Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

REAL TIME WEATHER DASHBOARD

CSI1007 – Software Engineering Principles Laboratory

BHARANI KUMAR.J(23MIC0039)

NIDEESH.C.G(23MIC0092)

DHARMA.S(23MIC0104)

REAL TIME WEATHER DASHBOARD

Description of the Project: A **Real-Time Weather Dashboard** is a web-based application designed to provide users with up-to-date and accurate weather information for their selected locations. The dashboard is powered by weather APIs, offering features like real-time updates, extended forecasts, and customizable settings, presented in an intuitive and visually appealing interface.

REAL TIME WEATHER DASHBOARD

Sope of the Project: To develop a real-time, user-friendly weather dashboard that provides accurate, up-to-date weather information and forecasts, catering to individual and business needs through a responsive and visually appealing interface, powered by reliable data sources.

REAL TIME WEATHER DASHBOARD

Impact of the developing Project: Economic Impact

✅ Boosts local, creates jobs, generates revenue, and reduces operational costs.

❌ High initial development and maintenance costs.

REAL TIME WEATHER DASHBOARD

Impact of the developing Project: Social Impact

✅ **Enhances safety, convenience, community awareness, and education.**

❌ **Risks over-reliance on technology and excludes those without digital access.**

REAL TIME WEATHER DASHBOARD

Impact of the developing Project: Technological Impact

✅ Drives innovation, scalability, data-driven decision-making, and knowledge sharing.

❌ Dependent on third-party APIs and vulnerable to security risks.

REAL TIME WEATHER DASHBOARD

Impact of the developing Project: Environmental Impact

✅ Aids disaster mitigation, resource optimization, and climate awareness.

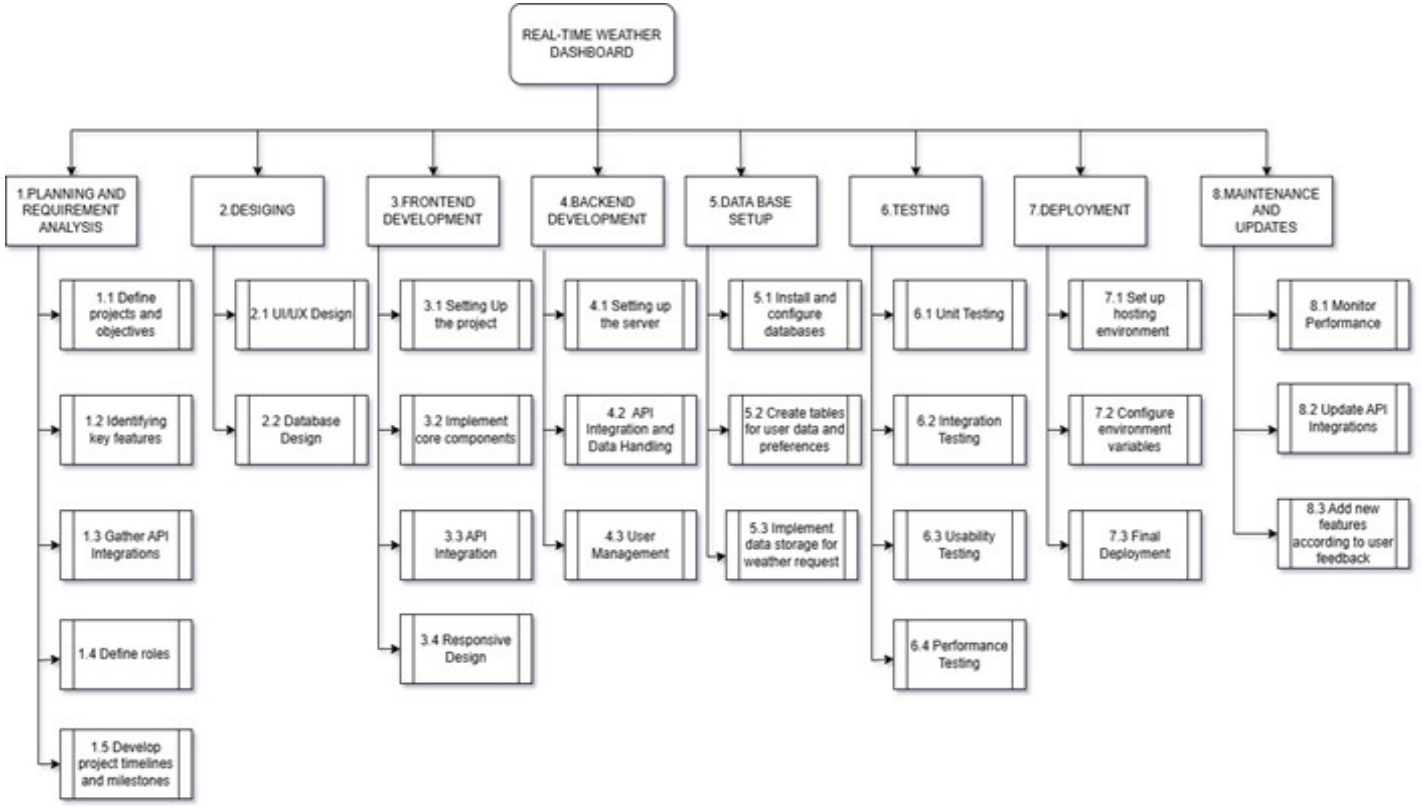
❌ High energy consumption and potential electronic waste.

