

THAPAR INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Deemed to be University)

Patiala, Punjab



A Mini Project On “University Management System”

For the partial complement of Database Management System 2025

Under the supervision of

Department of Computer Science and Engineering

Submitted By:

“Sajid Miya” 102367013

“Nikshitha” 102317043

“Arshdeep Singh” 102317064

“Manjot Singh Gill” 102317068

Submitted To:

THAPAR INSTITUTE OF ENGINEERING AND TECHNOLOGY

Department of Computer Science and Engineering

Patiala, Punjab, India

“June 2025”

Table of Contents

INTRODUCTION	1
Requirement Analysis	2
1. Functional Requirements.....	2
2. Non-Functional Requirements	2
ER Diagram	3
ER Diagram to Table.....	4

INTRODUCTION

The University Management System (UMS) is designed to efficiently manage various aspects of a university's operations, including student enrollment, faculty details, course management, exams, fee payments, and results processing. The system ensures data integrity, reduces redundancy, and improves administrative efficiency by organizing university-related data in a structured and secure manner.

This system is database-driven, making it easier for universities to manage academic records, track student progress, and streamline administrative processes. It is essential for maintaining accurate student and faculty records, ensuring smooth examination processes, and managing financial transactions related to tuition and fees.

Requirement Analysis

1. Functional Requirements

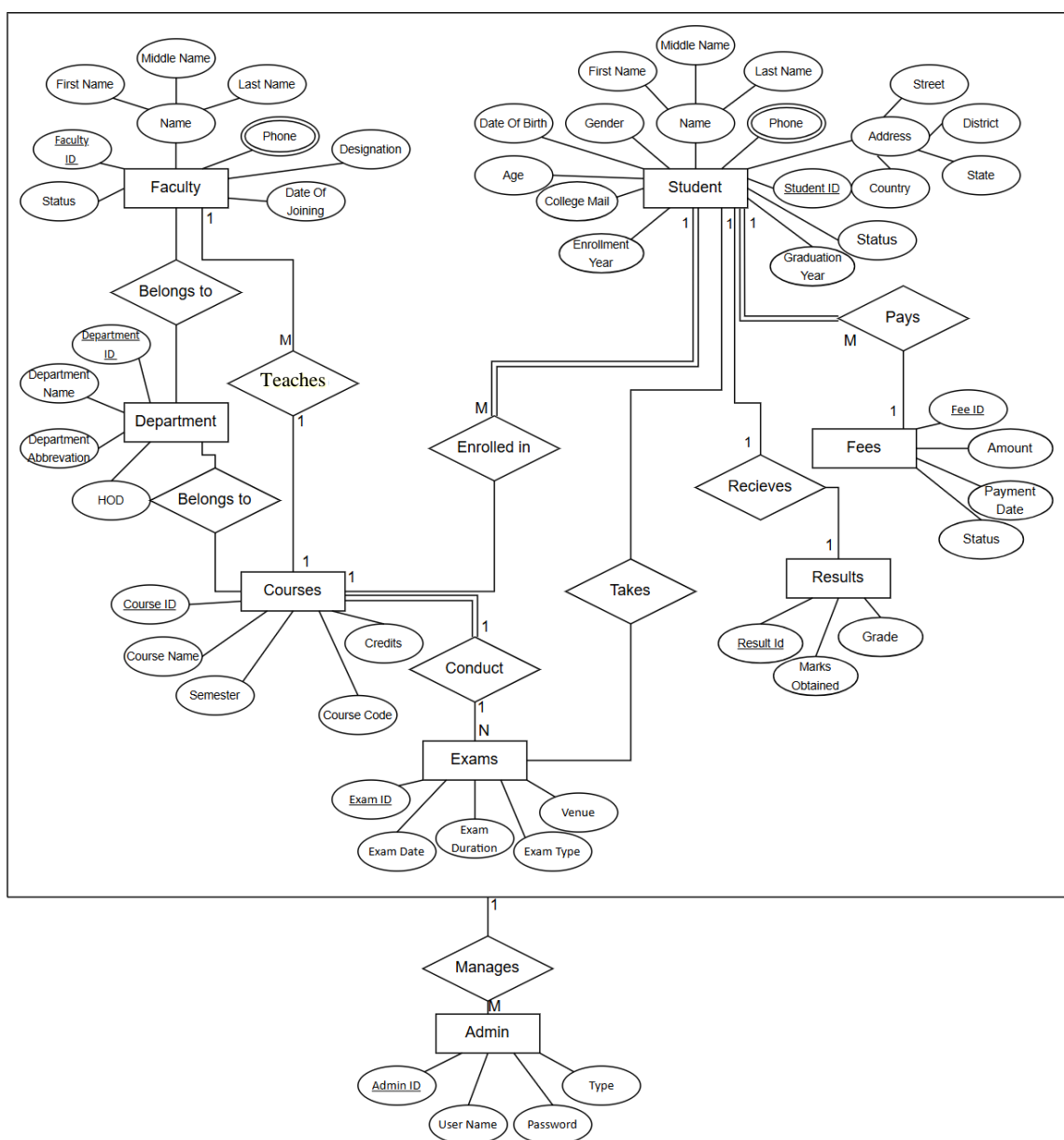
The system should provide the following functionalities:

- **Student Management** – Maintain student personal and academic records, enrollment details, and course selections.
- **Faculty Management** – Store faculty details, including personal information, designations, and assigned courses.
- **Course Management** – Manage courses, including course codes, credit hours, and department affiliations.
- **Enrollment System** – Allow students to enroll in courses and maintain their enrollment history.
- **Examination & Results** – Schedule exams, store exam details, and record student grades.
- **Fee Management** – Track student fee payments and statuses (paid, pending).
- **Admin Management** – Allow administrators to manage university operations efficiently.

2. Non-Functional Requirements

- **Scalability** – The system should handle a growing number of students, faculty, and courses.
- **Security** – Implement authentication and role-based access for students, faculty, and administrators.
- **Performance** – Ensure quick data retrieval and minimal response time for database queries.
- **Reliability** – Prevent data loss and ensure accurate record-keeping.
- **User-Friendly Interface** – Provide a simple and intuitive interface for users.

ER Diagram



ER Diagram to Table

Relation 'Enrolled In'

Students: student_id (PK), first_name, middle_name, last_name, street, district, state, country, gender, dob, age, phone, college_mail, enrollment_year, graduation_year, status
Courses: course_id (PK), course_name, course_code, credits, semester
Enrollment: enrollment_id (PK), student_id (FK), course_id (FK), semester, grade, enrollment_date

Relation 'Pays'

Students: student_id (PK), first_name, middle_name, last_name, ... (Other attributes)
Fees: fee_id (PK), student_id (FK), amount, payment_date, status

Relation 'Takes'

Students: student_id (PK), first_name, middle_name, last_name, ... (Other attributes)
Exams: exam_id (PK), exam_date, exam_duration, exam_type, venue
Takes: student_id (FK), exam_id (FK)

Relation 'Receives'

Students: student_id (PK), first_name, middle_name, last_name, ... (Other attributes)
Results: result_id (PK), student_id (FK), exam_id (FK), marks_obtained, grade

Relation 'Teaches'

Faculty: faculty_id (PK), first_name, middle_name, last_name, phone, date_of_joining, status, designation
Courses: course_id (PK), course_name, course_code, credits, semester
Teaches: faculty_id (FK), course_id (FK)

Relation 'Belongs to'

Faculty: faculty_id (PK), first_name, middle_name, last_name, ... (Other attributes)
Department: department_id (PK), department_name, department_abbreviation, head_of_department
BelongsTo: faculty_id (FK), department_id (FK)

Relation 'Conducts'

Courses: course_id (PK), course_name, course_code, credits, semester
Exams: exam_id (PK), exam_date, exam_duration, exam_type, venue
Conducts: course_id (FK), exam_id (FK)

Relation 'Belong to'

Courses: course_id (PK), course_name, course_code, credits, semester
Department: department_id (PK), department_name, department_abbreviation,

head_of_department

BelongTo: course_id (FK), department_id (FK)

Relation 'Manages'

Admin: admin_id (PK), username, password, type

Manages: admin_id (FK), student_id (FK), faculty_id (FK), course_id (FK), fee_id (FK)