

**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP. HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

**----🙣🕮🙡----**

A logo with a letter h and a flame

Description automatically generated

**MÔN: KIẾN TRÚC VÀ THIẾT KẾ PHẦN MỀM**

**BÁO CÁO ĐỒ ÁN**

***Đề tài: Lập trình ứng dụng quản lý sinh viên***

|  |
| --- |
|  |
| Nhóm: 19  Lớp học phần: DHKTPM16A - 420300154901  Giảng viên hướng dẫn: Võ Văn Hải  Thành viên:  Nguyễn Lê Mỹ Châu – 20046631  Nguyễn Thị Trung Hiếu - 20020381 |
|  |

**Mục Lục**

[**Chương 1: Giới Thiệu** 3](#_Toc167062431)

[**1. Tổng quan đề tài:** 3](#_Toc167062432)

[**2. Mô tả đề tài** 4](#_Toc167062433)

[**3. Mục tiêu đề tài:** 5](#_Toc167062434)

[**4. Mô tả yêu cầu chức năng** 5](#_Toc167062435)

[**5. Yêu cầu phi chức năng** 6](#_Toc167062436)

[**Chương 2: Cơ sở lý thuyết** 8](#_Toc167062437)

[**1. Java Spring Boot** 8](#_Toc167062438)

[**2. React** 9](#_Toc167062439)

[**3. MariaDB** 11](#_Toc167062440)

[**4. Postman** 12](#_Toc167062441)

[**CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ** 15](#_Toc167062442)

[**1. Class Diagram** 15](#_Toc167062443)

[**2. Software Architecture** 16](#_Toc167062444)

[**CHƯƠNG 4: HIỆN THỰC** 20](#_Toc167062445)

# **Chương 1: Giới Thiệu**

## **1. Tổng quan đề tài:**

Hiện nay, trong môi trường học tập và giảng dạy đại học, việc đăng ký học phần là một khâu quan trọng đối với sinh viên. Với sự phát triển không ngừng của công nghệ, quá trình này đã có nhiều thay đổi đáng kể, giúp sinh viên tiếp cận và quản lý học phần một cách dễ dàng và hiệu quả hơn.

Tuy nhiên, vẫn còn những hạn chế nhất định trong hệ thống đăng ký hiện tại. Việc phải thao tác trên các cổng thông tin phức tạp, giới hạn thời gian đăng ký, và sự thiếu minh bạch trong quá trình đăng ký có thể gây ra nhiều khó khăn và phiền toái cho sinh viên. Để giải quyết những vấn đề này, việc phát triển một ứng dụng đăng ký học phần trực tuyến trở nên cấp thiết hơn bao giờ hết.

Chuyển đổi sang nền tảng trực tuyến không chỉ mang lại sự tiện lợi mà còn tối ưu hóa tài nguyên và quy trình làm việc. Sinh viên có thể dễ dàng truy cập vào hệ thống đăng ký từ bất kỳ thiết bị nào có kết nối Internet mà không cần phải đến trực tiếp văn phòn. Điều này không chỉ tiết kiệm thời gian và công sức, mà còn giảm thiểu những rủi ro về mặt kỹ thuật và quản lý.

Với mục tiêu cải thiện trải nghiệm đăng ký học phần cho sinh viên, nhóm chúng tôi đã quyết định tập trung vào việc phát triển một phần mềm trực tuyến cho phép sinh viên quản lý thông tin cá nhân cũng như đăng ký học phần mỗi kỳ. Ứng dụng này không chỉ đơn thuần là công cụ đăng ký, mà còn là một giải pháp tổng thể giúp sinh viên theo dõi, quản lý và lên kế hoạch học tập một cách hiệu quả.

Bằng cách ứng dụng các công nghệ tiên tiến, chúng tôi hướng đến việc tạo ra một trải nghiệm người dùng mượt mà và linh hoạt. Ứng dụng sẽ hỗ trợ trực tuyến, giúp sinh viên dễ dàng truy cập và sử dụng mọi lúc, mọi nơi. Chúng tôi tin rằng đề tài này không chỉ đáp ứng nhu cầu hiện tại mà còn mở ra nhiều cơ hội phát triển cho việc quản lý học tập trong tương lai.

