Requirement Specification For Weather Forecast Application

The system is a web application that provides weather forecast data for cities worldwide. The application fetches and processes weather data from OpenWeatherMap API, presenting users with average weather conditions for the next 40 days.

Supplementary Specification

System Quality Requirements

- The system should present weather data in a clear, visually appealing format
- The system should provide helpful error messages for invalid inputs
- The system should follow modern web development standards
- The system should be secure
- The system should handle international characters gracefully

Other Considerations

- Support for non-Latin characters (e.g., Chinese: 北京, German: München)
- The system should support international country codes
- API key storage in environment variables

UC1 Get Weather Forecast

Main Scenario

1. Starts when a user wants to check weather forecasts
2. System asks for city name and country code
3. User provides valid city name and country code
4. System fetches and displays the weather forecast data:
    - Average temperature in Celsius
    - Average humidity percentage
    - Average wind speed in m/s
    - Maximum rainfall in mm

Alternate Scenarios

3a. User enters invalid or incomplete location data

1. System displays an error message
2. Return to step 2 in main scenario

3b. User enters non-Latin characters for city name (e.g., Chinese: 北京, German: München)

1. System should accept and process the input
2. System retrieves and displays weather data

3. System may display an error message despite successful data retrieval
    - Known Issue: Error handling needs improvement for non-Latin characters

4a. Weather data cannot be retrieved

1. System displays appropriate error message
2. Return to step 2 in main scenario

UC2 Process Weather Data

Preconditions

Valid city coordinates obtained from location input

Main Scenario

1. Starts when system receives raw weather data
2. System calculates average temperature, humidity, wind speed, maximum rainfall, over 40 days
3. System presents processed data in formatted cards

UC3 Convert Temperature

Main Scenario

1. Starts when system receives temperature in Kelvin
2. System converts temperature to Celsius
3. System rounds result to one decimal place
4. System displays converted temperature

Alternate Scenarios

1a. Invalid temperature value received

1. System displays error message
2. Returns to UC1

UC4 Location Validation

Main Scenario

1. Starts when user submits location
2. System validates city name format
3. System validates country code format
4. System confirms location exists in database
5. System proceeds with weather data retrieval

Alternate Scenarios

2a. Invalid city name format

1. System displays specific error message
2. Return to UC1 step 2

3a. Invalid country code format

1. System displays specific error message
2. Return to UC1 step 2

4a. Location not found

1. System displays location not found message
2. Return to UC1 step 2

Technical Requirements

1. React.js for frontend development
2. OpenWeatherMap API integration
3. Modular weather data processing component
4. Error handling for API and data processing
5. Responsive design for all screen sizes
6. Browser compatibility (Chrome, Firefox, Safari, Edge)