3/12/2024

***WeatherForecasting application***

By Faisal and Faraz

**Software Requirements Specification (SRS)**

**Project Title:** Weather Forecasting Application  
**Group Members:** Faisal and Faraz  
**Project Type:** Data Structures and Algorithms (DSA)

**1. Introduction**

**1.1 Purpose**

The Weather Forecasting Application is a Java-based project designed to demonstrate practical usage of DSA concepts while providing real-time weather data for a user-specified location. It integrates backend logic and GUI components to retrieve and display weather details such as temperature, humidity, windspeed, and weather conditions using APIs.

**1.2 Scope**

This project aims to:

* Utilize data structures like arrays, JSON objects, and parsing techniques to process and organize data.
* Fetch weather data in real-time from the Open-Meteo API based on user input.
* Display weather information via an intuitive GUI using Swing.
* Provide a practical understanding of combining API integration with DSA concepts.

**1.3 Target Audience**

* Students and instructors in computer science courses focusing on DSA.
* Developers interested in understanding API-based weather applications.

**1.4 References**

* **API Documentation:** Open-Meteo API
* **Java Libraries:** JSON-Simple, Swing

**2. System Features**

**2.1 Core Features**

1. **Location-Based Weather Retrieval**
   * Users input a location to fetch weather details (temperature, humidity, windspeed, weather condition).
2. **Real-Time Data Integration**
   * Data is fetched from the Open-Meteo API dynamically using HTTP GET requests.
3. **Graphical User Interface (GUI)**
   * Interactive GUI for user input and data display using Java Swing components.

**2.2 Additional Features**

* Error handling for invalid locations or API connectivity issues.
* Weather icons and visuals to enhance the user experience.

**3. Functional Requirements**

**3.1 Backend Functionality**

1. The application should fetch geolocation data (latitude and longitude) based on user input.
2. Weather data for the current hour should be retrieved using API responses.
3. JSON data must be parsed and transformed into readable weather information.

**3.2 Frontend Functionality**

1. The GUI should accept location input and trigger backend functionality upon user action.
2. Weather details must be displayed with appropriate icons and labels.

**4. Non-Functional Requirements**

1. **Performance**
   * Response time for API calls should be under 5 seconds.
2. **Scalability**
   * Designed to handle input for any location globally.
3. **Usability**
   * GUI should be simple and intuitive for users with minimal technical knowledge.
4. **Maintainability**
   * Code should be modular for ease of debugging and enhancement.

**5. Data Structures Used**

1. **Arrays**:
   * For storing API response data (e.g., time, temperature, humidity).
2. **JSON Objects and Arrays**:
   * For parsing and managing hierarchical weather and location data.
3. **Custom Object Mapping**:
   * Objects representing weather data for streamlined GUI integration.

**6. Hardware and Software Requirements**

* **Hardware**:
  + Minimum 4GB RAM, 500MB free disk space.
* **Software**:
  + Java Development Kit (JDK) 8 or later.
  + JSON-Simple Library for API integration.
  + Swing for GUI components.

**7. Constraints**

1. Requires internet connectivity for API calls.
2. Dependent on third-party Open-Meteo API for weather data.

**8. Future Enhancements**

1. Add support for weekly or daily weather forecasts.
2. Implement alerts for severe weather conditions.
3. Support for multiple languages.

**Prepared By:**

**Faisal and Faraz**  
*Group Members for the Weather Forecasting Application Project (DSA).*