# IV. Use-Case Diagram

# 1. Overview diagramA diagram of a network AI-generated content may be incorrect.

# 1.1. List of Actors

|  |  |  |
| --- | --- | --- |
| No. | Use-case | Description |
| 1 | Admin | The system administrator, manages accounts and general configurations. |
| 2 | AcademicOfficer | The academic officer, manages the curriculum, courses, and schedules. |
| 3 | Lecturer | The instructor/teacher, manages their assigned classes, enters grades, and uploads materials. |
| 4 | Student | The student, users who enroll in courses, view grades, and access study materials. |

# 1.2. List of Use Cases

|  |  |  |
| --- | --- | --- |
| No. | Use-case | Description |
| 1 | Login | Allows users (Admin, Academic Officer, Lecturer, Student) to log into the system using their credentials. |
| 2 | Logout | Enables users to securely log out of the system. |
| 3 | Change Password | Allows users to change their password for security reasons. |
| 4 | View Profile | Displays the user’s personal information such as name, ID, email, and role. |
| 5 | Manage Users | Manage user accounts in the system (view, add, edit, delete). |
| 6 | Create Account | Create new user accounts with default credentials. |
| 7 | Manage Permissions | Define and control access rights for each user role |
| 8 | Manage Fee | Manage tuition fee structures, payment records, and related settings. |
| 9 | Assign Teaching | Assign lecturers to specific courses or class sections. |
| 10 | Manage Courses | Maintain and update course information such as course code, name, credits, and assigned lecturer. |
| 11 | Manage Schedule | Create and manage class timetables and academic schedules. |
| 12 | Manage Students | Maintain student records including academic status and personal information |
| 13 | Manage Lecturers | Manage lecturer profiles and teaching assignments. |
| 14 | Manage Registrations | Oversee student course registration activities. |
| 15 | Process Academic Requests | Review, approve, or reject student academic requests (e.g., exam deferral, grade appeal, verification letters). |
| 16 | Manage Assigned Classes | View and manage the list of classes assigned to the lecturer. |
| 17 | Enter Grades | Input, update, and submit student grades for each class section. |
| 18 | Create Exam | Create or manage exam details and schedules for assigned courses. |
| 19 | View Schedule | View personal teaching schedule and class timetable. |
| 20 | Register Courses | Register for courses offered in the current semester. |
| 21 | View Schedule | View the class schedule for registered courses. |
| 22 | View Grades | View grades and academic results for completed courses. |
| 23 | Pay Fee | Pay tuition fees for registered courses online. |

2. Use Case: Login  
  
**2.1. Summary**  
This use case describes how a user (Admin, Academic Staff, Lecturer, Student) logs into the Academic Management System.  
**2.2. Flow of Events**None.  
**2.2.1. Main Flow**  
This use case begins when a user wants to log into the system.  
The system prompts the user to enter their Username (or ID) and Password.  
The user enters their Username, Password, and requests to log in.  
The system validates the entered information. If correct, the system authenticates the user's role (Admin, Lecturer, etc.) and grants the user access to the system with their corresponding permissions.  
**2.2.2. Alternative Flows**None.  
**2.2.2.1. Invalid Username/Password**  
If, in the Main Flow, the user enters an incorrect username or password, the system displays an error message. The user can choose to return to the beginning of the Main Flow or cancel the login, at which point the use case ends.  
**2.3. Special Requirements**  
None.  
**2.4. Pre-conditions**  
The user is not authenticated with the system.  
**2.5. Post-conditions**  
If the use case is successful, the user is logged into the system with their authenticated role. Otherwise, the system state remains unchanged.  
**2.6. Extension Points**  
Change Password Use Case: The user can navigate to the change password functionality.  
3. Use Case: Manage Users  
**3.1. Summary**  
This use case allows an Administrator (Admin) to manage (view, add, edit, delete) user account information in the system (including Students, Lecturers, Academic Staff).  
**3.2. Flow of Events**None.  
**3.2.1. Main Flo**  
This use case begins when the Admin selects the user management function.  
The system displays a list of users.  
If the Admin selects "Add", the Add User Account (CreateAccount) sub-flow is executed.  
If the Admin selects the "Update" function for a user, the Update User Account sub-flow is executed.  
If the Admin selects the "Delete" function for a user, the Delete User Account sub-flow is executed.  
**3.2.1.1. Add User Account (CreateAccount)**  
The system prompts the Admin to enter the user's information (e.g., Full Name, email, role, default ID, default password).  
The Admin enters the information and requests to add the user.  
The system validates the information (e.g., unique email/ID) and adds the account to the system.  
The system notifies the Admin of the result.  
**3.2.1.2. Update User Account**  
The system displays the information of the user to be updated.  
The Admin proceeds to edit the information (e.g., update role, reset password).  
The Admin requests to update the information.  
The system validates the information and applies the updates.  
The system notifies the Admin of the result.  
**3.2.1.3. Delete User Account**  
The system prompts the Admin to confirm the deletion.  
If the Admin confirms, the system deletes the user (or deactivates the account) from the system.  
The system notifies the Admin of the result.  
**3.3. Special Requirements**  
The user list must be paginated.  
New or updated passwords must be encrypted.  
**3.4. Pre-conditions**  
The Administrator (Admin) is logged into the system. **3.5. Post-conditions**  
If the use case is successful, user information is added, updated, or deleted from the system. Otherwise, the system state remains unchanged.  
**3.6. Extension Points**  
Manage Permissions Use Case: The Admin can navigate to the function for managing detailed role-based permissions.  
4. Use Case: Register Courses  
**4.1. Summary**  
This use case allows a Student to view the list of courses offered for the semester and register for their desired courses.  
**4.2. Flow of Events**None.  
**4.2.1. Main Flow**  
This use case begins when the Student selects the "Course Registration" function within the allowed registration period.  
The system displays the courses available for registration in the current semester (course name, credits, schedule, lecturer).  
The Student selects (adds) desired courses/course sections to a "Temporary Registration Cart".  
If the Student selects a course already in the cart, the system issues a notification.  
If the Student chooses to remove a course from the temporary cart, the system updates the cart.  
When the Student selects "Confirm Registration": a. The system checks for constraints: prerequisites, schedule conflicts, min/max credit load, class enrollment limits (class size). b. If valid, the system records the student's registration and updates the class enrollment count. c. The system displays a success message and the estimated tuition fee.  
**4.2.2. Alternative Flows**None.  
**4.2.2.1. Registration Constraint Violation**  
If the system detects a violation (e.g., schedule conflict, class full, missing prerequisite), the system will display a detailed error message for each violating course. The Student can return to adjust their registration cart.  
**4.3. Special Requirements**  
The system must ensure data integrity when multiple students register for the same course section concurrently (handle enrollment conflicts).

