

**NGUYỄN VĂN CHINH ĐỒ ÁN TỐT NGHIỆP HÀ NỘI, NĂM 2025**

|  |  |
| --- | --- |
| **bỘ GIÁO DỤC VÀ ĐÀO TẠO** | **BỘ NÔNG NGHIỆP VÀ PTNT** |

**TRƯỜNG ĐẠI HỌC THỦY LỢI**



NGUYỄN VĂN CHINH

**XÂY DỰNG HỆ THỐNG ĐĂNG KÝ HỌC TRƯỜNG ĐẠI HỌC tHỦY lỢI**

ĐỒ ÁN TỐT NGHIỆP

HÀ NỘI, NĂM 2025

|  |  |
| --- | --- |
| **bỘ GIÁO DỤC VÀ ĐÀO TẠO** | **BỘ NÔNG NGHIỆP VÀ PTNT** |

**TRƯỜNG ĐẠI HỌC THỦY LỢI**

NGUYỄN VĂN CHINH

**XÂY DỰNG HỆ THỐNG ĐĂNG KÝ HỌC TRƯỜNG ĐẠI HỌC tHỦY lỢI**

|  |  |
| --- | --- |
| Ngành : | Công nghệ thông tin |
| Mã số: | 7480201 |

|  |  |
| --- | --- |
| NGƯỜI HƯỚNG DẪN | 1. ThS. Trương Xuân Nam |
|  |  |

HÀ NỘI, NĂM 2025

|  |  |
| --- | --- |
| Logo-WRU | **CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**  **Độc lập - Tự do - Hạnh phúc**  ----------★----------  **NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP** |

Họ tên sinh viên: Nguyễn Văn Chinh Hệ đào tạo: Chính quy

Lớp: 63CNTT-NB Ngành: Công nghệ thông tin

Khoa: Công nghệ thông tin

**1- TÊN ĐỀ TÀI:**

**XÂY DỰNG HỆ THỐNG ĐĂNG KÝ HỌC TRƯỜNG ĐẠI HỌC THỦY LỢI**

**2- CÁC TÀI LIỆU CƠ BẢN:**

* .. a. V. S. p. t. Microsoft Developer Division, “.NET Microservices: Architecture for Containerized .NET Applications,” Microsoft, 22 March 2023. [Trực tuyến]. Available: https://learn.microsoft.com/en-us/dotnet/architecture/microservices.
* M. O. Source, “Quick Start - React,” [Trực tuyến]. Available: https://react.dev/learn.
* C. Patterson, “MassTranssit,” [Trực tuyến]. Available: <https://masstransit.io/documentation/concepts>.

**3- NỘI DUNG CÁC PHẦN THUYẾT MINH VÀ TÍNH TOÁN:**

|  |  |
| --- | --- |
| **Nội dung thuyết minh và tính toán** | **Tỷ lệ %** |
| CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI | 20 |
| CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ | 35 |
| CHƯƠNG 3: TRIỂN KHAI | 35 |
| CHƯƠNG 4: HẠN CHẾ VÀ PHƯƠNG HƯỚNG PHÁT TRIỂN | 10 |

**4- GIÁO VIÊN HƯỚNG DẪN TỪNG PHẦN**

Giáo viên hướng dẫn toàn bộ đồ án: ThS.Trương Xuân Nam

**5- NGÀY GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP**

Ngày ……tháng ……năm 2025

|  |  |
| --- | --- |
| **Trưởng Bộ môn**  *(Ký và ghi rõ Họ tên)* | **Giáo viên hướng dẫn chính**  *(Ký và ghi rõ Họ tên)* |

Nhiệm vụ Đồ án tốt nghiệp đã được Hội đồng thi tốt nghiệp của Khoa thông qua

Ngày……tháng……năm 2025 **Chủ tịch Hội đồng**

*(Ký và ghi rõ Họ tên)*

Sinh viên đã hoàn thành và nộp bản Đồ án tốt nghiệp cho Hội đồng thi ngày……tháng……năm 2025

**Sinh viên làm Đồ án tốt nghiệp**

*(Ký và ghi rõ Họ tên)*

|  |  |
| --- | --- |
| logo | TRƯỜNG ĐẠI HỌC THỦY LỢI  **KHOA CÔNG NGHỆ THÔNG TIN**  BẢN TÓM TẮT ĐỀ CƯƠNG ĐỒ ÁN TỐT NGHIỆP |

**Tên đề tài: XÂY DỰNG HỆ THỐNG ĐĂNG KÝ HỌC TRƯỜNG ĐẠI HỌC THỦY LỢI**

*Sinh viên thực hiện*: Nguyễn Văn Chinh

*Lớp*: 63CNTT-NB

*Mã sinh viên:* 2151062726

*Số điện thoại:* 0368849143

*Email:* 63cntt.nvchinh@aptech.vn

*Giáo viên hướng dẫn*: ThS. Trương Xuân Nam

**TÓM TẮT ĐỀ TÀI**

Dự án Hệ thống Điểm danh và Hành chính Một cửa được phát triển nhằm tạo liên kết mạnh mẽ giữa sinh viên và Phòng Công tác Chính trị, Giảng viên và Sinh viên. Mục tiêu chính của dự án là cung cấp một cơ sở hạ tầng cho việc điểm danh thuận tiện trong mỗi buổi học và giúp sinh viên thực hiện các thủ tục hành chính một cách dễ dàng và nhanh chóng. Đồng thời, dự án nhằm tối ưu hóa quy trình tiếp nhận, giải quyết, và thông báo kết quả xử lý các thủ tục hành chính cho sinh viên từ phía Phòng Công tác Chính trị và Sinh viên.

Đối với việc điểm danh, hệ thống cho phép sinh viên tự giác thực hiện quá trình này thông qua việc quét mã QR code. Điều này giúp tăng cường sự tự quản lý và tự chủ của sinh viên trong quá trình theo dõi sự tham gia vào các buổi học.

Trong khi đó, chức năng Hành chính Một cửa trên ứng dụng cho phép sinh viên thực hiện các thủ tục hành chính như xin cấp lại thẻ sinh viên, yêu cầu giấy xác nhận sinh viên, và các yêu cầu khác. Phòng Công tác Chính trị và Sinh viên sẽ tiếp nhận và xử lý các yêu cầu này một cách hiệu quả và trong thời gian ngắn nhất có thể. Kết quả sau cùng sẽ được thông báo cho sinh viên thông qua ứng dụng, đồng thời cũng cung cấp thông tin về thời gian và địa điểm để sinh viên đến nhận kết quả.

**CÁC MỤC TIÊU CHÍNH**

Tìm hiểu kiến thức cơ bản về Microservices

Xây dựng website dành cho giảng viên để có thể sử dụng hệ thống điểm danh

Xây dựng website dành cho Phòng Chính trị và Công tác Sinh viên để có thể sử dụng hệ thống hành chính một cửa

Xây dựng hệ thống API để sinh viên có thể điểm danh và thao tác các dịch vụ hành chính một cửa

**KẾT QUẢ DỰ KIẾN**

Lý thuyết:

Nắm vững các kiến thức về các công nghệ được sử dụng .

Các ngôn ngữ và công nghệ dự kiến sử dụng:

* + Cấu trúc: Microservices.
  + Cơ sở dữ liệu: MSSQL Server, MongoDB.
  + Thư viện lập trình backend: .Net.
  + Thư viện xây dựng giao diện người dùng: Mui, React.
  + Khác: Docker, RabbitMQ, MassTransit, …

Sản phẩm đạt được:

Xây dựng thành công website dành cho giảng viên để có thể sử dụng hệ thống điểm danh.

Xây dựng thành công website dành cho Phòng Chính trị và Công tác Sinh viên để có thể sử dụng hệ thống hành chính một cửa.

Xây dựng thành công hệ thống API để sinh viên có thể điểm danh và thao tác các dịch vụ hành chính một cửa.

**TÀI LIỆU THAM KHẢO**

**LỜI CAM ĐOAN**

Em xin cam đoan đây là Đồ án tốt nghiệp của bản thân em. Các kết quả trong Đồ án tốt nghiệp này là trung thực, và không sao chép từ bất kỳ một nguồn nào và dưới bất kỳ hình thức nào. Việc tham khảo các nguồn tài liệu (nếu có) đã được thực hiện trích dẫn và ghi nguồn tài liệu tham khảo đúng quy định.

|  |  |
| --- | --- |
|  | **Tác giả ĐATN**  *Chữ ký*  **Phạm Minh Tiến** |
|  |  |

**LỜI CÁM ƠN**

Em xin trân trọng cám ơn ThS. Trương Xuân Nam đã tận tình dẫn dắt, hỗ trợ, định hướng từ những ngày đầu tiên thực hiện Đồ án tốt nghiệp. Đồ án tốt nghiệp được thực hiện trong khoảng thời gian ngắn, nhưng thầy đã giúp đỡ em rất nhiều trong quá trình làm đồ án và trước đó.

Em xin chân thành cám ơn trường Đại học Thủy Lợi nói chung và khoa Công nghệ thông tin nói riêng, đã luôn tạo điều kiện tốt nhất về cả cơ sở vật chất và giảng dạy, hỗ trợ sinh viên một cách tối đa. Từ đó, sinh viên có cơ hội học tập, phấn đấu, trau dồi kiến thức trên trường lớp lẫn thực hành trong thực tiễn nhằm tạo ra những con người có ích cho đất nước, xã hội trong tương lai.

**Em xin chân thành cảm ơn rất nhiều!**

# CHƯƠNG 1 TỔNG QUAN ĐỀ TÀI

## 1.1 Thực trạng

Trong bối cảnh chuyển đổi số đang diễn ra mạnh mẽ trong lĩnh vực giáo dục, việc ứng dụng công nghệ thông tin vào công tác quản lý đào tạo là một xu hướng tất yếu. Tuy nhiên, tại nhiều trường đại học ở Việt Nam, trong đó có Trường Đại học Thủy Lợi, quá trình đăng ký học phần – một mắt xích quan trọng trong chu trình đào tạo – vẫn còn nhiều tồn tại, gây ảnh hưởng không nhỏ đến trải nghiệm học tập của sinh viên cũng như hiệu quả quản lý của nhà trường.

Hằng năm, vào mỗi đợt đăng ký học phần, hàng nghìn sinh viên cùng lúc truy cập hệ thống nhằm lựa chọn các môn học, lớp học phù hợp với kế hoạch học tập cá nhân. Tuy nhiên, do hệ thống hiện tại còn hạn chế về hiệu năng và khả năng mở rộng, hiện tượng nghẽn mạng, treo hệ thống, mất kết nối hoặc xử lý chậm thường xuyên xảy ra. Điều này không chỉ gây bức xúc, hoang mang cho sinh viên mà còn tạo áp lực rất lớn cho đội ngũ kỹ thuật và phòng Đào tạo của nhà trường.

Ngoài ra, giao diện hệ thống còn thiếu thân thiện, khó thao tác đối với sinh viên mới, không cung cấp đầy đủ thông tin như: số lượng sinh viên còn có thể đăng ký trong lớp, lịch học chi tiết, cảnh báo trùng lịch, hay gợi ý đăng ký tối ưu. Các chức năng quan trọng như thay đổi nguyện vọng, huỷ học phần, đăng ký lại, hoặc thống kê thời khóa biểu cá nhân thường được thực hiện một cách thủ công hoặc qua nhiều bước phức tạp.

Đối với cán bộ quản lý, việc theo dõi tiến độ đăng ký học, kiểm soát số lượng lớp, phân công giảng viên, sắp xếp phòng học… vẫn phụ thuộc nhiều vào thao tác thủ công và các hệ thống rời rạc. Việc tổng hợp dữ liệu, xuất báo cáo hay đưa ra quyết định điều chỉnh thường chậm trễ và thiếu chính xác, đặc biệt là trong những giai đoạn cao điểm.

Thực trạng này không chỉ ảnh hưởng đến chất lượng phục vụ sinh viên mà còn làm giảm hiệu suất điều hành và khả năng phản ứng nhanh của hệ thống đào tạo khi có thay đổi. Trong bối cảnh sinh viên ngày càng đòi hỏi tính minh bạch, công bằng và cá nhân hóa cao trong lộ trình học tập, thì hệ thống đăng ký học truyền thống hiện tại rõ ràng không còn đáp ứng được yêu cầu thực tiễn.

Chính vì vậy, nhu cầu cấp thiết đặt ra là phải xây dựng một hệ thống đăng ký học hiện đại, linh hoạt, dễ mở rộng, có khả năng xử lý đồng thời hàng nghìn yêu cầu, đồng thời cung cấp các tính năng thông minh như kiểm tra trùng lịch học, tối ưu thời khóa biểu, theo dõi tình trạng lớp học theo thời gian thực, và hỗ trợ ra quyết định cho cả sinh viên lẫn nhà quản lý. Đây không chỉ là một bài toán về kỹ thuật, mà còn là thách thức trong việc nâng cao chất lượng đào tạo, hướng tới một mô hình quản trị đại học thông minh và hiệu quả trong thời đại số hóa.

## 1.2 Lý do chọn đề tài

Là một sinh viên chuyên ngành Công nghệ thông tin, em nhận thức sâu sắc rằng công nghệ không chỉ đơn thuần là công cụ hỗ trợ, mà còn là yếu tố quyết định trong việc cải tiến và tái cấu trúc các quy trình quản lý giáo dục hiện nay. Trong quá trình học tập và tiếp xúc thực tế với hệ thống đào tạo tại Trường Đại học Thủy Lợi, em nhận thấy rằng quá trình đăng ký học phần – vốn là một nghiệp vụ diễn ra định kỳ và có tác động trực tiếp đến toàn bộ sinh viên – vẫn tồn tại nhiều bất cập cả về kỹ thuật lẫn trải nghiệm người dùng.

Bên cạnh các vấn đề như nghẽn hệ thống, thiếu cảnh báo trùng lịch, thao tác không trực quan hoặc giao diện chưa tối ưu cho thiết bị di động, hệ thống hiện tại còn chưa được thiết kế theo hướng mở rộng linh hoạt (scalable) hoặc hỗ trợ xử lý đồng thời hiệu quả. Điều này đặc biệt nghiêm trọng trong bối cảnh số lượng sinh viên đăng ký tăng cao theo từng năm học, và nhu cầu cá nhân hóa lịch học ngày càng lớn.

Việc nghiên cứu, thiết kế và triển khai một hệ thống đăng ký học hiện đại, áp dụng các mô hình kiến trúc tiên tiến như Microservices, kết hợp với CI/CD, cơ sở dữ liệu lai (hybrid database), triển khai trên nền tảng đám mây (Docker, Kubernetes) và giao diện thân thiện (React, TypeScript) không chỉ là một hướng đi phù hợp với xu thế công nghệ, mà còn giúp sinh viên tiếp cận sâu hơn với các bài toán thực tiễn và kỹ năng phát triển hệ thống phần mềm quy mô lớn.

Em lựa chọn đề tài này với mong muốn:

* Góp phần cải thiện một trong những quy trình quan trọng bậc nhất trong công tác đào tạo của nhà trường – đăng ký học phần.
* Vận dụng toàn diện kiến thức đã học trong suốt chương trình đào tạo để xây dựng một sản phẩm thực tế, có khả năng triển khai ứng dụng thật sự.
* Tiếp cận với những công nghệ hiện đại đang được sử dụng phổ biến trong các hệ thống doanh nghiệp lớn, từ đó nâng cao năng lực chuyên môn và chuẩn bị tốt hơn cho công việc sau khi tốt nghiệp.

Không chỉ dừng lại ở việc xây dựng một hệ thống kỹ thuật, em mong muốn đề tài này có thể trở thành một bước khởi đầu cho việc chuẩn hóa và hiện đại hóa quy trình đào tạo tại Trường Đại học Thủy Lợi nói riêng, và làm nền tảng tham khảo cho các cơ sở giáo dục khác nói chung trong công cuộc chuyển đổi số toàn diện ngành giáo dục đại học.

