**FPT APTECH**



**TÊN ĐỀ TÀI: ỨNG DỤNG ĐĂNG KÝ HỌC PHẦN**

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

**Lớp học phần:**

**Giảng viên bộ môn:**

**Nhóm 07 - Sinh viên thực hiện:**

T1.2208.M2

Trần Phước Sinh

Bùi Trần Anh Trí - Student1414230

MỤC LỤC

[I. GIỚI THIỆU 3](#_Toc167057580)

[1. Mô tả yêu cầu cơ bản 3](#_Toc167057581)

[2. Kiến trúc 3](#_Toc167057582)

[II. PHÂN TÍCH & THIẾT KẾ 10](#_Toc167057583)

[1. Architecture diagram 10](#_Toc167057584)

[2. Use case diagram 12](#_Toc167057585)

[3. Class diagram 13](#_Toc167057586)

[3.1. Dịch vụ đăng ký học phần 13](#_Toc167057587)

[3.2. Dịch vụ thông tin sinh viên 14](#_Toc167057588)

[3.3. Dịch vụ cựu sinh viên 15](#_Toc167057589)

[3.4. Dịch vụ tra cứu công nợ 16](#_Toc167057590)

[4. Components diagram 16](#_Toc167057591)

[5. Package diagram 17](#_Toc167057592)

[III. HIỆN THỰC 18](#_Toc167057593)

[IV. KẾT LUẬN 18](#_Toc167057594)

[1. Kết quả 18](#_Toc167057595)

[2. Hướng phát triển trong tương lai 18](#_Toc167057596)

# GIỚI THIỆU

## Mô tả yêu cầu cơ bản

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.

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.

Sau khi số tín chỉ đã đạt theo từng ngành, sinh viên có quyền đăng ký xét tốt nghiệp. Nếu mọi tiêu chuẩn đều thỏa mãn, sinh viên sẽ được cấp bằng tốt nghiệp và sẽ được đưa vào danh sách các cựu sinh viên. Thông tin về bằng cấp sẽ được công khai trên trang web của nhà trường. Thông tin của cựu sinh viên sẽ được lưu giữ để theo dõi quá trình làm việc (nếu sinh viên đồng ý), làm các cuộc survey, cũng như nhiều hoạt động khác.

