Software Requirements Specification

for

Digital Peer Support System

Version <1.0>

Group No.: 8

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

Date: 14 December 2025

Contents

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
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 Project Introduction

According to short interviews with a total amount of 30 students from various faculties in Multimedia University, we found that students in Multimedia University often face academic pressure, emotional stress, and personal challenges. However, many of them are hesitant to seek help because of stigma, fear of judgment, or a lack of awareness about available support services.

In Addition, the current support resources and student information are usually scattered across various platforms. The students interviewed pointed out that official departmental announcements can be obtained through the MMU online portal, while announcements from other departments (such as the SRC, SCC and etc.) are posted on social media platforms such as Instagram or Facebook. This can cause anxiety among students due to a lack of access to the latest announcements and reduce student engagement.

The Digital Peer Support System aims to provide a safe and accessible online platform where students can seek peer support, share experiences, find relevant information or get latest announcement, and enable relevant departments to track their well-being timely.

By combining peer interaction, mood tracking, and structured support pathways into one system, the platform is able to address students' reluctance to seek help and reduce the fragmentation in announcements from various departments in MMU. This system encourages students' engagement, promotes a platform for healthy communication, and improves access to educational and well-being resources within the student community.

1.1 Team Members

Name	Actor/Processes
Fatin Shamirah Kamal	Admin
Ashley Law Jia Ye	Student (Peer)
Rayyan Syahmi Bin Sukri Efendi	Moderator
Siti Aisyah Binti Omar	Counselor

1.2 Problem statement

Various obstacles hinder the access to proper mental health and emotional support among students in educational institutions. The current supporting infrastructure is associated with several critical limitations that deter timely assistance for the students.

Reluctance of Student to Seek Help: Most students do not seek help because they fear being judged, embarrassed, or feeling insecure. Students often refrain from reaching out when anonymous support alternatives are unavailable, making many students turn internal problems into significant current mental health crises.

Lack of a unified digital platform for students to get information: The existing support system is highly fragmented, since there is no single digital platform that integrates support services, announcements, peer groups, and mental health resources. Students navigate this across different disconnected channels, which increases confusion and reduces accessibility. At the same time, without a proper centralized system to link up counselors, peer mentors, and students, the support network is often weakened in the identification of at-risk populations, coordination among support workers is ineffective, and missed opportunities for timely intervention or appropriate peer matching become common. Such fragmentation and disconnection could eventually discourage students from approaching someone for help when they need it most.

Each of these challenges in turn contributes to a cycle of delayed intervention, increasing student isolation, underutilization of existing support resources, and overall worsening mental health status. There is an urgent need for a comprehensive Digital Peer Support System that provides a unified, accessible platform where students can connect with counselors and peers, access resources anonymously, and receive proactive support based on their wellbeing status.

1.3 Project Schedule

The project will follow the Agile Development Model. It is organized into five phases, which involved with Both Software Requirements Specification (SRS), System Design (SD), and Software Engineering Fundamentals (SEF).

Phase 1: Plan

This phase focuses on project planning/ requirement analysis:

- Requirements gathering
- Defining user roles (User, Admin, Moderator, Counselor), user stories and use cases
- Defining the functional/non-functional requirements
- Conducting System Models, including Entity Relationship Diagram (ERD)

Phase 2: Design

This phase is the core of structured design (SD) of the project:

- Designing system architecture
- Data Design (Data Dictionary)
- Behavioral Modeling
- Prototype Implementation, Component Design, and Interface Design
- Designing the software architecture

Phase 3: Develop & Test

This phase focuses on system development and system testing:

- Set up the technical environment
- Code implementation & unit testing
- Code integration
- System testing

Phase 4: Deploy & Review

This phase makes sure that all team members are on the right track and progressive during the project implementation:

- Weekly meetings will be conducted for phase 1 (project planning & requirement analysis) and phase 2 (structured design) is for brainstorming, and to confirm that all team members have reached consensus on the details of the project.
- While on phase 3, each team member will present the current progression of assigned tasks for the implementation during the weekly meetings. Any issues may be raised for discussion, and suggestions for improving the system may be proposed.
- One meeting will be held during phase 5 to distribute the tasks for the project report.

Phase 5: Write-up & Documentation

This phase aims to make the final deployment of the project:

- Project Report & Documentation
- Presentation Preparation
- Presentation Rehearsal

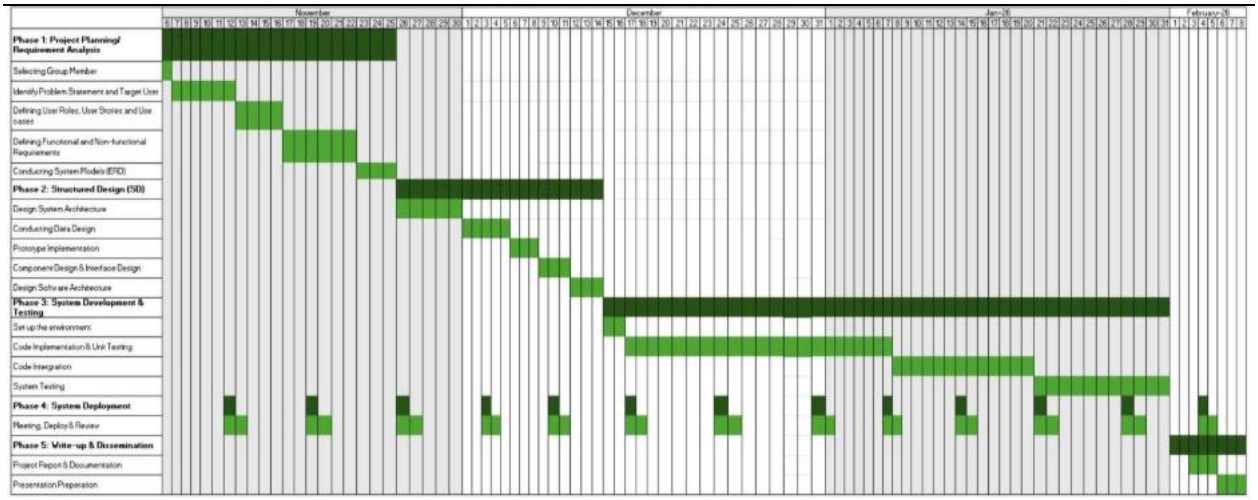

Diagram 1.3.1 T13L G8 Gantt Chart

2 System Overview

2.1 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 & Level Progression)**

- Actors: Student
- To encourage healthy participation, students earn points for positive engagement (posting, commenting, and checking-in their moods). Accumulated points increase their level, promoting consistent involvement.
- Key Functions: Earn points to Level Up through gamification

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.

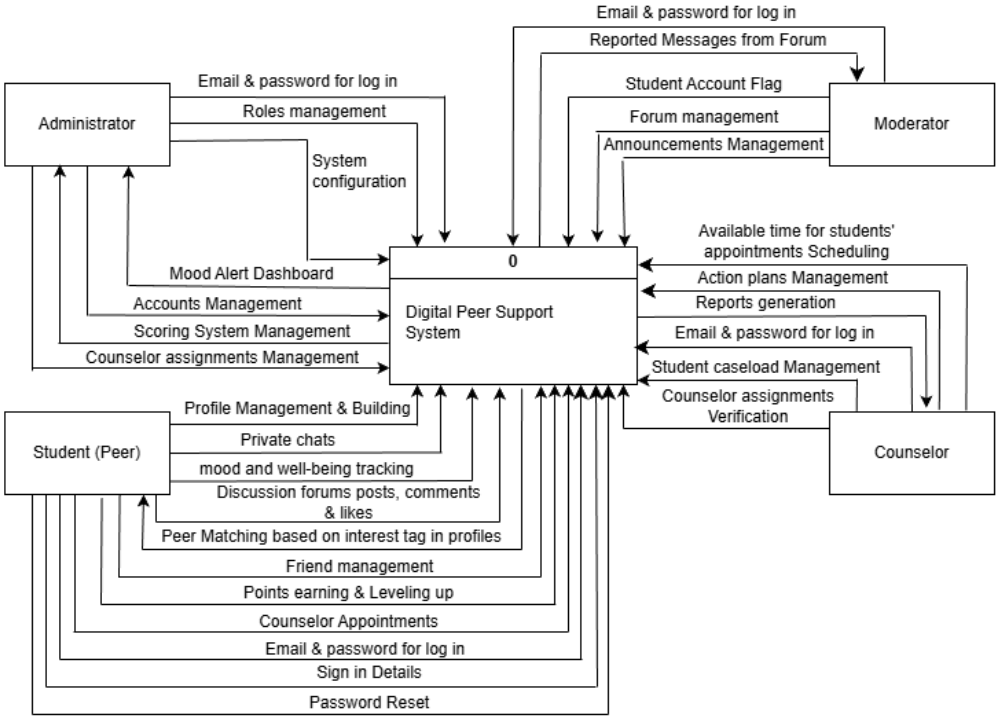

Diagram 2.1.1 Top Level Data Flow Diagram

2.2 User Story

2.2.1 Admin

1. As an admin, I want to log in with my own account so that I able to my work as an admin.
2. As an admin, I want to log out of my own account so that I can keep my information secure.
3. As an admin, I want to monitor mood alert dashboard in real-time with severity levels and trend patterns, so that I can prioritize urgent mental health interventions and assign counselors efficiently.
4. As an admin, I want to manage counselor assignment so that students receive timely and appropriate mental health support.

5. As an admin, I want to review reported content with full context and violation history, so that I can make fair moderation decisions and apply appropriate penalties consistently.
6. As an admin, I want to manage scoring system so that the scoring system encourages good behavior and fairly restricts posting privileges.
7. As an admin, I want to process score appeals from restricted users by evaluating their behavioral history, so that I can restore privileges for those demonstrating genuine improvement while maintaining system integrity.
8. As an admin, I want to temporarily or permanently suspend accounts for violations with documented reasons, so that harmful users are removed while maintaining a clear appeal process.

2.2.2 Student

1. As a student, I want to sign up for a new account so that I can access the system securely and keep my information private.
2. As a student, I want to log in with my own account so that I able to view my past posts and comments in the forum.
3. As a student, I want to reset my password so that I can sign in to my own account again even if I forget my password.
4. As a student, I want to log out with my own account so that I can keep my information secure.
5. As a student, I want to report irrational or inappropriate contents in the forum to moderators so that if users miss use the platform, moderators can be notified.
6. As a student, I want to build my profile with interest labels and a short introduction so that others can understand me better.
7. As a student, I want to check-in my mood and well-being every day so that I able to seek timely assistance from Counselor.
8. As a student, I want to add friends so that I can build new connections and keep track with their non-anonymously posts in the forum.
9. As a student, I want to chat privately so that I can stay connected with others.
10. As a student, I want to make posts in the forum anonymously or non-anonymously so that I can seek help or gain support in a way that feels comfortable to me.
11. As a student, I want to comment the posts on the forum with the option to show my name or stay anonymous so that I can share freely.
12. As a student, I want to like the posts and comments in the forum so that I can communicate my opinion (show agreement) without needing to write a full comment.
13. As a student, I want to earn points to level up when my own posts or comments reach a target number of likes so that I feel encouraged to participate in the forum.
14. As a student, I want to match-up with people who having similar interests such as having the same state, hobby, study style, or programme taken so that I can connect with like-minded peers.
15. As a student, I want to book counselor appointments myself, so I can reach out proactively.

2.2.3 Moderator

1. As a moderator, I need to be able to log into my account to access features on the system website.
2. As a moderator, I need to be able to log out of my account so that after i am done with my session to make sure nobody else can access my account if my device is left unattended.
3. As a moderator, I want to be able to review messages reported by students to determine if they break website guidelines.
4. As a moderator, I want to able to flag student accounts so that admins can ban or suspend the accounts
5. As a moderator, I want to able to make announcements that all students will get the latest information through the real-time announcements.
6. As a moderator, I want to able to delete forum posts made by students if they break the system website guidelines.
7. As a moderator, I want to able to delete comments under forums posts if they break the user guidelines.

2.2.4 Counselor

1. As a counselor, I need to quickly and securely log in to the system with my unique credentials so that I can access my confidential student caseload while maintaining HIPAA compliance and data protection standards.
2. As a counselor, I want to efficiently organize, search, and update my assigned student roster in one centralized view so that I can effectively monitor, prioritize, and track support for every student in my care.
3. As a counselor, I need to export structured, anonymized datasets in standard formats (CSV, Excel) so that I can conduct deeper analysis in external tools, share aggregate findings with stakeholders, and meet institutional reporting requirements.
4. As a counselor, I need to review and accept (or request adjustments to) new student assignments from administrators so that I can manage my caseload capacity effectively and ensure I can provide quality support to each assigned student.
5. As a counselor, I want to collaboratively develop, update, and monitor personalized therapeutic action plans with measurable goals and strategies so that students have a clear roadmap for growth, and we can systematically track progress toward objectives.
6. As a counselor, I need an integrated scheduling system that shows my availability, allows students to book sessions, and sends automated reminders so that I can minimize administrative coordination time and maximize direct counseling hours.
7. As a counselor, I need to securely log out of the system when finished so that I can protect sensitive student information and prevent unauthorized access to confidential clinical records between sessions.

2.3 Actors

Actor	Use Cases
Admin	• Log in account

	• Log out account • Monitor Mood Alert Dashboard • Manage Counselor Assignments • Review Reported Content • Manage Scoring System • Process Appeals • Suspend Accounts
Student	• Sign up a new account • Log in with own account • Log out with own account • Reset password • Report contents • Build profile • Check-in mood and well-being • Add friends • Chat privately • Make posts • Comment on posts • Like posts • Earn points • Match-up with people • Book counselor appointments
Moderator	• Log in account • Log out account • Review Reported Messages • Flag Student Account • Make Announcements • Delete Forum Posts • Delete Forum Comments
Counselor	• Log in own account • Manage caseload • Export data • Verify assignments • Manage action plans • Schedule appointments • Log out own account

2.4 Use Case Diagram

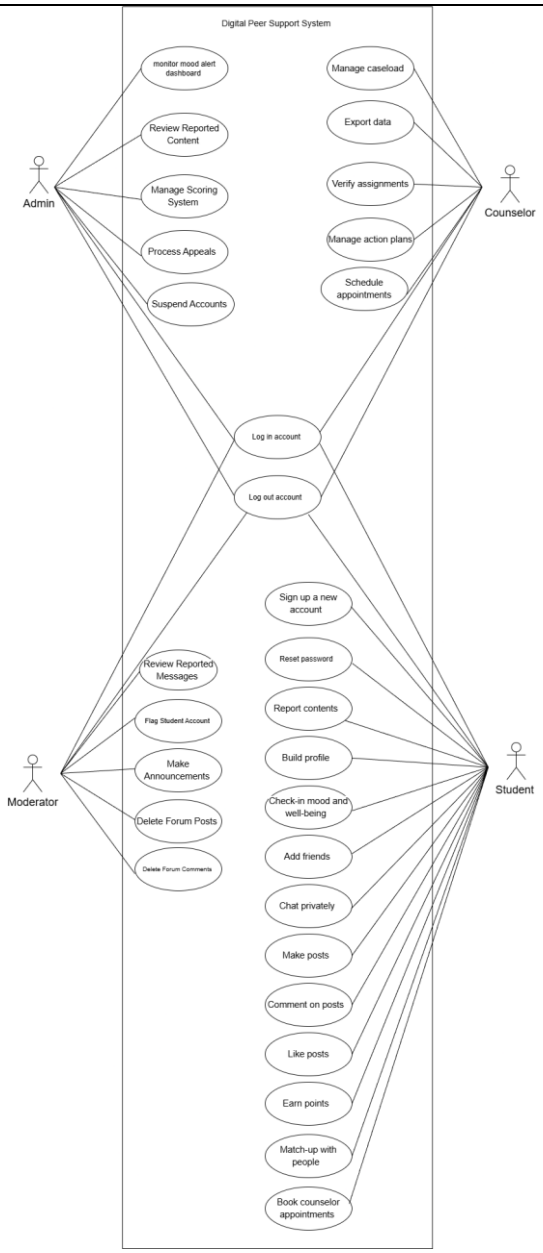

Diagram 2.4 Use Case Diagram

3 Functional Requirements

3.1 Admin

3.1.1 Log in account

The admin is able to enter their login credentials to access the admin dashboard and management features of the digital peer support platform.

