University Database Student, Course and Instructors Information System

Introduction

University Database Management System is developed to manage the University's student database. As the University grows their database also grows and for the proper management of the data a great database management system is required. It eliminates manual work and helps universities achieve maximum productivity by computerizing student, staff and administration lifecycles and minimizing the hassles of the university administration.

University database management system is required to manage the records of the students. Managing student information, generating of enrolment number, admit cards generation are some of the modules of the University Database Management System. University Database Management System starts with the registration of new staff and students.

University Database Management System deals with the maintenance of University data, records, instructions, and student's information within the University. It is an automation system, which is used to store the information's, students record, and information of courses. Starting from registration of a new student in the university.

The University Database Management System is an improved Student Service, increase information sharing. It can handle all details about a student and the management system is managed by an Administrator. It is the job of the Administrator to insert update and monitor the whole process. The system will serve the management to reduce cycle times, faster keep track of data, and improve the service, increase information sharing and providing facilities to store information centrally.

The University Database Management System is to eliminate their manual counting of student taking a particular subject to a computerized system which can query and to shorten the time of retrieving data when required.

Requirements and constraints for Building University Database

Managing thousands of students and their information in a single university database might seem like a lot of work. Still, an ERD makes it easier to understand how the entities are connected to create a seamless system that is equally fantastic for students, teachers and other staff. However, it is not always easy and seamless.

To create a unique system, you would need to overcome all the constraints and requirements. Here are a few issues you might encounter while building a university database

The university offers multiple programs. Each of these programs is made from several courses. One student can enroll in a program and then access the courses in it. Keeping in mind that a university will have multiple departments. Each of these departments will have a HOD or Head of Department. Along with a HOD, there will be several other instructors. An instructor will only belong to one department. Each of these departments offers multiple courses to multiple students. Each of these courses is taught by multiple instructor. A single student might enroll themselves in several courses in one department. A student then requires certain credit hours to graduate. Each of the courses is sequenced according to the years or semesters. Once the student takes the course, the year they took it in, and the grade is recorded. Each course has a code or identifier, a credit point value and the time it commenced. Every student has a student id or username. All details about a student and the management system is managed by an Administrator. It is the job of the Administrator to insert update and monitor the whole process.

These constraints must be paid attention to while constructing an information system or database. This will allow all users to have a seamless experience and record all the critical information.

.

Entity Relationship Diagram(ERD) for University Database Management System

Now that we have enlisted all the constraints relevant to designing a university database, it's time to plan an ER diagram for the University database management system. We should follow several steps for designing of the entity relationship diagram.

Step 1

Identify all the entity sets that is part of the database

- Student
- Instructor
- Department
- Course
- Section
- Grade
- Prerequisite

Step 2

Once we identified all the entities, we have to assign the attributes. The attributes are characteristics that describe the entity making it easier to identify entities within entity sets. We have listed down the attributes for these entities.

- Student
  - Student name, Student id, Sex, Section id, Birthdate, Phone number, Course id, Department id

- Instructor
  - Instructor name, Instructor id, Room number office, Building number, Phone number, Section id

- Department
  - Department name, Department id, Room number Office, Building number, Phone number, Department head id

- Course
  - Course name, Course id, Course hour, Department id, Instructor id

- Section
  - Section id, Semester, Semester year, Course id, Instructor id

- Grade
  - Numeric grade, Letter grade, Student id, Section id, Course id

- Prerequisite
  - Prerequisite id, Course id

Step 3

Now, the next step is identifying the primary key attributes of an entity set. A key or primary attribute is unique for each entity within an entity set.

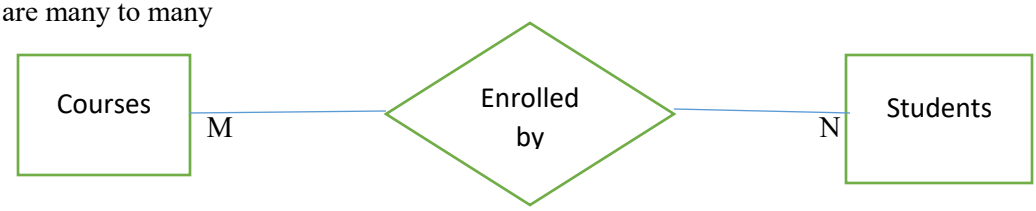
- Student id
- Instructor id
- Department id
- Course id
- Section id
- Prerequisite id

