

System Design Document

Team Members

Kosisochi Amobi-Oleka, Kai Dalen, Uroosa Imtiaz, Justin McDonald, Andrew Kim, Sophia Kim,
Aaron Ouyang, Jackson Reid, Andrew Ternopolsky

TA
Jihyeon Park

Nov 14, 2025

Introduction

Purpose of the System

Clubr is a web-based application designed to unify the fragmented communication channels between Queen's University students and campus clubs. At present, club-related information is distributed across social media, email, and static directories like the AMS Clubs page, making it difficult for students to discover and engage with organizations that interest them. The purpose of Clubr is to centralize these interactions into a single, interactive platform that allows students to explore clubs, follow updates, view upcoming events, and directly message club administrators. For club executives, Clubr provides tools to manage their pages, create posts and events, and collaborate with other clubs. Through this centralized system, Clubr aims to bridge the gap between static listings and the dynamic, real-time engagement needs of the university community.

Performance

The Clubr platform is designed for fast and responsive operation within a web browser. Page transitions such as navigating from the Home page to Discovery or Messages should occur within 1–2 seconds. Searching for clubs, following them, or posting updates must also return results promptly to create a fluid user experience. Notifications for new posts or events are expected to propagate to followers' feeds within one second. The system should maintain efficient database queries and light client-side rendering to support smooth usage across both desktop and mobile devices.

Dependability

The platform must support concurrent user interactions without compromising performance or data integrity. Validation checks will ensure that all submitted data (posts, events, or messages) is properly formatted and secure. Since authentication is mandatory for most features, OAuth2 and SSO integration will ensure safe user verification. Unit and integration testing will confirm that the system behaves consistently with expected outcomes across different browsers and devices.

Cost

Clubr will be developed using free, open-source web technologies. Deployment and hosting costs will be minimal during the prototype phase, with the potential for scalable cloud hosting in the future. Development costs are limited to time and effort, as no paid dependencies or hardware components are required.

Maintenance

The system is designed with maintainability in mind through modular decomposition. Each major feature exists in its own subsystem, allowing for isolated updates or replacements. This modular design allows future development to easily add new capabilities without disrupting

existing functions. Shared validation and role-permission modules ensure centralized control over user access, simplifying future maintenance and updates.

End User Criteria

End users should experience a clear, intuitive interface. Students should immediately see a personalized feed of posts and events upon login, while club administrators should have a visible indicator when operating in “Admin Mode.” The interface should remain visually consistent across pages and devices, maintaining accessibility and responsiveness. Holistically Clubr should provide an intuitive and easy way to interact and discover clubs in the university space.

Definitions & Acronyms

Admin: Users who have been granted permission by another Admin to edit fields of a specific Club Page.

AI: Artificial Intelligence.

AMS: Alma Mater Society, specifically at Queen’s University. They are the governing body for student clubs.

API: Application Programming Interface. Set of rules and protocols that allow different software applications to communicate with each other.

Club: An AMS ratified student group.

DB: Database.

DM: Direct message.

OAuth2: OAuth 2.0. Authorization framework that allows users to grant third-party applications limited access to their resources to their credentials.

SSO: Single sign-on. Authentication method allowing users to sign on to multiple platforms with a single set of credentials.

User: Any student registered on our platform.

Use Cases & Use Case Diagrams

Use Case #1 Name: FollowClub

Participating Actors: USER

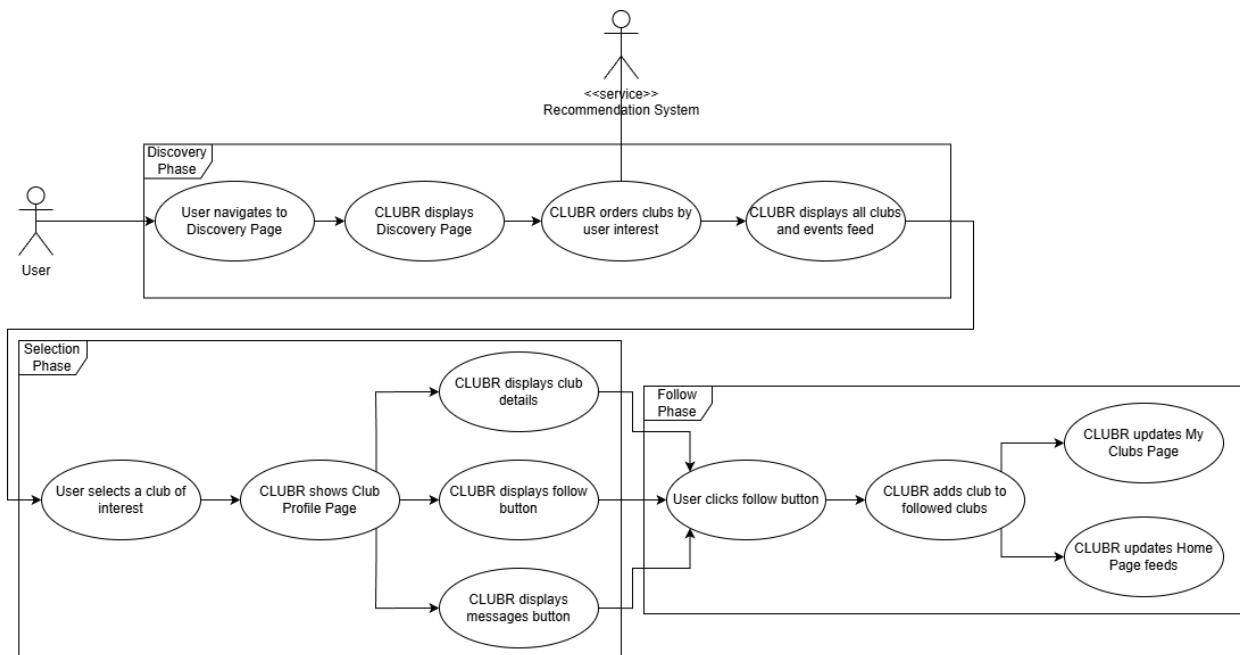
Flow of Events:

1. The USER wants to follow a club
2. The USER navigates to the Discovery Page
 - a. CLUBR responds by displaying the Discovery Page with all clubs, ordered by clubs that the user is most likely interested in (based on tagging), displayed first.
 - b. CLUBR displays a feed of all events from all clubs, as well as a search bar.
3. The USER selects a club that peaks their interest
 - a. CLUBR responds by showing the Club Profile Page for that particular club, which displays relevant club details, a “follow” button, and a “messages” button.
4. The USER clicks the “follow” button
 - a. CLUBR adds that club to the user’s followed clubs
 - b. CLUBR updates the My Clubs Page with this newly followed club
 - c. CLUBR updates the Home Page’s following feed and upcoming events feed.

