INSTITUTION OF TECHNOLOGY OF CAMBODIA

DEPARTMENT: GIC I4-B

Report of Software Engineering

Project 3: Course Enrollment and Classroom Scheduling System

Name of Students	Students ID
1. PANG Lythong	e20220161
2. SOEURY Sreyno	e20220908
3. VICHETH Sokhsedtha	e20221549
4. NGET Darapich	e20221646
5. KEO Chanponlork	e20220660

Lecturer: ROEUN Pacharoth

Academic year 2025-2026

Abstract

The Course Enrollment and Classroom Scheduling System is a web-based application designed to manage academic courses, classroom allocation, and student enrollment in an organized and secure manner. The system supports multiple user roles including administrators, lecturers, and students, each with distinct permissions. Built using Spring Boot and following the MVC architecture, the system integrates authentication, database migration, and role-based access control to ensure scalability, maintainability, and security.

Table Contents

I. Introduction
II. System Overview
III. Project Planning and Methodology
IV. System Architecture
V. UML Design
VI. Security and Authentication
VII. Core Entity Management
VIII. Scheduling and Enrollment Logic
IX. Frontend Design
X. Database Design
XI. API Endpoint Design
XII. System Flow
XIII. Testing and Validation
XIV. Challenges and Solutions
XV. Conclusion and Future Work

I. Introduction

1.1 Background

In educational institutions, managing course offerings, classroom scheduling, and student enrollment manually can lead to inefficiencies, scheduling conflicts, and data inconsistency. A centralized system is required to automate these processes and provide role-based access to users.

1.2 Problem Statement

Traditional course enrollment systems often suffer from:

- Poor access control
- Manual scheduling conflicts
- Lack of integration between courses, classrooms, and users
- Difficulty tracking student enrollment

1.3 Objectives

The objectives of this project are:

- To develop a secure course enrollment system
- To enable administrators to manage courses and classrooms
- To allow lecturers to view assigned courses and students
- To allow students to browse and enroll in courses
- To prevent scheduling conflicts and enrollment overflow

II. System Overview

2.1 System Description

The Course Enrollment and Classroom Scheduling System is a Spring Boot web application that allows:

- Admins to manage system data
- Lecturers to view teaching assignments

- Students to enroll in available courses

2.2 User Roles

Role	Description
ADMIN	Full system control
LECTURER	View assigned courses and students
STUDENT	Browse and enroll in courses

III. Project Planning and Methodology

3.1 Development Methodology

The project follows an iterative and modular development approach:

1. Requirement analysis
2. UML design
3. Database design
4. Backend development
5. Frontend development
6. Integration and testing

3.2 Task Division

Team Member	Responsibility
Vicheth Sokhsedtha	Security & Authentication
Keo Chanponlok	Core Entity CRUD
Nget Darapich	Enrollment & Scheduling Logic
Soeury Sreyno	Frontend&Theamleaf

Pang Lythong	Database & Migration Frontend / Thymeleaf

This division ensures clear ownership and parallel development.

IV. System Architecture

4.1 Architecture Pattern

The system uses Spring Boot MVC architecture, divided into:

- Controller Layer
- Service Layer
- Repository Layer
- Entity Layer
- View Layer (Thymeleaf)