Step 4

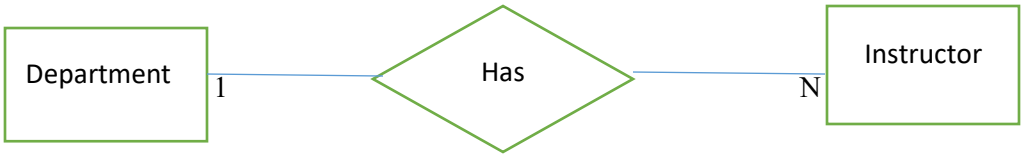
Once we have listed the entities and their attributes, now we must identify the relationships between entities.

- Department offers multiple courses and each courses belongs to only one department. Hence cardinality between department and course if one to many

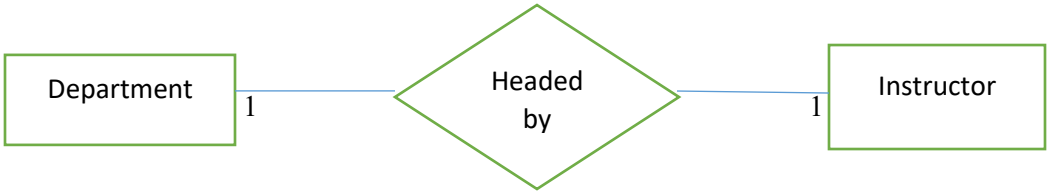
- Course is enrolled by multiple students and one student for multiple courses. Hence, relationships are many to many



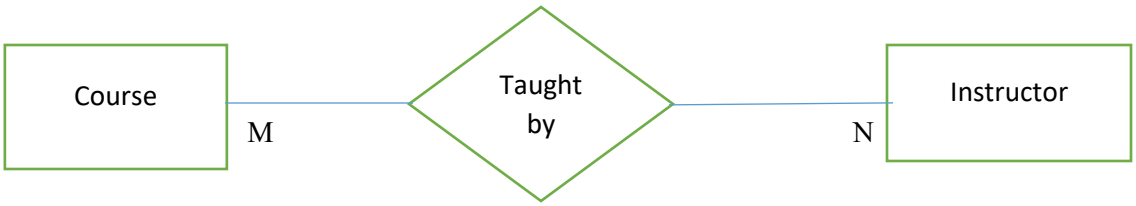
- One department has multiple instructor and one instructor belongs to one and only one department. Hence, the relationship is one to many



- Each department has one HOD and one instructor is HOD for only one department. Hence the relationship is one to one. Here, HOD refers to the head of the department



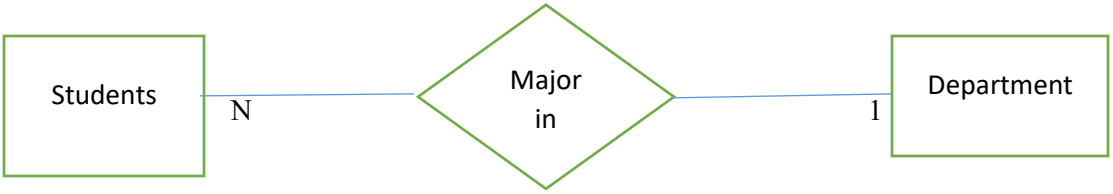
- One course is taught by multiple instructor and one instructor teaches many courses. Hence the relationship between course and instructor is many to many.



