

TRƯỜNG ĐẠI HỌC PHENIKAA

KHOA CÔNG NGHỆ THÔNG TIN



**Báo Cáo Dự Án Học Phần Phân Tích Và Thiết Kế Phần Mềm**

**Đề tài: PHÁT TRIỂN PHẦN MỀM  
QUẢN LÝ SINH VIÊN**

**Nhóm 03 – Lớp N06**

**Thành viên:**

- Nguyễn Cao Chiến – 22010014
- Nguyễn Đức Anh – 22010353
- Phạm Hoàng Anh – 22010477
- Nguyễn Hoàng Sơn – 22010249

**Giảng viên hướng dẫn: ThS. Vũ Quang Dũng**

**Hà Nội, tháng 3 năm 2025**

## BẢNG PHÂN CÔNG CÔNG VIỆC

<b>STT</b>	<b>Mã SV</b>	<b>Họ tên</b>	<b>Công việc</b>	<b>Tỉ lệ đóng góp</b>
1	22010014	Nguyễn Cao Chiến	<ul style="list-style-type: none"> <li>- Xác định yêu cầu hệ thống.</li> <li>- Phân tích đối tượng sử dụng, khách hàng, nhóm sản phẩm</li> <li>- Giám sát việc hoàn thiện thiết kế và chuẩn bị tài liệu</li> <li>- Đảm nhận code chính.</li> <li>- Thiết kế logic giữa các giao diện</li> </ul>	25%
2	22010353	Nguyễn Đức Anh	<ul style="list-style-type: none"> <li>- Vẽ sơ đồ usecase và mô tả chức năng</li> <li>- Đảm nhận viết báo cáo chính.</li> <li>- Thiết kế cơ sở dl, bao gồm mô hình dữ liệu</li> <li>- Code giao diện Thông tin cá nhân</li> <li>- Code giao diện sinh viên</li> </ul>	25%
3	22010047	Phạm Hoàng Anh	<ul style="list-style-type: none"> <li>- Xây dựng sơ đồ quan hệ module, sequence UML, state UML</li> <li>- Nghiên cứu cơ sở dl, đề xuất mô hình lưu trữ.</li> <li>- Làm slide thuyết trình</li> <li>- Thiết kế hệ thống con, mô hình flow diagram</li> <li>- Code giao diện</li> </ul>	30%
4	22010249	Nguyễn Hoàng Sơn	<ul style="list-style-type: none"> <li>- Nghiên cứu tài liệu liên quan để bổ sung cần thiết.</li> <li>- Viết báo cáo sơ bộ hệ thống</li> <li>- Viết tài liệu mô tả giao diện, chức năng</li> <li>- Thiết kế API</li> <li>- Code giao diện quản lý khoa</li> </ul>	20%

# Mục lục

<b>1. Yêu cầu(Requirements).....</b>	<b>4</b>
1.1. Đặt vấn đề (Problem statement).....	4
1.1.1 Mô tả vấn đề.....	4
1.1.2 Giải pháp .....	4
1.2. Thông số kỹ thuật bổ sung.....	5
1.2.1. Hiệu năng .....	5
1.2.2. Tốc độ .....	5
1.2.3. Độ tin cậy .....	5
1.2.4. Tính an toàn và bảo mật.....	5
1.2.5. Khả năng bảo trì.....	6
1.2.6. Khả năng sử dụng .....	6
1.2.7. Khả năng mở rộng.....	6
1.2.8. Tính khả dụng .....	6
1.3. Mô hình hóa chức năng.....	7
1.3.1 Các yêu cầu chức năng .....	7
1.3.2 Sơ đồ Use-case.....	10
1.4. Đặc tả các Use-case .....	11
1.4.1 UC1. Quản lý hệ thống .....	11
1.4.2 UC2. Quản lý tài khoản .....	17
1.4.3 UC3. Quản lý khoa.....	23
1.4.4 UC4.1. Quản lý môn học.....	28
Tiền điều kiện: .....	30
Hậu điều kiện: .....	30
1.4.5 UC4.2. Tra cứu điểm.....	32
1.4.6 UC5.1. Quản lý thông tin cá nhân .....	34
1.4.7 UC5.2. Quản lý thông tin sinh viên .....	37
1.4.8 UC6.1 Quản lý điểm sinh viên .....	41
1.4.9 UC6.1 Lịch học,lịch thi .....	43
UC1.4. Quản lý Lịch học/Lịch thi .....	43
Mô tả ngắn gọn: .....	43
Tác nhân thực hiện: .....	44
Luồng sự kiện: .....	44
Luồng thay thế: .....	45
Tiền điều kiện: .....	45
Hậu điều kiện: .....	45
<b>2. Phân tích trường hợp sử dụng (Use-case analysis).....</b>	<b>47</b>
2.1. Phân tích kiến trúc hệ thống .....	47
2.1.1. Kiến trúc mức cao của hệ thống .....	47
2.1.2. Các đối tượng trung tâm chính của hệ thống (Key abstractions) .....	48
2.2. Thực thi trường hợp sử dụng (Use-case realizations).....	49
2.2.1. Các biểu đồ tuần tự (Sequence diagrams) .....	49
2.2.2. Góc nhìn của các lớp trong hệ thống (Views of participating classes) .....	60
<b>3. Thiết kế (Use-case design) .....</b>	<b>65</b>
3.1. Xác định các thành phần thiết kế (Identify design elements) .....	65
3.1.1. Xác định các lớp (Identify classes) .....	65
3.1.2. Xác định các gói (Identify packages) .....	67
3.1.2. Thiết kế biểu đồ lớp (Class diagrams) .....	68
3.2. Thiết kế cơ sở dữ liệu (Database design) .....	69
3.2.1. Lược đồ cơ sở dữ liệu.....	69
1. Bảng users (Người dùng) .....	69
2. Bảng departments (Phòng ban) .....	70
3. Bảng students (Sinh viên) .....	70
4. Bảng subjects (Môn học) .....	70
5. Bảng grades (Điểm số).....	70
3.4. Thiết kế API.....	71
3.4.1. Cấu trúc thư mục của API.....	71
3.4.2. Danh sách các API.....	71
3.4.3. Thiết kế chi tiết API.....	72
<b>4. Cài đặt.....</b>	<b>85</b>
4.1. Lựa chọn công nghệ.....	85
4.1.1 Database .....	85
4.1.2 Server .....	87
4.1.3 Web Client.....	87
4.2. Cấu trúc mã nguồn.....	89

4.2.1	<i>Server</i> .....	89
4.2.2	<i>Web Client</i> .....	91
4.2.3	<i>Database</i> .....	92
<b>5.</b>	<b>Kết luận.....</b>	<b>94</b>
5.1.	Những việc đã làm được và kết quả đạt được .....	94
5.2.	Những điều chưa làm được.....	94
5.3.	Đề xuất trong tương lai .....	94
	<b>Tài liệu tham khảo.....</b>	<b>95</b>

# 1. Yêu cầu(Requirements)

## 1.1. Đặt vấn đề (Problem statement)

### 1.1.1 Mô tả vấn đề

Quản lý sinh viên là một nhiệm vụ quan trọng đối với các trường học và cơ sở giáo dục, đặc biệt trong bối cảnh số lượng sinh viên ngày càng gia tăng và yêu cầu về quản lý học tập trở nên phức tạp hơn. Việc theo dõi thông tin sinh viên, điểm số, tiến độ học tập, đăng ký môn học và quá trình tốt nghiệp đòi hỏi một hệ thống hiệu quả, chính xác và minh bạch. Các phương pháp quản lý thủ công hoặc hệ thống công nghệ lỗi thời không còn đáp ứng được nhu cầu ngày nay. Vì vậy, việc triển khai một hệ thống quản lý sinh viên hiện đại không chỉ giúp tối ưu hóa quy trình làm việc mà còn nâng cao chất lượng giáo dục, hỗ trợ nhà trường trong việc đưa ra quyết định chính xác và cải thiện trải nghiệm của sinh viên.

### 1.1.2 Giải pháp

Xây dựng một hệ thống có khả năng đáp ứng nhu cầu của nhiều nhóm người dùng khác nhau. Đối với ban quản lý nhà trường, hệ thống cung cấp các chức năng như quản lý thông tin sinh viên, quản lý điểm số, theo dõi tiến độ học tập và quản lý đăng ký môn học. Họ cũng có thể tạo và xuất báo cáo để phục vụ quá trình tổng hợp và ra quyết định chiến lược về giáo dục và đào tạo.

Còn đối với giảng viên, hệ thống giúp họ thực hiện các công việc hàng ngày như nhập điểm, theo dõi tiến độ học tập của sinh viên, quản lý danh sách lớp và hỗ trợ đánh giá kết quả học tập. Đặc biệt, hệ thống giúp giảng viên nhanh chóng tra cứu và tìm kiếm thông tin sinh viên, giảm thời gian xử lý và nâng cao độ chính xác trong công tác giảng dạy.

Không chỉ dừng lại ở việc quản lý học tập, hệ thống còn hỗ trợ phòng đào tạo trong việc quản lý lịch học, lịch thi và xét duyệt kết quả học tập. Nhờ hệ thống này, nhân viên có thể dễ dàng tra cứu thông tin sinh viên, theo dõi quá trình đăng ký môn học, xử lý yêu cầu của sinh viên một cách nhanh chóng và chuyên nghiệp. Điều này giúp nâng cao hiệu quả vận hành của nhà trường và cải thiện trải nghiệm học tập của sinh viên.

Về mặt bảo mật, hệ thống cần đảm bảo bảo vệ chặt chẽ thông tin quan trọng như điểm số, thông tin cá nhân và dữ liệu học tập của sinh viên. Quyền truy cập hệ thống sẽ được phân chia rõ ràng cho từng đối tượng người dùng như ban quản lý, giảng viên, sinh viên và nhân viên phòng đào tạo, đảm bảo mỗi người chỉ truy cập được các thông tin phù hợp với vai trò của mình. Ngoài ra, hệ thống cần hoạt động mượt mà, xử lý nhanh chóng các yêu cầu về đăng ký môn học, nhập điểm và quản lý hồ sơ sinh viên, giúp nhà trường vận hành hiệu quả và tránh gián đoạn trong quá trình đào tạo.

## 1.2. Thông số kỹ thuật bổ sung

### 1.2.1. Hiệu năng

- Thời gian phản hồi: Hệ thống phải có khả năng phản hồi các yêu cầu trong vòng 2 giây cho các tác vụ thông thường (như tìm kiếm sinh viên, xem danh sách môn học).
- Xử lý khối lượng lớn dữ liệu: Hệ thống cần có khả năng quản lý và xử lý khối lượng lớn dữ liệu từ hàng ngàn sinh viên và các bản ghi điểm, đăng ký môn học mỗi ngày.
- Khả năng mở rộng: Hệ thống cần hỗ trợ mở rộng linh hoạt khi số lượng sinh viên và giảng viên tăng mà không ảnh hưởng đến hiệu suất.

### 1.2.2. Tốc độ

- Tốc độ truy xuất cơ sở dữ liệu: Hệ thống cần truy xuất dữ liệu nhanh chóng, với thời gian tải trang dữ liệu (như danh sách sinh viên hoặc bảng điểm) không vượt quá 3 giây.
- Tốc độ xử lý giao dịch: Các thao tác như nhập điểm, đăng ký môn học, hay xét duyệt kết quả học tập phải được xử lý trong thời gian dưới 5 giây.

### 1.2.3. Độ tin cậy

- Tỷ lệ thời gian hoạt động: Hệ thống cần phải đảm bảo tỷ lệ hoạt động ít nhất 99.9% trong suốt thời gian hoạt động.
- Khả năng phục hồi sau sự cố: Hệ thống cần có khả năng phục hồi tự động sau các sự cố hệ thống và tránh làm mất dữ liệu quan trọng.
- Sao lưu dữ liệu: Dữ liệu phải được sao lưu định kỳ hàng ngày để phòng tránh rủi ro mất mát dữ liệu do lỗi hệ thống hoặc phần cứng.

### 1.2.4. Tính an toàn và bảo mật

- Xác thực và phân quyền: Hệ thống phải có cơ chế xác thực và phân quyền chặt chẽ để đảm bảo chỉ những người dùng có quyền mới có thể truy cập vào các chức năng quan trọng.
- Mã hóa dữ liệu: Tất cả dữ liệu nhạy cảm (như thông tin sinh viên, bảng điểm) phải được mã hóa khi lưu trữ và truyền tải qua mạng.
- Chống tấn công: Hệ thống cần bảo vệ chống lại các hình thức tấn công phổ biến như SQL Injection, Cross-Site Scripting (XSS), Cross-Site Request Forgery (CSRF).

#### **1.2.5. Khả năng bảo trì**

- Dễ nâng cấp: Hệ thống cần dễ dàng nâng cấp phần mềm và phần cứng mà không gây gián đoạn lớn trong hoạt động.
- Tính mô-đun: Hệ thống phải có kiến trúc mô-đun để cho phép việc nâng cấp, thay thế các thành phần dễ dàng mà không ảnh hưởng đến toàn bộ hệ thống.

#### **1.2.6. Khả năng sử dụng**

- Giao diện thân thiện: Giao diện hệ thống phải dễ sử dụng, dễ hiểu cho người dùng không chuyên về công nghệ, giúp giảm thời gian đào tạo.

#### **1.2.7. Khả năng mở rộng**

- Tăng trưởng số lượng người dùng: Hệ thống phải đáp ứng được nhu cầu tăng trưởng số lượng sinh viên, giảng viên và nhân viên nhà trường mà không ảnh hưởng đến hiệu suất.
- Mở rộng kho lưu trữ dữ liệu: Hệ thống cần khả năng mở rộng dung lượng lưu trữ khi số lượng sinh viên, môn học và dữ liệu học tập tăng lên.

#### **1.2.8. Tính khả dụng**

- Hoạt động 24/7: Hệ thống cần đảm bảo có khả năng hoạt động liên tục, đặc biệt với các trường học có hệ thống đào tạo trực tuyến.
- Cơ chế phục hồi sau sự cố: Hệ thống cần có kế hoạch phục hồi dữ liệu và khôi phục dịch vụ nhanh chóng sau sự cố.

Những yêu cầu phi chức năng này đảm bảo hệ thống không chỉ hoạt động hiệu quả mà còn duy trì tính ổn định, bảo mật và dễ sử dụng trong dài hạn.

### **1.3. Mô hình hóa chức năng**

#### **1.3.1 Các yêu cầu chức năng**

##### *R1: Các yêu cầu về quản lý hệ thống*

###### **R1.1. Hệ thống cho phép đăng nhập với các vai trò khác nhau**

Hệ thống yêu cầu tất cả người dùng phải đăng nhập để sử dụng các chức năng do hệ thống cung cấp. Chức năng này cho phép tất cả các đối tượng người dùng có thể đăng nhập vào hệ thống với các vai trò khác nhau bằng cách điền đầy đủ các thông tin về email, mật khẩu và chọn vai trò trong giao diện đăng nhập của hệ thống. Các đối tượng người dùng ở đây có thể là quản trị viên, giảng viên và sinh viên.

###### **R1.2. Hệ thống cho phép đăng xuất**

Hệ thống cho phép người dùng có thể đăng xuất ra khỏi chương trình khi muốn thoát ra khỏi chương trình hoặc khi hết phiên làm việc. Chức năng này phải đảm bảo có thể thực hiện được việc đăng xuất ra khỏi hệ thống. Các đối tượng người dùng ở đây có thể là quản trị viên, giảng viên và sinh viên.

###### **R1.3. Hệ thống cho phép thay đổi thông tin cá nhân**

Hệ thống cho phép người dùng quản lý thông tin cá nhân của mình. Chức năng này phải đảm bảo hiển thị thông tin cá nhân chi tiết của người dùng và phải đảm bảo người dân có thể thay đổi thông tin cá nhân của mình.

###### **R1.4 Hệ thống cho phép chủ kho quản lý phân quyền**

Hệ thống được tạo ra cho nhiều nhóm đối tượng người dùng khác nhau, mỗi nhóm người dùng sẽ sử dụng hệ thống cho nhiều mục đích khác nhau nên sẽ có những nhóm quyền khác nhau. Chính vì thế, chức năng này được tạo ra dành cho quản trị viên để dễ dàng quản lý các nhóm quyền. Chức năng này cho phép tác nhân có thể thực hiện việc tìm kiếm, thêm mới, sửa hoặc xóa các quyền trong hệ thống cho từng nhóm người dùng.

##### *R2: Các yêu cầu về quản lý tài khoản người dùng*

Khi sử dụng hệ thống, quản trị viên cần tạo tài khoản đăng nhập và mật khẩu để người dùng có thể đăng nhập vào hệ thống. Và khi có nhiều nhóm người dùng sử dụng các tài khoản khác nhau thì hệ thống cần phải quản lý tất cả các thông tin liên quan đến tài khoản của người sử dụng. Các chức năng đó bao gồm:

###### **R2.1. Hệ thống cho phép cập nhật tài khoản**

Chức năng này được tạo ra dành cho quản trị viên có thể thực hiện việc thêm, sửa, xóa các tài khoản có thể đăng nhập vào hệ thống.

### R2.2. Hệ thống cho phép tìm kiếm tài khoản

Chức năng này được tạo ra dành cho quản trị viên (tác nhân của hệ thống) có thể thực hiện việc tìm kiếm các tài khoản trong danh sách tài khoản có thể đăng nhập vào hệ thống.

### R2.3. Hệ thống cho phép xem thông tin tài khoản

Chức năng này được tạo ra dành cho quản trị viên có thể thực hiện việc xem thông tin các tài khoản trong danh sách tài khoản có thể đăng nhập vào hệ thống.

*R3: Các yêu cầu về quản lý giảng viên Một trường học có nhiều giảng viên thực hiện các công việc giảng dạy và nghiên cứu khác nhau. Hệ thống cần có khả năng quản lý thông tin liên quan đến giảng viên để hỗ trợ việc tổ chức và quản lý trường.*

*Các tính năng này bao gồm:*

### R3.1. Hệ thống cho phép cập nhật thông tin giảng viên

Chức năng này cho phép quản trị viên thêm, sửa, xóa thông tin của giảng viên trong hệ thống, đảm bảo thông tin luôn được cập nhật.

### R3.2. Hệ thống cho phép tìm kiếm giảng viên

Chức năng này cho phép quản trị viên tìm kiếm giảng viên dựa trên các thông tin như tên, mã giảng viên, hoặc các tiêu chí khác.

### R3.3. Hệ thống cho phép xem thông tin giảng viên

Chức năng này cho phép quản trị viên xem thông tin của các giảng viên trong hệ thống để phục vụ cho việc lập tài khoản hoặc đối chiếu thông tin.

*R4: Các yêu cầu về quản lý môn học*

Một trường học có nhiều môn học thuộc các khoa khác nhau, với nhiều chương trình giảng dạy và yêu cầu đầu vào khác nhau. Hệ thống cần quản lý các thông tin liên quan đến môn học để hỗ trợ việc đăng ký, quản lý và theo dõi kết quả học tập. Các đối tượng thực hiện chức năng này thường là quản trị viên và giảng viên.

### R4.1. Hệ thống cho phép quản lý thông tin môn học

Chức năng này cho phép thêm, sửa, xóa, xem thông tin cơ bản và yêu cầu của môn học, đảm bảo dữ liệu môn học luôn được duy trì cập nhật. Điều này rất quan trọng để quản lý có thể theo dõi chính xác từng môn học và đưa ra các quyết định phù hợp.

#### R4.2. Hệ thống cho phép quản lý thuộc tính môn học

Chức năng này giúp quản lý các đặc điểm chi tiết của môn học như số tín chỉ, điều kiện tiên quyết, chương trình giảng dạy, giúp sinh viên có đầy đủ thông tin khi đăng ký học.

#### R4.3. Hệ thống cho phép tìm kiếm môn học

Chức năng tìm kiếm môn học giúp sinh viên và giảng viên dễ dàng tra cứu các môn học theo mã môn học, tên môn học hoặc các tiêu chí khác, hỗ trợ quá trình lập kế hoạch học tập chính xác.

### *R5: Các yêu cầu về quản lý sinh viên*

Hệ thống cần có khả năng quản lý hoạt động nhập học, đăng ký môn học và theo dõi quá trình học tập của sinh viên để kiểm soát chất lượng đào tạo.

#### R5.1. Hệ thống cho phép quản lý hồ sơ sinh viên

Chức năng này cho phép quản trị viên cập nhật thông tin cá nhân, lịch sử học tập và kết quả học tập của sinh viên.

#### R5.2. Hệ thống cho phép quản lý đăng ký môn học

Chức năng này cho phép sinh viên đăng ký môn học trong từng học kỳ, kiểm tra điều kiện tiên quyết và theo dõi lịch học.

### *R6: Các yêu cầu về quản lý đối tác*

Hệ thống cần quản lý các thông tin liên quan đến các đối tác như nhà tuyển dụng, doanh nghiệp hợp tác để hỗ trợ việc thực tập và cơ hội việc làm cho sinh viên.

#### R6.1. Hệ thống cho phép quản lý thông tin của nhà tuyển dụng

Chức năng này cho phép lưu trữ và cập nhật thông tin về nhà tuyển dụng, hỗ trợ sinh viên trong việc tìm kiếm cơ hội thực tập và việc làm.

#### R6.2. Hệ thống cho phép quản lý thông tin của cựu sinh viên

Chức năng này cho phép lưu trữ và cập nhật thông tin về cựu sinh viên để hỗ trợ việc xây dựng mạng lưới kết nối và tư vấn nghề nghiệp.

### *R7: Các yêu cầu về quản lý dịch vụ sinh viên*

Hệ thống cần quản lý các dịch vụ hỗ trợ sinh viên như học bỗng, tư vấn học tập và hỗ trợ tài chính để đảm bảo điều kiện học tập tốt nhất.

#### R7.1. Hệ thống cho phép quản lý thông tin học bỗng

Chức năng này giúp quản trị viên ghi nhận và theo dõi thông tin học bỗng, giúp sinh viên dễ dàng tiếp cận cơ hội tài trợ.

#### R7.2. Hệ thống cho phép quản lý thông tin hỗ trợ tài chính

Chức năng này cho phép sinh viên đăng ký và theo dõi các khoản hỗ trợ tài chính để đảm bảo quá trình học tập không bị gián đoạn.

### R8: Các yêu cầu về quản lý báo cáo thống kê

Hệ thống cần có khả năng tổng hợp dữ liệu và cung cấp báo cáo để hỗ trợ việc ra quyết định dựa trên dữ liệu.

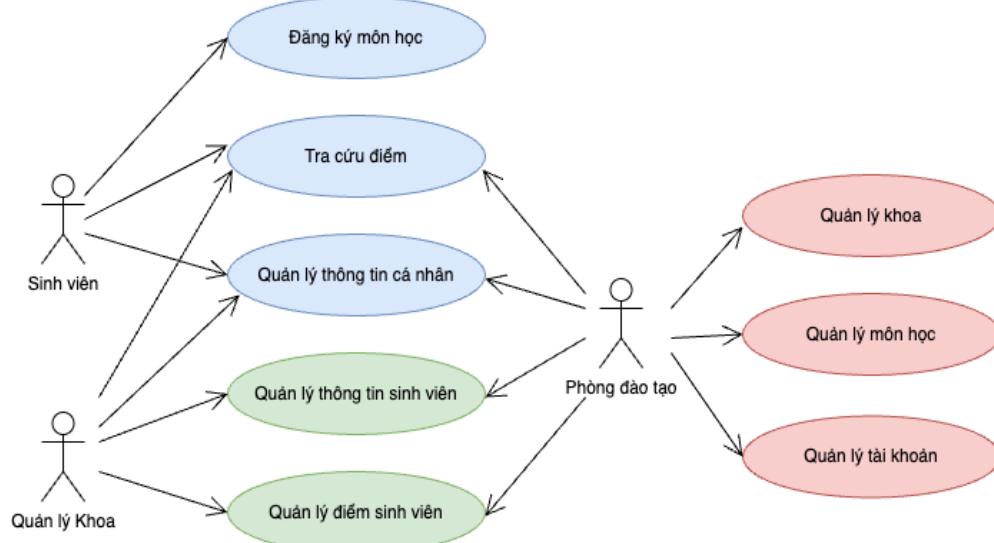
#### R8.1: Hệ thống cho phép xem báo cáo thống kê tổng quan, kết quả học tập, số lượng sinh viên theo ngành học

Chức năng này cung cấp các báo cáo tổng quan về tình hình học tập, giúp quản lý có cái nhìn tổng thể về chất lượng đào tạo.

#### R8.2: Hệ thống cho phép xuất báo cáo thành file Excel

Chức năng này giúp quản lý xuất các báo cáo thành file Excel để dễ dàng lưu trữ, chia sẻ và phân tích chi tiết ngoài hệ thống.

### 1.3.2 Sơ đồ Use-case



Hình 1.1 Sơ đồ Use-case

## 1. Sinh viên:

- **Đăng ký môn học:** Sinh viên có thể chọn và đăng ký môn học cho các kỳ học.
- **Tra cứu điểm:** Sinh viên có thể xem điểm của mình trong hệ thống.
- **Quản lý thông tin cá nhân:** Sinh viên có thể cập nhật hoặc xem thông tin cá nhân của mình.

## 2. Quản lý Khoa:

- **Quản lý thông tin sinh viên:** Quản lý khoa có thể theo dõi và cập nhật thông tin sinh viên trong khoa
- **Quản lý điểm sinh viên:** Quản lý khoa có quyền nhập hoặc chỉnh sửa điểm số của sinh viên trong khoa
- **Quản lý môn học:** Chịu trách nhiệm thêm, sửa, xóa, và quản lý danh sách môn học trong khoa

## 3. Phòng Đào Tạo:

- Bao gồm các chức năng của quản lý khoa
- **Quản lý khoa:** Chịu trách nhiệm quản lý thông tin các khoa trong trường.
- **Quản lý môn học:** Chịu trách nhiệm thêm, sửa, xóa, và quản lý danh sách môn học.
- **Quản lý tài khoản:** Chịu trách nhiệm quản lý tài khoản của sinh viên, giảng viên, và các nhân viên liên quan.

