2024

StudySync Design Document

Team: Nilisha Bhandari, Sai Monish, Ryan Bui, Devan Quinn, Sophie Konger, Joseph Fleming

Team # 36

Purpose

StudySync aims to address the challenges faced by students and learners in managing and optimizing their study sessions. The primary problem is the lack of a comprehensive digital platform that integrates essential study tools such as user profile management, pomodoro timers, music integration, chatbots for assistance, collaborative note taking, flashcard creation, cooperative study rooms, leaderboards, customizable environments, an interactive tree to grow, and messaging board integration. This platform seeks to provide an all-in-one solution that enhances both individual and

group study experiences, boosts productivity, and fosters a more engaging and efficient learning environment.

Functional Requirements

StudySync's functional requirements cover a broad range, enabling users to personalize profiles, manage study sessions with a Pomodoro Timer, integrate a music player, utilize a chatbot, engage in collaborative activities like note-taking and study rooms, and participate in a study leaderboard. Customization options extend to the study environment, task lists, and messaging boards. Integration with Google Docs facilitates collaborative document editing. The unique Functional Study Tree incentivizes users with visual aids tied to study goals. Overall, the application provides a comprehensive and customizable platform for diverse study preferences and needs.

- **User Profile Management**
  - As a user, I would like to create and manage my profile so that I can personalize my study experience.
  - As a user, I would like to save my preferences in my profile to quickly access my favorite study tools.
  - As a user, being able to personalize my character or pick a default character will enhance my experience.
  - As a user, I want the ability to change my profile picture for a more personalized experience.
  - As a user, I would like the ability to set study goals in my profile to track my progress over time.
  - As a user, I want the option to reset my password in case I forget it.
  - As a student, I would like the option to turn on or off notifications sent to my phone.
- **Pomodoro Timer**
  - As a user, I would like to use a pomodoro timer to manage my study sessions effectively.

- As a user, I want to customize the duration of my pomodoro sessions to fit my preferred study intervals.
- As a user, I want the option to receive notifications/alerts at the end of each pomodoro session.
- As a user, having an easy to navigate dashboard will help me be able to focus on what I am doing at the time, and to navigate the website.
- As a user, I want to hear a distinct sound when a pomodoro session ends.
- As a user, I want a visual countdown display during my pomodoro sessions.
- As a user, I want a list to visualize which tasks I need to complete and track the amount of time it took me to finish.

- **Music Player Integration**
  - As a user, I want to integrate a music player with Spotify API to listen to music while studying.
  - As a user, I would like to see what my friends are listening to from the spotify API.
  - As a user, I would like to be able to access the spotify API from anywhere in the application.
- **Chatbot for Assistance**
  - As a user, I would like to interact with a chatbot for quick assistance and FAQs.
- **Collaborative Note Taking**
  - As a group study member, I want to participate in collaborative note taking to enhance group learning.
  - As a group study member, I want the option to highlight and annotate shared notes for better understanding.
- **Flashcards (Quizlet-style)**
  - As a student, I want to create and use flashcards for efficient self-study.
  - As a student, I want to be able to place images, texts, or audio recordings on my digital flash cards.
  - As a student, I want to be able to receive performance reports on study sessions with flash cards.

- As a student, I want to be able to share my flash cards with other students.
- As a student, I want to be able to make a quiz with the flashcards I made.

- **Cooperative Study Rooms**
  - As a user, I want to join cooperative study rooms for group study sessions.
  - As a user, I want to be able to participate in a private text conversation with other students in a study room.
  - As a user, I want to be able to personally invite my friends, such as with a link.
  - As a user, I want to give permissions to participants within study rooms to allow them to customize the room.
- **Leaderboard for Study Time**
  - As a competitive student, I would like to see a leaderboard showing study time to increase motivation.
  - As a student, I would like to see my stats of my study time directly displayed on my dashboard, and easy to see.
  - As a student, I would like to earn points by successfully studying flashcards, spending time in study rooms, or studying alone.
  - As a competitive user, I want the option to challenge friends to study competitions/compare our progress.
  - As a user, I want to be able to visualize the study time stats of users with multiple graphs.
- **Customizable Study Environment**
  - As a user, I want to customize my study environment to create a personalized and comfortable study space.
  - As a user, I want the option to choose from different background themes for my study space.
  - As a user, I want to be able to change the colors of my study space to better suit my tastes.
  - As a user, I want to be able to pick and choose what tools should appear on my study dashboard.
  - As a user, I should be able to create task lists in my study space with titles.

- As a user, I should be able to effectively navigate my task lists.

- **Messaging Board Integration**
  - As a user, I should be able to make a profile page with basic information about me, such as my name, my current study leaderboard points, or what classes I am in.
  - As a user, I should be able to set this profile to public or private.
  - As a group study member, I need a messaging board to communicate with peers during study sessions or find other students in a class.
  - As a user, I would like to be able to make a post viewable by other students.
  - As a user, I would like to be able to tag my posts so that other users can find my posts.
  - As a user, I would like to be able to search through posts by tag or topic.
  - As a user, I would like a forum board for my university. I should be able to get my university added to the list of forum boards.
  - As a message board moderator, I would like to be able to verify tags are accurate and valid.
  - As a moderator, I would like to verify posts are using correct tags and remove inappropriate posts.
  - As a moderator, I should be able to edit the tags on a post to better reflect what the post is actually about if the original poster used the incorrect tags.
  - As a user, I would like to be able to like posts or comment on them in threads.
  - As a moderator, I should be able to delete comments without deleting entire posts.
  - As a user, I want the ability to receive notifications for new posts or replies in the messaging board.
- **Integration with Google Docs**
  - As a user, I want to integrate with Google Docs for collaborative document editing and whiteboarding.
  - As a user, I should be able to seamlessly share a google doc with my current study room to facilitate collaborative working easier

and without having to type everyone's individual emails while keeping the document private to the current group.
- As a user, I should be easily able to navigate through the dashboard to find recent docs I have edited, and collaborated docs.

- **Functional Study Tree:**
  - As a student, I want to see a visual aid that will keep me locked in and incentivized to do my work.
  - As a student, I want to be able to grow a virtual garden of trees, with each tree being successfully grown after an uninterrupted study session.
  - As a user, I should be able to share my garden of trees with other users on my profile page.
  - As a student, I want to be able to set study goals to unlock different types of virtual trees.
  - As a user, I would like to see a status bar for my tree, and how much time it will take to get to the next level.

Non-Functional Requirements

The non-functional requirements for StudySync emphasize optimal performance, scalability, usability, and security. Performance goals include swift loading times, responsive chatbot interactions, and seamless music player integration. The system is designed to handle a substantial number of simultaneous requests for scalability. Usability is centered around an intuitive interface, cross-device compatibility, and customizable workspaces. Security measures prioritize the protection of user information, secure profile editing, stringent registration processes, and role-based permissions within study rooms.

- **Performance**

The app is designed to be super fast in different areas. The front-end should populate within 500 milliseconds. Additionally, the chatbot should be able to respond to inquiries within 2 seconds. When connecting StudySync to Spotify, playback should be smooth with

minimal delays so music can be enjoyed without any interruptions. All services should maintain a 99.9% uptime.

- **Scalability**

The system should handle up to 10,000 simultaneous requests. User and database capacity should automatically scale in a cloud infrastructure with as little downtime as possible. New components should be simplistic to implement and ship to users. Proper documentation should allow for bug-free modification if need be.

- **Usability**

The user interface should be intuitive and easy to navigate for the average user. Compatibility with various devices and browsers is important to ensure functions such as our collaborative notebook work as intended. Having numerous different study features allow our users to utilize different strategies in one website instead of needing numerous tabs and increasing clutter within their work space. Each user's workspace will be fully customizable so the user can be in a space where they will feel the most comfortable and productive.

- **Security**

The security of StudySync will protect various pieces of information that the user can add to their profiles. The user can edit this information through their own profile page only accessible to them. Implementing a registration that requires a password and other means of verification will further secure each user's data. Hosts of study rooms will be able to give participants roles in the study room to give the participants certain permissions.

Design Outline

Design Decisions (for example client-server model)

For our design, we adopted a client-server model to facilitate the interactions between the user, the database, and the application.

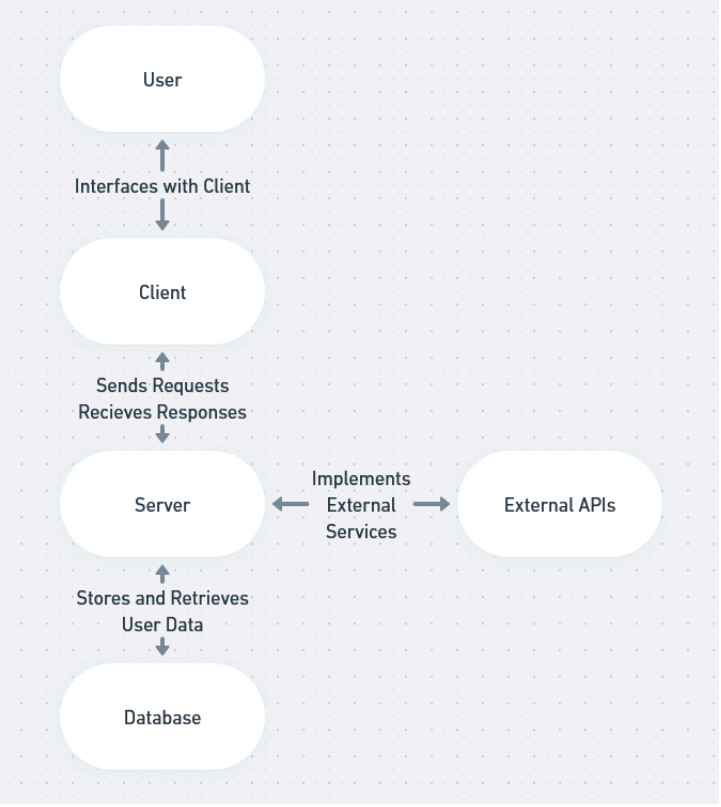
The user interacts only with the client. The client is implemented as a server frontend using React and Tailwind. The client sends requests to the server and receives responses.

