

Date: November 5th

Attendees: Emmanuel, Wail, Yesenia

Notes:

- Begin the Implementation Phase!!!!
- Create the GitHub repository for coding work. (Wail will take care of this)
- Assign tasks to each team member.
- Set deadlines and align on the timeline for implementation, code review, and unit testing.

Overall Schedule

- **Two weeks** to complete implementation (Nov 5 → Nov 17).
- **Unit testing:** Thanksgiving week (light work expected).
- **Presentation date:** Likely **Dec 1** (Dec 3 possible depending on class order).
- **Final Exam:** Dec 8.

Submission Deadline

- **All code must be submitted by next Friday** (about 1.5 weeks from meeting date).
- This allows time for:
 - Code review
 - Merging into main branch
 - Integration testing

Project Structure Overview

Wail explained that the implementation falls into **three main categories**:

A. Server-Side

- System Manager
- Data Manager
- Core logic classes
- CSV/Local “database” handling

B. Client-Side

- GUI
- Login creation
- Client-side classes

C. Common / Shared

- Messages

- Enums
- Helper classes
- ClientHandler

Task Assignments

Yesenia

- **System Manager**
- **Data Manager**
- **CSV file handling** (user, student, admin, course, university CSVs)
- **Logging & Reporting**
- Will call Emmanuel's use case classes through System Manager.

Wail

- **Networking classes:**
 - Server
 - Client
 - ClientHandler
 - Message classes
 - Enums
- **User, Student, Admin classes**
- **University, Course, Schedule classes**
- Will consolidate Data Manager & Reporting tickets.

Emmanuel

- **Use Case Method Implementations** (each as its own class):
 - Course Enrollment
 - Drop Course
 - Waitlist
 - Prerequisite Check
 - Access/Manage Student Schedule
 - Manage Holds
 - Enrollment Reporting
 - Update Reporting
 - Edit Course
 - Delete Course
 - (Excluding login)
- Each use case method will be a standalone class invoked by System Manager.

Shichang

- **GUI Implementation**

- Connect GUI to the client-side logic.

Clarifications Made During the Meeting

- **System Manager** acts as a facade, coordinating all features but not implementing each method internally.
- Use case methods will be implemented by Emmanuel and then called inside System Manager.
- Waitlist functionality will likely tie into Course or University data structures, not a separate CSV.
- Logging + Reporting will now be merged into Data Manager responsibilities.
- Everyone must ensure consistent CSV structure across all systems.

GitHub Setup & Workflow

Repository Setup

- Wail will create the **GitHub repository** and send the clone URL.

Workflow

- Each member will:
 1. Clone repo to Eclipse
 2. Create their own branch
 3. Commit + push their work
 4. Open Pull Requests
 5. Team reviews before merging

Important

- Everyone must **test pushing code TODAY** to confirm no configuration issues.
- If someone cannot push or import the repo, the team will resolve it immediately.

Next Steps

Today

- Wail: finalize GitHub repo setup.
- Team: confirm clone/push works in Eclipse.

This Week & Next

- Begin implementation.
- Communicate early if stuck.
- Submit all work by **next Friday**.

Next Meeting: **Next Wednesday** (weekly meeting schedule continues) At 3pm.