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

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, DataCore, Library).
- Produce a consistent, submission-ready specification document in LaTeX.

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 DATACORE.
  - *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-N-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:

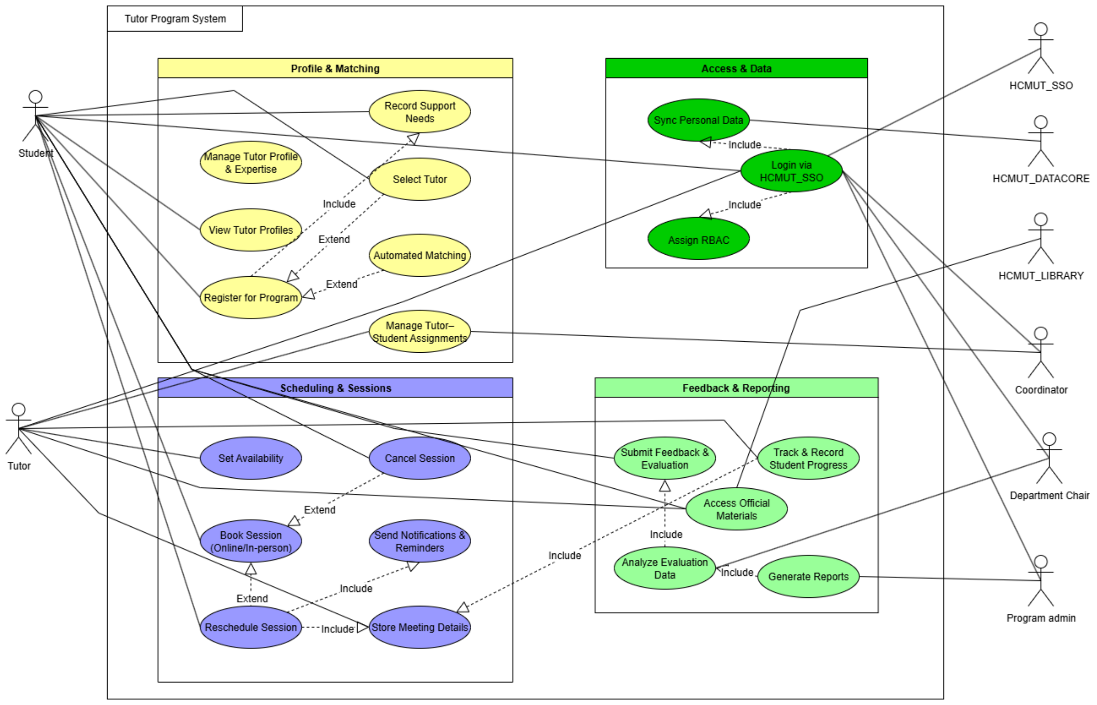

Figure 1: General Tutor program system

6.2 UC-01 Log in & Profile Management

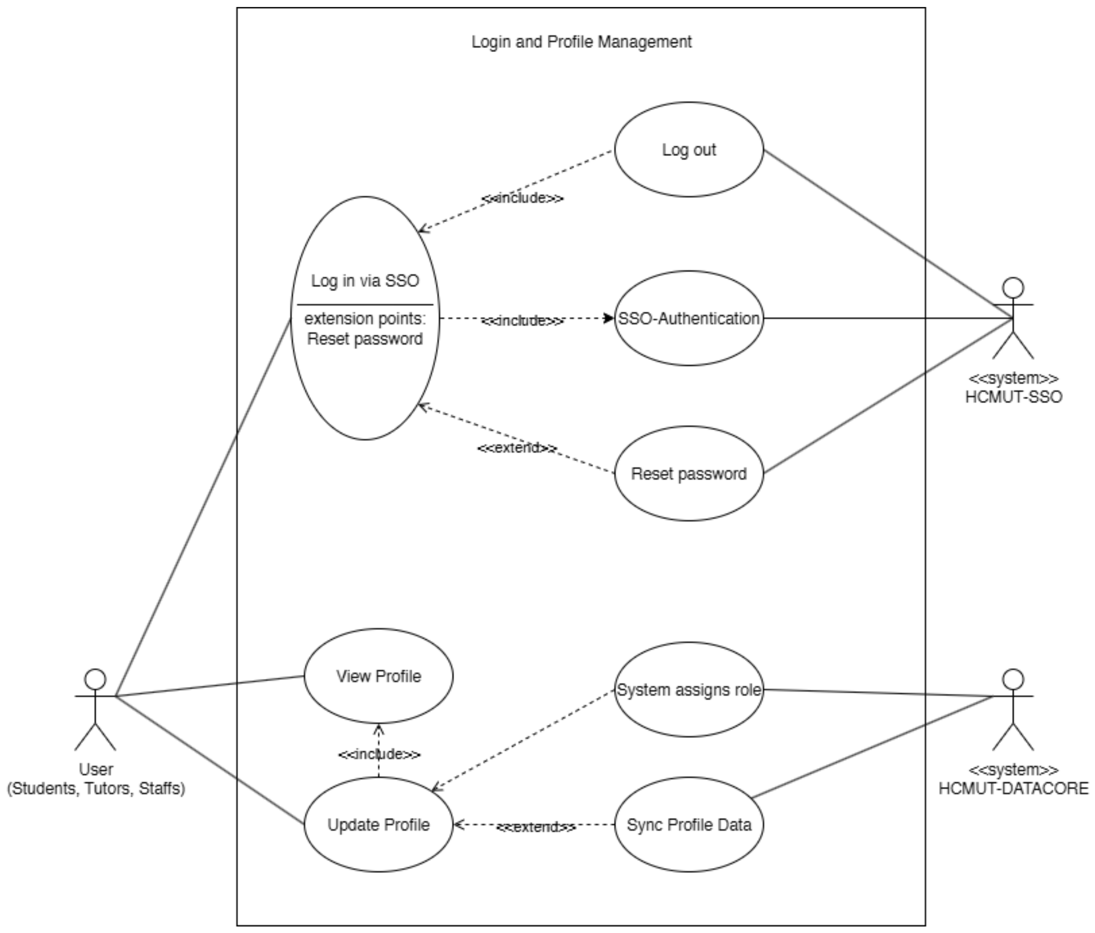

Figure 2: Log in and Manage Profile

Use-case ID	UC-01a
Use-case name	Login via SSO
Use-case overview	To allow students, tutors, and staff to log in securely via HCMUT_SSO with role assignment handled by the system.
Actors	User (Students, Tutors, Staff), HCMUT_SSO, System
Preconditions	1. User has valid SSO credentials. 2. HCMUT_SSO service is available.
Trigger	User selects the "Login via SSO" option.
Steps	1. User initiates login via HCMUT_SSO. 2. System sends authentication request to SSO service. 3. HCMUT_SSO validates credentials. 4. On success, the system fetches user data and assigns role. 5. On failure, the system denies access.
Postconditions	User is authenticated and session established; role assignment is ready for system use.
Alternative Flows	A1: Invalid login → Access denied with error message. A2: Role update in DATACORE synced during login.
Exception Flow	1. SSO service unavailable → System shows maintenance/unavailable message. 2. Network error → User prompted to retry login.

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

Use-case ID	UC-01b
Use-case name	Profile Management
Use-case overview	To allow students and tutors to view and update their profiles, with core data synchronized from HCMUT_DATACORE.
Actors	Student, Tutor, HCMUT_DATACORE, System
Preconditions	1. User is authenticated via SSO. 2. Profile data exists in DATACORE.
Trigger	User selects the "View/Update Profile" option.
Steps	1. System retrieves profile information from DATACORE. 2. User views profile fields (ID, name, email, faculty, role). 3. User updates non-core profile details. 4. System validates and saves changes. 5. System syncs updated data with DATACORE.
Postconditions	User profile is updated and synchronized with DATACORE; changes are timestamped and logged.
Alternative Flows	A1: DATACORE unavailable → Updates stored locally until sync resumes.
Exception Flow	1. Invalid update request → System rejects and shows error. 2. Sync conflict with DATACORE → DATACORE treated as source of truth.

