**University Management System Use Cases**

**1. Course Management Use Cases**

**Create a New Course**

* **Primary Actor**: Administrator
* **Description**: The administrator can create new courses by specifying details such as the course name, code, department, professor, and schedule.
* **Attributes**:
  + *CourseName, CourseCode, DepartmentID, Schedule, ProfessorID*
  + *(Tables Involved: Course, CourseAssignToProfessor, ClassSchedule, Professor)*
* **Preconditions**:
  + The administrator must be logged into the system with appropriate privileges.
* **Postconditions**:
  + A new course is created and added to the system database.

**Assign Course to Professors and Teaching Assistants**

* **Primary Actor**: Administrator
* **Description**: The administrator can assign courses to professors and teaching assistants for a specific semester.
* **Attributes**:
  + *CourseID, ProfessorID, TA\_ID, Semester, Year*
  + *(Tables Involved: CourseAssignToProfessor, Course, Professor, TeachingAssistant)*
* **Preconditions**:
  + The course must exist in the database.
  + Professors and teaching assistants must be registered users.
* **Postconditions**:
  + The assignment is saved to the system, and relevant users are notified.

**2. Student Use Cases**

**Enroll in a Course**

* **Primary Actor**: Student
* **Description**: Students can enroll in available courses for the current semester.
* **Attributes**:
  + *StudentID, CourseID, EnrollmentDate, SemesterID*
  + *(Tables Involved: EnrollStudentInCourse, Student, Course, Semester)*
* **Preconditions**:
  + The student must be logged in.
  + The student must meet the prerequisites for the course.
* **Postconditions**:
  + The student is enrolled, and a confirmation message is sent.

**View Current Courses**

* **Primary Actor**: Student
* **Description**: Students can view a list of courses they are currently enrolled in, along with details like the course name, professor, and schedule.
* **Attributes**:
  + *StudentID, CourseID, CourseName, ProfessorName, ClassSchedule*
  + *(Tables Involved: EnrollStudentInCourse, Course, CourseAssignToProfessor, ClassSchedule)*
* **Preconditions**:
  + The student must be logged into the system.
* **Postconditions**:
  + The system displays the student’s enrolled courses with details.

**View Grades and Transcripts**

* **Primary Actor**: Student
* **Description**: Students can view their grades for completed courses and their academic transcript.
* **Attributes**:
  + *StudentID, CourseID, Grade, Transcript, CourseName*
  + *(Tables Involved: StudentResult, Student, Course)*
* **Preconditions**:
  + The student must have completed at least one course.
* **Postconditions**:
  + The system displays the grades and transcript.

**3. Professor Use Cases**

**Create/Update Course Materials**

* **Primary Actor**: Professor
* **Description**: Professors can upload and manage course materials such as lectures, assignments, and reading materials.
* **Attributes**:
  + *CourseID, MaterialID, MaterialType, UploadDate, MaterialURL*
  + *(Tables Involved: CourseMaterial, Course)*
* **Preconditions**:
  + The professor must have access to the course.
* **Postconditions**:
  + New course materials are uploaded and made available to students.

**Assign and Grade Exams**

* **Primary Actor**: Professor
* **Description**: Professors can assign exams, grade them, and provide feedback.
* **Attributes**:
  + *StudentID, CourseID, ExamID, Grade, Feedback*
  + *(Tables Involved: StudentResult, Exam)*
* **Preconditions**:
  + The professor must have created an exam for the course.
* **Postconditions**:
  + Grades and feedback are saved for each student.

**View Student Performance Reports**

* **Primary Actor**: Professor
* **Description**: Professors can generate detailed performance reports for students.
* **Attributes**:
  + *StudentID, CourseID, Grade, Attendance, PerformanceReport*
  + *(Tables Involved: StudentResult, Attendance)*
* **Preconditions**:
  + The professor must have access to course and student data.
* **Postconditions**:
  + A performance report is generated.

**4. Teaching Assistant Use Cases**

**Grade Assignments**

* **Primary Actor**: Teaching Assistant
* **Description**: Teaching assistants can grade assignments and provide feedback.
* **Attributes**:
  + *AssignmentID, StudentID, Grade, Feedback*
  + *(Tables Involved: Assignments, StudentResult)*
* **Preconditions**:
  + Assignments must be submitted by students.
* **Postconditions**:
  + The grades and feedback are recorded.

**Moderate Discussion Forums**

* **Primary Actor**: Teaching Assistant
* **Description**: Teaching assistants moderate online discussion forums for courses.
* **Attributes**:
  + *ForumID, PostID, Message, StudentID, Timestamp*
  + *(Tables Involved: Forum, ForumPosts)*
* **Preconditions**:
  + The forum must be active.
* **Postconditions**:
  + Posts are moderated, and appropriate responses are provided.

**5. Administrator Use Cases**

**Add/Remove Users (Students, Professors, TAs)**

* **Primary Actor**: Administrator
* **Description**: Administrators can add or remove users from the system.
* **Attributes**:
  + *UserID, Role, Name, Email, Status*
  + *(Tables Involved: User, Student, Professor, TeachingAssistant)*
* **Preconditions**:
  + The administrator must have privileges.
* **Postconditions**:
  + Users are added or deactivated in the system.

**Generate Reports on Enrollment and Attendance**

* **Primary Actor**: Administrator
* **Description**: Administrators can generate comprehensive reports on enrollment and attendance.
* **Attributes**:
  + *CourseID, StudentID, EnrollmentDate, AttendanceStatus, Grade*
  + *(Tables Involved: EnrollStudentInCourse, Attendance, StudentResult)*
* **Preconditions**:
  + The administrator must have access to relevant data.
* **Postconditions**:
  + Reports are generated for analysis.

**6. Library Management Use Cases**

**Check Out Library Books**

* **Primary Actor**: Student, Professor
* **Description**: Users can check out library books for a specified period.
* **Attributes**:
  + *BookID, StudentID, CheckOutDate, ReturnDate*
  + *(Tables Involved: Library, BookLoans)*
* **Preconditions**:
  + The user must have a valid account.
* **Postconditions**:
  + The book is checked out, and the due date is recorded.

**View Loaned Books and Due Dates**

* **Primary Actor**: Student, Professor
* **Description**: Users can view a list of all books they have checked out, along with due dates.
* **Attributes**:
  + *BookID, StudentID, CheckOutDate, DueDate, Title*
  + *(Tables Involved: Library, BookLoans)*
* **Preconditions**:
  + The user must have books checked out.
* **Postconditions**:
  + A list of loaned books is displayed.

**7. Finance and Fees Use Cases**

**Pay Tuition Fees**

* **Primary Actor**: Student
* **Description**: Students can pay their tuition fees using an integrated payment system.
* **Attributes**:
  + *StudentID, FeeAmount, PaymentStatus, PaymentDate*
  + *(Tables Involved: Finance, Fees)*
* **Preconditions**:
  + The student must have outstanding fees.
* **Postconditions**:
  + The payment is recorded, and the balance is updated.

**Generate Fee Payment Receipts**

* **Primary Actor**: Administrator
* **Description**: Administrators can generate fee payment receipts for students.
* **Attributes**: *StudentID, PaymentAmount, PaymentDate, ReceiptID (from Finance)*

***Preconditions****: A fee payment must be processed.*

***Postcondition****s: A receipt is generated and sent to the student.*