## 1.3 Mục tiêu

Mục tiêu tổng thể của đề tài là thiết kế và xây dựng một hệ thống đăng ký học trực tuyến hiện đại, có khả năng hỗ trợ toàn diện quy trình đăng ký học phần cho sinh viên Trường Đại học Thủy Lợi, đồng thời cung cấp các công cụ giám sát, điều phối và phân tích dữ liệu học tập cho cán bộ quản lý đào tạo. Hệ thống hướng đến việc thay thế phương thức đăng ký học truyền thống vốn còn nhiều hạn chế, bằng một nền tảng công nghệ thông minh, linh hoạt và dễ mở rộng.

Cụ thể, đề tài đặt ra các mục tiêu sau:

1. Tự động hóa quy trình đăng ký học phần

* Cho phép sinh viên đăng nhập, tìm kiếm, lựa chọn và đăng ký học phần nhanh chóng theo thời khóa biểu cá nhân.
* Tự động kiểm tra xung đột lịch học, hạn chế tình trạng trùng giờ, sai sót khi đăng ký.
* Cung cấp giao diện trực quan giúp sinh viên dễ dàng theo dõi danh sách học phần đã đăng ký, lịch học cá nhân, trạng thái lớp học (đã đầy, còn chỗ), và điều kiện tiên quyết của học phần.

2. Tối ưu hóa công tác quản lý đào tạo

* Hỗ trợ cán bộ đào tạo trong việc quản lý danh sách lớp học, số lượng sinh viên, phân bổ giảng viên – phòng học theo năng lực và nhu cầu thực tế.
* Cung cấp hệ thống thống kê, báo cáo theo thời gian thực về tình hình đăng ký học của sinh viên theo môn học, lớp, khoa, khóa,...
* Cho phép xuất dữ liệu phục vụ phân tích, dự báo và ra quyết định quản lý.

3. Đảm bảo hiệu năng và khả năng mở rộng

* Ứng dụng kiến trúc Microservices để chia nhỏ hệ thống thành các module độc lập, dễ dàng phát triển, bảo trì và mở rộng trong tương lai.
* Sử dụng công nghệ container (Docker) và nền tảng quản lý triển khai (Kubernetes) để đảm bảo khả năng triển khai linh hoạt và độ tin cậy cao.
* Tối ưu hiệu năng để hệ thống có thể xử lý đồng thời hàng nghìn yêu cầu mà không bị gián đoạn.

4. Cung cấp trải nghiệm người dùng thân thiện

* Thiết kế giao diện web responsive, dễ sử dụng trên nhiều thiết bị (PC, máy tính bảng, điện thoại).
* Tích hợp các tiện ích như tìm kiếm nhanh, bộ lọc môn học, gợi ý đăng ký theo chương trình đào tạo, cảnh báo khi lớp đầy hoặc lịch học bị xung đột.

5. Đảm bảo an toàn và bảo mật dữ liệu

* Áp dụng cơ chế xác thực người dùng thông qua tài khoản Outlook sinh viên nhằm bảo đảm tính chính danh.
* Bảo vệ dữ liệu cá nhân và thông tin học phần bằng các chuẩn mã hóa và phân quyền truy cập hợp lý.

6. Hướng đến triển khai thực tế

* Xây dựng một hệ thống hoàn chỉnh có thể triển khai thử nghiệm hoặc tích hợp với hệ thống quản lý đào tạo hiện tại của nhà trường.
* Có khả năng mở rộng để tích hợp với các chức năng liên quan như quản lý điểm, lịch thi, đăng ký tín chỉ tốt nghiệp,...

## 1.4 Cơ sở lý thuyết

### 1.4.2 Ngôn ngữ lập trình

Hệ thống sử dụng đa dạng các ngôn ngữ lập trình để phục vụ tốt nhất cho yêu cầu về hiệu năng, mở rộng và trải nghiệm người dùng. Cụ thể:  
  
- \*\*React với TypeScript\*\*: React là thư viện UI phổ biến, kết hợp với TypeScript để tăng tính an toàn khi lập trình nhờ hệ thống kiểu tĩnh. Việc sử dụng React giúp tăng khả năng tái sử dụng mã nguồn, giảm thời gian phát triển và tạo ra giao diện người dùng linh hoạt, tương tác tốt. TypeScript hỗ trợ IntelliSense, tự động kiểm tra lỗi cú pháp, giúp lập trình viên giảm thiểu lỗi logic trong giai đoạn viết mã.  
  
- \*\*.NET Core (C#)\*\*: Là nền tảng đa nền tảng, hỗ trợ xây dựng các REST API mạnh mẽ. Với sự hỗ trợ tốt cho gRPC và khả năng tích hợp sâu với hệ sinh thái Microsoft, .NET Core phù hợp để xây dựng backend của các dịch vụ vi mô (microservices). C# cho phép xử lý đa luồng, kết nối cơ sở dữ liệu linh hoạt, đồng thời tối ưu bộ nhớ hiệu quả. Ngoài ra, thư viện Entity Framework hỗ trợ truy vấn cơ sở dữ liệu mạnh mẽ mà không cần viết SQL thuần.  
  
- \*\*YAML, Bash, và Dockerfile\*\*: Được sử dụng để định nghĩa pipeline CI/CD, cấu hình container và các kịch bản khởi tạo môi trường. YAML giúp khai báo tài nguyên trong Kubernetes rõ ràng và dễ kiểm soát.

### 1.4.3 Cơ sở dữ liệu

Dữ liệu trong hệ thống đăng ký học phần rất đa dạng: từ dữ liệu có cấu trúc như thông tin sinh viên, học phần, điểm số; đến dữ liệu phi cấu trúc như lịch sử đăng ký hoặc thông báo hệ thống. Do đó, hệ thống được thiết kế để sử dụng mô hình cơ sở dữ liệu kết hợp (hybrid model):  
  
- \*\*MongoDB\*\*: Là NoSQL dạng document, giúp lưu trữ dữ liệu linh hoạt mà không cần tuân thủ schema cố định. Các đối tượng như lịch sử đăng ký, thông tin học phần hoặc các mối quan hệ giữa sinh viên và lớp học phần được lưu ở dạng BSON (Binary JSON), dễ dàng mở rộng, thích hợp với Microservices.  
  
- \*\*PostgreSQL\*\*: Là hệ quản trị cơ sở dữ liệu quan hệ mạnh mẽ, hỗ trợ đầy đủ ACID, lý tưởng cho các phần liên quan tới bảo mật, phân quyền, thông tin tài khoản. PostgreSQL cũng hỗ trợ lưu trữ JSON, có thể truy vấn dữ liệu bán cấu trúc linh hoạt.  
  
- \*\*Redis\*\*: Là bộ nhớ cache in-memory, giúp giảm tải truy vấn cơ sở dữ liệu trong các tình huống cần tốc độ cao như xác thực, lưu session, hoặc xếp hàng đợi xử lý nhanh.

### 1.4.4 Kiến trúc Microservices

Microservices là kiến trúc chia nhỏ hệ thống thành các dịch vụ độc lập, mỗi dịch vụ chịu trách nhiệm một phần chức năng rõ ràng. Kiến trúc này đặc biệt phù hợp với hệ thống đăng ký học phần nhờ những lợi ích:  
  
- \*\*Tăng khả năng mở rộng (scalability)\*\*: Mỗi service có thể được scale độc lập theo nhu cầu thực tế. Ví dụ: dịch vụ đăng ký học có thể mở rộng trong các đợt cao điểm mà không ảnh hưởng tới các phần còn lại.  
- \*\*Tăng độ tin cậy\*\*: Khi một service bị lỗi, các phần khác của hệ thống vẫn có thể hoạt động bình thường.  
- \*\*Phát triển và triển khai độc lập\*\*: Các nhóm phát triển có thể làm việc trên các dịch vụ riêng biệt mà không ảnh hưởng đến nhau.  
  
Các thành phần chính bao gồm:  
- \*\*Identity Service\*\*: Xác thực người dùng và phân quyền theo vai trò (sinh viên, giảng viên, PĐT).  
- \*\*Student Service\*\*: Lưu trữ và xử lý toàn bộ thông tin sinh viên và kết quả học tập.  
- \*\*Training Service\*\*: Quản lý môn học, học phần, điều kiện tiên quyết, chương trình đào tạo.  
- \*\*Register Service\*\*: Đăng ký nguyện vọng, lớp học phần, xử lý kiểm tra trùng lịch, tín chỉ vượt giới hạn.  
- \*\*Notification Service\*\*: Gửi thông báo qua email, web push, hoặc tích hợp dịch vụ SMS.  
  
Tất cả các service giao tiếp với nhau thông qua REST API, gRPC, và hệ thống hàng đợi sự kiện Kafka để đồng bộ hoá dữ liệu.

### 1.4.5 CI/CD (Continuous Integration/Continuous Deployment)

CI/CD giúp quy trình triển khai phần mềm trở nên liên tục, chính xác và ít phụ thuộc con người. Trong hệ thống này, CI/CD bao gồm:  
  
- \*\*GitHub Actions\*\*: Quản lý các pipeline build, test và deploy. Mỗi khi có commit mới, hệ thống sẽ tự động build project, chạy kiểm thử đơn vị, kiểm tra bảo mật mã nguồn và đóng gói thành Docker image.  
- \*\*DockerHub\*\*: Là nơi lưu trữ các container images sau khi build, đảm bảo mỗi phiên bản phần mềm đều có thể kiểm soát, rollback nếu cần.  
- \*\*Kubernetes Deployment\*\*: Sau khi đẩy image, Kubernetes sẽ tự động lấy phiên bản mới và triển khai lên môi trường thực tế bằng rolling update, không làm gián đoạn hệ thống.  
  
CI/CD giúp nâng cao chất lượng phần mềm, giảm rủi ro khi triển khai và đẩy nhanh vòng đời phát triển.

### 1.4.6 Môi trường triển khai

Toàn bộ hệ thống được triển khai trên cụm Kubernetes để đảm bảo tính linh hoạt, mở rộng và khôi phục nhanh khi có sự cố. Một số đặc điểm nổi bật của môi trường triển khai:  
  
- \*\*Container hóa với Docker\*\*: Mỗi thành phần của hệ thống (frontend/backend/service) đều được đóng gói trong container, đảm bảo sự đồng nhất giữa các môi trường dev, staging và production.  
- \*\*Quản lý container bằng Kubernetes\*\*: Cho phép tự động hoá việc khởi động, cân bằng tải, scale in/out và khôi phục dịch vụ.  
- \*\*Ingress Controller\*\*: Làm cầu nối định tuyến request từ bên ngoài vào từng service.  
- \*\*ConfigMap & Secret\*\*: Dùng để quản lý cấu hình và thông tin nhạy cảm (API key, thông tin đăng nhập).

### 1.4.7 Monitor

Việc giám sát hệ thống là tối quan trọng để phát hiện sớm lỗi, đánh giá hiệu năng và hỗ trợ vận hành hiệu quả. Hệ thống sử dụng:  
  
- \*\*Prometheus + Grafana\*\*: Prometheus thu thập số liệu hệ thống như CPU, bộ nhớ, số request,... Sau đó, dữ liệu được hiển thị trên Grafana với biểu đồ sinh động. Các chỉ số được đặt ngưỡng để cảnh báo tự động.  
- \*\*Jaeger Tracing\*\*: Ghi nhận hành trình một request đi qua các dịch vụ, giúp xác định vị trí gây chậm, lỗi logic hoặc deadlock trong xử lý nghiệp vụ.  
- \*\*AlertManager\*\*: Khi hệ thống vượt ngưỡng cấu hình, AlertManager gửi thông báo đến email, Slack hoặc công cụ giám sát trung tâm khác.  
  
Ngoài ra, hệ thống có thể tích hợp với ELK Stack (Elasticsearch, Logstash, Kibana) để xử lý log tập trung, phục vụ truy vết lỗi và kiểm toán.

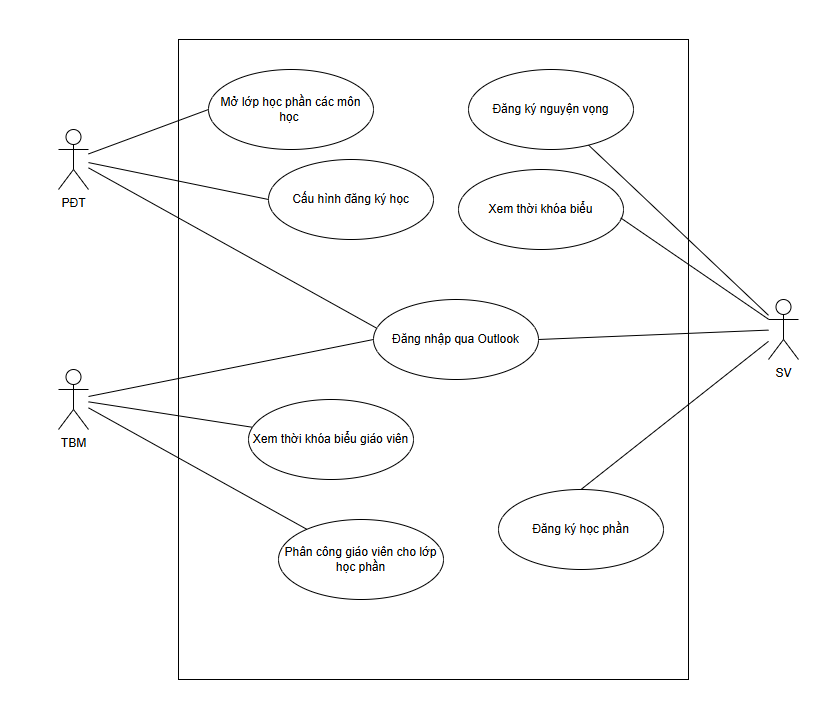
# CHƯƠNG 2 PHÂN TÍCH THIẾT KẾ

## 2.1 Thu thập yêu cầu

Bảng 1 Yêu cầu hệ thống

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Actor | Công việc | Ghi chú |
|  | PĐT, TBM, SV | Đăng nhập với tài khoản Outlook. | x |
|  | PĐT | Tạo kì học mới. |  |
|  | Tổ chức đăng ký nguyện vọng học cho sinh viên. |  |
|  | Báo cáo đăng ký nguyện vọng học. |  |
|  | Mở lớp học phần môn học. | x |
|  | Lập thời khóa biểu lớp học. | x |
|  | Tổ chức đăng ký lớp học phần. |  |
|  | Thay đổi sinh viên đăng ký học. | x |
|  | Hủy lớp học phần. | x |
|  | Lấy danh sách lớp học phần của môn học. |  |
|  | TBM | Lấy danh sách lớp học mở theo bộ môn trong kỳ học hiện tại. |  |
|  | Phân công giảng viên cho lớp học phần. | x |
|  | Xem thời khóa biểu giảng viên. |  |
|  | SV | Xem thông tin cá nhân. |  |
|  | Xem thông tin kết quả học tập. |  |
|  | Lấy danh sách môn học của chương trình đào tạo theo học. |  |
|  | Đăng ký nguyện vọng học. | x |
|  | Lấy thông tin lớp học phần đang mở. |  |
|  | Đăng ký học phần môn học. | x |
|  | Lấy thời khóa biểu học phần. | x |

## 2.2 Use Case tổng quát



Hình 1 Use-case tổng quát

## **2.3 Đặc tả use case**

### 2.3.1 Đăng nhập với Outlook

Bảng 2 Đăng nhập với OutLook

|  |  |  |
| --- | --- | --- |
| **Use case** | **Đăng nhập với OutLook** | |
| Mục tiêu | Đăng nhập vào hệ thống để sử dụng. | |
| Điều kiện tiên quyết | Người dùng chưa đăng nhập. | |
| Điều kiện kết thúc thành công | Nếu người dùng đăng nhập thành công: Người dùng được chuyển đến trang chủ phù hợp. | |
| Phân quyền Actors | PĐT, TBM, SV | |
| Kích hoạt | Người dùng thực hiện lệnh đăng nhập. | |
| **Luồng sự kiện chính** | **Bước** | **Hành động của tác nhân** |
|  |  | Hệ thống mở trang đăng nhập của Outlook. |
|  |  | Người dùng đăng nhập với tài khoản Outlook của trường. |
|  |  | Hệ thống cập nhật thông tin đăng nhập. |
|  |  | Hệ thống chuyển người đến giao diện trang chủ thích hợp. |
| **Luồng ngoại lệ** | **Bước** | **Hành động phân nhánh** |
|  | 1 | 1a. Người dùng thoát khỏi trang đăng nhập. Use case kết thúc. |
|  | 3 | 3a. Người dùng đăng nhập với email không phải của trường. Hệ thống quay trở lại trang đăng nhập. |
|  | 4 | 4a. Nếu người dùng đăng nhập lần đầu, hệ thống chuyển đến trang xác nhận lại thông tin người dùng. |

