Wachemo Universty

Online class scheduling System

2025/26

Group Members

Hlkma Mustefa..... WCU170132

Bekelcha Nuredin..... WCU175410

Kamil Ersido.....WCU17D2702

Tsion Aklilu..... WCU175475

Dagmawi Lencho.....WCU174538

SUBMISSION DATE **DECEMBER 6**

TABLE OF CONTENT

CHAPTER TWO: REQUIREMENT ANALYSIS AND SPECIFICATIONS

CHAPTER THREE: DATABASE DESIGN

Chapter One: Introduction

1.1 BACKGROUND OF THE ORGANIZATION

The Software Engineering Department at Wachemo University was established in 2008 E.C. Since then, the department has been working to provide quality education and produce skilled and professional graduates in the field of software engineering. It currently has three lecture classrooms and four laboratory rooms, which support both theoretical and practical learning. The department offers different software related courses each semester for students in various batches.

The current class scheduling system used in the department is manual, and this has created several challenges, such as class overlapping, conflicts, and time consuming schedule preparation. To address these problems, the department needs an online class scheduling system that makes the scheduling process easier, faster, and more accurate.

1.2 STATEMENT OF THE PROBLEM

The existing manual class scheduling system presents several challenges:

- The process is time-consuming for department administrators.
- Scheduling conflicts such as overlapping rooms, instructors, or time slots occur frequently.
- Any updates or timetable changes are not communicated effectively to students and instructors.
- Students rely on notice boards or classmates to know their schedules, which can cause misinformation.
- Keeping records manually increases the chance of errors, inconsistencies, and data loss.

These issues negatively impact academic workflow and reduce the efficiency of the learning process.

1.3 OBJECTIVES OF THE PROJECT

1.3.1 GENERAL OBJECTIVE

To design and develop an Online Class Scheduling System that automates timetable preparation and provides real-time access to schedules for students, instructors, and administrators.

1.3.2 SPECIFIC OBJECTIVES

- To simplify and digitalize the class scheduling process.
- To minimize timetable conflicts and overlapping sessions.
- To allow students and instructors to access updated schedules anytime.
- To enable administrators to update or modify schedules easily.
- To improve efficiency, accuracy, and reliability of academic schedule management.

1.4 METHODOLOGIES

1.4.1 REQUIREMENT ELICITATION METHODOLOGY

To gather system requirements, the following techniques were used:

- Interviews with department heads, instructors, and timetable coordinators.
- Observation of existing scheduling processes and current timetable preparation methods.
- Document review of departmental timetables, course lists, and class allocation records.

1.4.2 REQUIREMENT ANALYSIS AND MODELING

After gathering requirements, the information was analyzed and modeled using:

- Use Case Diagrams
- Activity Diagrams
- Sequence Diagrams
- Class Diagrams
- System Interfaces and Prototypes

1.4.3 SYSTEM IMPLEMENTATION METHODS

The project adopts the Incremental Development Model. The system is developed in multiple phases, where each module is implemented, tested, and integrated step by step. This approach allows early delivery of working features and easier modification during development.

1.5 SCOPE AND LIMITATION OF THE PROJECT

1.5.1 SCOPE

The Online Class Scheduling System covers:

- Admin panel for creating and managing schedules
- Automated conflict detection for rooms, instructors, and times
- Instructor access to personal schedules
- Student access to class schedules by department, batch, or section

1.5.2 LIMITATION

- Internet connection is required for accessing online features.
- Integration with the university's existing student portal is not included in the first version.
- The system will not cover other departments outside the Software Engineering Department.
- The system depends on accurate input data (courses, rooms, instructors).

1.6 SIGNIFICANCE AND BENEFICIARIES OF THE PROJECT

SIGNIFICANCE

- Reduces administrative workload and scheduling errors.
- Saves time and improves the accuracy of timetable preparation.
- Ensures faster communication of schedule changes.
- Enhances accessibility of timetable information for all users.
- Supports the digital transformation goals of Wachemo University.

BENEFICIARIES

- **Administrators:** Simplified timetable creation and editing.
- **Instructors:** Access to clear and updated personal teaching schedules.
- **Students:** Easy access to accurate class schedules anytime.

1.7 FEASIBILITY ANALYSIS

1.7.1 OPERATIONAL/ORGANIZATIONAL FEASIBILITY

The system aligns with the university's goals of improving academic administration and supports smooth adoption by users due to its simplicity and accessibility.

1.7.2 TECHNICAL FEASIBILITY

The system can be developed using commonly available technologies (PHP/Node.js, MySQL, HTML/CSS/JS). It runs on standard university hardware and requires no expensive infrastructure.

1.7.3 ECONOMIC FEASIBILITY

The project relies mainly on open-source tools, minimizing development cost. Implementation cost is low, and long-term benefits such as reduced workload justify the investment.

1.7.4 SCHEDULE FEASIBILITY

The system can be completed within the academic semester using the incremental model, dividing tasks into manageable phases.

1.7.5 LEGAL FEASIBILITY

The system respects user privacy and complies with the university's data handling policies. No legal conflicts are expected since the system stores only academic scheduling data.

1.8 RISK ASSESSMENT

- Risk of unauthorized access → Mitigated by authentication and role-based access.
- Risk of system downtime → Addressed using backups and stable hosting.
- Risk of data entry mistakes → Reduced through validation and conflict-checking features.
- Risk of low user adoption → Minimized through user-friendly design and basic training.

1.9 DEVELOPMENT TOOLS

- Front-end: HTML, CSS, JavaScript, Bootstrap
- Back-end: Node.js
- Database: MongoDB
- Framework (optional): Laravel
- Tools: VS Code, GitHub, XAMPP, Postman

1.10 WORK BREAKDOWN

1.10.1 PROJECT PLAN ACTIVITIES (SCHEDULE)

- Requirement gathering – 3 Days
- Analysis and modeling – 1 week
- System design – 1 week
- Implementation – 2 weeks
- Testing and evaluation – 4 Days
- Documentation – 5 Days

1.10.2 PROJECT ORGANIZATION

Tasks are divided based on modules such as scheduling, conflict detection, user access, and notifications.

1.10.3 TEAM ORGANIZATION

- Project Leader
- System Analyst
- Back-end Developer
- Front-end Developer
- Tester
- Documentation and Presentation Coordinator

1.11 BUDGET PLAN

Item	Estimated Cost (ETB)
Laptop and Software Tools	50,000 ETB
Internet and Hosting	5,000 ETB
Printing and Documentation	2,000 ETB
Miscellaneous	3,000 ETB
Total Estimated Cost	**60,000 ETB**

TABLE 1.1

Chapter Two: Requirement Analysis and Specifications

2.1. DESCRIPTION OF THE CURRENT SYSTEM

Wachemo University currently uses a manual, paper-based approach to prepare class schedules. Department heads collect instructor lists, available rooms, courses, and batch information separately. Timetables are prepared using paper sheets or basic spreadsheets.

This approach causes several problems:

- Schedules often overlap (one instructor assigned to two classes at the same time).
- Rooms are double-booked because there is no automated checking.
- Updating the timetable takes a long time, especially when instructors request changes.
- Students receive the schedule late because it must be posted on boards physically.
- There is no central digital record to track previous schedules or conflicts.

Overall, the current system is slow, error-prone, and lacks automation.