**4.4. Pre-conditions**  
The Student is logged in.  
The system is within the official course registration period.  
**4.5. Post-conditions**  
If the use case is successful, the student's course registration is recorded. Otherwise, the system state remains unchanged.  
**4.6. Extension Points**  
View Schedule Use Case: From this use case, the student can view their prospective schedule before confirming.  
Pay Fee Use Case: After registration, the student can proceed to the tuition payment function.  
5. Use Case: Enter Grades  
**5.1. Summary**  
This use case allows a Lecturer to enter and update grades for students in their assigned course sections.

**5.2. Flow of Events**

None.  
**5.2.1. Main Flow**  
This use case begins when the Lecturer selects the "Enter Grades" function.  
The system displays a list of course sections assigned to the Lecturer.  
The Lecturer selects a course section.  
The system displays the student roster for that class, along with grade columns (e.g., Attendance, Midterm, Final Exam).  
The Lecturer enters or edits the grades for the students.  
The Lecturer selects "Save Draft" to save changes without submitting, or "Finalize and Submit Grades" to lock the grade sheet.  
The system validates the grades (e.g., on a 10-point scale, no empty fields if submitting).  
If valid, the system saves the grades. If "Finalize" was selected, the system locks the grade sheet and forwards it to Academic Staff.  
The system notifies the Lecturer of the result.  
**5.2.2. Alternative Flows**None.  
**5.2.2.1. Invalid Grade Data**  
If the Lecturer enters invalid grades (e.g., "A", 11, or leaves a field empty upon submission), the system will report an error and highlight the invalid cells, requiring the Lecturer to correct them.  
**5.3. Special Requirements**  
The system must support grade entry via Excel file import.  
After "Finalize and Submit Grades", the Lecturer can no longer modify the grades.  
**5.4. Pre-conditions**  
The Lecturer is logged in.  
It is within the institution's allowed period for grade entry.

**5.5. Post-conditions**  
If the use case is successful, student grades are updated (as a draft or finalized). Otherwise, the system state remains unchanged.  
**5.6. Extension Point**  
None.  
6. Use Case: Process Academic Requests  
**6.1. Summary**  
This use case allows Academic Staff to view, approve, or reject online requests/petitions from students (e.g., exam deferral request, grade appeal, student verification letter).  
**6.2. Flow of Events**None.  
**6.2.1. Main Flow**  
This use case begins when the Academic Staff selects the "Process Requests" function.  
The system displays a list of requests currently in "Pending" status, filterable by request type.  
The Academic Staff selects a request to view its details.  
The system displays detailed information: Student information, request content, and any attached files (if applicable).  
The Academic Staff reviews the information and selects an action: "Approve" or "Reject".  
If "Approve" is selected, the system updates the request status to "Approved" and executes related tasks (e.g., adding the student to the exam deferral list).  
If "Reject" is selected, the system requires the Staff to enter a reason for rejection.  
The system updates the request status and sends a notification of the outcome (including the reason, if rejected) to the student.  
**6.3. Special Requirements**  
None.  
**6.4. Pre-conditions**  
The Academic Staff is logged in.  
At least one student request exists in the "Pending" state.  
**6.5. Post-conditions**  
If the use case is successful, the student's request is updated to a new status ("Approved" or "Rejected"). Otherwise, the system state remains unchanged.  
**6.6. Extension Points**  
None.  
**V. Class Diagram**

A diagram of a computer

AI-generated content may be incorrect.

**VI. Database Diagram**