### 2.3.2 Cấu hình lớp học phần môn học

Bảng 3 Cấu hình lớp học phần môn học

|  |  |  |
| --- | --- | --- |
| **Use case** | **Cấu hình lớp học phần môn học.** | |
| Mục tiêu | Cho phép PĐT có thể cấu hình các tham số để lập thời khóa biểu cho lớp học phần tại kì học đăng ký học. | |
| Điều kiện tiên quyết | Người thực hiện đã đăng nhập và có quyền quản trị hệ thống. Hệ thống đang hoạt động bình thường. | |
| Điều kiện kết thúc thành công | Hệ thống thông báo thông tin cấu hình môn học đã được cập nhật thành công. Dữ liệu cấu hình phản ánh đúng trên hệ thống. | |
| Phân quyền Actors | PĐT | |
| Kích hoạt | PĐT nhấp nhấp vào nút “Cấu hình lớp học” trên trang danh sách môn học. | |
| **Luồng sự kiện chính** | **Bước** | **Hành động của tác nhân** |
|  | 1 | PĐT chọn giai đoạn muốn cấu hình lớp học phần cho môn học tương ứng. |
|  | 2 | Hệ thống hiển thị thông tin lớp học cấu hình mặc định dựa trên số tín chỉ của môn học tương ứng với giai đoạn học tương ứng đã chọn. |
|  | 3 | PĐT điều chỉnh thông tin phù hợp với kì học đăng ký hiện tại. |
|  | 4 | Hệ thống kiểm tra thông tin PĐT nhập vào. |
|  | 5 | Hệ thống lưu trữ thông tin cấu hình vào cơ sở dữ liệu. |
|  | 6 | Hệ thống thông báo cập nhật cấu hình lớp học thành công. |
| **Luồng ngoại lệ** | **Bước** | **Hành động phân nhánh** |
|  | 2 | 2a Thông tin cấu hình lớp học phần của giai đoạn chọn chưa có dữ liệu, hệ thống thông báo tương ứng. |
|  | 3 | 3a. PĐT chưa nhập đầy đủ thông tin, hệ thống thông báo lỗi và yêu cầu nhập lại. |
|  | 3b. Thông tin PĐT nhập chưa hợp lệ, hệ thống thông báo lỗi và yêu cầu người dùng sửa lại. |

### 2.3.3 Tạo lớp học phần môn học

Bảng 4 Tạo lớp học phần môn học

|  |  |  |
| --- | --- | --- |
| **Use case** | **Tạo lớp học phần môn học.** | |
| Mục tiêu | Tạo danh sách lớp học phần môn học chuẩn bị cho việc đăng ký học phần. | |
| Điều kiện tiên quyết | PĐT đã đăng nhập hệ thống. | |
| Điều kiện kết thúc thành công | Hệ thống thông báo tạo lớp thành công. | |
| Phân quyền Actors | PĐT | |
| Kích hoạt | PĐT chọn chức năng ”Thêm lớp mới” trên trang danh sách lớp học. | |
| **Luồng sự kiện chính** | **Bước** | **Hành động của tác nhân** |
|  |  | PĐT chọn giai đoạn muốn tạo lớp học phần cho môn học tương ứng và nhấn chọn “Thêm lớp mới”. |
|  |  | Hệ thống hiển thị form nhập thông tin số lớp cần tạo và sinh viên dự kiến mỗi lớp. |
|  |  | PĐT nhập thông tin vào form và nhấn xác nhận. |
|  |  | Hệ thống kiểm tra thông tin người dùng đã nhập. |
|  |  | Hệ thống tự động tạo lớp mới dựa trên thông tin người dùng đã nhập cấu hình tương ứng với giai đoạn học đã chọn. |
|  |  | Hệ thống thông báo tạo lớp học thành công và hiển thị các lớp học đã tạo lên màn hình. |
|  |  | PĐT nhấn chọn “Chỉnh sửa” nếu muốn cập nhật lại thông tin từng lớp học. |
|  |  | Hệ thống hiển thị form bao gồm thông tin đã có của lớp học phần đã chọn tương ứng. |
|  |  | PĐT chỉnh sửa thông tin trên form và nhấn xác nhận. |
|  |  | Hệ thống kiểm tra thông tin người dùng đã nhập và lưu trữ vào cơ sở dữ liệu. |
|  |  | Hệ thống thông báo cập nhật thông tin thành công. |
| **Luồng ngoại lệ** | **Bước** | **Hành động phân nhánh** |
|  | 1 | 1a. Cấu hình lớp học phần môn học tại giai đoạn người dùng chọn chưa được cấu hình, hệ thống thông báo tương ứng và yêu cầu người dùng cấu hình. |

### 2.3.4 Lập thời khóa biểu lớp học phần

Bảng 5 Lập thời khóa biểu lớp học phần

|  |  |  |
| --- | --- | --- |
| **Use case** | **Lập thời khóa biểu lớp học phần.** | |
| Mục tiêu | .Sắp xếp lịch học cho các lớp học phần đã được mở trong một kì học. | |
| Điều kiện tiên quyết | PĐT đã đăng nhập vào hệ thống. | |
| Điều kiện kết thúc thành công | Hệ thống thông báo xếp lịch học thành công. | |
| Phân quyền Actors | PĐT | |
| Kích hoạt | PĐT chọn chức năng "Xếp thời khóa biểu tự động" từ giao diện danh sách lớp học phần. | |
| **Luồng sự kiện chính** | **Bước** | **Hành động của tác nhân** |
|  |  | PĐT nhấn chọn các lớp học phần muốn lập thời khóa biểu và nhấn chọn “Xếp thời khóa biểu tự động”. |
|  |  | Hệ thống kiểm tra thông tin các lớp học phần người dùng đã chọn và tiến hành xếp tự động thời khóa biểu cho các lớp học đã chọn. |
|  |  | Hệ thống thông báo đã xếp lịch học thành công . |
|  |  | PĐT chọn lớp học phần cần điều chỉnh lịch học. |
|  |  | Hệ thống hiển thị thông tin lớp học phần và hiển thị thời khóa biểu hiện tại. |
|  |  | PĐT điều chỉnh thời khóa biểu hiện tại và nhấn lưu lại. |
|  |  | Hệ thống kiểm tra thông tin người dùng đã điều chỉnh và lưu trữ vào cơ sở dữ liệu. |
|  |  | Hệ thống thông báo cập nhật thời khóa biểu thành công. |
| **Luồng ngoại lệ** | **Bước** | **Hành động phân nhánh** |
|  | 2 | 2a. Phòng học không còn chỗ trống, hệ thống thông báo không thể xếp lịch học |

### 2.3.5 Hủy lớp học phần

Bảng 6 Hủy lớp học phần

|  |  |  |
| --- | --- | --- |
| **Use case** | **Hủy lớp học phần.** | |
| Mục tiêu | Cho phép PĐT hủy một lớp học phần đã được tạo trong hệ thống khi không xếp được thời khóa biểu, hoặc không đủ số SV tối thiểu hoặc không xếp được GV. | |
| Điều kiện tiên quyết | PĐT đã đăng nhập hệ thống. Lớp học phần cần hủy phải tồn tại trong hệ thống. | |
| Điều kiện kết thúc thành công | Hệ thống thông báo hủy lớp học phần thành công và lớp học phần không còn hiển thị trong danh sách hoặc được đánh dấu là đã hủy. | |
| Phân quyền Actors | PĐT | |
| Kích hoạt | PĐT chọn chức năng "Hủy lớp học phần" từ giao diện chi tiết lớp học phần. | |
| **Luồng sự kiện chính** | **Bước** | **Hành động của tác nhân** |
|  | 1 | Hệ thống xác nhận lại với người dùng chắc chắn muốn hủy. |
|  | 2 | PĐT nhấn xác nhận hủy lớp học phần. |
|  | 3 | Hệ thống kiểm tra lại thông tin lớp học phần chọn hủy và cập nhật lại trạng thái vào cơ sở dữ liệu. |
|  | 4 | Hệ thống thông báo cập nhật trạng thái lớp học phần thành công. |
| **Luồng ngoại lệ** | **Bước** | **Hành động phân nhánh** |
|  | 3 | 3a. Trạng thái hiện tại của lớp phần là đã hủy, thông báo cập nhật trạng thái không thành công. |

### 2.3.7 Thay đổi lớp học phần sinh viên

Bảng 7 Thay đổi lớp học phần sinh viên

|  |  |  |
| --- | --- | --- |
| **Use case** | **Sinh viên thay đổi đăng ký học.** | |
| Mục tiêu | Cho phép PĐT thay đổi lớp học phần này sang lớp học phần khác của cùng một môn. | |
| Điều kiện tiên quyết | PĐT đã đăng nhập hệ thống. | |
| Điều kiện kết thúc thành công | Hệ thống thông báo cập nhật thông tin thành công. | |
| Phân quyền Actors | PĐT | |
| Kích hoạt | PĐT chọn chức năng “Thay đổi đăng ký học” trên trang danh sách SV. | |
| **Luồng sự kiện chính** | **Bước** | **Hành động của tác nhân** |
|  |  | Hệ thống hiển thị thông tin các lớp học phần đã đăng ký học của SV. |
|  |  | PĐT chọn lớp học phần của SV cần thay đổi đăng ký học. |
|  |  | Hệ thống hiển thị các lớp học phần cùng môn với lớp học phần đã chọn có thể thay thế. |
|  |  | PĐT chọn một lớp học phần trên danh sách hệ thống hiển thị có thể thay thế. |
|  |  | Hệ thống kiểm tra thông tin lớp học đã chọn và thông tin SV đã đăng ký. |
|  |  | Hệ thống thông báo cập nhật thành công và cập nhật thông tin đăng ký học của SV vào cơ sở dữ liệu. |
| **Luồng ngoại lệ** | **Bước** | **Hành động phân nhánh** |
|  | 3 | 3a. Không có lớp học nào có thể thay thế, hệ thống thông báo không có lớp tương ứng. |
|  | 3b. Lớp học người dùng chọn đã đủ số lượng SV, hệ thống hiển thị cảnh báo lớp đã đầy. |

### 2.3.8 Phân công giảng viên cho lớp học phần

Bảng 8 Phân công giảng viên cho lớp học phần

|  |  |  |
| --- | --- | --- |
| **Use case** | Phân công giảng viên cho lớp học phần**.** | |
| Mục tiêu | Cho phép TBM phân công GV cho lớp học phần phục vụ cho việc giảng dạy. | |
| Điều kiện tiên quyết | TBM đã đăng nhập vào hệ thống. | |
| Điều kiện kết thúc thành công | Hệ thống thông báo phân công giảng viên thành công. | |
| Phân quyền Actors | TBM | |
| Kích hoạt | TBM chọn chức năng "Phân công giảng viên" trên trang danh sách môn học. | |
| **Luồng sự kiện chính** | **Bước** | **Hành động của tác nhân** |
|  |  | TBM chọn môn học cần phân công GV. |
|  |  | Hệ thống hiển thị danh sách các lớp học phần tương ứng với môn học đã chọn và giai đoạn tương ứng. |
|  |  | TBM chọn lớp học phần cần phân công giảng viên. |
|  |  | Hệ thống hiển thị danh sách các giảng viên có thể phân công cho lớp học phần đã chọn, lần lượt theo cấp độ trong cùng bộ môn, trong cùng khoa, trong cùng trường. |
|  |  | TBM chọn GV từ danh sách hiển thị và xác nhận phân công. |
|  |  | Hệ thống kiểm tra tính hợp lệ của GV đã chọn. |
|  |  | Hệ thống cập nhật thông tin giảng viên được phân công vào lớp học phần trong cơ sở dữ liệu. |
|  |  | Hệ thống thông báo phân công giảng viên thành công. |
| **Luồng ngoại lệ** | **Bước** | **Hành động phân nhánh** |
|  | 4 | 4a. Không có giáo viên nào phù hợp cho lớp học phần, hệ thống thông báo tương ứng. |

### 2.3.9 Cấu hình đăng ký học phần

Bảng 9 Cấu hình đăng ký học phần

|  |  |  |
| --- | --- | --- |
| **Use case** | **Cấu hình đăng ký học phần.** | |
| Mục tiêu | Cho phép PĐT thiết lập các cấu hình cần thiết cho đợt đăng ký học phần. | |
| Điều kiện tiên quyết | PĐT đã đăng nhập hệ thống. | |
| Điều kiện kết thúc thành công | Thông tin kết quả học tập của SV được hiển thị đầy đủ. | |
| Phân quyền Actors | PĐT | |
| Kích hoạt | PĐT chọn chức năng “Cấu hình đăng ký học”. | |
| **Luồng sự kiện chính** | **Bước** | **Hành động của tác nhân** |
|  |  | Hệ thống hiển thị form cấu hình đăng ký học. |
|  |  | PĐT chọn học kì áp dụng và nhập thông tin vào form cấu hình đăng ký học. |
|  |  | Hệ thống kiểm tra dữ liệu người dùng đã nhập. |
|  |  | Hệ thống thông báo cập nhật thành công và cập nhật thông tin đăng ký học vào cơ sở dữ liệu. |
| **Luồng ngoại lệ** | **Bước** | **Hành động phân nhánh** |
|  | 2 | 2a. Kì học chọn đã được cấu hình, hệ thống thông báo tương ứng. |

### 2.3.10 Lấy thông tin kết quả học tập

Bảng 10 Lấy thông tin kết quả học tập

|  |  |  |
| --- | --- | --- |
| **Use case** | **Lấy thông tin kết quả học tập.** | |
| Mục tiêu | SV xem kết quả học tập của môn học theo kì học của chương trình đào tạo tương ứng mà SV đó đang theo học. | |
| Điều kiện tiên quyết | SV đã đăng nhập hệ thống. | |
| Điều kiện kết thúc thành công | Thông tin kết quả học tập của SV được hiển thị đầy đủ. | |
| Phân quyền Actors | SV | |
| Kích hoạt | SV chọn chức năng “Kết quả học tập”. | |
| **Luồng sự kiện chính** | **Bước** | **Hành động của tác nhân** |
|  |  | Hệ thống truy xuất dữ liệu các kì học tập của SV tương ứng. |
|  |  | Sinh viên chọn kì học muốn xem kết quả học tập. |
|  |  | Hệ thống truy xuất dữ liệu kết quả học tập theo kì của sinh viên tương ứng. |
|  |  | Hệ thống hiển thị kết quả học tập của sinh viên theo kì tương ứng. |
| **Luồng ngoại lệ** | **Bước** | **Hành động phân nhánh** |
|  | 2 | 2a. Kì học chọn không chưa có dữ liệu, hệ thống hiển thị dữ liệu chưa được cập nhật. |

### 2.3.11 Đăng ký lớp học phần

Bảng 11 Đăng ký lớp học phần

|  |  |  |
| --- | --- | --- |
| **Use case** | **Đăng ký lớp học phần.** | |
| Mục tiêu | SV đăng ký các lớp học phần muốn học trong học kì hiện tại. | |
| Điều kiện tiên quyết | SV đã đăng nhập hệ thống. | |
| Điều kiện kết thúc thành công | Thông báo đăng ký lớp học thành công. | |
| Phân quyền Actors | SV | |
| Kích hoạt | SV chọn chức năng “Đăng ký học mới”. | |
| **Luồng sự kiện chính** | **Bước** | **Hành động của tác nhân** |
|  |  | Sinh viên chọn chương trình đào tạo muốn đăng ký học phần. |
|  |  | Hệ thống hiển thị các môn học được đăng ký học phần tương ứng với CTĐT đã chọn. |
|  |  | SV chọn môn học muốn đăng ký học phần. |
|  |  | Hệ thống truy xuất dữ liệu học phần mở của môn học đã chọn và hiển thị danh sách lớp học phần tương ứng với môn học. |
|  |  | Sinh viên chọn lớp học phần muốn đăng ký. |
|  |  | Hệ thống kiểm tra kiểm tra thông tin SV đăng ký. |
|  |  | Hệ thống lưu trữ dữ liệu lớp học phần SV đã đăng ký vào cơ sở dữ liệu. |
|  |  | Hệ thống thông báo đăng ký học thành công. |
| **Luồng ngoại lệ** | **Bước** | **Hành động phân nhánh** |
|  | 4 | 4a. Môn học SV chọn không có lớp học phần nào, hệ thống thông báo dữ liệu trống. |
|  | 4b. Lớp học phần đã đầy, hệ thống hiện cảnh báo lớp đã đầy. |
|  | 6 | 6a. SV không đủ điều kiện đăng ký vào lớp HP, hệ thống thông báo lỗi tương ứng. |
|  | 6b. HP không còn, hệ thống thông báo lỗi tương ứng. |

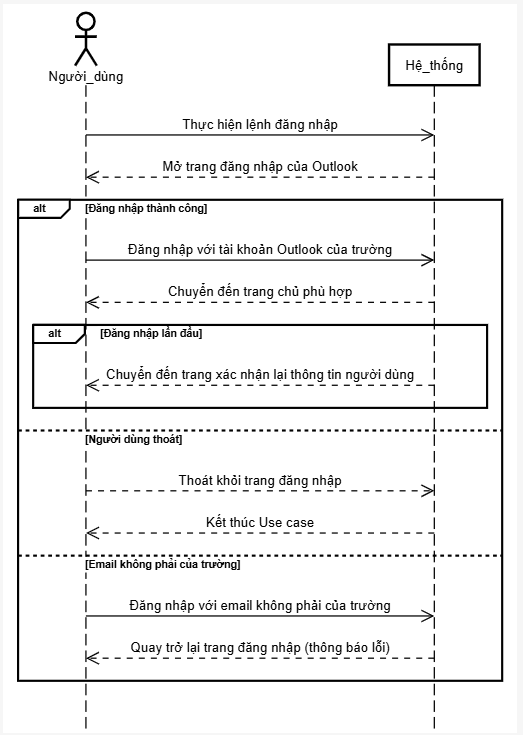
### 2.3.12 Lấy thông tin thời khoá biểu học phần đã đăng ký

Bảng 12 Lấy thông tin thời khóa biểu học phần đăng ký

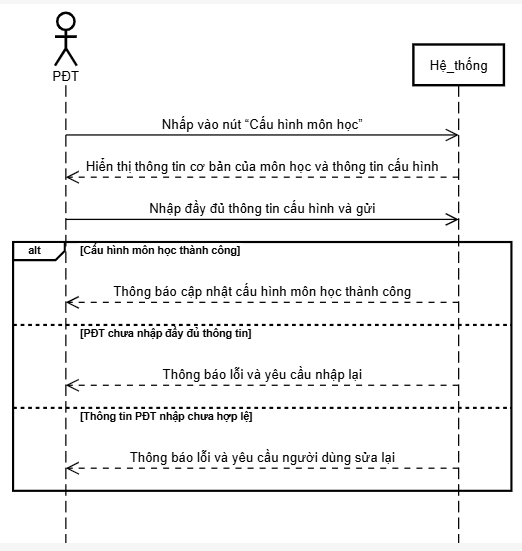
|  |  |  |
| --- | --- | --- |
| **Use case** | **Lấy thông tin thời khoá biểu học phần.** | |
| Mục tiêu | SV xem lịch học của các lớp học phần đã đăng ký. | |
| Điều kiện tiên quyết | SV đã đăng nhập hệ thống. | |
| Điều kiện kết thúc thành công | Hệ thống hiển thị thời khóa biểu của các lớp học phần theo tuần. | |
| Phân quyền Actors | SV | |
| Kích hoạt | SV chọn chức năng thời khóa biểu. | |
| **Luồng sự kiện chính** | **Bước** | **Hành động của tác nhân** |
|  |  | Hệ thống truy xuất kì học của SV và hiển thị. |
|  |  | Sinh viên học kì muốn xem thời khóa biểu. |
|  |  | Hệ thống truy xuất thời khóa biểu của các lớp học phần trong học kì đã chọn tương ứng. |
|  |  | Hiển thị dạng bảng các lớp học phần tương ứng theo tuần. |
| **Luồng ngoại lệ** | **Bước** | **Hành động phân nhánh** |
|  | 2 | 2a. Học kì chọn chưa có thời khóa biểu, hệ thống hiển thị dữ liệu trống. |

## 2.4 Biểu đồ tuần tự

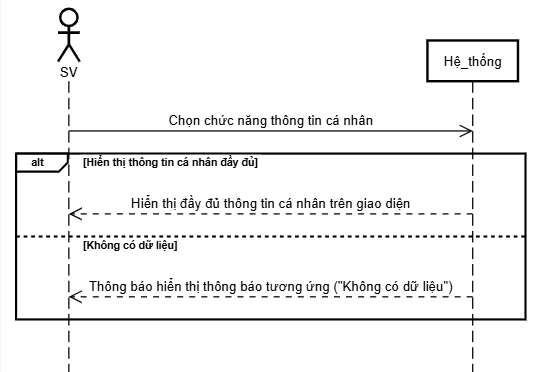
### 2.4.1 Usecase đăng nhập với Outlook



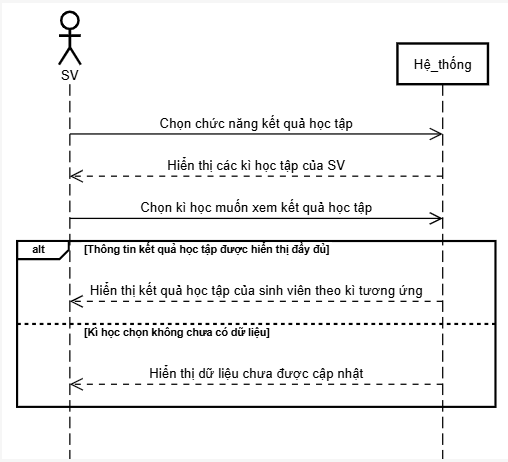
### 2.4.3 Usecase cấu hình môn học



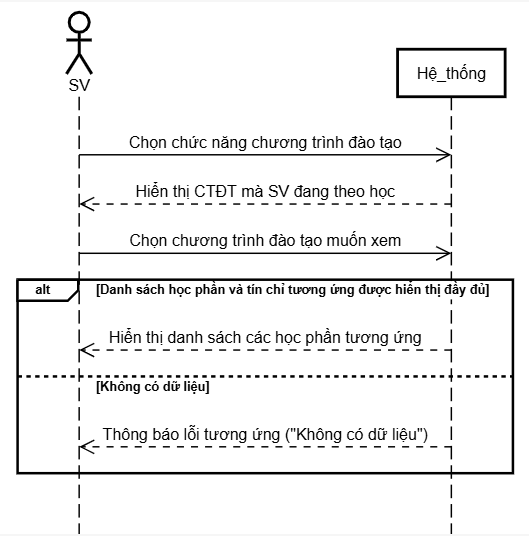
### 2.4.4 Usecase lấy thông tin cá nhân



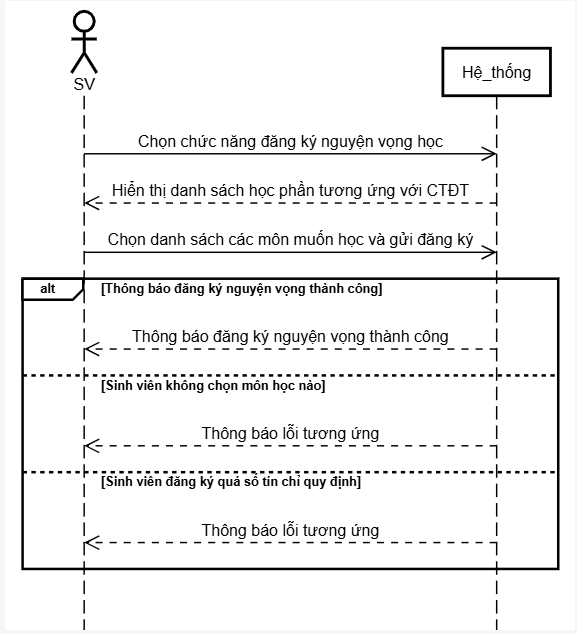
### 2.4.5 Usecase lấy thông tin kết quả học tập



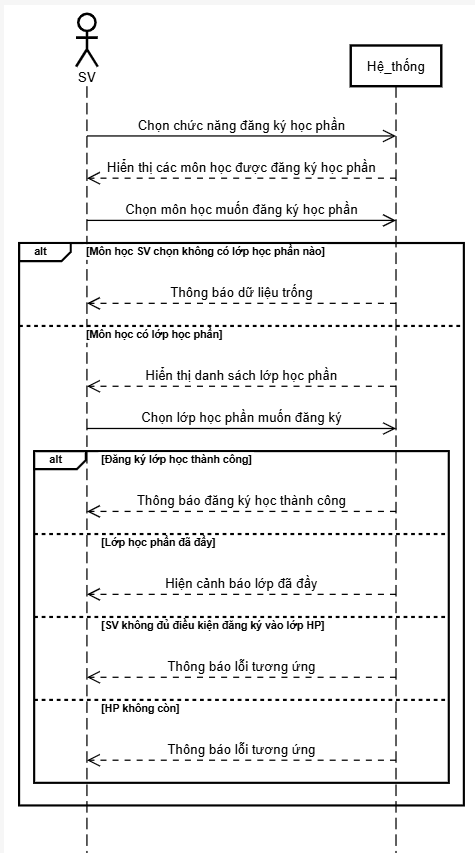
### 2.4.6 Usecase lấy thông tin chương trình đào tạo



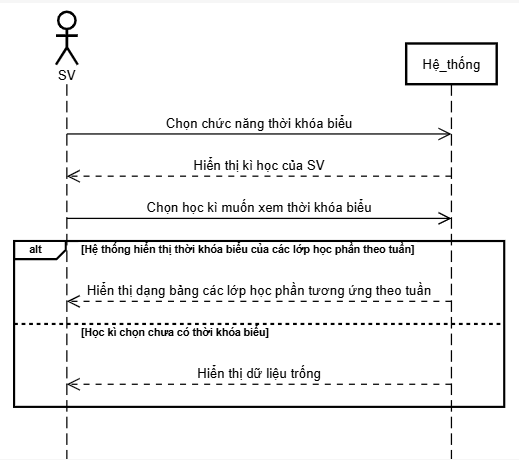
### 2.4.7 Usecase đăng ký nguyện vọng học



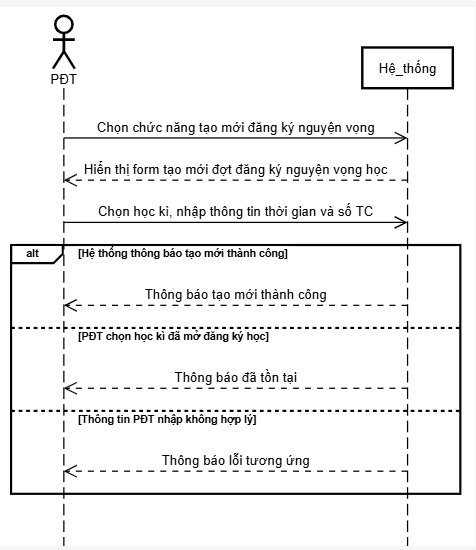
### 2.3.8 Đăng ký học phần



### 2.3.9 Usecase lấy thông tin thời khóa biểu học phần

****

### 2.3.11 Tạo đợt đăng ký nguyện vọng học



### 2.3.13 Usecase mở lớp học phần



### 2.3.14 Usecase sinh viên thay đổi đăng ký học

## 2.5 Lớp phân tích

### 2.5.1 Xác định thực thể

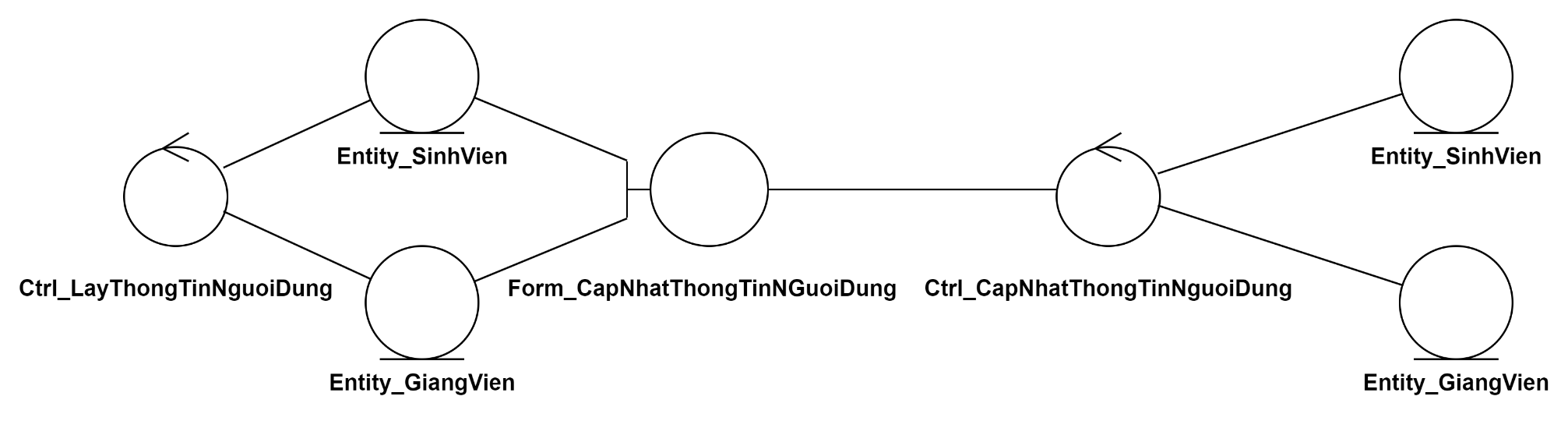
* Entity\_Student
* Entity\_Staff
* Entity\_Subject
* Entity\_CourseClass
* Entity\_Semester
* Entity\_Room
* Entity\_EducationProgram
* Entity\_Register
* Entity\_Notification
* Entity\_SlotTimeline
* Entity\_SubjectScheduleConfig
* Entity\_Department
* Entity\_StudentSemester
* Entity\_CourseClassCondition

### 2.5.2 Xác định các lớp biên

* Form\_GiaoDienDangNhap
* Form\_ GiaoDienThongTinCaNhanSV
* Form\_ GiaoDienKetQuaHocTapSV
* Form\_ GiaoDienChuongTrinhDaoTaoSV
* Form\_ GiaoDienDangKyNguyenVongSV
* Form\_ GiaoDienDangKyLopHocPhanSV
* Form\_ GiaoDienThoiKhoaBieuSV
* Form\_ GiaoDienCauHinhLopHocPhan
* Form\_ GiaoDienPhanCongGiangVien
* Form\_ GiaoDienThoiKhoaBieuGV

### 2.5.3 Xác định các lớp điều khiển

### 2.5.4 Biểu đồ



### 2.5.5 Cơ sở dữ liệu

### 2.5.6 Các bảng

* CourseClass

|  |  |  |
| --- | --- | --- |
| Trường | Kiểu dữ liệu | Mô tả |
| \_id | ObjectId | Định danh duy nhất của lớp học phần |
| CourseClassCode | String | Mã lớp học phần |
| CourseClassName | String | Tên lớp học phần |
| CreatedAt | ISODate | Thời điểm tạo lớp |
| Index | Int32 | Số thứ tự trong danh sách lớp |
| NumberStudents | Int32 | Số sinh viên đã đăng ký lớp |
| ParentCourseClassCode | String | Mã lớp học phần cha (nếu có) |
| SemesterCode | String | Mã học kỳ diễn ra lớp |
| SessionLengths | Array | Danh sách thời lượng các buổi học |
| Stage | Int32 | Giai đoạn mở lớp |
| Status | Int32 | Trạng thái lớp học |
| StudentIds | list | Danh sách ID sinh viên đã đăng ký |
| SubjectCode | String | Mã học phần được giảng dạy |
| TeacherCode | String | Mã giảng viên phụ trách |
| TotalSession | Int32 | Tổng số buổi học |
| UpdatedAt | String | Thời điểm cập nhật gần nhất |
| WeekEnd | Int32 | Tuần kết thúc học phần |
| WeekStart | Int32 | Tuần bắt đầu học phần |
| CourseClassType | Int32 | Loại lớp (lý thuyết, thực hành, lab, tích hợp) |
| NumberStudentsExpected | Int32 | Số sinh viên dự kiến |
| TeacherName | String | Tên giảng viên |

* CourseClassCondition

|  |  |  |
| --- | --- | --- |
| Trường | Kiểu dữ liệu | Mô tả |
| \_id | ObjectId | Định danh điều kiện |
| ConditionCode | String | Mã điều kiện |
| CreatedAt | ISODate | Ngày tạo |
| UpdatedAt | String | Ngày cập nhật |
| ConditionName | String | Tên điều kiện |

* Department

|  |  |  |
| --- | --- | --- |
| Trường | Kiểu dữ liệu | Mô tả |
| \_id | ObjectId | Định danh khoa |
| CreatedAt | ISODate | Ngày tạo |
| DepartmentCode | String | Mã khoa |
| DepartmentName | String | Tên khoa |
| Path | String | Chuỗi đường dẫn phân cấp tổ chức |
| UpdatedAt | String | Ngày cập nhật |

* EducationProgram

|  |  |  |
| --- | --- | --- |
| Trường | Kiểu dữ liệu | Mô tả |
| \_id | ObjectId | Định danh chương trình |
| Code | String | Mã chương trình đào tạo |
| CourseCode | String | Mã khoá học |
| CreatedAt | ISODate | Ngày tạo |
| EducationSubjects | Array | Danh sách học phần trong chương trình |
| Name | String | Tên chương trình |
| SpecialityCode | String | Mã chuyên ngành |
| TrainingTime | Double | Thời gian đào tạo (năm) |
| UpdatedAt | String | Ngày cập nhật |
| Type | Int32 | Loại chương trình (chính quy, liên thông...) |

* Notification

|  |  |  |
| --- | --- | --- |
| Trường | Kiểu dữ liệu | Mô tả |
| \_id | ObjectId | Định danh thông báo |
| Content | String | Nội dung thông báo |
| CreatedAt | ISODate | Ngày tạo |
| Recipients | String | Đối tượng nhận |
| Title | String | Tiêu đề thông báo |
| UpdatedAt | String | Ngày cập nhật |
| Roles | Array | Danh sách vai trò nhận thông báo |

* Register

|  |  |  |
| --- | --- | --- |
| Trường | Kiểu dữ liệu | Mô tả |
| \_id | UUID | Định danh đợt đăng ký |
| CurrentState | String | Trạng thái hiện tại |
| MaxCredit | Int32 | Số tín chỉ tối đa |
| NumberStudent | Int32 | Tổng số sinh viên |
| NumberSubject | Int32 | Số học phần mở |
| SemesterCode | String | Mã học kỳ |
| StudentRegisterEnd | ISODate | Kết thúc đăng ký chính thức |
| Version | Int32 | Phiên bản |
| WishEndDate | ISODate | Kết thúc đăng ký nguyện vọng |
| MinCredit | Int32 | Số tín chỉ tối thiểu |
| NumberWish | Int32 | Số nguyện vọng |
| StudentRegisterStart | ISODate | Bắt đầu đăng ký chính thức |
| WishStartDate | ISODate | Bắt đầu đăng ký nguyện vọng |

* Room

|  |  |  |
| --- | --- | --- |
| Trường | Kiểu dữ liệu | Mô tả |
| \_id | ObjectId | Định danh phòng |
| BuildingCode | String | Mã toà nhà |
| CreatedAt | ISODate | Ngày tạo |
| Name | String | Tên hiển thị phòng |
| SupportedConditions | Array | Điều kiện hỗ trợ |
| UpdatedAt | String | Ngày cập nhật |
| Capacity | Int32 | Sức chứa tối đa |
| Code | String | Mã phòng |

* Semester

|  |  |  |
| --- | --- | --- |
| Trường | Kiểu dữ liệu | Mô tả |
| \_id | ObjectId | Định danh học kỳ |
| CreatedAt | ISODate | Ngày tạo |
| EndDate | String | Ngày kết thúc học kỳ |
| SemesterCode | String | Mã học kỳ |
| StartDate | String | Ngày bắt đầu học kỳ |
| UpdatedAt | String | Ngày cập nhật |
| ParentSemesterCode | String | Mã học kỳ cha |
| SemesterName | String | Tên học kỳ |
| SemesterStatus | Int32 | Trạng thái học kỳ |

* SlotTimeline

|  |  |  |
| --- | --- | --- |
| Trường | Kiểu dữ liệu | Mô tả |
| \_id | ObjectId | Định danh lịch học |
| BuildingCode | String | Mã toà nhà |
| CourseClassCode | String | Mã lớp học phần |
| CreatedAt | ISODate | Ngày tạo |
| DayOfWeek | Int32 | Ngày trong tuần (1-7) |
| EndWeek | Int32 | Tuần kết thúc |
| RoomCode | String | Mã phòng học |
| Slots | Array | Các tiết học |
| StartWeek | Int32 | Tuần bắt đầu |
| UpdatedAt | String | Ngày cập nhật |

* Staff

|  |  |  |
| --- | --- | --- |
| Trường | Kiểu dữ liệu | Mô tả |
| \_id | ObjectId | Định danh giảng viên |
| Code | String | Mã giảng viên |
| CreatedAt | ISODate | Ngày tạo |
| DepartmentCode | String | Mã khoa |
| FullName | String | Họ tên |
| UpdatedAt | String | Ngày cập nhật |

* Student

|  |  |  |
| --- | --- | --- |
| Trường | Kiểu dữ liệu | Mô tả |
| \_id | ObjectId | Định danh sinh viên |
| CreatedAt | ISODate | Ngày tạo |
| EducationPrograms | list | Danh sách chương trình đã học |
| InformationBySchool | Object | Thông tin theo nhà trường |
| PersonalInformation | Object | Thông tin cá nhân |
| Status | Int32 | Trạng thái sinh viên |
| UpdatedAt | String | Ngày cập nhật |

* StudentSemester

|  |  |  |
| --- | --- | --- |
| Trường | Kiểu dữ liệu | Mô tả |
| \_id | ObjectId | Định danh bản ghi |
| CourseSubjects | list | Danh sách học phần đã học |
| CreatedAt | ISODate | Ngày tạo |
| SemesterCode | String | Mã học kỳ |
| StudentCode | String | Mã sinh viên |
| SubjectResults | list | Kết quả từng học phần |
| UpdatedAt | String | Ngày cập nhật |

* Subject

|  |  |  |
| --- | --- | --- |
| Trường | Kiểu dữ liệu | Mô tả |
| \_id | ObjectId | Định danh học phần |
| CreatedAt | ISODate | Ngày tạo |
| DepartmentCode | String | Mã khoa quản lý |
| IsCalculateMark | Boolean | Có tính điểm hay không |
| NumberOfCredits | Int32 | Số tín chỉ |
| Status | Int32 | Trạng thái hoạt động |
| SubjectCode | String | Mã học phần |
| UpdatedAt | String | Ngày cập nhật |
| SubjectDescription | String | Mô tả chi tiết |
| SubjectName | String | Tên học phần |
| SubjectNameEng | String | Tên tiếng Anh |

* SubjectScheduleConfig

|  |  |  |
| --- | --- | --- |
| Trường | Kiểu dữ liệu | Mô tả |
| \_id | ObjectId | Định danh cấu hình |
| CreatedAt | ISODate | Ngày tạo |
| LabRequiredConditions | list | Điều kiện phòng lab |
| SemesterCode | String | Mã học kỳ |
| SessionPriority | Int32 | Mức ưu tiên xếp lịch |
| Stage | Int32 | Giai đoạn mở lớp |
| SubjectCode | String | Mã môn học |
| TheorySessions | Array | Các buổi lý thuyết |
| TotalTheoryCourseClass | Int32 | Tổng lớp lý thuyết |
| UpdatedAt | String | Ngày cập nhật |
| WeekLabEnd | Int32 | Tuần kết thúc lab |
| LectureRequiredConditions | Array | Điều kiện phòng lý thuyết |
| PracticeSessions | list | Buổi thực hành |
| TheoryTotalPeriod | Int32 | Tổng tiết lý thuyết |
| WeekLabStart | Int32 | Tuần bắt đầu lab |
| PracticeTotalPeriod | Int32 | Tổng tiết thực hành |
| WeekLectureEnd | Int32 | Tuần kết thúc lý thuyết |
| WeekLectureStart | Int32 | Tuần bắt đầu lý thuyết |

### 2.5.7 Rằng buộc tham chiếu

* + **CourseClass** (SemesterCode) -> **Semester** (SemesterCode)
  + **CourseClass** (SubjectCode) -> **Subject** (SubjectCode)
  + **CourseClass** (TeacherCode) -> **Staff** (Code)
  + **CourseClass** (ParentCourseClassCode) -> **CourseClass** (CourseClassCode)
  + **Register** (SemesterCode) -> **Semester** (SemesterCode)
  + **Semester** (ParentSemesterCode) -> **Semester** (SemesterCode)
  + **SlotTimeline** (CourseClassCode) -> **CourseClass** (CourseClassCode)
  + **SlotTimeline** (RoomCode) -> **Room** (Code)
  + **Staff** (DepartmentCode) -> **Department** (DepartmentCode)
  + **Student** (EducationPrograms.Code) -> **EducationProgram** (Code)
  + **Student** (EducationPrograms.DepartmentCode) -> **Department** (DepartmentCode)
  + **StudentSemester** (SemesterCode) -> **Semester** (SemesterCode)
  + **StudentSemester** (StudentCode) -> **Student** (InformationBySchool.StudentCode)
  + **StudentSemester** (SubjectResults.SubjectCode) -> **Subject** (SubjectCode)
  + **Subject** (DepartmentCode) -> **Department** (DepartmentCode)
  + **SubjectScheduleConfig** (SemesterCode) -> **Semester** (SemesterCode)
  + **SubjectScheduleConfig** (SubjectCode) -> **Subject** (SubjectCode)

## 2.6. Thiết kế api

### 2.6.1 Api laays danh sách phòng học

* URL: /api/Building/Room
* Phương thức: **GET**
* Response:
  + **capacity**: Sức chứa của phòng (số sinh viên).
  + **code**: Mã phòng.
  + **name**: Tên phòng.
  + **buildingCode**: Mã tòa nhà.
  + **supportedConditions**: Mảng các điều kiện hỗ trợ của phòng (ví dụ: "Lecture", "Lab", ...).
  + **id**: Định danh duy nhất của phòng.
  + **createdAt**: Thời điểm tạo phòng (ISO 8601).
  + **updatedAt**: Thời điểm cập nhật gần nhất (nếu có).

### 2.6.2 Api lấy danh sách phòng học trống

URL: **/api/Building/Room/Free**

* Phương thức: **GET**
* Response:
  + **capacity**: Sức chứa của phòng (số sinh viên).
  + **code**: Mã phòng.
  + **name**: Tên phòng.
  + **buildingCode**: Mã tòa nhà.
  + **supportedConditions**: Mảng các điều kiện hỗ trợ của phòng (ví dụ: "Lecture", "Lab", ...).
  + **id**: Định danh duy nhất của phòng.
  + **createdAt**: Thời điểm tạo phòng (ISO 8601).
  + **updatedAt**: Thời điểm cập nhật gần nhất (nếu có).

2.6.3 Lấy danh sách lớp học

* URL: **/api/CourseClass**
* Phương thức: **GET**
* Response:
  + **index**: Vị trí hoặc thứ tự của lớp học trong danh sách.
  + **weekStart**: Tuần bắt đầu của lớp học (số nguyên).
  + **weekEnd**: Tuần kết thúc của lớp học (số nguyên).
  + **status**: Trạng thái lớp học (ví dụ: 0 = mặc định, 1 = đã khóa,...).
  + courseClassCode: Mã lớp học.
  + courseClassName: Tên lớp học.
  + **studentIds**: Mảng mã sinh viên đã đăng ký lớp học.
  + **courseClassType**: Loại lớp học (0 = lý thuyết, 1 = thực hành, ...).
  + **subjectCode**: Mã môn học.
  + **sessionLengths**: Mảng số tiết cho mỗi buổi học (ví dụ: [2, 2] nghĩa là mỗi buổi 2 tiết).
  + **totalSession**: Tổng số buổi học.
  + **semesterCode**: Mã học kỳ (có thể null nếu chưa gán kỳ học).
  + **numberStudents**: Số sinh viên hiện tại trong lớp.
  + **numberStudentsExpected**: Số sinh viên dự kiến tối đa.
  + **stage**: Giai đoạn lớp học (0, 1, 2,... - tùy theo nghiệp vụ).
  + **teacherName**: Tên giảng viên (có thể null nếu chưa phân công).
  + **teacherCode**: Mã giảng viên (có thể null nếu chưa phân công).
  + **parentCourseClassCode**: Mã lớp học cha (nếu có, dùng cho lớp học chia nhóm).
  + **id**: Định danh duy nhất của lớp học.
  + **createdAt**: Thời điểm tạo lớp học (ISO 8601).
  + **updatedAt**: Thời điểm cập nhật gần nhất (nếu có).

2.6.4 Tạo lớp học phần môn học

* URL: **/api/CourseClass**
* Phương thức: **POST**
* **Body:**
  + **courseClassCode**: Mã lớp học (kiểu chuỗi).
  + **courseClassName**: Tên lớp học (kiểu chuỗi).
  + **courseClassType**: Loại lớp học (số nguyên, ví dụ: 0 = lý thuyết, 1 = thực hành,...).
  + **subjectCode**: Mã môn học (kiểu chuỗi).
  + **semesterCode**: Mã học kỳ (kiểu chuỗi).
  + **numberStudentsExpected**: Số lượng sinh viên dự kiến (số nguyên).
  + **parentCourseClassCode**: Mã lớp học cha (nếu có, kiểu chuỗi, dùng cho lớp học chia nhóm).
  + **stage**: Giai đoạn lớp học (số nguyên).
  + **weekStart**: Tuần bắt đầu (số nguyên).
  + **weekEnd**: Tuần kết thúc (số nguyên).
  + **sessionLengths**: Mảng số tiết cho mỗi buổi học (mỗi phần tử là số nguyên, ví dụ: [2, 3] nghĩa là các buổi 2 tiết, 3 tiết,...).
* Response:
  + **index**: Vị trí hoặc thứ tự của lớp học trong danh sách.
  + **weekStart**: Tuần bắt đầu của lớp học (số nguyên).
  + **weekEnd**: Tuần kết thúc của lớp học (số nguyên).
  + **status**: Trạng thái lớp học (ví dụ: 0 = mặc định, 1 = đã khóa,...).
  + courseClassCode: Mã lớp học.
  + courseClassName: Tên lớp học.
  + **studentIds**: Mảng mã sinh viên đã đăng ký lớp học.
  + **courseClassType**: Loại lớp học (0 = lý thuyết, 1 = thực hành, ...).
  + **subjectCode**: Mã môn học.
  + **sessionLengths**: Mảng số tiết cho mỗi buổi học (ví dụ: [2, 2] nghĩa là mỗi buổi 2 tiết).
  + **totalSession**: Tổng số buổi học.
  + **semesterCode**: Mã học kỳ (có thể null nếu chưa gán kỳ học).
  + **numberStudents**: Số sinh viên hiện tại trong lớp.
  + **numberStudentsExpected**: Số sinh viên dự kiến tối đa.
  + **stage**: Giai đoạn lớp học (0, 1, 2,... - tùy theo nghiệp vụ).
  + **teacherName**: Tên giảng viên (có thể null nếu chưa phân công).
  + **teacherCode**: Mã giảng viên (có thể null nếu chưa phân công).
  + **parentCourseClassCode**: Mã lớp học cha (nếu có, dùng cho lớp học chia nhóm).
  + **id**: Định danh duy nhất của lớp học.
  + **createdAt**: Thời điểm tạo lớp học (ISO 8601).
  + **updatedAt**: Thời điểm cập nhật gần nhất (nếu có).

2.6.5 API chạy tự động thời khóa biểu

* URL: **/api/CourseClass/GenerateSchedule**
* Phương thức: **POST**
* **Body:**
  + **semesterCode**: Mã học kỳ (kiểu chuỗi).
  + **stage**: Giai đoạn (số nguyên).
  + **subjectCode**: Mã môn học (kiểu chuỗi).
  + **courseClassCodes**: Mảng mã lớp học cần chạy thời khóa biểu (danh sách chuỗi).
* Response:

2.6.6 API tạo lớp học tự động

* URL: **/api/CourseClass/GenerateCourseClasses**
* Phương thức: **POST**
* **Body:**
  + **semesterCode**: Mã học kỳ (kiểu chuỗi).
  + **subjectCode**: Mã môn học (kiểu chuỗi).
  + **stage**: Giai đoạn (số nguyên).
  + **totalTheoryCourseClass**: Tổng số lớp lý thuyết cần tạo (số nguyên).
  + **numberStudentsExpected**: Số sinh viên dự kiến cho mỗi lớp (số nguyên).

**2**.6.7 API tạo kế hoạch lập thời khóa biểu lớp học

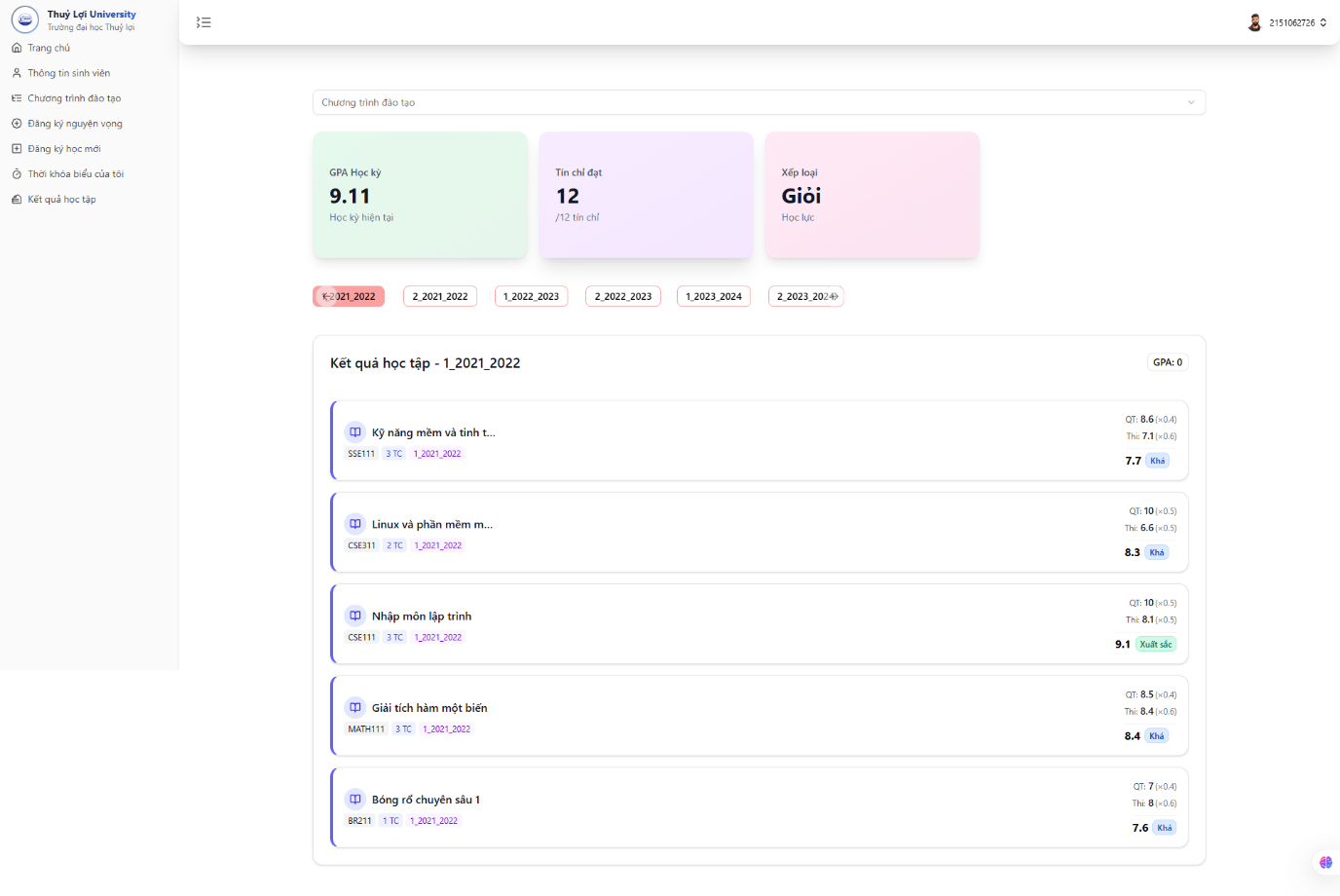
* URL: **/api/CourseClass/SubjectScheduleConfig**
* Phương thức: **POST**
* **Body:**
  + **subjectCode**: Mã môn học (kiểu chuỗi).
  + **stage**: Giai đoạn (số nguyên).
  + **theoryTotalPeriod**: Tổng số tiết lý thuyết (số nguyên).
  + **practiceTotalPeriod**: Tổng số tiết thực hành (số nguyên).
  + **theorySessions**: Mảng số tiết cho từng buổi lý thuyết (mỗi phần tử là số nguyên).
  + **practiceSessions**: Mảng số tiết cho từng buổi thực hành (mỗi phần tử là số nguyên).
  + **weekLectureStart**: Tuần bắt đầu dạy lý thuyết (số nguyên).
  + **weekLectureEnd**: Tuần kết thúc dạy lý thuyết (số nguyên).
  + **weekLabStart**: Tuần bắt đầu dạy thực hành (số nguyên).
  + **weekLabEnd**: Tuần kết thúc dạy thực hành (số nguyên).
  + **sessionPriority**: Ưu tiên sắp xếp lịch (số nguyên, phụ thuộc vào nghiệp vụ).
  + **lectureRequiredConditions**: Mảng điều kiện yêu cầu cho tiết lý thuyết (ví dụ: "Lecture", "Projector",...).
  + **labRequiredConditions**: Mảng điều kiện yêu cầu cho tiết thực hành (ví dụ: "Lab", "Computer",...).

2.6.8 Lấy thời khóa biểu lớp học

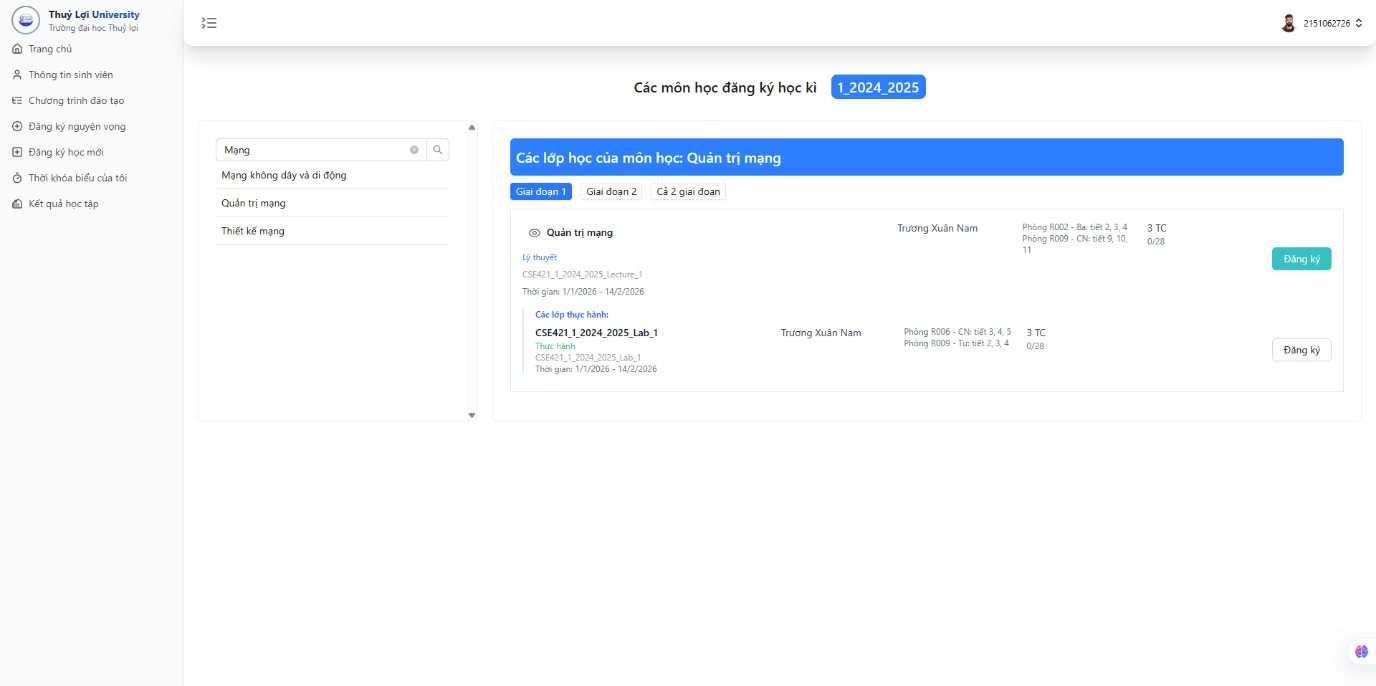
* URL: **/api/CourseClass/Schedule**
* Phương thức: **GET**
* **Response:**
  + courseClassCode: Mã lớp học.
  + **buildingCode**: Mã tòa nhà (có thể null nếu chỉ có mã phòng).
  + **startWeek**: Tuần bắt đầu sử dụng phòng (số nguyên).
  + **endWeek**: Tuần kết thúc sử dụng phòng (số nguyên).
  + **roomCode**: Mã phòng học.
  + **dayOfWeek**: Thứ trong tuần (0 = Chủ nhật, 1 = Thứ hai, ...).
  + **slots**: Mảng các tiết học trong ngày (dạng chuỗi, ví dụ: ["0", "1"]).
  + **id**: Định danh duy nhất của lịch phòng học.
  + **createdAt**: Thời điểm tạo (ISO 8601).
  + **updatedAt**: Thời điểm cập nhật gần nhất (nếu có).

## 2.7 THIẾT KẾ GIAO DIỆN

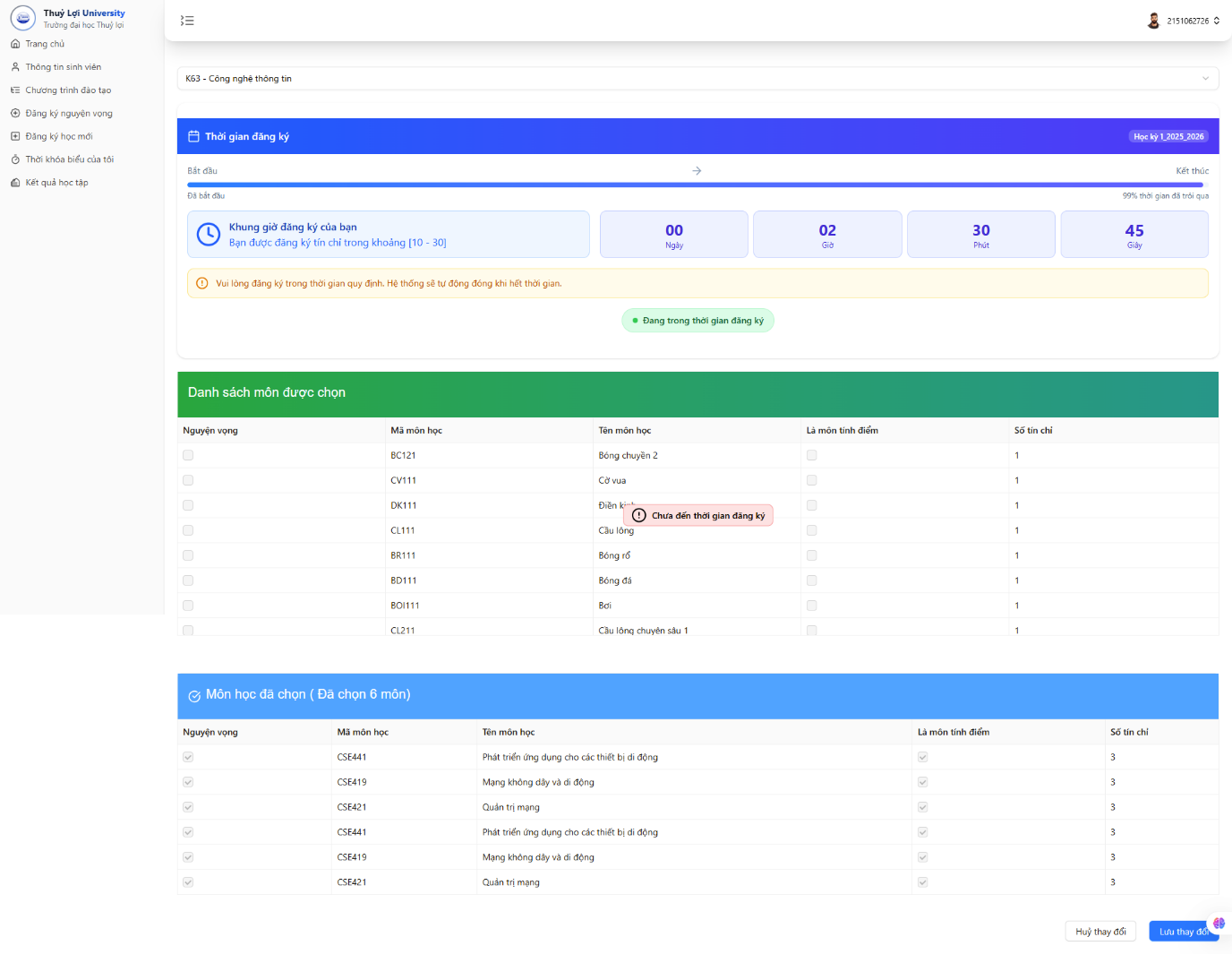
### 2.7.1 Giao diện xem kết quả học tập



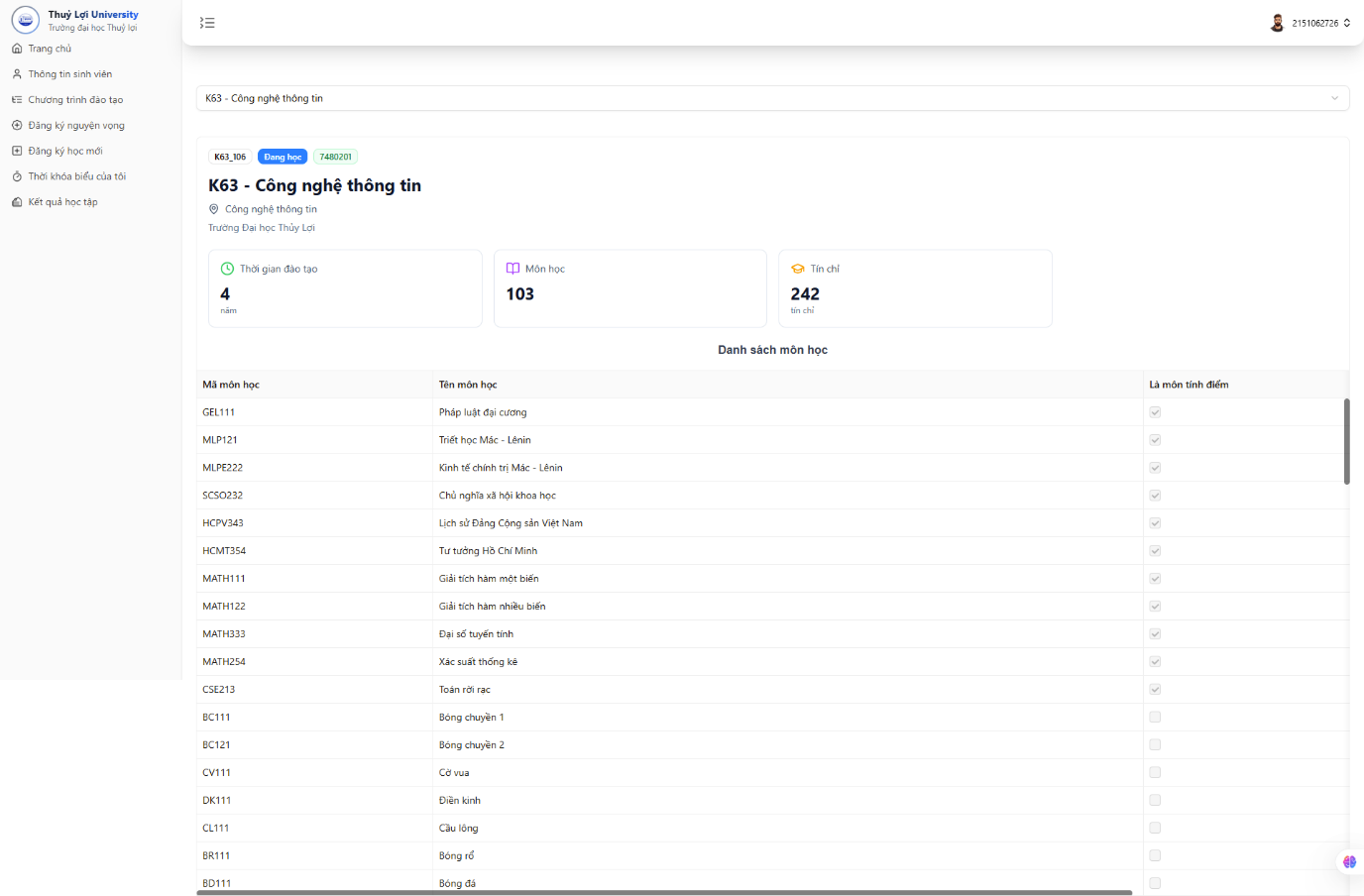
### 2.7.2 Giao diện đăng ký lớp học phần



### 2.7.3 Giao diện đăng ký nguyện vọng học



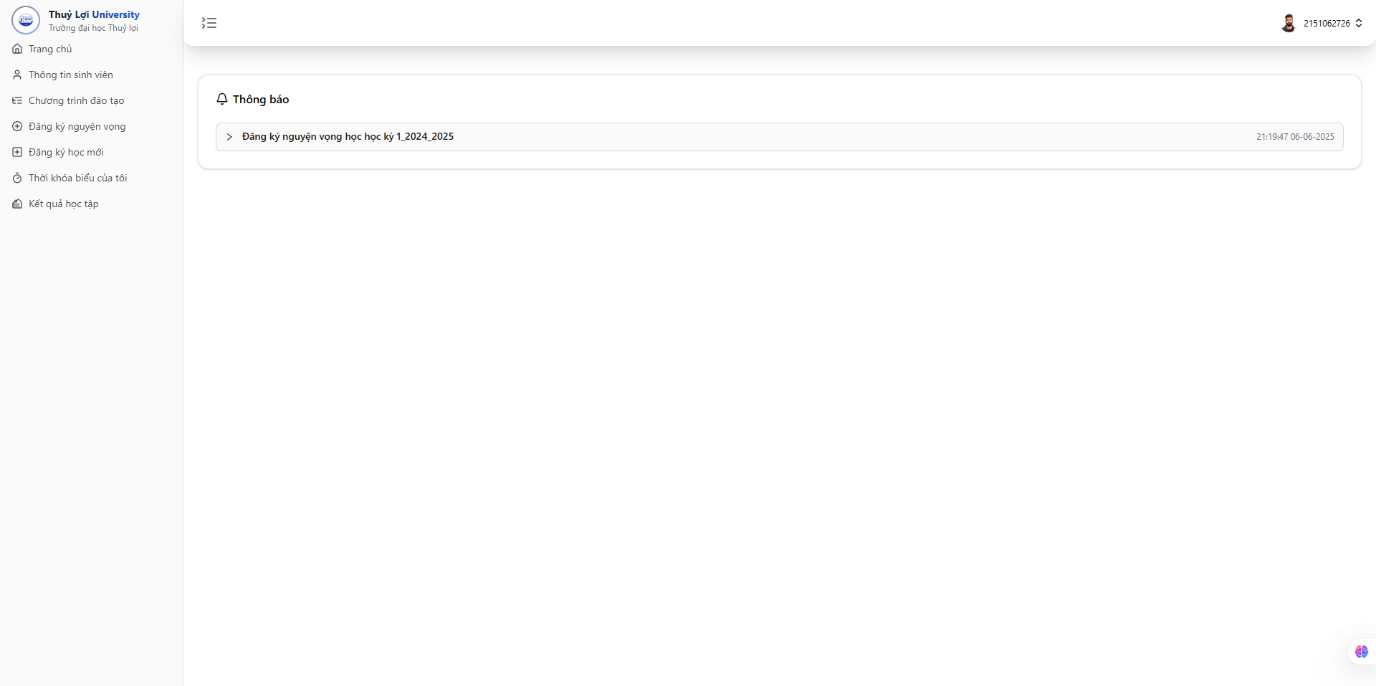
### 2.7.4 Giao diện xem thông tin chương trình đào tạo



### 2.7.5 Giao diện xem thông tin cá nhân



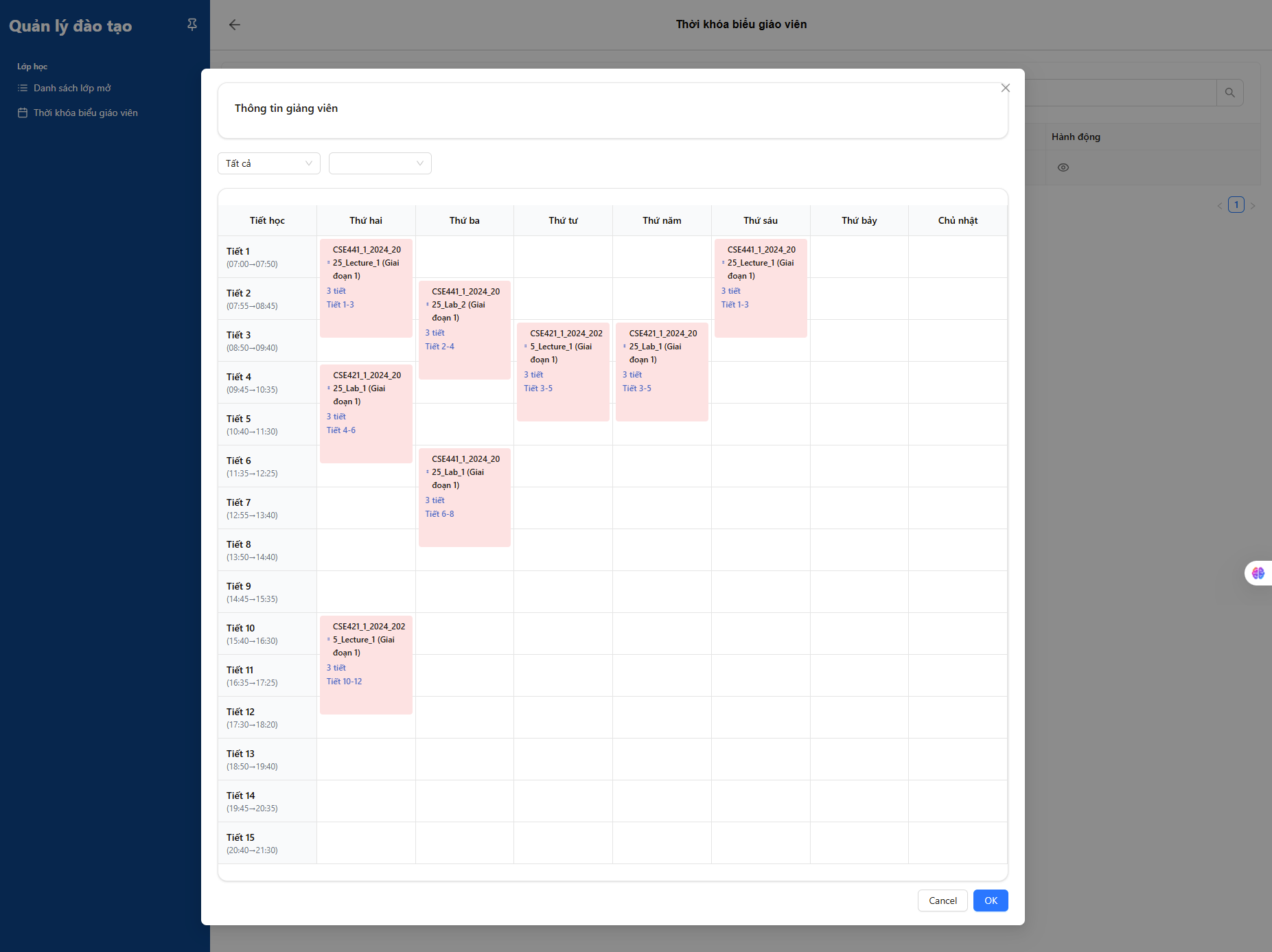
### 2.7.6 Giao diện trang chủ sinh viên



### 2.7.8 Giao diện đăng nhập sinh viên



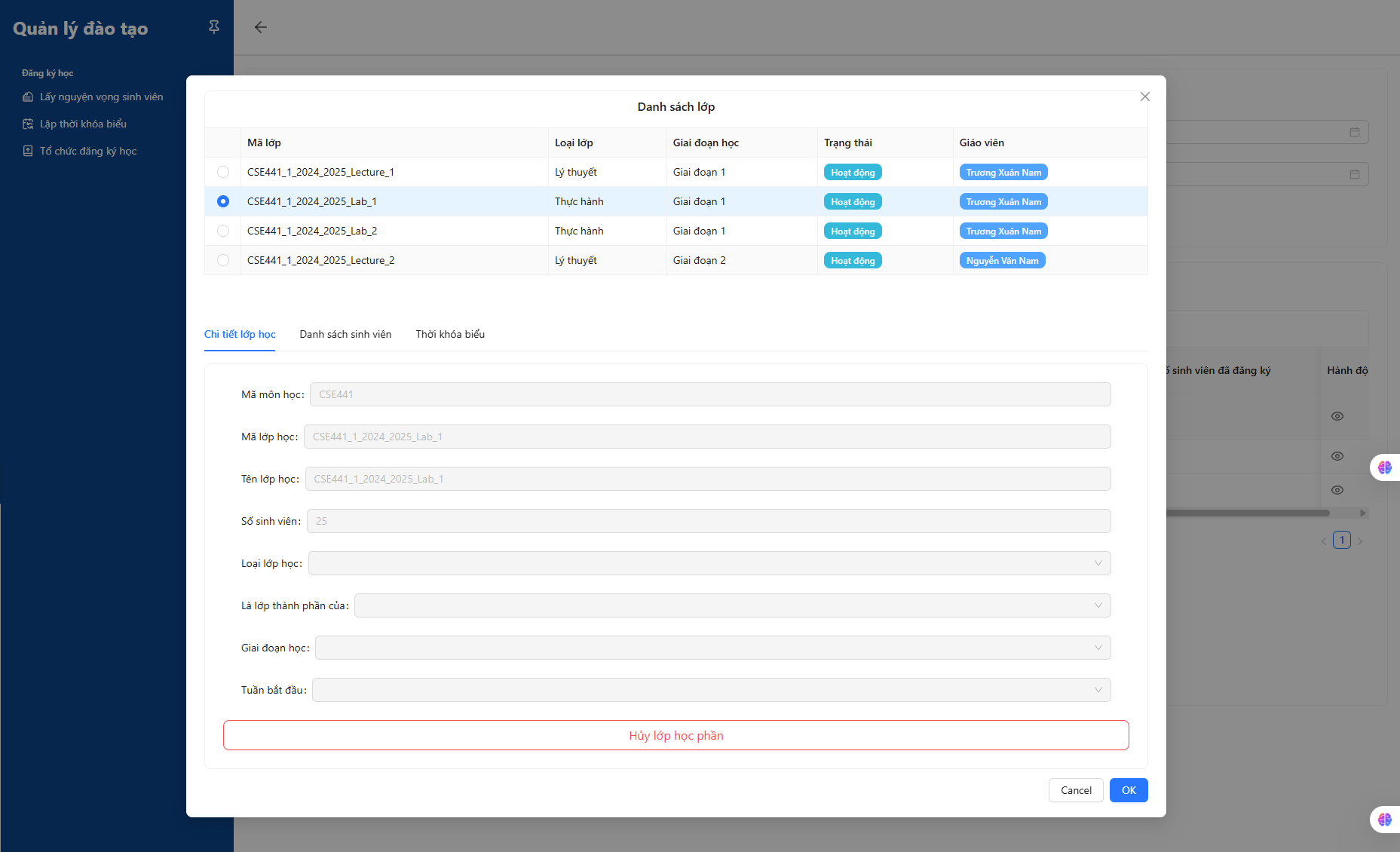
### 2.7.9 Giao diện xem thời khóa biểu giáo viên



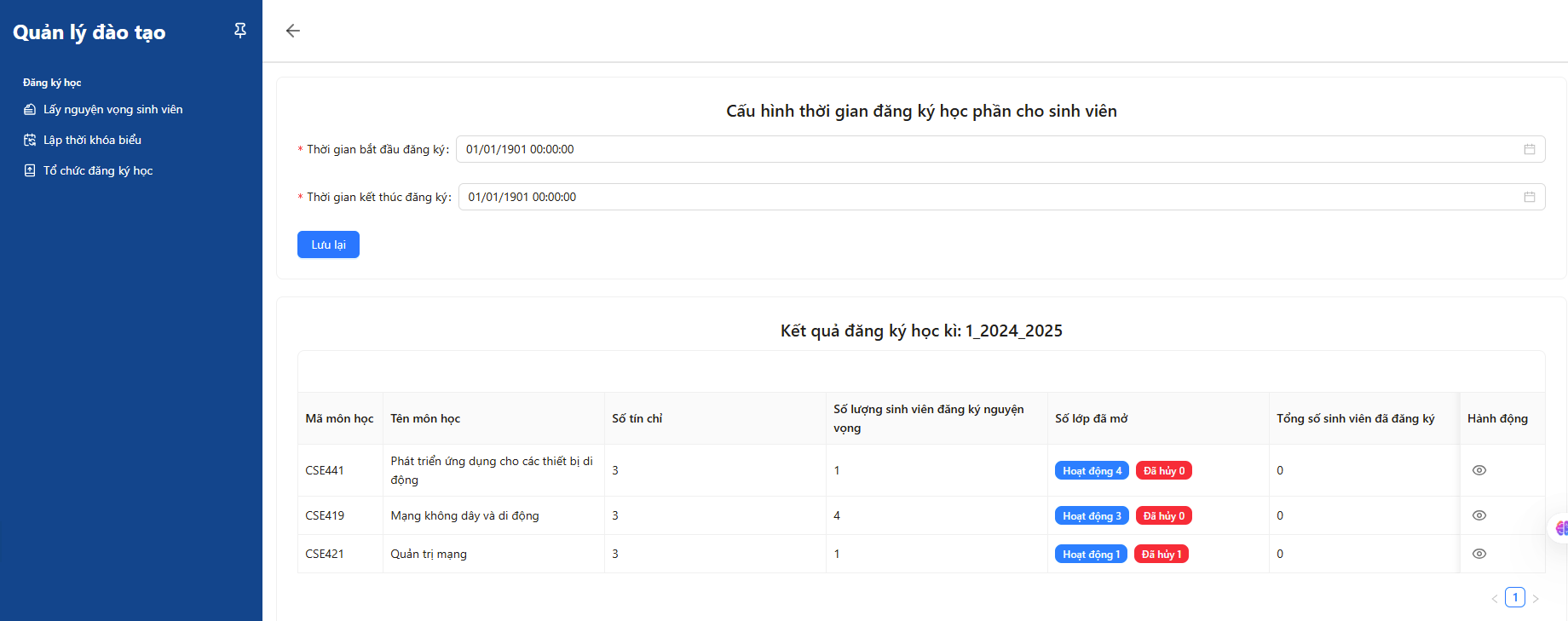
### 2.7.10 Giao diện phân công giáo viên cho lớp học phần



