

PROJECT OVERVIEW: University Management System using Spring MVC and Hibernate

This project has three sections – Instructor, Course and Student.

- Instructor – Displays a list of Instructors with all the Hibernate CRUD operations. We can see all the courses of each instructor. We can add those courses that does not have any instructor assigned.
- Course – Displays a list of Courses with all the Hibernate CRUD operations. We can see the instructor name for each course. We can see all the students of each course.
- Students – Displays a list of Students with all the Hibernate CRUD operations. We can see all the courses of each student. We can add those courses that the student did not take.

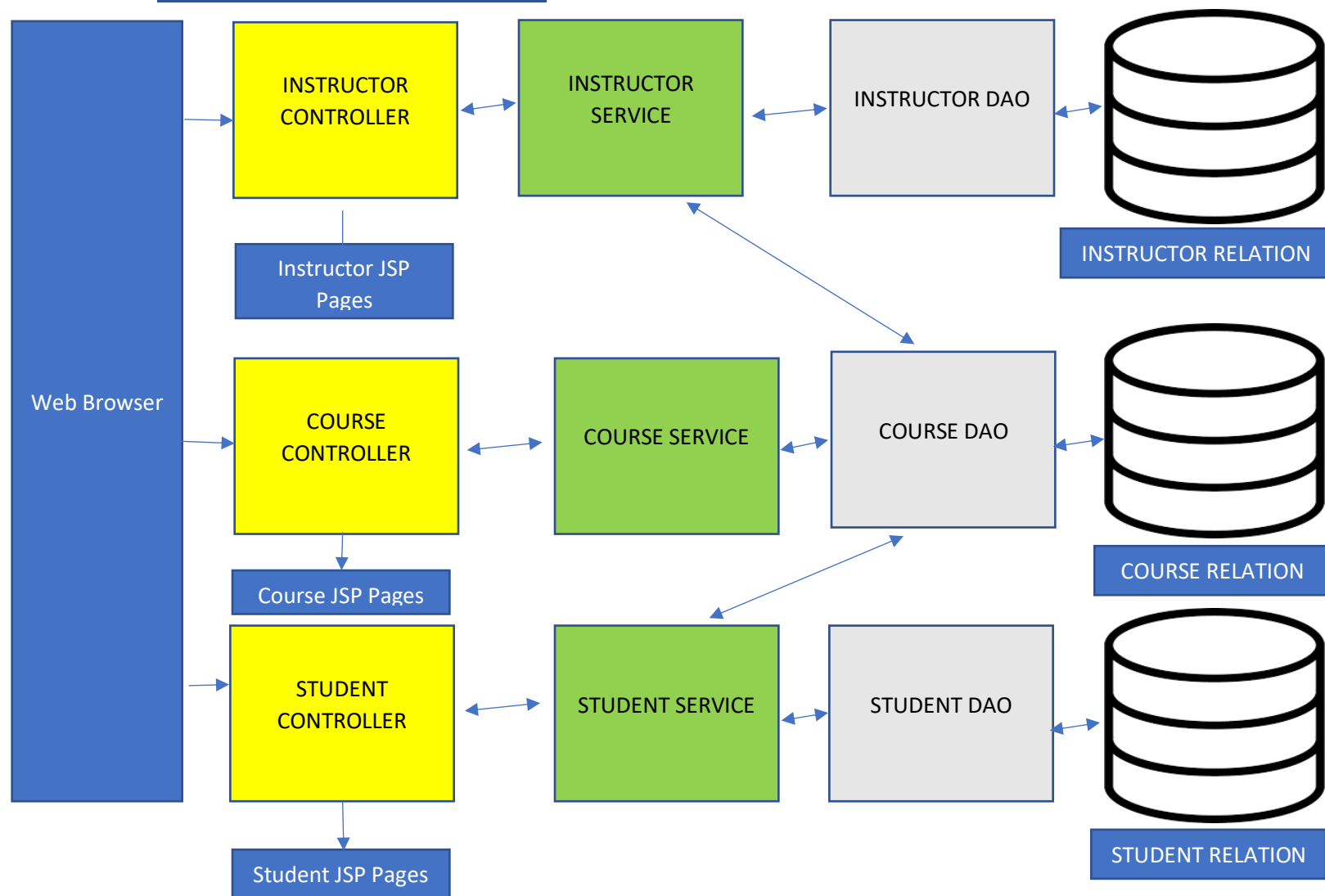
This project has two requirements –

1. A student can take many courses. Simultaneously, a course can have many students. (@ManyToMany)
2. An instructor can take many courses (@OneToMany). Many courses can have one instructor (@ManyToOne). (Or a course can only have one instructor).

Technologies Used:

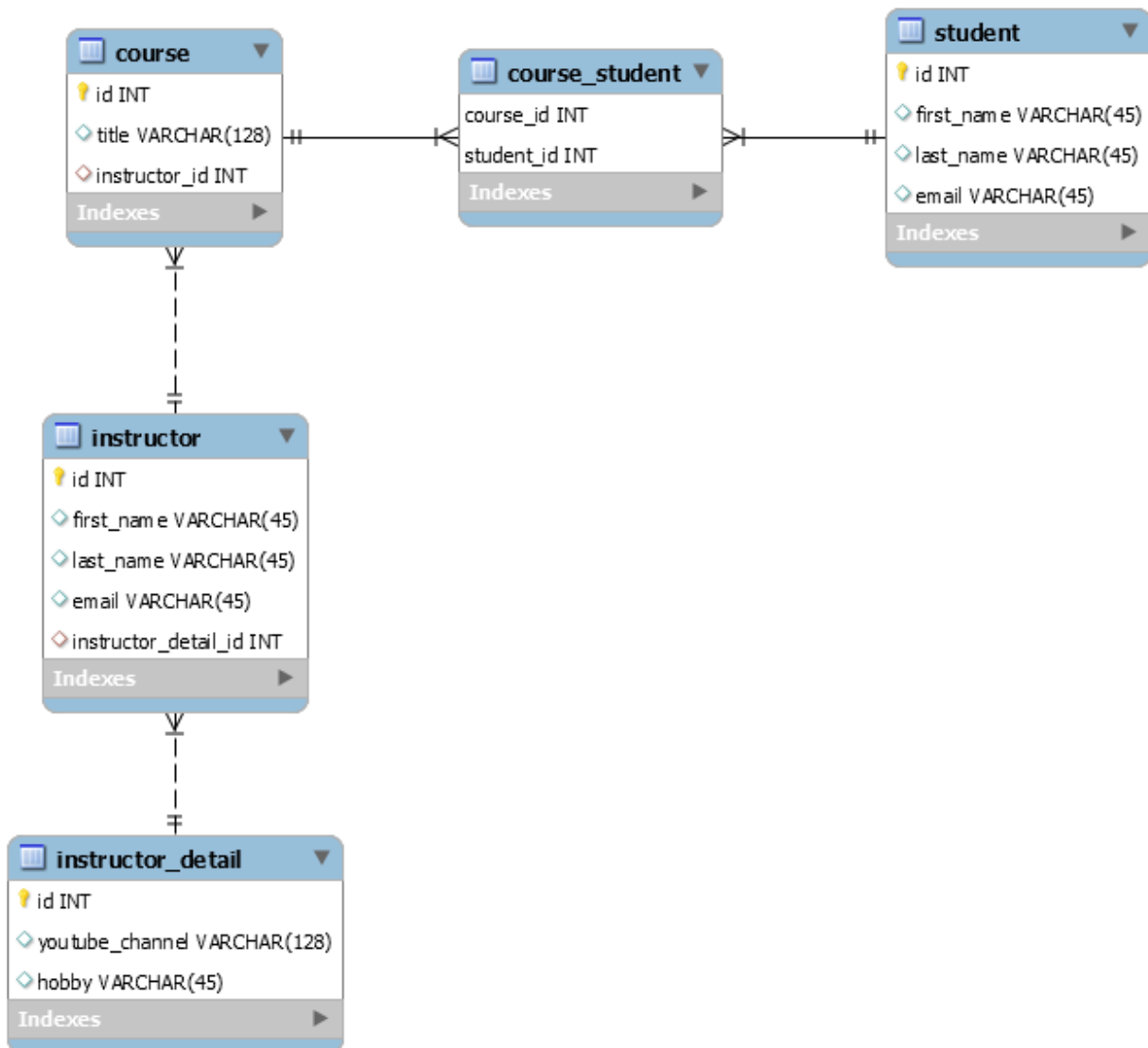
1. HTML CSS in JSP Pages
2. Spring Framework – Inversion of Control and Dependency Injection using Java Configuration
3. Spring MVC - Models, Views, Controllers, Services, Data Access Objects, Request Param and Request Mapping, Form Tags and Data Binding, Built-in Validation Rules
4. Hibernate CRUD and Hibernate Mappings - @OneToMany and @ManyToMany

PROJECT ARCHITECTURE



- **Data Access Objects** – The Instructor DAO, Course DAO and Student DAO are responsible for interfacing with the Instructor Table, Course Table and the Student Table respectively.
- **Services** – This is the intermediary layer between the controller and the database. The main job of services is to integrate data from multiple repositories/DAOs. For example, the Instructor Service uses both Instructor DAO and Course DAO to get its data. Moreover, the Student Service uses both Student DAO and Course DAO to get its data.
- **Controllers** – The controllers handle requests. Store data in model and send to appropriate view template (JSP Page). Retrieve data from model from JSP form pages.

EER DIAGRAM



Mappings –

1. A course can have many students and a student can have many courses (@ManyToMany)
2. A course can only have one instructor (@ManyToOne) but an instructor can have many courses (@OneToMany)

Foreign Keys –

1. **course_id** in **table course_student** refers to **id** in **table course**
2. **student_id** in **table course_student** refers to **id** in **table student**
3. **instructor_id** in **table course** refers to **id** in **table instructor**