**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP.HCM**

A logo with a red and blue letter and a drop of water

Description automatically generated

HỆ THỐNG QUẢN LÍ ĐĂNG KÍ

HỌC PHẦN

**Giảng Viên: Võ Văn Hải**

***Nhóm 21***

Ngô Quốc Dũng - 19491261

Trịnh Phú Quý - 20108831

Trần Công Minh - 20121801

# MỤC LỤC

HỆ THỐNG QUẢN LÍ ĐĂNG KÍ HỌC PHẦN ONLINE

[MỤC LỤC 2](#_Toc167567875)

[Chương 1: GIỚI THIỆU 2](#_Toc167567876)

[**1.1** **Tổng quan** 2](#_Toc167567877)

[**1.2** **Mục tiêu đề tài** 2](#_Toc167567878)

[ **Phạm vi đề tài** 2](#_Toc167567879)

[**1.3** **Mô tả yêu cầu chức năng** 3](#_Toc167567880)

[Chương 2: CƠ SỞ LÝ THUYẾT 4](#_Toc167567881)

[**1.1.** **Phân tích lựa chọn kiến trúc** 4](#_Toc167567882)

[1.2. Thuận lợi và khó khăn. 6](#_Toc167567883)

[1.3. Các công nghệ sử dụng 6](#_Toc167567884)

[Chương 3: PHÂN TÍCH VÀ THIẾT KẾ 8](#_Toc167567885)

[3.1. Sơ đồ luồng – Dataflow diagram: 8](#_Toc167567886)

[3.2. Sơ đồ lớp - Class diagram 9](#_Toc167567887)

[3.3. Áp dụng kiến trúc được chọn/ thiết kế: 10](#_Toc167567888)

[Chương 4: HIỆN THỰC 11](#_Toc167567889)

[4.1. Cấu hình phần cứng, phần mềm. 11](#_Toc167567890)

[4.2. Giao diện hệ thống 11](#_Toc167567891)

[Chương 5: KẾT LUẬN 13](#_Toc167567892)

[**5.1. Kết quả đạt được** 13](#_Toc167567893)

[5.2. Những phần chưa đạt được. 14](#_Toc167567894)

[5.3. Hướng phát triển. 14](#_Toc167567895)