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

6.3 UC-02 Tutor–Student Matching

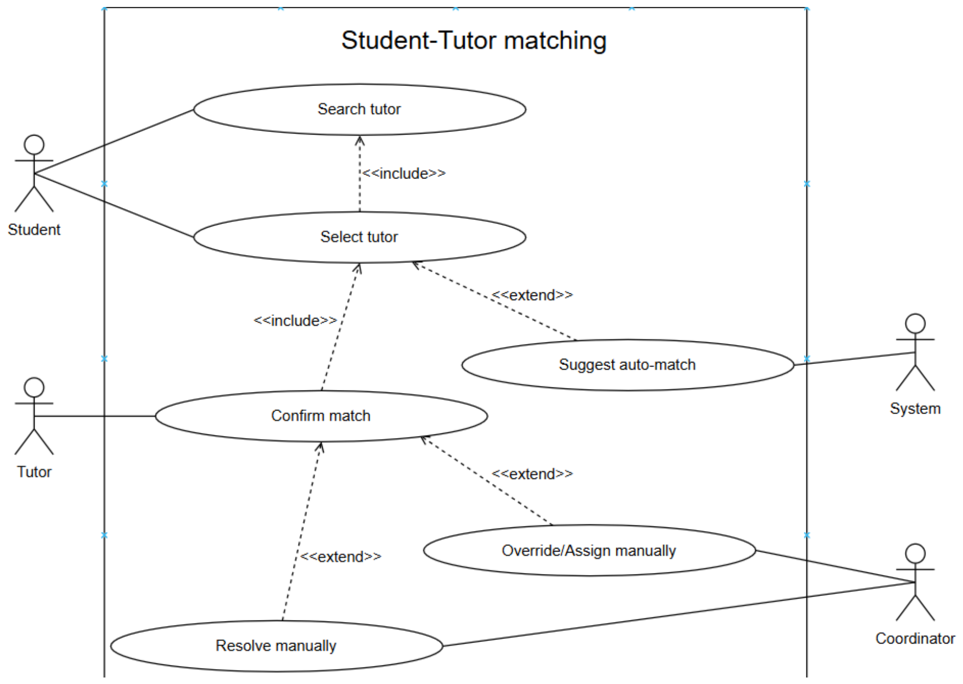

Figure 3: Tutor–Student Matching

Use-case ID	UC-02
Use-case name	Tutor–Student Matching
Use-case overview	To allow students to search and select tutors manually or request an automated match, with tutor confirmation and coordinator intervention when necessary.
Actors	Student (primary), Tutor, Coordinator, System
Preconditions	1. Student and tutor profiles exist in the system. 2. The system is operational and accessible. 3. Student is authenticated in the system.
Trigger	Student initiates a search for tutors or requests an auto-match.
Steps	1. Student searches for tutors by subject, availability, or preferences. 2. Student selects a tutor; a pending match is created. 3. System may suggest an auto-match (ranked list) based on the student's criteria. 4. Tutor reviews the pending match and confirms the match. 5. If confirmed, the system finalizes and logs the pairing.
Postconditions	A tutor–student pairing is established and logged in the system.
Alternative Flows	1. Auto-match rejected by student or tutor → Coordinator resolves manually. 2. No tutors found → System suggests broadening search criteria. 3. Tutor does not respond within time limit → Coordinator is notified to assign a tutor.
Exception Flow	1. Network failure prevents search or confirmation (system prompts user to retry). 2. Tutor or student profile missing/corrupted → System logs an error and notifies Coordinator.

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

6.4 UC-03 Session Scheduling Management

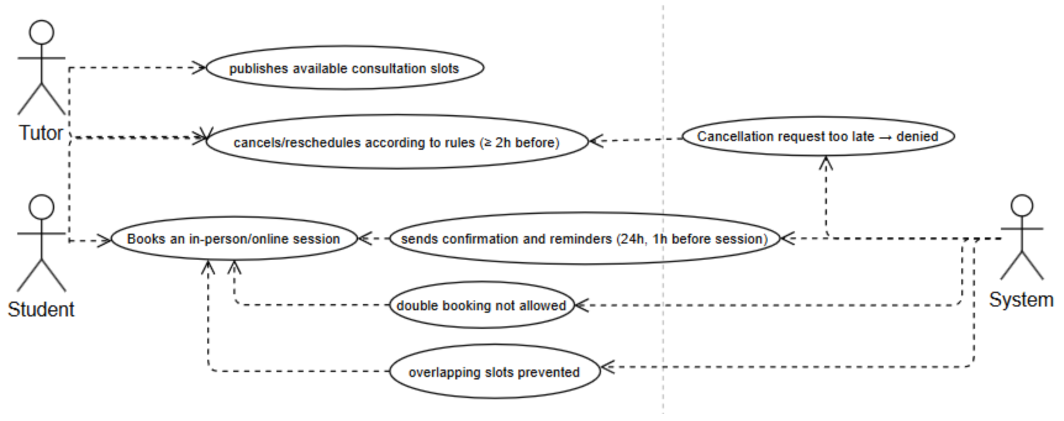
Tutor
Student
System
publishes available consultation slots
cancels/reschedules according to rules (≥ 2h before)
Cancellation request too late → denied
Books an in-person/online session
sends confirmation and reminders (24h, 1h before session)
double booking not allowed
overlapping slots prevented

