HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY
School of Information and Communication Technology

BÁO CÁO BÀI TẬP LỚN
LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Hệ thống quản lý sinh viên

Nhóm 6

Thành viên	*Email*
Bùi Quốc Bảo 20225601	bao.bb225601@sis.hust.edu.vn
Nguyễn Đăng Diện 20225805	dien.nd225805@sis.hust.edu.vn
Nguyễn Kiều Duyên 20225618	duyen.nk225618@sis.hust.edu.vn
Đặng Kim Ngân 20225751	ngan.dk225751@sis.hust.edu.vn
Lê Hải Yến 20225780	yen.lh225780@sis.hust.edu.vn

Lớp
151964 - IT3103
Giảng viên
Lê Đức Hậu
hau.leduc@hust.edu.vn

Ngày 26 tháng 12 năm 2024

Báo cáo
Hệ thống quản lý sinh viên
Lớp: 151964
Giảng viên: Lê Đức Hậu

Mục lục

1 Mô tả bài toán **3**
1.1 Mô tả sơ bộ . . . 3
1.2 Phân tích tổng quan chức năng hệ thống . . . 3

2 USECASE DIAGRAM **4**
2.1 Giới thiệu chung . . . 4
2.2 Các thành phần chính . . . 4
2.3 Phân tích biểu đồ . . . 5

3 ERD DIAGRAM **6**
3.1 Các thực thể chính . . . 6
3.2 Quan hệ giữa các thực thể . . . 6
3.3 ERD Diagram . . . 7

4 Kiến trúc hệ thống **8**
4.1 Mô hình MVC cơ bản . . . 8
4.2 Mô hình MVC mở rộng . . . 8
4.3 Giải thích . . . 9

5 CLASS DIAGRAM **9**
5.1 Các lớp và các phương thức chính . . . 9
5.2 Class Diagram . . . 10
5.3 Tổng quan hệ thống . . . 11
5.4 Chi tiết các thành phần . . . 11

6 Các tính chất OOP **11**

7 Các công nghệ sử dụng **12**

8 Demo **13**

9 Thu hoạch **28**

10 Tài liệu tham khảo **29**

Ý tưởng khởi đầu: Trong bối cảnh công nghệ ngày càng phát triển, việc quản lý và tối ưu hóa hoạt động trong các lĩnh vực giáo dục, đặc biệt là quản lý sinh viên, trở nên cần thiết hơn bao giờ hết. Những phương pháp truyền thống dựa trên giấy tờ không chỉ gây ra nhiều sai sót mà còn không đáp ứng được yêu cầu ngày càng phức tạp của hệ thống giáo dục hiện đại. Để đáp ứng nhu cầu quản lý toàn diện và hiệu quả, một hệ thống quản lý sinh viên thông minh và áp dụng công nghệ hiện đại được xem là giải pháp tối ưu.
Vậy nên bài báo cáo này sẽ đề xuất "Hệ thống quản lý sinh viên" để giải quyết vấn đề đã được trình bày ở trên.

1 Mô tả bài toán

1.1 Mô tả sơ bộ

- Hệ thống quản lý thông tin sinh viên cơ bản dùng để quản lý dữ liệu liên quan đến sinh viên, giảng viên và khóa học.
- Hệ thống được thiết kế với ba đối tượng người dùng chính đó là: quản trị viên, giảng viên và sinh viên.

1.2 Phân tích tổng quan chức năng hệ thống

Quản lý hoạt động trong hệ thống quản lý sinh viên bao gồm việc theo dõi và điều phối tất cả các hoạt động của sinh viên và giảng viên trong khóa học, đảm bảo rằng mọi quy trình diễn ra một cách trôi chảy và hiệu quả.

Quản trị viên (Admin)

- **Đăng nhập tài khoản**: Tài khoản của quản trị viên sẽ được lưu sẵn trong bộ nhớ.
- **Quản lý sinh viên**: Xem thông tin tất cả các sinh viên, xóa sinh viên, thêm khóa học cho sinh viên, xóa khóa học cho sinh viên.
- **Quản lý giảng viên**: Xem thông tin tất cả các giảng viên, xóa giảng viên, thêm khóa học cho giảng viên, xóa khóa học cho giảng viên.
- **Quản lý khóa học**: Tạo khóa học, phân công giảng viên cho khóa học, xem tất cả các khóa học, xóa khóa học, thêm sinh viên vào khóa học, xóa sinh viên khỏi khóa học.

Giảng viên

- **Đăng ký/Đăng nhập tài khoản**: Đăng ký tài khoản của giảng viên bao gồm các thông tin cá nhân của giảng viên đó. Nếu giảng viên đã có tài khoản thì chỉ cần đăng nhập với username và password mà giảng viên đã đăng ký.
- **Quản lý khóa học**: Xem thông tin tất cả các khóa học mà giảng viên đó phụ trách.
- **Quản lý sinh viên**: Xem danh sách sinh viên tham gia từng khóa học đó.
- **Quản lý bài tập**: Xem thông tin các bài tập trong từng khóa học, thêm bài tập mới, xóa bài tập, theo dõi trạng thái hoàn thành bài tập của sinh viên trong khóa học.
- **Quản lý điểm số**: Chỉnh sửa điểm số của từng sinh viên trong từng khóa học tương ứng.

Sinh viên

- **Đăng ký/Đăng nhập tài khoản**: Đăng ký tài khoản của sinh viên bao gồm các thông tin cá nhân của sinh viên đó. Nếu sinh viên đã có tài khoản thì chỉ cần đăng nhập với username và password mà sinh viên đã đăng ký.
- **Quản lý khóa học**: Xem thông tin tất cả các khóa học mà sinh viên đó tham gia.
- **Quản lý bài tập**: Xem thông tin chi tiết bài tập được giao trong từng khóa học, sinh viên cần nhấp vào nút "Mark as completed" cho mỗi bài tập đã làm để giảng viên có thể theo dõi trạng thái hoàn thành bài tập..
- **Quản lý điểm số**: Xem thông tin chi tiết về điểm số trong từng khóa học.

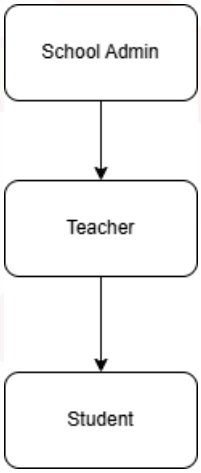
2 USECASE DIAGRAM

2.1 Giới thiệu chung

Use Case Diagram được thiết kế nhằm minh họa các chức năng chính của hệ thống "Web quản lý sinh viên", bao gồm các tác vụ mà từng vai trò người dùng (Admin, Teacher, Student) có thể thực hiện. Biểu đồ này giúp hình dung rõ ràng mối quan hệ giữa các chức năng và vai trò người dùng trong hệ thống.

2.2 Các thành phần chính

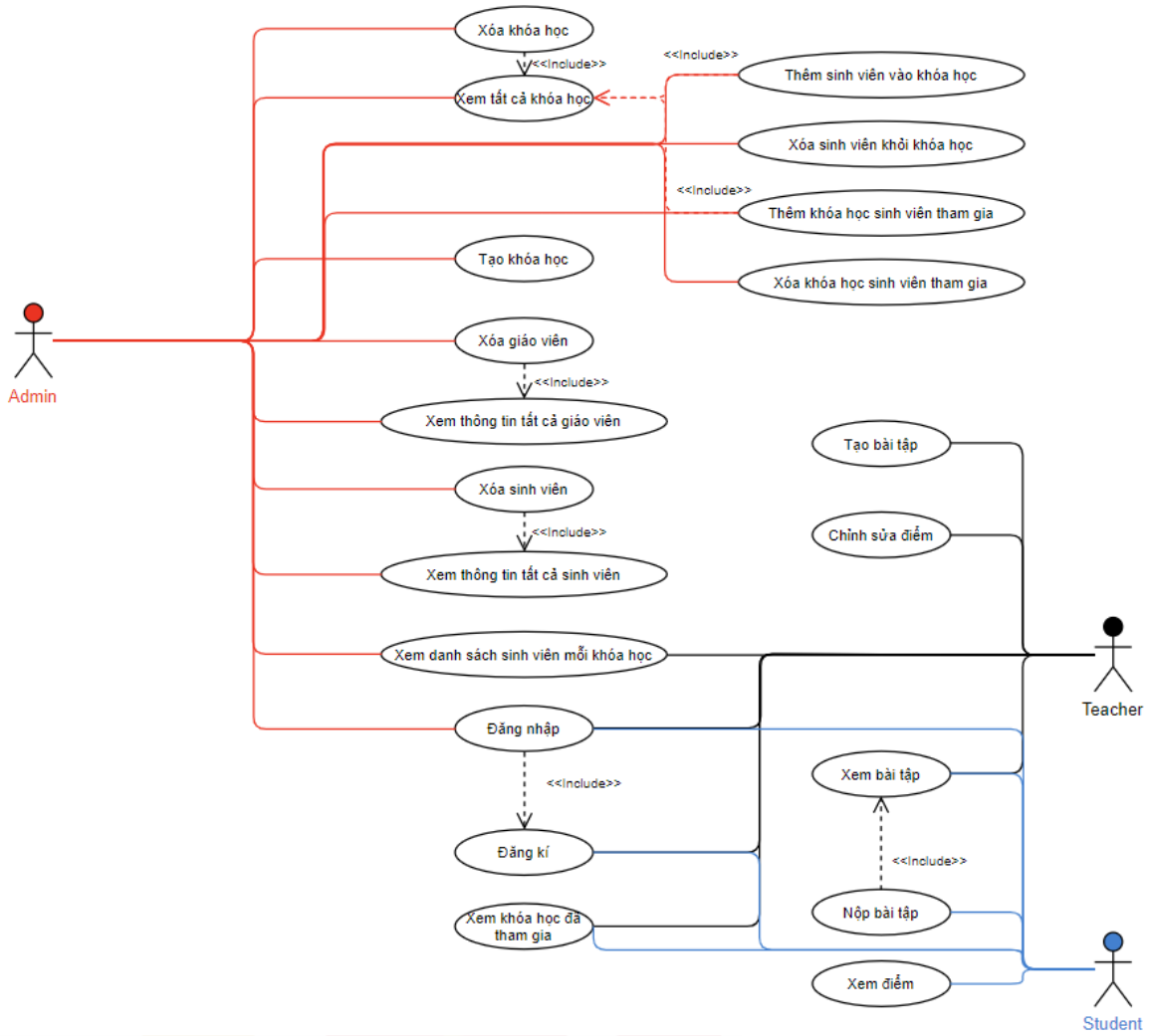
- **Actors**:



Hình 1: Kiến trúc cơ bản của hệ thống.

  - **Admin**: Người quản trị hệ thống với quyền quản lý toàn bộ dữ liệu như khóa học, giáo viên, và sinh viên.
  - **Teacher**: Giáo viên phụ trách chấm điểm, tạo bài tập, và theo dõi tiến độ sinh viên.
  - **Student**: Sinh viên với các chức năng liên quan đến tham gia khóa học, nộp bài tập và xem điểm.

- **UseCase Diagram và Mối quan hệ**:



Hình 2: UseCase Diagram của hệ thống.

  - **Admin**:
    * Tạo, xóa, và quản lý thông tin khóa học.
    * Quản lý thông tin giáo viên và sinh viên (thêm/xóa).
    * Gán sinh viên vào khóa học hoặc xóa khỏi khóa học.
    * Xem danh sách sinh viên trong mỗi khóa học.
  - **Teacher**:
    * Tạo bài tập cho sinh viên.
    * Chấm điểm và theo dõi trạng thái bài tập của sinh viên.
  - **Student**:
    * Xem danh sách các khóa học đã tham gia.
    * Nộp bài tập và xem thông tin về điểm số.

2.3 Phân tích biểu đồ

- **Quan hệ Include**: Một số chức năng phụ thuộc vào nhau, chẳng hạn, "Xóa khóa học" bao gồm việc "Xem tất cả các khóa học" để xác định khóa học cần xóa.

- **Luồng tương tác**: Admin có quyền kiểm soát toàn diện, Teacher thực hiện chức năng chuyên môn trong giảng dạy, và Student tập trung vào việc học tập và hoàn thành bài tập.

3 ERD DIAGRAM

3.1 Các thực thể chính

Các thực thể chính trong bài toán bao gồm:

- **Student**: Thông tin cá nhân; thông tin liên hệ; mã số sinh viên; khóa học;...
- **Teacher**: Thông tin cá nhân; thông tin liên hệ; mã giảng viên; khóa học;...
- **Course**: Thông tin về khóa học (mã khóa học, tên khóa học, giảng viên,...)
- **Assignment**: Thông tin về bài tập (mã bài tập, tên bài tập, hạn nộp, mô tả bài tập,....)
- **Assignment_details**: Thông tin chi tiết về bài tập (mã bài tập, mã số sinh viên, mã khóa học, mã thông tin điểm số, trạng thái bài tập...)
- **Grade_detail**: Thông tin chi tiết về điểm số của sinh viên (mã thông tin điểm số, điểm số thành phần, mã số sinh viên, mã khóa học...)
- **Student_course_detail**: Thông tin về quan hệ giữa sinh viên và khóa học (mã số sinh viên, mã khóa học, mã thông tin điểm,...)

3.2 Quan hệ giữa các thực thể

Quan hệ giữa các thực thể chính trong hệ thống được xác định như sau:

- **Admin - Teacher/Student/Course**:
  - Một admin có thể quản lý được nhiều sinh viên, nhiều giảng viên, nhiều khóa học.
  - Một sinh viên có thể được quản lý bởi một hoặc nhiều admin.
  - Một giảng viên có thể được quản lý bởi một hoặc nhiều admin.
  - Một khóa học có thể được quản lý bởi một hoặc nhiều admin.

  $\Rightarrow$ Quan hệ N-N.
- **Student - Course**:
  - Một sinh viên có thể tham gia nhiều khóa học.
  - Một khóa học có nhiều sinh viên tham gia.
  - Student phải được liên kết gián tiếp với Course thông qua Student_Course_Detail.

