

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA TOÁN - TIN HỌC



PHÁT TRIỂN PHẦN MỀM HƯỚNG ĐỐI TƯỢNG

ĐỒ ÁN CUỐI KÌ
WEBSITE QUẢN LÝ TRUNG TÂM DẠY HỌC

Giảng viên phụ trách

ThS Phạm Thị Vương

Thành phố Hồ Chí Minh, tháng 01, năm 2025

Mục lục

1 Mở đầu	3
1.1 Giới thiệu thành viên	3
1.2 Lý do chọn đề tài	3
1.3 Mục tiêu nghiên cứu	3
1.4 Phạm vi và đối tượng nghiên cứu	4
1.5 Phương pháp nghiên cứu	4
2 Tổng quan về hệ thống	5
2.1 Giới thiệu về trung tâm dạy học và các yêu cầu quản lý	5
2.2 Khái quát về phát triển phần mềm hướng đối tượng	5
2.3 Tổng quan về công nghệ phát triển web (HTML,CSS,Java Spring,JavaScript, Frameworks,SQL Sever,ReactJS,Postman,Swagger UI)	5
2.3.1 Giới thiệu Java Spring	5
2.3.2 SQL Server	6
2.3.3 HTML, CSS, JavaScript và ReactJS	6
2.4 Phân tích các hệ thống tương tự	7
2.5 Xác định vấn đề và khoảng trống cần giải quyết	8
3 Phân tích yêu cầu và thiết kế hệ thống	9
3.1 Phân tích yêu cầu người dùng	9
3.1.1 Yêu cầu chức năng	9
3.1.2 Yêu cầu phi chức năng	9
3.2 Xây dựng biểu đồ use case	10
3.3 Biểu đồ lớp và phân tích đối tượng	14
3.4 Thiết kế cơ sở dữ liệu	18
3.4.1 Các bảng trong cơ sở dữ liệu	19
3.4.2 Mô hình ERD	24
3.4.3 Sơ đồ quan hệ	25
3.5 Một số sơ đồ chính	25
3.5.1 Sơ đồ hoạt động	25
3.5.2 Sơ đồ tuần tự	27
3.6 Thiết kế giao diện người dùng	33
3.6.1 Trang đăng nhập	33
3.6.2 Trang student/profile	34
3.6.3 Trang lịch học	36
3.6.4 Trang hộp thoại và thông báo	37
3.6.5 Giao diện quản lý học sinh	40
3.6.6 Giao diện quản lý giáo viên	41
3.6.7 Giao diện quản lý lớp học	42
4 Triển khai hệ thống	44

4.1	Lựa chọn công cụ và công nghệ	44
4.1.1	Frontend: HTML, CSS, JavaScript, ReactJS	44
4.1.2	Backend: Java Spring, và một số công cụ phát triển Postman và Swagger UI	44
4.1.3	Cơ sở dữ liệu: SQL SERVER	45
4.2	Phát triển và tích hợp các module	45
4.2.1	Quản lý học viên	45
4.2.2	Quản lý giảng viên	45
4.2.3	Quản lý lớp học	45
4.3	Kiểm thử hệ thống	45
4.3.1	Kiểm thử chức năng	45
4.3.2	Kiểm thử hiệu năng	48
5	Kết quả và đánh giá	49
5.1	Kết quả triển khai	49
5.1.1	Kết quả đạt được:	49
5.1.2	Việc chưa làm được:	49
6	Kết luận và hướng phát triển	50
6.1	Kết luận	50
6.2	Hướng phát triển tương lai	50
7	Tài liệu tham khảo	51
8	Phụ lục	52
8.1	Mã nguồn chính	52
8.1.1	Thành phần mã nguồn chính	52
8.1.2	Cách thức hoạt động của mã nguồn	53
8.1.3	Điểm nổi bật	53
8.2	Các biểu đồ và hình ảnh bổ sung	54

1 Mở đầu

Ngày nay, công nghệ thông tin đóng vai trò quan trọng trong nhiều lĩnh vực, đặc biệt là giáo dục. Việc ứng dụng công nghệ vào quản lý và tổ chức giảng dạy đã mang lại hiệu quả cao, giúp tiết kiệm thời gian và nâng cao chất lượng dạy học. Tuy nhiên, nhiều trung tâm đào tạo vẫn quản lý thông tin thủ công hoặc bằng các công cụ đơn giản, dẫn đến khó khăn trong việc lưu trữ, tra cứu và theo dõi.

Để giải quyết vấn đề này, nhóm đã thực hiện đồ án "Website quản lý trung tâm dạy học". Hệ thống được xây dựng với các chức năng chính như quản lý học viên, giảng viên, lớp học và lịch trình. Ngoài ra, hệ thống cũng cung cấp giao diện thân thiện, dễ sử dụng, giúp trung tâm vận hành hiệu quả hơn.

Đồ án không chỉ giúp hiện đại hóa quy trình quản lý mà còn góp phần nâng cao chất lượng giáo dục tại các trung tâm đào tạo. Nhóm hy vọng rằng sản phẩm này sẽ trở thành một giải pháp hữu ích, mang lại lợi ích lâu dài cho các trung tâm dạy học.

Trân trọng cảm ơn,

Nhóm thực hiện.

1.1 Giới thiệu thành viên

Thành viên: Nhóm 3

- * 22110072 – Vũ Trần Thanh Hương
- * 22110194 – Nguyễn Văn Duy Tâm
- * 22110204 – Nguyễn Thiện Thanh
- * 22110210 – Võ Xuân Thiện

1.2 Lý do chọn đề tài

Nhiều hệ thống quản lý giáo dục trực tuyến đã được phát triển, nhưng phần lớn tập trung vào các trường học lớn và không phù hợp với quy mô của các trung tâm dạy học nhỏ lẻ. Do đó, cần thiết kế một hệ thống phù hợp với nhu cầu cụ thể của các trung tâm này. Thực tiễn: Các trung tâm dạy học hiện nay phải quản lý một lượng lớn thông tin liên quan đến học viên, giảng viên, khóa học và lịch học. Việc quản lý hiệu quả các thông tin này đóng vai trò quan trọng trong việc nâng cao chất lượng giảng dạy và dịch vụ.

1.3 Mục tiêu nghiên cứu

- Xây dựng một trang web quản lý trung tâm dạy học với các chức năng: quản lý thông tin học viên, giảng viên, khóa học và lịch học.
- Áp dụng nguyên tắc phát triển phần mềm hướng đối tượng để đảm bảo tính tái sử dụng

và mở rộng của hệ thống.

- Đảm bảo hệ thống dễ sử dụng, bảo mật cao và đáp ứng các yêu cầu thực tế của trung tâm dạy học.
- Hệ thống được thiết kế dành riêng cho các trung tâm dạy học nhỏ lẻ, phù hợp với nhu cầu thực tế hơn so với các nền tảng quản lý lớn như Moodle hoặc Google Classroom.
- Độ tùy chỉnh cao, dễ dàng mở rộng theo yêu cầu của từng trung tâm.

1.4 Phạm vi và đối tượng nghiên cứu

Phạm vi: Hệ thống được thiết kế cho các trung tâm dạy học quy mô vừa và nhỏ, tập trung vào quản lý học viên, giảng viên và khóa học.

Đối tượng: Học viên, giảng viên và nhân viên quản lý trung tâm.

1.5 Phương pháp nghiên cứu

- **Phân tích yêu cầu:** Khảo sát nhu cầu thực tế tại các trung tâm dạy học.
- **Thiết kế hệ thống:** Ứng dụng kiến trúc
- **Phương pháp phát triển phần mềm hướng đối tượng (OOP):** Sử dụng mô hình OOP để phân tích và thiết kế hệ thống.
- **Microservices** với Java Spring cho backend và ReactJS cho frontend.
- **Triển khai cơ sở dữ liệu:** Tối ưu hóa truy vấn và lưu trữ dữ liệu bằng SQL Server.
- **Kiểm thử hệ thống:** Sử dụng Postman và Swagger UI để kiểm tra các API, đảm bảo hoạt động ổn định.

2 Tổng quan về hệ thống

2.1 Giới thiệu về trung tâm dạy học và các yêu cầu quản lý

Trung tâm dạy học là nơi cung cấp các khóa học và dịch vụ giáo dục nhằm bổ trợ kiến thức cho học viên ở nhiều lứa tuổi và trình độ khác nhau. Để vận hành hiệu quả, trung tâm cần quản lý nhiều thành phần quan trọng khác nhau nên cần đảm bảo đầy đủ các yêu cầu như:

- **Chức năng quản lý toàn diện:** Cung cấp các tính năng thêm, sửa, xóa thông tin liên quan đến học viên, giảng viên, lớp học, và môn học.
- **Hệ thống đăng nhập bảo mật:** Phân quyền rõ ràng giữa admin, giảng viên và học viên để đảm bảo an toàn thông tin.
- **Tích hợp thông báo và giao tiếp:** Hỗ trợ gửi thông báo, tin nhắn giữa các đối tượng trong hệ thống.
- **Theo dõi tiến trình học tập:** Hiển thị lịch học, tình trạng tham gia buổi học và điểm số của học viên.
- **Thanh toán dễ dàng:** Quản lý hóa đơn và trạng thái thanh toán với các tùy chọn linh hoạt.

2.2 Khái quát về phát triển phần mềm hướng đối tượng

Phát triển phần mềm hướng đối tượng là phương pháp lập trình dựa trên việc chia hệ thống thành các đối tượng, bao gồm:

- **Đóng gói:** Bảo vệ dữ liệu, chỉ cho phép truy cập thông qua các phương thức được định nghĩa.
- **Kế thừa:** Cho phép các lớp con kế thừa đặc điểm của lớp cha, giảm bớt mã lặp.
- **Đa hình:** Cung cấp các cách triển khai khác nhau cho một phương thức cụ thể.
- Ứng dụng các nguyên lý OOP giúp thiết kế hệ thống dễ bảo trì, nâng cấp và mở rộng.

2.3 Tổng quan về công nghệ phát triển web (HTML,CSS,Java Spring,Java Script Frameworks,SQL Sever,ReactJS,Postman,Swagger UI)

2.3.1 Giới thiệu Java Spring

Java Spring là một framework mạnh mẽ, được sử dụng phổ biến trong việc phát triển các ứng dụng Java, đặc biệt là ứng dụng web và enterprise. Framework này hỗ trợ phát triển theo kiến trúc MVC (Model-View-Controller), giúp tách biệt các thành phần trong ứng dụng để dễ dàng quản lý và bảo trì. Một số đặc điểm nổi bật của Spring:

- Hỗ trợ Dependency Injection (DI) và Inversion of Control (IoC).

- Tích hợp dễ dàng với các công nghệ khác như Hibernate, JPA, và các cơ sở dữ liệu.
- Cung cấp mô hình xử lý bảo mật, quản lý giao dịch và Rest API.
- Spring Boot là một phần của hệ sinh thái Spring, giúp thiết lập và cấu hình các ứng dụng nhanh chóng với cấu trúc rõ ràng.

2.3.2 SQL Server

SQL Server là hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) do Microsoft phát triển. Đây là công cụ mạnh mẽ giúp lưu trữ, truy vấn và quản lý dữ liệu một cách hiệu quả. SQL Server được sử dụng rộng rãi trong các ứng dụng doanh nghiệp nhờ vào các tính năng:

- Hỗ trợ T-SQL: Ngôn ngữ truy vấn mạnh mẽ, cho phép thực hiện các thao tác trên dữ liệu.
- Bảo mật dữ liệu cao: Hỗ trợ mã hóa, xác thực và kiểm soát truy cập dữ liệu.
- Khả năng mở rộng: Phù hợp cho các hệ thống từ quy mô nhỏ đến lớn.
- Tích hợp dễ dàng với Java thông qua JDBC: Hỗ trợ phát triển các ứng dụng dựa trên Java Spring.

2.3.3 HTML, CSS, JavaScript và ReactJS

- **HTML, CSS, JavaScript:**

- **HTML (Hypertext Markup Language):** Ngôn ngữ cấu trúc dùng để xây dựng khung và nội dung cho website.
- **CSS (Cascading Style Sheets):** Dùng để thiết kế giao diện, tạo phong cách và bố cục cho trang web.
- **JavaScript:** Ngôn ngữ lập trình phía client giúp tăng tính tương tác động trên website, như xử lý sự kiện và làm việc với API.

- **ReactJS:**

- ReactJS là thư viện JavaScript phổ biến do Facebook phát triển, giúp xây dựng giao diện người dùng (UI) hiệu quả và tương tác cao. Những đặc điểm chính:
 - + **Component-based:** Giao diện được chia nhỏ thành các thành phần dễ quản lý và tái sử dụng.
 - + **Virtual DOM:** Giảm thiểu sự thay đổi trên DOM thực tế, tăng hiệu suất ứng dụng.
 - + **Kết hợp dễ dàng với backend:** Kết hợp với RESTful API hoặc GraphQL để giao tiếp với server.
- ReactJS thường được sử dụng để xây dựng các ứng dụng đồ họa trực quan trong hệ thống quản lý.

- **Postman:**

- Postman là công cụ mạnh mẽ để kiểm tra và phát triển API. Công cụ này hỗ trợ việc gửi các yêu cầu HTTP/HTTPS (GET, POST, PUT, DELETE) để kiểm thử các endpoint trong ứng dụng. Một số tính năng chính:
 - + Gửi yêu cầu API và kiểm tra phản hồi nhanh chóng.
 - + Hỗ trợ tạo test script để tự động kiểm tra.
 - + Tích hợp dễ dàng với các hệ thống CI/CD.
- Postman rất hữu ích trong việc kiểm thử các API của ứng dụng Java Spring.

- **Swagger UI:**

- Swagger UI là công cụ cung cấp giao diện trực quan để xem và kiểm tra các RESTful API. Được tích hợp với OpenAPI Specification, Swagger UI giúp phát triển và kiểm thử API hiệu quả.
 - + Hiển thị cấu trúc API: Cung cấp giao diện tương tác để kiểm tra tất cả các endpoint của hệ thống.
 - + Hỗ trợ thử nghiệm API: Gửi trực tiếp yêu cầu từ giao diện Swagger mà không cần dùng công cụ khác.
 - + Tích hợp tốt với Java Spring: Spring hỗ trợ tự động tạo tài liệu Swagger thông qua các thư viện như Springfox hoặc Springdoc.
- Swagger UI giúp các nhà phát triển dễ dàng hiểu và làm việc với API của hệ thống.

2.4 Phân tích các hệ thống tương tự

- **Moodle**

- **Mô tả:** Moodle là một hệ thống quản lý học tập (Learning Management System - LMS) mã nguồn mở được sử dụng rộng rãi trên toàn thế giới. Nó hỗ trợ các chức năng như quản lý khóa học, theo dõi tiến độ học tập, và tổ chức kỳ thi trực tuyến.

- **Điểm mạnh:**

- + Mã nguồn mở, linh hoạt trong việc tùy chỉnh.
- + Cộng đồng hỗ trợ lớn với nhiều tài liệu hướng dẫn.
- + Cung cấp các tính năng mạnh mẽ như quản lý bài tập, điểm số và diễn đàn học tập.

- **Hạn chế:**

- + Giao diện phức tạp, không thân thiện với người dùng không có kỹ thuật.
- + Quá tải tính năng, không phù hợp với các trung tâm nhỏ hoặc chỉ cần quản lý

cơ bản.

- **Google Classroom**

- **Mô tả:** Google Classroom là nền tảng miễn phí dành cho giáo dục, tích hợp với các công cụ của Google như Google Docs, Google Drive.

- **Điểm mạnh:**

- + Đơn giản, dễ tích hợp với các công cụ phổ biến khác của Google.
- + Phù hợp cho việc giảng dạy trực tuyến, tạo và giao bài tập.
- + Dễ dàng truy cập từ nhiều thiết bị.

- **Hạn chế:**

- + Không cung cấp chức năng quản lý học viên toàn diện.
- + Thiếu các công cụ phân tích dữ liệu hoặc báo cáo chi tiết.

