



---

SECD2523-06 – DataBase

---

## STUDENT MANAGEMENT SYSTEM

### PHASE 2

### CONCEPTUAL DESIGN

---

SECTION : 06

LECTURER'S NAME : Dr. Layla Hasan

DATE : 7.12.2024

No.	Name	Metric No.
1	Moaz Adil Abdugadir Jalal	A23CS3025
2	EYAD AIMEN ELSHEIKH KHALIL	A23CS3024
3	Abdalla Ali Abdalla Ali	A23CS3022
4	Ali Isameldin Ali Abdelrhman	A23CS3001
5	Othman Hassan Othman Ali	A23CS3026

## **Table of Contents**

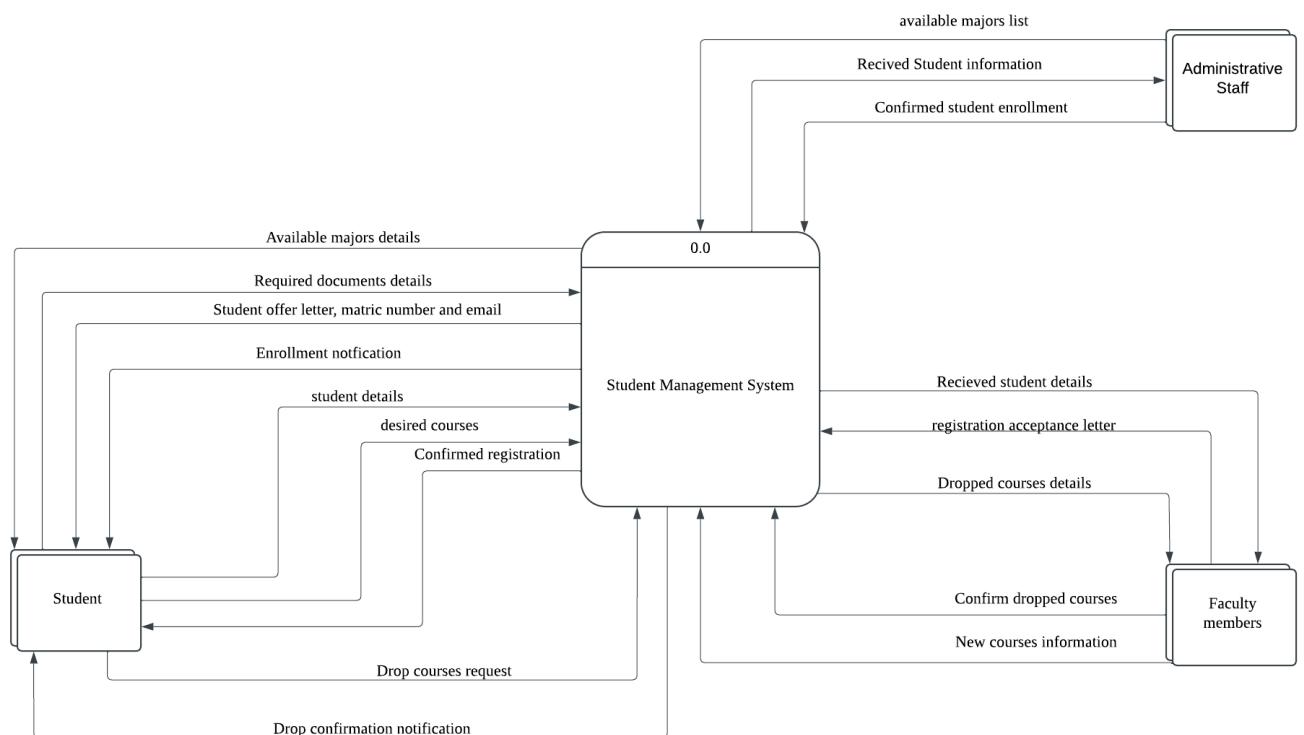
No.	Content	Page
1	<a href="#"><u>Introduction</u></a>	3
2	<a href="#"><u>Data Flow Diagram</u></a> <ul style="list-style-type: none"><li>● DFD TO BE</li></ul>	3
3	<a href="#"><u>Data &amp; Transaction Requirement</u></a> <ul style="list-style-type: none"><li>● Proposed Business rule</li><li>● Proposed Data &amp; Transactional</li></ul>	4
4	<a href="#"><u>Database Conceptual Design</u></a> <ul style="list-style-type: none"><li>● Conceptual ERD</li><li>● Enhanced ERD</li></ul>	7
5	<a href="#"><u>Data Dictionary</u></a>	9
6	<a href="#"><u>Summary</u></a>	13

# 1. Introduction

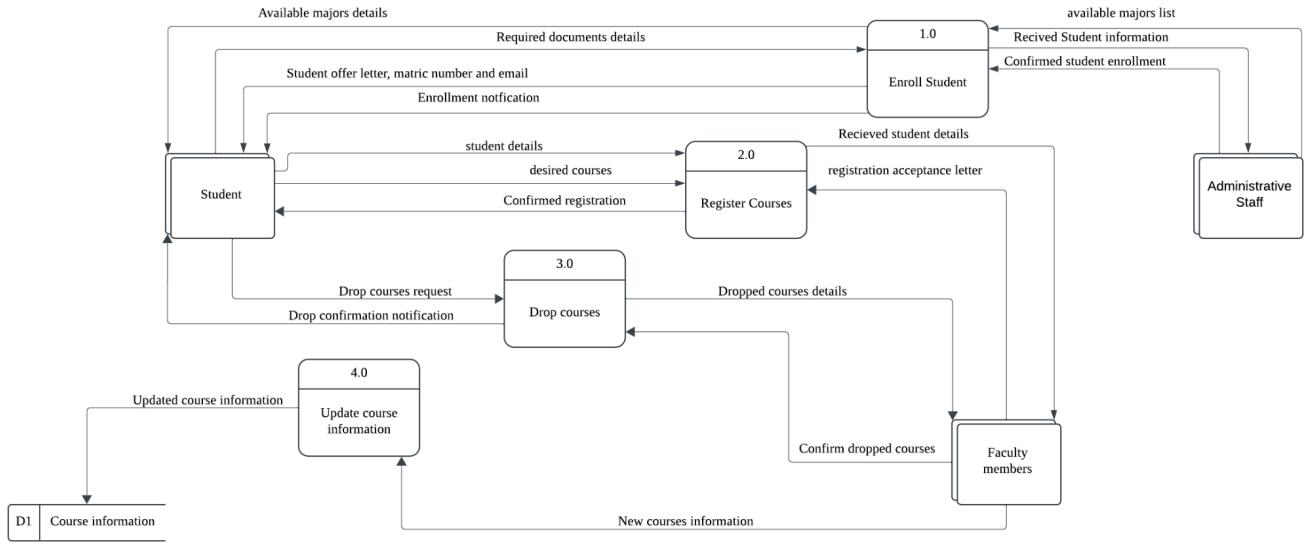
Recently there are a lot of students enrolling in different universities every year, because of their high ranking and reputation. And from that the students expect an efficient and high level of management. The university system may use an inefficient database system or a file-based approach which will make it difficult to maintain the growing volume of students' information recorded. And this requires a new system that uses the database approach to maintain the student's information efficiently and effectively. An effective student management system is the base of providing quality services for the different students and makes it easier for the different faculties to maintain and manipulate this information. From that the development of a new student management system will be effective, it will enhance the overall processing and retrieval of information, reduce data redundancy and the overall student registration experience.

## 2. Data Flow Diagram

### 2.1 Context Diagram

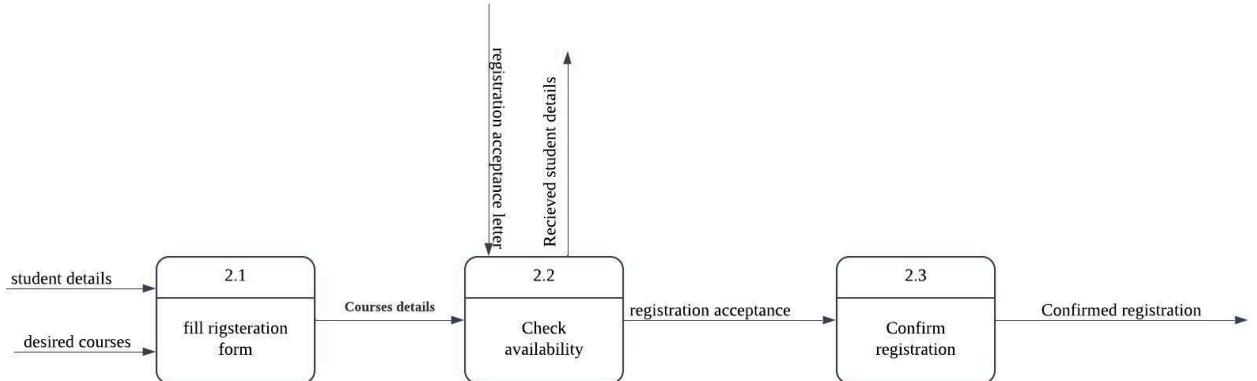


## 2.2 Parent Diagram (Level 0)



## 2.3 Child Diagram (Level 1)

### 2.3.1 Process 2.0 <Register Courses>



## 3. Data & Transaction Requirement

### 3.1 Proposed business rule:

#### Student:

1. Enrolls at the university.
2. View the courses.
3. Drop courses.
4. Pay tuition fees.

**Staff at faculty:**

1. Registers students at the specified faculty.
2. Manage students' issues regarding courses.
3. Update courses and their information.

**3.2 Proposed data & transactional:****3.2.1 Proposed data requirement:****Student:**

This table will include students' matric number as a primary key, students name, address, list of courses that contains the course codes of the registered courses by student, degree, faculty number as a foreign key to link to the Faculty table and transaction ID as a foreign key to link to Transaction table.

The student can enroll at one university by one administrative staff, pay one or many tuition fees for each semester, be registered by one faculty and enroll to one or many new courses and manage one or more current courses by one faculty member(drop courses requests).

**Faculty:**

This table will include faculty number as a primary key, a multivalued matric number containing the registered students matric numbers in the faculty and it is used as a foreign key and a multivalued staff ID as a foreign key containing the IDs of the faculty members to link to faculty member table.

The faculty has one or more faculty members, and administered by only one university

**Faculty member:**

This table will include staff ID as a primary key, name and faculty number as a foreign key to link to Faculty table.

The faculty member can manage one or more student information with the one or more student registered courses activities (enroll and drop), and can update zero or many courses' information.

**University:**

This table has information about the university. It contains university name as a primary key, university address, and multivalued faculty numbers inside it and it is used as a foreign key to link the faculty table.

The students can be enrolled at one university by one administrative staff, the university administers one or more faculties and the university one or many administrative staff.

**Administrative staff:**

This table is for the administrative staff in the university, it contains staff name, ID as primary key, and an email.

The administrative staff can process one or more students' enrollment requests at one university and belong to one university.

**Course:**

This table contains course code used as a primary key, faculty number used as a foreign key to faculty table, course name, section number and credit hours.

The course is enrolled by zero or many Students, it can be viewed and deleted (managed) by one or more from their registered courses along with one faculty member to approve the students requests, and updated by one faculty member.

**Transactions:**

This table contains the transaction id as a primary key, amountPaid, bills, and matric number as a foreign key to link to the Student table.

This table is accessed by one student which can transfer the tuition fees.