## **2. Mô tả đề tài**

Một trường đại học cần quản lý hệ thống đăng ký học phần của sinh viên theo theo cơ chế tín chỉ. Các sinh viên sẽ được biên chế ở các khoa theo mỗi ngành học của mình đăng ký.

Trong mỗi học kỳ, nhà trường sẽ chọn một số môn học của từng ngành học cho sinh viên đăng ký học. Sinh viên đăng ký theo học vào một lớp nhất định (lớp tín chỉ) với số sinh viên được đăng ký tối đa được qui định cho từng lớp (1 lớp tối đa bao nhiêu sinh viên).

Môn học được mở có thể là môn học tự chọn

Mỗi môn học có thể có môn học tiên quyết. Sinh viên sẽ không được đăng ký môn học mà có môn tiên quyết mình chưa học hay chưa pass.

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

Để phòng ngừa trường hợp sinh viên đăng ký xong rồi hủy bỏ, nhà trường yêu cầu sinh viên phải xác nhận trước khi đăng ký.

Vào ngày mở đăng ký, sinh viên sẽ đăng nhập vào hệ thống và sẽ nhìn thấy danh sách các môn học mà mình có khả năng đăng ký. Sinh viên chọn các môn học và tiến hành đăng ký. Trường hợp lớp đã đủ số lượng thì sinh viên sẽ không đăng ký được lớp đó mà phải chọn lớp khác

Sau khi đăng ký thành công, một email thông báo sẽ được gửi cho sinh viên xác nhận việc đăng ký và nhận quyết định đóng học phí.

Ngoài ra, hệ thống còn có khả năng cho phép sinh viên xem thông tin học tập của mình (số tín chỉ đã đạt, số môn đã học, điểm môn học, điểm trung bình tích lũy…), thời khóa biểu theo tuần và các tiện ích khác.

## **3. Mục tiêu đề tài:**

Mục tiêu của dự án là xây dựng một công cụ đăng ký học phần trực tuyến trên web, có khả năng cung cấp trải nghiệm tương tự như các hệ thống đăng ký hiện đại trên thị trường. Chúng tôi đặt ra mục tiêu tạo ra một môi trường đăng ký học phần dễ sử dụng, linh hoạt và đáng tin cậy cho sinh viên, đồng thời cung cấp các tính năng và chức năng tiện ích nhằm tối ưu hóa trải nghiệm người dùng.

Ngoài ra, chúng tôi cũng đặt mục tiêu phát triển một cộng đồng người dùng lớn mạnh, nơi sinh viên có thể trao đổi thông tin, chia sẻ kinh nghiệm học tập và hỗ trợ lẫn nhau một cách tự nhiên và thoải mái.

Chúng tôi cam kết cung cấp một giao diện thân thiện, các tính năng đa dạng trên trang web. Mục tiêu của chúng tôi không chỉ là đạt được tính tiện ích mà còn là tạo ra một môi trường đăng ký học phần trực tuyến sôi động và phong phú. Đây là một cơ hội để chúng tôi đóng góp vào việc tạo nên một không gian học tập và quản lý học phần hiện đại và hiệu quả.

## **4. Mô tả yêu cầu chức năng**

Công cụ sẽ tập trung vào việc cung cấp một trải nghiệm đăng ký học phần đa dạng và toàn diện cho sinh viên. Dưới đây là một số yêu cầu chức năng cơ bản mà chúng tôi dự định triển khai:

**1. Đăng ký học phần**: Sinh viên có thể dễ dàng tìm kiếm và đăng ký các học phần mở trong kỳ học một cách nhanh chóng.

**2. Xem lịch học**: Cho phép sinh viên xem lịch học cá nhân với các học phần đã đăng ký, bao gồm thời gian, địa điểm và giảng viên phụ trách.

**3. Xem thông tin chi tiết học phần**: Khả năng xem thông tin chi tiết về học phần, bao gồm mô tả, số tín chỉ, giảng viên và tài liệu liên quan.

**4. Bảo mật và riêng tư**: Đảm bảo tính riêng tư và bảo mật thông tin cá nhân của sinh viên thông qua các cơ chế xác thực và mã hóa dữ liệu.

