SC2006 - Software Engineering Project

Software Requirements Specification

Final Documentation (Lab #5 Deliverables)

Lab Group	SCSC	
Team	UpGrade (Group 4)	
Members	Shannon Wong Tze Shian (Huang Zhixuan)	U2421914B
	Yong Ming Xuan	U2420697J
	Joshua Tan Ying Jie	U2422477C
	Garlapati Shreeya	U2421092J
	Chloe Atipinati Djohan	U2423255A
	Aurellia Maharani Putri	U2423175L

1. Project mission statement 4
2. Functional Requirements 4
I. User Registration 4
II. Admin Moderation 5
III. Profile Management 5
IV. Homepage 6
V. Study Group Management 6
VI. Invitation Inbox 8
VII. Chat Management 8
VIII. Scheduling and Reminders 9
IX. Map and Locations 10
X. Interface Navigation 10
3. Non-functional requirements 10
4. Data dictionary 11
5. UI Mockups 12
6. Complete Use Case Model 17
I. Use Case Diagram 17
II. Use Case Descriptions 17
I. Account Management 17
II. Profile Management 22
III. Discovery and Search 24
IV. Study Group Creation and Membership 31
V. Invitation Inbox 36
VI. Scheduling and Conflict Resolution 38
VII. Map and Study Locations 44
VIII. Report Group and Manage Reports 47
IX. Private Chats 50
7. Design Model 53
I. Class Diagram 53
II. Sequence Diagrams 54
UC-03: Login 54
UC-04: ResetPassword 54
UC-06: BrowseProfile 55
UC-08: BrowseStudyGroup 55
UC-11: CreateStudyGroup 56
UC-12: CloseStudyGroup 56
UC-13: JoinStudyGroup 57
UC-14: InviteUser 57
UC-17: Accept/RejectInvite 58
UC-18a: NotifyUsers (accept/reject) 58
UC-18a: NotifyUsers (leave group) 59

UC-20: ResolveConflict....59
UC-21: ReminderNotificationDelivery....60
UC-24a: CloseExpiredGroups (close early)....60
UC-24b: CloseExpiredGroups (expired)....61
UC-27: SelectStudyLocation....61
III. Dialog Map....62
8. System Architecture....63
I. Presentation Layer....63
II. App Logic Layer....64
III. App Object Layer....65
IV. Persistent Data Layer....66
9. Application Skeleton....66
I. Frontend....66
II. Backend....66
10. Test Cases and Testing Results....67
I. Black Box Testing....67
a. AccountController (chosen class)....67
b. Equivalence Class and Boundary Value Testing....68
c. Test Cases and Results....68
II. White Box Testing....73
c. Edit Profile (chosen function)....73
I. Flow Graph....73
II. Basis Path Testing....73
III. Test Cases and Results....74
d. Create Group....75
I. Flow Graph....75
II. Basis Path Testing....76
III. Test Cases and Results....76
11. Website Demonstration Video....77
12. Live Presentation Script....77
13. Presentation Slides....82
14. WebApp UI....82

1. Project mission statement

Our team will develop a website (UpGrade) that connects students with like-minded, diligent peers to form effective study groups. Student users benefit directly from improved academic support, and our developers contribute to Singapore's Smart Nation vision by creating digital solutions that enhance education and community-building. This project offers a secure and user-friendly platform that encourages knowledge sharing and builds supportive study communities for students.

2. Functional Requirements

I. User Registration

1. UpGrade shall allow Users to be authenticated and register to join the platform.
    - 1.1. UpGrade shall allow each User to create an account using a valid school email address (ending in .edu.sg).
        - 1.1.1. UpGrade shall verify the provided school email by matching it against the School's Database to activate their account.
    - 1.2. UpGrade shall allow each User to create a unique username and password for an account.
        - 1.2.1. UpGrade shall verify that the username is unique
        - 1.2.2. UpGrade shall ensure that the password is valid (contain at least 12 characters, at least 1 uppercase letter, at least 1 lowercase letter, at least 1 number, at least 1 special character).
    - 1.3. UpGrade shall send a code to the registered school email as part of the account activation process upon successfully creating a username and password.
        - 1.3.1. The User will be required to enter the code to activate their account.
        - 1.3.2. UpGrade shall prompt the User to retype the code if the User enters an incorrect code.
        - 1.3.3. UpGrade shall send a new code to the registered school email when the User selects the "Resend Code" option.
2. UpGrade shall allow User to login with their account username and password
    - 2.1. UpGrade shall provide a password-reset function. UpGrade shall send a reset code to the User's registered school email address when the User clicks the "Forgot Password" button.
    - 2.2. UpGrade shall not activate the User's account until the User is verified using the code sent to the User's registered school email address and User also has to fill up all the mandatory fields such as Email Address, Gender, Year of study, Education level, Preferred Study Location(s), Preferred Study Timing, Current Study Course/subjects for account activation.

II. Admin Moderation

1. UpGrade shall allow Admins to perform admin-related tasks
    - 1.1. UpGrade shall allow Admins to review reported groups.
        - 1.1.1. UpGrade shall display a list of review reports by users on any study groups and admin shall inspect the group details: Group Name, Group Description and Self-written tags.
        - 1.1.2. Admins shall either choose to “Dismiss” or “Remove” the study group
    - 1.2. UpGrade shall Admins to remove a study group by selecting the “Remove” option
        - 1.2.1. UpGrade shall prompt Admin “Confirm” to remove the group
        - 1.2.2. UpGrade shall follow the procedure of closing a group
            - 1.2.2.1. Study group shall not be visible to Users
            - 1.2.2.2. Invitations to the specific study group shall be cancelled
            - 1.2.2.3. Study Group members shall be removed from the group and their schedules shall be cleared of the specific study group
        - 1.2.3. If the Host of the specific study group does not have a warning, UpGrade shall give the Host a warning.
        - 1.2.4. If the Host of specific study group have a prior warning, UpGrade shall allow Admin to “Ban” the Host
        - 1.2.5. If the Study group is expired, UpGrade shall follow the group closure procedure and give Admins an option to Warn or Ban the Host.
    - 1.3. UpGrade shall allow Admins to dismiss a study group report by selecting the "Dismiss" option.
        - 1.3.1. UpGrade shall prompt the Admin for confirmation to dismiss the report
        - 1.3.2. UpGrade shall list the report as “False Report”
        - 1.3.3. UpGrade shall give Admins an option to Warn or Ban the User.
            - 1.3.3.1. If the User who submitted the false report does not have a warning, Upgrade shall give the User a warning.
            - 1.3.3.2. If the User who submitted the false report has a prior warning, UpGrade shall allow Admin to “Ban” the User.

III. Profile Management

1. UpGrade shall allow Users to update personal information on their profiles.
    - 1.1. UpGrade shall allow Mandatory fields as well as Optional fields to be filled up in the Profile.
        - 1.1.1. If the User does not fill up the Mandatory fields, UpGrade shall not allow the User to save.
    - 1.2. UpGrade shall allow the User to make edits to their profile such as, their Year of Study, Preferred Study Location(s), Study Timing, Current Study Course / Subjects (Mandatory).
    - 1.3. UpGrade shall allow User to input their School/Institution Name, their Academic Grades and/or their Grade Point Average (GPA), or Usual Study Duration. (Optional).
2. UpGrade shall allow Users to view the profiles of other Users and their own.
    - 2.1. UpGrade shall allow Users to view the Mandatory fields of each student profile.

2.2. UpGrade shall only allow Users to view the profiles of other Users of the same educational level (Secondary / Junior-College / Polytechnic / University)

3. UpGrade shall allow Users to click the log out button under Profile, in order to log out of the website.

IV. Homepage

1. UpGrade shall display on the homepage a list of profiles of all registered Users of the same education level.
   1.1. UpGrade shall allow Users to filter the displayed profiles based on profile attributes (such as Year of Study, Preferred Study Duration) to narrow the list to those matching the filter criteria.
   1.2. UpGrade shall provide a search feature where a User can look up a specific other User by their unique username or name. This allows the User to find and invite them to a study group.

V. Study Group Management

1. UpGrade shall support the creation and management of study groups, each identified by a unique groupID. Any User may act as a sender to invite others or as a host to create a new study group.
   1.1. For private invitations: When a User clicks on the “+ Invite” button attached to the bottom of another User’s profile, UpGrade shall allow the sender to choose to invite the receiver to an existing study group the User has joined. UpGrade also allows User to create a new private study group under the “Study Group” page’s “Created groups” tab and set visibility to “private”.
      1.1.1. If creating a new private study group, the sender must specify the Group Name, Meeting Date, Time (starting and ending), Location and maximum number of participants. The sender can also specify Focus Subject(s), an optional field, if they wish to focus on a particular subject for their study group.
      1.1.2. When the sender User sends an invite, UpGrade shall place it in the receiver User’s invitation inbox with the group details.
      1.1.3. If a private study group reaches its maximum participant quota, UpGrade shall automatically remove any pending invitations for that group and close the group from further joining.
         1.1.3.1. If a User leaves a previously full study group, UpGrade shall allow other currently joined Users to invite new Users to join the study group again.
         1.1.3.2. If a User leaves a previously full study group, UpGrade shall decrement the group’s current participant count.
      1.1.4. Any User who has joined a private study group shall be able to invite additional Users to that group.
      1.1.5. Each time a User accepts and joins the group, UpGrade shall increment the group’s current participant count.

1.2. For public study groups: UpGrade shall allow any User to create a public study group and set visibility to “public”. Such a group shall be listed on the “Study Group” page’s “Created groups” tab and visible to all students.

1.2.1. If creating a new public study group, the host must specify the Group Name, Meeting Date, Time (starting and ending), Location and maximum number of participants. The host can also specify Focus Subject(s), an optional field, if they wish to focus on a particular subject for their study group.

1.2.2. If a public study group reaches its maximum participant quota, UpGrade shall automatically close the group. The study group shall no longer appear on the StudyGroup page but remain recorded in the schedules of the joined Users.

1.2.2.1. If a User leaves a previously full study group, UpGrade shall display the group on the “Study Group” page again and reopen the group.

1.2.2.2. If a User leaves a previously full study group, UpGrade shall decrement the group’s current participant count.

1.2.3. The host User of a public study group they created shall have the option to close the group early.

1.2.3.1. If the host closes the group before it is full, UpGrade shall remove the group from the “Study Group” page immediately.

1.2.4. Any User who has joined a public study group shall be able to invite additional Users to that group through their profiles on the homepage.

1.2.5. Each time a User accepts and joins the group, UpGrade shall increment the group’s current participant count.

1.2.6. Study Groups created by the host User will be displayed in a separated section at the top of the “Study Group” page to ensure quick access by hosts. UpGrade shall allow host Users to edit details of their study group before its expiration or the early closure of the study group by the host.

1.3. UpGrade shall allow Users to leave a study group at any time before expiration of the study group.

1.4. UpGrade automatically generates a random and unique groupID when a new study group is created. Unique groupID allows Users to confirm their study group members upon meeting up as every study group has their own specific groupID.

1.4.1. UpGrade shall ensure unique groupIDs are generated and persistent until deletion of the study group.

2. Upgrade shall allow Users to report a study group if its group details are inappropriate.

2.1. Group details include Group Name, Group Description and Self-written tags. Users shall be able to select one or more violation categories when submitting a report.

2.1.1. Violation categories shall include: 1.Vulgarities; 2. Sexual content; 3. Harassment or bullying; 4. Harmful or dangerous acts; 5. Spam or misleading content; 6. Misinformation

2.2. Upgrade shall store the report with the associated StudyGroupID, host username, reporting user username, relevant group details and selected violation categories.
2.3. UpGrade shall make the report accessible to Admins for review and action.

VI. Invitation Inbox

1. UpGrade shall have an inbox system for Users.
    1.1. The inbox system shall list all invitations that the user has received/applied for.
        1.1.1. The User shall be able to accept or reject the invitation.
    1.2. Each of the invitations shall show the Date, Time (starting and ending), Location and the current participant count of the group at the time of viewing.
    1.3. The invitations of the inbox shall also display the usernames of the Users that have joined.
    1.4. The invited User shall be able to view the Users' profiles by clicking onto the username of each User.
    1.5. UpGrade shall notify Users about membership changes in joined study groups.
        1.5.1. UpGrade shall send a notification to the inbox of every User in study group when a new User joined the study group
        1.5.2. UpGrade shall send a notification to the inbox of every User in study group when a User leave the study group
        1.5.3. UpGrade shall send a notification to the inbox of the sender User when the invited User rejects an invitation to a study group.
    1.6. UpGrade shall send a reminder notification to the inbox 24 hours before the meet time of a study group that the User has joined.

VII. Chat Management

1. UpGrade shall allow Users to message other individual Users.
    1.1. The User shall be able to start a one-to-one private chat with the Host.
        1.1.1. The User shall be able to send a message to start a new chat with the host when viewing group details of the invitation to that group via the inbox page.
        1.1.2. The User shall be able to send a message to start a new chat with the host when viewing group details to a specific public group via the study group page.
    1.2. The Host shall be able to start a one-to-one private chat with any one group member of the group he created.
        1.2.1. The Host shall be able to send a message to start a new chat with a group member of the group he created when viewing group details of his created group via the study group page.
            1.2.1.1. The Host is able to view members who joined their private groups under the created group tab in study group page
    1.3. The User shall be able to start a one-to-one private chat with any one user of the same educational level.

1.3.1. The User shall be able to send a message to start a new chat with another User when viewing profile details of a specific other user via the homepage.

1.4. The User shall be directed to the same chat that has been previously created if there already exists a chat between the User and another User, instead of creating a new one-to-one chat.

2. UpGrade shall have a chat page for Users.
   2.1. The chat page shall list all one-to-one chat conversations that the user has with every other user they have previously messaged to or received a message from.

VIII. Scheduling and Reminders

1. UpGrade shall handle scheduling constraints.
   1.1. UpGrade shall prevent scheduling conflicts. If a User has already joined a study group that overlaps with the study time and duration of a pending invite to another study group, UpGrade will not allow the User to accept the conflicting invitation. UpGrade will prompt the User to choose whether to leave the joined study group before accepting the new invitation.
2. UpGrade shall send reminders for confirmed study groups.
   2.1. For each study group that the User has joined, UpGrade shall send a reminder notification (via email, in-website alert and in Inbox) 24 hours before the scheduled meeting time.
   2.2. UpGrade shall allow Users to enable or disable email notifications for study group reminders via their Profile settings. In-app notifications and inbox reminders will remain unaffected.
3. UpGrade shall handle the expiration of study groups.
   3.1. UpGrade shall automatically expire the event at the scheduled start time of a study group.
      3.1.1. If the study group is listed publicly in the StudyGroup page, UpGrade removes it from the public list.
   3.2. UpGrade shall delete any remaining pending invitations for the expired study group from the invitation inboxes of all invited Users.
4. UpGrade shall display the study groups joined by a User in calendar format.
   4.1. Each study group displayed in calendar format shall show the Group Name and Location details in the correct time slot.
   4.2. UpGrade shall allow Users to click on “view details” option in each calendar entry to view full details of that study group, which includes Group Name, Date, Time (starting and ending) Location, current participants count and participant profiles, as well as its unique groupID. Users shall also view the “Leave Group” option under “view details”, where Users can manually choose to leave a specific study group.
   4.3. UpGrade shall display the study group in the schedule page and automatically delete it 72 hours after the end time of the study group.

4.3.1. UpGrade shall allow Users to sync their schedule view after auto-deletions.

IX. Map and Locations

1. UpGrade shall include a map for Hosts to select a Study location
   1.1. UpGrade shall display common study locations on the map for Hosts to select from.
      1.1.1. UpGrade shall display Study Locations as selectable markers on the map
      1.1.2. UpGrade shall group Study Locations markers by regions
   1.2. UpGrade shall provide a list of available Study Locations when creating a study group.
2. UpGrade shall include a map for Users view Study locations
   2.1. UpGrade shall display Study Locations as markers on the map

X. Interface Navigation

1. UpGrade shall allow Users to access and navigate to the system's primary functions though a Navigation/ Menu Panel.
   1.1. If the "Homepage" option is selected, UpGrade shall navigate the User to the Homepage, where they can view, filter and search user profiles.
   1.2. If the "Study Groups" option is selected, UpGrade shall navigate the User to StudyGroup page where they can view, filter, search through public study groups in the "All groups" tab and all (public and private) study groups they created in the "Created groups" tab.
   1.3. If the "Inbox" option is selected, UpGrade shall navigate the User to their inbox where they can view, accept or reject invitations
   1.4. If the "Schedule" option is selected, UpGrade shall navigate the User to a calendar view of all their currently active study group and its details.
   1.5. If the "Location Map" option is selected, UpGrade shall navigate the User an interactive map to view the geographical locations of their active study groups and other Study locations
   1.6. If the "Profile" option is selected, UpGrade shall navigate the User to their own profile page where they can view or update the Mandatory and optional fields.
   1.7. If the "Chats" option is selected, UpGrade shall navigate the User to their own chat page, where they can view all one-to-one private chats with other users.

3. Non-functional requirements

Usability	- The interface should be accessible on desktop - The platform shall be navigable with no more than 3 clicks to access primary functions

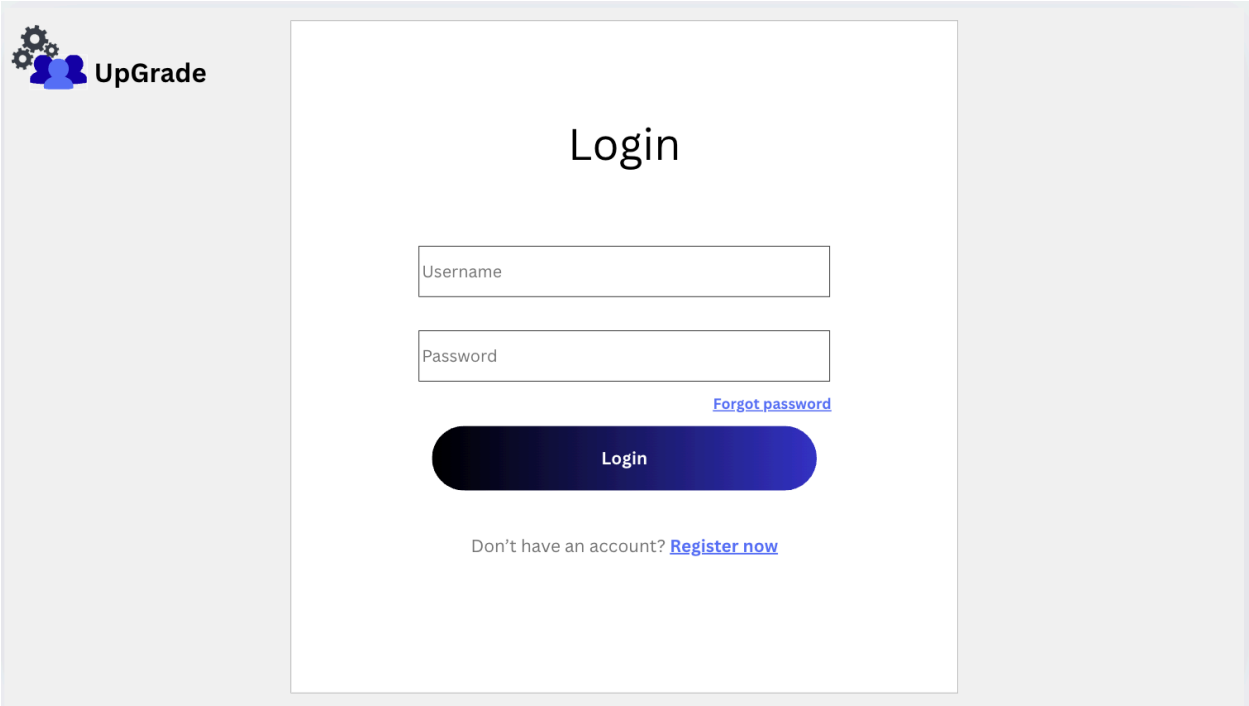
Reliability	- After rebooting/restarting the website, all functionalities must be restored within 5 seconds.
Performance	- After the user login, System should efficiently retrieve and display data to users - The website shall load any page within 3 seconds under normal conditions - System shall automatically delete expired student group records 72 hours after endtime, to minimize database load and improve performance
Maintainability	- The code shall be written in a modular way for easier addition of future features - System shall implement automated data cleanup to reduce database load
Security	- A user must log in with their school email to register. - When user input their password, it should be hidden - Only authenticated users can view the website

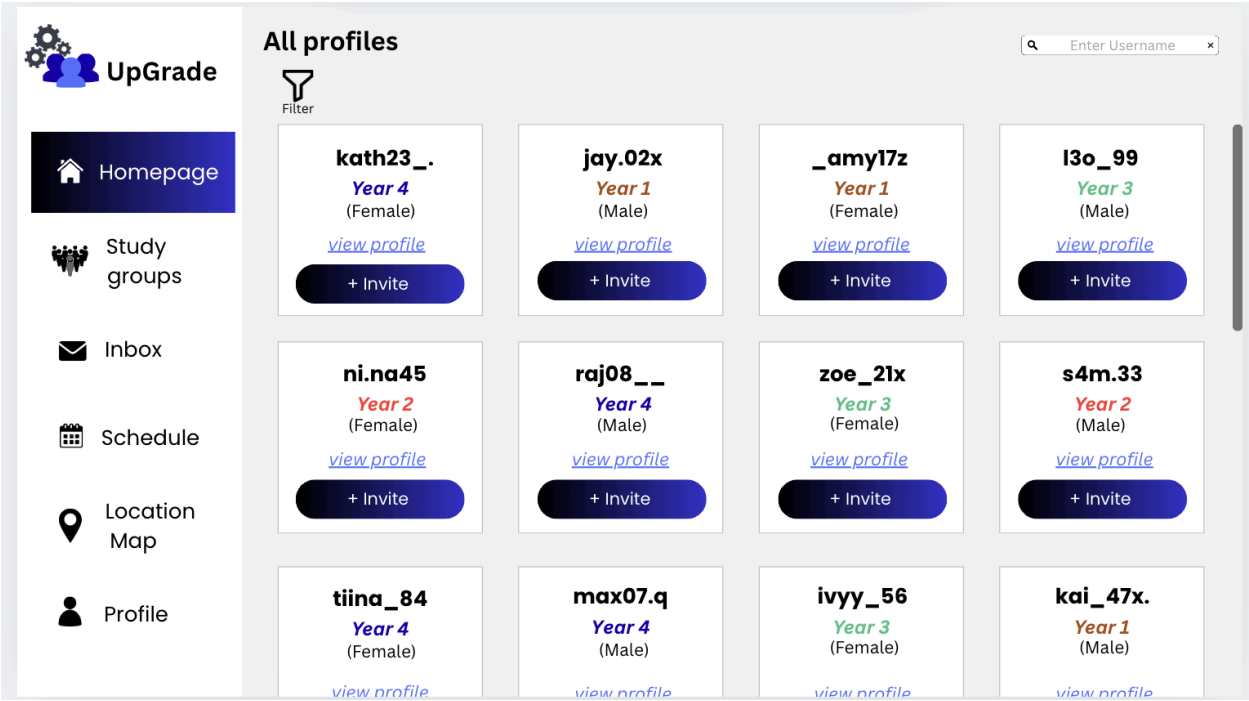
4. Data dictionary

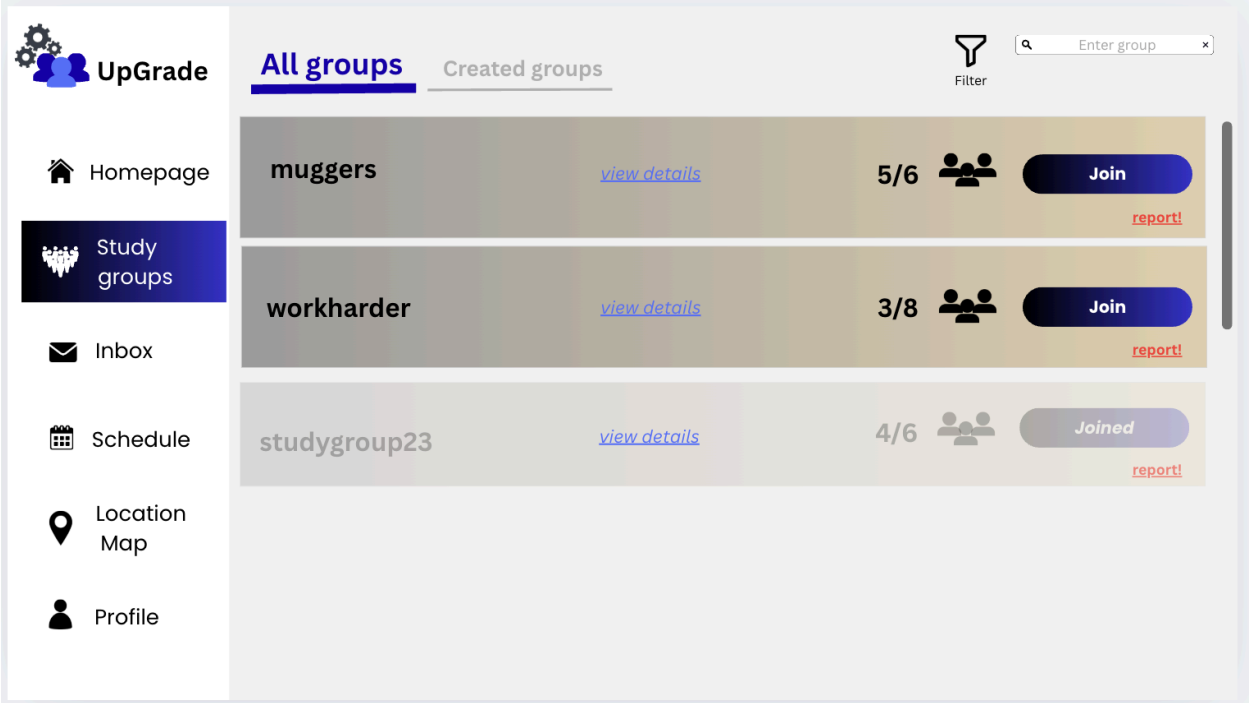
Term	Definition
User	The person interacting with UpGrade platform after creating an account
Host	A user that is in charge of a study group that they themselves have created. They are able to update meeting date, time (starting and ending), Location and maximum number of participants.
Study Group	A group of Users that have chosen to collaborate and study together on a common academic topic
School's Database	List of currently active student emails from individual schools for verification of authenticity of email used to register for the website.
Mandatory fields	Gender, Year of Study, Preferred Study Location(s), Preferred Study Duration, Study Timing, Current Study Course/ Subjects; will be displayed publicly on each User's profile for viewing on homepage; shall be used for filtering of profile list on homepage.
Optional fields	School Name, Academic Grades, Grade Point Average; shall be used for filtering of profile list on homepage.
groupID	Unique alphanumeric string of 9 characters long that is specific to a

