ClassPlus

Design Document

Group 4

Kevin Chen

Esther Chong

Tristan Hall

Ben Jansson

Eunyoung Kim

Perry Lam

Dec 8, 2023

Table of Contents

1. Project Overview

1-1. Problem Description, Solution, and Stakeholders

Problem Description

UNCC students frequently turn to Discord for communication with their classmates; however, gaining entry to class sessions proves challenging without an invitation. The search for study partners is common, yet the features provided by UNCC for this purpose are not widely embraced. Moreover, students want to review study sets, and although platforms such as Quizlet offer an array of options from different schools, there is a distinct yearning for personalized study sets tailored to each specific class.

Problem Solution

At ClassPlus, accessible study sessions are facilitated through a user-friendly platform where students can effortlessly create or join study sessions for diverse subjects. This approach ensures inclusivity and collaboration among all students. The study group feature allows users to engage in conversations with group members, providing the flexibility to choose study group locations and times. This seamless connection process fosters an environment where students can easily find study partners, ultimately enhancing their overall learning experience. Additionally, the platform empowers users to share notes and post subject-related questions on posting boards. By enabling students to create personalized study sets tailored to their specific courses, the platform optimizes retention and comprehension, promoting effective and personalized learning strategies.

Stakeholders

UNCC Student, Instructor and TA

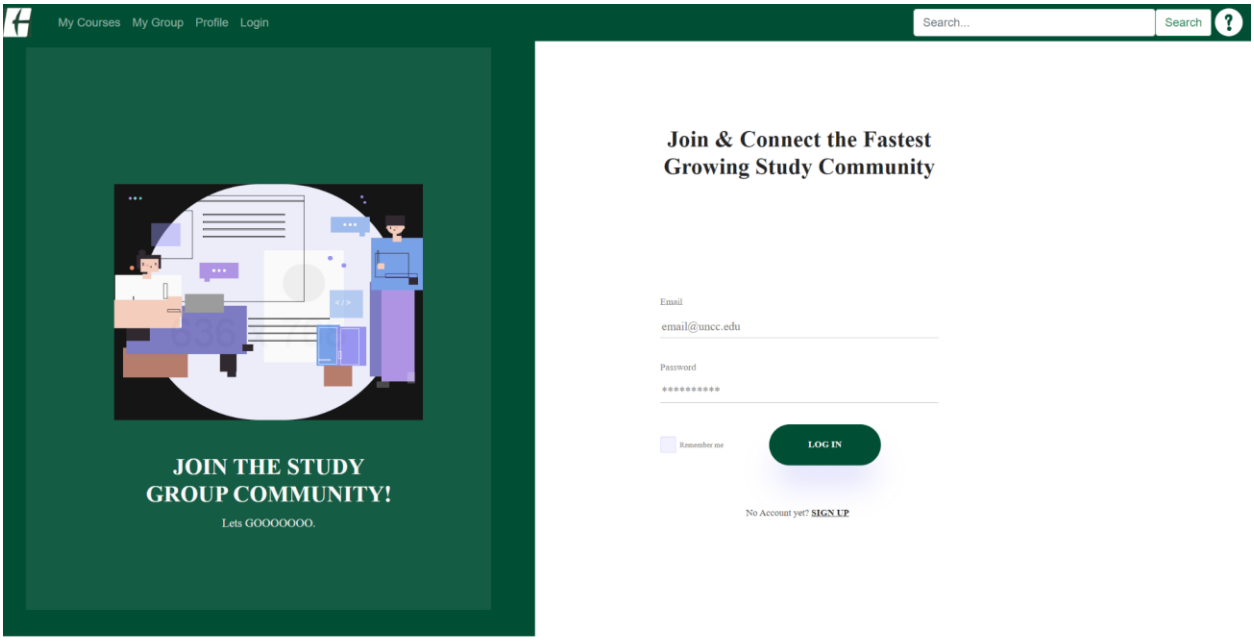
1-2. User Stories per Sprints

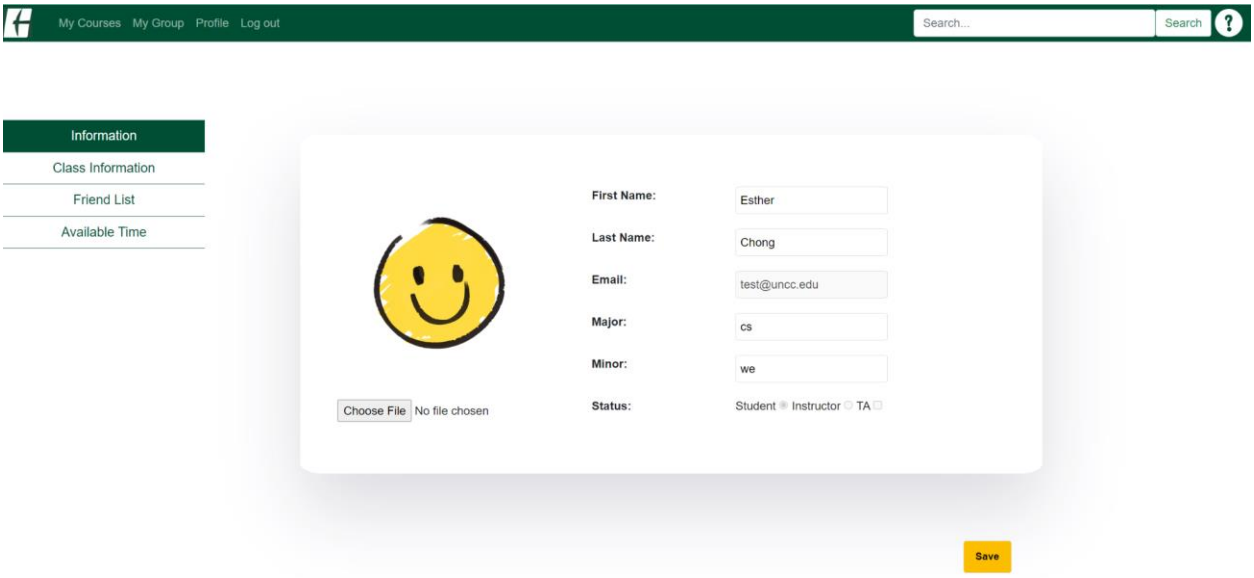
1-2-1. Sprint 2

ST-1	As a UNCC Student I need to see what classes my friends are in so that I can create groups of friends in the same classes and expand my own social circle.	Added a feature that allows users to see what classes other students are enrolled in and to be friends with each other
ST-5	As a student, I need to see the information available for my future class (but I don't need to participate in it) so that I can assess the difficulty level of the upcoming class.	All members are granted read access to all sessions. Write permission for sessions related to currently enrolled classes.
ST-13	As a User I need to be able to look up and add classes so that I can see what is going on in the class I am registered in	Implement a table in the database to List and lookup Classes
ST-14	As a UNCC student, I need to be able to create an account so that I can save my information and preferences.	Implement profile page and save classes that a user is enrolled in
ST-16	As an Instructor, I need to hide the course materials to those who are not in UNCC so that my course details and resources are protected.	When creating an account, make users use only the UNCC email and check if it is a valid email.