**5. Xem các môn đã đăng ký**: Cho phép sinh viên theo dõi các môn đã đăng ký

**6. Xem điểm số**: Cho phép sinh viên xem điểm số các môn trong các kỳ sinh viên đã học

**7. Xem thông tin**: Cho phép sinh viên xem thông tin sinh viên, tổng số tín chỉ đã đạt và còn cần bao nhiêu tín chỉ nữa để xét tốt nghiệp

**8. Giao diện người dùng thân thiện**: Thiết kế giao diện đơn giản, dễ sử dụng và thân thiện với người dùng trên trang web.

## **5. Yêu cầu phi chức năng**

Dưới đây là một số yêu cầu phi chức năng mà chúng tôi sẽ xem xét trong quá trình phát triển công cụ đăng ký học phần:

**1. Hiệu suất**: Ứng dụng hoạt động mượt mà và nhanh chóng, với thời gian phản hồi nhanh và không có độ trễ lớn trong quá trình đăng ký và quản lý học phần.

**2. Độ tin cậy**: Ứng dụng cần đảm bảo tính ổn định và không gặp sự cố thường xuyên, giúp sinh viên cảm thấy tin tưởng và hài lòng khi sử dụng.

**3. Bảo mật**: Đảm bảo an toàn và bảo mật của dữ liệu sinh viên thông qua các biện pháp bảo mật như mã hóa dữ liệu và xác thực đáng tin cậy, bảo vệ thông tin cá nhân và học tập của người dùng.

**4. Dễ dàng sử dụng**: Ứng dụng cần có giao diện người dùng thân thiện và dễ sử dụng, cũng như quy trình đăng nhập, xem thông tin và đăng ký đơn giản, thuận tiện.

**5. Tuân thủ quy định và luật pháp**: Đảm bảo rằng ứng dụng tuân thủ các quy định pháp luật liên quan đến bảo vệ dữ liệu cá nhân và các quy định của nhà trường về quản lý học phần.

**6. Mở rộng và phát triển cộng đồng**: Xây dựng một cộng đồng người dùng sôi nổi và tích cực, với khả năng mở rộng và phát triển theo thời gian, giúp sinh viên dễ dàng giao lưu, chia sẻ kinh nghiệm học tập và hỗ trợ lẫn nhau.

**Chương 2: Cơ sở lý thuyết**

## **1. Java Spring Boot**



**Java Spring Boot là một Framework nhỏ hơn thuộc Spring Framework, giúp phát triển ứng dụng dựa trên ngôn ngữ lập trình Java, trong khi Java Spring MVC (Model-View-Controller) là một mô hình thiết kế phần mềm dựa trên Spring Framework. Java Spring Boot được thiết kế để giúp đơn giản hóa việc xây dựng các ứng dụng web và dịch vụ RESTful, trong khi Java Spring MVC là một mô hình thiết kế phần mềm được sử dụng để phát triển các ứng dụng web.**

**Ưu điểm của Java Spring Boot:**

1. **Tự động cấu hình**: Spring Boot cho phép tự động cấu hình ứng dụng dựa trên thư viện và các tùy chọn cấu hình mặc định. Điều này giúp giảm thiểu khối lượng công việc liên quan đến cấu hình thủ công.
2. **Tích hợp sẵn**: Spring Boot tích hợp sẵn các thư viện và công nghệ phổ biến như Spring Framework, Hibernate, Thymeleaf, và nhiều thứ khác để giúp bạn dễ dàng xây dựng ứng dụng mà không cần cài đặt nhiều thứ bổ sung.
3. **Tạo ứng dụng độc lập**: Spring Boot cho phép bạn tạo ra các ứng dụng chạy độc lập mà không cần phải triển khai vào các máy chủ ứng dụng bên ngoài như Tomcat hay Jetty. Ứng dụng có thể được đóng gói thành các file JAR hoặc WAR.
4. **Quản lý phụ thuộc**: Spring Boot sử dụng công cụ quản lý phụ thuộc như Maven hoặc Gradle để quản lý các thư viện và phiên bản phụ thuộc.
5. **Hỗ trợ viết mã ít và cấu trúc tốt**: Spring Boot giúp viết ít mã hơn và tập trung vào logic ứng dụng hơn là việc cấu hình. Nó cũng thúc đẩy việc tuân theo các nguyên tắc cấu trúc tốt để dễ dàng duy trì ứng dụng.
6. **Hỗ trợ cài đặt và triển khai**: Spring Boot cung cấp các công cụ hỗ trợ cài đặt và triển khai ứng dụng, bao gồm cả việc tạo file thực thi (executable) cho ứng dụng.