2.5 Xác định vấn đề và khoảng trống cần giải quyết

Các hệ thống trên tuy mạnh mẽ nhưng vẫn chưa hoàn toàn đáp ứng được nhu cầu của các trung tâm dạy học quy mô vừa và nhỏ. Những khoảng trống chính bao gồm:

- **Tính đơn giản và phù hợp:**

- + Các trung tâm nhỏ thường không cần quá nhiều tính năng phức tạp mà các hệ thống lớn như Moodle cung cấp.
- + Cần một giao diện dễ sử dụng cho cả người quản lý và học viên.

- **Quản lý đa dạng chức năng:**

- + Kết hợp các chức năng như quản lý học viên, điểm danh, lịch học, và thanh toán, vốn ít được tích hợp trong các hệ thống hiện có.

- **Tùy chỉnh linh hoạt:**

- + Cho phép quản lý điều chỉnh hệ thống theo nhu cầu cụ thể của trung tâm.

- **Hiệu năng cao và chi phí thấp:**

- + Hệ thống cần đảm bảo hiệu năng tốt mà không yêu cầu cơ sở hạ tầng phức tạp, giảm chi phí triển khai.

3 Phân tích yêu cầu và thiết kế hệ thống

3.1 Phân tích yêu cầu người dùng

3.1.1 Yêu cầu chức năng

- **Đối với Admin:**

- Quản lý thông tin người dùng: thêm, sửa, xóa người dùng như giảng viên, học viên.
- Quản lý lớp học: tạo, phân bổ giảng viên và theo dõi số lượng học viên.
- Quản lý môn học và nội dung giảng dạy.
- Theo dõi và cập nhật trạng thái thanh toán của học viên.
- Tạo và gửi thông báo đến toàn bộ hệ thống.

- **Đối với Giảng viên:**

- Quản lý danh sách học viên trong các lớp giảng dạy.
- Gửi bài tập, tài liệu học tập cho học viên và cập nhật tiến độ hoàn thành.
- Theo dõi điểm danh và đánh giá kết quả học tập của từng học viên.
- Gửi thông báo và giao tiếp với học viên thông qua hệ thống nhắn tin.

- **Đối với Học viên:**

- Đăng ký lớp học mới hoặc xem các lớp đã tham gia.
- Xem lịch học, thông báo, và các bài tập được giao.
- Nộp bài tập và theo dõi kết quả học tập.
- Theo dõi trạng thái thanh toán, lịch sử học tập, và tham gia các buổi học.

3.1.2 Yêu cầu phi chức năng

- **Hiệu năng:**

- Hệ thống phải xử lý nhanh các yêu cầu từ người dùng, đảm bảo tốc độ tải trang đạt tối ưu hóa

- **Bảo mật:**

- Thông tin người dùng và dữ liệu thanh toán cần được mã hóa.
- Phân quyền truy cập rõ ràng để ngăn chặn truy cập trái phép.

- **Thân thiện với người dùng:**

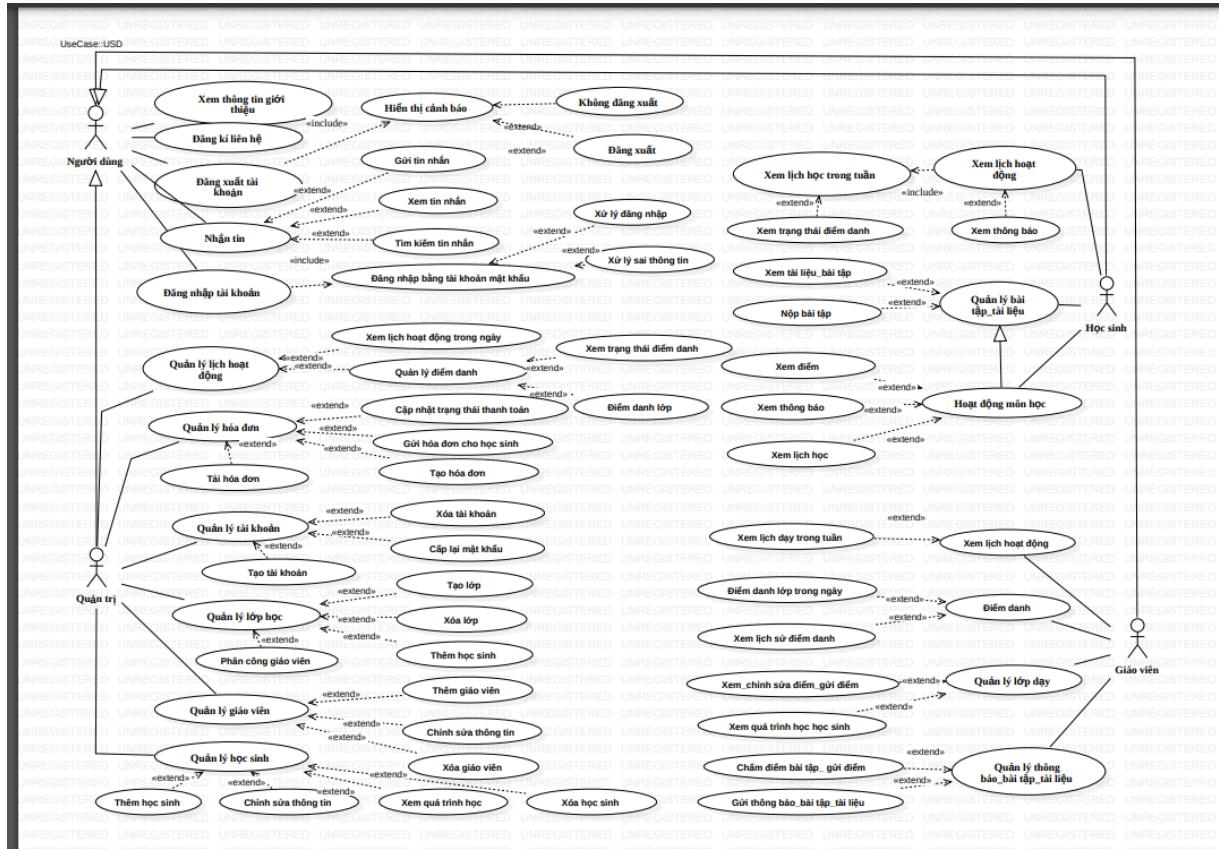
- Giao diện trực quan, dễ sử dụng cho cả người quản trị, giảng viên, và học viên.
- Hỗ trợ đa thiết bị, bao gồm máy tính và điện thoại di động.

- **Khả năng mở rộng:**

- Hệ thống cần hỗ trợ thêm người dùng, lớp học, và nội dung giảng dạy mà không ảnh hưởng đến hiệu suất.

3.2 Xây dựng biểu đồ use case

Sơ đồ use case tổng quát: Đặc tả Use case: Chi Tiết Hóa các Use Case:



Hình 1: Use Case

- **Use Case: Đăng nhập tài khoản**

- + **ID:** UC-01.
- + **Mục đích:** Cho phép người dùng truy cập hệ thống bằng tài khoản đã đăng ký.
- + **Tác nhân:** Học sinh, Giáo viên, Quản lý.
- + **Tiền điều kiện:** Tài khoản hợp lệ đã được tạo trên hệ thống.
- + **Hậu điều kiện:** Người dùng truy cập thành công vào giao diện chính của hệ thống.
- + **Luồng chính (Main Flow):**
 - ★ Người dùng mở trang đăng nhập.
 - ★ Nhập tên đăng nhập và mật khẩu.
 - ★ Nhấn nút 'Đăng nhập'.

★ Hệ thống kiểm tra thông tin:

- Nếu hợp lệ: Hệ thống chuyển đến giao diện chính.
- Nếu không hợp lệ: Chuyển sang luồng ngoại lệ.

+ **Luồng ngoại lệ (Exception Flow):**

- ★ Sai thông tin đăng nhập: Hiển thị thông báo 'Tên đăng nhập hoặc mật khẩu không chính xác'.
- ★ Tài khoản bị khóa: Hiển thị thông báo 'Tài khoản bị khóa. Vui lòng liên hệ quản trị viên'.

- **Use Case: Đăng xuất tài khoản**

- + **ID:** UC-02.
- + **Mục đích:** Cho phép người dùng kết thúc phiên làm việc.
- + **Tác nhân:** Học sinh, Giáo viên, Quản lý.
- + **Tiền điều kiện:** Người dùng đang trong trạng thái đăng nhập.
- + **Hậu điều kiện:** Người dùng được chuyển đến trang đăng nhập.
- + **Luồng chính (Main Flow):**

- ★ Người dùng nhấn nút 'Đăng xuất'.
- ★ Hệ thống kết thúc phiên làm việc và trả về giao diện đăng nhập.

- **Use Case: Quản lý tài khoản**

- + **ID:** UC-03.
- + **Mục đích:** Quản lý thông tin tài khoản người dùng (tạo, chỉnh sửa, xóa).
- + **Tác nhân:** Quản lý.
- + **Tiền điều kiện:** Quản lý đăng nhập thành công và có quyền truy cập.
- + **Hậu điều kiện:** Thông tin tài khoản được cập nhật hoặc xóa thành công.
- + **Luồng chính (Main Flow):**

- ★ Quản lý truy cập mục 'Quản lý tài khoản'.
- ★ Thực hiện một trong các chức năng:
 - Tạo tài khoản: Nhập thông tin tài khoản mới và nhấn lưu.
 - Chỉnh sửa tài khoản: Chọn tài khoản, thay đổi thông tin, và lưu.
 - Xóa tài khoản: Chọn tài khoản cần xóa và xác nhận.

+ **Luồng ngoại lệ (Exception Flow):**

- * Thông tin không hợp lệ: Hiển thị thông báo 'Thông tin không hợp lệ. Vui lòng kiểm tra lại'.

- **Use Case: Quản lý học sinh**

- + **ID:** UC-04.
- + **Mục đích:** Quản lý danh sách học sinh trong trung tâm.
- + **Tác nhân:** Quản lý.
- + **Tiền điều kiện:** Quản lý đã đăng nhập.
- + **Hậu điều kiện:** Danh sách học sinh được cập nhật chính xác.
- + **Luồng chính (Main Flow):**
 - * Quản lý truy cập mục 'Quản lý học sinh'.
 - * Thực hiện một trong các thao tác:
 - Thêm học sinh: Nhập thông tin và lưu.
 - Sửa thông tin: Chọn học sinh cần chỉnh sửa và lưu thay đổi.
 - Xóa học sinh: Chọn học sinh cần xóa và xác nhận.
- + **Luồng ngoại lệ (Exception Flow):**
 - * Thông tin thiếu hoặc không hợp lệ: Hiển thị thông báo 'Vui lòng nhập đầy đủ thông tin'.
 - * Học sinh không tồn tại: Hiển thị thông báo 'Học sinh không tìm thấy trong hệ thống'.

- **Use Case: Quản lý giáo viên**

- + **ID:** UC-05.
- + **Mục đích:** Quản lý danh sách giáo viên trong trung tâm.
- + **Tác nhân:** Quản lý.
- + **Tiền điều kiện:** Quản lý đăng nhập và có quyền truy cập.
- + **Hậu điều kiện:** Danh sách giáo viên được cập nhật.
- + **Luồng chính (Main Flow):**
 - * Quản lý truy cập mục 'Quản lý giáo viên'.
 - * Thực hiện:
 - Thêm giáo viên: Nhập thông tin và lưu.
 - Chỉnh sửa giáo viên: Cập nhật thông tin cần thay đổi và lưu.
 - Xóa giáo viên: Chọn giáo viên cần xóa và xác nhận.

+ **Luồng ngoại lệ (Exception Flow):**

- ★ Thông tin không hợp lệ: Hiển thị thông báo 'Thông tin nhập không chính xác'.

- **Use Case: Quản lý lớp học**

+ **ID:** UC-06.

+ **Mục đích:** Quản lý thông tin và danh sách các lớp học.

+ **Tác nhân:** Quản lý.

+ **Tiền điều kiện:** Quản lý đã đăng nhập và có quyền truy cập.

+ **Hậu điều kiện:** Danh sách lớp học được cập nhật.

+ **Luồng chính (Main Flow):**

- ★ Quản lý truy cập mục 'Quản lý lớp học'.

★ Thực hiện:

- Tạo lớp: Nhập thông tin lớp học mới và lưu.
- Thêm học sinh vào lớp: Chọn lớp và thêm học sinh.
- Phân công giáo viên: Chọn giáo viên và gán vào lớp.
- Xóa lớp: Chọn lớp cần xóa và xác nhận.

+ **Luồng ngoại lệ (Exception Flow):**

- ★ Lớp không tồn tại: Hiển thị thông báo 'Không tìm thấy lớp học'.

- **Use Case: Xem lịch học**

+ **ID:** UC-07.

+ **Mục đích:** Hiển thị lịch học của học sinh.

+ **Tác nhân:** Học sinh.

+ **Tiền điều kiện:** Học sinh đã đăng nhập thành công.

+ **Hậu điều kiện:** Lịch học được hiển thị.

+ **Luồng chính (Main Flow):**

- ★ Học sinh truy cập mục 'Xem lịch học'.

★ Chọn khoảng thời gian (ngày, tuần).

★ Hệ thống hiển thị chi tiết lịch học.

+ **Luồng ngoại lệ (Exception Flow):**

- ★ Không có lịch học: Hiển thị thông báo 'Không có lịch học cho thời gian đã chọn'.

- **Use Case: Điểm danh**

+ **ID:** UC-08.

+ **Mục đích:** Ghi nhận trạng thái có mặt/vắng mặt của học sinh.

+ **Tác nhân:** Giáo viên.

+ **Tiền điều kiện:** Giáo viên đã đăng nhập và có quyền điểm danh.

+ **Hậu điều kiện:** Trạng thái điểm danh được lưu vào hệ thống.

+ **Luồng chính (Main Flow):**

- ★ Giáo viên truy cập mục 'Điểm danh'.

- ★ Chọn lớp học cần điểm danh.

- ★ Ghi nhận trạng thái điểm danh cho từng học sinh.

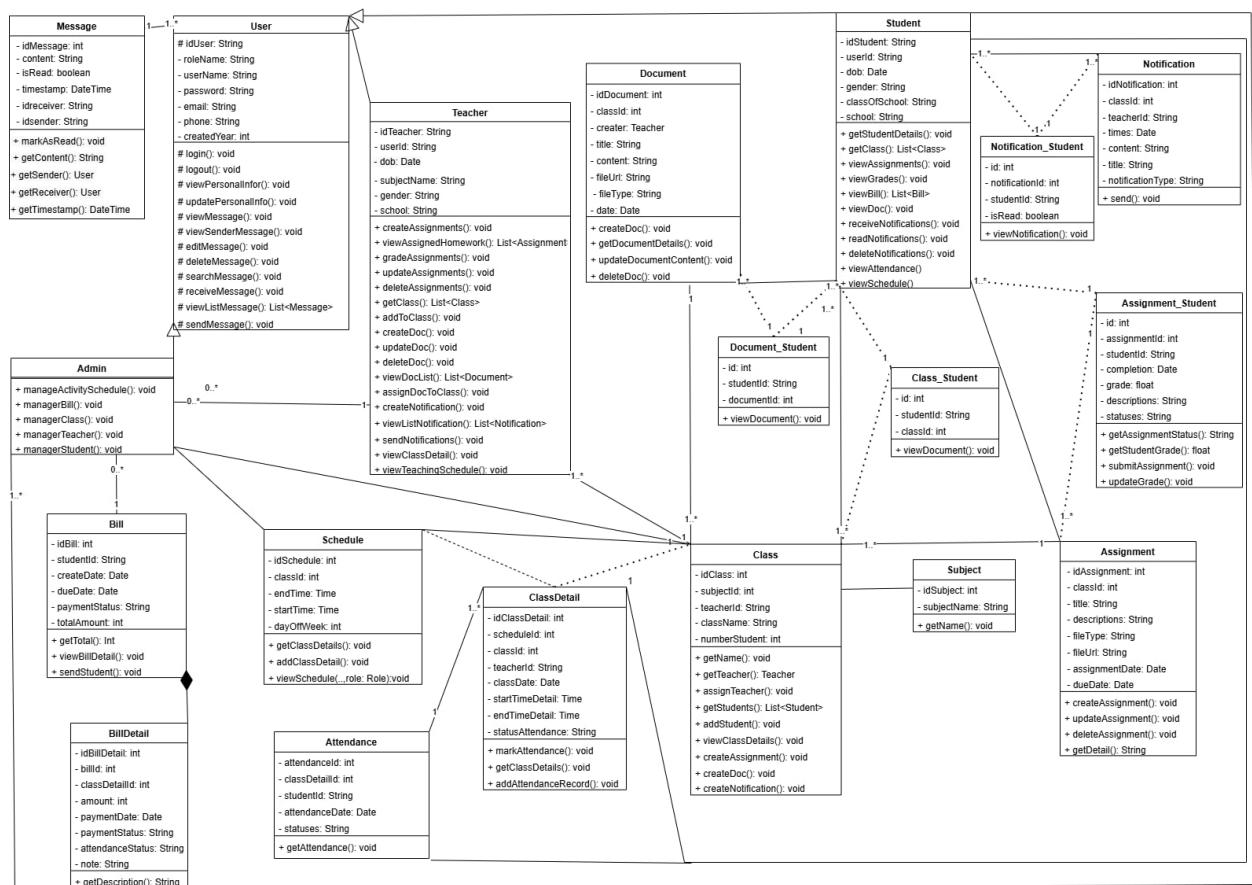
- ★ Lưu kết quả.

