

UniCollab: Deliverable 2

Part 1: Software Architecture

Attached as separate pdf

Question 1 - External Data Sources

What external data sources will your system be accessing?

The overarching goal of UniCollab is to centralise and simplify the process of group-making for university students. To do this requires access to information provided by users, the ability to access calendars to seek out potential meeting times and update one's own personal calendar. To do this, UniCollab relies on Application Programming Interfaces (APIs) to communicate with external sources. The critical API's used by UniCollab regard calendar information and communication between clients and servers; Google Calendar API and REST API.

Representational State Transfer (REST) API

- Makes a call from a client to a server to retrieve data via HTTP protocol
- Web apps can communicate with cloud-based servers via the REST API
- Industry known and used
- Communication between client and server, and restful web service
- Benefits:
 - Simple, standardised approach to communication
 - Scalable and stateless - as it grows in complexity, can make modifications
 - High performance - support caching
- Desirable actions to use with REST API:
 - Create, Read, Update, Delete
 - HTTP methods/operations for the above: POST, GET, PUT, DELETE
- Responses typically in the form of JSON data

Google Calendar

- Integratable with mobile and web applications
- Useful for UniCollab to organise and display current and proposed meeting times and sync personal schedules

Google Maps

- Easily integratable into web applications
- Provides basic features that UniCollab requires for group-matching recommendations
- Groups and users have locations that are used in algorithms

- Will use to find distances between users' preferred meeting locations and sort and colour code search results for groups/users by distance

UNSW WebCMS

- The search engine may be used to find groups or people that have experience with particular skills, e.g. Python. Our solution will locate 'Python' on the webCMS outlines, and display a list of users or groups that have completed courses pertaining to 'Python'.
- By scraping information underneath the 'Course Summary' and 'Assumed Knowledge' subheadings and looking for keywords, like 'Python', from the following directory:
 - <https://webcms3.cse.unsw.edu.au/{CourseID}/{TeachingPeriod}/outline>
 - CourseID = User's course name, e.g. COMP1000 (Capitalized)
 - TeachingPeriod = Year + Term, e.g. 20T3 (Capitalized)
 - Example: <https://webcms3.cse.unsw.edu.au/COMP1511/20T3/outline>

Question 2 - Software components

Software components: the selected Web stack showing major software components that comprise your solution. These will include both components that need to be developed and third-party components (e.g. web browser).

DATABASE: MySQL

Data Manager + API manager: Django

Frontend API handler: Axios

Application classes: Python

FRONTEND FRAMEWORK: Vue

Scripting language: JavaScript

UI Library: Vuetify

EXTERNAL CALENDAR API: Google Calendar API

EXTERNAL MAPS API: Google Maps API

EXTERNAL SKILL MATCHING RESOURCE: WebCMS (Course Outlines)

BROWSER: Chrome, Safari, Firefox, etc. (any apart from Internet Explorer 10 and lower)

Question 3 - Which language for which component

Relating choices to components: decide which language should be used for which component of the software architecture. This will be largely determined by the Web stack but at the same time you can make variations.

Database: SQLite | PostgreSQL

MySQL:

- Relational Database Management System (RDBMS) that makes database administration easier and more flexible

- Provides extensive documentation and an abundance of online resources
- Adherence to standard SQL to attain speed and reliability
- Multitude of third-party tools, libraries and applications
- Comes with a script to ensure security of database
- Supports granting privileges on a user-to-user basis
- Very useful for websites and web applications
 - Powers many powerful applications and websites, including FaceBook, Twitter and YouTube
- Disadvantage: not completely SQL compliant - lacks some features
 - Leads to certain functional limitations

SQLite:

- A Relational Database Management System (RDBMS) known for its portability, strong performance and reliability
- Very Fast
- ‘Serverless’ database - does not communicate with a host server through requests
- Requires access to a database disk file directly
- Major drawback is that network access is required
 - SQLite is a serverless database, thus if the data is located on a separate machine, it is expensive and inefficient to use
- For the sake of our project (which is to pitch our idea), the lack of cloud integration is not an issue.

PostgreSQL:

- Object-relational Database Management System (ORDBMS)
 - Primarily a RDBMS, but includes features like table inheritance and function overloading
- Efficient concurrency - can handle many tasks simultaneously
- Ensures atomicity, consistency, isolation and durability (ACID) of transactions
- Closely complies with standardised SQL language
- Maintains data integrity
- Compatible with many other programming languages and applications
 - Useful when migrating databases with other systems

Rationale:

MySQL and PostgreSQL each offer a multitude of benefits over SQLite, however, we will be choosing to go with SQLite for the sake of this project. The main advantage the former two have over SQLite are their client-server capabilities which SQLite does not support (since it is serverless and requires a high bandwidth engine-to-disk link if the data is not located client-side).

However, for this project and given the time frame we have, implementing a cloud server which can be accessed by multiple users on different machines seems to be far too great of a task. The end goal of this project is to create a minimum viable product (MVP) to pitch our idea

to the SENG teachers and Macquarie Group. This demonstration can be done locally on the client side and thus, the extra effort spent implementing network functionality would be wasted.

In the long term future, if we were to push our application to the real world, we would be transitioning to MySQL as we believe it's popularity puts it above PostgreSQL whose main advantage is its capability to perform complex operations (which are beyond the scope of our database requirements).

Server: Django | Flask

Django:

- Web application framework for dynamic applications built with Python
- Comes with a multitude of features to aid with authentication
- Django documentation is incredibly accessible and useful
 - Examples and tutorials are bountiful
- Makes serialisation very easy
 - Serialisation: the process of querying and converting database values into JSON format
- Ability to view backend information and update it from Django administration page
- Can define models for databases using python
- Django Object Relational Mapping (ORM) writes much of the SQL required to create and update current database models
 - Built-in template engine

Flask:

- Microweb framework written in Python
- Designed to get started with web development quickly
- Third party libraries can be added
- Ability to scale up to complex applications

Flask vs Django?

- Flask does less for developers - Django provides more automated code
 - Useful for beginners
- Flask is useful for python beginners, complete freedom over code and know everything about what is occurring inside the code
- Django is useful for python-experienced developers, and for web applications that are more complex than basic operations
- Flask is useful with non-sequel databases or applications with no databases required
- Django however is preferred for applications that utilise databases such as PostgreSQL, MySQL, Oracle etc.
- Django uses conventional project layout structure whereas Flask's is user-defined

Rationale:

Django and Flask are respectable choices for our web application's backend, since they are both implementable with Python, which we have chosen as our programming language for the project. They are both compatible with SQLite (our chosen RDBMS), which made us look at finer details of each to make a decision.

Django is ultimately our final choice due to its vast multitude of documentation and online resources to aid our progress, as well as its ORM feature which writes much of the required SQL material on-the-fly. Flask also lacks a default template engine and some useful built in features of Django, such as the ability to view backend information from Django's administration page.

Considering that we are relatively new developers, and for most of us this is our first major project, Django appears to be the more appropriate choice, due to its ease-of-use and available assistance.

Frontend: Vue + Nuxt | React | Angular | Plain Javascript + HTML + CSS

Vue:

- Progressive JavaScript framework used to build user interfaces and single page applications
- Adjustable for both desktop and mobile app development
- Having core library is focused on the view-layer only
- The production-ready build project weighs just 20KB after min+gzip
- Functions with Virtual DOM and it makes use of most of the common web technologies to build apps as per the needs of the users
- Supports the component-based approach to building web apps – it includes single-file components that are independent and loosely-coupled to enable better code reusability and quicker development

AngularJS:

- An open source framework used to develop dynamic web applications
- Based on client-side technology and bring together the functionalities of HTML, JavaScript and CSS
- Turns HTML-based documents into dynamic content

ReactJS:

- JavaScript library which is open sourced in nature and is used to develop a user interface for Single Page Applications
- Composes complex UIs from small and isolated pieces of code called “components”
- Combines the speed of JavaScript and uses a new way of rendering web pages, making them highly dynamic and responsive to user input

Vue.js vs ReactJs:

- Support from third-party libraries
 - React has bigger community, and long-term supported by Facebook

- Vue has fewer resources, packages, and third-party libraries
- Documentation
 - Vue has better documentation than React. Vue has one of the best API references
 - React's documentation is relatively difficult to understand
- Template:
 - React uses JSX, where both HTML and CSS are expressed in JavaScript with XML syntax
 - Vue has a more traditional approach with Single File Components and distinct blocks for HTML templates, style, and JS. This separation is familiar to us, making the framework easy to learn
- State Management
 - State Management is essential to React
 - External library for state management
- Ecosystem
 - React relies on external solutions
 - Vue is standalone

Nuxt:

- Is an opinionated framework for creating Vue apps.
- Makes it easy to develop universal apps
- Focuses on the UI rendering aspect and offers modes for server-side rendering which means the browser renders the page as soon as it receives the response from the server, meanwhile the browser downloads JavaScript.
 - Web page should be viewable before rendering all functionalities
- Presents all structure required to make development easier

Vue vs Angular:

- Angular is more complex and opinionated
- Angular is fast
- Angular uses TypeScript (provides static type checking for scaled application), and Vue uses TypeScript.
- Angular is best when projects take advantage of its many functionalities and features
- Vue is better-suited for smaller development and easy to get started.

JavaScript + HTML + CSS:

- Very standard, default tools for creating front end applications with functionality
- Takes much longer to code compared to existing javascript frameworks such as react, vue, angular, etc
- Hard to pass data around as you have to write your own functions
- Should not be a consideration for projects involving a lot of data handling as existing frameworks are much more suited for data handling between components.

Rationale:

Vue is the clear choice here. We need to be able to pass some data between the different states, however, our application is not extremely heavy on these features. Therefore, using something more heavy-weight like React is counter intuitive as we would be spending more effort than required to achieve our goals.

Vue is more suitable for small development as it is lightweight and fast. This feature would improve the performance of our website by accelerating the loading speed with the SSR function offered by Nuxt.

Vue also has clear documentation and API references and has extremely easy integration with Vuetify (its UI component styling library) which further simplifies the development process.

Finally, our group has experience with Vue and the members who do not, have experience with React. The transition from React to Vue is much easier than the transition from Vue to React as React is a much heavier and complex framework. Therefore, we could speed up development times faster by reducing the learning curve required for our members.

UI component styling: Vuetify | Bootstrap-Vue | Element UI

Vuetify:

- A material design component framework for Vue.js that empowers developers to create applications quickly and efficiently
- All of the framework components are meticulously crafted to provide an easy to use interface while still maintaining the flexibility for seriously complex implementations

Bootstrap:

- Provides a very comprehensive implementation of Bootstrap V4 components and grid system
- It offers more than 30 plugins and 80 UI components and is accessible by default
- A desktop UI toolkit for Vue.js
 - its components are not numerous, but very well implemented and easy to use.

Element UI:

- A component toolkit available for Vue.js, React and Angular.
- Support the workflows of developer

Rationale:

ElementUI can support more frameworks than Vuetify, but as we have decided to use Vue, the Vuetify can provide more appealing components and better material design standards.

Although the combination of Bootstrap and Vue does offer excellent flexibility for development, vuetify is still preferable for this project due to its cleanliness, semantics and reusable components. As well as this, our group has members who are experienced with Vue and vuetify, which in a time-restricted project such as this one, is a preferable choice rather than learning a completely new system.

Backend: Python | Java | C

Python:

- High-level and easy to learn - easy syntax
- Offers compatibility with external backend applications such as Flask and Django
- Many external libraries with useful functionality
- Easy memory management
- Slower compared to other languages
- These are minor drawbacks and are ultimately outweighed by the advantages

Java:

- Concurrent, general-purpose and object-oriented programming language
- Popularly used for web and mobile applications
- Java programs run above Java Virtual Machine (JVM) - consumes more memory
- Design process is a lot more complex and slower due to having to consider the relationships between all objects, classes, interfaces, etc.

C

- Very fast
- Speed comes at cost of managing memory yourself which makes it hard to debug
- Long development times compared to other languages such as Python
- Not many good infrastructures and integrations available for frontend usage

Rationale:

We have decided to use Python since it is very easy to use and allows our entire group to work on the project as everyone has experience with Python. Furthermore, its ease of integration with Django (our server) gives it a huge advantage compared to something like C which is extremely hard to integrate. Speed is not really a concern, especially for the scope of our course as we are not extensively relying on time-intensive algorithms. The few time-intensive tasks we are relying on (such as dynamically updating the skills of a group or comparing the availability of a group to the availability of the user) are handled by our SQLite database which has extremely fast queries.

External Calendar API | Google Calendar | iCloud Calendar

Google Calendar:

- Very good API documentation
- Extremely easy to use
- Very popular world wide

iCloud Calendar:

- Decent API documentation
- Far less popular (almost everyone has a Google account but not everyone has an iCloud account)
- Easy to use

Rationale:

We will be going with the Google Calendar API due to its ubiquity and ease of integration. In the future, we would like to extend our project to cater to many more calendars, however, due to the time constraints and because we only need to pitch our idea by demonstrating the calendar functionality, it makes far more sense to only include one Calendar API and we have decided to use Google's.

External Maps API | Google Maps | PubNub

Google Maps:

- Reliable and experienced API - over 15 years of data collection
- 25 million daily updates
- Easily implementable alongside Google Calendar
- Extensive documentation and online resources

PubNub:

- Offers location tracking
- Focuses more on tools that assist to form routes and enhance geolocation
- Offers interface to view other user's locations
- These features venture beyond the scope of our goals
 - UniCollab wishes to protect the information (location) of users and does not require route-making

Rationale:

We chose to implement the Google Maps API primarily due to its ease of use and ability to do the basic features that UniCollab requires. Implementation alongside Google's Calendar API is widely advertised online in a multitude of online tutorials. Furthermore, alternatives to Google Maps offer more complicated features that we do not require. UniCollab just requires a user to input a general area (suburb) for which the group-matching algorithm can recommend near and far users.

UNSW CSE Course Information | WebCMS Course Outlines | UNSW Handbook

WebCMS Course Outlines:

- Contain more information than the UNSW Handbook "Overview" sections
- Contains information about all applicable comp courses
- Process of web scraping will match skills that groups are requesting

- Regex pattern matching on Course Outline's "Course Summary", "Assumed Knowledge" and "Student Learning Outcomes" fields
- Entering a course as completed will dynamically update a user's skills based on the above fields

UNSW Handbook API:

- Handbook outline information is up-to-date and public
- Easier to implement than the above option - API rather than collecting all course outlines and web scraping each one
- However, "Overview" sections lack much detail regarding the course
 - For example, the COMP1531 handbook overview lacks the term "Python"
 - Whereas, the course outlines name "Python" numerous times
 - Moreover, the provided webcms links are not always accurate or consistent

Rationale:

_____ An important feature of UniCollab is to recommend groups and users based on skill-matching; that is, recommend users whose skill sets complement groups nicely. Rather than asking users to input all possible skills, which would take some time and careful consideration, it is much simpler to enter a UNSW CSE course that they have completed.

