

# **Software Design Specification**

**for**

# **Digital Peer Support System**

**Version <2.0>**

**Group No.: 8**

Fatin Shamirah Kamal	1211111996
Ashley Law Jia Ye	252UC2552B
Rayyan Syahmi Bin Sukri Efendi	242UC244FE
Siti Aisyah Binti Omar	243UC245Y6

**Date:** 8 January 2026

Revisions .....	5
1 System Overview .....	6
1.1 Description.....	6
1.2 Actors .....	9
1.3 Assumptions and Dependencies .....	10
1.3.1 Assumptions.....	10
1.3.2 Dependencies .....	10
1.4 Use Case Diagram.....	11
2 Flow Charts .....	12
2.1.1 Sign up a new account.....	12
2.1.2 Log in with own account.....	13
2.1.3 Log out with own account .....	14
2.1.4 Reset password.....	15
2.1.5 Report contents .....	16
2.1.6 Build profile.....	17
2.1.7 Check-in mood and well-being.....	18
2.1.8 Add friends .....	19
2.1.9 Chat privately .....	20
2.1.10 Make posts .....	21
2.1.11 Comment on posts.....	22
2.1.12 Like posts and comments .....	23
2.1.13 Earn points .....	24
2.1.14 Match-up with people .....	25
2.1.15 Book counselor appointments.....	26
2.1.16 Review Reported Messages .....	27
2.1.17 Make Announcements.....	28
2.1.18 Flag Student Account .....	29
2.1.19 Delete Forum Comments.....	30
2.1.20 Delete Forum Posts .....	31
2.1.21 Monitor Mood Alert Dashboard.....	32

2.1.22	Manage Counselor Assignment.....	33
2.1.23	Review Reported Content .....	34
2.1.24	Manage Scoring System.....	36
2.1.25	Process Appeals .....	36
2.1.26	Suspend Accounts.....	37
2.1.27	manage caseload.....	39
2.1.28	Export data .....	39
2.1.29	Verify assignments.....	40
2.1.30	Manage action plans .....	41
2.1.31	Schedule appointments.....	42
3	Data Design.....	44
3.1	Design Class Diagram.....	44
3.2	Data Dictionary .....	45
3.3	Data Structures.....	59
3.3.1	Data Structure 1 : User Identity & Management .....	59
3.3.2	Data Structure 2 : Mental Health & Counseling.....	59
3.3.3	Data Structure 3 : Community & Social Interaction .....	60
3.3.4	Data Structure 4 : Safety, Moderation & Governance.....	60
3.3.5	Data Structure 5 : Authentication & Session Logs.....	61
3.3.6	Data Structure 6 : Gamification.....	61
4	Behavioral Modeling .....	61
4.1	Sequence Diagrams .....	61
4.1.1	Sign up a new account.....	61
4.1.2	Log in with own account.....	62
4.1.3	Log out with own account .....	64
4.1.4	Reset Password .....	65
4.1.5	Report contents .....	66
4.1.6	Build profile.....	68
4.1.7	Check-in mood and well-being.....	68
4.1.8	Add friends .....	69
4.1.9	Chat privately .....	70

4.1.10 Make posts .....	71
4.1.11 Comment on posts.....	72
4.1.12 Like posts and comments .....	73
4.1.13 Match-up with people .....	74
4.1.14 Book counselor appointments.....	75
4.1.15 Earn points .....	76
4.1.17 Manage Counselor Assignments.....	80
4.1.19 Manage Scoring System.....	84
4.1.20 Process Appeals .....	86
4.1.21 Suspend Accounts .....	88
4.1.22 Review Reported Messages .....	90
4.1.23 Flag Student Account .....	91
4.1.24 Make Announcements.....	92
4.1.25 Delete Forum Posts .....	93
4.1.26 Delete Forum Comments.....	94
4.1.27 Manage caseload.....	95
4.1.28 Export data .....	96
4.1.29 Verify assignments.....	98
4.1.30 Manage action plans .....	99
4.1.31 Schedule appointments.....	101
4.2 State Diagram .....	102
5 Architecture Design.....	104
5.1 Software Architecture .....	104
5.1.1 Student Subsystem.....	105
5.1.2 Moderator Subsystem .....	106
5.1.3 Admin Subsystem .....	107
5.1.4 Counselor Subsystem .....	107
6 Interface Design .....	108
6.1 Main Screens .....	108
6.2 Student Subsystem Screens .....	109
6.3 Moderator Subsystem Screens.....	115

6.4	Admin Subsystem Screens.....	117
6.5	Counselor Subsystem Screens.....	120
7	Component Design .....	122
7.1	Main Components .....	122
7.1.1	Flask Server (Controller) .....	123
7.1.2	Jinja2 Templates (Boundary) .....	123
7.1.3	Web Browser (User Interface).....	123
7.1.4	MySQL Database (Entity) .....	124
7.1.5	External Services .....	124
8	Deployment Design.....	124
8.1	Deployment Diagram .....	124
9	Summary .....	124
	References .....	125

## Revisions

Version	Primary Author(s)	Description of Version	Date Completed
SRS in Part 1(as Ver 1.0) SDS in Part 2(as Ver 2.0.X) *System Documentation in Part 3 (as Ver 3.0) Draft Type and Number	Full Name	Information about the revision. This table does not need to be filled in whenever a document is touched, only when the version is being upgraded.	00/00/00

# 1 System Overview

## 1.1 Description

The Digital Peer Support System is an educational support platform designed to help the Multimedia University students connect with peers, moderators, and counselors for emotional, academic, and communicative support. Its main functions include user profile management, sharing experience or seek help through posts in the forum anonymously or non-anonymously, real-time privately chat communication, peer matching based on interests or goals, mood and well-being tracking, and administrative oversight for content and system management. The primary users include Students or peer who are general users in this system, Moderators who responsible for updating announcements on time and maintaining safe interactions, Counselors that providing professional guidance, and Administrators who manage roles and platform configuration. The system offers a unified digital platform that involved peer-to-peer support, moderated discussion spaces, structured connections for academic and emotional assistance, and announcements or information from the university.

The major functional groups of the digital peer support system included but not limited to:

### A. User Management & Authentication

- Actors included: Student, Moderator, Counselor, Administrator
- The system shall allow students to create (sign in) accounts, and allow all users (students, admin, counselor, and moderator) to log in securely. Admins can manage user roles and account status.
- Key function: Account sign in & login; Reset account password; Profile building; Role assignment Account management

### B. Experience Sharing & Peer Support

- Actors: Student, Moderator
- Students can create posts, share personal experiences, ask for support, and participate in discussions anonymously or non-anonymously. through Forum. While moderators oversee and manage the content of the forum (including posts and comments) to maintain a safe environment.
- Key Functions: Make posts; Comment on the posts; Like the posts and comments; Review Reported Message (Moderators); Delete forum posts and comments (Moderators); Flag student account (Moderators)

### C. Automated Peer Matching

- Actors: Student

- Students are able to build and manage their profiles which include interest label tags such as hobbies, MBTI, Age, Horoscope, Zodiac, DOB, goals, or even communication preferences. The system must match users with peers based on the interest tags in their profile to encourage productive support sessions.
- Key Functions: Match-up with people having similar interest labels

**D. Communication Platform (Chat & Appointment)**

- Actors: Student, Counselor
- The system enables students to add friends and chat privately by providing a real-time and asynchronous communication features. Besides, student can book appointments with preferred counselor effectively by selecting counselor's available date and time. Students can
- Key Functions: Add friend feature; Real-time private chat; Counselor appointments scheduling

**E. Well-Being Tracking & Mood Monitoring**

- Actors: Student, Counselor
- Students can track check-in their mood and well-being daily, while counselors able to get access with those records and care for students with scores below the threshold effectively.
- Key functions: Manage assigned student caseload (Counselor); Scheduling appointments (Counselor); Export well-being and engagement data (Counselor); Verifying assignments from administrators (Counselor); Manage therapeutic action plans (Counselor); Check-in mood and well-being (Student)

**F. Gamification (Points for duality on safety score & reward)**

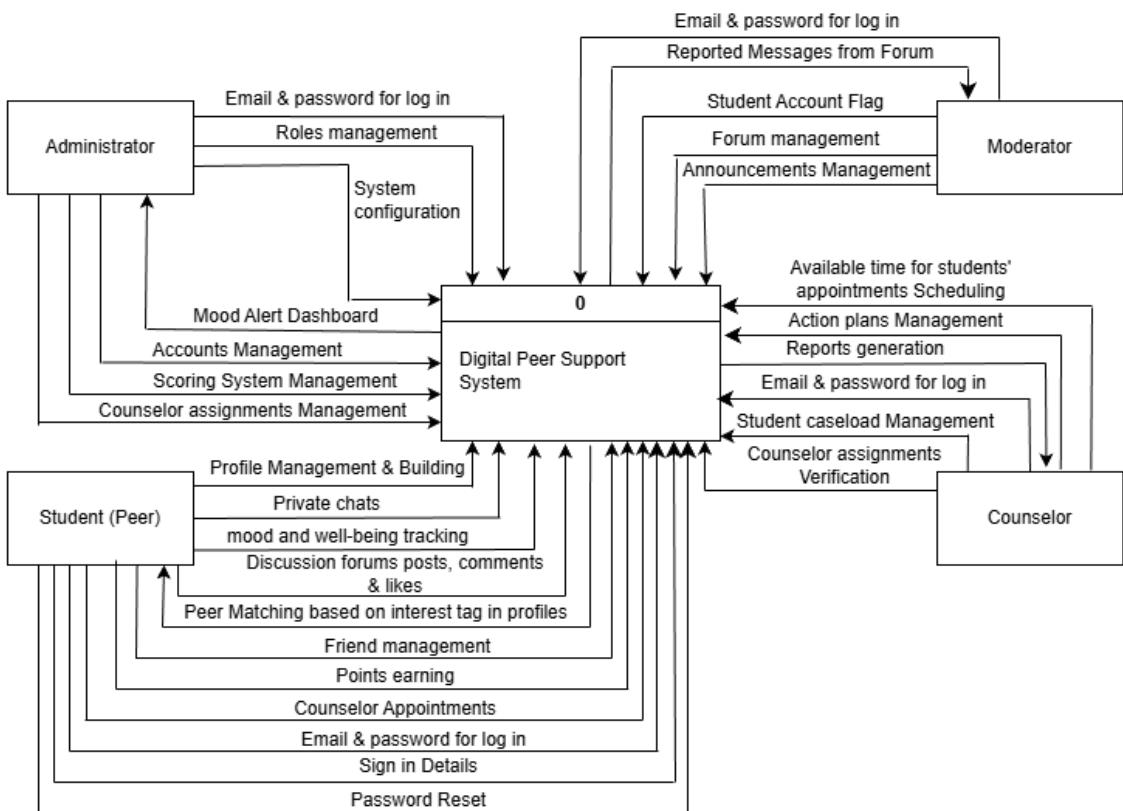
- Actors: Student, Admin
- To encourage healthy participation, students earn points for positive engagement (posting, commenting, and checking-in their moods). The maximum points will be 100, and every new account will start with 100 points initially. While certain points will be deducted from the student account if admin determines the student violated rules, and accounts will be restricted once points <60. Accounts that manage to maintain 100 points will award a verified tick beside their username; while larger or equal to 95 points leading rainbow colour for their username when they are involved in the forum, promoting consistent involvement.
- Key Functions: Scoring management (Admin), Earn Points (Student)

**G. Administrative Management & System Configuration**

- Actors: Admin

- Admins are responsible for overseeing the platform's safety, performance, and operational rules. Their functions focus on monitoring student well-being alerts, managing counselor assignments, handling content issues, and maintaining the scoring and account systems.
- Key Functions: Monitor Mood Alert Dashboard; Manage Counselor Assignments; Review Reported Content; Manage Scoring System; Process Score Appeals; Suspend Account

Overall, the system aims to offer safe communication channels and supportive interactions in an educational environment. The features included with discussion forums, private chats, mood check-ins, tools for peer matching, moderation dashboards, and tracking user progress. The main challenges involve protecting data privacy, ensuring secure sign-ins, and limiting harmful or inappropriate content timely. Overall, the system helps users share experiences, seek support, and develop healthy communication habits while allowing staff to keep track of safety and activity on the platform.



*Diagram 1.1.1 Top Level Data Flow Diagram*

## 1.2 Actors

The actors including administrators, students (peer), moderators, and counselors.

Actor	Use Cases
Admin	<ul style="list-style-type: none"><li>• Log in with own account</li><li>• Log out with own account</li><li>• Monitor Mood Alert Dashboard</li><li>• Manage Counselor Assignments</li><li>• Review Reported Content</li><li>• Manage Scoring System</li><li>• Process Appeals</li><li>• Suspend Accounts</li></ul>
Student	<ul style="list-style-type: none"><li>• Sign up a new account</li><li>• Log in with own account</li><li>• Log out with own account</li><li>• Reset password</li><li>• Report contents</li><li>• Build profile</li><li>• Check-in mood and well-being</li><li>• Add friends</li><li>• Chat privately</li><li>• Make posts</li><li>• Comment on posts</li><li>• Like posts and comments</li><li>• Match-up with people</li><li>• Book counselor appointments</li><li>• Earn points</li></ul>
Moderator	<ul style="list-style-type: none"><li>• Log in with own account</li><li>• Log out with own account</li><li>• Review Reported Messages</li><li>• Flag Student Account</li><li>• Make Announcements</li><li>• Delete Forum Posts</li><li>• Delete Forum Comments</li></ul>
Counselor	<ul style="list-style-type: none"><li>• Log in with own account</li><li>• Manage caseload</li><li>• Export data</li><li>• Verify assignments</li></ul>

	<ul style="list-style-type: none"><li>• Manage action plans</li><li>• Schedule appointments</li><li>• Log out with own account</li></ul>
--	--

*Table 1.2.1 Actors included in the Digital Peer Support System*

## 1.3 Assumptions and Dependencies

### 1.3.1 Assumptions

The Digital Peer Support System design and implementation work according to these built-in assumptions. The system will experience operational issues if either of these two assumptions proves false or undergoes modification.

- **User Honesty:** Students will provide truthful responses about their daily moods when they complete the "Daily Mood Check-in" (1-5 scale) assessment. The system uses self-reported data to calculate "Wellness Battery" scores which trigger "Counselor Alerts."
- **Institutional Email Format:** The university will continue using its existing email domain format (@student.mmu.edu.my) to identify students according to this assumption. The system uses a specific Regex pattern to validate student registration; changes to this domain format would require code refactoring.
- **Counselor Availability:** The system requires enough Counselors and Administrators to operate its dashboard monitoring functions. The system creates alerts when users reach "Critical" mood scores which drop below 30% but the software needs human action because it expects a counselor to check the dashboard every day.
- **Internet Connectivity:** The assumption states that target users (students and counselors) will have access to uninterrupted reliable internet service. The application operates entirely through the web while users currently lack access to an "Offline Mode" which would allow them to submit journals and read forum posts.
- **User Volume:** The initial deployment (pilot phase) will start with an assumption that user activity will remain below the free-tier hosting plan (Render) and database plan (Aiven MySQL) limits. A server upgrade will become necessary when a traffic increase exceeds 500 users who access the system at the same time.

### 1.3.2 Dependencies

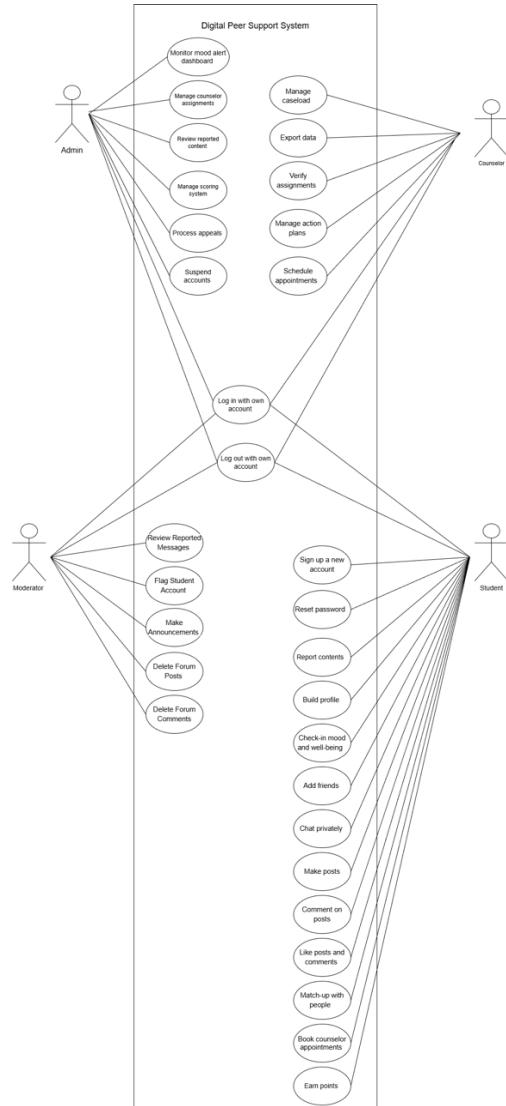
The project depends on both external factors and third-party components which function as its fundamental elements. The system will experience complete operational failure when these services reach their end or when they become outdated.

- **Cloud Database Provider (Aiven MySQL):** The system requires Aiven to keep all its data. The application functions only when Aiven maintains its service operational.

- **Email Gateway (Google SMTP):** The system uses Google's Gmail SMTP server to operate both the "Forgot Password" and OTP (One-Time Password) functions. The system assumes Google will not block the application's IP address or revoke the App Password credentials. Network policies that block port 587 or 465 create a dependency risk because they occur frequently in university networks.
- **Media Hosting (Cloudinary):** The "Forum" feature enables users to upload both images and videos. The system transfers files to the Cloudinary API instead of keeping them on its local server. The ability to view and upload media files depends on the availability of Cloudinary's API and the project's compliance with Cloudinary's storage limitations.
- **Hosting Infrastructure (Render):** The entire application operation process including SSL certificate handling and Python environment deployment depends on Render PaaS (Platform as a Service) system.
- **Browser Compatibility:** The frontend requires modern web standards which include HTML5 and CSS3. The system needs users to access it through modern browsers which include Google Chrome and Safari and Microsoft Edge. The system does not support legacy browsers which include Internet Explorer.

## 1.4 Use Case Diagram

The use case diagram shows every use case involved in our Digital Peer Support System.



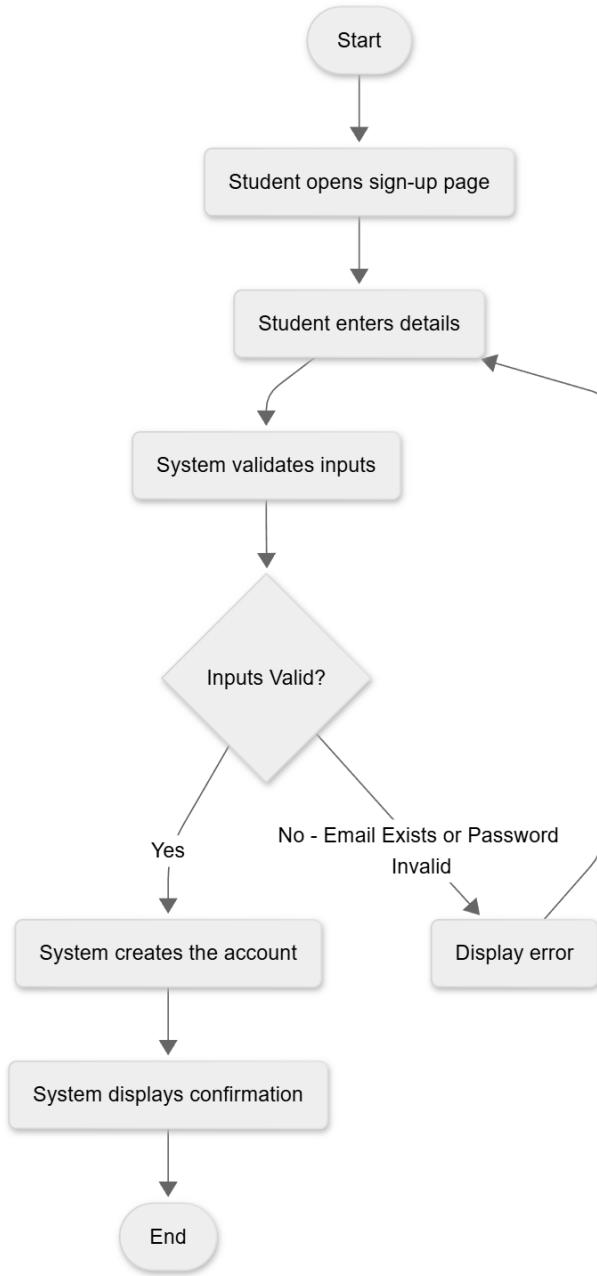
*Diagram 1.4.1 Use Case Diagram*

## 2 Flow Charts

### 2.1 Flow Charts

#### 2.1.1 Sign up a new account

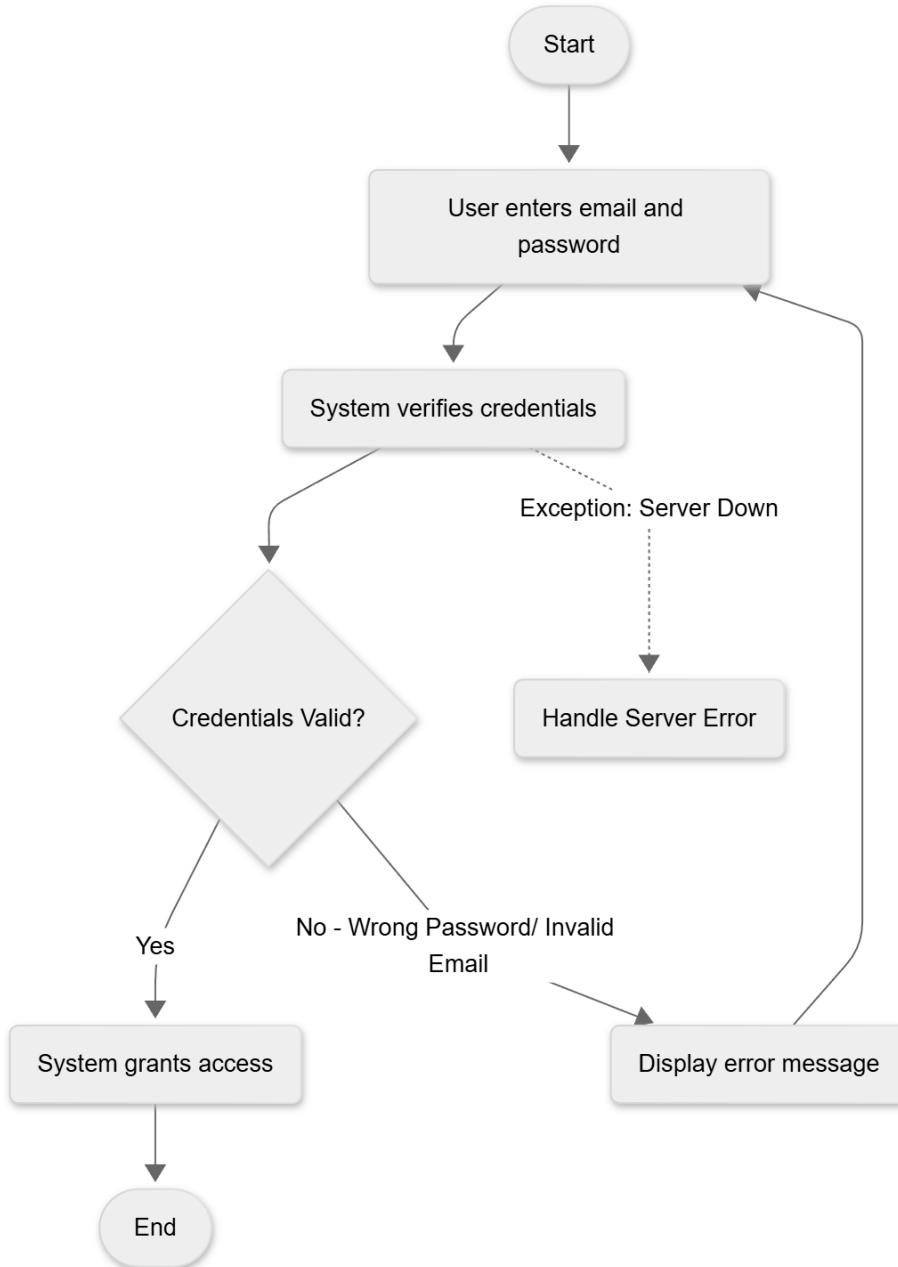
The student is able to sign up for a new account using MMU official email student accounts and providing the required information, including username and password that consist at least eight characters including uppercase, lowercase, number and symbol.



*Diagram 2.1.1 Flow Chart Sign Up a New Account*

### **2.1.2 Log in with own account**

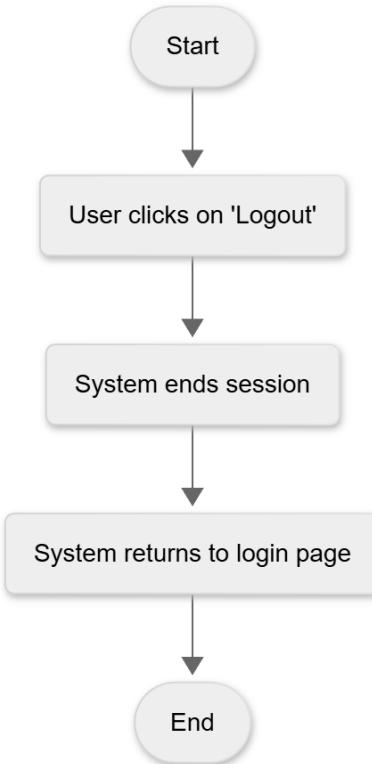
The actors (both student, moderator, counselor and admin) are able to log in to the system using their own registered email and password and prompted to try again if email or password is not correct.



*Diagram 2.1.2 Flow Chart Log in with Own Account*

### **2.1.3 Log out with own account**

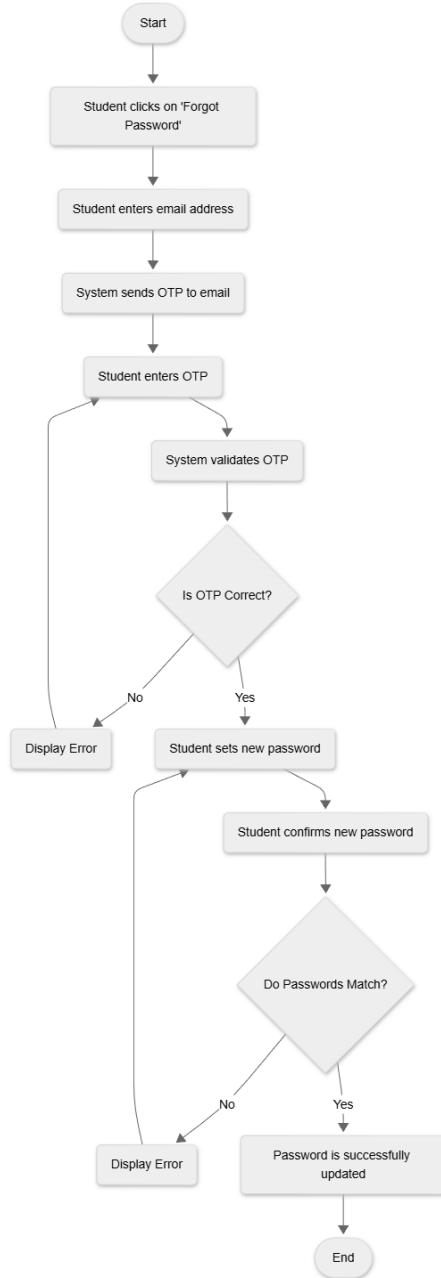
The student is able to log out their own account through the system.



*Diagram 2.1.3 Flow Chart Log out with Own Account*

#### **2.1.4 Reset password**

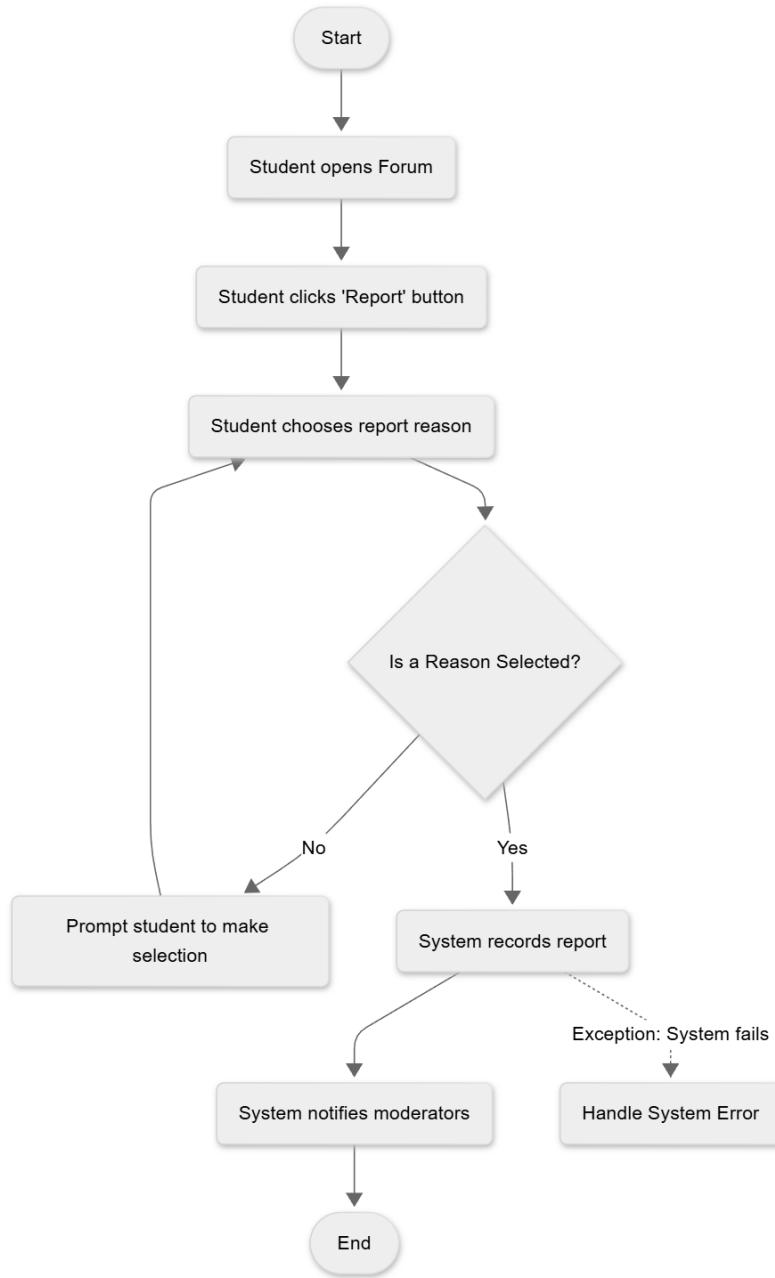
The student is able to reset password by getting one-time password (OTP) through their registered email to get access with their own account again by clicking “forgot password” in the sign in page.



*Diagram 2.1.4 Flow Chart Reset Password*

## 2.1.5 Report contents

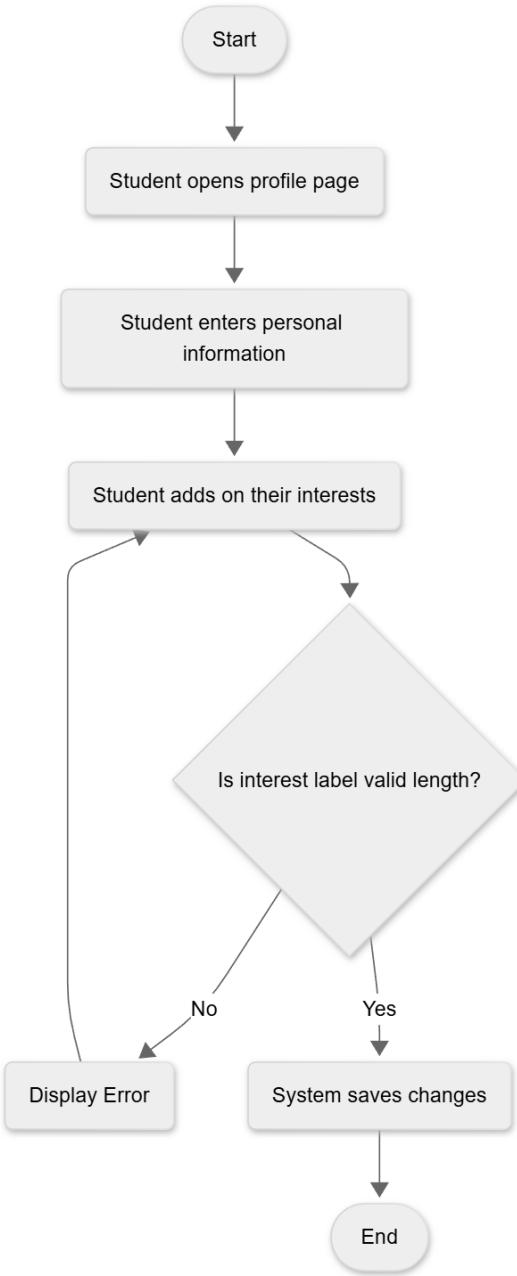
The student is able to report harmful or inappropriate contents including posts and comments in the forum to the moderator through the report button implemented.



*Diagram 2.1.5 Flow Chart Report Contents*

## 2.1.6 Build profile

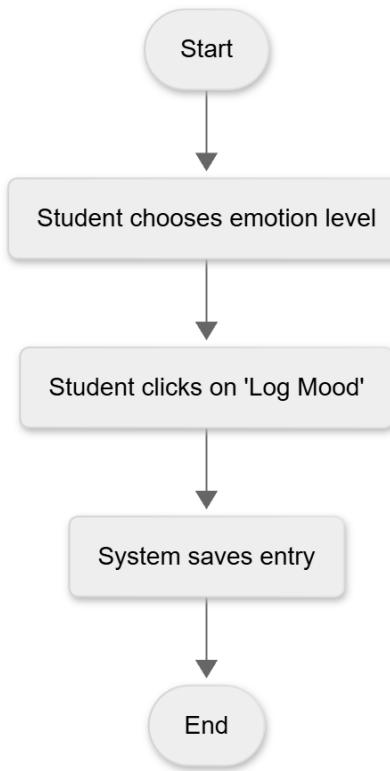
The student is able to build and update their personal profile with details such as interest labels, programme taken, and a short description.



*Diagram 2.1.6 Flow Chart Build Profile*

## **2.1.7 Check-in mood and well-being**

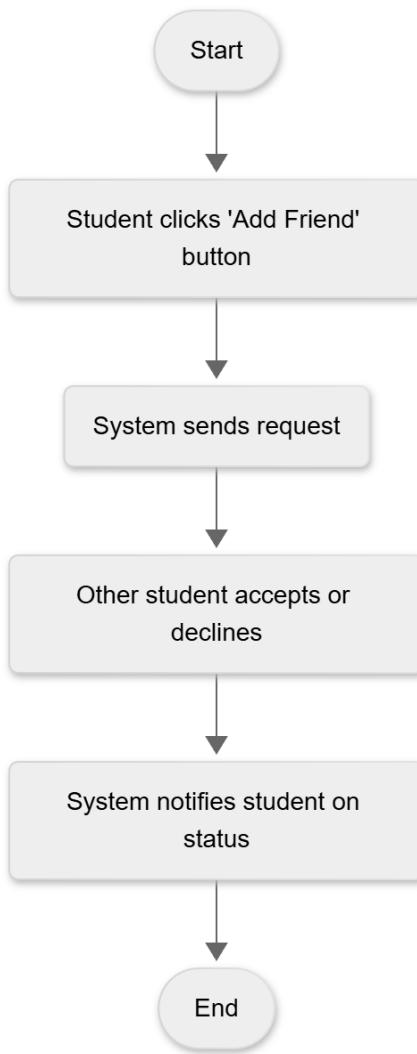
The student is able to check in their daily mood and well-being with available options implemented in the system, and the respond will convert to score, and the score will be recorded in the database.



*Diagram 2.1.7 Flow Chart Check-in Mood and Well-being*

### **2.1.8 Add friends**

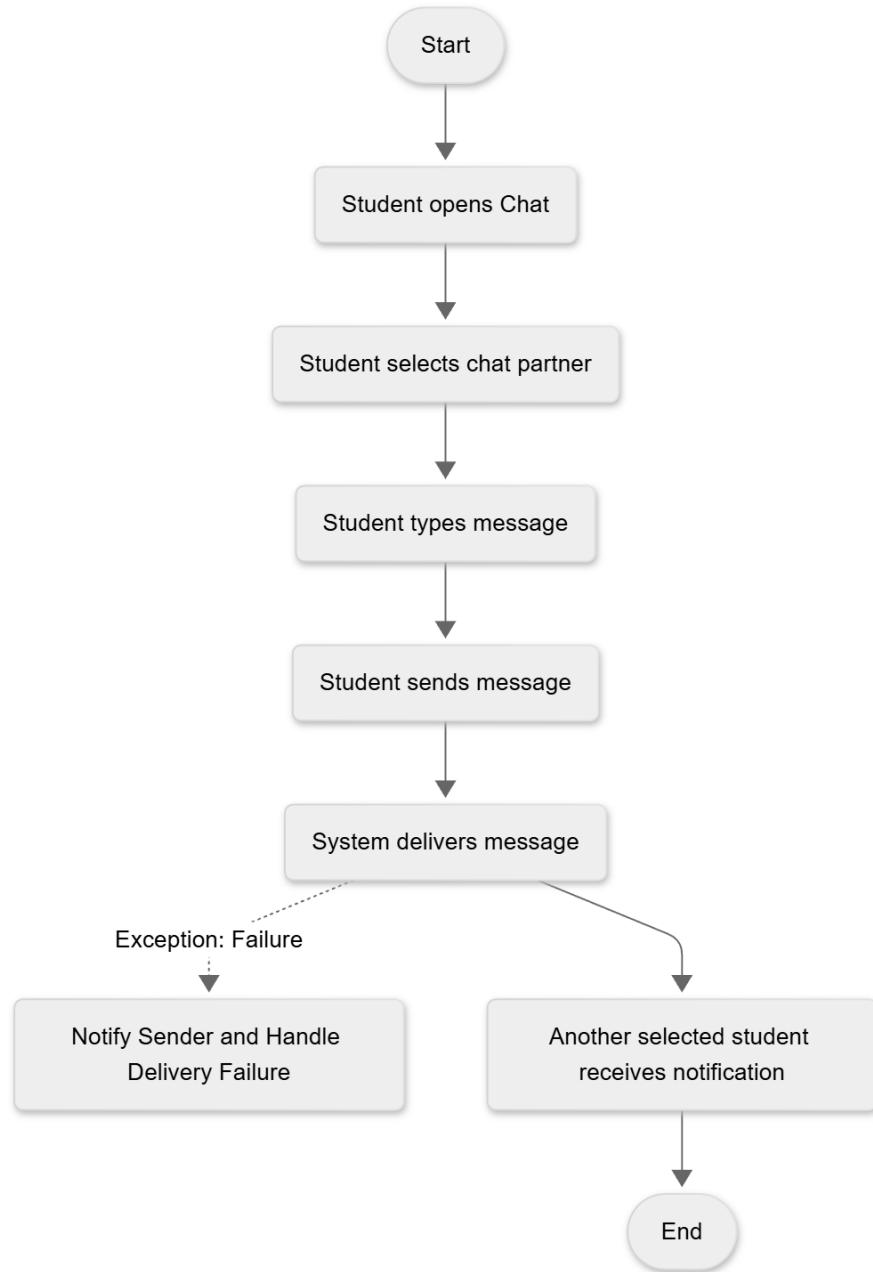
The student is able to maintain social connections through adding new friends by sending and accepting friend requests.



*Diagram 2.1.8 Flow Chart Add Friends*

### **2.1.9 Chat privately**

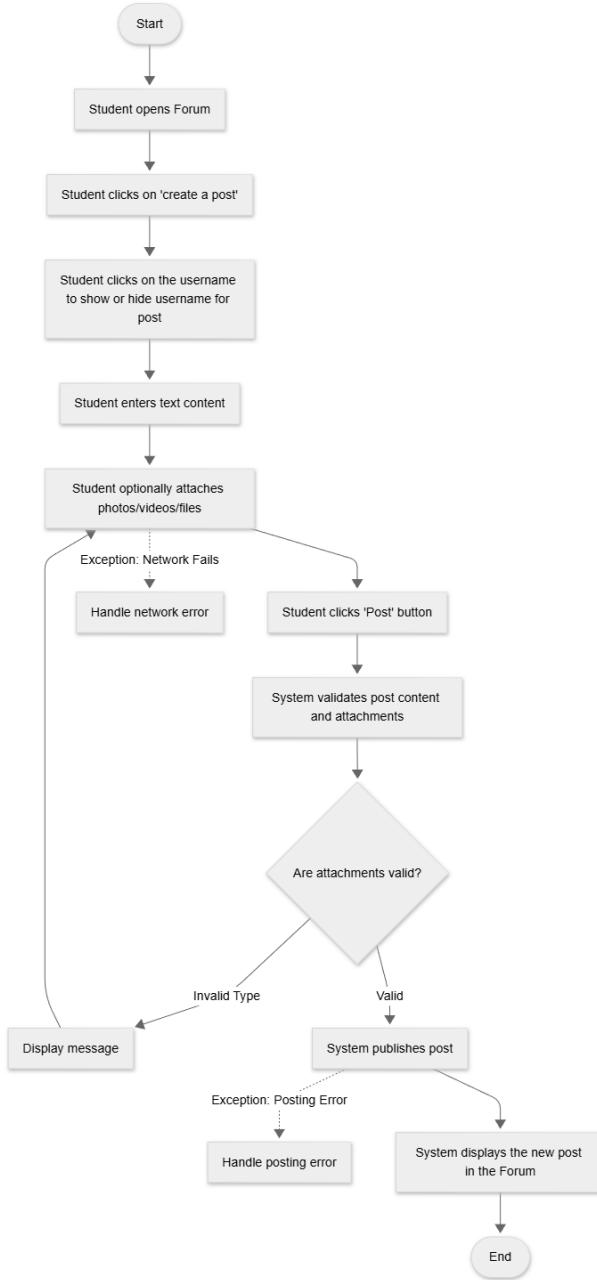
The student is able to maintain social connections through engaging in private chat conversations with anyone including or excluding their friends through others' profile, and chat history is recorded.