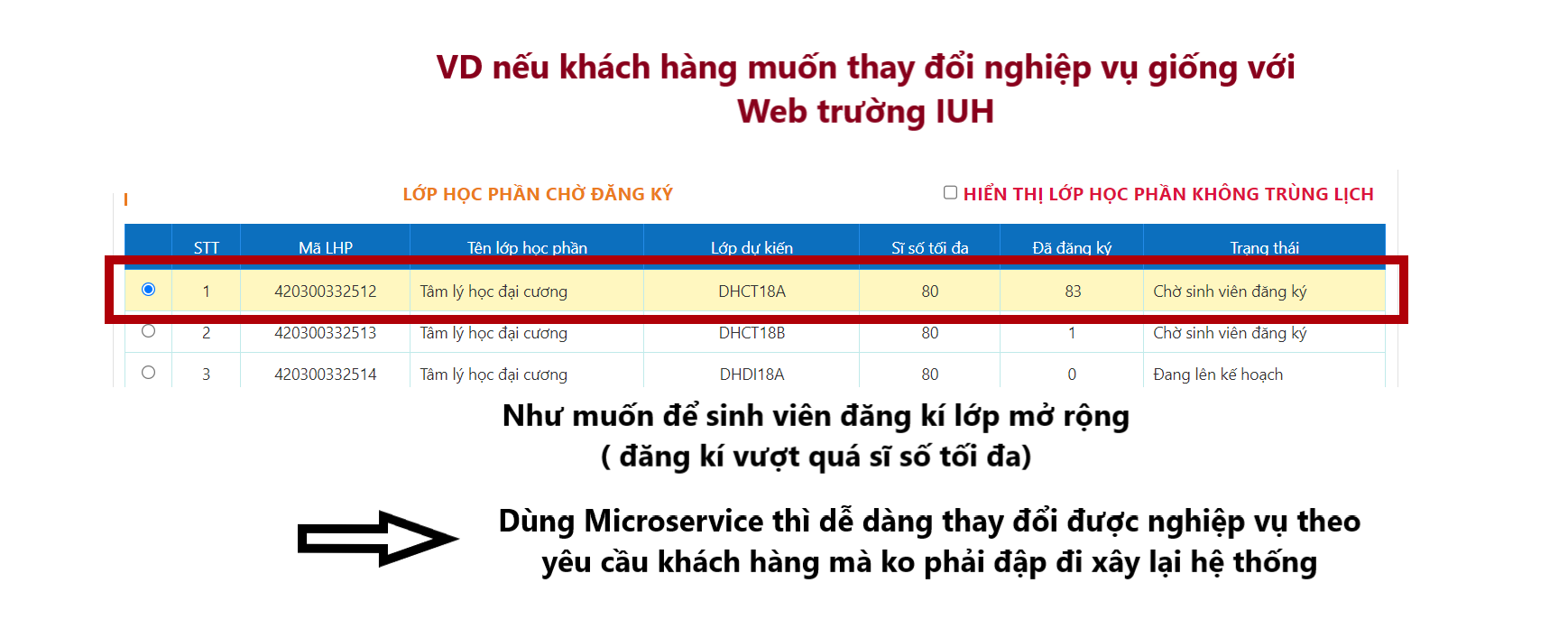
## Kiến trúc

Với những mô tả theo yêu cầu, em đưa ra 1 thiết kế hệ thống theo mô hình **Microservice**.

* Đối với 1 hệ thống lớn như quản lý học phần việc bổ sung tính năng mới hay sửa đổi nghiệp vụ là điều không thể tránh khỏi.

Việc dùng Microservice giúp dễ dàng tích hợp tính năng mới và không ảnh hưởng đến hệ thống.

Ví dụ sau khi hệ thống đã được deploy ra thị trường mà nhà trường muốn thay đổi 1 số luồng ngoại lệ như là : Chèn sinh viên dẫu lớp đã đủ số lượng. Lúc này dùng micro service sẽ giúp ta dễ dàng thêm, sửa tính năng mới.

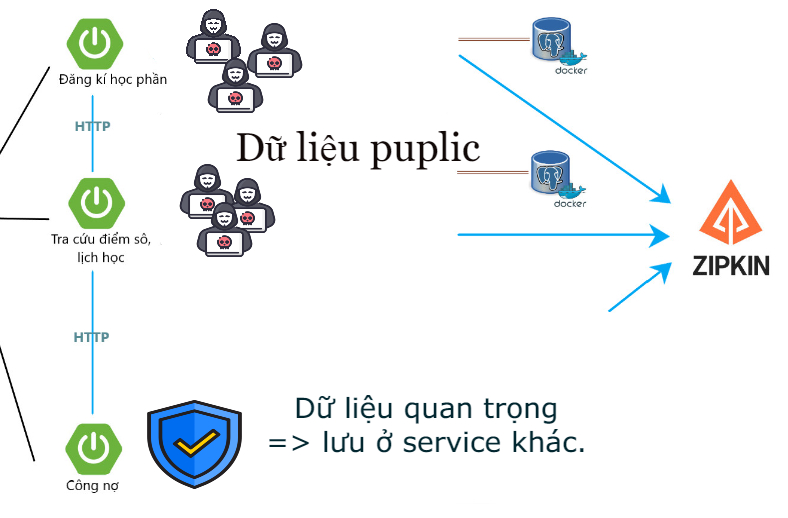


* Khả năng chịu lỗi cao hơn. Nếu có 1 dịch vụ bị lỗi, các dịch vụ khác vẫn hoạt động bình thường.

A screen shot of a computer error

Description automatically generated

* Các dữ liệu quan trọng được lưu ở các service riêng, đảm bảo tính an toàn của dữ liệu



* 1 Service có nhiều vùng dữ liệu.

VD service đăng kí học phần dữ liệu môn học đăng kí được phân vùng theo từng khoa. => Đỡ dành nhau => Cải thiện hiệu suất.

* Nâng cấp và bảo trì: nâng cấp hay bảo trì 1 dịch vụ sẽ không làm ảnh hưởng đến các dịch vụ khác.

