Yoobee College MSE800 2507 Section A Database Design

1. Project Scope

This database concept for Yoobee College MSE800 2507 Section A is designed to create a repository for the records keeping and management of all student information attending the MSE-July 2025 intake Section A. The system manages basic academic information including student info, lecturer details, course subjects, student assessments, and student status whether the student passed or failed the subject.

The project scope needs to design and implement a relational database, create user interfaces that should be easy to use and understand for data input, data maintenance, and reports generation.

2. Entities and EER Diagram

STUDENT (Stores basic information about students)

- Attributes:

- Stud\_ID

- Name

- Email

- Enrol\_Date

- Subj\_ID

LECTURER (Stores information about lecturers)

- Attributes:

- Lect\_ID

-Name

- Email

- Department

SUBJECT (Stores information about subjects included in the MSE program)

- Attributes:

- Subj\_ID

- Subj\_Name

- SubjectCode

- CreditHours

- Lect\_ID

ASSESSMENT (Stores information about activities, assessments, and projects)

- Key Attributes:

- Assess\_ID

- AssessmentName

- AssessmentType (Activity, Assessment, Project)

- MaxGrade

- DueDate

- Subj\_ID

- Stud\_ID

- Stud\_Grade

- Status (Pass/Fail)

Entity Relationships:

* Lecturer to Subject: One-to-Many (Each lecturer can teach multiple subjects, but each subject has one primary lecturer)
* Subject to Student: One-to-Many (Each subject can have multiple students, but we track one current subject per student)
* Subject to Assessment: One-to-Many (Each subject can have multiple assessments)
* Student to Assessment: One-to-Many (Each student can have multiple assessments)

3. Table Design

This program’s database requires 4 tables:

Tables:

1. STUDENT

- Maintains student information

- Connects to their current subject

2. LECTURER

- Stores teacher information

3. SUBJECT

- Stores subject information

- References to the lecturer who teaches it

4. ASSESSMENT

- Stores assessment details and grades

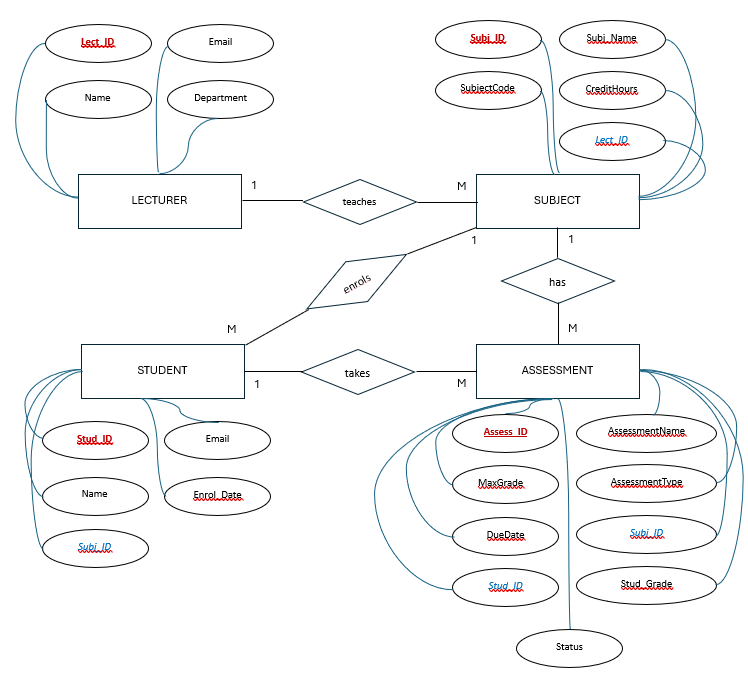
- Links to both the subject and the student

Table Structure Summary:

Table Name - Primary Key - Foreign Keys – Purpose

|  |  |  |  |
| --- | --- | --- | --- |
| **Table Name** | **Primary Key** | **Foreign Keys** | **Purpose** |
| Student | Stud\_ID | Subj\_ID | Keep student data |
| Lecturer | Lect\_ID | - | Store lecturer data |
| Subject | Subj\_ID | Lect\_ID | Subject data and corresponding lecturer |
| Assessment | Assess\_ID | Subj\_ID, Stud\_ID | Assessment and grades for each student |

4. ER Design



**Week 4 - Activity 1:**

**Step 1:** Write the number of actors and use cases for your college project, defining the scope of the project in the same way as the activity completed in Week 3 for your college.