✅ Aids disaster mitigation, resource optimization, and climate awareness.

❌ High energy consumption and potential electronic waste.

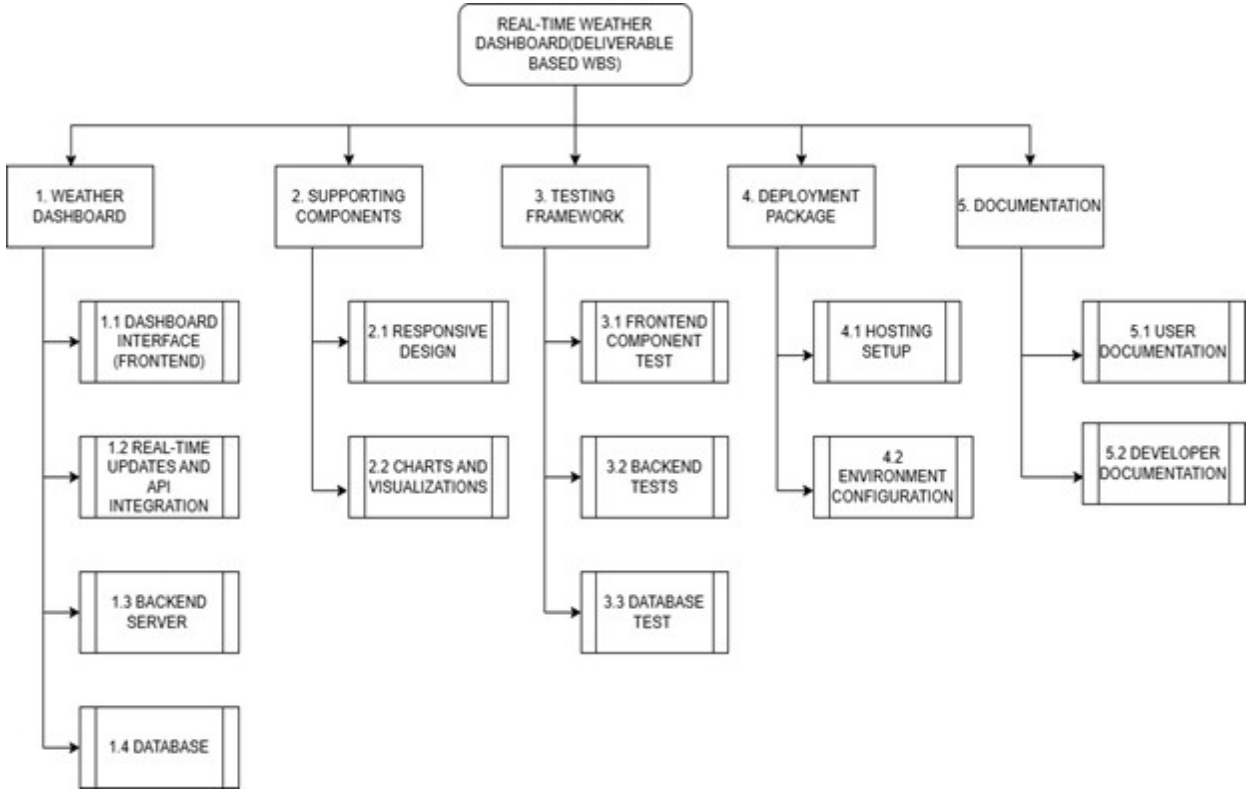
Work Breakdown Structure

- **PROCESS-BASED WORK BREAKDOWN STRUCTURE:-**



Work Breakdown Structure

PRODUCT-BASED WORK BREAKDOWN STRUCTURE:-



SRS Document

- **FUNCTIONAL REQUIREMENTS:-**

1. Real-Time Weather Data Display
    - The system shall fetch and display real-time weather information, including:
        - Temperature (Celsius/Fahrenheit)
        - Humidity (%)
        - Wind speed (km/h or mph)
        - Atmospheric pressure (hPa)
        - Weather conditions (e.g., sunny, cloudy, rainy)
2. Forecast Information
    - The system shall provide hourly and daily weather forecasts for up to 7 days.