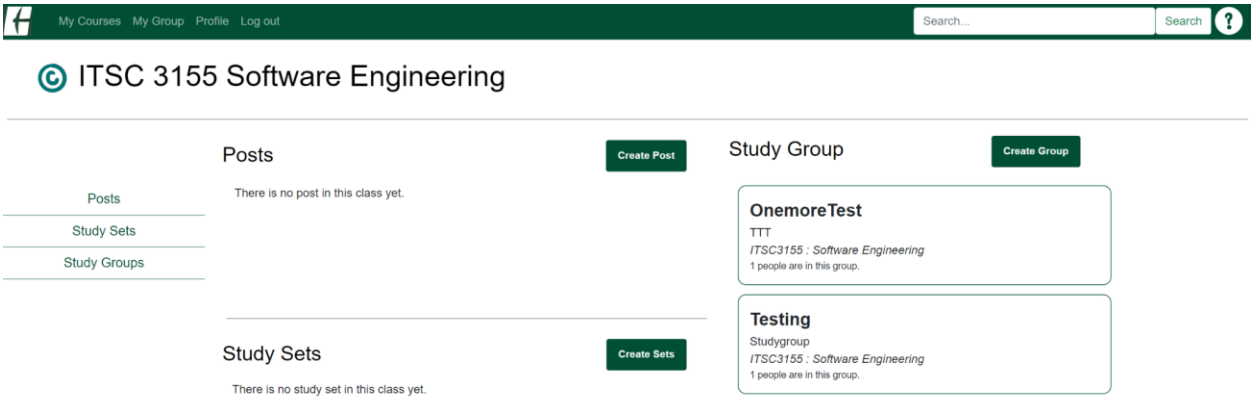
Major User Stories

ST-14 - Profile Page




My Courses
My Group
Profile
Log out
Search...
Search
Information
Class Information
Friend List
Available Time
First Name:
Esther
Last Name:
Chong
Email:
test@uncc.edu
Major:
cs
Minor:
we
Status:
Student
Instructor
TA
Choose File
No file chosen
Save

ST-5

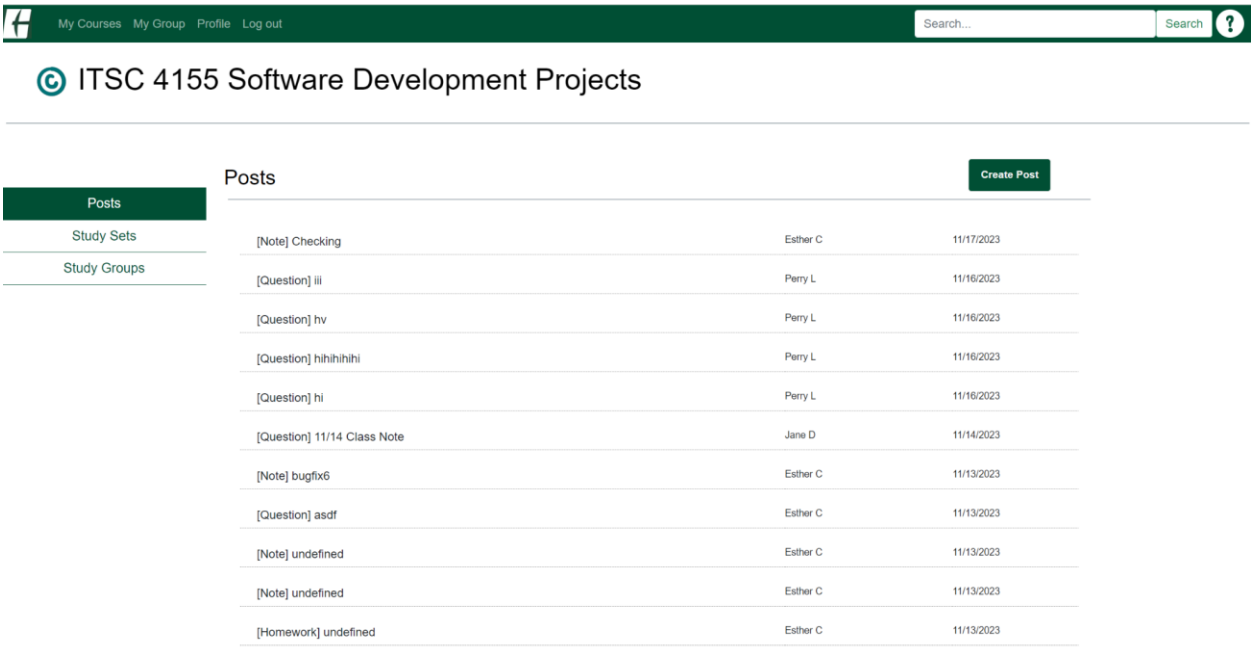

My Courses
My Group
Profile
Log out
Search...
Search
ITSC 3155 Software Engineering
Posts
Study Sets
Study Groups
Posts
Create Post
There is no post in this class yet.
Study Sets
Create Sets
There is no study set in this class yet.
Study Group
Create Group
OnemoreTest
TTT
ITSC3155 : Software Engineering
1 people are in this group.
Testing
Studygroup
ITSC3155 : Software Engineering
1 people are in this group.

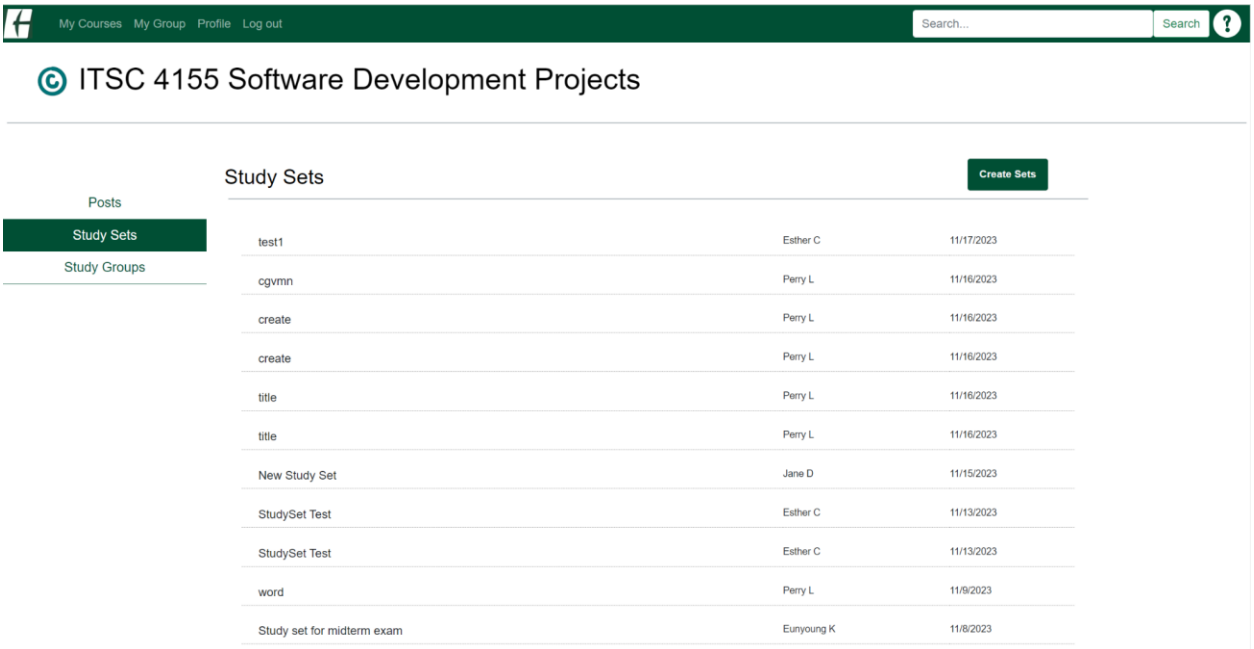
1-2-2. Sprint 3

ST-2	As a student site member, I need to search a group to study with, that fits my study times and needs, so that I can study in a group setting.	Student should be allowed to search for group
ST-3	As a student, I need to create a new study session for my upcoming class so that I can collaborate with my peers in understanding the course material.	Users shall be able to create groups that focus on what they wish to study
ST-4	As a UNCC student I need to join a study session with other classmates in my class so that I will have better understanding of the concepts that were taught in class.	Allow students to join study groups
ST-7	As a student, I need to review study sets within the study session so that I can ensure I fully grasp the concepts and am well-prepared for the upcoming test.	Implement study sets function
ST-11	As a student, I need to make an appointment for group study so that I can meet my classmates to study together.	Implement location/time booking system
ST-17	As a student, I need a feature where I can share my notes with people so that I can see what we did in class and review it.	Implement a notes/hw sharing feature

