

ĐẠI HỌC HUẾ

TRƯỜNG ĐẠI HỌC KHOA HỌC

**KHOA CÔNG NGHỆ THÔNG TIN**

-----🙞🙜🕮🙞🙜-----

**KHÓA LUẬN**

**TỐT NGHIỆP ĐẠI HỌC**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

**Đề tài:**

**XÂY DỰNG HỆ THỐNG**

**HỌC TẬP TRỰC TUYẾN**

**Sinh viên thực hiện:** **Phạm Văn Nhân**

**Khoá: K45 – Hệ chính quy**

**Giáo viên hướng dẫn: ThS. Hoàng Nguyễn Tuấn Minh**

**Huế, tháng 05 năm 2025**



ĐẠI HỌC HUẾ

TRƯỜNG ĐẠI HỌC KHOA HỌC

**KHOA CÔNG NGHỆ THÔNG TIN**

**🙞 🕮 🙜**

**KHOÁ LUẬN**

**TỐT NGHIỆP ĐẠI HỌC**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

**Đề tài:**

**XÂY DỰNG HỆ THỐNG**

**HỌC TẬP TRỰC TUYẾN**

**Sinh viên thực hiện:** **Phạm Văn Nhân**

**Khoá: K45 – Hệ chính quy**

**Giáo viên hướng dẫn: ThS. Hoàng Nguyễn Tuấn Minh**

**Huế, tháng 04 năm 2025**

**LỜI CẢM ƠN**

Trong quá trình học tập về ngành Công nghệ thông tin tại Trường Đại học Khoa học Huế, em đã tiếp thu được nhiều kiến thức mới bổ sung thêm vào hành trang và khả năng vận dụng kiến thức đó của mình vào thực tiễn một cách hợp lý và khoa học. Em thấy được tính hữu dụng, cũng như sự cần thiết của các môn học, đã tác động mạnh mẽ đến em khi áp dụng vào thực tế khi làm bài khóa luận tốt nghiệp.

Để hoàn thành chương trình đại học ngành công nghệ phần mềm cũng như khóa luận tốt nghiệp; qua đây, em xin chân thành cảm ơn đến quý Thầy, Cô của Trường Đại học Khóa học - Đại Học Huế nói chung; quý Thầy, Cô tại khoa Công nghệ thông tin nói riêng, đã tạo điều kiện tốt nhất cho học viên được học tập. Đặc biệt, sự tận tình hướng dẫn của Giảng viên ThS. Hoàng Nguyễn Tuấn Minh đã tận tình hướng dẫn tác giả thực hiện đề tài. Đồng thời, em xin chân thành cảm ơn đến bạn bè, gia đình, người thân đã tận tình hỗ trợ và động viên em trong suốt thời gian học tập và nghiên cứu. Trong quá trình thực hiện, mặc dù đã hết sức cố gắng hoàn thiện khóa luận tốt nghiệp, trao đổi và tiếp thu ý kiến đóng góp từ quý Thầy, Cô và bạn bè nhưng với thời gian nghiên cứu và kiến thức còn hạn chế, bài khóa luận này không tránh khỏi những sai sót.

Rất mong nhận được những góp ý của quý Thầy/Cô.

Huế, tháng 5, năm 2024

Phạm Văn Nhân

**DANH MỤC CHỮ VIẾT TẮT**

|  |  |
| --- | --- |
| … | … |

**DANH MỤC BẢNG BIỂU**

….

**DANH MỤC HÌNH ẢNH**

….

**MỤC LỤC**

[**PHẦN MỞ ĐẦU** 1](#_Toc195089178)

[**1. Lý do chọn đề tài** 1](#_Toc195089179)

[**2. Mục tiêu của đề tài** 1](#_Toc195089180)

[**2.1. Mục tiêu tổng quát** 1](#_Toc195089181)

[**2.2. Mục tiêu cụ thể** 1](#_Toc195089182)

[**3. Phương pháp nghiên cứu** 1](#_Toc195089183)

[**4. Bố cục tổng quát của khóa luận** 2](#_Toc195089184)

[**NỘI DUNG NGHIÊN CỨU** 2](#_Toc195089185)

[**CHƯƠNG 1: TỔNG QUAN VỀ CÔNG NGHỆ SỬ DỤNG** 2](#_Toc195089186)

[**1.1 Mô hình hệ thống** 2](#_Toc195089187)

[**1.2. Công nghệ sử dụng** 4](#_Toc195089188)

[**1.2.1. Nodejs** 4](#_Toc195089189)

[**1.2.2. Express framework** 4](#_Toc195089190)

[**1.2.3. Reactjs** 4](#_Toc195089191)

[**1.2.4. Mongodb** 5](#_Toc195089192)

[**CHƯƠNG 2: PHÂN TÍCH, THIẾT KẾ, VẬN HÀNH HỆ THỐNG QUẢN LÝ TRUNG TÂM GIÁO DỤC** 5](#_Toc195089193)

[**2.1. Tổng quan nghiệp vụ.** 5](#_Toc195089194)

[**2.2. Cơ sở dữ liệu** 5](#_Toc195089195)

# **PHẦN MỞ ĐẦU**

## **1. Lý do chọn đề tài**

Trong kỷ nguyên số hiện nay, nhu cầu học tập trực tuyến ngày càng gia tăng, đặc biệt là trong bối cảnh công nghệ phát triển nhanh chóng và việc truy cập Internet trở nên phổ biến. Học tập trực tuyến không chỉ mang lại sự linh hoạt về thời gian và địa điểm cho người học mà còn mở ra cơ hội tiếp cận tri thức một cách đa dạng, cá nhân hóa và hiệu quả hơn.

Tuy nhiên, nhiều nền tảng học trực tuyến hiện nay vẫn còn tồn tại một số hạn chế như giao diện phức tạp, thiếu thân thiện với người dùng Việt Nam, hoặc chưa tối ưu trong việc hỗ trợ học ghi nhớ – ôn tập – luyện tập theo cách đơn giản và trực quan. Điều này đặt ra nhu cầu thiết thực về việc xây dựng một hệ thống học tập trực tuyến thân thiện, dễ sử dụng, hỗ trợ người học tạo nội dung kiến thức, ôn luyện qua các hình thức như trắc nghiệm, flashcard, ghi nhớ, và theo dõi tiến trình học tập của bản thân.

Xuất phát từ nhu cầu thực tế đó, đề tài “Xây dựng hệ thống học tập trực tuyến” được lựa chọn nhằm mục tiêu tạo ra một nền tảng học tập hiện đại, tập trung vào trải nghiệm người dùng và hỗ trợ học tập hiệu quả. Hệ thống không chỉ phù hợp cho học sinh – sinh viên mà còn có thể mở rộng cho người đi làm hoặc bất kỳ ai có nhu cầu tự học.

Ngoài ra, việc thực hiện đề tài còn là cơ hội để áp dụng các kiến thức chuyên môn đã học như lập trình web với Node.js, xây dựng giao diện hiện đại với React.js, thiết kế cơ sở dữ liệu với MongoDB, cũng như vận dụng các kỹ năng phân tích, thiết kế hệ thống và xử lý dữ liệu. Đây là một đề tài phù hợp với xu hướng chuyển đổi số trong giáo dục, đồng thời có tiềm năng phát triển thực tế trong tương lai.

## **2. Mục tiêu của đề tài**

### **2.1. Mục tiêu tổng quát**

Xây dựng một hệ thống học tập trực tuyến dưới dạng website, hỗ trợ người dùng tạo, lưu trữ và ôn luyện kiến thức thông qua các chức năng như tạo bộ thẻ học (flashcard), làm bài trắc nghiệm, ghi nhớ và theo dõi tiến trình học tập. Hệ thống hướng đến việc đơn giản hóa quá trình học tập, tạo môi trường học tập thân thiện, dễ sử dụng, góp phần nâng cao hiệu quả tiếp thu kiến thức của người học.

### **2.2. Mục tiêu cụ thể**

* Xây dựng một website học tập trực tuyến cho phép người dùng dễ dàng tạo và quản lý nội dung học tập như thẻ ghi nhớ (flashcard), câu hỏi trắc nghiệm và bài luyện tập.
* Phát triển giao diện người dùng thân thiện, dễ sử dụng và có khả năng phản hồi nhanh bằng công nghệ **ReactJS**, giúp nâng cao trải nghiệm học tập.
* Xây dựng hệ thống phía máy chủ (**backend**) sử dụng **NodeJS** và **ExpressJS** để xử lý các chức năng chính như xác thực người dùng, quản lý nội dung học và lưu trữ tiến trình học tập.
* Thiết kế và xây dựng cơ sở dữ liệu bằng **MongoDB** để lưu trữ thông tin người dùng, nội dung học tập và kết quả ôn luyện một cách hiệu quả và linh hoạt.
* Triển khai chức năng đăng ký, đăng nhập và quản lý tài khoản người dùng.
* Hỗ trợ người dùng tạo bộ học (gồm flashcard hoặc câu hỏi trắc nghiệm), chỉnh sửa, xóa và phân loại theo chủ đề.
* Phát triển tính năng ôn luyện kiến thức dưới nhiều hình thức như: ghi nhớ thẻ, trắc nghiệm, kiểm tra ngẫu nhiên.
* Tích hợp hệ thống theo dõi tiến trình học tập, cho phép người dùng xem lại lịch sử học, kết quả các lần ôn tập và thống kê hiệu suất học tập.
* Đảm bảo hệ thống hoạt động ổn định, dễ sử dụng và có khả năng mở rộng về sau (học nhóm, chia sẻ nội dung học …)

## **3. Phương pháp nghiên cứu**

* Phương pháp thu thập thông tin:

1. Phương pháp phỏng vấn: thực hiện phỏng vấn các bạn học sinh, sinh viên trên địa bàn tỉnh Thừa Thiên Huế để đưa ra các giải pháp phù hợp, cũng như thiết kế hệ thống đảm bảo được các tiêu chí đề ra.
2. Phương pháp điền phiếu khảo sát: thực hiện khảo sát các học sinh, sinh viên về sự thuận lợi, bất lợi khi sử dụng các ứng dụng học tập khác

* Phương pháp kiểm thử:

1. Kiểm thử hộp đen: tiến hành kiểm thử các mặt liên quan đến nghiệp vụ của hệ thống, nhằm đảm bảo tính nhất quán phù hợp của hệ thống.
2. Unit Test: Tiến hành test các đơn vị khi cài đặt các chức năng.

## **4. Bố cục tổng quát của khóa luận**

Chương 1: Tổng quan về công nghệ sử dụng.

Chương 2: Phân tích, thiết kế, vân hành hệ thống quản lý trung tâm giáo dục.

Chương 3: Kết luận và hướng phát triển.

# **NỘI DUNG NGHIÊN CỨU**

# **CHƯƠNG 1: TỔNG QUAN VỀ CÔNG NGHỆ SỬ DỤNG**

## **1.1 Mô hình hệ thống**

Hệ thống học tập trực tuyến được xây dựng theo kiến trúc **Client - Server**, bao gồm ba thành phần chính: **Giao diện người dùng (Frontend)**, **Hệ thống xử lý phía máy chủ (Backend)** và **Cơ sở dữ liệu (Database)**. Mỗi thành phần đảm nhận một vai trò cụ thể, đảm bảo hệ thống hoạt động mượt mà, ổn định và dễ mở rộng.

**4.1. Mô hình tổng thể**

* **Client (ReactJS):**

Giao diện được xây dựng bằng ReactJS giúp người dùng dễ dàng tương tác với hệ thống: đăng ký, đăng nhập, tạo bộ học, ôn tập, xem tiến trình... Ứng dụng hướng đến trải nghiệm người dùng trực quan, thân thiện và hiệu năng cao.

* **Server (NodeJS + Express):**

Máy chủ tiếp nhận các yêu cầu từ client, xử lý logic nghiệp vụ như xác thực người dùng, quản lý nội dung học, xử lý kết quả ôn tập,... Express hỗ trợ xây dựng API RESTful giúp việc giao tiếp giữa frontend và backend trở nên hiệu quả và rõ ràng.

* **Database (MongoDB):**

Lưu trữ dữ liệu hệ thống như thông tin người dùng, các bộ học, kết quả ôn luyện,... MongoDB phù hợp với mô hình dữ liệu linh hoạt và có khả năng mở rộng tốt cho hệ thống học tập có lượng dữ liệu lớn.

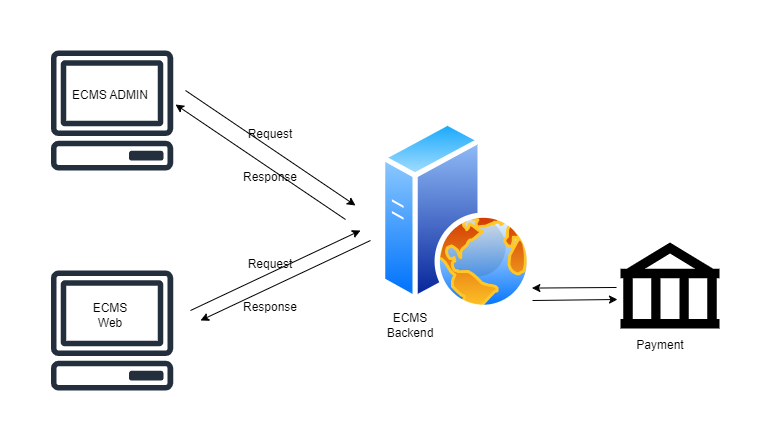
**4.2. Mô hình luồng dữ liệu cơ bản**

* Người dùng truy cập vào website (giao diện ReactJS).
* Người dùng gửi yêu cầu (như đăng nhập, tạo flashcard, luyện tập...).
* Server (NodeJS) nhận yêu cầu, xử lý logic và truy vấn dữ liệu nếu cần.
* Kết quả được lấy từ MongoDB và trả lại phía client.
* Client hiển thị kết quả cho người dùng.

**4.3. Mô hình triển khai (Deployment)**

Hệ thống có thể được triển khai trên nền tảng đám mây hoặc máy chủ riêng với cấu hình như sau:

* **Frontend và Backend** triển khai trên VPS.
* **Cơ sở dữ liệu MongoDB** có thể lưu trữ tại Cloud MongoDB Atlas.



*Hình 1: Mô hình hệ thống*

## **1.2. Công nghệ sử dụng**

### **1.2.1. Nodejs**

Node.js là một môi trường chạy JavaScript phía máy chủ, được xây dựng trên nền tảng V8 Engine của Google. Node.js cho phép xây dựng các ứng dụng web có khả năng xử lý bất đồng bộ và hiệu suất cao, đặc biệt phù hợp với các hệ thống yêu cầu khả năng xử lý thời gian thực như học trực tuyến.  
Việc sử dụng Node.js giúp tăng tốc độ phát triển ứng dụng nhờ vào việc dùng chung một ngôn ngữ (JavaScript) cho cả frontend và backend. Ngoài ra, Node.js còn có một hệ sinh thái package rất lớn thông qua npm, hỗ trợ việc mở rộng và tích hợp hệ thống một cách linh hoạt.

### **1.2.2. Express framework**

Express là một framework nhẹ của Node.js dùng để xây dựng các ứng dụng web và API. Express hỗ trợ định tuyến mạnh mẽ, middleware linh hoạt và khả năng mở rộng cao.  
Trong đồ án này, Express được sử dụng để xây dựng backend API phục vụ cho các chức năng như quản lý khóa học, đăng nhập, phân quyền người dùng, và xử lý dữ liệu giữa client và server.  
Việc kết hợp Express với Node.js tạo ra một nền tảng vững chắc, đơn giản nhưng hiệu quả cho hệ thống học tập trực tuyến.

### **1.2.3. Reactjs**

React là một thư viện mã nguồn mở được phát triển bởi Facebook để xây dựng giao diện người dùng cho các ứng dụng web. React là một công nghệ thuận lợi cho việc phát triển trang web sigle page. Với hiệu xuất cao khi các thành phần của trang web được thực thể hóa, không phụ thuộc chặt chẽ vào một màn hình nhất định. Do đó việc xử dụng React được xem là một công nghệ phù hợp đối với đồ án này.

Bên cạnh đó React được hỗ trợ bởi nhiều package từ nodejs khiến ứng dụng trở nên da dạng hơn.

### **1.2.4. Mongodb**

MongoDB là một cơ sở dữ liệu NoSQL dạng tài liệu, lưu trữ dữ liệu dưới định dạng BSON (một biến thể của JSON). MongoDB cho phép lưu trữ dữ liệu linh hoạt, dễ mở rộng và phù hợp với các hệ thống có cấu trúc dữ liệu không cố định.  
Với hệ thống học tập trực tuyến, MongoDB hỗ trợ việc lưu trữ thông tin người dùng, khóa học, bài học, điểm số... một cách hiệu quả và linh hoạt.  
Khả năng tích hợp tốt với Node.js thông qua thư viện Mongoose giúp việc tương tác với cơ sở dữ liệu trở nên dễ dàng và rõ ràng hơn.

# **CHƯƠNG 2: PHÂN TÍCH, THIẾT KẾ, VẬN HÀNH HỆ THỐNG QUẢN LÝ TRUNG TÂM GIÁO DỤC**

* 1. **Tổng quan nghiệp vụ**

**…..**

*Hình 2: Use case tổng quát*

-  **Quản lý phân quyền:** Các quyền được tạo nên từ nghiệp vụ hoạt động của trung tâm, dưới được định nghĩa là các vai trò, thấp hơn vai trò là những thành phần phân quyền nhỏ hơn (permission).

- **Quản lý người dùng:** người dùng là người tạo tài khoản nhưng chưa đăng ký bất kỳ khóa học nào. Từ lớp người dùng ta có thể mở rộng thành hai lớp là học viên và giảng viên, là các tác nhân chính thao tác với hệ thống.

- **Quản lý flashcard:** các danh mục bao gồm quản lý phòng học, quản lý khóa học sẽ được quản lý bởi danh mục.

- **Quản lý quá trình học:** Lớp học là lớp quan trọng trong luồng hoạt động của nghiệp vụ trung tâm.

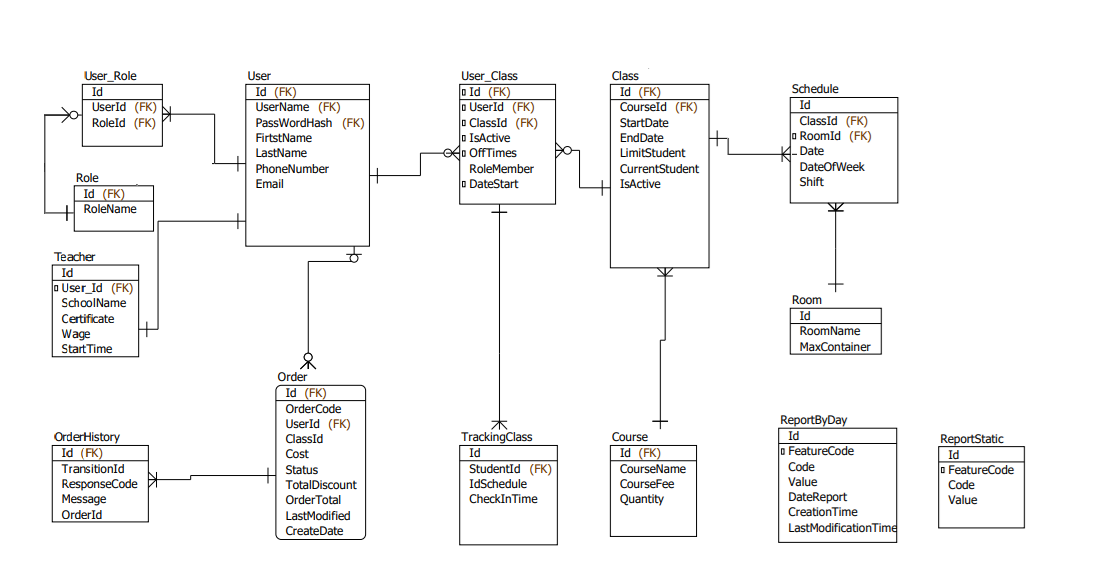
- **Quản lý lịch học:** Nhằm tối ưu tối đa công năng của cơ sở vật chất lịch học phải được quản lý một cách chặt chẽ.