A group of images of sandwiches

Description automatically generated

Sinh viên ngành công nghệ thông tin khi đăng kí thì chỉ hiển thị các môn của ngành công nghê thông tin thôi, không hiển thị môn của các ngành khác như âm nhạc, mỹ thuật chẳng hạn.

* Có thể giúp nhiều đội nhóm hoạt động với nhau bằng nhiều ngôn ngữ khác nhau!

A group of people with text

Description automatically generated

**Tuy nhiên vẫn có những nhược điểm cần lưu ý:**

Việc chia nhỏ service khiến kinh phí tăng cao

Việc triển khai, quản lý phức tạp hơn

Khó khăn để đám bảo sự nhất quán cũng như đồng bộ dữ liệu giữa các services

Khó khăn khi chia tách các service để giảm tối đa ảnh hưởng của các services với nhau

**A diagram of a diagram of money

Description automatically generated**

* Việc giao tiếp giữa các service mất nhiều thời gian hơn

A clock and a pointer

Description automatically generated with medium confidence

**Chia làm 4 service con:**

**+ Service đăng kí học phần : Pipeline.**

Việc chọn pipeline có thể Tăng hiệu suất và tốc độ xử lý cho đăng kí học phần.

Đăng kí học phần cần chú trọng hiệu suất cao => Pipeline.

**A cartoon of a child sitting at a computer

Description automatically generated**

Phân chia trách nhiệm rõ ràng:

Mỗi bước trong pipeline phân chia công việc rõ ràng, dễ dàng quản lý, bảo trì và mở rộng và tự động hóa quá trình kiểm tra

**+ Service quản lý điểm và lịch học: Layer.**

Tăng tính bảo trì và dễ dàng quản lý: Với kiến trúc Layer, các thành phần của hệ thống được tách biệt rõ ràng. Điều này giúp việc bảo trì, sửa lỗi và nâng cấp trở nên dễ dàng hơn vì mỗi tầng có thể được thay đổi mà không ảnh hưởng đến các tầng khác.

Giảm sự phụ thuộc lẫn nhau: Sự tách biệt các tầng giúp giảm sự phụ thuộc lẫn nhau giữa các thành phần của hệ thống, làm cho hệ thống dễ dàng mở rộng và tích hợp với các hệ thống khác.

**+ Service Tra cứu công nợ: Layer.**

Cải thiện tính bảo mật: Kiến trúc Layer cho phép triển khai các biện pháp bảo mật ở từng tầng riêng biệt, đảm bảo rằng dữ liệu nhạy cảm được bảo vệ tốt hơn.

**+ Service tốt nghiệp: pipeline.**

Dễ dàng theo dõi và giám sát: Các bước trong pipeline có thể được theo dõi riêng lẻ, giúp dễ dàng phát hiện và khắc phục sự cố nhanh chóng.

# PHÂN TÍCH & THIẾT KẾ

## Architecture diagram

A diagram of a computer

Description automatically generated

**Thành phần:**

+ API Gateway: là điểm truy cập duy nhất cho các ứng dụng bên ngoài. Nó chịu trách nhiệm định tuyến các yêu cầu đến các services thích hợp, xử lý xác thực và ủy quyền và quản lý lưu lượng truy cập.

+ Config Server: là một kho lưu trữ tập trung cho các cấu hình ứng dụng. Điều này cho phép các services tải xuống cấu hình của chúng từ một vị trí duy nhất, giúp dễ dàng quản lý và triển khai các thay đổi cấu hình.

+ Eureka: là một máy chủ khám phá dịch vụ cho phép các services đăng ký và tìm kiếm nhau. Điều này rất quan trọng để các services có thể giao tiếp với nhau một cách hiệu quả

+ HTTP: Giao tiếp các dịch vụ.

+ Zipkin: theo dõi phân tán, nâng cao khả năng quan sát ứng dụng và cho phép trực quan hóa cũng như khắc phục sự cố về độ trễ.