Figure 4: Session Scheduling Management

Use-case ID	UC-03
Use-case name	Session Scheduling Management
Use-case overview	This use case describes the end-to-end process of session scheduling between a student and a tutor, including slot publication, booking, cancellation, rescheduling, and automatic notifications.
Actors	Student, Tutor, System
Preconditions	1. Student is matched with a tutor. 2. Tutor has published available slots. 3. System is operational and connected to the scheduling database.
Trigger	1. Tutor publishes or edits availability slots. 2. Student attempts to book a session. 3. Student or tutor submits a cancellation or rescheduling request.
Steps	1. Tutor publishes available consultation slots. 2. System prevents overlapping slots. 3. Student views available slots. 4. If a slot is suitable, student books a session. 5. System prevents double-booking and stores session details. 6. System confirms the booking and sends notifications. 7. Before the session, the system sends reminders (24h and 1h before). 8. Student or tutor may request cancellation or rescheduling at least 2 hours before the session. 9. System validates the request: if valid, cancels/reschedules and sends notifications; otherwise denies the request.
Postconditions	1. Session is successfully scheduled, rescheduled, or cancelled according to rules. 2. Notifications and reminders are delivered to all relevant parties.
Exception Flow	1. Overlapping slots → publishing is denied. 2. Double-booking attempt → booking is denied. 3. No suitable slot → student is prompted to wait or check later. 4. Late cancellation → request is denied.