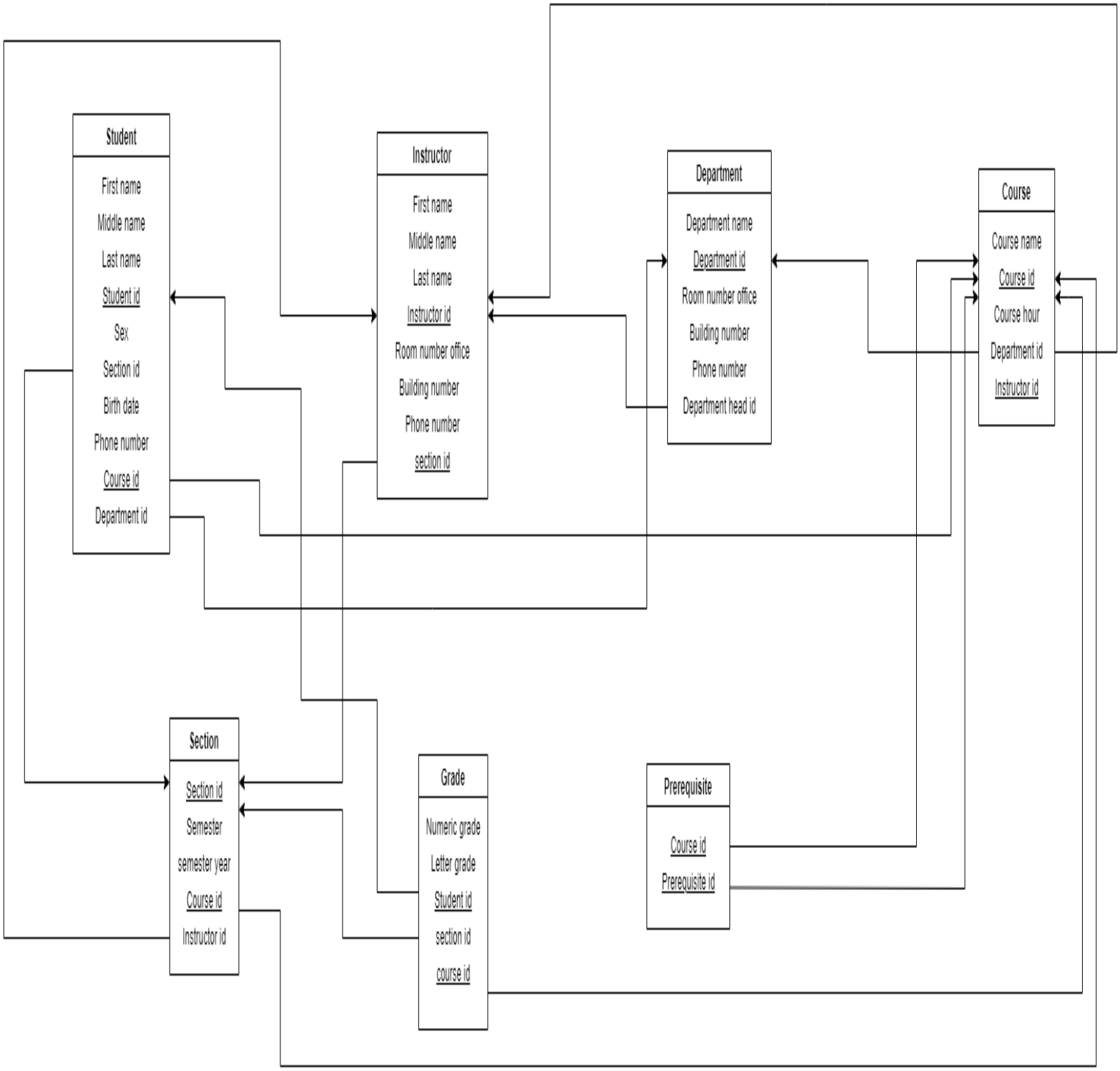
- Each course has multiple prerequisite courses and one course can be prerequisite to many courses. Hence the relationship between course and prerequisite course is many to many.

- Each instructor teaches multiple sections and each section can be taught by multiple instructor. Hence, the relationship is many to many.

