StudBud: Term Project Requirements and Specification

Authors: Alessandro Guerrero, Sofia Guarisma, Fabio Villatoro, América Marín

Guidelines: (Public) Term Project Specification Guide

Section 1: Introduction

1.1 Project Specific Details

Our term project is called StudBud, a college social media site that lets students connect with other students based on the classes they are registered for.

Our team members are Alessandro Guerrero (aguerr17) , Sofia Guarisma (sguarism), Fabio Villatoro (fvillato), América Marín (amarin11). Each team member will complete approximately one major component for the application:

Alessandro will be responsible for: Frontend development of the home page and user interactions, focusing on displaying courses and peers
Sofia will be responsible for: user registration and profile creation
Fabio will be responsible for: Market research, figma design, frontend design (specifically profiles and messaging related **components**), and figuring out how to get the list of classes for user input.
América will be responsible for: User authentication through Clerk and SSO, as well as our messaging module including messaging interaction and storage

The project is expected to take approximately 4 weeks to complete, with the following weekly milestones: Week 1 for user registration and profile creation, Week 2 for inputting and standardizing class data (with C@B API), Week 3 for developing study group marching and messaging features, and Week 4 for comprehensive testing, debugging, and refining.

1.2 Purpose

There's a growing need for academic collaboration among students who often feel isolated in their courses. One survey study that we reviewed to gather knowledge on the problem stated the aspects of group study in undergraduates across multiple institutions to analyze the benefits. The study indicated that students participating in study groups experienced higher levels of motivation and learning, had more opportunities to use collaborative strategies, had a more enjoyable experience studying, and had more exposure to diverse perspectives. Another study, based on feedback from students particularly in an undergraduate introductory biology course, found that students who did not participate in a study group most commonly cited their reason as logistical difficulties- namely, an inability to find a group.

We further supported this scholarly research with user interviews conducted among peers and within our group members. These interviews confirmed that many students face similar challenges. With over 20 million students in U.S. higher education alone, many lack easy access to peer networks for studying. Through our personal experiences as students, we also find this problem to be common. Study groups would be incredibly helpful for not only achieving academic success but also fully comprehending material and its application. However, many students enter courses with study groups already formed or no intention of forming one, leaving others increasingly isolated. In our discussions, we noted that having access to collaborative study groups would have saved us significant time, energy, and stress.

This problem primarily affects undergraduate students, especially those taking introductory lecture STEM courses where collaboration through study groups is particularly valuable.

StudBud fills this gap by offering a platform for students to find study partners, creating opportunities for improved academic performance and social connection. The booming EdTech market and the increased reliance on digital tools for education present a prime space for launching a solution that enhances both academic outcomes and student well-being. StudBug provides this innovative solution that aligns with modern learning trends.

One alternative solution would be to reach out to a peer in person. Unfortunately, this approach can feel daunting due to the possibility of rejection, lack of time for meaningful interaction, or difficulty finding a reliable partner. Other general study tools, like tutoring applications, also fail to address the class-specific peer collaboration StudBud provides.

Our motivation stems from personal experiences and professional insights. For example, one team member, as a TA, has frequently observed students struggling to access timely support. StudBud addresses this gap by creating a structured way for students to connect with peers for academic help, which eases the burden on teaching staff and enhances the overall learning environment. Most of our team members have had experiences or heard experiences of their peers where finding classes with friends made it a more fulfilling experience.

StudBud specifically addresses the unmet need for student collaboration in academic settings. While other tools focus broadly on academic resources or social networking, StudBud is built around class-specific study connections, making it a unique and target solution. With features like study session organization, peer matching, and a freemium model offering accessibility and incentives, StudBud directly addresses both academic and social needs.

1.3 Intended Audience and Intended Use

StudBud's primary audience comprises Brown University's undergraduate students, particularly those in large lecture-based courses where forming study groups can be challenging. These students often experience difficulty finding study partners due to logistical barriers or social hesitation, leading to feelings of isolation and reduced academic performance. Our target users value convenience, efficiency, and inclusivity in tools that support their academic success.