Table 4: Use Case UC-03: Session Scheduling Management

6.5 UC-04 Feedback & Progress Tracking

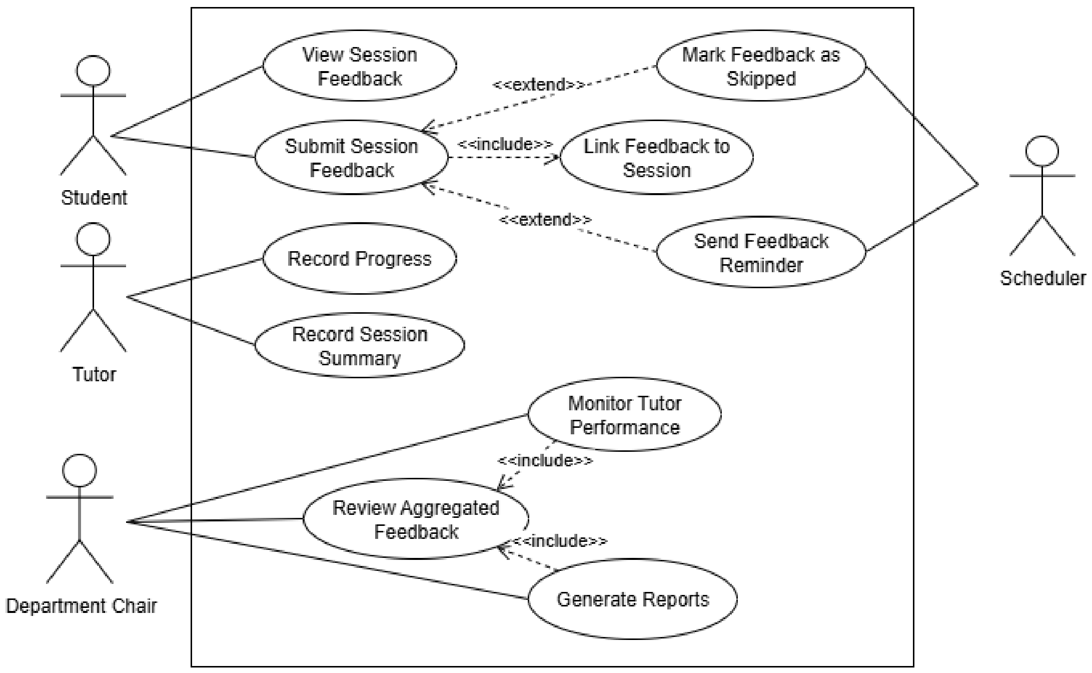

Figure 5: Submit Session Feedback

Use-case ID	UC-04
Use-case name	Submit Session Feedback
Use-case overview	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.
Actors	Student (primary), Scheduler (secondary, for reminders)
Preconditions	1. A tutoring session has been completed. 2. The system is running and accessible. 3. Student is authenticated in the system.
Trigger	The student clicks the "Submit Feedback" option after the session is completed.
Steps	1. System displays a feedback form linked to the completed session. 2. Student fills in and submits the structured feedback. 3. System validates the input and saves the feedback in the database. 4. System links the feedback to the corresponding session. 5. If no feedback is submitted within the allowed time, the Scheduler triggers reminders. 6. If the deadline passes without submission, the system marks the feedback as "Skipped."
Postconditions	1. Feedback is stored in the database and linked to the correct session. 2. If skipped, the system records a "Feedback Skipped" status for that session. 3. Data is available for tutors and department chairs in aggregated reports.
Alternative Flows	1. Multiple Session Feedback → If multiple sessions are pending, the system displays a list and allows the student to submit feedback sequentially. 2. Draft Save → The student may save incomplete feedback as a draft and return later within the allowed timeframe. 3. Feedback Revision → Within 24h, the student may edit and resubmit feedback; the updated version replaces the original record.
Exception Flow	1. If the student loses connection before submission, the system prompts the student to retry. 2. If the session record is missing or corrupted, the system logs an error and notifies the coordinator.

Table 5: Use Case UC-04: Submit Session Feedback

6.6 UC-05 Reporting & Analytics

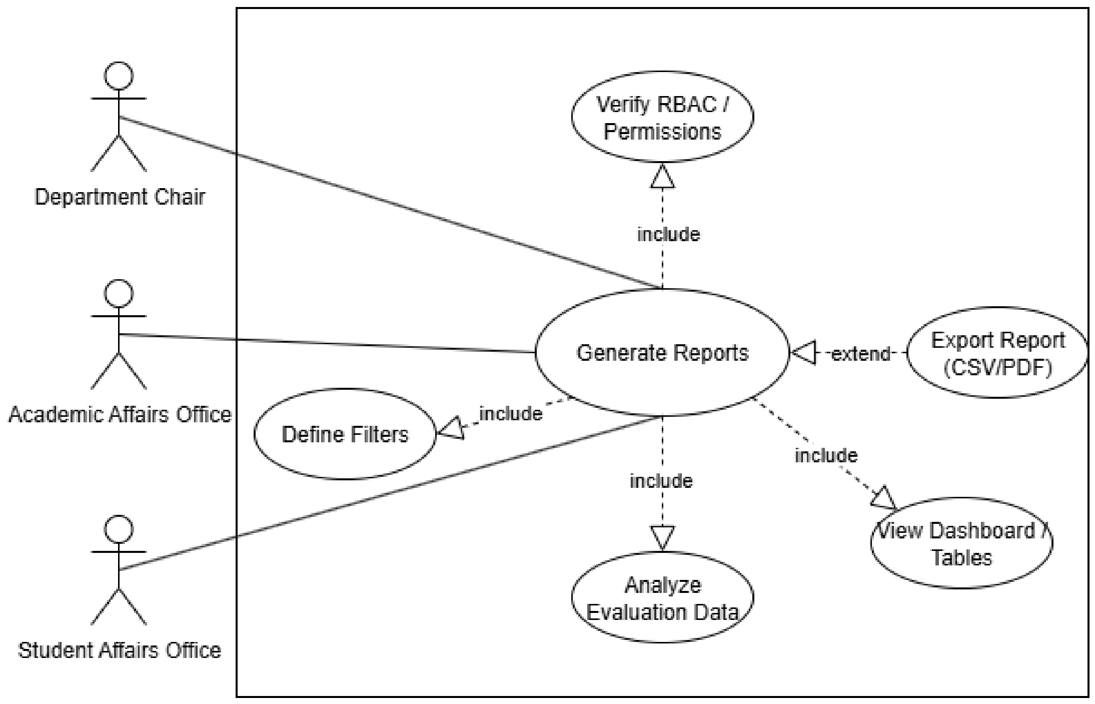

Figure 6: Reporting & Analytics

Use-case ID	UC-05
Use-case name	Reporting & Analytics
Use-case overview	To allow academic units to view, analyze, and export reports (attendance, performance, tutor workload, student demand, participation) for monitoring and decision-making.
Actors	Department Chair (primary), Academic Affairs Office (primary), Student Affairs Office (primary)
Preconditions	1. The user is authenticated via SSO and authorized to view reports. 2. Session, feedback, and progress data exist. 3. The system is operational.
Trigger	Users open the "Reporting" module.
Steps	1. Retrieve the Reporting page. 2. Select a report type (Departmental / Workload & Demand / Participation). 3. Set filters (term/date range, department/program, tutor, cohort). 4. Retrieve relevant data and analyze metrics (attendance, performance ratings, tutor workload, student demand, participation). 5. Display dashboards/tables with results and data-source notes. 6. Optional: export the report as CSV or PDF; add metadata (generation time, filters, version) and record the action in the audit log.
Postconditions	1. The report is displayed with analyzed metrics. 2. If exported, a CSV/PDF file is produced with metadata. 3. Access/export actions are logged.
Exception Flow	1. Access denied: insufficient permissions → show denial message and log the attempt. 2. Invalid/empty filters → show "No data" and allow the actor to adjust filters. 3. Export error: I/O or file-size issue → show an error and suggest retrying or narrowing scope.