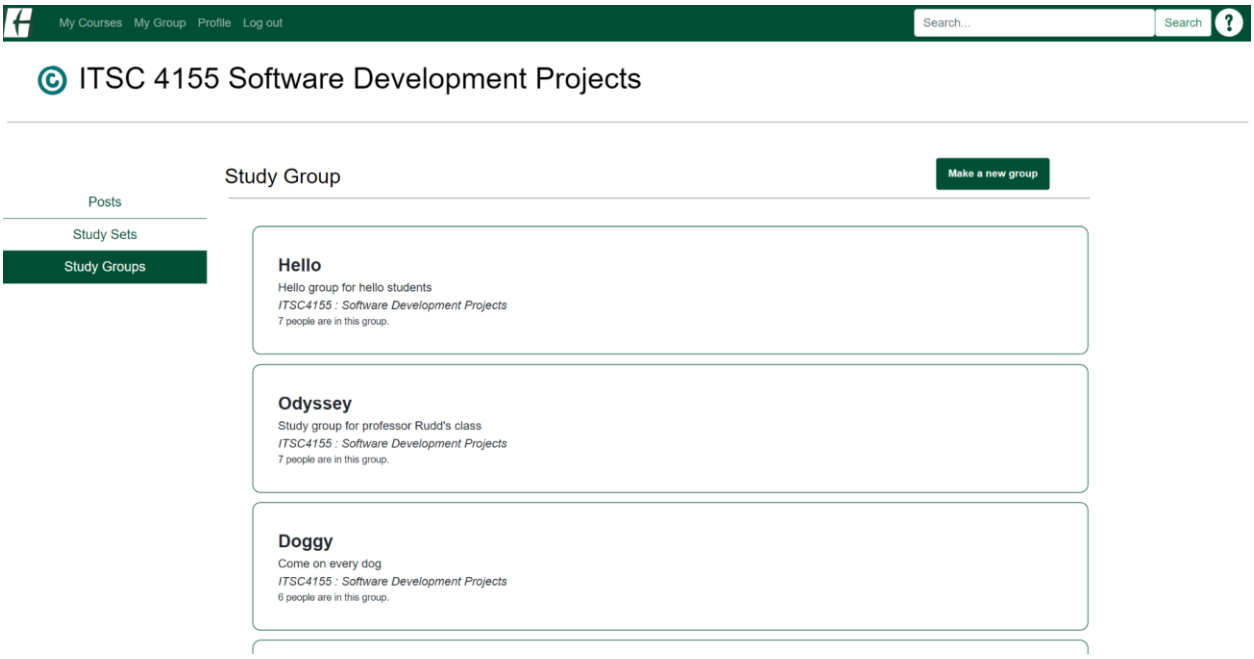
Major User Stories

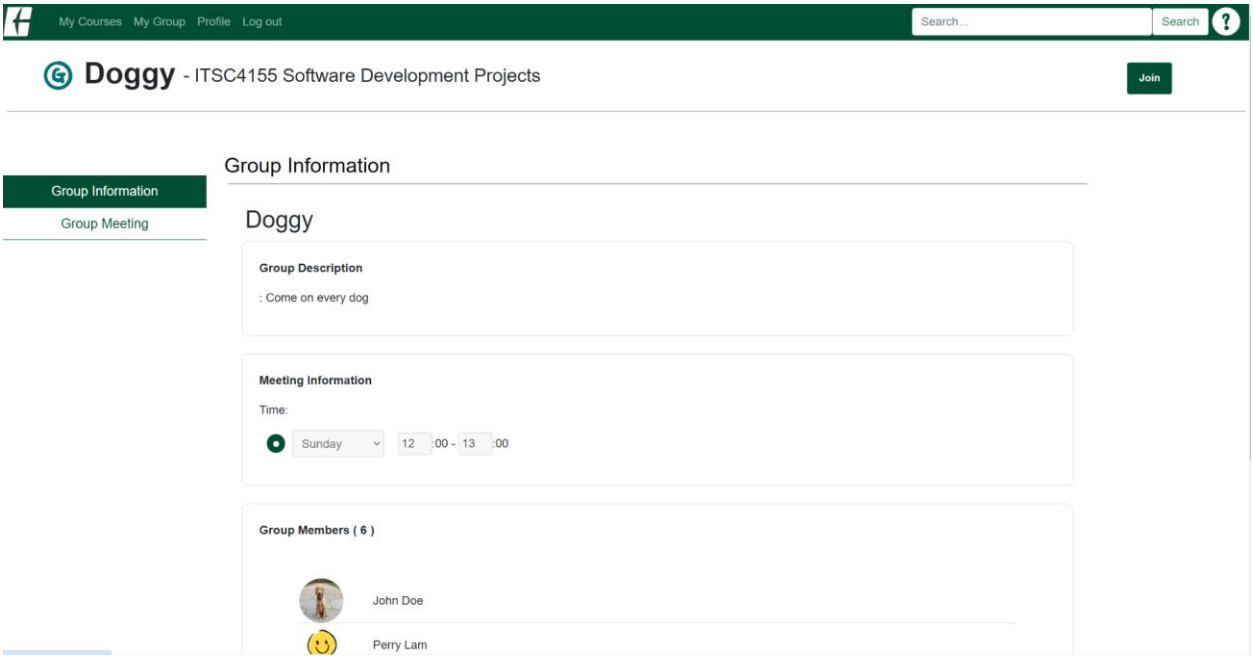
ST-7, ST-17 - Post/Note/Study Sets





ST-4 - Create/Join Group



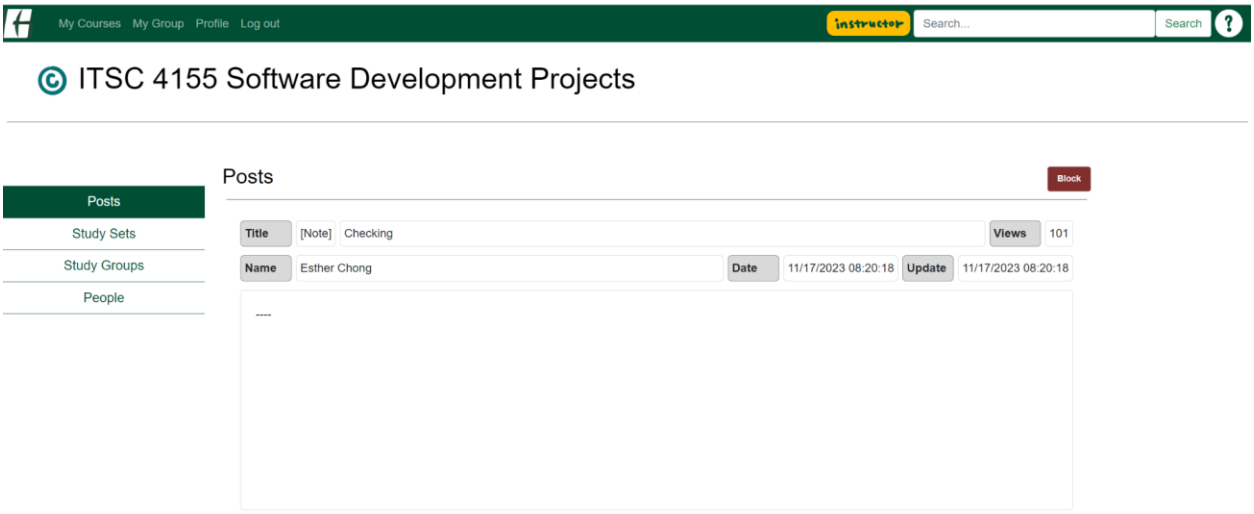


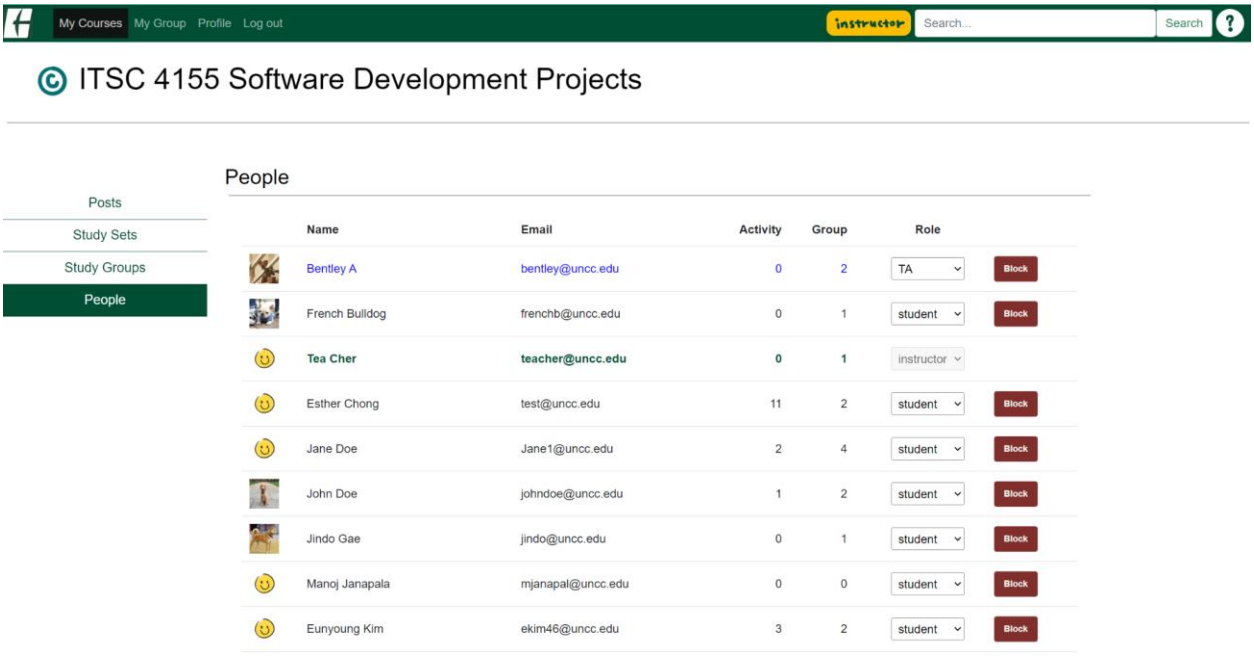
1-2-3. Sprint 4

ST-6	As an instructor, I need to view students' reactions regarding my class so that I can identify areas where students may be struggling.	Allow read and write to class sessions for instructor members
ST-9	As a UNCC Teachers Assistant, I need to be able to help students on topics they are struggling with so that they will do well in the class.	TA should be able to join any group under the same course they are working with
ST-10	As a Moderator and a UNCC student, I need to be able to remove any inappropriate comments so that there is no inappropriate language or image in the course page.	Moderators should be able to remove posts/study sets/groups from the list.
ST-12	As a Moderator and a UNCC student, I need to be able to block people displaying inappropriate behavior, so that there is no inappropriate behavior going on in our website.	Moderators should be able to block anyone.