- **Thanh toán:** Hệ thống quản lý việc thanh toán học phí, bao gồm các hình thức chuyển tiền qua ngân hàng, ví điện tử và trực tiếp

- **Thống kê báo cáo:** thu thập đánh giá hiệu quả kinh doanh của doanh nghiệp.

## **2.2. Cơ sở dữ liệu**

### **2.2.1 Lược đồ quan hệ của cơ sở dữ liệu**



*Hình 3: Cơ sở dữ liệu hệ thống quản lý trung tâm giáo dục*

### **2.2.2 Các trường thông dụng:**

Các trường nhằm mục đích theo dõi các bảng.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| 1 | CreatedAt | Datetime | not null | Thời gian tạo bản ghi |
| 2 | UpdatedAt | Datetime | null | Thời gian lần cuối cập nhật |

*Bảng 1: Danh sách các trường thông dụng có trong các bảng*

### **2.2.3 Bảng User**

Lưu trữ thông tin cơ bản của người dùng.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| 1 | \_id | ObjectId | auto increment | Khóa chính của người dùng |
| 2 | username | String | required, unique, minlength: 6 | Tên đăng nhập |
| 3 | password | String | required, minlength: 8 | Mật khẩu đã được mã hóa |
| 4 | email | String | required, unique | Địa chỉ email người dùng |
| 5 | name | String | required, unique | Họ tên người dùng |
| 6 | bio | String | optional, default: '' | Tiểu sử hoặc giới thiệu ngắn |
| 7 | profilePic | String | optional, default: '' | URL ảnh đại diện |
| 8 | isAdmin | Boolean | default: false | Người dùng có phải là admin hay không |

*Bảng 2: Bảng User*

### **2.2.4 Bảng Card**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên Trường | Kiểu Dữ Liệu | Ràng Buộc | Mô Tả |
| 1 | title | String | required, default: "" | Tiêu đề của thẻ |
| 2 | description | String | default: "" | Mô tả nội dung của thẻ |
| 3 | slug | String | default: "" | Slug thân thiện dùng cho URL |
| 4 | user | ObjectId (ref: User) | required | Người tạo thẻ |
| 5 | share | Boolean | required, default: true | Trạng thái chia sẻ công khai hay riêng tư |
| 6 | views | Number | default: 5 | Số lượt xem của thẻ |
| 7 | background | String | default: "" | Màu nền hoặc ảnh nền của thẻ |
| 8 | rate.total | Number | default: 0 | Tổng số điểm đánh giá |
| 9 | rate.quantity | Number | default: 0 | Tổng số lượt đánh giá |
| 10 | type | String | default: "regular" | Loại thẻ (ví dụ: regular, private,...) |
| 11 | password | String | default: "" | Mật khẩu bảo vệ thẻ nếu có |

*Bảng 3: Bảng Card*

### **2.2.5 Bảng Term**

Lưu trữ thông tin vai trò của người dùng trong hệ thống.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên Trường | Kiểu Dữ Liệu | Ràng Buộc | Mô Tả |
| 1 | prompt | String | default: "" | Từ/câu hỏi gợi ý (mặt trước thẻ) |
| 2 | answer | String | default: "" | Câu trả lời tương ứng (mặt sau thẻ) |
| 3 | cardId | ObjectId (ref: Card) | optional | Thẻ học (card) chứa term này |
| 4 | position | Number | optional | Vị trí của term trong card |

*Bảng 4: Bảng Term*

### **Bảng TickMark**

Bảng lưu trữ thông tin giáo viên giảng dạy tại trung tâm.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên Trường | Kiểu Dữ Liệu | Ràng Buộc | Mô Tả |
| 1 | term | ObjectId (ref: Term) | optional | Term được đánh dấu (liên kết đến bảng Term) |
| 2 | card | ObjectId (ref: Card) | optional | Card chứa term tương ứng |
| 3 | isLearn | Array | default: [] | Dữ liệu học (Learn mode) |
| 4 | isTest | Array | default: [] | Dữ liệu kiểm tra (Test mode) |
| 5 | isWrite | Array | default: [] | Dữ liệu luyện viết (Write mode) |
| 6 | isListen | Array | default: [] | Dữ liệu nghe (Listen mode) |
| 7 | isFlashCard | Array | default: [] | Dữ liệu flashcard (FlashCard mode) |
| 8 | isMatch | Array | default: [] | Dữ liệu ghép cặp (Match mode) |
| 9 | timestamps | Auto | createdAt, updatedAt tự động tạo khi lưu |  |

*Bảng 5: Bảng Teacher*

### **2.2.7 Bảng Room**

Lưu trữ thông tin các phòng học. Nhẳm đảm bảo nắm được thông tin cơ sở vật chất.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên Trường | Kiểu Dữ Liệu | Ràng Buộc | Mô Tả |
| 1 | user | ObjectId (ref: User) | optional | Người tạo hoặc liên kết với người dùng trong hệ thống |
| 2 | roomId | String | required, unique | ID phòng (duy nhất) |
| 3 | members | Array of Objects | optional | Danh sách thành viên trong phòng (gồm: id, mark, name) |
| 4 | members.id | String | optional | ID của thành viên |
| 5 | members.mark | Number | optional | Điểm số của thành viên |
| 6 | members.name | String | optional | Tên của thành viên |
| 7 | status | String | required, default: "waiting" | Trạng thái của phòng (VD: waiting, active, done,...) |
| 8 | timestamps | Auto | createdAt, updatedAt tự động sinh |  |

*Bảng 6: Bảng Room*

### **2.2.8 Bảng Rate**

Bảng lịch biểu lưu trử thông tin về thời khóa biểu cho nhiều lớp. Đồng thời là nỏi lưu trữ thông tin phòng học, thời gian học của lớp học.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên Trường | Kiểu Dữ Liệu | Ràng Buộc | Mô Tả |
| 1 | card | ObjectId (ref: Card) | required | Tham chiếu đến thẻ (Card) được đánh giá |
| 2 | user | ObjectId (ref: User) | required | Người đánh giá |
| 3 | rate | Number | default: 1 | Giá trị đánh giá (1 đến 5) |

*Bảng 7: Bảng Rate*

### **2.2.9 Bảng Notification**

Bảng notification lưu trữ thông tin các thông báo đến người dung.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| 1 | Id | int | primary key, not null, auto increment | ID của từng tài khoản |
| 2 | CourseName | String | Not null | Tên khóa học |
| 3 | CourseFee | Decema | Not null | Học phí |
| 4 | Quantity | int | Not null | Số lượng |

*Bảng 8: Bảng Notification*

### **2.2.10 Bảng Class**

**Bảng Achieve** ung để lưu trữ thông tin thành tích học tập của người ung theo từng hình thức học cụ thể.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Ràng buộc | Mô tả |
| 1 | Id | ObjectId | Primary key, Not null | ID của bản ghi thành tích |
| 2 | user | ObjectId | Ref: User | ID của người dùng |
| 3 | target | Number | Default: 100 | Mục tiêu tổng số hoạt động cần hoàn thành |
| 4 | achieveLearn | Number | Default: 0 | Số lần học (Learn) đã hoàn thành |
| 5 | achieveTest | Number | Default: 0 | Số lần kiểm tra (Test) đã hoàn thành |
| 6 | achieveWrite | Number | Default: 0 | Số lần viết (Write) đã hoàn thành |
| 7 | achieveListen | Number | Default: 0 | Số lần nghe (Listen) đã hoàn thành |

*Bảng 9: Bảng Class*

### **2.2.11 Bảng Board**

**Bảng Board** dùng để lưu thông tin bảng quản lý các thẻ học tập của người dùng, phục vụ cho việc sắp xếp và tổ chức nội dung học.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Ràng buộc | Mô tả |
| 1 | Id | ObjectId | Primary key, Not null | ID của bảng |
| 2 | title | String | Default: “Sắp xếp thời gian học tập” | Tiêu đề của bảng |
| 3 | user | ObjectId | Ref: User | ID của người ung sở hữu bảng |
| 4 | cardOrder | Array | Default: [] | Danh sách ID của các thẻ (Card) được sắp xếp trong bảng |

*\*

*Bảng 10: Bảng TrackingClass*

### **2.2.12 Bảng Session**

Là bảng xác định một người dùng là một học viên của lớp học nhất định. Đồng thời vai trò của học viên và giảng viên cũng được quy định.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| 1 | Id | int | primary key, not null, auto increment | ID của từng tài khoản |
| 2 | UserId | Bigint | Not null | ID người dùng |
| 3 | ClassId | Bigint | Not null | ID lớp học |
| 4 | IsActive | bit | Not null | Trạng thái học sinh |
| 5 | OffTimes | int | Not null | Số lần nghỉ học |
| 6 | RoleMember | Int | Not null | Vai trò thành viên |
| 7 | DateStart | datetime | Not null | Ngày bắt đầu |

*Bảng 11: Bảng User\_Class*

### **2.2.13 Bảng Order**

Lưu trữ thông tin các đơn hàng. Đối với các đơn hàng thanh toán trực tuyến trạng thái của hóa đơn sẽ được khởi tạo với trạng thái đang chờ, và sẽ tự động được cập nhật lại trạng thái khi có phản hồi từ bên thứ ba (vnPay).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| STT | Trường | Kiểu dữ liệu | Ràng buộc | Mô tả |
| 1 | \_id | ObjectId | Primary key, tự động sinh | ID của hóa đơn |
| 2 | gateway | String | Mặc định: "sepay" | Tên cổng thanh toán được sử dụng |
| 3 | transaction\_date | Date | Mặc định: ngày gốc (01/01/1970) | Ngày giao dịch thanh toán |
| 4 | account\_number | String | Có thể null | Số tài khoản thanh toán |
| 5 | sub\_account | String | Có thể null | Tài khoản phụ nếu có |
| 6 | amount | Number | Not null, mặc định: 0 | Số tiền giao dịch |
| 7 | code | String | Not null | Mã giao dịch hoặc mã hóa đơn |
| 8 | transaction\_content | String | Có thể null | Nội dung giao dịch từ cổng thanh toán |
| 9 | description | String | Có thể null | Mô tả thêm cho hóa đơn |
| 10 | content | String | Có thể null | Nội dung thêm liên quan đến hóa đơn |
| 11 | userId | ObjectId (ref: User) | Not null, khóa ngoại | ID người dùng thực hiện giao dịch |
| 12 | planType | String | Not null, chỉ nhận: MONTHLY, YEARLY | Loại gói đăng ký (Theo tháng hoặc theo năm) |
| 13 | status | String | Mặc định: PENDING, chỉ nhận: PENDING, SUCCESS, FAILED | Trạng thái hiện tại của hóa đơn |

*Bảng 12: Bảng Oder*

### **2.2.14 Bảng OrderHistory**

Lịch sử hóa đơn nơi lưu lại các phát sinh của quá trình thanh toán đối với các giao dịch trực tuyến. Trong đó TransitionId là trường do bên thứ 3 phát hành.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| 1 | Id | int | primary key, not null, auto increment | ID của từng tài khoản |
| 2 | TransitionId | Nvarchar(20) | Not null | Mã theo dõi |
| 3 | ReponseCode | Navchar(10) | Not null | Mã trạng thái đơn vị thanh toán trả về |
| 4 | Message | String | Not null | Thông tin |
| 5 | OrderId | Bigint | Not null | Mã đơn đặt hàng |

*Bảng 13: Bảng OrderHistory*

**2.2.15 Bảng ReportByDay**

Lưu trữ các thông tin thông kê theo từng ngày. Với mục đích tối ưu thời gian truy vấn đối với các tác vụ thống kê

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| 1 | Id | int | primary key, not null, auto increment | ID của từng tài khoản |
| 2 | FeatureCode | Nvarchar(20) | Not null | Mã gom nhóm thống kê |
| 3 | Code | Navchar(10) | Not null | Mã thống kê chi tiết |
| 4 | DateReport | DateTime | Not null | Ngày báo cáo |
| 5 | LastModification | DateTime | Not null | Lần cuối cập nhật |

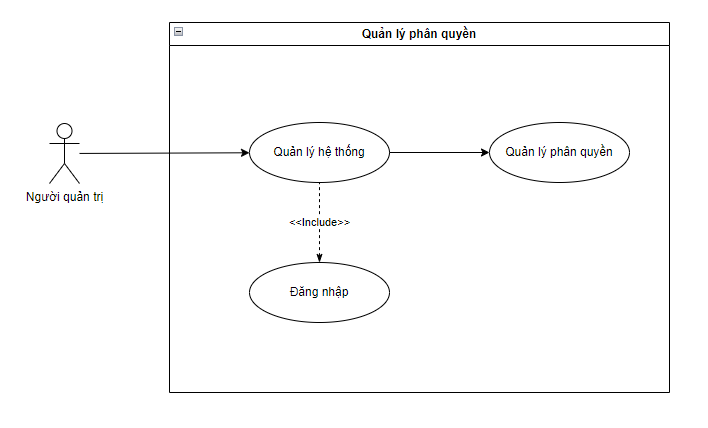
**2.2.15 Bảng ReportStatic**

Lưu trữ các thông tin thông kê theo cố định không yêu cầu các tác vụ nâng cao như lọc, phân loại thông báo.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| 1 | Id | int | primary key, not null, auto increment | ID của từng tài khoản |
| 2 | FeatureCode | Nvarchar(20) | Not null | Mã gom nhóm thống kê |
| 3 | Code | Navchar(10) | Not null | Mã thống kê chi tiết |
| 4 | LastModification | DateTime | Not null | Lần cuối cập nhật |

## **2.3. Phân tích chức năng.**

### **2.3.1 Quản lý phân quyền.**



Hình 4: Biểu đồ Use Case quản lý phân quyền

#### **2.3.1.1 Mô tả chức năng**

**a. Giới thiệu**

Phân quyền là chức năng quan trọng đối với hệ thống nhằm phân bố quyền truy cập của người dùng khi tác động vào hệ thống.

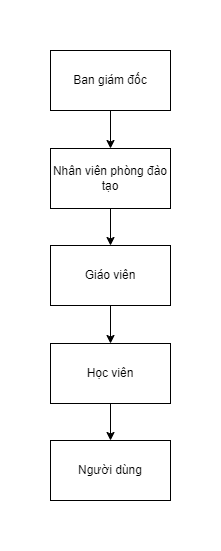
Đối với trung tâm giáo dục quyền hạn sẽ được phân theo hình ảnh tuyến tính sau, quyền truy cập vào hệ thống sẽ bị giới hạn theo đó.

**b. Mục tiêu**

Các API được phân quyền đúng, để đảm bảo quyền truy cập của người dùng.

Ngăn chặn các truy cập không hợp lệ vào những tài nguyên quan trọng.

Cơ chế vai trò giúp giảm sự phức tạo khi phân quyền.

****

- **Ban giám đốc:** là tác nhân có quyền cao nhất bao gồm các quyền về báo cáo thông kê, theo dõi thanh toán, quản lý nhân viên dưới quyền.

**- Nhân viên phòng đào tạo:** là tác nhân có các quyền vè quản lý lớp học, sắp xếp lịch học, xử lý các vấn đề phát sinh trong việc đăng ký lớp và mở lớp.

**- Giảng viên:** Là tác nhân có các quyền về quản lý lớp học do người đó trực tiếp giảng dạy, giảng viên có thể thay đổi các thông tin cơ bản về các chỉ số học tập của học viên.

**- Học viên:** là tác nhân quan trọng, tham gia trực tiếp vào các lớp học, có các quyền xem thông tin cá nhân và theo dõi lớp học (lịch học, bài tập, bài đăng, điểm số…)

**- Người dùng:** là tác nhân chỉ có thể xem thông tin cá nhân và các thông tin cơ bản của các lớp học. Quyền của người dùng sẽ là học viên khi đăng ký ít nhất một lớp học.

*Hình 5: Sơ đồ luồng dữ liệu*

#### **2.3.1.2: Triển khai trên hệ thống**

**a. Định nghĩa:**

Trong quản lý phân quyền có một số định nghĩa sau đây:

Role: vai trò của người dùng trong hệ thống các role dược phân chia theo hướng chức năng. Một người dùng có nhiều role khác nhau, người dùng có nhiều role đồng nghĩa với việc người dùng có khả năng tác động lên hệ thống càng lớn.

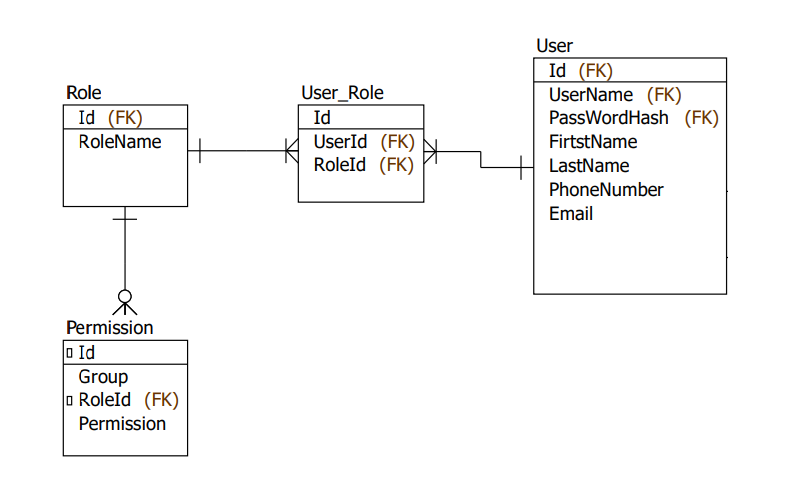
Permission: là thành phân phân bố sự truy cập đến các dịch vụ của hệ thống. Một role được cấu thành từ nhiều permission. Mỗi chức năng sẽ có 4 permission cơ bản dựa trên RESTful API đó là: thêm, xóa, sửa, đọc.

Một vài chức năng mở rộng có thể như là import, export, scan.

Ví dụ: Nhân viên phòng đào tạo vừa quản lý lớp học, vừa quản lý việc thanh toán. Nhân viên sẽ có 2 role là role\_class và role\_payment. Để đảm bảo tính minh bạch trong thu chi nhân viên chỉ có quyền đọc đối với chức năng thanh toán, trong khi đó vai trò quản lý lớp học được cấp cho nhân viên sẽ đầy đủ các permission dể có thể quản lý hiệu quả.

|  |  |  |
| --- | --- | --- |
| **Permission** | **Class\_role** | **Payment\_role** |
| Read | x | x |
| Create | x |  |
| Update | x |  |
| Delete | x |  |

Sơ đồ ERD:

****

*Hình 6: Sơ đồ ERD quản lý phân quyền*

**b. Xử lý ở phía server**

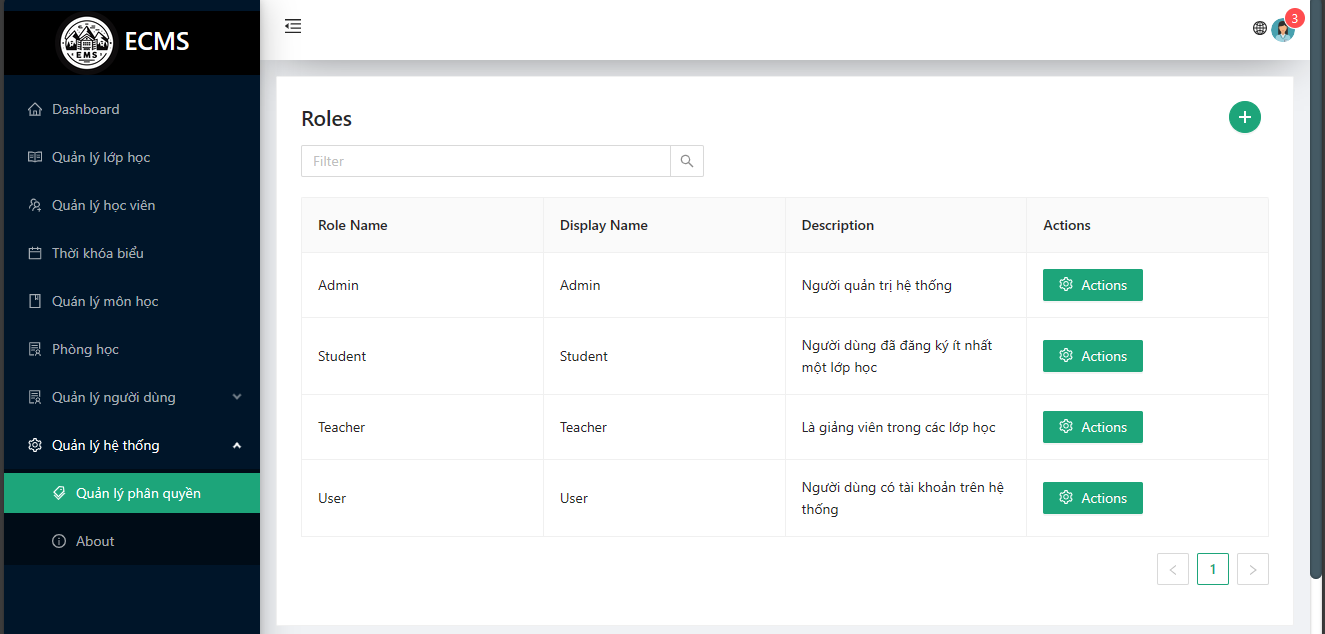
- Chức năng thêm xóa sửa một permission.