+ Service đăng kí học phần:

+ Service xem điểm:

+ Service công nợ:

+ Service tốt nghiệp:

## Use case diagram

A diagram of a system

Description automatically generated

## Class diagram

### Dịch vụ đăng ký học phần

A screenshot of a computer

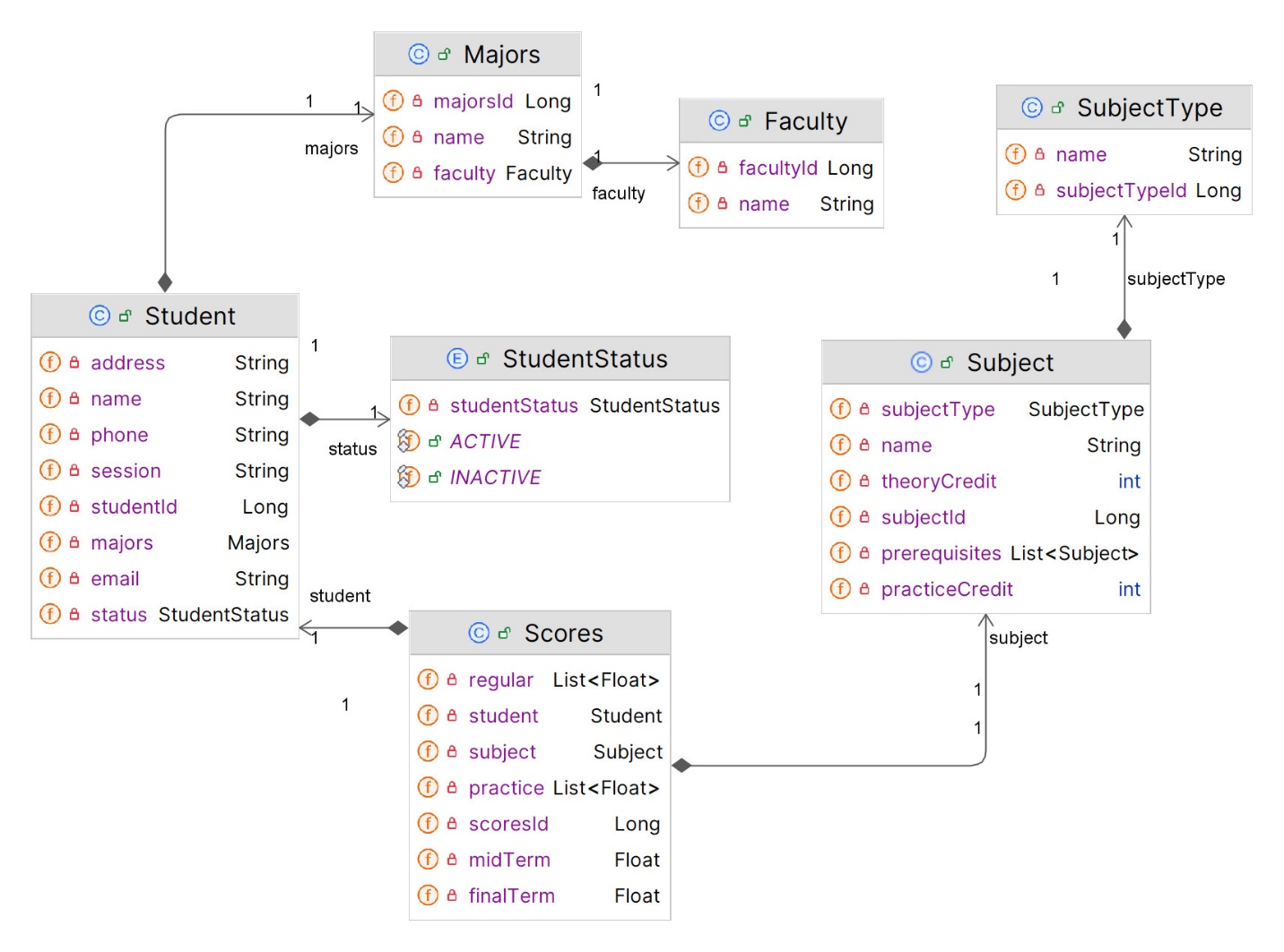
Description automatically generated

### Dịch vụ thông tin sinh viên

A screenshot of a computer diagram

Description automatically generated

### Dịch vụ cựu sinh viên



### Dịch vụ tra cứu công nợ

A diagram of a program

Description automatically generated

## Components diagram

A diagram of a company

Description automatically generated

## Package diagram

A diagram of a diagram

Description automatically generated

# HIỆN THỰC

# KẾT LUẬN

## Kết quả

## Hướng phát triển trong tương lai