

**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP.HCM**

**KHOA: CÔNG NGHỆ THÔNG TIN**

Đoàn Thế Long - 19475471

Lê Hoàng Minh - xxxxxxxx

**PHẦM MỀM HỖ TRỢ HỌC VỤ CHO SINH VIÊN TẠI TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP THÀNH PHỐ HỒ CHÍ MINH**

**Ngành: Kỹ Thuật Phần Mềm**

**Giảng viên Hướng dẫn: TS. Võ Văn Hải**

**TP.HỒ CHÍ MINH, THÁNG 5 NĂM 2024**

# LỜI CẢM ƠN

Để hoàn thành ứng dụng phần mềm này, chúng tôi xin tỏ lòng biết ơn sâu sắc đến thầy Võ Văn Hải đã tận tình hướng dẫn trong suốt quá trình làm đồ án tốt nghiệp và viết báo cáo.

Chúng tôi chân thành cảm ơn quý thầy, cô trong khoa công nghệ thông tin, Trường Đại Học Công Nghiệp thành phố Hồ Chí Minh đã tận tình truyền đạt kiến thức trong những năm chúng tôi học tập. Với vốn kiến thức được tiếp thu trong quá trình học không chỉ là nền tảng cho quá trình nghiên cứu khóa luận mà còn là hành trang quý báu để chúng tôi bước vào đời một cách vững chắc và tự tin.

Cuối cùng chúng tôi kính chúc quý thầy, cô dồi dào sức khỏe và thành công trong sự nghiệp cao quý. Đồng kính chúc các cô, chú, anh, chị trong khoa, ngành công nghệ thông tin luôn dồi dào sức khỏe, đạt được nhiều thành công tốt đẹp trong công việc.

# *Tháng 5 – 2024*

*Đoàn Thế Long – Lê Hoàng Minh*

# NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

|  |
| --- |
| *Ngày ....... tháng ........ năm .........*  **GIẢNG VIÊN HƯỚNG DẪN**  *(Ký và ghi rõ họ tên)* |

# NHẬT KÝ LÀM VIỆC

|  |  |  |  |
| --- | --- | --- | --- |
| **Tuần** | **Từ ngày** | **Đến ngày** | **Nội dung** |
| 1 | 13/04/2024 | 14/04/2024 | - Nhận đề tài thầy giao.  - Họp google meet chọn công nghệ sử dụng.  - Phân chia công việc:  + Long: Code phí backend resful springboot api.  + Minh: Code phía frontend bằng html5, javascript, css,.., Thiết kế UI |
| 2 | 15/04/2024 | 21/04/2024 | Long: Design hoàn tất các entities và ánh xạ xuống tables database + Generate data mẫu.  Minh: Vẽ figma, code xong giao diện Login và Homepage trang sinh viên (fake data tạm thời). |
| 3 | 22/04/2024 | 26/04/2024 | Long: Thiết kế các api theo yêu cầu trả về data cho frontend mong muốn. (Api chức năng login, api get student data,..)  Minh: Sửa lại UI với real data từ api trả về các chức năng: Login, Thông tin sinh viên, đăng ký học phần,.. |
| 4 | 29/04/2024 | 05/05/2024 | Long:  + Thêm các api cho user Manager.  + Bổ sung chức năng redis cache api.  + Làm tài liệu docs (vẽ usecase, sequence diagram, activy diagram,..)  Minh: Code UI cho user Manager |
| 5 | 06/05/2024 | 12/05/2024 | Long: Hoàn thiện document.  Minh: Họp meet phụ document, tập dợt báo cáo. Chuẩn bị báo cáo. |

**MỤC LỤC**

# CHƯƠNG 1: GIỚI THIỆU

## Tổng quan

Trường ĐH Công nghiệp TP.HCM (IUH) là một trong những trường đại học lớn nhất Bộ Công thương, cũng là trường có quy mô đào tạo lớn với hơn 36000 sinh viên, đa ngành, đa bậc học và nhiều chương trình liên kết quốc tế. Trường được đầu tư cơ sở vật chất tốt, đáp ứng nhu cầu của người học và các tiêu chuẩn quốc tế. Bên cạnh, đội ngũ giảng viên có trình độ cao cùng với nhiều chương trình đào tạo đạt chuẩn quốc tế như AUN. Chính vì lẻ đó, nhà trường cần có một hệ thống website hỗ trợ đào tạo một cách hiệu quả và kịp thời cho người hoc.

Ứng dụng phần mềm hỗ trợ học vụ cho sinh viên là một công cụ quan trọng nhằm nâng cao hiệu quả quản lý và học tập trong môi trường đại học. Ứng dụng này tích hợp nhiều tính năng hữu ích như quản lý thông tin sinh viên, theo dõi quá trình học tập, đăng ký môn học, tra cứu điểm số, và giao tiếp giữa sinh viên và giảng viên. Thông qua việc sử dụng công nghệ hiện đại, ứng dụng giúp đơn giản hóa các thủ tục hành chính, giảm bớt khối lượng công việc cho bộ phận học vụ, đồng thời cung cấp cho sinh viên một nền tảng tiện lợi để theo dõi và quản lý thông tin cá nhân cũng như học tập của mình. Ứng dụng còn cho phép sinh viên truy cập vào lịch học, lịch thi, và các thông báo quan trọng từ nhà trường một cách nhanh chóng và chính xác. Điều này không chỉ giúp sinh viên dễ dàng lập kế hoạch học tập và chuẩn bị tốt hơn cho các kỳ thi mà còn tạo điều kiện thuận lợi cho việc trao đổi thông tin và phản hồi từ cả hai phía – sinh viên và nhà trường. Ngoài ra, ứng dụng cũng hỗ trợ chức năng nhắc nhở các sự kiện quan trọng, hạn nộp bài tập, và cập nhật thông tin học bổng, giúp sinh viên không bỏ lỡ những cơ hội quan trọng trong quá trình học tập. Với giao diện thân thiện, dễ sử dụng và tương thích trên nhiều thiết bị, ứng dụng phần mềm hỗ trợ học vụ không chỉ mang lại sự tiện lợi mà còn góp phần tạo nên một môi trường học tập hiện đại, chuyên nghiệp và hiệu quả. Việc triển khai ứng dụng này không chỉ đáp ứng nhu cầu ngày càng cao của sinh viên và giảng viên mà còn là bước tiến quan trọng trong quá trình số hóa và hiện đại hóa quản lý giáo dục tại các trường đại học.

## Mục tiêu đề tài

Phát triển một ứng dụng phần mềm hỗ trợ học vụ nhằm tối ưu hóa quy trình quản lý học tập và hành chính tại các trường đại học. Ứng dụng hướng đến việc cung cấp cho sinh viên một công cụ tiện ích để theo dõi quá trình học tập, tra cứu thông tin cá nhân, và giao tiếp hiệu quả với giảng viên cũng như bộ phận quản lý học vụ. Bên cạnh đó, đề tài còn đặt mục tiêu giảm tải khối lượng công việc hành chính cho nhà trường, tăng cường tính minh bạch và hiệu quả trong việc quản lý dữ liệu sinh viên, từ đó nâng cao chất lượng dịch vụ và trải nghiệm học tập. Đề tài cũng kỳ vọng sẽ đóng góp vào việc hiện đại hóa và số hóa môi trường giáo dục, đáp ứng nhu cầu ngày càng cao về công nghệ trong học tập và quản lý giáo dục.

## Phạm vi đề tài

.Phạm vi của đề tài bao gồm việc nghiên cứu, phát triển và triển khai ứng dụng phần mềm hỗ trợ học vụ dành cho sinh viên tại các trường đại học. Đề tài tập trung vào các tính năng chính như quản lý thông tin cá nhân, theo dõi quá trình học tập, đăng ký môn học, tra cứu điểm số, và giao tiếp giữa sinh viên và giảng viên. Ứng dụng sẽ được thiết kế để hoạt động trên nhiều nền tảng, bao gồm cả máy tính và thiết bị di động, nhằm đảm bảo tính tiện dụng và linh hoạt cho người dùng. Trong quá trình thực hiện, đề tài sẽ tiến hành khảo sát nhu cầu và phản hồi từ sinh viên, giảng viên, và bộ phận học vụ để đảm bảo rằng ứng dụng đáp ứng được các yêu cầu thực tế và cải thiện trải nghiệm người dùng. Ngoài ra, phạm vi đề tài cũng bao gồm việc thử nghiệm và đánh giá ứng dụng trong môi trường thực tế tại trường đại học, từ đó tinh chỉnh và hoàn thiện sản phẩm trước khi triển khai chính thức. Đề tài không chỉ giới hạn trong việc phát triển phần mềm mà còn chú trọng đến việc xây dựng tài liệu hướng dẫn sử dụng, tổ chức các buổi đào tạo cho người dùng, và thiết lập hệ thống hỗ trợ kỹ thuật sau khi triển khai. Mục tiêu cuối cùng là tạo ra một giải pháp toàn diện, hiệu quả và bền vững, góp phần nâng cao chất lượng quản lý và học tập trong môi trường giáo dục đại học.

## Mô tả yêu cầu chức năng

Đại học công nghiệp Hồ Chí Minh là đại học có số lượng sinh viên khá lớn với hơn 30000 sinh viên. Hệ thống trường hiện tại đang sử dụng phục vụ cho các công việc đào tạo như thêm học phần, lớp học phần, quản lý sinh viên trong các kỳ tuyển sinh, đồng thời cho sinh viên sử dụng trong mục đích học tập, theo dõi kết quả học tập, tình trạng học tập hiện tại.

Mục đích học tập cho sinh viên như đăng ký học phần, xem kết quả học tập, theo dõi lịch học hằng tuần, cập nhật lịch học cho sinh viên để nắm tình hình học tập theo tiến độ.

Phân chia hệ thống dành cho **quản lý, sinh viên.**

Các chức năng của sinh viên như

* **Thông tin cá nhân**: có thể xem, đổi mật khẩu

Sinh viên có thể xem tất cả thông số của mình nhưng không được quyền chỉnh sửa.

Đổi mật khẩu yêu cầu sinh viên phải cung cấp mật khẩu cũ, mật khẩu phải thoả ràng buộc lượng ký tự là tám, chỉ gồm số và chữ cái. Mật khẩu lưu xuống cơ sở dữ liệu phải được mã hoá theo một cơ chết nhất định.

* **Xem lịch học theo tuần**

Sinh viên có thể xem lịch học theo tuần dựa vào ngày hiện tại. Hệ thống sẽ gửi lịch học của bảy ngày trong tuần có chứa ngày đó. Lịch học cần thể hiện rõ giáo viên, phòng học, môn học, lớp lý thuyết hay thực hành, thể hiện lịch thi, lịch học.

* **Xem kết quả học tập**

Sinh viên xem kết quả học tập từ khi vào trường đến thời điểm hiện tại, kết quả học tập cần thể hiện rõ chi tiết bảng điểm, các cột thường kỳ, giữa kỳ,…. Thể hiện rõ điểm chữ và số. Kết quả xếp loại của môn học đó.

* **Xem tiến độ học tập**

Có thể xem được tổng số tín chỉ đã đạt dựa trên tổng số tín chỉ của khoa/chuyên ngành. Xem được chỉ số điểm trung bình của bản thân so với lớp học phần.

* **Quản lý học phần:**
* **Xem danh sách học kỳ**: sinh viên xem danh sách học kỳ từ khoá học sinh viên bắt đầu học đến thời điểm hiện tại. Ví dụ sinh viên nhập học năm 2017 thì sẽ thấy được học kỳ từ 2017 đến nay, nếu sinh viên nhập học từ 2018 thì sẽ thấy được học kỳ từ 2018 đến nay.
* **Xem danh sách học phần có trong học kỳ**: cho thấy được học phần của chuyên ngành sinh viên đang theo học của học kỳ hiện tại hoặc học kỳ sinh viên chọn.
* **Xem danh sách lớp học phần có trong học kỳ**: cho thấy được lớp học phần của chuyên ngành sinh viên đang theo học mà sinh viên chưa qua môn, chưa đăng ký, hoặc có mở ở học kỳ này.
* **Xem thông tin chi tiết lớp học**: xem được thông tin giáo viên, phòng, mã môn học của lớp học phần.
* **Kiểm tra trùng lịch học**: kiểm tra được lớp học phần sinh viên đang chọn có trùng với lịch học hiện nay hay không.
* **Xem danh sách lớp học phần đã đăng ký trong học kỳ**: sau khi đăng ký học phần thì sẽ xem lại danh lớp đã đăng ký, bao gồm thời khoá biểu và các thông tin của lớp học phần.
* **Đăng ký học phần**: đăng ký học phần mà sinh viên đang chọn, cần kiểm tra đã trùng hoặc đã đăng ký trước đó hay chưa.
* **Huỷ học phần đã đăng ký**: huỷ học phần sinh viên đã đăng ký.
* **Xem lịch học của lớp học phần**: xem chi tiết lịch học, giáo viên, phòng học của lớp học phần sinh viên chọn.

Các chức năng của quản lý như:

* **Quản lý sinh viên**: cho phép thêm một sinh viên khi có sinh viên nhập học, cần điền các thông tin sinh viên như chuyên ngành, khoa,… các thông tin cá nhân khác. Đồng thời tạo mã sinh viên và mật khẩu mặc định cho sinh viên vào hệ thống là 123456. Cho phép sửa đổi thông tin sinh viên, nhưng không được sửa đổi rỗng các thông tin. Xoá sinh viên khi cần thiết như đuổi học…. Mỗi sinh viên có một mã hồ sơ, mã hồ sơ không được trùng.
* **Quản lý khoa**: hiển thị danh sách khoa của trường, thêm một khoa mới khi có yêu cầu, sửa đổi tên khoa, xoá khoa khi trường không còn đào tạo. Xoá cần phải hỏi xác nhận.
* **Quản lý chuyên ngành**: hiển thị danh sách chuyên ngành của trường, thêm một khoa mới khi có yêu cầu, sửa đổi tên chuyên ngành, số tín chỉ, xoá chuyên ngành khi trường không còn đào tạo. Sửa đổi khoa của chuyên ngành đó, khoa phải tồn tại trong cơ sở dữ liệu. Xoá cần phải hỏi xác nhận. Có chức năng lọc các chuyên ngành theo khoa.
* **Quản lý môn học**: hiển thị danh sách môn học của trường, thêm một môn học mới khi có yêu cầu của khoa, xoá môn học khi trường không còn đào tạo. Sửa đổi khoa của chuyên ngành đó, khoa phải tồn tại trong cơ sở dữ liệu. Xoá cần phải hỏi xác nhận. Có chức năng lọc các chuyên ngành theo khoa.
* **Quản lý học kỳ**: quản lý thứ tự học kỳ cũng như năm học bắt đầu và năm học kết thúc của học kỳ đó.
* **Quản lý học phần**: một học phần thuộc một chuyên ngành, có thể sửa đổi môn học của học phần, cũng như học phần tiên quyết, bắt buộc, tương đương.
* **Quản lý lớp học phần**: kiểm tra lớp học phần và thể hiện được các trạng thái học phần và số lượng sinh viên trong lớp đó như số lượng sinh viên đăng ký tối đa và hiện tại.

# CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

## Ngôn ngữ lập trình Java:

Java là một ngôn ngữ lập trình hướng đối tượng được bảo trì và tiếp tục phát triển bởi Oracal. Ngôn ngữ này được sử dụng để phát triển phần mềm ứng dụng cho nhiều mục đích khác nhau, bao gồm cả ứng dụng desktop, website, game hay ứng dụng trên các thiết bị di động. Ngoài ra, Java còn là một ngôn ngữ cho phép phát triển các ứng dụng chạy trên nhiều nền tảng khác nhau.

Hiện nay, đội ngũ phát triển ứng dụng sử dụng ngôn ngữ Java ngày càng lớn mạnh. Họ cung cấp các chương trình hữu ích giúp lập trình viên phát triển và chạy các ứng dụng một cách hiệu quả. Nền tảng Java bao gồm một công cụ thực thi, trình biên dịch và bộ thư viện. James Gosling đã phát triển nền tảng này tại Sun Microsystems, sau đó được tập đoàn Oracle mua lại và tiếp tục phát triển đến ngày nay.

## Ngôn ngữ lập trình Html, Javascript, Css:

HTML, JavaScript, và CSS là ba ngôn ngữ quan trọng trong phát triển web hiện đại. HTML5 (HyperText Markup Language) là phiên bản mới nhất của HTML, cung cấp cấu trúc và nội dung cho trang web. HTML5 mang lại nhiều tính năng mới, bao gồm các thẻ cấu trúc như <header>, <footer>, <article>, và hỗ trợ tốt hơn cho video và âm thanh với các thẻ <video> và <audio>. JavaScript là ngôn ngữ lập trình phía client, giúp tạo ra các tương tác động trên trang web. Với sự hỗ trợ của các thư viện và framework như React, Angular, và Vue.js, JavaScript giúp phát triển các ứng dụng web phức tạp và tương tác một cách dễ dàng hơn. CSS (Cascading Style Sheets) là ngôn ngữ dùng để định dạng giao diện trang web. CSS3, phiên bản mới nhất của CSS, cung cấp nhiều tính năng như flexbox, grid layout, và các hiệu ứng chuyển động, giúp thiết kế trang web trở nên linh hoạt và đẹp mắt hơn. Kết hợp lại, HTML5, JavaScript và CSS là nền tảng của các trang web hiện đại, mang đến trải nghiệm người dùng phong phú và hiệu quả. (*Hình 2.2.1*)



*Hình 2.2.1 Công dụng của bộ 3 html, css, javascript dùng để thiết kế web*

|  |
| --- |
|  |

## Công nghệ sử dụng

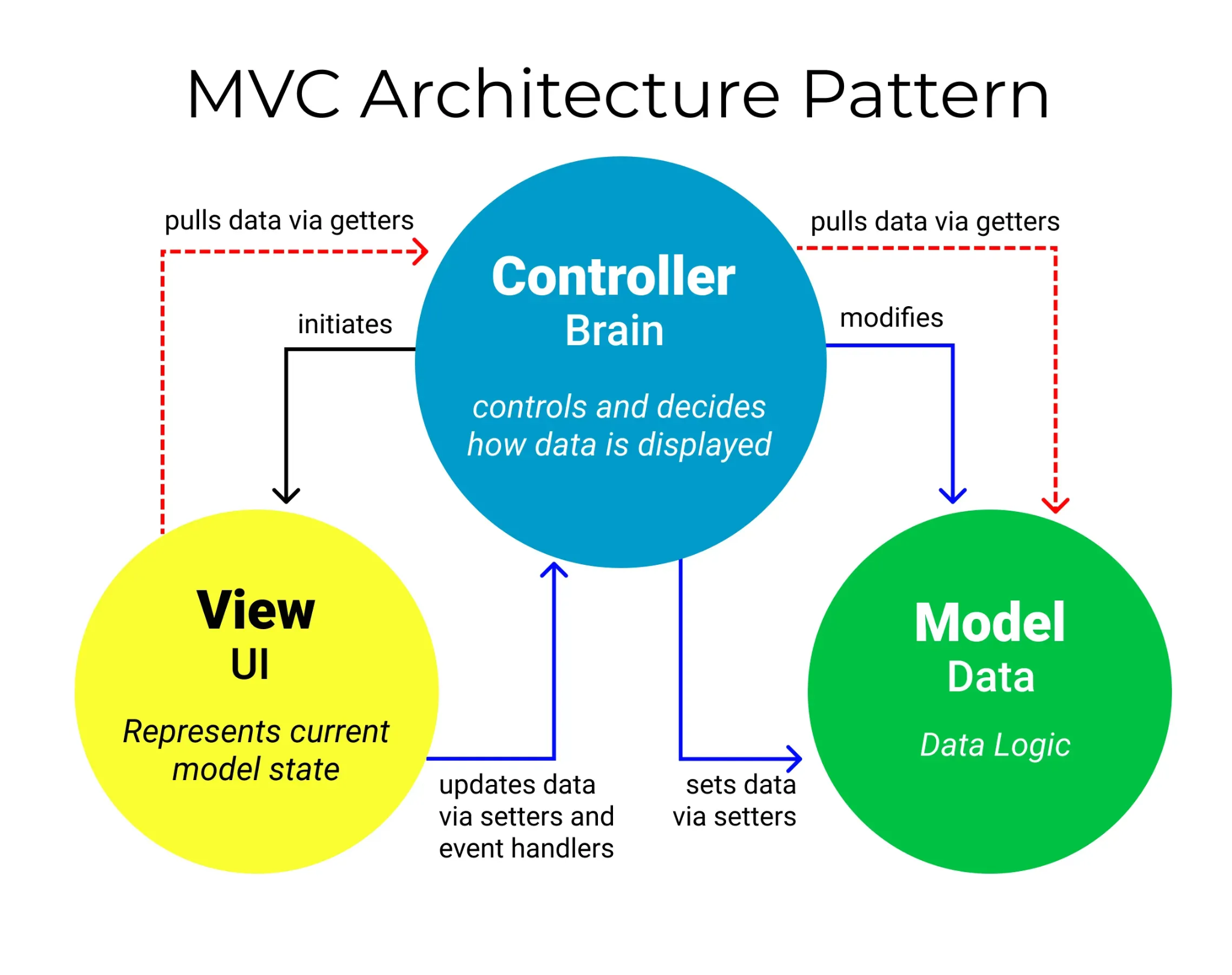
### Spring Boot Framework:

Spring Boot là khuôn mẫu (framework) phát triển dựa trên ngôn ngữ Java trong hệ sinh thái Spring framework. Nó giúp cho các lập trình viên đơn giản hóa quá trình lập trình một ứng dụng với Spring. Trong Spring Boot, người lập trình chỉ tập trung vào việc phát triển các nghiệp vụ cho ứng dụng mà không quan tâm nhiều về kiến trúc và mã nguồn của ứng dụng. Một kiến trúc webserver xây dựng dựa trên spring boot gồm các thành phần cơ bản như sau: Repository Class Extending, Service Layer, Model, Controller.

|  |
| --- |
|  |
| *Hình 2.3.1 Kiến trúc của luồng hoạt động của Spring Boot.* |

Repository Class Extending là các thư viện hoặc framework được phát triển và duy trì bởi cộng đồng các lập trình viên Java để hỗ trợ quá trình xây dựng hệ thống một cách nhanh chóng; Service Layer là lớp dịch vụ chứa các phương thức (hàm) xử lý các yêu cầu như chỉnh sửa, truy xuất, thêm, xóa dữ liệu trong cơ sở dữ liệu. Nó là nơi tập trung các xử lý nghiệp vụ của hệ thống back-end; Model là thành phần được đính kèm theo các truy vấn của Services Layer xuống cơ sở dữ liệu. Model đóng nhiều vai trò quan trọng như phát sinh hệ cơ sở dữ liệu theo liên kết các lớp của lập trình viên, chuyển đổi kết quả truy xuất từ cơ sở dữ liệu thành các thực thể lớp; Controller là thành phần giao tiếp trực tiếp với phía client, tồn tại dưới dạng các api. Model có trách nhiệm xử lý yêu cầu của client, xác minh các yêu cầu đó. Model truy cập đến phương thức phù hợp trong Service Layer để đạt được kết quả mong muốn, trả kết quả về cho phía client.

