

# Tutor Support System at HCMUT

Assignment's Specification

Ho Chi Minh City University of Technology (HCMUT)  
Faculty of Computer Science and Engineering

September 2025



*Version 1.0*

Instructor: Dr. Truong Thi Thai Minh

Student:	Tran Huu Hoang Long	2352704
	Doan Anh Khoi	2352601
	Arthur Bardot	2460080
	Nguyen Manh Quoc Khanh	2352525
	Pham Quang Tuan	2353275
	Phan Ngoc Lan Chi	2352137
	Truong Quoc Thai	2353094

*This specification follows the structure and conventions used in prior course specifications, adapted to the Tutor Support System context.*

# Contents

<b>1 Assignment's outcome</b>	<b>3</b>
<b>2 Introduction</b>	<b>3</b>
<b>3 Description</b>	<b>3</b>
3.1 Stakeholders, Roles, and Expectations . . . . .	3
3.1.1 Stakeholders . . . . .	3
3.1.2 Roles . . . . .	3
3.1.3 Expectations . . . . .	3
3.2 Objectives and Scope . . . . .	3
<b>4 Functional Requirements</b>	<b>4</b>
4.1 Functional Requirements List . . . . .	4
4.2 User & Information Management (FR-UM) . . . . .	4
4.3 Tutor–Student Matching (FR-MAT) . . . . .	5
4.4 Session & Scheduling Management (FR-SCH) . . . . .	6
4.5 Feedback & Progress Tracking (FR-FBK) . . . . .	7
4.6 Reporting & Analytics (FR-RPT) . . . . .	7
4.7 Integration with HCMUT Infrastructure (FR-INT) . . . . .	8
4.8 Advanced / Optional Features (FR-ADV) . . . . .	9
4.9 Non-interactive Functional Requirements (FR-NI) . . . . .	9
<b>5 Non-Functional Requirements</b>	<b>11</b>
5.1 Performance Requirements . . . . .	11
5.2 Security & Reliability Requirements . . . . .	11
5.3 Usability & Accessibility Requirements . . . . .	11
5.4 Software Quality Attributes . . . . .	11
5.5 Business Rules . . . . .	12
<b>6 Use-Case View</b>	<b>12</b>
6.1 General Use-Case Diagram . . . . .	12
6.2 UC-01 Log in & Profile Management . . . . .	13
6.3 UC-02 Tutor-Student Matching . . . . .	15
6.4 UC-03 Session Scheduling Management . . . . .	17
6.5 UC-04 Feedback & Progress Tracking . . . . .	20
6.6 UC-05 Reporting & Analytics . . . . .	24
6.7 UC-06 Integration with HCMUT Infrastructure . . . . .	28
6.8 UC-07 Advanced / Optional Features . . . . .	31
<b>7 Sequence Diagram</b>	<b>33</b>
7.1 SQ-01 Log in, Log out & Profile Management . . . . .	33
7.2 SQ-02 Tutor-Student Matching . . . . .	34
7.3 SQ-03 Session Scheduling Management . . . . .	34
7.4 SQ-04 Feedback & Progress Tracking . . . . .	37
7.5 SQ-05 Reporting & Analytics . . . . .	40

<b>8 Activity Diagram</b>	<b>41</b>
8.1 AC-01 Log in, Log out & Profile Management . . . . .	41
8.2 AC-02 Tutor-Student Matching . . . . .	42
8.3 AC-03 Session Scheduling Management . . . . .	43
8.4 AC-04 Feedback & Progress Tracking . . . . .	47
8.5 AC-05 Reporting & Analytics . . . . .	49
<b>9 State Diagram</b>	<b>50</b>
9.1 SD-01 Log in, Log out & Profile Management . . . . .	50
9.2 SD-02 Tutor-Student Matching . . . . .	51
9.3 SD-03 Session Scheduling Management . . . . .	51
9.4 SD-04 Reporting & Analytics . . . . .	52
<b>10 Mockup</b>	<b>53</b>
10.1 MU-01 Log in, Log out & Profile Management . . . . .	53
10.2 MU-02 Tutor-Student Matching . . . . .	56
10.3 MU-03 Session Scheduling Management . . . . .	59
10.4 MU-04 Feedback & Progress Tracking . . . . .	62
10.5 MU-05 Reporting & Analytics . . . . .	63
10.6 MU-06 Integration with HCMUT Infrastructure . . . . .	64
10.7 MU-07 Study & Advanced Features . . . . .	65
10.8 MU-08 Admin Tools . . . . .	72
<b>11 System Integration</b>	<b>75</b>
<b>12 Coding Rules and Constraints</b>	<b>75</b>
<b>13 Submission and Deliverables</b>	<b>75</b>
<b>14 Other Regulations</b>	<b>75</b>
<b>15 Changelog</b>	<b>75</b>

# 1 Assignment's outcome

Upon completion of this assignment, students will be able to:

- Identify stakeholders, roles, objectives, and scope for a university-scale information system.
- Specify functional and non-functional requirements clearly and traceably.
- Model core use-cases for tutoring workflows (registration, matching, booking, feedback).
- Define system integration touchpoints with HCMUT infrastructure (SSO, Data-Core, Library).
- Produce a consistent, submission-ready specification document in L<sup>A</sup>T<sub>E</sub>X.

## 2 Introduction

HCMUT requires a platform to support students in their academic and skills development journey. The **Tutor Support System (TSS)** will modernize the management of the Tutor/Mentor program, enabling scalable operations and data-driven improvement across departments.

## 3 Description

### 3.1 Stakeholders, Roles, and Expectations

#### 3.1.1 Stakeholders

Stakeholders are individuals or entities who can affect or are affected by the project outcomes.

**Primary:** HCMUT staff (customer), course instructors (project managers), development team, designers.

**Secondary:** Students (end-users), tutors, government, competitors.

Stakeholders may be internal (within HCMUT: Office of Academic Affairs, Office of Student Affairs, departments) or external (students, tutors).

#### 3.1.2 Roles

Roles are permissions and capabilities within the system: Student, Tutor, Coordinator, Department Chair, Program Administrator.

#### 3.1.3 Expectations

Each role expects secure access, clear workflows, and reliable performance. Students expect easy registration and booking; tutors expect manageable scheduling; administrators expect robust reporting.

### 3.2 Objectives and Scope

**Objectives** Design and develop an efficient, secure, and scalable software supporting the Tutor/Mentor program:

- Manage tutor/student information (profiles, expertise, support needs).
- Enable registration, selection or automated matching.
- Support scheduling, booking, cancellation, rescheduling (online or in-person).
- Provide progress tracking, feedback, evaluation.
- Generate reports for departments and offices to optimize resources and recognition.
- Integrate with HCMUT SSO/DataCore/Library for consistency and security.

**Scope** This specification covers:

- Core features: profile management, matching, scheduling, notifications, feedback, reporting.
- Integrations: HCMUT\_SSO (auth), HCMUT\_DATACORE (personal data), HCMUT\_LIBRARY (learning resources).
- Roles: Student, Tutor, Coordinator, Department Chair, Program Admin (RBAC).

**Out of Scope (MVP):** Full production DB, advanced AI features (smart matching, personalization), external integrations beyond HCMUT.

## 4 Functional Requirements

### 4.1 Functional Requirements List

The following table summarizes the functional requirements of the Tutor Support System. Requirements are grouped into thematic categories to ensure clarity and traceability.

**Prioritization Method:** In this project, we applied the MoSCoW prioritization technique to classify functional requirements. This method categorizes requirements into four levels:

- Must: Essential for the system to function; without them, the system fails to meet its objectives.
- Should: Important but not vital; the system can still operate without them in the first release.
- Could: Desirable enhancements that improve usability or efficiency if time/resources allow.
- Won't (this time): Explicitly excluded from the current scope, possibly considered for future releases.

This approach ensures clarity in requirement importance and helps manage project scope effectively.

### 4.2 User & Information Management (FR-UM)

#### • FR-UM.01 – Profile

- *Description:* The system shall allow students and tutors to view and update their personal profiles, with core information (name, student ID, email, role, faculty/major) synchronized from the university's database.
- *Acceptance Criteria:*

- \* Profiles automatically include core fields (name, student ID, email, role, faculty/major) synced from the university database; users are not required to manually enter these fields.
- \* Profile changes are timestamped and stored.
- *Priority:* Must
- **FR-UM.02 – Role-based Access Control**
  - *Description:* The system shall enforce role-based access control to regulate permissions for students, tutors, coordinators, department heads, and administrators.
  - *Acceptance Criteria:*
    - \* Each role has defined permissions.
    - \* Unauthorized access attempts are logged.
  - *Priority:* Must

#### **4.3 Tutor–Student Matching (FR-MAT)**

- **FR-MAT.01 – Manual Tutor Selection**
  - *Description:* The system shall allow students to register for the tutoring program.
  - *Acceptance Criteria:*
    - \* Students can successfully submit a registration request to join the tutoring program.
  - *Priority:* Must
- **FR-MAT.02 – Manual Tutor Selection**
  - *Description:* The system shall allow students to search for and manually select tutors based on expertise, availability, and preferences. Core tutor information (subject, department, schedule) is synchronized from the university database, while teaching preferences are provided by tutors.
  - *Acceptance Criteria:*
    - \* Students can filter tutors by at least three criteria (e.g., subject, availability, preferences).
    - \* Selection creates a pending match awaiting tutor confirmation.
  - *Priority:* Must
- **FR-MAT.03 – Smart Matching**
  - *Description:* The system shall provide automated tutor–student matching using predefined criteria such as subject, availability, and tutor workload. Matching relies on synchronized data from DATACORE combined with tutor-specified preferences.
  - *Acceptance Criteria:*
    - \* System generates a ranked list of tutors with explanation of matching factors.
    - \* Confirmation from both tutor and student finalizes the match.

- *Priority*: Should
- **FR-MAT.04 – Coordinator Assignment**
  - *Description*: The system shall allow coordinators, department chairs, or administrators to manually assign tutors to students when necessary, overriding automated or student-selected matches.
  - *Acceptance Criteria*:
    - \* Only authorized roles can assign tutors.
    - \* Manual assignment overrides previous matches.
    - \* Assignment details (who, when, reason) are logged and traceable.
  - *Priority*: Must

#### **4.4 Session & Scheduling Management (FR-SCH)**

- **FR-SCH.01 – Tutor Availability**
  - *Description*: The system shall allow tutors to set and manage their availability for consultation sessions, synchronized with official university timetables where applicable.
  - *Acceptance Criteria*:
    - \* Only tutors can create, edit, and delete available slots.
    - \* The system prevents overlapping slots.
    - \* Slots cannot conflict with official class schedules imported from DATA-CORE.
  - *Priority*: Must
- **FR-SCH.02 – Session Booking**
  - *Description*: The system shall allow students to book in-person or online sessions with tutors based on available slots.
  - *Acceptance Criteria*:
    - \* Booking is allowed only within available tutor slots.
    - \* The system prevents double-booking of the same slot.
  - *Priority*: Must
- **FR-SCH.03 – Session Modification**
  - *Description*: The system shall allow students to cancel or reschedule booked sessions.
  - *Acceptance Criteria*:
    - \* Cancellation and rescheduling must follow configured rules (e.g., at least 2 hours before session start).
    - \* The system ensures new booking adheres to availability and no conflicts.
  - *Priority*: Must
- **FR-SCH.04 – Notifications & Reminders**
  - *Description*: The system shall automatically send notifications and reminders for upcoming sessions or schedule changes.

- *Acceptance Criteria:*
  - \* Notification sent immediately upon booking, cancellation, or reschedule.
  - \* Reminder sent at least 24h and 1h before session start.
- *Priority:* Must

#### **4.5 Feedback & Progress Tracking (FR-FBK)**

- **FR-FBK.01 – Session Feedback**
  - *Description:* The system shall enable students to provide structured feedback for each completed session.
  - *Acceptance Criteria:*
    - \* Feedback form is available only after session completion.
    - \* Each student can submit one feedback entry per session.
    - \* Feedback is linked to session ID and timestamped.
  - *Priority:* Must
- **FR-FBK.02 – Progress Recording**
  - *Description:* The system shall allow tutors to record mentee progress and generate optional summaries after sessions.
  - *Acceptance Criteria:*
    - \* Only tutors can log progress, which is linked to session ID.
    - \* Summaries may include text notes and optional attachments.
    - \* All records are timestamped and stored for reporting.
  - *Priority:* Should

#### **4.6 Reporting & Analytics (FR-RPT)**

- **FR-RPT.01 – Departmental Reports**
  - *Description:* The system shall generate reports for academic departments to monitor student learning performance.
  - *Acceptance Criteria:*
    - \* Reports include attendance, performance indicators, and session counts.
    - \* Data exportable to CSV/PDF.
  - *Priority:* Should
- **FR-RPT.02 – Academic Affairs Overview**
  - *Description:* The system shall provide overview reports for the Office of Academic Affairs to optimize resource allocation.
  - *Acceptance Criteria:*
    - \* Reports show tutor workload distribution and student demand trends.
    - \* Dashboards update with latest synced data.
  - *Priority:* Should
- **FR-RPT.03 – Student Affairs Outcomes**

- *Description:* The system shall provide summarized participation data for the Office of Student Affairs to support training credits and scholarship considerations.
- *Acceptance Criteria:*
  - \* Reports list eligible students based on configured rules.
  - \* Calculation rules are transparent and logged.
- *Priority:* Should

## 4.7 Integration with HCMUT Infrastructure (FR-INT)

- **FR-INT.01 – HCMUT\_SSO Integration**
  - *Description:* The system shall integrate with HCMUT\_SSO for unified authentication.
  - *Acceptance Criteria:*
    - \* Only valid SSO accounts can log in.
    - \* Single sign-out follows HCMUT SSO rules.
  - *Priority:* Must
- **FR-INT.02 – DATACORE Synchronization**
  - *Description:* The system shall synchronize core personal and academic data from HCMUT\_DATACORE.
  - *Acceptance Criteria:*
    - \* Sync occurs periodically or near real-time.
    - \* Conflicts resolved with DATACORE as source of truth.
  - *Priority:* Must
- **FR-INT.03 – Role assignment**
  - *Description:* The system shall automatically assign roles (student, tutor, coordinator, department chair, administrator) based on centralized HCMUT role data.
  - *Acceptance Criteria:*
    - \* System assigns roles upon login via SSO.
    - \* Role updates in DATACORE are reflected in the system within defined sync intervals.
  - *Priority:* Must
- **FR-INT.04 – Library Resource Linking**
  - *Description:* The system shall connect with HCMUT\_LIBRARY to allow tutors and students to share relevant materials.
  - *Acceptance Criteria:*
    - \* Users can attach library resources to sessions or summaries.
    - \* Access permissions follow HCMUT library rules.
  - *Priority:* Could

## 4.8 Advanced / Optional Features (FR-ADV)

These features are not mandatory for the MVP but can enhance the Tutor Support System if resources permit:

- **FR-ADV.01 – Intelligent Matching (AI Integration)**

- *Description*: The system may leverage AI techniques to optimize tutor-student pairing by analyzing multiple factors such as performance history, learning style, and tutor workload.
- *Acceptance Criteria*:
  - \* AI suggestions ranked with justification.
  - \* Users can compare AI suggestion with manual choice.
- *Priority*: Optional

- **FR-ADV.02 – Online Community Platform**

- *Description*: The system may provide a forum or community space where tutors and students can exchange resources, discuss topics, and collaborate outside formal sessions.
- *Acceptance Criteria*:
  - \* Users can create discussion threads and share files.
  - \* Moderation tools available for coordinators.
- *Priority*: Optional

- **FR-ADV.03 – Personalized Learning Support**

- *Description*: The system may use AI-driven recommendations to suggest learning materials, exercises, or tutoring approaches tailored to individual students.
- *Acceptance Criteria*:
  - \* Recommendations adapt to student's history and feedback.
  - \* Users can accept or reject suggestions.
- *Priority*: Optional

- **FR-ADV.04 – Multi-Program Tutoring**

- *Description*: The system may support both academic tutoring (courses, skills) and non-academic mentoring (career guidance, soft skills).
- *Acceptance Criteria*:
  - \* System allows defining and tracking multiple tutoring program types.
  - \* Reports distinguish between academic and non-academic activities.
- *Priority*: Optional

## 4.9 Non-interactive Functional Requirements (FR-NI)

- **FR-NI-01 – Automatic Notifications**

- *Description*: The system shall automatically send confirmation, reminder, and cancellation/rescheduling notifications to students and tutors without manual intervention.
- *Priority*: Must

- **FR-NI-02 – DataCore Sync**

- *Description:* The system shall periodically synchronize personal data (name, ID, faculty, email, status) from HCMUT\_DATACORE.
- *Trigger:* Hourly schedule + on change-webhook
- *Output:* Up-to-date user records with change log
- *Acceptance Criteria:*  $\geq 99\%$  updates reflected within 10 minutes of source change.
- *Priority:* Must

- **FR-NI-03 – Automatic Inactive Detection**

- *Description:* The system should detect logged-in accounts with no activity for a specific period of time and log them out to maintain stability and security.
- *Acceptance Criteria:*
  - \* The system saves the state of the inactive account before logging out.
  - \* The threshold for inactive time is dynamic, depending on the state of the system.
- *Priority:* Optional

- **FR-NI-04 – Scheduled Database Cleanup**

- *Description:* The system shall automatically perform database cleanup on a scheduled basis, removing obsolete temporary data, expired logs, and error records to maintain storage efficiency and system performance.
- *Acceptance Criteria:*
  - \* Temporary data and logs older than 12 months are automatically deleted or archived.
  - \* Cleanup runs during off-peak hours to avoid disruption.
  - \* A cleanup summary report (records removed, storage freed) is logged.
  - \* Cleanup failures trigger an alert for administrators.
- *Priority:* Should

- **FR-NI-05 – Disaster Recovery & Backup**

- *Description:* The system shall maintain automated backup and disaster recovery mechanisms to ensure data resilience and continuity in case of failure or outage.
- *Acceptance Criteria:*
  - \* Full backups daily; incremental backups every 15 minutes.
  - \* Backups encrypted and stored in two geographically separate locations.
  - \* RPO  $\leq 15$  minutes, RTO  $\leq 4$  hours.
  - \* Backup integrity verified after each operation.
  - \* Failed backups trigger administrator alerts.
- *Priority:* Should

- **FR-NI-06 – Attendance Logging**

- *Description:* The system shall automatically mark attendance when a student

joins an online tutoring session via the platform.

- *Acceptance Criteria:*
  - \* Attendance log created within 1 minute of session start.
  - \* Logs include session ID, student ID, timestamp.
- *Priority:* Should

## 5 Non-Functional Requirements

### 5.1 Performance Requirements

- NFR1: Concurrent Users: The system shall handle at least **1000 concurrent users** without degradation, verified by load testing.
- NFR2: Response Time: 95% of key actions (login, enrollment, course access, page load) shall complete within **3 seconds** under normal load.
- NFR3: Real-Time Notifications: Notifications (reminders, announcements, deadlines) shall be delivered within **2 seconds** of the triggering event in 95% of cases.

### 5.2 Security & Reliability Requirements

- NFR4: Data Encryption: All personal and academic data shall be encrypted at rest with **AES-256** and in transit with **TLS 1.2 or higher**.
- NFR5: Access Control & Logging: 100% of authentication and access attempts shall be logged and retained for at least **90 days**.
- NFR6: Uptime: The system shall provide at least **99% uptime** during the academic year (excluding scheduled maintenance), monitored monthly.
- NFR7: Data Integrity: Recovery testing shall confirm **zero data loss** during transient failures; operations shall be idempotent.

### 5.3 Usability & Accessibility Requirements

- NFR8: User Interface: Usability testing (System Usability Scale) shall achieve a score of at least **80/100**.
- NFR9: Task Simplicity: Key workflows (course enrollment, submission, reschedule) shall require **no more than 3 steps**.
- NFR10: Multi-Platform Support: The system shall function correctly on at least **3 major browsers** (Chrome, Firefox, Edge) and on **desktop, tablet, and smartphone**.
- NFR11: Accessibility Standards: The system shall meet **WCAG 2.1 AA** accessibility compliance.

### 5.4 Software Quality Attributes

- NFR12: Scalability: The system shall support a **50% increase in users** beyond baseline load without major reconfiguration.

- NFR13: Maintainability: The codebase shall achieve at least **80% unit test coverage** and pass linting with no critical errors.
- NFR14: Extensibility: Adding a new feature (e.g., adaptive learning, AI tutor matching) shall require no more than **2 person-weeks** of integration effort.
- NFR15: Observability: The system shall provide logs covering **100% of authentication and enrollment events** and collect usage metrics (active users, completed courses) with **5-minute granularity**.

## 5.5 Business Rules

- NFR16: Legal Compliance: The system shall comply with Vietnamese data protection laws and pass an annual **compliance audit with zero critical findings**.