### **3.2.2 Proposed transactional requirement:**

**Data entry:**

1. Students enter the required information for enrollment, such as name, passport number, phone number, email, academic certificate, and faculty that the student wishes to enroll for.
2. Faculty members enter their identification information, such as name or identification number and password in order to see the various student requests and approve it / deny it and add course information such as, course name, course code and credit hours that the students can register to.
3. Students select the courses from a list of courses they can register to submit the registration request.
4. Students enter their identification information, such as name or matric number and password to access their academic records and submit the various requests.

**Data update/delete:**

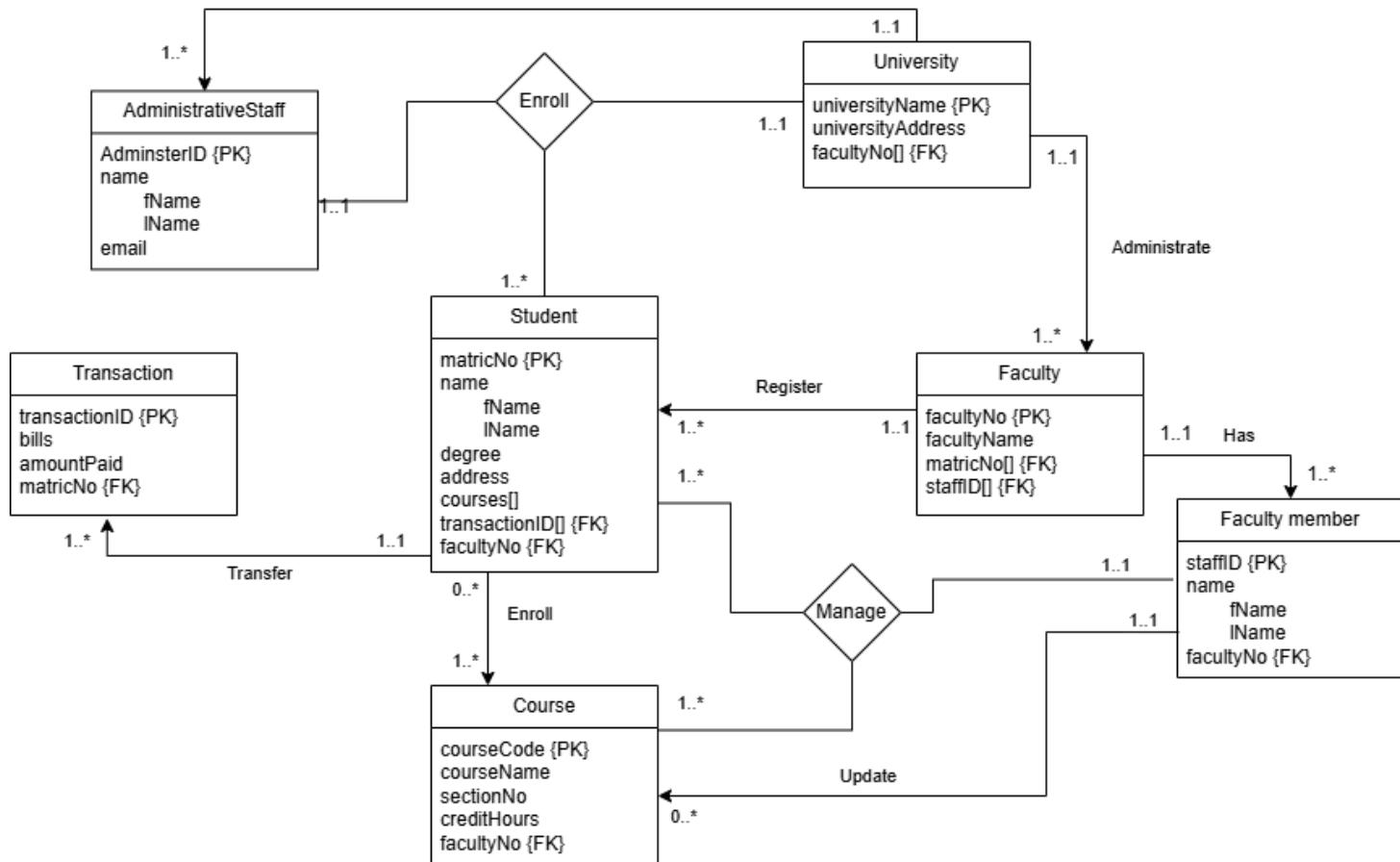
1. Faculty members can update/delete student records.
2. Faculty members can update/delete courses based on students' requests.
3. The system will Update/delete the available courses.

**Data queries:**

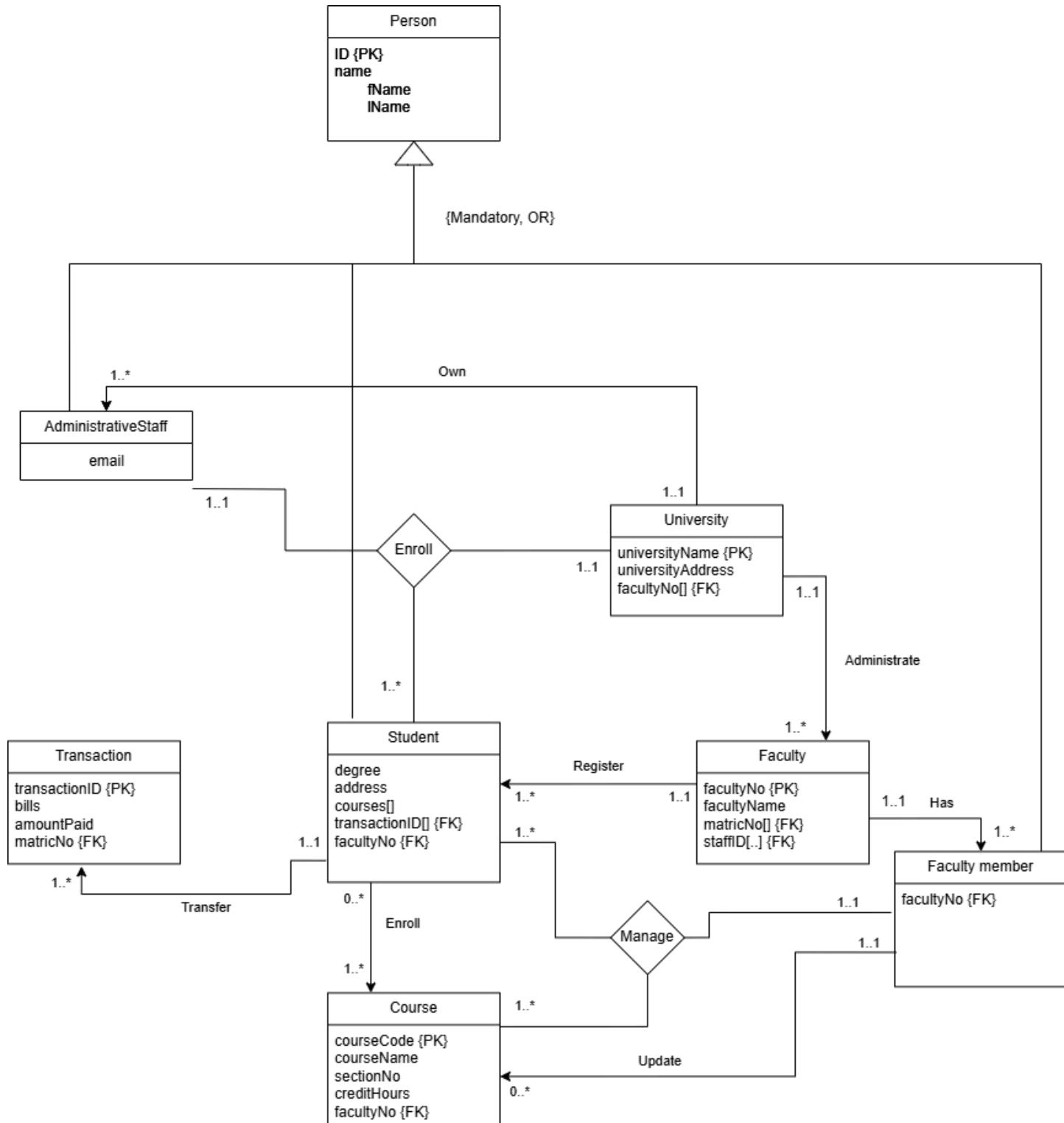
1. Display students' information.
2. Display student registered courses.
3. Display students' faculty enrollment requests.
4. Display student academic records.
5. Display students' course registration requests.
6. Display students' amendment requests.
7. Display students drop course requests.
8. Display course information.

## 4. Database Conceptual Design

### 4.1 Conceptual ERD



## 4.2 Enhanced ERD (EERD)



## 5. Data Dictionary

### 5.1 Description of Entity

entity	description	occurrence
student	Student information	Students can enroll in university, make transfers based on the university fees, and get registered in a faculty while being managed by faculty members along with student courses.
faculty	Faculty's information	Faculty has faculty members, registers students in a faculty and gets administered by the university.
course	Course's information	Course is enrolled by the student, it also gets updated and managed along with the student by the faculty members.
university	University's information	a University has administrative staff who get enrolled along with students, and administrate the faculties.
transaction	Transaction's information	The transaction gets transferred by the student.
administrativeStaff	administrative staff's information	administrative staff is a part of the university, handles the enrollment of students helping the university.
faculty members	faculty members's information	faculty members are a part of a faculty, manages courses, students and updates the course information.

### 5.2 Description of Relationship

Entity	Multiplicity	Relationship	Multiplicity	Entity
student	1...*	register	1...1	faculty
	1...1	manage	1...*	course

	1...*	enroll	1...1	university
	0...*	enroll	1...*	course
	1...1	transfer	1...*	transaction
faculty	1...1	register	1...*	student
	1...*	administrate	1...1	university
	1...1	has	1...*	faculty member
university	1...1	enroll	1...*	student
	1...1	administrate	1...*	faculty
	1...1	has	1...*	Administrative Staff
course	1...*	manage	1...1	student
	0...*	update	1...1	faculty member
	1...*	enroll	0...*	student
transaction	1...*	transfer	1...1	student
administrative staff	1...1	enroll	1...*	student
	1...*	has	1...1	university
Faculty member	1...*	has	1...1	faculty
	1...1	manage	1...*	course
	1...1	update	0...*	course

### 5.3 Description of Attributes

Entity	Attributes	Description	Datatype	Null	Multivalued
student	MatricNo	Uniquely identifies a student (PK)	Varchar(10)	no	no
	name	Name of student	Varchar(20)	no	no
	fName	First name of student	Varchar(10)	no	no
	lName	Last name of student	Varchar(10)	no	no
	degree	Degree of student	varchar(50)	no	no
	address	Address of student	Varchar(40)	no	no

	FacultyNo	Foreign key of faculty which uniquely identifies the faculty	INT	no	no
	transactionID	Foreign key of transaction which uniquely identifies the transaction	Varchar(20)	no	yes
	Courses	the courses of student	varchar(100)	no	yes
faculty	FacultyNO	Uniquely identifies a faculty (PK)	INT	no	no
	FacultyName	Holds the name of each faculty in the university.	Varchar(20)	no	no
	MatricNo	Foreign key of student which uniquely identifies the student (FK)	Varchar(10)	no	yes
	StaffID	Foreign key of faculty member which uniquely identifies the faculty members.	Varchar(10)	no	yes
university	UniversityName	Uniquely identifies a university (PK)	Varchar(30)	no	no
	UniversityAddress	Address of University	Varchar(40)	no	no
	facultyNo	Foreign key of faculty which uniquely identifies the faculty (FK)	INT	no	yes
course	CourseCode	Uniquely identifies a course (PK)	Varchar(10)	no	no
	CourseName	Name of course	Varchar(25)	no	no