Entry Condition: User is logged into Clubr

Exit Conditions: User has successfully followed their desired club

Quality Requirements: Each response should occur within 1-2 seconds after input



Use Case #2 Name: Admin2AdminMessaging

Participating Actors: ADMIN (type of USER)

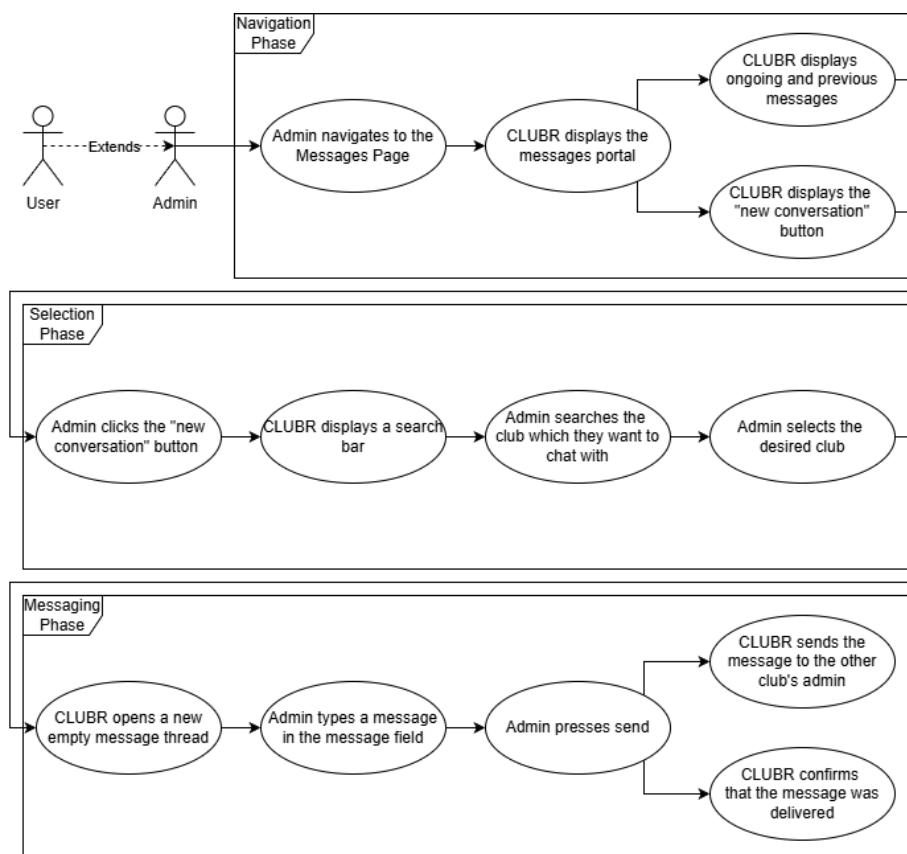
Flow of Events:

1. The ADMIN wants to message another club by messaging its admin
2. The ADMIN navigates to the Messages Page
 - 2.1. CLUBR displays the messages portal with all ongoing and previous messages conducted between this club and other clubs or users
3. The ADMIN clicks the “new conversation” button to message the admin of a club with whom they had not previously had a conversation
 - 3.1. CLUBR displays a search bar that allows the admin to find the club with which they wish to chat
4. The ADMIN selects the desired club
 - 4.1. CLUBR opens a new empty message thread between this admin and the selected club’s admin
5. The ADMIN types a message in the message field
6. The ADMIN presses send
 - 6.1. CLUBR sends the message to the other club’s admin
 - 6.2. CLUBR confirms that message was delivered successfully

Entry Condition: User has entered admin mode for a club

Exit Conditions: Message successfully sent to other club/club admin

Quality Requirements: Each response should occur within 1-2 seconds after input



Object Model

Entity Objects

User, Clubs, Membership, Posts, Events, Messages, Tags, Feed

Boundary Objects

LoginPage, SignupPage, InterestSelection, HomeView, DiscoveryView, MyClubsView, MessagesView, ClubProfileView, UserProfileView, ClubAdminView, Navigation Bar

Control Objects

AuthControl, TagsControl, FeedControl, PostsControl, EventsControl, ClubControl, MyClubsControl, DiscoveryControl, MessageControl, UserProfileControl, ClubProfileControl, ModeControl, MembershipControl

System Architecture

Overview

The Clubr system architecture has been divided into four subsystems. First, the User Interface displays pages and collects information from the user. With the input, it will either read or write from the Database subsystem. This will then trigger the correct operation in the Content Management subsystem, or it will switch pages and build lists through the Discovery and Messaging subsystem. The entire program is fully isolated from outside hardware and will run in a single operation mode for one user model. Both students and admins will run the same app, but will be given different permissions, mostly through the frontend.

Subsystem Decomposition

These are the four main subsystems of Clubr:

- User Interface
- Database
- Content Management
- Discovery and Messaging

User Interface

The User Interface is the only subsystem that directly interacts with the user. When the user clicks and uses form entries, other subsystems get triggered to deliver information back to the user. Here is an example:

- Pages shown to the user/admin: Home, Discover, My Clubs, Club Page, Messages, Admin Mode (for admins)
- Submit actions: follow/unfollow, create/edit post, create/edit event, send message

The subsystem has moderate coupling because UI changes need to be reflected in the calls to other subsystems. It has strong cohesion because each page and component has a single purpose, and it is not reused for other operations.

Database

The Database subsystem is used to store and retrieve application data. This includes:

- Users, clubs, membership/roles.
- Posts and events.
- Message threads and messages.

Data gets accessed through simple queries that are requested by the other subsystems through the UI. It has weak coupling because stored data and table layouts can be adjusted without changing UI behaviour, and each data area is self-contained. It has strong cohesion because it only provides basic operations like create/read/update for the records.

Content Management

The Content Management subsystem is used to perform actions that change club content. This includes:

- Follow/unfollow a club and manage member roles.
- Create/edit/delete posts and events.
- Update club profile information.