# 6 Use-Case View

## 6.1 General Use-Case Diagram

Placeholder for diagram:

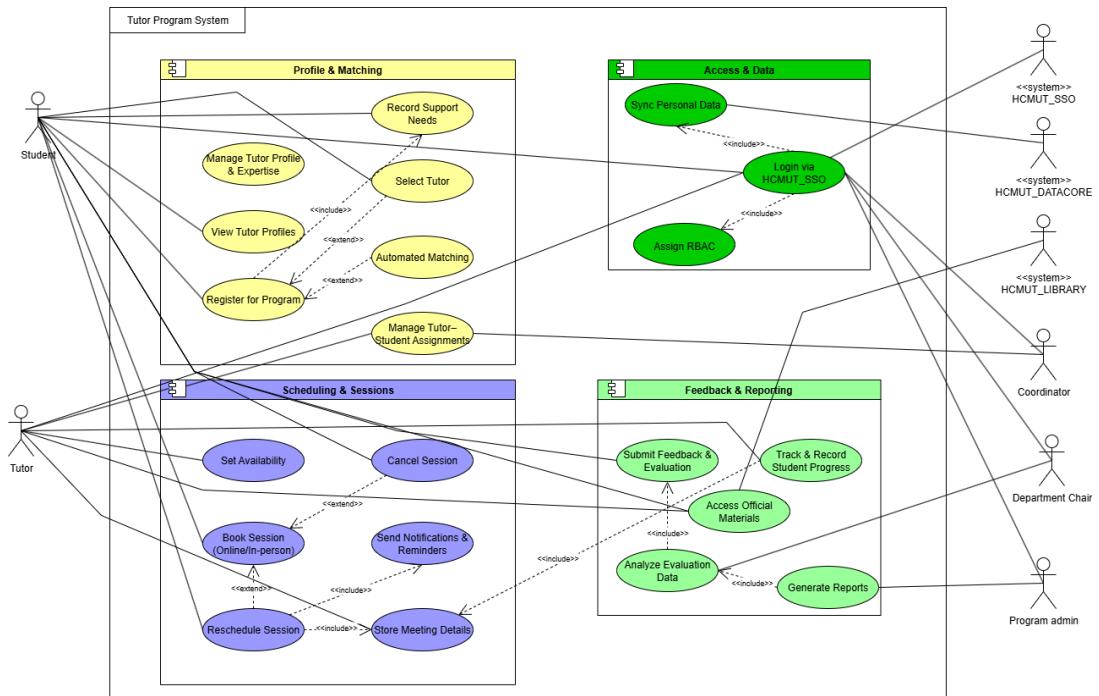
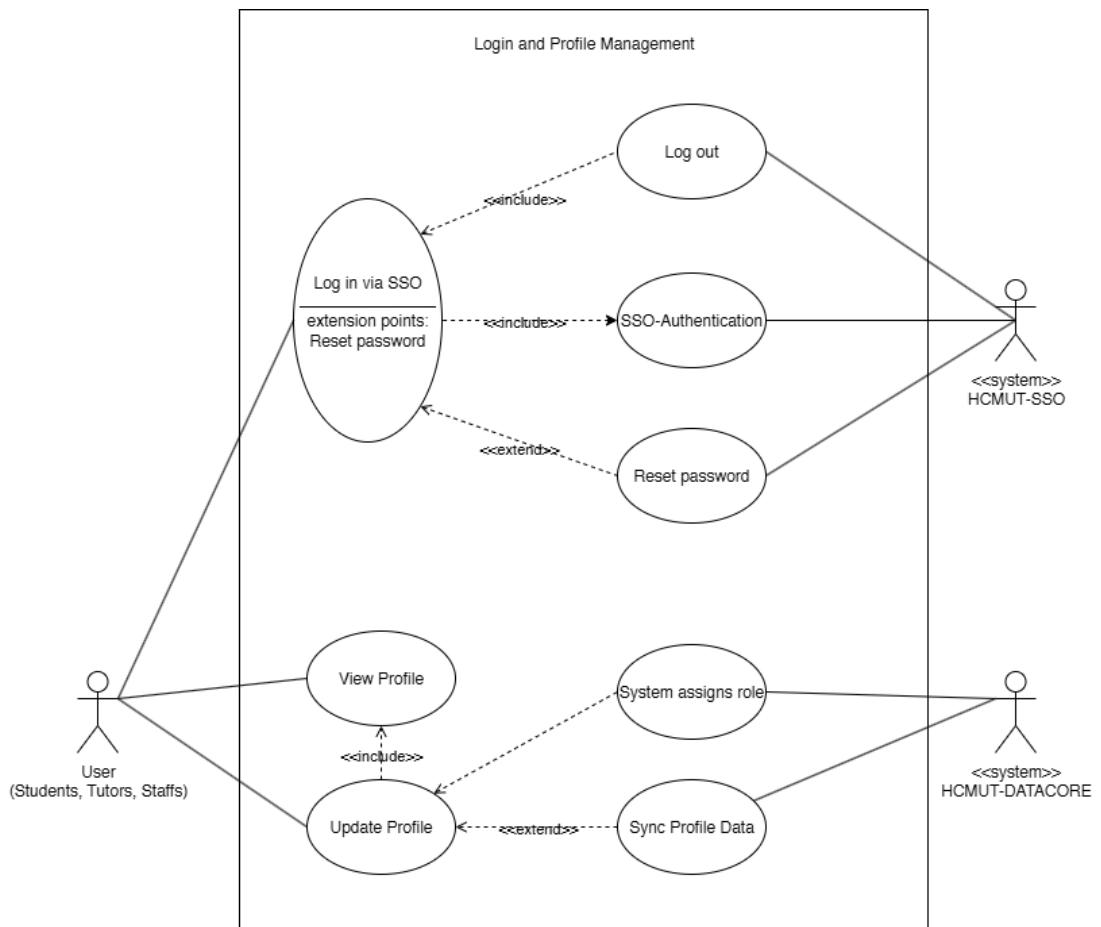


Figure 1: General Tutor program system

## 6.2 UC-01 Log in & Profile Management



**Figure 2:** Log in and Manage Profile

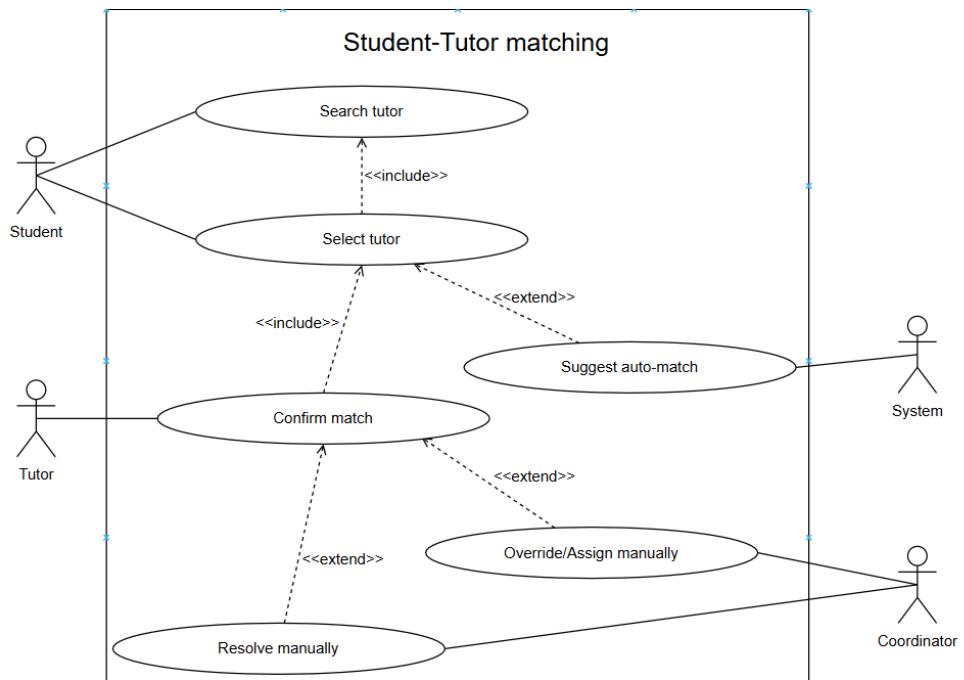
<b>Use-case ID</b>	UC-01a
<b>Use-case name</b>	Login via SSO
<b>Use-case overview</b>	To allow students, tutors, and staff to log in securely via HCMUT_SSO with role assignment handled by the system.
<b>Actors</b>	User (Students, Tutors, Staff), HCMUT_SSO, System
<b>Preconditions</b>	<ol style="list-style-type: none"> <li>1. User has valid SSO credentials.</li> <li>2. HCMUT_SSO service is available.</li> </ol>
<b>Trigger</b>	User selects the “Login via SSO” option.
<b>Steps</b>	<ol style="list-style-type: none"> <li>1. User initiates login via HCMUT_SSO.</li> <li>2. System sends authentication request to SSO service.</li> <li>3. HCMUT_SSO validates credentials.</li> <li>4. On success, the system fetches user data and assigns role.</li> <li>5. On failure, the system denies access.</li> </ol>
<b>Postconditions</b>	User is authenticated and session established; role assignment is ready for system use.
<b>Alternative Flows</b>	A1: Invalid login → Access denied with error message. A2: Role update in DATACORE synced during login.
<b>Exception Flow</b>	<ol style="list-style-type: none"> <li>1. SSO service unavailable → System shows maintenance/unavailable message.</li> <li>2. Network error → User prompted to retry login.</li> </ol>

Table 1: Use Case UC-01a: Login via SSO

<b>Use-case ID</b>	UC-01b
<b>Use-case name</b>	Profile Management
<b>Use-case overview</b>	To allow students and tutors to view and update their profiles, with core data synchronized from HCMUT_DATACORE.
<b>Actors</b>	Student, Tutor, HCMUT_DATACORE, System
<b>Preconditions</b>	<ol style="list-style-type: none"> <li>1. User is authenticated via SSO.</li> <li>2. Profile data exists in DATACORE.</li> </ol>
<b>Trigger</b>	User selects the “View/Update Profile” option.
<b>Steps</b>	<ol style="list-style-type: none"> <li>1. System retrieves profile information from DATACORE.</li> <li>2. User views profile fields (ID, name, email, faculty, role).</li> <li>3. User updates non-core profile details.</li> <li>4. System validates and saves changes.</li> <li>5. System syncs updated data with DATACORE.</li> </ol>
<b>Postconditions</b>	User profile is updated and synchronized with DATACORE; changes are timestamped and logged.
<b>Alternative Flows</b>	A1: DATACORE unavailable → Updates stored locally until sync resumes.
<b>Exception Flow</b>	<ol style="list-style-type: none"> <li>1. Invalid update request → System rejects and shows error.</li> <li>2. Sync conflict with DATACORE → DATACORE treated as source of truth.</li> </ol>

Table 2: Use Case UC-01b: Profile Management

### 6.3 UC-02 Tutor-Student Matching

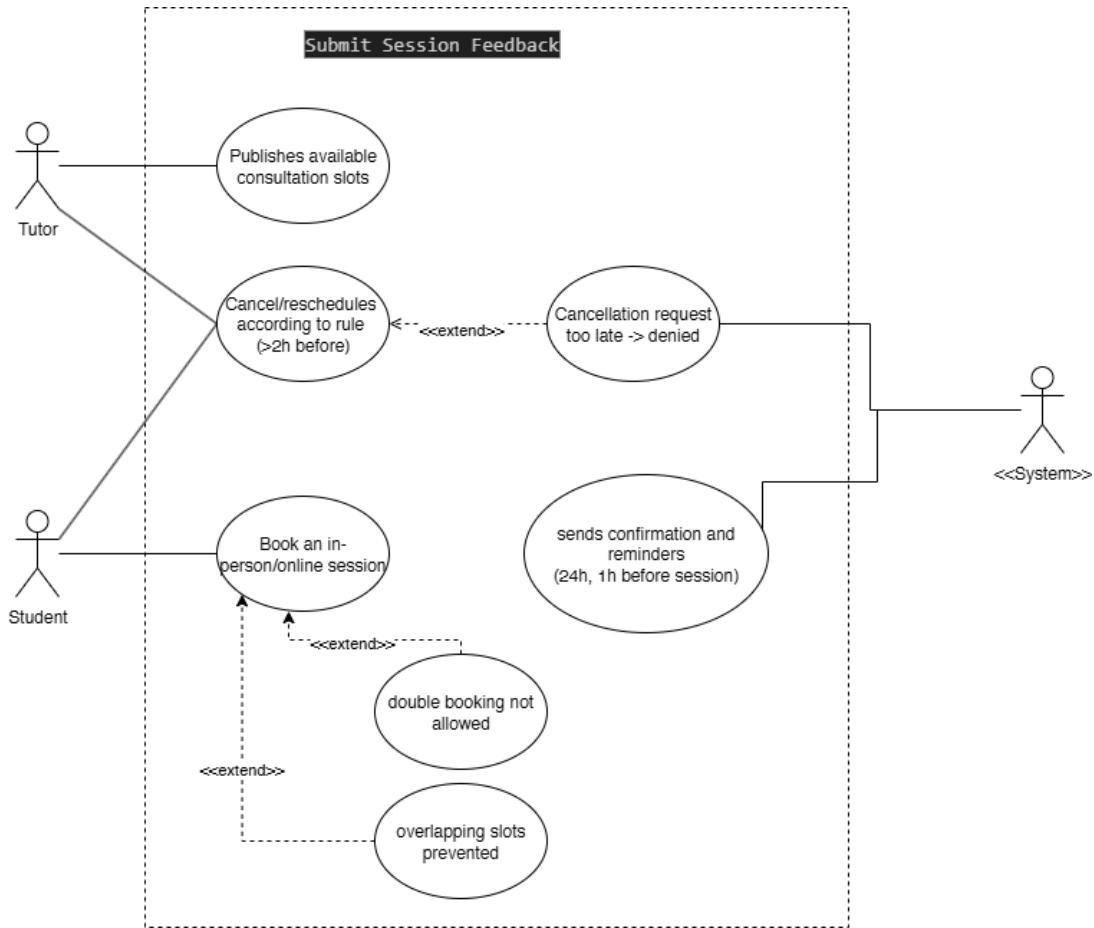


**Figure 3:** Tutor–Student Matching

<b>Use-case ID</b>	UC-02
<b>Use-case name</b>	Tutor–Student Matching
<b>Use-case overview</b>	To allow students to search and select tutors manually or request an automated match, with tutor confirmation and coordinator intervention when necessary.
<b>Actors</b>	Student (primary), Tutor, Coordinator, System
<b>Preconditions</b>	<ol style="list-style-type: none"> <li>1. Student and tutor profiles exist in the system.</li> <li>2. The system is operational and accessible.</li> <li>3. Student is authenticated in the system.</li> </ol>
<b>Trigger</b>	Student initiates a search for tutors or requests an auto-match.
<b>Steps</b>	<ol style="list-style-type: none"> <li>1. Student searches for tutors by subject, availability, or preferences.</li> <li>2. Student selects a tutor; a pending match is created.</li> <li>3. System may suggest an auto-match (ranked list) based on the student's criteria.</li> <li>4. Tutor reviews the pending match and confirms the match.</li> <li>5. If confirmed, the system finalizes and logs the pairing.</li> </ol>
<b>Postconditions</b>	A tutor–student pairing is established and logged in the system.
<b>Alternative Flows</b>	<ol style="list-style-type: none"> <li>1. Auto-match rejected by student or tutor → Coordinator resolves manually.</li> <li>2. No tutors found → System suggests broadening search criteria.</li> <li>3. Tutor does not respond within time limit → Coordinator is notified to assign a tutor.</li> </ol>
<b>Exception Flow</b>	<ol style="list-style-type: none"> <li>1. Network failure prevents search or confirmation (system prompts user to retry).</li> <li>2. Tutor or student profile missing/corrupted → System logs an error and notifies Coordinator.</li> </ol>

Table 3: Use Case UC-02: Tutor–Student Matching

## 6.4 UC-03 Session Scheduling Management



**Figure 4:** Session Scheduling Management

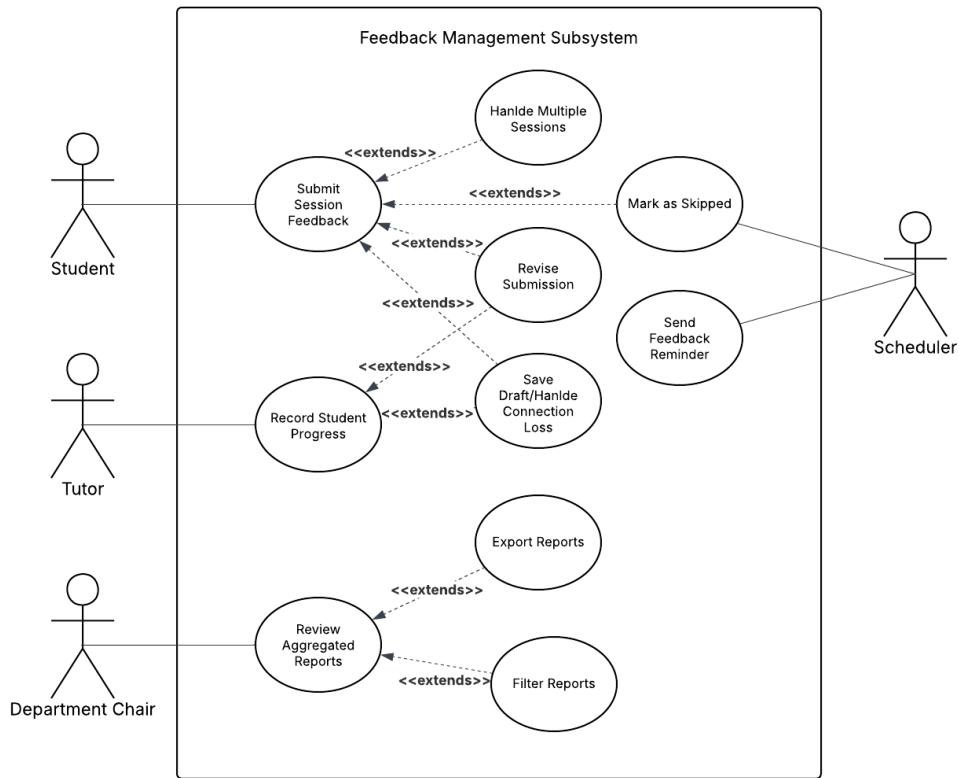
<b>Use-case ID</b>	UC-03a
<b>Use-case name</b>	Publish and Manage Tutor Availability
<b>Overview</b>	To allow a tutor to create, edit, and manage available consultation slots, while the system prevents overlapping sessions and validates scheduling rules.
<b>Actors</b>	Tutor, System
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>• Tutor is authenticated in the system.</li> <li>• The scheduling database is accessible.</li> </ul>
<b>Trigger</b>	Tutor selects “Set Availability” or modifies existing slots.
<b>Main Flow (Steps)</b>	<ol style="list-style-type: none"> <li>1. Tutor publishes available consultation slots.</li> <li>2. System checks for overlapping slots.</li> <li>3. If no conflicts are found, slots are confirmed and saved.</li> <li>4. Tutor may later edit or delete slots before they are booked.</li> </ol>
<b>Postconditions</b>	<ul style="list-style-type: none"> <li>• Valid time slots are stored in the system and visible to students.</li> </ul>
<b>Alternative Flows</b>	AF1: Overlapping slot detected → system rejects slot creation and displays conflict message.
<b>Exception Flows</b>	<ul style="list-style-type: none"> <li>• Database or connection error → slot creation aborted, system logs error.</li> </ul>

Table 4: Use Case UC-03a: Publish and Manage Tutor Availability

<b>Use-case ID</b>	UC-03b
<b>Use-case name</b>	Session Booking, Reminders, and Cancellation Rules
<b>Overview</b>	To allow a student to book an in-person or online session with a tutor, receive automatic confirmations and reminders, and manage cancellations or reschedules according to defined rules.
<b>Actors</b>	Student, Tutor, System
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>• Tutor has published available slots.</li> <li>• Student is authenticated and matched with a tutor.</li> </ul>
<b>Trigger</b>	Student initiates a booking or a cancellation/reschedule request.
<b>Main Flow (Steps)</b>	<ol style="list-style-type: none"> <li>1. Student views tutor's available consultation slots.</li> <li>2. Student selects a preferred slot and submits booking.</li> <li>3. System checks for double booking or slot conflicts.</li> <li>4. Booking is confirmed, and notifications are sent to student and tutor.</li> <li>5. System automatically sends reminders (24h and 1h before the session).</li> <li>6. If cancellation or reschedule is requested <math>\geq 2h</math> before session it is approved.</li> </ol>
<b>Postconditions</b>	<ul style="list-style-type: none"> <li>• Session status updated (booked, rescheduled, or cancelled).</li> <li>• All notifications are logged and delivered.</li> </ul>
<b>Alternative Flows</b>	<p>AF1: Cancellation request too late (<math>&lt;2h</math> before) → denied with reason.</p> <p>AF2: Tutor unavailable → student notified and slot reopened.</p>
<b>Exception Flows</b>	<ul style="list-style-type: none"> <li>• Network or database failure → transaction aborted, retry suggested.</li> </ul>

Table 5: Use Case UC-03b: Session Booking, Reminders, and Cancellation Rules

## 6.5 UC-04 Feedback & Progress Tracking



**Figure 5:** Submit Session Feedback

<b>Use-case ID</b>	UC-04a
<b>Use-case name</b>	Submit Session Feedback
<b>Use-case overview</b>	To allow a student to submit structured feedback after a tutoring session. The feedback is linked to the session and stored for later evaluation and reporting.
<b>Actors</b>	<ul style="list-style-type: none"> <li>• Student (primary)</li> <li>• Scheduler (secondary, for reminders and marking ‘Skipped’)</li> </ul>
<b>Preconditions</b>	<ol style="list-style-type: none"> <li>1. A tutoring session has been completed.</li> <li>2. The system is running and accessible.</li> <li>3. Student is authenticated in the system.</li> </ol>
<b>Trigger</b>	The student clicks the “Submit Feedback” option after the session is completed.

Main Flow (Steps)	<ol style="list-style-type: none"> <li>1. System displays a feedback form linked to the completed session.</li> <li>2. Student fills in and submits the structured feedback.</li> <li>3. System validates the input and saves the feedback in the database.</li> <li>4. System links the feedback to the corresponding session.</li> </ol>
Postconditions	<ul style="list-style-type: none"> <li>• Feedback is stored in the database and linked to the correct session.</li> <li>• If skipped, the system records a “Feedback Skipped” status for that session.</li> <li>• Data is available for tutors and department chairs in aggregated reports.</li> </ul>
Alternative Flows	<ol style="list-style-type: none"> <li>1. <b>Multiple Session Feedback:</b> If multiple sessions are pending, the system displays a list and allows the student to submit feedback sequentially.</li> <li>2. <b>Draft Save/Connection Loss:</b> The student may save incomplete feedback as a draft and return later. System automatically saves the draft if connection is lost before final submission.</li> <li>3. <b>Feedback Revision:</b> Within a defined grace period (e.g., 24h), the student may edit and resubmit feedback; the updated version replaces the original.</li> <li>4. <b>Skipped Feedback:</b> If no feedback is submitted within the allowed time, the Scheduler triggers a process to mark the session’s feedback status as “Skipped.”</li> </ol>
Exception Flows	<ul style="list-style-type: none"> <li>• <b>Database Error:</b> If the system fails to save the final submission due to a database error, it logs the error and prompts the student to retry later.</li> <li>• <b>Missing Session Record:</b> If the session record is missing or corrupted, the system logs an error and notifies the coordinator.</li> </ul>

Table 6: Use Case UC-04a: Submit Session Feedback

Use-case ID	UC-04b
Use-case name	Record Student Progress
Use-case overview	To allow tutors to record student progress and optionally add session summaries for reference and tracking purposes.
Actors	Tutor

<b>Preconditions</b>	1. A tutoring session has been completed. 2. The system is running and accessible. 3. Tutor is authenticated in the system.
<b>Trigger</b>	Tutor accesses the session record and selects “Record Progress.”
<b>Steps</b>	1. Tutor opens the specific session record. 2. Tutor enters progress notes and, optionally, a session summary. 3. Tutor submits the record. 4. System validates the input and saves the record, linking it to the session.
<b>Postconditions</b>	1. Student progress and optional session summary are saved. 2. Data is linked to the session and available for departmental review.
<b>Alternative Flows</b>	A1: Optional Summary → Tutor skips writing a session summary (only progress notes are saved). A2: Draft Save / Connection Loss → Tutor may save incomplete notes as a draft; the system auto-saves the draft if the connection is lost. A3: Note Revision → Within a 24-hour grace period, tutor may edit and resubmit notes; the new version replaces the previous record.
<b>Exception Flow</b>	1. Database error → System logs the issue and notifies the tutor to retry. 2. Missing or locked session → System notifies the tutor and logs the error.
<b>Priority</b>	Should

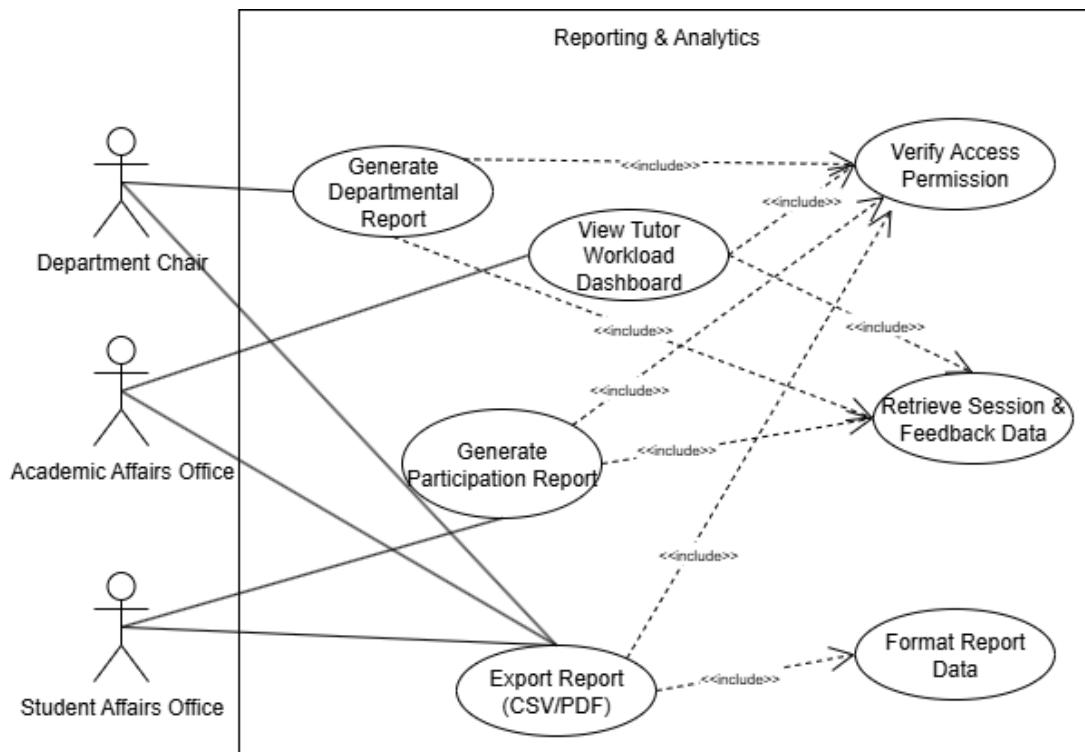
Table 7: Use Case UC-04b: Record Student Progress

<b>Use-case ID</b>	UC-04c
<b>Use-case name</b>	Review Aggregated Reports (Feedback & Progress)
<b>Overview</b>	To allow the department chair to review aggregated feedback and progress data for monitoring tutor performance and program effectiveness.
<b>Actors</b>	Department Chair
<b>Preconditions</b>	1. Feedback and progress records exist in the database. 2. The system is running and accessible. 3. Department Chair is authenticated in the system.
<b>Trigger</b>	Department Chair selects “Aggregated Reports Overview.”