SRS Document

- **FUNCTIONAL REQUIREMENTS:-**

3. Weather Alerts

  • The system shall notify users of severe weather events (e.g., storms, floods, hurricanes) via pop-up notifications or messages.

4. Search Functionality

  • The system shall allow users to search for weather information by:

    ◦ City name

    ◦ Region

    ◦ GPS-based location detection

SRS Document

- **FUNCTIONAL REQUIREMENTS:-**

5. Customization Options
    - The system shall allow users to customize the dashboard by:
        - Selecting preferred weather metrics (e.g., wind direction, UV index)
        - Choosing units of measurement (e.g., Celsius/Fahrenheit, km/h/mph)
        - Setting a preferred language

6. Location Management
    - The system shall allow users to:
        - Save multiple locations for quick access
        - Set a default location for weather updates

SRS Document

- **FUNCTIONAL REQUIREMENTS:-**

7. Multilingual Support
    - The system shall support displaying weather information in multiple languages.

8. Responsive Design
    - The system shall adapt its layout and functionality for devices including:
        - Desktops
        - Tablets
        - Smartphones

9. Data Refresh Interval
    - The system shall automatically refresh weather data every 5 minutes to ensure accuracy.

SRS Document

- **FUNCTIONAL REQUIREMENTS:-**

10. Weather Map Integration
    - The system shall provide an interactive weather map displaying precipitation, temperature zones, and wind patterns.

11. API Integration
    - The system shall integrate with third-party weather APIs to fetch accurate real-time data.

12. Error Handling
    - The system shall display appropriate error messages in case of:
        - API unavailability
        - Network connection issues
        - Invalid location search queries

SRS Document

- **NON - FUNCTIONAL REQUIREMENTS:-**

1. Performance Requirements

• Response Time: The dashboard shall load real-time weather data within 2 seconds after a user request, assuming a stable internet connection.

• Data Refresh Rate: The system shall update weather data every 5 minutes without interrupting the user experience.

• Scalability: The system shall handle up to 10,000 concurrent users without performance degradation.

• Cross-Platform Compatibility: The system shall perform consistently across major platforms, including desktops, tablets, and smartphones.

• API Call Optimization: The system shall minimize the number of API calls to avoid exceeding provider rate limits while maintaining accurate data updates.

SRS Document

- **NON - FUNCTIONAL REQUIREMENTS:-**

  2. Safety Requirements

  • Weather Alert Reliability: Alerts for severe weather conditions shall be delivered with a high degree of accuracy to avoid misinformation.

  • Data Validation: The system shall validate all incoming data from external APIs to prevent displaying incorrect or corrupted information.

  • Safe Usage Guidelines: If integrated with IoT devices (e.g., weather sensors), the system shall include instructions to ensure the proper and safe use of connected hardware.

SRS Document

- **NON - FUNCTIONAL REQUIREMENTS:-**

3. Security Requirements

• Secure Data Transmission: All communications with external APIs and user interactions shall use HTTPS to ensure encrypted data transfer.

• Authentication: If user accounts are implemented, the system shall use secure authentication methods (e.g., two-factor authentication).

• Data Privacy: The system shall not store sensitive user information, such as precise GPS location, without explicit consent.

• API Key Protection: The system shall securely store API keys used to access weather services to prevent unauthorized access.

• Error Handling: The system shall log errors and provide generic error messages without exposing sensitive details about the backend or APIs.

SRS Document

- **NON - FUNCTIONAL REQUIREMENTS:-**

4. Software Quality Attributes

• Usability:

◦ The system shall have an intuitive and user-friendly interface suitable for users with varying levels of technical expertise.

◦ It shall comply with accessibility standards (e.g., WCAG 2.1) to support users with disabilities.

• Maintainability:

◦ The system's codebase shall follow modular design principles to simplify updates and bug fixes.

◦ Documentation for developers shall include API integrations, system architecture, and maintenance guidelines.

SRS Document

- **NON - FUNCTIONAL REQUIREMENTS:-**

  • Reliability:

  ◦ The system shall provide a 99.9% uptime guarantee, ensuring continuous availability of weather updates.

  ◦ The system shall include fallback mechanisms (e.g., cached data) to handle temporary API outages.

  • Portability:

  ◦ The system shall be compatible with all major web browsers (e.g., Chrome, Firefox, Safari, Edge) and operating systems.

