Institute of Technology of Cambodia

Course Enrollment & Class Schedule

Lecturer Mr. Roeun Pacharoth

Group members

Team Leader
Pang Lythong
e20220161

Member
Soeury Sreyno
e20220908

Member
Nget Darapich
e20221646

Member
Vicheth Sokhsedtha
e20221549

Member
Keo Chanponlork
e20220660

Table of Contents

I. Project Overview and Objectives

An introduction to the Software Engineering GIC project, focusing on the development of a system for course registration and classroom management.

Planning Project

Week 1: Foundation & Database Design

Week 2: Security & Identity Management

Week 3: Core CRUD Development

Week 4: Enrollment, UI Polishing & Final Delivery

Planning task responsible

Pang Lythong	Team Lead, Database
Vichet Sokhsedtha	Security and Authentication
Keo Chanponlork	Main Entity CRUD (The Core Data)
Nget Darapich	Secondary Module CRUD (The Actions/Logic)
Soeury Sreyno	Frontend/Thymeleaf

II. Project Flow and Diagram

Project Flow

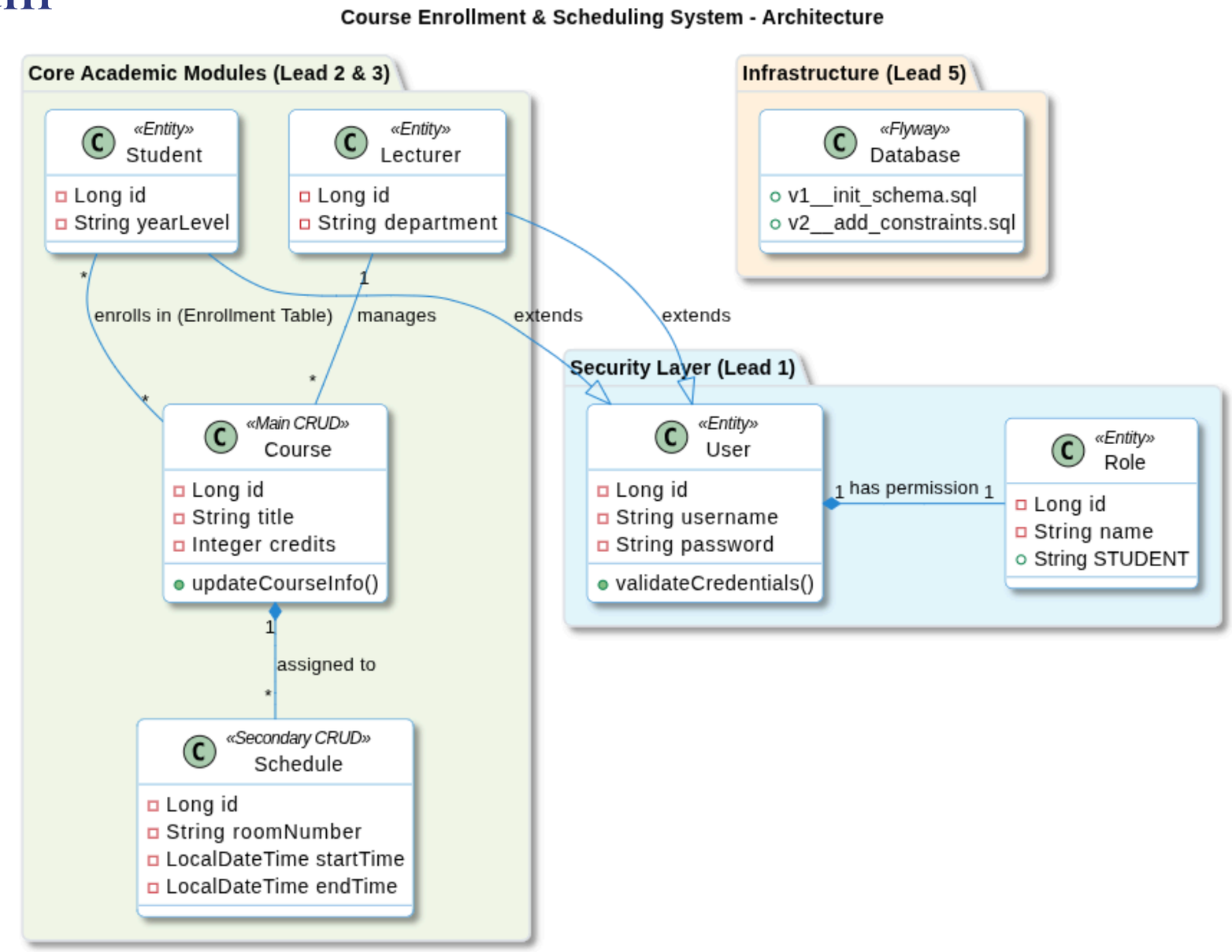
The Foundation (Database & Setup)