Some parameters that define the space of potential users, include accessibility, privacy and safety, and scalability. Values like convenience and inclusivity were prioritized based on students indicating they were a more effective way to build study groups.

The app fits into students' lives by offering an easy-to-use platform for personal use. Users can log in at any time to find study partners, schedule group sessions, or access course-specific study materials. We anticipate students will engage with StudBud on a weekly basis- either at the start of a semester to form groups or before major exams or assignments when collaboration is crucial.

While primarily designed for students, StudBud also considers secondary users and stakeholders, two of which come to mind. Faculty, including professors and TAs, might indirectly benefit from the app as students become more engaged and collaborative, requiring less intervention to clarify concepts or facilitate group work. University administrators will also be impact as, by encouraging collaboration, StudBud can contribute to higher student satisfaction and retention rates, aligning with institutional goals. Due to constraints, we will not be overly accounting for these stakeholders- while they will have the same access as students to see who is using the app and participation in study groups, which will provide insights into student collaboration trends to possibly shape future institutional policies or support services, the project will not have specific faculty features until potential future developments.

1.4 Scope and User Stories

For clarity and context of this course we will be dividing our project by user stories based on how we think we will best build up to our final program.

Here we are prioritizing features that directly address student's ability to connect with peers for studying and collaboration, but there are several potential features identifies that fall outside of the current project scope. This includes advanced AI matching, where machine learning or AI algorithms would be able to suggest study partners based on profile information and course enrollment of users. This also includes integration with school systems like Canvas or C@B that would allow students to automatically sign in and link their registered courses without the need to manually input course information. Other possibilities include calendar integrations, which would allow for users to pair up and easily plan study sessions within the same messaging feature, and notifications or reminders about upcoming study sessions. Currently our project scope limits us to users at Brown, but possible future iterations could expand use to other institutions,

User Story 1 (Developer and User Authentication)

As a developer I want to make sure that all users of my program are from a specific University campus. Upon opening my application students should be “greeted” with a StudBud banner and be prompted to Sign Up or Login based on the fact if they have an account or not.

Students can sign up or login through user-authentication provider Clerk with a @brown.edu email address using Chrome or manual sign up/login. As an alternative students can also Login in using their School’s SSO (ex. DUO Push).

User Story 2 (Student and Profile Information)

Upon signing up for this application as a student I want to be able to create a profile with a profile picture, a bio(including: preferred name, concentration, interests, favorite place to study, best spot on campus, “I’m most productive when” …), that I can access on a Profile Tab in the application. A profile will help me best match with other students who have similar course/concentration interests as well as similar study habits and methods.

However things can change in my four years at Brown University. Therefore I would prefer to be able to also be able to edit my account as I wish, including my profile picture and written bio information.

User Story 3 (Developer and Course Listing)

Based on my user demographics, I need to have access to the course available at Brown University that my users are registered too. Therefore, when students sign up for my program they are free to enter 3-5 courses that they are registered to while using the application. Course information is restricted to the department abbreviation and course number.

This information needs to be cached so that throughout a student’s semester, our program remembers this information and it's able to routinely match students based on their course selection.

Upon organizing students based on course selection, the course will be organized alphabetically with student names and profiles listed below.

User Story 4 (Student and interaction with other students)

When I use this application, the courses and fellow peers are listed on the home page of this application that I can easily interact with by scrolling through the list and pressing on other students' profiles. This way I can recognize familiar faces and match studying styles.

In tandem with this I would love to share pictures of myself and my favorite study spots with small written descriptions aside from my profile picture and bio. This way I know that I am interacting with other real students based on their profile threads.

