

**ĐẠI HỌC BÁCH KHOA HÀ NỘI  
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

# **ĐỒ ÁN MÔN HỌC**

## **NHẬP MÔN CÔNG NGHỆ PHẦN MỀM**

# TÊN ĐỀ TÀI: HUSTHUB – HỆ THỐNG QUẢN LÝ TRƯỜNG ĐẠI HỌC

**Nhóm sinh viên thực hiện :** Trần Quang Huy – 20226109  
Nguyễn Lương Hoàng Tùng – 20226129  
Ngô Mạnh Hùng – 20226083  
Nguyễn Trung Hiếu – 20226082

Lớp : NMCNPM Mã lớp học: 152259

HÀ NỘI, 12/2024

# Mục lục

<b>Mục lục .....</b>	<b>1</b>
<b>Chương 1 Giới thiệu đề tài .....</b>	<b>6</b>
1.1.    Đặt vấn đề .....	6
1.2.    Mục tiêu và phạm vi đề tài.....	6
1.3.    Định hướng giải pháp .....	8
1.4.    Bố cục đồ án .....	8
<b>Chương 2 Khảo sát và phân tích yêu cầu.....</b>	<b>10</b>
2.1.    Khảo sát hiện trạng .....	10
2.2.    Tổng quan chức năng.....	11
2.2.1.    Biểu đồ use case tổng quan .....	11
2.2.2.    Biểu đồ use case phân rã .....	12
2.2.3.    Quy trình nghiệp vụ .....	16
2.3.    Đặc tả chức năng.....	19
2.3.1.    Đặc tả use case 001 .....	19
2.3.2.    Đặc tả use case 002 .....	21
2.3.3.    Đặc tả use case 003 .....	22
2.3.4.    Đặc tả use case 004 .....	23
2.3.5.    Đặc tả use case 005 .....	25
2.3.6.    Đặc tả use case 006 .....	26
2.3.7.    Đặc tả use case 007 .....	28
2.3.8.    Đặc tả use case 008 .....	30
2.3.9.    Đặc tả use case 009 .....	31
2.3.10.    Đặc tả use case 010 .....	33
2.4.    Yêu cầu phi chức năng.....	34

2.4.1.	Hiệu suất(Performance) .....	34
2.4.2.	Bảo mật(Security) .....	35
2.4.3.	Tính dễ dùng (Usability):.....	35
2.4.4.	Tính dễ bảo trì (Maintainability):.....	35
<b>Chương 3</b>	<b>Công nghệ sử dụng .....</b>	<b>37</b>
3.1.	Hệ quản trị cơ sở dữ liệu PostgreSQL .....	37
3.2.	Thư viện Express.js.....	37
3.3.	Nền tảng Node.js .....	38
<b>Chương 4</b>	<b>Phát triển và triển khai ứng dụng.....</b>	<b>39</b>
4.1.	Thiết kế kiến trúc .....	39
4.1.1.	Lựa chọn kiến trúc phần mềm.....	39
4.1.2.	Thiết kế tổng quan.....	42
4.1.3.	Thiết kế chi tiết các chức năng.....	44
4.2.	Thiết kế chi tiết .....	45
4.2.1.	Thiết kế giao diện.....	45
4.2.2.	Thiết kế cơ sở dữ liệu.....	46
4.2.2.1.	Phân tích thiết kế cơ sở dữ liệu .....	46
4.2.2.2.	Lược đồ cơ sở dữ liệu .....	48
4.2.2.3.	Sơ đồ thực thể liên kết.....	48
4.3.	Xây dựng ứng dụng .....	49
4.3.1.	Thư viện và công cụ sử dụng .....	49
4.3.2.	Kết quả đạt được .....	50
4.3.3.	Minh họa các chức năng chính.....	51
4.3.1.1.	Giao diện chính .....	51
4.3.1.2.	Giao diện đăng nhập.....	53
4.3.1.3.	Chức năng của Admin.....	55
4.3.1.4.	Chức năng của giáo viên .....	59
4.3.1.5.	Chức năng của sinh viên .....	61
4.4.	Kiểm thử .....	63

4.4.1.	Chức năng quản lý thông tin sinh viên .....	63
4.4.2.	Chức năng đăng ký lớp của sinh viên .....	65
4.5.	Triển khai.....	66
4.5.1.	Mô hình triển khai.....	66
4.5.2.	Cài đặt môi trường: .....	66
4.5.3.	Clone dự án .....	67
4.5.4.	Thiết lập biến môi trường: .....	67
4.5.5.	Chạy dự án .....	67
4.5.6.	Truy cập ứng dụng: .....	67
5.	<b>Kết luận và hướng phát triển .....</b>	<b>68</b>
5.1.	Kết luận.....	68
5.2.	Hướng phát triển .....	68

# DANH SÁCH HÌNH ẢNH

Hình 1 Use case tổng quan .....	11
Hình 2 Use case "Quản lý giảng viên" .....	12
Hình 3 Use case "Quản lý sinh viên".....	13
Hình 4 Use case "Quản lý học phần".....	13
Hình 5 Use case "Quản lý lớp học" .....	14
Hình 6 Use case " Quản lý điểm số".....	14
Hình 7 Use case "Thanh toán học phí" .....	15
Hình 8 Use case "Quản lý lịch học" .....	15
Hình 9 Quy trình nghiệp vụ quản lí thông tin sinh viên .....	16
Hình 10 Quy trình quản lí khoá học .....	17
Hình 11 Nghiệp vụ quản lí điểm số .....	18
Hình 12 Nghiệp vụ thanh toán học phí.....	19
Hình 13 Mô hình MVC.....	39
Hình 14 Thư mục view trong dự án.....	40
Hình 15 Thư mục controller trong dự án .....	41
Hình 16 Biểu đồ phụ thuộc gói.....	42
Hình 17 Biểu đồ thành phần .....	44
Hình 18 Lược đồ cơ sở dữ liệu .....	48
Hình 19 Sơ đồ thực thể liên kết .....	49
Hình 20 Giao diện chính khách .....	51
Hình 21 Giao diện phần About us .....	51
Hình 22 Giao diện hiển thị thông báo.....	52
Hình 23 Giao diện hiển thị phần Events .....	52
Hình 24 Contact và Footer.....	53
Hình 25 Giao diện đăng nhập .....	53

Hình 26 Giao diện chọn role.....	54
Hình 27 Giao diện đăng nhập của admin.....	54
Hình 28 Giao diện đăng nhập của sinh viên .....	54
Hình 29 Giao diện đăng nhập của giáo viên.....	55
Hình 30. Giao diện chính của admin .....	55
Hình 31 Chức năng quản lí thông sinh viên .....	56
Hình 32 Chức năng thêm giáo viên .....	56
Hình 33 Chức năng cập nhật giáo viên.....	57
Hình 34 Xác nhận xoá giáo viên.....	57
Hình 35 Giao diện quản lí khoá học .....	58
Hình 36 Chức năng thêm khoá học .....	58
Hình 37 Chức năng cập nhật khoá học .....	58
Hình 38 Chức năng xoá khoá học.....	59
Hình 39 Giao diện quản lí lớp học.....	59
Hình 40 Giao diện người dùng giáo viên.....	59
Hình 41 Giao diện quản lí lớp giảng dạy .....	60
Hình 42 Giao diện quản lí sinh viên của giảng viên.....	60
Hình 43 Giao diện nhập điểm của giáo viên.....	61
Hình 44 Giao diện hiển thị của sinh viên.....	61
Hình 45 Thời khoá biểu cho sinh viên.....	62
Hình 46 Giao diện hiển thị điểm số của sinh viên.....	62
Hình 47 Giao diện đăng ký khoá học .....	63
Hình 48 Giao diện hiển thị các lớp đã đăng ký .....	63

# **Chương 1 Giới thiệu đề tài**

## **1.1. Đặt vấn đề**

Hiện nay, các quy trình quản lý trong trường học như đăng ký lớp học, theo dõi thông tin sinh viên, giảng viên và quản lý lớp học chủ yếu được thực hiện thủ công. Việc này không chỉ tốn thời gian mà còn dễ dẫn đến sai sót, thiếu sót thông tin, ảnh hưởng đến hiệu quả và chất lượng công việc.

Triển khai hệ thống quản lý tự động sẽ giúp giảm thiểu các thao tác thủ công, tăng cường kiểm soát thông tin chặt chẽ và kịp thời. Các công việc như đăng ký lớp, cập nhật thông tin, hay phân công lớp học sẽ trở nên nhanh chóng và chính xác, nâng cao hiệu quả quản lý và trải nghiệm của người dùng.

Hệ thống tự động còn mang lại lợi ích lớn cho sinh viên, giảng viên và cán bộ quản lý. Sinh viên dễ dàng đăng ký lớp học, kiểm tra thời khóa biểu; giảng viên quản lý lớp học hiệu quả hơn; và nhà trường dễ dàng tổng hợp, phân tích dữ liệu để hỗ trợ các quyết định chiến lược. Đây là bước đi cần thiết để hiện đại hóa môi trường học tập và bắt kịp xu thế chuyển đổi số trong giáo dục.

## **1.2. Mục tiêu và phạm vi đề tài**

### **Mục tiêu:**

Việc phát triển một ứng dụng web quản lý giáo dục là giải pháp tối ưu để hiện đại hóa quy trình quản lý và nâng cao hiệu quả hoạt động của nhà trường. Ứng dụng sẽ giúp nhà trường quản lý tập trung các công việc như thông tin sinh viên, giảng viên, đăng ký lớp học, điểm danh, và kết quả học tập.

Đối với giảng viên, ứng dụng giúp thầy cô có thể dễ dàng theo dõi danh sách lớp, điểm danh trực tuyến, nhập và quản lý điểm số, giảm thiểu công việc hành chính để tập trung vào giảng dạy.

Đối với sinh viên, ứng dụng sẽ cung cấp các chức năng như đăng ký lớp học, kiểm tra thời khóa biểu và kết quả học tập mọi lúc, mọi nơi, giúp cải thiện trải nghiệm học tập và đảm bảo tính minh bạch trong quản lý. Với khả năng tự động hóa và tích hợp nhiều tính năng, ứng dụng này không chỉ nâng cao hiệu quả quản lý mà còn xây dựng một môi trường giáo dục hiện đại, thông minh và kết nối chặt chẽ giữa nhà trường, giảng viên và sinh viên.

## **Phạm vi đề tài:**

Dự án HUSTHUB nhằm xây dựng một hệ thống quản lý giáo dục hiện đại, hỗ trợ nhà trường, giảng viên và sinh viên trong các hoạt động quản lý và học tập. Hệ thống sẽ bao gồm các chức năng chính sau:

- **Quản lý sinh viên:** Hệ thống cho phép nhà trường quản lý thông tin sinh viên một cách tập trung, bao gồm thông tin cá nhân, quá trình học tập, và kết quả học tập. Hệ thống hỗ trợ phân loại sinh viên theo khoa, khóa học, hoặc các tiêu chí khác, đồng thời lưu trữ lịch sử học tập của từng sinh viên một cách chi tiết và dễ dàng truy xuất.
- **Đăng ký lớp học:** Sinh viên có thể đăng ký lớp học trực tuyến thông qua hệ thống, giúp tiết kiệm thời gian và giảm sai sót trong quá trình quản lý. Hệ thống cung cấp thông tin đầy đủ về các lớp học như lịch học, giảng viên phụ trách, số lượng chỗ trống, và điều kiện tham gia lớp học.
- **Quản lý môn học và lớp học:** Với đặc thù chương trình học theo tín chỉ của Đại học Bách Khoa Hà Nội, mỗi môn học thường được chia thành nhiều lớp học khác nhau, phù hợp với nhu cầu và lịch học của sinh viên. Hệ thống HUSTHUB sẽ cung cấp các công cụ quản lý thông minh, cho phép người quản trị dễ dàng quản lý tiến trình đào tạo của giảng viên và sinh viên.
- **Quản lý giảng viên:** HUSTHUB hỗ trợ nhà trường lưu trữ và quản lý thông tin giảng viên, bao gồm lịch dạy, các môn học phụ trách và kết quả giảng dạy. Giảng viên có thể truy cập hệ thống để cập nhật thông tin cá nhân, theo dõi tiến độ học tập của sinh viên, và quản lý tài liệu giảng dạy.
- **Quản lý thời khóa biểu và phòng học:** Hệ thống hiển thị trực quan thời khóa biểu của từng lớp học và tình trạng sử dụng phòng học. Nhà trường có thể sắp xếp và điều chỉnh lịch học, phòng học một cách linh hoạt để tối ưu hóa việc sử dụng cơ sở vật chất.
- **Quản lý kết quả học tập:** HUSTHUB hỗ trợ giảng viên theo dõi tình trạng tham gia học tập của sinh viên. Kết quả học tập của sinh viên được cập nhật thường xuyên, bao gồm điểm số, bài kiểm tra, và đánh giá tổng quan, giúp sinh viên dễ dàng theo dõi tiến trình học tập của mình.
- **Quản lý thanh toán học phí:** Hệ thống tích hợp các phương thức thanh toán đa dạng, bao gồm thanh toán trực tuyến, hỗ trợ sinh viên đóng học phí một cách nhanh chóng và an toàn. Thông tin về các khoản học phí được lưu trữ và quản lý chặt chẽ để đảm bảo tính minh bạch.
- **Báo cáo và thống kê:** Hệ thống cung cấp các báo cáo chi tiết về tình hình học tập, giảng dạy, quản lý lớp học, cũng như doanh thu từ học phí. Các dữ liệu này giúp nhà trường đưa ra quyết định quản lý phù hợp, tối ưu hóa nguồn lực, và nâng cao hiệu quả hoạt động giáo dục.

Phạm vi này hướng tới việc xây dựng một hệ thống quản lý giáo dục toàn diện, hỗ trợ tối ưu hóa quy trình vận hành, cải thiện trải nghiệm học tập và giảng dạy, đồng thời đáp ứng yêu cầu chuyển đổi số trong giáo dục hiện đại.

### **1.3. Định hướng giải pháp**

#### **(i) Định hướng công nghệ và phương pháp:**

Dự án sẽ được phát triển dựa trên kiến trúc RESTful API với Node.js đảm nhiệm phần backend để xử lý logic và quản lý dữ liệu. PostgreSQL được sử dụng làm cơ sở dữ liệu quan hệ để lưu trữ và truy xuất thông tin một cách có cấu trúc, đảm bảo tính toàn vẹn và khả năng mở rộng. Phần frontend sẽ được xây dựng bằng ReactJS, tạo ra giao diện người dùng hiện đại, thân thiện và dễ tùy chỉnh.

#### **(ii) Giải pháp đề xuất:**

Giải pháp là thiết kế và phát triển một hệ thống web quản lý giáo dục tích hợp các chức năng như quản lý sinh viên, giảng viên, môn học, lớp học, đăng ký và theo dõi kết quả học tập. Hệ thống đảm bảo khả năng tự động hóa quy trình, cung cấp trải nghiệm mượt mà cho cả quản trị viên lẫn người dùng cuối.

#### **(iii) Đóng góp chính và kết quả mong đợi:**

Đồ án sẽ đóng góp một hệ thống quản lý giáo dục hiệu quả, giúp nhà trường số hóa quy trình quản lý và cải thiện khả năng tương tác với sinh viên, giảng viên. Kết quả mong đợi là một hệ thống ổn định, bảo mật cao, tối ưu hóa các thao tác quản lý, giảm thiểu sai sót và cải thiện trải nghiệm người dùng trong môi trường học tập số hóa.

### **1.4. Bộ cục đồ án**

#### **Chương 2: Khảo sát và phân tích yêu cầu hệ thống**

Chương này trình bày tổng quan về HUSTHUB, nêu bật các vấn đề mà sinh viên và giảng viên hiện đang gặp phải trong việc quản lý học tập, trao đổi thông tin và cộng tác. Phần này sẽ làm rõ sự cần thiết của một hệ thống mới nhằm hỗ trợ toàn diện cho các hoạt động học tập và nghiên cứu tại HUST. Các chức năng cơ bản cần thiết của hệ thống sẽ được mô tả chi tiết, bao gồm: quản lý thông tin giảng viên, sinh viên, quản lý khoá học của admin; quản lý điểm số của giảng viên; quản lý học tập, đăng ký lớp và thanh toán học phí. Ngoài ra, chương này sẽ phân tích chi tiết các use case của hệ thống, giúp minh họa các tình huống sử dụng thực tế của người dùng.

#### **Chương 3: Công nghệ phát triển hệ thống**

Trong chương này, nhóm sinh viên sẽ giới thiệu các công nghệ chủ đạo được sử dụng để xây dựng hệ thống HUSTHUB, bao gồm Node.js, PostgreSQL, React.js, và Express.js. Mỗi

công nghệ sẽ được trình bày với vai trò cụ thể trong hệ thống, như cách PostgreSQL được sử dụng để lưu trữ dữ liệu một cách hiệu quả, hay React.js giúp xây dựng giao diện người dùng tương tác. Chương này cũng sẽ giải thích lý do lựa chọn các công nghệ này, nhằm đảm bảo tính ổn định, khả năng mở rộng và bảo trì lâu dài của hệ thống.

## **Chương 4: Phát triển và triển khai hệ thống**

Chương này tập trung mô tả chi tiết quy trình phát triển hệ thống HUSTHUB, từ thiết kế kiến trúc tổng quan đến triển khai thực tế. Nhóm sinh viên sẽ trình bày thiết kế giao diện người dùng (UI/UX), kiến trúc phần mềm (bao gồm mô hình MVC), và các chức năng chi tiết. Quy trình phát triển sẽ được minh họa qua các giai đoạn chính: thu thập yêu cầu, thiết kế, mã hóa, kiểm thử và triển khai. Ngoài ra, chương này cũng sẽ đề cập đến các công cụ và phương pháp kiểm thử được sử dụng để đảm bảo hệ thống hoạt động ổn định, đáp ứng tốt các nhu cầu của người dùng.

## **Chương 5: Kết luận và hướng phát triển**

Chương cuối sẽ tổng kết các kết quả đã đạt được trong quá trình phát triển hệ thống HUSTHUB, đồng thời phân tích những ưu điểm và hạn chế của hệ thống hiện tại. Nhóm sinh viên cũng sẽ đề xuất các hướng phát triển trong tương lai, bao gồm: mở rộng tính năng quản lý dự án nghiên cứu, tích hợp công cụ hỗ trợ học tập tiên tiến, cải thiện giao diện người dùng, và ứng dụng các công nghệ mới như trí tuệ nhân tạo (AI) để tối ưu hóa trải nghiệm người dùng.

Bộ cục này được xây dựng nhằm đảm bảo người đọc có cái nhìn toàn diện về hệ thống HUSTHUB, từ ý tưởng ban đầu, quy trình phát triển, cho đến triển khai và định hướng cải tiến trong tương lai.

# Chương 2 Khảo sát và phân tích yêu cầu

## 2.1. Khảo sát hiện trạng

### 2.2.1. Khảo sát từ nguồn người dùng/khách hàng

Hiện nay, Đại học Bách Khoa Hà Nội có nhiều hệ thống web app độc lập để quản lý các chức năng như thông tin sinh viên, giảng viên, học phần, lớp học, nhập điểm và đăng ký lớp học. Tuy nhiên, việc này gây khó khăn cho người dùng trong việc chuyển đổi giữa các nền tảng và làm giảm hiệu quả quản lý. Vì vậy, việc xây dựng một nền tảng duy nhất để tích hợp tất cả các chức năng này là rất cần thiết. Qua khảo sát, các yêu cầu chính từ người dùng bao gồm:

- **Quản lý thông tin chặt chẽ:** Cần hệ thống hóa thông tin sinh viên, giảng viên, học phần và lớp học một cách chính xác và hiệu quả.
- **Dễ dàng nhập và tra cứu điểm số:** Giảng viên cần công cụ hỗ trợ nhập điểm nhanh chóng và chính xác, đồng thời có thể xem danh sách sinh viên từng lớp học.
- **Tự động hóa quy trình đăng ký lớp học:** Sinh viên cần nền tảng để đăng ký lớp học trực tuyến, thanh toán học phí và xem thời khóa biểu cá nhân.
- **Hợp nhất các chức năng:** Một nền tảng duy nhất giúp giảm thiểu sự phức tạp khi chuyển đổi giữa các hệ thống.