+ **Luồng ngoại lệ (Exception Flow):**

- ★ Lỗi lưu dữ liệu: Hiển thị thông báo 'Không thể lưu điểm danh. Vui lòng thử lại'.

3.3 Biểu đồ lớp và phân tích đối tượng

Biểu đồ lớp:



Hình 2: Biểu Đồ Lớp

Phân tích đối tượng:

- Users:

- + **Mô tả:** Đại diện cho người dùng trong hệ thống, gồm cả quản trị viên, giảng viên và học viên.
- + **Thuộc tính:**
 - * id_user: ID của người dùng (Khóa chính).
 - * username: Tên đăng nhập của người dùng.
 - * password: Mật khẩu được mã hóa.
 - * role_name: Vai trò (Admin, Teacher, Student).
 - * email: Địa chỉ email.
 - * phone: Số điện thoại.
 - * created_year: Năm tạo tài khoản.
- + **Mối quan hệ:** Kết nối với Teacher và Student thông qua user_id.

- Teacher:

- + **Mô tả:** Đại diện cho giảng viên trong hệ thống.
- + **Thuộc tính:**
 - * id_teacher: ID của giảng viên (Khóa chính).
 - * user_id: Liên kết đến bảng Users.
 - * subject_name: Môn học mà giảng viên dạy.
 - * dob: Ngày sinh của giảng viên.
 - * gender: Giới tính.
 - * school: Tên trường giảng viên công tác.
- + **Mối quan hệ:** Gắn với Users.
- + **Liên kết với Classes thông qua teacher_id.**

- Student:

- + **Mô tả:** Đại diện cho học viên trong hệ thống.
- + **Thuộc tính:**
 - * id_student: ID của học viên (Khóa chính).
 - * dob: Ngày sinh.
 - * gender: Giới tính.

- * class_of_school: Khối lớp học.

- * school: Trường học hiện tại.

- + **Mối quan hệ:** Gắn với Users.

- + Tham gia các lớp học qua ClassStudent và Schedule.

- + Liên kết với Bill, Attendance, và AssignmentStudent.

- **Subjects:**

- + **Mô tả:** Đại diện cho môn học được quản lý trong hệ thống.

- + **Thuộc tính:**

- * id_subject: ID của môn học (Khóa chính).

- * subject_name: Tên môn học.

- + **Mối quan hệ:** Kết nối với Classes qua subject_id.

- **Classes:**

- + **Mô tả:** Đại diện cho lớp học, liên kết giảng viên, học viên và môn học.

- + **Thuộc tính:**

- * id_class: ID của lớp học (Khóa chính).

- * subject_id: Liên kết đến bảng Subjects.

- * teacher_id: Liên kết đến bảng Teacher.

- * class_name: Tên lớp học.

- * number_student: Số lượng học viên trong lớp.

- * start_time: Thời gian bắt đầu.

- * end_time: Thời gian kết thúc.

- + **Mối quan hệ:** Liên kết với Teacher, Subjects, và Student qua ClassStudent.

- **ClassStudent:**

- + **Mô tả:** Liên kết giữa học viên và lớp học.

- + **Thuộc tính:**

- * id: ID của liên kết (Khóa chính).

- * student_id: Liên kết đến bảng Student.

- * class_id: Liên kết đến bảng Classes.

- + **Mối quan hệ:** Nối Student và Classes.

- **Schedule:**

+ **Mô tả:** Quản lý lịch học cho từng lớp và học viên.

+ **Thuộc tính:**

- ★ id_schedule: ID lịch học (Khóa chính).

- ★ student_id: Liên kết đến bảng Student.

- ★ class_id: Liên kết đến bảng Classes.

- ★ date: Ngày học.

+ **Mối quan hệ:** Liên kết Classes và Student.

- **Attendance:**

+ **Mô tả:** Theo dõi điểm danh của học viên trong từng buổi học.

+ **Thuộc tính:**

- ★ id_attendance: ID điểm danh (Khóa chính).

- ★ class_detail_id: Liên kết đến chi tiết lớp học.

- ★ student_id: Liên kết đến bảng Student.

- ★ attendance_date: Ngày điểm danh.

- ★ status: Trạng thái điểm danh (Present/Absent).

+ **Mối quan hệ:** Liên kết với ClassDetail và Student.

- **Bill:**

+ **Mô tả:** Quản lý hóa đơn thanh toán của học viên.

+ **Thuộc tính:**

- ★ id_bill: ID hóa đơn (Khóa chính).

- ★ student_id: Liên kết đến bảng Student.

- ★ create_date: Ngày tạo.

- ★ due_date: Hạn thanh toán.

- ★ payment_status: Trạng thái thanh toán (Paid/Unpaid).

- ★ total_amount: Tổng số tiền.

+ **Mối quan hệ:** Liên kết với Student qua student_id.

- **Assignment:**

+ **Mô tả:** Quản lý bài tập được giao trong lớp học.

+ **Thuộc tính:**

- ★ id_assignment: ID bài tập (Khóa chính).

- ★ description: Mô tả bài tập.
- ★ assign_date: Ngày giao bài.
- ★ due_date: Hạn nộp bài.

+ **Mối quan hệ:** Liên kết với học viên qua AssignmentStudent.

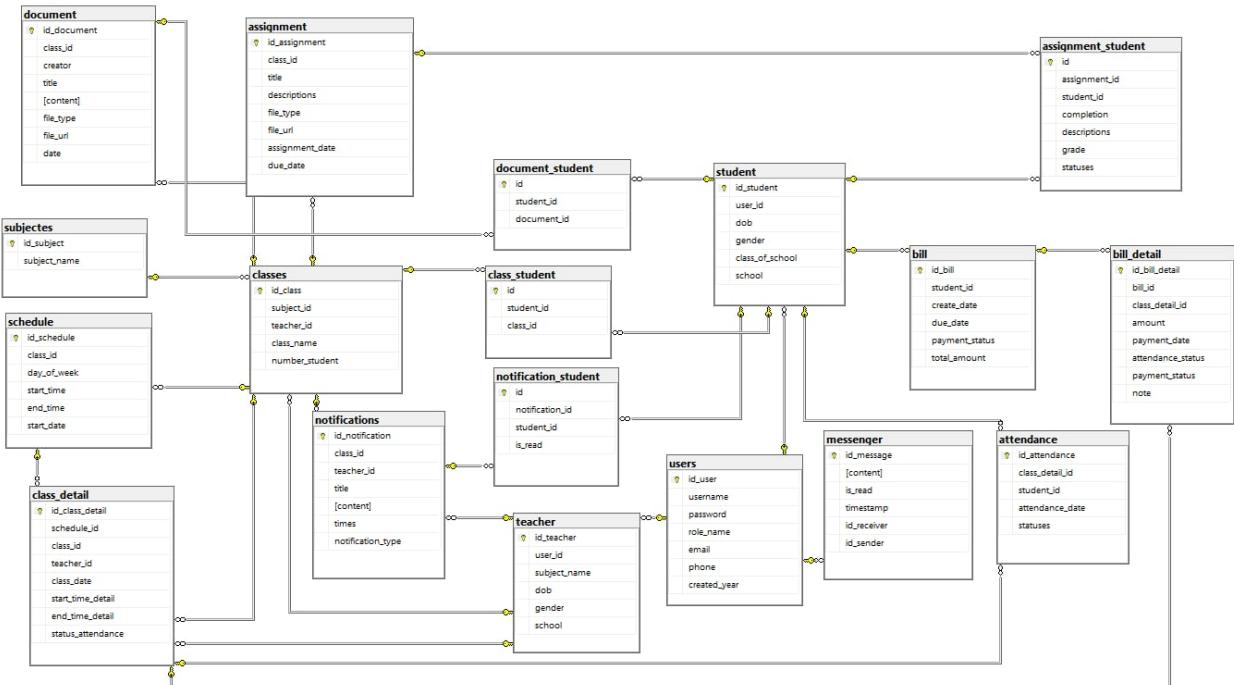
Mối quan hệ tổng quát:

- **Kế thừa:** Teacher và Student kế thừa từ User.

- **Liên kết chính:**

- + Classes là trung tâm, kết nối Teacher, Subjects, và Student.
- + Bill, Attendance, và Assignment hỗ trợ quản lý dữ liệu liên quan đến học viên.

3.4 Thiết kế cơ sở dữ liệu



Hình 3: Thiết kế CSDL

3.4.1 Các bảng trong cơ sở dữ liệu

- Bảng users:

	Column Name	Data Type	Allow Nulls
!	id_user	char(8)	<input type="checkbox"/>
	username	varchar(50)	<input type="checkbox"/>
	password	varchar(255)	<input type="checkbox"/>
	role_name	nvarchar(10)	<input type="checkbox"/>
	email	varchar(100)	<input checked="" type="checkbox"/>
	phone	varchar(20)	<input checked="" type="checkbox"/>
	created_year	int	<input type="checkbox"/>
▶			

Hình 4: Bảng users

- Bảng teacher:

	Column Name	Data Type	Allow Nulls
!	id_teacher	char(8)	<input type="checkbox"/>
	user_id	char(8)	<input type="checkbox"/>
	subject_name	nvarchar(25)	<input checked="" type="checkbox"/>
	dob	date	<input type="checkbox"/>
	gender	varchar(6)	<input type="checkbox"/>
	school	varchar(100)	<input checked="" type="checkbox"/>
▶			

Hình 5: Bảng teacher

- Bảng students:

	Column Name	Data Type	Allow Nulls
!	id_student	char(8)	<input type="checkbox"/>
	user_id	char(8)	<input type="checkbox"/>
	dob	date	<input type="checkbox"/>
	gender	varchar(6)	<input type="checkbox"/>
	class_of_school	nvarchar(2)	<input type="checkbox"/>
	school	nvarchar(100)	<input checked="" type="checkbox"/>
▶			

Hình 6: Bảng students

- Bảng subject:

	Column Name	Data Type	Allow Nulls
!	id_subject	int	<input type="checkbox"/>
	subject_name	nvarchar(100)	<input type="checkbox"/>
▶			

Hình 7: Bảng subject

- **Bảng schedule:**

Column Name	Data Type	Allow Nulls
id_schedule	int	<input type="checkbox"/>
class_id	int	<input type="checkbox"/>
day_of_week	int	<input type="checkbox"/>
start_time	time(7)	<input type="checkbox"/>
end_time	time(7)	<input type="checkbox"/>
start_date	date	<input type="checkbox"/>

Hình 8: Bảng schedule

- **Bảng notification:**

Column Name	Data Type	Allow Nulls
id_notification	int	<input type="checkbox"/>
class_id	int	<input type="checkbox"/>
teacher_id	char(8)	<input type="checkbox"/>
title	nvarchar(255)	<input type="checkbox"/>
[content]	nvarchar(MAX)	<input checked="" type="checkbox"/>
times	date	<input checked="" type="checkbox"/>
notification_type	nvarchar(50)	<input checked="" type="checkbox"/>

Hình 9: Bảng notification

- **Bảng notification student:**

Column Name	Data Type	Allow Nulls
id	int	<input type="checkbox"/>
notification_id	int	<input type="checkbox"/>
student_id	char(8)	<input type="checkbox"/>
is_read	bit	<input type="checkbox"/>

Hình 10: Bảng notification student

- **Bảng message:**

Column Name	Data Type	Allow Nulls
id_message	int	<input type="checkbox"/>
[content]	text	<input type="checkbox"/>
is_read	bit	<input type="checkbox"/>
timestamp	datetime	<input checked="" type="checkbox"/>
id_receiver	char(8)	<input checked="" type="checkbox"/>
id_sender	char(8)	<input checked="" type="checkbox"/>

Hình 11: Bảng message

- **Bảng document student:**

Column Name	Data Type	Allow Nulls
id	int	<input type="checkbox"/>
student_id	char(8)	<input type="checkbox"/>
document_id	int	<input type="checkbox"/>
		<input type="checkbox"/>

Hình 12: Bảng document student

- **Bảng document:**

Column Name	Data Type	Allow Nulls
id_document	int	<input type="checkbox"/>
class_id	int	<input type="checkbox"/>
creator	nvarchar(100)	<input checked="" type="checkbox"/>
title	nvarchar(100)	<input checked="" type="checkbox"/>
[content]	nvarchar(MAX)	<input checked="" type="checkbox"/>
file_type	nvarchar(20)	<input checked="" type="checkbox"/>
file_url	nvarchar(255)	<input checked="" type="checkbox"/>
date	date	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Hình 13: Bảng document

- **Bảng classes:**

Column Name	Data Type	Allow Nulls
id_class	int	<input type="checkbox"/>
subject_id	int	<input type="checkbox"/>
teacher_id	char(8)	<input type="checkbox"/>
class_name	nvarchar(100)	<input type="checkbox"/>
number_student	int	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Hình 14: Bảng classes

- **Bảng classes student:**

Column Name	Data Type	Allow Nulls
id	int	<input type="checkbox"/>
student_id	char(8)	<input type="checkbox"/>
class_id	int	<input type="checkbox"/>
		<input type="checkbox"/>

Hình 15: Bảng classes student

- **Bảng classdetail:**

Column Name	Data Type	Allow Nulls
id_class_detail	int	<input type="checkbox"/>
schedule_id	int	<input type="checkbox"/>
class_id	int	<input type="checkbox"/>
teacher_id	char(8)	<input type="checkbox"/>
class_date	date	<input type="checkbox"/>
start_time_detail	time(7)	<input type="checkbox"/>
end_time_detail	time(7)	<input type="checkbox"/>
status_attendance	char(20)	<input type="checkbox"/>

Hình 16: Bảng classdetail

- **Bảng billdetail:**

Column Name	Data Type	Allow Nulls
id_bill_detail	int	<input type="checkbox"/>
bill_id	int	<input checked="" type="checkbox"/>
class_detail_id	int	<input checked="" type="checkbox"/>
amount	decimal(10, 2)	<input type="checkbox"/>
payment_date	date	<input checked="" type="checkbox"/>
attendance_status	nvarchar(50)	<input checked="" type="checkbox"/>
payment_status	nvarchar(50)	<input checked="" type="checkbox"/>
note	nvarchar(255)	<input checked="" type="checkbox"/>

Hình 17: Bảng billdetail

- **Bảng bill:**

Column Name	Data Type	Allow Nulls
id_bill	int	<input type="checkbox"/>
student_id	char(8)	<input type="checkbox"/>
create_date	date	<input checked="" type="checkbox"/>
due_date	date	<input checked="" type="checkbox"/>
payment_status	nvarchar(50)	<input type="checkbox"/>
total_amount	decimal(10, 2)	<input type="checkbox"/>

