SWENG 837 COURSE PROJECT

JESSICA KUNKEL

SOFTWARE SYSTEM DESIGN

SUMMER II 2024

Introduction

- Course Scheduling System
  - Aims to provide an effective means for students to build their course schedules and enroll in courses
  - Professors can select course sections that fit with their schedule
  - Course Administrators can manipulate courses and students’ enrollment as necessary

DESIGN SOLUTION OVERVIEW

Design Solution

- The Course Scheduling System
  - Provides course search and scheduling functionality
  - Ensures students don’t enroll in classes that overlap in time, that they do not meet prerequisites for, or that they cannot take due to total semester credit limits.
  - Upon creation of a new course or course section, assigns each section to a time and location, utilizing classroom preferences for each school in the university (e.g., an English class usually wouldn’t occur in the Chemistry building) and the classrooms’ availability

Design Solution

- Students build their schedules by searching for a course, selecting one section of the course, and adding that section to their schedule.
  - The system checks the student's course history vs the class's prerequisites, the number of credits the student has added to their schedule, and whether any course sections overlap in their scheduled times (enrollment rules). The student is not allowed to complete enrollment if one of these criteria are not met.

Design Solution

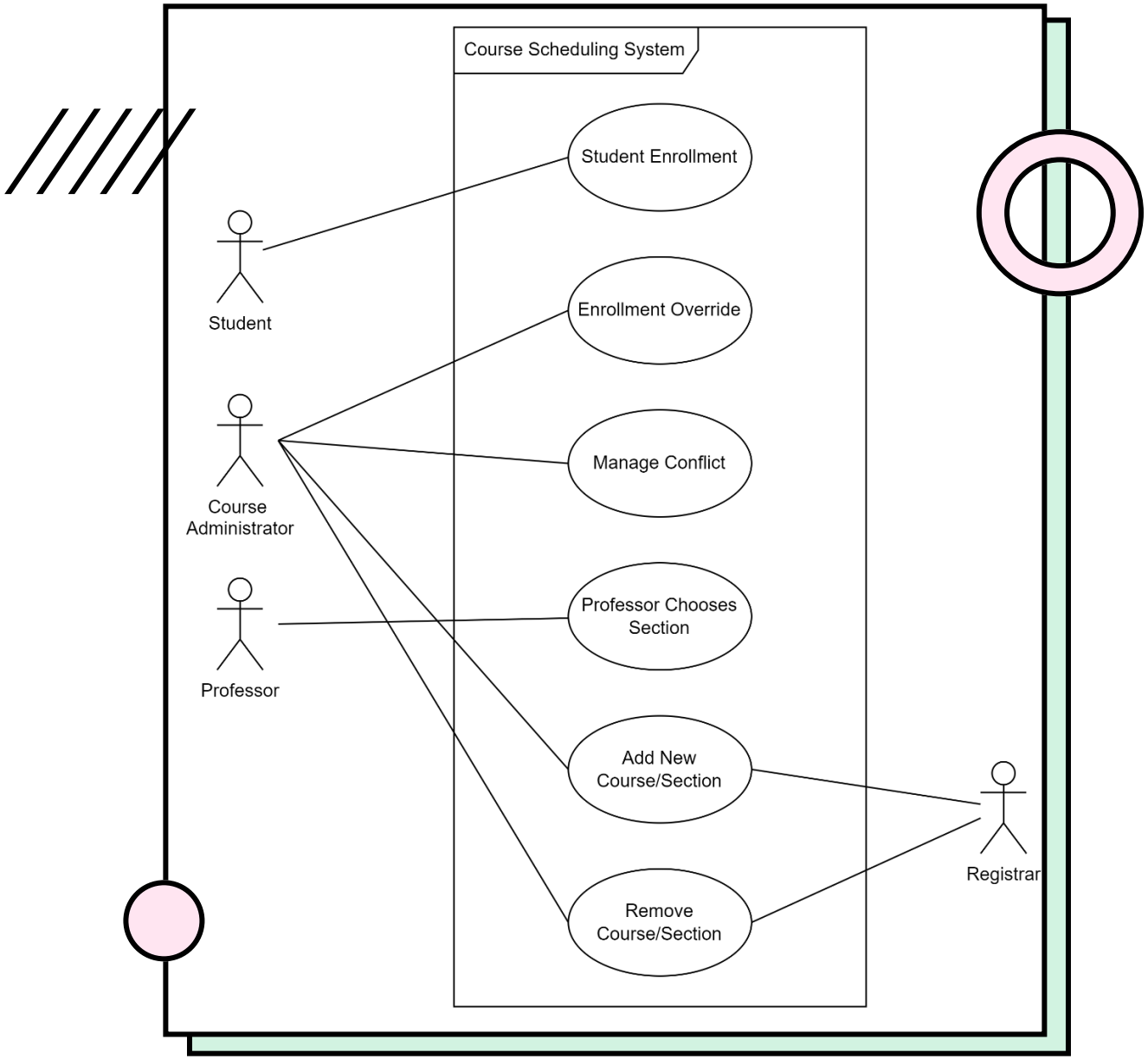
- Professors build their schedules by searching for a course, selecting one section of the course, and adding that section to their schedule, effectively assigning themselves to a course section.
  - The system checks whether any course sections overlap in their scheduled times and will not allow the professor to complete their registration for a course section if there are two courses occurring at the same time on their schedule.