User Story 5 (Student and interaction with other students - messaging)

Once I find a perfect studying match I would love to coordinate a place and time to meet with them. School can be very busy at times so I think it would be best to message peers back and forth so we can come to a consensus of when we are free to meet. However, sometimes it can be hard to start a conversation therefore I would be helpful to have prompted greetings, sentence starters, and phrases. For instance, “Hi my name is ____, nice to meet you!”, “When are you free to study?”, “That sounds great!, “What times work best for you?” “I am a bit busy at that time, what about another time?”, “I am sorry, what if we reschedule? :(”, … etc.

User Story 6 (Developer - messaging)

As a developer I would like for my users to be able to send as many messages as possible for clear and consistent communication. Messages should be unlimited and stored somewhere in memory so that if a user reloads the application or logs in and out of the application They can still view previous messages.

The messaging platform should also be able to connect asynchronously to others so that other computers/phones off/online can send messages that they successfully reach another user. *This means potentially using a third party chatting api to create a consistent flow of messages on/offline?*

User Story 7 (Student – access to chats)

After messaging my studying partners and revisiting StudBud I would like to revisit my individual Direct Messages on a messages tab, so that they are nicely organized based on chronological order with the most recent chats being at the top and oldest at the bottom.

The chats are also scrollable if there are more than fit on my devices screen and my DM Tabs are distinguished by my peers name and profile picture and how long ago they texted me (ex. 2 hours ago, 1 day ago, 1 week ago…).

Section 2: Overall Description

2.1 User Needs

Our team performed a Quantitative and Qualitative investigation of what the user needs are for this product and problem.

Our User Segment

Before addressing specific user needs or pain points, we analyzed a few user segments to determine which one to focus on:

1. High School Students: This is a valid user segment because, like college students, high school students in the U.S. often select their classes and may feel nervous about meeting new people in unfamiliar settings. However, we concluded that we could not add as much value to this group because it is generally more difficult for high school students to make plans and meet after school for studying or collaboration. In contrast, college students can typically meet on campus within minutes, making it easier to foster connections.
2. Online Students: Similarly, this is another valid user segment, as online students often experience challenges related to class selection and feelings of isolation from their peers. However, we determined that we could not provide as much value to this group, as remote students are often spread across the country—or even the globe—making it difficult to facilitate in-person connections. Furthermore, other tools already address the needs of remote learners more effectively.
3. **Our Chosen User Segment**: College Students: We identified college students as the group where we could immediately add the most value. Our app uniquely combines community building with a social experience tailored to university life. College students often have a strong need for connection, and our app helps meet that need by enabling them to connect with peers from their college or classes. By facilitating the discovery of study partners—and potentially new friends—our app directly addresses the challenges faced by this user segment.

College Student Needs (Qualitative)

We conducted around 25 interviews here at Brown and this is what we heard. You can look at our interview sheet here.

- We heard a lot of how students would appreciate a resource to quickly form connections in a new classroom environment.
- The three top needs/things we heard:
  - Connecting with other students based on the classes we are taking. This allows students to create fruitful bonds in classes where students might need it.
  - A solid texting feature would streamline the process of connecting with other students.
  - There should be a way of knowing more about a person I want to connect with. A profile section per user would be a good idea to implement.
- Other things we learned during the interviews
  - For some students these needs are quite important because not having friends in the class pushes them to sn/c or drop the class.
  - Some students recommended special features like adding a matching feature or allowing students to add their past classes.
  - A great deal of responders reported that they did not have anyone to study with the day of the interview.

College Student Needs (Quantitative)
We found three main statistics that show important needs that our app will address.

Statistic 1 - 70% of students who feel socially connected show higher engagement and academic success compared to their disconnected peers. (Bowen, 2021)

- **NEED:** Students need to be socially connected in their class to show engagement in lectures and feel more motivated to learn and work on assignments and study for tests. StudBud aims to tackle this need by providing an engaging social platform for college students based on the classes they are taking.