	SectionNo	Number of the section	INT	yes	no
	CreditHours	The credit hours of course	INT	no	no
	FacultyName	Foreign key of faculty which uniquely identifies the faculty (FK)	Varchar(20)	yes	no
transaction	TransactionID	Uniquely identifies a transaction (PK)	Varchar(20)	no	no
	bills	bills paid by transaction	Varchar(100)	no	no
	MatricNo	Foreign key of student which uniquely identifies the student (FK)	Varchar(10)	no	no
	AmountPaid	The amount paid by transaction	INT	no	no
administrative Staff	AdministerID	Uniquely identifies an administer (PK)	Varchar(10)	no	no
	name	Name of student	Varchar(20)	no	no
	fName	First name of administer	Varchar(10)	no	no
	lName	Last name of administer	Varchar(10)	no	no
faculty member	staffID	Uniquely identifies a faculty member (PK)	Varchar(10)	no	no
	name	name of faculty member	Varchar(20)	no	no
	fName	First name of faculty member	Varchar(10)	no	no
	lName	last name of faculty member	Varchar(10)	no	no
	facultyNo	Foreign key of faculty with uniquely identifies (FK)	INT	no	no

## **6. Summary**

During this phase, our group is able to learn more of the ground on which the student management system is established in terms of functionality because the database conceptual design has been accomplished. We improved the proposed business rules, data needs and transactional needs in order to fulfil the goals of the system. Also, we designed the context data flow diagram, the parent data flow diagram, and the child data flow diagram. A Conceptual Entity-Relationship diagram or ERD was also created as well as an Enhanced ERD and the data dictionary that accompanied it.

Through the data flow diagrams as a result it became easier to identify how different entities and processes relate to one another to form the architecture of the system. At level (0), within the parent diagram and at level (1), within child diagrams both demonstrate a detailed account of processes like student enrollment, course registration as well as modifications to courses.

From the Data and Transaction Requirements, we learned that the new system should be powerful enough to solve the problems of delays, inefficiency, and data management.

Both the Conceptual ERD and Enhanced ERD defined the core database structure, drew out relationship details between entities and acted as roadmaps to the proposed system. It made it easier to understand the various fields displayed under the tabs, and the flows between them were easier to perceive due to the existence of the data dictionary.



Faculty of  
Computing

SECD2523-06 – DataBase

---

STUDENT MANAGEMENT SYSTEM

PHASE 3

DATABASE LOGICAL DESIGN

---

SECTION : 06

LECTURER'S NAME : Dr. Layla Hasan

DATE : 5.1.2024

No.	Name	Metric No.
1	Moaz Adil Abdugadir Jalal	A23CS3025
2	EYAD AIMEN ELSHEIKH KHALIL	A23CS3024
3	Abdalla Ali Abdalla Ali	A23CS3022
4	Ali Isameldin Ali Abdelrhman	A23CS3001
5	Othman Hassan Othman Ali	A23CS3026

## Table of Contents

No.	Content	Page
1	<a href="#"><u>Introduction</u></a>	3
2	<a href="#"><u>Overview of project</u></a>	3
3	<a href="#"><u>Database Conceptual design</u></a>	4
4	<a href="#"><u>Database logical design</u></a>	6
5	<a href="#"><u>Relational Database Schemas</u></a>	14
6	<a href="#"><u>SQL Statements (DDL &amp; DML)</u></a>	17
7	<a href="#"><u>Interface</u></a>	38
8	<a href="#"><u>Summary</u></a>	45

## **1.0 Introduction:**

Recently there are a lot of students enrolling in different universities every year, because of their high ranking and reputation. And from that the students expect an efficient and high level of management. The university system may use an inefficient database system or a file-based approach which will make it difficult to maintain the growing volume of students' information recorded. And this requires a new system that uses the database approach to maintain the student's information efficiently and effectively.

An effective student management system is the base of providing quality services for the different students and makes it easier for the different faculties to maintain and manipulate this information. From that the development of a new student management system will be effective, it will enhance the overall processing and retrieval of information, reduce data redundancy and the overall student registration experience.

## **2.0 Overview of project:**

This project is about the design of a database for a student management system. The first thing we did is to identify the users of the system and their interactions with the data. After that we produced the Entity Relationship Diagram ERD and the data dictionary, after that we produced the Enhanced ERD (EERD). and finally, from the EERD we produced the logical ERD and updated the data dictionary.

From these we did the normalization to ensure there are no data redundancy in our designed relational database, from that we produce the relational schema that is used to produce the actual database.

From the relational schema we produced our database using SQL commands and designed the user interface and ensured that the proposed design matches with the business rule and data requirements of the users.

## **3.0 Database conceptual design:**

### **3.1 Updated business rule:**

#### **Student:**

1. Enrolls at the university.
2. View the courses.
3. Drop courses.
4. Pay tuition fees.

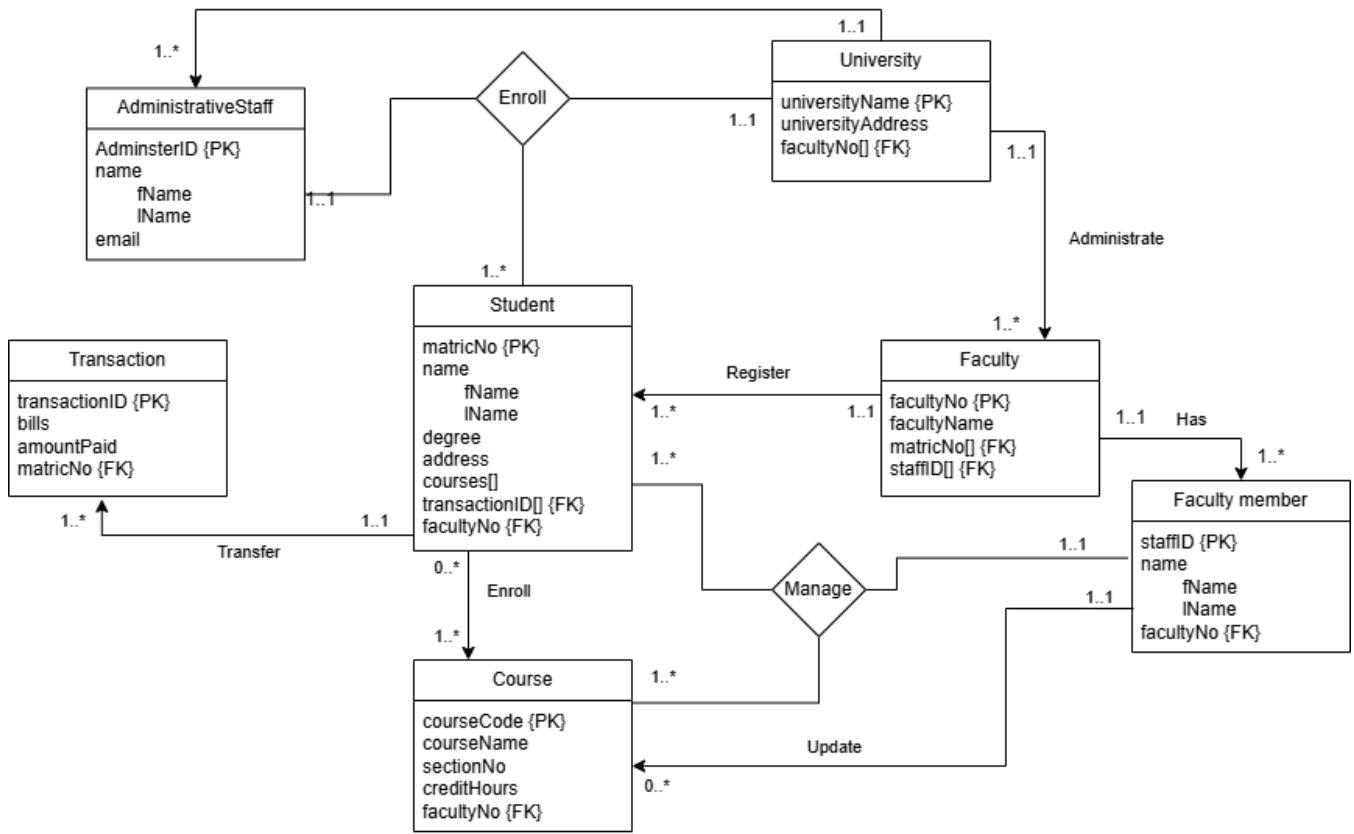
#### **Staff at faculty:**

1. Registers students at the specified faculty.
2. Manage students' issues regarding courses.
3. Update courses and their information.

#### **Administrative:**

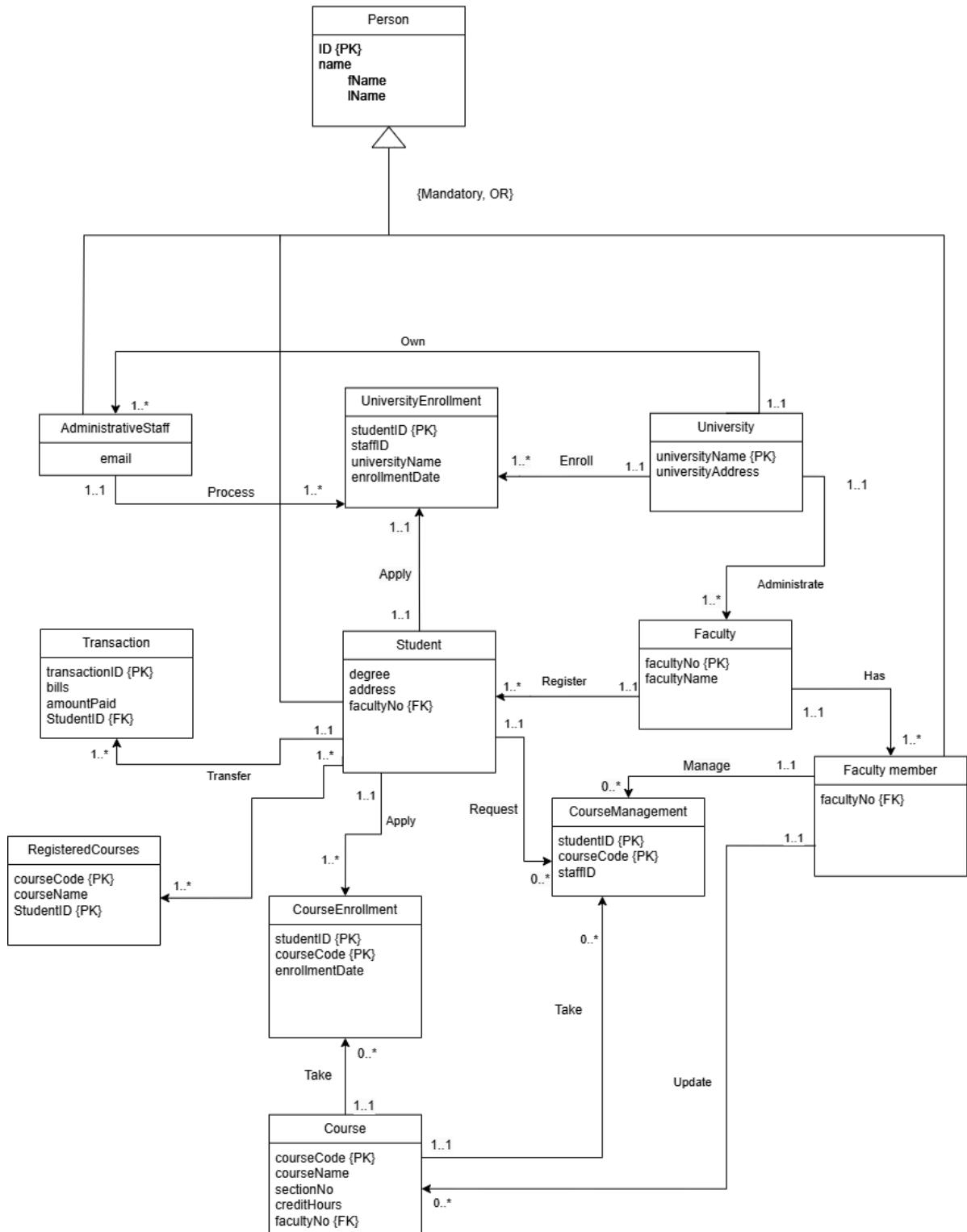
1. Enroll students at the university.