Các permission là thông tin quan trọng được định nghĩa bằng một lơp chứa các hằng số. Để đảm bảo không có bất kỳ một permission không hợp lệ nào tồn tại. Các permission sẽ tự động thêm vào cơ sở dữ liệu khi tiến hành sử dụng hằng số permission cho một API bất kỳ.

- Xây dựng middleware để cấp quyền truy cập cho các API dựa trên permission. Mỗi API sẽ có một permission.

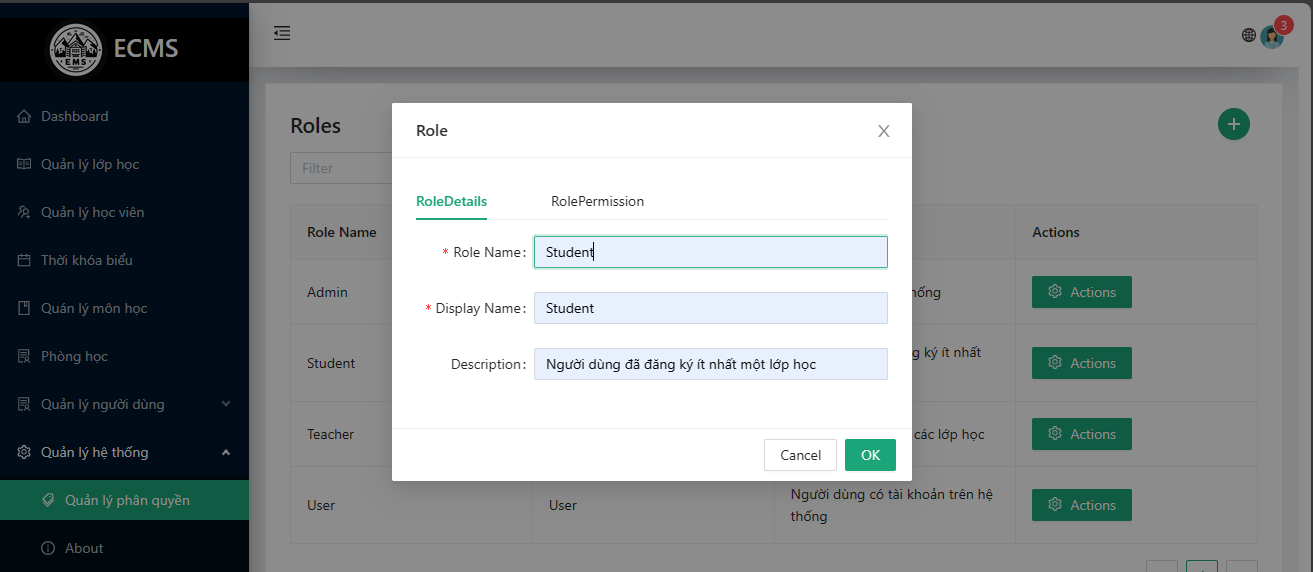
- Chức năng thêm xóa sửa vai trò (role)

Giao diện thêm xem danh sách vai trò:



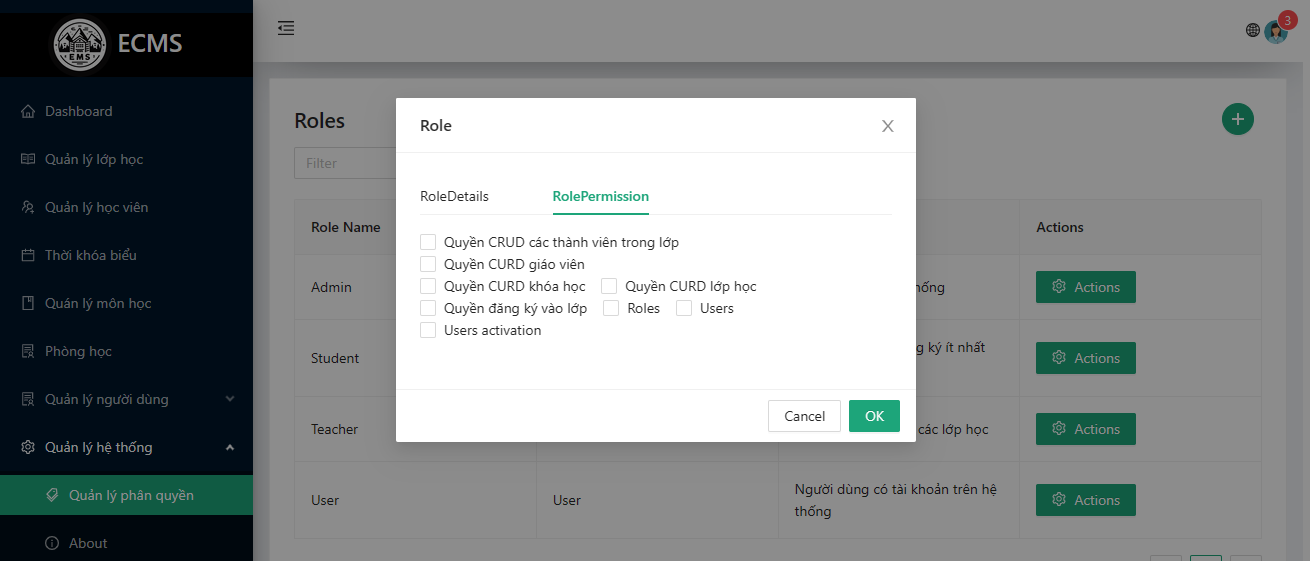
*Hình 7: Màn hình danh sách vai trò.*

Giao diện thêm vai trò:



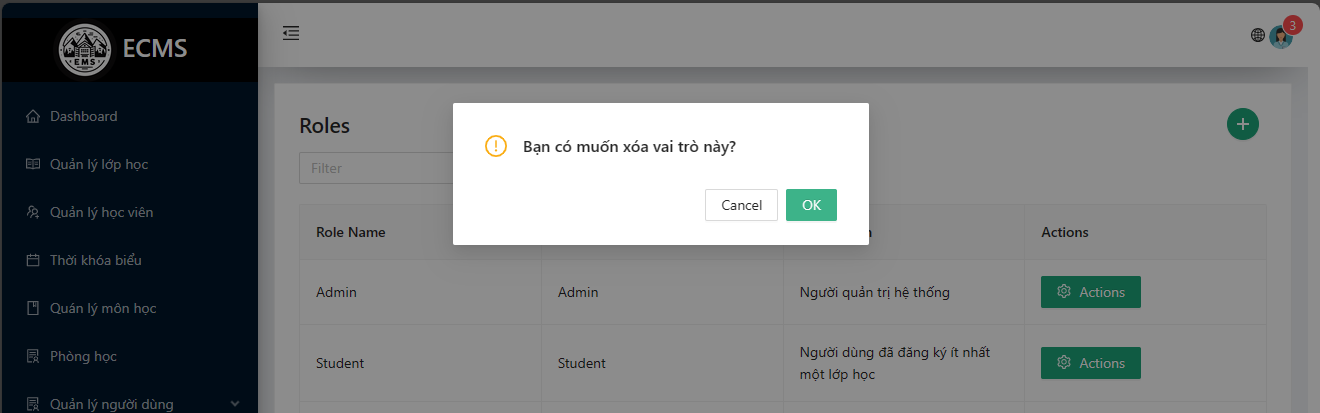
*Hình 8: Màn hình thêm vai trò*

+ Chọn permisison:



*Hình 9: Màn hình chọn permisison*

+ Xóa một vai trò:



*Hình 10: Màn hình xóa vai trò*

- Chức năng phân quyền cho người dùng. Người dùng sẽ được khởi tạo với role ban đầu là user để thao tác trực tiếp trên trang chủ.

**c. Xác thực người dùng.**

Sau khi đã phân quyền cho người dùng việc xác thực người dùng là một yếu tố bắt buộc dể cung cấp đúng các dịch vụ đồng thời ngăn chặn những truy cập trái phép.

JSON Web Token (JWT) là một chuỗi dữ liệu được mã hóa nhằm trao đổi thông tin một cách an toàn.

Về cấu trúc chia làm ba phần:

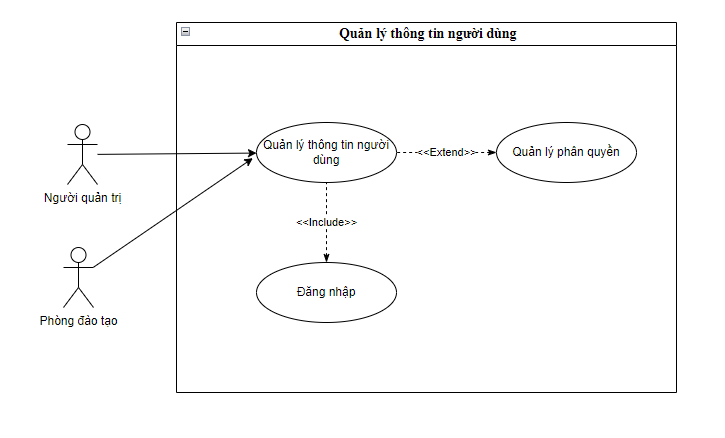
+ Header chứa thông tin về thuật toán mã hóa.

+ Payload chứa các thông tin về các role, permisison.

+ Signature là phần chữ ký bí mật phần này được tạo ra bởi thuật toán mã hóa phần Header.

Khi ba phần này kết hợp sẽ tạo ra một chuỗi chỉ có thể dịch theo một chiều. Và việc thay đổi nội dung bên trong thông qua việc thay đổi token là không khả thi.

### **2.3.2. Quản lý thông tin người dùng**



*Hình 11: Use Case chức năng quản lý người dùng.*

#### **2.3.2.1. Mô tả chức năng**

**a. Giới thiệu**

Chức năng quản lý người dùng giúp quản lý thông tin người dùng trong hệ thống, bao gồm học viên, giáo viên, và quản trị viên. Đây là chức năng quan trọng để đảm bảo sự hiệu quả trong việc quản lý và sử dụng hệ thống.

Quản lý thông tin người dùng bao gồm các chức năng thêm xóa sửa tìm kiếm.

Trong đó:

- Chức năng thêm sẽ do 2 tác nhân chính sử dụng đó là người dùng đăng ký mới mặc định sẽ gán với vai trò user và do người quản trị thêm. Khi người quản trị thêm vào thì các vai trò sẽ do người quản trị chỉ định.

- Đối với chức năng đọc phân chia thành hai phần nhỏ hơn Admin sẽ là người đọc và kiểm soát toàn bộ thông tin của người dùng. Trong khi đó người dùng chỉ có thể đọc thông tin cá nhân thông qua chính token của mình.

- Đối với quyền xóa và sửa, Admin là người có thể kiểm soát các chức năng này. Người dùng chỉ có thế thay đổi xóa bỏ chính thông tin của mình thông qua token đã cấp phát.

**b. Mục tiêu**

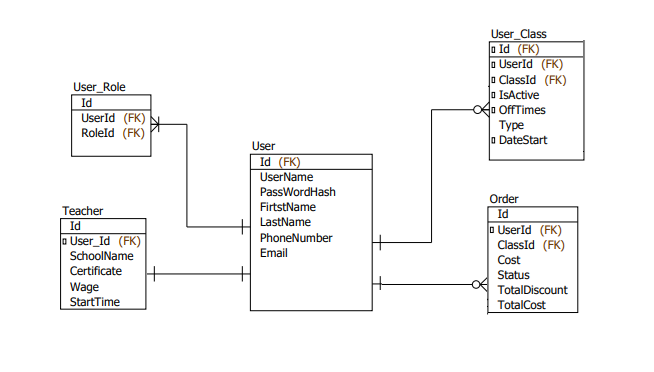
Quản lý thông tin cá nhân của người dùng.

Quản lý quyền truy cập và vai trò của người dùng khi bắt đầu đăng ký.

Đảm bảo bảo mật thông tin người dùng.

#### **2.3.2.2. Triển khai trên hệ thống**

**a) Sơ đồ ERD.**



*Hình 12: Sơ đồ ERD quản lý người dùng*

**b) Xử lý ở phía server.**

- Thêm mới người dùng:

Nhận dữ liệu từ form nhập liệu.

Kiểm tra tính hợp lệ của dữ liệu (ví dụ: email không trùng lặp, mật khẩu đúng quy ước).

Đối với user đăng ký ở web người dùng quyền mặc định sẽ là user, đối với user được thêm bởi trang admin sẽ có thể chỉ định được quyền.

Lưu thông tin người dùng vào cơ sở dữ liệu.

- Chỉnh sửa thông tin người dùng:

Nhận dữ liệu chỉnh sửa từ giao diện người dùng.

Kiểm tra tính hợp lệ của dữ liệu.

Cập nhật thông tin trong cơ sở dữ liệu.

Xóa người dùng:

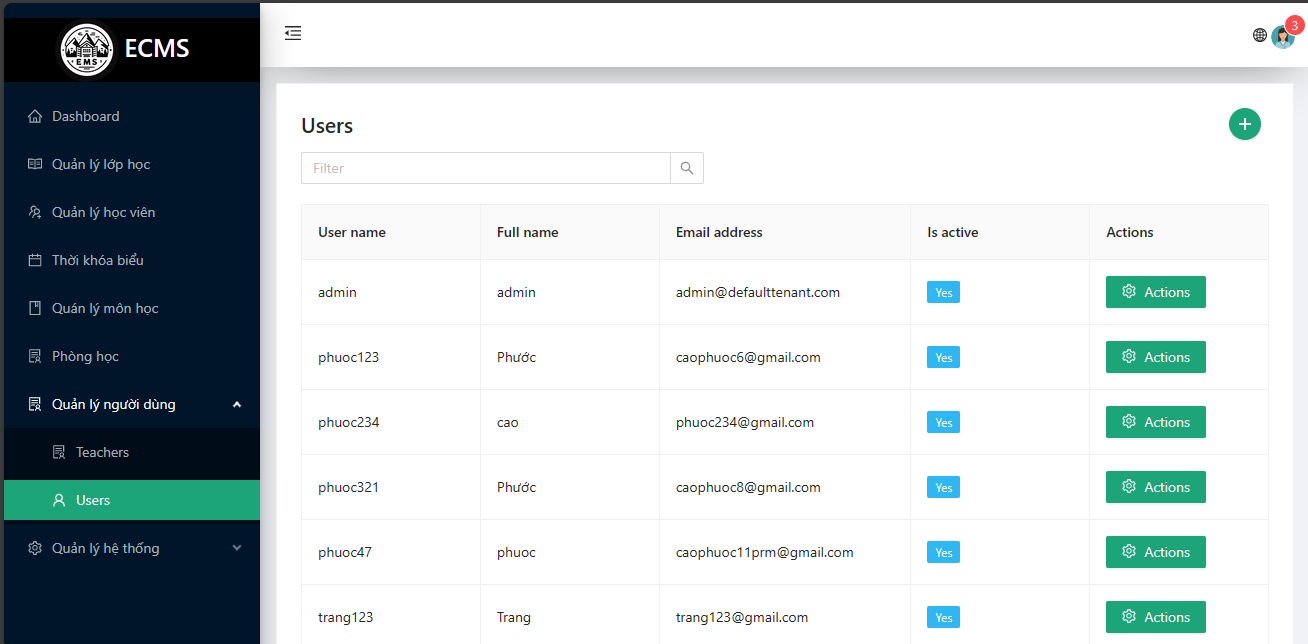
Nhận yêu cầu xóa từ giao diện người dùng.

Xóa mềm thông tin người dùng (isDelete = true).

**c) Các bước triển khai chức năng.**

- Trang quản lý Người dùng:

Danh sách người dùng: Hiển thị danh sách tất cả người dùng trong hệ thống với các thông tin cơ bản.

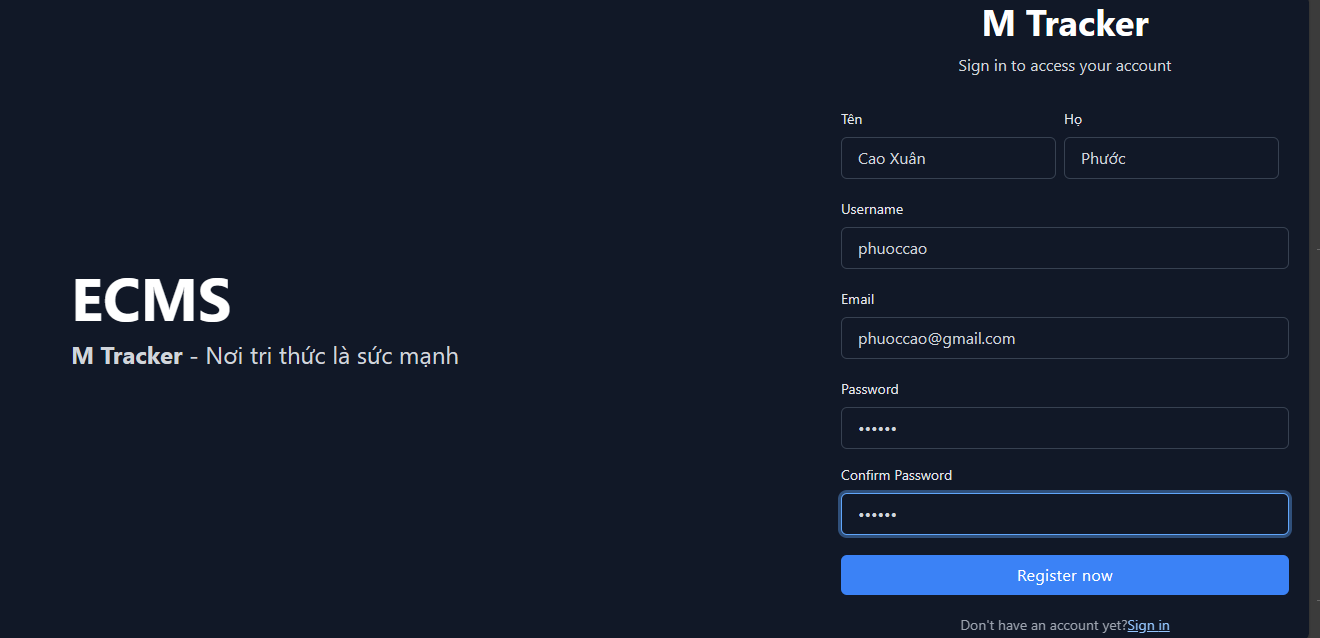


*Hình 13: Màn hình danh sách người dùng*

Thêm người dùng mới: Form nhập liệu để thêm người dùng mới bao gồm thông tin cá nhân và vai trò.

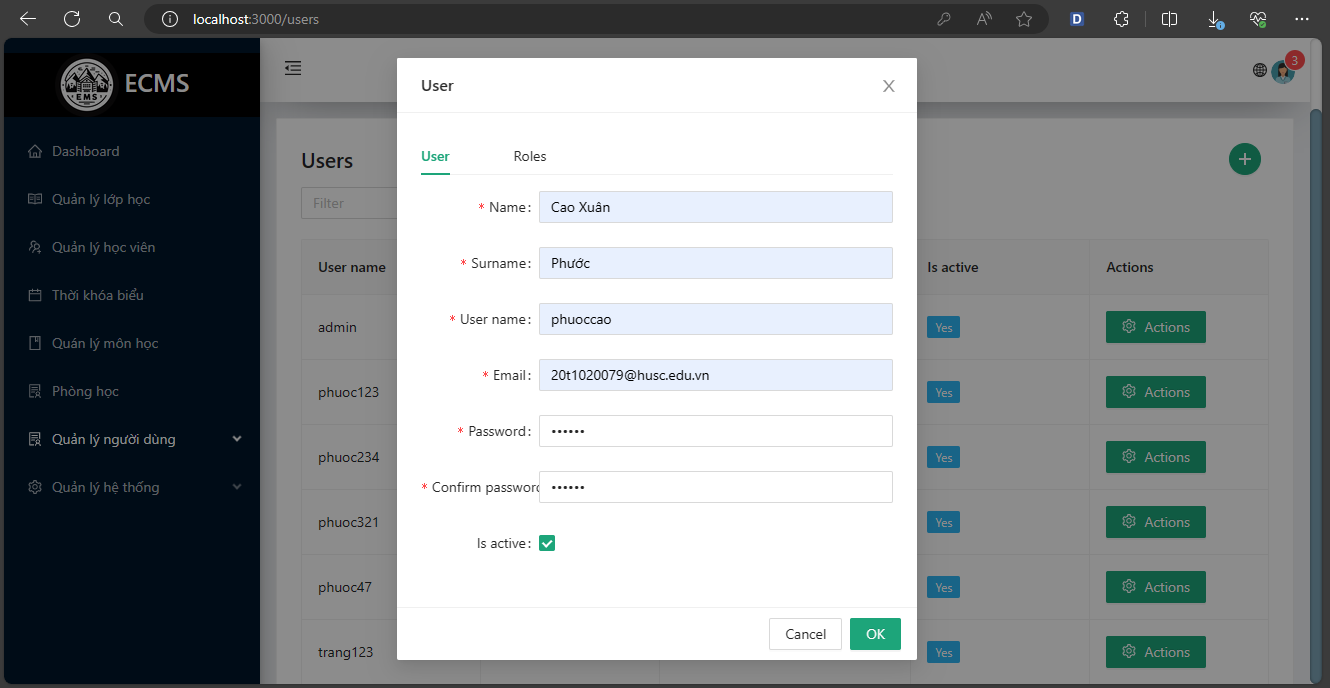
Chỉnh sửa thông tin người dùng: Form để cập nhật thông tin cá nhân và vai trò của người dùng.

Đối với người dùng:

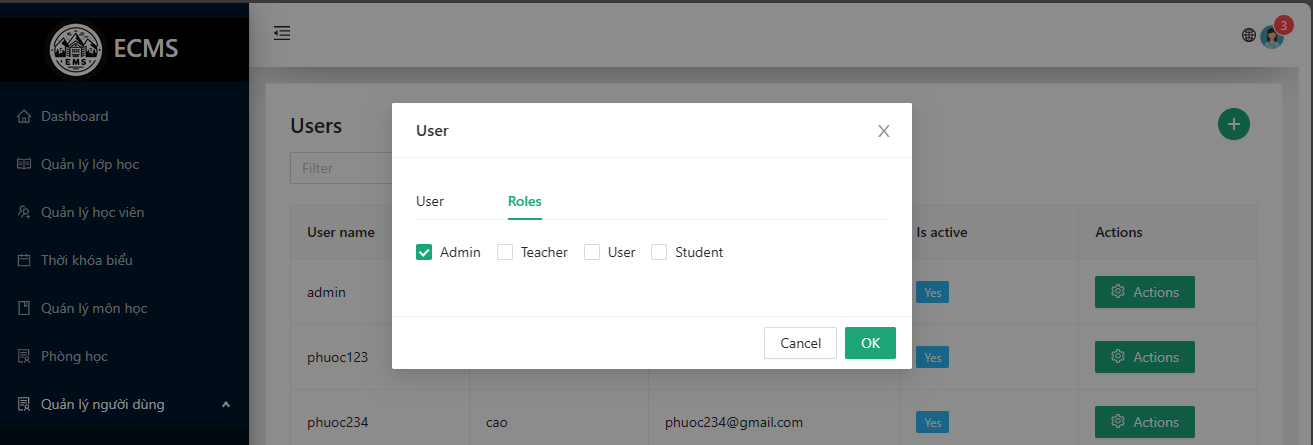


*Hình 14: Màn hình đăng ký tài khoản*

Đối với trang admin.

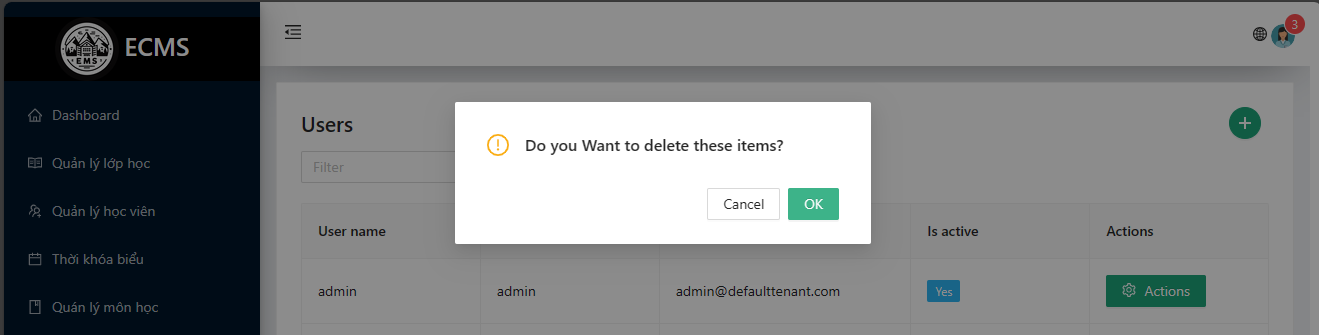


*Hình 15: Màn hình thêm người dùng - tab thông tin*



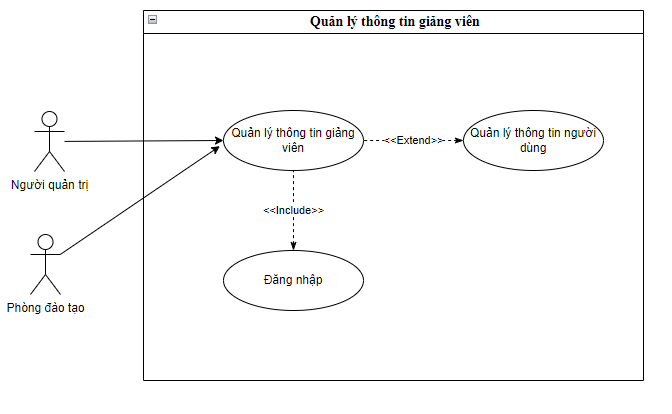
*Hình 16: Màn hình thêm người dùng - tab vai trò*

Xóa người dùng: Tùy chọn để xóa người dùng khỏi hệ thống.



*Hình 17: Màn hình xóa người dùng*

### **2.3.3. Quản lý thông tin giảng viên**



*Hình 18: Use case quản lý thông tin giảng viên*

#### **2.3.3.1. Mô tả chức năng**

**a) Giới thiệu**

Giảng viên là tác nhân quan trọng đóng vai trò trực tiếp giảng dạy học viên. Quản lý thông tin giảng viên là chức năng cần thiết trong việc quản lý, phân bố lịch giảng dạy một cách hợp lý.

**b) Mục tiêu**

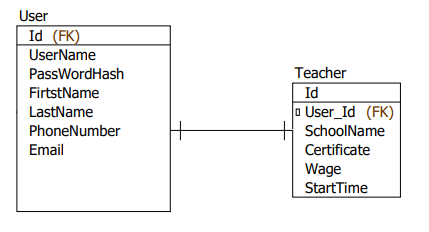
Quản lý thông tin của giảng viên.

Phân quyền truy cập và vai trò của người giảng viên đúng với lớp học được phân bố.

#### **2.3.3.2. Triển khai trên hệ thống**

**a) Sơ đồ ERD**

Giảng viên là một thực thể được kế thừa từ thực thể người dùng. Giảng viên sẽ có thêm các thông tin sau:



*Hình 19: Sơ đồ ERD quản lý giáo viên*

**b) Xử lý ở phía server.**

Hoạt động thêm, xóa, sửa thông tin giảng viên, người trực tiếp giảng dạy các lớp học. Thêm khi một User tham gia giảng dạy, sửa khi cần điều chỉnh thông tin, xóa khi không còn giảng dạy tại trung tâm.

Chỉ admin mới có quyền tác động vào thông tin giảng viên.

Đảm bảo thông tin user được nối với thông tin giảng viên. Dữ liệu giảng viên được tạo khi người đó giảng dạy.

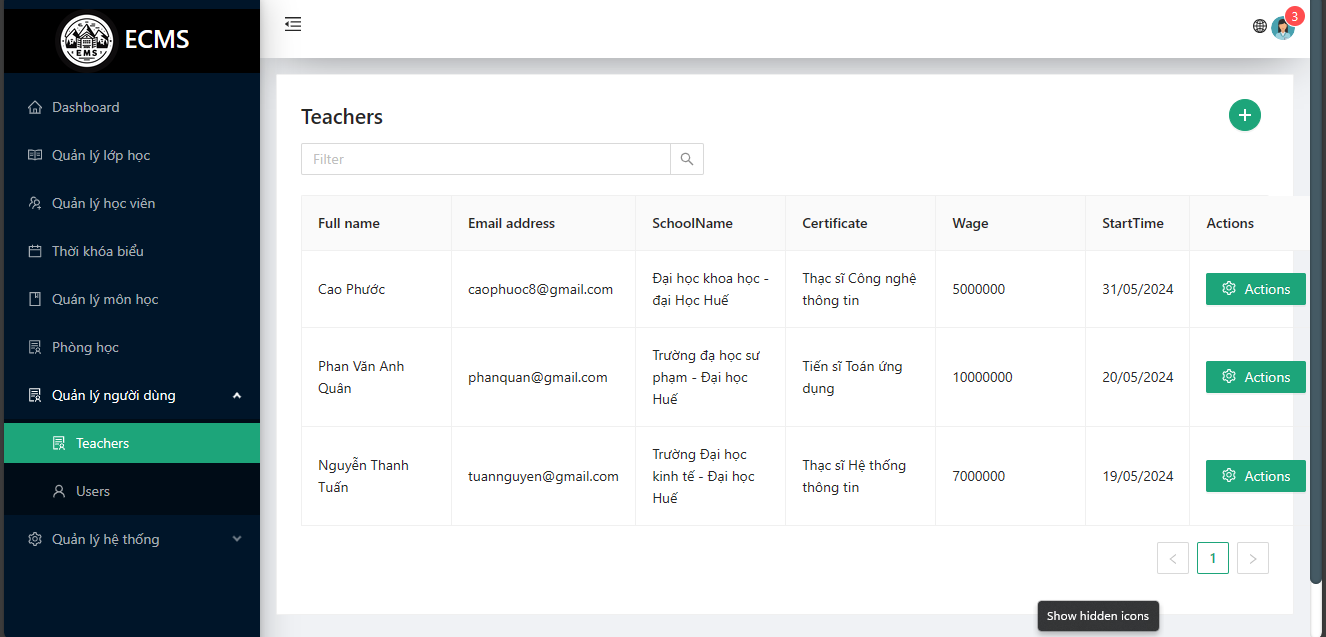
Học sinh admin có thể xem thông tin giảng viên

Học sinh chỉ xem được thông tin cơ bản bao gồm. Họ tên, bằng cấp.

Admin có thể tìm theo họ tên, bằng cấp, lương.

**c) Các bước triển khai chức năng**

- Tại giao diện admin Quản lý thông tin giảng viên.



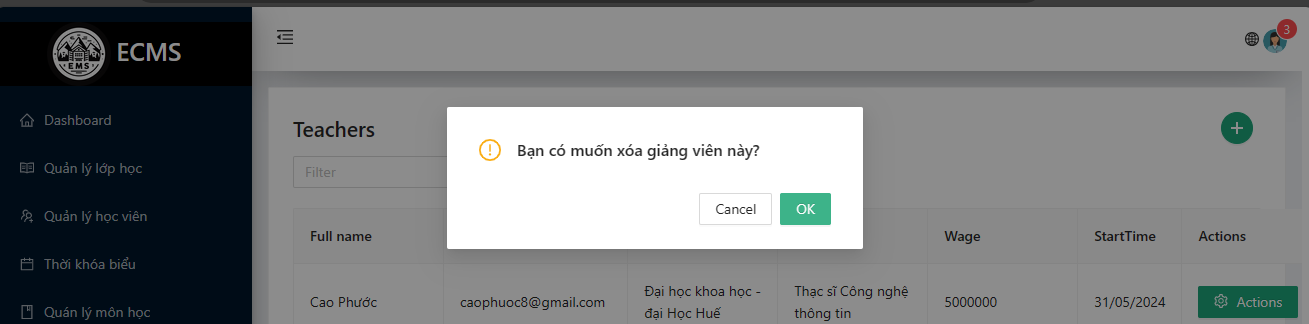
*Hình 20: Màn hình quản lý thông tin giảng viên*

- Thêm/chỉnh sửa giảng viên:



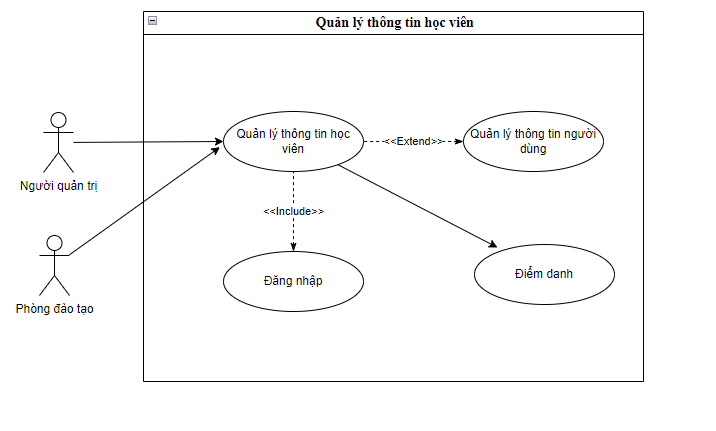
*Hình 21: Màn hình thêm/chỉnh sửa giảng viên*

Xóa giảng viên:



*Hình 22: Màn hình xóa giảng viên*

### **2.3.4. Quản lý học viên**



*Hình 23: Use case quản lý học viên.*

#### **2.3.4.1. Mô tả chức năng**

**a) Giới thiệu**

Học sinh là tác nhân chính sử dụng hệ thống, tác nhân này được kế thừa từ người dùng khi đăng ký ít nhất một khóa học của hệ thống.

**b) Mục tiêu**

- Đảm bảo thông tin học sinh được theo dõi, nhằm mục đích thông kê, tìm kiếm thông tin.

- Ngăn cách thông học sinh đối với các lớp học khác.

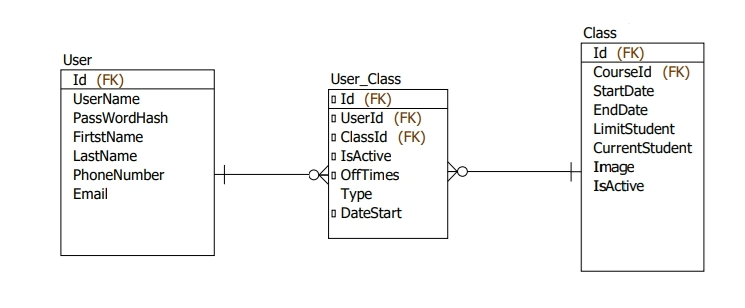
- Thông kê lại số lượng học sinh.

- Xác định được tác nhân học sinh từ đó gắn quyền phù hơp.

#### **2.3.4.2. Triển khai trên hệ thống**

**a) Sơ đồ ERD.**

Tương tự như thực thể giáo viên. Học viên cũng là một thực thể được kế thừa lại từ lớp người dùng. Đồng thời một học sinh được gắn với một lớp học.



*Hình 24: Màn hình quản lý học sinh*

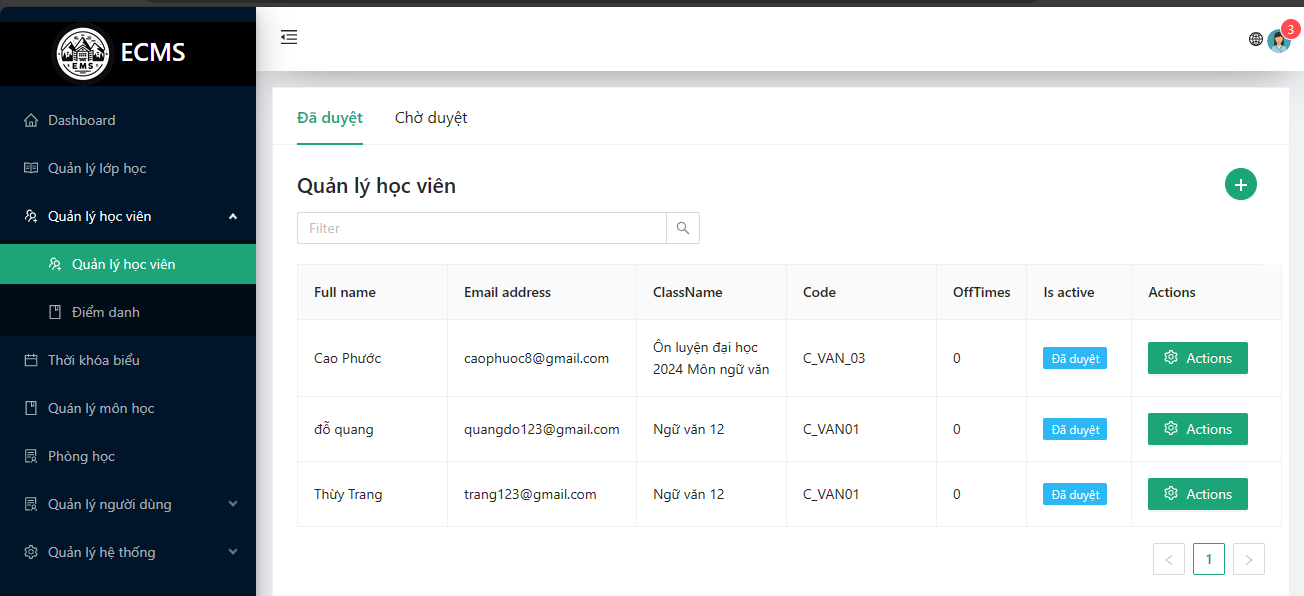
**b) Xử lý ở phía server.**

+ Đối với trường họp học sinh đăng ký trực tiếp phòng đào tạo có hành động phát sinh hóa đơn trực tiếp tại quầy. Yêu cầu bắt buộc học sinh đã có tài khoản. Nếu chưa có phòng đào tạo có thể tạo và cung cấp cho học viên tài khoản. Giá trị hóa đơn sẽ tính theo giá của khóa học mà học sinh muốn đăng ký.

+ Đối với trường hợp thanh toán các nền tảng trực tuyến như chuyển khoản, đối với ví điện tử VnPay hóa đơn sẽ được xử lý tự động. Đối với hình thứ chuyển khoản thông qua ngân hàng, phòng đào tạo tiến hành xét duyệt học viên sau khi đối chiếu với mã hóa đơn trong nội dung chuyển khoản của học viên,

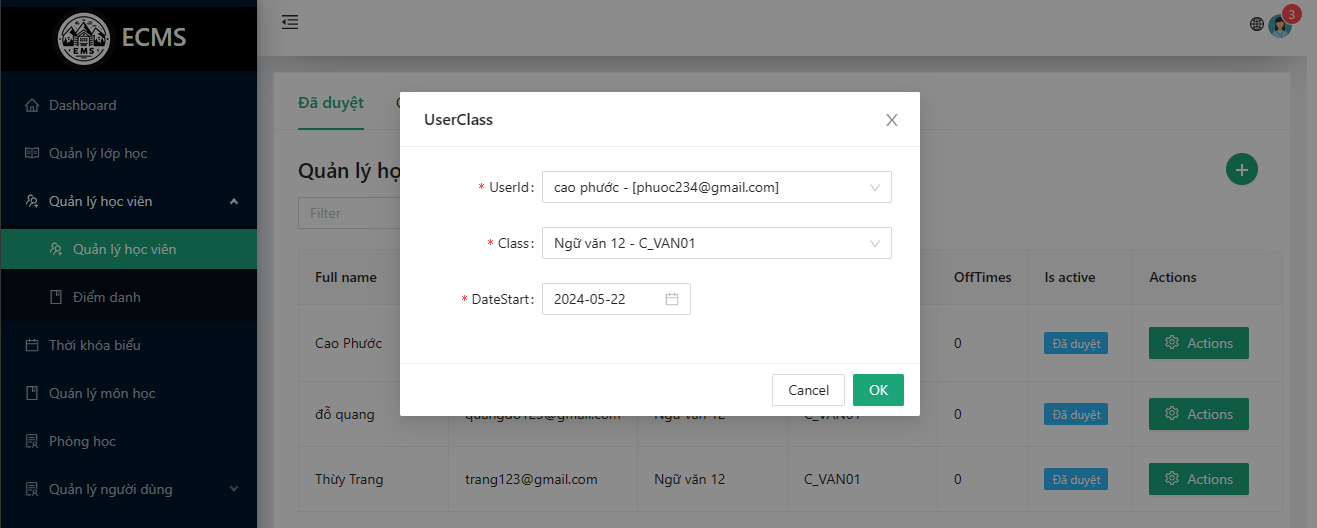
**c) Các bước triển khai chức năng**

Tại giao diện admin Quản lý thông tin học sinh.

+ Đối với học viên đăng ký trực tiếp tại trung tâm. Nhân viên phòng đào tạo trực tiếp thêm mới học viện. Điều này đồng nghĩa với việc xuất một hóa đơn ngay tại thời điểm đó dó nhân viên phòng đào tạo thu phí.  
 

*Hình 25: Màn hình quản lý học sinh*

- Thêm/chỉnh sửa học viên



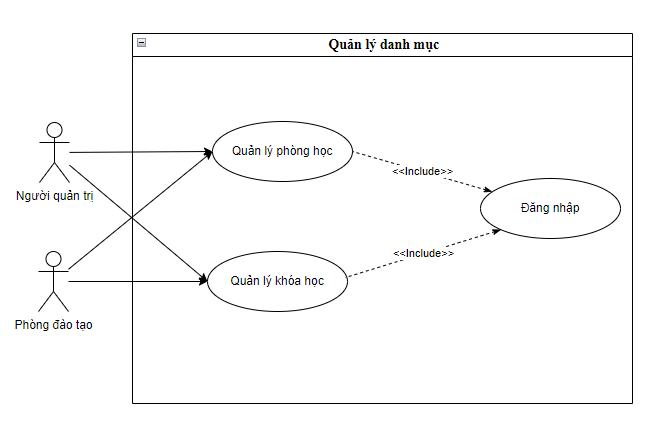
*Hình 26: Màn hình thêm/chỉnh sửa học viên*

- Xóa học viên:



*Hình 27: Màn hình xóa học viên*

### **2.3.5. Quản lý danh mục**



*Hình 28: Use case quản lý danh mục*

#### **2.3.5.1. Mô tả chức năng**

**a) Giới thiệu**

Quản lý danh mục là chức năng quản lý tìm kiếm, có vai trò quan trong trong việc vận hành. Các chức năng nằm trong quản lý danh mục bao gồm;

+ Quản lý khóa học.

+ Quản lý phòng học.

**b) Mục tiêu**

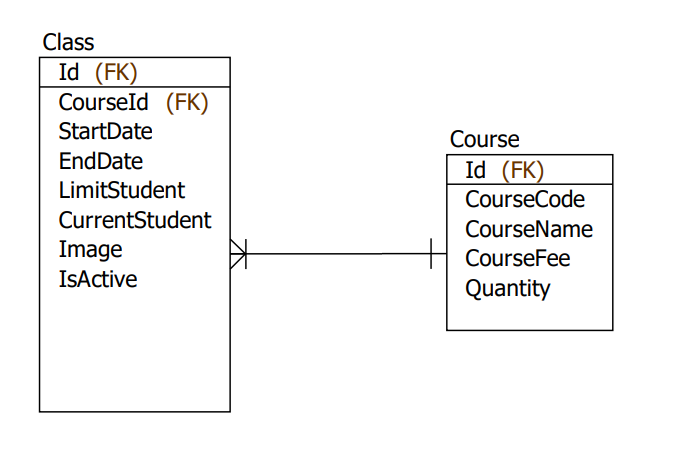
Vận hành, phân loại hệ thống.

Các nghiệp vụ khác có cơ sở để xác định. Ví dụ phòng học được tại ngày 6-5-2024 có bao nhiêu lớp sử dụng.

#### **2.3.5.2. Triển khai trên hệ thống**

**\* Quản lý khóa học**

**a. Sơ đồ ERD**



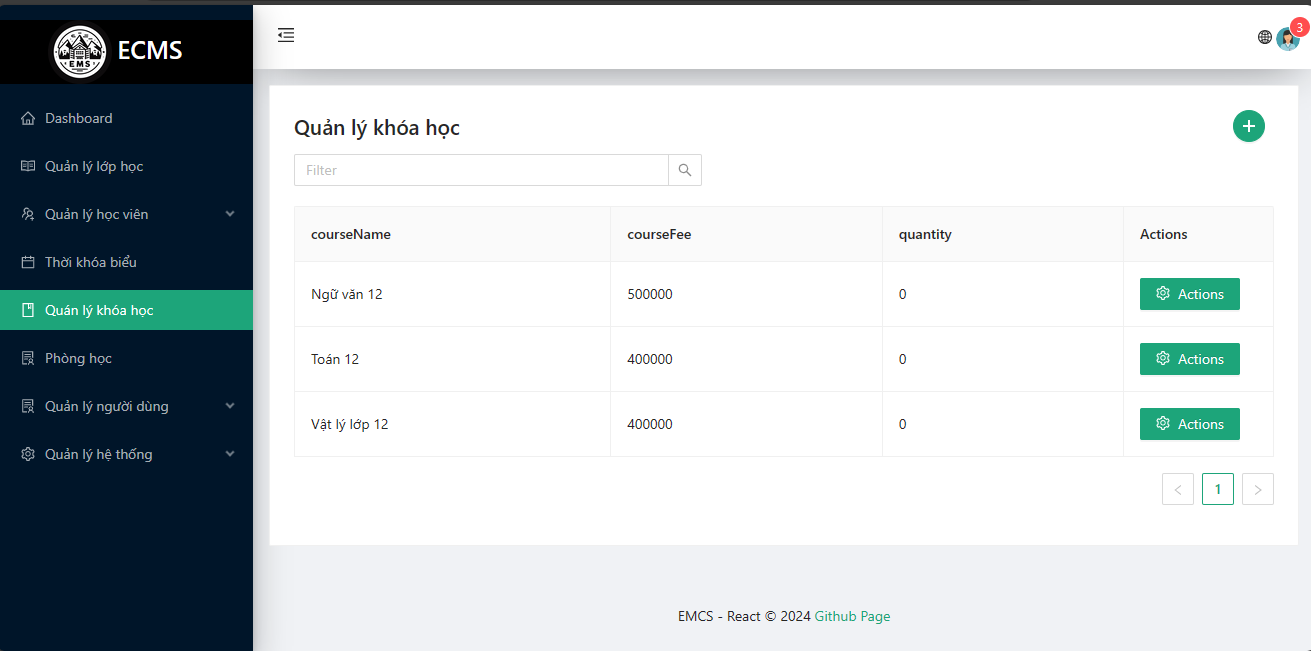
*Hình 23: Sơ đồ ERD quản lý khóa học*

**b. Xử lý ở phía server.**

Các lớp học được liên kết với lớp học nhằm mục đích phân loại. Phù hợp với nghiệp vụ của trung tâm. Với mỗi khóa học các lớp học điều có giá như nhau.

**c. Các bước triển khai chức năng**

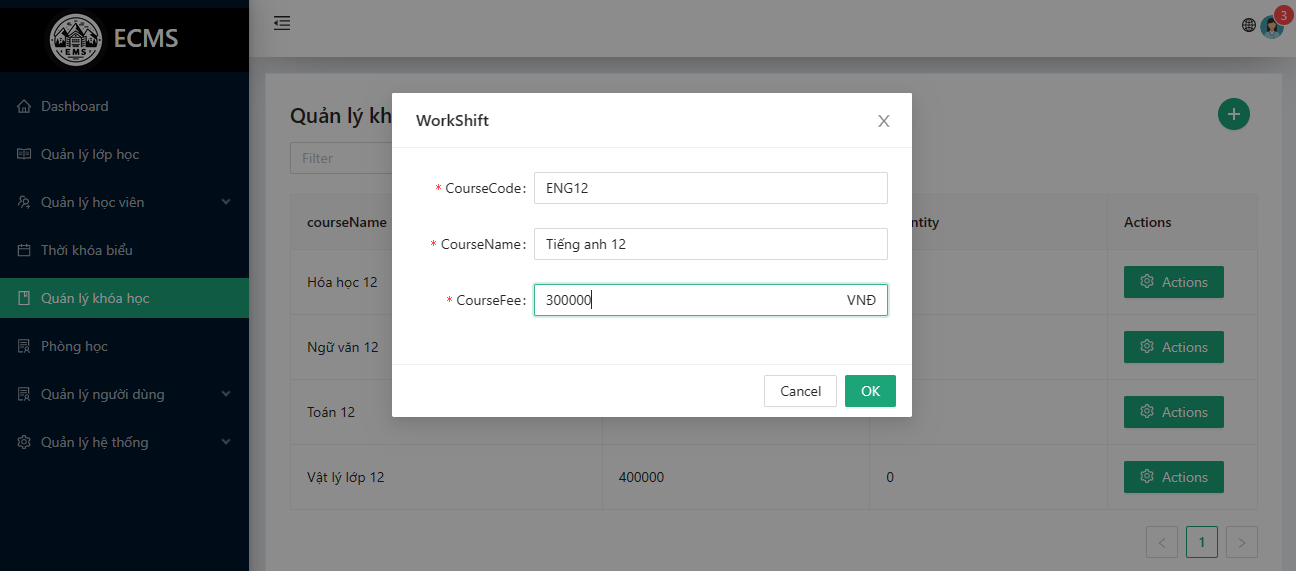
**- Tại giao diện admin Quản lý khóa học**



*Hình 29: Màn hình quản lý khóa học*

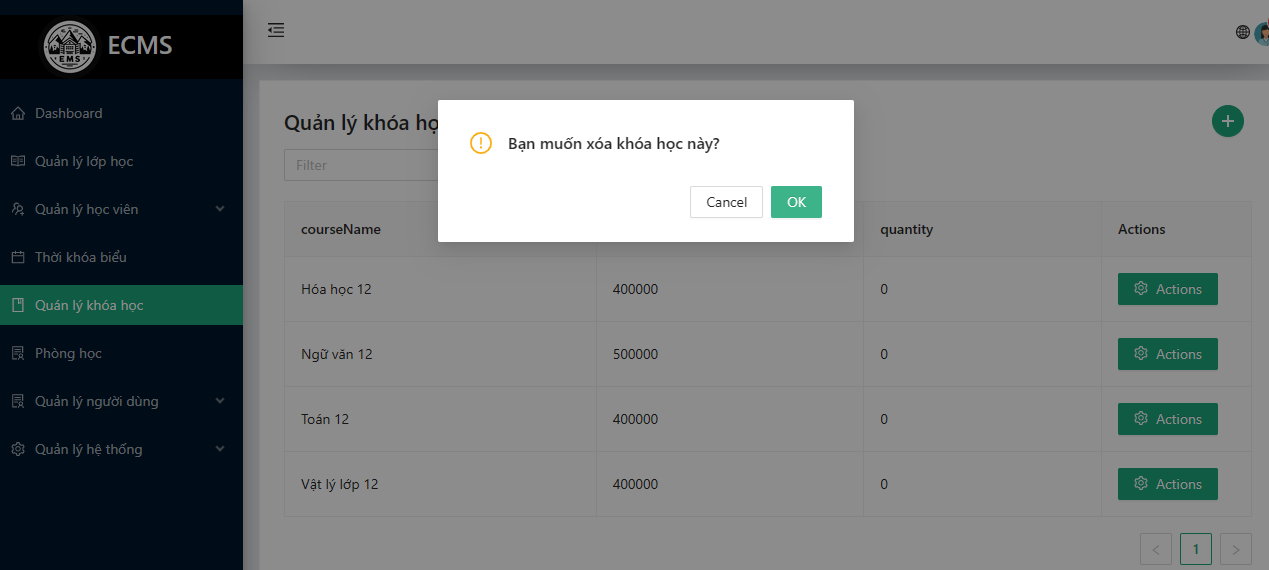
**.**

**- Thêm/chỉnh sửa :**



*Hình 30: Màn hình thêm/chỉnh sửa*

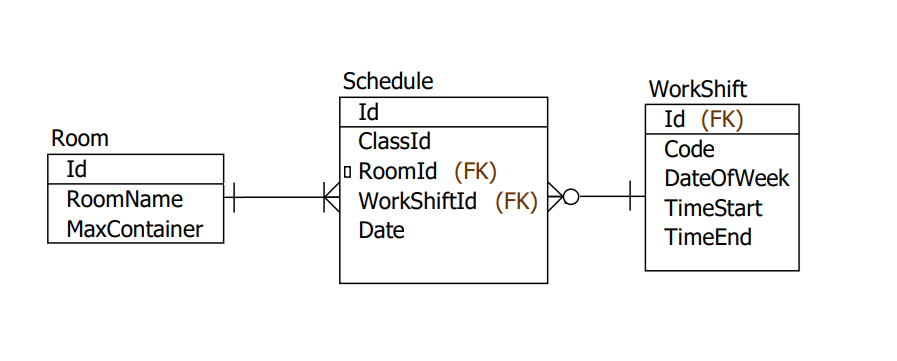
**- Xóa :**



*Hình 31: Màn hình xóa*

##### **\* Quản lý phòng học**

**a. Sơ đồ ERD**



*Hình 32: Sơ đồ ERD quản lý phòng học*

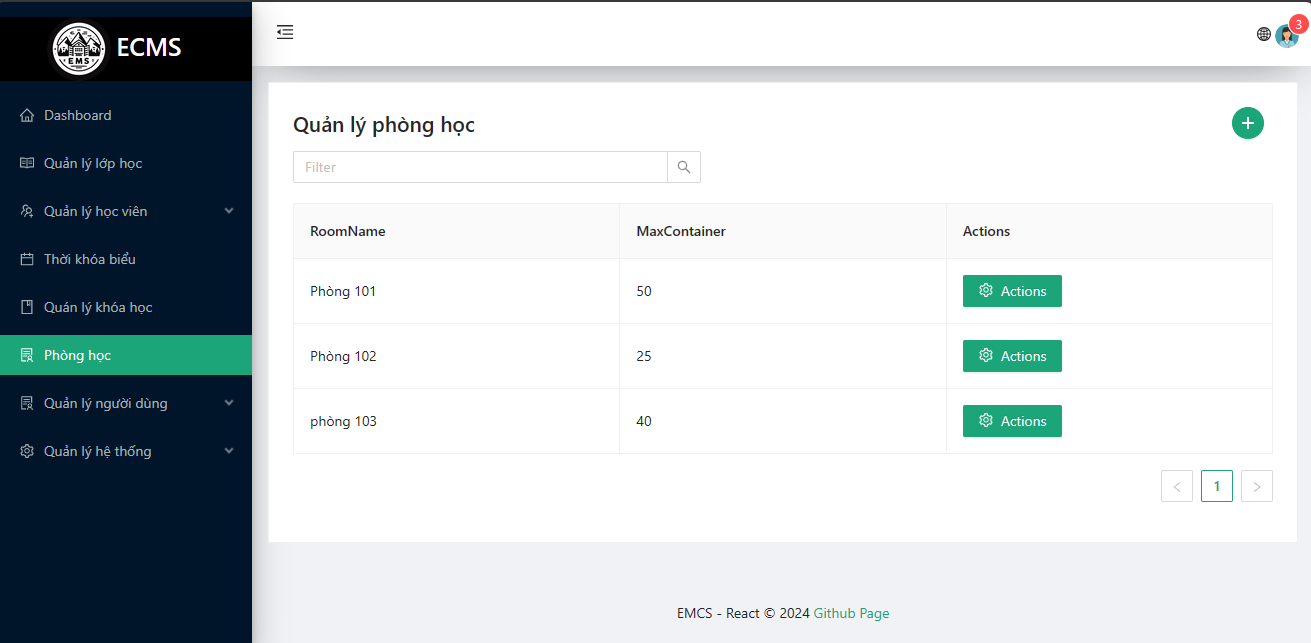
**b. Xử lý ở phía server.**

Xử lý các hành động thêm xóa sửa đối với phòng học.

Vì phòng mang tính vật lý, trạng thái của phòng sẽ thay đổi theo thời gian, khi có một lớp học đang ký một lịch học tại phòng này.

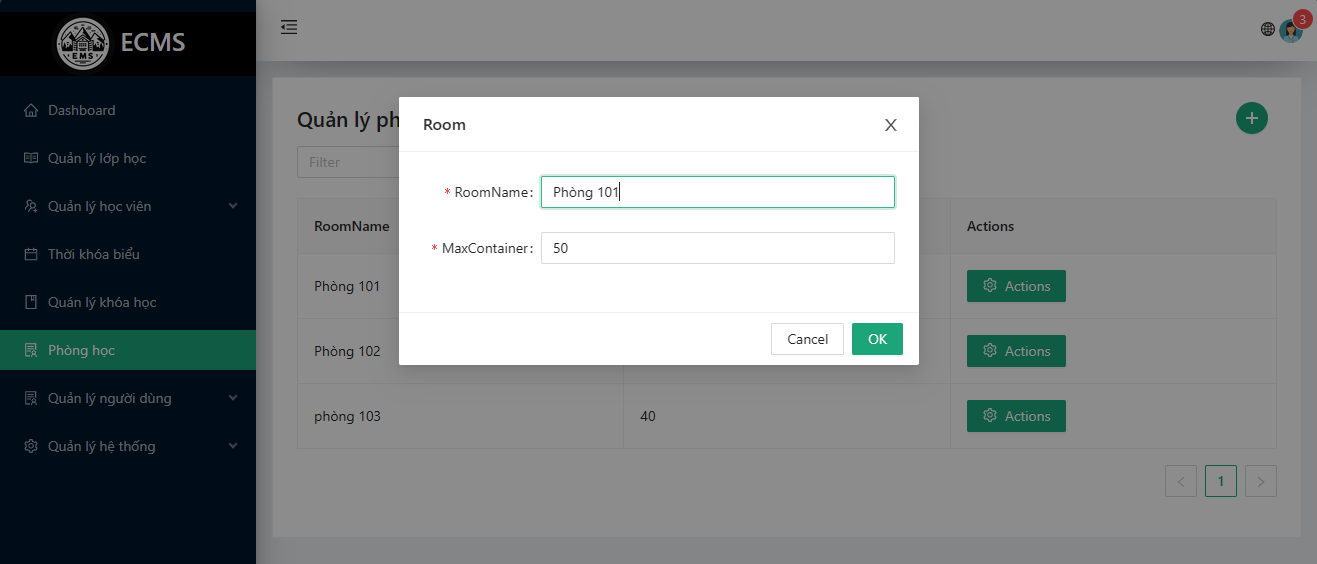
**c. Các bước triển khai chức năng**

- Tại giao diện admin Quản lý thông tin phòng học.



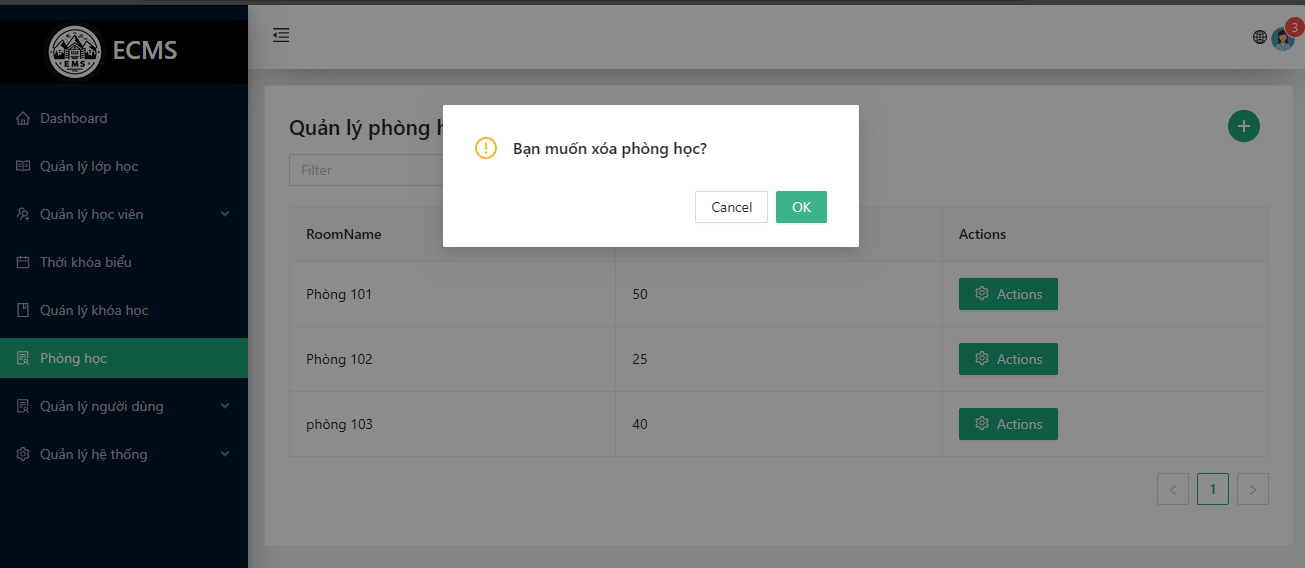
*Hình 33: Màn hình quản lý thông tin phòng học*

- Thêm/chỉnh sửa:



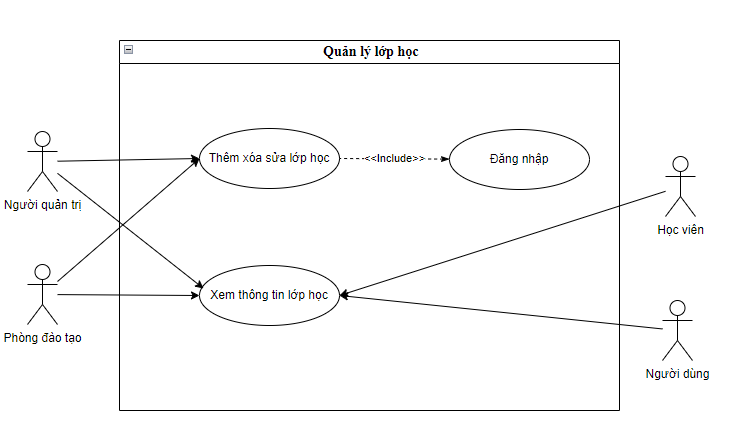
*Hình 34: Màn hình thêm/chỉnh sửa*

- Xóa :



*Hình 35: Màn hình xóa*

### **2.3.6. Quản lý lớp học**



*Hình 36: Quản lý lớp học*

#### **2.3.6.1. Mô tả chức năng**

**a. Giới thiệu**

Lớp học là thực thể chính đóng vai trò là cầu nối giữa các tác nhân trong hệ thống, khi đã là thành viên lớp học sẽ có nhiều chức năng được mở rộng

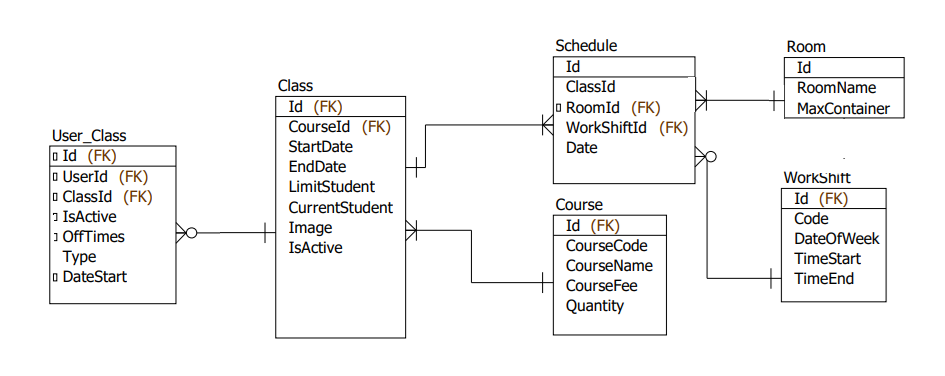
**b. Mục tiêu**

Xác định được người dùng là học sinh

Xác định người dùng là giảng viên

Ngăn chặn truy cập trái phép vào lớp học

#### **2.3.6.2. Triển khai trên hệ thống**

**a. Sơ đồ ERD**

*Hình 37: Sơ đồ ERD quản lý lớp học*

**b. Xử lý ở phía server.**

Lớp học được tạo ra khi thõa mãn các điều kiện, tồn tại phòng học, môn học, giảng viên, lịch học.

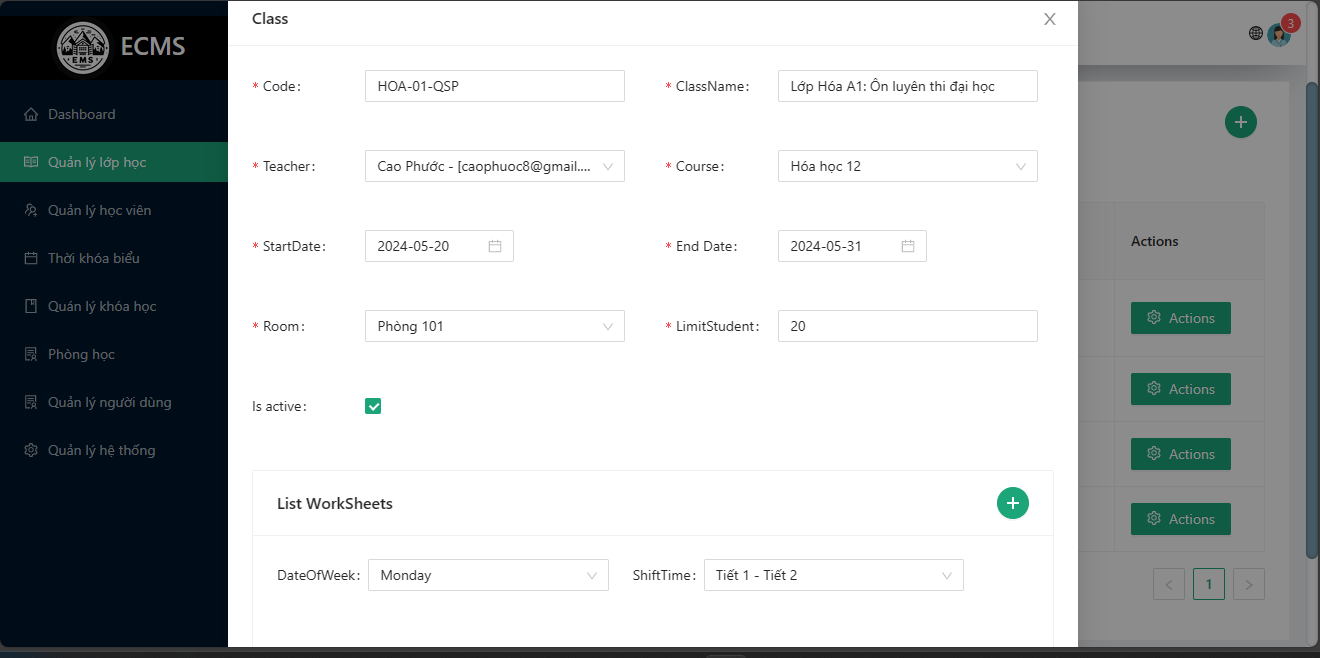
Học sinh là các tác nhân tương tác với lớp học thông qua user\_class để xác định được các tác nhân đóng vài trò gì trong đó (học sinh, giảng viên)

**c. Các bước triển khai chức năng**

- Tại giao diện admin Quản lý thông tin lớp học.

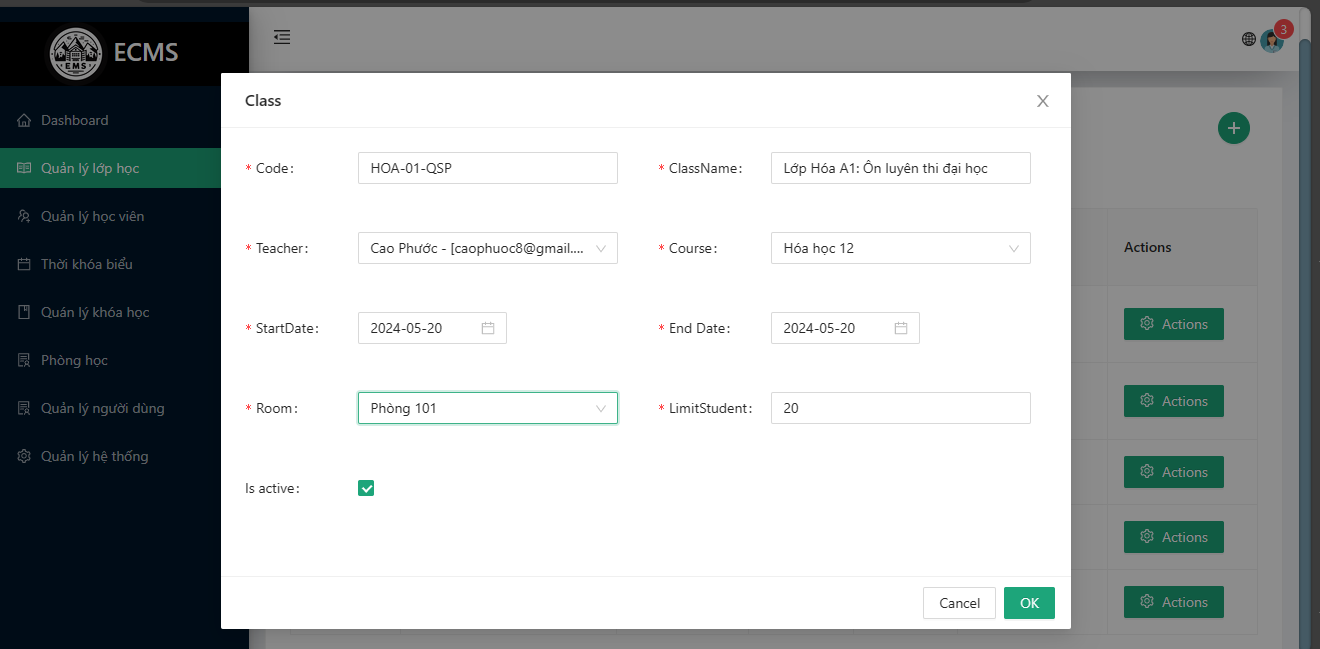
- Thêm/chỉnh sữa:

Giao diện thêm lớp học



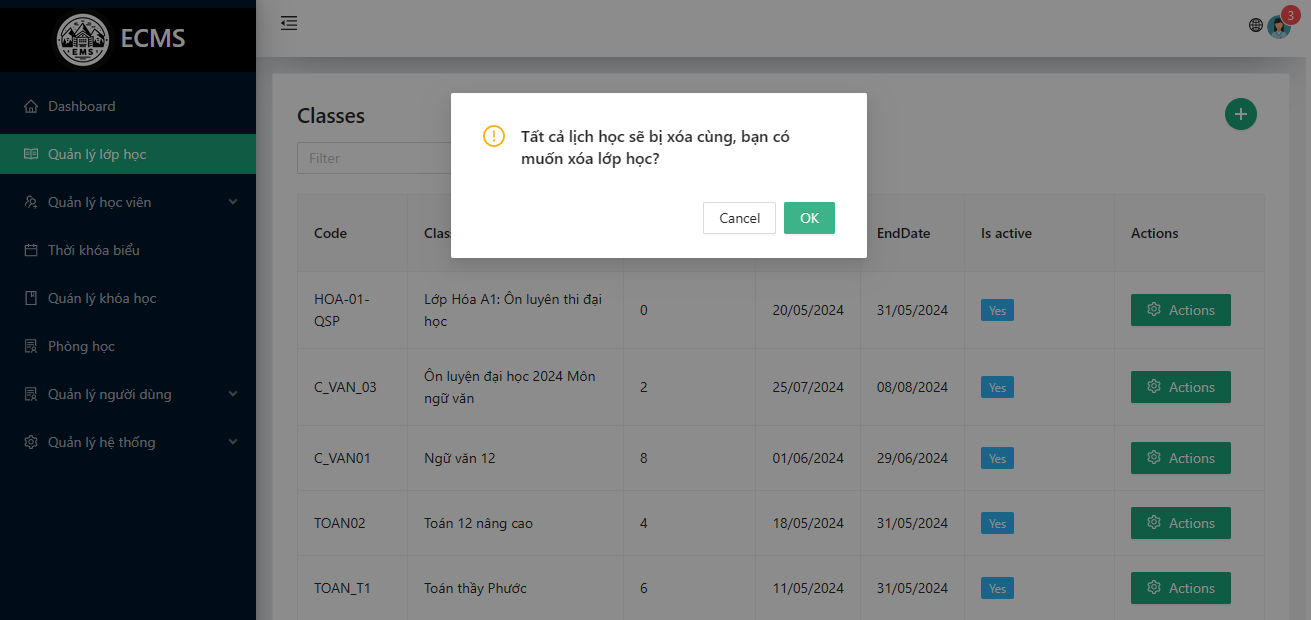
*Hình 38: Màn hình thêm lớp học*

Giao diện sửa lớp học



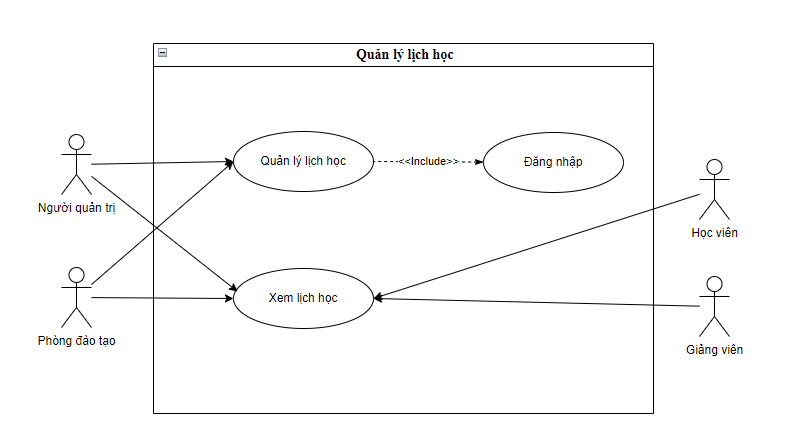
*Hình 39: Màn hình sửa lớp học*

- Xóa :



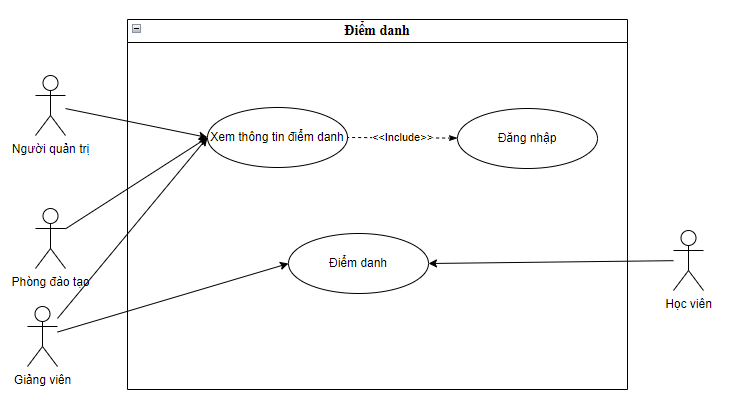
*Hình 40: Màn hình xóa lớp học*

### **2.3.7. Quản lý lịch học**



*Hình 41: Use case quản lý lịch học*

#### **2.3.7.1. Mô tả chức năng**



**a. Giới thiệu**

Lịch học là chức nhằm giúp người quản lý nắm rõ thời gian biểu giảng dạy của mỗi lớp học. Giúp việc quản lý, vận hành các lớp học trở nên trơn tru hơn nhất là khi có các sự cố đột xuất khô thể giảng dạy.

**b. Mục tiêu**

Tạo được lịch học tự động khi tạo một lớp học.

Các lịch học phải đảm bảo không được trừng phòng tại cùng một khung giờ.

Đảm bảo thêm xóa lịch học trong một khoảng thời gian.

#### **2.3.7.2. Triển khai trên hệ thống**

**a. Sơ đồ ERD**

A diagram of a data flow

Description automatically generated

*Hình 42: Sơ đồ ERD quản lý lịch học*

**b. Xử lý ở phía server.**

Lớp học được tạo ra một lịch học cũng được tạo ra đồng thời, kéo dài từ ngày bắt đầu đến ngày kết thúc

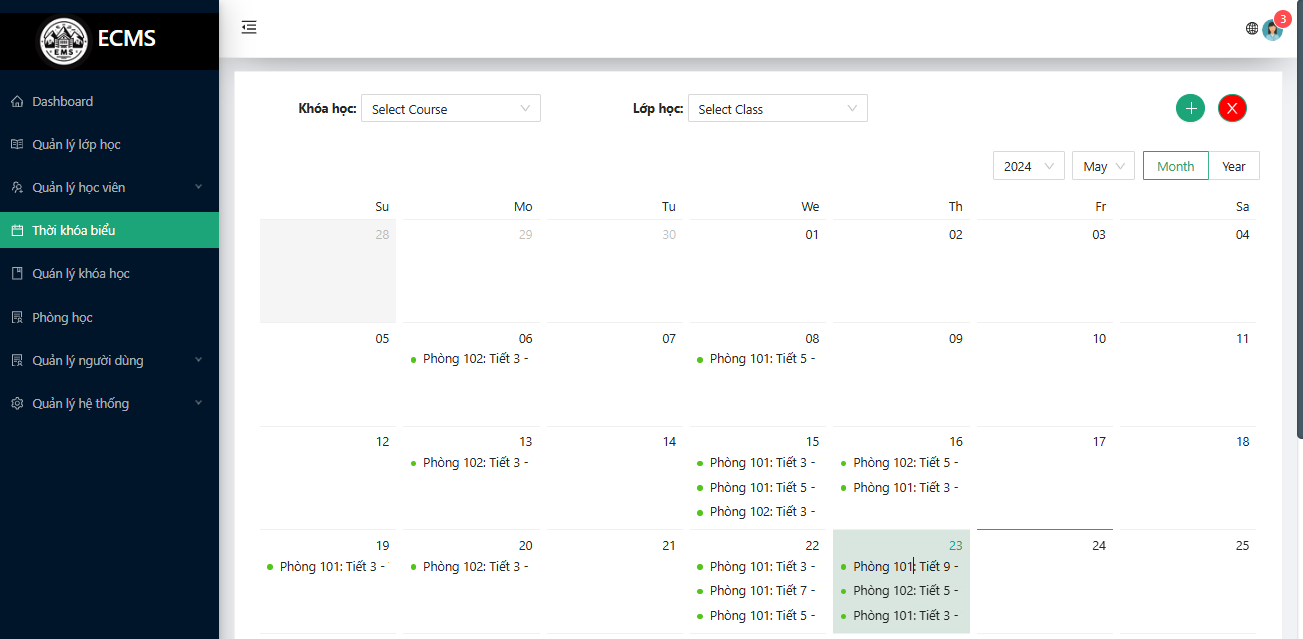
Trong suốt quá trình tạo lịch học luôn phải đảm bảo tính nhất quán về phòng học, tại một phòng học trong cùng một khung thời gian chỉ có duy nhất một lớp học.

Cho phép tạo lịch học hàng loạt dựa trên khoảng thời gian.

Cho phép xóa lịch học hàng loạt dựa trên khoảng thời gian.

**c. Các bước triển khai chức năng**

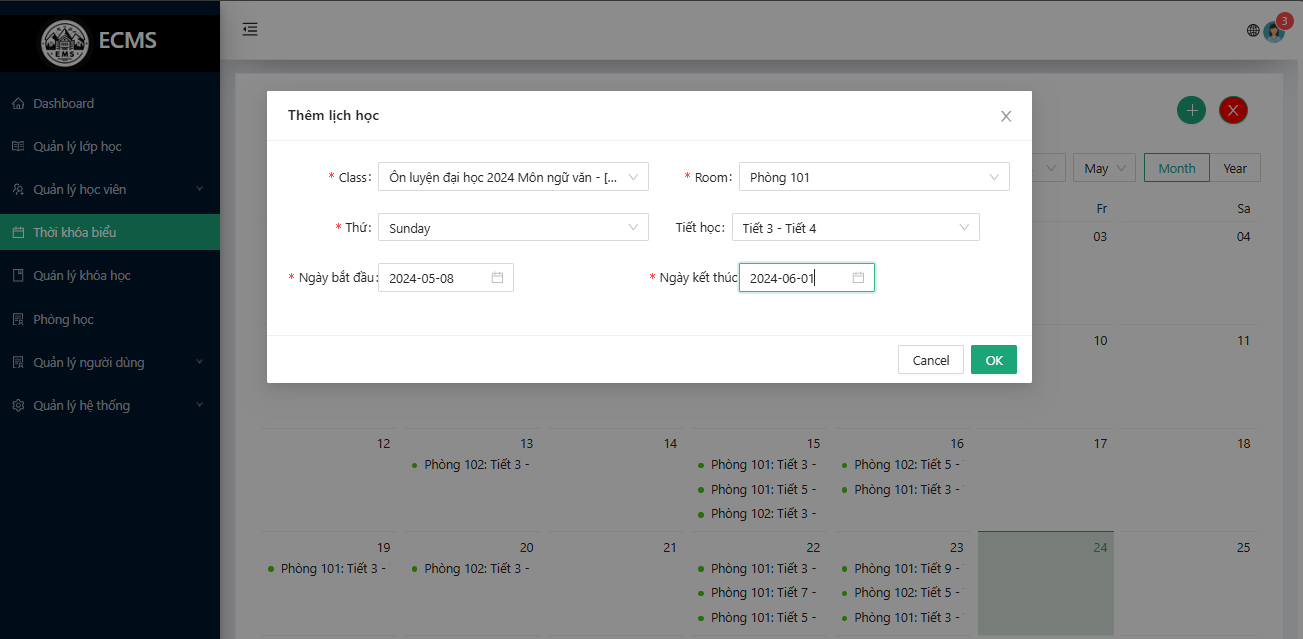
- Tại giao diện admin Quản lý thông tin lịch học được lọc theo khóa học và lớp học.



*Hình 43: Màn hình quản lý thông tin lịch học*

- Thêm/chỉnh sửa:

Giao diện thêm lích học theo khoảng thời gian.



*Hình 44: Màn hình thêm/chỉnh sửa lịch học*

Giao diện xóa lịch học theo khoảng thời gian.

A screenshot of a computer

Description automatically generated

*Hình 45: Màn hình xóa lịch học*

### **2.3.8. Điểm danh**

#### **2.3.8.1. Mô tả chức năng**

**a. Giới thiệu**

Điểm danh nhằm mục đích theo dõi quá trinh học tập của học viên. Học viên đến lớp phải điểm danh băng QR code đã được cấp phát khi tạo tài khoản. Thông tin được lưu trữ nhằm mục đích đánh giá thái độ học tập. Đồng thời tạo các chương trình thi đua như “Đi học đúng giờ, Nhận quà hấp dẫn” nhằm gia tăng sư hứng thu của học viên.

**b. Mục tiêu**

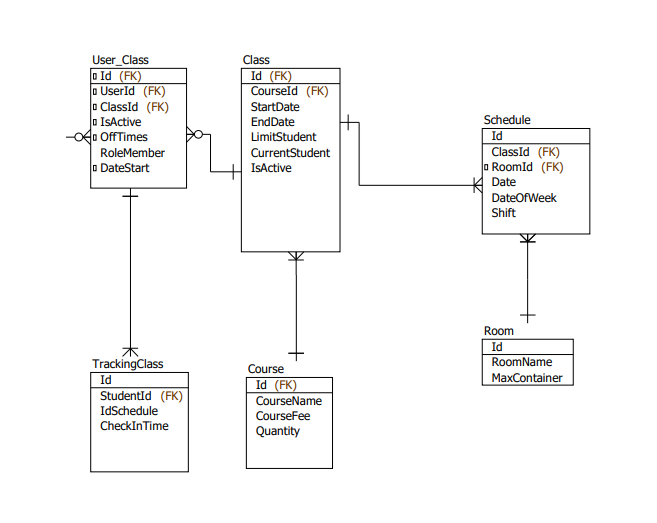
- Theo dõi được quá trình học tập của học viên.

- Thuật toán kiểm tra giờ đến lớp hoạt động chính xác.

- Thông kê, đối chiếu qua các tháng, quý.

#### **2.2.8.2. Triển khai trên hệ thống**

**a. Sơ đồ ERD**



*Hình 46: Sơ đồ ERD điểm danh*

**b) Xử lý ở phía server.**

QR code được cấp cho người dùng là một chuỗi chứa thông tin người dùng đã được mã hóa.

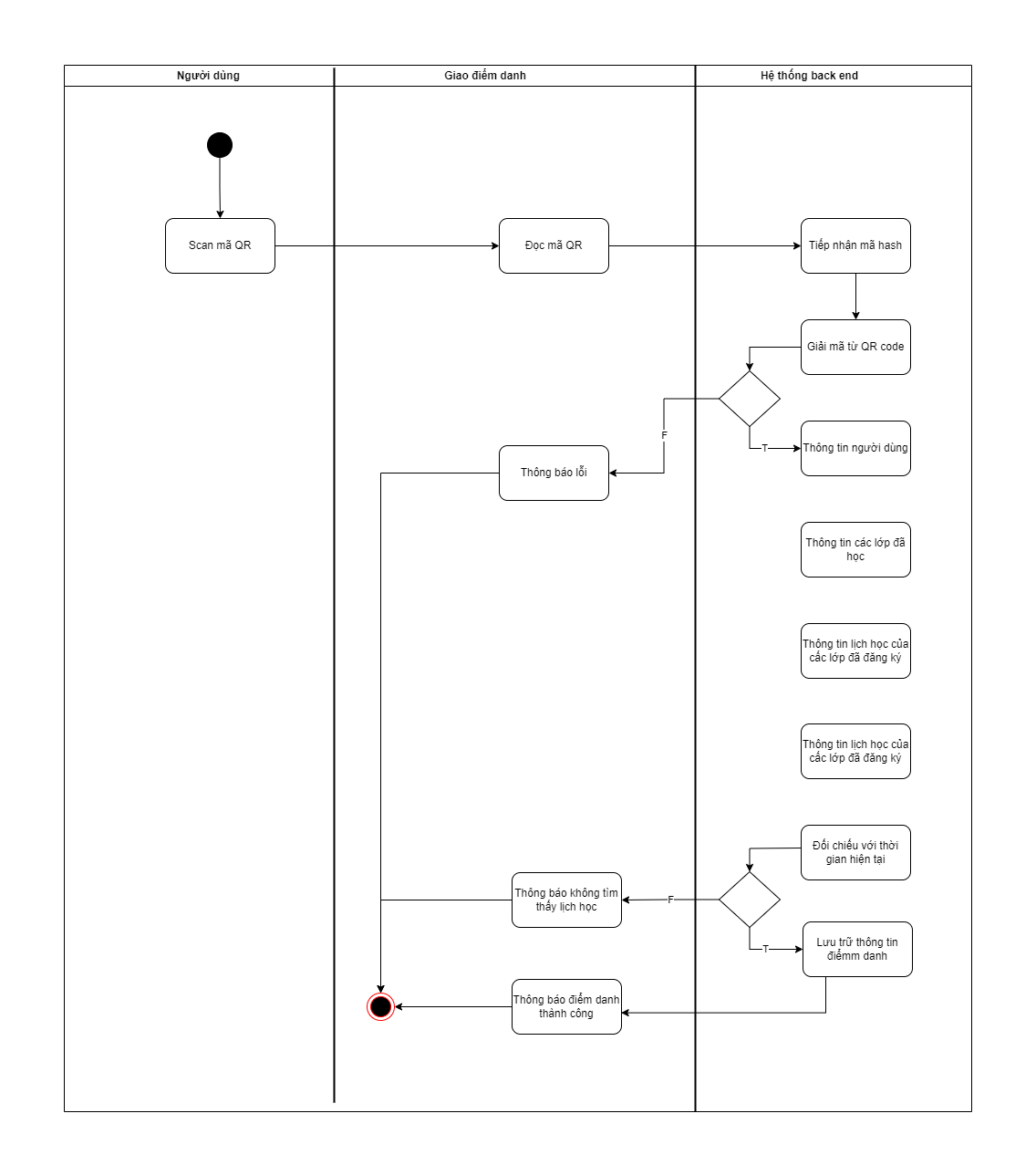
Khi quét QR hệ thống sẽ nhận diện và gửi chuổi mã hóa về server thông qua API, Sau khi tiến hành xác minh hệ thống sẽ lưu trữ lần điểm danh này cho học sinh.

Một học sinh được điểm danh cần đảm bảo về các tiếu chí sau:

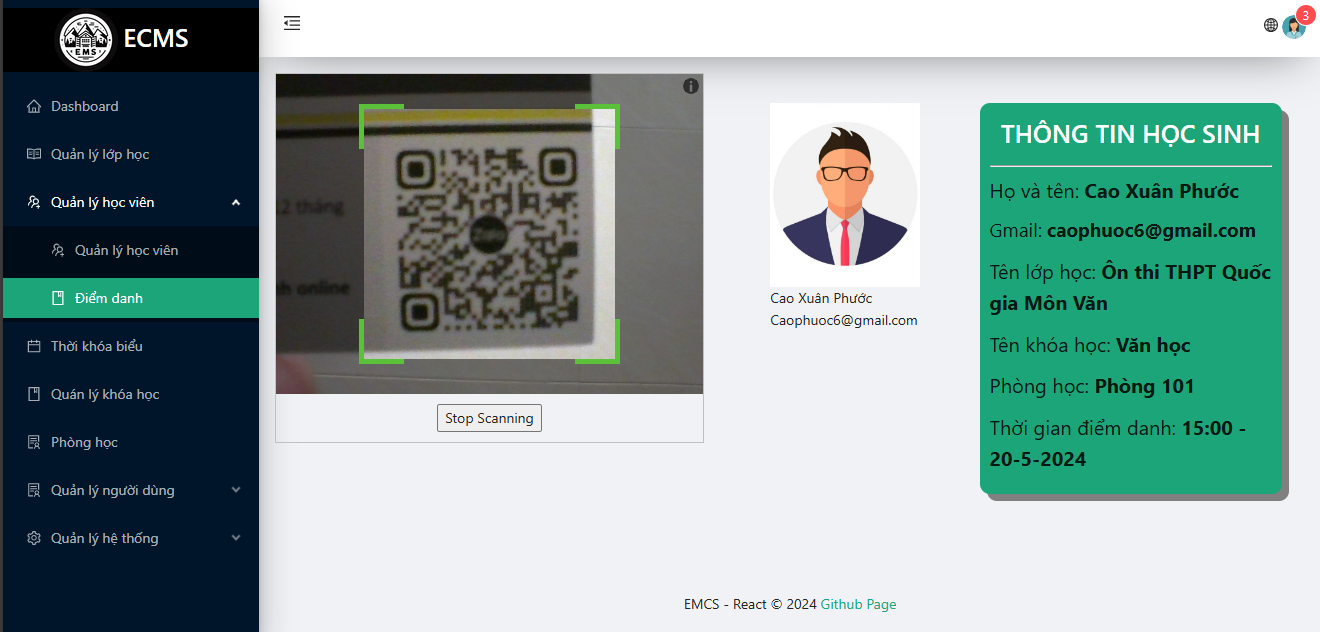
- Thời gian điểm danh bắt đầu từ khi buổi học diễn ra trước 30p đến khi buổi học kết thúc.

- Học sinh phải đảm bảo, đã đăng ký lớp học thì mới có thể điểm danh vào dúng lớp.

- Trường hợp tìm thấy học sinh có 2 lịch học bị trùng nhau thì sẽ lấy lớp học được đăng ký gần nhất để ghi nhận.

****

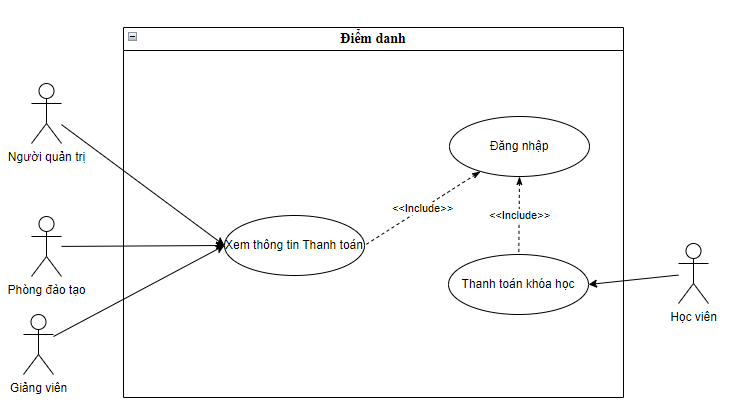
*Hình 47: Biểu dò hoạt động điểm danh*



*Hình 48: Màn hình điểm danh*

### **2.3.8. Thanh toán**

#### **2.3.8.1. Mô tả chức năng**



*Hình 49: Use case thanh toán*

**a. Giới thiệu**

Thanh toán là chức năng quan trọng đòi hỏi tính chính xác cao. Nhằm giúp tối ưu quy trình thanh toán giúp nghiệp vụ trở nên có quy trình và vận hành nhanh chống

**b. Mục tiêu**

- Phát sinh được hóa đơn.

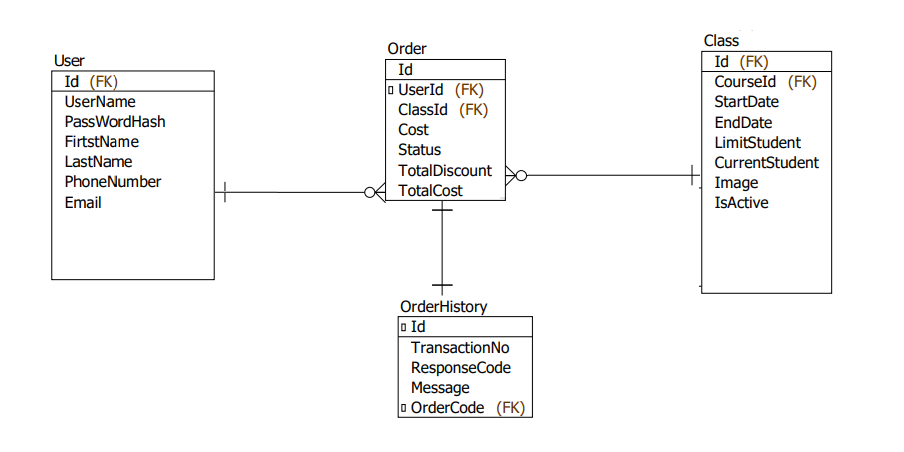
- Lưu trữ thông tin hóa đơn

- Lưu trữ lịch sử thanh toán, các trạng thái.

- Có cơ chế truy xuất đối với các bên thứ 3 cung cấp dịch vụ thanh toán.

#### **2.3.8.2. Triển khai trên hệ thống**

**a. Sơ đồ ERD**



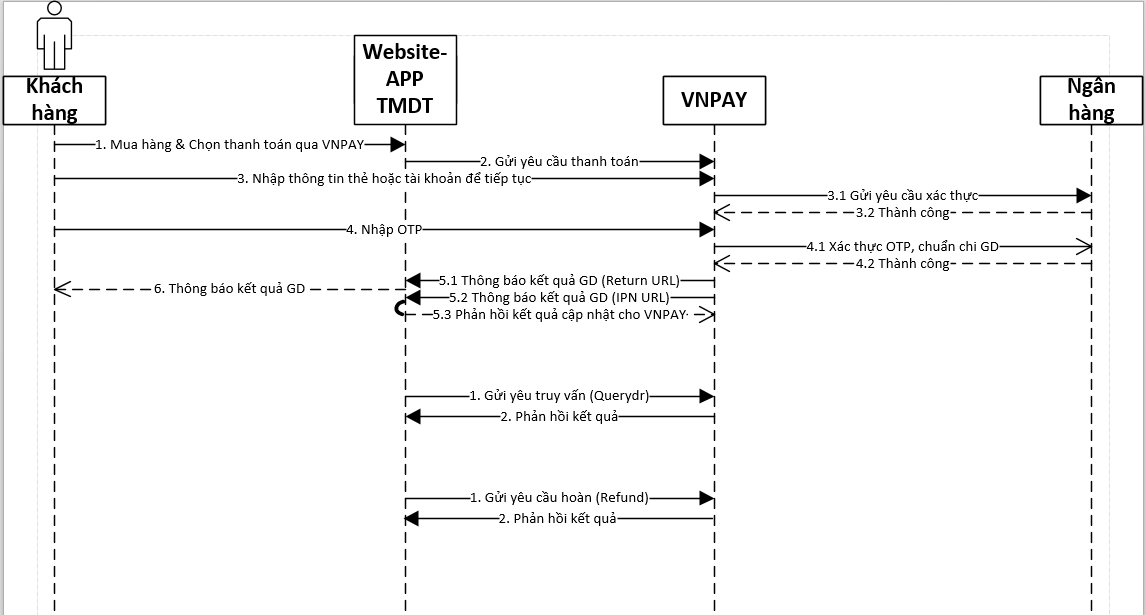
*Hình 50: Sơ đồ ERD thanh toán*

**b) Xử lý ở phía server.**

Phương thức thanh toán có 2 dạng:

* Trực tiếp: khi học viên đến đăng ký học tại quầy nhân viên phòng đạo tạo tiến hành phát sinh hóa đơn cho học viên sau khi thanh toán học viên được thêm vào lớp học đã đăng ký.
* Trực tuyến: Khi hoc viên đăng ký online. Hóa đơn được cấp phát ngay khi có yêu cầu với một cờ quy định là pending, Sau khi học viên thao tác với phía cung cấp dịch vụ thanh toán nhà cung cấp dịch vụ sẽ gửi lại request phản hồi về giao dịch với phía service của hệ thống.

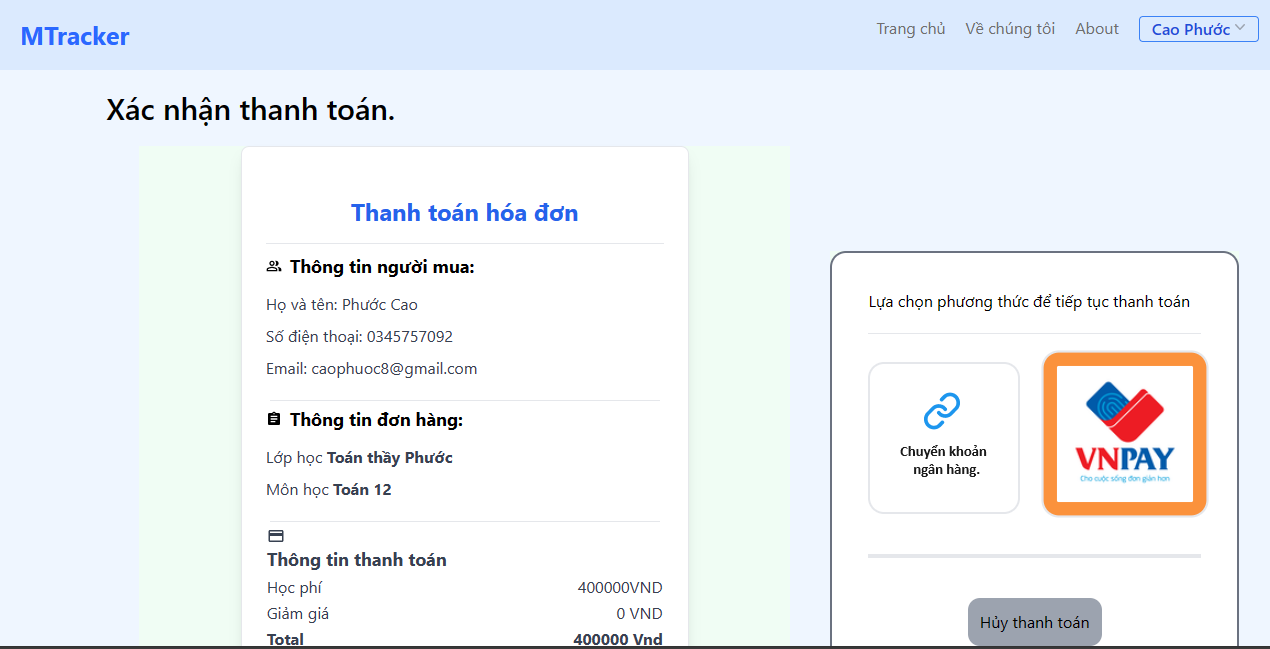
Hệ thống dựa vào các thông tin có được tiến hành cập nhật lại trạng thái thành công hay thất bại đồng thời mọi phát sinh giao dịch điều được ghi lại.



*Hình 51: Sơ đồ hoạt động thanh toán VNPay*

**c) Các bước triển khai chức năng**

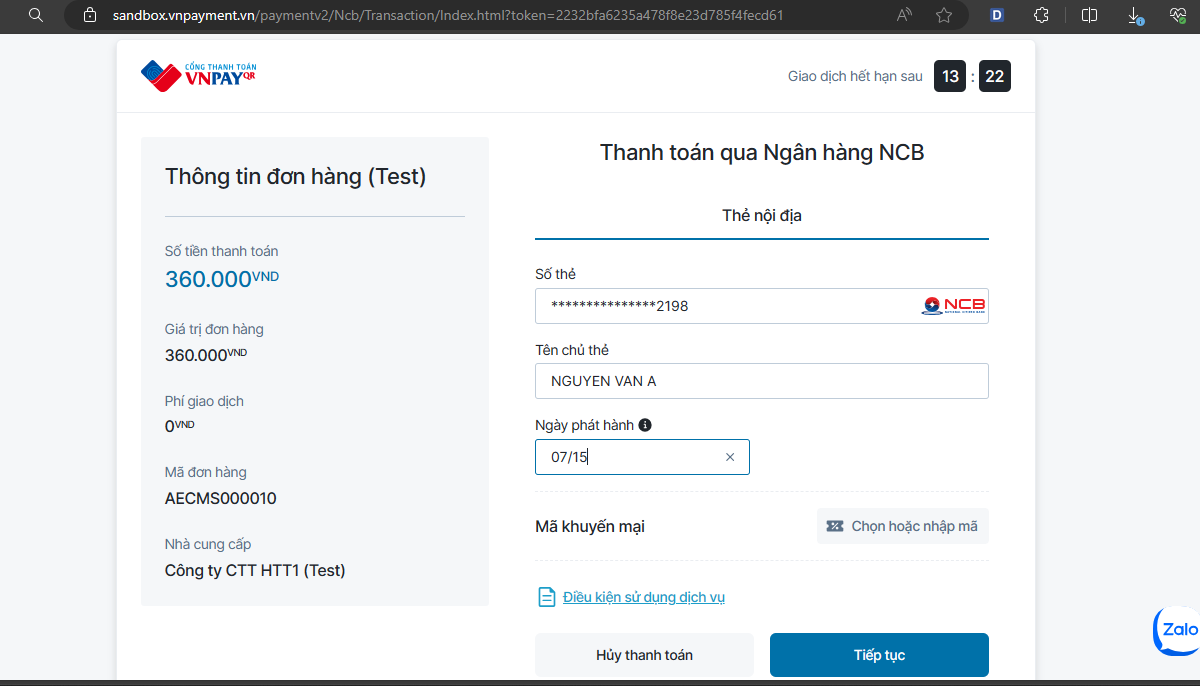
**Chọn phương thức thanh toán trực tuyến VnPAy**



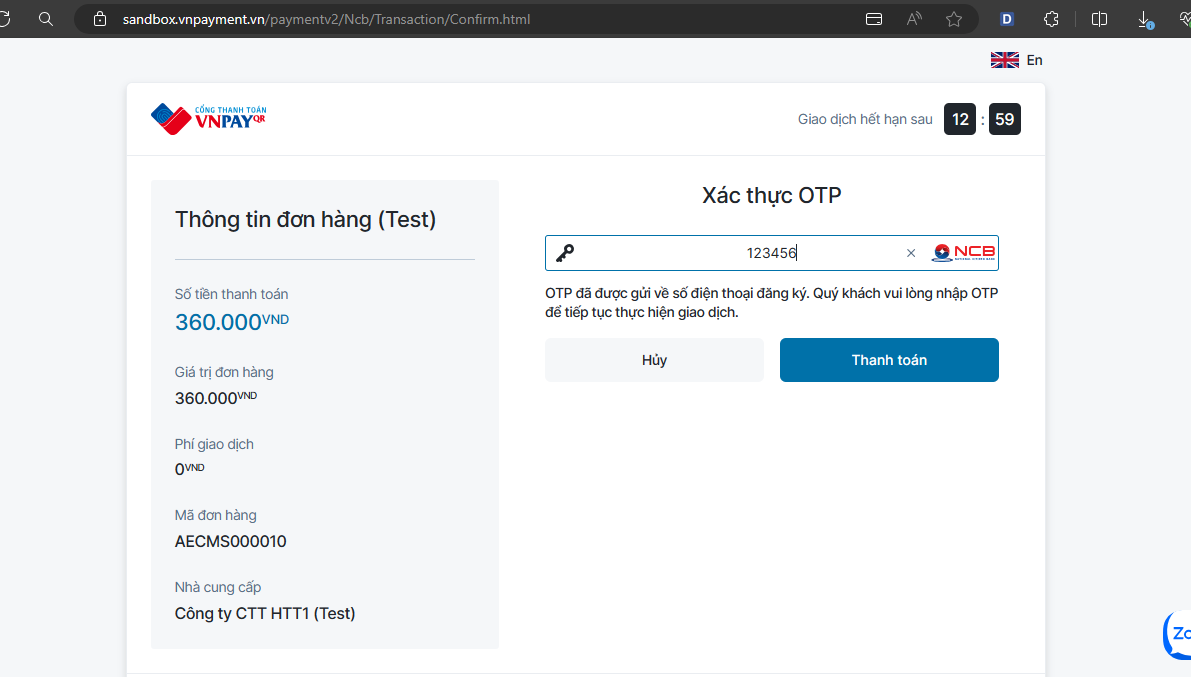
*Hình 52: Màn hình thanh toán trực tuyến*



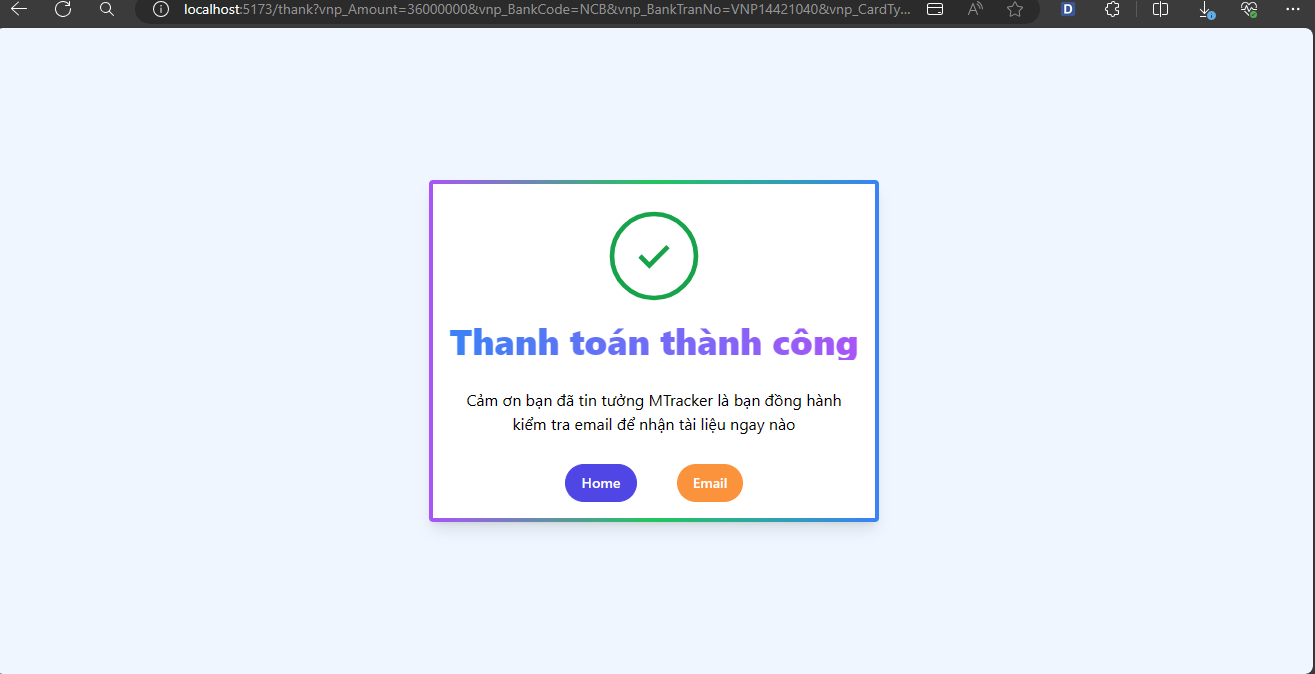
*Hình 53: Màn hình chọn phương thức thanh toán trên VNPAY*



*Hình 54: Màn hình xác minh thẻ ngân hàng trên VNPAY*

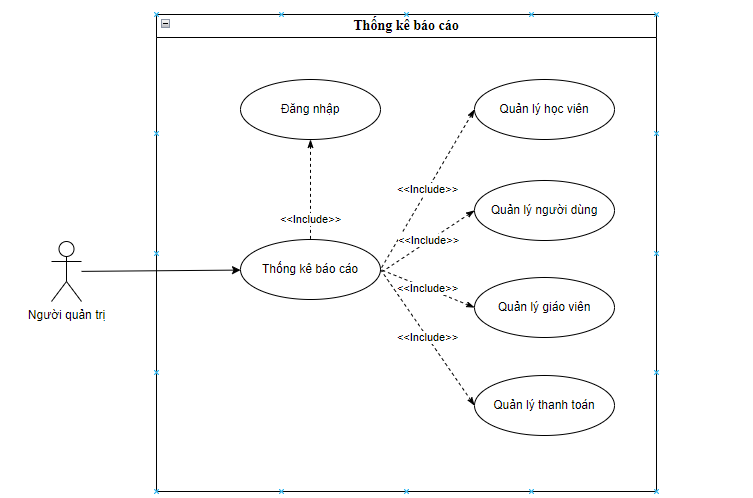


*Hình 55: Màn hình nhập OTP trên VNPAY*



*Hình 56: Màn hình thanh toán thành công*

### **2.3.8. Thống kê báo báo**



*Hình 57: Use case thông kê báo cáo*

#### **2.3.8.1. Mô tả chức năng**

**a. Giới thiệu**

Thống kê báo cáo là chức năng nhằm đảm bảo cho người quản trị có cái nhìn tổng quan về trình trạng vận hành của hệ thống. Chức năng này đòi hỏi lượng truy vấn cao từ hệ thống. Việc đảm bảo truy vấn được dữ liệu của thống kê vào báo cáo là một vấn đề cần được giải quyết

**b. Mục tiêu**

- Theo dõi được sự thay đổi của quá trình vận hành, phát triển một cách trực quan.

- Tính chính xác của các bản báo cáo.

- Một công việc chạy nền nhằm tối ưu hiệu suất của hệ thống.

#### **2.2.8.2. Triển khai trên hệ thống**

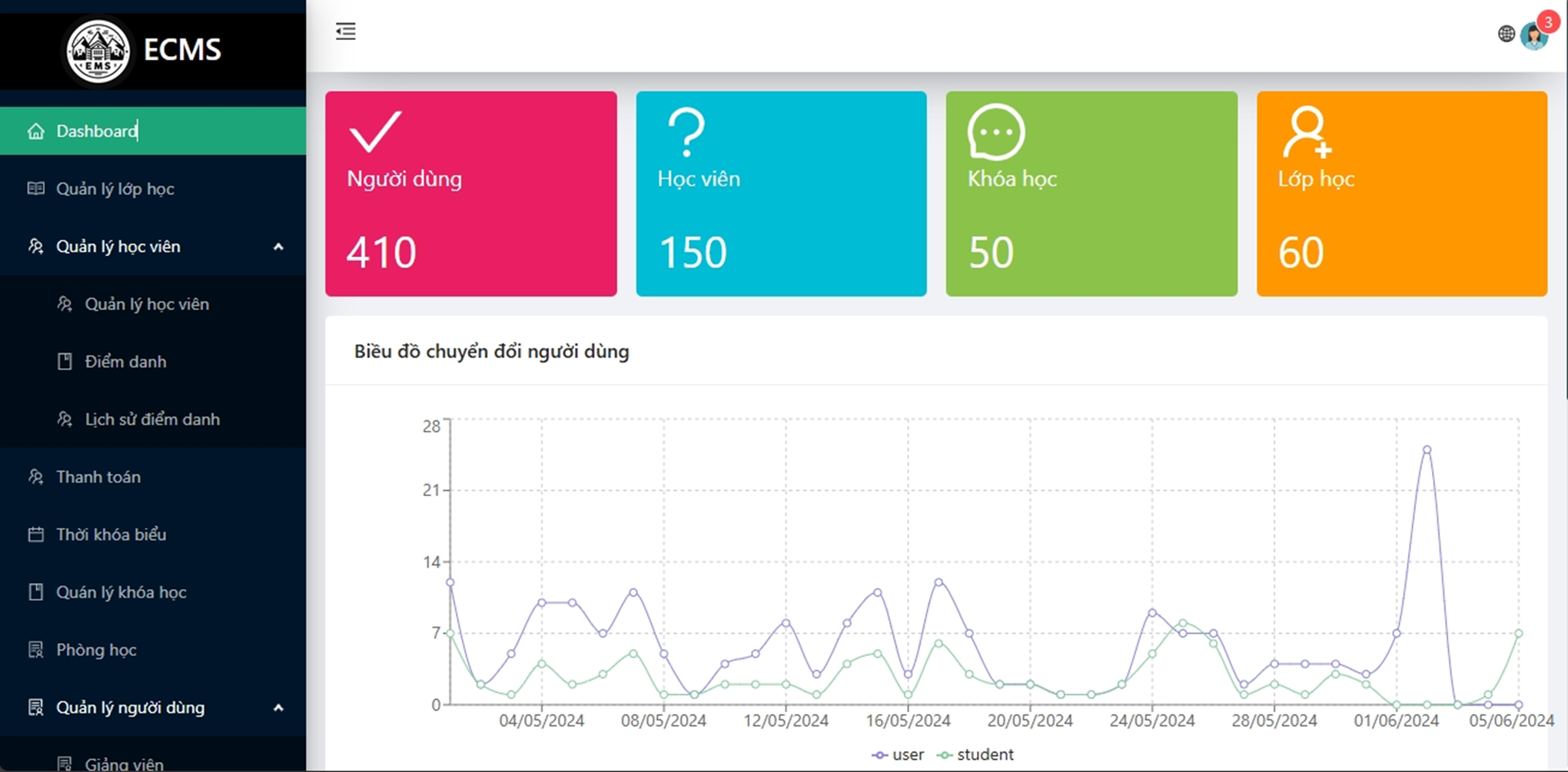
**a. Xử lý phía server**

**-** Hangfire job là một thư viện cho phép điều phối các công việc chạy ngầm trong hệ thống. Điều này thực sự hữu ích đối với các vấn đề mà bài toán thống kê đã đặt ra.

- Một công việc sẽ được lặp lại mỗi 30 phút nhằm mục đích cập nhật toàn bộ dữ liệu trong ngày. Mỗi ngày sẽ có một bản ghi mới được hình thành. Các truy vấn thống kê khác sẽ được thực thi và ghi đè lên bản ghi này trong ngày nhằm có dữ liệu mới nhất.

- Hiển thị thống kê báo cáo thông qua bảng đã được cập nhật dữ liệu

**b. Giao diện thống kê**



- Các biểu đồ chính:

+ Số lượng người dùng: Số người dùng đã đăng ký tài khoản trên hệ thống

+ Số lượng học viên: Số lượng người dùng đã đăng ký ít nhất một khóa học.

+ Khóa học: Thống kê số lượng khóa học.

+ Lớp học: Thống kê số lượng lớp học trong hệ thống.

+ Biểu đồ chuyển đổi người dùng: cho phép so sánh sự chuyển đổi từ người dùng sang học viên trong từng ngày.

# **CHƯƠNG 3: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

## **3.1 Kết luận**

Với khả năng hiện tại hệ thống quản lý trung tâm giáo dục đã đáp ứng các nhu cầu cơ bản của việc vận hành và phát triển của hệ thống. Với mục đích đề ra ban đầu là giảm thiểu tối đa các công việc thủ công bao gồm cấp phát thẻ học vật lý, đánh đâu thẻ bằng hệ thống đang ký online, điểm danh thông qua QR code. Từ danh sách các học viên được lưu trữ bằng các nhóm trên mạng xã hội thành phân lớp cho từng lớp học, quản lsy chi tiết thông tin học sinh. Bên cạnh đó trong quá trình phát triên gặp một số khó khăn và chưa thực hiện được như chỉ đáp ứng được một phần nhu cầu quản trị. Màn hình thống kê báo cáo chưa được hoàn thiện là một thiếu sót trong quá trình thực hiện, vận hành hệ thống.

## **3.2 Hướng phát triển**

Với khả năng hiện tại hệ thống quản lý trung tâm giáo dục chỉ đáp ứng được một phần nhu cầu quản trị, các chức năng có thể phát triển trong tương lai có thể phát triển thêm các danh mục sau:

Trang người dùng sẽ có một mục để học sinh có thể theo dõi các thông báo từ phía giảng viên thông qua một mạng xã hội thu nhỏ, nơi mà giảng viên có thể đang bài viết, bài tập, thông báo chuyển lịch, để học viên có thể theo dõi, chủ động, đồng thời rút ngắn khoảng cách giao tiếp giữa người học và người giảng dạy giúp tăng tính tương tác có ảnh hưởng tích cực đến tinh thần của học viên.

Ứng dụng moblie, để tiện theo dõi, ứng dụng moblie là định hướng tiếp theo của đồ án. Các hệ thống sẽ được nâng cấp các chương trình khuyến mãi, và việc tích điểm điểm danh sẽ được tích hợp vào trong ứng dụng với mực đích thay thế thẻ vật lý để điểm danh, dễ dang theo dõi các thông báo khi có thông tin từ phía giảng viên hoạc phía trung tâm.

Triển khai ứng dụng lên môi trường production hiện tại hệ thống có 2 tên miền để lưu trữ, vận hành ứng dụng đó là ecms-dev.site nơi cung cấp API logic cho phía giao diện người cùng và giao diện quản lý. Về giao diện người dùng đang triển khai tại ecms.io.vn là một trung tâm sử dụng hạ tầng của hệ thống để quản lý trung tâm nơi học sinh có thể đăng ký trực tiếp vào học.

# **TÀI LIỆU THAM KHẢO**

1. Asp Net Boilerplate (2013). ASP.NET Abp Boilerplate Document: <https://aspnetboilerplate.com/Pages/Documents> ( Truy cập ngày 20-02-2024)
2. GeeksforGeeks (2008). Domain-Driven Design: [https://www.geeksforgeeks.org/domain-driven-design-ddd](https://www.geeksforgeeks.org/domain-driven-design-ddd/)  ( Truy cập ngày 20-02-2024)
3. ReactJS (2011). React Document: <https://legacy.reactjs.org/tutorial/tutorial.html> ( Truy cập ngày 16-03-2024 )
4. Vite - Evan You (2019). Vite Document: <https://vitejs.dev/guide> ( Truy cập ngày 16-03-2024)
5. Tailwind Css Document: <https://tailwindcss.com/docs/installation> ( Truy cập ngày 16-03-2024)