Use Case ID	UC1
Use Case Name	Log in Account
Primary Actor	Admin
***Pre-Conditions**	1. Admin account has been created in the system 2. Admin has valid username and password
Post-Conditions	Admin is logged onto the system
Basic Flow	1. Admin opens the platform website in web browser 2. System displays login page 3. Admin clicks "Admin Login" button 4. System shows admin login form with username and password fields 5. Admin enters username in username field 6. Admin enters password in password field 7. Admin clicks "Login" button 8. System checks if username exists in database 9. System checks if password matches username 10. System creates login session for admin 11. System records login time and date 12. System redirects admin to Admin page 13. System displays welcome message: "Welcome, [Admin Name]"
Alternate Flow	8a. If username does not exist in database; 1. System displays error message: "Username not found" 2. System returns to login form 9a. if password does not match; 1. System displays error message: "Invalid password. Please try again." 2. System return to login form 3. After 3 failed attempts, system locks account for 30 minutes.
Exception	None

Table 3.1.1 Table for Use Case 1 Specification

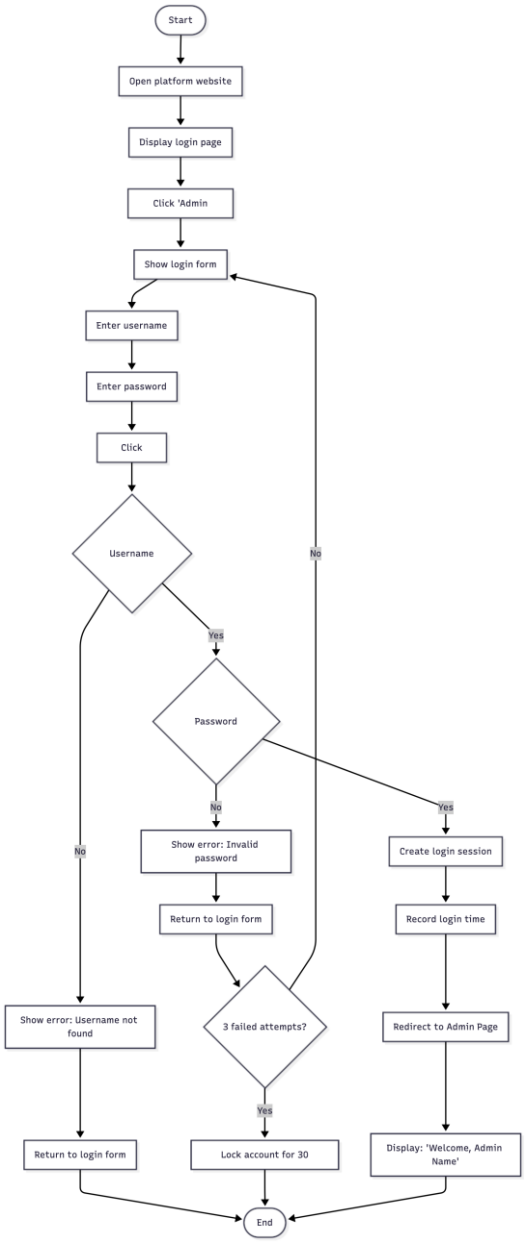

Diagram 3.1.1 Flow Chart UC1

3.1.2 Log out account

The admin is able to end their session and logs out from the admin dashboard to secure their account.

Use Case ID	UC2
Use Case Name	Log Out Account
Primary Actor	Admin
***Pre-Conditions**	1. Admin session is active

Post-Conditions	1. Admin is logged out from the system
Basic Flow	1. Admin clicks on their profile icon in top right corner 2. System displays dropdown menu with options 3. Admin clicks "Logout" option 4. System displays confirmation: "Are you sure you want to logout?" 5. Admin clicks "Yes, Logout" button 6. System ends admin session 7. System redirects admin to login page 8. System displays message: "You have been logged out successfully"
Alternate Flow	None
Exception	None

Table 3.1.2 Table for Use Case Specification 2

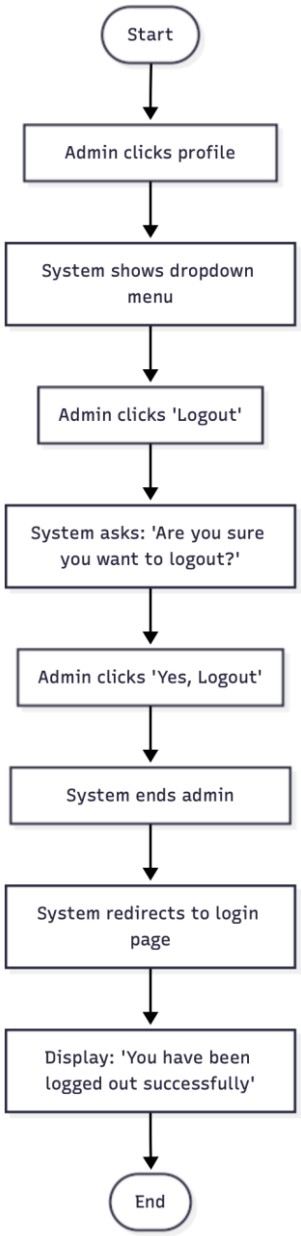

Diagram 3.1.2 Flow Chart UC2

3.1.3 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.

Use Case ID	UC3
Use Case Name	Monitor Mood Alert Dashboard
Primary Actor	Admin
***Pre-Conditions**	1. Admin is logged into the system

	2. Mood tracking system is active
Post-Conditions	1. Student and counselor receive notifications if assignment made 2. Admin dashboard updates to reflect current alert status 3. Alert is reviewed and appropriate action taken
Basic Flow	1. Admin navigates to Mood Alerts section from left sidebar 2. System displays mood alert dashboard with all flagged students sorted by severity 3. System shows key information for each alert: student name, ID, program, mood score percentage, severity level, duration below threshold, assignment status 4. Admin reviews alert details including mood trend patterns and student activity 5. Admin filters alerts by severity level (Critical, High, Medium) 6. Admin identifies critical cases (below 30% mood score) requiring immediate action 7. Admin clicks "Assign" button for unassigned critical alert 8. System displays available counselors with specializations and current caseload 9. Admin selects appropriate counselor and confirms assignment 10. System sends automated notification to both student and counselor 11. System updates alert status to "Assigned" 12. Admin marks alert as "Reviewed" in the system
Alternate Flow	6a. if no critical alerts exist; 1. admin reviews high and medium priority alerts instead 8a. if all appropriate counselors are at maximum caseload (20 students); 1. System displays warning message about counselor capacity
Exception	None

Table 3.1.3 Table for Use Case Specification 3

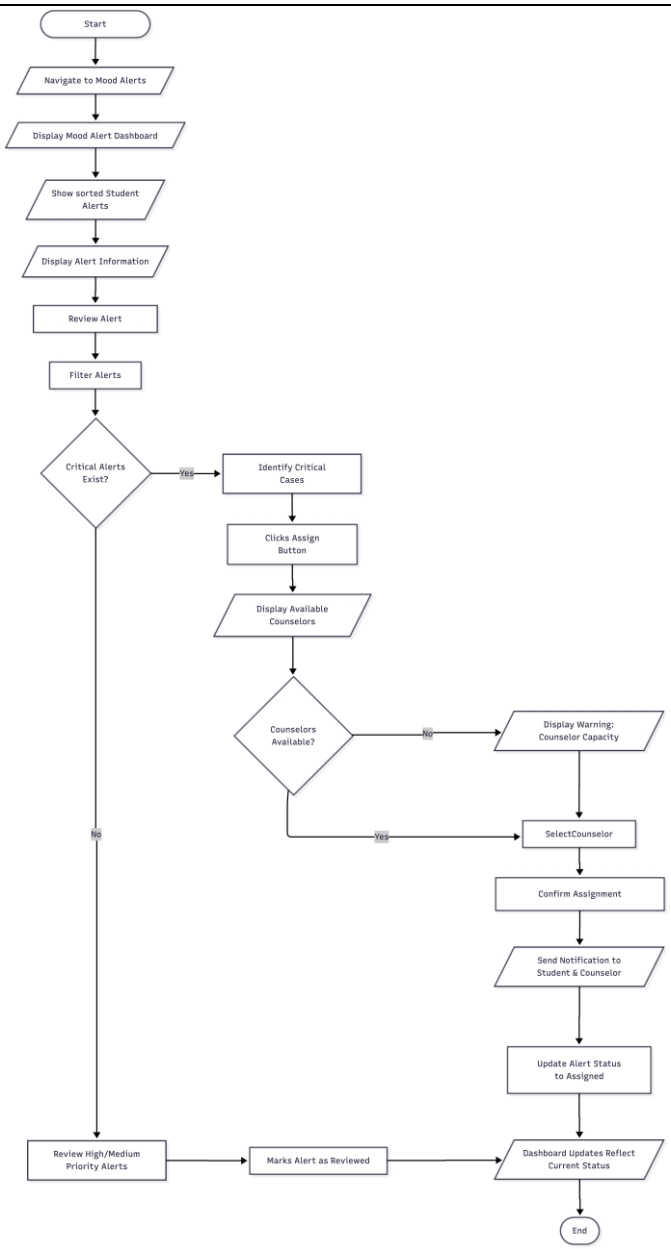

Diagram 3.1.3 Flow Chart UC3

3.1.4 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).

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

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

Table 3.1.4 Table for Use Case Specification 4

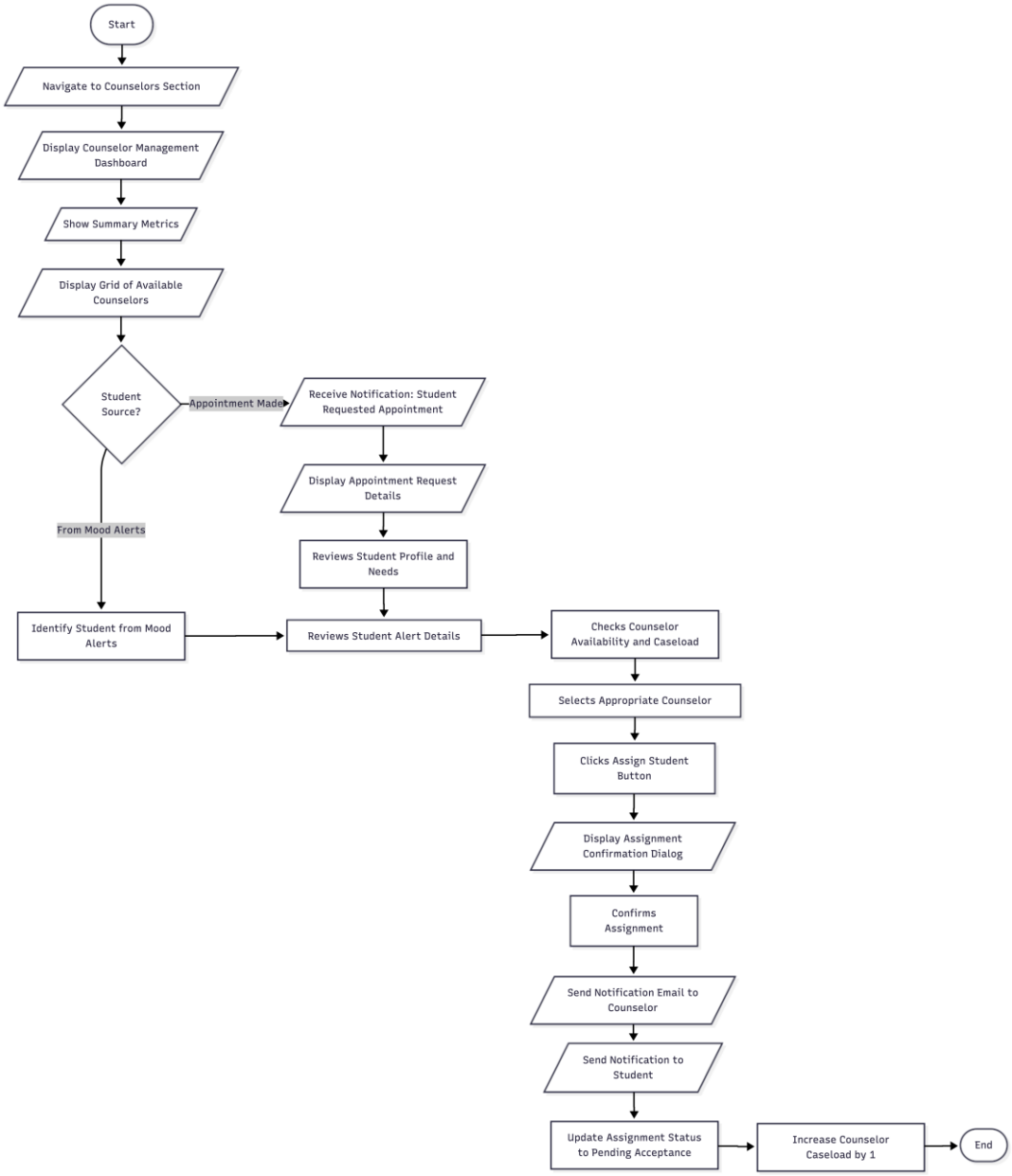

Diagram 3.1.4 Flow Chart UC4

3.1.5 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).

Use Case ID	UC5
Use Case Name	Review Reported Contents
Primary Actor	Admin

***Pre-Conditions**	1. Admin is logged into the system 2. Scoring system is active and configured with penalty values
Post-Conditions	1. If violation confirmed, content removed and points deducted automatically 2. Report marked "Resolved" 3. If user score falls below 60%, posting privileges restricted
Basic Flow	1. Admin navigates to Reports & Moderation section from left sidebar 2. System displays report queue with all pending reports sorted by priority 3. System shows report summary including: report ID, violation type, reporter info, reported user, content preview, time submitted, priority level 4. Admin reviews priority indicators (High Priority reports highlighted with orange border) 5. Admin selects report to review by clicking on report card 6. System displays full report details 7. Admin evaluates whether content violates community guidelines 8. Admin determines violation severity: Minor (guideline violation), Moderate (harassment), Severe (threatening behavior) 9. Admin selects appropriate moderation action: o "Dismiss Report" if no violation found o "Warn User" for first-time minor violation o "Remove Content (-15)" for guideline violation o "Remove Content (-30)" for harassment o "Remove Content (-50)" for severe violation 10. Admin clicks selected action button 11. System displays confirmation dialog showing penalty points to be deducted 12. Admin confirms action 13. System automatically deducts points from reported user's score (15, 30, or 50 points based on severity) 14. System removes content from platform if "Remove" action selected 15. System sends automated notification to reported user explaining violation, penalty applied, and updated score 16. System checks if reported user's score falls below 60% threshold

	17. If score below 60%, system automatically restricts posting privileges 18. System marks report as "Resolved"
Alternate Flow	None
Exception	None

Table 3.1.5 Table for Use Case Specification 5

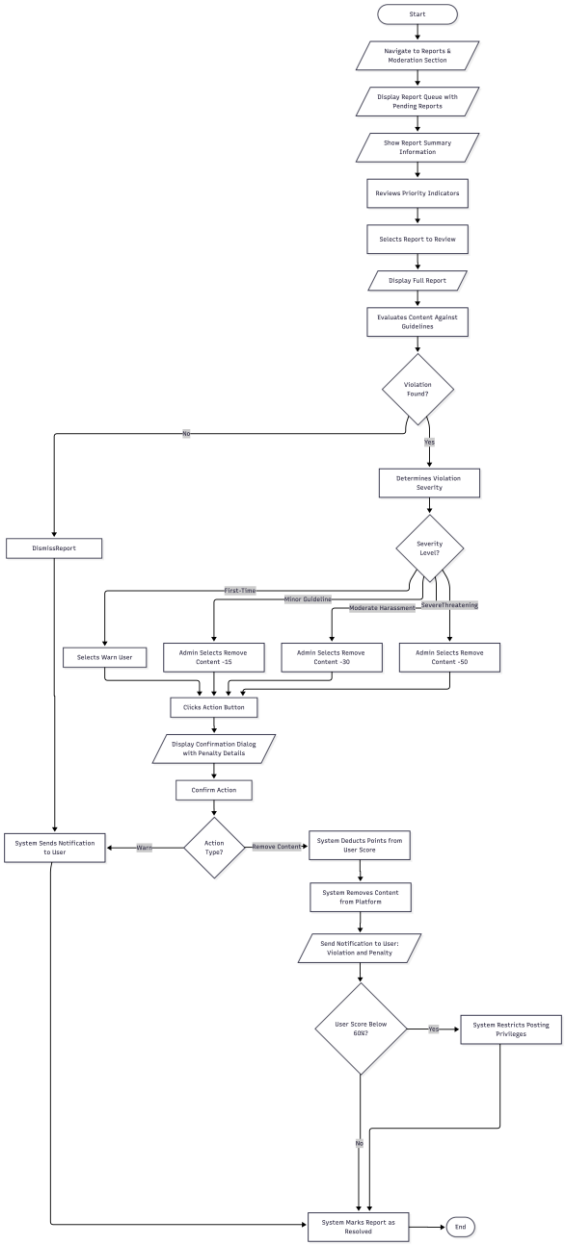

Diagram 3.1.5 Flow Chart UC5

3.1.6 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.