<b>Main Flow (Steps)</b>	<ol style="list-style-type: none"> <li>1. Department Chair selects report criteria (e.g., date range, tutor, course).</li> <li>2. System retrieves all relevant feedback and progress data.</li> <li>3. System processes and displays aggregated statistics, charts, or reports.</li> </ol>
<b>Postconditions</b>	<ul style="list-style-type: none"> <li>• Aggregated data is available for decision-making.</li> <li>• Reports may be exported or stored for institutional use.</li> </ul>
<b>Alternative Flows</b>	<p>AF1 <b>Filtering:</b> Department Chair filters or drills down data by tutor, course, or specific time period.</p> <p>AF2 <b>Export Report:</b> Department Chair exports the report in a chosen format (e.g., PDF, CSV) for external analysis.</p>
<b>Exception Flows</b>	<ul style="list-style-type: none"> <li>• <b>Database/Retrieval Failure:</b> If database retrieval fails, the system shows a technical error message and logs the issue for system administrators.</li> <li>• <b>No Data:</b> If no data matches the selected criteria, the system displays a clear message: “No data available for this selection.”</li> </ul>

Table 8: Use Case UC-04c: Review Aggregated Reports

## 6.6 UC-05 Reporting & Analytics



**Figure 6:** Reporting and Analytics

<b>Use-case ID</b>	UC-05a
<b>Use-case name</b>	Generate Departmental Report
<b>Use-case overview</b>	Department Chair generates a department-level report (attendance, performance, session counts) for academic monitoring and decisions.
<b>Actors</b>	Department Chair
<b>Preconditions</b>	<ol style="list-style-type: none"> <li>1. Authenticated and authorized to view departmental reports.</li> <li>2. Sessions, feedback, and (if applicable) progress data exist.</li> <li>3. System is operational.</li> </ol>
<b>Trigger</b>	Chair opens “Reporting” and selects “Departmental Report”.
<b>Steps</b>	<ol style="list-style-type: none"> <li>1. Set filters (term/date, program/department, tutor/cohort).</li> <li>2. System retrieves and analyzes metrics (attendance, performance, session counts).</li> <li>3. System displays tables/charts with data-source notes.</li> <li>4. Optional: Export CSV/PDF with metadata (generation time, filters, version); log action.</li> </ol>
<b>Postconditions</b>	<ol style="list-style-type: none"> <li>1. Departmental report displayed; optional file generated.</li> <li>2. Action recorded in the audit log.</li> </ol>
<b>Alternative Flows</b>	<ol style="list-style-type: none"> <li>1. Change filters/scope → system refreshes results.</li> </ol>
<b>Exception Flow</b>	<ol style="list-style-type: none"> <li>1. No data for selected criteria → show “No data available for this selection.”</li> <li>2. Export error (I/O or size) → suggest narrowing scope or retrying.</li> <li>3. Data source unavailable → show service notice; allow retry when available.</li> </ol>

Table 9: Use Case UC-05a: Generate Departmental Report

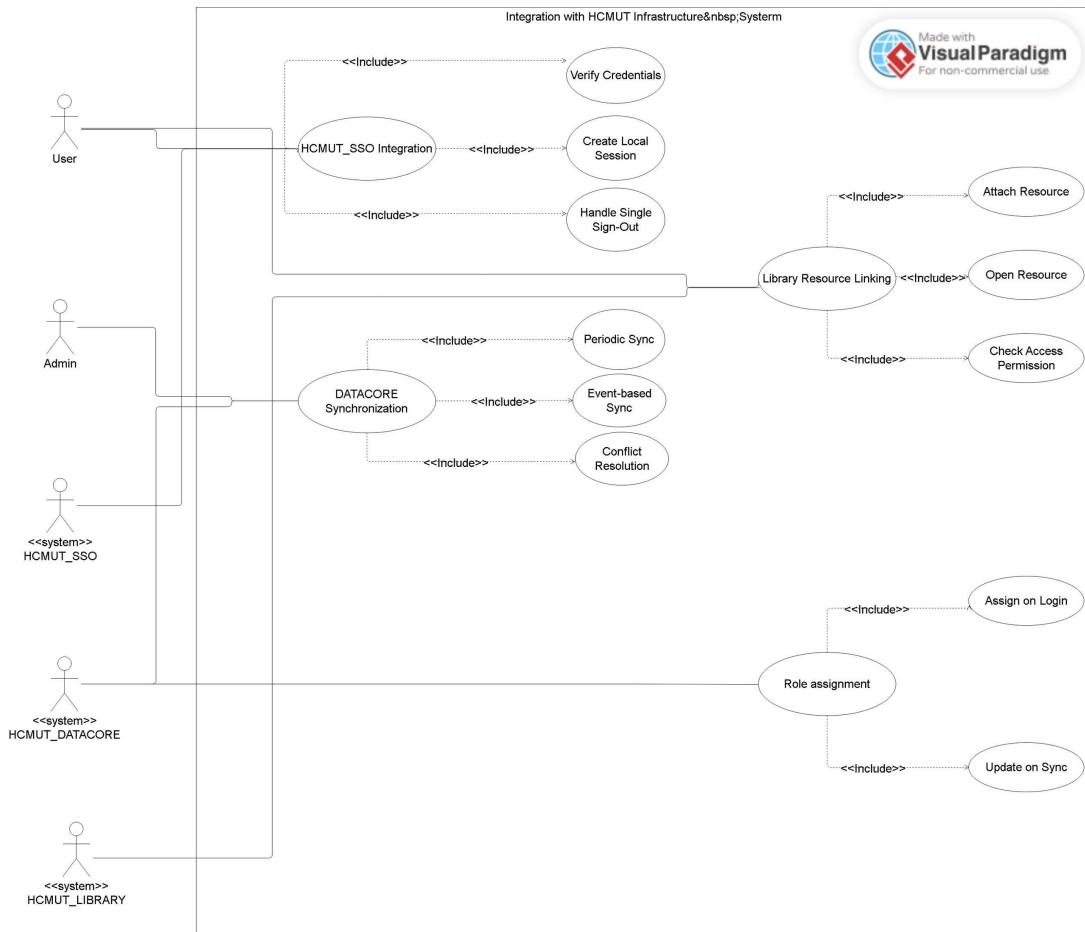
<b>Use-case ID</b>	UC-05b
<b>Use-case name</b>	View Tutor Workload Dashboard
<b>Use-case overview</b>	Academic Affairs views tutor workload and student demand to allocate resources effectively.
<b>Actors</b>	Academic Affairs Office
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>1. Authenticated and authorized for workload dashboards.</li> <li>2. Sessions/booking and feedback data exist.</li> <li>3. System is operational.</li> </ul>
<b>Trigger</b>	Office opens “Workload & Demand” dashboard.
<b>Steps</b>	<ul style="list-style-type: none"> <li>1. Select filters (term/date, department/program).</li> <li>2. System aggregates tutor load and demand indicators.</li> <li>3. System displays dashboard (tables/charts, drill-down).</li> <li>4. Optional: Export CSV/PDF with metadata; log action.</li> </ul>
<b>Postconditions</b>	<ul style="list-style-type: none"> <li>1. Workload dashboard visible; optional export ready.</li> <li>2. Access recorded in the audit log.</li> </ul>
<b>Alternative Flows</b>	<ul style="list-style-type: none"> <li>1. Drill-down by tutor/program → refresh view.</li> </ul>
<b>Exception Flow</b>	<ul style="list-style-type: none"> <li>1. No data for selected criteria → show “No data available for this selection.”</li> <li>2. Export error (I/O or size) → suggest narrowing scope or retrying.</li> <li>3. Data source unavailable → show service notice; allow retry when available.</li> </ul>

Table 10: Use Case UC-05b: View Tutor Workload Dashboard

<b>Use-case ID</b>	UC-05c
<b>Use-case name</b>	Generate Participation Report
<b>Use-case overview</b>	Student Affairs generates participation reports for credit/scholarship consideration.
<b>Actors</b>	Student Affairs Office
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>1. Authenticated and authorized for participation reports.</li> <li>2. Sessions and attendance/feedback records exist.</li> <li>3. System is operational.</li> </ul>
<b>Trigger</b>	Office selects “Participation Report”.
<b>Steps</b>	<ul style="list-style-type: none"> <li>1. Choose filters (term/date, cohort, program, thresholds).</li> <li>2. System compiles participation metrics and eligibility indicators.</li> <li>3. System displays results; Optional: export CSV/PDF with metadata; log action.</li> </ul>
<b>Postconditions</b>	<ul style="list-style-type: none"> <li>1. Participation report displayed; optional file generated.</li> <li>2. Action logged.</li> </ul>
<b>Alternative Flows</b>	<ul style="list-style-type: none"> <li>1. Adjust filters/criteria → refresh results.</li> </ul>
<b>Exception Flow</b>	<ul style="list-style-type: none"> <li>1. No data for selected criteria → show “No data available for this selection.”</li> <li>2. Export error (I/O or size) → suggest narrowing scope or retrying.</li> <li>3. Data source unavailable → show service notice; allow retry when available.</li> </ul>

Table 11: Use Case UC-05c: Generate Participation Report

## 6.7 UC-06 Integration with HCMUT Infrastructure



**Figure 7:** Integration with HCMUT Infrastructure

<b>Use-case ID</b>	UC-06a
<b>Use-case name</b>	HCMUT_SSO Integration
<b>Use-case overview</b>	The system integrates with HCMUT_SSO for unified authentication, allowing users to log in using university credentials and automatically manage single sign-out.
<b>Actors</b>	User, System, HCMUT_SSO
<b>Preconditions</b>	HCMUT_SSO service is operational and reachable.
<b>Trigger</b>	User initiates login via HCMUT_SSO.
<b>Steps</b>	<ol style="list-style-type: none"> <li>1. User selects “Login with HCMUT_SSO”.</li> <li>2. System redirects to the authentication portal.</li> <li>3. HCMUT_SSO validates credentials and returns a token.</li> <li>4. System verifies the token and creates a session.</li> <li>5. Upon sign-out at SSO, the system terminates the local session.</li> </ol>
<b>Postconditions</b>	<ol style="list-style-type: none"> <li>1. User is successfully authenticated.</li> <li>2. Single sign-out ensures session consistency.</li> </ol>
<b>Exception Flow</b>	<ol style="list-style-type: none"> <li>1. Invalid or expired token → login attempt rejected.</li> <li>2. SSO service unavailable → system displays maintenance message.</li> </ol>
<b>Priority</b>	Must

Table 12: Use Case UC-06a: HCMUT\_SSO Integration

<b>Use-case ID</b>	UC-06b
<b>Use-case name</b>	DATACORE Synchronization
<b>Use-case overview</b>	The system synchronizes personal and academic data from HCMUT_DATACORE periodically or in near real-time to ensure data consistency and reduce manual entry.
<b>Actors</b>	System, HCMUT_DATACORE, Administrator
<b>Preconditions</b>	DATACORE APIs are online and accessible.
<b>Trigger</b>	Scheduled synchronization or data-change event detected.
<b>Steps</b>	<ol style="list-style-type: none"> <li>1. System triggers synchronization with DATACORE.</li> <li>2. Retrieve updated profiles and academic data.</li> <li>3. Validate and compare data with local records.</li> <li>4. Update local data using DATACORE as source of truth.</li> <li>5. Log synchronization status and timestamp.</li> </ol>
<b>Postconditions</b>	<ol style="list-style-type: none"> <li>1. Local data mirrors DATACORE.</li> <li>2. Synchronization events logged for auditing.</li> </ol>
<b>Exception Flow</b>	<ol style="list-style-type: none"> <li>1. Connection timeout or API error → retry with exponential backoff.</li> <li>2. Invalid data format → skip record and log validation error.</li> </ol>
<b>Priority</b>	Must

Table 13: Use Case UC-06b: DATACORE Synchronization

<b>Use-case ID</b>	UC-06c
<b>Use-case name</b>	Role Assignment
<b>Use-case overview</b>	The system automatically assigns user roles (student, tutor, coordinator, department chair, or administrator) based on centralized role data from DATACORE and SSO.
<b>Actors</b>	User, System, HCMUT_DATACORE
<b>Preconditions</b>	User is authenticated via SSO and DATACORE role data is available.
<b>Trigger</b>	User login or scheduled role update.
<b>Steps</b>	<ol style="list-style-type: none"> <li>1. Retrieve role mapping from DATACORE.</li> <li>2. Match user ID with centralized role data.</li> <li>3. Assign permissions based on role.</li> <li>4. Apply changes in access-control policies.</li> <li>5. Log the role-assignment event.</li> </ol>
<b>Postconditions</b>	<ol style="list-style-type: none"> <li>1. User roles align with centralized data.</li> <li>2. Updated permissions take effect immediately.</li> </ol>
<b>Exception Flow</b>	<ol style="list-style-type: none"> <li>1. Role data missing → assign default “student” role and notify admin.</li> <li>2. Role conflict → logged and flagged for manual verification.</li> </ol>
<b>Priority</b>	Must

Table 14: Use Case UC-06c: Role Assignment

<b>Use-case ID</b>	UC-06d
<b>Use-case name</b>	Library Resource Linking
<b>Use-case overview</b>	The system connects with HCMUT_LIBRARY to let tutors and students attach or access library materials securely within tutoring sessions or summaries.
<b>Actors</b>	Student, Tutor, System, HCMUT_LIBRARY
<b>Preconditions</b>	Library API and authentication services are available.
<b>Trigger</b>	User attaches or opens a library resource.
<b>Steps</b>	<ol style="list-style-type: none"> <li>1. User searches or selects a library resource.</li> <li>2. System requests metadata and access permissions.</li> <li>3. Verify user eligibility via role-based access.</li> <li>4. Attach or open the resource in session view.</li> <li>5. Log the access event.</li> </ol>
<b>Postconditions</b>	<ol style="list-style-type: none"> <li>1. Resource successfully linked to the session or summary.</li> <li>2. Access rights enforced per library policy.</li> </ol>
<b>Exception Flow</b>	<ol style="list-style-type: none"> <li>1. Access denied → system displays permission error.</li> <li>2. Library service unavailable → prompt user to retry or queue attachment.</li> </ol>
<b>Priority</b>	Could

Table 15: Use Case UC-06d: Library Resource Linking

## 6.8 UC-07 Advanced / Optional Features

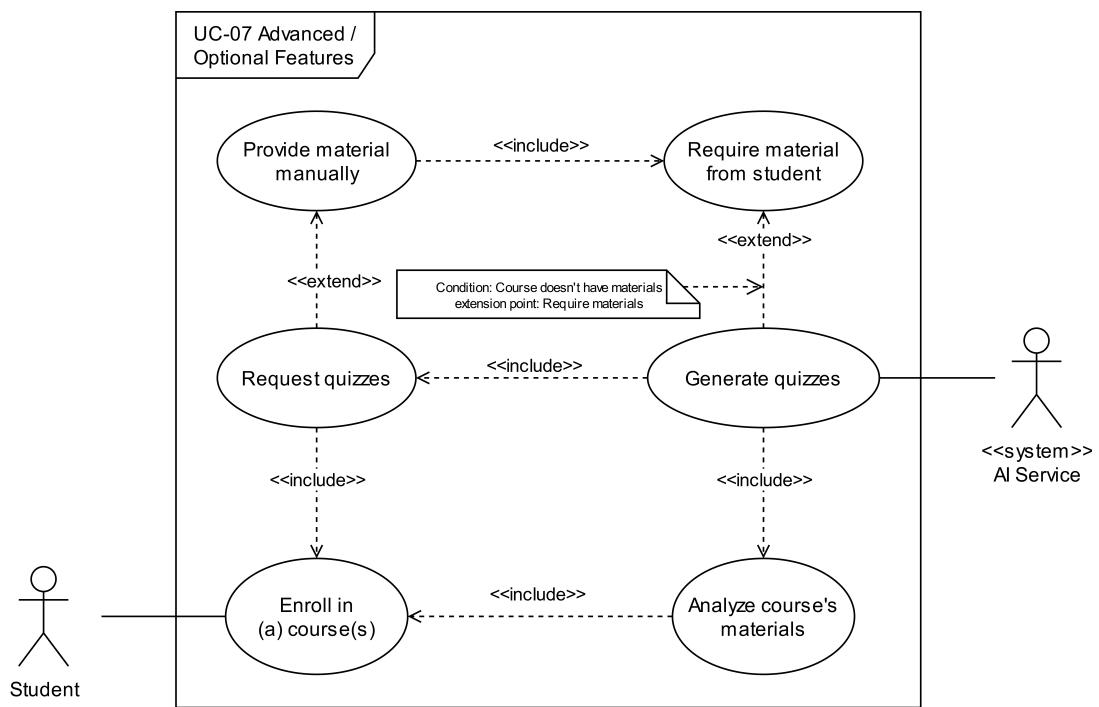


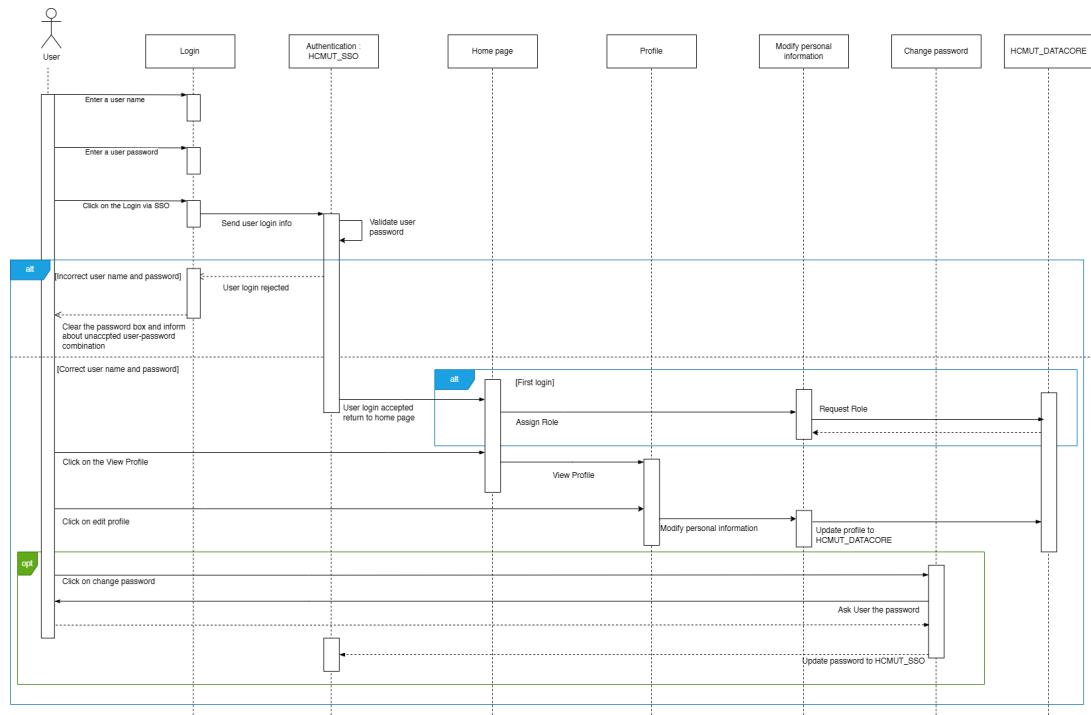
Figure 8: AI-generated review quizzes

<b>Use-case ID</b>	UC-07
<b>Use-case name</b>	AI-generated review quizzes
<b>Use-case overview</b>	Provide students with quizzes related to the courses they are enrolling.
<b>Actors</b>	Students, AI service
<b>Preconditions</b>	<ol style="list-style-type: none"> <li>1. The system is running.</li> <li>2. Internet connection is available.</li> <li>3. AI service must be available.</li> <li>4. The courses must have learning materials uploaded by tutors.</li> </ol>
<b>Trigger</b>	Students use the AI-based quiz generation function.
<b>Steps</b>	<ol style="list-style-type: none"> <li>1. Retrieve uploaded course materials of the course.</li> <li>2. Process and analyze the content with AI.</li> <li>3. Search the internet for related academic resources and quizzes.</li> <li>4. Generate quiz questions covering the key topics.</li> <li>5. Display the quiz to the requested students.</li> </ol>
<b>Postconditions</b>	<ol style="list-style-type: none"> <li>1. The quizzes are displayed on the screen of the requested students, and they can download the quizzes as a document file.</li> <li>2. The quizzes can be available until the end of the login session of the requested students, or until they finished the course if they chose to save the quizzes</li> </ol>
<b>Alternative Flows</b>	<ul style="list-style-type: none"> <li>- If the course doesn't have any learning materials, notify the user and request them for manual typing in key topics needed for review. A1: Invalid login → Access denied with error message.</li> </ul>
<b>Exception Flow</b>	<ol style="list-style-type: none"> <li>1. If AI service isn't available, display an error.</li> </ol>