### 2.2.2. Khảo sát các hệ thống đã có

Qua nghiên cứu, các hệ thống quản lý đã triển khai tại nhiều cơ sở giáo dục khác bao gồm:

- **Hệ thống SIS:** Quản lý thông tin sinh viên, điểm thi, và học phần. Tuy nhiên, hạn chế ở giao diện phức tạp và khả năng tương tác thấp.
- **EduSoft:** Hỗ trợ quản lý lớp học, lịch học và nhập điểm. Điểm yếu là thiếu tính năng tích hợp thanh toán và giao diện không thân thiện.
- **ctt-sis.hust.edu.vn:** Được sử dụng như công cụ cho đăng ký lớp học, nhưng thiếu khả năng tự động cập nhật dữ liệu.

### 2.2.3. Tính năng quan trọng cần phát triển

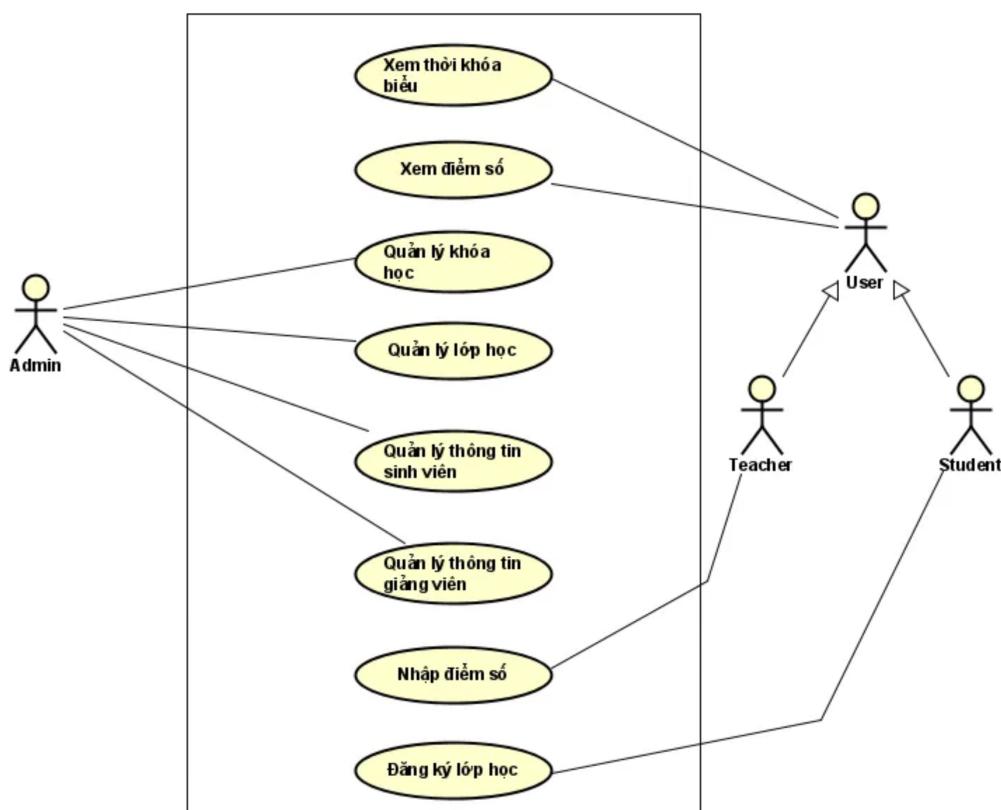
Dựa trên khảo sát, các tính năng quan trọng cho HUSTHUB bao gồm:

- **Quản lý thông tin sinh viên/giảng viên:** Hệ thống hóa thông tin cá nhân, lớp học, và lịch học.
- **Quản lý học phần và lớp học:** Tích hợp chức năng thêm, sửa, xóa và truy cập thông tin các lớp học.

- **Nhập và quản lý điểm số:** Hỗ trợ giảng viên nhập điểm và xem danh sách sinh viên lớp học.
- **Đăng ký lớp học và thanh toán học phí:** Hỗ trợ sinh viên đăng ký lớp học trực tuyến, quản lý giao dịch thanh toán, và hiển thị thời khóa biểu cá nhân.
- **Xem thời khóa biểu:** Sinh viên và giảng viên có thể tra cứu thời gian biểu một cách thuận tiện.
- **Tích hợp chức năng trên một nền tảng duy nhất:** Giảm thiểu sự phân mảnh và tăng tính hiệu quả trong sử dụng hệ thống.

## 2.2. Tổng quan chức năng

### 2.2.1. Biểu đồ use case tổng quan



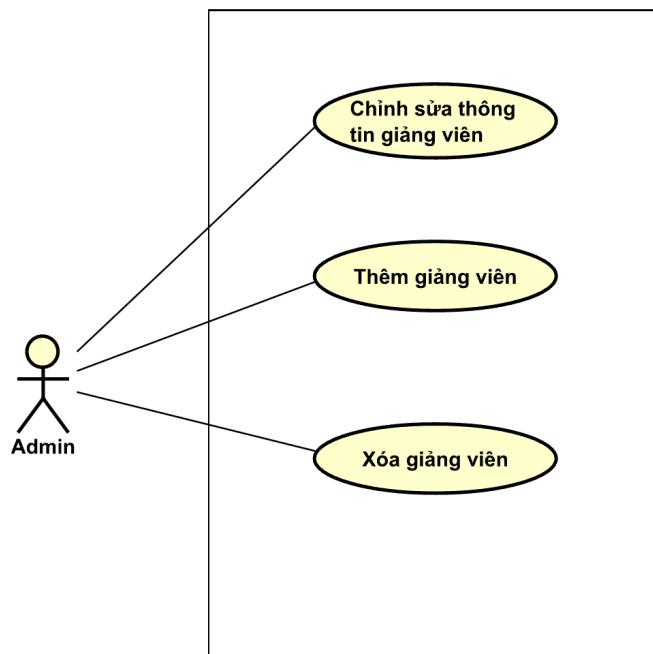
Hình 1 Use case tổng quan

- **Tác nhân tham gia**
  - **Admin:** Quản lý thông tin sinh viên, giảng viên, khóa học, lớp học.
  - **Teacher:** Nhập điểm, xem danh sách sinh viên lớp học.
  - **Student:** Đăng ký lớp học, thanh toán học phí, xem thời khóa biểu, theo dõi điểm số.
- **Mô tả ngắn gọn các Use Case chính**
  - **Quản lý thông tin sinh viên:** Admin thêm, sửa, xóa, truy vấn thông tin sinh viên.
  - **Quản lý thông tin giảng viên:** Admin thêm, sửa, xóa, truy vấn thông tin giảng viên.

- **Quản lý khóa học:** Admin thêm, sửa, xóa khóa học trong hệ thống.
- **Quản lý lớp học:** Admin quản lý thông tin lớp học liên quan đến khóa học.
- **Nhập điểm số:** Giáo viên nhập điểm cho sinh viên theo lớp học được phân công.
- **Đăng ký lớp học:** Sinh viên đăng ký lớp học theo thời khóa biểu.
- **Xem thời khóa biểu:** Sinh viên xem thời khóa biểu cá nhân.
- **Xem điểm số:** Sinh viên xem điểm các môn học đã tham gia và mô tả ngắn gọn các use case chính.

### 2.2.2. Biểu đồ use case phân rã

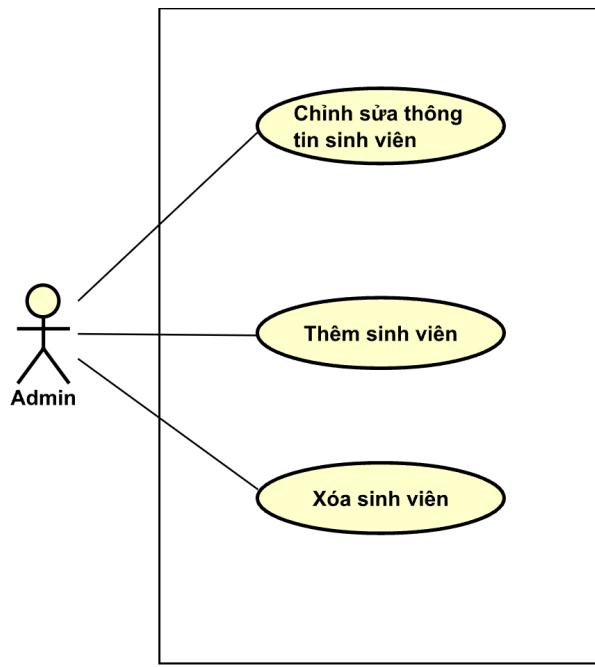
#### 2.2.2.1. Use case “Quản lí giảng viên”



Hình 2 Use case "Quản lí giảng viên"

Admin có thể thực hiện các thao tác quản lý giảng viên bao gồm: thêm mới giảng viên bằng cách nhập thông tin cơ bản, chỉnh sửa thông tin giảng viên đã có trong hệ thống và xóa giảng viên không còn hoạt động. Các thao tác này giúp đảm bảo dữ liệu giảng viên luôn chính xác và được cập nhật.

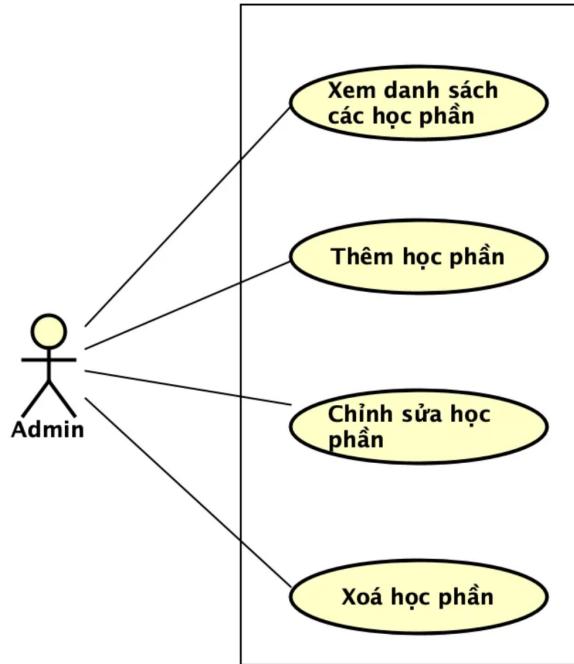
#### 2.2.2.2. Use case “Quản lí sinh viên”



Hình 3 Use case "Quản lí sinh viên"

Admin có thể quản lý thông tin sinh viên thông qua các thao tác như thêm mới sinh viên bằng cách nhập thông tin cần thiết, chỉnh sửa thông tin sinh viên đã có, và xóa sinh viên không còn trong hệ thống. Các chức năng này giúp duy trì và cập nhật dữ liệu sinh viên một cách chính xác và kịp thời.

#### 2.2.2.3. Use case “Quản lí học phần”

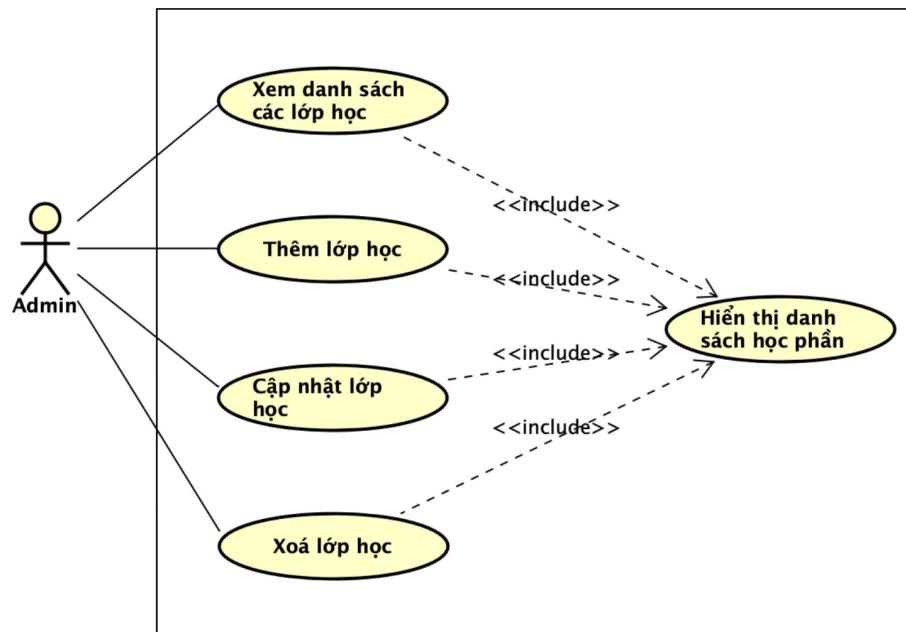


Hình 4 Use case "Quản lí học phần"

Admin thực hiện các chức năng quản lý học phần bao gồm: xem danh sách các học phần hiện có, thêm học phần mới với các thông tin chi tiết, chỉnh sửa thông tin học phần để cập

nhập dữ liệu, và xóa các học phần không còn phù hợp. Các thao tác này giúp đảm bảo hệ thống học phần luôn được duy trì chính xác và đầy đủ.

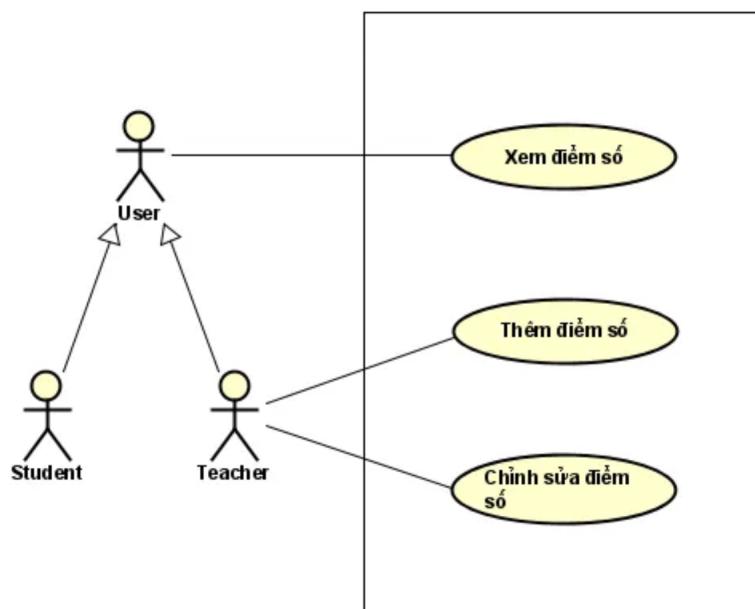
#### 2.2.2.4. Use case “Quản lý lớp học”



Hình 5 Use case "Quản lý lớp học"

Admin có thể quản lý thông tin lớp học thông qua các chức năng chính: xem danh sách các lớp học hiện có, thêm mới lớp học, cập nhật thông tin lớp học, và xóa lớp học không cần thiết. Tất cả các chức năng này đều được thực hiện dựa trên việc hiển thị danh sách học phần, đảm bảo tính liên kết và sự nhất quán trong dữ liệu quản lý.

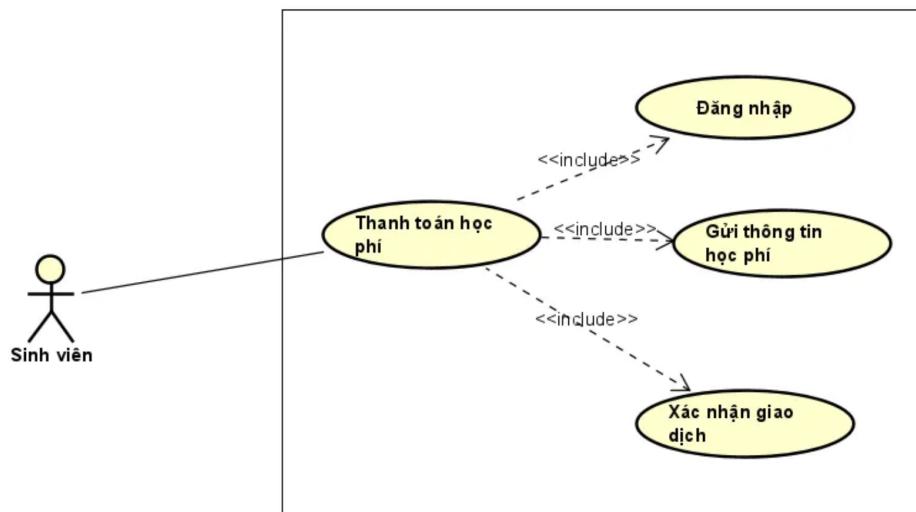
#### 2.2.2.5. Use case “Quản lý điểm số”



Hình 6 Use case " Quản lý điểm số"

Hệ thống hỗ trợ hai tác nhân: Student và Teacher. Student có thể xem điểm số của mình để theo dõi kết quả học tập. Teacher có quyền thêm điểm số mới và chỉnh sửa điểm số đã nhập, đảm bảo thông tin điểm số chính xác và kịp thời. Cả hai tác nhân đều sử dụng hệ thống để thực hiện các chức năng này một cách dễ dàng.

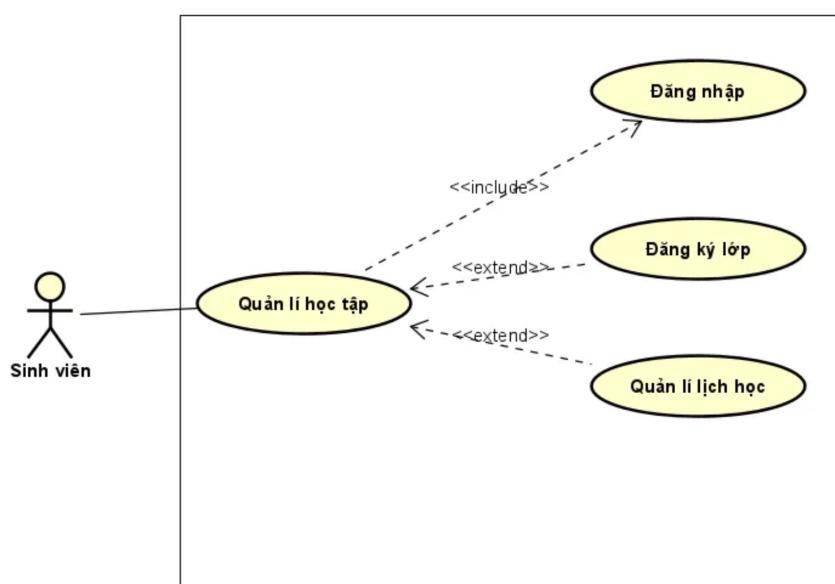
#### 2.2.2.6. Use case “Thanh toán học phí”



Hình 7 Use case "Thanh toán học phí"

Sinh viên có thể thực hiện chức năng thanh toán học phí thông qua hệ thống. Quy trình này bao gồm các bước: đăng nhập để xác thực tài khoản, gửi thông tin học phí cần thanh toán, và xác nhận giao dịch để hoàn tất thanh toán. Hệ thống đảm bảo tính chính xác và an toàn cho các giao dịch.