**Nhược điểm của Java Spring Boot:**

**1. Khởi động chậm hơn**: So với các framework nhẹ hơn, ứng dụng Spring Boot có thể mất nhiều thời gian hơn để khởi động, đặc biệt là với các ứng dụng lớn.

**2. Tài nguyên sử dụng nhiều**: Ứng dụng Spring Boot có thể sử dụng nhiều tài nguyên hơn do tích hợp nhiều tính năng và thư viện sẵn có.

**3. Độ phức tạp cao**: Dù Spring Boot giảm thiểu cấu hình, việc hiểu và sử dụng hiệu quả tất cả các tính năng của nó vẫn đòi hỏi kiến thức sâu rộng về Spring Framework và hệ sinh thái của nó.

**4. Bộ nhớ tiêu thụ**: Ứng dụng Spring Boot có thể tiêu thụ nhiều bộ nhớ hơn so với các giải pháp tối ưu hóa khác, điều này cần được xem xét khi triển khai trên các môi trường có tài nguyên hạn chế.

## **2. React**

A computer screen shot of a science model

Description automatically generated

React là một thư viện JavaScript phổ biến được sử dụng để xây dựng giao diện người dùng cho các ứng dụng web.

**Tổng quan về React:**

1. **Component - Based:** React sử dụng một cách tiếp cận

Component - based, cho phép lập trình viên chia nhỏ giao diện người dùng thành các thành phần độc lập và tái sử dụng được.

1. **Virtual DOM:** React sử dụng Virtual DOM để tối ưu hóa hiệu suất và tốc độ render của người dùng, giúp tạo ra trải nghiệm người dùng mượt mà và hiệu quả.
2. **JSX**: React sử dụng JSX ( JavaScript XML ) để tạo ra các component UI, giúp kết hợp giữa JavaScript và HTML một cách dễ dàng và linh hoạt.
3. **Unidirectional Data Flow:** React thúc đẩy việc sử dụng unidirectional data flow, giúp dễ dàng quản lý trạng thái của ứng dụng và tránh các vấn đề liên quan đến tính nhất quán của dữ liệu.

**Ưu điểm của React:**

1. **Tái sử dụng Component:** React cho phép lập trình viên tái sử dụng các components UI, giúp giảm thiểu mã lặp lại và tăng tính hiệu quả của mã nguồn.
2. **Virtual DOM:** Sử dụng Virtual DOM giúp tối ưu hóa hiệu suất và tốc độ render của ứng dụng, đồng thời giảm thiểu lượng tải lên trình duyệt.
3. **Community và Ecosystem:** React có một cộng đồng lớn và hệ sinh thái đa dạng của các thư viện và công cụ, giúp giảm thời gian phát triển và tăng tính linh hoạt của ứng dụng.

**Nhược điểm của React:**

1. **Khả năng học:** Đối với những người mới bắt đầu, việc học JSX và cách làm việc với React có thể là một thách thức, đặc biệt nếu họ không có kinh nghiệm trước đó với các khái niệm này.
2. **Cần phải sử dụng các thư viện bên thứ ba:** Mặc dù React cung cấp một cơ sở tốt cho việc xây dựng giao diện người dùng, nhưng nhiều tính năng phức tạp vẫn cần phải sử dụng các thư viện bên thứ ba.
3. **Khó khăn trong việc quản lý trạng thái ứng dụng:** Trong các ứng dụng lớn, việc quản lý trạng thái của ứng dụng có thể trở nên phức tạp và khó khăn.

