CSE327: Software Engineering [Fall 2025]

Term Project: Class Schedule Management System

Course: Software Engineering
Project Type: Group
Submission Format: Two Milestones (SRS + Design, then Final Report)
Goal: Apply requirements engineering, system modeling, documentation, and software design processes

1. Project Overview

Universities require structured, conflict-free class schedules each semester.
Scheduling must consider:

- Predefined timeslot patterns (e.g., ST1, MW2, RA3)
- Instructor availability
- Room availability
- Lab vs theory room requirements
- Course sections and capacity

This project requires students to analyze, document, and design a **Class Schedule Management System** that assists administrators in managing academic schedules under these constraints.

Students may not implement the full system, but must produce professional-level documentation:

- A complete **Software Requirements Specification (SRS)**
- System **modeling diagrams**
- A final **system design & analysis report**
- A **prototype (complete/semi-complete)**

The emphasis is on *requirements engineering*, *modeling*, *architecture*, *design thinking*, and *documentation quality*.

2. System Description (High-Level)

The system should support the following conceptual capabilities:

- Managing courses, sections, instructors, rooms
- Using **timeslot codes** such as:
    - ST (e.g., Sunday-Tuesday pattern)

  - MW (Monday-Wednesday pattern)
  - RA (alternate-day pattern)
- Each timeslot has a fixed time range (e.g., ST1 = 08:00–09:30)
- Lab courses must be assigned only to lab rooms
- Instructors and rooms cannot be double-booked at the same timeslot
- Administrators can add, edit, view, and review schedules
- Optional: visualization of weekly timetable

Students must interpret and refine detailed requirements through the SRS.

3. Constraints and Assumptions

The project should consider the following:

- A schedule entry consists of **Course + Section + Instructor + Timeslot + Room**
- A class scheduled in a given timeslot pattern repeats on both days of that pattern (e.g., ST = Sunday & Tuesday)
- Rooms have types: **THEORY / LAB**
- Courses have types: **THEORY / LAB**
- Lab courses must be scheduled in lab rooms
- No instructor can handle two courses in the same timeslot
- No room can host two classes in the same timeslot
- The system must support searching and reviewing schedules in multiple ways
- Students must list additional assumptions where needed

4. Required Deliverables

This project has **two submissions**.

Submission 1 (Milestone 1)

SRS + System Design Models

A. Software Requirements Specification (SRS)

The SRS must include (IEEE-style structure recommended):

1. **Introduction**

- Purpose
- Scope
- Definitions
- Overview

2. **Overall Description**
   - Product perspective
   - Product functions (high-level descriptions only)
   - User characteristics
   - Constraints
   - Assumptions & dependencies
3. **System Requirements**
   - Description of requirements (Functional and Non-Functional)
   - Data descriptions
   - Interface descriptions
   - Timeslot rules and scheduling constraints
   - Room and lab requirements
4. **External Interface Descriptions**
   - UI expectations (conceptual)
   - System interaction points

B. Modeling & Design Diagrams

Students must submit the following:

- **Use Case Diagram**
- **At least 5 Use Case Descriptions**
   - Suggested: Add class, assign instructor, assign room, view schedule, validate schedule
- **Sequence Diagrams** (minimum two major interactions)
- **Activity Diagrams** (minimum two)

- **Class Diagram**
- **High-Level Architecture Diagram**
- **UI Wireframes / Mockups**
    - Must include screens for:
        - Add class
        - View schedule
        - Conflict warning screen

Evaluation for Submission 1

Component	**Weight**
SRS Document	40%
Modeling Diagrams	40%
Completeness & clarity	20%

Submission 2 (Milestone 2)

Final Report + Revised Documentation + Prototype

A. Revised SRS (if needed)

Update based on feedback from Submission 1.

B. Updated Modeling & Design Documents

Incorporate instructor feedback.

C. Prototype Demonstration (Required)

A **simple prototype** must be created:

It must show the workflow of:

- Adding a class schedule
- Time conflict detection
- Lab room validation
- Viewing the schedule

- Searching by instructor/room

Full software implementation is NOT required but will be highly appreciated.

D. Testing & Validation

Students must submit:

- Test plan
- At least 5 test cases
- Expected vs actual results from prototype
- Validation scenarios (e.g., double-booking, lab assignment mistake)

E. Final Report

The final report must include:

1. Title page
2. Abstract
3. Introduction
4. Summary of SRS
5. System modeling summary
6. Timeslot and schedule constraint analysis
7. Prototype demonstration
8. Testing summary
9. Challenges
10. Conclusion

Evaluation for Submission 2

Component	**Weight**
Revised SRS & Models	20%
Prototype	30%
Testing & Validation	20%
Report Quality	30%

5. Key Project Features Students Must Capture (Conceptual)

Timeslot Encoding

- Each timeslot code represents both day pattern and time range
- Examples:
  - ST1 = 08:00–09:30
  - MW2 = 09:40–11:10
  - RA3 = 11:20–12:50

Conflict Detection Rules

Students must model and document:

- Instructor-time conflict
- Room-time conflict
- Room type mismatch (LAB in THEORY room → invalid)
- Duplicate class entries
- Timeslot consistency

System Views

Document the conceptual system behavior:

- View schedule by timeslot
- Search by instructor
- Search by room
- Print weekly pattern schedule

6. Optional Enhancements (Bonus)

For extra marks, students may conceptualize (not fully implement):

- Automatic schedule generator
- Visualization of timetable grids
- Instructor availability constraints
- Room capacity modeling
- Notification workflow

- Authentication & user roles

7. Summary of Milestones

Milestone	**Deliverables**	**Due**
Submission 1	SRS + Modeling Diagrams	07 December 2025, Sunday
Submission 2	Final Report + Prototype + Updated Models	27 December 2025, Saturday
Final Presentation	A brief Demonstration + Q/A	28 December 2025, Sunday (Online in the Evening)