**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

**BÁO CÁO ĐỀ TÀI XÂY DỰNG**

**HỆ THỐNG ĐĂNG KÝ HỌC PHẦN**

***Môn học:*** Kiến trúc và thiết kế phần mềm

***Giảng viên*:** Ts.Võ Văn Hải

**Nhóm 1**

***TP.HCM, ngày 20 tháng 05 năm 2024***

|  |  |  |
| --- | --- | --- |
| **STT** | **Họ tên** | **MSSV** |
| 1 | Nguyễn Thanh Sang | 20118191 |
| 2 | Bạch Văn Cường | 20084161 |
| 3 | Trương Phan Đình Trúc | 20033521 |
| 4 |  |  |
| 5 |  |  |

**MỤC LỤC**

[DANH MỤC CÁC HÌNH VẼ 4](#_Toc167098236)

[DANH MỤC CÁC BẢNG BIỂU 5](#_Toc167098237)

[Chương 1: Giới thiệu 6](#_Toc167098238)

[1.1 Tổng quan 6](#_Toc167098239)

[1.1.1 Mô tả bài toán 6](#_Toc167098240)

[1.2 Mục tiêu đề tài 8](#_Toc167098241)

[1.2.1 Các mục tiêu của đề tài đề ra 8](#_Toc167098242)

[1.3. Quyết định kiến trúc xây dựng hệ thống 9](#_Toc167098243)

[1.4. Sevice-Based Architecture (kiến trúc dựa trên dịch vụ). 10](#_Toc167098244)

[Chương 2: Cơ sở lý thuyết 12](#_Toc167098245)

[2.1. Các công nghệ liên quan 12](#_Toc167098246)

[2.1.1 Ngôn ngữ phát triển ứng dụng 12](#_Toc167098247)

[2.1.2 Những framework hỗ trợ phát triển ứng dụng 12](#_Toc167098248)

[Chương 3: Phân tích thiết kế 14](#_Toc167098249)

[3.1. Sơ đồ kiến trúc tổng quát 14](#_Toc167098250)

[3.2. Usecase tổng quát 23](#_Toc167098251)

[3.2. Danh sách tác nhân và mô tả 28](#_Toc167098252)

[3.3. Danh sách các tình huống 30](#_Toc167098253)

[3.4 Tình huống hoạt động 31](#_Toc167098254)

[3.5. Data view 59](#_Toc167098255)

[3.6. Các mẫu thiết kế và cơ chế chung 59](#_Toc167098256)

[3.6.1. API Gateway 59](#_Toc167098257)

[3.6.2. Service Discovery 61](#_Toc167098258)

[3.6.3. Circuit Breaker 63](#_Toc167098259)

[Chương 4: Kết luận 64](#_Toc167098260)

[TÀI LIỆU THAM KHẢO 65](#_Toc167098261)

DANH MỤC CÁC HÌNH VẼ

[Hình 1 : Mô hình kiến trúc Sevice - Based Architecture 11](#_Toc167098275)

[Hình 2 : Mô hình kiến trúc của hệ thống 14](#_Toc167098276)

[Hình 3: Mô hình thiết kế kiến trúc của service 15](#_Toc167098277)

[Hình 4 : Giao tiếp với notification service thông qua kafka 21](#_Toc167098278)

[Hình 5 : Sơ đồ use-case tổng quát 23](#_Toc167098279)

[Hình 6: Sơ đồ usecase Major management 24](#_Toc167098280)

[Hình 7: Sơ đồ usecase Course management 25](#_Toc167098281)

[Hình 8: Sơ đồ usecase Registration management 26](#_Toc167098282)

[Hình 9: Sơ đồ use-case Payment management 27](#_Toc167098283)

[Hình 10: UC00\_Activity 32](#_Toc167098284)

[Hình 11: UC00\_Sequence 33](#_Toc167098285)

[Hình 1*2*: UC01\_Activity 35](#_Toc167098286)

[Hình 13: UC01\_Sequence 36](#_Toc167098287)

[Hình 14: UC02\_Activity 39](#_Toc167098288)

[Hình 15: UC03\_Activity 41](#_Toc167098289)

[Hình 16: UC04\_Activity 43](#_Toc167098290)

[Hình 17: UC05\_Activity 46](#_Toc167098291)

[Hình 18: UC06\_Activity 48](#_Toc167098292)

[Hình 19: ER Diagram 59](#_Toc167098293)

[Hình 20 : Luồng hoạt động của API Gateway 60](#_Toc167098294)

[Hình 21 : Luồng đăng nhập qua API Gateway 61](#_Toc167098295)

[Hình 22 : Luồng hoạt động của service registry 62](#_Toc167098296)

[Hình 23 : Luồng hoạt động của circuit breaker 63](#_Toc167098297)

DANH MỤC CÁC BẢNG BIỂU

[Bảng 1 : Danh sách các tác nhân và mô tả 32](#_Toc167012903)

[Bảng 2: Danh sách các tình huống 32](#_Toc167012904)