Use Case ID	UC6
Use Case Name	Manage Scoring System
Primary Actor	Admin
***Pre-Conditions**	1. Admin is logged into the system 2. Base scoring rules exist in system
Post-Conditions	1. New point values saved and active
Basic Flow	1. Admin navigates to Scoring System section from left sidebar 2. System displays scoring system management dashboard with two main sections: Configuration and Restricted Users 3. Admin reviews current scoring configuration showing: o Positive Contributions section (green cards): Post Creation +5, Helpful Answer +10, Peer Support +15 o Violations & Penalties section (red cards): Content Removal -15, Harassment -30, Severe Violation -50 4. Admin decides to adjust point values based on system effectiveness analysis 5. Admin clicks on point value input field for specific action (e.g., "Helpful Answer") 6. Admin enters new point value (e.g., changes from +10 to +12) 7. Admin repeats for other actions requiring adjustment 8. Admin clicks "Save Configuration" button 9. System validates all point values are within acceptable ranges 10. System displays confirmation: "Scoring configuration updated successfully" 11. System applies new point values to all future actions (existing scores unchanged)
Alternate Flow	None
Exception	None

Table 3.1.6 Table for Use Case Specification 6

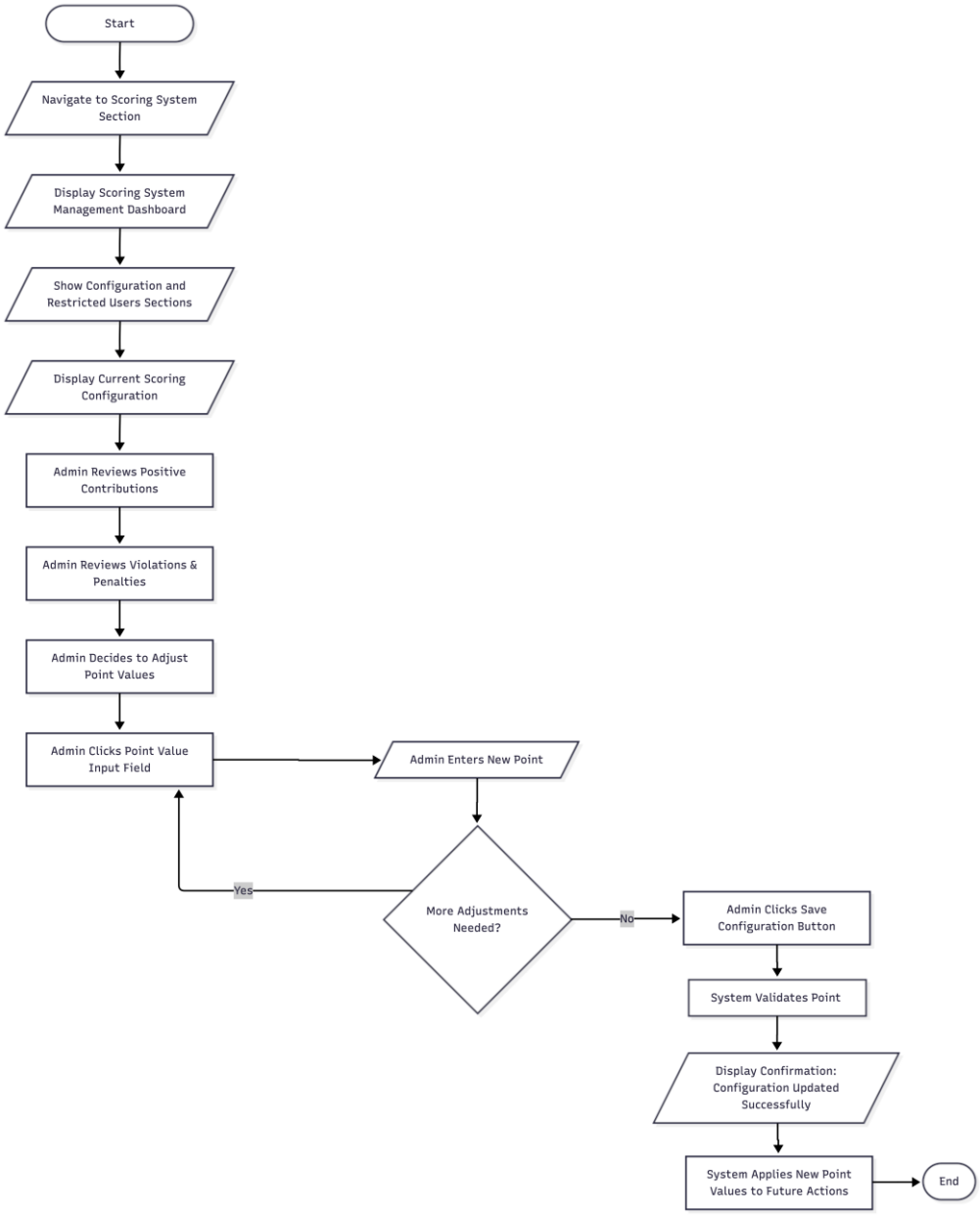

Diagram 3.1.6 Flow Chart UC6

3.1.7 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.

Use Case ID	UC7
Use Case Name	Process Score Appeals
Primary Actor	Admin

***Pre-Conditions**	1. Admin is logged into the system 2. User has submitted appeal
Post-Conditions	1. Appeal marked "approved" in system 2. User's score restored to exactly 60 points (60%)
Basic Flow	1. Admin navigates to Scoring System section from left sidebar 2. System displays "Restricted Users (Below 60%)" table 3. Admin identifies users with pending appeals (shown in "Appeals" column with count) 4. Admin clicks "Review Appeal" button for user with pending appeal 5. System opens appeal review interface displaying: o User's name and current score percentage o Restriction start date and duration restricted o Total number of violations and violation types o User's written appeal explanation 6. Admin clicks "History" button to view complete behavioral timeline 7. System displays comprehensive user history: o Chronological list of all violations with dates and penalties o Content that was removed with violation descriptions o Warnings issued and admin notes o Previous appeals and outcomes 8. Admin evaluates appeal 9. Admin determines appeal decision: o Approve and restore to 60% threshold o Deny appeal with 7-day extension before resubmission 10. Admin selects "Approve Appeal" option 11. System displays confirmation dialog: "Restore user to 60% score? Current score: $$%" 12. Admin confirms decision 13. System immediately restores user's score to exactly 60 points (60%) 14. System automatically removes posting restrictions 15. System sends automated notification to user 16. System marks appeal as "Approved - Resolved"
Alternate Flow	9a. If admin select "deny appeal" option 1. System prompts for denial reason selection from dropdown 2. Admin enter detailed explaination of denial reason 3. Admin confirm denial decision 4. system send notifications to user
Exception	None

Table 3.1.7 Table for Use Case Specification 7

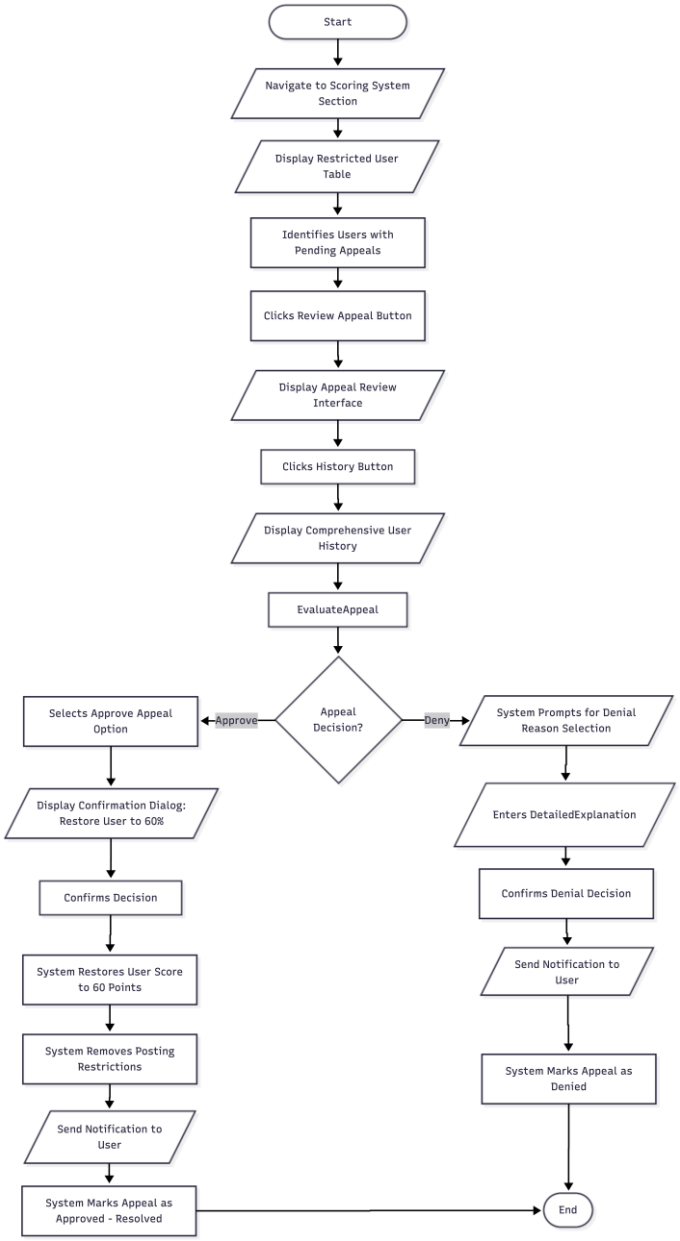

Diagram 3.1.7 Flow Chart UC7

3.1.8 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.

Use Case ID	UC8
Use Case Name	Suspend Accounts
Primary Actor	Admin
***Pre-Conditions**	1. Admin is logged into the system 2. User account has score below 60%

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

Table 3.1.8 Table for Use Case Specification 8

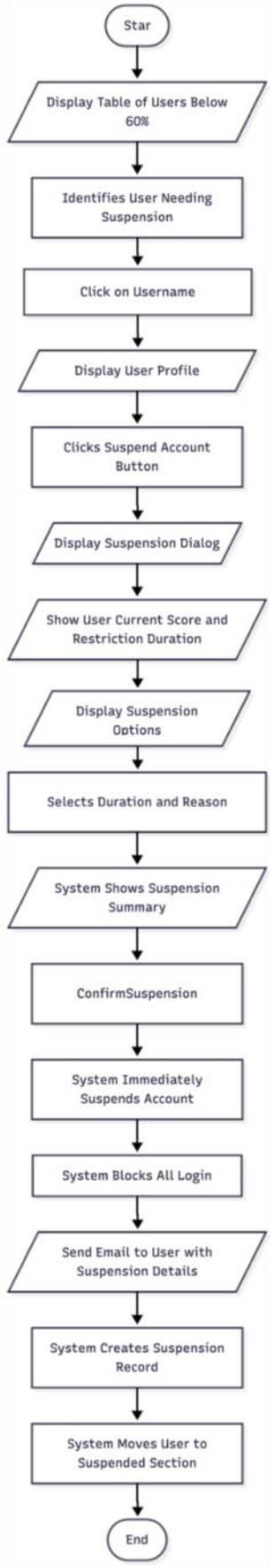

Diagram 3.1.8 Flow Chart UC8

3.2 Student (Peer)

3.2.1 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.

Use Case ID	UC9
Use Case Name	Sign up a new account
Primary Actor	Student
***Pre-Conditions**	Student is not registered.
Post-Conditions	New account is created and stored.
Basic Flow	1. Student opens sign-up page. 2. Student enters required details, including student ID, username, MMU official email, and password. 3. System validates inputs. 4. System creates the account. 5. System displays confirmation.
Alternate Flow	3a. Student ID or email already exists 3b. Password does not fulfill requirements
Exception	None

Table 3.2.1 Table for Use Case Specification 9

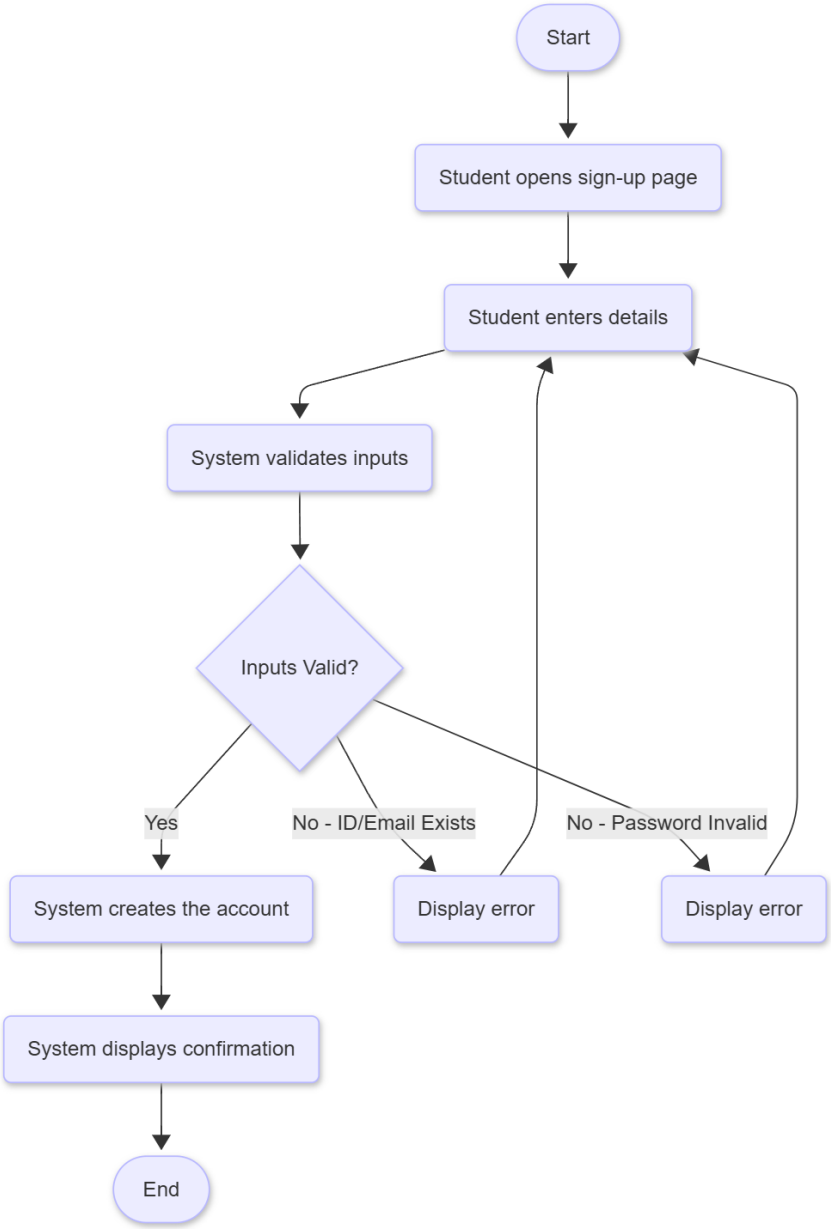

Diagram 3.2.1 Flow Chart UC9

3.2.2 Log in with own account

The student is 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.