# Chương 1: GIỚI THIỆU

* 1. **Tổng quan**
* Trong bối cảnh giáo dục hiện đại, việc triển khai hệ thống đăng ký học phần online theo hệ thống tín chỉ đã trở thành một yêu cầu thiết yếu của các trường đại học. Hệ thống này không chỉ giúp cải thiện quy trình đăng ký môn học, mà còn đảm bảo tính công bằng và minh bạch trong việc phân bổ tài nguyên giảng dạy, đồng thời theo dõi tiến độ học tập của sinh viên một cách hiệu quả. Đề tài này sẽ tập trung vào việc xây dựng một hệ thống đăng ký học phần online theo hệ thống tín chỉ, nhằm đáp ứng các nhu cầu của trường đại học trong việc quản lý học phần, lớp học và thông tin học tập của sinh viên.
  1. **Mục tiêu đề tài**
* Tối ưu hóa quy trình đăng ký môn học
* Xây dựng một hệ thống đăng ký học phần trực tuyến giúp sinh viên dễ dàng lựa chọn và đăng ký các môn học phù hợp với kế hoạch học tập cá nhân.
* Giảm thiểu sai sót và tăng tính chính xác trong quá trình đăng ký môn học**.**
* Cung cấp giao diện thân thiện và dễ sử dụng, giúp sinh viên dễ dàng quản lý thời khóa biểu và theo dõi tiến độ học tập.
* Tích hợp chức năng thông báo và nhắc nhở, giúp sinh viên cập nhật thông tin về các môn học và lịch học một cách nhanh chóng.
* Tích hợp chức năng thanh toán học phí trực tuyến, giúp sinh viên dễ dàng hoàn tất các thủ tục tài chính.
* **Phạm vi đề tài**
* Phạm vi của đề tài bao gồm:
  + Đăng ký môn học trực tuyến: Xây dựng hệ thống cho phép sinh viên đăng ký môn học qua mạng, bao gồm việc lựa chọn môn học, kiểm tra điều kiện tiên quyết, và xác nhận đăng ký.
  + Quản lý thời khóa biểu: Tự động cập nhật và quản lý thời khóa biểu cá nhân của sinh viên dựa trên các môn học đã đăng ký.
  + Theo dõi tiến độ học tập: Cung cấp chức năng theo dõi và báo cáo tiến độ học tập của sinh viên, bao gồm số tín chỉ đã hoàn thành, các môn học bắt buộc, và các yêu cầu tốt nghiệp.
  + Quản lý tài nguyên giảng dạy: Hỗ trợ nhà trường trong việc phân bổ giảng viên, phòng học, và các tài nguyên khác một cách hiệu quả.
  + Thanh toán học phí: Tích hợp chức năng thanh toán học phí trực tuyến, giúp sinh viên dễ dàng hoàn tất các thủ tục tài chính liên quan đến đăng ký học phần.
  + Hỗ trợ kỹ thuật: Cung cấp các dịch vụ hỗ trợ kỹ thuật để giải quyết các vấn đề mà sinh viên và giảng viên có thể gặp phải trong quá trình sử dụng hệ thống.
  1. **Mô tả yêu cầu chức năng**
* Hệ thống quản lý lớp học tín chỉ cần đảm bảo các yêu cầu chức năng chính sau:
* Services Đăng Ký Tài Khoản và Đăng Nhập:
  + Cho phép sinh viên tạo tài khoản mới và đăng nhập vào hệ thống.
  + Bảo mật thông tin tài khoản và hỗ trợ chức năng khôi phục mật khẩu.
* Services Đăng ký môn học trực tuyến:
  + Sinh viên có thể xem danh sách các môn học được mở cho kỳ học hiện tại.
  + Cung cấp thông tin chi tiết về từng môn học như số tín chỉ, giảng viên, thời gian, địa điểm và mô tả môn học.
  + Cho phép sinh viên chọn các môn học mà họ muốn đăng ký.
  + Sau khi chọn môn học, sinh viên sẽ nhận được thông báo xác nhận đăng ký thành công.
  + Hệ thống gửi email hoặc thông báo qua ứng dụng về thông tin đăng ký môn học.
* Services Quản lý thời khóa biểu:
  + Tự động cập nhật thời khóa biểu cá nhân:
    - Hệ thống sẽ tự động cập nhật thời khóa biểu của sinh viên dựa trên các môn học đã đăng ký.
    - Hiển thị thời khóa biểu dưới dạng lịch tuần/tháng, cho phép sinh viên dễ dàng theo dõi.
  + Tránh xung đột thời gian:
    - Hệ thống sẽ kiểm tra và cảnh báo sinh viên nếu có xung đột về thời gian giữa các môn học đã chọn.
    - Đề xuất các môn học thay thế nếu có xung đột thời gian.
* Services Theo dõi tiến độ học tập:
  + Báo cáo tiến độ học tập:
    - Hệ thống cung cấp các báo cáo về số tín chỉ đã hoàn thành, số tín chỉ còn lại, và các môn học bắt buộc.
    - Hiển thị biểu đồ tiến độ học tập, giúp sinh viên dễ dàng theo dõi quá trình hoàn thành chương trình học.
  + Thông báo nhắc nhở:
    - Gửi thông báo nhắc nhở về các môn học cần hoàn thành, hạn đăng ký, và các mốc thời gian quan trọng trong kỳ học.
* Services Thanh toán học phí trực tuyến:
  + Tính toán và hiển thị học phí:
    - Hệ thống tự động tính toán học phí dựa trên số tín chỉ đăng ký và các khoản phí liên quan khác.
    - Hiển thị thông tin học phí chi tiết cho sinh viên trước khi xác nhận đăng ký.
  + Thanh toán trực tuyến:
    - Tích hợp các cổng thanh toán trực tuyến như thẻ tín dụng,

