONS:**

Weather Wise incorporates geographical positioning technology to generate specific weather reports that consider users' precise set location. Weather Wise provides point-specific weather forecasting to users by making accurate weather predictions which general forecast services cannot achieve at individual locations.

Weather Wise utilizes this feature to detect weather patterns in particular regions which traditional platforms tend to ignore in remote locations Weather Wise provides users with location-specific real-time announcements that increase security operations while improving operational performance by delivering precise alerts related to their present situation..

The app employs advanced geolocation to display exact weather details which correspond to users' specific locations. Specific locational targeting within the Weather Wise forecast provides exact weather conditions while generic forecasts provide generalized data of unknown accuracy levels. Rural and remote areas experience significant weather advantages thanks to the essential information delivery capability of the platform that other typical weather systems overlook. Through environmental pattern analysis Weather Wise shows users localized information updates that provides them with protection and lets them create advance plans enabling their decision-making processes. Weather Wise achieves precision along with significance in its operations through its dedicated approach to provide local weather data for each designated area during warning alerts and temperature reports plus rainfall predictions.

Weather Wise delivers forecast predictions at a highly localized level in addition to providing real-time updates and weather warning alerts to its users. Through specific area location monitoring the app warns users about storms and heavy rainstorms and extreme temperatures enabling them to take necessary actions. Excellent predictive forecasts from Weather Wise support both recreational and traveling population groups as well as the residents of unreliable climate zones. Weather Wise puts itself apart from standard weather apps by using accurate information combined with specific user information to create a trustworthy weather warning system alongside daily operational assistance..

**SAFETY AS A PRIORITY:**

Weather Wise devotes its main concern to user safety because the platform transmits prompt severe weather warnings and ongoing real-time data that maintains continuous awareness of hazardous conditions. The platform enables immediate response to dangerous weather events including heat waves storm conditions and heavy rainfall because its design allows for it.

Through the software platform users receive adapted genuine alerts for emergencies that help them make proper decisions based on their geographic location. The system maintains its proactive safety mechanism whether the weather conditions change because it protects end users from possible hazards.

Weather established its necessary function in human activities since early times as human beings adjust their professional decisions and personal choices and day-to-day practices because of weather conditions. The active nature of modern global weather precedes society's need for exact real-time weather data to make proper decisions. Large-scale weather predictions cannot deliver accurate details about specific locations which means users stay unaware of rapidly developing severe weather. Weather Wise relies on advanced live location services to provide exact region-specific weather information helping both safety and knowledge transfer for users.

Weather Wise modifies weather data exchange systems by providing exact location-based predictions alongside severe storm alerts in its service offering. Through the application users can execute immediate appropriate actions that lead to well-informed decisions for implementing operational safety protocols. Weather Wise maintains accurate and dependable solutions that serve family outdoor event needs and unpredictable weather requirements of travelers and residents in remote areas.

Users benefit from Weather Wise due to three key geolocation components alongside current weather warnings for severe condition tracking and emergency preparedness features that grant instant control over situations. Advanced design methods integrated with accurate technology make Weather Wise operate as a system that improves user convenience and produces better safety outcomes.

**THE NEED FOR ACCURATE LOACTION-SPECIFIC WEATHER FORECASTING:**

Because of their broad nature standard weather information broadcasts do not provide details about the specific weather adjustments which affect individual smaller regions. Standard weather forecast data lacks precise accuracy required for safety-based decisions as well as routine operation management.

The majority of rural areas and distant locations receive no broadcast weather information because the coverage mainly targets urban regions. Weather-dependent organizations operating outdoors struggle to obtain useful precise data about unspecified geographic regions. Accurate forecast data becomes unavailable during sudden meteorological attacks which puts plan execution and severe weather safety at risk.

Weather Wise uses current satellite technology to deliver customized weather forecasts for every user based exactly on their physical position. Users receive location-specific weather data from the app through its local weather tracking system which does not use regional information..

**KEY ADVANTAGES:**

* Every weather prediction based on localized data incorporates microclimate information to achieve high precision in forecasting.
* BridgeHold provides users with emergency alerts for storm and hot weather conditions and heavy rainfall to help keep them safe.
* The application provides exact weather data coverage to both urban and rural areas in every part of the country.
* Accurate weather data linked to precise locations allows users to plan their activities freely because location-specific weather information stays available to them.

REAL-TIME UPDATES:

The main characteristic of Weather Wise is its immediate information delivery system. Frequent unpredictable weather patterns occur in plants leading to immediate changes in local meteorological situations. Users relying on outdated forecasts face danger to their safety in emergency situations arising from unexpected environmental shifts.

User position tracking together with weather pattern observation ensures the Weather Wise system delivers the latest environmental condition updates to users. The software alerts users instantly about temperature declines and thunderstorm or rainstorm formation so they can quickly modify their plans..

**BENEFITS OF REAL-TIME UPDATES:**

* The system builds an instant alert system for users to hear about upcoming weather patterns so they can react at once.
* Users gain instant updates to alter their plans while choosing suitable shelter between activities by adapting necessary modifications.
* The automatic system alerts keep users in real-time with updated weather conditions to protect them from surprise changes in the weather.
* The app gives users peace of mind through continuous updates that minimize their weather-related uncertainties.

**EXAMPLE SCNERIO:**

The picnic planners obtain essential precipitation predictions which enable them to decide their best event time configuration. The commuter going to work receives information about sudden snowstorm conditions that help them change their travel route for safety. The hiker in remote areas receives timely alerts about upcoming thunderstorms thus enabling them to find protected locations. The Weather-Wise system maintains continuous weather update delivery to all users regardless of speed of condition development.

**SEVERE WEATHER NOTIFICATIONS:**

Users stay protected by launching prompt alerts to set appropriate shelters while they refrain from movement and secure their owned possessions. The optimization of Weather Wise operations helps protect all customer safety. Users get severe weather alerts directly in their notifications which warn them about approaching dangerous conditions and live weather status lets them stay informed at all times. The system allows users to take immediate action for protecting themselves as well as their family members. Extreme meteorological events including stormy weather systems with heatwaves and hurricanes and uncontrolled rainfall create multiple safety risks for both lives and properties as well as daily operational routines. As a user of Weather Wise you get important warnings for your location so you can make safety preparations before threatening situations become dangerous.

**Severe weather alerts serve three essential purposes for everyone:**

* Ready populations can buy necessary supplies at once but additionally they maintain the capability to adapt plans and start evacuation movements.
* Users gain protection from dangerous situations when warning alerts enable them to avoid high-risk weather areas.
* The prevention-based weather notification system lets organizations and people create preparedness tactics for severe conditions to reduce disruptions..

**PRACTICAL APPLICATIONS OF SEVERE WEATHER ALERTS:**

Heatwave alerts through this system tell individuals to remain inside their homes while they should drink fluids and look after elderly persons alongside children. The system provides severe storm and thunderstorm warnings that explain sheltering procedures and safe road practices and suitable protection methods for outdoor assets. The system informs users within flood areas about necessary preparations to evacuate during heavy rainfall and protective instructions for flood situations. The system provides icy road alerts and blizzard warnings that help drivers and commuters to make either secure travel decisions or reschedule their trips respectively. The severe weather notifications in Weather Wise allow users to take vital safety decisions during dangerous situations thus safeguarding both personnel and lives.

**The system enables users to obtain data needed for making knowledgeable decisions:**

Users gain access to more than weather predictions through Weather Wise to make wise decisions about their different life domains. Weather Wise delivers exact geographically specific real-time weather reports that enable users to schedule modifications and make better plans and emergency preparations confidently..

**EXAMPLES OF INFORMED DECISION-MAKING WITH WEATHER WISE:**

Through the application users can reschedule outdoor activities from weddings to sports events through weather predictions made accessible to them. Weather Wise provides commuters with precise alerts that help them avoid unsafe weather conditions when driving through roads with storms and heavy rain or ice formation. The application supports both family groups and individual members through alert-driven purchasing recommendations for emergency supplies and protective measures and safety instructions. Clients use exact weather forecast data from Weather Wise to schedule their outfits and travel plans for daily decisions. Weather Wise provides predefined informed actions to its users leading to decreased interruptions coupled with enhanced safety practices and optimal daily activity effectiveness

**ENHANCING SAFTEY IN RURAL AND REMOTE AREAS:**

Traditional weather-related applications prioritize their main features to major cities while neglecting rural and remote locations. People who reside in under-served areas must navigate limited access to weather information even though they need this information to ensure personal security and develop effective plans. Weather Wise enables all people to obtain weather updates for rural and remote destinations including isolated areas. The targeted service benefits all users throughout the network since Weather Wise distributes real-time accurate weather reports without limitation by geographical positions.

**KEY BENEFITS FOR RURALAND REMOTE USERS:**

The area-specific prediction services in WeatherWise offer better forecasting precision for local rural areas beyond regular weather prediction systems at other platforms. WeatherWise offers forecasts based on precise locations to provide reliable planting solutions and protective guidance for livestock and emergency storm alerts tailored to farmers, ranchers and small towns residents at every field and residential area. WeatherWise activates instant alert notifications immediately when weather conditions shift allowing protection of farmers and hikers as well as outdoor workers from sudden storms and flash floods and high-risk temperatures. The vital warning system through WeatherWise enables remote field workers as well as those on mountain trails to receive warnings about dangerous conditions and those traveling through rural areas. The system helps rural dwellers and outdoor workers and farmers to create improved choices about farming operations and path scheduling and protective measures for safety. Weather Wise provides essential weather data for rural and remote groups impartially so those people can stay aware of sudden weather hazards safely.

**WEATHER WISE: A TOOL FOR SAFTEY AND EFFICIENCY:**

Modern people who perform activities in their daily lives need clear weather information that displays real-time details and specific locations while unpredictable and rapid weather patterns develop. Weather Wise brings users forecasts and safeguards their safety and efficiency through advanced technology that enhances planning capabilities.

**DATA PRIVACY:**

Operations at Weather Wise center their core activities on customer data privacy which enables the application to deliver personalized alerts by accessing user locations together with adjusted settings. The Weather Wise team implements encryption for data security requirements during server-application data transfer operations. Weather Wise users hold complete oversight of their data because the application provides them with full control to adjust location tracking permissions and notification settings. User confidence at Weather Wise builds because they safeguard their data before all international privacy requirements. The application employs a privacy-first approach because this approach functions as its key privacy protection mechanism for sensitive data. The application depends on location data to generate specific location-based predictions because it implements all required safeguards to protect user data. Data protection throughout the storage process and transmission stages happens system-wide encryption to ensure unauthorized users cannot access data and sustain security. While fulfilling privacy rules protects user data security the company manages to provide high-quality location-based services to users.

**METHODOLOGY:**

Weather Wise development works together with research testing and the systematic process used to create the software. Real-time accurate location-based weather forecasting requires accuracy as well as intuitiveness and high reliability for the app..

# **RESEARCH AND REQUIRMENT GATHERING:**

* **Investigation and Needs Collecting Weather Research Data Sources:**

Research both Google Maps API and Open Weather Map API for finding the most suitable sources which provide trusted and timely weather data.

* **Examine User Preferences:**

The essential requirements for potential users need exact clarification especially for rural and remote users where offline functionality and real-time alerts and local precise information represent their requirements.

* **Analyze Competitors:**

An analysis of Yahoo Weather, AccuWeather and The Weather Channel apps demonstrates their drawbacks as well as possible enhancements.

Weather Wise’s development team concluded the definition of clear functional and non-functional requirements for their platform..

* **The development team must finalize all essential requirements including functionality together with non-functionality for Weather Wise.:**

Architecture development should proceed through comprehensive research which emphasizes the incorporation of modern weather information services including current data monitoring and precise five-day prediction data. The system requires implementation of advanced meteorological factors including surface pressure alongside wind speed and humidity levels to deliver accurate full weather information. Making the platform more usable requires an offline enablement feature so users can access previously downloaded weather information without needing internet connection. A strong user-friendly weather monitoring system emerges from the combination of real-time accuracy with reliable forecasts and offline functionality..

## **APP AND WEB DEVELOPMENT:**

**Frontend Development:**

* **UI for Mobile Apps:**

Kotlin together with Jetpack Compose allowed the development of WeatherWise as a contemporary Android app which delivers efficient performance through its modern and responsive design. The declarative user interface adopts a minimalist design to deliver instant weather data access to users. Through Framer Motion implementations users experience better interaction quality because of the added animations which deliver a more user-friendly interface..