*Diagram 2.1.9 Flow Chart Chat Privately*

## 2.1.10 Make posts

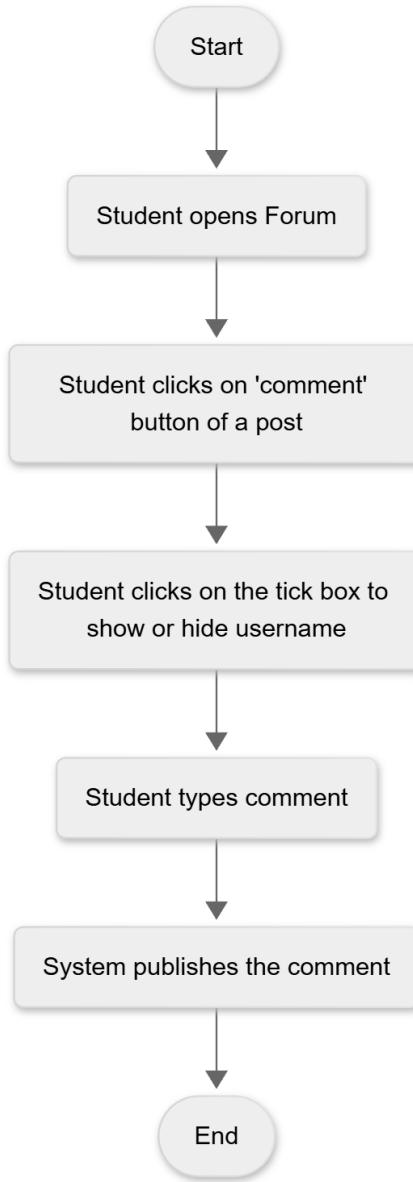
The student is able to create and publish posts in the forum anonymously or non-anonymously (showing the username).



*Diagram 2.1.10 Flow Chart Make Posts*

## 2.1.11 Comment on posts

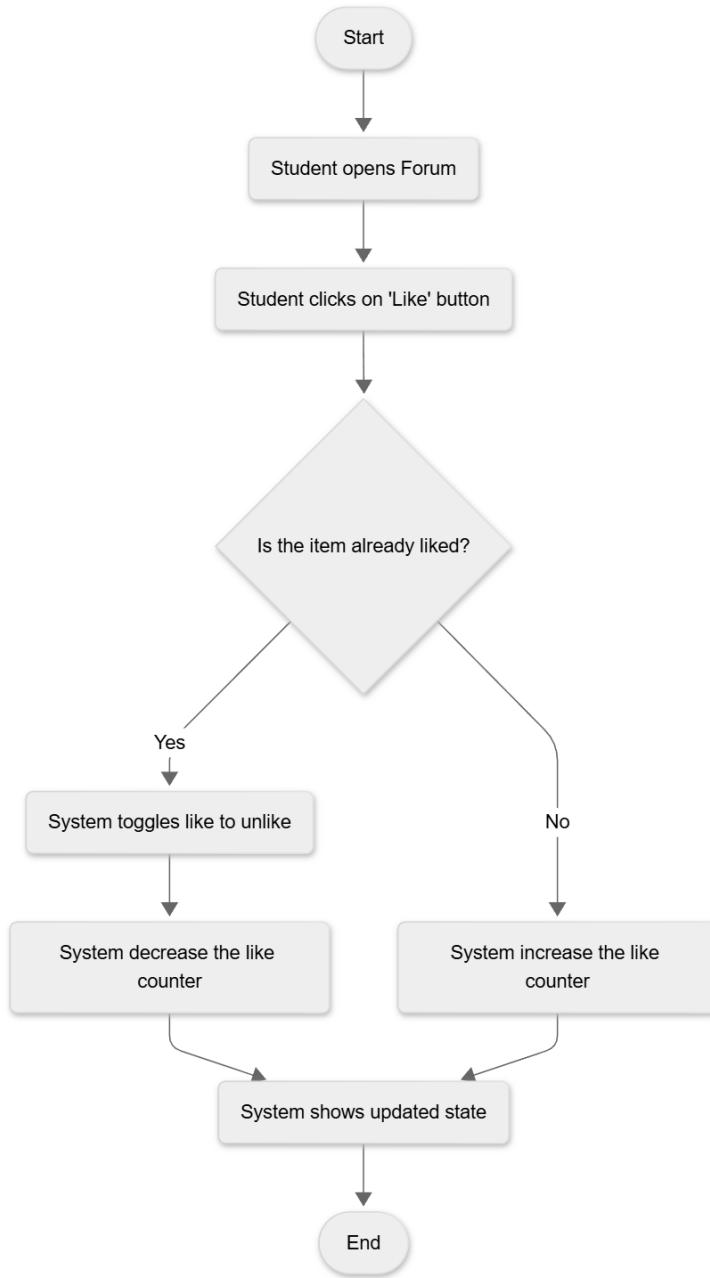
The student is able to expand social connections through commenting anonymously or non-anonymously (showing the username) on the forum posts created by others.



*Diagram 2.1.11 Flow Chart Comment on Posts*

## **2.1.12 Like posts and comments**

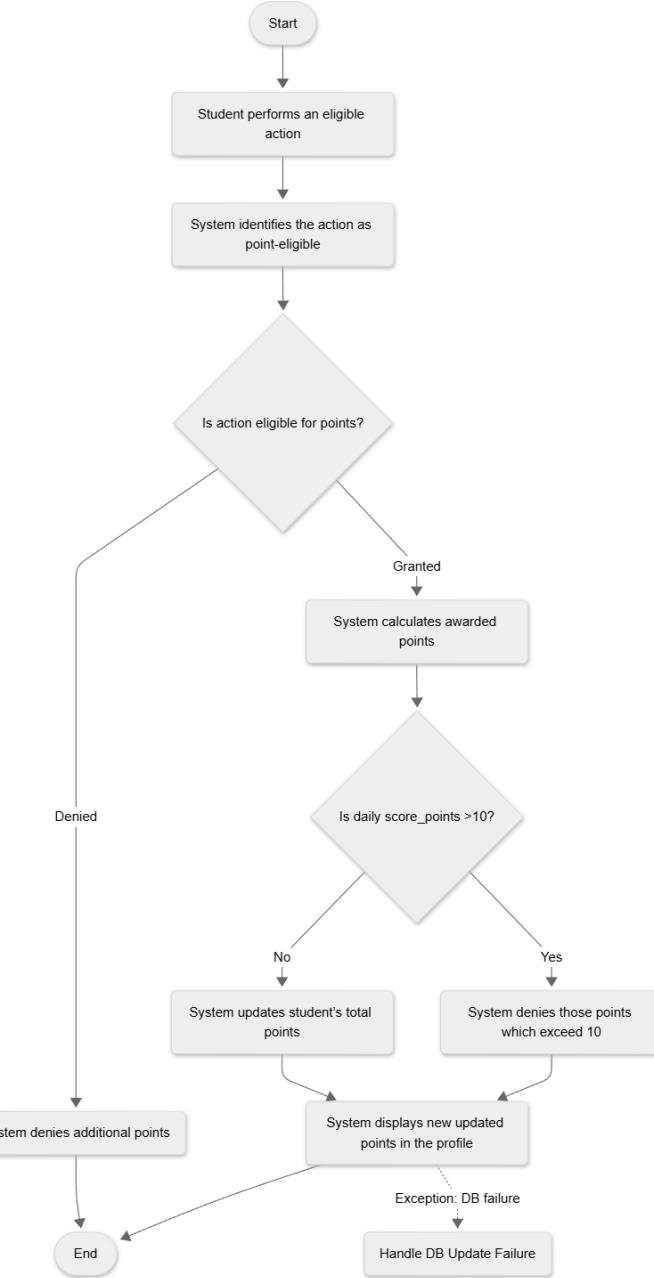
The student is able to expand social connections through like or unlike the posts and comments in the forum.



*Diagram 2.1.12 Flow Chart Like Posts and Comments*

### **2.1.13 Earn points**

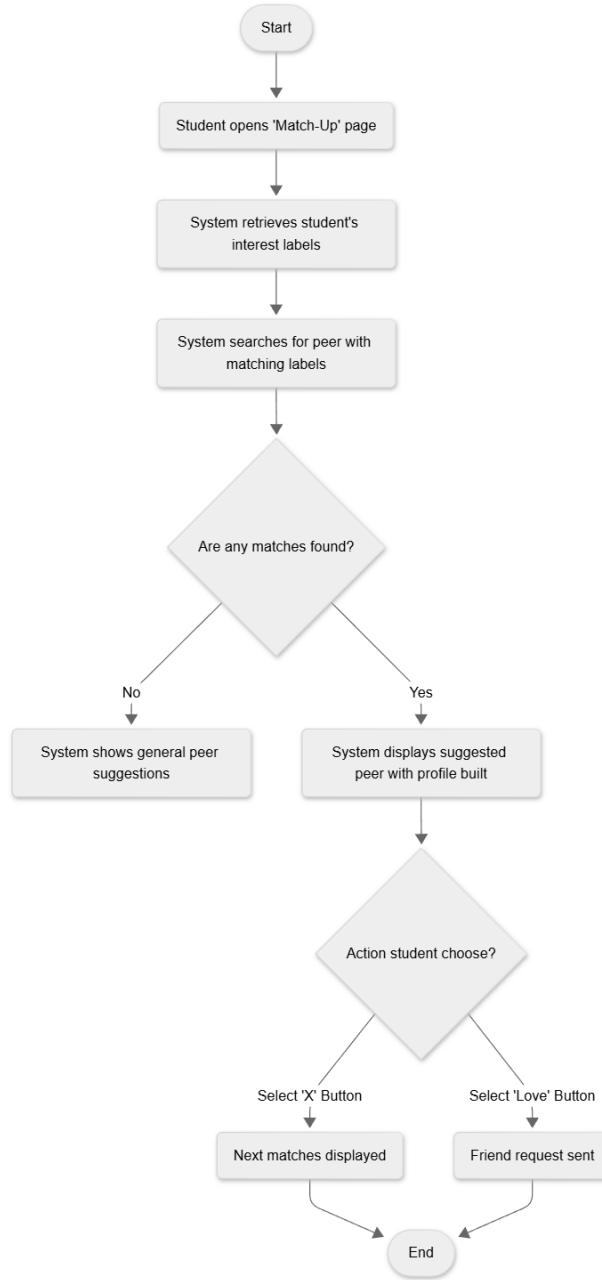
The student is able to earn points to sustain the premium feature of the system through check-in daily mood, or getting likes in a single post or comment published anonymously or non-anonymously in the forum.



**Diagram 2.1.13 Flow Chart Earn Points**

## 2.1.14 Match-up with people

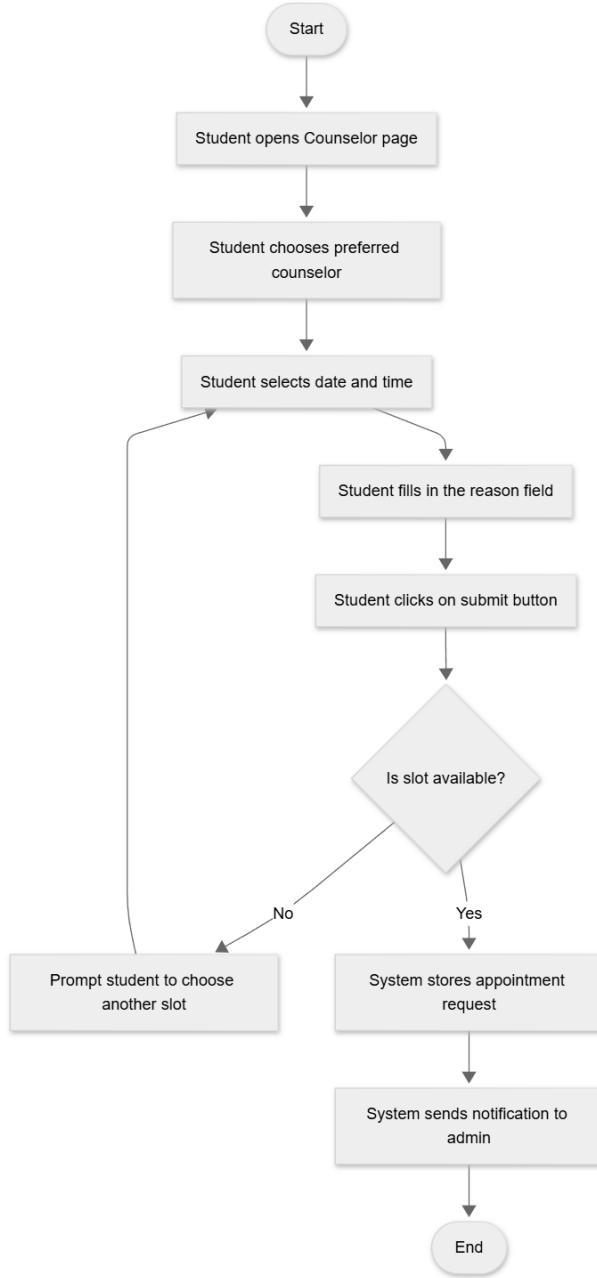
The student is able to expand social connections through matching with other students who having similar interest tags in their profile.



*Diagram 2.1.14 Flow Chart Match-up with People*

## 2.1.15 Book counselor appointments

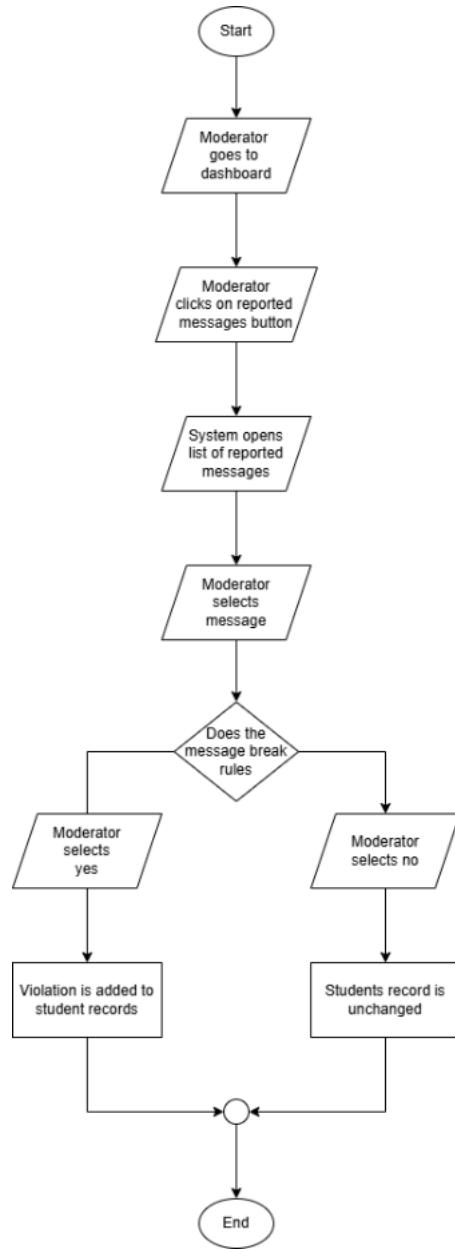
Student is able to book an appointment with a counselor through the system.



*Diagram 2.1.15 Flow Chart Book Counselor Appointments*

## **2.1.16 Review Reported Messages**

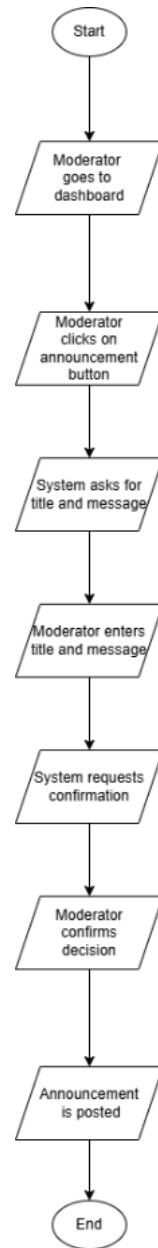
The moderator is able to review messages reported by students



*Diagram 2.1.16 Flow Chart Review Reported Messages*

### **2.1.17 Make Announcements**

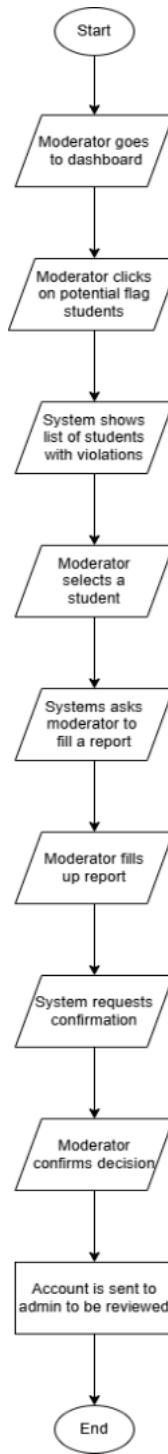
The moderator is able to make announcements that all students will be notified of.



*Diagram 2.1.17 Flow Chart Make Announcements*

### **2.1.18 Flag Student Account**

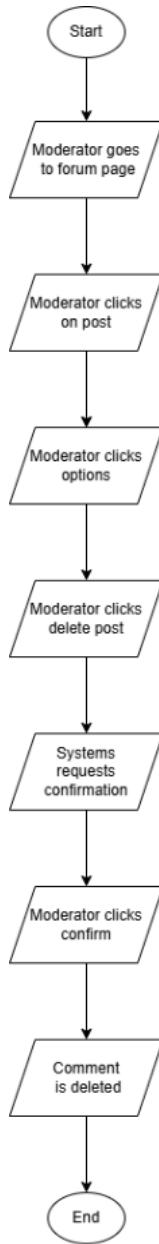
The moderator is able to flag accounts to be reviewed by admins.



*Diagram 2.1.18 Flow Chart Flag Student Account*

### **2.1.19 Delete Forum Comments**

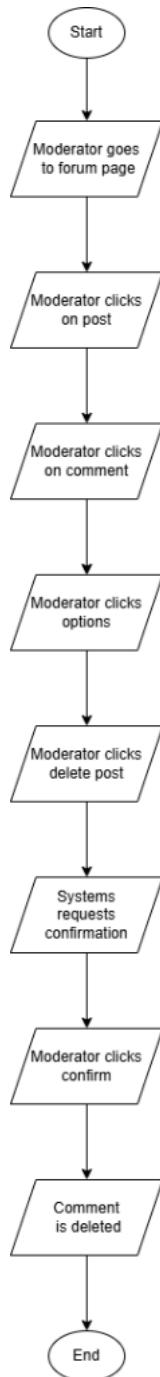
The moderator is able to delete comments under forums posts that break guidelines



*Diagram 2.1.19 Flow Chart Delete Forum Comments*

## **2.1.20 Delete Forum Posts**

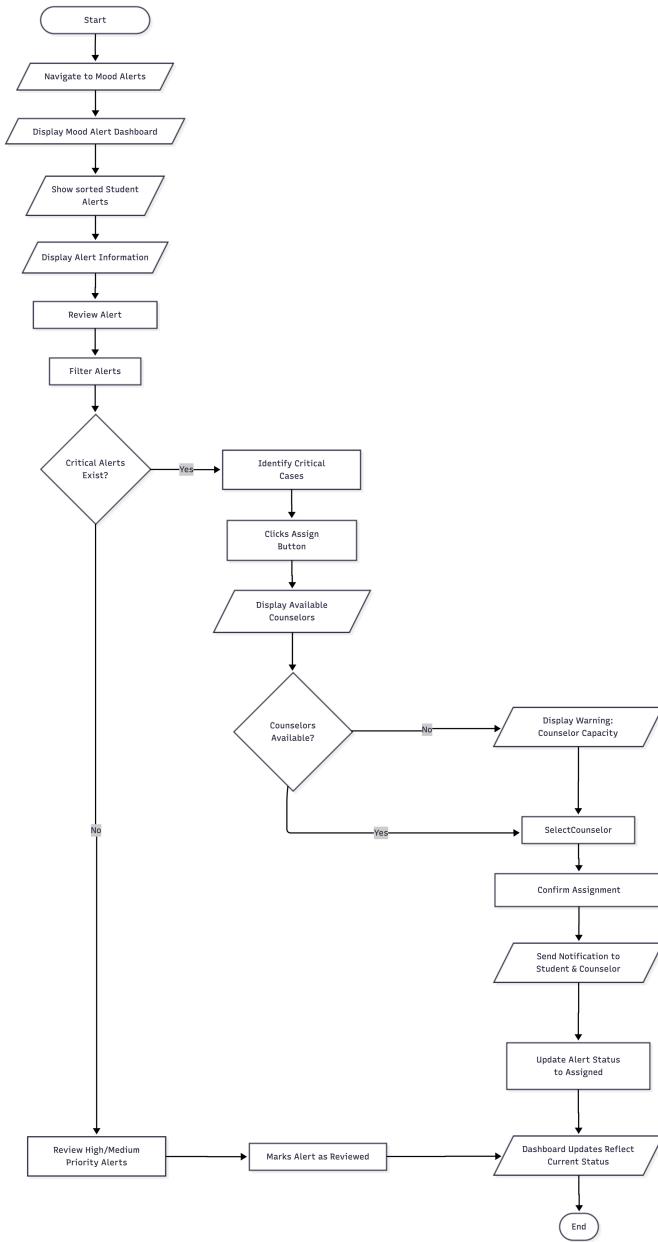
The moderator is able to delete posts that appear on the public forum, preventing inflammatory content from appearing on the site



*Diagram 2.1.20 Flow Chart Delete Forum Posts*

## **2.1.21 Monitor Mood Alert Dashboard**

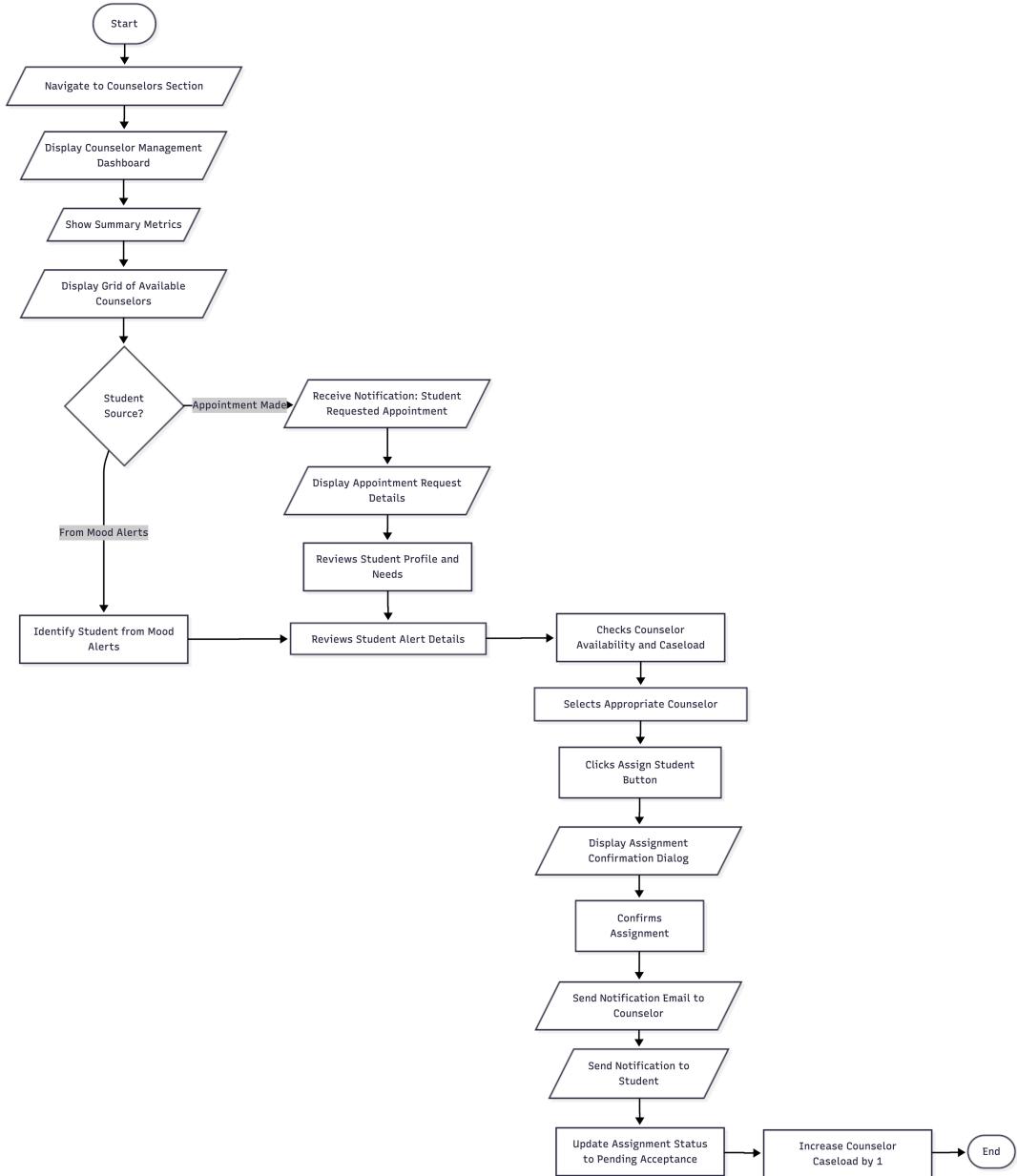
The admin is able to display all mood alerts in prioritized queue sorted by severity level (critical, high, medium, low) showing student identification, weekly mood percentage, trend patterns, and last active timestamp.



*Diagram 2.1.21 Flow Chart Monitor Mood Alert Dashboard*

## 2.1.22 Manage Counselor Assignment

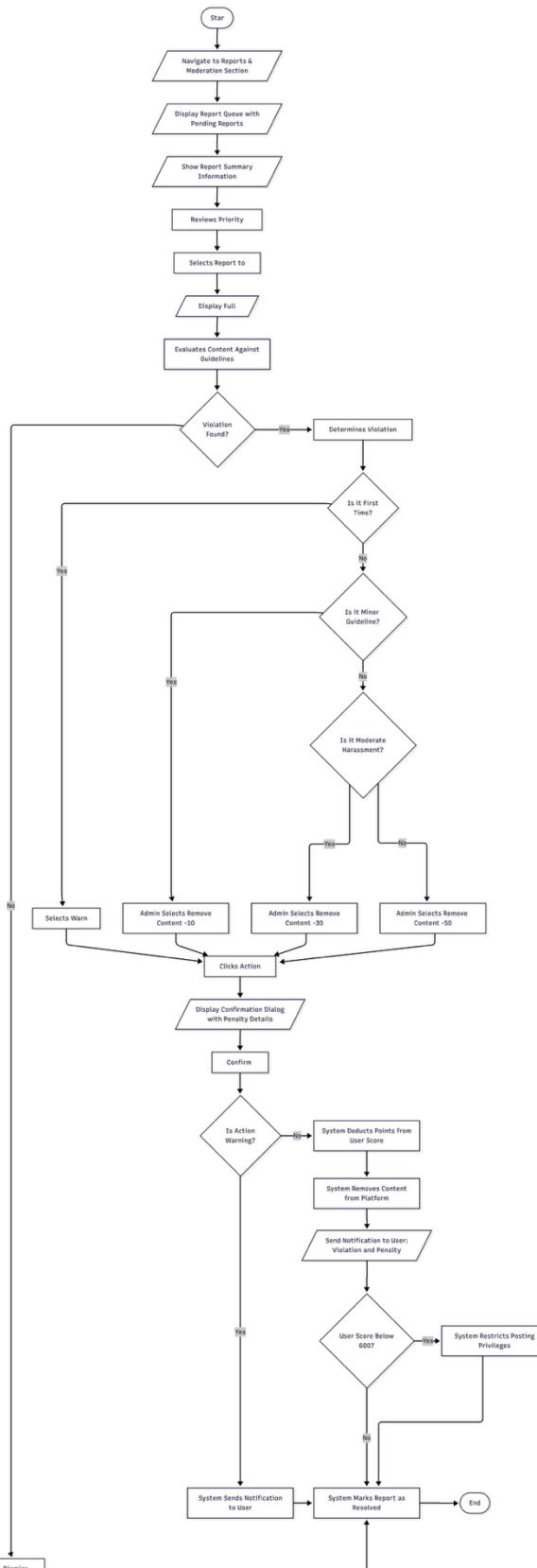
The admin is able to provide student assignment interface displaying available counselors with current caseload count and specializations, send automated notifications to both counselor and student , and track assignment status (pending acceptance, active, on hold, completed).



**Diagram 2.1.22 Flow Chart Manage Counselor Assignments**

### 2.1.23 Review Reported Content

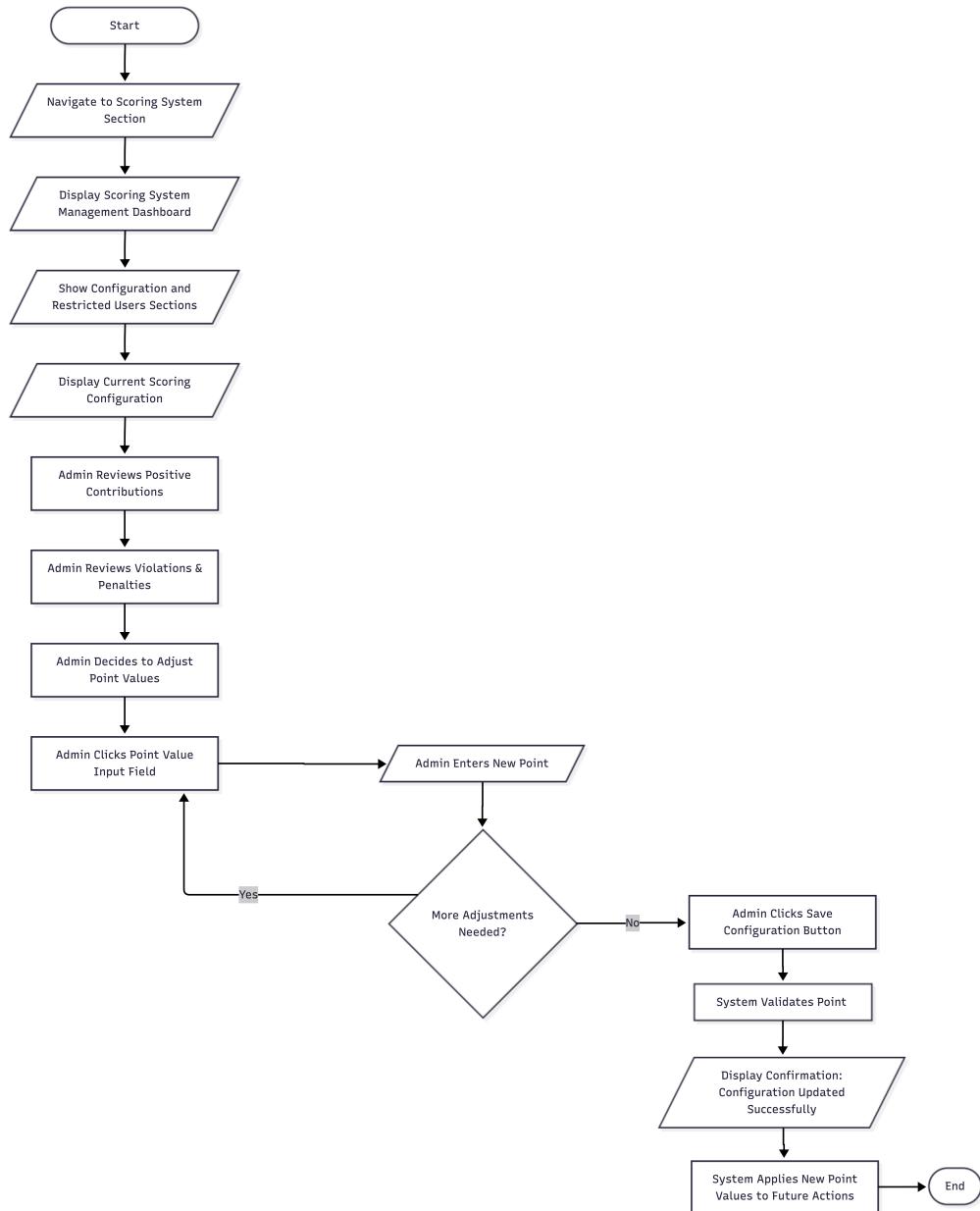
The admin is able to view all reported content with full context including thread conversation and user history, provide moderation actions (dismiss, warn, remove content, apply penalty, suspend), and automatically deduct points when content removal is selected (15 points for guideline violation, 30 points for harassment, 50 points for severe violations).



**Diagram 2.1.23 Flow Chart Review Reported Content**

### 2.1.24 Manage Scoring System

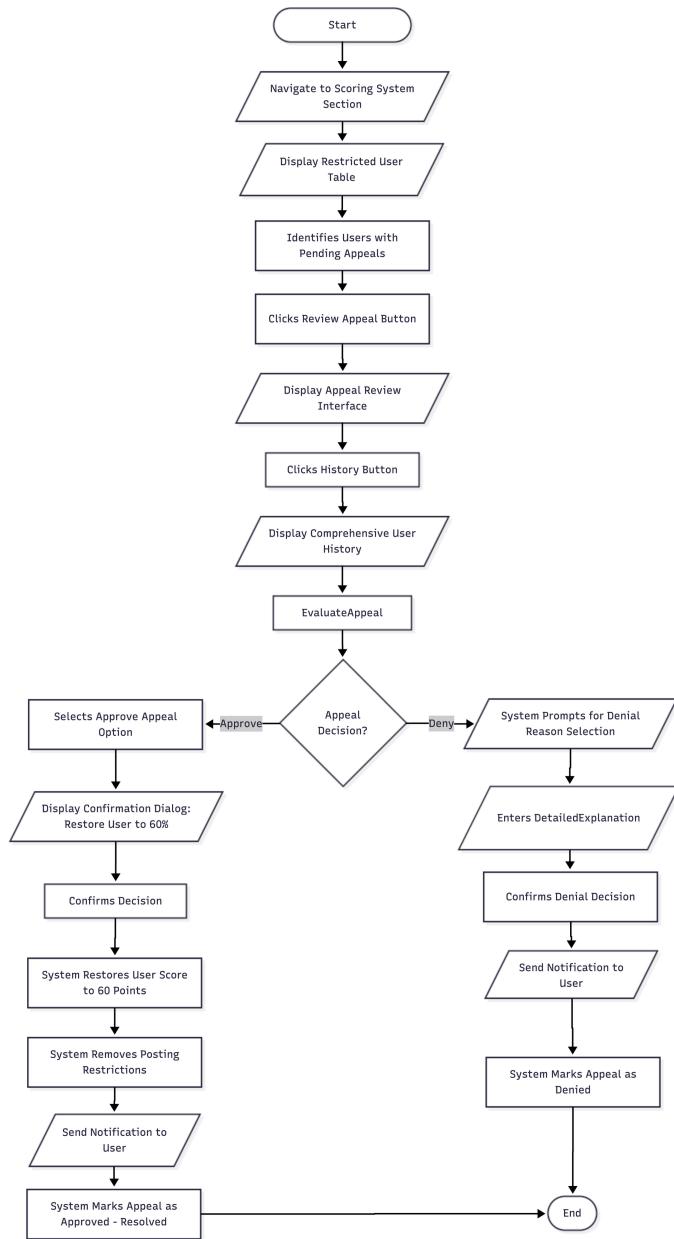
The admin is able to configure point values for positive contributions or violations and manually adjusts scores when necessary to maintain a fair and effective behavioral accountability system.



*Diagram 2.1.24 Flow Chart Manage Scoring System*

### 2.1.25 Process Appeals

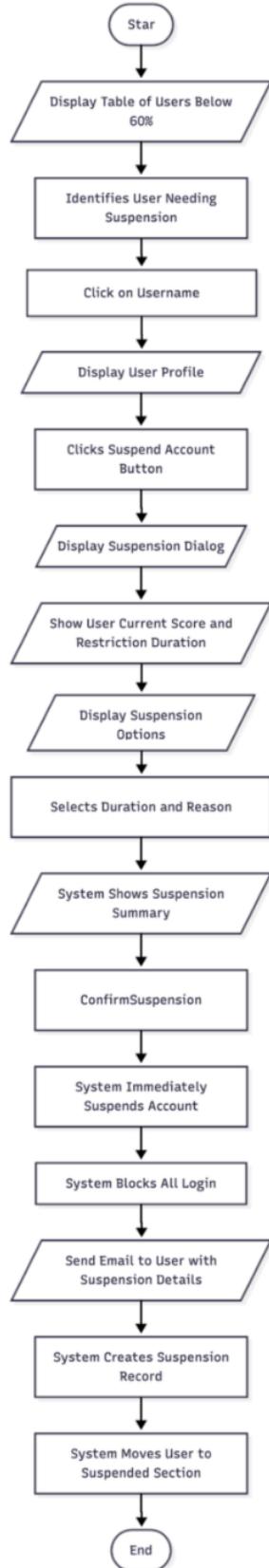
The admin is able to display appeal queue showing user's current score, restriction duration, violation history, and appeal reason, offer decision options (approve with point restoration, deny, deny with extended restriction), and send automated notification within 24 hours of decision with explanation.



*Diagram 2.1.25 Flow Chart Process Appeals*

## 2.1.26 Suspend Accounts

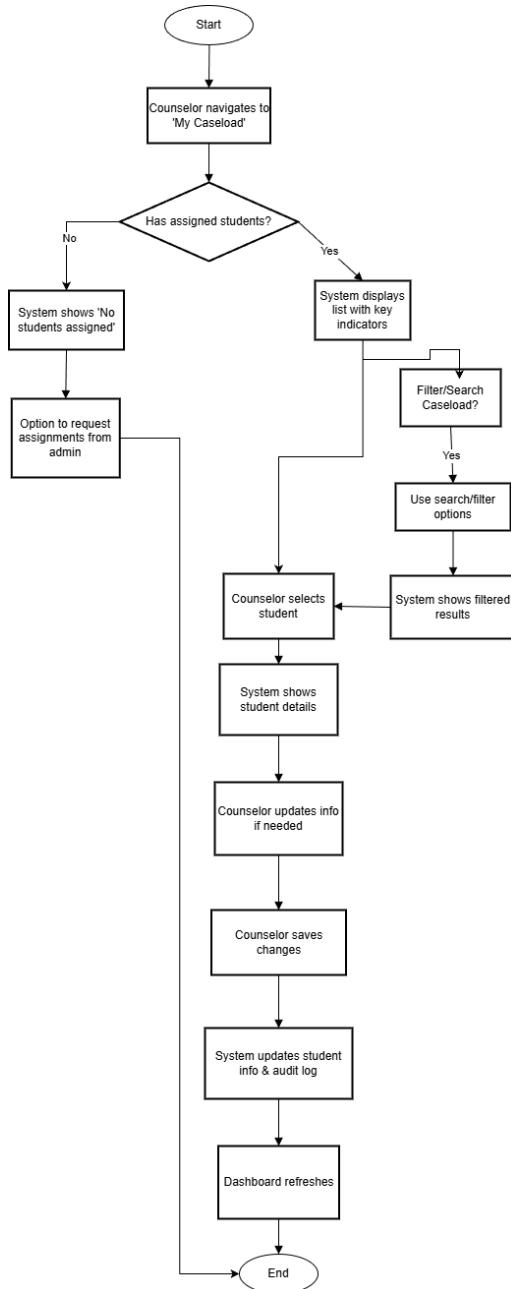
The admin is able to temporarily or permanently suspends user accounts for serious or repeated violations by documenting suspension reasons, applying automatic score penalties, and managing appeal processes to maintain platform safety and community standards.