Table 6: Use Case UC-05: Reporting & Analytics

6.7 UC-06 Integration with HCMUT Infrastructure

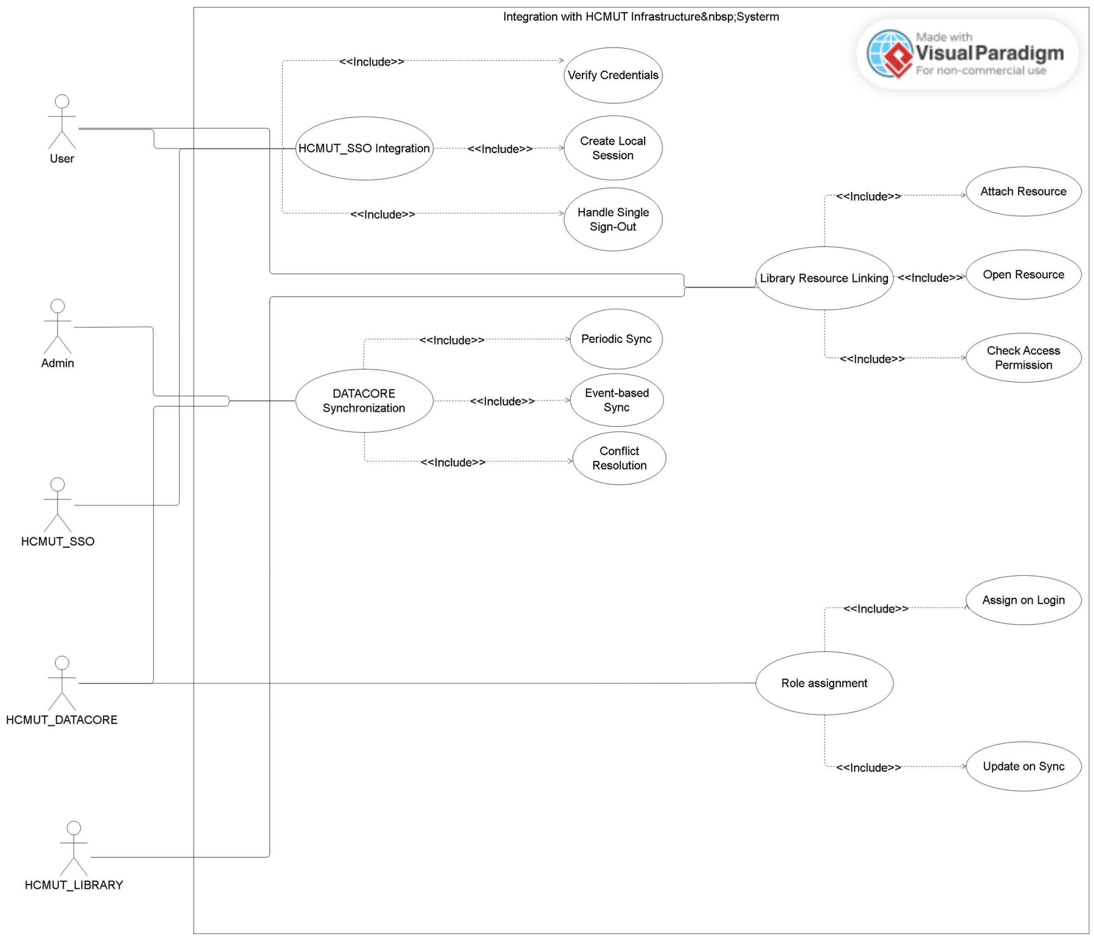

Figure 7: Integration with HCMUT Infrastructure

Use-case ID	UC-06.01
Use-case name	HCMUT_SSO Integration
Use-case overview	The system integrates with HCMUT_SSO for unified authentication, allowing users to log in using university credentials and automatically manage single sign-out.
Actors	User, System, HCMUT_SSO
Preconditions	HCMUT_SSO service is operational and reachable.
Trigger	User initiates login via HCMUT_SSO.
Steps	1. User selects "Login with HCMUT_SSO". 2. System redirects to the authentication portal. 3. HCMUT_SSO validates credentials and returns a token. 4. System verifies the token and creates a session. 5. Upon sign-out at SSO, the system terminates the local session.
Postconditions	1. User is successfully authenticated. 2. Single sign-out ensures session consistency.
Exception Flow	1. Invalid or expired token → login attempt rejected. 2. SSO service unavailable → system displays maintenance message.
Priority	Must