Hình 18: Bảng bill

- **Bảng attendance:**

Column Name	Data Type	Allow Nulls
id_attendance	int	<input type="checkbox"/>
class_detail_id	int	<input type="checkbox"/>
student_id	char(8)	<input type="checkbox"/>
attendance_date	date	<input type="checkbox"/>
statuses	varchar(50)	<input type="checkbox"/>
▶		<input type="checkbox"/>

Hình 19: Bảng attendance

- **Bảng assignment student:**

Column Name	Data Type	Allow Nulls
id	int	<input type="checkbox"/>
assignment_id	int	<input type="checkbox"/>
student_id	char(8)	<input type="checkbox"/>
completion	date	<input checked="" type="checkbox"/>
descriptions	nvarchar(255)	<input checked="" type="checkbox"/>
grade	float	<input checked="" type="checkbox"/>
statuses	nvarchar(15)	<input checked="" type="checkbox"/>
▶		<input type="checkbox"/>

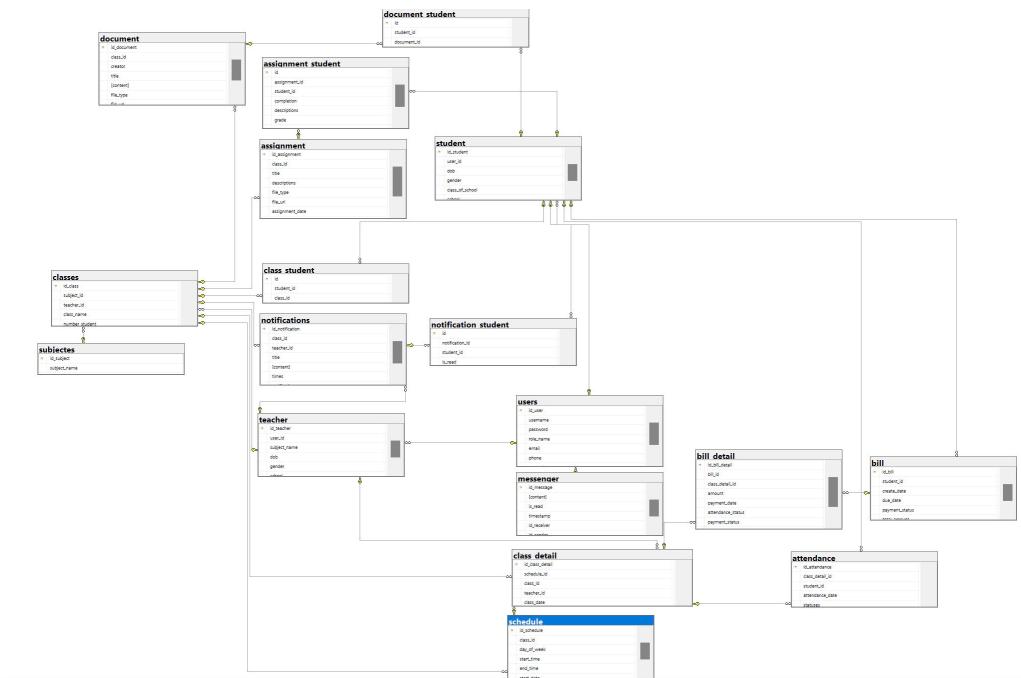
Hình 20: Bảng assignment student

- **Bảng assignment:**

Column Name	Data Type	Allow Nulls
id_assignment	int	<input type="checkbox"/>
class_id	int	<input type="checkbox"/>
title	nvarchar(100)	<input type="checkbox"/>
descriptions	nvarchar(MAX)	<input checked="" type="checkbox"/>
file_type	nvarchar(20)	<input checked="" type="checkbox"/>
file_url	nvarchar(255)	<input checked="" type="checkbox"/>
assignment_date	date	<input type="checkbox"/>
due_date	date	<input type="checkbox"/>
▶		<input type="checkbox"/>

Hình 21: Bảng assignment

3.4.2 Mô hình Erd



Hình 22: Mô HÌnh Erd

3.4.3 Sơ đồ quan hệ



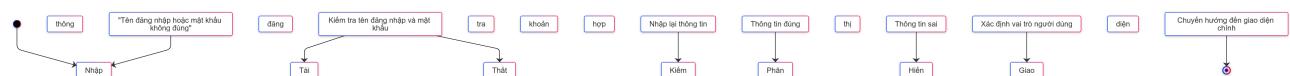
dbdiagram.io

Hình 23: Sơ Đồ Quan Hệ

3.5 Một số sơ đồ sơ đồ chính

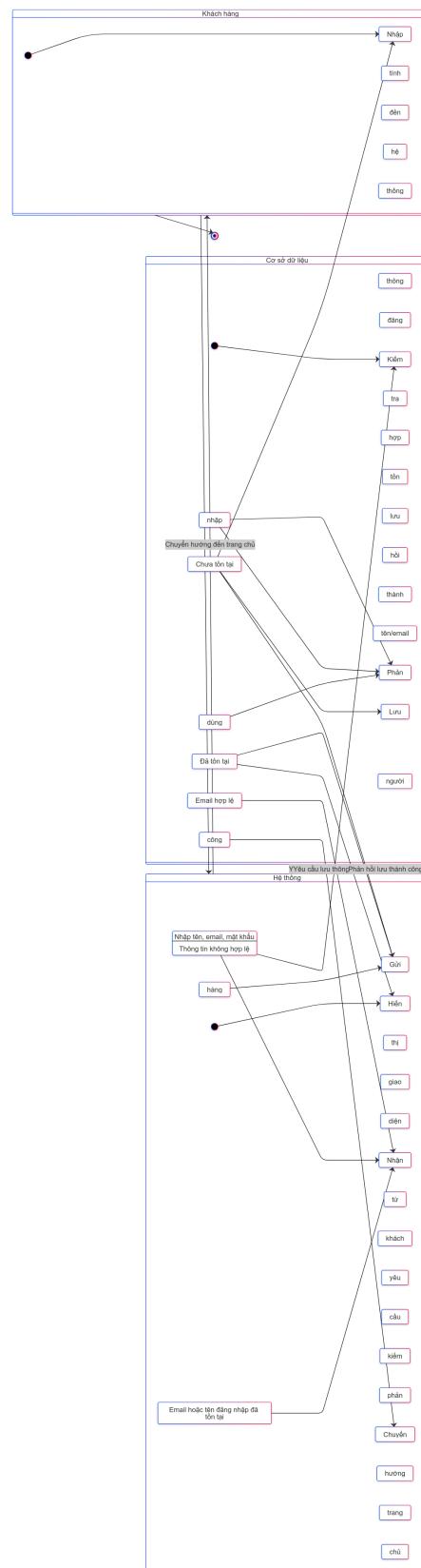
3.5.1 Sơ đồ hoạt động

Sơ đồ hoạt động cho chức năng đăng nhập:



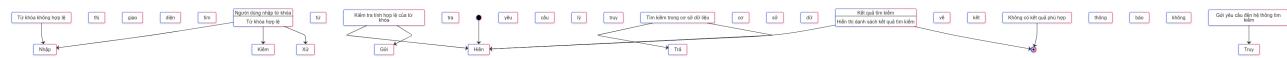
Hình 24: Sơ đồ hoạt động cho chức năng đăng nhập

Sơ đồ hoạt động cho chức năng đăng ký:



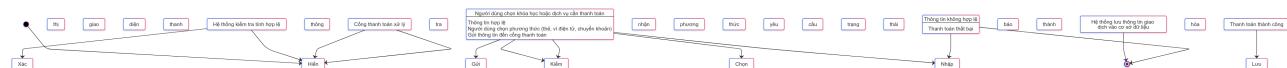
Hình 25: Sơ đồ hoạt động cho chức năng đăng ký

Sơ đồ hoạt động cho chức năng tìm kiếm:



Hình 26: Sơ đồ hoạt động cho chức năng tìm kiếm

Sơ đồ hoạt động cho chức năng thanh toán:



Hình 27: Sơ đồ hoạt động cho chức năng thanh toán

Sơ đồ hoạt động cho chức năng giao bài tập:



Hình 28: Sơ đồ hoạt động cho chức năng giao bài tập

Sơ đồ hoạt động cho chức năng đổi mật khẩu:

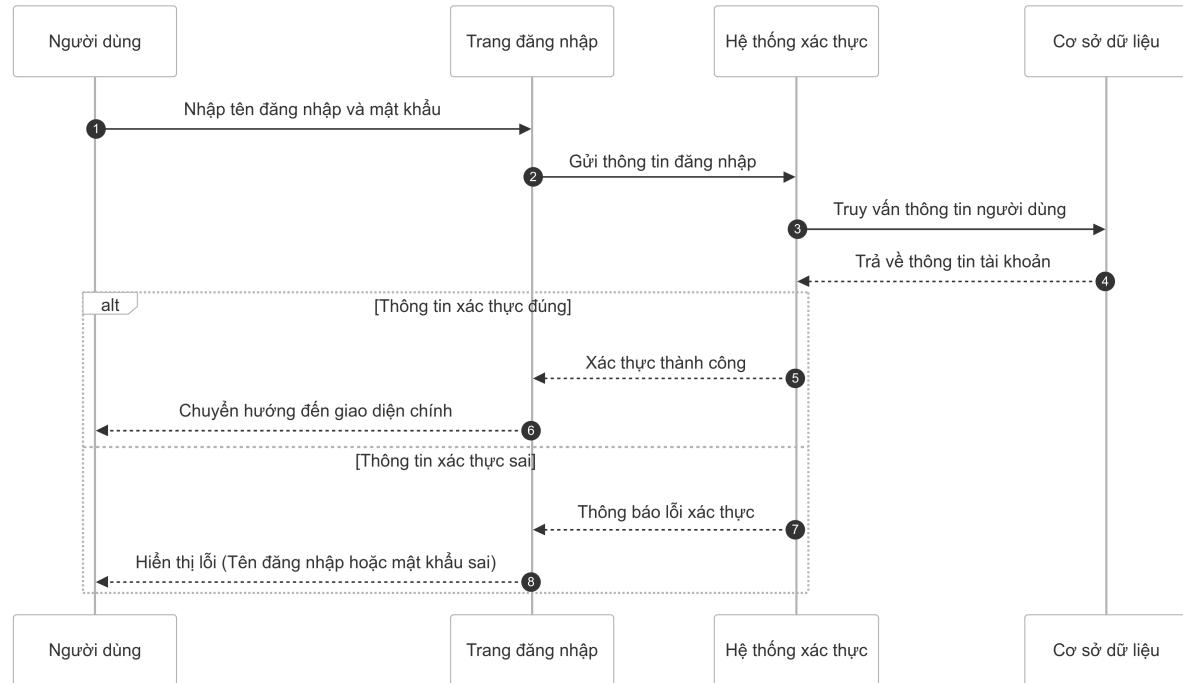


Hình 29: Sơ đồ hoạt động cho chức năng đổi mật khẩu

3.5.2 Sơ đồ tuần tư

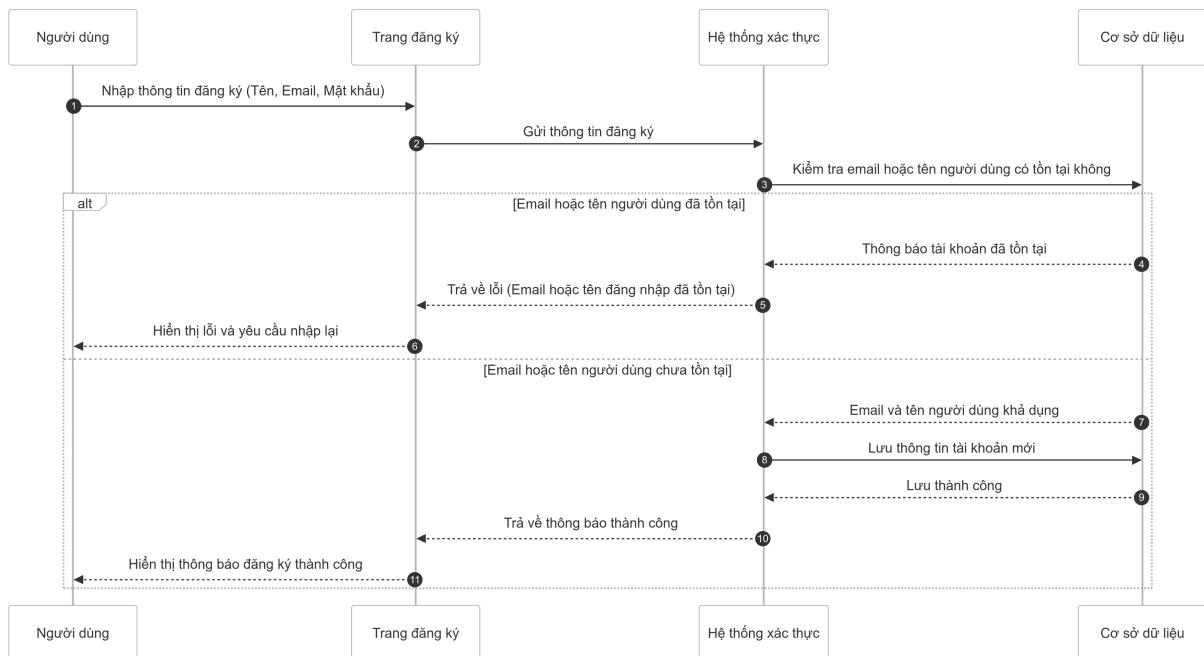
Một số sơ đồ tuần tư chính như sau:

- Sơ đồ tuần tự cho trang đăng nhập:



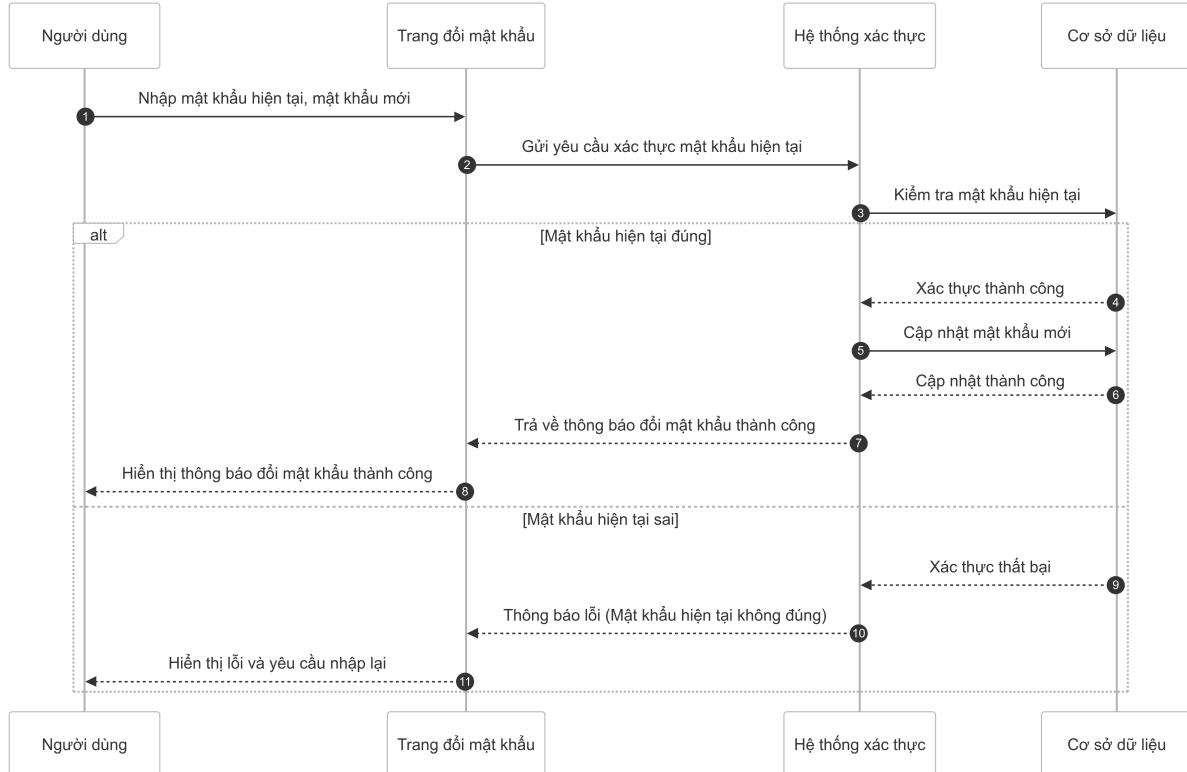
Hình 30: Sơ đồ tuần tự cho trang đăng nhập