* **The Web Interface:**

The web-based application utilizes JavaScript libraries with React, React DOM and React Scripts for its development. The user interface components of the application employ React as its basic framework to provide components that boost maintainability and expandability. React DOM provided the connection between React components and real DOM instances which enabled fluid element presentation on browser screens. The use of React Scripts enabled developers to set up and manage projects efficiently since it provided a standardized build configuration which eliminated the need to deal with complex tooling. The combined use of these technologies provided a well-functioning interface which delivered both efficiency and a great user experience..

**Backend Development:**

* **Integration of Weather APIs in Backend Development:**

The OpenWeatherMap API must be integrated into the system because it allows users to access current weather conditions and extended forecasts alongside environmental data and real-time statistics. The application gets precise and real-time weather data according to user location through this API. Both the Google Maps API must be activated to get specific geolocation services. The implementation of Google Maps enables the application to precisely locate users or let them choose locations which results in relevant weather data retrieval for their specific geographic area. These APIs create a robust base that enables the development of an application which automatically adjusts to location changes..

* **Managing Data and Local Storage:**

The application should implement Room Database for maintaining local storage of user preferences together with weather data cache to offer offline functionality. Android offers Room as its SQLite abstraction library which provides an easy method for local data storage while preserving type safety and delivers optimized data access. The application keeps functioning properly without internet access because it stores important data including weather information and user settings in the app cache. The application's design yields improved user interaction because users can access vital weather data during loss of internet connectivity while their individual settings remain stored between different device sessions..

* **Web-Based Server Interaction:**

Effective client-side routing in the application requires the implementation of React Router DOM to enable smooth navigation between different views without full page reloads. The user experience improves when users get faster transitions because of this implementation which creates an app-like interface. Luxon operates as an effective dating-time management system which delivers present-day time zone-aware capabilities to maintain clear time and date operations throughout the application especially within weather forecast and historical data sections. The interface requires Tailwind CSS for obtaining a modern and clean design along with response capabilities. Through its utility-first model Tailwind allows developers to design custom CSS styles during markup development which accelerates rapid building of visual elements. This technology delivers the perfect solution for developing adaptable interfaces that provide smooth performance on various devices of different sizes. Front-end architecture benefits from these tools due to their collaborative capability in creating a modern architecture that remains simple to maintain..

**TESTING AND QUALITY ASSURANCE:**

* **Testing for Functionality:**

The system requires testing to verify both essential features which need real-time weather accuracy and 5-day forecast display correctness. All triggering functionality along with appropriate display of severe weather alerts should function correctly whenever requirements apply. The offline mode must be tested to check both data caching functionality and Room Database storage of user preferences. All scenarios receive a stable reliable user-friendly experience because of extensive testing procedures..

* **Testing for Usability:**

The application needs to provide an enjoyable experience through its accessibility and user-friendliness for every user. Real users should perform beta tests to provide meaningful feedback about the application's UI/UX which will guide improvements. Test results should guide the interface development process to make it more usable while resolving encountered problems. The application development process directed towards users leads to creating a system which offers both simplicity and aesthetic attractiveness while fulfilling user expectations in the outcome..

* **Evaluation of Performance:**

All API calls must be monitored for their response times to verify that performance limits remain intact while investigators seek out potential delays. The app should undergo tests in remote locations as well as low-speed networks to validate its dependable operation in restricted conditions. Users should optimize data handling protocols and create caching strategies to achieve quicker load times together with enhanced system responsiveness. The implemented measures work together to maintain effective operation and reliability throughout networks with different qualities..

* **Testing for Security and Privacy:**

Secure protection of user location data becomes possible through encryption together with limited access standards to safeguard sensitive information. Users must receive transparent notification requests which let them determine their location access within the app while practices confirm compliance with privacy regulations. Developing apps with strong data protection and compliance measures will keep customers trusting the system..

**DEVELOPMENT TOOLS AND TECHNOLOGIES:**

|  |  |
| --- | --- |
| **AREA** | **TOOL/TECHNOLOGIES** |
| APP Development | Kotlin+ Jetpack Compose |
| API Integration | Open Weather Map API, Google Maps API |
| WEB Development | React.js, Tailwind CSS, React Router DOM |
| Local Storage | Room Database |
| Styling and Animations | Tailwind CSS, Framer Motion |
| Data/Time Handling | Luxon |
| Performance | Web Vitals(for Web) |

Weather Wise implements a complete approach which positions it as a sophisticated localized real-time forecasting system above basic weather applications. Weather Wise focuses on fixing existing weather service defects through accurate prediction systems with real-time updates coupled with site-specific alerts. The lack of dependable weather information presents a problem for people living in rural areas and underserved regions since many existing platforms give insufficient up-to-date details that their daily activities and agricultural operations and personal safety require. The primary purpose behind Weather Wise development was to develop an application that empowers users to base critical decisions on precise local weather data through detailed accurate reporting.

**AVAILABLE METHODOLOGIES:**

Weather Wise requires the implementation of different acknowledged software development approaches. The development follows these methodologies due to project requirements which include a mobile app and real time weather data collection and user feedback implementation.

### ****Waterfall Model**** (Traditional Approach)

The software development using the Waterfall Model requires linear execution because developers need to finish a stage before starting on the next step. The Waterfall Model represents one of the pioneer methods that software developers used during the early days because it offers a systematic and easy-to-understand process:

1. The requirement analysis phase should collect all application demands including functionality linked to user position tracking and storing user details safely. The definition would establish how data from users is managed by determining encryption protocols and user permissions regarding location access**.**.
2. During systematic design the architects create the application structure. You would establish the encryption process for location data storage as well as permission-based feature integration to request user authorization..
3. The actual programming of the app takes place at this development phase. Your implementation includes building functions to monitor location data and providing secure user data encryption together with privacy setting options for users. Constructing all essential permission aspects takes place within this development stage..
4. This phase focuses on executing complete testing procedures of the application. Testing of the app occurs to guarantee proper functionality regarding security measures and location data management and user privileges. The testing procedure should verify GDPR and CCPA compliance to guarantee proper data protection along with full compliance with privacy legislation..
5. When the app reaches completion point the deployment team moves it to its proper hosting such as Android users access it through Google Play Store. Users can begin installing the app once they get to this point while location services become available through the permissions they authorized..
6. The application requires permanent maintenance procedures after its initial release. The app requires maintenance through updates and security vulnerability fixes as well as location service adjustments and user permission changes to fulfill privacy regulations and user input..

The Waterfall Model requires completing every stage in full before advancing to next steps thus giving structure to development but creating constraints after phases are finished. The development process might become challenging if developers need to implement changes after operation commences.

The Waterfall Model works optimally for projects which follow the criteria established by the mentioned criteria. The characteristics of this project match the methods of the Waterfall approach in the following manner:

* Project with precise, unchanging requirements:

The Waterfall model succeeds best in development situations where project specifications are set in advance and will not change during development stages. A linear sequential workflow makes it difficult to implement requirements modifications that occur after project commencement because changes become financially demanding to implement..

* Uncomplicated timetable with few modifications:

Waterfall succeeds in managing projects where developers know all aspects of timeline and scope in advance. The model functions best for projects requiring stable development timelines because it blocks frequent alterations during the process..

* Projects with predetermined, unambiguous requirements:

Success in this model depends on obtaining detailed understanding of requirements before project commencement. A project benefiting from this methodology moves forward consistently because all requirements remain crystal clear throughout different phases..

* The model provides excellent results for simple and small projects with basic requirements:

The Waterfall methodology operates best for projects of limited scope and low complexity that possess defined requirements since the beginning. The application might face challenges when used on big complex endeavors because such projects demand repeated feedback loops with continuous change implementation..

* These development projects require minimal client interaction conflicts with the start and finish of development activities:

According to its fundamental principles the Waterfall model provides limited client interaction after the initial requirements creation phase. The client sets the initial product specifications before receiving the completed development product at the end while minimal exchange occurs between developer and client during the development period.

The Waterfall Model provides the best solution for basic projects that feature uncomplicated requirements together with definite time constraints and restricted client involvement in development stages. The project needs Agile development if it has potential modifications or needs continuous user feedback throughout development..

**Restrictions:**

* The alteration of customer needs during a mobile app project often proves difficult to modify throughout its development process.
* All stages of development finish before testers conduct their work.
* Long and complex and creative work tasks cannot benefit from this methodology.
* Brief corrections needed after project completion become expensive to resolve.
* The system lacks ability to handle market demand changes and user requirements adjustments..

1. **Agile Model:**

Sprint-based development through multiple cycles makes up the Agile Model which provides flexible iterative software development methods to deliver betterment in small increments. Agile provides ongoing participant feedback together with teamwork and adaptability through entire project development time. The Agile Model follows this comprehensive pattern which I will explain next.

The Agile Model consists of several main characteristics which guide its operations.:

### Key Features of the Agile Model:

1. **Iterative Development**:
   * Agile transforms programming into multiple short periods named "sprints" which normally last between 2 to 4 weeks. A product that may be ready for delivery is generated by the end of each sprint so team members can evaluate it before starting the next development cycle..
2. **Continuous Feedback**:
   * Throughout project execution the model enables consistent feedback which comes directly from clients and stakeholders as well as the team members. Constant feedback from multiple sources helps developers shape the project through such changes as evolving requirements or customer needs or newly obtained information..
3. **Collaboration**:
   * Agile promotes teamwork between different groups of developers testers and designers as well as between the development team and the client organization. The project stays on track with client developments through this practice..
4. **Flexibility and Adaptability**:
   * Agile projects have the essential benefit that they adjust smoothly to changing circumstances. The development team has the capability to adjust the project structure seamlessly when new requirements or issues appear during the development cycle..
5. **Customer Involvement**:
   * The development process with agile projects needs continuous participation from clients. Occasional meetings between clients and team members involve sprint reviews along with retrospectives and planning activities. The product stays track and meets expectations through continuous client feedback which leads to fast adjustments by the team..
6. **Incremental and Continuous Delivery**:
   * Agile delivers discrete features to users before completing the full product just as Waterfall does. The system enables users to start working and providing feedback on product sections at an earlier stage..

### Phases of the Agile Model:

1. **Concept/Inception**:
   * When the project starts it requires a definition of total scope and a set of objectives together with stated requirements. Agile amounts to a different approach than Waterfall because it acknowledges that specifications change throughout the process..
2. **Iteration/Increment Planning**:
   * The team defines the content of the initial sprint by selecting components for building new features together with functionality. A new iteration continues to develop from existing work while maintaining concrete outcome requirements at every development cycle..
3. **Design and Development**:
   * The team conducts feature development and programming tasks for the planned items within the sprint documents. The development process runs through brief workable segments and tests are done continuously throughout..
4. **Testing**:
   * Testing activities happen at every stage of development since each sprint includes various levels of assessments including unit tests in addition to integration tests. The testing process which keeps running throughout development results in high-quality code and functionality at every stage..
5. **Deployment/Release**:
   * Theteam produces a deployable or releasable product version during each sprint endpoint. Staging environments or production environments receive the deployment according to project specifications..
6. **Review and Retrospective**:
   * The team gathers feedback through reviews following every sprint duration. The team conducts two meetings to analyze progress retrospective data for future enhancement while exploring improvements that should be implemented during the next sprint process..
7. **Repeat**:
   * Each cycle of the process builds upon itself by making progressive improvements to the product while using client input and test outcomes..

**CHOOSEN MEDTHODOLOGY:**

Agile methodology serves as the chosen method for developing the Weather Wise application. If you consider Agile for this project it turns out to be the most suitable choice since constant user engagement and flexible product updates and real-time modifications are needed based on weather data alongside customer demands. The main functions of the Weather Wise app include producing live emergency weather alerts together with detailed local weather predictions as well as providing an instant user-friendly interface. Agile development characteristics of being user-driven with flexibility and effectiveness fit perfectly into the development requirements. .

**SELECTION OF AGILE METHODOLOGY REQUIRES CONSIDERING THE FOLLOWING ESSENTIAL FACTORS:**

* **Iterative Development:**

Minuscule manageable sprints in iterative development make it possible for the project team to deliver multiple features incrementally including real-time updates, 5-day forecasting, offline access and severe weather alarms..

* **Constant User Input:**

Beta testing and early feedback loops enable genuine users to improve the design along with functionality before final product release..

* **Adaptability and Flexibility:**

Agile supports fast alteration processes that do not affect overall projects regardless of the new requirements appearing (such as offline mode enhancement).

* **Put Quality First:**

Program dependability together with user experience improves through both defect identification and correction that happens during regular testing sessions following each sprint..

* **Team Collaboration:**

Agile practices support project advancement through better communication between software developers and testing teams together with supervision teams.