### 2.3.2. Kiến trúc MVC



Hình 2.3.2 Kiến trúc MVC sẽ được áp dụng vào ErukaLearn

Mô hình MVC (Model-View-Controller) là một kiến trúc phần mềm được sử dụng rộng rãi trong phát triển ứng dụng. Nó giúp tách biệt logic xử lý dữ liệu (Model), giao diện người dùng (View), và luồng điều khiển (Controller). Model đại diện cho dữ liệu và logic xử lý liên quan. Nó là nơi lưu trữ thông tin, thực hiện các thao tác cơ bản như truy xuất, cập nhật, xóa dữ liệu. Model không phụ thuộc vào giao diện người dùng và không trực tiếp tương tác với người dùng. View chịu trách nhiệm hiển thị thông tin cho người dùng. Nó biểu diễn dữ liệu từ Model và cung cấp giao diện để người dùng tương tác. Mỗi View có thể hiển thị dữ liệu theo cách riêng, phụ thuộc vào yêu cầu cụ thể của ứng dụng. Controller là thành phần xử lý sự kiện và điều khiển luồng thông tin giữa Model và View. Nó nhận các yêu cầu từ người dùng thông qua View, gọi phương thức tương ứng trong Model để lấy/gửi dữ liệu và cập nhật View phù hợp. Mô hình MVC tách biệt các thành phần, giúp mã nguồn dễ quản lý, bảo trì và mở rộng. Nó cũng thúc đẩy tái sử dụng code và cho phép phát triển song song giữa các thành viên trong nhóm. Bằng cách sử dụng MVC, các nhà phát triển có thể phân chia công việc một cách rõ ràng và nhanh chóng xử lý các yêu cầu của khách hàng.

### 2.3.3. Kiến trúc ResfulAPI

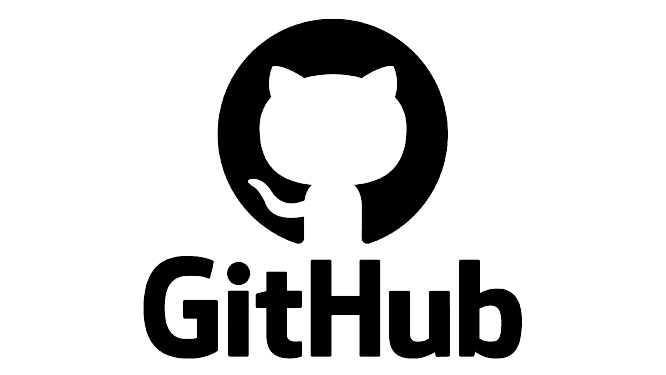


*Hình 2.3.3 Kiến trúc ResfulAPI sẽ được áp dụng vào ErukaLearn*

RESTful API sử dụng các phương thức HTTP (GET, POST, PUT, DELETE, v.v.) để truy cập và thao tác với các tài nguyên (resource) trên một server. Kiến trúc RESTful giúp cho các ứng dụng của chúng tôi có thể tương tác với nhau một cách dễ dàng và đồng nhất.

Bằng cách sử dụng Spring Boot, phía backend có thể tạo ra các API một cách nhanh chóng, dễ dàng mở rộng, và đảm bảo tính bảo mật cho các ứng dụng. Hay nói cách khác việc đọc ghi, truy vấn data, sử lý ở tầng service và tạo ra controller cho API khi làm việc với framework springboot bằng ngôn ngữ java dữ liệu hướng đối tượng và kiến trúc resful thì việc thao tác dữ liệu của dưới cơ sở dữ liệu phải nói là cực kì dễ dàng, nhanh chóng mà còn đảm bảo hiệu xuất server cực kỳ tốt.

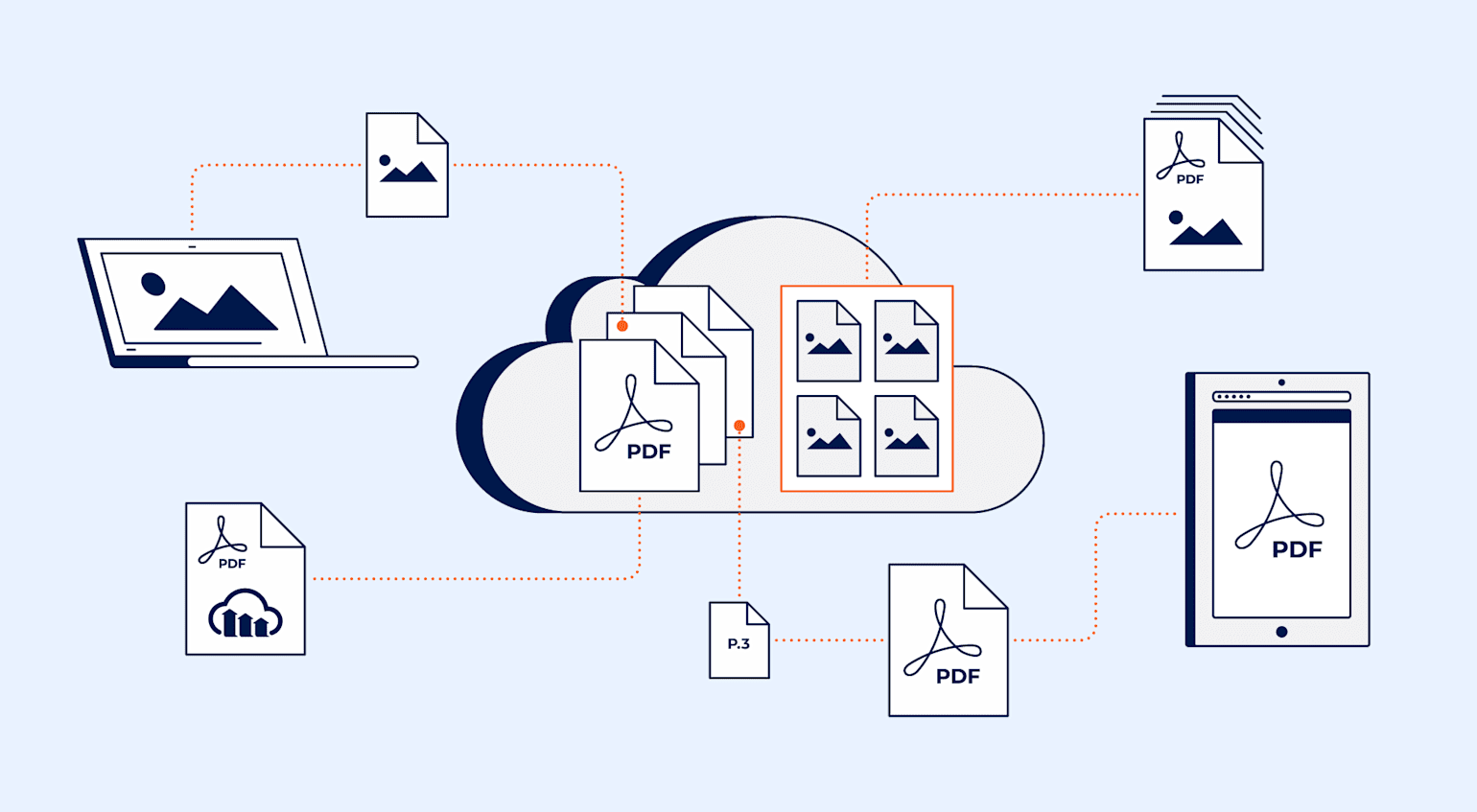
### 2.3.4. Github



*Hình 2.3.4 Đề tài quản lý source code thông qua Github*

GitHub là một nền tảng lưu trữ mã nguồn mở (open-source) trên internet cho phép các lập trình viên lưu trữ, quản lý và chia sẻ mã nguồn của họ. Nền tảng này cung cấp một giao diện dễ sử dụng để quản lý mã nguồn, bao gồm cả hệ thống kiểm tra phiên bản và quản lý sửa đổi (version control) để theo dõi lịch sử thay đổi của mã nguồn. GitHub cho phép người dùng tạo ra các kho lưu trữ (repositories) để lưu trữ các dự án của mình và chia sẻ mã nguồn này với cộng đồng. Các nhà phát triển có thể đóng góp vào dự án của người khác bằng cách gửi yêu cầu kéo (pull requests) để đưa những thay đổi của họ vào mã nguồn chính. Nền tảng này cũng cho phép lập trình viên tạo ra các vấn đề (issues) và bình luận trực tiếp trên mã nguồn để giải quyết các vấn đề liên quan đến dự án. Ngoài ra, GitHub cũng cung cấp các tính năng hỗ trợ quản lý dự án, như lên kế hoạch và theo dõi tiến độ dự án bằng các công cụ như bảng (board) và trạm (milestone). Tất cả những tính năng này giúp tạo điều kiện cho việc hợp tác giữa các lập trình viên và quản lý mã nguồn một cách dễ dàng và hiệu quả.

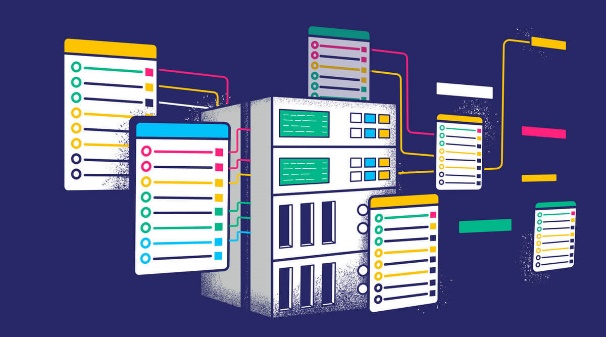
### 2.3.5. Cloudinary



*Hình 2.3.5 Cloudinary sẽ dùng lưu trữ hình ảnh avatar 3x4 của ErukaLearn*

Cloudinary là một dịch vụ quản lý hình ảnh và video trên đám mây, cho phép các nhà phát triển và nhà thiết kế quản lý, lưu trữ, chuyển đổi và phân phối các tệp đa phương tiện của mình. Cloudinary cung cấp các tính năng quản lý hình ảnh và video như tạo bản nhỏ, xử lý tập tin, cắt hình ảnh, chỉnh sửa hình ảnh, quản lý màu sắc, phân phối và quản lý định dạng tập tin. Dịch vụ này còn cho phép người dùng tùy chỉnh, xây dựng và tích hợp các tính năng quản lý đa phương tiện của mình vào các ứng dụng của họ thông qua API. Với các tính năng như quản lý tập tin đa phương tiện, tích hợp đơn giản và khả năng mở rộng linh hoạt, Cloudinary là một lựa chọn phổ biến cho các ứng dụng web, di động và thương mại điện tử. Dịch vụ này cũng cung cấp các tính năng bảo mật đáng tin cậy để đảm bảo an toàn cho dữ liệu của người dùng.

### 2.3.6 Database MySQL

*Hình 2.3.6 Database MySQL sẽ được dùng trong ELearning để ánh xạ mối quan hệ*

MySQL: cơ sở dữ liệu có quan hệ, dùng lưu trữ dữ liệu của toàn hệ thống như dữ liệu sinh viên, dữ liệu học phần, thời khóa biểu…vv (Chi tiết xem ở sơ đồ class).

Database sẽ được đặt tên là `sv\_dkhp` và thuộc loại kiểu dữ liệu `UTF-8 General CI`. Kiểu dữ liệu này phù hợp với các trường thuộc ngôn ngữ Tiếng Việt (Việt Nam) có dấu chúng ta hay dùng ngoài ra nó còn tối ưu thêm bit ghi dữ liệu hỗ trợ dài hơn cho kiểu varchar.

Database được sinh ra có thể do tạo tay sau đó import file script chứa data vào. File script chứa data mẫu sẽ được đính kèm trong source code với đường dẫn sau: `src/main/resources/static/initdb.sql`.

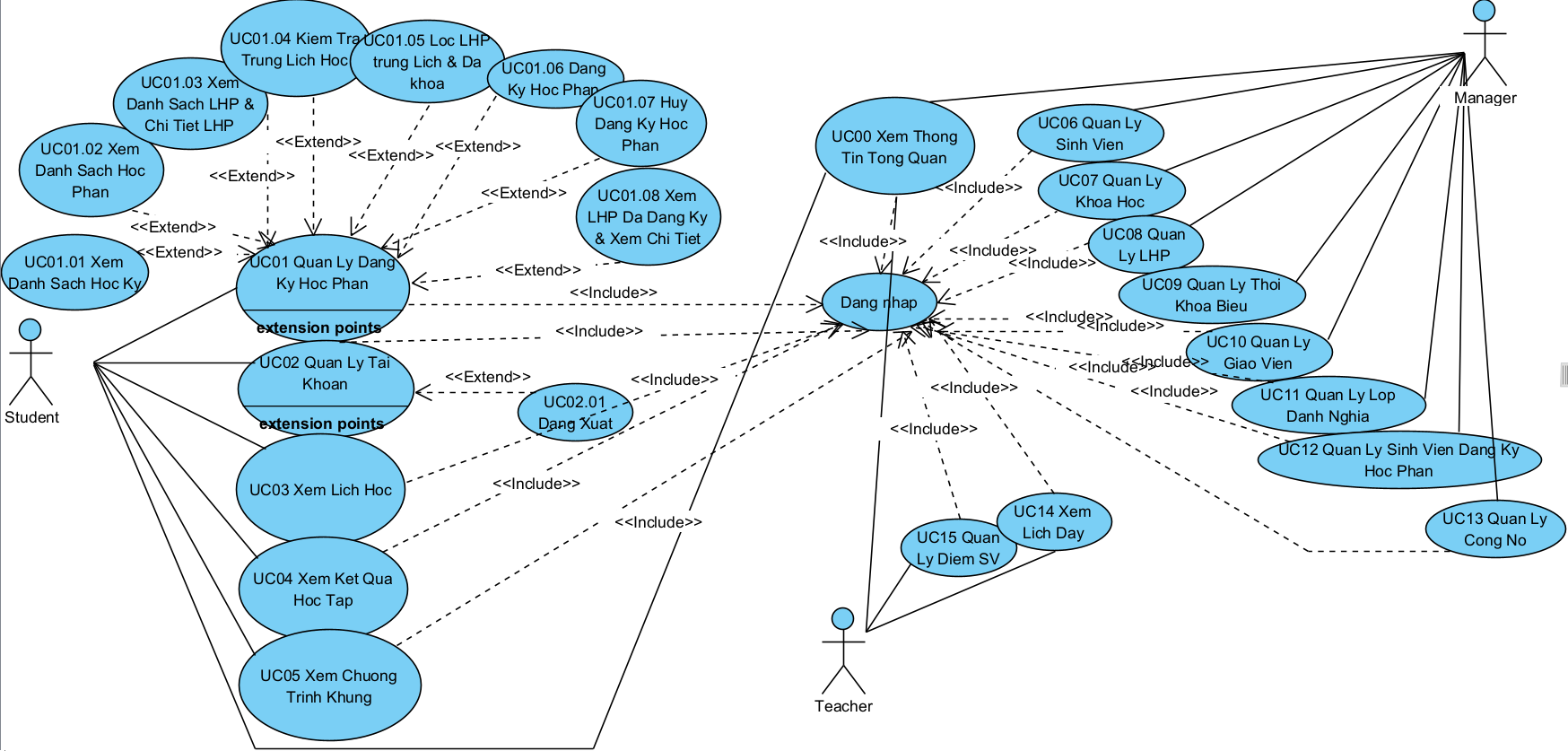
Database sẽ sử dụng database RDS của AWS. Khi khởi tạo xong RDS trên AWS ta sẽ cấu hình inbound rule port để connect Mysql Connector vào database trên RDS từ xa dưới máy tính chúng ta. Sau đó sẽ dùng query tạo database và import file seed chứa dữ liệu mẫu vào database. Các ứng dụng truy vấn xuống database cũng sẽ sử dụng dường dẫn, ip host kết nối tới database do AWS RDS cung cấp thông tin. Chúng tôi cũng sẽ cấu hình RDS của AWS sao cho có khả năng autoscaling để database luôn ổn định và mở rộng tài nguyên (dung lượng) khi gần hết.

# CHƯƠNG 3: PHÂN TÍCH

Hiện nay, hệ thống IUH cung cấp hai phân hệ chức năng chính, bao gồm phân hệ cho sinh viên và phân hệ cho quản trị viên. Hệ thống này có hai tác nhân chính, đó là sinh viên và quản trị viên. Quản trị viên có trách nhiệm quản lý thông tin liên quan đến giáo dục, bao gồm thêm mới sinh viên, quản lý thông tin môn học và chuyên ngành. Trong khi đó, sinh viên sử dụng ứng dụng di động IUH để theo dõi thông tin học tập của bản thân.

## 3.1. Sơ đồ Usecase:

Hệ thống của chương trình hỗ trợ đào tạo sinh viên gồm các actor chính như Sinh viên, quản lý. Tuy nhiên chương trình tập trung vào việc phát triển hỗ trợ cho sinh viên trên nền tảng điện thoại thông minh. Bảng use-case thể hiện rõ các chức năng hiện có trong ứng dụng của sinh viên cũng như người quản lý. Người quản lý có thể quản lý đào tạo sinh như việc sinh viên nhập học, mở lớp học phần, học phần... Còn sinh viên thực hiện các chức năng cơ bản như theo dõi lịch học, kết quả học tập, đăng ký học phần. Dưới đây là bảng vẽ lược đồ use-case.