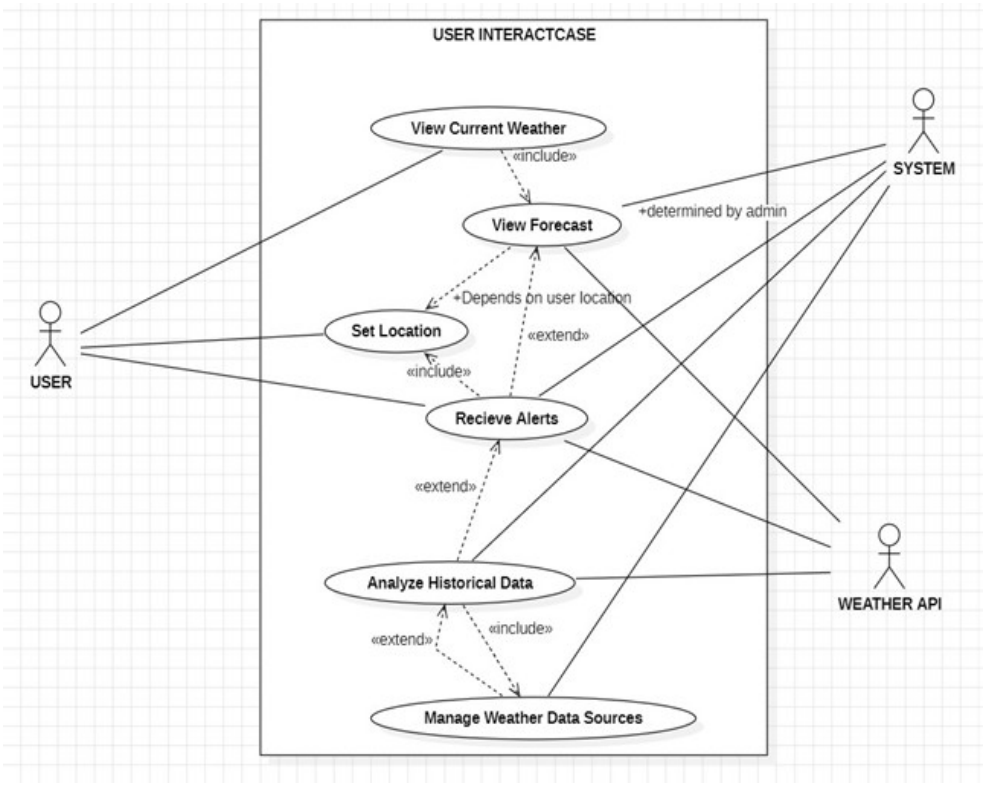
  ◦ Mobile versions shall be optimized for Android and iOS devices.

  • Extensibility:

  ◦ The system shall be designed to accommodate future integrations, such as additional weather data providers or advanced features like pollen forecasts or air quality indices.