Table 16: Use Case UC-07: AI-generated review quizzes

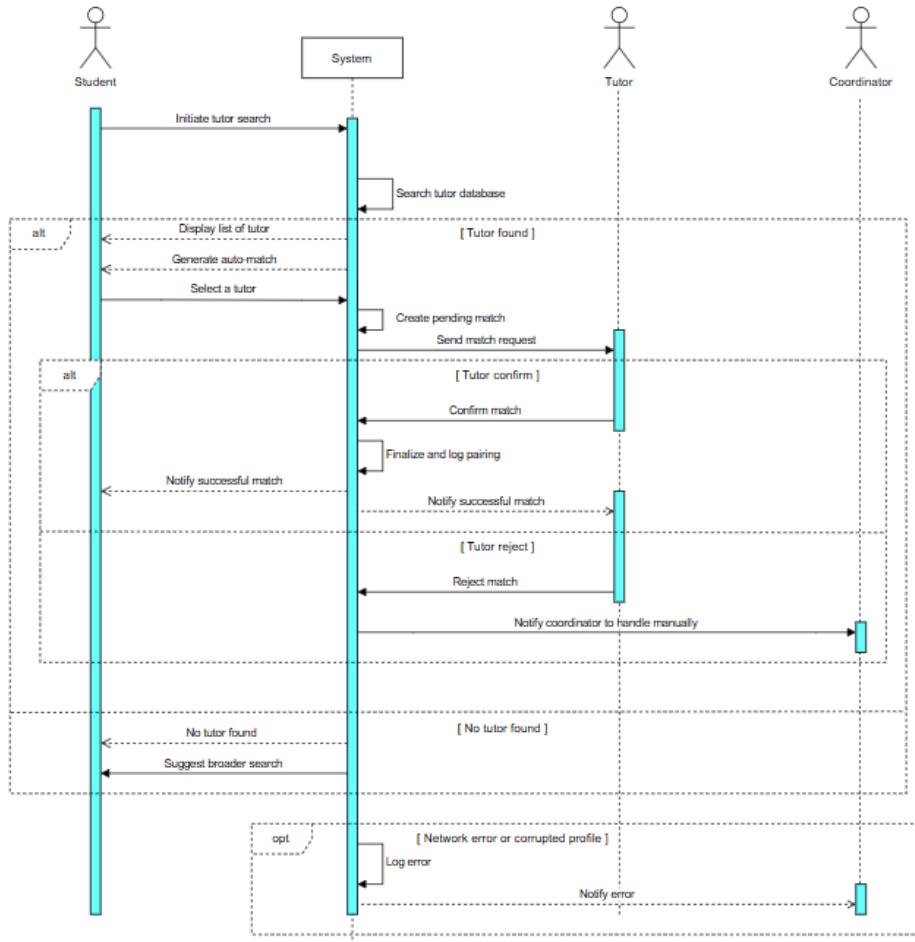
## 7 Sequence Diagram

### 7.1 SQ-01 Log in, Log out & Profile Management



**Figure 9:** Log in, Log out & Profile Management

## 7.2 SQ-02 Tutor-Student Matching



**Figure 10:** Tutor-Student Matching

## 7.3 SQ-03 Session Scheduling Management

Session Scheduling Management

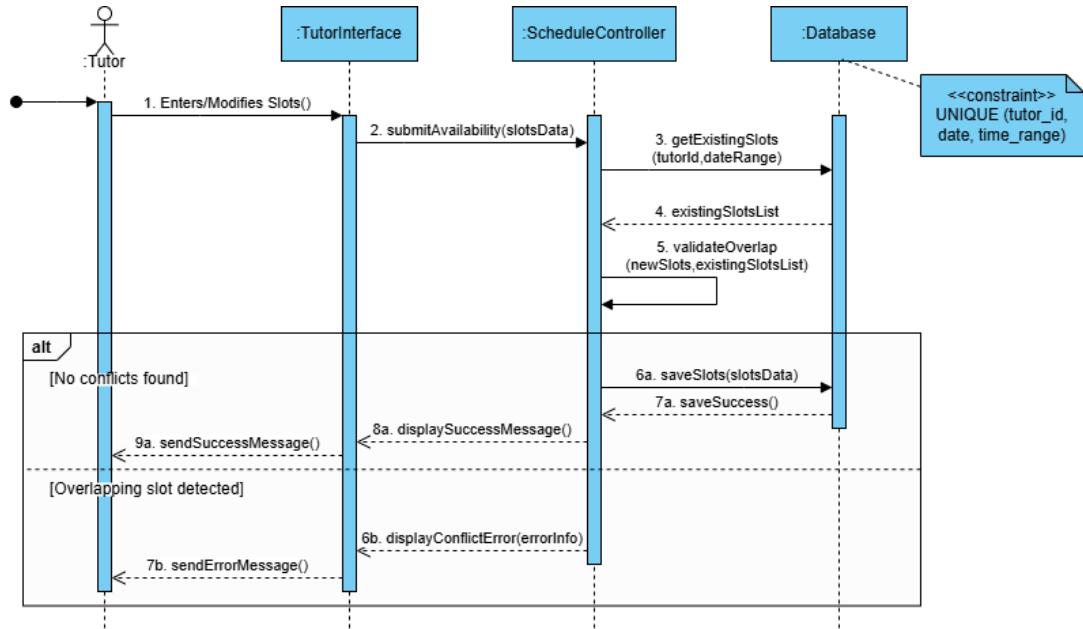


Figure 11: Tutor published available time slots

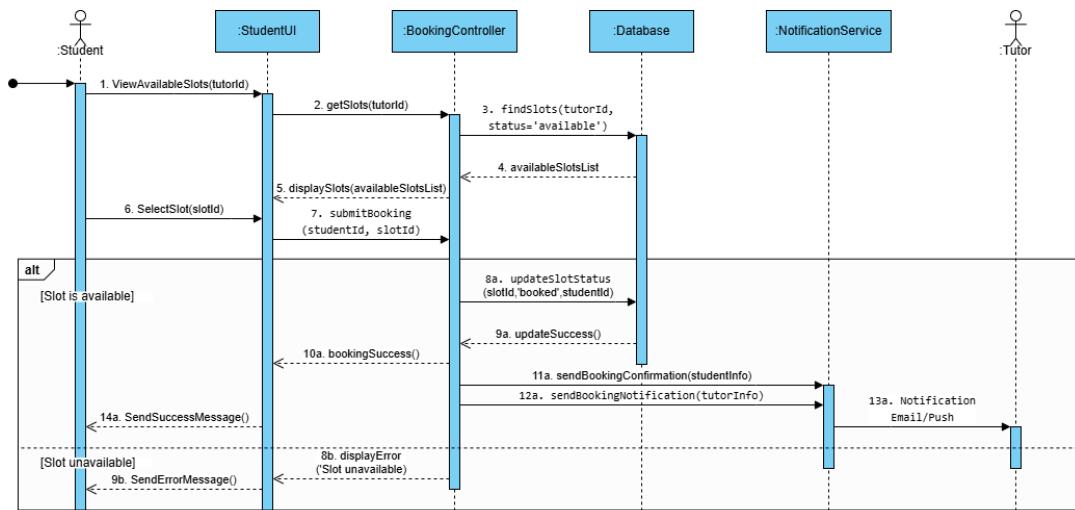


Figure 12: Student view and book session

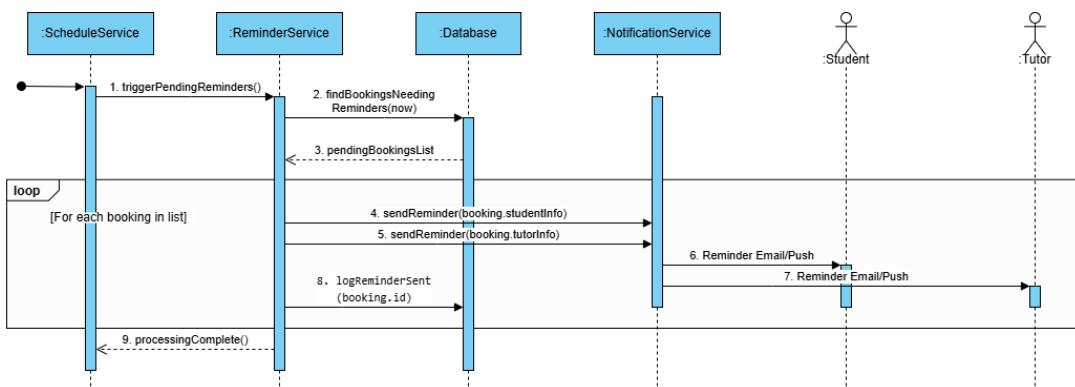
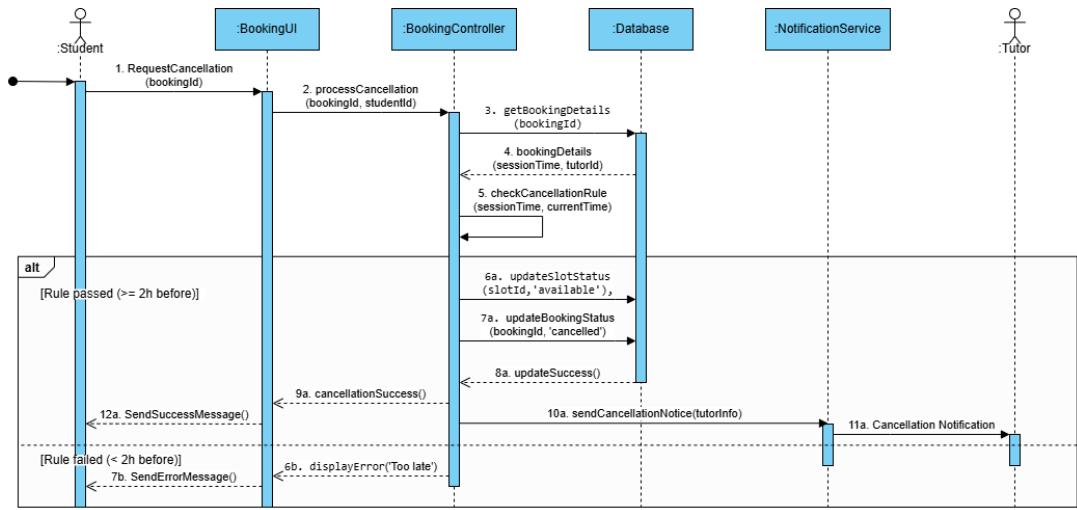


Figure 13: System sends notification reminders



**Figure 14:** Cancellation and rescheduling

## 7.4 SQ-04 Feedback & Progress Tracking

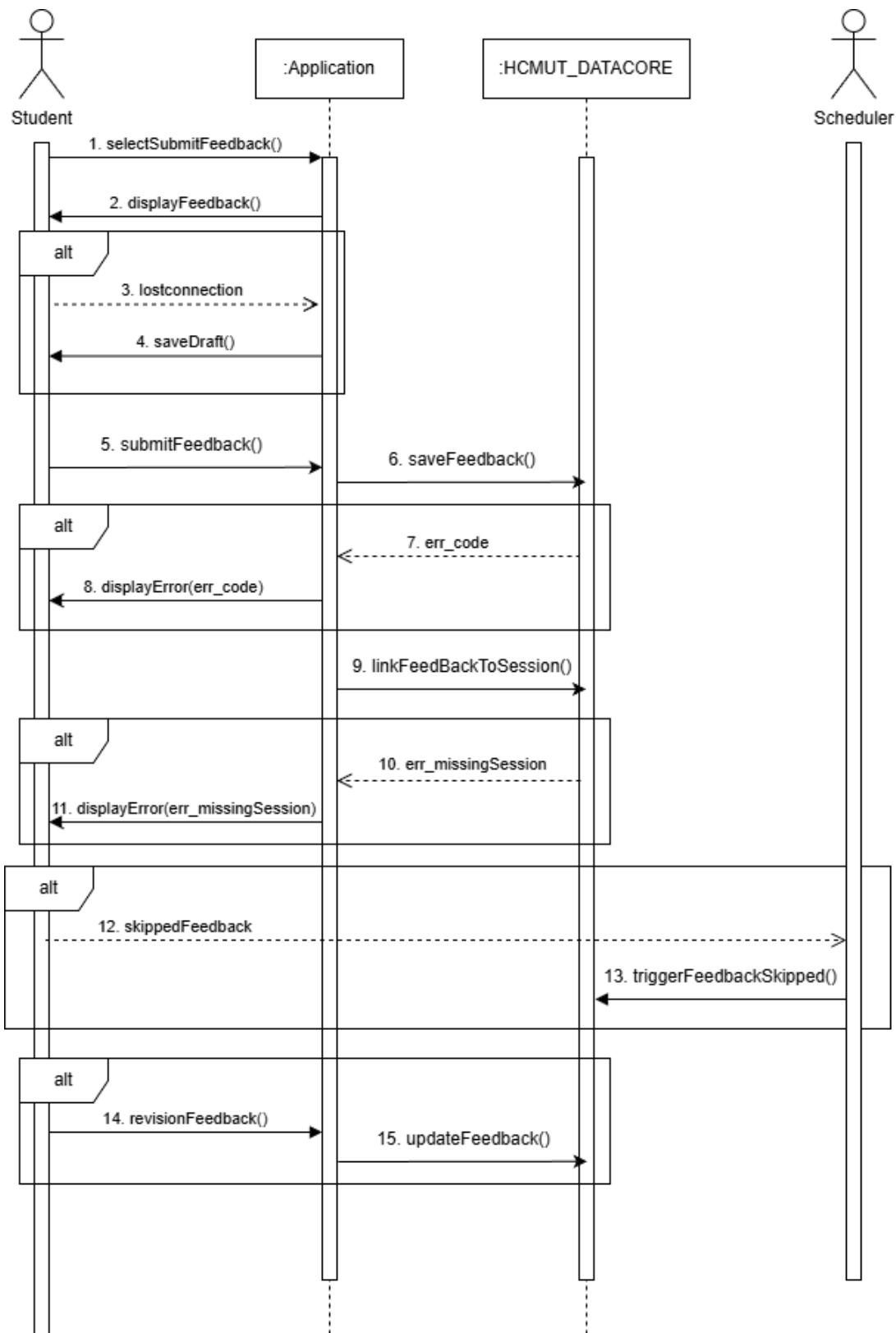
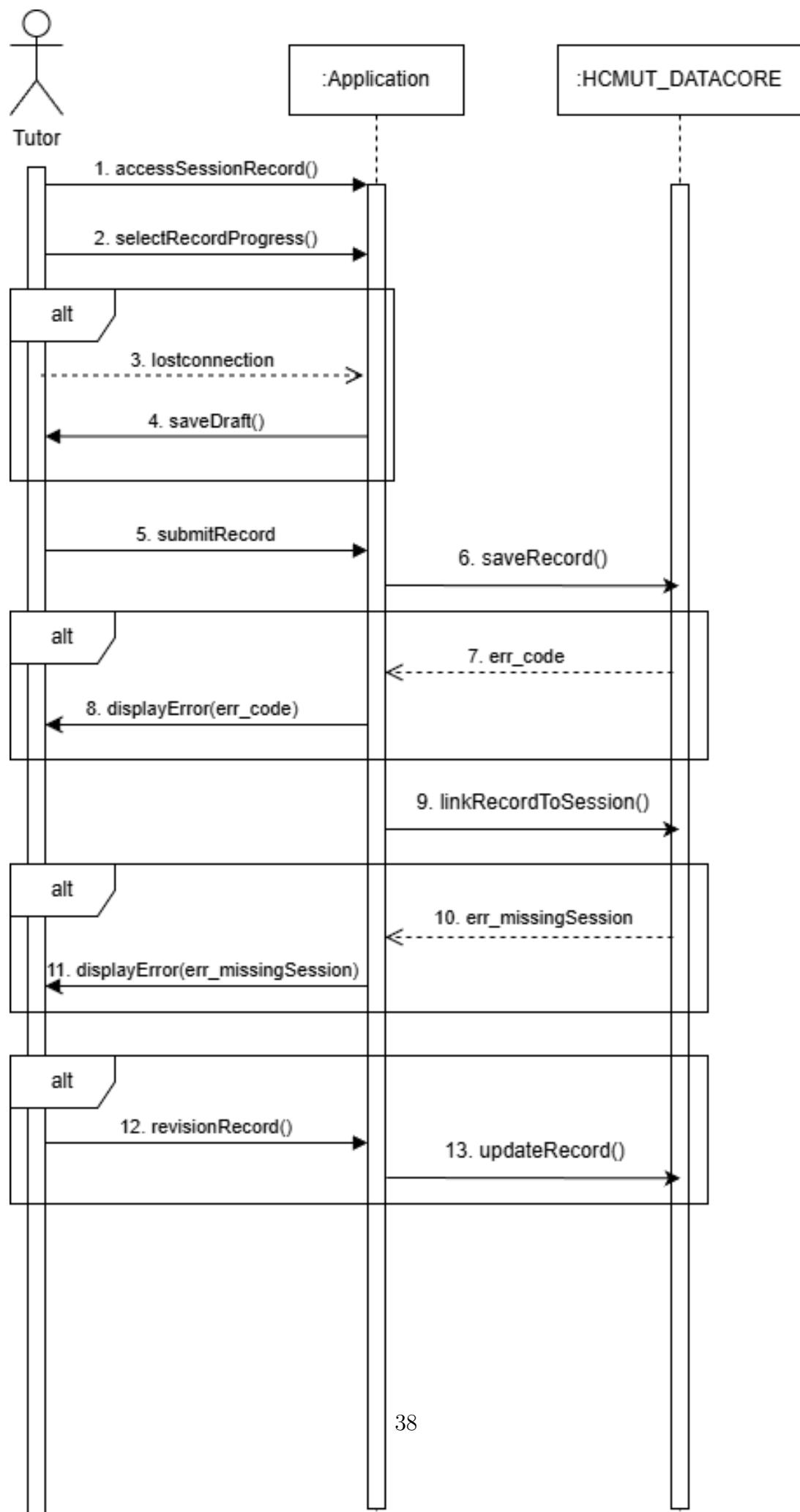
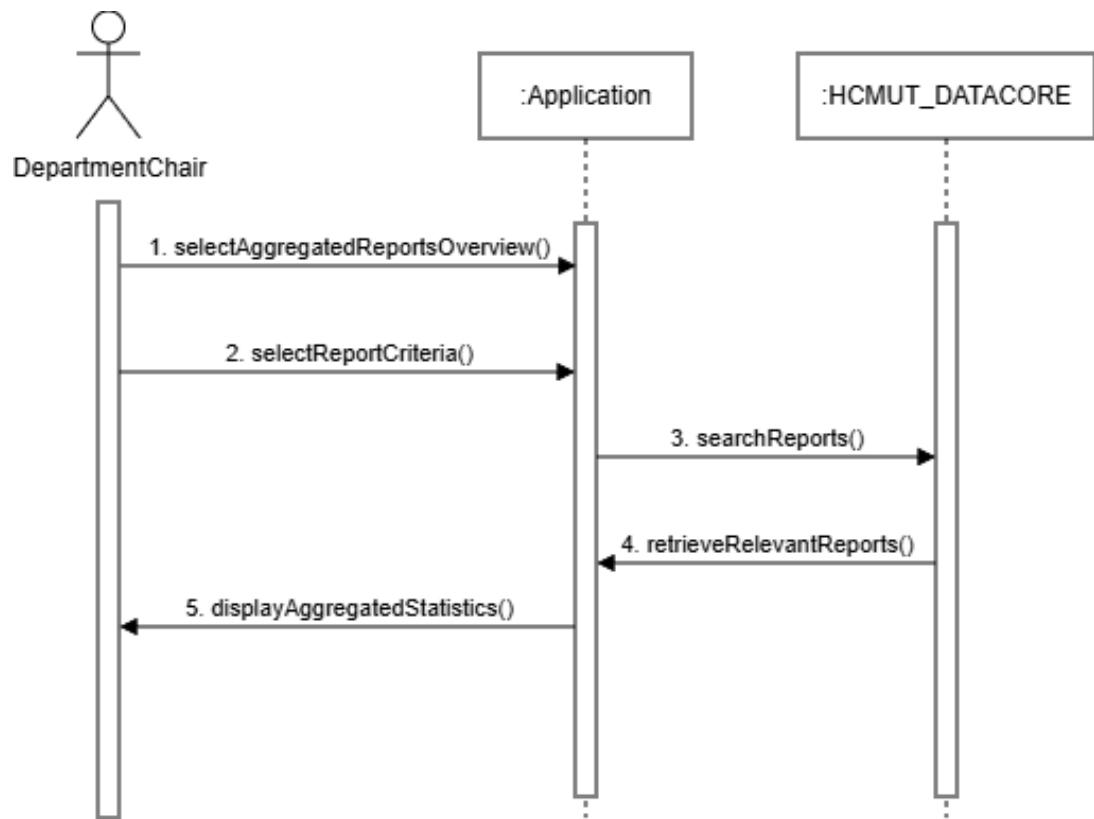


