# 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

# 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 | Đăng nhập với tài khoản và mật khẩu |  |
|  | Cấu hình môn học |  |
|  | Tạo đợt đăng ký học |  |
|  | Báo cáo đăng ký nguyện vọng học |  |
|  | Mở lớp học phần |  |
|  | Lập thời khóa biểu |  |
|  | Sinh viên thay đổi đăng ký học |  |
|  | TBM | Đăng nhập với tài khoản và mật khẩu |  |
|  | Phân công giảng viên dạy cho lớp học phần |  |
|  | Xem thời khóa biểu giáo viên |  |

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 **Lấy thông tin cá nhân**

|  |  |  |
| --- | --- | --- |
| **Use case** | **Lấy thông tin cá nhân** | |
| Mục tiêu | Lấy thông tin cá nhân. | |
| 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 yêu cầu xem thông tin. | |
| **Luồng sự kiện chính** | **Bước** | **Hành động của tác nhân** |
|  |  | Hệ thống truy xuất dữ liệu. |
|  |  | Hệ thống hiển thị thông tin tương ứng. |
| **Luồng ngoại lệ** | **Bước** | **Hành động phân nhánh** |
|  | 2 | 2a. Không có dữ liệu, thông báo lỗi tương ứng. |

2.3.3 **Lấy thông tin kết quả học tập**

|  |  |  |
| --- | --- | --- |
| **Use case** | **Lấy thông tin kết quả học tập.** | |
| Mục tiêu | Xem điểm và kết quả học tập của SV. | |
| Điều kiện tiên quyết | SV đã đăng nhập hệ thống. | |
| Điều kiện kết thúc thành công | Thông tin kết quả học tập của sinh viên 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** |
|  |  | 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. Không có dữ liệu, thông báo lỗi tương ứng. |

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** |
|  |  | 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 danh sách lớp học phần tương ứng với chương trình đào tạo được phép đăng ký. |
|  |  | Sinh viên chọn lớp học phần muốn đăng ký. |
|  |  | Hệ thống kiểm tra kiểm tra các lớp học phần của sinh viên và lưu vào cơ sở dữ liệu. |
|  |  | Hệ thống thông báo đăng ký học thành công. |
| **Luồng ngoại lệ** | **Bước** | **Hành động phân nhánh** |
|  | 4 | 4a. Sinh viên không chọn lớp học nào, 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** |
|  |  | 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 | PDT 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** |
|  |  | 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. |
|  |  | Hệ thống kiểm tra thông tin người dùng đã nhập và lưu vào cơ sở dữ liệu. |
|  |  | Hệ thống thông báo tạo mới thành công. |
| **Luồng ngoại lệ** | **Bước** | **Hành động phân nhánh** |
|  | 1 | 1a. Người dùng chọn học kì đã mở đăng ký học, hệ thống thông báo đã tồn tại. |

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 | PDT chọn chức năng tạo lớp. | |
| **Luồng sự kiện chính** | **Bước** | **Hành động của tác nhân** |
|  |  | Hệ thống truy xuất thông tin cấu hình môn học. |
|  |  | PĐT nhập thông tin lớp học và chọn loại lớp học. |
|  |  | PĐT chọn phòng muốn lập thời khóa biểu cho lớp học phần. |
|  |  | Hệ thống truy xuất lịch học của phòng đã chọn tại học kì đăng ký. |
|  |  | Hệ thống hiển thị theo tuần danh sách lịch học của phòng học đã chọn. |
|  |  | PĐT chọn ngày học trong tuần và tiết học trong ngày. |
|  |  | 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** |
|  | 1 | 1a. Môn học chưa được cấu hình, hệ thống thông báo tương ứng. |

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. |

## 2.6. 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.

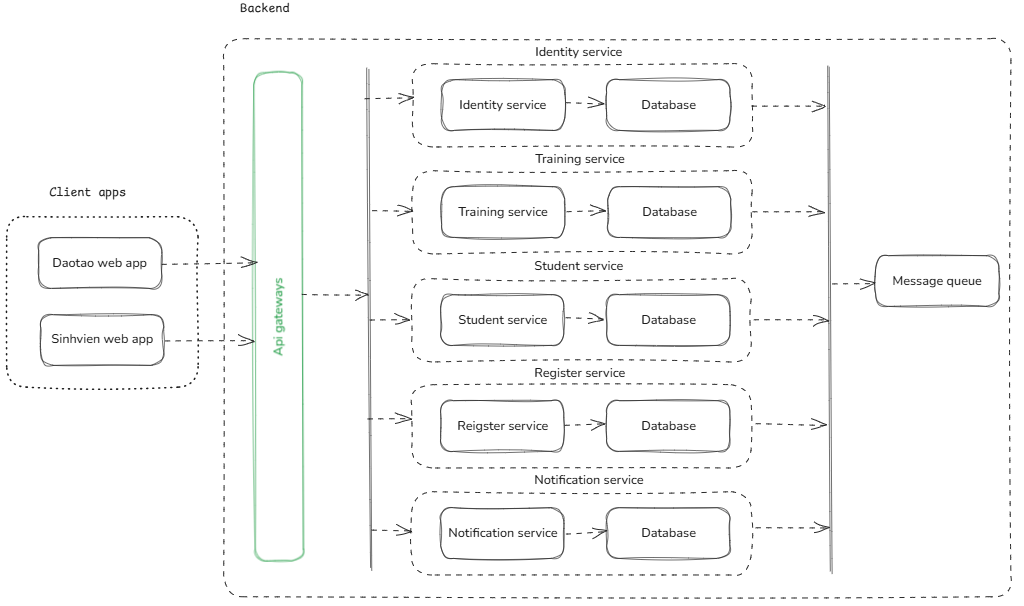
2.6.12

**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.

# CHƯƠNG 3 TRIỂN KHAI

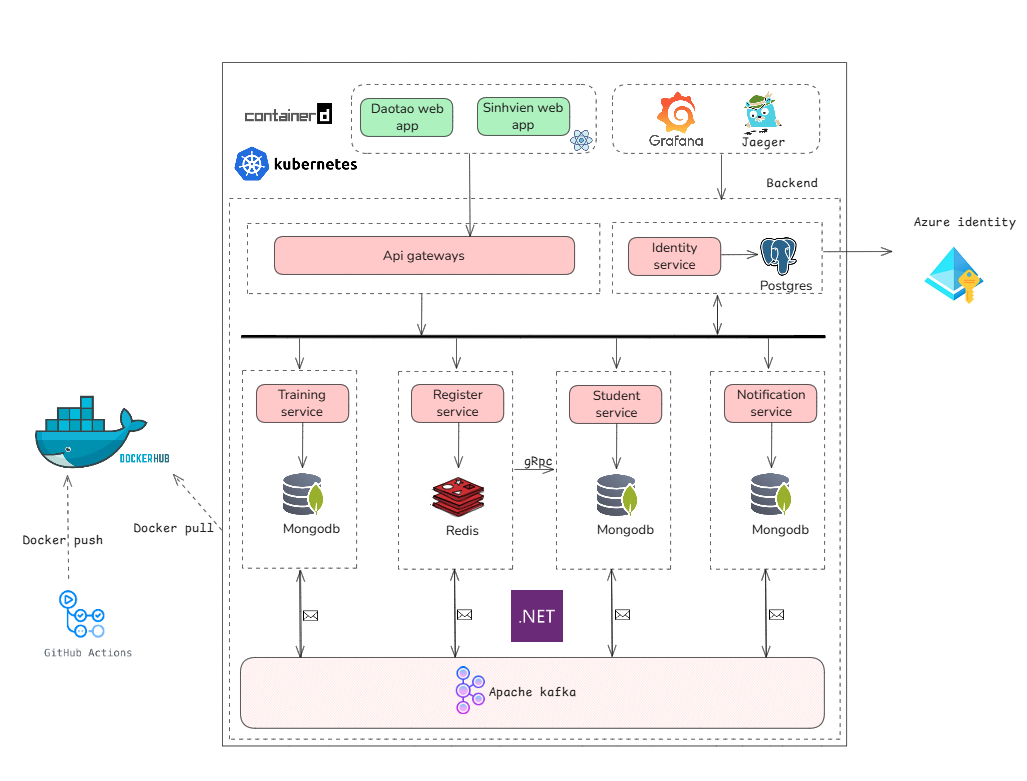
## 3.1 Sơ đồ thành phần



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).