SOFTWARE REQUIREMENTS SPECIFICATION

**For**

**Weather app with db**

**Prepared by**

*Mithun Balaaji*

*Mohammed Yaseen*

*Manoj Jegan*

# Introduction:

## Purpose:

The main objective of this document is to illustrate the requirements of the project weather app with DB. The document gives the detailed description of the both functional and non-functional requirements proposed by the client. The purpose of this project is to provide a friendly environment to maintain the details of weather- and weather-related notifications. The main purpose of this project is to maintain easy circulation system using computers and to provide different reports. This project describes the hardware and software interface requirements using ER diagrams and UML diagrams.

## Document Conventions:

* + - Entire document should be justified.
    - Convention for Main title

Font face: Satoshi

Font style: Bold

Font Size: 14

* + - Convention for Sub title

Font face: Times New Roman Font style: Bold

Font Size: 12

* + - Convention for body

Font face: Bahnschrift

Font Size: 10

## Scope of Development Project:

1. Top of Form

The Weather Forecasting Application aims to transform traditional weather monitoring into a web-based platform, empowering users to access real-time weather information, input data, and monitor changes effectively.

This project caters specifically to the needs of the general public, ensuring a user-friendly experience. Serving as a comprehensive user interface, the application allows ordinary users to seamlessly engage with weather forecasts. Its utility extends to any educational institution, facilitating easy content adaptation based on temperature fluctuations.

The application's adaptability under diverse circumstances is a standout feature. New features can be seamlessly incorporated as needed, ensuring flexibility across all modules. This inherent flexibility promotes reusability and scalability.

Java is the chosen programming language for this project due to its numerous advantages including superior performance, a rich array of tools, cross-platform compatibility, extensive libraries, cost- effectiveness (freely available), and a streamlined development process. These attributes collectively contribute to the efficiency and success of the Weather Forecasting Application

## Definitions, Acronyms and Abbreviations:

JAVA -> platform independence SQL-> Structured query Language ER-> Entity Relationship

UML -> Unified Modeling Language

IDE-> Integrated Development Environment SRS-> Software Requirement Specification

## Product Function:

Entity Relationship Diagram of Weather App With DB

## 

## 2.5 User Classes and Characteristics:

2.6 **Requirement Software Configuration: -**

* This software package is developed using java as front end which is supported by sun micro system. Microsoft SQL Server as the back end to store the database. Operating System: Windows 8 ,10,11.
* Language: Java 11(Amazon Corretto - 11 Version 11.0.21), IntelliJ Idea 2023.2.5, JavaFX
* Database: MySQL Server 8.0
* Connectivity: Weather API, ClimaCell API,

**Hardware Configuration: -**

* Processor: Dual core, Tetra-core CPU
* Hard Disk: 500 MB
* RAM: 256 MB or more2.9 Data Requirement

The Weather Forecasting Application retrieves user queries from a database, processing them to deliver comprehensive solutions. It furnishes users with current weather conditions, forecasts, and historical data, tailored to their specified parameters and preferences. The output comprises user-friendly, accurate weather information presented in a timely manner.

# External Interface Requirement:

3.1 GUI:

The software provides good graphical interface for the user and the administrator can operate on the system, performing the required task such as create, update, viewing the details of the weather app.

 It allows user to view quick reports in the weather at a particular time.

 It provides accurate information based on different criteria.

 The user interface must be customizable by the administrator

 All the modules provided with the software must fit into this graphical user interface and accomplish to the standard defined. The design should be simple and all the different interfaces should follow a standard template

 The user interface should be able to interact with the user management module and a part of the interface must be dedicated to the login/logout module

**Login Interface: -**

In case the user is not yet registered, he can enter the details and register to create his account. Once his account is created, he can ‘Login’ which asks the user to type his username and password. If the user entered either his username or password incorrectly then an error message appears.

**Search: -**

The member or user can enter the climatic conditions in a place he is looking for and then he can search for the required weather conditions by entering the place name.

**Category’s View: -**

Category’s view shows the categories of climatic conditions in that place.

**Admin’s Control Panel: -**

This control panel will allow Admin to add/remove users; add, edit, or remove a resource. And manage lending options.

**Other Non-functional Requirements:**

5.1 **Performance Requirement:**

The proposed system that we are going to develop will be used as the accurate weather system within the different Environment of the particular area which fetch data from the places. Therefore, it is expected that the database would perform functionally all the requirements that are specified by the App.

 The performance of the system should be fast and accurate

 The system should be able to handle large amount of data. Thus, it should accommodate high number of accuracy and users without any fault

**System Features for Weather Forecasting Application:**

The Weather Forecasting Application prioritizes the security and privacy of user accounts, ensuring confidence in the following ways:

**User Authentication and Validation:**

User authentication and validation for a weather forecasting application involve implementing secure mechanisms to verify and authorize user identities. This ensures that only authorized individuals can access the application's features and data. Techniques like password authentication, multi-factor authentication, and encrypted communication channels help authenticate users while safeguarding their information. Validation processes ensure that the entered user data, such as login credentials or personal information, meets the application's predefined criteria, enhancing security and preventing unauthorized access or misuse.