Figure 15: Feedback Submission

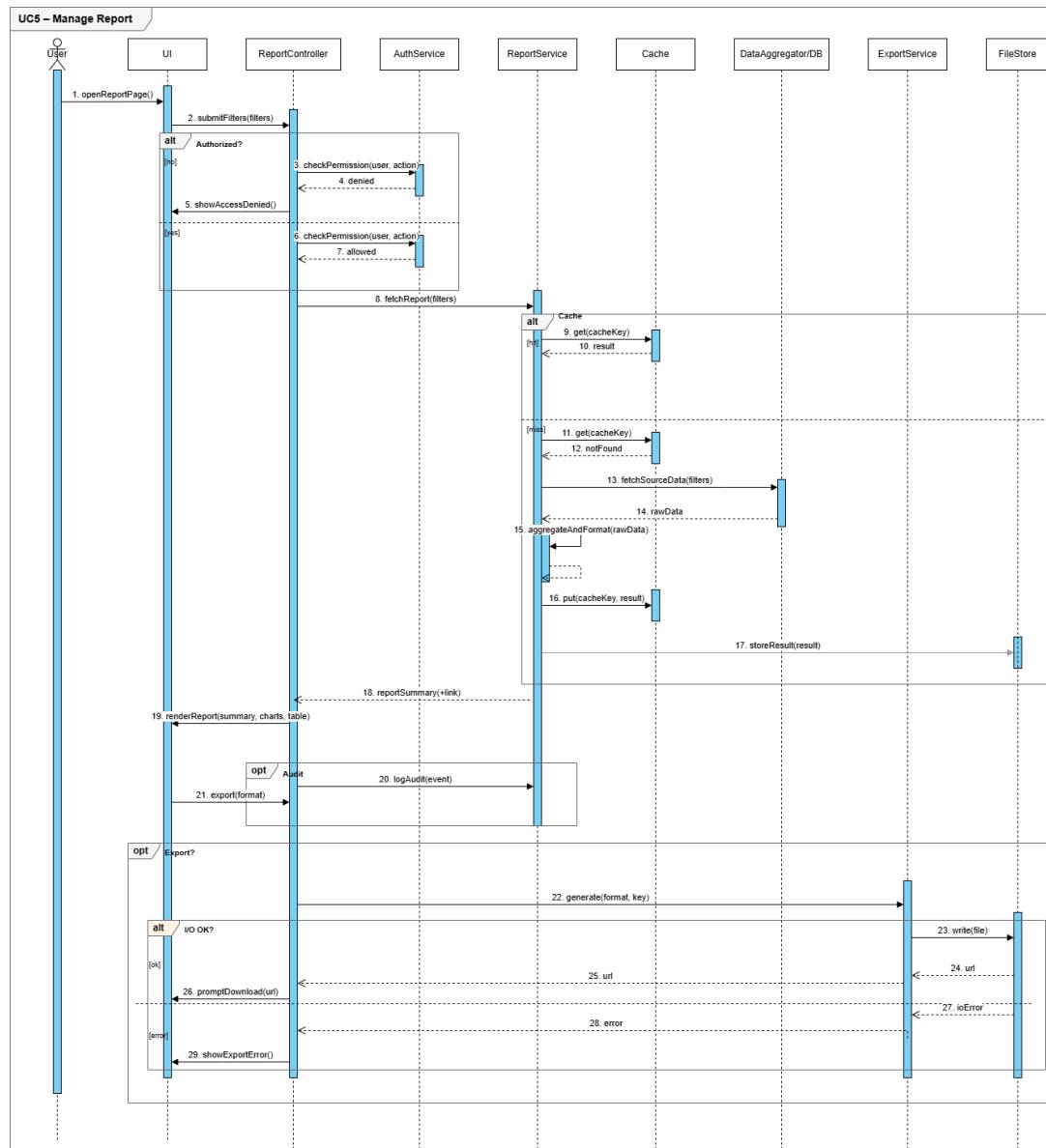


**Figure 16:** Session Recording Access



**Figure 17:** Report Tracking

## 7.5 SQ-05 Reporting & Analytics



**Figure 18:** Reporting & Analytics

## 8 Activity Diagram

### 8.1 AC-01 Log in, Log out & Profile Management

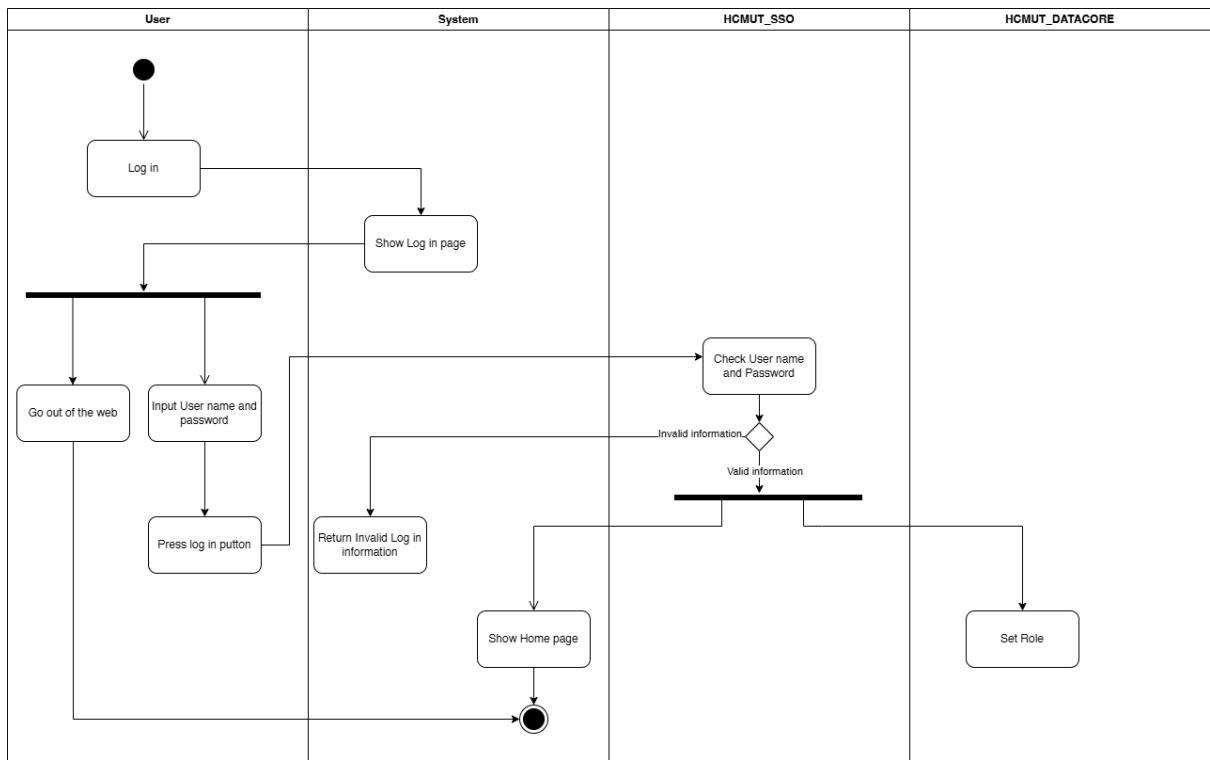
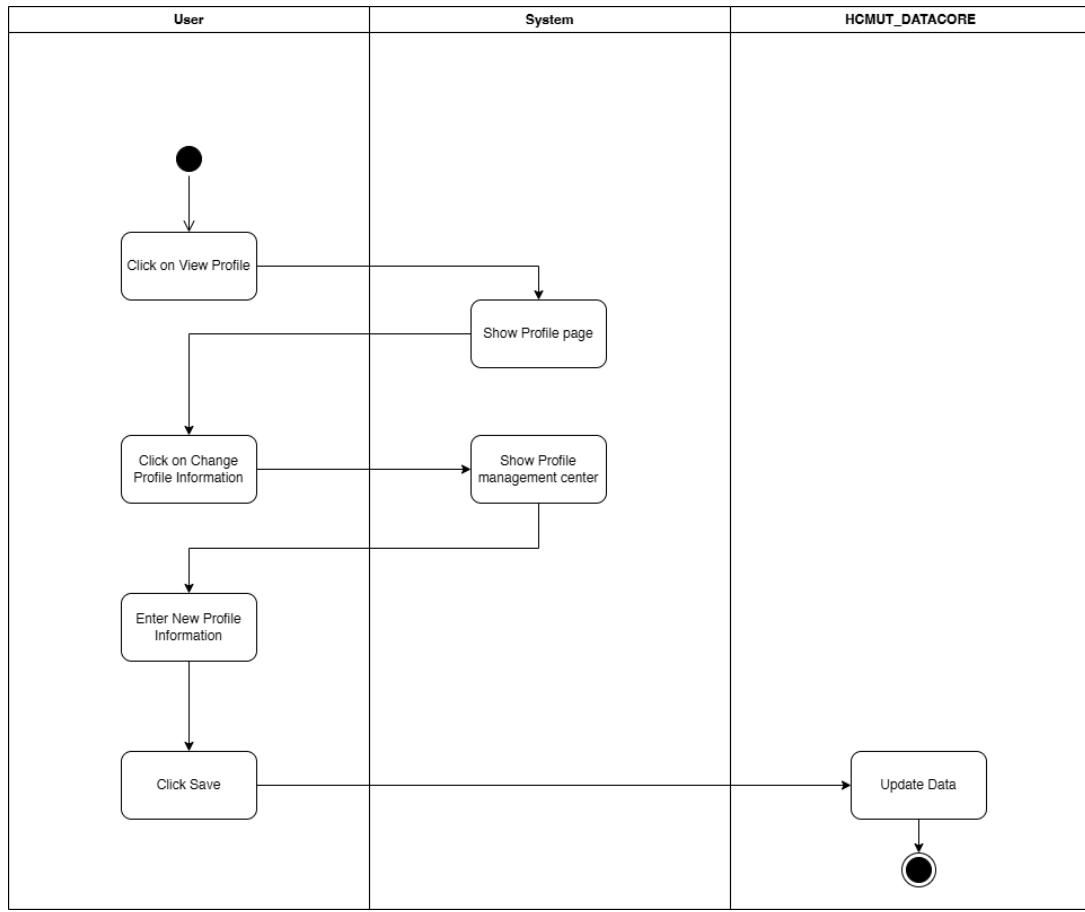
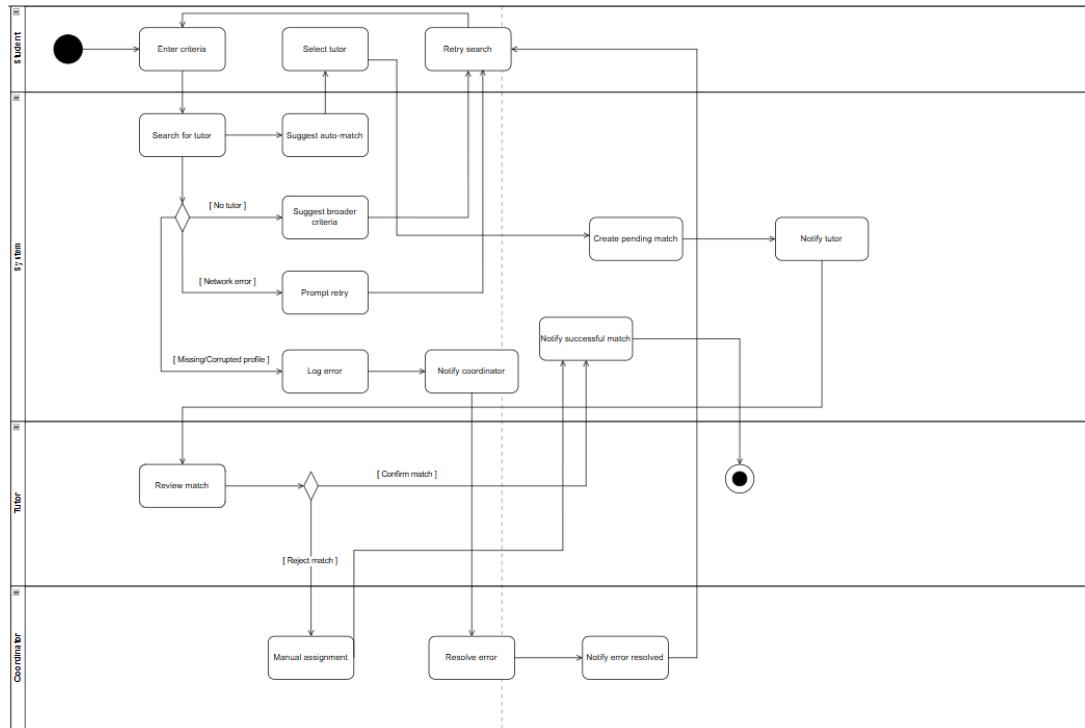


Figure 19: Log in



**Figure 20:** View Profile & Edit Profile

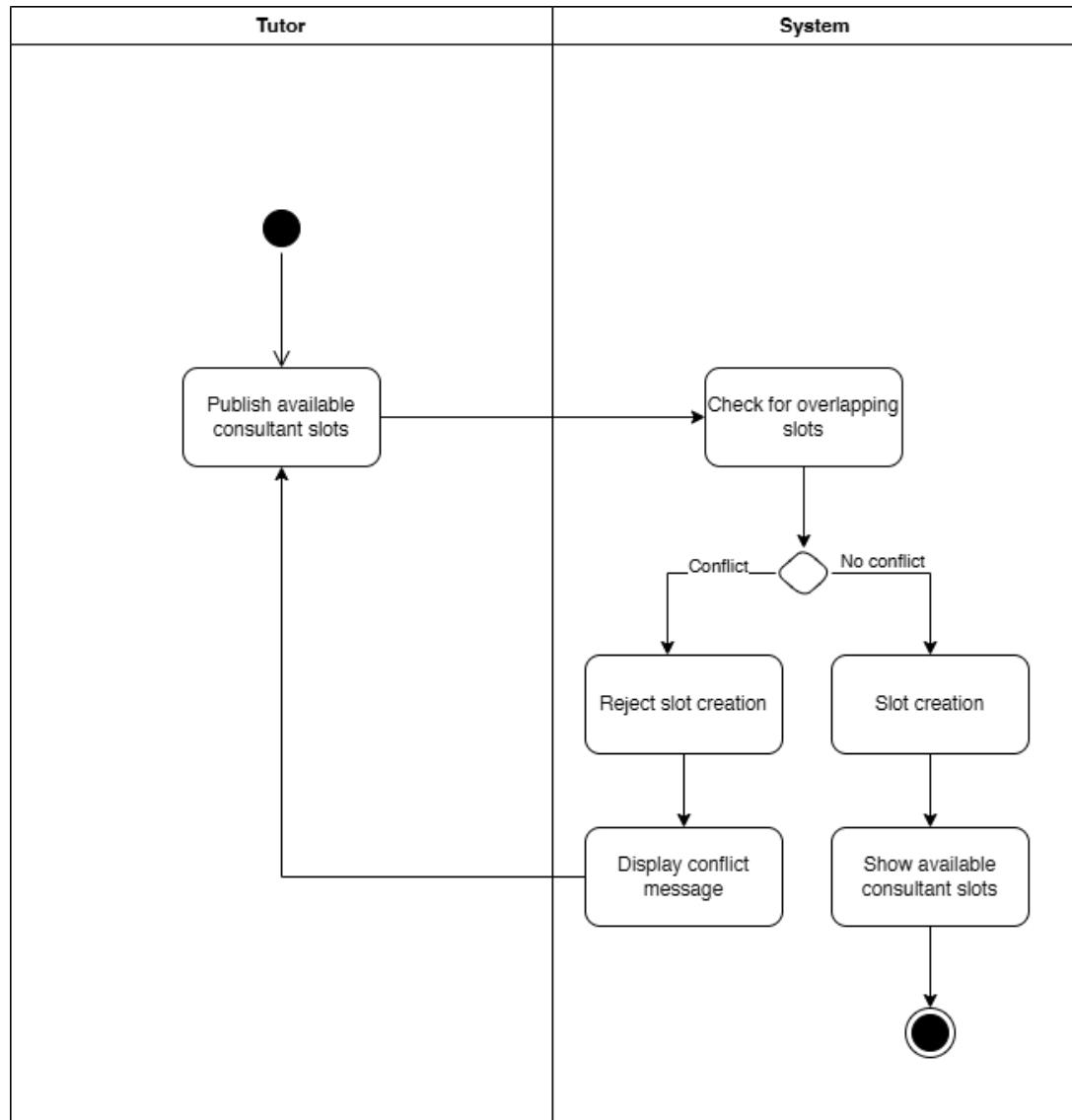
## 8.2 AC-02 Tutor-Student Matching



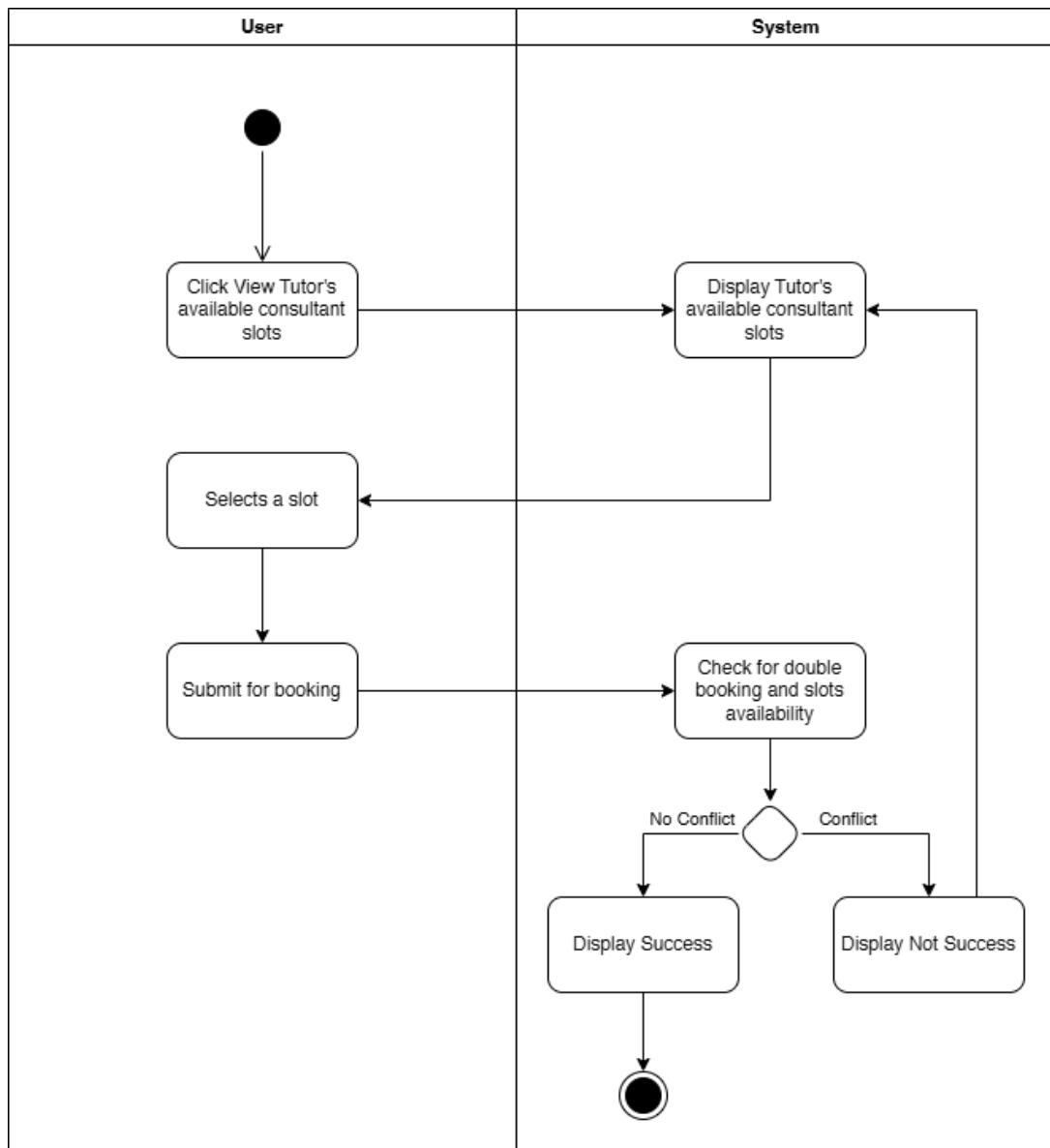
**Figure 21:** Tutor-Student Matching

### 8.3 AC-03 Session Scheduling Management

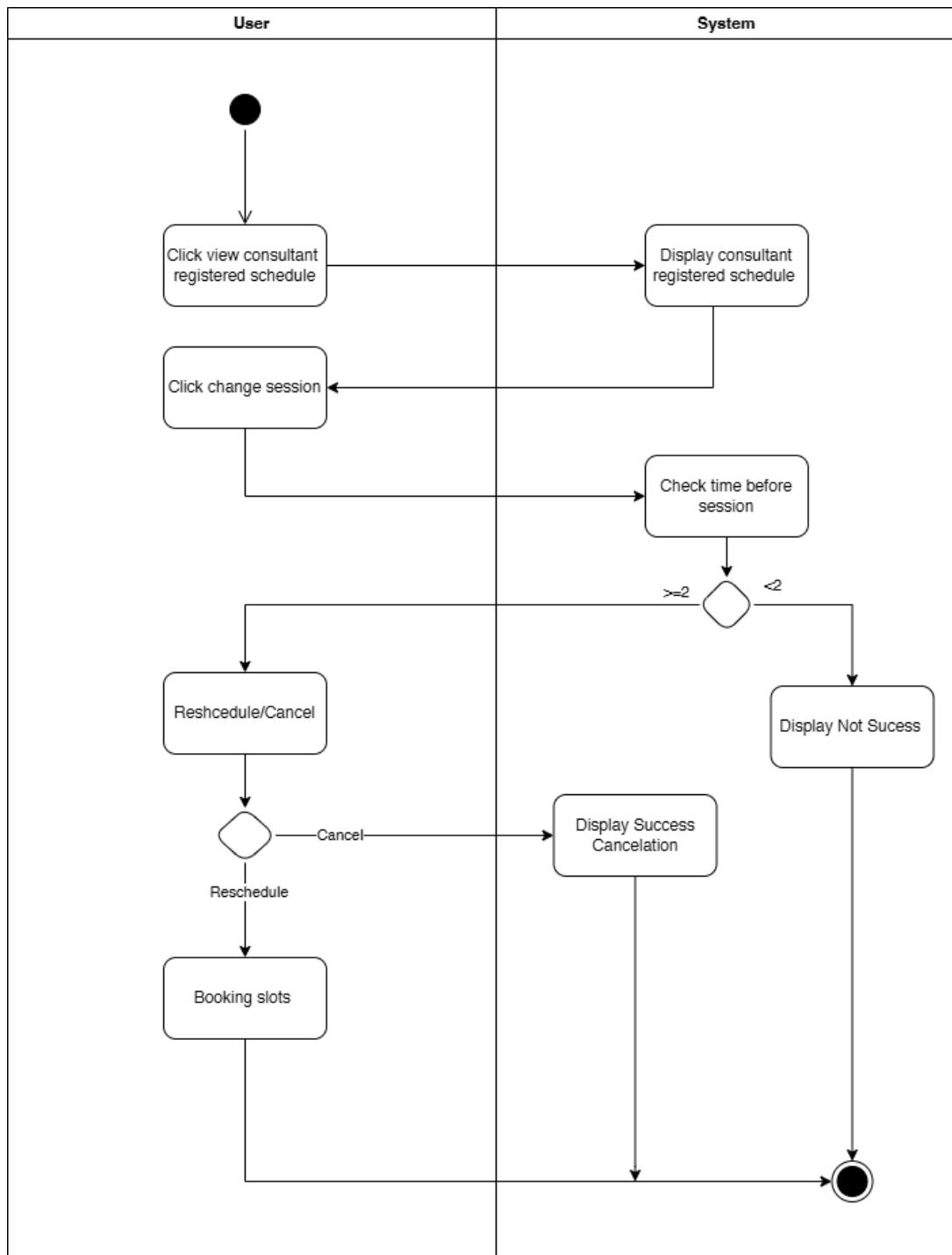
#### Session Scheduling Management



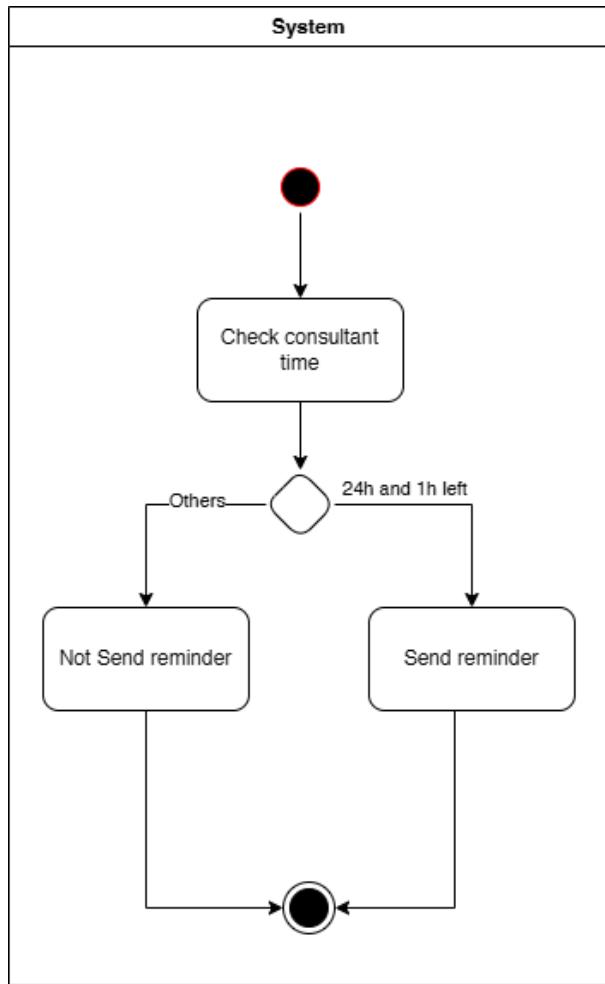
**Figure 22:** Tutor published available time slots