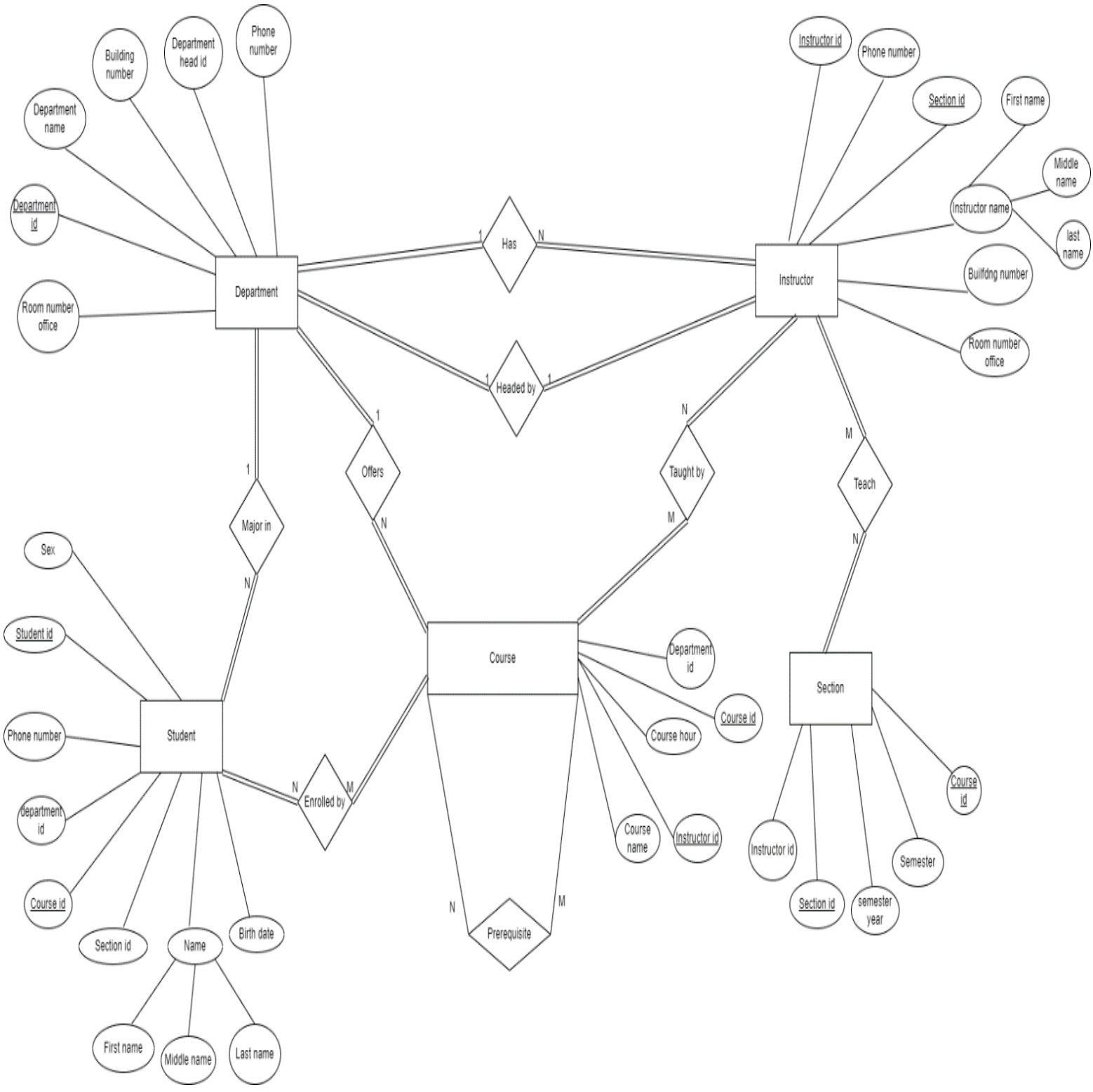
- One Student major in one department but one in department multiple students can major in. Hence the relationship between student and department is many to one.