#### 2.2.2.7. Use case “Quản lý học tập”

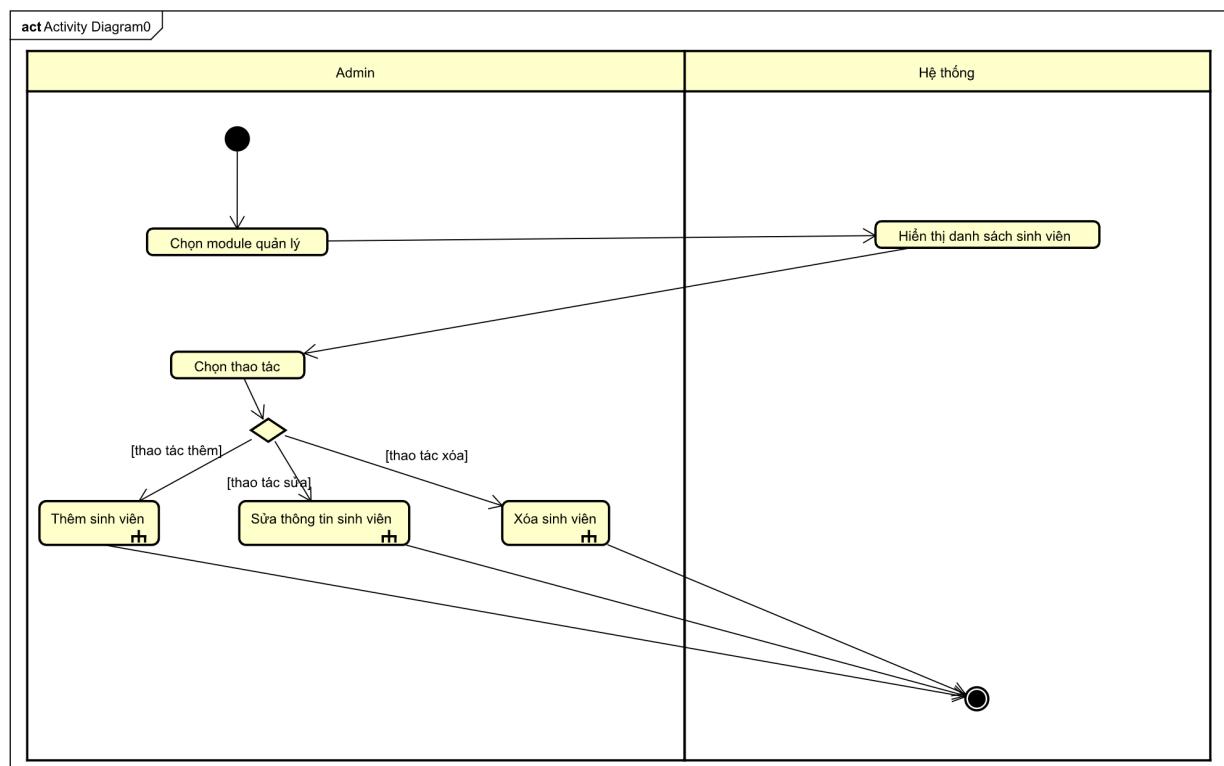


Hình 8 Use case "Quản lý lịch học"

Sinh viên sử dụng hệ thống để thực hiện chức năng quản lý học tập, bao gồm đăng nhập để xác thực tài khoản. Từ đó, sinh viên có thể đăng ký lớp học và quản lý lịch học của mình. Các chức năng này đảm bảo sinh viên dễ dàng theo dõi và sắp xếp quá trình học tập một cách hiệu quả.

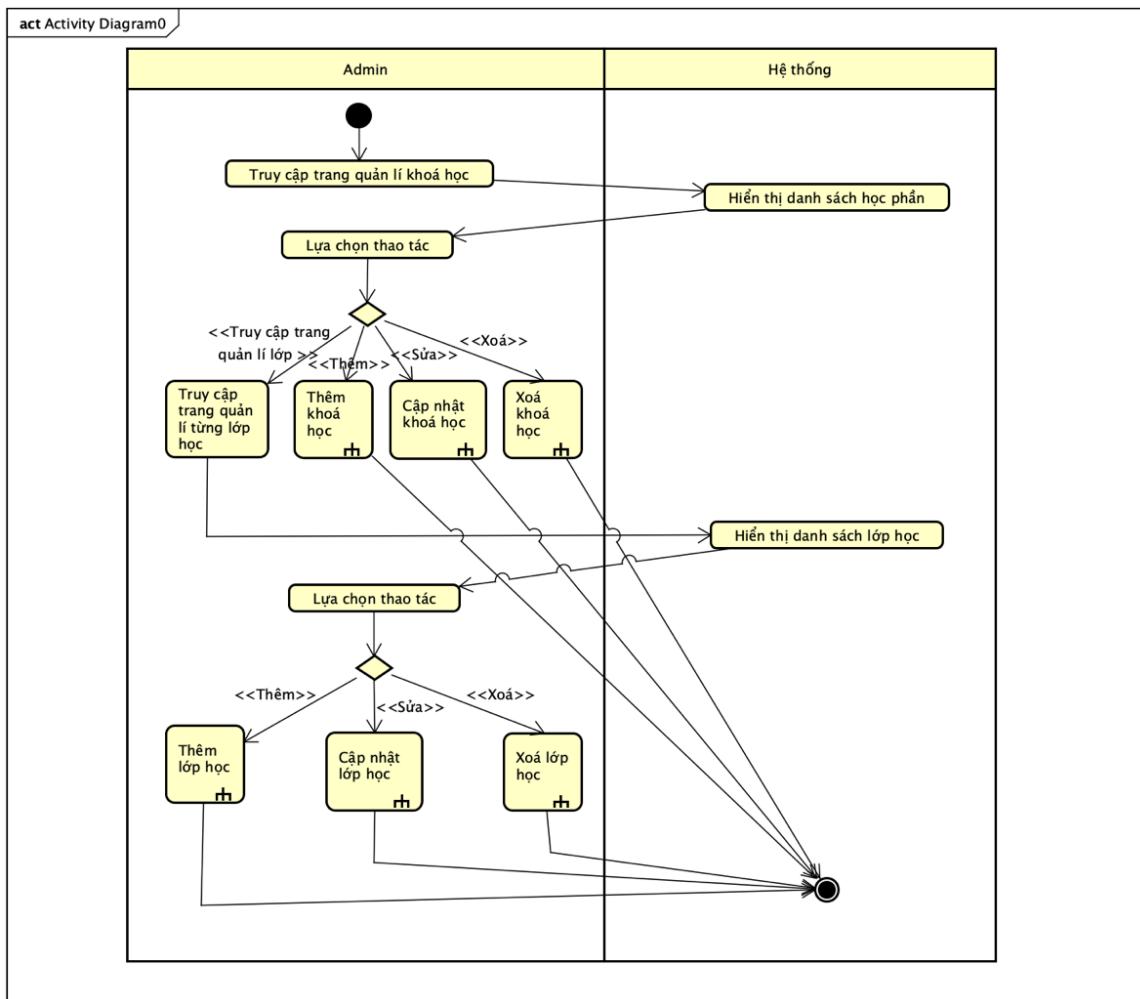
### 2.2.3. Quy trình nghiệp vụ

#### 2.2.3.1. Quy trình nghiệp vụ quản lý thông tin sinh viên



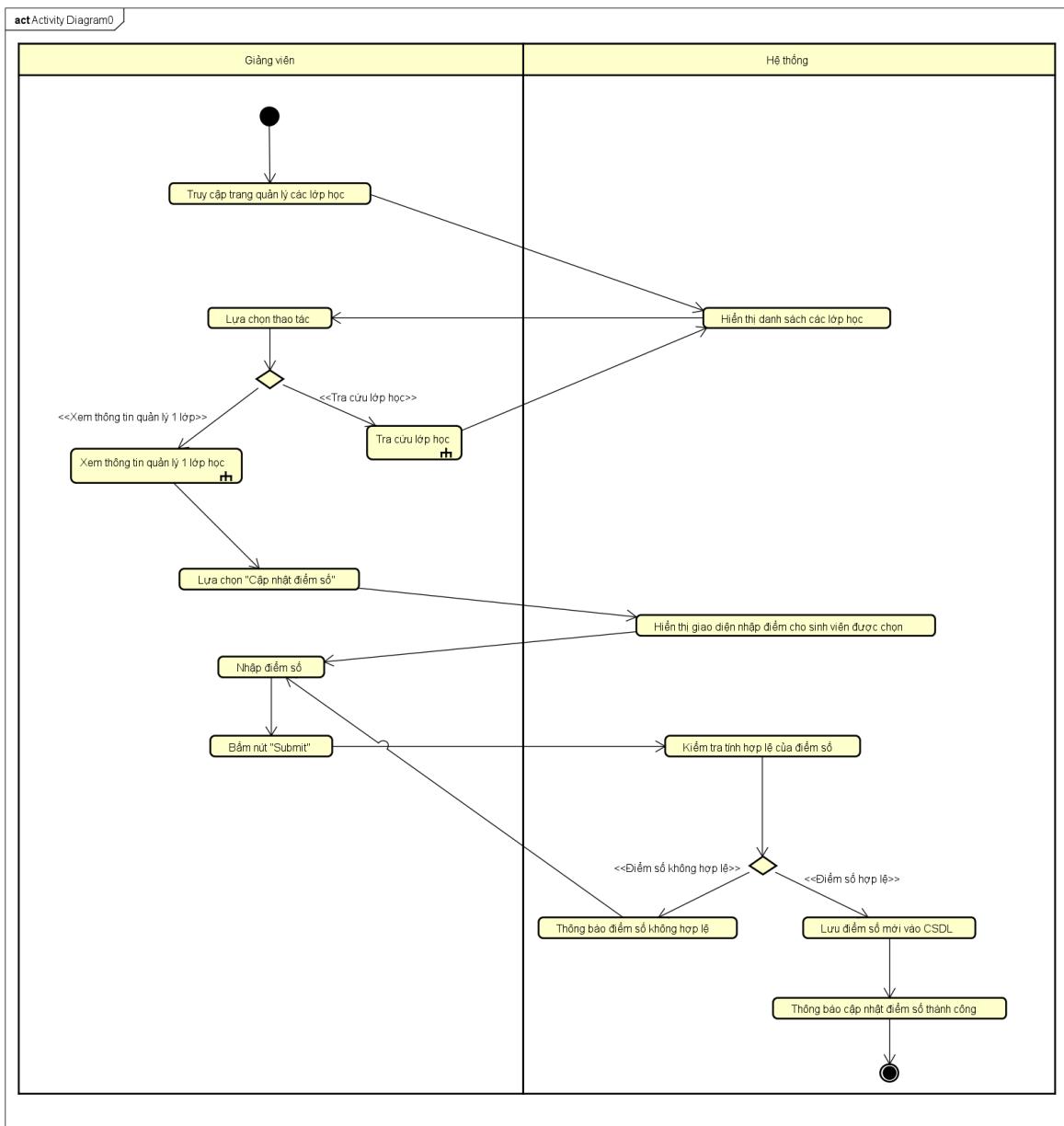
Hình 9 Quy trình nghiệp vụ quản lý thông tin sinh viên

#### 2.2.3.2. Quy trình quản lý khóa học



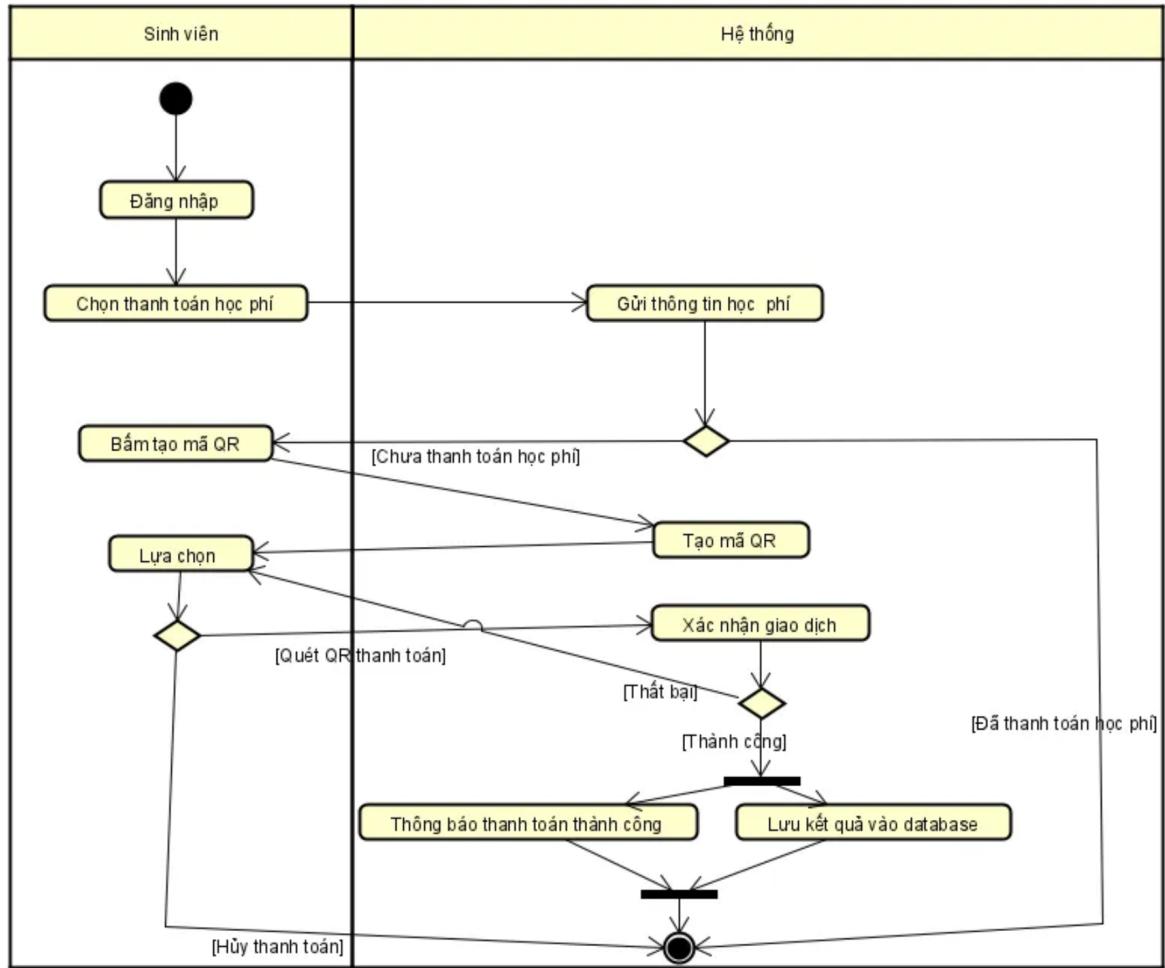
Hình 10 Quy trình quản lí khoá học

### 2.2.3.3. Quy trình quản lý điểm số



Hình 11 Nghiệp vụ quản lý điểm số

#### 2.2.3.4. Quy trình thanh toán học phí



Hình 12 Nghiệp vụ thanh toán học phí

### 2.3. Đặc tả chức năng

#### 2.3.1. Đặc tả use case 001

Use Case “Thêm sinh viên”	
<b>1. Mã use case</b>	UC002
<b>2. Mô tả ngắn gọn</b>	Trường hợp sử dụng này mô tả sự tương tác giữa admin và hệ thống khi admin muốn thêm sinh viên
<b>3. Tác nhân</b>	<ul style="list-style-type: none"> <li>Admin</li> </ul>
<b>4. Tiền đề kiện:</b>	Admin đã đăng nhập và có quyền truy cập vào danh sách sinh viên.
<b>5. Luồng sự kiện cơ sở</b>	<ol style="list-style-type: none"> <li>Admin xem danh sách sinh viên</li> <li>Hệ thống hiển thị danh sách sinh viên.</li> </ol>

3. Admin chọn chức năng thêm sinh viên.
4. Admin nhập trường dữ liệu cần thêm vào.
5. Hệ thống lưu bản ghi mới vào cơ sở dữ liệu (CSDL).
6. Hệ thống thông báo thông tin được thêm thành công.
7. Kết thúc UC

## 6. Luồng sự kiện thay thế

**Bảng N-Các luồng sự kiện thay thế cho thứ tự UC Place**

No	Vị trí	Điều kiện	Hành động	Vị trí quay lui
1.	Ở bước 4	Nếu thông tin sinh viên không hợp lệ	Hệ thống yêu cầu nhập lại	Quay lại bước 4
2.	Ở bước 4	Nếu thông tin sinh viên đã tồn tại	Hệ thống không thêm sinh viên, thông báo đã tồn tại sinh viên và yêu cầu nhập lại	Quay lại bước 4

## 7. Dữ liệu đầu vào

**Bảng đặc tả dữ liệu của trường ...**

No	Data fields	Description	Mandatory	Valid condition	Example
1.	MSSV	Mã số sinh viên	Có	MSSV phải hợp lệ	20221292
2.	Tên sinh viên		Có	Không chứa ký tự đặc biệt hoặc số	Nguyễn Văn B
3.	Ngày sinh		Có	Định dạng ngày hợp lệ (dd/mm/yyyy)	01/01/2003
4.	Lớp		Có	Tồn tại trong hệ thống	IT1

5.	Email		Có	Định dạng email hợp lệ	thu.px234762@sis.hust.edu.vn
----	-------	--	----	------------------------	------------------------------

### 2.3.2. Đặc tả use case 002

#### Use Case “Xóa sinh viên”

##### 1. Mã use case

UC002

##### 2. Mô tả ngắn gọn

Trường hợp sử dụng này mô tả sự tương tác giữa admin và hệ thống khi admin muốn xóa thông tin sinh viên

##### 3. Tác nhân

- Admin

4. Tiền điều kiện: Admin đã đăng nhập và có quyền truy cập vào danh sách sinh viên.

##### 5. Luồng sự kiện cơ sở

1. Admin xem danh sách sinh viên
2. Hệ thống hiển thị danh sách sinh viên
3. Admin chọn chức năng Tìm kiếm sinh viên
4. Admin chọn sinh viên cần xóa từ kết quả tìm kiếm
5. Hệ thống yêu cầu xác nhận xóa
6. Admin xác nhận
7. Hệ thống kiểm tra ràng buộc và lưu vào CSDL
8. Hệ thống thông báo xóa thành công
9. Kết thúc UC

##### 6. Luồng sự kiện thay thế

**Bảng N-Các luồng sự kiện thay thế cho thứ tự UC Place**

No	Vị trí	Điều kiện	Hành động	Vị trí quay lui
3.	Ở bước 4	Nếu không chọn sinh viên cần xóa	Hệ thống quay về danh sách	Đến bước 2
4.	Ở bước 6	Admin không xác nhận xóa	Hệ thống quay về danh sách	Đến bước 2

##### 7. Dữ liệu đầu vào

Bảng đặc tả dữ liệu của trường					
No	Data fields	Description	Mandatory	Valid condition	Example
6.	MSSV	Mã số sinh viên để tìm kiếm sinh viên	Không	MSSV phải tồn tại trong hệ thống	20221292
7.	Tên sinh viên		Có	Tên hợp lệ	Trần Minh Ánh

### 2.3.3. Đặc tả use case 003

#### Use Case “Chỉnh sửa thông tin giảng viên”

##### 1. Mã use case

UC003

##### 2. Mô tả ngắn gọn

Trường hợp sử dụng này mô tả sự tương tác giữa admin và hệ thống khi admin muốn chỉnh sửa thông tin giảng viên.

##### 3. Tác nhân

- Admin

4. Tiền điều kiện: Admin đã đăng nhập và có quyền truy cập vào danh sách giảng viên.

##### 5. Luồng sự kiện cơ sở

8. Admin xem danh sách giảng viên.
9. Hệ thống hiển thị danh sách giảng viên.
10. Admin chọn chức năng Tìm kiếm giảng viên.
11. Admin chỉnh sửa và lưu các trường dữ liệu cần thay đổi.
12. Hệ thống lưu thay đổi vào cơ sở dữ liệu (CSDL).
13. Hệ thống thông báo thông tin được cập nhật thành công.
14. Kết thúc UC.