Using the course outlines and web scraping to match regex patterns allows us to dynamically compile a list of skills for each user, simplifying the user experience significantly.

Question 4 - Choice of platform (Linux, Windows, etc)

The choice of a platform: decide on machine or machines requirements (Linux, Windows etc.) for the final system.

Our system has no major issues with compatibility. The primary concern we need to be aware of is that Vuetify is unsupported by Internet Explorer 10 and below. It would be preferable for the website to be run on Chrome, since due to its large user base, the majority of new updates to libraries such as Vuetify are first optimised for Chrome, and then other platforms. However, there should be no major issues.

As long as we have a machine which can run our tech stack which consists of very mainstream, generic languages (Python, Javascript, SQLite,...), it is fine. This encompasses almost all modern machines as every machine in 2021 can run these aforementioned tools.

In conclusion, our system does not use any niche tools to warrant a specific platform requirement.

Question 5 - Summary of benefits/achievements of architectural choices

Make a summary of the key benefits/achievements of your architectural choices.

Ultimately, our software architecture decision-making pivoted on the requirements and expectations of this SENG2021 project, notably regarding our group's fields of expertise and allotted time to create a minimum viable product to pitch. It is critical to note that many of our design decisions hinge upon our implementation timeline and final design vision.

This is indicated by our team's decision to implement a database for UniCollab via SQLite, rather than its server-client supportive alternatives. For the scope of this project, it is sufficient to create an appreciation that can be run locally, and thus committing to a more complex RDBMS is an inefficient choice due to our time restrictions. Also, most of our group has experience using SQLite, due to university courses and internships, which is especially beneficial when implementing a database in the space of a few weeks.

The decision to use Python as our programming language for the backend was made due to its strong compatibility with web application frameworks such as Django and Flask. Like SQLite, most of our team members have experience using Python, which would require less self-teaching than Java or C. A difficult choice was made when deciding which web framework application we would utilise; Django or Flask. Each option has vast advantages and ultimately would be sufficient for our project. Ultimately, Django was chosen due to Object Relational Mapping (ORM), which assists relatively new developers with automated SQL information. After researching both alternatives and following online tutorials, the group decided the Django was the more appropriate choice for UniCollab.

The frontend of the project and its corresponding rendering and component styling frameworks was primarily chosen to our team's experience, since all of the potential platforms worked with HTML/CSS/JS (which is our team's chosen frontend language). After considering Vue, Angular React and Plain HTML/CSS/JS, we decided upon using Vue. Not only did this decision factor upon our group's experience with frontend development, but also due to its suitability to smaller projects, such as ours. Using frontend alternatives such as React or Angular would require more self-teaching and require developing components that are out of the scope of this project. Additionally, Vue offers UI styling via vuetyf, which further cemented our decision.

The external resources and APIs are all very useful for the primary feature of UniCollab; to recommend users and groups to those based on three different criteria: location, meeting times and skill-matching. The Google Calendar and Maps APIs allow us to fulfil the first two requirements, whereas the process of regex pattern-matching against WebCMS Course Outlines allow users to dynamically update their skills behind the scenes by simply inputting their completed UNSW CSE courses.

Ultimately, after separating into smaller subgroups within the team, each learning about all of the possible choices to make up our application, our choices were finalised due to experience, online resources and simplification. This project must be completed within a relatively short period of time to present a minimum viable product, so relying upon languages and frameworks that members are already familiar with, which offer extensive resources and documentation was the most appropriate choice for us.

Part 2: Initial Software Design

Question 1 - Updated User Stories

Summary of changes since Deliverable 1:

Our user stories have been updated primarily in two ways; to reflect the changes to our application and to convey a more presentable and concise structure.

Changes to application features:

Based on feedback received by our mentoring sessions, we have decided to improve upon UniCollab's group-matching process by including more external resources to utilise additional complementing information. Originally, users would manually enter an abundance of skills that they possess into their profile page, which would be time-consuming and subjective. We have decided upon an alternative solution, one which improves the user experience significantly - users will input completed UNSW courses into their page, and then UniCollab will dynamically pattern match against these courses' Course Outlines on WebCMS, retrieving all possible skills.

Due to this, we have narrowed the general audience from 'all University students' to 'all UNSW CSE students', as we need to access as many Course Outlines as possible.

Additionally, we will utilise Google Maps and Google Calendar APIs to further improve the group-matching experience, recommending users to groups who fit a group's scheduled meeting times and locations. Users can simply sync their location and personal Google Calendar to their profile, and this information is used to recommend groups.

Based on these improvements, the group-matching algorithm has advanced to include 4 options: recommendation via location, meeting time compatibility, skill-matching, or a combination of all four. For each filter, there will be a backend algorithm to display colour-coded recommendations.

Changes to user story structure:

Our Deliverable 1 user stories were generally structured around specific features of UniCollab, where the requirements and subsequent features would often range from user profiles to dashboard pages, and hence was difficult to comprehend. We have updated these user stories to a version that is more appropriately structured and straight-forward; where all features that are similar are grouped together.

For example, all features and requirements regarding main dashboard components are all grouped together, and all user profile complements are grouped together under a different user story.

Additionally, the changes to our application as stated above have been reflected in our new User stories, especially those that are relatively new. Below is a general overview (table of contents) of our user stories, problem statements, requirements and features.

Overview

PS 1: No central hub (JOIN PROCESS + ROOMS)

- R1: Have personal account
 - F1: sign-up
 - F2: login
 - F3: logout
- R2: Main Dashboard (home) Page
 - F1: Access groups & rooms I'm in from home page
 - F2: Search for existing rooms from home page
 - F3: Create room
 - F4: Join room
 - F5: Leave room

PS 2: Unable to provide information about myself to potential group members (PROFILE)

- R1: Easily edit and add publicly viewable info to my profile
 - F1: Description (bio)
 - F2: Photo
 - F3: Courses
- R2: Provide private information for purposes of group matching based on timetabling and location
 - F1: Sync Personal Calendar
 - F2: Enter preferred meeting location
- R3: View another user's profile
 - F1: Click a user's name from anywhere on website takes you to their profile with their photo, bio, courses and active groups

PS 3: No way to form groups compatible with timetable, skill-levels and locations (GROUP FORMATION)

- R1: Group creation and management (control over who is a member of my group)
 - F1: Create Group
 - F2: Search for member (search results - overall match)
 - 2.1 - Search by location
 - 2.2 - Search by timetable
 - 2.3 - Search by skill
 - F3: Invite member
 - F4: Update group info
 - 4.1 Manually edit group info
 - 4.2 Dynamically update skills when someone join/leaves
 - F5: Remove group member
- R2: Groupless user can find a preferable group
 - F1: Search for groups (default list - overall match)
 - 1.1 - Search by location
 - 1.2 - Search by timetable
 - 1.3 - Search by skill
 - F2: Join a group
- R3: Group member can interact with group
 - F1: Chat
 - F2: Leave group

PS 4: Poor existing notification systems (NOTIFICATIONS)

- R1: Process of requesting to join notifies two users involved at each stage
 - F1: Sending join request notifies group leader
 - F2: Group leader receives notification with ability to accept/decline/view profile
 - F3: User notified if group leader accepts their join request
 - F4: User notified if group leader declines their join request
- R2: Process of inviting a user to a group notifies two users involved at each stage
 - F1: When group leader clicks invite notification sent to user
 - F2: User receives notification and can accept/decline/view group
 - F3: Group leader notified of outcome
- R3: Notified when group members want to communicate with each other
 - F1: Notified when someone sends message to chat
- R4: Aware when I have new notifications and see a list of past notifications
 - F1: Red circle appears if I have received a new notification
 - F2: Clicking on notification icon allows me to view scrollable list of past notifications

Question 2 - Sequence Diagrams (matched to user stories)

Problem Statement 1

There is no central hub for organising my involvement in UNSW Computer Science and Engineering (CSE) groups, making it difficult to keep track of group-specific information in one location.

Requirement 1

As a UNSW CSE student, I want to use a personal UniCollab account so I can participate in the UniCollab group formation ecosystem with my own identity

FEATURE 1: Students can sign up for a UniCollab account

As a student who wishes to be involved in the UniCollab group formation process

So that I can manage my UniCollab groups safely and securely

I want to create a personal, password protected UniCollab account

SCENARIO: A student wishes to join UniCollab and participate in a group formation process

Given I am on the the UniCollab Welcome Page

When I click 'Get Started'

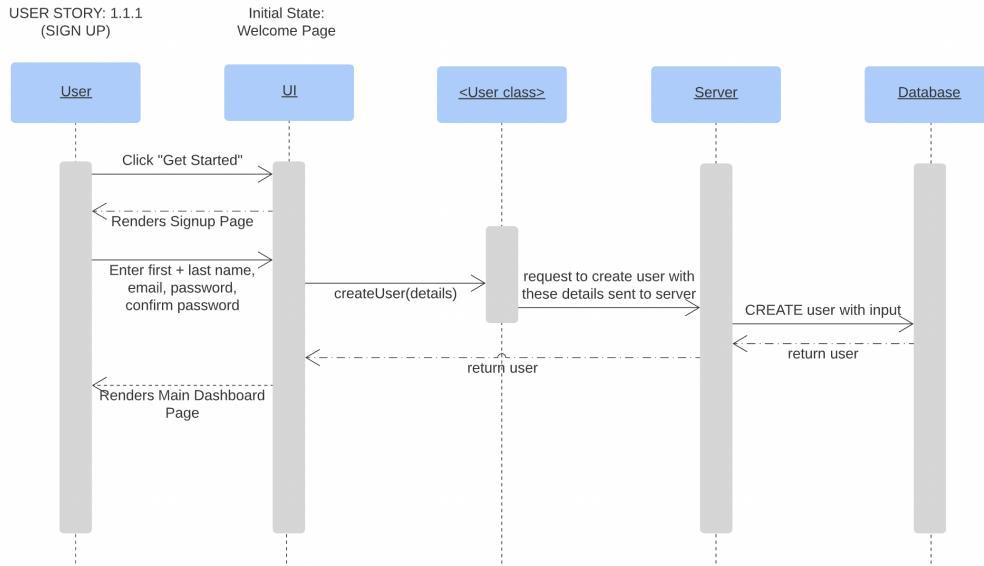
Then I am taken to the Sign Up Page

When I enter my 'First Name', 'Last Name', 'Email', 'Password' and 'Confirm Password'

And I click the 'Sign Up' button

Then I am logged in to my newly created UniCollab account

And am taken to the Main Dashboard Page



FEATURE 2: Students can log in to their UniCollab account
As an existing UniCollab user
So that I can access and manage my UniCollab groups
I want to login to my UniCollab account

SCENARIO: A student wishes to login to their UniCollab account to view their UniCollab groups

Given I am on the UniCollab Welcome page

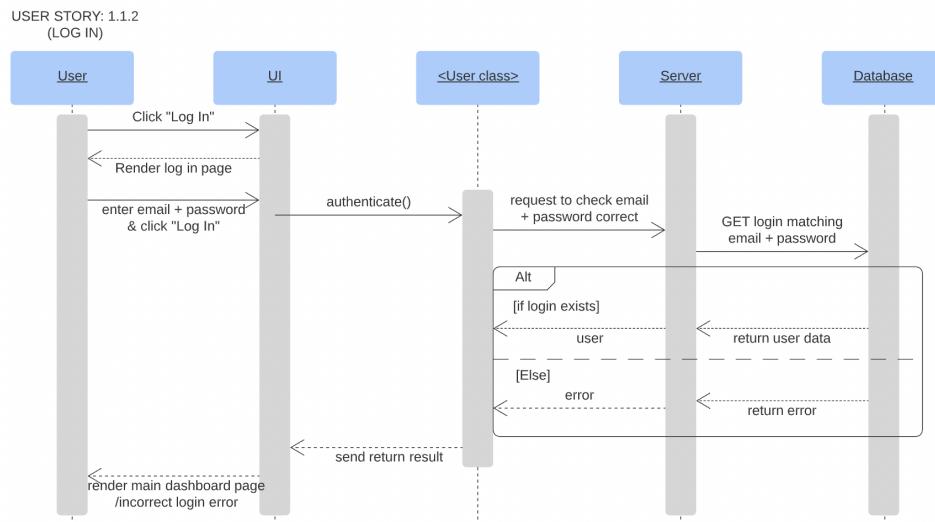
When I click 'Log In'

Then I am taken to the Login Page

When I enter the correct email and password for my account

Then I am logged in to my UniCollab account

And I am taken to the UniCollab Main Dashboard Page



FEATURE 3: Students can log out of their UniCollab account

As a logged in UniCollab user

So that I can discontinue using UniCollab and prevent unauthorised actions

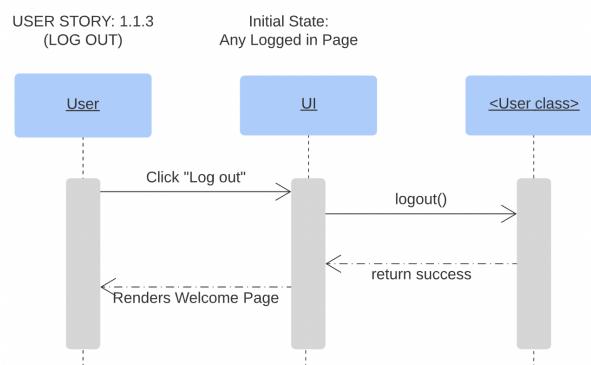
I want to logout of my UniCollab account

SCENARIO: A student wishes to logout of their UniCollab account

Given I am on my UniCollab Main Dashboard Page (home)