Schema Diagram for University Database Management System

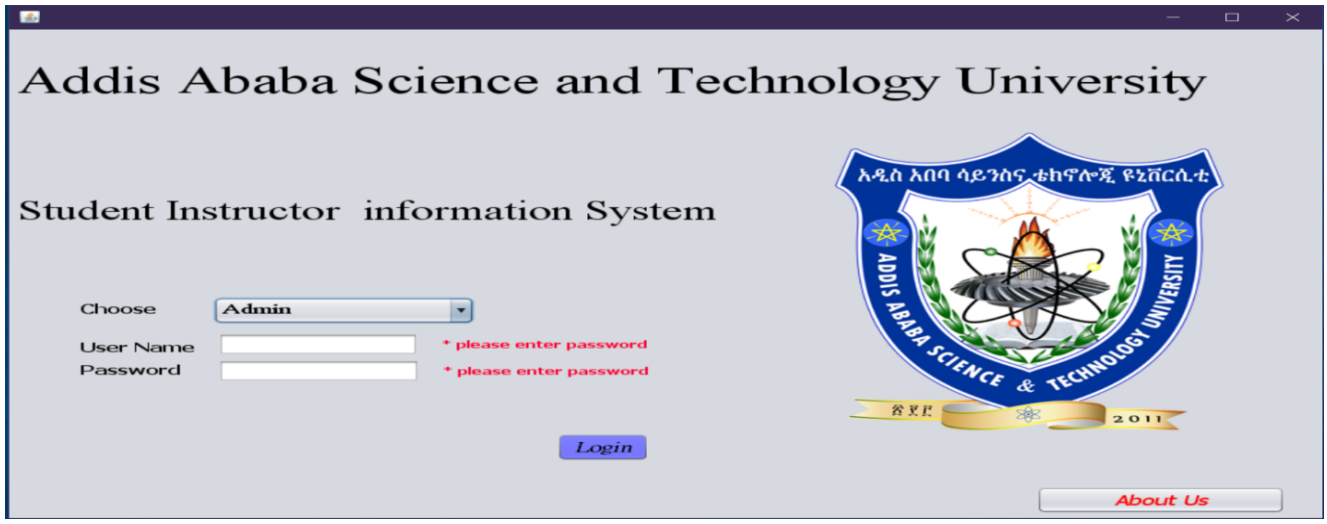


ER Diagram of University Database Management System

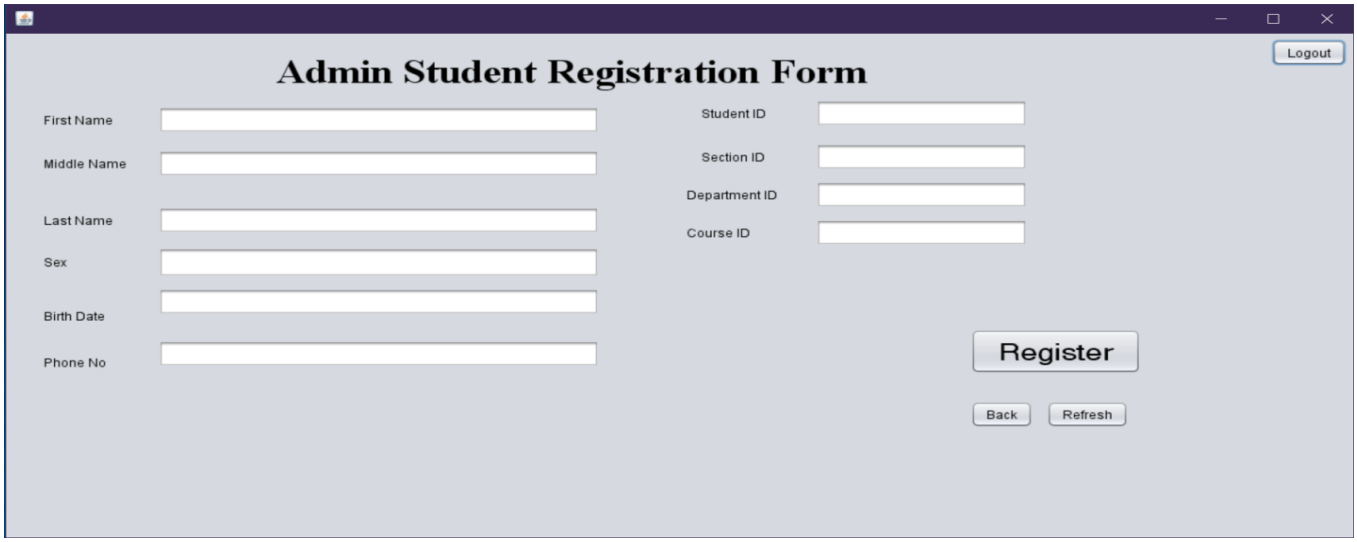


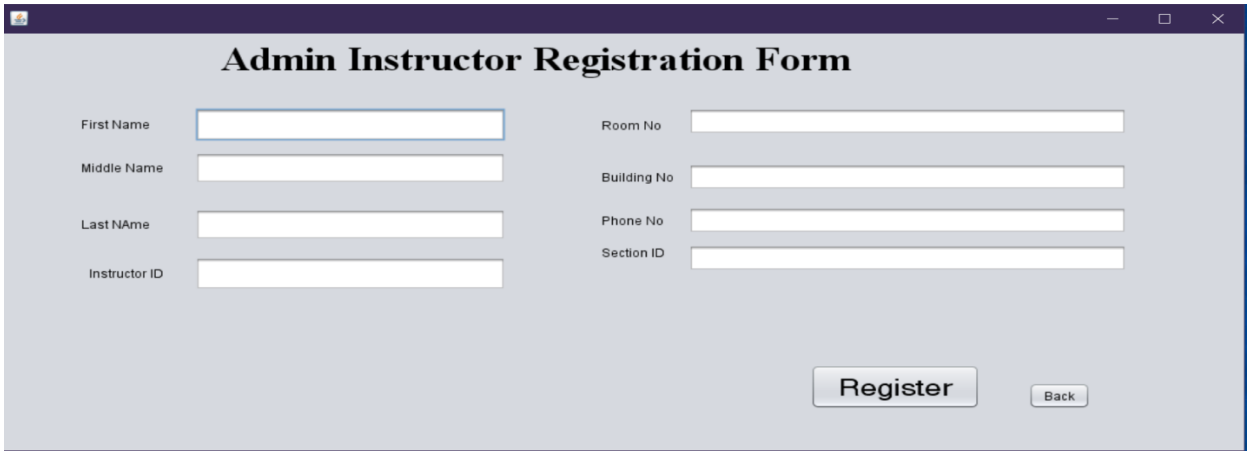
