Project Specification for Group #175

Team Name: Study Buddy Finder

Domain: Matchmaking and Learning Collaboration

Software Specification:

This program allows users (students) to be matched with and connect with other students for collaborative study sessions based on their availability, courses, and/or program of study.

User Stories:

Team user story (core functionality):

As a student, I want to be matched with a study buddy who is in the same course as me and is available during my study times, so that we can collaborate effectively and keep each other accountable.

Tanaya:

I want to be able to create an account and fill in my availability and personal information (such as name, courses, bio, etc.) and see a list of potential study buddy matches. I also want to be able to see the HomePage when the application is opened.

Cooper:

I want to be effectively matched with other students I can study with based on our common availability and shared courses.

Jinbo:

I want to be able to edit my profile/personal details, including name, availability, bio, etc.

Alex:

I want a user interface which allows me to input the information necessary for account creation.
If nobody with my availability shares my courses, I want to be shown that there are no matches, and to be given the option of expanding my search to find other study buddies in my same program.

Harris:

Once I have made an account, I want to be able to log into my account and see my potential study buddy matches. I also want to be able to log out.

Proposed Entities for the Domain:

1. **User (Student)**
    - Instance variables: username, email, password, name, enrolled courses, program of study, bio, resourceID*, schedulerID*, matches**
    - * These are Slotify-generated UUIDs for the user's Slotify user (*resource*) and availability (*scheduler*) respectively
    - ** Matches are in Map<User, List<Timeslot>> format. This is a map of the Users who are determined to be study buddy matches and a list of Timeslots for which they share availability with the user seeking matches.
2. **Course**
    - Instance variables: course code, course title
    - For example: *CSC207, Software Design*
3. **Timeslot**
    - Used in the functionality of the scheduler/availability.
    - For simplicity due to time constraints, all time slots will be 1 hour long, starting/ending on the hour, and availability can only be declared between 9 AM and 5 PM.
    - Instance variables: day of week (int 1-7), start time (int 9-16)

4. **Availability ***
    - Utility class (no instance variables)
    - getAvailability(String schedulerID) - fetches the specified scheduler's availability from Slotify and convert it into a format better suited for the matching algorithm (Map<Timeslot, Boolean>), e.g. Monday 10:00-11:00, false - indicating the student is unavailable during this time.
5. **Matcher ***
    - Utility class (no instance variables)
    - findMatches(User user) - performs the study buddy matching algorithm to find potential matches according to shared availability and courses*.
        i. * In the case that no matches are found for study buddies taking the same courses, the user will be prompted to expand their search to consider students in the same program rather than the same courses. This method will be overloaded to accommodate for this differing match criteria: findMatches(User user, boolean expand)

* Availability and Matcher entities were ultimately removed towards the end of the project. These utility classes made it more difficult to adhere to Clean Architecture and the SOLID design principles. It was simpler and more efficient to instead move these API-calling methods to our DAO and use dependency injection where needed.

Proposed API for the project:

- **Meeting scheduler/availability API — Slotify**
    - https://slotify.ca
    - Slotify provides the user with options to designate their availability and to schedule meetings/events with other users.
    - We will use this for the functionality of declaring availability and matching students based on their specified available times.
    - More specifically, we use Slotify in creating and deleting *resources* (i.e. users), *schedulers* (the representation of availability), and retrieving and updating users' availability.

Scheduled Meeting Times + Mode of Communication:

Meeting time outside of lab:

- In addition to meeting in the lab each week, we will also meet virtually via Zoom once per week on Thursdays at 6 PM.

Mode of Communication:

- Communication outside of labs will be via Zoom during the time stated above, as well as via Instagram direct messaging throughout the week.