

## Cover Page

Study Partner Matching App  
System Specification  
Minnie Cao  
CSC 3150  
Andy Cameron  
06/03/2024



## Table of Contents

## Table of Contents

Cover Page.....	1
Table of Contents.....	2
1. Executive Summary .....	3
2. Introduction .....	4
2.1. Problem Statement / Project Vision .....	4
2.3. Non-functional Requirements and Design Constraints .....	4
2.4. System Evolution .....	5
2.5. Document Outline .....	6
3. Structural Model.....	8
3.1. Model Introduction .....	8
3.2. Class Diagrams.....	9
3.3. Metadata .....	10
Match Class Diagram.....	10
Preferences Class Diagram .....	11
StudyPartnerMa tching Class Diagram .....	12
User Class Diagram.....	13
4. Architecture Design.....	14
4.1. Architecture Overview.....	14
4.2. Infrastructure Model .....	14
4.2.1. Deployment Diagram 1 – Architecture Overview.....	14
4.2.2. Deployment Diagram 2 – Nodes and Artifacts.....	14
4.3. Hardware and Software Requirements.....	15
4.3.1. Hardware Components .....	15
4.3.2. Required Software Components .....	15
4.4. Security Plan.....	15
4.4.1. Security Overview.....	15
4.4.2. Security Plan .....	17
5. User-Interface .....	18
5.1. User-Interface Requirements and Constraints .....	18
5.2. Window/Screen Navigation Diagram .....	18
5.3. UI Wireframes.....	19
6. Appendices.....	20
6.1. Glossary.....	20
6.2. References / Bibliography .....	20
6.3. Supporting documentation .....	20

## 1. Executive Summary

### **Overview:**

Study Partner Matching is an innovative mobile application designed to help university and college students find compatible study partners. The app utilizes a swiping interface for browsing potential matches and suggests partners based on provided information such as school, major, study habits, location preferences, and more. It aims to streamline browsing and finding compatible study partners, creating a pleasant and exciting user experience.

### **Contents:**

This system specification document gives an in-depth look at how Study Partner Matching will be developed. This document summarizes the system's information covered in the System Proposal document. Otherwise, this document focuses on:

**Structural Model:** Detailed UML Class Diagrams and Metadata descriptions to give developers a clear understanding of the system's structure. The section organizes the information by first introducing the model, then presenting the Class Diagrams, and finally providing detailed Metadata.

**Architecture Design:** Detailed discussion of the app's architecture, supported by UML Deployment Diagrams. This section outlines the systems and services used, such as AWS for backend and database services.

**User-interface:** Detailed UI/UX design plans, including UML Window Navigation Diagrams and Wireframes for each page. These designs aim to guide developers in creating an intuitive and user-friendly interface.

### **Next Steps:**

The next step of developing Study Partner Matching is starting the coding process. For the success of this process, the key steps include setting up Cloud system, coding the core features of Study Partner Matching, and implementing its user interface.

### **Final Points:**

We have planned Study Partner Matching's launch date for Spring 2025. The app's first version will be only for iOS operating system, with the plan of expanding to support both iOS and Android mobile devices in future versions.

## **2. Introduction**

Study Partner Matching is an innovative mobile application designed to help university and college students find their compatible study partners. The app allows users to create and manage their accounts, along with adjustable personal and browsing preferences.

The main feature of Study Partner Matching is a fun and interactive platform that motivates collaborative learning. The app provides a safe environment for students to build connections and enhance their learning experience. Users can chat with their matched study partners or send chat requests to suggested profiles.

For more details on the requirements of Study Partner Matching, please refer to Section 4 of the System Proposal.

### **2.1. Problem Statement / Project Vision**

In today's fast-paced academic environment, finding a suitable study partner can be challenging for many students, hindering their learning and overall academic performance. Despite the growth of the internet, there are not many platforms catering to students' needs for academic collaboration, making finding compatible study partners a lack-luster, time-consuming process. Recognizing these challenges, Study Partner Matching aims to connect students with like-minded peers, improving their academic experience and performance.

The project vision is to make finding ideal study buddies effortless and exciting. Study Partner Matching will serve as a centralized platform, connecting students based on their profiles and study preferences. The app will feature a seamless user experience with swipe-based matching, real-time messaging, and shared study calendars. The scope includes designing an intuitive interface and implementing a robust matching algorithm, adhering to privacy and security guidelines to protect user data. The app will be a standalone mobile application, initially available on the iOS operating system.

The stakeholders of Study Partner Matching include students, educators, and parents, since they all are interested in the students' academic growth and performance improvement. More specifically, students, who are the targeted end-users of Study Partner Matching, can use the app to seek compatible study partners. Moreover, the project team is also a stakeholder of Study Partner Matching, since they are responsible for building the app. Thus, all stakeholders benefit from the app's success.

### **2.2. System Capabilities**

Following are the functional requirements of Study Partner Matching, including Use Case names and ID numbers. For more detailed information, please refer to Section 5 of the System Proposal:

- Create Profile (UC – 001): Users register and create a new profile, including personal information, academic interests, study preferences, and availability.
- Adjust Profile (UC – 002): Users update their profile information, including changes to personal details and study preferences.
- Match User (UC – 003): The system matches users using compatibility factors such as study preferences, academic interests, and availability.
- Swipe to Browse (UC – 004): Users swipe left or right to browse through potential study buddies. Swiping right indicates interest, while swiping left dismisses the match.
- Chat (UC – 005): Users communicate with their suggested or matched study partners through text messages.
- Report (UC – 006): Users report inappropriate behavior or issues with other users, which are then reviewed by the app's support team.