## 1.4. Đặc tả các Use-case

### 1.4.1 UC1. Quản lý hệ thống

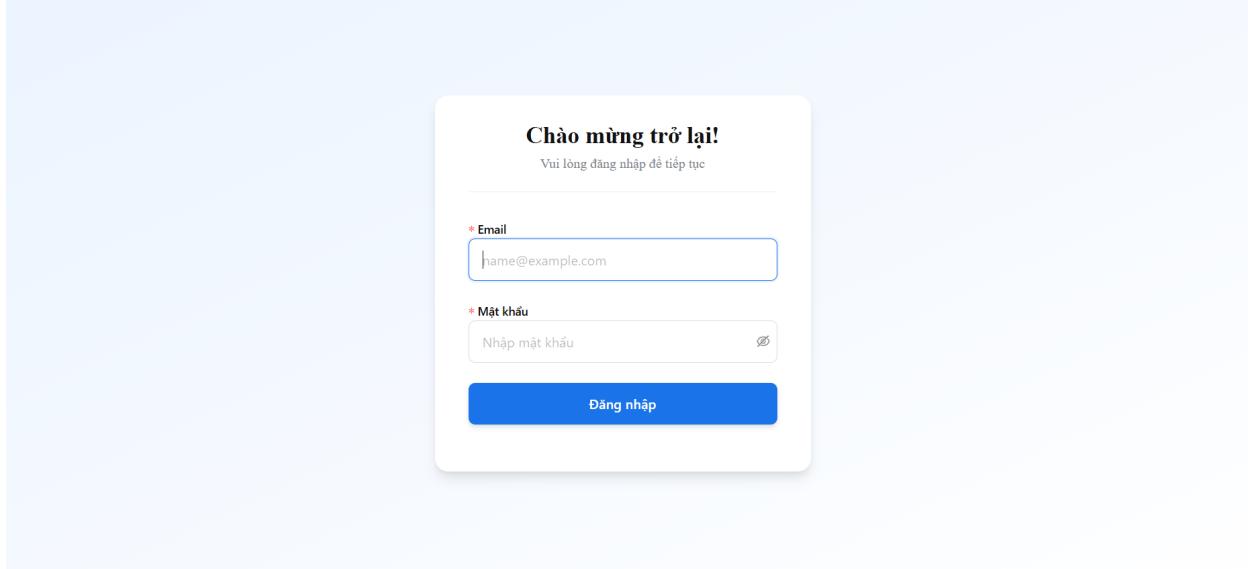
#### UC1.1. Đăng nhập

- Mô tả ngắn gọn: UC này cho phép tất cả người dùng đăng nhập với các vai trò khác nhau để sử dụng các chức năng do hệ thống cung cấp.
- Tác nhân thực hiện: Quản lý khoa, phòng đào tạo, sinh viên.
- Luồng sự kiện:

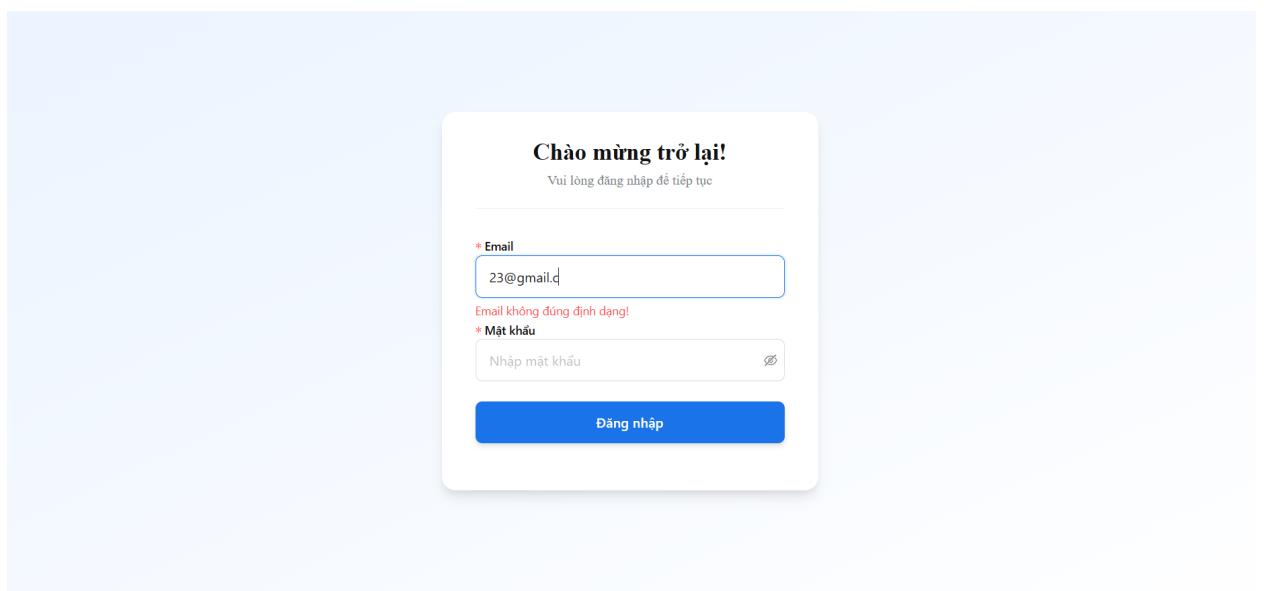
\***Luồng cơ bản:** UC này bắt đầu khi người dùng muốn đăng nhập vào hệ thống với vai trò tương ứng.

1. Người dùng truy cập giao diện đăng nhập của hệ thống.
2. Hệ thống yêu cầu người dùng nhập thông tin đăng nhập (email, mật khẩu)

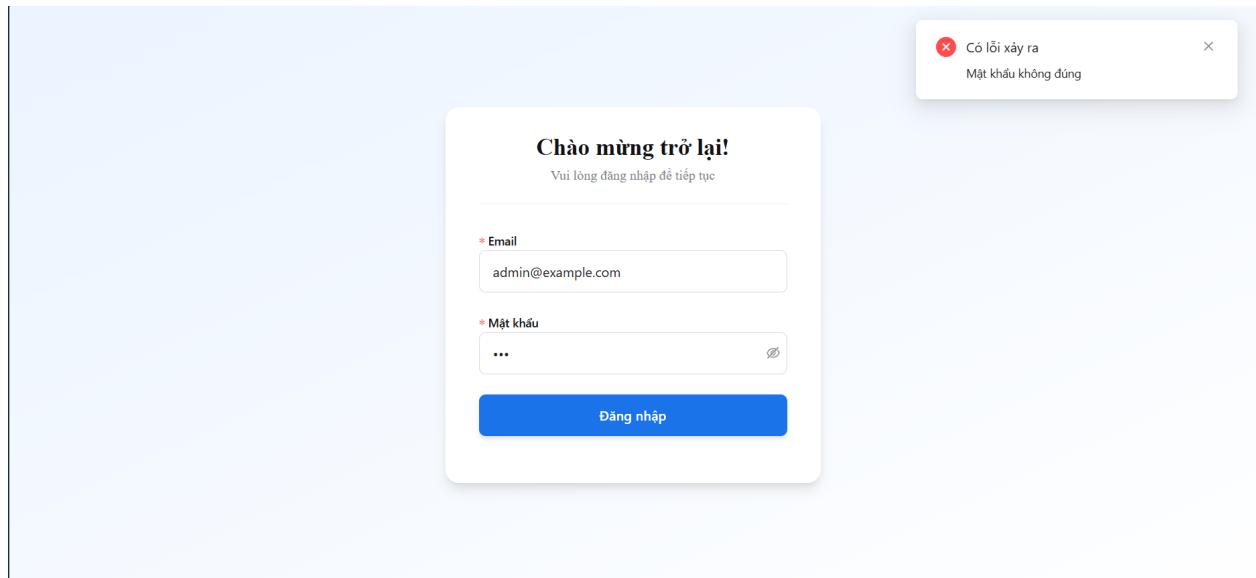
3. Sau khi người dùng nhập đủ thông tin và bấm nút “Đăng nhập”, hệ thống kiểm tra tính hợp lệ của thông tin.
  4. Nếu thông tin đăng nhập hợp lệ, người dùng sẽ được truy cập vào hệ thống với vai trò tương ứng với tài khoản
  5. Nếu thông tin không hợp lệ, hệ thống hiển thị thông báo lỗi và yêu cầu người dùng nhập lại.
- Giao diện minh họa:



Hình 1.2 Giao diện đăng nhập tài khoản của người dùng



Hình 1.3 Giao diện lỗi email không đúng định dạng



Hình 1.4 Giao diện lỗi mật khẩu vừa nhập không chính xác

### UC1.2. Đăng xuất

- Mô tả ngắn gọn: UC này cho phép người dùng đăng xuất khỏi hệ thống khi muốn thoát ra khỏi chương trình hoặc khi hết phiên làm việc. Chức năng này đảm bảo người dùng có thể đăng xuất an toàn và kết thúc phiên làm việc của mình.
- Tác nhân thực hiện: Quản lý khoa, Sinh viên, phòng đào tạo.
- Luồng sự kiện:

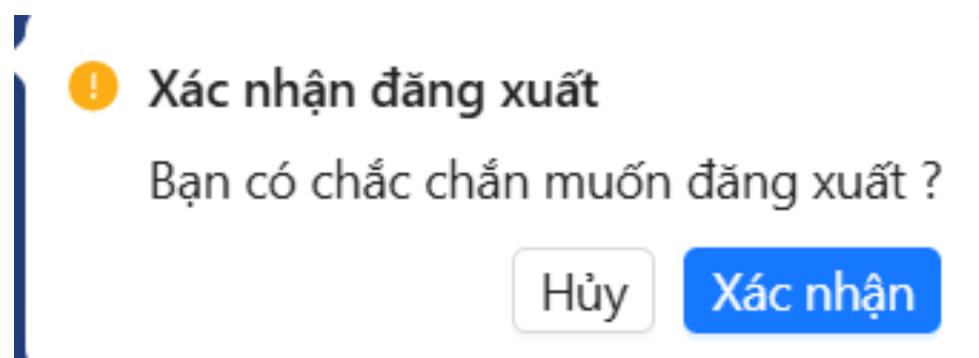
**\*Luồng cơ bản:** UC này bắt đầu khi người dùng muốn đăng xuất khỏi hệ thống.

1. Người dùng chọn chức năng Đăng xuất từ giao diện chính của hệ thống.
2. Hệ thống hiển thị thông báo yêu cầu xác nhận việc đăng xuất.
3. Người dùng xác nhận việc đăng xuất bằng cách bấm nút “Yes” hoặc “No”.
4. Hệ thống xử lý yêu cầu đăng xuất, chấm dứt phiên làm việc hiện tại và đưa người dùng trở về màn hình đăng nhập.

**\*Luồng thay thế:**

- Hủy thao tác đăng xuất: Nếu người dùng chọn “Hủy” khi được yêu cầu xác nhận đăng xuất, hệ thống sẽ hủy yêu cầu đăng xuất và người dùng tiếp tục phiên làm việc hiện tại.
- Tiền điều kiện:
  - Người dùng phải đang đăng nhập và có phiên làm việc trong hệ thống.
  - Hệ thống đang hoạt động bình thường và sẵn sàng xử lý yêu cầu đăng xuất.
- Hậu điều kiện:

- Nếu thành công: Phiên làm việc của người dùng được kết thúc, người dùng được đưa về màn hình đăng nhập và tất cả thông tin phiên làm việc được xóa an toàn.
  - Nếu không thành công: Phiên làm việc của người dùng vẫn còn hiệu lực và người dùng tiếp tục làm việc trên hệ thống.
- Giao diện minh họa:



Hình 1.5 Thông báo xác nhận đăng xuất

### UC1.3. Thay đổi thông tin cá nhân

- Mô tả ngắn gọn: UC này cho phép người dùng quản lý và thay đổi thông tin cá nhân của mình trong hệ thống. Chức năng đảm bảo hiển thị đầy đủ thông tin chi tiết của người dùng và cho phép người dùng cập nhật các thông tin cá nhân khi cần thiết.
- Tác nhân thực hiện: Quản lý khoa, sinh viên, phòng đào tạo.
- Luồng sự kiện:

\***Luồng cơ bản:** UC này bắt đầu khi người dùng muốn thay đổi thông tin cá nhân.

1. Người dùng chọn chức năng “Thông tin cá nhân” trong giao diện cá nhân của hệ thống.
2. Hệ thống hiển thị đầy đủ thông tin cá nhân hiện tại của người dùng, bao gồm: tên, ngày sinh, CCCD, số điện thoại, địa chỉ email, địa chỉ nhà, và các thông tin khác.
3. Người dùng có thể chỉnh sửa các thông tin cần thay đổi.
4. Sau khi hoàn tất việc chỉnh sửa, người dùng bấm nút “Lưu”.
5. Hệ thống kiểm tra tính hợp lệ của thông tin mới (ví dụ: định dạng số điện thoại, địa chỉ email).
6. Hệ thống xác nhận việc thay đổi thông tin cá nhân thành công và thông báo cho người dùng.

\***Luồng phụ:**

### S-1: Kiểm tra tính hợp lệ của thông tin:

1. Khi người dùng nhập các thông tin mới, hệ thống sẽ tự động kiểm tra tính hợp lệ của từng trường dữ liệu (ví dụ: định dạng email, số điện thoại).
2. Nếu có lỗi định dạng, hệ thống hiển thị thông báo lỗi cho người dùng và yêu cầu chỉnh sửa lại thông tin không hợp lệ.

#### \* Luồng thay thế:

- Thông tin không hợp lệ: Nếu người dùng nhập thông tin không đúng định dạng hoặc để trống các trường bắt buộc (ví dụ: email không đúng định dạng), hệ thống sẽ hiển thị thông báo lỗi và yêu cầu người dùng nhập lại thông tin chính xác.
- Hủy thao tác thay đổi thông tin: Nếu người dùng quyết định không thay đổi thông tin cá nhân, người dùng có thể chọn “Hủy” để quay lại màn hình trước mà không thực hiện bất kỳ thay đổi nào.
- Tiền điều kiện:
  - Người dùng phải đăng nhập vào hệ thống và có quyền truy cập vào chức năng quản lý thông tin cá nhân.
  - Hệ thống phải hoạt động bình thường và hiển thị đầy đủ thông tin hiện tại của người dùng.
- Hậu điều kiện:
  - Nếu thành công: Thông tin cá nhân của người dùng được cập nhật vào cơ sở dữ liệu và người dùng có thể thấy thông tin mới sau khi thay đổi.
  - Nếu không thành công: Thông tin cũ của người dùng vẫn giữ nguyên, và hệ thống yêu cầu người dùng thực hiện lại thao tác thay đổi thông tin

### UC1.4. Quản lý nhóm quyền

- Mô tả ngắn gọn: UC này cho phép phòng đào tạo thực hiện việc quản lý các nhóm quyền cho các đối tượng người dùng khác nhau. Có thể tìm kiếm, thêm mới, sửa đổi hoặc xóa các quyền trong hệ thống, đảm bảo mỗi nhóm người dùng có quyền hạn phù hợp với vai trò của họ trong hệ thống.
- Tác nhân thực hiện: Phòng đào tạo
- Luồng sự kiện:

**\*Luồng cơ bản:** UC này bắt đầu khi tác nhân muốn tạo, sửa, xoá, tìm kiếm, xem chi tiết các nhóm quyền trong hệ thống.

1. Tác nhân chọn chức năng “Phân quyền” trong giao diện quản lý của hệ thống.
2. Hệ thống hiển thị danh sách các nhóm quyền hiện có.

3. Tác nhân có thể thực hiện các chức năng sau: tìm kiếm nhóm quyền, thêm sửa xóa nhóm quyền, xem chi tiết nhóm quyền.
4. Sau khi chọn chức năng, một trong các luồng phụ sau sẽ được thực hiện:
  - o Nếu tác nhân chọn “Tìm kiếm nhóm quyền”, luồng phụ S-1: Tìm kiếm nhóm quyền sẽ được thực hiện.
  - o Nếu tác nhân chọn “Thêm mới nhóm quyền”, luồng phụ S-2: Thêm mới nhóm quyền sẽ được thực hiện.
  - o Nếu tác nhân chọn “Sửa nhóm quyền”, luồng phụ S-3: Sửa nhóm quyền sẽ được thực hiện.
  - o Nếu tác nhân chọn “Xóa nhóm quyền”, luồng phụ S-4: Xóa nhóm quyền sẽ được thực hiện.
  - o Nếu tác nhân chọn “Xem chi tiết nhóm quyền”, luồng phụ S-5: Xem chi tiết nhóm quyền sẽ được thực hiện.

#### **Luồng phụ:**

##### **S-1: Tìm kiếm nhóm quyền:**

1. Tác nhân nhập tên nhóm quyền hoặc một phần thông tin của nhóm quyền vào ô tìm kiếm và bấm nút tìm kiếm để thực hiện chức năng.
2. Hệ thống tìm kiếm và hiển thị danh sách các nhóm quyền phù hợp với từ khóa tìm kiếm.

##### **S-2: Thêm mới nhóm quyền:**

1. Hệ thống hiển thị form thêm mới nhóm quyền gồm: tên nhóm quyền, chọn hành động tương ứng với chức năng muốn phân quyền.
2. Tác nhân nhập thông tin nhóm quyền và các quyền hạn tương ứng.
3. Sau khi hoàn tất, tác nhân bấm nút “Thêm nhóm quyền”.
4. Hệ thống kiểm tra tính hợp lệ của thông tin và lưu nhóm quyền mới vào cơ sở dữ liệu.
5. Hệ thống xác nhận việc thêm mới thành công và cập nhật danh sách nhóm quyền.

##### **S-3: Sửa nhóm quyền:**

1. Tác nhân chọn nhóm quyền cần sửa từ danh sách và bấm nút “Sửa”
2. Hệ thống hiển thị form thông tin chi tiết của nhóm quyền.
3. Tác nhân chỉnh sửa các thông tin cần thiết và bấm nút “Lưu thông tin”.

4. Hệ thống kiểm tra tính hợp lệ và cập nhật nhóm quyền trong cơ sở dữ liệu.
5. Hệ thống xác nhận việc sửa nhóm quyền thành công.

#### **S-4: Xóa nhóm quyền:**

1. Tác nhân chọn nhóm quyền cần xóa từ danh sách và bấm nút “Xoá”
2. Hệ thống hiển thị thông báo yêu cầu xác nhận việc xóa nhóm quyền.
3. Tác nhân xác nhận bằng cách bấm nút “OK”.
4. Hệ thống xóa nhóm quyền khỏi cơ sở dữ liệu và cập nhật danh sách nhóm quyền.

#### **S-5: Xem chi tiết nhóm quyền:**

1. Tác nhân chọn nhóm quyền cần xem từ danh sách và bấm nút “Chi tiết”.
2. Hệ thống hiển thị form thông tin chi tiết của nhóm quyền.

#### **\*Luồng thay thế:**

- Thông tin nhóm quyền không hợp lệ: Nếu tác nhân nhập thông tin nhóm quyền không hợp lệ (ví dụ: bỏ trống trường bắt buộc hoặc nhập sai định dạng), hệ thống sẽ hiển thị thông báo lỗi và yêu cầu tác nhân nhập lại thông tin chính xác.
- Hủy thao tác quản lý quyền: tác nhân quyết định không thực hiện thêm mới, sửa hoặc xóa nhóm quyền, tác nhân có thể chọn “Hủy” để quay lại màn hình quản lý mà không thực hiện thay đổi nào.
- Tiền điều kiện:
  - Tác nhân phải đăng nhập vào hệ thống và có quyền truy cập vào chức năng quản lý phân quyền.
  - Hệ thống phải hoạt động bình thường và sẵn sàng xử lý yêu cầu.
- Hậu điều kiện:
  - Nếu thành công: Nhóm quyền được thêm mới, sửa đổi hoặc xóa khỏi cơ sở dữ liệu và hệ thống cập nhật danh sách quyền.
  - Nếu không thành công: Thông tin quyền không thay đổi, và hệ thống yêu cầu tác nhân thực hiện lại thao tác.

#### **1.4.2 UC2. Quản lý tài khoản**

- Mô tả ngắn gọn: UC này cho phép quản lý kho tạo, cập nhật, tìm kiếm và xem thông tin tài khoản của người dùng trong hệ thống. Các tài khoản này được sử dụng để đăng nhập và phân quyền cho các nhóm người dùng khác nhau.
- Tác nhân thực hiện: Phòng đào tạo

- Luồng sự kiện:

**\*Luồng cơ bản:** UC này hoạt động khi Phòng Đào Tạo muốn quản lý tài khoản người dùng, bao gồm các thao tác như cập nhật thông tin, tìm kiếm hoặc xem chi tiết tài khoản người dùng.

1. Phòng Đào Tạo truy cập vào chức năng "Quản lý tài khoản" trên giao diện hệ thống.
2. Hệ thống hiển thị các tùy chọn: cập nhật tài khoản, tìm kiếm tài khoản, hoặc xem thông tin tài khoản.
3. Sau khi Phòng Đào Tạo chọn một trong các chức năng, một trong các luồng phụ sau sẽ được thực thi:
  - o Nếu chọn "Thêm tài khoản", thực thi S-1: Thêm tài khoản.
  - o Nếu chọn "Sửa tài khoản", thực thi S-2: Sửa tài khoản.
  - o Nếu chọn "Xóa tài khoản", thực thi S-3: Xóa tài khoản.
  - o Nếu chọn "Tìm kiếm tài khoản", thực thi S-4: Tìm kiếm tài khoản.
  - o Nếu chọn "Xem thông tin tài khoản", thực thi S-5: Xem thông tin tài khoản.

**\*Luồng phụ:**

#### S-1: Thêm tài khoản

1. Hệ thống hiển thị form tạo tài khoản mới với các thông tin sau:
  - o Mã nhân viên/Sinh viên
  - o Mật khẩu
  - o Nhóm quyền
2. Phòng Đào Tạo nhập đầy đủ thông tin và bấm "Thêm tài khoản".
3. Hệ thống kiểm tra tính hợp lệ và lưu vào cơ sở dữ liệu.
4. Hệ thống xác nhận việc thêm tài khoản thành công và cập nhật danh sách tài khoản.

#### S-2: Sửa tài khoản

1. Phòng Đào Tạo chọn tài khoản cần chỉnh sửa từ danh sách và bấm "Sửa".
2. Hệ thống hiển thị form chỉnh sửa thông tin tài khoản.
3. Phòng Đào Tạo thực hiện thay đổi thông tin cần thiết.
4. Sau khi sửa, bấm "Lưu thông tin".
5. Hệ thống kiểm tra tính hợp lệ và cập nhật thông tin vào cơ sở dữ liệu.
6. Hệ thống xác nhận việc sửa tài khoản thành công.

### S-3: Xóa tài khoản

1. Phòng Đào Tạo chọn tài khoản cần xóa từ danh sách và bấm "Xóa".
2. Hệ thống hiển thị thông báo xác nhận xóa tài khoản.
3. Phòng Đào Tạo xác nhận bằng cách bấm "OK".
4. Hệ thống xóa thông tin tài khoản khỏi cơ sở dữ liệu và cập nhật danh sách tài khoản.

### S-4: Tìm kiếm tài khoản

1. Phòng Đào Tạo nhập từ khóa tìm kiếm và bấm "Tìm kiếm".
2. Hệ thống lọc danh sách tài khoản dựa trên từ khóa.
3. Hệ thống hiển thị danh sách tài khoản phù hợp.
4. Phòng Đào Tạo có thể chọn tài khoản từ danh sách để xem, sửa hoặc xóa.

### S-5: Xem thông tin tài khoản

1. Phòng Đào Tạo chọn tài khoản từ danh sách và bấm "Chi tiết".
2. Hệ thống hiển thị thông tin chi tiết của tài khoản, bao gồm:
  - o Mã tài khoản
  - o Mã nhân viên/Sinh viên
  - o Tên nhân viên/Sinh viên
  - o Email
  - o Nhóm quyền
  - o Trạng thái
3. Phòng Đào Tạo có thể xem chi tiết và bấm "Quay lại" để trở về danh sách tài khoản.

### Luồng thay thế:

- Thông tin tài khoản không hợp lệ: Nếu Quản lý Khoa nhập thông tin không hợp lệ trong quá trình cập nhật (ví dụ: email không đúng định dạng, bỏ trống trường bắt buộc), hệ thống sẽ hiển thị thông báo lỗi và yêu cầu Quản lý Khoa nhập lại thông tin đúng.
- Không tìm thấy tài khoản: Nếu từ khóa tìm kiếm không khớp với bất kỳ tài khoản nào trong hệ thống, hệ thống sẽ hiển thị thông báo "Không tìm thấy kết quả phù hợp." Quản lý Khoa có thể thay đổi tiêu chí tìm kiếm và thực hiện lại thao tác.

- Hủy thao tác: Trong quá trình thực hiện các thao tác (cập nhật, tìm kiếm, xem thông tin), nếu Quản lý Khoa chọn "Hủy bỏ," hệ thống sẽ quay lại màn hình quản lý tài khoản mà không thực hiện bất kỳ thay đổi nào.

### **Tiền điều kiện:**

- Quản lý Khoa phải đăng nhập thành công vào hệ thống và có quyền truy cập để quản lý tài khoản.
- Hệ thống phải hoạt động bình thường và sẵn sàng xử lý yêu cầu quản lý tài khoản.

### **Hậu điều kiện:**

- **Nếu thành công:**

- Thông tin tài khoản được cập nhật, tìm kiếm hoặc xem chi tiết sẽ được hệ thống phản ánh chính xác và lưu vào cơ sở dữ liệu.
- Quản lý Khoa có thể thấy danh sách các tài khoản và thông tin liên quan đến tài khoản được cập nhật.

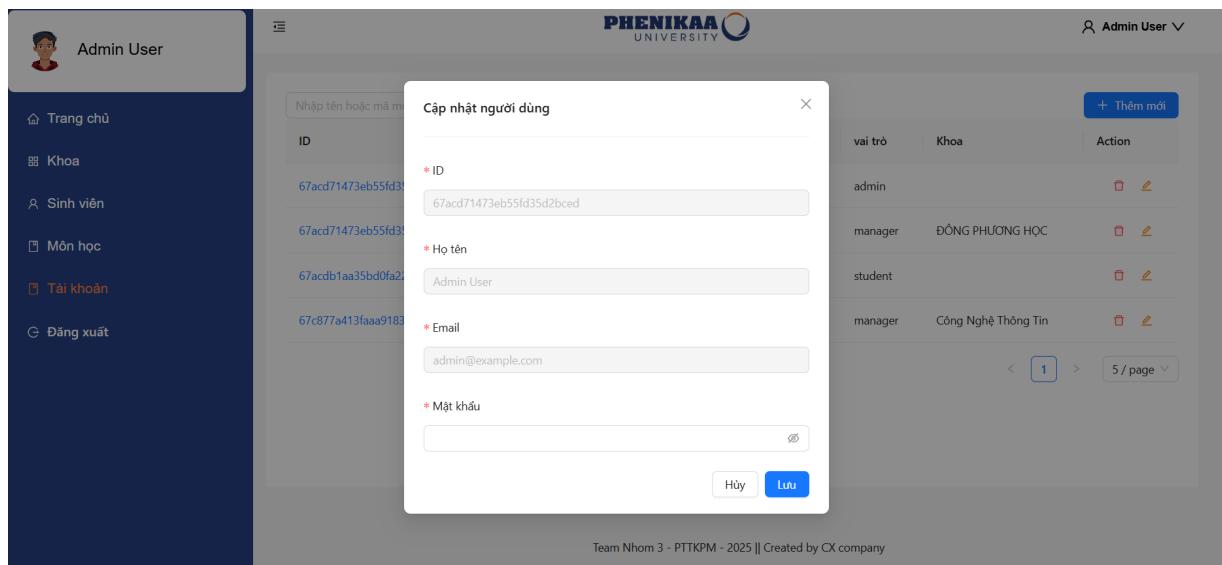
- **Nếu không thành công:**

- Hệ thống không thay đổi trạng thái dữ liệu.
- Quản lý Khoa sẽ nhận được thông báo lỗi hoặc yêu cầu nhập lại thông tin chính xác.