The behaviour of this system is triggered by the UI, so it does not directly interact with the user. Coupling to the UI is moderate, while cohesion is strong because each method has a small and specific scope, so it does not affect unrelated features.

Discovery and Messaging

The Discovery and Messaging subsystem is used for viewing and communication. This includes:

- Building Discovery lists and simple recommendations based on the selected interests/tags.
- Loading the Home feed by gathering recent posts/events from the clubs the user follows.
- Open or continue message threads between users and club admins.

The subsystem communicates with the Database subsystem to read what it needs, while receiving triggers from the UI. It has weak coupling to the rest of the system because it can change its listing logic without affecting how content gets created. The subsystem has strong cohesion because its methods focus only on finding/showing content and sending/receiving messages.

Software/Hardware Mapping

This software will not be mapped to any hardware, so this is not applicable to this program.

Persistent Data Management

We use PostgreSQL for our external database to store user information, club information, posts, events, and messages. PostgreSQL is an open-source object-relational database management system.

Access Control and Security

Table 1.0: Access Control Matrix

Access level	Posts	Personal Profile	Club Information	Messages
General Student	Read	Read/write	Read	Read/write
Club Administrator	Read/write	Read	Read/write	Read/write

The matrix above describes the permissions of two types of user levels. Although there are two types of access levels, the “Club Administrator” level is not a separate user, but rather a mode with additional permissions that can be toggled. This “Club Administrator” access level is granted initially to the creator of the club and can be consequently granted to users by anyone with the “Club Administrator” access level for a specific club. It is important to note that the “Club Administrator” access level is specific to one club; having this access level does not mean the user has read/write access to all posts for example.

The authentication for users would happen through password authentication, which is encrypted in the database for storage.

Global Software Control

Each subsystem executes independently and sometimes sequentially, so there is no concern with synchronization. All requests initiated by the user are handled in one sequential set of operations. One note of significance is that due to the sequential nature of some operations, any operation or subsystem may cause a bottleneck in the time to perform the sequential set of operations.

Boundary Conditions

1. Configuration Use Cases
 - Persistent Objects:
 - The following entities are stored permanently in the PostgreSQL database and must be created, initialized, or archived by specific system operations:
 - Users (credentials, profiles)
 - Clubs (name, description, tags)
 - Posts and Events associated with a club
 - Message Threads and Individual Messages
 - Membership information (role, admin status)
 - Creation Use Cases
 - A User Account is created through the “Sign Up” use case and initialized with encrypted credentials.
 - A Club is created through the “Create Club” use case. The creator is automatically assigned Club Administrator status for that club.
 - Posts and Events are created through the relevant UI forms in the Content Management subsystem.

- Message threads are created automatically upon the first message between two users.
 - Archiving / Removal Use Cases
 - A User account can be deactivated but not deleted, to preserve referential integrity for their content.
 - A Club may be archived by a Club Administrator. Archiving hides the club from discovery and prevents new posts/events.
 - Posts, events, and messages can be removed by the appropriate access level (e.g., club admin or message sender).
 - Archived objects remain in the database for auditability.
2. System Start-Up and Shut-Down Use Cases
- Start-Up Sequence
 - The web server initializes environment variables (database URL, encryption keys).
 - The backend connects to the PostgreSQL database.
 - Database migrations are applied if required.
 - API endpoints become available to the User Interface subsystem.
 - Cached data (e.g., interest tags, club lists) may be preloaded to improve performance.
 - Shut-Down Sequence
 - New incoming requests are rejected once shutdown begins.
 - Ongoing operations (writes/updates) are completed to avoid partial transactions.
 - Active database connections are closed gracefully.
 - Cached in-memory data is flushed.
 - The server process shuts down.
3. Exception Handling Use Cases
- The system must identify potential failures and respond predictably to preserve reliability and robustness.
 - Database Connection Failure
 - The API returns a standardized “Service Unavailable” error.
 - No write operations are attempted while disconnected.
 - The system retries the connection using exponential backoff.
 - Invalid or Malicious User Input
 - All form inputs are validated before hitting the database.
 - The UI displays descriptive error messages.
 - Server-side sanitization prevents SQL injection or script injection.
 - Authentication Failure
 - Users entering incorrect credentials receive an error without revealing whether the username exists.
 - Accounts are temporarily locked after repeated failed login attempts.
 - Permission Violations

- If a user attempts an action requiring Club Administrator status, the system blocks the request and returns an “Unauthorized” response.
 - Unauthorized modifications do not reach the Content Management subsystem.
- Network Outage or Browser Disconnect
 - Partially submitted forms are cancelled server-side.
 - No incomplete database writes occur.
 - Users may retry the failed operation after reconnecting.
- Unexpected Server Error
 - The system logs the error and returns a generic failure message.
 - Sensitive details are not exposed to the client.
 - The system remains available for other operations.

Subsystem Services

Each subsystem within the Clubr application provides independent and essential functionality. While each subsystem can operate on its own to fulfill a specific purpose, together they contribute to the full behaviour of the system and support the overall design goals of the application. The subsystems communicate with one another through clearly defined interfaces, ensuring the system remains cohesive while maintaining low coupling.

The User Interface subsystem is responsible for receiving and interpreting all user input. It displays application data in a clear and navigable format and translates user actions into requests for the other subsystems. All subsystem operations begin with a user action captured through the User Interface. The User Interface also updates its visible state based on the results returned from the Content Management, Discovery and Messaging, and Database subsystems.