### **2.3. Non-functional Requirements and Design Constraints**

Study Partner Matching must meet several key non-functional requirements and design constraints to ensure its success. Firstly, strict security standards are essential to protect user data, ensuring that it is only used for matching purposes. Additionally, the app must support prompt chatting and quickly display matching results to provide a seamless user experience.

The feasibility analysis reveals that the project is technically feasible despite some challenges. While the development team consists of students with limited experience in developing complete

mobile applications, they have relevant coursework and personal project experience. The familiarity of college and university students with mobile apps and swipe-based matching applications also contributes to the app's technical feasibility. However, the project must be completed within one year, requiring careful planning and regular meetings to ensure timely progress.

Resource feasibility is another significant consideration. The project demands substantial funding, human resources, and time. Regular meetings and guidance from experienced advisors are necessary to keep the development on track. Financial resources are also needed for development, marketing, and maintenance. The schedule feasibility is achievable with careful planning and project management to prevent delays.

Organizationally, the project aligns well with the goals and interests of stakeholders, including students, parents, educators, and the development team, making it beneficial for all involved. Legally, there are no significant barriers to developing the app, but compliance with relevant privacy laws and regulations is crucial to protect user data. Contractually, there are no significant obstacles, but clear agreements are essential to ensure smooth development and implementation.

Several constraints must be considered during development. Limited technology and human resources could impact the app's development, implementation, and maintenance. Security concerns require a design that ensures the strict protection of user data. User adoption could be a challenge, with potential resistance or reluctance from students. Promoting the app through social media and collaborating with academic institutions can help mitigate this. Lastly, budgetary constraints may affect development, implementation, and maintenance, necessitating careful financial planning and resource management.

Overall, these non-functional requirements and constraints provide a framework for developing Study Partner Matching, ensuring it meets the needs of its users while addressing potential challenges effectively. For more details, refer to Sections 1 and 4 of the System Proposal.

## **2.4. System Evolution**

Study Partner Matching will become better through regular updates and enhancements based on user feedback and emerging technologies. In the initial version (MVP), we will deliver the core functionality, including user account management, preference customization, profile browsing, and real-time chat.

For future versions, we recommend several enhancements to continually improve the user experience and meet changing student needs. In Version 2, we plan to extend availability to Android mobile devices and incorporate advanced matching algorithms for more accurate and efficient pairings. In version 3 and beyond, features like scheduling study sessions and virtual study rooms will be added to facilitate better collaboration and engagement among users. These updates will ensure that the app remains relevant and valuable to students over time.

Future changes may include availability on Android mobile devices, advanced matching algorithms, and features like scheduling study sessions and virtual study rooms continually improve the user experience and meet changing student needs.

### **2.4.1. Version 2 Changes**

In Version 2 of Study Partner Matching, we will introduce several substantial changes and additional use cases to enhance the user experience and functionality:

- **Android Compatibility:** Extend the app's compatibility to Android devices, allowing a broader range of students to use the app across different mobile platforms.
  - Users can install the app from the Google Play Store.
  - Users can access all existing features on an Android device.
- **Advanced Matching:** Implement a more sophisticated matching algorithm to pair users based on deeper insights into their study habits, academic goals, and preferences.
  - System provides users with more accurate and compatible study partner suggestions.

#### 2.4.2. Version 3 and beyond Changes

In Version 3 and beyond of Study Partner Matching, we will introduce several substantial changes and additional use cases to enhance the user experience and functionality:

- **Scheduling Study Sessions:** Allow users to schedule study sessions directly within the app, including setting reminders and notifications.
  - Users can choose a date, time, and duration for the study session with their selected study partner(s).
  - Participants will get notifications and reminders of the session.
  - Participants can provide feedback to the system by rating the session.
- **Virtual Study Room:** Introduce virtual study rooms where users can join video or audio sessions to collaborate in real-time.
  - Users can send requests to join an existing virtual study room.
  - Users can create a virtual study room.
  - Users can invite their matched study partners to join the room.
  - Users participate in the virtual study room via video or audio.

#### 2.5. Document Outline

This document is organized into six main sections, each covering different aspects of Study Partner Matching. Below is a brief description of the content of each section:

The Executive Summary section provides a high-level overview of Study Partner Matching, its purpose, key features, and the benefits it offers to users and stakeholders.

The Introduction section is divided into several subsections. The Problem Statement / Project Vision summarizes the challenges students face in finding compatible study partners and the app's vision to address these issues. System Capabilities lists and describes the functional requirements and use cases for the app. Non-functional Requirements and Design Constraints outlines the significant non-functional requirements, constraints, and feasibility considerations for the app. System Evolution describes the future enhancements planned for the app, including additional features and functionalities. Finally, the Document Outline provides an overview of the structure and content of the document.

The Structural Model section includes a Model Introduction, which briefly describes the content of the structural model section, including class diagrams and metadata. The Class Diagrams subsection presents the UML Class Model, detailing the classes, attributes, methods, and relationships within the system. The Metadata subsection provides descriptions of the classes, attributes, and methods, including high-level pseudocode to facilitate coding.

The Architecture Design section starts with an Architecture Overview that offers a high-level view of the system's architecture. The Infrastructure Model subsection includes deployment diagrams that illustrate the system's architecture and the nodes and artifacts involved. The Hardware and Software Requirements subsection lists the required hardware and software components for the system. Finally, the Security Plan describes the security measures and plans to ensure user data protection and system integrity.