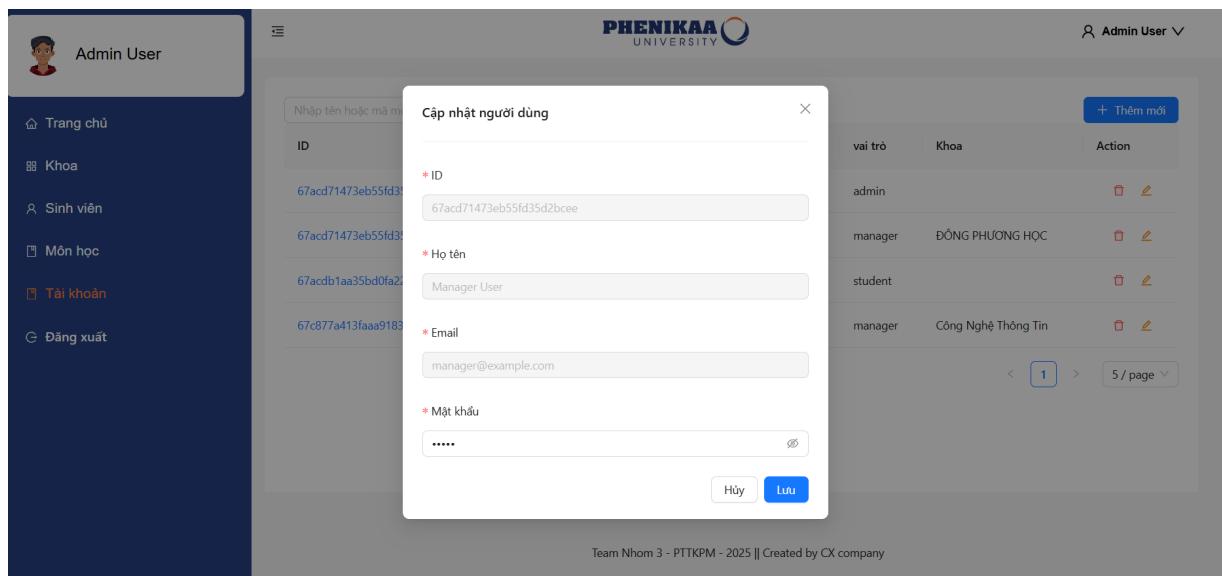
Giao diện:

ID	Họ và tên	Email	vai trò	Khoa	Action
67acd71473eb55fd35d2bc0ed	Admin User	admin@example.com	admin	ĐỘNG PHƯƠNG HỌC	
67acd71473eb55fd35d2bc0ee	Manager User	manager@example.com	manager	ĐỘNG PHƯƠNG HỌC	
67acdb1aa35bd0fa2213cf2	Nguyen Cao Chien	caochien@gmail.com	student	ĐỘNG PHƯƠNG HỌC	
67c877a413faaa91839f5617	Nguyễn Cao Chiến	nguyencaochien03@gmail.com	manager	CÔNG NGHỆ THÔNG TIN	

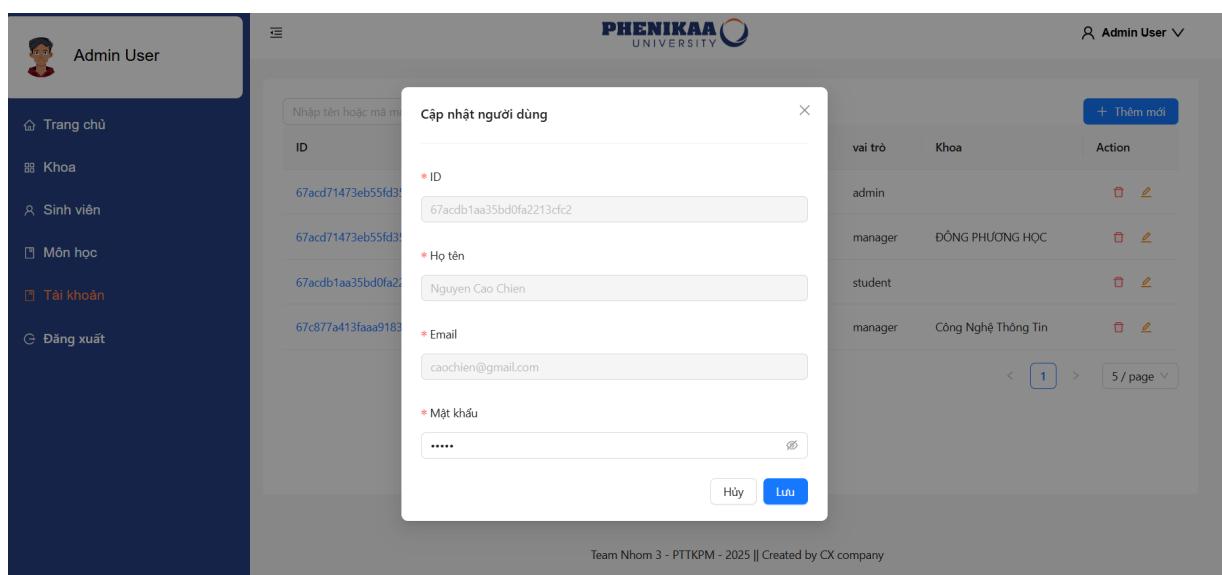
*Hình 1.6 Giao diện quản lý tài khoản*



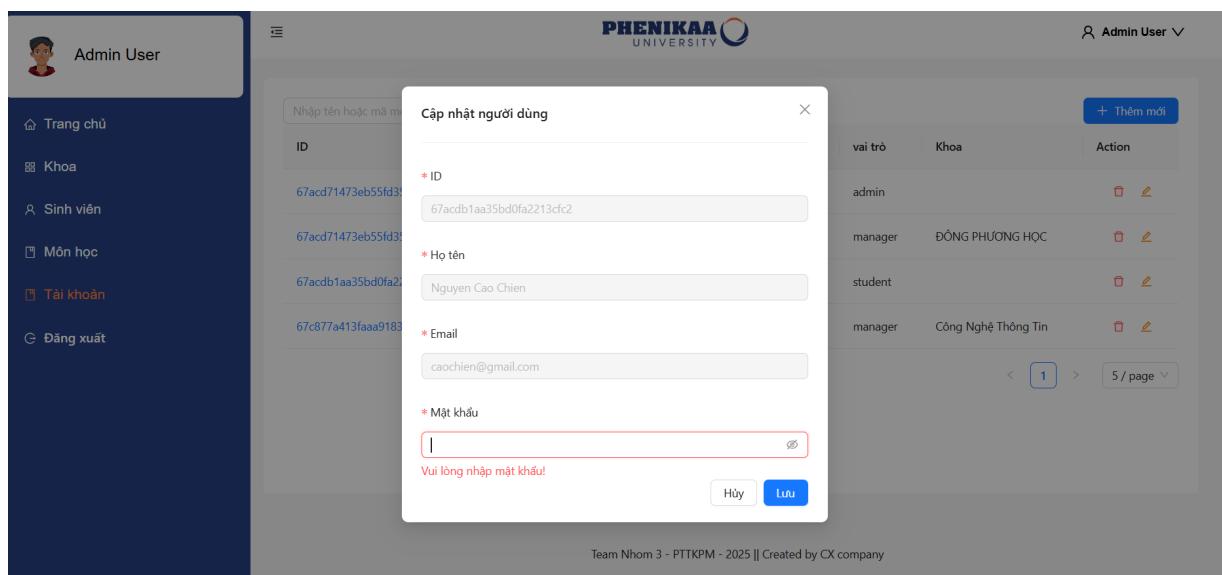
Hình 1.7 Thông báo giao diện chỉnh sửa tài khoản admin



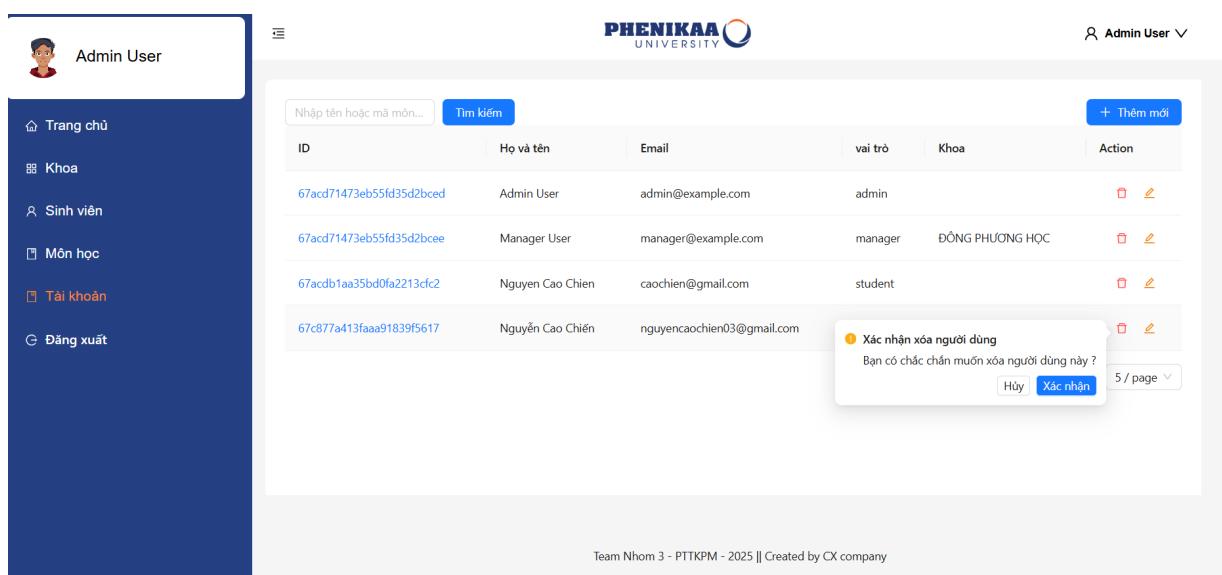
Hình 1.8 Thông báo giao diện chỉnh sửa tài khoản quản lý khoa



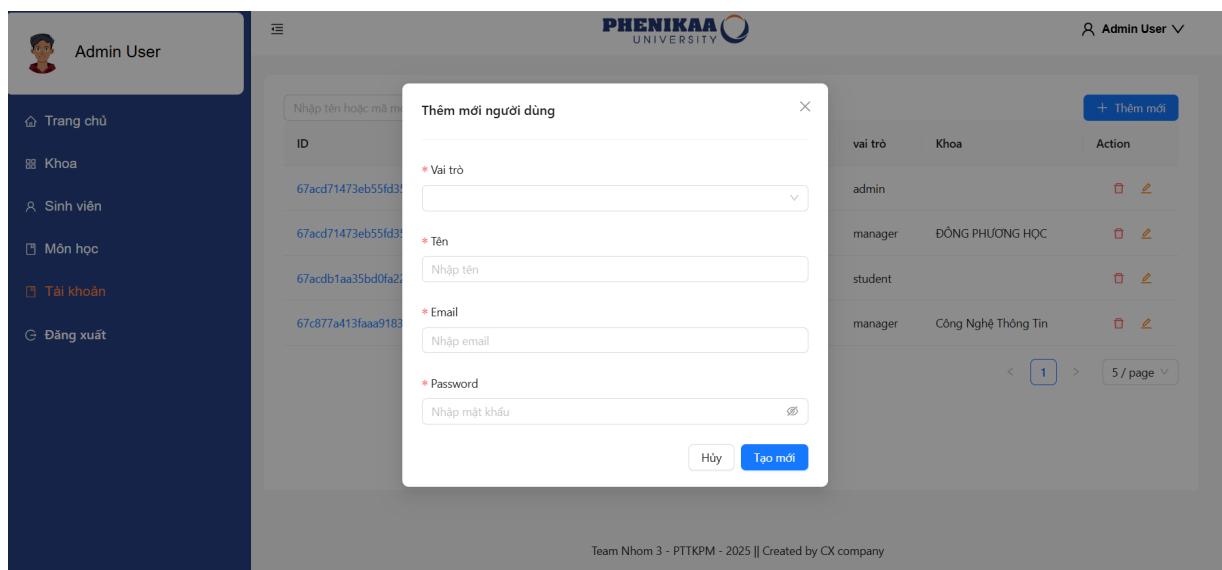
Hình 1.9 Thông báo giao diện chỉnh sửa tài khoản sinh viên



Hình 1.10 Thông báo giao diện khi người dùng nhập thiếu thông tin



Hình 1.11 Thông báo giao diện xác nhận xóa người dùng



Hình 1.12 Thông báo giao diện thêm tài khoản quản lý

#### **1.4.3 UC3. Quản lý khoa**

- Mô tả ngắn gọn: UC này cho phép quản lý thông tin nhân viên trong hệ thống, bao gồm: thêm mới, chỉnh sửa, xóa, xem chi tiết và tìm kiếm thông tin khoa.
- Tác nhân thực hiện: Phòng đào tạo.
- Luồng sự kiện:

**\*Luồng cơ bản:** UC này hoạt động khi Phòng Đào Tạo muốn thêm mới, sửa đổi, xóa, xem chi tiết hoặc tìm kiếm thông tin khoa trong hệ thống.

1. Phòng Đào Tạo chọn chức năng Quản lý nhân viên bằng cách bấm vào mục “Nhân viên” trong thanh công cụ.
2. Hệ thống yêu cầu Phòng Đào Tạo chọn chức năng muốn thực hiện (thêm nhân khoa, sửa khoa, xóa, xem chi tiết, tìm kiếm khoa).
3. Sau khi chọn chức năng, một trong các luồng phụ sau sẽ được thực hiện:
  - o Nếu chọn “Thêm khoa”, luồng phụ S-1: Thêm nhân viên sẽ được thực thi.
  - o Nếu chọn “Sửa nhân viên”, luồng phụ S-2: Sửa nhân viên sẽ được thực thi.
  - o Nếu chọn “Xóa nhân viên”, luồng phụ S-3: Xóa nhân viên sẽ được thực thi.
  - o Nếu chọn “Tìm kiếm thông tin nhân viên”, luồng phụ S-4: Tìm kiếm nhân viên sẽ được thực thi.

**\*Luồng phụ:**

##### **S-1: Thêm nhân viên:**

1. Hệ thống hiển thị form thêm mới khoa và yêu cầu người dùng cung cấp các thông tin như:
  - o Tên khoa
  - o Mã khoa
  - o Quản lý của khoa
2. Sau khi người dùng cung cấp đầy đủ thông tin, bấm nút “Tạo mới” để lưu thông tin.
3. Hệ thống kiểm tra tính hợp lệ của thông tin và lưu vào cơ sở dữ liệu.

4. Hệ thống xác nhận việc thêm khoa thành công và cập nhật danh sách khoa.

#### S-2: Sửa khoa:

1. Người dùng chọn khoa cần sửa từ danh sách và bấm nút “Sửa”.
2. Hệ thống hiển thị form thông tin khoa hiện tại.
3. Người dùng thực hiện chỉnh sửa các thông tin cần thay đổi.
4. Sau khi sửa xong, bấm nút “Lưu”.
5. Hệ thống kiểm tra tính hợp lệ của thông tin và cập nhật thông tin mới vào cơ sở dữ liệu.
6. Hệ thống xác nhận việc sửa khoa thành công và cập nhật danh sách khoa.

#### S-3: Xóa khoa:

1. Người dùng chọn khoa cần xóa từ danh sách và bấm nút “Xoá”.
2. Hệ thống hiển thị thông báo yêu cầu xác nhận việc xóa khoa.
3. Người dùng xác nhận việc xóa bằng cách bấm nút “OK”.
4. Hệ thống xóa thông tin khoa khỏi cơ sở dữ liệu và cập nhật lại danh sách khoa.

#### S-4: Tìm kiếm thông tin khoa:

1. Người dùng nhập thông tin nhân viên cần tìm kiếm vào ô tìm kiếm (có thể là Tên khoa, mã khoa hoặc quản lý của khoa) và bấm nút tìm kiếm.
2. Hệ thống tiến hành tìm kiếm thông tin khoa dựa trên từ khóa mà người dùng cung cấp.
3. Hệ thống hiển thị kết quả tìm kiếm là danh sách các khoa phù hợp với từ khóa tìm kiếm.
4. Người dùng có thể chọn khoa từ danh sách kết quả và thực hiện các thao tác như xem chi tiết, sửa hoặc xóa.

#### \*Luồng thay thế:

- Thông tin khoa không hợp lệ:
  - Nếu người dùng nhập thông tin không hợp lệ (ví dụ: bỏ trống trường bắt buộc hoặc định dạng số điện thoại không đúng), hệ thống sẽ hiển thị thông báo lỗi và yêu cầu người dùng nhập lại thông tin đúng.

- Người dùng sửa lại thông tin và tiếp tục thực hiện lại thao tác.
  - Hệ thống xác minh lại và nếu hợp lệ, thông tin sẽ được thêm mới hoặc cập nhật vào hệ thống.
- Không tìm thấy khoa cần tìm kiếm:
    - Nếu người dùng nhập thông tin tìm kiếm không chính xác hoặc không có khoa nào thỏa mãn tiêu chí tìm kiếm, hệ thống sẽ hiển thị thông báo “Không tìm thấy kết quả phù hợp.”
    - Người dùng có thể điều chỉnh lại tiêu chí tìm kiếm và thực hiện lại thao tác tìm kiếm.
  - Hủy thao tác cập nhật thông tin khoa:
    - Trong quá trình thực hiện các thao tác (thêm, sửa, xóa), nếu người dùng chọn “Hủy bỏ”, hệ thống sẽ quay lại màn hình quản lý khoa mà không thực hiện bất kỳ thay đổi nào.
- Tiền điều kiện:
    - Người dùng (phòng đào tạo) phải đăng nhập vào hệ thống thành công và có đủ quyền truy cập để thực hiện các thao tác thêm, sửa, xóa, xem chi tiết và tìm kiếm thông tin khoa.
    - Hệ thống phải hoạt động bình thường, sẵn sàng tiếp nhận và xử lý yêu cầu từ người dùng.
  - Hậu điều kiện:
    - Nếu thành công:
      - Thông tin về khoa được thêm mới, sửa đổi, hoặc xóa sẽ được cập nhật vào cơ sở dữ liệu của hệ thống.
      - Danh sách khoa sẽ phản ánh những thay đổi này (nhân viên mới được thêm vào, khoa cũ được cập nhật thông tin, hoặc nhân viên bị xóa khỏi danh sách).
      - Kết quả tìm kiếm sẽ phản ánh đúng thông tin khoa dựa trên từ khóa người dùng nhập vào.
    - Nếu không thành công:
      - Hệ thống sẽ không thay đổi trạng thái dữ liệu.

- Người dùng sẽ nhận được thông báo lỗi hoặc hướng dẫn để sửa lại thông tin và thực hiện thao tác lại.

- Giao diện minh họa:

The screenshot shows a web-based application for managing academic departments. On the left is a dark sidebar with a user profile icon and the text "Admin User". Below it are navigation links: Trang chủ, Khoa (highlighted in orange), Sinh viên, Môn học, Tài khoản, and Đăng xuất. The main content area has the PHENIKAA UNIVERSITY logo at the top right. A search bar with placeholder "Nhập tên hoặc mã khoa" and a "Tim kiếm" button is followed by a table titled "Danh sách khoa". The table columns are ID, Mã Khoa, Tên Khoa, Quản lý, and Action. It lists four departments: CNTT (Công Nghệ Thông Tin), KHCB (Khoa học cơ bản), KRD (Ngôn Ngữ Hàn Quốc), and ĐÔNG PHƯƠNG HỌC (Manager User). Below the table are navigation buttons (< >), page number (1), and a dropdown for "5 / page". At the bottom center is the text "Team Nhóm 3 - PTTKPM - 2025 || Created by CX company".

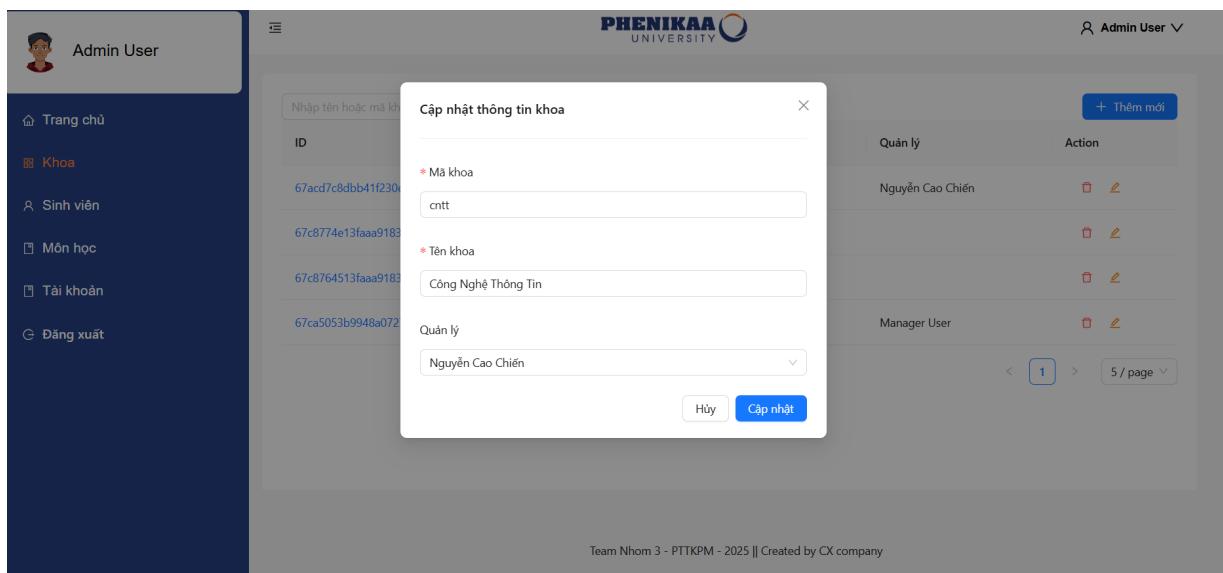
Hình 1.13 Giao diện chính quản lý khoa

This screenshot shows the "Thêm mới khoa" (Add new department) dialog box overlaid on the main interface. The dialog has fields for "Mã khoa" (Department code) and "Tên khoa" (Department name), both marked with a red asterisk indicating they are required. A dropdown menu for "Quản lý" (Manager) is shown with "Select manager". At the bottom are "Hủy" (Cancel) and "Tạo mới" (Create new) buttons. The background shows the same department list as in the previous screenshot.

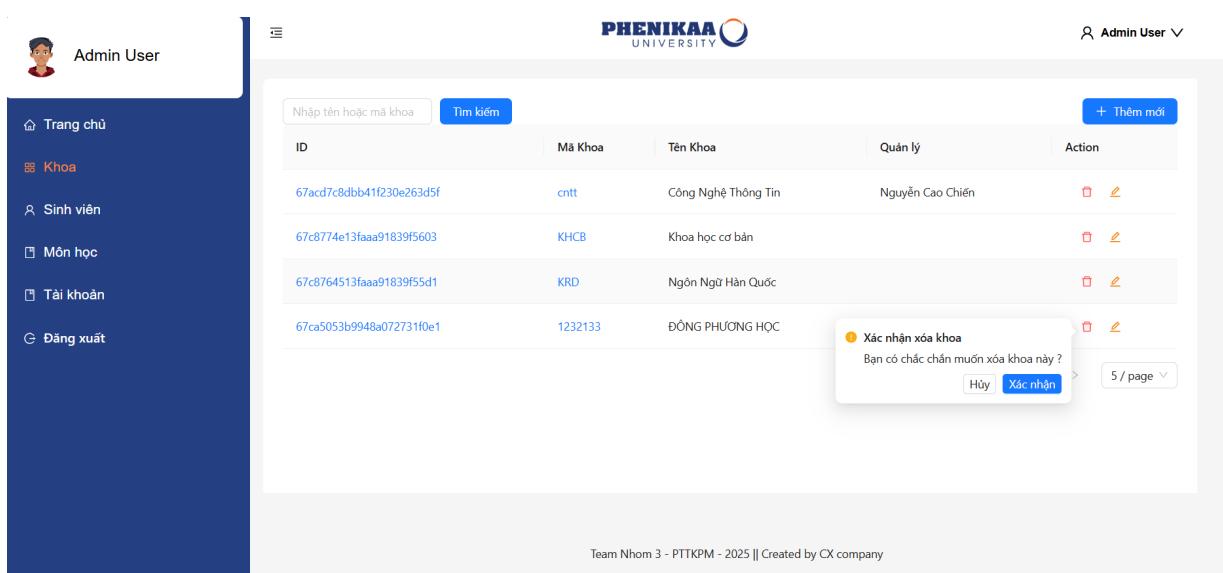
Hình 1.14 Giao diện thêm khoa

This screenshot shows the "Thêm mới khoa" dialog with validation errors. The "Mã khoa" field is empty and highlighted in red with the error message "Vui lòng nhập mã khoa!". The "Tên khoa" field is also empty and highlighted in red with the error message "Vui lòng nhập tên khoa!". The "Quản lý" dropdown is empty. The "Hủy" and "Tạo mới" buttons are visible at the bottom.

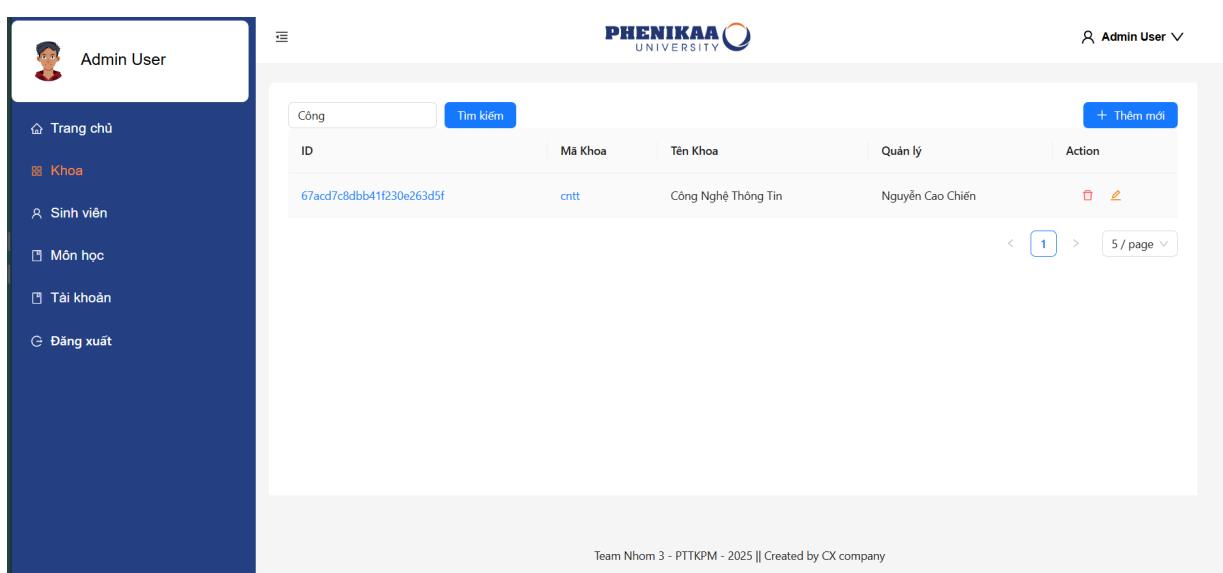
Hình 1.15 Thông báo lỗi thông tin không hợp lệ



Hình 1.16 Giao diện sửa thông tin khoa



Hình 1.17 Thông báo xác nhận xóa khoa



Hình 1.18 Giao diện tìm kiếm khoa

#### **1.4.4 UC4.1. Quản lý môn học**

Mô tả ngắn gọn: UC này cho phép người dùng quản lý thông tin các môn học trong hệ thống bao gồm: thêm, sửa, xoá, tìm kiếm, xem.

- Tác nhân thực hiện: Phòng đào tạo.
- Luồng sự kiện:

**\*Luồng cơ bản:** UC này hoạt động khi người dùng muốn thêm, sửa, xoá thông tin môn học trong hệ thống.

1. Người dùng chọn chức năng Quản lý môn học bằng cách bấm vào nút "Môn học" trên thanh công cụ.
2. Hệ thống yêu cầu người dùng chọn chức năng mà họ muốn thực hiện (thêm môn học, sửa môn học, xoá môn học).
3. Sau khi người dùng cung cấp thông tin được yêu cầu, một trong các luồng phụ sau sẽ thực hiện.
  - Nếu người dùng chọn “Thêm môn học”, luồng phụ S-1: Thêm môn sẽ được thực thi.
  - Nếu người dùng chọn “Sửa môn học”, luồng phụ S-2: Sửa môn học sẽ được thực thi.
  - Nếu người dùng chọn “Xoá môn học”, luồng phụ S-3: Xóa môn học sẽ được thực thi.
  - Nếu người dùng chọn “Tìm kiếm”, luồng phụ S-4: Tìm kiếm môn học sẽ được thực thi.

##### **S-1: Thêm sản phẩm:**

1. Hệ thống hiển thị form thêm mới sản phẩm và yêu cầu người dùng cung cấp các thông tin về sản phẩm bao gồm:
  - Tên môn học
  - Mã môn học
  - Số tín chỉ
  - Khoa
2. Sau khi người dùng cung cấp các thông tin được yêu cầu xong thì người dùng tiếp tục chọn “Lưu”.
3. Sau khi thêm thông tin về các môn học thì người dùng xác nhận việc thêm môn học đó vào danh sách.
4. Hệ thống xác nhận, lưu và hiển thị kết quả.

##### **S-2: Sửa môn học:**

1. Người dùng chọn môn học cần sửa và bấm "Sửa".
2. Hệ thống hiển thị form thông tin hiện tại.
3. Người dùng chỉnh sửa thông tin cần thiết.
4. Hệ thống xác minh và cập nhật thông tin.