Statistic 2 - Students with higher levels of engagement and connectedness have a 20% increase in academic achievement. (Dyer, 2015)

- **NEED:** Students need to feel connected to a class to maximize their academic achievement in class. StudBud will allow students to connect with students in the same class to foster involvement with assignments.

Statistic 3 - Research shows that a significant portion of students—up to 40%—report not feeling connected to others in their academic environment. (“School Connectedness Helps Students Thrive | DASH | CDC,” 2023)

- **NEED:** This statistic expands on the concept of connectedness and proves that there is a huge number of students (40%) that need an app like this to form those friendships. Not everyone can be as social and make connections in class. StudBud is an easy way of connecting students with likeminded people in their classes, without the stress and awkwardness of conventional speed friending.

2.2 Assumptions

Technical Assumptions

- Class Input Consistency: We assume that students will enter their clas information in a consistent format or that we will have a way of accounting for this (CSCI 1210 and not cs 121)
    - Basis: Students will probably not feel the format but the algorithm to convert any class input into the right format should be very feasible.
    - Potential Harm: If the assumption is incorrect, users will struggle to connect with classmates, and we would not be able to connect people accurately.
- Existence of a C@B API: A C@B API could streamline the process of getting a list of classes.
    - Basis: It is valid to believe Brown has an API that gives access to this information.
    - Potential Harm: If there is no such feature, we will have a hard time making sure the class input works perfectly.
- User Interface Intuitiveness: We assume that users will intuitively understand how to navigate the app without tutorials:

    - Basis: Most college students are tech-savvy and familiar with similar apps.
    - Potential Harm: Users may feel frustrated using the app.

Non-technical Assumptions

- Desire for Connection: We assume students feel isolated in their classes and are actively looking for tools to help them connect with peers.
    - Basis: Quantitative user research shows some students feel lost in large classes; this does not necessarily apply universally.
    - Potential Harm: Overestimating the need for connection at Brown could lead to wasted resources on features students don't use.
- Ease of adoption: We assume students will be willing to add another app to their academic toolkit without resistance.
    - Basis: Observations of students using multiple apps like Canvas, Google Drive, etc.
    - Potential Harm: If students see the app as redundant, adoption rates may stagnate.
- Classmate Willingness to Collaborate: We assume students will actively engage with others once connections are made through the app.
    - Basis: User feedback suggested students want collaboration tools, but this isn't guaranteed behavior.
    - Potential Harm: If students connect but don't engage, the app may not deliver meaningful value.

2.3 Dependencies

Our project has a lot of crucial technical and non technical dependencies.

Technical Dependencies

- Backend Infrastructure: We will need a cloud service like the ones used in previous projects (Firebase) to store our data, enforce user authentication, and allow for future scalability.
- Frontend Development: We will probably be using the react framework for the website's development. Moreover, we will also depend on specific libraries for user interface components and responsiveness.
- Data Security and Privacy: We might implement encryption protocols to secure user data.
- Third-Party Tools: We will be using Figma for UI/UX design and prototyping.

Non-Technical Dependencies

- User participation: We rely on students to input accurate class information and actively use the platform to connect with classmates.

- Institutional Support: Having help from Brown to get the list of classes or get access to a C@B API would be extremely helpful during development. Moreover, once the website is deployed, Brown could help us promote the project in large lectures!
- Social and Cultural Context: The project assumes and relies on a cultural and social context where students value collaboration and seek peer connections for academic and social support.
- Legality of Data Collection: We depend on user consent to collect and process personal information, such as class schedules and study habits, in compliance with privacy laws.
- Inclusivity: We expect and rely on students, the vast majority of college students having a phone or computer to ensure equal participation and engagement.