4.2 Package Structure

controller → handles HTTP requests

service → business logic

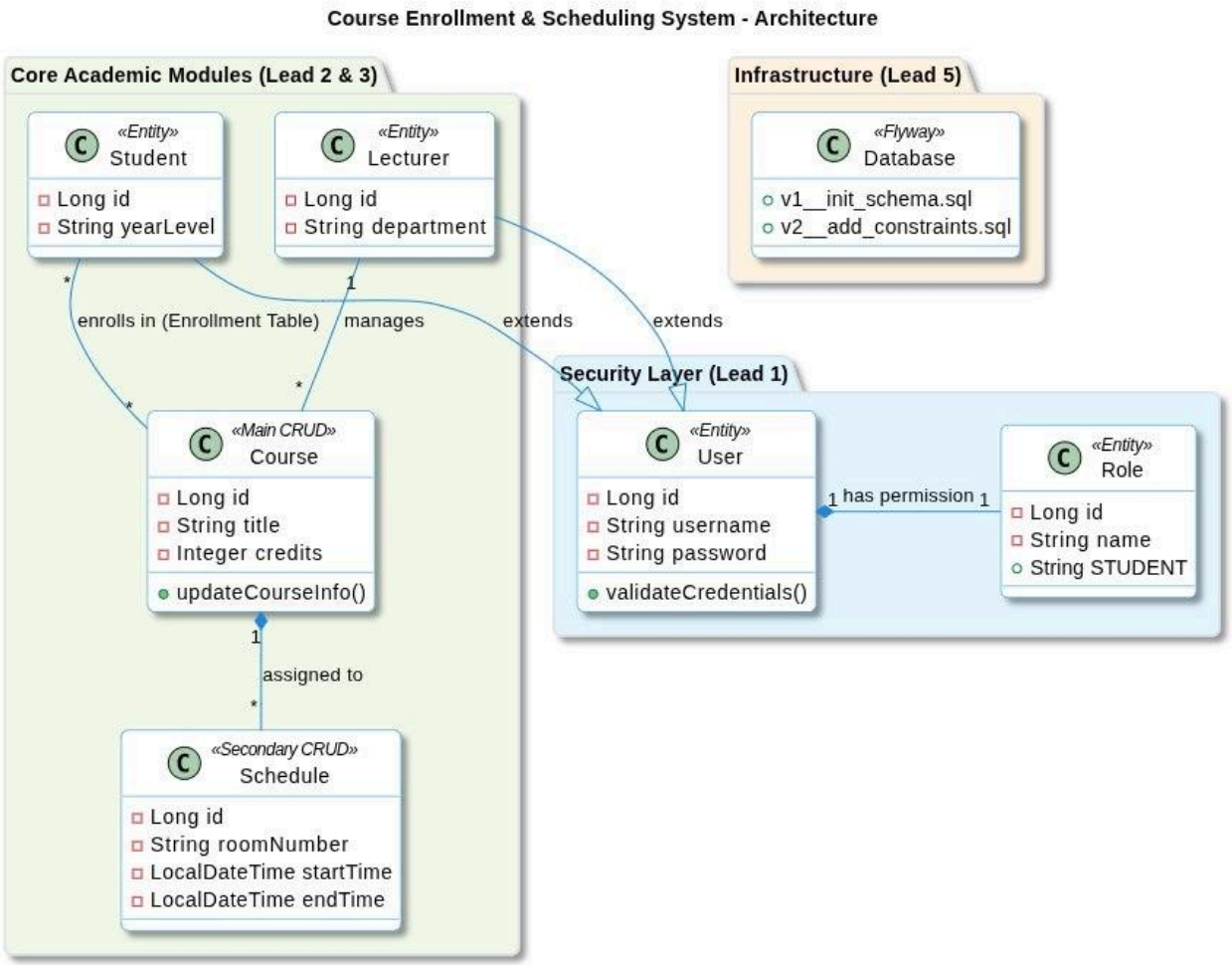
repository → database access

entity → JPA entities

config → security configuration

V. UML Design

5.1 Class Diagram (Description)



Key classes include:

- User
- Role
- Course
- Classroom
- ClassSchedule
- Enrollment

Relationships:

- One User has one Role
- One Course has many ClassSchedules
- Many Students enroll in many Courses via Enrollment

5.2 Use Case Diagram (Description)

Actors:

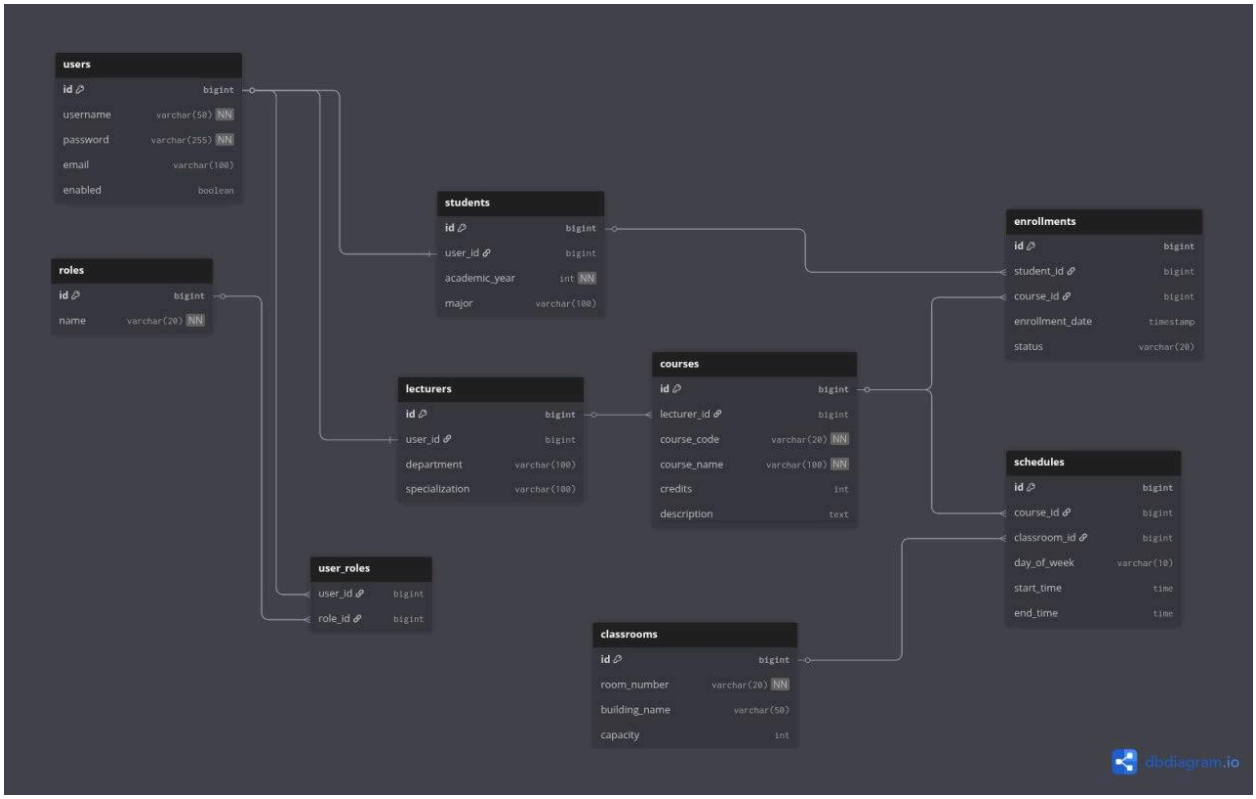
- Admin
- Lecturer
- Student

Use cases:

- Admin manages courses and classrooms
- Lecturer views courses and students
- Student enrolls in courses

5.2 Activity Diagram

5.2 Database



VI. Security and Authentication

6.1 Spring Security Configuration

Security is implemented using Spring Security:

- `SecurityConfig.java` defines public and protected routes
- `/login`, `/register`, and `/css/**` are public
- Role-based access is enforced

6.2 User Authentication

- Custom `UserDetailsService` loads users from the database
- Passwords are encrypted using `BCryptPasswordEncoder`

6.3 Authorization

Role	Permissions
ADMIN	Full CRUD access
LECTURER	View-only access
STUDENT	Enrollment access

Unauthorized access redirects to an Access Denied page.

VII. Core Entity Management

7.1 Course Management

Admins can:

- Create courses
- Edit course details
- View course list

Course fields:

- Course name
- Course code (unique)
- Credits
- Description
- Capacity

7.2 Classroom Management

Admins can manage classrooms with:

- Room number
- Building
- Maximum capacity

VIII. Scheduling and Enrollment Logic

8.1 Class Scheduling

A class schedule links:

- Course
- Classroom
- Day of week
- Time

Validation ensures:

- No double booking
- Classroom capacity is respected

8.2 Enrollment Logic

Students can enroll only if:

- Course is not full
- Student is not already enrolled

8.3 Student Views

Students can view:

- Enrolled courses
- Personal schedules

IX. Frontend Design

9.1 Thymeleaf Templates

Templates are organized by role:

- `admin/`
- `lecturer/`
- `student/`

9.2 Role-Based UI

Thymeleaf `sec:authorize` is used to:

- Hide admin-only buttons
- Display student-only actions

9.3 User Feedback

System provides:

- Success messages
- Error alerts

X. Database Design

10.1 Database Technology

- MySQL (or compatible RDBMS)

- JPA for ORM
- Flyway for migration

10.2 Flyway Migrations

- V1: Create tables
- V2: Insert roles
- V3: Insert sample data
- V4–V6: Admin setup and security

10.3 Relationships

- One Course → Many ClassSchedules
- Many Students ↔ Many Courses via Enrollment

XI. API Endpoint Design

Endpoints are role-based and RESTful.