2.2 PLAYERS/ACTORS IN THE EXISTING SYSTEM

- Department Head / Admin
  - Collects course and instructor information
  - Prepares the timetable manually
  - Assigns rooms and class hours
- Instructors
  - Submit their preferred time or availability
  - Use the posted timetable to attend classes
- Students
  - Receive the printed schedule from notice boards

2.3 USE CASE DIAGRAM FOR EXISTING SYSTEM

Since the existing system is manual, the main use cases are:

- Create Schedule
- Assign Instructor and Class
- Update Schedule
- Manage Schedule
- Display Schedule

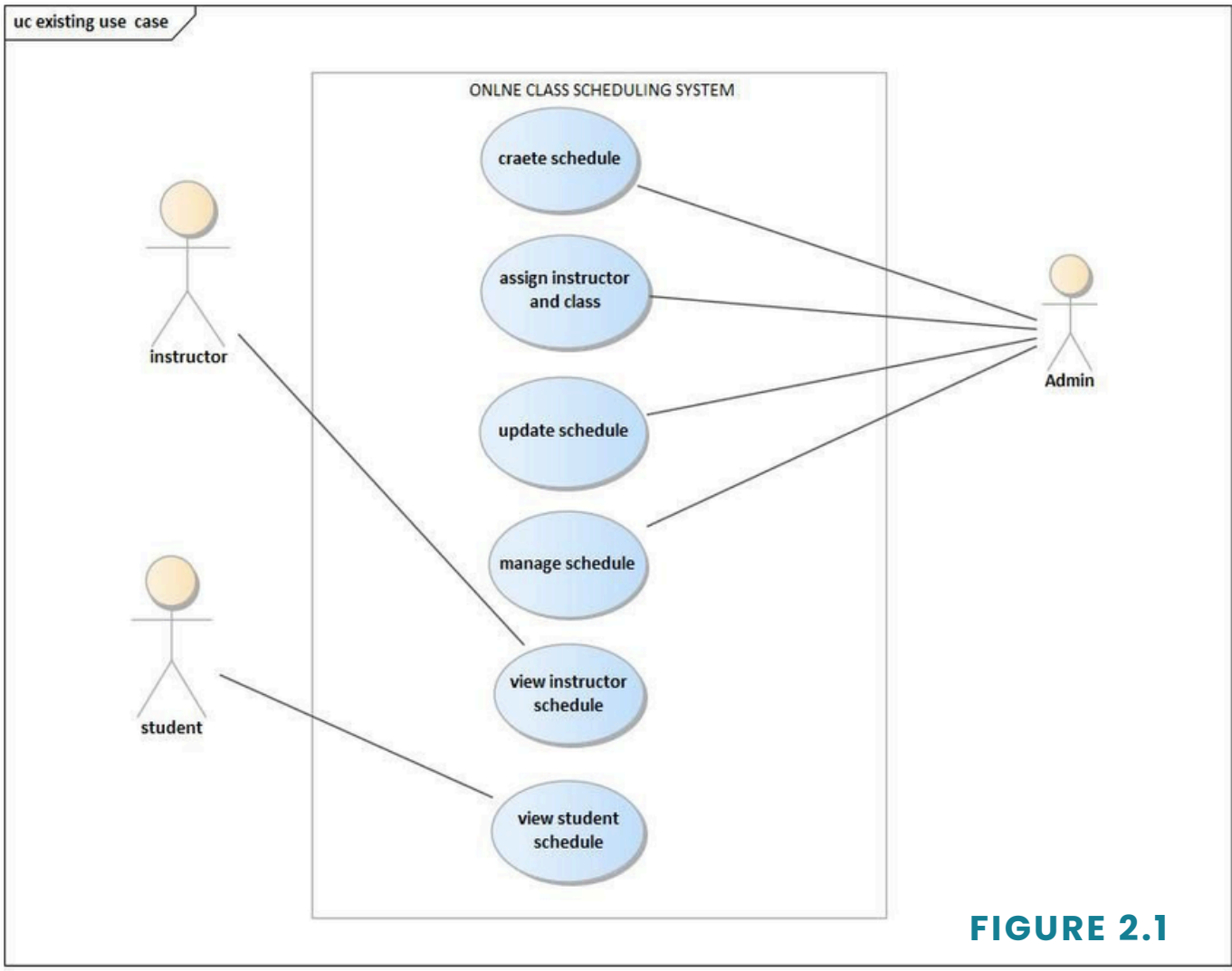


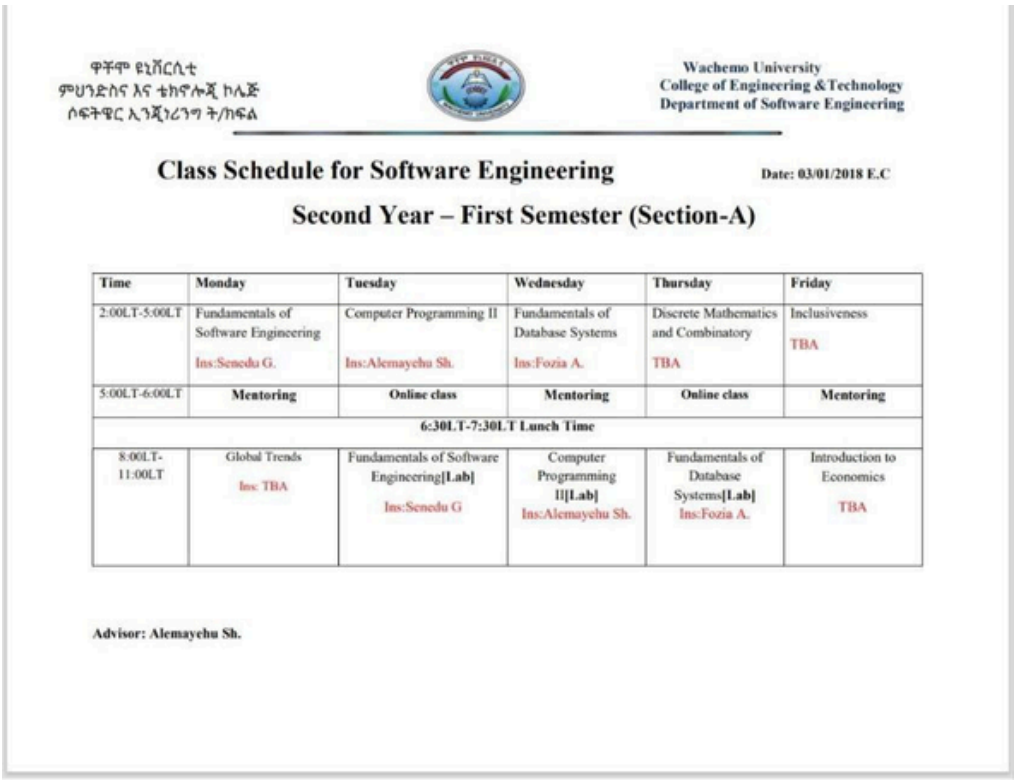
FIGURE 2.1

2.4. FORMS AND DOCUMENTS USED IN THE EXISTING SYSTEM

The current system uses several manual documents:

- Instructor availability form
- Course assignment sheet
- Batch and section registration list
- Printed timetable (final output)
- Room usage list (if available)

These are mostly paper based or simple Word Office.