Design Solution

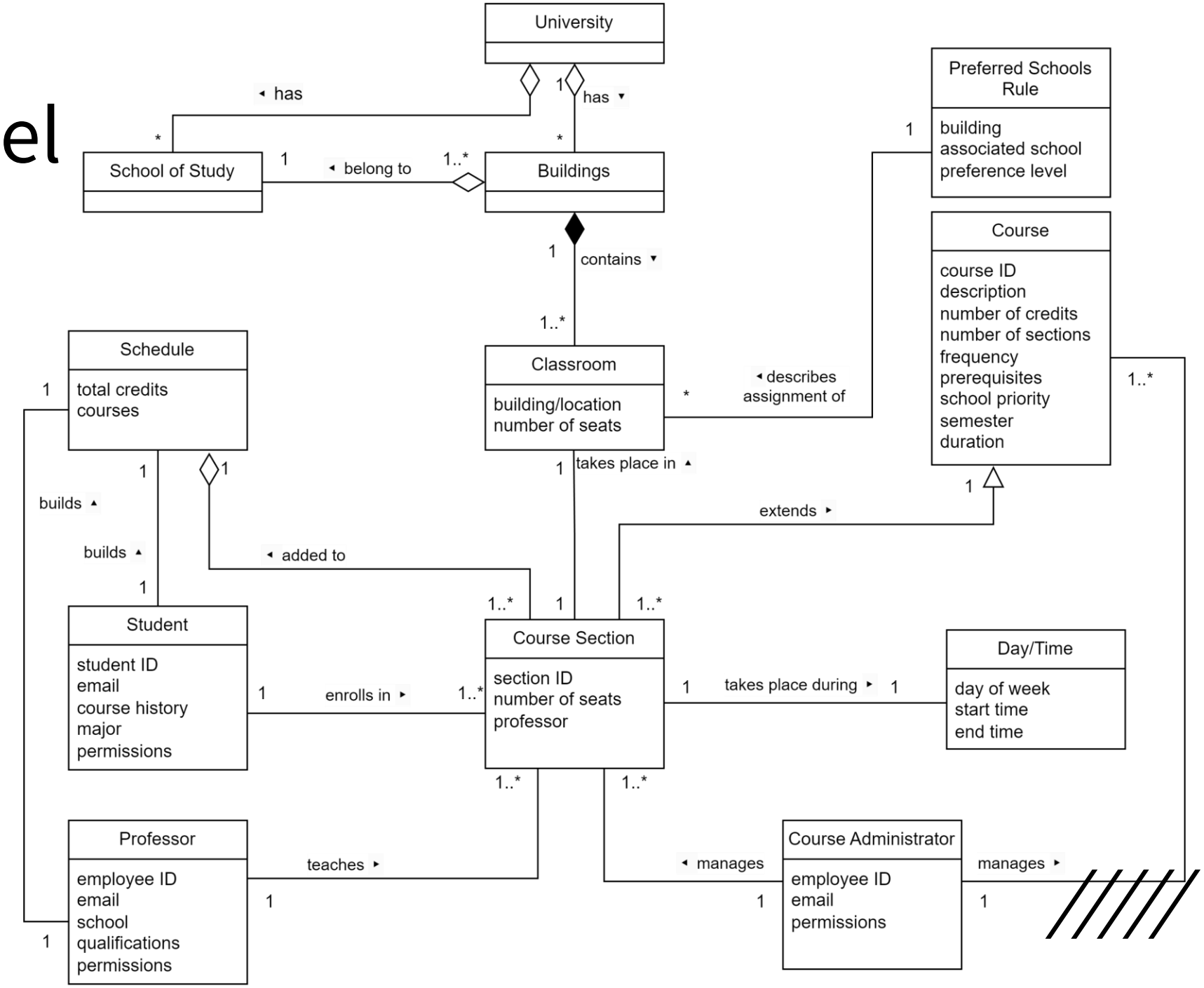
- Course Administrators manage the CSS
  - Add new courses and course sections using information provided by Registrar
  - Remove courses and course sections due to lack of staff or lack of enrollment
  - Receive requests from students to allow exceptions to enrollment rules, and manually enroll students in those courses
  - Manage any classroom or time conflicts such that no two courses occur in the same classroom at the same time.