ST-15	As a UNCC student with limited experience with technology, I need a tutorial to help with my account setup so that non-technology savvy students like me can also use the website with ease.	Implement onboarding help screen to guide new users of website features.

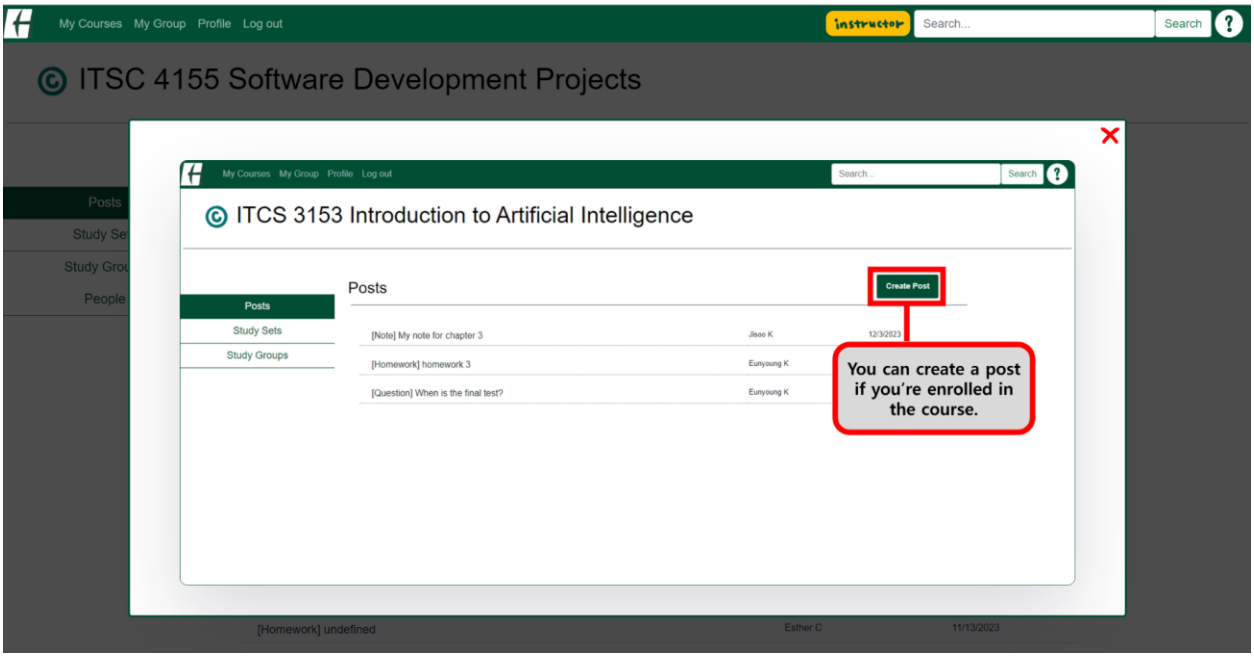
Major User Stories

ST-6, ST-10, ST-12 - Instructor Member's Permission/Blocking





ST-15 - Tutorial Page



2. Architectural Overview

ClassPlus adopts a three-tier architecture model, encompassing distinct layers: the presentation layer, business layer, and database layer.

Presentation Layer: Users engage with ClassPlus through a web-based interface (HTML, CSS, JS), facilitating access to functionalities like course management, group interactions, and profile settings.

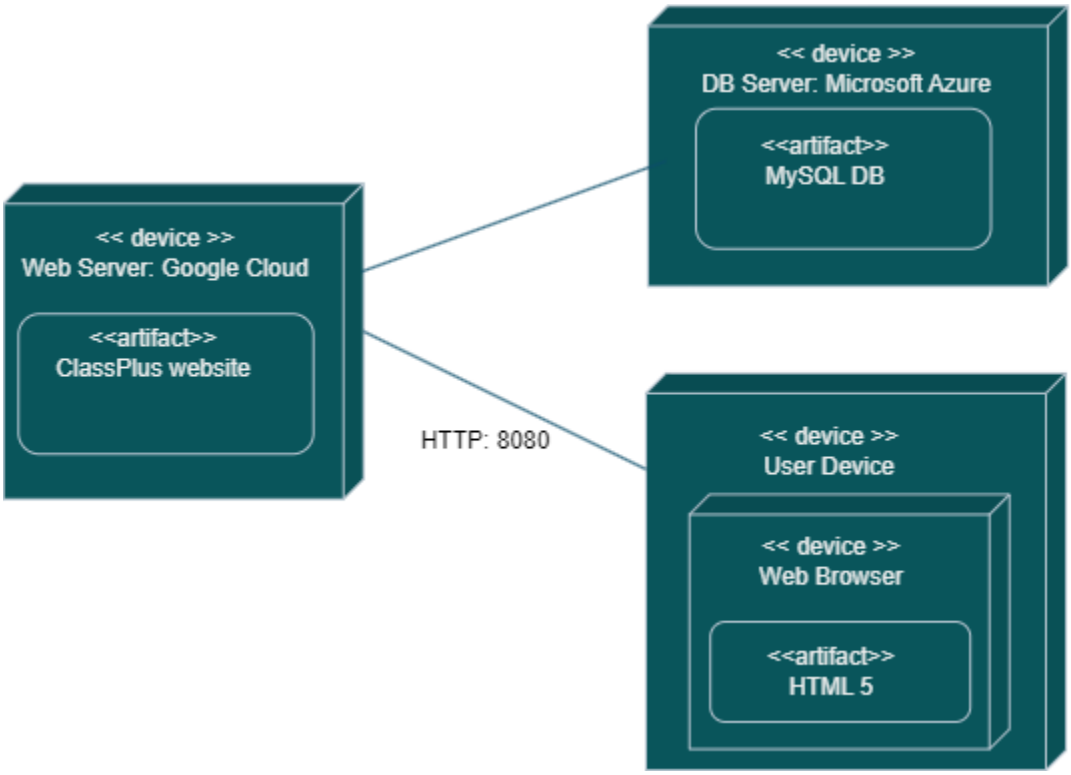
Business Layer: The frontend component manages user interface logic, ensuring an engaging and responsive user experience. Meanwhile, the backend, powered by Node.js, handles server-side operations, encompassing authentication, data processing, and seamless interaction with the database while adhering to defined business rules.