**AGILE SPRINT PLAN FOR WEATHER WISE:**

Weather Wise development will proceed through sprints lasting two to three weeks per sprint. The main target of each two to three week sprint is to deliver specific features that resulted from user feedback and testing procedures.

**Sprint 1: Requirement Gathering:**

1. Instruction: Construct Requirements to generate functional and non-functional specifications that include location tracking along with weather data and performance demands.
2. The required data and features will be investigated through expert research of OpenWeather Map and Google Maps APIs.
3. The team will use Jetpack Compose to make UI prototypes that represent main and secondary screen designs.
4. The process starts with simple sketches of wireframes that show basic layouts as well as system pathways.
5. Show your supervisor the developed wireframes and requirements documents for their feedback so that needed changes can be made.
6. The team concludes the project by finishing the accepted prototype together with the requirements document.
7. The structural organization enables project refinement through iterative cycles because it lets you adapt based on received feedback..

This approach keeps the project focused and iterative, allowing you to adjust based on feedback.

**Sprint 2: Integration of Location Services:**

### ****Set up the OpenWeatherMap API****:

* The initial step involves account sign-up at OpenWeatherMap then acquiring an API key for weather data access.
* The system retrieves current weather information from the API using both user position coordinates (latitude and longitude).
* Extracted vital weather statistics should be presented to users such as temperature readings together with weather descriptions ("clear sky") and humidity levels.

### ****Include the Google Maps API****:

* **Users need to acquire Google Maps API Key by registering for the Google Maps API to enable mapping capabilities.**
* **The application includes a display map during which users may view their present location through the integrated Google Maps system. The app uses the Google Maps technology to obtain and display current GPS location for each user.**
* **The application should enable location tracking which displays real-time placement of users on the displayed map.**.

### ****The application must retrieve and display current temperature and weather conditions.****:

* **After detecting user position using Google Maps the application should transmit latitude and longitude data to the OpenWeatherMap API.**
* **The app will show both the weather temperature readings alongside the current weather condition and other required weather data on screen.**.

### ****Save Location Settings Locally****:

* The application saves user geographic coordinates at their previous known position through the SharedPreferences local storage solution.
* When the program restarts it will retrieve the stored location coordinates from local storage to restore weather data along with showing the most recent map view.

To achieve this functionality while combining OpenWeatherMap API (weather) with Google Maps API (location) developers must enable storage of user-selected settings for persistent locations. The app maintains location awareness by using this feature to supply weather reports from the user's current location while storing both location points between app sessions..

**Sprint 3: App Development Frontend:**

### ****Create Home, Forecast, and Settings Screens****:

The application shows three main interfaces that simplify the user experience for weather information. From the Home Screen users obtain essential weather information including temperature readouts together with weather condition reports and humidity levels and symbolic representations. Users can find a navigation element to view more advanced predictions on this screen. The Forecast Screen provides users with daily weather forecasts for the next five days and displays maximum and minimum temperatures with the weather status and matching weather symbols for each day. The interface enables simple viewing through user gestures of day-by-day scrolling or swiping. The Settings section lets users improve their interface by selecting the location either through place search or GPS position and selecting temperature units and managing notifications for weather reports..

### ****Create Functions for 5-Day Forecast and Current Weather****:

Real-time weather updates are delivered through the app by using user location data to acquire their present weather data. Users get access to three core weather items including the current temperature readings and weather type and air humidity measurements. The application offers both real-time updates as well as predictions for the upcoming five days. Users receive a complete weather outlook for the following days because each forecast shows high and low temperature ranges along with a simplified weather description and symbolic representation..

### ****Use Jetpack Compose for Real-Time UI Updates****:

* **Jetpack Compose should enable the creation of an up-to-date user interface that remains flexible and responsive for dynamic data updates with smooth user experiences. The application must implement real-time state management because it allows the user interface to update automatically whenever new weather data becomes available including location changes and Celsius and Fahrenheit swaps. The app must include smooth transitions that link the Home screen with the Forecast page and the Settings webpage to ensure users experience a smooth fluid navigation system..**

### ****Use Tailwind CSS for Web Styling**** :

Tailwind CSS acts as a powerful tool to help developers create web versions of their app by making style application quick and efficient. Tailwind CSS applies its utility-first design system to make button and grid element design as well as card design easier to handle without requiring custom built CSS. The application presents responsive design capabilities natively which ensures that all screen sizes from mobile phones through to desktop monitors produce functional interfaces. Developers who implement Tailwind CSS UI classes for buttons and text as well as containers and grids achieve design process improvement while maintaining design consistency.

### ****Deliverable: A Working Weather Wise App****:

Users should have access to both real-time weather information and five-day forecasts which can be viewed from the Home and Forecast screens of the application. Through the Settings screen users gain control of their experience by managing options that involve location selection and temperature scale choice. A user-friendly interaction between the Home screen and Forecast screen and Settings screen needs to function flawlessly for delivering a smooth experience to users. The app should maintain real-time communication with its backend system to obtain and show continually updated weather details to users.

The specified plan enables users to obtain a meaningful weather application experience that shows real-time weather information updates. The application will feature core functionalities that include showing present weather along with five-day predictions along with user preference management options. Container design and mobile-friendly functionality of the app will support responsive usage across multiple platforms from field devices to web-based systems..

**Sprint 4: Enhanced Functionalities and Offline Support:**

#### **Detailed Weather Information**

The application must present expanded weather information by adding humidity numbers together with wind indicator and wind speed and air pressure measures. The application should present this information using a readable interface which appears simultaneously on the Home Screen and Forecast Screen. The user benefits from comprehensive weather display through this feature which boosts the application's value..

#### **Room Database for Offline Storage**

An implementation of the Room Database will serve to locally store weather data. The app will use this database to retrieve weather information which it can show when internet access is unavailable. Your application will store fundamental data that encompasses present weather conditions combined with prediction data and user-preference settings which contains their current location along with temperature unit selection. Room ensures storage of weather information which allows users to access this data offline during times of internet disconnection..

#### **Caching for Current Weather Data**

The current weather data should be cached to boost system performance along with decreasing network traffic. The application will store local device data with an expiration period of one hour. The application avoids repeated requests from the server by activating data cache functions. The application retrieves fresh data from the OpenWeatherMap API while the user is offline or the data reaches its expiration period to update the cache and automatically refreshes the user interface..

#### **Offline Mode**

Now have a matching offline mode which stands as an integral part of your application program interface. Under offline conditions the application switches to offline mode and retrieves data either from the Room Database or existing cache. Heretypost users can obtain weather information even when offline because of the application's automatic fallback mechanisms. Users will receive alerts about being disconnected while actively viewing current weather conditions obtained from the last successful connection..

### ****Deliverables****

**An implementation of the Room Database will serve to locally store weather data. The app will use this database to retrieve weather information which it can show when internet access is unavailable. Your application will store fundamental data that encompasses present weather conditions combined with prediction data and user-preference settings which contains their current location along with temperature unit selection. Room ensures storage of weather information which allows users to access this data offline during times of internet disconnection..**

Users will obtain precise and detailed weather information through an application which operates efficiently without internet connections.

**Sprint 5: Real Time Severe Weather Alerts:**

#### **Severe Weather Warning System**

Users should receive warnings about severe weather through a notification system which delivers caution alerts about dangerous weather occurrences. The system needs to activate alerts through current weather information while delivering complete warning notices.

#### **Push Alerts**

The application should utilize push notifications for sending weather notifications to users. Users will receive immediate real-time updates from the app through push notifications regardless of their application status. The application requires integration of Firebase Cloud Messaging (FCM) or comparable push notification service.

#### **Alert Categories**

Include the following **alert categories**:

**Toward public safety the system will release three different alert types: Heatwave along with Heavy Rain and Storm. Users will get alerts for heatwave situations when temperatures reach higher than 40°C through a Heatwave Warning that says "Heatwave Warning: Temperatures are expected to exceed 40°C." Hydrate yourself at all times while also staying indoors during extended outdoor periods. Heavy Rain Warning systems will activate if meteorologic forecasts indicate more than 50mm rainfall in the upcoming 24 hours before predicting possible flood occurrences: "Expect over 50mm rainfall in the next 24 hours."Flooding may occur. Flooding is possible." Users receive a Storm Alert that warns about severe storms with 100-kilometer-per-hour winds through real-time weather information. Take cover immediately."**

#### **Real-Time Weather Alerts**

The application needs to retrieve its weather data from trusted platforms including OpenWeatherMap and AccuWeather. The software must check weather information regularly to deliver instant push warnings whenever detected heatwave or storm or heavy rainfall conditions match set criteria..

### ****Implementation Steps****

The system needs real-time weather data acquisition through a weather API to activate notifications based on temperature thresholds for heatwaves as well as storm alerts from the API. Users can manage their preferences by grouping push notifications into heatwave, heavy rain and storm channels through the integration of Firebase Cloud Messaging (FCM). Users need an option to customize their alerts through the app settings by defining notification hours during daytime or when they are on the move. A notification system will deliver specific guidance instructions to users enabling proper responses to severe weather conditions...

### ****Deliverable****

The produced solution enables real-time push notifications through a functional weather alert system that operates with weather condition-dependent alerts. The system immediately warns users about severe weather incidents starting from heatwaves extending to heavy rain and storms thus providing protective information.

**Sprint 6: Quality Assurance and Testing:**

#### **Performance, Usability, and Functional Testing**

The Weather Wise app needs complete testing which assesses its functional capabilities and performance quality. The performance assessment of the Weather Wise App should analyze its speed and interactive capabilities most significantly impacting data loading and interface operations while checking performance on different mobile devices including entry-level devices. The interface needs usability testing because real users must assess how easy it is to use along with testing the interface for its natural operation to evaluate user experience and weather detail clarity while testing navigation. Testing must validate the complete functionality by assessing both weather information precision and alert notiﬁcation timing and Room database performance in addition to data caching capabilities.

#### **Beta Testing with Limited Users**

The beta application release will happen to limited testers who must examine real-world functionality across diverse network connections and GPS precision with severe weather alert functionality. Gather information about usability problems and bugs through this phase by asking users to report all performance and functional and user interface matters. The necessary feedback will help discover problems that require resolution before the widespread launch of the application.

#### **Bug Fixes, API Optimization, and UI Enhancements**

The main goal of post-beta testing should be to fix problems that cause program failures and service connection errors especially with weather API functionality to improve application stability. API call performance optimization requires attention because it improves data retrieval speed which benefits 5-day forecast and severe weather alert features particularly. User interface enhancements should be implemented for providing responsive and visually comparable interfaces throughout multiple screen sizes and devices while design improvements will occur through user feedback to create a smoother system.

#### **Security Testing**

Testing all security protocols must happen to verify the app's capability to defend user privacy and sensitive data. The security tests will confirm how data encryption performs for storage under Room Database and network communication through API calls to ensure complete protection of sensitive user location information. Compatible permission requests should undergo evaluation through testing which demonstrates proper consent protocols for accessing data privacy information. A vulnerability scan using security tools should be conducted to detect SQL injection threats and data leak vulnerabilities as well as verify the app's authorization resilience through industry-standard security guidelines.

#### **Stable and Bug-Free Version for Release**

A final regression test should be performed to confirm the app's stability along with its functionality after all previous steps achieve completion. This test will verify the preservation of established functionality while recent updates are in effect. The process of preparing for app stores starts by collecting all needed materials such as accurate descriptions and high-quality screenshots together with platform-specific privacy policies for Google Play Store submissions.

### ****Deliverable****

The perfect version of the Weather Wise app features complete optimization with secure testing for stability without bugs. Performance alongside usability and security features have been guaranteed for the app which is now prepared for its launch to users.