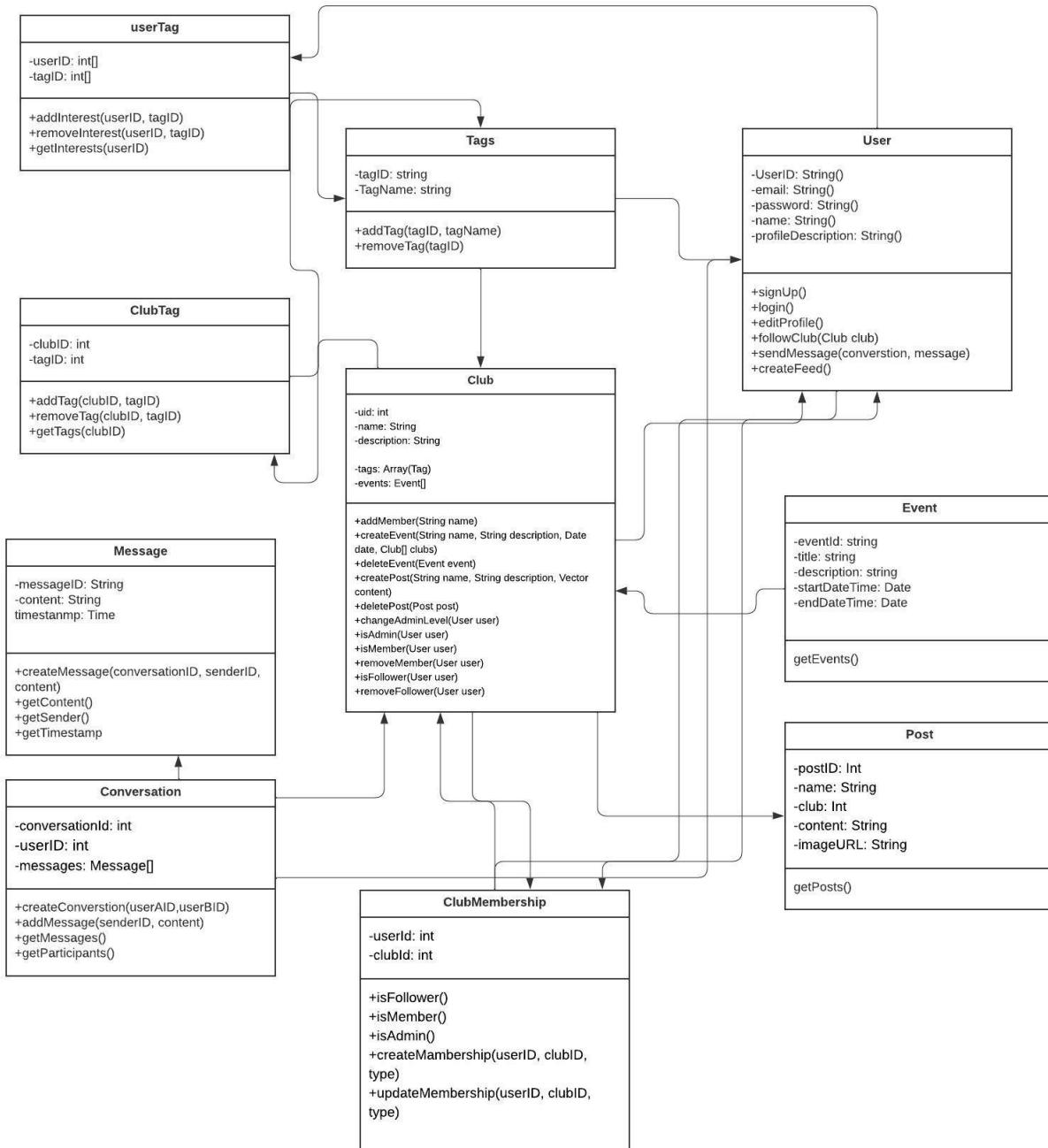
The Database subsystem is responsible for the storage, retrieval, and organization of all persistent data in the application. When the user requests information, such as club lists, posts, events, or messages, the Database subsystem locates the required records and returns them to the User Interface through the requesting subsystem. The information stored includes many types of objects, such as users, clubs, posts, events, and message threads, each of which may be displayed in different formats or views in the interface.

The Content Management subsystem is responsible for modifying club-related content within the application. It processes requests such as creating or editing posts and events, updating club information, and managing follow/unfollow behaviour. It receives its input from the User Interface, applies the required business rules (including permission checks), and then updates the Database subsystem accordingly. The results of these changes are reflected back to the User Interface for display to the user.

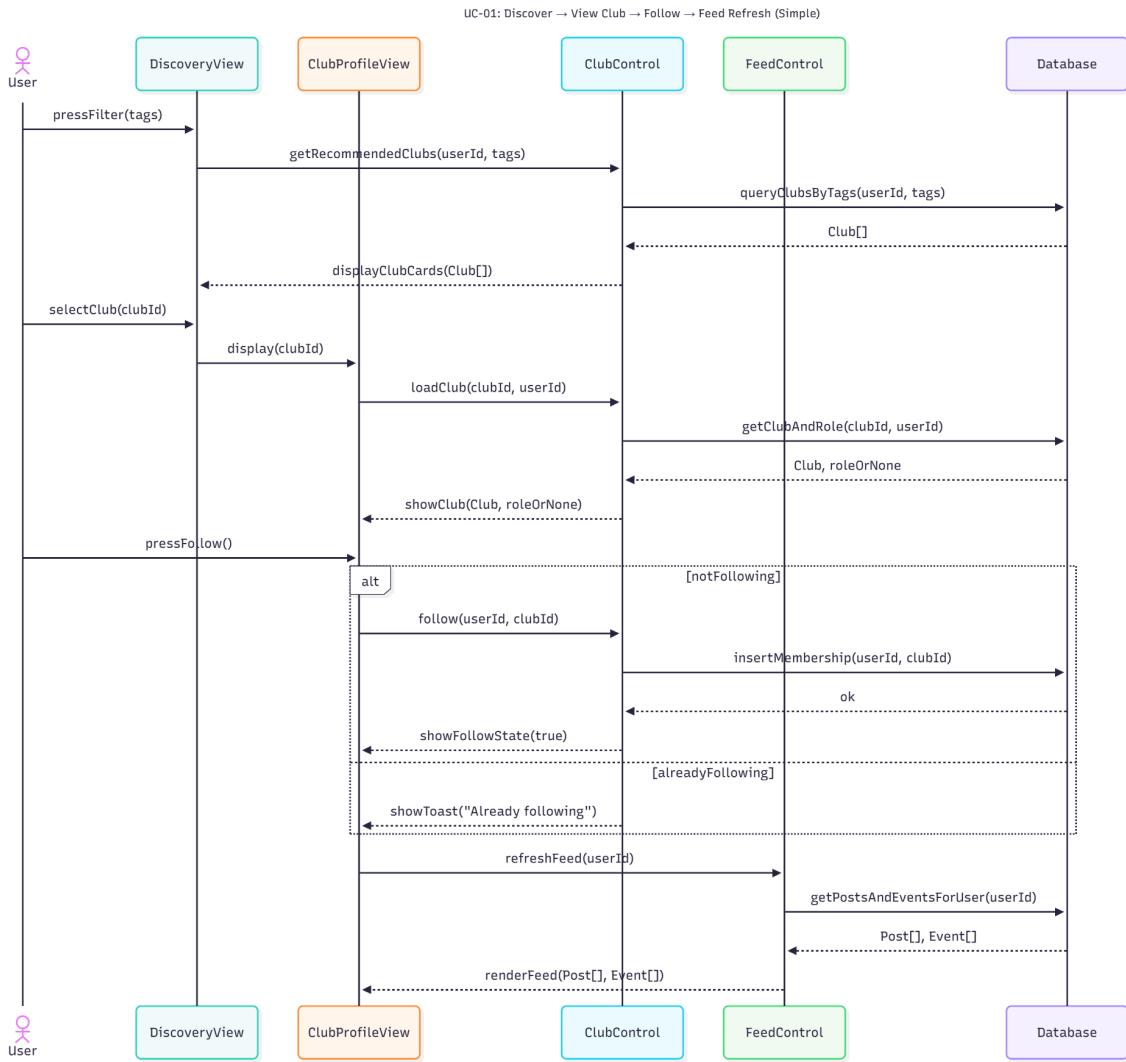
The Discovery and Messaging subsystem is responsible for gathering and presenting information that helps users navigate and communicate within the platform. It builds the Discover page listings based on tags and interests, generates the Home feed by collecting recent posts and events from followed clubs, and manages message threads between users. It