##### 6. Luồng sự kiện thay thế

#### Bảng N-Các luồng sự kiện thay thế cho thứ tự UC Place

No	Vị trí	Điều kiện	Hành động	Vị trí quay lui
5.	Ở bước 4	Nếu không thay đổi thông tin	Hệ thống không cập nhật, thoát ra khỏi phần chỉnh sửa	Đến bước 7

## 7. Dữ liệu đầu vào

Bảng đặc tả dữ liệu của trường

No	Data fields	Description	Mandatory	Valid condition	Example
1	Mã GV	Mã giảng viên để tìm kiếm giảng viên	Có	MAGV phải tồn tại trong hệ thống	16500
2	Tên giảng viên	Tên đầy đủ của giảng viên	Có	Không chứa ký tự đặc biệt hoặc số	Trần Minh Anh
3	Ngày sinh	Ngày sinh của giảng viên	Không	Định dạng ngày hợp lệ (dd/mm/yyyy)	15/03/1980
4	Bộ môn	Bộ môn thuộc khoa	Không	Tồn tại trong hệ thống	Công nghệ thông tin
5	Email	Email được cấp bởi admin	Không	Định dạng email hợp lệ	minh.anh@gmail.vn

### 2.3.4. Đặc tả use case 004

#### Use Case “Thêm học phần”

##### 1. Mã use case

UC004

##### 2. Mô tả ngắn gọn

Use case này miêu tả sự tương tác giữa admin và hệ thống khi admin muốn thêm học phần mới.

##### 3. Tác nhân

- Admin

##### 4. Tiền điều kiện: Admin đã đăng nhập và có quyền quản lý các học phần

##### 5. Luồng sự kiện cơ sở

- 1) Admin truy cập danh sách các học phần từ hệ thống.

- 2) Admin chọn chức năng "Thêm học phần" trên giao diện.

3) Hệ thống hiển thị biểu mẫu yêu cầu nhập thông tin học phần (Course ID, Course Name, Course Credit).

4) Admin nhập thông tin học phần vào biểu mẫu.

5) Hệ thống kiểm tra tính hợp lệ của thông tin nhập vào.

6) Admin xác nhận hoàn thành và hệ thống cập nhật danh sách học phần.

7) Kết thúc UC.

## 6. Luồng sự kiện thay thế

Bảng N-Các luồng sự kiện thay thế cho thứ tự UC Place

No	Vị trí	Điều kiện	Hành động	Vị trí quay lui
6.	Ở bước 5	Nếu thông tin nhập vào không hợp lệ	Hệ thống yêu cầu nhập lại	Quay lại bước 4
7.	Ở bước 5	Nếu thông tin trùng lặp trong CSDL	Hệ thống thông báo lỗi, yêu cầu nhập lại.	Quay lại bước 4

## 7. Dữ liệu đầu vào

Bảng đặc tả dữ liệu của trường ...

No	Data fields	Description	Mandatory	Valid condition	Example
8.	Course ID	Mã học phần	Có	Không trùng lặp, hợp lệ	IT1013
9.	Course Name	Tên học phần	Có	Không chứa ký tự đặc biệt	Lập trình C
10.	Course Credit	Số tín chỉ	Có	Là số nguyên dương	3

## 8. Dữ liệu đầu ra

Bảng đặc tả dữ liệu đầu ra

No	Data fields	Description	Display format	Example
1.	Thông báo	Kết quả xử lý	Text	"Thêm học phần thành công"

### 2.3.5. Đặc tả use case 005

#### Use Case “Chỉnh sửa lớp học”

**1) Mã use case**

UC005

**2) Mô tả ngắn gọn**

Use case này miêu tả sự tương tác giữa admin và hệ thống khi admin muốn chỉnh sửa thông tin của một lớp học.

**3) Tác nhân**

- Admin

**4) Tiền điều kiện:** Admin đã đăng nhập và có quyền quản lý các học phần và các lớp học

**5) Luồng sự kiện cơ sở**

- 1) Admin truy cập danh sách các học phần từ hệ thống.
- 2) Admin chọn một học phần cụ thể.
- 3) Hệ thống hiển thị danh sách các lớp học của học phần đó.
- 4) Admin chọn một lớp học cần chỉnh sửa và nhấn chọn chức năng "Update".
- 5) Hệ thống hiển thị biểu mẫu yêu cầu nhập thông tin lớp học (Class ID, Teacher ID, Room ID, Time Start, Time End, Day, Course ID).
- 6) Admin nhập thông tin vào biểu mẫu.
- 7) Hệ thống kiểm tra tính hợp lệ của thông tin nhập vào
- 8) Admin xác nhận hoàn thành và hệ thống cập nhật danh sách lớp học.
- 9) Kết thúc UC.

**6) Luồng sự kiện thay thế**

Bảng N-Các luồng sự kiện thay thế cho thứ tự UC Place

No	Vị trí	Điều kiện	Hành động	Vị trí quay lui
1.	Ở bước 7	Nếu thông tin nhập vào không hợp lệ	Hệ thống yêu cầu nhập lại	Quay lại bước 6
2.	Ở bước 7	Nếu thông tin trùng lặp trong CSDL	Hệ thống thông báo lỗi, yêu cầu nhập lại.	Quay lại bước 6

**7) Dữ liệu đầu vào**

Bảng đặc tả dữ liệu của trường

No	Data fields	Description	Mandatory	Valid condition	Example

1.	Class ID	Mã lớp học	Có	Không trùng lặp, hợp lệ	151672
2.	Teacher ID	Mã giảng viên	Có	Phải tồn tại trong hệ thống	11521
3.	Room ID	Mã phòng học	Có	Phải tồn tại trong hệ thống	D9-501
4.	Time Start	Thời gian bắt đầu	Có	Định dạng hh:mm	08:00:00
5.	Time End	Thời gian kết thúc	Có	Định dạng hh:mm và sau Time Start	10:00:00
6.	Day	Ngày học	Có	Là các giá trị hợp lệ (Mon, Tue, ...)	Thứ hai
7.	Course ID	Mã phần	Có	Phải tồn tại trong hệ thống	IT1012

### 8) Dữ liệu đầu ra

Bảng đặc tả dữ liệu đầu ra

No	Data fields	Description	Display format	Example
1.	Thông báo	Kết quả xử lý	Text	"Cập nhật lớp học thành công"

### 9) Hậu điều kiện (nếu có)

- Hệ thống tự động cập nhật danh sách lớp học hiển thị cho admin.

#### 2.3.6. Đặc tả use case 006

### Use Case “Xoá học phần”

#### 1) Mã use case

UC006

#### 2) Mô tả ngắn gọn

Use case này miêu tả sự tương tác giữa admin và hệ thống khi admin muốn xoá một học phần ra khỏi lớp học.

### 3) Tác nhân

- Admin

### 4) Tiền điều kiện:

- Admin đã đăng nhập và có quyền quản lý các học phần và các lớp học
- Học phần muốn xoá phải tồn tại trong cơ sở dữ liệu.

### 5) Luồng sự kiện cơ sở

- 1) Admin truy cập danh sách học phần từ hệ thống.
- 2) Admin chọn một học phần cần xóa.
- 3) Admin nhấp chọn chức năng "Delete".
- 4) Hệ thống hiển thị biểu mẫu yêu cầu nhập Course ID của học phần.
- 5) Admin nhập Course ID và xác nhận xoá.
- 6) Hệ thống kiểm tra tính hợp lệ
- 7) Admin xác nhận hoàn thành và hệ thống cập nhật danh sách học phần.
- 8) Kết thúc UC.

### 6) Luồng sự kiện thay thế

Bảng N-Các luồng sự kiện thay thế cho thứ tự UC Place

No	Vị trí	Điều kiện	Hành động	Vị trí quay lui
1.	Ở bước 6	Nếu Course ID không hợp lệ	Hệ thống yêu cầu nhập lại	Quay lại bước 5
2.	Ở bước 6	Nếu thông tin trùng lặp trong CSDL	Hệ thống thông báo lỗi, yêu cầu nhập lại.	Quay lại bước 5

### 7) Dữ liệu đầu vào

Bảng đặc tả dữ liệu của trường

No	Data fields	Description	Mandatory	Valid condition	Example
1.	Course ID	Mã học phần	Có	Phải tồn tại trong hệ thống	IT1012

### 8) Dữ liệu đầu ra

Bảng đặc tả dữ liệu đầu ra				
No	Data fields	Description	Display format	Example
1.	Thông báo	Kết quả xử lý	Text	"Xóa học phần thành công"

**9) Hậu điều kiện (nếu có)**  
- Hệ thống tự động cập nhật danh sách học phần hiển thị cho admin.

### 2.3.7. Đặc tả use case 007

Use Case “Cập nhật điểm số”	
<b>Mã use case</b>	UC007
<b>Mô tả ngắn gọn</b>	Trường hợp sử dụng này mô tả sự tương tác giữa giảng viên và HUSTHub khi giảng viên muốn cập nhật điểm số của các sinh viên trong một lớp học.
<b>Tác nhân</b>	<ul style="list-style-type: none"> <li>-Giảng viên: Người có quyền cập nhật và điều chỉnh điểm số của sinh viên trong lớp học. Giảng viên chịu trách nhiệm nhập đúng điểm số và đảm bảo tính chính xác của dữ liệu trước khi cập nhật vào hệ thống.</li> </ul>
<b>Tiền điều kiện</b>	<ul style="list-style-type: none"> <li>- Giảng viên đã đăng nhập vào hệ thống.</li> <li>- Hệ thống hiển thị giao diện quản lý lớp học và điểm số sinh viên.</li> </ul>
<b>Luồng sự kiện cơ sở</b>	<ol style="list-style-type: none"> <li>Giảng viên chọn lớp học muốn cập nhật điểm số.</li> <li>Hệ thống hiển thị danh sách sinh viên trong lớp học đã chọn.</li> <li>Giảng viên chọn "Cập nhật điểm số" để chuyển đến giao diện cập nhật điểm.</li> <li>Hệ thống hiển thị giao diện nhập điểm cho từng sinh viên được chọn.</li> <li>Giảng viên nhập điểm số của sinh viên.</li> <li>Giảng viên nhấn nút "Submit" để xác nhận cập nhật điểm.</li> <li>Hệ thống kiểm tra tính hợp lệ của điểm số: <ul style="list-style-type: none"> <li>- Nếu điểm số không hợp lệ, hệ thống hiển thị thông báo lỗi và yêu cầu giảng viên nhập lại (chuyển sang Luồng sự kiện thay thế).</li> <li>- Nếu điểm số hợp lệ, chuyển sang bước tiếp theo.</li> </ul> </li> <li>Hệ thống lưu điểm số mới vào cơ sở dữ liệu.</li> <li>Hệ thống hiển thị thông báo "Cập nhật điểm số thành công".</li> </ol>
<b>Luồng sự kiện thay thế</b>	

**Bảng N-Các luồng sự kiện thay thế cho thứ tự UC Place**

No	Vị trí	Điều kiện	Hành động	Vị trí quay lui
1	Bước 7	Nếu điểm số không hợp lệ	Hệ thống hiển thị thông báo lỗi và yêu cầu nhập lại	Quay lại bước 5

### Dữ liệu đầu vào

No	Trường dữ liệu	Mô tả	Bắt buộc	Điều kiện hợp lệ	Ví dụ
1	Lớp học	Lớp mà giảng viên muốn cập nhật điểm số	Có	Lớp học phải tồn tại	156001
2	Mã số sinh viên	Sinh viên được chọn để cập nhật điểm số	Có	Sinh viên phải tồn tại	20226129
3	Điểm số	Điểm số của sinh viên	Không	Điểm số trong khoảng từ 0 đến 10	8.5

### Dữ liệu đầu ra

**Bảng đặc 1tả dữ liệu đầu ra**

No	Trường dữ liệu	Mô tả	Định dạng hiển thị	Ví dụ
1	Thông báo thành công	Thông báo cập nhật điểm số thành công	Hiển thị trên giao diện	"Cập nhật điểm số thành công"
2	Điểm số cập nhật	Điểm số mới của sinh viên	Hiển thị trong danh sách điểm số	Điểm của sinh viên A: 8.5

### Hậu điều kiện (nếu có)

- Điểm số của sinh viên được cập nhật thành công vào cơ sở dữ liệu.
- Điểm số hiển thị chính xác trên giao diện quản lý điểm của lớp học.

### 2.3.8. Đặc tả use case 008

#### Use Case “Xem thời khóa biểu của giảng viên”

##### 1. Mã use case

UC008

##### 2. Mô tả ngắn gọn

Trường hợp sử dụng này mô tả sự tương tác giữa giảng viên và HUSTHub khi giảng viên muốn xem thời khóa biểu cá nhân.

##### 3. Tác nhân

-Giảng viên: Người có quyền truy cập và xem thời khóa biểu cá nhân.

##### 4. Tiền điều kiện

- Giảng viên đã đăng nhập vào hệ thống.
- Hệ thống hiển thị giao diện cá nhân hóa của giảng viên.

##### 5. Luồng sự kiện cơ sở

- Giảng viên chọn mục "Thời khóa biểu" từ giao diện chính..
- Hệ thống hiển thị giao diện thời khóa biểu mặc định cho tuần hiện tại.
- Giảng viên có thể chọn một khoảng thời gian khác (ví dụ: tuần, tháng) để xem chi tiết thời khóa biểu.
- Hệ thống hiển thị thời khóa biểu tương ứng với khoảng thời gian đã chọn..
- Giảng viên có thể nhấp vào từng buổi học để xem chi tiết thông tin (môn học, phòng học, thời gian, v.v.).
- Hệ thống đảm bảo hiển thị thông tin chính xác và đầy đủ.

##### 6. Luồng sự kiện thay thế

Không có

##### 7. Dữ liệu đầu vào

No	Trường dữ liệu	Mô tả	Bắt buộc	Điều kiện hợp lệ	Ví dụ
1.	Mã giảng viên	Giảng viên muốn xem thời khóa biểu	Có	Giảng phải tồn tại	380691
2.	Khoảng thời gian	Thời gian muốn xem thời khóa biểu	Không	Tuần thủ định dạng thời gian	Tuần 12/2024

##### 8. Dữ liệu đầu ra

**Bảng đặc tả dữ liệu đầu ra**

No	Trường dữ liệu	Mô tả	Định dạng hiển thị	Ví dụ
2.	Thời khóa biểu	Danh sách các buổi học trong khoảng thời gian	Hiển thị dưới dạng bảng	Môn A - Thứ 2, 8:00 - 10:00

#### 9. **Hậu điều kiện (nếu có)**

- Giảng viên xem được thời khóa biểu tương ứng.
- Không có thay đổi nào được thực hiện trên hệ thống.

#### 2.3.9. **Đặc tả use case 009**

#### **Use Case “Thanh toán học phí”**

##### 1. **Mã use case**

UC009

##### 2. **Mô tả ngắn gọn**

Use case này mô tả tương tác giữa sinh viên và hệ thống khi thực hiện thanh toán học phí thông qua quét mã QR.

##### 3. **Tác nhân**

-Sinh viên: Người thực hiện thanh toán học phí.

##### 4. **Tiền điều kiện**

- Sinh viên đã đăng nhập vào hệ thống.
- Sinh viên có học phí cần thanh toán

##### 5. **Luồng sự kiện cơ sở**

1. Sinh viên đăng nhập vào hệ thống
2. Sinh viên chọn chức năng thanh toán học phí
3. Hệ thống gửi thông tin học phí
4. Hệ thống tạo mã QR cho giao dịch
5. Sinh viên lựa chọn phương thức thanh toán
6. Sinh viên quét mã QR để hoàn tất thanh toán
7. Hệ thống xác nhận giao dịch
8. Hệ thống lưu kết quả vào cơ sở dữ liệu
9. Hệ thống hiển thị xác nhận thanh toán

##### 6. **Luồng sự kiện thay thế**

**Bảng N-Các luồng sự kiện thay thế cho thứ tự UC Place**

No	Vị trí	Điều kiện	Hành động	Vị trí quay lui
----	--------	-----------	-----------	-----------------

1.	Bước 7	Xác nhận thanh toán thất bại	Hệ thống hiện lỗi và yêu cầu thử lại	Bước 5
2.	Bước 5	Sinh viên hủy thanh toán	Hệ thống kết thúc giao dịch	Kết thúc
3.	Bước 3	Không có học phí cần thanh toán	Hệ thống thông báo và kết thúc	Kết thúc

## 7. Dữ liệu đầu vào

No	Trường dữ liệu	Mô tả	Bắt buộc	Điều kiện hợp lệ	Ví dụ
1.	Thông tin đăng nhập	Email và mật khẩu	Có	Tài khoản tồn tại	Hieu.nt226082@sis.hust.edu.vn

## 8. Dữ liệu đầu ra

Bảng đặc tả dữ liệu đầu ra

No	Trường dữ liệu	Mô tả	Định dạng hiển thị	Ví dụ
1.	Trạng thái thanh toán	Kết quả giao dịch	Thông báo trạng thái	“You have paid successfully”

## 9. Hậu điều kiện (nếu có)

- Kết quả giao dịch của sinh viên được cập nhật thành công vào cơ sở dữ liệu.
- Thông tin học phí thanh toán thành công được hiển thị chính xác

### 2.3.10.Đặc tả use case 010

#### Use Case “Đăng ký lớp”

##### 1. Mã use case

UC010

##### 2. Mô tả ngắn gọn

Trường hợp sử dụng này mô tả sự tương tác giữa học sinh, hệ thống khi học sinh muốn đăng ký vào một lớp học.

##### 3. Tác nhân

- Học sinh: Người đăng ký vào lớp học bằng cách nhập mã lớp.
- Admin: Tạo khóa học và lớp học.

##### 4. Tiền điều kiện

- Admin đã tạo các khóa học và lớp học.
- Học sinh đăng nhập vào hệ thống và có quyền đăng ký lớp học.

##### 5. Luồng sự kiện cơ sở

1. Học sinh: Lựa chọn "Đăng ký lớp".
2. Học sinh: Nhập mã lớp.
3. Hệ thống: Kiểm tra mã lớp.  
Nếu mã lớp không tồn tại, hệ thống báo mã không hợp lệ và chờ nhập lại.
4. Hệ thống: Kiểm tra trùng thời khóa biểu (nếu mã lớp tồn tại).  
Nếu trùng thời khóa biểu, báo lỗi và chờ nhập lại.
5. Hệ thống: Nếu lớp hợp lệ, hiển thị thông tin lớp.
6. Hệ thống: Lưu thông tin đăng ký vào cơ sở dữ liệu.