**JUSTIFICATION FOR THE SELECTED APPROACH:**

Developing mobile applications requires AngularJS because it offers the best features for application development that depend on real-time updates and continuous improvements through user feedback. Weather Wise along with its competitors in the weather application field must adapt their approach due to user preference changes and technical progress in order to deliver real-time hyper-local vital alerts. The project demands adaptability beyond what traditional linear development methods presented by Waterfall can deliver. The project works to enhance community preparedness together with resilience and user satisfaction through systematic analysis of rural area user requirements because these locations bear the greatest risk from weather unpredictability. Weather Wise stands out as the superior choice in intense market competition because it supports safer communities that also experience increased education levels.

|  |  |  |
| --- | --- | --- |
| **FEATURE** | **AGILE MODEL** | **WATERFALL MODEL** |
| **Approach** | Iterative and Incremental | Linear and Sequential |
| **Flexibility** | Highly Flexible | Hard to modify after beginning. |
| **Participation of User** | Constant feedback along the process. | Mostly at the start and finish. |
| **Rate of Development** | Quicker supply of functional features. | The Complete system is only supplied at the end. |
| **Testing** | Continuous testing follows each sprint. | Only once development is finished does testing take place. |
| **Controlling Risk** | Early risk detection. | Risky if problem are discovered after the fact. |
| **Adaptability** | New criteria can be easily added while working. | Difficult to modify. |
| **Ideal for** | Dynamic projects such as real-time systems. | Static and have set specifications. |
| **Example** | Weather Wise App that is driven by user feedback in real time. | Manufacturing projects and extensive government systems. |

**COMPARISON OF AGILE VS WATERFALL METHODOLOGY FOR WEATHER WISE:**

**DEFINITIONS, ACRONYMS, AND ABBREVIATIONS:**

|  |  |
| --- | --- |
| **TERM** | **DEFINITION** |
| **UI** | User Interface (The area where users interact with the visual components of the application is known as the user interface). |
| **UX** | User experience is the sum of a user's interactions with the application. |
| **GPS** | The Global Positioning System, or GPS is a system that determines a user’s current location. |
| **JSON** | A popular lightweight data format for data transmission between servers and apps is JSON, or JavaScript object Notation. |
| **Kotlin** | A contemporary programming language for creating apps on android. |
| **Jetpack Compose** | A contemporary framework for declaratively creating native android user interfaces. |
| **Room Database** | Local data is stored in Android apps using a SQlite object-mapping library. |
| **Tailwind CSS** | Tailwind CSS is a framework that prioritizes functionality for quick site design and styling. |
| **Framer Motion** | A Web Application animation and transition library. |
| **React** | An interface-building JavaScript library, mostly for online applications. |
| **Offline Mode** | The Function that permits access to weather data that has already been retrieved even in the absence of internet connectivity. |

**LITERATURE REVIEW:**

Historians have maintained that high mountain glaciers seemed to be absent when worldwide climate conditions were mild in various regions. Evidence shows that fruit trees together with vineyards and paved roads and towns and even cities inhabited elevations believed unfit for such development because the climate conditions were ideal during this period. Since global temperatures have modified while simultaneously impacting farm production and settlement patterns this image of previous climate warmth stimulates serious thought. This transformation in medicine and public health history during the eighteenth century and later North America and Europe merits careful study to understand its evolution. During many centuries humans have perfected their forecasting abilities for weather patterns in parallel with their medical expertise. People had a direct understanding of how weather affected public health outcomes because they could easily observe its influence on agriculture and infections as well as living conditions. The growing prominence of scientific ideas during the Enlightenment period caused meteorology to change and lead to modern complex weather forecasting algorithms. (Weather wise: Speaking folklore to science in Leukerbad., 2021)

The cause of many aviation fatalities across general aviation operations in the USA along with other countries stem from poor pilot decision-making and ill-advised responses to weather conditions. Pilots encounter unpredictable changes in air conditions that result in significant flight safety hazards and therefore constitute a severe concern for the aviation field. aviation safety training aims primarily to build pilots' skills for recognizing and correctly responding to worsening weather conditions. A special computer-based training platform underwent evaluation for its purpose of supplying visual pilots with abilities to detect and respond properly to aerial warning signals during flight. Before operational risks emerge pilots received training to develop their ability to recognize and respond to hazardous weather conditions which aimed at minimizing weather-related fatalities and accidents resulting from poor weather decisions. Computers improve weather-related decision quality when used for pilot training according to research findings. Such programs could reduce weather-related aviation accidents combined with their associated fatalities because pilots learn to recognize weather patterns change and acquire proper response methods. (Weatherwise: Evaluation of a Cue-Based Training Approach for the Recognition of Deteriorating Weather Conditions during Flight. , 2003)

Love of agriculture led human civilization to become the first occupation through farming domesticated food plants. Farming represents both practice and scientific discipline used for crop cultivation and animal rearing with the aim of obtaining food and other materials alongside spiritual values for transforming parts of Earth's surface. The national welfare depends significantly on sustainable agriculture making it generate leading innovative technological developments throughout recent history. Present-day scientists collaborate with farmers to develop sustainable cultivation systems which minimize water usage while reducing environmentally damaging practices during production. Deep learning joins machine learning as one of the technological approaches used in these developments. Climate plays the most important role in determining how crops develop towards maturity. Local climate elements such as temperatures together with humidity and rainfall and wind speeds stand as the primary contributors in determining agricultural production outcomes. Before planting crops the farmer gains profit by knowing the weather forecast in advance. Machine learning represents a modern development which solves daily human problems. Through this method computers emulate human skills to understand experiences and different forms of data. Agriculture is an active field of machine learning because diverse machine learning algorithms forecast crop production based on meteorological patterns. Today industry sector agriculture stands as one that depends on machine learning applications. Different machine learning algorithms allow farmers to estimate agricultural yields through climate data analysis which leads to elevated crop yields. The studies will use LSTM model to make crop yield predictions through modeled meteorological data. The research paper "Machine learning based crop prediction on region wise weather data." was published in 2023.

Early attempts at making numerical weather predictions required a substantial number of workers since manual practices and limited technology were common at that time. Modern weather prediction has reintroduced previous modeling principles because rapid advancements in computer technology occurred. Complex and precise forecasting methods have become possible through the use of powerful computers leading to an advancement of meteorological prediction from infancy to modern times. The current weather forecasting models rely on fundamental meteorological concepts to create their predictive equations. The fundamental equations used for weather prediction serve both as a mathematical system to predict weather changes and as the controlling factors for atmospheric variables. The equations can generate new meteorological variable values when you start with atmospheric initial conditions leading to improved weather predictions. Forecasters produce predictions about temperature pressure and humidity changes by using these equations under different conditions over time. The study conducted by Rani explores the development of a weather prediction model through machine learning methods (Rani, Designing a model for weather forecasting using machine learning., 2020).

The essential element in wind power forecasting (WPF) is numerical weather prediction (NWP) because it delivers future weather projections. The NWP projected wind speed (NWPWS) demonstrates a significant distinction from actual wind speeds because it measures their difference. The predicted wind speed follows the actual wind speed at different time points resulting in wind speed lag becoming the main cause of this difference. The timeframe-dependent characteristics of this time delay make WPF methods doubtful when selecting appropriate forecasted time periods. Block-Sparse Attention Range functions as the first component to extract crucial time segments from a wide variety of temporal delays. Through this stage the most significant lags for predictions are minimized in number. Lag Recognition compares the time series of NWPWS against real wind speed data to discover up to the top k probable temporal lags together with their estimated probability. The evaluation method determines series-wise match levels between predicted wind speeds and actual measurements. The Weighted NWP dataset achieves forecasting reliability through the Feature Fusion component that performs the final step of fusing adjusted NWP values based on temporal lag metrics. TLA functions as a vital feature processing component within WPF models which grants strong scalability in dealing with different forecasting needs. The algorithm has proven itself during practical field implementations which led developers to embed it into a modified MED system in order to demonstrate its impact on prediction efficiency. Retention lag prediction serves as an effective solution to enhance ultra-short-term wind power forecasting by resolving the timing challenges that lead to reduced accuracy levels. In the renewable energy industry TLA proves valuable because it enhances the analysis of temporal properties to produce better reliable wind power forecasting results.(Liu, 2023)

Jack's happy grin proves beyond any doubt that this time defines the best moment of all. The gentle wind today takes hold of his robust Irish hair and runs through it with ease. The copper-brown sunlight dances with tempting light upon his head despite the blowing wind which modifies the sun's position. The atmospheric mood radiates warmth although it is not the traditional summer time. His spouse expresses equal excitement because she constantly displays cheerful movements which align with the festive spirit of the day. She dons her sunglasses periodically and proceeds to take them off immediately showing she enjoys observing the transition between bright light and dim light. An effort to shield her eyes from the scorching midday sun leads her to put on the sunglasses before the roof's open position becomes too tempting to choose eye protection. Everything suggested that no discomfort would spoil the day's happiness as the perfect and vibrant situation remained unmoved by any difficulties. (Monarth, 2024)

A driver needs complete awareness of their environment to reach smart choices and prevent traffic accidents. Various weather elements such as fog create conditions which completely degrade driving visibility in actual road conditions. Areas with a high prevalence of fog encounters such difficulties in visual observation similarly in hilly terrain. Traffic security management encounters major risks during foggy conditions because visibility becomes severely restricted and the likelihood of accidents increases. The estimation of visibility during foggy weather conditions must be enhanced because it directly affects both driving safety and smooth traffic flow. Improved traffic management methods together with better driving choices and exact real-time visibility measurements could lower traffic accidents. The majority of present-day visibility evaluations during fog depend on specialized equipment positioned at permanent highway locations. These system devices manage to provide useful data but their costs stay high and their application choices stay restricted. The fixed-location systems operate too slowly to detect sudden fog development and route-fog-density variations in their limited operational area. The high expenses for system establishment and operation prevent them from being adopted across a wide range since many regions face financial barriers. (You, 2023)

This study details an 8-hour capturing session of WISE 1049AB nearby binary brown dwarf pair using James Webb Space Telescope (JWST) equipment that scanned the system first with MIRI LRS and then spent 7 hours in NIRSpec prism mode. Due to its proximity and high visibility WISE 1049AB stands as the most suitable brown dwarf system for thorough investigation of atmospheric characteristics which show signatures from water and methane and carbon monoxide (CO) along with an 3.3 µm methane absorption signature and potential silicate absorption detection in WISE 1049A exceeding 8.5 µm. (Biller, 2024) (Biller, 2024)

The energy simulation software demands weather simulation data consisting of minimum three-year averages for each hour regarding solar global radiance and diffuse and directed solar radiation. The dataset used for the study began at September 2011 because it contained weather records without measurements for both diffuse and direct irradiation obtained from the Centre of Energy and Environmental Engineering. The SOLYS 2 Sun Tracker performed annual measurements at the same facility to complete missing but necessary data about direct and diffuse radiation. The clearness index helped evaluate the connection between diffuse solar radiation and global solar radiation.

This study implements three linear regression methods which consider yearly, seasonal and hourly classifications spanning from six in the morning to six in the evening. Statistical tests such as mean bias error (MBE) and root mean square error (RMSE) and correlation coefficients measured the prediction strengths among different models which included the newly developed ones and those already utilized. The hour-wise model achieves superior performance based on results compared to the yearly and seasonal models. The results from the hour-wise model served alongside other prediction models to generate estimates for 6 AM, 11 AM, 1 PM and 6 PM periods. Based on the results of the assessment the authors recommend adopting the hour-wise model since it provides superior diffuse radiation predictions from the available data sets. (Sharma, A., & Marwaha, B. M. , 2015)

Energy systems with sustainable approaches emerged as a response to fight climate change effects that lead to intensified extreme weather occurrences during longer periods. The main power disruptions worldwide stem from severe weather that heavily damages essential infrastructure. Extreme and prolonged power outages that result from weather-caused damage to electrical systems extend from hours to days. The grid must receive immediate attention to become more resistant to weather storms due to current critical needs.

A study focuses on analyzing weather-operated power system component functionality and dependability regarding climate change effects. The paper examines existing strategies to highlight essential methods and diagnostic needs alongside conditional challenges for investigating severe thunderstorm influence on power disruptions.

Authors expand their research about vital infrastructure electrical systems with endurance concepts by demonstrating various protection strategies designed to enhance grid resilience against weather events. A detailed study research model outlines the process of predicting extreme weather network effects on power systems so future solutions can reduce these impacts. (Panteli, M., & Mancarella, P. , 2015)

This research investigated different caste groups' local weather adaptation methods in two Nepalese community forest user groups in Gorkha district. The study sites focused on Laxmi Mahila CF and Jalbire Mahila CF. The ability to adapt was assessed by examining factors that differently affected caste groups throughout the study. The study locations showed as their primary climate-related matters extreme heat, altered rainfall distribution and persistent drought periods.