### 3.2 Conceptual ERD:



## 4.0 Database logical design:

### 4.1 Logical ERD:



## **4.2 Updated Data Dictionary:**

Description of Entity

### **4.2.1 Description of Entity**

<b>entity</b>	<b>description</b>	<b>occurrence</b>
Person	Superclass for Patient, Dentist, and Receptionist	Person can be associated with either Patient, Dentist, or Receptionist, but not more than one at a time.
student	Student information	Students can enroll in university, make transfers based on the university fees, and get registered in a faculty while being managed by faculty members along with student courses.
faculty	Faculty's information	Faculty has faculty members, registers students in a faculty and gets administered by the university.
courseEnrollment	Course's enrollment information	a student can apply for course enrollment and be updated in Course table.
courseManagement	Course's management information	a student can request for course management, managed by the faculty member to be updated in Course table.
course	course's information	updated by the faculty member, takes the courseManagement and apply for the courseEnrollment.
RegisteredCourses	list of student's courses	student has courses registered with matric number
university	University's information	a University has administrative staff who get enrolled along with students, and administrate the faculties.
UniversityEnrollment	student's enrollment information	students apply for the university enrollment and get processed by the administrative staff, and enrolled by the university.
transaction	Transaction's information	The transaction gets transferred by the student.

administrativeStaff	administrative staff's information	administrative staff is a part of the university, handles the enrollment of students helping the university.
faculty members	faculty members's information	faculty members are a part of a faculty, manages courses, students and updates the course information.

#### 4.2.2 Description of Relationship

Entity	Multiplicity	Relationship	Multiplicity	Entity
student	1...*	register	1...1	faculty
	1...1	request	0...*	courseManagement
	1...1	apply	1...1	universityEnrollment
	1...1	apply	1...*	courseManagement
	1...1	transfer	1...*	transaction
	1...*	has	1...*	RegisteredCourses
faculty	1...1	register	1...*	student
	1...*	administrate	1...1	university
	1...1	has	1...*	faculty member
university	1...1	enroll	1...*	universityEnrollment
	1...1	administrate	1...*	faculty
	1...1	own	1...*	AdministrativeStaff
universityEnrollment	1...*	enroll	1...1	university
	1...*	process	1...1	AdministrativeStaff
	1...1	apply	1...1	student
course	1...1	take	0...*	courseManagemet
	0...*	update	1...1	faculty member
	1...1	accept	0...*	courseEnrollment
courseManagement	1...*	manage	1...1	Faculty member

	0...*	take	1...1	course
	1...*	apply	1...1	student
courseEnrollment	1...*	request	1...1	student
	0...*	accept	1...1	course
RegisteredCourses	1...*	has	1...*	student
transaction	1...*	transfer	1...1	student
administerativeStaff	1...1	process	1...*	universityEnrollment
	1...*	own	1...1	university
Faculty member	1...*	has	1...1	faculty
	1...1	manage	1...*	courseManagement
	1...1	update	0...*	course

#### 4.2.3 Description of Attributes

Entity	Attributes	Description	Datatype	Null	Multivalued
person	ID	Uniquely identifies a person (PK)	Varchar(10)	no	no
	name	Name of person	Varchar(20)	no	no
	fName	First name of person	Varchar(10)	no	no
	lName	Last name of person	Varchar(10)	no	no
student	degree	Degree of student	varchar(50)	no	no
	address	Address of student	Varchar(40)	no	no
	FacultyNo	Foreign key of faculty which uniquely identifies the faculty	INT	no	no

	transactionID	Foreign key of transaction which uniquely identifies the transaction	Varchar(20)	no	yes
faculty	FacultyNO	Uniquely identifies a faculty (PK)	INT	no	no
	FacultyName	Holds the name of each faculty in the university.	Varchar(20)	no	no
university	UniversityName	Uniquely identifies a university (PK)	Varchar(30)	no	no
	UniversityAddress	Address of University	Varchar(40)	no	no
UniversityEnrollment	studentID	Uniquely identifies a student (PK)	Varchar(10)	no	no
	staffID	Uniquely identifies a staff	Varchar(10)	no	no
	universityName	Uniquely identifies a university	Varchar(30)	no	no
	enrollmentDate	Date of enrollment	DATE	no	no
course	CourseCode	Uniquely identifies a course (PK)	Varchar(10)	no	no
	CourseName	Name of course	Varchar(25)	no	no
	SectionNo	Number of the section	INT	yes	no
	CreditHours	The credit hours of course	INT	no	no
	FacultyName	Foreign key of faculty which uniquely identifies the faculty (FK)	Varchar(20)	yes	no
courseManagement	studentID	Uniquely identifies a student (PK)	Varchar(10)	no	no
	courseCode	Uniquely	Varchar(10)	no	no

		identifies a course (PK)			
	staffID	Uniquely identifies a staff	Varchar(10)	no	no
courseEnrolment	studentID	Uniquely identifies a person (PK)	Varchar(10)	no	no
	courseCode	Uniquely identifies a course (PK)	Varchar(10)	no	no
	enrollmentDate	the date of course enrollment	DATE	no	no
	TransactionID	Uniquely identifies a transaction (PK)	Varchar(20)	no	no
transaction	bills	bills paid by transaction	Varchar(100)	no	no
	MatricNo	Foreign key of student which uniquely identifies the student (FK)	Varchar(10)	no	no
	AmountPaid	The amount paid by transaction	INT	no	no
	email	email of administer	Varchar(25)	no	no
administrativeStaff	facultyNo	Foreign key of faculty with uniquely identifies (FK)	INT	no	no

### **4.3 Normalization:**

1. Student(ID, fName, lName, degree, address, facultyNo)

fd1: ID → fName, lName, degree, address, facultyNo

1NF&2NF&3NF&BNCF:

Student(ID, fName, lName, degree, address, facultyNo)

2. AdministrativeStaff(ID, fName, lName, email, UniversityName)

fd1: ID → fName, lName, email, UniversityName

1NF&2NF&3NF&BNCF:

AdministrativeStaff(ID, fName, lName, email, UniversityName)

3. FacultyMember(ID, fName, lName, FacultyNo)

fd1: ID → fName, lName, FacultyNo

1NF&2NF&3NF&BNCF:

FacultyMember(ID, fName, lName, FacultyNo)

4. University(universityName, universityAddress)

fd1: universityName → universityAddress

1NF&2NF&3NF&BNCF:

University(universityName, universityAddress)

5. Faculty(FacultyNo, facultyName)

fd1: FacultyNo → facultyName

1NF&2NF&3NF&BNCF:

Faculty(FacultyNo, facultyName)

6. Course(courseCode, courseName, sectionNo, creditHours)

fd1: courseCode → facultyName, courseName, sectionNo, creditHours

1NF&2NF&3NF&BNCF:

Course(courseCode, courseName, sectionNo, creditHours)

7. Transaction(TransactionID, bills, amountPaid, studentID)

fd1: TransactionID → bills, amountPaid, studentID

1NF&2NF&3NF&BNCF:

Transaction(TransactionID, bills, amountPaid, studentID)

8. UniversityEnrollment(StudentID, staffID, universityName, enrolmentDate)

fd1: StudentID → staffID, universityName, enrolmentDate

1NF&2NF&3NF&BNCF:

UniversityEnrollment(StudentID, staffID, universityName, enrolmentDate)

9. CourseManagement(StudentID, courseCode, staffID)

fd1: StudentID, courseCode → staffID

1NF&2NF&3NF&BNCF:

CourseManagement(StudentID, courseCode, staffID)

10. RegisteredCourse(courseCode, studentID, courseName)

fd1: courseCode → courseName, studentID

1NF&2NF&3NF&BNCF:

RegisteredCourse(courseCode, studentID, courseName)

11. CourseEnrollment(courseCode, studentID, enrollmentDate)

fd1: courseCode, StudentID → enrollmentDate

1NF&2NF&3NF&BNCF:

CourseEnrollment(courseCode, studentID, enrollmentDate)

## 5.0 Relational Database Schemas:

Student	( <u>ID</u> , fName, lName, degree, address, facultyNo)
adminsterativeStaff	( <u>ID</u> , fName, lName, email, UniversityName)
FacultyMember	( <u>ID</u> , fName, lName, FacultyNo)
University	( <u>universityName</u> , universityAddress)
Faculty	( <u>FacultyNo</u> , facultyName)
Course	( <u>courseCode</u> , courseName, sectionNo, creditHours)
Transaction	( <u>TransactionID</u> , bills, amountPaid, studentID)
UniversityEnrollment	( <u>StudentID</u> , staffID, universityName, enrolmentDate)
courseManagement	( <u>StudentID</u> , <u>courseCode</u> , staffID)
RegisteredCourses	( <u>courseCode</u> , <u>studentID</u> , courseName,)
courseEnrollment	( <u>courseCode</u> , <u>courseName</u> , studentID)

**Student**

ID	fName	lName	degree	address	facultyNo
----	-------	-------	--------	---------	-----------

**administrativeStaff:**

ID	fName	lName	email	universityName
----	-------	-------	-------	----------------

**FacultyMember**

ID	fName	lName	facultyNo
----	-------	-------	-----------

**University**

universityName	universityAddress
----------------	-------------------

**Faculty**

FacultyNo	facultyName
-----------	-------------

**Course**

CourseCode	CourseName	sectionNo	CreditHours
------------	------------	-----------	-------------

**Transaction**

TransactionId	bills	amountPaid	studentID
---------------	-------	------------	-----------

**UniversityEnrollment**

studentID	staffID	universityName	enrolmentDate
-----------	---------	----------------	---------------

**RegisteredCourses**

courseCode	studentID	courseName
------------	-----------	------------

**courseManagement**

courseID	studentID	staffID:
----------	-----------	----------

**courseEnrollment**

courseCode	courseName	studentID
------------	------------	-----------

## 6.0 SQL Statements (DDL & DML):

### 6.1 DDL:

-- Making sure that the database doesn't exist

```
DROP DATABASE IF EXISTS STUDENT_MANAGEMENT_SYSTEM;
```

-- Creating System DataBase

```
CREATE DATABASE STUDENT_MANAGEMENT_SYSTEM;
```

-- using the database in order to create tables

```
USE STUDENT_MANAGEMENT_SYSTEM;
```

-- Creating Student Table

```
CREATE TABLE IF NOT EXISTS Student (
```

```
    ID VARCHAR(10) PRIMARY KEY,      -- The student ID or matric number, Primary Key.
```

```
    fName VARCHAR(10) NOT NULL,     -- First name of the student, must not be empty.
```

```
    lName VARCHAR(10) NOT NULL,     -- Last name of the student, must not be empty.
```

```
    degree VARCHAR(50) NOT NULL,    -- The degree which the student is enrolling in, must not be  
    empty.
```

```
    Address VARCHAR(40) NOT NULL,   -- The address of the student, must not be empty.
```

```
    facultyNO VARCHAR(10) NOT NULL  -- The student's faculty number, must not be empty.
```

```
);
```

-- Creating AdministrativeStaff Table

CREATE TABLE IF NOT EXISTS AdministrativeStaff (

    ID VARCHAR(10) PRIMARY KEY,

        -- The administrative staff ID, Primary Key.

    fName VARCHAR(10) NOT NULL,  
    empty.