DETAILED DESIGN SOLUTION

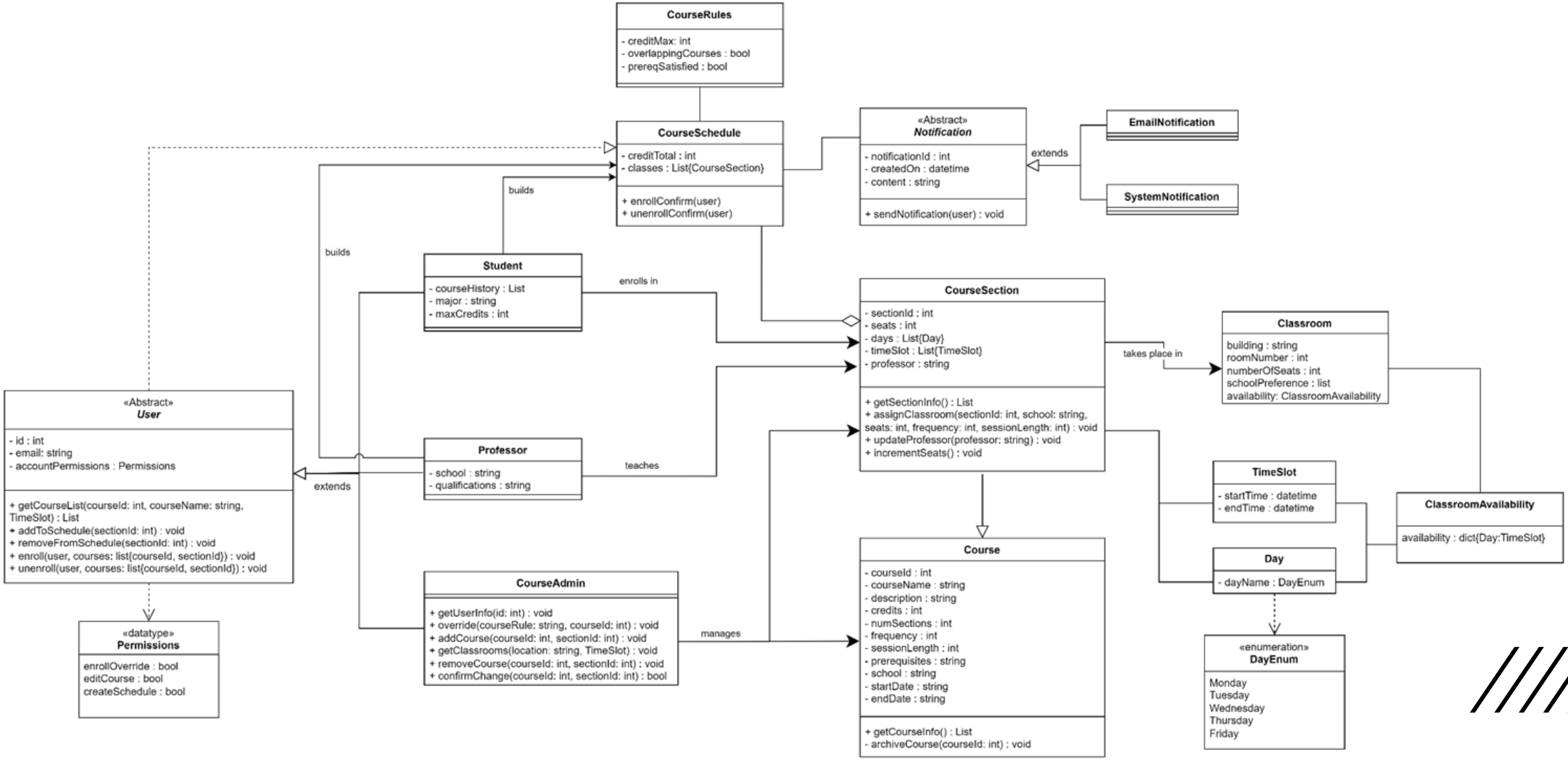
USE CASES


Course Scheduling System
Student Enrollment
Enrollment Override
Manage Conflict
Professor Chooses Section
Add New Course/Section
Remove Course/Section
Student
Course Administrator
Professor
Registrar