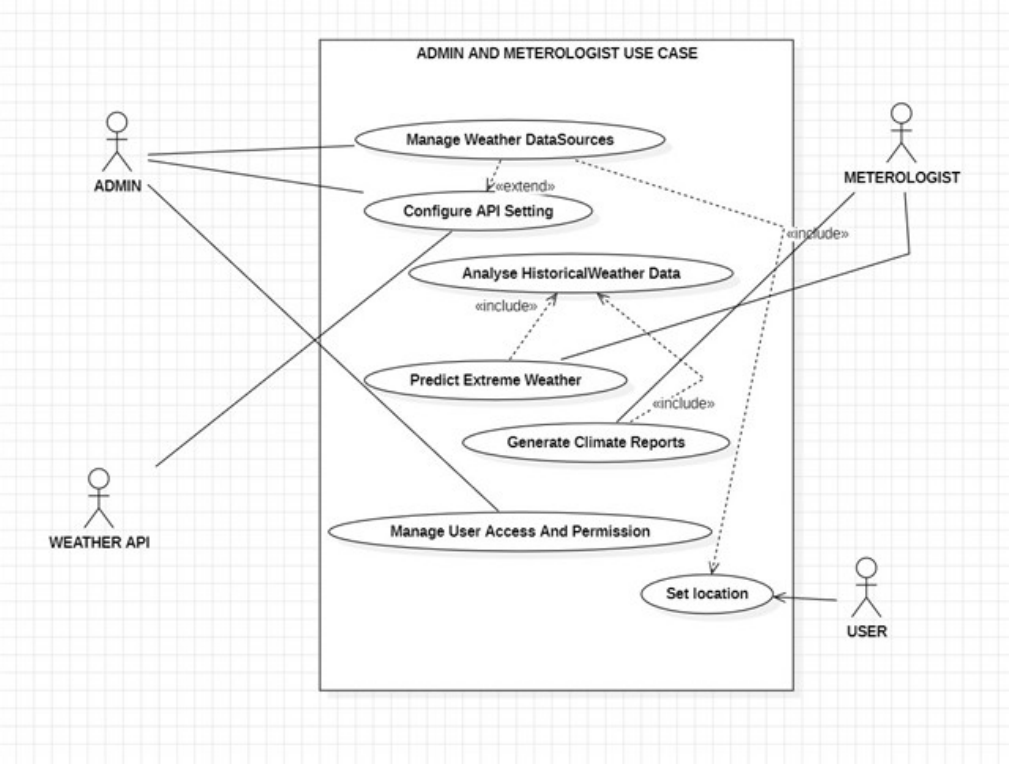
UML - Diagrams

USE-CASE DIAGRAM FOR USER INTERACTION:-



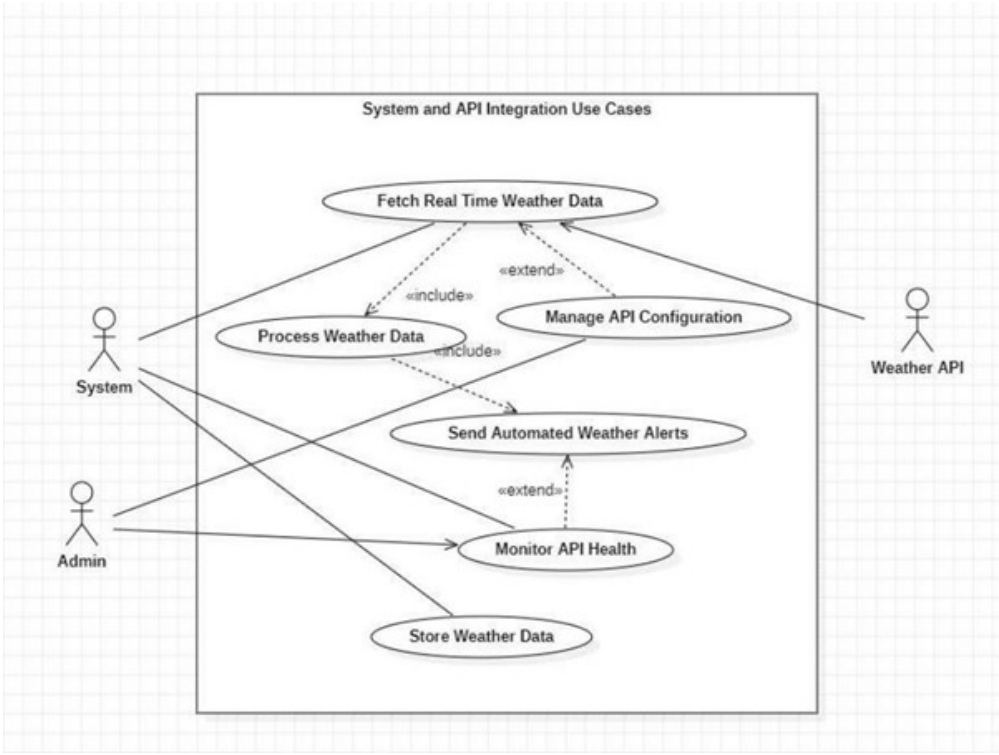
UML - Diagrams

USE-CASE DIAGRAM FOR ADMIN AND METEROLOGIST:-



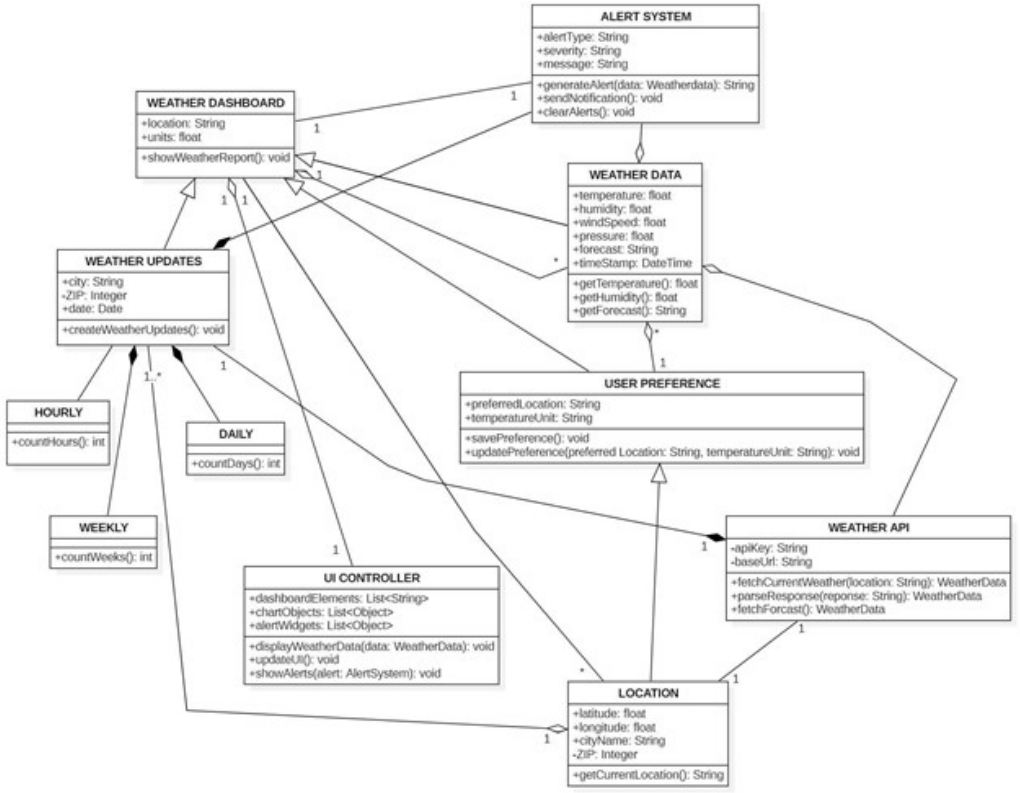
UML - Diagrams

USE-CASE DIAGRAM FOR SYSTEM AND API INTEGRATION:-



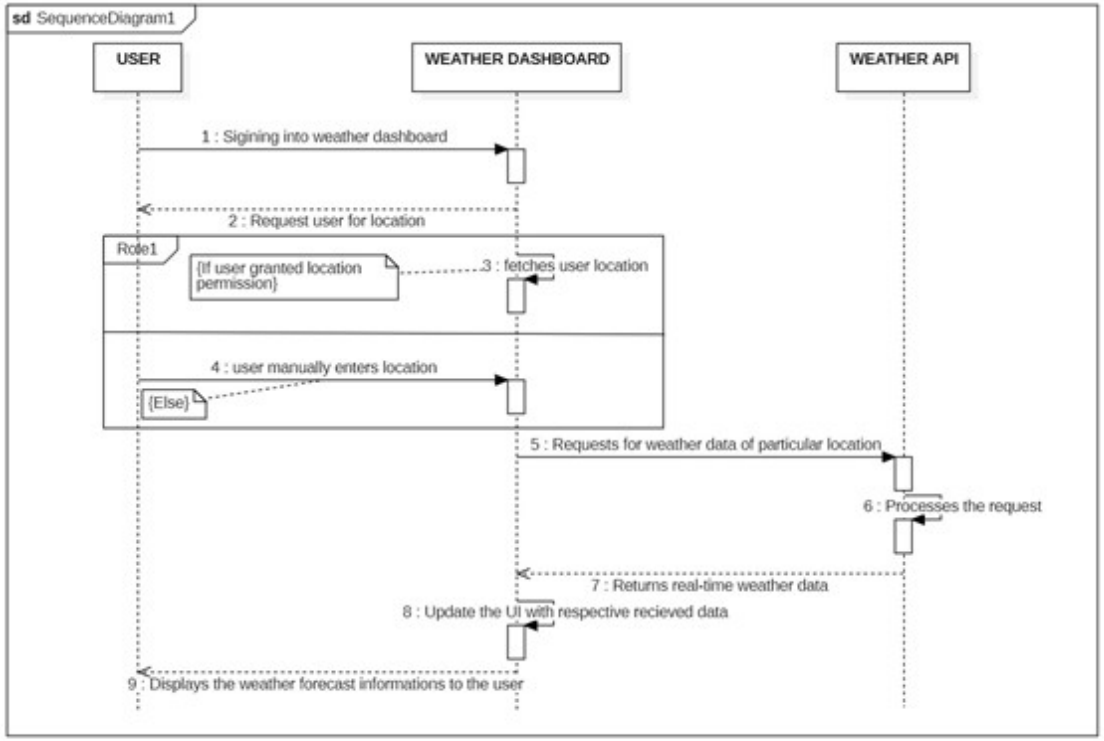
UML - Diagrams

CLASS DIAGRAM:-



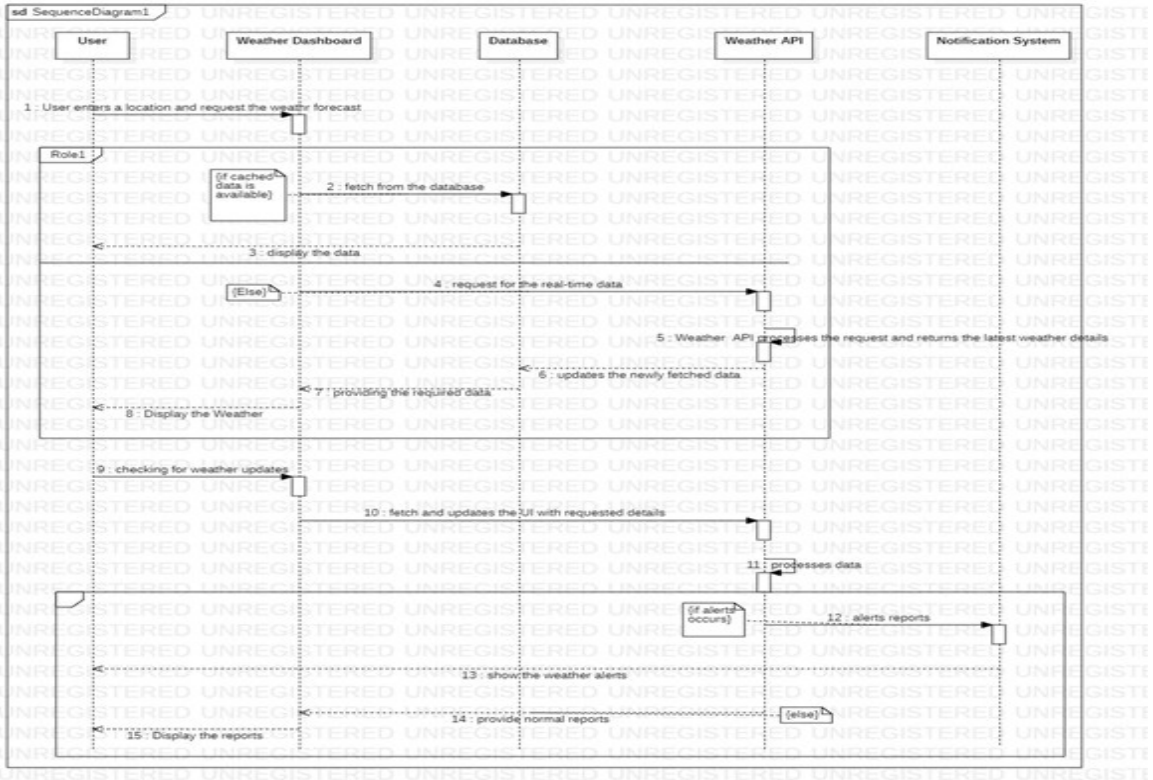
UML - Diagrams

SEQUENCE DIAGRAM FOR USER ACCESSING:-



UML - Diagrams

SEQUENCE DIAGRAM FOR USER RECEIVING WEATHER ALERTS :-



Testing

- **COMPREHENSIVE TESTING**

Scenario 1:

Test Case ID: TC001

Test Scenario: Verifying real-time weather forecast update functionality.

Test Case Description: Ensure that the weather dashboard updates data in real time as per the configured refresh interval.