*Hình 3.1. Lược đồ usecase của hệ thống ELearning*

### 3.1.1. Danh sách các tác nhân và mô tả:

|  |  |  |
| --- | --- | --- |
| **Tác nhân** | **Mô tả tác nhân** | **Ghi chú** |
| Sinh viên | Sinh viên là những người đang học tại trường đại học |  |
| Quản Lý | Là những người phụ trách thêm, xóa, sửa, nhập thông tin sinh viên, khoa, môn học, học phần,… |  |
| Giáo viên | Là những người phụ trách giảng dạy, nhập điểm,... |  |

### 3.1.2. Danh sách các Usecase:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Tên Use case** | **Mô tả ngắn gọn Use case** | **Chức năng** | **Ghi chú** |
| UC00 | Xem Thong Tin Tong Quan | Cho phép user xem thông tin cá nhân của mình tại trang homepage | Hiển thị thông tin cá nhân | Chung cho tất cả user |
| UC01.01 | Xem Danh Sach Hoc Ky | Cho phép sinh viên chọn học kỳ để đăng ký học phần | Chọn học kỳ đăng ký học phần |  |
| UC01.02 | Xem Danh Sach Hoc Phan | Hiển thị tất cả môn học phần được chọn từ học kỳ | Hiển thị học phần có thể đăng ký |  |
| UC01.03 | Xem Danh Sach LHP & Chi Tiet LHP | Hiển thị tất cả LHP từ môn học phần sinh viên chọn và thông tin chi tiết LHP đó | Hiển thị các LHP và thông tin chi tiết các LHP đó |  |
| UC01.04 | Kiem Tra Lich Trung LHP | Cho phép sinh viên checkbox kiểm tra trùng lịch học giữa các LHP muốn đăng ký và các LHP đã đăng ký | Hiển thị LHP trùng lịch học LHP |  |
| UC01.05 | Dang nhap | Cho phép sinh viên đăng nhập vào trang web | đăng nhập |  |
| UC01.06 | Dang Ky Hoc Phan | Cho phép sinh viên đăng ký một LHP đã chọn | Đăng ký học phần |  |
| UC01.07 | Huy Dang Ky Hoc Phan | Cho phép sinh viên hủy một LHP đã đăng ký | Hủy đăng ký học phần |  |
| UC01.08 | Xem LHP Da Dang Ky & Xem Chi Tiet | Hiển thị tất cả LHP sinh viên đã đăng ký trong 1 học kỳ nào đó và có thể xem chi tiết LHP đã đăng ký đó | Hiển thị danh sách LHP đã đăng ký và có thể xem chi tiết |  |
| UC02 | Quan Ly Tai Khoan | Cho phép quản lý tài khoản gồm: đổi mật khẩu, đăng xuất,.. | Đổi mật khẩu, đăng xuất tài khoản |  |
| UC03 | Xem Lich Hoc | Hiển thị thời khóa biểu lịch học theo tuần. | Xem lịch học |  |
| UC04 | Xem Ket Qua Hoc Tap | Hiển thị điểm tất cả môn học qua các học kỳ và nhận xét của GV dưới dạng bảng để sinh viên năm bắt kết quả học tập. | Xem điểm kết quả học tập |  |
| UC05 | Xem Chuong Trinh Khung | Cho phép sinh viên xem bộ chương trình khung chứa tất cả học kì và danh sách môn học tương ứng, số tín chỉ và chỉ tiêu. | Xem chương trình khung của ngành |  |
| UC06 | Quan Ly Sinh Vien | Cho phép quản trị viên chỉnh sửa danh sách sinh viên. | CRUD sinh viên |  |
| UC07 | Quan Ly Khoa Hoc | Cho phép quản trị viên chỉnh sửa thông tin các khóa học học kỳ | CRUD khóa học |  |
| UC08 | Quan Ly LHP | Cho phép quản trị viên chỉnh sửa các LHP | CRUD lớp học phần |  |
| UC09 | Quan Ly Thoi Khoa Bieu | Cho phép quản trị viên thêm xóa sửa thời khóa biểu. | CRUD thời khóa biểu |  |
| UC10 | Quan Ly Giao Vien | Cho phép quản trị viên them xóa sửa giáo viên. | Thêm môn học |  |
| UC11 | Quan Ly Lop Danh Nghia | Cho phép quản trị viên chỉnh sửa các lớp danh nghĩa | CRUD lớp danh nghĩa |  |
| UC12 | Quan Ly Sinh Vien Dang Ky Hoc Phan | Cho phép quản trị viên quản lý dữ liệu sinh viên đăng ký học phần các môn học | CRUD sinh viên đăng ký học phần |  |
| UC13 | Quan Ly Cong No | Cho phép quản trị viên gạch nợ sinh viên đăng ký học phần | CRUD công nợ |  |
| UC14 | Xem Lich Day | Cho phép giáo viên xem lịch dạy học | Hiển thị lịch dạy |  |
| UC15 | Quan Ly Diem Sinh Vien | Cho giao viên chỉnh sửa điểm học cho sinh viên | CRUD điểm |  |

## 3.2. Đặc tả các yêu cầu chứng năng (FUNCTIONAL):

- Sinh viên đăng ký theo học vào một lớp nhất định (lớp tín chỉ) với số sinh viên được đăng ký tối đa được qui định cho từng lớp.

- Mỗi môn học có một hoặc nhiều môn học tiên quyết. Sinh viên sẽ không được đăng ký môn học mà có môn tiên quyết mình chưa học.

- Mỗi học kỳ sinh viên được đăng ký tối đa 30 tín chỉ. Nếu quá số này, hệ thống sẽ không cho đăng ký.

## 3.3. Đặc tả các yêu cầu phi chứng năng (NON-FUNCTIONAL):

- Giao diện đẹp, dễ sử dụng.

- Bố cục layout các page phải đồng nhất

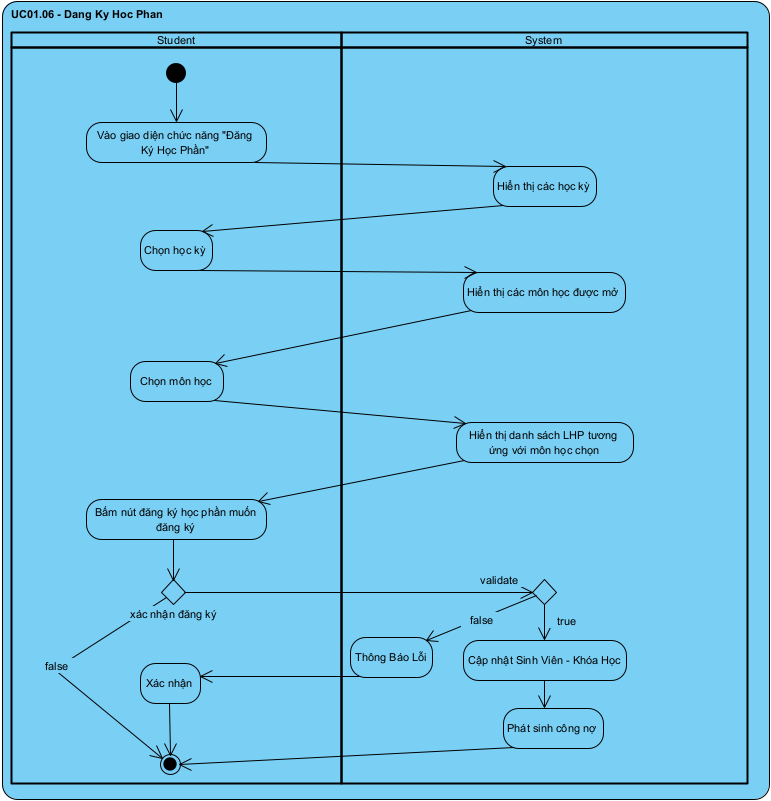
## 3.4. Các mô hình UML:

### 3.4.1. UC01.06 – Dang Ky Hoc Phan:

#### *3.4.1.1. Đặc tả usecase:*

|  |  |
| --- | --- |
| **Tên use case:** UC01.06 Đăng Ký Học Phần | |
| **Mô tả sơ lược chức năng:** chức năng Đăng Ký Học Phần giúp sinh viên đăng ký môn học mong muốn trong học kỳ. | |
| **Actor chính:** sinh viên | |
| **Actor phụ:** không có | |
| **Tiền điều kiện (Pre-condition):** Hệ thống hoạt động tốt, sinh viên còn đang học, học kỳ mới đã được giáo vụ mở, sinh viên đã đăng nhập vào trang homepage. | |
| **Hậu điều (Post-condition):** Cập nhật lại số lượng lớp học phần, thiết lập công nợ cho sinh viên. | |
| **Dòng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Sinh viên chọn nút chức năng “Đăng ký Học phần” |  |
|  | 2. Hệ thống chuyển trang giao diện Đăng Ký Học Phần. |
| 3. Sinh viên chọn học kỳ hiện tại để đăng ký học phần |  |
|  | 4. Hệ thống hiển thị danh sách môn học được mở |
| 5. Sinh viên chọn môn học |  |
|  | 6. Hệ thống hiển thị danh sách LHP được mở ứng với môn học |
| 7. Sinh viên chọn lớp & nhóm học phần muốn đăng ký |  |
| 8. Sinh viên bấm nút “Đăng Ký Học Phần” |  |
|  | 9. Hệ thống kiểm tra thông tin |
|  | 10. Hệ thống cập nhật tin sinh viên đăng ký khóa học thành công. |
|  | 11. Hệ thống phát sinh công nợ sinh viên. |
| **Dòng sự kiện thay thế (alternate flow):** | |
| 9.x Sinh viên xác nhận | 9.1.1 Hệ thống phát hiện LHP đã khóa  9.1.2 Hệ thống thông báo lỗi  9.2.1 Hệ thống phát hiện sinh viên chưa hoàn thành học phần tiên quyết  9.2.2 Hệ thống thông báo lỗi  9.3.1 Lớp học phần đã đủ số lượng người đăng ký  9.3.2 Hệ thống thông báo lỗi  9.4.1 Hệ thống phát hiện sinh viên bị trùng lịch  9.4.2 Hệ thống thông báo lỗi |
| **Dòng sự kiện ngoại lệ (exception flow):** không có | |

#### 3.4.1.2. Activity Diagram UC01.06:



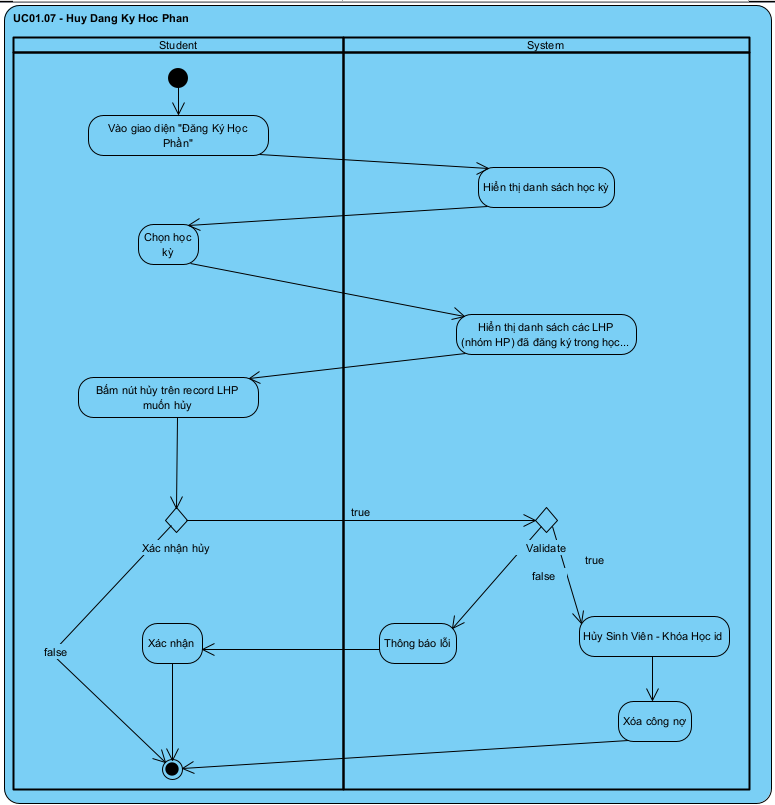
#### 3.4.1.3. Sequence Diagram UC01.06:

### 3.4.2. UC01.07 – Huy Dang Ky Hoc Phan:

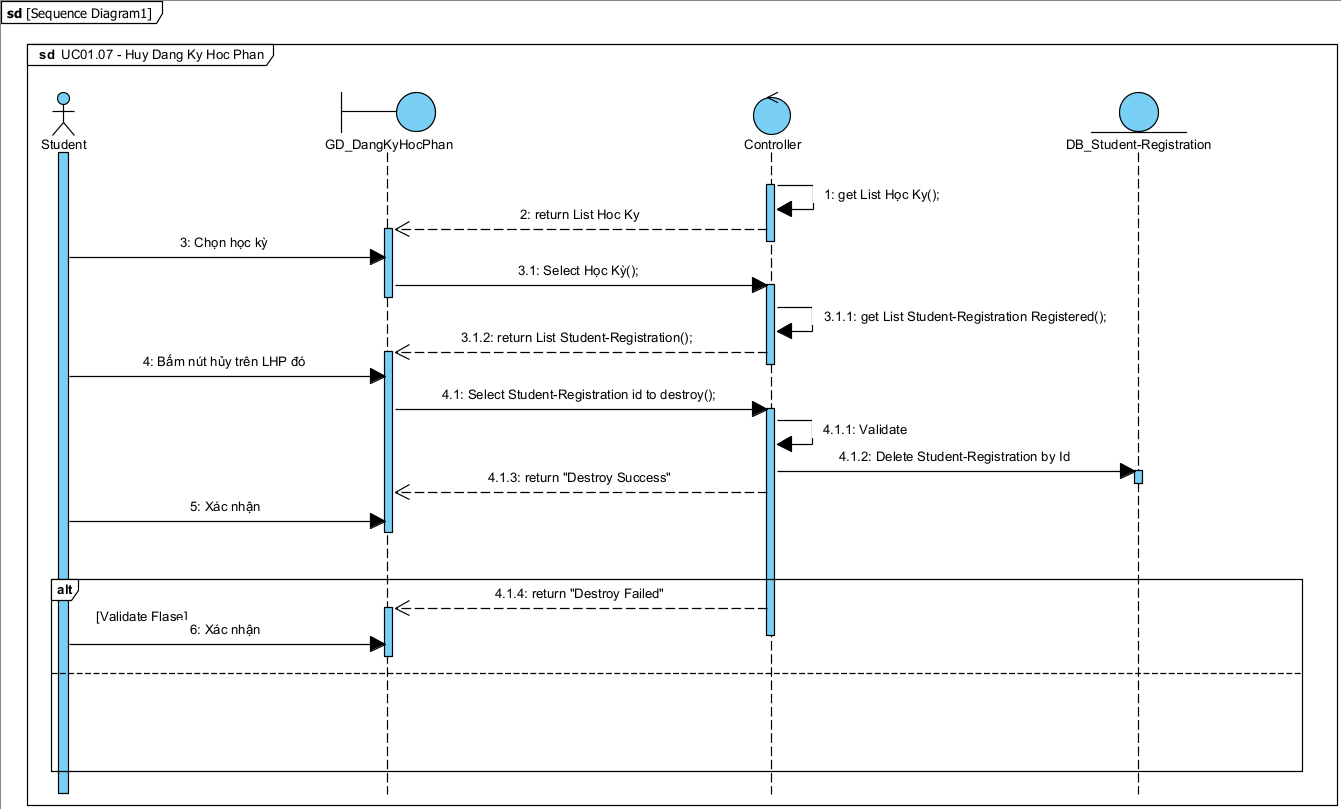
#### *3.4.2.1. Đặc tả usecase:*

|  |  |
| --- | --- |
| **Tên use case:** UC01.07 Huy Dang Ky Hoc Phan | |
| **Mô tả sơ lược chức năng:** chức năng hủy Học Phần đã đăng ký giúp sinh viên có thể suy nghĩ lại | |
| **Actor chính:** sinh viên | |
| **Actor phụ:** không có | |
| **Tiền điều kiện (Pre-condition):** Hệ thống hoạt động tốt, sinh viên còn đang học, lớp học phần vẫn chưa ở trạng thái Chấp Nhận Mở Lớp hoặc Đã Khóa, sinh viên đã đăng nhập vào trang homepage. | |
| **Hậu điều (Post-condition):** Cập nhật lại số lượng lớp học phần trừ đi 1 | |
| **Dòng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Sinh viên chọn nút chức năng “Đăng ký Học phần” |  |
|  | 2. Hệ thống chuyển trang giao diện Đăng Ký Học Phần. |
| 3. Sinh viên chọn học kỳ hiện tại để đăng ký học phần |  |
|  | 4. Hệ thống hiển thị danh sách lớp học phần đã đăng ký trong học kỳ |
| 5. Sinh viên bấm nút HỦY trên lớp học phần (hoặc nhóm học phần). |  |
|  | 6. Hệ thống kiểm tra thông tin |
|  | 7. Hệ thống hủy đăng ký LHP cho sinh viên và thông báo hủy thành công |
| 7. Xác nhận |  |
| **Dòng sự kiện thay thế (alternate flow):** | |
| 6.x Xác nhận | 6.1.1 Hệ thống phát hiện LHP đã ở trạng thái “Chấp Nhận Mở Lớp” hoặc “Đã Khóa” |
|  | 6.1.2 Hệ thống thông báo lỗi |
|  | 6.2.1 Hệ thống phát hiện sinh viên đã đóng học phí |
|  | 6.2.2 Hệ thống báo lỗi |
| **Dòng sự kiện ngoại lệ (exception flow):** | |
|  | 8. Hệ thống cập nhật lại số lượng sinh viên trong lớp học phần |

#### *3.4.2.2. Activity Diagram:*



#### *3.4.2.3. Sequence Diagram:*



### 3.4.3. UC01.04 – Kiem Tra Trung Lich LHP:

#### *3.4.3.2. Activity Diagram:*

*<<Dán Ảnh Dô Đây Dùm Con Mấy Ba, Mấy Má>>*

#### *3.4.3.3. Sequence Diagram:*

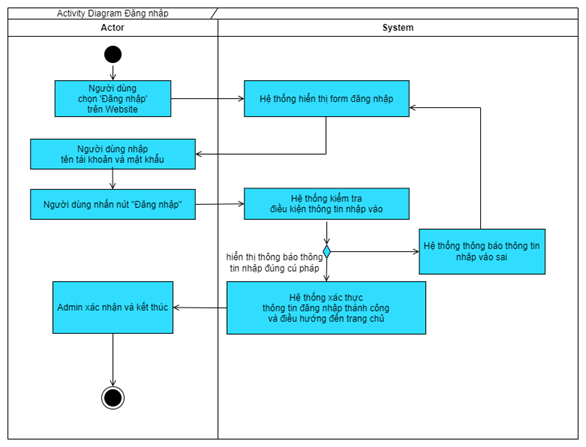
*<<Dán Ảnh Dô Đây Dùm Con Mấy Ba, Mấy Má>>*

### 3.4.4. UC01.05 – Đăng nhập:

**3.4.4.1. Đặc tả usecase:**

|  |  |
| --- | --- |
| **Tên use case:** Đăng nhập | |
| **Mô tả sơ lược:** Người dùng đăng nhập để thao tác được các chức năng yêu cầu đăng nhập | |
| **Actor:** Admin, Sinh viên, Giảng viên | |
| **Tiền điều kiện:** Phải có tài khoản đã tạo và kích hoạt trước trên Website | |
| **Hậu điều kiện:** Đăng nhập thành công | |
| **Luồng sự kiện chính (Basic Flow)** | |
| **Tác nhân (Actor)** | **Hệ thống (System)** |
| 1. Người dùng chọn ‘Đăng nhập’ trên Website |  |
|  | 1. Hệ thống hiển thị form đăng nhập |
| 1. Người dùng nhập tên tài khoản và mật khẩu |  |
| 1. Người dùng nhấn nút Đăng nhập |  |
|  | 1. Hệ thống kiểm tra điều kiện thông tin nhập vào |
|  | 1. Hệ thống thông báo đăng nhập thành công |
|  | 1. Hệ thống cập nhật dữ liệu lên giao diện |
| **Luồng sự kiện phụ (Alternative Flow)** | |
|  |  |
| **Luồng sự kiện ngoại lệ (Exception Flow)** | |
|  | 5.1. Hệ thống thông báo thông tin nhập vào sai. |
|  | 5.2. Quay lại bước 2 |

#### *3.4.4.1. Activity Diagram:*



#### *3.4.4.2. Sequence Diagram:*

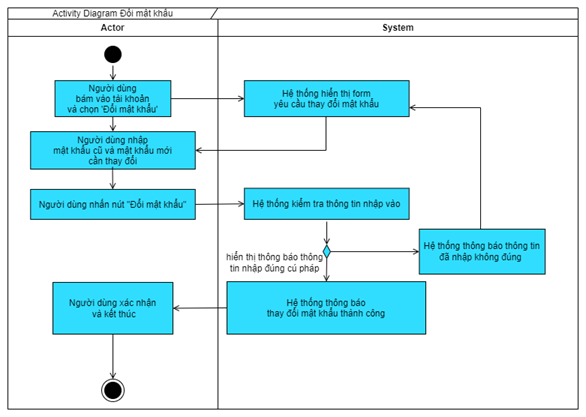


### 3.4.5. UC01.09 – Doi Mat Khau:

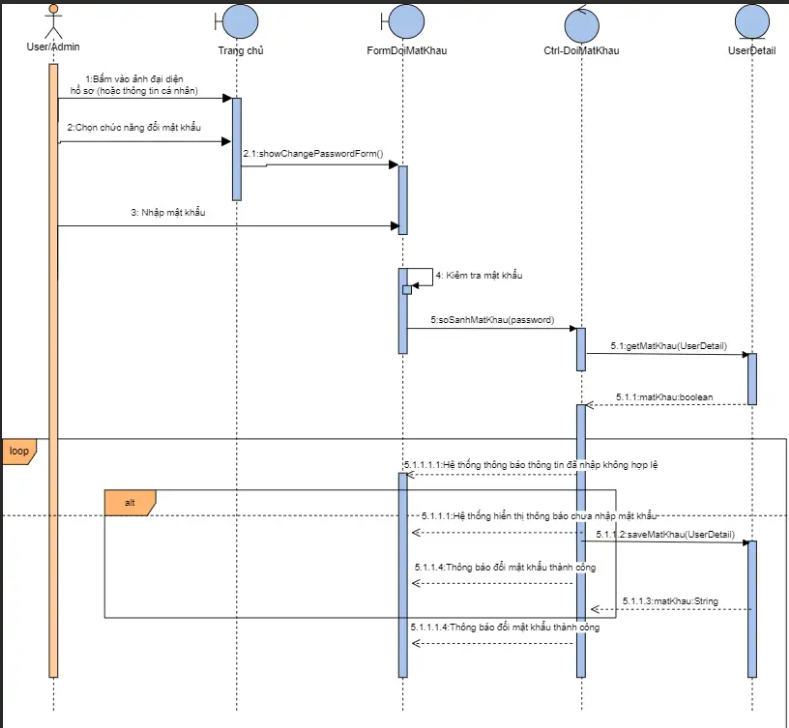
**3.4.5.1. Đặc tả usecase:**

|  |  |
| --- | --- |
| **Tên use case: Đổi mật khẩu** | |
| **Mô tả sơ lược:** Chức năng giúp người dùng thay đổi mật khẩu của tài khoản cá nhân khi cần | |
| **Actor:** Admin, sinh viên | |
| **Tiền điều kiện:** Nhập đúng Email, Tên đăng nhập và mật khẩu | |
| **Hậu điều kiện:** Sau khi làm mới mật khẩu thành công, hệ thống sẽ lưu tài khoản cùng mật khẩu mới. Người dùng có thể đăng nhập bằng mật khảu mới | |
| **Luồng sự kiện chính (Basic Flow)** | |
| **Tác nhân (Actor)** | **Hệ thống (System)** |
| 1. Người dùng bấm vào tài khoản và chọn ‘Đổi mật khẩu’ |  |
|  | 1. Hệ thống hiển thị form yêu cầu thay đổi mật khẩu |
| 1. Người dùng nhập mật khẩu cũ và mật khẩu mới cần thay đổi |  |
| 1. Người dùng nhấn nút Đổi mật khẩu |  |
|  | 1. Hệ thống kiểm tra thông tin nhập vào |
|  | 1. Hệ thống thông báo thay đổi mật khẩu thành công |
| **Luồng sự kiện phụ (Alternative Flow)** | |
|  |  |
| **Luồng sự kiện ngoại lệ (Exception Flow)** | |
|  | 5.1. Hệ thống thông báo thông tin đã nhập không đúng. |
|  | 5.2. Quay lại bước 2 |