The Security Gate (Authentication)

The User Workflow (Admin,Lecturer and Student)

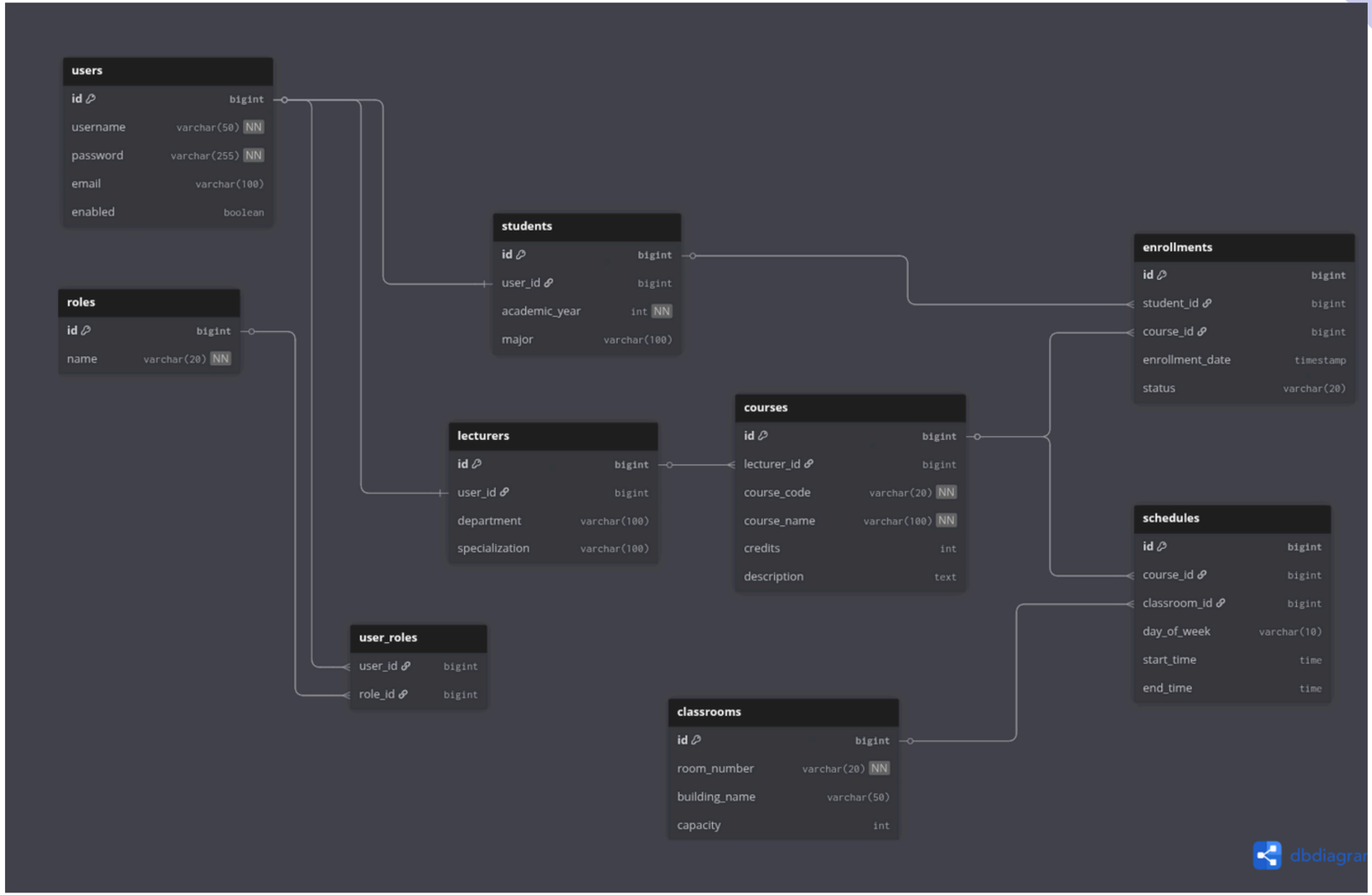
UML Diagram

PUML



UML Diagram

Database



III. Technical Stack and Architecture

Built tool vscode

Library

API

Programming and Markup Languages

IV. Core Functional Modules With API

1. Security and Authentication Module

- *Functionality: Handles user login, logout, and role-based access control (RBAC) for Admins,Lecturers, and Students.*
- *Core Endpoints (Inferred):*
  - *POST /login: Authenticates users and establishes sessions.*
  - *GET /logout: Terminates the user session.*
  - *GET /api/user/role: Returns the current user's permissions to the Thymeleaf UI for access restriction*

2.Course Management Module (Main CRUD)

Functionality: Allows lecturers to define the academic catalog by managing course data.

- *Core API Actions (Inferred):*
  - *GET /courses: Retrieves a list of all available courses for students and lecturers.*
  - *POST /courses: Creates a new course (requires validation for course codes and names).*
  - *PUT /courses/{id}: Updates existing course details.*
  - *DELETE /courses/{id}: Removes a course from the system.*

3. Classroom Scheduling Module (Secondary CRUD)

Functionality: Allows lecturers to organize and view classroom schedules.

- Core API Actions (Inferred):
  - GET /schedules: Displays the master schedule or room availability.
  - POST /schedules: Maps a specific course_id to a classroom_id and a time slot (requires validation to prevent double-booking).
  - DELETE /schedules/{id}: Cancels a specific scheduled session.

4. Student Enrollment Module

- *Functionality: Enables students to register for courses managed by lecturers.*
- *Core API Actions (Inferred):*
  - *GET /enrollment/available: Lists courses that have open seats.*
  - *POST /enroll: Creates an entity relationship between a student_id and a course_id in the database.*
  - *GET /student/my-courses: Retrieves the specific list of courses a student has joined*

5. Database & UI Integration Layer

This represents the work of the Database Lead and Frontend Lead to ensure data persistence and visibility.

- *Flyway Migrations: Ensures the database schema supports the relationships between users, courses, schedules, and enrollments.*
- *Thymeleaf Access Restrictions: Ensures that "Management" API actions (like POST /courses) are hidden from students in the UI*

V. Security Architecture and Development Workflow

Security Architecture