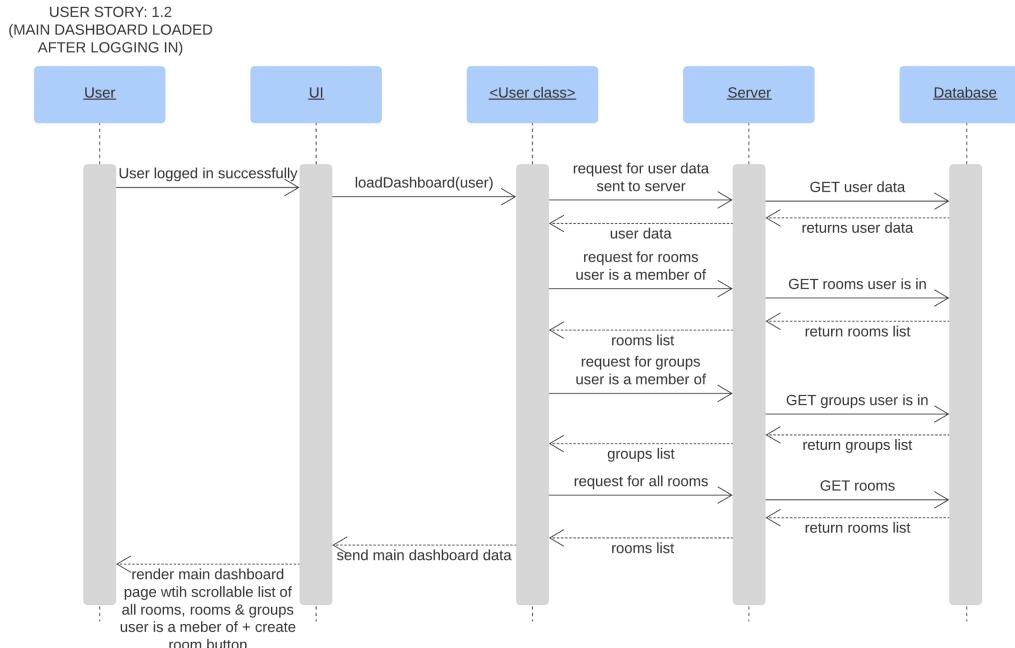
When I click the 'Logout' button

Then I am taken to the UniCollab Welcome Page



Requirement 2

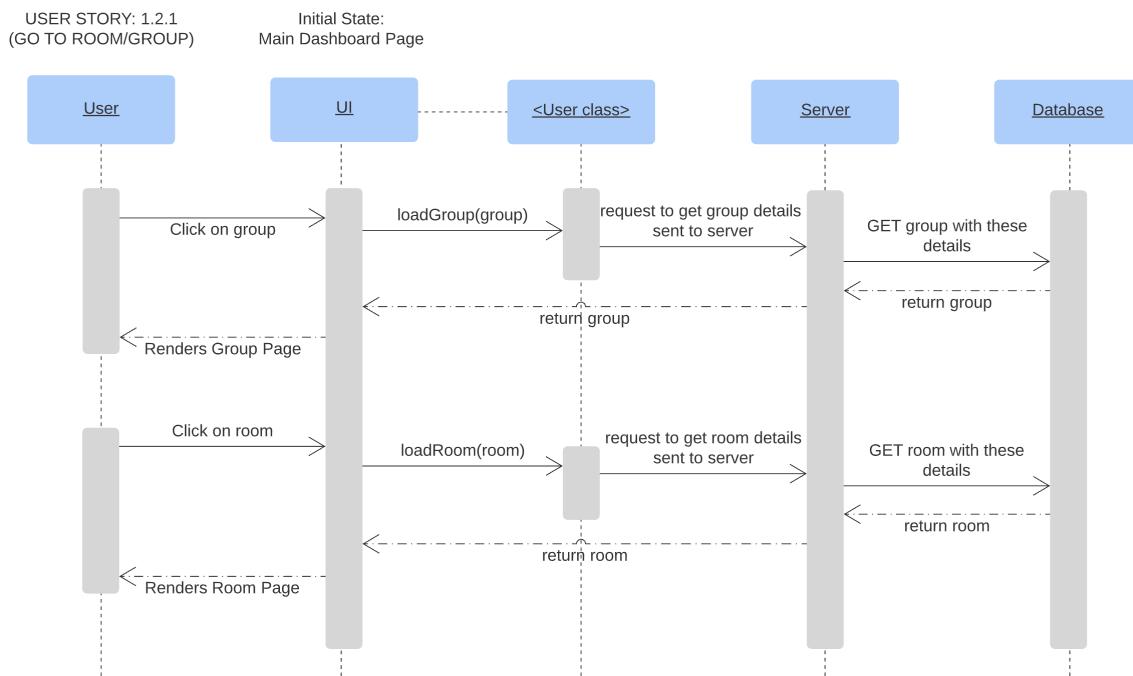
As a user of UniCollab, I would like to access a personalised dashboard page so that I can access my rooms and groups as well as create, join and leave rooms from one location



This diagram shows what data will be loaded to display the users main dashboard page. The below features detail how users can interact with the data displayed on the dashboard page.

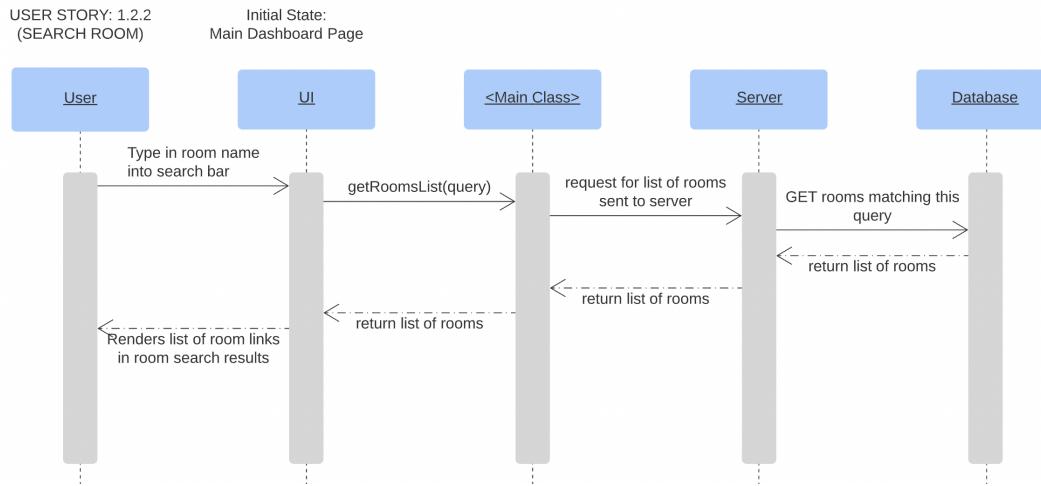
FEATURE 1: Students can access rooms and groups that they are currently in
As a UniCollab user
So that I can directly navigate to groups and rooms I am a part of
I want to view a scrollable list of my current groups and rooms on my dashboard page

SCENARIO: A student wishes to have quick access to their groups and rooms
Given I am an active UniCollab user
When I log in to my account
Then I see a scrollable list of my groups and rooms from my dashboard page
When I click on a particular group or room
Then I will be transported to that group/room's page



FEATURE 2: Students can search for existing rooms
As an existing UniCollab user
So that I can find rooms to join
I want to view a searchable list of existing rooms from my dashboard page

SCENARIO: A student wants to find a room relating to their course/project
Given I am an active UniCollab user
When I log in to my account
Then I see a scrollable list of existing rooms with a search bar on my dashboard
When I enter a query into the search bar
Then I see a filtered list of rooms matching my query



FEATURE 3: Can create a UniCollab Room

As a user of UniCollab

So that students can form groups relevant to a course or project

I want to create a UniCollab Room for them to join and form groups with each other

SCENARIO: At the start of the term, the lecturer of SENG2021 wishes to create a UniCollab Room for his students so they can form groups for the group project

Given I am on the Main Dashboard Page for my account

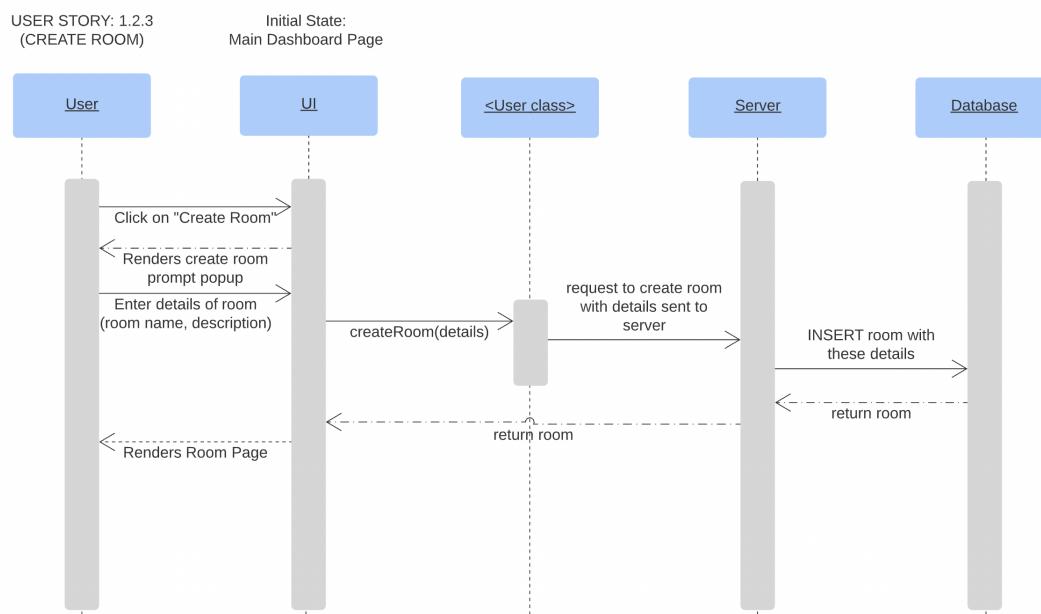
When I click on the 'Create Room' button

Then I am prompted to enter a room name and description

When I enter a room name ('SENG2021 21T1') and description ('Form SENG2021 Project groups here!') and click 'Create Room'

Then I am taken to my newly created UniCollab Room page

And my room is added to the existing rooms list viewable on the dashboard page



FEATURE 4: Joining UniCollab rooms

As a user of UniCollab

So that I can form a group with other people mutually interested in the room topic

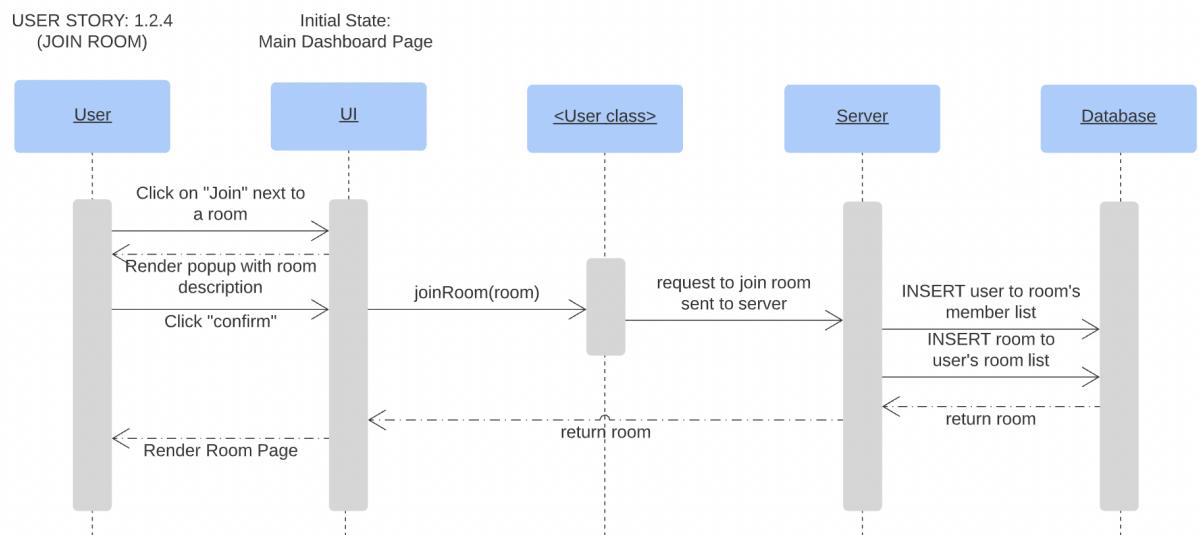
I want to join an existing UniCollab room

SCENARIO: A student wishes to join the SENG2021 room to form a SENG2021 project group with other SENG2021 members

Given I am on my main dashboard page

When I click the 'Join' button next to a room name on the scrollable list

Then I am taken to that room page



FEATURE 5: Leaving UniCollab rooms

As a student using UniCollab

So that I can remove myself from a room I no longer wish to be a part of

I want to leave that UniCollab room

SCENARIO: A student dropped COMP1531 and no longer needs to form a group for that course

Given I am on my main dashboard page

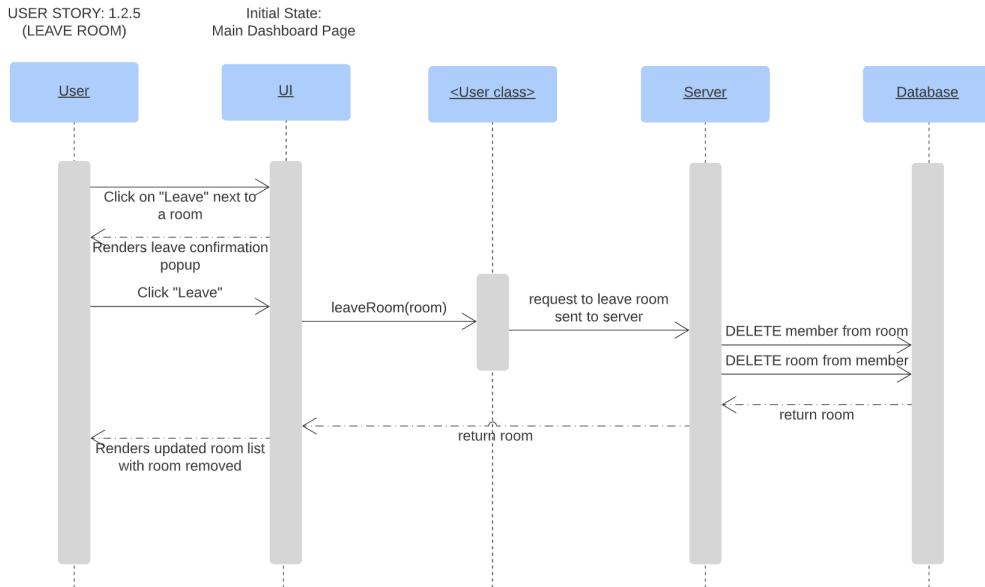
When I click on the Leave button next to one of my rooms (e.g. COMP1531)

Then I see a popup confirming if I want to leave room

When I click 'Leave'

Then I will be removed from that room and any UniCollab group I am involved with in that room

And the room will be removed from my UniCollab dashboard page



Problem Statement 2

It is currently difficult to view information of other UNSW CSE students, as there is no common interaction application.