The User-Interface section outlines the User-Interface Requirements and Constraints, providing the requirements and constraints for the user interface design. The Window/Screen Navigation Diagram subsection provides a visual representation of the navigation flow between different screens in the app. The UI Wireframes subsection presents wireframes of the user interface, showcasing the design and layout of the app's screens. Lastly, the Reports: "Formal Output" Design subsection describes the design and format of any formal reports generated by the app.

The Appendices section includes a Glossary that defines key terms and acronyms used in the document. The References / Bibliography subsection lists the sources and references cited in the document. The Supporting Documentation subsection includes any additional documents or materials that support the content of the main document.

Each section is designed to provide comprehensive information on the corresponding aspect of Study Partner Matching, ensuring that developers and stakeholders have a clear understanding of the system's design, requirements, and future development plans.

### **3. Structural Model**

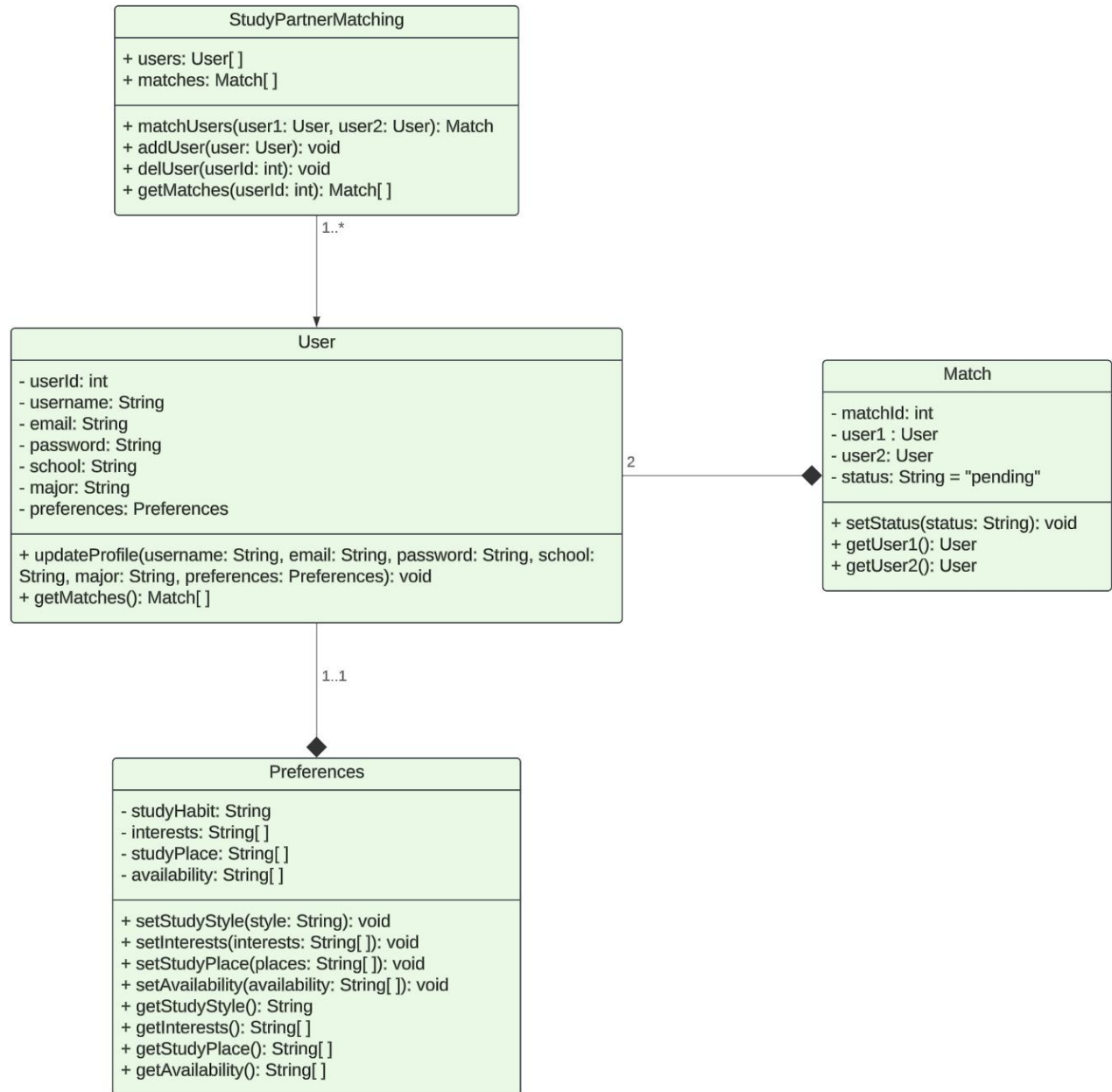
#### **3.1. Model Introduction**

This section contains the structural model of Study Partner Matching, focusing on the class diagrams and their associated metadata. The Class Diagrams subsection presents UML Class Models that detail the system's structure, including class names, attributes, methods, and relationships. Each diagram provides a comprehensive view of the system's components and their interactions.

Following the class diagrams, the Metadata subsection provides detailed descriptions of each class, including attributes and methods. It also includes high-level pseudocode to facilitate understanding and implementation. This metadata ensures that developers have all the necessary information to code the system effectively. Together, these subsections provide a complete and organized representation of the system's structure, aiding in its development and maintenance.



