SmartWeather

Requirement Specification Document

Table of Contents

Table of contents... 1

1. Introduction.. 2

1.1 Purpose... 2

1.2 Intended Audience.. 2

1.3 Scope... 2

2. System Requirements.. 3

2.1 Functional Requirements.. 3

2.2 Non-Functional Requirements... 4

1. Introduction

1.1 Purpose

SmartWeather is an application designed to provide users with real-time weather updates, forecasts, and severe weather alerts. The application aims to deliver accurate meteorological data such as temperature, humidity, wind speed, and precipitation. By integrating a reliable weather API, SmartWeather ensures that users receive up-to-date weather information for their specific location.

The primary goal of SmartWeather is to offer an **intuitive, fast, and user-friendly experience**, making weather tracking accessible to everyone, including travelers, outdoor workers, and general users seeking accurate forecasts. See section 1.2 below for more information on the intended audience.

1.2 Intended Audience

SmartWeather can be used by anyone looking to stay up-to-date with weather information while having a user-friendly experience. Individuals who the app is intended for are as follows:

- Travelers
- Outdoor Workers
- Outdoor enthusiasts
- Commuters
- Farmers
- Many more!

Updated SRS III- Matt Laskowski

2. System Requirements

2.1 Functional Requirements

- The application shall allow users to search for weather by using their location or searching for a city.
- The application shall display real-time weather data, including:
    - Temperature in Celsius
    - Temperature in Fahrenheit
    - Wind speed in km/h and mph
    - Humidity percentage
    - Precipitation level in mm and inches
- The application shall provide a 7-day weather forecast, including:
    - Daily high and low temperatures
    - Expected precipitation percentage
    - Wind speed forecast
    - General weather conditions (e.g., sunny, cloudy, rainy)
- The application shall display an hourly forecast for up to 24 hours, including:
    - Hourly temperature
    - Hourly precipitation chance
    - Hourly wind speed in km/h and mph

- The application shall provide severe weather alerts for users when issued by weather authorities.
    - Users shall be able to enable or disable specific types of notifications.
    - Users shall be able to customize the threshold for temperature change alerts (e.g., notify when the temperature drops below 32°F).
    - The application shall provide vibration and sound options for notifications.
- The application shall allow users to toggle between Celsius and Fahrenheit for temperature display.
    - Users shall be able to switch between Celsius and Fahrenheit from the settings menu.
    - The application shall remember the user’s preferred temperature unit across sessions.
- The application shall provide a feels-like temperature based on humidity, wind speed, and air pressure.
    - The application shall display both the actual and feels-like temperature side by side for user comparison.
    - Feels-like temperature shall update dynamically based on real-time weather data.
- The application shall display UV index levels to inform users of sun exposure risks.
    - The UV index shall be displayed numerically (e.g., 0–11+) with an associated risk level (Low, Moderate, High, Very High, Extreme)
    - The application shall provide color-coded indicators to help users quickly assess UV risk.
    - The UV index shall be updated in real-time based on the user’s location.
- The application shall display air quality index (AQI) levels for a user’s location, categorized as good, moderate, unhealthy, and hazardous.
    - The AQI shall be displayed numerically, following standard AQI scales (e.g., 0–500).
    - The application shall categorize AQI levels into at least five risk categories (e.g., Good, Moderate, Unhealthy for Sensitive Groups, Unhealthy, Hazardous).
    - Users shall receive alerts when AQI levels reach unhealthy thresholds.

2.2 Non-Functional Requirements

1. **User Interface and Experience**
   - The application shall have a user-friendly UI with smooth navigation.
   - The UI shall be visually appealing, accessible, and responsive across various devices and screen sizes.
   - The application shall support dark mode for user preference.
2. **Data Accuracy and Update Frequency**
   - Weather data shall be updated from the API every 10 minutes for accuracy.
   - The system shall fetch real-time updates without noticeable delays.
   - The application shall guarantee consistency of weather data across different views.
3. **General Performance**
   - The system shall support multiple simultaneous users.
   - Response time for fetching weather data shall be less than 2 seconds under normal conditions.
   - The application shall handle increased traffic during severe weather conditions without crashing.
4. **Reliability and Availability**
   - The application shall maintain 99.9% uptime to ensure availability.
   - The system shall include failover mechanisms to prevent downtime.
   - The application shall provide error messages and fallback options if the API is unreachable.
5. **Security and Privacy**
   - User location data shall be encrypted and never shared without consent.
   - The application shall comply with GDPR and other data privacy regulations.
   - The system shall use secure API connections (HTTPS) to prevent data breaches.