Requirement 1

As a UniCollab user, I would like to be able to add and update my personal information, so that it is always relevant and accurately reflects who I am to other users viewing my profile

FEATURE 1: Students can update their profile page biography

As a student using UniCollab

So that I can convey information about myself to other UniCollab users

I want to add/edit my own biography

SCENARIO: A student wishes to update their biography to reflect their interests, hobbies, co-curricular commitments and other personal information

Given I am on my own profile page

When I click the 'Edit' button

Then I can edit the text of my biography

When I type in a biography and click the 'Confirm Changes' button

Then my biography will be updated and viewable to other users

FEATURE 2: Students can update their profile photo

As a student using UniCollab

So that I can display an updated image of myself

I want to change my existing profile page to a different image

SCENARIO: A student wishes to update their profile picture, to one that has been taken more recently

Given I am on my own profile page

When I click the 'Upload Profile Picture' button

Then a popup will open, prompting me to update my profile picture

When I select an image from my device and click the 'Upload' button

Then my profile picture will be updated and viewable to other users

FEATURE 3: Students can update the completed courses on their profile page

As a student using UniCollab

So that I can update my completed course list and hence the skills I possess

I want to edit my own completed courses list

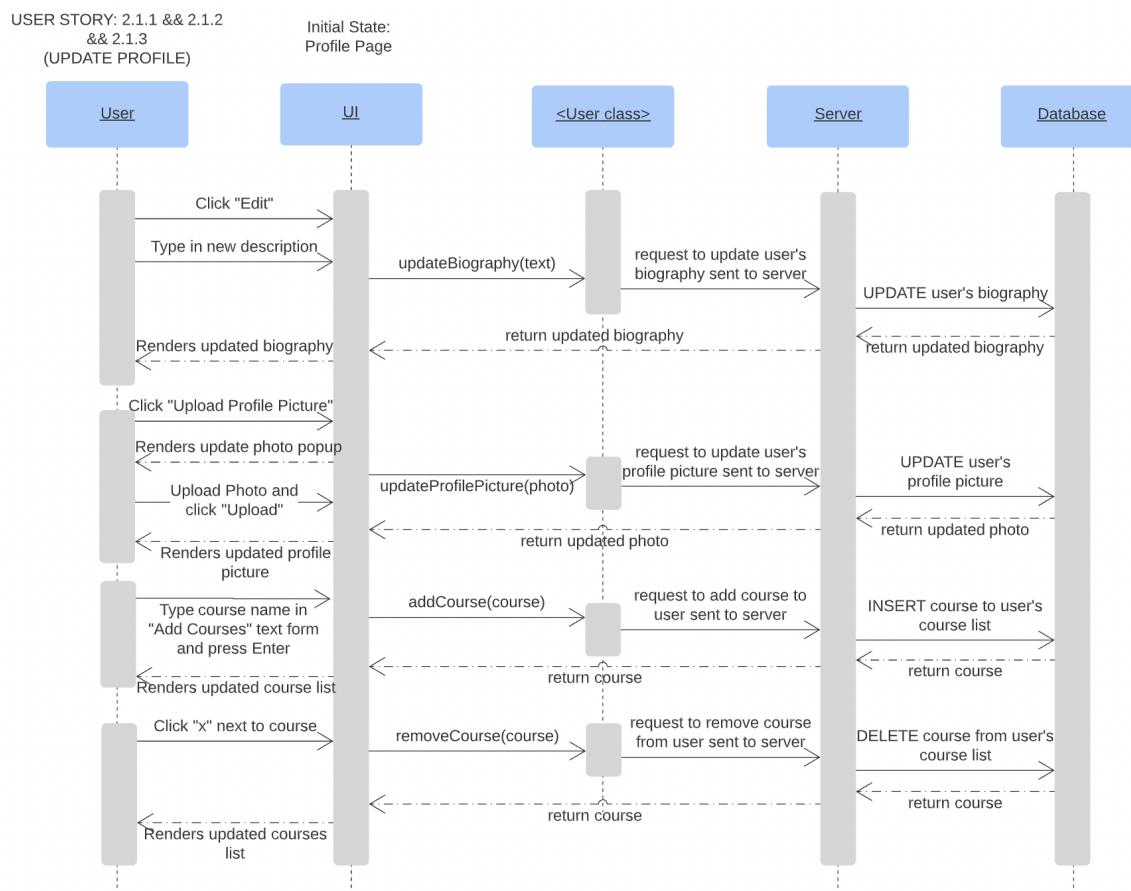
SCENARIO: A student has completed COMP2521 and wishes to add this UNSW course to their list of completed courses

Given I am on my own profile page

Then I can remove courses by clicking the cross next to them or add courses by typing in a valid UNSW course name

When I remove/add a course

Then my profile page will update to reflect these changes



Requirement 2

As a UniCollab user, I would like to enter relevant private data about myself such as upload my personal calendar and location so that I can optimise group matching by forming groups with people whose availabilities and location align with mine

FEATURE 1: Sync UniCollab Calendar with Google Calendar

As a user looking to form groups with people that suit my schedule

So that I can easily find groups with compatible availabilities for group meetings

I want to sync my UniCollab Calendar with my personal calendar

SCENARIO: A student wants their UniCollab calendar to align with their Google calendar including all their personal, work and university commitments.

Given I am on my Profile Page

When I click the Calendar button

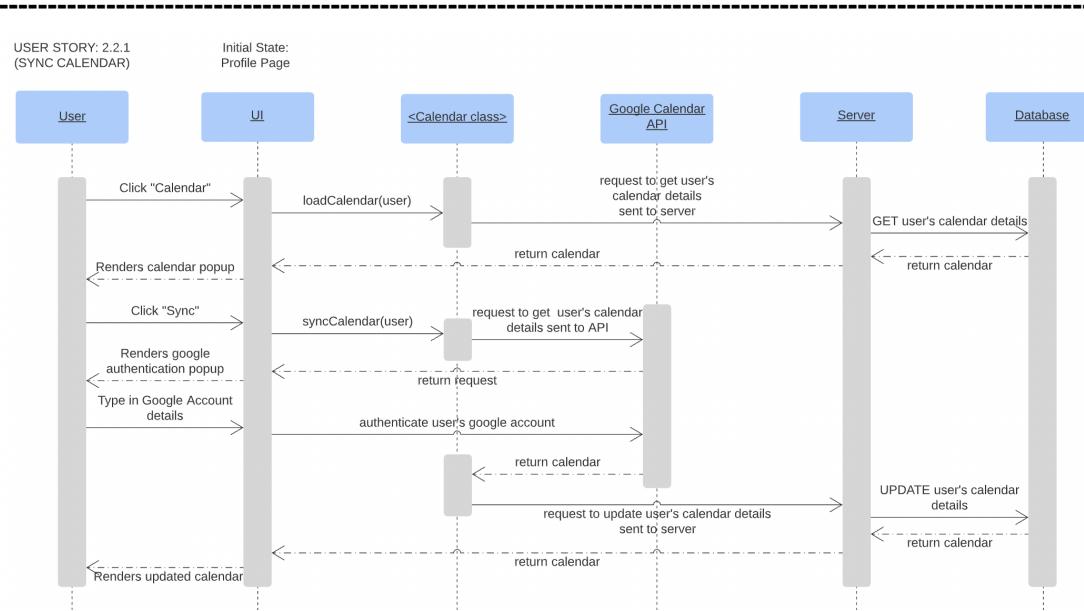
Then a Calendar Popup appears of the current week with arrows to cycle to the next

When I click the Sync button

Then I am prompted to enter my Google Account details by a popup

When I enter my details

Then the UniCollab calendar syncs with my Google Calendar, meaning it now accurately displays all my Google Calendar commitments



FEATURE 2: Enter preferred meeting location to profile

As a UniCollab user

So that I can easily find groups and members who want to meet near the same place

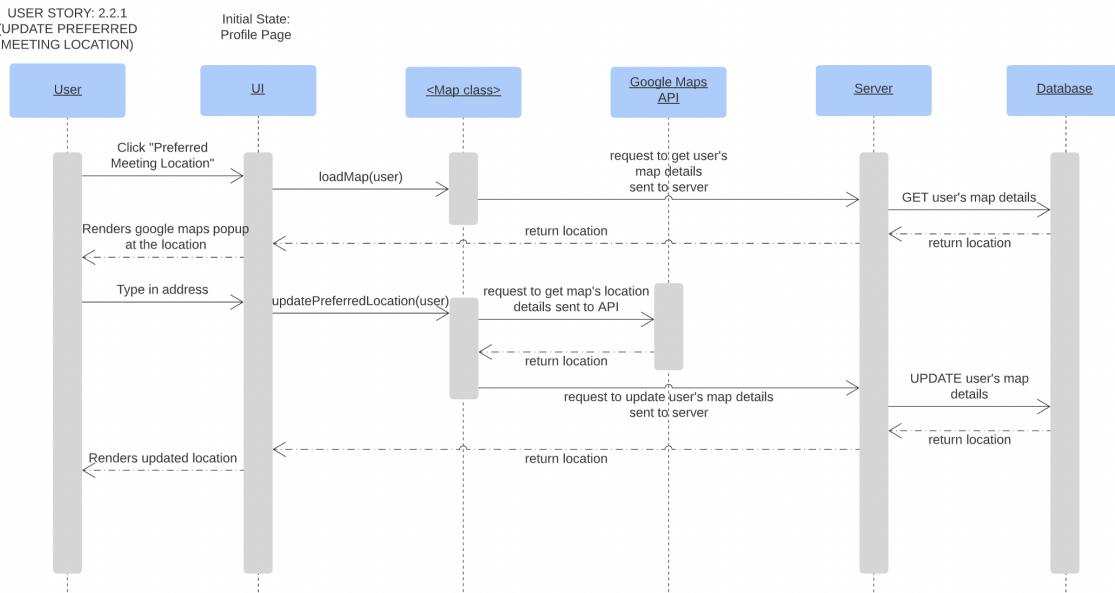
I want to enter my location into UniCollab

SCENARIO: A student wants to be able to search for groups and users based on their distance

Given I am on my Profile Page and have clicked 'Edit Profile'

When I type the address of my preferred meeting location into the 'Preferred Meeting Location' field

Then my location is updated for group matching purposes



Requirement 3

As a UniCollab user, I would like to view a person's profile page so that I can learn more about their interests, groups and their courses to determine if I want to group with them

FEATURE 1: Able to navigate another user's profile from any instance they appear on the UniCollab site to view information about them

As a UniCollab user

So that I can view information regarding another user

I want to be able to click their name or image and be navigated to their profile page

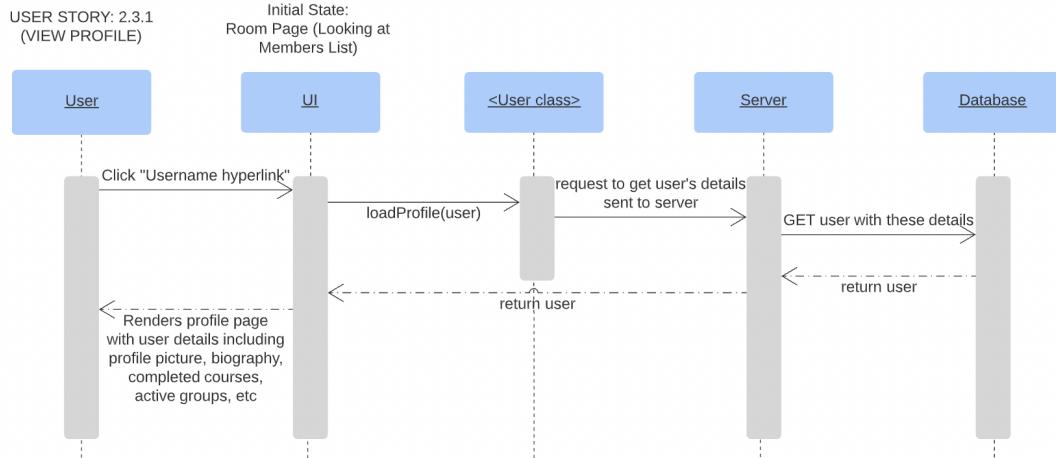
SCENARIO: A student sees a groupless member in a room that they are both part of

Given I am viewing a list of members in a UniCollab room

When I click on another user's name or profile picture

Then I will be navigated to their profile page

And able to view their profile picture, biography, completed courses and active groups



Problem Statement 3

Forming compatible groups based on location, scheduling and skill-matching for specific UNSW projects and courses is difficult and often requires the use of various platforms.

Requirement 1

As a group leader of a UniCollab group, I would like to manage the information regarding my group and control the list of members it holds.

FEATURE 1: Create a group within a UniCollab room

As a groupless user looking to become a group leader

So that I can form my own group and manage who is a part of it

I want to create a new UniCollab group

SCENARIO: A groupless member in the SENG2021 room wants to form their own group and appeal to people with similar mindsets

Given I am in a UniCollab Room and I am not in a group

When I click the 'Create Group' button from the Room Page

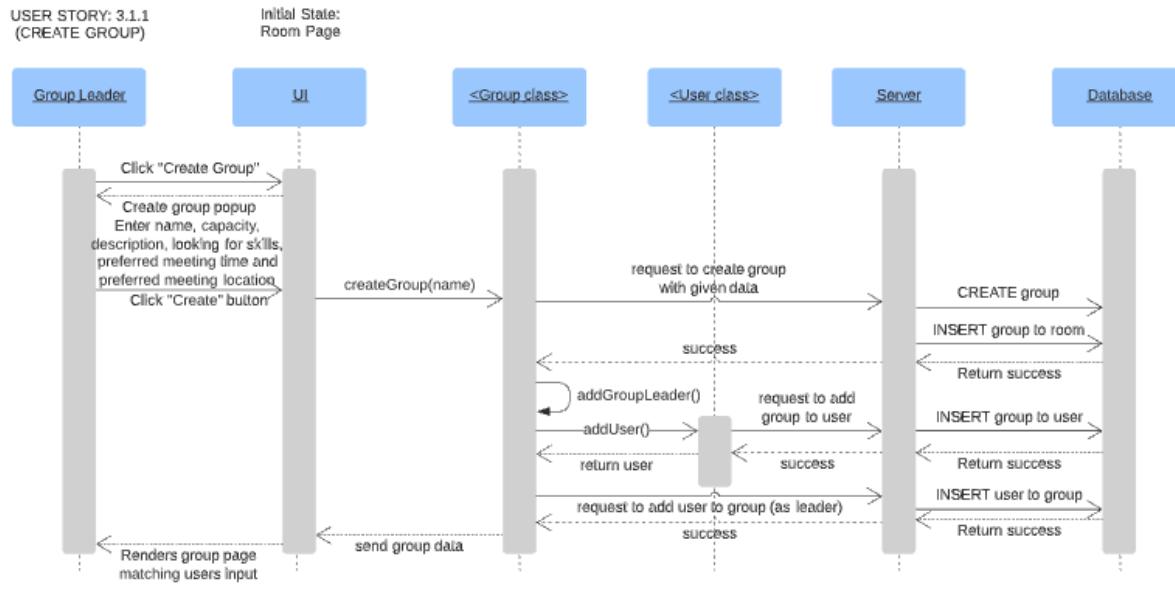
Then I will see a prompt with 'Select Image' and editable fields for my 'Group Name', 'Group Capacity', 'Description', 'Looking For Skill', 'Preferred Meeting Times' and Preferred Meeting Locations