- Sơ đồ tuần tự cho trang đăng ký:



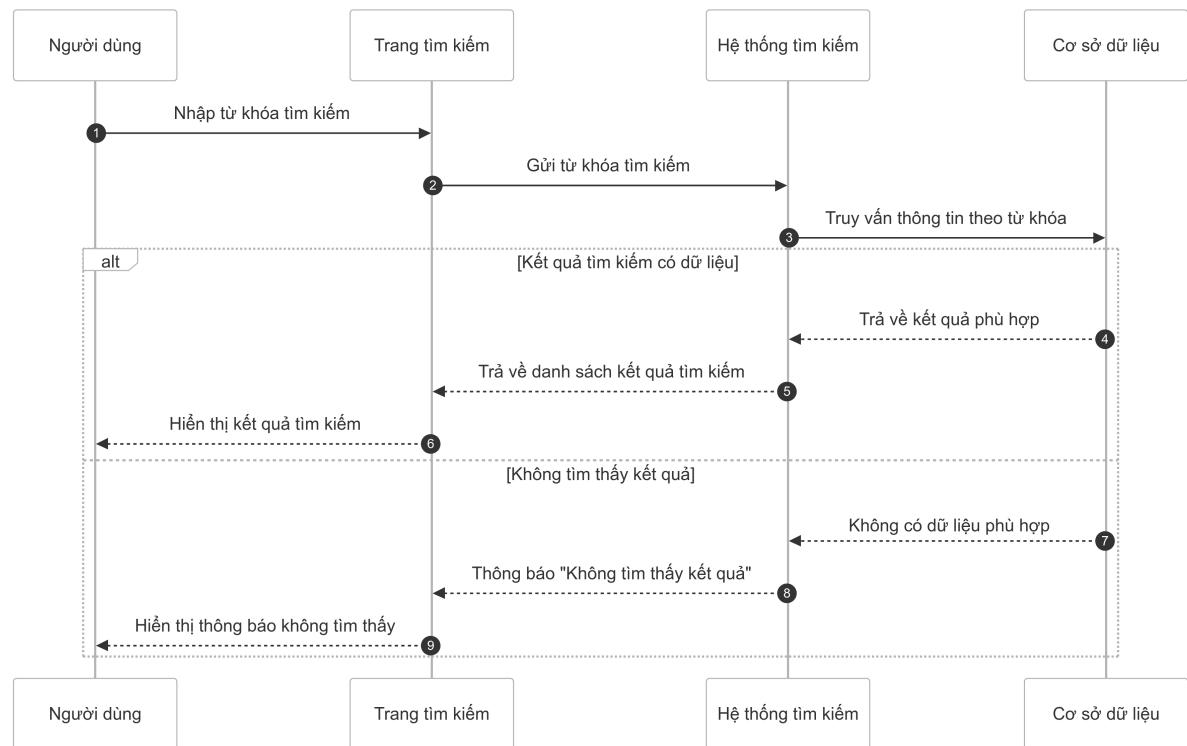
Hình 31: Sơ đồ tuần tự cho trang đăng ký

- Sơ đồ tuần tự đổi mật khẩu:



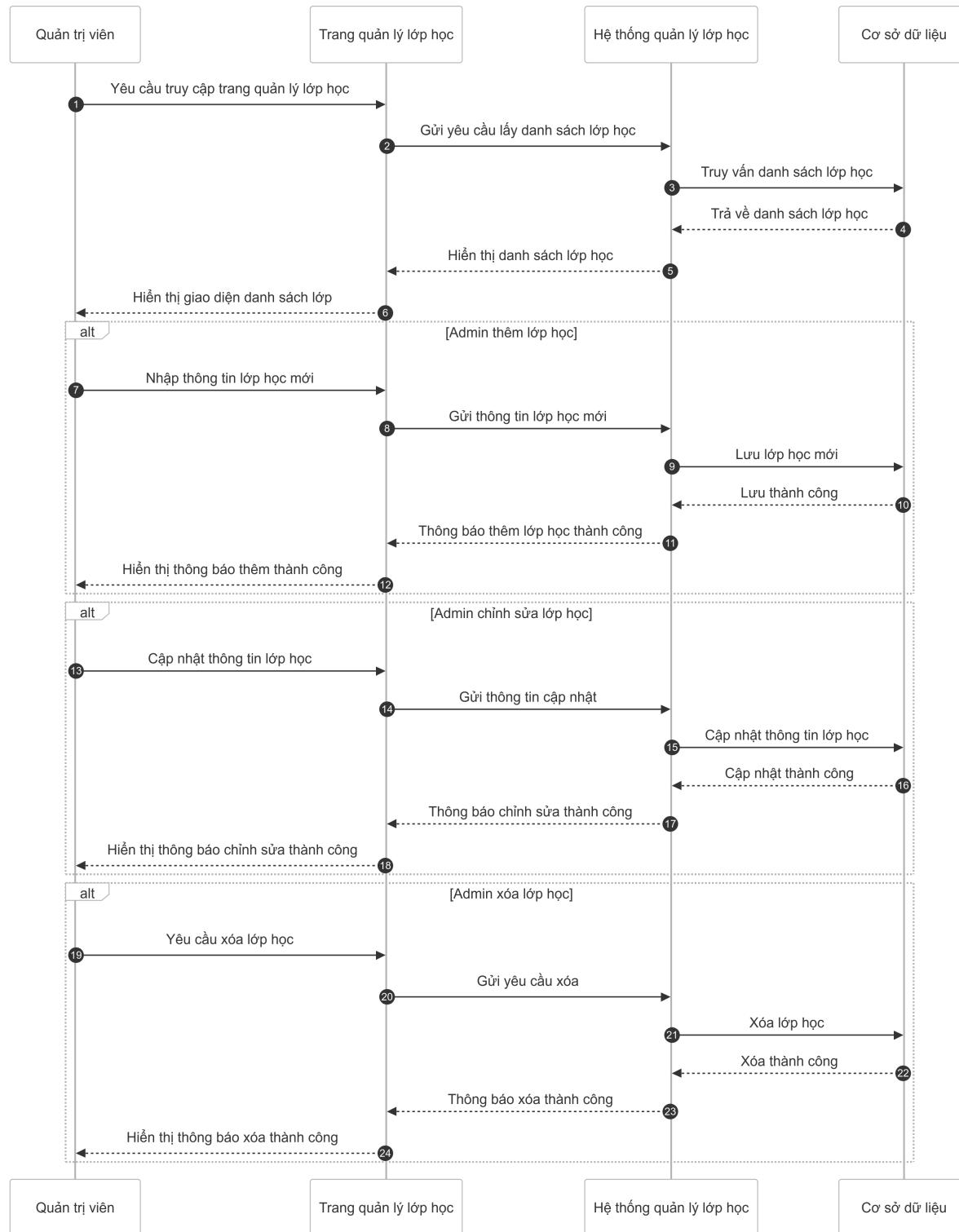
Hình 32: Sơ đồ tuần tự đổi mật khẩu

- Sơ đồ tuần tự tìm kiếm:



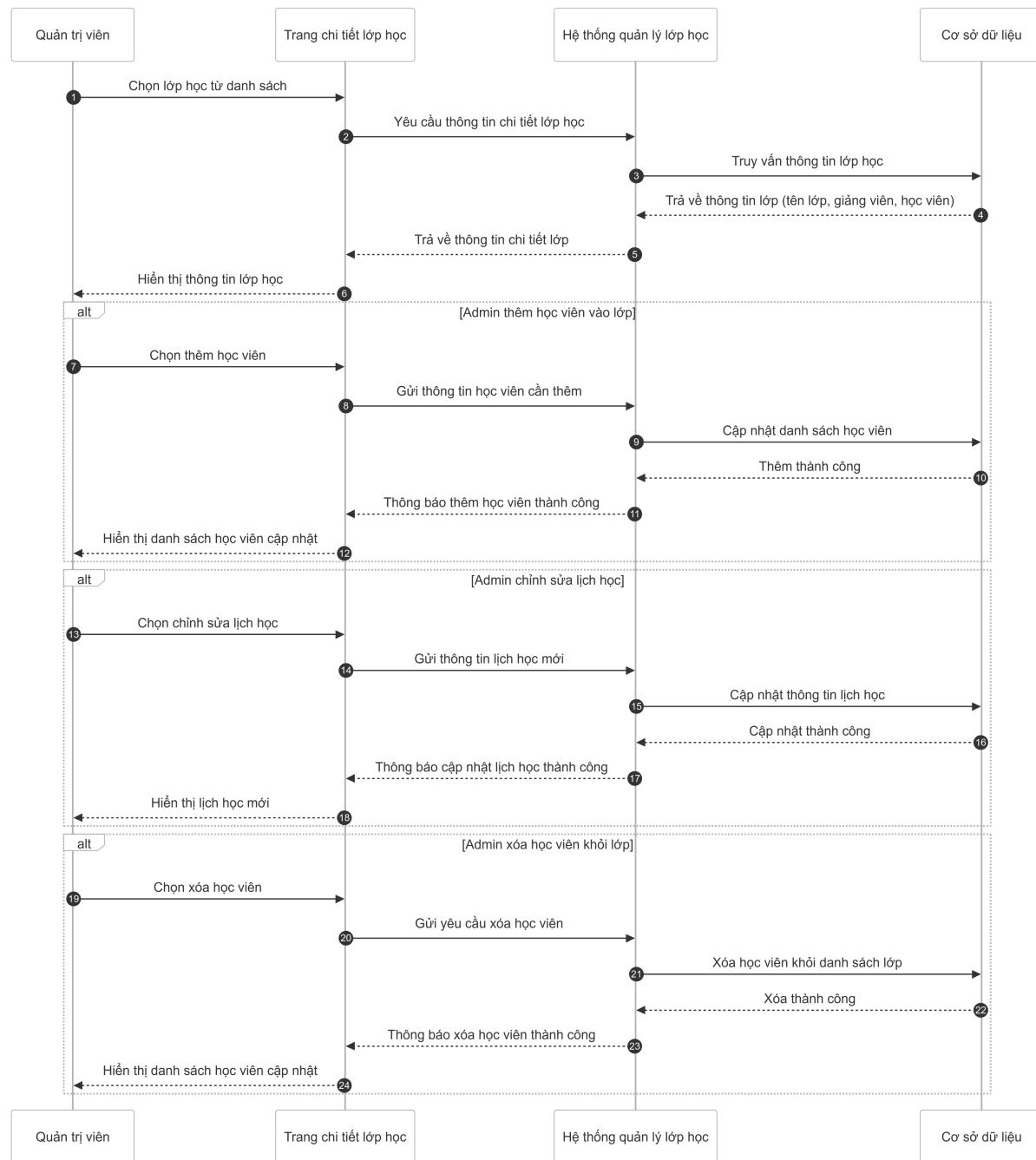
Hình 33: Sơ đồ tuần tự tìm kiếm

- Sơ đồ tuần tự quản lý lớp học:



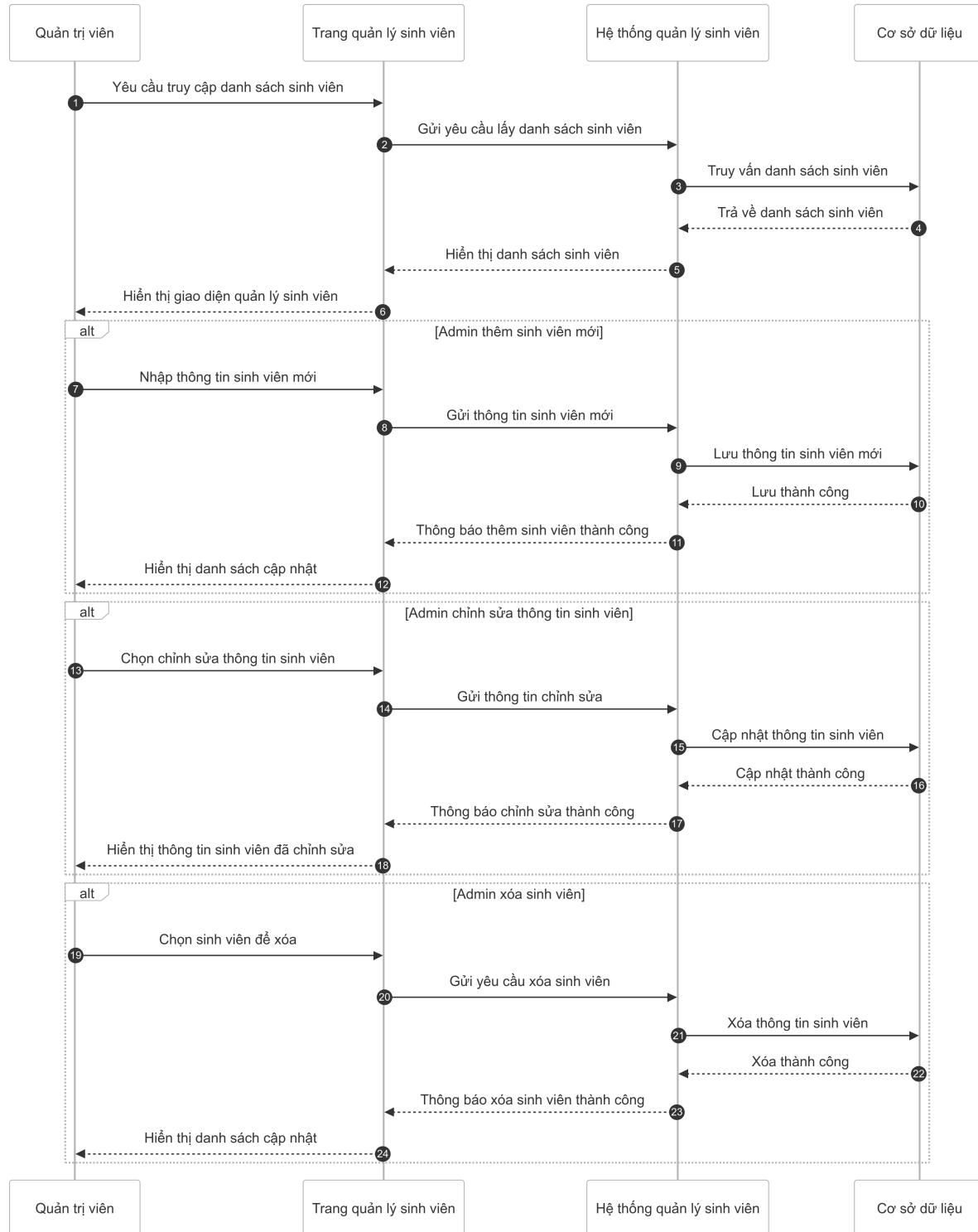
Hình 34: Sơ đồ tuần tự quản lý lớp học

- Sơ đồ tuần tự chi tiết lớp học:



Hình 35: Sơ đồ tuần tự chi tiết lớp học

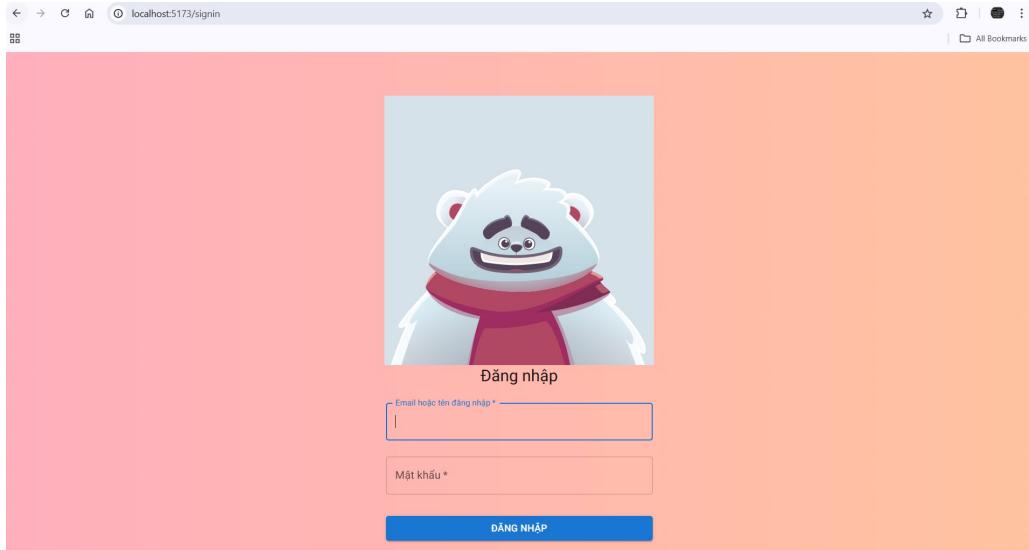
- Sơ đồ tuần tự quản lý sinh viên:



Hình 36: Sơ đồ tuần tự quản lý sinh viên

3.6 Thiết kế giao diện người dùng

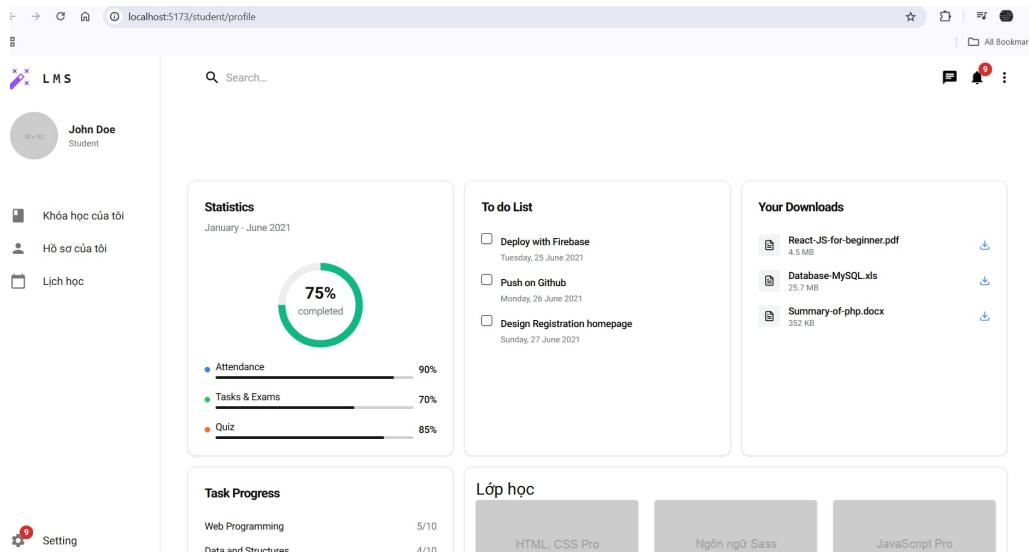
3.6.1 Trang đăng nhập



Hình 37: hình ảnh Trang Đăng nhập

- **Đặc tả giao diện trang đăng nhập:** Giao diện là trang đăng nhập (Sign-in) của một hệ thống, với thiết kế đơn giản, tập trung vào trải nghiệm người dùng.
- **Giao diện bao gồm:**
 - + Một hình minh họa trung tâm.
 - + Hai trường nhập liệu: email/tên đăng nhập và mật khẩu.
 - + Một nút bấm ĐĂNG NHẬP.
- **Chức năng của các trường nhập liệu:** Ô nhập tài khoản, ô mật khẩu: nhập tài khoản/mật khẩu do trung tâm cấp khi đăng ký khóa học sau khi liên hệ trung tâm.
- **Nút đăng nhập:** Đăng nhập vào website dựa vào tài khoản chia vai trò.

3.6.2 Trang student/profile



Hình 38: Giao diện trang student

- **Đặc tả giao diện trang student:** Giao diện Student Profile là trang cá nhân dành cho học viên trong hệ thống quản lý học tập. Trang này hiển thị các thông tin quan trọng liên quan đến học viên, bao gồm: tiến trình học tập, danh sách công việc cần làm, tài liệu tải xuống, và danh sách lớp học.

- **Các thành phần giao diện:**

+ **Logo:**

- Hiển thị logo với thiết kế đơn giản ở góc trên cùng bên trái.

+ **Thông tin học viên bao gồm:**

· **Tên học viên.**

· **Vai trò:** "Student".

· **Ảnh đại diện.**

+ **Danh sách menu:**

· **Khóa học của tôi:** Chuyển đến danh sách các khóa học của học viên.

· **Hồ sơ của tôi:** Trang hiển thị thông tin cá nhân của học viên (trang hiện tại).

· **Lịch học:** Xem lịch học hoặc các sự kiện liên quan.

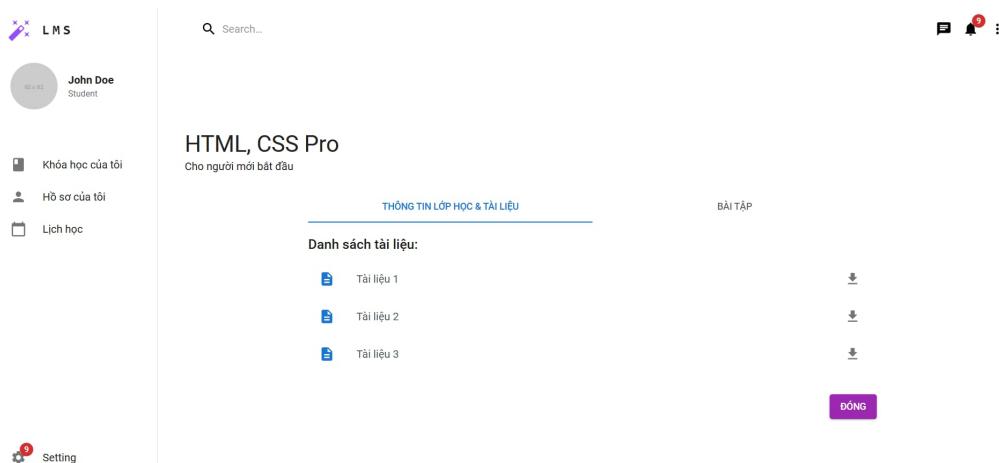
+ **Setting (Cài đặt):** Biểu tượng bánh răng để truy cập trang cài đặt tài khoản.

+ **Phần tìm kiếm (Search):**

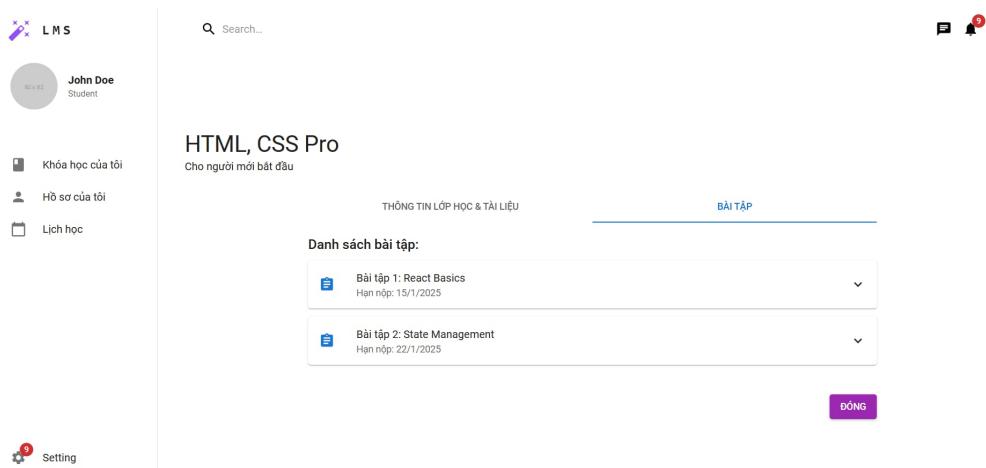
· **Thanh tìm kiếm:** Hiển thị ở giữa phía trên giao diện, cho phép tìm kiếm thông tin trong hệ thống (ví dụ: khóa học, tài liệu).

+ **Khu vực thông báo và cấu hình tài khoản:**

- **Biểu tượng thông báo:** Nút hình chuông ở góc trên bên phải, hiển thị số lượng thông báo (9 thông báo chưa đọc).
 - **Cài đặt khác:** Menu ba chấm ở góc trên bên phải để truy cập các tùy chọn nâng cao hoặc thoát tài khoản.
- + **Chức năng chi tiết bao gồm:**
- **Chức năng sidebar:** Chuyển đổi giữa các trang "Khóa học của tôi", "Hồ sơ của tôi", và "Lịch học".
 - **Tìm kiếm:** Cho phép tìm kiếm các thông tin như tài liệu, lớp học, hoặc công việc cần làm.
- + **Khu vực các mục lớn:**
- **Thống kê tiến độ (Statistics):** Hiển thị trạng thái tiến độ học tập chung và từng hạng mục chi tiết (attendance, quiz, tasks).
 - **Quản lý công việc (To-Do List):** Hiển thị các công việc cần làm kèm theo thời hạn. Người dùng có thể đánh dấu hoàn thành công việc bằng cách chọn checkbox.
 - **Tải xuống tài liệu (Your Downloads):** Hiển thị danh sách tài liệu tải xuống với định dạng và kích thước rõ ràng. Người dùng tải xuống bằng cách nhấn vào biểu tượng tải ở mỗi tài liệu.
 - **Tiến độ nhiệm vụ (Task Progress):** Hiển thị danh sách các nhiệm vụ đã hoàn thành và số lượng còn lại.
 - **Danh sách lớp học (Classes):** Liệt kê các lớp học mà học viên đã tham gia, kèm thông tin cơ bản.
- Sau khi truy cập vào giao diện của mình, học sinh có thể truy cập sang tiếp các giao diện như danh sách tài liệu, danh sách bài tập.



Hình 39: Giao diện danh sách tài liệu



Hình 40: Giao diện danh sách bài tập

3.6.3 Trang lịch học



Hình 41: Giao diện trang lịch học

- **Đặc tả giao diện trang lịch học:** Giao diện lịch học là trang hiển thị lịch trình, các sự kiện quan trọng, và các hạn chót (ví dụ: nộp bài tập, nghỉ lễ). Giao diện sử dụng bố cục dạng lịch tháng, giúp người dùng dễ dàng theo dõi thời gian và kế hoạch.
- **Chức năng chi tiết bao gồm:**
 - + **Chuyển đổi giữa các chế độ hiển thị:** Người dùng có thể chuyển đổi giữa các chế độ "Tháng", "Tuần", "Ngày", và "Lịch trình" để xem chi tiết theo nhu cầu.
 - + **Chế độ mặc định:** Hiển thị toàn bộ lịch tháng.
 - + **Xem chi tiết sự kiện:** Khi người dùng nhấp vào một sự kiện (ví dụ: "Hạn nộp bài tập"), giao diện mở ra cửa sổ hoặc chuyển đến trang chi tiết sự kiện.
 - Chi tiết có thể bao gồm: Tên sự kiện, Ngày giờ.
 - Mô tả hoặc thông tin bổ sung (ví dụ: tài liệu liên quan đến bài tập).

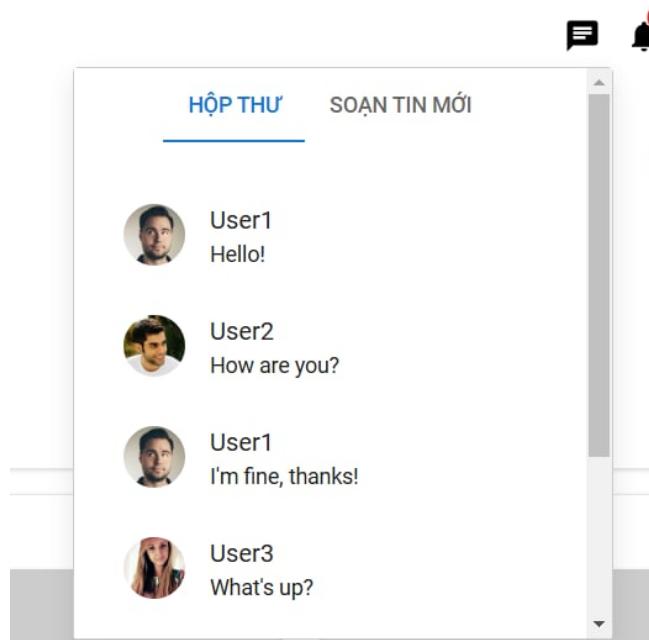
- **Điều hướng lịch:**

- + **Hôm nay:** Quay lại ngày hiện tại trong lịch.
- + **Trước và Sau:** Dùng để duyệt qua các tháng khác.
- **Tích hợp sự kiện học tập:** Sự kiện liên quan đến học tập (ví dụ: hạn nộp bài, lịch học) được tự động đồng bộ từ hệ thống.

3.6.4 Trang hộp thoại và thông báo

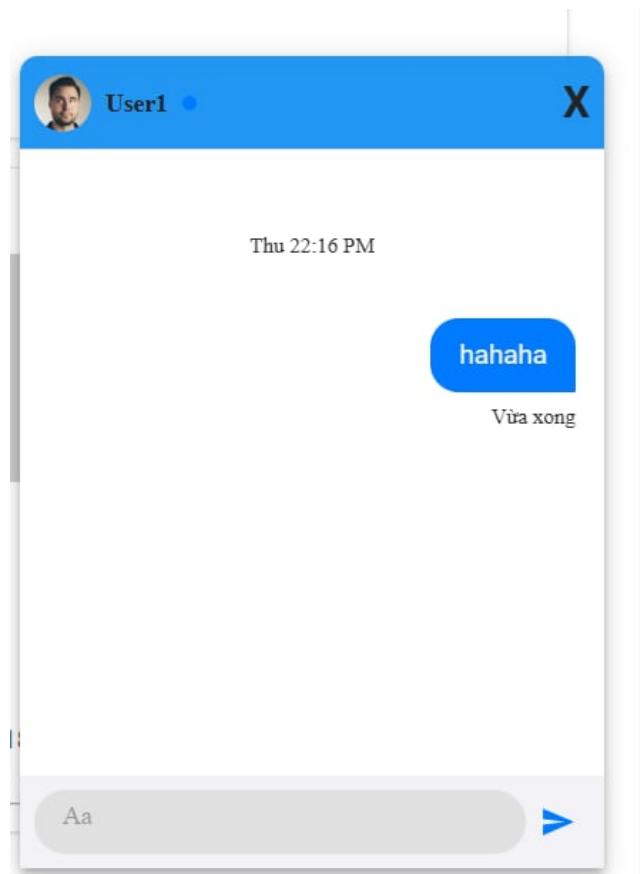
Hộp thoại là cửa sổ nhỏ xuất hiện trên giao diện chính để hiển thị các thông tin hoặc yêu cầu người dùng thực hiện một hành động cụ thể. Hộp thoại thường được sử dụng để xác nhận, nhập thông tin hoặc thông báo lỗi.