#### *3.4.5.2. Activity Diagram:*



#### *3.4.5.3. Sequence Diagram:*

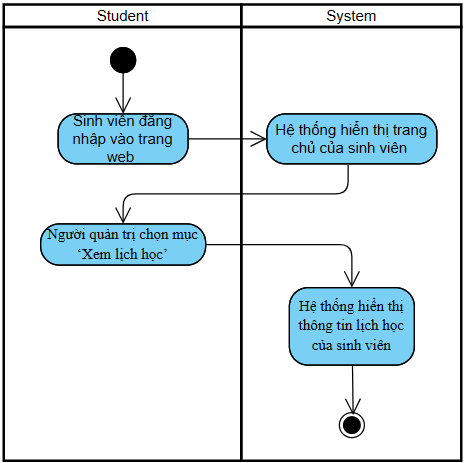


### 3.4.6. UC01.10 – Xem Lich Hoc:

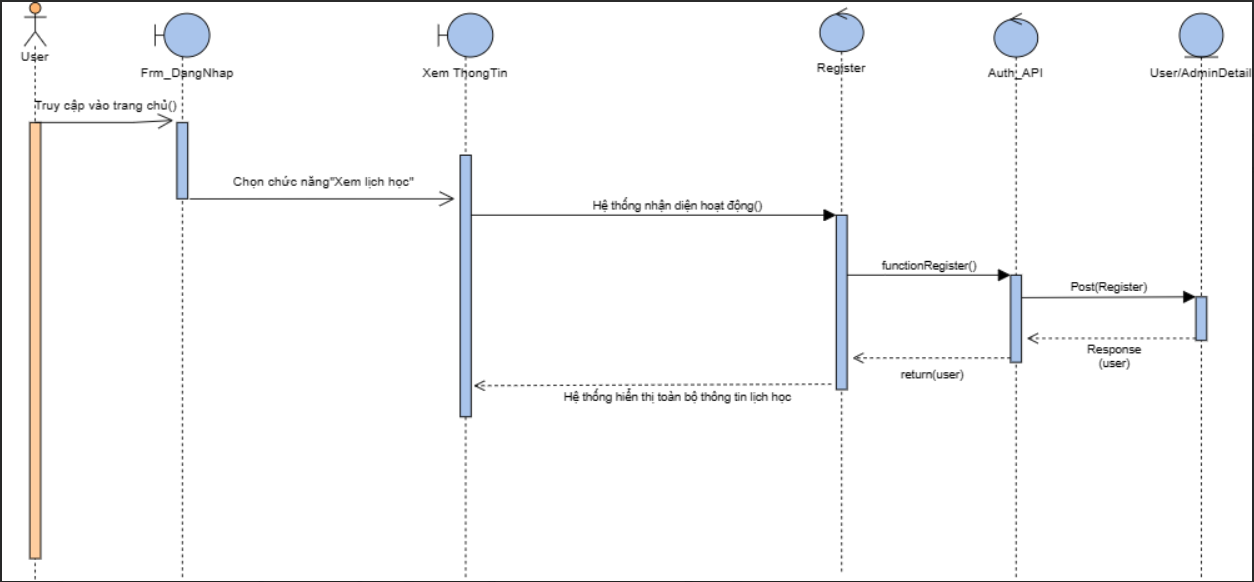
**3.4.6.1. Đặc tả usecase:**

|  |  |
| --- | --- |
| **Tên use case: Xem thông tin CV ứng viên** | |
| **Mô tả sơ lược:** Usecase xem lịch học sinh viên đã đăng ký | |
| **Actor:** Admin, Sinh viên | |
| **Tiền điều kiện:** Đăng nhập thành công | |
| **Hậu điều kiện:** Xem được thông tin lịch học sinh viên trên Website | |
| **Luồng sự kiện chính (Basic Flow)** | |
| **Tác nhân (Actor)** | **Hệ thống (System)** |
| 1. Sinh viên đăng nhập vào trang web |  |
|  | 1. Hệ thống hiển thị trang chủ của sinh viên |
| 1. Người quản trị chọn mục ‘Xem lịch học’ |  |
|  | 1. Hệ thống hiển thị thông tin lịch học của sinh viên |
| **Luồng sự kiện phụ (Alternative Flow)** | |
|  |  |
| **Luồng sự kiện ngoại lệ (Exception Flow)** | |
|  |  |

#### *3.4.6.2. Activity Diagram:*



#### *3.4.6.3. Sequence Diagram:*

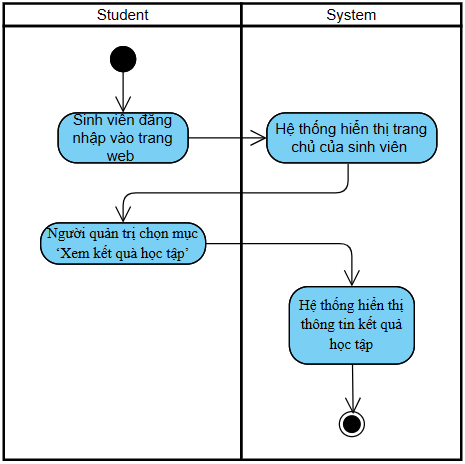


### 3.4.7. UC01.11 – Xem Ket Qua Hoc Tap:

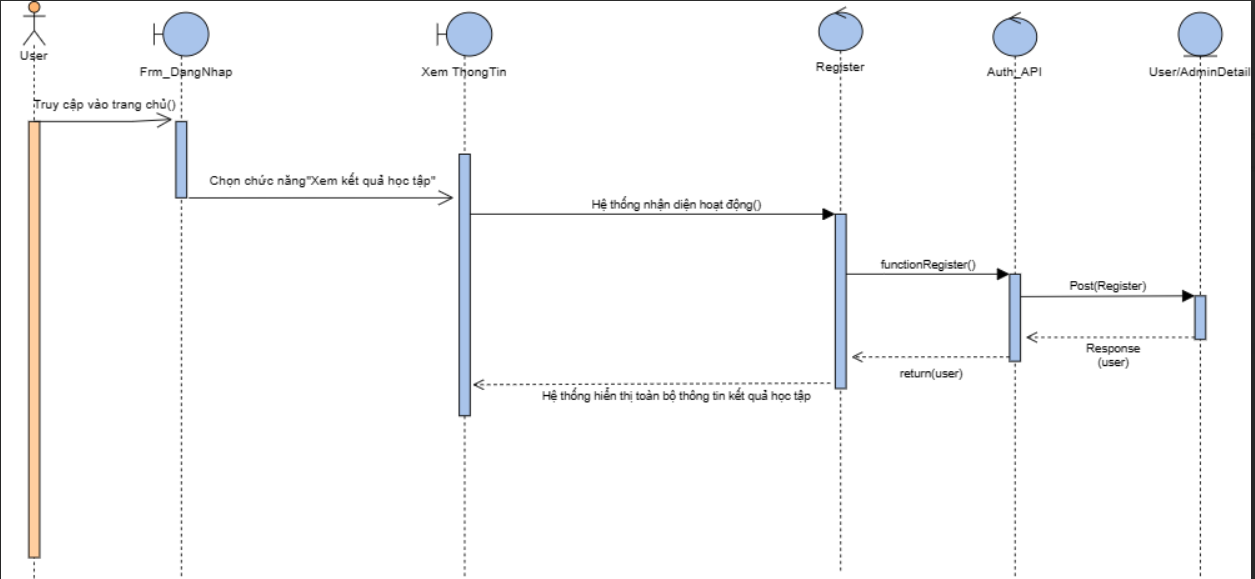
**3.4.7.1. Đặc tả usecase:**

|  |  |
| --- | --- |
| **Tên use case: Xem Kết Quả Học Tập** | |
| **Mô tả sơ lược:** Usecase xem kết quả học tập của sinh viên | |
| **Actor:** Admin, Sinh viên | |
| **Tiền điều kiện:** Đăng nhập thành công | |
| **Hậu điều kiện:** Xem được kết quả học tập của sinh viên | |
| **Luồng sự kiện chính (Basic Flow)** | |
| **Tác nhân (Actor)** | **Hệ thống (System)** |
| 1. Sinh viên đăng nhập vào trang web |  |
|  | 1. Hệ thống hiển thị trang chủ của sinh viên |
| 1. Người quản trị chọn mục ‘Xem kết quả học tập’ |  |
|  | 1. Hệ thống hiển thị thông tin kết quả học tập |
| **Luồng sự kiện phụ (Alternative Flow)** | |
|  |  |
| **Luồng sự kiện ngoại lệ (Exception Flow)** | |
|  |  |

#### *3.4.7.2. Activity Diagram:*



#### *3.4.7.3. Sequence Diagram:*

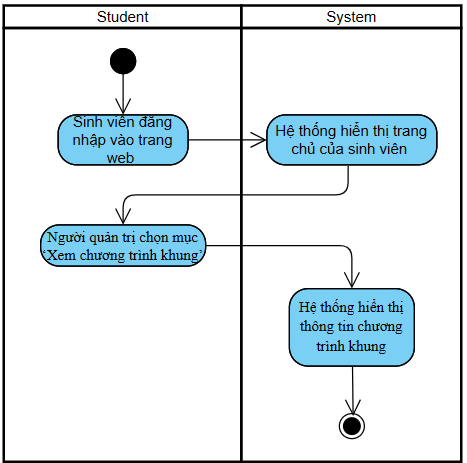


### 3.4.8. UC01.12 – Xem Chuong Trinh Khung:

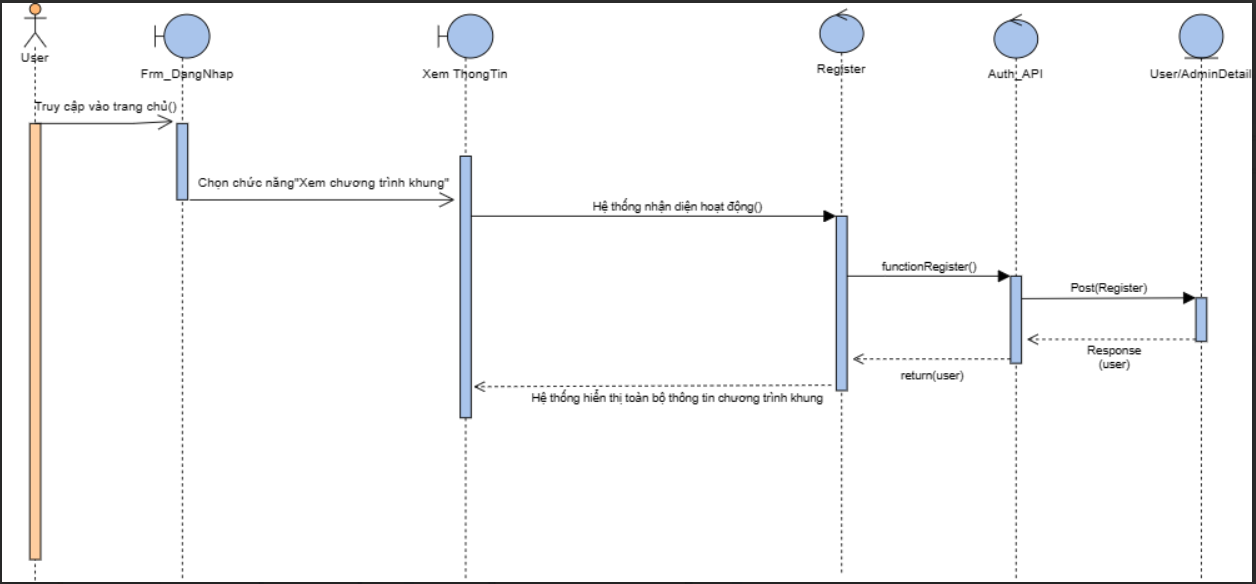
**3.4.8.1. Đặc tả usecase:**

|  |  |
| --- | --- |
| **Tên use case: Xem Kết Quả Học Tập** | |
| **Mô tả sơ lược:** Usecase xem chương trình khung của sinh viên | |
| **Actor:** Admin, Sinh viên | |
| **Tiền điều kiện:** Đăng nhập thành công | |
| **Hậu điều kiện:** Xem được chương trình khung của sinh viên | |
| **Luồng sự kiện chính (Basic Flow)** | |
| **Tác nhân (Actor)** | **Hệ thống (System)** |
| 1. Sinh viên đăng nhập vào trang web |  |
|  | 1. Hệ thống hiển thị trang chủ của sinh viên |
| 1. Người quản trị chọn mục ‘Xem chương trình khung’ |  |
|  | 1. Hệ thống hiển thị thông tin chương trình khung |
| **Luồng sự kiện phụ (Alternative Flow)** | |
|  |  |

#### *3.4.8.2. Activity Diagram:*



#### *3.4.8.3. Sequence Diagram:*



### 3.4.9. UC01.15 – Mo Lop Hoc Phan:

#### *3.4.9.2. Activity Diagram:*

*<<Dán Ảnh Dô Đây Dùm Con Mấy Ba, Mấy Má>>*

#### *3.4.9.3. Sequence Diagram:*

*<<Dán Ảnh Dô Đây Dùm Con Mấy Ba, Mấy Má>>*

### 3.4.10. UC01.16 – Tao Thoi Khoa Bieu:

#### *3.4.10.2. Activity Diagram:*

*<<Dán Ảnh Dô Đây Dùm Con Mấy Ba, Mấy Má>>*

#### *3.4.10.3. Sequence Diagram:*

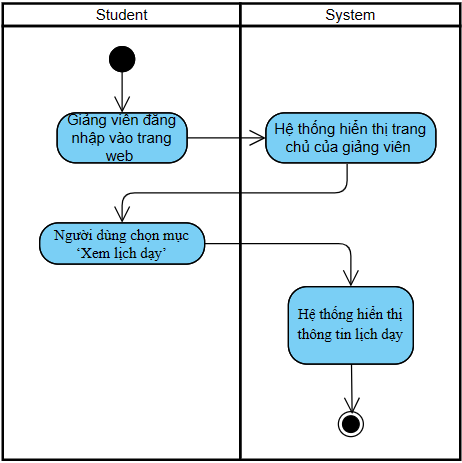
*<<Dán Ảnh Dô Đây Dùm Con Mấy Ba, Mấy Má>>*

### 3.4.11. UC01.21 – Xem Lich Day:

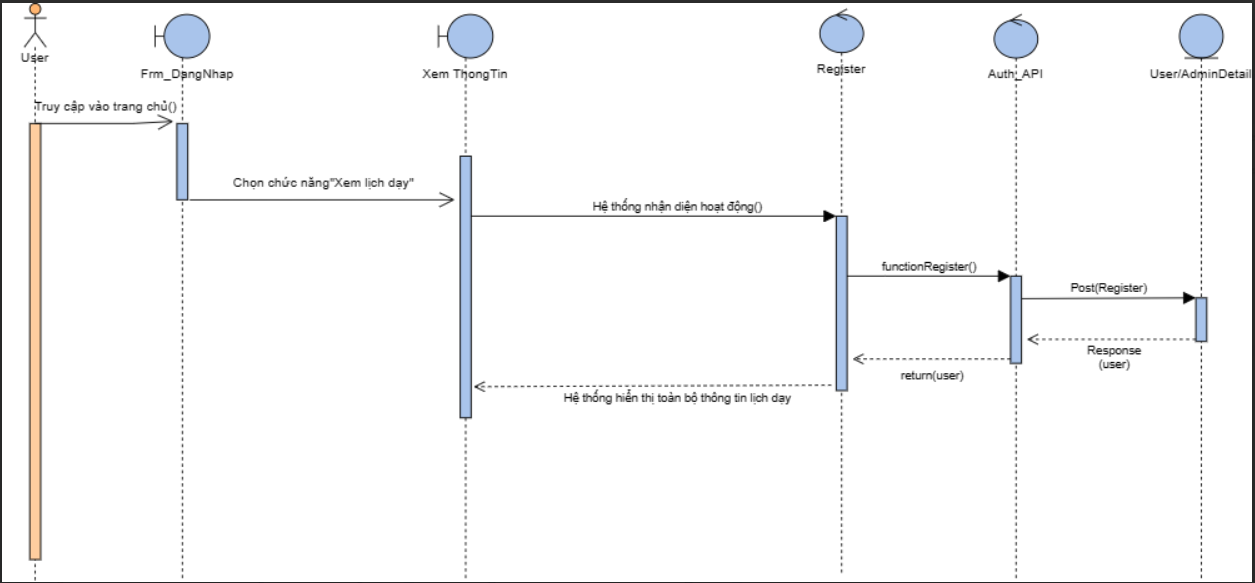
**3.4.11.1. Đặc tả usecase:**

|  |  |
| --- | --- |
| **Tên use case: Xem Kết Quả Học Tập** | |
| **Mô tả sơ lược:** Usecase xem lịch dạy của giảng viên | |
| **Actor:** Admin, Giảng viên | |
| **Tiền điều kiện:** Đăng nhập thành công | |
| **Hậu điều kiện:** Xem được lịch dạy của giảng viên | |
| **Luồng sự kiện chính (Basic Flow)** | |
| **Tác nhân (Actor)** | **Hệ thống (System)** |
| 1. Giảng viên đăng nhập vào trang web |  |
|  | 1. Hệ thống hiển thị trang chủ của giảng viên |
| 1. Người dùng chọn mục ‘Xem lịch dạy của giảng viên’ |  |
|  | 1. Hệ thống hiển thị thông tin lịch dạy của giảng viên |
| **Luồng sự kiện phụ (Alternative Flow)** | |
|  |  |

#### *3.4.11.2. Activity Diagram:*



#### *3.4.11.3. Sequence Diagram:*

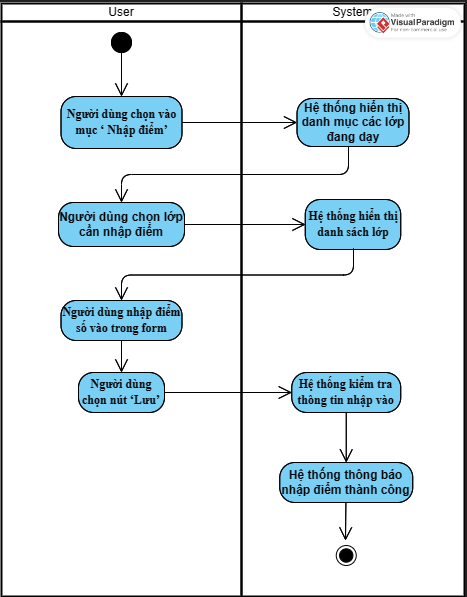


### 3.4.12. UC01.22 – Nhập Điểm Cho Sinh Viên:

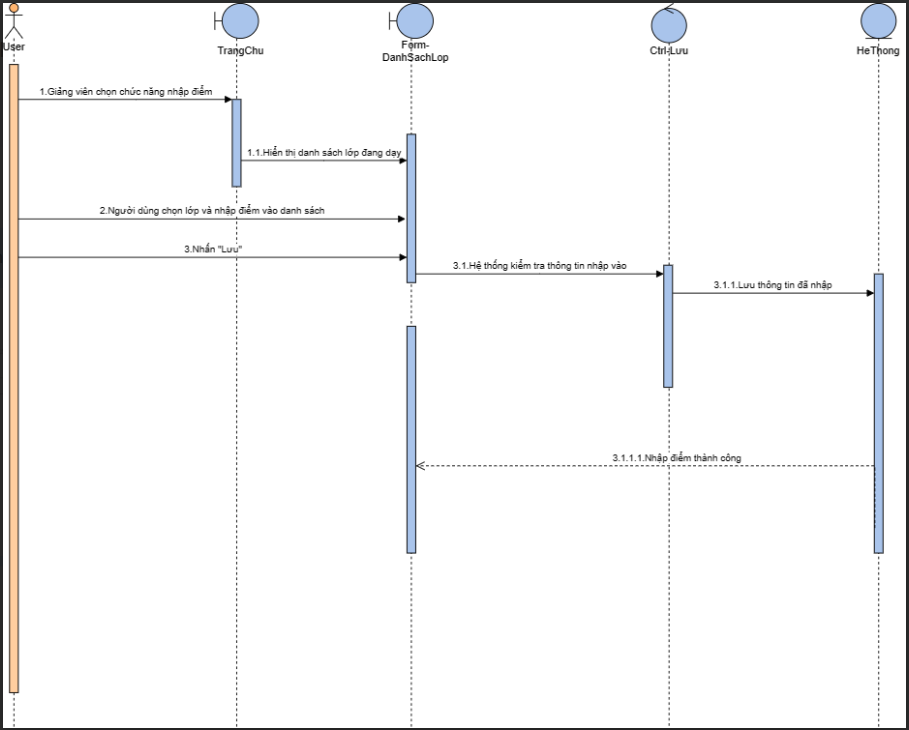
**3.4.12.1. Đặc tả usecase:**

|  |  |
| --- | --- |
| **Tên use case: Nhập Diểm Cho Sinh Viên** | |
| **Mô tả sơ lược:** Usecase nhập điểm cho sinh viên | |
| **Actor:** Admin, Giảng viên | |
| **Tiền điều kiện:** Đăng nhập thành công | |
| **Hậu điều kiện: N**hập được điểm cho sinh viên | |
| **Luồng sự kiện chính (Basic Flow)** | |
| **Tác nhân (Actor)** | **Hệ thống (System)** |
| 1. Người dùng chọn vào mục ‘ Nhập điểm’ |  |
|  | 1. Hệ thống hiển thị danh mục các lớp đang dạy |
| 1. Người dùng chọn lớp cần nhập điểm |  |
|  | 1. Hệ thống hiển thị danh sách lớp |
| 1. Người dùng nhập điểm số vào trong form |  |
| 1. Người dùng chọn nút ‘Lưu’ |  |
|  | 1. Hệ thống kiểm tra thông tin nhập vào |
|  | 1. Hệ thống thông báo nhập điểm thành công |
| **Luồng sự kiện phụ (Alternative Flow)** | |
|  |  |

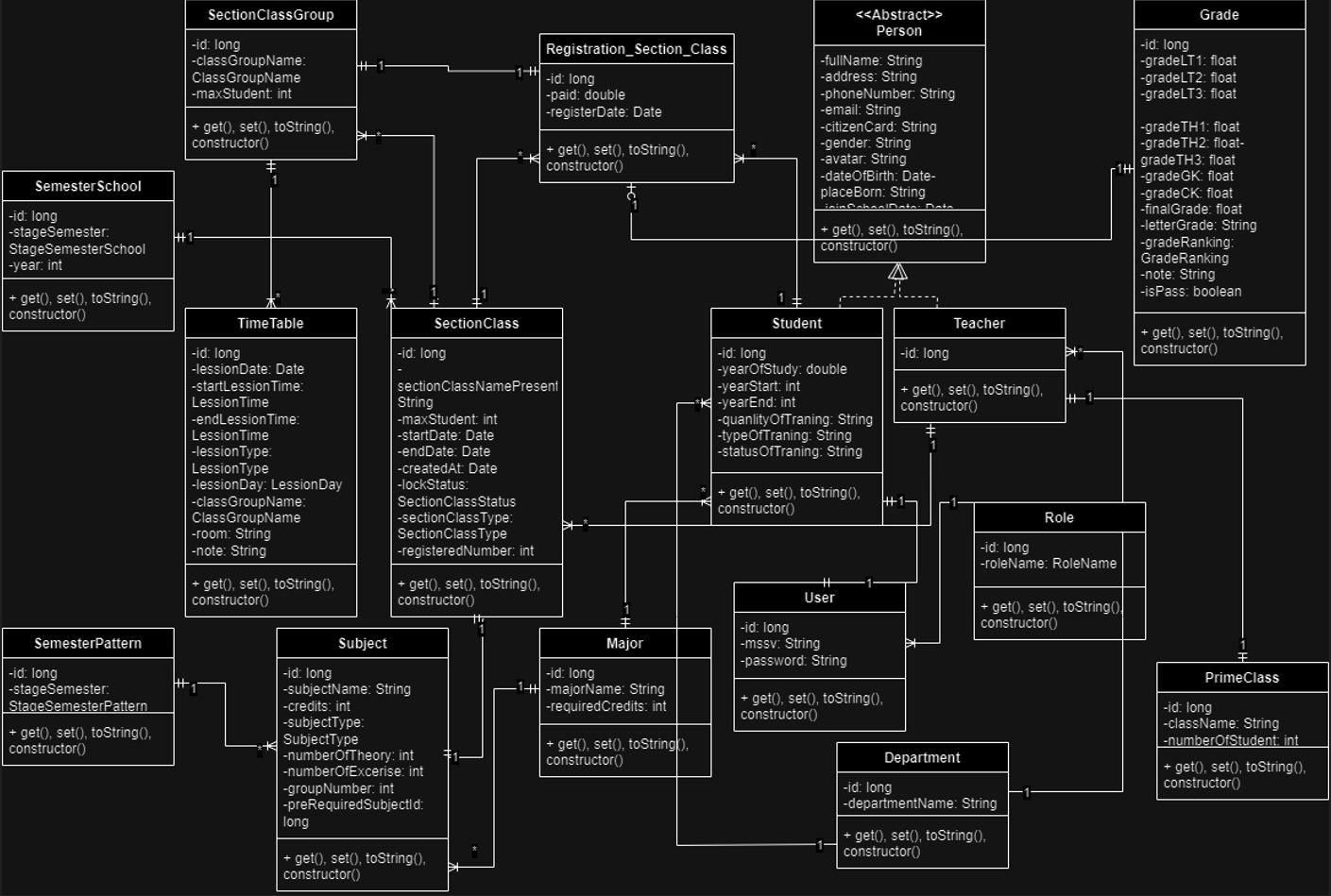
#### *3.4.12.2. Activity Diagram:*



#### *3.4.12.3. Sequence Diagram:*



## 3.5. Class Diagram:



Sơ đồ class trên mô tả tất cả đối tượng chính tham gia vào việc quản lý và xây dựng chức năng cho tất cả (phần lớn) đối tượng phục vụ cho học vụ tại trường đại học. Tất cả các lớp đều có mối quan hệ chặt chẽ với nhau.

Class **Student**: là lớp chứa các thuộc tính của thông tin sinh viên và các thuộc tính đa trị liên đới tới các lớp liên quan thuận tiện cho việc truy vấn dữ liệu object relation.

Class **SemesterPattern**: mô tả các môn học sinh viên đã hoàn thành trong chương trình học phần theo học kì. Từ đó truy vấn được tín chỉ hoàn thành cho sinh viên.

Class **SectionClass**: mô tả thuộc tính cơ bản của lớp học phần được tạo ra do nhân viện (giáo vụ hoặc manager) quản lý chính.

Class **RegistrationSectionClass**: là lớp sinh ra do việc ‘Student’ thực hiện chức năng đăng ký học phần với lớp ‘SectionClass’ và được lưu trữ lại.

Class **Teacher**: chứa các thông tin giáo viên/giảng viên.

Class **User** & **Role**: là lớp chứa thông tin tài khoản, lưu trữ tên người dùng, mật khẩu, email của sinh viên phục vụ cho việc đăng nhập vào hệ thống và quản lý thông tin sinh viên. Lớp tài khoản chứa nhiều đối tượng người dùng như bao gồm cả: sinh viên, giáo viên, quản lý.

Class **SemesterSchool**: là lớp chứa thông tin các khóa học theo các năm. Mỗi năm sẽ có 3 học kỳ (tương ứng học kì 1,2,3). Khóa học được tạo ra do nhân viên tạo để mở lớp học phần và LHP đưa vào khóa học đó.

Class **Department**: lưu trữ thông tin các khoa tồn tại tại trường đại học. Các khoa được phân chia sẵn và lớp này sẽ lưu trữ data import vào. Một khoa sẽ chứa nhiều ngành (1-\*).

Class **Major**: hay chuyên ngành là lớp chứa thông tin các ngành học có dạy tại trường đại học. Mỗi sinh viên đầu vào đều chọn ngành yêu thích. Ngành thuộc đơn vị quản lý của khoa.

Class **TimeTable**: là lớp chưa thông tin của thời khóa biểu dùng để cho sinh viên xem lịch học hoặc giảng viên xem lịch dạy. Thời khóa biểu được sinh ra dựa trên lớp học phần. Mỗi LHP sẽ có thời khóa biểu riêng do nhân viên quản lý. Thời khóa biểu còn chia làm 2 loại là Lý Thuyết và Thực Hành. 1 thời khóa biểu LT lại có thể chứa nhiều TKB thực hành do phân nhóm thực hành vì linh kiện 1 phòng không đủ. Thường 1 thời khóa biểu TH sẽ thuộc 1 phòng học khác nhau và sẽ khác số lượng sinh viên.

Class **Subject**: là lớp đơn trị chứa các thông tin của một môn học. Mỗi môn học sẽ thuộc một ngành nào đó và nằm thuộc một chương trình khung (SemesterPattern) nào đó. Để thuận tiện việc lấy ra tất cả môn học theo trường trình khung cho sinh viên.

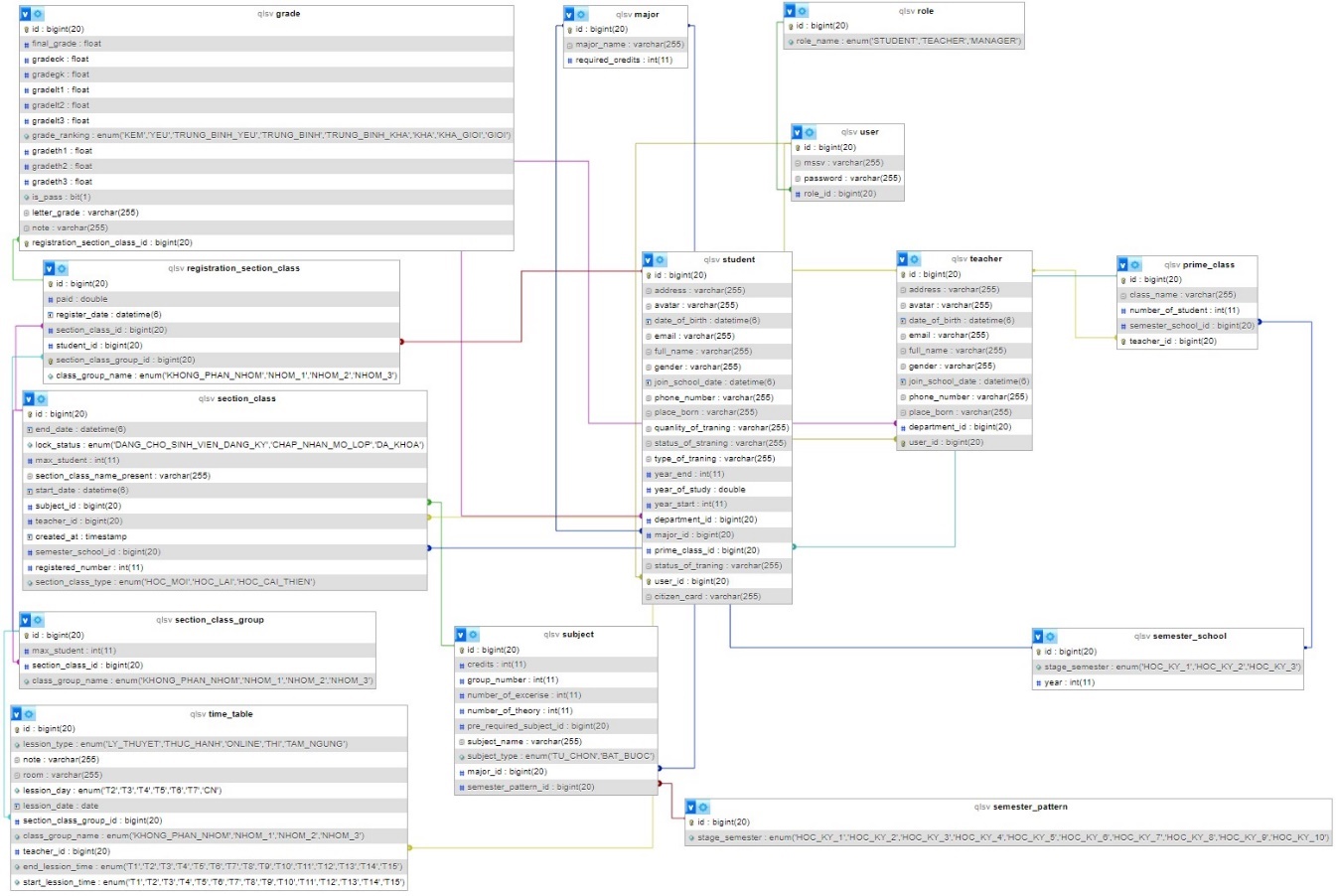
Class **PrimeClass**: là lớp lưu trữ thông tin của một lớp học cơ bản như lớp học ta học hồi tiểu học hoặc trung học. Mộ lớp học danh nghĩa chứa nhiều sinh viên thuộc ngành nào đó do nhân viên quản lý. Lớp này hỗ trợ cho việc import, export ra các thành phần có nhiều sinh viên trong đó. Ví dụ: danh sách lịch thi, danh sách để điểm danh…

Class **Grade**: là lớp chứa các thuộc tính điểm của 1 sinh viên. Điểm rất da dạng vì lại chia làm loại điểm LT và điểm LT mỗi đối tượng điểm sẽ có công thức tính điểm riêng do công thức nhà trường quy định. Công thức đã được tích hợp sẵn trong hàm. Lớp này còn giúp sinh viên thống kê lại kết quả học tập, biết được năng lực của sinh viên và nhà trường có thể đánh giá được học lực của sinh viên đó.

Class **Manager**: là lớp đơn trị lưu trữ thông tin của tất cả nhân viên với nhiều chức vụ hoạt động trong nhà trường.

## 3.6. ERD Diagram

Ở đây chúng tôi có 16 bảng cơ sở dữ liệu quan hệ. Chúng tôi xin tập trung giới thiệu vào các bảng chính của lược đồ này. Bảng student là bảng trung tâm lưu trữ thông tin cá nhân của sinh viên và liên đới tới các bảng khác. Tất cả các bảng đều có khóa chính với name là “id” kiểu long và tự generate khi có record mới được them vào. Kiểu dữ liệu này mang lại lợi ích tự động phát sinh khi thêm mới, tốc độ truy vấn nhanh, tiết kiệm tài nguyên lưu trữ. Ở bảng user chúng tôi có access\_token dùng để phân quyền và xác thực cho user là một trong ba đối tượng sinh viên/giảng viên/quản lý. Các relationship giữa các bảng sẽ không bị *CASCADE.ALL* khi có thao tác CUD sai nhằm đảm bảo dữ liệu không bị mất một cách vô lý (sai logic).



*Hình 3.6: Lược đồ cơ sở dữ liệu hệ thống ELearning*

Bảng mon\_hoc (môn học) chứa thông tin của các môn học. Mỗi môn học lại có nhiều học phần liên quan (bảng hoc\_phan) với số tiết, số lượng tín chỉ khác nhau. Mỗi học phần có nhiều lớp học phần được mở trong các học kì khác nhau (lop\_hoc\_phan). Mỗi lớp học phần sẽ có các thời khóa biểu (bảng thoi\_khoa\_bieu). Mỗi thời khóa biểu là 1 ngày học trong tuần với các thông tin như tiết học, mã phòng học, thời gian bắt đầu và thời gian kết thúc.

# CHƯƠNG 4: THIẾT KẾ VÀ HIỆN THỰC

## Bài toán

Sinh viên hiện nay sử dụng smartphone tương đối đông đảo, hầu như chiếm đa số trong toàn trường, nhưng nhà trường chưa có một ứng dụng được xây dựng trên hệ điều hành android/IOS để phục vụ mục đích học tập sinh viên, giả sử sinh viên cần xem lịch học trong ngày, thay vì vào web trên điện thoại sẽ rất bất tiện. Thì sinh viên có thể truy cập app. Như đến thời điểm đăng ký học phần mới cho năm học, sinh viên phải rất vất vả như không có laptop, không có Ngoài ra, sinh viên sẽ truy cập số lượng lớn vào các thời điểm như đăng ký học phần, số lượng request xảy ra nhiều khi tuyển sinh. Và sẽ giảm sau đó, cần server sẽ tự scaling giúp giảm chi phí và giúp server không sập lúc cao điểm. Dẫn đến sẽ cần quản lý thêm vấn đề có nhiều hơn 1 container trên 1 instance. Với cách deploy truyền thống, nếu có 1 application server trên 1 instance, và trỏ server đó đến port 80, và vậy là xong, nó có thể được truy cập thông qua giao thức HTTP. Vậy trong trường hợp có hai container trên nhiều instance thì sao? Chúng ta không thể trỏ trực tiếp 2 container tới cùng 1 cổng 80 được.

## Kiến trúc hệ thống

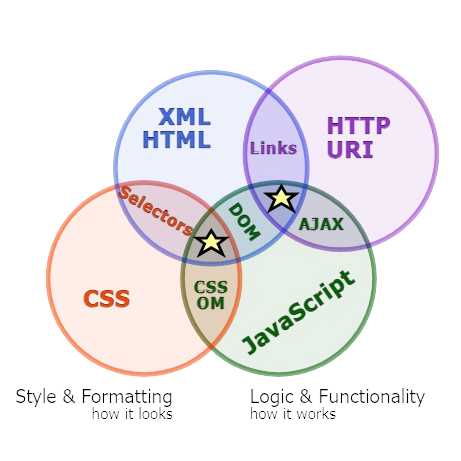
Để triển khai các giải pháp mà chúng tôi nêu ra trong phần giới thiệu, chúng tôi đề xuất xây dựng ứng dụng IUH. Ứng dụng này chúng tôi xây dựng dựa vào các công nghệ như Spring Boot, ReactJS, React Native. Đồng thời, để triển khai ứng dụng IUH đến thiết bị người dùng (sinh viên, giảng viên và nhân viên), chúng tôi cũng dùng các dịch vụ như AWS và CI/CD. Hệ thống IUH gồm ứng dụng trên điện thoại (app) và hệ thống website. Những kiến trúc này được áp dụng từ kiến thức từ chương trình đã học từ các đã môn học tại trường IUH.

|  |
| --- |
|  |
| Hình 4.2.1 Sơ lược tổng quan mô hình kiến trúc toàn hệ thống |

Trong đó, React Native tạo ứng dụng trên điện thoại hệ điều hành Android/iOS, phục vụ cho các chức năng của sinh viên. ReactJS tạo website cho admin, các chức năng cho admin như tuyển sinh, quản lý đào tạo… Spring Boot chịu trách nhiệm tạo ra các API, xử lý các request. MySQL dùng lưu trữ dữ liệu của toàn hệ thống như dữ liệu sinh viên, dữ liệu học phần…Các dịch vụ điện toán đám mây sử dụng trong AWS gồm: Elastic Container Service (ECS); RDS; Auto Scaling; Load Balancer; Elastic Container Registry (ECR hoặc Docker hub). Cụ thể các dịch vụ này đảm nhiệm các công việc như sau: ECS điều phối container hiệu suất cao có hỗ trợ Docker; RDS lưu trữ dữ liệu quan hệ của hệ thống trên cloud; AutoScaling giúp tạo hoặc tắt số lượng server tùy vào lượng truy cập vào web/app trong một thời điểm; Load Balancer giúp điều hướng lượng truy cập, giúp cân bằng tải cho số lượng server hiện có và ECR, nơi lưu trữ các hình ảnh trên AWS.

Dựa vào kiến trúc Microservices, chúng tôi tách hệ thống thành các dịch vụ riêng như hình bên dưới (Hình 4.2.1). Đồng thời, chúng tôi cũng đề xuất phần AWS Load Balancer cho việc triển khai hệ thống trên nền tảng điện toán đám mây. Ngoài ra, dữ liệu chúng tôi cũng đề xuất tổ chức lưu trữ trên AWS RDS MySQL.

### 4.2.1. Triển khai Web Flatform



Hình 4.2.2 Flatform web ELearning

Ứng dụng ELearning hỗ trợ học vụ được triển khai trên nền tảng là web (ứng dụng web). Mang lại nhiều lợi ích đáng chú ý nhất là tốc độ. Nó giúp tối ưu hóa quá trình phát triển và triển khai, giảm thiểu thời gian chờ đợi và tăng hiệu suất xử lý dữ liệu, giúp chạy mượt hơn. Ngoài ra website ELearning còn có khả năng Responsive tự động điều chỉnh giao diện và nội dung theo kích thước màn hình của các thiết bị khác nhau phù hợp cho sinh viên nếu có iPad, máy tính bảng hay điện thoại di động. Về mặt tương thích, flatform web hỗ trợ nhiều trình duyệt và hệ điều hành khác nhau, giúp các nhà phát triển không cần lo lắng về việc trang web của họ có hoạt động tốt trên mọi thiết bị hay không. Điều này đặc biệt quan trọng trong bối cảnh người dùng ngày càng đa dạng về thiết bị và nền tảng sử dụng. Cuối cùng, flatform web được hỗ trợ bởi một cộng đồng người dùng rộng lớn và năng động. Cộng đồng này không chỉ đóng góp các công cụ, tài liệu hướng dẫn và plugin hữu ích mà còn cung cấp hỗ trợ và chia sẻ kinh nghiệm, giúp các nhà phát triển giải quyết vấn đề nhanh chóng và hiệu quả hơn. Sự tương tác và hợp tác trong cộng đồng là một yếu tố then chốt giúp flatform web liên tục phát triển và cải thiện.

### 4.2.2. Mua tên miền & chứng chỉ SSL



Hình 4.2.3 Tên miền elearning sau khi mua và xác thực chứng chỉ ssl

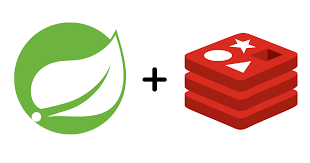
Chúng tôi xin phép đặt tên cho ứng dụng mang tên là “**ELEARNING”** và các tên liên quan được đề cập trong ứng dụng, tài liệu, video demo sản phẩm hoặc tên miền đặt cho ứng dụng đều sẽ nhắc đến cụm từ ELearning này để nhất quán dễ gợi nhớ khi nhắc đến chương trình này.

Do chỉ có nền tảng web là cốt lỗi chính của chương trình (vì ứng dụng web có chứa các service resful như microservce) cần sử dụng tên miền (domain) công khai trên toàn cầu. Mà nền tảng web lại được triển khai dùng chung cho cả 3 đối tượng gồm có cả sinh viên, nhân viên, giáo viên (do ứng dụng sử dụng cơ chế Server Side Render, sẽ nói chi tiết ở mục sau) trong môi trường học vụ. Nên tên miền cần có cụm từ edu đi kèm. Vậy nên chúng tôi quyết định (hoặc dự kiến) đặt tên cho ứng dụng là elearning.edu.vn. Trong đó subdomain cho trang dành cho đối tượng sinh viên là elearning.edu.vn/student ; cho đối tượng nhân viên là elearning.edu.vn/manager; và cho đối tượng giáo viên là erukalearn.edu.vn/teacher

Ngoài ra, để tăng tính xác thực và bảo mật, an toàn cho ứng dụng, chúng tôi đã mua thêm chứng chỉ SSL (hình ổ khóa bảo mật) được tặng kèm khi mua domain .edu từ nhà cung cấp dịch vụ NameCheap.com. Từ đó, ổ khóa bảo mật SSL mang lại lợi ích cho người dùng tin tưởng và còn mang lại một vài lợi cho người phát triển tránh được các vấn để về an ninh mạng, cấu hình tập tinh, data khi triển khai microservice gọi giao tiếp qua nhiều tầng application khác nhau.

Ngoài ra ứng dụng còn sử dụng vài tên miền chạy ngầm với các cổng (port) khác nhau như: elearning.edu.vn:8080 - domain của service Springboot;elearning.edu.vn:8000 – domain của service Apache PHP; Để các ứng dụng có thể giao tiếp frontend và backend với nhau.

### 4.2.3. Thêm kỹ thuật Redis Cache



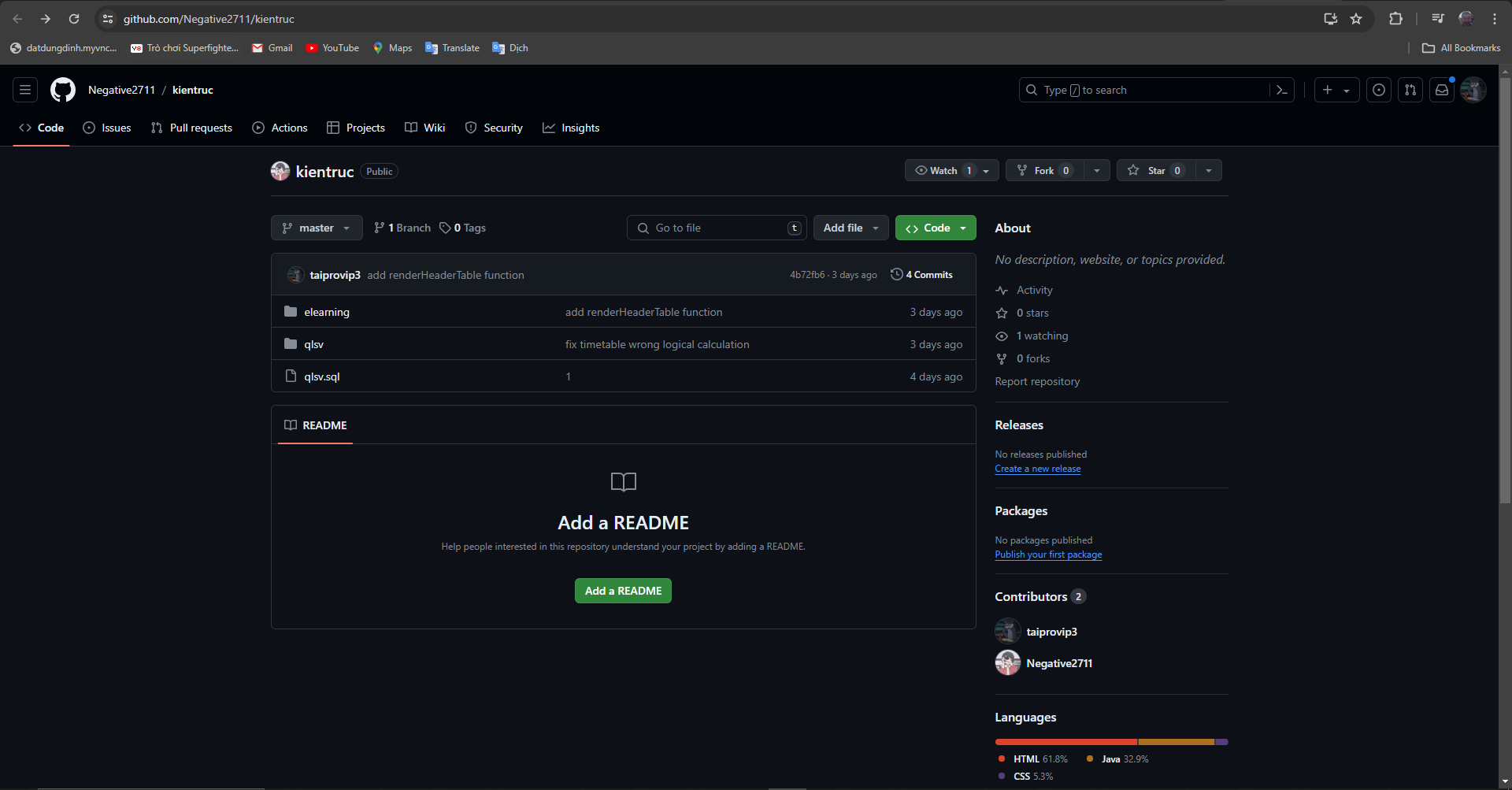
Hình 4.2.5 Kỹ thuật Redis Cache được áp dụng vào hệ thống ELearning để gia tang performance

Redis Cache là một giải pháp mạnh mẽ để tối ưu hóa hiệu suất ứng dụng bằng cách giảm tải truy vấn cơ sở dữ liệu. Redis, một kho lưu trữ dữ liệu trong bộ nhớ, cung cấp tốc độ truy cập nhanh chóng nhờ khả năng lưu trữ dữ liệu dưới dạng cặp khóa-giá trị. Khi tích hợp với Spring Boot, Redis Cache giúp các ứng dụng Java cải thiện thời gian phản hồi và tăng hiệu suất tổng thể. Spring Boot cung cấp các công cụ đơn giản để cấu hình và sử dụng Redis Cache, cho phép các nhà phát triển dễ dàng thêm chức năng bộ nhớ đệm vào ứng dụng của mình. Bằng cách chú thích các phương thức với @Cacheable, dữ liệu được truy xuất sẽ được lưu trong Redis, và khi cần, dữ liệu này sẽ được lấy ra từ bộ nhớ đệm thay vì truy vấn lại từ cơ sở dữ liệu, giúp tiết kiệm tài nguyên và tăng tốc độ xử lý.

## Hiện thực

* + 1. **Github và quản lý source code**

Github repo:



*Hình 4.3.1 Minh chứng cho việc hiện thực tổ chức quản lý source code trên github*

Trong ứng dụng lần này, chúng tôi tổ chức quản lý source code trên github làm việc nhóm như sau:

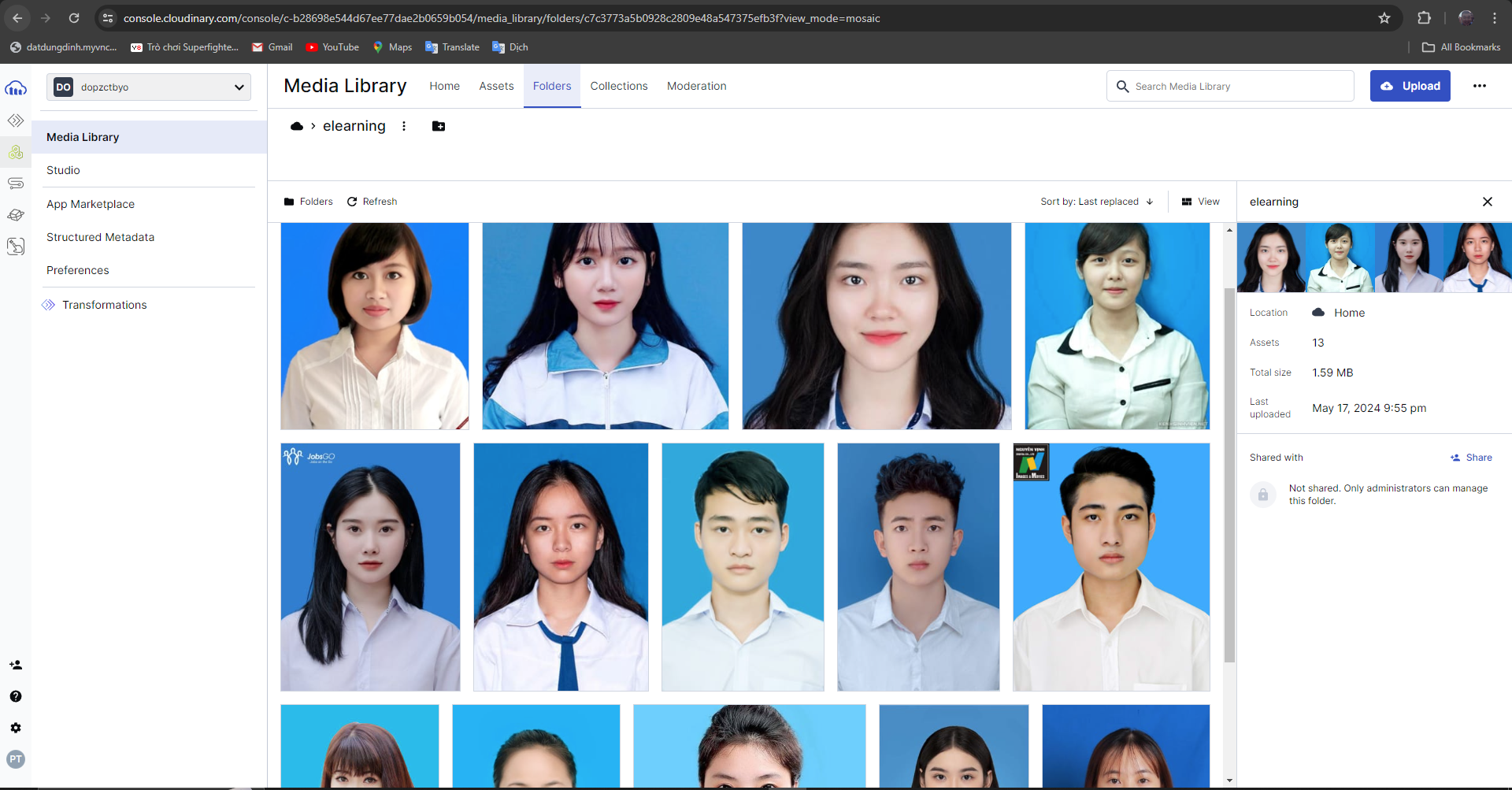
Nhánh làm việc là nhánh tên master, tổ chức source theo kiểu gom tất cả project sẽ thuận tiện cho việc merge code giữa backend và frontend.

Đầu tiên thành viên leader (Long) sẽ init github và push các source cơ bản lên. Thành viên khác (Minh) sẽ pull về và phát triển trên nhánh khác copy từ nhánh main, sau đó sẽ tạo merge request lên nhánh main và chờ review code.

Do việc phát triển dưới local và triển khai lên môi trường thực tế đòi hỏi source code phải thay đổi các biến môi trường và cấu hình rất khó khăn nên chúng tôi chia làm 2 repository để chứa source code cho local và staging master (tạm gọi là source code chạy dưới local và source để deploy).

Link github: <https://github.com/Negative2711/kientruc>

* + 1. **Upload ảnh sinh viên lên Cloudinary**



Thay vì lưu trự ảnh của sinh viên trực (ảnh chụp nền xanh 3x4) tiếp trong database với số lượng lớn. Chúng tôi sẽ lưu trữ các file ảnh trên dịch vụ bên thứ 3 (third party) hỗ trợ đó là Cloudinary (hoàn toàn miễn phí). Việc này sẽ giúp giảm gánh nặng cho phía backend truy vấn & lưu trữ. Từ đó giúp tăng performance hệ thống giảm tải bớt cho phía server.

## 4.4 Một số hình ảnh demo

Công cụ ELearning triển khai trên hai nền tảng là web. Công cụ này sẽ có giao diện người dùng thân thiện và đảm bảo tính ổn định trên các dòng điện thoại khác nhau. Chúng tôi cũng xin giới thiệu vài giao diện cho các chức năng trên ứng dụng trên. Đầu tiên, người dùng đăng nhập vào hệ thống https://elearning.edu.vn[/](http://13.215.185.98:4000/?fbclid=IwAR3PZrRpLUFHS1NWNa8OpGhat6zEbdjClreTghV_75ummK-YL547cv1dxLA)student, tài khoản sinh viên sử dụng là tài khoản do nhà trường cung cấp với tên đăng nhập ứng là mã số sinh viên và mật khẩu (*hình 4.4.1*)



Hình 4.4.1 Giao diện đăng nhập ELearning trên nền tảng web

Ngoài ra chức năng đăng nhập cho phép lưu cookie thông tin đăng nhập và tạo ra 1 khoản session login tiện cho việc sinh viên vào lại trang chủ mà không cần login lại sau mỗi lần thoát khỏi page (thoát khỏi google). Ứng dụng app cũng tương tự; Lưu session cho phiên đăng nhập thông qua AsyncStorage giúp mỗi lần vào app sẽ tự động đăng nhập cho sinh viên. Sau một khoản thời gian (hoặc tắt máy) các session sẽ mất.

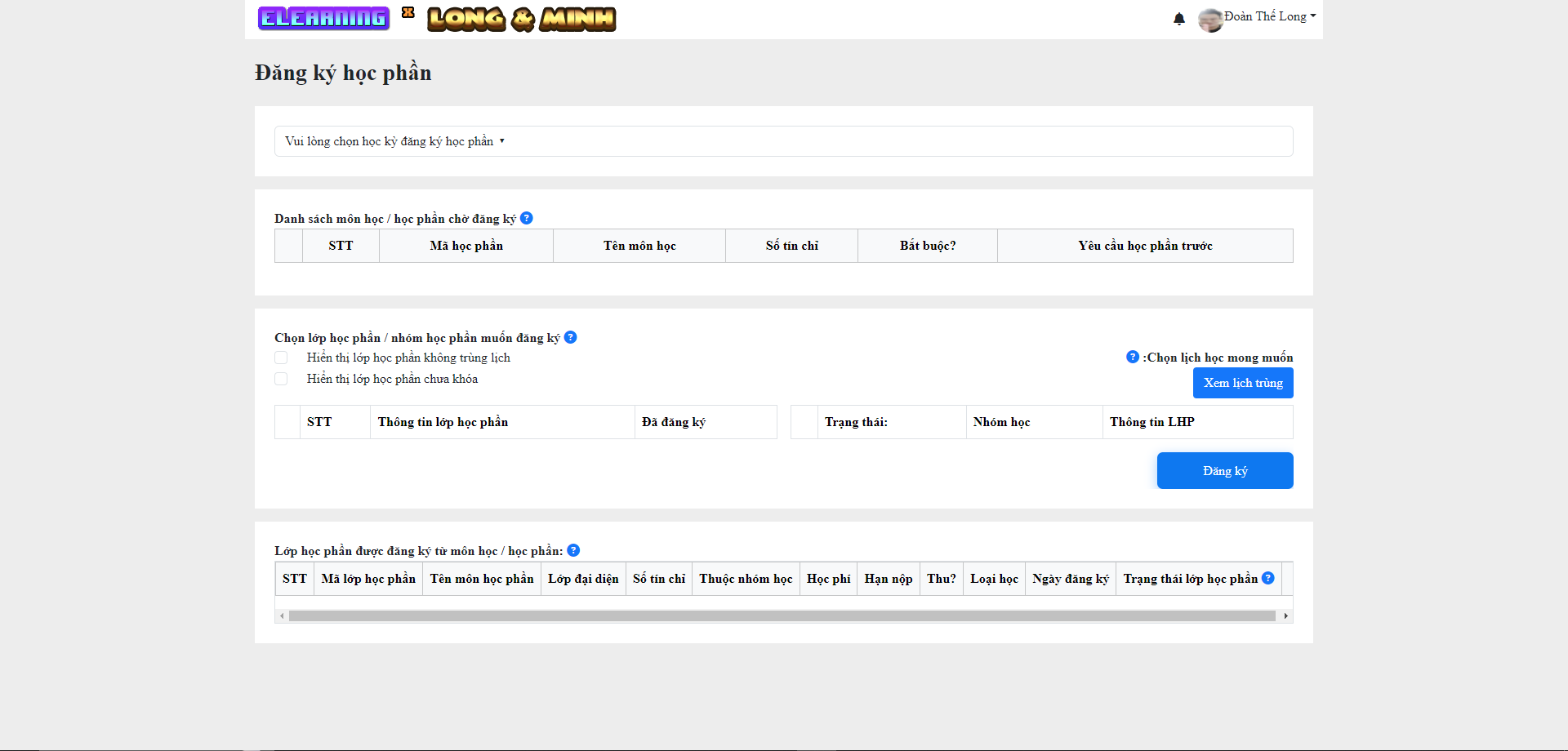


*Hình 4.4.2. Giao diện trang chủ trang sinh viên ELearning System*

Tại giao diện trang chủ người dùng có thể tổng quan nắm bắt được thông tin cá nhân, tiến độ học tập, điểm số các môn vừa đạt được thông qua lớp học phần đang đăng ký học tập và danh mục menu chức năng giúp sinh viên dễ dàng thao tác thông qua bố cục UX rõ ràng chia làm 4 khung như hình. Ngoài ra còn các tiện ích nhỏ bên góc phải như một menu bar để sinh viên thao tác như thông báo, cài đặt, hoặc đăng xuất tài khoản.

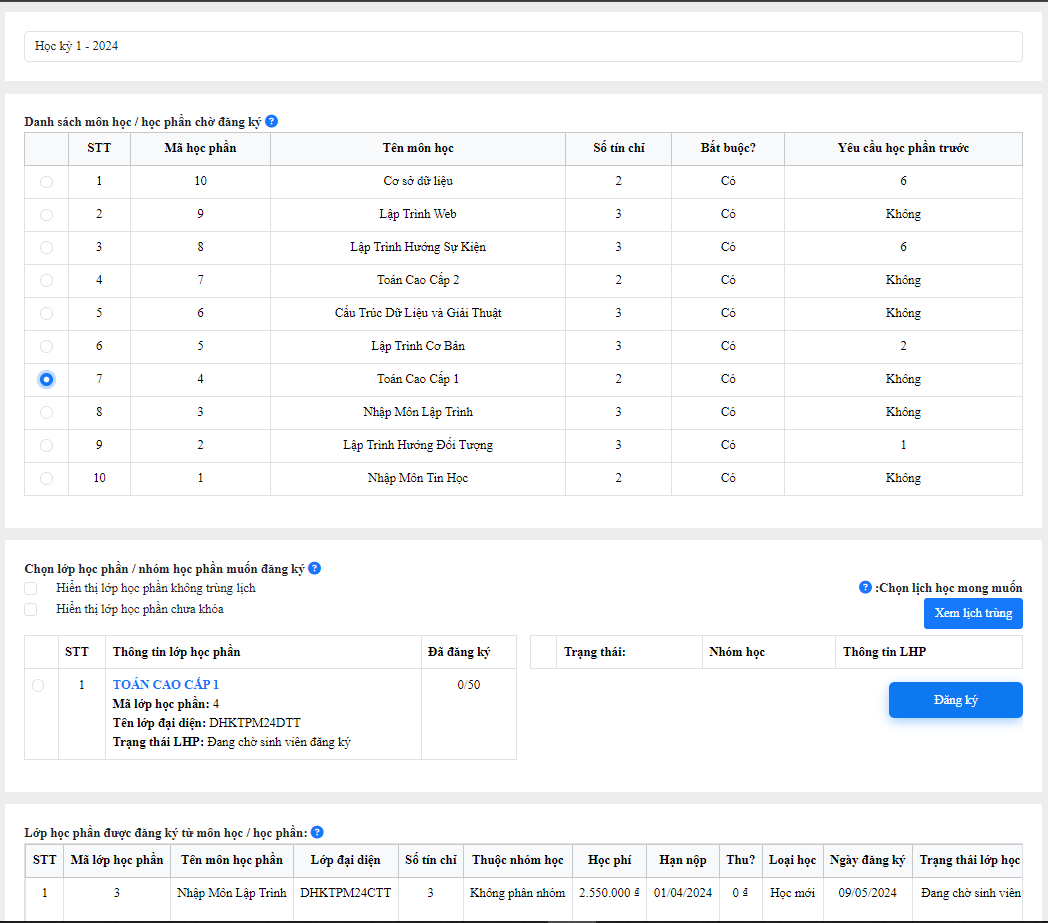
Từ trái sang phải:

* Khung 1 hiển thị thông tin sinh viên, có thể xem chi tiết kiểm tra thông tin
* Khung 2 hiển thị tiến độ học tập của sinh viên theo % tín chỉ hoàn thành.
* Khung 3 hiển thị danh sách môn học / học phần sinh viên đang đăng ký theo học hoặc các môn học, học phần của khóa học trước đó sinh viên đã học hoàn tất. Đồng thời ứng mới mỗi học kỳ chọn trên select box sẽ hiển thị biểu đồ (chart) điểm của sinh viên đó.
* Khung 4 là danh sách các chức năng hỗ trợ cho việc học vụ của sinh viên. Tên chức năng ứng với nhệm vụ của nó bên trong, chúng tôi sẽ đi chi tiết ở các giao diện sau.



*Hình 4.4.3. Giao diện đăng ký học phần của ELearning System*

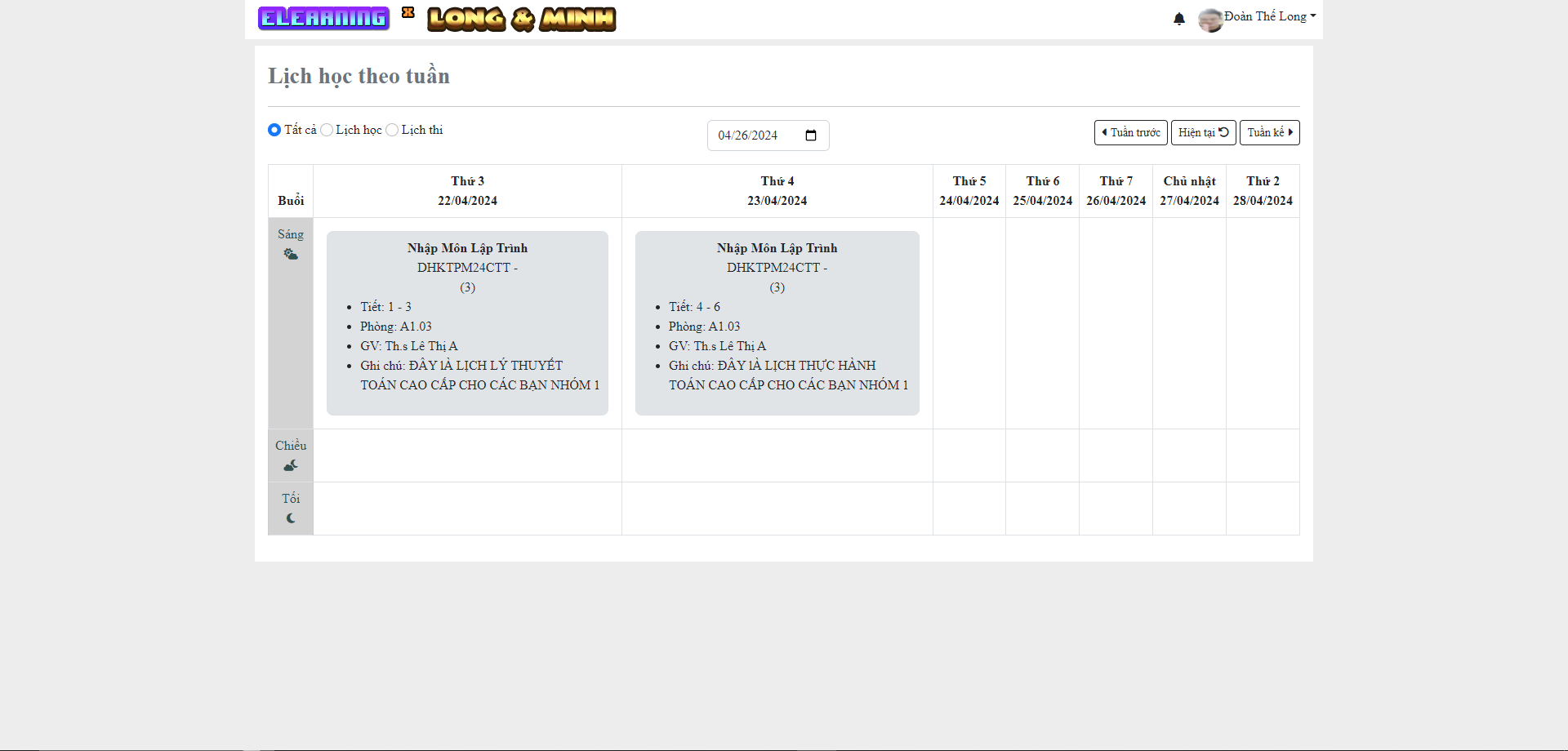
Trong giao diện đăng ký học phần, đầu tiền sinh viên sẽ chọn học kỳ cần đăng ký, học kỳ nào vừa được giáo vụ (manager) mở thì sẽ được đánh tick xanh lá để sinh viên biết và chọn. Các select cho học kỳ cũ sẽ hiển thị các môn học đã mở ở quá khứ và các lớp học phần đã đăng ký tương đương cho học kỳ đó.



*Hình 4.4.4. Giao đăng ký học phần sau khi chọn học kỳ ELearning System*

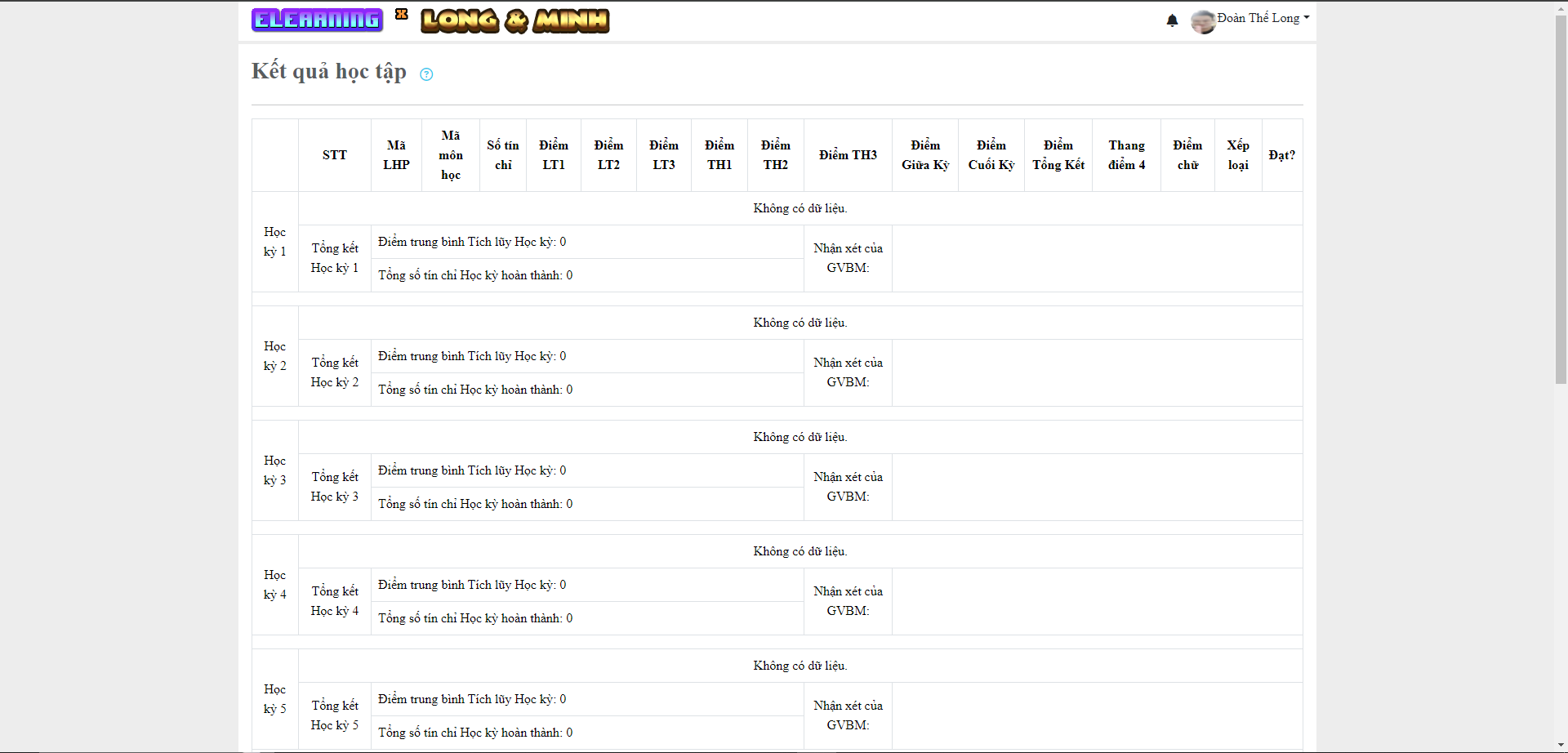
Sau khi chọn học kì sẽ hiển thị các môn học được mở ra cho học kỳ đó, sinh viên sẽ chọn học phần mong muốn và sẽ list ra các lớp học phần được mở cho môn đó (các thông tin chi tiết xem trên *hình 4.4.4*). Và ứng với mối lớp học phần lại có thời khóa biểu riêng. Sinh viên có thể tự do lựa chọn lịch mong muốn và nhấp nút đăng ký.

Ngoài ra, trên giao dện còn hỗ trợ cho sinh viên các lợi ích chức năng khác như là: nút xem lịch trùng, checkbox hiển thị lớp học phần không trùng lịch, hiển thị lớp học phần không khóa,... để có thể kiểm tra xem lớp học phần và nhóm thực hành đã chọn có trùng với những môn học đã đăng ký trước đó hay không. Ứng dụng sẽ hiển thị danh sách các học phần bị trùng lịch học (*hình 4.4.4*). Ở cuối giao diện là danh sách các lớp học phần đã đăng ký trong học kỳ này. Người dùng có thể chọn hủy lớp học phần nếu muốn. Khi hủy, ứng dụng sẽ thông báo thành công hay không và lý do đi kèm nếu thất bại (lớp học phần đã chấp nhận mở lớp, lớp học phần đã khóa…). Người dùng có thể chọn xem lịch học chi tiết của lớp học phần đã đăng ký đó. Hệ thống sẽ hiển thị chi tiết thời khóa biểu.

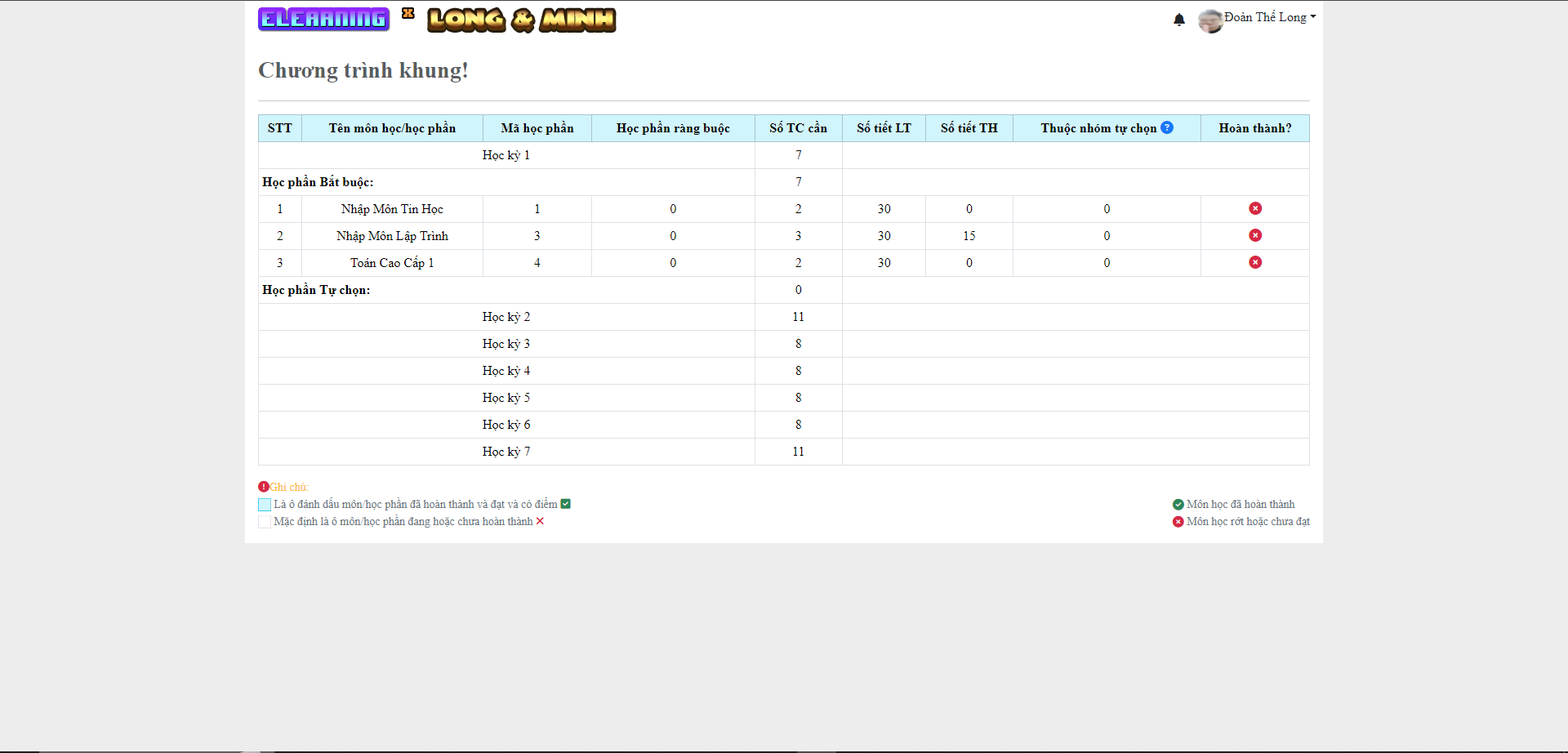


*Hình 4.4.5. Giao diện xem lịch học dành cho sinh viên*

Tính năng xem lịch học theo tuần, có thể xem lịch học ở các tuần quá khứ, hiện tại, tương lai. Lịch nếu là lịch thi thì sẽ có màu nổi bật. Ngày hiện tại sẽ được đánh dấu bằng check màu xanh lá giúp sinh viên dễ dàng nắm bắt ngày đi học. Ngoài ra, sinh viên có thể lọc lịch học và lịch thi, đối với lịch thi thì sẽ nổi bật màu vàng ▄ .

*Hình 4.4.6. Giao diện xem kết quả học tập (xem điểm) dành cho sinh viên.*

Tính năng này giúp hiển thị tất cả điểm của tất cả môn trong qua các học kỳ của sinh viên. Mỗi một sinh viên sẽ có số học kỳ tương ứng với số năm mà sinh viên đó học tại trường \* (nhân x) với 2. Những môn học phần nào có điểm sẽ được hệ thống cập nhật ngay lập tức. Mỗi học kỳ khi kết thúc và hoàn thành tất cả các môn trong chương trình khung thì sẽ được tính điểm Tổng Kết theo thang 4 và 10. Ngoài ra, ở hệ thống này, chúng tôi còn bổ sung them khả năng nhận xét của giáo viên. Giúp sinh viên nắm rõ tình trạng cũng như lời phê hoặc khen của giáo viên tổng hợp trong năm đó.



*Hình 4.4.7. Giao diện xem chương trình khung theo ngành của sinh viên.*

Tính năng này giúp hiển thị tất cả môn học nằm trong chương trình khung đóng vai trò như một nấc thang để sinh viên có thể khái quát tổng quan lại tất cả môn học mà bản thân sẽ cần phải hoàn thành trong quá trình học tập. Chương trình khung bao gồm tất cả các môn học thuộc chuyên ngành riêng của sinh viên (sinh viên mỗi ngành khác nhau sẽ có chương trình khung khác nhau). Những môn học/học phần đã hoàn thành thì sẽ được tô đậm và đánh dấu pass. Trong chương trình khung còn có hiển thị những môn học tự chọn để sinh viên có thể tìm hiểu và phân tích trước khi ra quyết định đăng ký học môn tự chọn đó để đạt tín chỉ.

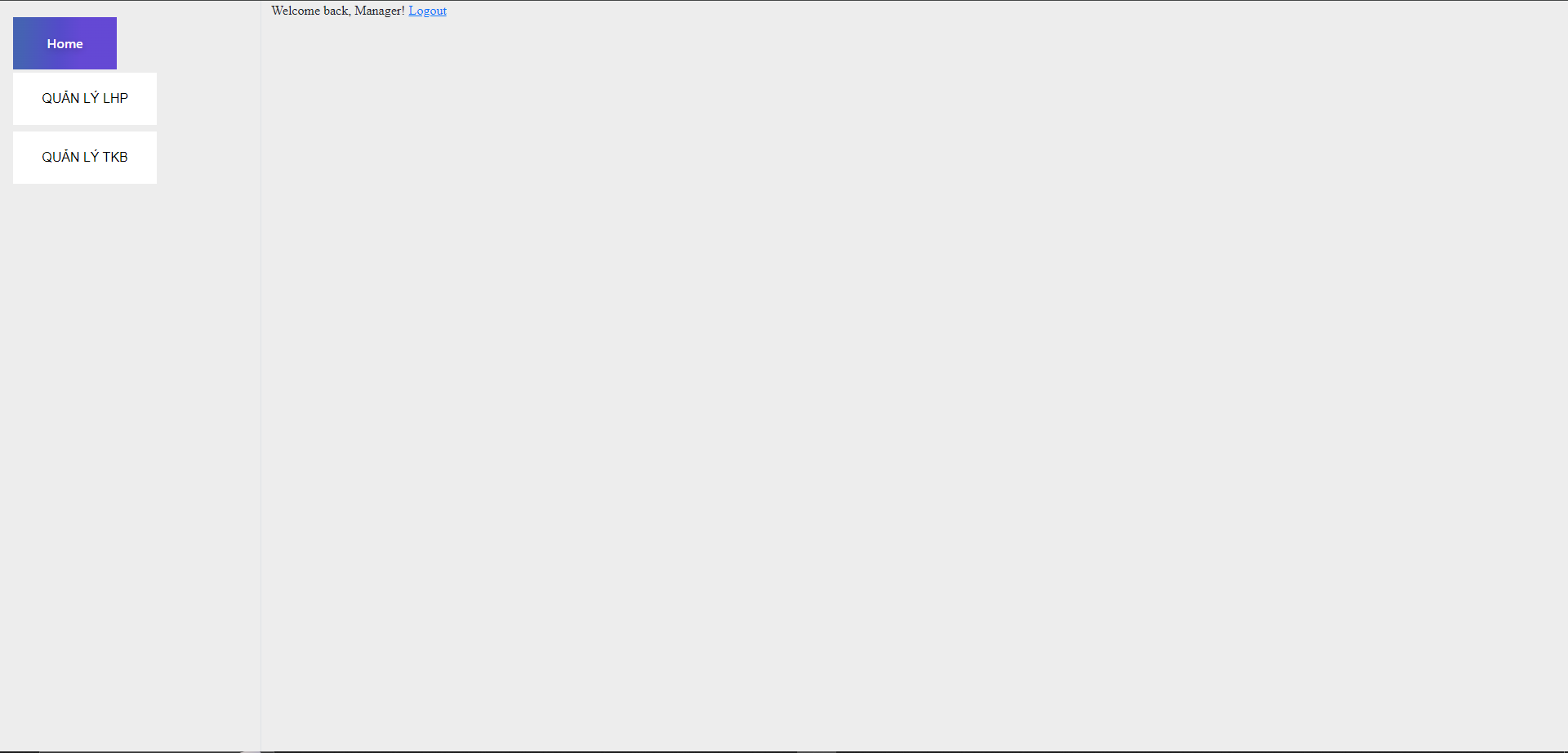
Về phân hệ cho **Manager,** chúng tôi đã triển khai ứng dụng web và đã triển khai trên domain https://elearning.edu.vn[/](http://13.215.185.98:4000/?fbclid=IwAR3PZrRpLUFHS1NWNa8OpGhat6zEbdjClreTghV_75ummK-YL547cv1dxLA)employee. Đây là và tên miền do sẽ được map với server backend cung cấp đã nói trước đó.

Đầu tiên manager sẽ đăng nhập hệ thống:



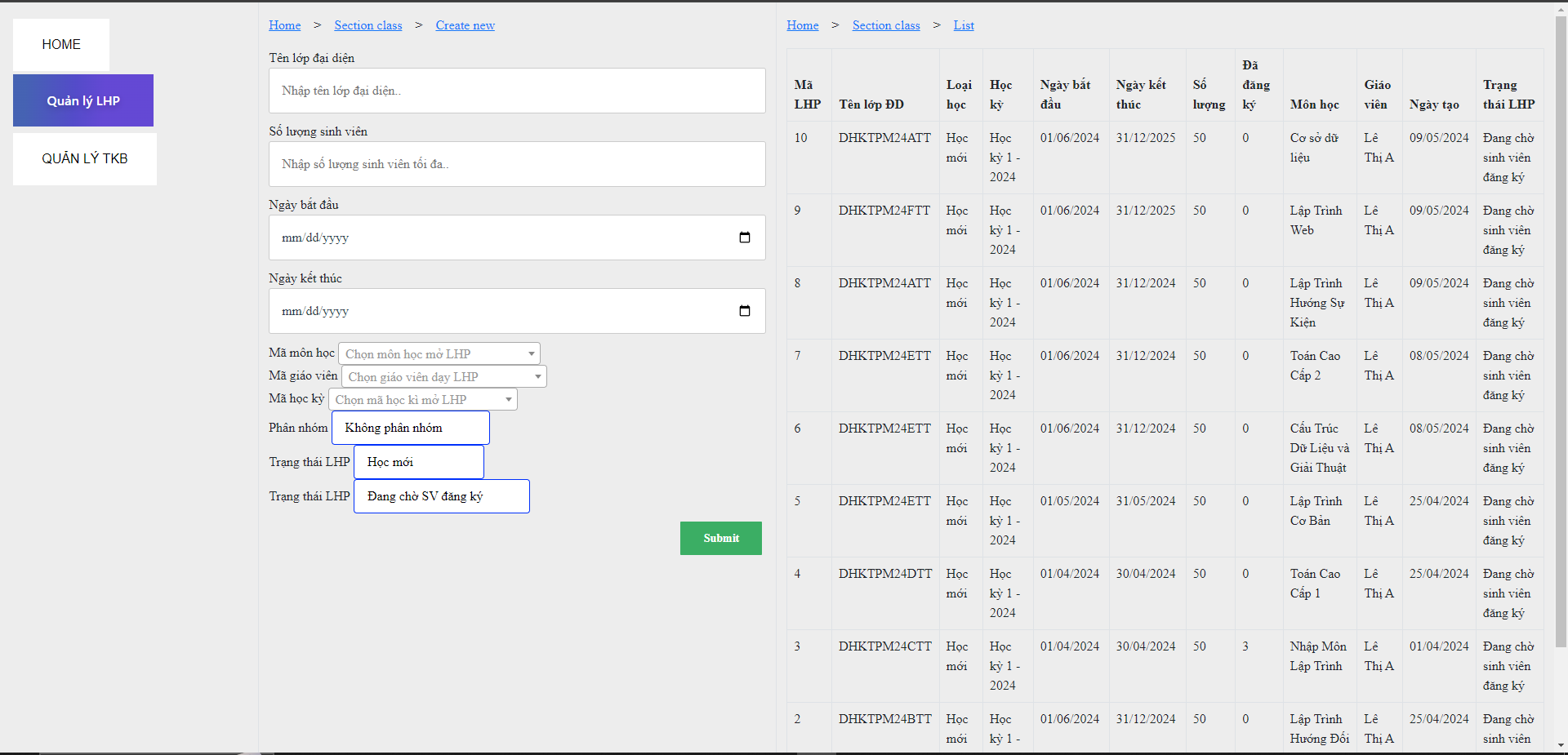
Hình 4.4.8 Trang đăng nhập manager ELearning System

Các chức năng như save cookie, session cho user đăng nhập chúng tôi đều đảm bảo thực thi được trên cả 3 đối tượng `Student`, `Manager`, `Teacher`. Cũng như sinh viên nên sẽ không nói lại phần đăng nhập.

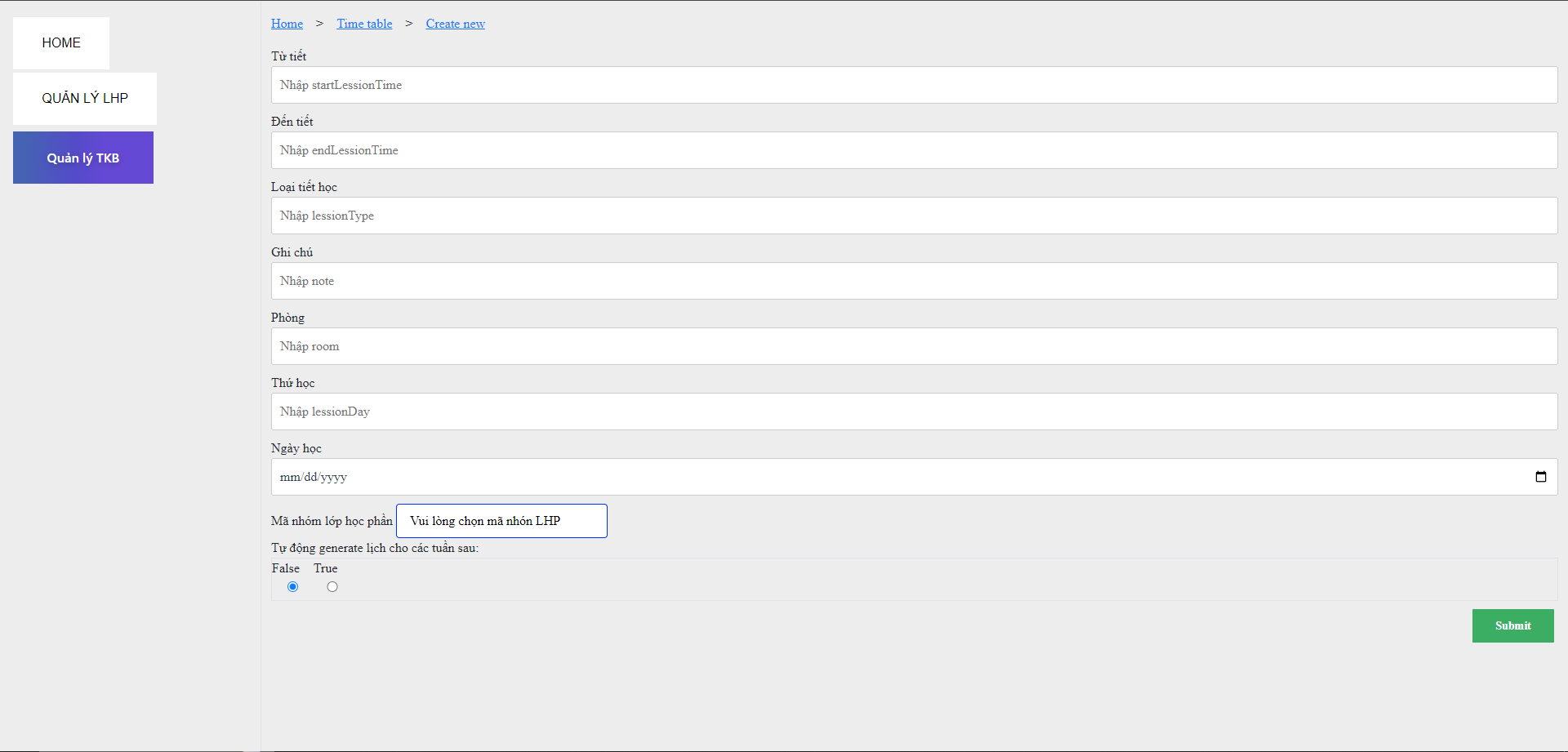


*Hình 4.4.9 Trang chủ của ứng dụng web cho Manager*

Sau khi đăng nhập, manager sẽ được điều hướng đến trang chủ (hình 4.4.9). Ở trang manager, đối với thanh chức năng bên trái, hiện nay chúng tôi triển khai 3 bộ chức năng bao gồm quản lý sinh viên, quản lý khoa, quản lý chuyên ngành, quản lý môn học, quản lý học kì, quản lý học phần, quản lý lớp học phần,... Mỗi bộ chức năng sẽ bao gồm 4-7 chức năng chính là xem, thêm, xóa, chỉnh sửa, import, export excel (nếu có).



*Hình 4.4.10. Trang quản lý lớp học phần ELearning System*



Hình 4.4.11. Trang quản lý thời khóa biểu

Trang quản lý LHP và TKB thường dùng để phục vụ cho việc mở lớp học phần qua theo các học kỳ trong năm để sinh viên tiến hành đăng ký học phần tương ứng theo học phần mở (hình 4.4.10 và hình 4.4.11).

# CHƯƠNG 5: KẾT QUẢ ĐẠT ĐƯỢC VÀ HƯỚNG PHÁT TRIỂN

## Kết quả đạt được:

Hiện nay, chúng tôi đã bước đầu tạo ra được 1 ứng dụng quản lý học tập cá nhân cho sinh viên, tạo ra ứng dụng web cho người quản lý của nhà trường. Chúng tôi đã giải quyết thành công các nhu cầu học vụ cơ bản của hệ thống ELearning System

Chúng tôi đã phân chia hệ thống thành hai phần chính là Front-end và Back-end, hai thành phần này giúp chia sẻ gánh nặng cho nhau. Front-end chủ yếu thực hiện kết xuất giao diện và tương tác với người dùng, trong khi đó Back-end phụ trách chủ yếu việc xử lý dữ liệu. Xu hướng hiện nay và tạo cơ sở để chúng tôi triển khai module thông báo theo thời gian thực sắp tới. Ở phía website cho người quản lý chúng tôi đã triển khai cơ chế Server render Client-side, đây là cơ chế mà chúng tôi đã giới thiệu ở phần đầu của tài liệu. Cơ chế kết xuất này khắc phục được những yếu điểm của cơ chế Client-side Rendering giúp tăng tốc độ tải trang, cải thiện trải nghiệm người dùng và bảo mật cao, các tính chất đó phù hợp với các Website dành cho người quản lý và cả sinh viên, giáo viên.

Về phía Back-end, chúng tôi áp dụng thành công kiến trúc Base Service, clean source code.

Chúng tôi cũng đã giải quyết được vấn đề về chứng chỉ SSL cho domain của website dành cho quản trị viên cũng như domain của server Back-end. Đây là chứng chỉ cực kì quan trọng bởi lẽ thiếu nó thì khả năng bảo mật của ứng dụng sẽ cực kì kém.

Về sử lý logic cho thanh toán các giao dịch điện tử, chúng tôi đảm bảo trong quá trình code có áp dụng các thuật toán giải quyết những case bảo mật, tránh gây bugs, ví dụ mẫu là áp dụng token xác thực người dùng cho từng giao dịch được tạo ra. Sử lý các if else chuyển trang hợp lệ,..vv và đặc biệt là tạo ra khả năng sử lý lỗi trong trường hợp thanh toán giao dịch bị lỗi do server chết hoặc client mất mạng (chúng tôi phát triển ra chức năng xác thực lại giao dịch người dùng chi tiết trong phần demo usecase sản phẩm). Nhằm tăng tính sẵn dùng của hệ thống.

## Hạn chế của đồ án

Chức năng trên hệ thống web dành cho nhà quản trị vẫn còn hạn chế, chưa đáp ứng hết các chức năng hỗ trợ. Phần ứng dụng trên thiết bị di động cũng chưa được triển khai trên các store để hỗ trợ cho sinh viên sử dụng. Trong tương tai, chúng tôi sẽ dành tài nguyên và nhân lực để giải quyết các hạn chế trên.

Đề tài vẫn còn một số hạn chế là quá trình chúng tôi thử nghiệm chỉ với số lượng truy xuất vừa và nhỏ. Hệ thống cần thêm việc DOS server, tạo ra lượng truy cập ảo (có thể lên tới 100.000 request) để tiến hành monitoring, maintain nhằm đảm bảo tính mở rộng của hệ thống và khẳng định khả năng Scale và Load Balancing thật sự hiệu quả khi mà áp dụng cloud computing thay vì physical server.

Các chức năng dành cho app di động (mobile) chưa thực sự tối ưu bằng ứng dụng web và vẫn còn lag app hoặc render chậm do việc dev quản lý data bằng các hook (memo, state,...) chưa hiệu quả do trình độ code chưa cao.

Ứng dụng chỉ mới tập trung chính vào đối tượng người dùng sinh viên, chưa thật sự đủ tính năng cho người quản lý và giáo viên. Vì trên thực tế việc học vụ đòi hỏi rất nhiều case chức năng khác không chỉ nhập điểm, xem lịch dạy, mà còn là chấm công, đăng ký lịch dạy học bù, lịch thi... Trang nhân viên hiện tại vẫn còn nhập thủ công và nhớ khá nhiều chưa thực sự tối ưu UX/UI 10 quy tắc của nielsen.

## Hướng phát triển:

Hiện nay, vì là sinh viên nên điều kiện kinh tế còn hạn hẹp cũng như thời gian làm quen với Cloud Computing chưa nhiều nên việc sử dụng các dịch vụ của Amazon Web Services (AWS) rất khó khăn. Chúng tôi chưa thể triển khai kiến trúc hệ thống theo đúng mong muốn được đặt ra từ đầu. Tương lai, chúng tôi muốn triển khai kiến trúc hệ thống theo kiến trúc Cloud Computing. Cụ thể chúng tôi muốn tách nhỏ hệ thống Back-end thành nhiều dịch vụ khác nhau chạy trên các CPU riêng biệt đúng với tinh thần của kiến trúc Microservices. Điều này giúp các dịch vụ có tính độc lập cao, vừa giúp khả năng mở rộng tốt vừa tăng tính an toàn cho hệ thống, giả sử một dịch vụ bị lỗi thì các dịch vụ khác vẫn có khả năng hoạt động bình thường. Chúng tôi sẽ tách thành các dịch vụ riêng biệt như Authentication Service (phục vụ việc đăng nhập, phân quyền chức năng), Sign Up Modules Service (phục vụ việc đăng ký học phần),… Giả sử sau này chúng tôi muốn đầu tư và phát triển cho hệ thống dịch vụ đăng ký học phần thì cũng không cần phải lo lắng các dịch vụ khác sẽ bị ảnh hưởng.

Vì điều kiện thiết bị khó khăn, nên chúng tôi vẫn chưa thể thử nghiệm được ứng dụng di động trên nền tảng iOS. Trong tương lai, chúng tôi sẽ thử nghiệm ứng dụng trên điện thoại iPhone. Chúng tôi cũng có kế hoạch đưa ứng dụng di động ELearning System lên cả CH Play và Appstore.

Chúng tôi sẽ triển khai chức năng thông báo thông báo trên ứng dụng di động. Khi lịch thi của một môn học được cập nhật hoặc có các thông báo của nhà trường, cảnh báo học vụ,… thì sinh viên sẽ được thông tin kịp thời.

Về phần hệ chức năng cho người quản lý, thời gian sắp tới, chúng tôi hướng đến việc phân quyền chức năng chặt chẽ hơn nữa. Cụ thể chung tôi sẽ phân quyền chức năng theo khoa, quản trị viên của khoa nào thì chỉ có quyền chỉnh sửa những thông tin thuộc quyền quản lý của khoa đó. Và trong mỗi khoa, chúng tôi hướng đến việc phân cấp chức năng kĩ lưỡng hơn nữa. Ví dụ, chỉ có trưởng khoa hoặc trưởng bộ môn mới được quyền chỉnh sửa thông tin môn học.

Tương lai, chúng tôi cũng muốn phát triển phân hệ cho giảng viên.

Ngoài ra, bảo mật cũng là yếu tố cực kì quan trọng trong một hệ thống. Sắp tới, chúng tôi sẽ tập trung nhiều hơn nữa cho vấn đề này.

# TÀI LIỆU THAM KHẢO

## Các tài liệu Tiếng Việt

## Các tài liệu Tiếng Anh

[1] Liu, J., Ahmed, E., Shiraz, M., Gani, A., Buyya, R., & Qureshi, A. (2015). Application partitioning algorithms in mobile cloud computing: Taxonomy, review and future directions. Journal of Network and Computer Applications, 48, 99-117.

[2] Kishore, P., & Mahendra, B. M. (2020). Evolution of Client-Side Rendering over Server-Side Rendering. Recent Trends in Information Technology and its Application, 3(2).

[3] Balalaie, A., Heydarnoori, A., & Jamshidi, P. (2016). Microservices architecture enables devops: Migration to a cloud-native architecture. IEEE Software, 33(3), 42-52.

[4] Philips, J. (2019). Workflow automation using Docker Swarm and GitLab CI at Flanders Institute for Biomechanical Experimentation. In NL-RSE19 conference on Research Software Engineering, Date: 2019/11/20-2019/11/20, Location: Amsterdam.

[7] Walls, C. (2016). Spring Boot in action. Manning Publications.

[8] Danielsson, W. (2016). React Native application development. Linköpings universitet, Swedia, 10(4).

[9] Rawat, P., & Mahajan, A. N. (2018) ReactJS: A Modern Web Development Framework.

[12] Correia, R. C. M. (2020). Analysis of Node. js Application Performance Using MongoDB Drivers. Information Technology and Systems: Proceedings of ICITS 2020, 1137, 213.

## Các tài liệu từ Internet

[5] <https://java.com/en/download/help/whatis_java.html>, accessed 03/4/2023

[6] <https://www.javatpoint.com/what-are-the-uses-of-javascript>, accessed 03/4/2023

[10] https://ordergroup.co/blog/best-web-frameworks-and-languages-to-use-in-2022/ acccessed 16/4/2023

[11] [https://scoutapm.com/blog/nodejs-architecture-and-12-best-practices-for-nodejs-development acccessed 03/4/2021 acccessed 05/4/2021](https://scoutapm.com/blog/nodejs-architecture-and-12-best-practices-for-nodejs-development%20acccessed%2003/4/2021%20acccessed%2005/4/2021)

[13] <https://aws.amazon.com/vi/getting-started/hands-on> accessed 05/4/2023

[14] <https://docs.aws.amazon.com/whitepapers/latest/introduction-devops-aws/infrastructure-as-code.html> acccessed 05/4/2023

[15] <https://www.terraform.io/docs/cloud/> acccessed 05/4/2023

[16] <https://www.suntechnologies.com/service/infrastructure-as-code> acccessed 22/5/2023

[17] <https://dzone.com/articles/the-devops-toolchain>, acccessed 22/5/2023

[18] Cách tính điểm Trung Bình Chung môn học - Đại học Công nghiệp TP Hồ Chí Minh <https://giusetrantoan.wordpress.com/2017/06/29/cach-tinh-diem-tbc-tich-luy-iuh/?fbclid=IwAR1cMSfX2V-kaCUjof1QT50J0Q3sqBPGDAnpmlgnrnTCe1_0XXciqqUmtPg> acccessed 18/02/2023

[19] Setting up an HTTPS ec2 Nodejs Server and Connecting to Frontend <https://www.youtube.com/watch?v=q-XEGbipOVw> accessed 23/04/2023

[20] Deploy Node JS và MySQL lên AWS-EC2 instance (VPS) trong 8 phút <https://www.youtube.com/playlist?list=PLiyt_MVS2nFXmjaoWdEDTMSDPuw3U8CU4> accessed 26/04/2023

[21] Đưa ứng dụng spring boot lên AWS 100% thành công <https://www.youtube.com/watch?v=v77hbQ8Mg5Y&list=PLiyt_MVS2nFXmjaoWdEDTMSDPuw3U8CU4&index=1&t=85s> accessed 26/04/2023

[22] PHP CSS Mysqli | Tích hợp thanh toán MOMO <https://www.youtube.com/watch?v=oyA7KfcVM1k&t=34s> accessed 03/01/2023

[23] Demo tích hợp thanh toán trực tuyến Paypal | Setup Payments With Node.js <https://www.youtube.com/watch?v=Z-4MaR8tP4Y&t=254s> accessed 19/03/2023

# PHỤ LỤC