        -- First name of the administrative staff, must not be

    lName VARCHAR(10) NOT NULL,  
    empty.

        -- Last name of the administrative staff, must not be

    email VARCHAR(25) NOT NULL,

        -- Email of the administrative staff, must not be empty.

    UniversityName VARCHAR(30) NOT NULL

        -- The administrative staff's university name.

);

-- Creating facultyMember Table

CREATE TABLE IF NOT EXISTS facultyMember (

    ID VARCHAR(10) PRIMARY KEY,

        -- The faculty member ID, Primary Key.

    fName VARCHAR(10) NOT NULL,  
    empty.

        -- First name of the faculty member, must not be

    lName VARCHAR(10) NOT NULL,  
    empty.

        -- Last name of the faculty member, must not be

    facultyNo INT NOT NULL

        -- The faculty number of the member, must not be empty.

);

-- Creating University Table

CREATE TABLE IF NOT EXISTS University (

    universityName VARCHAR(30) PRIMARY KEY, -- University name, Primary Key.

    universityAddress VARCHAR(40) NOT NULL -- University Address, must not be empty.

);

-- Creating Faculty Table

CREATE TABLE IF NOT EXISTS Faculty (

    facultyNO INT PRIMARY KEY, -- Faculty number, Primary Key.

    facultyName VARCHAR(20) NOT NULL -- Faculty name, must not be empty.

);

-- Creating Course Table

CREATE TABLE IF NOT EXISTS Course (

    courseCode VARCHAR(10) PRIMARY KEY, -- The course code, Primary Key.

    courseName VARCHAR(25) NOT NULL, -- Name of the course, must not be empty.

    sectionNo INT, -- Number of the section.

    creditHours INT NOT NULL -- The credit hours of the course, must not be empty.

);

-- Creating Transaction Table

CREATE TABLE IF NOT EXISTS Transaction (

```
    TransactionId VARCHAR(20) PRIMARY KEY,      -- The transaction ID, Primary Key.  
  
    Bills VARCHAR(100) NOT NULL,                -- The bill of the transaction, must not be empty.  
  
    amountPaid INT NOT NULL,                   -- The amount of money paid in the transaction, must not be empty.  
  
    studentID VARCHAR(10) NOT NULL            -- The ID of the student who does the transaction,  
must not be empty.  
  
);
```

-- Creating UniversityEnrollment Table

```
CREATE TABLE IF NOT EXISTS UniversityEnrollment (  
  
    studentID VARCHAR(10) PRIMARY KEY,          -- The student ID or matric number, Primary  
Key.  
  
    staffID VARCHAR(10) NOT NULL,               -- The staff ID, Foreign Key.  
  
    universityName VARCHAR(30) NOT NULL,         -- Name of the university, Foreign Key.  
  
    enrolmentDate DATE NOT NULL,                -- Date of the university enrollment, must not be  
empty.  
  
    CONSTRAINT fk_staff FOREIGN KEY (staffID) REFERENCES AdministrativeStaff (ID), --  
Referencing staffID from AdministrativeStaff table.  
  
    CONSTRAINT fk_university FOREIGN KEY (universityName) REFERENCES  
University(universityName) -- Referencing universityName from University table.  
  
);
```

-- Creating RegisteredCourse Table

CREATE TABLE IF NOT EXISTS RegisteredCourse (

courseCode VARCHAR(10) NOT NULL,	-- The course code, must not be empty.
courseName VARCHAR(25) NOT NULL,	-- Name of the course, must not be empty.
studentID VARCHAR(10) NOT NULL,	-- The student ID or matric number, Foreign key.
PRIMARY KEY (courseCode, studentID),	-- Composite primary key

CONSTRAINT fk\_student\_RegisteredCourse FOREIGN KEY (studentID) REFERENCES UniversityEnrollment(studentID) -- referencing the studentID as foreign key from the UniversityEnrollment table.

);

-- Creating courseManagement Table

CREATE TABLE IF NOT EXISTS courseManagement (

courseID VARCHAR(10) PRIMARY KEY,	-- The course ID, Primary Key.
studentID VARCHAR(10) NOT NULL,	-- The student ID or matric number, Foreign key from UniversityEnrollment table.
staffID VARCHAR(10) NOT NULL,	-- The ID of the Staff, Foreign key from AdministrativeStaff table.

CONSTRAINT fk\_student\_courseManagement FOREIGN KEY (studentID) REFERENCES UniversityEnrollment(studentID), -- Referencing studentID from UniversityEnrollment table.

CONSTRAINT fk\_staff\_courseManagement FOREIGN KEY (staffID) REFERENCES UniversityEnrollment(staffID) -- Referencing staffID from AdministrativeStaff table.

);

-- Creating CourseEnrollment Table

CREATE TABLE IF NOT EXISTS CourseEnrollment (

courseCode VARCHAR(10), -- The course code, part of the composite primary key.

studentID VARCHAR(10), -- The student ID or matric number, part of the composite primary key.

enrollmentDate DATE NOT NULL, -- Date of the course enrollment, must not be empty.

PRIMARY KEY (courseCode, studentID) -- Creating composite primary key using courseCode and studentID.

);

-- Add foreign key for facultyNO in Student table linking to Faculty table and modify facultyNo to INT

ALTER TABLE Student

MODIFY facultyNO INT NOT NULL,

ADD CONSTRAINT fk2\_facultyNO\_Student FOREIGN KEY (facultyNO) REFERENCES Faculty(facultyNO);

-- Add foreign key for AdministrativeStaff table linking UniversityName to University table

ALTER TABLE AdministrativeStaff

ADD CONSTRAINT fk2\_universityName\_AdministrativeStaff FOREIGN KEY (UniversityName) REFERENCES University(universityName);

-- Add foreign key for facultyMember table linking facultyNO to faculty table

ALTER TABLE facultyMember

ADD CONSTRAINT fk2\_facultyMember\_faculty FOREIGN KEY (facultyNo) REFERENCES Faculty(facultyNO);

-- Add foreign key for Transaction table linking studentID to UniversityEnrollment table

ALTER TABLE Transaction

ADD CONSTRAINT fk2\_student\_Transaction FOREIGN KEY (studentID) REFERENCES UniversityEnrollment(studentID);

## 6.2 DML:

-- Inserting a record into the University table

INSERT INTO University (universityName, universityAddress) VALUES  
('UTM', 'Johor Bahru, Malaysia');

-- Inserting records into the Faculty table

INSERT INTO Faculty (facultyNO, facultyName) VALUES  
(1, 'Computer Science'), -- Faculty number 1  
(2, 'Engineering'), -- Faculty number 2  
(3, 'Mathematics'); -- Faculty number 3

-- Inserting multiple records into the Student table

```
INSERT INTO Student (ID, fName, lName, degree, Address, facultyNO) VALUES  
('A23CS3026', 'Othman', 'Hassan', 'Computer Science', 'Khartoum', 1),  
('A23CS3022', 'Abdalla', 'Ali', 'Electrical Engineering', 'USA', 2),  
('A23CS3024', 'Eyad', 'Ayman', 'Mathematics', 'France', 3),  
('A23CS3001', 'Ali', 'Essameldin', 'Computer Science', 'London', 1),  
('A23CS3023', 'Muaz', 'Adil', 'Electrical Engineering', 'Italy', 2);
```

-- Inserting multiple records into the Administrative Staff table

```
INSERT INTO AdministrativeStaff (ID, fName, lName, email, UniversityName) VALUES  
('A001', 'Ahmad', 'Zulkifl', 'ahmad_zulkifl@email.com', 'UTM'),  
('A002', 'Rizwan', 'Mohammed', 'rizwan_mohammed@email.com', 'UTM'),  
('A003', 'Rifqi', 'Faiz', 'rifqi_faiz@email.com', 'UTM');
```

-- Insert values into facultyMember table

```
INSERT INTO facultyMember (ID, fName, lName, facultyNo)
```

```
VALUES
```

```
('FM001', 'Ahmed', 'brown', 1),  
('FM002', 'Lamar', 'Smith', 2),
```

```
('FM003', 'Taylor', 'Swift', 1),  
('FM004', 'Lebron', 'James', 3),  
('FM005', 'Michael', 'Townley', 3);
```

-- Inserting multiple records into the Course table

```
INSERT INTO Course (courseCode, courseName, sectionNo, creditHours) VALUES  
('SECJH1013', 'Programming Technique I', 6, 3),  
('SCSR2213', 'Network Communications', 7, 3),  
('SCSR1013', 'Digital Logic', 3, 8),  
('ULRF2002', 'Astronomy', 8, 2),  
('UHLM1122', 'Malay Language', 4, 2);
```

-- Insert values into CourseEnrollment table

```
INSERT INTO CourseEnrollment (courseCode, studentID, enrollmentDate)  
VALUES  
('SECJH1013', 'A23CS3026', '2025-01-06'),  
('SCSR2213', 'A23CS3026', '2025-01-06'),  
('SECJH1013', 'A23CS3022', '2025-01-06'),  
('SCSR2213', 'A23CS3022', '2025-01-06'),  
('SCSR1013', 'A23CS3024', '2025-01-08'),  
('ULRF2002', 'A23CS3001', '2025-01-08'),  
('UHLM1122', 'A23CS3023', '2025-01-11');
```

-- Inserting records into courseManagement table

```
INSERT INTO courseManagement (courseID, studentID, staffID) VALUES  
('SECJH1013', 'A23CS3026', 'A001'),  
('SCSR2213', 'A23CS3026', 'A001'),  
('SCSR1013', 'A23CS3022', 'A002'),  
('ULRF2002', 'A23CS3024', 'A003'),  
('UHLM1122', 'A23CS3001', 'A003');
```

### **6.3 Testing The different Functions of the system:**

#### **6.3.1 University Enrollment**

-- Student Enrollment in university

```
INSERT INTO UniversityEnrollment (studentID, staffID, universityName, enrolmentDate) VALUES  
('A23CS3026', 'A001', 'UTM', '2025-01-06 09:00:00'),  
('A23CS3022', 'A002', 'UTM', '2025-01-06 09:00:00'),  
('A23CS3024', 'A002', 'UTM', '2025-01-07 12:00:00'),  
('A23CS3001', 'A003', 'UTM', '2025-01-08 14:30:00'),  
('A23CS3023', 'A003', 'UTM', '2025-01-11 15:21:50');
```