#### **S-3: Xóa môn học:**

1. Người dùng chọn sản phẩm cần xoá và bấm nút “Xoá”.
2. Hệ thống hiển thị form xác nhận.
3. Người dùng xác nhận xóa sản phẩm đã chọn.
4. Hệ thống xác nhận và cập nhật vào hệ thống.

#### **S-4: Tìm kiếm môn học:**

1. Người dùng nhập từ khoá và bấm "Tìm kiếm".
2. Hệ thống hiển thị danh sách môn học khớp.
3. Người dùng chọn môn học cần xem.
4. Hệ thống hiển thị kết quả.

#### **\*Luồng thay thế:**

- Thông tin môn học được nhập vào không hợp lệ: Nếu người dùng cung cấp thông tin không hợp lệ trong các form (ví dụ: thông tin bắt buộc bị bỏ trống hoặc giá trị nhập vào không đúng định dạng), hệ thống sẽ hiển thị thông báo lỗi và yêu cầu người dùng kiểm tra và nhập lại thông tin đúng.
  - Người dùng sửa lại thông tin môn học và thực hiện lại hành động đó.
  - Hệ thống xác minh lại và nếu hợp lệ, môn học sẽ được thêm vào hệ thống/cập nhật thông tin mới vào hệ thống.
- Hủy thao tác cập nhật thông tin môn học: Nếu trong quá trình cập nhật (thêm, sửa, xoá), người dùng quyết định không muốn cập nhật thông tin môn học nữa, người dùng có thể chọn hủy thao tác đang thực hiện.
  - Hệ thống sẽ quay lại màn hình trước đó mà không thực hiện việc cập nhật thông tin môn học.
  - Luồng chính được khởi động lại từ đầu.
- Không có môn học khớp với từ khoá tìm kiếm:
  - Nếu không có môn học nào trong hệ thống khớp với từ khoá đã nhập, hệ thống sẽ hiển thị thông báo "Không tìm thấy kết quả phù hợp."
  - Người dùng có thể sửa lại từ khoá tìm kiếm hoặc hủy thao tác tìm kiếm.

- Hệ thống quay lại bước nhập từ khoá tìm kiếm.
- Người dùng không nhập từ khoá tìm kiếm:
  - Nếu người dùng không nhập tên môn học trong trường tìm kiếm và nhấn nút tìm kiếm, hệ thống sẽ hiển thị thông báo yêu cầu người dùng nhập thông tin.
  - Hệ thống quay lại bước nhập từ khoá tìm kiếm và chờ người dùng nhập thông tin cần thiết.

Tiền điều kiện:

- Người dùng (quản lý hoặc nhân viên) phải đăng nhập vào hệ thống thành công và có đủ quyền truy cập để thực hiện các thao tác thêm, sửa, xóa môn học.
- Hệ thống phải hoạt động bình thường, sẵn sàng tiếp nhận và xử lý yêu cầu từ người dùng.

Hậu điều kiện:

- Nếu thành công:
  - Thông tin về môn học được thêm, sửa, hoặc xóa sẽ được cập nhật vào cơ sở dữ liệu của hệ thống.
  - Danh sách môn học sẽ phản ánh những thay đổi này (môn học mới được thêm vào, môn học cũ được cập nhật thông tin, hoặc môn học bị xóa khỏi danh sách).
  - Hệ thống hiển thị danh sách môn học tương ứng với từ khoá tìm kiếm đã nhập.
  - Người dùng có thể xem chi tiết môn học hoặc thực hiện các thao tác khác trên môn học đó (nếu cần).
- Nếu không thành công:
  - Hệ thống sẽ không thay đổi trạng thái dữ liệu.
  - Người dùng sẽ nhận được thông báo lỗi hoặc hướng dẫn để sửa lại thông tin và thực hiện thao tác lại.
- Giao diện minh họa:

The screenshot shows the main dashboard of the course management system. On the left is a sidebar with a user profile icon and the text "Admin User". Below it are navigation links: Trang chủ, Khoa, Sinh viên, Môn học (highlighted in orange), Tài khoản, and Đăng xuất. The main content area features the PHENIKAA UNIVERSITY logo at the top right. Below it is a search bar with placeholder text "Nhập tên hoặc mã môn..." and a "Tim kiếm" button. A table lists courses with columns: ID, Mã môn học, Tên môn học, số tín chỉ, Khoa, and Action. The table contains four entries:

ID	Mã môn học	Tên môn học	số tín chỉ	Khoa	Action
67acd817dbb41f230e263d72	it001	Phân tích và thiết kế phần mềm	3	Công Nghệ Thông Tin	
67b0bfb904be1139020ee207	abc	Tiếng Anh 1	2	Công Nghệ Thông Tin	
67c8777c13faaa91839f560e	GT01	Giải tích 1	3	Khoa học cơ bản	
67c92c0d41e9e300e9258eb2	DSTT	Đại Số Tuyển Tính	3	Khoa học cơ bản	

At the bottom right of the main content area, there are navigation buttons: < (disabled), 1, >, and 5 / page.

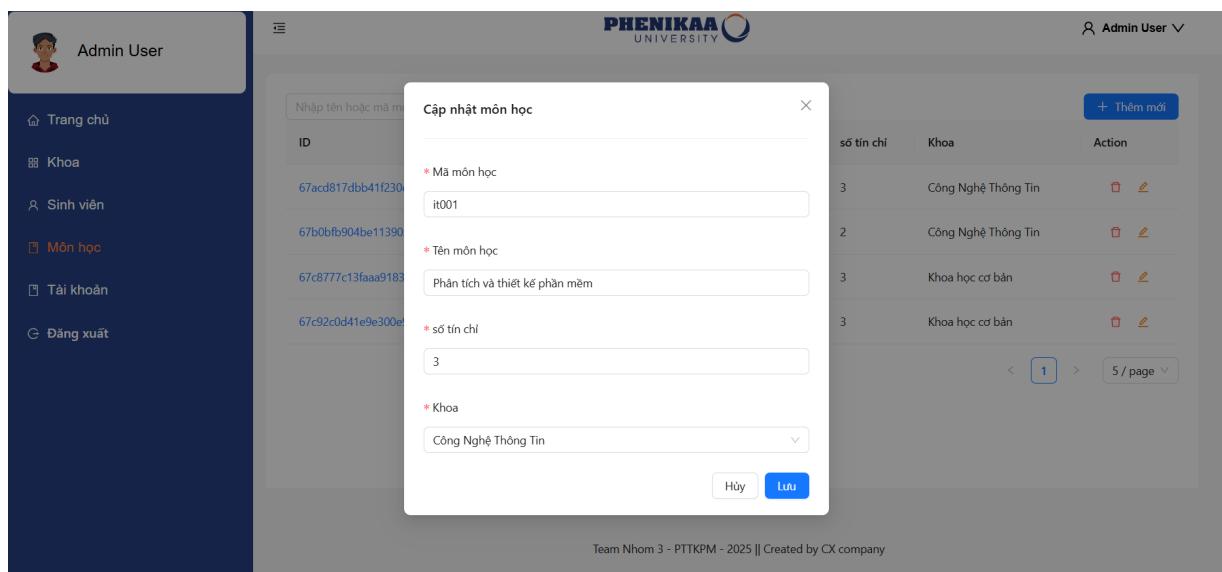
Hình 1.19 Giao diện chính của quản lý môn học

This screenshot shows the "Thêm mới môn học" (Add new course) dialog box. It has fields for ID (disabled), Mã môn học, Tên môn học, số tín chỉ, and Khoa. The "Khoa" field is a dropdown menu labeled "Select department". At the bottom are "Hủy" and "Tạo mới" buttons. In the background, the main course management interface is visible with a list of courses.

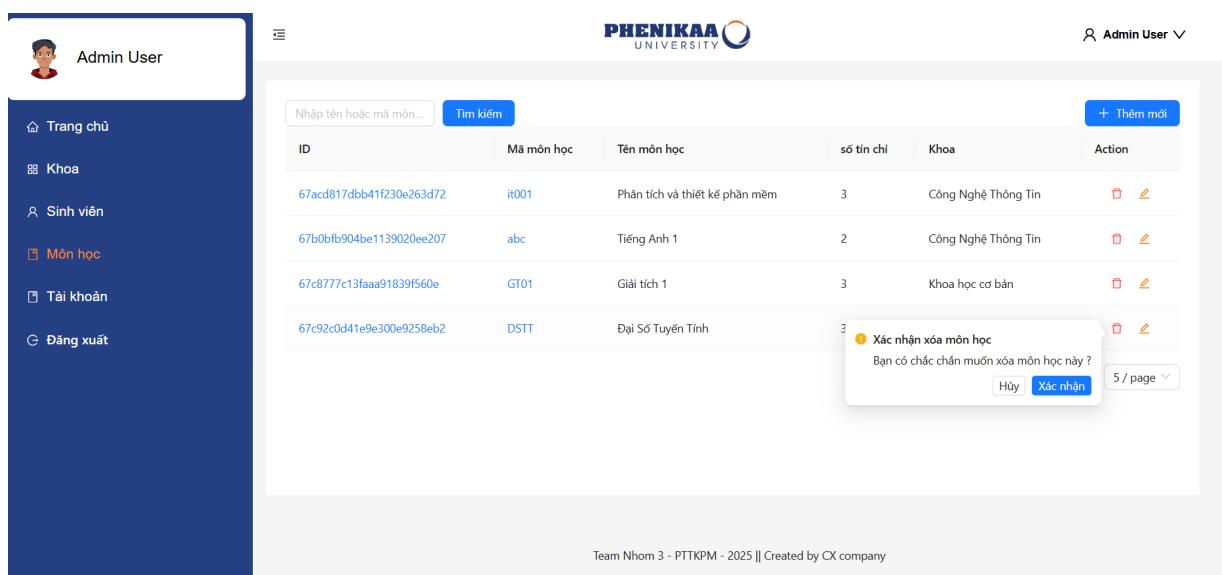
Hình 1.20 Giao diện thêm môn học mới

This screenshot shows the "Thêm mới môn học" dialog box with validation errors. The "Mã môn học" field is highlighted in red with the error message "Vui lòng nhập mã môn học!". The "số tín chỉ" field is also highlighted in red with the error message "Vui lòng nhập số tín chỉ!". The "Khoa" field dropdown is highlighted in red with the error message "Vui lòng nhập chọn khoa!". The "Tạo mới" button is disabled.

Hình 1.21 Thông báo lỗi thông tin không hợp lệ



Hình 1.22 Giao diện chỉnh sửa môn học



Hình 1.3 Thông báo xác nhận xóa môn học

#### 1.4.5 UC4.2. Tra cứu điểm

- Mô tả ngắn gọn: UC này cho phép người dùng quản lý điểm số của sinh viên, bao gồm nhập điểm, chỉnh sửa điểm, và xoá điểm của từng môn học trong hệ thống. Người dùng có thể cập nhật điểm theo từng kỳ học, môn học, hoặc theo từng sinh viên cụ
- Tác nhân thực hiện: Phòng đào tạo, sinh viên
- Luồng sự kiện:
  - \***Luồng cơ bản:** UC này hoạt động khi người dùng muốn thêm, sửa, xoá các giá trị của một thuộc tính nào đó của các sản phẩm có trong kho.
  - 1. Người dùng chọn nút “Thuộc tính” trên giao diện của hệ thống
  - 2. Người dùng chọn thuộc tính muốn thêm, sửa, xoá giá trị.

3. Hệ thống hiển thị giao diện của thuộc tính tương ứng và cho phép người dùng chọn chức năng mà họ muốn thực hiện với loại thuộc tính đó (thêm, sửa, xóa giá trị).

- Nếu người dùng chọn “Thêm”, luồng phụ S-1: Thêm giá trị sẽ được thực thi.
- Nếu người dùng chọn “Sửa”, luồng phụ S-2: Sửa giá trị sẽ được thực thi.
- Nếu người dùng chọn “Xoá”, luồng phụ S-3: Xóa giá trị sẽ được thực thi.

#### **S-1: Thêm giá trị:**

1. Người dùng nhập giá trị mới của loại thuộc tính đó
2. Người dùng chọn nút “Thêm”
3. Hệ thống thực hiện xác minh và lưu vào hệ thống.

#### **S-2: Sửa giá trị:**

1. Người dùng chọn giá trị thuộc tính muốn sửa
2. Người dùng nhập giá trị muốn sửa của loại thuộc tính đó
3. Người dùng chọn “Sửa”
4. Hệ thống xác minh và cập nhật thông tin mới.

#### **S-3: Xóa giá trị:**

1. Người dùng chọn giá trị cần xoá và bấm nút “Xoá”.
2. Hệ thống hiển thị form xác nhận.
3. Người dùng xác nhận xóa giá trị đã chọn.
4. Hệ thống xác nhận và cập nhật vào hệ thống.

#### **\*Luồng thay thế:**

Điểm số nhập vào không hợp lệ:

- Nếu người dùng nhập điểm ngoài phạm vi cho phép hoặc sai định dạng (ví dụ: nhập chữ thay vì số), hệ thống sẽ hiển thị thông báo lỗi và yêu cầu nhập lại.
- Người dùng điều chỉnh thông tin và thực hiện lại thao tác.
- Hệ thống kiểm tra lại và nếu hợp lệ, điểm số sẽ được cập nhật.

Hủy thao tác xóa điểm:

- Nếu trong quá trình xóa điểm, người dùng thay đổi ý định, họ có thể hủy thao tác.
- Hệ thống sẽ quay lại màn hình trước đó mà không thay đổi dữ liệu.

- Hủy thao tác xóa giá trị thuộc tính của sản phẩm Nếu trong quá trình xóa một giá trị của thuộc tính nào đó, người dùng quyết định không muốn xóa giá trị đó nữa, người dùng có thể chọn hủy thao tác đang thực hiện.
- Tiên điều kiện:
- Người dùng (phòng đào tạo) phải đăng nhập vào hệ thống thành công và có đủ quyền truy cập để thực hiện các thao tác thêm, sửa, xóa các giá trị của thuộc tính.
  - Sinh viên chỉ có quyền xem điểm
  - Hệ thống phải hoạt động bình thường, sẵn sàng tiếp nhận và xử lý yêu cầu từ người dùng.
- Hậu điều kiện:
- Nếu thành công:
    - Điểm số của sinh viên được cập nhật vào hệ thống
    - Danh sách điểm sẽ phản ánh mới nhất
  - Nếu không thành công:
    - Dữ liệu điểm sẽ không bị thay đổi
    - Hệ thống hiển thị thông báo lỗi và hướng dẫn người dùng chỉnh sửa thông tin.

- Giao diện minh họa:

Tổng tín chỉ	Tổng tín chỉ tích lũy
8	8
<b>Điểm trung bình hệ 10</b>	<b>Điểm trung bình hệ 4</b>
8.39	3.56
<b>Điểm trung bình tích lũy hệ 10</b>	<b>Điểm trung bình tích lũy hệ 4</b>
8.39	3.56

Bảng Điểm							
Môn học	Chuyên cần	Giữa kỳ	Cuối kỳ	Hệ 4	Hệ 10	Trạng thái	
Phân tích và thiết kế phần mềm	1	10	10	4	9.1	<b>Đạt</b>	
Tiếng Anh 1	10	6	8	3	7.6	<b>Đạt</b>	
Giải tích 1	10	8	8	3.5	8.2	<b>Đạt</b>	

Team Nhóm 3 - PTTKPM - 2025 || Created by CX company

*Hình 1.23 Giao diện chính của tra cứu điểm*

#### **1.4.6 UC5.1. Quản lý thông tin cá nhân**

- Mô tả ngắn gọn: UC này cho phép sinh viên cập nhật và xem thông tin cá nhân của mình trong hệ thống. Các thông tin bao gồm: họ tên, ngày sinh, giới tính, địa chỉ, số điện thoại, email, lớp, khoa và các thông tin liên quan khác.
- Tác nhân thực hiện: Sinh viên, Phòng đào tạo
- Luồng sự kiện:

**\*Luồng cơ bản:** UC này hoạt động khi sinh viên muốn xem hoặc cập nhật thông tin cá nhân.

1. Sinh viên chọn chức năng "Quản lý thông tin cá nhân" trên giao diện hệ thống.
2. Hệ thống hiển thị thông tin cá nhân hiện có của sinh viên.
3. Sinh viên có thể thực hiện các thao tác sau:
  - Nếu sinh viên chỉ muốn xem thông tin, không cần thao tác thêm.
  - Nếu sinh viên muốn cập nhật thông tin, luồng phụ S-1: Cập nhật thông tin sẽ được thực thi
4. Hệ thống lưu thông tin vào cơ sở dữ liệu hoặc cập nhật/xóa thông tin cá nhân dựa trên thao tác của người dùng.

**\*Luồng phụ:**

#### S-1: Cập nhật thông tin cá nhân:

1. Hệ thống hiển thị form yêu cầu người dùng cung cấp các thông tin cá nhân như: số điện thoại, email, địa chỉ.
2. Sinh viên nhập thông tin vào hệ thống.
3. Sau khi chỉnh sửa xong bấm “Lưu” để cập nhật thông tin.
4. Hệ thống sẽ xác minh:
  - a. Nếu hợp lệ, thông tin sẽ được lưu mới vào cơ sở dữ liệu.
  - b. Nếu không hợp lệ, hệ thống sẽ hiển thị và yêu cầu nhập lại dữ liệu.

**\*Luồng thay thế:**

- Thông tin nhập vào không hợp lệ: Nếu người dùng cung cấp thông tin không hợp lệ trong form (ví dụ: số điện thoại không đúng, email không hợp lệ), hệ thống sẽ hiển thị thông báo lỗi và yêu cầu nhập lại thông tin.
- Hủy thao tác: Trong quá trình thêm, sửa, xóa hoặc xem nếu người dùng quyết định không thực hiện nữa, có thể chọn hủy thao tác. Hệ thống sẽ quay lại màn hình trước mà không lưu thay đổi.
- Tiền điều kiện:
  - Người dùng đã đăng nhập thành công vào hệ thống và có quyền truy cập vào chức năng quản lý thông tin cá nhân.

- Thông tin của sinh viên phải tồn tại trong dữ liệu hệ thống.
- Hậu điều kiện:
  - Nếu thành công: Thông tin cá nhân sẽ được thêm, sửa hoặc xóa thành công trong cơ sở dữ liệu.
  - Nếu không thành công: Hệ thống sẽ không thay đổi dữ liệu và người dùng sẽ nhận được thông báo lỗi hoặc hướng dẫn để thực hiện lại thao tác.
- Giao diện minh họa:

The screenshot shows a user profile page for Nguyen Cao Chien. At the top, there is a logo for PHENIKAA UNIVERSITY and a search bar with the name 'Nguyen Cao Chien'. Below the header, there is a section titled 'THÔNG TIN CƠ BẢN' (Basic Information) with a decorative background image of a hand holding a four-leaf clover against a blue sky with clouds. The profile information is organized into two columns:

Mã SV	Tên	Ngày sinh
22010014	Nguyen Cao Chien	2003-01-23
Email	Lớp	Khoá đào tạo
caochien@gmail.com	K16-CNTT_4	DH_K16.40
Khoa quản lý	Ngành	Niên khoá
Công Nghệ Thông Tin	Công Nghệ Thông Tin	2022-2026
Trạng thái	Chức vụ	Giới tính
Đang học	Sinh viên	Nam
Hệ đào tạo	Chương trình đào tạo	
Đại học chính quy	Công nghệ thông tin	

At the bottom of the page, it says 'Team Nhóm 3 - PTTKPM - 2025 || Created by CX company'.

*Hình 1.24 Giao diện chính của quản lý thông tin cá nhân*

#### **1.4.7 UC5.2. Quản lý thông tin sinh viên**

- Mô tả ngắn gọn: UC này cho phép Phòng Đào Tạo và Quản lý Khoa quản lý thông tin sinh viên trong hệ thống. Các chức năng bao gồm: thêm mới sinh viên, chỉnh sửa thông tin sinh viên, xem danh sách sinh viên, tìm kiếm sinh viên theo các tiêu chí (họ tên, mã sinh viên, lớp, khoa), và xóa sinh viên khỏi hệ thống nếu cần thiết.
- Tác nhân thực hiện: Phòng đào tạo, quản lý khoa.
- Luồng sự kiện:

**\*Luồng cơ bản:** UC này hoạt động khi Phòng Đào Tạo hoặc Quản lý Khoa muốn thực hiện các thao tác quản lý thông tin sinh viên.

1. Người dùng (Phòng Đào Tạo hoặc Quản lý Khoa) chọn chức năng "Quản lý thông tin sinh viên" trên giao diện hệ thống.
2. Hệ thống hiển thị danh sách sinh viên hiện có trong hệ thống.
3. Người dùng có thể thực hiện các thao tác sau:
  - Nếu người dùng muốn thêm sinh viên mới, luồng phụ S-1: Thêm sinh viên mới sẽ được thực thi.
  - Nếu người dùng muốn chỉnh sửa thông tin sinh viên, luồng phụ S-2: Chính sửa thông tin sinh viên sẽ được thực thi.
  - Nếu người dùng muốn xóa sinh viên, luồng phụ S-3: Xóa sinh viên sẽ được thực thi.
4. Hệ thống lưu thông tin vào cơ sở dữ liệu hoặc cập nhật/xóa thông tin sinh viên dựa trên thao tác của người dùng.

**\*Luồng phụ:**

##### **S-1: Thêm sinh viên mới:**

1. Người dùng chọn nút "Thêm sinh viên"
2. Hệ thống hiển thị biểu mẫu nhập thông tin sinh viên (họ tên, mã sinh viên, ngày sinh, lớp, khoa, email, số điện thoại).
3. Người dùng nhập thông tin và chọn "Lưu".
4. Hệ thống xác minh dữ liệu:
  - Nếu hợp lệ, hệ thống lưu thông tin sinh viên mới vào cơ sở dữ liệu.
  - Nếu không hợp lệ, hệ thống hiển thị thông báo lỗi và yêu cầu nhập lại thông tin.

## **S-2: Chính sửa thông tin sinh viên:**

1. Người dùng chọn sinh viên cần chỉnh sửa từ danh sách.
2. Hệ thống hiển thị biểu mẫu với thông tin hiện tại của sinh viên.
3. Người dùng chỉnh sửa thông tin cần thay đổi.
4. Người dùng chọn "Cập nhật".
5. Hệ thống xác minh dữ liệu và cập nhật vào hệ thống.

## **S-3: Xóa sinh viên:**

1. Người dùng chọn sinh viên cần xóa.
2. Hệ thống hiển thị form xác nhận việc xóa sinh viên.
3. Sau khi người dùng xác nhận, hệ thống xóa thông tin sinh viên khỏi cơ sở dữ liệu.

### **\*Luồng thay thế:**

- Thông tin nhập vào không hợp lệ: Nếu người dùng cung cấp thông tin không hợp lệ trong form (ví dụ: email không hợp lệ, thiếu mã sinh viên), hệ thống sẽ hiển thị thông báo lỗi và yêu cầu nhập lại thông tin.
  - Hủy thao tác: Trong quá trình thêm, sửa hoặc xóa, nếu người dùng quyết định không thực hiện nữa, có thể chọn hủy thao tác. Hệ thống sẽ quay lại màn hình trước mà không lưu thay đổi.
- Tiền điều kiện:
- Người dùng đã đăng nhập thành công vào hệ thống và có quyền truy cập vào chức năng quản lý xuất kho.
  - Thông tin sinh viên phải tồn tại trong hệ thống.
- Hậu điều kiện:
- Nếu thành công: Thông tin sinh viên sẽ được thêm, sửa hoặc xóa thành công trong cơ sở dữ liệu.
  - Nếu không thành công: Hệ thống sẽ không thay đổi dữ liệu và người dùng sẽ nhận được thông báo lỗi hoặc hướng dẫn để thực hiện lại thao tác.

- Giao diện minh họa:

The screenshot shows the main dashboard of a student management system. On the left, a sidebar menu for 'Admin User' includes links for Trang chủ, Khoa, Sinh viên, Môn học, Tài khoản, and Đăng xuất. The main content area displays a table of student records with columns: Mã sinh viên, Họ và tên, Email, Giới tính, Ngày sinh, Khoa, Năm nhập học, and Action. The table contains several entries, such as '22010014 Nguyen Cao Chien caochien@gmail.com Nam 2003-01-23 Công Nghệ Thông Tin 2022'. At the bottom right of the main area, there is a footer note: 'Team Nhóm 3 - PTTKPM - 2025 || Created by CX company'.

Hình 1.25 Giao diện chính của quản lý sinh viên

This screenshot shows a modal dialog titled 'Thêm mới sinh viên' (Add new student). It contains fields for Mã sinh viên (student ID), Tên sinh viên (student name), Email, Giới tính (gender), Ngày sinh (date of birth), and Khoa (faculty). The background shows the same student list as in Figure 1.25, with the footer 'CX company' visible.

Hình 1.26 Giao diện tạo mới sinh viên

This screenshot shows the 'Thêm mới sinh viên' modal with validation errors. The 'Email' field has an error message 'Vui lòng nhập email!'. The 'Giới tính' (Gender) dropdown has an error message 'Vui lòng chọn giới tính!'. The 'Ngày sinh' (Date of birth) field has an error message 'Vui lòng nhập ngày sinh!'. The 'Khoa' (Faculty) dropdown has an error message 'Vui lòng chọn khoa!'. The 'Năm nhập học' (Year of entry) field has an error message 'Vui lòng nhập năm nhập học!'. Below these fields is a file upload section with a placeholder 'Ảnh sinh viên:' and a 'Upload' button. The background shows the student list and the footer 'CX company'.

Hình 1.27 Form khi nhập thiếu thông tin

Hình 1.28 Giao diện sửa thông tin sinh viên

Hình 1.29 Thông báo xác nhận xóa phiếu xuất

Hình 1.30 Giao diện xem thông tin sinh viên

#### **1.4.8 UC6.1 Quản lý điểm sinh viên**

- Mô tả ngắn gọn: UC này cho phép Phòng Đào Tạo và Quản lý Khoa nhập, chỉnh sửa, xem, tìm kiếm và xóa điểm của sinh viên trong hệ thống. Các chức năng bao gồm: nhập điểm theo môn học, cập nhật điểm khi có sai sót, tra cứu điểm theo mã sinh viên/lớp/khoa, và xóa điểm nếu có sai phạm.
- Tác nhân thực hiện: Phòng đào tạo, quản lý khoa.
- Luồng sự kiện:

**\*Luồng cơ bản:** UC này hoạt động khi Phòng Đào Tạo hoặc Quản lý Khoa muốn thực hiện các thao tác quản lý điểm sinh viên.

1. Người dùng chọn chức năng "Quản lý điểm sinh viên" trên giao diện hệ thống.
2. Hệ thống hiển thị danh sách sinh viên và điểm số tương ứng.
3. Sau khi người dùng chọn chức năng, một trong các luồng phụ sau sẽ được thực hiện:
  - o Nếu người dùng muốn nhập điểm mới, luồng phụ S-1: Nhập điểm mới sẽ được thực thi.
  - o Nếu người dùng muốn chỉnh sửa điểm, luồng phụ S-2: Chỉnh sửa điểm sẽ được thực thi.
  - o Nếu người dùng chọn xóa điểm, luồng phụ S-3: Xóa điểm sẽ được thực thi.

**\*Luồng phụ:**

##### **S-1: Nhập điểm mới:**

1. Người dùng chọn môn học và danh sách sinh viên cần nhập điểm.
2. Hệ thống hiển thị biểu mẫu nhập điểm (gồm điểm giữa kỳ, điểm cuối kỳ, điểm tổng kết).
3. Người dùng nhập điểm và chọn "Lưu".
4. Hệ thống xác nhận việc thêm điểm thành công và cập nhật điểm.

##### **S-2: Chỉnh sửa điểm:**

1. Người dùng chọn sinh viên cần chỉnh sửa điểm.
2. Hệ thống hiển thị thông tin điểm hiện tại.

3. Người dùng nhập giá trị điểm mới.
4. Sau khi sửa xong, bấm nút “Lưu thông tin”.
5. Hệ thống kiểm tra tính hợp lệ của thông tin và cập nhật thông tin mới vào cơ sở dữ liệu.