[Bảng 3 : UC00\_Đăng nhập 34](#_Toc167012905)

[Bảng 4 : UC01 - Đăng ký học phần 37](#_Toc167012906)

[Bảng 5 : UC02 – Đánh giá học phần 39](#_Toc167012907)

[Bảng 6 : UC03 – Quản lý thông tin 40](#_Toc167012908)

[Bảng 7 : UC04 – Xem lịch học 42](#_Toc167012909)

[Bảng 8 : UC05 – Xem kết quả học tập 43](#_Toc167012910)

[Bảng 9 : UC06 – Xem lịch giảng dạy 45](#_Toc167012911)

[Bảng 10 : UC07 – Nhập điểm 46](#_Toc167012912)

[Bảng 11 : UC08 – Xem tiến độ giảng dạy 48](#_Toc167012913)

[Bảng 12 : UC09 – Quản lý giảng viên 50](#_Toc167012914)

[Bảng 13 : UC10 – Quản lý chương trình học 52](#_Toc167012915)

[Bảng 14 : UC11 – Quản lý học phần 53](#_Toc167012916)

[Bảng 15 : UC012\_Quản lý sinh viên 55](#_Toc167012917)

Chương 1: Giới thiệu

1.1 Tổng quan

1.1.1 Mô tả bài toán

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.

* Với mỗi khóa học, nhà trường sẽ xây dựng khung chương trình đào tạo. Khung chương trình đào tạo được xây dựng dựa vào danh sách các môn học và danh sách các ngành đào tạo(tên ngành đào tạo). Khung chương trình học sẽ bao gồm : ngành đào tạo, khóa đào tạo, danh sách môn học đào tạo, hệ đào tạo, số học kì.
* Các môn học quản lý các thông tin bao gồm : tên môn học, số tín chỉ, số tín chỉ lý thuyết, số tín chỉ thực hành, danh sách các môn học tiên quyết.
* Các nhân viên phòng đào tạo sẽ quản lý thông tin các môn học, chương trình học của mỗi ngành học khác nhau. Các môn học sẽ được sinh viên đăng ký học qua các học phần. Một môn học thì có thể sẽ có nhiều lớp học phần.
* Trước mỗi năm học, nhà trường sẽ lên kế hoạch giảng dạy cho năm học đó, bao gồm thời gian của các học kỳ, ngày bắt đầu, ngày kết thúc.
* Trước mỗi học kỳ, phòng đào tạo sẽ dựa vào khung chương trình chọn một số môn học của từng ngành học cho sinh viên đăng ký học. Các nhân viên phòng đào tạo sẽ phân công giảng viên, phòng học cho các lớp học phần theo chỉ định của khoa.
* Mỗi học phần sẽ được quản lý các thông tin như là : môn học, giảng viên, phòng học, ngày học, tiết học bắt đầu, tiết học kết thúc, sĩ số.
* Sinh viên đăng ký theo học vào một lớp nhất định (lớp học phần) với số sinh viên được đăng ký tối đa được qui định cho từng lớp học phần. Sinh viên sẽ không được đăng ký học phần chứa các môn tiên quyết mà mình chưa học. Nếu là môn học vừa có lý thuyết và vừa có thực hành, sinh viên sẽ được chọn nhóm thực hành.
* Mỗi học kì sinh viên chỉ được phép đăng ký tối đa 30 tín chỉ. Sinh viên không được phép đăng ký vượt quá số tín chỉ quy định. Các lớp học phần mà sinh viên đăng ký thì không được trùng giờ với nhau.
* Các học phần sẽ có học phí khác nhau theo từng hệ đào tạo, chương trình học khác nhau. Học phí sẽ được tính theo tín chỉ. Đơn giá học phí nhân cho số lượng tín chỉ sẽ ra học phí của một học phần mà sinh viên phải đóng.
* Mỗi học phần mà sinh viên đăng ký sẽ được quản lý kết quả học tập.
* Vào ngày mở đăng ký, sinh viên sẽ đăng nhập vào hệ thống bằng tài khoản được cấp của mình 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 các lớp đã đầy, sinh viên sẽ được đưa vào một danh sách dự bị để nhà trường cân nhắc có mở thêm lớp hay không. Nếu không mở thêm lớp, sinh viên sẽ bị hủy đăng ký môn đó.
* Vào các đợt mở đăng ký học phần, số lượng người dùng truy cập cùng lúc là khoảng 20000 người.
* 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.
* 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.
* Nếu một học phần không được thanh toán học phí trong khoảng thời gian yêu cầu thì sinh viên sẽ bị xóa ra khỏi danh sách tham gia lớp học phần.
* Trong trường hợp sinh viên muốn gia hạn học phí thì cần phải làm đơn xin gia hạn học phí và gửi đến nhà trường. Nếu đơn xin gia hạn học phí được duyệt, sinh viên cần phải đóng một khoản học phí (phí giữ chỗ) cho mỗi học phần theo quy định của nhà trường. Số học phí còn lại sinh viên có thể thanh toán bất cứ lúc nào muộn nhất là trước khi kiểm tra giữa kì của môn học đó.
* Nếu đến kì thi giữa kì sinh viên chưa hoàn thành thanh toán học phí theo như quy định của nhà trường thì sẽ không được phép dự thi.
* 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 (cập nhật thông tin cá nhân như tài khoản ngân hàng, số điện thoại, email).
* Với mỗi học phần sau khi sinh viên đã có đầy đủ các cột điểm, hệ thống sẽ hiển thị form cho sinh viên khảo sát về học phần đó. Việc truy cập vào hệ thống sẽ bị chặn lại nếu như sinh viên chưa tham gia khảo sát.
* Các giảng viên cũng có thể truy cập vào hệ thống riêng để kiểm tra các lớp học phần mà mình giảng dạy. Xem thời khóa biểu và nhập điểm cho sinh viên.
* Sau khi số tín chỉ đã đạt theo từng chương trình học, 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.
* Việc nhập các dữ liệu vào hệ thống như môn học, phòng học, thông tin chương trình khung, các ngành học sẽ được hỗ trợ bằng việc import các file theo một mẫu cho trước.