- ***Authentication and Identity Management:*** *The Security and Authentication Lead is responsible for the login and logout mechanisms. This ensures that every user—whether a student, lecturer, or admin—is identified before accessing the system.*

- ***Authorization and Role-Based Permissions:*** *Spring Security is used to enforce authorization. The system distinguishes between roles, such as Lecturers (who manage courses and schedules) and Students (who enroll in them).*

- ***Frontend Access Restrictions:*** *Beyond backend security, the Frontend/Thymeleaf Lead implements access control directly in the UI templates.*

- ***Secure Database Integration:*** *The Database Lead supports security by creating a schema that includes user and role tables, managed via Flyway migrations to ensure consistent data structures across all development environments.*

Development Workflow

- ***Build and Dependency Management:*** *The project uses Gradle as its build automation tool, as evidenced by the presence of build.gradle, settings.gradle, and the gradle/wrapper files in the repository.*

- ***Continuous Version Control:*** *The workflow requires Git commit logs from all five members to track contributions and progress. The current repository shows a history of 35 commits and active participation from four contributors so far.*

- ***Multi-Language Integration:*** *The codebase is a mix of Java (48.2%) for the backend logic and security, HTML (42.6%) for Thymeleaf templates, and CSS (9.2%) for styling.*

- ***Automated Migrations:*** *The team uses Flyway to handle database changes. This allows the "Database Lead" to push schema updates that are automatically applied to other members' local environments when they pull the latest code.*

- ***Deliverable Lifecycle:*** *The workflow culminates in Complete Documentation and a Group Presentation, where the team demonstrates that the CRUD modules, security protocols, and UI restrictions work together seamlessly.*

VI. Conclusion

The Course Enrollment and Classroom Scheduling System represents a comprehensive software engineering project that integrates backend logic, database management, and secure user interfaces to solve academic administrative challenges. Based on the project requirements and current repository status, the following points conclude the project's scope and objectives: Integrated Technical Stack, Robust Security and Access Control, Functional Excellence and Collaborative Development .

Demonstration

Group 3

Thank You

13 Jan, 2026

Q & A