Software Requirements and Design Document

For

Group 15

Version 2.0

Authors:
Asia T
Kayla H
Nathan P

1. Overview (5 points)

Study Circle is a mobile app designed to help students connect with classmates and find study partners or project teammates. We know that approaching people in class to form groups can be awkward or stressful, so this app makes the process easier and more comfortable. With a Tinder-style swiping system, students can quickly find and match with others in the same courses who are also looking for study groups or project teams.

The app provides a simple way for users to create a profile, select their enrolled classes, and start swiping through potential study partners. If two students swipe right on each other, they get matched and can start chatting right away. By making the process of finding study groups more natural and accessible, Study Circle helps students collaborate more effectively and build stronger academic connections.

2. Functional Requirements (10 points)

1. **User Authentication (High Priority):** The system shall allow users to register using their college email and password. The system shall send a verification email before allowing access. The system shall allow users to log in using their registered email and password. The system shall provide a password reset option via email if a user forgets their password.

2. **User Profile Creation and Management (High Priority):** The system shall require users to create a profile upon first login, including their full name, college email (pre-filled and uneditable), graduation year, and a short bio or fun fact. The system shall allow users to update their profile information at any time except for their email. The system shall store and retrieve user profile information from the database.

3. **Class Selection for Matching (High Priority):** The system shall allow users to select their enrolled courses from a predefined list of courses offered at their institution. *The system shall allow users to manually add a class if it is not in the predefined list. The system shall store and retrieve the selected classes for each user in the database.*

4. ***Tinder-Style Matching System* (High Priority):** The system shall display user profiles one at a time, prioritizing those who share at least one selected class with the logged-in user. The system shall allow users to swipe right to connect or left to pass on another user. The system shall match two users when both have swiped right on each other. The system shall prevent users from seeing profiles of users they have already swiped left on.

5. **Real-Time Messaging for Matched Users (High Priority):** The system shall provide a messaging interface for matched users to communicate in real-time. The system shall store messages in the database and allow users to view their message history. The system shall display notifications when a new message is received.

6. **Navigation Between Screens (High Priority):** The system shall allow users to navigate between the home, messages, and profile screens using a bottom navigation bar.

7. **Match Management and User Actions (Medium Priority)***:* The system shall allow users to view their list of matches. The system shall allow users to unmatch a study partner.

8. **Push Notifications** (**Medium Priority):** The system shall send push notifications when a new message is received and a new match is found. The system shall allow users to enable or disable push notifications in settings.

9. **User Safety and Moderation (Medium Priority):** The system shall provide a "Report" feature to allow users to report inappropriate behavior or messages. The system shall provide a "Block" feature to prevent further communication between users. Reported users shall be flagged for administrator review.

10. **Logout and Account Deletion (Medium Priority):** The system shall allow users to log out of their accounts from the profile settings. The system shall allow users to request account deletion, which removes their profile, matches, and messages from the database permanently.

3. Non-functional Requirements (10 points)

1. **Security & Data Protection (High Priority):** The system shall enforce authentication using email/password-based login with encrypted credentials. The system shall use Supabase authentication to handle user authentication securely. The system shall prevent unauthorized access by restricting user data access based on authentication status.

2. **Performance & Scalability (High Priority):** The system shall retrieve user profiles and matches in under 5 seconds for a smooth experience. The system shall be designed to scale up to support thousands of users without performance degradation.

