Group Project Specification Breakdown

1. General Project Overview

Task: Design and implement a software that retrieves data from two separate services (APIs), combines the data in a meaningful way, and visualizes it. The visualization should be user-adjustable and flexible. TAs must be able to use the application without requiring credentials for third-party services.

Creativity/AI usage: You are required to use an AI tool to generate the idea for the application. If the idea is impractical or overly complex, common sense should be applied to adjust it. Any AI tools used must be documented.

2. Requirements

Practical Requirements:

- Design Patterns: Apply design patterns and principles taught in the course.
- Documentation: Comprehensive documentation is essential.
- Code Readability: AI-generated code must be understandable and explainable.
- Version Control: Use Git for versioning and tag your commits clearly.
- Interface Design: Thoughtful design and implementation of the UI.
- Unit Testing: Implement unit tests for core components in Java.
- AI Integration: Use AI in the design and implementation, and document the usage.

Functional Requirements:

1. Data Retrieval: Retrieve data from two separate APIs.
2. Data Combination: Combine the data meaningfully and display it using advanced visualizations (graphs, maps).
3. Visualizations: Users must be able to adjust parameters (date, time, location) and select which data is shown.
4. Save Preferences: Allow users to save preferences for future visualizations.
5. Scalability: Design should be flexible to allow the addition of more data sources or APIs later.

3. Implementation Details

Language: Core components must be written in Java. You can use other languages for the UI (e.g., JavaScript).
Web apps are allowed if the core functionality is in Java and can be run locally without hosting.

4. Grading Breakdown

Prototype Submission (0–1 points)

Due date: September 29. Submit a prototype that demonstrates your understanding of the requirements. Options for submission:
- A UI sketch (e.g., Figma)
- A rough UI implementation with basic layout and flow

- An intermediate version with basic UI functionality
Submit a design document describing the structure, components, and APIs you plan to use.

Mid-term Submission (0–2 points)

Due date: October 27. Submit partially implemented software and an updated design document, including self-evaluation and proposed changes.

Design Document (0–3 points)

Points are awarded for:
- High-level system description (0–1 points)
- Component descriptions and responsibilities (0–1 points)
- Internal interfaces (0–0.5 points)
- Component structure and functions (0–0.5 points)

Describe the design patterns and architectures used.

Final Submission (0–7 points)

Due date: December 1. Evaluation criteria include:
- Functionality (0–2 points)
- User interface (1 point)
- Use of design patterns (2 points)
- Compilation instructions (0.5 points)
- Documentation and readability (1 point)
- Unit testing for core components (0.5 points)
- Documentation of AI usage (pass/fail)

Peer Review (0–2 points)

Each group will review another group’s project and provide feedback.

5. Guidelines and Restrictions

Make sure to submit all deliverables on time. Late submissions may result in penalties unless there is a legitimate reason. Version control must be used, with all submissions tagged in the Git repository. A clear README file is required with setup instructions.