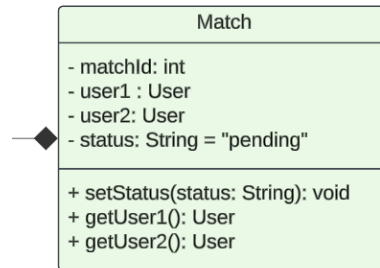
### 3.2. Class Diagrams



[https://lucid.app/lucidchart/55679d9b-b7ca-4df5-876c-551fbacc1efe/edit?viewport\\_loc=-1227%2C-489%2C4028%2C2011%2CHWEp-vi-RSFO&invitationId=inv\\_bee94f8f-11d2-4f41-a270-c2d4b7e4b2c0](https://lucid.app/lucidchart/55679d9b-b7ca-4df5-876c-551fbacc1efe/edit?viewport_loc=-1227%2C-489%2C4028%2C2011%2CHWEp-vi-RSFO&invitationId=inv_bee94f8f-11d2-4f41-a270-c2d4b7e4b2c0)

### 3.3. Metadata

#### Match Class Diagram



**Description:** The Match class represents a match between two users, identified by matchId, and includes references to the two users involved and the status of the match.

**Visibility:** Public

**Abstract:** No

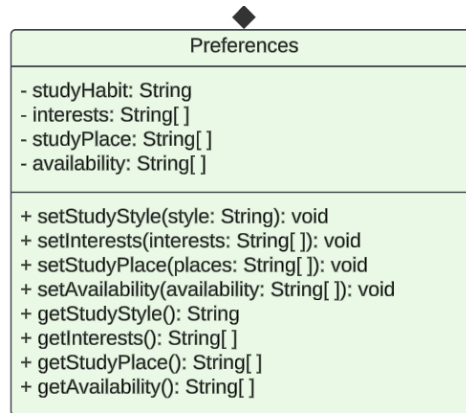
**Attributes:**

Name	Description	Read Only?	Multiplicity
matchId	The unique identifier for the match.	yes	1
user1	The first user in the match.	no	1
user2	The second user in the match.	no	1
status	The status of the match (e.g., "pending", "accepted", "rejected").	no	1

**Operations:**

Name	Description	Is Query?	Is Polymorphic?
setStatus	Sets the status of the match (e.g., "pending", "accepted", "rejected").	no	no
getUser1	Returns the first user in the match.	yes	no
getUser2	Returns the second user in the match.	yes	no

## Preferences Class Diagram



**Description:** The Preferences class represents the study preferences of a user, including study style, availability, and interests.

**Visibility:** Public

**Abstract:** No

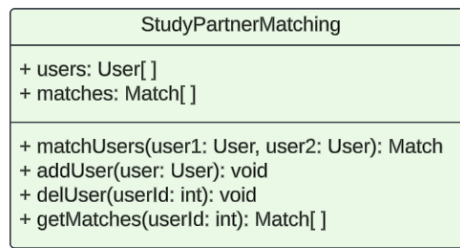
**Attributes:**

Name	Description	Read Only?	Multiplicity
studyHabit	The user's preferred study habit.	no	0..1
studyPlace	The user's preferred study location.	no	0..*
availability	The user's availability for study sessions.	no	0..*
interests	The user's academic interests.	no	0..*

**Operations:**

Name	Description	Is Query?	Is Polymorphic?
setStudyHabit	Sets the user's preferred study habit.	no	no
setStudyPlace	Sets the user's preferred study locations.	no	no
setAvailability	Sets the user's availability for study sessions.	no	no
setInterests	Sets the user's academic interests.	no	no
getStudyHabit	Gets the user's preferred study habit.	yes	no
getStudyPlace	Gets the user's preferred study locations.	yes	no
getAvailability	Gets the user's availability for study sessions.	yes	no
getInterests	Gets the user's academic interests.	yes	no

## StudyPartnerMatching Class Diagram



| 1..\*

**Description:** The StudyPartnerMatching class represents the main application class with attributes for storing users and matches.

**Visibility:** Public

**Abstract:** No

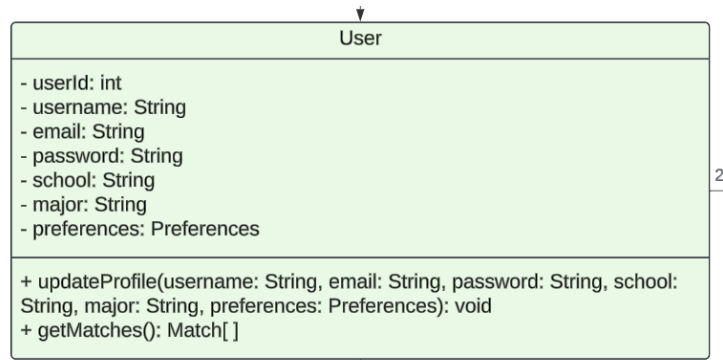
**Attributes:**

Name	Description	Read Only?	Multiplicity
users	An array of all users in the app.	no	0..*
matches	An array of all matches in the app.	no	0..*

**Operations:**

Name	Description	Is Query?	Is Polymorphic?
addUser	Adds a new user to the app.	no	no
removeUser	Removes a user from the app.	no	no
matchUsers	Creates a new match between two users.	no	no
getMatches	Returns a list of matches for a given user.	yes	no

## User Class Diagram



**Description:** The User class represents a user of the Study Partner Matching app. Each user has a unique identifier, a username, an email address, a password, and preferences for study styles, availability, and interests.