At the research locations three main caste factions existed consisting of the disadvantaged subgroups Damai and Sarki together with ethnic Newar Gurung and Magar while the Brahmin and Chhetri comprised the higher caste group. Homeowners were categorized into wealth groups which included rich and two other classifications of medium and poor. Variable livelihood activities among hierarchy groups created direct consequences for their adaptive abilities. Members of higher caste groups preferred to obtain government employment while also working as teachers whereas ethnic community members primarily focused on local business ventures. Lower caste individuals who were economically marginalized primarily generated their money from work activities and tree practices as their primary income sources. Due to lesser options for gainful employment the adaptive capacity remained restricted among lower caste members. Community forests that depend heavily on forest resources lacked an adequate life sustaining solution during their establishment. The research data revealed that Laxmi Mahila CF held stronger adaptive capabilities than Jalbire Mahila CF. The two communities adopted different adaptation approaches through variations in their species distribution and local trade dynamics and their income activity patterns and livestock practices and employment systems. (Neupane, B., 2014

The accurate identification of pedestrians within different surroundings represents a crucial factor for boosting autonomous vehicle driving security. Increasingly hazy weather conditions present considerable performance obstacles to standard pedestrian detection systems because of poor visibility and image outline blurriness. This paper develops three new deep learning methods built on YOLO engineering to solve performance limitations using depth wise separable convolution with linear bottleneck setups to cut expenses and reduce parameter numbers. An innovative approach within the proposed method implements a weighted combination layer that combines different scale feature maps through a squeeze-and-excitation block to enhance feature extraction. The training process received additional enhancement by employing six distinct methods to augment pedestrian images taken from hazy weather conditions. The experimental findings show that the proposed approaches succeed in recognizing pedestrians during hazy conditions while showing better processing time and accuracy performance than current leading detection methods. (Li, 2019)9kl

# **Chapter-2: Software Requirement Specification (Sample Attached)**

# **SOFTWARE REQUIREMENTS SPECIFICATIONS (SRS)**

## **Stakeholder Overview**

Stakeholder understanding allows the application to match both user requirements and technical requirements..

### ****Primary Users****

Users who need precise weather updates to organize activities like planning or conduct farming or plan their travels form the key demographics for this application. The location-specific weather forecast becomes available to users irrespective of their residential setting which ranges from cities to countryside districts. The app provides simple design alongside user-friendly features which ensure both mobile and web platforms present a clear user experience for everyone..

### ****Development Team****

Students from the final year of computer science actively develop the application by using current technological frameworks. The development team uses Kotlin as well as Jetpack Compose for Android creation and implements React to develop the web interface. The developers build the program foundation while integrating weather-related API functions and testing program performance across every targeted operating system..

### ****Academic Supervisor****

The project supervisor from the university faculty acts as the oversight authority to guarantee development adherence to academic requirements and institutional criteria. At each project stage the main duty of this team member consists of providing guidance alongside milestone tracking while delivering feedback for continuous improvement. The project quality remains intact through their oversight which also confirms that research and development practices remain correct..

### ****External Service Providers****

The project depends on collaborative efforts between OpenWeatherMap and Google Maps to receive fundamental real-time data processing. Through their APIs these third-party platforms provide recent weather data together with precise location details that becoming part of the application functionality. The application gets precise weather information and mapping capabilities by utilizing this approach which delivers location-specific data needed for optimal user experience.

## **DOMAIN CONTEXT:**

Weather Wise App/Web operates from the domain which includes meteorological services together with smart weather technology..

**Meteorological Information Systems**

Weather Wise operates as a division within meteorological information system domains. The system collects real-time data for delivering both prompt and exact weather forecast information to its users. Weather updates are delivered locally through the system which provides measurements of temperature alongside humidity readings and wind speed reports together with severe severe weather warnings.

**Mobile Application Development**

Weather Wise implements the modern Android development tool Kotlin and Jetpack Compose for its mobile application development. Users encounter a smooth operating system because the interface follows an intuitive structure that works on multiple mobile devices. The mobile application delivers real-time weather reports to users who experience connectivity issues through its offline operation function.

**3. Web-Based User Interface**

The Weather Wise application offers a web interface that uses React development while integrating Tailwind CSS and Framer Motion together with support for React. The web-based equivalent of Weather Wise allows users to retrieve weather data using any current browser thus maintaining coherent functionality and design elements between platforms.

**Integration with External APIs**

Accurate weather predictions and locations-based information require integration with Open Weather Map and Google Maps. The APIs function as a bridge between the application and web interface so they can instantly retrieve and present weather data based on actual user position..

**Real-Time Data Processing and User Interaction**

The application alongside the web platform maintains effective processing capabilities for real-time data. The system admits users to suitable updates about severe weather alerts and live information and multi-day predictions by delivering essential data urgently. The system provides an interactive design together with user-friendly elements which creates an engaging and operational experience.

## **FUNCTIONAL REQUIREMENTS:**

System users need the main capabilities outlined in these functional requirements.

### ****Live Weather Display****

Users can view current weather conditions through the application which shows temperature together with humidity data and wind speed and direction as well as atmospheric pressure measurements. The app depends on its built-in location services to discover the device user's exact spot and deliver weather reports for that precise location automatically. Users receive the newest relevant weather information thanks to this system so they maintain current access.

### ****Multi-Day Forecast****

Users can access five-day forecasts through the application to see weather conditions that will occur during the upcoming period. Temperature patterns together with precipitation probabilities along with other significant metrics are included for each day in the weather forecast. Visual indicators such as icons along with graphical charts allow users to interpret information easily so they can develop better activity plans according to weather forecast predictions. Users who rely on this extended weather forecast can choose their activities well in advance for travel planning and outdoor events as well as agricultural work. Looking ahead with extended weather information lowers the chances of encountering unexpected weather changes during the day. Users can access the detailed presentation style easily to make effective daily plans.

### ****Critical Weather Notifications****

A real-time alert system within the application alerts users instantly about stormy conditions and heavy rainfall as well as intense wind situations. Users obtain critical weather alerts in real time through push notifications since these notifications provide instant warning capabilities for prompt action. The early warning system of the application protects users in unsafe environments by delivering timely warnings that boost safety together with preparedness. The application delivers notifications through alerts to users about important updates regardless of their current application usage.

### ****Support Without Internet****

### Users can access weather data using the offline mode after it fetches new updates from the server even when the application detects no internet connection. The app function depends on local storage to preserve critical data for operating normally when networks are unavailable. Users can check essential weather data no matter what their current internet status is because the application maintains key weather information display. This proves beneficial in isolated locations or crisis situations where network connectivity might be reduced.

### ****Location-Based Data****

Weather reports on the app are generated through automatic geographic location detection from its built-in mapping services. The system provides weather predictions that match exactly the user's current location through this functionality. Real-time location tracking within the application enables precise accurate weather updates for each user without demanding any user involvement.

### ****Simple User Experience****

Users can easily navigate the application interface because developers designed it for both beginners and experts. The application integrates contemporary design features which simplify weather information access through interactive elements. The system presents an attractive structure which enables users to locate their required information without ambiguity. The combination of visual elements and responsive design functions helps users to engage more actively with the platform leading to better satisfaction.

## **NON-FUNCTIONAL REQUIREMENTS:**

Such attributes determine system operational performance rather than specifying actual system functions.

### ****Speed and Responsiveness****

The app should focus on rapid load times together with quick data retrieval to provide users with an optimal smooth experience. Responses from smooth transitions improve user experience through a more realistic interface because these features create smooth processing that feels responsive to the user. Users will experience an efficient and pleasant app navigation when the optimization combines speed performance with smooth visual flow..

### ****Data Accuracy****

App users must be able to rely on forecasts that maintain a direct correspondence to genuine world conditions since these predictions should be both precise and dependable. Security in data sourcing demands selecting validated information from established providers to guarantee reliable and standardized results. The application delivers dependable and precise predictions through its use of verified data sources allowing users to base their decisions on reliable information.

### ****Ease of Use****

Users can easily navigate through the app because it displays its features through a well-designed interface that maintains simple navigation. Users can access fundamental and simplified information when needed but the system also supports detailed precise forecasts for advanced users. The system design makes the app usable by every type of user from novices to experts.

### ****System Maintainability****

The program uses modular design with reusable code to enable effortless maintenance operations and application update processes. The app maintains effective addition of new features through this approach which doesn't affect current performance. The app's structural design supports both speedy repairs of bugs and development of performance enhancements that allow developers to fix problems swiftly and provide users with smooth enhancements.

### ****Future Scalability****

Future updates include new functionality because the application maintains scalability by enabling new features like air quality monitoring and radar maps and pollen count tracking as user requirements change. As a result of its adaptable design the app maintains stable and effective performance levels when dealing with growing user numbers and larger data requirements throughout its expansion period.

### ****Secure Data Usage****

The application secures user location information along with API data through encryption approaches that fulfill industry standards. The app secures both services and personal user information through effective security measures that create trustworthiness in addition to protection from potential threats for user data when accessing the app..

### ****Accessibility and Inclusivity****

The app is carefully designed with a focus on font readability and visual clarity to ensure that all text is easy to read and accessible to users. Looking ahead, there are plans to enhance accessibility further by potentially incorporating voice commands or adding compatibility with screen readers, making the app even more inclusive for users with varying needs.

**TECHNICAL REQUIRMENTS:**

|  |  |
| --- | --- |
| **API Integration** | Seamless connection with third-party weather APIs. |
| **Data Processing and**  **AI Forecasting** | Uses machine learning to improve predictions. |
| **Database Management System** | Stores user settings and offline data. |
| **Push Notification System** | Ensures timely alerts for severe weather conditions. |
| **User-Friendly UI/UX** | Optimized design for accessibility and readability. |

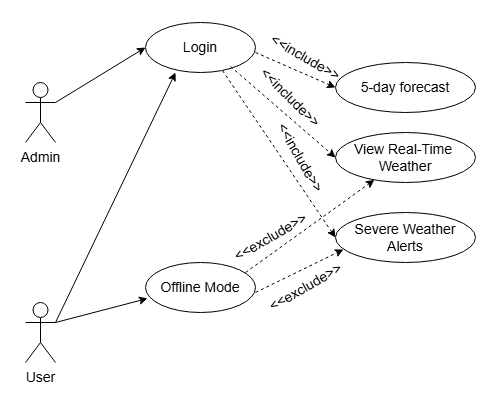
# **Chapter 3: Analysis**

**COMPETITOR ANALYSIS:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **FEATURE** | **WEATHER WISE** | **ACCUWEATHER** | **YAHOO WEATHER** | **THE WEATHER CHANNEL** |
| **Real-Time Updates** | Yes | Yes | Yes | Yes |
| **Hyper Local Accuracy** | Yes | No | No | No |
| **Severe Weather Alerts** | Yes | Yes | No | Yes |
| **Advanced Weather Data** | Yes | Yes | No | Yes |
| **Offline Mode** | Yes | No | No | No |
| **Customizable UI** | Yes | No | No | Yes |

**CHAPTER 3**

**USE CASE DIAGRAM:**



**DESCRIPTION:**The weather application use case overview diagram shows how roles and main functions interact in the system with specific point-to-point relationships.

**Actors:**

General users possess access to weather information through this application.

System administration duties fall within the scope of Admin role because this privileged account provides users with both system management capabilities and monitoring access.

#### **Core Use Cases:**

**1. The application demands a successful authorization process before giving users access to its available features.**

**The system demands this function before users can start using other application features.**

**2. The system shows current meteorological information according to user-defined location or chosen area both in real time.**

**Through the product users obtain complete detailed weather projections spanning five days which support their planning needs.**

**4. The application alerts users through notification when it detects severe weather warning conditions such as heatwaves or heavy rainstorms or storms.**

**5. Users have access to stored weather information through Offline Mode when an Internet connection is not available.**

#### **Relationships:**

The Login use case serves as a prerequisite for entering any of the fundamental use cases including View Real-Time Weather along with 5-Day Forecast and Severe Weather Alerts and Offline Mode. The authentication process becomes guaranteed through this measure. Users cannot view Real-Time Weather and Severe Weather Alerts because they have enabled Offline Mode. The offline functionality depends on cached data yet it becomes incapable of showing real-time data unless there exists a network link.