##### 6. Luồng sự kiện thay thế

Bảng N-Các luồng sự kiện thay thế cho thứ tự UC Place

No	Vị trí	Điều kiện	Hành động	Vị trí quay lui
4.	Bước 3	Nếu mã lớp không hợp lệ	Hệ thống hiển thị thông báo lỗi và chờ nhập lại	Quay lại bước 2
5.	Bước 4	Nếu trùng thời khóa biểu	Hệ thống hiển thị thông báo lỗi và chờ nhập lại	Quay lại bước 2

##### 7. Dữ liệu đầu vào

No	Trường dữ liệu	Mô tả	Bắt buộc	Điều kiện hợp lệ	Ví dụ

2.	Mã lớp	Mã của lớp học	Có	Lớp học phải tồn tại, không trùng thời khóa biểu	156001
----	--------	----------------	----	--	--------

## 8. Dữ liệu đầu ra

### Bảng đặc tả dữ liệu đầu ra

No	Trường dữ liệu	Mô tả	Định dạng hiển thị	Ví dụ
2.	Thông báo thành công	Thông báo cập nhật điểm số thành công	Hiển thị trên giao diện	"Đăng ký lớp thành công"
3.	Đăng ký cập nhật	Thông tin đăng ký lớp	Hiển thị trong danh sách đăng ký	150001

## 9. Hậu điều kiện (nếu có)

Thông tin đăng ký lớp của học sinh được lưu thành công trong cơ sở dữ liệu

## 2.4. Yêu cầu phi chức năng

### 2.4.1. Hiệu suất(Performance)

- **Khả năng đáp ứng:** Hệ thống quản lý trường học cần đảm bảo xử lý nhanh chóng các yêu cầu từ người dùng, bao gồm việc tra cứu thông tin học sinh, giáo viên, thời khóa biểu và báo cáo. Đặc biệt, hệ thống phải hoạt động ổn định, nhất là khi có nhiều yêu cầu xử lý đồng thời
- **Khả năng mở rộng:** Hệ thống cần thiết kế để dễ dàng mở rộng cả về cơ sở hạ tầng phần cứng và kiến trúc phần mềm, nhằm đáp ứng sự gia tăng của số lượng sinh viên, giảng viên và dữ liệu khi trường học phát triển.
- Đảm bảo tốc độ phản hồi các thao tác của người dùng, ví dụ như truy xuất thông tin sinh viên hoặc xử lý cập nhật trạng thái điểm số
- **Cấu trúc truy vấn hiệu quả:** Các truy vấn SQL phức tạp nên được viết rõ ràng và tránh việc lặp lại không cần thiết

#### **2.4.2. Bảo mật(Security)**

- **Quản lý phiên làm việc:** Áp dụng xác thực dựa trên JWT (JSON Web Token) để đảm bảo rằng mỗi phiên làm việc đều được bảo vệ khỏi truy cập trái phép.
- **Kiểm tra và xác thực dữ liệu đầu vào:** Kiểm tra và làm sạch dữ liệu đầu vào từ người dùng để ngăn ngừa các cuộc tấn công như SQL Injection.
- **Phân quyền người dùng:** Xây dựng phân quyền đơn giản nhưng rõ ràng (ví dụ: admin, giáo viên, sinh viên) để đảm bảo người dùng chỉ có thể truy cập những dữ liệu và chức năng được phép.

#### **2.4.3. Tính dễ dùng (Usability):**

Hệ thống được xây dựng với sự phân chia rõ ràng các chức năng theo từng module riêng biệt, giúp cho việc quản lý trở nên dễ dàng và thuận tiện hơn. Người dùng có thể dễ dàng tìm kiếm và truy cập vào các chức năng cần thiết mà không phải mất nhiều thời gian tìm kiếm. Các chức năng được bố trí một cách trực quan, phù hợp với luồng công việc thực tế, giúp giảm thiểu thời gian thao tác và nâng cao trải nghiệm người dùng.

Bên cạnh đó, hệ thống còn tích hợp một hệ thống thông báo lỗi chi tiết, hỗ trợ người dùng nhanh chóng nhận diện và khắc phục những sai sót trong quá trình thao tác. Các thông báo lỗi được thiết kế dễ hiểu và cụ thể, cho phép người dùng biết chính xác vị trí và nguyên nhân của lỗi, từ đó giúp người dùng thao tác đúng đắn hơn mà không cảm thấy bối rối hoặc mất thời gian.

Giao diện của hệ thống được thiết kế dưới dạng bảng biểu quen thuộc, giống với các công cụ phổ biến mà người dùng thường sử dụng như Excel, giúp người dùng dễ dàng thao tác mà không cần thời gian làm quen. Nhờ vào thiết kế quen thuộc này, người dùng có thể nhanh chóng thực hiện các thao tác cần thiết một cách hiệu quả.

#### **2.4.4. Tính dễ bảo trì (Maintainability):**

- **Tổ chức mã nguồn rõ ràng:** Theo mô hình MVC, mã nguồn được chia thành các phần riêng biệt như: Model (làm việc với cơ sở dữ liệu), View (giao diện), và Controller (xử lý logic). Điều này giúp cho việc phát triển và sửa lỗi dễ dàng hơn.
- **Mã nguồn chuẩn hóa:** Áp dụng các quy tắc viết mã (coding standards) nhất quán để giảm thiểu lỗi phát sinh do sai sót khi bảo trì.
- **Kiểm tra lỗi (Debugging):** Hệ thống được tích hợp các công cụ kiểm tra và thông báo lỗi rõ ràng (ví dụ: log các lỗi truy vấn PostgreSQL, hoặc thông báo lỗi giao diện).

- **Khả năng mở rộng tính năng:** Hệ thống được thiết kế module hóa, nghĩa là có thể thêm mới hoặc thay đổi chức năng mà không ảnh hưởng lớn đến các phần khác. Ví dụ: thêm tính năng quản lý tài liệu giảng dạy hoặc báo cáo học phí.

# Chương 3 Công nghệ sử dụng

## 3.1. Hệ quản trị cơ sở dữ liệu PostgreSQL

PostgreSQL là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) mạnh mẽ và mã nguồn mở. PostgreSQL hỗ trợ các tính năng tiên tiến như ACID (Atomicity, Consistency, Isolation, Durability), giúp đảm bảo tính toàn vẹn và chính xác của dữ liệu.

### Ưu điểm:

- **Tính toàn vẹn dữ liệu:** PostgreSQL hỗ trợ đầy đủ ACID, đảm bảo các giao dịch được thực hiện đúng và an toàn.
- **Hỗ trợ phong phú:** PostgreSQL có khả năng quản lý dữ liệu dưới dạng quan hệ, JSON, và hệ thống index mạnh mẽ như B-tree.
- **An ninh:** Tích hợp các cơ chế bảo mật tiên tiến như SSL/TLS, mã hoá dữ liệu, và xác thực người dùng linh hoạt.

### Nhược điểm:

- **Cài đặt và quản trị:** Yêu cầu kiến thức về cài đặt và tối ưu hóa hệ thống.
- **Tính phức tạp của truy vấn:** Khi các truy vấn trở nên quá phức tạp, thời gian xử lý có thể tăng.

## 3.2. Thư viện Express.js

Express.js là một framework ứng dụng web dành cho Node.js, được thiết kế để hỗ trợ xây dựng các ứng dụng web và API một cách nhanh chóng và hiệu quả. Framework này cung cấp các tính năng phong phú để quản lý các yêu cầu HTTP, định tuyến và tích hợp middleware nhằm xử lý dữ liệu trước khi gửi phản hồi.

### Ưu điểm:

- **Đơn giản và linh hoạt:** Express.js rất dễ học và sử dụng
- **Routing mạnh mẽ:** Hỗ trợ các định tuyến động để quản lý các đường dẫn và phương thức HTTP.
- **Middleware phong phú:** Cho phép dễ dàng mở rộng và quản lý các yêu cầu, xác thực, và xử lý lỗi.

## Áp dụng

- **Tích hợp với cơ sở dữ liệu:** Sử dụng Express.js để xây dựng các API giao tiếp với hệ quản trị cơ sở dữ liệu như PostgreSQL.
- **Quản lý yêu cầu HTTP:** Hỗ trợ các phương thức như GET, POST, PUT, DELETE để xử lý các chức năng quản lý sinh viên, giáo viên, khóa học.
- **Xác thực người dùng:** Xử lý việc đăng nhập, đăng ký, và xác thực quyền truy cập thông qua các middleware bảo mật.
- **Hỗ trợ frontend:** Làm cầu nối giữa ứng dụng phía người dùng và cơ sở dữ liệu phía máy chủ.

### 3.3. Nền tảng Node.js

Node.js là một môi trường runtime JavaScript được xây dựng trên V8 JavaScript engine của Chrome. Nó cho phép chạy JavaScript ở phía server.

#### Ưu điểm:

- **Non-blocking I/O:** Xử lý nhiều yêu cầu đồng thời mà không bị block, giúp tăng hiệu suất.
- **Đồng nhất ngôn ngữ:** Sử dụng JavaScript cho cả frontend và backend, giúp giảm độ phức tạp.

#### Áp dụng:

- **Chạy server backend:** Sử dụng Node.js với Express.js để quản lý các route cho hệ thống quản lý trường học.
- **Xử lý logic nghiệp vụ:** Thực hiện xác thực người dùng, phân quyền, và xử lý các yêu cầu từ phía client.
- **Kết nối với cơ sở dữ liệu:** Tương tác với PostgreSQL để lưu trữ và truy vấn dữ liệu.
- **Xây dựng giao diện động với EJS:** Kết hợp EJS (Embedded JavaScript) để render giao diện phía server, tạo các trang động cho người dùng như bảng điểm, danh sách lớp học.
- **Sử dụng CSS thuần:** Để định dạng và tạo phong cách cho giao diện người dùng, đảm bảo hiệu suất cao và tương thích trên nhiều trình duyệt.

# Chương 4 Phát triển và triển khai ứng dụng

## 4.1. Thiết kế kiến trúc

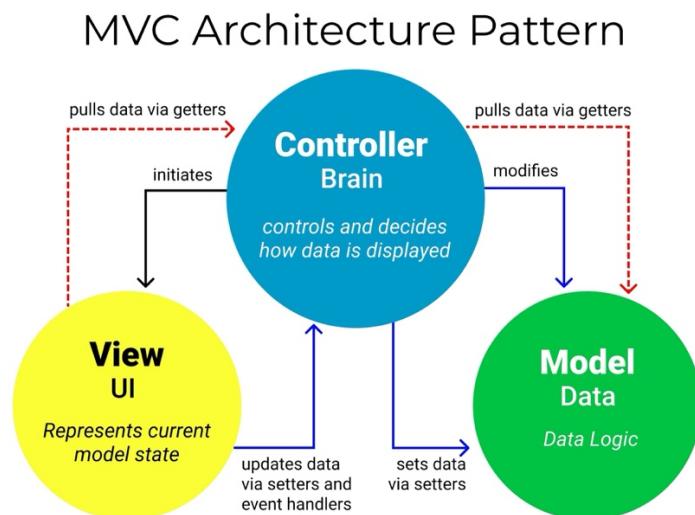
### 4.1.1. Lựa chọn kiến trúc phần mềm

Dự án Husthub sử dụng kiến trúc **MVC (Model-View-Controller)**. Kiến trúc này được sử dụng rộng rãi trong phát triển phần mềm vì nó cho phép phân tách rõ ràng các mối quan tâm, giúp ứng dụng dễ dàng bảo trì, kiểm thử và mở rộng. Kiến trúc này chia ứng dụng thành ba thành phần chính:

**Model:** Đại diện cho dữ liệu và logic nghiệp vụ của ứng dụng. Model chịu trách nhiệm truy xuất, thao tác và lưu trữ dữ liệu, thường tương tác với cơ sở dữ liệu.

**View:** Đại diện cho giao diện người dùng (UI). View chịu trách nhiệm hiển thị dữ liệu và thu thập đầu vào của người dùng, cung cấp cách thức để người dùng tương tác với hệ thống.

**Controller:** Là trung gian giữa Model và View. Controller xử lý đầu vào từ người dùng từ View, thao tác dữ liệu qua Model và cập nhật View tương ứng.



Hình 13 Mô hình MVC

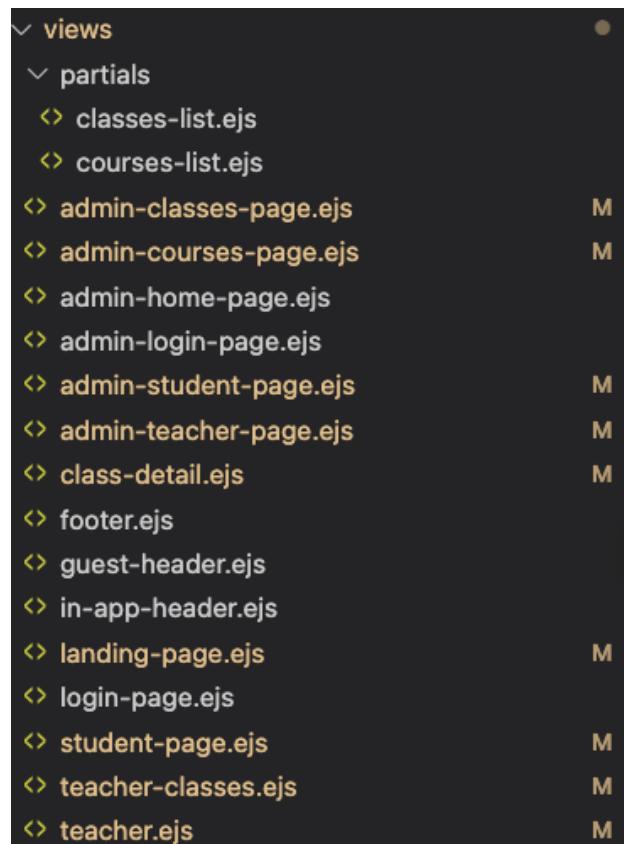
Trong kiến trúc này, mỗi thành phần có trách nhiệm rõ ràng, thúc đẩy tính mô-đun và khả năng bảo trì. Model hoàn toàn chịu trách nhiệm về logic nghiệp vụ và dữ liệu, View xử lý việc trình bày, còn Controller quản lý sự giao tiếp giữa Model và View.

## Áp dụng với dự án

### Model

- **db.js:** Đây là thành phần chính của **Model**, chịu trách nhiệm kết nối với cơ sở dữ liệu và thực thi các câu truy vấn.

**View:** Phần **View** trong dự án chịu trách nhiệm hiển thị giao diện người dùng (User Interface).



Hình 14 Thư mục view trong dự án

### Chức năng chính:

- Hiển thị nội dung và thông tin từ server tới trình duyệt của người dùng.
- Sử dụng **EJS (Embedded JavaScript)** để nhúng dữ liệu động vào các trang HTML.

### Những gì được hiển thị:

- **Giao diện cho Admin:**

- Trang quản lý lớp học (admin-classes-page.ejs), khóa học (admin-courses-page.ejs), giáo viên (admin-teacher-page.ejs), và sinh viên (admin-student-page.ejs).
- Trang đăng nhập cho Admin (admin-login-page.ejs).

- **Giao diện cho User:**

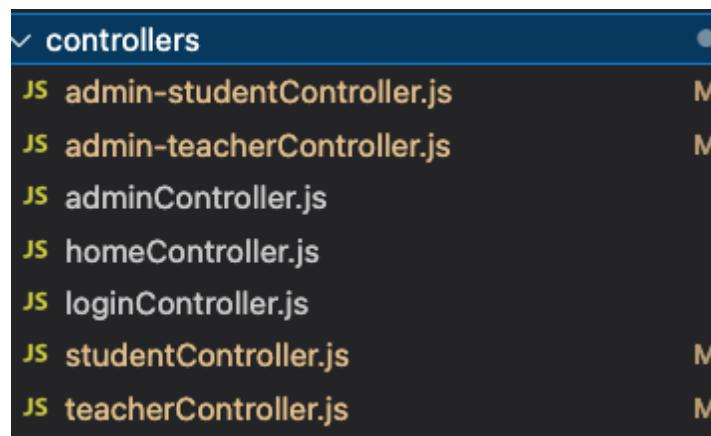
- Hiển thị danh sách lớp (classes-list.ejs), khóa học (courses-list.ejs), và chi tiết lớp (class-detail.ejs).
- Trang cá nhân của sinh viên (student-page.ejs) và giáo viên (teacher-classes.ejs, teacher.ejs).

### Cấu trúc chung:

- Thành phần tái sử dụng: Header (guest-header.ejs, in-app-header.ejs) và footer (footer.ejs).
- Trang chủ: Landing page (landing-page.ejs).
- Trang đăng nhập: (login-page.ejs).

Phần này giúp tổ chức giao diện linh hoạt, tái sử dụng và hỗ trợ tương tác giữa người dùng và hệ thống.

**Controller:** Chịu trách nhiệm xử lý logic ứng dụng, đóng vai trò là cầu nối giữa Model và View. Nó nhận yêu cầu từ người dùng, xử lý dữ liệu (từ Model), và gửi kết quả đến View để hiển thị.

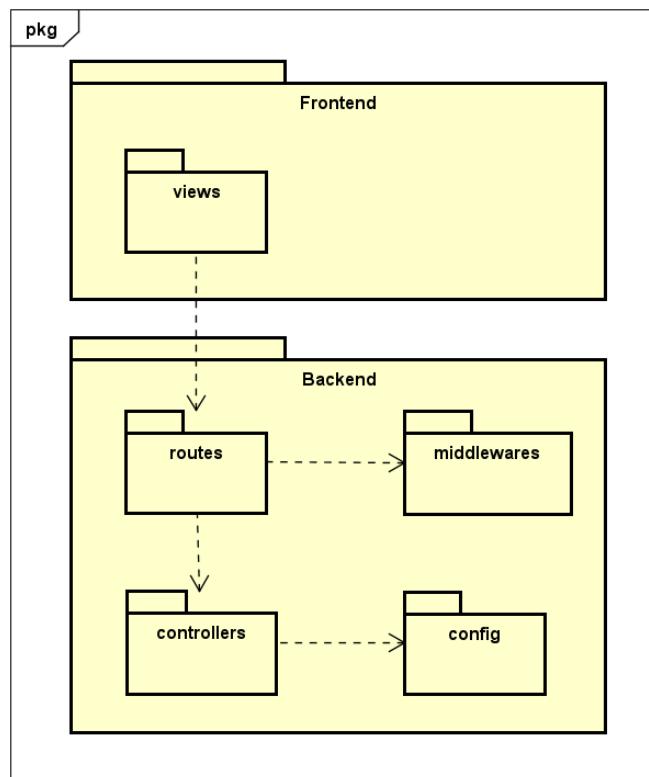


Hình 15 Thư mục controller trong dự án

- **Admin Controllers:**
  - adminController.js: Xử lý các chức năng chung của admin (truy cập trang quản trị, thống kê, điều hướng).
  - admin-studentController.js: Quản lý thông tin sinh viên, bao gồm thêm, sửa, xóa, và truy xuất danh sách sinh viên.
  - admin-teacherController.js: Quản lý thông tin giáo viên, xử lý tương tự như sinh viên.
- **User Controllers:**
  - homeController.js: Quản lý logic hiển thị trang chính (trang chủ, thông báo).
  - loginController.js: Xử lý xác thực và đăng nhập cho người dùng (Admin, Giáo viên, Sinh viên).

- studentController.js: Quản lý các chức năng liên quan đến sinh viên, như đăng ký khóa học, xem lịch học, xem điểm.
- teacherController.js: Xử lý các chức năng cho giáo viên, như quản lý lớp học, cập nhật điểm số.

#### 4.1.2. Thiết kế tổng quan



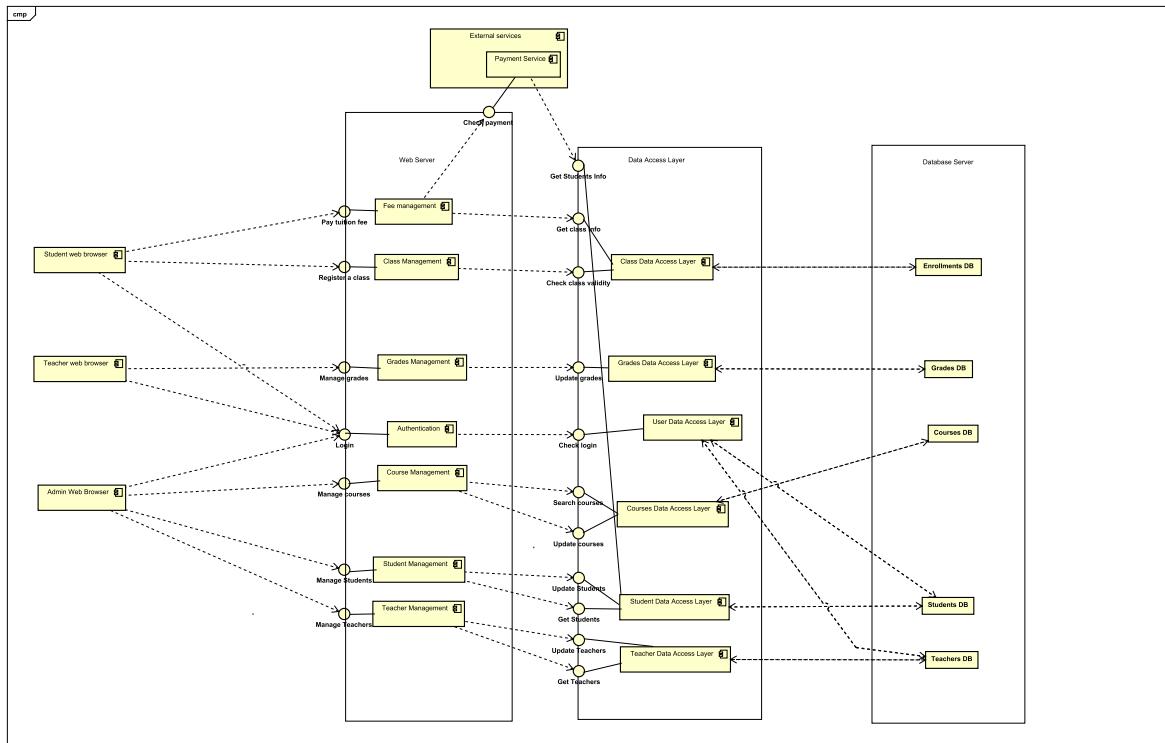
Hình 16 Biểu đồ phụ thuộc gói

**Giải thích các gói:**

- **Package views:**
  - Mục đích:
    - Chịu trách nhiệm hiển thị giao diện người dùng (UI/UX).
    - Tương tác với người dùng và gửi yêu cầu (requests) tới routes trong Backend.
  - Mối quan hệ:
    - views phụ thuộc vào routes trong Backend để lấy dữ liệu cần thiết thông qua giao tiếp API hoặc giao thức HTTP.
- **Package routes:**
  - Mục đích:
    - Định nghĩa các đường dẫn URL và ánh xạ chúng tới các logic xử lý trong controllers.

- Mỗi quan hệ:
  - Phụ thuộc vào middlewares để thực hiện các xử lý trung gian.
  - Gọi đến controllers để thực thi logic xử lý chính.
- **Package middlewares:**
  - Mục đích:
    - Xử lý các tác vụ trung gian như:
      - Kiểm tra xác thực người dùng (authentication).
      - Xử lý dữ liệu đầu vào.
      - Ghi log hoặc xử lý lỗi trước khi chuyển tiếp yêu cầu đến controllers.
  - Mỗi quan hệ:
    - Được sử dụng bởi routes để xử lý yêu cầu trước khi đến controllers.
- **Package controllers:**
  - Mục đích:
    - Thực hiện logic nghiệp vụ chính (business logic).
    - Xử lý yêu cầu từ routes, làm việc với dữ liệu và trả kết quả về cho frontend (qua routes).
  - Mỗi quan hệ:
    - Phụ thuộc vào config để lấy các thông tin cấu hình cần thiết.
- **Package config:**
  - Mục đích:
    - Lưu trữ các cấu hình toàn hệ thống như:
      - Thông tin kết nối cơ sở dữ liệu.
      - Các biến môi trường.
      - Cấu hình API hoặc các thông tin chung cho backend.
  - Mỗi quan hệ:
    - Được sử dụng bởi controllers để lấy thông tin cần thiết phục vụ xử lý logic.

### 4.1.3. Thiết kế chi tiết các chức năng



Hình 17 Biểu đồ thành phần

#### Giải thích thiết kế:

##### Tổng quan hệ thống:

- Hệ thống được chia thành các thành phần chính: Giao diện người dùng, Web Server, Data Access Layer, và Cơ sở dữ liệu.
- Có sự kết nối chặt chẽ giữa các thành phần để đảm bảo luồng xử lý thông tin xuyên suốt từ giao diện người dùng đến cơ sở dữ liệu.

##### Giao diện người dùng (User Interface):

- Gồm ba loại người dùng chính: Sinh viên, giáo viên, và quản trị viên.
- Mỗi loại người dùng có các chức năng riêng như:
  - + Sinh viên: Đăng ký lớp, thanh toán học phí.
  - + Giáo viên: Quản lý điểm.
  - + Quản trị viên: Cập nhật khóa học, quản lý sinh viên và giáo viên.

##### Web Server:

- Đóng vai trò xử lý logic nghiệp vụ, với các thành phần như:

- + Fee Management: Xử lý thông tin thanh toán.
- + Class Management: Kiểm tra tính hợp lệ của lớp.
- + Grades Management: Quản lý và cập nhật điểm.
- + Authentication: Xác thực người dùng.
- + Course Management: Tìm kiếm và tạo khóa học mới.
- + Student Management/Teacher Management: Quản lý thông tin người dùng.

#### **Data Access Layer:**

Chịu trách nhiệm giao tiếp với cơ sở dữ liệu thông qua các lớp truy cập dữ liệu (Data Access Layer) riêng biệt cho từng thành phần như Lớp học, Điểm, Khóa học, Sinh viên, và Giáo viên.

#### **Cơ sở dữ liệu:**

Lưu trữ dữ liệu của toàn bộ hệ thống, đảm bảo dữ liệu có thể được truy xuất nhanh chóng và chính xác.

#### **External Services (Payment Service):**

Tích hợp dịch vụ thanh toán bên ngoài để xử lý các giao dịch học phí.

## **4.2. Thiết kế chi tiết**

### **4.2.1. Thiết kế giao diện**

#### **4.2.1.1. Đặc tả thông tin về màn hình**

Hệ thống được thiết kế để hoạt động trên các thiết bị phổ biến hiện nay bao gồm máy tính để bàn, laptop, máy tính bảng. Cụ thể:

- Độ phân giải màn hình tối thiểu: 1280 x 720 pixel.
- Kích thước màn hình khuyến nghị: 5.5 inch trở lên (điện thoại), 10 inch trở lên (tablet).
- Số lượng màu sắc hỗ trợ: 16 triệu màu (True Color).

Yêu cầu tối thiểu:

- Browser hỗ trợ: Chrome, Firefox, Edge, Safari và Cốc Cốc phiên bản mới nhất.
- Tương thích: Tất cả các thiết bị có hỗ trợ HTML5, CSS3 và JavaScript ES6 trở lên.

#### **4.2.1.2. Chuẩn hóa thiết kế giao diện**

**Màu sắc:** -Xanh lá đậm: #186A3B

-Xanh nhạt pha xám: #CFE2CF

-Xám nhạt: #F4F4F4

-Teal: #009688

-Bootstrap Blue: #007BFF

### Kiểu chữ:

-Tiêu đề chính (For title/heading): Arial, Roboto, sans-serif, kích thước: 48px, màu sắc: #333

-Tiêu đề phụ (For sub title/heading): Arial, Roboto, sans-serif, kích thước: 1rem, màu sắc: #666

-Nội dung chính (For body text): Arial, Roboto, sans-serif, kích thước: 20px, màu sắc: black

## 4.2.2. Thiết kế cơ sở dữ liệu

### 4.2.2.1. Phân tích thiết kế cơ sở dữ liệu

#### Sinh viên (Students)

##### - Đặc điểm chính:

- *student\_id*: Mã số sinh viên (khóa chính), đảm bảo sự duy nhất cho mỗi sinh viên.
- *student\_name*: Tên sinh viên.
- *student\_dob*: Ngày sinh của sinh viên.
- *student\_email*: Địa chỉ email của sinh viên, duy nhất để xác định mỗi sinh viên trong hệ thống.
- *student\_major*: Ngành học của sinh viên.

- **Vai trò:** Lưu trữ thông tin chi tiết về sinh viên trong hệ thống. Đây là đối tượng trung tâm vì nó là người sử dụng chính của hệ thống, bao gồm việc ghi danh các lớp học, theo dõi điểm và tham gia các khóa học.

#### Giảng viên (Teachers)

##### - Đặc điểm chính:

- *teacher\_id*: Mã giảng viên (khóa chính).
- *teacher\_name*: Tên giảng viên.
- *teacher\_faculty*: Khoa mà giảng viên trực thuộc.
- *teacher\_email*: Địa chỉ email của giảng viên, đảm bảo duy nhất để dễ dàng liên hệ.

- **Vai trò:** Quản lý thông tin về giảng viên dạy các khóa học trong hệ thống. Giảng viên chịu trách nhiệm giảng dạy các lớp và đánh giá sinh viên trong các khóa học.

## **Khóa học (Courses)**

- **Đặc điểm chính:**
  - *course\_id*: Mã khóa học (khóa chính).
  - *course\_name*: Tên khóa học.
  - *course\_credit*: Số tín chỉ của khóa học.
- **Vai trò:** Xác định các khóa học có trong hệ thống. Các khóa học này là nền tảng để xây dựng các lớp học (Classes), nơi sinh viên và giảng viên tham gia.

## **Phòng học (Rooms)**

- **Đặc điểm chính:**
  - *room\_id*: Mã phòng học (khóa chính).
  - *room\_name*: Tên phòng học, phải duy nhất để không có trùng lặp giữa các phòng.
  - *capacity*: Sức chứa của phòng.
- **Vai trò:** Quản lý các phòng học, đảm bảo phòng đủ sức chứa cho lớp học. Phòng học được gán cho các lớp học để xác định địa điểm dạy và học.

## **Lớp học (Classes)**

- **Đặc điểm chính:**
  - *class\_id*: Mã lớp học (khóa chính, tự động tăng).
  - *course\_id*: Mã khóa học (tham chiếu đến Courses).
  - *teacher\_id*: Mã giảng viên (tham chiếu đến Teachers).
  - *room\_id*: Mã phòng học (tham chiếu đến Rooms).
  - *semester*: Học kỳ mà lớp học diễn ra.
  - *class\_time\_start* và *class\_time\_end*: Thời gian bắt đầu và kết thúc của lớp học.
  - *class\_time\_day*: Ngày trong tuần mà lớp học diễn ra.
- **Vai trò:** Tổ chức các khóa học thành các lớp học cụ thể trong học kỳ, với giảng viên và phòng học xác định. Mỗi lớp học có thời gian cụ thể để sinh viên ghi danh.

## **Ghi danh (Enrollments)**

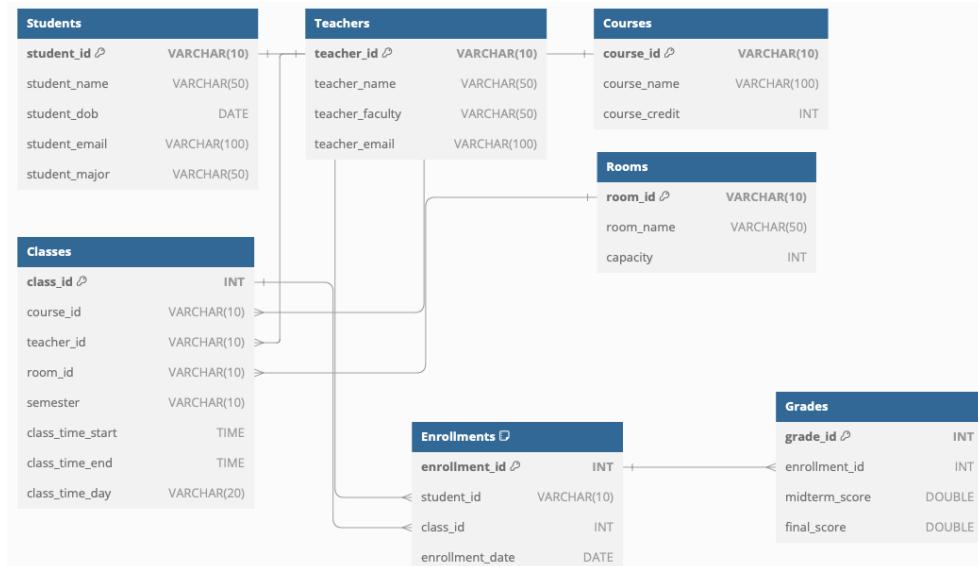
- **Đặc điểm chính:**
  - *enrollment\_id*: Mã ghi danh (khóa chính, tự động tăng).
  - *student\_id*: Mã sinh viên (tham chiếu đến Students).
  - *class\_id*: Mã lớp học (tham chiếu đến Classes).
  - *enrollment\_date*: Ngày ghi danh.
- **Vai trò:** Theo dõi việc ghi danh của sinh viên vào các lớp học. Mỗi sinh viên chỉ có thể ghi danh vào một lớp học duy nhất một lần (kết hợp *student\_id* và *class\_id* là duy nhất).

## **Điểm số (Grades)**

- **Đặc điểm chính:**
  - *grade\_id*: Mã điểm (khóa chính, tự động tăng).
  - *enrollment\_id*: Mã ghi danh (tham chiếu đến Enrollments).
  - *midterm\_score*: Điểm giữa kỳ.
  - *final\_score*: Điểm cuối kỳ.

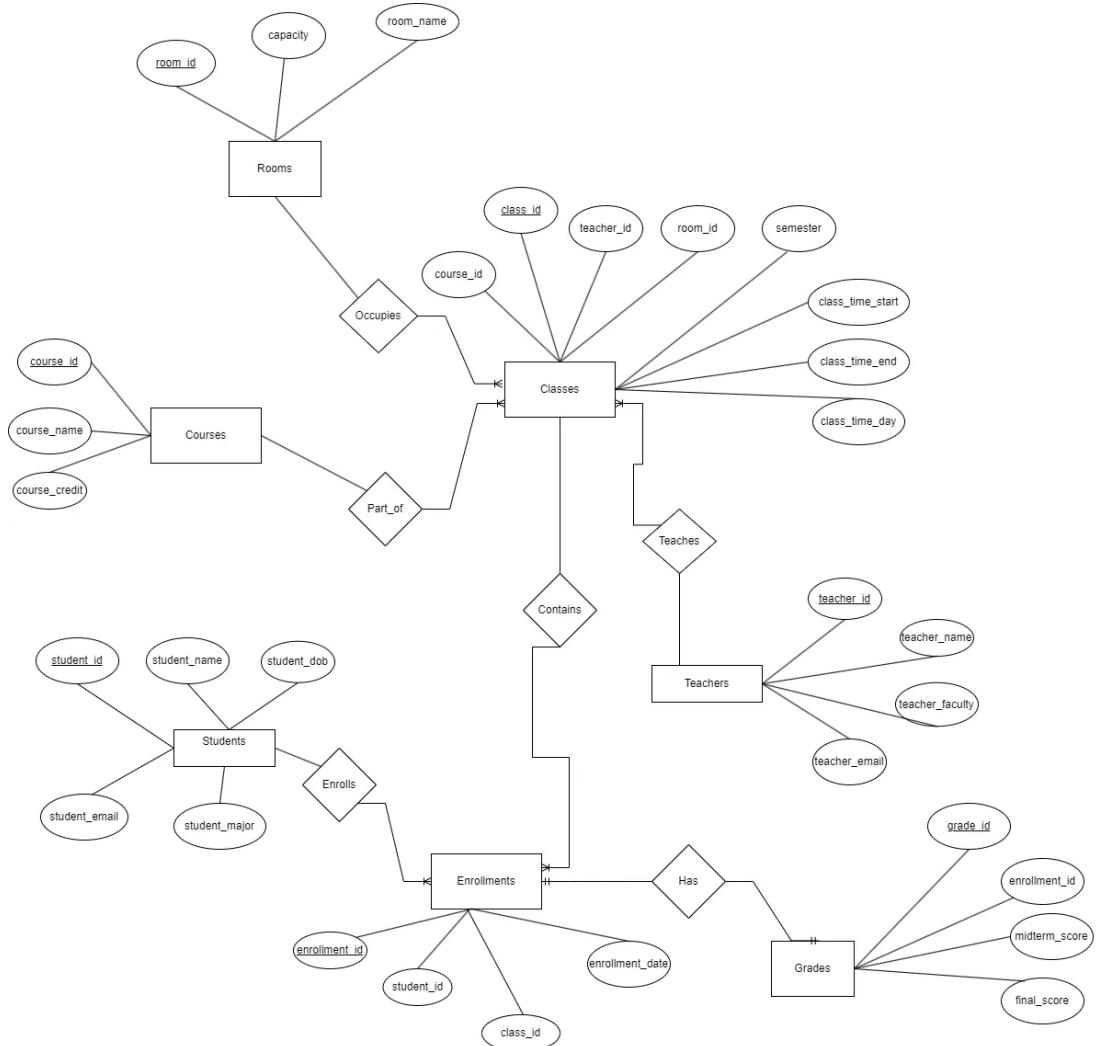
- **Vai trò:** Lưu trữ điểm của sinh viên cho các lớp học mà họ đã ghi danh. Mỗi lớp học sẽ có điểm giữa kỳ và điểm cuối kỳ.

#### 4.2.2.2. Lược đồ cơ sở dữ liệu



Hình 18 Lược đồ cơ sở dữ liệu

#### 4.2.2.3. Sơ đồ thực thể liên kết



Hình 19 Sơ đồ thực thể liên kết

### 4.3. Xây dựng ứng dụng

#### 4.3.1. Thư viện và công cụ sử dụng

Sinh viên liệt kê các công cụ, ngôn ngữ lập trình, API, thư viện, IDE, công cụ kiểm thử, v.v. mà mình sử dụng để phát triển ứng dụng. Mỗi công cụ phải được chỉ rõ phiên bản sử dụng. SV nên kẻ bảng mô tả tương tự như Bảng 1. Nếu có nhiều nội dung trình bày, sinh viên cần xoay ngang bảng.

**Bảng 1** Danh sách thư viện và công cụ sử dụng

Mục đích	Công cụ	Địa chỉ URL
Ngôn ngữ lập trình	Node.js	<a href="https://nodejs.org/">https://nodejs.org/</a>
Framework	Express.js	<a href="https://expressjs.com/">https://expressjs.com/</a>
Template Engine	EJS	<a href="https://ejs.co">https://ejs.co</a>

Quản lý Database	PostgreSQL	<a href="https://www.postgresql.org/">https://www.postgresql.org/</a>
Thư viện kết nối Database	node-postgres (pg)	<a href="https://node-postgres.com/">https://node-postgres.com/</a>
Công cụ kiểm thử	Postman	<a href="https://www.postman.com/">https://www.postman.com/</a>
IDE lập trình	Visual Studio Code	<a href="https://code.visualstudio.com/">https://code.visualstudio.com/</a>
Quản lý phiên bản	Git	<a href="https://git-scm.com/">https://git-scm.com/</a>
Trình quản lý gói	npm	<a href="https://www.npmjs.com/">https://www.npmjs.com/</a>
Thư viện CSS	Font Awesome	<a href="https://fontawesome.com/">https://fontawesome.com/</a>
Công cụ giám sát thay đổi	nodemon	<a href="https://nodemon.io/">https://nodemon.io/</a>

### 4.3.2. Kết quả đạt được

#### Tổng quan

- Tăng cường tự động hóa
- Tự động hóa các quy trình như tính học phí, quản lý đăng ký môn học, và gửi thông báo.
- Tiết kiệm thời gian cho người quản lý và nhân viên hành chính.

#### Các trang chức năng:

- LandingPage
- Quản lý sinh viên
- Quản lý giảng viên
- Quản lý khóa học
- Quản lý điểm số
- Admin

Tên trường	Số lượng
Tổng số lượng dòng code	14,000
Dung lượng toàn bộ mã nguồn	90 MB
Số routes	60
Số gói	10

### 4.3.3. Minh họa các chức năng chính

#### 4.3.1.1. Giao diện chính



Hình 20 Giao diện chính khách

Khi người dùng truy cập vào website HUSTHUB, giao diện trang **Home** sẽ tự động hiển thị, mang đến một trải nghiệm dễ dàng và thân thiện mà không cần phải đăng nhập. Trang này được thiết kế để cung cấp thông tin cơ bản nhưng quan trọng về dự án, đồng thời tạo sự kết nối giữa người dùng và các hoạt động chính của trường Đại học Bách Khoa Hà Nội. Các thành phần chính của trang bao gồm:

- **About Us:**

A screenshot of the 'About Us' page from the HUSTHUB School Management System. The top navigation bar is identical to the homepage. Below it is another large photograph of the university campus. At the bottom of the page, there is a light green section titled 'ABOUT US' in bold capital letters. This section contains four yellow rectangular cards, each with a circular profile picture and the name of a staff member: 'Tran Quang Huy 20226109', 'Ng Luong Hoang Tung 20226129', 'Ngo Manh Hung 20226083', and 'Nguyen Trung Hieu 20226082'. Each card also has a small 'LOG IN' button in the top right corner.

Hình 21 Giao diện phần About us

- **News:**

[HOME](#)    [ABOUT US](#)    [NEWS](#)    [EVENTS](#)    [CONTACT](#)

**News**



Thủ tướng Phạm Minh Chính dự và chúc mừng Đại học Bách Khoa Hà Nội khánh thành BK Smart - Tech Center

Thứ ba - 01/10/2024 16:48



Techconnect & Innovation Vietnam 2024: Thủ trò Bách khoa Hà Nội ứng dụng KHCN - DMST góp phần phát triển Thủ đô, đất nước

Thứ hai - 30/09/2024 20:03



Bách khoa Hà Nội sẵn sàng đồng hành cùng AUF thúc đẩy phát triển KHKT và cộng đồng NCKH Pháp ngữ tại Việt Nam

Thứ tư - 25/09/2024 17:08

[VIEW MORE NEWS](#)

Hình 22 Giao diện hiển thị thông báo

Khu vực này tổng hợp các thông tin mới nhất và đáng chú ý từ website chính thức của Đại học Bách Khoa Hà Nội. Người dùng có thể cập nhật các tin tức liên quan đến hoạt động học tập, nghiên cứu, tuyển sinh, hoặc các thông báo quan trọng. Tính năng này đảm bảo người dùng luôn được tiếp cận với thông tin chính thống và cập nhật kịp thời.

**- Events:**

[HOME](#)    [ABOUT US](#)    [NEWS](#)    [EVENTS](#)    [CONTACT](#)

**Events**



TRIỂN LÃM  
GIÁO DỤC HOA KỲ  
BẬC SAU ĐẠI HỌC - KHỐI NGÀNH STEM

Thứ ba, ngày 1 tháng 10 năm 2024  
15h00 - 18h30

TRUNG TÂM VIỆT ĐỨC  
Hà Nội, Số 100 Nguyễn Văn Hỗn

Thứ ba - 24/09/2024 00:24



Hội thảo tuyển dụng của công ty TNHH LG INNOTEK VN  
Hải Phòng

Thứ hai - 10/06/2024 05:52



Kết quả xét đạt tiêu chuẩn GS, PGS năm 2024 tại Hội đồng Giáo sư cơ sở của ĐHBK Hà Nội

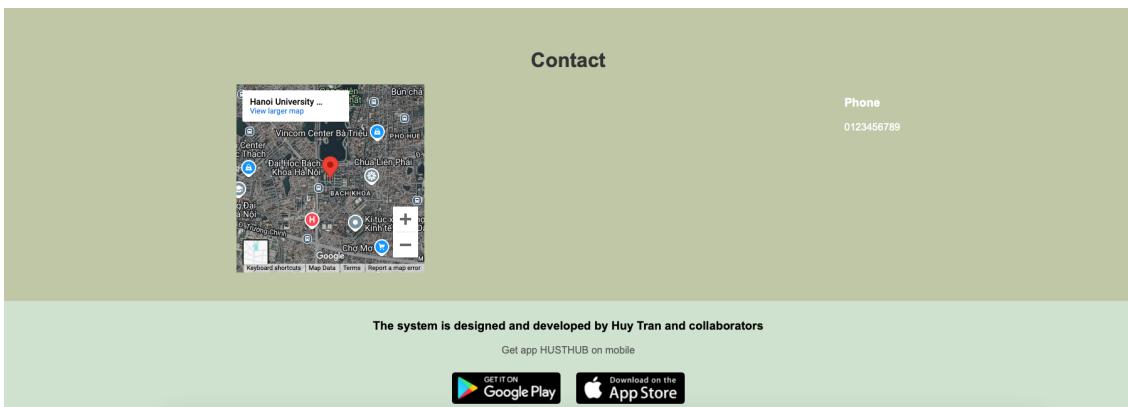
Thứ năm - 18/07/2024 05:11

[VIEW MORE EVENTS](#)

Hình 23 Giao diện hiển thị phần Events

Phần sự kiện cung cấp danh sách các hoạt động và sự kiện nổi bật, bao gồm hội thảo, hoạt động ngoại khóa, các cuộc thi, và sự kiện đặc biệt do nhà trường hoặc các câu lạc bộ tổ chức. Mỗi sự kiện đều có thông tin chi tiết về thời gian, địa điểm, và cách thức tham gia, giúp người dùng dễ dàng lên kế hoạch tham gia và kết nối với cộng đồng.

**- Contact và Footer:**

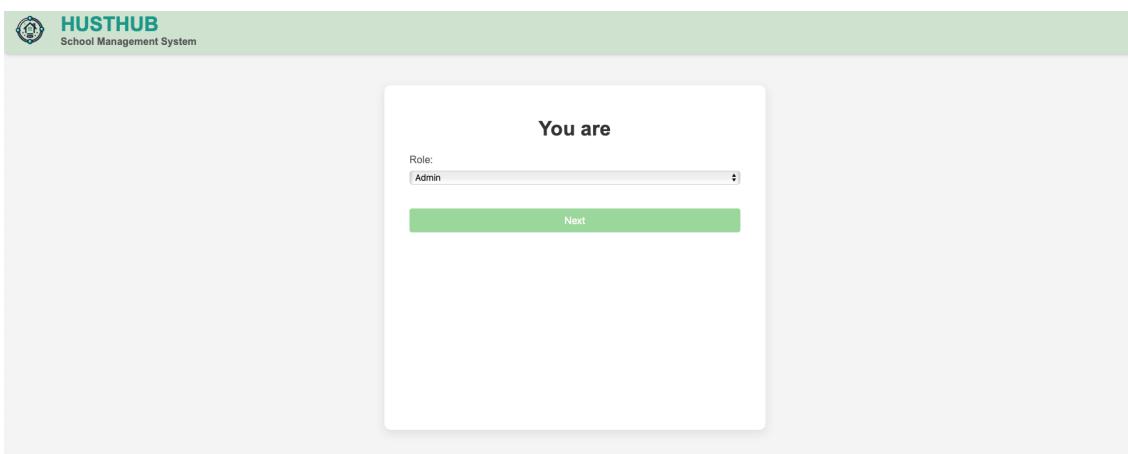


Hình 24 Contact và Footer

Cuối trang là khu vực liên hệ (**Contact**) và phần **Footer**, được thiết kế để xuất hiện xuyên suốt mọi trang trong website. Khu vực liên hệ cung cấp thông tin chi tiết như email, số điện thoại, địa chỉ, và biểu mẫu liên lạc để người dùng gửi phản hồi hoặc yêu cầu hỗ trợ. Footer bao gồm các liên kết quan trọng như Chính sách bảo mật, Điều khoản sử dụng, và thông tin bản quyền, giúp tạo sự minh bạch và chuyên nghiệp cho nền tảng.

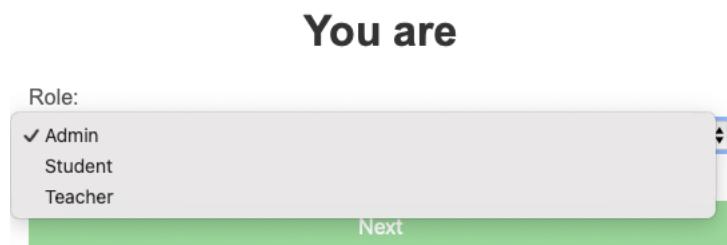
Với thiết kế giao diện trực quan, các tính năng này đảm bảo người dùng dễ dàng tiếp cận thông tin cần thiết ngay từ lần truy cập đầu tiên, đồng thời tạo ấn tượng về một hệ thống hiện đại, thân thiện và gắn kết chặt chẽ với các hoạt động của trường Đại học Bách Khoa Hà Nội.

#### 4.3.1.2. Giao diện đăng nhập



Hình 25 Giao diện đăng nhập

Khi nhấp **Log in**, giao diện đăng nhập dành cho người dùng xuất hiện. Ở đây có 3 lựa chọn tương ứng với 3 đối tượng người dùng mà website hướng đến:



Hình 26 Giao diện chọn role

Hình ảnh cụ thể khi chọn role:

A screenshot of the admin login interface. It features a title "You are admin". Below it are two input fields: "Username:" and "Password:", each with a corresponding input box. At the bottom is a green "Login" button.

Hình 27 Giao diện đăng nhập của admin

A screenshot of the student login interface. It features a title "You are student". Below it are two input fields: "Username:" and "Password:", each with a corresponding input box. At the bottom is a green "Login" button.

Hình 28 Giao diện đăng nhập của sinh viên

# You are teacher

Username:

Password:

**Login**

Hình 29 Giao diện đăng nhập của giáo viên

### 4.3.1.3. Chức năng của Admin

The screenshot shows the Admin dashboard with the following elements:

- GENERAL**: Three cards showing statistics:
  - 1000 Students
  - 301 Teachers
  - 90 Classes
- NOTIFICATION**: A box containing the message "Thông báo nhập điểm lớp Project 1".
- TEACHER**: An icon of a teacher holding books. Below it, text states: "Admins can manage information about teachers, including their name, date of birth, qualifications, fields, and subjects taught." A green button "Go Teacher page" is present.
- Student**: An icon of a student holding a pencil. Below it, text states: "Students have the right to view the timetable, class schedule, lecturers, and academic results. However, they do not have the right to edit any information, including their personal information." A green button "Go Student page" is present.
- Courses**: An icon of a classroom. Below it, text states: "Admins have the authority to create, add, delete, and modify courses, classes, credit hours, timetables, schedules, and course materials. Additionally, admins can enrol students and lecturers into classes and manage class-related information such as grades, attendance, and more." A green button "Go Courses page" is present.

Hình 30. Giao diện chính của admin

## Chức năng quản lí thông tin giảng viên

Teacher ID	Name	Faculty	Email	Actions	
226109	Huy	Kỹ thuật máy tính	Huy.Tq@gmail.com	<button>Edit</button>	<button>Delete</button>
69296	Hồ Việt Hương	Khoa Công nghệ giáo dục1	huong.hv069296@ais.hust.edu.vn	<button>Edit</button>	<button>Delete</button>
31295	Phạm Anh Hương	Khoa Ngoại ngữ	huong.pa031295@ais.hust.edu.vn	<button>Edit</button>	<button>Delete</button>
52219	Vũ Việt Hương	Khoa Cơ khí	huong.vv052219@ais.hust.edu.vn	<button>Edit</button>	<button>Delete</button>
23594	Nguyễn Anh Nam	Khoa Kỹ thuật máy tính	nam.na023594@ais.hust.edu.vn	<button>Edit</button>	<button>Delete</button>
41378	Vũ Xuân Nam	Khoa Điện tử Viễn thông	nam.vx041378@ais.hust.edu.vn	<button>Edit</button>	<button>Delete</button>
27522	Đỗ Việt Tú	Khoa Kỹ thuật máy tính	tu.dv02752@ais.hust.edu.vn	<button>Edit</button>	<button>Delete</button>
57155	Trần Việt Tú	Khoa Cơ khí	tu.tv057755@ais.hust.edu.vn	<button>Edit</button>	<button>Delete</button>

Hình 31 Chức năng quản lí thông tin sinh viên

**Add New Teacher**

**Teacher ID:** \_\_\_\_\_

**Name:** \_\_\_\_\_

**Faculty:** \_\_\_\_\_

**Email:** \_\_\_\_\_

**Add Teacher**

Hình 32 Chức năng thêm giáo viên

**Edit Teacher**

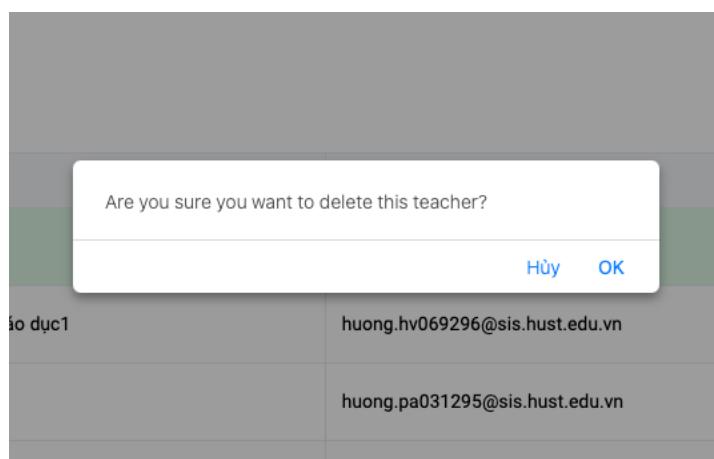
**Name:**  
Phạm Anh Hương

**Faculty:**  
Khoa Ngoại ngữ

**Email:**  
huong.pa031295@sis.hust.edu.vn

[Update Teacher](#)

Hình 33 Chức năng cập nhật giáo viên



Hình 34 Xác nhận xoá giáo viên

### Chức năng quản lý thông tin sinh viên

Chức năng này tương tự như quản lý thông tin giảng viên.

## Chức năng quản lí khoá học

The screenshot shows the 'MANAGE COURSES' page of the HUSTHUB system. At the top, there is a navigation bar with links for HOME, TEACHER, STUDENT, and COURSES, along with a 'Log out' button. Below the navigation bar is a search bar labeled 'Search by Courses ID...'. There are three main buttons: 'CREATE' (green), 'UPDATE' (orange), and 'DELETE' (brown). The main area displays a grid of course cards. Each card contains a course ID (e.g., IT3011, IT3070, IT3080, etc.), the course name, and its credit value (e.g., TC: 3, TC: 5, TC: 5, etc.).

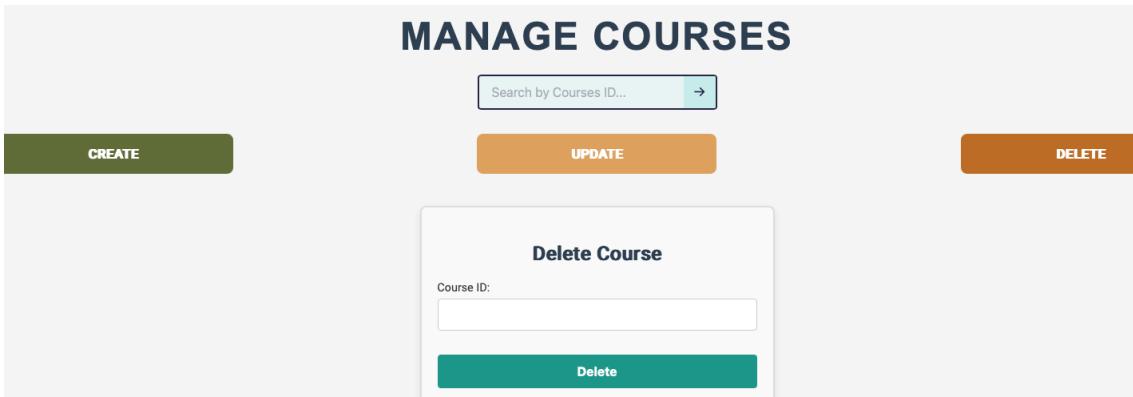
Hình 35 Giao diện quản lí khoá học

The screenshot shows the 'Create Course' form within the 'MANAGE COURSES' interface. The title 'Create Course' is at the top. It includes fields for 'Course ID' (with a placeholder box), 'Course Name' (with a placeholder box), and 'Course Credit' (with a dropdown menu). A large green 'Create' button is at the bottom.

Hình 36 Chức năng thêm khoá học

The screenshot shows the 'Update Course' form within the 'MANAGE COURSES' interface. The title 'Update Course' is at the top. It includes fields for 'Course ID' (with a placeholder box), 'Course Name' (with a placeholder box), and 'Course Credit' (with a dropdown menu). A large green 'Update' button is at the bottom.

Hình 37 Chức năng cập nhật khoá học



Hình 38 Chức năng xoá khóa học

Class ID	Teacher Name	Date	Time	Location
160008	Đặng Ngọc Anh	Thứ 4	09:20:00 - 11:45:00	D8-404
160009	Hồ Ngọc Tú	Thứ 5	12:30:00 - 15:00:00	B1-502
160007	Phạm Thị Trang	Thứ 3	06:45:00 - 10:05:00	
160010	Đỗ Việt Tú	Thứ 6	15:05:00 - 17:30:00	

Hình 39 Giao diện quản lý lớp học

#### 4.3.1.4. Chức năng của giáo viên

Welcome, Lê Văn Tú

Khoa Khoa học máy tính

Classes: 9 Classes

My Profile: View and update your profile

The system is designed and developed by Huy Tran and collaborators

Get app HUSTHUB on mobile

[GET IT ON Google Play](#) [Download on the App Store](#)

Hình 40 Giao diện người dùng giáo viên

## My Classes

Search classes by ID, course ID, or						
Class ID	Course ID	Course Name	Time	Room	Action	
160004	IT3011	Cấu trúc dữ liệu và thuật toán	Thứ 5, 12:30:00 - 15:00:00	B1-501	<a href="#">Detail</a>	
160019	IT3103	Lập trình hướng đối tượng	Thứ 5, 12:30:00 - 15:00:00	C7-304	<a href="#">Detail</a>	
160034	IT3160	Nhập môn Tri tuệ nhân tạo	Thứ 5, 12:30:00 - 15:00:00	B1-301	<a href="#">Detail</a>	
160036	IT3180	Nhập môn Công nghệ phần mềm	Thứ 2, 06:45:00 - 09:10:00	D9-503	<a href="#">Detail</a>	
160046	IT3220	C Programming (Introduction)	Thứ 2, 06:45:00 - 09:10:00	D9-504	<a href="#">Detail</a>	
160058	IT3280	Thực hành kiến trúc máy tính	Thứ 4, 09:20:00 - 11:45:00	B1-203	<a href="#">Detail</a>	
160064	IT3283	Kiến trúc máy tính	Thứ 5, 12:30:00 - 15:00:00	B1-402	<a href="#">Detail</a>	
160070	IT3290	Thực hành cơ sở dữ liệu	Thứ 6, 15:05:00 - 17:30:00	D9-504	<a href="#">Detail</a>	
160074	IT3292	Cơ sở dữ liệu	Thứ 5, 12:30:00 - 15:00:00	B1-201	<a href="#">Detail</a>	

Hình 41 Giao diện quản lí lớp giảng dạy

The screenshot shows the 'Class Details' page of the HUSTHUB system. At the top, there's a header bar with the logo 'HUSTHUB School Management System' and a 'Log out' button. Below the header, there are navigation links for 'HOME' and 'CLASSES'. A blue button labeled 'Back to All Classes' is visible. The main content area is titled 'Class Details' and shows the following information for Class ID 160001:

- Class ID: 160001
- Course Name: Cấu trúc dữ liệu và thuật toán
- Teacher: Nguyễn Anh Nam
- Time: Thứ 2, 06:45:00 - 09:10:00
- Room: D9-501

Below this, there's a section titled 'Students' which lists eight students with their details and edit buttons:

Student ID	Student Name	Email	Midterm Score	Final Score	Actions
20236083	Lê Quang Anh	anh.lq236083@sis.hust.edu.vn	5	10	<a href="#">Edit</a>
20204086	Trần Quang Dũng	dung.tq204086@sis.hust.edu.vn	9	9	<a href="#">Edit</a>
20244914	Hồ Xuân Nam	nam.hx244914@sis.hust.edu.vn	8	7	<a href="#">Edit</a>
20215326	Hoàng Minh Nam	nam.hm215326@sis.hust.edu.vn	4	6	<a href="#">Edit</a>
20243335	Đỗ Huy Thảo	thao.dh243335@sis.hust.edu.vn	10	10	<a href="#">Edit</a>
20248721	Phạm Huy Thảo	thao.ph248721@sis.hust.edu.vn	4	1	<a href="#">Edit</a>
20227723	Lê Ngọc Thảo	thao.ln227723@sis.hust.edu.vn	2	2	<a href="#">Edit</a>
20233893	Bùi Văn Trang	trang.bv233893@sis.hust.edu.vn	5	5	<a href="#">Edit</a>

Hình 42 Giao diện quản lí sinh viên của giảng viên

**Class Details**

Class ID: 160001  
 Course Name: Cấu trúc dữ liệu và thuật toán  
 Teacher: Nguyễn Anh Nam  
 Time: Thứ 2, 06:45:00 - 09:10:00  
 Room: D9-501

Student ID	Student Name	Midterm Score	Final Score	Actions
20236083	Lê Quang Anh	5	10	<button>Edit</button>
20204088	Trần Quang Dũng	9	9	<button>Edit</button>
20244914	Hồ Xuân Nam	8	7	<button>Edit</button>
20215328	Hoàng Minh Nam	4	6	<button>Edit</button>
20243335	Đỗ Huy Thảo	10	10	<button>Edit</button>
20248721	Phạm Huy Thảo	4	1	<button>Edit</button>
20227723	Lê Ngọc Thảo	2	2	<button>Edit</button>
20233893	Bùi Văn Trang	5	5	<button>Edit</button>

**Edit Grades**

Student Name: Lê Quang Anh

Midterm Score: 5  
Final Score: 10

**Submit**

Hình 43 Giao diện nhập điểm của giáo viên

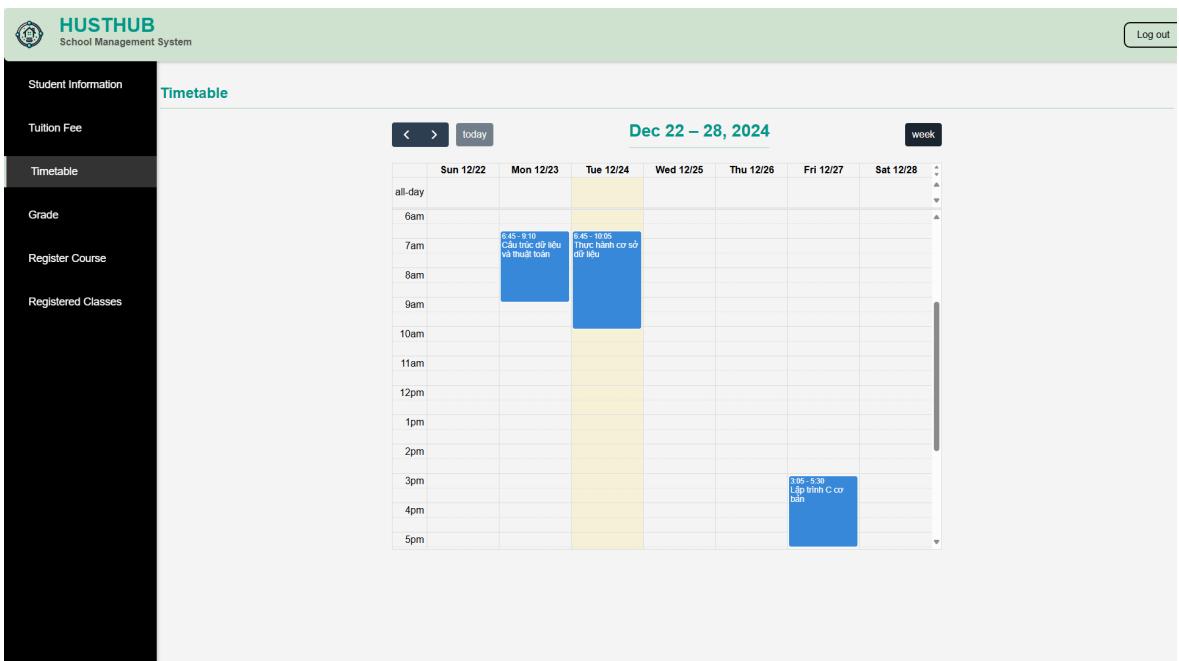
#### 4.3.1.5. Chức năng của sinh viên

**Student Information**

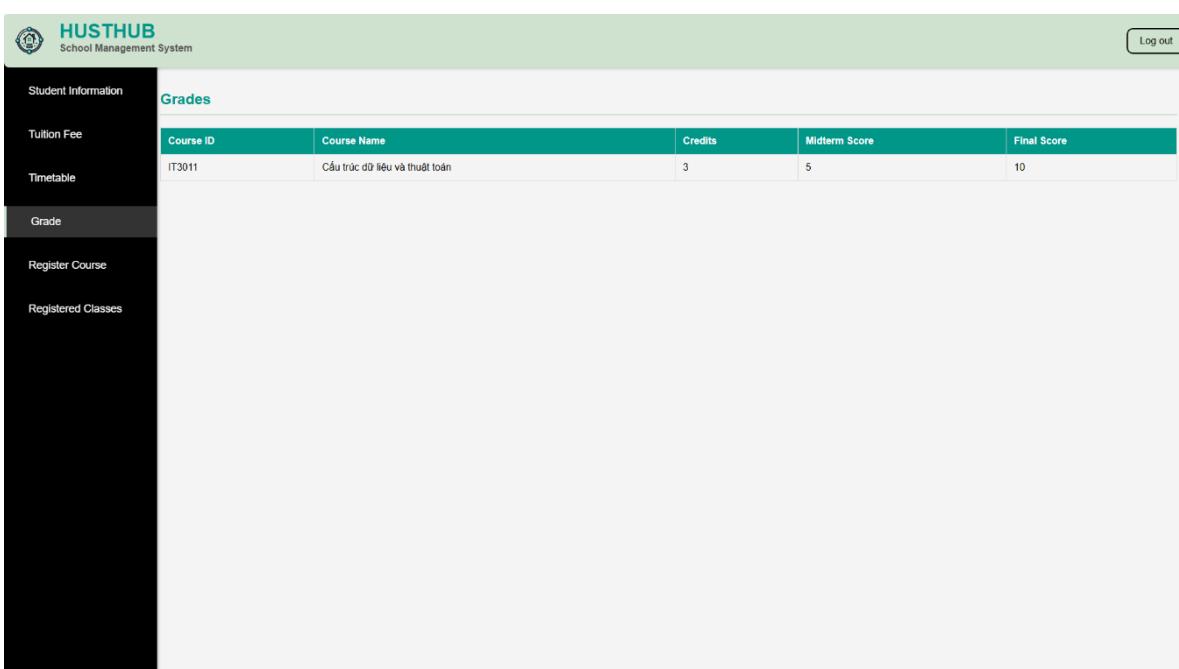
Student ID: 20236083

<b>Student Photo</b>	Full Name: Lê Quang Anh	Date of Birth: 25/10/2005
<b>Choose File</b> <b>Change Avatar</b>	Academic Year: 2023	Email: anh.lq236083@sis.hust.edu.vn
	Major: IT-E15	University: Hanoi University of Science and Technology

Hình 44 Giao diện hiển thị của sinh viên



Hình 45 Thời khóa biểu cho sinh viên



Hình 46 Giao diện hiển thị điểm số của sinh viên

The screenshot shows the 'Course Catalog' section of the HUSTHUB system. On the left sidebar, under 'Registered Classes', there is a 'Project I' card. The main area displays a grid of course cards:

- Cấu trúc dữ liệu và thuật toán** (3 Credits)
- Nguyên lý hệ điều hành** (5 Credits)
- Mạng máy tính** (5 Credits)
- Lập trình hướng đối tượng** (5 Credits)
- Phân tích và thiết kế hệ thống** (6 Credits)
- Project II** (5 Credits)
- Nhập môn Trí tuệ nhân tạo** (6 Credits)
- Nhập môn công nghệ phần mềm** (5 Credits)
- C Programming Language** (2 Credits)
- C Programming (Introduction)** (2 Credits)
- Lập trình C cơ bản** (3 Credits)
- Thực hành kiến trúc máy tính** (4 Credits)
- Kiến trúc máy tính** (4 Credits)
- Thực hành cơ sở dữ liệu** (4 Credits)
- Cơ sở dữ liệu** (4 Credits)
- Project III** (7 Credits)
- Nhập môn an toàn thông tin** (6 Credits)
- Ngôn ngữ và phương pháp dịch** (8 Credits)
- Quản trị phát triển phần mềm** (6 Credits)
- Course ID: IT3150**
- Course ID: IT3160**
- Course ID: IT3180**
- Course ID: IT3210**
- Course ID: IT3220**
- Course ID: IT3230**
- Course ID: IT3280**
- Course ID: IT3290**
- Course ID: IT3292**
- Course ID: IT3930**
- Course ID: IT3940**
- Course ID: IT4015**
- Course ID: IT4079**

Each card includes a 'View Classes' button.

Hình 47 Giao diện đăng ký khóa học

The screenshot shows the 'Registered Classes' section of the HUSTHUB system. On the left sidebar, under 'Registered Classes', there are three entries:

- Project I**: Course: Cấu trúc dữ liệu và thuật toán, Class ID: 160001, Professor: Nguyễn Anh Nam, Room: D9-501, Time: 06:45:00 - 09:10:00, Day: Thứ 2. Includes a 'Remove' button.
- Lập trình C cơ bản**: Course: Lập trình C cơ bản, Class ID: 160055, Professor: Bùi Thị Anh, Room: D9-408, Time: 15:05:00 - 17:30:00, Day: Thứ 6. Includes a 'Remove' button.
- Thực hành cơ sở dữ liệu**: Course: Thực hành cơ sở dữ liệu, Class ID: 160067, Professor: Phạm Thị Huy, Room: D8-403, Time: 06:45:00 - 10:05:00, Day: Thứ 3. Includes a 'Remove' button.

Hình 48 Giao diện hiển thị các lớp đã đăng ký

## 4.4. Kiểm thử

### 4.4.1. Chức năng quản lý thông tin sinh viên

#### a) Thêm giảng viên mới

**Kỹ thuật kiểm thử:** Kiểm thử hộp đen với kỹ thuật phân vùng tương đương

### **Phân vùng dữ liệu:**

- Hợp lệ:
  - Teacher ID: Số nguyên, không rỗng (VD: 12345).
  - Name: Chuỗi ký tự hợp lệ, không rỗng (VD: Nguyễn Văn A).
  - Faculty: Chuỗi ký tự hợp lệ, không rỗng (VD: CNTT, Toán).
  - Email: Đúng định dạng (VD: [teacher@domain.com](mailto:teacher@domain.com)).
- Không hợp lệ:
  - Teacher ID: Rỗng, chứa ký tự đặc biệt (VD: "", "123@", "-456").
  - Name: Rỗng, chứa ký tự không hợp lệ (VD: "", "!@#\$%^").
  - Faculty: Rỗng, chứa ký tự không hợp lệ (VD: "", "!@#\$%^").
  - Email: Sai định dạng (VD: "abc.com", "@domain.com", "abc@").

### **Test case:**

Test Case ID	Input	Expected Output	Result
TC01	Teacher ID: 12345	Thêm thành công.	PASS
	Name: Nguyễn Văn A		
	Faculty: Khoa học máy tính		
	Email: teacher@domain.com		
TC02	Teacher ID: ""	Hiển thị thông báo ở ô Teacher ID: "Điền vào trường này"	PASS
	Name: Nguyễn Văn A		
	Faculty: Khoa học máy tính		
	Email: teacher@domain.com		
TC03	Teacher ID: 12345	Hiển thị thông báo ở ô Name: "Điền vào trường này"	PASS
	Name: ""		
	Faculty: Khoa học máy tính		
	Email: teacher@domain.com		
TC04	Teacher ID: 12345	Hiển thị thông báo ở ô Faculty: "Điền vào trường này"	PASS
	Name: Nguyễn Văn A		
	Faculty: ""		
	Email: teacher@domain.com		
TC05	Teacher ID: 12345	Hiển thị thông báo ở ô Email: "Nhập địa chỉ email"	PASS
	Name: Nguyễn Văn A		
	Faculty: Khoa học máy tính		
	Email: abc.com		

### **b) Xoá giảng viên**

**Kỹ thuật kiểm thử:** Kiểm thử hộp đen với kỹ thuật Decision Table

### **Điều kiện đầu vào**

- Người dùng nhấn nút "Delete": Y (Yes) hoặc N (No).
- Người dùng xác nhận trong hộp thoại: Y (Yes) hoặc N (No).

### **Hành động**

- Xóa giáo viên: Giáo viên bị xóa khỏi danh sách.
- Giữ nguyên danh sách: Không thay đổi gì trong danh sách giáo viên.

#### Decision Table

Case	Nhấn nút Delete	Xác nhận OK	Hành động	Kết quả mong đợi
1	Y	Y	Xóa giáo viên	Giáo viên bị xóa khỏi danh sách.
2	Y	N	Giữ nguyên danh sách	Giáo viên không bị xóa.
3	N	-	Giữ nguyên danh sách	Không có gì thay đổi.

#### c) Cập nhật thông tin giảng viên

Phương pháp kiểm thử và phân vùng dữ liệu giống mục 4.4.1.a nên nhóm dự án xin phép không trình bày kiểm thử tính năng này.

#### 4.4.2. Chức năng đăng ký lớp của sinh viên

**Kỹ thuật kiểm thử:** Kiểm thử hộp đen với kỹ thuật Decision Table

##### Điều kiện kiểm tra

- Lớp đã đăng ký trước đó (cùng học phần): Yes (Y) hoặc No (N).
- Trùng thời gian với lớp đã đăng ký: Yes (Y) hoặc No (N).

##### Hành động

- Cho phép đăng ký: Thêm lớp học vào danh sách đã đăng ký.
- Hiển thị lỗi trùng học phần: "You are already registered for a class in this course."
- Hiển thị lỗi trùng thời gian: "Time conflict with another registered class."

#### Decision Table

Case	Kiểm tra lớp đã đăng ký trước đó	Kiểm tra trùng thời gian	Hành động	Kết quả mong đợi
1	N	N	Cho phép đăng ký	Hiển thị: "Successfully registered for the class!"

2	Y	-	Hiển thị lỗi trùng học phần	Hiển thị: "You are already registered for a class in this course."
3	N	Y	Hiển thị lỗi trùng thời gian	Hiển thị: "Time conflict with another registered class."

## 4.5. Triển khai

### 4.5.1. Mô hình triển khai

- **Kiến trúc hệ thống**
  - + Frontend: HTML, CSS, JS
  - + Backend: Node JS
  - + Database: PostgreSQL
  - + Payment method: Payos
- **Môi trường triển khai:**
  - + Laptop/PC: Windows/MAC
  - + RAM: 8GB (trở lên)
  - + Processor: Core i5 (hoặc tương đương)
- **Phần mềm cần thiết:**
  - + IDE (VS Code)
  - + Git
  - + pgAdmin 4 (Dùng để quản lý database)
  - + Nodejs v16+

### 4.5.2. Cài đặt môi trường:

- Node.js và npm: Đảm bảo bạn đã cài đặt Node.js và npm trên máy tính. Bạn có thể tải về và cài đặt từ trang chủ Node.js: <https://nodejs.org/>

- Git: Cài đặt Git để clone dự án từ repository. Bạn có thể tải về và cài đặt từ trang chủ Git: <https://git-scm.com/>

#### 4.5.3. Clone dự án

Mở terminal và chạy lệnh sau để clone dự án về máy tính:

Bash

```
git clone https://github.com/nlhtungg/Project_Group6_IT3180_20241.git
```

#### 4.5.4. Thiết lập biến môi trường:

- Tạo database mới trên hệ thống quản trị cơ sở dữ liệu MongodB.
- Cấu hình đường dẫn kết nối database, và các biến môi trường liên quan trong file .env ở thư mục server và admin.

#### 4.5.5. Chạy dự án

Bash

```
node app.js
```

#### 4.5.6. Truy cập ứng dụng:

- Mở trình duyệt và truy cập vào địa chỉ <http://localhost:3000> để sử dụng giao diện khách và thao tác.

## 5. Kết luận và hướng phát triển

### 5.1. Kết luận

Sau quá trình thực hiện, nhóm đã hoàn thành dự án HUSTHub theo kế hoạch đề ra với các chức năng cơ bản đáp ứng được yêu cầu của người dùng. Hệ thống hiện đã được triển khai thành công trên môi trường cục bộ (localhost) và cho thấy sự ổn định trong các bài kiểm thử. Các chức năng nổi bật bao gồm:

- **Quản lý thông tin sinh viên và giảng viên:** Hệ thống lưu trữ và cho phép cập nhật thông tin cá nhân chính xác, giúp giảm thiểu sai sót so với quản lý thủ công trước đây.
- **Quản lý thời khóa biểu:** Thời khóa biểu cá nhân hóa cho từng sinh viên và giảng viên đã được xây dựng và đồng bộ hóa, đảm bảo tính linh hoạt khi có thay đổi. Tuy nhiên giao diện người dùng vẫn chưa thực sự trực quan.
- **Quản lý điểm số:** Điểm số được nhập và truy xuất dễ dàng qua giao diện trực tuyến, tăng tính minh bạch và giảm thời gian xử lý.
- **Quản lý học phí:** Sinh viên có thể tra cứu trạng thái học phí và thực hiện thanh toán trực tuyến trên hệ thống, đơn giản hóa quy trình quản lý tài chính.

### 5.2. Hướng phát triển

- **Mở rộng phạm vi triển khai:** Triển khai hệ thống trên máy chủ thực (online) để hỗ trợ truy cập từ xa và sử dụng quy mô lớn. Cân nhắc tích hợp thêm dịch vụ đám mây (cloud-based) để tăng khả năng mở rộng và tính bảo mật.
- **Cải thiện hiệu suất và bảo mật:** Tối ưu hóa cơ sở dữ liệu để xử lý lượng lớn dữ liệu hiệu quả hơn. Nâng cấp các biện pháp bảo mật, như mã hóa dữ liệu, bảo vệ chống tấn công DDoS, và xác thực đa yếu tố.
- **Tích hợp các công nghệ hiện đại:** Sử dụng trí tuệ nhân tạo (AI) để phân tích dữ liệu học tập và đề xuất lộ trình học tập phù hợp cho sinh viên.
- **Bổ sung các chức năng tùy chọn:** Tính năng thông báo tự động cho sinh viên về học phí, thời khóa biểu, và kết quả học tập. Hỗ trợ học tập trực tuyến qua tích hợp các nền tảng như Zoom hoặc Microsoft Teams.
- **Cải thiện trải nghiệm người dùng (UX/UI):** Tăng tính trực quan cho giao diện người dùng, đảm bảo người dùng dễ dàng thao tác.