ዋቾሞ ዩኒቨርሲቲ
ምህንድስና እና ቴክኖሎጂ ኮሌጅ
ሶፍትዌር ኢንጂነሪንግ ት/ክፍል

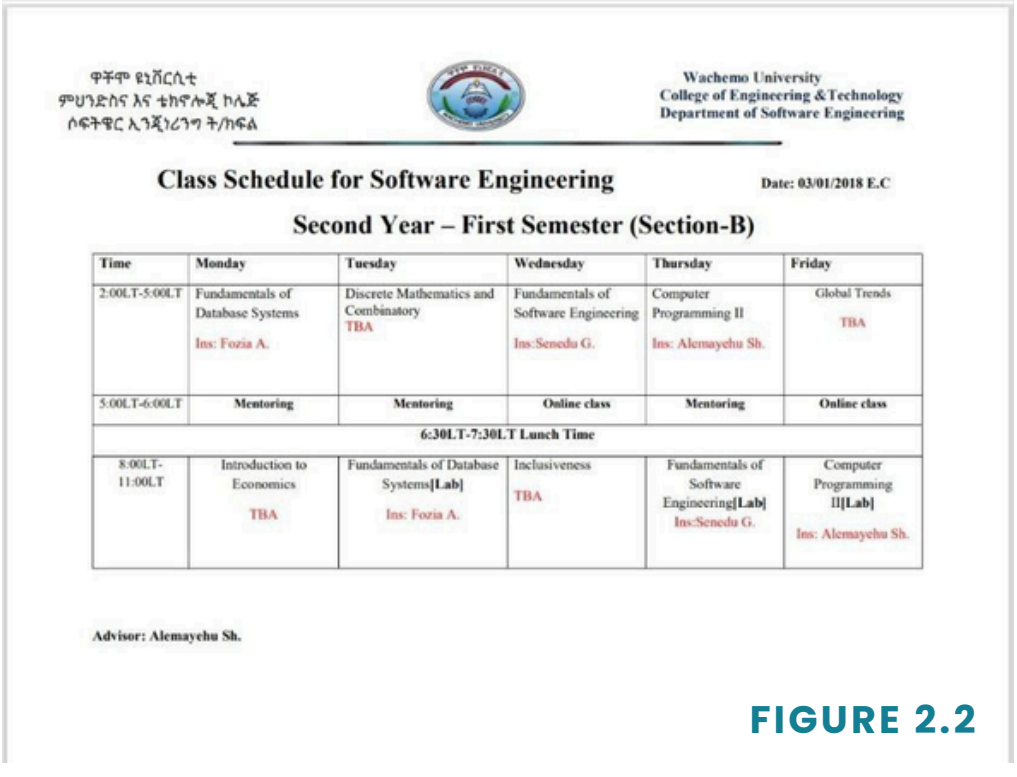
Wachemo University
College of Engineering & Technology
Department of Software Engineering

Class Schedule for Software Engineering Date: 03/01/2018 E.C

Second Year – First Semester (Section-A)

Time	Monday	Tuesday	Wednesday	Thursday	Friday
2:00LT-5:00LT	Fundamentals of Software Engineering Ins:Senedu G.	Computer Programming II Ins:Alemayehu Sh.	Fundamentals of Database Systems Ins:Fozia A.	Discrete Mathematics and Combinatory TBA	Inclusiveness TBA
5:00LT-6:00LT	**Mentoring**	**Online class**	**Mentoring**	**Online class**	**Mentoring**
6:30LT-7:30LT Lunch Time					
8:00LT-11:00LT	Global Trends Ins: TBA	Fundamentals of Software Engineering[Lab] Ins:Senedu G	Computer Programming II[Lab] Ins:Alemayehu Sh.	Fundamentals of Database Systems[Lab] Ins:Fozia A.	Introduction to Economics TBA

Advisor: Alemayehu Sh.



ዋቾሞ ዩኒቨርሲቲ
ምህንድስና እና ቴክኖሎጂ ኮሌጅ
ሶፍትዌር ኢንጂነሪንግ ት/ክፍል

Wachemo University
College of Engineering & Technology
Department of Software Engineering

Class Schedule for Software Engineering Date: 03/01/2018 E.C

Second Year – First Semester (Section-B)

Time	Monday	Tuesday	Wednesday	Thursday	Friday
2:00LT-5:00LT	Fundamentals of Database Systems Ins: Fozia A.	Discrete Mathematics and Combinatory TBA	Fundamentals of Software Engineering Ins:Senedu G.	Computer Programming II Ins: Alemayehu Sh.	Global Trends TBA
5:00LT-6:00LT	**Mentoring**	**Mentoring**	**Online class**	**Mentoring**	**Online class**
6:30LT-7:30LT Lunch Time					
8:00LT-11:00LT	Introduction to Economics TBA	Fundamentals of Database Systems[Lab] Ins: Fozia A.	Inclusiveness TBA	Fundamentals of Software Engineering[Lab] Ins:Senedu G.	Computer Programming II[Lab] Ins: Alemayehu Sh.

Advisor: Alemayehu Sh.

FIGURE 2.2

2.5. BUSINESS RULES OF THE EXISTING SYSTEM

- One course must be assigned to only one instructor.
- No two classes should use the same room at the same time.
- Major courses are usually placed in the morning.
- Common courses are placed in the afternoon.
- Timetable must match the department's academic calendar.
- Conflicts must be manually checked by the department head.

2.6. PROPOSED SYSTEM

2.6.1. OVERVIEW

The proposed system is an Online Class Scheduling System that automates the creation, updating, and management of timetables.

It eliminates manual conflicts, speeds up schedule preparation, and allows instructors and students to view schedules digitally.

The system will intelligently generate schedules based on:

- Instructor availability
- Course type (major/common)
- Room capacity & type
- Batch and section
- Credit hours
- Day and time preferences

2.6.2. BUSINESS RULES OF THE NEW SYSTEM

- The system must not generate overlapping classes.
- Rooms must be assigned based on course requirements (lab, lecture, etc.).
- Credit hours determine weekly class frequency.
- Major courses → morning; Common courses → afternoon.
- Instructors cannot be assigned to two classes at the same time.
- Each batch receives only one course per time slot.
- All schedules must be stored in the system database.

2.6.3. FUNCTIONAL REQUIREMENTS

FR1: Admin Login

- The admin must log into the system to manage scheduling activities.

FR2: Manage Courses

- Admin can add, update, and register course details (course code, name, type, credit hours, year, semester).

FR3: Manage Instructors

- Admin can add instructor profiles and assign courses to them.

FR4: Manage Rooms

- Admin can register classrooms and labs.

FR5: Create Schedule

- System generates timetable based on selected semester, batch, and available data.

FR6: Assign Lecture and Class

- Admin can manually assign a lecturer or override system-generated assignments.

FR7: Update Schedule

- Admin can modify or adjust existing schedules.

FR8: Manage Schedule

- System stores, updates, and maintains all timetable data.

FR9: View Schedule

- Instructors and students can view their schedules.

2.6.4. NON-FUNCTIONAL REQUIREMENTS

Performance Requirements

- Schedule generation must be fast (within a few seconds).
- System must support multiple users at the same time.

Security Requirements

- Role-based access (Admin, Instructor, Student).
- Login authentication required.

Usability Requirements

- Simple and clean user interface.
- Clear timetable display for users.

Reliability Requirements

- System should prevent data loss during updates.
- Timetable correctness must be ensured by conflict detection.

2.6.5. ACTOR AND USE CASE IDENTIFICATION

Actors:

- *Admin* – Manages all scheduling activities.
- *Instructor* – Views schedule.
- *Student* – Views schedule.
- *Database* - Store and Retrive Data
- *Controller* - Detect Overlap

Use Cases:

1. create schedule
2. assigning instructor to the course
3. assign the class
4. manage schedule
5. update schedule
6. detect overlap
7. view assigned class and the instructor assigned to the class
8. view student schedule
9. view timetable and semester
10. view instructor schedule and assigned class
11. store and retrieve the data

2.6.6. SYSTEM MODEL

We can describe our system using different modelling tools like use case diagrams, object models, and dynamic models. These diagrams help us clearly show the problem we want to solve and how the system should behave. Because they are visual, they make the requirements easier to understand compared to long paragraphs of text.