1.2 Mục tiêu đề tài

Mục tiêu của đề tài là xây dựng một hệ thống quản lý đăng ký học phần tích hợp, linh hoạt và hiệu quả cho sinh viên, giáo viên và nhân viên quản lý hệ thống tại trường đại học. Hệ thống này sẽ cho phép sinh viên đăng ký, quản lý và theo dõi lịch học, kết quả học tập. Đồng thời, giáo viên sẽ có khả năng nhập điểm, xem lịch giảng dạy và tiến độ giảng dạy. Nhân viên quản lý sẽ được cung cấp công cụ để quản lý thông tin hệ thống, bao gồm quản lý giảng viên, chương trình học, học phần và sinh viên. Mục tiêu là xây dựng một hệ thống đáp ứng đầy đủ các yêu cầu quản lý học tập, đồng thời tối ưu hóa quy trình và tăng cường tính linh hoạt và hiệu quả trong việc quản lý.

1.2.1 Các mục tiêu của đề tài đề ra

* Đăng ký học phần cho sinh viên:
* Cho phép sinh viên xem danh sách học phần đã đăng ký.
* Cho phép sinh viên đăng ký lớp học phần, hủy đăng ký lớp học phần.
* Cung cấp chi tiết lịch học của lớp học phần sau khi đăng ký.
* Đánh giá học phần:
* Sinh viên có thể phản hồi về khóa học hoặc môn học đã tham gia.
* Quản lý thông tin cá nhân:
* Sinh viên có thể quản lý thông tin cá nhân của mình sau khi đăng nhập vào hệ thống.
* Giáo viên có thể xem lịch giảng dạy và nhập điểm cho sinh viên.
* Xem lịch giảng dạy và tiến độ giảng dạy cho giáo viên:
* Giáo viên có thể xem lịch giảng dạy của mình sau khi đăng nhập vào hệ thống.
* Giáo viên có thể nhập điểm và theo dõi tiến độ giảng dạy của sinh viên.
* Quản lý thông tin hệ thống cho nhân viên quản lý:
* Nhân viên quản lý hệ thống có thể thêm, xóa, sửa thông tin của giảng viên, chương trình học, học phần và sinh viên.
* Cung cấp danh sách chi tiết về giảng viên, chương trình học, học phần và sinh viên để quản lý.

1.3. Quyết định kiến trúc xây dựng hệ thống

* Theo mô tả bài toán, có thể nhận thấy rằng hệ thống quản lý lớp tín chỉ thường sẽ có những đợt đăng ký học phần mà lượng người dùng truy cập tăng đột biến so với những ngày bình thường. Hệ thống cũng sẽ được truy cập rất thường xuyên bởi các người dùng cho nhu cầu kiểm tra thời khóa biểu, lịch học. Do đó ta cần một phong cách kiến trúc xây dựng hệ thống có khả năng đàn hồi tốt (elasticity) để đáp ứng được số lượng truy cập cùng lúc lớn so với những ngày bình thường, và chịu được lỗi (Faul tolerance) đảm bảo cho hệ thống vẫn có thể hoạt động được nếu có một số lỗi xảy ra với máy chủ không làm ảnh hưởng đến các hoạt động giảng dạy khác.
* Dựa vào những tính chất này, Microservices, EDA là những kiến trúc rất phù hợp với yêu cầu bài toán đã đưa ra. Tuy nhiên khi sử dụng Microservices, việc quản lý các dịch vụ có thể trở nên rất phức tạp, yêu cầu kiến thức kỹ thuật cao và kỹ năng quản lý hệ thống phức tạp. Bên cạnh đó, việc chia nhỏ database cho từng service cũng rất khó khăn, mang lại thách thức về mặt đồng bộ dữ liệu và xử lý các transaction trong hệ thống.
* Với yêu cầu bài toán thì việc tách rời database là rất khó và hoàn toàn không khả thi.
* Xét về trình độ, kỹ năng các thành viên trong nhóm đều có ít kinh nghiệm về việc quản lý database. Do đó chúng em quyết định chọn thiết kế hệ thống theo phong cách kiến trúc Sevice-Based Architecture (kiến trúc dựa trên dịch vụ).
* Sevice-Based Architecture tuy không có được khả năng co giản tốt bằng Microservices nhưng có thể triển khai các service sử dụng chung database với nhau, điều này giúp dễ dàng quản lý hơn và không cần chia nhỏ database. Các service trong Sevice-Based Architecture thường lớn hơn và thực hiện nhiều chức năng hơn, điều này làm giảm tính độc lập của các service nhưng lại giúp đơn giản hơn cho việc quản lý các service đó. Sevice-Based Architecture cũng là phong cách kiến trúc có khả năng chịu lỗi tốt phù hợp với yêu cầu bài toán.

