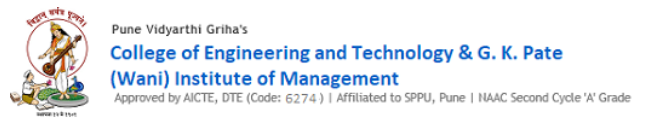
****

**Remote Monitoring Weather System**

(under PBL)

Group – 4 [s3]

**[Functional 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**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Topic** | **Page No.** |
| **1.**   * 1.1 * 1.2 * 1.3 * 1.4 * 1.5 | **Introduction**  Purpose  Scope  Background  Users | 3 |
| **2.**   * 2.1 * 2.2 * 2.3 * 2.4 | **Data Flow Diagram**  FrontEnd  BackEnd  Notification Service  OpenWeather API | 4 |
| **3.**   * 3.1 * 3.2 | **Overview of Document**  Functional Objectives  Non-Functional Objectives | 5 |

**1. Introduction**

**1.1 Purpose of FRS:**

The FRS helps align expectations among all stakeholders, including meteorologists, software developers, project managers, and end-users. It ensures everyone has a common understanding of the system’s capabilities and limitations and that developers understand the functionalities that the system must support.

**1.2 Scope of FRS:**

It outlines what the system will do (and sometimes what it will not do) in terms of collecting, analysing, and presenting weather data and. encompasses the boundaries and limitations within which the system will operate, as well as the functionalities it will provide.

**1.3 Background**:

Our team is trying to make a free remote monitoring weather system for better knowing and understanding about the threats and weather behaviour pattern of a place for interested users.

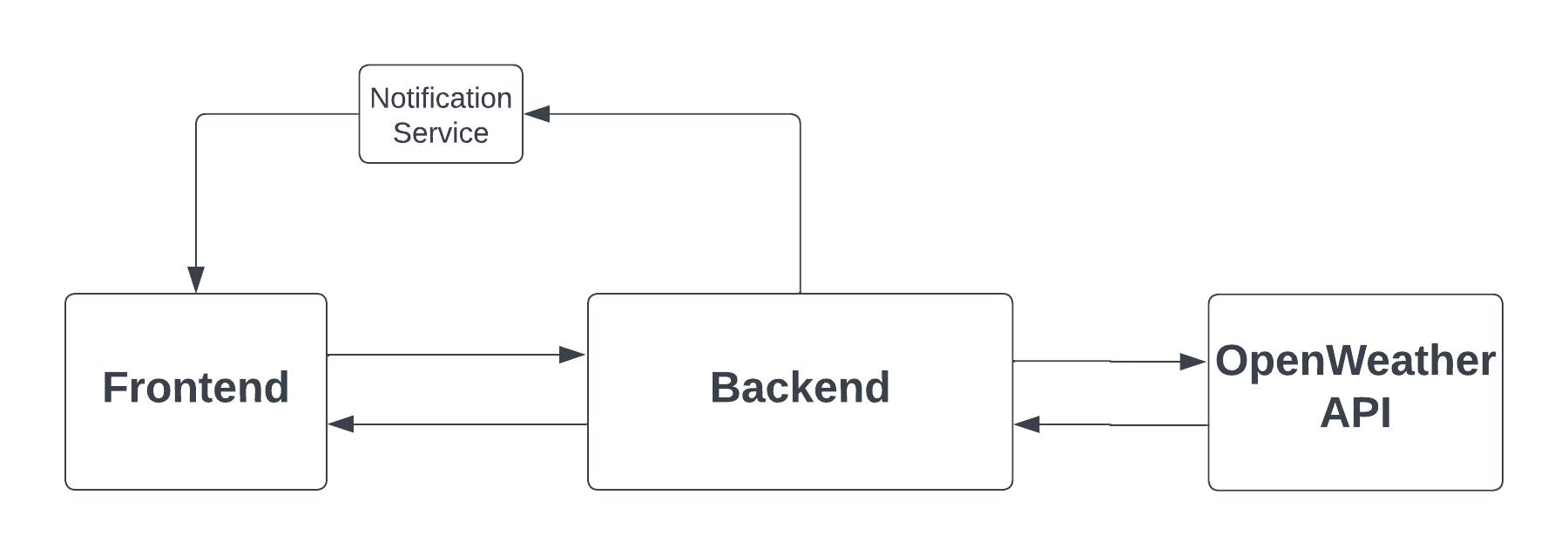
**1.4 Users:**

General public, outdoor enthusiasts, travellers, weather scientists, weather forecasters, news channels and many more.

**1.5 Need:**

To assist users in planning their activities based on weather conditions. Also to know the threats and conditions of the pattern of the weather in the upcoming short future i.e. some days.

1. **Data flow diagram**



* 1. **Frontend:**
  + The **Frontend** component represents the user interface of your system.
  + Users interact with the application through this interface.
  + It sends requests to the Backend and receives responses.
  1. **Backend:** The **Backend** is the core processing component Responsibilities:

**Data Processing**: It handles requests from the Frontend and processes data.

**Business Logic**: It enforces business rules, validations, and workflows.

**Integration**: It communicates with external services (like the Open Weather API).

**Database Interaction**: It stores and retrieves data from the database.

**User Authentication**: It also helps to authenticate users, and allow authorized user to access appropriate data.

* 1. **Notification Service:** The **Notification Service** is a separate component connected to the Backend.

It likely handles notifications (e.g., email alerts, push notifications) based on specific triggers or events.

Notification can be pushed to user if user wishes to subscribe to daily or hours alerts or alerts in case of extreme weather conditions.

* 1. **Open Weather API:** The **Open Weather API** is an external service. The Backend communicates with it to fetch weather data.

**3. Overview of Document:**

**3.1Functional Objectives:**

It describes the specific functionalities that the system must have to accurately predict weather conditions and provide valuable information to its users.

**3.1.1 Weather Data Retrieval:**

* + Obtain real-time weather information from external sources or APIs.
  + Display current weather conditions, including temperature, humidity, wind speed, and precipitation.
    1. **Location-Based Search:**
  + Allow users to search for weather data based on location, such as cities or geographic coordinates.
  + Provide autocomplete suggestions for location search queries.

**3.1.3 Detailed Weather Display:**

* + Present detailed weather forecasts, including hourly and daily forecasts.
  + Display weather maps, radar images, and satellite imagery for enhanced visualization.
    1. **User Account Management:**

Enable users to create accounts, log in, and manage their profiles.

Provide options for users to customize their preferences, such as preferred units of measurement and notification settings.

**3.2 Non-Functional Objectives:**

**3.21. The Context Model:**

**• Goal Statement:** To provide users with accurate and timely weather forecasts.

**• Context Diagram:** Illustration of the weather forecasting interactions with external systems and users.

**• System Externals:** Identification of external entities interacting with the it.

**3.2.2 Reliability:**

* Specifies how dependable the system should be, including measures for uptime and failure recovery.

**3.2.3 Usability:**

* Intuitive user interface for easy navigation.
* **Can be easily accessed by user on web browser.**

**3.2.4 Security:**

* Outlines measures to protect the system and its data from unauthorized access or malicious attacks.
* Protection of user data and secure transmission of information.