## **3. MariaDB**



MariaDB là một hệ quản trị cơ sở dữ liệu mã nguồn mở được phát triển từ MySQL, nổi bật với tính năng mạnh mẽ, hiệu suất cao và tính bảo mật. MariaDB được sử dụng rộng rãi trong nhiều ứng dụng web và doanh nghiệp.

**Tổng quan về MariaDB:**

**1. Hiệu suất cao**: MariaDB cải thiện hiệu suất so với MySQL thông qua các tính năng như thread pooling và tối ưu hóa truy vấn.

**2. Mã nguồn mở**: Là một dự án mã nguồn mở, MariaDB cung cấp sự tự do và minh bạch cho người dùng và nhà phát triển.

**3. Tính tương thích cao**: MariaDB tương thích ngược với MySQL, giúp việc chuyển đổi và tích hợp trở nên dễ dàng hơn.

**4. Bảo mật nâng cao**: MariaDB cung cấp các tính năng bảo mật nâng cao, bao gồm kiểm soát truy cập chi tiết và mã hóa dữ liệu.

**Ưu điểm của MariaDB:**

**1. Hiệu suất và khả năng mở rộng**: MariaDB được thiết kế để tối ưu hóa hiệu suất và có thể mở rộng để xử lý các khối lượng dữ liệu lớn.

**2. Cộng đồng và hỗ trợ mạnh mẽ**: Với một cộng đồng lớn và tích cực, MariaDB cung cấp nhiều tài liệu, hỗ trợ và các công cụ hữu ích cho phát triển và vận hành.

**3. Cải tiến liên tục**: Dự án MariaDB liên tục phát triển và cải tiến, với các bản cập nhật và tính năng mới thường xuyên được phát hành.

**4. Dễ dàng chuyển đổi**: Do tính tương thích cao với MySQL, việc chuyển đổi từ MySQL sang MariaDB thường rất dễ dàng và không đòi hỏi nhiều thay đổi trong mã nguồn hoặc cấu trúc cơ sở dữ liệu.

**Nhược điểm của MariaDB:**

**1. Khả năng tương thích**: Mặc dù tương thích ngược với MySQL, một số tính năng mới của MariaDB có thể không tương thích hoàn toàn với MySQL.

**2. Tài nguyên sử dụng**: MariaDB có thể sử dụng nhiều tài nguyên hệ thống, đặc biệt là trong các môi trường lớn và phức tạp.

**3. Tài liệu phân tán**: Dù có một cộng đồng lớn, tài liệu về các tính năng nâng cao của MariaDB có thể không đầy đủ và khó tìm kiếm.

**4. Hỗ trợ thương mại**: Dù có nhiều dịch vụ hỗ trợ từ cộng đồng, hỗ trợ thương mại chuyên nghiệp của MariaDB có thể ít hơn so với các giải pháp cơ sở dữ liệu thương mại khác như Oracle.

MariaDB là một lựa chọn tuyệt vời cho các dự án yêu cầu một hệ quản trị cơ sở dữ liệu mạnh mẽ, hiệu suất cao và dễ dàng mở rộng, đồng thời tận dụng được sự linh hoạt và tự do của phần mềm mã nguồn mở.

## **4. Postman**



Postman là một công cụ phổ biến được sử dụng để kiểm thử API (Application Programming Interface). Nó cung cấp một giao diện thân thiện cho người dùng để tạo, chia sẻ, kiểm thử và quản lý các yêu cầu HTTP mà không cần viết mã. Postman hỗ trợ nhiều phương thức HTTP như GET, POST, PUT, DELETE, v.v., giúp dễ dàng kiểm thử và phát triển các API.

**Tổng quan về Postman**

* + - 1. **Giao diện và chức năng:**
* Giao diện đồ họa: Postman có giao diện người dùng trực quan, cho phép người dùng tạo và gửi các yêu cầu HTTP dễ dàng.
* Collection: Tổ chức và quản lý các yêu cầu thành các tập hợp (collections) để dễ dàng truy cập và chia sẻ.
* Environment: Quản lý các môi trường khác nhau (development, staging, production) với các biến số môi trường.
* Test script: Hỗ trợ viết các kịch bản kiểm thử tự động bằng JavaScript.
* Documentation: Tạo tài liệu tự động cho API trực tiếp từ các yêu cầu và phản hồi.
* Mock server: Tạo các mock server để mô phỏng các API và kiểm thử khi backend chưa sẵn sàng.