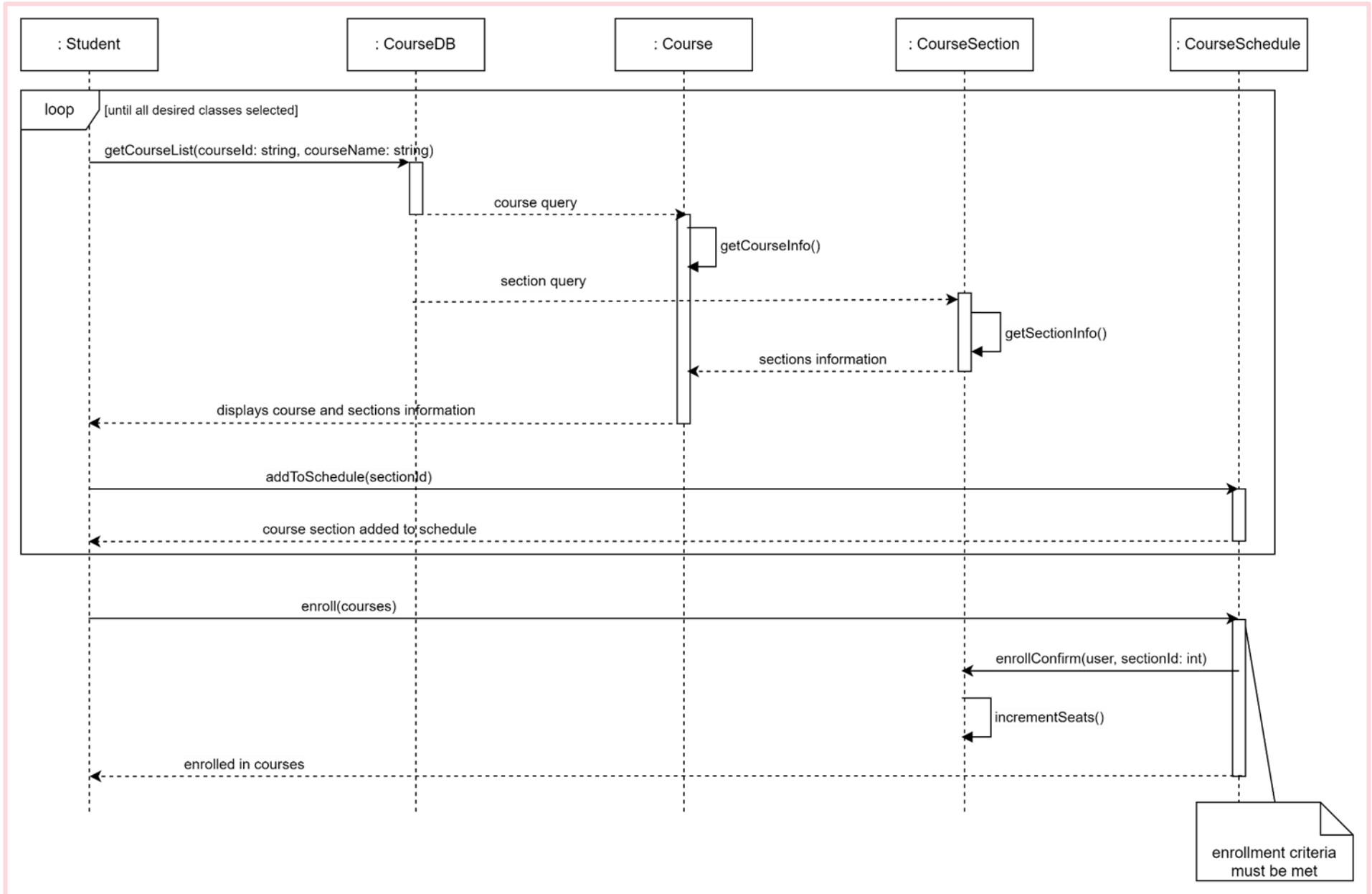
Domain Model



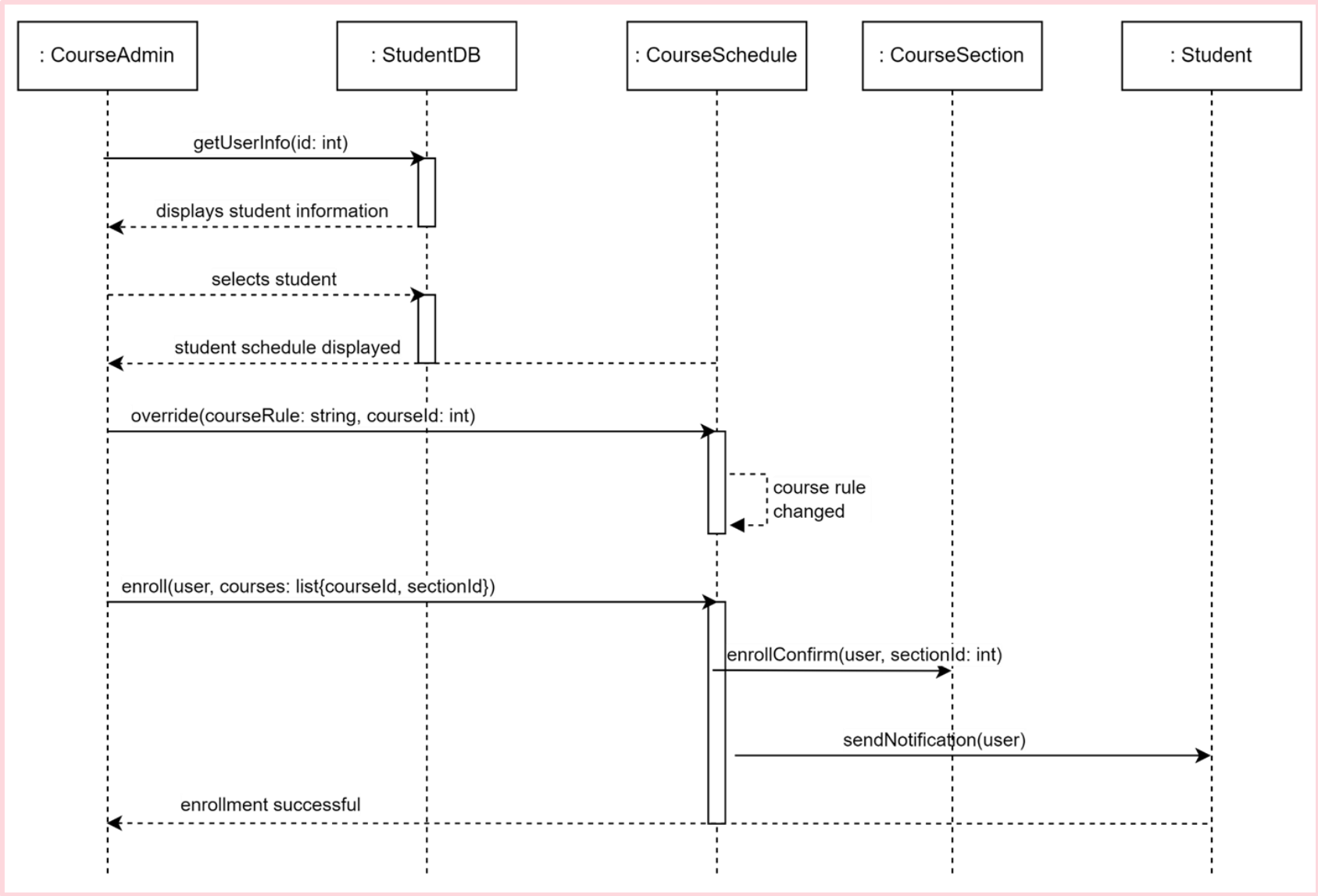
Class Diagram



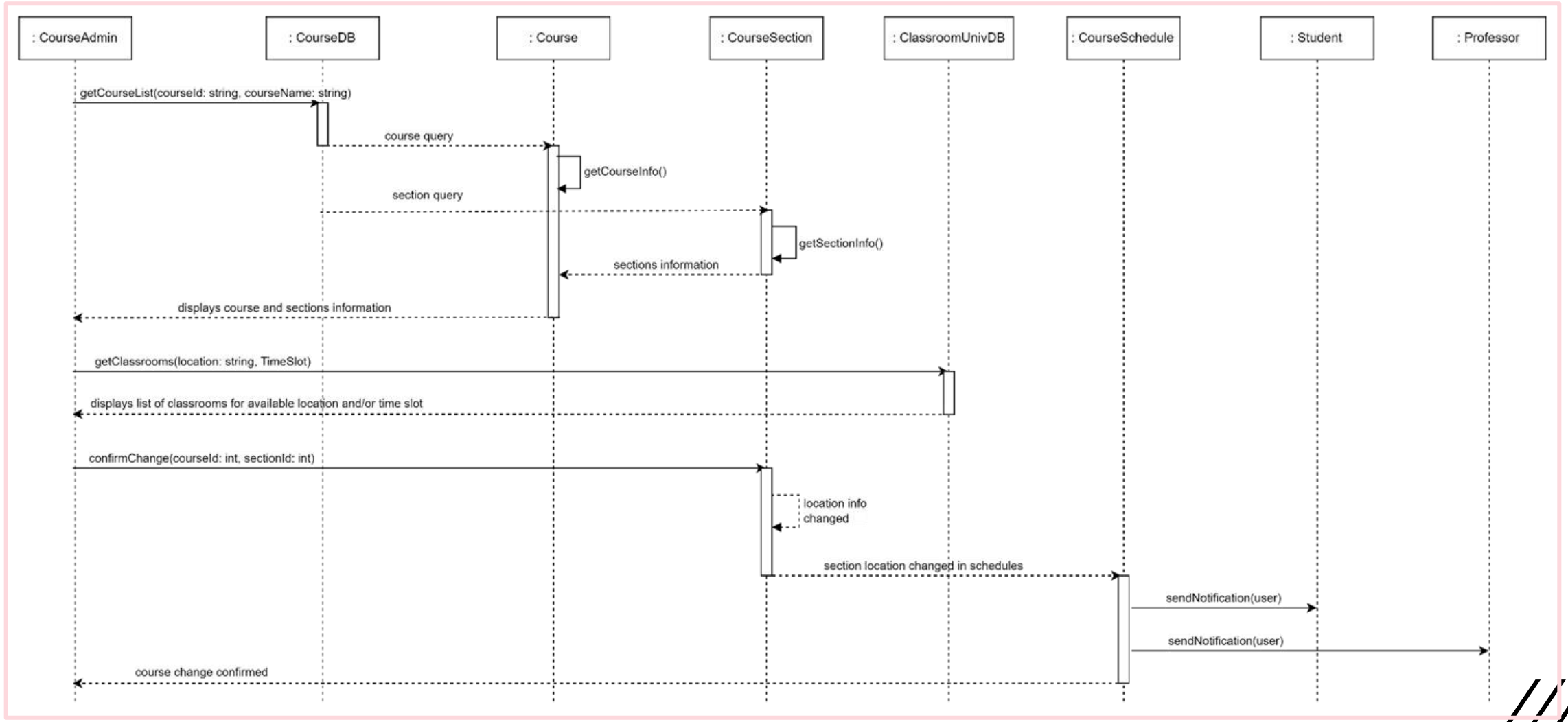
Student Enrollment



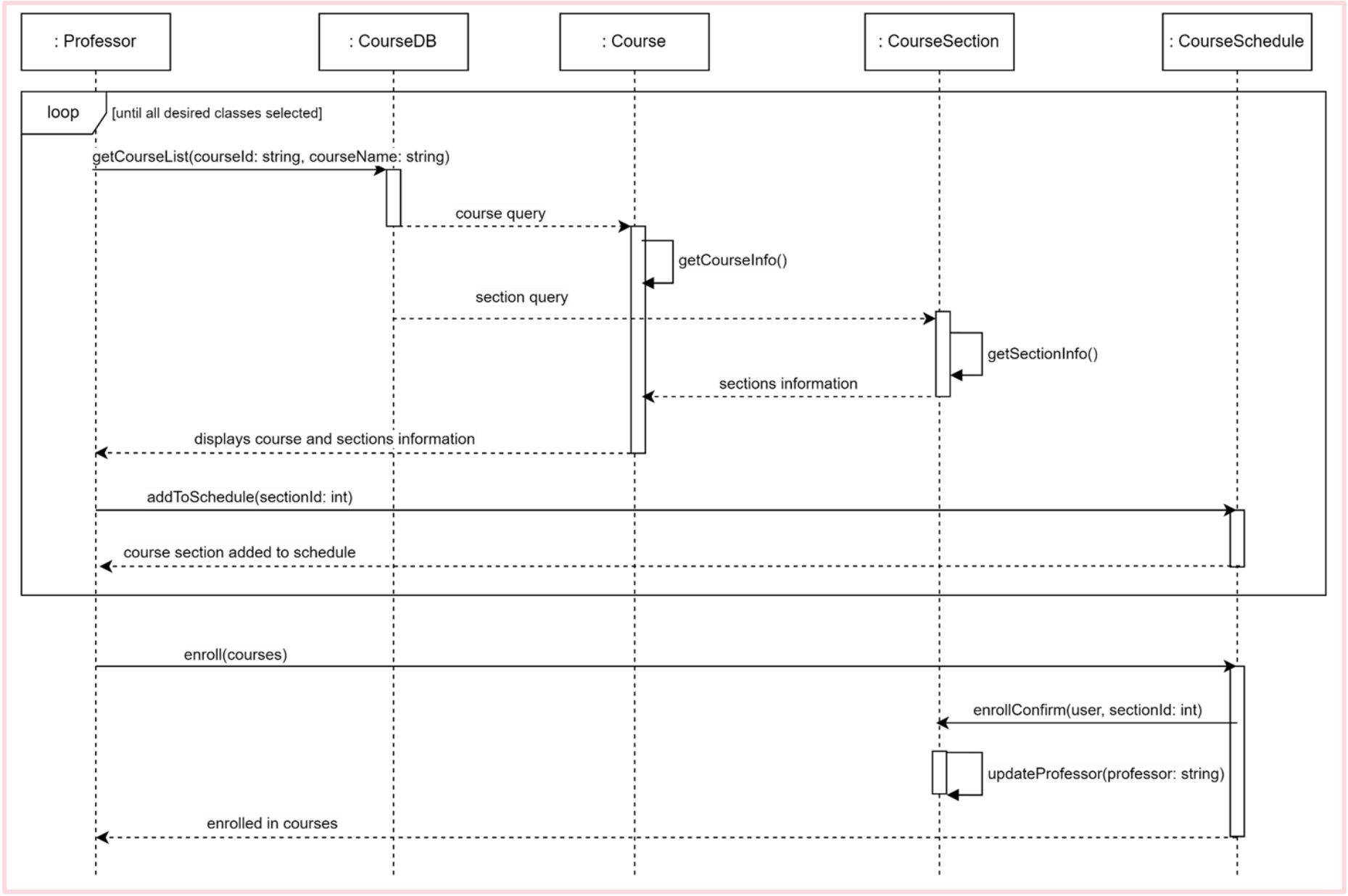
Enrollment Override