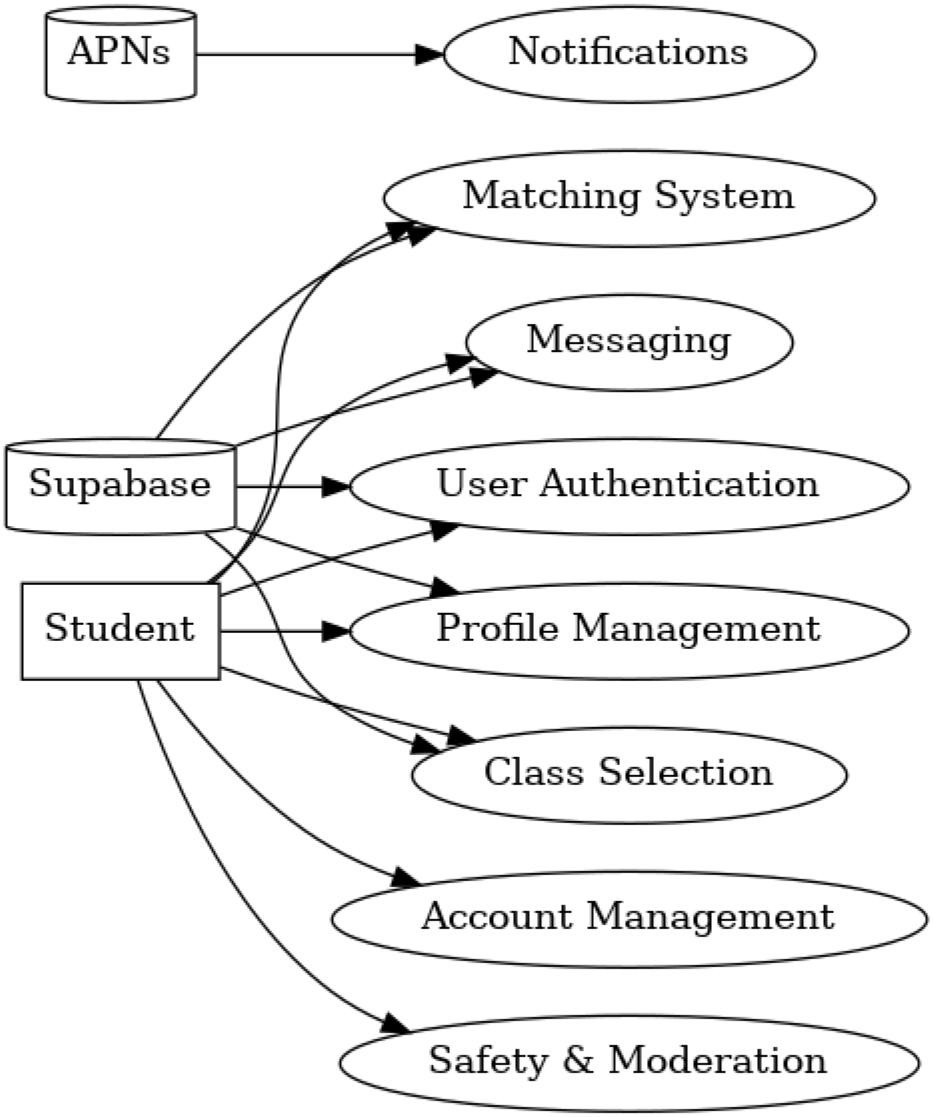
3. **Reliability & Availability (High Priority):** The system shall maintain 99.9% uptime, ensuring users can access the service anytime. The system shall handle errors gracefully with user-friendly error messages instead of app crashes.

4. **Usability & Accessibility (High Priority):** The system shall follow Material Design guidelines for intuitive navigation and UI consistency. The system shall provide clear labels and error messages to guide users through interactions.

5. **Maintainability & Code Quality (Medium Priority):** The system shall maintain clean and well-documented code. The system shall use GitHub issue tracking to document all bugs and feature requests.

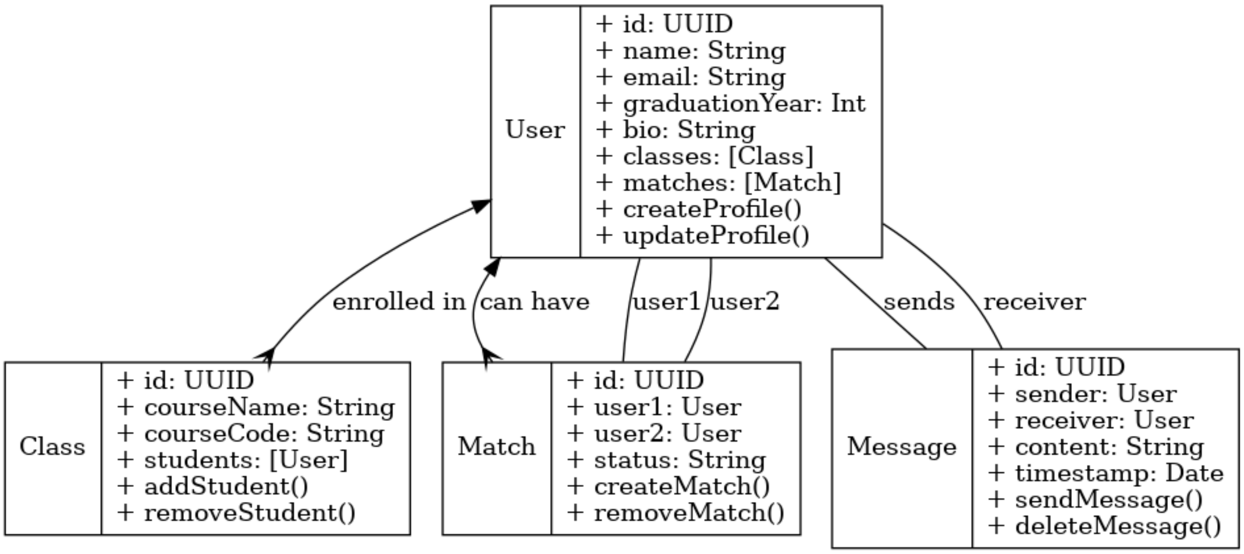
6. **Energy Efficiency & Battery Usage (Low Priority):** The system shall minimize background tasks to reduce battery consumption. The system shall optimize image and data loading to minimize network usage.