Database Layer: Critical application data, including user profiles, course details, group information, and activity logs, is stored and managed in a MySQL database hosted on Microsoft Azure.

When deliberating over alternative architectures, an event-driven architecture was considered due to its distributed asynchronous nature, suitable for systems with asynchronous API endpoints like ClassPlus. However, after thorough consideration and evaluating various factors such as process availability, responsiveness, and failover mechanisms in the event of mediator or broker failures, the decision was made to adopt the layered architecture. This decision was primarily driven by the team's familiarity with layered architectures, ensuring a more confident and efficient implementation of ClassPlus's functionalities.

2-1. Deployment Architecture

2-1-1. Deployment Diagram



User Device

Users access the ClassPlus application from their personal computers, laptops, or other devices equipped with web browsers. The users interact with the application's frontend through their devices' browsers, allowing them to access various features, such as course enrollment, group interactions, profile management, and more.

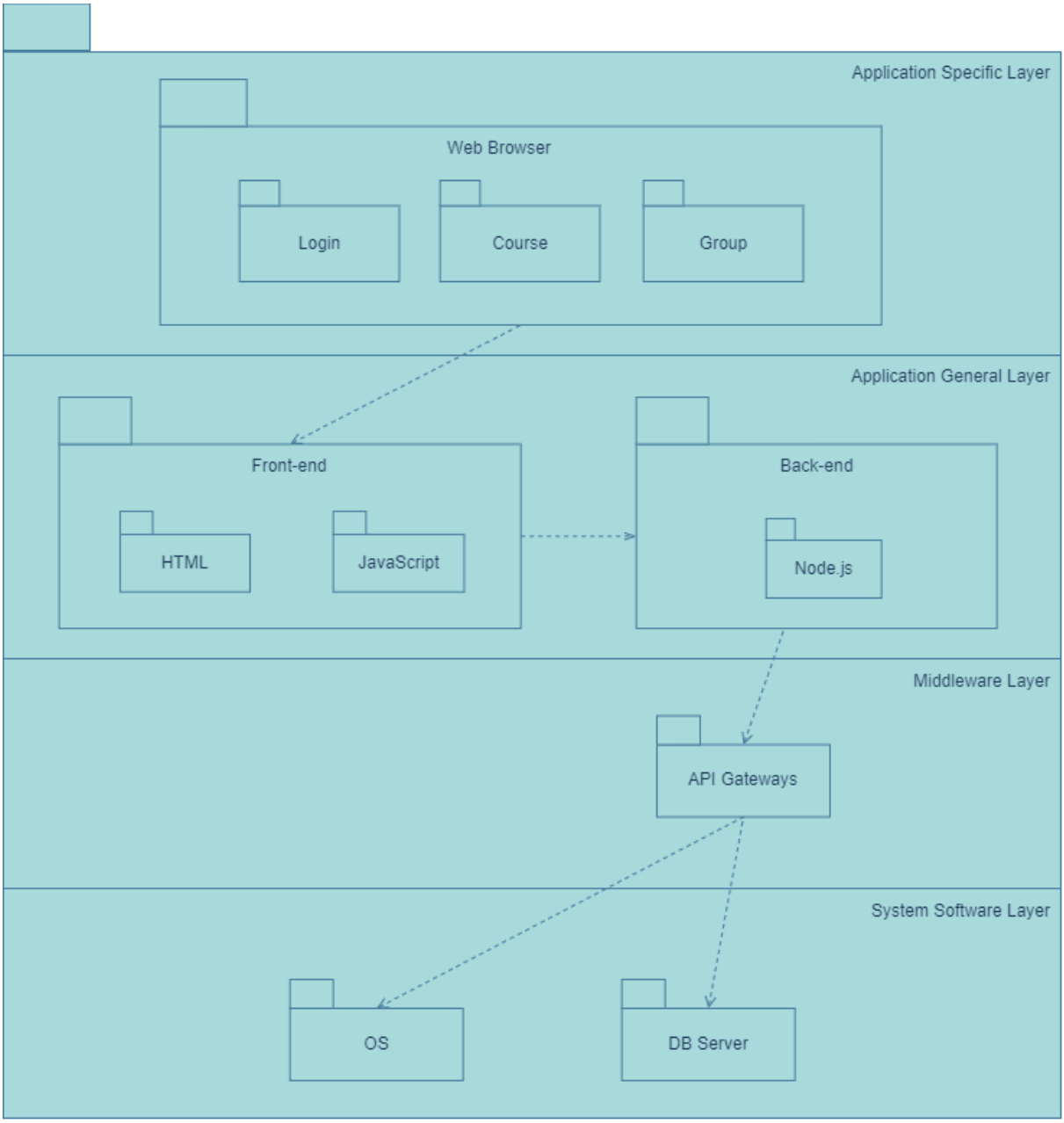
Web Server: Google Cloud

The ClassPlus website is hosted and managed on the Google Cloud platform. This web server acts as the central hub for serving the application to users' devices. It hosts the frontend components of ClassPlus, including HTML, CSS, and JavaScript files, ensuring seamless interaction and user experience.

Database Server: Microsoft Azure

The database for ClassPlus is hosted on the Microsoft Azure platform. This server stores and manages all the application's data, including user information, course details, group memberships, and more. It houses the MySQL database utilized by the application, providing robust data storage and retrieval capabilities.

2-1-2. Subsystem Architecture Dependency Diagram



The application-specific layer comprises the web browser, serving as the user interface for login, course management, and group interactions. This interface acts as the bridge to the application-general layer, where the frontend components (HTML, CSS, JS) manage the user interface structure and interactions, while the backend, powered by Node.js, handles server-side logic and facilitates seamless communication between the frontend and the database.

Moving into the middleware layer, API gateways serve as intermediaries, facilitating communication between the backend and the database, streamlining data flow and ensuring efficient interaction.

Finally, in the system software layer, we find the database hosted on Microsoft Azure using MySQL, which manages and stores the application's data. Additionally, this layer includes the operating systems running on the servers hosting the ClassPlus application, providing the foundational environment for its functionality.

2-2. Persistent Data Storage

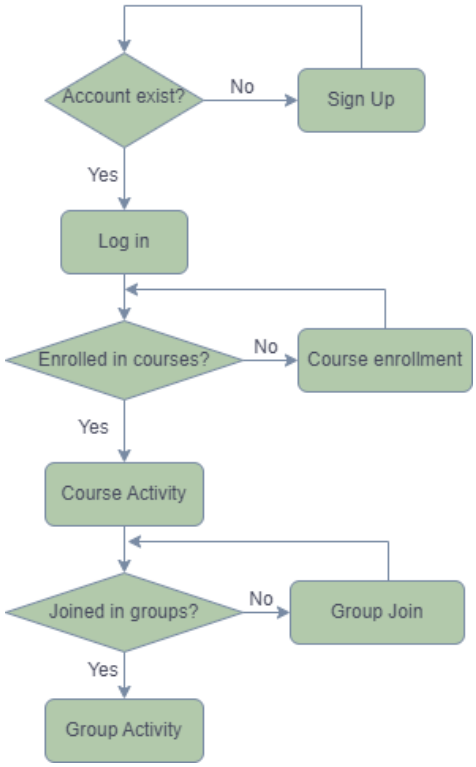
The MySQL database utilized for this system hosted on Microsoft Azure comprises ten key tables, each serving a specific purpose in storing and organizing data essential to the functionality of the system:

- **Users** Table: Stores user information registered within the system, accessed across various system functionalities such as authentication, profile management, and saving class-related data.
- **Friends** Table: Holds data regarding users' connected friends by storing the IDs of users' following friends.
- **Available Time** Table: Manages and stores users' available free time during a specific semester, visible within group pages the user has joined.
- **Courses** Table: Contains comprehensive information about each course, including subject, course number, and title.