Share your GitHub Link with your scenario from Week 3

Step 1: Write the number of actors and use cases for your college project, defining the scope of the project in the same way as the activity completed in Week 3 for your college.

Share your GitHub Link with your scenario from Week 3.

Actor 1 – Admin

* Enter Student details
* Enter Lecturer details
* Enter Subject details
* Enter Course details

Actor 2 – Student

* Enroll/Drop Course
* Select Subject
* Submits Assessment
* Checks Assessment Grade
* Checks Assessment Status

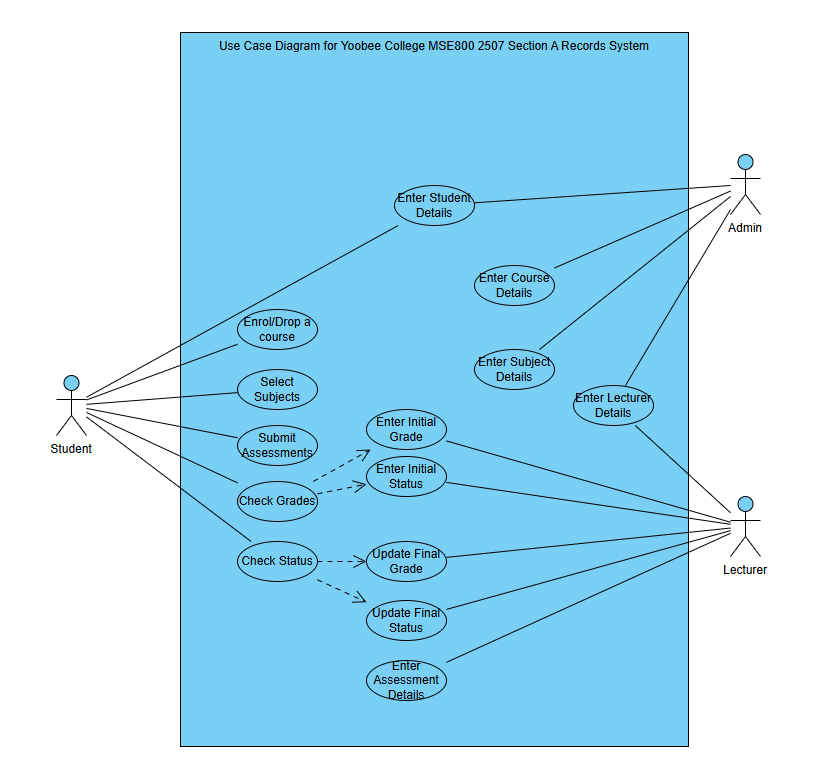
Actor 3 – Lecturer

* Teach a subject
* Enter Student Assessment details
* Enter Initial Student Assessment Grade
* Enter Initial Student Assessment Status
* Update Final Student Assessment Grade
* Update Final Student Assessment Status

**Week 4 - Activity 1:**

**Step 2:** Design the use-case diagram and share the GitHub link, including a brief scenario/description of your project.

The use case diagram below illustrates how each actor of the project interacts with the system and maps what each actor can accomplish when they interact with the system. The diagram also shows the links between each use case and how multiple actors can have different roles for each use case.



**Week 4 - Activity 1:**

**Step 3:** Design a **complete use-case diagram**, showing the relationships between all use cases. Share your GitHub link with the final diagram, due date: by today 11:59 PM.

The use case diagram below illustrates the first iteration of the complete use case diagram for the Yoobee College MSE800 2507 Section A Records Management System. It identifies how each actor of the project, namely Student, Lecturer, and Admin, interacts with the system and maps what each actor can accomplish when they interact with the system. The diagram also shows the links between each use case and how multiple actors can have different roles for each use case.

The diagram further illustrates how dependencies between use cases flow whether they are <<include>>, meaning the relationship represents a mandatory functionality from one use case to another, or <<extend>> which represents an optional extension of the functionality of one use case to another.