#### **Use Case Flow:**

**1. After authentication the app provides complete feature access to the user through its login process.**

* **Login → View Real-Time Weather**
* **Login → 5-Day Forecast**
* **Login → Severe Weather Alerts**
* **Login → Offline Mode**

**When users log into the system they gain access to present weather conditions together with notifications about dangerous weather systems. Weather data along with live alerts become unavailable during periods when offline mode is enabled.**

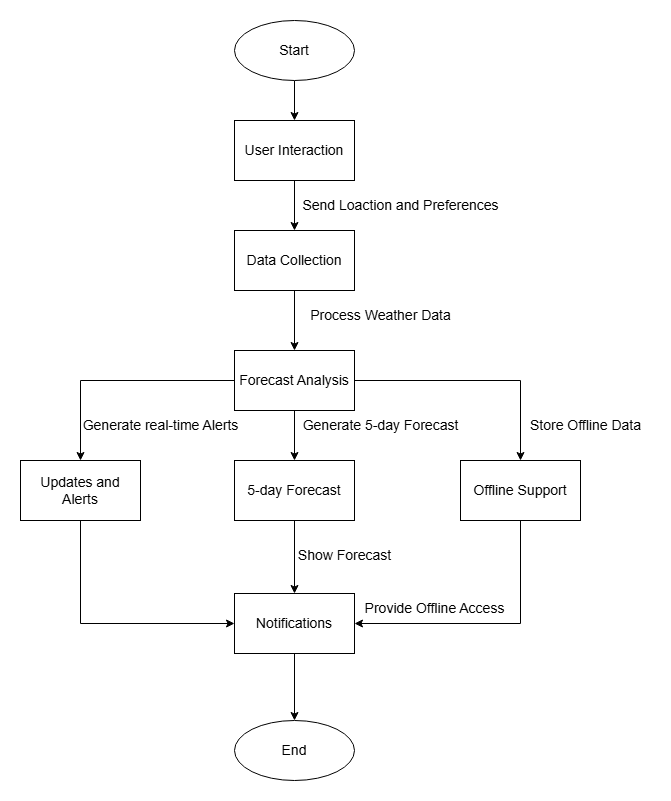
**Users in Offline Mode have access to their cache data for monitoring but will only get updated weather information once their internet connection becomes active.**

### ****Summary****:

The user and administrator roles operate within this use case model while defining the essential functions which include entry authentication followed by weather visualization with predictive services and warning alerts and an offline capability. The system succeeds because login authentication serves as the basic requirement while offline operation limits access to various platform capabilities when the user becomes disconnected.

# **Chapter-4: Design**

**FLOW DIAGRAM:**



**Start**

A user’s journey within the system starts by executing the application. The system accepts user interaction at this initial stage which starts its workflow routine.

**User Interaction**

Users use the interface to reach features and conduct data entry. The system progresses through its following stages after receiving this user interaction.

**Data Collection**

Customer data collection happens through the system by obtaining the user's information including their location specifications and personal preferences. The supplied information functions as an important element that enables result customization.

**Sending Location and Preferences**

The processing unit along with server receives information from users. The system at this point performs analysis of practical conditions as well as user requirements.

**Forecast Analysis**

After processing the gathered data the system produced output predictions or recommendations. The analysis process represents the main operational component of the offered service.

**Updates and Alerts**

The system sends prompt warnings and updates using processed information. The system generates timely notices which provide awareness about pending upcoming occasions or new developments to users.

**5-Day Forecast**

The system displays a forecast which extends through five consecutive days for user viewing. The generated information helps users make effective decisions while boosting their planning abilities.

**Offline Support**

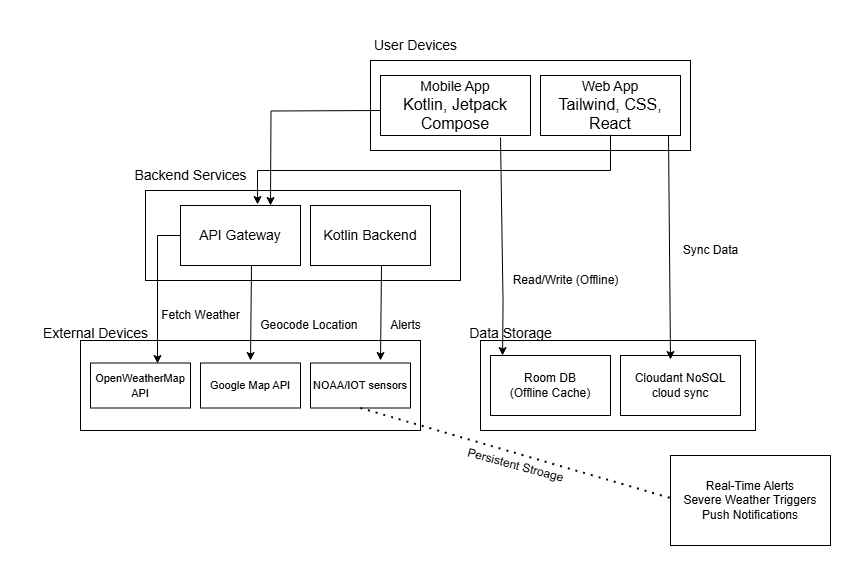
The system enables users to continue utilizing specific tools when internet connection is unavailable. Users can maintain uninterrupted service access when their location has restricted or no internet connectivity.

**Notifications**

Users receive instant notifications through the system to stay updated. Alerts in the system transmit a combination of essential information that consists of updates alongside warnings with possible reminders.

**End**

The system procedure terminates at the user's choice for concluding their workspace session. All system functions reach their termination point in this stage.

**ARCHITECTURE DIAGRAM:**

**DESCRIPTION:**

#### **User Layer**

* **Mobile App (Android)**
  + **Tech Stack:** Jetpack Compose (Kotlin) for declarative UI + Room Database for offline-first design.
  + **Key Role:**Handles GPS/input-based location, displays real-time weather with offline fallback.
* **Web App**
  + **Tech Stack:** The web application uses React together with Tailwind CSS utilities and Framer Motion micro-interactions for its functionality.
  + **Key Role:**Delivers responsive, animated forecasts accessible via browsers.

#### **2. Backend Layer**

* **API Gateway**
  + **Integration:** Unified endpoint for OpenWeatherMap (weather metrics) and Google Maps (lat/long conversion).
* **Business Logic (Kotlin)**
  + **Core Functions:**
    - Data normalization from disparate APIs
    - Rule-based alert triggers (e.g., NOAA tornado warnings)
    - Cache invalidation strategies

#### **3. Data Layer**

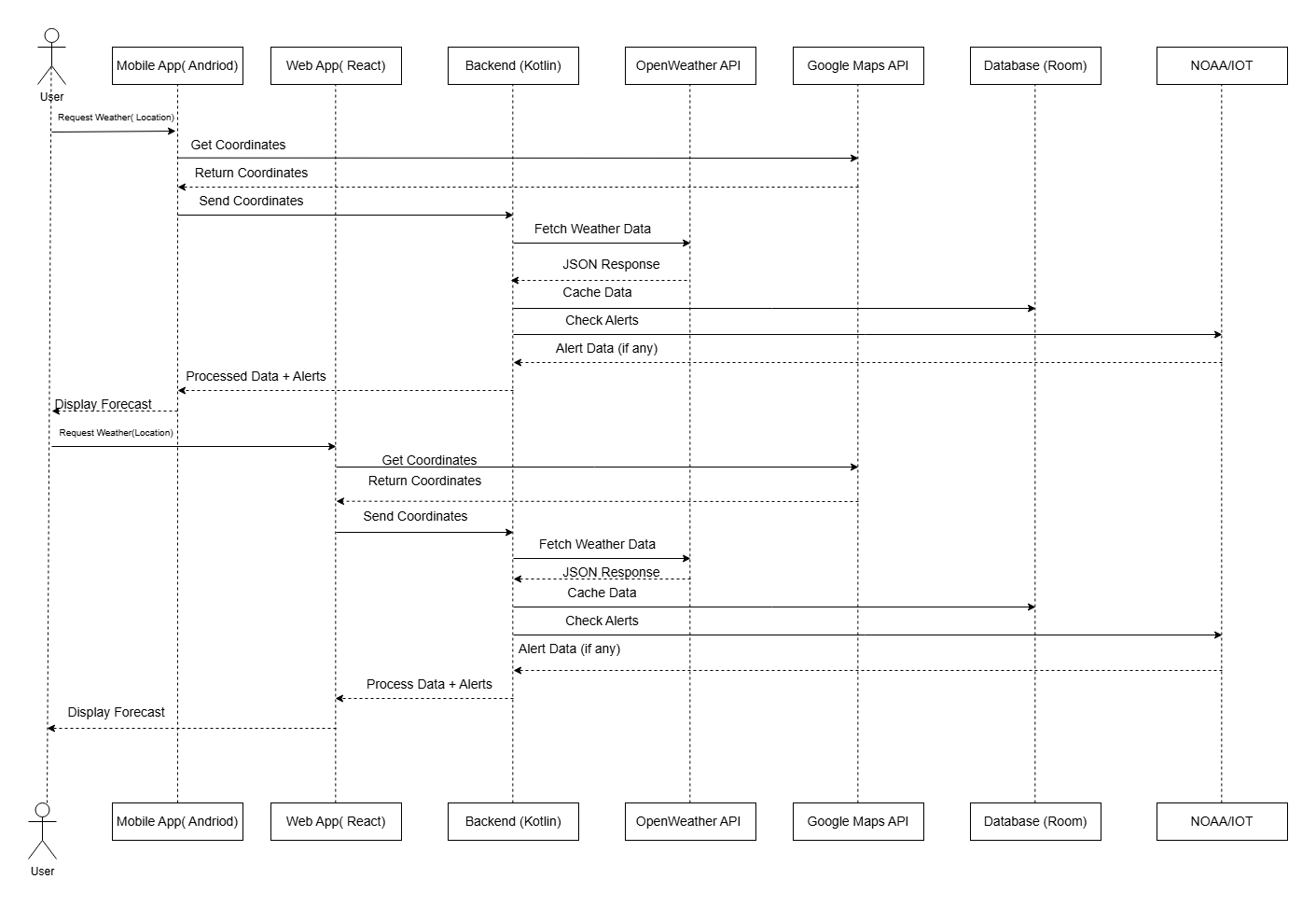
* **Local Storage (Room DB)**
  + **Mobile-Exclusive:** SQLite abstraction for offline access to recent forecasts.
* **Cloud Storage (Cloudant NoSQL)**
  + **Web-Focused:**JSON document store enabling multi-device sync via IBM Cloud.

#### **4. External Services**

* **NOAA/IoT Sensors**
  + **Real-Time Feeds:**The application receives push notifications for flood and storm alerts through its real-time feed system.
* **Third-Party APIs**
  + **OpenWeatherMap:** 5-day forecasts at 3-hour intervals.
  + **Google Maps:** Google Maps converts coordinate data to human-address formats through reverse address geocoding.

### ****Flow****:

* The application and web interface allow users to initiate weather information requests.
* The app retrieves data from OpenWeatherMap/Google Maps APIs.
* The application processes and stores data either in Room which operates at a local level or in Cloudant which operates at cloud level before sending results to users.
* The system shows immediate alerts by push notifications when severe weather conditions occur.

**SEQUENCE DIAGRAM:**

**DESCRIPTION:**

#### **1. Location Acquisition**

Whenever we seek to determine a user's location the process starts from this point. The system retrieves the user position automatically through GPS Access when GPS permissions are given. When GPS access is blocked or unavailable the system lets users manually add their location through a manual input method (for instance on web or privacy-restricted devices).

#### **2. Coordinate Conversion**

The data collected from location capture moves forward to Google Maps API where it performs the following actions: The system uses precise latitude and longitude as the final destination of any addresses or place names. The system validates the obtained location to guarantee its accuracy. Elevation information can also be included as an optional aspect when the data is used for terrain-based forecasting.

#### **3. Backend Processing**

The Kotlin-backed backend system manages location information through the following procedures: The API request preparation includes all required keys together with necessary parameters. The backend system applies rate limit management to stop API throttling events. The system uses error handling modules to maintain data stability while ensuring complete data consistency.

#### **4. Weather Data Retrieval**

The backend system sends processed coordinates to OpenWeatherMap in order to retrieve weather data. Current conditions: temperature, humidity, wind speed, pressure, etc. 5-Day Forecast: hourly and daily weather predictions. The alerts system provides notices about threatening severe weather conditions which include floods and storms.