#### **6.3.2 Course Registration**

-- Registering Course

```
INSERT INTO RegisteredCourse (courseCode, courseName, studentID) VALUES  
('SECJH1013', 'Programming Technique I', 'A23CS3026'),  
('SCSR2213', 'Network Communications', 'A23CS3026'),  
('SCSR1013', 'Digital Logic', 'A23CS3026'),  
('ULRF2002', 'Astronomy', 'A23CS3026'),  
('UHLM1122', 'Malay Language', 'A23CS3026');
```

### **6.3.3 Drop Course**

```
-- Dropping courses
```

```
DELETE FROM RegisteredCourse
```

```
WHERE courseCode = 'UHLM1122';
```

### **6.3.4 Make Transactions**

```
-- Making required Transaction for the students
```

```
INSERT INTO Transaction (TransactionId, Bills, amountPaid, studentID)
```

```
VALUES
```

```
('T001', 'Tuition Fee - Semester 1', 1200, 'A23CS3026'),  
('T002', 'Hostel Fee - Room B101', 800, 'A23CS3022'),  
('T003', 'Library Fine', 50, 'A23CS3024'),  
('T004', 'Sports Facility Fee', 300, 'A23CS3001'),  
('T005', 'Event Registration - Science Fair', 100, 'A23CS3023');
```

## 6.4 Test Query:

### 6.4.1 View Student Table:

-- View Student Table

```
SELECT * FROM Student;
```

Output:

ID	fName	lName	degree	Address	facultyNO
A23CS3001	Ali	Essameldin	Computer Science	London	1
A23CS3022	Abdalla	Ali	Electrical Engineering	USA	2
A23CS3023	Muaz	Adil	Electrical Engineering	Italy	2
A23CS3024	Eyad	Ayman	Mathematics	France	3
A23CS3026	Othman	Hassan	Computer Science	Khartoum	1

### 6.4.2 View AdministrativeStaff Table

-- View AdministrativeStaff Table

```
SELECT * FROM AdministrativeStaff;
```

Output:

ID	fName	lName	email	UniversityName
A001	Ahmad	Zulkifl	ahmad_zulkifl@email.com	UTM
A002	Rizwan	Mohammed	rizwan_mohammed@email.com	UTM
A003	Rifqi	Faiz	rifqi_faiz@email.com	UTM

#### **6.4.3 View FacultyMember Table**

-- View FacultyMember Table

```
SELECT * FROM facultyMember;
```

Output:

ID	fName	lName	facultyNo
FM001	Ahmed	brown	1
FM002	Lamar	Smith	2
FM003	Taylor	Swift	1
FM004	Lebron	James	3
FM005	Michael	Townly	3

#### **6.4.4 View University Table**

-- View University Table

```
SELECT * FROM University;
```

Output:

universityName	universityAddress
UTM	Johor Bahru, Malaysia

#### 6.4.5 View Faculty Table

-- View Faculty Table

```
SELECT * FROM Faculty;
```

Output:

facultyNO	facultyName
1	Computer Science
2	Engineering
3	Mathematics

#### 6.4.6 View Course Table

-- View Course Table

```
SELECT * FROM Course;
```

Output:

courseCode	courseName	sectionNo	creditHours
SCSR1013	Digital Logic	3	8
SCSR2213	Network Communications	7	3
SECJH1013	Programming Technique I	6	3
UHLM1122	Malay Language	4	2
ULRF2002	Astronomy	8	2

#### **6.4.7 View Transaction Table**

-- View Transaction Table

```
SELECT * FROM Transaction;
```

Output:

TransactionId	Bills	amountPaid	studentID
T001	Tuition Fee - Semester 1	1200	A23CS3026
T002	Hostel Fee - Room B101	800	A23CS3022
T003	Library Fine	50	A23CS3024
T004	Sports Facility Fee	300	A23CS3001
T005	Event Registration - Science Fair	100	A23CS3023

#### **6.4.8 View UniversityEnrollment Table**

-- View UniversityEnrollment Table

```
SELECT * FROM UniversityEnrollment;
```

Output:

studentID	staffID	universityName	enrolmentDate
A23CS3001	A003	UTM	2025-01-08
A23CS3022	A002	UTM	2025-01-06
A23CS3023	A003	UTM	2025-01-11
A23CS3024	A002	UTM	2025-01-07
A23CS3026	A001	UTM	2025-01-06

#### 6.4.9 View RegisteredCourse Table

-- View RegisteredCourse Table

```
SELECT * FROM RegisteredCourse;
```

Output:

courseCode	courseName	studentID
SCSR1013	Digital Logic	A23CS3026
SCSR2213	Network Communications	A23CS3026
SECJH1013	Programming Technique I	A23CS3026
ULRF2002	Astronomy	A23CS3026

#### 6.4.10 View CourseEnrollment Table

-- View CourseEnrollment Table

```
SELECT * FROM CourseEnrollment;
```

Output:

courseCode	studentID	enrollmentDate
SCSR1013	A23CS3024	2025-01-08
SCSR2213	A23CS3022	2025-01-06
SCSR2213	A23CS3026	2025-01-06
SECJH1013	A23CS3022	2025-01-06
SECJH1013	A23CS3026	2025-01-06
UHLM1122	A23CS3023	2025-01-11
ULRF2002	A23CS3001	2025-01-08

#### 6.4.11 View CourseManagement Table

-- View Course Management Table

```
SELECT * FROM courseManagement;
```

Output:

courseID	studentID	staffID
SCSR1013	A23CS3022	A002
SCSR2213	A23CS3026	A001
SECJH1013	A23CS3026	A001
UHLM1122	A23CS3001	A003
ULRF2002	A23CS3024	A003

## 6.5 Extra Queries:

-- Additional Useful Queries

-- View Student Full Information with Faculty

```
SELECT Student.ID, CONCAT(Student.fName, ' ', Student.lName) AS fullName,  
Student.degree, Faculty.facultyName, Student.Address  
FROM Student JOIN Faculty ON Student.facultyNO = Faculty.facultyNO;
```

Output:

ID	fullName	degree	facultyName	Address
A23CS3001	Ali Essameldin	Computer Science	Computer Science	London
A23CS3026	Othman Hassan	Computer Science	Computer Science	Khartoum
A23CS3022	Abdalla Ali	Electrical Engineering	Engineering	USA
A23CS3023	Muaz Adil	Electrical Engineering	Engineering	Italy
A23CS3024	Eyad Ayman	Mathematics	Mathematics	France

-- View Course Enrollment Details

```
SELECT CourseEnrollment.courseCode, Course.courseName,  
CourseEnrollment.studentID, CourseEnrollment.enrollmentDate,  
Course.creditHours FROM CourseEnrollment  
JOIN Course ON CourseEnrollment.courseCode = Course.courseCode;
```

Output:

courseCode	courseName	studentID	enrollmentDate	creditHours
SCSR1013	Digital Logic	A23CS3024	2025-01-08	8
SCSR2213	Network Communications	A23CS3022	2025-01-06	3
SCSR2213	Network Communications	A23CS3026	2025-01-06	3
SECJH1013	Programming Technique I	A23CS3022	2025-01-06	3
SECJH1013	Programming Technique I	A23CS3026	2025-01-06	3
UHLM1122	Malay Language	A23CS3023	2025-01-11	2
ULRF2002	Astronomy	A23CS3001	2025-01-08	2

-- View Transaction History

SELECT

```
Transaction.TransactionId, Transaction.Bills,  
Transaction.amountPaid,  
CONCAT(Student.fName, ' ', Student.lName) AS studentName,  
Student.ID as studentID  
  
FROM Transaction  
  
JOIN Student ON Transaction.studentID = Student.ID;
```

Output:

TransactionId	Bills	amountPaid	studentName	studentID
T001	Tuition Fee - Semester 1	1200	0	A23CS3026
T002	Hostel Fee - Room B101	800	0	A23CS3022
T003	Library Fine	50	0	A23CS3024
T004	Sports Facility Fee	300	0	A23CS3001
T005	Event Registration - Science Fair	100	0	A23CS3023

-- View University Enrollment Details

```
SELECT UniversityEnrollment.studentID,  
CONCAT(Student.fName, ' ', Student.lName) AS studentName,  
CONCAT(AdministrativeStaff.fName, ' ', AdministrativeStaff.lName) AS staffName,  
UniversityEnrollment.universityName,  
UniversityEnrollment.enrolmentDate  
FROM UniversityEnrollment  
JOIN Student ON UniversityEnrollment.studentID = Student.ID  
JOIN AdministrativeStaff ON UniversityEnrollment.staffID = AdministrativeStaff.ID;
```

Output:

studentID	studentName	staffName	universityName	enrolmentDate
A23CS3026	Othman Hassan	Ahmad Zulkifl	UTM	2025-01-06
A23CS3022	Abdalla Ali	Rizwan Mohammed	UTM	2025-01-06
A23CS3024	Eyad Ayman	Rizwan Mohammed	UTM	2025-01-07
A23CS3001	Ali Essameldin	Rifqi Faiz	UTM	2025-01-08
A23CS3023	Muaz Adil	Rifqi Faiz	UTM	2025-01-11

-- View Course Management Information

SELECT

courseManagement.courseID,

Course.courseName,

CONCAT(Student.fName, ' ', Student.lName) AS studentName,

CONCAT(AdministrativeStaff.fName, ' ', AdministrativeStaff.lName) AS staffName

FROM courseManagement

JOIN Course ON courseManagement.courseID = Course.courseCode

JOIN Student ON courseManagement.studentID = Student.ID

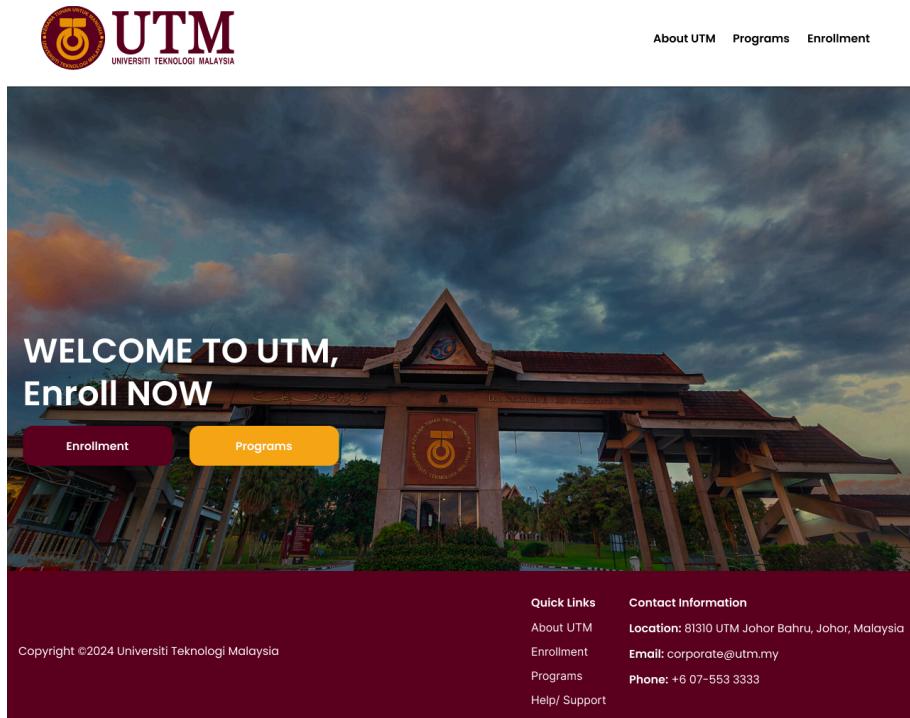
JOIN AdministrativeStaff ON courseManagement.staffID = AdministrativeStaff.ID;