Section 3: System Features and Requirements

3.1 Module Design

Figma mobile design (need to work on desktop version):
https://www.figma.com/design/NE2j1jqHTKSuhI35YdT93w/StudBud-Prototype?node-id=0-1&t=bocvtNIvBOHR0OSU-1

Authentication Module

The Authentication Module is responsible for user sign-up, login, and session management. It ensures that only users with valid `@brown.edu` email addresses or those authenticated via university Single Sign-On (SSO) can access the platform. This module integrates with Clerk to handle secure authentication and session tokens. It exposes interfaces for other modules to verify active sessions and restrict access to authenticated users.

Profile Module

The Profile Module allows users to create, view, and edit their profiles. Profiles include personal information such as profile pictures, bios, and preferences. Users can manage their privacy settings to control what information is visible to others. This module interacts with the Matching Module to provide relevant profile data for partner suggestions and the Messaging Module to display user details during conversations.

Course Input and Standardization Module

The Course Input and Standardization Module processes and validates user-provided course data. It ensures consistency in course formatting, converting inputs like `cs 121` into standardized formats such as `CSCI 1210`. This module stores course selections and, if available, integrates with the C@B API to validate course entries. It serves as a critical data source for the Matching Module.

Matching Module

The Matching Module generates study partner recommendations based on shared courses and user preferences. It uses profile data and course enrollments to match users effectively. Match history is stored for future reference. This module interacts with the Profile and Course Input Modules to retrieve relevant data and provides users with recommendations accessible on the homepage.

Messaging Module

The Messaging Module facilitates communication between users. It allows users to send and receive direct messages asynchronously and stores message history for future reference. Features like canned responses and notifications enhance user interaction. The module ensures message data is encrypted during transmission and stored securely, with accessibility through a dedicated Messages tab.

Privacy and Settings Module

The Privacy and Settings Module enables users to manage their visibility and notification preferences. It enforces privacy settings across all modules, ensuring user data is protected. Users can access and modify their settings through a dedicated interface, and the module integrates with the Profile, Matching, and Messaging Modules to ensure consistent application of privacy rules.

Communication with External Software

StudBud interacts with several external systems. It uses Clerk for authentication and session management, Firebase or a similar database service for data storage, and the C@B API (if available) to validate course data. Additionally, a chat API may be used to enable asynchronous messaging functionality.

3.2 Data Requirements

Consider the modules that relate to storing data and computation. By now, you may have an idea of what kinds of data you wish to collect.
[Exercise] Write out each datum you wish to collect. *These questions are guides!*

- What process of consent will you provide your user when collecting this data from them?
- What do you want this data for? If you couldn’t gain access to it, would your software still work?
- Is this data being stored, but not used? If so, why collect it at all?
- Is any of the data publicly available?

The data to be collected includes:

User Personal Information: We will gather users' full names, university email addresses, profile pictures, and bio details such as preferred name, concentration, and interests. This information is essential for creating personalized user profiles that enable students to identify and connect with peers who share similar interests and study habits. Without this data, users cannot effectively personalize their profiles or find suitable study partners, which is crucial for fostering connections.

Course Enrollment Information: Users will provide details of their current courses, specified by department abbreviation and course number, and optionally, past courses. This data is critical for matching users with classmates enrolled in the same courses. Without access to this information, the app cannot fulfill its primary purpose of connecting students based on shared classes.

Authentication Credentials: To ensure secure access to the platform, we will collect login credentials, including passwords, and Single Sign-On (SSO) tokens if users opt to use university sign-on methods. This data is essential for authenticating users securely and verifying that only authorized students can access StudBud.

Messaging Data: We will collect messages sent and received between users. This data is fundamental for enabling communication within the app, allowing users to coordinate study sessions and collaborate effectively.

User Preferences: Information on privacy settings and notification preferences will be collected to respect user choices regarding their personal data and how they receive updates. This data is important for maintaining user trust and ensuring compliance with data protection regulations.

User Activity Data: We will collect interaction history, such as matches made and messages sent, along with usage statistics. This data helps us improve the app through internal analytics and provide better user experiences. While beneficial, this data is not essential; core functionalities remain unaffected if users choose to opt out.