2.6.6.1. USE CASE MODEL

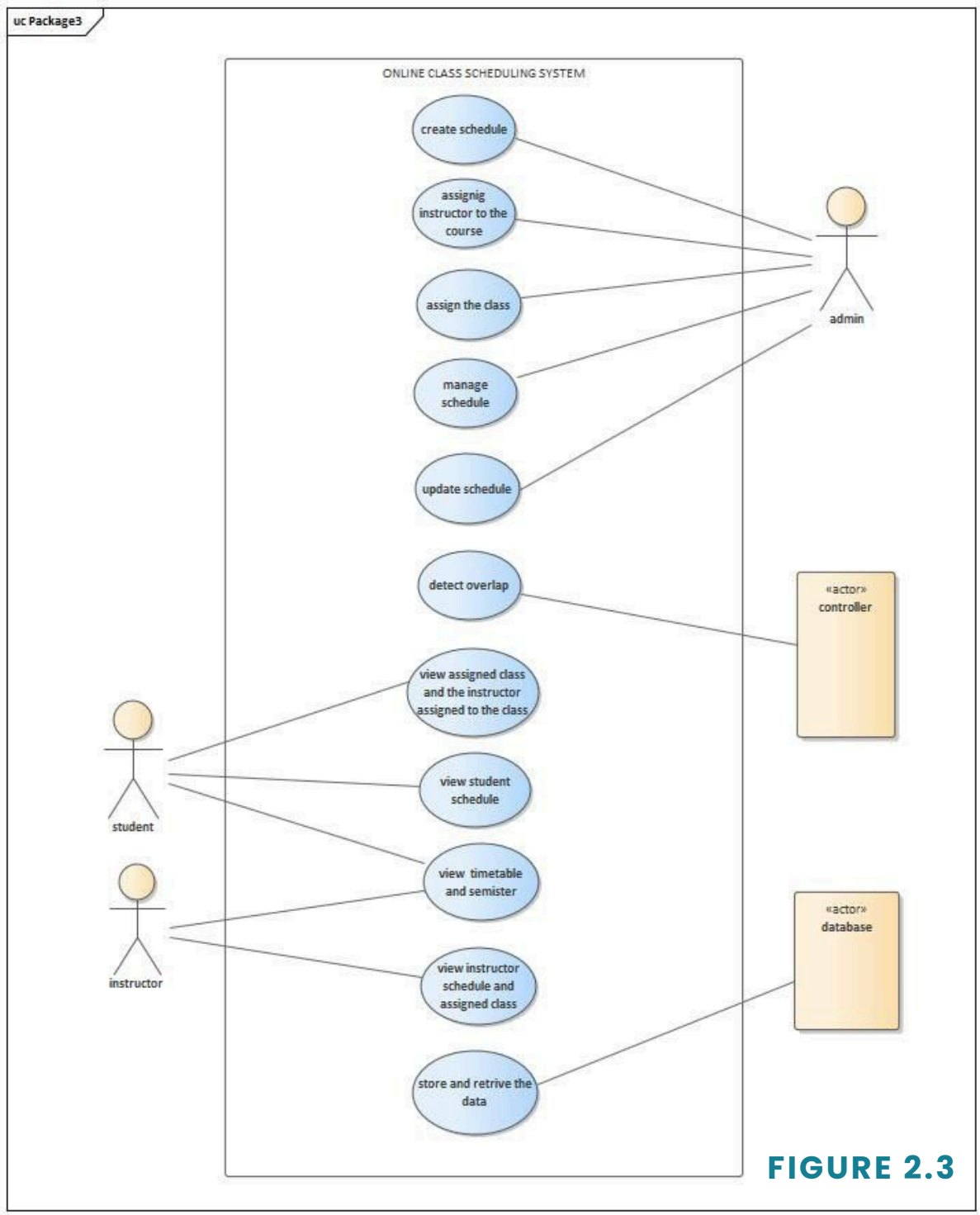


FIGURE 2.3

2.6.6.1.1. USE CASE DESCRIPTIONS

The Online Class Scheduling System involves three main user actors: the admin, the student, and the instructor, along with two external system actors: the controller and the database. The admin is responsible for all scheduling operations. These include creating schedules, assigning instructors to specific courses, and assigning classes such as rooms, sections, and times. The admin also manages the entire schedule by updating existing schedules and ensuring that no conflicts occur. To support this, the system provides a detect overlap function that checks for conflicting class times or instructor availability.

Students interact with the system mainly to view information. They can view their assigned classes, the instructors teaching those classes, their full schedule, and the semester timetable. Instructors use the system to view the classes assigned to them as well as their full weekly schedule. Behind the scenes, the controller actor processes all scheduling-related logic, such as verifying assignments, detecting overlaps, and managing updates. The database actor stores all data related to schedules, instructors, classes, and student information, and retrieves it whenever needed by any use case. Together, these interactions ensure that the system provides an organized, conflict-free, and easily accessible class scheduling environment for all users.

SAMPLE FOR USECASES

Use Case id	UC-01	
Use Case Name	**Create schedule**	
Actor	Admin	
Description	The admin creates a new class schedule by entering course details, selecting time slots, and setting the instructor or room if required	
Goal	To successfully create and save a class schedule in the system	
Precondition	The admin must be logged into the system	
Basic Flow of Event	**Actor Action**	**System Response**
	Step 1: Admin enters username and password	System verifies login credentials
	Step 3: Admin selects "Create Schedule" option	System opens the schedule creation form
	Step 5: Admin enters schedule details (course, time, instructor, semester, etc.)	System validates the entered details
	Step 7: Admin submits the sched-ule	System saves the schedule and confirms creation
Post Condition	The system stores and displays the newly created schedule	

TABLE 2.1

Use Case id	UC-02	
Use Case Name	**Assign instructor to course**	
Actor	Admin	
Description	The admin assigns an instructor to a course by selecting the instructor and the course	
Goal	To successfully assign an instructor to a course	
Precondition	The admin must be logged into the system	
Basic Flow of Event	**Actor Action**	**System Response**
	Step 1: Admin selects 'Assign Instructor' option	System displays an instactor course
	Step 2: Admin selects the instructor	System displays available courses
	Step 3: Admin selects the course	System validates selection
	Step 4: Admin submits the assignment	System confirms the assignment
	Step 5: Admin submits the	System conirms the assignment
Post Condition	The system updates and displays the course with the assigned instructor	

TABLE 2.2

Use Case id	UC-03	
Use Case Name	**Assign class**	
Actor	Admin	
Description	The admin assigns a class to a course by selecting the class and the course	
Goal	To successfully assign an class to a course	
Precondition	The admin must be logged into the system	
Basic Flow of Event	**Actor Action**	**System Response**
	Step 1: Admin selects 'Assign Class' option	System displays an instructor form
	Step 2: Admin selects the class	System displays available courses
	Step 3: Admin selects the course	System validates selection
	Step 4: Admin submits the assignment	System confirms the assignment
	Step 5: Admin submits	System confirms the assignment
Post Condition	The system updates and displays the course with the assigned class	