### S-3: Xóa điểm:

1. Người dùng chọn sinh viên cần xóa điểm.
2. Hệ thống hiển thị thông báo xác nhận.
3. Người dùng xác nhận xóa điểm.
4. Hệ thống xóa điểm khỏi cơ sở dữ liệu.

### \*Luồng thay thế:

- Thông tin điểm không hợp lệ:
  - Nếu người dùng nhập thông tin không hợp lệ (ví dụ: điểm ngoài phạm vi, sai định dạng điểm), hệ thống sẽ hiển thị thông báo lỗi và yêu cầu người dùng nhập lại thông tin đúng.
  - Người dùng sửa lại thông tin và tiếp tục thực hiện lại thao tác.
  - Hệ thống xác minh lại và nếu hợp lệ, thông tin sẽ được thêm mới hoặc cập nhật vào hệ thống.
- Hủy thao tác cập nhật điểm:
  - Trong quá trình thực hiện các thao tác (thêm, sửa, xóa), nếu người dùng chọn “Hủy bỏ”, hệ thống sẽ không thay đổi dữ liệu.
- Tiền điều kiện:
  - Người dùng (quản lý khoa và phòng đào tạo) phải đăng nhập vào hệ thống thành công và có đủ quyền truy cập để thực hiện các thao tác thêm, sửa, xóa, xem chi tiết và tìm kiếm điểm.
  - Hệ thống phải hoạt động bình thường, sẵn sàng tiếp nhận và xử lý yêu cầu từ người dùng.
- Hậu điều kiện:
  - Nếu thành công:
    - Thông tin điểm được thêm mới, sửa đổi, hoặc xóa sẽ được cập nhật vào cơ sở dữ liệu của hệ thống.

- Kết quả tìm kiếm sẽ phản ánh đúng điểm dựa trên từ khóa người dùng nhập vào.
- Nếu không thành công:
  - Hệ thống sẽ không thay đổi trạng thái dữ liệu.
  - Người dùng sẽ nhận được thông báo lỗi hoặc hướng dẫn để sửa lại thông tin và thực hiện thao tác lại.

Giao diện minh họa:

The screenshot shows a user interface for managing grades. On the left, a sidebar menu for an 'Admin User' includes links for Trang chủ, Khoa, Sinh viên, Môn học, Tài khoản, and Đăng xuất. The main content area displays a table titled 'Bảng Điểm' (Grade Table) with the following data:

Môn học	Chuyên cần	Giữa kỳ	Cuối kỳ	Hệ 4	Hệ 10	Trạng thái	Hành động
Phân tích và thiết kế phần mềm	1	10	10	4	9.1	Đạt	<button>Chỉnh sửa</button>
Tiếng Anh 1	10	6	8	3	7.6	Đạt	<button>Chỉnh sửa</button>
Giải tích 1	10	8	8	3.5	8.2	Đạt	<button>Chỉnh sửa</button>

A message 'No data' is visible above the table. At the bottom right of the main content area, there is a footer note: 'Team Nhóm 3 - PTTKPM - 2025 || Created by CX company'.

Hình 1.31 Giao diện xem thông tin chỉnh sửa điểm

This screenshot shows a modal dialog titled 'Chỉnh sửa điểm' (Edit Grade) overlaid on the grade table from the previous screenshot. The dialog contains fields for 'Chuyên cần' (1), 'Giữa kỳ' (10), and 'Cuối kỳ' (10). Below the dialog, the grade table shows the updated values: Chuyên cần 1, Giữa kỳ 10, Cuối kỳ 10, Hệ 4 4, Hệ 10 9.1, Trạng thái Đạt, and Hành động (button).

Hình 1.32 Giao diện chỉnh sửa điểm

#### 1.4.9 UC6.1 Lịch học, lịch thi

##### UC1.4. Quản lý Lịch học/Lịch thi

**Mô tả ngắn gọn:**

UC này cho phép Phòng Đào Tạo và Quản lý Khoa nhập, chỉnh sửa, xem, tìm kiếm và xóa lịch học/lịch thi trong hệ thống. Các chức năng bao gồm: tạo mới lịch học/lịch thi theo môn học, cập nhật khi có thay đổi, tra cứu lịch theo mã sinh viên/lớp/khoa, và xóa lịch nếu có sai phạm.

#### **Tác nhân thực hiện:**

Phòng đào tạo, Quản lý khoa.

#### **Luồng sự kiện:**

##### ***Luồng cơ bản:***

UC này hoạt động khi Phòng Đào Tạo hoặc Quản lý Khoa muốn thực hiện các thao tác quản lý lịch học/lịch thi.

1. Người dùng chọn chức năng "Quản lý Lịch học/Lịch thi" trên giao diện hệ thống.
2. Hệ thống hiển thị danh sách lịch học/lịch thi hiện có.
3. Sau khi người dùng chọn chức năng, một trong các luồng phụ sau sẽ được thực hiện:
  - o Nếu người dùng muốn tạo lịch học/lịch thi mới, luồng phụ S-1: Tạo lịch mới sẽ được thực thi.
  - o Nếu người dùng muốn chỉnh sửa lịch học/lịch thi, luồng phụ S-2: Chỉnh sửa lịch sẽ được thực thi.
  - o Nếu người dùng muốn xóa lịch học/lịch thi, luồng phụ S-3: Xóa lịch sẽ được thực thi.

#### ***Luồng phụ:***

##### **S-1: Tạo lịch mới**

1. Người dùng chọn môn học và lớp học cần tạo lịch.
2. Hệ thống hiển thị biểu mẫu nhập thông tin lịch (ngày, giờ, phòng học, hình thức thi nếu có).
3. Người dùng nhập thông tin và chọn "Lưu".
4. Hệ thống kiểm tra tính hợp lệ của thông tin và lưu lịch mới vào cơ sở dữ liệu.
5. Hệ thống xác nhận việc tạo lịch thành công và cập nhật danh sách lịch.

##### **S-2: Chỉnh sửa lịch**

1. Người dùng chọn lịch cần chỉnh sửa.
2. Hệ thống hiển thị thông tin chi tiết của lịch học/lịch thi.
3. Người dùng chỉnh sửa thông tin cần thiết và chọn "Lưu thông tin".
4. Hệ thống kiểm tra tính hợp lệ và cập nhật thông tin mới vào cơ sở dữ liệu.
5. Hệ thống xác nhận việc chỉnh sửa thành công.

### S-3: Xóa lịch

1. Người dùng chọn lịch cần xóa.
2. Hệ thống hiển thị thông báo xác nhận.
3. Người dùng xác nhận xóa lịch.
4. Hệ thống xóa lịch khỏi cơ sở dữ liệu và cập nhật danh sách lịch.

### Luồng thay thế:

- **Thông tin lịch không hợp lệ:**
  - Nếu người dùng nhập thông tin không hợp lệ (ví dụ: trùng lịch, phòng học không khả dụng), hệ thống sẽ hiển thị thông báo lỗi và yêu cầu nhập lại.
  - Người dùng sửa lại thông tin và tiếp tục thao tác.
  - Hệ thống xác minh lại và nếu hợp lệ, thông tin sẽ được lưu vào hệ thống.
- **Hủy thao tác cập nhật lịch:**
  - Trong quá trình thực hiện các thao tác (tạo, sửa, xóa), nếu người dùng chọn "Hủy bỏ", hệ thống sẽ không thay đổi dữ liệu.

### Tiền điều kiện:

- Người dùng (quản lý khoa và phòng đào tạo) phải đăng nhập vào hệ thống thành công và có đủ quyền truy cập để thực hiện các thao tác thêm, sửa, xóa, xem chi tiết và tìm kiếm lịch học/lịch thi.
- Hệ thống phải hoạt động bình thường, sẵn sàng tiếp nhận và xử lý yêu cầu từ người dùng.

### Hậu điều kiện:

- **Nếu thành công:**
  - Thông tin lịch học/lịch thi được thêm mới, sửa đổi, hoặc xóa sẽ được cập nhật vào cơ sở dữ liệu của hệ thống.
  - Kết quả tìm kiếm sẽ phản ánh đúng thông tin lịch dựa trên từ khóa người dùng nhập vào.

- **Nếu không thành công:**

- Hệ thống sẽ không thay đổi trạng thái dữ liệu.
- Người dùng sẽ nhận được thông báo lỗi hoặc hướng dẫn để sửa lại thông tin và thực hiện thao tác lại.

Mã lớp	Tên môn học	Thời gian	Phòng học	Giảng viên	Trạng thái
CSE101	Nhập môn lập trình	Thứ 2 (7:00 - 10:00)	A2-301	TS. Nguyễn Văn A	Đang diễn ra

*Hình 1.33 Giao diện xem thông tin lịch thi*

Mã lớp	Tên môn học	Thời gian	Phòng học	Giảng viên	Trạng thái
CSE101	Nhập môn lập trình	Thứ 2 (7:00 - 10:00)	A2-301	TS. Nguyễn Văn A	Đang diễn ra

*Hình 1.33 Giao diện tìm kiếm lịch*

The screenshot shows a university management system interface. On the left, there is a sidebar with a user profile picture and name ('Nguyen Cao Chien'), and a navigation menu with links like 'Trang chủ', 'Profile', 'đăng ký môn học', 'Lịch học/Lịch thi', and 'Đăng xuất'. The main content area has a header 'PHENIKAA UNIVERSITY' and a search bar with placeholder 'Tra cứu thông tin lịch học, lịch thi theo môn học hoặc thời gian'. Below the search bar, there are two tabs: 'Lịch học' and 'Lịch thi', with 'Lịch thi' being active. A search form below the tabs includes fields for 'nhập môn' (subject), 'Từ ngày' (from date), 'Đến ngày' (to date), 'Hình thức thi' (exam type), and a 'Tim kiếm' (Search) button. A table lists exam details:

Mã lớp	Tên môn học	Ngày thi	Giờ thi	Phòng thi	Hình thức thi
CSE101	Nhập môn lập trình	15/05/2025	08:00 - 10:00	H1-301	Trắc nghiệm
MAT202	Đại số tuyến tính	20/05/2025	13:00 - 15:00	H2-401	Tư luận
ENG305	Tiếng Anh chuyên ngành	25/05/2025	07:30 - 09:30	H3-201	Vấn đáp

At the bottom of the page, there is a footer note: 'Team Nhóm 3 - PTTKPM - 2025 || Created by CX company'.

*Hình 1.33 Giao diện lịch thi*

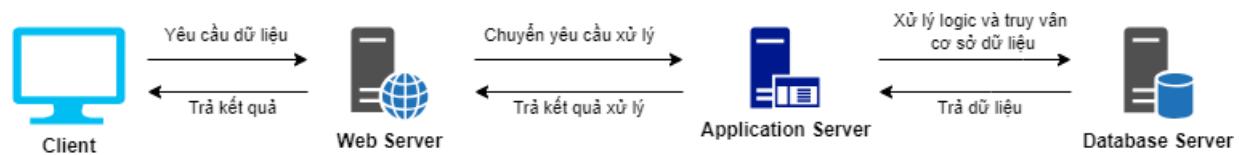
This screenshot shows the same university management system interface as Figure 1.33, but the table only displays one row of data for the exam on May 15th. The rest of the interface, including the sidebar and search bar, remains the same.

*Hình 1.34 Giao diện tìm kiếm lịch thi*

## 2. Phân tích trường hợp sử dụng (Use-case analysis)

### 2.1. Phân tích kiến trúc hệ thống

#### 2.1.1. Kiến trúc mức cao của hệ thống



*Hình 2.1 Sơ đồ kiến trúc client-server*

\*Vai trò của từng thành phần:

- **Client:** là thiết bị đầu cuối mà người dùng sử dụng để tương tác với hệ thống. Nó có thể là máy tính, điện thoại di động, hoặc bất kỳ thiết bị nào có thể truy cập vào ứng dụng thông qua giao diện web hoặc ứng dụng di động. Client đóng vai trò gửi các yêu cầu của người dùng đến hệ thống, như xem sản phẩm trong kho hoặc kiểm tra chi tiết sản phẩm, và nhận lại kết quả phản hồi từ hệ thống.
- **Web Server:** đóng vai trò như một lớp trung gian giữa Client và Application Server. Nó chịu trách nhiệm tiếp nhận các yêu cầu HTTP từ Client và truyền tải những yêu cầu này tới Application Server để xử lý. Ngoài ra, Web Server cũng đảm bảo trả về kết quả từ Application Server tới Client. Đây là thành phần đảm bảo kết nối giữa người dùng cuối và hệ thống, cho phép người dùng tương tác với hệ thống một cách mượt mà.
- **Application Server:** chịu trách nhiệm chính trong việc xử lý các nghiệp vụ và logic phức tạp của hệ thống. Nó quản lý các quy trình xử lý chính, điều phối và gửi yêu cầu đến Database Server khi cần truy xuất hoặc cập nhật dữ liệu. Application Server là nơi diễn ra hầu hết các thao tác xử lý như tính toán và xử lý thông tin về số lượng hàng còn lại trong kho khi có yêu cầu từ Web Server, quản lý thông tin sản phẩm và đáp ứng các yêu cầu từ Web Server.
- **Database Server:** đóng vai trò là nơi lưu trữ dữ liệu của toàn bộ hệ thống, bao gồm thông tin về sản phẩm, kho hàng, lịch sử giao dịch và người dùng. Nó đảm bảo việc truy xuất và cung cấp dữ liệu chính xác khi có yêu cầu từ Application Server. Mọi thông tin được cập nhật hoặc truy vấn trong hệ thống đều được quản lý bởi Database Server.

#### **2.1.2. Các đối tượng trừu tượng chính của hệ thống (Key abstractions)**

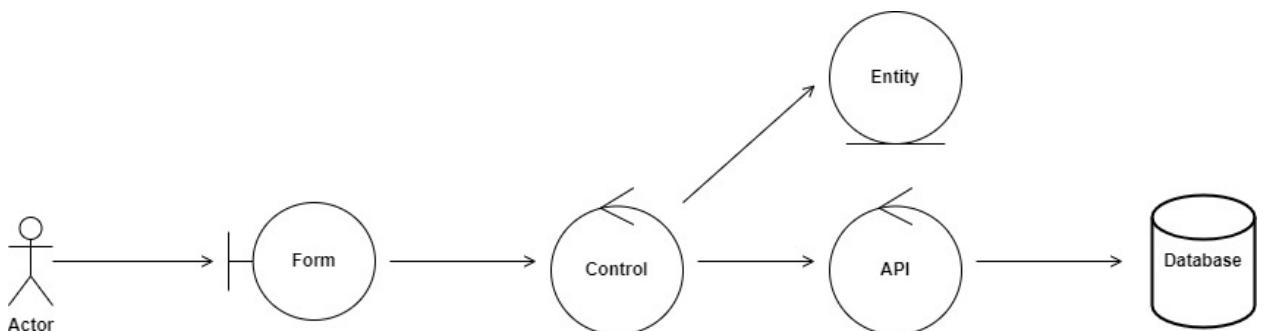
STT	Tên đối tượng	Vai trò	Thuộc tính	Phương thức
1	Sinh viên	Đại diện cho thông tin chung của các sinh viên.	Mã sinh viên, họ và tên, ngày sinh, giới tính, email, số điện thoại, khoa	Đăng ký môn học, tra cứu điểm và cập nhật thông tin
2	Môn học	Quản lý thông tin các môn học	Mã môn học, số tín chỉ, Khoa quản lý, tên môn học	Thêm sửa xóa môn học

3	Điểm	Quản lý điểm sinh viên	Mã sinh viên, mã môn học, điểm	Nhập điểm, cập nhật điểm và tra cứu điểm
4	Phòng đào tạo	Quản lý các hoạt động của khoa và sinh viên	Mã sinh viên, chức vụ, họ tên, tài khoản đăng nhập, khoa, môn học	Quản lý các khoa, môn học, tài khoản, thông tin sinh viên, điểm sinh viên
5	Quản lý khoa	Quản lý thông tin của khoa	Mã khoa, tên khoa, danh sách môn học	Quản lý thông tin khoa, danh sách môn học.

Bảng 2.1 Các đối tượng trùu tượng chính của hệ thống

## 2.2. Thực thi trường hợp sử dụng (Use-case realizations)

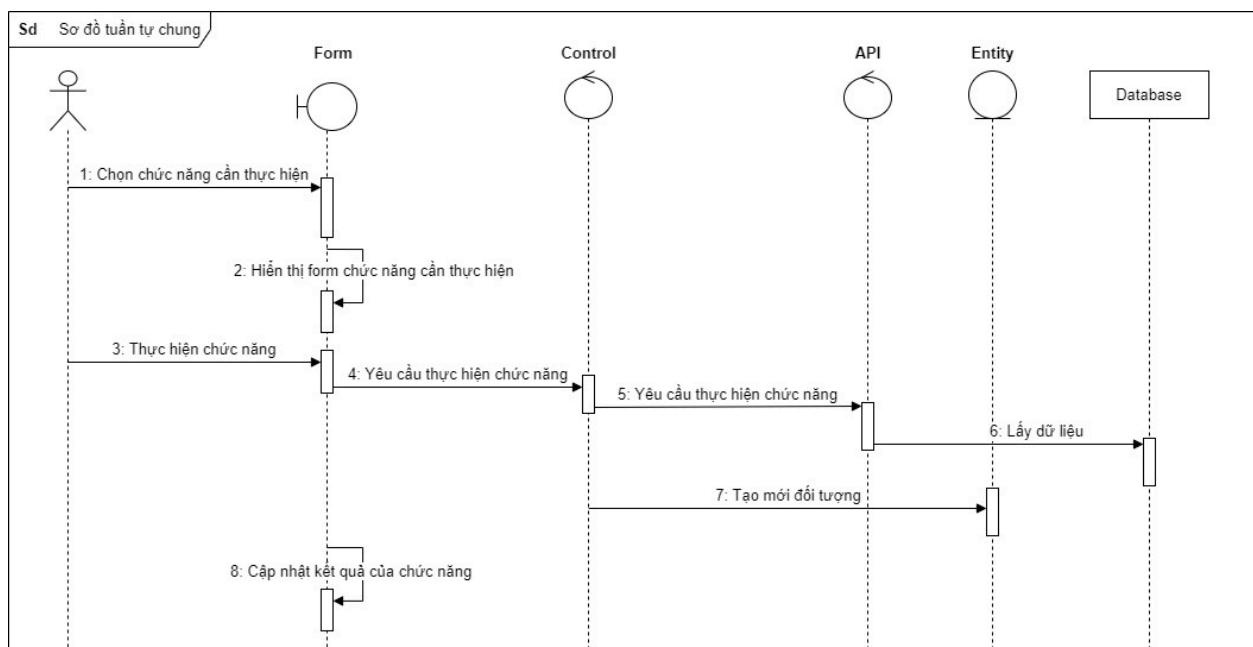
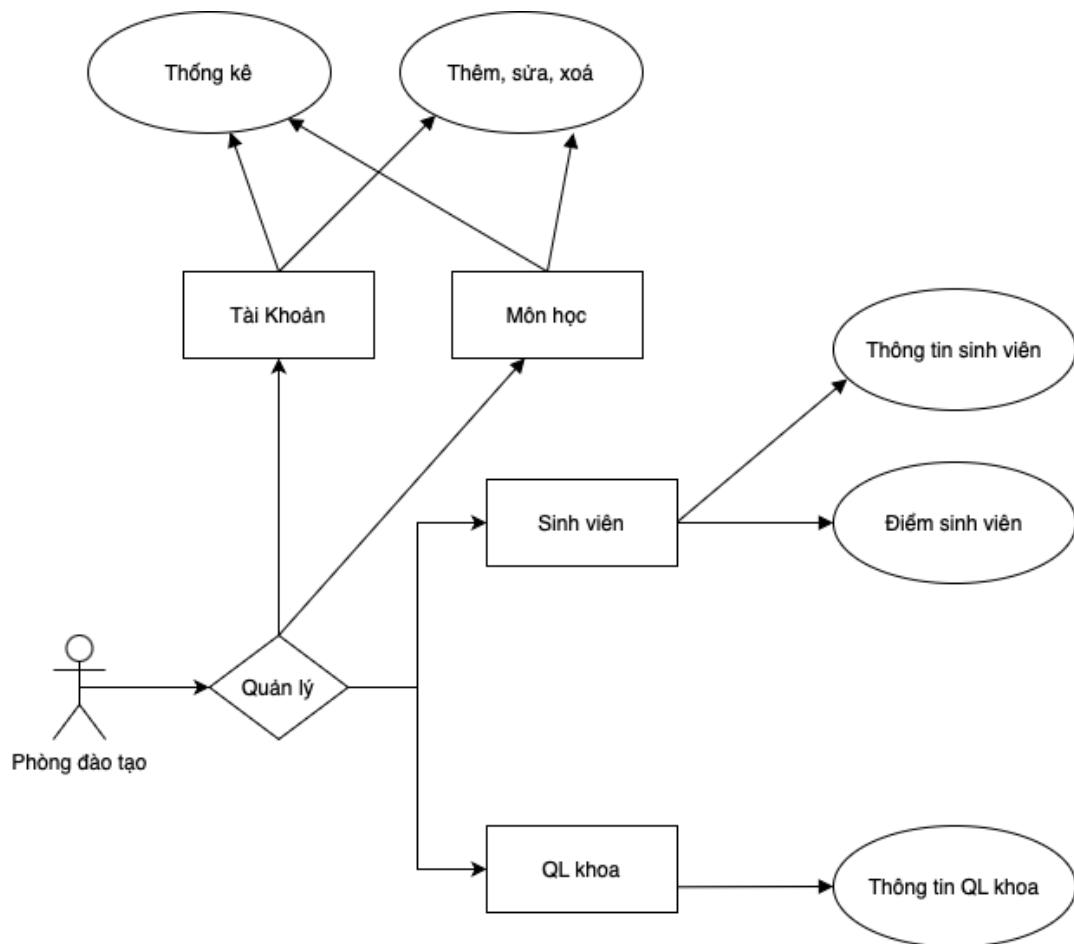
### 2.2.1. Các biểu đồ tuần tự (Sequence diagrams)



Hình 2.2 Sơ đồ lớp phân tích chung

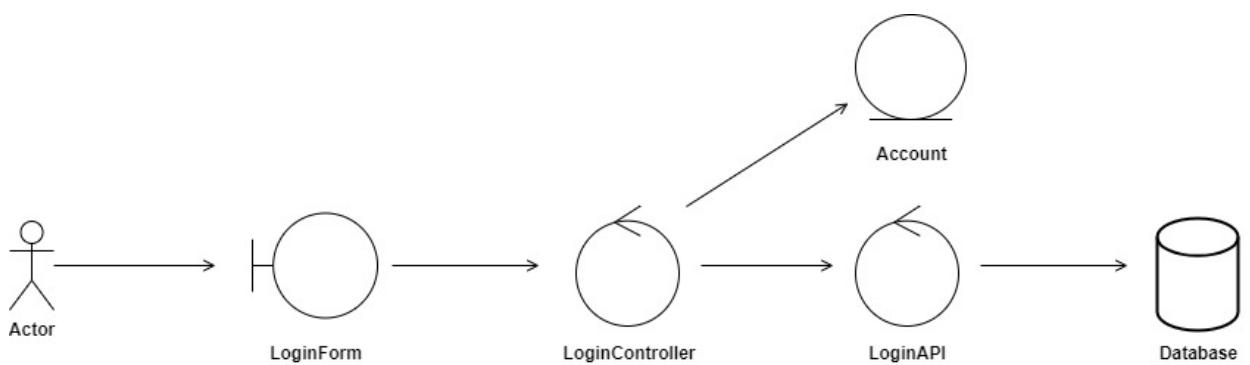
- **Actor:** Tác nhân tham gia sử dụng hệ thống.
- **Form:** Form là lớp trung gian thể hiện sự tương tác giữa hệ thống và những gì bên ngoài hệ thống. Ở đây là giao diện giữa người dùng và hệ thống.
- **Control:** Lớp điều khiển khi triển khai chức năng chính của UC.
- **Entity:** Lớp thực thể chứa những thông tin tồn tại và được lưu trữ lâu dài trong hệ thống.
- **API:** Lớp điều khiển thực hiện việc nhận/trả các API.
- **Database:** Cơ sở dữ liệu.