  $\Rightarrow$ Quan hệ N-N.
- **Teacher - Course**:
  - Một giảng viên có thể phụ trách nhiều khóa học.
  - Một khóa học chỉ do một giảng viên phụ trách.

⇒ Quan hệ 1-N.

- **Course - Assignment**:
  - Một khóa học có thể có nhiều bài tập.
  - Assignment phải được liên kết gián tiếp với Course thông qua Assignment_Details

  ⇒ Quan hệ 1-N.

- **Assignment - Assignment_Details**:
  - Một bài tập có thể có nhiều chi tiết bài tập (nhiều sinh viên thực hiện bài tập).
  - Một chi tiết bài tập thuộc về một bài tập cụ thể.
  - Assignment_Details đóng vai trò kết nối giữa Assignment, Student và Course.

  ⇒ Quan hệ 1-N.

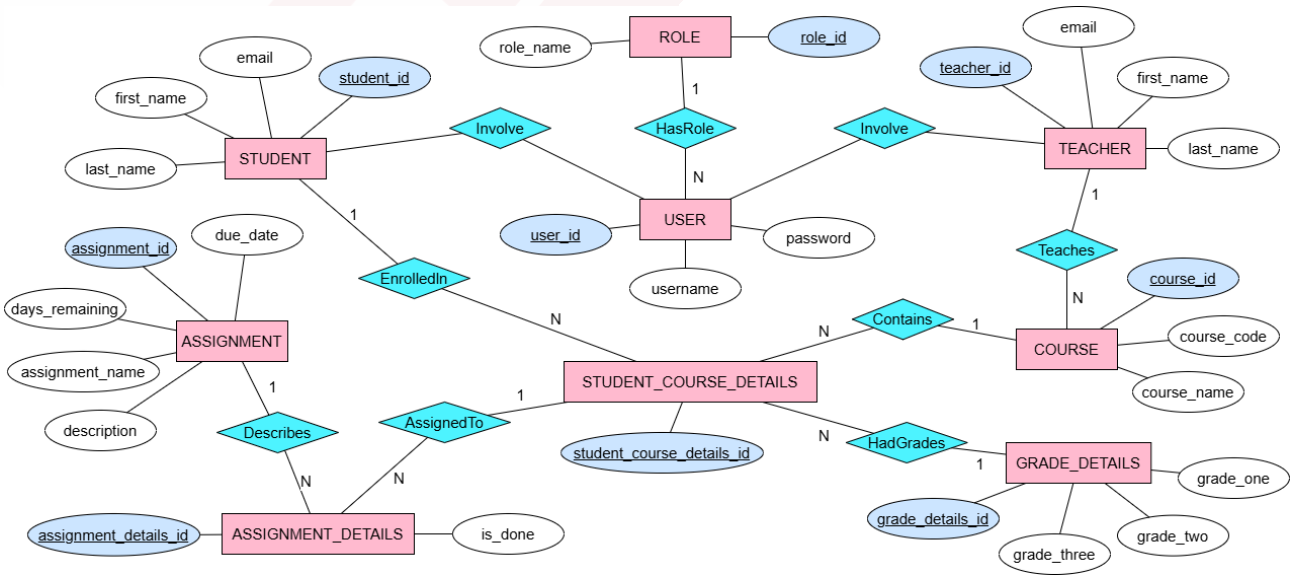
- **Student - Assignment_Details**:
  - Một sinh viên có thể thực hiện nhiều bài tập thông qua Assignment_Details.
  - Assignment_Details kết nối giữa một sinh viên và một bài tập trong bối cảnh một khóa học cụ thể

  ⇒ Quan hệ 1-N.

- **Student_Course_Detail**:
  - Kết nối giữa một sinh viên và một khóa học.
  - Bao gồm thông tin về điểm số và bài tập của sinh viên trong khóa học đó.

- **Grade_Detail**:
  - Một sinh viên có thể có nhiều điểm số cho từng khóa học.
  - Điểm số được gắn với Student_Course_Detail.

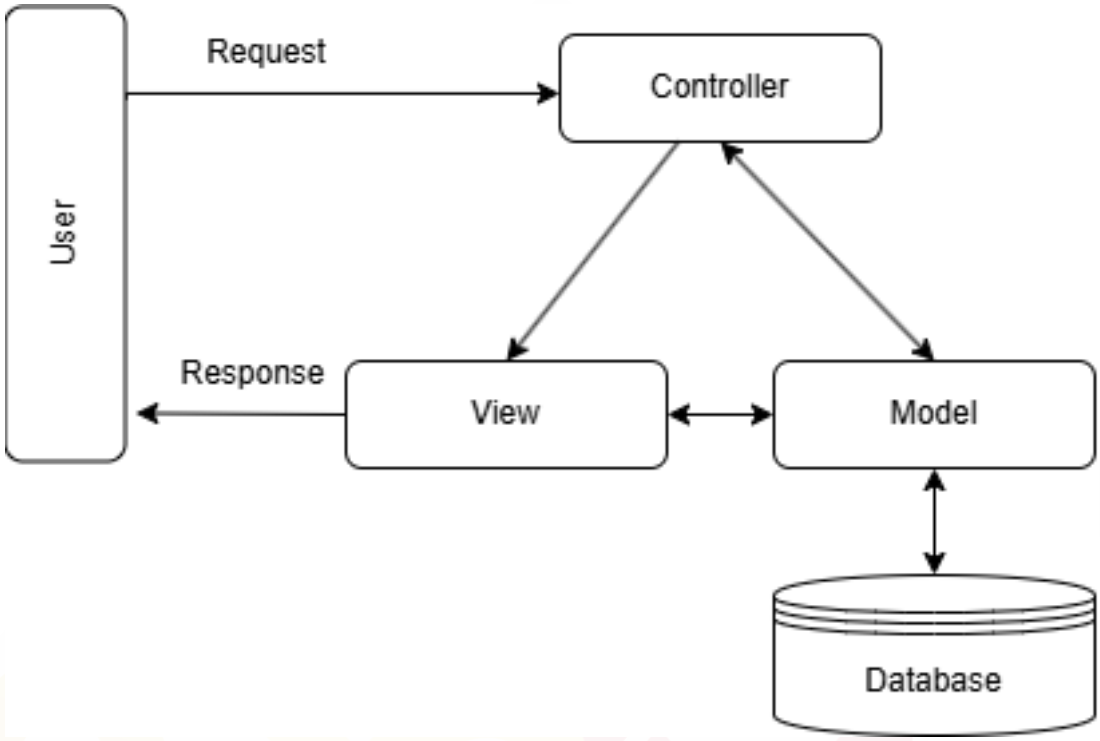
3.3 ERD Diagram



Hình 3: ERD Diagram của hệ thống.

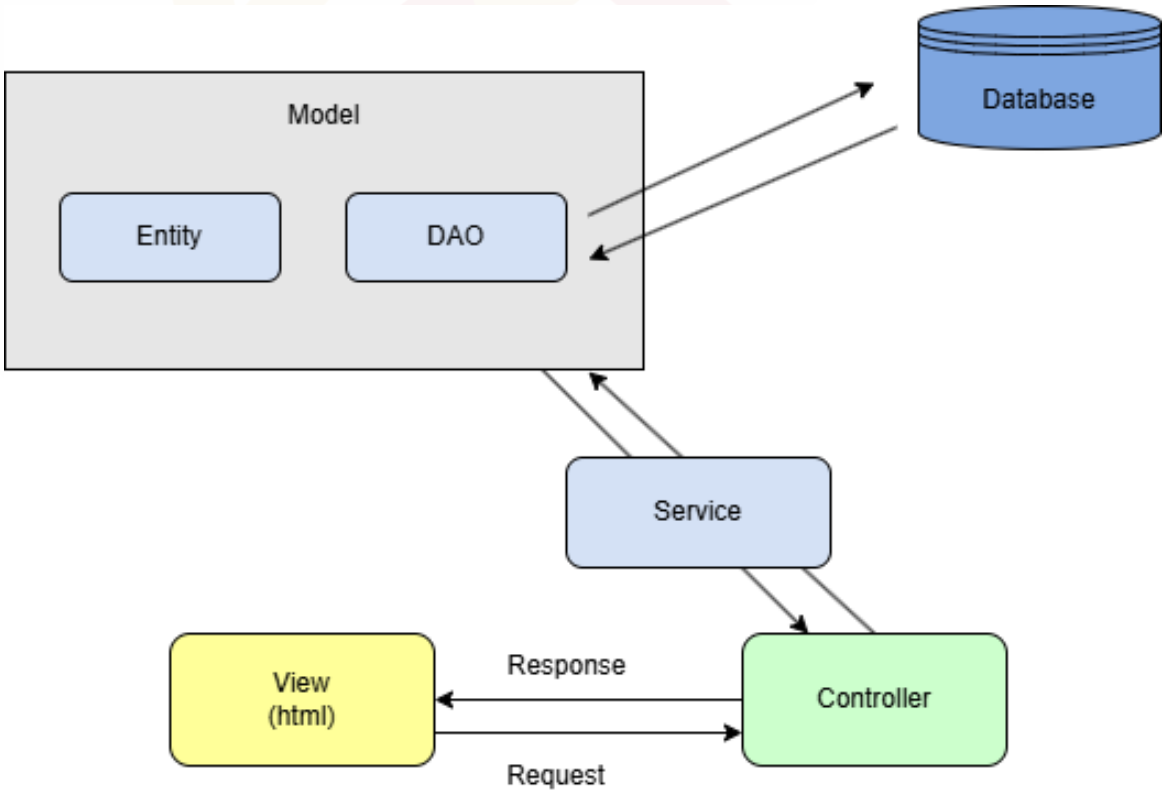
4 Kiến trúc hệ thống

4.1 Mô hình MVC cơ bản



Hình 4: Mô hình MVC cơ bản.

4.2 Mô hình MVC mở rộng



Hình 5: Mô hình MVC mở rộng.

4.3 Giải thích

- Người dùng (client) tương tác với giao diện người dùng (View) của ứng dụng, chẳng hạn như nhấn vào nút, gửi form hoặc yêu cầu một trang cụ thể.
- Khi người dùng thực hiện hành động này, một request (yêu cầu) sẽ được gửi từ trình duyệt đến Controller.
- Dựa trên yêu cầu của người dùng, Controller sẽ gọi các Service (Lớp dịch vụ), nơi chứa các logic nghiệp vụ. Service chịu trách nhiệm xử lý các yêu cầu phức tạp và tương tác với Model để truy xuất hoặc thay đổi dữ liệu từ Database bằng Repository.
- Service sẽ thực hiện các thao tác với Model, bao gồm:
  - Entity: Là đối tượng dữ liệu đại diện cho một bảng trong cơ sở dữ liệu. Entity chứa các thuộc tính cần thiết (cột trong bảng) và có thể có các phương thức để xử lý dữ liệu.
  - DAO (Data Access Object): Lớp này cung cấp các phương thức để truy xuất dữ liệu từ cơ sở dữ liệu. DAO giúp ẩn đi chi tiết thực hiện các thao tác dữ liệu.
- Sau khi nhận được dữ liệu, Controller chuyển dữ liệu vừa lấy được tới View để hiển thị lên trình duyệt cho người dùng.

5 CLASS DIAGRAM

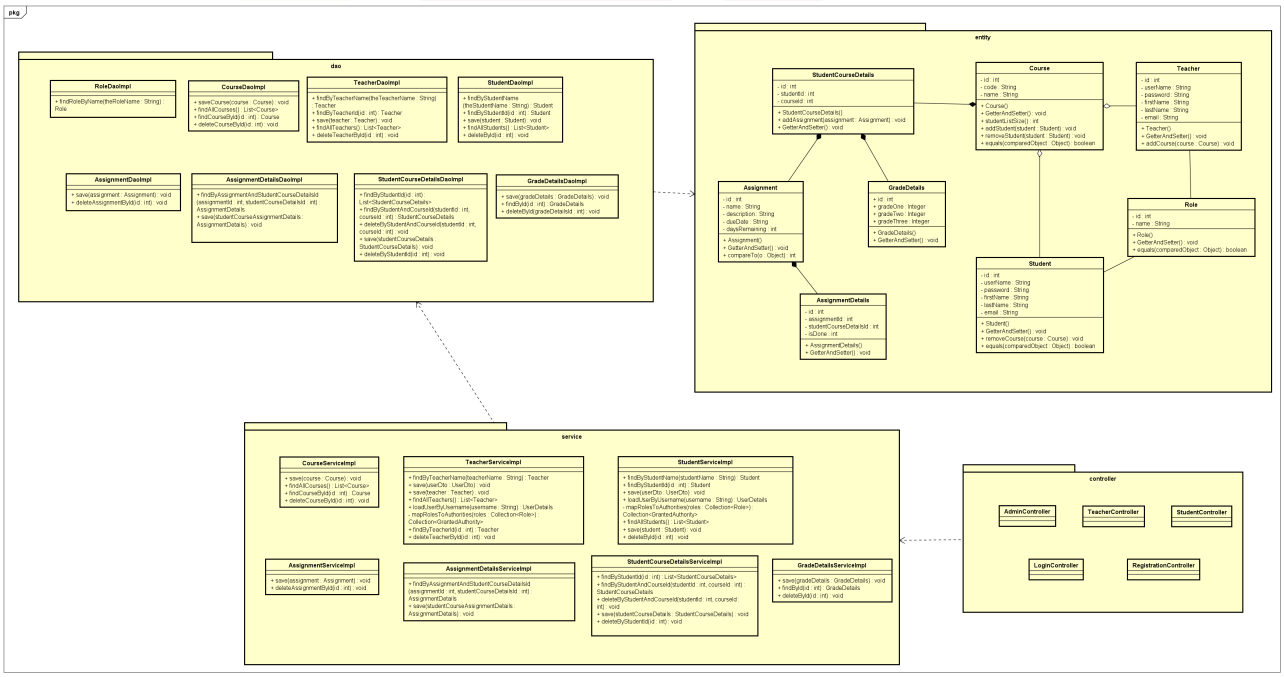
5.1 Các lớp và các phương thức chính

1. **Các lớp chính**:
   - **Role**:
     - id: Mã định danh vai trò.
     - name: Tên vai trò.
   - **Course**:
     - id: Mã khóa học.
     - name: Tên khóa học.
     - code: Code của khóa học.
   - **Student**:
     - id: Mã sinh viên.
     - userName: Tên đăng nhập của sinh viên.
     - password: Mật khẩu.
     - firstname, lastname: Tên sinh viên
     - email: Email sinh viên
   - **Teacher**:
     - id: Mã giảng viên.
     - userName: Tên đăng nhập của giảng viên.
     - password: Mật khẩu.
     - firstname, lastname: Tên giảng viên

  - email: Email giảng viên.
- **Assignment**:
  - id: Mã bài .
  - name, description: Thông tin bài tập.
  - dueDate: Ngày hạn.
  - daysRemaining: Số ngày còn lại
- **StudentCourseDetails, AssignmentDetails, GradeDetails**:
  - Các lớp phụ trợ để quản lý thông tin như điểm số, chi tiết bài tập và khóa học.

2. **Các phương thức chính**:
   - GetterAndSetter(): Phương thức getter và setter cho thuộc tính.
   - equals(Object obj): So sánh.
   - addStudent(Student student): Thêm sinh viên vào khóa học.
   - studentListSize(): Xem số lượng học sinh có trong danh sách.
   - removeStudent(Student student): Xóa sinh viên khỏi khóa học.
   - addCourse(Course course): Thêm khóa học.
   - addAssignment(assignment Assignment): Thêm bài tập
   - removeCourse(Course course): Xóa khóa học.
   - addCourse(Course course): Thêm khóa học.
   - compareTo(Object obj): So sánh bài tập theo thứ tự.

5.2 Class Diagram



Hình 6: Class Diagram của hệ thống.

5.3 Tổng quan hệ thống

Hệ thống quản lý sinh viên này được chia thành nhiều lớp với kiến trúc MVC (Model - View - Controller), bao gồm:

- Model: Quản lý dữ liệu và các lớp đại diện cho đối tượng thực.
- Service: Xử lý logic nghiệp vụ như thêm xóa, sửa, tìm kiếm.
- Controller: Quản lý luồng dữ liệu giữa người dùng và hệ thống, điều hướng đến giao diện mong muốn và gửi dữ liệu từ service lên giao diện, và lấy dữ liệu từ web về cho service xử lý.

5.4 Chi tiết các thành phần

1. **Controller (Giao tiếp người dùng)**:
   - **AdminController, TeacherController, StudentController**:
     - Điều phối dữ liệu giữa các lớp service và giao diện.
     - Xử lý các yêu cầu như thêm giáo viên, sinh viên hoặc quản lý bài tập.
   - **LoginController, RegistrationController**:
     - Xử lý logic đăng nhập và đăng ký.
2. **Service (Logic nghiệp vụ)**: Các lớp service xử lý logic nghiệp vụ.
   - **CourseServiceImpl, TeacherServiceImpl, StudentServiceImpl**:
     - Cung cấp các chức năng quản lý như thêm, sửa, xóa khóa học, giáo viên, sinh viên.
   - **AssignmentServiceImpl, GradeDetailsServiceImpl**:
     - Quản lý bài tập và điểm số.
3. **DAO (Data Access Object)**: Lớp DAO cung cấp các phương thức để giao tiếp với cơ sở dữ liệu.
   - **RoleDaoImpl, CourseDaoImpl, TeacherDaoImpl, StudentDaoImpl**:
     - Các phương thức CRUD cơ bản như findById, save, deleteById.
   - **AssignmentDetailsDaoImpl, GradeDetailsDaoImpl, StudentCourseDetailsDaoImpl**:
     - Quản lý các thực thể liên quan đến điểm số, bài tập và chi tiết khóa học của sinh viên.

6 Các tính chất OOP

- **Tính đóng gói**: Các phương thức getter, setter
- **Tính kế thừa**: Lớp SecurityConfig kế thừa từ lớp WebSecurityConfigurerAdapter.
- **Tính đa hình**: Implement method trong DAO, Service
- **Tính trừu tượng**: Interface

7 Các công nghệ sử dụng

Các công nghệ được sử dụng trong dự án:

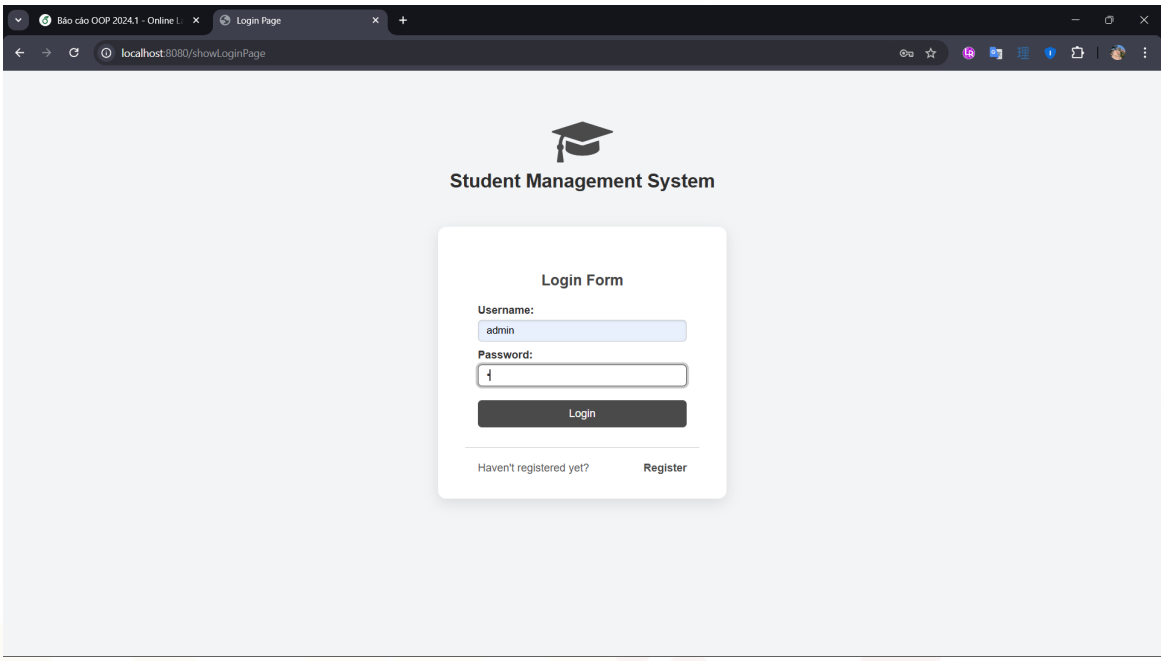
- **Ngôn ngữ**: Java
- **Framework**: Spring boot
- **Database**: MySQL

Hình 7: Các công nghệ sử dụng.

8 Demo

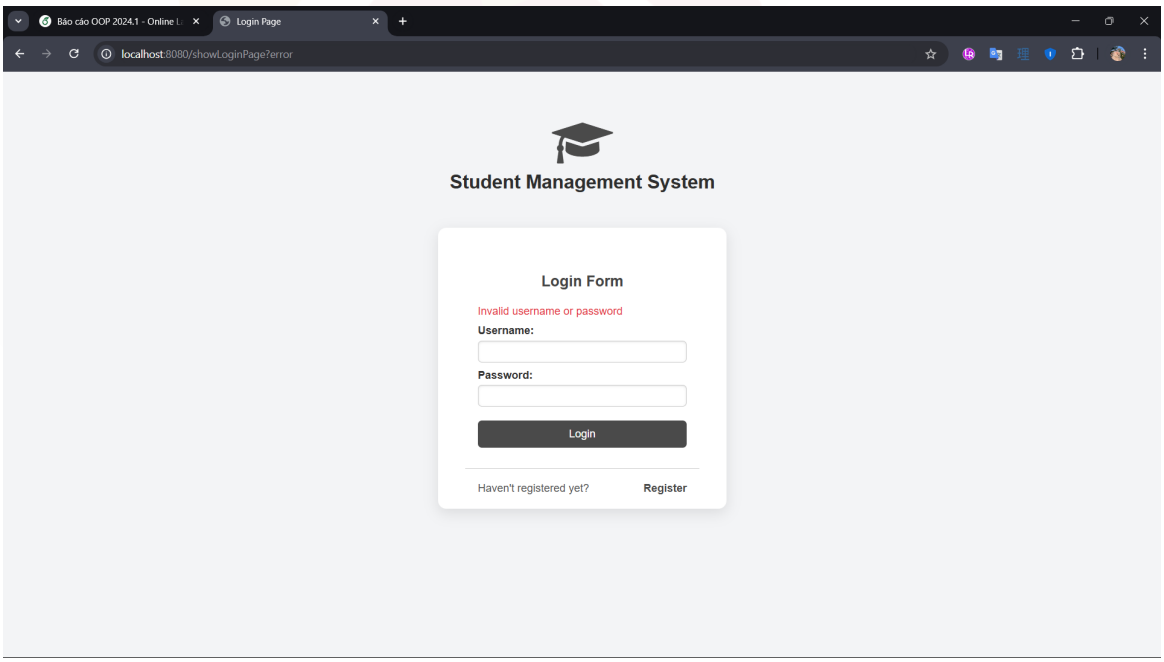
1. **Login Logout**:

   Giao diện Login gồm tên tài khoản, mật khẩu. Có thể chuyển hướng sang Register nếu chưa có tài khoản.



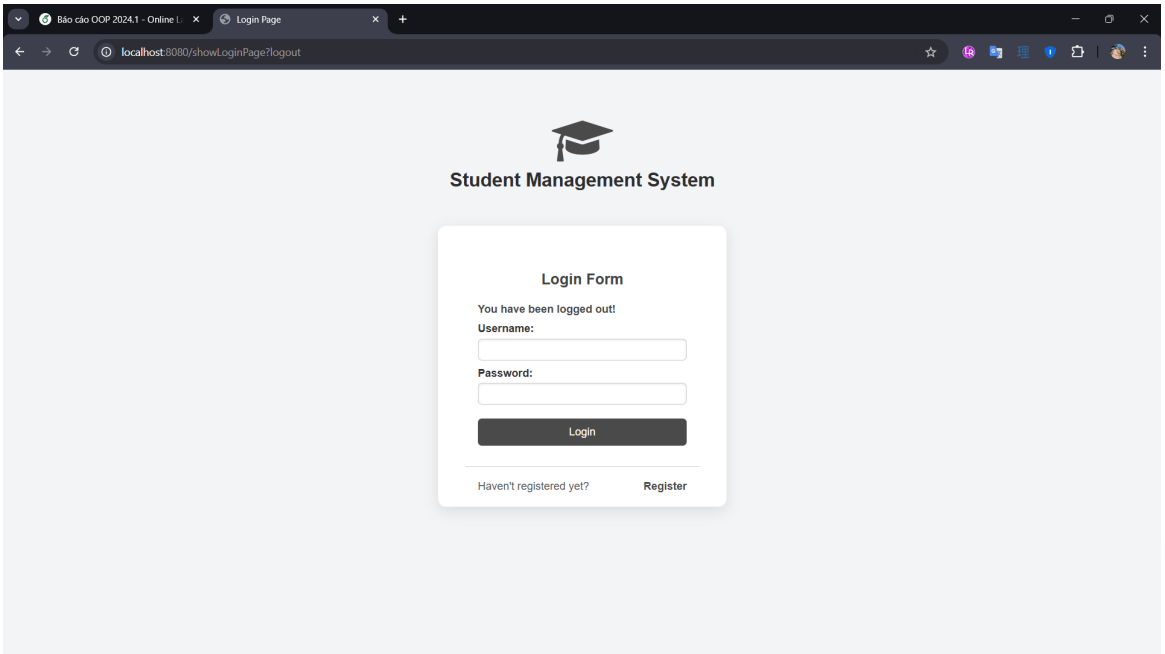
Hình 8: Login

Khi người dùng không nhập đúng tài khoản, mật khẩu, màn hình sẽ hiện như sau:



Hình 9: Login Error

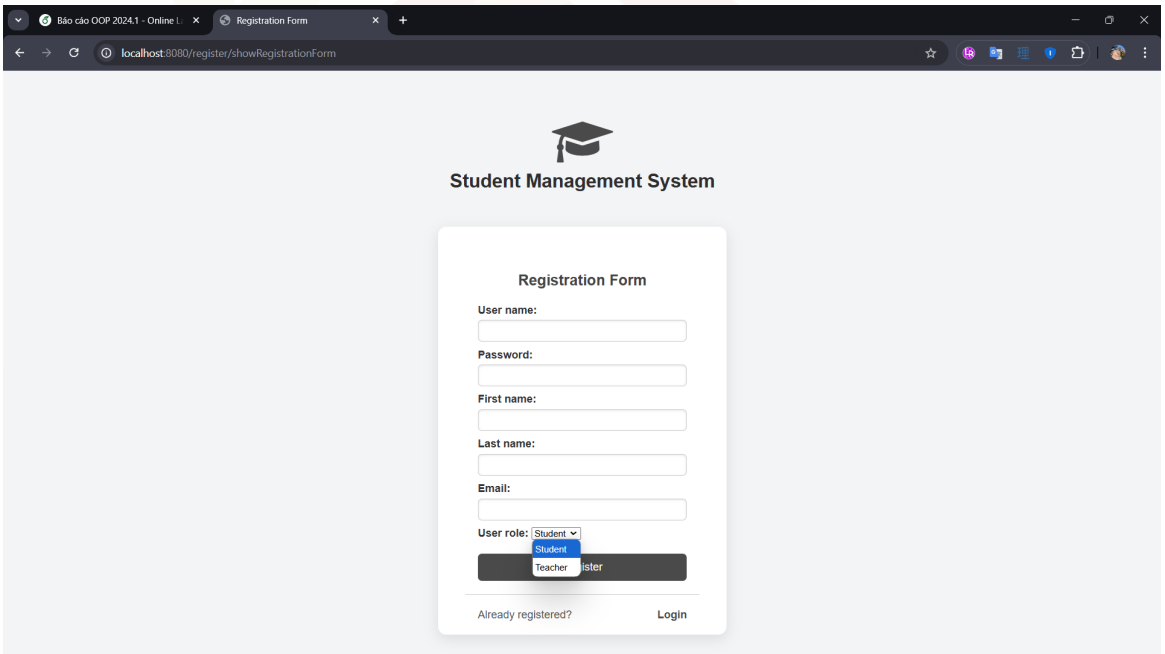
Khi người dùng Logout thành công, màn hình sẽ chuyển về Login như sau:



Hình 10: Logout Successful

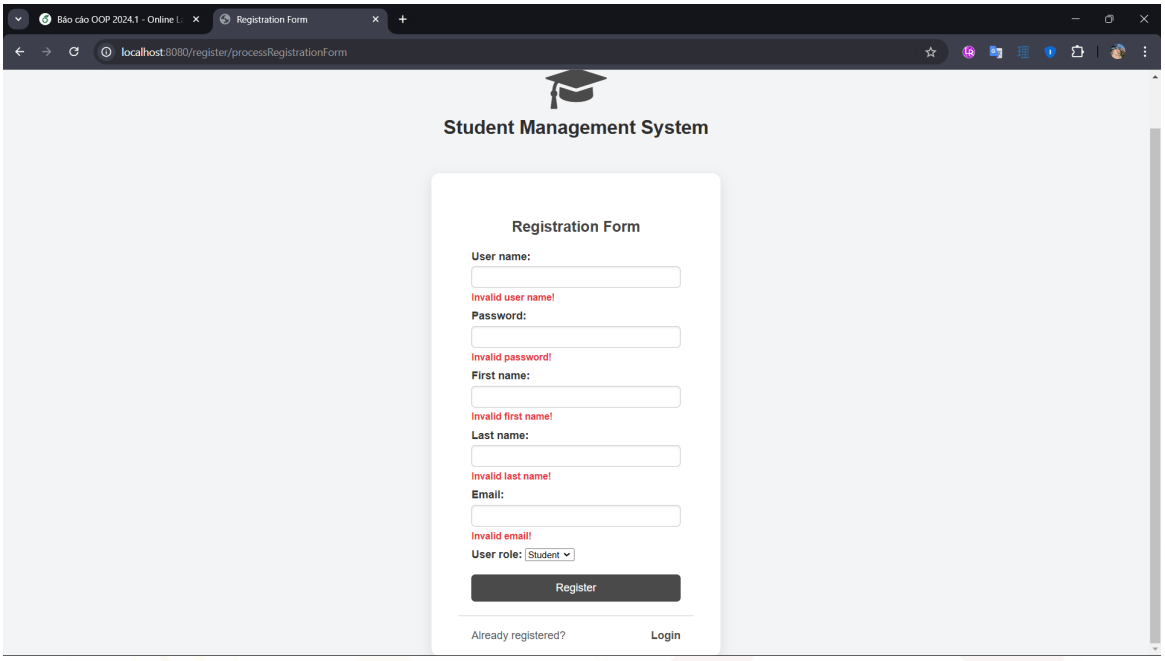
2. **Register**:

   Giao diện Register gồm tên tài khoản, mật khẩu, First Name, Last Name, Email và Role. Người dùng có thể chọn Role là Student hoặc Teacher.
   Nếu người dùng đã có tài khoản có thể chuyển về trang Login.



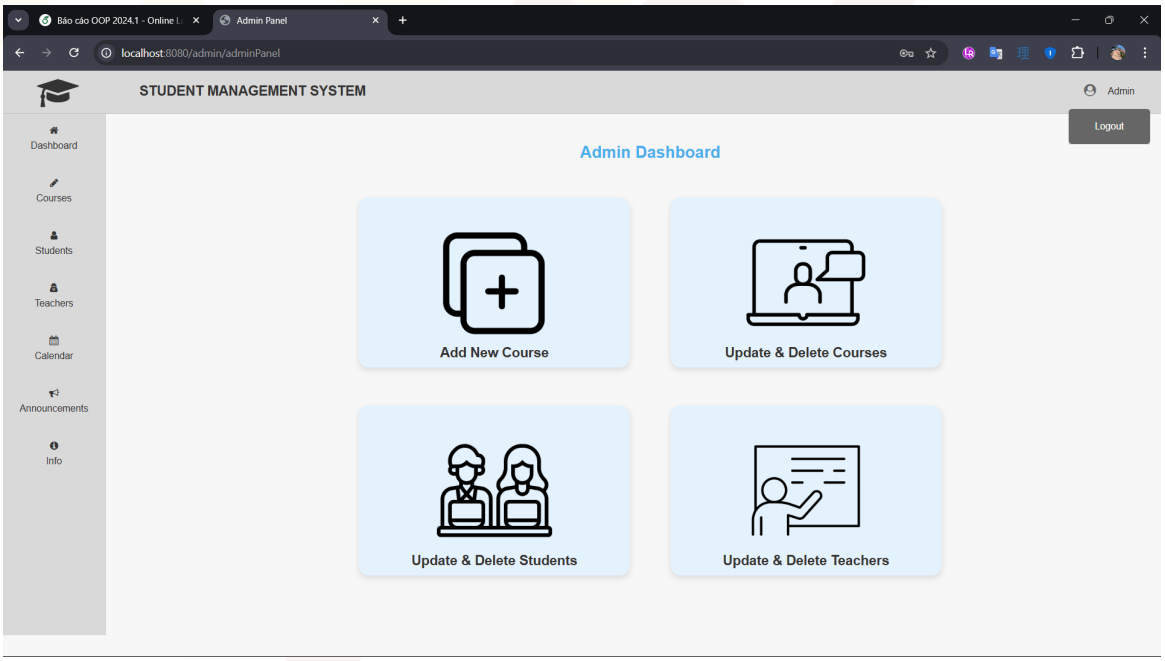
Hình 11: Register

Khi nhập sai các thông tin, màn hình Register sẽ hiển thị như sau:

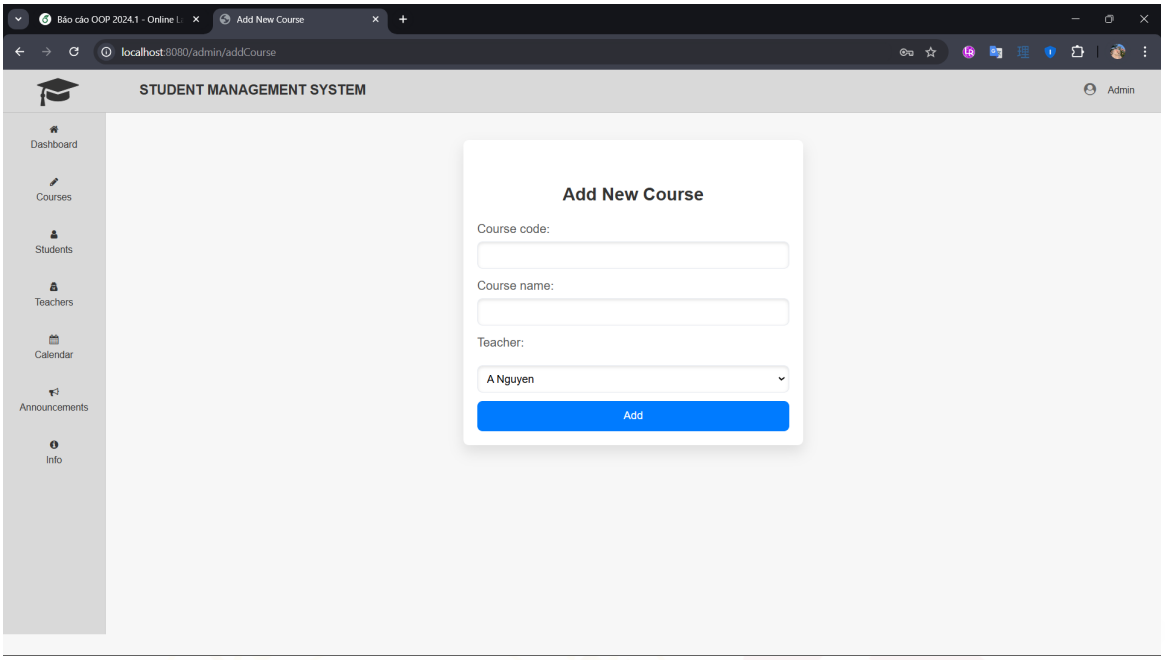


Hình 12: Register Error

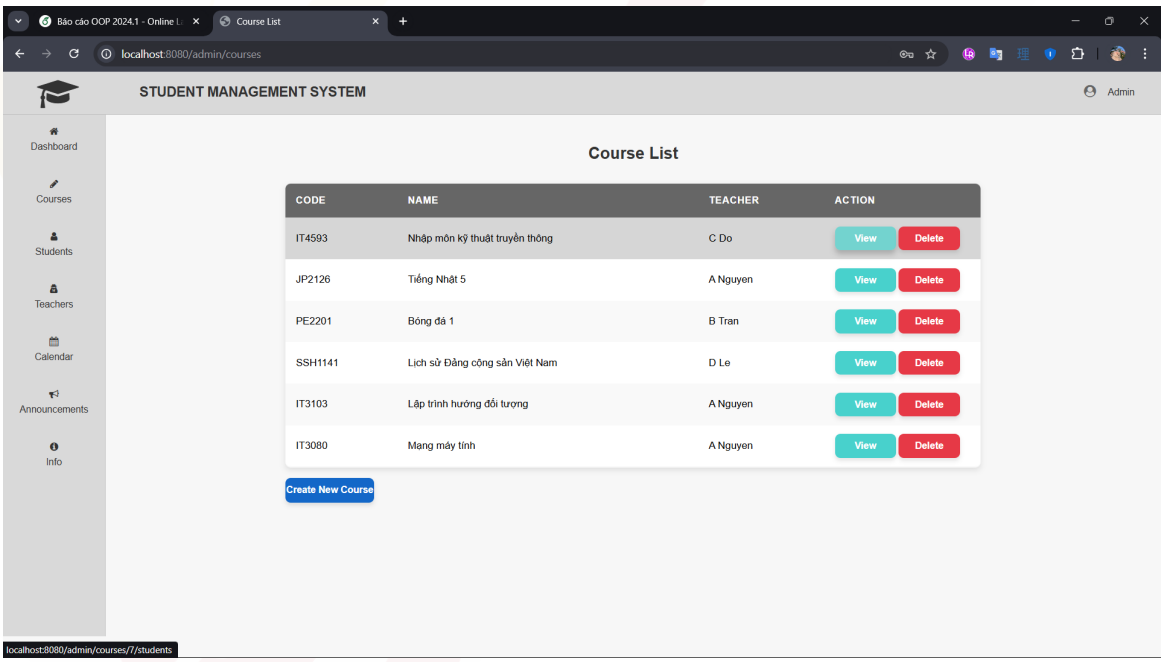
3. **Admin**:



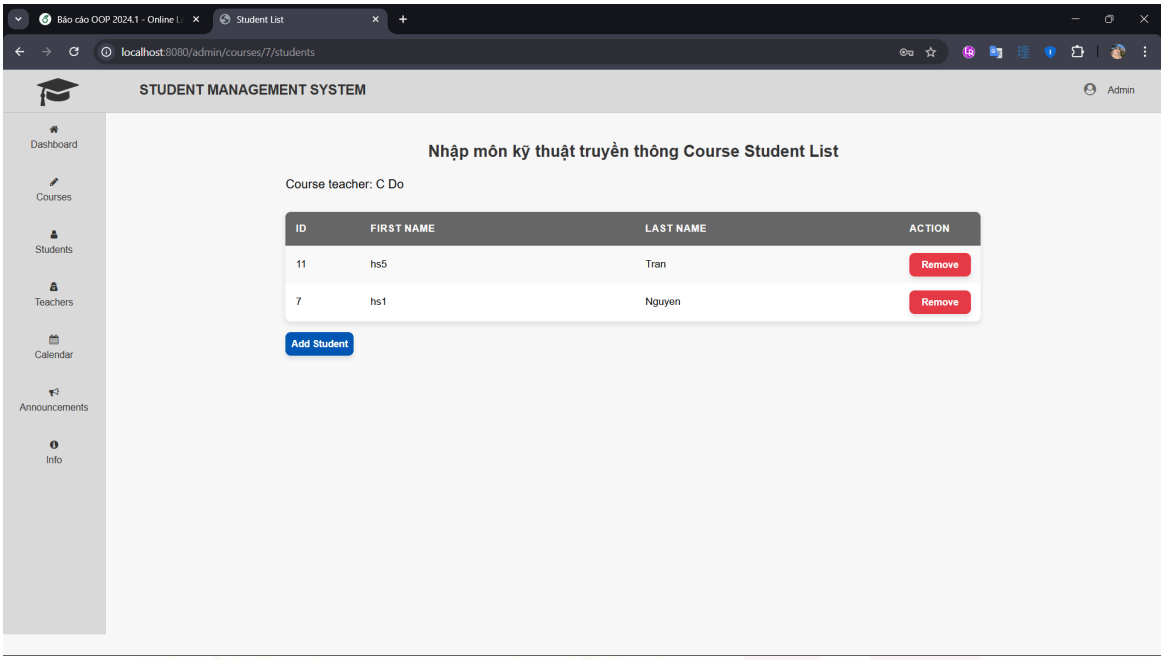
Hình 13: Admin Panel



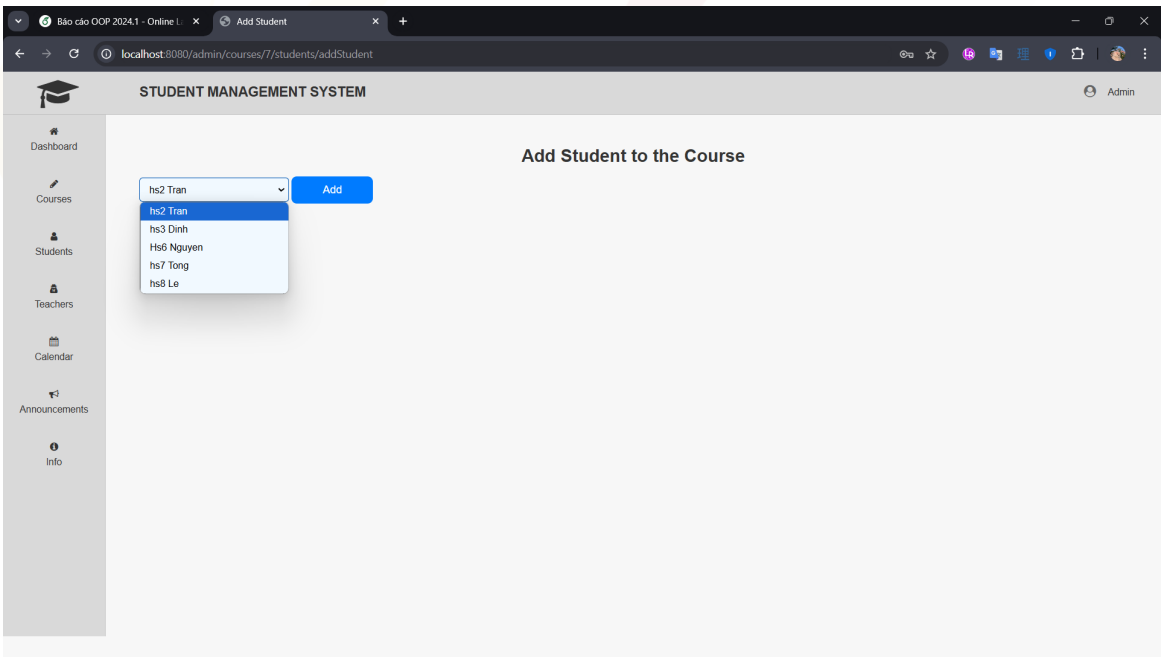
Hình 14: Admin Add new Course



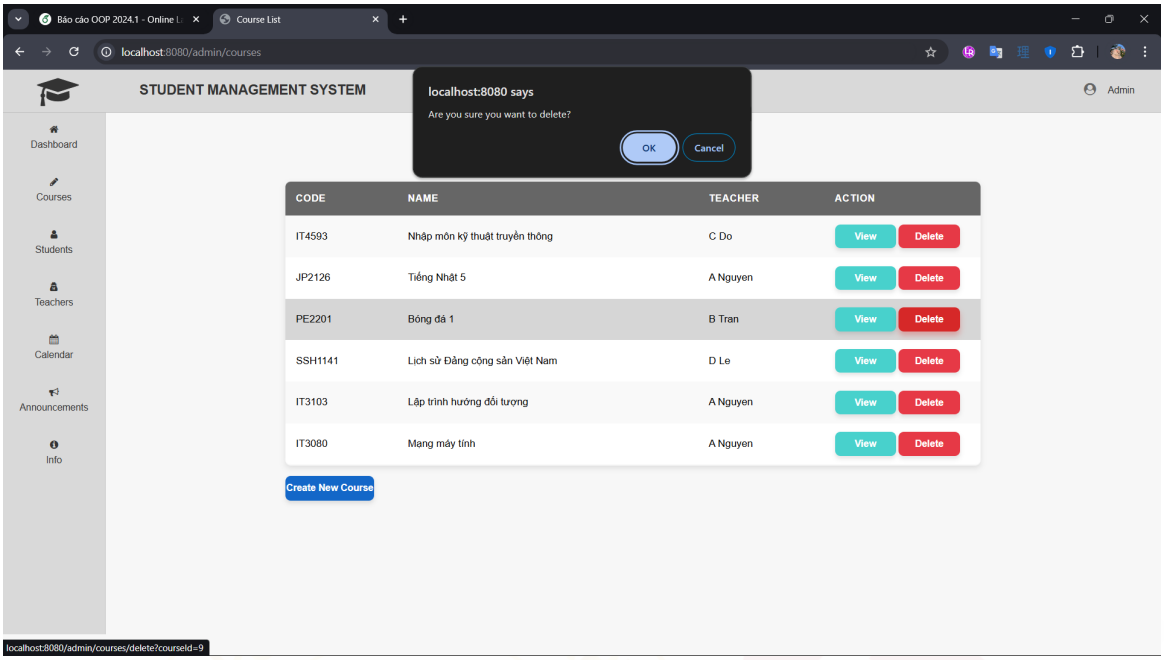
Hình 15: Admin Course List



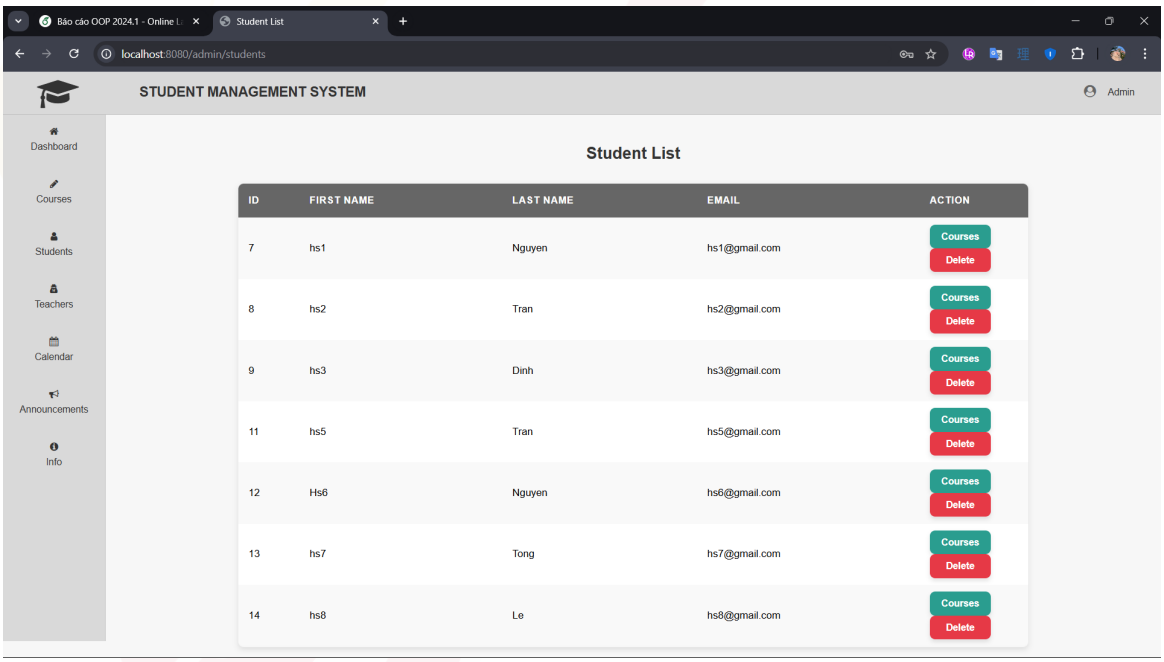
Hình 16: Admin Course Details



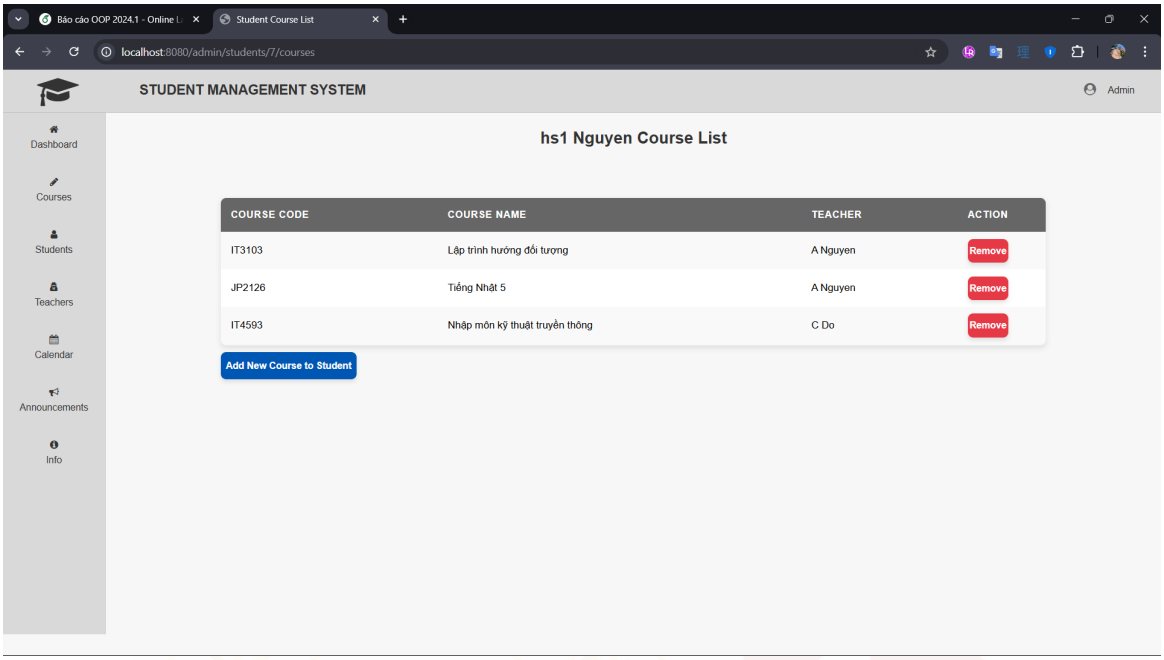
Hình 17: Admin Add Student To Course



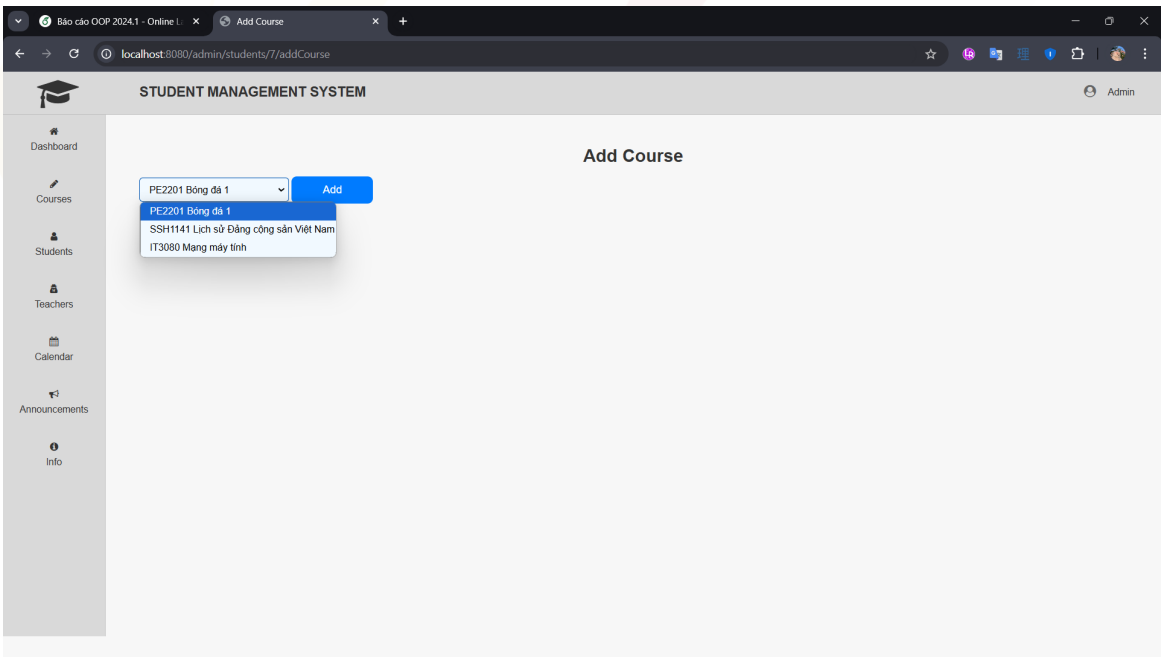
Hình 18: Admin Remove Course



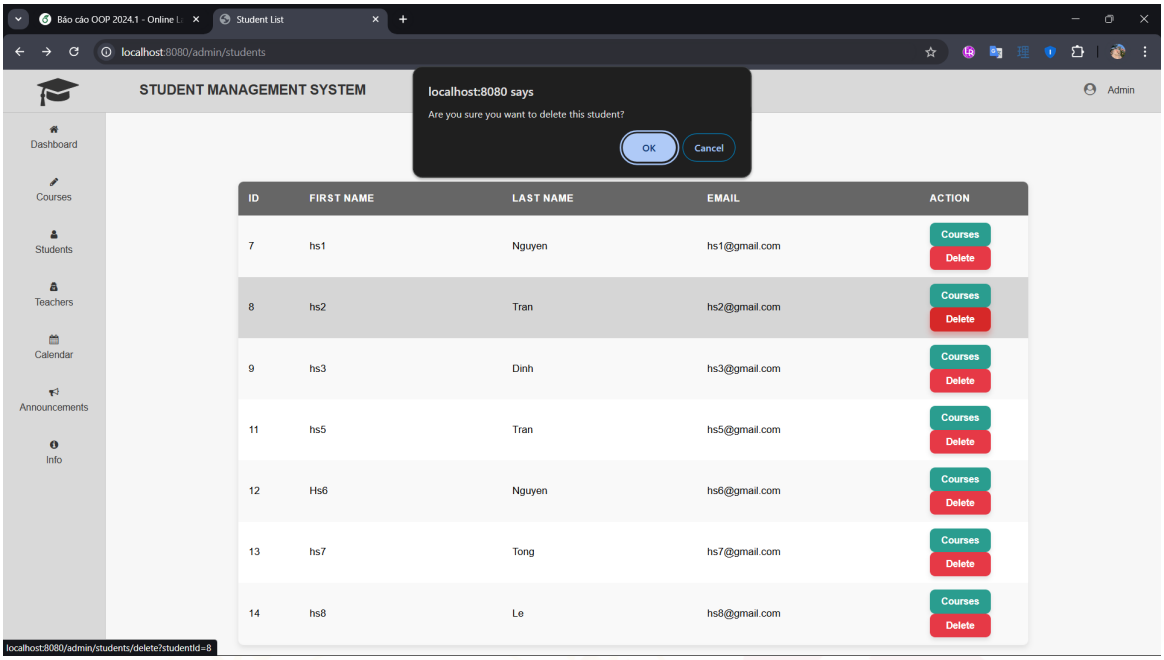
Hình 19: Admin Student List



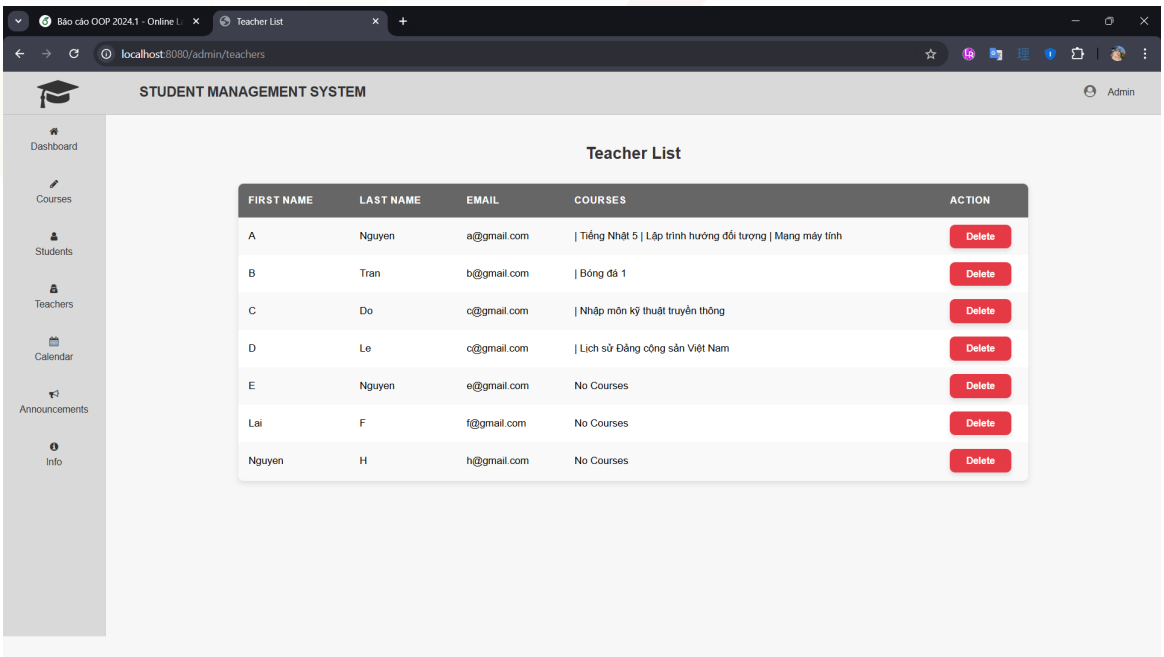
Hình 20: Admin Student Course List



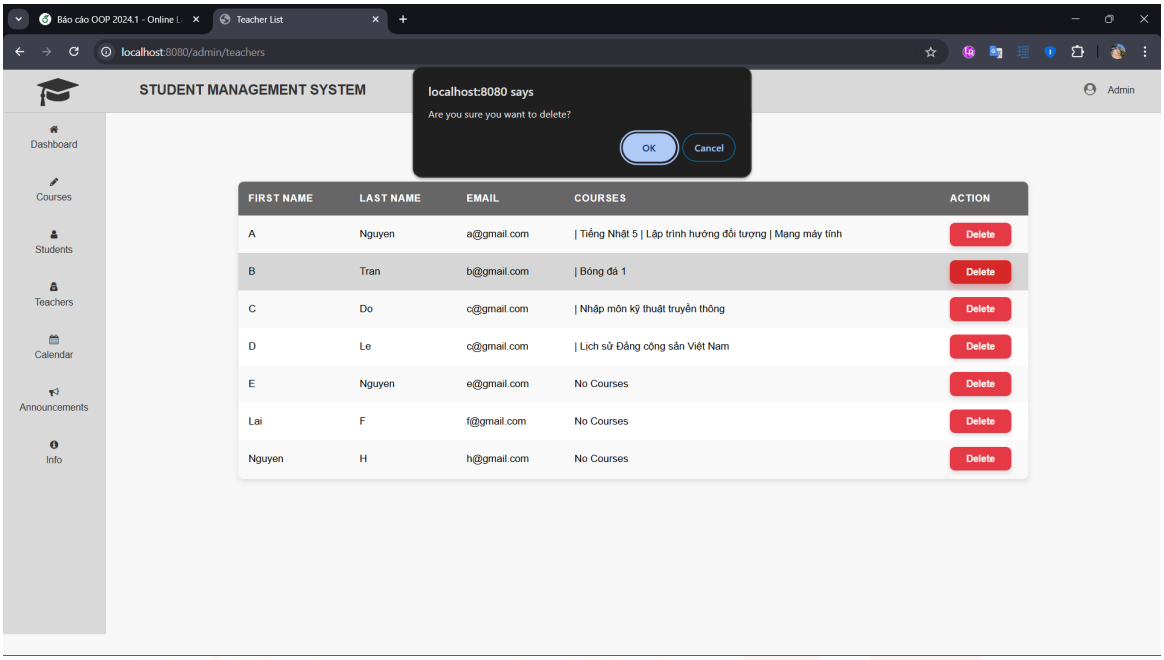
Hình 21: Admin Add Course To Student



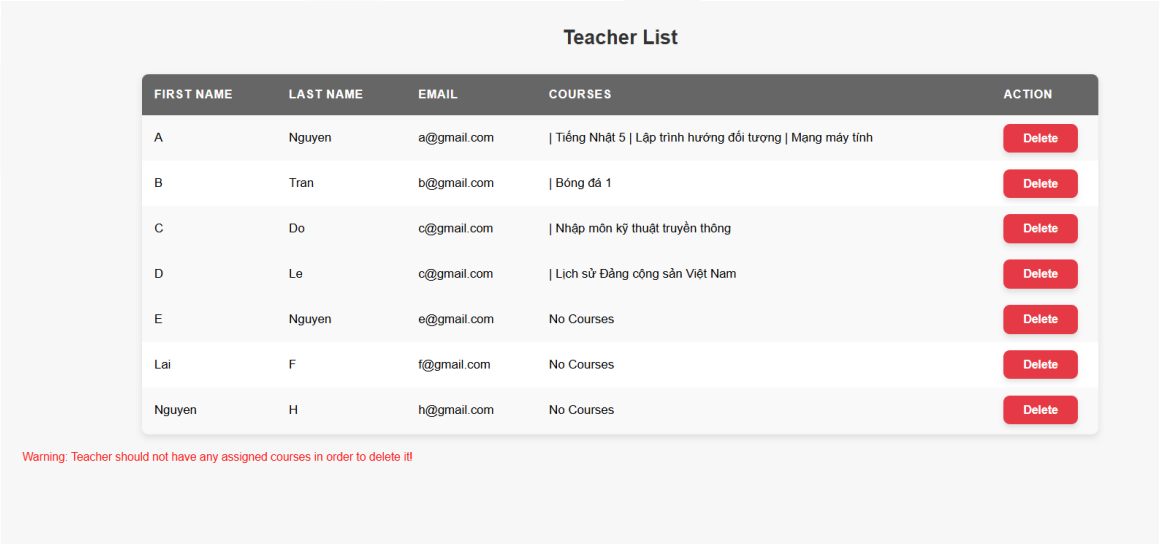
Hình 22: Admin Remove Student



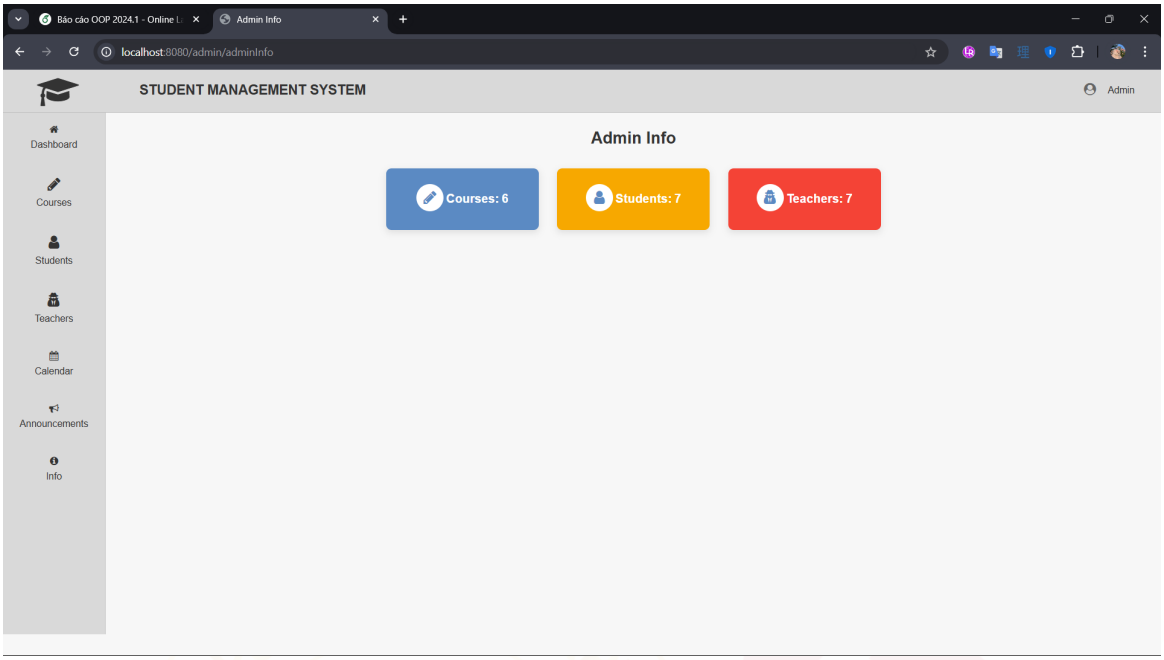
Hình 23: Admin Teacher List



Hình 24: Admin Remove Teacher

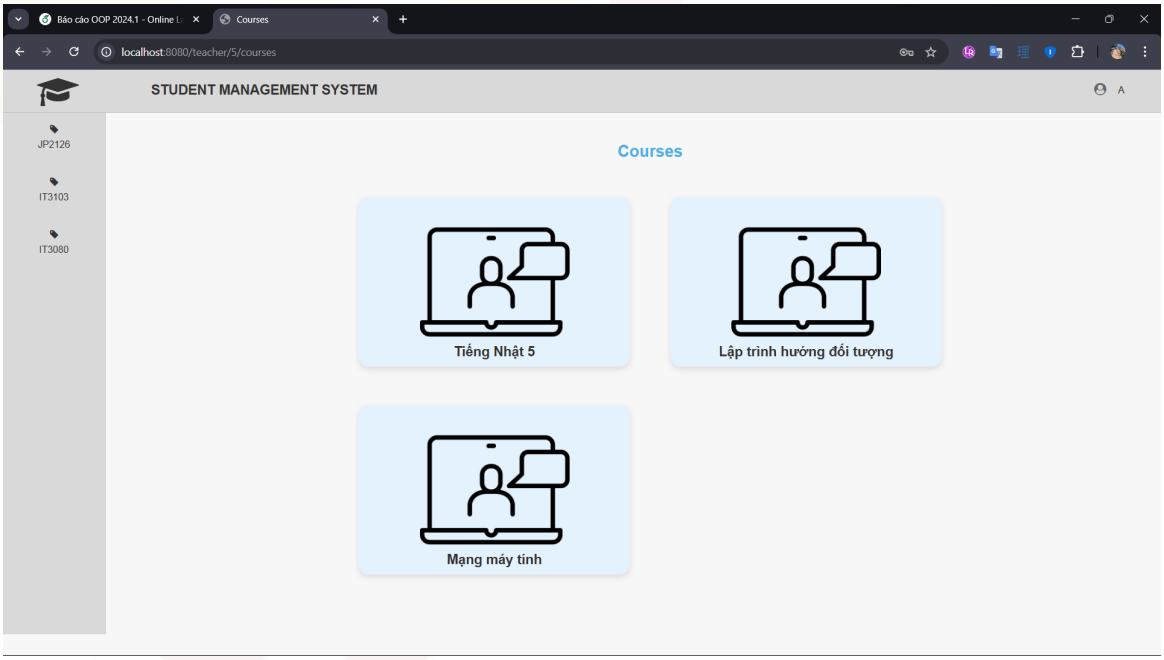


Hình 25: Admin Remove Teacher Error

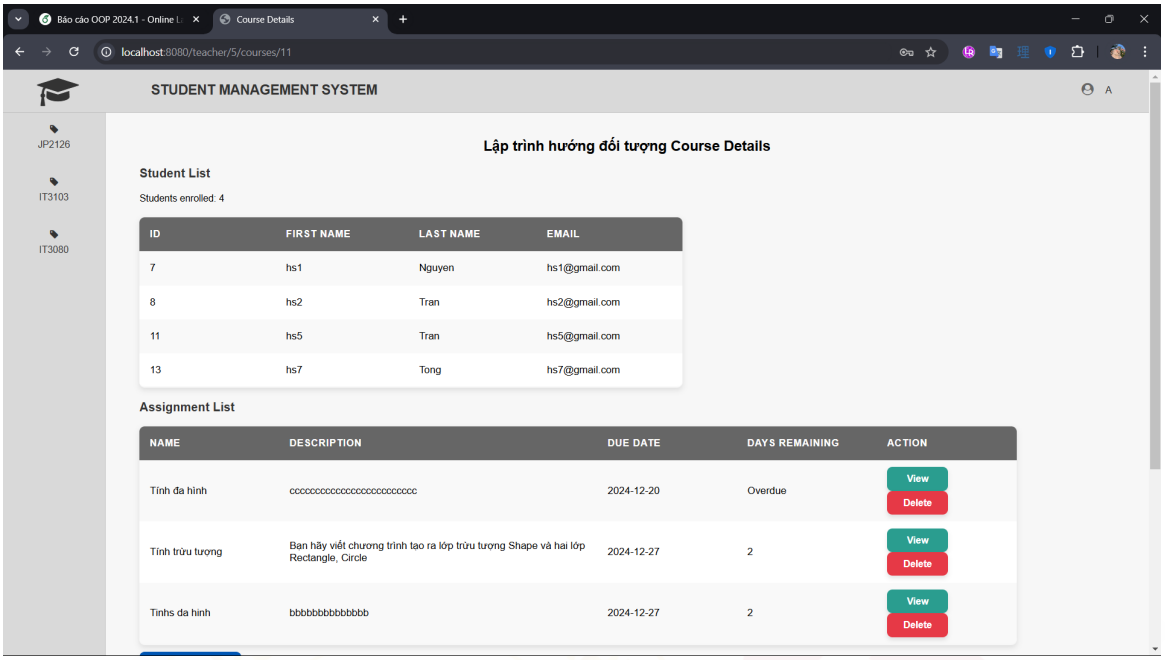


Hình 26: Information of System

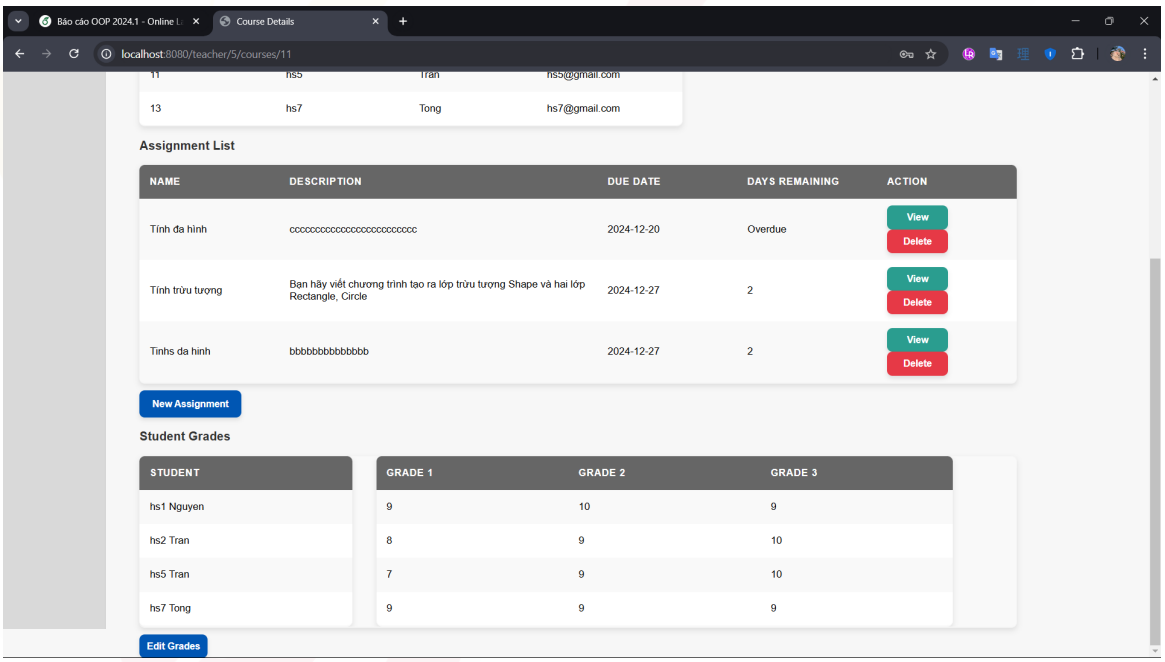
4. **Teacher**:



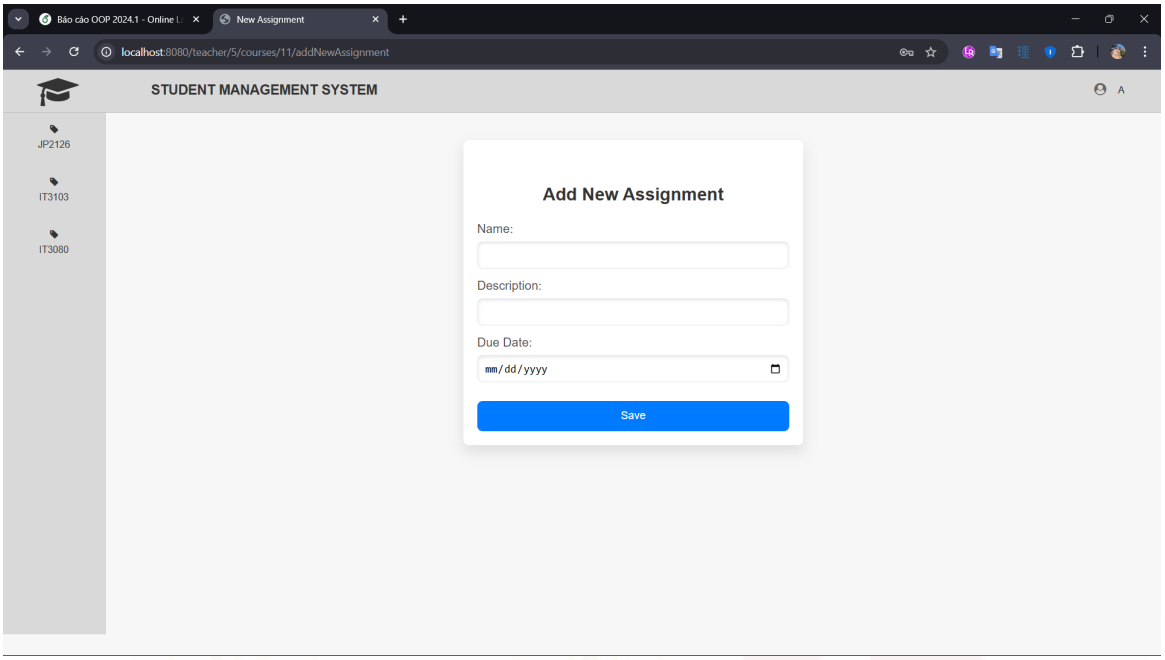
Hình 27: Teacher Course List



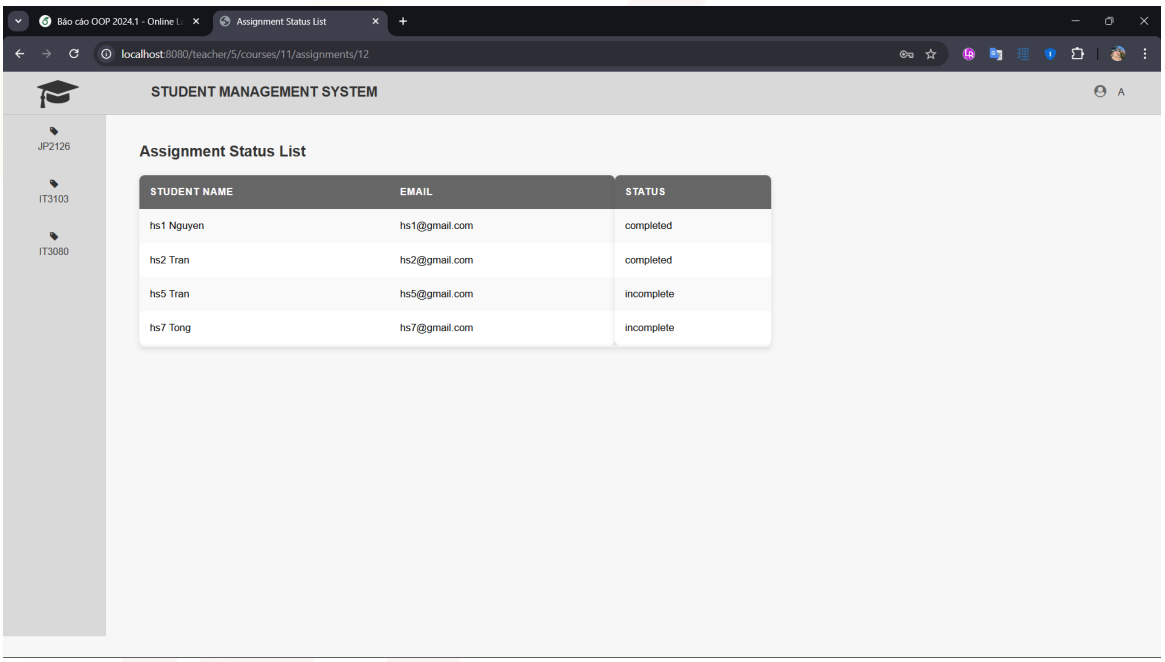
Hình 28: Teacher Course Details 1



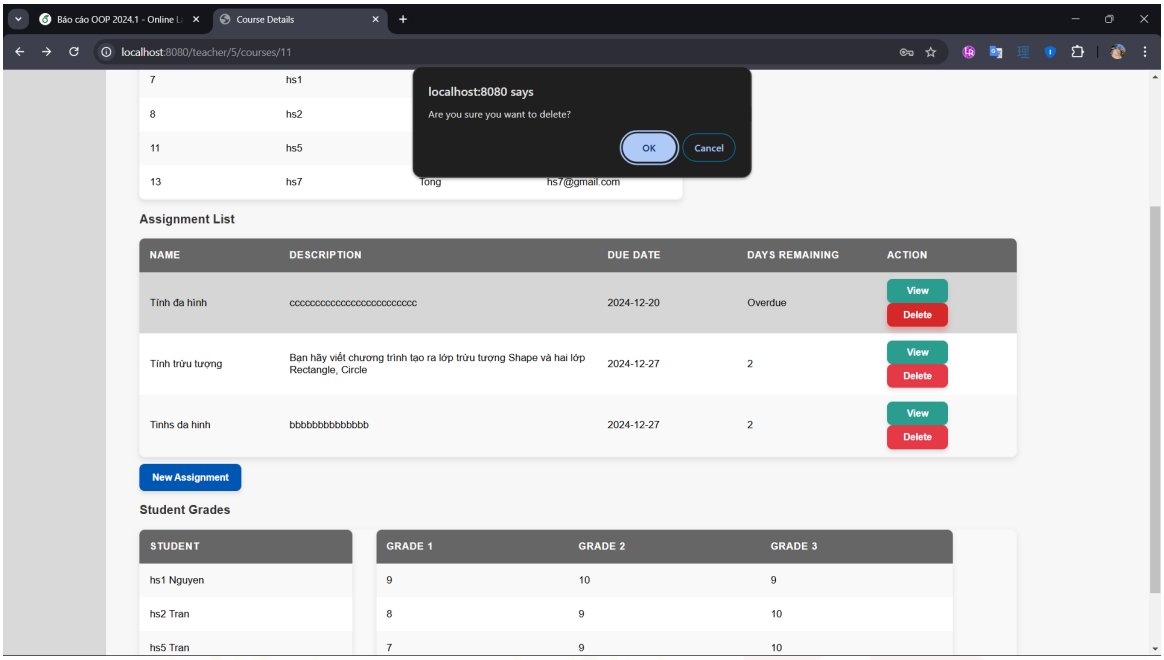
Hình 29: Teacher Course Details 2



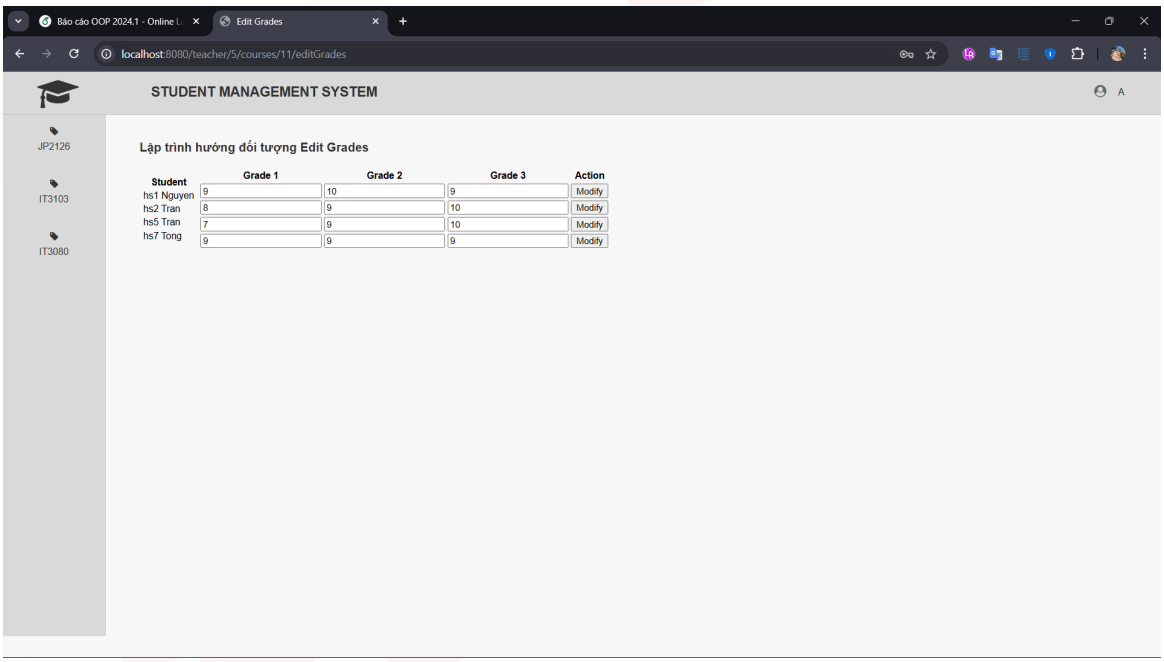
Hình 30: Teacher Add Assignment



Hình 31: Teacher Assignment Details

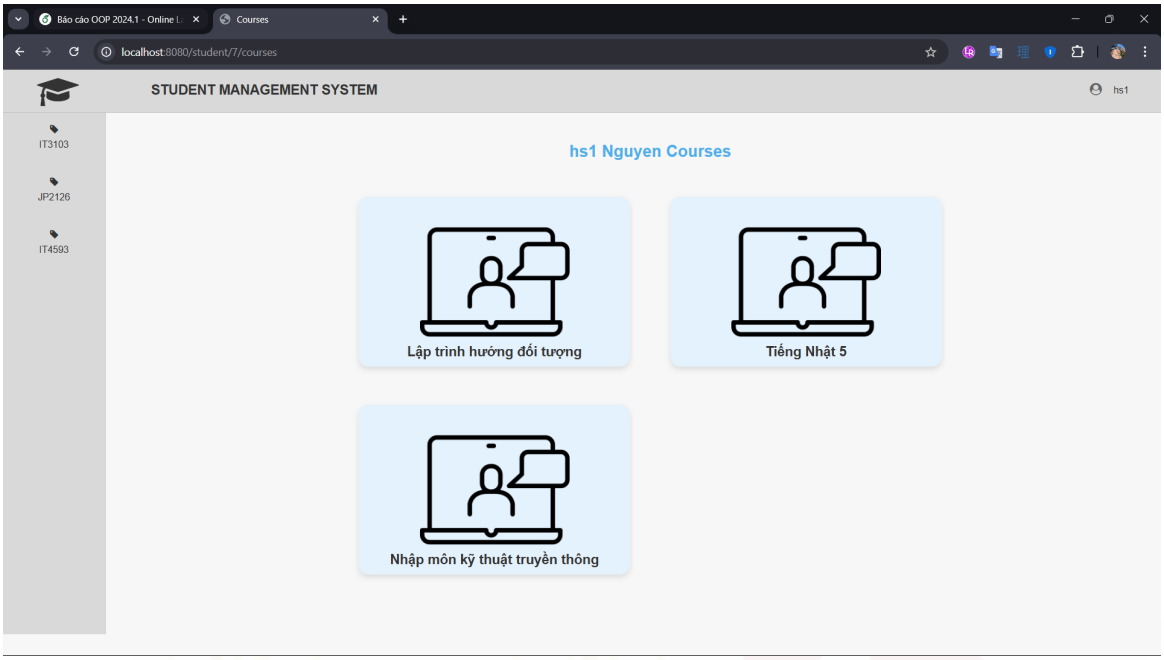


Hình 32: Teacher Remove Assignment

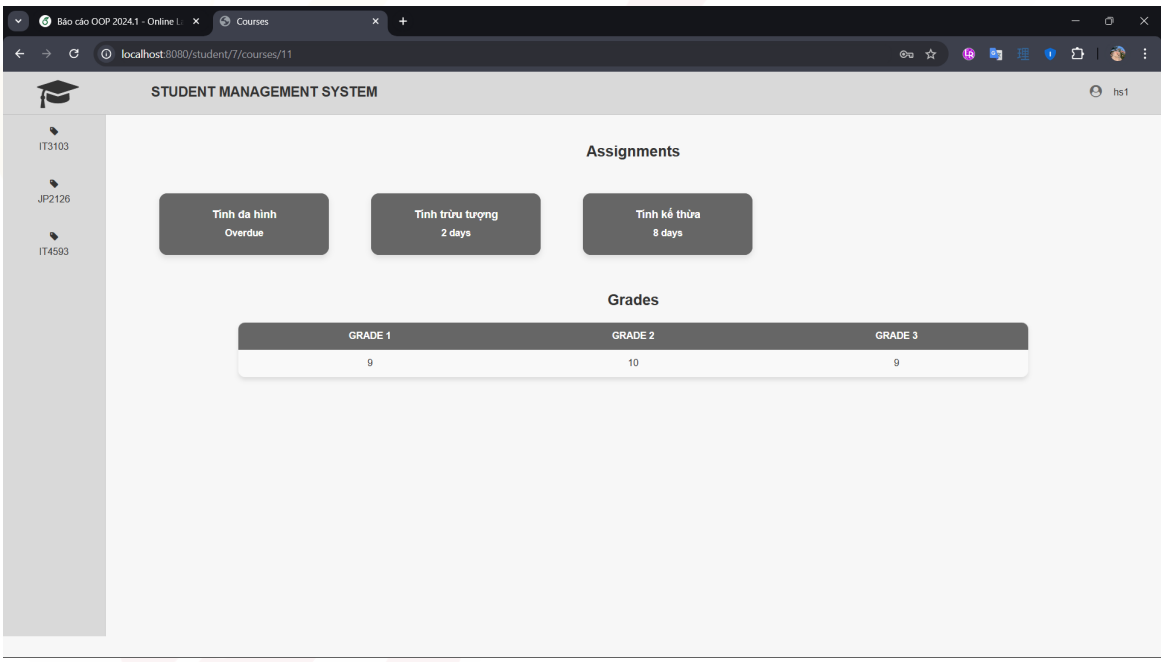


Hình 33: Teacher Edit Grade

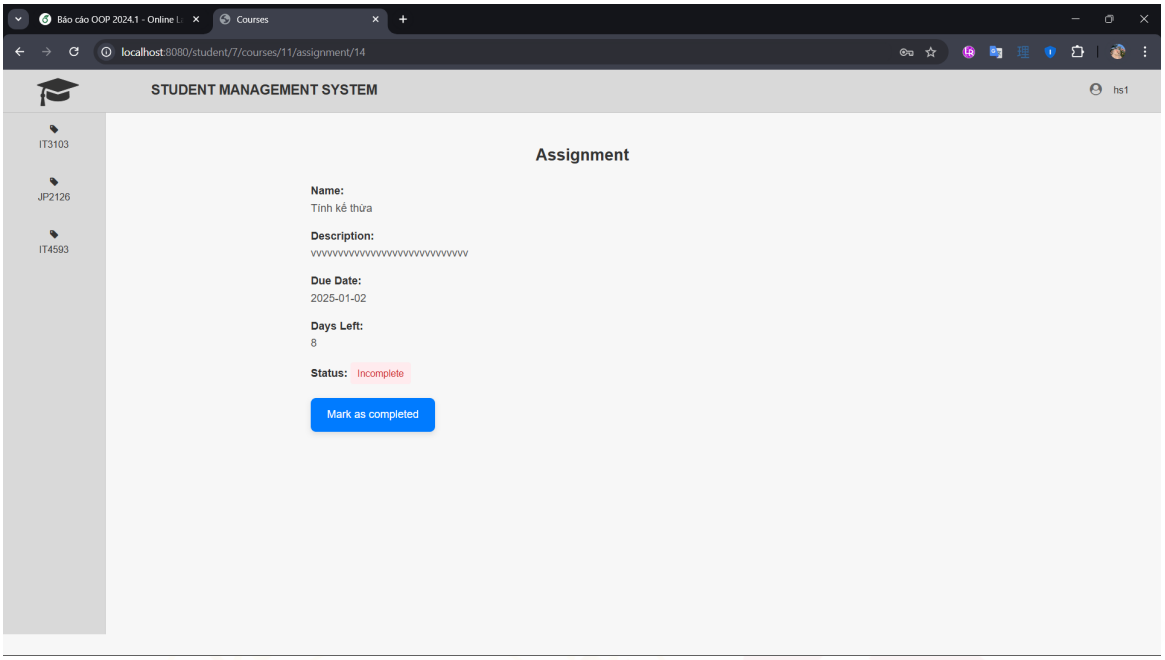
5. **Student**:



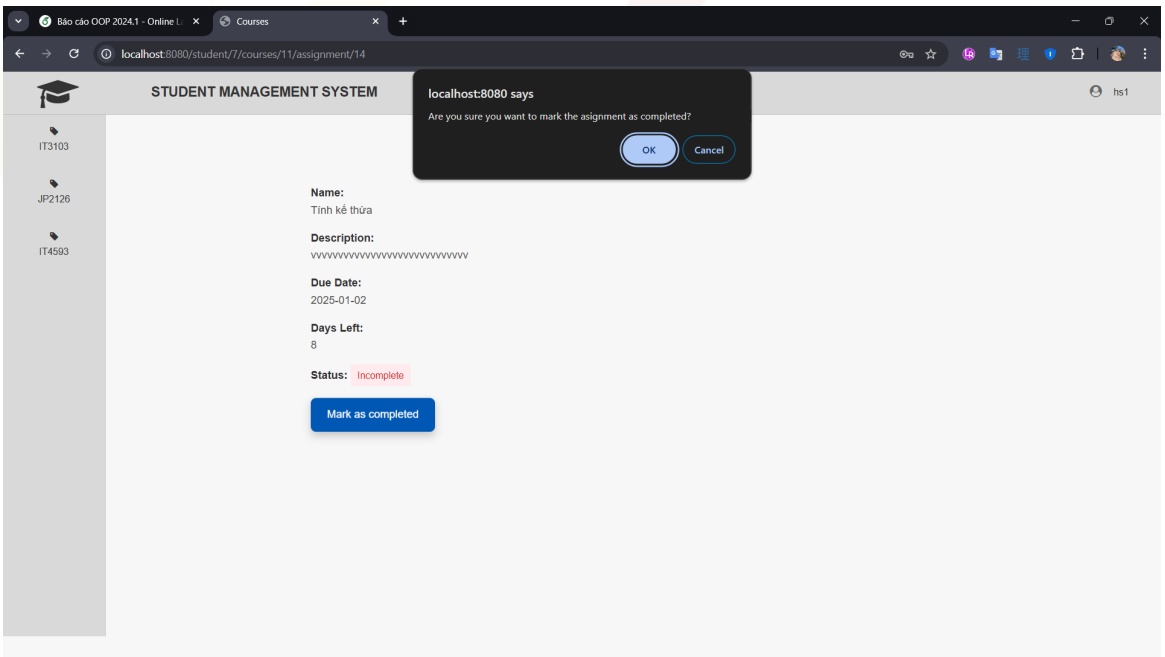
Hình 34: Student Course List



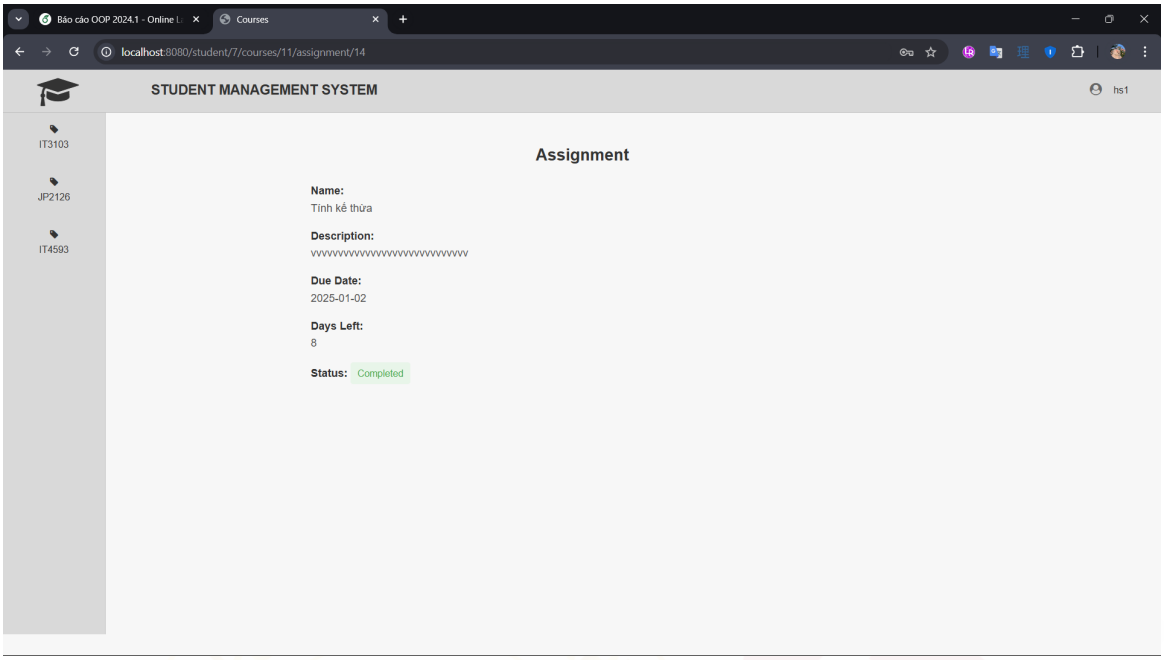
Hình 35: Student Course Details



Hình 36: Student Assignment Details



Hình 37: Student Submit Assignment



Hình 38: Student Submit Assignment Successful

6. **Other**:

Hình 39: Access Denied

9 Thu hoạch

Bài tập lớn về dự án quản lý sinh viên không chỉ giúp nhóm củng cố kiến thức lý thuyết mà còn rèn luyện các kỹ năng thực tiễn cần thiết cho công việc sau này. Nhóm tin rằng những kinh nghiệm quý báu từ dự án này sẽ là hành trang quan trọng cho sự nghiệp lập trình và phát triển của mỗi thành viên trong nhóm.

10 Tài liệu tham khảo

References

[1] Lê Quốc Tuấn, (2021), Tìm hiểu về cách thiết kế Class Diagram.

[2] Thống PM, (2022), Tất tần tật về mô hình MVC.

[3] Vu Tong, (2021), Học Spring Boot bắt đầu từ đâu?