Requirement Specification ReportGroup 8

By: Bilal Ahmed, Jeth Balabagno, Hazzem Cordash, Elijah Kruse, Timothy Xiong, Julian Moreno-Johns

1. Introduction:

Scope of the weather dashboard:

In general, our group is providing an easy to use, accurate, and complete weather forecasting application. Some of our key features are listed below

- Real time weather updates
    - This displays real time weather data from available free online APIs
- Interactive global weather maps
    - We are going to integrate using advanced mapping libraries
- Weather for trip prediction/calculations
    - Predict weather information up to 24 hours for the locations planned on travel to.

Key Objectives:

Our main objective is to provide users with access to up-to-date weather information and forecasts. Our application will allow businesses to make real-time decisions, give individuals up-to-date weather information, ensure the safety of individuals and communities, provide environmental scientists and researchers a user-friendly dashboard, and give users accurate weather information from around the world. The application should also provide weather predictions up to 24 hours to and from a location that they provide.

Concerns:

Front-end Challenges:

- Data Visualization: Displaying real-time data in a user-friendly manner.
- Responsive Design: The app should look good on all device sizes.
- Real-time Updates: Pushing real-time updates to the client.

Back-end Challenges:

- Data Collection: Gathering real-time data from various APIs can be a hassle.
- Data Storage: Logging historical data, we'll need a strategy for how and where to store it.
- Concurrency: Need to handle multiple requests simultaneously.

2. Program Requirements

Frontend requirements:

The frontend of our weather dashboard will be designed to provide users with an interactive experience while accessing real-time weather updates. Given the vital role of user experience in creating an application, following are several key frontend functionalities;

1. *User Interface design*

An intuitive interface will be developed that will ensure users can consistently and efficiently use the app regardless of their platform.

- Key weather metrics (temperature, humidity, precipitation) will be prominently displayed

2. *Interactive Weather maps*

The application will integrate advanced mapping libraries to visually represent global weather conditions in a clear manner.

- Features such as zooming, panning, and clicking on specific regions to retrieve their weather statistics

3. *Trip Prediction Feature*

A unique feature that will allow users to input their travel route, for example;

Milwaukee -> Chicago

to get weather updates for multiple points along their journey. This will aid travelers in anticipating weather changes during their trips, allowing for them to make necessary preparations on the go.

4. *Responsiveness*

All features, including the interactive, will be fully functional regardless of the device and its size.

5. *Data Visualization*

Weather data, which will be fetched in real time, will be presented using charts and graphs. Visual elements like temperature scales or icons representing different weather conditions will be implemented for quick comprehension.

6. *Feedback integration*

A consistent feedback integration loop between our backend team ensures that we effectively showcase and optimize all the data they provide.

7. *Notification & Alerts*

In important or extreme situations when there are significant weather updates (or warnings), real-time notifications will be pushed to active users. This can be set in their preferences.

Backend requirements:

Creating a weather forecast app involves various components. The backend, in particular, manages data retrieval, storage, and processing. This involves establishing seamless integrations with third-party weather data providers ensuring the retrieval of accurate and current forecasts. It's a continuous cycle of fetching, storing, and processing.

1. *API Integration*

We need to integrate with one or more third-party weather data providers like OpenWeatherMap, WeatherStack, or AccuWeather to fetch the forecast and other relevant details. We need to set up API endpoints to give weather data to the frontend based on time and location.

2. *Database*

The app will store some weather data for the trip prediction feature so this will require a good and reliable database system like MongoDB. We can also look at and track analytics of user interaction.

3. *Data Processing*

Process raw data from the third-party provider to extract and send only the necessary details to the frontend. Implement some logic to interpret data for the true prediction feature.

4. *Performance*

Design the back to handle spikes in traffic during severe weather events. Look at potentially a cloud base solution which can really change up to scalability and performance, auto-scaling resources and such. We need to be able to handle multiple tasks or requests simultaneously.

3. Data Requirements:

Most of the data can be processed into object classes. For example:

- Location Data:
    - Location Name (String)
        - Identifies the place for the requested weather data
    - Timezone (String)
        - Ensures proper time representation for the given location

  *Note that we could use a hash map or dictionary to map the location to its timezone

- Weather Information:
    - Temperature(F or C)

        - Mapped to location and time, simple method will be implemented to change between Fahrenheit and Celsius (default will depend on the location)
    - Description(string):
        - Textual representation of weather conditions
    - Chance of Rain (and severe weather) (double):
        - Percentage, represents the probability of rainfall
    - Time (double array mapped to Temperature):
        - Ensures correct temperature data for the given time

- User Class:
  General login information, we could put more saved user settings in here as well.
    - Username (String)
    - Password (String)
    - First Name(String)
    - Last Name (String)

4. Life Cycle Requirements:

The app would have plenty of primary users and also a team to maintain the app. These users include the general public, travelers, businesses, event planners, media, and environmentalists. The maintenance team would ensure the app is functional. Users should have access to functions listed below:

1) Navigate to application.
2) Enter username and password. This should take the user to the main page if the username and password are in the database.
3) The main page should have a dashboard, displaying their location, time and weather for the next 24 hours.
    a) Weather features displayed include
        i) Temperature (Both F and C)
        ii) Chance of raining/snowing
        iii) Humidity
        iv) Wind
4) The main page should also show a map of their location. This map shows the user's state and the relative temperatures surrounding his location.
5) The main page should also have a button called "Plan". Once a user clicks the button, they will be redirected to the plan page.
6) The plan page should be able to take input from multiple locations and display the weather for those locations.
    a) A user should be able to search these locations using
        i) Zip code
        ii) City Name

7) The plan page should also be able to plan a trip from a location, to a location. A user can enter these locations and the app displays the weather in between the two locations.