Sprint 2 Retrospective

Date: 24/09

Sprint Number: 2

Attendees: Nick, Will, Dave, Michelle, Jason

---

1. Overview of the Project

Our project was initially understood as a timetable management webpage for the Victoria Institute of Technology, accommodating three types of users: students, teachers, and admins. The platform would allow students to register, log in, and view their enrolled course timetables, while admins could create new courses by inputting relevant course and staff information.

As part of our deliverables, we developed a fully functional login, registration, password recovery, and timetable display page. All the pages were interconnected with proper routing, and the backend was supported by a SQL database on Supabase. Middleware for registration, login, and password recovery was successfully implemented and linked to the backend. Midway through development, we were also working on implementing multi-factor authentication (MFA) using Google Authenticator.

However, during our final client meeting, we discovered that we misunderstood the intended scope of the project. The system was intended to be designed solely for admin users, functioning as a timetable generation tool, without student-facing features. This unexpected change required us to quickly pivot and restructure the application for the next sprint, focusing on delivering what the client actually required.

2. What Went Well:

- Adjusted confluence based on tutor feedback: Progress Report 1 Feedback
  - Adjusted Weekly meeting notes to include attendance and tasks assigned Meeting Notes
  - Added Tools used in Confluence page Team & Tools
  - Added Non-functional requirements User Stories
  - Updated motivational model to more accurately represent userbase Motivational Model
- Solid progress in early stages: We successfully developed core features like login, registration, password recovery, timetable display, and admin display with seamless routing between most pages.
- Migrated backend from Django to Supabase efficiently.
  - Backend functionality: The backend, created using Supabase, effectively handled user authentication and data management.
- Smooth team collaboration: Despite the sudden shift in requirements, we were able to work together efficiently to reorient the project without significant delays.
- Adaptability: Our robust application structure allowed for quick adjustments when the scope change was identified, showing flexibility in both technical implementation and teamwork.
- MFA progress: Although still in progress, we began implementing MFA using Google Authenticator, showcasing advanced security practices.
- Testing pipeline: We've updated our test cases in the documentation to reflect the new requirements. Github CI/CD also includes test cases specific to admin page functionalities and login page functionality.

3. Challenges Faced:

- Misunderstanding of requirements: A significant issue arose due to our misunderstanding of the client's actual needs. This led to developing features (like student registration and timetable viewing) that weren't required. In addition, we had to formulate a new solution that pivoted off from our initial structure and plans promptly.

- Time constraints for changes: After the last-minute realisation of the change in scope, we had limited time to refactor our project to meet the admin-only use case, which put pressure on the team. Fortunately, we managed to overhaul our architectural design models and high fidelity prototype accordingly.
- Incomplete MFA implementation: We were midway through developing MFA integration when the requirement change occurred, which interrupted the finalisation of that feature.

4. What Could Be Improved

- More thorough requirement gathering: We should ensure to double and triple-check client requirements during the early stages of the project. This could involve more detailed questioning, better documentation, and regular client check-ins to avoid misunderstandings.
- More agile development: Introducing a more agile process with frequent iterations and client feedback after each sprint could have helped us catch the requirement misunderstanding sooner.
- Prioritizing feature completion: In cases where there are multiple ongoing developments, focusing on completing critical features (like MFA) before starting new ones can reduce risks of incomplete features during scope changes.

5. Lesson Learned

- Importance of clear communication: Miscommunication with the client regarding requirements can lead to wasted time and effort. Moving forward, we will focus on asking clarifying questions and validating our understanding of the project scope more rigorously.
- Flexibility in development: Having a flexible and modular project structure helped us adapt to the last-minute change in requirements, which was a critical factor in keeping the project on track.
- Collaboration and adaptability: The team demonstrated great collaboration in coming up with prompt solutions when faced with unexpected challenges, which is a strength we can build on for future projects.

6. Next Steps

- Complete the implementation of the MFA feature for admins to enhance security.
- Finalise the changes to the platform to cater solely to admin users for timetable generation and management.
- Perform additional testing to ensure that the refactored system meets all of the client’s revised needs.
- Schedule another client meeting to ensure we’re aligned with expectations and gather any final feedback before project delivery.