Consent Process: During the registration process, users will be presented with the Terms of Service and Privacy Policy, which outline what data is collected, how it will be used, and how it will be protected. Users must explicitly agree to these terms before creating an account, ensuring informed consent. Additionally, users will have control over what personal information is visible to others through adjustable privacy settings in their profiles, with options to opt-in or opt-out of data sharing for certain features.

Data Usage and Storage: All collected data serves a specific purpose in enhancing user experience or app functionality. We will not collect or store data that is not used, minimizing privacy risks and respecting user trust. None of the user data is publicly available; all personal information is private to the app and is only visible to other authenticated users. Visibility of personal data is controlled by the user's privacy settings, allowing users to manage who can view their information.

[Exercise] Write out each module that handles data. *These questions are guides!*

- Is this module storing data, or processing it?
- How will you ensure that the collecting, storing, and processing of data is secure?

Authentication Module: This module is responsible for storing and processing user credentials during login and registration. It handles both the storage of credentials and the processing of authentication requests. To ensure the security of authentication data, we implement strict access controls to prevent unauthorized access. Regular updates are performed on authentication libraries to patch any security vulnerabilities promptly. Secure communication protocols like SSL/TLS are used to protect data transmission during the authentication process.

Profile Module: The profile module manages user profiles, including personal information and preferences. It involves storing profile data and processing profile updates and retrievals. To minimize risk, we practice data minimization by collecting only the necessary information required for the app's functionality. Input sanitization is applied to all user-provided data to prevent malicious entries. Users have control over their profile visibility through adjustable privacy settings, enhancing trust and compliance with data protection regulations.

Course Input and Standardization Module: This module handles user input of course information and ensures that the data is standardized for consistency. It stores course selections and processes inputs by validating and standardizing them. Security measures include sanitizing all inputs to prevent malicious data entry and ensuring secure storage by protecting course data with appropriate database security measures. This ensures that the matching process is accurate and secure.

Matching Module: The matching module generates recommended study partners based on shared courses and interests. It processes data by running matching algorithms and may store match history to improve user experience. To protect this module, we implement algorithm security measures to prevent tampering or exploitation of the matching logic. Access to matching data is restricted to authorized processes only, ensuring that user data used in matching remains confidential and secure.

Messaging Module: This module enables communication between users through messaging. It involves both storing message content and processing the sending and receiving of messages. Security is maintained by implementing strict access controls so that only intended recipients can access message content. Messages are encrypted during transmission using secure protocols to protect user privacy. Content moderation mechanisms may be in place to detect and handle inappropriate use while respecting user privacy.

Privacy and Settings Module: The privacy and settings module manages user privacy settings and application preferences. It stores user settings and processes the application of these settings throughout the app. To ensure users have control over their data, we provide easy access for users to adjust their settings at any time. Enforcement mechanisms ensure that privacy settings are consistently applied across all modules. Audit trails are maintained to log changes to settings, enhancing transparency and security.

3.3 Risks

Stakeholders	Potential Risks
Users (Brown University Students)	- Users may feel uncomfortable with the applications use of their personal data and how this data is being used - Users may feel uncomfortable releasing personal information, name, identity, location, courses that they are enrolled to because information can be used maliciously by other users - Direct Messages can be inappropriate, and developers can't completely promise a thorough check of proper interactions, due to data privacy and vague user interaction codes that user accept to following upon sign up of the application
The University's Academic Code	- Brown University has a student academic code of integrity that can be broken through student interactions in this application, the university may not want to collaborate with our application(ex. Through School SSO and access to active course list) due to student liability: - For instance, it is up to student discretion as to how they communicate with other students through direct messages, meaning chats can be inappropriate and discriminative against other students, student groups, communities, and identities - Furthermore, how closely can StudBud make sure that students aren't just sharing assignment/exam answers over direct messages. What are the chat restrictions that an application can have while still respecting private conversations?
Professors/instructors/teaching assistants	- *Similar to the risks above.* - Professors have a variety of collaborative policies that can be potentially broken based on a lack of supervision of the substance of direct messages. For instance, Students can be sharing assignment/exam answers and answer keys through these messages. - Professors/instructors/teaching assistants are also individuals that can be inappropriately talked about and discriminated against on social media platforms, what are the policies against these conversations.