#### **5. Data Caching & Storage**

Both speed optimization and offline operation require the following features:

**Mobile App**: The mobile application stores data in Room Database (SQLite) which enables offline operation.

**Web App**: Syncs weather data with **Cloudant NoSQL**, enabling persistent, cross-device access.

#### **6. Alert Verification**

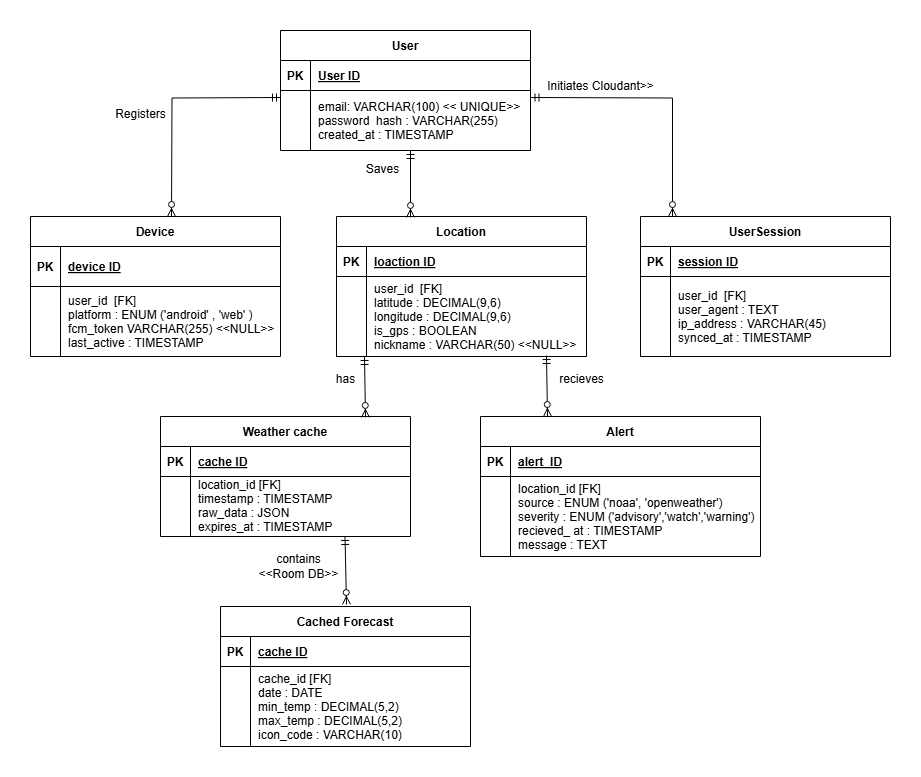
* NOAA verifies all weather alerts through an evaluation system.
* The application verifies weather alert information from OpenWeatherMap through its process.
* Supplements with any **missing or additional warnings**.
* Severe threats such as tornadoes or flash floods take precedence as the notifications appear first.

#### **7. User Delivery**

The final, verified weather data reaches users through:

The application displays dynamic visualizations and animated forecasting elements through Jetpack Compose or React programming interfaces. Mobile devices show timely notifications when severe weather threatens the area. SMS alerts are optional in the system to send critical information during severe weather conditions when mobile data does not function.

The predefined workflow enables precise real-time weather delivery along with offline support and early safety warning notifications to create Weather Wise as a robust user-friendly system.

 **ERD DIAGRAM:**

### ****Weather Wise – ERD Description****

#### **1. Core Structure**

#### The system implements dual-storage architecture which connects shared centralized storage with storage techniques optimized for each platform. The user identity together with devices and locations remain consistent through Central Entities which are accessible by both mobile and web applications. The system implements Platform-Specific Entities that optimize each platform while using web on the right and mobile on the left.

#### **2. Central Entities**

**User Management**

Resident credentials together with metadata find a home in the User Table. All account creation requires a distinctive email to avoid account duplicates. The application uses hashed passwords for enhanced security protection of user information. The system tracks user account creation times for purposes of audits.

**Device Tracking**

User logins across different platforms receive management through the Device Table system. The platform enum enables the system to assign access labels between 'android' and 'web'. The application stores Firebase Cloud Messaging (FCM) tokens in order to activate mobile push notifications. The system records time stamps for active session tracking which affects both expiration logic and analytic purposes.

**Location Services**

* Location Table maintains user-saved places.
* The application stores spatial coordinates using a maximum precision of 9 decimal points for precision purposes.
* Differentiates between GPS and manual input.
* Users can give names to their entries such as “Home” and “Work” for better identification.

#### **3. Weather Data System**

**Shared Components**

**Weather Cache**

* The system maintains an interim storage of weather API JSON data prior to its processing.
* The system employs timestamps in combination with expiration fields for the purpose of cache management.
* The system arranges location-specific data for immediate and appropriate data retrieval.

**Alert System**

#### The application merges current weather alerts from NOAA and OpenWeather into one consolidated system.

#### The system provides an alert classification mechanism which types alerts into three different severity categories such as advisory together with warning and emergency.

#### The system stores entire alert messages to access them during display and review purposes.

#### **4. Platform-Specific Implementations**

**Mobile (Room Database)**

**Cached Forecast** **table:**

* The database includes three structured fields with space for temperature ranges and weather icons and dates of prediction.
* The database implements relational consistency which maintains data integrity when not connected to the internet.

**Web (Cloudant NoSQL)**

* **User Session** documents:
  + Tracks **browser/device info** for each login.
  + Logs **IP addresses** for security and anomaly detection.
  + The system runs timestamps for record synchronization which helps maintain consistent user experiences.
  + Uses a **flexible document-based schema**, ideal for diverse browser environments.

#### **5. Key Relationships**

* **1-to-Many Mappings**:
* **Each user maintains a direct relationship with different devices since multiple logins occur from individual devices.**
* **Through this connection a single user possesses the capability to add multiple stored locations.**
* **Weather data exist for every single saved location between the schema.**
* **Specialized Links**:
  + **Weather Cache → Forecasts** (mobile-only): The Weather Cache service enables mobile users to access forecasts through its restricted Forecasts (mobile-only) connection which functions directly together.
  + **User → Sessions** (web-only): Used for tracking active and past logins on browsers.

The modular design together with the standardized structure leads to scalable operations and data reliability and promotes optimized platform utilization which enables effortless joint usage of Android and Web interfaces.

# **Chapter 6: Testing**

## **6.1 Test Scenario**

Tests scenarios describe at a high level the items that require evaluation. The system evaluation tests actual user activities to verify proper operation from the perspective of end-users. Scenarios serve to reveal the proper function of the system when users engage with it as expected.

Here are the **Use Cases** for your **WeatherWise** app features, presented in a clean, detailed table format:

**🔹 UC-12: Live Weather Display**

|  |  |
| --- | --- |
| **Field** | **Description** |
| **Use Case ID** | UC-12 |
| **Name** | Live Weather Display |
| **Primary Actor** | User |
| **Preconditions** | Device location services enabled and internet connection available |
| **Description** | Displays real-time weather (temperature, humidity, wind speed/direction, pressure) based on the user’s current location. |
| **Main Flow** | 1. User launches the app 2. App requests GPS location 3. Weather API called with coordinates 4. Real-time weather displayed |
| **Alternate Flow** | A1: Location not available → App prompts user to enable it A2: API fails → Show error message or cached data |
| **Postconditions** | Weather data relevant to the user’s current location is displayed |

**🔹 UC-13: Multi-Day Forecast**

|  |  |
| --- | --- |
| **Field** | **Description** |
| **Use Case ID** | UC-13 |
| **Name** | Multi-Day Forecast |
| **Primary Actor** | User |
| **Preconditions** | Internet connection available; location or city selected |
| **Description** | Displays weather forecast for the next 5 days with visual icons and temperature data. |
| **Main Flow** | 1. User navigates to forecast screen 2. Forecast API is called 3. Data displayed with icons and summaries per day |
| **Alternate Flow** | A1: API returns error → Show fallback or cached data |
| **Postconditions** | User sees 5-day weather trends for planning |

**🔹 UC-14: Critical Weather Notifications**

|  |  |
| --- | --- |
| **Field** | **Description** |
| **Use Case ID** | UC-14 |
| **Name** | Critical Weather Notifications |
| **Primary Actor** | User |
| **Preconditions** | Notification permission granted; active internet |
| **Description** | Sends real-time alerts for severe weather conditions via push notifications. |
| **Main Flow** | 1. App receives alert from server 2. Displays in-app banner or push alert 3. User clicks for details |
| **Alternate Flow** | A1: Notification disabled → No alert shown |
| **Postconditions** | User is notified of potential weather threats promptly |

**🔹 UC-15: Offline Support**

|  |  |
| --- | --- |
| **Field** | **Description** |
| **Use Case ID** | UC-15 |
| **Name** | Offline Support |
| **Primary Actor** | User |
| **Preconditions** | Cached weather data exists from last online session |
| **Description** | Displays most recent weather information when offline using locally stored data. |
| **Main Flow** | 1. App detects no internet 2. Loads and displays last saved weather data |
| **Alternate Flow** | A1: No cached data available → Show error or “No data” message |
| **Postconditions** | User can still access weather data offline |

**🔹 UC-16: Location-Based Data**

|  |  |
| --- | --- |
| **Field** | **Description** |
| **Use Case ID** | UC-16 |
| **Name** | Location-Based Forecast |
| **Primary Actor** | User |
| **Preconditions** | Location access granted |
| **Description** | Uses GPS to fetch weather for the user’s current location automatically. |
| **Main Flow** | 1. GPS detects location 2. App sends coordinates to weather API 3. Displays localized forecast |
| **Alternate Flow** | A1: GPS fails → Prompt user to manually enter city |
| **Postconditions** | User receives accurate, location-specific weather updates |

**🔹 UC-17: Simple User Experience**

|  |  |
| --- | --- |
| **Field** | **Description** |
| **Use Case ID** | UC-17 |
| **Name** | Simple User Experience |
| **Primary Actor** | User |
| **Preconditions** | App installed |
| **Description** | User navigates and interacts with the app easily due to clean layout and intuitive design. |
| **Main Flow** | 1. User launches app 2. User navigates tabs and screens 3. Weather data is easily accessible and readable |
| **Alternate Flow** | A1: Accessibility needs (e.g., larger text) → Adjust via settings |
| **Postconditions** | User can understand and interact with all app functions without confusion |

**login**

**Use Case Table**

|  |  |
| --- | --- |
| **Use Case ID** | **UC-01** |
| Name | User Registration & Login |
| Actors | User |
| Description | Allows users to register and authenticate into the system |
| Preconditions | User has access to the app or website |
| Postconditions | User is authenticated and redirected to the home/dashboard |
| Main Flow | 1. User opens app 2. Enters email and password 3. System authenticates 4. User is redirected |
| Alternate Flows | A1. Invalid credentials A2. Weak password A3. Email format invalid |

**Test Scenarios Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID** | **Description** | **Input** | **Expected Result** |
| TC-01 | Register with valid details | Valid email & strong password | Registration success |
| TC-02 | Register with invalid email | test@ | Error: Invalid email |
| TC-03 | Register with weak password | Abc | Error: Weak password |
| TC-04 | Login with correct credentials | Email + Password | Login success |
| TC-05 | Login with wrong password | Email + wrong password | Error: Incorrect password |
| TC-06 | Login with unregistered email | Unregistered email | Error: Email not found |

|  |  |
| --- | --- |
| **Use Case ID** | **UC-07** |
| Name | Search for Cities |
| Actors | User |
| Description | Search and select a city for weather info |
| Preconditions | User is on home/search screen |
| Postconditions | Weather info for selected city is shown |
| Main Flow | 1. Enter city name 2. Suggestions appear 3. Select city |
| Alternate Flows | A1. No matching city |
| **Test Case ID** | **Description** | **Input** | **Expected Result** |
| TC-24 | Search by city name | "Lahore" | Lahore shown |
| TC-25 | Typo in city | "Lahoor" | Suggestion not found |
| TC-26 | Use recent search | Tap recent | City loaded |

**5 DAY FORECAST**

|  |  |
| --- | --- |
| **UC ID** | **2.1** |
| **Name** | View 5-Day Weather Forecast |
| **Actor(s)** | User (General User, Student, Teacher) |
| **Precondition** | App is running with internet & location access |
| **Description** | User views a 5-day weather forecast with daily temperature, precipitation, and icons |
| **Main Flow** | 1. Open app 2. Detect location 3. Open Forecast tab 4. Display 5-day data |
| **Alternate Flow** | No internet → show cached data or error No data for location → show error |
| **Postcondition** | User sees forecast & plans accordingly |
| **Extensions** | Tap day → see detailed view Pull to refresh → update forecast |

Here are the **test case scenarios** for the **WeatherWise** app features using the format you requested:

**Live Weather Display**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID** | **Description** | **Input** | **Expected Result** |
| TC-01 | Display weather with internet and GPS | Internet ON, GPS ON | Current location weather data shown |
| TC-02 | Display weather without internet | Internet OFF, GPS ON | Last known weather data shown |
| TC-03 | Display weather without GPS | Internet ON, GPS OFF | Prompt for location access |

**Multi-Day Forecast**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID** | **Description** | **Input** | **Expected Result** |
| TC-04 | View 5-day forecast | Tap “Forecast” tab | Display upcoming 5 days weather |
| TC-05 | Tap on specific day | Select Day (e.g., Thursday) | Show detailed weather for selected day |