# Chương 2: CƠ SỞ LÝ THUYẾT

* 1. **Phân tích lựa chọn kiến trúc**
* **Microservice Architecture**.
* Chọn kiến trúc **Microservices** cho phần mềm quản lý hệ thống đăng ký học phần online có thể mang lại nhiều lợi ích so với các kiến trúc khác như kiến trúc nguyên khối (Monolithic) hay kiến trúc theo lớp (Layered). Dưới đây là một số lý do chính:
  + **Tính mô-đun và khả năng mở rộng**
    - **Tính mô-đun**: Microservices chia nhỏ hệ thống thành các dịch vụ độc lập, mỗi dịch vụ đảm nhận một chức năng cụ thể. Điều này cho phép các nhóm phát triển có thể làm việc trên các dịch vụ khác nhau một cách độc lập, giảm thiểu xung đột và tăng hiệu quả làm việc.
    - **Khả năng mở rộng**: Với Microservices, từng dịch vụ có thể được mở rộng độc lập dựa trên nhu cầu. Ví dụ, dịch vụ đăng ký môn học có thể được mở rộng thêm nhiều phiên bản trong mùa đăng ký cao điểm mà không ảnh hưởng đến các dịch vụ khác
* **Tính linh hoạt và đổi mới nhanh chóng**
* **Linh hoạt**: Các dịch vụ có thể được phát triển và triển khai bằng các công nghệ khác nhau phù hợp nhất với yêu cầu cụ thể của từng dịch vụ. Điều này cho phép sử dụng công nghệ tối ưu nhất cho mỗi phần của hệ thống.
* **Đổi mới nhanh chóng**: Do các dịch vụ được phát triển và triển khai độc lập, việc thêm mới tính năng hay cải tiến hệ thống có thể được thực hiện nhanh chóng mà không cần thay đổi toàn bộ hệ thống.
* **Dễ bảo trì và khắc phục sự cố**
* **Dễ bảo trì**: Với Microservices, mỗi dịch vụ là một đơn vị nhỏ và dễ quản lý. Khi có sự cố xảy ra, việc tìm kiếm và khắc phục lỗi chỉ cần tập trung vào dịch vụ cụ thể thay vì phải xem xét toàn bộ hệ thống.
* **Tự phục hồi**: Hệ thống Microservices có thể được thiết kế để có khả năng tự phục hồi. Nếu một dịch vụ gặp sự cố, các dịch vụ khác vẫn tiếp tục hoạt động, đảm bảo tính liên tục của hệ thống.
* **Quản lý tài nguyên và chi phí hiệu quả**
* **Quản lý tài nguyên**: Các dịch vụ nhỏ hơn dễ dàng theo dõi và quản lý tài nguyên, giúp tối ưu hóa hiệu suất và sử dụng tài nguyên hiệu quả.
* **Chi phí hiệu quả**: Bằng cách mở rộng và thu nhỏ các dịch vụ dựa trên nhu cầu thực tế, doanh nghiệp có thể giảm chi phí hạ tầng và tối ưu hóa chi phí vận hành.
  1. Thuận lợi và khó khăn.
* **Thuận lợi:**
  + Hệ thống được chia nhỏ thành các dịch vụ độc lập như dịch vụ quản lý môn học, dịch vụ đăng ký, dịch vụ thanh toán, dịch vụ thông báo, vv. Điều này giúp các nhóm phát triển làm việc đồng thời trên các dịch vụ khác nhau mà không gây xung đột.
  + Các dịch vụ có thể được mở rộng độc lập dựa trên nhu cầu. Ví dụ, trong mùa đăng ký cao điểm, chỉ cần mở rộng dịch vụ đăng ký môn học mà không ảnh hưởng đến các dịch vụ khác.
  + Các dịch vụ có thể được phát triển bằng các công nghệ khác nhau, phù hợp nhất với yêu cầu của từng dịch vụ cụ thể. Ví dụ, sử dụng Node.js cho dịch vụ đăng ký để đảm bảo xử lý nhanh và Java cho dịch vụ thanh toán để đảm bảo tính bảo mật.
  + Việc triển khai và nâng cấp các dịch vụ riêng lẻ có thể được thực hiện nhanh chóng mà không cần phải đợi toàn bộ hệ thống sẵn sàng.
  + Việc quản lý và bảo trì từng dịch vụ nhỏ lẻ dễ dàng hơn so với hệ thống nguyên khối lớn. Khi có lỗi xảy ra, việc xác định và khắc phục sự cố cũng trở nên nhanh chóng và chính xác hơn.
* **Khó khăn – Trade off:**
* **Phức tạp hóa kiến trúc**: Việc chia nhỏ hệ thống thành nhiều dịch vụ dẫn đến sự phức tạp trong quản lý và phát triển. Mỗi dịch vụ phải được quản lý độc lập, bao gồm việc xây dựng, triển khai, và bảo trì, đòi hỏi các nhóm phát triển phải có kỹ năng và kiến thức sâu rộng về Microservices.
* **Quản lý phiên bản và phụ thuộc**: Mỗi dịch vụ có thể có các phiên bản và phụ thuộc riêng, tạo ra sự phức tạp trong việc quản lý và đồng bộ hóa. Việc kiểm soát và đảm bảo các dịch vụ tương thích với nhau cần một chiến lược rõ ràng và công cụ hỗ trợ hiệu quả.
* **Bảo mật**: Mỗi dịch vụ phải được bảo mật riêng, bao gồm xác thực, ủy quyền, và mã hóa dữ liệu. Việc quản lý bảo mật cho nhiều dịch vụ có thể phức tạp và đòi hỏi một chiến lược bảo mật toàn diện.
  1. Các công nghệ sử dụng
     1. Java
* Java là một lựa chọn phổ biến cho phát triển hệ thống đăng ký học phần do tính đa nền tảng và khả năng mở rộng cao của nó.
  + 1. JWT (JSON Web Token)
* JWT (JSON Web Token) là một tiêu chuẩn mã nguồn mở (RFC 7519) dùng để truyền tải thông tin an toàn, gọn nhẹ và khép kín giữa các bên tham gia dưới format JSON. Thông tin được chia sẻ trong JWT được xác thực và tin cậy thông qua chữ ký số (Digital signature).
  + 1. Spring Framework:
* Spring cung cấp một cơ sở cho việc xây dựng ứng dụng Java theo mô hình MVC (Model-View-Controller), cũng như các tính năng như Dependency Injection và Spring Boot giúp tạo ra các ứng dụng tự chạy.
  + 1. Postman
* Postman hiện là một trong những công cụ phổ biến nhất được sử dụng trong thử nghiệm các API. Với Postman, ta có thể gọi Rest API mà không cần viết dòng code nào.
  + 1. MariaDB
* MariaDB là một hệ quản trị cơ sở dữ liệu quan hệ, là một nhánh của MySQL. MariaDB được thiết kế để có khả năng tương thích cao với MySQL, MariaDB hỗ trợ các tính năng mở rộng như sharding, replication, và clustering, giúp cho hệ thống có thể mở rộng dễ dàng khi cần thiết, đảm bảo khả năng mở rộng linh hoạt và hiệu quả.
  + 1. Github
* GitHub là một nền tảng lưu trữ mã nguồn mở và dự án phần mềm trực tuyến. Nó cung cấp các công cụ quản lý phiên bản (version control), quản lý dự án, và hợp tác cho các nhà phát triển phần mềm.
  + 1. NextJS:
* Framework JavaScript mạnh mẽ và phổ biến, cung cấp các công cụ và thư viện để xây dựng giao diện người dùng phong phú và linh hoạt.