When I fill in these fields

And I click 'Create'

Then my group is created with all this information, meaning it will appear in the groups list of the UniCollab room where it is viewable by others

And I am taken to my Group Page



FEATURE 2: A group leader can search for users in the room based on their preferences for potential candidates

As a UniCollab group leader

So that I can find optimal candidates to invite to my group

I want to be able to view a list of members based off our location, timetable and skill compatibilities

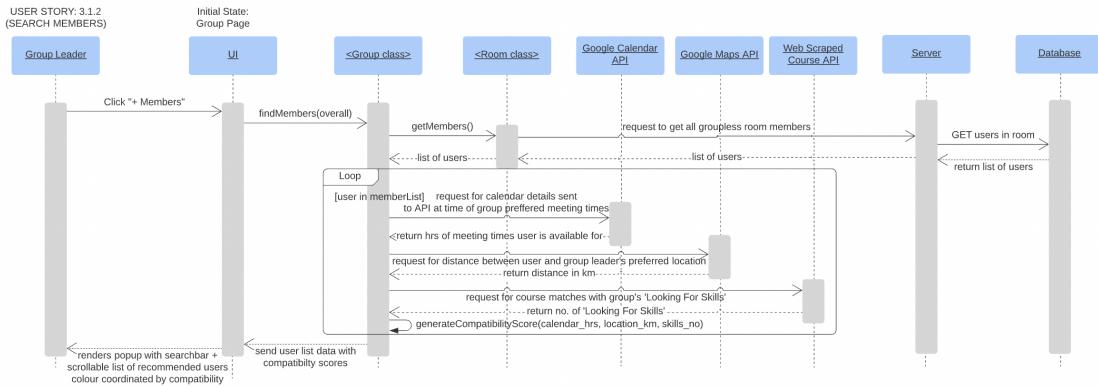
SCENARIO: A SENG2021 group leader would like to fill their group by finding users that are most compatible in reference to meeting times, locations and skill sets

Given I am a group leader on my group page

When I click the '+' next to Members

Then a popup will open with a search bar and list of groupless members in the current room ranked and colour-coded by the following criteria:

- Green → user is available for 70 - 100% of the groups preferred meeting times (Google Calendar API), is within 15km of group leader (Google Maps API) and matches 50-100% of the groups 'looking for' skills (Custom Web Scrapped API from UNSW course websites)
- Yellow → user is available for 30 - 70% of the groups preferred meeting times, is within 30km of group leader and matches 20 - 50% of the groups 'looking for' skills
- Red → user is available for 0 - 30% of the groups preferred meeting times, is over 30km away from group leader and matches 0 - 20% of the groups 'looking for' skills



FEATURE 2.1: A group leader can search a list of recommended group members in a room, based on location compatibility

As a UniCollab group leader

So that I can find members within the room that are located near me

I want to view a searchable list of members based off their distance from my preferred meeting location

SCENARIO: A SENG2021 group leader would like to fill their group by finding users that live nearby so in-person group meetings can be easily facilitated

Given I have clicked ‘+’ next to ‘Members’ on my group page

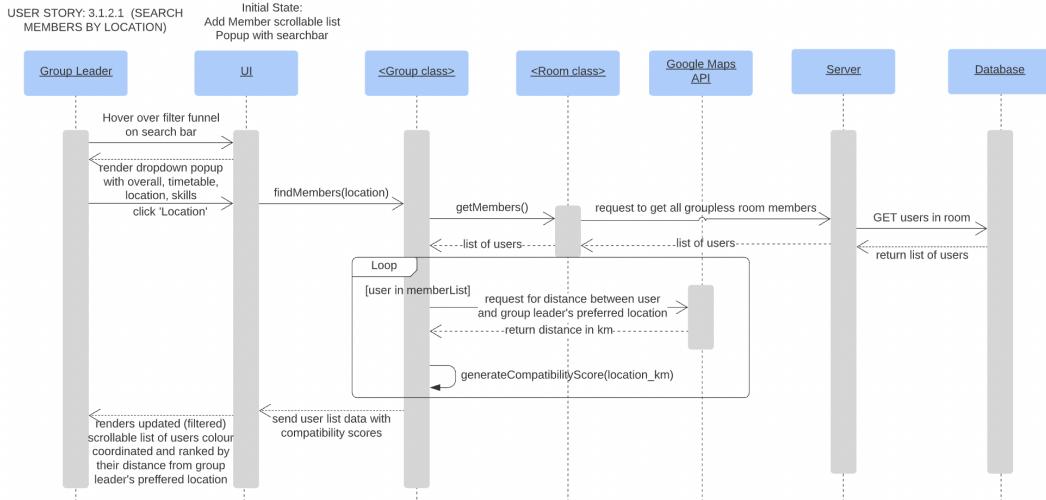
When I hover over the filter funnel icon next to the search bar

Then a drop-down popup appears with the options ‘Overall Match’ ‘Location’ ‘Timetable’ and ‘Skills’

When I click ‘Location’

Then I will see a list of members ranked and colour-coded based on their proximity to my preferred location:

- Green → within 15km of my preferred location
- Yellow → within 30km of my preferred location
- Red → over 30km away from my preferred location



FEATURE 2.2: A group leader can search a list of recommended group members in a room, based on timetable compatibility

As a UniCollab group leader

So that I can find members that are available for my groups preferred meeting times
I want to view a searchable list of members based off our calendar compatibility

SCENARIO: A SENG2021 group leader would like to fill their group by finding users that are available for the meeting times they put forward

Given I have clicked '+' next to 'Members' on my group page

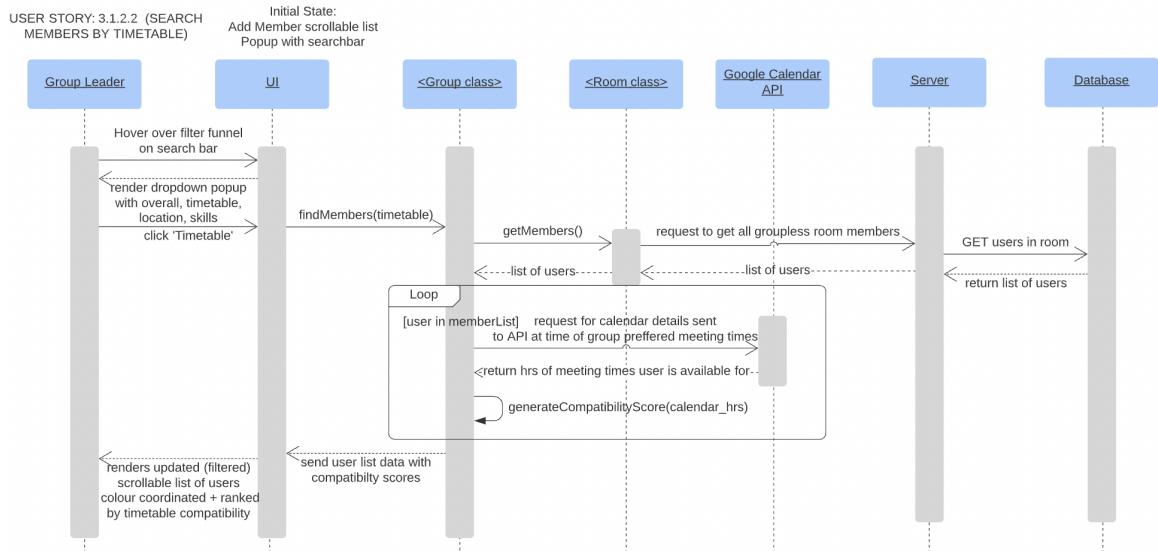
When I hover over the filter funnel icon next to the search bar

Then a drop-down popup appears with the options 'Overall Match' 'Location' 'Timetable' and 'Skills'

When I click 'Timetable'

Then I will see a list of members ranked and colour-coded based on how compatible their calendar is with my preferred group meeting times:

- Green → user is available for 70 - 100% of groups preferred meeting times
 - Yellow → user is available for 30 - 70% of groups preferred meeting times
 - Red → user is available for 0 - 30% of groups preferred meeting times
-



FEATURE 2.3: A group leader can search for recommended group members in a room, based on skill-matching between a user and the group's desired skills

As a UniCollab group leader

So that I can find members within the room that possess my desired 'Looking For' skills
I want to view a searchable list of members based on the correlation of my group's 'Looking For' skills and the courses that a user has completed (and hence, the skills that they possess)

SCENARIO: A SENG2021 group leader would like to fill their group by finding users that are available for the meeting times they put forward

Given I have clicked '+' next to 'Members' on my group page

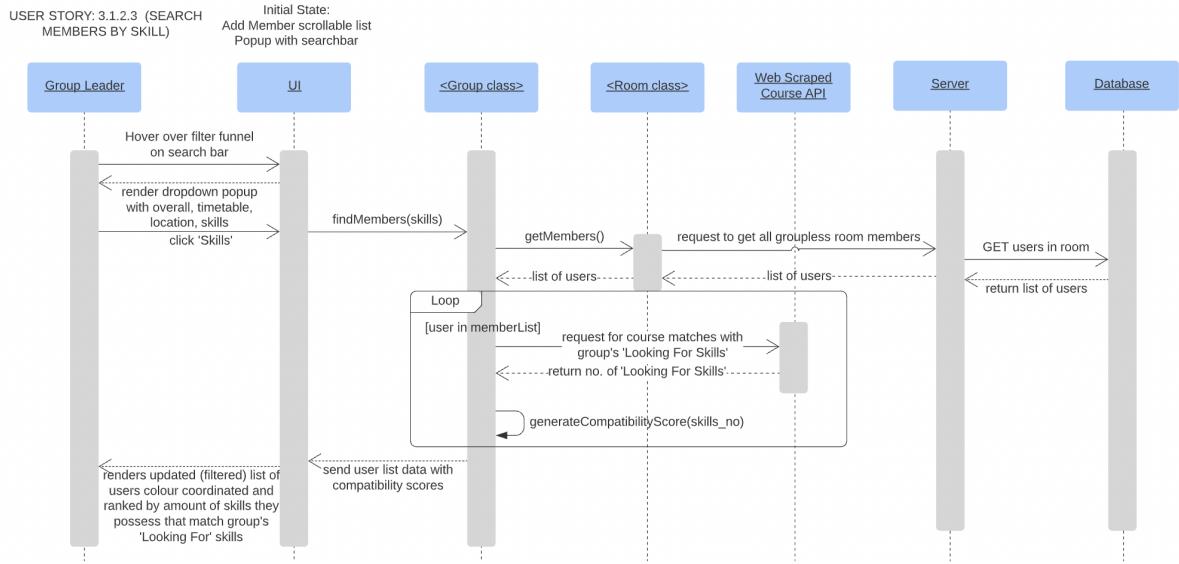
When I hover over the filter funnel icon next to the search bar

Then a drop-down popup appears with the options 'Overall Match' 'Location' 'Timetable' and 'Skills'

When I click 'Skills'

Then I will see a list of members ranked and colour-coded based on how many of my groups 'Looking For' skills they possess:

- Green → user matches 50-100% of the groups 'looking for' skills
 - Yellow → user matches 20 - 50% of the groups 'looking for' skills
 - Red → user matches 0 - 20% of the groups 'looking for' skills
-



FEATURE 3: Invite users to my group

As a UniCollab group leader

So that I can select users with complementary skills and mindsets to me and let them know I want them in my group

I want to invite them to join my group

SCENARIO: A group leader wishes to invite a certain individual with the right skillset to join their groups

Given I am a group leader on my group page

When I click the '+' button next to 'Members'

Then I am shown a list of scrollable users who do not currently belong to a group

When I click the 'Invite' button next to a user

Then a popup appears informing 'Invite Successful'

And the invitation is sent to the user

FEATURE 4: Update information pertaining to my group

FEATURE 4.1: Change the group information fields

As a UniCollab group leader

So that I can update my group's information

I want to edit my group's image, description, skills, or maximum capacity fields

SCENARIO: A group leader has discovered another skill his group needs and wishes to advertise this in the 'Looking For' section of his group page

Given I am a group leader on my group's page

When I click the 'Edit' button

Then all fields become editable to the group leader only

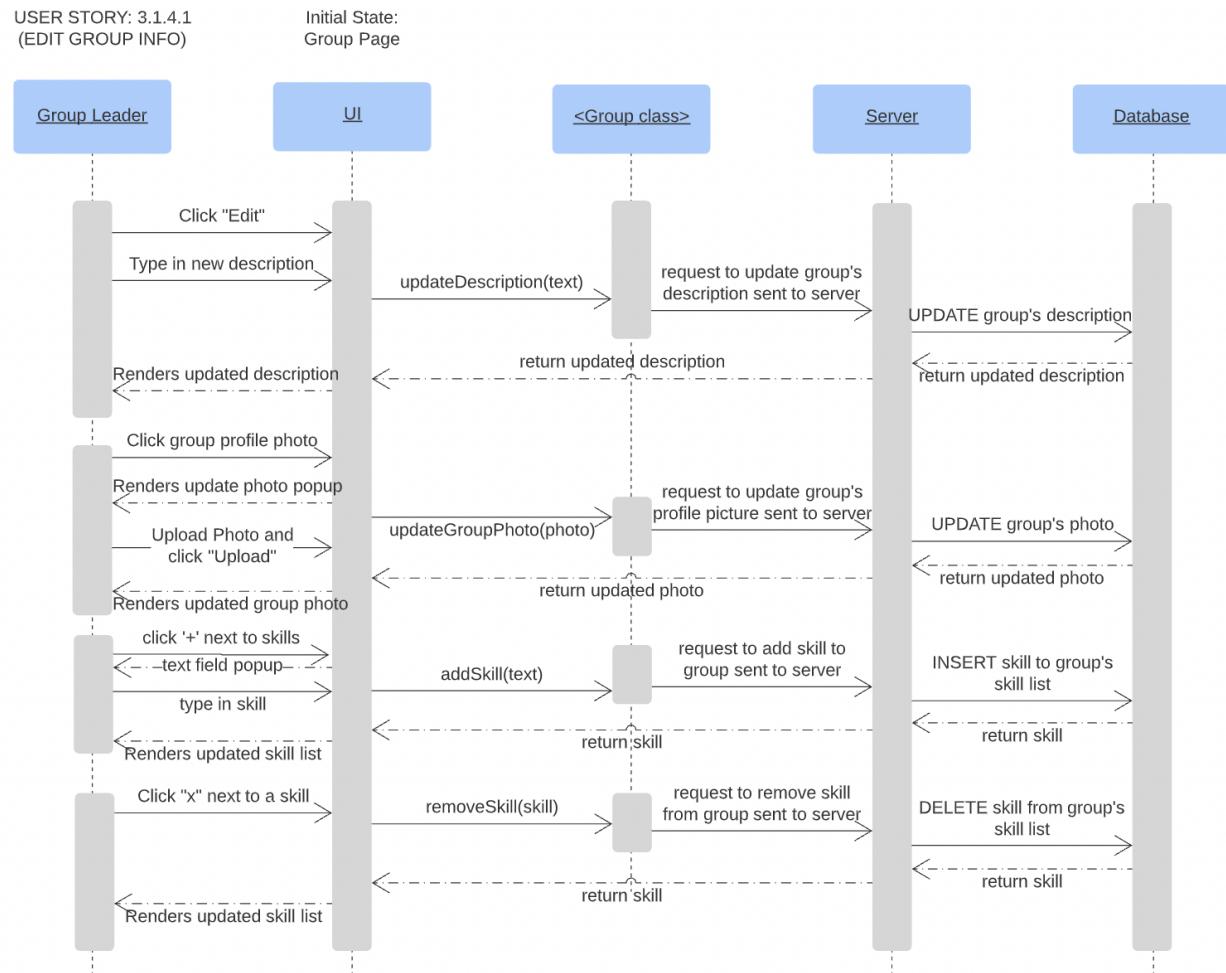
When I click '+' next to the "Looking For" skills section

Then A popup editable text field appears next to the '+' icon

When I type in a skill

And click enter

Then my groups "Looking For" skills will be updated to reflect this new skill



FEATURE 4.2: Dynamically update the 'We Have' and 'Looking For' skills tables as well as capacity when the students join/leave the room

As a member of a UniCollab group

So that I can see the current skills our group is possessing and searching for

I want to ensure that the skills the group has and is looking for are dynamically changed when a user leaves or joins the group

SCENARIO: A group member possesses the skills of 'Object-Oriented Design' and 'SQLite' by completing the courses COMP2511 and COMP3311. No other member has these skills and this group member has decided to leave the group.