**Visibility:** Public

**Abstract:** No

**Attributes:**

Name	Description	Read Only?	Multiplicity
userId	The unique identifier for the user.	yes	1
username	The username chosen by the user.	no	1
email	The email address of the user.	no	1
password	The password for the user's account.	no	1
school	The school that user is attending.	no	1
major	The major that user is studying for.	no	1
preferenes	An object containing the user's study preferences.	no	0..*

**Operations:**

Name	Description	Is Query?	Is Polymorphic?
updateProfile	Updates the user's profile information and preferences.	no	no
getMatches	Returns a list of matches for the user.	yes	no

## 4. Architecture Design

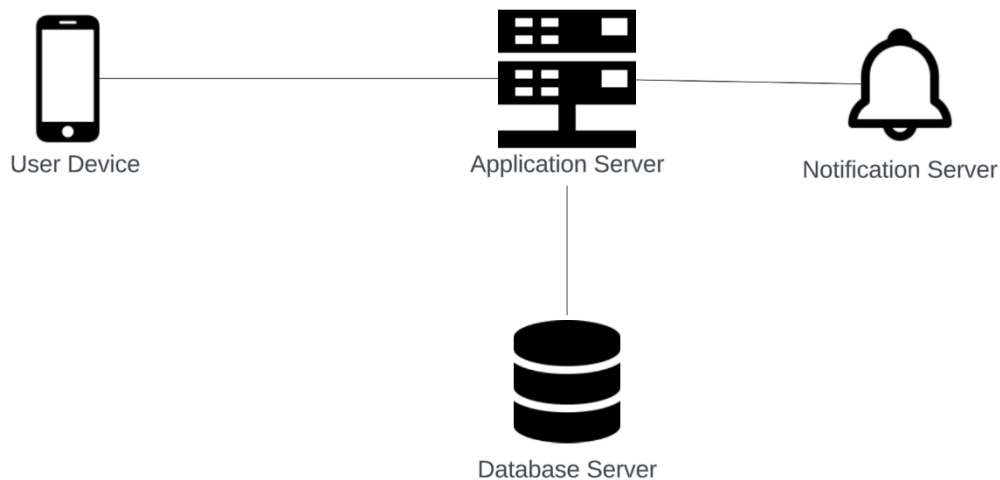
### 4.1. Architecture Overview

The Architecture Overview section provides a simplified outline of the Study Partner Matching app's system design. It presents a broad view of the system's components and their interactions. Additionally, it outlines the proposed architecture for the app and any potential new infrastructure needs.

This section serves as an introductory glimpse into the app's structural layout. It offers a fundamental understanding of how the app will be organized and operated. Essentially, it offers a foundational understanding before delving into the finer technical aspects. Thus, for those interested in the system's operational framework, this section serves as a useful primer.

### 4.2. Infrastructure Model

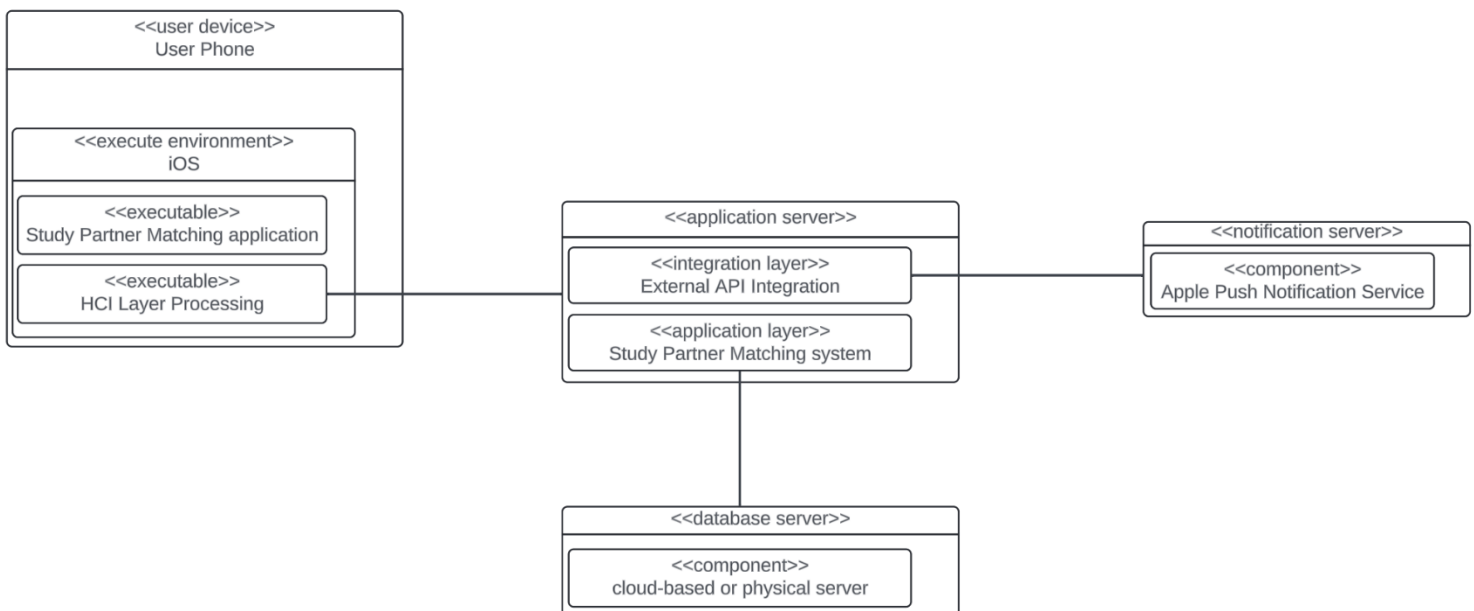
#### 4.2.1. Deployment Diagram 1 – Architecture Overview



[https://lucid.app/lucidchart/9fd89755-0955-4c69-8b61-](https://lucid.app/lucidchart/9fd89755-0955-4c69-8b61-7005c01df519/edit?viewport_loc=57%2C155%2C1529%2C893%2C0_0&invitationId=inv_acaf8848-59b0-4be0-b16c-c1c46adad6f7)

[7005c01df519/edit?viewport\\_loc=57%2C155%2C1529%2C893%2C0\\_0&invitationId=inv\\_acaf8848-59b0-4be0-b16c-c1c46adad6f7](https://lucid.app/lucidchart/9fd89755-0955-4c69-8b61-7005c01df519/edit?viewport_loc=57%2C155%2C1529%2C893%2C0_0&invitationId=inv_acaf8848-59b0-4be0-b16c-c1c46adad6f7)

#### 4.2.2. Deployment Diagram 2 – Nodes and Artifacts



[https://lucid.app/lucidchart/25dfd901-f21d-4605-95f8-f4dded6ad72e/edit?viewport\\_loc=-812%2C-722%2C2785%2C1391%2C0\\_0&invitationId=inv\\_e5436523-3b5f-46cb-b28a-02ed9d365d8f](https://lucid.app/lucidchart/25dfd901-f21d-4605-95f8-f4dded6ad72e/edit?viewport_loc=-812%2C-722%2C2785%2C1391%2C0_0&invitationId=inv_e5436523-3b5f-46cb-b28a-02ed9d365d8f)

## 4.3. Hardware and Software Requirements

### 4.3.1. Hardware Components

- **Application Server:** A server capable of running the application logic and handling user requests. This could be a cloud-based server or a physical server.
- **Database Server:** A server to host the database containing user information, preferences, and matching algorithms. This could also be a cloud-based or physical server.
- **User Devices:** Users will need an iOS device with internet connectivity to use Study Partner Matching.

The hardware requirements can be met by purchasing new servers or by using existing hardware if it meets the necessary specifications. Existing hardware can be reused to reduce costs if requirements are met. Otherwise, an alternative for the hardware is using virtual servers instead of physical servers or using cloud-based services for hosting. These alternative configurations can be considered based on factors such as cost, scalability, and ease of management.

### 4.3.2. Required Software Components

For the Study Partner Matching app, an iOS environment is needed for development and testing, focusing on iOS 14+ versions. This includes Xcode and iOS Simulator. The necessary software components are:

- **Xcode:** The main IDE for iOS app development, providing a code editor, interface builder, and testing tools, along with simulators for different iOS devices.
- **iOS Simulator:** Built into Xcode, it allows testing the app on various iOS devices virtually.
- **Visual Studio Code (VS Code):** A lightweight IDE with extensions to enhance productivity, particularly useful for backend services and scripts.
- **Git:** A version control system integrated with VS Code, facilitating easy tracking of changes and team collaboration.
- **Figma:** A design tool for wireframes, mockups, and visual interfaces, promoting collaboration and streamlining design handoff.
- **Node.js:** A JavaScript runtime for backend service development, essential for creating APIs and server-side logic.
- **Express.js:** A framework for Node.js, used to build the backend server and handle HTTP requests and responses.
- **MongoDB:** A NoSQL database for storing user data, preferences, and match information, chosen for its scalability and flexibility.

## 4.4. Security Plan

### 4.4.1. Security Overview

Study Partner Matching concerns include unauthorized access to user data, data breaches, and disruptions caused by cyber-attacks. Protecting user privacy is the most important, especially given the app's reliance on personal information to facilitate effective matching. The potential threats include:

- **Theft/Loss:** If a user loses their phone or gets stolen, unauthorized individuals may access the app and obtain sensitive user information.
- **Hardware Failures:** Hardware failures involve malfunctions or breakdowns of physical components like servers or network devices. This can lead to data loss, service interruptions, and the need for costly replacements.
- **MitM Attacks:** MitM attacks happen when someone intercepts communication between two parties, maybe changing or stealing data. This threatens the integrity and confidentiality of the information exchanged.
- **Service Disruptions:** Service disruptions can happen for various reasons like cyber-attacks, software bugs, or hardware issues. This affects how available and reliable the app is for users.
- **Malware:** Malware is bad software designed to damage or disrupt systems or gain unauthorized access. It can mess up user devices, leading to data theft or system problems.

- **Unauthorized Access:** Unauthorized access happens when someone who shouldn't be able to get into the system, does. This can mess up sensitive information and be used wrongly.
- **Data Breaches:** Data breaches involve unauthorized parties gaining access to confidential data, such as user personal information. This can result in privacy violations, identity theft, and legal repercussions for the organization.
- **Inadequate User Authentication:** Weak or not enough user authentication methods can let unauthorized people access the app. This messes up the security and can lead to potential data leaks.



#### 4.4.2. Security Plan

Threats Component	Physical		Network		Application			User Security
	Theft/Losses	Hardware Failures	MitM Attacks	Service Disruptions	Malware	Unauthorized Access	Data Breaches	Weak Authentication
User Devices	4				1			
Application Server				6		2,3		
Database Server		6					2,3,6	
User Data								1,2,3,6
Network			5,7,8					

#### Controls:

1. Encrypt sensitive data both at rest and in transit.
2. Regularly audit and monitor access logs.
3. Enforce strong password policies and implement multi-factor authentication.
4. Require strong authentication (e.g., biometrics, PINs).
5. Use encryption protocols (e.g., SSL/TLS) for data transmission.
6. Regularly update and patch all software components and conduct regular security audits and vulnerability assessments.
7. Implement secure network configurations and firewalls.
8. Regularly update and patch network devices and software and monitor systems continuously for early detection of issues.

## 5. User-Interface

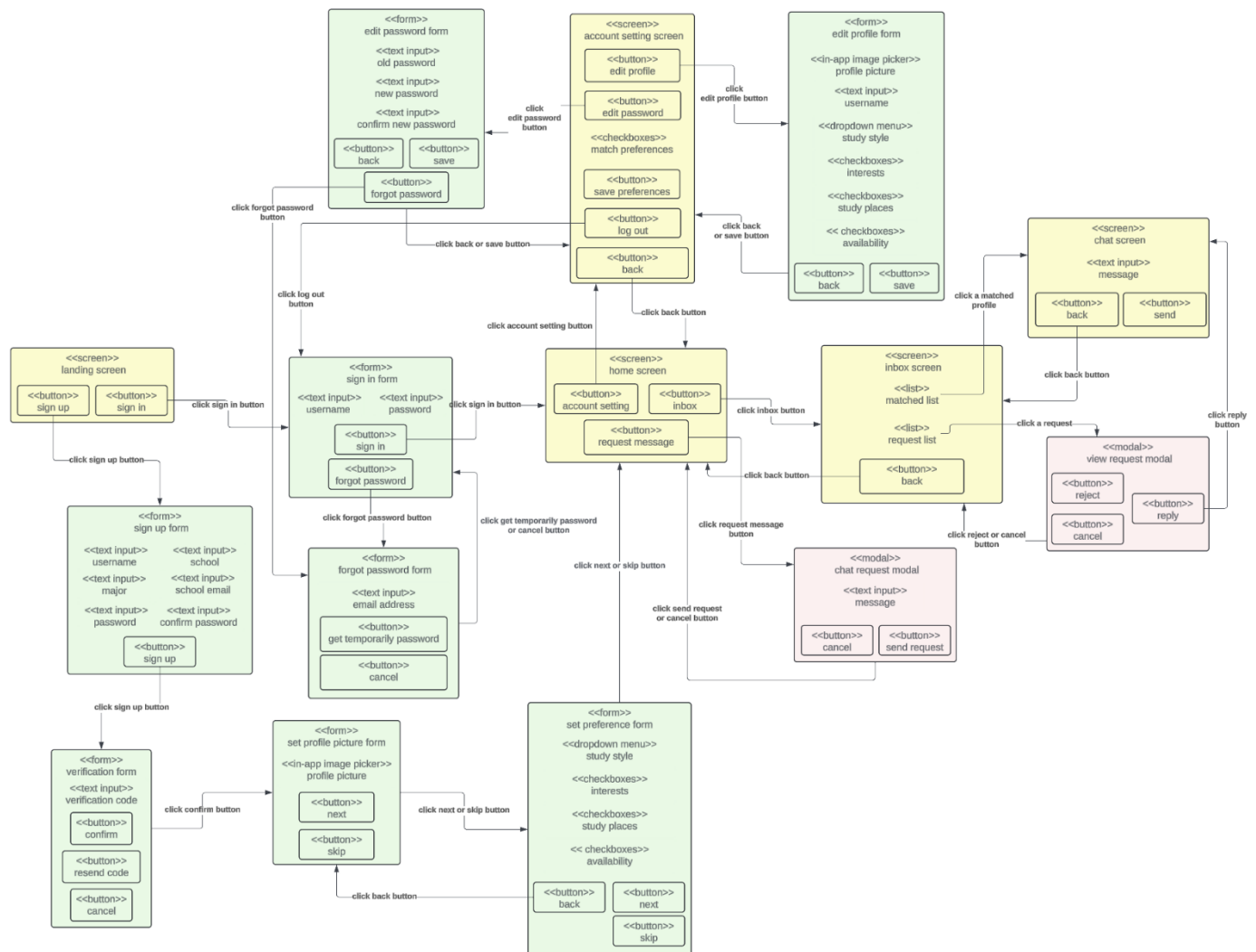
### 5.1. User-Interface Requirements and Constraints

In this section, we're going to talk about how the Study Partner Matching App should look and work for users. We want the app to be easy to use, helpful, and look good. Our goal is to make it simple for users to find study partners and use the app without confusion. We'll make sure the design is clear and consistent across all parts of the app.

Our focus is on keeping things easy to understand and use, so users don't have to think too hard about what to do. We'll also make sure the app works well for everyone, including those who might need extra help, like people using special devices. We want the app to be welcoming and useful for all kinds of users.

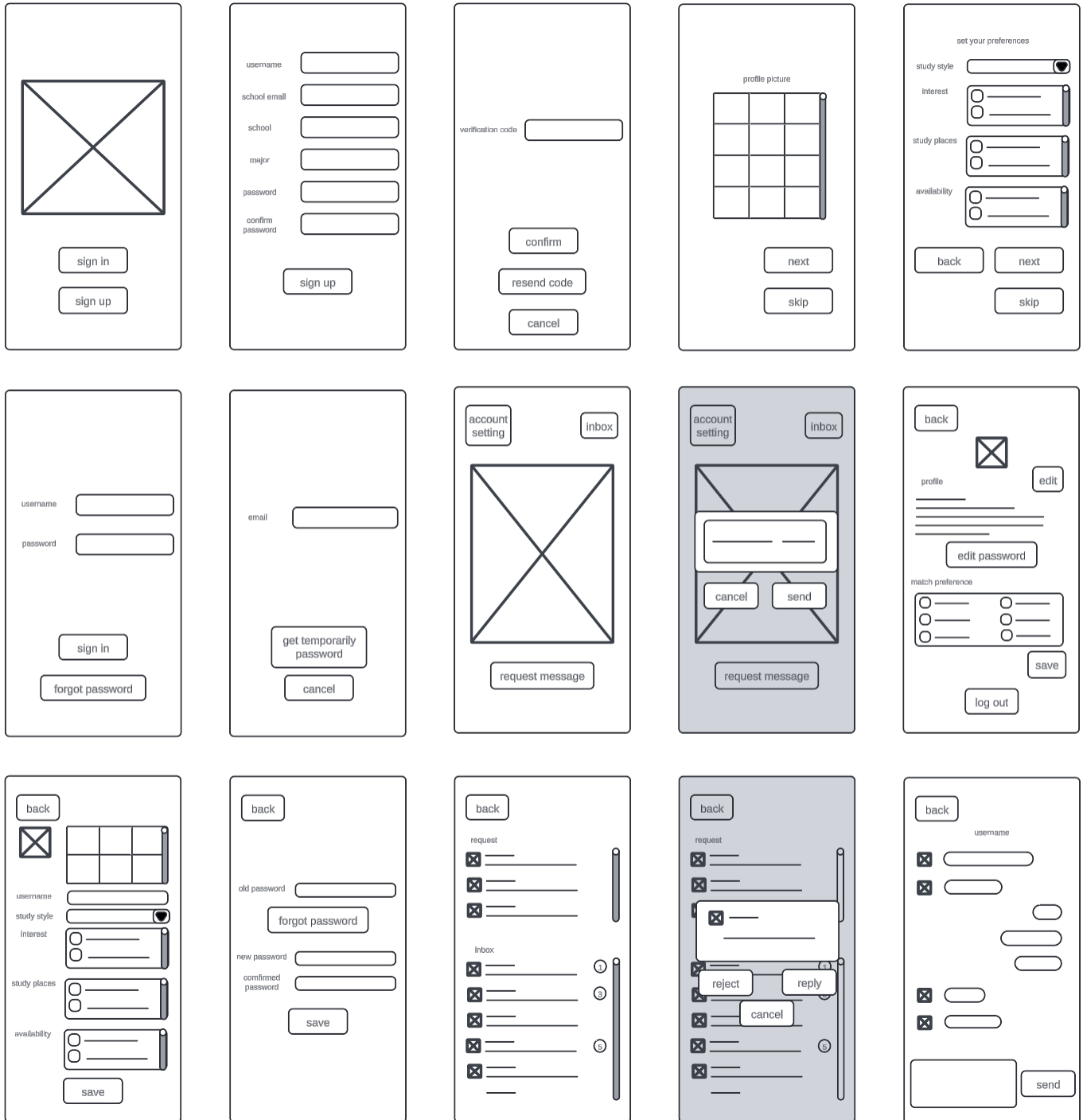
In the following pages, we'll talk about how the app will be laid out, how users will move around in it, and what it will look like. Everything we do is to make the Study Partner Matching App as user-friendly and effective as possible.

### 5.2. Window/Screen Navigation Diagram



[https://lucid.app/lucidchart/1e159b36-761f-4319-ae4e-6fa2b04c67b8/edit?viewport\\_loc=-1245%2C-755%2C5371%2C2296%2C0\\_0&invitationId=inv\\_1e865d96-6358-4266-a926-2d07919702ff](https://lucid.app/lucidchart/1e159b36-761f-4319-ae4e-6fa2b04c67b8/edit?viewport_loc=-1245%2C-755%2C5371%2C2296%2C0_0&invitationId=inv_1e865d96-6358-4266-a926-2d07919702ff)

### 5.3. UI Wireframes



[https://lucid.app/lucidchart/f22eaa56-15f7-49d9-ad96-5ab4a7bdc56b/edit?viewport\\_loc=-1460%2C-162%2C3575%2C1785%2C0\\_0&invitationId=inv\\_8fd1b682-098a-4029-b5be-e52c005b48b6](https://lucid.app/lucidchart/f22eaa56-15f7-49d9-ad96-5ab4a7bdc56b/edit?viewport_loc=-1460%2C-162%2C3575%2C1785%2C0_0&invitationId=inv_8fd1b682-098a-4029-b5be-e52c005b48b6)

## **6. Appendices**

### **6.1. Glossary**

N/A

### **6.2. References / Bibliography**

Larman, C. (2004). Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development, 3rd Edition. Upper Saddle River: Prentice Hall.

Cameron, Andrew. Class Lecture, System Design, Seattle Pacific University, Seattle, Washington, 2024.

### **6.3. Supporting documentation**

N/A