TABLE 2.3

2.6.6.2. SEQUENCE DIAGRAM

1. CREATE SCHEDULE

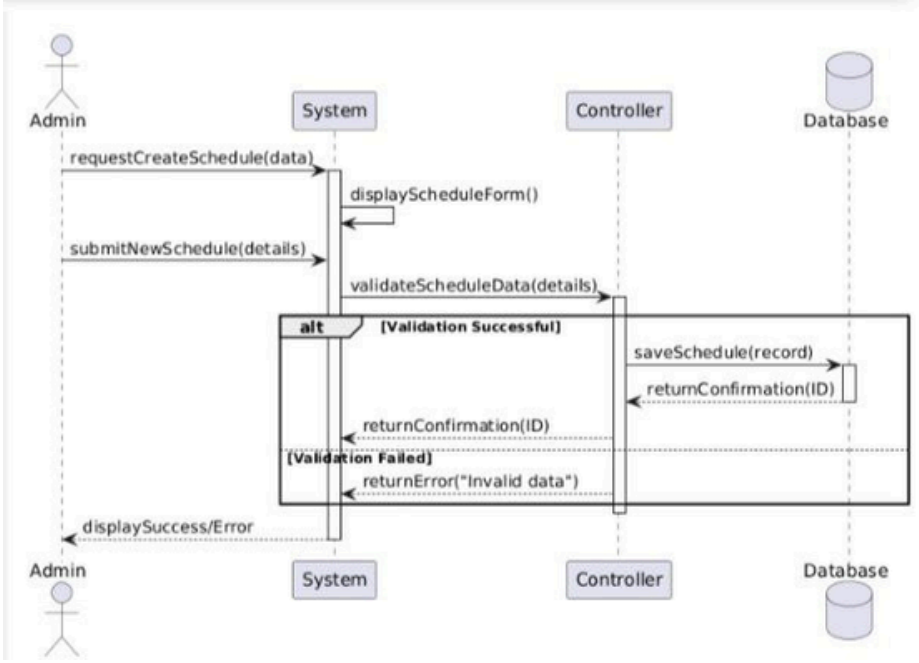


FIGURE 2.4

2. ASSIGN LECTURER AND CLASS

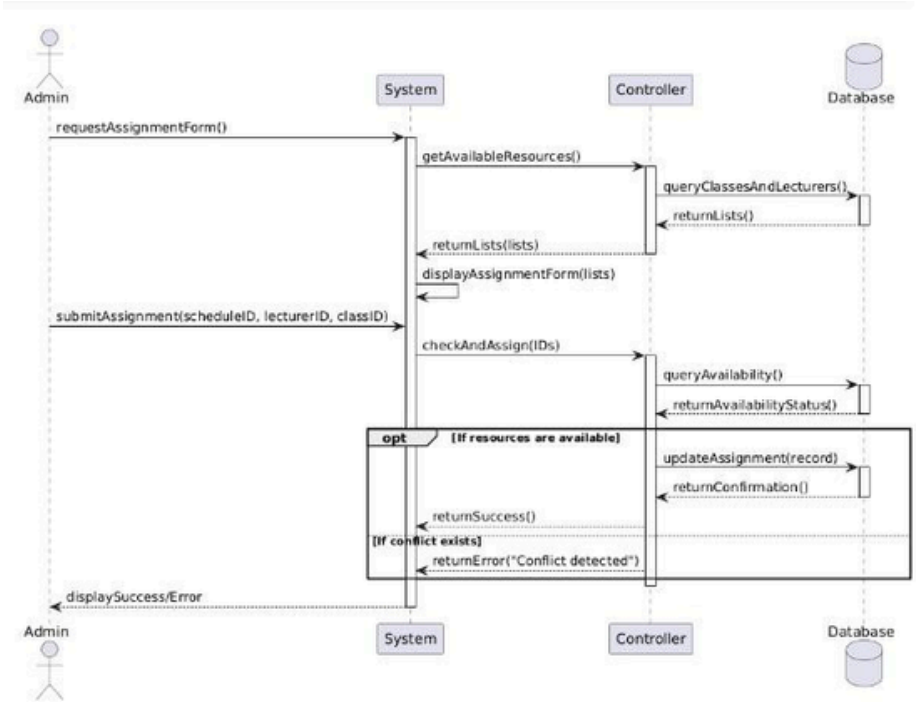


FIGURE 2.5

3. UPDATE AND SCHEDULE

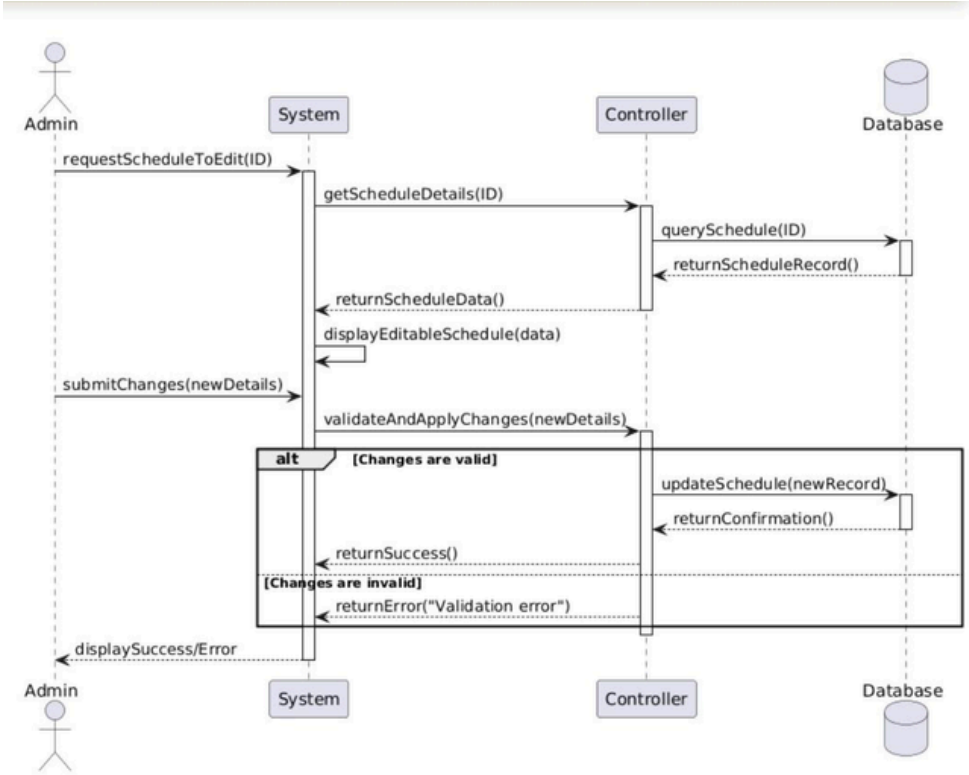


FIGURE 2.6

4. VIEW SCHEDULE(INSTRUCTOR)

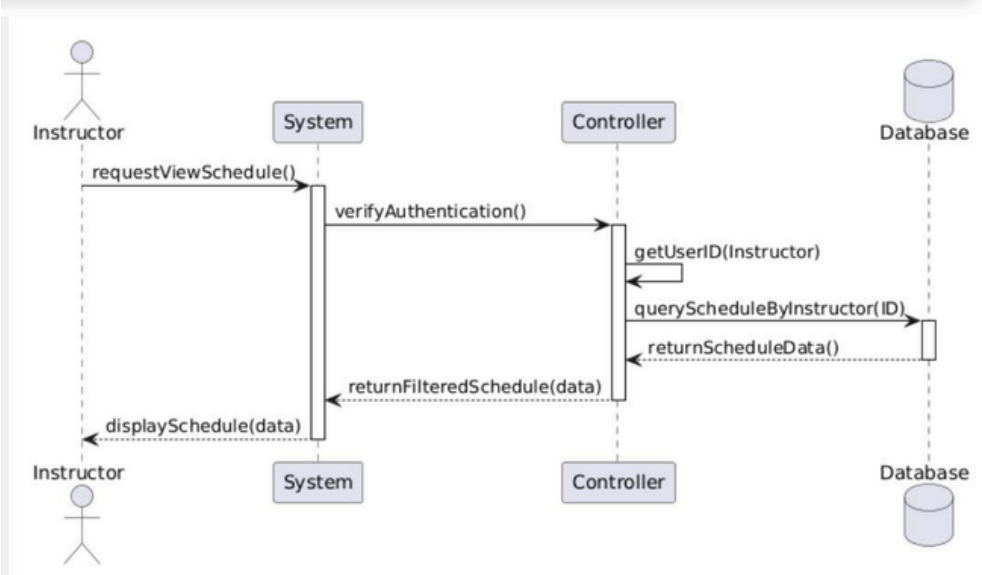


FIGURE 2.7

5. VIEW SCHEDULE(STUDENT)

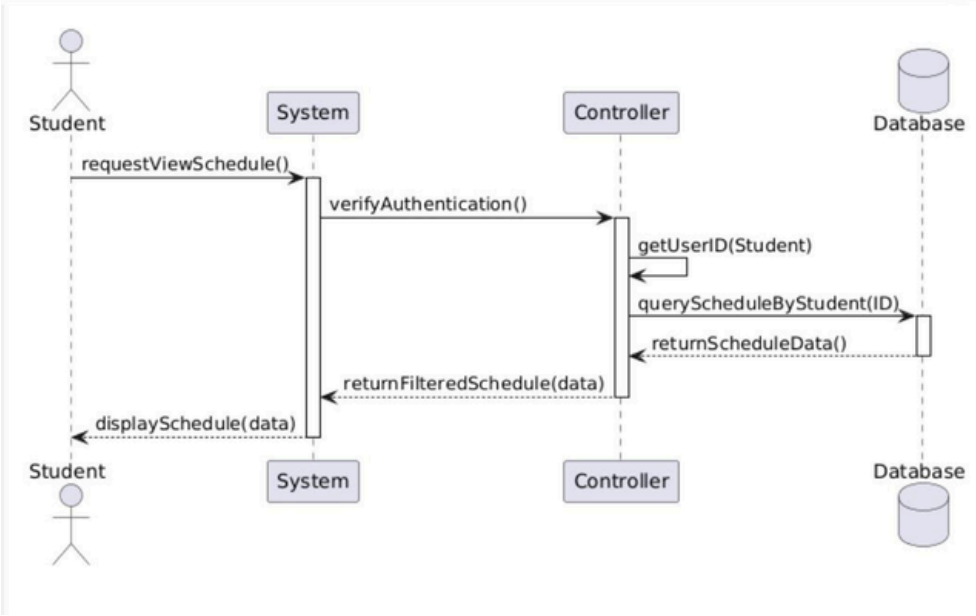


FIGURE 2.8

2.6.6.3. ACTIVITY DIAGRAM

1 VIEW SCHEDULE (STUDENT)

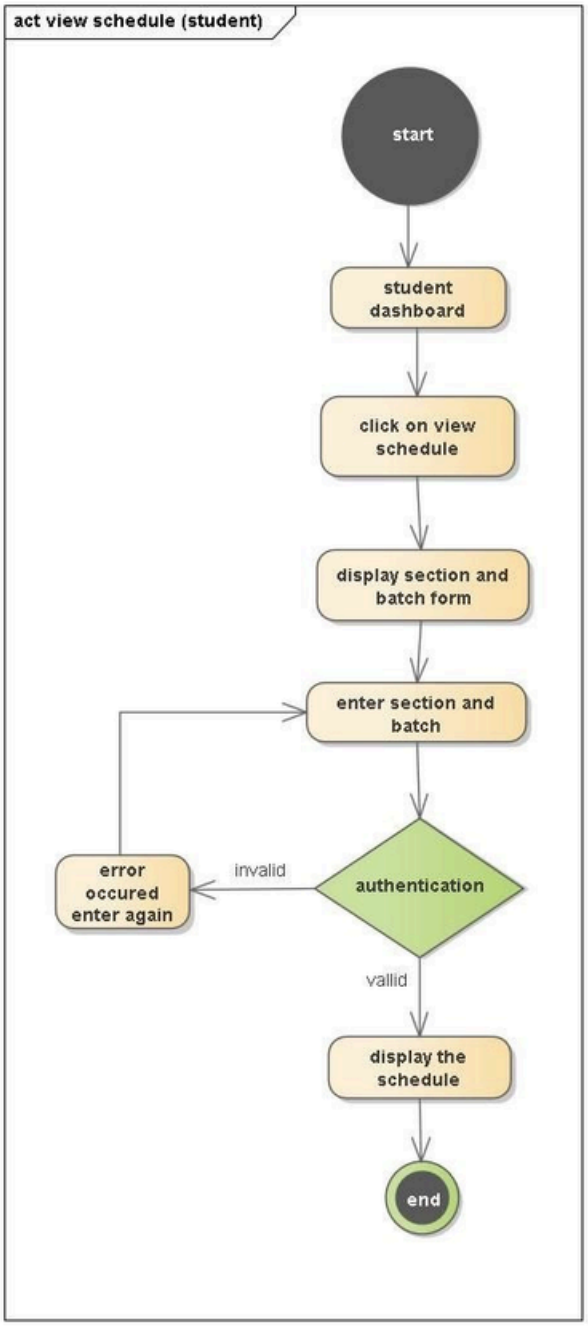


FIGURE 2.9

2 CREATE SCHEDULE

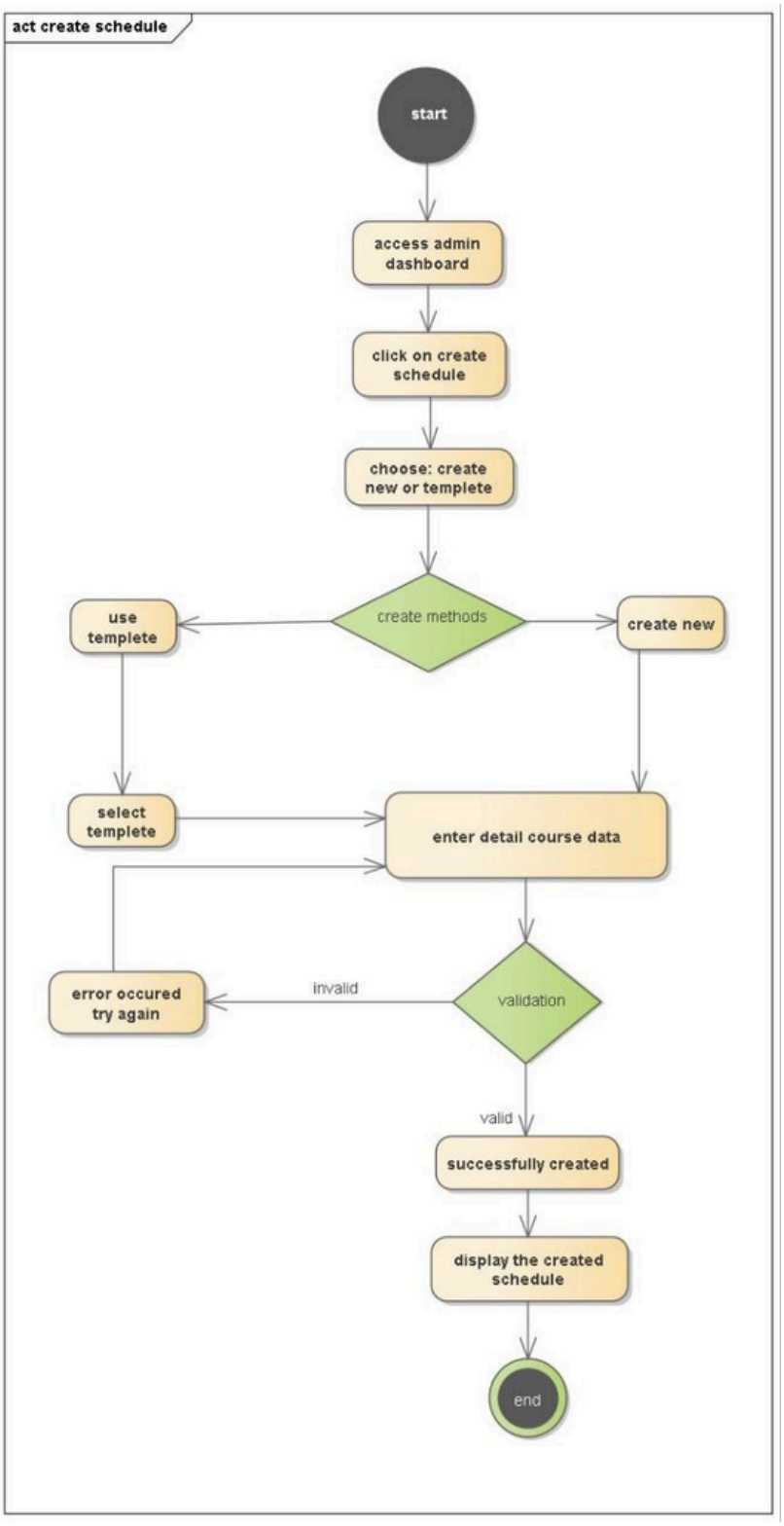


FIGURE 2.10

3 INSTRUCTOR ASSIGNING

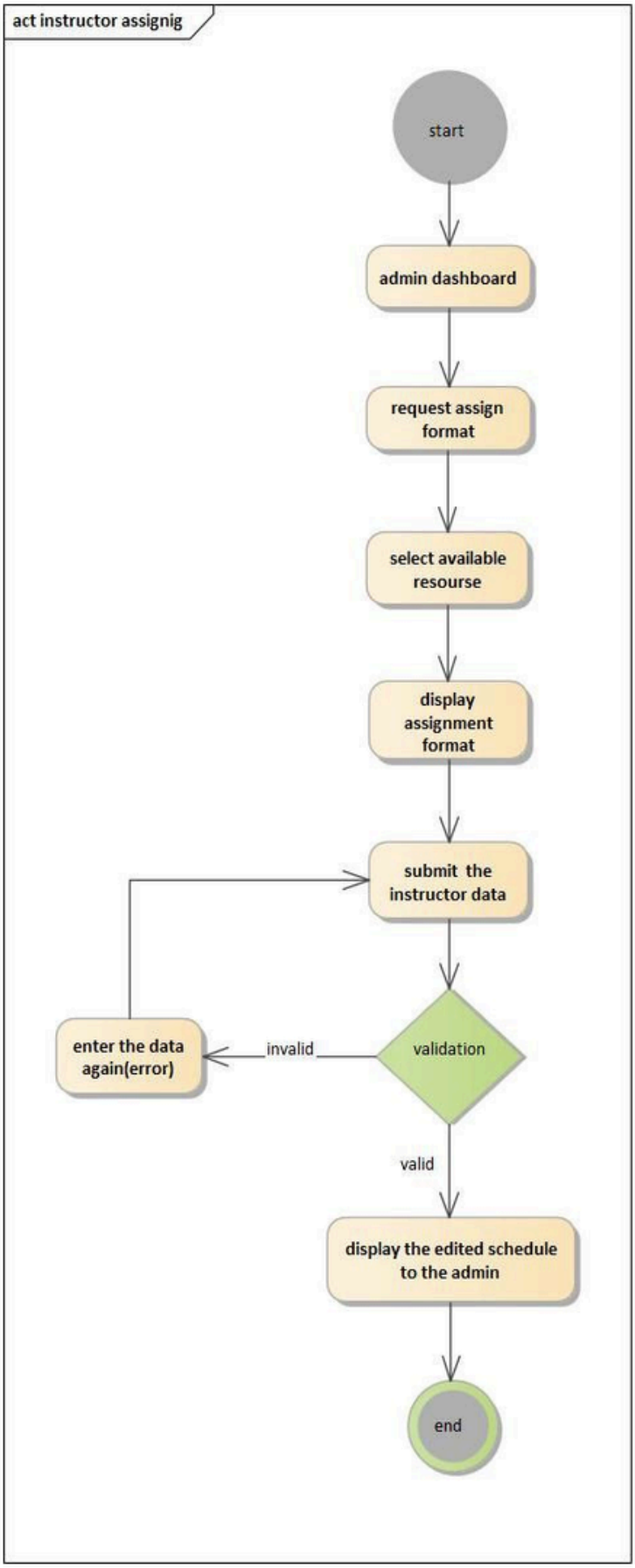


FIGURE 2.11

4 UPDATE SCHEDULE

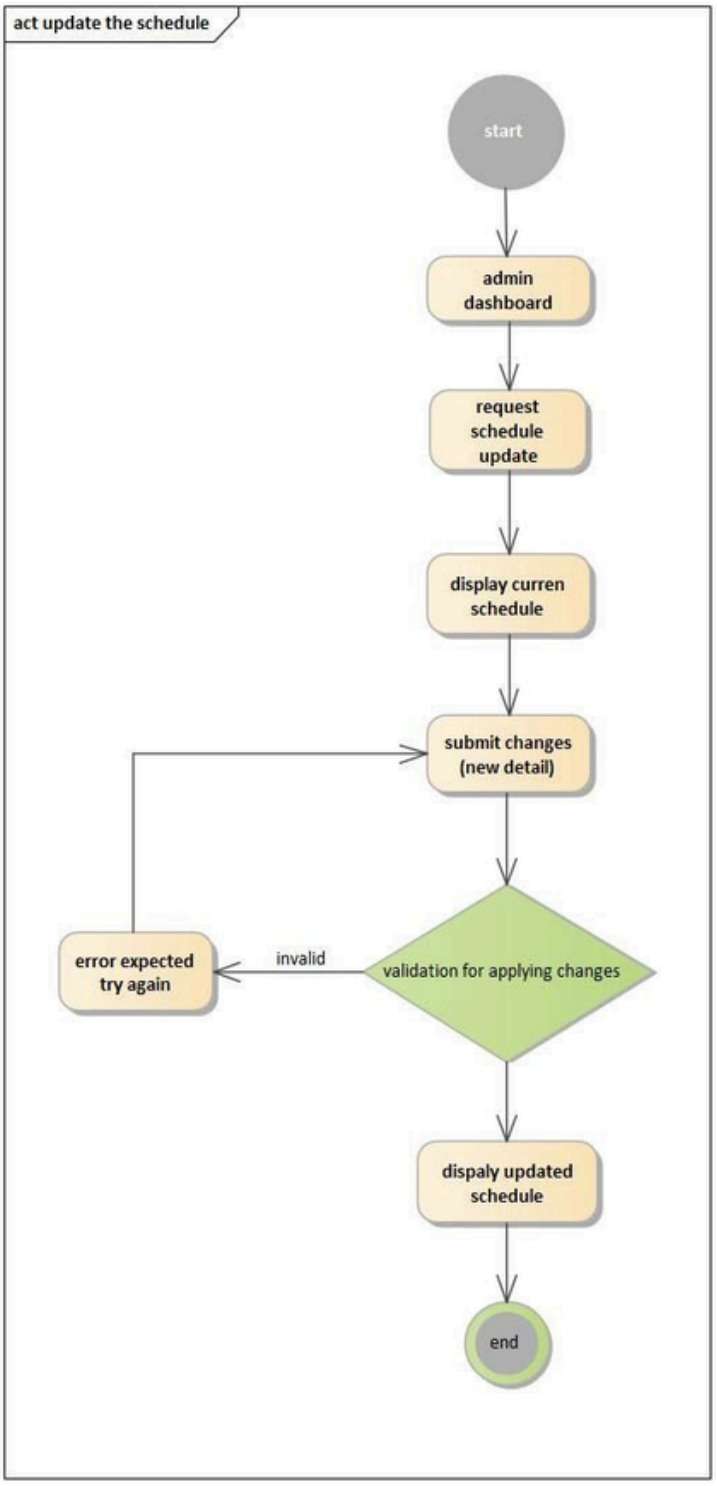


FIGURE 2.12

5 VIEW SCHEDULE (INSTRUCTOR)

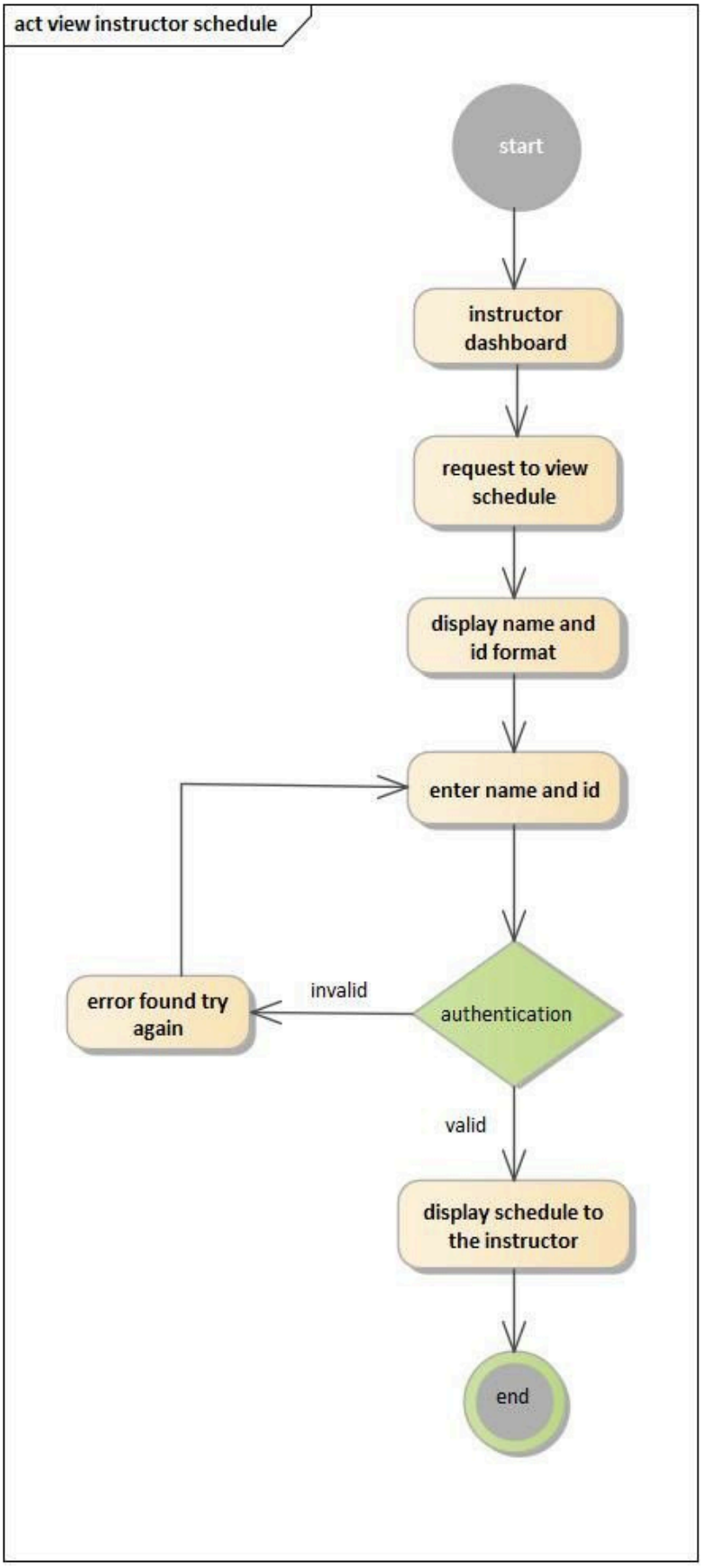


FIGURE 2.13

2.6.6.5. CLASS DIAGRAM