**Figure 23:** Student view and book session



**Figure 24:** Cancellation and rescheduling



**Figure 25:** System sends notification reminders

## 8.4 AC-04 Feedback & Progress Tracking

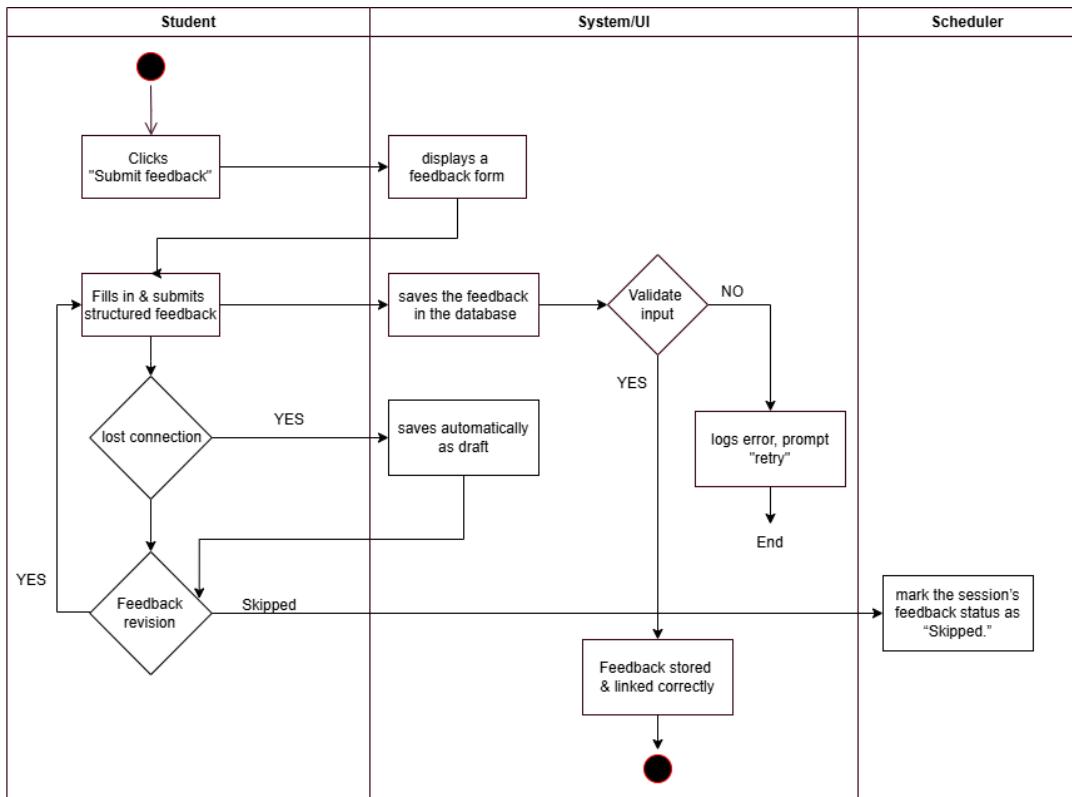
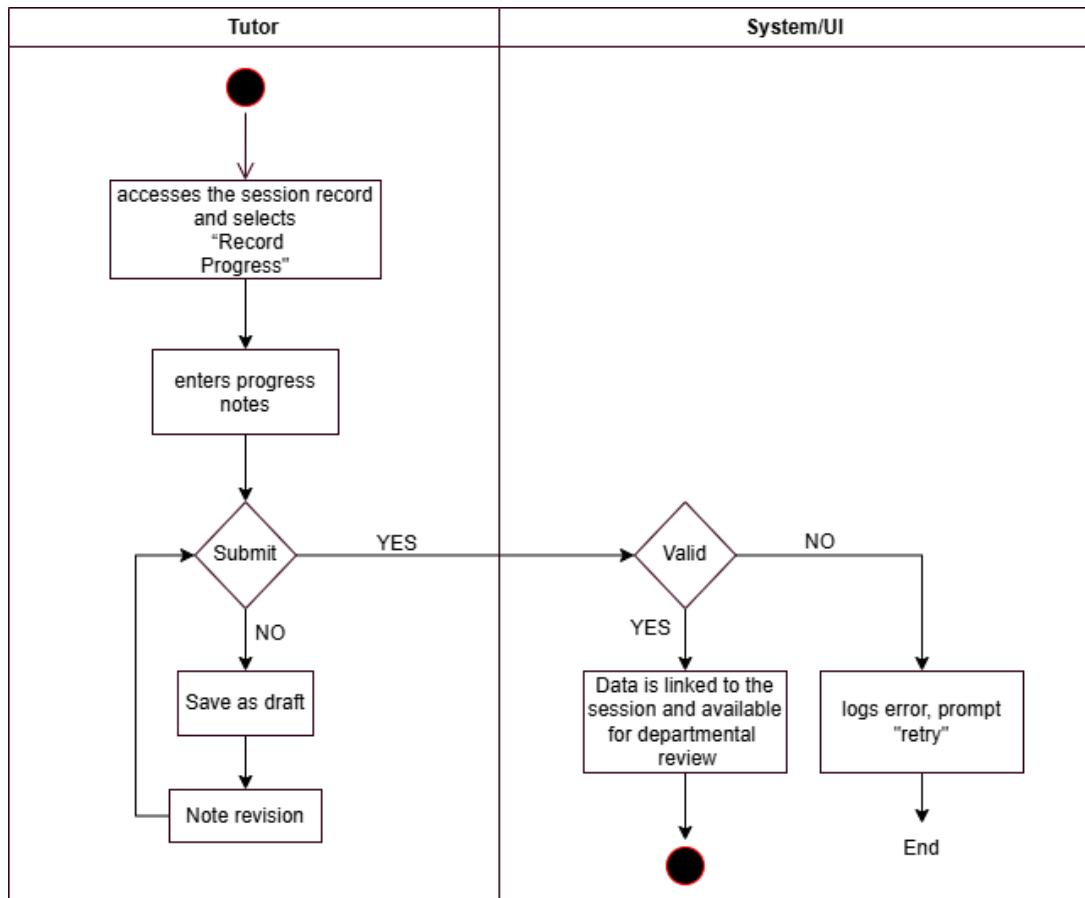
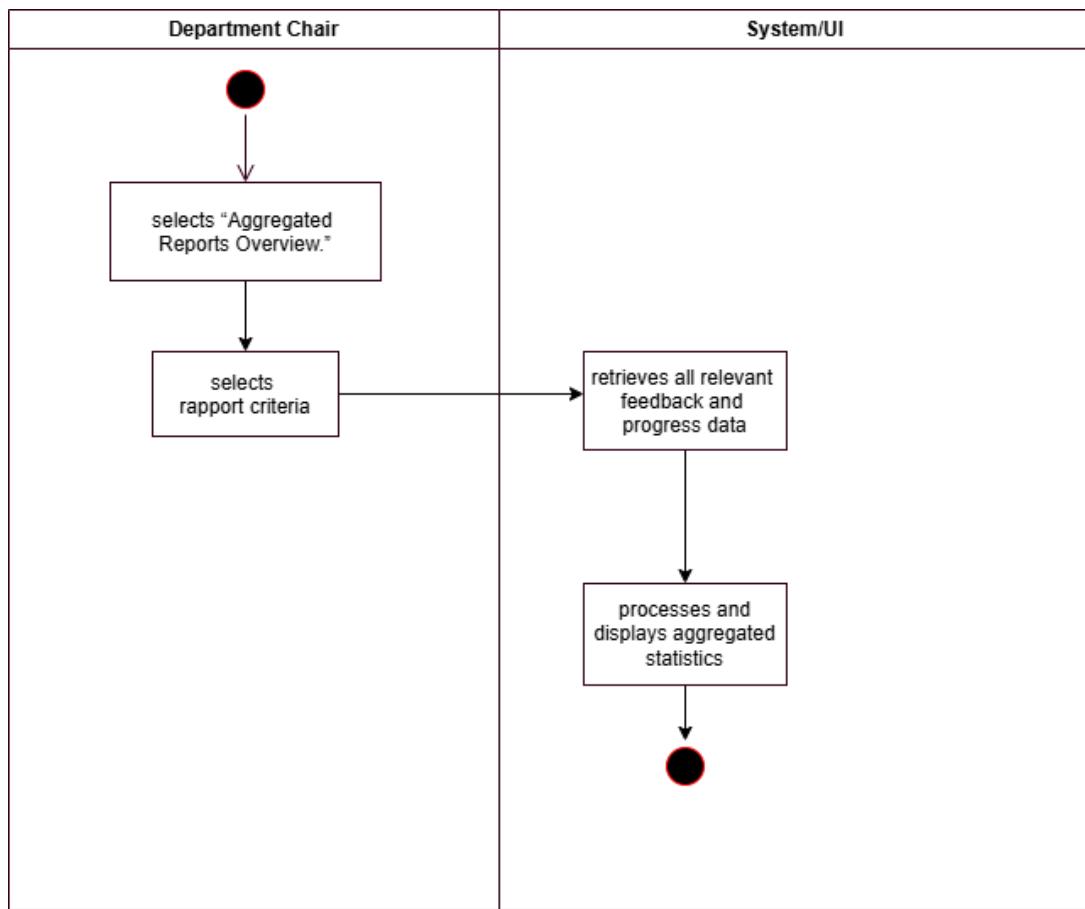


Figure 26: Feedback Submission

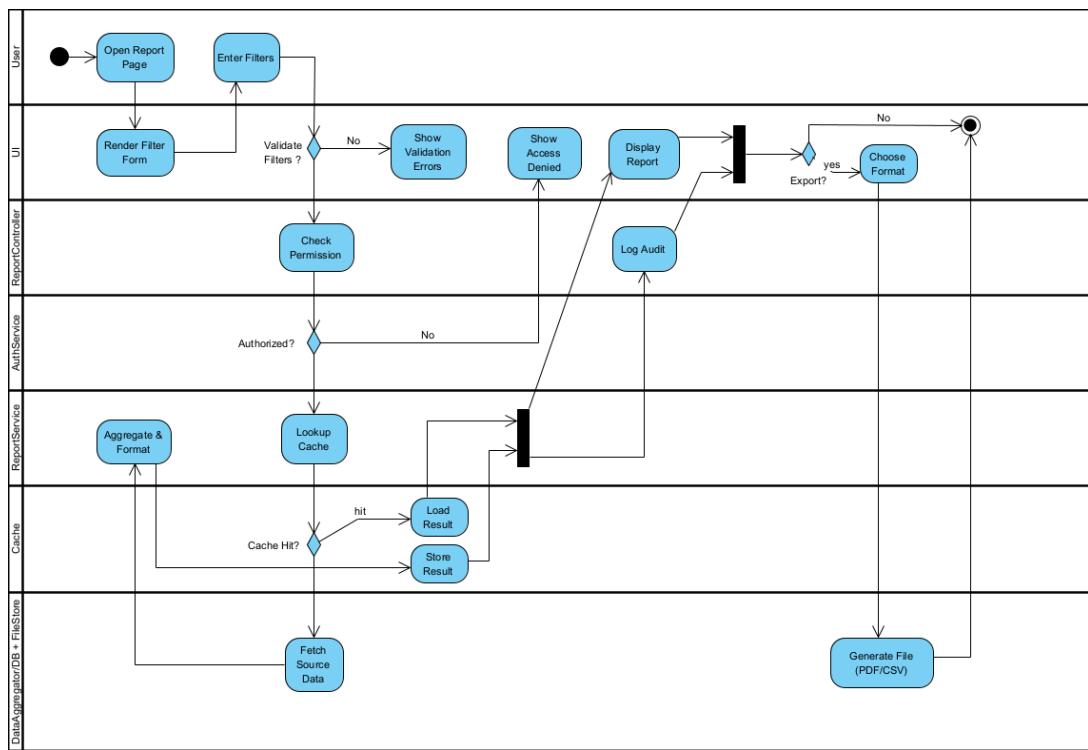


**Figure 27:** Session Recording Access



**Figure 28:** Report Tracking

## 8.5 AC-05 Reporting & Analytics



**Figure 29:** Reporting & Analytics

## 9 State Diagram

### 9.1 SD-01 Log in, Log out & Profile Management

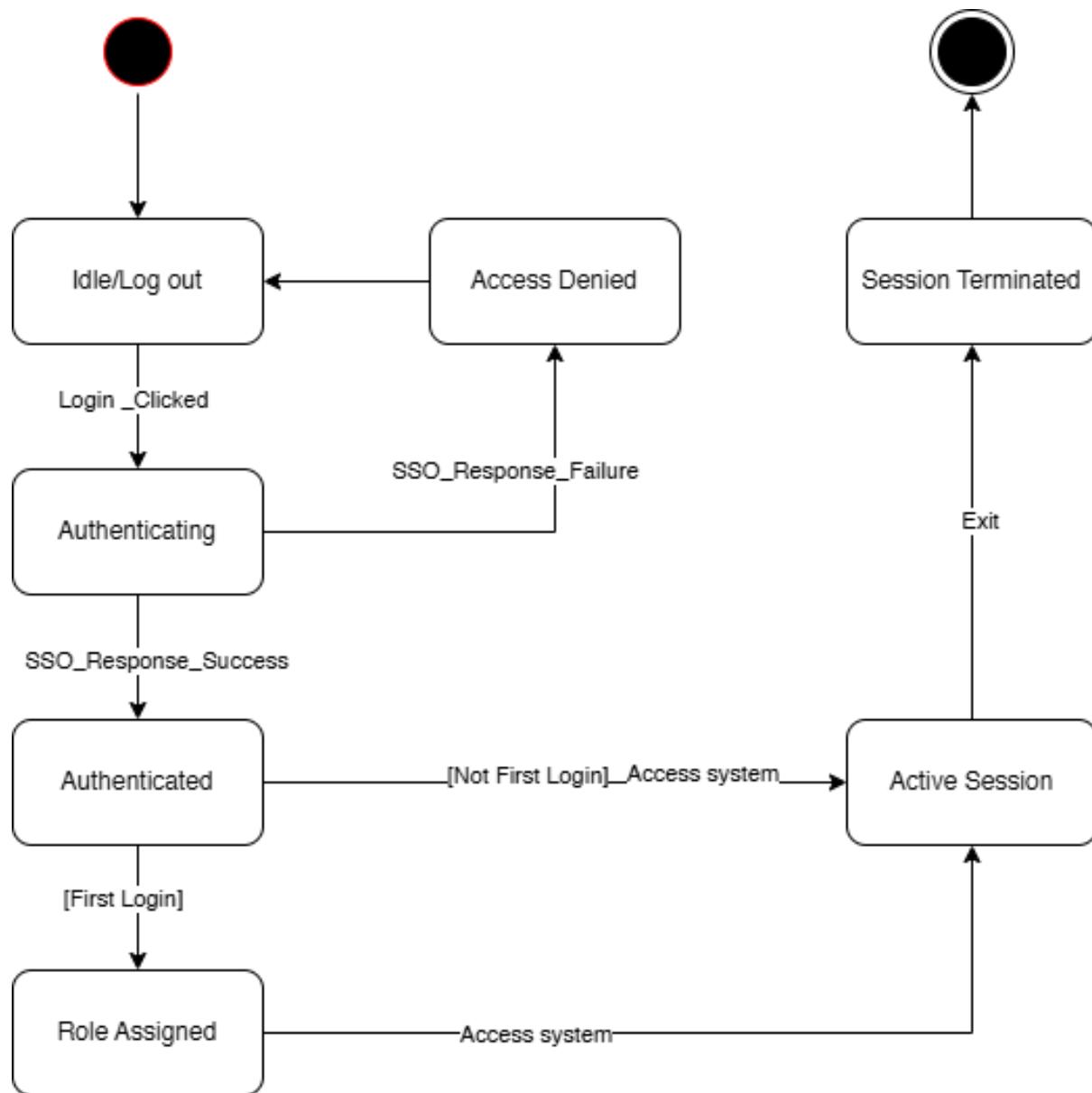
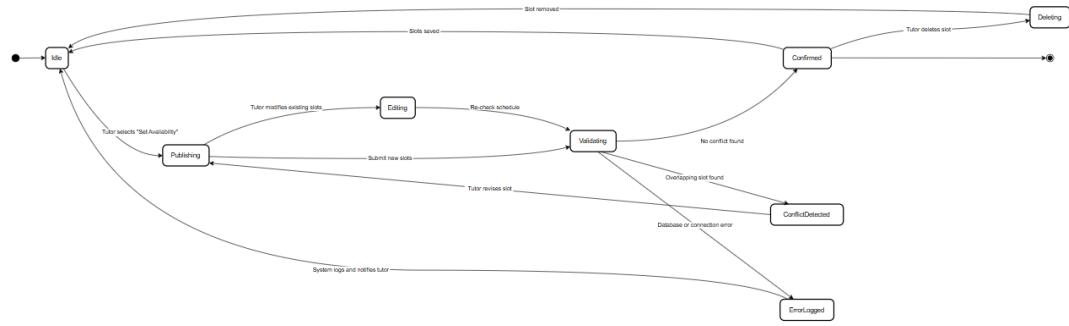


Figure 9: Log in

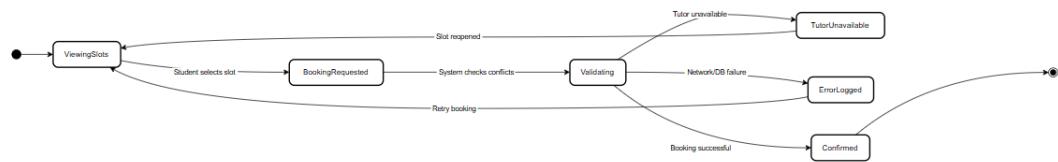
## 9.2 SD-02 Tutor-Student Matching



**Figure 10:** Tutor-Student Matching

## 9.3 SD-03 Session Scheduling Management

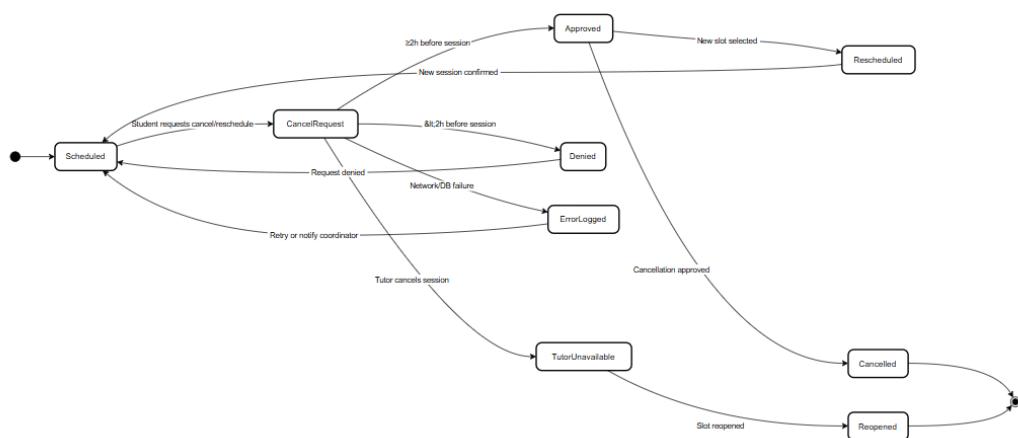
### Session Scheduling Management



**Figure 11:** Booking a session

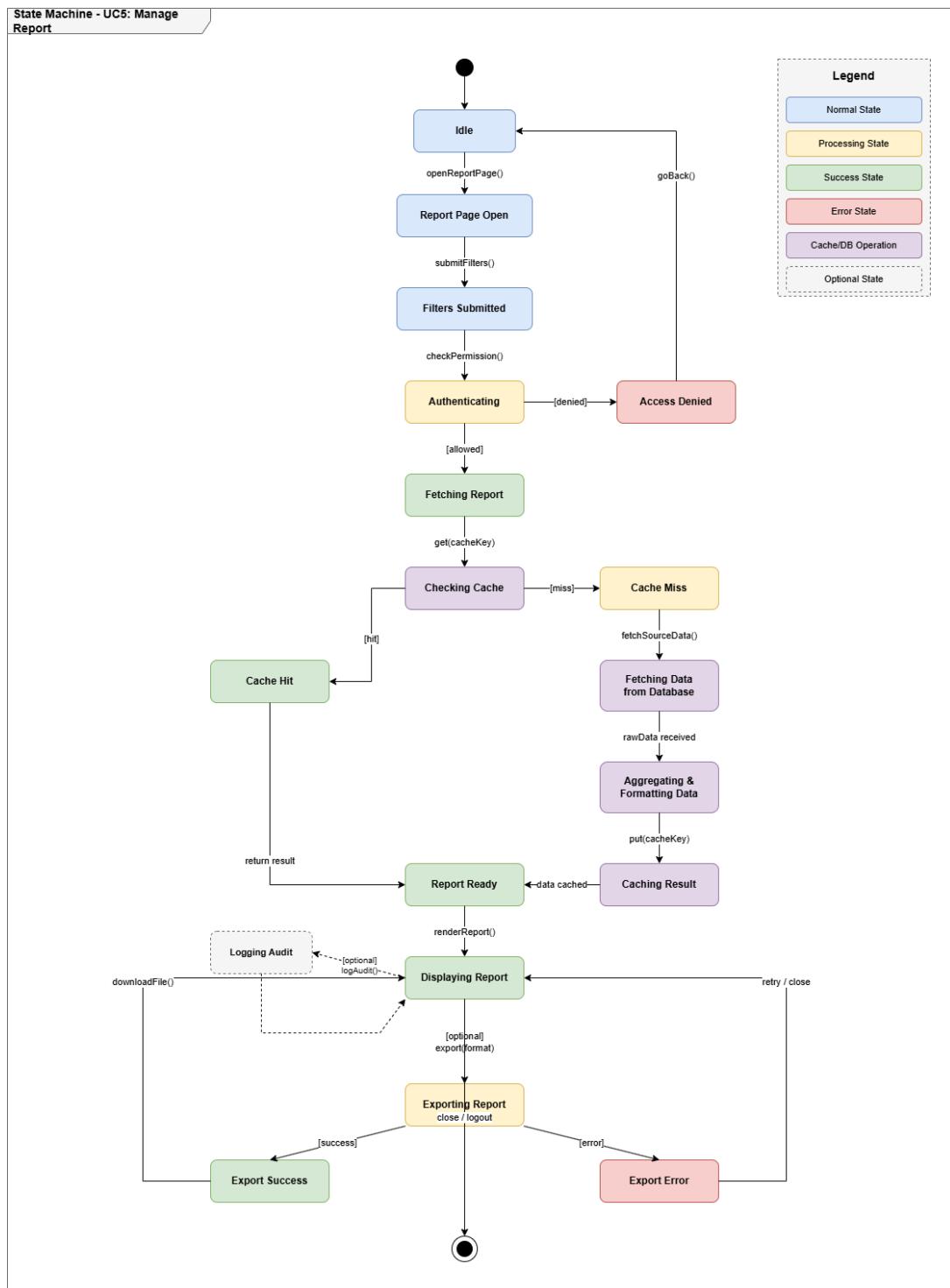


**Figure 12:** System Reminders



**Figure 13:** Cancellation and rescheduling

## 9.4 SD-04 Reporting & Analytics



**Figure 18:** Manage Reports

## 10 Mockup

### 10.1 MU-01 Log in, Log out & Profile Management

The landing page for the Tutor Support System is designed to provide users with a clear overview of its features and benefits. It includes a header with the university logo and navigation links, a main content area with sections for user guides, system integrations, and performance metrics, and a footer with contact information and legal disclaimers.

**Header:**

- TUTOR SUPPORT SYSTEM
- Ho Chi Minh City University of Technology (HCMUT)
- Home About Us Find a Tutor Book a Session Dashboard Community Profile

**Main Content Area:**

**Get the help you need, when you need it.**

Find a tutor, book a session, and track your learning progress — all integrated with HCMUT\_SSO and DATACORE.