*Diagram 2.1.26 Flow Chart Suspend Account*

### 2.1.27 manage caseload

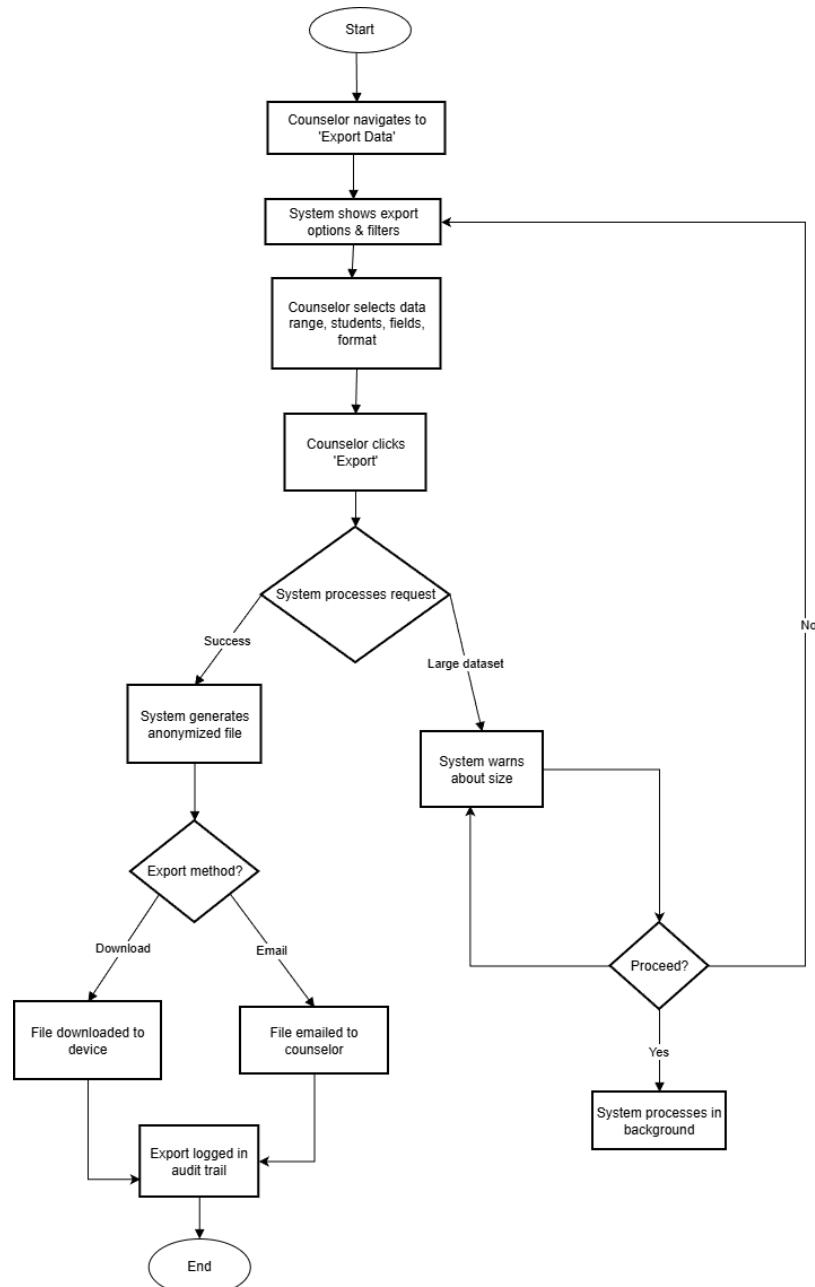
The counselor is able to efficiently organize, view, and update their assigned student roster through an intuitive dashboard that displays student status, priority levels, and recent activities, enabling effective tracking and prioritization of student support needs.



*Diagram 2.1.27 Flow Chart Manage Caseload*

### 2.1.28 Export data

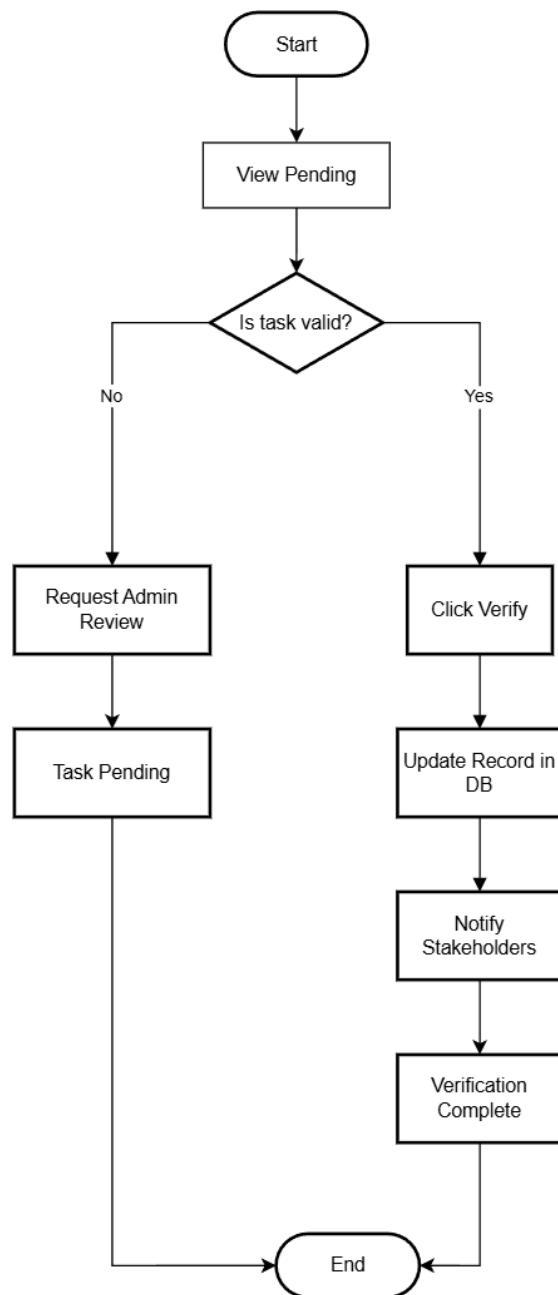
The counselor is able to generate and download structured reports of student well-being data, intervention records, and appointment histories in various formats (Excel, CSV, PDF) while ensuring compliance with data privacy regulations through automatic anonymization of sensitive information.



*Diagram 2.1.28 Flow Chart Export Data*

## 2.1.29 Verify assignments

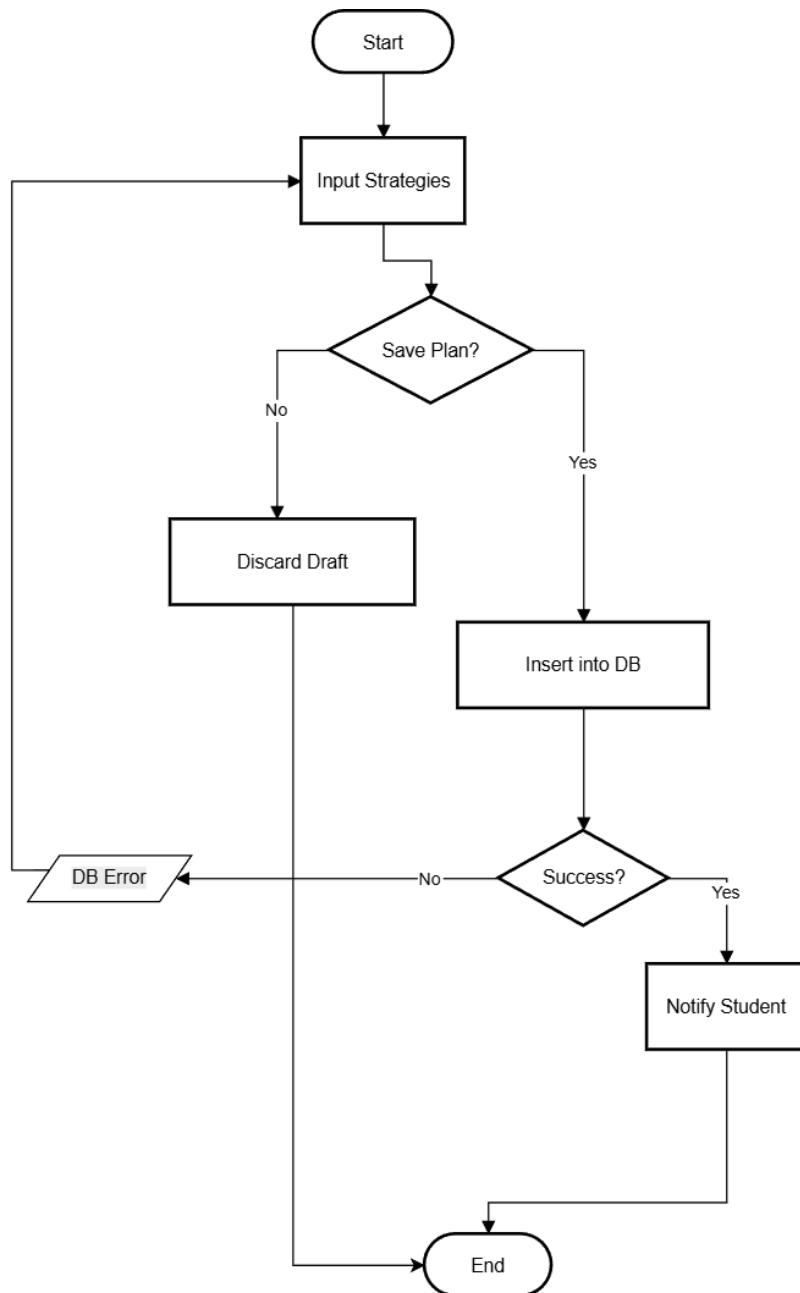
The counselor is able to review and respond to new student assignment requests from administrators by accepting or declining them based on current caseload capacity, with the ability to request additional information before making decisions.



*Diagram 2.1.29 Flow Chart Verify Assignments*

### **2.1.30 Manage action plans**

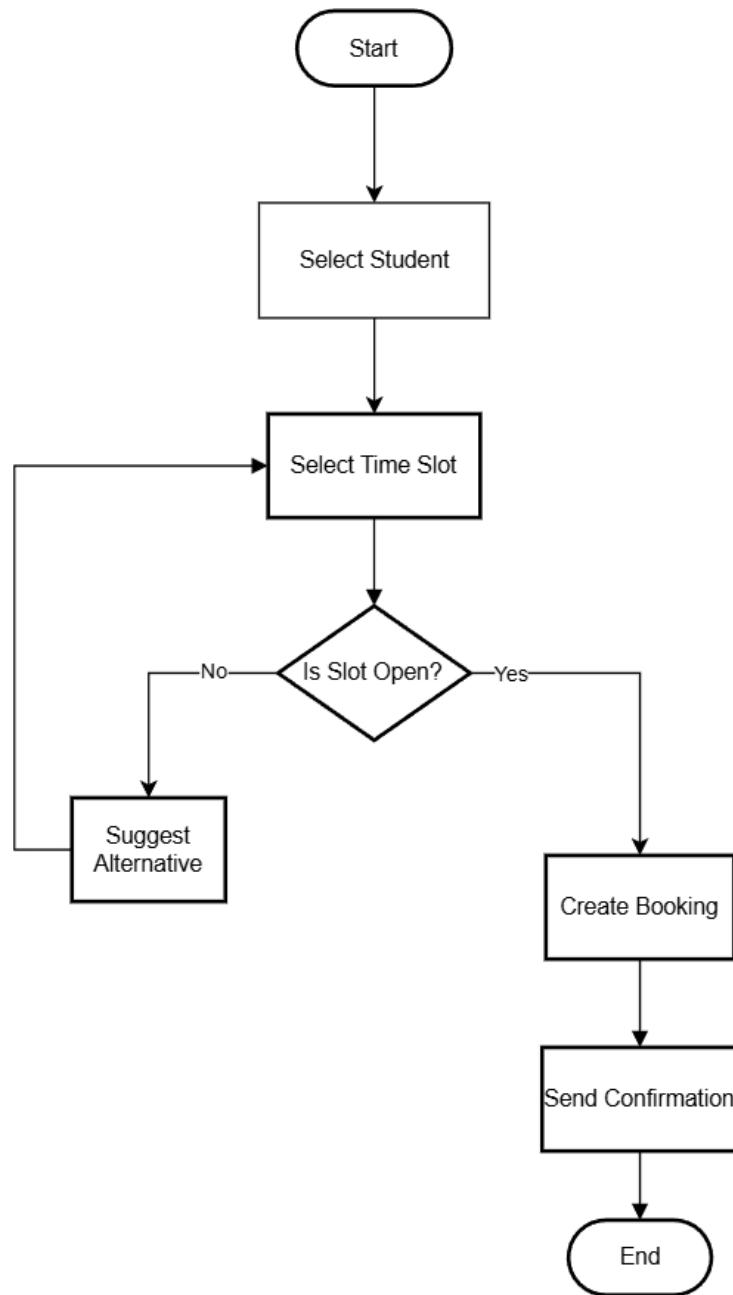
The counselor is able to create, update, and monitor personalized therapeutic action plans for each student, setting measurable goals, defining intervention strategies, and tracking progress toward positive outcomes through collaborative planning with students.



*Diagram 2.1.30 Flow Chart Manage Action Plans*

### **2.1.31 Schedule appointments**

The counselor is able to manage their counseling calendar by setting availability, booking appointments with students, sending automated reminders, and handling rescheduling or cancellations through an integrated scheduling system.



*Diagram 2.1.31 Flow Chart Schedule Appointments*

## 3 Data Design

### 3.1 Design Class Diagram

#### System Architecture: Digital Peer Support System

- User Hierarchy:** The system is built on an inheritance model where **Student**, **Counselor**, **Admin**, and **Moderator** subclasses all inherit authentication data from a central **Account** class.
- Student Centricity:** The **Student** is the primary entity, connecting to two main modules:
  - Social:** Post, Comment, Like, Friendship, and PrivateChat.
  - Wellness:** MoodCheckIn for tracking health and TherapeuticActionPlan for recovery goals.
- Counseling:** **Counselors** interact with students through Assignment (caseload management) and CounselorAppointment(scheduled sessions).
- Moderation & Safety:** **Moderators** handle Reports and issue FlagAccount records, while **Admins** oversee Appeals and user suspensions. ScoreTransaction tracks user reputation points.

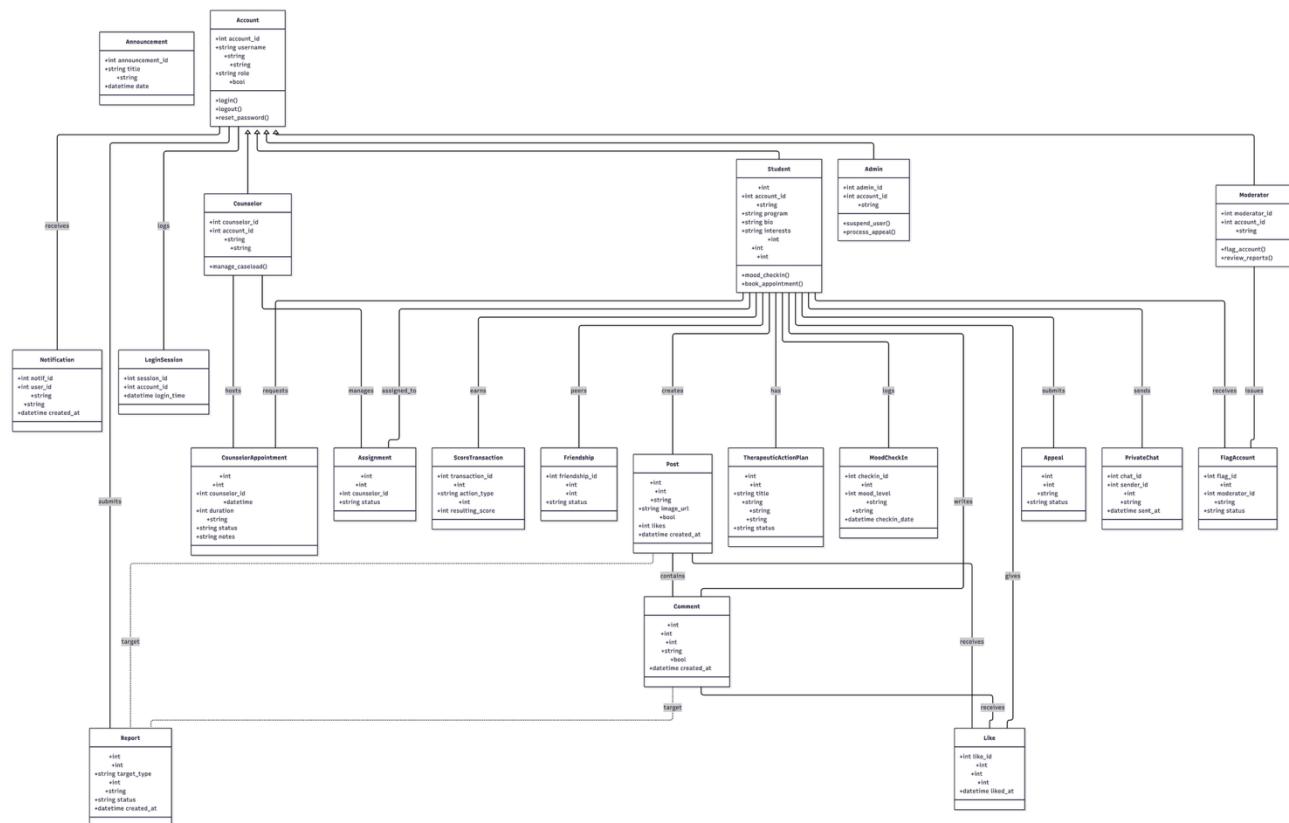


Diagram 3.1.1 [Class Diagram](#)

### 3.2 Data Dictionary

Table Name	Attribute Name	Data Type	Key / Constraint	Description
<b>Account</b>	account_id	INT	PK	Unique identifier for the account.
	email	VARCHAR	UNIQUE	User's email address.
	password	VARCHAR		Hashed password for authentication.
	username	VARCHAR		Unique username for the account.
	created_at	DATETIME		Timestamp when the account was created.
	is_active	BOOLEAN		Status of the account (active/inactive).
<b>Admin</b>	admin_id	INT	PK	Unique identifier for the administrator.
	account_id	INT	FK	Reference to the Account table.
	full_name	VARCHAR		Full name of the administrator.
	department	VARCHAR		Department the administrator belongs to.
<b>Student</b>	student_id	INT	PK	Unique identifier for the student.
	account_id	INT	FK	Reference to the Account table.
	full_name	VARCHAR		Full name of the student.

	program	VARCHAR		Academic program/course the student is enrolled in.
	total_points	INT		Current gamification points balance.
	score_percenta ge	FLOAT		Overall engagement or health score percentage.
<b>Moderator</b>	moderator_id	INT	PK	Unique identifier for the moderator.
	account_id	INT	FK	Reference to the Account table.
	full_name	VARCHAR		Full name of the moderator.
	department	VARCHAR		Department the moderator belongs to.
<b>Counselor</b>	counselor_id	INT	PK	Unique identifier for the counselor.
	account_id	INT	FK	Reference to the Account table.
	full_name	VARCHAR		Full name of the counselor.
	specialization	VARCHAR		Area of counseling expertise.
	current_caseload	INT		Number of active students currently assigned.
<b>Profile</b>	profile_id	INT	PK	Unique identifier for the profile.

*Software Design Specification for Digital Peer Support System (Version 2.0)*

	student_id	INT	FK	Reference to the Student table.
	interests	TEXT		List or description of student interests.
	introduction	TEXT		Short bio or self-introduction.
	program	VARCHAR		(Redundant if in Student table) Program details.
	updated_at	DATETIME		Timestamp of the last profile update.
<b>LoginSession</b>	session_id	INT	PK	Unique identifier for the login session.
	account_id	INT	FK	Reference to the Account table.
	login_time	DATETIME		Timestamp when the user logged in.
	is_active	BOOLEAN		Indicates if the session is currently active.
<b>LogoutSession</b>	logout_id	INT	PK	Unique identifier for the logout record.
	account_id	INT	FK	Reference to the Account table.
	logout_time	DATETIME		Timestamp when the user logged out.
<b>PasswordReset</b>	reset_id	INT	PK	Unique identifier for the reset request.

	student_id	INT	FK	Reference to the Student table.
	otp_code	VARCHAR		One-time password code for verification.
	requested_at	DATETIME		Timestamp when the reset was requested.
	expires_at	DATETIME		Timestamp when the OTP expires.
	new_pass	VARCHAR		Temporary or new password hash.
	is_used	BOOLEAN		Flag if the OTP has been used.
<b>MoodCheckIn</b>	checkin_id	INT	PK	Unique identifier for the mood entry.
	student_id	INT	FK	Reference to the Student table.
	mood_level	INT		Numeric representation of mood (e.g., 1-5).
	mood_score	FLOAT		Calculated score derived from the mood level.
	checkin_date	DATETIME		Date and time of the check-in.
	notes	TEXT		Optional journal entry or comments.
<b>MoodAlert</b>	alert_id	INT	PK	Unique identifier for the alert.

*Software Design Specification for Digital Peer Support System (Version 2.0)*

	student_id	INT	FK	Reference to the Student table.
	admin_id	INT	FK	Reference to the Admin reviewing the alert.
	weekly_avg_mood	FLOAT		Average mood score over the past week.
	severity_level	VARCHAR		Level of concern (e.g., Low, High, Critical).
	weeks_below_threshold	INT		Count of consecutive weeks with low scores.
	assignment_status	VARCHAR		Status of intervention assignment.
	created_at	DATETIME		Timestamp when the alert was generated.
	reviewed_at	DATETIME		Timestamp when the alert was reviewed.
<b>Post</b>	post_id	INT	PK	Unique identifier for the post.
	student_id	INT	FK	Reference to the Student table.
	content	TEXT		Main text content of the post.
	is_anonymous	BOOLEAN		Flag if the post is anonymous.
	like_count	INT		Total number of likes on the post.

	created_at	DATETIME		Timestamp when the post was created.
	updated_at	DATETIME		Timestamp when the post was last edited.
<b>Comment</b>	comment_id	INT	PK	Unique identifier for the comment.
	post_id	INT	FK	Reference to the Post table.
	student_id	INT	FK	Reference to the Student table.
	content	TEXT		Text content of the comment.
	is_anonymous	BOOLEAN		Flag if the comment is anonymous.
	like_count	INT		Total number of likes on the comment.
	created_at	DATETIME		Timestamp when the comment was created.
<b>Like</b>	like_id	INT	PK	Unique identifier for the like action.
	student_id	INT	FK	Reference to the Student table.
	post_id	INT	FK	Nullable reference to the Post table.
	comment_id	INT	FK	Nullable reference to the Comment table.

**Software Design Specification for Digital Peer Support System (Version 2.0)**

	liked_at	DATETIME		Timestamp when the like occurred.
<b>Report</b>	report_id	INT	PK	Unique identifier for the report.
	student_id	INT	FK	Reference to the reporting Student.
	admin_id	INT	FK	Reference to the reviewing Admin.
	moderator_id	INT	FK	Reference to the reviewing Moderator.
	post_id	INT	FK	Reference to reported Post (nullable).
	comment_id	INT	FK	Reference to reported Comment (nullable).
	violation_type	VARCHAR		Category of violation (e.g., harassment).
	reason	TEXT		Detailed reason for the report.
	priority_level	VARCHAR		Priority of the report (e.g., High, Low).
	status	VARCHAR		Status (e.g., Pending, Resolved).
	submitted_at	DATETIME		Timestamp when report was submitted.
	reviewed_at	DATETIME		Timestamp when report was reviewed.

*Software Design Specification for Digital Peer Support System (Version 2.0)*

<b>Assignment</b>	assignment_id	INT	PK	Unique identifier for the assignment case.
	admin_id	INT	FK	Reference to the Admin creating assignment.
	student_id	INT	FK	Reference to the Student being assigned.
	counselor_id	INT	FK	Reference to the Counselor assigned.
	assigned_at	DATETIME		Timestamp of assignment.
	status	VARCHAR		Status of the case (e.g., Open, Closed).
	assignment_notes	TEXT		Administrative notes regarding the assignment.
<b>Appeal</b>	appeal_id	INT	PK	Unique identifier for the appeal.
	student_id	INT	FK	Reference to the Student appealing.
	admin_id	INT	FK	Reference to the reviewing Admin.
	explanation	TEXT		Student's explanation for the appeal.
	status	VARCHAR		Status of the appeal (e.g., Approved, Denied).
	admin_decision	TEXT		Admin's final decision text.

	submitted_at	DATETIME		Timestamp of submission.
	reviewed_at	DATETIME		Timestamp of review.
<b>Suspension</b>	suspension_id	INT	PK	Unique identifier for the suspension.
	student_id	INT	FK	Reference to the Student suspended.
	admin_id	INT	FK	Reference to the issuing Admin.
	duration	VARCHAR		Duration string (e.g., "7 days").
	reason	VARCHAR		Short reason code or title.
	violation_type	VARCHAR		Type of violation causing suspension.
	justification	TEXT		Detailed justification for the suspension.
	start_date	DATETIME		Start date of suspension.
	end_date	DATETIME		End date of suspension.
	is_active	BOOLEAN		Flag if suspension is currently active.
<b>ScoreTransaction</b>	transaction_id	INT	PK	Unique identifier for the point transaction.
	student_id	INT	FK	Reference to the Student.
	action_type	VARCHAR		Type of action (e.g., "Post", "Helpful Comment").

	points_change	INT		Number of points added or subtracted.
	resulting_score	INT		Total score after transaction.
	transaction_date	DATETIME		Timestamp of the transaction.
<b>RestrictionPeriod</b>	restriction_id	INT	PK	Unique identifier for restriction.
	student_id	INT	FK	Reference to the Student.
	start_date	DATETIME		Start of restriction.
	end_date	DATETIME		End of restriction.
	violation_count	INT		Count of violations leading to restriction.
	is_active	BOOLEAN		Flag if restriction is active.
<b>CounselorAppointment</b>	appointment_id	INT	PK	Unique identifier for the appointment.
	student_id	INT	FK	Reference to the Student.
	counselor_id	INT	FK	Reference to the Counselor.
	admin_id	INT	FK	Reference to Admin (if they booked it).
	appointment_date	DATETIME		Date of appointment.
	appointment_time	DATETIME		Time of appointment.
	reason	TEXT		Reason for the appointment.

	status	VARCHAR		Status (e.g., Scheduled, Completed).
<b>FlagAccount</b>	flag_id	INT	PK	Unique identifier for the flag.
	moderator_id	INT	FK	Reference to the Moderator flagging.
	student_id	INT	FK	Reference to the Student flagged.
	admin_id	INT	FK	Reference to Admin reviewing flag.
	violation_type	VARCHAR		Type of violation detected.
	evidence	TEXT		Links or text evidence of violation.
	justification	TEXT		Moderator's notes on why to flag.
	status	VARCHAR		Status of flag review.
	submitted_at	DATETIME		Timestamp of submission.
	reviewed_at	DATETIME		Timestamp of review.
<b>Announcement</b>	announcement_id	INT	PK	Unique identifier for the announcement.
	moderator_id	INT	FK	Reference to the creating Moderator.
	title	VARCHAR		Headline of the announcement.
	content	TEXT		Body text of the announcement.

*Software Design Specification for Digital Peer Support System (Version 2.0)*

	category	VARCHAR		Category (e.g., "Maintenance", "Event").
	display_duration_days	INT		Number of days to show the announcement.
	published_at	DATETIME		Timestamp when published.
	expires_at	DATETIME		Timestamp when it expires.
<b>PostDeletion</b>	deletion_id	INT	PK	Unique identifier for deletion record.
	moderator_id	INT	FK	Reference to the Moderator deleting.
	post_id	INT	FK	Reference to the deleted Post.
	reason	VARCHAR		Reason for deletion.
	deleted_at	DATETIME		Timestamp of deletion.
<b>CommentDeletion</b>	deletion_id	INT	PK	Unique identifier for deletion record.
	moderator_id	INT	FK	Reference to the Moderator deleting.
	comment_id	INT	FK	Reference to the deleted Comment.
	reason	VARCHAR		Reason for deletion.
	deleted_at	DATETIME		Timestamp of deletion.
<b>ExportData</b>	export_id	INT	PK	Unique identifier for the export data.

**Software Design Specification for Digital Peer Support System (Version 2.0)**

	counselor_id	INT	FK	Reference to the Counselor.
	export_date	DATETIME	FK	date when the export data happen.
	file_path	STRING		File path like pdf or words.
<b>TherapeuticActionPlan</b>	plan_id	INT	PK	Unique identifier for the plan.
	counselor_id	INT	FK	Reference to the Counselor.
	student_id	INT	FK	Reference to the Student.
	plan_title	VARCHAR		Title of the action plan.
	objectives	TEXT		Goals/Objectives of the plan.
	tasks	TEXT		Specific tasks assigned to the student.
	start_date	DATETIME		Start date of the plan.
	end_date	DATETIME		Target end date.
	status	VARCHAR		Status (e.g., Active, Completed).
<b>Friendship</b>	friendship_id	INT	PK	Unique identifier for friendship link.
	student_id_1	INT	FK	Reference to the initiating Student.
	student_id_2	INT	FK	Reference to the receiving Student.
	status	VARCHAR		Status (e.g., Pending, Accepted).

*Software Design Specification for Digital Peer Support System (Version 2.0)*

	requested_at	DATETIME		Timestamp of request.
	accepted_at	DATETIME		Timestamp of acceptance.
<b>PrivateChat</b>	chat_id	INT	PK	Unique identifier for the chat message.
	student_id_1	INT	FK	Reference to the sender Student.
	student_id_2	INT	FK	Reference to the receiver Student.
	message	TEXT		Content of the chat message.
	sent_at	DATETIME		Timestamp when sent.
	is_read	BOOLEAN		Flag if the message was read.
<b>PeerMatch</b>	match_id	INT	PK	Unique identifier for the match.
	student_id_1	INT	FK	Reference to the first Student.
	student_id_2	INT	FK	Reference to the second Student.
	compatibility_score	FLOAT		Calculated score of compatibility.
	matched_at	DATETIME		Timestamp when match was created.
	status	VARCHAR		Status of the match (e.g., Suggested).

### 3.3 Data Structures

#### 3.3.1 Data Structure 1 : User Identity & Management

This structure manages authentication, roles, and user profiles. It centers around the **Account** table, which feeds into specific role tables.

Table Name	Description	Key Attributes
<b>Account</b>	The central entity for authentication and login status.	account_id (PK), email, password, is_active
<b>Student</b>	Stores gamification scores (total_points), health scores, and academic details for the primary users.	student_id (PK), account_id (FK), score_percentage, total_points
<b>Counselor</b>	Stores professional details and caseload capacity for staff.	counselor_id (PK), specialization, current_caseload
<b>Admin</b>	Managing administrators and their departments.	admin_id (PK), department
<b>Moderator</b>	Content overseers responsible for flagging and deleting violations.	moderator_id (PK), department
<b>Profile</b>	Extended student details visible to peers (bio, interests).	profile_id (PK), interests, introduction

#### 3.3.2 Data Structure 2 : Mental Health & Counseling

This structure supports the clinical workflow, from mood tracking to professional intervention and action plans.

Table Name	Description	Key Attributes
<b>MoodCheckIn</b>	Daily student logs used to calculate wellness scores.	checkin_id (PK), mood_level, mood_score, checkin_date
<b>MoodAlert</b>	System-generated alerts when a student's mood drops below the threshold.	alert_id (PK), severity_level, weeks_below_threshold
<b>Assignment</b>	Links a high-risk student to a counselor for long-term care.	assignment_id (PK), student_id (FK), counselor_id (FK), status
<b>CounselorAppointment</b>	Manages booking slots for real-time sessions.	appointment_id (PK), appointment_date, status, reason
<b>TherapeuticActionPlan</b>	Long-term goals and tasks assigned by counselors to students.	plan_id (PK), objectives, tasks, start_date, end_date

<b>ExportData</b>	Logs of caseload reports generated by counselors.	export_id (PK), file_path, export_date
-------------------	---	--

### 3.3.3 Data Structure 3 : Community & Social Interaction

This structure manages peer-to-peer interactions, including the forum, friendships, and chatting.

Table Name	Description	Key Attributes
<b>Post</b>	User-generated content shared on the public feed.	post_id (PK), content, is_anonymous, like_count
<b>Comment</b>	Replies to posts.	comment_id (PK), post_id (FK), content
<b>Like</b>	Tracking engagement on posts and comments.	like_id (PK), post_id (FK), comment_id (FK)
<b>Friendship</b>	Tracks peer connection status (Pending/Accepted).	friendship_id (PK), student_id_1, student_id_2, status
<b>PeerMatch</b>	Suggested connections based on compatibility scores.	match_id (PK), compatibility_score, status
<b>PrivateChat</b>	Direct messages between connected friends.	chat_id (PK), message, is_read, sent_at

### 3.3.4 Data Structure 4 : Safety, Moderation & Governance

This structure manages peer-to-peer interactions, including the forum, friendships, and chatting.

Table Name	Description	Key Attributes
<b>Report</b>	Flags submitted by users against specific content.	report_id (PK), violation_type, priority_level, status
<b>FlagAccount</b>	Moderator-level flags against specific user accounts.	flag_id (PK), moderator_id (FK), violation_type, evidence
<b>Suspension</b>	Records of students temporarily or permanently banned.	suspension_id (PK), duration, reason, is_active
<b>RestrictionPeriod</b>	Logs of "Read-Only" mode applied to low-score users.	restriction_id (PK), start_date, end_date, violation_count
<b>Appeal</b>	Requests by students to overturn restrictions.	appeal_id (PK), explanation, admin_decision, status
<b>Post/CommentDeletion</b>	Logs of content removed by moderators for auditing.	deletion_id (PK), reason, deleted_at

<b>Announcement</b>	System-wide broadcasts from moderators.	announcement_id (PK), display_duration_days, expires_at
---------------------	---	---

### 3.3.5 Data Structure 5 : Authentication & Session Logs

This structure handles security logs and session validity.

Table Name	Description	Key Attributes
<b>LoginSession</b>	Logs active and past user login times.	session_id (PK), login_time, is_active
<b>LogoutSession</b>	Logs when users exit the system.	logout_id (PK), logout_time
<b>PasswordReset</b>	Manages OTPs for account recovery.	reset_id (PK), otp_code, expires_at, is_used

### 3.3.6 Data Structure 6 : Gamification

This structure tracks the point economy within the system.

Table Name	Description	Key Attributes
<b>ScoreTransaction</b>	Ledger of every point addition or deduction (e.g., +5 for posting).	transaction_id (PK), action_type, points_change, resulting_score

## 4 Behavioral Modeling

### 4.1 Sequence Diagrams

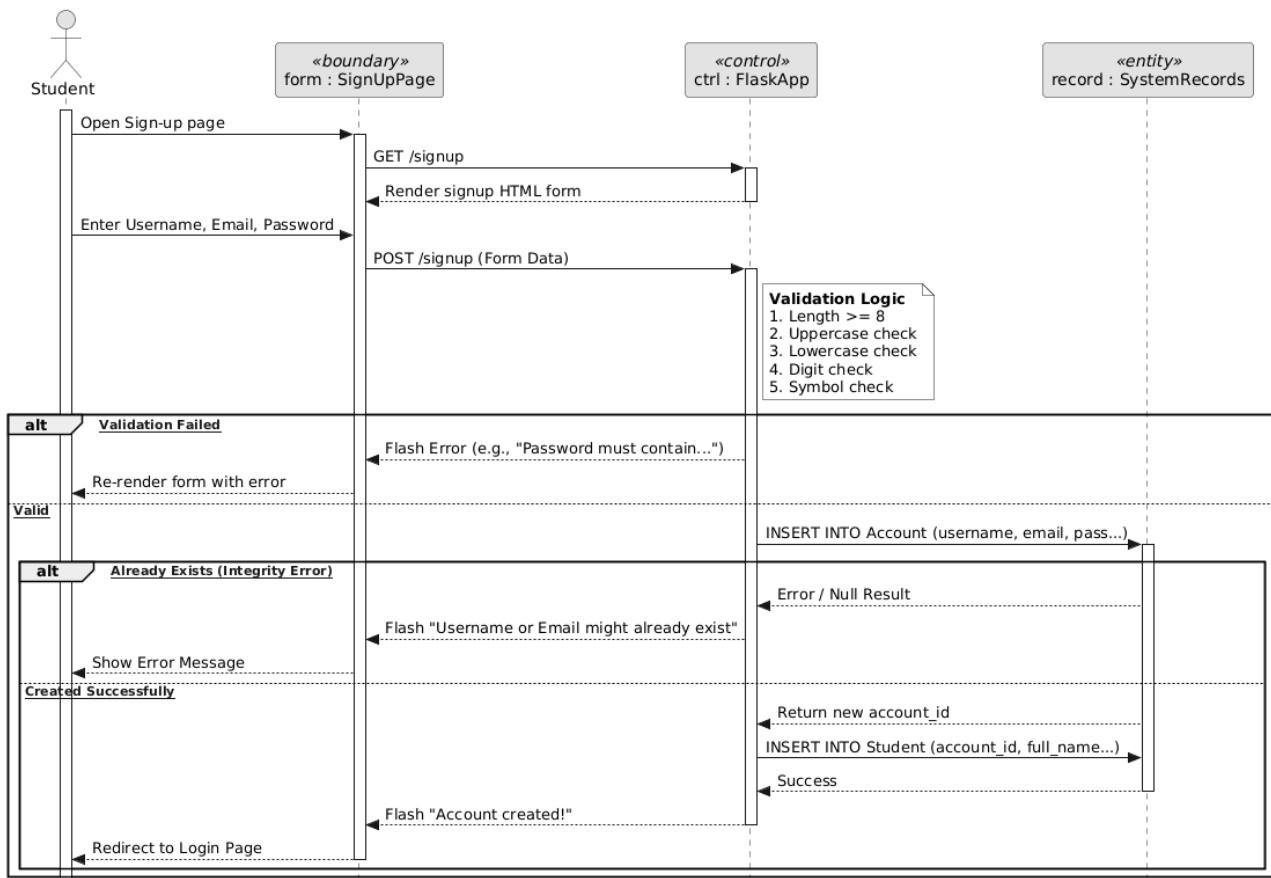
#### 4.1.1 Sign up a new account

The student is able to sign up for a new account using MMU official email student accounts and providing the required information, including username and password that consist at least eight characters including uppercase, lowercase, number and symbol.

<b>Use Case ID</b>	UC9
<b>Use Case Name</b>	Sign up a new account
<b>Primary Actor</b>	Student
<b>*Pre-Conditions</b>	Student is not registered.
<b>Post-Conditions</b>	New account is created and stored.
<b>Basic Flow</b>	1. Student opens sign-up page.

	<ol style="list-style-type: none"> <li>2. Student enters required details, including student ID, username, MMU official email, and password.</li> <li>3. System validates inputs.</li> <li>4. System creates the account.</li> <li>5. System displays confirmation.</li> </ol>
<b>Alternate Flow</b>	<ol style="list-style-type: none"> <li>3a. Student ID or email already exists</li> <li>3b. Password does not fulfill requirements</li> </ol>
<b>Exception</b>	None

*Table 4.1.1 Table for Use Case Specification 1*



*Diagram 4.1.1 Sequence Diagram Sign Up New Account*

#### **4.1.2 Log in with own account**

The actors (both student, moderator, counselor and admin) are able to log in to the system using their own registered email and password and prompted to try again if email or password is not correct.

<b>Use Case ID</b>	UC10
<b>Use Case Name</b>	Log in with own account

<b>Primary Actor</b>	User
<b>*Pre-Conditions</b>	User has a registered account
<b>Post-Conditions</b>	User is authenticated and session begins
<b>Basic Flow</b>	<ol style="list-style-type: none"> <li>1. The Use Case begins when the User navigates to the application homepage.</li> <li>2. The System displays the welcome card containing the login form with fields for username and password.</li> <li>3. The User enters their credentials into the respective fields and clicks the "Sign In" button.</li> <li>4. The System verifies the credentials by querying the Account table in the database for a matching username and password.</li> <li>5. The System confirms the account status is active and not suspended.</li> <li>6. The System establishes a session for the User, storing their unique account ID and role.</li> <li>7. The System records the login timestamp in the LoginSession database table.</li> <li>8. The System redirects the User to the dashboard route.</li> <li>9. The System detects the User's specific role and renders the appropriate dashboard view (Student, Admin, Moderator, or Counselor).</li> </ol>
<b>Alternate Flow</b>	2a. Wrong password 2b. Invalid email account
<b>Exception</b>	Server is not responding

*Table 4.1.2 Table for Use Case Specification 2*

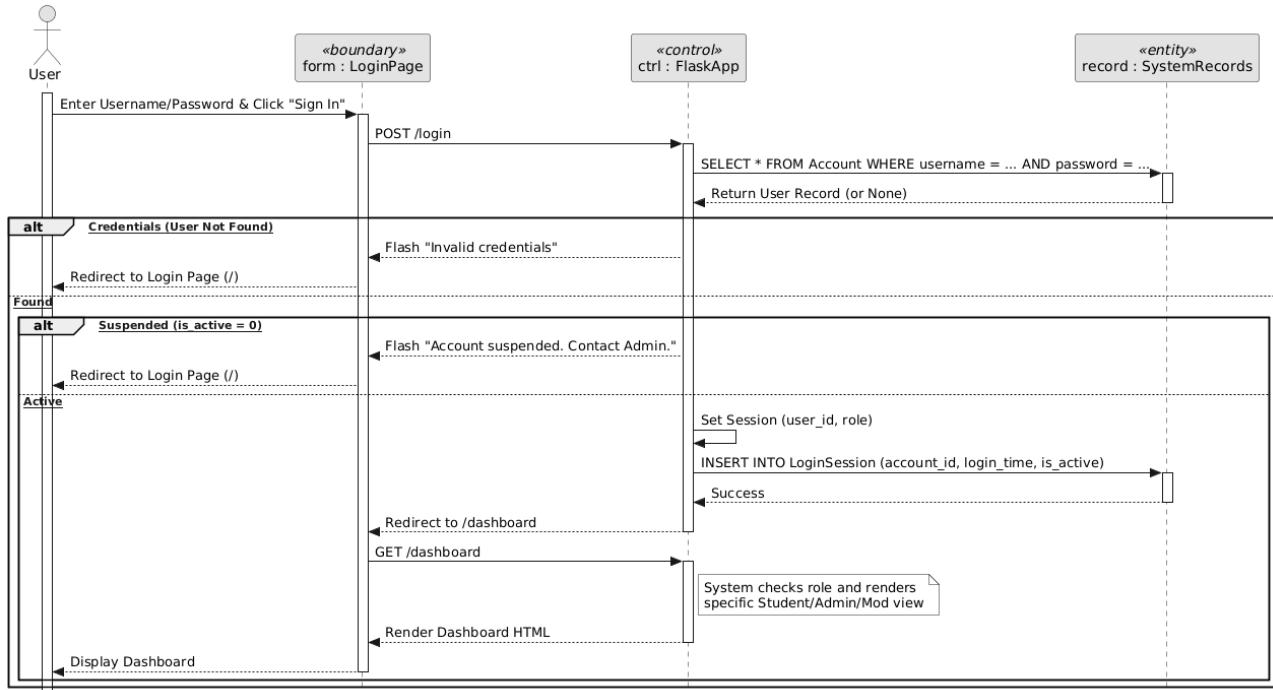


Diagram 4.1.2 Sequence Diagram Log in

#### 4.1.3 Log out with own account

The actors (both student, moderator, counselor and admin) are able to log out their own account through the system.