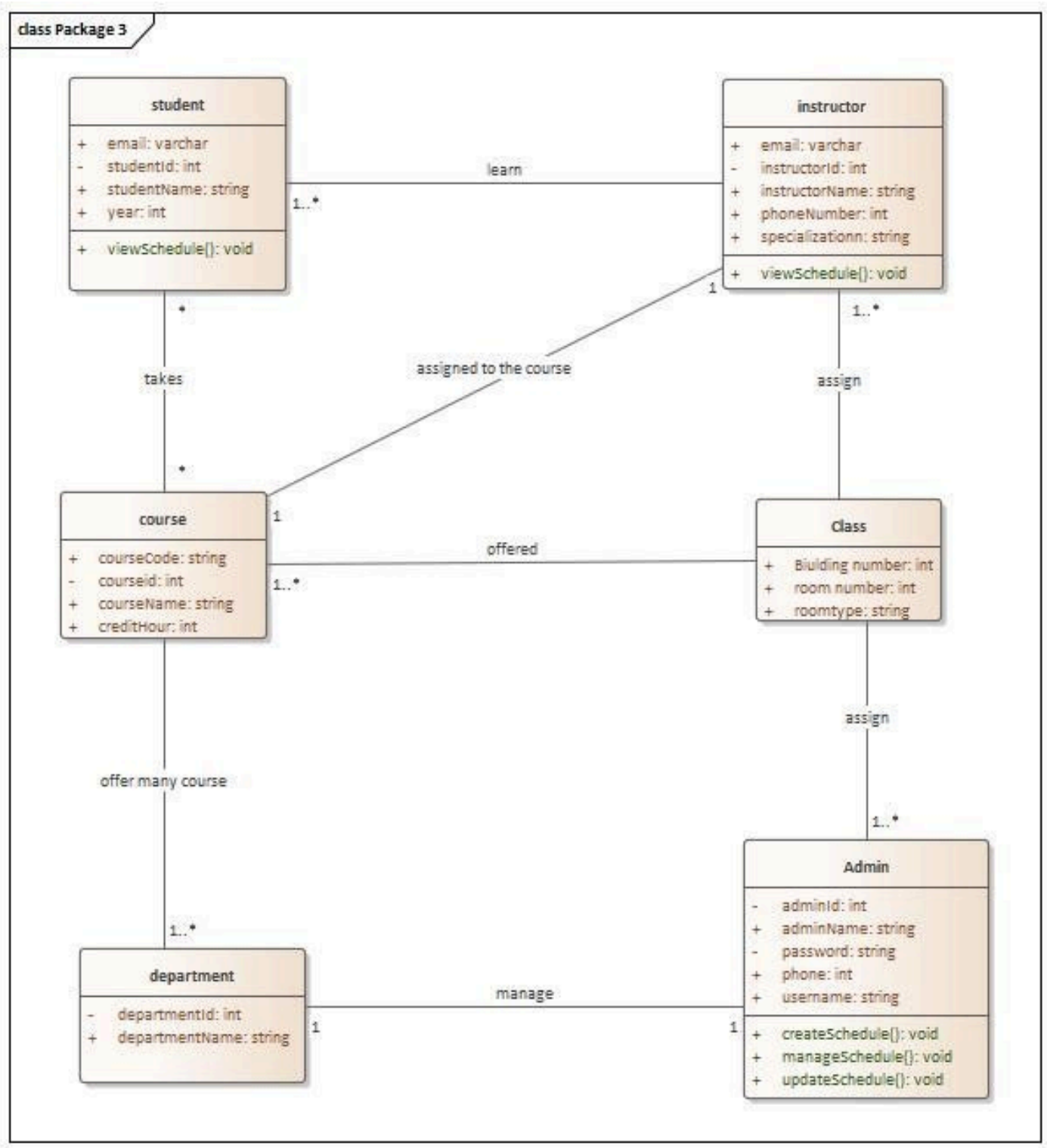


FIGURE 2.14

2.6.6.6. USER INTERFACE PROTOTYPING

FIGURE 2.15

Chapter Three: Database Design

3.1 CONCEPTUAL DATABASE DESIGN

The conceptual design identifies the essential entities required by the system and describes how they interact. The goal is to capture all information needed to manage instructors, students, courses, classes, and departments in a structured way. This design helps ensure the database supports all scheduling functions efficiently.

3.1.1 ENTITIES IDENTIFICATION AND DESCRIPTION

1. ADMIN

Represents the system administrator responsible for managing the entire scheduling process. The admin controls course registration, instructor assignment, and class creation.

2. INSTRUCTOR

Stores information about instructors who teach courses. Each instructor is associated with a specific department.

3. COURSE