1.4. Sevice-Based Architecture (kiến trúc dựa trên dịch vụ).

* Service Based Architecture (SBA), hay Kiến trúc Dựa trên Dịch vụ, là một phong cách kiến trúc phần mềm mà trong đó ứng dụng được xây dựng dưới dạng một tập hợp các dịch vụ. Nó tương tự như kiến trúc hướng dịch vụ (SOA) nhưng thường được áp dụng trong phạm vi hạn chế hơn và ít phức tạp hơn.
* Trong SBA, hệ thống được chia thành các phần chức năng độc lập, mỗi phần là một dịch vụ (service) riêng biệt:

+ Các dịch vụ này thường được xây dựng xung quanh các chức năng kinh doanh cụ thể.

+ Chúng có thể giao tiếp với nhau thông qua APIs hoặc messaging protocols.

+ Mỗi dịch vụ là tự chủ và có thể được phát triển, triển khai, và quản lý một cách độc lập.

**A diagram of a user interface

Description automatically generated**

Hình 1 : Mô hình kiến trúc Sevice - Based Architecture

* **Ưu điểm của SBA :**

+ Các dịch vụ là độc lập, điều này tạo điều kiện cho việc phát triển và bảo trì module.

+ Hệ thống có thể dễ dàng mở rộng bằng cách thêm mới hoặc cải thiện các dịch vụ mà không ảnh hưởng đến toàn bộ hệ thống.

+ Các dịch vụ có thể được tái sử dụng trong các ứng dụng khác nhau, hỗ trợ khả năng tái sử dụng components.

+ Các dịch vụ có thể được hoàn thiện sử dụng những công nghệ khác nhau và liên kết thông qua các giao thức chung.

* **Nhược điểm của SBA :**

+Khi số lượng dịch vụ tăng lên, việc quản lý sẽ trở nên phức tạp hơn, lúc này cần nghĩ đến việc chia nhỏ database ra. Thông thường sẽ là 3-12 service cho 1 database

+Có thể bị ảnh hưởng do giao tiếp giữa các dịch vụ qua mạng có thể gây ra độ trễ.

+ Việc quản lý transaction qua nhiều dịch vụ có thể gặp phải thách thức, đặc biệt là liên quan đến tính nhất quán và độ tin cậy.

Chương 2: Cơ sở lý thuyết

2.1. Các công nghệ liên quan

2.1.1 Ngôn ngữ phát triển ứng dụng

**Ngôn ngữ lập trình java**

Java là một ngôn ngữ lập trình và nền tảng tính toán được Sun Microsystems phát hành lần đầu tiên vào năm 1995. Nó đã phát triển từ khởi đầu khiêm tốn để trở thành nền tảng đáng tin cậy cho phần lớn thế giới kỹ thuật số ngày nay, cung cấp nền tảng vững chắc cho nhiều dịch vụ và ứng dụng. Các sản phẩm và dịch vụ kỹ thuật số mới, sáng tạo được thiết kế cho tương lai cũng tiếp tục dựa vào Java.

Trong khi hầu hết các ứng dụng Java hiện đại kết hợp Java runtime và ứng dụng cùng nhau, vẫn có nhiều ứng dụng và thậm chí một số trang web sẽ không hoạt động trừ khi bạn đã cài đặt Java trên máy tính để bàn. Java.com, trang web này, được dành cho người dùng có thể vẫn cần Java cho các ứng dụng máy tính để bàn của họ – cụ thể là các ứng dụng nhắm đến Java 8. Các nhà phát triển cũng như người dùng muốn học lập trình Java nên truy cập vào trang web dev.java và người dùng doanh nghiệp nên truy cập vào oracle.com/java để biết thêm thông tin.

2.1.2 Những framework hỗ trợ phát triển ứng dụng

**React**

React là một thư viện JavaScript mã nguồn mở, được phát triển bởi Facebook, dùng để xây dựng giao diện người dùng (UI) cho các ứng dụng web. Được phát hành lần đầu vào năm 2013, React đã trở thành một trong những công cụ phổ biến nhất trong lĩnh vực phát triển web. Dưới đây là một số điểm nổi bật về React:

**Component-based Architecture (Kiến trúc dựa trên thành phần):**

React xây dựng UI dựa trên các thành phần độc lập, có thể tái sử dụng, giúp dễ dàng quản lý và phát triển.