- **Courses Enrollment** Table: Records details of users' enrolled courses, including the semester, year, and their role within the course (student, instructor, TA).
- **Courses Activity** Table: Captures user activity related to a course, encompassing posts and study sets' activity IDs, titles, content, course information, views, writer ID, post date, and blocking status.
- **Study Sets** Table: Stores information about study sets, such as terms and definitions, linked with the courses' activity table using activity IDs.
- **Study Groups** Table: Manages data pertinent to study groups, including group name, description, course information, year, semester, and blocking status.
- **Study Groups Member** Table: Records membership information for each study group, linking group IDs with user IDs.
- **Group Available** Table: Tracks meeting details for each group, encompassing time, location, and whether suggested time and location are selected for the meeting.

2-3. Global Control Flow

ClassPlus is a procedure-driven system since users should go through each step to use any further features. Users access the website, starting with login or registration. After logging in, users navigate through the website using links to different sections, such as courses, groups, or profile pages. Here, users can go to profile pages to change their information or add available time (optional), or they can enroll in courses for course activities such as creating posts and study sets. Next, once they are enrolled in at least one course, they can create or join any group for group activities such as setting group meetings. Each action follows a set procedure consistent for all users.



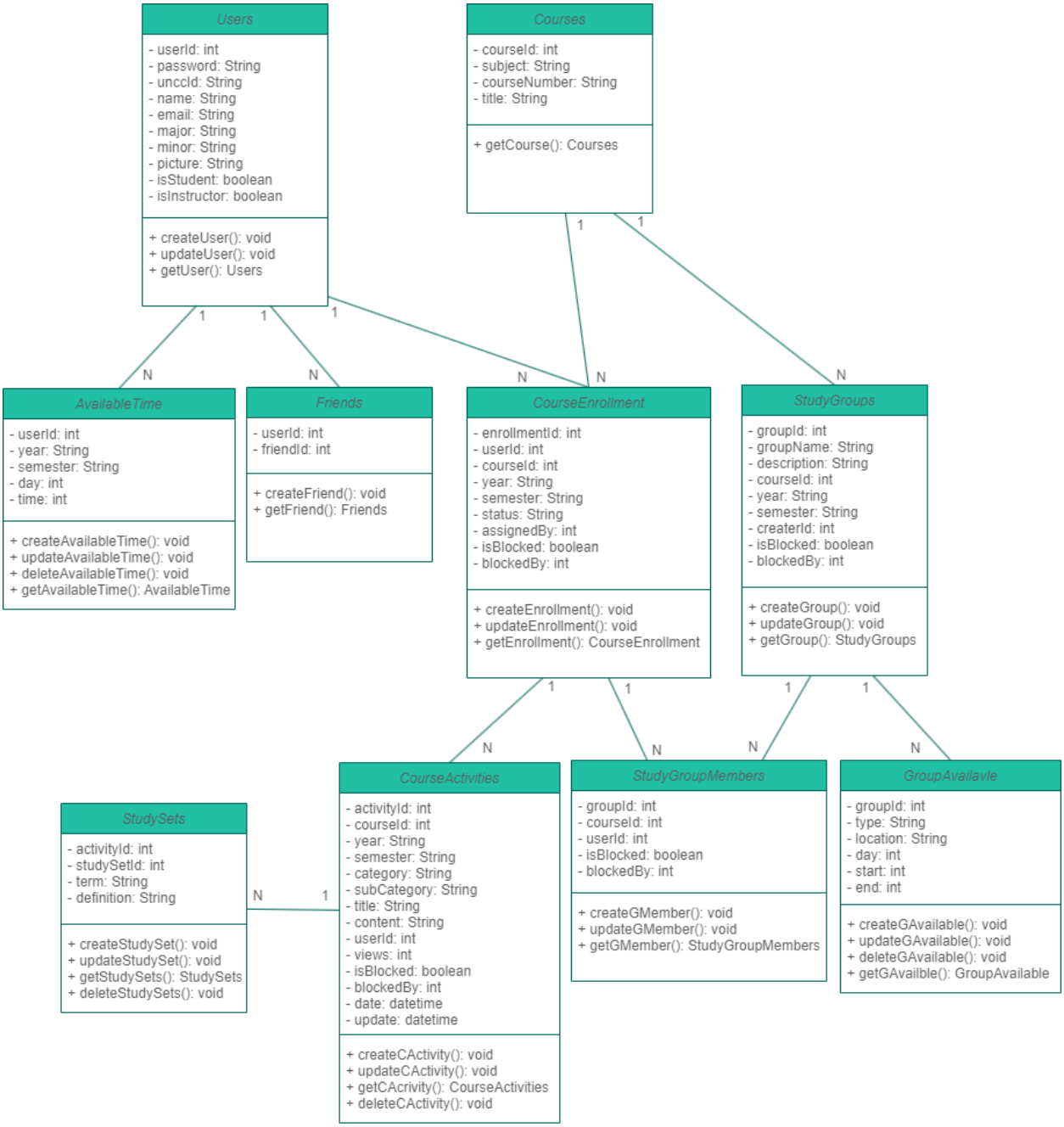
ClassPlus also has event-driven functionalities such as buttons for posting, editing, blocking, and following friends. When users click those buttons, operations corresponding to the button occur.

ClassPlus implements Async Tasks to handle multiple user interactions concurrently, preventing discrepancies. This setup ensures that incoming requests are managed individually, minimizing race conditions or crashes during interactions. Concurrency management allows multiple users to interact with the application simultaneously without impacting the system's stability or causing conflicts.

3. Detailed System Design

3-1. Static View

ClassPlus has 10 main classes alongside the database design.