**Critical Weather Notifications**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID** | **Description** | **Input** | **Expected Result** |
| TC-06 | Receive severe weather alert | Trigger storm alert | Push notification displayed |
| TC-07 | Open weather alert | Tap on notification | Detailed alert info displayed |

**Support Without Internet**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID** | **Description** | **Input** | **Expected Result** |
| TC-08 | Access weather in offline mode | Internet OFF | Cached data displayed |
| TC-09 | Reconnect and update weather | Internet ON | Latest weather data loaded |

**Location-Based Data**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID** | **Description** | **Input** | **Expected Result** |
| TC-10 | Detect current location | GPS ON | Weather data shown for current location |
| TC-11 | Move to new location | Change GPS location | Weather updates to new location |

**Simple User Experience**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID** | **Description** | **Input** | **Expected Result** |
| TC-12 | Launch application | Tap app icon | Weather dashboard displayed clearly |
| TC-13 | Navigate to forecast section | Tap forecast tab | Easy transition to forecast screen |
| TC-14 | Use back navigation | Tap back button | Return to previous screen without issues |

**5-Day Forecast – Test Case Scenarios**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID** | **Description** | **Input** | **Expected Result** |
| TC-01 | Open 5-day forecast screen | Tap on "Forecast" tab | Display weather for the next 5 days |
| TC-02 | View forecast for Day 1 | Tap on Day 1 (e.g., Monday) | Show detailed forecast (temperature, precipitation, etc.) |
| TC-03 | View forecast for Day 2 | Tap on Day 2 (e.g., Tuesday) | Show detailed forecast for Day 2 |
| TC-04 | View forecast for Day 3 | Tap on Day 3 (e.g., Wednesday) | Show detailed forecast for Day 3 |
| TC-05 | View forecast for Day 4 | Tap on Day 4 (e.g., Thursday) | Show detailed forecast for Day 4 |
| TC-06 | View forecast for Day 5 | Tap on Day 5 (e.g., Friday) | Show detailed forecast for Day 5 |
| TC-07 | Display weather icons and charts | Open forecast screen | Weather icons and trend charts are visible and accurate |
| TC-08 | Refresh 5-day forecast data | Pull to refresh or relaunch app | Forecast updates to latest available data |
| TC-09 | No data available from API | Open forecast with no internet | Show error or cached forecast data |
| TC-10 | Forecast with changing location | Move to a different city | Forecast updates to new location automatically |

Here is a **complete Functional Requirements Table for the WeatherWise application**, covering all the main features described in your document:

**Functional Requirements – WeatherWise App**

|  |  |  |  |
| --- | --- | --- | --- |
| **FR No.** | **Feature** | **Requirement Description** | **Priority** |
| FR1.1 | Live Weather Display | The system shall display real-time weather data including temperature, humidity, wind speed/direction, and pressure. | High |
| FR1.2 | Live Weather Display | The system shall use device GPS to detect the user's current location automatically. | High |
| FR2.1 | 5-Day Forecast | The system shall display weather forecasts for the next 5 days. | High |
| FR2.2 | 5-Day Forecast | The forecast shall include temperature, precipitation, and weather conditions for each day. | High |
| FR2.3 | 5-Day Forecast | The system shall show visual elements like icons and charts for each day's weather. | Medium |
| FR2.4 | 5-Day Forecast | The user shall be able to tap a day to view detailed forecast information. | Medium |
| FR3.1 | Critical Weather Notifications | The system shall notify users of severe weather events via push notifications. | High |
| FR3.2 | Critical Weather Notifications | The system shall deliver alerts even when the app is not open. | High |
| FR4.1 | Offline Support | The system shall store the latest fetched weather data for offline access. | Medium |
| FR4.2 | Offline Support | The system shall display cached weather data if the device is offline. | Medium |
| FR5.1 | Location-Based Data | The app shall use real-time location data to deliver localized weather updates. | High |
| FR5.2 | Location-Based Data | The system shall update the weather data automatically when the user's location changes. | Medium |
| FR6.1 | Simple User Experience | The system shall feature a clean and easy-to-navigate user interface. | High |
| FR6.2 | Simple User Experience | The system shall use icons, layouts, and responsive design to enhance readability and interaction. | Medium |

**Non-Functional Requirements – WeatherWise App**

|  |  |  |  |
| --- | --- | --- | --- |
| **NFR No.** | **Category** | **Requirement Description** | **Priority** |
| NFR1.1 | Performance | The app shall load live weather data within 2 seconds under normal network conditions. | High |
| NFR1.2 | Performance | The app shall update forecast and location data in real-time with minimal lag. | High |
| NFR2.1 | Scalability | The system shall support at least 10,000 concurrent users without performance degradation. | High |
| NFR3.1 | Reliability | The app shall maintain 99.9% uptime, ensuring availability of alerts and data. | High |
| NFR3.2 | Offline Access | The app shall display cached weather data when the user is offline. | Medium |
| NFR4.1 | Usability | The UI shall follow accessibility standards (e.g., WCAG) for users with visual or motor impairments. | High |
| NFR4.2 | Usability | The app shall provide a consistent, responsive interface across all supported devices (Android, iOS). | High |
| NFR5.1 | Security | All user data, including location information, shall be encrypted in transit and at rest. | High |
| NFR5.2 | Security | The app shall use secure authentication protocols (e.g., OAuth2 for user settings and notifications, if any). | Medium |
| NFR6.1 | Maintainability | The app shall follow modular architecture to allow easy updates and feature additions. | Medium |
| NFR6.2 | Error Handling | The app shall log errors and alert the system administrator of any critical failures. | Medium |
| NFR7.1 | Compliance | The app shall comply with GDPR/CCPA for handling user data and location tracking. | High |
| NFR8.1 | Battery Optimization | The app shall minimize background GPS and network usage to preserve device battery life. | Medium |

This table ensures a full picture of what the WeatherWise app should *do* (Functional Requirements) and *how* it should perform (Non-Functional Requirements).

1. 6.1 Test Scenario

A test scenario is a high-level description of what needs to be tested. It focuses on real-world user activities to ensure the system functions correctly from an end-user perspective. Scenarios are used to confirm that the system behaves properly under expected usage conditions.

**Example Scenario:**

A user opens the application, enters their username and password, and logs into the system successfully to access the dashboard.

This scenario checks whether the authentication mechanism works correctly and whether access to the dashboard is granted only after successful login.

1. 6.2 Test Plan

The test plan is a comprehensive document that defines the scope, approach, resources, and schedule for testing activities. It serves as a blueprint to guide the testing process throughout the software development life cycle.

**Main Components of a Test Plan:**

* **Objective:** To verify that the system fulfills all specified requirements and performs as expected.
* **Scope:** Includes testing functionalities such as registration, login, service request submission, response tracking, and logout.
* **Test Strategy:** Describes how testing will be performed (manual or automated), and what levels of testing will be used (unit, integration, system, and acceptance).
* **Resources Needed:** Testers, development team members, testing tools, and test environments.
* **Test Schedule:** Defines when testing will start, key milestones, and deadlines for completion.
* **Deliverables:** Test cases, test scripts, test results, defect reports, and the final test summary report.

1. 6.3 Definition of Test Cases

Test cases are detailed documents specifying the steps, inputs, and expected results used to verify that specific functionalities of the system work as intended. They serve as a reference to ensure that no critical test is overlooked during execution.

**Each test case includes:**

* **Test Case ID:** Unique identifier for tracking.
* **Test Scenario:** A short description of what is being tested.
* **Test Steps:** Step-by-step instructions for executing the test.
* **Input Data:** The actual data used during testing.
* **Expected Result:** The outcome expected if the system behaves correctly.
* **Actual Result:** The observed outcome after execution.
* **Status:** Pass or Fail, based on the match between expected and actual results.

1. 6.4 Test Case Specification

This section organizes all defined test cases into a structured format, usually presented in tables. It ensures that every function and user requirement has at least one associated test case.

**Example Specification Table:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | **Description** | **Input** | **Expected Output** | **Status** |
| TC001 | Login with correct input | Username: user Password: 1234 | User redirected to dashboard | Pass |
| TC002 | Login with wrong password | Username: user Password: xyz | Error message displayed | Pass |
| TC003 | Submit new service request | Form with valid service details | Confirmation message shown | Pass |

1. 6.5 Test Case Results

After executing the defined test cases, results are collected and documented. These results show whether the system works as expected under various conditions.

***6.5.1 Black Box Testing***

Black box testing evaluates the functionality of the application without looking into the internal structure or code. It checks whether the system produces the correct output for a given input, focusing solely on what the system should do.

**Key Characteristics:**

* Testers do not need coding knowledge.
* Based on requirements and specifications.
* Useful for validating user interfaces and system interactions.

**Example:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | **Function Tested** | **Input** | **Expected Output** | **Result** |
| BB001 | Login Function | Valid credentials | Access granted | Pass |
| BB002 | Invalid Login Attempt | Incorrect password | Display login error | Pass |
| BB003 | Submit service request | Valid data | Confirmation message | Pass |

***6.5.2 White Box Testing***

White box testing involves checking the internal workings of the application, such as control structures, data flow, and logic conditions. It is performed by developers or testers with coding knowledge.

**Key Characteristics:**

* Focuses on internal logic and code coverage.
* Ensures all code paths, branches, loops, and conditions are tested.
* Helps in optimizing and debugging code.

**Example:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | **Module** | **Technique Used** | **Objective** | **Status** |
| WB001 | Authentication | Condition Coverage | Ensure all conditional branches are executed | Pass |
| WB002 | Database Module | Path Testing | Verify all code paths are tested | Pass |
| WB003 | Service Handler | Loop Testing | Check loop behavior with multiple inputs | Pass |

# Chapter 7: Conclusion And Future Work

FUTURE WORK

The Weather-Wise project's future advancement presents diverse opportunities for capability strengthening while simultaneously improving user interaction and platform integration and social effects. The project should expand by developing native iOS applications through SwiftUI or Flutter which serves as cross-platform frameworks. Weather Wise would gain broader reach and ensure consistent features when it expands to Android and iOS software platforms. The addition of wearable device support for smartwatches running Wear OS or Apple Watch platforms will deliver instant weather data as well as severe weather alerts through wrist notifications to users for more convenient protection. Users who enable voice assistant integration with Google Assistant, Alexa or Siri can ask for weather information through hands-free conversations because this feature supports accessibility needs and allows multitasking.

The implementation of artificial intelligence alongside machine learning systems represents a major opportunity to boost forecasting accuracy and user-specific relevance. Machine learning models that analyze historical data as well as environmental patterns can outperform traditional APIs in predicting weather situations at the hyper-local level. This model was able to generate customized forecasts through individualized delivery providing predictions that apply to users' daily schedules and their individual risk levels. The app's performance can be boosted by establishing alert features that allow users to create notifications based on definite weather conditions like wind speeds exceeding 30 km/h or UV index reaching elevated levels.

As a method to enhance user retention and engagement the application should incorporate gamified elements through achievements for daily weather monitoring and interactive community competitions and interactive weather feedback features. The app would gain community involvement potential when social sharing tools let users send both alerts and forecasts to friends through the platform. User-contributed reports that enable users to report rain, haze and high winds serve to improve real-time data accuracy while strengthening localized prediction power through crowd-sourced input.

The weather app should expand its inclusive features by adding multi-language interfaces that adapt both display language and regional measurement and date standards as well as weather descriptor terms. User-friendly accessibility components including screen reader functionality with adjustable font sizes coupled with high-contrast modes as well as simplified interface designs will make the app suitable for people who have visual or cognitive impairments. The implemented changes both fulfill regulatory requirements like WCAG standards and establish a wider digital environment beneficial to everyone.

The platform could develop its environmental features by providing in-depth information about long-term climate patterns which include multi-month rainfall and temperature tendencies along with annual weather extremes. Additional climate-related information would improve Researchers' and Farmers' and Planners' and Individual Decision Makers' use of the app to solve their seasonal and environmental challenges. The system would present crucial weather conditions to farmers through dashboards that show soil moisture status and frost threats as well as forecast the monsoon period.

The app needs offline functionality because it must work effectively even in areas with unreliable connectivity and emergency situations. The app should enable users to acquire radar maps as well as access emergency contact databases and send SMS-based notification during internet connectivity loss. Future research on Bluetooth mesh network and peer-to-peer data transfer technologies could enable alert notification without cellular connectivity particularly when power outages and natural disasters occur.

Future development for the Weather Wise project requires the creation of a smart weather interface that includes universal access and automatic customization options. Weather Wise possesses the capability to transform into a vital instrument for everyday users as well as professional and community-based users through its technological advancements and improved interface capabilities and platform enhancements.