retrieves the required data through the Database subsystem and presents it to the User Interface in a structured, user-friendly way.

Class Diagram

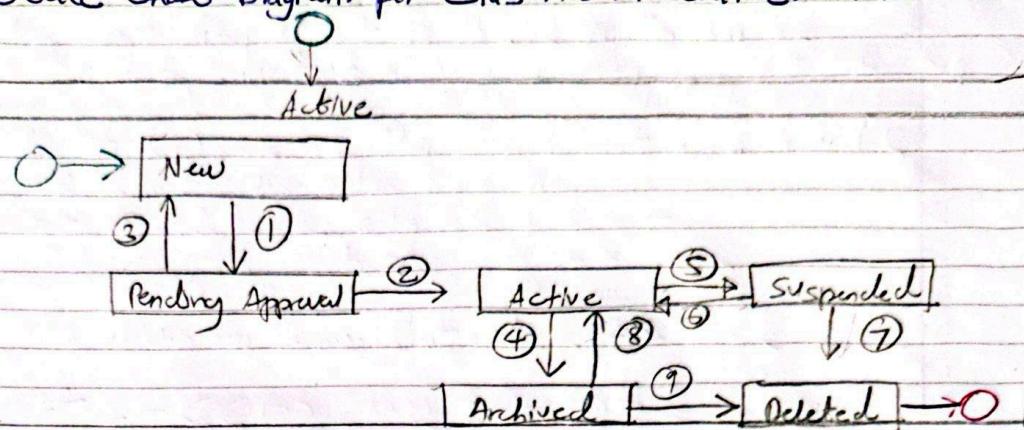


Sequence Diagram



State Chart Diagram

State Charts Diagram for Club Administrative Status



①: Details Complete

②: AMS Approved

③: AMS Rejected

④: Club Disbands

⑤: Admin Suspends

⑥: Admin Reinstates

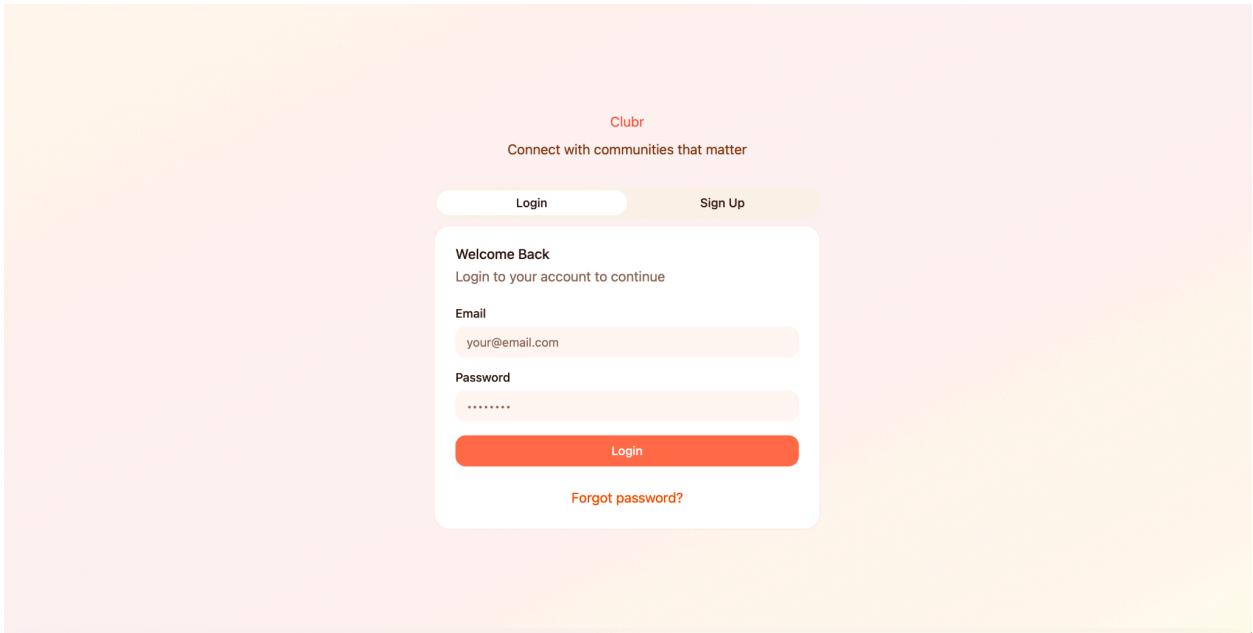
⑦: Admin Deletes

⑧: Archived Relunched

⑨: Delete Purge Policy

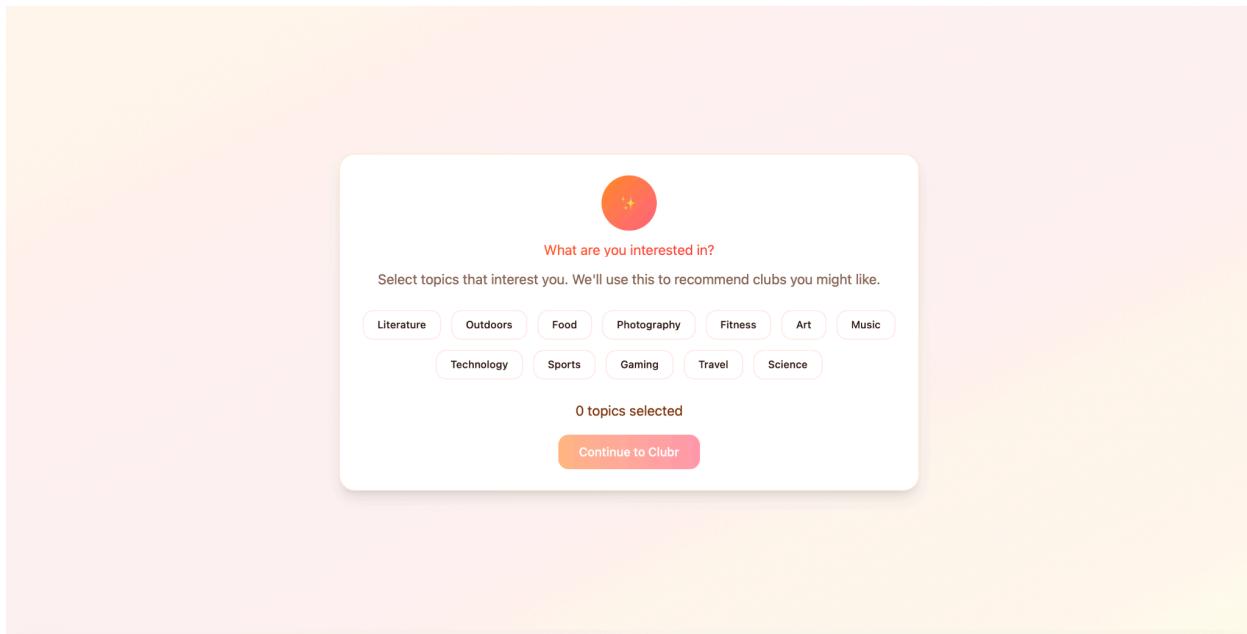
GUI Screenshots

Authorization



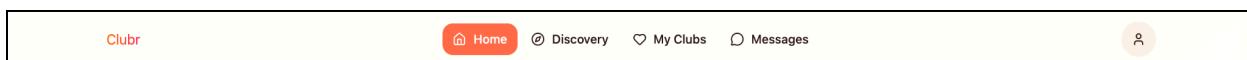
User entry into the app begins with an account. Users with an existing account will login, or if their login details are saved, will be logged in automatically. New users will be prompted to create an account.

Interests



Upon creating an account, users will be prompted to select topics that interest them. This information will be stored and used later to recommend clubs to join via the discovery feed.

Navigation Bar



Becoming visible upon logging-in, the navbar will be present on any page, allowing the user to quickly move to whatever page they want to. Clicking the Clubr logo or Home button will return to the Home Page. Clicking Discovery will bring the user to their Discovery Page. Clicking My Clubs will show the user an index of the clubs they've joined, allowing them to navigate to the club's landing page. Messages will bring the user to their messages tab.

The profile button will show the user their account, where they can make changes to their photo, about-me, and preferences, as well as being able to sign out. Additionally, club admins can also change to "admin mode" from the profile dropdown, which will allow them to access their club's specific view and all its accompanying features.

Home Page

The screenshot shows the Clubr application's Home Page. At the top, there is a navigation bar with tabs for Home, Discovery, My Clubs, and Messages. A user profile icon is also present.

Following Feed: This section displays posts from clubs the user follows. It includes a post from "Queen's Journal" about their latest edition and another from "Queen's Badminton Team" about a recent win.

Upcoming Events: This section shows events from clubs the user follows. It lists the "Journalism Workshop: Investigative Reporting" on Nov 4, 2025, and the "OUA Championship Tournament" on Nov 7, 2025.