Use Case ID	UC10
Use Case Name	Log in with own account
Primary Actor	Student
***Pre-Conditions**	Student has a registered account
Post-Conditions	Student is authenticated and session begins
Basic Flow	1. Student enters email and password

	2. System verifies credentials 3. System grants access
Alternate Flow	2a. Wrong password 2b. Invalid email account
Exception	Server is not responding

Table 3.2.2 Table for Use Case Specification 10

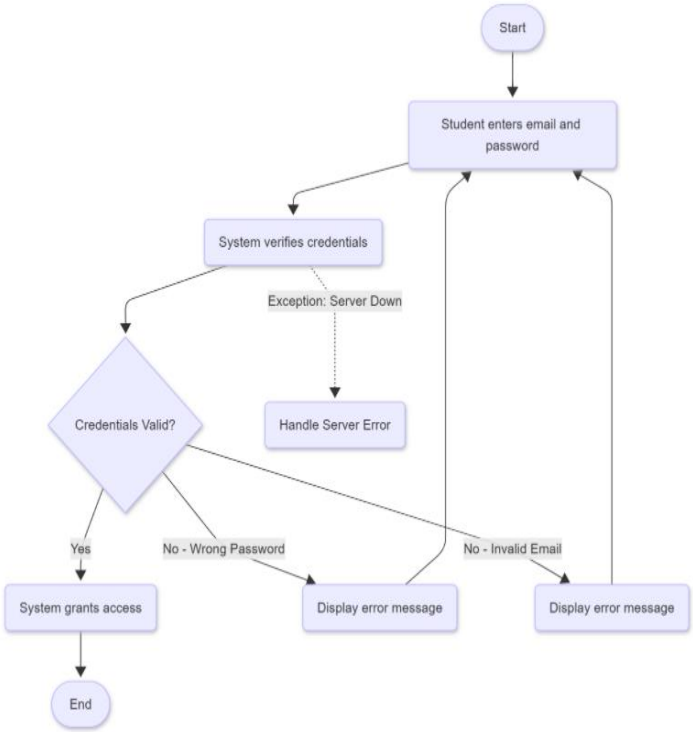

Diagram 3.2.2 Flow Chart UC10

3.2.3 Log out with own account

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

Use Case ID	UC11
Use Case Name	Log out with own account
Primary Actor	Student
***Pre-Conditions**	Active logged-in session
Post-Conditions	Session ends
Basic Flow	1. Student clicks on “Logout” 2. System ends session 3. System returns to login page
Alternate Flow	None

Exception	None

Table 3.2.3 Table for Use Case Specification 11

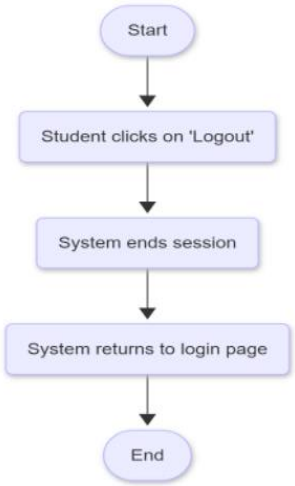

Diagram 3.2.3 Flow Chart UC11

3.2.4 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.

Use Case ID	UC12
Use Case Name	Reset password
Primary Actor	Student
***Pre-Conditions**	Student has a valid email account
Post-Conditions	Password is successfully updated
Basic Flow	1. Student clicks on “Forgot Password” 2. Student enters email address 3. Student clicks on “Send” 4. System sends OTP to the email address 5. Student enters OTP sent through email 6. System validates OTP 7. Student sets new password 8. Student confirms new password
Alternate Flow	6a. Incorrect OTP

	8a. Different input with the set new password field
Exception	None

Table 3.2.4 Table for Use Case Specification 12

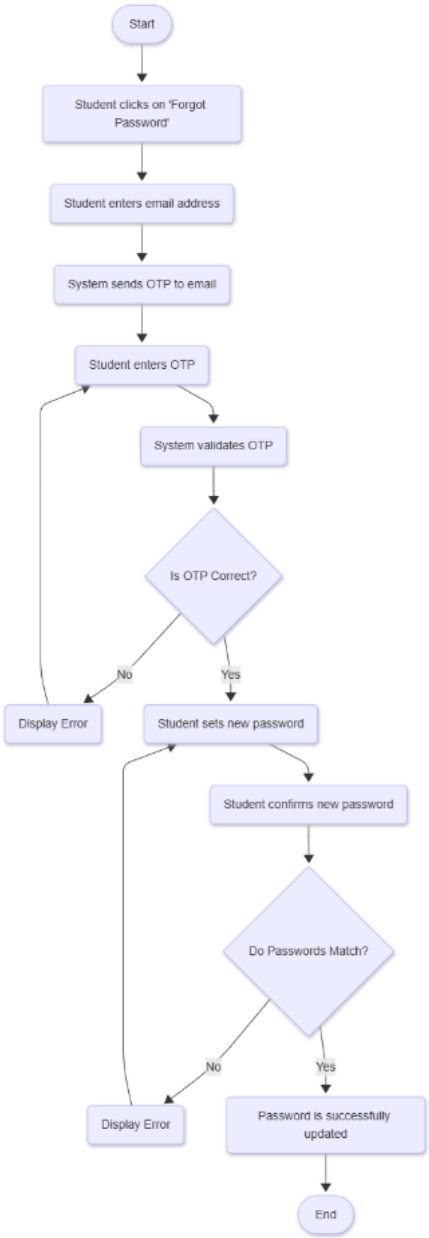

Diagram 3.2.4 Flow Chart UC12

3.2.5 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.

Use Case ID	UC13
Use Case Name	Report Contents
Primary Actor	Student

***Pre-Conditions**	Student is logged in
Post-Conditions	Report is submitted to moderators
Basic Flow	1. Student opens Forum 2. Student clicks on “Report” on a post/comment 3. Student chooses report reason 4. System records report 5. System notifies moderators
Alternate Flow	2a. No reason selected during submission
Exception	System fails to submit report

Table 3.2.5 Table for Use Case Specification 13

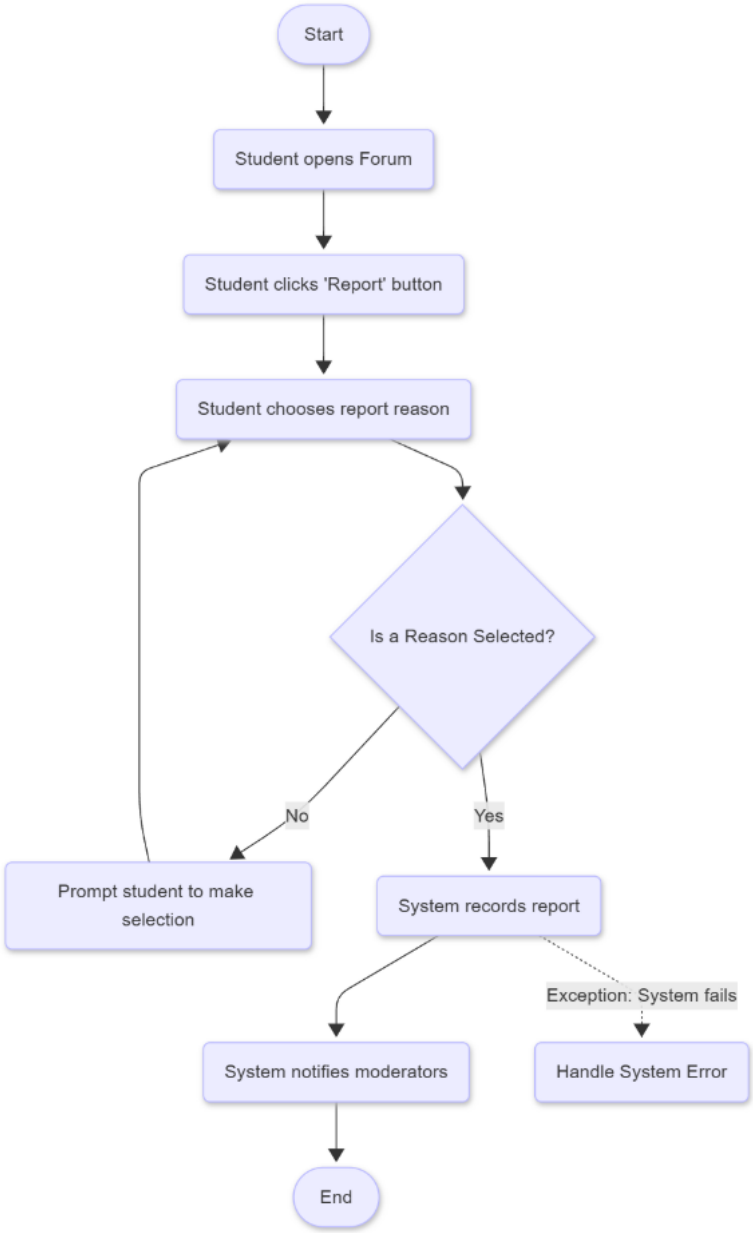

Diagram 3.2.5 Flow Chart UC13

3.2.6 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.

Use Case ID	UC14
Use Case Name	Build profile
Primary Actor	Student
***Pre-Conditions**	Student is logged in
Post-Conditions	Profile information is stored
Basic Flow	1. Student opens profile page 2. Student enters personal information 3. Student adds on their interests 4. System saves changes
Alternate Flow	3a. Interest label longer than 10 characters
Exception	None

Table 3.2.6 Table for Use Case Specification 14

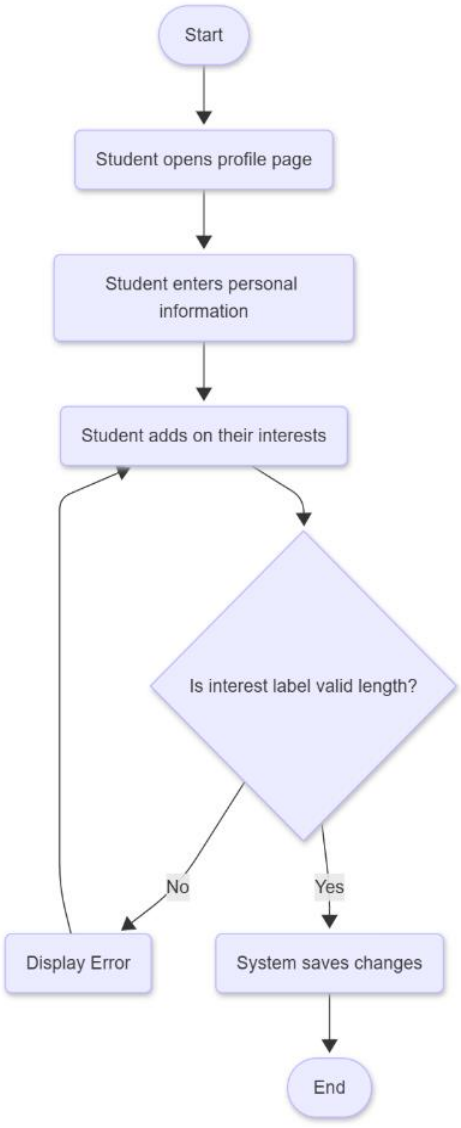

Diagram 3.2.6 Flow Chart UC14

3.2.7 Check-in mood and well-being

The student is able to check in their daily mood and well-being with available options through the prompted pop out message, and the respond will convert to score, and the score will be recorded in the database.

Use Case ID	UC15
Use Case Name	Check-in mood and well-being
Primary Actor	Student
***Pre-Conditions**	Student is logged in
Post-Conditions	Check-in entry stored
Basic Flow	1. Student chooses emotion level 2. Student clicks on “Confirm” 3. System saves entry
Alternate Flow	None
Exception	None

Table 3.2.7 Table for Use Case Specification 15

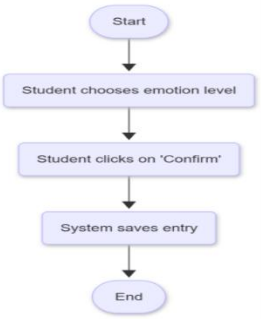

Diagram 3.2.7 Flow Chart UC15

3.2.8 Add friends

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

Use Case ID	UC16
Use Case Name	Add friends
Primary Actor	Student
***Pre-Conditions**	Student is logged in
Post-Conditions	Friend request is sent or accepted
Basic Flow	1. Student clicks on “Add Friend” button 2. System sends request 3. Other student accepts or declines

	4. System notifies student on status of adding friend
Alternate Flow	4a. Friend request accepted 4b. Friend request declined
Exception	None

Table 3.2.8 Table for Use Case Specification 16

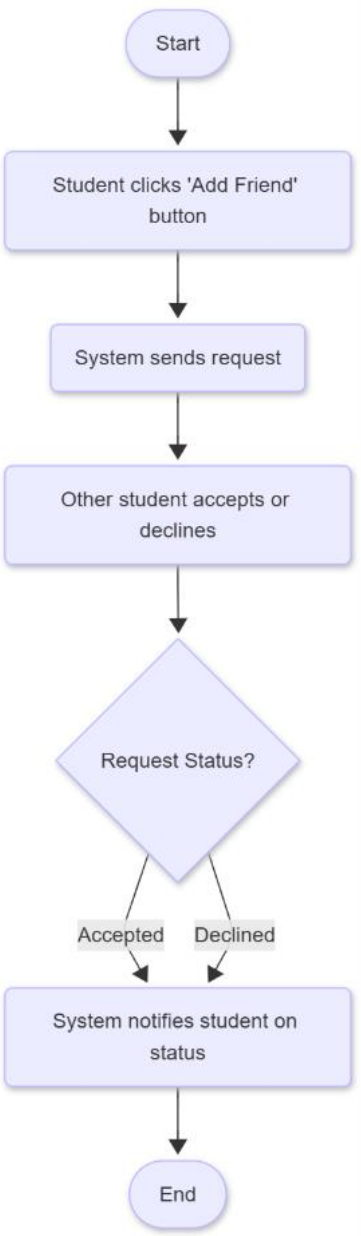

Diagram 3.2.8 Flow Chart UC16

3.2.9 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.

Use Case ID	UC17
Use Case Name	Chat privately
Primary Actor	Student
***Pre-Conditions**	Student is logged in and has a friend
Post-Conditions	Message sent/received
Basic Flow	1. Student opens Chat 2. Student selects chat partner 3. Student types message 4. Student sends message 5. System delivers message 6. Another selected student receives notification
Alternate Flow	None
Exception	Delivery failure

Table 3.2.9 Table for Use Case Specification 17

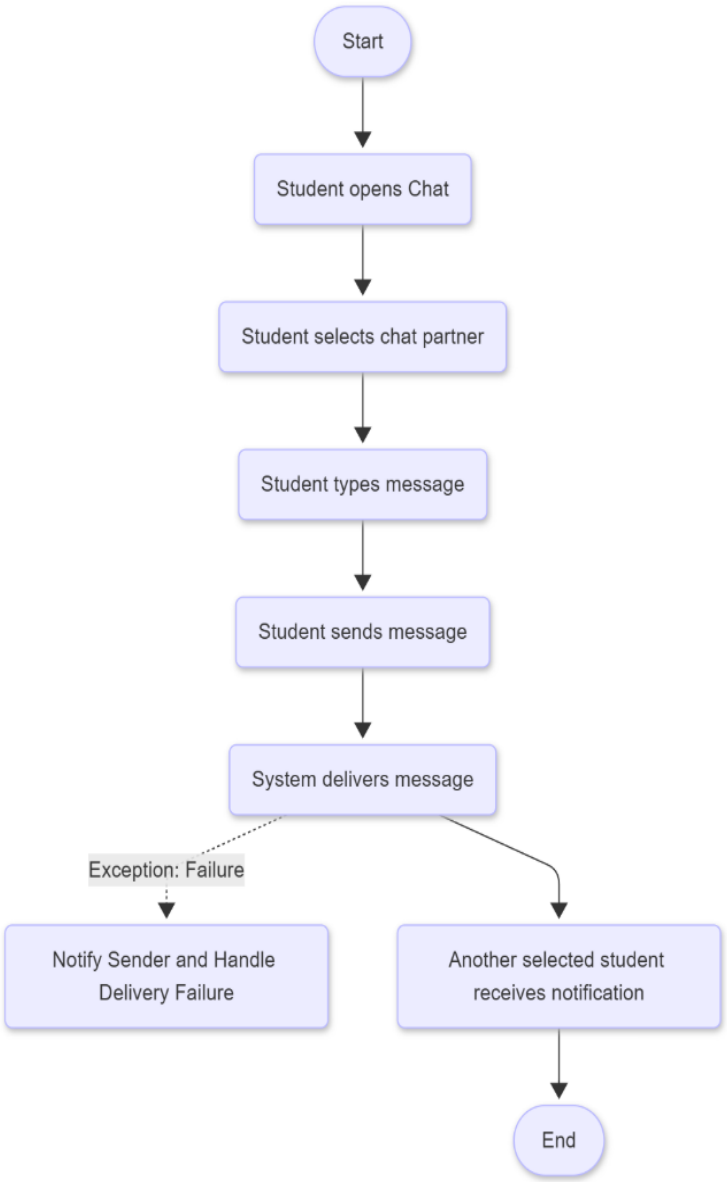

Diagram 3.2.9 Flow Chart UC17

3.2.10 Make posts

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

Use Case ID	UC18
Use Case Name	Make posts
Primary Actor	Student
***Pre-Conditions**	Student is logged in
Post-Conditions	New forum post is created
Basic Flow	1. Student opens Forum 2. Student clicks on “creates a post” 3. Student clicks on the username to show or hide username for post 4. Student enters text content 5. Student optionally attaches photos/videos/files 6. Student clicks on “Post” button 7. System validates post content 8. System publishes post 9. System displays the new post in the Forum
Alternate Flow	4a. Preview of attachment(s) and allow removal 6a. Invalid attachment type 6b. File too large
Exception	Attachment upload fails due to network issues; posting error