# Chương 3: PHÂN TÍCH VÀ THIẾT KẾ

## 3.1. Sơ đồ luồng – Dataflow diagram:

A diagram of a student

Description automatically generated

## 3.2. Sơ đồ lớp - Class diagram

A computer screen shot of a computer

Description automatically generated

## 3.3. Áp dụng kiến trúc được chọn/ thiết kế:

A computer screen shot of a diagram

Description automatically generated

.

# Chương 4: HIỆN THỰC

## 4.1. Cấu hình phần cứng, phần mềm.

## 4.2. Giao diện hệ thống

* A screen shot of a login screen

  Description automatically generatedMàn hình đăng nhập: Đăng nhập = username và password được cấp
* A close-up of a screen

  Description automatically generatedTrang Thông tin Cá nhân
* Trang Đăng ký học phần

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

* A screenshot of a computer

  Description automatically generatedTrang Chi tiết lớp học phần:
* Trang Danh sách lớp Học đã đăng ký:

A screenshot of a computer

Description automatically generated

# Chương 5: KẾT LUẬN

## **5.1. Kết quả đạt được**

* Làm giao diện dễ sử dụng.
* Làm các chức năng của hệ thống như:
  + Đăng nhập
  + Xem thông tin sinh viên
  + Hiển thị danh sách môn
  + Xem danh sách lớp học đã đăng ký

## 5.2. Những phần chưa đạt được.

* Chưa làm chức năng xem điểm, xem thời khóa điểm, xem chương trình khung
* Chưa làm được chức năng thanh toán học phí

## 5.3. Hướng phát triển.

* Hoàn thiện những chức năng còn chưa xong.
* Thêm chức năng đăng nhập cho giảng viên.
* Thêm mục tin tức, thông tin thêm về trường. Thông tin thêm ngoài lề cho sinh viên.
* Thêm trang chủ thể hiện các sơ đồ học phần, điểm

# 