<b>Use Case ID</b>	UC11
<b>Use Case Name</b>	Log out with own account
<b>Primary Actor</b>	User
<b>*Pre-Conditions</b>	Active logged-in session
<b>Post-Conditions</b>	Session ends
<b>Basic Flow</b>	<ol style="list-style-type: none"> <li>1. The Use Case begins when the User clicks the "Log Out" link in the navigation menu.</li> <li>2. The System retrieves the User's unique identifier from the current session. The System records the logout event by inserting the account ID and the current timestamp into the LogoutSession database table.</li> <li>3. The System clears all stored data from the User's active browser session.</li> <li>4. The System redirects the User back to the application landing page.</li> </ol>
<b>Alternate Flow</b>	None
<b>Exception</b>	None

Table 4.1.3 Table for Use Case Specification 3

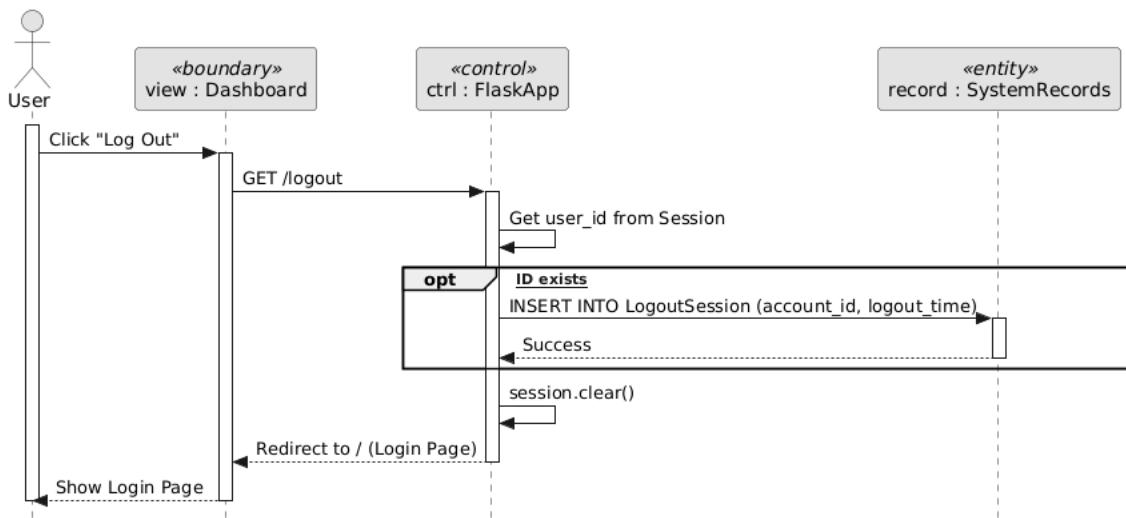


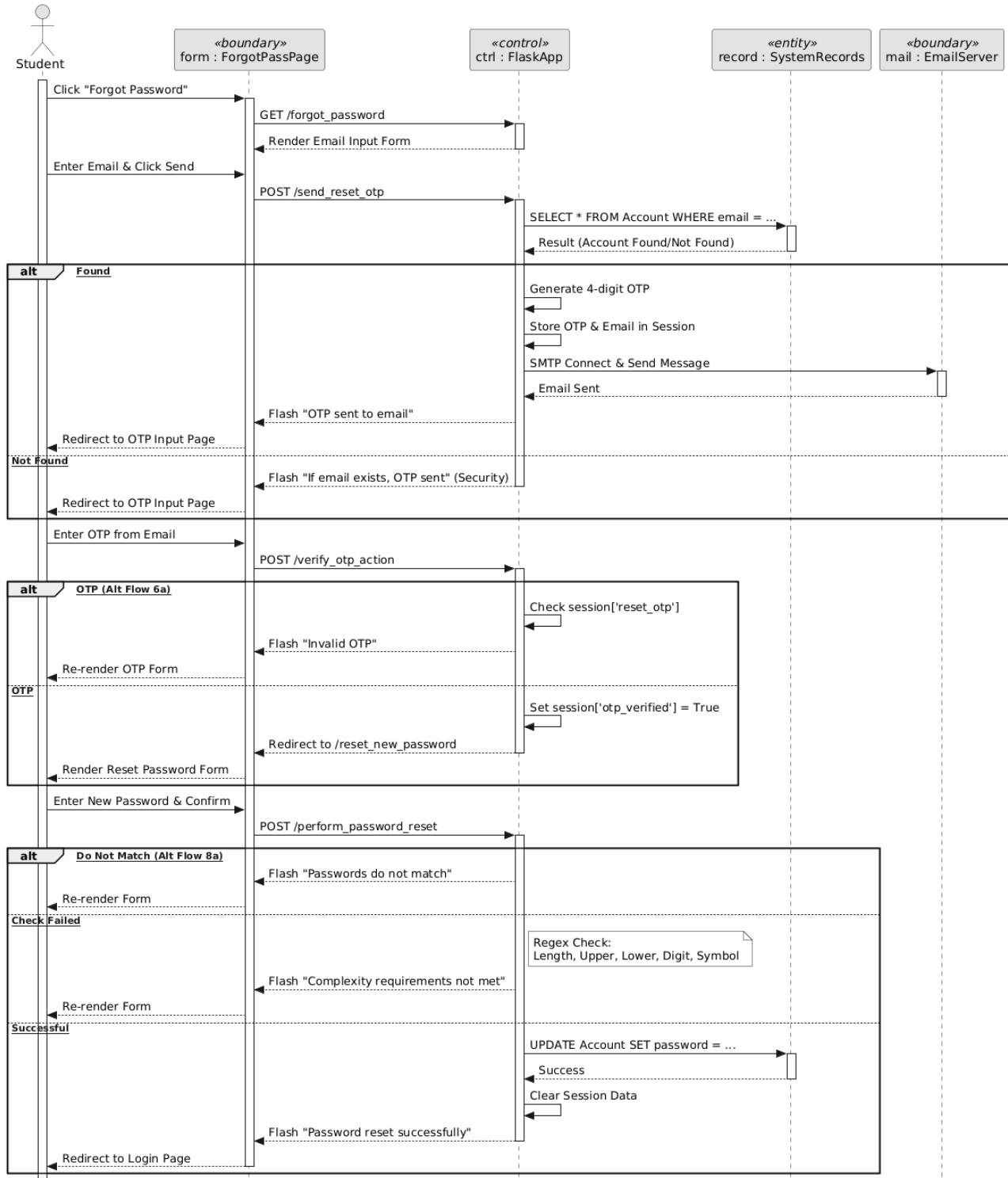
Diagram 4.1.3 Sequence Diagram Log out

#### 4.1.4 Reset Password

The student is able to reset password by getting one-time password (OTP) through their registered email to get access with their own account again by clicking “forgot password” in the sign in page.

<b>Use Case ID</b>	UC12
<b>Use Case Name</b>	Reset password
<b>Primary Actor</b>	Student
<b>*Pre-Conditions</b>	Student has a valid email account
<b>Post-Conditions</b>	Password is successfully updated
<b>Basic Flow</b>	<ol style="list-style-type: none"> <li>1. Student clicks on “Forgot Password”</li> <li>2. Student enters email address</li> <li>3. Student clicks on “Send”</li> <li>4. System sends OTP to the email address</li> <li>5. Student enters OTP sent through email</li> <li>6. System validates OTP</li> <li>7. Student sets new password</li> <li>8. Student confirms new password</li> </ol>
<b>Alternate Flow</b>	6a. Incorrect OTP 8a. Different input with the set new password field
<b>Exception</b>	None

Table 4.1.4 Table for Use Case Specification 4



*Diagram 4.1.4 Sequence Diagram Reset Password*

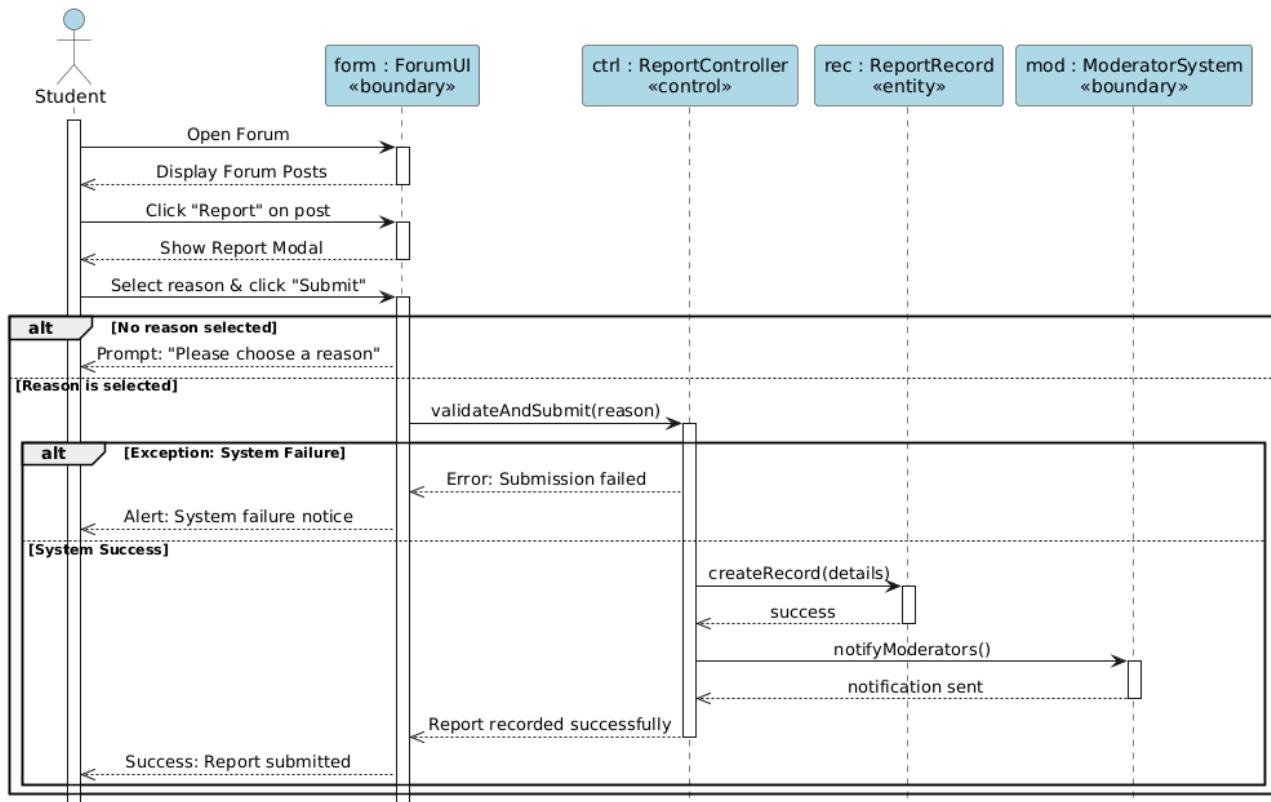
#### 4.1.5 Report contents

## *Software Design Specification for Digital Peer Support System (Version 2.0)*

The student is able to report harmful or inappropriate contents including posts and comments in the forum to the moderator through the report button implemented.

<b>Use Case ID</b>	UC13
<b>Use Case Name</b>	Report Contents
<b>Primary Actor</b>	Student
<b>*Pre-Conditions</b>	Student is logged in
<b>Post-Conditions</b>	Report is submitted to moderators
<b>Basic Flow</b>	<ol style="list-style-type: none"> <li>1. Student opens Forum</li> <li>2. Student clicks on “Report” on a post/comment</li> <li>3. Student chooses report reason</li> <li>4. System records report</li> <li>5. System notifies moderators</li> </ol>
<b>Alternate Flow</b>	3a. No reason selected during submission
<b>Exception</b>	System fails to submit report

*Table 4.1.5 Table for Use Case Specification 5*



*Diagram 4.1.5 Sequence Diagram Report Contents*

#### 4.1.6 Build profile

The student is able to build and update their personal profile with details such as interest labels, programme taken, and a short description.

<b>Use Case ID</b>	UC14
<b>Use Case Name</b>	Build profile
<b>Primary Actor</b>	Student
<b>*Pre-Conditions</b>	Student is logged in
<b>Post-Conditions</b>	Profile information is stored
<b>Basic Flow</b>	<ol style="list-style-type: none"> <li>1. Student opens profile page</li> <li>2. Student enters personal information</li> <li>3. Student adds on their interests</li> <li>4. System saves changes</li> </ol>
<b>Alternate Flow</b>	3a. Interest label longer than 10 characters
<b>Exception</b>	None

Table 4.1.6 Table for Use Case Specification 6

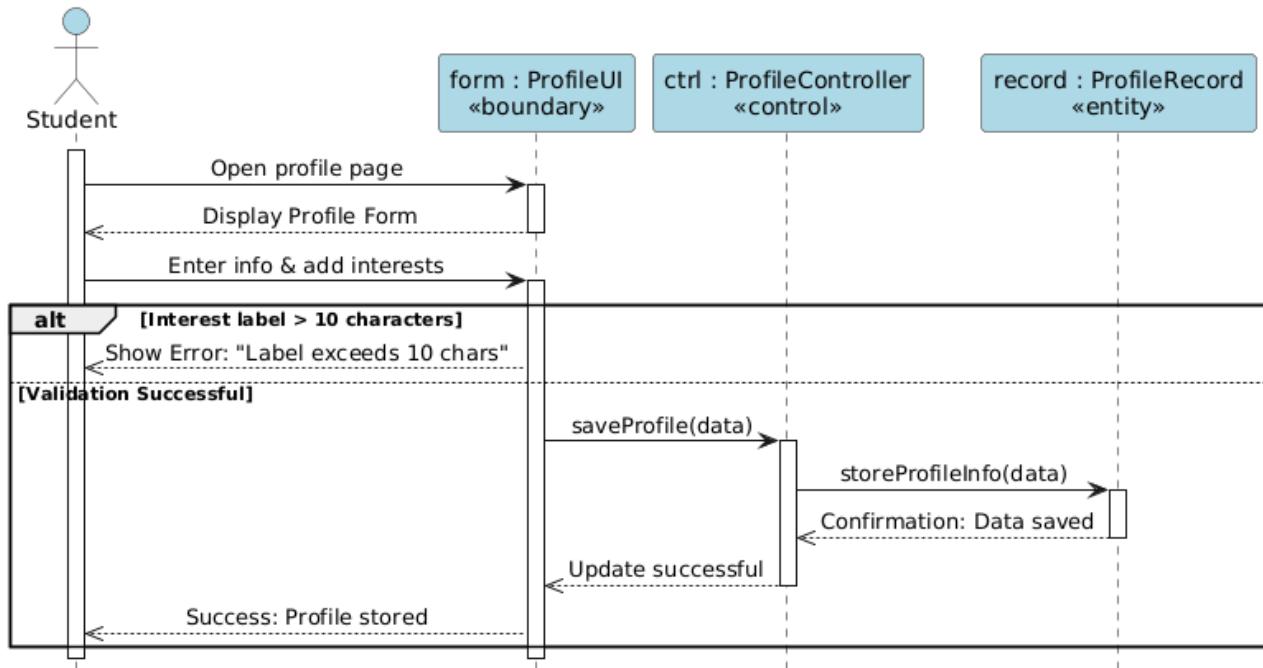


Diagram 4.1.6 Sequence Diagram Build Profile

#### 4.1.7 Check-in mood and well-being

The student is able to check in their daily mood and well-being with available options implemented in the system, and the respond will convert to score, and the score will be recorded in the database.

<b>Use Case ID</b>	UC15
<b>Use Case Name</b>	Check-in mood and well-being
<b>Primary Actor</b>	Student
<b>*Pre-Conditions</b>	Student is logged in
<b>Post-Conditions</b>	Check-in entry stored
<b>Basic Flow</b>	<ol style="list-style-type: none"> <li>1. Student chooses emotion level</li> <li>2. Student clicks on "Confirm"</li> <li>3. System saves entry</li> </ol>
<b>Alternate Flow</b>	None
<b>Exception</b>	None

Table 4.1.7 Table for Use Case Specification 7

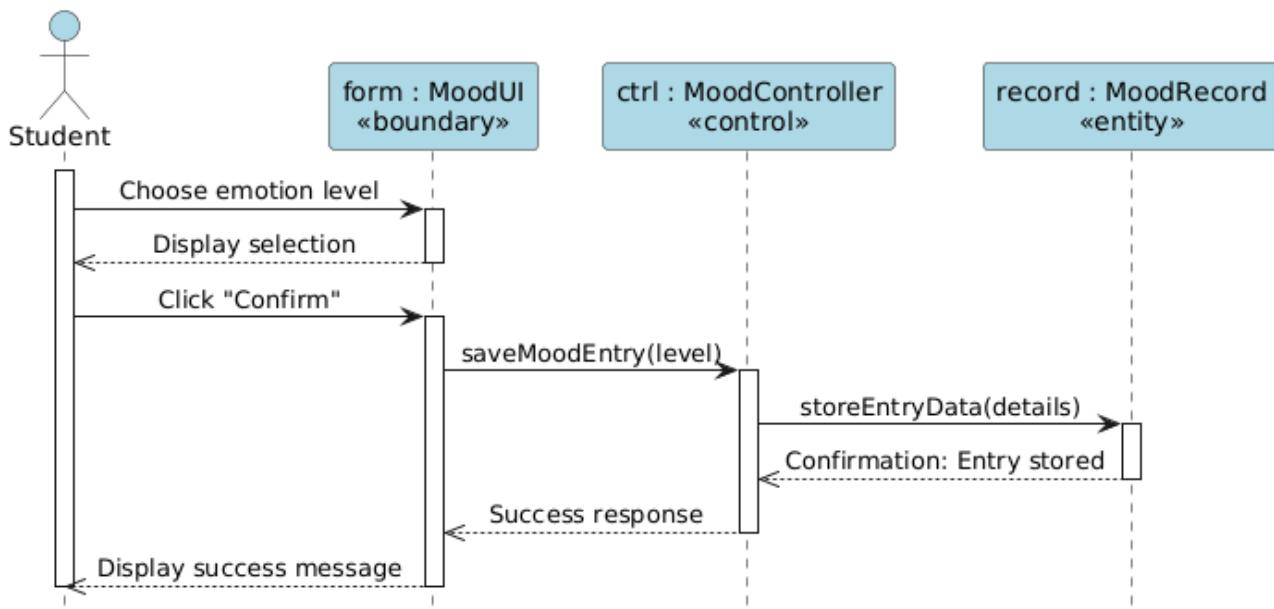


Diagram 4.1.7 Sequence Diagram Check-in mood and Well-being

#### 4.1.8 Add friends

The student is able to maintain social connections through adding new friends by sending and accepting friend requests.

<b>Use Case ID</b>	UC16
<b>Use Case Name</b>	Add friends
<b>Primary Actor</b>	Student
<b>*Pre-Conditions</b>	Student is logged in
<b>Post-Conditions</b>	Friend request is sent or accepted
<b>Basic Flow</b>	<ol style="list-style-type: none"> <li>1. Student clicks on "Add Friend" button</li> <li>2. System sends request</li> </ol>

	3. Other student accepts or declines 4. System notifies student on status of adding friend
<b>Alternate Flow</b>	4a. Friend request accepted 4b. Friend request declined
<b>Exception</b>	None

Table 4.1.8 Table for Use Case Specification 8

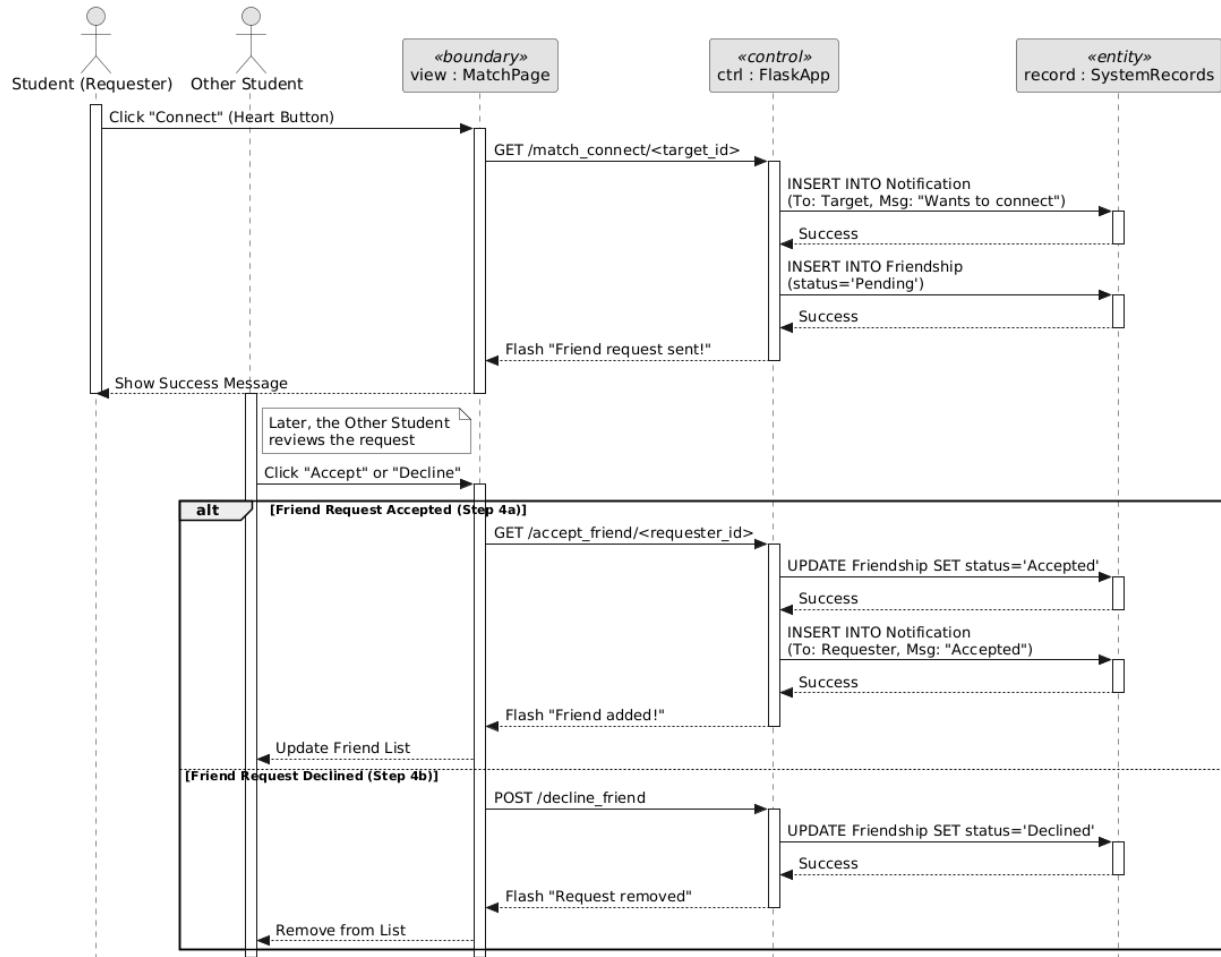


Diagram 4.1.8 Sequence Diagram Add Friends

#### 4.1.9 Chat privately

The student is able to maintain social connections through engaging in private chat conversations with anyone including or excluding their friends through others' profile, and chat history is recorded.

<b>Use Case ID</b>	UC17
<b>Use Case Name</b>	Chat privately
<b>Primary Actor</b>	Student
<b>*Pre-Conditions</b>	Student is logged in and has a friend
<b>Post-Conditions</b>	Message sent/received

<b>Basic Flow</b>	<ol style="list-style-type: none"> <li>1. Student selects a matched user.</li> <li>2. System opens the private chat interface.</li> <li>3. Student types and sends a message.</li> <li>4. System stores and delivers the message.</li> <li>5. Recipient views the message.</li> </ol>
<b>Alternate Flow</b>	
<b>Exception</b>	Delivery failure

Table 4.1.9 Table for Use Case Specification 9

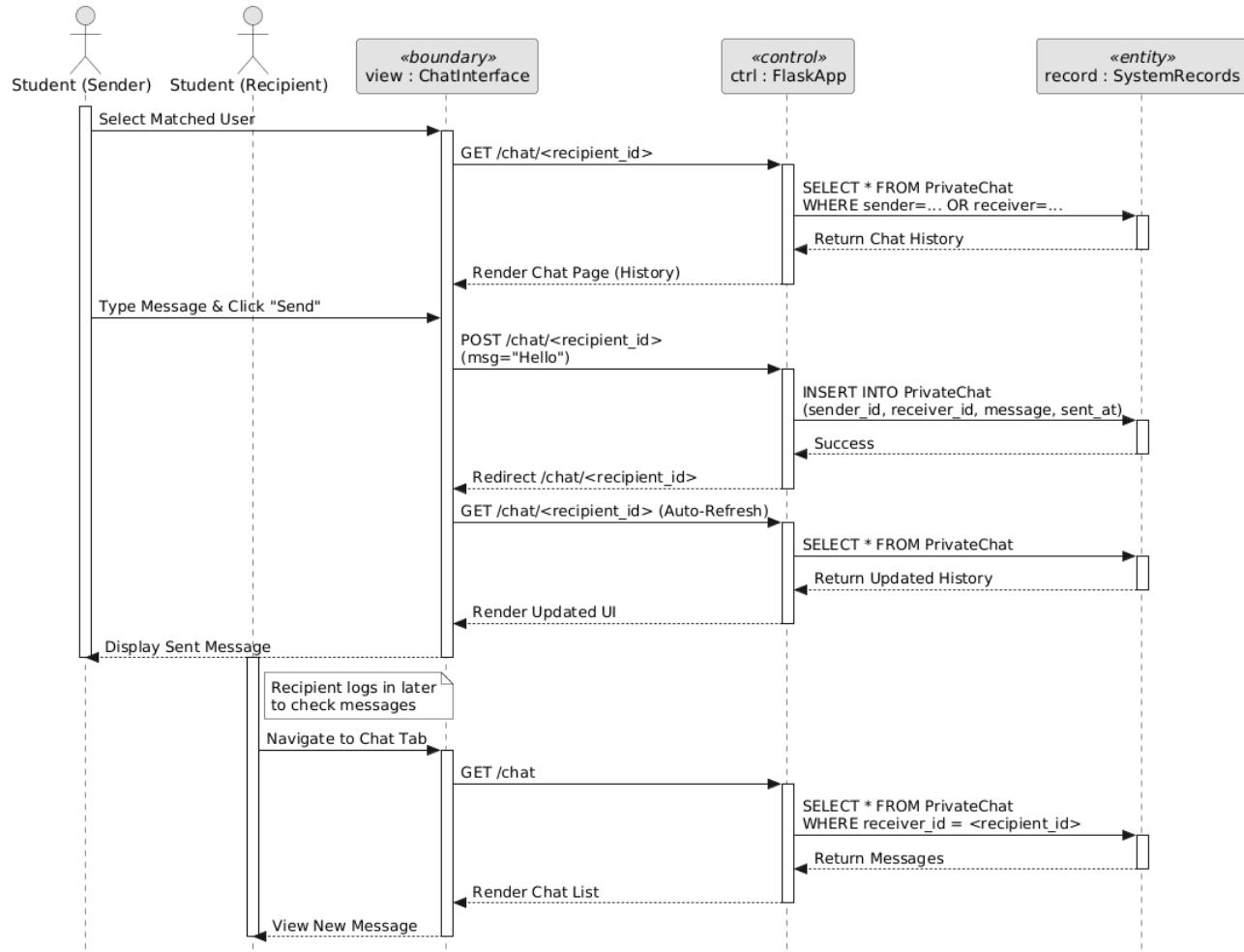


Diagram 4.1.9 Sequence Diagram Chat Privately

#### 4.1.10 Make posts

The student is able to create and publish posts in the forum anonymously or non-anonymously (showing the username).

<b>Use Case ID</b>	UC18
--------------------	------

<b>Use Case Name</b>	Make posts
<b>Primary Actor</b>	Student
<b>*Pre-Conditions</b>	Student is logged in
<b>Post-Conditions</b>	New forum post is created
<b>Basic Flow</b>	<ol style="list-style-type: none"> <li>1. Student opens Forum</li> <li>2. Student clicks on the tick box to show or hide username for post</li> <li>3. Student enters text content</li> <li>4. Student optionally attaches photos/videos/files</li> <li>5. Student clicks on “Tweet” button</li> <li>6. System validates post content</li> <li>7. System publishes post</li> <li>8. System displays the new post in the Forum</li> </ol>
<b>Alternate Flow</b>	<p>6a. Invalid attachment type      6b. More than 5 files</p>
<b>Exception</b>	Attachment upload fails due to network issues; posting error

Table 4.1.10 Table for Use Case Specification 10

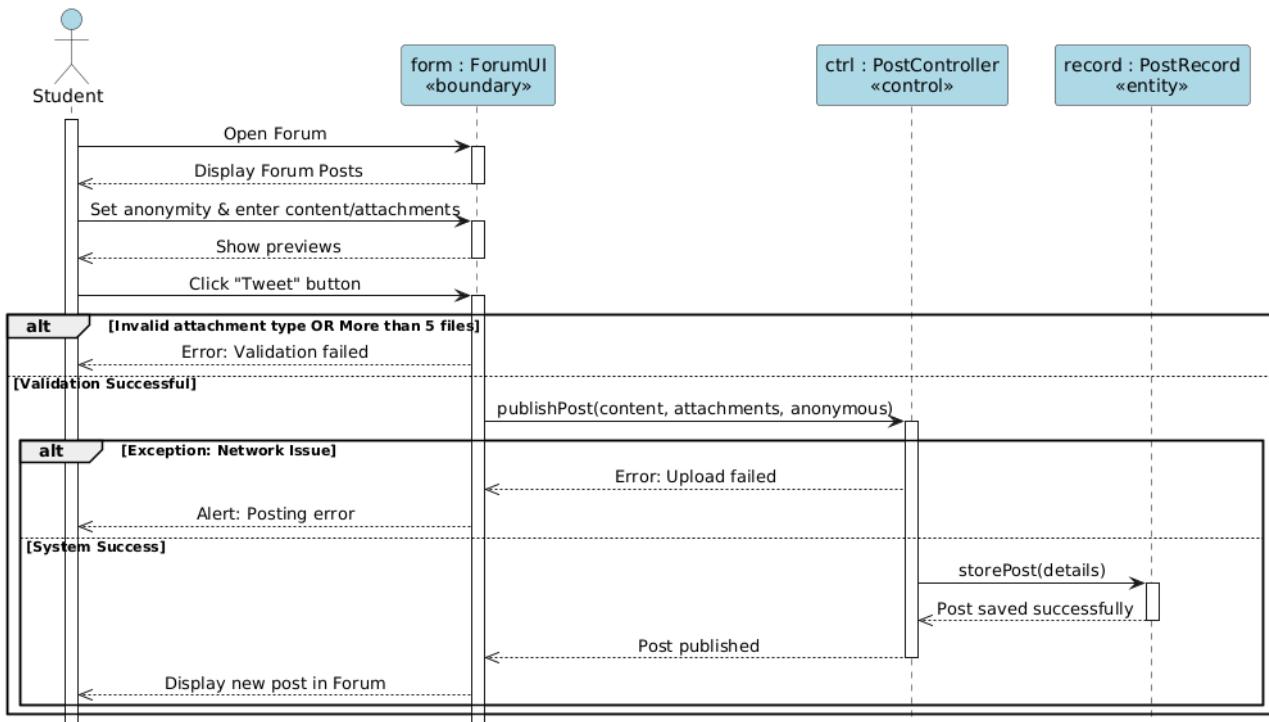


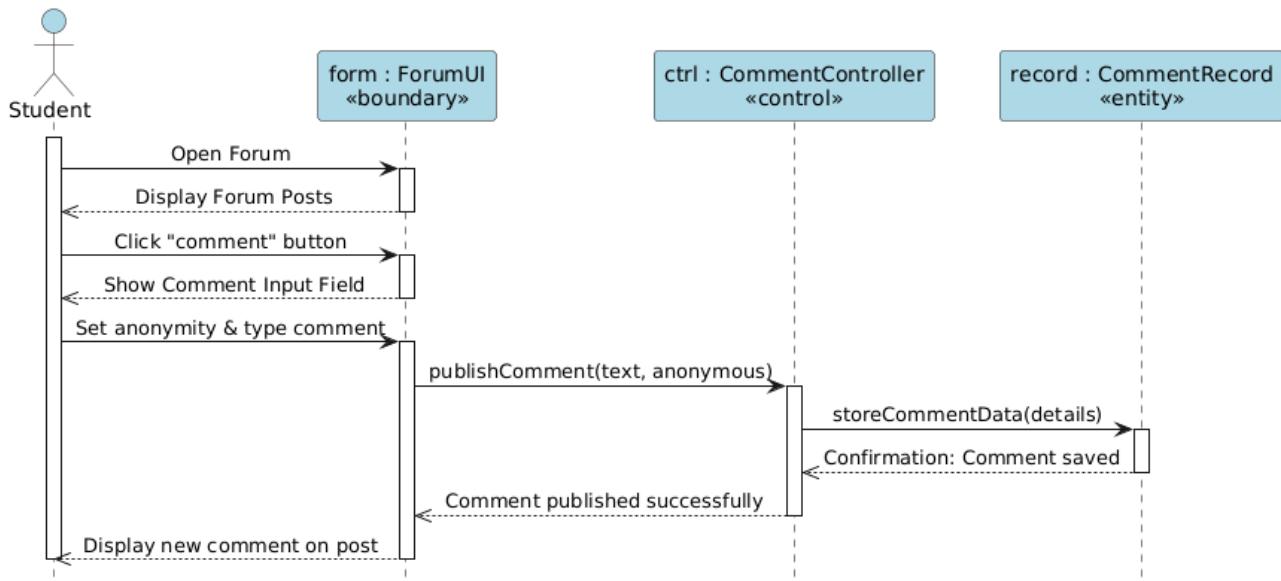
Diagram 4.1.10 Sequence Diagram Make Posts

#### 4.1.11 Comment on posts

The student is able to expand social connections through commenting anonymously or non-anonymously (showing the username) on the forum posts created by others.

<b>Use Case ID</b>	UC19
<b>Use Case Name</b>	Comment on posts
<b>Primary Actor</b>	Student
<b>*Pre-Conditions</b>	Student is logged in
<b>Post-Conditions</b>	Comment is added
<b>Basic Flow</b>	<ol style="list-style-type: none"> <li>1. Student opens Forum</li> <li>2. Student clicks on “comment” button of a post</li> <li>3. Student clicks on the tick box to show or hide username for comment</li> <li>4. Student types comment</li> <li>5. System publishes the comment</li> </ol>
<b>Alternate Flow</b>	None
<b>Exception</b>	None

*Table 4.1.11 Table for Use Case Specification 11*



*Diagram 4.1.11 Sequence Diagram Comment on Posts*

#### **4.1.12 Like posts and comments**

The student is able to expand social connections through like or unlike the posts and comments in the forum.

<b>Use Case ID</b>	UC20
<b>Use Case Name</b>	Like posts and comments

<b>Primary Actor</b>	Student
<b>*Pre-Conditions</b>	Student is logged in
<b>Post-Conditions</b>	Like count updated
<b>Basic Flow</b>	<ol style="list-style-type: none"> <li>1. Student opens Forum</li> <li>2. Student clicks on “Like” button for a post or a comment</li> <li>3. System updates counter</li> <li>4. System shows updated state</li> </ol>
<b>Alternate Flow</b>	2a. System toggles unlike if student already liked
<b>Exception</b>	None

Table 4.1.12 Table for Use Case Specification 12

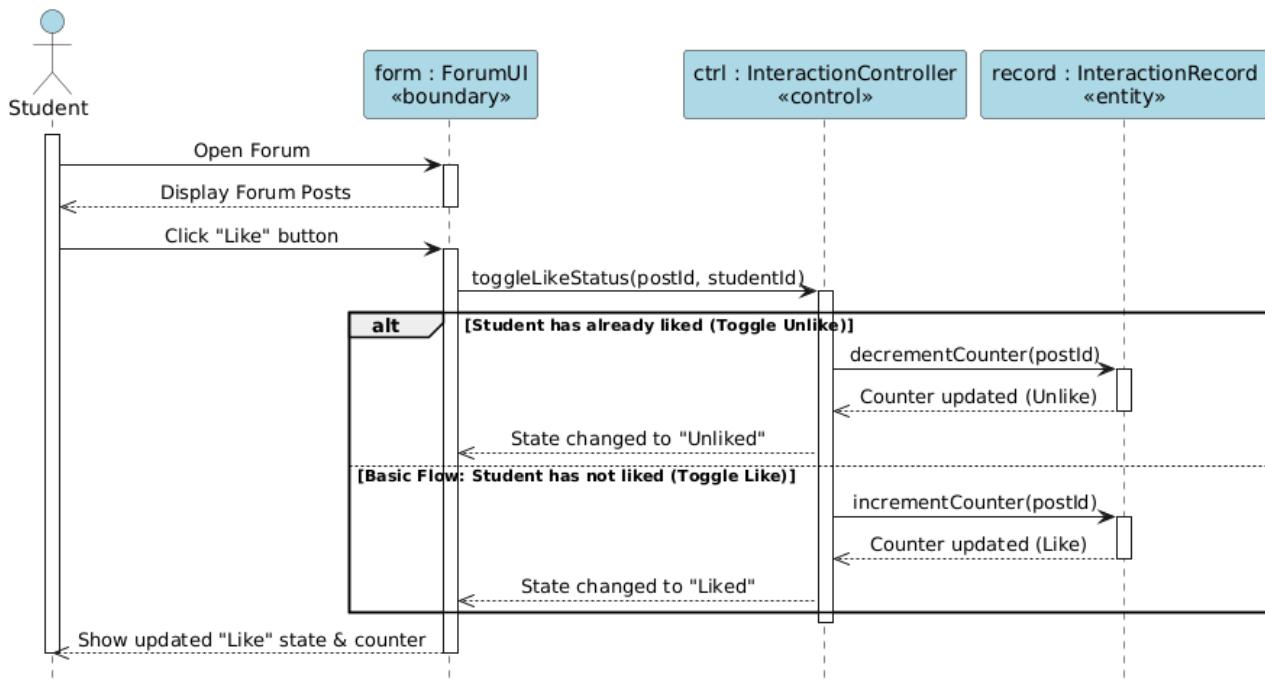


Diagram 4.1.12 Sequence Diagram Like Posts and Comments

#### 4.1.13 Match-up with people

The student is able to expand social connections through matching with other students who having similar interest tags in their profile.