In this section, you'll identify the key risks your app creates for users, external stakeholders, and the project itself. If there are risks you deem **necessary** to take, we want you to explain why! We also want you to recognize larger social, economic, and political risks that contextualize the project you're pursuing. **Where does your work fit into this context?**

[Exercise] Write out a list of risks that could come from your project.

- Is there a security risk related to the data you collect and store? What is it caused by? How can you mitigate this risk? (Remember that metadata can also carry risks.)
- Are there accessibility risks involved in the kind of presentation or interaction you will provide? E.g., are you concerned about repetitive stress caused by your UI design, or about the way your page works making it difficult for users to employ screen readers or magnifiers?How can you adjust the design of your View to account for this accessibility need and others?

[Exercise] Write out a list of risks that could affect your project.

- Is there a larger social, economic, political, moral, philosophical and or academic context around your project that poses a risk to your users, or the project itself? How will you mitigate this risk? Can you avoid or curb it at all?
- Are there people who would oppose your project? (E.g., competitors, people your project puts at risk, etc.)

3.4 Testing Plan

In this section, you'll outline how you plan to test each module of your project! You **don't** need to write out each test case at this stage, but you should have a general idea of what edge and generic cases need to be covered by your test suite. (Consider unit testing, integration testing, mocks where appropriate, etc.)

[Exercise] For each module you listed earlier, describe how you would test its functionality and accessibility. Link each set of tests to a user story if possible.

Module 1: Authentication Module

User Story 1: As a developer, I want to ensure that all users are from a specific university campus.

Testing Strategies:

1. **Functionality Tests:**
   - **Unit Tests:**
     - Test sign-up with valid `@brown.edu` emails.
     - Test login with correct and incorrect passwords.
     - Verify SSO login via university authentication.

   - **Edge Cases:**
     - Attempt sign-up with non-university emails.
     - Test login with invalid email formats.
     - Simulate failed authentication attempts.
2. **Security Tests:**
   - **Access Control:**
     - Ensure unauthorized users cannot access protected resources.
3. **Integration Tests:**
   - **Database Interaction:**
     - Confirm that user credentials are correctly stored and retrieved from the database.
   - **SSO Integration:**
     - Test the interaction between the app and the university's SSO system.
4. **Accessibility Tests:**
   - Ensure that the sign-up and login forms are accessible via keyboard navigation.
   - Verify that screen readers correctly announce form fields and error messages.

Module 2: Profile Module

Related User Story:

- **User Story 2:** As a student, I want to create and edit my profile with personal information.

Testing Strategies:

1. **Functionality Tests:**
   - **Unit Tests:**
     - Test profile creation with all fields completed.
     - Verify editing capabilities for profile picture and bio.
   - **Edge Cases:**
     - Input maximum characters in text fields.
     - Upload invalid or large profile pictures.
2. **Security Tests:**
   - **Access Control:**
     - Ensure that only the profile owner can edit their profile.
3. **Integration Tests:**
   - **Profile Display:**
     - Confirm that profile updates are reflected in other modules (e.g., Matching Module).
   - **Database Interaction:**
     - Verify that profile data is correctly saved and retrieved.
4. **Accessibility Tests:**

    - Ensure profile editing interfaces are accessible via keyboard and screen readers.
    - Verify that images have appropriate alt text.

Module 3: Course Input and Standardization Module

Related User Story:

- **User Story 3:** As a developer, I need to handle course listings for students.