## 2.3. Sơ đồ tổng quản lý (PDT và quản lý khoa)

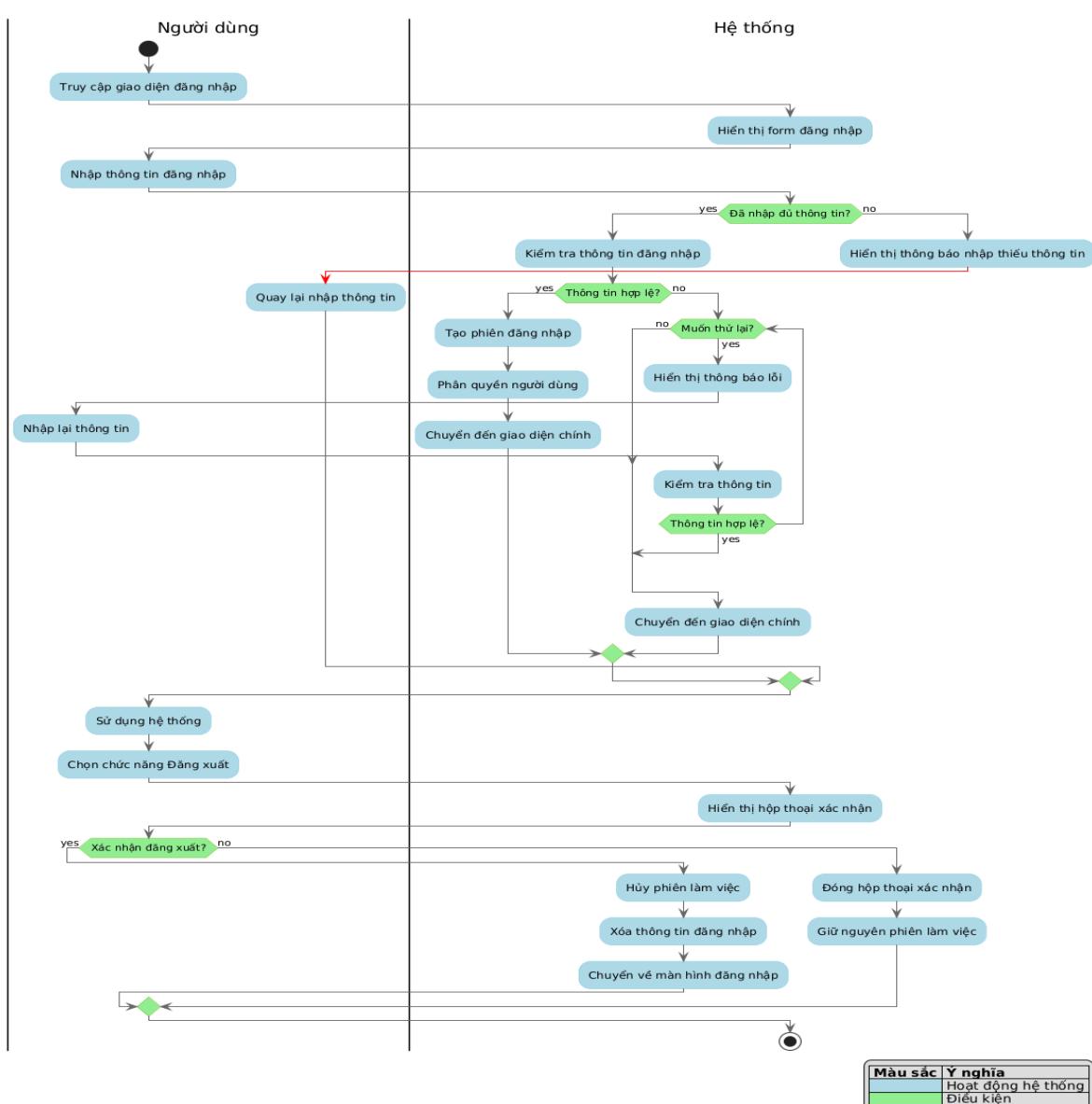


Hình 2.3 Sơ đồ tuần tự chung

❖ Biểu đồ tuần tự chức năng Đăng nhập



Hình 2.4 Sơ đồ lớp phân tích của chức năng Đăng nhập



Hình 2.5 Sơ đồ tuần tự của chức năng Đăng nhập

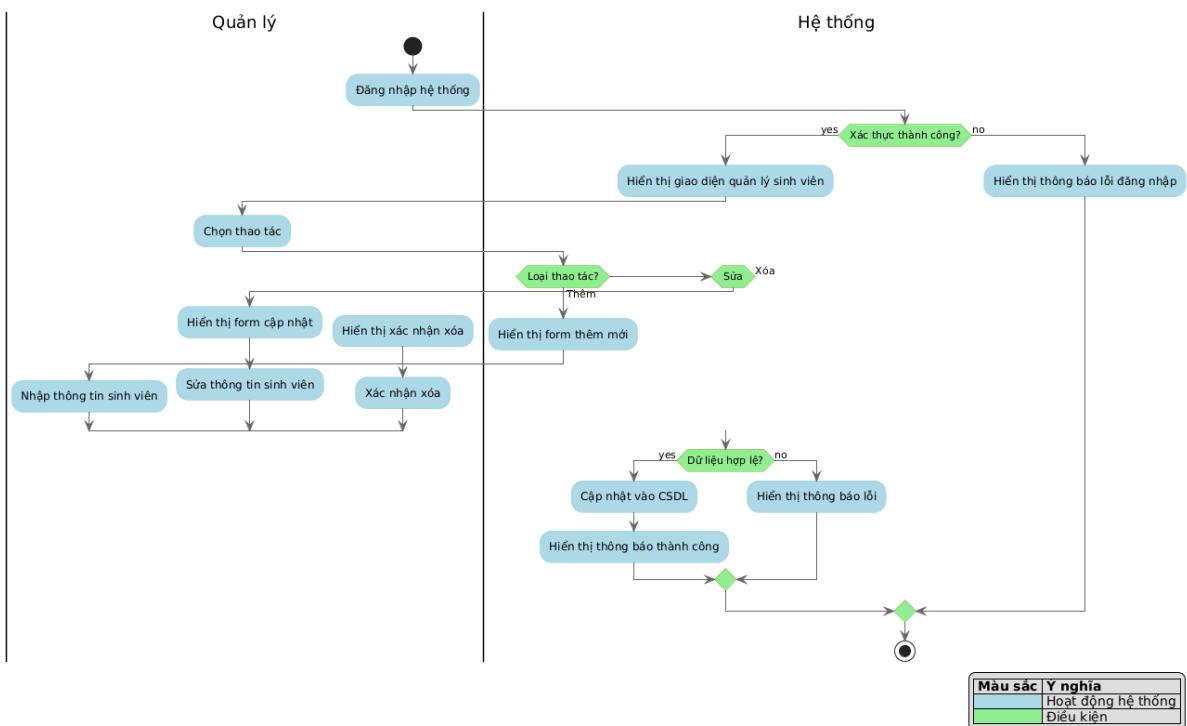
**Đăng nhập:**

Bước	Mô tả
1	Người dùng truy cập giao diện đăng nhập.
2	Hệ thống yêu cầu nhập thông tin (email, mật khẩu).
3	Người dùng nhập thông tin và bấm “Đăng nhập”.
4	<b>Hệ thống kiểm tra tính hợp lệ của thông tin:</b> <ul style="list-style-type: none"> <li>- Nếu hợp lệ, người dùng đăng nhập thành công và vào hệ thống.</li> <li>- Nếu không hợp lệ, hiển thị thông báo lỗi và yêu cầu nhập lại.</li> </ul>

### **Đăng xuất:**

Bước	Mô tả
1	Người dùng chọn “Đăng xuất” từ giao diện chính.
2	Hệ thống yêu cầu xác nhận đăng xuất.
3	<b>Người dùng xác nhận:</b> <ul style="list-style-type: none"> <li>• Nếu đồng ý (Yes), hệ thống đăng xuất và trở về màn hình đăng nhập.</li> <li>• Nếu không đồng ý (No), hủy thao tác đăng xuất và tiếp tục làm việc.</li> </ul>

❖ Biểu đồ tuần tự chức năng Quản lý sinh viên

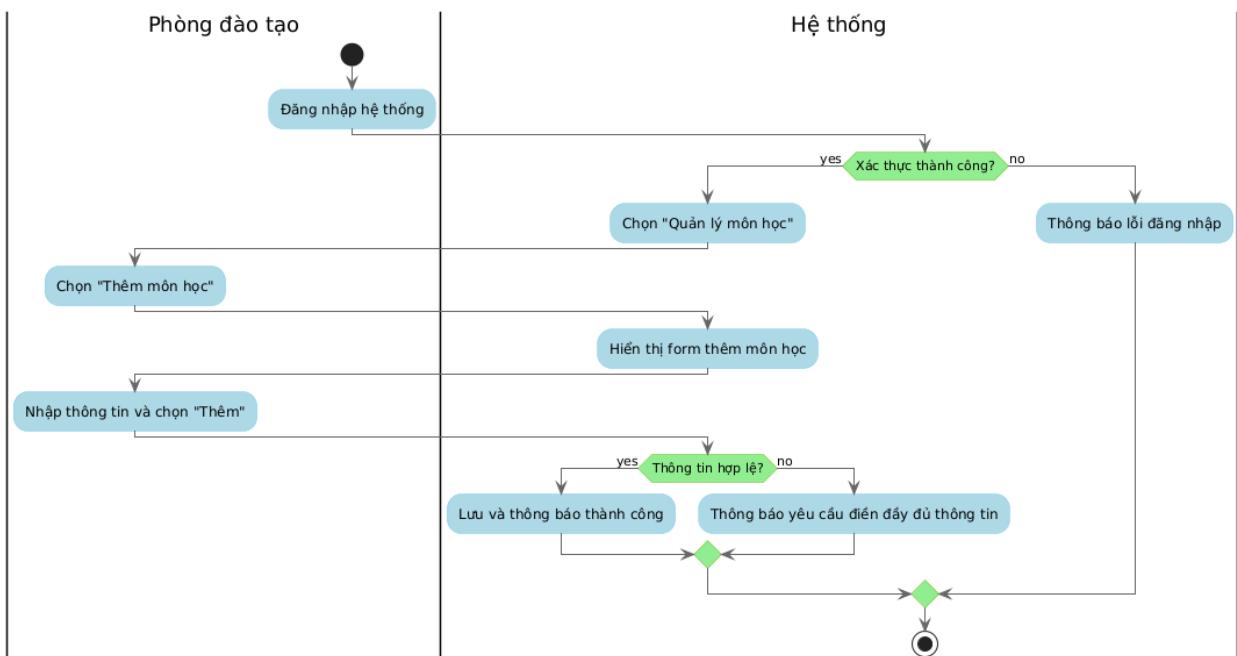


Hình 2.6 Biểu đồ tuần tự chức năng Quản lý sinh viên

### Giải thích

Bước	Mô tả
1	Với tài khoản có quyền quản lý(PĐT, QLK), hệ thống cho phép truy cập các chức năng quản lý. Đăng nhập được xác thực thành công qua email hoặc số điện thoại.
2	Người quản lý chọn chức năng quản lý sinh viên
3	Thông tin sinh viên bao gồm: <ul style="list-style-type: none"> <li>- Tên tài khoản</li> <li>- Tên Sinh viên</li> <li>- Mã Sinh viên</li> <li>- Email/SĐT</li> <li>- Ngày sinh</li> <li>- Khoa</li> <li>- khoa</li> <li>- Địa chỉ</li> </ul>

❖ Biểu đồ tuần tự chức năng Quản lý môn học

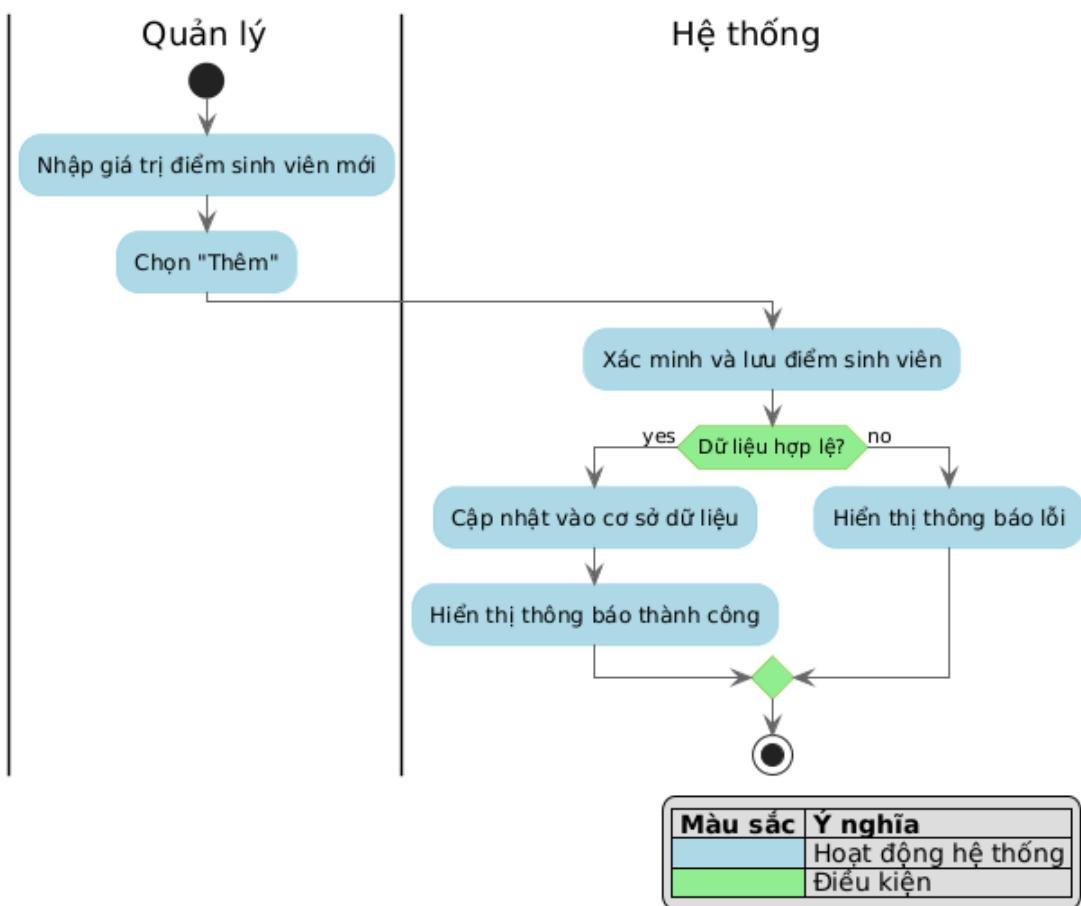


Hình 2.7 Biểu đồ tuần tự chức năng Quản lý môn học

### Giải thích

Bước	Mô tả
1	Phòng đào tạo bắt đầu bằng việc đăng nhập vào hệ thống để có quyền truy cập và thực hiện các thao tác quản lý môn học.
2	Sau khi đăng nhập thành công, phòng đào tạo chọn chức năng quản lý môn học để bắt đầu các thao tác thêm, sửa, xóa.
3	Phòng đào tạo nhập các thông tin cần thiết về môn học mới, như tên môn, mã môn, số tín chỉ, v.v. Sau đó, chọn nút "Thêm" để hệ thống tiếp nhận thông tin.
4	Hệ thống sẽ kiểm tra xem thông tin môn học có đầy đủ và hợp lệ không. <ul style="list-style-type: none"> <li>Nếu thông tin hợp lệ: Hệ thống sẽ lưu thông tin môn học và thông báo thành công.</li> <li>Nếu thông tin không hợp lệ: Hệ thống yêu cầu phòng đào tạo điền đầy đủ và chính xác các thông tin cần thiết.</li> </ul>

❖ Biểu đồ tuần tự chức năng Quản lý điểm sinh viên

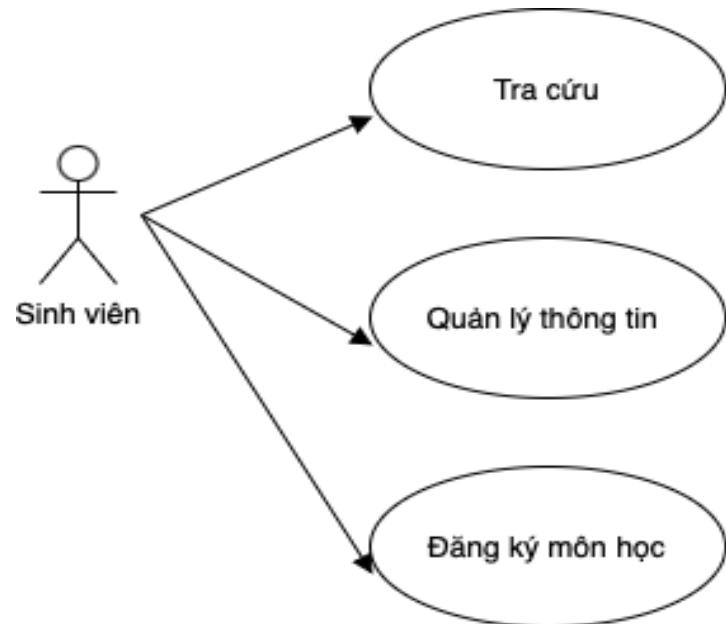


Hình 2.8 Biểu đồ tuần tự chức năng Quản lý điểm

### Giải thích

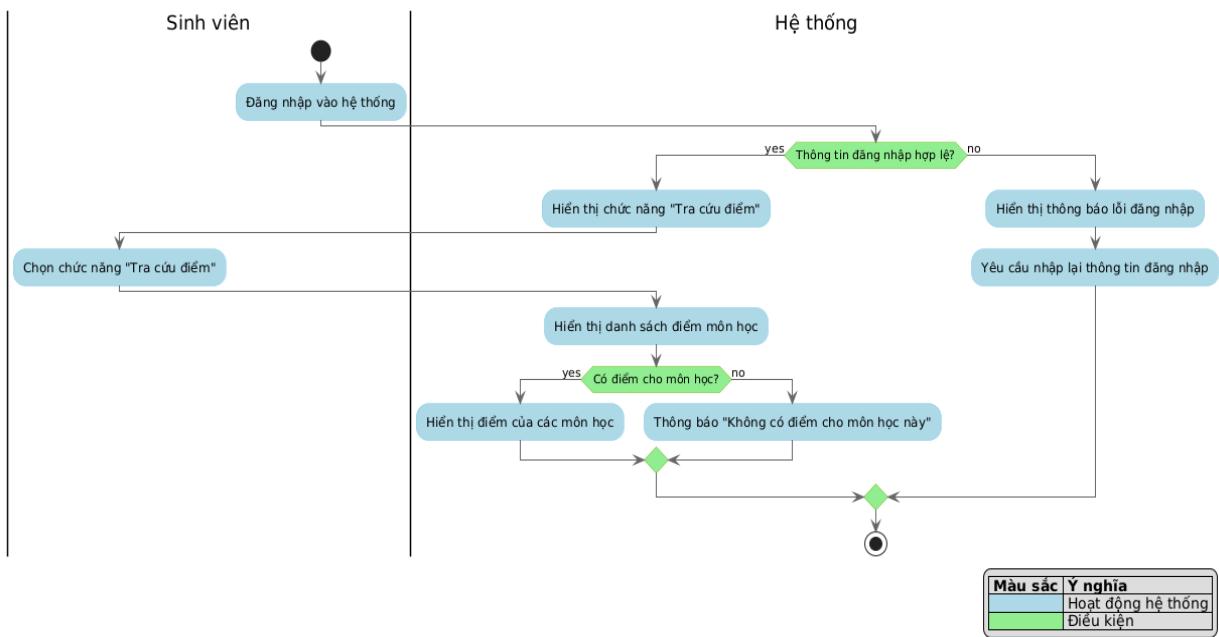
Bước	Mô tả
1	Với tài khoản có quyền quản lý (PDT, QLK), hệ thống cho phép truy cập các chức năng quản lý. Đăng nhập được xác thực thành công qua email hoặc số điện thoại.
2	Người quản lý chọn chức năng Quản lý điểm
3,4	Thông tin sản phẩm bao gồm: <ul style="list-style-type: none"> <li>- STT</li> <li>- Mã sinh viên</li> <li>- Tên sinh viên</li> <li>- Khoa</li> <li>- Điểm tích luỹ</li> <li>- Điểm hệ 10</li> <li>- Điểm hệ 4</li> </ul>

## 2.4. Sơ đồ chức năng sinh viên



### 2.4.1. Chức năng tra cứu điểm

Luồng hoạt động

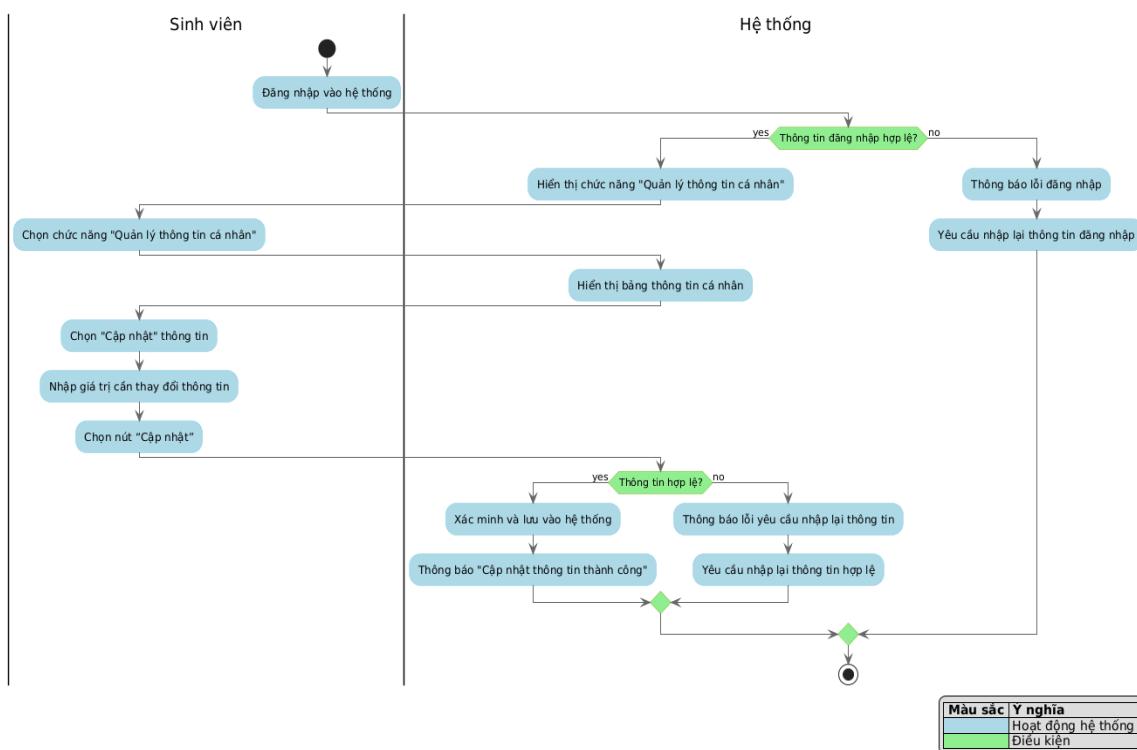


Hình 2.9 Biểu đồ tuần tự chức năng Tra cứu điểm

Bước	Mô tả
1	Sinh viên đăng nhập vào hệ thống
2	<b>Hệ thống kiểm tra thông tin đăng nhập:</b> <ul style="list-style-type: none"> <li>Nếu thông tin đăng nhập hợp lệ, sinh viên có thể truy cập vào chức năng "Tra cứu điểm".</li> <li>Nếu thông tin đăng nhập không hợp lệ, hệ thống yêu cầu nhập lại thông tin đăng nhập.</li> </ul>
3	<b>Sinh viên chọn chức năng "Tra cứu điểm".</b>
4	<b>Hệ thống hiển thị danh sách điểm:</b> <ul style="list-style-type: none"> <li>Nếu có điểm cho các môn học, hệ thống hiển thị điểm của các môn.</li> <li>Nếu không có điểm cho môn học nào, hệ thống thông báo "Không có điểm cho môn học này".</li> </ul>

### 2.4.2. Chức năng chỉnh sửa thông tin sinh viên

Luồng hoạt động:



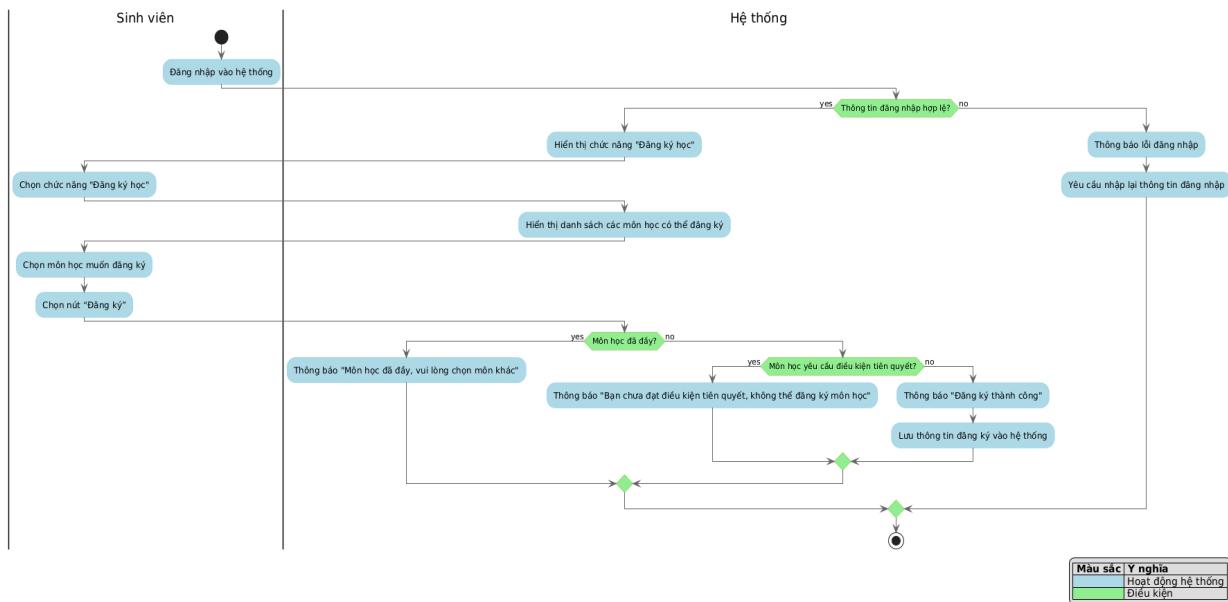
Hình 2.10 Biểu đồ tuần tự chức năng chỉnh sửa thông tin

Bước	Mô tả
1	Sinh viên đăng nhập vào hệ thống
2	<b>Hệ thống kiểm tra thông tin đăng nhập:</b> <ul style="list-style-type: none"> <li>Nếu thông tin đăng nhập hợp lệ, sinh viên có thể truy cập vào chức năng "Tra cứu điểm".</li> <li>Nếu thông tin đăng nhập không hợp lệ, hệ thống yêu cầu nhập lại thông tin đăng nhập.</li> </ul>
3	<b>Sinh viên chọn chức năng "Quản lý thông tin cá nhân" → Sinh viên chọn "Cập nhật" thông tin. → Sinh viên nhập giá trị cần thay đổi và bấm "Cập nhật".</b>
4	<b>Hệ thống xác minh thông tin:</b> <ul style="list-style-type: none"> <li>Nếu thông tin hợp lệ, hệ thống lưu và thông báo cập nhật thành công.</li> </ul>

	<ul style="list-style-type: none"> <li>Nếu thông tin không hợp lệ, hệ thống yêu cầu nhập lại thông tin chính xác.</li> </ul>
--	------------------------------------------------------------------------------------------------------------------------------

#### 2.4.3. Chức năng đăng ký học

Luồng hoạt động



Hình 2.11 Biểu đồ luồng tự chúc năng đăng ký học

Bước	Mô tả
1	Sinh viên đăng nhập vào hệ thống
2	<b>Hệ thống kiểm tra thông tin đăng nhập:</b> <ul style="list-style-type: none"> <li>Nếu thông tin đăng nhập hợp lệ, sinh viên có thể truy cập vào chức năng "Tra cứu điểm".</li> <li>Nếu thông tin đăng nhập không hợp lệ, hệ thống yêu cầu nhập lại thông tin đăng nhập.</li> </ul>
3	<b>Sinh viên chọn chức năng "Đăng ký học".</b> → <b>Hệ thống hiển thị danh sách các môn học có thể đăng ký.</b> → <b>Sinh viên chọn môn học muốn đăng ký.</b> → <b>Sinh viên chọn nút "Đăng ký".</b>
4	<b>Hệ thống kiểm tra điều kiện:</b>

	<ul style="list-style-type: none"> <li>• Nếu môn học đã đầy, hệ thống thông báo "Môn học đã đầy, vui lòng chọn môn khác".</li> <li>• Nếu môn học yêu cầu điều kiện tiên quyết mà sinh viên chưa đạt, hệ thống thông báo lỗi và không cho phép đăng ký.</li> <li>• Nếu không có vấn đề gì, hệ thống thông báo "Đăng ký thành công" và lưu thông tin vào hệ thống.</li> </ul>
--	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

### 2.2.2. Góc nhìn của các lớp trong hệ thống (Views of participating classes)

STT	Chức năng	Tên lớp	Loại lớp	Vai trò
1	Quản lý sinh viên	StudentForm	Boundary Class	Cung cấp giao diện cho người dùng để quản lý thông tin nhân viên (thêm, sửa, xóa). Tương tác với EmployeeController để xử lý các yêu cầu về quản lý nhân viên.
		StudentController	Control Class	Xử lý các yêu cầu từ StudentForm, tương tác với StudentAPI để lấy và cập nhật thông tin nhân viên.
		StudentAPI	Control Class	Giao tiếp với lớp thực thể Student để truy xuất và cập nhật dữ liệu nhân viên từ cơ sở dữ liệu.
		Student	Entity Class	Lưu trữ thông tin chi tiết về nhân viên, bao gồm thông tin cá nhân và trạng thái công việc.
		GradeForm	Boundary Class	Cung cấp giao diện cho người dùng để quản lý

				thông tin sản phẩm (thêm, sửa, xóa). Tương tác với GradeController để gửi yêu cầu thao tác dữ liệu sản phẩm.
2	Quản lý điểm sinh viên	GradeController	Control Class	Xử lý yêu cầu từ GradeForm, tương tác với GradeAPI để lấy và cập nhật điểm số sinh viên.
		GradeAPI	Control Class	Tương tác với lớp thực thể Product để truy cập và thao tác với dữ liệu sản phẩm trong cơ sở dữ liệu.
		Grade	Entity Class	Lưu trữ điểm số của sinh viên theo môn học, kỳ học.
3	Tra cứu điểm	ScoreLookupForm	Boundary Class	Cung cấp giao diện cho sinh viên tra cứu điểm. Tương tác với ScoreLookupController
		ScoreLookupController	Control Class	Xử lý yêu cầu từ ScoreLookupForm, tương tác với ScoreAPI để lấy và cập nhật thông tin nhà cung cấp.
		ScoreAPI	Control Class	Giao tiếp với lớp thực thể Grade để truy xuất và cập nhật dữ liệu nhà cung cấp từ cơ sở dữ liệu.
4		CourseRegistrationForm	Boundary Class	Cung cấp giao diện cho người dùng để quản lý thông tin khách hàng (thêm, sửa, xóa). Tương tác với CourseRegistrationController

	Đăng ký môn học			ller để xử lý các yêu cầu về khách hàng.
	CourseRegistrationController	Control Class	Nhận và xử lý các yêu cầu từ CourseRegistrationForm, tương tác với CourseAPI để lấy và cập nhật thông tin khách hàng.	
	CourseAPI	Control Class	Giao tiếp với lớp thực thể Course để truy xuất và cập nhật dữ liệu môn học từ cơ sở dữ liệu.	
	Course	Entity Class	Lưu trữ thông tin môn học, lịch học, số tín chỉ.	
5	Quản lý Khoa	DepartmentForm	Boundary Class	Cung cấp giao diện cho người dùng để quản lý thuộc tính sản phẩm (thêm, sửa, xóa). Tương tác với DepartmentController để gửi yêu cầu về thuộc tính sản phẩm.
		DepartmentController	Control Class	Xử lý yêu cầu từ DepartmentForm, tương tác với DepartmentAPI.
		DepartmentAPI	Control Class	Giao tiếp với Department để lấy và cập nhật thông tin khoa.
		Department	Entity Class	Lưu trữ thông tin về các thuộc tính của khoa.
		AccountForm	Boundary Class	Giao diện đăng nhập, đổi mật khẩu, quản lý tài khoản người dùng.
		AccountController	Control Class	Xử lý yêu cầu từ AccountForm, kiểm tra xác thực, đổi mật khẩu.