- **Giao diện trang hộp thoại:**



Hình 42: Trang Hộp Thoại ảnh 1

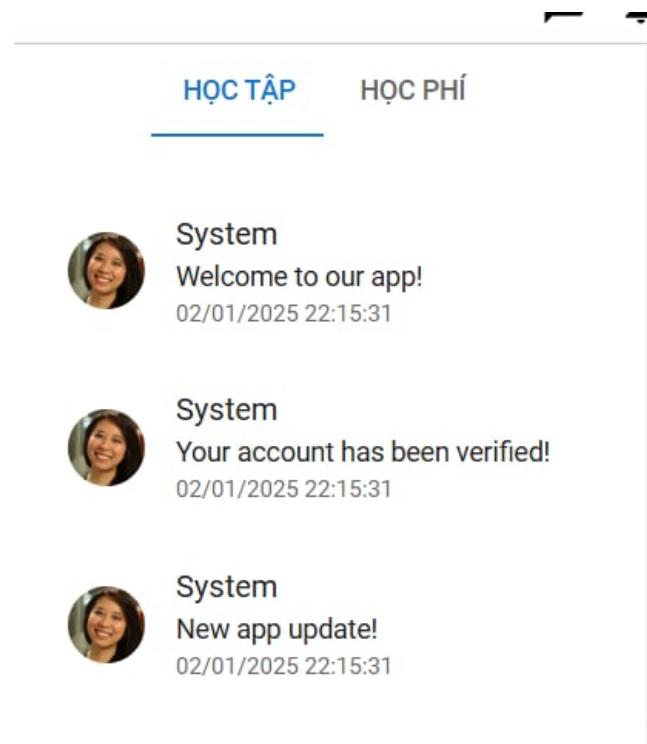
- Giao diện sau khi bấm vào hộp thoại:



Hình 43: Hộp thoại sau khi mở

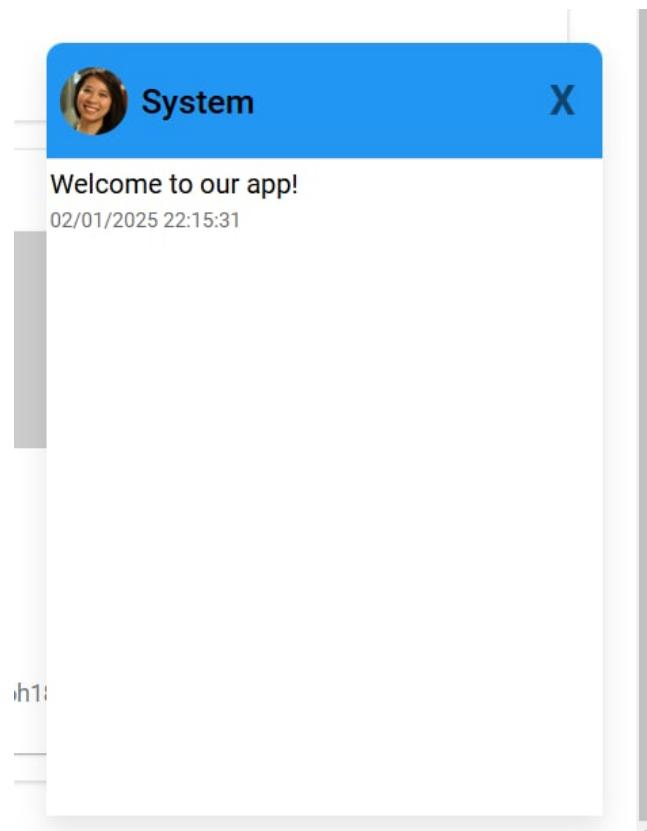
- **Trang thông báo:** Hộp thông báo là thành phần giao diện dùng để hiển thị các thông tin ngắn gọn và tức thời cho người dùng. Nó thường xuất hiện ở góc màn hình hoặc trên đầu trang, không làm gián đoạn hoạt động của người dùng.

- Giao diện bên ngoài của trang thông báo:



Hình 44: Giao diện bên ngoài của trang thông báo

- Giao diện bên ngoài của trang thông báo:



Hình 45: Giao diện bên trong của hộp thông báo

3.6.5 Giao diện quản lý học sinh

The screenshot shows a web-based LMS interface for managing students. On the left is a sidebar with a user profile for 'John Doe' (Admin) and links for Trang chính, Quản lý học viên, Quản lý giáo viên, Quản lý lớp học, Quản lý thu chi, Hóa đơn, Học phí, and Ý kiến phản hồi. Below this is a 'Setting' section with a red notification badge. The main area is titled 'Danh sách học viên' and contains a table with student information. Each row includes a 'Chỉnh sửa' (Edit) button. The table columns are: ID, Họ Tên (Name), Giới Tính (Gender), Ngày Sinh (Date of Birth), Lớp (Class), Trường Học (School), Lớp học thêm (Additional Class), and Chỉnh Sửa (Edit). The data in the table is as follows:

ID	Họ Tên 0901234567	Giới Tính	Ngày Sinh	Lớp	Trường Hoc	Lớp học thêm	Chỉnh Sửa
1	Nguyễn Văn A 0901234567	Nam	2000-01-15	10	THPT Nguyễn Du	Math 101 Physics 201	Chỉnh sửa
2	Trần Thị B 0909876543	Nữ	2001-05-20	11	THPT Lê Hồng Phong	Chemistry 301 English 101	Chỉnh sửa
3	Lê Văn C 0912345678	Nam	1999-11-10	12	THPT Trần Phú	History 201 Art 101	Chỉnh sửa
4	Phạm Thị D 0987654321	Nữ	2002-03-08	9	THPT Nguyễn Thị Minh	Math 101 English 101	Chỉnh sửa

Below the table are buttons for 'Tim kiếm...' (Search...) and '+ Thêm học viên' (Add student). At the bottom right are buttons for 'Rows per page' (10), page numbers (1, 2), and a refresh icon.

Hình 46: Giao diện quản lý học sinh

- **Đặc tả giao diện quản lý học sinh:** Giao diện quản lý học viên là trang giúp admin quản lý thông tin học viên, bao gồm danh sách học viên, thông tin cá nhân, lớp học và các chức năng chỉnh sửa dữ liệu.

- **Thành phần giao diện gồm:**

- Sidebar điều hướng:

- + **Vị trí:** Bên trái màn hình.

- + **Chức năng:** Cho phép admin chuyển đổi giữa các tính năng khác nhau trong hệ thống.

- + **Các mục:**

- **Trang chính:** Quay về trang tổng quan.
- **Quản lý học viên (đang được chọn):** Hiển thị danh sách học viên.
- **Quản lý giáo viên:** Quản lý danh sách giảng viên.
- **Quản lý lớp học:** Xem và chỉnh sửa danh sách các lớp học.
- **Quản lý thu chi:** Xem và quản lý thu nhập/chi phí.
- **Hóa đơn:** Quản lý hóa đơn liên quan đến học phí.
- **Ý kiến phản hồi:** Xem phản hồi từ học viên hoặc giảng viên.
- **Cài đặt (Setting):** Thay đổi cài đặt hệ thống.

- Thanh tìm kiếm:

- + **Vị trí:** Trên cùng của bảng danh sách học viên.

- + **Chức năng:**

- Nhập từ khóa (tên, ID hoặc thông tin liên quan) để tìm kiếm học viên.

- Tự động lọc danh sách khi nhập từ khóa.

- **Danh sách học viên:**

- + **Vị trí:** Trung tâm màn hình.
- + **Chức năng:** Hiển thị danh sách chi tiết của các học viên dưới dạng bảng.

3.6.6 Giao diện quản lý giáo viên

ID	Họ Tên	Giới Tính	Ngày Sinh	Lớp	Trường Học	Lớp đang học	Lương/tháng	
1	Nguyen Van A 0912345678	Nam	1990-05-10	Toán	Đại học Sư Phạm	10A1 10A2	\$100	Chỉnh sửa
2	Tran Thi B 0987654321	Nữ	1992-02-15	Văn	Đại học Khoa Học Xã Hội và Nhân Vân	11A1 11A2 12A3	\$120	Chỉnh sửa

1-8 of 8

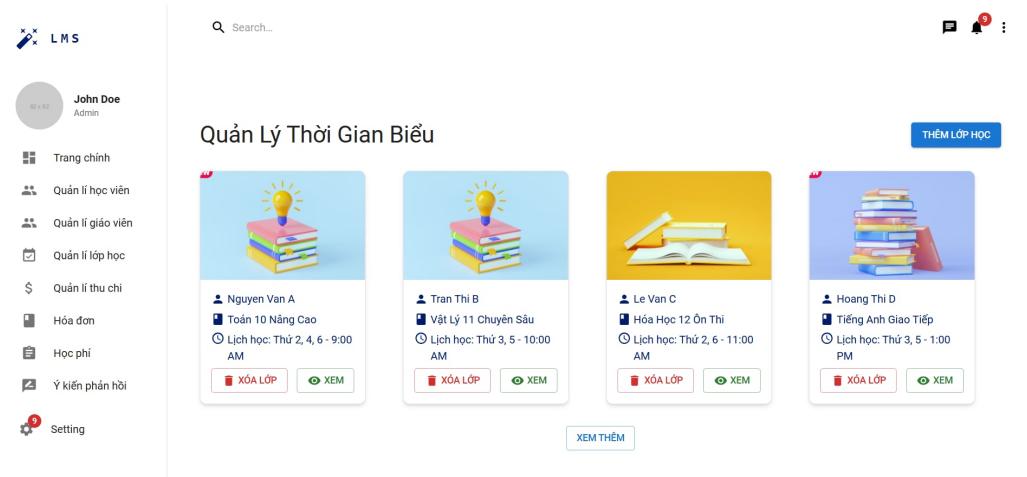
Rows per page: 10 | 1

Hình 47: Giao diện quản lý giáo viên

- **Đặc tả giao diện quản lý giáo viên:** Giao diện quản lý giáo viên là trang hỗ trợ admin trong việc quản lý danh sách giáo viên, thông tin cá nhân, lớp học đảm nhận, và các chức năng chỉnh sửa dữ liệu liên quan.
- **Mục đích:** Cung cấp thông tin chi tiết về giáo viên, bao gồm họ tên, chuyên ngành, lớp đang dạy và mức lương, đồng thời hỗ trợ các chức năng thêm mới, chỉnh sửa, và tìm kiếm giáo viên.
- **Thành phần chính:**
 - + Sidebar điều hướng.
 - + Thanh tìm kiếm.
 - + Bảng danh sách giáo viên.
 - + Nút thêm giáo viên mới.
- **Các tính năng chính bao gồm:**
 - + **Hiển thị danh sách giáo viên:** Danh sách được trình bày rõ ràng với đầy đủ thông tin cần thiết.
 - + **Tìm kiếm giáo viên:** Lọc danh sách theo các tiêu chí như họ tên, lớp đang dạy, hoặc trường học.
 - + **Thêm giáo viên mới:** Nhập thông tin chi tiết để thêm giáo viên vào hệ thống.

- + **Chỉnh sửa thông tin giáo viên:** Cho phép admin cập nhật thông tin như họ tên, ngày sinh, lớp đang dạy, hoặc mức lương.
- + **Phân trang:** Hỗ trợ admin quản lý danh sách giáo viên dài bằng cách chia nhỏ danh sách thành các trang.

3.6.7 Giao diện quản lý lớp học



Hình 48: Giao diện quản lý lớp học

- **Đặc tả giao diện quản lý lớp học:** Giao diện quản lý lớp học cung cấp công cụ trực quan cho admin để theo dõi và quản lý danh sách các lớp học, thông tin học viên, lịch học, và các thao tác liên quan đến từng lớp học.
- **Mục đích:** Cung cấp thông tin chi tiết về các lớp học, bao gồm tên lớp, giảng viên, lịch học, và các hành động như chỉnh sửa hoặc xóa lớp.

- Thành phần chính:

- + Sidebar điều hướng.
- + Thanh tìm kiếm.
- + Danh sách các lớp học hiển thị dưới dạng card (thẻ).
- + Nút thêm lớp học mới.

- Tính năng chính:

- + **Hiển thị danh sách lớp học:** Danh sách các lớp học được trình bày dưới dạng card, dễ nhìn và dễ truy cập thông tin.
- + **Tìm kiếm lớp học:** Tìm kiếm theo từ khóa liên quan đến tên học viên, tên lớp hoặc lịch học.
- + **Thêm lớp học mới:** Cho phép admin thêm thông tin lớp học như tên lớp, lịch học, giảng viên, và học viên tham gia.

- + **Xem chi tiết và chỉnh sửa lớp học:** Hiển thị thông tin chi tiết lớp học và cho phép chỉnh sửa các thông tin liên quan.
- + **Xóa lớp học:** Loại bỏ lớp học không còn sử dụng hoặc không cần thiết.
- + **Phân trang/xem thêm:** Hiển thị thêm lớp học khi cần thiết với nút "XEM THÊM".

4 Triển khai hệ thống

4.1 Lựa chọn công cụ và công nghệ

4.1.1 Frontend: HTML, CSS, JavaScript, ReactJS

Giao diện người dùng được xây dựng với các công nghệ sau:

- **HTML:** Tạo cấu trúc nội dung cho trang web.
- **CSS:** Thiết kế giao diện, định dạng bố cục, màu sắc, và phong cách hiển thị.
- **JavaScript:** Thêm các tương tác động, xử lý logic phía người dùng.
- **ReactJS:** Framework frontend hiện đại, mạnh mẽ, cho phép phát triển giao diện động, quản lý trạng thái hiệu quả, và tăng hiệu suất render.

4.1.2 Backend: JavaSpring, và một số công cụ phát triển Postman và Swagger UI

Phần backend của hệ thống chịu trách nhiệm xử lý logic nghiệp vụ, giao tiếp với cơ sở dữ liệu, và cung cấp các API cho frontend sử dụng.

- Java Spring:

- + Là một framework mạnh mẽ để xây dựng ứng dụng web và API backend.
- + Được sử dụng để xử lý các logic nghiệp vụ, quản lý cơ sở dữ liệu và cung cấp API cho frontend.
- + Ưu điểm của Java Spring:
 - Hỗ trợ cấu hình linh hoạt thông qua Dependency Injection (DI).
 - Bảo mật mạnh mẽ với Spring Security.
 - Khả năng mở rộng tốt, phù hợp với các dự án lớn.

- **Postman:** Công cụ kiểm thử API, giúp đảm bảo các API hoạt động đúng như mong đợi.
- **Swagger UI:** Hỗ trợ thiết kế và tài liệu hóa API, giúp việc giao tiếp giữa frontend và backend trở nên dễ dàng hơn.

Lợi ích tổng thể:

- **Tích hợp đồng bộ:** Java Spring đảm bảo backend mạnh mẽ, Postman hỗ trợ kiểm thử nhanh chóng, và Swagger UI tạo sự minh bạch trong giao tiếp API.
- **Hiệu quả phát triển:** Cải thiện hiệu suất làm việc giữa các nhóm phát triển thông qua các công cụ kiểm thử và tài liệu hóa API.
- **Tăng cường bảo mật và hiệu năng:** Spring Security cung cấp các cơ chế bảo mật tiêu chuẩn, đảm bảo dữ liệu và hệ thống được bảo vệ tốt nhất.

4.1.3 Cơ sở dữ liệu: SQL SERVER

SQL Server được sử dụng để quản lý và lưu trữ toàn bộ dữ liệu của hệ thống. Các ưu điểm bao gồm:

- Hỗ trợ lưu trữ dữ liệu lớn với độ tin cậy cao.
- Hỗ trợ các truy vấn phức tạp để xử lý thông tin nhanh chóng.
- Đảm bảo an toàn dữ liệu và tích hợp tốt với các nền tảng hiện đại.

4.2 Phát triển và tích hợp các module

4.2.1 Quản lý học viên

Module quản lý học viên chịu trách nhiệm xử lý thông tin học viên:

- Tạo, sửa, xóa hồ sơ học viên (họ tên, ngày sinh, lớp học tham gia).
- Theo dõi tiến độ học tập, điểm danh, và bài tập.
- Kết nối với các module khác như quản lý lớp học và lịch học để cung cấp thông tin tổng quát.

4.2.2 Quản lý giảng viên

Module này quản lý thông tin giảng viên và công việc giảng dạy của họ:

- Lưu trữ thông tin cá nhân (tên, trường công tác, môn học phụ trách).
- Quản lý thời khóa biểu giảng dạy, danh sách lớp, và học viên.
- Hỗ trợ gửi thông báo, tài liệu, hoặc bài tập cho học viên thông qua hệ thống.

4.2.3 Quản lý lớp học

Module quản lý các lớp học trong hệ thống:

- Tạo mới, chỉnh sửa thông tin lớp học (giảng viên, số lượng học viên, lịch học).
- Theo dõi danh sách học viên tham gia, trạng thái học tập, và kết quả.
- Hỗ trợ quản lý mối quan hệ giữa học viên và giảng viên trong từng lớp học.