Given I am on the group page for a group I am in

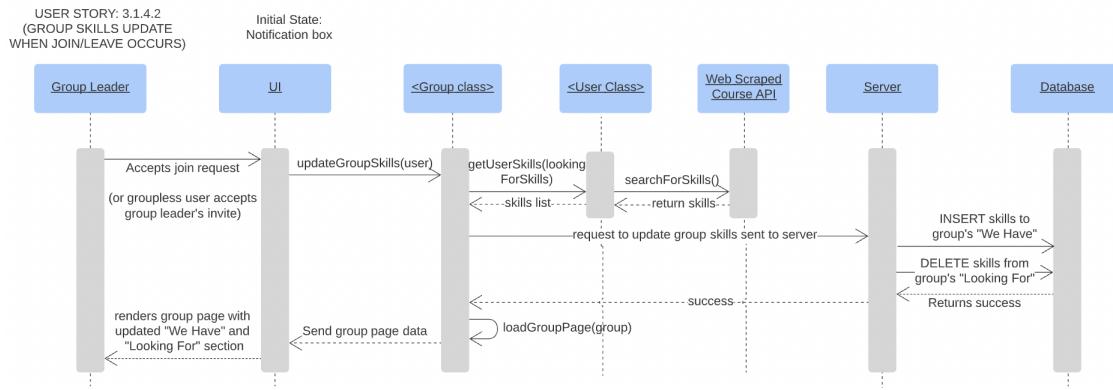
When I click the 'Leave Group' button

Then a popup will appear, asking me to confirm/cancel this decision

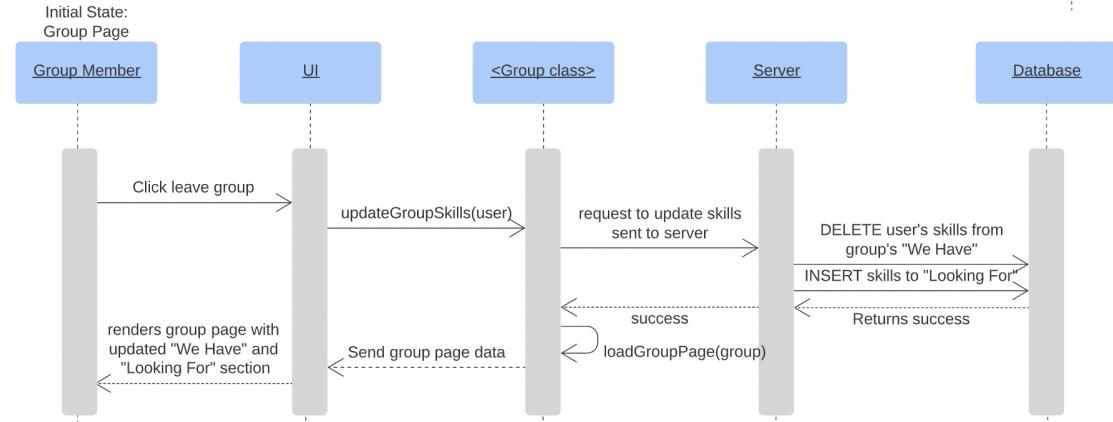
When I click 'Leave', the current capacity will be decreased by one

And the skills 'Object-Oriented Design' and 'SQLite' will move from the 'We have' component to the 'Looking For' component

And the group will no longer appear on my main dashboard page or profile page



Our `searchForSkills` fn between the user class and Web Scrapped Course API will work by looping through each course in the users list of courses and scrape the course outlines of each course through navigating a web cms link to match the `lookingForSkills` of the group. It will put these skills into a list and return it to the group class so that it can update its skills to reflect the addition of a new member.



FEATURE 5: Remove members from my group

As a UniCollab group leader

So that I can ensure my group is only filled with people of complementary skills and similar mindsets to me

I want to remove users from my group

SCENARIO: A group leader no longer feels like a certain group member meshes well with their group atmosphere

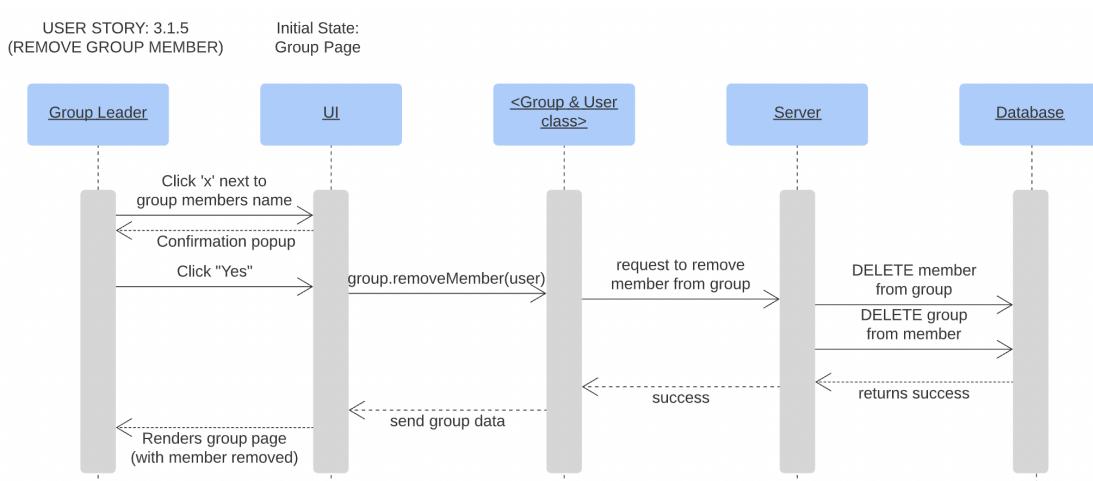
Given I am a group leader on my group page

When I click the 'Edit' button

Then I can see red crosses next to my members' names

When I click on a the cross next to their name

Then that member is removed from the group



Requirement 2

As a UniCollab member who does not belong to a group for the room I am in, I would like a system that assists me to find appropriate groups and allow me to join them.

FEATURE 1: Room page displays a searchable list of existing groups based on overall compatibility in relation to location, timetabling and skills combined

As a groupless user

So that I can find a group that best matches my location, calendar and skillset

I want to be able to view a list of groups ranked by these criteria

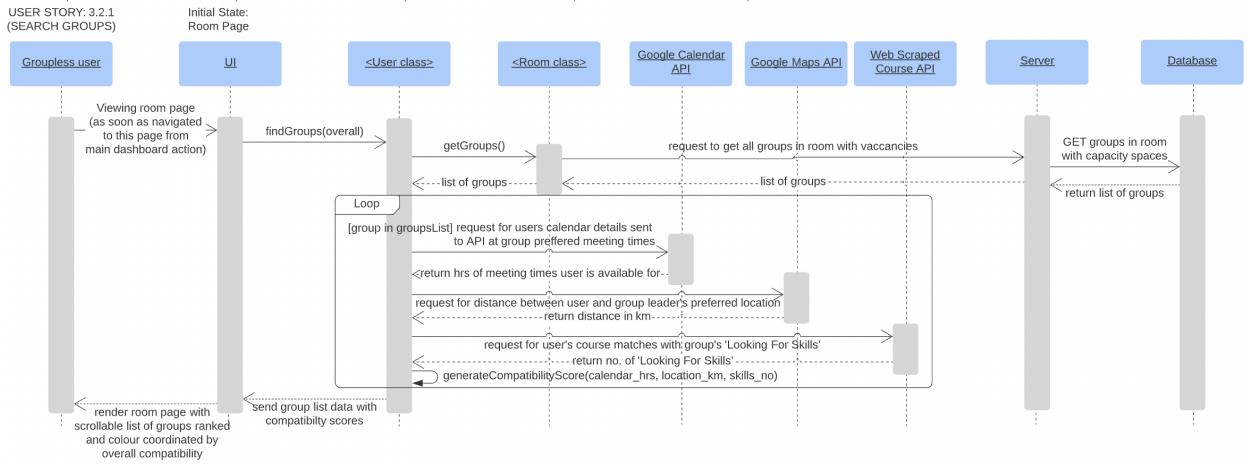
SCENARIO: A UniCollab user would like to find an overall best fit group for their needs so that they are able to achieve an optimal result in their group work

Given I have joined a room

When I click on the room page

Then I will see a search bar with a list of existing groups ranked and colour-coded by the following criteria:

- Green → I am available for 70 - 100% of the groups preferred meeting times (Google Calendar API), within 15km of all group members (Google Maps API) and match 50-100% of the groups 'looking for' skills (Custom Web Scrapped API from UNSW course websites)
- Yellow → I am available for 30 - 70% of the groups preferred meeting times, within 30km of all group members and match 20 - 50% of the groups 'looking for' skills
- Red → I am available for 0 - 30% of the groups preferred meeting times, am over 30km away from all group members and match 0 - 20% of the groups 'looking for' skills



FEATURE 1.1: User can filter the list of groups, based on location compatibility

As a groupless user

So that I can find groups with members located near me

I want to view a searchable list of group based off their distance from my entered postcode

SCENARIO: A UniCollab user wants to find a group where members live nearby so that they can easily meet with in person

Given I am on a room page

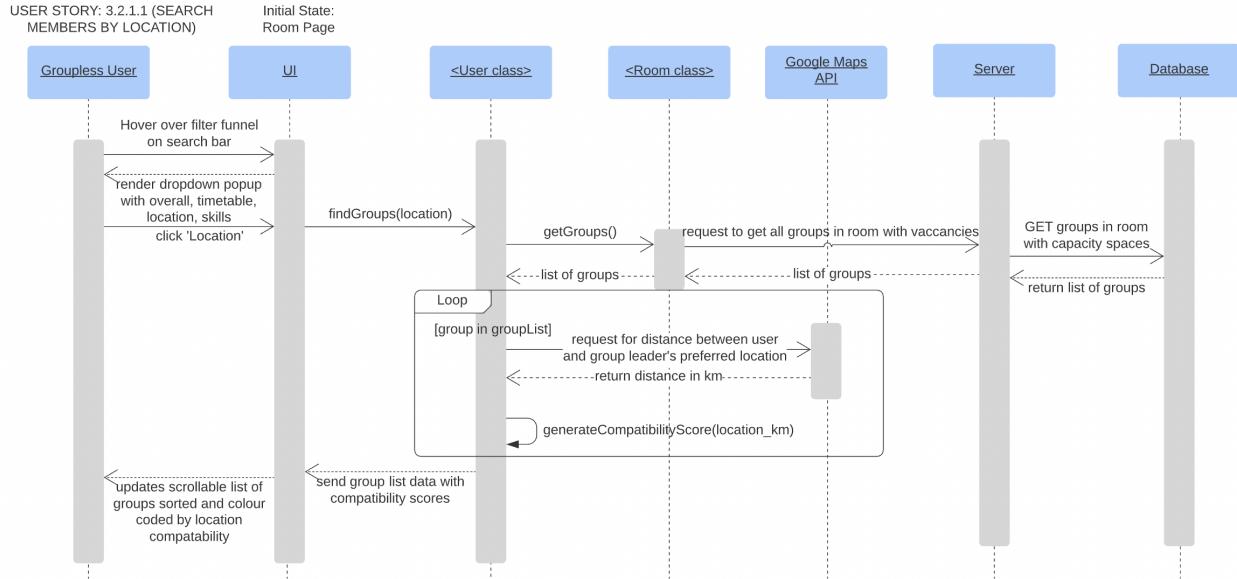
When I hover over the filter funnel icon next to the group search bar

Then a drop-down popup appears with the options 'Overall Match' 'Location' 'Timetable' and 'Skills'

When I click 'Location'

Then I will see a list of groups ranked and colour-coded based on their proximity to me:

- Green → all members are within 15km of me
 - Yellow → all members are within 30km of me
 - Red → all members are 30km away from me
-



FEATURE 1.2: User can filter the list of groups on the room page based of their availability for the group meeting times

As a groupless user

So that I can find groups that I know I will be available to meet up with

I want to view a searchable list of groups based off our calendar compatibility

SCENARIO: A user wants to find a group that they can be certain of how available they are to meet with them based of their calendar