Table 7: Use Case UC-06.01: HCMUT_SSO Integration

Use-case ID	UC-06.02
Use-case name	DATACORE Synchronization
Use-case overview	The system synchronizes personal and academic data from HCMUT_DATACORE periodically or in near real-time to ensure data consistency and reduce manual entry.
Actors	System, HCMUT_DATACORE, Administrator
Preconditions	DATACORE APIs are online and accessible.
Trigger	Scheduled synchronization or data-change event detected.
Steps	1. System triggers synchronization with DATACORE. 2. Retrieve updated profiles and academic data. 3. Validate and compare data with local records. 4. Update local data using DATACORE as source of truth. 5. Log synchronization status and timestamp.
Postconditions	1. Local data mirrors DATACORE. 2. Synchronization events logged for auditing.
Exception Flow	1. Connection timeout or API error → retry with exponential backoff. 2. Invalid data format → skip record and log validation error.
Priority	Must

Table 8: Use Case UC-06.02: DATACORE Synchronization

Use-case ID	UC-06.03
Use-case name	Role Assignment
Use-case overview	The system automatically assigns user roles (student, tutor, coordinator, department chair, or administrator) based on centralized role data from DATACORE and SSO.
Actors	User, System, HCMUT_DATACORE
Preconditions	User is authenticated via SSO and DATACORE role data is available.
Trigger	User login or scheduled role update.
Steps	1. Retrieve role mapping from DATACORE. 2. Match user ID with centralized role data. 3. Assign permissions based on role. 4. Apply changes in access-control policies. 5. Log the role-assignment event.
Postconditions	1. User roles align with centralized data. 2. Updated permissions take effect immediately.
Exception Flow	1. Role data missing → assign default "student" role and notify admin. 2. Role conflict → logged and flagged for manual verification.
Priority	Must

Table 9: Use Case UC-06.03: Role Assignment

Use-case ID	UC-06.04
Use-case name	Library Resource Linking
Use-case overview	The system connects with HCMUT_LIBRARY to let tutors and students attach or access library materials securely within tutoring sessions or summaries.
Actors	Student, Tutor, System, HCMUT_LIBRARY
Preconditions	Library API and authentication services are available.
Trigger	User attaches or opens a library resource.
Steps	1. User searches or selects a library resource. 2. System requests metadata and access permissions. 3. Verify user eligibility via role-based access. 4. Attach or open the resource in session view. 5. Log the access event.
Postconditions	1. Resource successfully linked to the session or summary. 2. Access rights enforced per library policy.
Exception Flow	1. Access denied → system displays permission error. 2. Library service unavailable → prompt user to retry or queue attachment.
Priority	Could

