Smart Journaling Project Managerial Report

GitHub Repositories Link: WIX1002-Occ1-Group6/work_SMJournal

Team Formation and Roles

The **Smart Journaling** project was undertaken by a five-member team. Each member was assigned a specific component of the project to encourage accountability and to parallelise development. According to the project's root README, the roles were distributed as follows:

- **Li Yuchen** – responsible for implementing user login and registration, creating the `User` class and handling data storage.
- **Yang Yuting** – developed the welcome page.
- **Li Zhaoziyu** – implemented the journal page functionality (listing, creating, viewing and editing journal entries).
- **Meng Hanyue** – integrated the weather API, including value extraction from the response.
- **Chen Yuhan** – integrated the sentiment analysis API for mood classification.
- **Farhan Tahzib** – developed the weekly summary page.

This clear division of labour allowed team members to focus on their core tasks while coordinating through regular meetings and version control.

Project Timeline

The following high-level timeline summarises the project's progress:

Week	Activities
Week 1: Requirement Gathering & Design	Discussed assignment requirements, reviewed the SmartJournal specifications, drafted use cases and finalised team roles. Decided on a CLI implementation and identified needed APIs (weather and sentiment).
Week 2: User Management & Data Storage	Li Yuchen designed the `User` class and implemented registration and login functionality. The team decided on using plain text files for data persistence in the first phase.
Week 3: Journal Module & CLI Interface	Li Zhaoziyu developed the journal page to list past dates and allow creation, viewing and editing of entries. The overall CLI menu system was integrated.
Week 4: API Integration	Meng Hanyue integrated the Malaysian weather API (GET) and implemented the value extraction logic. Chen Yuhan integrated the sentiment analysis API (POST) using the DistilBERT SST-2 model. Both modules were tested and linked to the journal module.

Week	Activities
Week 5: Weekly Summary & Finalisation	Farhan Tahzib implemented the weekly summary page, summarising weather and mood over the last seven days. The team performed integration testing, fixed bugs and prepared reports.

Problems Encountered and Solutions

Several challenges arose during the project:

1. **API integration and network issues.** When first integrating the weather API, the team encountered network timeouts and inconsistent responses. The solution was to add error handling and retry logic in the API module and to cache results locally to reduce repeated calls.

2. **Different data formats.** The weather API returned JSON with nested arrays, while the sentiment API returned a list of labelled scores. To handle this complexity, the team wrote dedicated parsing functions that extracted the `summary_forecast` and selected the highest-score label, respectively. Thorough testing ensured correct extraction.

3. **Time-zone differences and greeting logic.** The greeting message depends on the local time (GMT+8). Some members initially had problems during testing due to their own computer time zones. The team resolved this by standardising all time calculations to GMT+8 and explicitly documenting this in the code.

4. **Version control conflicts.** As work progressed concurrently, merge conflicts occasionally occurred in shared files. The team adopted a workflow where each member worked on separate branches and merged their changes via pull requests after peer review, minimising conflicts.

5. **File I/O limitations.** During testing, concurrent reads/writes led to corrupted journal files. The solution was to synchronise access to files and to ensure that file handles were properly closed after each operation.

6. **Balancing workload and timeline.** Because some tasks took longer than expected (e.g., debugging API calls), the team reallocated some responsibilities. Other members temporarily assisted with debugging and documentation to stay on schedule.

Conclusion

The Smart Journaling project benefited from a clear division of responsibilities and consistent communication. By assigning each member ownership over a specific module and setting intermediate milestones, the team delivered a functional application within the deadline. The project not only meets the core requirements—user registration, journaling with weather and mood enrichment, and weekly summaries—but also lays the groundwork for future improvements. The experience underscored the importance of modular design, effective problem-solving strategies, and adaptability in collaborative software development.