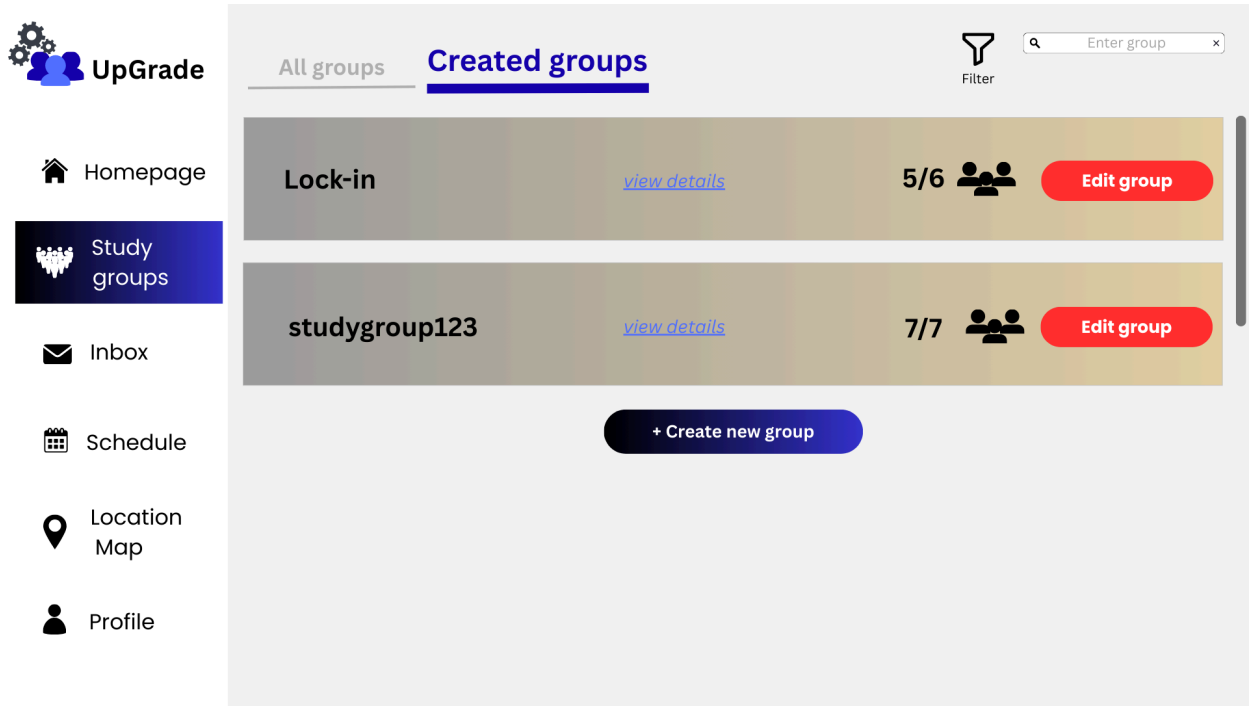
	particular private or public group created; enables study groups to be differentiated when there is an overlap in group names; helps users identify their study mates accurately when meeting in person.
Invitation	A request from a User who is from a private study group to another user to join the group.
Map	An interactive map using Google Maps API that displays study locations categorized by geographic region to assist users in selecting a preferred location.
Study Locations	Common places for studying such as campus libraries, cafes, university study areas.
expire/expiration of study group	The start time of the study group has been reached or passed. The study group is no longer displayed in the StudyGroup page (if public), the host can no longer edit details, and participants can no longer leave the group. Pending invitations are deleted from invited Users' inboxes.

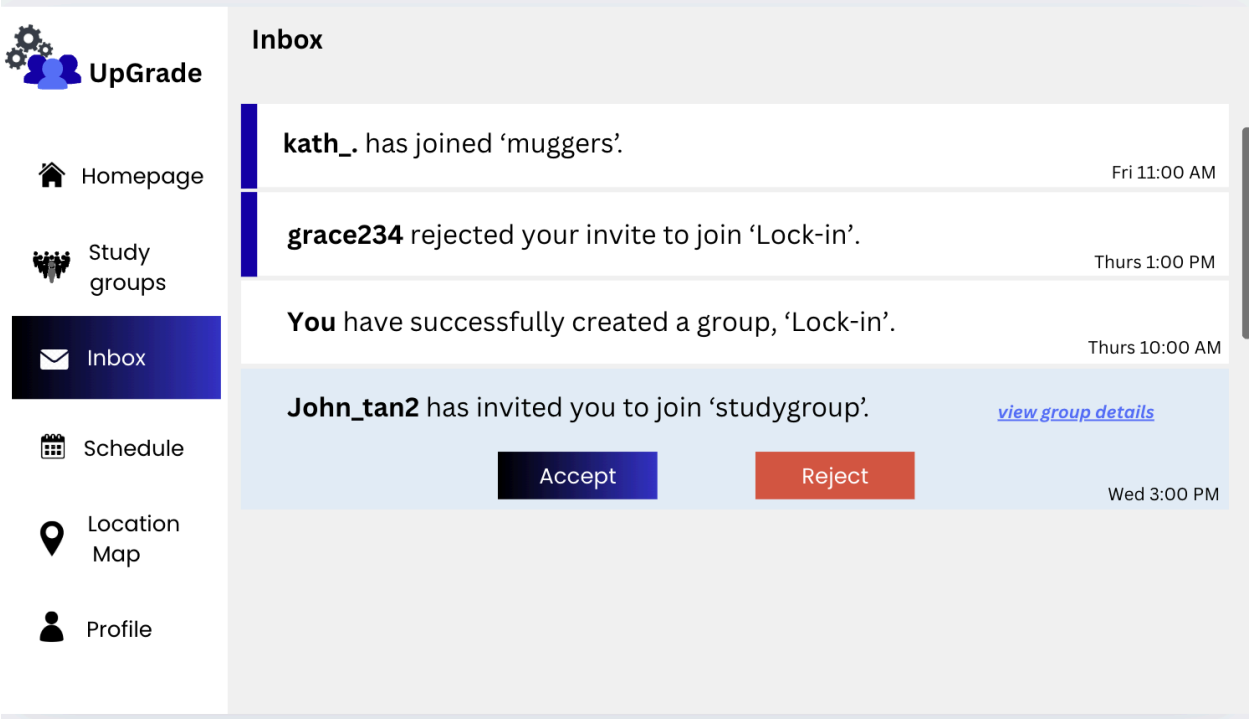
5. UI Mockups

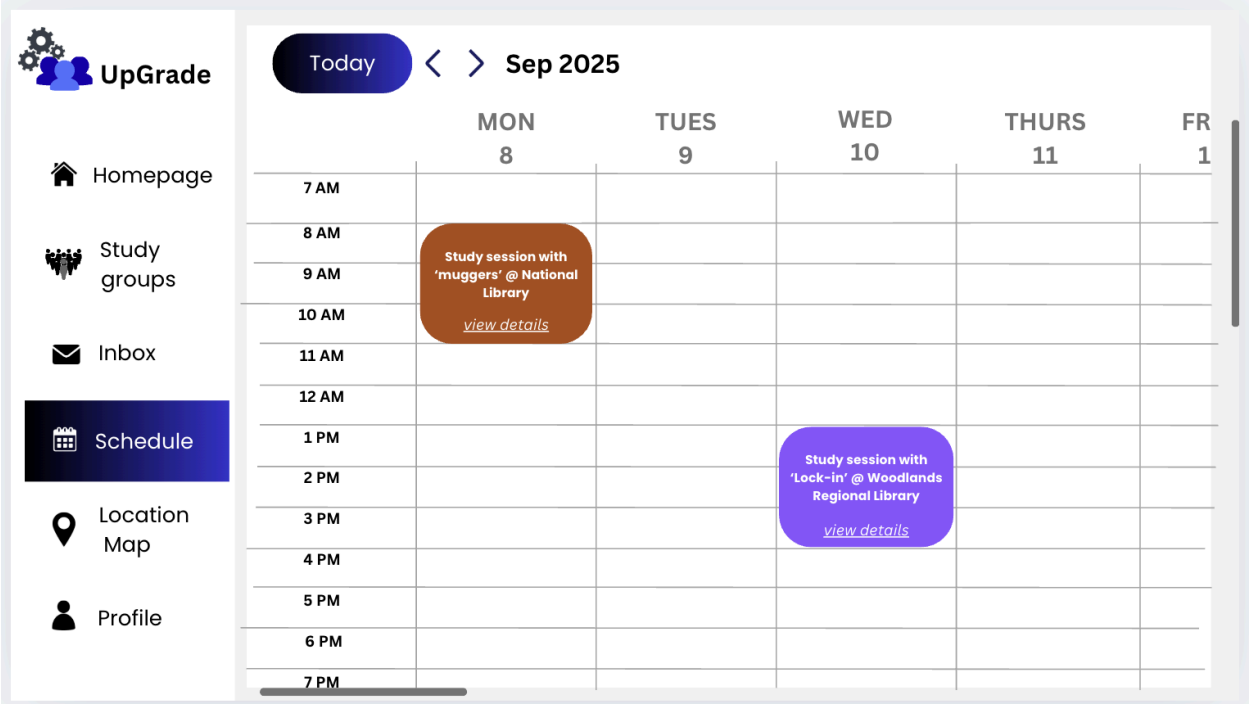


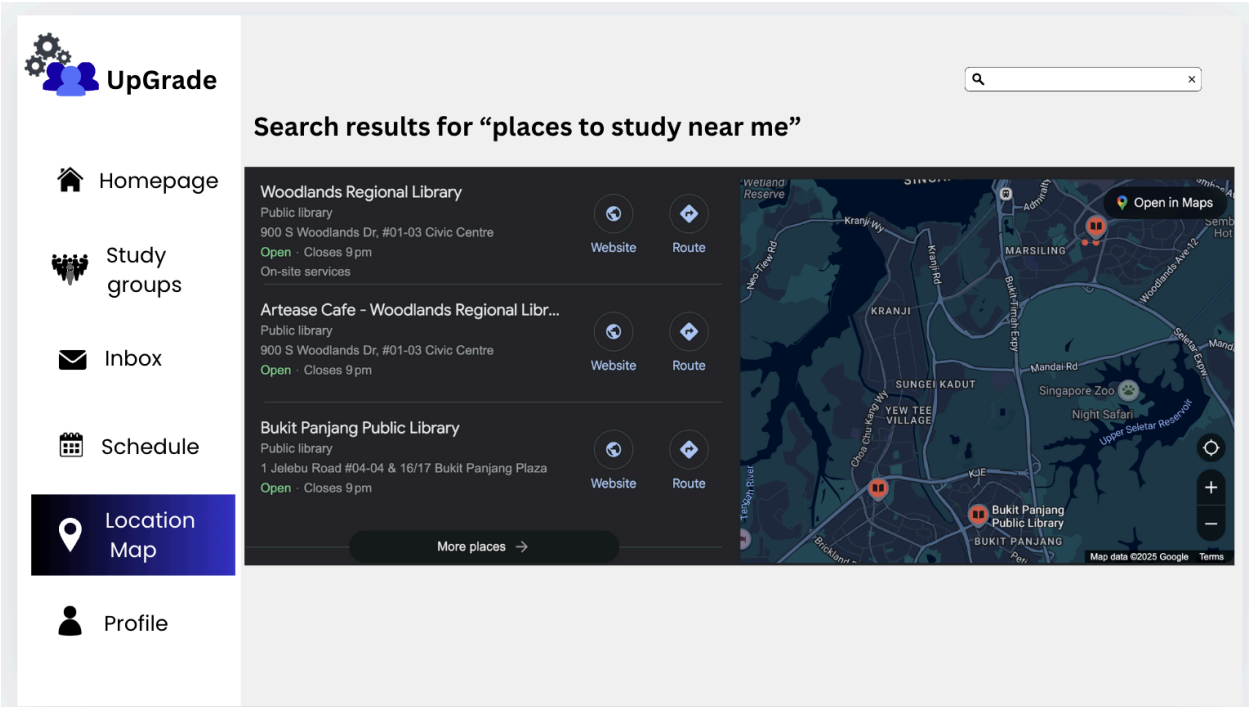

UpGrade
Homepage
Study groups
Inbox
Schedule
Location Map
Profile
All profiles
Filter
Enter Username
kath23_.
Year 4
(Female)
view profile
+ Invite
jay.02x
Year 1
(Male)
view profile
+ Invite
_amy17z
Year 1
(Female)
view profile
+ Invite
l3o_99
Year 3
(Male)
view profile
+ Invite
ni.na45
Year 2
(Female)
view profile
+ Invite
raj08__
Year 4
(Male)
view profile
+ Invite
zoe_21x
Year 3
(Female)
view profile
+ Invite
s4m.33
Year 2
(Male)
view profile
+ Invite
tiina_84
Year 4
(Female)
max07.q
Year 4
(Male)
ivyy_56
Year 3
(Female)
kai_47x.
Year 1
(Male)

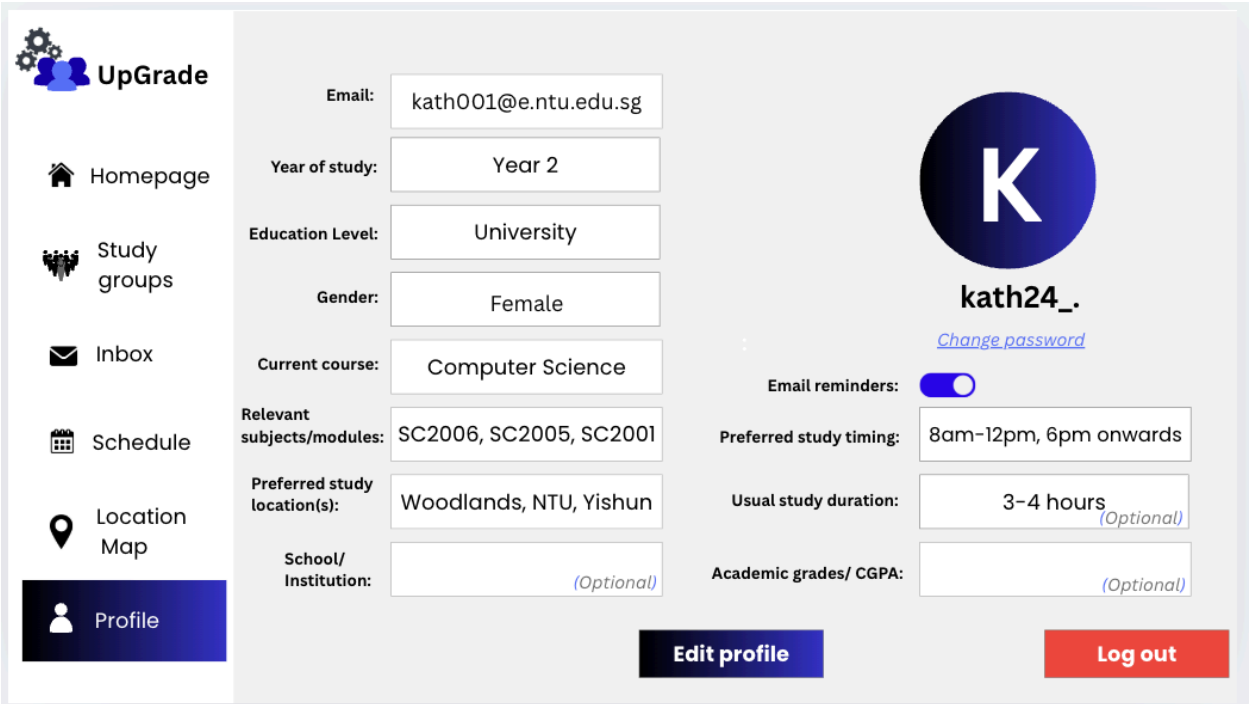

UpGrade
Homepage
Study groups
Inbox
Schedule
Location Map
Profile
All groups
Created groups
Filter
Enter group
muggers
view details
5/6
Join
report!
workharder
view details
3/8
Join
report!
studygroup23
view details
4/6
Joined
report!


UpGrade
Homepage
Study groups
Inbox
Schedule
Location Map
Profile
All groups
Created groups
Filter
Enter group
Lock-in
view details
5/6
Edit group
studygroup123
view details
7/7
Edit group
+ Create new group


UpGrade
Homepage
Study groups
Inbox
Schedule
Location Map
Profile
Inbox
kath_. has joined 'muggers'.
Fri 11:00 AM
grace234 rejected your invite to join 'Lock-in'.
Thurs 1:00 PM
You have successfully created a group, 'Lock-in'.
Thurs 10:00 AM
John_tan2 has invited you to join 'studygroup'.
view group details
Accept
Reject
Wed 3:00 PM

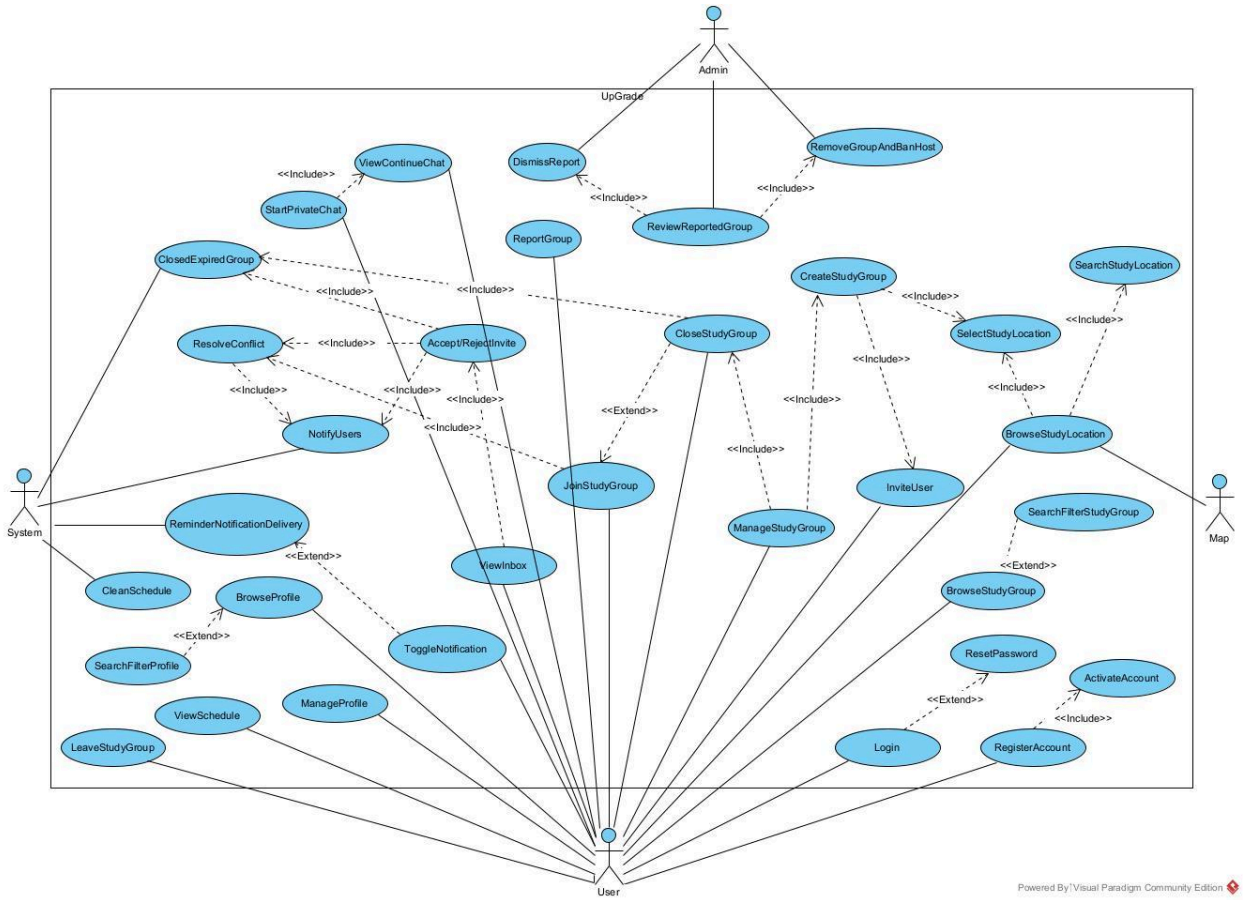

UpGrade
Homepage
Study groups
Inbox
Schedule
Location Map
Profile
Today
Sep 2025
MON 8
TUES 9
WED 10
THURS 11
FR 1
7 AM
8 AM
9 AM
10 AM
11 AM
12 AM
1 PM
2 PM
3 PM
4 PM
5 PM
6 PM
7 PM
Study session with 'muggers' @ National Library
view details
Study session with 'Lock-in' @ Woodlands Regional Library
view details


UpGrade
Homepage
Study groups
Inbox
Schedule
Location Map
Profile
Search results for "places to study near me"
Woodlands Regional Library
Public library
900 S Woodlands Dr, #01-03 Civic Centre
Open · Closes 9 pm
On-site services
Website
Route
Artease Cafe - Woodlands Regional Libr...
Public library
900 S Woodlands Dr, #01-03 Civic Centre
Open · Closes 9 pm
Website
Route
Bukit Panjang Public Library
Public library
1 Jelebu Road #04-04 & 16/17 Bukit Panjang Plaza
Open · Closes 9 pm
Website
Route
More places
Open in Maps
MARSILING
KRANJI
SUNGEI KADUT
YEW TEE VILLAGE
Singapore Zoo
Night Safari
Upper Seletar Reservoir
Bukit Panjang Public Library
BUKIT PANJANG
Map data ©2025 Google
Terms


UpGrade
Homepage
Study groups
Inbox
Schedule
Location Map
Profile
Email:
kath001@e.ntu.edu.sg
Year of study:
Year 2
Education Level:
University
Gender:
Female
Current course:
Computer Science
Relevant subjects/modules:
SC2006, SC2005, SC2001
Preferred study location(s):
Woodlands, NTU, Yishun
School/ Institution:
(Optional)
K
kath24_.
Change password
Email reminders:
Preferred study timing:
8am-12pm, 6pm onwards
Usual study duration:
3-4 hours
(Optional)
Academic grades/ CGPA:
(Optional)
Edit profile
Log out

6. Complete Use Case Model

I. Use Case Diagram



II. Use Case Descriptions

I. Account Management

Use Case ID:	UC-01		
Use Case Name:	RegisterAccount		
Created By:	Garlapati Shreeya	Last Updated By:	Garlapati Shreeya
Date Created:	7/9/2025	Date Last Updated:	7/9/2025

Actor:	User
Description:	Allows the user to register for a new account using his/her respective school email and can create a unique username to be used for subsequent logins. The user also has to fill up all mandatory fields such as Email Address, Gender, Year of study, Education level, Preferred Study Location(s), Preferred Study Timing, Current Study Course/subjects for account activation.

Preconditions:	The user has to have access to a valid school email address ending with edu.sg and did not register before with the same email.
Postconditions:	A new account is successfully created and activated upon verification. If verification is incomplete, the account remains inactive.
Priority:	High
Frequency of Use:	Occasionally across the year
Flow of Events:	1. The user navigates to the registration page. 2. User enters: Username, password, Email Address, Gender, Year of study, Education level, Preferred Study Location(s), Preferred Study Timing, Current Study Course/subjects for account activation. (mandatory fields) 3. System validates: a. Email format and match against the school's database. b. Username uniqueness to make sure the new username does not duplicate any existing ones. c. Password requirements. 4. The system goes to UC-02 to validate the email address.
Alternative Flows:	**AF-S3a:** If there is an invalid or non-.edu.sg email, the system will reject and prompt re-entry. It should display "This email is not registered with your school/institution! Please use an email address provided by your school/institution." **AF-S3b:** If there is a duplicate username, the system displays "Username already exists! Please select a new username." to request for a different username. **AF-S3c:** If a user types a password that does not meet ALL the requirements which are: ● contain at least 12 characters, ● at least 1 uppercase letter, ● at least 1 lowercase letter, ● at least 1 number, ● at least 1 special character the system displays "Password does not meet all requirements!" and prompts the user to modify the password. **AF-S6:** If there is a wrong verification code typed by the user, the system rejects and displays "Verification code entered is incorrect. Please check the code sent to your email and try again.". The account will not be activated until the code entered is correct.
Exceptions:	None

Includes:	UC-02 (ActivateAccount)
Special Requirements:	Passwords must be encrypted and stored securely.
Assumptions:	The user has active access to their school email inbox. The school database is accurate and accessible.
Notes and Issues:	None

Use Case ID:	UC-02		
Use Case Name:	ActivateAccount		
Created By:	Garlapati Shreeya	Last Updated By:	Shannon Wong
Date Created:	7/9/2025	Date Last Updated:	10/9/2025

Actor:	User
Description:	User has to verify the newly created account by entering a verification code sent to his/her respective email address.
Preconditions:	User has completed the RegisterAccount (UC-01) process and has filled in all the mandatory fields.
Postconditions:	Account is activated if verification code is valid, account remains inactive if invalid verification code is typed or if any mandatory fields are not filled.
Priority:	High
Frequency of Use:	Occasionally across the year
Flow of Events:	1. User has to complete the RegisterAccount process first by filling in all the mandatory fields. 2. User will receive a verification code to email address 3. User has to type the code into the verification page 4. System validates the code. 5. System activates the account and the user is now able to login with username and password
Alternative Flows:	**AF-S1:** If any of the mandatory profile fields are missing, system does not activate account and prompts User to complete them **AF-S2:** If the user does not receive a code, the user requests the system to resend the code. The system will display “Didn’t receive any code? Click here to resend the code.” where the user can click and the code will be resent to the email. **AF-S4:** If wrong code is typed, the system will prompt the user to retype the code until it matches with the one sent to his/her email.
Exceptions:	None
Includes:	None
Special Requirements:	None

Assumptions:	Users have access to their school email inbox. Email system delivers verification code successfully.
Notes and Issues:	None

Use Case ID:	UC-03		
Use Case Name:	Login		
Created By:	Aurellia Maharani Putri	Last Updated By:	Shannon Wong
Date Created:	7/9/2025	Date Last Updated:	10/9/2025

Actor:	User
Description:	Allows the user to log into his/her UpGrade Account using his/her username and password.
Preconditions:	User must have a registered account.
Postconditions:	The user is logged into the UpGrade application and is navigated to the homepage of the website.
Priority:	High
Frequency of Use:	High (login whenever the user uses the website)
Flow of Events:	1. The system allows the user to login with his/her username and password 2. The User enters his/her username and password. The password is masked as asterisks, but the user can choose to unmask it by clicking on the eye icon. 3. The user selects the “Login” button. 4. The user is able to view the homepage of the website if and only if both username and password are valid and correct.
Alternative Flows:	**AF-S4:** If the username does not exist or the password is incorrect, the system will prompt the user to re-enter username or password.
Exceptions:	If any required information is missing, an error message will be displayed to request the user to fill in both the username and password fields.
Includes:	UC-04 (ResetPassword) if the user fails to remember the password and is unable to login.
Special Requirements:	The system needs to validate user input data.
Assumptions:	The User has an existing UpGrade account.
Notes and Issues:	None

Use Case ID:	UC-04
Use Case Name:	ResetPassword

Created By:	Aurellia Maharani Putri	Last Updated By:	Shannon Wong
Date Created:	7/9/2025	Date Last Updated:	10/9/2025

Actor:	User
Description:	Allows registered users of the UpGrade website to reset their password when they have forgotten it or need to change it for security reasons. The system provides a secure mechanism to verify the user's identity through email verification before allowing password reset.
Preconditions:	1. User must have a registered account on the UpGrade website. 2. The user's email address must be valid and accessible. 3. User must have access to their registered email account. 4. The system must be operational and accessible. 5. Email service must be functional.
Postconditions:	1. User's password is successfully updated in the system. 2. The old password is no longer valid. 3. User receives confirmation of successful password reset via email. 4. User can log in with the new password.
Priority:	High
Frequency of Use:	Low (only when users wish to change password)
Flow of Events:	1. User navigates to the UpGrade login page. 2. User clicks on "Forgot Password?"option. 3. System displays password reset request form. 4. User enters their registered email address. 5. User clicks the "Send Reset Code" button. 6. System validates the email format and checks if email exists in the user database. 7. System generates a secure, time-limited reset code (10 minutes). 8. System sends password reset email with reset code to the user's email. 9. System displays a confirmation message: "Password reset code has been sent to your email" and displays a field to enter the reset code. 10. The user enters the reset code in the website and is prompted to enter a new password. 11. User enters new password 12. The user confirms the new password by re-entering it a second time.