<b>Use Case ID</b>	UC22
<b>Use Case Name</b>	Match-up with people
<b>Primary Actor</b>	Student
<b>*Pre-Conditions</b>	Student is logged in; Student has interest label in their profile
<b>Post-Conditions</b>	Suggested matches displayed

<b>Basic Flow</b>	1. Student opens the match-up feature. 2. System displays available matches. 3. Student selects or skips users. 4. System records the match decision.
<b>Alternate Flow</b>	3a. No match found 5a. Next Matches displayed if 'x' button selected 5b. Friend request sent if 'love' icon button selected
<b>Exception</b>	None

Table 4.1.13 Table for Use Case Specification 13

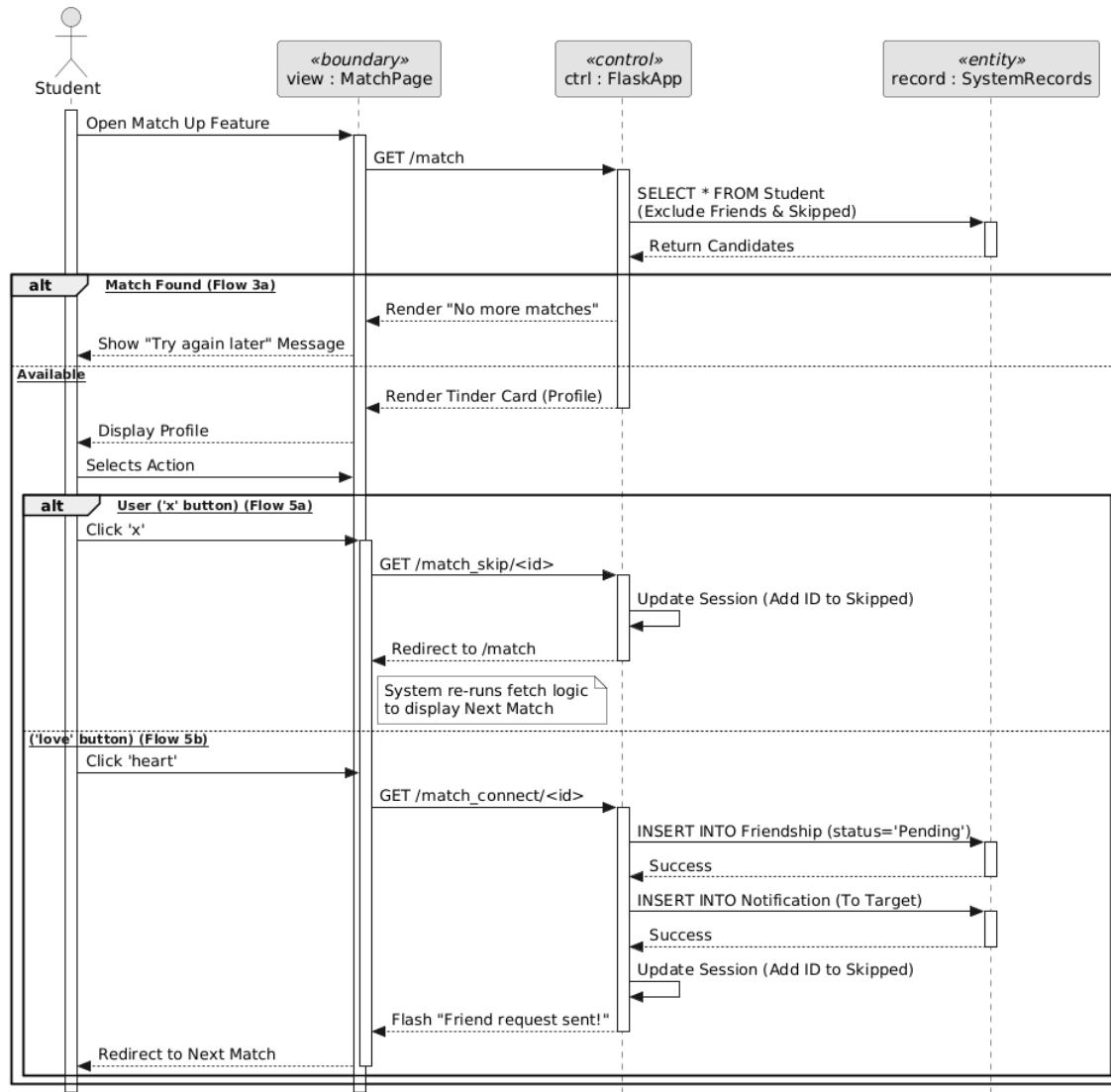


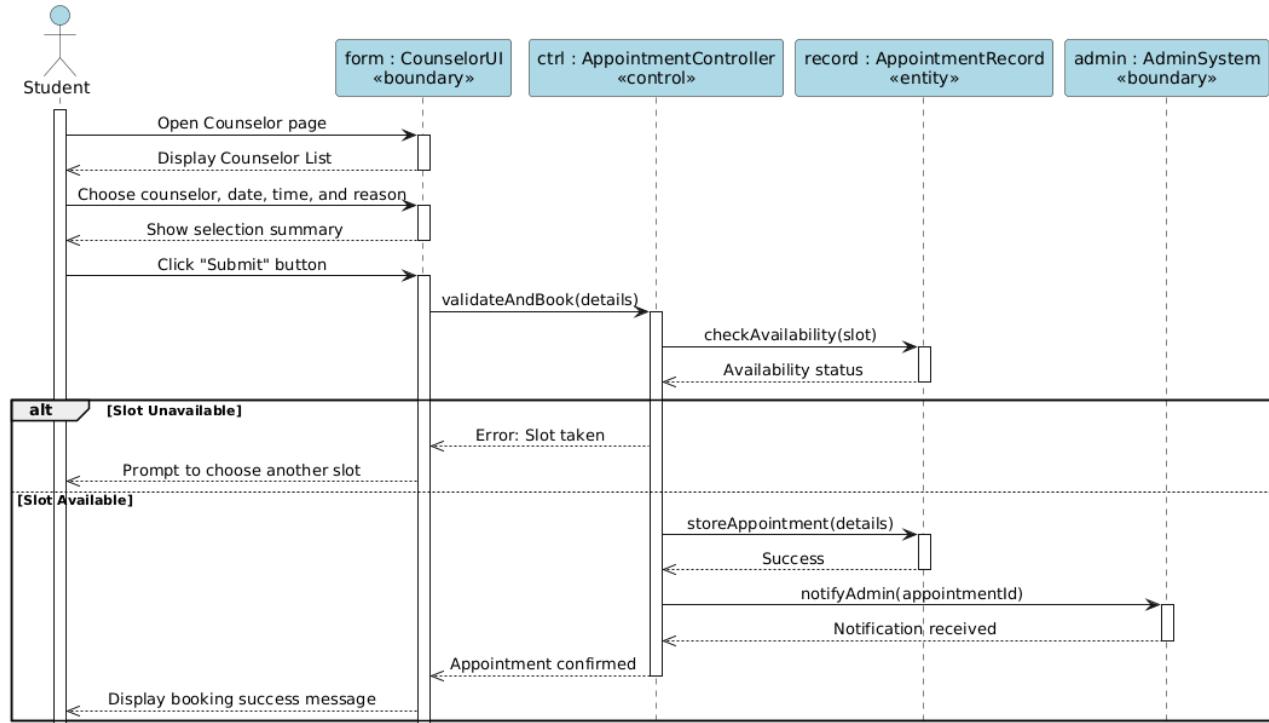
Diagram 4.1.13 Sequence Diagram Match Up with People

#### 4.1.14 Book counselor appointments

Student is able to book an appointment with a counselor through the system.

<b>Use Case ID</b>	UC23
<b>Use Case Name</b>	Book counselor appointments
<b>Primary Actor</b>	Student
<b>*Pre-Conditions</b>	Student is logged in
<b>Post-Conditions</b>	Appointment request stored and sent to admin
<b>Basic Flow</b>	<ol style="list-style-type: none"> <li>1. Student opens Counselor page</li> <li>2. Student chooses preferred counselor</li> <li>3. Student selects date and time preferred</li> <li>4. Student fills in the reason field</li> <li>5. Student clicks on submit button</li> <li>6. System checks availability</li> <li>7. System sends notification to admin</li> </ol>
<b>Alternate Flow</b>	6a. Prompt student to choose another slot if the slot chosen is unavailable
<b>Exception</b>	None

*Table 4.1.14 Table for Use Case Specification 14*



*Diagram 4.1.14 Sequence Diagram Book Counselor Appointments*

#### 4.1.15 Earn points

***Software Design Specification for Digital Peer Support System (Version 2.0)***

---

The student is able to earn points to sustain the premium feature of the system through check-in daily mood, or getting likes in a single post or comment published anonymously or non-anonymously in the forum.

<b>Use Case ID</b>	UC21
<b>Use Case Name</b>	Earn points
<b>Primary Actor</b>	Student
<b>*Pre-Conditions</b>	Student is logged in
<b>Post-Conditions</b>	Points updated
<b>Basic Flow</b>	<ol style="list-style-type: none"><li>1. Student performs an action eligible for earning points</li><li>2. System calculates awarded points</li><li>3. System checks and verify the daily awarded points</li><li>4. System updates student's total points</li><li>5. System displays new updated points in the profile</li></ol>
<b>Alternate Flow</b>	<ol style="list-style-type: none"><li>2a. System denies additional points if mood check-in more than once per day</li><li>2b. System denies additional points if a post/comment that already received points</li><li>3a. System denies when the points added per day greater than 10.</li><li>4a. System change the colour of the username if the points of that student account are equal or above 95</li></ol>
<b>Exception</b>	Database update failure

*Table 4.1.15 Table for Use Case Specification 15*

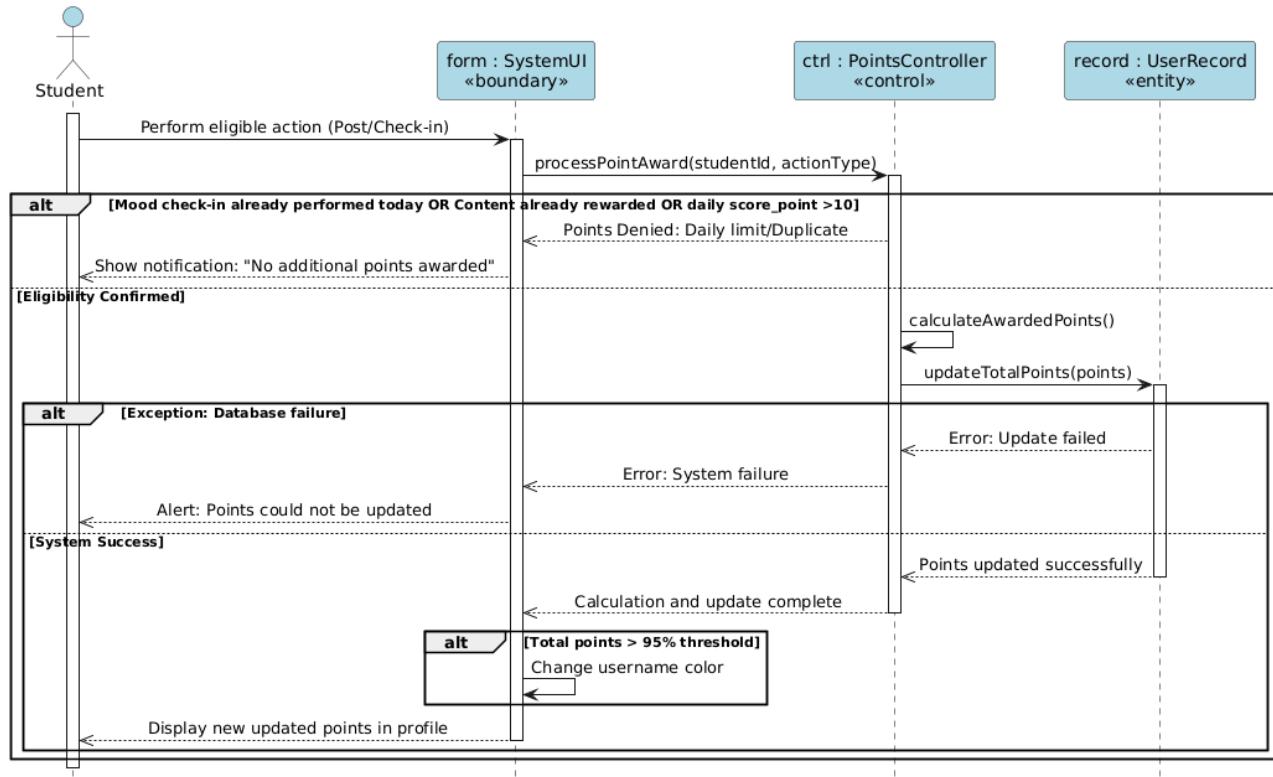


Diagram 4.1.15 Sequence Diagram Earn Points

#### 4.1.16 Monitor Mood Alert Dashboard

The admin is able to display all mood alerts in prioritized queue sorted by severity level (critical, high, medium, low) showing student identification, weekly mood percentage, trend patterns, and last active timestamp with automatic refresh every 5 minutes.

<b>Use Case ID</b>	UC3
<b>Use Case Name</b>	Monitor Mood Alert Dashboard
<b>Primary Actor</b>	Admin
<b>*Pre-Conditions</b>	<ul style="list-style-type: none"> <li>1. Admin is logged into the system</li> <li>2. Mood tracking system is active</li> </ul>
<b>Post-Conditions</b>	<ul style="list-style-type: none"> <li>1. Student and counselor receive notifications if assignment made</li> <li>2. Admin dashboard updates to reflect current alert status</li> <li>3. Alert is reviewed and appropriate action taken</li> </ul>
<b>Basic Flow</b>	<ul style="list-style-type: none"> <li>1. Admin navigates to Mood Alerts section from left sidebar</li> </ul>

	<ol style="list-style-type: none"> <li>2. System displays mood alert dashboard with all flagged students sorted by severity</li> <li>3. System shows key information for each alert: student name, ID, program, mood score percentage, severity level, duration below threshold, assignment status</li> <li>4. Admin reviews alert details including mood trend patterns and student activity</li> <li>5. Admin filters alerts by severity level (Critical, High, Medium)</li> <li>6. Admin identifies critical cases (below 30% mood score) requiring immediate action</li> <li>7. Admin clicks "Assign" button for unassigned critical alert</li> <li>8. System displays available counselors with specializations and current caseload</li> <li>9. Admin selects appropriate counselor and confirms assignment</li> <li>10. System sends automated notification to both student and counselor</li> <li>11. System updates alert status to "Assigned"</li> <li>12. Admin marks alert as "Reviewed" in the system</li> </ol>
<b>Alternate Flow</b>	<ol style="list-style-type: none"> <li>6a. if no critical alerts exist;             <ol style="list-style-type: none"> <li>1. admin reviews high and medium priority alerts instead</li> </ol> </li> <li>8a. if all appropriate counselors are at maximum caseload (20 students);             <ol style="list-style-type: none"> <li>1. System displays warning message about counselor capacity</li> </ol> </li> </ol>
<b>Exception</b>	None

*Table 4.1.16 Table for Use Case Specification 16*

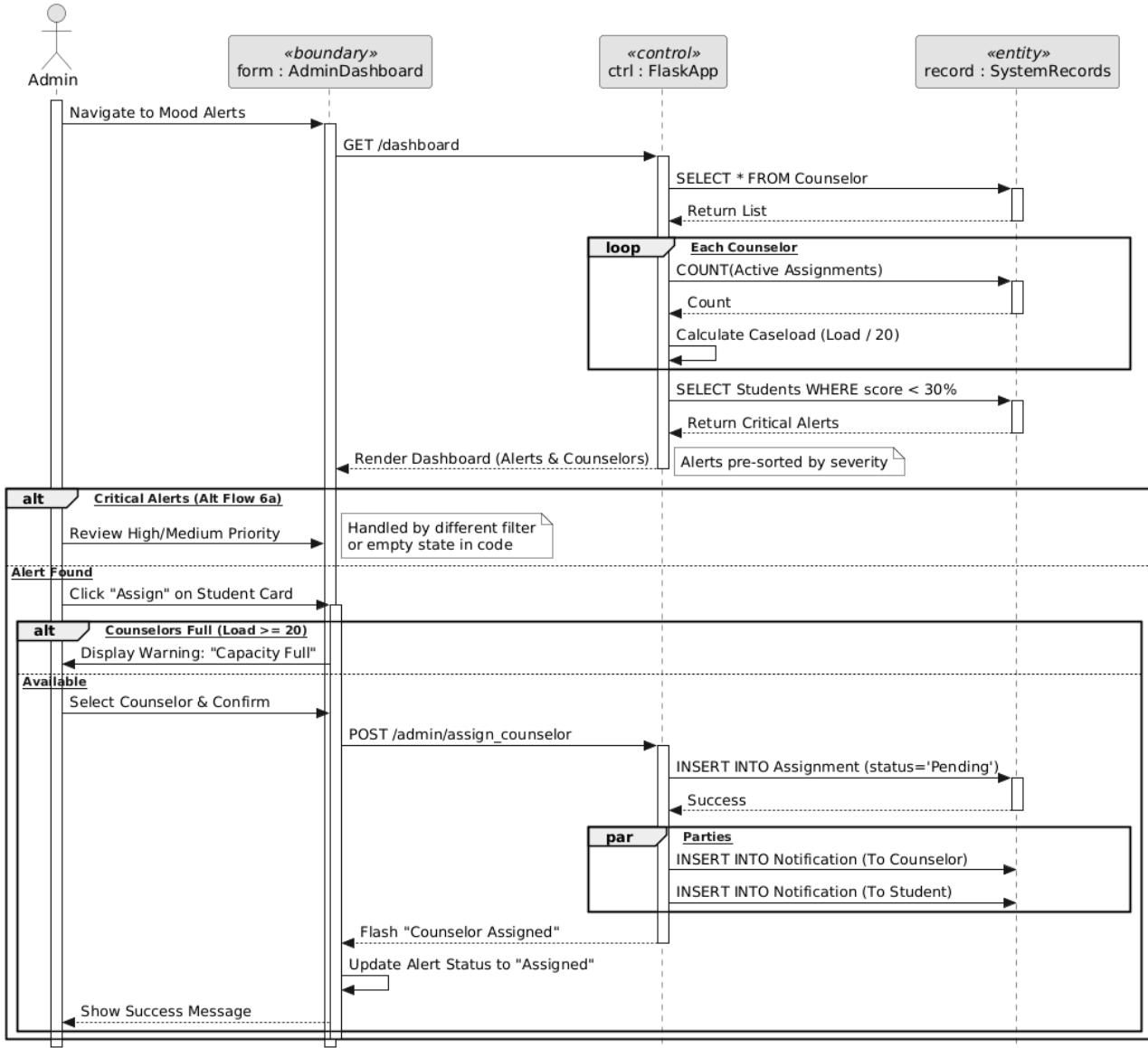


Diagram 4.1.16 Sequence Diagram Monitor Mood Alert Dashboard

#### **4.1.17 Manage Counselor Assignments**

The admin is able to provide student assignment interface displaying available counselors with current caseload count and specializations, send automated notifications to both counselor and student , and track assignment status (pending acceptance, active, on hold, completed).

<b>Use Case ID</b>	UC4
<b>Use Case Name</b>	Manage Counselor Assignments
<b>Primary Actor</b>	Admin
<b>*Pre-Conditions</b>	1. Admin is logged into the system

<b>Post-Conditions</b>	<ol style="list-style-type: none"> <li>1. Student assigned to counselor</li> <li>2. Status is "Pending Acceptance"</li> <li>3. Counselor's caseload increased by 1</li> </ol>
<b>Basic Flow</b>	<ol style="list-style-type: none"> <li>1. Admin navigates to Counselors section from left sidebar</li> <li>2. System displays counselor management dashboard with summary metrics (total counselors, active cases, average caseload, response rate)</li> <li>3. Admin views grid of available counselors showing availability status and current caseload</li> <li>4. Admin identifies student requiring counselor assignment (from mood alerts or appointment made by students)</li> <li>5. Admin reviews student's alert details: mood score and severity level</li> <li>6. Admin checks counselor availability and current caseload</li> <li>7. Admin selects appropriate counselor based on capacity and preferable</li> <li>8. Admin clicks "Assign Student" button on counselor card</li> <li>9. System displays assignment confirmation dialog with student and counselor details</li> <li>10. Admin confirms assignment</li> <li>11. System sends automated notification email to counselor within 5 minutes with student information and alert details</li> <li>12. System sends automated notification to student introducing their assigned counselor</li> <li>13. System updates assignment status to "Pending Acceptance"</li> </ol>
<b>Alternate Flow</b>	<ol style="list-style-type: none"> <li>4a. If student booked appointment directly;             <ol style="list-style-type: none"> <li>1. Admin receives notification: "Student [Name] requested counselor appointment"</li> <li>2. System displays student's appointment request with: - Preferred date/time - Reason for appointment - Student's current status (mood score if available)</li> <li>3. Admin reviews student's profile and needs</li> <li>4. Admin selects appropriate counselor</li> </ol> </li> </ol>
<b>Exception</b>	None

*Table 4.1.17 Table for Use Case Specification 17*

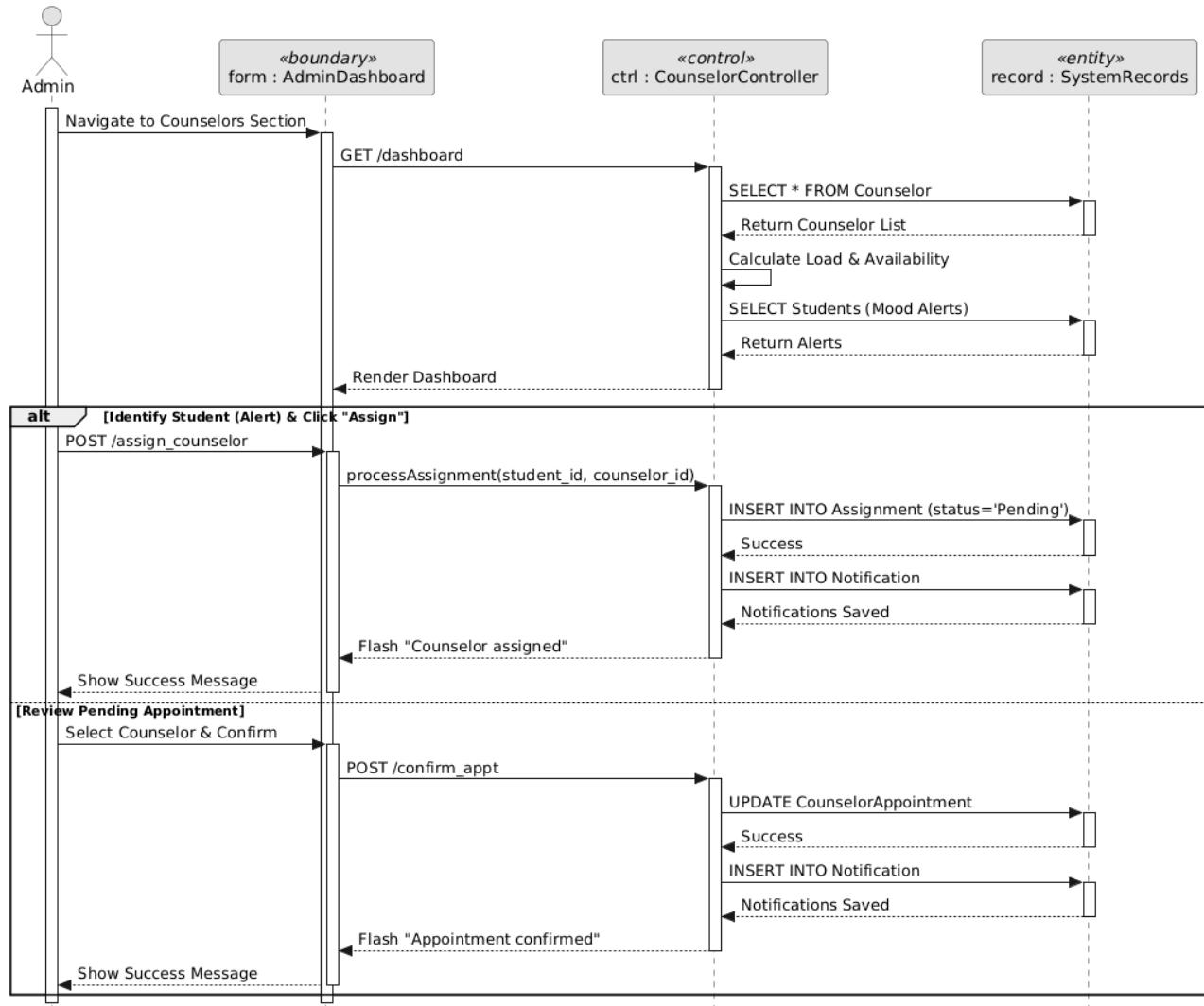


Diagram 4.1.17 Sequence Diagram Manage Counselor Assignment

#### 4.1.18 Review Reported Content

The admin is able to view all reported content with full context including thread conversation and user history, provide moderation actions (dismiss, warn, remove content, apply penalty, suspend), and automatically deduct points when content removal is selected (15 points for guideline violation, 30 points for harassment, 50 points for severe violations).

<b>Use Case ID</b>	UC5
<b>Use Case Name</b>	Review Reported Contents
<b>Primary Actor</b>	Admin
<b>*Pre-Conditions</b>	1. Admin is logged into the system

	<ol style="list-style-type: none"> <li>2. Scoring system is active and configured with penalty values</li> </ol>
<b>Post-Conditions</b>	<ol style="list-style-type: none"> <li>1. If violation confirmed, content removed and points deducted automatically</li> <li>2. Report marked "Resolved"</li> <li>3. If user score falls below 60%, posting privileges restricted</li> </ol>
<b>Basic Flow</b>	<ol style="list-style-type: none"> <li>1. Admin navigates to Reports &amp; Moderation section from left sidebar</li> <li>2. System displays report queue with all pending reports sorted by priority</li> <li>3. System shows report summary including: report ID, violation type, reporter info, reported user, content preview, time submitted, priority level</li> <li>4. Admin reviews priority indicators (High Priority reports highlighted with orange border)</li> <li>5. Admin selects report to review by clicking on report card</li> <li>6. System displays full report details</li> <li>7. Admin evaluates whether content violates community guidelines</li> <li>8. Admin determines violation severity: Minor (guideline violation), Moderate (harassment), Severe (threatening behavior)</li> <li>9. Admin selects appropriate moderation action</li> <li>10. Admin clicks selected action button</li> <li>11. System displays confirmation dialog showing penalty points to be deducted</li> <li>12. Admin confirms action</li> <li>13. System automatically deducts points from reported user's score (15, 30, or 50 points based on severity)</li> <li>14. System removes content from platform if "Remove" action selected</li> <li>15. System sends automated notification to reported user explaining violation, penalty applied, and updated score</li> <li>16. System checks if reported user's score falls below 60% threshold</li> <li>17. If score below 60%, system automatically restricts posting privileges</li> <li>18. System marks report as "Resolved"</li> </ol>

<b>Alternate Flow</b>	None
<b>Exception</b>	None

Table 4.1.18 Table for Use Case Specification 18

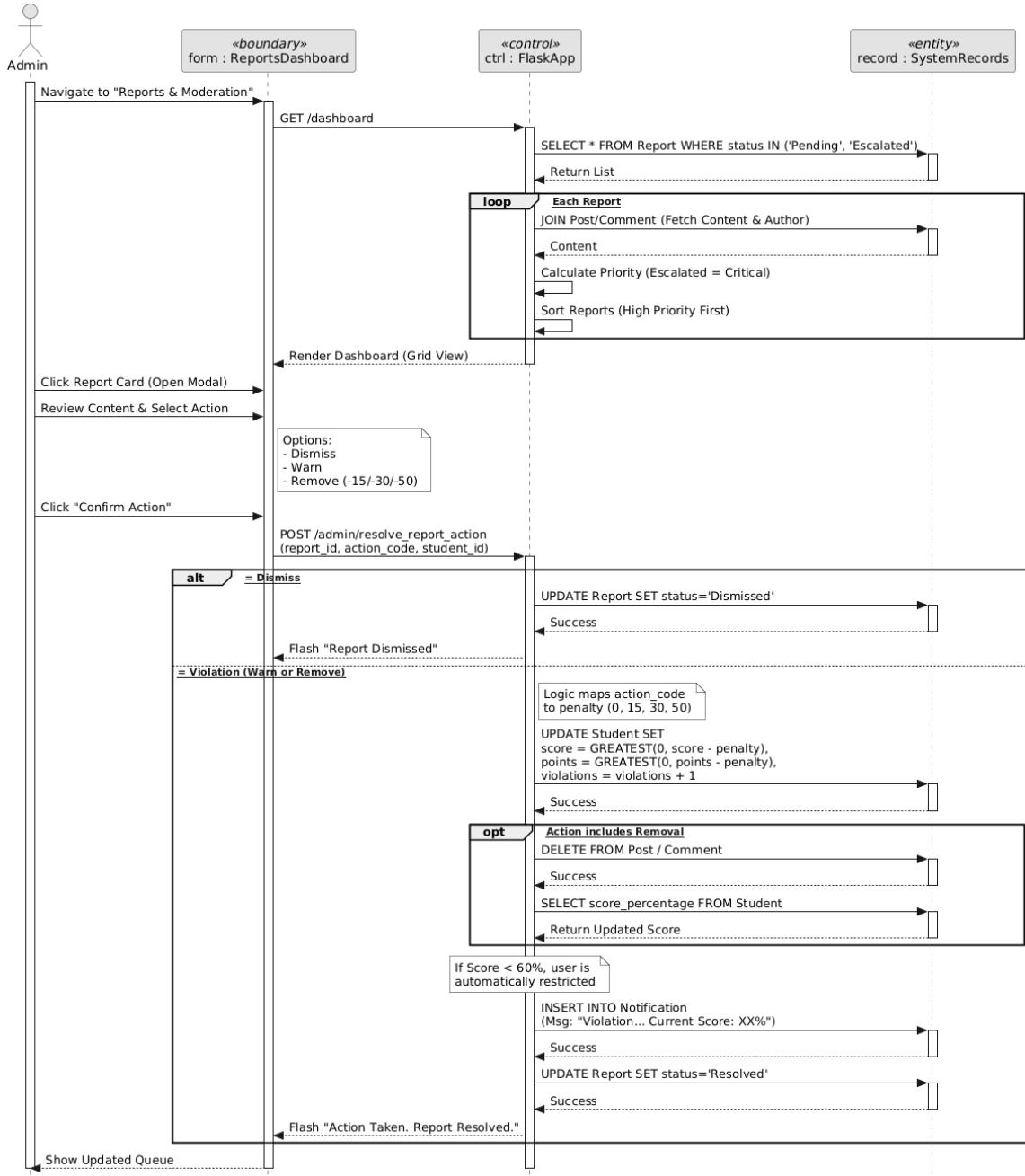


Diagram 4.1.18 Sequence Diagram Review Reported Content

#### 4.1.19 Manage Scoring System

The admin is able to configure point values for positive contributions or violations and manually adjusts scores when necessary to maintain a fair and effective behavioral accountability system.

<b>Use Case ID</b>	UC6
<b>Use Case Name</b>	Manage Scoring System
<b>Primary Actor</b>	Admin
<b>*Pre-Conditions</b>	<ul style="list-style-type: none"> <li>1. Admin is logged into the system</li> <li>2. Base scoring rules exist in system</li> </ul>
<b>Post-Conditions</b>	<ul style="list-style-type: none"> <li>1. New point values saved and active</li> </ul>
<b>Basic Flow</b>	<ul style="list-style-type: none"> <li>1. Admin navigates to Scoring System section from left sidebar</li> <li>2. System displays scoring system management dashboard with two main sections: Configuration and Restricted Users</li> <li>3. Admin reviews current scoring configuration</li> <li>4. Admin decides to adjust point values based on system effectiveness analysis</li> <li>5. Admin clicks on point value input field for specific action (e.g., "Helpful Answer")</li> <li>6. Admin enters new point value (e.g., changes from +10 to +12)</li> <li>7. Admin repeats for other actions requiring adjustment</li> <li>8. Admin clicks "Save Configuration" button</li> <li>9. System validates all point values are within acceptable ranges</li> <li>10. System displays confirmation: "Scoring configuration updated successfully"</li> <li>11. System applies new point values to all future actions (existing scores unchanged)</li> </ul>
<b>Alternate Flow</b>	None
<b>Exception</b>	None

*Table 4.1.19 Table for Use Case Specification 19*

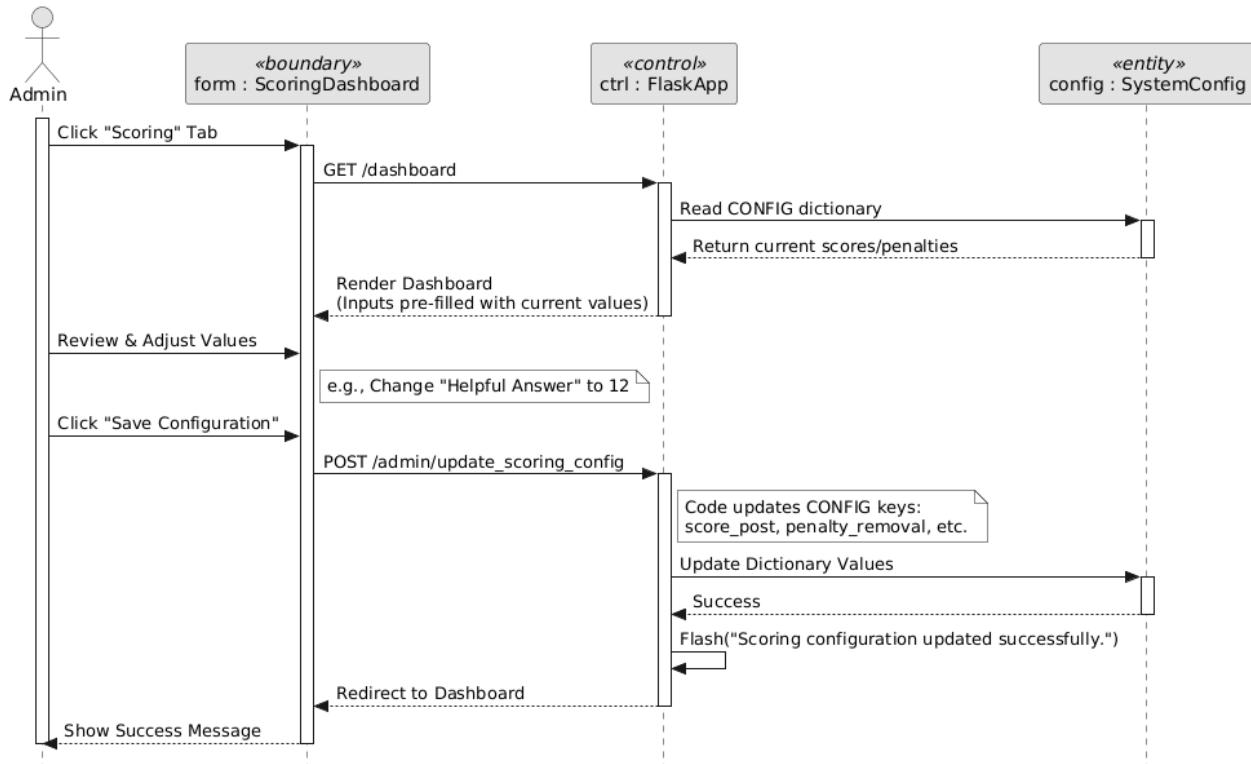


Diagram 4.1.19 Sequence Diagram Manage Scoring System

#### 4.1.20 Process Appeals

The admin is able to display appeal queue showing user's current score, restriction duration, violation history, and appeal reason, offer decision options (approve with point restoration, deny, deny with extended restriction), and send automated notification within 24 hours of decision with explanation.

<b>Use Case ID</b>	UC7
<b>Use Case Name</b>	Process Score Appeals
<b>Primary Actor</b>	Admin
<b>*Pre-Conditions</b>	<ul style="list-style-type: none"> <li>1. Admin is logged into the system</li> <li>2. User has submitted appeal</li> </ul>
<b>Post-Conditions</b>	<ul style="list-style-type: none"> <li>1. Appeal marked "approved" in system</li> <li>2. User's score restored to exactly 60 points (60%)</li> </ul>
<b>Basic Flow</b>	<ol style="list-style-type: none"> <li>1. Admin navigates to Scoring System section from left sidebar</li> <li>2. System displays "Restricted Users (Below 60%)" table</li> <li>3. Admin identifies users with pending appeals (shown in "Appeals" column with count)</li> <li>4. Admin clicks "Review Appeal" button for user with pending appeal</li> </ol>

	<p>5. System opens appeal review interface displaying:</p> <ul style="list-style-type: none"> <li>○ User's name and current score percentage</li> <li>○ Restriction start date and duration restricted</li> <li>○ Total number of violations and violation types</li> <li>○ User's written appeal explanation</li> </ul> <p>6. Admin clicks "History" button to view complete behavioral timeline</p> <p>7. System displays comprehensive user history:</p> <ul style="list-style-type: none"> <li>○ Chronological list of all violations with dates and penalties</li> <li>○ Content that was removed with violation descriptions</li> <li>○ Warnings issued and admin notes</li> <li>○ Previous appeals and outcomes</li> </ul> <p>8. Admin evaluates appeal</p> <p>9. Admin determines appeal decision:</p> <ul style="list-style-type: none"> <li>○ Approve and restore to 60% threshold</li> <li>○ Deny appeal with 7-day extension before resubmission</li> </ul> <p>10. Admin selects "Approve Appeal" option</p> <p>11. System displays confirmation dialog: "Restore user to 60% score? Current score: \$\$%"</p> <p>12. Admin confirms decision</p> <p>13. System immediately restores user's score to exactly 60 points (60%)</p> <p>14. System automatically removes posting restrictions</p> <p>15. System sends automated notification to user</p> <p>16. System marks appeal as "Approved - Resolved"</p>
<b>Alternate Flow</b>	<p>9a. If admin select "deny appeal" option</p> <ol style="list-style-type: none"> <li>1. System prompts for denial reason selection from dropdown</li> <li>2. Admin enter detailed explanation of denial reason</li> <li>3. Admin confirm denial decision</li> <li>4. system send notifications to user</li> </ol>
<b>Exception</b>	None

*Table 4.1.20 Table for Use Case Specification 20*

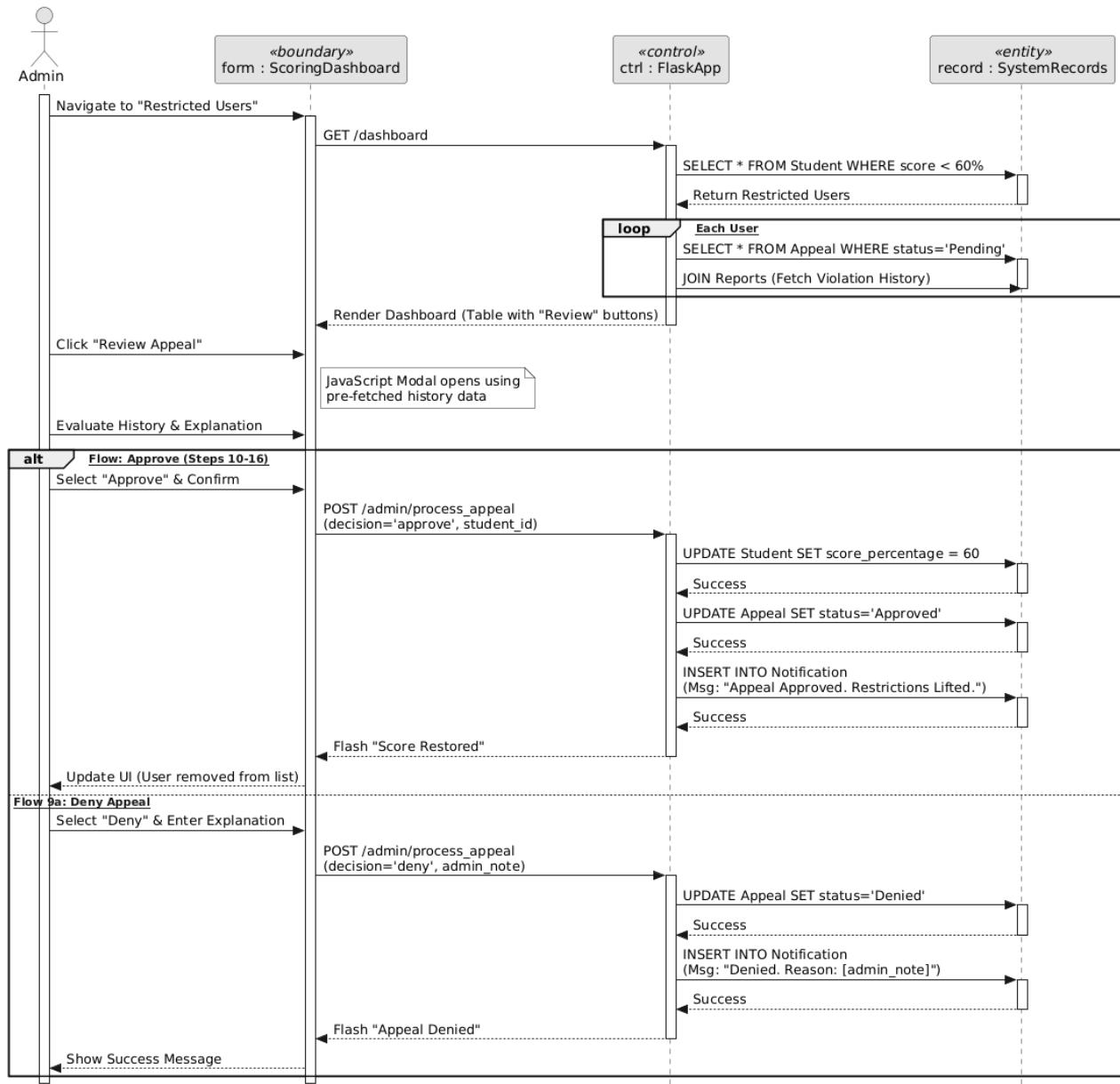


Diagram 4.1.20 Sequence Diagram Process Appeals

#### 4.1.21 Suspend Accounts

The admin is able to temporarily or permanently suspends user accounts for serious or repeated violations by documenting suspension reasons, applying automatic score penalties, and managing appeal processes to maintain platform safety and community standards.

