**Remote Monitoring Weather System**

(under PBL)

Group – 4 [s3]

**[Software Requirements Specification]**

SE AIDS 23-24

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Name** | **Roll No.** |
| 1 | Ishita Kadam | 2077 |
| 2 | Vedant Banaitkar | 3101 |
| 3 | Pradnya Gore | 3102 |
| 4 | Krishna Shah | 2086 |

Version – 3.0

(06/03/2023)

**Table of Contents**

1. Introduction……………………………………………………………………………………….2

1.1 Purpose

1.2 Scope

1.3 Intended Audience (Users)

1. System Overview…………………………………………………………………………..…..6
2. Functional Requirement………………………………………………………………..…..6
   1. Severe Weather Notifications
   2. Location Tracking
   3. Login System
3. Non-Functional Requirement…………………………………………………………….7
   1. Performance
   2. Reliability
   3. Usability
   4. Scalability
   5. Compatibility

**Introduction:**

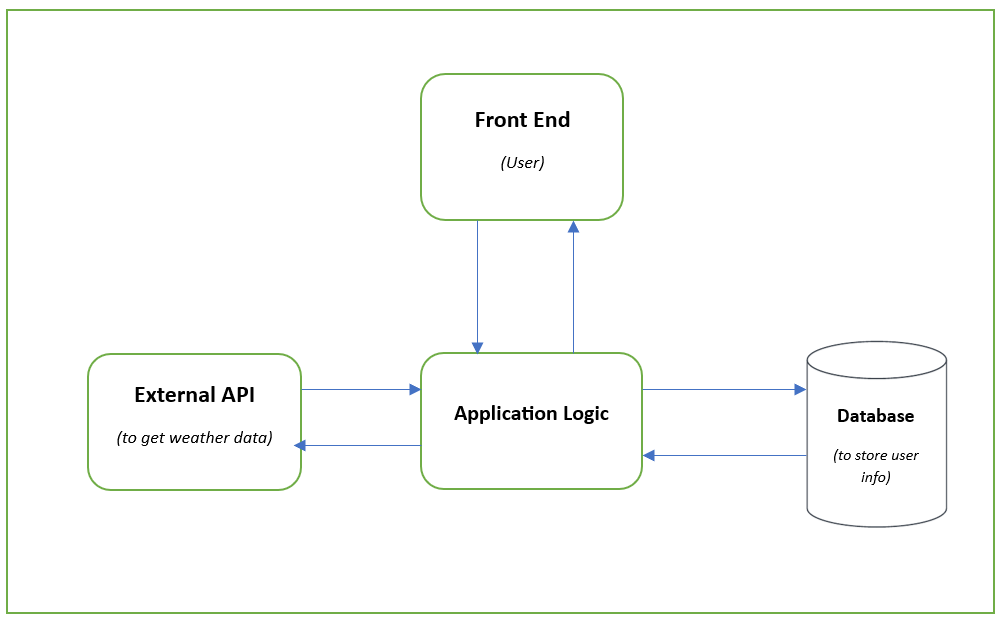
**1.1 Purpose**

The purpose of software Requirement specification is It provides a clear and formalized document that communicates the requirements of the software system to all stakeholders, including clients, developers, testers, and project managers. This ensures that everyone involved has a common understanding of what the software is supposed to do.

**1.2 Scope**

The Software Requirements Specification (SRS) document for the weather monitoring project outlines the objectives and boundaries of the application. It encompasses detailed functional and non-functional requirements, integration with external interfaces, constraints, assumptions, dependencies, and appendices containing glossary and references for comprehensive project documentation.

**1.3 System Architecture:**

****

**Web Service**

1. **Front End (User Interface)**:
   * The **Front-End** component represents the user-facing part of the application.
   * It is where users interact with the system.
   * Users input data, view results, and perform actions through this interface.
2. **Application Logic**:
   * The **Application Logic** layer is the heart of the system. It processes data, performs business logic, and orchestrates interactions between different components.
   * Responsibilities:
     + **Data Processing**: It handles requests from the Front End, processes data, and communicates with external services (like the weather API).
     + **Business Rules**: It enforces business rules, validations, and workflows.
     + **Integration**: It interacts with the External API to fetch weather data.
     + **User Authentication/Authorization**: If applicable, this layer manages user authentication and authorization.
3. **External API (to get weather data)**:
   * The **External API** component represents an external service that provides weather data.
   * It could be a third-party weather API (e.g., OpenWeatherMap, AccuWeather) or a custom service.
4. **Database (to store user info)**:
   * The **Database** component stores persistent data related to the application.
   * In our case, it stores user information.
   * Used to store user information that is used to authenticate users and stores data related to what services the user has subscribed

**1.4 Intended Audience (Users):**

The intended audience includes individuals who require weather forecasts for planning activities, project manager, marketing people, stake holders, travellers, outdoor enthusiasts, and professionals in various industries.

**2. System Overview**

This Remote Monitoring of Weather System will consist of the following main components:

**2.1 API Interface**:

This component provides a standardized interface for accessing weather forecast data. It defines endpoints and protocols for communication between the weather API and client applications

**2.2 Data Processing Module**:

This module is responsible for cleaning, filtering, and formatting the raw data obtained from the data sources. It may involve techniques such as data normalization, interpolation, and quality control to ensure the accuracy and reliability of the data.

**2.3 Real-time Updates**

The system should support real-time updates to deliver the latest weather information to users 24 by 7 hours.

**3. Functional Requirements**

* 1. **Severe Weather Notifications:**
* Users should receive timely notifications for severe weather events such as thunderstorms, hurricanes, tornadoes, and other hazardous conditions.

* 1. **Locations tracking:**
* Users must be able to input or select multiple locations to view weather forecasts for different places, such as their current location, hometown, or travel destinations.
  1. **User Authentication/Login System:**

**-**The app must include a user authentication/login system, where users can create accounts with their credentials (e.g., email, password) to access personalized features.

**4. Non-Functional Requirements**

**4.1 Performance**

* Quick response times for weather updates and forecast retrieval are crucial for providing users with timely and relevant information.
* The system should respond to user requests within 3 seconds under normal load conditions.
* Optimizing performance ensures that users can access weather data efficiently without experiencing significant delays.

**4.2 Reliability**

* High availability of weather data is essential to ensure that users can rely on the system for 24/7.

**4.3 Usability**  
A remote monitoring weather system project can be highly usable and beneficial in various contexts:

* **Agriculture**: Farmers can monitor weather conditions remotely to make informed decisions about planting, irrigation, and harvesting
* **Transportation:** Airlines, shipping companies, and transportation authorities can use weather data to plan routes, avoid hazardous conditions, and ensure the safety and efficiency of their operations.
* **Renewable Energy:** Companies and governments investing in renewable energy sources like solar and wind power can optimize their energy production by monitoring weather patterns and predicting energy output.

**4.4 Scalability**

* The system should be able to handle a minimum of 1000 concurrent request in a day.
* Scalability allows the system to maintain optimal responsiveness user demand and data volumes increase over time.

**4.5 Compatibility:**

**-** The web interface should be compatible with modern web browsers (Chrome, Firefox, Safari).

**-** Supporting multiple platforms and devices ensures broad accessibility for users across different operating systems and device type.