**2. Sử dụng trong quy trình phát triển:**

* Postman được sử dụng trong các giai đoạn phát triển phần mềm, từ thiết kế, phát triển, kiểm thử cho đến tài liệu hóa và bảo trì API.

**Ưu điểm của Postman**

**1. Dễ sử dụng:**

Giao diện thân thiện và dễ hiểu giúp người mới bắt đầu có thể làm quen và sử dụng nhanh chóng.

**2. Tính năng mạnh mẽ:**

Hỗ trợ đầy đủ các phương thức HTTP và nhiều loại dữ liệu khác nhau (JSON, XML, form data, v.v.).

Các tính năng như Collection, Environment, và Test script giúp quản lý và kiểm thử API hiệu quả.

**3. Khả năng tích hợp:**

Tích hợp dễ dàng với các hệ thống CI/CD và các công cụ khác như Newman (command-line runner cho Postman).

**4. Chia sẻ và cộng tác:**

Dễ dàng chia sẻ collections và environments với đồng nghiệp thông qua Postman Cloud hoặc xuất thành tệp.

**5. Tài liệu và cộng đồng:**

Tài liệu phong phú và cộng đồng người dùng lớn mạnh hỗ trợ nhanh chóng các vấn đề gặp phải

**Nhược điểm của Postman**

**1. Hiệu năng:**

Đối với các dự án lớn với hàng nghìn yêu cầu và script phức tạp, Postman có thể trở nên chậm và khó quản lý.

**2. Hạn chế về tính năng tự động hóa:**

Mặc dù Postman có hỗ trợ kiểm thử tự động bằng script, nhưng nó không mạnh mẽ bằng các framework kiểm thử chuyên dụng như JUnit hoặc TestNG.

**3. Giới hạn miễn phí:**

Phiên bản miễn phí có giới hạn về số lượng yêu cầu hàng tháng và các tính năng cao cấp như phân tích chi tiết, kiểm thử tải (load testing) chỉ có trong các gói trả phí.

**4. Phụ thuộc vào giao diện:**

Việc phụ thuộc vào giao diện đồ họa có thể là một bất lợi cho những người quen làm việc trên dòng lệnh hoặc cần tích hợp vào các quy trình tự động hóa phức tạp.

**CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ**

## **1. Class Diagram**

A diagram of a computer

Description automatically generated

## **2. Software Architecture**

**Service-based**

A diagram of a user interface

Description automatically generated

**1. Ưu điểm của mô hình service-based:**

**-** Dễ xây dựng, dễ triển khai, đơn giản và chi phí không quá cao so với các kiến trúc phân tán khác

- Tính Độc Lập và Linh Hoạt: Nếu một service gặp sự cố, các service khác vẫn hoạt động bình thường, tăng độ tin cậy của hệ thống 🡪 Nếu service đăng ký học phần gặp sự cố, có thể tạm thời dừng service này để sửa chữa mà không ảnh hưởng đến các service khác như xem điểm hay nhận nhắc nhở.

- Mở Rộng Độc Lập (Scalability): Các service hoạt động như một đơn vị độc lập. Khi nhu cầu sử dụng một dịch vụ cụ thể tăng cao, bạn có thể mở rộng chỉ dịch vụ đó mà không ảnh hưởng đến các dịch vụ khác 🡪 Trong một hệ thống quản lý học tập, nếu service đăng ký học phần đang nhận một lượng lớn yêu cầu, có thể mở rộng dịch vụ này bằng cách tăng số lượng máy chủ chỉ dành cho dịch vụ này, mà không cần phải thay đổi hoặc mở rộng các dịch vụ khác như dịch vụ xem điểm hay dịch vụ lịch học.