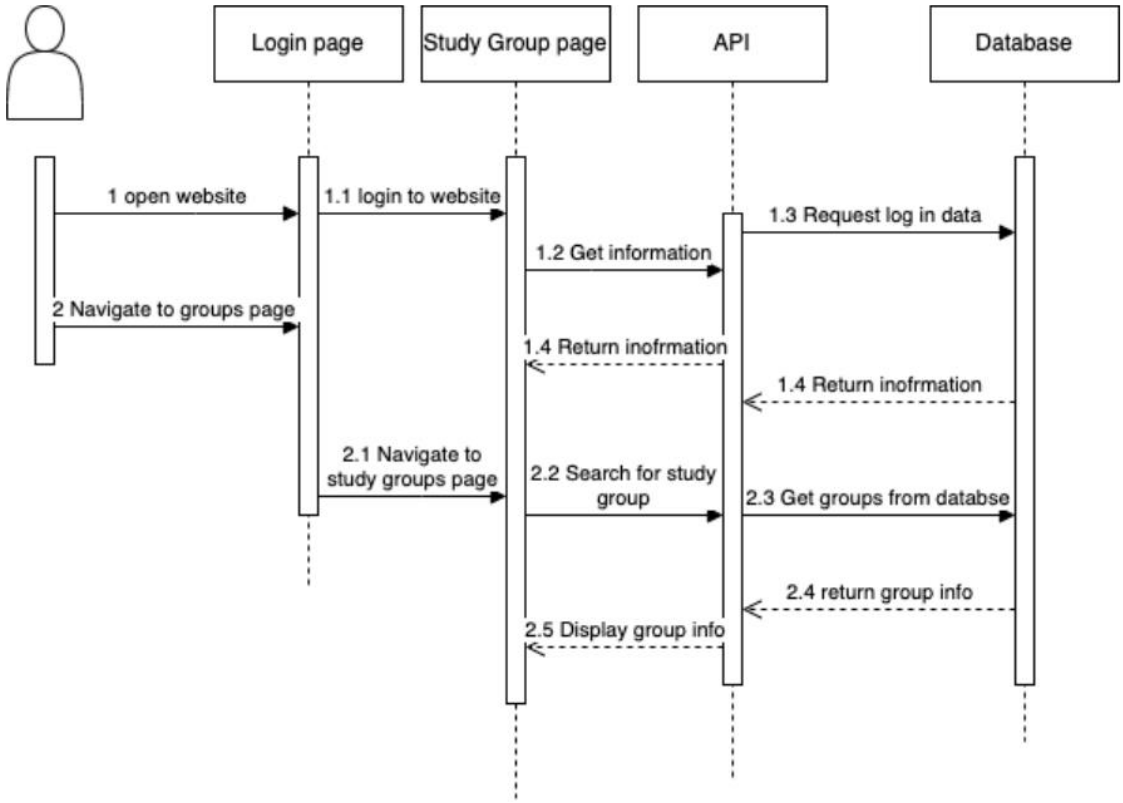


Key Classes and Relationships

- **Users** Class
    - Multiple instances of **AvailableTime**, capturing their free time during specific semesters. (one-to-many)
    - Numerous friends through a one-to-many relationship with the **Friends** class. (one-to-many)
    - Multiple enrollments in courses via **CourseEnrollment**, allowing users to be part of various courses. (one-to-many)
- **Courses** Class
    - Multiple **CourseEnrollment** instances, handling enrollments from various users in a course. (one-to-many)
    - Multiple **StudyGroups**, facilitating the existence of multiple study groups within a course. (one-to-many)
- **CourseEnrollment** Class
    - Multiple **CourseActivity** instances, managing various activities (posts, study sets) related to a specific enrollment. (one-to-many)
    - Multiple **StudyGroupMember** instances, enabling enrollment in multiple study groups. (one-to-many)
- **StudyGroups** Class
    - Numerous **StudyGroupMembers**, representing multiple users enrolled in a particular study group. (one-to-many)
    - Multiple instances of **GroupAvailable**, indicating various available times for group meetings. (one-to-many)
- **CourseActivity** Class
    - Multiple **StudySets**, signifying multiple terms and definitions within a course activity, particularly in study sets. (one-to-many)

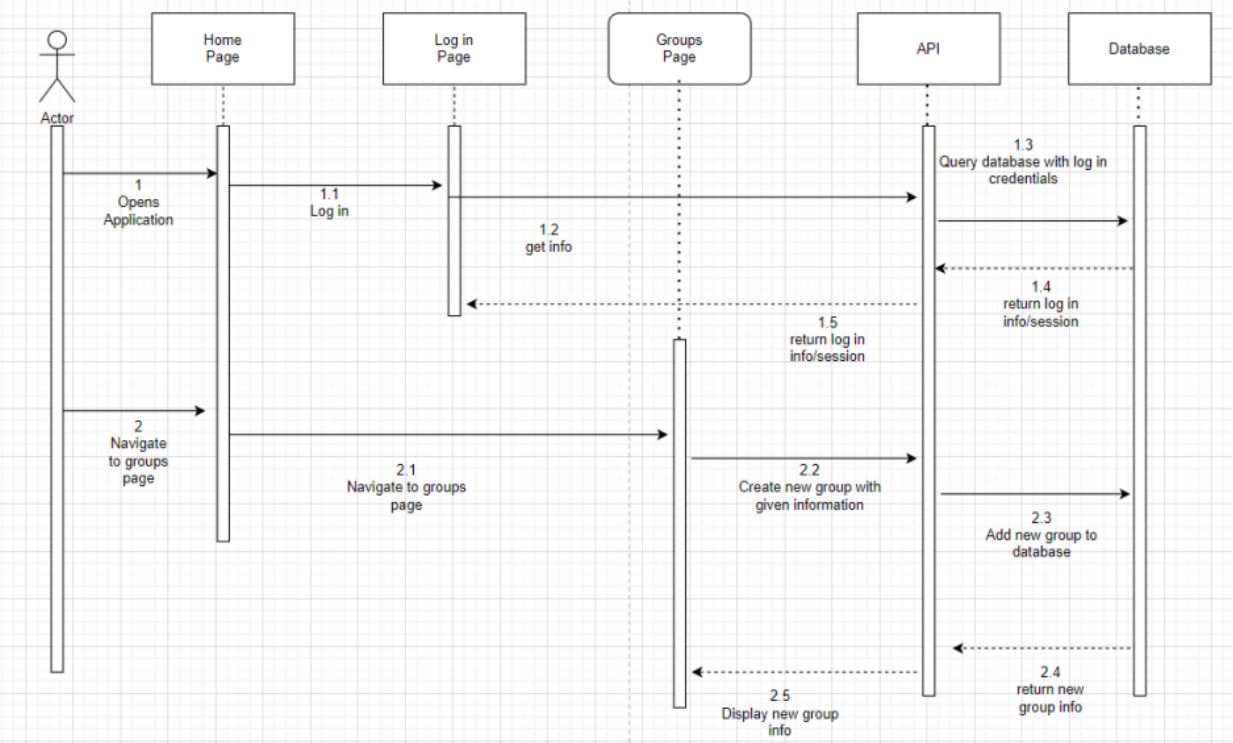
3-2. Dynamic View

3-2-1. Searching Group



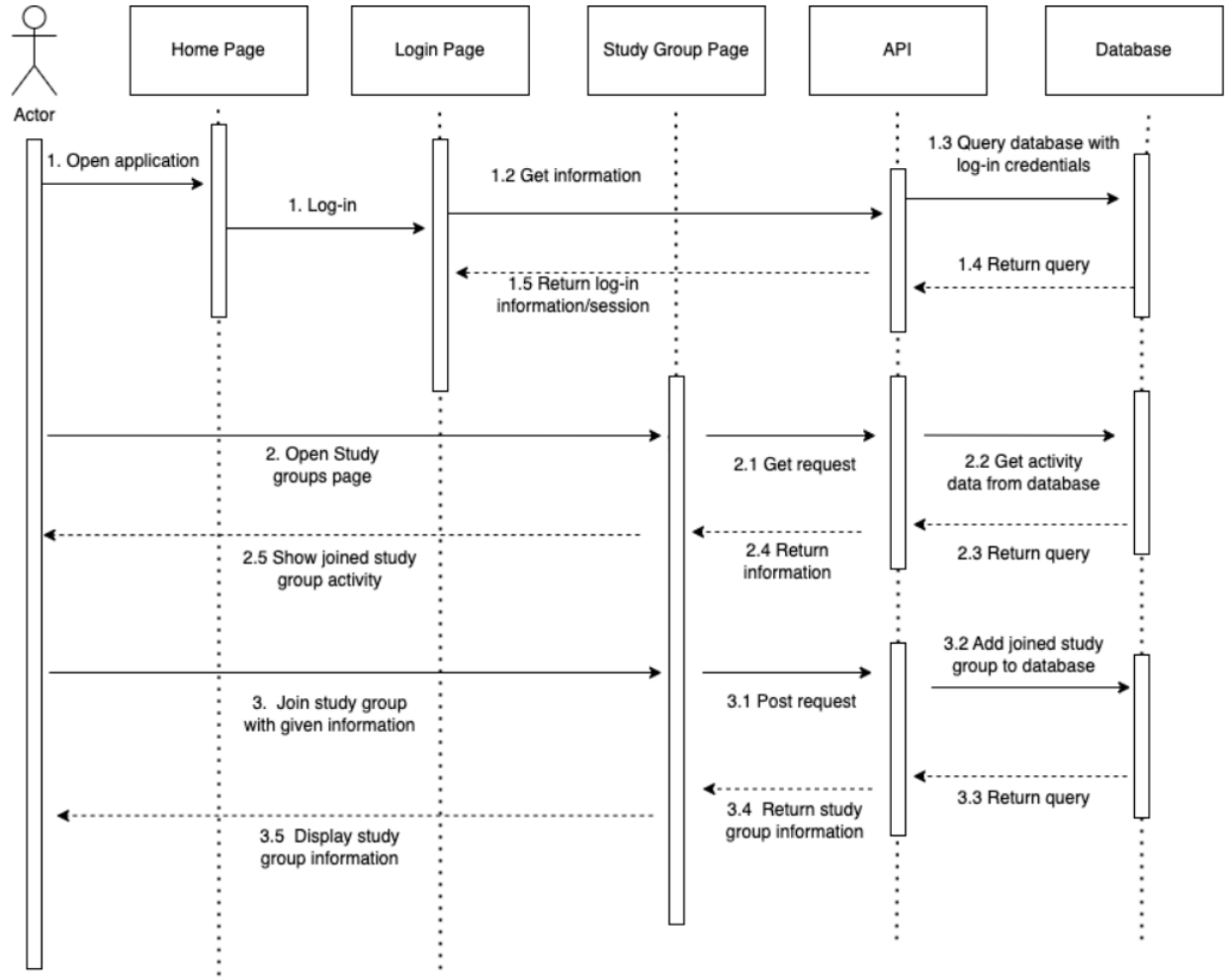
The user logs into the system, and the API requests data from the database. The database returns the information to the login page. Subsequently, the user, looking to join a study group, utilizes the search bar to find groups based on the information of existing ones. The API is then called to retrieve the group data from the database. The database responds by sending the group information back to the study groups page, enabling the user to find and join the group.