Represents the academic courses offered by the university. Courses are assigned to instructors and delivered to class groups.

4. STUDENT

Contains information about students enrolled in different departments and class groups.

5. CLASS

Represents a group of students (e.g., Year 2 Section B). The class participates in multiple courses during a semester.

6. DEPARTMENT

Represents academic departments within the university. Each department manages its instructors, classes, and course offerings.

3.1.2 ATTRIBUTES IDENTIFICATION AND DESCRIPTION

ADMIN

- admin_id
- name
  - Fname
  - Lname
- username
- password
- phone

INSTRUCTOR

- instructor_id
- name
  - Fname
  - Lname
- phone
- email
- specialization

COURSE

- course_id
- course_name
- course_code
- credit_hour
- semester

STUDENT

- student_id
- name
  - Fname
  - Lname
- Email
- Year

CLASS

- Building Number
- Room Type
- Room Number

DEPARTMENT

- department_id
- department_name

3.1.3 RELATIONSHIP IDENTIFICATION AND DESCRIPTION

ADMIN – DEPARTMENT

- One admin manages multiple departments.
- Relationship: 1-to-many

DEPARTMENT – INSTRUCTOR

- A department has many instructors.
- An instructor belongs to one department.
- Relationship: 1-to-many

DEPARTMENT – COURSE

- A department offers many courses.
- Each course belongs to one department.
- Relationship: 1-to-many

DEPARTMENT – CLASS

- A department manages multiple classes.
- Each class belongs to one department.
- Relationship: 1-to-many

CLASS – STUDENT

- A class contains many students.
- Each student belongs to one class.
- Relationship: 1-to-many

INSTRUCTOR – COURSE

- One instructor may teach several courses.
- A course is usually assigned to one instructor.
- Relationship: 1-to-many

CLASS – COURSE

- A class can take many courses.
- Each course can be taken by multiple classes.
- Relationship: many-to-many

3.1.4 ER DIAGRAM