6	Quản lý tài khoản	AccountAPI	Control Class	Tương tác với Account để xác thực đăng nhập, quản lý quyền.
		Account	Entity Class	Lưu trữ thông tin tài khoản, quyền hạn người dùng.
7	Quản lý nhóm quyền	RoleGroupForm	Boundary Class	Cung cấp giao diện cho người dùng quản lý nhóm quyền. Tương tác với RoleGroupController để yêu cầu thêm, sửa, hoặc xóa quyền.
		RoleGroupController	Control Class	Xử lý các thao tác thêm, sửa, xóa từ RoleGroupForm và tương tác với RoleGroupAPI để thực hiện quản lý nhóm quyền.
		RoleGroupAPI	Control Class	Giao tiếp với RoleGroup để lưu trữ và truy xuất dữ liệu quyền từ cơ sở dữ liệu.
		RoleGroup	Entity Class	Lưu trữ và quản lý thông tin về quyền và nhóm quyền của người dùng.
8	Đăng nhập	LoginForm	Boundary Class	Cung cấp giao diện cho người dùng nhập thông tin đăng nhập. Tương tác với LoginController để xử lý yêu cầu đăng nhập.
		LoginController	Control Class	Xử lý yêu cầu đăng nhập từ LoginForm và tương tác với LoginAPI để xác thực thông tin người dùng.
		LoginAPI	Control Class	Giao tiếp với Account để xác thực và lấy thông tin

				người dùng từ cơ sở dữ liệu.
		Account	Entity Class	Quản lý và lưu trữ thông tin tài khoản người dùng.
9	Đăng xuất	LogoutForm	Boundary Class	Cung cấp giao diện cho người dùng thực hiện đăng xuất. Tương tác với LogoutController để xử lý yêu cầu đăng xuất.
		LogoutController	Control Class	Xử lý yêu cầu đăng xuất từ LogoutForm và tương tác với LogoutAPI để thực hiện quá trình đăng xuất.
		LogoutAPI	Control Class	Giao tiếp với Account để quản lý trạng thái đăng nhập và đăng xuất của tài khoản người dùng.
		Account	Entity Class	Quản lý và lưu trữ thông tin trạng thái đăng nhập của người dùng.

Bảng 2.2 Góc nhìn của các lớp trong hệ thống

### 3. Thiết kế (Use-case design)

#### 3.1. Xác định các thành phần thiết kế (Identify design elements)

##### 3.1.1. Xác định các lớp (Identify classes)

STT	Tên lớp	Phương thức
1	StudentForm	+ selectManageStudents() + displayStudentUI(studentList) + displayAddForm() + updateAndDisplayStudentList() + displayStudentInfo(studentData)...
2	GradeForm	+ displayGradeUI(studentGrades) + displayAddGradeForm(studentID) + displayEditGradeForm(gradeData)
3	ScoreLookupForm	+ displayScoreLookupUI() + requestStudentGrades(studentID)
4	DepartmentForm	+ displayDepartmentUI(departmentList) + displayAddDepartmentForm()
5	AccountForm	+ displayLoginForm() + displayChangePasswordForm()
6	StudentController	+ requestStudentList() + createStudent(studentData) + requestDetails(studentID) + updateInfo(studentData) + requestDeletion(studentID) + requestSearch(keyword)
7	ScoreLookupController	+ requestStudentGrades(studentID)
8	GradeController	+ requestGradeList(studentID) + addGrade(gradeData)

STT	Tên lớp	Phương thức
		+ updateGrade(gradeData)
9	CourseRegistrationController	+ requestAvailableCourses() + registerCourse(studentID, courseID)
10	DepartmentController	+ requestDepartmentList() + addDepartment(departmentData)
11	AccountController	+ validateLogin(username, password) + updatePassword(userID, newPassword)
12	Student	+ createStudentObject(studentData)  + retrieveStudentData(studentID)  + deleteStudent(studentID)
13	Grade	+ createGradeObject(gradeData) + retrieveGrades(studentID) + updateGrade(gradeData)
14	Course	+ createCourseObject(courseData) + retrieveCourseData(courseID)
15	Department	+ createDepartmentObject(departmentData) + retrieveDepartmentData(departmentID)
16	Account	+ createAccountObject() + createAccountObject(accountData) + getAccountInfo(accountID) + updateAccountInfo(accountID) + deleteAccountRecord(accountID)
17	RoleGroup	+ createRoleGroupList(RoleGroupList) + createRoleGroupObject(groupData) + retrieveData(groupID) + deleteRoleGroup(groupData)

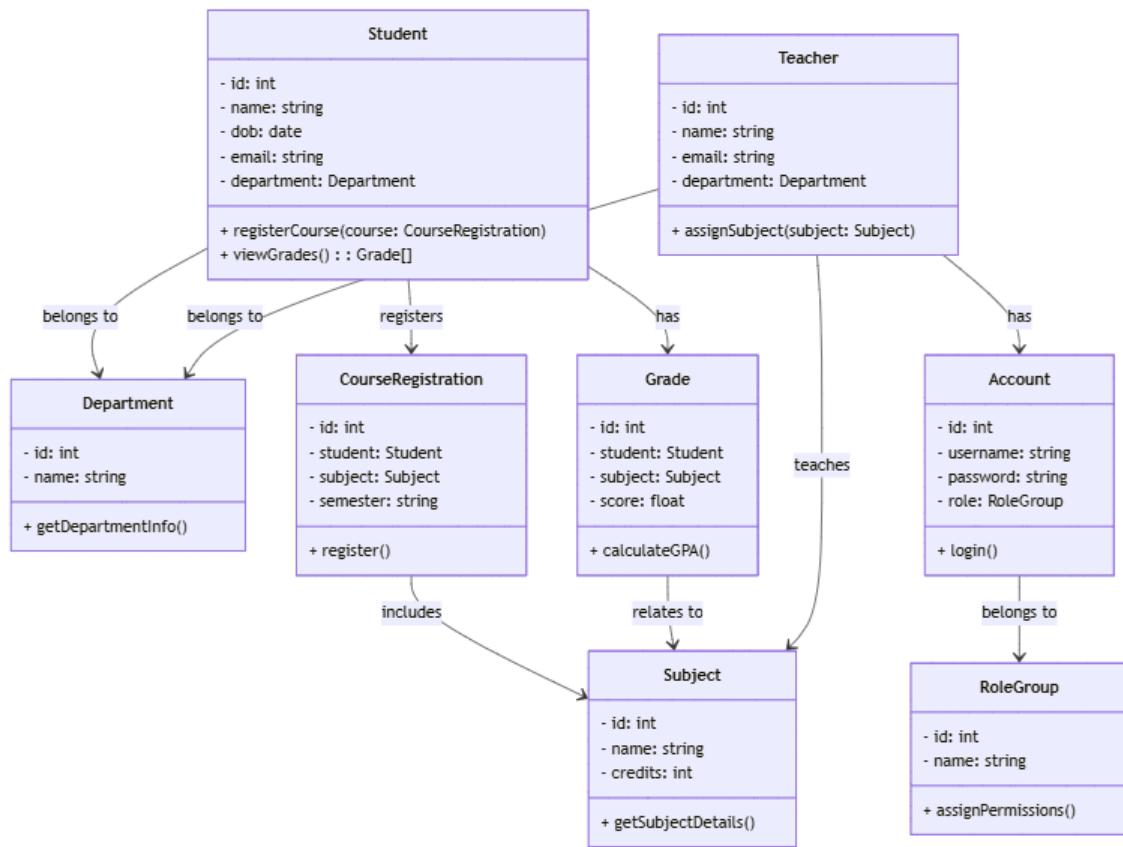
Bảng 3.1 Xác định các lớp

### 3.1.2. Xác định các gói (Identify packages)

ST T	Tên gói	Thành phần
HCI layer		
1	UI	<ul style="list-style-type: none"> <li>- StudentForm</li> <li>- SubjectForm</li> <li>- GradeForm</li> <li>- DepartmentForm</li> <li>- CourseRegistrationForm</li> </ul>
PD layer		
2	StudentManagement	<ul style="list-style-type: none"> <li>- StudentController</li> <li>- Student</li> </ul>
3	SubjectManagement	<ul style="list-style-type: none"> <li>- SubjectController</li> <li>- Subject</li> </ul>
4	GradeManagement	<ul style="list-style-type: none"> <li>- GradeController</li> <li>- Grade</li> </ul>
5	CourseRegistrationManagement	<ul style="list-style-type: none"> <li>- CourseRegistrationManagementController</li> <li>- CourseRegistrationManagement</li> </ul>
DM layer		
6	StudentDataManagement	<ul style="list-style-type: none"> <li>- StudentAPI</li> <li>- Các bảng dữ liệu thuộc gói SinhVien trong CSDL</li> </ul>
7	SubjectDataManagement	<ul style="list-style-type: none"> <li>- SubjectAPI</li> <li>- Các bảng dữ liệu thuộc gói MonHoc trong CSDL</li> </ul>
8	DepartmentDataManagement	<ul style="list-style-type: none"> <li>- DepartmentAPI</li> <li>- Các bảng dữ liệu thuộc gói Khoa trong CSDL</li> </ul>

Bảng 3.2 Xác định các gói

### 3.1.2. Thiết kế biểu đồ lớp (Class diagrams)



Hình 3.1 Biểu đồ lớp

Sơ đồ lớp này mô tả các thành phần chính trong hệ thống quản lý sinh viên, bao gồm quản lý sinh viên, môn học, điểm số, tài khoản, giảng viên và các dịch vụ liên quan.

- **Student (Sinh viên)** có thể đăng ký nhiều môn học (**Subject**) và có bảng điểm (**Grade**) để lưu trữ kết quả học tập.
- **Subject (Môn học)** có thể có nhiều giảng viên (**Teacher**) phụ trách giảng dạy, và mỗi môn học đều liên kết với điểm số để theo dõi quá trình học tập của sinh viên.
- **Department (Khoa)** đóng vai trò tổ chức các môn học và giảng viên trong hệ thống, giúp phân loại và quản lý các nhóm môn học chuyên ngành.

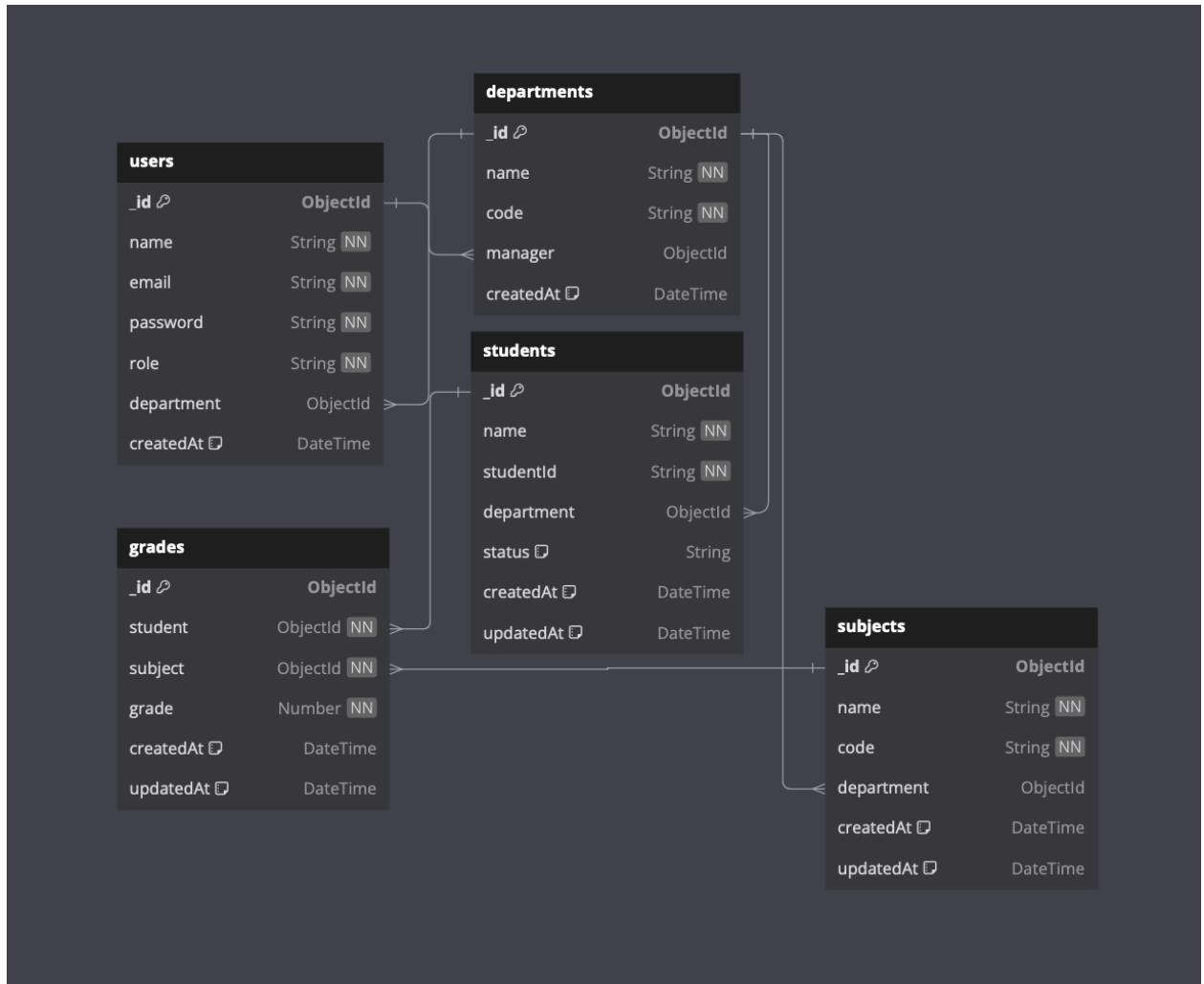
Bên cạnh đó, **CourseRegistration** là bảng đăng ký học phần của sinh viên, dùng để quản lý thông tin đăng ký các môn học theo từng kỳ. Điều này giúp theo dõi tiến trình học tập và lịch trình môn học của sinh viên trong hệ thống.

**Phần quản lý tài khoản trong hệ thống** được thể hiện qua các lớp **Account** và **Teacher**, trong đó mỗi giảng viên có một tài khoản (**Account**) để truy cập vào hệ

thông. Tài khoản này được gán cho một **RoleGroup** (Nhóm vai trò), giúp phân quyền và xác định các chức năng mà người dùng có thể thực hiện trong hệ thống.

## 3.2. Thiết kế cơ sở dữ liệu (Database design)

### 3.2.1. Lược đồ cơ sở dữ liệu



Hình 3.1 Lược đồ cơ sở dữ liệu

### 1. Bảng users (Người dùng)

Quản lý thông tin đăng nhập của người dùng trong hệ thống.

- **Các cột quan trọng:**

- \_id: Khóa chính (ObjectId).
- name, email, password: Thông tin cá nhân của người dùng.
- role: Vai trò của người dùng (quản trị viên, giảng viên, v.v.).

- department: Liên kết với phòng ban của người dùng (ObjectId tham chiếu bảng departments).
- createdAt: Ngày tạo tài khoản.

## 2. Bảng departments (Phòng ban)

Lưu thông tin về các phòng ban trong hệ thống.

- **Các cột quan trọng:**

- \_id: Khóa chính (ObjectId).
- name, code: Tên và mã phòng ban.
- manager: Người quản lý phòng ban (ObjectId tham chiếu bảng users).
- createdAt: Ngày tạo phòng ban.

## 3. Bảng students (Sinh viên)

Quản lý thông tin sinh viên trong hệ thống.

- **Các cột quan trọng:**

- \_id: Khóa chính (ObjectId).
- name: Tên sinh viên.
- studentId: Mã số sinh viên.
- department: Phòng ban mà sinh viên thuộc về (ObjectId tham chiếu bảng departments).
- status: Trạng thái học tập của sinh viên.
- createdAt, updatedAt: Ngày tạo và cập nhật thông tin.

## 4. Bảng subjects (Môn học)

Quản lý danh sách các môn học trong hệ thống.

- **Các cột quan trọng:**

- \_id: Khóa chính (ObjectId).
- name, code: Tên và mã môn học.
- department: Môn học thuộc phòng ban nào (ObjectId tham chiếu bảng departments).
- createdAt, updatedAt: Ngày tạo và cập nhật thông tin.

## 5. Bảng grades (Điểm số)

Lưu thông tin điểm số của sinh viên theo từng môn học.

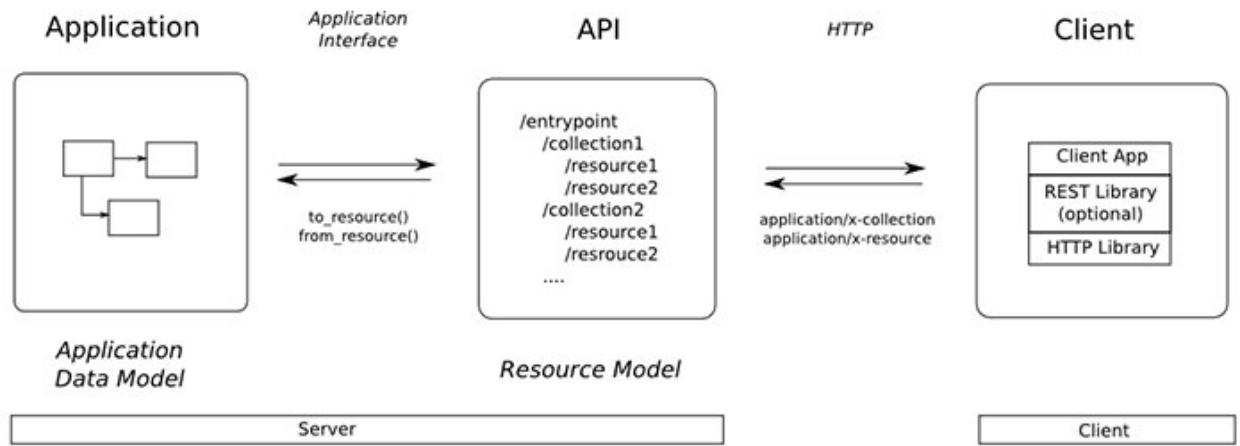
- **Các cột quan trọng:**

- \_id: Khóa chính (ObjectId).
- student: Sinh viên được chấm điểm (ObjectId tham chiếu bảng students).
- subject: Môn học được chấm điểm (ObjectId tham chiếu bảng subjects).
- grade: Điểm số của sinh viên.
- createdAt, updatedAt: Ngày tạo và cập nhật thông tin.

Sơ đồ này thể hiện mối quan hệ giữa các thực thể trong hệ thống quản lý sinh viên, giúp dễ dàng truy xuất và quản lý thông tin.

### 3.4. Thiết kế API

#### 3.4.1. Cấu trúc thư mục của API



Hình 3.2 Cấu trúc thư mục của API

REST hoạt động chủ yếu dựa vào giao thức HTTP. Các hoạt động cơ bản nêu trên sẽ sử dụng những phương thức HTTP riêng.

- **GET (SELECT):** Trả về một Resource hoặc một danh sách Resource.
- **POST (CREATE):** Tạo mới một Resource.
- **PUT (UPDATE):** Cập nhật thông tin cho Resource.
- **DELETE (DELETE):** Xoá một Resource.

Những phương thức hay hoạt động này thường được gọi là CRUD tương ứng với Create, Read, Update, Delete – Tạo, Đọc, Sửa, Xóa. [\[3\]](#)

#### 3.4.2. Danh sách các API

STT	Chức năng	URL	Phương thức
1	Đăng nhập	/api/auth/login	POST

2	Đăng ký	/api/auth/register	POST
3	Đổi mật khẩu	/api/auth/change-password	PUT
4	Xác thực người dùng	/api/auth	POST,GET
5	Quản lý người dùng	/api/users	GET,POST,PUT,DELETE
6	Quản lý khoa/bộ môn	/api/departments	GET,POST,PUT,DELETE
7	Quản lý sinh viên	/api/students	GET,POST,PUT,DELETE
8	Quản lý môn học	/api/subjects	GET,POST,PUT,DELETE
9	Quản lý điểm số	/api/grades	GET,POST,PUT,DELETE
10	Đăng ký môn học	/api/courseRegistrations	GET,POST,PUT,DELETE

Bảng 3.3 Danh sách các API

#### 3.4.3. Thiết kế chi tiết API

STT	URL	Mô tả chi tiết
1	/api/auth/login	<p>Phương thức: POST</p> <p>Mục đích: Xác thực thông tin đăng nhập người dùng.</p> <p>Tham số đầu vào: {     mail: string,     password: string }</p> <p>Kết quả trả về:</p> <ul style="list-style-type: none"> <li>- Thành công: {         userid: int,         result: boolean,         success: true     }</li> <li>- Thất bại: {         result: false,         success: true     }</li> </ul>
2	/api/auth/register	<p>Phương thức: POST</p> <p>Mục đích: Tạo tài khoản quản lý cho người dùng.</p> <p>Tham số đầu vào: {     mail: string,     password: string }</p>

STT	URL	Mô tả chi tiết
		<p>}</p> <p>Kết quả trả về:</p> <ul style="list-style-type: none"> <li>- Thành công: {</li> <li>    message: "Account added",</li> <li>    success: true,</li> <li>    Data: [ {</li> <li>        hoten: string,</li> <li>        sodienthoai: string,</li> <li>        mail: string,</li> <li>        tenhienthi: string</li> <li>    } ] }</li> <li>- Thất bại: {</li> <li>    message: "Registration failed",</li> <li>    success: false }</li> </ul>
3	/api/auth/change-password	<p>Phương thức: PUT</p> <p>Mục đích: Đổi mật khẩu tài khoản người dùng.</p> <p>Tham số đầu vào: {</p> <p>    mataikhoan: int,</p> <p>    matkhaucu: string,</p> <p>    maukhaumoi: string</p> <p>}</p> <p>Kết quả trả về:</p> <ul style="list-style-type: none"> <li>- Thành công: {</li> <li>    message: "Password updated",</li> <li>    success: true</li> <li>}</li> <li>- Thất bại: {</li> <li>    message: "Update failed",</li> <li>    success: false</li> <li>}</li> </ul>
4	/api/users	<p>Phương thức: GET</p> <p>Mục đích: Lấy danh sách nhóm quyền.</p> <p>Tham số đầu vào: không có.</p> <p>Kết quả trả về:</p>

STT	URL	Mô tả chi tiết
		<ul style="list-style-type: none"> <li>- Thành công: {           <pre>"users": [             {               "id": "123",               "name": "Nguyễn Văn A",               "email": "userA@example.com",               "role": "admin",               "department": "IT"             },             {               "id": "124",               "name": "Trần Thị B",               "email": "userB@example.com",               "role": "teacher",               "department": "Math"             }           ],           "success": true         }</pre> </li> <li>- Thất bại: {           <pre>roles: [],           success: false         }</pre> </li> </ul>
		<p>Phương thức: POST</p> <p>Mục đích: Thêm nhóm quyền vào cơ sở dữ liệu.</p> <p>Tham số đầu vào: {           <pre>"name": "Nguyễn Văn C",           "email": "userC@example.com",           "password": "123456",           "role": "student",           "department": "Physics"         }</pre> </p> <p>Kết quả trả về:</p> <ul style="list-style-type: none"> <li>- Thành công: {</li> </ul>

STT	URL	Mô tả chi tiết
		<p>"message": "User added successfully",      "success": true,      "body": {        "id": "125",        "name": "Nguyễn Văn C",        "email": "userC@example.com",        "role": "student",        "department": "Physics"      }      }      - Thất bại: {        "message": "User creation failed",        "success": false,        "body": {}      }</p> <p>Phương thức PUT:      Mục đích: Cập nhật thông tin môn học      Tham số đầu vào: {        "id": "2001",        "name": "Lập trình Python nâng cao",        "code": "PY201",        "department": "CNTT"      }      Kết quả trả về: {        "message": "Subject updated",        "success": true      }</p> <p>Phương thức DELETE:      Tham số đầu vào: {        "id": "2001"      }      Kết quả trả về: {        "message": "Subject deleted",      }</p>

STT	URL	Mô tả chi tiết
		<pre>"success": true }</pre>
5	/api/auth	<p>Phương thức: POST Mục đích: Đăng nhập người dùng. Tham số đầu vào: {          "email": "user@example.com",          "password": "123456"      } Kết quả trả về:          - Thành công: {              "message": "Login successful",              "success": true,              "token": "eyJhbGciOiJIUzI1..."          }          - Thất bại: {              "message": "Invalid email or              password",              "success": false          }</p> <p>Phương thức: GET Mục đích: Lấy thông tin người dùng đã đăng nhập Kết quả trả về: Thành công: {          "user": {              "id": "123",              "name": "Nguyễn Văn A",              "email": "user@example.com",              "role": "admin"          },          "success": true      } Thất bại: {          "message": "Unauthorized",          "success": false     }</p>

STT	URL	Mô tả chi tiết
		<p>}</p>
6	/api/departments	<p>Phương thức: GET Mục đích: Lấy danh sách quyền hạn của một nhóm quyền. Tham số đầu vào: mã nhóm quyền trên url. Kết quả trả về:</p> <pre>{     "departments": [         {             "id": "1",             "name": "Công nghệ thông tin",             "code": "CNTT",             "manager": "Nguyễn Văn A"         }     ],     "success": true }</pre> <p>- Thất bại: {success: false}</p> <p>Phương thức: POST Mục đích: Thêm một khoa/môn học Tham số đầu vào:</p> <pre>{     "name": "Toán ứng dụng",     "code": "MATH",     "manager": "Trần Thị B" }</pre> <p>Kết quả trả về:</p> <pre>{     "message": "Department added",     "success": true }</pre>

STT	URL	Mô tả chi tiết
		<p>Phương thức: POST  Mục đích: Thêm khoa/bộ môn mới  Tham số đầu vào: {  "code": "MATH",  "name": "Khoa Toán",  "manager": "Trần Văn B"  }  Kết quả trả về: {  "message": "Department added",  "success": true  }</p> <p>Phương thức PUT:  Mục đích: Cập nhật thông tin khoa/bộ  môn  Tham số đầu vào: {  "code": "MATH",  "name": "Khoa Toán",  "manager": "Trần Văn B"  }  Kết quả trả về: {  "message": "Department updated",  "success": true  }</p> <p>Phương thức DELETE:  Tham số đầu vào: {  "code": "MATH",  }  Kết quả: {  "message": "Department deleted",  "success": true  }</p>
7	/api/students	<p>Phương thức: GET  Mục đích: Lấy danh sách sinh viên</p>