A database application program made by java swing

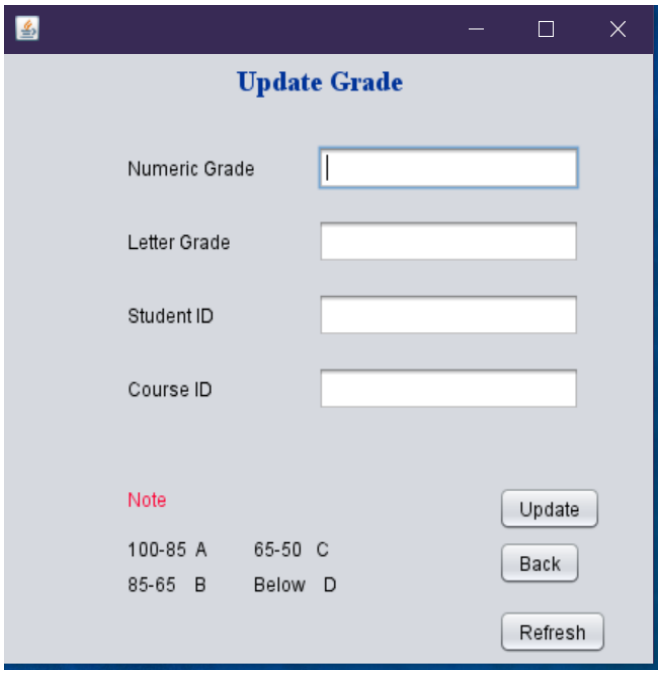
Main Dashboard and Login Page



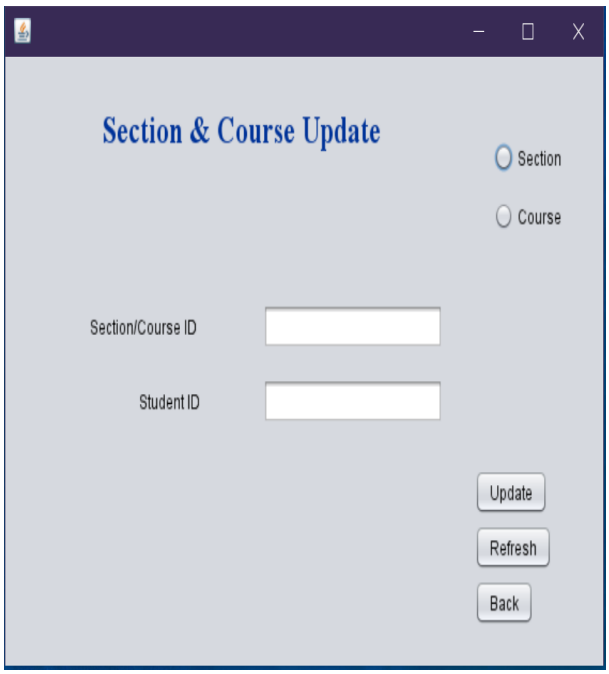
Admin Dashboard and its functional units