Table 3.2.10 Table for Use Case Specification 18

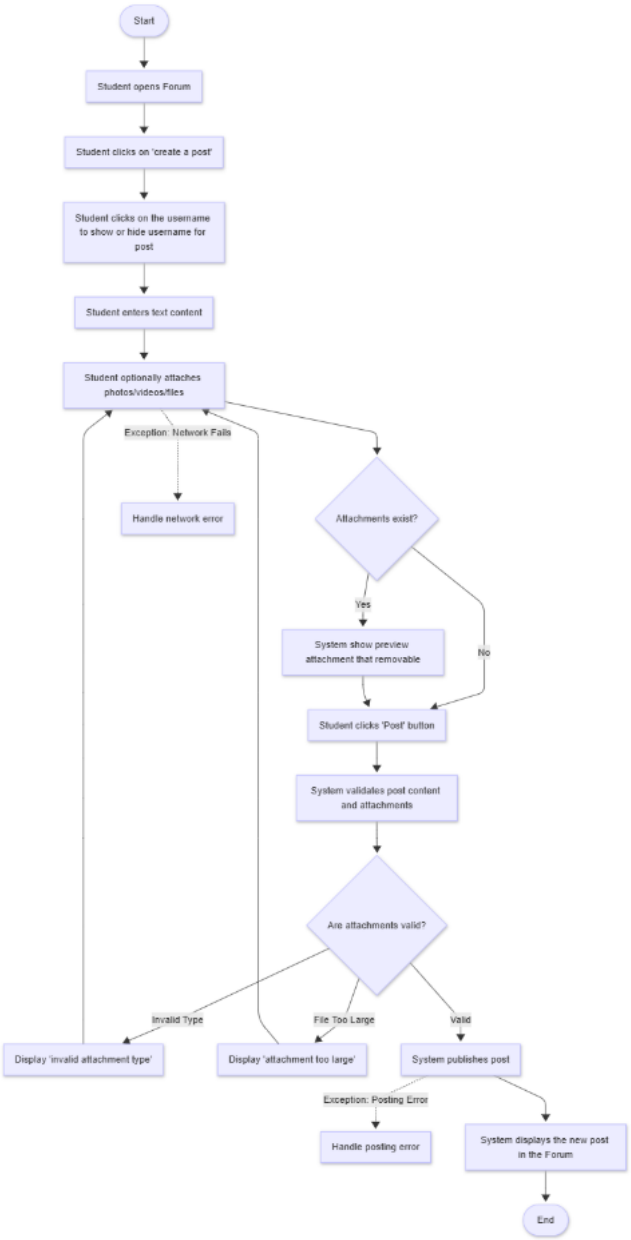

Diagram 3.2.10 Flow Chart UC18

3.2.11 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.

Use Case ID	UC19
Use Case Name	Comment on posts
Primary Actor	Student
***Pre-Conditions**	Student is logged in
Post-Conditions	Comment is added
Basic Flow	1. Student opens Forum 2. Student clicks on “comment” button of a post

	3. Student clicks on the username to show or hide username for comment 4. Student types comment 5. System publishes the comment
Alternate Flow	None
Exception	None

Table 3.2.11 Table for Use Case Specification 19

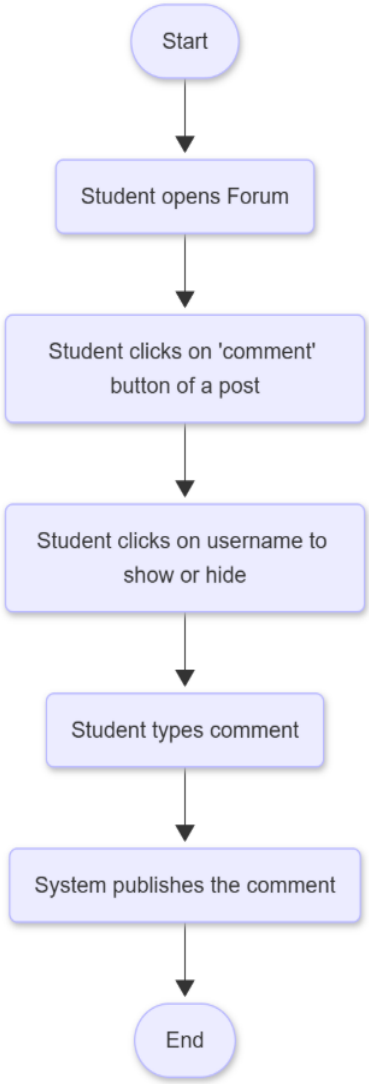

Diagram 3.2.11 Flow Chart UC19

3.2.12 Like posts

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

Use Case ID	UC20

Use Case Name	Like posts
Primary Actor	Student
***Pre-Conditions**	Student is logged in
Post-Conditions	Like count updated
Basic Flow	1. Student opens Forum 2. Student clicks on “Like” button for a post or a comment 3. System updates counter 4. System shows updated state
Alternate Flow	2a. System toggles unlike if student already liked
Exception	-

Table 3.2.12 Table for Use Case Specification 20

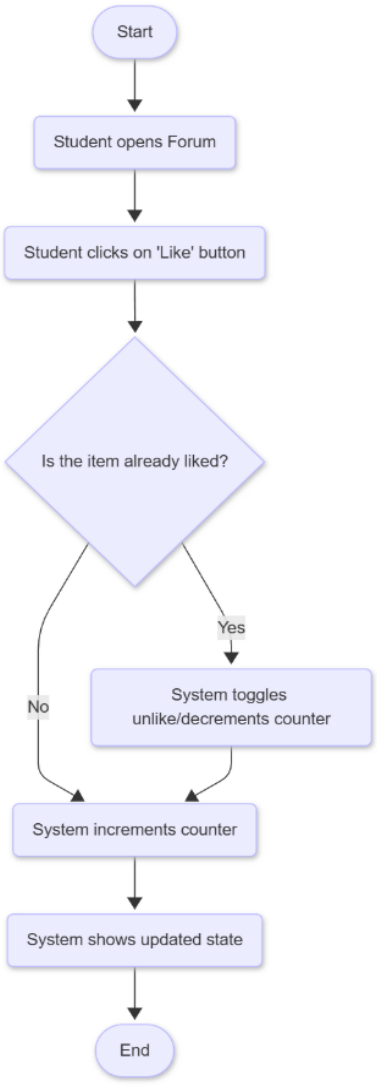

Diagram 3.2.12 Flow Chart UC20

3.2.13 Earn points

The student is able to earn points to level up through check-in daily mood, or getting 50 likes in a single post or comment published anonymously or non-anonymously in the forum.

Use Case ID	UC21
Use Case Name	Earn points
Primary Actor	Student
***Pre-Conditions**	Student is logged in
Post-Conditions	Points updated; Level updated if threshold reached
Basic Flow	1. Student performs an action eligible for earning points (a. Creates a post in forum (+5 points) b. Provides helpful answer/comment (+10 points) c. Gives peer support recognized by likes (+15 points when reaches 50 likes 2. System calculates awarded points 3. System updates student's total points 4. System check if level-up conditions are met 5. System displays new level in the profile
Alternate Flow	2a. System denies additional points if mood check-in more than once per day 2b. System denies additional points if a post/comment with having 50 likes already received points
Exception	Database update failure

Table 3.2.13 Table for Use Case Specification 21

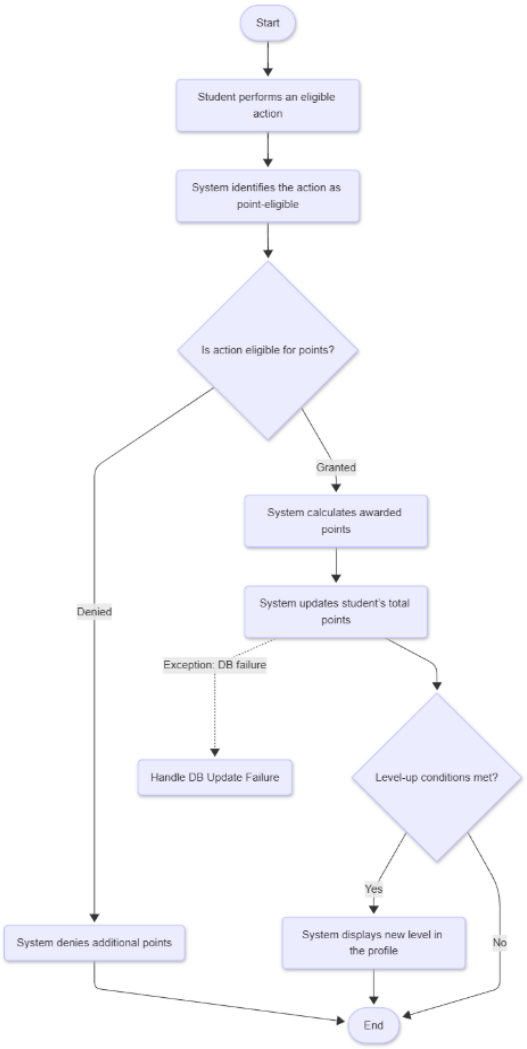

Diagram 3.2.13 Flow Chart UC21

3.2.14 Match-up with people

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

Use Case ID	UC22
Use Case Name	Match-up with people
Primary Actor	Student
***Pre-Conditions**	Student is logged in; Student has interest label in their profile
Post-Conditions	Suggested matches displayed
Basic Flow	1. Student opens “Match-Up” page 2. System retrieves the student’s interest labels 3. System searches for peer with matching interest labels 4. System displays suggested peer with profile built

	5. Student chooses available action they preferred (swipe left, swipe right, go back, add friend, or chat)
Alternate Flow	3a. No match found 5a. Show previous matched result if swiped left 5b. Match with next peers if swiped right 5c. Back to home page if 'x' button selected 5d. Friend request sent if add friend icon button selected 5e. Open private chats if chat icon button selected
Exception	None

Table 3.2.14 Table for Use Case Specification 22

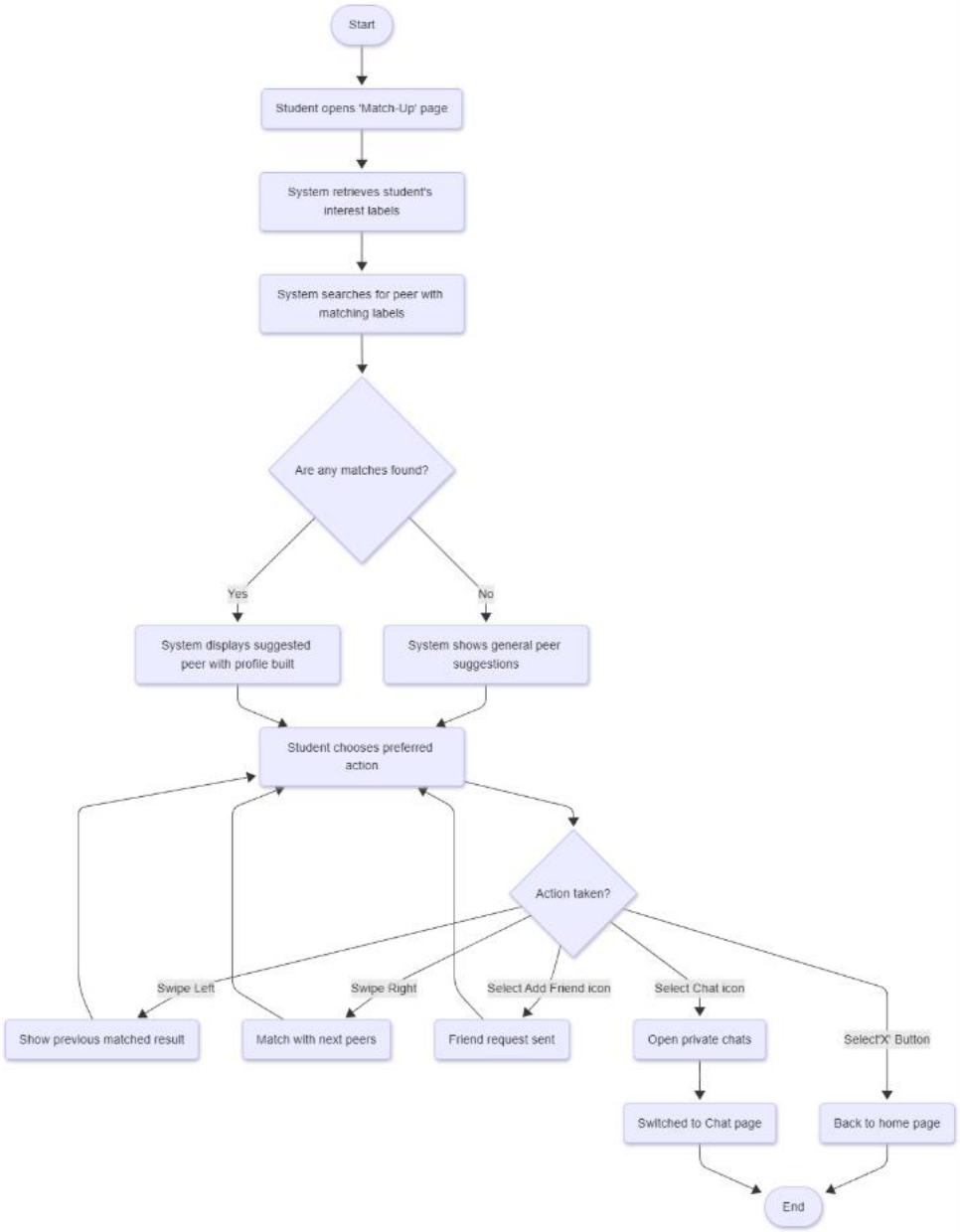

Diagram 3.2.14 Flow Chart UC22

3.2.15 Book counselor appointments

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

Use Case ID	UC23
Use Case Name	Book counselor appointments
Primary Actor	Student
***Pre-Conditions**	Student is logged in
Post-Conditions	Appointment request stored and sent to admin
Basic Flow	1. Student opens Counselor page 2. Student choses preferred counselor 3. System displays the preferred counselor availability 4. Student selects date and time based on the availability displayed 5. Student fills in the content field 6. Student clicks on submit button 7. System checks availability 8. System sends notification to admin
Alternate Flow	7a. Suggest alternatives if slot unavailable
Exception	None

Table 3.2.15 Table for Use Case Specification 23

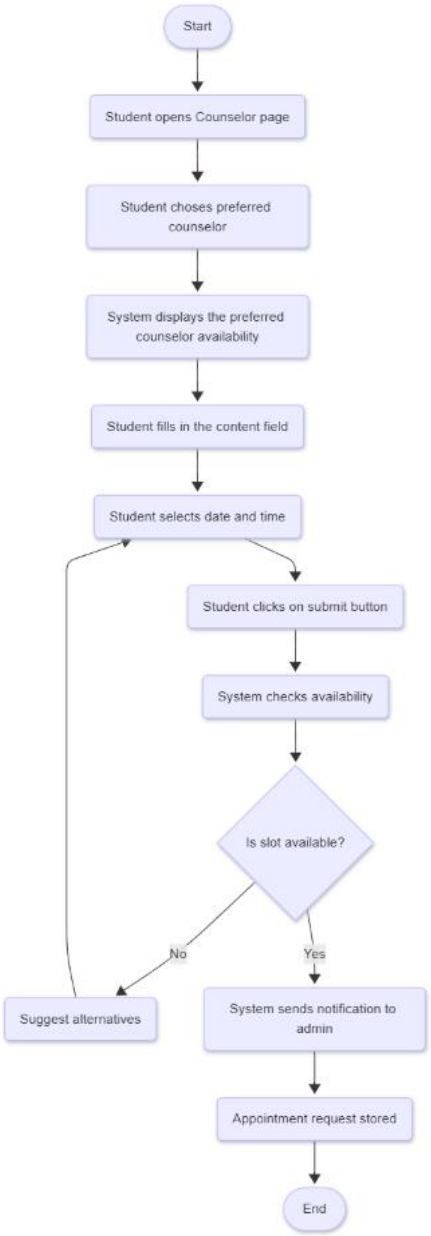

Diagram 3.2.15 Flow Chart UC23

3.3 Moderator

3.3.1 Log in account

The moderator is able to log into their account through email and password.