Mỗi thành phần có thể có trạng thái riêng (state) và các thuộc tính (props) để quản lý dữ liệu và hành vi.

**Virtual DOM (DOM ảo):**

React sử dụng DOM ảo để tối ưu hóa việc cập nhật UI. Khi có sự thay đổi trong UI, React sẽ cập nhật DOM ảo trước, sau đó so sánh với DOM thật và chỉ thực hiện những thay đổi cần thiết. Cách tiếp cận này giúp cải thiện hiệu suất của ứng dụng.

**Declarative Syntax (Cú pháp khai báo):**

React cho phép bạn mô tả UI dưới dạng kết quả mong muốn và React sẽ chịu trách nhiệm duy trì trạng thái UI phù hợp với mô tả đó. Cú pháp này giúp code dễ đọc, dễ hiểu và dễ bảo trì.

**One-way Data Binding (Ràng buộc dữ liệu một chiều):**

Dữ liệu trong React chỉ đi theo một hướng, từ cha đến con qua các props. Điều này giúp dễ dàng theo dõi và kiểm soát luồng dữ liệu.

**Ecosystem (Hệ sinh thái):**

React có một hệ sinh thái phong phú với nhiều thư viện và công cụ hỗ trợ, chẳng hạn như Redux, React Router, và nhiều plugin khác.

Chương 3: Phân tích thiết kế

3.1. Sơ đồ kiến trúc tổng quát

**A diagram of a service

Description automatically generated**

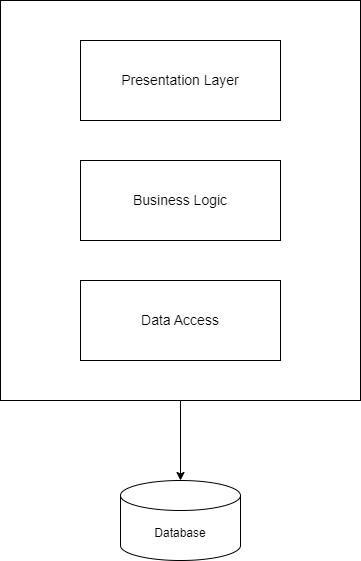
Hình 2 : Mô hình kiến trúc của hệ thống

Hệ thống sẽ được xây dựng trên phong cách kiến trúc dựa trên dịch vụ (Service-Based Architecture), bao gồm nhiều dịch vụ riêng biệt giao tiếp với nhau thông qua API. Mỗi dịch vụ sẽ chịu trách nhiệm cho một khía cạnh chức năng của hệ thống.

**Các thành phần của hệ thống bao gồm :**

* User interface: Client hoặc là các component sử dụng các dịch vụ này.
* API Layer : Được thiết kế để quản lý định tuyến các request từ client đến các dịch vụ khác. Nó cũng đảm nhận vai trò xác thực các request.
* Service: Các đơn vị khép kín cung cấp các quy trình công việc và chức năng cụ thể và thường sử dụng chung database với nhau.
* Service registry : Là nơi để các service đăng ký và được quản lý. Service registry giúp các dịch vụ có thể tìm và kết nối với nhau.
* Database : Là nơi lưu trữ toàn bộ dữ liệu của hệ thống.

**Mô hình kiến trúc của từng service:**



Hình 3: Mô hình thiết kế kiến trúc của service

* Presentation Layer: Nơi chứa các API để lấy dữ liệu của tầng Bussiness xử lý xong dữ liệu
* Business Logic: chịu trách nhiệm xử lý các quy tắc nghiệp vụ của hệ thống, bao gồm xử lý logic, mô hinh dữ liệu và các thuật toán liên quan
* Data Access: Tương tác với cơ sở dữ liệu, thực hiện các thao tác lưu trữ, truy xuất và quản lý dữ liệu
* Database: Nơi lưu trữ dữ liệu hệ thống

**Nhiệm vụ của từng dịch vụ:**

* Major management service:
  + Chịu trách nhiệm quản lý thông tin về các ngành học, chương trình đào tạo của từng ngành:

Gồm các chức năng thêm, xóa, sửa các thông tin về ngành học và chương trình đào tạo

* + Cung cấp thông tin về ngành học, chương trình đào tạo cho các service khác.  
    Cho phép lấy dữ liệu ngành học và chương trình đào tạo thông qua API Với các chức năng cơ bản như vậy thì Major management service nên sử dụng kiến trúc thiết kế Layer để có thể dễ dàng bảo trì, mở rộng, dễ triển khai và khả năng bảo mật cao
* Course management service:
  + Quản lý thông tin các môn học bao gồm tên môn học, mô tả môn học.  
    Gồm các chức năng thêm, xóa, sửa các thông tin về môn học
  + Quản lý điều kiện tiên quyết của các môn học.