Testing Strategies:

1. **Functionality Tests:**
    - **Unit Tests:**
        - Test adding and removing courses.
        - Verify that courses are stored and retrieved accurately.
    - **Edge Cases:**
        - Input invalid course codes.
        - Handle duplicate course entries.
2. **Security Tests:**
    - **Input Validation:**
        - Ensure that only valid course formats are accepted.
    - **Sanitization:**
        - Prevent malicious inputs.
3. **Integration Tests:**
    - **Matching Module Interaction:**
        - Confirm that course data is correctly used for matching.
    - **Database Interaction:**
        - Verify that course selections are properly saved.
4. **Accessibility Tests:**
    - Ensure that course input forms are accessible.
    - Verify that auto-complete features work with assistive technologies.

Module 4: Matching Module

Related User Stories:

- **User Story 4:** As a student, I want to interact with peers based on courses.
- **User Story 5:** As a student, I want to coordinate with study partners.

Testing Strategies:

1. **Functionality Tests:**

    - **Unit Tests:**
        - Test the matching algorithm with various data inputs.
        - Verify that matches are accurate and relevant.
    - **Edge Cases:**
        - Handle cases with no available matches.
        - Test with large datasets.
2. **Security Tests:**
    - **Access Restrictions:**
        - Ensure that matching data is only accessible to authorized processes.
    - **Data Protection:**
        - Use anonymized data where appropriate.
3. **Integration Tests:**
    - **Profile and Course Data Interaction:**
        - Verify that the module uses up-to-date user information.
    - **Messaging Module Interaction:**
        - Ensure that users can initiate communication after matching.
4. **Accessibility Tests:**
    - Confirm that match results are presented in an accessible manner.
    - Ensure that users can navigate and select matches using assistive technologies.

Module 5: Messaging Module

Related User Stories:

- **User Story 5:** As a student, I want to message peers.
- **User Story 6:** As a developer, I want messages to be persistent and reliable.
- **User Story 7:** As a student, I want to access my chats easily.

Testing Strategies:

1. **Functionality Tests:**
    - **Unit Tests:**
        - Test sending and receiving messages.
        - Verify that message history is stored and retrievable.
    - **Edge Cases:**
        - Handle offline users.
        - Test message delivery under poor network conditions.
2. **Security Tests:**
    - **Access Control:**
        - Ensure messages are only accessible to intended recipients.
    - **Encryption:**
        - Verify that messages are encrypted in transit.
    - **Content Moderation:**

        - Implement and test mechanisms for reporting abuse.

3. **Integration Tests:**
    - **Real-Time Communication:**
        - Test message synchronization across devices.
    - **Database Interaction:**
        - Verify that messages are stored securely and efficiently.
4. **Accessibility Tests:**
    - Ensure that the messaging interface is accessible via keyboard.
    - Verify that notifications and messages are announced by screen readers.

Module 6: Privacy and Settings Module

Related User Stories:

- **User Story 2:** As a student, I want control over my profile information.
- **User Story 4:** As a student, I want to manage my interactions with others.

Testing Strategies:

1. **Functionality Tests:**
    - **Unit Tests:**
        - Test updating privacy settings.
        - Verify that changes take effect immediately.
    - **Edge Cases:**
        - Handle conflicting settings.
        - Test default settings for new users.
2. **Security Tests:**
    - **Enforcement:**
        - Ensure that privacy settings cannot be bypassed.
    - **Audit Trails:**
        - Verify that changes to settings are logged.
3. **Integration Tests:**
    - **Module Interaction:**
        - Confirm that privacy settings are respected across all modules (e.g., Messaging, Matching).
    - **User Interface:**
        - Ensure that settings are accessible and understandable.
4. **Accessibility Tests:**
    - Ensure settings interfaces are accessible via assistive technologies.
    - Verify that instructions and labels are clear and descriptive.