STT	URL	Mô tả chi tiết
		<p>Tham số đầu vào: Không có</p> <p>Kết quả trả về:</p> <pre>{     "students": [         {             "id": "123",             "name": "Nguyễn Văn A",             "studentId": "SV001",             "department": "CNTT",             "status": "Đang học"         }     ],     "success": true }</pre> <p>Phương thức: POST</p> <p>Mục đích: Thêm sinh viên mới</p> <p>Tham số đầu vào: {</p> <pre>"name": "Lê Thị B", "studentId": "SV002", "department": "MATH", "status": "Đang học"</pre> <p>}</p> <p>Kết quả trả về: {</p> <pre>"message": "Student added", "success": true</pre> <p>}</p> <p>Phương thức: PUT</p> <p>Mục đích: Cập nhật thông tin sinh viên</p> <p>Tham số đầu vào: {</p> <pre>"id": "123", "status": "Bảo lưu"</pre> <p>}</p> <p>Kết quả trả về: {</p> <pre>"message": "Student updated",</pre>

STT	URL	Mô tả chi tiết
		<pre>     "success": true }  Phương thức: DELETE Mục đích: Xóa sinh viên Phương thức đầu vào: {     "id": "123" } Kết quả trả về: {     "message": "Student deleted",     "success": true } </pre>
8	/api/subjects	<pre> {     "subjects": [         {             "id": "101",             "name": "Toán rời rạc",             "code": "MATH101",             "department": "MATH"         }     ],     "success": true } </pre>
		<p>Phương thức: POST  Mục đích: Thêm môn học mới.  Tham số đầu vào: {      "name": "Lập trình Python",      "code": "CS102",      "department": "CNTT"  }  Kết quả trả về: {      "message": "Subject added",      "success": true }</p>
		Phương thức: PUT

STT	URL	Mô tả chi tiết
		<p>Mục đích: Cập nhật thông tin môn học.</p> <p>Tham số đầu vào: {</p> <pre>"id": "101", "name": "Toán cao cấp"</pre> <p>}</p> <p>Kết quả trả về:</p> <pre>{   "message": "Subject updated",   "success": true }</pre>
		<p>Phương thức: DELETE</p> <p>Mục đích: Xóa môn học.</p> <p>Tham số đầu vào: {</p> <pre>"id": "101"</pre> <p>}</p> <p>Kết quả trả về:</p> <pre>{   "message": "Subject deleted",   "success": true }</pre>
9	/api/grades	<p>Phương thức: GET</p> <p>Mục đích: Lấy danh sách điểm số của học sinh.</p> <p>Tham số đầu vào: không có.</p> <p>Kết quả trả về:</p> <pre>{   "grades": [     {       "student": "SV001",       "subject": "MATH101",       "grade": 8.5     }   ],   "success": true }</pre>

STT	URL	Mô tả chi tiết
		<p>Phương thức: POST</p> <p>Mục đích: Nhập điểm số cho sinh viên.</p> <p>Tham số đầu vào: {</p> <pre>"student": "SV001", "subject": "MATH101", "grade": 9.0</pre> <p>}</p> <p>Kết quả trả về:</p> <ul style="list-style-type: none"> <li>- Thành công: {</li> <li>    "message": "Grade added",</li> <li>    "success": true</li> </ul> <p>}</p> <ul style="list-style-type: none"> <li>- Thất bại: {</li> <li>    message: "Added fail",</li> <li>    success: false,</li> <li>    Data: []</li> </ul> <p>}</p>
		<p>Phương thức: PUT</p> <p>Mục đích: Cập nhật điểm số của sinh viên.</p> <p>Tham số đầu vào: {</p> <pre>"student": "SV001", "subject": "MATH101", "grade": 9.5</pre> <p>}</p> <p>Kết quả trả về: {</p> <pre>"message": "Grade updated", "success": true</pre> <p>}</p>
		<p>Phương thức: DELETE</p> <p>Mục đích: Xóa điểm của sinh viên.</p> <p>Tham số đầu vào: {</p> <pre>"student": "SV001", "subject": "MATH101"</pre>

STT	URL	Mô tả chi tiết
		<p>}</p> <p>Kết quả trả về:{</p> <p>    "message": "Grade deleted",</p> <p>    "success": true</p> <p>}</p>
10	/api/products/CourseRegistrations	<p>Phương thức: GET</p> <p>Mục đích: Lấy danh sách môn học sinh viên đã đăng ký.</p> <p>Tham số đầu vào: {</p> <p>    "studentId": "SV001"</p> <p>}</p> <p>Kết quả trả về:</p> <p>{</p> <p>    "registrations": [</p> <p>        {</p> <p>            "id": "5001",</p> <p>            "studentId": "SV001",</p> <p>            "subjectId": "MATH101",</p> <p>            "semester": "2024-1"</p> <p>        },</p> <p>        {</p> <p>            "id": "5002",</p> <p>            "studentId": "SV001",</p> <p>            "subjectId": "CS102",</p> <p>            "semester": "2024-1"</p> <p>        }</p> <p>    ],</p> <p>    "success": true</p> <p>}</p> <p>Phương thức:POST</p> <p>Mục đích: Sinh viên đăng ký môn học</p> <p>Tham số đầu vào: {</p> <p>    "studentId": "SV001",</p> <p>    "subjectId": "MATH101",</p>

STT	URL	Mô tả chi tiết
		<p>"semester": "2024-1" }  Kết quả trả về: {      "message": "Course registered",      "success": true  }</p> <p>Phương thức:PUT  Mục đích:Cập nhật đăng ký môn học  Tham số đầu vào: {      "id": "5001",      "subjectId": "CS102"  }</p> <p>Kết quả trả về: {      "message": "Course registration updated",      "success": true  }</p> <p>Phương thức: DELETE  Mục đích:Hủy đăng ký môn học  Tham số đầu vào: {      "id": "5001"  }</p> <p>Kết quả trả về: {      "message": "Course registration deleted",      "success": true  }</p>

## 4. Cài đặt

### 4.1. Lựa chọn công nghệ



Hình 4.1 Sơ đồ triển khai hệ thống

#### \*Mô tả sơ đồ:

Sơ đồ kiến trúc tổng thể của hệ thống trong Hình 4.1 mô tả quy trình thực hiện một ứng dụng Web. Người dùng sẽ sử dụng các chức năng của ứng dụng, các yêu cầu đó sẽ được gửi đến API trên Web Server. Web Server sẽ xử lý các yêu cầu đó và kết nối đến cơ sở dữ liệu, lấy dữ liệu và trả lại về cho Web Server. Sau khi Web Server nhận được dữ liệu sẽ trả về kết quả theo yêu cầu cho ứng dụng Web. Cuối cùng sẽ hiển thị kết quả đó cho người dùng.

#### \*Giới thiệu các công nghệ sử dụng:

- Công nghệ sử dụng tạo Server: NodeJs tạo các RESTful API cho bên client gọi để lấy dữ liệu từ cơ sở dữ liệu.
- Cơ sở dữ liệu sử dụng: MongoDB lưu trữ dữ liệu của phần mềm.
- Công nghệ sử dụng xây dựng Web Client: ReactJs thiết kế giao diện bên phía người dùng trên ứng dụng Web.

#### 4.1.1 Database



#### \*Giới thiệu:

MongoDB là một hệ quản trị cơ sở dữ liệu NoSQL mã nguồn mở phổ biến, được phát triển bởi MongoDB Inc. MongoDB sử dụng cấu trúc dữ liệu dạng document thay vì

bảng và dòng như trong hệ cơ sở dữ liệu quan hệ. Dữ liệu được lưu trữ dưới dạng BSON (Binary JSON), giúp linh hoạt trong việc truy vấn và xử lý dữ liệu. MongoDB thường được sử dụng cho các ứng dụng web, phân tán và xử lý dữ liệu lớn nhờ tính linh hoạt và khả năng mở rộng cao.

\*Các thành phần chính trong kiến trúc của MongoDB:

- **Query Language (Ngôn ngữ truy vấn):** MongoDB sử dụng ngôn ngữ truy vấn dựa trên JSON, cho phép thao tác dữ liệu linh hoạt hơn so với SQL truyền thống.
- **Storage Engine (Bộ máy lưu trữ):** MongoDB hỗ trợ nhiều bộ máy lưu trữ, bao gồm WiredTiger (bộ máy lưu trữ mặc định) và In-Memory Engine (lưu trữ dữ liệu trong RAM để tăng tốc truy vấn).
- **Replication (Sao chép dữ liệu):** MongoDB hỗ trợ Replication thông qua **Replica Set**, giúp đảm bảo dữ liệu đồng bộ giữa các máy chủ khác nhau và đảm bảo khả năng khôi phục khi hệ thống gặp sự cố.
- **Sharding (Phân mảnh dữ liệu):** MongoDB cho phép phân chia dữ liệu thành các mảnh nhỏ (“shards”) để tăng khả năng mở rộng ngang.
- **Aggregation Framework (Khung xử lý dữ liệu):** Cung cấp một cách linh hoạt để xử lý dữ liệu tổng hợp thay thế cho truy vấn SQL truyền thống.
- **Security (Bảo mật):** MongoDB cung cấp các tính năng bảo mật như xác thực người dùng, kiểm soát truy cập và mã hóa dữ liệu.
- **Indexing (Chỉ mục hóa):** MongoDB hỗ trợ nhiều loại chỉ mục như Single Field, Compound Indexes, Geospatial Indexes giúp tăng tốc truy vấn.

\*Ưu điểm và nhược điểm của MongoDB:

- **Ưu điểm:**
  - **Linh hoạt:** Do sử dụng cấu trúc document thay vì bảng quan hệ, MongoDB cho phép lưu trữ dữ liệu phi cấu trúc một cách linh hoạt.
  - **Mở rộng ngang:** Hỗ trợ sharding giúp dễ dàng tăng quy mô hệ thống khi dữ liệu tăng.
  - **Hiệu suất cao:** Hỗ trợ truy vấn nhanh chóng nhờ chỉ mục hóa và caching.
  - **Dễ sử dụng:** Các thao tác truy vấn dựa trên JSON rất gần gũi và trực quan.
  - **Bảo mật cao:** Hỗ trợ nhiều tính năng bảo mật và quyền truy cập chi tiết.
  - **Hỗ trợ từ cộng đồng:** Có cộng đồng người dùng lớn, dễ dàng học hỏi và khắc phục sự cố.

- **Nhược điểm:**
  - **Không hỗ trợ giao dịch ACID hoàn chỉnh:** MongoDB chỉ hỗ trợ giao dịch trên phạm vi document thay vì hệ thống.
  - **Tiêu tốn tài nguyên:** Do không hỗ trợ quan hệ, dữ liệu lặp lại nên chiếm dung lượng bộ nhớ lớn hơn so với RDB

#### **4.1.2 Server**



##### \*Giới thiệu:

Node.js là một JavaScript runtime được build dựa trên Chrome's V8 JavaScript engine. Node.js sử dụng mô hình event-driven, non-blocking I/O khiến nó trở nên nhẹ và hiệu quả. NodeJS được phát triển bởi Ryan Dahl vào năm 2009 và có thể chạy trên nhiều hệ điều hành khác nhau: OS X, Microsoft Windows, Linux. [\[5\]](#)

##### \*Lý do sử dụng:

- NodeJS được viết bằng JavaScript với cộng đồng người dùng lớn mạnh. Nếu bạn cần hỗ trợ gì về NodeJS thì sẽ nhanh chóng có người hỗ trợ bạn.
- Tốc độ xử lý nhanh do có cơ chế xử lý bất đồng độ (non-blocking) nên NodeJS có thể xử lý hàng ngàn kết nối cùng lúc mà không gặp bất cứ khó khăn nào.
- Dễ dàng mở rộng: Nếu bạn có nhu cầu phát triển website thì tính năng dễ dàng mở rộng của NodeJS là một lợi thế cực kỳ quan trọng.

#### **4.1.3 Web Client**



##### \*Giới thiệu:

React là một thư viện JavaScript declarative, hiệu quả và linh hoạt cho việc xây dựng giao diện người dùng. React cho phép bạn tạo những giao diện (UI) phức tạp từ những đoạn code nhỏ và độc lập. Những đoạn code này được gọi là “components”. [\[6\]](#)

\*Các thành phần cơ bản trong React:

- Virtual DOM (DOM ảo): Giúp tăng hiệu năng cho ứng dụng. Thay vì phải vẽ lại giao diện (rerender) mỗi lần thay đổi dữ liệu trong các component. Virtual DOM sẽ được tạo ra khi dữ liệu bị thay đổi, và được tính toán sự thay đổi giữa DOM ảo và DOM thật. Khi dữ liệu này thay đổi, trình duyệt sẽ rerender lại DOM đó.
- JSX (Javascript Syntax Extension): Là một dạng ngôn ngữ cho phép viết mã HTML trong Javascript nhanh hơn, an toàn hơn và dễ dàng hơn.
- Components: Ứng dụng web sử dụng ReactJS được xây dựng dựa trên các component. Và các component này có thể tái sử dụng ở nhiều nơi với các trạng thái và thuộc tính khác nhau. Mỗi component trong ReactJS có một trạng thái riêng, có thể thay đổi và ReactJS sẽ thực hiện cập nhật component dự trên sự thay đổi về trạng thái.
- Props và State:
  - Props: Giúp các component tương tác với nhau, component nhận dữ liệu sẽ gọi là props và những dữ liệu đó sẽ được component con render ra.
  - State: là một tính năng riêng tư của một component duy nhất. State có thể thay đổi output trong quá trình chạy để đáp ứng với các hành động nhất định.

\*Lý do sử dụng:

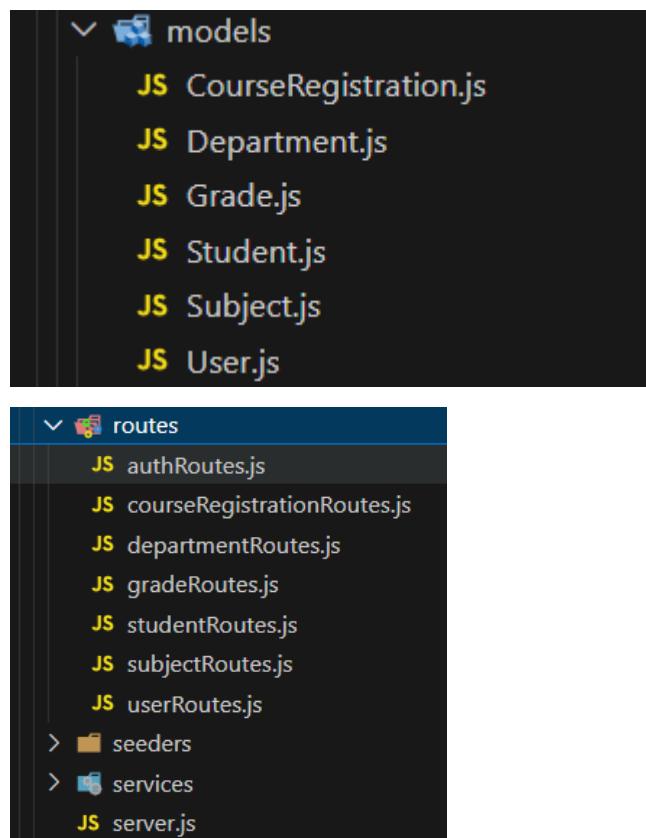
- Giúp viết mã Javascript dễ dàng hơn với JSX: ReactJS sử dụng JSX (Javascript Syntax Extention) hay còn gọi là phần bổ xung cú pháp Javascript. Là một sự hòa trộn giữa Javascript và XML.
- ReactJS cho phép tạo ra các component (thành phần) tương ứng với các giao diện. Các component này có thể tái sử dụng, hoặc kết hợp với các component khác để tạo giao diện hoàn chỉnh. Đây là chìa khóa giải quyết vấn đề khó khăn khi dự án ngày càng mở rộng.
- Thân thiện với SEO: Vì ReactJS có thể chạy tại phía Server, vì vậy dữ liệu trả về cho trình duyệt là văn bản HTML và nó không gây khó khăn cho Search Engine (công cụ tìm kiếm).
- Dễ dàng viết testcase để kiểm thử ứng dụng.
- Được nhiều thư viện hỗ trợ.

- Các chủ đề liên quan đến các vấn đề khó khăn khi phát triển ứng dụng web với ReactJS nhiều, hỗ trợ các nhà phát triển giải quyết vấn đề nhanh chóng

## 4.2. Cấu trúc mã nguồn

Link mã nguồn: <https://github.com/vjintageboy/N06-PTTKPM-NHOM3/tree/caochien>

### 4.2.1 Server



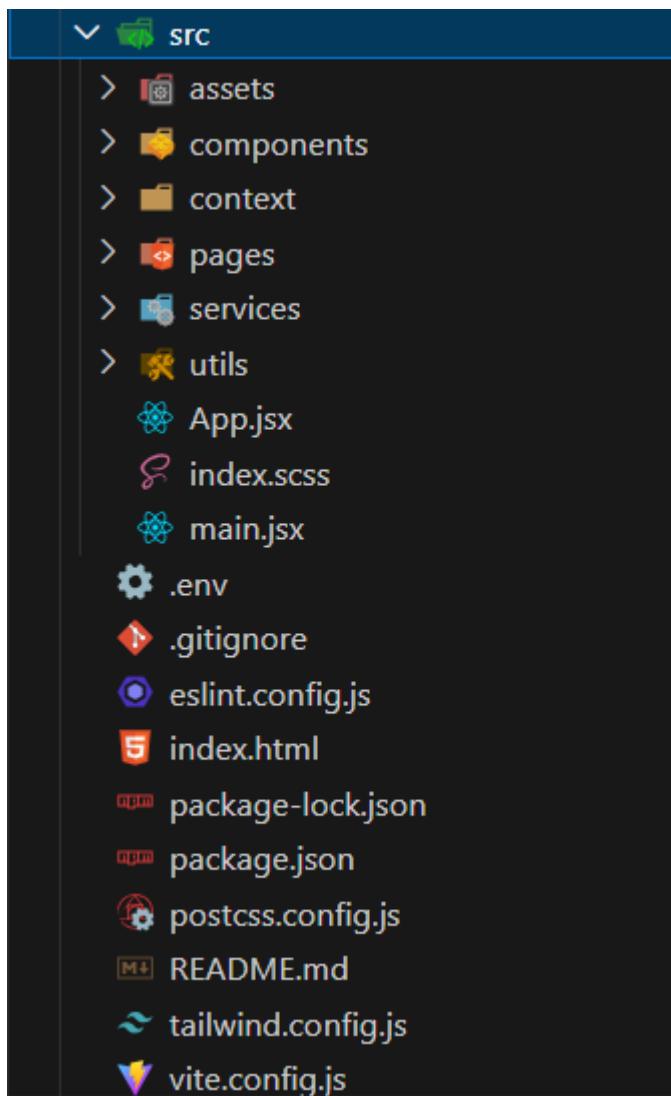
Hình 4.2 Các thư mục mã nguồn Server

STT	Thành phần	Mô tả
1	src	Thư mục chứa mã nguồn chương trình
2	src/models	Tệp chứa mã nguồn API truy suất với cơ sở dữ liệu của chương trình
4	src/models/CourseRegistration	Tệp chứa mã nguồn của API quản lý đăng ký môn học
5	src/models/Department	Tệp chứa mã nguồn của API quản lý khoa
6	src/models/Grade	Tệp chứa mã nguồn của API quản lý điểm

7	src/models/Student	Tệp chứa mã nguồn của API quản lý sinh viên
8	src/models/Subject	Tệp chứa mã nguồn của API quản lý môn học
9	src/models/User	Tệp chứa mã nguồn của API quản lý người dùng
10	src/routes/authRouters	Xử lý xác thực người dùng, đăng nhập, đăng ký và phân quyền.
11	src/routes/courRegistrationRoutes	Xử lý các API liên quan đến đăng ký môn học.
12	src/routes/departmentRoutes	Xử lý các API liên quan đến khoa.
13	src/routes/gradeRoutes	Xử lý các API liên quan đến điểm số của sinh viên.
14	src/routes/subjectRoutes	Xử lý các API liên quan đến môn học.
15	src/routes/userRoutes	Xử lý các API liên quan đến tài khoản người dùng.
16	src/routes/studentRoutes	Xử lý các API liên quan đến quản lý sinh viên.

Bảng 4.1 Chi tiết các thư mục mã nguồn Server

#### 4.2.2 Web Client



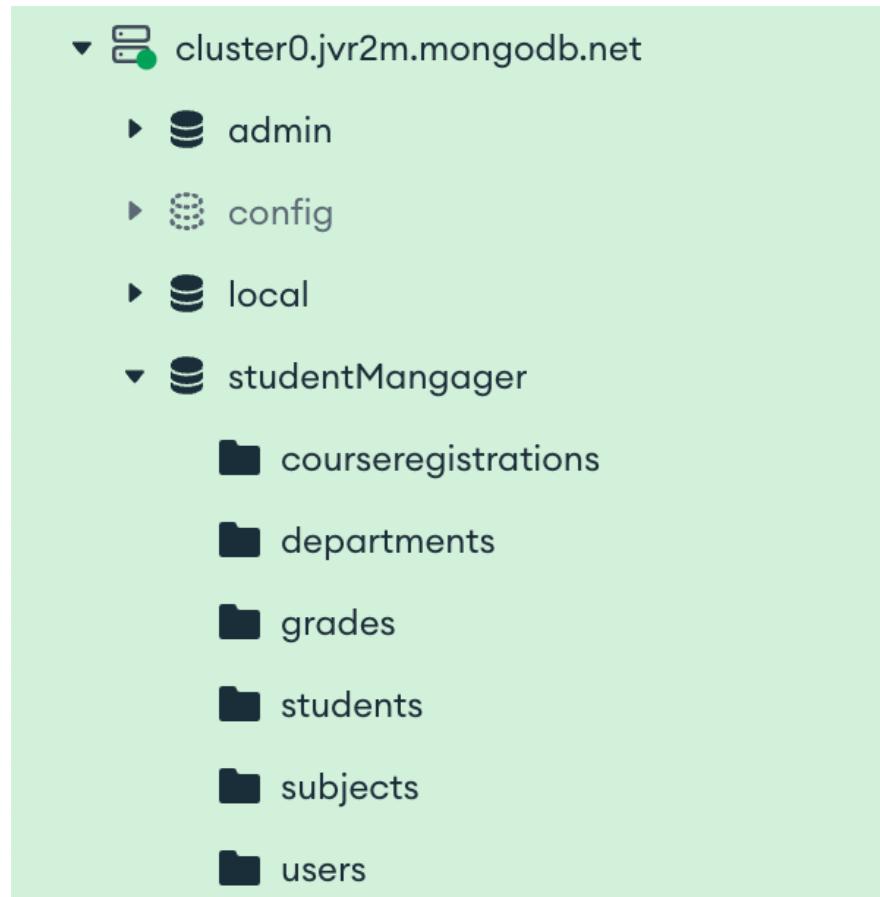
Hình 4.2 Các thư mục mã nguồn của Web Client

STT	Thành phần	Mô tả
1	src	Thư mục chứa mã nguồn chương trình
2	assets	Chứa các tài nguyên tĩnh như hình ảnh, biểu tượng, hoặc tệp CSS chung.
3	src/components	Chứa các thành phần giao diện dùng chung như button, modal, navbar, sidebar, v.v.
4	src/components/departments	Chứa các component liên quan đến quản lý khoa.
5	src/components/footer	Chứa phần footer (chân trang) của ứng dụng.
6	src/components/grades	Chứa các component liên quan đến quản lý điểm số.
7	src/components/header	Chứa thành phần header (tiêu đề) của trang.

8	src/components/ protectedRoutes	Xử lý logic bảo vệ các route yêu cầu quyền truy cập (ví dụ: chỉ quản trị viên mới có thể truy cập).
9	src/components/sidebar	Chứa giao diện thanh điều hướng bên trái của ứng dụng.
10	src/components/student	Chứa các component liên quan đến quản lý sinh viên.
11	src/components/subjects	Chứa các component liên quan đến quản lý môn học.
12	src/components/user	Chứa các component liên quan đến quản lý người dùng.

Bảng 4.3 Chi tiết các thư mục mã nguồn Web Client

#### 4.2.3 Database



Hình 4.4 Các thư mục mã nguồn của Database

STT	Thành phần	Mô tả
1	admin	Database hệ thống mặc định của MongoDB, dùng để quản lý người dùng và quyền truy cập.

2	config	Database dùng để lưu trữ thông tin cấu hình của MongoDB.
3	local	Database nội bộ của MongoDB, chứa dữ liệu tạm thời như thông tin replication.
4	courseregistrations	Lưu thông tin đăng ký môn học của sinh viên.
5	department	Lưu thông tin về các khoa trong trường.
6	Grades	Lưu điểm số của sinh viên.
7	Students	Lưu thông tin của sinh viên.
8	subjects	Lưu thông tin về các môn học.
9	users	Lưu thông tin tài khoản người dùng trong hệ thống (có thể là sinh viên, giáo viên hoặc quản trị viên).

Bảng 4.5 Chi tiết các thư mục mã nguồn Database

## 5. Kết luận

### 5.1. Những việc đã làm được và kết quả đạt được

Cơ bản đã hoàn thiện các nhóm chức năng sau:

- Hoàn thiện giao diện quản lý sinh viên với các chức năng cơ bản, hệ thống có thể lưu trữ và hiển thị thông tin sinh viên một cách chính xác, chức năng đăng ký môn học hoạt động ổn định.
- Hỗ trợ phân quyền tài khoản cơ bản cho sinh viên và Quản lý Khoa.
- Quản lý sinh viên: có thể thực hiện các thao tác thêm, sửa, xoá, tìm kiếm thông tin nhân viên. Cung cấp thông tin đầu vào cho chức năng Quản lý tài khoản người dùng để quản lý tài khoản của sinh viên và kết nối với quản lý tài khoản người dùng để quản lý thông tin về sinh viên.
- Quản lý nhóm quyền: có thể thực hiện thêm, sửa, xoá, xem chi tiết, tìm kiếm các nhóm quyền. Cung cấp thông tin đầu vào cho chức năng Quản lý tài khoản người dùng để thực hiện phân quyền cho tài khoản của các khoa .
- Quản lý dịch vụ: Cơ bản hoàn thiện xong giao diện theo thiết kế.
- Thống kê: có thể thực hiện lấy dữ liệu thống kê tổng quan, danh sách học sinh, danh sách khoa, danh sách lớp.
- Đăng ký tài khoản: đã hoàn thiện giao diện
- Đăng nhập: đã hoàn thiện giao diện, cho phép người dùng đăng nhập
- Đăng xuất: cho phép người dùng đăng xuất khỏi hệ thống

### 5.2. Những điều chưa làm được

- Hệ thống hiện tại mới chỉ dừng lại ở việc quản lý tài khoản cho sinh viên và cho phép họ đăng nhập vào hệ thống với các vai trò khác nhau, chưa hoàn thiện giao diện của sinh viên được phân quyền.
- Chưa tích hợp tính năng xuất báo cáo, bảng điểm dưới dạng file PDF hoặc Excel.
- Chưa có cơ chế thông báo (notification) khi có thay đổi về điểm số hoặc lịch học của sinh viên.
- Chưa có chức năng xem và thanh toán học phí.

### 5.3. Đề xuất trong tương lai

- Cải thiện bảo mật: Tích hợp xác thực hai yếu tố (2FA) và mã hóa dữ liệu.
- Mở rộng chức năng: Thêm tra cứu lịch học, lịch thi, thêm xuất báo cáo học tập

- Tối ưu hóa giao diện: Đảm bảo hệ thống hoạt động tốt trên cả desktop và thiết bị di động.

## Tài liệu tham khảo

- [1] K. CNTT, Slide bài giảng môn học Phân tích thiết kế phần mềm, 2024.
- [2] Sách giáo trình: Systems Analysis and Design\_ An Object-Oriented Approach with UML-Wiley (2020)
- [3] <https://topdev.vn/blog/restful-api-la-gi/>
- [4] <https://tenten.vn/tin-tuc/mysql-la-gi/>
- [5] <https://viblo.asia/p/gioi-thieu-ve-nodejs-3P0lPyRg5ox>
- [6] <https://monamedia.co/reactjs-la-gi/>