Gồm các chức năng thêm, xóa, sửa các điều kiện tiên quyết. Có thể có môn không có điều kiện tiên quyết. Môn có điều kiện tiên quyết là những môn không có điều kiện tiên quyết

Với các chức năng cơ bản như vậy thì Course management service nên sử dụng kiến trúc thiết kế Layer để có thể dễ dàng bảo trì, mở rộng, dễ triển khai và khả năng bảo mật cao

* Registration management service:
  + Chịu trách nhiệm quản lý quy trình đăng ký học phần của sinh viên. Cho phép sinh viên chọn lớp học phần, kiểm tra lịch học, lịch giảng dạy của giảng viên.  
    Quy trình đăng ký học phần cần có:

-Nhân viên chọn mở đăng ký học phần, chọn khoa được đăng ký học phần trong thời gian thiết lập. Khi đến thời gian đăng ký học phần, phần mềm sẽ mở để sinh viên vào đăng ký học phần. Nếu sinh viên không trong thời gian đăng ký học phần hoặc không phải khoa được mở đăng ký thì sẽ không hiển thị các môn học phần để đăng ký và thông báo chưa mở đăng ký học phần

- Sinh viên chọn học kỳ cần đăng ký học phần sẽ hiển thị các môn học phần được phép đăng ký

- Sinh viên chọn môn học phần muốn đăng ký. Nếu sinh viên chưa hoàn thành điều kiện tiên quyết của môn học phần đó sẽ không thể đăng ký được và hiện thông báo môn học cần hoàn thành trước

- Sinh viên sẽ xem và chọn được lịch học của môn học đó. Nếu môn học có thêm tiết thực hành thì sinh viên cần chọn thêm lịch học thực hành. Nếu lớp học đó đã đầy sinh viên đăng ký thì không thể đăng ký được

- Sau khi đăng ký xong môn học sẽ xuất hiện lịch học của môn học đó và số tiền cần thanh toán để mở lớp. Nếu như sinh viên đăng ký môn học trùng lịch với môn học đã đăng ký trước đó thì sẽ không thể đăng ký được và hiện thông báo các môn bị trùng lịch

- Sau khi hết thời gian đăng ký học phần hệ thống sẽ gửi gmail thông báo để sinh viên thanh toán tiền học phí. Nếu đủ số lượng sinh viên đã đóng tiền học phí trước thời hạn hệ thống sẽ chấp nhận mở lớp. Nếu sinh viên nào chưa đóng tiền học phí sẽ tự động xóa khỏi lớp học.

* + Cung cấp lịch học chi tiết, phòng học, và giảng viên giảng dạy để sinh viên tham khảo.   
    Hệ thống lấy thông tin lịch học, phòng học và giảng viên thông qua API để hiển thị lịch học
  + Quản lý danh sách chờ và ưu tiên đăng ký dựa trên quy định của trường.  
      
    Hệ thống đăng ký học phần cần đáp ứng số lượng lớn sinh viên truy cập trong thời gian nhất định vì thế nên cần kiến trúc có độ co dãn tốt. Nếu không có thể gây quá tải hệ thống, làm chậm các hoạt động khác. Dựa vào đó kiến trúc Space-Based là lựa chọn tốt nhất có thể đáp ứng được số lượng lớn lượng truy cập trong khoảng thời gian nhất định. Nhưng việc xây dựng kiến trúc Space-Based rất phức tạp, cần có nhiều kiến thức và kinh nghiệm mới xây dựng được. Vì thế có thể đổi sang sử dụng dịch vụ khác là Cloud của AWS. Cloud có thể đáp ứng được lượng người dùng truy cập lớn mà vẫn giữ được tốc độ xử lý mà không bị quá tải. Chi phí cũng tính theo lượng sử dụng dịch vụ. Khi kết thúc đăng ký học phần không còn nhiều người truy cập thì chi phí cũng giảm xuống.

Để giảm lượng truy cập cùng một lúc có thể đăng ký sẵn các môn học phần bắt buộc cho sinh viên, sinh viên chỉ đăng ký các môn tự chọn giảm thời gian duy trì kết nối. Hoặc có thể chia các đợt đăng ký học phần ra nhiều đợt theo từng khoa để giảm số lượng người dùng truy cập cùng một lúc giúp tránh tình trạng quá tải hệ thống.

* Payment management service :
  + Chịu trách nhiệm xử lý thanh toán các khoản thu như học phí, phí xét tốt nghiệp.
  + Hỗ trợ thanh toán trực tiếp qua các ngân hàng, chuyển khoản.
  + Quản lý biên lai thanh toán và cung cấp báo cáo tài chính cho sinh viên và phòng tài chính.
  + **Dịch vụ quản lý giao dịch:** Dịch vụ này chịu trách nhiệm xử lý các giao dịch thanh toán, bao gồm nhận yêu cầu thanh toán, xác minh thông tin thanh toán, thực hiện thanh toán và tạo biên lai thanh toán.

