**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP.HCM**



**quản lí ĐĂNG KÝ HỌC PHẦN**

**IUH LEARN**

***Nhóm 04 - Sinh viên thực hiện***

1. Trương Thị Cẩm Nhung - 20083771
2. Lưu Thành Nhân - 200666711
3. Vũ Tiến Đức - 20089731

**Nội Dung**

[CHƯƠNG 1 : GIỚI THIỆU 2](#_Toc167041907)

[1.1 Tổng quan 2](#_Toc167041908)

[1.2 Mục tiêu đề tài 2](#_Toc167041909)

[1.3 Phạm vi đề tài 3](#_Toc167041910)

[1.3.1.2 Chức năng dành cho người quản trị 3](#_Toc167041911)

[1.4 Mô tả yêu cầu chức năng 3](#_Toc167041912)

[1.4.1.2 Mô tả hoạt động của hệ thống 4](#_Toc167041913)

[CHƯƠNG 2 : CƠ SỞ LÝ THUYẾT 5](#_Toc167041914)

[2.1 Những Công Nghệ Sử Dụng 5](#_Toc167041915)

[2.1.1 NodeJS 5](#_Toc167041916)

[2.1.2 NextJS 6](#_Toc167041917)

[2.1.3 MongoDB 7](#_Toc167041918)

[2.1.4 MySQL 9](#_Toc167041919)

[2.1.5 Spring Boot 10](#_Toc167041920)

[2.2 Kiến Trúc Phần Mềm 12](#_Toc167041921)

[2.2.1 Auth Service 13](#_Toc167041922)

[2.2.2 Manage Learning Service 14](#_Toc167041923)

[2.2.3 Study Service 14](#_Toc167041924)

[2.2.4 SignUp Module Service 15](#_Toc167041925)

[2.2.5 Tại Sao Chúng Tôi Thiết Kế Theo Mô Hình Này? 15](#_Toc167041926)

[CHƯƠNG 3 : PHÂN TÍCH THIẾT KẾ HỆ THỐNG 15](#_Toc167041927)

[3.1 Phân Tích Hệ Thống 15](#_Toc167041928)

[3.2 Cơ sở dữ liệu MySQL (Quản lý các thực thể) 20](#_Toc167041929)

[3.3 Cơ sở dữ liệu của MongoDB (Quản lý các thông tin thêm của học phần) 21](#_Toc167041930)

[3.4 Cơ sở dữ liệu của MongoDB (Quản lý đăng ký học phần) 22](#_Toc167041931)

[CHƯƠNG 4 : PHÂN TÍCH GIAO DIỆN ĐỒ HỌA 22](#_Toc167041932)

[4.1 Giao Diện Đăng Nhập 22](#_Toc167041933)

[4.2 Giao Diện Quản Lý Sinh Viên 23](#_Toc167041934)

[4.3 Giao Diện Quản Lý Khoa 23](#_Toc167041935)

[4.4 Giao Diện Quản Lý Chuyên Ngành 24](#_Toc167041936)

[4.5 Giao Diện Quản Lý Hệ Đào Tạo 24](#_Toc167041937)

[4.6 Giao Diện Quản Lý Phòng 25](#_Toc167041938)

[4.7 Giao Diện Quản Lý Lớp 25](#_Toc167041939)

[4.8 Giao Diện Quản Lý Giáo Viên 26](#_Toc167041940)

[4.9 Giao Diện Quản Lý Môn Học 26](#_Toc167041941)

[4.10 Giao Diện Quản Lý Học Kỳ 27](#_Toc167041942)

[4.11 Giao Diện Quản Lý Học Phần 27](#_Toc167041943)

[4.12 Giao Diện Đăng Nhập Mã Sinh Viên 28](#_Toc167041944)

[4.13 Giao Diện Thông Tin Chung 28](#_Toc167041945)

[4.14 Giao Diện Đăng Ký Học Phần 29](#_Toc167041946)

[CHƯƠNG 5 : MỤC TIÊU VÀ HƯỚNG PHÁT TRIỂN 30](#_Toc167041947)

[5.1 Mục Tiêu và Giải Pháp 30](#_Toc167041948)

[5.2 Kiểm Chứng và Hạn Chế 30](#_Toc167041949)

[5.3 Kế Hoạch Tương Lai 30](#_Toc167041950)

# : GIỚI THIỆU

## Tổng quan

Hệ thống quản lý lớp học tín chỉ của trường đại học được thiết kế với mục tiêu cung cấp cho sinh viên một trải nghiệm học tập linh hoạt và tiện lợi. Hệ thống này không chỉ giúp sinh viên dễ dàng truy cập và đăng ký các khóa học mà còn hỗ trợ họ trong việc quản lý lịch trình học tập và theo dõi tiến trình học tập của mình. Bên cạnh việc đăng ký môn học, hệ thống còn cung cấp các tính năng hỗ trợ cho việc chuẩn bị và xin cấp bằng tốt nghiệp sau khi sinh viên hoàn thành chương trình học. Điều này giúp tạo ra một môi trường học tập hiệu quả, nơi mà sinh viên có thể linh hoạt quản lý và điều chỉnh lộ trình học tập của mình theo ý muốn và nhu cầu học vấn cá nhân. Với hệ thống này, sinh viên không chỉ có được sự linh hoạt và thuận tiện trong quá trình học tập mà còn được khuyến khích và hỗ trợ để phát triển sự nghiệp sau này. Điều này giúp tạo ra một môi trường học tập đa dạng và phong phú, đồng thời đảm bảo rằng sinh viên có mọi công cụ và nguồn lực cần thiết để thành công trong học tập và sự nghiệp của mình.