3.2 **Security Requirement:**

* When using our weather forecasting application, your security and privacy are our top priorities. Your information remains safe through several measures. All data, including your personal information, is encrypted while being transmitted or stored, preventing unauthorized access. You'll securely access the app through multi-factor authentication, adding an extra layer of protection to your account. Our app also follows the latest security standards in its development to guard against common vulnerabilities. We regularly update the app to address any discovered security issues and keep your information safe. Additionally, we monitor user activities, maintaining logs to detect and respond to any potential threats. Our commitment includes compliance with data protection regulations to safeguard your privacy. Your security matters to us, and we're dedicated to ensuring a safe and reliable experience while using our weather forecasting application.

3.3 **Requirement attributes:**

 When you use our weather app, we've made sure it's got everything you need for a great experience. You'll find the weather forecasts and updates to be really accurate, giving you reliable information whenever you check. Even when a lot of people are using it, the app works smoothly, so you can always count on it being available when you need it. It's designed to be really easy to use, with a simple interface that helps you find the weather info you want without any hassle. And don't worry, your information is kept safe with us—we've made sure that only you can access your account and that your data is secure. You can use the app on different devices, and you can even customize it to show you exactly the weather details you're most interested in. We've made sure to follow all the rules and regulations, so you can trust that everything's legal and done the right way.

3.4 **Business Rules:**

* Our weather forecasting application operates under important guidelines to ensure a reliable and effective experience for you. We prioritize using trustworthy sources for weather data, ensuring the information you receive is accurate and dependable. Your privacy is crucial, and we strictly adhere to regulations to safeguard your data, ensuring transparency in how we handle your information. We're committed to providing you with timely updates and forecasts, so you can make informed decisions. As our user base grows and technology advances, our system is designed to effortlessly adapt, ensuring consistent and high-quality performance to meet your needs.

**Other Requirements for Weather Forecasting Application:**

4.2 **Appendix:**

**Appendix Sections:**

The document includes an appendix (Appendix A to Appendix U) covering a range of topics, including administrators, abbreviations, acronyms, assumptions, books, business rules, class, client, conventions, data requirement, dependencies, GUI, key, non-functional requirement, operating environment, performance, perspective, purpose, requirement, requirement attributes, safety, scope, security, system features, user, user class and characteristics, and user requirement.

4.3 **Glossary:**

**Conventions and Acronyms:**

The glossary provides a list of conventions and acronyms used in the document and the project. Key terms include Administrator, User, Client, SQL, SQL Server, Layer, User Interface Layer, Application Logic Layer, Data Storage Layer, Use Case, Class Diagram, Interface, Unique Key, etc.

4.4 **Class Diagram:**

**Overview:**

In our weather forecasting app, the class diagram acts like a blueprint, outlining how different parts work together behind the scenes. The User class manages your account details, while Location stores information about different places. When you check the weather, Forecast gives you predictions based on specific locations, and Historical Data shows past weather records. Your preferences, like your favorite locations or temperature units, are handled by the User Preferences class, making sure you get the weather info you want. These classes all work together to make sure you get accurate forecasts and personalized weather details whenever you need them.

**Main Classes:**

* User Class: This class manages your account details, including your username, password, email, and the type of access you have within the app.
* Location Class: It stores information about different places such as cities and countries, ensuring accurate weather forecasts for your chosen areas.
* Forecast Class: This class predicts the weather, giving you details like temperature, humidity, wind speed, and expected weather conditions for specific dates and locations.
* Historical Data Class: It holds past weather records, allowing you to review and compare weather data from earlier dates and locations.
* User Preferences Class: This class handles your personal settings, like your favorite locations, preferred temperature units, and how you receive notifications about weather updates.

As for the appendix, it could be customized for the weather app database, listing abbreviations and acronyms specific to meteorology and database management. For instance:

A: Admin, API, Assumptions;

C: Current Weather, Client, Conventions;

F: Forecast, Functional Requirement;

H: Historical Data, Humidity;

S: System features, SQL, SQL Server;

U: User, User class characteristics, User requirement;

W: Wind Speed.

The glossary for the weather app could explain terms and acronyms commonly used in meteorology and database systems specific to this application. For example:

- Admin: A login ID representing a user with administration privileges for the weather app.

- User: A general login ID assigned to most users of the weather app.

- SQL: Structured Query Language used to retrieve information from the weather app database.

- Data Storage Layer: The section of the weather app database where all meteorological data is stored.

- Unique Key: A data attribute used to differentiate entries in the weather app database.

The class diagram for the weather app with a database could illustrate the structure of the data and relationships between different types of weather-related information. It would include classes for weather data, user profiles, administrative functions, and potentially external data sources. The relationships between these classes would help visualize how the various components of the weather app database interact with each other.

**Class Diagram:**