**Dịch vụ quản lý tài khoản:** Dịch vụ này chịu trách nhiệm quản lý tài khoản của sinh viên, bao gồm cập nhật thông tin tài khoản, theo dõi số dư tài khoản và tạo báo cáo tài chính.

**Dịch vụ quản lý phương thức thanh toán:** Dịch vụ này chịu trách nhiệm quản lý các phương thức thanh toán được hỗ trợ, bao gồm tích hợp với các cổng thanh toán và ngân hàng.

Do trình độ và kỹ năng của các thành viên trong nhóm đều có ít kinh nghiệm về quản lý giao dịch và thanh toán. Nên Payment management service sẽ sử dụng thiết kế Layer để có thể dễ dàng xây dựng. Đồng thời kiến trúc Layer cũng dễ dàng mở rộng hoặc thay đổi sang kiến trúc thiết kế khác. Nếu như trong quá trình xây dựng Payment management cảm thấy sử dụng kiến trúc khác phù hợp hơn thì cũng dễ dàng chuyển đổi từ kiến trúc Layer sang kiến trúc khác.

* Academic result management service :
  + Quản lý kết quả học tập của sinh viên, lưu trữ bảng điểm, tiến độ học tập và các dữ liệu khác:

Dịch vụ này chịu trách nhiệm lưu trữ và quản lý kết quả học tập của sinh viên trong hệ thống. Bao gồm việc lưu trữ bảng điểm chi tiết của từng sinh viên, thông tin về tiến độ học tập và các dữ liệu khác liên quan đến kết quả học tập.

* + Gửi khảo sát đến cho sinh viên thực hiện sau khi đã hoàn thành học phần:

Dịch vụ này tự động gửi các khảo sát đánh giá đến cho sinh viên sau khi họ hoàn thành một học phần. Các khảo sát này được thiết kế để thu thập ý kiến và phản hồi từ sinh viên về chất lượng của học phần, cũng như để cải thiện quy trình giảng dạy và học tập.

* + Cung cấp báo cáo phân tích kết quả học tập cho sinh viên và giảng viên:

Dịch vụ này tạo ra các báo cáo phân tích về kết quả học tập cho cả sinh viên và giảng viên. Báo cáo này cung cấp cái nhìn tổng quan về hiệu suất học tập của sinh viên trong một khoảng thời gian nhất định, bao gồm điểm số, tiến độ, và các chỉ số khác. Báo cáo cũng có thể phân tích và so sánh hiệu suất học tập giữa các lớp học, ngành học hoặc giảng viên khác nhau để giúp trường đánh giá và cải thiện chất lượng giáo dục.

- **Chức năng**:

- Lưu trữ và quản lý bảng điểm của sinh viên.

- Theo dõi tiến độ học tập của sinh viên qua các kỳ học.

- Gửi khảo sát cho sinh viên sau khi hoàn thành học phần.

* Student management service :
  + Quản lý thông tin về sinh viên, thông tin liên hệ, hồ sơ học tập của sinh viên:

Dịch vụ này chịu trách nhiệm lưu trữ và quản lý thông tin cá nhân của sinh viên trong hệ thống. Bao gồm các thông tin như họ tên, ngày sinh, địa chỉ, email, số điện thoại, và các thông tin khác liên quan đến sinh viên.

* + Theo dõi và cập nhật status học tập của sinh viên (ví dụ: đang học, đã tốt nghiệp, nghỉ học):

Dịch vụ này theo dõi và cập nhật trạng thái học tập của sinh viên trong quá trình học. Trạng thái học tập có thể bao gồm các thông tin như "đang học", "đã tốt nghiệp", "nghỉ học", hoặc các trạng thái khác tùy thuộc vào quy định của trường. Cập nhật trạng thái này giúp quản lý sinh viên và hỗ trợ trong việc xử lý các yêu cầu học tập của sinh viên.

**- Chức năng:**

- Lưu trữ thông tin cá nhân và liên hệ của sinh viên.

- Quản lý hồ sơ học tập của sinh viên.

- Theo dõi và cập nhật status học tập của sinh viên (ví dụ: đang học, đã tốt nghiệp, nghỉ học).

* Notification service :
  + Chịu trách nhiệm gửi thông báo cho sinh viên:

Dịch vụ này chịu trách nhiệm gửi thông báo đến sinh viên thông qua các kênh như email, tin nhắn, hoặc ứng dụng di động. Thông báo có thể liên quan đến các sự kiện quan trọng trong quá trình học tập như thay đổi lịch học, thông báo kết quả học tập, hoặc các thông báo khác từ trường.

* + Phát triển hệ thống cảnh báo học tập, ví dụ: nhắc nhở về deadline nộp bài, đóng học phí:

Dịch vụ này phát triển và quản lý hệ thống cảnh báo học tập để nhắc nhở sinh viên về các trạng thái quan trọng và các deadline trong quá trình học. Cảnh báo có thể bao gồm nhắc nhở về deadline nộp bài, hạn chót đóng học phí, hoặc các thông báo khác liên quan đến học tập.