Use Case ID	UC24
Use Case Name	Log in account
Primary Actor	Moderator
Pre-Conditions	Moderator has an account
Post-Conditions	Moderator is logged in

Basic Flow	1. Moderator opens login page 2. Moderator enters email and password 3. System verifies details 4. Moderator is logged in 5. Moderator is taken to the home page
Alternate Flow	3a. Email and password are incorrect
Exception	None

Table 3.3.1 Table for Use Case Specification 24

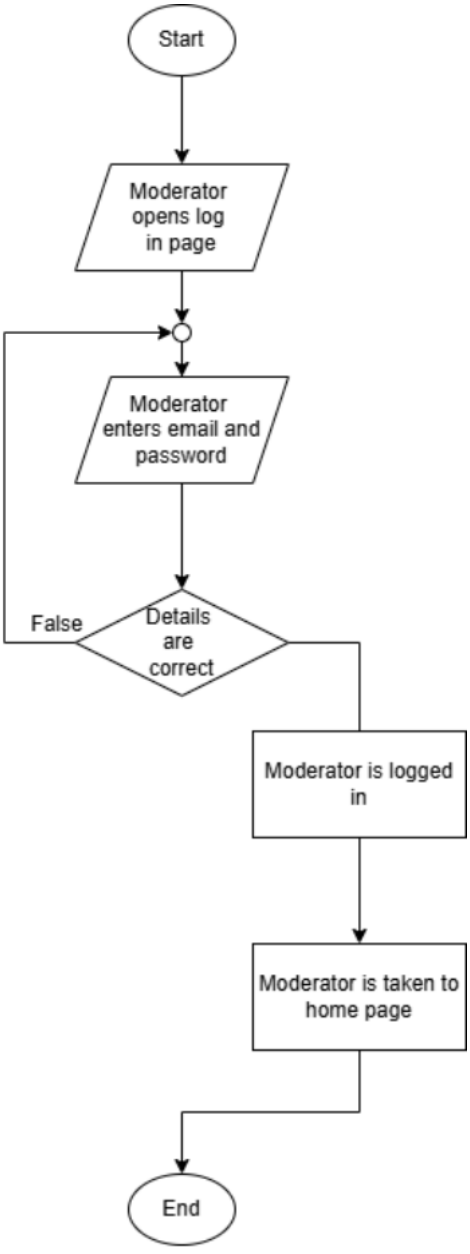

Diagram 3.3.1 Flow Chart UC24

3.3.2 Log out account

The moderator is able to log out of their account.

Use Case ID	UC25
Use Case Name	Log out account
Primary Actor	Moderator
Pre-Conditions	Moderator is logged in
Post-Conditions	Moderator is logged out of account
Basic Flow	1. Moderator navigates to account page 2. Moderator clicks the log out button 3. System requests confirmation 4. Moderator confirms decision 5. System logs out moderator 6. Moderator is taken to the log in page
Alternate Flow	None
Exception	None

Table 3.3.2 Table for Use Case Specification 25

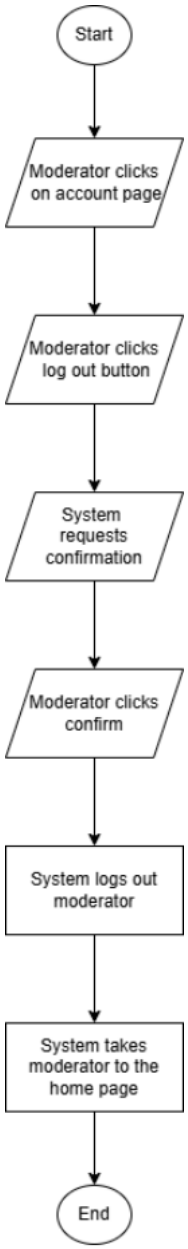

Diagram 3.3.2 Flow Chart UC25

3.3.3 Review Reported Messages

The moderator is able to review messages reported by students.

Use Case ID	UC26
Use Case Name	Review reported messages
Primary Actor	Moderator
Pre-Conditions	Moderator is logged in; Student has reported password
Post-Conditions	Students record is updated; students record remains the same
Basic Flow	1. Moderator navigates to moderator dashboard page 2. Moderator clicks on reported messages button 3. System displays list of reported messages 4. Moderator clicks on a message in the list 5. System asks if the message breaks user guidelines 6. Moderator selects decision 7. System requests confirmation 8. Moderator confirms decision
Alternate Flow	6a. Moderator clicks yes, violation is added record 6b. Moderator clicks no, student record is unchanged
Exception	-

Table 3.3.3 Table for Use Case Specification 26

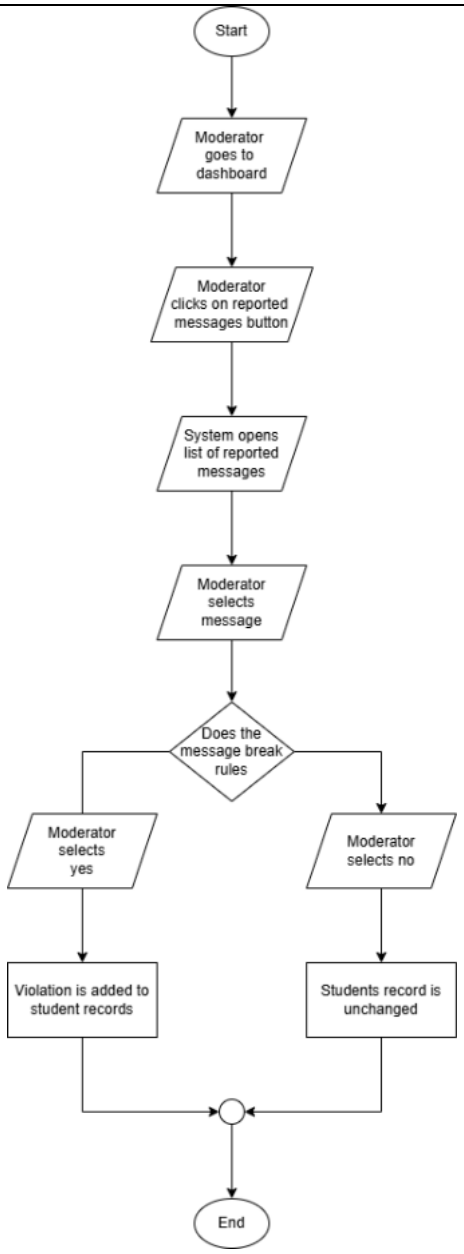

Diagram 3.3.3 Flow Chart UC26

3.3.4 Flag Student Account

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

Use Case ID	UC27
Use Case Name	Flag Student Account
Primary Actor	Moderator
Pre-Conditions	Moderator is logged in; student has more than one violation
Post-Conditions	Student account is to be reviewed by the admin
Basic Flow	1. Moderator opens moderator dashboard 2. Moderator clicks on potential flag students list 5. Moderator clicks on student name

	6. System opens popup to fill in report 7. Moderator fills in report 8. System requests confirmation 9. Moderator clicks confirm 10. Student account is sent to Admin to be reviewed
Alternate Flow	None
Exception	None

Table 3.3.4 Table for Use Case Specification 27

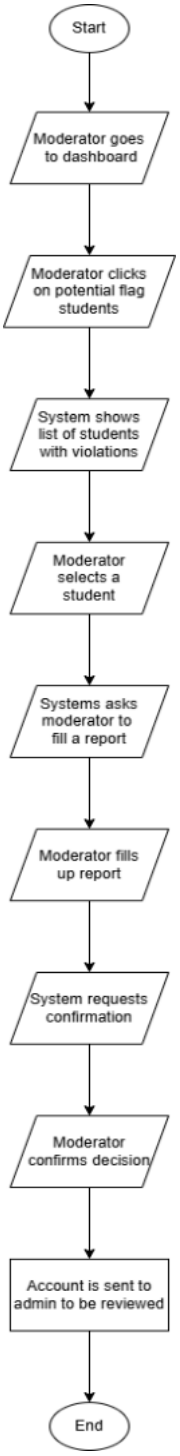

Diagram 3.3.4 Flow Chart UC27

3.3.5 Make Announcements

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

Use Case ID	UC28
Use Case Name	Make Announcements
Primary Actor	Moderator
Pre-Conditions	Moderator is logged in
Post-Conditions	Announcement is sent to students
Basic Flow	1. Moderator navigates moderator dashboard 2. Moderator clicks on new announcement button 3. System opens popup to enter title and content 4. Moderator enters title content 5. System requests confirmation 6. Moderator confirms decision 7. Announcement is posted
Alternate Flow	None
Exception	None

Table 3.3.5 Table for Use Case Specification 28

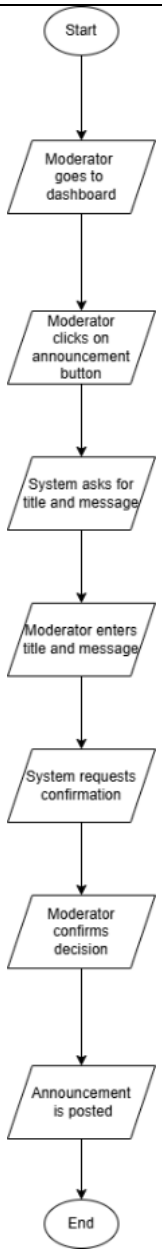

Diagram 3.3.5 Flow Chart UC28

3.3.6 Delete Forum Posts

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

Use Case ID	UC29
Use Case Name	Delete Forum Posts
Primary Actor	Moderator
Pre-Conditions	Moderator is logged in; post is on the forum
Post-Conditions	Post is deleted
Basic Flow	3. Moderator opens home page 4. Moderator clicks on post 5. Moderator clicks on options button

	6. Moderator clicks on delete button 7. System requests confirmation 8. Moderator confirms decision 9. Post is deleted
Alternate Flow	None
Exception	None

Table 3.3.6 Table for Use Case Specification 29

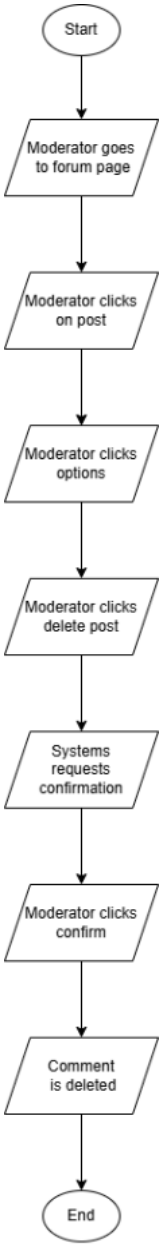

Diagram 3.3.6 Flow Chart UC29

3.3.7 Delete Forum Comments

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

Use Case ID	UC30
Use Case Name	Delete Forum Comments
Primary Actor	Moderator
Pre-Conditions	Moderator is logged in; Post is in the forum; Comment is under post
Post-Conditions	Comment is deleted
Basic Flow	1. Moderator opens home page 2. Moderator clicks on post 3. Moderator clicks on comment 4. Moderator clicks on options button 5. Moderator clicks on delete button 6. System requests confirmation 7. Moderator confirms decision 8. Comment is deleted
Alternate Flow	None
Exception	None

Table 3.3.7 Table for Use Case Specification 30

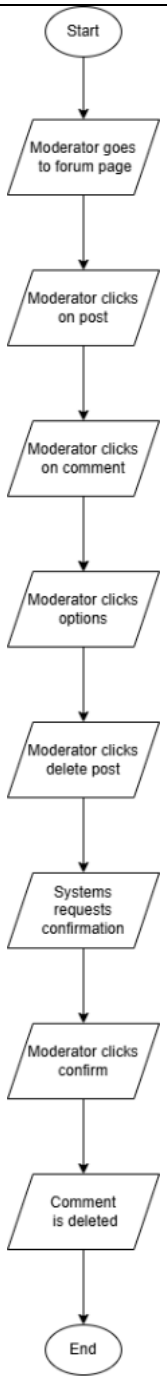

Diagram 3.3.7 Flow Chart UC30

3.4 Counselor

3.4.1 Log in own account

The counselor is able to securely authenticate to the system using their unique credentials, ensuring only authorized personnel can access sensitive student information and counseling tools, with multi-factor authentication available for enhanced security.

Use Case ID	UC31

Use Case Name	Log in own account
Primary Actor	Counselor
***Pre-Conditions**	1. Counselor has valid system credentials 2. The system is online and accessible 3. Counselor's account is active and not locked
Post-Conditions	1. Counselor is authenticated and redirected to dashboard 2. Session is established with appropriate permissions 3. Login time is recorded in audit log
Basic Flow	1. System displays login page 2. Counselor enters username and password 3. Counselor clicks "Login" 4. System validates credentials 5. System grants access and redirects to dashboard
Alternate Flow	2a. Forgot Password 1. Counselor clicks "Forgot Password" 2. System sends password reset link to registered email 3. Counselor follows the link and sets new password 4. Continue from step 2 of basic flow
Exception	Invalid Credentials 1. System displays error message 2. Login attempt is logged 3. After 3 failed attempts, account is temporarily locked

Table 3.4.1 Table for Use Case Specification 31

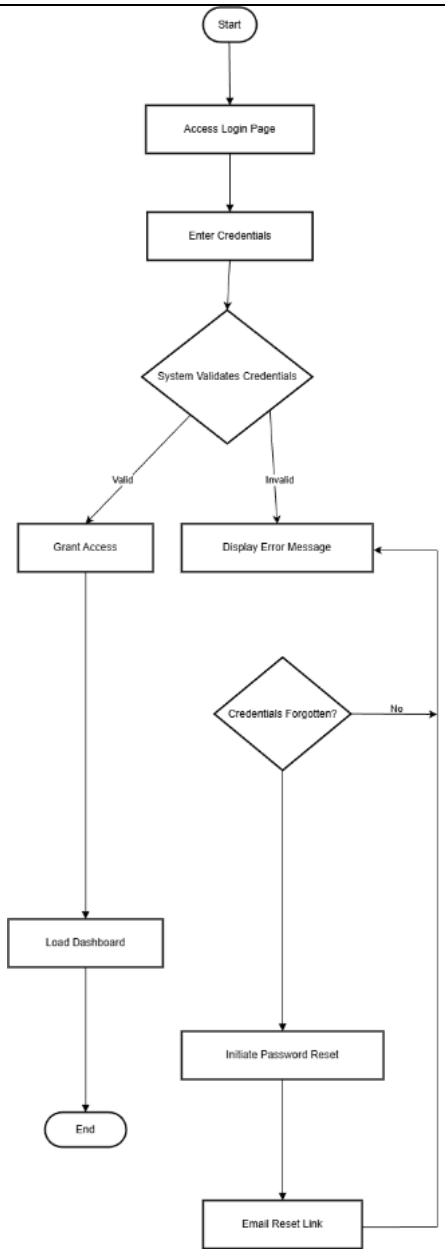

Diagram 3.4.1 Flow Chart UC31

3.4.2 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.