	13. System validates password requirements (as specified in UC-01). 14. System updates the password in the database. 15. System sends the password change confirmation email. 16. System displays a success message and redirects to the login page. 17. User logs in with the new password.
Alternative Flows:	**AF-S13**: If the new password entered for the second time does not match the new password entered for the first time, the system displays an error message and redirects the user to S-12 again.
Exceptions:	1. Invalid Email Format At step 6: System displays error "Please enter a valid email address" User must correct email format and resubmit 2. Expired Reset Code At step 10: System displays error "Reset code has expired. Please request a new one." User must restart the process from step 5 3. Invalid Reset Code The system will prompt the user to re-enter code or request a new reset code, where the user restarts the process from step 5. 4. Password Doesn't Meet Requirements At step 13: the system displays “Password does not meet all requirements!” and prompts the user to modify the password.
Includes:	None
Special Requirements:	None
Assumptions:	1. User has access to their registered email account 2. Email service provider delivers emails reliably 3. User's email client doesn't block UpGrade emails 4. User's browser supports the UpGrade website functionality
Notes and Issues:	None

II. Profile Management

Use Case ID:	UC-05		
Use Case Name:	ManageProfile		
Created By:	Garlapati Shreeya	Last Updated By:	Shannon Wong

Date Created:	7/9/2025	Date Last Updated:	10/9/2025

Actor:	User
Description:	Users are able to manage their own profile in UpGrade. A User can view and update their personal information anytime. The system enforces completion of mandatory fields before saving.
Preconditions:	The user is logged into the system.
Postconditions:	User's profile information is saved and updated if all mandatory fields are filled.
Priority:	Medium (necessary to be matched with profiles in similar educational levels)
Frequency of Use:	Occasionally (when user wants to review the information and update)
Flow of Events:	1. User navigates to the Profile feature. 2. The system displays the User's personal details, including both optional and mandatory fields filled during the RegisterAccount (UC-01) process. 3. User clicks on Edit Profile. 4. User updates any personal information, such as: a. Mandatory fields: Year of Study, Preferred Study Location(s), Preferred Study Timing, Current Study Course / Subjects. b. Optional fields: School Name, Academic Grades/GPA and Usual Study Duration. c. Turn on/off email reminders for study group meetings using a toggle switch. 5. User clicks Save. 6. System validates that all mandatory fields are completed. 7. System displays a confirmation message "Profile is updated successfully". 8. Updated profile is displayed back to the User.
Alternative Flows:	**AF-S5:** If the user does not click save, no updates are saved and the system keeps previous information. **AF-S6:** If User omits any of the mandatory fields, system prompts User to complete mandatory fields before saving. It should display "All mandatory fields are not filled!".
Exceptions:	None
Includes:	None
Special Requirements:	None
Assumptions:	The user provides accurate and truthful information.
Notes and Issues:	None

III. Discovery and Search

Use Case ID:	UC-06		
Use Case Name:	BrowseProfile		
Created By:	Aurellia Maharani Putri	Last Updated By:	Shannon Wong
Date Created:	8/9/2025	Date Last Updated:	10/9/2025

Actor:	User
Description:	Allows registered users to browse through all user profiles on the UpGrade homepage. Users can only view profiles of other users who are at the same educational level (Secondary, Junior-College, Polytechnic, or University) to find potential study mates with similar academic backgrounds.
Preconditions:	1. User must be logged into the UpGrade system 2. User must have a complete profile with educational level specified 3. Other users' profiles exist in the system.
Postconditions:	1. User successfully views a list of profiles from users at the same educational level 2. User can see mandatory profile information of other users 3. User can navigate through multiple profiles
Priority:	High
Frequency of Use:	High
Flow of Events:	1. The system identifies the user's educational level from the filled mandatory field in the profile. 2. System displays the homepage. 3. System automatically loads and displays profiles of all users at the same educational level. 4. System shows profile cards with mandatory information: ● Username/Name ● Gender ● Year of Study ● Preferred Study Location(s) ● Preferred Study Timing ● Current Study Course/Subjects There is a "+ Invite" button under each profile card. 5. User scrolls through the list of profiles. 6. User can click on individual profiles to view full details. 7. System displays selected profile in detailed view. 8. User can navigate back to browse more profiles.
Alternative Flows:	**AF-S4:** If no other users exist at the same educational level (no other profiles). The system displays a message: "No profiles available at your educational level."

Exceptions:	System Load Error: At step 4, if profiles fail to load, the system displays "Unable to load profiles. Please refresh the page." User can retry by refreshing the webpage.
Includes:	UC-07(SearchFilterProfile) to narrow down viewed profiles
Special Requirements:	None
Assumptions:	1. Users have specified their educational level correctly. 2. Profile information is current and accurate.
Notes and Issues:	None

Use Case ID:	UC-07		
Use Case Name:	SearchFilterProfile		
Created By:	Aurellia Maharani Putri	Last Updated By:	Shannon Wong
Date Created:	8/9/2025	Date Last Updated:	10/9/2025

Actor:	User
Description:	Allows users to search for specific users by username or name and apply filters based on profile attributes to narrow down the list of profiles displayed on the homepage, making it easier to find compatible study partners.
Preconditions:	1. User must be logged into the UpGrade system 2. User must be on the homepage with profiles visible 3. Profile data must be available in the system
Postconditions:	Success: 1. Search results display relevant user profiles matching search criteria 2. Filtered results show profiles matching selected filter attributes 3. User can view refined list of potential study partners Failure: 1. No results message displayed if no matches found 2. Original profile list maintained if search/filter fails 3. Error message for invalid search/filter parameters
Priority:	High
Frequency of Use:	Very High (Users frequently search for specific study partners)
Flow of Events:	**Search:** 1. User is in homepage with profiles displayed 2. User clicks on search bar at top right of page 3. User enters search term (username or display name) 4. System processes search in real-time or after clicking search button 5. System searches through:

	○ Usernames (exact and partial matches) ○ Display names (exact and partial matches) ○ Both fields within same educational level only 6. System displays matching profiles 7. Search results replace current profile list 8. User reviews search results and can click on profiles **Filter:** 1. User locates filter panel/dropdown on homepage 2. User selects filter categories: ○ **Year of Study:** Dropdown with available years ○ **Preferred Study Location:** Drop down with search feature ○ **Preferred Study Duration:** Radio buttons (1hr, 2hr, 3hr+, flexible) ○ **Study Timing:** Checkboxes (Morning, Afternoon, Evening, Night) ○ **Study Subjects:** Multi-select dropdown with search feature 3. User applies filters by clicking "Apply" button 4. System processes filter criteria with AND logic 5. System displays profiles matching ALL selected criteria 6. Filter tags show active filters above results 7. User can remove individual filters or clear all **Combined Search and Filter:** 1. User enters search term AND selects filter criteria 2. System applies search first, then filters the search results 3. Results show profiles matching both search term AND filter criteria
Alternative Flows:	**AF-S6 (Search): No Search Results** No profiles match search term ● Display "No students found matching '[search term]'" ● Show suggestions: "Try checking the spelling or searching with different terms" ● Provide "Clear search" option **AF-S5: (Filter): No Filter Results** No profiles match filter criteria ○ Display "No students match your selected criteria" ○ Show active filters with option to remove them

	○ Suggest "Try removing some filters to see more results"
Exceptions:	1. Invalid search characters → "Please enter valid search terms" 2. Search/Filter processing error → "Search temporarily unavailable" 3. Too many filter combinations → "Please select fewer filter options" 4. Network timeout → Retry option with cached results if available
Includes:	UC-06 (BrowseProfile) to view profiles of other users of similar educational level upon search and filter.
Special Requirements:	None
Assumptions:	1. Users will use reasonable search terms 2. Profile data is indexed for fast searching 3. Filter categories align with profile data structure
Notes and Issues:	

Use Case ID:	UC-08		
Use Case Name:	BrowseStudyGroup		
Created By:	Aurellia Maharani Putri	Last Updated By:	Shannon Wong
Date Created:	8/9/2025	Date Last Updated:	10/9/2025

Actor:	User
Description:	Allows users to browse through all public study groups available on the UpGrade platform. Users can discover existing study groups to join based on their academic interests and schedule preferences.
Preconditions:	1. User must be logged into UpGrade system 2. Public study groups must exist in the system 3. Study groups section must be accessible
Postconditions:	Success: ● User views list of available public study groups that are open to join ● Study group details are accessible for review ● User can identify groups to potentially join ● User can view separate list of public study groups he/she created Failure: ● Error message if study groups cannot be loaded ● User remains on current page with retry option

Priority:	High
Frequency of Use:	Regular (Users browse study groups when seeking collaborative learning)
Flow of Events:	1. User logs into UpGrade system 2. User navigates to "Study Groups" section, then goes to “All groups” tab 3. System retrieves all public study groups from database that are currently open (not full and not yet pass the meeting start time) 4. System displays study group cards showing: ● Group name ● Current joined participant count and capacity ● “Join” button 5. User scrolls through available study groups 6. User clicks “view details” button at the right side of a study group card and system shows: ● Meeting Date, Time (start and end) ● Meeting location ● Focus subject/topic (if specified) ● Current members’ profiles (specifies Group Host) 7. User can return to group list to continue browsing 8. User goes to “Created Groups tab” 9. System retrieve all public study groups from database created by user (where user is labelled as host) 10. System displays study group cards showing the same details, except “Join” button is replaced by “Edit group button”
Alternative Flows:	**AF-S3:** No public study groups exist ● Display "No public study groups are currently available" ● Show "Create New Group" button ● Suggest "Be the first to create a study group!
Exceptions:	1. Database error loading groups → "Unable to load study groups. Please refresh." 2. Group details load failure → "Group information unavailable. Try again." 3. Network connectivity issues → Offline message with retry option
Includes:	UC-09(SearchFilterStudyGroup) to narrow down viewed groups
Special Requirements:	1. **Privacy:** Only display public groups and appropriate member information 2. **Performance:** Load study groups within 4 seconds 3. **Real-time Updates:** Reflect current group status and member counts

	4. **Visual Design:** Clear group cards with essential information
Assumptions:	1. Group creators set appropriate publicity settings 2. Meeting information is current and accurate 3. System automatically removes closed (by host), full or expired study groups
Notes and Issues:	None

Use Case ID:	UC-09		
Use Case Name:	SearchFilterStudyGroup		
Created By:	Aurellia Maharani Putri	Last Updated By:	Shannon Wong
Date Created:	8/9/2025	Date Last Updated:	10/9/2025

Actor:	User
Description:	Enables users to search for study groups by name and apply sophisticated filters based on meeting details and focus subjects. This helps users find study groups that precisely match their academic needs and schedule availability.
Preconditions:	1. User must be logged into UpGrade system 2. User must be in the study groups section 3. Public study groups with searchable data must exist
Postconditions:	**Success:** ● Relevant study groups displayed based on search/filter criteria ● User can identify suitable groups to join ● Search/filter preferences saved for session **Failure:** ● No results message with suggestions to broaden criteria ● Original group list restored if search/filter fails
Priority:	High
Frequency of Use:	Very High (Users regularly search for specific study groups)
Flow of Events:	**Search:** 1. User is in Study Groups section's "All groups" tab with groups displayed 2. User locates search bar for study groups 3. User enters group name or keywords

	4. System searches through:
	  ○ Group names (exact and partial matches)
	5. System displays matching study groups
	6. User reviews search results
	
	Filter:
	
	1. User accesses filter panel for study groups
	2. User selects filter criteria
	3. **Meeting Date Filter:**
	  ○ Today's meetings
	  ○ This week's meetings
	  ○ Next week's meetings
	  ○ Custom date range picker
	  ○ Recurring meeting patterns
	4. **Meeting Time Filter:**
	  ○ Customizable time range (7 AM - 12 AM)
	5. **Location Filter:**
	  ○ Online meetings
	  ○ On-campus locations
	  ○ Off-campus locations
	  ○ Specific buildings/rooms
	  ○ Distance from user location
	6. **Focus Subject Filter:**
	  ○ Academic subjects dropdown
	  ○ Course codes
	  ○ General categories (STEM, Humanities, etc.)
	  ○ Multiple subject selection
	7. User applies selected filters
	8. System processes filters with AND logic
	9. System displays groups matching ALL criteria
	10. Active filters displayed as removable tags
	11. User can modify or clear filters
	
	Combined Search and Filter:
	
	1. User enters search term AND applies multiple filters
	2. System processes search first, then applies filters
	3. Results show groups matching search AND all filter criteria
	4. System shows result count and applied criteria

Alternative Flows:	**AF-S5 (Search):** No groups match search criteria ● Display "No study groups found matching '[search term]'"

	● Suggest alternative search terms ● Show "Create New Group" option **AF-S5 (Filter):** No groups match filter combination ● Display "No study groups match your selected criteria" ● Show active filters with individual remove options ● Suggest "Try adjusting your criteria" with filter recommendations
Exceptions:	1. Invalid date range selection → "Please select a valid date range"
Includes:	UC-08 (BrowseStudyGroup)
Special Requirements:	1. **Filter Memory:** Remember user's preferred filters 2. **Progressive Disclosure:** Show basic filters first, advanced on request 3. **Clear State:** Easy way to clear all filters and start over 4. **Result Context:** Show why results match (which filters triggered match)
Assumptions:	1. Study group information is accurate and up-to-date 2. Meeting times are stored in standardized format 3. Location data is geocoded for distance calculations 4. Subject classifications are consistent across groups
Notes and Issues:	1. **Performance:** Index frequently searched fields for faster results 2. **Enhancement:** Add predictive search suggestions 3. **UX:** Consider map view for location-based results 4. **Analytics:** Track popular search terms and filters for optimization

IV. Study Group Creation and Membership

Use Case ID:	UC - 10		
Use Case Name:	ManageStudyGroup		
Created By:	Joshua Tan	Last Updated By:	Joshua Tan
Date Created:	8/9/2025	Date Last Updated:	8/9/2025

Actor:	User
Description:	Allows User to manage a study group
Preconditions:	User must be logged in and is authenticated
Postconditions:	The status of the group should be changed according to the intent of User
Priority:	High

Frequency of Use:	High
Flow of Events:	1. UpGrade shall allow user to either create or close a study group 2. If User selects to create a study group, User will use the included Use Case **CreateStudyGroup** 3. If User selects to close a study group, User will use the included Use Case **CloseStudyGroup**
Alternative Flows:	None
Exceptions:	None
Includes:	UC-11(CreateStudyGroup), UC-12 (CloseStudyGroup)
Special Requirements:	None
Assumptions:	User is a host of the study group they want to close
Notes and Issues:	None

Use Case ID:	UC - 11		
Use Case Name:	CreateStudyGroup		
Created By:	Joshua Tan	Last Updated By:	Shannon Wong
Date Created:	8/9/2025	Date Last Updated:	24/9/2025

Actor:	User
Description:	User can create a study group with a visibility status (Public/ Private), User then becomes the host of the study group. The study group can be joined by any other Users.
Preconditions:	User must be logged in and is authenticated
Postconditions:	A study group is created with a unique Group ID randomly generated by UpGrade. UpGrade shall display the created Study Group in a separated section at the top of the StudyGroup page for the Hosts.
Priority:	High
Frequency of Use:	High
Flow of Events:	1. User navigates to the study groups page. The system will display a “ + Create new group” button under Study Group section’s “Created groups” tab 2. When the user selects the “+ Create new group” button, the system will request the user to fill in details for the new study group - Group Name - Meeting Date and Time (start and end) - Location → includes UC-27 - Maximum number of participants. - Focus Subjects (Optional) - Tags (optional) eg. “A-levels”, “exam revision”

	Created study groups are displayed in the “Created groups” tab and are specific to each user. 3. If the User selects Study Group Visibility to be Public: 3.1. UpGrade shall list the created Study Group on the public StudyGroup page’s “All groups” tab.
Alternative Flows:	AF - S3: If the User selects Study Group Visibility to be Private: 1. UpGrade shall display “+ Invite” option for each profile in the homepage 2. If User selects “+ Invite” option, UC-14 will be triggered
Exceptions:	1. If any mandatory details are missing or invalid, the system will display an error message.
Includes:	UC-14(InviteUser), UC-27(SelectStudyLocation)
Special Requirements:	UpGrade needs to validate User input data
Assumptions:	None
Notes and Issues:	None

Use Case ID:	UC - 12		
Use Case Name:	CloseStudyGroup		
Created By:	Joshua Tan	Last Updated By:	Joshua Tan
Date Created:	8/9/2025	Date Last Updated:	18/10/2025

Actor:	User
Description:	Allow Hosts to close a specific Study Group
Preconditions:	Study group is not expired Host user has created a study group previously
Postconditions:	The Study group is successfully closed
Priority:	Medium
Frequency of Use:	Medium
Flow of Events:	1. If Host chooses to close the group: 1.1. UpGrade shall display a separated section named “Created groups” at the top of StudyGroup page which displays a list of study groups that the User is a host of. 1.2. Hosts selects the “edit group” option on the study group from the list that the Host wants to close. 1.3. UpGrade displays the selected study group details which includes a “Close” option 1.4. Host can select the “Close” option any time 1.5. UpGrade will update the Group Database accordingly 2. If study group has reached maximum capacity: 2.1. UpGrade shall automatically close the group

	2.2. UpGrade shall remove any pending invitations for the study group from other Users inbox 2.3. If the study group visibility is “Public”: 2.3.1. UpGrade shall remove the study group from the StudyGroup page
Alternative Flows:	None
Exceptions:	1. If User is not the host of the study group, they will not be able to close the study group 2. System error in capacity check → may allow more than max number of participants to join the group
Includes:	None
Special Requirements:	System needs to detect when a study group is full
Assumptions:	User is the Host of the study group
Notes and Issues:	None

Use Case ID:	UC - 13		
Use Case Name:	JoinStudyGroup		
Created By:	Joshua Tan	Last Updated By:	Joshua Tan
Date Created:	8/9/2025	Date Last Updated:	8/9/2025

Actor:	User
Description:	User can join any public study groups from the homepage
Preconditions:	User must be logged in and is authenticated
Postconditions:	User will be added to the study group The capacity of the group will be updated.
Priority:	High
Frequency of Use:	High
Flow of Events:	1. User selects “Study Groups” option and UpGrade will navigate User to the StudyGroup page where they can view and join public study groups 2. If User selects “Join” on a specific study group, the system checks for conflicting schedules using the included use case UC-20 3. If valid, System adds User to the study group and calls UC-18 to notify other members of joining 4. UpGrade increments the number of joined participant count displayed in the “All groups” tab.
Alternative Flows:	None
Exceptions:	None
Includes:	UC-12 (CloseStudyGroup) when a user joining a study group causes it to be full UC-18 (NotifyUsers), UC-20 (ResolveConflict)

Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

Use Case ID:	UC - 14		
Use Case Name:	InviteUser		
Created By:	Joshua Tan	Last Updated By:	Shannon Wong
Date Created:	8/9/2025	Date Last Updated:	10/9/2025

Actor:	User
Description:	Any Users from a study group can invite other users
Preconditions:	User must be in the study group to send invitations User must be logged in and authenticated.
Postconditions:	An invitation is sent to the intended User and is displayed in their inbox
Priority:	High
Frequency of Use:	High
Flow of Events:	1. User selects “Homepage” and UpGrade shall navigate user to the page where they can invite other Users to their Study Group 2. User selects “+ Invite” and chooses a private or public study group they had joined to invite the selected other user 3. Upgrade shall send an invitation to the recipients inbox
Alternative Flows:	None
Exceptions:	None
Includes:	None
Special Requirements:	None
Assumptions:	The study group is not at maximum capacity
Notes and Issues:	None

Use Case ID:	UC - 15		
Use Case Name:	LeaveStudyGroup		
Created By:	Joshua Tan	Last Updated By:	Shannon Wong
Date Created:	8/9/2025	Date Last Updated:	10/9/2025

Actor:	User
Description:	Users can leave the study group any time before the study group expires.
Preconditions:	User has to be in the specific study group

Postconditions:	UpGrade shall update the User's schedule calendar accordingly UpGrade shall remove the User from the study group and decrement the group's capacity
Priority:	Medium
Frequency of Use:	Medium
Flow of Events:	1. User selects "Schedule" option on the side bar and UpGrade will navigate the user to schedule page where they can view their current study groups 2. UpGrade will display the "View details" option. 3. User selects the "View details" option and UpGrade will display the details of the selected study group and a "Leave" option. 4. User selects "Leave" option and UpGrade will remove user from the study group
Alternative Flows:	If the study group was previously full, the study group automatically reopens when a user leaves. Current remaining members in the group are allowed to invite other users to the group. - Public study group becomes visible in the Study Group page "All groups" tab again and other users are allowed to join
Exceptions:	1. If the study group has already expired, User cannot leave the group
Includes:	None
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

V. Invitation Inbox

Use Case ID:	UC-16		
Use Case Name:	ViewInbox		
Created By:	Chloe Atipinati Djohan	Last Updated By:	Chloe Atipinati Djohan
Date Created:	9/9/2025	Date Last Updated:	9/9/2025

Actor:	User
Description:	Allows the user to view pending study invites with group details and member list.
Preconditions:	User is logged in
Postconditions:	None
Priority:	High
Frequency of Use:	Medium
Flow of Events:	1. User navigates to the Inbox page.

	2. System lists out all pending invites with a “view group details” option each. 3. User can then accept or reject invites using UC-17 (Accept/RejectInvite)
Alternative Flows:	AF-S2: If there are no pending invites. 1. System displays the message “No pending invites.”
Exceptions:	None
Includes:	UC-17 (Accept/RejectInvite)
Special Requirements:	Needs real-time counts for the most recent participant counts.
Assumptions:	None
Notes and Issues:	None

Use Case ID:	UC-17		
Use Case Name:	Accept/RejectInvite		
Created By:	Chloe Atipinati Djohan	Last Updated By:	Chloe Atipinati Djohan
Date Created:	9/9/2025	Date Last Updated:	10/9/2025

Actor:	User
Description:	Allows User to accept or reject pending study invites. Adds User to the study group if User chooses to accept the invite, there is no schedule conflict, and the group is not full.
Preconditions:	Invitation exists and the group has not expired.
Postconditions:	Invitation is deleted. User is added as a study group member if the user accepts invitation and acceptance is valid. Invitation sender is notified if the user rejects the invitation.
Priority:	High
Frequency of Use:	Medium
Flow of Events:	1. User navigates to Inbox page. 2. System displays choices “Accept” and “Reject” for pending invitations. 3. User clicks “Accept” button. 4. System checks study group’s capacity and schedule conflicts using the included use case UC-20. 5. If valid, System adds User to the study group and increments the group’s member count. 6. System removes invitation from Inbox.
Alternative Flows:	AF-S3: If User wants to reject invite. 1. User clicks “Reject” button. 2. System sends a notification to the sender that User has rejected the invite using the included use case UC-18. 3. System removes the invitation from Inbox.