Table 10: Use Case UC-06.04: Library Resource Linking

6.8 UC-07 Advanced / Optional Features

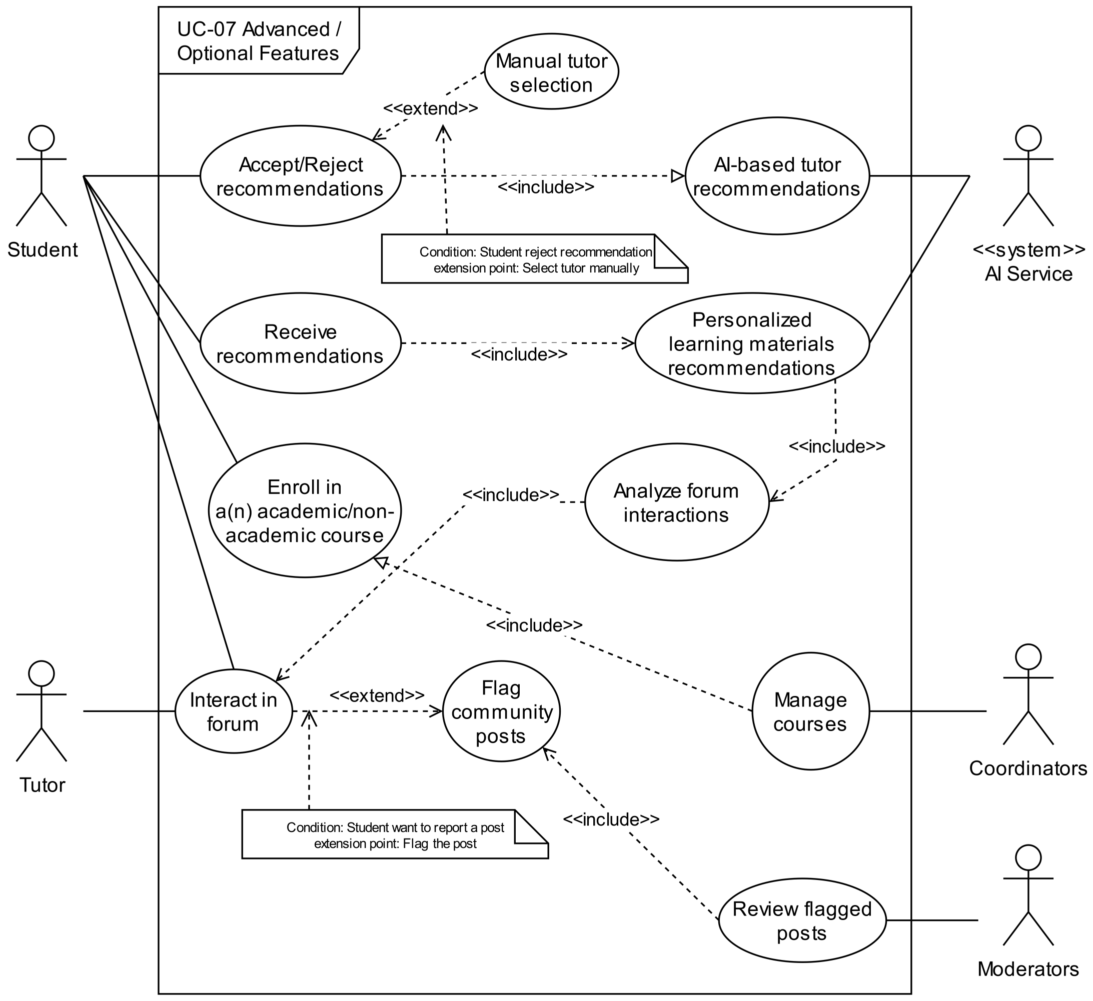
UC-07 Advanced / Optional Features
Manual tutor selection
<<extend>>
Accept/Reject recommendations
<<include>>
AI-based tutor recommendations
Student
<<system>> AI Service
Condition: Student reject recommendation extension point: Select tutor manually
Receive recommendations
<<include>>
Personalized learning materials recommendations
<<include>>
Enroll in a(n) academic/non-academic course
<<include>>
Analyze forum interactions
<<include>>
Tutor
Interact in forum
<<extend>>
Flag community posts
Manage courses
Coordinators
Condition: Student want to report a post extension point: Flag the post
<<include>>
Review flagged posts
Moderators

Figure 8: AI-generated review quizzes

Use-case ID	UC-07
Use-case name	AI-generated review quizzes
Use-case overview	Provide students with quizzes related to the courses they are enrolling.
Actors	Students, AI service
Preconditions	1. The system is running. 2. Internet connection is available. 3. AI service must be available. 4. The courses must have learning materials uploaded by tutors.
Trigger	Students use the AI-based quiz generation function.
Steps	1. Retrieve uploaded course materials of the course. 2. Process and analyze the content with AI. 3. Search the internet for related academic resources and quizzes. 4. Generate quiz questions covering the key topics. 5. Display the quiz to the requested students.
Postconditions	1. The quizzes are displayed on the screen of the requested students, and they can download the quizzes as a document file. 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
Alternative Flows	- 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.
Exception Flow	1. If AI service isn't available, display an error.

Table 11: Use Case UC-07: AI-generated review quizzes

7 System Integration

- **HCMUT_SSO**: OAuth2/OIDC-based single sign-on.
- **HCMUT_DATACORE**: Read-only sync of personal data (name, ID, faculty, email, status).
- **HCMUT_LIBRARY**: Link and share official materials within sessions.

8 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.)

9 Submission and Deliverables

Deliverables:

- PDF generated from this LaTeX 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.

10 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.

11 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).