- Cập Nhật Độc Lập (Independent Updates): Mỗi service trong kiến trúc dựa trên dịch vụ có thể được phát triển, triển khai và cập nhật một cách độc lập. Điều này có nghĩa là có thể nâng cấp, sửa lỗi hoặc thêm tính năng mới vào một dịch vụ cụ thể mà không cần phải dừng hoặc ảnh hưởng đến toàn bộ hệ thống. 🡪 Nếu bạn cần cập nhật tính năng của dịch vụ xem thông tin cá nhân (chẳng hạn như cải thiện bảo mật), bạn có thể triển khai phiên bản mới của dịch vụ này mà không cần phải dừng hệ thống hay cập nhật các dịch vụ khác như dịch vụ đăng ký học phần hoặc dịch vụ xem lịch học.

- Sự tách biệt và độc lập của các service trong kiến trúc service-based làm cho việc triển khai và quản lý trở nên linh hoạt hơn. Có thể quản lý và triển khai các dịch vụ một cách độc lập, giảm thiểu rủi ro và đơn giản hóa quy trình quản lý 🡪 Khi triển khai một tính năng mới, có thể triển khai và kiểm thử service đó mà không cần phải ngừng hoạt động của các dịch vụ khác

- Các dịch vụ trong kiến trúc service-based thường có quyền truy cập trực tiếp vào cơ sở dữ liệu, giúp đảm bảo tính toàn vẹn dữ liệu. Điều này quan trọng đối với các tính năng như đăng ký học phần, quản lý kết quả, v.v..

**2. Hạn chế của mô hình service-based:**

**-** Truy cập trực tiếp vào database có thể tạo ra rủi ro bảo mật cao hơn, đặc biệt khi không kiểm soát chặt chẽ quyền truy cập của các dịch vụ. Nếu một dịch vụ bị tấn công, hacker có thể có khả năng truy cập vào dữ liệu quan trọng của hệ thống 🡪Ví dụ, trong hệ thống quản lý sinh viên, thông tin nhạy cảm như thông tin cá nhân, điểm số, hoặc thông tin tài khoản ngân hàng có thể được lưu trữ trong cơ sở dữ liệu. Nếu một dịch vụ trong mô hình dựa trên dịch vụ không được bảo vệ chặt chẽ và bị tấn công, hacker có thể tận dụng lỗ hổng để truy cập trái phép vào database và đánh cắp thông tin của sinh viên

- Trong thời điểm đăng ký học phần, nếu hệ thống dựa trên dịch vụ không được thiết kế để chịu tải cao, nó có thể gặp phải vấn đề quá tải khi có số lượng lớn sinh viên cùng truy cập đồng thời. Ví dụ, các dịch vụ xử lý đăng ký có thể trở nên chậm chạp hoặc không phản hồi do không đủ khả năng xử lý lượng yêu cầu lớn cùng một lúc được 🡪 Không có tính chịu tải cao so với các kiến trúc phân tán khác như microservices hay event-driven

- Trong mô hình dựa trên dịch vụ, thông tin của sinh viên có thể được phân tán qua nhiều dịch vụ khác nhau, chẳng hạn như dịch vụ quản lý hồ sơ, dịch vụ đăng ký học phần, và dịch vụ tài chính. Nếu một trong những dịch vụ này gặp sự cố và mất dữ liệu, điều này có thể dẫn đến mất mát thông tin quan trọng và không nhất quán giữa các phần của hệ thống

- Mặc dù mô hình dựa trên service cho phép mở rộng dễ dàng bằng cách thêm service mới, việc quản lý một hệ thống lớn với nhiều dịch vụ có thể trở nên phức tạp. Mỗi dịch vụ mới được thêm vào đều tạo ra một điểm độc lập 🡪 Đòi hỏi sự quản lý cẩn thận nếu không có thể ảnh hưởng đến độ tin cậy của hệ thống và khả năng mở rộng linh hoạt trong tương lai.

**3. Một số biện pháp giảm thiểu rủi ro**

- Tăng cường bảo mật dữ liệu: Sử dụng các biện pháp bảo mật như mã hóa dữ liệu, quản lý danh tính và phân quyền truy cập để đảm bảo rằng chỉ những người được ủy quyền mới có thể truy cập vào dữ liệu nhạy cảm của sinh viên. Thực hiện kiểm tra bảo mật định kỳ, đảm bảo rằng các lỗ hổng bảo mật được phát hiện và khắc phục kịp thời.