Test Steps:

1. Open the weather dashboard.
2. Observe the displayed weather data (temperature, humidity, windspeed, etc.).
3. Wait for the predefined refresh interval (e.g., 30 seconds).
4. Verify whether the weather data updates automatically without manual refresh.
5. Compare the new data with previous data to check if there is a valid update.
6. Manually refresh the page and confirm if the latest weather data is displayed.

Testing

- **COMPREHENSIVE TESTING**

Test Data:

Location: London

API Response (before refresh): Temperature: 20°C, Humidity: 50%, Wind Speed:10 km/h

API Response (after refresh): Temperature: 21°C, Humidity: 35%, Wind Speed:12km/h

Test Expected Result:

1.The weather data should update automatically based on the refresh interval.

2.The updated weather values should match the latest API response.

3.A manual refresh should also fetch the latest data.

Actual Result:

1.Weather data updated as expected after the refresh interval.

2.Manual refresh also fetched updated weather data.

Pass/Fail: Pass

Testing

- **COMPREHENSIVE TESTING**

Scenario 2:

Test Case ID: TC002

Test Scenario: Verifying weather alerts and notifications functionality.

Test Case Description: Ensuring that the dashboard correctly displays weather alerts(e.g.,storm warnings, extreme temperatures) based on API responses.

Test Steps:

1.Open the weather dashboard.
2.Verify if any weather alerts are displayed based on the current location.
3.Check if the alert message matches the API response.
4.If an alert is displayed, click on it to view more details.
5.Check if the user receives notifications (if enabled).