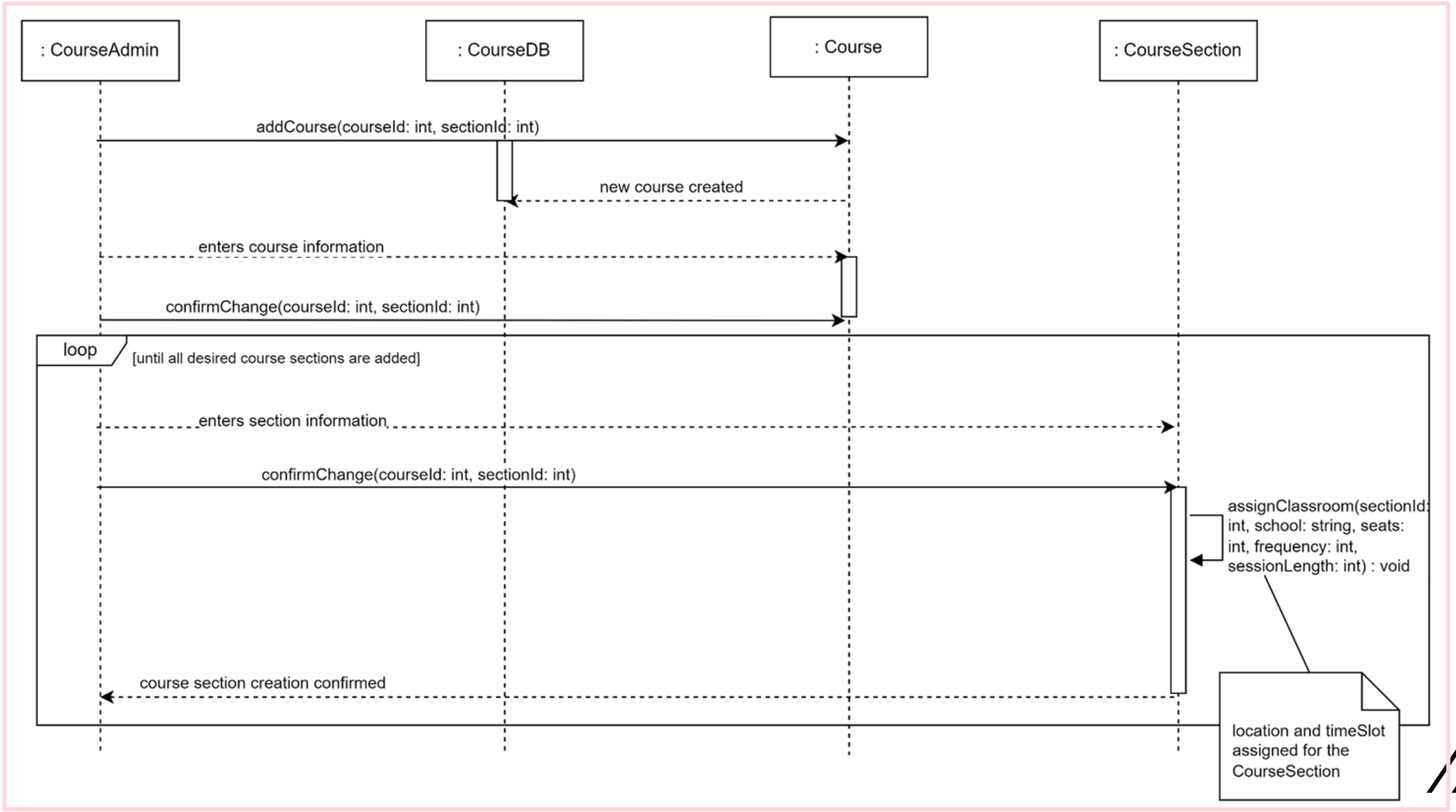
Manage Conflict



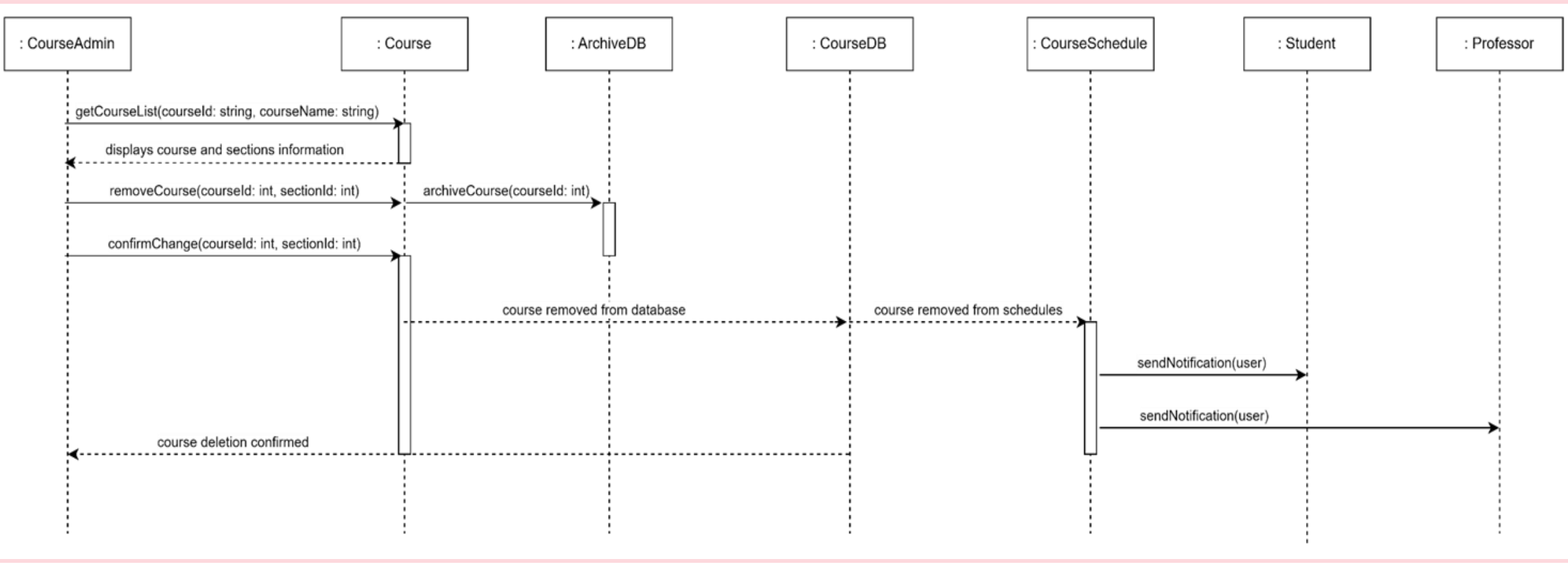
Professor Chooses Section



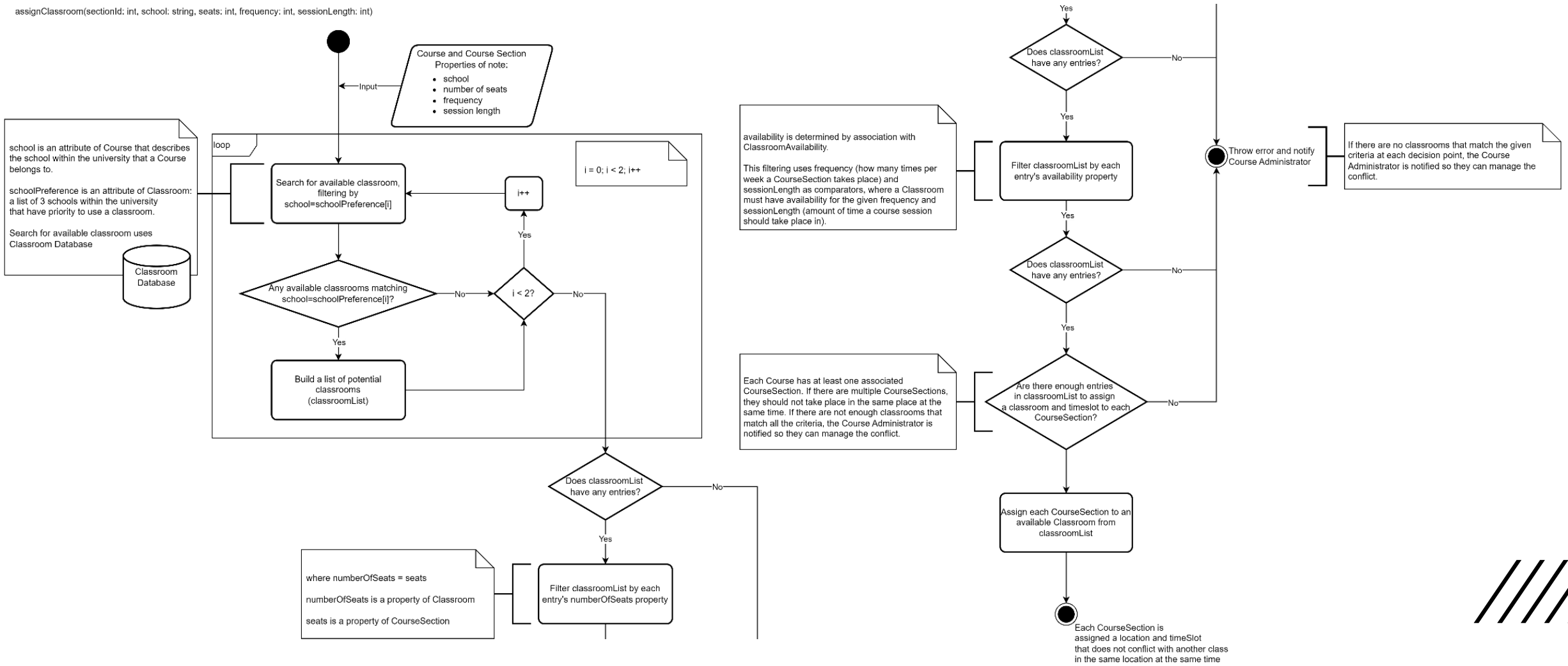
Add New Course/Section



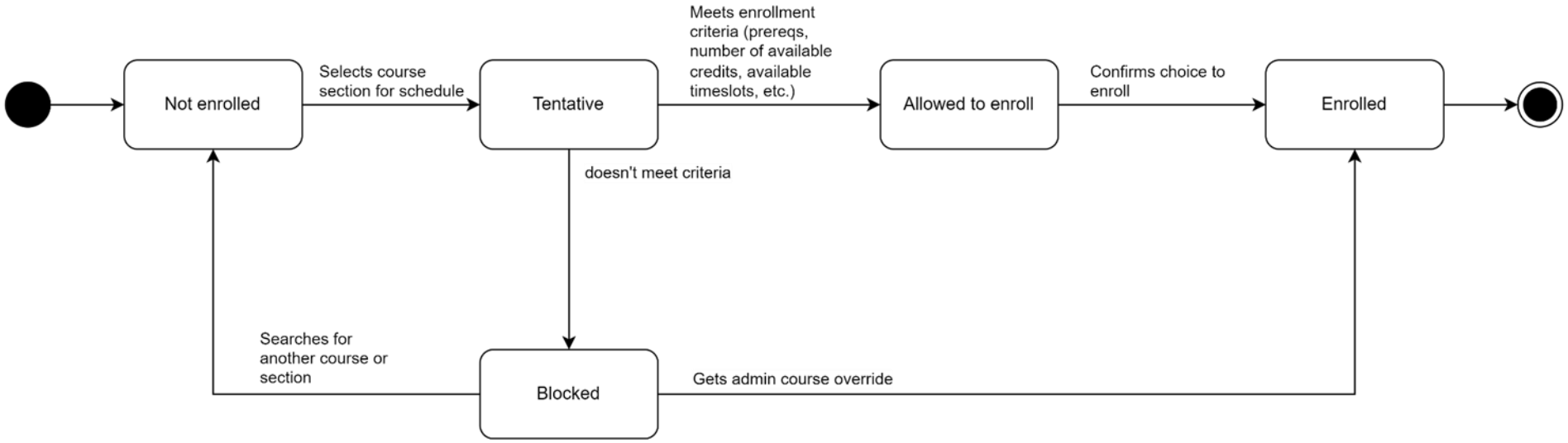
