SCRUM MEETINGS LOG – TAConnect Platform

Project: Smart Office Hours & Scheduling Platform **Tech Stack:** React 19, Django 5, PostgreSQL, Docker

Meeting 1 – Vision & Tech Stack Finalization

Date: 14-10-2025

1. What we finished since last time

- [cite_start]Defined the core problem: Eliminating manual spreadsheets and Zoom link chaos for university TAs.
- [cite_start]Finalized the tech stack: **React 19 (Vite)** for frontend and **Django 5 (REST Framework)** for backend.
- [cite_start]Established the project brand: **TAConnect**.

2. Problems we faced & suggested solutions

- **Problem:** Deciding between a monolithic or decoupled architecture.
- [cite_start]**Solution:** Decoupled React + Django REST API to allow for a more responsive UI and future mobile expansion.
- **Problem:** Identifying how to handle institutional data privacy.
- [cite_start]**Solution:** Committed to a self-hostable Docker approach to keep data on university servers.

3. Next steps

- Initialize GitHub repository with `frontend/` and `backend/` directories.
- Set up basic Docker Compose for local development.
- Draft the initial database schema for Users and Office Hour Slots.

Meeting 2 – Architecture & Database Design

Date: 23-10-2025

1. What we finished since last time

- [cite_start]Initialized **Docker + Docker Compose** setup for PostgreSQL, Backend, and Frontend.

- [cite_start]Drafted the data model: `User`, `InstructorProfile`, `StudentProfile`, and `OfficeHourSlot`.
- [cite_start]Set up the **accounts** app with custom user roles (Instructor vs. Student).

2. Problems we faced & suggested solutions

- **Problem:** Complexity in managing student permissions for specific sections.
- [cite_start]**Solution:** Created the `BookingPolicy` and `AllowedStudents` models to act as a whitelist mechanism.
- **Problem:** Sensitive student IDs in the database.
- [cite_start]**Solution:** Planned for encrypted fields in the `AllowedStudents` table.

3. Next steps

- Implement JWT-based authentication.
- Build the TA dashboard shell in React.
- [cite_start]Create the `OfficeHourSlot` creation logic in the **instructor** app.

Meeting 3 – TA Logic & Core Booking Engine

Date: 02-11-2025

1. What we finished since last time

- Implemented **JWT Authentication** and login/registration flows.
- [cite_start]Completed the TA dashboard view for creating recurring office hour patterns.
- [cite_start]Added CSV upload functionality for the `AllowedStudents` whitelist.

2. Problems we faced & suggested solutions

- **Problem:** Managing recurring datetime logic and buffer times.
- [cite_start]**Solution:** Created a dedicated `utils` app with datetime helpers to handle slot generation.
- **Problem:** UI state management for complex booking forms.
- [cite_start]**Solution:** Standardized on an API client layer in React for consistent JSON communication.

3. Next steps

- Develop the student-facing booking calendar.
- Implement conflict prevention logic to prevent double-booking.

Note: Midterm Break (Exams)Date: 08-11-2025 – 20-11-2025 *No meetings held during this period.*

Meeting 4 – Student Interface & Conflict Prevention

Date: 24-11-2025

1. What we finished since last time

- [cite_start]Launched the **student** app module for browsing and booking slots.
- [cite_start]Integrated **conflict prevention** logic: backend now checks existing bookings and policies before confirming.
- [cite_start]Added booking status tracking (Pending, Confirmed, Cancelled, Completed).

2. Problems we faced & suggested solutions

- **Problem:** Students booking too many slots per week.
- [cite_start]**Solution:** Implemented `set_student_limit` within the `BookingPolicy` model.
- **Problem:** Handling cancellations and cleaning up the schedule.
- [cite_start]**Solution:** Added automated status transitions and boolean flags for cancellation logic.

3. Next steps

- Build the notification subsystem (Email and Web Push).
- Create the analytics dashboard for TAs.

Meeting 5 – Notifications & Analytics

Date: 07-12-2025

1. What we finished since last time

- [cite_start]Developed the **Notification Layer**: Email templates for confirmations and Web Push notifications using VAPID keys.
- [cite_start]Built the **Analytics Dashboard**: TA views for demand patterns and CSV exports of booking data.
- [cite_start]Integrated **Swagger/OpenAPI** documentation for backend endpoints.

2. Problems we faced & suggested solutions

- **Problem:** Service workers for push notifications failing in some browsers.
- [cite_start]**Solution:** Refined the service worker implementation in `public/sw.js` and added environment variable toggles for VAPID.
- **Problem:** Performance lag when loading large analytics datasets.
- [cite_start]**Solution:** Added database indexes to booking and slot queries to maintain speed.

3. Next steps

- Perform full end-to-end workflow testing.
- Finalize Docker documentation and `.env` setup guides.

Meeting 6 – Final Review & Project Delivery

Date: 19-12-2025

1. What we finished since last time

- [cite_start]Conducted final validation of the entire booking lifecycle.
- [cite_start]Polished the UI with **Tailwind CSS**, including theme toggles and responsive layouts.
- [cite_start]Finalized the **System Architecture** and **UML Diagrams** for documentation.

2. Problems we faced & suggested solutions

- **Problem:** Difficulty setting up the environment for new users.
- [cite_start]**Solution:** Wrote comprehensive `SETUP.md` and `ENVIRONMENT_VARIABLES.md` files.
- **Problem:** Minor bugs in the email rendering for mobile devices.
- [cite_start]**Solution:** Refined the HTML templates in `backend/ta_connect/templates`.

3. Next steps

- [cite_start]Upload the final codebase and Docker configurations to the repository.
- Submit the project and final documentation.