**What you can do**

- Browse approved tutors by subject & department
- Smart matching based on availability and course needs
- Book, reschedule, and receive reminders
- View feedback and progress across sessions

**Integration**

Securely integrated with HCMUT\_SSO, DATACORE and Library. Single sign-on lets you access services with one login.

Please exit your browser after using authenticated services for security.

**Technical support**

E-mail: [support@hcmut.edu.vn](mailto:support@hcmut.edu.vn)  
Tel: (84-8) 38847256 - 7204

**About Us – Tutor Support System**

Tutor Support System is an academic support platform for HCMUT students: find suitable tutors, schedule support sessions, track progress and provide quality feedback. The system securely integrates with HCMUT\_SSO, DATACORE and library to ensure a unified experience within the university ecosystem.

**Who uses the system?**

Students	Tutors
<ul style="list-style-type: none"><li>Find tutors, book sessions, track learning progress.</li><li>Receive personalized learning suggestions (AI Personalized).</li><li>Submit feedback after sessions, join Forum/Q&amp;A.</li></ul>	<ul style="list-style-type: none"><li>Manage availability, accept support requests.</li><li>Record progress logs for each student.</li><li>Review feedback, improve teaching quality.</li></ul>
Coordinators	Departments / Admin
<ul style="list-style-type: none"><li>Điều phối tutor, xử lý xung đột/overbook.</li><li>Cấu hình Matching Rules, xem Matching Suggestions.</li><li>Báo cáo Workload/Utilization, theo dõi vấn đề feedback.</li></ul>	<ul style="list-style-type: none"><li>Departmental &amp; Participation reports.</li><li>SSO/Library health, notifications log, export jobs.</li><li>RBAC &amp; audit logs đảm bảo tuân thủ.</li></ul>

**Integrations**

- HCMUT\_SSO
- DATACORE
- Library

**Impact (mock)**

128	3,240
Active tutors	Sessions / term
4.7/5	14
Avg. rating	Programs covered

**Tutor Support System**

Academic support network for HCMUT students.  
Find tutors, book sessions, and get guided learning.  
Securely integrated with HCMUT\_SSO and DATACORE.

**Quick Links**

Find a Tutor  
Book a Session  
Your Dashboard  
Submit Feedback

**Support**

Academic Affairs  
Student Affairs  
Library Resources  
Report an Issue

© 2025 Tutor Support System – Ho Chi Minh City University of Technology (HCMUT)

Figure 9: Landing page

**TUTOR SUPPORT SYSTEM**  
Ho Chi Minh City University of Technology (HCMUT)

**Enter your Username and Password**

**Username**  
student@hcmut.edu.vn

**Password**  
\*\*\*\*\*

**Sign in**

**Languages**

- Vietnamese
- English

**Please note**

The Login page enables single sign-on to multiple websites at HCMUT. This means that you only have to enter your user name and password once for websites that subscribe to the Login page.

You will need to use your HCMUT Username and password to login to this site. The "HCMUT" account provides access to many resources including the HCMUT Information System, e-mail, ...

For security reasons, please Exit your web browser when you are done accessing services that require authentication!

**Technical support**

E-mail: [support@hcmut.edu.vn](mailto:support@hcmut.edu.vn)  
Tel: (84-8) 38847256 - 7204

**Tutor Support System**  
Academic support network for HCMUT students.  
Find tutors, book sessions, and get guided learning.  
Securely integrated with HCMUT\_SSO and DATACORE.

**Quick Links**

- Find a Tutor
- Book a Session
- Your Dashboard
- Submit Feedback

**Support**

- Academic Affairs
- Student Affairs
- Library Resources
- Report an Issue

© 2025 Tutor Support System – Ho Chi Minh City University of Technology (HCMUT)

**Figure 9:** Log in page

**My Profile**  
Update your personal information.

**Account Information**

<b>Full name</b> Nguyen Viet Trung	<b>Student ID</b> 2353xxxx
<b>Program</b> Computer Engineering (OISP)	<b>Department</b> Computer Science & Engineering
<b>Email (HCMUT)</b> 2353xxxx@hcmut.edu.vn	

**Personal Information**

<b>Phone</b> +84 901 234 567	<b>Personal Email</b> trung.nguyen@gmail.com
<b>Short Introduction</b> Computer Science student passionate about web development and AI.	

**Save changes**

**Figure 9:** View profile & Edit profile page

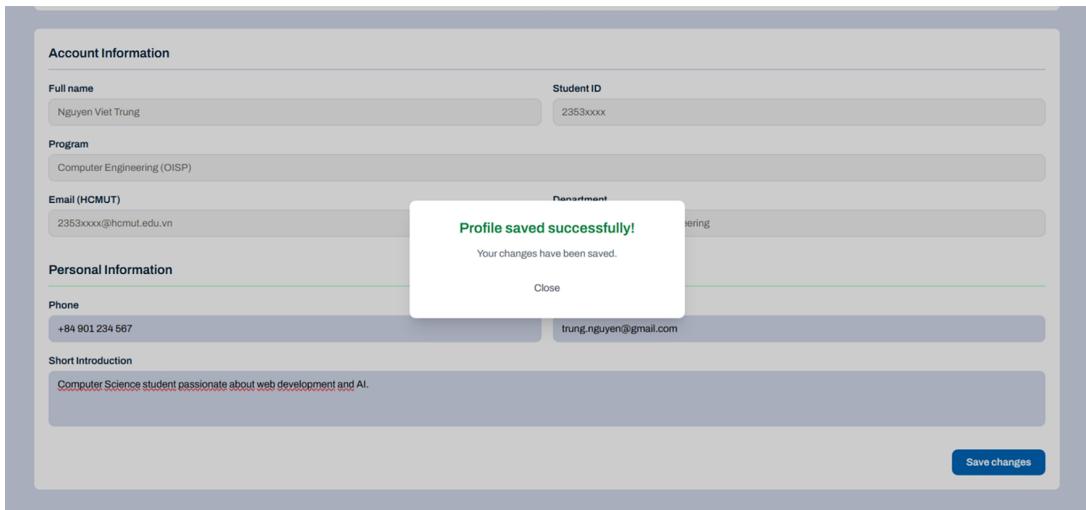


Figure 9: Save change Profile page

## 10.2 MU-02 Tutor-Student Matching

The screenshot shows the 'AI Tutor Match' section of the 'TUTOR SUPPORT SYSTEM'. At the top, there's a navigation bar with links for Learning, Feedback, Community, AI Tools, and Profile.

**AI Tutor Match**

We analyze your subject, schedule, and learning style to recommend the best tutors for you. You can still browse and book manually.

AI Suggestion • Last updated: just now • This does not auto-book.

**Tell us what you need**

FR-ADV01 Intelligent Matching

**Course / Topic**

CO1001 – Programming Fundamentals

Used to filter tutors with matching expertise.

**When can you study?**

Weekdays 18:00 – 21:00

We'll only show tutors who are free in this range.

**Your preferred style (optional)**

Step-by-step explanation + Q&A

Helps rank tutors whose style matches you.

**What are you struggling with?**

Example: I don't understand pointer-to-pointer and recursion stack frames. I freeze during lab check.

We'll use this (privately) to improve the match explanation.

**Generate AI Match**

Prefer to choose manually? [Browse all tutors](#) →

**Suggested Tutors** [Personalized ranking](#)

Top 3 matches based on availability, style, workload

**Nguyen T. A. Match #1**

**EXPERTISE**  
CO1001 - Pointers & Recursion

**NEXT AVAILABILITY**  
Wed 19:00 - Fri 20:00

**STYLE**  
Step-by-step, asks you to explain back

**Workload OK**

Confidence: High

Why we think this is a good match

- Available Wed after 18:00, same as you.
- Recently helped 3 students pass Lab 03 pointer debugging.
- Teaches using "you explain it back", which matches your preferred style.

**Request Session**

or [View profile](#)  
or [See timetable](#)

**Doan A. K. Match #2**

**EXPERTISE**  
Data Structures - Linked Lists

**NEXT AVAILABILITY**  
Sat 09:00 - Sun 10:30

**STYLE**  
Whiteboard simulation, timed questions

**Workload OK**

Confidence: Medium

Why we think this is a good match

- Strong at exam-style drilling under time pressure.
- You said you "freeze during lab check", this tutor focuses on confidence under pressure.
- Weekend morning slots match your "weekends morning" preference.

**Request Session**

or [View profile](#)  
or [See timetable](#)

**Truong Q. T. Match #3**

**EXPERTISE**  
Systems / Debugging live code

**NEXT AVAILABILITY**  
Fri 21:00 (online only)

**STYLE**  
Hands-on screen share & live fix

**High Load**

Confidence: Medium

Why we think this is a good match

- Specializes in debugging C code under real conditions.
- Prefers remote / live share — matches "lab debugging together".
- Warning: workload is high, session acceptance may take longer.

**Request Session**

or [View profile](#)  
or [See timetable](#)

FR-ADV01 Intelligent Matching: Ranked suggestions with explanation - User can still choose manually

Figure 10: AI Tutor-Student Matching page

The screenshot shows the 'Find a Tutor' section of the Tutor Support System. At the top, there is a search bar with placeholder text 'e.g. CO1001, Calculus I', a dropdown for 'Availability' set to 'Any time', and a dropdown for 'Department' set to 'All Departments'. Below the search bar are three tutor profiles:

- Nguyen T. A.**: High match - 82%.  
Senior student - CSE  
CO1001 - Programming Fundamentals  
Computer Science and Engineering  
Next available:  
Wed 14:00 - Online / B4-205 (2 left) | Fri 09:30 - Online (1 left)  
Status: Available  
Book session | View profile
- Pham Q. T.**: Good match - 74%.  
Tutor - Applied Math  
MA1001 - Calculus I  
Applied Mathematics  
Next available:  
Thu 09:00 - C2-301 (2 left) | Sat 10:15 - Online (no)  
Status: Available  
Book session | View profile
- Truong Q. T.**: Available soon.  
Tutor - EEE  
EE2002 - Digital Systems  
Electrical & Electronics Engineering  
Next available:  
Fri 16:30 - Lab D1 (1 left)  
Status: 1 slot this week  
Book session | View profile

**Figure 10:** Find a Tutor page

**TUTOR SUPPORT SYSTEM**  
COORDINATOR

Operations Tutors Students Reports

### Coordinator Assignment

Manually assign a tutor to a student. This will override any previous or pending matches. Only Coordinator / Dept Chair / Admin can do this. (FR-MAT.04)

- Assignment is logged with your coordinator ID. • Student and tutor will receive notifications.

**Student Information** Student

Student ID	2352525	Full name (from DATACORE)	Nguyen Manh Quoc Khanh
Search or enter the student who needs tutoring support.		Read-only from DATACORE.	
Faculty / Major	Computer Science and Engineering	Requested Support In	Programming Fundamentals (CO1001)
Course or skill area the student needs help with.			

**Tutor Selection** Tutor

Tutor	Nguyen T. A. · CO1001 · CS Faculty · 3 mentees
Pick a tutor. Their current workload and subject expertise is shown.	
Earliest Available Slot	Wed · 14:00 · B4-205
Pulled from tutor availability (FR-SCH.01).	
Coordinator Note / Reason	Urgent support before midterm; previous tutor is at capacity.
This reason will be logged in audit, visible to department chairs.	

**Override Policy**

- This assignment will override any existing pending match.
- Both student and tutor will receive notifications.
- This action is recorded with your Coordinator ID and timestamp.

**Assignment Summary**

Draft - Not saved FR-MAT.04

Student	2352525 — Nguyen Manh Quoc Khanh Computer Science and Engineering
Focus Course	CO1001 — Programming Fundamentals
Tutor	Nguyen T. A. · CO1001 · CS Faculty · 3 mentees
Earliest Slot	Wed · 14:00 · B4-205 / Online
Coordinator Reason	Urgent support before midterm; previous tutor is at capacity.

**Assign Now**

By assigning, you confirm this pairing is appropriate and acknowledge it will be logged.

HOMUT Tutor Support System · Coordinator View · Manual override of tutor-student match

**Figure 10:** Coordinator assignment page

The screenshot shows a 'Smart Match Suggestion' page from the Tutor Support System. It displays three identical profiles for a tutor named Nguyen T. A., who is teaching Programming Fundamentals (CO1001) at the Faculty of Computer Science. Each profile has a matching score of 98%. The profiles include sections for Availability Fit, Style & Focus, and Student Feedback. Both Availability Fit and Style & Focus mention a preference for 'Wednesday - 14:00 (B4-205 or Online)' and 'Afternoon / On campus'. The Student Feedback section notes a 'High clarity rating from first-year students who struggled with pointers.' At the bottom of each profile are 'Request This Tutor' and 'View Tutor Profile' buttons.

The footer navigation bar includes links for 'Tutor Support System', 'Quick Links' (Find a Tutor, Book a Session, Your Dashboard, Submit Feedback), and 'Support' (Academic Affairs, Student Affairs, Library Resources, Report an Issue). The footer also contains a copyright notice: '© 2025 Tutor Support System - Ho Chi Minh City University of Technology (HCMUT)'.

**Figure 10:** Smart match suggestion page

### 10.3 MU-03 Session Scheduling Management

#### Session Scheduling Management

The screenshot shows the 'Tutor Availability Setup' page. It displays a grid of weekly availability slots for Monday through Friday. Monday has two slots: '09:00 – 10:30' (Offline) and '14:00 – 15:30' (Online). Tuesday has 0 slots. Wednesday has one slot: '13:30 – 15:00' (Offline). Thursday has 0 slots. Friday has one slot: '09:00 – 10:30' (Online). Below the grid is a form to 'Add new slot' with fields for Day, Start Time, End Time, and Status (Online/Offline). A note at the bottom states: 'Your changes will be synchronized with HCMUT\_DATACORE. Students will only see available slots not conflicting with your official timetable.' A 'Save availability' button is at the bottom right.

**Figure 11:** Tutor published available time slots

**Session Details**

Tutor must have published an available slot before booking.

**Student Name / ID**  
Nguyen M. Q. Khanh · 2352525

**Choose Time Slot**  
Wed 14:00 - Online / B4-205

**Mode**  
In-person at campus

**What do you want to work on?**

**Heads up**  
You already have a session near this time. You can still continue, but system may warn about back-to-back sessions.

By clicking "Request Session", the system will hold this slot, notify the tutor, and send reminders (24h & 1h before).

**Booking Summary**

Draft · Not submitted

**Tutor**  
Nguyen T. A.  
CO1001 – Programming Fundamentals

**Selected slot**  
Wed 14:00 - Online / B4-205

**Mode**  
In-person at campus

**Goal for session**  
— (no note)

**Policy**

- Reschedule/cancel ≥ 2h before start.
- System prevents double-booking of the same slot.
- Notifications sent to student & tutor automatically.

Figure 12: Student book session page

**Upcoming & Pending**

Your next bookings, including those waiting for tutor approval.

**Wed - 14:00 - 15:30 - Room B4-205 / Online**

**Tutor**  
Nguyen T. A.  
Programming Fundamentals (CO1001)

**Mode**  
In-person or Online

**Status**  
Tutor accepted · Scheduled

**Your request to tutor**  
I need help with recursion and pointer debugging from lab 03. I'm getting segmentation faults.

**Reschedule**

**Cancel Session**

Can reschedule / cancel up to 2h before start (FR-SCH.03).

**Fri - 09:00 - 10:30 - Online**

**Tutor**  
Doan A. K.  
Data Structures

**Mode**  
Online only

**Status**  
Waiting · You're in queue

**Your request to tutor**  
I need a whiteboard-style walkthrough of linked lists. Please test me after each step.

**Awaiting Tutor Confirmation**

**Cancel Request**

Can reschedule / cancel up to 2h before start (FR-SCH.03).

Figure 13: Cancellation and rescheduling page

**Notifications & reminders**

Your system messages, reminders, and alerts.

**Reminder**  
Your session #S-2002 starts in 2 hours.  
Today 09:33

**Feedback posted by tutor for session #S-1994**  
Yesterday 13:01

**Scheduled update**  
CO1001 office-hour moved to Wed 15:30.  
Yesterday 09:15

Figure 14: System sends notification reminders

**Upcoming & Pending**

Wed - 14:00 - 15:30 - Room B4-205 / Online	Mode	Status	Confirmed	Reminder
Tutor: Nguyen T. A. Programming Fundamentals (CO1001)	In-person or Online	Tutor accepted - Scheduled		

Fri - 09:00 - 10:30 - Online	Mode	Status	Awaiting Tutor Confirmation	
Tutor: Nguyen T. A. Programming Fundamentals (CO1001)	In-person or Online	Tutor accepted - Scheduled		

**Completed Sessions**

Wed - 14:00 - 15:30 - Room B4-205 / Online	Mode	Status	Needs Feedback	Submit Feedback
Tutor: Nguyen T. A. Programming Fundamentals (CO1001)	In-person or Online	Tutor accepted - Scheduled		

Fri - 09:00 - 10:30 - Online	Mode	Status	View Session Summary	
Tutor: Nguyen T. A. Programming Fundamentals (CO1001)	In-person or Online	Tutor accepted - Scheduled		

Figure 14: My registered sessions page

**Session Detail**

**Session Information**

Date & Time Wed - 14:00 - 15:30	Location / Mode Room B4-205 (on campus) or Online (via Google Meet)
Tutor Nguyen T. A. Faculty of Computer Science "We go through your code step by step."	Status Confirmed by tutor Reminder already sent to you

**Your Request / Goal for this Session**

I need help with recursion and pointer debugging from lab 03. I keep getting segmentation faults and I don't understand stack frames.

**Tutor Notes / Session Summary**

We'll walk through your code together. You will explain each pointer manipulation line-by-line. We'll also prepare 2 short "exam-style" recursion questions.

**Cancellation Policy**

You can cancel or reschedule this session up to 2 hours before start. After that, last-minute cancellations may be recorded for review by the coordinator.

**Actions**

**Reschedule / Cancel**  
Move this session or cancel it if you can't attend. Must be > 2h before start.

**Reschedule Session**

**Cancel Session**

**After the Session**

When this session is completed, you'll be asked to submit feedback about the tutor's performance. If you don't respond in time, it will be marked "Feedback Skipper".

**Submit Feedback**

Figure 14: Session details page

**Tutor Support System**  
Academic support network for HCMUT students.  
Find a Tutor | Book a Session | Dashboard | Community | Profile

**Quick Links**  
Find a Tutor | Book a Session | Your Dashboard | Submit Feedback

**Support**  
Academic Affairs | Student Affairs | Library Resources | Report an Issue

## 10.4 MU-04 Feedback & Progress Tracking

**Submit Feedback**

This feedback will be linked to your tutoring session and used for quality improvement. Only aggregated results are shared with departments.

**Session Summary**

Mon · Oct 27 · 09:00 – 10:30 · C2-301  
Tutor: Pham Q. T. · Focus: Midterm-style timed drills  
Your goal (from booking): "I keep getting segmentation faults and I don't understand stack frames."

**Overall helpfulness**  
5 - Extremely helpful

**Clarity of explanation**  
Very clear

Did you understand the concepts better?

This is used in tutor quality reports.

**What helped you most?**  
Example: They made me trace pointer values step by step and explain out loud, which made it click.

We may quote this (anonymized) to improve tutor training.

**Anything we should improve for next time?**  
Example: I wish we had 15 more minutes to finish the last recursion question.

If you report serious issues (behavior, respect, safety), coordinators will be alerted.

**Save Draft** **Submit Feedback**

Figure 15: Submit feedback page

**Tutor Progress Log**

After each tutoring session, record the mentee's progress. This is visible to coordinators and used for department reports.

**SESSIONS TO LOG**

- Nguyen M. Q. Khanh** Completed  
CO1001 - Programming Fundamentals  
Wed, Oct 27 - 14:00 - 15:30 - Room B4-205 / Online
- Tran H. Minh** Completed  
MA1001 - Calculus I  
Tue, Oct 26 - 09:00 - 10:15 - C2-301
- Le T. Cam Tu** In progress  
EE2002 - Digital Systems  
Tue, Oct 26 - 18:30 - 18:00 - Lab D1

Once submitted, this log is visible to coordinator / department. You can edit within 24h.

**Understanding level**  
Excellent - fully grasped

**Engagement**  
Highly engaged

Used for student progress analytics.

**Session summary**  
Student could trace recursion stack correctly by the end. Needs follow-up on pointer-to-pointer usage.

This will appear in student dashboard + coordinator reports.

**Next recommendation / plan (optional)**  
Example: practice recursion problems 7-10, review memory model next session.

**Attach file (optional)**  
Click to upload notes or summary files (PDF, DOCX)

Attach lab notes, summary, or exercise sheets used in this session.

FR-FBK.02 - Progress Recording - Auto-timestamped - Editable 24h

**Save draft** **Submit progress log**

Figure 16: Tutor progress log page

**Completed Sessions**  
Fill in feedback to help improve tutoring quality.

**Mon · Oct 27 · 09:00 – 10:30 · C2-301**

Tutor Pham Q. T. Calculus I (MA1001)	Focus Midterm-style timed drills	Status Completed - Feedback pending
--	-------------------------------------	--

**Session summary**  
Practiced derivative problems under exam timing. You were asked to explain each step.

**Sat · Oct 18 · 08:30 – 10:00 · Online**

Tutor Truong Q. T. Digital Systems (EE2002)	Focus Timing diagrams, flip-flop troubleshooting	Status Completed - Feedback deadline passed
---	---	--

**System note**  
Marked as "Feedback Skipped" and visible to department analytics for summary only.

Your feedback is linked to this session (FR-FBK-01).

Your feedback is linked to this session (FR-FBK-01).

**Needs Feedback** **Submit Feedback** **View Session Summary**

Figure 17: Completed session page

## 10.5 MU-05 Reporting & Analytics

**TUTOR SUPPORT SYSTEM** DEPARTMENT CHAIR **Reports**

**Participation report**  
Filter by cohort/program, set thresholds.

2023 CSE Min sessions per student 2

Class	# Students	Participated
K23-CSE-01	62	41
K23-CSE-02	58	35

**Export report**

Figure 18: Participation report page

**TUTOR SUPPORT SYSTEM** STUDENT AFFAIRS **Dashboard** **Reports** **Export**

**Student Affairs Dashboard**  
Term 2025-1 · Today: 2/11/2025 **Participation** **Eligibility**

**Program Overview**

Total Participants <b>342</b> +12% vs Last Term	Eligible Students <b>287</b> +8% vs Last Term	Sessions Completed <b>1456</b> +15% vs Last Term	Avg Feedback Rate <b>89%</b> +3% vs Last Term
---	---	--	---

**System Alerts**

- ⚠️ 3 tutors have overbooked sessions this week.
- 💡 12 students have not submitted session feedback yet.
- 💡 5 students are near eligibility threshold (need 1 more session).

**Quick Actions**

- View Participation Report**
- Check Eligibility**
- Export Reports**
- Export Summary Report**

**Recent Activity**

- Participation report exported 2 hours ago By: sa\_admin
- Eligibility rules updated 5 hours ago By: sa\_admin
- PDF report generated 1 day ago By: sa\_staff

**View All Logs**

**Figure 18:** Student Affairs dashboard page

The screenshot shows the 'Eligibility / Credits' section of the dashboard. It includes fields for 'Minimum Sessions Attended' (set to 4) and 'Minimum Feedback Completion (%)' (set to 75), with a 'Recalculate Eligibility' button. Below this is a table titled 'Eligible Students (5)' listing student IDs, names, sessions attended, and feedback percentages. A summary statistics box indicates 8 total students, 5 eligible, and 3 not eligible.