Use Case ID	UC32
Use Case Name	Manage caseload
Primary Actor	Counselor
***Pre-Conditions**	1. Counselor is logged in 2. Counselor has assigned students 3. Counselor has permission to view/manage caseload

Post-Conditions	1. Student information is updated in system 2. Changes are recorded in audit log 3. Dashboard reflects updated caseload
Basic Flow	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
Alternate Flow	3a. Filter/Search Caseload 1. Counselor uses search/filter options 2. System displays filtered results 3. Counselor selects from filtered list
Exception	No Students Assigned 1. System displays "No students assigned" message 2. Option to request assignments from admin

Table 3.4.2 Table for Use Case Specification 32

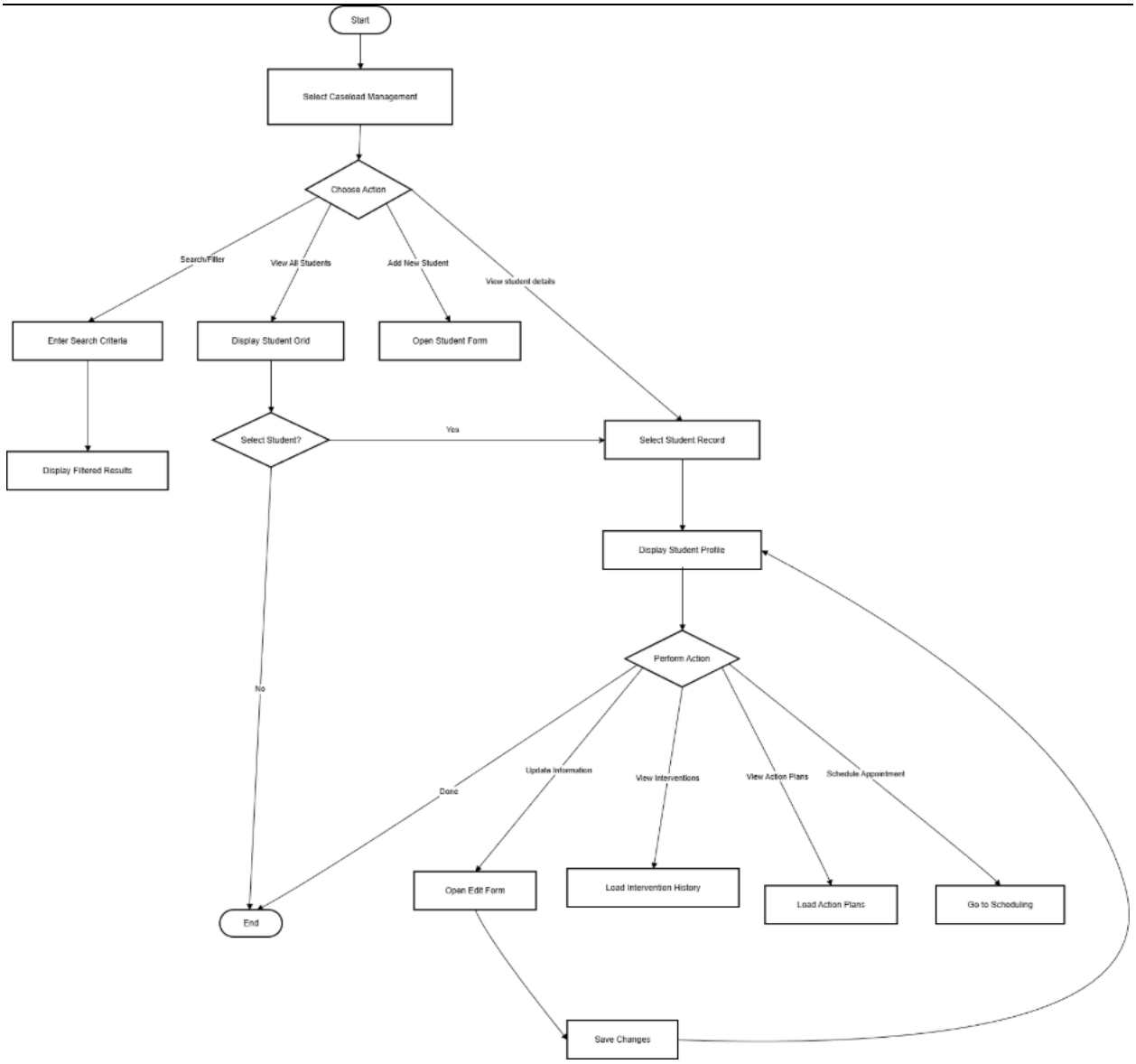

Diagram 3.4.2 Flow Chart UC32

3.4.3 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.

Use Case ID	UC33
Use Case Name	Export data
Primary Actor	Counselor
***Pre-Conditions**	1. Counselor is logged in 2. Counselor has data viewing permissions 3. Export functionality is enabled for counselor's role

Post-Conditions	1. Data file is generated and downloaded 2. Export action is logged in audit trail 3. Downloaded file is in specified format
Basic Flow	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
Alternate Flow	**Email Export** 6a. Counselor selects "Email to me" option 1. System generates file and sends to counselor's email
Exception	**Large Data Set** 1. System warns about large export size 2. Option to narrow filters or proceed with background processing

Table 3.4.3 Table for Use Case Specification 33

Diagram 3.4.3 Flow Chart UC33

3.4.4 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.

Use Case ID	UC34
Use Case Name	Verify assignments
Primary Actor	Counselor
***Pre-Conditions**	1. Counselor is logged in 2. Admin has assigned new students to counselor 3. Counselor has permission to review assignments

Post-Conditions	1. Assignment is accepted/declined 2. Caseload is updated accordingly 3. Admin is notified of decision
Basic Flow	1. System notifies counselor of pending assignments 2. Counselor navigates to "Pending Assignments" 3. System displays list with student details 4. Counselor reviews each assignment 5. Counselor selects "Accept" or "Decline" 6. If declined, counselor provides reason
Alternate Flow	4a. Request More Information 1. Counselor selects "Need More Info" 2. System sends request to admin 3. Decision deferred until info received
Exception	Assignment Expired 1. System shows assignment is no longer pending 2. Option to contact admin about reassignment

Table 3.4.4 Table for Use Case Specification 34

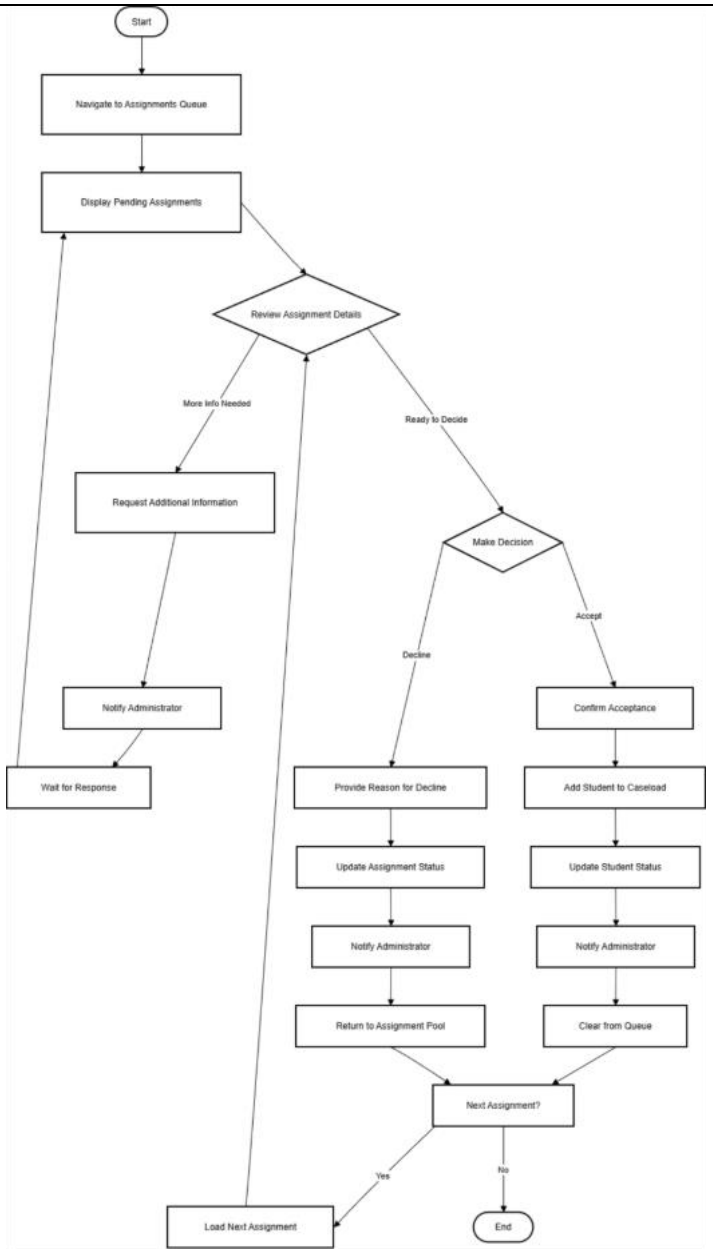

Diagram 3.4.4 Flow Chart UC34

3.4.5 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.

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

Post-Conditions	1. Action plan is created/updated 2. Plan is linked to student profile 3. Progress tracking is initialized
Basic Flow	1. Counselor selects student from caseload 2. Counselor selects "Action Plans" 3. System displays existing plans or options to create new 4. Counselor creates or edits plans with: - Goals - Strategies - Timeline - Progress metrics 5. Counselor saves plan 6. System generates plan summary
Alternate Flow	4a. Use Template 1. Counselor selects from template library 2. System pre-populates plan structure 3. Counselor customizes for student
Exception	Student Has Active Plan 1. System warns about existing active plan 2. Option to archive old plan or create additional plan

Table 3.4.5 Table for Use Case Specification 35

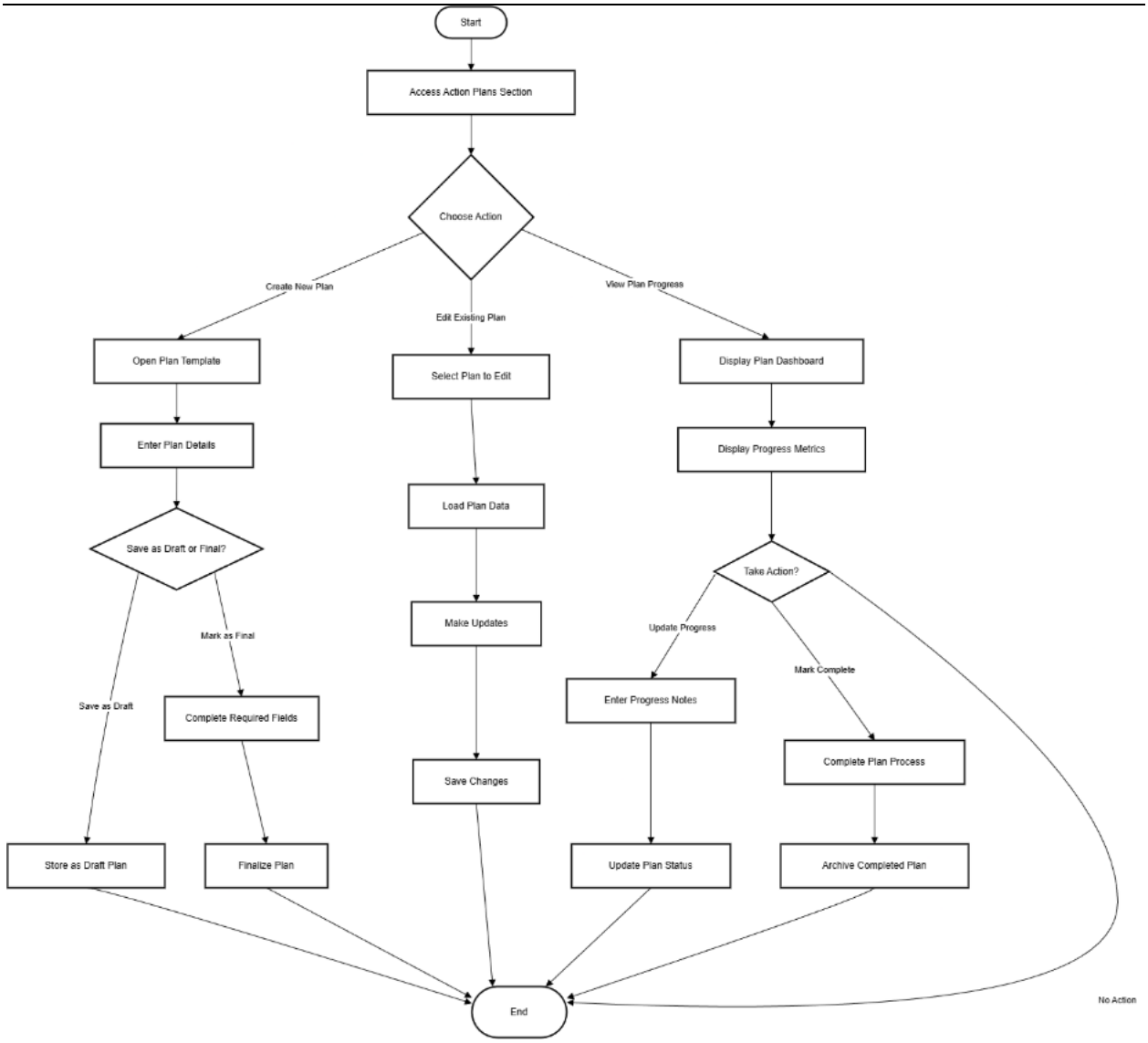

Diagram 3.4.5 Flow Chart UC35

3.4.6 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.

Use Case ID	UC36
Use Case Name	Schedule appointments
Primary Actor	Counselor
***Pre-Conditions**	1. Counselor is logged in 2. Counselor has set availability 3. Student is in counselor's caseload
Post-Conditions	1. Appointment is scheduled in calendar 2. Notifications are sent to student 3. Counselor's calendar is updated

Basic Flow	1. Counselor selects student from caseload 2. Counselor clicks "Schedule Appointment" 3. System displays counselor's available slots 4. Counselor selects date/time 5. Counselor selects appointment type and duration 6. The counsellor confirms booking 7. System sends confirmation to student
Alternate Flow	2a. Student Self-Booking 1. Student books from available slots 2. System confirms with both parties
Exception	Time Slot Unavailable 1. System shows slot is no longer available 2. Option to select different time or waitlist

Table 3.4.6 Table for Use Case Specification 36

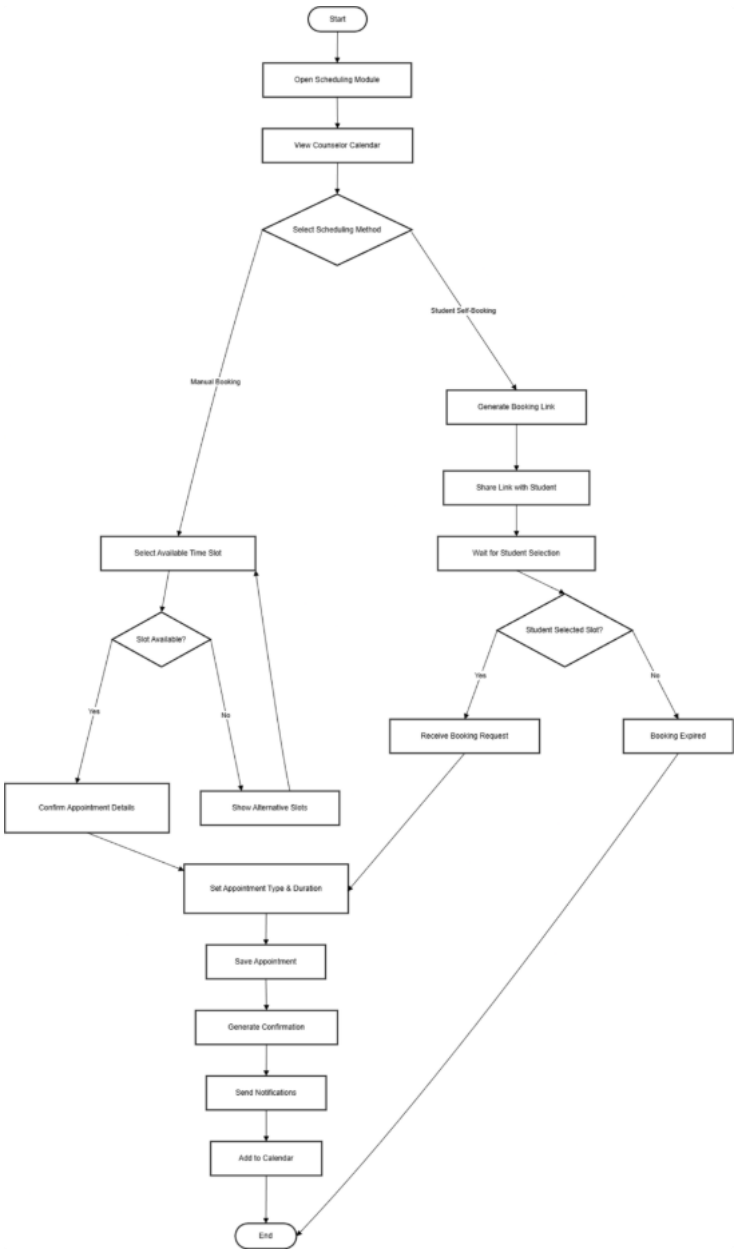

Diagram 3.4.6 Flow Chart UC36

3.4.7 Log out own account

The counselor is able to securely terminate their system session to prevent unauthorized access to confidential student data, with the system automatically logging the logout event for audit trail purposes.