- Nâng cấp khả năng chịu tải: Tối ưu hóa hiệu suất và mở rộng hạ tầng hệ thống để đảm bảo rằng nó có thể chịu được lượng truy cập lớn trong các thời điểm cao điểm như đăng ký học phần. Sử dụng các giải pháp như tải cân bằng và tự động mở rộng để phản ứng linh hoạt với tải cao.

- Chia thời gian đối với các service dễ bị quá tải như đăng ký học phần: Có thể chia theo khoa hoặc theo khoá (chia nhỏ lượt truy cập) để tiến hành đăng ký học phần trong một khung thời gian nhất định

- Xây dựng các quy trình sao lưu và phục hồi dữ liệu định kỳ để đảm bảo rằng thông tin quan trọng không bị mất mát trong trường hợp có sự cố.

# **CHƯƠNG 4: HIỆN THỰC**

**1. Màn hình đăng nhập**

A screenshot of a computer login

Description automatically generated

**2. Màn hình trang chủ**: Hiển thị thông tin của sinh viên và số tín chỉ sinh viên đã đạt cùng tổng số tín chỉ của sinh viên cần đạt để đủ điều kiện xét tốt nghiệp. Đồng thời có các lựa chọn để truy cập đến các tính năng của trang

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**3. Màn hình xem lịch học:**

- Đây là các môn đã đăng ký trong kỳ này:

A screenshot of a computer

Description automatically generated

- Đây là môn của kỳ trước

A screenshot of a computer

Description automatically generated

**4. Màn hình xem điểm**: Hiển thị điểm các kỳ đã học, sau khi có điểm cuối kỳ các môn mới hiển thị điểm trung bình và điểm tích luỹ. Và sau khi điểm được nhập thì sẽ update số tín chỉ của sinh viên

A screenshot of a computer

Description automatically generated

**5. Màn hình đăng ký học phần:**

- Đợt đăng ký sẽ là kỳ hiện tại (ban đầu được đặt theo cơ chế: Trước tháng 7 là học kỳ 2, sau tháng 7 là học kỳ 1 nhưng để thuận tiện cho việc test đăng ký nên việc này được dời đến tháng 5). Và danh sách đợt đăng ký này sẽ lấy từ HK1 năm sinh viên nhập học đến hiện tại)

- Hiển thị danh sách các môn học và hiển thị các môn sinh viên đã đăng ký trong kỳ này

- Trong trường hợp sinh viên muốn chọn kỳ trước đó và kỳ sau đó so với kỳ hiện tại sẽ không được chấp nhận khi chưa đến thời gian

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

- Khi chọn môn học muốn đăng ký 🡪 Hiển thị các lớp và thông tin của lớp

A screenshot of a computer

Description automatically generated

- Chọn lớp muốn đăng ký khi trùng lịch học Thứ 4 Tiết 4-6 sẽ hiển thị thông báo cho biết

A screenshot of a computer

Description automatically generated

- Và khi chọn lớp đã đầy thì yêu cầu sẽ bị từ chối cũng như hiển thị ra cho sinh viên biết

A screenshot of a computer

Description automatically generated

- Khi xét đủ các điều kiện cho thấy có thể đăng ký thì hệ thống sẽ hỏi sinh viên có xác nhận đăng ký không và nếu sinh viên bấm xác nhận thì sẽ hiển thị thông báo sinh viên được đăng ký vào lớp học

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

- Email cũng được gửi đến cho việc đăng ký thành công môn và tổng công nợ (có thể chỉnh sửa thành sau khi hết thời gian đăng ký học phần sẽ gửi email tất cả môn và tổng công nợ nhưng để test nên sẽ gửi vào mỗi môn đăng ký thành công)

A black text on a white background

Description automatically generated

- Huỷ đăng ký học phần và khi nhấn xác nhận thì hiển thị thông báo và môn sẽ được huỷ

A screenshot of a computer

Description automatically generated

- Chưa học môn tiên quyết sẽ không cho đăng ký

A screenshot of a computer

Description automatically generated

- Ràng buộc 30 tín chỉ nhưng không đủ 30 nên sửa lại thành 8 để test (sau khi huỷ môn chỉ còn 2 môn với 6 tín chỉ hiện tại) A screen shot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated