Submitted By:

Ammar Jamil

01-134231-010

Zameer Hussain

01-134231-096

Zaryab Ahmed

01-134231-097

Course Registration Conflict Resolution System

Bachelor of Science in Computer Science – 4(A)

Submitted to: Mam Fatima Khalique

Department of Computer Science

Bahria University, Islamabad

Table of Contents

3.1 Conceptual Modelling Approach

We have adopted a **centralized approach** for the Course Registration and Conflict Resolution System, which combines the requirements of all user types into a single, unified model. This ensures efficient management of functionalities and simplified the interactions within the system.

3.1.1 Overlapping User Needs

In the system, **different users** (Students, Advisors) share several similar needs:

- Students and Advisors both interact with course and timetable information.
- Course registration and conflict resolution are central to all user roles.
- Shared data like course details and schedules reduces the need for multiple, redundant systems.

3.1.2 Manageable Database Complexity

While the system involves **several entities** (Student, Course, Timetable, Classroom), the relationships and operations are not overly complex. The centralized approach works well here because:

- It avoids unnecessary splitting of the database into smaller modules.
- All core functionalities remain connected in one logical structure.

3.1.3 Single Focus of the System

The system focuses on solving course registration problems, such as timetable clashes. It integrates related functions like:

- Course registration.
- Scheduling classes and sections.
- Allocating classrooms efficiently. This single focus makes the centralized model suitable, as all functionalities contribute to the same goal.

3.1.4 Consistency and Data Integration

Using a centralized approach ensures:

- All user requirements are represented in a single data model.
- Data like schedules, classroom capacities, and registration details are integrated into one structure, avoiding duplication.

3.1.5 Simplified Maintenance

The centralized approach simplifies system updates and maintenance:

- Any changes to course offerings, schedules, or classroom allocations can be managed from a single place.
- The unified structure ensures that updates are reflected across all modules without inconsistency.

By combining overlapping needs, maintaining manageable complexity, and focusing on a single purpose, this centralized approach ensures a smooth, efficient, and user-friendly system.

3.2 ER Modelling

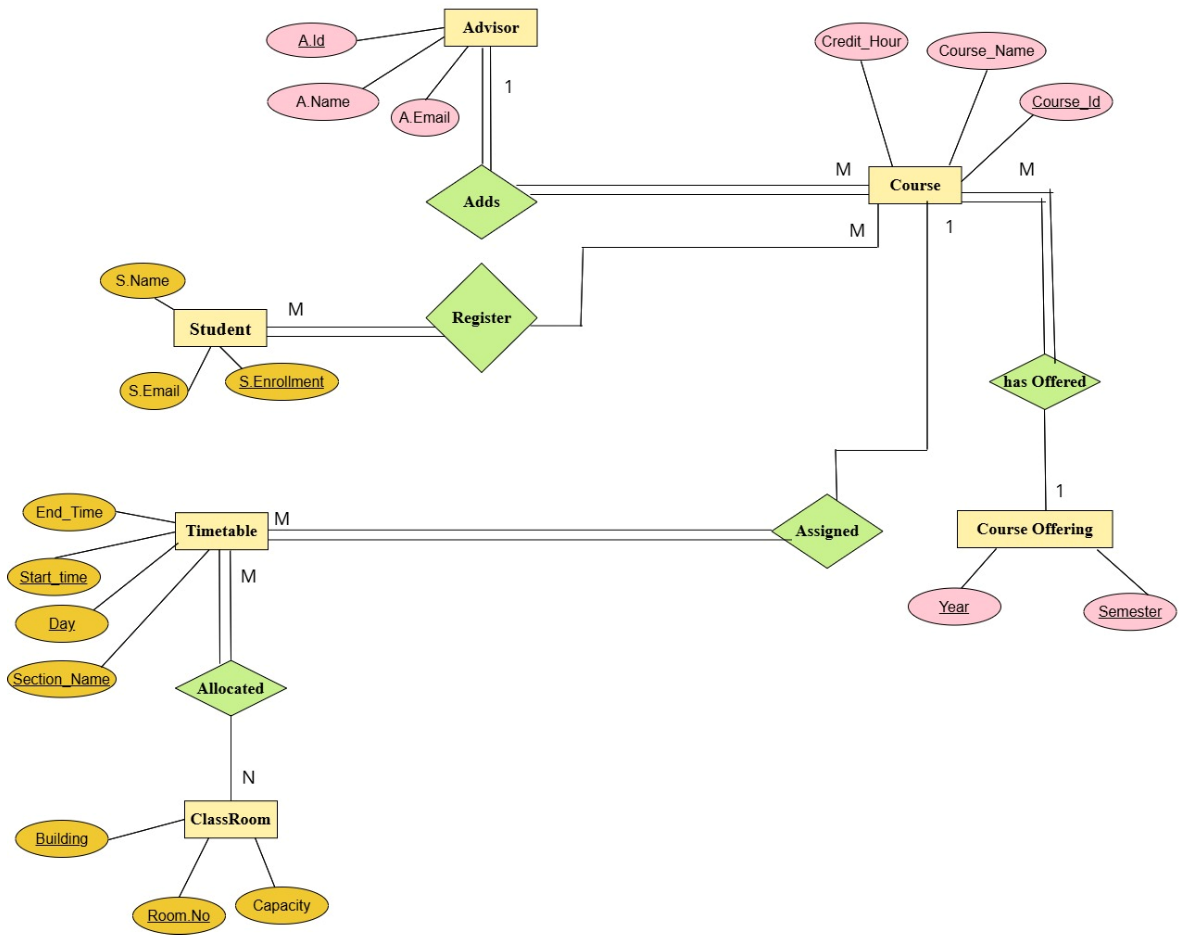
List of Entities and their Attributes

- Advisor:
  Attributes: A.ID, A.Name, A.Email.
- Student:
  Attribute: S.Enrollment, S.Name, S.Email.
- Course-Offering:
  Attribute: Semester, Year.
- Course:
  Attribute: Course.ID, Course-Name, Credit-Hours.
- Timetable:
  Attribute: Day, Start-time, End-time, Section-Name
- Classroom:
  Attribute: Location, Room.No, Capacity.

Relationships

- Adds (Advisor – Course)
- Register (Student – Course)
- Has offered (Course-Course Offering)
- Assigned (TimeTable-Course)
- Allocated (Timetable-Classroom)

ER Diagram



Logical Modelling

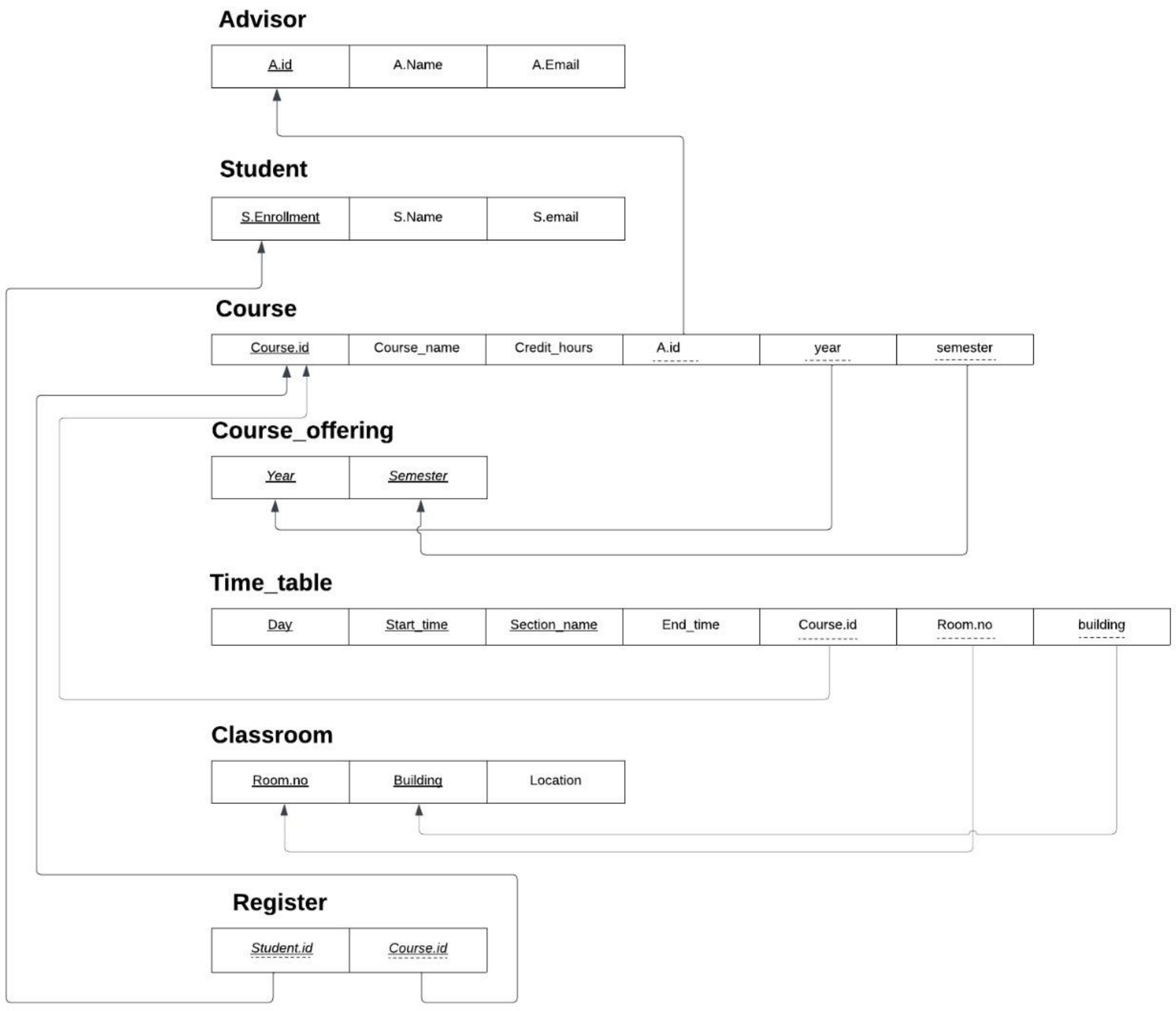
Relational Schema

- Advisor(A.Id, A.Name, A.Email)
- Student(S.Enrollment, S.Name, S.Email)
- Course(Course.Id, Course_Name, Credit_Hours, A.Id, Year, Semester)
- Course Offering(Year, Semester)
- Time Table(Day, Start-time, Section-Name, End-time, Course.Id, Room.No, Building)
- ClassRoom(Room.No, Building, Capacity)
- Register(Student.Id, Course.Id)

NOTE:

- **A single bold straight underline represents a primary key.**
- **A single straight dotted underline represents a foreign key.**
- **A double underline represents a primary and foreign key.**

Block and Arrow Diagram


Advisor
A.id
A.Name
A.Email
Student
S.Enrollment
S.Name
S.email
Course
Course.id
Course_name
Credit_hours
A.id
year
semester
Course_offering
Year
Semester
Time_table
Day
Start_time
Section_name
End_time
Course.id
Room.no
building
Classroom
Room.no
Building
Location
Register
Student.id
Course.id