Testing

- **COMPREHENSIVE TESTING**

Test Data:

API Response with Alert:

Location: New York

Alert: "Severe Thunderstorm Warning"

API Response without Alert:

Location: India

No alerts expected

Test Expected Result:

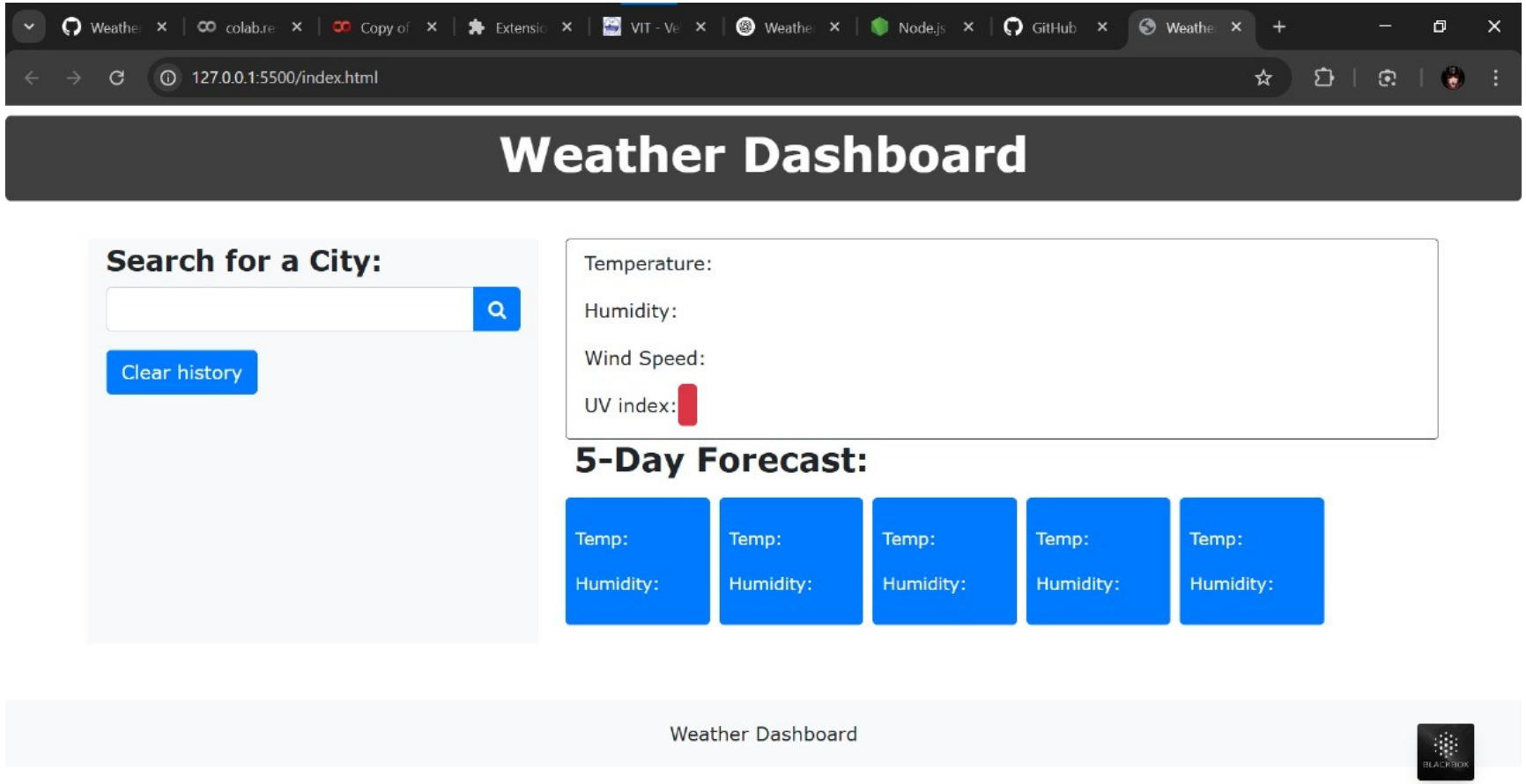
1. If a severe weather alert exists in the API response, it should be displayed on the dashboard.
2. Clicking on the alert should show detailed weather warnings.
3. If notifications are enabled, the user should receive a pop-up or in-app notification.
4. If no alerts exist, the dashboard should not display unnecessary warnings.

Actual Result:

1. Weather alert displayed for New York with correct details.
2. No alert displayed for India as expected.

Pass/Fail: Pass

Project – UI Screenshots



Project – UI Screenshots