## Mục tiêu đề tài

* Hệ thống quản lý lớp học tín chỉ nhằm mang lại trải nghiệm học tập linh hoạt và hiệu quả cho sinh viên. Nó giúp sinh viên dễ dàng đăng ký các khóa học phù hợp với ngành học của mình và quản lý lịch trình học tập một cách thuận tiện. Sinh viên cũng có thể xem tiến trình học tập, chương trình khung và đăng ký học phần một cách thuận lợi thông qua nền tảng website. Phần mở rộng của hệ thống có thể bao gồm phân tích dữ liệu học tập để cá nhân hóa trải nghiệm học tập và kết nối sinh viên với cơ hội thực tập và việc làm sau này.

## Phạm vi đề tài

#### Chức năng dành cho người dùng:

* Sinh Viên xem thông tin cá nhân, đăng ký học phần, xem lịch học, xem kết quả học tập, tra cứu công nợ, xem lịch học theo tiến độ, tiến trình học tập, xem chương trình khung.
* Có giao diện trực quan và thân thiện, dễ tương tác và sử dụng thuận tiện.
* Có thể thay đổi : Thông tin tài khoản,và thông tin cá nhân,…

#### **Chức năng dành cho người quản trị**:

Ngoài các yêu cầu giống như của người dùng, thì hệ thống phải đảm bảo những yêu cầu sau:

* Thêm, xóa, chỉnh sửa học phần.

## Mô tả yêu cầu chức năng

#### Phân tích yêu cầu của hệ thống

- Người dùng có thể thay đổi thông tin cá nhân như: tên, giới tính…

- Người dùng thực hiện đăng ký học phần trên web, hệ thống sẽ tiếp nhận và phản hồi cho đối tượng đăng kí học phần.

- Người dùng có thể hủy đăng ký học phần( nếu hệ thống cho phép).

- Từ mô tả trên, có thể đưa ra yêu cầu của hệ thống với hai đối tượng chính tương tác với hệ thống như sau:

• Đối với hệ thống:

- Thêm, sửa, xóa các học phần

- Tìm kiếm thông tin của học phần

- Quản lý thông tin của người dùng

• Đối với người dùng:

- Đăng kí học phần,xem lịch học, xem điểm, tra cứu công nợ, xem lịch học theo tiến độ, tiến trình học tập, xem chương trình khung, thanh toán trực tuyến, xem và chỉnh sửa thông tin cá nhân.

- Hủy đăng ký học phần

#### Mô tả hoạt động của hệ thống

1. **Chọn môn học cho học kỳ**: Nhà trường lựa chọn danh sách môn học cần mở cho mỗi ngành học và nhập vào hệ thống.
2. **Đăng ký học phần**: Sinh viên đăng nhập vào hệ thống, xem danh sách các môn học có sẵn và đăng ký các môn phù hợp cho học kỳ.
3. **Xác nhận đăng ký**: Sinh viên xác nhận đăng ký trước khi hệ thống chấp nhận để tránh đăng ký nhầm lẫn hoặc hủy bỏ sau đó.
4. **Xử lý đăng ký**: Hệ thống xử lý đăng ký, kiểm tra số tín chỉ và quản lý số lượng sinh viên trong từng lớp học.

# : CƠ SỞ LÝ THUYẾTTop of Form

## Những Công Nghệ Sử Dụng

### NodeJS



Hình 2‑1 Node.js development

**Tổng quan về Node.js:**

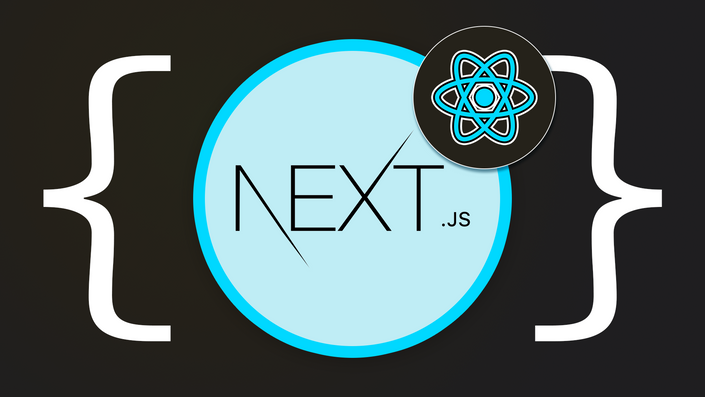
Node.js là một nền tảng phát triển độc lập, được xây dựng dựa trên môi trường thực thi JavaScript của Google Chrome. Nền tảng này cho phép phát triển các ứng dụng mạng một cách nhanh chóng và dễ dàng mở rộng.

**Ưu điểm của Node.js:**

1. **Đơn ngôn ngữ:** Node.js sử dụng JavaScript, một trong những ngôn ngữ lập trình phổ biến nhất, cho phép phát triển ứng dụng đồng nhất từ client đến server.
2. **Kiến trúc hiệu quả:** Với kiến trúc hướng sự kiện và cơ chế non-blocking I/O, Node.js mang lại hiệu suất cao và khả năng xử lý đồng thời lớn, làm cho nó trở thành lựa chọn lý tưởng cho các ứng dụng cần xử lý nhiều kết nối cùng một lúc.
3. **Đa nền tảng:** Node.js có thể chạy trên nhiều hệ điều hành khác nhau như macOS, Windows và Linux, đảm bảo tính linh hoạt và dễ dàng triển khai trong môi trường phát triển đa dạng.
4. **Cộng đồng mạnh mẽ:** Có một cộng đồng lớn và tích cực hỗ trợ Node.js, với hàng ngàn mô-đun miễn phí có sẵn trên npm (Node Package Manager), giúp tăng tốc độ phát triển và cung cấp các giải pháp sẵn có cho nhiều vấn đề phức tạp.
5. **Miễn phí:** Node.js là một nền tảng mã nguồn mở, cung cấp miễn phí cho các nhà phát triển, giúp giảm chi phí phát triển phần mềm.

Nhờ vào những ưu điểm này, Node.js đã trở thành một trong những lựa chọn hàng đầu cho các nhà phát triển web và ứng dụng di động trên toàn thế giới.

### NextJS



**Tổng quan:**

Next.js là một framework phát triển web được xây dựng trên nền tảng React, cung cấp nhiều tính năng mạnh mẽ giúp tối ưu hóa hiệu suất và trải nghiệm người dùng. Next.js hỗ trợ render phía server (Server-Side Rendering - SSR) và tạo trang tĩnh (Static Site Generation - SSG), giúp cải thiện tốc độ tải trang và SEO. Nó cũng cung cấp hệ thống routing tự động, giúp đơn giản hóa việc phát triển ứng dụng React.

**Ưu điểm:**

* Môi trường phát triển với hot-reloading giúp lập trình viên thấy ngay những thay đổi mà không cần reload trang.
* Tích hợp sẵn với TypeScript, tăng cường độ tin cậy và giảm lỗi trong quá trình phát triển.
* Hệ thống routing dựa trên cấu trúc thư mục giúp định tuyến dễ dàng mà không cần cấu hình phức tạp.
* Cho phép tạo các API endpoint trực tiếp trong ứng dụng, giúp xây dựng ứng dụng full-stack chỉ với một framework.

**Nhược điểm:**

* Cấu hình Next.js phức tạp hơn so với chỉ dùng React, cần hiểu rõ về SSR, SSG và tối ưu hóa chúng.
* Quản lý trạng thái trong ứng dụng Next.js có thể phức tạp, đặc biệt khi cần phối hợp giữa SSR và client-side state.
* SSR yêu cầu server để render nội dung, tăng chi phí vận hành và phức tạp hóa hệ thống hạ tầng, đặc biệt với các ứng dụng lớn.

### MongoDB



**Tổng quan về MongoDB:**

MongoDB là một hệ thống quản lý cơ sở dữ liệu (DBMS) mã nguồn mở, được thiết kế với mô hình cơ sở dữ liệu hướng đối tượng. Đặc điểm nổi bật của MongoDB là khả năng lưu trữ dữ liệu một cách linh hoạt mà không yêu cầu phải tuân theo một cấu trúc nhất định.

**Ưu điểm của MongoDB:**

1. **Hiệu suất cao:** MongoDB được tối ưu hóa để đạt hiệu suất cao trong việc truy xuất và xử lý dữ liệu.
2. **Tính sẵn sàng cao - Nhân rộng:** MongoDB hỗ trợ tính năng nhân rộng tự động, giúp cân bằng tải và tăng khả năng chịu tải của hệ thống một cách linh hoạt.
3. **Khả năng mở rộng cao - Sharding:** MongoDB cho phép phân chia dữ liệu thành các phân đoạn nhỏ hơn (shards), giúp phân tải dữ liệu và cải thiện hiệu suất của hệ thống.
4. **Năng động:** MongoDB không yêu cầu một lược đồ cứng nhắc, cho phép thêm, xóa trường dữ liệu một cách linh hoạt mà không ảnh hưởng đến ứng dụng.
5. **Linh hoạt:** MongoDB cho phép dữ liệu được biểu diễn dưới dạng JSON hoặc BSON, đồng thời hỗ trợ các truy vấn phong phú và phức tạp.
6. **Dữ liệu không đồng nhất:** MongoDB cho phép lưu trữ dữ liệu không đồng nhất, tức là các bản ghi trong cùng một bảng không cần phải có cấu trúc giống nhau.
7. **Không joins:** MongoDB hạn chế việc sử dụng joins trong truy vấn, thay vào đó, nó sử dụng mô hình nhúng dữ liệu để tối ưu hóa hiệu suất.
8. **Phân phối được:** MongoDB có thể được triển khai trên nhiều nền tảng và môi trường, từ máy tính cá nhân đến các môi trường điện toán đám mây.

Nhờ vào những ưu điểm này, MongoDB trở thành một lựa chọn phổ biến cho các ứng dụng cần xử lý dữ liệu lớn và có tính linh hoạt cao.

### MySQL

A logo with a dolphin

Description automatically generated

**Tổng quan về MySQL :**

MySQL là một hệ thống quản lý cơ sở dữ liệu (RDBMS) mã nguồn mở rộng, được sử dụng rộng rãi trong nhiều ứng dụng web và doanh nghiệp. Đây là một trong những cơ sở quản trị cơ sở dữ liệu phổ biến nhất, đặc biệt là trong các web ứng dụng.

**Ưu điểm:**

* **Hiệu suất cao:** Tốc độ MySQL được thiết kế để xử lý các truy vấn nhanh và hiệu quả, tối ưu hóa cho hiệu suất yêu cầu cao của ứng dụng.
* **Khả năng mở rộng:** MySQL có thể mở rộng để đáp ứng nhu cầu của hệ thống với hàng triệu giao dịch mỗi ngày.
* **Bảo mật tốt:** Cung cấp cơ chế xác thực mạnh mẽ và quản lý quyền truy cập.
* Đa nền tảng: MySQL hoạt động trên nhiều hệ điều hành như Windows, Linux, macOS và Unix.
* **Quản lý giao diện:** Các công cụ như MySQL Workbench và phpMyAdmin giúp người dùng quản lý cơ sở dữ liệu một cách dễ dàng.

**Nhược điểm:**

* **Thiếu tính năng cao cấp:** Vì vậy , đối với các hệ thống quản trị cơ sở dữ liệu thương mại như Cơ sở dữ liệu Oracle hoặc Microsoft SQL Server, MySQL có thể thiếu một số tính năng cao cấp như quản lý người dùng nâng cao, tối ưu hóa tự động, và vùng chứa dữ liệu phức tạp.
* **Hiệu suất giảm:** Khi cơ sở phát triển dữ liệu phát triển về quy mô, MySQL có thể gặp khó khăn trong việc duy trì hiệu suất cao.
* **ACID không hoàn chỉnh:** Mặc dù MySQL hỗ trợ các tính năng ACID (Tính nguyên tử, Tính nhất quán, Cách ly, Độ bền) nhưng không có giao diện như các cấp cao cấp dữ liệu cơ sở quản trị khác.

### Spring BootĐể học Spring Boot cần bắt đầu từ đâu?

**Tổng quan về Spring boot:**

Spring Boot là một mã nguồn mở dựa trên nền tảng Spring. Spring Boot trợ giúp đơn giản hóa quá trình phát triển ứng dụng Java bằng cách cung cấp một cấu hình khởi động (bản soạn sẵn) và một tập hợp cấu hình mặc định thông tin minh. Mục tiêu chính của Spring Boot là giảm cấu hình cần thiết để xây dựng các ứng dụng Spring, từ đó giúp các nhà phát triển nhanh chóng tạo ra các ứng dụng web và dịch vụ vi mô.

**Ưu điểm:**

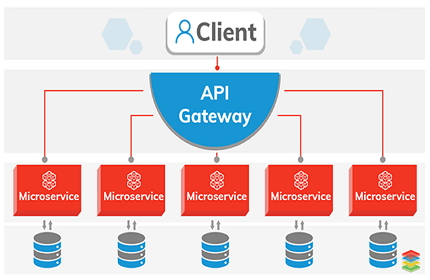
* **Cấu hình tự động (Auto-Configuration):** Spring Boot tự động cấu hình các thành phần của ứng dụng dựa trên các thư viện hiện có trong đường dẫn lớp.
* **Khả năng tích hợp dễ dàng:** Spring Boot hỗ trợ tích hợp nhiều công nghệ và framework khác nhau như Hibernate, JPA, JDBC, Thymeleaf, v.v.
* **Máy chủ nhúng:** Spring Boot đi kèm với các máy chủ nhúng như Tomcat, Jetty, hoặc Undertow, giúp đơn giản hóa quá trình phát triển khai báo.
* **Khả năng mở rộng:** Dễ dàng tích hợp các công nghệ và framework khác như Hibernate, JPA, JDBC, v.v.
* **Cộng đồng và hỗ trợ tốt:** Một sự hỗ trợ cộng đồng lớn và phong phú về tài liệu trong quá trình phát triển.

**Nhược điểm:**

* **Cấu hình tự động có thể khó kiểm soát:** Quá nhiều cấu hình tự động có thể làm việc kiểm soát ứng dụng trở nên khó khăn, đặc biệt khi xảy ra lỗi hoặc cần tùy chỉnh chi tiết.
* **Dung lượng ứng dụng lớn:** Ứng dụng Spring Boot thường có dung lượng lớn hơn khả năng tích hợp sẵn nhiều thành phần và thư viện.
* **Thời gian khởi động lâu:** Do nhiều cấu hình tự động và khởi động nhiều thành phần, thời gian ứng dụng khởi động có thể lâu hơn nên các framework sẽ nhẹ hơn.
* **Tài nguyên sử dụng nhiều hơn:** Spring Boot yêu cầu nhiều hệ thống tài nguyên hơn, có thể không phù hợp với các ứng dụng nhỏ hoặc hệ thống tài nguyên giới hạn.
* **Tăng độ phức tạp khi mở rộng:** Khi ứng dụng phát triển lớn hơn, công việc quản lý cấu hình và phụ thuộc có thể trở nên phức tạp và khó khăn.
* **Phụ thuộc vào hệ sinh thái Spring:** Ứng dụng Spring Boot phụ thuộc nhiều vào hệ sinh thái Spring, có thể làm việc chuyển đổi các công nghệ khác trở nên khó khăn.

## Kiến Trúc Phần Mềm

Chúng tôi sử dụng kiến trúc Microservice trong ứng dụng IUH Learn. Microservice là một kỹ thuật phát triển phần mềm, một biến thể của kiến trúc hướng dịch vụ (SOA), trong đó cấu trúc một ứng dụng như một tập hợp các dịch vụ được ghép với nhau. Trong kiến trúc micro-service, các dịch vụ được xử lý tốt và các giao thức rất nhẹ. Đó cũng là lý do chúng tôi đề xuất cho kiến trúc của công cụ IUH Learn ở theo mô hình Microservice.



Trong kiến trúc này, chúng tôi đã tách service chung thành 4 service nhỏ hơn bao gồm AuthService, SignUpModuleService, StudyService và ManageLearningService.

A diagram of a software company

Description automatically generated

### Auth Service

Dùng để xử lý những chức năng về đăng nhập, xác thực người dùng qua Token, xác thực thời gian sống của Token (khi Token hết hạn sẽ chuyển ra trang đăng nhập)

AuthService sẽ dùng chung một Database MySQL với ManageLearningService để truy xuất đến thông tin của sinh viên khi công việc đăng nhập hoặc xác thực diễn ra

Sử dụng MySQL rất phù hợp để quản lý những thông tin về nghiệp vụ học phần. Vì chúng cần phải có mối quan hệ chặt chẽ và ràng buộc lẫn nhau, giúp tránh tình trạng mất dữ liệu của một thực thể mà các thực thể khác có mối quan hệ với thực thể đó

### Manage Learning Service

Manage Learning Service được dùng để quản lý các chức năng dành cho admin( Quản Trị Viên ) như: quản lý Khoa, quản lý Chuyên Ngành, quản lý Môn Học, quản lý Sinh Viên, quản lý Giáo Viên, ...

Trong Cơ Sở dữ liệu của Manage Learning Service sẽ quản lý **một phần dữ liệu của Học Phần** như: mã Học Phần, mã Môn Học, mã Học Kỳ, mã Sinh Viên. Giúp liên kết đến khác thực thể cần thiết, nhằm khai thác dữ liệu theo một cách tối ưu khi sử dụng thư viện Spring Data JPA của Spring Boot

### Study Service

Study Service dùng để xử lý các nghiệp vụ liên quan đến **quá trình học tập** của sinh viên. Ngoài ra Study Service sẽ giữ một phần **dữ liệu còn lại của học phần** như:

* Tiết Lý Thuyết và Tiết Thực Hành: sẽ chứa các thông tin như thời gian( ngày, giờ ), thông tin về lớp, thông tin về giáo viên và sỉ số sinh viên
* Quản lý tính bắt buộc của học phần: để biết được đây là học phần bắt buộc hay học phần tự chọn
* Quản lý tính tiên Quyết: để biết đây có phải là học phần tiên quyết, là nền tảng cho các môn học khác hay không
* Quản lý các môn học trước/song hành: bao gồm mã Môn và tên Môn. Nhằm thỏa mãn điều kiện khi sinh viên đăng ký học phần mà học phần đó cần phải hoàn thành môn học khác, thì phải đảm bảo là sinh viên đã học qua môn đó (dữ liệu được lấy từ collection **thông tin học tập** cùng cơ sở dữ liệu) thì mới được đăng ký
* Quản Lý Trạng Thái: để biết được trạng thái của học phần (nếu như trạng thái là Mở Lớp thì mới được đăng ký)

Do các dữ liệu do Study Service nắm giữ khá linh hoạt và đa dạng (tiết lý thuyết, tiết thực hành có thể có nhiều tiết trong một học phần), nên chúng tôi lựa chọn cơ sở dữ liệu MongoDB để lưu trữ phần dữ liệu còn lại của Học Phần

### SignUp Module Service

SignUp Module Service được sử dụng trong nghiệp vụ đăng ký học phần của sinh viên bao gồm các chức năng như: xem học phần, xem học phần đã đăng ký, đăng ký học phần ( đã chọn đầy đủ các thông tin như lớp, tiết lý thuyết, tiết thực hành )

Tất Cả những thông tin đăng ký được sinh viên đăng ký, sẽ được lưu trữ bằng cơ sở dữ liệu MongoDB. Giúp tăng tốc độ truy xuất và tránh rủi ro mất dữ liệu

### Tại Sao Chúng Tôi Thiết Kế Theo Mô Hình Này?

Đối với thực trạng hiện nay, theo các trường đại học nói chung và trường đại học Công Nghiệp nói riêng, chúng ta thường gặp phải tình trạng lag, khó đăng ký khi giai đoạn đăng ký học phần xảy ra. Một phần do số lượng đăng ký tại một thời điểm quá nhiều, phần còn lại có thể là tất cả những dịch vụ của trường được tích hợp hoạt động chung một service nên mới gây ra tình trạng tắt nghẽn trên.

Để khắc phục hiện trạng đó, chúng tôi đã chia từ một service thành nhiều service nhỏ hơn, đảm nhận các nghiệp vụ, chức năng riêng biệt, và hoạt động song song với nhau. Giúp giảm thiểu tình trạng quá tải khi sinh viên đăng ký học phần, và sinh viên vẫn có thể truy cập và hệ thống để xem điểm hay xem lịch học. Ngoài ra khi giai đoạn đăng ký học phần kết thúc, chúng ta có thể ngưng hoạt động service đó để tiết kiệm chi phí.

# : PHÂN TÍCH THIẾT KẾ HỆ THỐNG

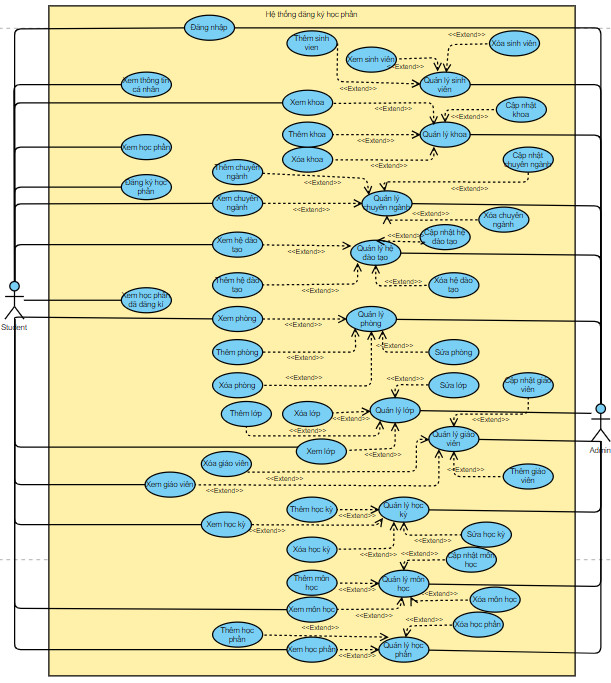
## Phân Tích Hệ Thống

Đối tượng người dùng cho hệ thống IUH Learn gồm có sinh viên và quản trị viên của trường. Trong đó, IUH Learn tập trung vào các chức năng chính của hệ thống mà hiện nay nhà trường đang gặp phải vấn đề khó khăn như đăng ký học phần. Ngoài ra, các chức năng của quản trị viên cũng được đưa vào nhằm hỗ trợ cho hệ thống đồng bộ các nghiệp vụ đào tạo và các thao tác của sinh viên.

|  |  |  |
| --- | --- | --- |
| **Đối Tượng** | **Tên Chức Năng** | **Mô Tả** |
| **Sinh viên** | Đăng nhập tài khoản | Đăng nhập vào hệ thống với mã số sinh viên được tạo |
| Xem thông tin cá nhân | Xem thông tin cá nhân của sinh viên đã đăng nhập vào ứng dụng |
| Xem học phần | Xem các học phần có trong học kỳ đã chọn |
| Đăng ký học phần | Sinh viên ghi danh của mình vào học phần đã chọn |
| Xem học phần đã đăng ký | Xem lại những học phần mà sinh viên đã đăng ký trước đó |
| **Quản trị viên** | Quản lý sinh viên | Cho phép thêm, xóa thông tin của sinh viên, xem danh sách sinh viên |
| Quản lý khoa | Cho phép thêm, xóa, sửa thông tin của khoa, xem danh sách khoa |
| Quản lý chuyên ngành | Cho phép thêm, xóa, sửa thông tin của chuyên ngành, xem danh sách chuyên ngành |
| Quản lý hệ đào tạo | Cho phép thêm, xóa, sửa thông tin của hệ đào tạo, xem danh sách hệ đào tạo |
| Quản lý phòng | Cho phép thêm, xóa, sửa thông tin của phòng, xem danh sách phòng |
| Quản lý lớp | Cho phép thêm, xóa, sửa thông tin của lớp, xem danh sách lớp |
| Quản lý giáo viên | Cho phép thêm, xóa, sửa thông tin của giáo viên, xem danh sách giáo viên |
| Quản lý môn học | Cho phép thêm, xóa, sửa thông tin của môn học, xem danh sách môn học |
| Quản lý học kỳ | Cho phép thêm, xóa, sửa thông tin của học kỳ, xem danh sách học kỳ |
| Quản lý học phần | Cho phép thêm, xóa thông tin của học phần, xem danh sách học phần |

Trong mô hình use-case của chúng tôi, có hai actor chính: sinh viên và nhân viên quản trị trường học. Use-case chính và phức tạp về nghiệp vụ, yêu cầu nhiều xử lý cho hệ thống bao gồm: Đăng Ký Học Phần

#### Mô Hình UseCase IUH Learn



#### UseCase Đăng Ký Học Phần

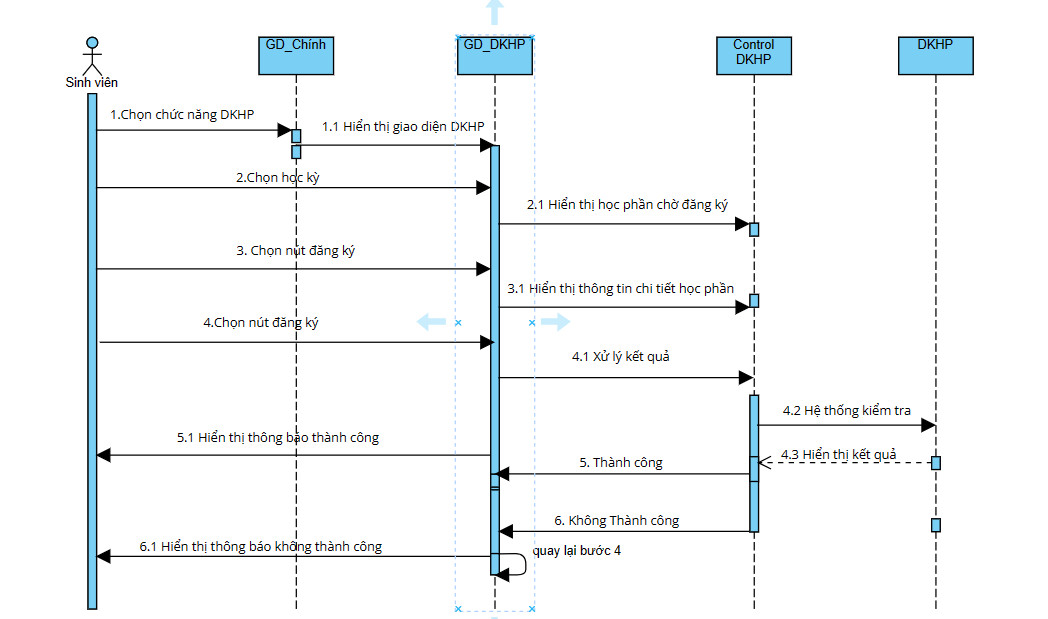
|  |  |
| --- | --- |
| * **Tên use case**: UC01\_Đăng ký học phần | |
| * **Mô tả sơ lược**: mô tả quá trình sinh viên đăng ký các học phần cho kỳ học. | |
| * **Actor chính**: Sinh viên | |
| * **Actor phụ**: | |
| * **Tiền điều kiện (Pre-condition):** Sinh viên đã đăng nhập vào hệ thống, hệ thống đã cập nhật các học phần mở đăng ký cho kì mới. | |
| * **Hậu điều kiện (Post-condition):** Sinh viên đã đăng ký thành công các học phần, hệ thống cập nhật số lượng sinh viên đăng ký cho từng học phần. | |
| * **Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Chọn chức năng đăng ký học phần từ giao diện chính | 1. Hệ thống hiển màn hình đăng ký học phần |
| 1. Chọn học kỳ đăng ký học phần | 1. Hệ thống hiển thị học phần chờ đăng ký. |
| 1. Chọn nút đăng ký học phần | 1. Hệ thống hiển thị học phần chờ đăng ký |
| 1. Chọn lớp học phần đăng ký | 1. Hệ thống kiểm tra điều kiện đăng ký (số tín chỉ, điều kiện tiên quyết, xung đột thời gian). |
| 1. Chọn nút đăng ký | 1. Hệ thống lưu thông tin đăng ký của sinh viên vào cơ sở dữ liệu và cập nhật số lượng sinh viên đăng ký. |
|  | 1. Hệ thống hiển thị thông báo đăng ký thành công và cập nhật thông tin đăng ký của sinh viên. |
| * **Luồng sự kiện thay thế (alternate flow):** | |
|  | 8.1. Hệ thống phát hiện sinh viên không đủ điều kiện để đăng ký một hoặc nhiều học phần (do chưa đạt học phần tiên quyết, xung đột thời gian, hoặc vượt quá số tín chỉ). |
|  | 8.2. Hệ thống quay lại bước 4 |

**Activity diagram:**

A screenshot of a computer

Description automatically generated

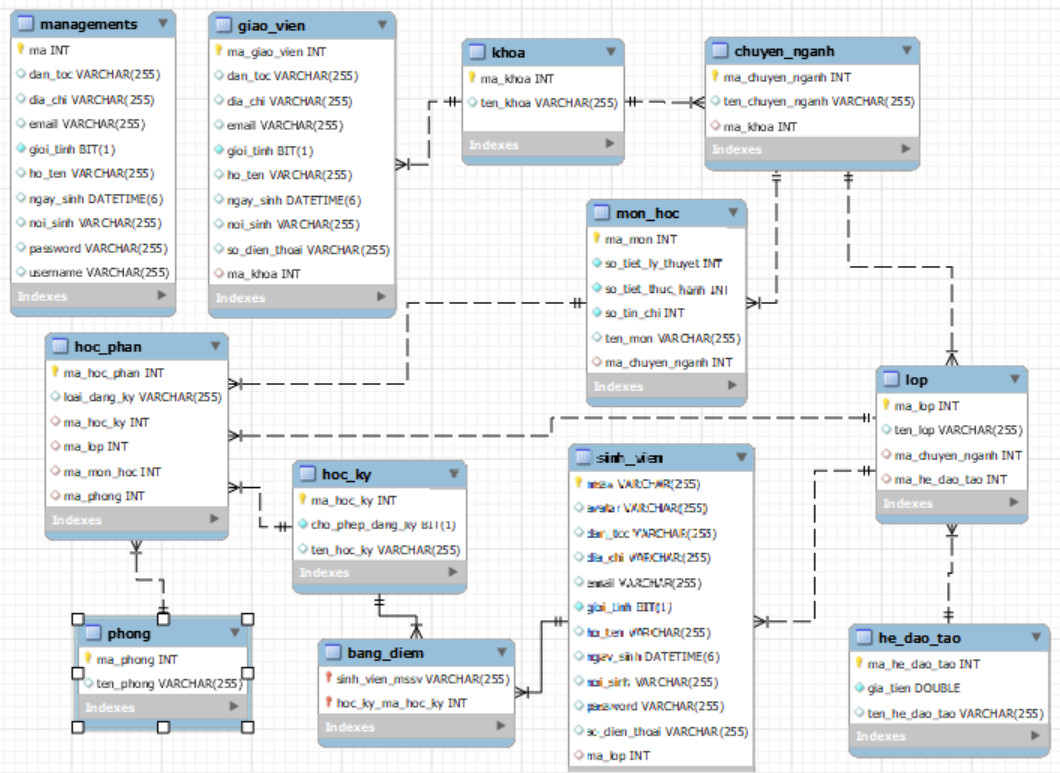
**Sequence diagram:**



Với danh sách các chức năng trên, chúng tôi xây dựng mô hình dữ liệu thể hiện toàn bộ cấu trúc cơ sở dữ liệu của IUH Learn. Trong đó, chúng tôi đặc biệt quan tâm đến quan hệ sinh\_vien, do số lượng thuộc tính nhiều và phức tạp, đồng thời có số lượng mối kết hợp nhiều nhất. Việc xác định chi tiết các thuộc tính của sinh\_vien giúp đặc tả sinh viên trong thế giới thực một cách rõ ràng và tường minh.

Bên cạnh đó, quan hệ hoc\_phan cũng là quan hệ cần được quan tâm. Đây là quan hệ mà cả sinh viên và nhân viên quản trị đều thường xuyên truy cập và thao tác. Việc xác định chính xác các thuộc tính của hoc\_phan sẽ góp phần làm cho các nghiệp vụ xử lý trở nên dễ dàng và hiệu quả hơn.

## Cơ sở dữ liệu MySQL (Quản lý các thực thể)



## Cơ sở dữ liệu của MongoDB (Quản lý các thông tin thêm của học phần)

**A screenshot of a computer code

Description automatically generated**

## Cơ sở dữ liệu của MongoDB (Quản lý đăng ký học phần)

**A screenshot of a computer

Description automatically generated**

# : PHÂN TÍCH GIAO DIỆN ĐỒ HỌA

## Giao Diện Đăng Nhập

A yellow bus in front of a building

Description automatically generated

## Giao Diện Quản Lý Sinh Viên

A screenshot of a computer

Description automatically generated

## Giao Diện Quản Lý Khoa

A screenshot of a chat

Description automatically generated

## Giao Diện Quản Lý Chuyên Ngành

A screenshot of a computer

Description automatically generated

## Giao Diện Quản Lý Hệ Đào Tạo

A screenshot of a computer

Description automatically generated

## Giao Diện Quản Lý Phòng

A screenshot of a computer

Description automatically generated

## Giao Diện Quản Lý Lớp

A screenshot of a computer

Description automatically generated

## Giao Diện Quản Lý Giáo Viên

A screenshot of a computer

Description automatically generated

## Giao Diện Quản Lý Môn Học

A screenshot of a computer

Description automatically generated

## Giao Diện Quản Lý Học Kỳ

A white background with black text

Description automatically generated

## Giao Diện Quản Lý Học Phần

A screenshot of a computer

Description automatically generated

## Giao Diện Đăng Nhập Mã Sinh Viên

A yellow bus in front of a building

Description automatically generated

## Giao Diện Thông Tin Chung

A white background with black text

Description automatically generated

## Giao Diện Đăng Ký Học Phần

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

# : MỤC TIÊU VÀ HƯỚNG PHÁT TRIỂN

## Mục Tiêu và Giải Pháp

**Mục tiêu:** Xây dựng trang web IUHLearn để hỗ trợ sinh viên đang ký học phần trong quá trình nghiên cứu và theo học tại trường.

**Giải pháp:** Phát triển một trang web giúp sinh viên theo dõi qua trình đang theo học tại trường, đăng ký học phần, xem điểm… một cách dễ dàng và nhanh chóng.

## Kiểm Chứng và Hạn Chế

**Kiểm chứng:** Đã kiểm chứng tính ổn định của hệ thống khi triển khai tại chỗ.

**Hạn chế:**

* Chỉ thử nghiệm với số lượng truy xuất vừa và nhỏ.
* Chưa thử nghiệm hệ thống với lượng truy cập lớn (tới 100.000 request).
* Chức năng trên hệ thống web dành cho quản trị viên còn hạn chế.

## Kế Hoạch Tương Lai

Khắc phục hạn chế: Dành tài nguyên và nhân lực để giải quyết các hạn chế hiện tại.

Mở rộng và hoàn thiện: Thực hiện thêm các thử nghiệm với lượng truy cập lớn và hoàn thiện các chức năng trên hệ thống.