	AF-S5: If the study group is full or expired. 1. Trigger UC-24
Exceptions:	None
Includes:	UC-18 (NotifyUsers), UC-20 (ResolveConflict), UC-24(ClosedExpiredGroups)
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

Use Case ID:	UC-18		
Use Case Name:	NotifyUsers		
Created By:	Chloe Atipinati Djohan	Last Updated By:	Chloe Atipinati Djohan
Date Created:	10/9/2025	Date Last Updated:	10/9/2025

Actor:	System
Description:	Sends a notification to all group members if a User joins or leaves the group. Sends a notification to a User if someone they invited rejected their invitation.
Preconditions:	A User joins/leaves a group or a User rejects a study invite.
Postconditions:	Notifications delivered. Targeted Users receive notifications.
Priority:	Medium
Frequency of Use:	Medium-High
Flow of Events:	1. A user joins or leaves a study group 2. System sends a notification to all the group members' Inboxes. It displays the message "{username} has joined/left '{group name}'."
Alternative Flows:	AF-S1: User rejects invitation. 1. A User clicks reject on an invite. 2. System sends a notification to the inviter's Inbox. It displays the message "{username} rejected your invite to join '{group name}'."
Exceptions:	None
Includes:	None
Special Requirements:	None
Assumptions:	Users have notification preferences as on.
Notes and Issues:	None

VI. Scheduling and Conflict Resolution

Use Case ID:	UC-19

Use Case Name:	ViewSchedule		
Created By:	Shannon Wong	Last Updated By:	Shannon Wong
Date Created:	7/9/2025	Date Last Updated:	7/9/2025

Actor:	User
Description:	Allows the user to view the study groups they joined in calendar format. Each calendar entry includes Group Name, Date, Start Time, End Time, Location, Current Participant Count and groupID.
Preconditions:	The user must be logged in their account and have at least 1 active study group (upcoming or ongoing) or past study group in their account (that has yet to be auto-deleted).
Postconditions:	User views scheduled study group entries displayed in calendar view
Priority:	High
Frequency of Use:	Daily to weekly
Flow of Events:	1. The user navigates to the “Schedule” page. 2. The system queries the database for all study groups joined by the user. 3. The system displays entries in a calendar format (month view), where each entry is in its respective date and time slot. 4. Each entry shows Group Name and Location. 5. The user can view full group details by clicking onto “View Details”, which shows Group Name, Date, Start Time, End Time, Location, Current Participant Count, Participant profiles, unique groupID and “Leave Group” option.
Alternative Flows:	AF-1: If no groups are joined, the system displays “No study groups scheduled.” AF-2: If the “Leave Group” option is selected, the old study group is removed from the user’s schedule.
Exceptions:	None
Includes:	Could include UC-1B (Clean Schedule) when expired groups are removed
Special Requirements:	Calendar must be responsive.
Assumptions:	Users expect calendar updates in real-time after joining or leaving new study groups, and/or deletions of expired study groups.
Notes and Issues:	None

Use Case ID:	UC-20
Use Case Name:	ResolveConflict

Created By:	Shannon Wong	Last Updated By:	Shannon Wong
Date Created:	7/9/2025	Date Last Updated:	7/9/2025

Actor:	User
Description:	The user attempts to accept an invitation to join a new study group or clicks join to join a public study group. The system detects if the timing conflicts with an existing joined group. If a conflict exists, the system prompts the user to leave the existing group before accepting or joining the new one.
Preconditions:	The user must have at least one joined group in their schedule and a pending invite exists.
Postconditions:	The new group invite is either accepted (and the user left any old conflicting group) or the invite is rejected.
Priority:	High
Frequency of Use:	Moderate (whenever invites overlap)
Flow of Events:	1. The user joins via invite. a. The user opens Invitation Inbox b. The user selects the “Accept” option on a pending invitation. c. The system compares study group timing with the user’s existing schedule. d. If there is no conflict, the system adds the user to the new study group. e. If there is a conflict, the system displays a prompt: - “This invite overlaps with Group X. Leave Group X to join this group?” f. If the user confirms, the system removes the user from their old study group and joins the new one. g. If the user cancels, the system does not join the new group. 2. The user joins via study group page. a. The user views all public study group listings. b. The user selects the “Join” option for a specific study group. c. The system compares study group timing with the user’s existing schedule. d. If there is no conflict, the system adds the user to the new study group. e. If there is a conflict, the system displays a prompt: - “This invite overlaps with Group X. Leave Group X to join this group?” f. If the user confirms, the system removes the user from their old study group and joins the new one.

	g. If the user cancels, the system does not join the new group.
Alternative Flows:	AF-1: The user manually rejects the invitation instead of resolving conflict.
Exceptions:	System error retrieving the user's schedule of groups joined may prevent conflict detection, resulting in overlap in timings of at least 2 study groups.
Includes:	None
Special Requirements:	Conflict detection algorithm must consider overlapping time ranges, not just exact matches.
Assumptions:	Only one overlapping group conflict will be handled at a time.
Notes and Issues:	Edge case – two groups ending/starting within minutes from each other.

Use Case ID:	UC-21		
Use Case Name:	ReminderNotificationDelivery		
Created By:	Shannon Wong	Last Updated By:	Shannon Wong
Date Created:	7/9/2025	Date Last Updated:	10/9/2025

Actor:	System
Description:	The system automatically sends reminders for study groups the user has joined. Reminders are sent 24 hours before scheduled start time.
Preconditions:	A study group must exist and the User must be a confirmed participant.
Postconditions:	The user receives reminder notification (via email and in-app).
Priority:	High
Frequency of Use:	Daily (depending on group activity)
Flow of Events:	1. The system checks upcoming study groups. 2. If a study group is exactly 24 hours away from its start time, the system triggers reminder job. 3. A reminder is sent via email to the registered school email address if email notifications are enabled. 4. A reminder is sent to the inbox of the User. 5. A reminder is also displayed as an in-website alert when the user logs in. 6. System logs notification delivery status.
Alternative Flows:	AF-1: User has disabled email reminders. Only in-app alert and inbox reminder are sent.
Exceptions:	Email delivery failure.

Includes:	UC-23 (Toggle Notifications) when a user disables email notifications and does not receive a study group reminder via email.
Special Requirements:	Notifications must be time-zone aware.
Assumptions:	All users are in the same Singapore timezone.
Notes and Issues:	Retry logic needed for email failures.

Use Case ID:	UC-22		
Use Case Name:	CleanSchedule		
Created By:	Shannon Wong	Last Updated By:	Shannon Wong
Date Created:	7/9/2025	Date Last Updated:	7/9/2025

Actor:	System
Description:	The system cleans up the User's schedule calendar by automatically deleting expired study group entries 72 hours after their end time.
Preconditions:	The study group must have ended.
Postconditions:	The expired groups no longer appear in the calendar.
Priority:	Medium
Frequency of Use:	Daily (system cleanup job)
Flow of Events:	1. The system checks all study groups in the user's calendar. 2. For each group with end time 72 hours ago, the system deletes its entry from the schedule calendar of the user. 3. Calendar view refreshes with remaining groups.
Alternative Flows:	None
Exceptions:	Database cleanup job fails, expired groups remain visible after 72 hours.
Includes:	UC-19 (ViewSchedule) when re-displaying updated calendar.
Special Requirements:	The cleanup job must not remove groups still within 72 hours of end time.
Assumptions:	Expired data is no longer needed for analytics or reporting.
Notes and Issues:	Deleting expired data helps to save memory.

Use Case ID:	UC-23		
Use Case Name:	Toggle Notification		
Created By:	Shannon Wong	Last Updated By:	Shannon Wong
Date Created:	10/9/2025	Date Last Updated:	10/9/2025

Actor:	User

Description:	The user can enable or disable email notifications for study group reminders. In-app notifications (website alert and inbox reminder) remain unaffected.
Preconditions:	The user must be logged in and the user must have a valid registered school email.
Postconditions:	The system saves the user's preference in their profile settings. Reminders are only sent to email if email notifications are enabled.
Priority:	Medium
Frequency of Use:	Low (users set this once, occasionally update)
Flow of Events:	1. The user navigates to "Profile". 2. The system retrieves current notification preferences 3. The user toggles the "Email Reminders" switch ON (to the right) or OFF (to the left). 4. Confirmation message is displayed: a. "Email reminders are enabled." OR b. "Email reminders are disabled. You will still receive notifications in-website."
Alternative Flows:	AF-1: User cancels without saving, no change is made to email notification settings.
Exceptions:	Database update fails. An error message is displayed: "Unable to update preferences, try again later."
Includes:	UC-21(ReminderNotificationDelivery), behaviour is modified depending on toggle.
Special Requirements:	Preference change must take effect immediately, and be persistent across sessions.
Assumptions:	A user always receives in-website notifications (website alert and inbox reminder). Toggle only affects reminders sent to email.
Notes and Issues:	None

Use Case ID:	UC-24		
Use Case Name:	ClosedExpiredGroups		
Created By:	Shannon Wong	Last Updated By:	Shannon Wong
Date Created:	10/9/2025	Date Last Updated:	10/9/2025

Actor:	System
Description:	A study group is expired when its scheduled start time has passed. A study group is closed when the host user who created the study group decides to close it early before it is full. In either case, the group is removed from the Study Group page "All groups" tab, pending invitations are deleted and new invitations cannot be sent.

Preconditions:	Study group must exist and be either active or about to expire. Users may have pending invitations linked to the group.
Postconditions:	Expired groups are permanently closed (cannot reopen). Closed groups can be reopened if the host chooses to do so.
Priority:	High
Frequency of Use:	Frequent (every time a study group passes start time or when host decides to close it early)
Flow of Events:	1. Closed by host: a. Host user selects close option when editing created group details b. The system removes the public group from the public "All groups" tab in the Study Group page. The system deletes all pending invitations in invited users' inbox, prevent further invitations by current members in group for both public and private groups c. If the host selects open option later, the system redisplays the public study group. The system allows current members to invite other users for public/private study group 2. Time-based expiration: a. System checks study groups at scheduled start times b. System automatically expires the group when the meeting start time is reached: - Remove public group from "All groups" tab and remove private/public group from "Created groups" tab in Study Groups page - Deletes all pending invitations - Prevents further invitations c. Expired groups cannot be reopened and user can no longer leave
Alternative Flows:	None
Exceptions:	System may fail to run, allow expired groups to still appear in Study Group page
Includes:	None
Special Requirements:	None
Assumptions:	System clock is reliable and synchronised
Notes and Issues:	None

VII. Map and Study Locations

Use Case ID:	UC-25
Use Case Name:	BrowseStudyLocation

Created By:	Ming Xuan	Last Updated By:	Ming Xuan
Date Created:	8/9/2025	Date Last Updated:	8/9/2025

Actor:	User
Description:	Allows the user to look for available study locations using an interactive map. Users can click on markers in the map to view specific details regarding the location such as name and address. Users can then either search for specific locations and/or select their location of choice.
Preconditions:	User is logged in System has loaded in the Map
Postconditions:	User is able to Search and Select study locations
Priority:	Medium
Frequency of Use:	High
Flow of Events:	1. User navigates to the Location Map page. 2. The system then loads in the map showing different marked locations. 3. If User decides to search for their desired location, then User uses the included use case SearchStudyLocation. 4. If User has decided on their location, then User uses the included use case SelectStudyLocation.
Alternative Flows:	AF-S3: If no location exists after user search, the system displays no location available.
Exceptions:	System fails to load the map data.
Includes:	UC-26(SearchStudyLocation), UC-27(SelectStudyLocation)
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

Use Case ID:	UC-26		
Use Case Name:	SearchStudyLocation		
Created By:	Ming Xuan	Last Updated By:	Ming Xuan
Date Created:	8/9/2025	Date Last Updated:	8/9/2025

Actor:	User
Description:	Allows the user to search for desired location by entering keywords. The system will then display the locations matching the keywords.
Preconditions:	User is logged in and at Location Map page
Postconditions:	Search results displayed on map

Priority:	Medium
Frequency of Use:	Medium
Flow of Events:	1. User enters keywords (such as the region) into the search bar within the page. 2. System searches the database for matches. 3. System displays the location matching/similar to the keywords entered. 4. User clicks the location to view details.
Alternative Flows:	AF-2: If the search term does not match any locations in the database, the system displays no location found matching the search term.
Exceptions:	Search service unavailable
Includes:	None
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

Use Case ID:	UC-27		
Use Case Name:	SelectStudyLocation		
Created By:	MingXuan	Last Updated By:	Joshua Tan
Date Created:	8/9/2025	Date Last Updated:	18/10/2025

Actor:	User
Description:	Allows the user to choose their desired study location when creating a study group. The user can either search for their desired location or select from a list of locations (Libraries) that are near them.
Preconditions:	User is logged in User is in process of creating a new study group
Postconditions:	The selected location is saved as part of the study group details If no locations is selected, the study group cannot be formed
Priority:	High
Frequency of Use:	High
Flow of Events:	1. User chooses to create a study group 2. System prompts the user to select the location 3. User selects a study location from the interactive map 4. The selected study location is saved into the study group details.
Alternative Flows:	AF-3a: If the user fails to select a location, the system prevents the group from being created and displays a message prompting the user to select a location

Exceptions:	None
Includes:	None
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

VIII. Report Group and Manage Reports

Use Case ID:	UC-28		
Use Case Name:	ReviewReportedGroup		
Created By:	MingXuan	Last Updated By:	MingXuan
Date Created:	11/9/2025	Date Last Updated:	11/9/2025

Actor:	Admin
Description:	Allows the admin to review reported study groups, inspecting the group description or self-written tags to decide whether to DISMISS the report, REMOVE the group and/or BAN the host User who created the group.
Preconditions:	Admin is logged in, at least one report exists
Postconditions:	Report remains open, or is closed due to group being removed or is a false report
Priority:	High
Frequency of Use:	Low
Flow of Events:	1. Admin opens the list of Reports 2. Admin selects a report 3. The report details the information about the group such as the group name, description, tags, groupID, host's username and reporting user's username. 4. Admin decides whether to a. Remove Group & give the User who created the group a Warning. b. Remove Group & Ban the User if the User already has a Warning. c. Dismiss the Report
Alternative Flows:	AF-1: If there are no reports in the list, show No Reports Pending
Exceptions:	None
Includes:	UC-29 (RemoveGroupAndBanHost), UC-30 (DismissReport)
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

Use Case ID:	UC-29		
Use Case Name:	RemoveGroupAndBanHost		
Created By:	MingXuan	Last Updated By:	MingXuan
Date Created:	11/9/2025	Date Last Updated:	11/9/2025

Actor:	Admin
Description:	Admin removes a group that has been reported and found to have violated the policy. Ban the User who created the group.
Preconditions:	Admin is logged in At least one report exists Admin reviewed the report
Postconditions:	The group is removed from the list, pending invites to be canceled and group members' schedules are cleaned.
Priority:	High
Frequency of Use:	Low
Flow of Events:	1. From the report details the admin chooses Remove Group a. If the host User who created the group has not received any warnings before, System prompts the admin to Confirm to remove the group. b. If the host User who created the group has received warnings before, System notifies the Admin that the User has had a prior warning and prompts the admin to Confirm to remove the group and to Ban User. c. If the Group has already expired/closed. System shall BAN the host User if the User has a warning. d. If the Group has already expired/closed. System shall prompt the Admin whether to give the host User a warning if the User has had no prior warnings.
Alternative Flows:	None
Exceptions:	None
Includes:	None
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

Use Case ID:	UC-30		
Use Case Name:	DismissReport		
Created By:	MingXuan	Last Updated By:	MingXuan
Date Created:	11/9/2025	Date Last Updated:	11/9/2025

Actor:	Admin
Description:	Admin chooses to dismiss the report due to it being false.
Preconditions:	Admin is logged in At least one report exists Admin reviewed the report
Postconditions:	No changes to the group.
Priority:	High
Frequency of Use:	Low
Flow of Events:	1. From the report details, admin clicks Dismiss Report 2. System prompts admin for confirmation 3. Admin clicks Confirm 4. System closes the report as False Report 5. Admin decides how to deal with User who submitted the False Report a. If the User who submitted the report has not received any warnings before, System shall prompt the Admin whether to give the host User a warning if the User has had no prior warnings. b. If the User who reported the group has received warnings before, System notifies the Admin that the User has had a prior warning and prompts the admin to Confirm to Ban User.
Alternative Flows:	None
Exceptions:	None
Includes:	None
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

Use Case ID:	UC-31		
Use Case Name:	ReportGroup		
Created By:	Shannon Wong	Last Updated By:	Shannon Wong
Date Created:	24/9/2025	Date Last Updated:	24/9/2025

Actor:	User
Description:	A user reports a study group whose details are inappropriate. The user selects one or more violation categories from a predefined list (vulgarities, sexual content, harassment, harmful acts, spam, misinformation) and submits the report. The report is stored in the system for admins to view.
Preconditions:	User must be logged in and a study group must exist in the system

Postconditions:	Report is stored with groupID, group name, group description, group self-written tags, host's username and reporting user's username.
Priority:	High
Frequency of Use:	Low to moderate (only when inappropriate groups exist)
Flow of Events:	1. The user views specific study group in StudyGroup page 2. The user selects "report!" option of specific study group 3. System displays a list of violation categories 4. The user selects one or more categories 5. The user submits the report 6. The system validates that at least one category is selected 7. The system stores the report 8. The user receives a confirmation message that the report has been submitted via in-website alert displaying "Report is submitted successfully!"
Alternative Flows:	AF-5: The user selects cancel before submitting → no report is created, user is redirected back to StudyGroup page AF-6: The user submits without selecting a violation category → system shows an error message via in-website alert displaying "Error! Please select a category to report the group."
Exceptions:	None
Includes:	UC-28(ReviewReportedGroup) for admins to manage the submitted reports later on
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

IX. Private Chats

Use Case ID:	UC-32		
Use Case Name:	StartPrivateChat		
Created By:	Shannon Wong	Last Updated By:	Shannon Wong
Date Created:	16/10/2025	Date Last Updated:	28/10/2025

Actor:	User (primary), System (secondary)
Description:	A user or host initiates a one-to-one private chat conversation with another user. Any users can start a chat with: - the group host (via the study group page, or an invitation in the inbox page) - Another user of the same education level (via user's profile in the homepage)

	Hosts can start a chat with any joined member from the study group page (via group details under “Created Groups”). A chat starts when either participant sends the first message. The system checks if an existing chat exists between the 2 users and the system creates a new chat thread for both parties if it doesnt exist.
Preconditions:	Both participants must be registered users and the group exists. User have an invitation to that group (if starting a chat via the inbox page)
Postconditions:	A chat record is created and both participants can now send messages in the chat
Priority:	High
Frequency of Use:	Moderate (depends on interactions between host and user)
Flow of Events:	1. The user or host clicks *Message* button on one of the following: a. group details page (User → Host or Host → Member) b. invitation in inbox page (User → Host) c. profile of another user in homepage (User → User) 2. System checks if a chat already exists between the two users 3. If none exists, a new chat is created for both participants 4. A chat interface is opened in the chat page. 5. User or host sends the first message 6. The system saves the message in the database and updates both users’ chat pages.
Alternative Flows:	AF-3: Existing chat found, system redirects user to existing chat in chat page. AF-5: If the user does not send the 1st message, the chat will not be created and saved in the chat page.
Exceptions:	None
Includes:	UC-33(ViewContinueChat)
Special Requirements:	Chat must be private, only 2 participants in each chat New chats should appear instantly in both participants’ chat list (real-time update)
Assumptions:	Each unique pair of users has only one chat
Notes and Issues:	None

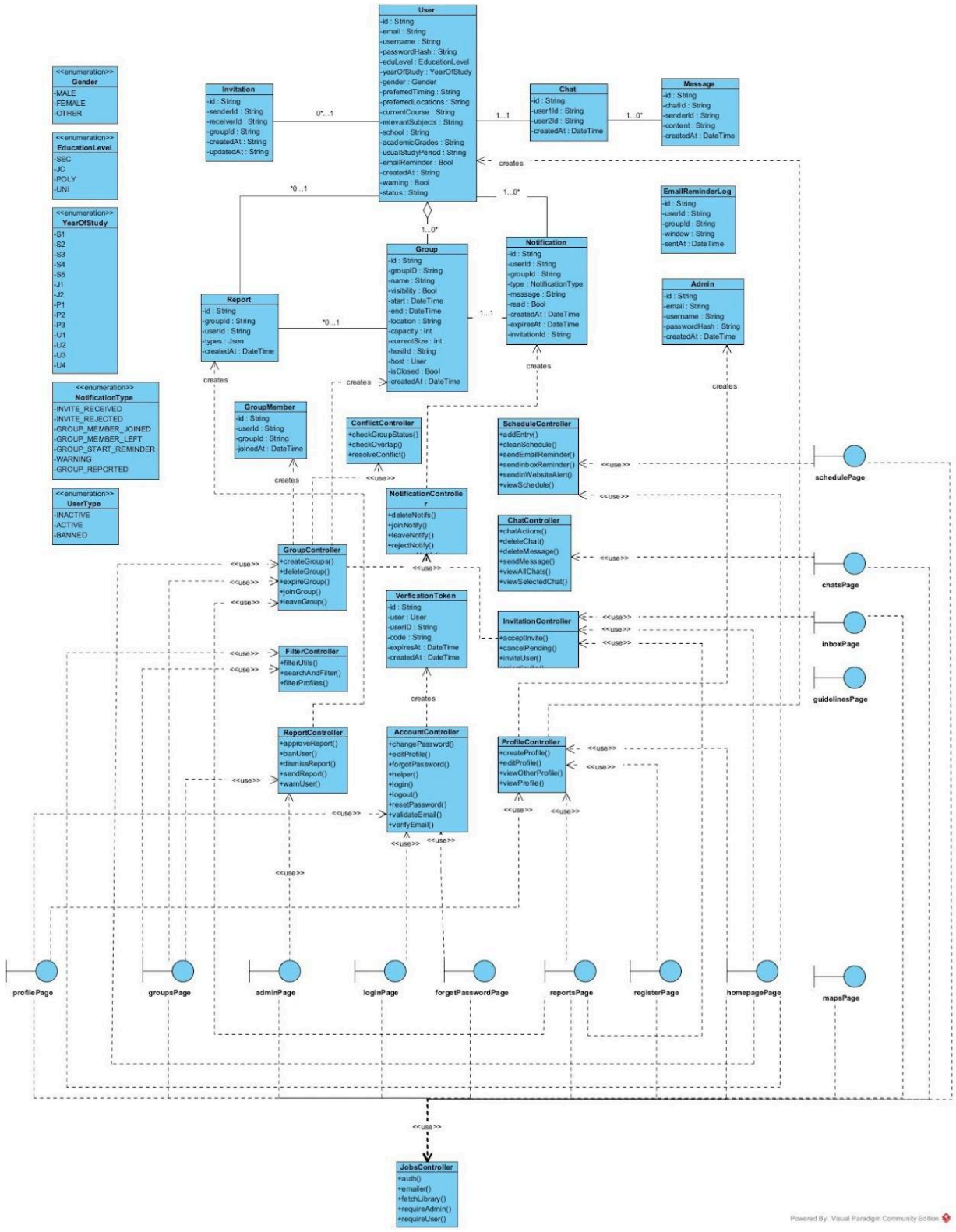
Use Case ID:	UC-33
Use Case Name:	ViewContinueChat

Created By:	Shannon Wong	Last Updated By:	Shannon Wong
Date Created:	16/10/2025	Date Last Updated:	28/10/2025

Actor:	User (primary), System (secondary)
Description:	The user or host views existing one-to-one chat conversations and continues messaging within them. The chat page displays all chats the user has previously sent or received messages. Messages appear in chronological order. User can reply in real-time
Preconditions:	The user is logged in. At least one chat exists
Postconditions:	New messages are saved in chat and updated for both sender and receiver
Priority:	High
Frequency of Use:	High (ongoing interaction)
Flow of Events:	1. The user navigates to Chat Page from navigation bar 2. System retrieves all chats involving the user 3. User selects a chat 4. System displays conversation history 5. User types and sends a message 6. System stores the message 7. Receiver is able to see new message upon opening chat 8. User can return to chat list and open other private chats
Alternative Flows:	AF-2: No chats yet, chat page displayed “No conversations yet”
Exceptions:	None
Includes:	UC-22(StartPrivateChat)
Special Requirements:	Chats sync in real time. Messages persist (stored in database) and load when chat is reopened
Assumptions:	None
Notes and Issues:	None

7. Design Model

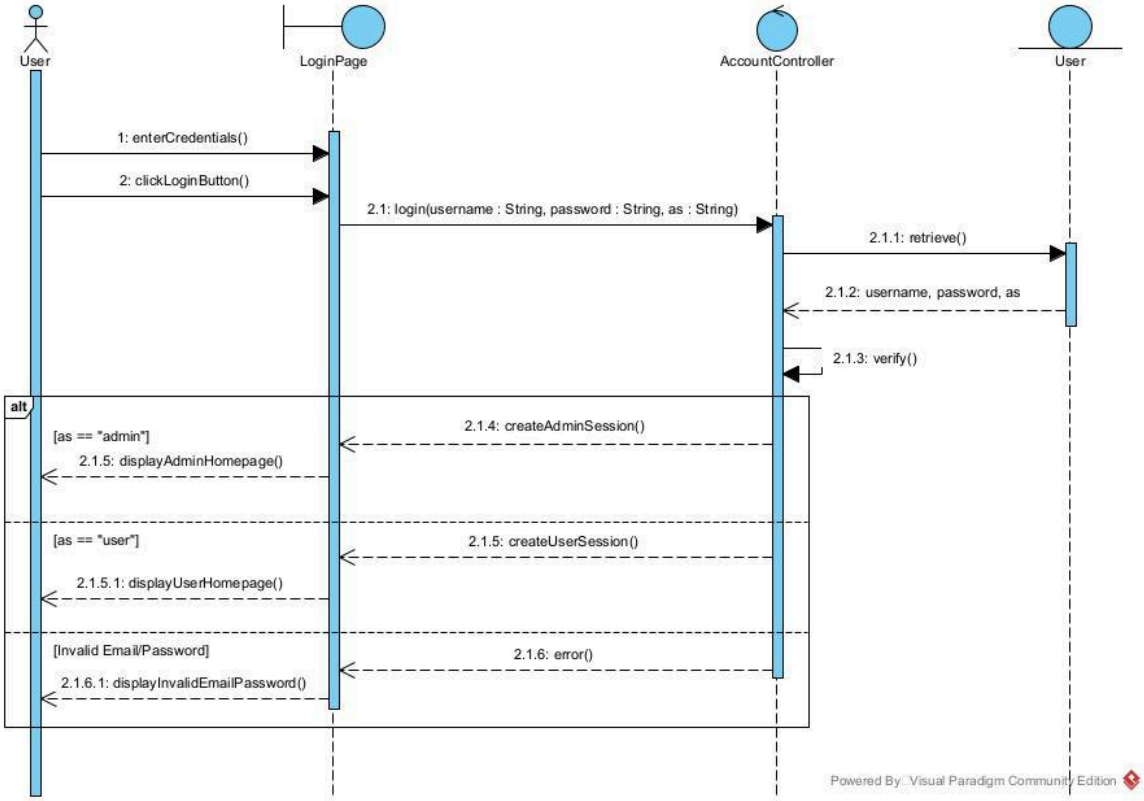
I. Class Diagram



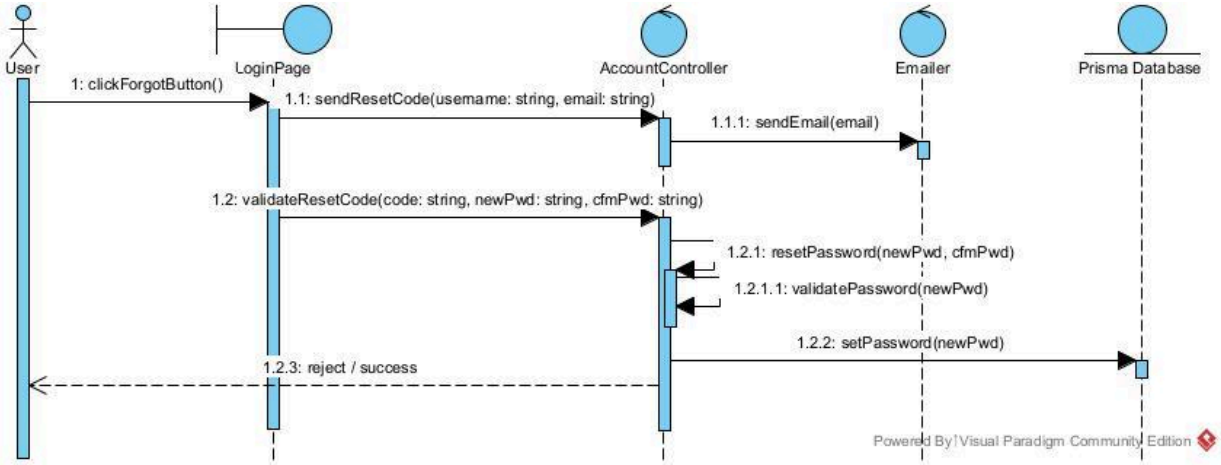
(please refer to "class diagram".pdf file attached in the same folder for a clearer image)

II. Sequence Diagrams

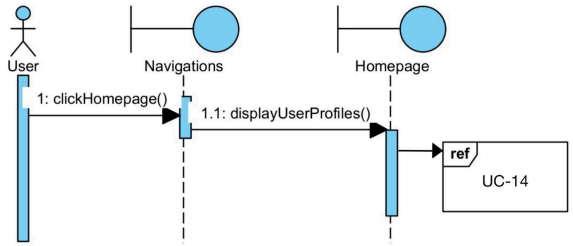
UC-03: Login



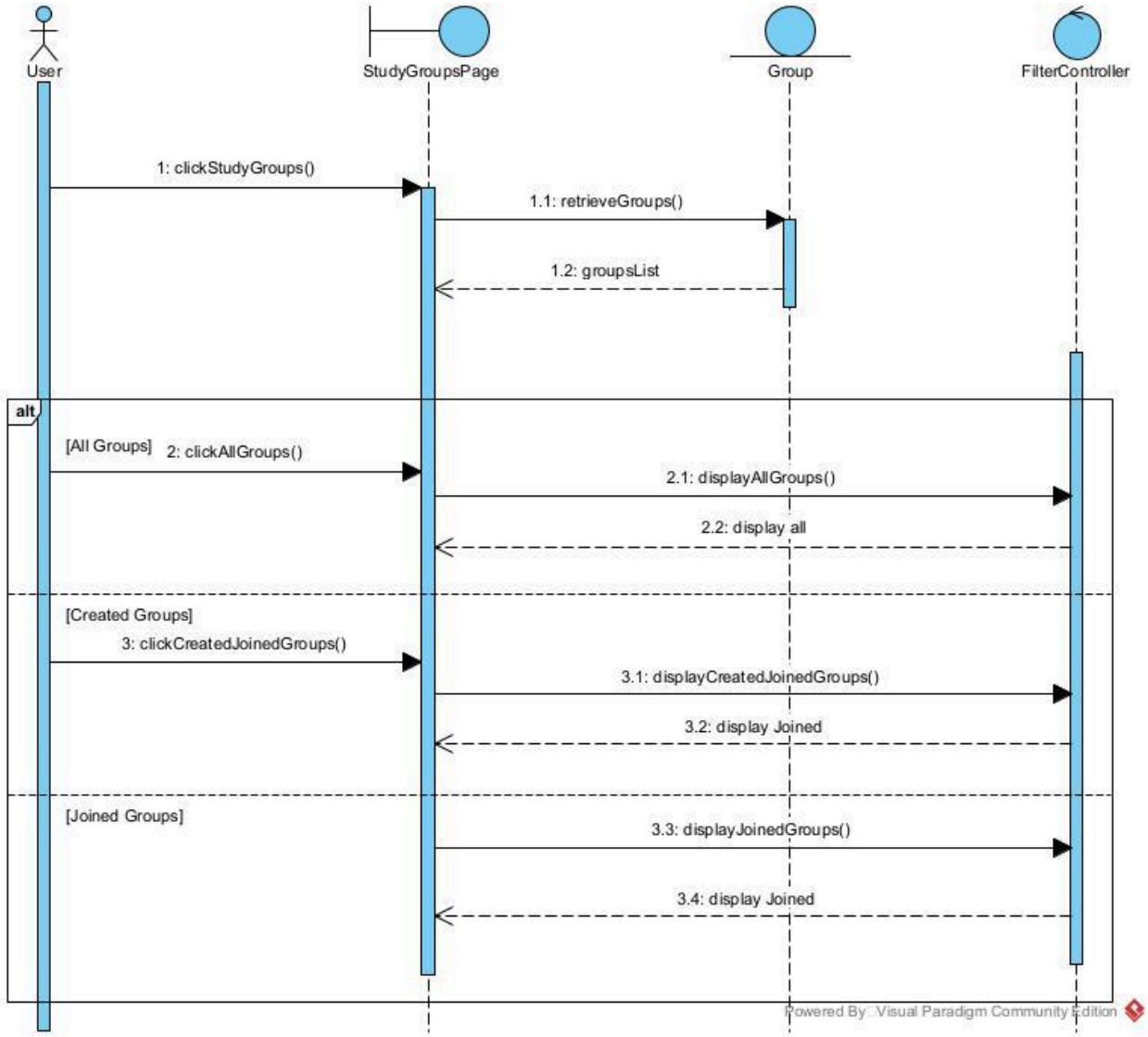
UC-04: ResetPassword



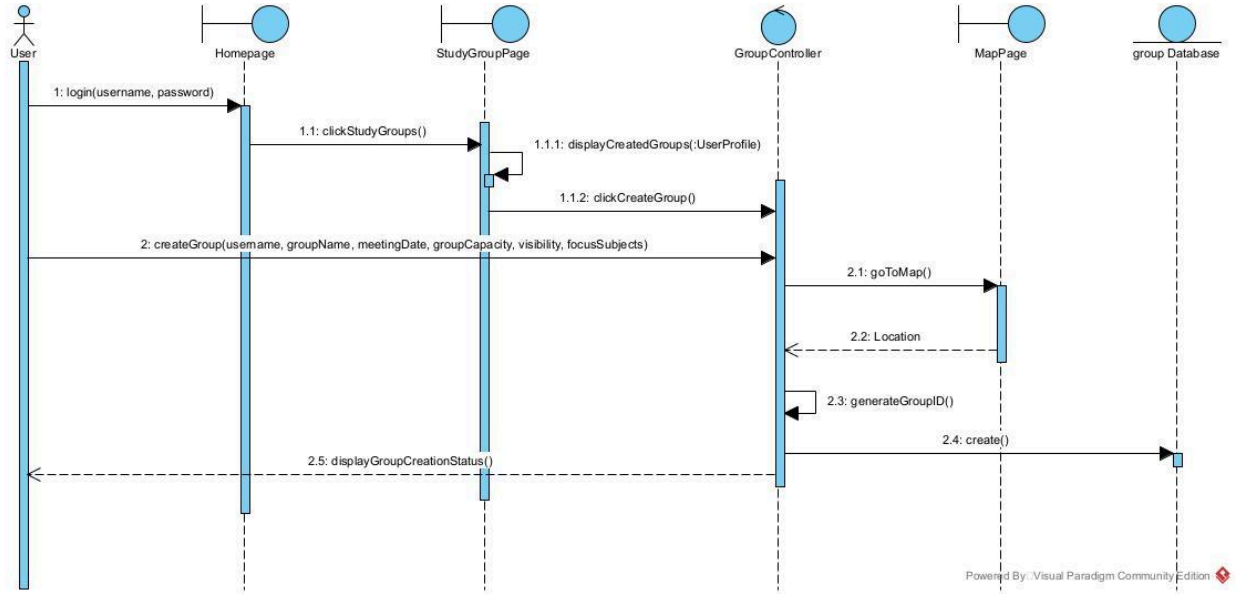
UC-06: BrowseProfile



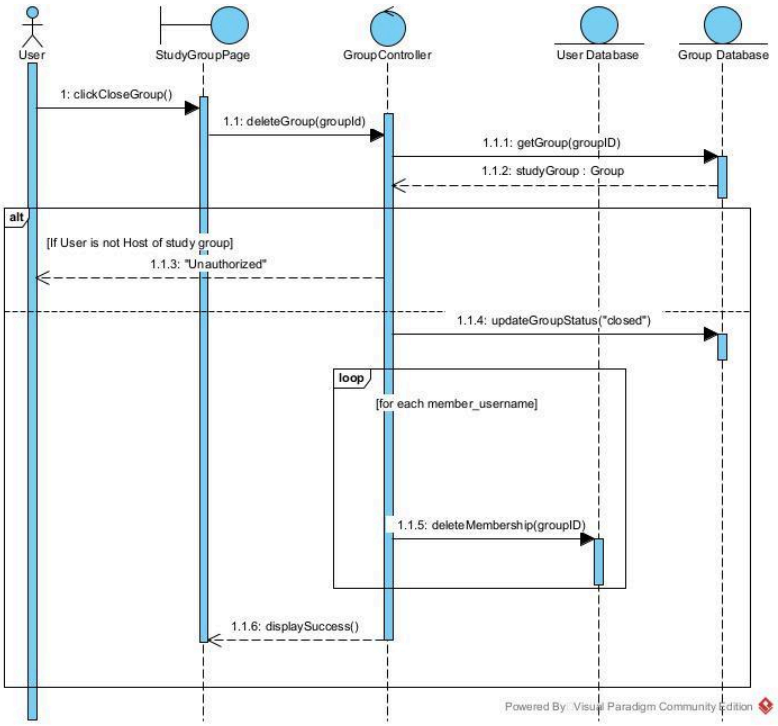
UC-08: BrowseStudyGroup



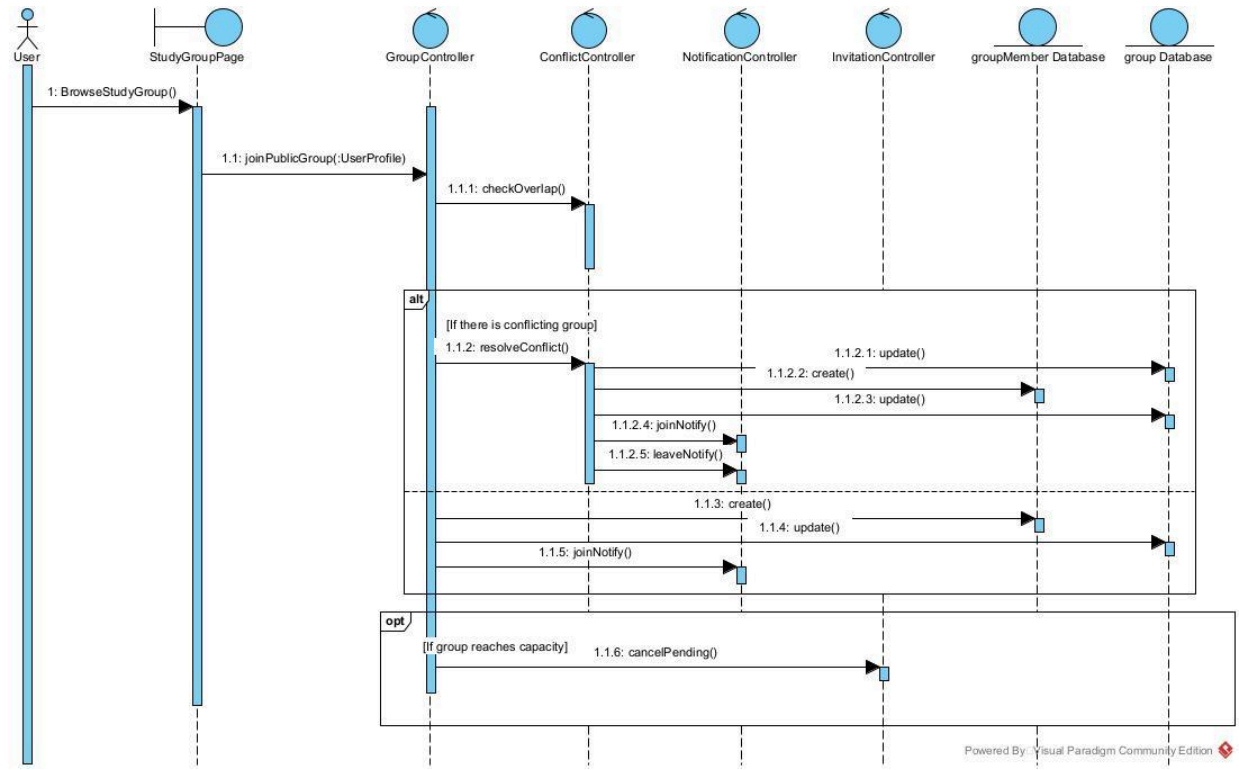
UC-11: CreateStudyGroup



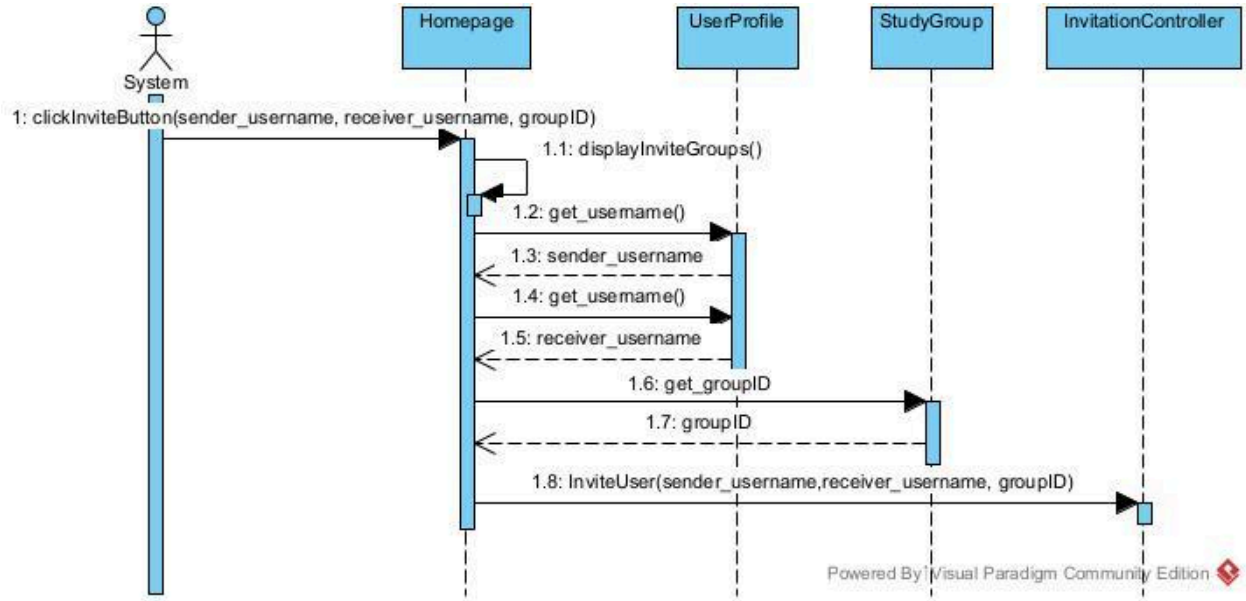
UC-12: CloseStudyGroup



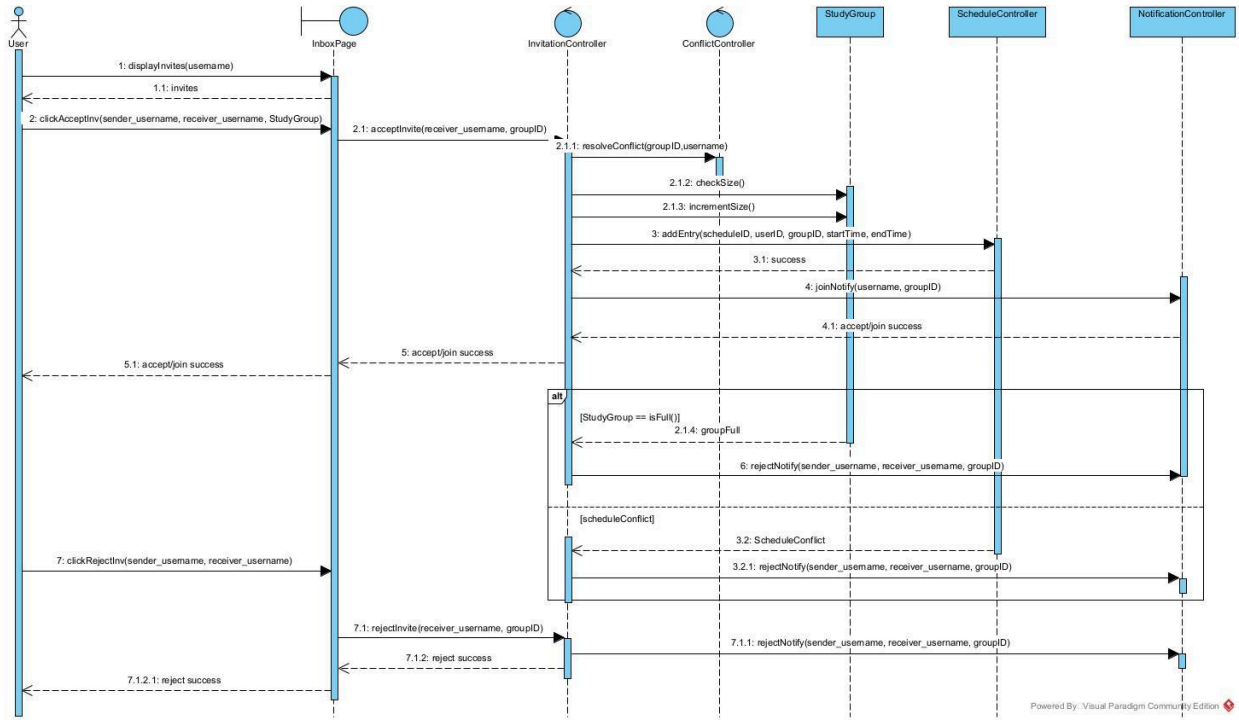
UC-13: JoinStudyGroup



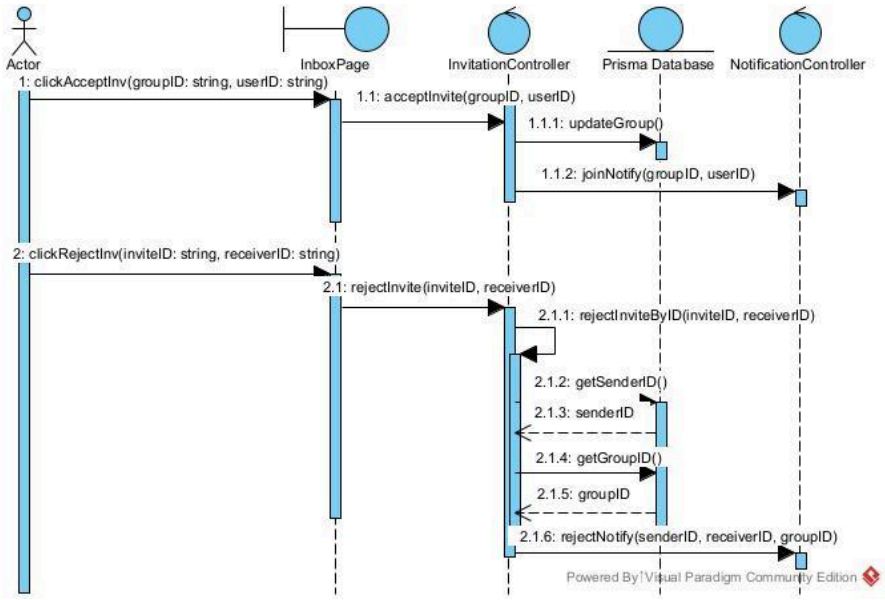
UC-14: InviteUser



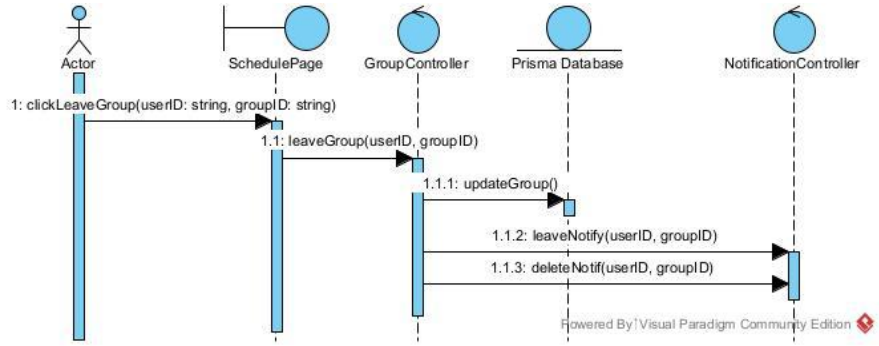
UC-17: Accept/RejectInvite



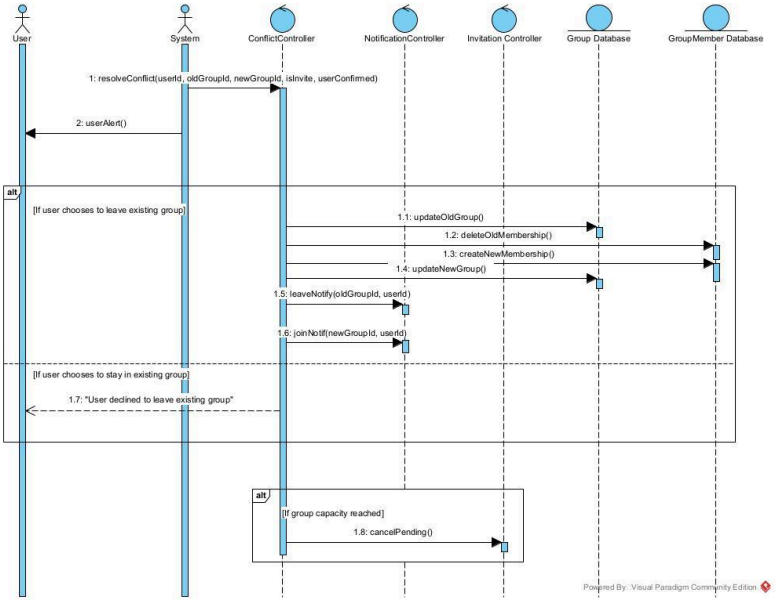
UC-18a: NotifyUsers (accept/reject)



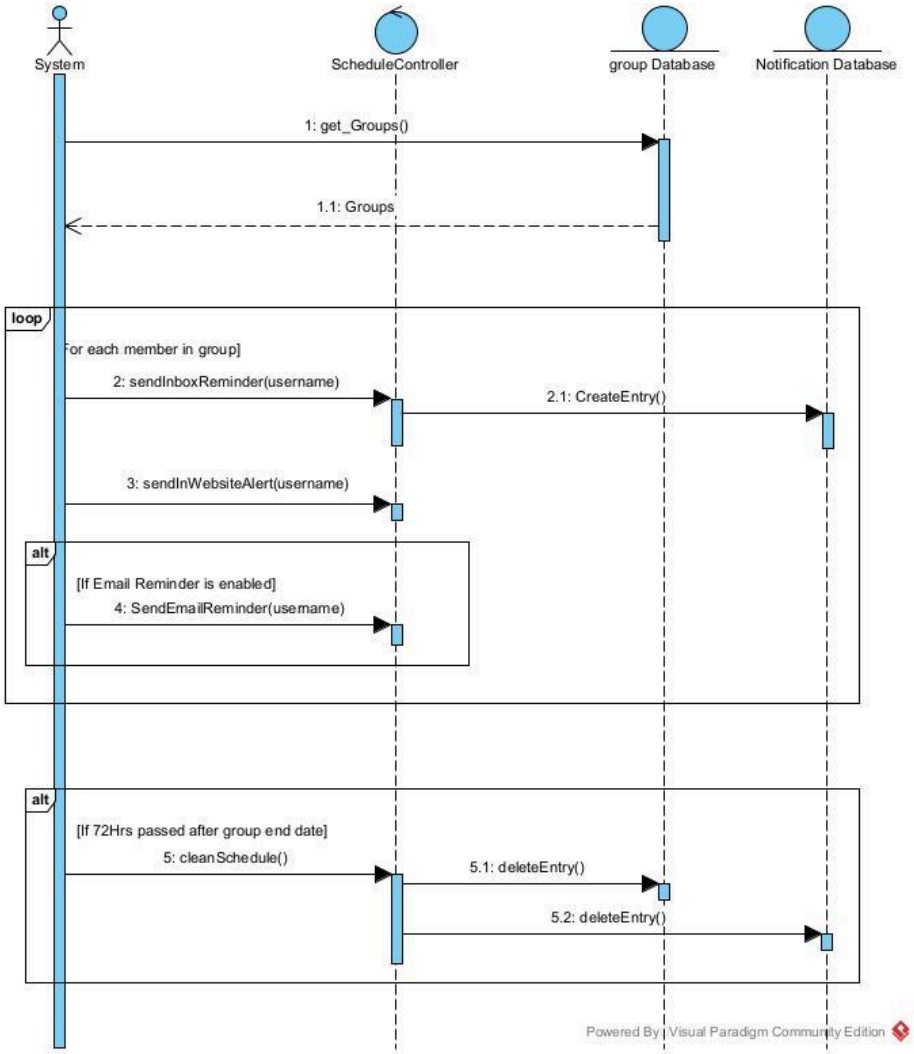
UC-18a: NotifyUsers (leave group)



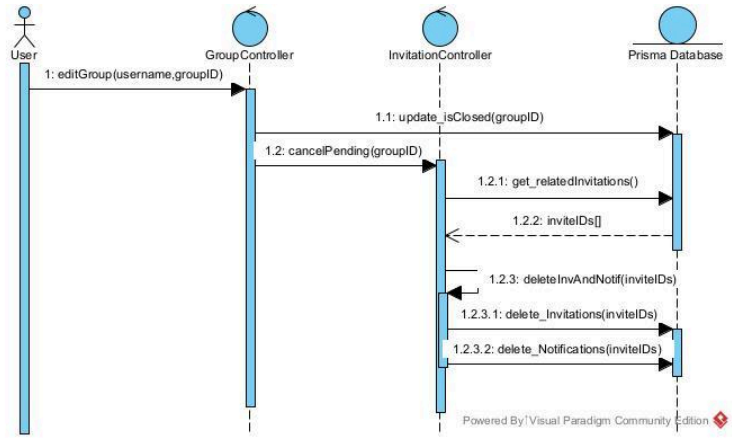
UC-20: ResolveConflict



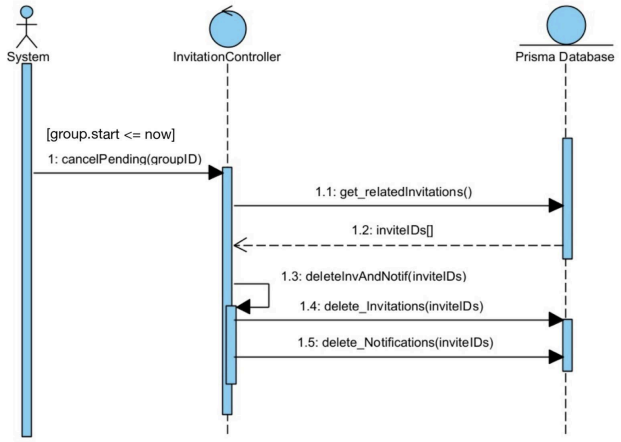
UC-21: ReminderNotificationDelivery



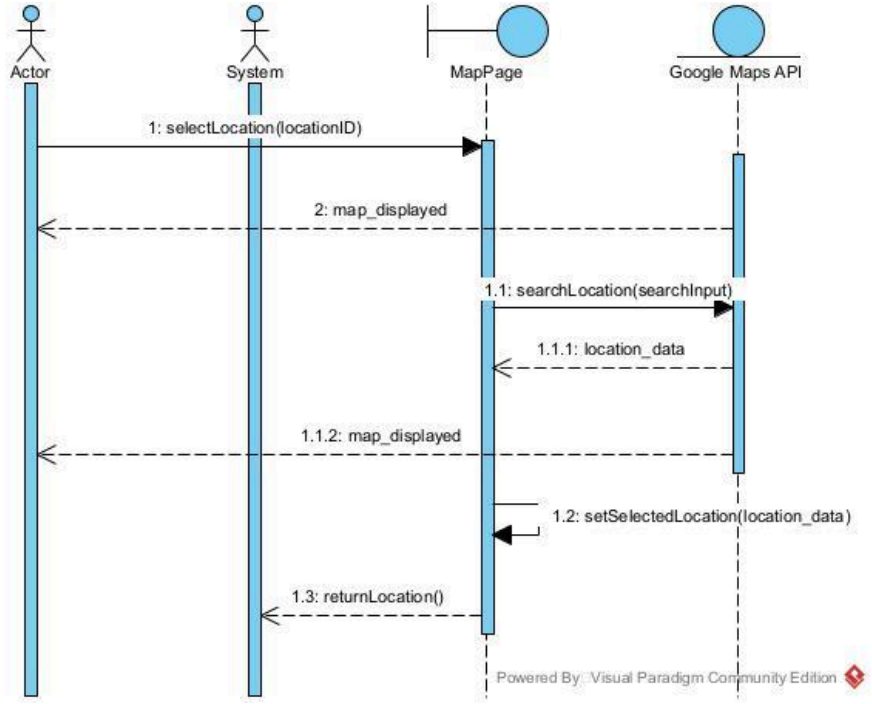
UC-24a: CloseExpiredGroups (close early)



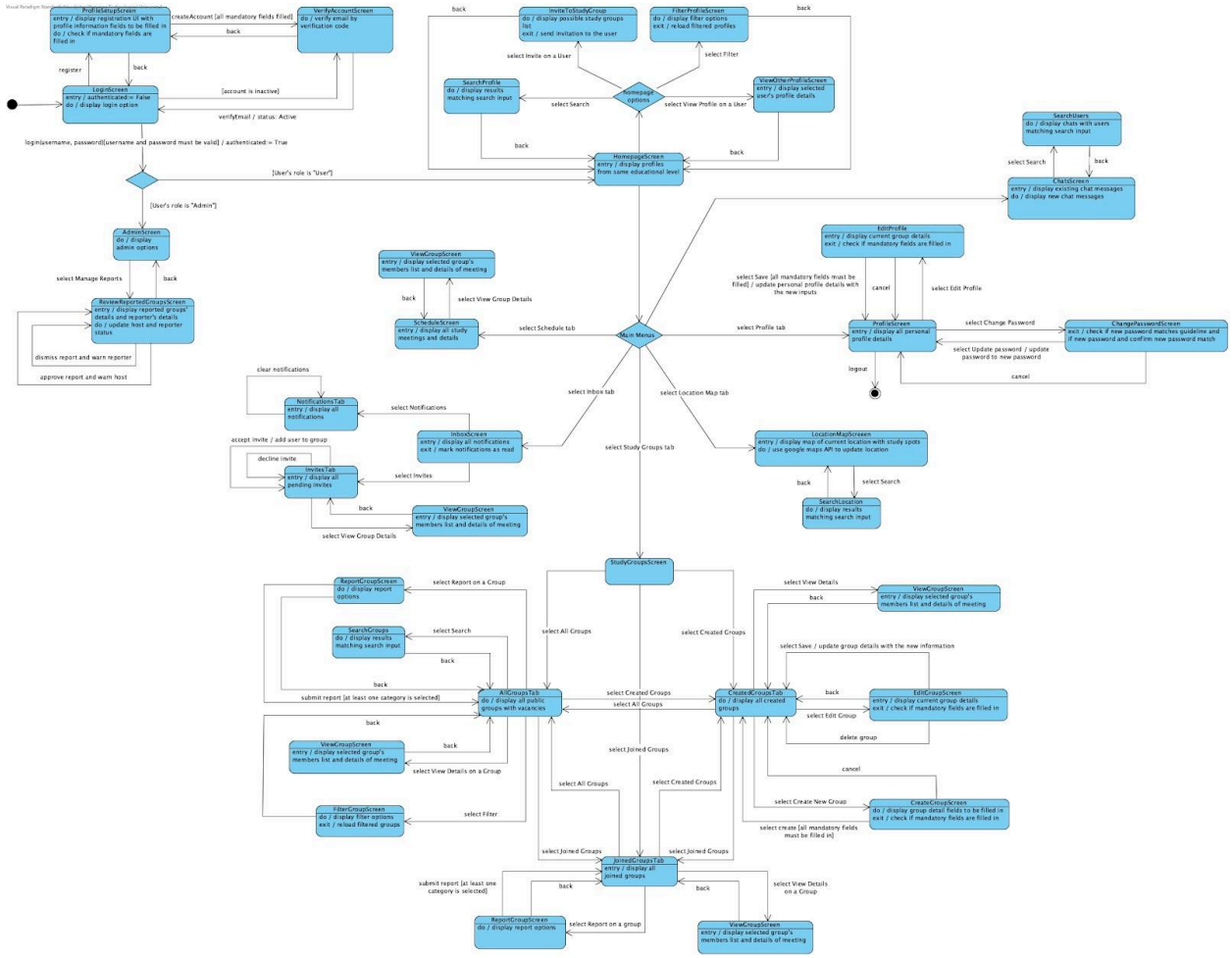
UC-24b: CloseExpiredGroups (expired)



UC-27: SelectStudyLocation

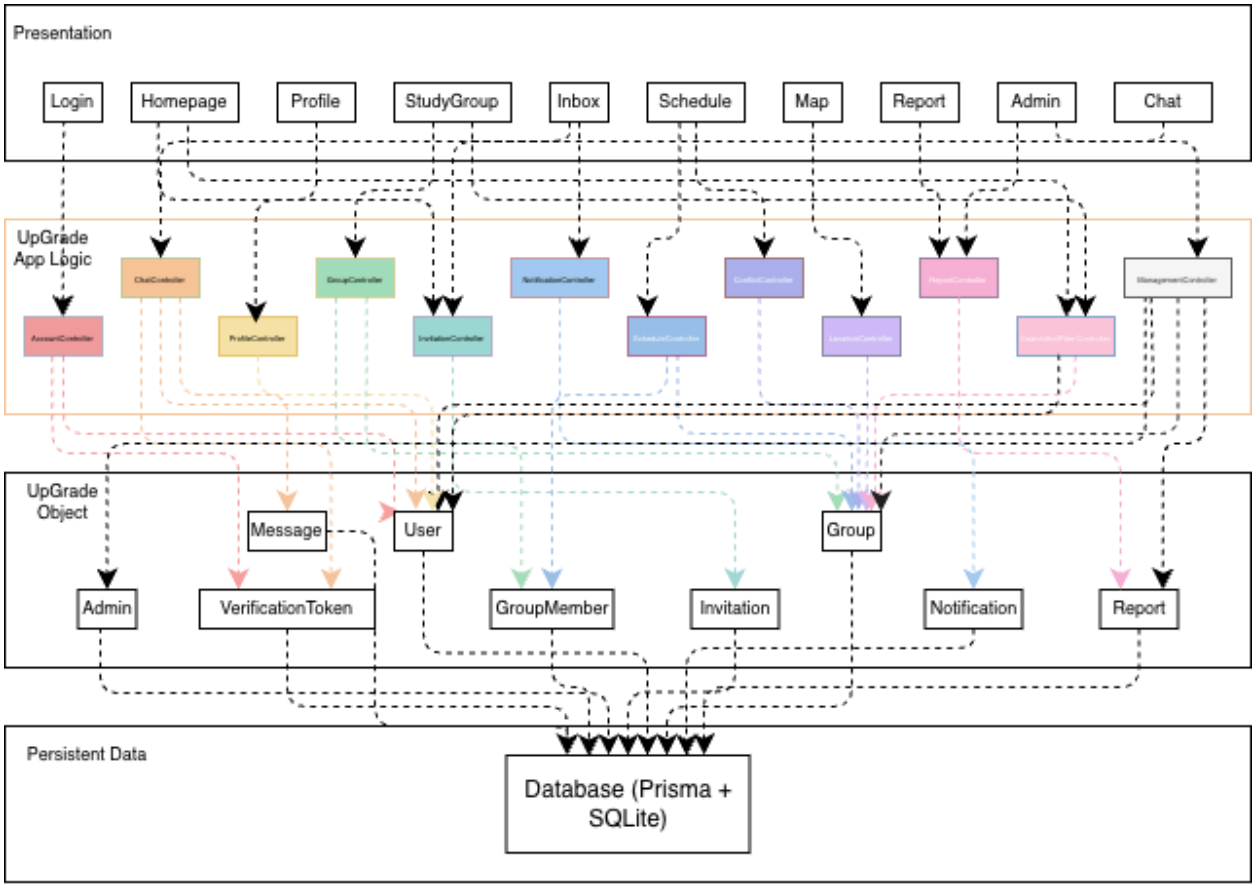


III. Dialog Map



(please refer to "dialog map".pdf file attached in the same folder for a clearer image)

8. System Architecture



(please refer to "system architecture diagram".pdf file attached in the same folder for a clearer image)

I. Presentation Layer

This layer is mainly responsible for the interaction between Users and UpGrade. The different UIs will then call for the respective controllers to run the App Logic. This layer consists of:

1. **Login** Allows Users to log in and register for accounts by using the services of AccountController and VerificationController.
2. **Homepage** Homepage will use the services of SearchAndFilterController and InvitationController to display user profiles and send invitations.
3. **Profile** Profile allows Users to view and edit their profile information by calling ProfileController.
4. **StudyGroup** StudyGroup allows Users to browse, create, and manage study groups by calling GroupController and SearchAndFilterController.
5. **Inbox** allows Users to view, accept, and reject study group invitations by calling InvitationController and NotificationController.
6. **Schedule** allows Users to view their study group calendar and detect time conflicts by calling ScheduleController and ConflictController.

7. **Map** Map allows Users to view study locations on an interactive map by calling LocationController.
8. **Report** Report allows Users to report inappropriate study groups by calling ReportController.
9. **Admin** is accessible only to administrators, allowing them to review reports and manage users/groups by calling ReportController and ManagementController.
10. **Navigation** Navigation provides routing across all pages and connects to all controllers.

II. App Logic Layer

This layer contains all the controller classes that will provide the presentation layer with its services. The controller classes will request for entities from the Object Layer if necessary to run its logic. This layer consists of:

1. **AccountController** AccountController is called by Login for Users to register and log in. Logic within this controller includes authentication, password validation, and session management.
2. **VerificationController** VerificationController is called by Login to handle email verification. Logic within this controller includes sending verification codes, validating codes, and activating accounts.
3. **ProfileController** ProfileController is called by Profile to handle user profile management. Logic within this controller includes viewing and updating profile information, toggling email notifications, and validating mandatory fields.
4. **GroupController** GroupController is called by StudyGroup to handle study group operations. Logic within this controller includes creating groups, closing groups, joining groups, leaving groups, and managing group capacity.
5. **InvitationController** InvitationController is called by Homepage and Inbox to handle study group invitations. Logic within this controller includes sending invitations, accepting invitations, rejecting invitations, and removing expired invitations.
6. **NotificationController** NotificationController is called by Inbox to handle user notifications. Logic within this controller includes displaying notifications, marking notifications as read, and sending notifications for group events (member joins/leaves, invitation rejections).
7. **ScheduleController** ScheduleController is called by Schedule to handle user's study group calendar. Logic within this controller includes displaying joined groups in calendar format, viewing group details, and cleaning up expired groups.
8. **ConflictController** ConflictController is called by Schedule to detect time conflicts. Logic within this controller includes checking for overlapping study group times and prompting users to resolve conflicts before accepting new invitations.
9. **LocationController** LocationController is called by Map to handle study location data. Logic within this controller includes displaying study locations on a map, filtering locations by region, and selecting locations when creating groups.

10. **ChatController** - ChatController is called by Chat to handle real-time messaging between study group members. Logic within this controller includes sending messages, retrieving message history, and managing group chat rooms.
11. **ReportController** ReportController is called by Report and Admin to handle group reports. Logic within this controller includes submitting reports, reviewing reports (admin), and tracking report types (vulgarities, harassment, spam, etc.).
12. **SearchAndFilterController** SearchAndFilterController is called by Homepage and StudyGroup to handle search and filter functionality. Logic within this controller includes searching for users by username, filtering users by profile attributes, searching for study groups, and filtering groups by date/time/location/subject.
13. **ManagementController** ManagementController is called by Admin to handle administrative actions. Logic within this controller includes banning users, warning users, removing groups, and managing user accounts.

III. App Object Layer

This layer contains the entity classes that will be called by the App Logic layer to implement its logic. The entity classes are stored in the database of the persistent layer. This layer consists of:

1. **User** User contains all information regarding the user. These include email, username, password hash, education level, year of study, gender, preferred timing, preferred locations, current course, relevant subjects, school, academic grades, usual study period, email reminder preference, warning status, and user status (INACTIVE, ACTIVE, BANNED).
2. **Admin** Admin contains admin user information including email, username, and password hash for administrative access.
3. **VerificationToken** VerificationToken contains the verification code, expiration time, and user reference for email verification during account activation.
4. **Group** Group contains study group information including group ID, name, visibility (public/private), meeting date and time (start and end), location, capacity, current size, host user, and closed status.
5. **GroupMember** GroupMember is a join table linking Users to Groups they have joined, containing the user reference, group reference, and join date.
6. **Invitation** Invitation contains invitation information including sender user, receiver user, group reference, and timestamps for when the invitation was created and updated.
7. **Message** - Message contains chat message information including sender user, group reference, message content, and timestamp for when the message was sent.
8. **Notification** contains notification information including user reference, optional group reference, notification type (INVITE_RECEIVED, GROUP_MEMBER_JOINED, etc.), message content, read status, and optional invitation reference.
9. **Report** Report contains report information including the reported group, reporting user, report types (stored as JSON array for multiple categories like vulgarities, harassment, spam), and creation timestamp.

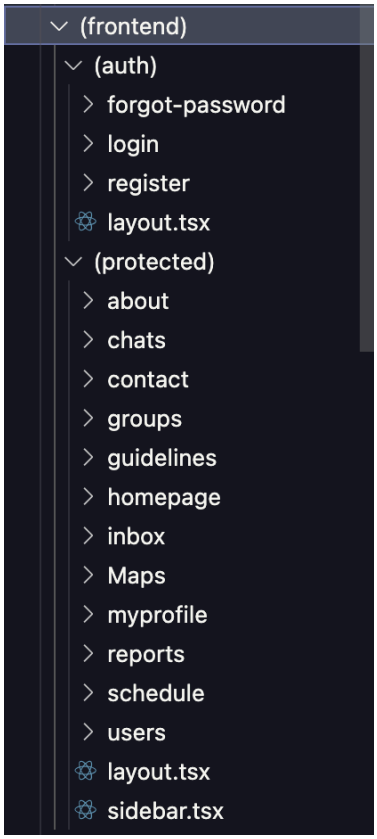
IV. Persistent Data Layer

This layer contains the database that will store all of the entities. UpGrade uses **Prisma ORM with SQLite** for development to persist all user data, study groups, invitations, notifications, and reports.

9. Application Skeleton

(please refer to the source code in the "source code" sub-folder in the same folder)

I. Frontend



The frontend (Next.js 15) mainly consists of different **User Interfaces (Screens)**, which are structured and organized into protected and public routes as depicted in the folder structure.

Protected Routes: Contains all authenticated user screens including groups, guidelines, homepage, inbox, Maps, myprofile, reports, schedule, and users. These pages require user authentication to access.

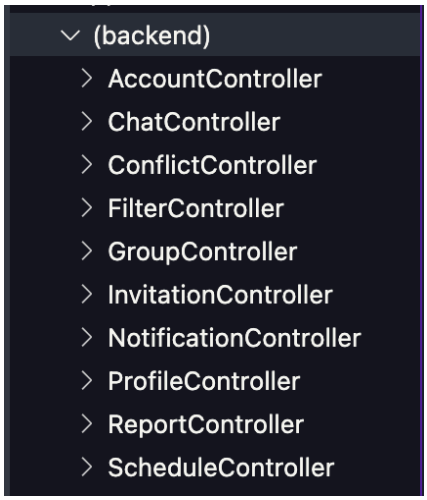
Public Routes: Contains the login page which serves as the entry point for unauthenticated users.

The **layout.tsx** is the root layout component that wraps all pages and provides consistent navigation and styling across the application.

The **page.tsx** serves as the entry point/landing page of the frontend application.

Other files such as **globals.css** contain global styling that makes the frontend code more organized and maintainable (as recommended by the Next.js framework).

II. Backend



Controllers: Contains all the controllers that provide an "abstracted interface" of the detailed implementation of the various services.

Lib (Library): Contains utility functions and helper modules:

a) jobs/: Background job processing

b) auth.ts: Authentication and authorization utilities

c) db.ts: Database connection and query utilities

d) emailer.ts: Email notification service

e) requireAdmin.ts: Middleware to protect admin-only routes

f) requireUser.ts: Middleware to protect authenticated routes

Prisma: Database management layer

a) **schema.prisma:** Defines the database schema with all models and relationships
b) **seed.ts:** Seeds the database with initial/test data
c) **dev.db:** SQLite development database file

Database Schema: Entry points to a database that stores all of the entities including Users, Groups, Sessions, Invitations, Notifications, and Reports with their relationships defined in the Prisma schema.

10. Test Cases and Testing Results

I. Black Box Testing

a. AccountController (chosen class)

The AccountController class manages User account information used during the entire session of when the user logs in until they log out, it also covers part of the registration, profile editing, resetting of password and Login

RESET PASSWORD
The AccountController is responsible for password resetting. When a User has forgotten their password, they are able to reset their password. The controller will first collect required information (Username, Email) of the account that needs to be reset. The controller retrieves the specific account. After retrieval, the controller will send a random verification code to the email. The user will have to enter this code before proceeding to resetting his password. Once a user confirms the new password, the password has to fit the requirements which will be checked by the controller. These requirements consist of at least 12 characters, 1 uppercase letter, 1 lowercase letter, 1 digit and 1 special character. The controller will then hash the new password and update the database and the user will be directed to the login screen where he can log in with the new password.

REGISTRATION/VERIFY EMAIL
The AccountController is also partially responsible for User registration. When a User registers for an account, the controller should validate the email that the user has entered during the registration. It will first confirm whether the email is a valid school email. After confirmation, the controller will send a randomly generated six digit verification code to that particular email. This code has a validation window where the user must enter it into the website within that window. Once the window/ExpirationDate has been reached, the code will no longer work and the user will have to acquire another verification code.

PROFILE EDITING
The AccountController is also responsible for user profile editing. Users are able to edit profile information, there are both essential or optional details that the User can change. The essential fields would be Year of study, Current Course, Preferred study timing, Preferred study locations. Meanwhile, the optional fields are Relevant subjects/modules, School/Institution, Academic grades/ CGPA, Email reminders and Usual study duration. The controller will check for the essential fields and prompt the user to fill in the field. If the user attempts to save their profile with an empty essential field, the controller will prevent them from proceeding. Else if there are no issues, the controller will update the database with the changes.

LOGIN
The AccountController is also responsible for user login and logout. On the login page, there are 3 fields, Sign in as User or Admin, Username or Email and Password. Once the User clicks Sign in, the controller will attempt to find the account in the database, if there is no entry matching either the username or email, or if the password does not match the password of the account, it will send an error. The controller will also check the account status (Active/ Inactive/ Banned), The controller redirects the user according to their status.

b. Equivalence Class and Boundary Value Testing

- Login Function
    - Valid Equivalence Class: Username/Email and Passwords exists
    - Invalid Equivalence Class: Username/Email and Passwords incorrect or any missing parameters
- Registration/ Verify Email
    - Valid Equivalence Class: Valid verification code and Email
    - Invalid Equivalence Class: Invalid verification code and Email or any missing parameters
- Reset Password
    - Valid Equivalence Class: Email and Username, Valid verification code, Password in correct format
    - Invalid Equivalence Class: Email and Username and Password in incorrect format, invalid verification code or any missing parameters
- Profile Editing
    - Valid Equivalence Class:

      Essential: Year of study, Current Course, Preferred study timing, Preferred study locations
      Optional: Relevant subjects/modules, School/Institution, Academic grades/ CGPA, Email reminders and Usual study duration.

    - Invalid Equivalence Class: Any missing essential parameters

c. Test Cases and Results

- Login
- Input parameters: Email/Username and Password

No.	Test Input	Expected Output	Actual Output	Pass?
1	**(Valid)** Email/Username: “josh@e.ntu.edu.sg” **(Valid)** Password: “Test1234567@”	Login Successful	Login Successful	Yes
2	**(Valid)** Email/Username: “josh” **(Valid)** Password: “Test1234567@”	Login Successful	Login Successful	Yes
3	**(Invalid)** Email/Username: “” **(Valid)** Password: “Test1234567@”	Login Unsuccessful Prompt: “Please fill out this field” on Email/Username box	Login Unsuccessful Prompt: “Please fill out this field” on Email/Username box	Yes
4	**(Valid)** Email/Username: “josh” **(Invalid)** Password: “”	Login Unsuccessful Prompt: “Please fill out this field” on Password box	Login Unsuccessful Prompt: “Please fill out this field” on Password box	Yes
5	**(Valid)** Email/Username: “josh” **(Valid)** Password: “Test1234567@”	Login Unsuccessful Prompt: Account Banned	Login Unsuccessful Prompt: Account Banned	Yes
6	**(Valid)** Email/Username: “josh” **(Valid)** Password: “Test1234567@”	Login Unsuccessful Prompt: “Please verify your email before logging in” Option to redirect to Verify-email page	Login Unsuccessful Prompt: “Please verify your email before logging in” Option to redirect to Verify-email page	Yes

- Registration/ Verify Email
- Input parameters: Email, Verification code

No.	Test Input	Expected Output	Actual Output	Pass?
1	**(Valid)** Email: “alice@e.ntu.edu.sg” **(Valid)** Verification code: “??????”	Account Verified	Account Verified	Yes
2	**(Invalid)** Email: “alice@banana.sg” **(Valid)** Verification code: “??????”	Account Unverified Prompt: Invalid email address	Account Unverified Prompt: Invalid email address	Yes
2	**(Invalid)** Email: “alice@” **(Valid)** Verification code: “??????”	Account Unverified Prompt: “Please enter a part following ‘@’. ‘alice@’ is incomplete	Account Unverified Prompt: “Please enter a part following ‘@’. ‘alice@’ is incomplete	Yes

3	**(Valid)** Email: “alice@e.ntu.edu.sg” **(Invalid)** Verification code: “??????”	Account Unverified Prompt: Invalid/expired code	Account Unverified Prompt: Invalid/expired code	Yes

- Reset Password
- Input parameters: Email, Username, Verification code, Password, Confirm Password

No.	Test Input	Expected Output	Actual Output	Pass?
1	**(All Valid Inputs)** Email: “josh@e.ntu.edu.sg” Username: “josh” Verification code: “??????” Password: “Test1234567@” Confirm Password: “Test1234567@”	Password resetted Prompt: Password successfully reset. Please login again.	Password resetted Prompt: Password successfully reset. Please login again	Yes
3	**(All Valid Inputs except Username)** Username: “”	No Password Reset Prompt: Please fill out this field	No Password Reset Prompt: Please fill out this field	Yes
5	**(All Valid Inputs except Email)** Email: “”	No Password Reset Prompt: Please fill out this field	No Password Reset Prompt: Please fill out this field	Yes
6	**(All Valid Inputs except Verification code)** Verification code: “??????”	No Password Reset Prompt: Invalid reset code	No Password Reset Prompt: Invalid reset code	Yes
7	**(All Valid Inputs except Password)** Password: “”	No Password Reset Prompt: Please fill out this field	No Password Reset Prompt: Please fill out this field	Yes
9	**(All Valid Inputs except Confirm Password)** Confirm Password: “”	No Password Reset Prompt: Please fill out this field	No Password Reset Prompt: Please fill out this field	Yes
10	**(All Valid Inputs except Password and Confirm Password)** Password: “Test1234567@” Confirm Password: “Test1234567@mismatch”	No Password Reset Prompt: Passwords do not match	No Password Reset Prompt: Passwords do not match	Yes
11	**(All Valid Inputs except Password)** Password: “Test1@”	No Password Reset Prompt: Password must be at least 12 characters	No Password Reset Prompt: Password must be at least 12 characters	Yes

12	**(All Valid Inputs except Password)** Password: “test1234567@”	No Password Reset Prompt: Must contain an uppercase letter	No Password Reset Prompt: Must contain an uppercase letter	Yes
13	**(All Valid Inputs except Password)** Password: “TEST1234567@”	No Password Reset Prompt: Must contain a lowercase letter	No Password Reset Prompt: Must contain a lowercase letter	Yes
14	**(All Valid Inputs except Password)** Password: “Test1234567”	No Password Reset Prompt: Must contain a special character	No Password Reset Prompt: Must contain a special character	Yes
15	**(All Valid Inputs except Password)** Password: “Testtesttesttest@”	No Password Reset Prompt: Must contain a digit	No Password Reset Prompt: Must contain a digit	Yes

- Profile editing
- Input parameters:

Essential: Year of study, Current Course, Preferred study timing, Preferred study locations
Optional: Relevant subjects/modules, School/Institution, Academic grades/ CGPA, Email reminders and Usual study duration.