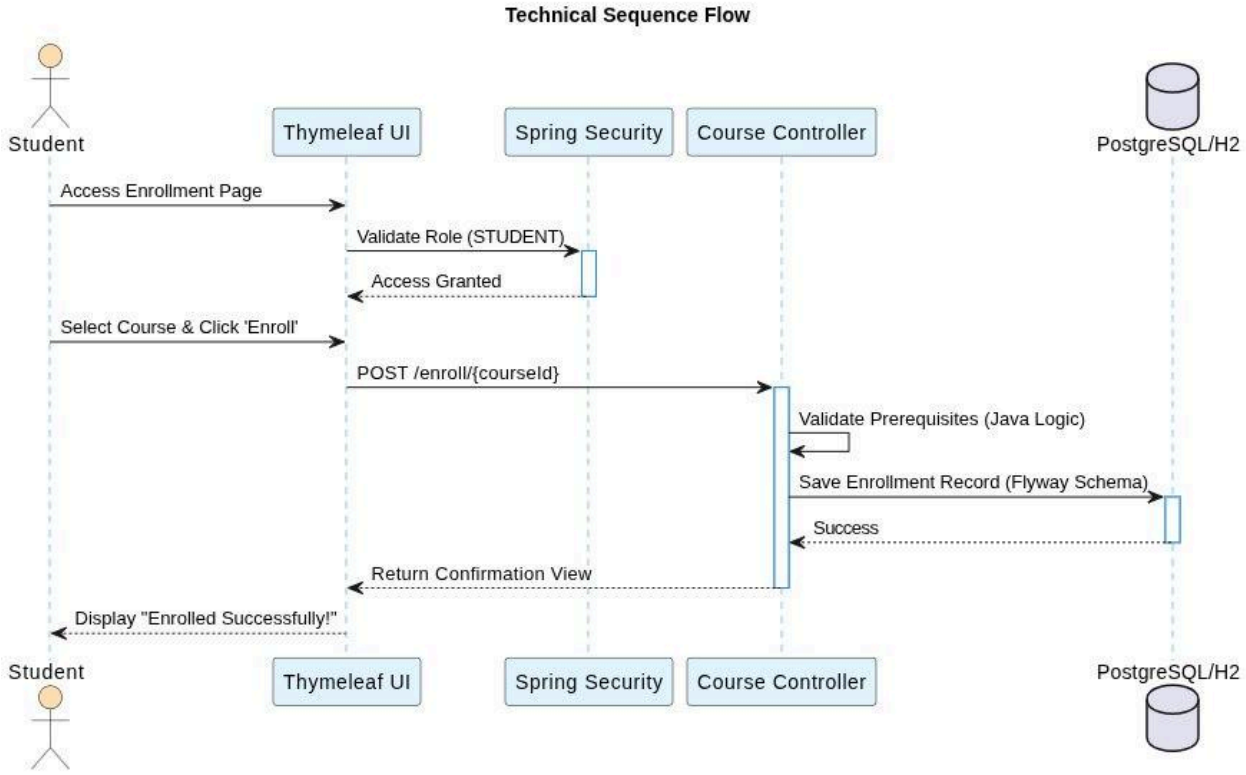
```
/login

/courses

/enroll/{courseId}

/admin/dashboard
```

Access is enforced using Spring Security.

XII. System Flow

1. User logs in
2. Role is verified
3. User is redirected to dashboard
4. Action is validated

5. Data is saved
6. UI feedback is shown



XIII. Testing and Validation

Manual testing for:

- Authentication
- Role access
- Enrollment validation
- Scheduling conflicts

Database integrity validated via constraints

XIV. Challenges and Solutions

Challenge	Solution
Role-based access	Spring Security
Scheduling conflicts	Validation logic
Data consistency	Flyway migration

XV. Conclusion

The Course Enrollment and Classroom Scheduling System successfully fulfills its objectives by providing a secure, role-based, and scalable platform for managing academic resources. The system follows best practices in software engineering, including MVC architecture, separation of concerns, and database version control.

XVI. Future Improvements

- Add timetable conflict detection for students
- Add REST API for mobile integration
- Improve UI with advanced JavaScript
- Add automated testing

References

- Spring Boot Documentation
- Spring Security Reference
- Thymeleaf Documentation