<b>Use Case ID</b>	UC8
<b>Use Case Name</b>	Suspend Accounts
<b>Primary Actor</b>	Admin

<b>*Pre-Conditions</b>	<ol style="list-style-type: none"> <li>1. Admin is logged into the system</li> <li>2. User account has score below 60%</li> </ol>
<b>Post-Conditions</b>	<ol style="list-style-type: none"> <li>1. Account suspended for specified duration</li> <li>2. User moved to suspended list</li> <li>3. User cannot log in at all during suspension</li> </ol>
<b>Basic Flow</b>	<ol style="list-style-type: none"> <li>1. Admin is on Scoring System page reviewing restricted users</li> <li>2. Admin sees table of users with scores below 60%</li> <li>3. Admin identifies user who needs suspension:             <ul style="list-style-type: none"> <li>o Score has been below 60% for long time</li> <li>o Multiple violations continue</li> <li>o Refuses to improve behavior</li> </ul> </li> <li>4. Admin clicks on user's name in the table</li> <li>5. System displays user profile with:             <ul style="list-style-type: none"> <li>o Current score (e.g., 45%)</li> <li>o Time restricted (e.g., 30 days)</li> <li>o Total violations (e.g., 5)</li> <li>o Recent activity</li> <li>o Previous appeals (denied 3 times)</li> </ul> </li> <li>6. Admin clicks "Suspend Account" button</li> <li>7. System opens suspension dialog showing:             <ol style="list-style-type: none"> <li>8. User's current score: 45%</li> <li>9. Restriction duration: 30 days</li> </ol> </li> <li>10. System displays suspension options:             <ul style="list-style-type: none"> <li>o Duration: 7 days, 14 days, 30 days, Indefinite</li> <li>o Reason dropdown (Required violations, Refused to improve, Harmful behavior)</li> </ul> </li> <li>11. System shows suspension summary</li> <li>12. Admin clicks "Confirm Suspension"</li> <li>13. System immediately suspends account</li> <li>14. System completely blocks all login access</li> <li>15. System sends email to user with:             <ul style="list-style-type: none"> <li>o Suspension reason</li> <li>o Duration (14 days)</li> <li>o Current score (45%)</li> <li>o Violations summary</li> <li>o Appeal instructions (can appeal after 7 days)</li> <li>o Reinstatement date</li> </ul> </li> <li>16. System creates suspension record</li> <li>17. System removes user from restricted users table (moved to suspended section)</li> </ol>

<b>Alternate Flow</b>	None
<b>Exception</b>	None

Table 4.1.21 Table for Use Case Specification 21

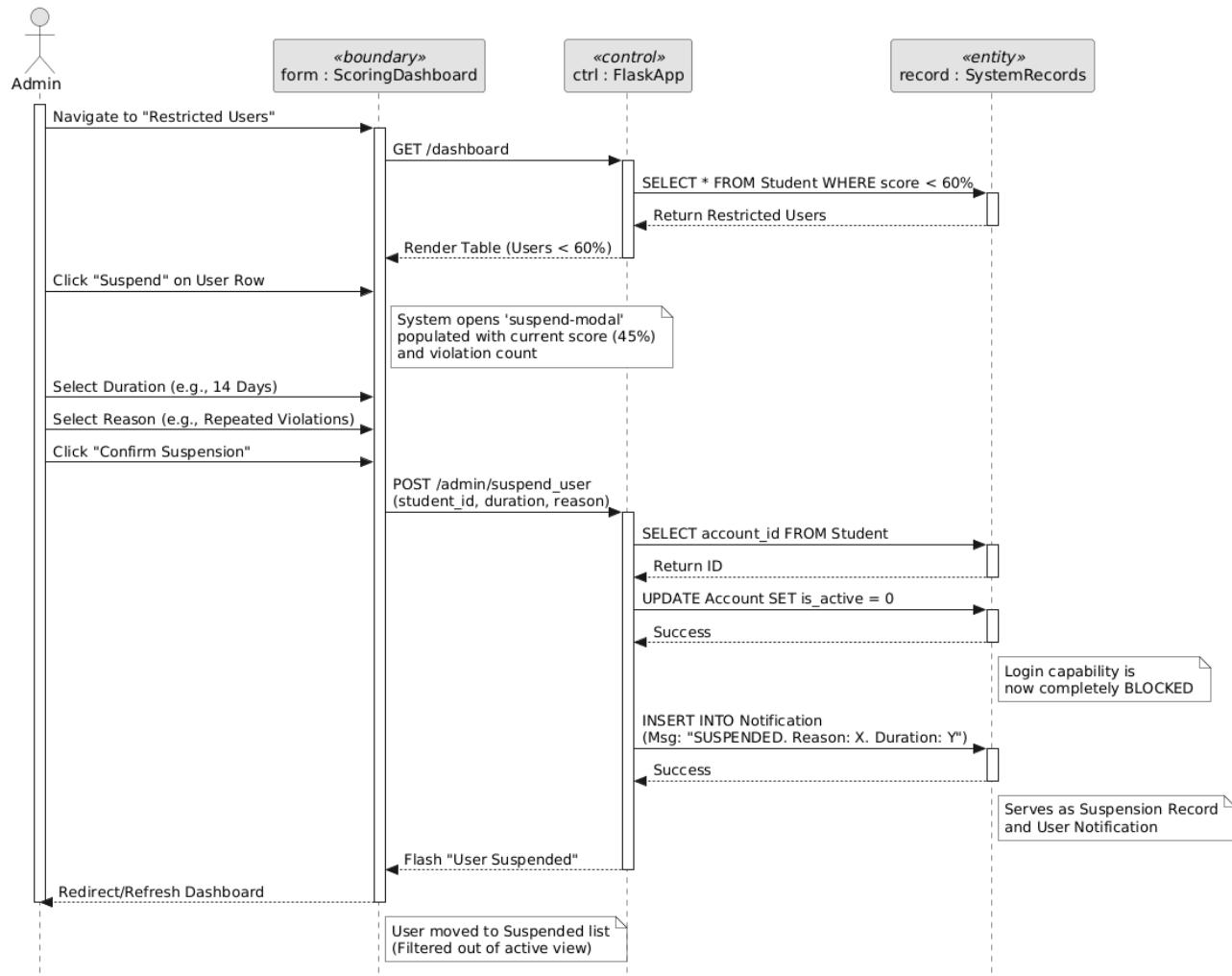


Diagram 4.1.21 Sequence Diagram Suspend Account

#### 4.1.22 Review Reported Messages

The moderator is able to review messages reported by students.

<b>Use Case ID</b>	UC21
<b>Use Case Name</b>	Review reported messages
<b>Primary Actor</b>	Moderator
<b>Pre-Conditions</b>	Moderator is logged in; Student has reported password
<b>Post-Conditions</b>	Students record is updated; students record remains the same

<b>Basic Flow</b>	<ol style="list-style-type: none"> <li>1. Moderator opens the dashboard.</li> <li>2. System displays list of reported messages.</li> <li>3. Moderator selects a message to review.</li> <li>4. Moderator marks the message as violation.</li> <li>5. System updates the database and removes the message from pending reports.</li> </ol>
<b>Alternate Flow</b>	<ol style="list-style-type: none"> <li>4a. Moderator decides the message does <b>not</b> break guidelines.</li> <li>4b. Moderator marks the message as safe.</li> <li>4c. System updates the database</li> </ol>
<b>Exception</b>	-

Table 4.1.22 Table for Use Case Specification 22

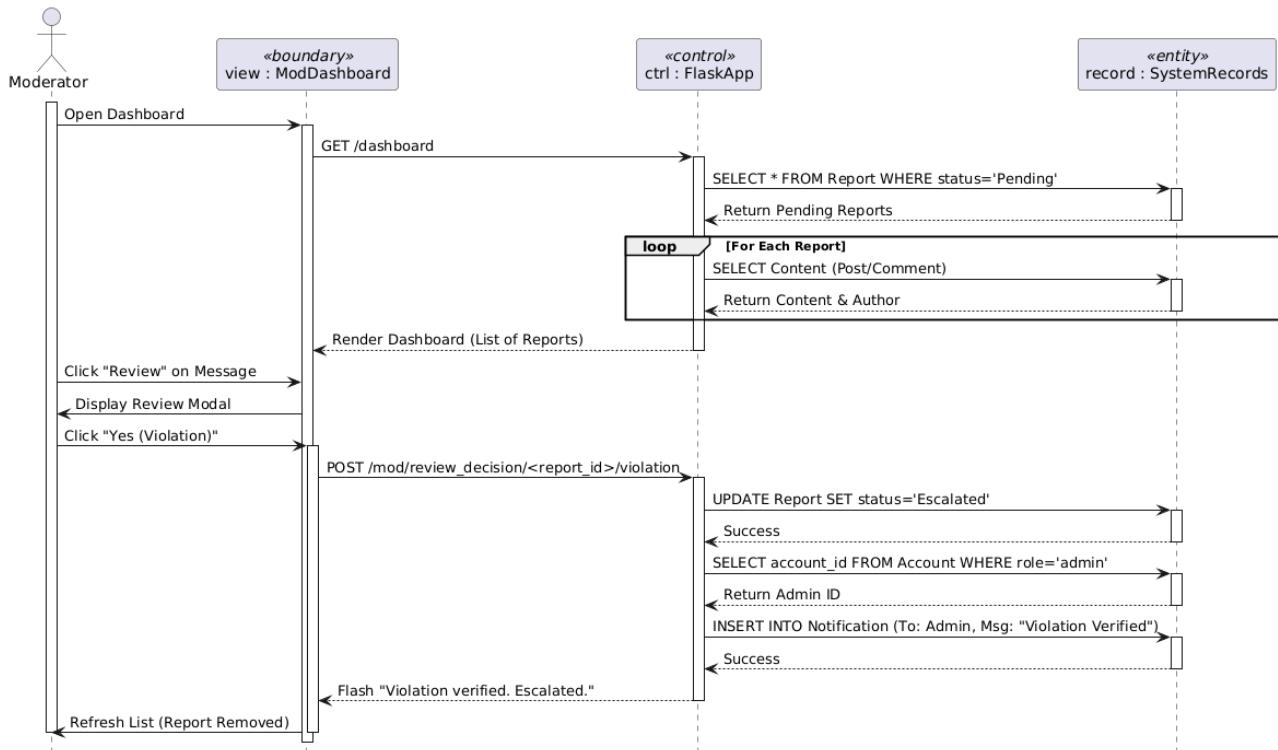


Diagram 4.1.22 Sequence Diagram Suspend Account

#### 4.1.23 Flag Student Account

The moderator is able to flag accounts to be reviewed by admins.

<b>Use Case ID</b>	UC27
<b>Use Case Name</b>	Flag Student Account
<b>Primary Actor</b>	Moderator

<b>Pre-Conditions</b>	Moderator is logged in; student has more than one violation
<b>Post-Conditions</b>	Student account is to be reviewed by the admin
<b>Basic Flow</b>	<ol style="list-style-type: none"> <li>1. Moderator opens moderator dashboard</li> <li>2. Moderator selects a student account.</li> <li>3. Moderator chooses the flag option.</li> <li>4. System flags the student account.</li> </ol>
<b>Alternate Flow</b>	3a. Moderator cancels the operation.
<b>Exception</b>	None

Table 4.1.23 Table for Use Case Specification 23

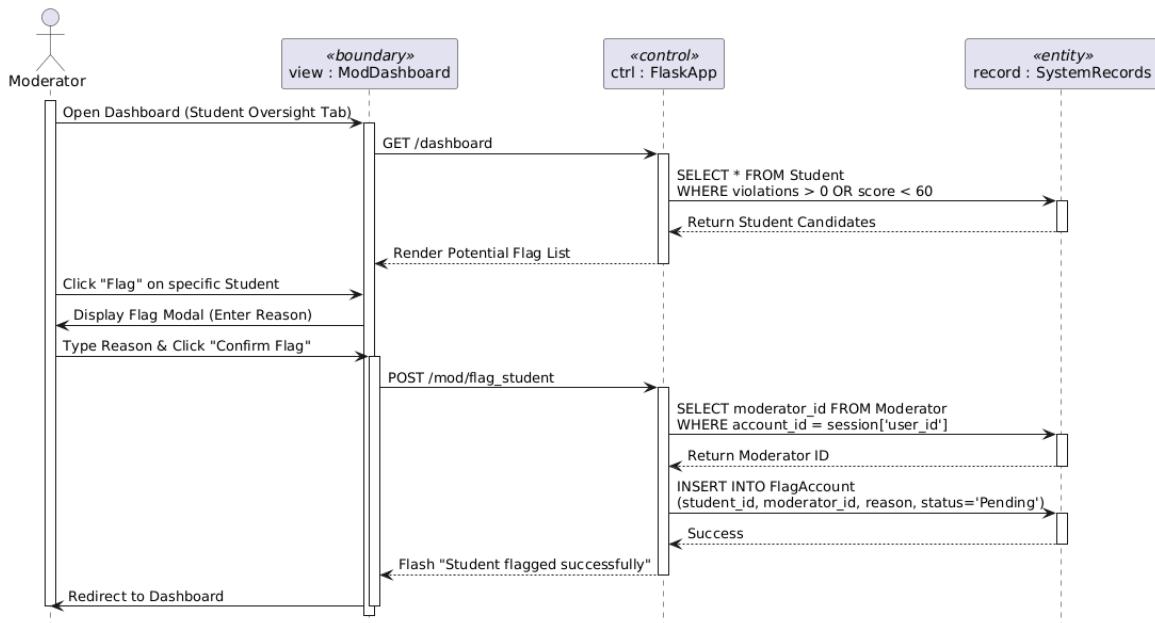


Diagram 4.1.23 Sequence Diagram Flag Student Account

#### 4.1.24 Make Announcements

The moderator is able to make announcements that all students will be notified of.

<b>Use Case ID</b>	UC28
<b>Use Case Name</b>	Make Announcements
<b>Primary Actor</b>	Moderator
<b>Pre-Conditions</b>	Moderator is logged in
<b>Post-Conditions</b>	Announcement is sent to students
<b>Basic Flow</b>	<ol style="list-style-type: none"> <li>1. Moderator selects create announcement.</li> <li>2. Moderator enters announcement details.</li> <li>3. Moderator submits the announcement.</li> <li>4. System publishes the announcement</li> </ol>
<b>Alternate Flow</b>	None

<b>Exception</b>	None
------------------	------

Table 4.1.24 Table for Use Case Specification 24

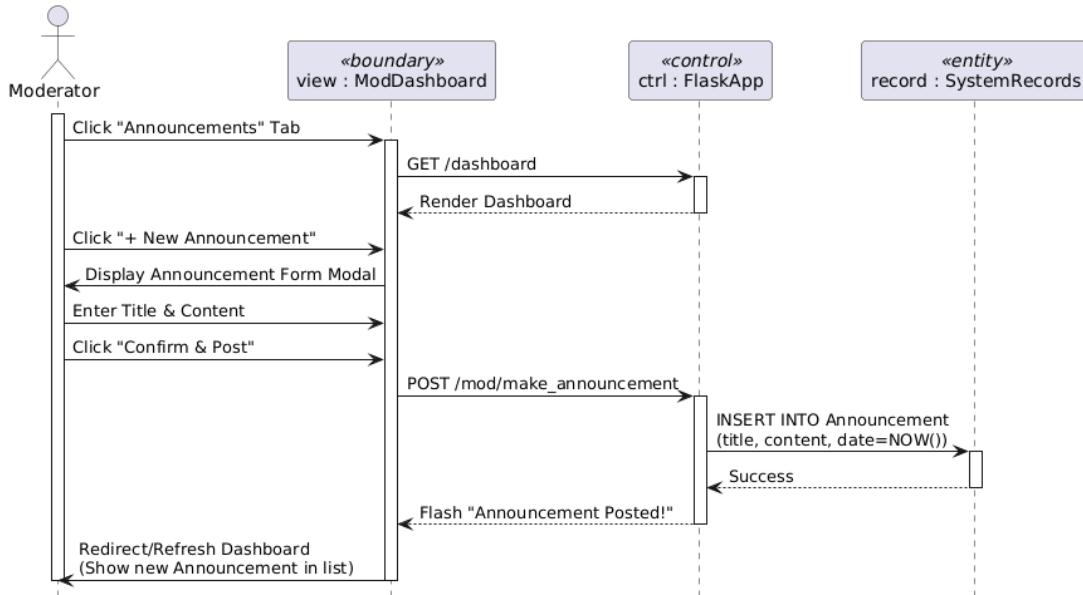


Diagram 4.1.24 Sequence Diagram Make Announcements

#### 4.1.25 Delete Forum Posts

The moderator is able to delete forum posts that break platform guidelines.

<b>Use Case ID</b>	UC29
<b>Use Case Name</b>	Delete Forum Posts
<b>Primary Actor</b>	Moderator
<b>Pre-Conditions</b>	Moderator is logged in; post is on the forum
<b>Post-Conditions</b>	Post is deleted
<b>Basic Flow</b>	<ol style="list-style-type: none"> <li>3. Moderator selects a forum post.</li> <li>4. Moderator confirms deletion.</li> <li>5. System deletes the forum post.</li> <li>6. System updates the forum display</li> </ol>
<b>Alternate Flow</b>	None
<b>Exception</b>	None

Table 4.1.25 Table for Use Case Specification 25

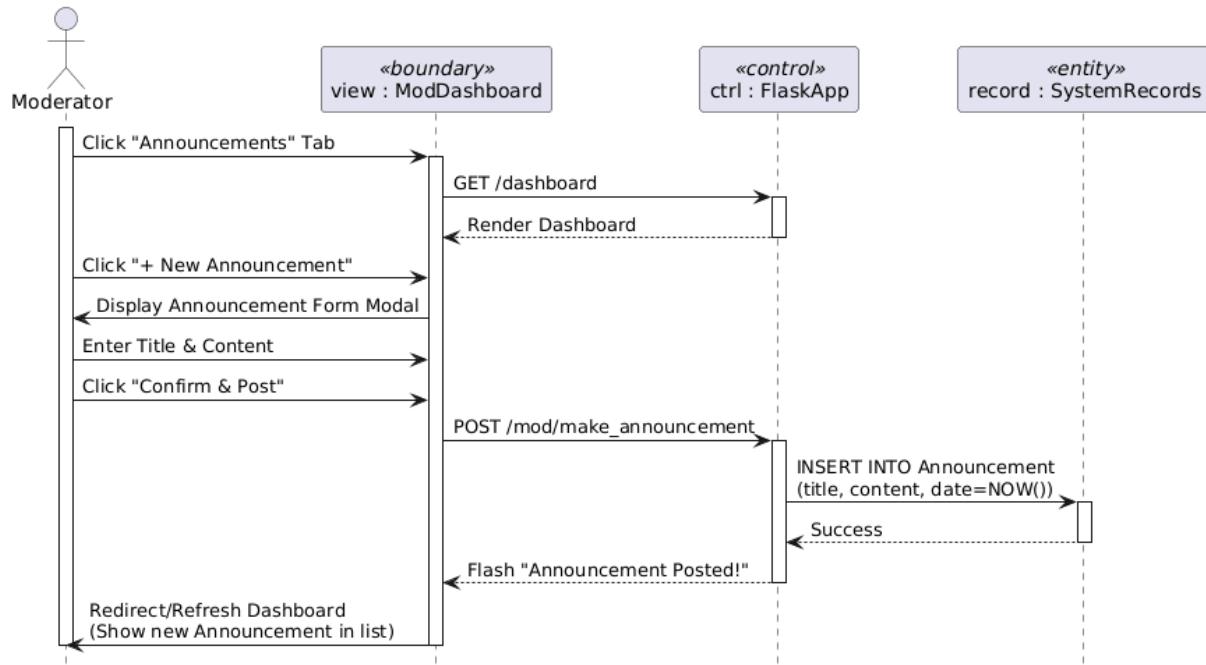


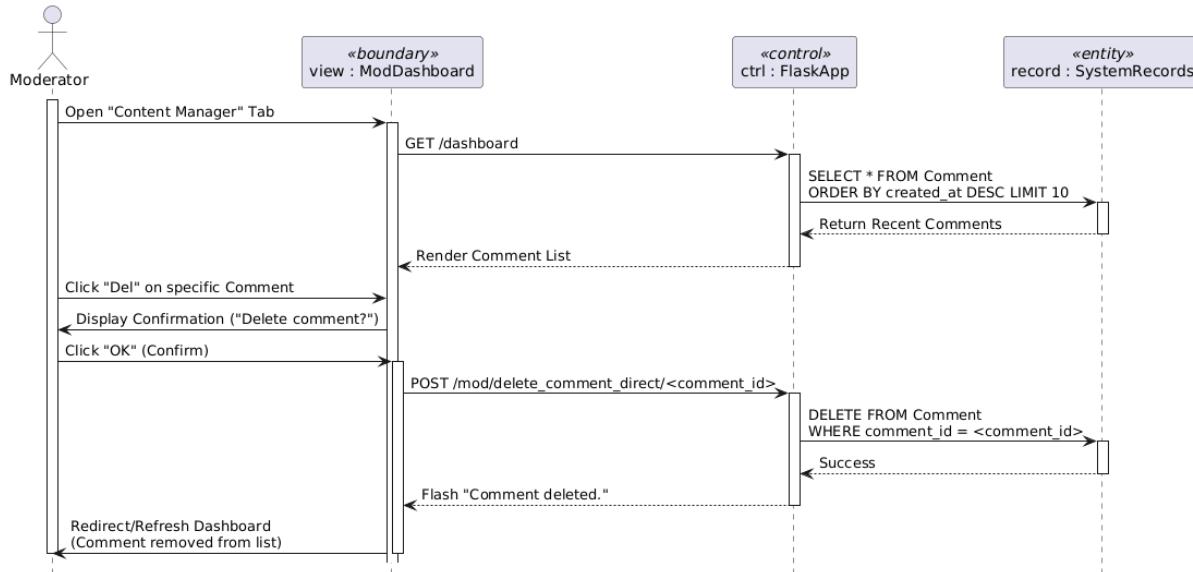
Diagram 4.1.25 Sequence Diagram Delete Forum Posts

#### 4.1.26 Delete Forum Comments

The moderator is able to delete comments under forums posts that break guidelines.

<b>Use Case ID</b>	UC30
<b>Use Case Name</b>	Delete Forum Comments
<b>Primary Actor</b>	Moderator
<b>Pre-Conditions</b>	Moderator is logged in; Post is in the forum; Comment is under post
<b>Post-Conditions</b>	Comment is deleted
<b>Basic Flow</b>	<ol style="list-style-type: none"> <li>1. Moderator selects a forum comment.</li> <li>2. Moderator confirms deletion.</li> <li>3. System deletes the comment.</li> <li>4. System updates the forum display</li> </ol>
<b>Alternate Flow</b>	None
<b>Exception</b>	None

Table 4.1.26 Table for Use Case Specification 26



**Diagram 4.1.26 Sequence Diagram Delete Forum Comments**

#### 4.1.27 Manage caseload

The counselor is able to efficiently organize, view, and update their assigned student roster through an intuitive dashboard that displays student status, priority levels, and recent activities, enabling effective tracking and prioritization of student support needs.

<b>Use Case ID</b>	UC32
<b>Use Case Name</b>	Manage caseload
<b>Primary Actor</b>	Counselor
<b>*Pre-Conditions</b>	1. Counselor is logged in 2. Counselor has assigned students 3. Counselor has permission to view/manage caseload
<b>Post-Conditions</b>	1. Student information is updated in system 2. Changes are recorded in audit log 3. Dashboard reflects updated caseload
<b>Basic Flow</b>	1. Counselor navigates to "My Caseload" page 2. System displays a list of assigned students with key indicators 3. Counselor selects a student to view details 4. Counselor updates student information as needed 5. Counselor saves changes
<b>Alternate Flow</b>	3a. Filter/Search Caseload 1. Counselor uses search/filter options 2. System displays filtered results 3. Counselor selects from filtered list
<b>Exception</b>	No Students Assigned 1. System displays "No students assigned" message

Table 4.1.27 Table for Use Case Specification 27

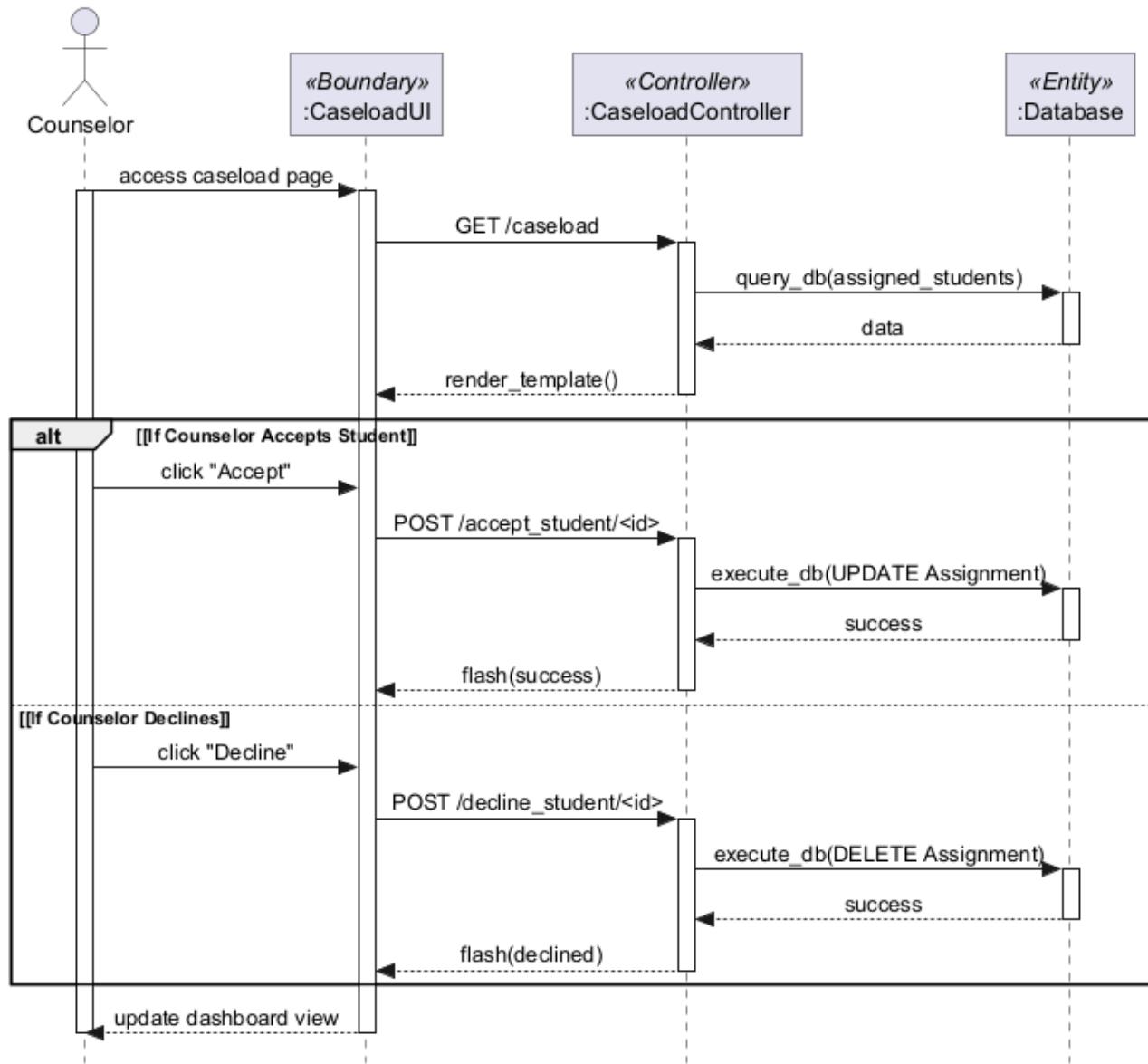


Diagram 4.1.27 Sequence Diagram Manage Caseload

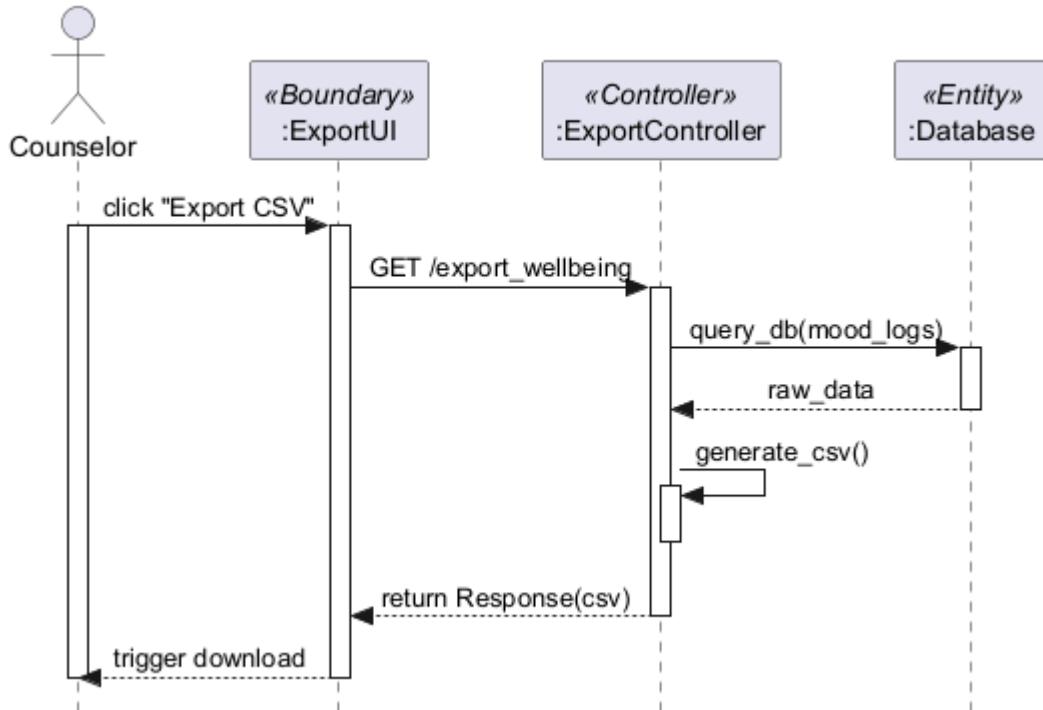
#### 4.1.28 Export data

The counselor is able to generate and download structured reports of student well-being data, intervention records, and appointment histories in various formats (Excel, CSV, PDF) while ensuring compliance with data privacy regulations through automatic anonymization of sensitive information.

<b>Use Case ID</b>	UC33
<b>Use Case Name</b>	Export data
<b>Primary Actor</b>	Counselor

<b>*Pre-Conditions</b>	1. Counselor is logged in 2. Counselor has data viewing permissions 3. Export functionality is enabled for counselor's role
<b>Post-Conditions</b>	1. Data file is generated and downloaded 2. Export action is logged in audit trail 3. Downloaded file is in specified format
<b>Basic Flow</b>	1. Counselor navigates to "Export Data" section 2. System displays export options and filters 3. Counselor selects: - Data range - Student(s) - Data fields - Format like CSV, Excel and PDF 4. Counselor clicks "Export" 5. System generates anonymized data file 6. File is downloaded to counselor's device
<b>Alternate Flow</b>	<b>Email Export</b> 6a. Counselor selects "Email to me" option 1. System generates file and sends to counselor's email
<b>Exception</b>	<b>Large Data Set</b> 1. System warns about large export size 2. Option to narrow filters or proceed with background processing

Table 4.1.28 Table for Use Case Specification 28



*Diagram 4.1.28 Sequence Diagram Export Data*

#### **4.1.29 Verify assignments**

The counselor is able to review and respond to new student assignment requests from administrators by accepting or declining them based on current caseload capacity, with the ability to request additional information before making decisions.

<b>Use Case ID</b>	UC34
<b>Use Case Name</b>	Verify assignments
<b>Primary Actor</b>	Counselor
<b>*Pre-Conditions</b>	<ol style="list-style-type: none"><li>1. Counselor is logged in</li><li>2. Admin has assigned new students to counselor</li><li>3. Counselor has permission to review assignments</li></ol>
<b>Post-Conditions</b>	<ol style="list-style-type: none"><li>1. Assignment is accepted/declined</li><li>2. Caseload is updated accordingly</li><li>3. Admin is notified of decision</li></ol>
<b>Basic Flow</b>	<ol style="list-style-type: none"><li>1. System notifies counselor of pending assignments</li><li>2. Counselor navigates to "Pending Assignments"</li><li>3. System displays list with student details</li><li>4. Counselor reviews each assignment</li><li>5. Counselor selects "Accept" or "Decline"</li><li>6. If declined, counselor provides reason</li></ol>
<b>Alternate Flow</b>	<ol style="list-style-type: none"><li>4a. Request More Information<ol style="list-style-type: none"><li>1. Counselor selects "Need More Info"</li><li>2. System sends request to admin</li><li>3. Decision deferred until info received</li></ol></li></ol>
<b>Exception</b>	Assignment Expired <ol style="list-style-type: none"><li>1. System shows assignment is no longer pending</li><li>2. Option to contact admin about reassignment</li></ol>

*Table 4.1.29 Table for Use Case Specification 29*

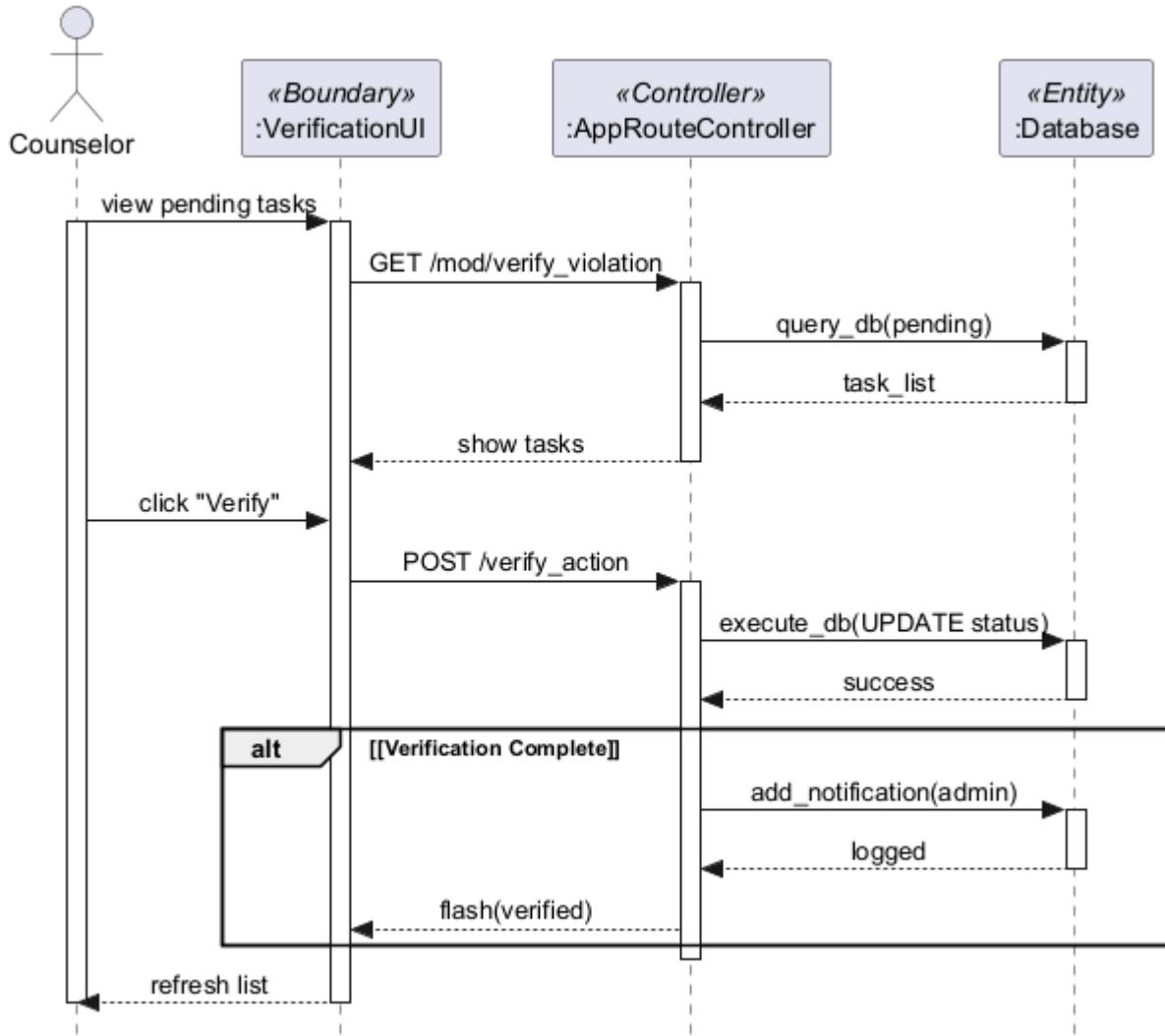


Diagram 4.1.29 Sequence Diagram Verify Assignments

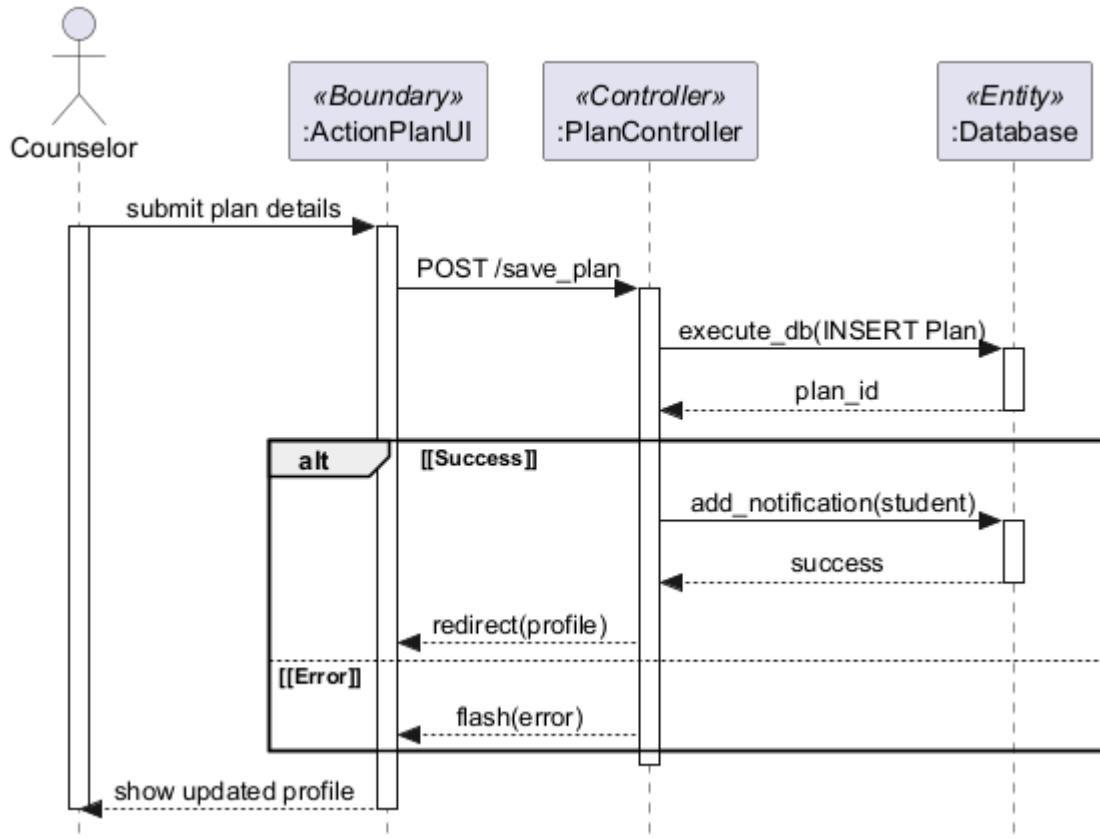
#### 4.1.30 Manage action plans

The counselor is able to create, update, and monitor personalized therapeutic action plans for each student, setting measurable goals, defining intervention strategies, and tracking progress toward positive outcomes through collaborative planning with students.

<b>Use Case ID</b>	UC35
<b>Use Case Name</b>	Manage action plans
<b>Primary Actor</b>	Counselor
<b>*Pre-Conditions</b>	1. Counselor is logged in 2. Students have been assessed 3. Counselor has permission to create/manage plans

<b>Post-Conditions</b>	<ol style="list-style-type: none"> <li>Action plan is created/updated</li> <li>Plan is linked to student profile</li> <li>Progress tracking is initialized</li> </ol>
<b>Basic Flow</b>	<ol style="list-style-type: none"> <li>Counselor selects student from caseload</li> <li>Counselor selects "Action Plans"</li> <li>System displays existing plans or options to create new</li> <li>Counselor creates or edits plans with: <ul style="list-style-type: none"> <li>- Goals</li> <li>- Strategies</li> <li>- Timeline</li> <li>- Progress metrics</li> </ul> </li> <li>Counselor saves plan</li> <li>System generates plan summary</li> </ol>
<b>Alternate Flow</b>	<ol style="list-style-type: none"> <li>Use Template <ol style="list-style-type: none"> <li>Counselor selects from template library</li> <li>System pre-populates plan structure</li> <li>Counselor customizes for student</li> </ol> </li> </ol>
<b>Exception</b>	<b>Student Has Active Plan</b> <ol style="list-style-type: none"> <li>System warns about existing active plan</li> <li>Option to archive old plan or create additional plan</li> </ol>

Table 4.1.30 Table for Use Case Specification 30



*Diagram 4.1.30 Sequence Diagram Manage Action Plans*

#### **4.1.31 Schedule appointments**

The counselor is able to manage their counseling calendar by setting availability, booking appointments with students, sending automated reminders, and handling rescheduling or cancellations through an integrated scheduling system.