Output:

courseID	courseName	studentName	staffName
SCSR2213	Network Communications	Othman Hassan	Ahmad Zulkifl
SECJH1013	Programming Technique I	Othman Hassan	Ahmad Zulkifl
SCSR1013	Digital Logic	Abdalla Ali	Rizwan Mohammed
UHLM1122	Malay Language	Ali Essameldin	Rifqi Faiz
ULRF2002	Astronomy	Eyad Ayman	Rifqi Faiz

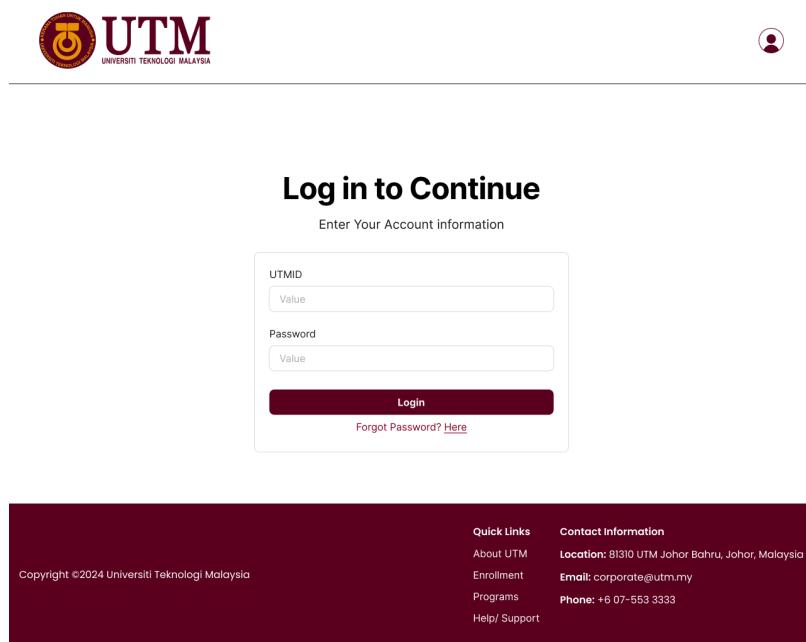
## 7.0 Interface:

### 7.1 Landing Page (Student View)



The screenshot shows the UTM student landing page. At the top left is the UTM logo with the text "UTM" and "UNIVERSITI TEKNOLOGI MALAYSIA". At the top right are links for "About UTM", "Programs", and "Enrollment". The main header features a large image of a traditional UTM building under a dramatic, cloudy sky. Overlaid on the image is the text "WELCOME TO UTM, Enroll NOW". Below the header are two buttons: "Enrollment" (purple) and "Programs" (yellow). A dark red footer bar at the bottom contains the copyright notice "Copyright ©2024 Universiti Teknologi Malaysia" and two columns of links: "Quick Links" (About UTM, Enrollment, Programs, Help/ Support) and "Contact Information" (Location: 81310 UTM Johor Bahru, Johor, Malaysia; Email: corporate@utm.my; Phone: +6 07-553 3333).

### 7.2 Login Page



The screenshot shows the UTM login page. At the top left is the UTM logo. To the right is a user icon. The main title "Log in to Continue" is centered above a form field labeled "Enter Your Account information". The form contains two input fields: "UTMID" and "Password", each with a placeholder "Value". Below the fields is a "Login" button. Underneath the button is a link "Forgot Password? Here". A dark red footer bar at the bottom contains the copyright notice "Copyright ©2024 Universiti Teknologi Malaysia" and two columns of links: "Quick Links" (About UTM, Enrollment, Programs, Help/ Support) and "Contact Information" (Location: 81310 UTM Johor Bahru, Johor, Malaysia; Email: corporate@utm.my; Phone: +6 07-553 3333).

## 7.3 Enrollment Center (Student View)



About UTM Programs Enrollment

### Create Account

First Name \*

Last Name \*

Phone Number \*

+60123334444

Email \*

Password \*

Confirm Password \*

**Create Account**

Already have an account? [Login](#)

---

Copyright ©2024 Universiti Teknologi Malaysia



About UTM Programs Enrollment

### Enrollment Application

#### Section A: Personal Info

Full Name \*

Date of Birth \*

DD/MM/YYYY

Nationality \*

Select Nationality

IC/ Passport Number \*

IC/ Passport \*

Upload 1 PDF file. Max 5 MB.

Address \*

**Continue**

---

Copyright ©2024 Universiti Teknologi Malaysia



About UTM Programs Enrollment

### Enrollment Application

#### Section B: Field of Study

Program \*

Select Program

**Back** **Continue**

---

Copyright ©2024 Universiti Teknologi Malaysia



About UTM Programs Enrollment

### Enrollment Application

#### Section C: Academic Qualification

Academic Qualification \*

Select Qualification type

Institution Name \*

Graduation Date \*

DD/MM/YYYY

Certificate \*

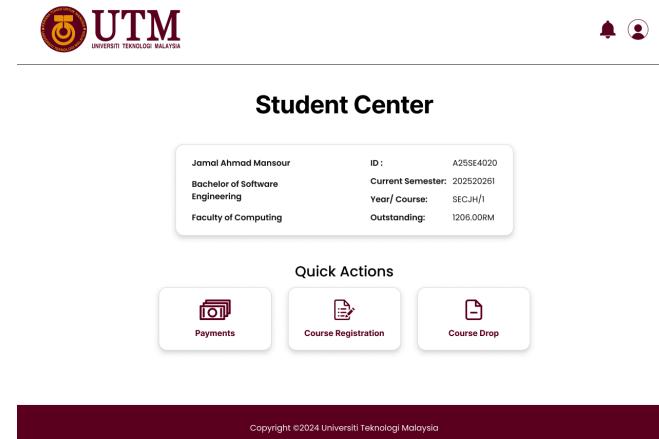
Upload 1 PDF file. Max 5 MB.

**Back** **Submit**

---

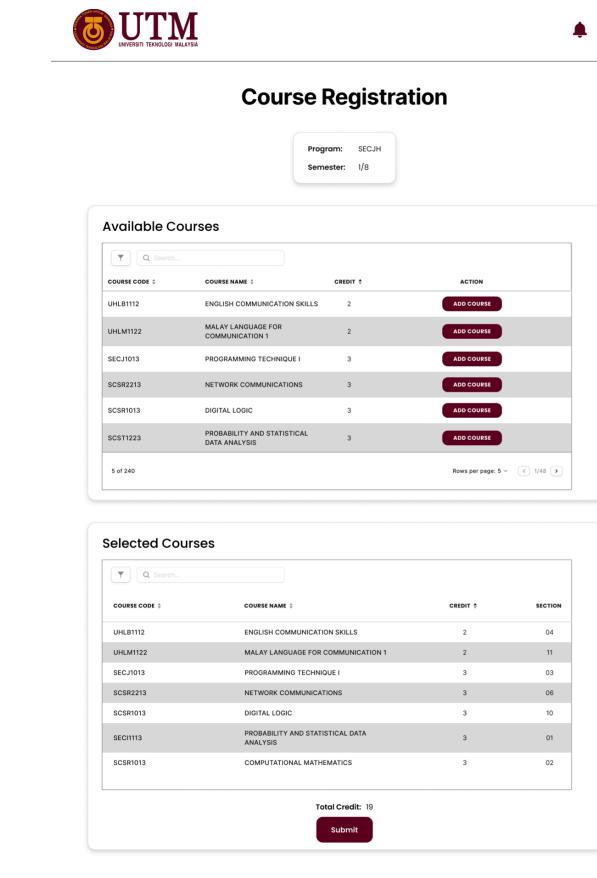
Copyright ©2024 Universiti Teknologi Malaysia

## 7.4 Student View (Home Page)



The screenshot shows the UTM Student Center home page. At the top right are notifications and user profile icons. Below the header is a student profile box containing basic information: Name (Jamal Ahmad Mansour), ID (A25SE4020), Program (Bachelor of Software Engineering), Semester (Current Semester: 2025/2026I), Year/Course (SECJH/I), and Outstanding (1206.00RM). Below this is a "Quick Actions" section with three buttons: Payments, Course Registration, and Course Drop. A footer bar at the bottom contains the copyright notice: Copyright ©2024 Universiti Teknologi Malaysia.

## 7.5 Course Registration



The screenshot shows the UTM Course Registration page. At the top right are notifications and user profile icons. Below the header is a search bar and filter options for Program (SECJH) and Semester (1/B). The main area is divided into two sections: "Available Courses" and "Selected Courses". The "Available Courses" section lists various modules with their codes, names, credits, and an "ADD COURSE" button. The "Selected Courses" section lists the modules chosen by the student, showing course code, name, credit, and section. A summary at the bottom indicates a total credit of 19 and a "Submit" button. A footer bar at the bottom contains the copyright notice: Copyright ©2024 Universiti Teknologi Malaysia.

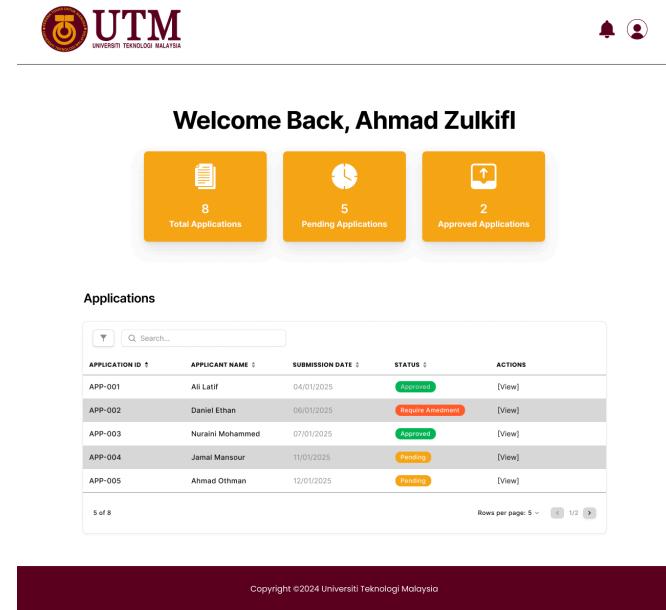
## 7.6 Course Drop

The screenshot shows the UTM Course Drop interface. At the top, there's a header with the UTM logo and navigation icons. Below it, a title 'Course Drop' is centered. A summary box at the top right provides program information: Program: SEC.JH, Registered Courses: 6, Semester: 1/8, and Registered Credit: 19. The main area is divided into two sections: 'Registered Courses' and 'Dropped Courses'.  
**Registered Courses:** This section displays a table of courses with columns: COURSE CODE, COURSE NAME, CREDIT, and ACTION. The table includes rows for UHLB1112 (ENGLISH COMMUNICATION SKILLS), UHLM1122 (MALAY LANGUAGE FOR COMMUNICATION 1), SECJ1013 (PROGRAMMING TECHNIQUE I), SCSR2213 (NETWORK COMMUNICATIONS), SCSR1013 (DIGITAL LOGIC), and SCST1223 (PROBABILITY AND STATISTICAL DATA ANALYSIS). Each row has a 'DROP COURSE' button.  
**Dropped Courses:** This section shows a table with one row for UHLM1122 (MALAY LANGUAGE FOR COMMUNICATION 1) with a credit of 2 and a section of 11. Below the table, a message says 'Total Credit After Dropping: 17' and a 'Confirm' button.