Given I am on a room page

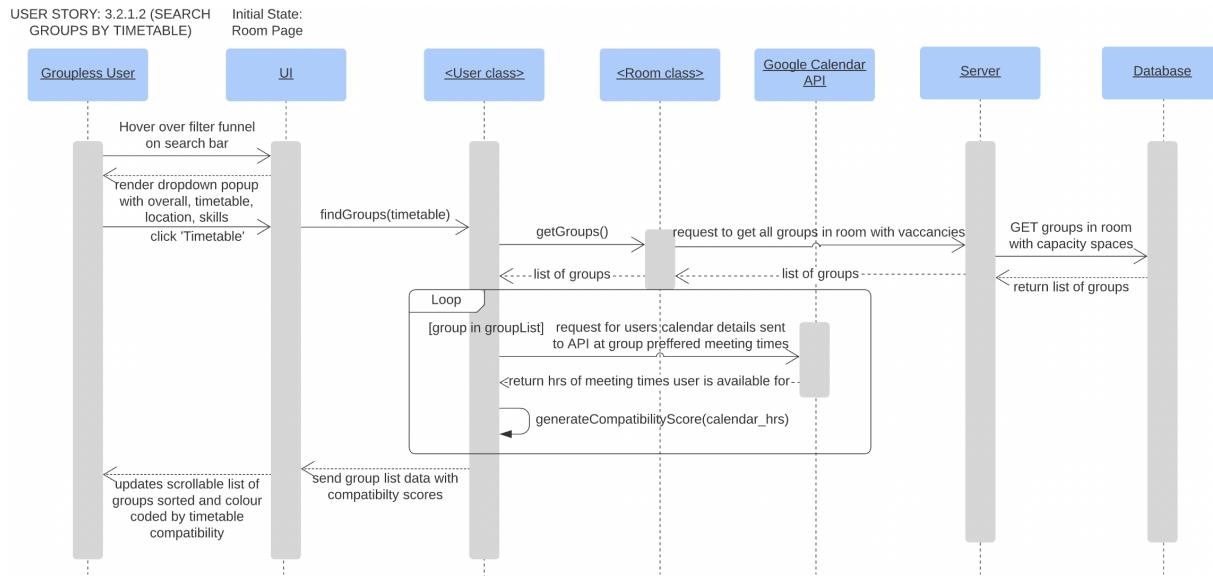
When I hover over the filter funnel icon next to the group search bar

Then a drop-down popup appears with the options 'Overall Match' 'Location' 'Timetable' and 'Skills'

When I click 'Timetable'

Then I will see a list of groups ranked and colour-coded based on our calendar compatibilities:

- Green → I'm available for 70 - 100% of groups preferred meeting times
 - Yellow → I'm available for 30 - 70% of groups preferred meeting times
 - Red → I'm available for 0 - 30% of groups preferred meeting times
-



FEATURE 1.3: User can filter the list of groups on the room page, based on skill-matching between me and the group's desired skills

As a groupless user

So that I can find groups within the room that I match the 'Looking For' skills for
I want to view a searchable list of groups based on the correlation of their 'Looking For' skills and the courses that I have completed (and hence, the skills I possess)

SCENARIO: A groupless user would like to find a group that they know they can be an asset to based on what courses they've completed and hence what skills they have

Given I am on a room page

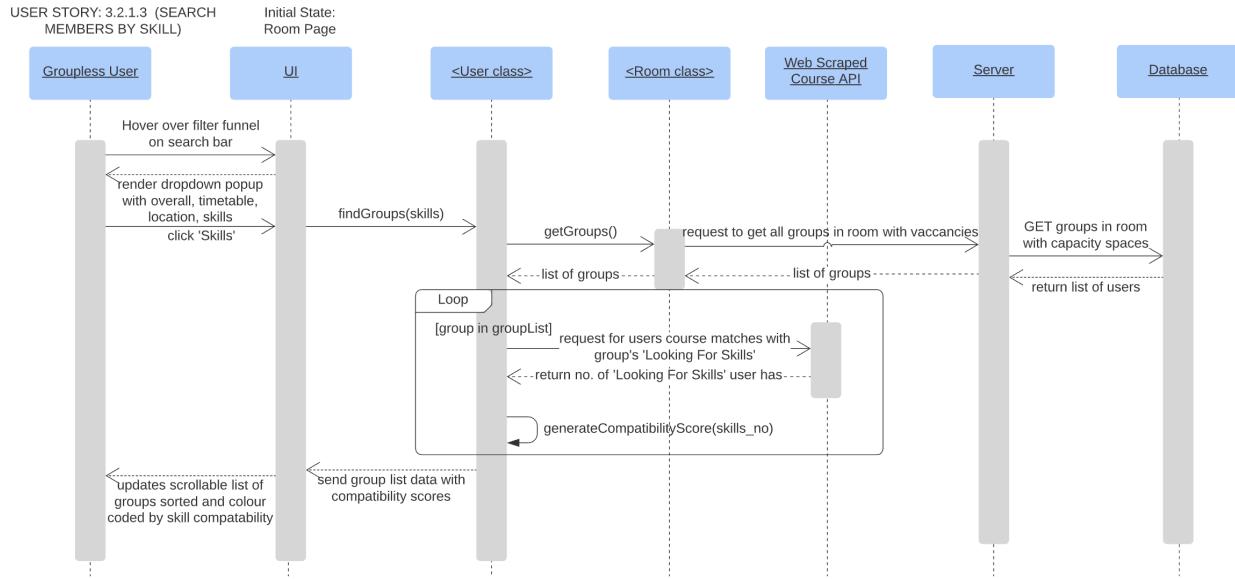
When I hover over the filter funnel icon next to the group search bar

Then a drop-down popup appears with the options 'Overall Match' 'Location' 'Timetable' and 'Skills'

When I click 'Skills'

Then I will see a list of groups ranked and colour-coded based on how many of their 'looking for' skills I have:

- Green → I match 50-100% of the groups 'looking for' skills
- Yellow → I match 20 - 50% of the groups 'looking for' skills
- Red → I match 0 - 20% of the groups 'looking for' skills



FEATURE 2: Users may request to join a group

As a groupless UniCollab user

So that I can communicate my desire to join a group I am interested in
I want to send a join request message to the group leader

SCENARIO: A groupless user in the room 'SENG2011 Project' wishes to join a specific group

Given I am a groupless UniCollab user the room 'SENG2011 Project'

When I click on a group's name or image

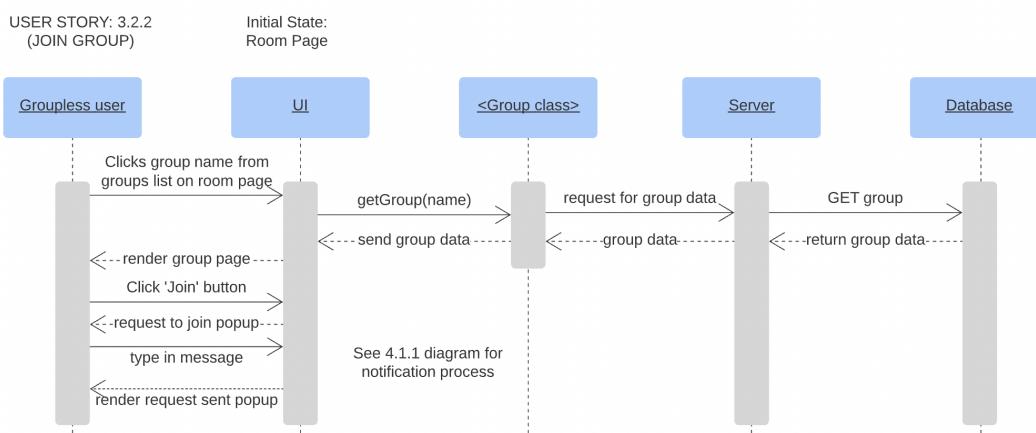
Then I will be taken to their group page

When I click on the 'Join' button

Then a popup will appear, prompting me to type a message

When I type in a message and click 'Join'

Then the group leader will be notified of my request



Requirement 3

As a group member I would like to be able to interact with my group so that I can communicate my intentions.

FEATURE 1: Group members can communicate with each other

As a group member

So that I can discuss any relevant information with my group

I want to be able to chat with the members of said group

SCENARIO: A member wishes to communicate a message with their group

Given I am a member on my group's page

When I click the chat icon in the bottom right

Then I will see a chat area private to my group

When I type in the chat area

And I press enter

Then my group members will see my message

Sequence diagram modelled in problem statement 4 for notification purposes

FEATURE 2: Group members can leave a group

As a group member

So that I can look for other opportunities if I feel my goals no longer align with my group

I want to be able to leave the group

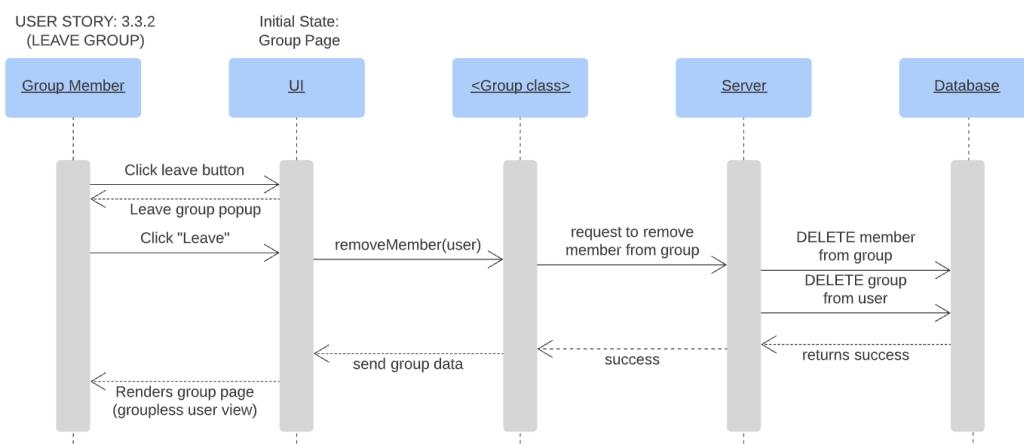
SCENARIO: A group member is unsatisfied with their current group

Given I am on a Group Page that I am a member of

When I click on the 'Leave' button

Then I am removed from the group

And I am now on the Room Page of the room the group was in



Problem Statement 4

Current alternatives in place (such as posting on a course forum) do not have responsive notification systems.

Requirement 1

As a UniCollab group leader, I would like to be notified when a fellow UniCollab user requests to join my group.

Note: All features for this requirement are displayed in one diagram below feature 4

FEATURE 1: Group leaders are notified when a user requests to join their group

As a UniCollab group leader

So that I may stay informed in the group-making process

I want to receive notifications when users request to join my group

Given I am a group leader of a UniCollab group and have received a join request

When I open my notifications inbox

Then I can see the request in my notifications along with the user who sent it and their message

FEATURE 2: Group leaders can accept or decline requests

As a UniCollab group leader

So that I may proceed in forming my ideal group,

I want to be able to accept or decline invitation requests that users have sent.

SCENARIO: The leader of a group has received a request from another student who wishes to join their group.

Given I am a group leader of a UniCollab group and am viewing a request

When I click the 'Accept Request' or 'Decline Request' button to accept or decline

Then my decision will be communicated to the user that sent the request

FEATURE 3: Students are notified that their request is accepted

As a user who sent a request to join a group

So that I know if I have successfully formed a group or need to keep searching

I want to be notified if my request to join a group has been accepted

SCENARIO: A group member is awaiting approval from their request to join a group

Given I have requested to join a group

When the group leader has decided to accept my request

Then a red circle will appear on the notification icon in the top right of the screen
When I click the notification icon
Then I can view the notification
And see that my request was accepted
And be granted full access to this group via my main dashboard page and profile page

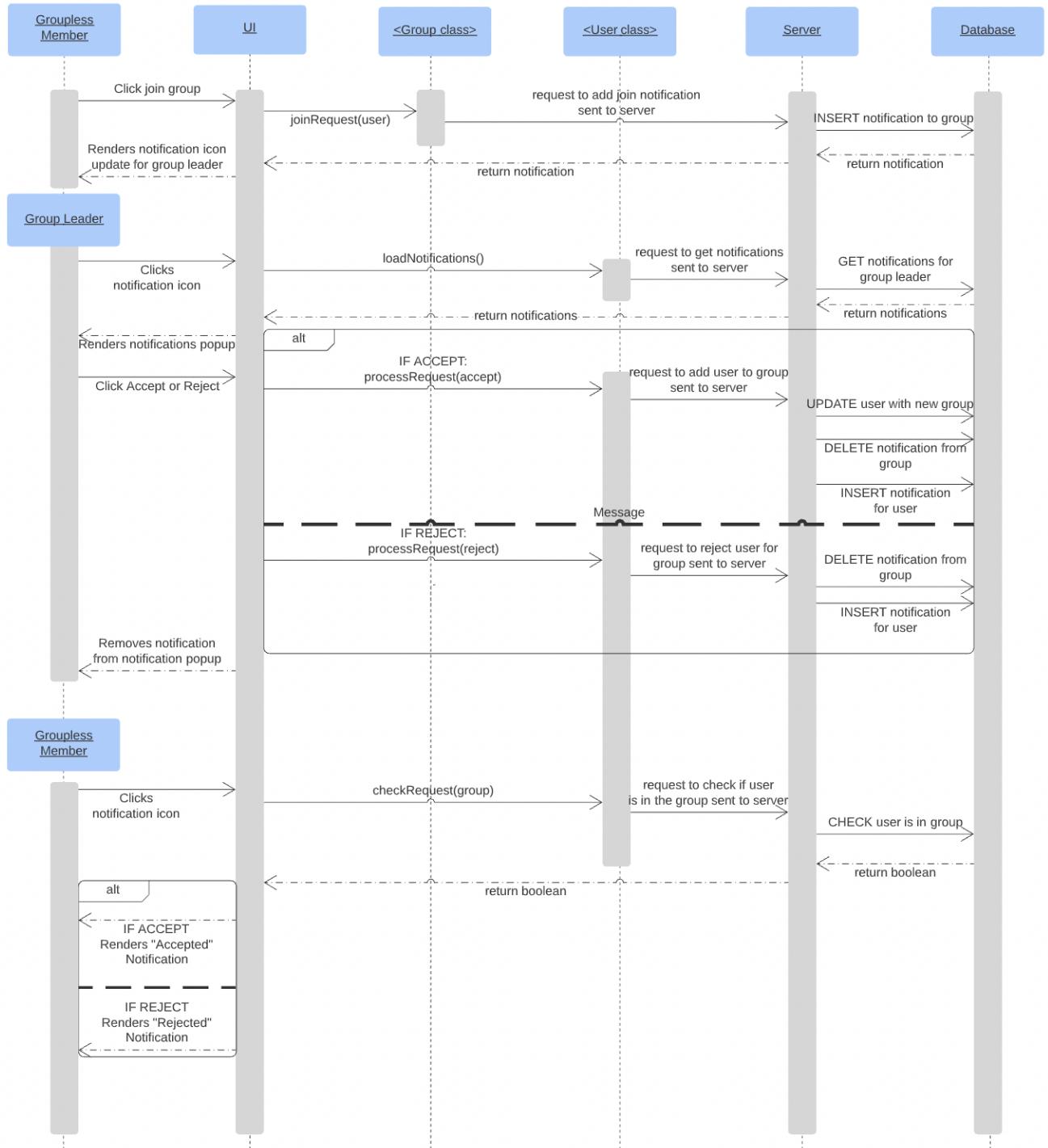
FEATURE 4: Students are notified that their request is declined
As a user who sent a request to join a group
So that I know if I have successfully formed a group or need to keep searching
I want to be notified if my request to join a group has been declined

SCENARIO: A group member is awaiting approval from their request to join a group
Given I have requested to join a group
When the group leader has decided to decline my request
Then a red circle will appear on the notification icon in the top right of the screen
When I click the notification icon
Then I can view the notification
And see that my request was declined

Sequence Diagram is on page below

USER STORY: 4.1.1 &&
4.1.2 && 4.1.3 && 4.1.4
(USER WISHES TO JOIN GROUP)

Initial State:
Groupless Member is on
Group Page. Group Leader is
on any logged in page



Requirement 2 - *again sequence diagram is below all features*

As a groupless UniCollab member, I would like to request to join a group when I see one that is an attractive choice to me.