For users that are already following a club they will be taken to the default Home Page. Contained within this Home Page is a “Following” feed which will display posts from clubs which the user follows. Clicking on these posts will take the user to the post within the club’s own page. Next to that will be some form of calendar, displaying upcoming events for clubs that the user is following. Similar to posts, clicking on events will bring the user to the event on the club page. For users that are not following any clubs, the Home Page will prompt them to explore and follow clubs using the Discovery Page to populate their Home Page.

Discovery Page

The screenshot shows the Discovery Page interface. At the top, there is a navigation bar with links for Home, Discovery (which is highlighted in red), My Clubs, and Messages. A user profile icon is also present. Below the navigation bar, there are two main sections: "Discover Clubs" and "All Events".

Discover Clubs: This section displays recommended clubs based on user interests. It includes a search bar and two club cards:

- Queen's Model Parliament** (Politics): An image of the Palace of Westminster at night. Description: "Experience Canadian parliamentary democracy firsthand. Debate current issues,...". Members: 63. Follow button.
- Vogue Charity Fashion Show** (Fashion): An image of a fashion show runway. Description: "Queen's largest student-run fashion show raising funds for local charities. Designers,...". Members: 128. Follow button.

All Events: This section displays upcoming events from all clubs. It includes a search bar and three event cards:

- Journalism Workshop: Investigative Reporting** (Nov 4, 2025): Queen's Journal. Description: "Nov 4, 2025", "7:00 PM", "Journal Office - Carruthers Hall".
- OUA Championship Tournament** (Nov 7, 2025): Queen's Badminton Team. Description: "Nov 7, 2025", "9:00 AM", "Queen's Athletics & Recreation Centre".
- Annual Charity Fashion Show** (Nov 9, 2025): Vogue Charity Fashion Show. Description: "Nov 9, 2025".

The Discovery Page will look similar to the Home Page, with a feed and an upcoming events calendar. In this case, rather than posts the feed will contain all clubs which the user is not a member of. This feed will be populated in-order based on the user's interests with clubs similar to clubs they're already in, being shown near the top. Next to the feed will be a display for all upcoming events from all clubs, with events from recommended clubs from prioritized.

Clicking on the clubs in the Discovery Page will bring the user to the club's landing page, where they can learn more about the club and follow/message them.

My Clubs Page

The screenshot shows the 'My Clubs' page with two club profiles displayed:

- Queen's Journal** (Media): Queen's independent student newspaper since 1873. Join our team of writers, editors,...
87 members
Following
- Queen's Badminton Team** (Sports): Competitive and recreational badminton for all skill levels. Weekly practices, tournaments,...
45 members
Following

The My Club Page contains all the clubs the User has joined or followed. Clicking on the Club icon will take you to the club's page. Users can also unfollow clubs here.

Club Page

The screenshot shows the Queen's Journal Club Page with the following details:

- Queen's Journal** (Media)
- 87 members
- Queen's independent student newspaper since 1873. Join our team of writers, editors, photographers, and designers to cover campus news, arts, sports, and more.
- Following
- Message Admins
- Posts Events Members About
- Queen's Journal

The Club Page functions as a landing page for any given club. Users will be able to see the club's posts, upcoming events, as well as additional information about the club including club

members. From here users will be able to follow and/or apply for the club. Additionally the user can press a button to be redirected from this page to a chat with club admins.

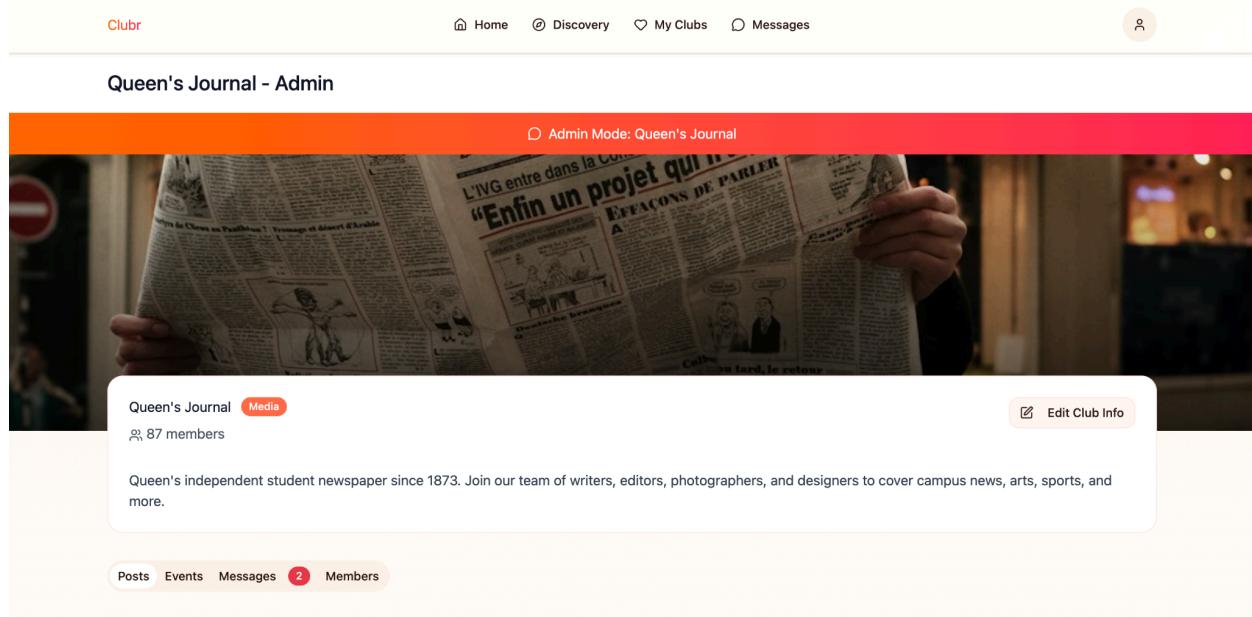
Club Page (Admin Mode)

The Club Page (in admin mode) will appear slightly different to admins, with more options visible. Users flagged as admins for a club will be able to edit the about section, make and remove posts, create events, and add and remove members.



This screenshot shows the Queen's Journal club page on Clubr. At the top, there are navigation links: Home, Discovery, My Clubs, and Messages. A user profile icon is in the top right. On the left, a thumbnail image shows a person holding a newspaper. The main content area has a header "Queen's Journal". Below it, a box displays the club's name, a "Media" button, and a member count of 87. A bio text describes the club as an independent student newspaper since 1873. At the bottom, there are buttons for Posts, Events, Members, and About.

Admin mode for a club can be entered from the top right member button if the user is an authorized Admin for a club.



This screenshot shows the Queen's Journal club page in Admin Mode. The interface is identical to the standard mode screenshot above, but the user profile icon in the top right now indicates "Admin Mode". A red banner at the top of the content area reads "Admin Mode: Queen's Journal".

While in Admin Mode, it will show a ribbon on top (can only enter admin mode for a single club at a time).

The screenshot shows the Clubr platform in Admin Mode. At the top, there is a navigation bar with links for Home, Discovery, My Clubs, and Messages. Below the navigation is a ribbon containing the club name "Queen's Journal", a "Media" button, a member count of "87 members", and an "Edit Club Info" button. The main content area displays a post from "Queen's Journal" made 2 hours ago. The post content is: "This week's edition is out! Featuring an exclusive interview with the AMS President and coverage of Homecoming 2025. Pick up your copy around campus!". Below the post are engagement metrics: 34 likes, a comment icon, and a share icon.

Admin - Add Posts function

The screenshot shows the Clubr platform in Admin Mode, specifically the "Add Posts" function. The top navigation bar and ribbon are identical to the previous screenshot. The main content area shows a post from "Queen's Journal" about a "Journalism Workshop: Investigative Reporting". The post includes the date "Nov 4, 2025". Below the post, event details are listed: "Nov 4, 2025", "7:00 PM", and "Journal Office - Carruthers Hall".

Admin - Add/Edit Events

The screenshot shows a club profile page for "Queen's Journal". At the top, there are navigation links: Clubr, Home, Discovery, My Clubs, and Messages. Below the header, the club's name "Queen's Journal" is displayed with a "Media" button. It shows 87 members and a brief description: "Queen's independent student newspaper since 1873. Join our team of writers, editors, photographers, and designers to cover campus news, arts, sports, and more." A "Edit Club Info" button is visible. Below this, a navigation bar includes Posts, Events, Messages (with a red notification badge), and Members. A section titled "Member Messages" displays a message from "Alex Thompson" posted 2 hours ago: "Hi! I'm interested in joining the club. When is the next meeting?". A "Reply" button is shown. The overall theme is light orange and white.

Admin - View/Edit Messages

This screenshot shows the same club profile page for "Queen's Journal" but with "Admin Mode" active. A red banner at the top indicates "Admin Mode: Queen's Journal". On the right side, a vertical menu titled "My Account" is open, showing options: Profile, Admin Mode, Exit Admin Mode, Queen's Journal (Active), Queen's Badminton Team, and Sign Out. The "Admin Mode" option is highlighted with a red background. The rest of the page content is identical to the first screenshot, including the member message from Alex Thompson.

Admin - Exit Admin Mode

Messages Page

The screenshot shows the 'Messages' page within the Clubr app. At the top, there are navigation links: 'Clubr' (highlighted in red), 'Home', 'Discovery', 'My Clubs', 'Messages' (highlighted in red), and a user profile icon. Below the navigation is a header bar with 'Messages' on the left and a '+ New Conversation' button on the right.

The main area displays a list of conversations. On the left, under 'Conversations', there are two messages:

- A message from 'Queen's Journal' (Q) at 10:35 AM: "Our next meeting is November 5..."
- A message from 'Queen's Badminton Team' (Q) at Yesterday: "Practice is at 7pm tomorrow. D..." followed by a red notification badge with the number '2'.

On the right, a detailed conversation with 'Queen's Journal' (Q) is shown:

- Message from 'Queen's Journal' (Q): "Club Admins"
- Message from 'Emma - Journal Editor' (E): "Thanks for your interest in the Queen's Journal! We'd love to have you on our writing team."
- Message from 'You': "Thank you! When is the next writers meeting?" (sent at 10:32 AM)
- Message from 'Emma - Journal Editor' (E): "Our next meeting is November 5th at 7 PM in Carruthers"

At the bottom of the messaging area, there is a text input field with placeholder text 'Type your message...' and a red send button with a white arrow icon.

From here users can contact club admins. Admins are also able to talk with each other from this page. The page will display a list of ongoing chats, which upon clicking will display a messaging box where the user can send and receive messages.

Profile Page

The screenshot shows the 'Profile' page within the Clubr app. At the top, there are navigation links: 'Clubr' (highlighted in red), 'Home', 'Discovery', 'My Clubs', 'Messages' (highlighted in red), and a user profile icon. Below the navigation is a header bar with a user photo, the name 'Alex Thompson', and an 'Edit Profile' button.

The main area displays user information and club roles:

- User photo: A circular orange gradient placeholder for a profile picture.
- User details: Name 'Alex Thompson', email 'alex.thompson@queensu.ca', location 'Kingston, ON', and joining date 'Joined September 2023'.
- User description: 'Third-year Commerce student passionate about community building and trying new activities! Love meeting new people and exploring different interests.'
- Statistics: '2 Clubs Joined', '2 Admin Roles', and '8 Total Clubs'.
- Admin Roles: Shows '2 Admin Roles' managing 'Queen's Journal' (87 members) and 'Queen's Badminton Team' (45 members).
- My Clubs: Shows '8 Total Clubs' where the user is a member.

Users can edit their profile from default values by clicking 'Profile' on the top right dropdown menu. The profile page contains the user's name, photo, description, email, club roles and memberships.

Coding Assignments

Member	Code Assignments
Andrew K.	Frontend implementation of LoginPage, SignupPage, InterestSelection, Navigation Bar, and shared layout/styling components.
Kosi	Frontend implementation of HomeView (following feed and events calendar), DiscoveryView (recommended clubs feed), and tag-filtering UI elements.
Justin	Frontend implementation of MyClubsView, ClubProfileView, ClubAdminView (admin mode UI), and MessagesView (messaging interface).
Jackson	Backend implementation of AuthControl, UserProfileControl, MembershipControl and related APIs for signup/login, profile updates, and membership/role management.
Kai	Backend implementation of ClubControl, PostsControl, EventsControl and APIs for creating/editing clubs, posts, and events.
Sophia	Backend implementation of FeedControl, TagsControl, DiscoveryControl, MessageControl and APIs for discovery (interests/tags), home feed generation, and messaging.
Aaron	Design and maintenance of the database schema (User, Clubs, Membership, Posts, Events, Messages, Tags, Feed), migrations/seed data, and integration layer between backend services and PostgreSQL.
Uroosa, Andrew T.	Security and reliability features: authentication/session security (password hashing, tokens), enforcement of the access-control matrix, input validation/sanitization, and exception-handling for security-related failures.

Timeline

Nov 14–16

- Finalize overall GUI design and page flow (Login, Home, Discover, My Clubs, Club Page, Messages).
- Set up project repository, frameworks, and shared components.
- Design and implement initial database schema (Users, Clubs, Membership/Roles, Posts, Events, Messages).

Nov 17–19

- Frontend: build core page layouts and navigation bar (Login/Signup, Home, Discover, My Clubs, basic Club Page).
- Backend: implement core APIs for login/signup, user profiles, and basic club creation/joining.
- Database: refine tables and add seed data for test users and clubs.

Nov 20

- Team meeting (tutorial 11:30): review design, confirm coding assignments, and refine milestones.
- Begin implementing authentication (AuthControl, user profiles, sessions).

Nov 21–23

- Frontend: implement detailed Club Page, Admin Mode shell, and Messages page UI.
- Backend: implement club management (ClubControl), posts/events creation, and messaging endpoints.
- Begin connecting frontend components to live backend APIs.

Nov 24–26

- Start end-to-end tests for main user flows (sign up, follow clubs, view posts/events, send messages).
- Implement and test access control and security features (roles, permissions, input validation).
- Debug integration issues across frontend, backend, and database.
- Polish GUI (styling, responsiveness, small usability tweaks).

Nov 27

- Team meeting (tutorial 11:30): integration check-in and bug triage; finalize demo plan.
- Finalize and proofread GUI and all visible text.
- Prepare and finalize PowerPoint slides and demo script.
- Full practice run of final presentation and live demo.

Nov 28

- Final project presentation.

Nov 29–Dec 1

- Address any feedback from the presentation (if required).
- Final cleanup of codebase, comments, and documentation.
- Prepare and submit any remaining written deliverables and backups.