# 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

### **1.4.1 Kiến trúc hướng sự kiện (EDA - Event-Driven Architecture)**

Kiến trúc hướng sự kiện (Event-Driven Architecture – EDA) là một mô hình thiết kế hệ thống trong đó các thành phần giao tiếp với nhau thông qua việc phát sinh, truyền và xử lý các sự kiện. Khác với mô hình tuần tự truyền thống, nơi các thành phần gọi trực tiếp hàm hoặc API của nhau, EDA cho phép các thành phần phát sự kiện (emit) mà không cần biết ai sẽ xử lý chúng. Sự kiện có thể là bất kỳ hành động nào phát sinh trong hệ thống như “sinh viên đăng ký lớp học”, “lớp học phần đầy”, “sinh viên huỷ đăng ký”, hoặc “giảng viên bị thay đổi”.  
  
EDA giúp hệ thống hoạt động theo cơ chế phản ứng (reactive), cho phép xử lý sự kiện ngay khi chúng xảy ra mà không cần chờ đợi luồng đồng bộ. Trong hệ thống đăng ký học phần tại Trường Đại học Thủy Lợi, EDA đặc biệt hữu ích trong việc đảm bảo tính mở rộng và phân tách rõ ràng các nghiệp vụ giữa các microservices.  
  
Một kiến trúc EDA cơ bản bao gồm ba thành phần chính:  
- \*\*Event Producer (Nhà phát sinh sự kiện)\*\*: Các thành phần tạo ra sự kiện. Ví dụ, khi sinh viên hoàn thành bước đăng ký lớp học phần, Register Service sẽ phát đi một sự kiện “ClassRegistered”.  
- \*\*Event Broker (Bộ phân phối sự kiện)\*\*: Là cầu nối trung gian chịu trách nhiệm tiếp nhận, lưu trữ và chuyển tiếp sự kiện đến các thành phần đăng ký nhận (subscriber). Trong hệ thống này, Apache Kafka được sử dụng như một Event Broker nhờ khả năng xử lý bất đồng bộ với thông lượng cao, đảm bảo khả năng mở rộng và độ bền vững.  
- \*\*Event Consumer (Thành phần tiêu thụ sự kiện)\*\*: Các dịch vụ lắng nghe và xử lý sự kiện. Ví dụ, Notification Service sẽ lắng nghe sự kiện “ClassRegistered” để gửi email xác nhận; Analytics Service sẽ cập nhật thống kê số lượng đăng ký.  
  
Lợi ích của việc áp dụng kiến trúc EDA trong hệ thống:  
- \*\*Tăng khả năng mở rộng\*\*: Các sự kiện có thể được xử lý song song bởi nhiều dịch vụ khác nhau, phù hợp với hệ thống có lượng truy cập lớn.  
- \*\*Giảm sự phụ thuộc\*\*: Các thành phần không cần biết lẫn nhau, từ đó tăng tính module hóa và dễ bảo trì.  
- \*\*Phản ứng thời gian thực\*\*: Các sự kiện được xử lý gần như ngay lập tức, tăng tính tức thời của hệ thống.  
- \*\*Hỗ trợ dễ dàng cho logging và auditing\*\*: Lưu vết tất cả các sự kiện để phục vụ kiểm tra, giám sát và phân tích hành vi người dùng.  
  
Một số tình huống thực tiễn trong đề tài áp dụng EDA:  
- Khi sinh viên huỷ đăng ký học phần, hệ thống sẽ phát sinh sự kiện “ClassUnregistered”. Hệ thống có thể cập nhật số lượng ghế trống, gửi lại email xác nhận, hoặc kích hoạt gợi ý học phần khác.  
- Khi PĐT mở lớp học phần mới, sự kiện “CourseClassCreated” được phát đi để cập nhật lịch học, gửi thông báo cho sinh viên và giảng viên liên quan.  
- Khi giảng viên được phân công, hệ thống phát sự kiện “TeacherAssignedToClass” để các dịch vụ như Room Service và Timetable Service đồng bộ lịch và kiểm tra xung đột.  
- Khi kết thúc đăng ký, Event “RegisteringPeriodClosed” sẽ kích hoạt tổng hợp thống kê, phân lớp tự động hoặc gửi thông báo tới giảng viên.  
  
Ngoài Kafka, hệ thống có thể tích hợp thêm các mô hình EDA tiên tiến hơn như event sourcing (lưu toàn bộ sự kiện để tái hiện trạng thái), CQRS (phân tách read/write logic), giúp hệ thống đảm bảo tính nhất quán eventual consistency và dễ dàng phân tích hành vi trong tương lai.  
  
Tóm lại, kiến trúc EDA mang đến một hướng tiếp cận hiện đại, phù hợp với xu hướng phát triển hệ thống phân tán quy mô lớn. Việc áp dụng EDA trong hệ thống đăng ký học phần không chỉ giúp hệ thống phản ứng linh hoạt với các thay đổi mà còn tăng cường khả năng mở rộng, bảo trì và tích hợp với các công nghệ phân tích dữ liệu và AI trong tương lai.

### 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

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Actor | Công việc | Ghi chú |
|  | SV | Đăng nhập với tài khoản Outlook |  |
|  | Lấy thông tin cá nhân |  |
|  | Lấy thông tin kết quả học tập |  |
|  | Lấy thông tin chương trình đào tạo |  |
|  | Đăng ký nguyện vọng môn học |  |
|  | Đăng ký học phần |  |
|  | Lấy thông tin thời khóa biểu học phần |  |
|  | PĐT | Tạo kì học mới |  |
|  | Cấu hình môn học |  |
|  | Tạo đợt đăng ký nguyện vọng học cho sinh viên |  |
|  | Báo cáo đăng ký nguyện vọng học |  |
|  | Mở lớp học phần |  |
|  | Tổ chức đăng ký học phần |  |
|  | Thay đổi sinh viên đăng ký học |  |
|  | Hủy lớp học phần |  |
|  | Xem danh sách lớp học phần |  |
|  | PĐT, TBM | Đăng nhập với tài khoản và mật khẩu |  |
|  | TBM | Phân công giảng viên cho lớp học phần |  |
|  |  | Xem thời khóa biểu giáo viên |  |

## 2.2 Use Case tổng quát

## 2.3 Đặc tả use case

### 2.3.1 **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 | 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 Tạo kì học mới

|  |  |  |
| --- | --- | --- |
| **Use case** | **Tạo học kì mới** | |
| Mục tiêu | Cho phép PĐT có thể tạo mới một học kỳ trong hệ thống để chuẩn bị cho việc đăng ký học phần. | |
| Điều kiện tiên quyết | Người thực hiện đã đăng nhập và có quyền quản trị hệ thống. | |
| Điều kiện kết thúc thành công | Hệ thống thông báo tạo học kì mới thành công và học kỳ mới hiển thị đầy đủ thông tin (mã học kỳ, tên học kỳ, thời gian bắt đầu/kết thúc, trạng thái). | |
| Phân quyền Actors | PĐT | |
| Kích hoạt | PĐT nhấn vào nút “Tạo học kì mới”. | |
| **Luồng sự kiện chính** | **Bước** | **Hành động của tác nhân** |
|  |  | PĐT nhấp vào nút “Tạo học kì mới” trên trang danh sách học kì. |
|  |  | Hệ thống hiển thị form tạo kì học kì mới bao gồm tên học kì, mã học kì, thời gian bắt đầu học kì, thời gian kết thúc học kì và các giai đoạn thời gian của học kì. |
|  |  | PĐT nhập đẩy đủ thông tin vào form và nhấp nút “Xác nhận”. |
|  |  | Hệ thống kiểm tra:   * Tên/mã học kì đã trùng với học kì đã có sẵn hay chưa * Thông tin ngày tháng hợp lệ hay chưa. |
|  |  | Hệ thống tạo mới học kì và lưu vào cơ sở dữ liệu. |
|  |  | Hệ thống thông báo tạo học kì mới thành công. |
| **Luồng ngoại lệ** | **Bước** | **Hành động phân nhánh** |
|  | 4 | 4a. Mã hoặc tên học kỳ bị trùng, hệ thống thông báo lỗi và yêu cầu chỉnh sửa lại. |
|  | 4b Thời gian người dùng nhập không hợp lệ, hệ thống thông báo lỗi và yêu cầu người dùng sửa lại. |

2.3.3 Cấu hình môn học

|  |  |  |
| --- | --- | --- |
| **Use case** | **Cấu hình môn học** | |
| Mục tiêu | Cho phép PĐT có thể chỉnh sửa thông tin môn học bao gồm các thông số để tạo lớp học phần. | |
| 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 môn học phản ánh đúng trên hệ thống. | |
| Phân quyền Actors | PĐT | |
| Kích hoạt | PĐT nhấp nhấp vào nút “Cấu hình môn học” trên trang danh sách môn học. | |
| **Luồng sự kiện chính** | **Bước** | **Hành động của tác nhân** |
|  | 1 | PĐT nhấp vào nút “Cấu hình môn học” trên trang danh sách học môn học. |
|  |  | Hệ thống hiển thị thông tin cơ bản của môn học và thông tin để cấu hình lớp học phần. |
|  | 2 | PĐT nhập đầy đủ thông tin cấu hình lớp học phần. |
|  | 3 | Hệ thống kiểm tra:   * Đã nhập đẩy đủ thông tin hay chưa. * Thông tin PĐT nhập đã hợp lệ hay chưa. |
|  | 4 | Hệ thống cập nhật thông tin cấu hình môn học và lưu vào cơ sở dữ liệu. |
|  | 5 | Hệ thống thông báo cập nhật cấu hình môn học thành công. |
| **Luồng ngoại lệ** | **Bước** | **Hành động phân nhánh** |
|  | 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. |

2.3.2 **Lấy thông tin cá nhân**

|  |  |  |
| --- | --- | --- |
| **Use case** | **Lấy thông tin cá nhân** | |
| Mục tiêu | Sinh viên xem thông tin cá nhân trên hệ thống. | |
| Điều kiện tiên quyết | SV đã đăng nhập hệ thống. | |
| Điều kiện kết thúc thành công | Hiển thị thông tin cá nhân đầy đủ. | |
| Phân quyền Actors | SV | |
| Kích hoạt | SV chọn chức năng thông tin cá nhân. | |
| **Luồng sự kiện chính** | **Bước** | **Hành động của tác nhân** |
|  |  | Hệ thống xác thực người dùng và truy xuất dữ liệu thông tin cá nhân của SV tương ứng trong cơ sở dữ liệu. |
|  |  | Hệ thống hiển thị đầy đủ thông tin cá nhân trên giao diện. |
| **Luồng ngoại lệ** | **Bước** | **Hành động phân nhánh** |
|  | 2 | 2a. Không có dữ liệu, thông báo hiển thị thông báo tương ứng. |

### 2.3.3 **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. |

### 2.3.4 Lấy thông tin chương trình đào tạo

|  |  |  |
| --- | --- | --- |
| **Use case** | Lấy thông tin chương trình đào tạo**.** | |
| Mục tiêu | Xem chi tiết các học phần, tín chỉ trong CTĐT. | |
| Điều kiện tiên quyết | SV đã đăng nhập hệ thống. | |
| Điều kiện kết thúc thành công | Danh sách học phần và tín chỉ tương ứng được hiển thị đầy đủ. | |
| Phân quyền Actors | SV | |
| Kích hoạt | SV chọn chức năng chương trình đào tạo. | |
| **Luồng sự kiện chính** | **Bước** | **Hành động của tác nhân** |
|  |  | Hệ thống truy xuất CTĐT mà SV đó đang theo học và hiển thị. |
|  |  | Sinh viên chọn chương trình đào tạo muốn xem. |
|  |  | Hệ thống truy xuất dữ liệu danh sách học phần tương ứng với chương trình đào tạo đã chọn. |
|  |  | Hệ thống hiển thị danh sách các học phần tương ứng. |
| **Luồng ngoại lệ** | **Bước** | **Hành động phân nhánh** |
|  | 2 | 2a. Không có dữ liệu, thông báo lỗi tương ứng. |

### 2.3.5 Đăng ký nguyện vọng môn học

|  |  |  |
| --- | --- | --- |
| **Use case** | **Lấy thông tin chương trình đào tạo.** | |
| Mục tiêu | Cho phép SV chọn các môn muốn học trong học kỳ. | |
| Điều kiện tiên quyết | SV đã đăng nhập hệ thống. | |
| Điều kiện kết thúc thành công | Thông báo đăng ký nguyện vọng thành công. | |
| Phân quyền Actors | SV | |
| Kích hoạt | SV chọn chức năng đăng ký nguyện vọng học. | |
| **Luồng sự kiện chính** | **Bước** | **Hành động của tác nhân** |
|  |  | Sinh viên chọn chương trình đào tạo muốn đăng ký nguyện vọng. |
|  |  | Hệ thống hiển thị danh sách học phần tương ứng với chương trình đào tạo được phép đăng ký. |
|  |  | Sinh viên chọn danh sách các môn muốn học và gửi đăng ký. |
|  |  | Hệ thống kiểm tra kiểm tra các môn học nguyện vọng của sinh viên và lưu vào cơ sở dữ liệu. |
|  |  | Hệ thống thông báo đăng ký nguyện vọng thành công. |
| **Luồng ngoại lệ** | **Bước** | **Hành động phân nhánh** |
|  | 4 | 4a. Sinh viên không chọn môn học nào, hệ thống thông báo lỗi tương ứng. |
|  |  | 4b Sinh viên đăng ký quá số tín chỉ quy định trong học kì, hệ thống thông báo lỗi tương ứng. |

### 2.3.6 Đăng ký học phần

|  |  |  |
| --- | --- | --- |
| **Use case** | **Đăng ký học phần.** | |
| Mục tiêu | SV chính thức chọn lớp học phần để học. | |
| 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 phần. | |
| **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:   * Điều kiện SV đăng ký. * Điều kiện lớp học phần |
|  |  | Hệ thống lưu trữ dữ liệu lớp học phần SV đã đăng ký vào CSDL. |
|  |  | 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. |

### 2.3.7 Lấy thông tin thời khoá biểu học phần

|  |  |  |
| --- | --- | --- |
| **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. |

### 2.3.8 Đăng nhập với tài khoản và mật khẩu

|  |  |  |
| --- | --- | --- |
| **Use case** | **Đăng nhập với tài khoản và mật khẩu.** | |
| Mục tiêu | Cho phép cán bộ đăng nhập hệ thống bằng tài khoản nội bộ. | |
| Điều kiện tiên quyết | PĐT, TBM 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 | PĐT, TBM | |
| Kích hoạt | PĐT, TBM chọn chức năng đăng nhập. | |
| **Luồng sự kiện chính** | **Bước** | **Hành động của tác nhân** |
|  |  | Hệ thống hiển thị ô nhập tài khoản và mật khẩu. |
|  |  | Người dùng nhập tài khoản và mật khẩu vào ô tương ứng. |
|  |  | Hệ thống xác thực người dùng và quyền hạn người dùng trên hệ thống. |
|  |  | Hệ thống chuyển hướng giao diện người đến giao diện tương ứng. |
| **Luồng ngoại lệ** | **Bước** | **Hành động phân nhánh** |
|  | 2 | 2a. Người dùng bỏ trống, hệ thống thông báo lỗi tương ứng. |

### 2.3.9 Tạo đợt đăng ký nguyện vọng học

|  |  |  |
| --- | --- | --- |
| **Use case** | Tạo đợt đăng ký nguyện vọng học**.** | |
| Mục tiêu | Tạo đợt đăng ký để SV có thể chọn môn học nguyện vọng. | |
| Điều kiện tiên quyết | PĐT đã đăng nhập hệ thống. | |
| Điều kiện kết thúc thành công | Hệ thống thông báo tạo mới thành công. | |
| Phân quyền Actors | PĐT | |
| Kích hoạt | PĐT chọn chức năng tạo mới đăng ký nguyện vọng. | |
| **Luồng sự kiện chính** | **Bước** | **Hành động của tác nhân** |
|  |  | Hệ thống hiển thị form tạo mới đợt đăng ký nguyện vọng học. |
|  |  | PĐT chọn học kì muốn mở đăng ký nguyện vọng. |
|  |  | PĐT nhập thông tin thời gian bắt đầu, thời gian kết thúc của đợt đăng ký học, số TC tối thiểu và tối đa. |
|  |  | Hệ thống kiểm tra:   * Kì học PĐT đã chọn. * Thời gian PĐT đã nhập. * Số TC PĐT đã nhập. |
|  |  | Hệ thống lưu trữ cấu hình đăng ký nguyện vọng học vào CSDL. |
|  |  | Hệ thống thông báo tạo mới thành công. |
| **Luồng ngoại lệ** | **Bước** | **Hành động phân nhánh** |
|  | 2 | 2a. PĐT chọn học kì đã mở đăng ký học, hệ thống thông báo đã tồn tại. |
|  |  | 2b. Thông tin PĐT nhập không hợp lý, hệ thống thông báo lỗi tương ứng. |

### 2.3.10 Báo cáo đăng ký nguyện vọng học

|  |  |  |
| --- | --- | --- |
| **Use case** | Báo cáo đăng ký nguyện vọng học**.** | |
| Mục tiêu | Xem thống kê số lượng sinh viên đăng ký từng môn. | |
| Điều kiện tiên quyết | PDT đã đăng nhập hệ thống. | |
| Điều kiện kết thúc thành công | Hệ thống hiển thị tổng số sinh viên đăng ký, tổng số nguyện vọng và tổng số môn học nguyện vọng. | |
| Phân quyền Actors | PĐT | |
| Kích hoạt | PĐT chọn chức năng báo cáo đăng ký nguyện vọng học. | |
| **Luồng sự kiện chính** | **Bước** | **Hành động của tác nhân** |
|  |  | Hệ thống truy xuất dữ liệu đăng ký nguyện vọng học. |
|  |  | Hệ thống hiển thị dang sách môn học đã đăng ký và tổng số lượng nguyện vọng đã đăng ký. |
| **Luồng ngoại lệ** | **Bước** | **Hành động phân nhánh** |
|  | 1 | 1a. Chưa kết thúc thời gian đăng ký học, hệ thống hiển thị dữ liệu trống. |

### 2.3.11 Mở lớp học phần

|  |  |  |
| --- | --- | --- |
| **Use case** | Mở lớp học phần**.** | |
| Mục tiêu | Mở lớp chính thức để SV 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 tạo lớp. | |
| **Luồng sự kiện chính** | **Bước** | **Hành động của tác nhân** |
|  |  | PĐT chọn môn học muốn tạo lớp học phần. |
|  |  | Hệ thống hiển thị các lớp học phần đã được tạo tại kì học hiện tại. |
|  |  | PĐT chọn Thêm lớp học phần mới. |
|  |  | Hệ thống thêm một dòng mới vào bảng danh sách lớp học phần. Dòng này gồm nhiều ô tương ứng với các trường thông tin của lớp học phần. Hệ thống đánh dấu ô vào ô checkbox tương ứng với dòng đó và cập nhật trạng thái lựa chọn tương ứng. |
|  |  | PĐT nhập thông tin lớp học vào dòng mới. Nếu lớp có nhiều lớp thành phần, PĐT chọn thêm lớp học phần. Hệ thống thêm một dòng vào bảng con chứa nhiều ô tương ứng với các trường thông tin của lớp học phần. |
|  |  | PĐT tích chọn các checkbox tương ứng với các lớp học phần muốn lập thời khóa biểu. PĐT nhấn giữ chuột vào một ô trên lịch, kéo chuột để mở rộng vùng chọn qua các ô thời gian mong muốn và thả chuột để kết thúc vùng chọn. Hệ thống hiển thị cửa sổ modal để người dùng chọn phòng cho buổi học vừa tạo và nút xóa nếu người dùng muốn hủy buổi học vừa tạo. Hệ thống cập nhật lịch vừa tạo vào các lớp học phần đã tích chọn trước đó. |
|  |  | Hệ thống truy xuất lịch học của phòng đã chọn tại học kì đăng ký và hiển thị vào lịch. |
|  |  | PĐT điều chỉnh thông tin các lớp học phần đã tạo vào nhấn lưu lại. |
|  |  | Hệ thống kiểm tra dữ liệu người dùng đã nhập và lưu vào cơ sử dữ liệu. |
|  |  | Hệ thống thông báo thành công. |
| **Luồng ngoại lệ** | **Bước** | **Hành động phân nhánh** |
|  | 2 | 2a. Môn học chưa được cấu hình, hệ thống thông báo tương ứng. |
|  | 7 | 7a. Phòng học chọn không còn đủ chỗ trống cho xếp lịch, hệ thống thông báo hết chỗ và yêu cầu người dùng nhập lại. |
|  | 8 | 8a. Dữ liệu người dùng nhập không hợp lệ, hệ thống thông báo lỗi tương |

### 2.3.13 Sinh viên thay đổi đăng ký học

|  |  |  |
| --- | --- | --- |
| **Use case** | Sinh viên thay đổi đăng ký học**.** | |
| Mục tiêu | Cho phép PĐT thay đổi lớp học phần sinh viên. | |
| Điều kiện tiên quyết | PĐT đã đăng nhập hệ thống. | |
| Điều kiện kết thúc thành công | Hệ thống thông báo thành công. | |
| Phân quyền Actors | PĐT | |
| Kích hoạt | PĐT chọn chức năng thay đổi đăng ký học. | |
| **Luồng sự kiện chính** | **Bước** | **Hành động của tác nhân** |
|  |  | Hệ thống truy xuất sinh viên của lớp học phần. |
|  |  | Hệ thống hiển thị danh sách sinh viên trong lớp học phần đã chọn tương ứng. |
|  |  | PĐT chọn thêm mới hoặc xóa sinh viên khỏi lớp học. |
|  |  | Hệ thống kiểm tra dữ liệu PĐT đã chọn và lưu vào cơ sở dữ liệu. |
|  |  | Hệ thống thông báo thành công |
| **Luồng ngoại lệ** | **Bước** | **Hành động phân nhánh** |
|  | 4 | 4a. Không thể thêm sinh viên vào lớp học, hệ thống thông báo lỗi tương ứng. |

## 2.4 Biểu đồ tuần tự

## 2.5 Lớp phân tích

### 2.5.1 Xác định thực thể

* Entity\_Student.
* Entiy\_Subject.
* Entity\_CourseClass.
* Entity\_EducationProgram.
* Entity\_Semester.
* Enitity\_Room.
* Entity\_Staff

### 2.5.2 Xác định các lớp biên

* Form\_DangNhapOutlook
* Form\_DangNhapTaiKhoan
* Form\_ThongTinCaNhan
* Form\_KetQuaHocTap
* Form\_ChuongTrinhDaoTao
* Form\_DangKyNguyenVong
* Form\_DangKyHocPhan
* Form\_ThoiKhoaBieuSinhVien
* Form\_QuanLyLopHocPhan
* Form\_ThongBaoHeThong

### 2.5.3 Xác định các lớp điều khiển

* Ctrl\_DangNhapOutlook
* Ctrl\_DangNhapHeThong
* Ctrl\_ThongTinCaNhan
* Ctrl\_KetQuaHocTap
* Ctrl\_ChuongTrinhDaoTao
* Ctrl\_DangKyNguyenVong
* Ctrl\_DangKyHocPhan
* Ctrl\_LichHocSinhVien
* Ctrl\_QuanLyLopHocPhan
* Ctrl\_ThongBao

### 2.5.4 Biểu đồ

### 2.5.5 Cơ sở dữ liệu

### 2.5.6 Các bảng

- Bảng student

|  |  |  |
| --- | --- | --- |
| Trường | Kiểu dữ liệu | Mô tả |
| \_id | ObjectId | ID duy nhất của sinh viên (MongoDB ID) |
| CreatedAt | ISODate | Ngày tạo hồ sơ |
| UpdatedAt | String | Ngày cập nhật hồ sơ gần nhất |
| Version | Int64 | Phiên bản dữ liệu |
| Status | Int32 | Trạng thái hồ sơ (1: hoạt động, 0: không hoạt động,...) |
| PersonalInformation.FullName | String | Họ và tên đầy đủ |
| PersonalInformation.FirstName | String | Tên |
| PersonalInformation.LastName | String | Họ |
| PersonalInformation.Gender | Int32 | Giới tính (0: nữ, 1: nam,...) |
| PersonalInformation.BirthDate | ISODate | Ngày sinh |
| PersonalInformation.Ethnic | String | Dân tộc |
| PersonalInformation.PlaceOfBirth | String | Nơi sinh |
| PersonalInformation.IdNumber | String | Số CCCD/CMND |
| PersonalInformation.PhoneNumber | String | Số điện thoại |
| PersonalInformation.Email | String | Email cá nhân |
| PersonalInformation.OfficeEmail | String | Email do trường cấp |
| PersonalInformation.ContactAddress | String | Địa chỉ liên hệ |
| PersonalInformation.CurrentLive | String | Nơi đang sinh sống hiện tại |
| PersonalInformation.Note | String | Ghi chú thêm |
| InformationBySchool.StudentCode | String | Mã số sinh viên |
| InformationBySchool.BankCode | String | Mã ngân hàng nhận học bổng/lương |
| InformationBySchool.BankName | String | Tên ngân hàng |
| InformationBySchool.YearOfHighSchoolGraduation | Int32 | Năm tốt nghiệp THPT |
| EducationPrograms.Code | String | Mã chương trình đào tạo |
| EducationPrograms.Name | String | Tên chương trình đào tạo |
| EducationPrograms.CourseYear | Int32 | Khóa học |
| EducationPrograms.Type | Int32 | Loại chương trình (1: chính quy, 2: liên thông,...) |
| EducationPrograms.TrainingTime | Double | Thời gian đào tạo (năm) |
| EducationPrograms.Status | Int32 | Trạng thái chương trình đào tạo |
| EducationPrograms.DepartmentCode | String | Mã khoa |
| EducationPrograms.DepartmentName | String | Tên khoa |
| EducationPrograms.SpecialityCode | String | Mã chuyên ngành |
| EducationPrograms.SpecialityName | String | Tên chuyên ngành |
| EducationPrograms.StudentClassCode | String | Mã lớp học |
| EducationPrograms.StudentClassName | String | Tên lớp học |

- Bảng StudentSemester

|  |  |  |
| --- | --- | --- |
| Trường | Kiểu dữ liệu | Mô tả |
| \_id | ObjectId | ID duy nhất (MongoDB ObjectId) |
| StudentCode | String | Mã số sinh viên |
| SemesterCode | String | Mã học kỳ |
| SemesterName | String | Tên học kỳ |
| StartDate | ISODate | Ngày bắt đầu học kỳ |
| EndDate | ISODate | Ngày kết thúc học kỳ |
| EducationStartDate | ISODate | Ngày bắt đầu đào tạo |
| EducationEndDate | ISODate | Ngày kết thúc đào tạo |
| CreatedAt | ISODate | Ngày tạo bản ghi |
| UpdatedAt | String | Ngày cập nhật bản ghi |
| Version | Int64 | Phiên bản dữ liệu |
| CourseSubjects.CourseClassCode | String | Mã lớp học phần |
| CourseSubjects.CourseClassName | String | Tên lớp học phần |
| CourseSubjects.SubjectCode | String | Mã môn học |
| CourseSubjects.SubjectName | String | Tên môn học |
| CourseSubjects.TeacherCode | String | Mã giảng viên |
| CourseSubjects.TeacherName | String | Tên giảng viên |
| SubjectResults.SubjectCode | String | Mã môn học |
| SubjectResults.SubjectName | String | Tên môn học |
| SubjectResults.SubjectNameEng | String | Tên tiếng Anh của môn học |
| SubjectResults.NumberOfCredits | Int32 | Số tín chỉ |
| SubjectResults.Coeffiecient | Double | Hệ số điểm |
| SubjectResults.Mark | Decimal128 | Điểm tổng kết |
| SubjectResults.OriginalMark | Decimal128 | Điểm gốc |
| SubjectResults.MarkTypeChar | String | Loại điểm (ký tự) |
| SubjectResults.SubjectMarkType | Int32 | Loại điểm (số) |
| SubjectResults.ExamRound | Int32 | Lần thi |
| SubjectResults.Description | String | Ghi chú mô tả |
| SubjectResults.Result | String | Kết quả (Đạt/Không đạt,...) |

- Bảng Subject

|  |  |  |
| --- | --- | --- |
| Tên Cột | Kiểu Dữ Liệu | Mô Tả |
| \_id | ObjectId | Định danh duy nhất của môn học |
| CreatedAt | ISODate | Ngày tạo bản ghi |
| DepartmentCode | String | Mã khoa quản lý môn học |
| IsCalculateMark | Boolean | Có tính điểm hay không |
| NumberOfCredits | Int32 | Số tín chỉ của môn học |
| Status | Int32 | Trạng thái của môn học |
| SubjectCode | String | Mã môn học |
| SubjectDescription | String | Mô tả chi tiết môn học |
| UpdatedAt | String | Ngày cập nhật gần nhất |
| SubjectName | String | Tên môn học |
| SubjectNameEng | String | Tên môn học (tiếng Anh) |

- Bảng SubjectReigster

|  |  |  |
| --- | --- | --- |
| Tên Cột | Kiểu Dữ Liệu | Mô Tả |
| \_id | ObjectId | Định danh duy nhất |
| CorrelationId | UUID | Mã định danh tương quan |
| CreatedAt | ISODate | Ngày tạo bản ghi |
| StudentCodes | Array | Danh sách mã sinh viên đăng ký |
| SubjectCode | String | Mã môn học |
| UpdatedAt | String | Ngày cập nhật gần nhất |
| Version | Int64 | Phiên bản |

- Bảng SubjectTimelineConfig

|  |  |  |
| --- | --- | --- |
| Tên Cột | Kiểu Dữ Liệu | Mô Tả |
| \_id | ObjectId | Định danh duy nhất |
| CreatedAt | ISODate | Ngày tạo bản ghi |
| LabLesson | Int32 | Số buổi thực hành |
| MinDaySpaceLab | Int32 | Khoảng cách ngày tối thiểu giữa các buổi lab |
| PeriodTotal | Int32 | Tổng số tiết |
| Stage | Int32 | Giai đoạn học |
| SubjectCode | String | Mã môn học |
| UpdatedAt | String | Ngày cập nhật gần nhất |
| LabMinStudent | Int32 | Số sinh viên tối thiểu cho lớp thực hành |
| MinDaySpaceLecture | Int32 | Khoảng cách ngày tối thiểu giữa các buổi lý thuyết |
| LabPeriod | Int32 | Số tiết thực hành |
| LabRequiredConditions | Array | Điều kiện bắt buộc của thực hành |
| LabStartWeek | Int32 | Tuần bắt đầu thực hành |
| LabTotal | Int32 | Tổng số tiết thực hành |
| LectureLesson | Int32 | Số buổi lý thuyết |
| LectureMinStudent | Int32 | Số sinh viên tối thiểu cho lớp lý thuyết |
| LecturePeriod | Int32 | Số tiết lý thuyết |
| LectureRequiredConditions | Array | Điều kiện bắt buộc của lý thuyết |
| LectureStartWeek | Int32 | Tuần bắt đầu lý thuyết |
| LectureTotal | Int32 | Tổng số tiết lý thuyết |

- Bảng staff

|  |  |  |
| --- | --- | --- |
| Tên Cột | Kiểu Dữ Liệu | Mô Tả |
| \_id | ObjectId | Định danh duy nhất |
| Code | String | Mã cán bộ giảng viên |
| CreatedAt | ISODate | Ngày tạo bản ghi |
| DepartmentCode | String | Mã khoa |
| FullName | String | Họ và tên |
| UpdatedAt | String | Ngày cập nhật gần nhất |

- Bảng SlotTimeline

|  |  |  |
| --- | --- | --- |
| Tên Cột | Kiểu Dữ Liệu | Mô Tả |
| \_id | ObjectId | Định danh duy nhất |
| BuildingCode | String | Mã tòa nhà |
| CourseClassCode | String | Mã lớp học phần |
| CreatedAt | ISODate | Ngày tạo bản ghi |
| DayOfWeek | Int32 | Thứ trong tuần |
| RoomCode | String | Mã phòng học |
| Slots | Array | Danh sách các khung giờ học |
| UpdatedAt | String | Ngày cập nhật gần nhất |
| Version | Int64 | Phiên bản |

- Bảng CourseClass

|  |  |  |
| --- | --- | --- |
| Tên Cột | Kiểu Dữ Liệu | Mô Tả |
| \_id | ObjectId | Định danh duy nhất |
| ClassIndex | Int32 | Số thứ tự lớp |
| CourseClassCode | String | Mã lớp học phần |
| CourseClassName | String | Tên lớp học phần |
| CreatedAt | ISODate | Ngày tạo bản ghi |
| NumberStudents | Int32 | Số lượng sinh viên |
| SemesterCode | String | Mã học kỳ |
| Session | Int32 | Phiên học |
| Stage | Int32 | Giai đoạn |
| Status | Int32 | Trạng thái |
| StudentIds | list | Danh sách ID sinh viên |
| SubjectCode | String | Mã môn học |
| TotalSession | Int32 | Tổng số buổi học |
| UpdatedAt | String | Ngày cập nhật gần nhất |
| Version | Int64 | Phiên bản |
| CourseClassType | Int32 | Loại lớp học phần |
| SessionLength | Int32 | Thời lượng mỗi phiên học |

- Bảng CourseClassCondition

|  |  |  |
| --- | --- | --- |
| Tên Cột | Kiểu Dữ Liệu | Mô Tả |
| \_id | ObjectId | Định danh duy nhất |
| ConditionCode | String | Mã điều kiện lớp học phần |
| CreatedAt | ISODate | Ngày tạo bản ghi |
| UpdatedAt | String | Ngày cập nhật gần nhất |
| ConditionName | String | Tên điều kiện |

- Bảng EducationProgram

|  |  |  |
| --- | --- | --- |
| Tên Cột | Kiểu Dữ Liệu | Mô Tả |
| \_id | ObjectId | Định danh duy nhất |
| Code | String | Mã chương trình đào tạo |
| CourseCode | String | Mã khóa học |
| CreatedAt | ISODate | Ngày tạo bản ghi |
| EducationSubjects | Array | Danh sách môn học trong chương trình |
| Name | String | Tên chương trình đào tạo |
| SpecialityCode | String | Mã chuyên ngành |
| TrainingTime | Double | Thời gian đào tạo |
| UpdatedAt | String | Ngày cập nhật gần nhất |
| Type | Int32 | Loại chương trình |

- Bảng EventStore

|  |  |  |
| --- | --- | --- |
| Tên Cột | Kiểu Dữ Liệu | Mô Tả |
| \_id | ObjectId | Định danh duy nhất |
| AggregateId | ObjectId | ID tổng hợp sự kiện |
| CreatedAt | ISODate | Ngày tạo sự kiện |
| EventData | String | Dữ liệu sự kiện |
| EventType | String | Loại sự kiện |
| Metadata | Object | Thông tin metadata |
| UpdatedAt | String | Ngày cập nhật |
| Version | Int64 | Phiên bản |
| AggregateType | String | Loại tổng hợp |
| Metadata.PerformedBy | String | ID người thực hiện |
| Metadata.PerformedByName | String | Tên người thực hiện |

- Bảng ReigsterSaga

|  |  |  |
| --- | --- | --- |
| Tên Cột | Kiểu Dữ Liệu | Mô Tả |
| \_id | UUID | Định danh duy nhất |
| CurrentState | String | Trạng thái hiện tại |
| EducationEnd | ISODate | Ngày kết thúc học tập |
| EducationStart | ISODate | Ngày bắt đầu học tập |
| EndDate | ISODate | Ngày kết thúc đăng ký |
| EventStoreId | String | Mã sự kiện liên quan |
| MaxCredit | Int32 | Số tín chỉ tối đa |
| MinCredit | Int32 | Số tín chỉ tối thiểu |
| NumberStudent | Int32 | Số lượng sinh viên |
| NumberSubject | Int32 | Số lượng môn học |
| SemesterCode | String | Mã học kỳ |
| StartDate | ISODate | Ngày bắt đầu đăng ký |
| StudentChangeEnd | ISODate | Ngày kết thúc thay đổi sinh viên |
| Version | Int32 | Phiên bản |
| NumberWish | Int32 | Số nguyện vọng |
| StudentChangeStart | ISODate | Ngày bắt đầu thay đổi sinh viên |

- Bảng Room

|  |  |  |
| --- | --- | --- |
| Tên Cột | Kiểu Dữ Liệu | Mô Tả |
| \_id | ObjectId | Định danh duy nhất |
| BuildingCode | String | Mã tòa nhà |
| Capacity | Int32 | Sức chứa phòng học |
| Code | String | Mã phòng học |
| CreatedAt | ISODate | Ngày tạo bản ghi |
| Name | String | Tên phòng học |
| SupportedConditions | Array | Điều kiện hỗ trợ |
| UpdatedAt | String | Ngày cập nhật gần nhất |

- Bảng Semester

|  |  |  |
| --- | --- | --- |
| Tên Cột | Kiểu Dữ Liệu | Mô Tả |
| \_id | ObjectId | Định danh duy nhất |
| CreatedAt | ISODate | Ngày tạo học kỳ |
| EndDate | ISODate | Ngày kết thúc học kỳ |
| SemesterCode | String | Mã học kỳ |
| SemesterName | String | Tên học kỳ |
| StartDate | ISODate | Ngày bắt đầu học kỳ |
| UpdatedAt | String | Ngày cập nhật gần nhất |
| Version | Int64 | Phiên bản |
| SemesterStatus | Int32 | Trạng thái học kỳ |

## 2.6. Thiết kế api

2.6.1 Api danh sách phòng học

**GET /api/Building/Room**

* **Mô tả**: Lấy danh sách các phòng học.
* **Tham số**:
  + Filters: Mảng các bộ lọc (Education.Core.Domain.FilterModel) để lọc dữ liệu theo điều kiện (ví dụ: field=buildingId, operator=eq, value=1).
  + Sorts: Mảng chuỗi để sắp xếp (ví dụ: name asc).
  + Includes: Mảng chuỗi để bao gồm dữ liệu liên quan (ví dụ: thông tin tòa nhà).
  + Page và PageSize: Phân trang (số nguyên, định dạng int32).
* **Phản hồi**: Trả về mã trạng thái 200 OK với danh sách phòng học (định dạng JSON, nhưng không xác định schema cụ thể trong tài liệu).

2.6.2

**GET /api/Course**

* **Mô tả**: Lấy danh sách các khóa học.
* **Tham số**: Tương tự như /api/Building/Room (Filters, Sorts, Includes, Page, PageSize).
* **Phản hồi**: Mã 200 OK với danh sách khóa học.

2.6.3

**GET /api/CourseClass**

* **Mô tả**: Lấy danh sách lớp học.
* **Tham số**: Filters, Sorts, Includes, Page, PageSize.
* **Phản hồi**: Mã 200 OK.

2.6.4

**POST /api/CourseClass**

* **Mô tả**: Tạo mới một lớp học.
* **Body**: Sử dụng schema CreateCourseClassCommand:
  + courseClassCode: Mã lớp học (string).
  + courseClassName: Tên lớp học (string).
  + courseClassType: Loại lớp học (integer).
  + subjectCode: Mã môn học (string).
  + semesterCode: Mã kỳ học (string).
  + slotTimelines: Mảng các lịch học (SlotTimelineModel), bao gồm:
    - roomCode: Mã phòng (string).
    - dayOfWeek: Ngày trong tuần (integer, ví dụ: 1 = Thứ Hai).
    - slot: Mảng các khe thời gian (string).
* **Phản hồi**: Mã 200 OK.

2.6.5

**GET /api/CourseClass/Schedule**

* **Mô tả**: Lấy thời khóa biểu của lớp học.
* **Tham số**: Filters, Sorts, Includes, Page, PageSize.
* **Phản hồi**: Mã 200 OK.

**2**.6.6

**GET /api/CourseClass/Condition**

* **Mô tả**: Lấy danh sách điều kiện mở lớp (ví dụ: số lượng sinh viên tối thiểu).
* **Tham số**: Filters, Sorts, Includes, Page, PageSize.
* **Phản hồi**: Mã 200 OK.

**2**.6.7

**GET /api/Education**

* **Mô tả**: Lấy danh sách các chương trình đào tạo.
* **Tham số**: Filters, Sorts, Includes, Page, PageSize.
* **Phản hồi**: Mã 200 OK.

2.6.8

**GET /api/EventStore**

* **Mô tả**: Lấy lịch sử thay đổi của các bản ghi (có thể dùng để theo dõi audit log).
* **Tham số**: Filters, Sorts, Includes, Page, PageSize.
* **Phản hồi**: Mã 200 OK.

2.6.10

**GET /api/Semester**

* **Mô tả**: Lấy danh sách các kỳ học.
* **Tham số**: Filters, Sorts, Includes, Page, PageSize.
* **Phản hồi**: Mã 200 OK.

2.6.11

**GET /api/Subject**

* **Mô tả**: Lấy danh sách các môn học.
* **Tham số**: Filters, Sorts, Includes, Page, PageSize.
* **Phản hồi**: Mã 200 OK.

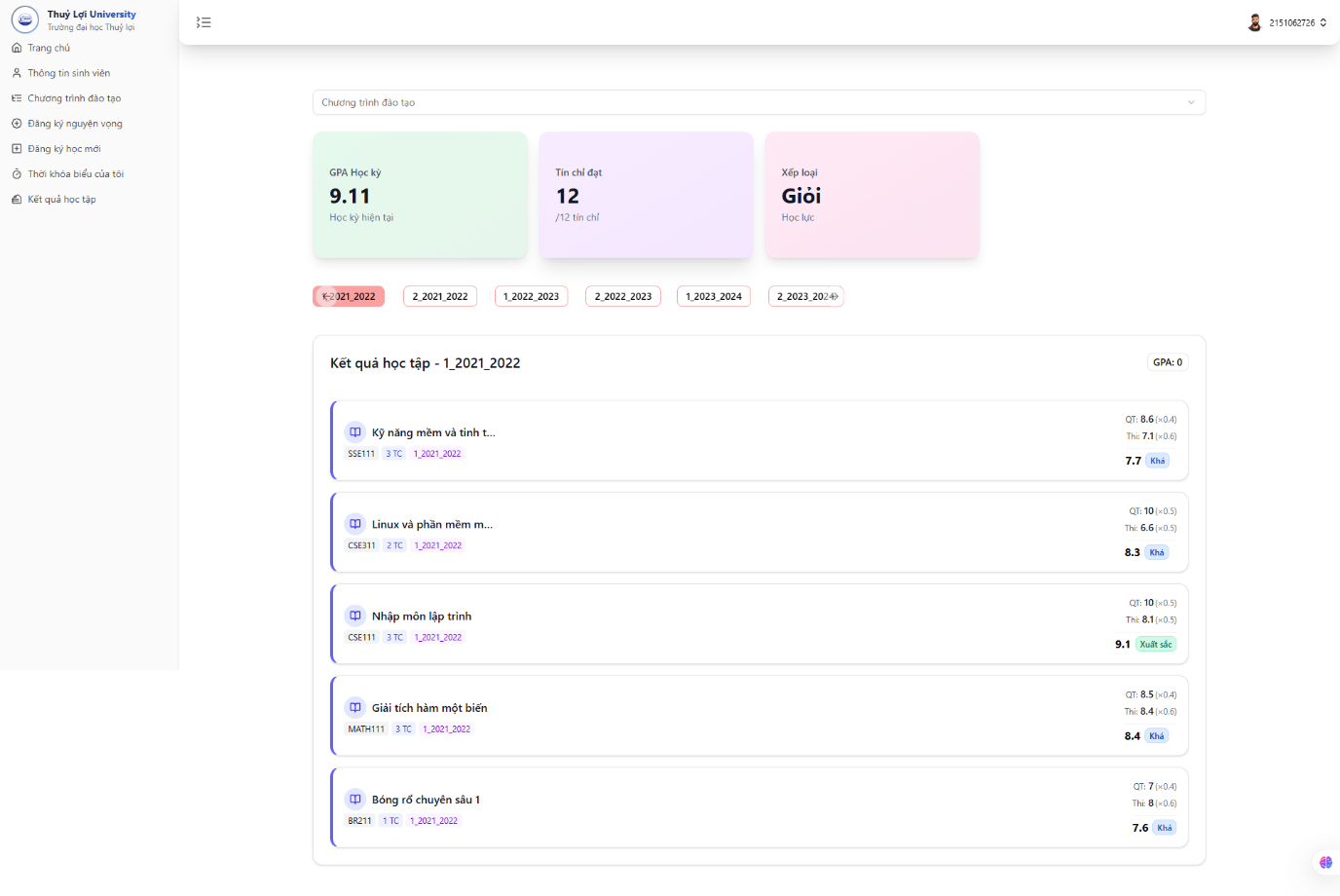
2.6.12

**GET /api/Subject/{subjectCode}/timeline-config**

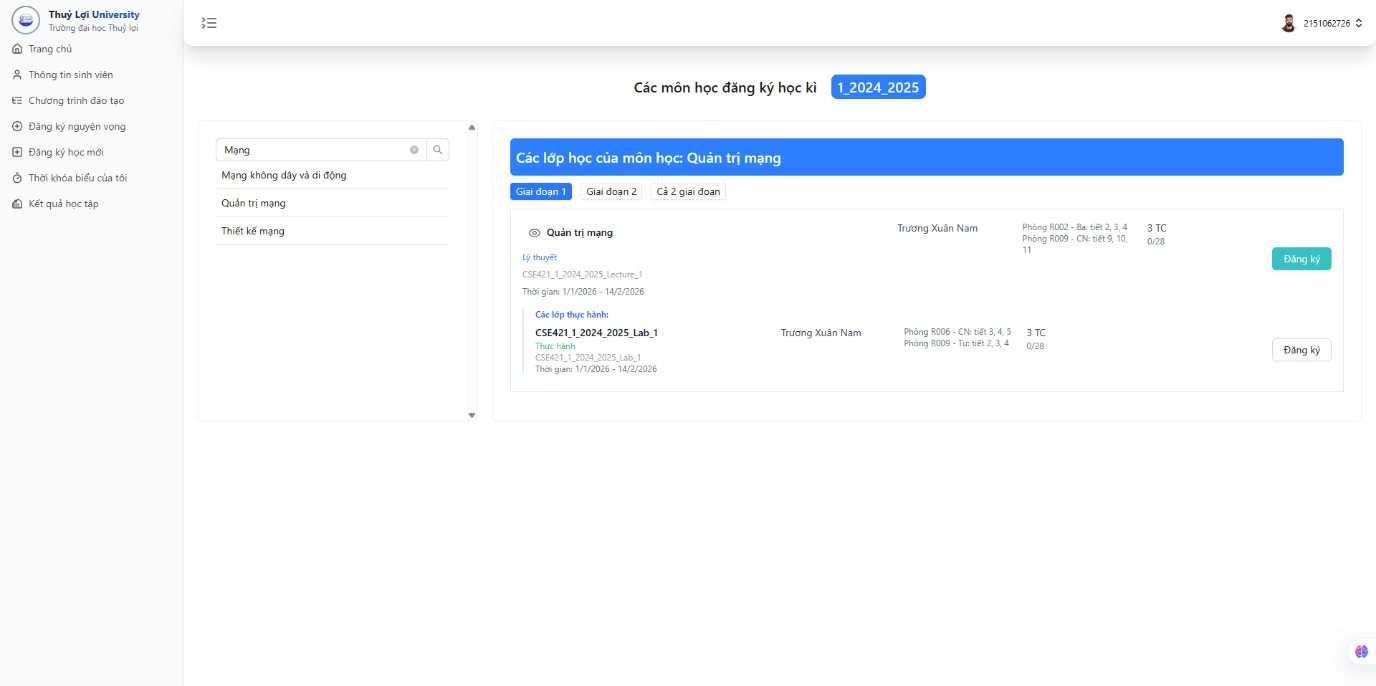
* **Mô tả**: Lấy cấu hình thời khóa biểu cho một môn học cụ thể.
* **Tham số**:
  + subjectCode: Mã môn học (bắt buộc, string).
* **Phản hồi**: Mã 200 OK.

## 2.7 THIẾT KẾ GIAO DIỆN

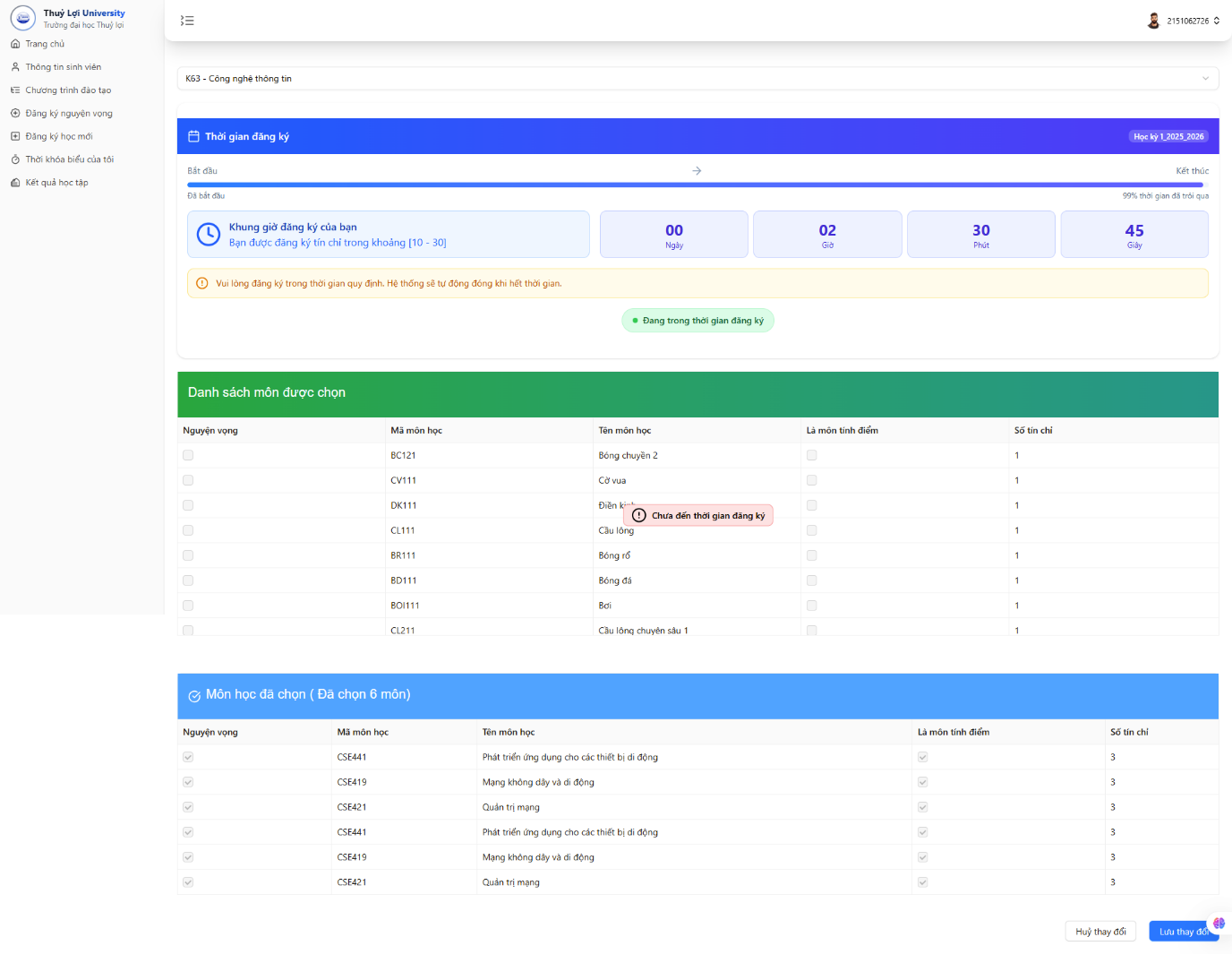
### 2.7.1 Giao diện xem kết quả học tập



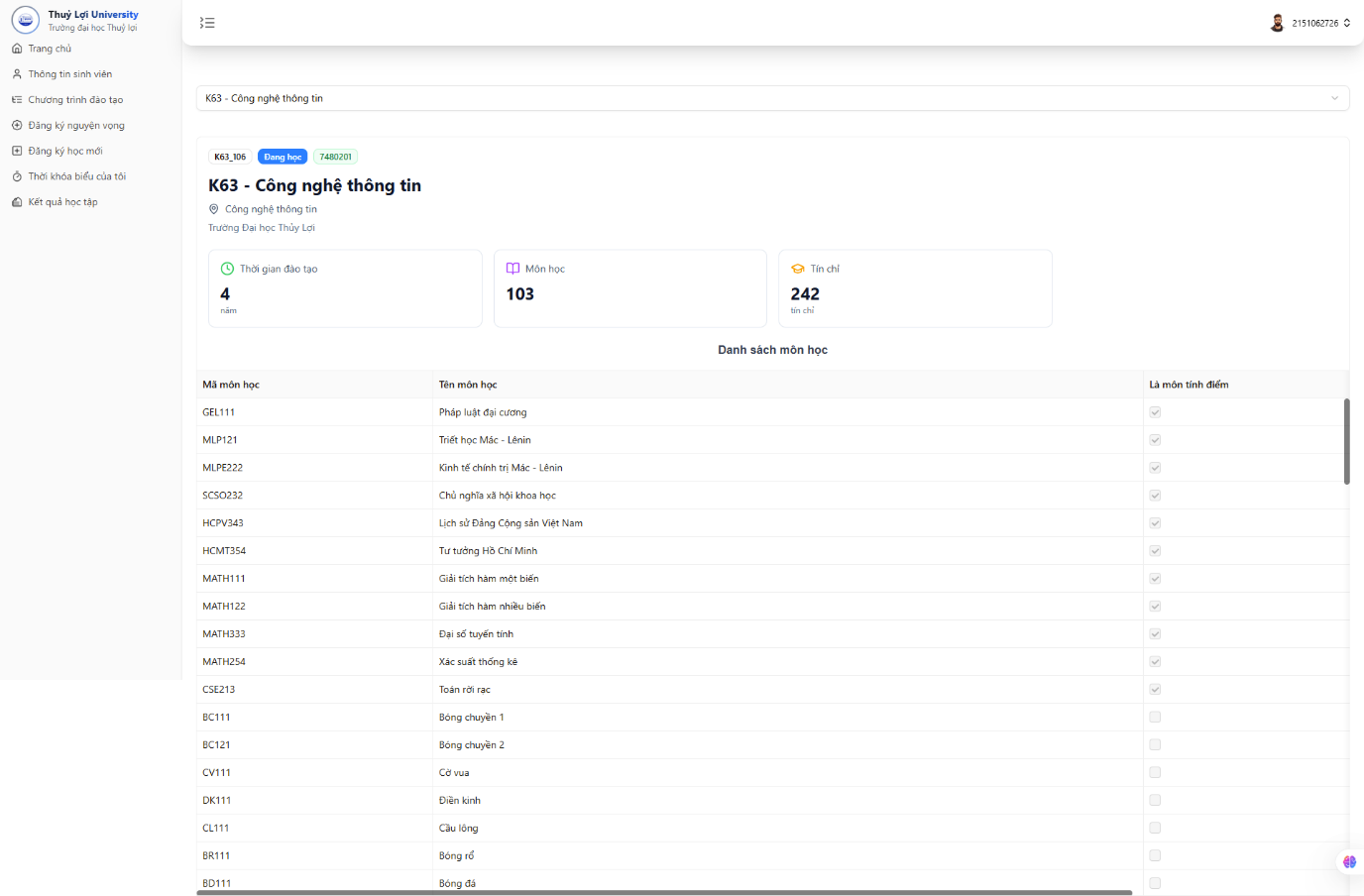
### 2.7.2 Giao diện đăng ký lớp học phần



## 2.7.3 Giao diện đăng ký nguyện vọng học



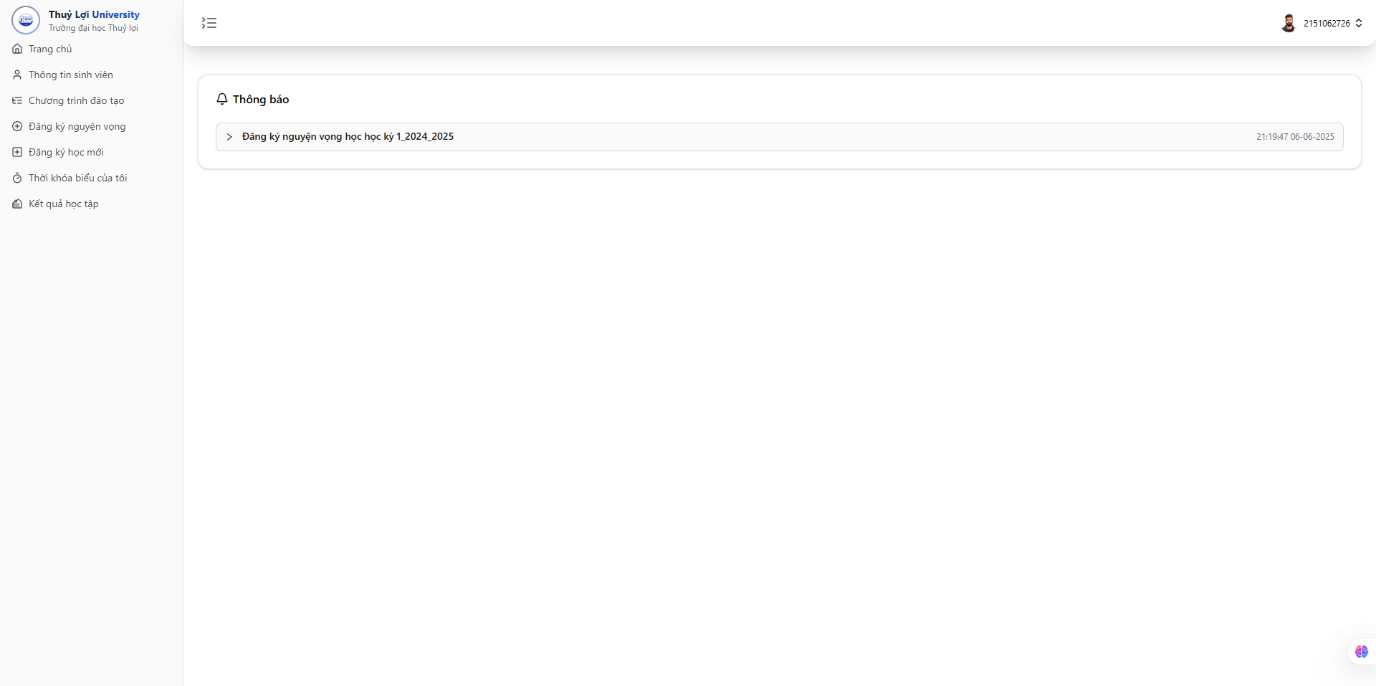
### 2.7.4 Giao diện xem thông tin chương trình đào tạo



### 2.7.5 Giao diện xem thông tin cá nhân



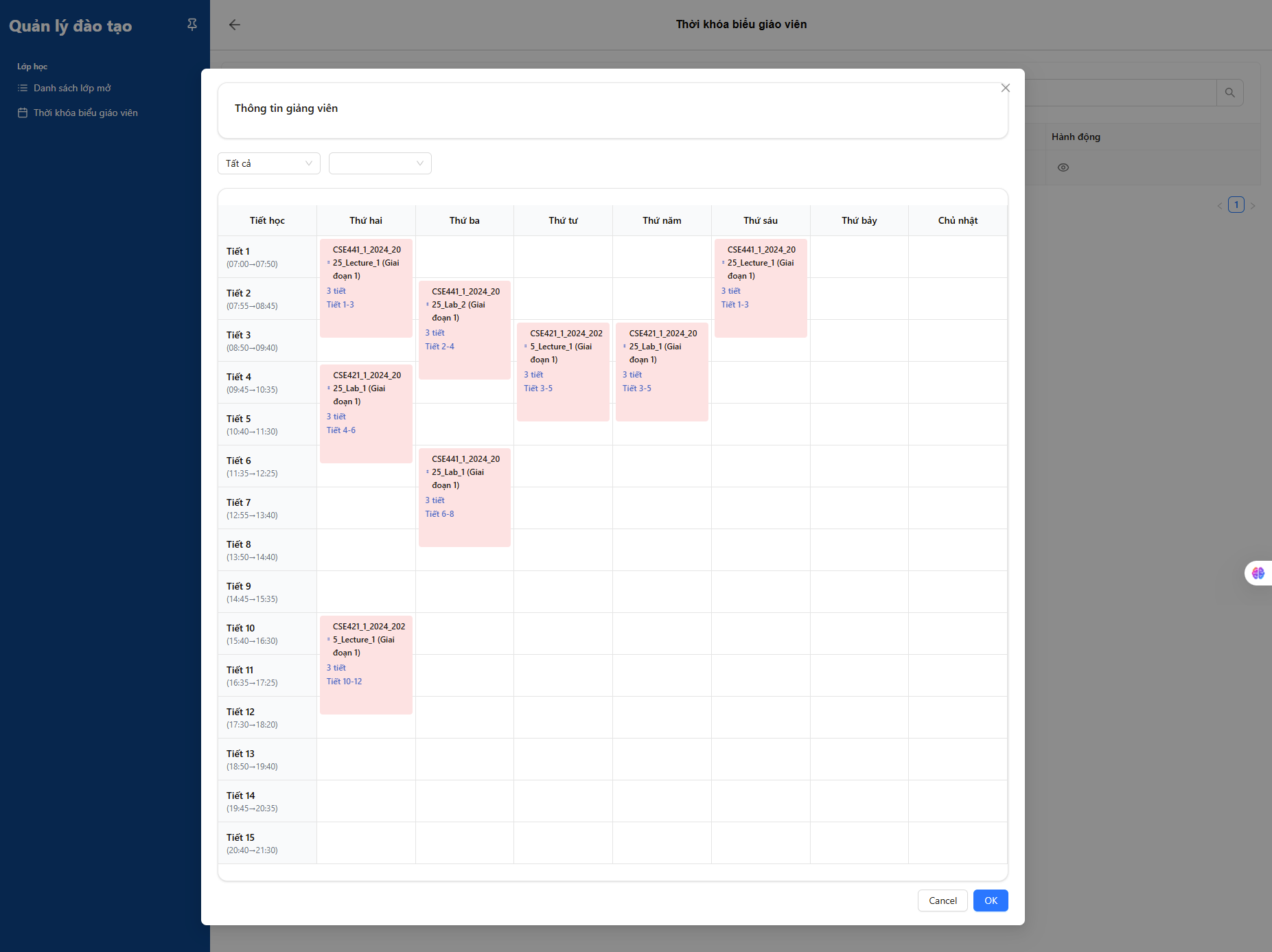
### 2.7.6 Giao diện trang chủ sinh viên



### 2.7.8 Giao diện đăng nhập sinh viên



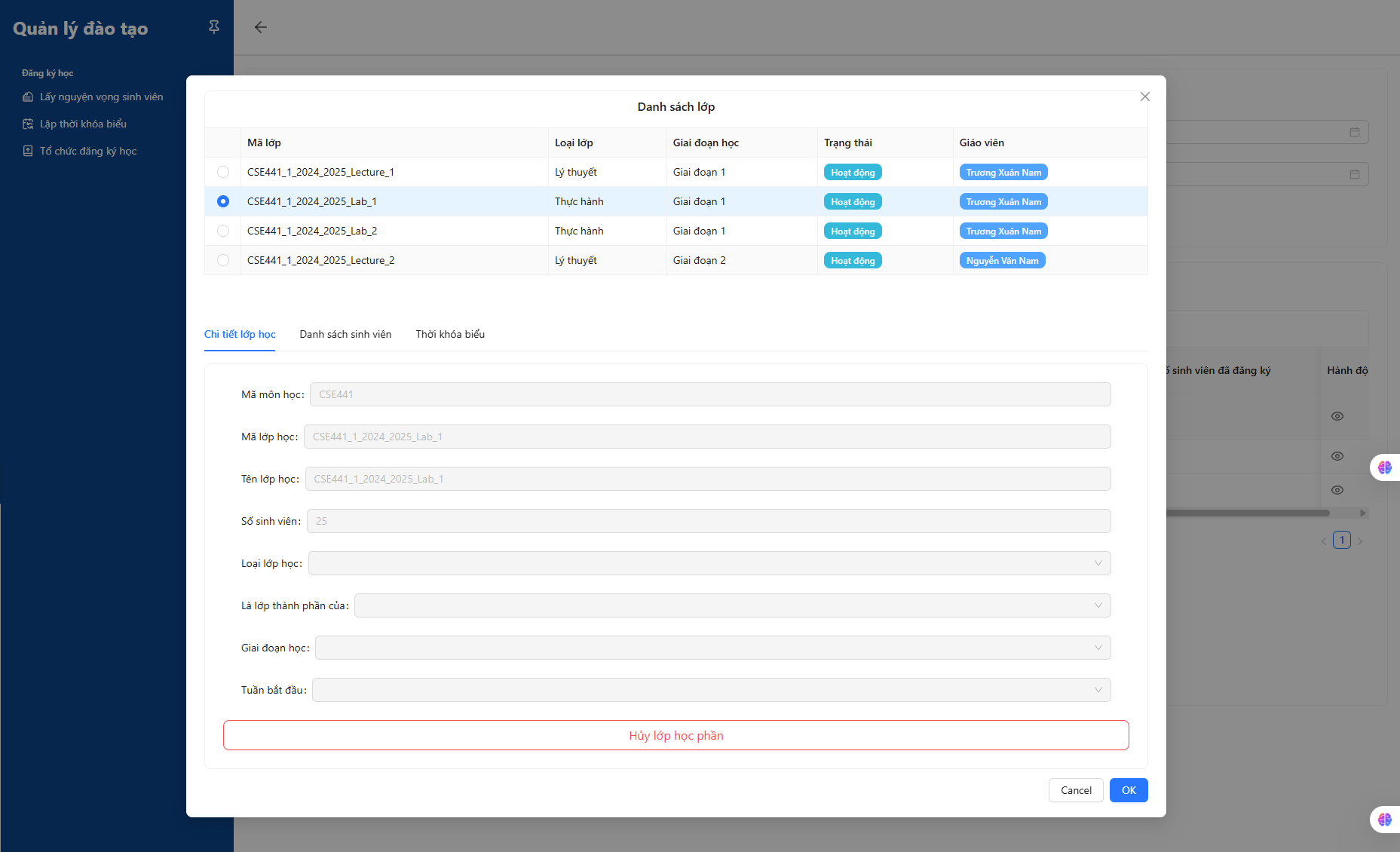
### 2.7.9 Giao diện xem thời khóa biểu giáo viên



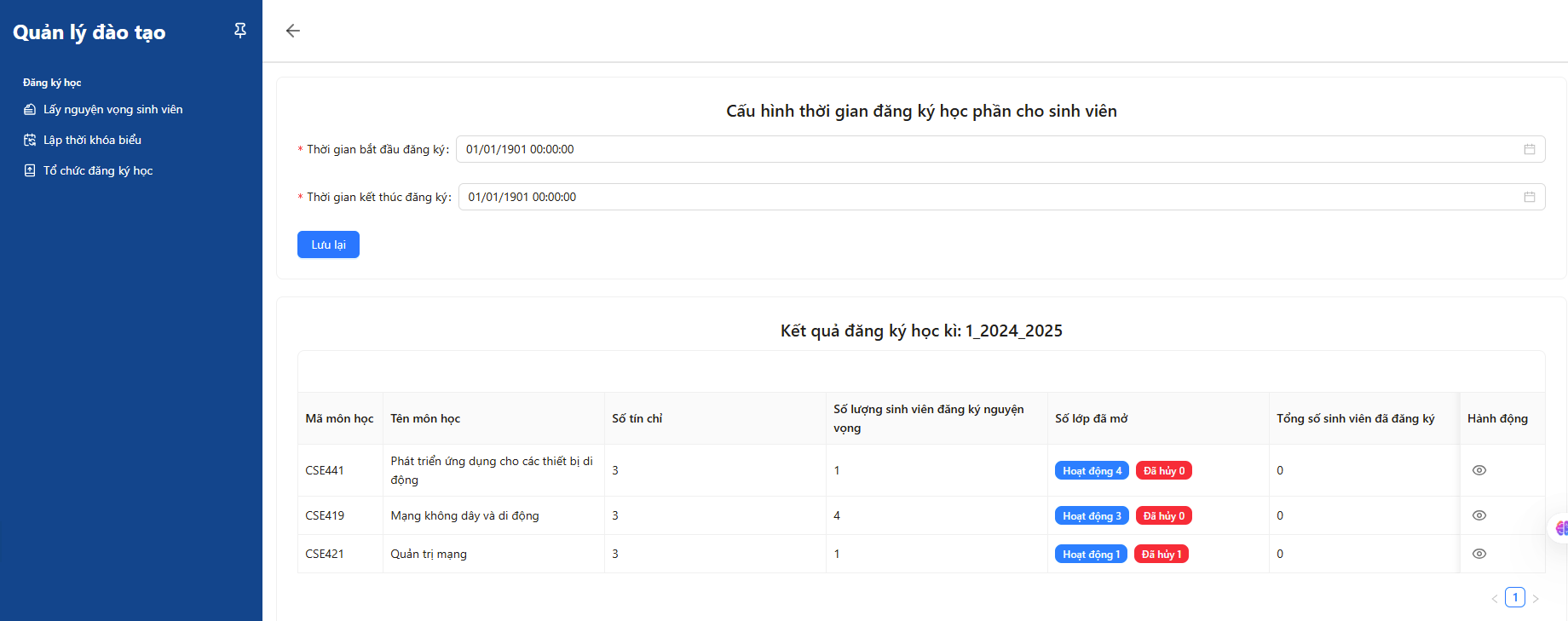
### 2.7.10 Giao diện phân công giáo viên cho lớp học phần



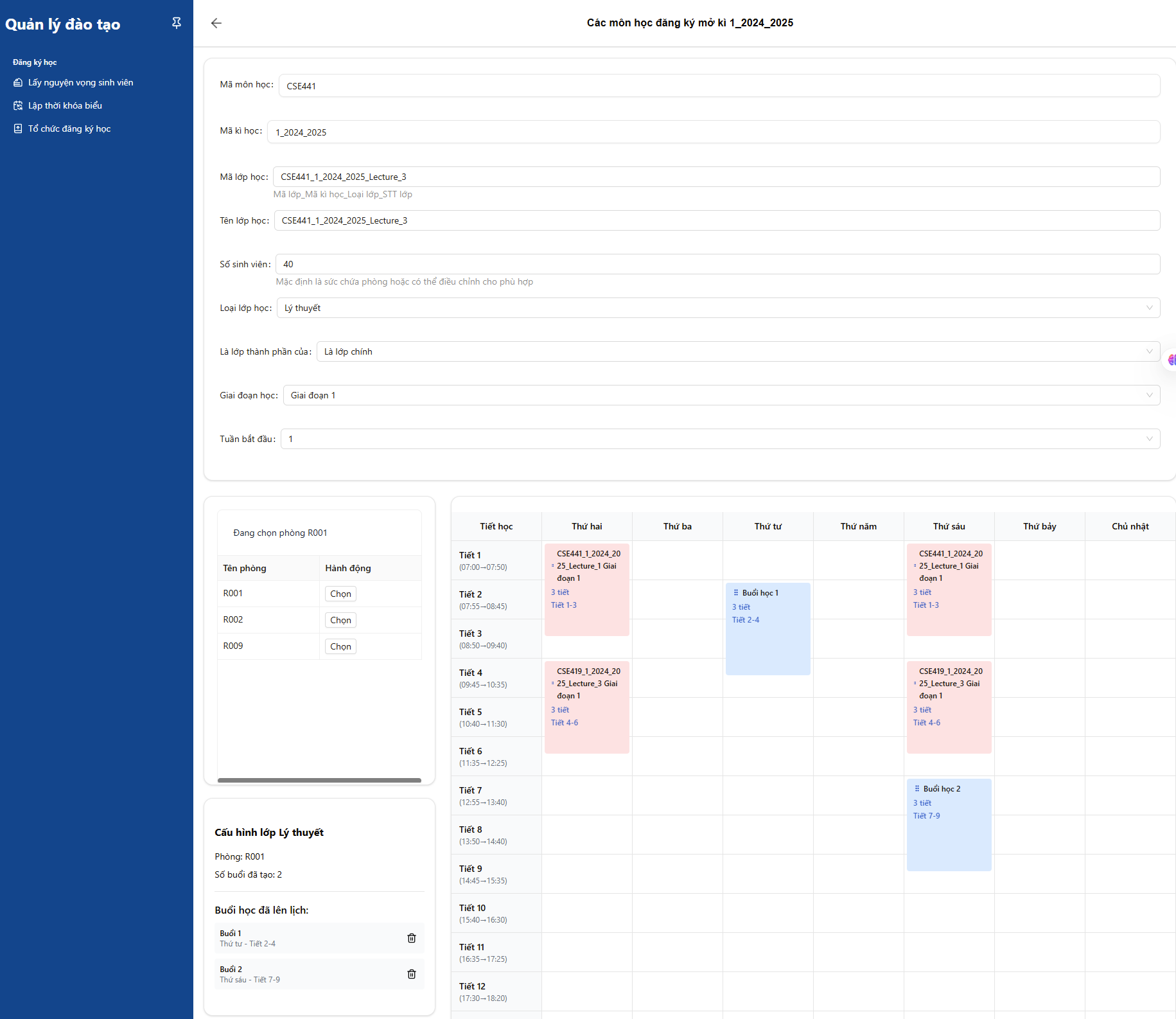
### 2.7.14 Giao diện quản lý lớp học phần



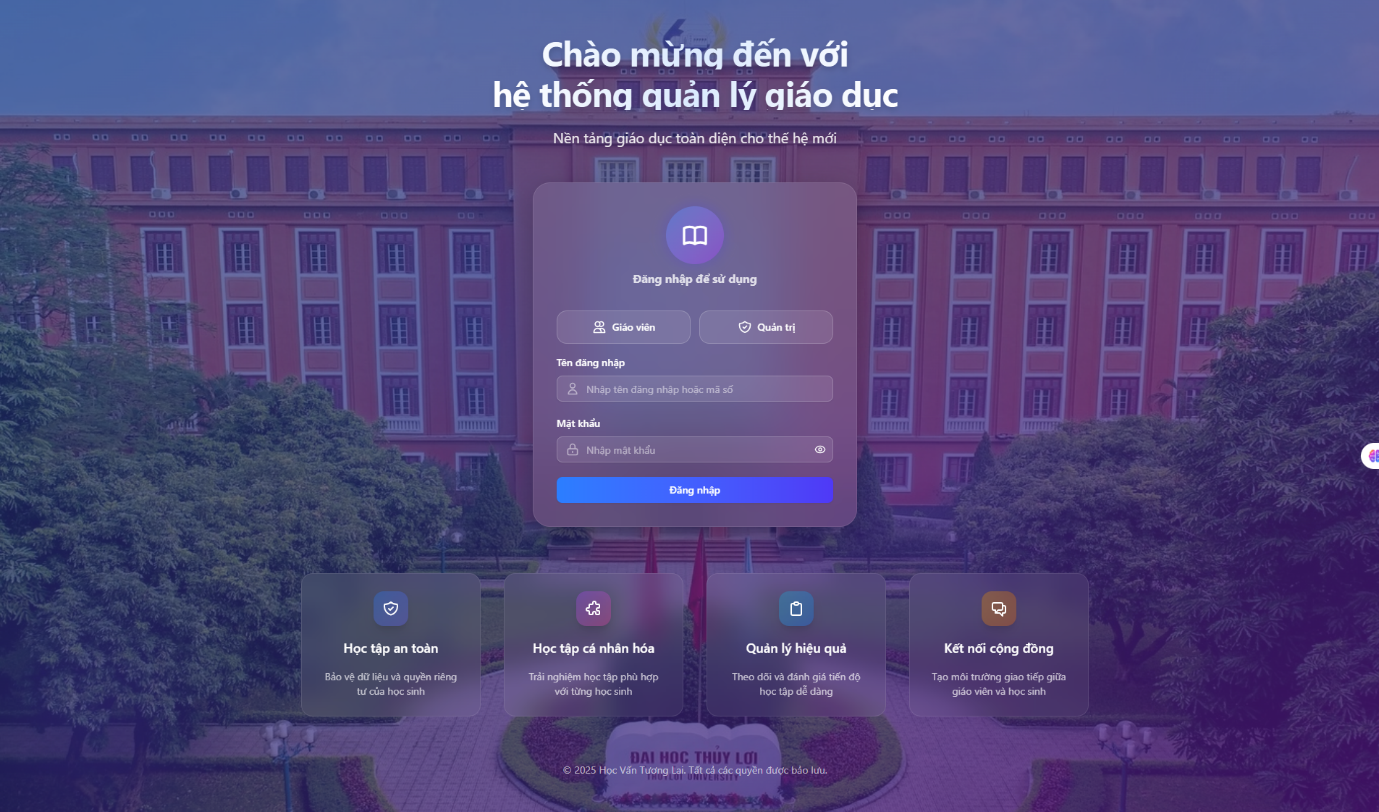
### 2.7.15 Giao diện cấu hình đăng ký học phần



### 2.7.16 Giao diện tạo mới lớp học phần

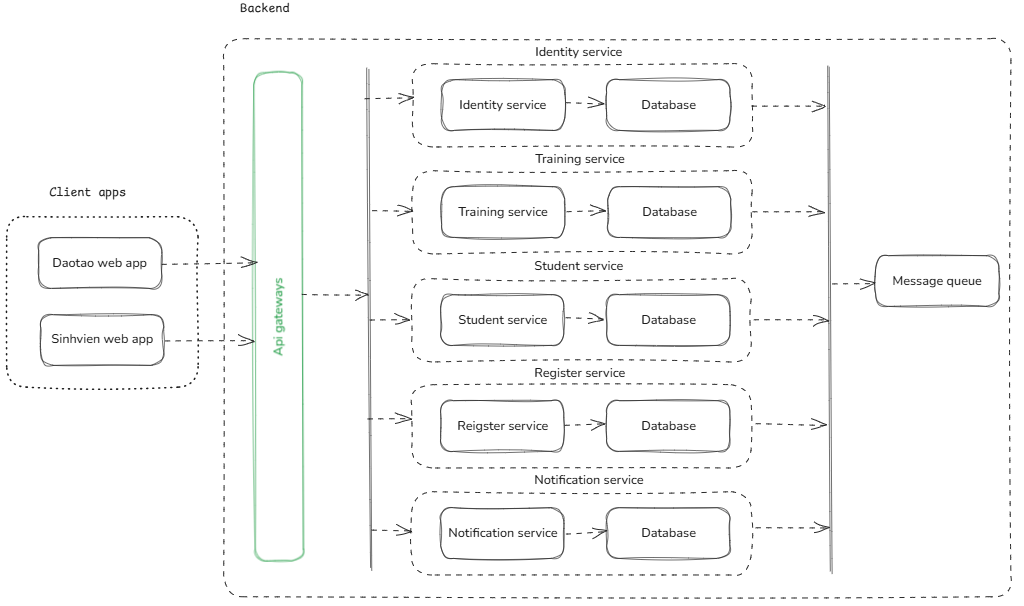


### 2.7.17 Giao diện đăng nhập của PĐT



# CHƯƠNG 3 TRIỂN KHAI

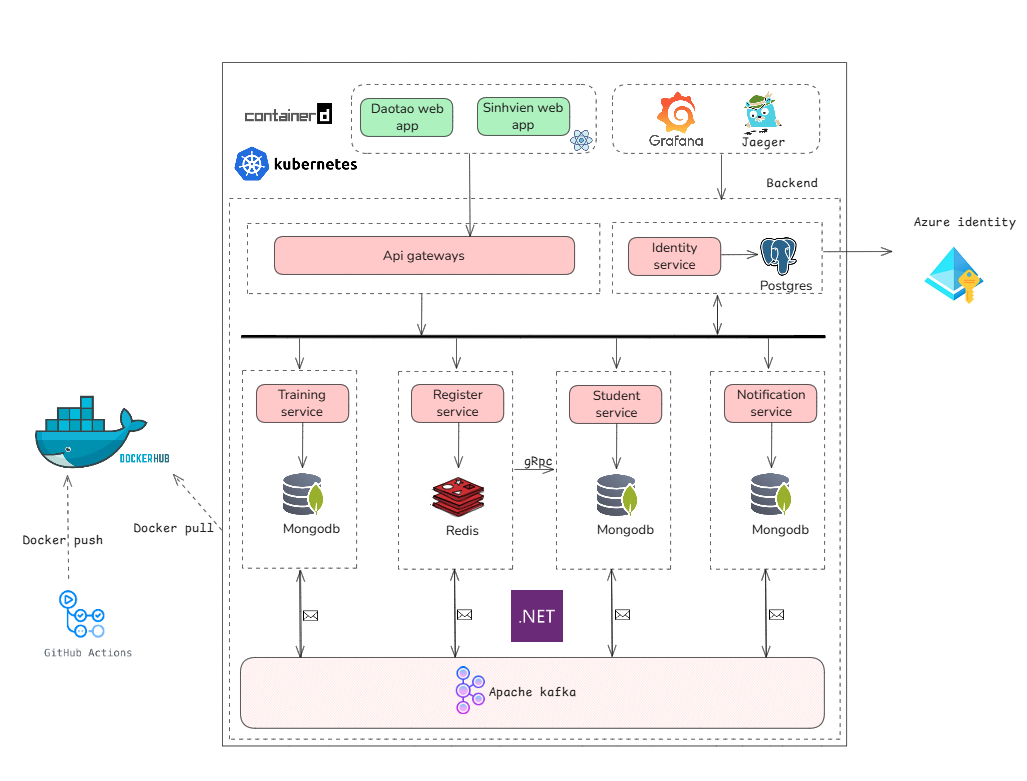
## 3.1 Sơ đồ thành phần



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).