3-2-2. Create Group



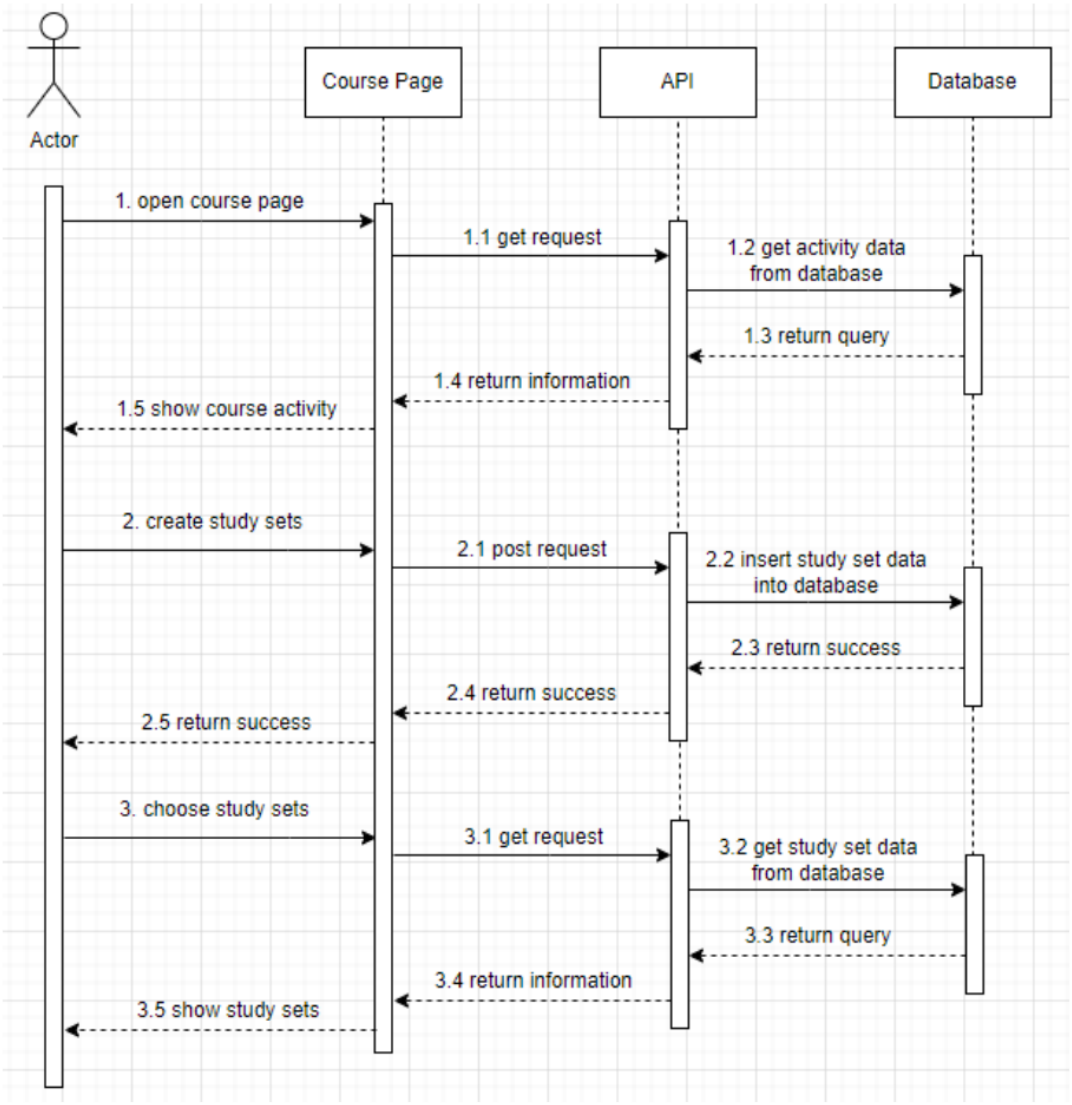
At the outset, the user opens the application and lands on the homepage, where they are prompted to log in via the login page. After entering their credentials, the API initiates a call to the database, which responds by either redirecting the user to the homepage, indicating a successful login, or notifying them of a login failure with a prompt to try again. Once successfully logged in, the user can navigate to the groups page. Here, they have the option to create a new group by clicking on the 'create group' button and providing necessary details such as the class, members, meetup times, etc. The API processes this information, communicates with the database to add the new group details, and subsequently returns the information. The API then redirects the user to the new group page.

3-2-3. Join Group



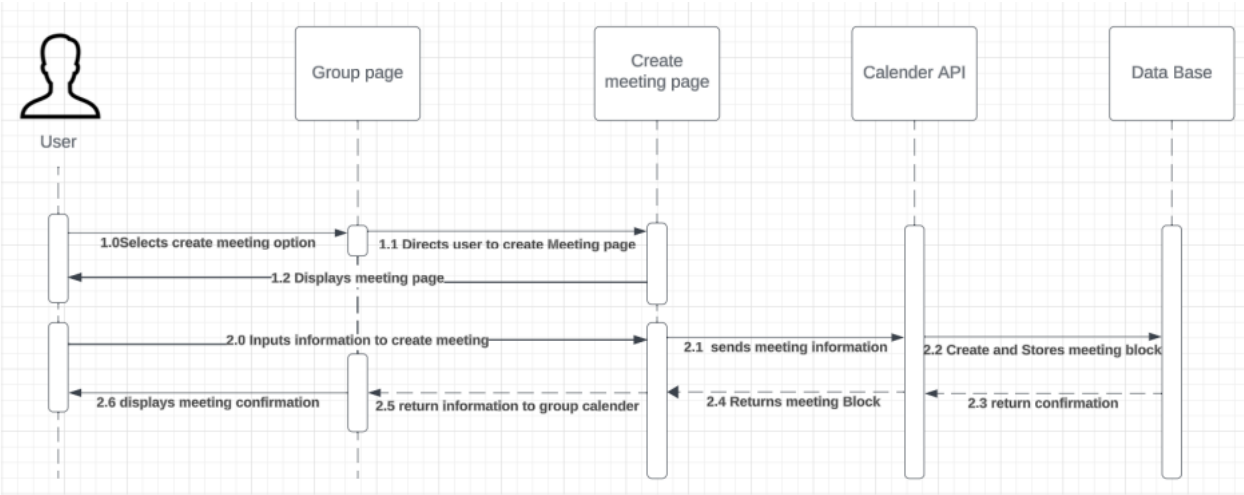
After entering user credentials, a request is sent to the API to validate the information against the database. Upon successful validation, the user gains access to the study groups page, where they can view their joined study groups through an API request. Additionally, users have the option to join new study groups, providing information for a POST request, which is then added to the database. After the database is updated, the user can view the information on their joined study groups.

3-2-4. Posting Post/Note/Study Set



When a user visits the course page, the API initiates a GET request to the database to fetch activity data, and the database responds by transmitting the information through the API. Subsequently, if the user creates a note, homework, or post on the course page, the API receives the data and seamlessly inserts it into the database. The database, in turn, sends a message—either confirming success or indicating failure—back to the user through the API. Furthermore, when the user opts to view any existing note, homework, or post, the API processes the request, retrieves the pertinent information from the database, and promptly delivers it to the user.

3-2-5. Create Group Meeting Time/Location



The user will access the page to create a meeting from the group page, when they click the button labeled Create meeting. In step 2, on the create meeting page the user will input the meeting information which will be relayed to the calendar API, which will create a meeting block and store it in our database. After the meeting is created, the calendar API and Database will send a confirmation message to the user on the Create Meeting page. Finally, the meeting will be marked on the group calendar on the group page.

4. Appendices

4-1. Unimplemented User Stories

ST-8	As a new student, I need to be able to chat with students in similar classes or majors as me so that I can make friends with similar interests.	Chat feature implemented

**Before Sprint 3 began, we informed the professor that we wouldn't be able to complete this task due to time constraints*

4-2. Mini User Manual

Sprint 2 | Sprint 3 | Sprint 4