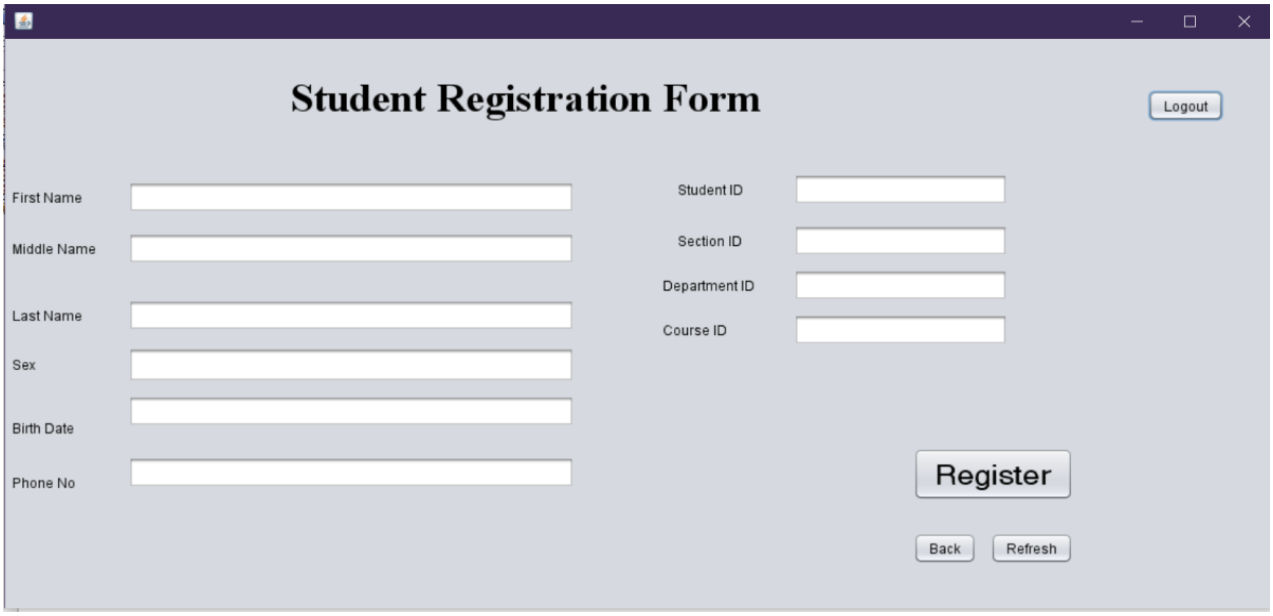


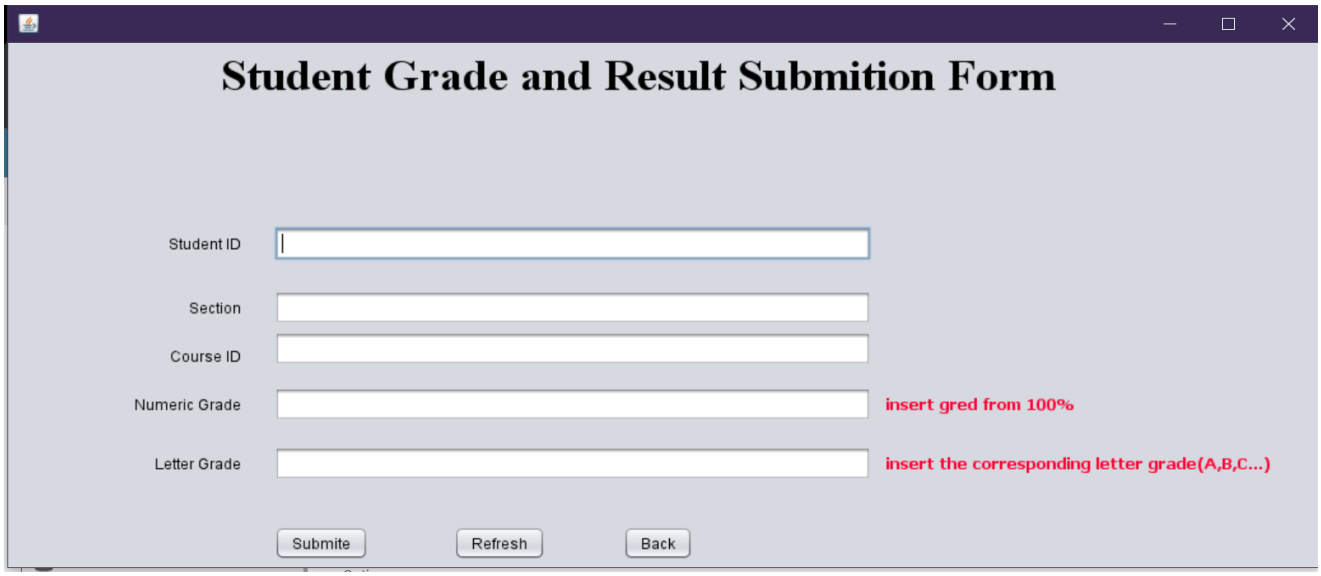


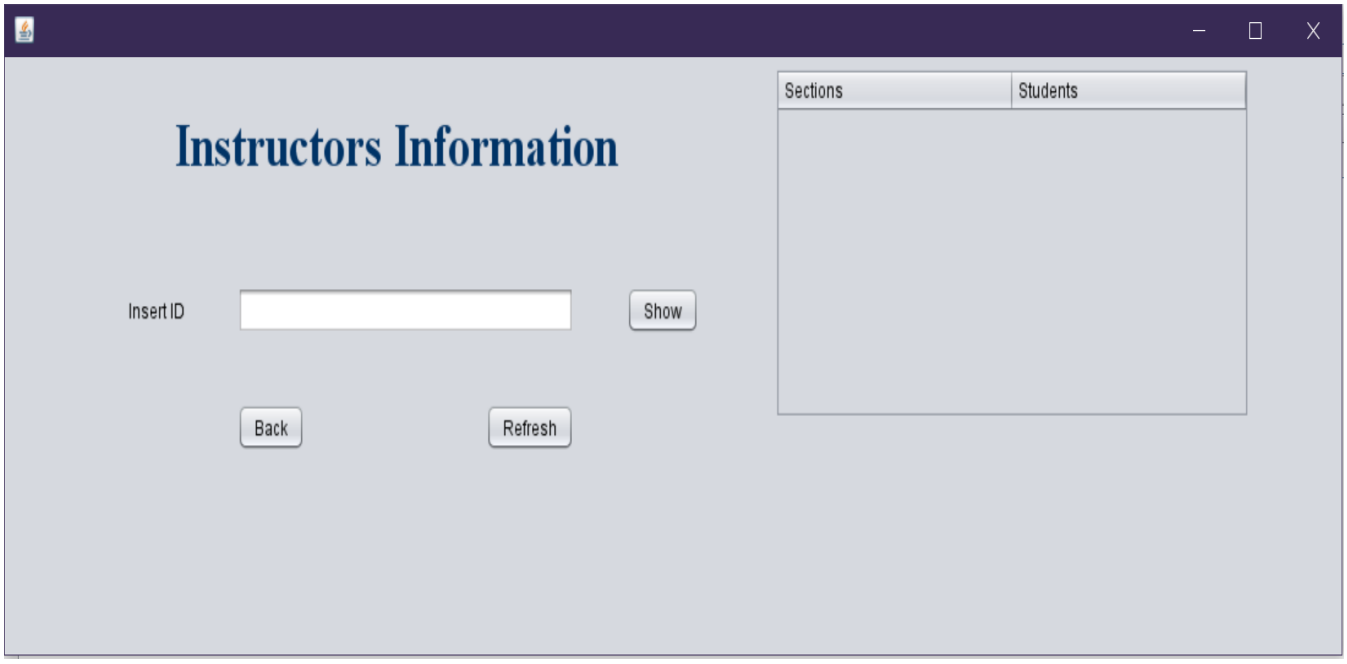
Instructors



Dashboard and its components




Student Grade and Result Submition Form
Student ID
Section
Course ID
Numeric Grade
insert gred from 100%
Letter Grade
insert the corresponding letter grade(A,B,C...)
Submite
Refresh
Back


Instructors Information
Sections
Students
Insert ID
Show
Back
Refresh

Student Dashboard and its components