FEATURE 1: Users receive a notification if a UniCollab group leader has sent an invitation to join their group

As a UniCollab user

So that I can stay up to date with my group making process

I want to be notified when a group leader invites me to join their group

SCENARIO: A UniCollab user receives an invitation to join a group for the room 'COMP1511 Assignment 2'

Given I am a participant of the 'COMP1511 Assignment 2' room (grouped or ungrouped)

When a group leader under this room sends me an invitation to join their room

Then I will be notified of this occurrence

And access the notification via my notifications icon

FEATURE 2: Users can accept or decline invitations sent by group leaders

As a UniCollab user

So that I can progress in the group-making process

I want to be able to accept or decline an invitation to a group I am not currently in

SCENARIO: A user receives a notification to join a group for a room they are active in, and would like to respond to this invitation

Given I am a UniCollab user who has received an invitation to a group

When I click on the group's name or image

Then I am transported to the publically viewable group page

When I click 'Accept'

Then I become a member of the group

When I click 'Decline'

Then I do not become a member of the group

And in either circumstance, the group leader is informed via a notification

FEATURE 3: When a user has responded to an invitation to join a group, the group leader who sent the invitation is notified of the outcome

As a UniCollab group leader

So that I can determine whether users have joined my group

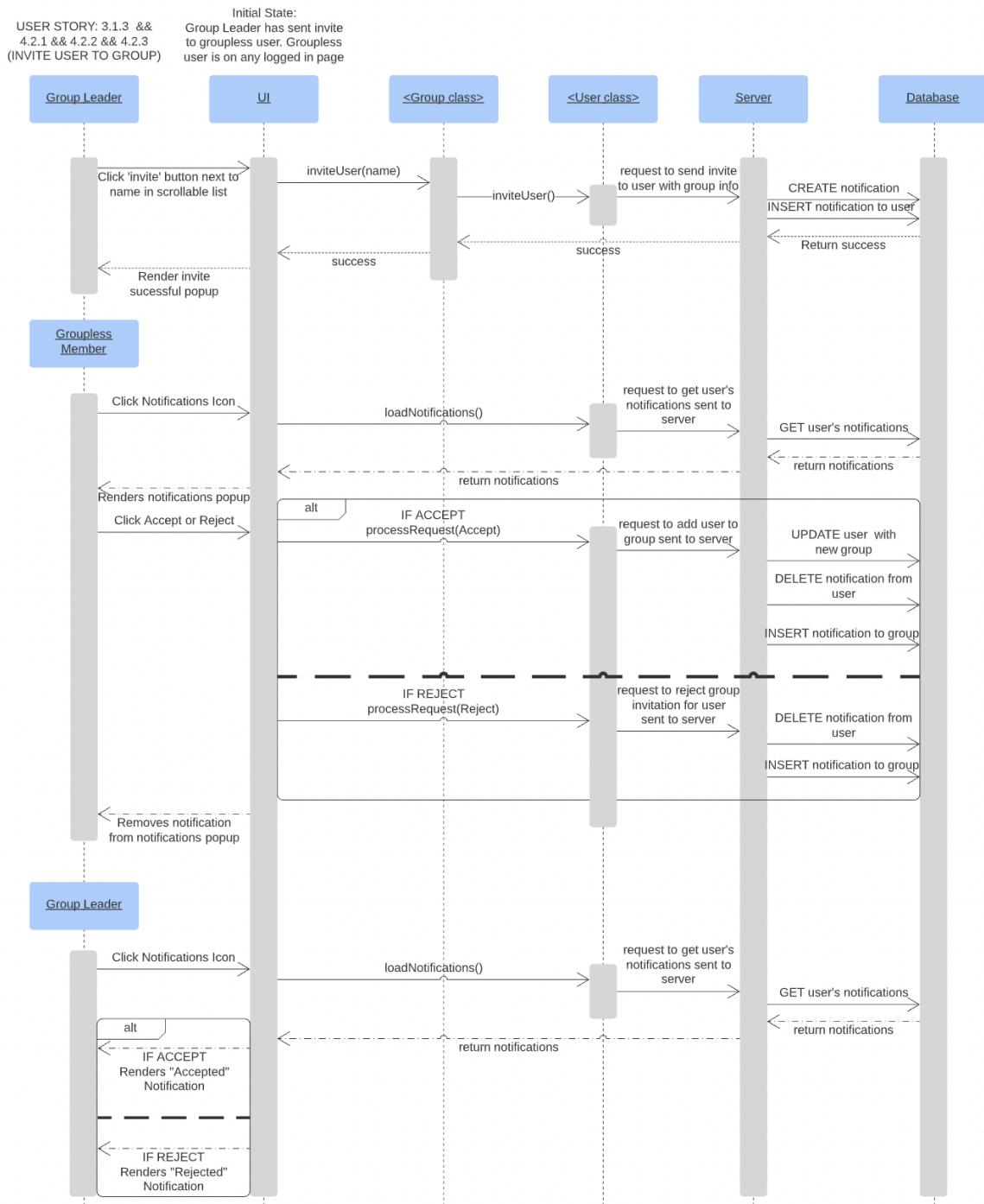
I want to receive a notification regarding the response of my group invitation

SCENARIO: A group leader has sent an invitation to another UniCollab user. This user has decided to decline the invitation.

Given I am a group leader who has sent an invitation to a user

When that user has declined my invitation

Then I will receive a notification that informs me that this invitation to said user has been declined



Requirement 3

As a member of a UniCollab group, I would like to be notified when new messages are sent between members of said group.

FEATURE 1: Students receive notifications when group chats possess messages that a user has not yet viewed

As a UniCollab group member

So that I can stay connected to my group and informed on our project's developments
I want to be notified when chat messages are sent to my group that I have not yet seen

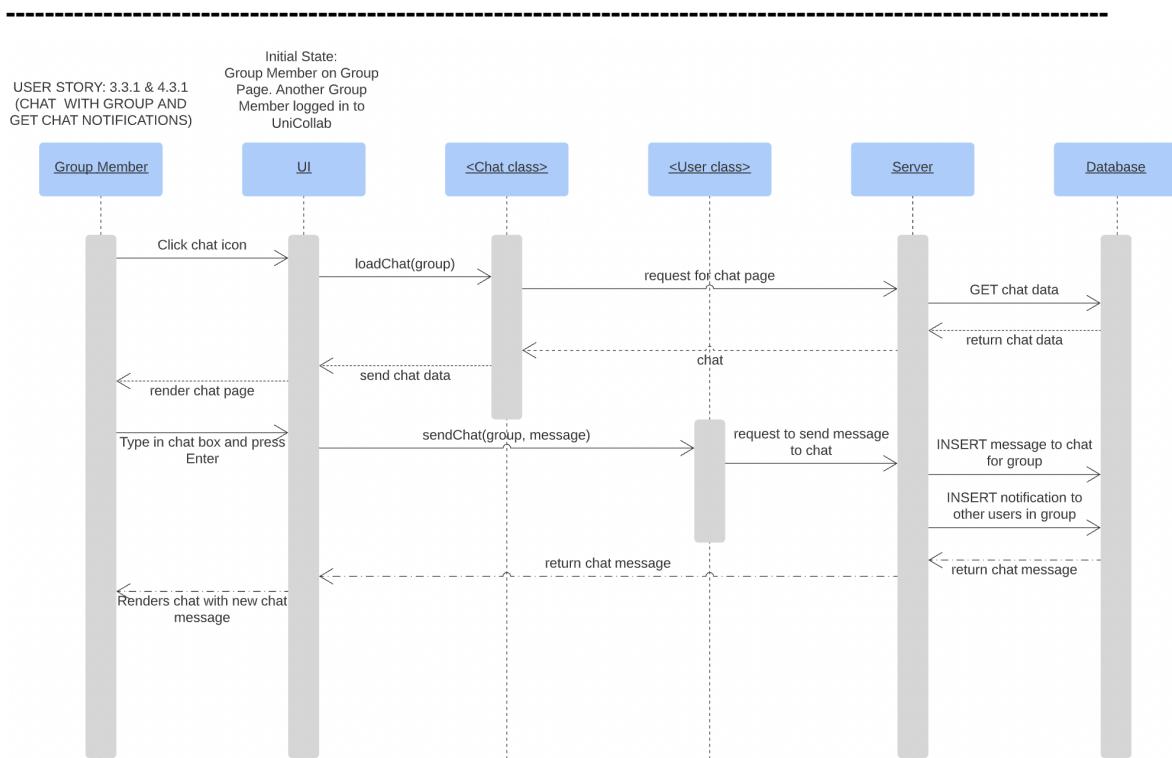
SCENARIO: Anna is in a group for COMP1531 and Bob sends a link for a recorded lecture

Given that Anna and Bob are in the same group for COMP1531

When Bob sends a message containing a URL to the group

Then all group members who are not currently viewing the chat component will receive a notification immediately (if browsing another UniCollab page) or next time they log in (if the group member has logged out of UniCollab)

And said notification will be added to their notification list



Requirement 4

As a UniCollab user, I would like to be able to easily keep track of my notifications and be informed when I receive new ones.

FEATURE 1: A red circle appears above the notification icon for a student who has received an unviewed notification

As a UniCollab user

So that I can stay informed on all updates relevant to me

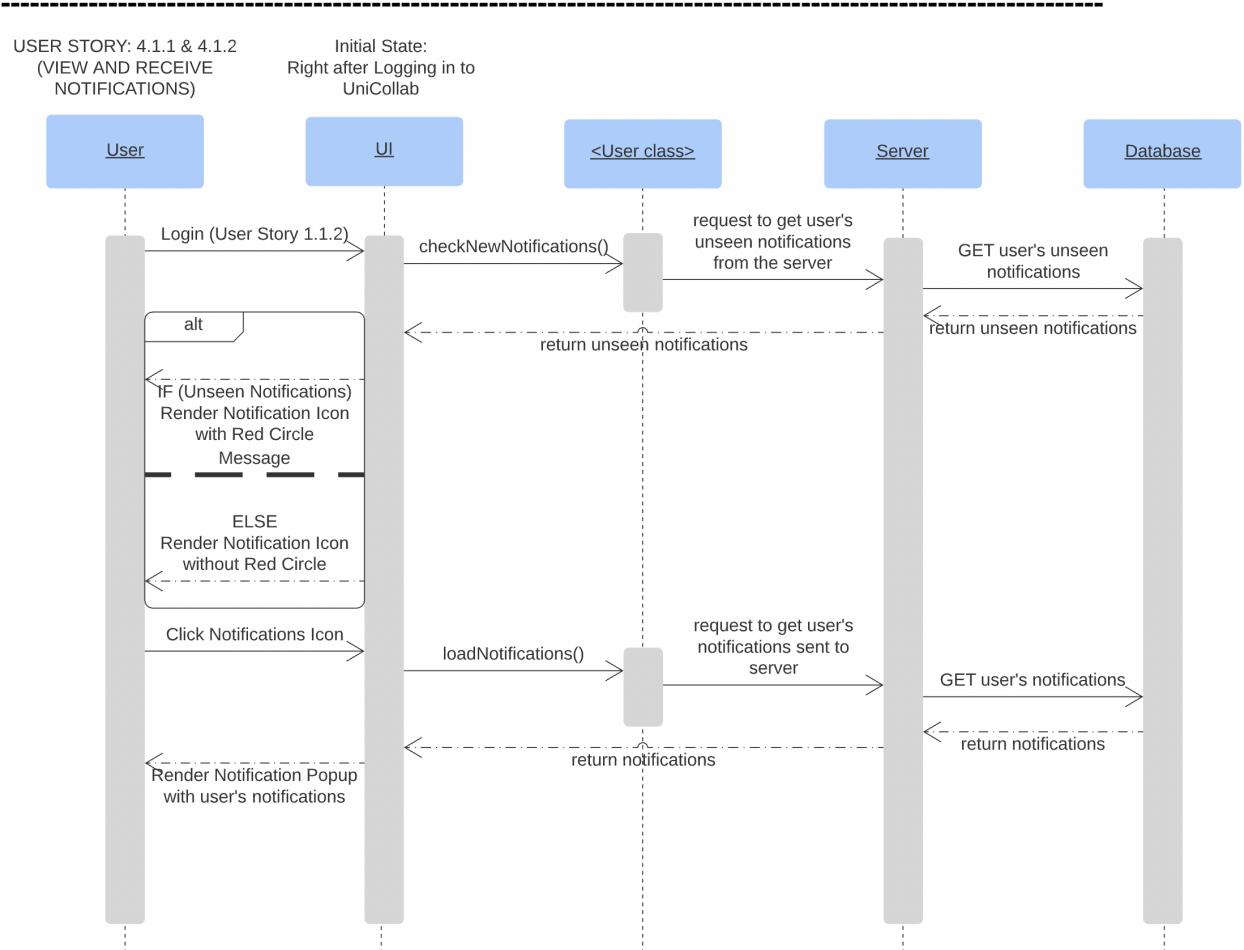
I want to know when I have new notifications that I have not opened

SCENARIO: While a student was logged out, a member of one of their groups sent a message to said group

Given I am a UniCollab user who is part of a group that has been communicating since I was last logged in

When I log into my UniCollab account

Then my notifications icon will possess a small red circle to indicate that I have received unviewed notifications



FEATURE 2: Students may view notifications

As a UniCollab user wanting to browse my notifications

So that I can view all updates I have missed

I want to be able to access a list of notifications

SCENARIO: A student receives a notification and would like to view it

Given I am a user of UniCollab and have received an notification

When I click on the notification icon to the top right beneath my profile box

Then I will view a list of all my notifications

When I press an individual notification

Then it expands to show the full message (if the message was cut off)

DATABASE MODELLING:

Attached as separate pdfs