<b>Use Case ID</b>	UC36
<b>Use Case Name</b>	Schedule appointments
<b>Primary Actor</b>	Counselor
<b>*Pre-Conditions</b>	<ol style="list-style-type: none"><li>1. Counselor is logged in</li><li>2. Counselor has set availability</li><li>3. Student is in counselor's caseload</li></ol>
<b>Post-Conditions</b>	<ol style="list-style-type: none"><li>1. Appointment is scheduled in calendar</li><li>2. Notifications are sent to student</li><li>3. Counselor's calendar is updated</li></ol>
<b>Basic Flow</b>	<ol style="list-style-type: none"><li>1. Counselor selects student from caseload</li><li>2. Counselor clicks "Schedule Appointment"</li><li>3. System displays counselor's available slots</li><li>4. Counselor selects date/time</li><li>5. Counselor selects appointment type and duration</li><li>6. The counselor confirms booking</li><li>7. System sends confirmation to student</li></ol>
<b>Alternate Flow</b>	<ol style="list-style-type: none"><li>2a. Student Self-Booking<ol style="list-style-type: none"><li>1. Student books from available slots</li><li>2. System confirms with both parties</li></ol></li></ol>
<b>Exception</b>	<p>Time Slot Unavailable</p> <ol style="list-style-type: none"><li>1. System shows slot is no longer available</li><li>2. Option to select different time or waitlist</li></ol>

*Table 4.1.31 Table for Use Case Specification 31*

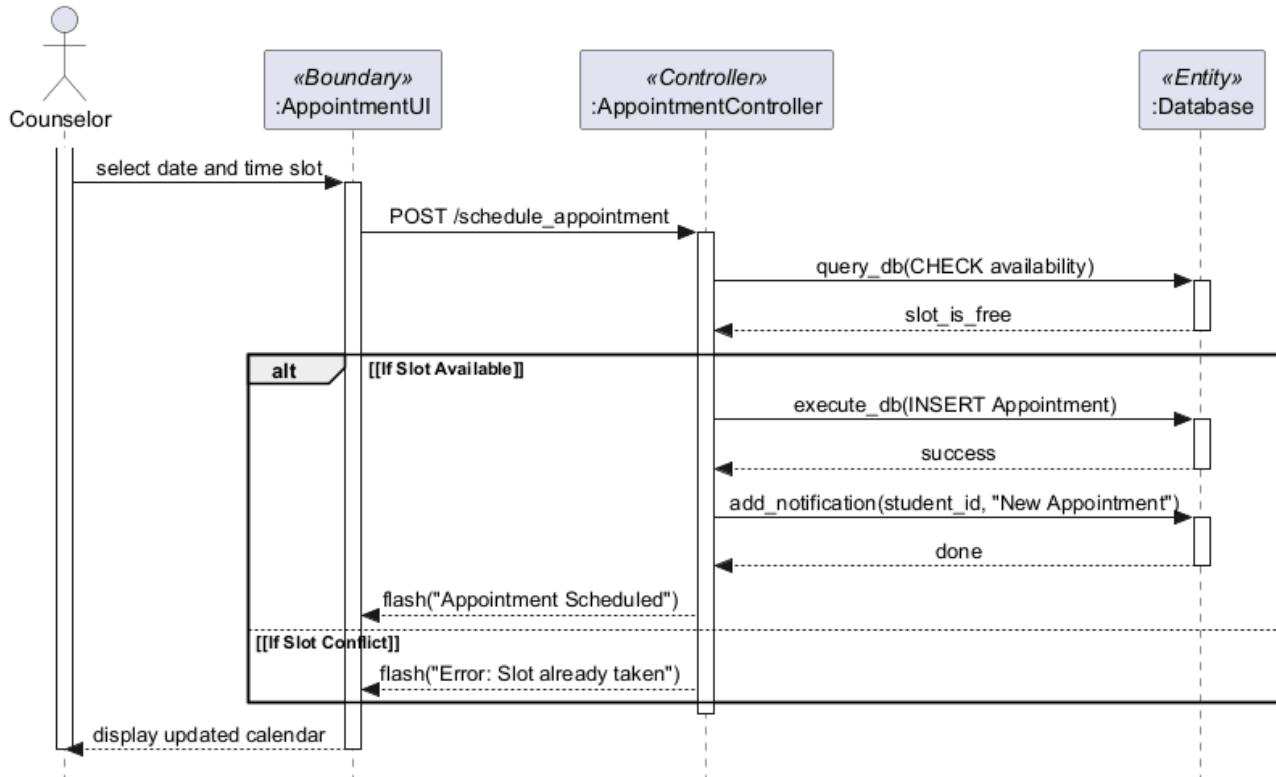


Diagram 4.1.31 Sequence Diagram Schedule Appointments

## 4.2 State Diagram

This State Diagram illustrates the dynamic behavior of the **Digital Peer Support System**, mapping the various states a user account can inhabit and the transitions between them based on specific events and conditions. The system logic is segmented into five distinct lifecycles corresponding to the primary user roles: **Guest**, **Student**, **Admin**, **Moderator**, and **Counselor**.

### 1. Authentication & Entry Point

The lifecycle begins at the **Guest** state (bottom right), which serves as the system's entry point.

- **Transitions:** From the Guest state, a user can transition to an authenticated role via the Login event.
- **Sign Up & Recovery:** Users can also transition to the SignUp state to create a new account or the ForgotPass state to initiate password recovery before returning to the Guest state to log in.
- **Role Routing:** Upon successful authentication, the system directs the user to one of four role-specific composite states: Student, Admin, Moderator, or Counselor.

### 2. Student Lifecycle

The Student lifecycle (left swimlane) is the most complex, featuring nested substates that govern user privileges and access.

- **Active State:** Upon login, students enter the Active composite state. This state allows parallel execution of social and wellness activities:
  - **Social Interactions:** Students can access features such as MatchUp (to find peers), Chatting, and forum activities (Post, Comment, Like).
  - **Wellness Tools:** Students can perform a Daily Check-in to log their mood or BookAppointment to request sessions with counselors.
- **Restricted State:** If a student's safety score drops below a critical threshold (e.g., < 60%), they automatically transition from Active to Restricted. In this state, social features are disabled. The user must submit an **Appeal** to regain access. If the appeal is approved (Score restored to > 60%), they transition back to Active.
- **Suspended State:** Repeated or severe violations result in a transition to the Suspended state. This is a terminal state for the session where access is revoked until administrative intervention occurs or the suspension duration expires.

### **3. Admin Lifecycle**

The Admin lifecycle (center-left swimlane) focuses on high-level system management.

- **Monitoring State:** This is the central hub for administrators. From here, they can transition to specific management tasks:
  - **User Management:** Admins handle UserSuspension and review Appeals submitted by restricted students.
  - **Counselor Assignment:** Admins manually link students to counselors via the AssignCounselor state.
  - **Escalations:** Admins review critical reports escalated by Moderators in the ReviewReports state.
  - **Configuration:** Admins can adjust system parameters, such as scoring thresholds, in the SystemConfig state.

### **4. Moderator Lifecycle**

The Moderator lifecycle (center-right swimlane) is dedicated to content safety and community standards.

- **Report Queue:** Moderators primarily reside in this state, reviewing flagged content.
- **Decision Logic:**
  - **Dismiss:** If a report is invalid, the moderator dismisses it (Dismissing), returning the content to a safe state.
  - **Action:** If a violation is found, the moderator can Remove the content or Warn the user.
  - **Escalate:** Severe issues are transitioned to Escalated for Admin review.
- **Announcements:** Moderators also have the authority to create system-wide broadcasts via the CreateAnnouncementstate.

### **5. Counselor Lifecycle**

The Counselor lifecycle (right swimlane) manages the therapeutic workflow.

- **Manage Case:** This is the default state where counselors view their assigned student list.
- **Session Workflow:**
  - **Assignment:** Counselors can Accept or Decline new student assignments from Admins.

- **Scheduling:** Counselors manage their calendar by entering the ScheduleSession state.
- **Documentation:** During or after sessions, counselors transition to SessionNotes to record clinical observations.
- **Action Plans:** Counselors create and update TherapeuticActionPlans, tracking student progress through Active and Completed statuses.

## 6. Exit Transition

All user lifecycles share a common Logout transition, which terminates the current session and returns the system to the initial **Guest** state, completing the cycle.

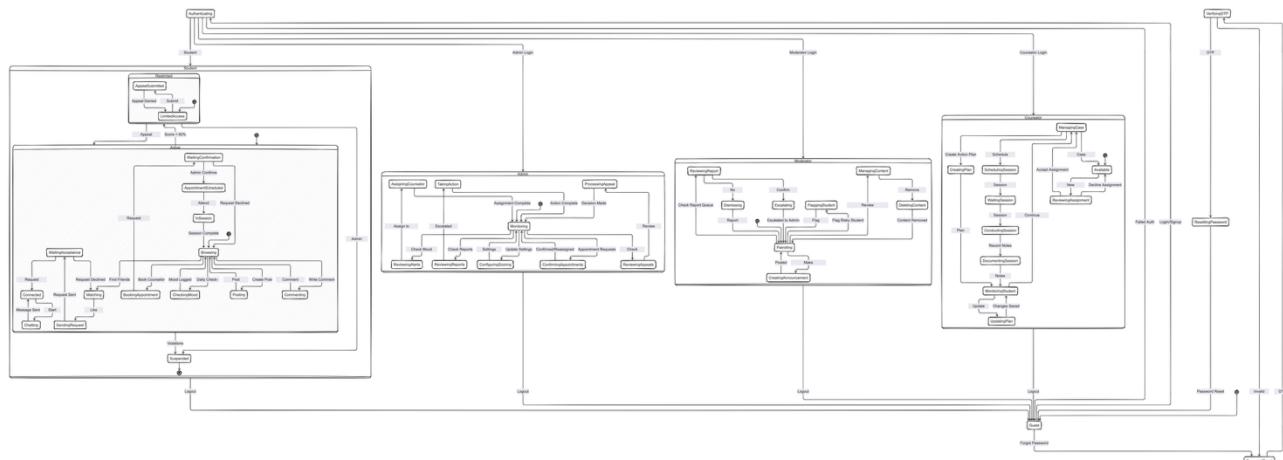


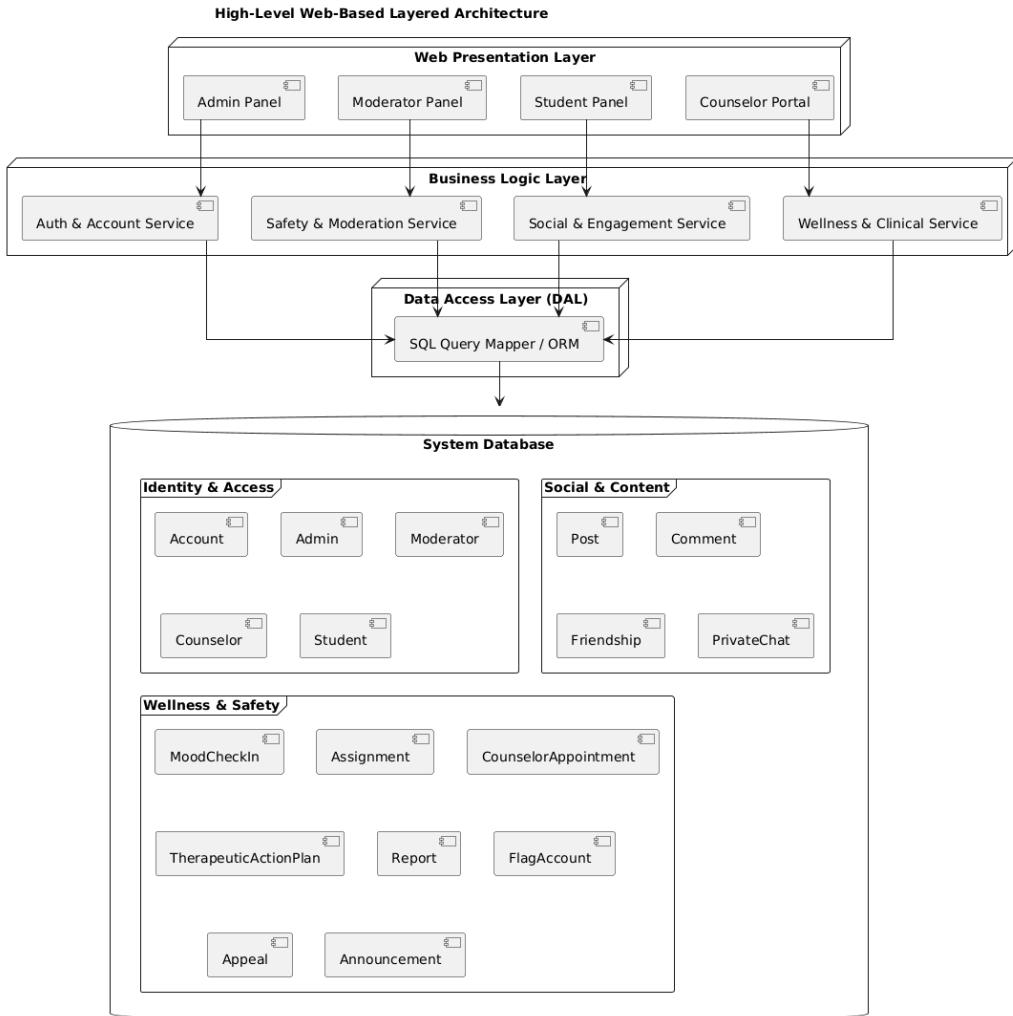
Diagram 4.2.1 [State Diagram](#)

# 5 Architecture Design

## 5.1 Software Architecture

We choose to follow the **layered architecture style** for the digital peer support system, organizing the system website into distinct horizontal layers. This style was chosen to maintain a clear separation of concerns, improve maintainability, and ensure security. By separating the user interface from the underlying data, the architecture lets the system functioning well for every panel implemented based on Student, Moderator, Counselor and Admin actors respectively.

The system is divided into four functional subsystems based on the four actors (Student, Moderator, Counselor, Admin) and their use cases to facilitate efficient development. This modular approach allows team members to work in parallel on isolated domains.



*Diagram 5.1.1 High-Level Layered Architecture Diagram*

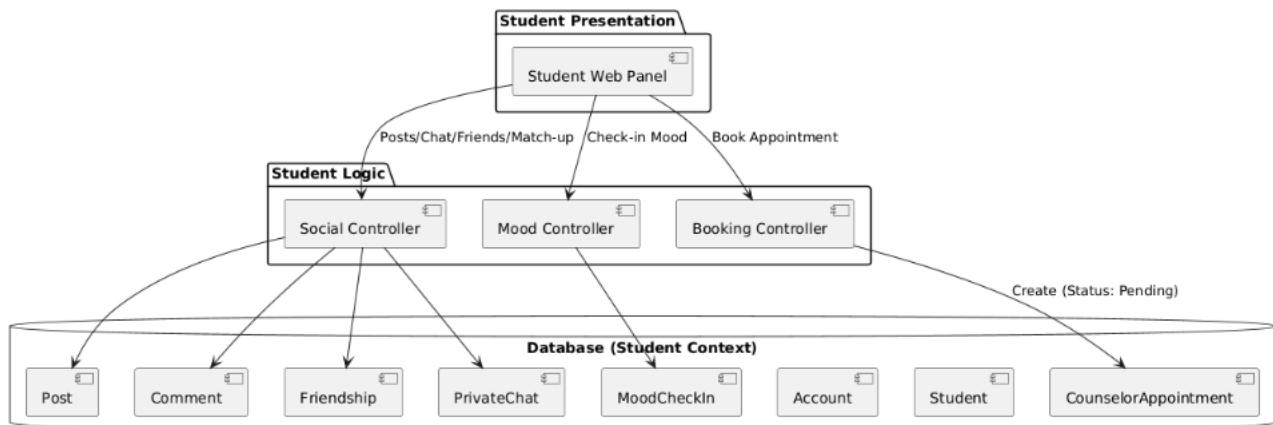
The high-level layered architecture diagram shows how the actors flow through the logic layers to reach the specific table groups.

- **Presentation Layer (Web Panels):** This layer provides actor-specific web interfaces (Panels/Portals) that handle user input and display data.
- **Business Logic Layer (Service Layer):** This "brain" of the system processes the core rules, such as generating mood alerts, managing social engagements, and processing clinical action plans.
- **Data Access Layer (DAL):** Acts as a translation layer (ORM/Query Mapper) that allows the Business Logic to interact with the database without writing raw SQL, ensuring data integrity.
- **Data Layer (Relational Database):** The physical storage based on the provided SQL schema, containing tables for accounts, social content, and clinical wellness records.

### 5.1.1 Student Subsystem

This subsystem is for the Student Panel, focuses on social interaction and well-being.

- Database Tables included: Account, Student, MoodCheckIn, Post, Comment, Friendship, PrivateChat
- Use Cases included: Match-up with people, Report contents, Check-in mood and well-being, Make posts, Comment on posts, Like posts and comments, Build profile, Book counselor appointments, Add friends and Chat Privately

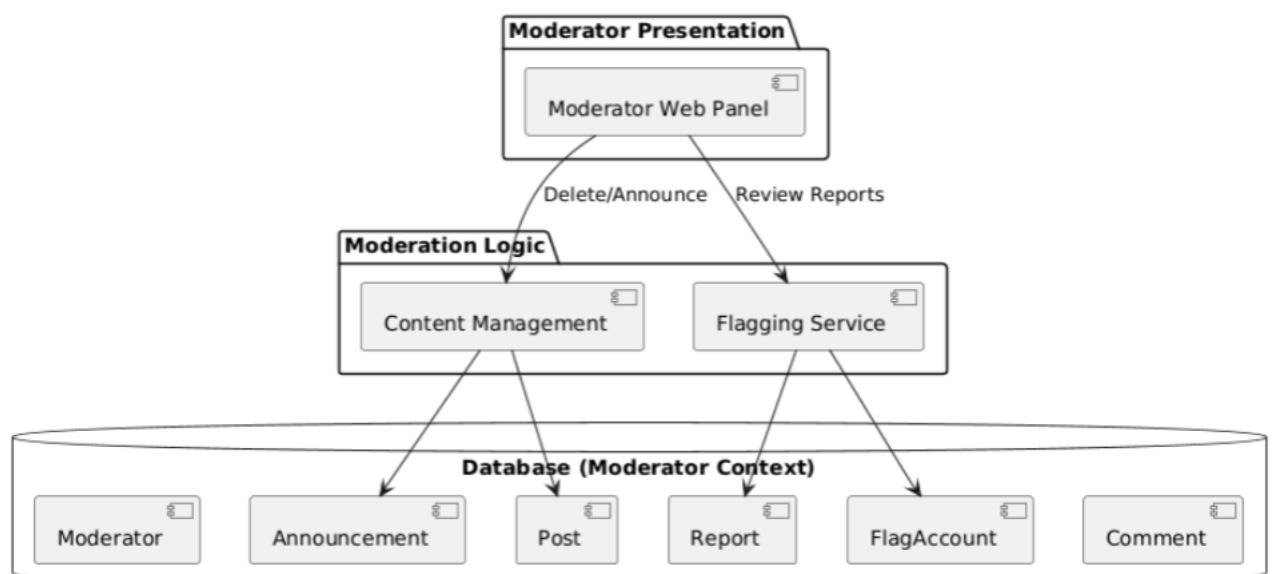


*Diagram 5.1.1.1 Student Subsystem Architecture Diagram*

## 5.1.2 Moderator Subsystem

This subsystem is for the Moderator Panel, focuses on platform safety and communication.

- Database Tables included: Report, FlagAccount, Announcement, Post, Comment
- Use Cases included: Reviewing Reported messages, flag student account, and make announcements, delete forum posts, delete forum comments

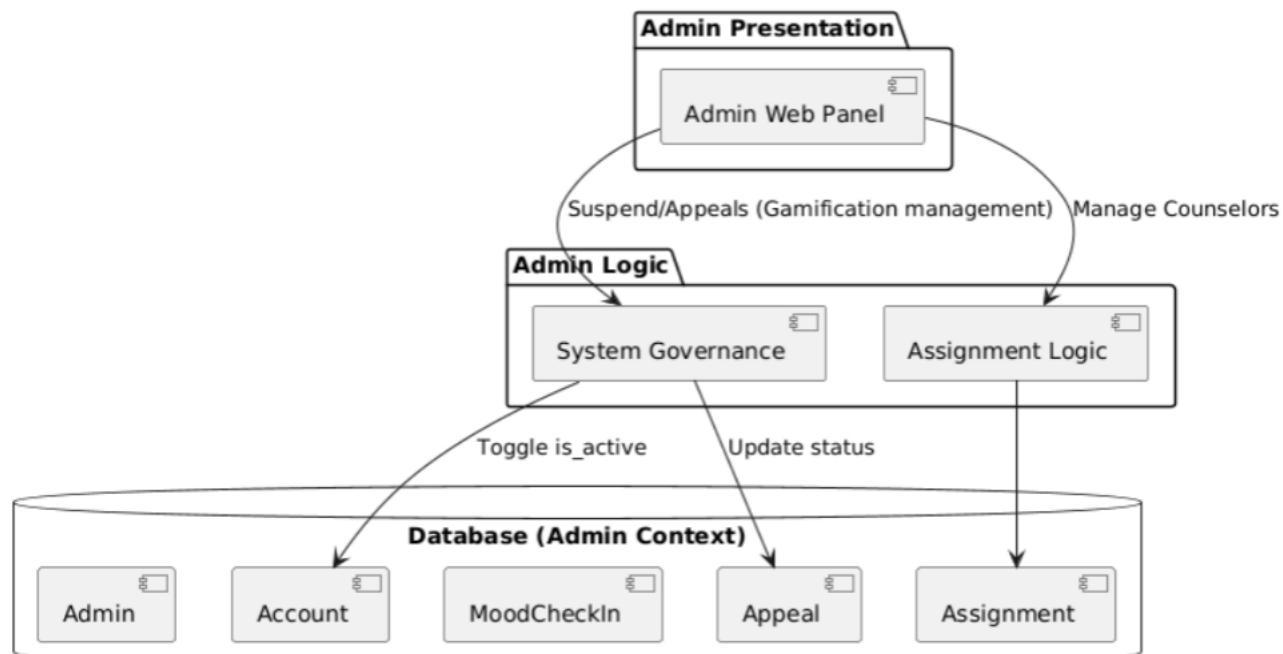


*Diagram 5.1.2.1 Moderator Subsystem Architecture Diagram*

### 5.1.3 Admin Subsystem

This subsystem is for the Admin Panel, focuses on system-wide governance and high-level appeals.

- Database Tables included: Admin, Account (status management), MoodCheckIn (for dashboard monitoring), Appeal, Assignment
- Key Use Cases: Monitor Mood Alert Dashboard, Manage Counselor Assignments, Review Reported Content, Manage Scoring System, Process Appeals, and Suspend Accounts

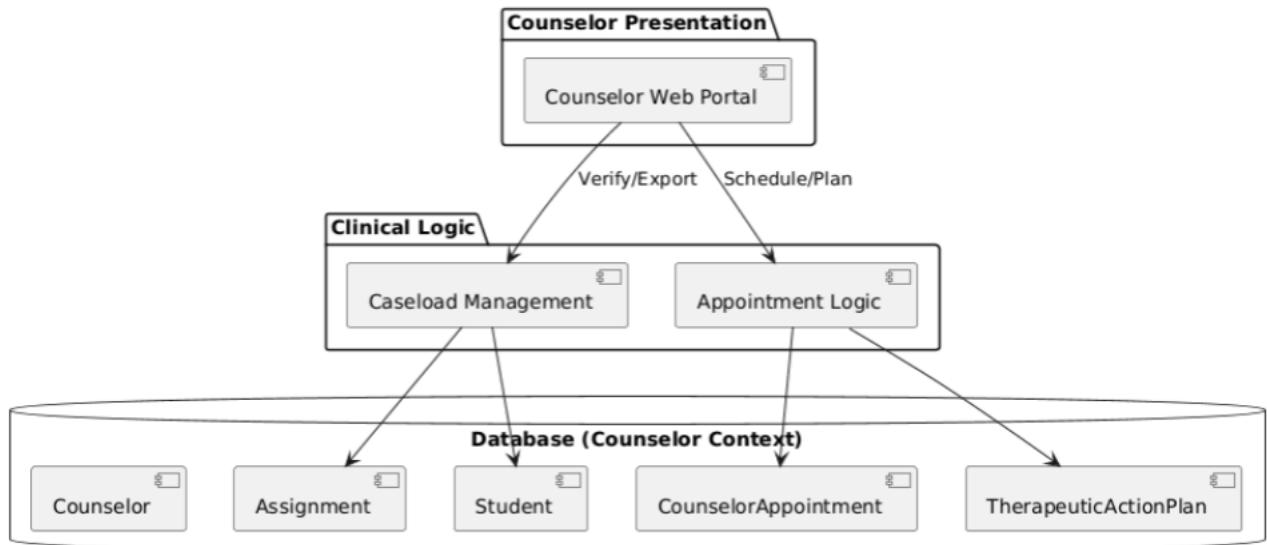


*Diagram 5.1.3.1 Admin Subsystem Architecture Diagram*

### 5.1.4 Counselor Subsystem

This subsystem is for the Counselor Portal, focuses on the caseload management and appointment logic.

- Database Tables included: Counselor, Assignment, CounselorAppointment, TherapeuticActionPlan
- Use Cases included: Manage Caseload, Export data, Verify assignments, Schedule appointments, and Manage action plans



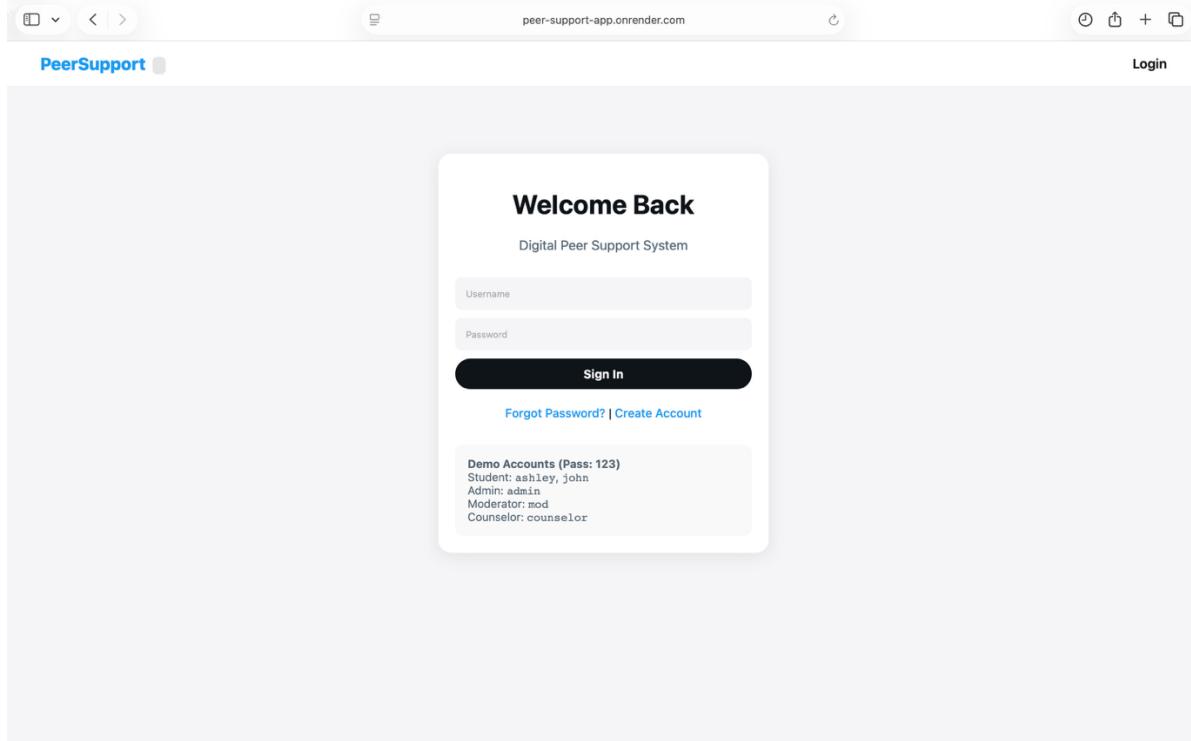
*Diagram 5.1.4.1 Counselor Subsystem Architecture Diagram*

## 6 Interface Design

### 6.1 Main Screens

The design errors we successfully avoided in our digital peer support system included, Lack of consistency, and too much memorization.

1. **Lack of consistency:** The content beneath the whole document is constructed consistently, and aligned with our digital peer support system. Every use cases stated are implemented aligned with the sequence diagrams, and state diagram, allowing the user to put the current task into a meaningful context. While the whole system including database are implemented based on the architecture diagrams, component diagrams, and deployment diagram. These **make the interface consistent**, and aligned with the documentation.
2. **Too much memorization:** To **reduce the user's memory load**, we tried to implement a user-friendly system website which discloses information in a progressive fashion. We had split the page into different pages based on the usage scenario to reduce the navigation steps for users to reach various functions of the system website.

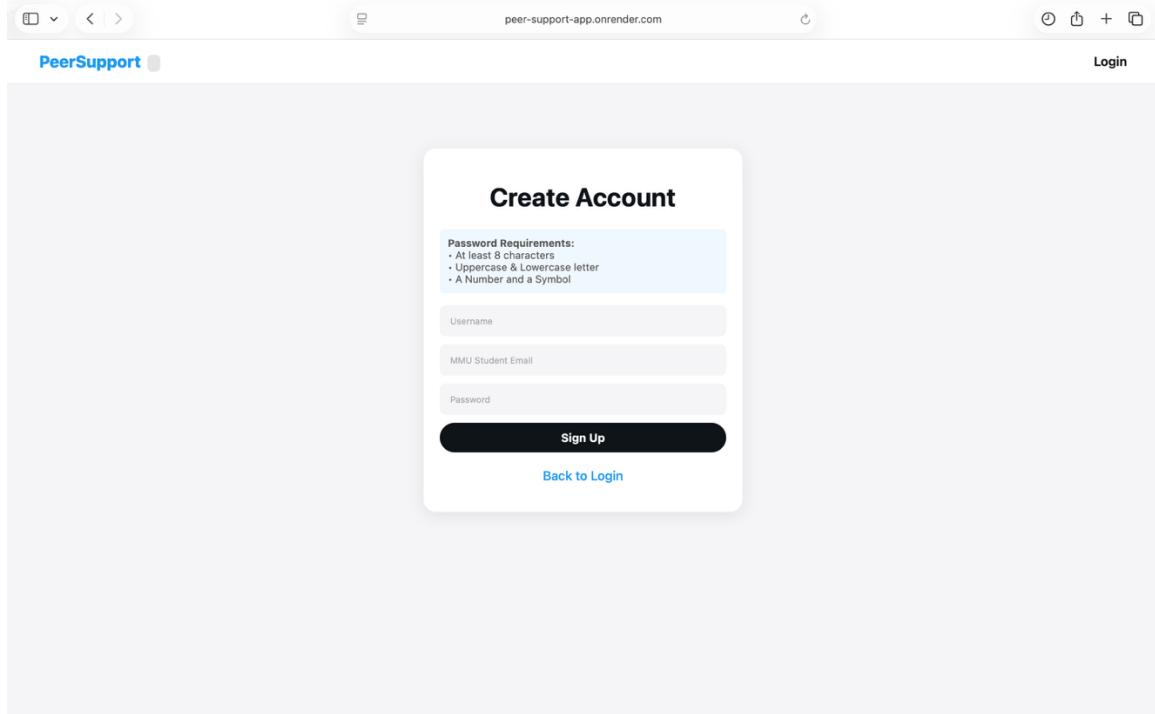


*Photo 6.1.1 System Log in Page*

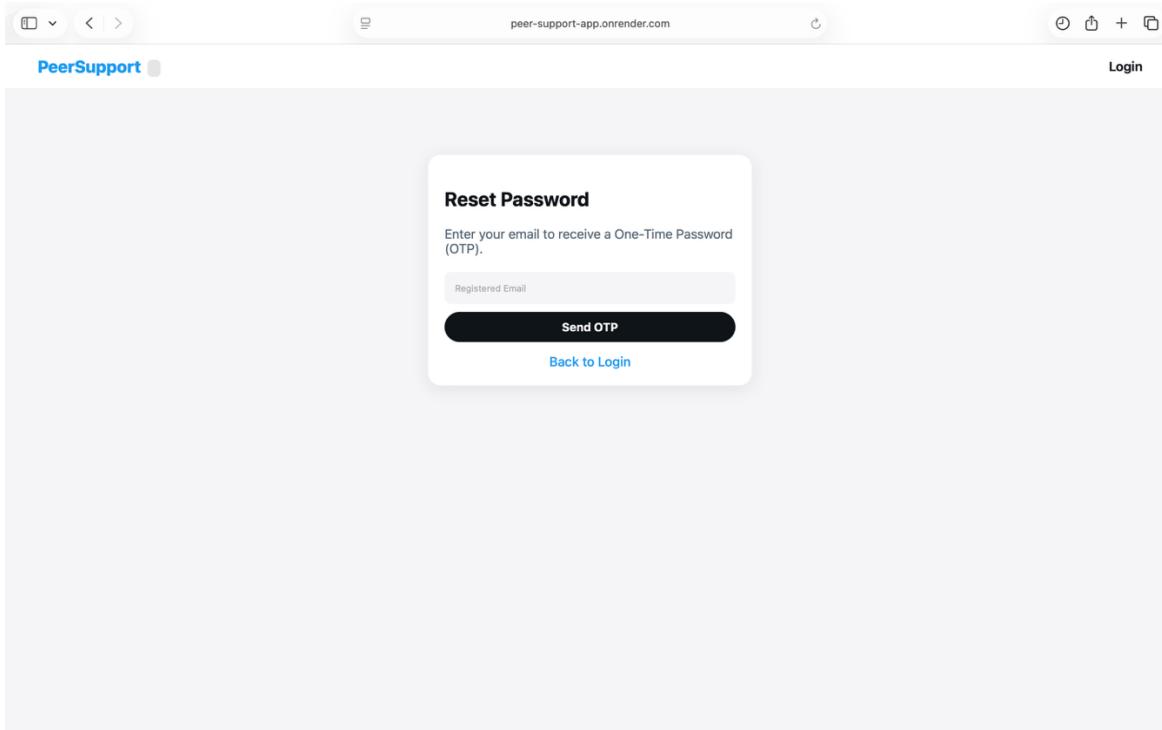
The system will validate the credentials and navigate to the panels based on the role (student/ moderator/ admin/ counselor) respectively.

## **6.2 Student Subsystem Screens**

Students are able to create accounts using their own using their MMU official email. They are able to reset their account password through the OTP sent to their email registered.



*Photo 6.2.1 Sign up a new account page*



*Photo 6.2.2 System Reset password page*

The system will navigate to the student dashboard if the user logs in as a student. The home page of the student will be the student dashboard. Students can check in their mood and well-being, checking on the announcements posted by the moderator, and check the upcoming sessions in the dashboard.

## Software Design Specification for Digital Peer Support System (Version 2.0)

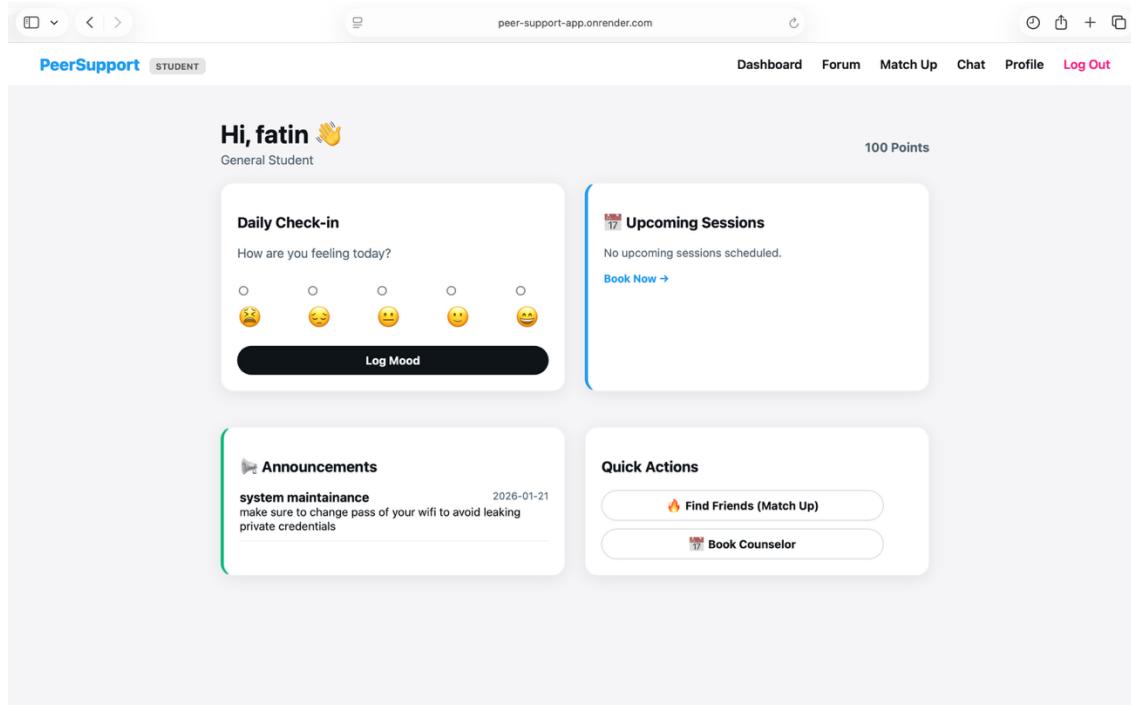
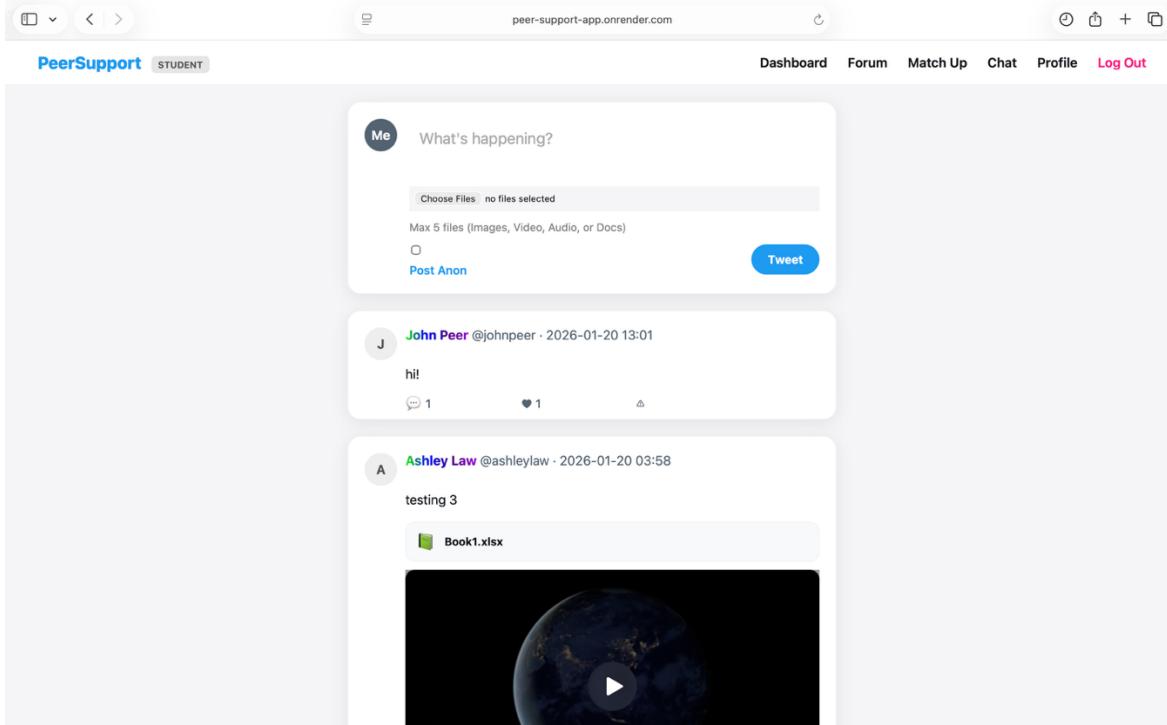


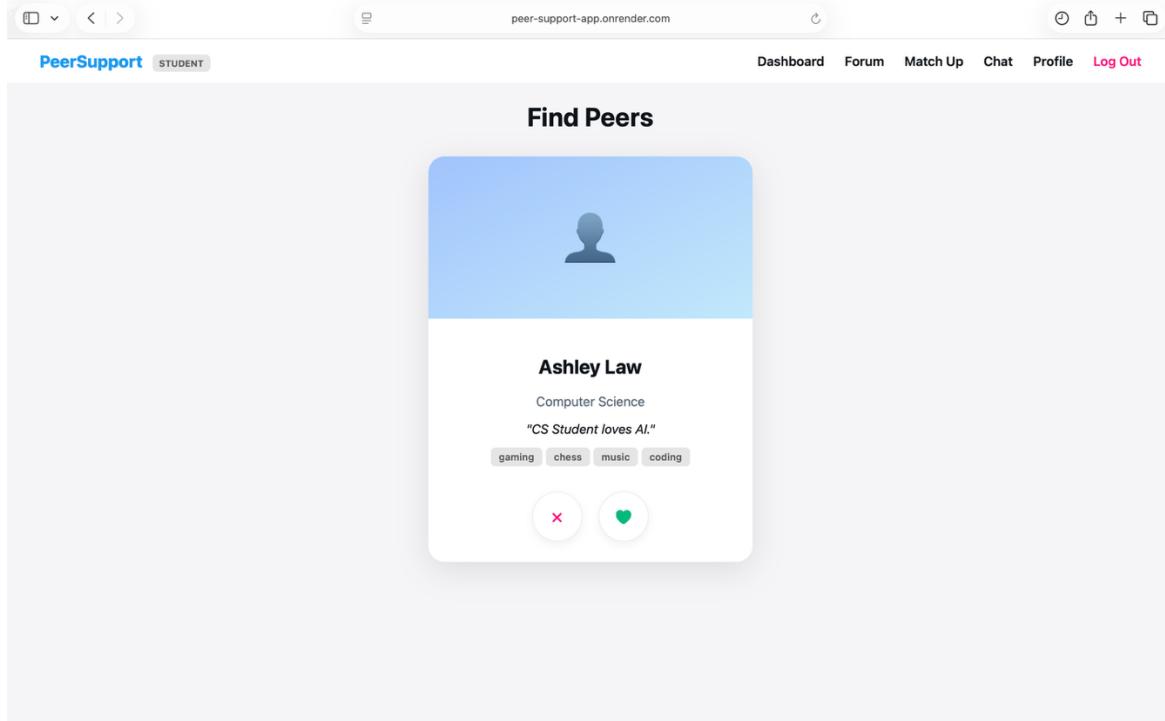
Photo 6.2.3 Student Dashboard page

Students able to share at most 5 files anonymously or non-anonymously with extensions of 'png', 'jpg', 'jpeg', 'gif', 'mp4', 'mp3', 'mov', 'mpeg', 'mpg', 'doc', 'docx', 'pptx', 'xls', and 'xlsx' only in one post.