Use Case ID	UC37
Use Case Name	Log out with own account
Primary Actor	Counselor
***Pre-Conditions**	1. The counselor is currently logged into an active session within the system.
Post-Conditions	1. The user session is securely terminated, all local session data is cleared, and the system redirects to the public login page. 2. The session remains active, so the user stays on the current page.
Basic Flow	1. The counselor selects the “Log Out” option from the user menu or navigation bar. 2. The system presents a confirmation dialog: “Are you sure you want to log out?” 3. The counselor confirms the action. 4. The system:
Alternate Flow	1a. Automatic Logout Due to Inactivity 1. The system displays a warning modal: “Your session will expire due to inactivity in 60 seconds. Click ‘Stay Logged In’ to continue.” 2. If the counselor clicks “Stay Logged In,” the session timer is reset and the modal closes. The use case ends. 3. If the counselor takes no action or clicks “Log Out Now,” the system proceeds to step 4 of the Basic Flow, logging the event as an “auto-logout.”
Exception	**Network Failure During Logout** 1. The system displays an error: “Unable to securely log out currently. Please try again or close your browser.” 2. The system performs a best-effort local cleanup (clears client-side tokens) and advises the user to close the browser tab for maximum security. The session may remain valid on server-side until its natural expiration.

Table 3.4.7 Table for Use Case Specification 37

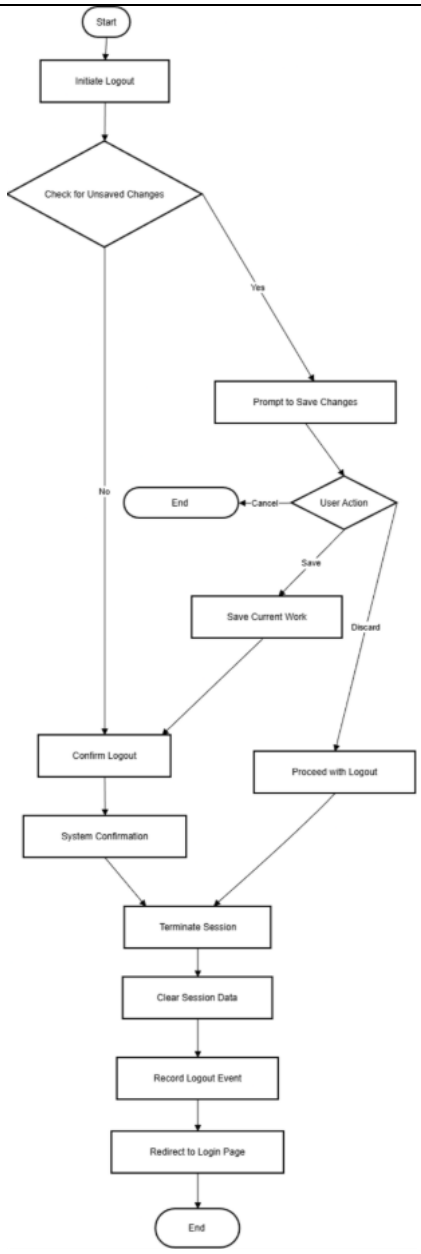

Diagram 3.4.7 Flow Chart UC37

4 System Models

4.1 Business Rules

1. One Admin should have one Account / One Account should belongs to one Admin
2. One Student shall generate many MoodAlerts / One MoodAlert is generated by one Student
3. One Admin shall review many MoodAlerts / One MoodAlert is reviewed by one Admin
4. One Admin shall create many Assignments / One Assignment is created by one Admin

5. One Student is assigned to one Counselor / One Counselor is assigned to at most 20 Student.
6. One Admin shall review many Reports / One Report is reviewed by one Admin
7. One Student shall have many ScoreTransactions / One ScoreTransaction belongs to one Student
8. One Student shall have at most 3 Restrictions / One Restriction belongs to one Student
9. One Student may submit many Appeals / One Appeal is submitted by one Student
10. One Admin shall review many Appeals / One Appeal is reviewed by one Admin
11. One Admin must issue many Suspensions / One Suspension is issued by one Admin
12. One Student shall register one Account / One Account shall be registered by one Student
13. One Account shall initiate many LoginSessions / One LoginSession shall be initiated by one account
14. One Student may request many PasswordResets / One PasswordReset may be requested by one Student
15. One Account shall initiate many LogoutSessions / One LogoutSession shall be initiated by one Account
16. One Student may submit many Reports / One Report may be submitted by one Student
17. One Content may receive many Reports / One Report may be about one Content
18. One Student shall have one Profile / One Profile shall belong to one Student
19. One Student shall submit many MoodCheckIns / One MoodCheckIn shall belong to one Student
20. One Student may have many Friendships / One Friendship may connect two Students
21. One Student may participate in many PrivateChats / One PrivateChat may involve two Students
22. One Student may create many Posts / One Post may be created by one Student
23. One Student may create many Comments / One Comment may be created by one Student
24. One Student may create many Likes / One Like may be created by one Student
25. One Post may receive many Likes / One Like may be for one Post or Comment
26. One Student may match with many Students / One PeerMatch may be connected two Students
27. One Student may book many CounselorAppointments / One CounselorAppointment may belong to one Student
28. One Moderator should have one Account / One Account should belong to one Moderator.
29. One Moderator shall review many Reported Messages / One Reported Message is reviewed by one Moderator.
30. One Moderator may flag many Student Accounts / One Student Account ban is flagged by one Moderator.
31. One Moderator shall create many Announcements / One Announcement is created by one Moderator
32. One Moderator shall delete many Forum Posts / One Forum Post deletion is performed by one Moderator.
33. One Moderator shall delete many Forum Comments / One Forum Comment deletion is performed by one Moderator.
34. One Counselor has one Account / One Account belongs to one Counselor

35. One Counselor manages many Students / One Student is managed by one Counselor
36. One Counselor generates many DataExports / One DataExport is generated by one Counselor
37. One Counselor verifies many StudentAssignments / One StudentAssignment is verified by one Counselor
38. One Counselor manages many TherapeuticPlans / One TherapeuticPlan is managed by one Counselor
39. One Counselor schedules many Appointments / One Appointment is scheduled by one Counselor
40. One Counselor terminates one Session / One Session is terminated by one Counselor

4.2 ERD

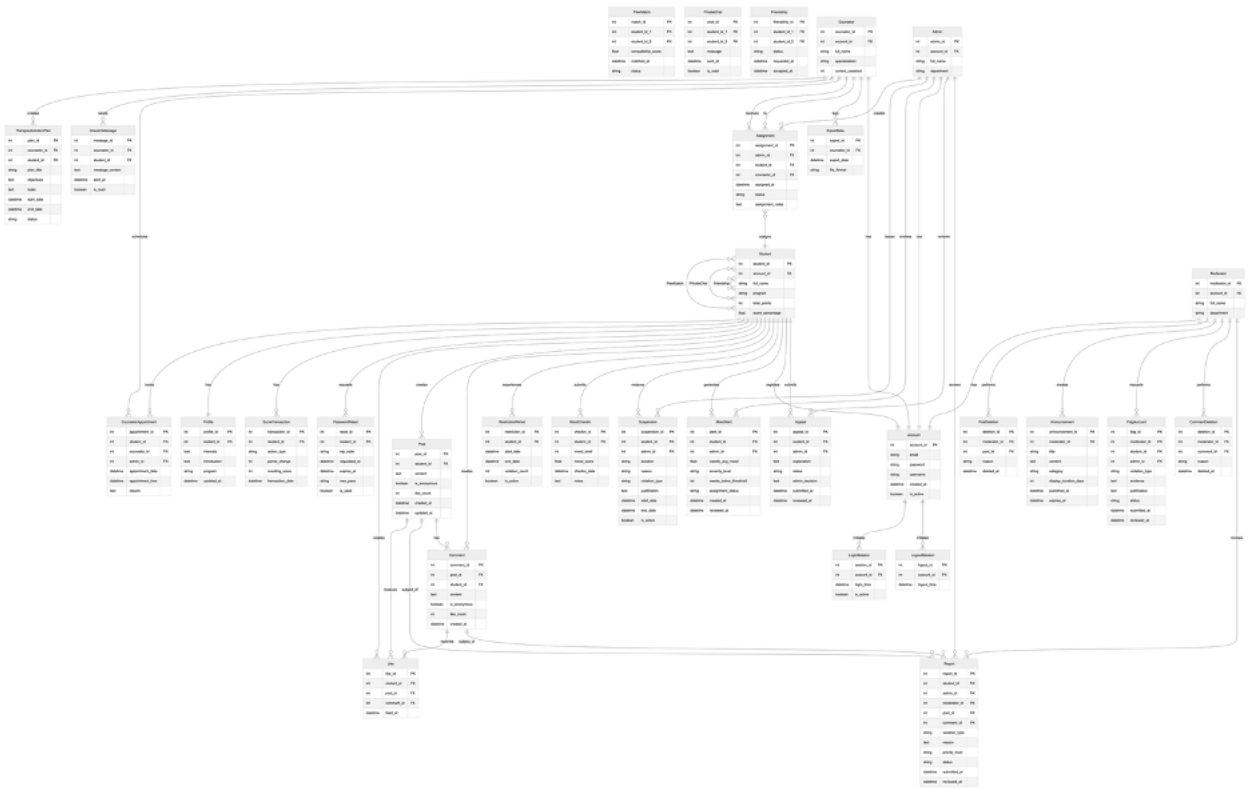

Diagram 4.2.1 Entity-Relationship Diagram (ERD)

4.3 Class Diagram

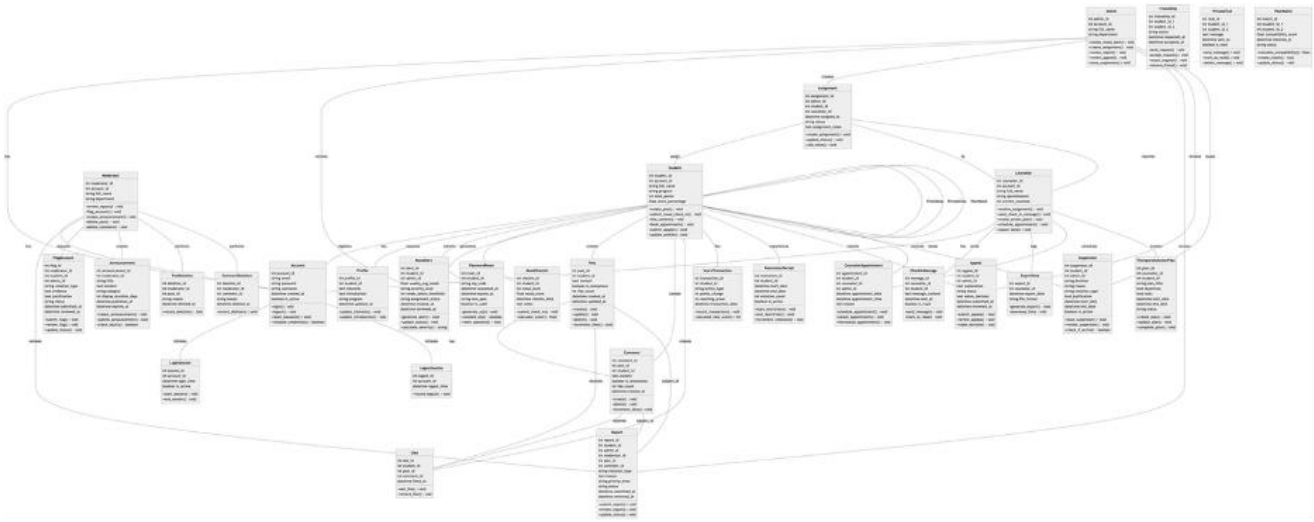

Diagram 4.3.1 Class Diagram

4.4 Classes / Entities

Table Name	Attribute Name	Data Type	Key / Constraint	Description
Account	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).

Admin	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.
Student	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_percentage	FLOAT		Overall engagement or health score percentage.
Moderator	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.
Counselor	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.
Profile	profile_id	INT	PK	Unique identifier for the profile.
	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.
LoginSession	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.
LogoutSession	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.
PasswordReset	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.

MoodCheckIn	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.
MoodAlert	alert_id	INT	PK	Unique identifier for the alert.
	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.
Post	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.
Comment	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.
Like	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.
	liked_at	DATETIME		Timestamp when the like occurred.
Report	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.
Assignment	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.
Appeal	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.
Suspension	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.
ScoreTransaction	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.
RestrictionPeriod	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.
CounselorAppointment	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).
FlagAccount	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.
Announcement	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.
	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.
PostDeletion	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.
CommentDeletion	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.
CheckInMessage	message_id	INT	PK	Unique identifier for the message.
	counselor_id	INT	FK	Reference to the Counselor sending.
	student_id	INT	FK	Reference to the Student receiving.
	message_content	TEXT		Content of the check-in message.
	sent_at	DATETIME		Timestamp when sent.

	is_read	BOOLEAN		Flag if the student has read the message.
ExportData	export_id	INT	PK	Unique identifier for the export data.
	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.
TherapeuticActionPlan	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).
Friendship	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).
	requested_at	DATETIME		Timestamp of request.
	accepted_at	DATETIME		Timestamp of acceptance.
PrivateChat	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.
PeerMatch	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).

Table 4.4.1 Classes / Entities

5 Non-Functional Requirements

Privacy Requirement (Anonymous Posting & Commenting)
The system shall protect the privacy of all users by ensuring that any content posted or commented is given choice anonymously or non-anonymously in the forum, and cannot be traced back to the user by other students and moderators, or counselors if the post or comment is anonymously. User identity related to anonymous actions must be securely stored, and accessible only to system administrators strictly for safety, legal, or misuse investigations.

6 References

1. Visual Paradigm. (n.d.). What is a user story? https://www.visual-paradigm.com/guide/agile-software-development/what-is-user-story/
2. GeeksforGeeks. (n.d.). Use case diagram https://www.geeksforgeeks.org/system-design/use-case-diagram/
3. Visual Paradigm. (n.d.). What is use case specification? https://www.visual-paradigm.com/guide/use-case/what-is-use-case-specification/
4. New York University. (n.d.). Use case template (PDF). https://cs.nyu.edu/~jcf/classes/CSCI-GA.2440001_sp18/handouts/UseCaseTemplate.pdf
5. QBI Institute. (n.d.). Use case template (PDF). https://www.qbi.in/uploads/5/5/9/0/55906279/use_case_template-1.pdf
6. Visual Paradigm. (n.d.). What is a class diagram? https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-class-diagram/
7. EdrawSoft. (n.d.). System architecture diagram https://www.edrawsoft.com/architecture-diagram.html#system
8. GeeksforGeeks. (n.d.). How to draw architecture diagrams https://www.geeksforgeeks.org/system-design/how-to-draw-architecture-diagrams/
9. Visual Paradigm. (n.d.). What is a deployment diagram? https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-deployment-diagram/
10. Lucid Software Inc. (n.d.). UML deployment diagram https://www.lucidchart.com/pages/uml-deployment-diagram
11. Visual Paradigm. (n.d.). Sequence diagram tutorial https://online.visual-paradigm.com/diagrams/tutorials/sequence-diagram-tutorial/
12. Visual Paradigm. (n.d.). Sequence diagram https://www.visual-paradigm.com/learning/handbooks/software-design-handbook/sequence-diagram.jsp
13. Digital Peer Support. (n.d.). Digital peer support https://digitalpeersupport.org/