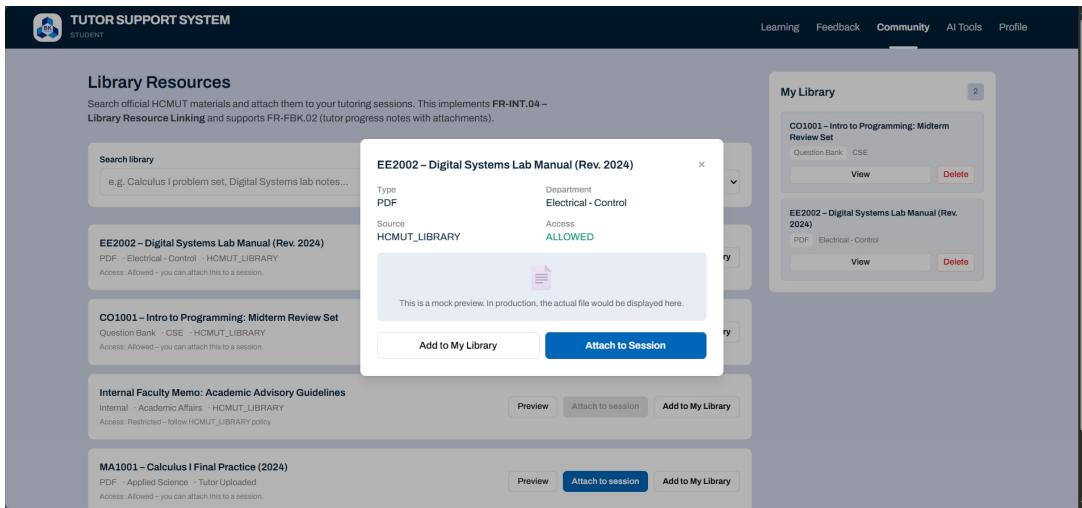
Student ID	Name	Sessions	Feedback (%)
2252001	Nguyen Van A	7	100%
2252003	Le Minh K	5	80%
2252005	Hoang Tuan L	6	83%
2252006	Nguyen Anh T	4	75%
2252007	Vo Thanh C	8	88%

**Figure 18:** Eligibility & Credits page

## 10.6 MU-06 Integration with HCMUT Infrastructure

The screenshot shows the 'Library Resources' section of the system. It lists several documents: 'EE2002 – Digital Systems Lab Manual (Rev. 2024)', 'CO1001 – Intro to Programming: Midterm Review Set', 'Internal Faculty Memo: Academic Advisory Guidelines', and 'MA1001 – Calculus I Final Practice (2024)'. Each item has 'Preview', 'Attach to session', and 'Add to My Library' buttons. To the right, a 'My Library' sidebar shows two items: 'CO1001 – Intro to Programming: Midterm Review Set' and 'EE2002 – Digital Systems Lab Manual (Rev. 2024)', each with a 'View' and 'Delete' button.

**Figure 18:** Library resource search page



**Figure 18:** Attachment to session page

## 10.7 MU-07 Study & Advanced Features

**Figure 18:** Personalized study plan page

The screenshot shows a document view page from a 'TUTOR SUPPORT SYSTEM'. At the top, there's a navigation bar with icons for user profile, Learning, Feedback, Community, AI Tools, and Profile. Below the header, a breadcrumb trail shows 'Back to Library' and the current document title: 'EE2002 - Digital Systems Lab Manual (Rev. 2024) PDF - Electrical - Control'. The main content area features the title 'Introduction to Programming' and code 'CO1001'. Below the title is a section titled 'Lecture Notes & Lab Manual' with the note 'Revision 2024'. A 'Course Overview' box contains a brief description of the course goals. A 'Learning Objectives' box lists two items: 'Understand basic programming concepts and problem-solving techniques' and 'Master C programming syntax and semantics'.

Figure 18: Document view page

This screenshot shows the 'Table of Contents' page for the same document. On the left, a sidebar displays a 'Table of Contents' with five chapters: Chapter 1: Introduction (Page 1), Chapter 2: Basic Concepts (Page 15), Chapter 3: Advanced Topics (Page 32), Chapter 4: Applications (Page 48), and Chapter 5: Exercises (Page 85). The main content area is identical to the previous screenshot, showing the course title, revision, and overview sections.

Figure 18: Table of contents page

The screenshot shows the 'Community Forum / Q&A' section of the Tutor Support System. At the top, there are dropdown menus for 'All course' and 'All status', and a search bar with placeholder text 'Find by title or keyword...'. Below this is a 'Create new question' form with a title 'CO1001 – Question about assignment' and a text area containing the message 'Can I delete all and do from scratch?'. A 'Post Question' button is at the bottom right of the form. To the right of the form is a 'Moderation' button. The main content area displays a 'Question List (9)' with the following items:

- Pinned CO1001 – How to pass the last test case?** by 2353xxxx · Today 10:12 · 5 replies
- CO1001 – Question about assignment** by you · just now · 0 replies
- CO1001 – Question about assignment** by you · just now · 0 replies
- a CO1001** by you · just now · 0 replies
- a CO1001** by you · just now · 0 replies
- CO1001 – Question about pointer and array** by 2350xxxx · 1 week ago · 12 replies
- PH1001 – Conservation of momentum law** by 2251xxxx · 3 days ago · 8 replies
- MA1001 – Differential homework week 5** by 2278xxxx · 2 days ago · 0 replies
- EE2002 – Any quick tips for Karnaugh map?** by 2252xxxx · Yesterday 21:03 · 2 replies

Figure 18: Community forum page

The screenshot shows the same 'Community Forum / Q&A' section after a question has been posted. A modal window in the center displays a green checkmark icon and the message 'Question posted successfully! Your question has been published to the forum.' with a 'Close' button. The rest of the page is identical to Figure 18, showing the question list and the creation form.

Figure 18: Successful question post page

The screenshot shows a forum post titled "CO1001 - How to pass the last test case?". The post was asked by user 2353xxxx and has 5 replies. The first reply from tutor\_co1001 suggests using fgets or read line by line then parse. The second reply from 2353xxxx says they'll try it. The third reply from 2354xxxx says they had the same issue and using fgets + sscanf worked for them. There is a "Post Reply" button at the bottom.

**Figure 18:** Reply forum's question page

The screenshot shows the "AI Quiz Generator" page. It has tabs for "Generator", "My Questions", and "Analysis", with "Generator" selected. The "Your request" section asks for practice questions based on topic, difficulty, and confusion notes. It includes fields for "Question source" (set to "Trusted bank (recommended)"), "Topic / Course" (set to "CO1001 - Recursion / Pointer Debugging"), "Difficulty" (set to "Intro / warm-up"), and "What confuses you most?" (with an example note about pointer references). A "Generate Quiz" button is at the bottom.

**Figure 18:** AI Quiz generator (1) page

The screenshot shows the 'AI Quiz Generator' page of the Tutor Support System. At the top, there's a navigation bar with links for Learning, Feedback, Community, AI Tools, and Profile. Below the navigation is a title 'AI Quiz Generator' and a subtitle 'Generate practice questions based on your topic, difficulty, and confusion notes.' There are three tabs: 'Generator' (selected), 'My Questions', and 'Analysis'. A 'Your request' section asks for a description of what the user is struggling with. Two options for 'Question source' are shown: 'Trusted bank (recommended)' (selected) and 'AI creative'. Below these are dropdowns for 'Topic / Course' (CO1001 – Recursion / Pointer Debugging) and 'Difficulty' (Intro / warm-up). A note says 'Higher difficulty simulates midterm/final.' A 'What confuses you most?' text area contains the placeholder 'Example: I keep losing track of pointer references inside recursive calls when they return new addresses.' A note below it says 'This will bias the quiz to force you to explain that exact weak spot.' A 'Generate Quiz' button is at the bottom left of this section. To the right, a large box displays 'Last generated (3 questions)'. It lists three questions with multiple-choice answers and a 'Bank' button next to each. The first question is 'Q1. What is the base case in a recursive function?'. The second is 'Q2. What happens when you dereference a NULL pointer in C?'. The third is 'Q3. Where is local variable data stored during function execution?'. At the bottom of this box, a note says 'Generated quizzes are automatically saved to "My Questions" page.'

**Figure 18:** AI Quiz generator (2) page

The screenshot shows the 'My AI Quizzes' page. At the top, there's a navigation bar with links for Learning, Feedback, Community, AI Tools, and Profile. Below the navigation is a title 'My AI Quizzes' and a subtitle 'Review generated questions. Skip ones you don't like, or solve them to track your progress.' There are three tabs: 'Generator' (disabled), 'My Questions' (selected), and 'Analysis'. A 'Filters' section allows filtering by Status (ALL), Topic (ALL), and Source (ALL). Below this, a message says '3 questions'. Three quiz cards are listed: 1. 'What is the base case in a recursive function?' (Topic: CO1001 – Recursion / Pointer Debugging, Difficulty: Intro / warm-up, Source: Trusted Bank, Date: 11/2/2025, 2:54:47 PM). 2. 'What happens when you dereference a NULL pointer in C?' (Topic: CO1001 – Recursion / Pointer Debugging, Difficulty: Intro / warm-up, Source: Trusted Bank, Date: 11/2/2025, 2:54:47 PM). 3. 'Where is local variable data stored during function execution?' (Topic: CO1001 – Recursion / Pointer Debugging, Difficulty: Intro / warm-up, Source: Trusted Bank, Date: 11/2/2025, 2:54:47 PM). Each card has 'Solve', 'Skip', and 'Delete' buttons.

**Figure 18:** My AI Quizzes page

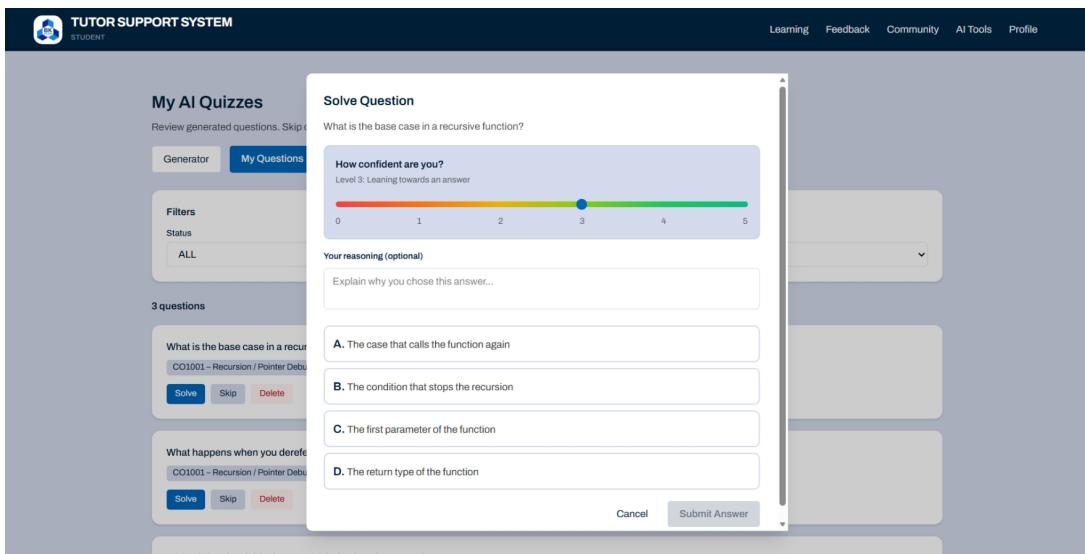


Figure 18: Solve question page (1)

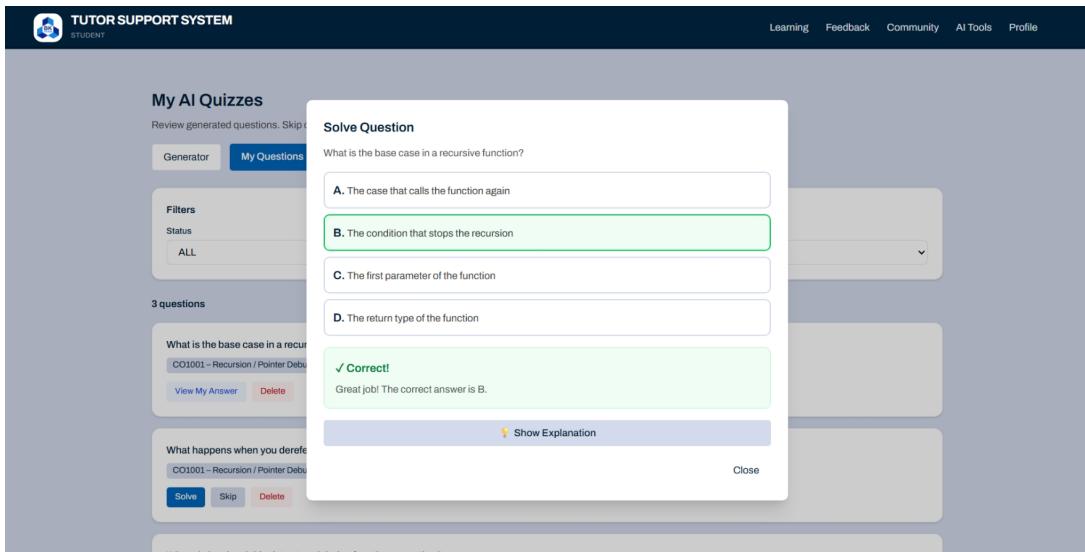
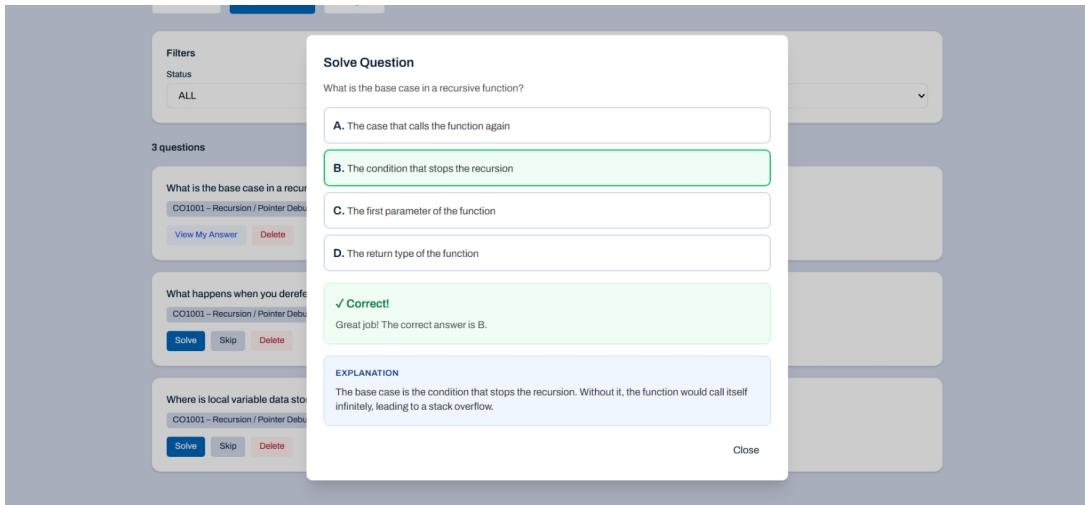


Figure 18: Solve question page (2)-answer



**Figure 18:** Solve question page (3)-explanation

The screenshot displays the 'Quiz Analysis' section of the Tutor Support System. At the top, there are summary statistics: 'TOTAL GENERATED' (3), 'SOLVED' (1), 'CORRECT' (1), 'SKIPPED' (0), and 'TRUSTED VS AI' (1/0). Below this is a 'Confidence Level Analysis' section with three categories: 'LOW CONFIDENCE (0-1)' (0 attempts yet), 'MID CONFIDENCE (2-3)' (1 attempt, 100% correct), and 'HIGH CONFIDENCE (4-5)' (0 attempts yet). A detailed breakdown of confidence levels follows: 0: No Idea (0), 1: Wild Guess (0), 2: Unsure (0), 3: Leaning (1, 100% correct), 4: Confident (0), and 5: Absolutely (0). An 'AI RECOMMENDATIONS' section is present but empty. The next section, 'Weak Topics (lowest accuracy first)', shows a single topic: 'CO1001 – Recursion / Pointer Debugging' with 'Generated: 3', 'Solved: 1', 'Correct: 1', and 'Skipped: 0', resulting in '100% accuracy'. The 'Source Effectiveness' section compares 'Trusted Bank' (100% solved) and 'AI Freeform' (0% solved). Finally, the 'Difficulty Breakdown' section shows 'Intro / warm-up' with '1/3 solved'.

**Figure 18:** Quiz analysis page

## 10.8 MU-08 Admin Tools

The screenshot shows the Admin Dashboard of the Tutor Support System. At the top, there are four status boxes: SSO Status (Online, green), Last Datacore Sync (5 mins ago, blue), Pending Exports (2, orange), and Audit Events (134, gray). Below this is a section for Recent Exports, listing three reports: Departmental Report (CSV) E-090 (Done), Participation Report (PDF) E-091 (Queued), and Audit Logs (CSV) E-092 (Processing). The final section is Integrations, showing three entries: HCMUT\_SSO (OK, last checked 2025-11-02 10:05), DATACORE (OK, last checked 2025-11-02 10:03), and HCMUT\_LIBRARY (DEGRADED, last checked 2025-11-02 09:58).

Figure 18: Admin dashboard page

The screenshot shows the Audit Logs page. It features a filter section with fields for Actor (user or system) and Event Type (set to All Events). Below this is a table titled "Audit Events (8)" with columns: Time, Actor, Event, Target, and Details. The events listed are:

Time	Actor	Event	Target	Details
2025-11-02 10:10	admin@hcmut.edu.vn	EXPORT	Departmental CSV	Job E-091 created
2025-11-02 10:05	system	SYNC	DATACORE	34 profiles updated
2025-11-02 09:55	coord01	ROLE_CHANGE	student 2352525	Role -> TUTOR (temporary)
2025-11-02 09:40	admin@hcmut.edu.vn	EXPORT	Audit Logs CSV	Job E-092 created
2025-11-02 09:30	2352525@hcmut.edu.vn	LOGIN	Student Portal	SSO authentication successful
2025-11-02 09:15	system	SYNC	HCMUT_LIBRARY	12 resources indexed
2025-11-02 08:50	coord01	ROLE_CHANGE	student 2353001	Role -> COORDINATOR (temporary)
2025-11-02 08:30	admin@hcmut.edu.vn	LOGIN	Admin Portal	SSO authentication successful

Figure 18: Audit logs page

**Integrations**

Monitor and test connections to HCMUT\_SSO, DATACORE, and HCMUT\_LIBRARY.

Integration	Status	Last sync
HCMUT_SSO	Online	2025-11-02 10:05
DATACORE	Online	2025-11-02 10:03
HCMUT_LIBRARY	Degraded	2025-11-02 09:58

**Last Integration Errors**

Integration	Error Message	Timestamp
HCMUT_LIBRARY	403 - access restricted for course resources	2025-11-02 09:58

Figure 18: Integrations page

**Exports & Reports**

Recent exports and their statuses. You can trigger new exports for Departmental, Participation, and Audit reports.

**Create New Export**

- New: Departmental (CSV)
- New: Participation (PDF)
- New: Audit Logs (CSV)

**Export Jobs**

Job ID	Name	Requested at	Status	Actions
E-090	Departmental Report (CSV)	2025-11-01 21:15	Done	Download
E-091	Participation Report (PDF)	2025-11-02 08:05	Queued	Cancel
E-092	Audit Logs (CSV)	2025-11-02 09:40	Processing	Cancel
E-093	System Errors (JSON)	2025-11-02 10:05	Failed	Retry

Figure 18: Export & Report page

**Users & Roles**

Inspect current roles (Student, Tutor, Coordinator, Department Chair, Program Admin) and local overrides.

**Search Users**

**Sync roles from DATACORE (mock)**

**Users (5)**

User	Email / ID	Current Role(s)	Source	Actions
2352525 – Khanh	2352525@hcmut.edu.vn	Student	DATACORE	<a href="#">View</a> <a href="#">Add role</a> <a href="#">Remove role</a>
coord01 – Student Affairs	coord01@hcmut.edu.vn	Coordinator, StudentAffairs	Local Override	<a href="#">View</a> <a href="#">Add role</a> <a href="#">Remove role</a>
admin	admin@hcmut.edu.vn	ProgramAdmin	Local	<a href="#">View</a> <a href="#">Add role</a> <a href="#">Remove role</a>
2353001 – Minh	2353001@hcmut.edu.vn	Student, Tutor	Local Override	<a href="#">View</a> <a href="#">Add role</a> <a href="#">Remove role</a>
dept01 – CS Dept Chair	dept01@hcmut.edu.vn	DepartmentChair	DATACORE	<a href="#">View</a> <a href="#">Add role</a> <a href="#">Remove role</a>

Figure 18: Users & Roles page

The screenshot shows the 'System Tasks' page of the 'TUTOR SUPPORT SYSTEM'. At the top, there is a navigation bar with links for Dashboard, Exports, Integrations, Audit, Users & Roles, and System Tasks. The 'System Tasks' link is highlighted.

The main content area is titled 'System Tasks' and includes a sub-instruction: 'View and run scheduled tasks such as DB cleanup, log archiving, and backups.'

**Scheduled Tasks**

Task	Schedule	Last run	Status
Daily DB cleanup	Everyday at 03:00	2025-11-02 03:01	Success
Incremental backup	Every 15 minutes	2025-11-02 10:00	Success
Full backup	Everyday at 01:00	2025-11-02 01:02	Success

**Task History**

Time	Task	Result
2025-11-02 03:01	Daily DB cleanup	OK (12 MB freed)
2025-11-02 01:02	Full backup	OK (stored in dc-hcmut-bucket-02)
2025-11-01 03:01	Daily DB cleanup	OK (9 MB freed)

**Figure 18:** System tasks page

## 11 System Integration

- **HCMUT\_SSO**: OAuth2/OIDC-based single sign-on.
- **HCMUT\_DATACORE**: Read-only sync of personal data (name, ID, role, faculty, email).
- **HCMUT\_LIBRARY**: Link and share official materials within sessions.

## 12 Coding Rules and Constraints

To align with prior functional-programming constraints used in course assignments, if applicable:

- Only allowed imports per assignment rules.
- Prefer pure functions; avoid global state.
- Prefer higher-order functions and list comprehensions over loops.
- Single-assignment variables within functions to encourage immutability.

(Adapt or remove this section if your instructor does not require functional-programming constraints.)

## 13 Submission and Deliverables

### Deliverables:

- PDF generated from this L<sup>A</sup>T<sub>E</sub>X project.
- Any supporting diagrams (PNG/PDF) in the `images/` folder.

### Submission notes:

- Ensure the document compiles on Overleaf without custom fonts or non-standard packages.

## 14 Other Regulations

- Work must be original and comply with academic integrity policies.
- Instructor decisions are final.
- Post-grading test cases or rubrics may be summarized but not fully disclosed.

## 15 Changelog

Version	Notes
1.0	Initial draft tailored for Tutor Support System ( 26 September 2025).
1.1	Functional Requirement, Non-Functional Requirement and General Use-case diagram (24 September 2025).

- 1.2 Detail Use-case diagram and Table, Non-Interactive Functional Requirement and Fix 1.1 log submission(01 October 2025).
- 2.0 Complete State, Activity, Sequence and Mockup diagrams (01 November 2025).
-