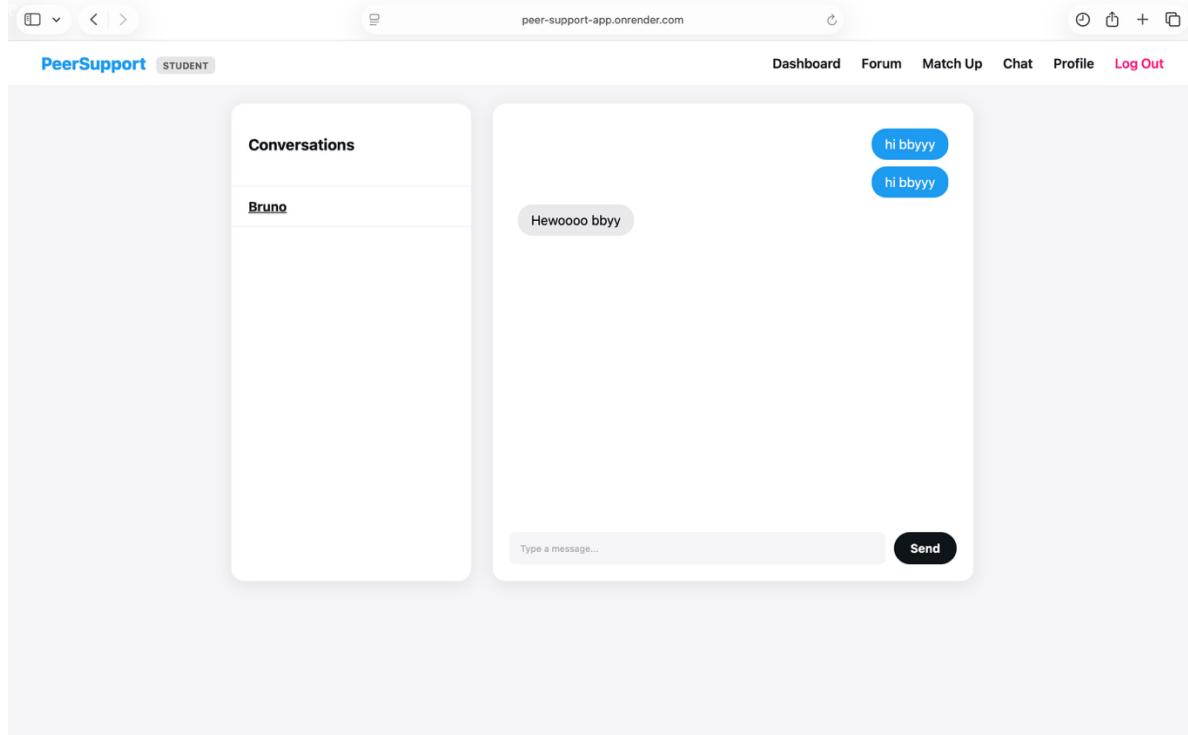
*Photo 6.2.4 Student Forum page*

Student able to match-up with others who having the same interest in Match Up page. If there's no students having the same interest, system will prompt the student to update profile.

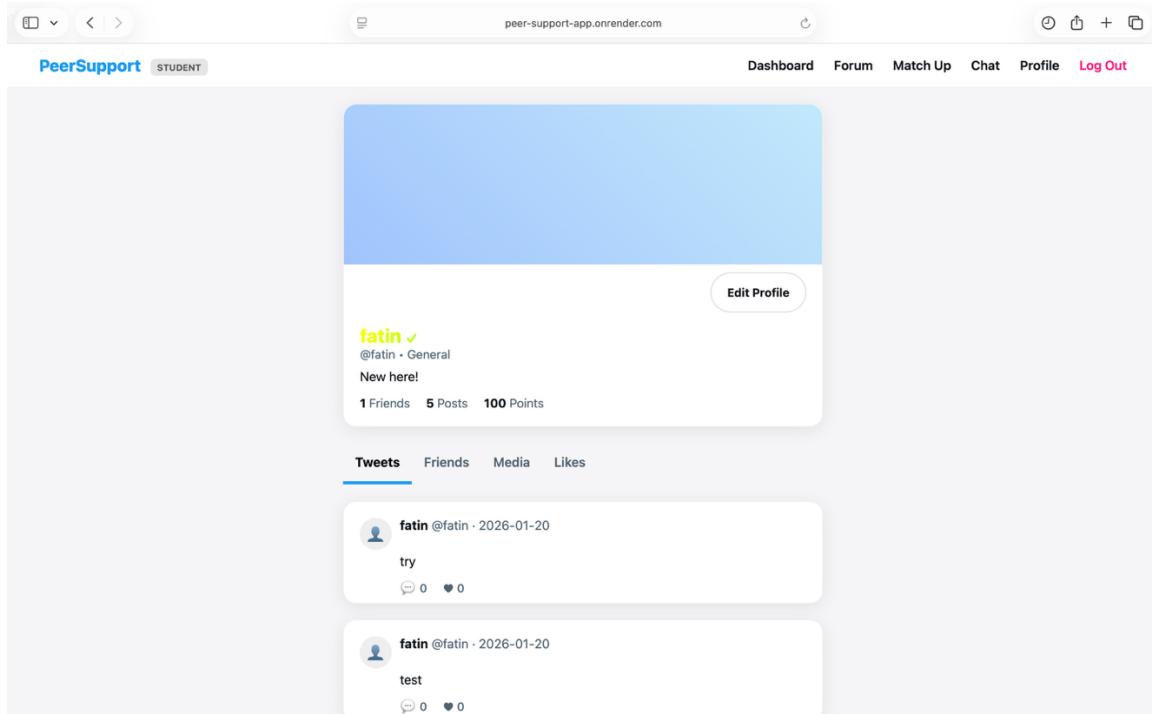


*Photo 6.2.5 Interest match-up page*

Students can only send private messages to those who are friends with themselves.

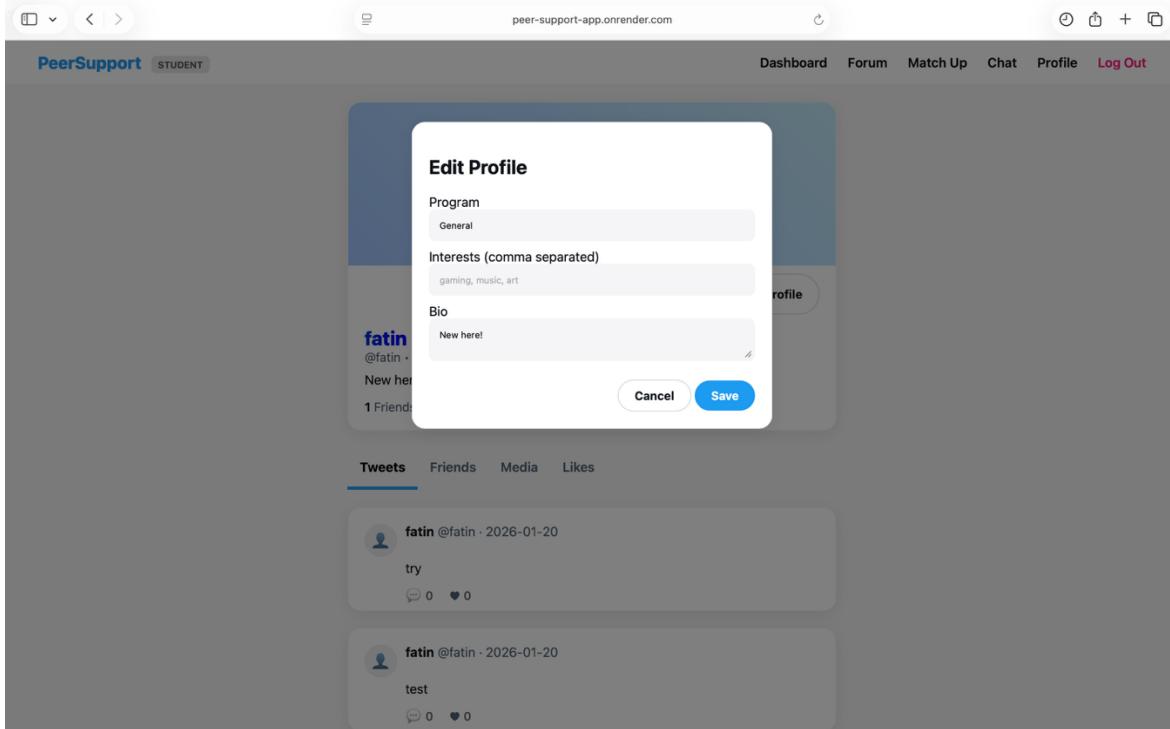


*Photo 6.2.6 Private Chat page*



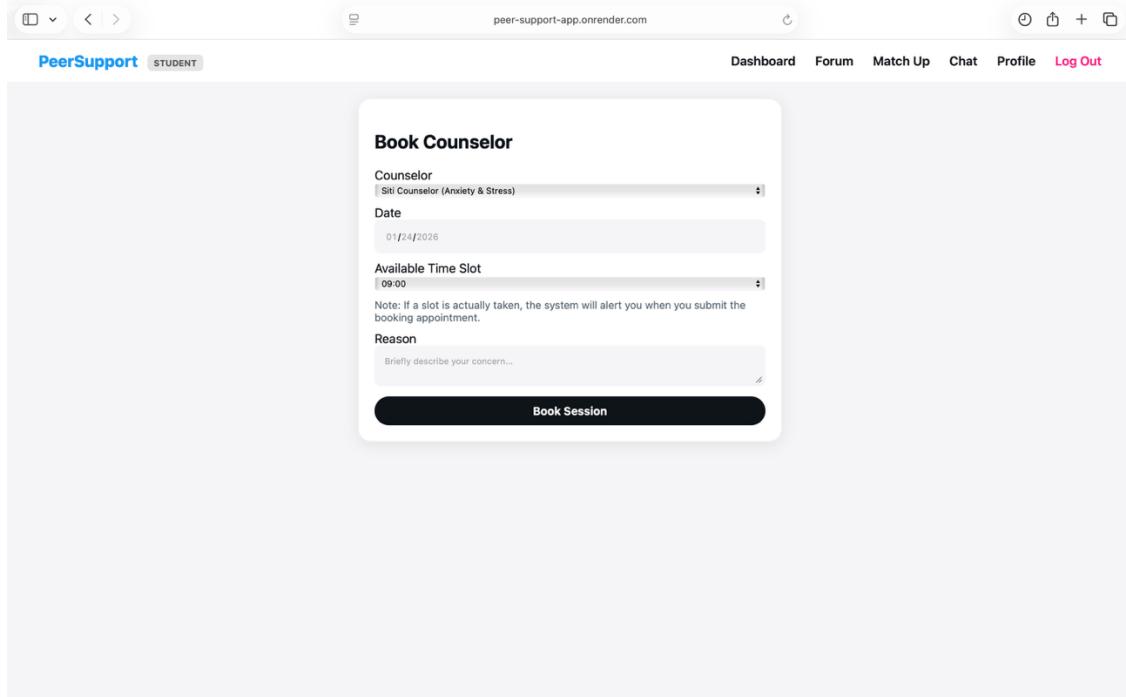
*Photo 6.2.7 Profile page*

## **Software Design Specification for Digital Peer Support System (Version 2.0)**



*Photo 6.2.8 Edit/built up profile*

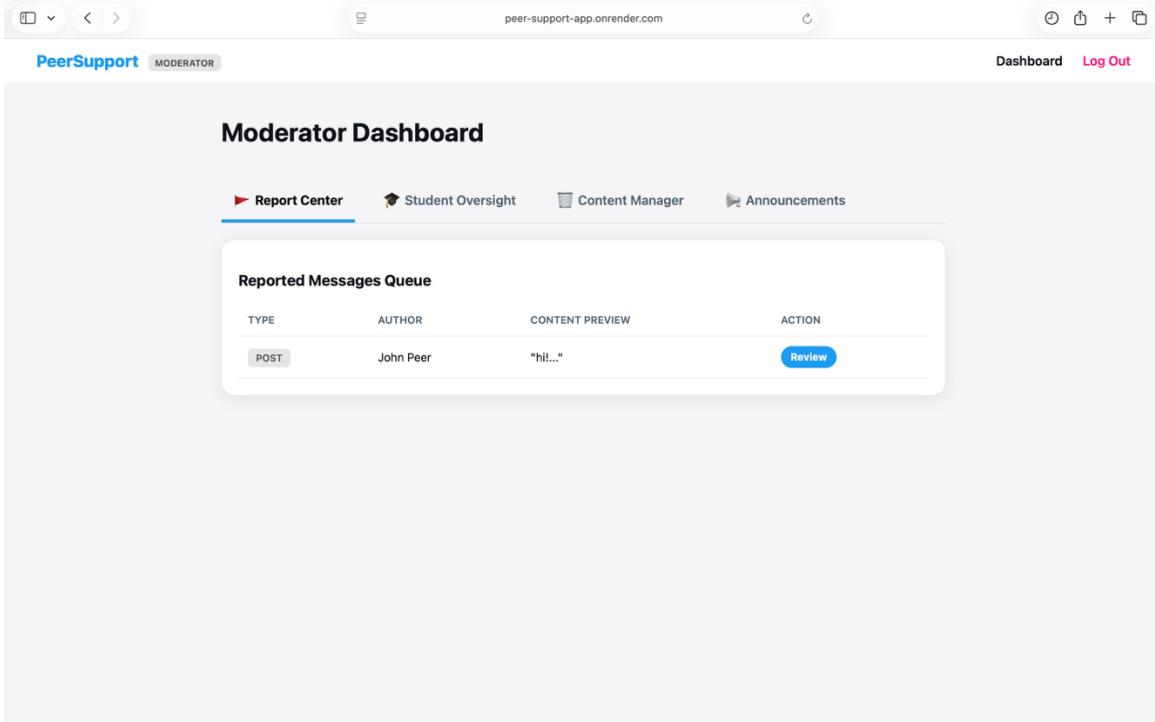
The student is also able to book an appointment slot for counselors.



*Photo 6.2.9 Counselor Appointment*

## 6.3 Moderator Subsystem Screens

The system navigates to moderator panel if the user logs in as moderator. The moderator can review the reported messages in the moderator panel.



*Photo 6.3.1 Moderator Dashboard page*

The moderator can flag student accounts that detected potential problems by the system to admin.

## **Software Design Specification for Digital Peer Support System (Version 2.0)**

The screenshot shows the 'Moderator Dashboard' with the 'Student Oversight' tab selected. A table titled 'Potential Flag List' displays student information:

STUDENT	VIOLATIONS	SCORE	ACTION
Troubled Student Arts	2	50.0%	<b>Flag</b>

*Photo 6.3.2 Moderator Flag student accounts*

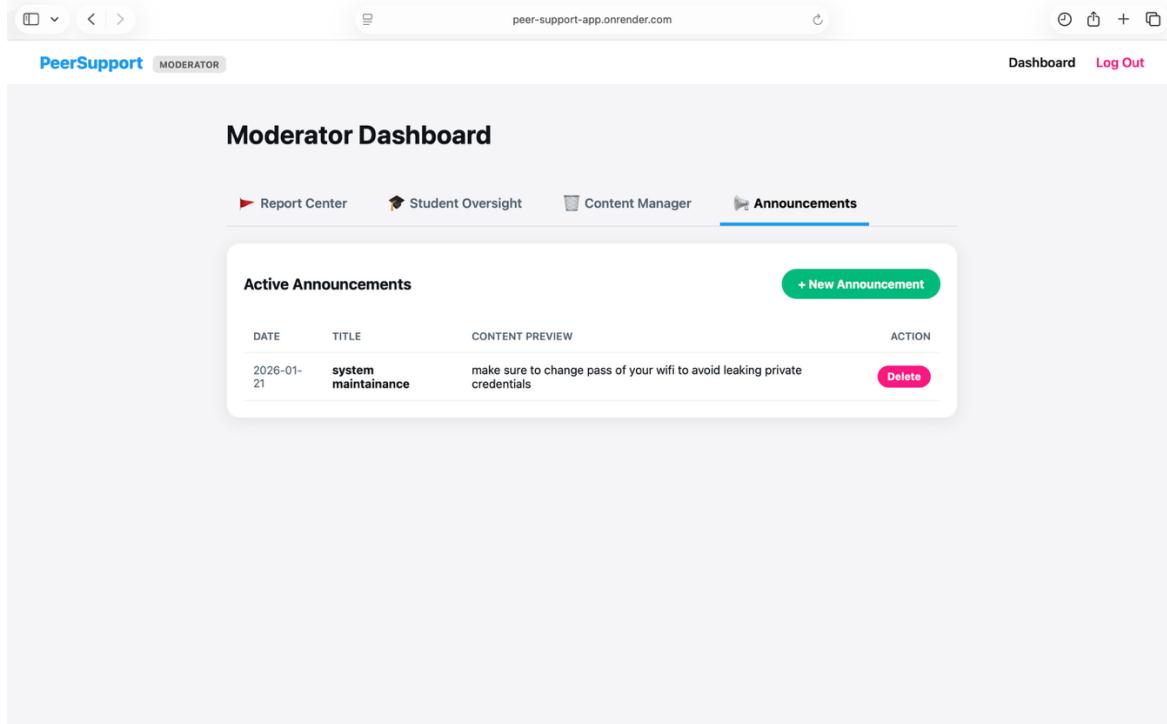
The moderator can delete the student forum posts and comments (manage forum).

The screenshot shows the 'Moderator Dashboard' with the 'Content Manager' tab selected. Two panels are visible:

- Recent Posts:**
  - John Peer: hi! Del
  - Ashley Law: testing 3 Del
  - fatin: try Del
  - Ashley Law: 22 Del
  - fatin: test Del
  - fatin: can? Del
  - fatin: MI Del
  - Ashley Law: tryinggg Del
  - fatin: hi Del
  - Ashley Law: share here :) Del
- Recent Comments:**
  - fatin: ehehhehhehee Del

*Photo 6.3.3 Moderator Forum Content Management*

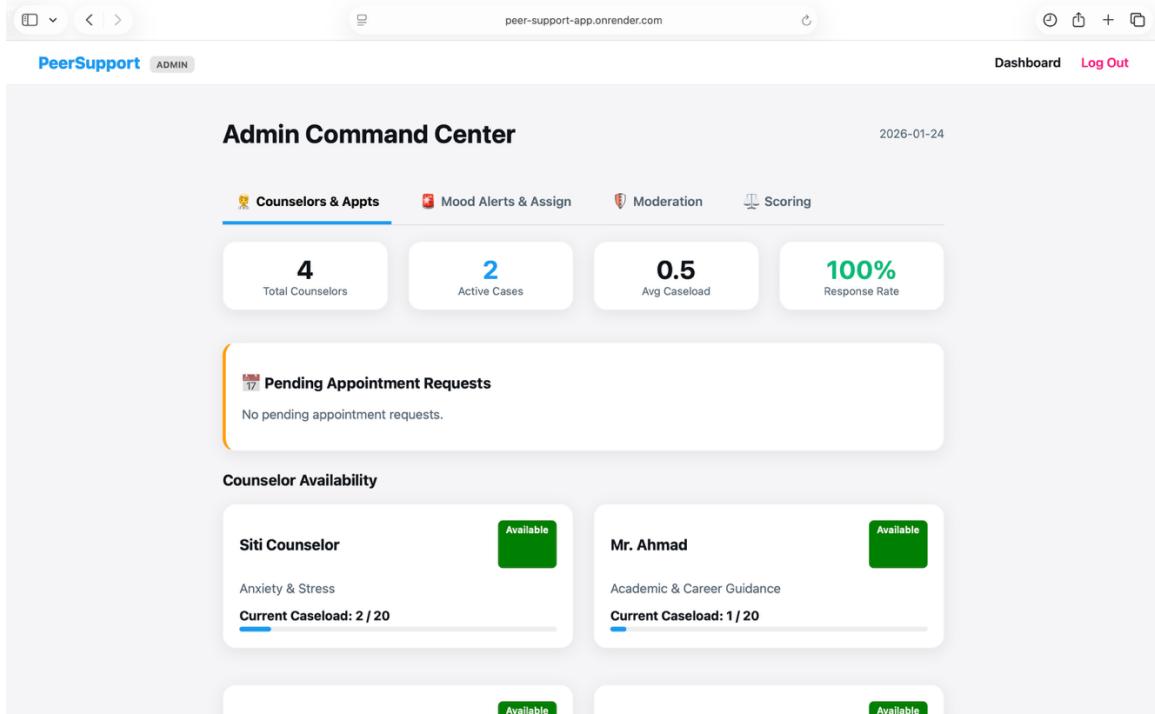
The moderator is able to make announcements to students (manage announcements).



*Photo 6.3.4 Moderator Announcement Management*

## 6.4 Admin Subsystem Screens

The system navigates to admin panel if the user logs in as admin.



*Photo 6.4.1 Counselor and Appointments Dashboard*

Admin can monitor mood alert from students who fail under the average mood threshold.

The screenshot shows the Admin Command Center interface. At the top, there are tabs for Counselors & Appts, Mood Alerts & Assign (which is currently selected), Moderation, and Scoring. The date 2026-01-24 is displayed. Below the tabs, a section titled "Students Requiring Attention (Score < 30%)" lists three students:

STUDENT DETAILS	ALERT SEVERITY	LAST ACTIVE	ASSIGNMENT ACTION
fatin Logs: 2	14% (Critical)	2026-01-22 14:02:49	✓ Assigned/Booked
John Peer Logs: 1	14% (Critical)	2026-01-22 14:09:58	✓ Assigned/Booked
kira Logs: 1	28% (Critical)	2026-01-22 14:16:21	Select Counselor: -- Choose based on capacity -- Confirm Assignment

A modal window is open for student "kira", showing a critical score of 28% and a "Select Counselor" dropdown with the placeholder "Choose based on capacity". A "Confirm Assignment" button is at the bottom of the modal.

*Photo 6.4.2 Mood Alerts Dashboard*

The admin can review the reported content, suspend the flagged student accounts and from Moderators, and process appeals.

The screenshot shows the Admin Command Center interface. At the top, there are navigation links for Counselors & Appts, Mood Alerts & Assign, and Moderation (which is currently selected). Below the navigation is a timestamp of 2026-01-24.

**Moderation Queue:** A red box highlights a recent moderation action. The details are as follows:

- MODERATOR VERIFIED VIOLATION**
- CRITICAL Priority**
- POST**
- Reported User:** John Peer
- Violation Type:** Inappropriate
- Reason:** "hi!"
- Date:** 2026-01-24 03:48:18
- Review & Act** button

**Flagged Students:** A table showing one flagged student:

STUDENT	SCORE	REASON	ACTION
Troubled Student	50.0	no reason	<b>Suspend</b>

*Photo 6.4.3 Content Moderator Management*

The admin is able to set the scoring points for the gamification.

The screenshot shows the Admin Command Center interface with the Scoring tab selected. The timestamp is 2026-01-24.

**Scoring Configuration:** This section contains two main tables: Positive Contributions (+) and Violations & Penalties (-).

**Positive Contributions (+):**

Activity	Score
Post Creation	5
Helpful Answer	10
Peer Support	15

**Violations & Penalties (-):**

Violation	Score
Content Removal	15
Harassment	30
Severe Violation	50

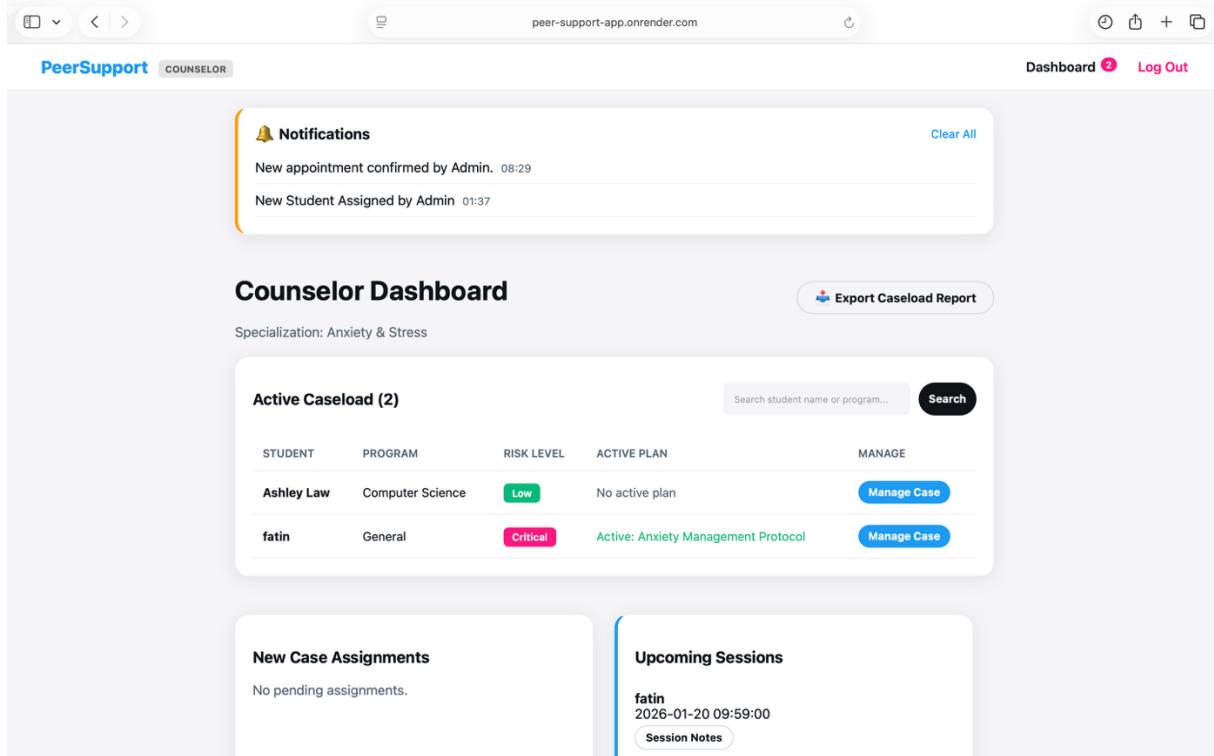
**Restricted Users (< 60%):** A table showing a user who has violated the system.

USER	SCORE	ACTION
Troubled Student Violations: 2	60.0%	<b>Suspend</b>

*Photo 6.4.4 Scoring System*

## 6.5 Counselor Subsystem Screens

The system will navigate to counselor panel if the user logs in as counselor. The counselor is able to manage caseload, export data, verify assignments from admin, manage action plans, and schedule the appointments.



*Photo 6.5.1 Counselor Dashboard*

This image shows a counselor's dashboard for a student named *fatin*, designed to support therapy management. At the top, notifications alert the counselor to updates like appointments and new assignments. Below, the student's profile displays basic information for personalized care. The counselor can create and track Therapeutic Action Plans to structured goals such as reducing panic attacks used during sessions and monitor the student's well-being through a Mood History Tracker that logs emotional check-ins to spot trends and guide treatment.

## Software Design Specification for Digital Peer Support System (Version 2.0)

The screenshot shows the Counselor Dashboard interface for the PeerSupport app. At the top, there are navigation icons and the URL 'peer-support-app.onrender.com'. The top right includes 'Dashboard' with a question mark icon, 'Log Out', and other user icons.

**Notifications:** A sidebar on the left lists two notifications: 'New appointment confirmed by Admin. 08:29' and 'New Student Assigned by Admin 01:37'. There is a 'Clear All' button at the top right of this sidebar.

**Case File: Ashley Law:** This section contains:

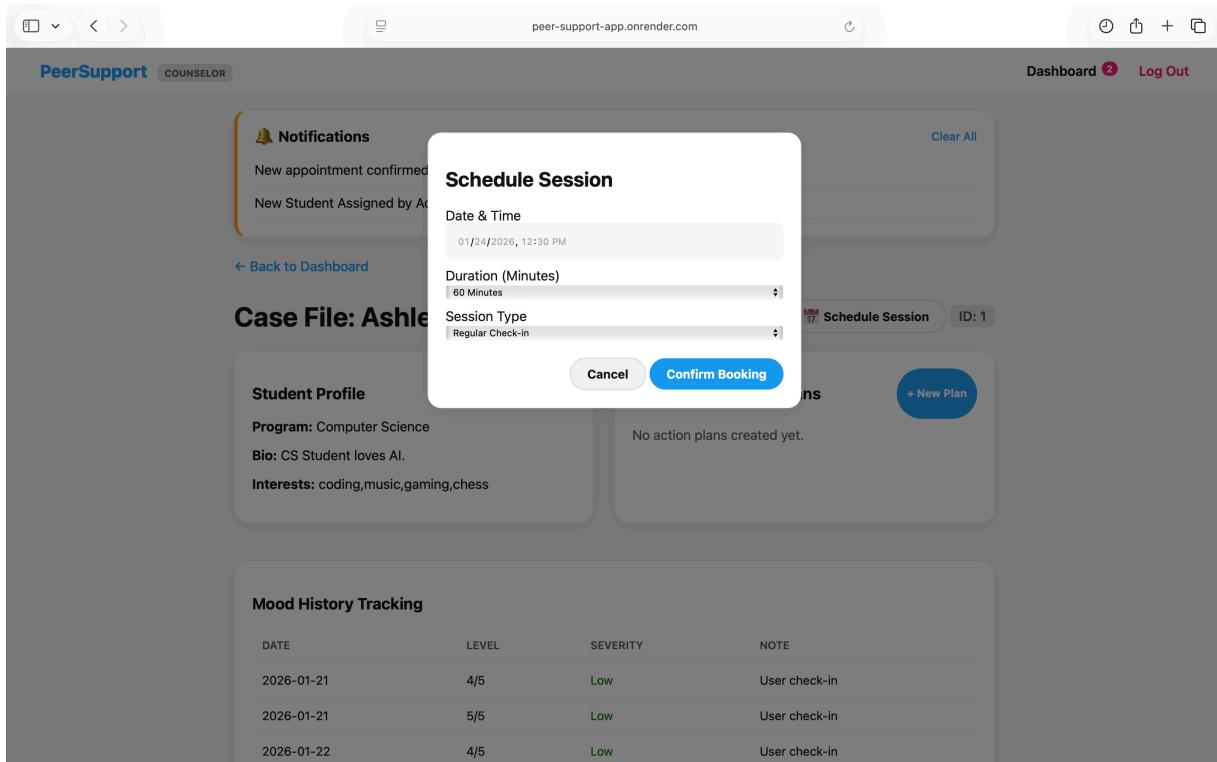
- Student Profile:** Shows 'Program: Computer Science', 'Bio: CS Student loves AI.', and 'Interests: coding,music,gaming,chess'.
- Therapeutic Action Plans:** Shows a message 'No action plans created yet.' and a blue '+ New Plan' button.

**Mood History Tracking:** A table showing mood history for three dates:

DATE	LEVEL	SEVERITY	NOTE
2026-01-21	4/5	Low	User check-in
2026-01-21	5/5	Low	User check-in
2026-01-22	4/5	Low	User check-in

Photo 6.5.2 Counselor Dashboard

This image shows the session scheduling interface on a student's case file page, allowing a counselor to reschedule or book a new therapy session with the student. The counselor can select the date and time, choose a duration, for example, 60 minutes, and specify the session type such as a regular check-in before confirming the booking. This interface helps the counselor efficiently manage appointments while keeping track of scheduled sessions and their statuses.

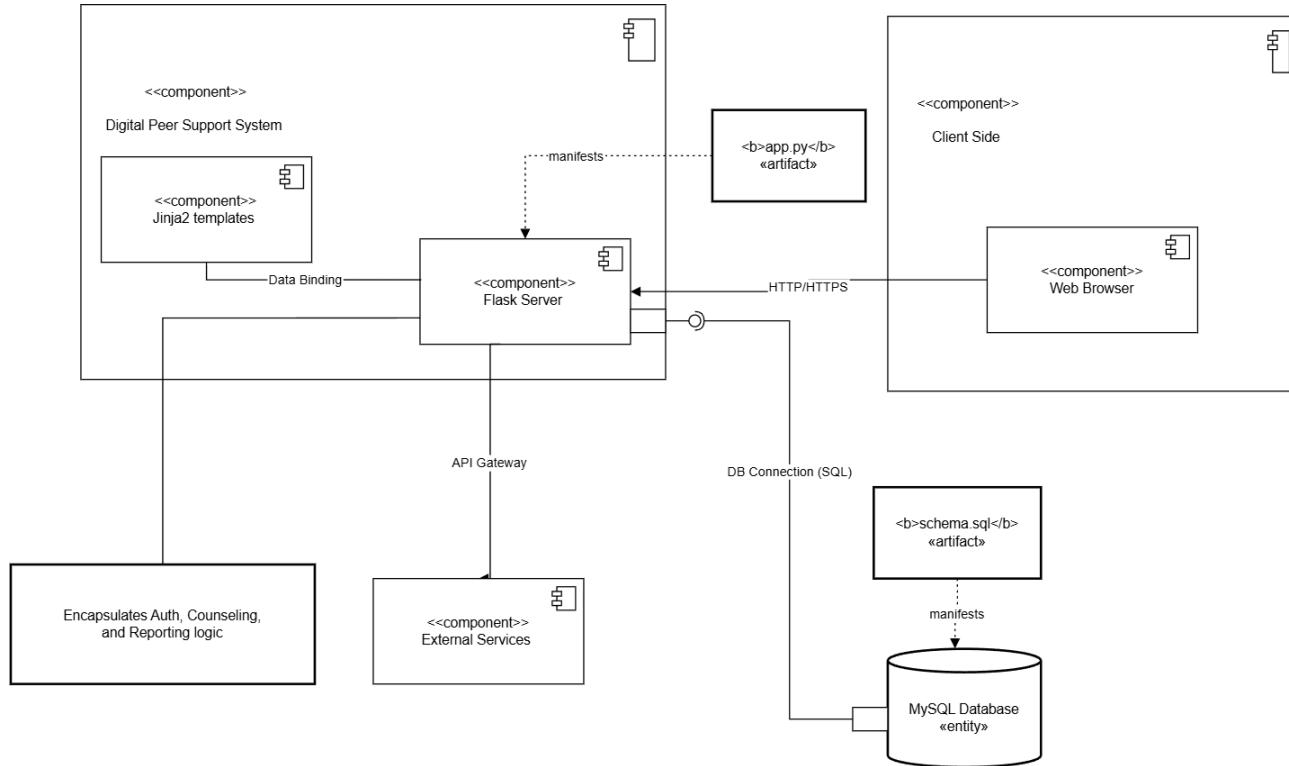


*Photo 6.5.3 Counselor Manage Caseload*

## 7 Component Design

### 7.1 Main Components

The Digital Peer Support System is the main container that organizes internal parts like the Flask Server and Jinja2 Templates. It ensures that the system is modular and that important tasks like authentication are kept separate.



*Diagram 7.1.1 Component Diagram*

### 7.1.1 Flask Server (Controller)

The Flask Server acts as the controller component within the system, responsible for handling the primary business logic and routing. It processes incoming HTTP/HTTPS requests from the client side, executes core logic for counselor and student use cases, and communicates with external services via an API Gateway. This component is physically manifested by the `app.py` artifact.

### 7.1.2 Jinja2 Templates (Boundary)

The Jinja2 Templates represent the boundary component of the system, facilitating the interaction between the application logic and the user's browser. It is responsible for Data Binding, which takes data processed by the Flask Server and injects it into HTML structures to be rendered as dynamic web pages for the client.

### 7.1.3 Web Browser (User Interface)

The Web Browser is a client-side component that serves as the primary User Interface for students and counselors. It resides within the "Client Side" boundary and interacts with the main system by sending HTTP/HTTPS requests and receiving rendered responses, allowing users to navigate dashboard views and perform actions like submitting reports.

### 7.1.4 MySQL Database (Entity)

The MySQL Database is an entity component that provides persistent storage for all system data, including user records and caseload information. It interacts with the Flask Server through a DB Connection (SQL), responding to queries such as `query_db` and `execute_db`. This component is physically manifested by the `schema.sql` artifact.

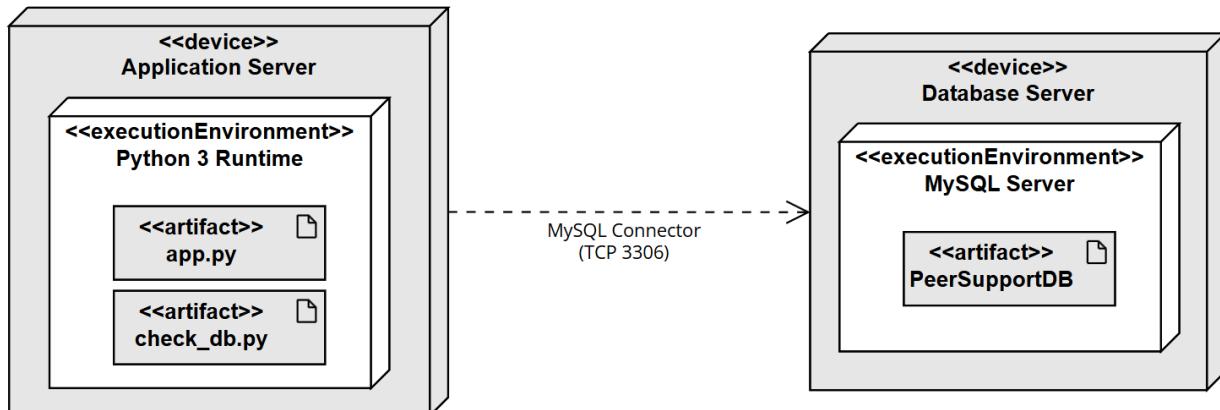
### 7.1.5 External Services

The External Services component encapsulates third-party integrations required for specialized functionalities outside the core Flask application. This includes Cloudinary for managing image uploads and the SMTP Server for sending automated email notifications to users, both of which are accessed by the controller through defined service protocols.

## 8 Deployment Design

### 8.1 Deployment Diagram

The deployment diagrams show the computing environment of our digital peer support system.



*Diagram 8.1.1 Deployment Diagram*

## 9 Summary

**Digital Peer Support System** version 2.0 serves as a complete web-based educational platform which **Multimedia University (MMU)** uses for its educational needs. The system establishes a connection between social contact and professional mental health treatment by enabling users to connect with each other while receiving qualified counseling services.

The system uses Layered Architecture design which divides application into Presentation Business Logic, Data Access, and Data components. The system implements this method to achieve three objectives which include **maintaining system security** and enabling **easier system maintenance and development work**. The system operates through four essential functional subsystems which include **Student** and **Moderator** and **Counselor** and **Admin** while it uses Role-Based Access Control (RBAC) as its main security method to protect data and maintain system operations.

### **Key Technical Components**

- **Data Design:** The system uses MySQL to create a strong relational database schema which handles multiple user authentication and mood tracking and appointment scheduling and gamification logic (points and rewards) relationships.
- **Behavioral Modeling:** The project uses sequence and state diagrams to create detailed maps which show all essential workflows that range from account creation to private chatting and crisis intervention (mood alerts) and administrative oversight (suspensions/appeals).
- **Interface Design:** The UI design delivers a consistent user experience which enables students to find support features easily while staff members (Counselors/Moderators) maintain control of their caseload and content through effective dashboards.
- **Readiness for Implementation:** The Software Design Specification encompasses all design details which developers need to start the development process. The document defines all data structures together with sequenced API-ready logic and specifications which detail how components should be deployed on Web

## **References**

1. GeeksforGeeks. (n.d.). Use case diagram.  
<https://www.geeksforgeeks.org/system-design/use-case-diagram/>
2. Visual Paradigm. (n.d.). What is use case specification?  
<https://www.visual-paradigm.com/guide/use-case/what-is-use-case-specification/>
3. New York University, Courant Institute of Mathematical Sciences. (2018). Use case template.  
[https://cs.nyu.edu/~jcf/classes/CSCI-GA.2440-001\\_sp18/handouts/UseCaseTemplate.pdf](https://cs.nyu.edu/~jcf/classes/CSCI-GA.2440-001_sp18/handouts/UseCaseTemplate.pdf)
4. QBI Solutions. (n.d.). Use case template.  
[https://www.qbi.in/uploads/5/5/9/0/55906279/use\\_case\\_template-1.pdf](https://www.qbi.in/uploads/5/5/9/0/55906279/use_case_template-1.pdf)
5. Visual Paradigm. (n.d.). What is a class diagram?  
<https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-class-diagram/>
6. EdrawSoft. (n.d.). Architecture diagram (system architecture).  
<https://www.edrawsoft.com/architecture-diagram.html#system>
7. GeeksforGeeks. (n.d.). How to draw architecture diagrams.

<https://www.geeksforgeeks.org/system-design/how-to-draw-architecture-diagrams/>

8. Visual Paradigm. (n.d.). What is a deployment diagram?

<https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-deployment-diagram/>

9. Lucidchart. (n.d.). UML deployment diagram tutorial.

<https://www.lucidchart.com/pages/uml-deployment-diagram>

10. Lucidchart. (n.d.). Component diagram tutorial.

<https://www.lucidchart.com/pages/component-diagram-tutorial>

11. Visual Paradigm. (n.d.). What is a component diagram?

<https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-component-diagram/>

12. Visual Paradigm Online. (n.d.). Sequence diagram tutorial.

<https://online.visual-paradigm.com/diagrams/tutorials/sequence-diagram-tutorial/>

13. Visual Paradigm. (n.d.). Sequence diagram.

<https://www.visual-paradigm.com/learning/handbooks/software-design-handbook/sequence-diagram.jsp>

14. Visual Paradigm. (n.d.). What is a state machine diagram?

<https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-state-machine-diagram/>

15. Lucidchart. (n.d.). UML state machine diagram tutorial.

<https://www.lucidchart.com/pages/tutorial/uml-state-machine-diagram>