4. Use Case Diagram (10 points)

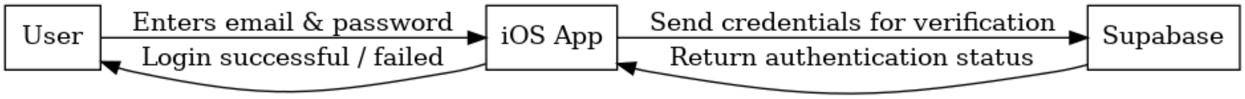


5. Class Diagram and/or Sequence Diagrams (15 points)

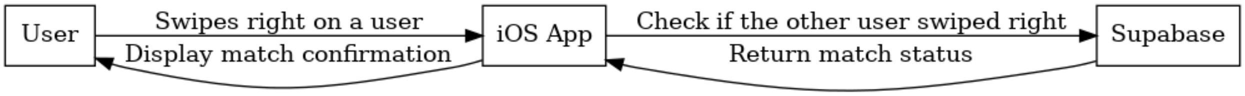
Class Diagram



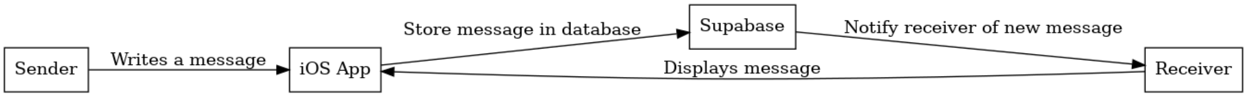
Sequence Diagram for User Authentication


User
Enters email & password
Login successful / failed
iOS App
Send credentials for verification
Return authentication status
Supabase

Sequence Diagram for the Matching System


User
Swipes right on a user
Display match confirmation
iOS App
Check if the other user swiped right
Return match status
Supabase

Sequence Diagram for the Messaging System



6. Operating Environment (5 points)

The software will operate in the iOS ecosystem, targeting iPhones and iPads running iOS 16 and later. It will be developed using Swift to ensure a smooth, native experience while adhering to Apple's Human Interface Guidelines. The app will integrate Supabase as the backend, providing a PostgreSQL database, authentication, real-time updates, and file storage.

7. Assumptions and Dependencies (5 points)

Some assumptions can be…

We assume that Supabase will remain a reliable backend solution for authentication, database management, and real-time data storage. Any disruptions in service could impact user authentication, data retrieval, and messaging functionality.

We assume that course-related data will be accessible in a structured format, either through APIs or manual input, to enable class-based matching. Limited or inconsistent data availability may require alternative matching approaches.

The success of Study Circle depends on students actively using the app for study group formation. We assume that a sufficient number of students will sign up and engage with the platform to ensure effective matching.

Some dependencies can be…

Supabase for Backend Services – Study Circle relies on Supabase for authentication, real-time database operations, and user data storage. Any changes to Supabase's pricing, API structure, or availability may impact development.

Third-Party Libraries and Frameworks – The app depends on external libraries for front-end development, UI components, and backend integration. Any breaking changes, deprecations, or security vulnerabilities in these libraries could affect development timelines.