Remove Course/Section



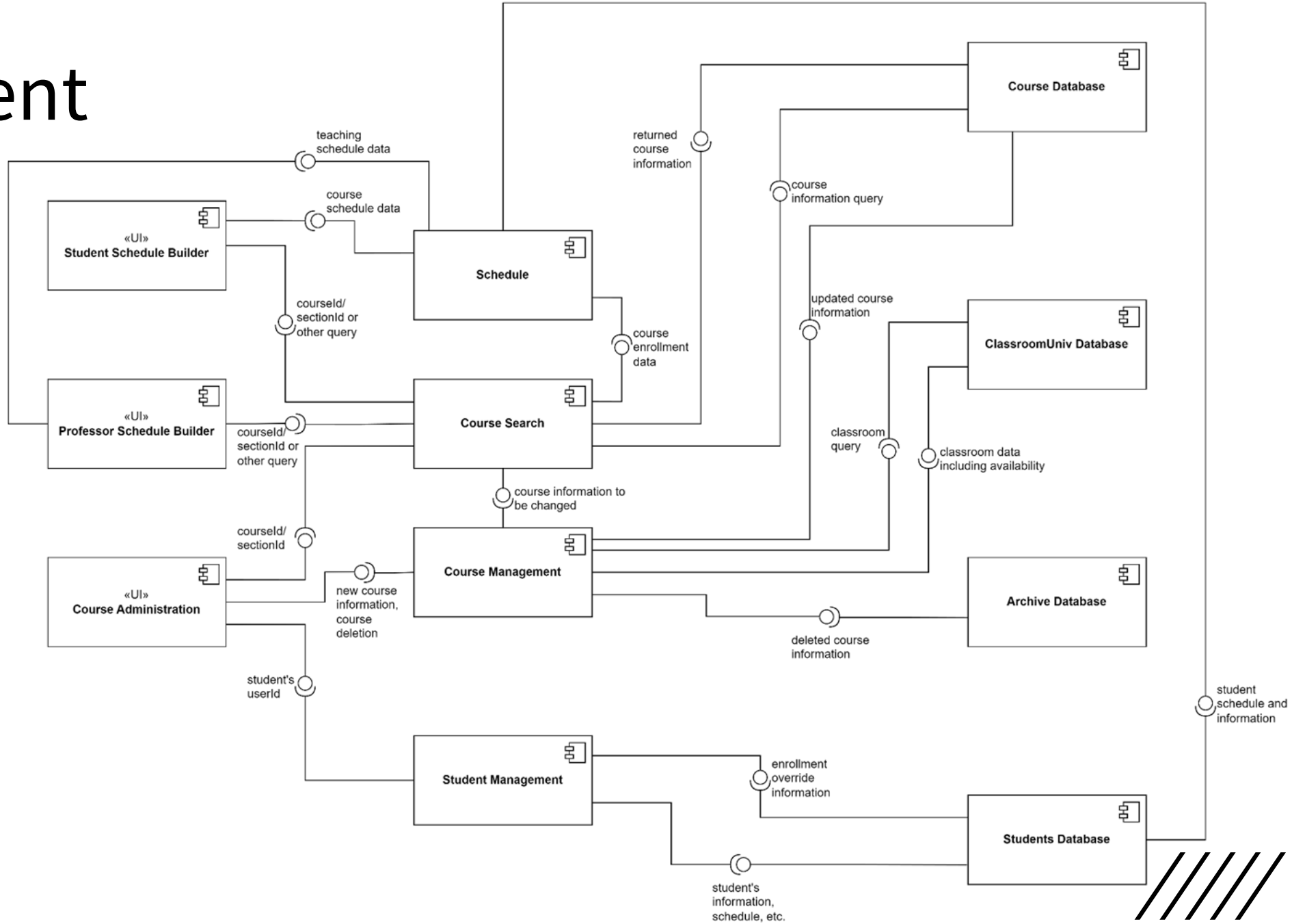
Activity Diagram: Assign Classroom



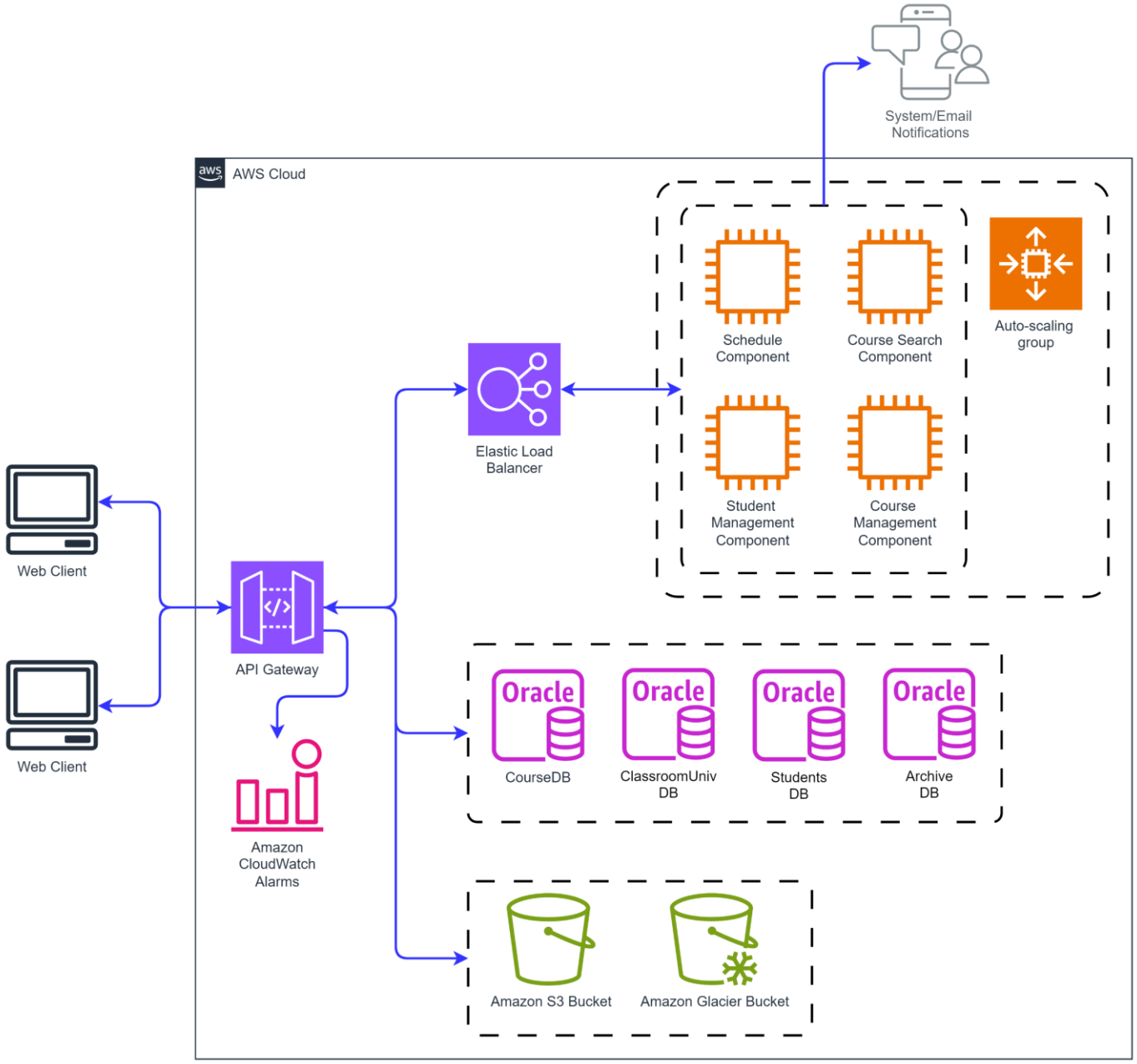
State Diagram: Student Object



Component Diagram



Deployment Diagram



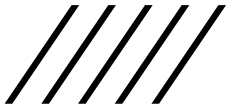
Design Patterns

- GRASP
  - Expert
  - High Cohesion
- Cloud-Native
  - API Gateway

CONCLUSION

Conclusion and Next Steps

- Gained valuable experience designing an object-oriented software system and building software system design skills
- Next Steps
  - Refine dependencies between components
  - Introduce more GoF Patterns
  - Turn this exercise into a full-blown, portfolio-worthy software system



THANK YOU!