The server interfaces with our database management system to handle data such as messages and user data. The server also interacts with our external APIs to deliver services to the client. The server is implemented with Node.js and Express.

The database stores all user data by responding to update from the server. It also fulfills all queries sent by the server and delivers the data requested back to the server.

External APIs, such as our chatbot or Spotify integration, communicate with the server to implement the functionality

A high-level overview of these core components are outlined below.



System Components Overview

The selection of system components for StudySync was driven by the need for a robust, scalable, and user-friendly platform capable of

integrating various study tools. Visual Studio Code was chosen for its comprehensive support for the technology stack used, including JavaScript, React, and Node.js. Git and GitHub provide essential version control and collaboration features. Supabase offers an open-source alternative to Firebase, simplifying database management with PostgreSQL. Postman assists in API development by enabling easy testing of API endpoints. The frontend leverages React for dynamic interfaces and Tailwind CSS for its utility-first styling approach. Node.js and Express form the backbone of the server-side application, ensuring scalability and efficient API development. Project management is streamlined through Notion, while Jest facilitates testing. Docker and Kubernetes address containerization and orchestration needs, ensuring the application's scalability and reliability. Monitoring with Datadog ensures optimal performance, and integration with Spotify and ChatGPT APIs enhances the user experience with music and chatbot functionalities.

Development Environment:

- Visual Studio Code: A powerful editor with support for JavaScript, React, Node.js, and a multitude of extensions for different technologies and languages

Version Control:

- Git: Essential for version control
- Github: For hosting the repository, and for tools like issue tracking, code review, and CI/CD pipeline

Database:

- Supabase: open-source alternative to Firebase. Includes database management, authentication, and uses PostgreSQL. Stores all relevant data and interacts with server backend to deliver it to the frontend as needed.

API Development:

- Postman: to test API endpoints

Frontend Development:

- React: Utilized for building a dynamic and responsive user interface.
- Tailwind CSS: Chosen for styling due to its utility-first approach.

Backend Development:

- Node.js and Express: Used for creating scalable and fast server-side applications. Express simplifies routing, middleware integration, and API development

Project Management:

- Notion

Testing:

- Jest: For unit and integration testing of React and Node.js application

Containerization and Orchestration:

- Docker: For creating containerized applications
- Kubernetes: For orchestrating and managing containerized applications to scale

Monitoring and Performance:

- Datadog: monitoring service that provides insights into application's performance.

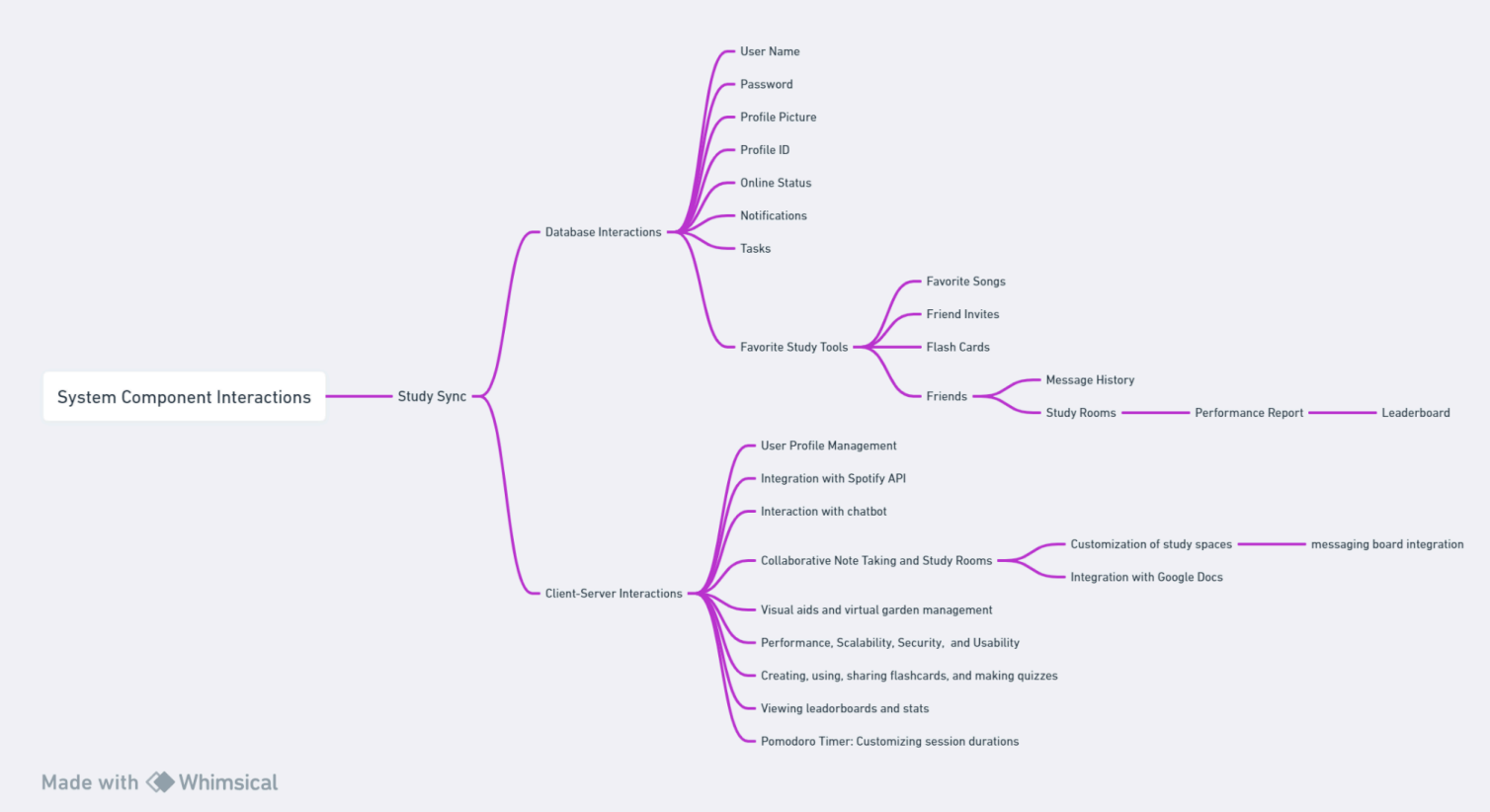
API integration:

- Spotify API: For music player integration, allowing users to listen to music, access playlists, and see what friends are listening to.
- ChatGPT API: To power the chatbot for assistance, leveraging AI to provide users with quick answers to queries and facilitate interactive learning.

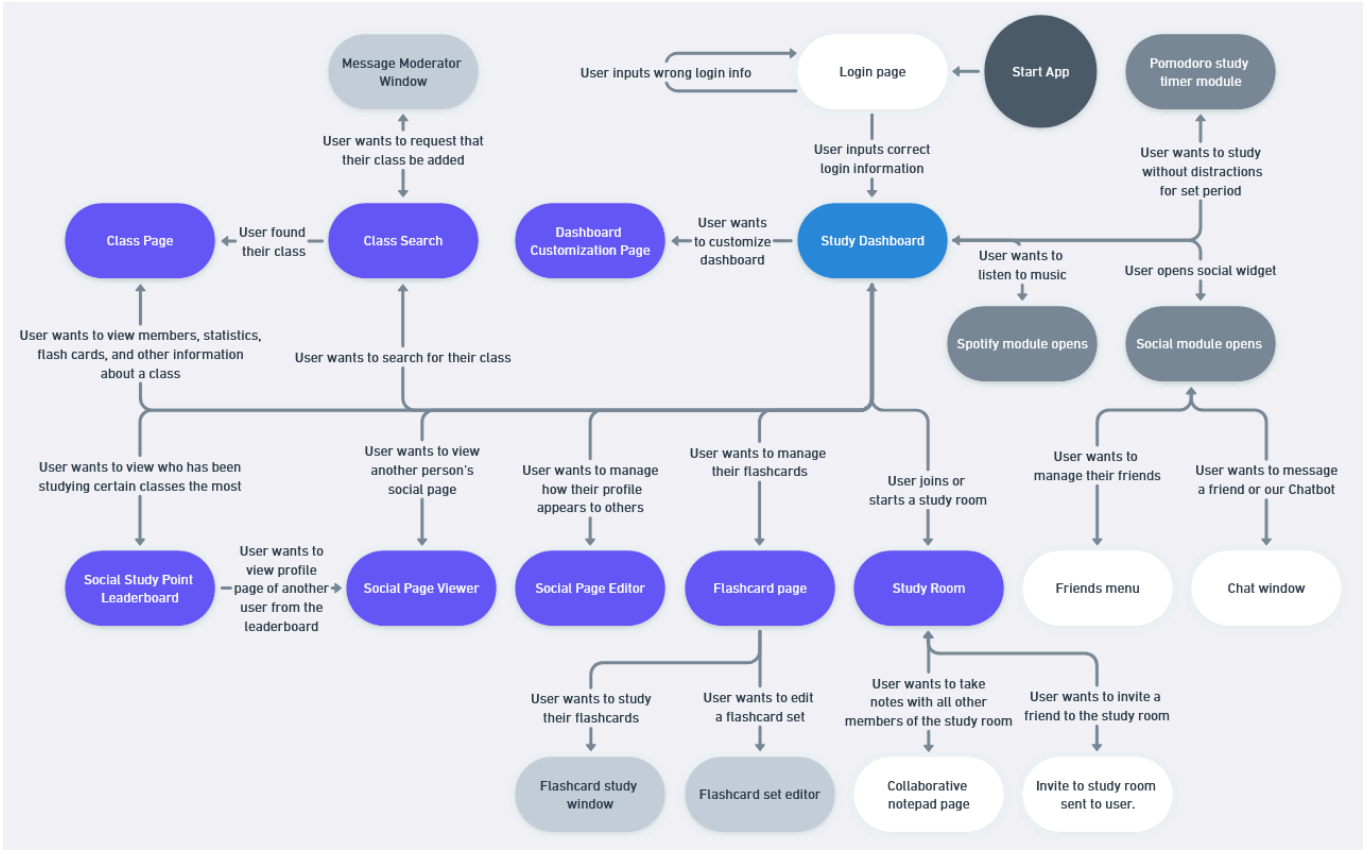
System Component Interactions

The StudySync platform is designed with a complex but cohesive system of component interactions to ensure a seamless user experience across various study tools and features. At the heart of these interactions is the client-server model, where the user-facing

client, developed using React and Tailwind CSS, communicates with a backend server built on Node.js and Express. This server acts as the intermediary between the frontend, the Supabase database, and external APIs like Spotify for music integration and ChatGPT for chatbot assistance. Users interact with the system through a responsive UI, managing their study sessions, profiles, and interactions with others. For instance, when a user initiates a Pomodoro timer, the request is processed by the server, which then updates the user's dashboard and study statistics stored in the database. Similarly, collaborative features such as study rooms and flashcard sharing involve real-time data exchange between participants, managed by the server to ensure data consistency and provide a collaborative environment. This integrated network of components supports the platform's goal of providing a comprehensive, customizable, and interactive study experience, leveraging the strengths of each component to deliver a wide range of functionalities from time management to social engagement.



State Diagram



Design Issues

1. **UI Complexity:**
    - **Issue:** With features like pomodoro timers, music integration, etc., there's a risk of overwhelming the user with a complex UI.
    - **Mitigation:** Prioritize simplicity and user-friendly design. Consider customizable interfaces, option toggles, and clear navigation to ensure users can easily access and manage features without feeling overwhelmed.
2. **Accessibility and Inclusivity:**

- **Issue:** A diverse user base may face challenges without an inclusive design.
- **Mitigation:** Implement accessible design principles, such as adjustable text sizes, color contrasts, and compatibility with screen readers. Consider user feedback and conduct accessibility testing to ensure the platform is usable by individuals with different needs and abilities.

3. **Data Security and Privacy:**
   - **Issue:** User profiles and collaborative features raise concerns about the security and privacy of user data.
   - **Mitigation:** Employ robust encryption methods for data transmission and storage. Implement strict access controls, conduct regular security audits, and ensure compliance with data protection regulations. Clearly communicate the platform's privacy policy to users.
4. **Performance and Responsiveness:**
   - **Issue:** Multiple features may lead to performance issues, especially on various devices and network conditions.
   - **Mitigation:** Optimize code for efficiency, use lazy loading for non-essential features, and conduct thorough testing on different devices and network speeds. Implement responsive design to ensure a consistent user experience across various screen sizes and resolutions. Regularly update and optimize the platform for improved performance.

Design Details

The Class Design in StudySync encapsulates the functionalities and interactions among different components of the application, ensuring modularity and ease of maintenance. The User Class forms the foundation, managing authentication, profile updates, and social

interactions. The Dashboard Class extends User functionalities to manage study tools like the Pomodoro timer and music preferences. Flashcard Class offers features for creating and using flashcards, embodying self-study tools. The Study Room Class facilitates group study sessions, integrating collaborative tools and document sharing. Tree Class introduces gamification by visualizing study progress through tree growth. Leaderboard Class promotes competition and motivation by tracking and displaying user achievements. The Message Class underpins the communication features, enabling messaging within study rooms and across the platform. Notifications Class ensures users stay informed about important events and interactions. This design supports a cohesive and comprehensive study platform, enabling a wide range of functionalities to enhance the learning experience.

Class Design

User Class

- Attributes: `profileID`, `password`, `stats`, `friendsList`, `friendInvites`, `messageHistory`, `profileCustomization`, `permissions`, `groups`, `onlineStatus`
- Methods: `login()`, `logout()`, `updateProfile()`, `sendFriendInvite()`, `acceptFriendInvite()`, `updateOnlineStatus()`

Dashboard Class (extends User)

- Attributes: `pomodoroTimerSettings`, `todoList`, `musicPreferences`
- Methods: `startPomodoro()`, `addTodoItem()`, `removeTodoItem()`, `setMusicPreferences()`

Flashcard Class (extends User)

- Attributes: `flashcardSets`, `timeStudied`, `accuracy`
- Methods: `createSet()`, `addFlashcard()`, `studySet()`, `trackStudySession()`

Study Room Class (extends User)

- Attributes: `studyRoomID`, `participants`, `roomCustomization`, `documentLinks`
- Methods: `createRoom()`, `inviteToRoom()`, `customizeRoom()`, `shareDocument()`

Tree Class (extends User)

- Attributes: `trees`, `statusBar`
- Methods: `growTree()`, `checkTreeStatus()`

Leaderboard Class (utilized by Study Room)

- Attributes: `leaderboardData`
- Methods: `updatePoints()`, `retrieveLeaderboard()`

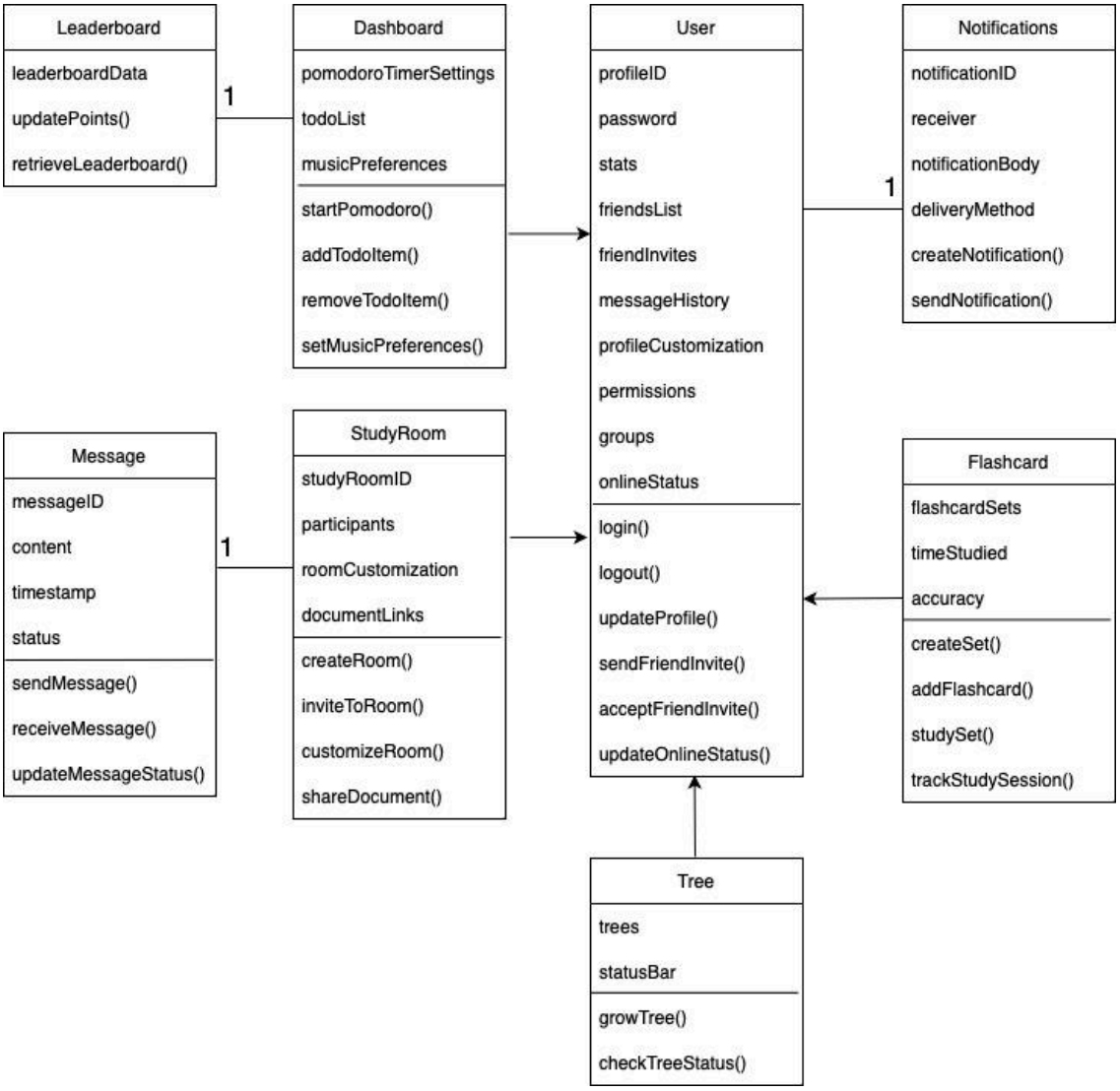
Message Class (extends User)

- Attributes: `messageID`, `content`, `timestamp`, `status`
- Methods: `sendMessage()`, `receiveMessage()`, `updateMessageStatus()`

Notifications Class

- Attributes: `notificationID`, `receiver`, `notificationBody`, `deliveryMethod`
- Methods: `createNotification()`, `sendNotification()`

Class and Inheritance Diagram

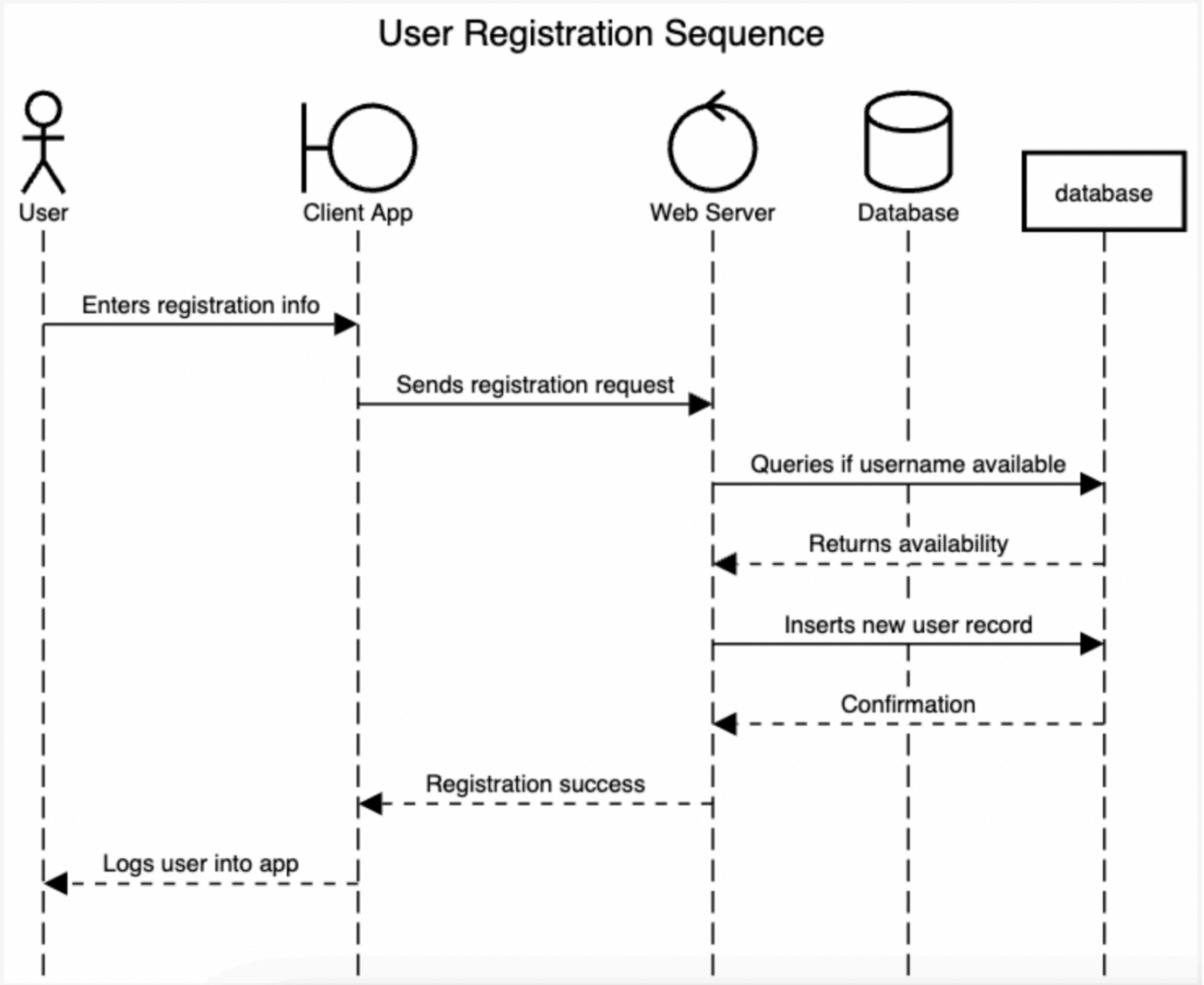


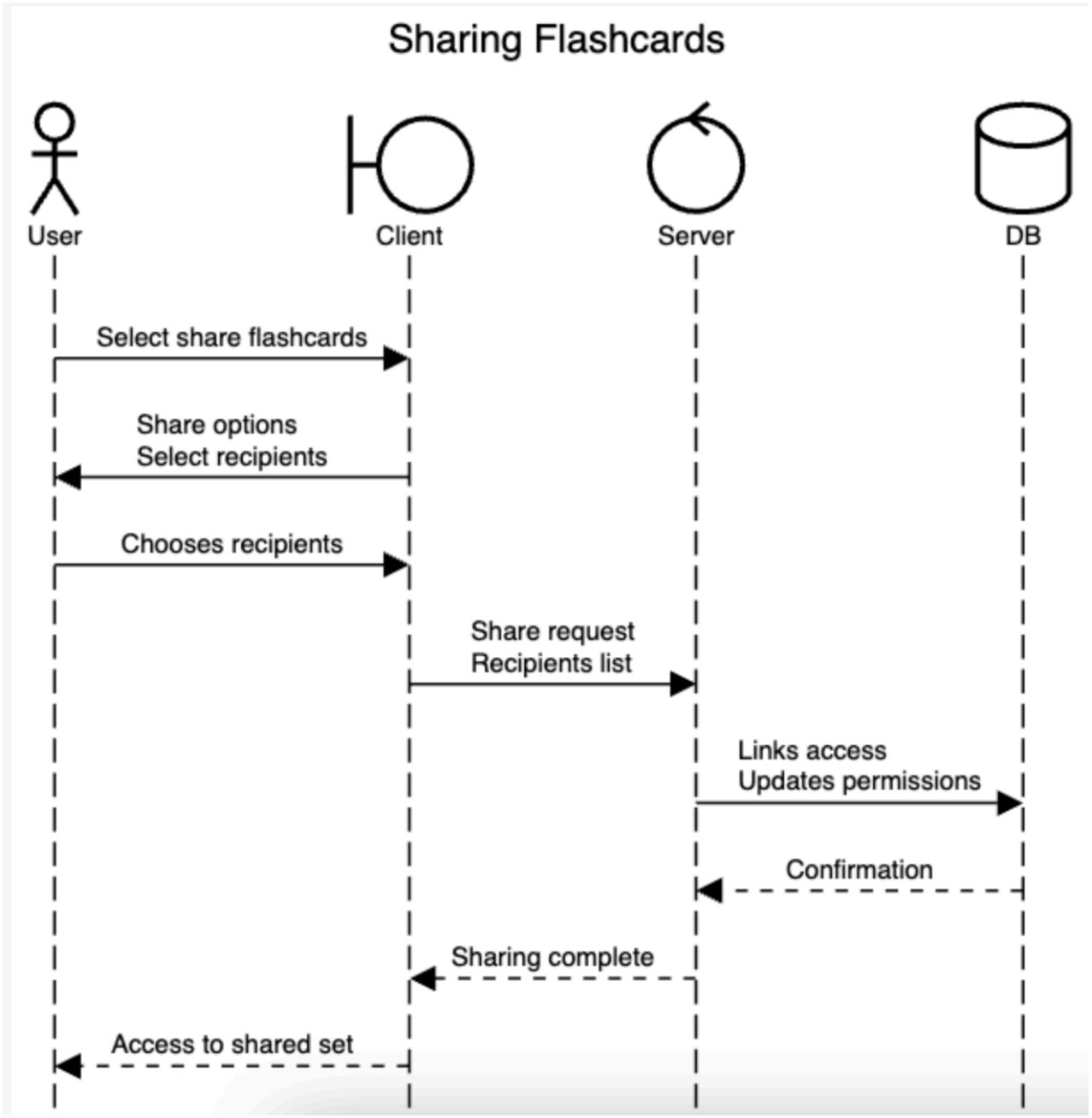
Interactions Between Classes

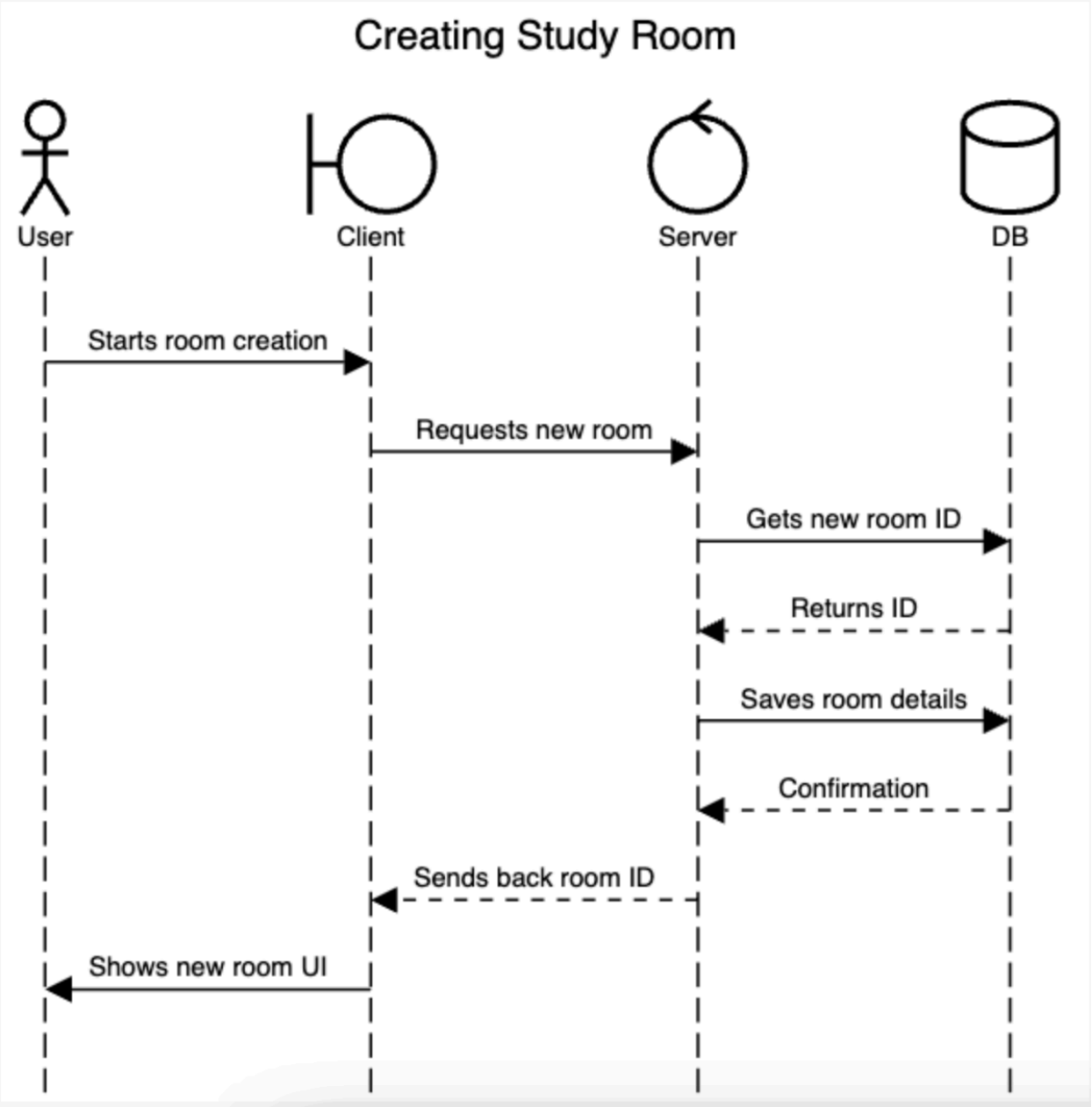
- User Authentication and Profile Management: The User class is central, with other classes extending it to inherit basic user functionalities like authentication and profile management.
- Dashboard and Study Tools: The Dashboard class interacts with the Pomodoro, TodoList, and Music API (through Spotify API integration) to provide a personalized study environment.

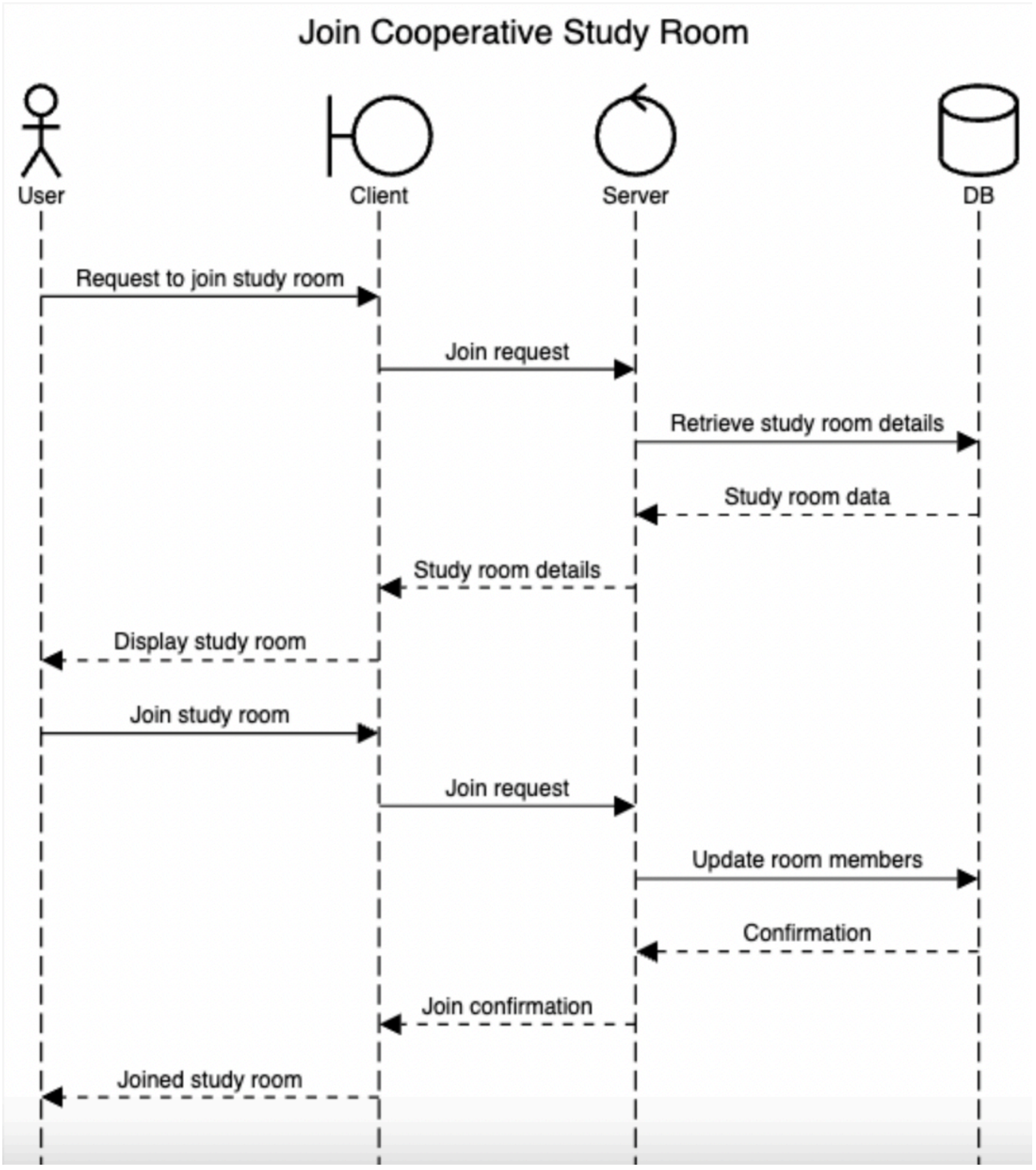
- Collaborative Study: The Study Room class uses the Message class for communication among participants and integrates with Google Docs for collaborative document editing.
- Flashcards and Study Sessions: The Flashcard class allows users to create, study, and share flashcards, tracking study time and accuracy.
- Growth and Motivation: The Tree class is linked to study sessions, incentivizing users by showing progress through virtual tree growth.
- Competition and Engagement: The Leaderboard class encourages competition by tracking and displaying user points earned through various study activities.

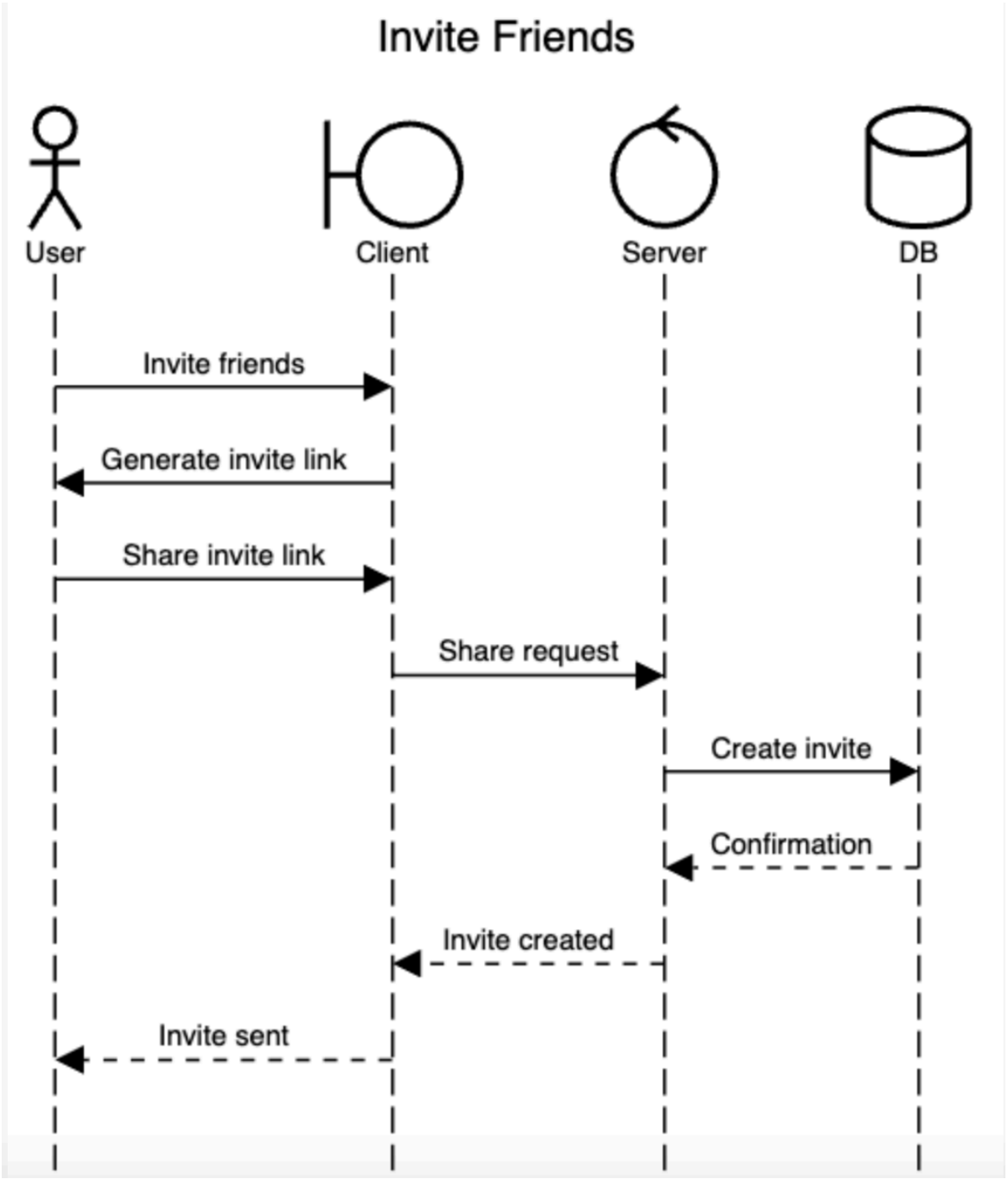
Sequence Diagrams

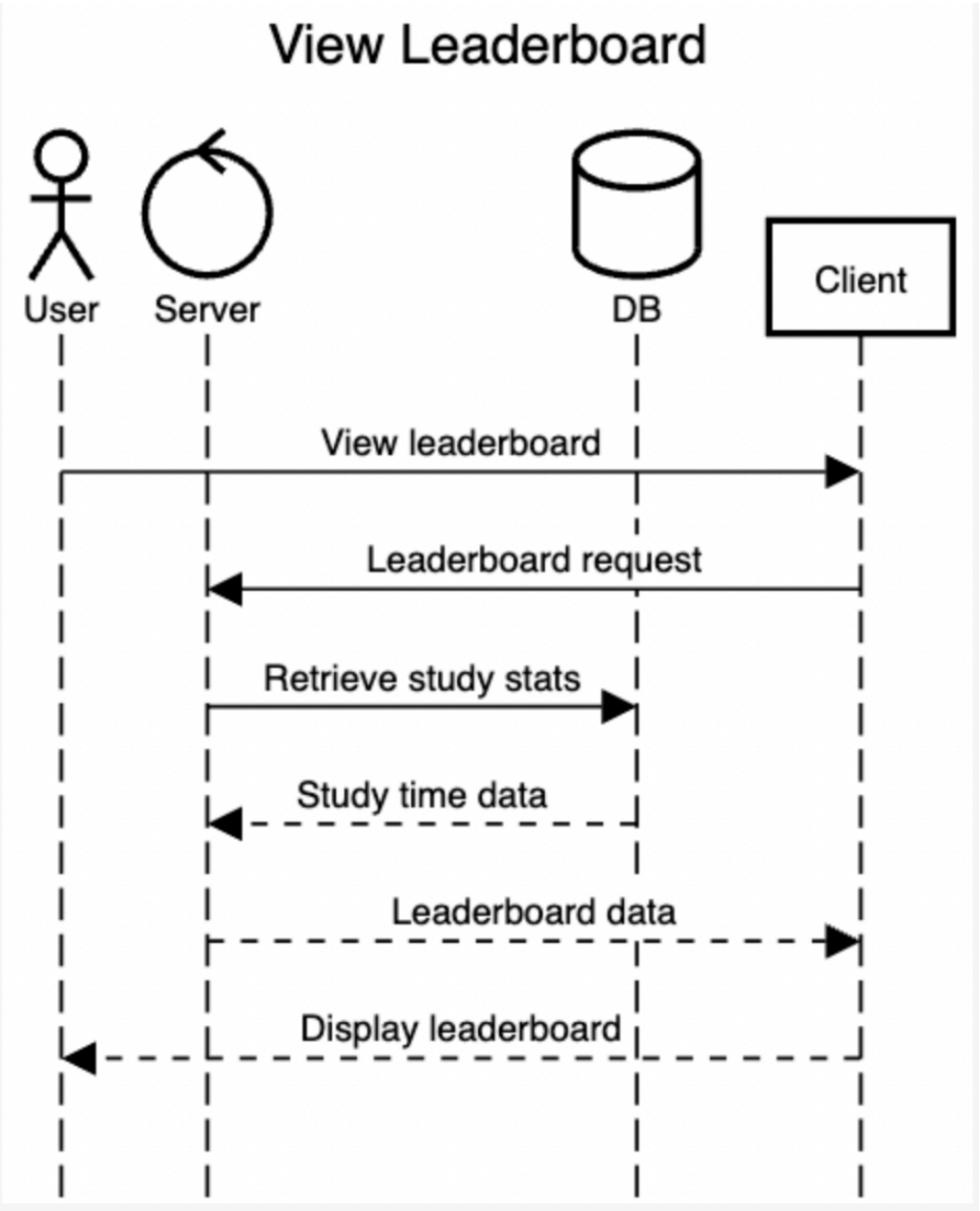


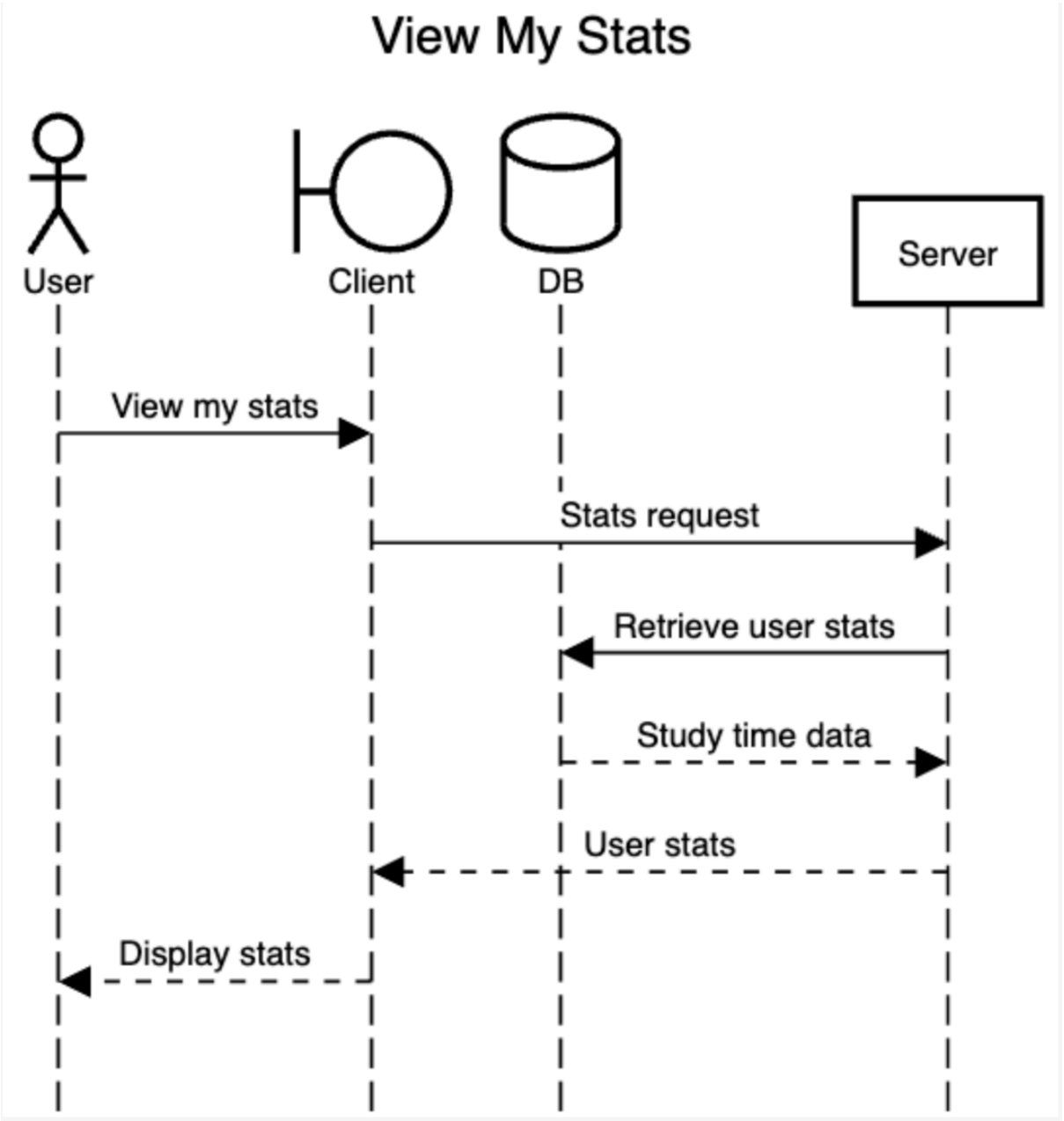

Sharing Flashcards
User
Client
Server
DB
Select share flashcards
Share options
Select recipients
Chooses recipients
Share request
Recipients list
Links access
Updates permissions
Confirmation
Sharing complete
Access to shared set

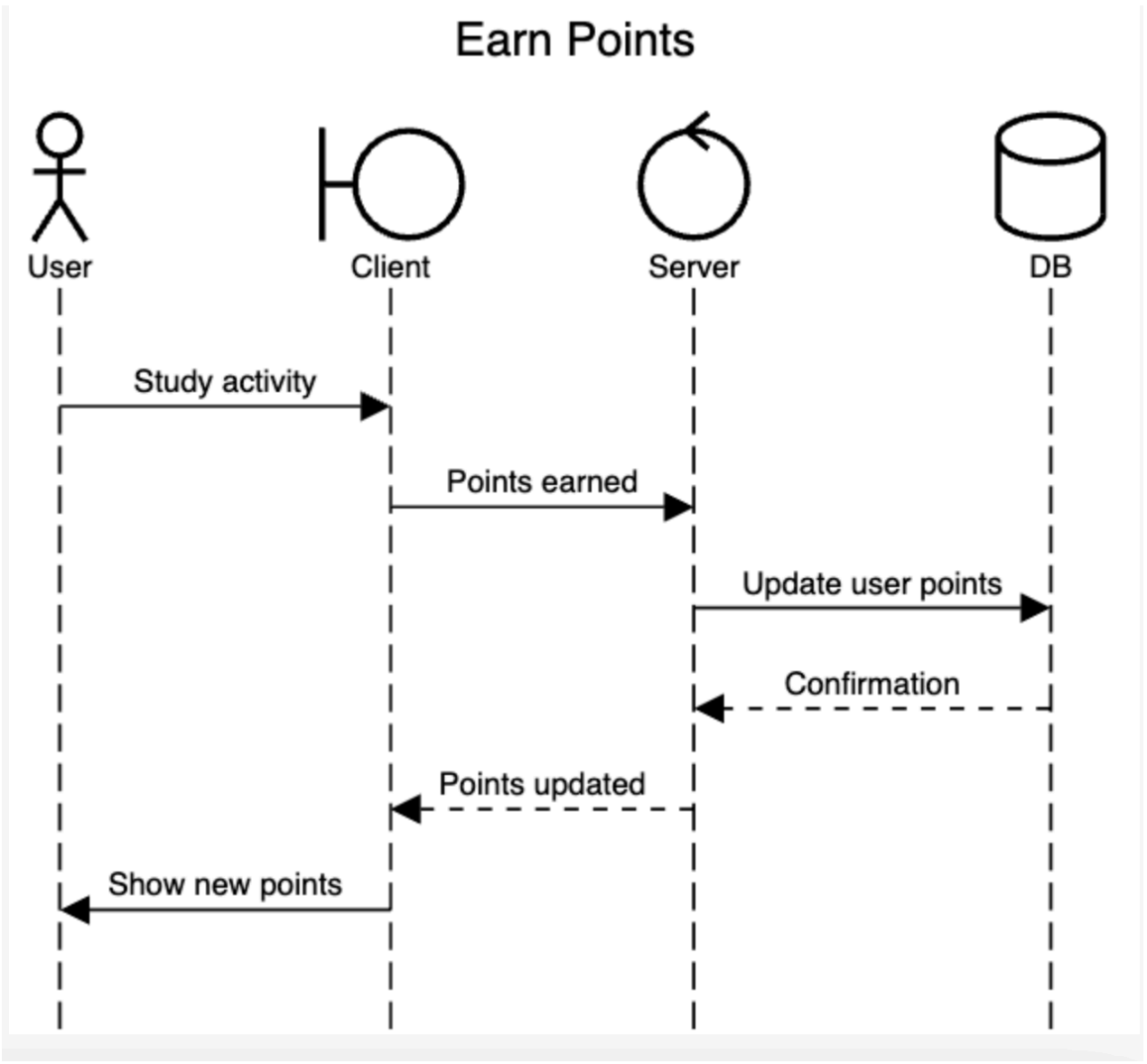

Creating Study Room
User
Client
Server
DB
Starts room creation
Requests new room
Gets new room ID
Returns ID
Saves room details
Confirmation
Sends back room ID
Shows new room UI

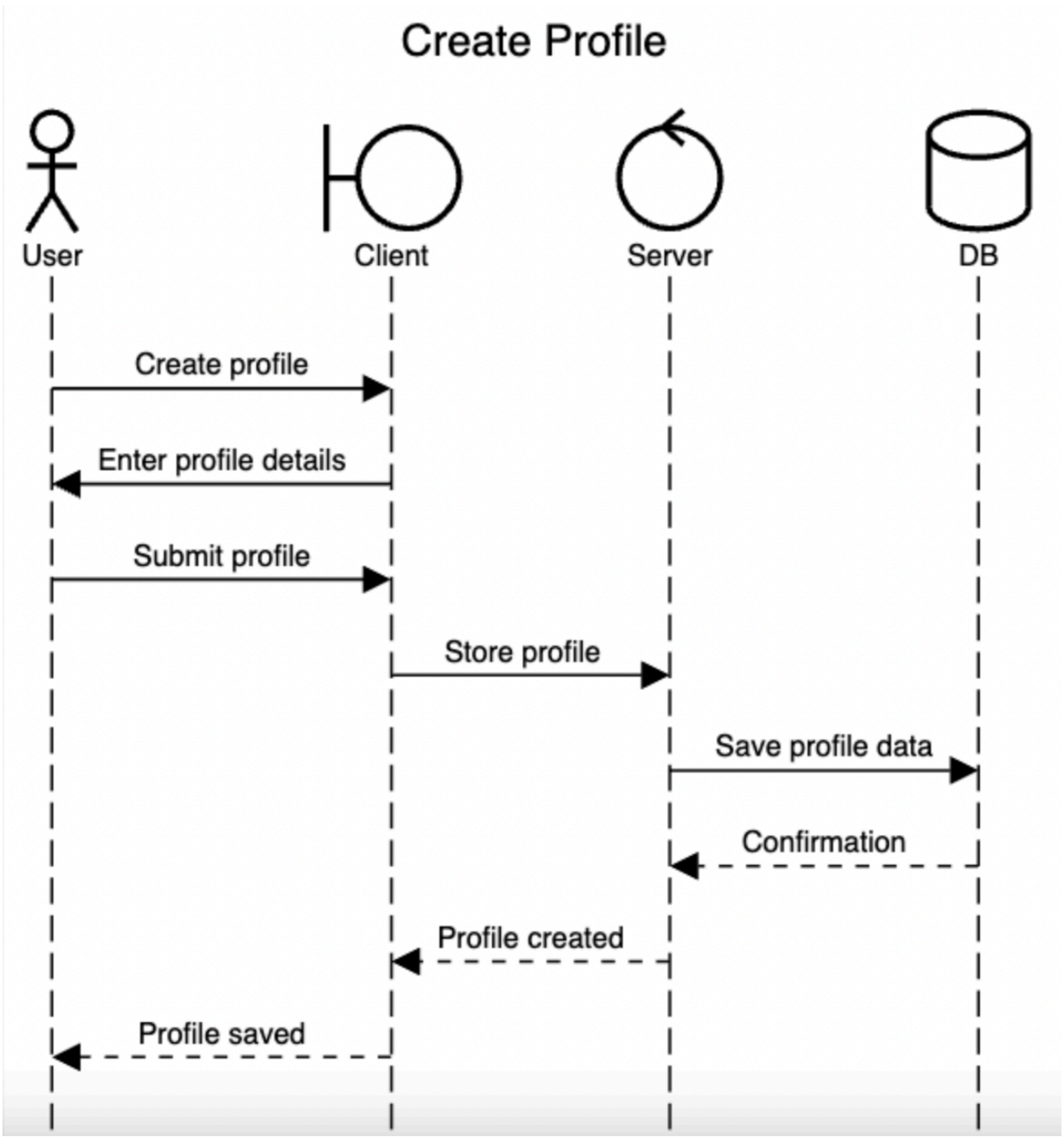

Join Cooperative Study Room
User
Client
Server
DB
Request to join study room
Join request
Retrieve study room details
Study room data
Study room details
Display study room
Join study room
Join request
Update room members
Confirmation
Join confirmation
Joined study room

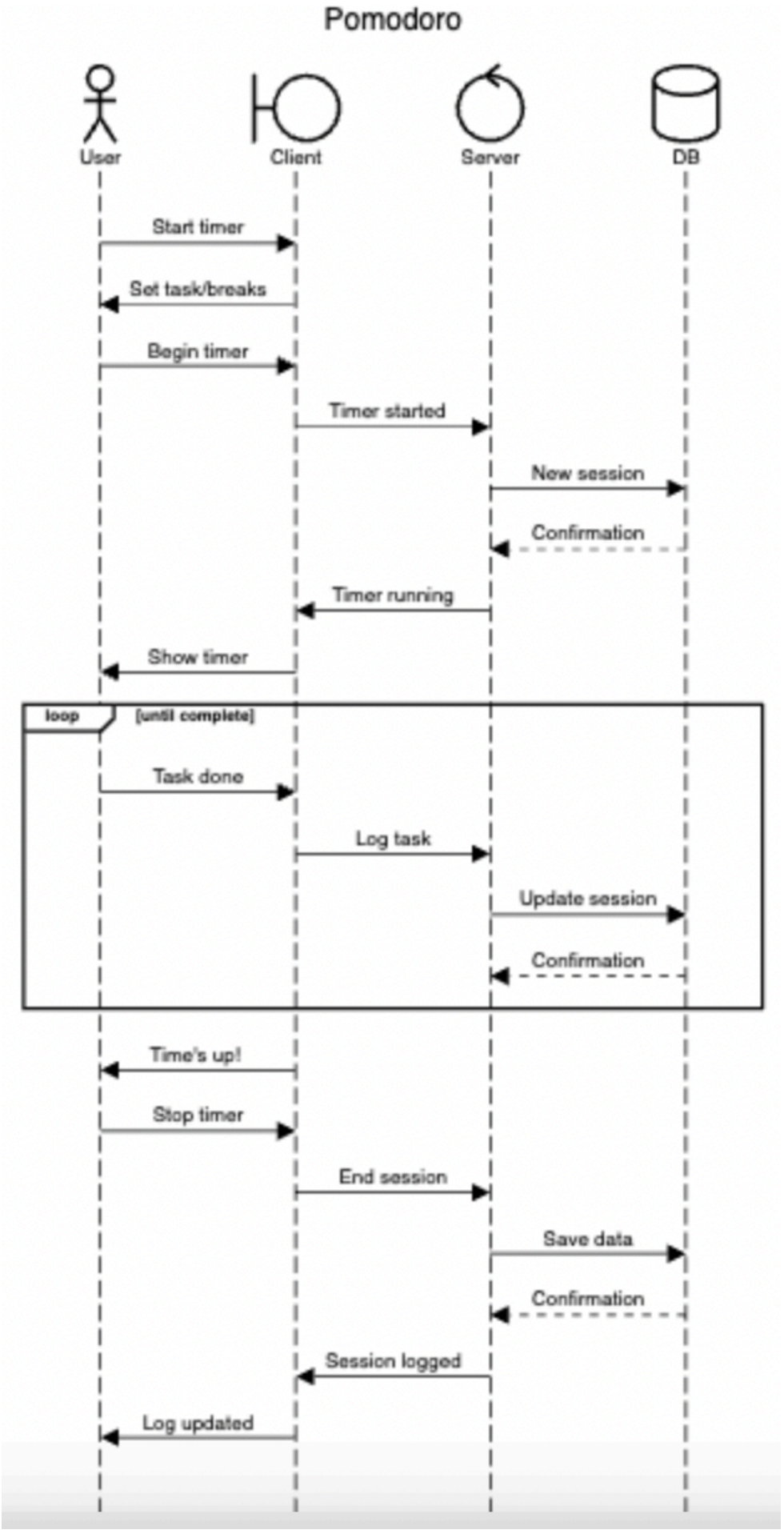

Invite Friends
User
Client
Server
DB
Invite friends
Generate invite link
Share invite link
Share request
Create invite
Confirmation
Invite created
Invite sent


View Leaderboard
User
Server
DB
Client
View leaderboard
Leaderboard request
Retrieve study stats
Study time data
Leaderboard data
Display leaderboard


View My Stats
User
Client
DB
Server
View my stats
Stats request
Retrieve user stats
Study time data
User stats
Display stats

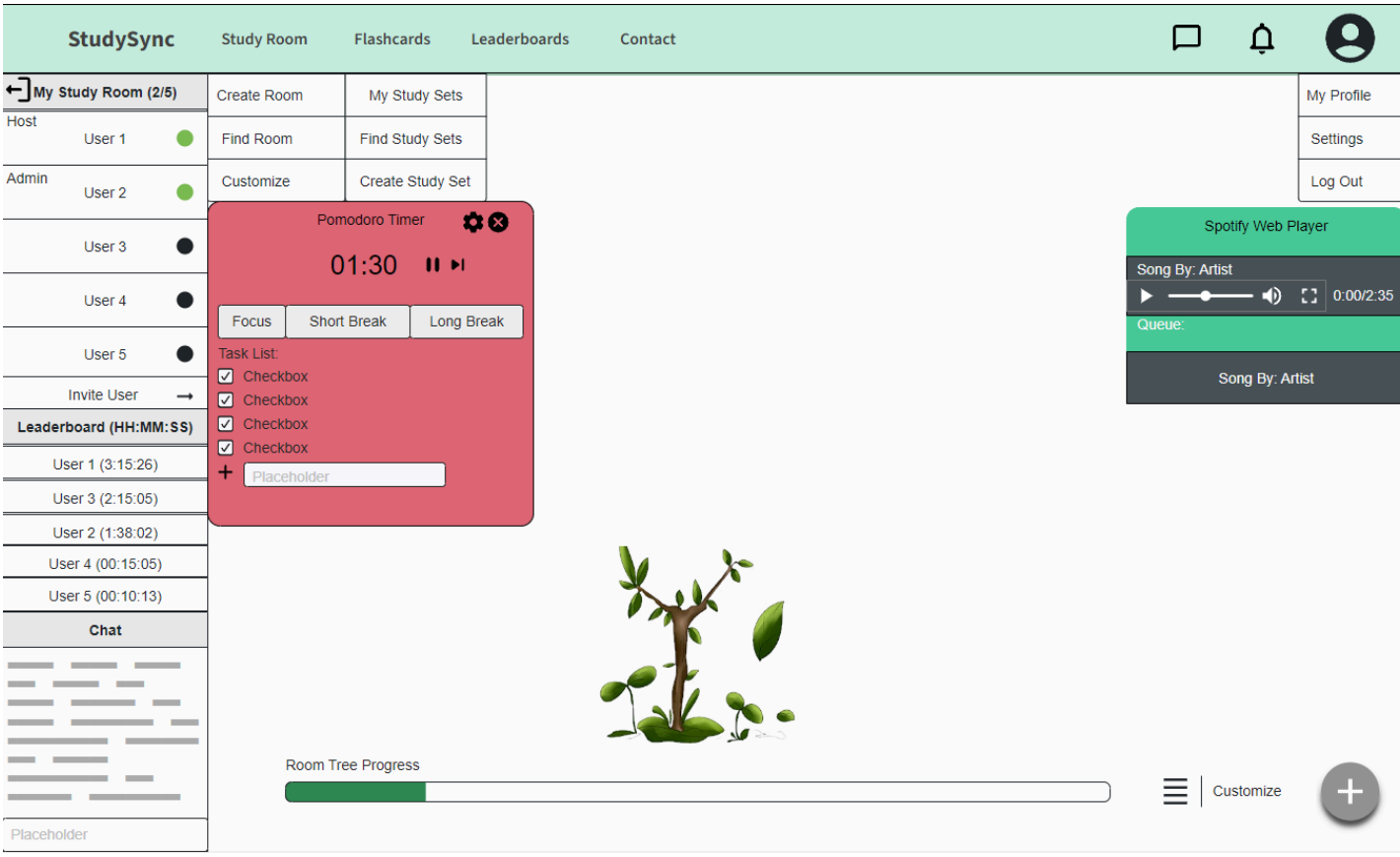

Earn Points
User
Client
Server
DB
Study activity
Points earned
Update user points
Confirmation
Points updated
Show new points


Create Profile
User
Client
Server
DB
Create profile
Enter profile details
Submit profile
Store profile
Save profile data
Confirmation
Profile created
Profile saved


Pomodoro
User
Client
Server
DB
Start timer
Set task/breaks
Begin timer
Timer started
New session
Confirmation
Timer running
Show timer
loop
[until complete]
Task done
Log task
Update session
Confirmation
Time's up!
Stop timer
End session
Save data
Confirmation
Session logged
Log updated

UI Mockups

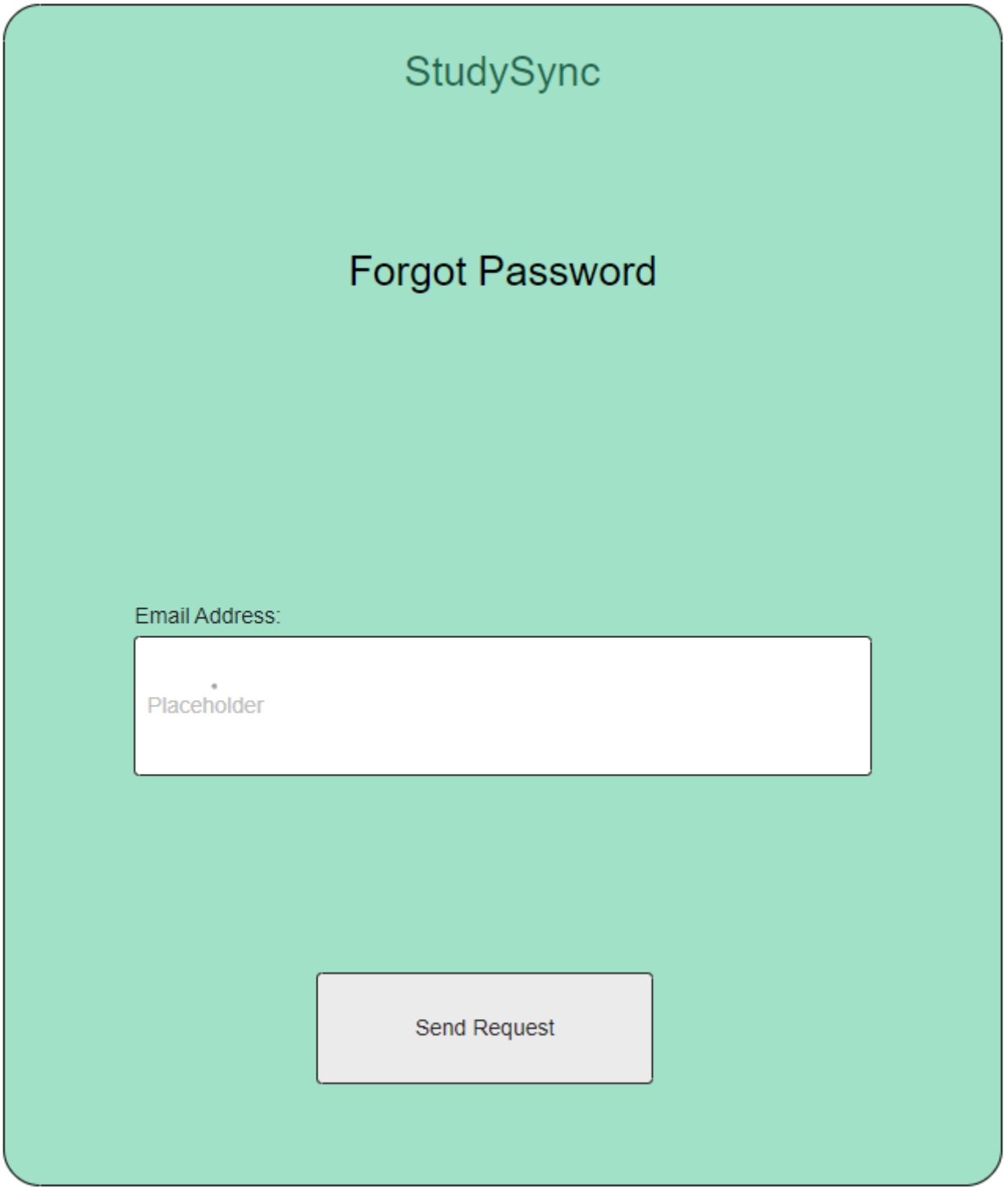
This is the mock study room, the plus button in the corner will allow the user to add numerous widgets on the screen such as the Pomodoro Timer and the Spotify web player. The customize button will allow the user to change the background of their own study space whether that be importing an image or selecting stock backgrounds in the web app. Study room, flashcards, chats, messages, and the user profile will all have dropdown menus to display relevant data/choices. The AI chatbot will be stored within the dropdown menu of the messages in the top right corner.



These are the mockups for the user creation/log-in pages. They will be prompted to enter a username and a password if they are already registered, if not there is a hyperlink to creating a new account which will display the sign-up page. If the user forgets their password, we will request the email address with which the account was created.


StudySync
Log-In
Username:
Placeholder
Password:
Placeholder
New To This Site
Forgot Password?
Log In


StudySync
Sign-Up
Username:
Placeholder
Email:
Placeholder
Password:
Placeholder
Confirm Password:
Placeholder
Register



This is the mockup for creating a study set, this will function similarly to how Quizlet created their study set by also allowing their users to import images into their flash cards. Each set will have a name and a description attached to it. The user will modify the amount of flashcards there are in their set by using the plus button at the

bottom to create more flash cards. Each flashcard will have a term/question, and the corresponding definition/answer.