4.3 Kiểm thử hệ thống

4.3.1 Kiểm thử chức năng

Kiểm thử chức năng nhằm đảm bảo các module hoạt động chính xác theo yêu cầu:

- Thủ nghiệm từng module (quản lý học viên, giảng viên, lớp học, khóa học) để đảm bảo chúng thực hiện đúng nhiệm vụ.
- Kiểm tra các API do backend cung cấp thông qua công cụ như Postman:

The screenshot shows the Postman interface with a successful POST request to the '/api/auth/login' endpoint. The request body contains the following JSON:

```

1 [
2   "id": "240000009",
3   "password": "12345678"
4 ]
5
6 // {
7 //   "id":240000009,
8 //   "username": "newUser", "email": "newemail@example.com", "phone": "8987654321"
9 // }
10
11
    
```

The response status is 200 OK, and the response body is:

```

1 {
2   "message": "Đăng nhập thành công.",
3   "role": "Admin",
4   "id": "240000009",
5   "token": "eyJhbGciOiJIUzI1NiJ9eyJzdWIiOiJhZG1pbmFsInJvbui0i3BZG1pbmFsInIhdCf6HfzcNTg2MzgNyiwZXhwIjoxNzM100Y50013fQ.k0IBcs50LTIkwBaMaJkx0zSK0K-aZygmJATF9z2d79wHs1Lag#955f09E9b4387mnxgdlx2MBuE3iyk14b2Q",
6   "redirectTo": "/admin/dashboard"
7 }
    
```

Hình 49: Test API đăng nhập hệ thống bằng Postman

The screenshot shows the Postman interface with a successful GET request to the '/api/user/profile' endpoint. The request body contains the following JSON:

```

1 [
2   "id": "240000009",
3   "password": "12345678"
4 ]
5
6 // {
7 //   "id":240000009,
8 //   "username": "newUser", "email": "newemail@example.com", "phone": "8987654321"
9 // }
10
11
    
```

The response status is 200 OK, and the response body is:

```

1 {
2   "idUser": "240000009",
3   "roleName": "Admin",
4   "userName": "admin",
5   "password": "$argon2id$v=19$m=6533,t=10,p=1$xielTffMzYb26RTEWNeROA$ejjjYXtfUcuzuooBqc3taS8M+bC8JMRlEk",
6   "email": "nguyenthamh51024@gmail.com",
7   "phone": "0328664697",
8   "createdYear": 2024
9 }
    
```

Hình 50: Test API lấy thông tin tài khoản bằng Postman

- Kiểm tra các API do backend cung cấp thông qua Swagger - UI:

The screenshot shows the Swagger UI for the '/api/auth/login' POST endpoint. The 'Parameters' section shows 'No parameters'. The 'Request body' section is set to 'application/json' and contains the following JSON:

```

{
  "password": "12345678",
  "id": "240000009"
}
    
```

At the bottom, there are 'Execute' and 'Clear' buttons.

Hình 51: Test API đăng nhập hệ thống bằng Swagger - UI

```

curl -X 'POST' \
  'http://localhost:8888/api/auth/login' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
    "password": "12345678",
    "id": "24000000"
  }'

```

Request URL
<http://localhost:8888/api/auth/login>

Server response

Code	Details
200	<p>Response body</p> <pre>{ "message": "Đăng nhập thành công.", "role": "Admin", "id": "24000000", "token": "eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJhZG1pbWlzbmVhbmVtYmVubGUiOiJBIGipbiIsImhlhCIGNTczNTgzaMsQ5HsW1ZXmTjoxNzE1M0Y5NDkxFQ.v3tAGSHIx5B-R5jDorDyLOnf1HteNfZ4vrQkGPFXh-WtZzIc_EHlmQuLSwC2m57pRijbenI3q5lk2zurI", "redirectUrl": "/admin/dashboard" }</pre> <p>Response headers</p> <pre> cache-control: no-cache,no-store,max-age=0,must-revalidate connection: keep-alive content-type: application/json date: Thu, 02 Jan 2025 15:58:11 GMT expires: 0 pragma: no-cache transfer-encoding: chunked vary: Origin,Access-Control-Request-Method,Access-Control-Request-Headers x-content-type-options: nosniff x-frame-options: DENY x-xss-protection: 0 </pre>

Hình 52: Test API lấy thông tin tài khoản bằng Swagger - UI

- Test api lấy thông tin tài khoản:

GET /api/user/search

GET /api/user/profile

Parameters

No parameters

Responses

Execute Clear

Curl

```

curl -X 'GET' \
  'http://localhost:8888/api/user/profile' \
  -H 'accept: */*'

```

Request URL
<http://localhost:8888/api/user/profile>

Server response

Code	Details
200	<p>Response body</p> <pre>{ "idUser": "24000000", "roleName": "Admin", "password": "argon2idv20m=65536,t=10,p=1\$cieTFMzYb26RTENNeRO4\$ujjY7XtfuCuuzuoBgC3TaS8H+8IQYSB+bC8JMRlfk", "email": "nghiahanh51824@gmail.com", "phone": "0328664697", "createdYear": 2024 }</pre> <p>Response headers</p> <pre> cache-control: no-cache,no-store,max-age=0,must-revalidate connection: keep-alive content-type: application/json date: Thu, 02 Jan 2025 15:59:44 GMT expires: 0 pragma: no-cache transfer-encoding: chunked vary: Origin,Access-Control-Request-Method,Access-Control-Request-Headers x-content-type-options: nosniff x-frame-options: DENY x-xss-protection: 0 </pre>

Hình 53: API lấy thông tin tài khoản

Curl

```

curl -X 'GET' \
  'http://localhost:8888/api/user/profile' \
  -H 'accept: */*'

```

Request URL
<http://localhost:8888/api/user/profile>

Server response

Code	Details
200	<p>Response body</p> <pre>{ "idUser": "24000000", "roleName": "Admin", "password": "argon2idv20m=65536,t=10,p=1\$cieTFMzYb26RTENNeRO4\$ujjY7XtfuCuuzuoBgC3TaS8H+8IQYSB+bC8JMRlfk", "email": "nghiahanh51824@gmail.com", "phone": "0328664697", "createdYear": 2024 }</pre> <p>Response headers</p> <pre> cache-control: no-cache,no-store,max-age=0,must-revalidate connection: keep-alive content-type: application/json date: Thu, 02 Jan 2025 15:59:44 GMT expires: 0 pragma: no-cache transfer-encoding: chunked vary: Origin,Access-Control-Request-Method,Access-Control-Request-Headers x-content-type-options: nosniff x-frame-options: DENY x-xss-protection: 0 </pre>

Hình 54: API lấy thông tin tài khoản

- Đảm bảo giao diện và logic xử lý trên frontend và backend tương thích với nhau.

4.3.2 Kiểm thử hiệu năng

Kiểm thử hiệu năng đánh giá khả năng hoạt động của hệ thống dưới các điều kiện tải khác nhau:

- Thử nghiệm hệ thống với nhiều người dùng truy cập cùng lúc để đảm bảo không bị chậm hoặc lỗi.
- Dánh giá thời gian phản hồi của API và cơ sở dữ liệu.
- Tối ưu hóa các truy vấn SQL và logic xử lý của backend để đạt hiệu suất cao.

5 Kết quả và đánh giá

Vì thời gian không cho phép và kiến thức về lập trình web còn hạn chế nên chúng em chưa tìm hiểu rõ và kỹ hơn về đề tài này. Do đó đề tài vẫn còn có nhiều thiếu sót, mong thầy thông cảm và giúp đỡ chúng em nhiều hơn, chúng em sẽ tiếp tục tìm hiểu sâu hơn về những công nghệ này, nhằm mục đích lập trình web:

- Giao diện đẹp và thân thiện hơn với người dùng.
- Thiết kế và lập trình bổ sung thêm các chức năng còn thiếu hay chưa hoàn chỉnh trong quá trình sử dụng.

5.1 Kết quả triển khai

5.1.1 Kết quả đạt được:

- Tạo thành công website quản lý trung tâm dạy học cho các trung tâm vừa và nhỏ.
- Ứng dụng được thêm các chức năng bao gồm: Quản lý thu chi trung tâm, trả lương giáo viên, thu học phí học sinh, reset mật khẩu nếu người dùng quên, điểm bài tập của học sinh, chuyển lớp nếu cần thiết.
- Tìm hiểu và nắm cơ bản được những công cụ để lập trình website như: các ngôn ngữ lập trình Website Java Spring, Javascript, CSS.
- Ứng dụng Web socket trong tin nhắn real time.
- Giao diện website được thiết kế đơn giản, thân thiện và dễ sử dụng.

5.1.2 Việc chưa làm được:

- Việc biểu diễn các thông tin trên website chưa được linh hoạt.
- Phân tích thiết kế chưa được hoàn chỉnh.
- Nhiều phần web thiết kế chưa đủ, cần nhiều thời gian để rèn luyện thêm.
- Nhiều chức năng còn thiếu và chưa sử dụng được.

6 Kết luận và hướng phát triển

6.1 Kết luận

- Hệ thống quản lý trung tâm dạy học đã đáp ứng tốt nhu cầu quản lý học viên, giảng viên và lịch học một cách trực quan và hiệu quả.
- Giao diện thân thiện, dễ sử dụng cho cả người quản trị và người dùng.
- Hệ thống vận hành ổn định và xử lý tốt các nghiệp vụ như đăng nhập, quản lý lớp học, và lưu trữ dữ liệu.
- Những hạn chế còn tồn tại:
 - + Một số chức năng chưa được hoàn thiện hoặc cần tối ưu hơn (ví dụ: báo cáo chi tiết, giao tiếp giữa các module).
 - + Các phần liên quan đến giao diện hoặc xử lý dữ liệu lớn có thể cần cải thiện.

6.2 Hướng phát triển tương lai

- Mở rộng phạm vi sử dụng:
 - + Dưa hệ thống vào ứng dụng trên các thiết bị di động với giao diện thân thiện hơn.
 - + Hỗ trợ đa ngôn ngữ để tiếp cận nhiều người dùng hơn ở các khu vực khác nhau.
- Tối ưu hiệu năng:
 - + Cải thiện tốc độ truy vấn và xử lý dữ liệu, đặc biệt với cơ sở dữ liệu lớn.
 - + Tối ưu hóa giao diện để đảm bảo hiệu suất trên các thiết bị có cấu hình thấp.

7 Tài liệu tham khảo

1. MDN Web Docs: Cung cấp tài liệu về HTML, CSS, JavaScript và các công cụ web khác.
URL: <https://developer.mozilla.org/>
2. W3Schools :Tài liệu học lập trình web cơ bản, bao gồm SQL, PHP, và JavaScript.
URL: <https://www.w3schools.com/>
3. Stack Overflow: Nền tảng hỏi đáp về lập trình với nhiều ví dụ thực tế.
URL: <https://stackoverflow.com/>
4. SQL Server Official Documentation: Tài liệu chính thức về SQL Server, bao gồm thiết kế, tối ưu hóa và quản trị cơ sở dữ liệu.
URL: <https://learn.microsoft.com/en-us/sql/>
5. Node.js Documentation: Hướng dẫn sử dụng Node.js để xây dựng ứng dụng backend hiệu quả.
URL: <https://nodejs.org/en/docs/>
6. ReactJS Documentation: Tài liệu chính thức giúp bắt đầu với ReactJS và xây dựng giao diện người dùng.
URL: <https://reactjs.org/docs/getting-started.html>
7. Postman Learning Center: Hướng dẫn sử dụng Postman để kiểm thử API.
URL: <https://learning.postman.com/>
8. Swagger Documentation: Hỗ trợ thiết kế, tài liệu hóa, và kiểm thử API RESTful.
URL: <https://swagger.io/docs/>

8 Phụ lục

8.1 Mã nguồn chính

Mã nguồn chính của hệ thống quản lý trung tâm dạy học là bộ khung cốt lõi, đảm bảo toàn bộ chức năng của hệ thống hoạt động hiệu quả. Nó được phát triển với sự kết hợp giữa frontend, backend và cơ sở dữ liệu, được tổ chức khoa học để dễ dàng bảo trì và mở rộng.

8.1.1 Thành phần mã nguồn chính

1. Frontend

- Mô tả:
 - + Phát triển giao diện người dùng, nơi học viên và giảng viên tương tác với hệ thống.
 - + Sử dụng HTML, CSS, JavaScript kết hợp với ReactJS để tạo giao diện hiện đại, tối ưu hiệu suất.
- Chức năng chính:
 - + Hiển thị danh sách khóa học, lịch học, bài tập, và các thông báo.
 - + Quản lý tiến trình học tập (progress tracking).
 - + Giao tiếp với backend thông qua API RESTful.
- Thành phần nổi bật:
 - + Components tái sử dụng: Form đăng nhập, biểu đồ thống kê, danh sách khóa học.
 - + Routing: Điều hướng giữa các trang như "Hồ sơ", "Lịch học", và "Khóa học".
 - + State Management: Quản lý trạng thái ứng dụng (Redux hoặc Context API).

2. Backend

- Mô tả:
 - + Phát triển các logic nghiệp vụ, xử lý dữ liệu, và bảo mật.
 - + Sử dụng Java Spring kết hợp với các công cụ như Postman (kiểm thử API) và Swagger UI (tài liệu hóa API).
- Chức năng chính:
 - + Xử lý logic nghiệp vụ:
 - Đăng nhập, đăng ký tài khoản.
 - Quản lý thông tin khóa học, học viên, bài tập.
 - Lập lịch học và thông báo sự kiện.

- + Bảo mật:
 - Sử dụng JWT để xác thực người dùng.
 - Mã hóa mật khẩu và bảo vệ API khỏi truy cập trái phép.
- Thành phần nổi bật:
 - + Controllers: Quản lý các endpoint API.
 - + Services: Xử lý các logic nghiệp vụ phức tạp.
 - + Repositories: Tương tác với cơ sở dữ liệu thông qua các truy vấn tối ưu.

3. Cơ sở dữ liệu

- Mô tả:
 - + Sử dụng SQL Server làm nơi lưu trữ toàn bộ dữ liệu của hệ thống, bao gồm thông tin tài khoản, khóa học, bài tập và lịch học.
- Chức năng chính:
 - + Lưu trữ thông tin học viên, giảng viên, khóa học, bài tập, và tiến trình học tập.
 - + Đảm bảo dữ liệu được truy vấn nhanh chóng và chính xác.
 - + Được sử dụng để tối ưu hóa các truy vấn dữ liệu phức tạp.

8.1.2 Cách thức hoạt động của mã nguồn

1. Frontend:

- Gửi yêu cầu đến API backend khi người dùng thực hiện thao tác, ví dụ: xem lịch học, nộp bài tập.
- Hiển thị dữ liệu phản hồi từ backend dưới dạng giao diện thân thiện.

2. Backend:

- Nhận yêu cầu từ frontend, xử lý logic và kiểm tra bảo mật.
- Tương tác với cơ sở dữ liệu để lấy hoặc cập nhật dữ liệu.
- Trả về kết quả hoặc thông báo lỗi cho frontend.

3. Cơ sở dữ liệu: Lưu trữ dữ liệu dài hạn và cung cấp thông tin theo yêu cầu của backend.

8.1.3 Điểm nổi bật

- **Tích hợp mạnh mẽ:**
 - + Swagger UI giúp tài liệu hóa và kiểm tra API dễ dàng.
 - + Postman hỗ trợ kiểm thử nhanh các endpoint trong quá trình phát triển.
- **Bảo mật cao:**

- + Sử dụng JWT cho việc xác thực và phân quyền.
- + Mã hóa mật khẩu và dữ liệu nhạy cảm trong cơ sở dữ liệu.

- **Hiệu suất tốt:**

- + Frontend tận dụng ReactJS để tối ưu hóa hiệu suất.
- + Backend sử dụng Java Spring với cấu trúc rõ ràng, đảm bảo tốc độ xử lý nhanh.
- + Cơ sở dữ liệu được tối ưu với indexing và stored procedures.

8.2 Các biểu đồ và hình ảnh bổ sung