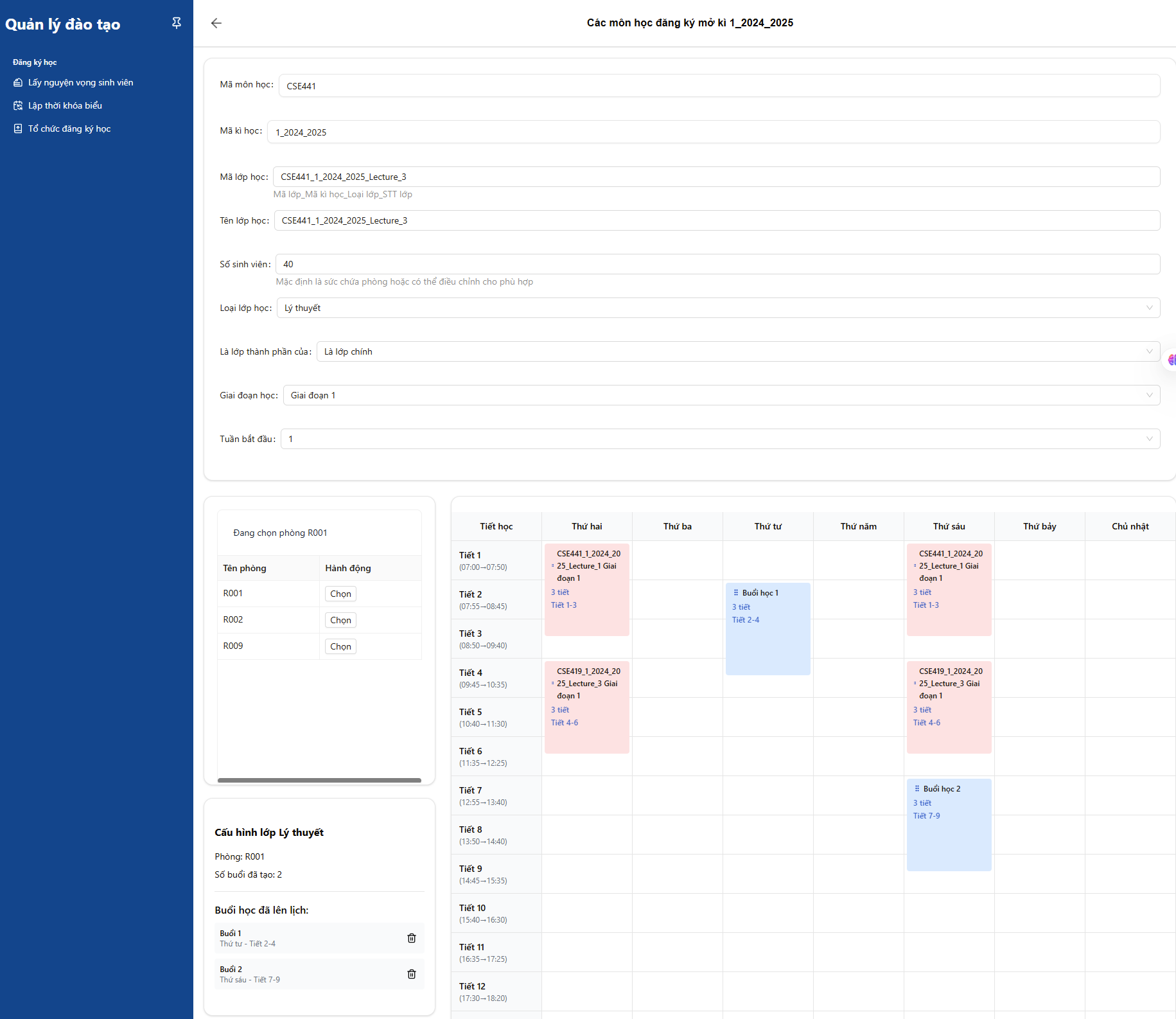
### 2.7.14 Giao diện quản lý lớp học phần



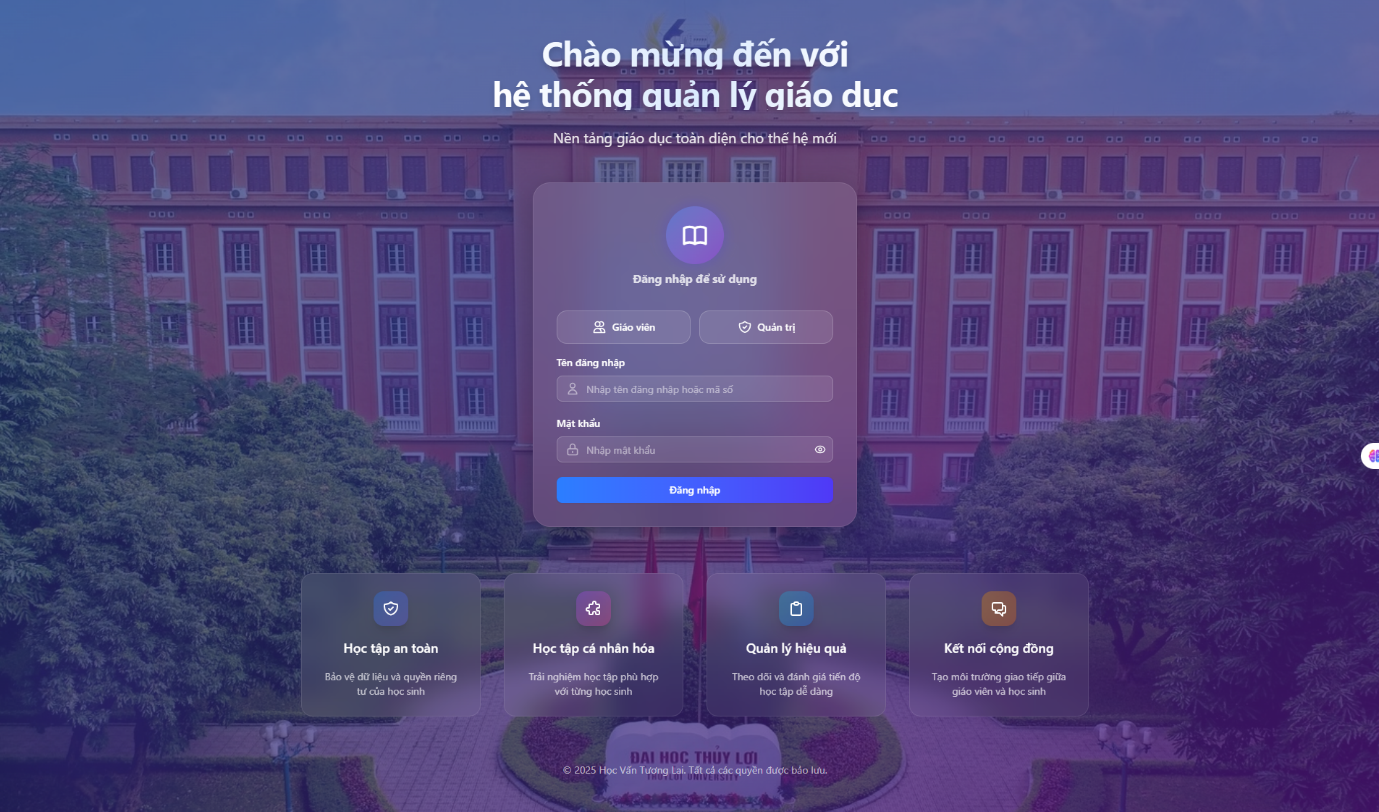
### 2.7.15 Giao diện cấu hình đăng ký học phần



### 2.7.16 Giao diện tạo mới lớp học phần

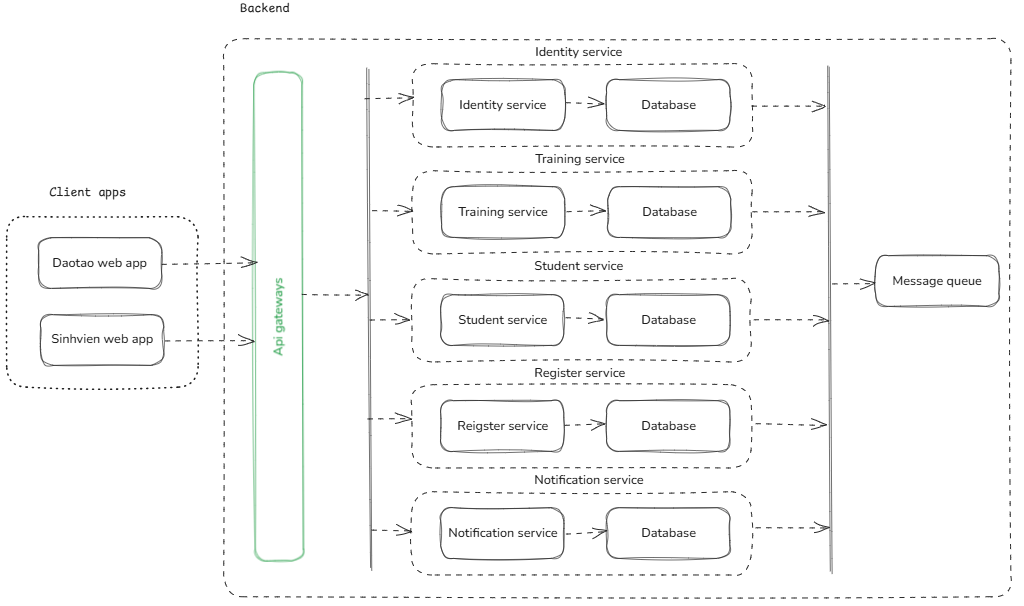


### 2.7.17 Giao diện đăng nhập của PĐT



# CHƯƠNG 3 TRIỂN KHAI

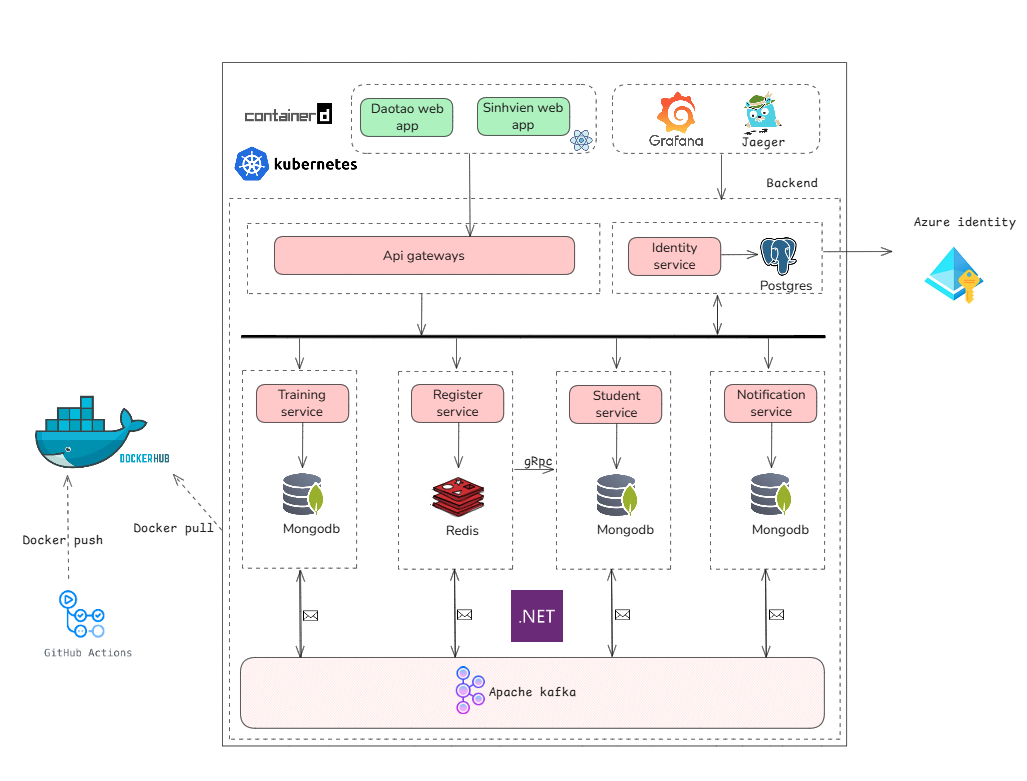
## 3.1 Sơ đồ thành phần



Chi tiết các thành phần:

* **Client apps:** 
  + **Daotao web app:** Web app dành cho PĐT và GV.
  + **Sinhvien web app:** Web app dành cho sinh viên.
* **API Gateways:**  đóng vai trò điều hướng các API request từ người dùng đến các service.
* **Identity Service:** Đóng vai trò xác thực và phân quyền người dùng trên hệ thống.
* **Training Service:** Đóng vài trò xử lý thông tin môn học và cấu hình lớp học phần môn học.
* **Student Service**: Đóng vai trò xử lý các tác vụ của sinh viên.
* **Register Service**: Đóng vai trò xử lý tác vụ đăng ký học.
* **Notification Service**: Đóng vai trò xử lý các tác vụ thông báo.

## 3.2 Sơ đồ triển khai



* Hạ tầng triển khai:
  + **Kubernetes + containerd**: Toàn bộ hệ thống chạy trên Kubernetes với containerd là container runtime.
  + **DockerHub**: Hình ảnh container được đẩy lên từ GitHub Actions và được Kubernetes pull về từ DockerHub để triển khai.
* CI/CD Pipeline:
  + **GitHub Actions**: Tự động build và đẩy image Docker lên DockerHub.
  + **Docker push/pull**: CI/CD đẩy hình ảnh lên DockerHub và Kubernetes sẽ pull về khi triển khai.
* Frontend:
  + **Daotao web app** và **Sinhvien web app**: Giao diện người dùng, chạy bên ngoài và giao tiếp với backend thông qua API Gateway.
* API Gateway và Identity:
  + **API Gateway**: Là điểm vào của tất cả các request, chuyển tiếp đến các dịch vụ tương ứng.
  + **Identity Service**: Dịch vụ xác thực, kết nối tới **PostgreSQL** để lưu trữ thông tin người dùng, đồng thời tích hợp với **Azure Identity** (AAD - Azure Active Directory).
* Các dịch vụ backend:
  + **Training Service**: Quản lý các tác vụ liên quan đến đào tạo, sử dụng **MongoDB** để lưu trữ dữ liệu.
  + **Register Service**: Xử lý các yêu cầu đăng ký, sử dụng **Redis** để lưu trữ dữ liệu tạm thời và hỗ trợ giao tiếp gRPC với các dịch vụ khác.
  + **Student Service**: Quản lý thông tin sinh viên, lưu trữ trên **MongoDB**.
  + **Notification Service**: Gửi thông báo, cũng sử dụng **MongoDB** để lưu trữ dữ liệu thông báo.
  + Các dịch vụ này giao tiếp với nhau thông qua **Apache Kafka**, một hệ thống hàng đợi tin nhắn, để xử lý các sự kiện bất đồng bộ.
* Quan sát hệ thống:
  + **Grafana**: Theo dõi metrics, giám sát hiệu năng hệ thống.
  + **Jaeger**: Theo dõi trace/phân tích đường đi của request trong hệ thống (distributed tracing).

# CHƯƠNG 4: Hạn chế và phương hướng phát triển