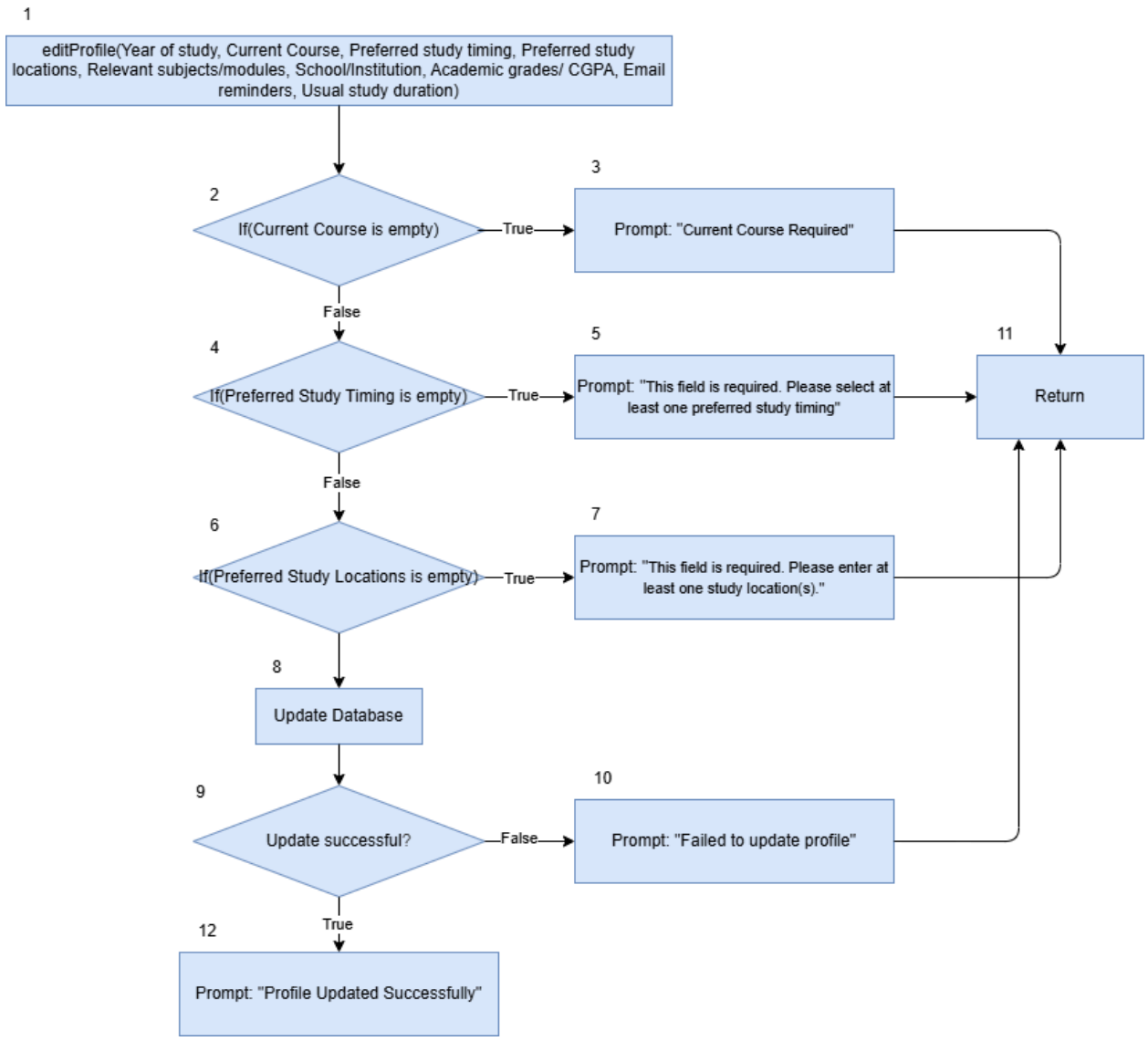
No.	Test Input	Expected Output	Actual Output	Pass?
1	**(All Valid Inputs)** Year of study: “U1” Current Course: “Computer Science” Preferred study timing: Morning Preferred study locations: “Woodlands” Optional—---------- Relevant subjects/modules: “” School/Institution: “” Academic grades/ CGPA: “” Email reminders: True Usual study duration: “”	Profile Updated Successfully	Profile Updated Successfully	Yes
2	**(All Valid Inputs except Current Course)** Current Course: “”	Profile Updated Unsuccessfully Prompt: Current Course Required	Profile Updated Unsuccessfully Prompt: Current Course Required	Yes
3	**(All Valid Inputs except Preferred Study Timing)** Preferred study timing: “”	Profile Updated Unsuccessfully Prompt: This field is required. Please select at least one preferred study timing.	Profile Updated Unsuccessfully Prompt: This field is required. Please select at least one preferred study timing.	Yes

4	**(All Valid Inputs except Preferred Study Timing)** Preferred study locations: “”	Profile Updated Unsuccessfully Prompt: This field is required. Please enter at least one study location(s).	Profile Updated Unsuccessfully Prompt: This field is required. Please enter at least one study location(s).	Yes

II. White Box Testing

c. Edit Profile (chosen function)

I. Flow Graph



II. Basis Path Testing

|DesicionPoint| = 4
CC = 4 + 1 = 5

<u>Basis Paths</u>

Baseline path: 1, 2, 4, 6, 8, 9, 12

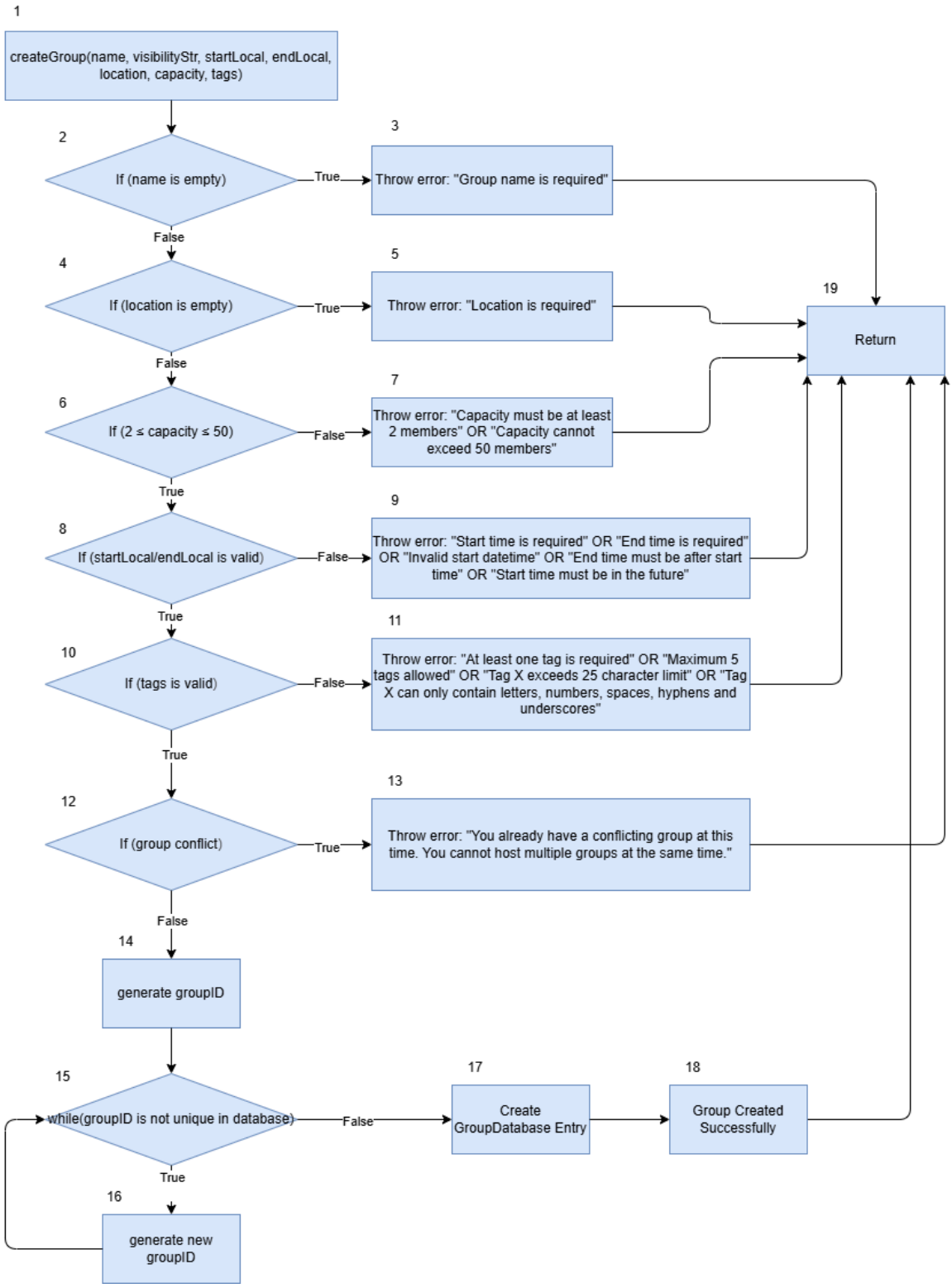
Basis path #1: 1, 2, 3, 11
Basis path #2: 1, 2, 4, 5, 11
Basis path #3: 1, 2, 4, 6, 7, 11
Basis path #4: 1, 2, 4, 6, 8, 9, 10, 11

III. Test Cases and Results

No.	Path	Test Input	Expected Output	Actual Output	Pass?
1.	Baseline Path	Current Course not empty, Preferred Study Timing not empty, Preferred Study Locations not empty	Profile Updated Successfully	Profile Updated Successfully	Yes
2.	Path #1	Current Course is empty, Preferred Study Timing not empty, Preferred Study Locations not empty	Profile Updated Unsuccessfully Prompt: Current Course Required	Profile Updated Unsuccessfully Prompt: Current Course Required	Yes
3.	Path #2	Current Course not empty, Preferred Study Timing is empty, Preferred Study Locations not empty	Profile Updated Unsuccessfully Prompt: This field is required. Please select at least one preferred study timing.	Profile Updated Unsuccessfully Prompt: This field is required. Please select at least one preferred study timing.	Yes
4.	Path #3	Current Course not empty, Preferred Study Timing not empty, Preferred Study Locations is empty	Profile Updated Unsuccessfully Prompt: This field is required. Please enter at least one study location(s)	Profile Updated Unsuccessfully Prompt: This field is required. Please enter at least one study location(s)	Yes
5.	Path #4	Current Course not empty, Preferred Study Timing not empty, Preferred Study Locations not empty	Profile Update Unsuccessfully Prompt: Failed to update profile	Profile Updated Unsuccessfully Prompt: Failed to update profile	Yes

d. Create Group

I. Flow Graph



II. Basis Path Testing

|DesicionPoint| = 7
CC = 7 + 1 = 8

<u>Basis Paths</u>

Baseline path: 1, 2, 4, 6, 8, 10, 12, 14, 15, 17, 18, 19
Basis path #1: 1, 2, 3, 19
Basis path #2: 1, 2, 4, 5, 19
Basis path #3: 1, 2, 4, 6, 7, 19
Basis path #4: 1, 2, 4, 6, 8, 9, 19
Basis path #5: 1, 2, 4, 6, 8, 10, 11, 19
Basis path #6: 1, 2, 4, 6, 8, 10, 12, 13, 19
Basis path #7: 1, 2, 4, 6, 8, 10, 12, 14, 15, 16, 15, 17, 18, 19

III. Test Cases and Results

No.	Path	Test Input	Expected Output	Actual Output	Pass?
1.	Baseline Path	name not empty, location not empty, capacity within 2 and 50, startLocal/endLocal is valid, tags are valid, groupconflict is false, groupID is unique	Group Created Successfully	Group Created Successfully	Yes
2.	Path #1	name is empty, location not empty, capacity within 2 and 50, startLocal/endLocal is valid, tags are valid, groupconflict is false, groupID is unique	Group Created Unsuccessfully Error thrown: Group name is required	Group Created Unsuccessfully Error thrown: Group name is required	Yes
3.	Path #2	name not empty, location is empty, capacity within 2 and 50, startLocal/endLocal is valid, tags are valid, groupconflict is false, groupID is unique	Group Created Unsuccessfully Error thrown: Location is required	Group Created Unsuccessfully Error thrown: Location is required	Yes
4.	Path #3	name not empty, location not empty, capacity is 1 / capacity is more than 51, startLocal/endLocal is valid, tags are valid, groupconflict is false, groupID is unique	Group Created Unsuccessfully Error thrown: Capacity must be at least 2 members / Capacity cannot exceed 50 members	Group Created Unsuccessfully Error thrown: Capacity must be at least 2 members / Capacity cannot exceed 50 members	Yes
5.	Path #4	name not empty, location not empty, capacity within 2 and 50, startLocal is empty / endLocal is empty / startLocal is invalid / endLocal < startLocal / startLocal < current time, tags are valid, groupconflict is false, groupID is unique	Group Created Unsuccessfully Error thrown: Start time is required / End time is required / Invalid start datetime / End time must be after start time / Start time must be in the future	Group Created Unsuccessfully Error thrown: Start time is required / End time is required / Invalid start datetime / End time must be after start time / Start time must be in the future	Yes
6.	Path #5	name not empty, location not empty, capacity within 2 and 50, startLocal/endLocal is valid, no tags given / 6 tags given / tag1 has more than 25 characters /	Group Created Unsuccessfully Error thrown: At least one tag is required /	Group Created Unsuccessfully Error thrown: At least one tag is required /	Yes

		tag1 contain special characters, groupconflict is false, groupID is unique	Maximum 5 tags allowed / Tag 1 exceeds 25 character limit / Tag 1 can only contain letters, numbers, spaces, hyphens and underscores	Maximum 5 tags allowed / Tag 1 exceeds 25 character limit / Tag 1 can only contain letters, numbers, spaces, hyphens and underscores	
7.	Path #6	name not empty, location not empty, capacity within 2 and 50, startLocal/endLocal is valid, tags are valid, groupconflict is true (exists another group created that has overlapping timing), groupID is unique	Group Created Unsuccessfully Error thrown: You already have a conflicting group this time. You cannot host multiple groups at the same time.	Group Created Unsuccessfully Error thrown: You already have a conflicting group this time. You cannot host multiple groups at the same time.	Yes
8.	Path #7	name not empty, location not empty, capacity within 2 and 50, startLocal/endLocal is valid, tags are valid, groupconflict is true (exists another group created that has overlapping timing), groupID generated is not unique	Group Created Successfully Actions: New unique group ID generated	Group Created Successfully Actions: New unique group ID generated	Yes

11. Website Demonstration Video

https://youtu.be/HZC9oiu_NFA?si=hcQ_sBqpjGpLh444

12. Live Presentation Script

Shreeya:

(slides)

Good afternoon, we are group 4 and today we are going to be presenting on our website “UpGrade”. So how did we come up with Upgrade? Well, as students, we realised that the most difficult aspect of studying as a group was the planning and organising of available times & dates. This caused us to waste quite some time as well. It was also difficult to find people with the same motivation. The ‘shyness’ factor also played a role in depriving one of joining a study group as they were afraid to make the approach. Hence we created Upgrade, a web-based platform that matches students with peers at their education level and streamlines group organization, so you spend less time planning and more time 'Locking-in' and ‘mugging’ ! For this website, we used next.js for our website, so typescript for our backend & typescript and jsx components for our frontend. This is our overall use case diagram, where we have 4 actors; the user, admin, map and system, and the lines represent the respective relationships & associations. Next we will be doing a live demo of how UpGrade works.

(live demo)

So we will run the following commands in the terminal to open the localhost, which will bring us to the login page. Over here, I will be registering as shreeya123, which is my username. I would be using my personal email as of now. So I will fill up the remaining details and click ‘create account’. Once this button is clicked, the registered email should receive an otp verification code. If a random or wrong code is typed it will not allow me to register. If for some reason, the

code does not appear in the email, I can always request for the code to be resent. If I login without verifying, the website does not allow me to login until the correct code is typed. I can then login with my newly created account. The default page after logging in would be this homepage. We will now log out. If I already have an existing account in Upgrade and I do not remember my password, I can click 'forgot password' which will require me to type my username and email and a reset code will be sent to my email.

Joshua:

This is our profile page, where the system will display User profile information. Eg. Email, Gender, etc. This page has 2 features. First would be the Edit Profile feature that enables users to conveniently update their personal information just by clicking the "Edit profile" button, such as Current Course Preferred study timing, etc. Existing details are pre-filled, allowing quick edits without re-entering all data. Details marked with an asterisk are compulsory and the system will flag an error if empty. Once the user saves the changes, the system updates the database and refreshes the profile view to reflect the new details.

For the second feature, the system allows users to change their passwords. When the user clicks the "Change Password" button. To improve usability and reduce input errors, it includes password visibility toggles, Caps Lock detection, and a checklist that validates password strength as the user types. The "Update Password" button remains disabled until all conditions are met and both new password fields match, ensuring strong, valid inputs. This checklist ensures the new password is at least 12 characters long and contains an uppercase letter, a lowercase letter, a number, and a special character, with a bar that visually indicates progress. Upon submission, the system will verify the current password, and securely update the new one using bcrypt hashing in the database. If successful, a confirmation message is shown; otherwise, clear error feedback is provided such as incorrect current password.

Moving on to our Location Map page, using google maps API, it is designed to help users easily find and select a study location for creating a group. Users can either search for any locations in Singapore, click directly on the map to drop a pin, or simply select a library from the displayed list; each library will also appear as a blue marker on the map. Users can then click on any library to choose it as their preferred study location. These library locations are sourced from NLB through official gov.sg open data, ensuring that all library information is accurate and up to date.

When a location is searched or clicked on the map, nearby libraries are automatically updated and shown with their distance from that point. — making it simple and intuitive to find locations for forming study groups.

Once the user has selected a location, they can click on the "Use This Location" button which will redirect the user to the create group page with that location.

Ming Xuan:

Now, we'll be demonstrating how users can create a study group and how the filter and search functions make it easier to find suitable groups. After navigating to the study groups page we can see that there are 3 different tabs, All groups, Created groups & Joined groups. The All groups tab shows every public group that is currently open for users to join. The Created

Groups tab shows all the groups that are hosted by the user. And the Joined Groups tab shows all the groups that the user has joined.

Let's go through the process of creating a new group.
The red asterisks signifies the fields that are mandatory to be filled up. Let's name our group and set the visibility to public. A Main Tag is required to help other users to quickly identify the Main subject/theme of the Study Group. Users can add in additional optional tags which allows for them to further specify what kind of dynamic/vibes they want for the group to have, it would also make the group more identifiable via the search bar.
Next we would also need to specify the start date/time and the end date/time and for demonstration purposes we shall enter a start time in the past to show the error handling. For the location, we can either type in the general location, or we can navigate to the location map to select a location from the map. Lastly we would also need to specify the group capacity. As you can see, it says that our start time must be in the future. And to demonstrate our cron scheduler, we are going to pick a start time that is 2 hours later.

Now lets look at the filter in action, the user can specify the date range. Let's choose the start and end time of the group we've just created, and we can also leave the end time empty to show all the groups that start at the start time to the future. We can also click reset to remove all the filters.

For the search bar, the user can search groups by their group names, main tags, optional tags and their locations. In the Created groups tab, the user is able to edit the details of their group based by clicking on the Edit Group button which allows them to pretty much change whatever options that are available during the create group process.

Now let's move on to the schedule tab, where we can see our planned/upcoming study group sessions. There are two ways of viewing the schedule, either in monthly view or weekly view. We can also click on the view details to see a quick summary of the study group, upon which we would also be able to click View Full Page to see more specific details of the group.

Lastly, let's see our inbox. For demonstration purposes, we have set our scheduler to run a check every 1 minute, so there will be a notification reminder showing the group that is starting every time the scheduler is run. The scheduler also sends an email notification reminder to the user's email.

Aurellia:
So as we are currently logged in as [shreeya123], let’s try to join a group, [Math Revision], once we click on its details, we are allowed to send a message to the host. Now lets test out the chat function, we will try to chat with the host, [Alice]. For example, we can write to the host “Where is the venue?” then click the [send] button to deliver. As you can see, as this user is the last one to join and the member slot is filled up, the group is indicated as full and will no longer show up in the study groups page. So this will be updated dynamically, if [shreeya] tries to leave the group [click leave group], this group [Group Name] will be displayed back to the home page.

Moving on to homepage, [click filter], let's try to fill in [Year: year 2, gender: female, preferred study timing: morning], the respective user will show up that fulfills all the filtering criteria. In this case user Mary is displayed. Going back to all available users in the homepage, now we will try to invite [Alice] and choose which group we can invite the user into. We will do the same with [Sarah]. In the case we want to message [Sarah], what is needed is to just click on the message button and it will redirect us to the chats screen where we can just send a text message. For example we send "Hi, this is [shreeya]" and an immediate text "Do you wanna study?" This will allow the timing to be displayed at a 5 minute interval only and clean the user interface. If we want to delete one of the messages, all you need to do is to click the three dots beside the message and press the [Delete] button and the message will be deleted.

Shannon:

Now, we log in as [Alice], and we can view the message sent by [shreeya123] previously. As shown, our website allows users to engage in private one-to-one chats, and a chat can be started by messaging another user via their profile in the homepage or by messaging the host of a public group through the group page. The host of private groups can also be messaged by an invited user, when they view group details of the invitation in their Inbox, and hosts of the groups can message their group members as well. In [Alice]'s inbox, join group notification and leave group notification by [shreeya123] are received as all members of a group are notified when a user joins or leaves the group. The invitation by [shreeya123] to join [Cool Group] is also displayed, and in this case, we can choose to accept this invitation.

Next, we log in as [Sarah], and we can view the message sent by [shreeya123], but not the deleted message. We can also send back a reply to [shreeya123]. The inbox of [Sarah] also displays the invitation, and we can try to accept the invitation here. However, as [Sarah] is already in a group [Programming Workshop] whose timing clashes with [Cool Group], we have to decide whether to stay in the previous group or to join [Cool Group]. Here, we can settle the conflict by leaving [Programming Workshop] and joining [Cool Group]. However, since Sarah is the host of [Programming workshop], she is unable to leave the group, and has to delete this group to join [Cool Group]. So we will reject the invitation here.

Finally, we log back in as [shreeya123], where we can view the replies sent by [Sarah]. In the inbox, we can also view the reject notification from [Sarah] as the inviter is notified when the invitee rejects their invite, and the join group notification of [Alice].

Chloe:

[Group A is a real report, which means host is at fault]
[Group B is a fake report, which means reporter is at fault]

[Signed in as shreeya123] We will try to report Programming Workshop with 'spam' as the reason, and Math Revision with 'vulgar language and misinformation' as the reasons. The reports are sent, so we will try to login as admin1. We can see in the admin dashboard that there are 2 pending reports to be reviewed.

We will now look at the report for Programming Workshop. The details of the group, host, and reporter are shown. After reviewing the details, if the report is legitimate, we can select delete

group and this will target the user action to the host. We will warn the host sarah. [Sign in as sarah_56] In sarah's account, we can see a warning notification in the inbox and also a warning sign in the profile. The group that was reported, Programming Workshop is also now not in the study groups page anymore.
[Sign in to admin1] We will now look at the report for Math Revision. If the report is fake, we can select keep group and this will target the user action to the reporter. We will warn the reporter, shreeya123 of faking a report. [Sign in as shreeya123], we can also see a warning in the inbox and the profile. Now if this user shreeya123 tries to report the same group, Math Revision, again, for example with 'harmful acts' as the reason, then admin can ban this user. [Sign in as admin1] In the report for Math Revision, we will select keep group and ban the reporter. Now if the banned user (shreeya123) tries to sign in, [Sign in as shreeya123] they will not be able to access our website and will go to a banned page instead.

Shannon:
(slides)
Last but not least, we will be explaining a few software engineering practices we have implemented in our project, and discussing future updates that can be added to our website.
Firstly, Single Responsibility Principle. Each function implemented has a sole responsibility. For example, acceptInvite() and rejectInvite() under the InvitationController class do exactly one thing each – to accept or reject an invitation to join a group. This is better than combining them into a single function like "manageInvite()", ensuring cleaner code, easier testing and better code reusability.
Next, Encapsulation. The Prisma SQLite database is not directly manipulated, and any access to the database is through functions such as editProfile() to access the User model or updateGroup() to access the Group model in Prisma. This reduces coupling between different architecture layers.
Thirdly, Enumeration. We use enums in the Prisma database, such as UserType and EducationLevel to enforce consistent and safe data. This reduces invalid or inconsistent states compared to using strings, ensuring consistency and fewer runtime errors.
Then, Layered Architecture. There is a clear layering between app logic, user interface and persistent data layer. The frontend user interface calls backend functions, which in turn read and write entity class values stored in the database.
Finally, Modularity. We have one control class for each user interface. For example, the ChatController class encapsulates all chat logic for the Chat page, and each function in the class has a specific sole purpose. sendMessage() creates a new Message object while viewSelectedChat() retrieves all messages between 2 users from the database. This improves maintainability and simplifies debugging, and enhances extensibility for future development and scalability.

As such, given more time to work on this project, we plan to include group chats by leveraging its extensible architecture. The existing one-to-one messaging module can be generalised to support multiple participants by introducing a new GroupChat model to link to multiple Users. The current chat functions can be easily adapted with minimal changes to database logic. On the frontend, chat components and modals can be extended to display messages from multiple

users within the same chat interface. This shows our project's extensibility to integrate new features without disrupting existing modules. We also hope to collaborate with Singapore schools and universities to validate the legitimacy of school emails used to register on our website, to ensure the safety of minor students (under the age of 18) such that only those in similar age groups can meet up through our website to study together.

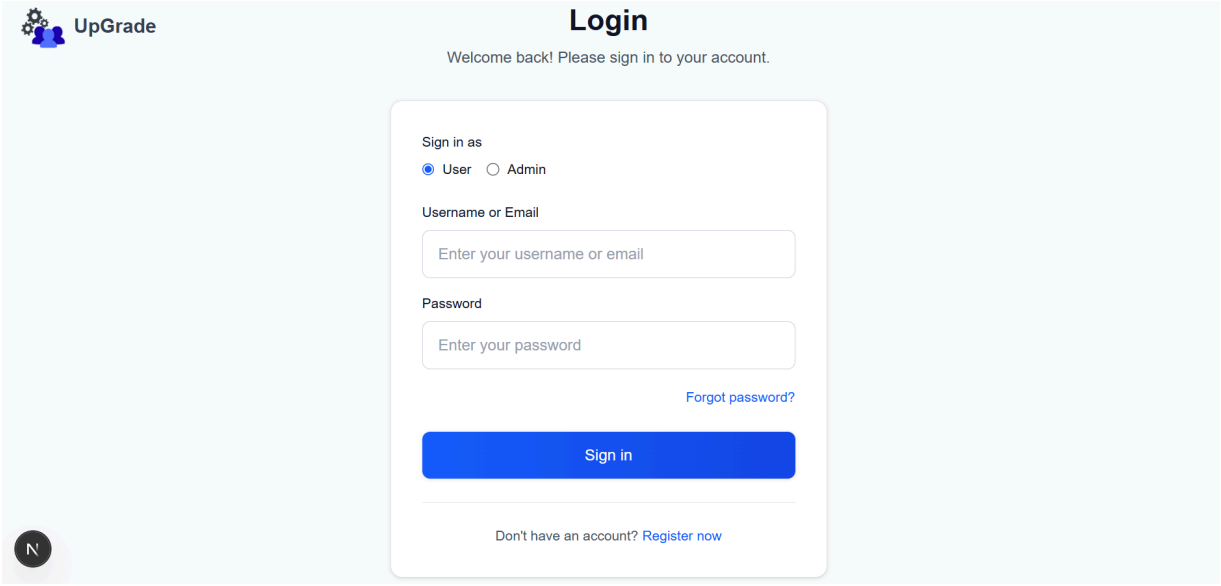
We also plan on giving admins more functions to manage user accounts, as currently user accounts are permanently banned. We can allow users to email admins to request for their account ban to be lifted, and allow admins to permit the request via the admin page in the website. Furthermore, we are also considering adding more study locations in our maps, as we currently only mark libraries on our map. These locations can include cafes, as well as any other shared spaces that allows students to study at, enabling students to utilise these areas for studying.

Thank you.

13. Presentation Slides

https://docs.google.com/presentation/d/1VOmONyVIOlM-6wZs3VPh_oW5EXzV25ZHslo0NQoNFoA/edit?slide=id.g398875277bf_2_307#slide=id.g398875277bf_2_307

14. WebApp UI



UpGrade

Signed in as **john** Logout

- Home
- Study groups
- Chats
- Inbox
- Schedule
- Location Map
- Profile

All profiles

Filter

Search by username...

alice ♀
Year 1
Morning, Evening
Computer Science
Message + Invite
View Profile ›

josh ♀
Year 1
Morning, Evening
Computer Science
Message + Invite
View Profile ›

mary ♀
Year 2
Morning
Data Science and AI
Message + Invite
View Profile ›

UpGrade

Signed in as **john** Logout

- Home
- Study groups
- Chats
- Inbox
- Schedule
- Location Map
- Profile

All groups

Filter

Search groups...

All groups | Created groups | Joined groups

Math Revision #Mathematics
15/12/2025, 6:00:00 pm • Library Room 101
view details 1/5 members Join report

UpGrade

Signed in as **john** Logout

- Home
- Study groups
- Chats
- Inbox
- Schedule
- Location Map
- Profile

Joined groups

Filter

Search groups...

All groups | Created groups | Joined groups

You haven't joined any groups yet

Explore available study groups and join ones that match your interests and schedule.

Browse All Groups | Create Your Own Group

UpGrade

Signed in as **john** Logout

- Home
- Study groups
- Chats
- Inbox
- Schedule
- Location Map
- Profile

Search users...

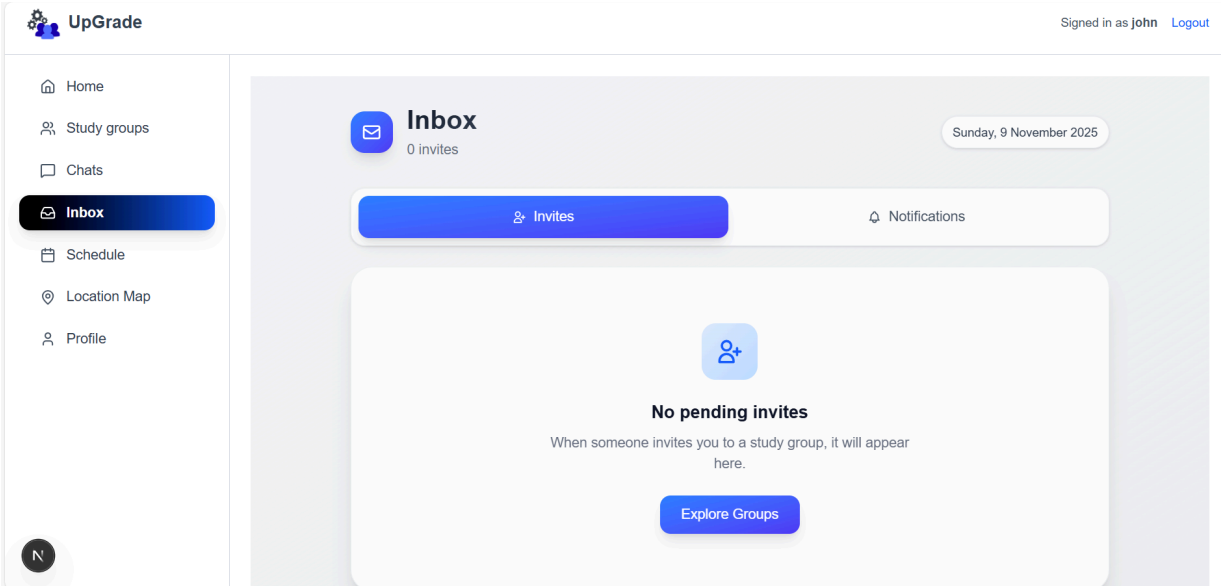
MESSAGES

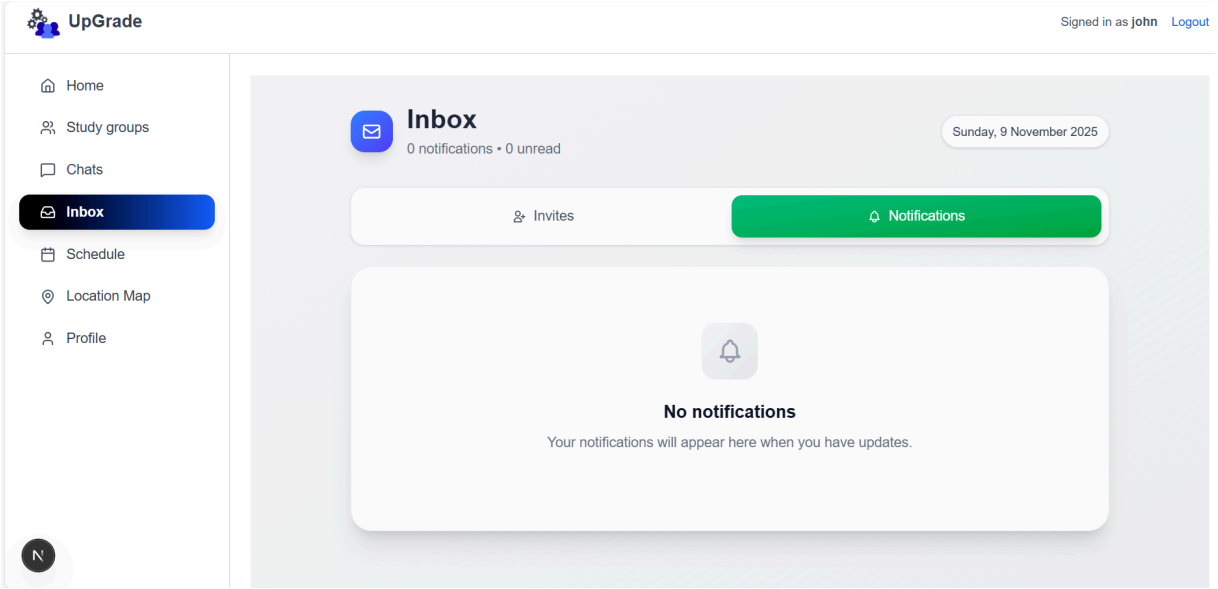
No chats yet

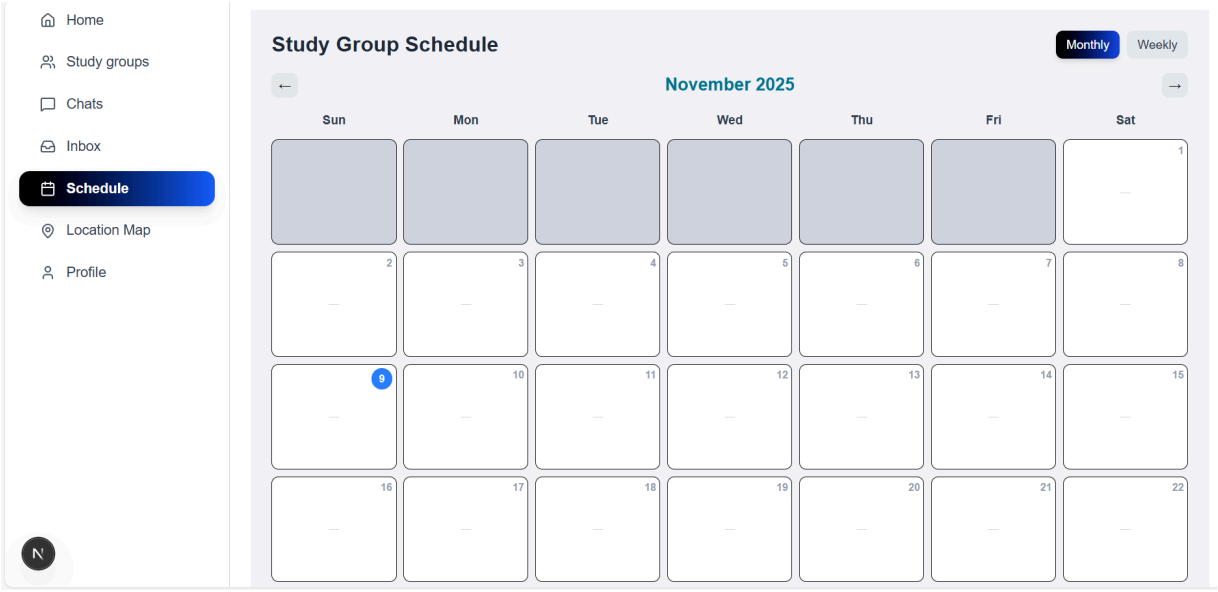
Search for users to start chatting

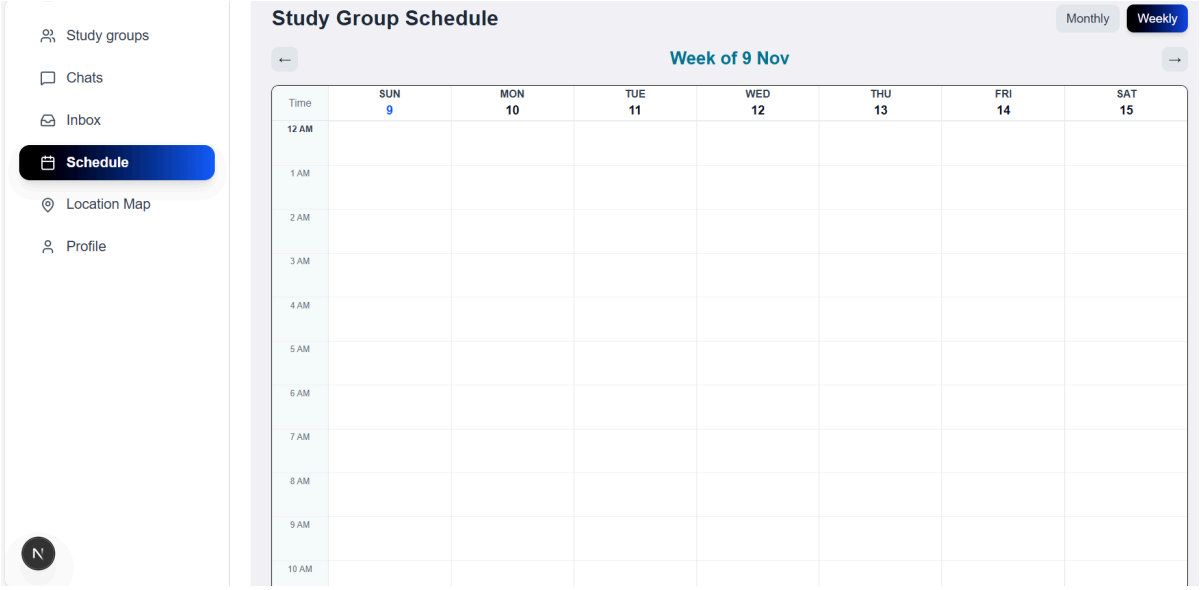
UpGrade Messenger

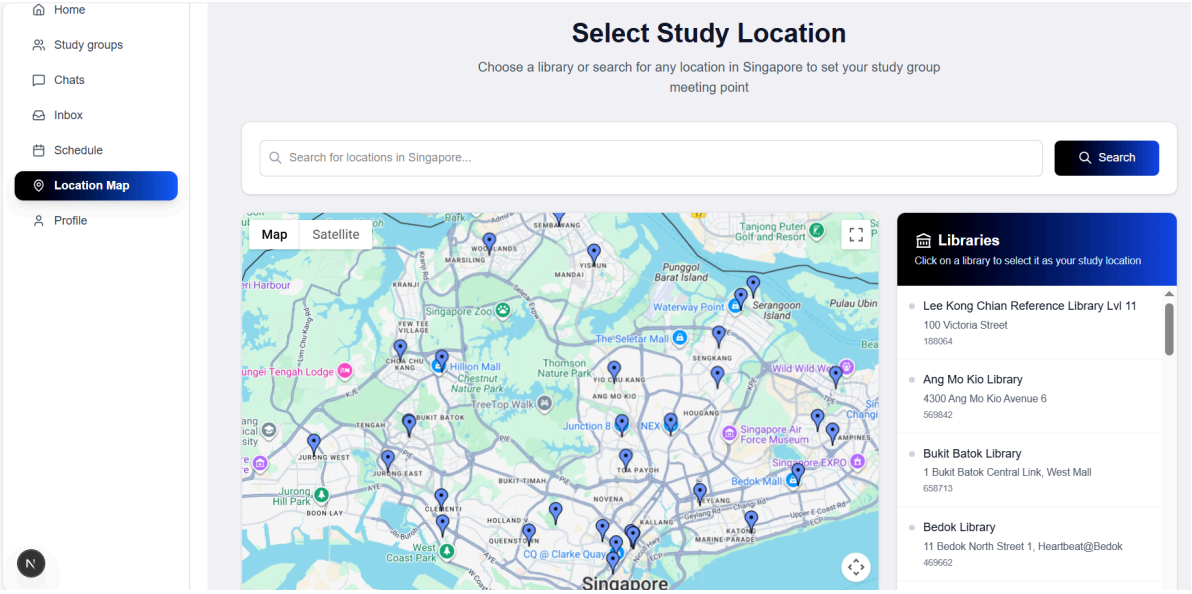
Select a chat from the sidebar to start messaging

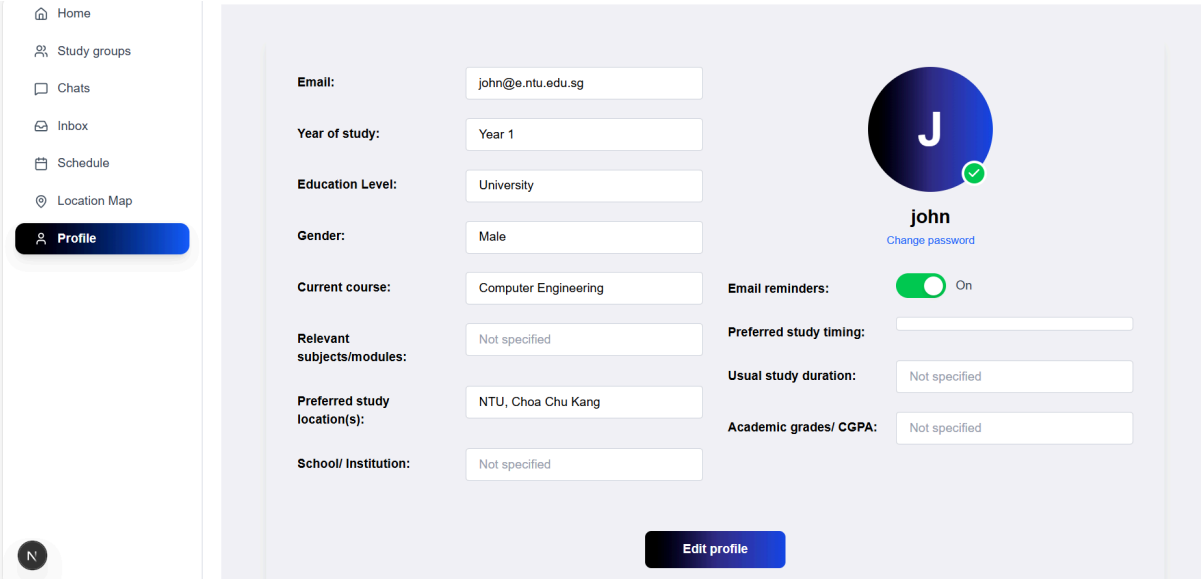

UpGrade
Signed in as john
Logout
Home
Study groups
Chats
Inbox
Schedule
Location Map
Profile
Inbox
0 invites
Sunday, 9 November 2025
Invites
Notifications
No pending invites
When someone invites you to a study group, it will appear here.
Explore Groups

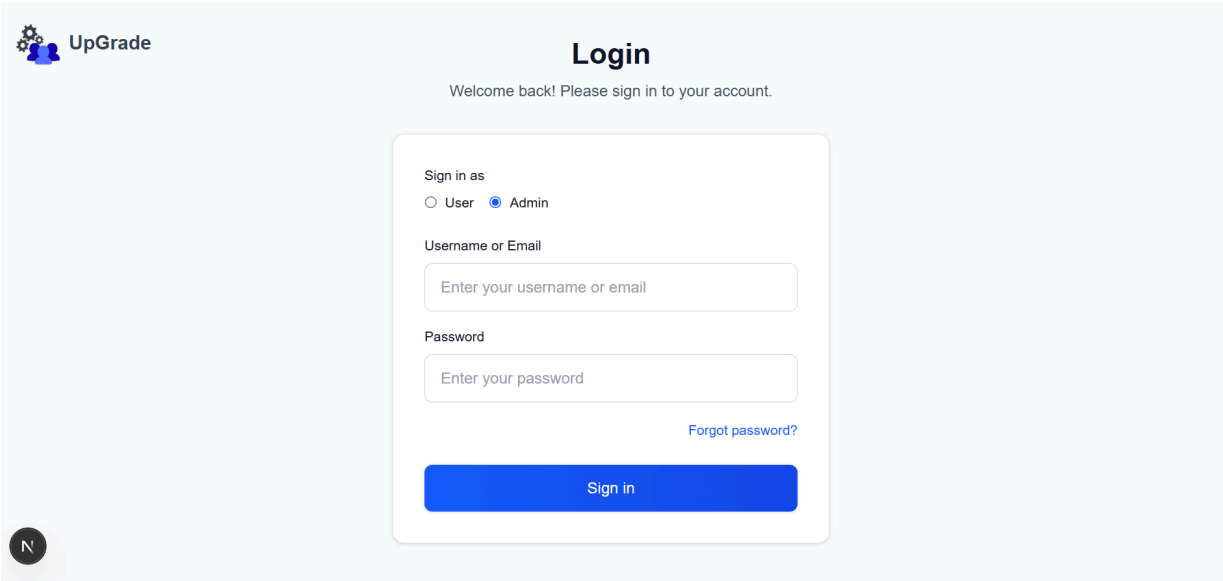

UpGrade
Signed in as john
Logout
Home
Study groups
Chats
Inbox
Schedule
Location Map
Profile
Inbox
0 notifications • 0 unread
Sunday, 9 November 2025
Invites
Notifications
No notifications
Your notifications will appear here when you have updates.

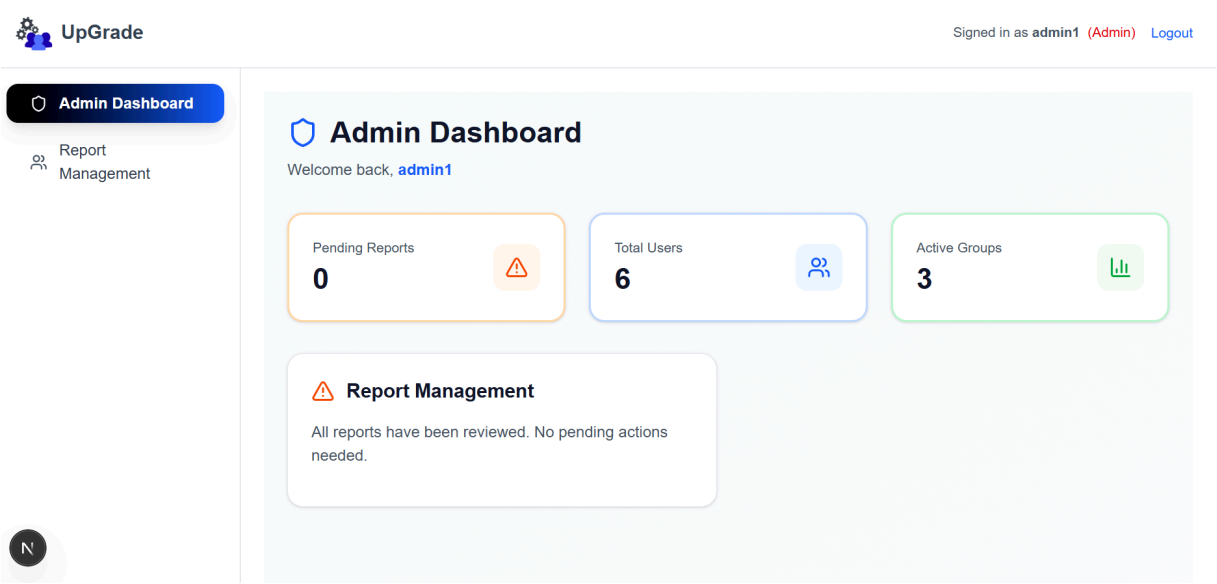

Home
Study groups
Chats
Inbox
Schedule
Location Map
Profile
Study Group Schedule
Monthly
Weekly
November 2025
Sun
Mon
Tue
Wed
Thu
Fri
Sat


Study groups
Chats
Inbox
Schedule
Location Map
Profile
Study Group Schedule
Monthly
Weekly
Week of 9 Nov
Time
SUN 9
MON 10
TUE 11
WED 12
THU 13
FRI 14
SAT 15
12 AM
1 AM
2 AM
3 AM
4 AM
5 AM
6 AM
7 AM
8 AM
9 AM
10 AM


Home
Study groups
Chats
Inbox
Schedule
Location Map
Profile
Select Study Location
Choose a library or search for any location in Singapore to set your study group meeting point
Search for locations in Singapore...
Search
Map
Satellite
Libraries
Click on a library to select it as your study location
Lee Kong Chian Reference Library Lvl 11
100 Victoria Street
188064
Ang Mo Kio Library
4300 Ang Mo Kio Avenue 6
569842
Bukit Batok Library
1 Bukit Batok Central Link, West Mall
658713
Bedok Library
11 Bedok North Street 1, Heartbeat@Bedok
469662


Home
Study groups
Chats
Inbox
Schedule
Location Map
Profile
Email:
john@e.ntu.edu.sg
Year of study:
Year 1
Education Level:
University
Gender:
Male
Current course:
Computer Engineering
Relevant subjects/modules:
Not specified
Preferred study location(s):
NTU, Choa Chu Kang
School/ Institution:
Not specified
J
john
Change password
Email reminders:
On
Preferred study timing:
Usual study duration:
Not specified
Academic grades/ CGPA:
Not specified
Edit profile


UpGrade
Login
Welcome back! Please sign in to your account.
Sign in as
User
Admin
Username or Email
Enter your username or email
Password
Enter your password
Forgot password?
Sign in


UpGrade
Signed in as admin1 (Admin) Logout
Admin Dashboard
Report Management
Admin Dashboard
Welcome back, admin1
Pending Reports
0
Total Users
6
Active Groups
3
Report Management
All reports have been reviewed. No pending actions needed.

UpGrade

Signed in as **admin1** (Admin) Logout

Admin Dashboard

Report Management

Report Management

0 pending reports to review

All Clear!

No pending reports to review.