## 7.7 Payments

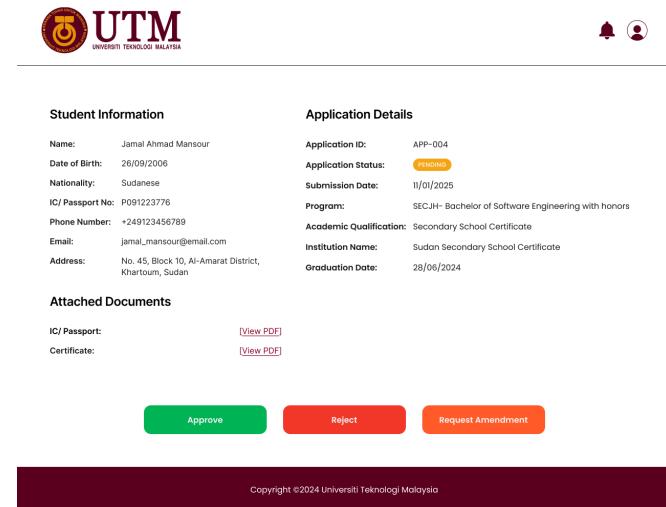
The screenshot shows the UTM Payment interface. It features a header with the UTM logo and navigation icons. Below the header, there are two main sections: 'Financial Center' and 'Payment'.  
**Financial Center:** This section shows an 'Outstanding' balance of 1206.00RM. Below this is a table of fees with columns: NO., FEES, AMOUNT, and ACTION. One row is shown: SEMESTER 1 FEES with an amount of 1206.00RM and a 'PAY' button.  
**Payment:** This section allows users to select a payment method (Card) and enter card information (Card number, MM/YY, CVC, Holder Name). A 'Pay' button is at the bottom.  
At the bottom of each section, there is a copyright notice: 'Copyright ©2024 Universiti Teknologi Malaysia'.

## 7.8 Administrative Staff View (Home Page)



The screenshot shows the UTM administrative staff home page. At the top, there is the UTM logo and a navigation bar with icons for notifications and user profile. Below the header, a welcome message "Welcome Back, Ahmad Zulkifl" is displayed. Three orange cards provide a summary of application status: "Total Applications" (8), "Pending Applications" (5), and "Approved Applications" (2). A section titled "Applications" lists five entries with columns for Application ID, Applicant Name, Submission Date, and Status. The applications are: APP-001 (Ali Latif, 04/01/2025, Approved), APP-002 (Daniel Ethan, 06/01/2025, Require Amendment), APP-003 (Nuraini Mohammed, 07/01/2025, Approved), APP-004 (Jamal Mansour, 11/01/2025, Pending), and APP-005 (Ahmad Othman, 12/01/2025, Pending). The footer contains a copyright notice: "Copyright ©2024 Universiti Teknologi Malaysia".

## 7.9 Enrollment Application Details



The screenshot shows the details of an enrollment application. At the top, there is the UTM logo and a navigation bar with icons for notifications and user profile. The page is divided into two main sections: "Student Information" and "Application Details". Under "Student Information", the student's name is Jamal Ahmad Mansour, date of birth is 26/09/2006, nationality is Sudanese, IC/Passport No. is P09123776, phone number is +249123456789, email is jamal\_mansour@email.com, and address is No. 45, Block 10, Al-Amarat District, Khartoum, Sudan. Under "Application Details", the application ID is APP-004, status is PENDING, submission date is 11/01/2025, program is SEC.JH- Bachelor of Software Engineering with honors, academic qualification is Secondary School Certificate, institution name is Sudan Secondary School Certificate, and graduation date is 28/06/2024. Below these sections, there is a "Attached Documents" section showing PDF links for "IC/Passport" and "Certificate". At the bottom, there are three buttons: "Approve" (green), "Reject" (red), and "Request Amendment" (orange). The footer contains a copyright notice: "Copyright ©2024 Universiti Teknologi Malaysia".

## 7.10 Faculty Member View (Home Page)



Welcome Back, Jamal Eisa



### Quick Actions



#### Requests

STUDENT ID	TYPE	REQUEST DATE	STATUS	ACTIONS
A23CS3024	Course Register	04/01/2025	Approved	[View]
A23CS3025	Course Drop	06/01/2025	Rejected	[View]
A23CS3026	Course Register	11/01/2025	Pending	[View]
A23CS3027	Course Drop	12/01/2025	Pending	[View]

5 of 8 Rows per page: 5 ▾ 1/2 3

Copyright ©2024 Universiti Teknologi Malaysia

## 7.11 Faculty Request Details



#### Student Information

Name:	Ahmed Othman Musa
ID:	A23CS3026
Email:	ahmed_musa@mail.com
Phone Number:	+249123456789
Faculty:	Faculty of Computing
Program:	Bachelor of Software Engineering
Program/ Year:	SECJH/ 1
Credit Counted:	0/127
Registered Credit:	0

#### Request Information

Request Status:	PENDING
Request Date:	11/01/2025
Request Type:	Course Registration
No of Courses:	5

#### Student Information

Name:	Jamal Ahmad Mansour
ID:	A23CS3027
Email:	jamal_mansour@email.com
Phone Number:	+249123456789
Program/ Year:	SECJH/ 1
Credit Counted:	0/127
Registered Courses:	6
Registered Credit:	19

#### Request Information

Request Status:	PENDING
Request Date:	12/01/2025
Request Type:	Course Drop
No of Courses:	1

#### Course Registration Request

COURSE CODE	COURSE NAME	CREDIT	SECTION
UHLB1112	ENGLISH COMMUNICATION SKILLS	2	04
SCSR2213	NETWORK COMMUNICATIONS	3	06
SECJ1013	PROGRAMMING TECHNIQUE I	3	03
SECI1113	PROBABILITY AND STATISTICAL DATA ANALYSIS	3	01
SCSR1013	DIGITAL LOGIC	3	10
SCSR1013	COMPUTATIONAL MATHEMATICS	3	02

Registered Credit Before: 0  
Registered Credit After: 17

[Approve](#) [Reject](#)

#### Course Drop Request

COURSE CODE	COURSE NAME	CREDIT	SECTION
UHLM1122	MALAY LANGUAGE FOR COMMUNICATION 1	2	11

Registered Credit Before: 19  
Registered Credit After: 17

[Approve](#) [Reject](#)

Copyright ©2024 Universiti Teknologi Malaysia

Copyright ©2024 Universiti Teknologi Malaysia

## 7.12 Student Management



### Student Management

Faculty: Faculty of Computing  
Total Student: 430  
New Student: 120

#### Students

ID	NAME	PROGRAM/YEAR	ACTION
A23CS3021	Ali Latif	SECJH/1	[View]
A23CS3022	Daniel Ethan	SECJH/2	[View]
A23CS3023	Nuraini Mohammed	SECPH/2	[View]
A23CS3024	Jamal Mansour	SECJH/1	[View]
A23CS3025	Sufyan Ahmad	SECBH/2	[View]
A23CS3026	Ahmad Othman	SECJH/1	[View]

5 of 240

Rows per page: 5 < 1/48 >

#### Student Information

Name: Ahmad Othman Musa  
ID: A23CS3026  
Email: ahmad\_musa@email.com  
Phone Number: +249123456789  
Faculty: Faculty of Computing  
Program: Bachelor of Software Engineering  
Program/ Year: SECJH/1  
Credit Counted: 0/127  
Registered Credit: 0

#### Student Requests

STUDENT ID	TYPE	REQUEST DATE	STATUS	ACTIONS
A23CS3026	Course Register	11/01/2025	Pending	[View]

Rows per page: 5 < 1/1 >

Copyright ©2024 Universiti Teknologi Malaysia

Copyright ©2024 Universiti Teknologi Malaysia

## 7.13 Course Management



### Course Management



Add Course

#### Courses

COURSE CODE	COURSE NAME	CREDIT	ACTION
UHLB1112	ENGLISH COMMUNICATION SKILLS	2	[UPDATE] [DELETE]
UHLM1122	MALAY LANGUAGE FOR COMMUNICATION 1	2	[UPDATE] [DELETE]
SECJ1013	PROGRAMMING TECHNIQUE I	3	[UPDATE] [DELETE]
SCSR2213	NETWORK COMMUNICATIONS	3	[UPDATE] [DELETE]
SCSR1013	DIGITAL LOGIC	3	[UPDATE] [DELETE]
SCST1223	PROBABILITY AND STATISTICAL DATA ANALYSIS	3	[UPDATE] [DELETE]

5 of 240

Rows per page: 5 < 1/48 >

### Course Update

#### Course Information

Course Code: SECJ1013  
Course Name: PROGRAMMING TECHNIQUE I  
Sections: 4  
Credit Hours: 3

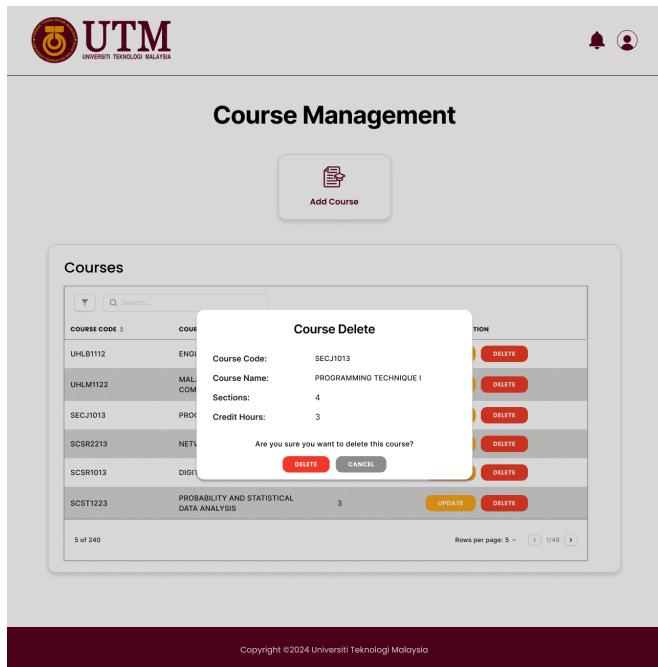
#### Information Update

Course Code: SECJ1013  
Course Name: PROGRAMMING TECHNIQUE I  
Credit Hours: 3

Update

Copyright ©2024 Universiti Teknologi Malaysia

Copyright ©2024 Universiti Teknologi Malaysia



## 8.0 Summary:

During this phase, we transformed our Conceptual Entity Relationship Diagram (ERD) into a Logical ERD, adjusting it to improve the student registration process. We removed features that didn't fit the relational model and refined relationships to follow database principles.

We also created the relational schema, turning each entity—like student, staff, university, faculty, and transaction—into tables with clear attributes and primary keys. By following the Boyce-Codd Normal Form (BCNF), we normalized the database to remove redundancies and dependencies, ensuring the information about patients, appointments, and transactions stays accurate and reliable.

The final Logical ERD visually shows how the database is organized, highlighting the relationships between entities and guiding efficient data retrieval and management. We also updated the data dictionary to match the normalized structure, ensuring everything is consistent and accurate.

In summary, we learned how to turn conceptual ideas into well-organized, normalized database structures. This work sets the stage for a smooth and reliable student enrollment process that simplifies university enrollment, registering courses, updating student information which really improves the experience for Students and staff.