* + Gửi email sau khi sinh viên đăng ký học phần.

Dịch vụ này tự động gửi email thông báo cho sinh viên sau khi họ đăng ký học phần thành công. Email có thể chứa các thông tin như lịch học, phòng học, giảng viên, và các thông tin liên quan đến môn học mà sinh viên đã đăng ký.

**- Giao tiếp với notification service:**



Hình 4 : Giao tiếp với notification service thông qua kafka

* + Producer gửi thông điệp (event) đến một topic trong Kafka Cluster.
  + Kafka Cluster lưu trữ thông điệp này trong các broker.
  + Notification Service, hoạt động như một Consumer, lắng nghe các thông điệp từ topic trong Kafka Cluster và xử lý chúng.

**- Chức năng:**

- Gửi thông báo qua email hoặc tin nhắn về deadline nộp bài, đóng học phí, và các sự kiện quan trọng khác.

- Phát triển hệ thống cảnh báo học tập để nhắc nhở sinh viên về các trạng thái quan trọng và deadline.

* Teacher management service:
  + Chịu trách nhiệm quản lý các thông tin về giảng viên, bao gồm tên, thông tin liên lạc, trình độ,…:

Dịch vụ này chịu trách nhiệm lưu trữ và quản lý các thông tin về giảng viên bao gồm tên, thông tin liên lạc, trình độ chuyên môn, kinh nghiệm giảng dạy, và các thông tin khác liên quan. Thông tin này được cập nhật và duy trì để đảm bảo tính chính xác và đầy đủ.

* + Cung cấp nền tảng để giảng viên có thể nhập điểm, đánh giá sinh viên và lập báo cáo giảng dạy.:

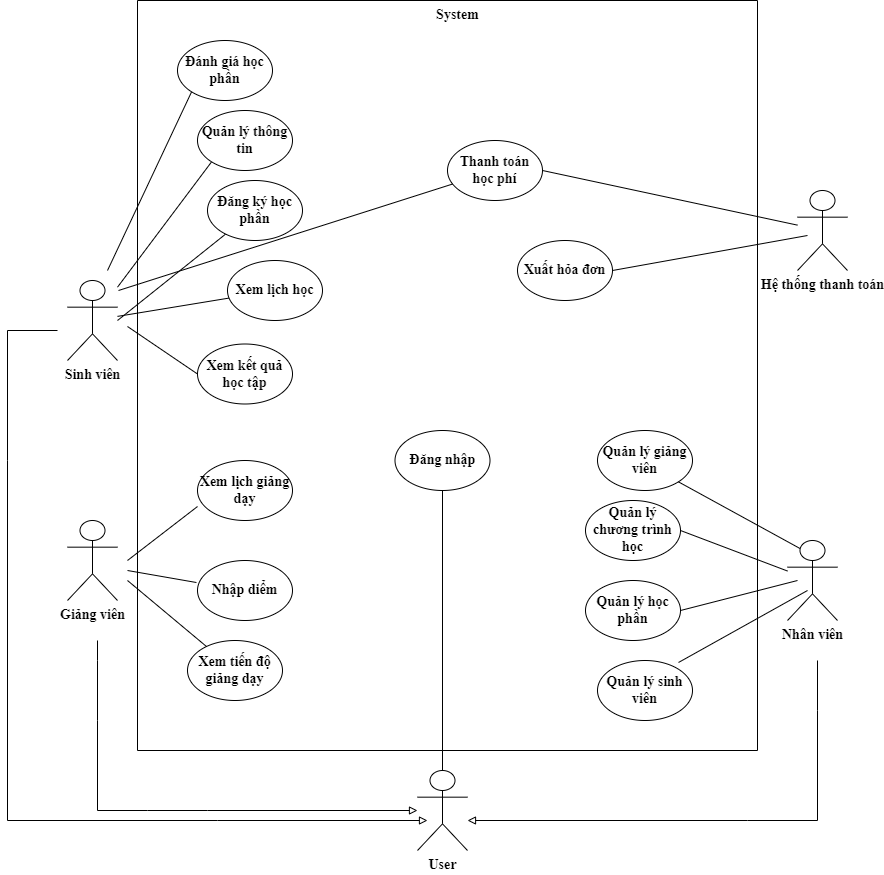
Dịch vụ này cung cấp một nền tảng trực tuyến cho giảng viên để nhập điểm, đánh giá sinh viên và lập báo cáo về quá trình giảng dạy của mình. Nền tảng này có thể bao gồm các tính năng như nhập điểm, xem danh sách sinh viên, đánh giá bài tập và bài kiểm tra, ghi chú về tiến độ học tập của sinh viên, và tạo báo cáo tổng kết về hoạt động giảng dạy. Đồng thời, nền tảng cũng phải đảm bảo tính bảo mật và quản lý quyền truy cập cho từng giảng viên.

**- Chức năng:**

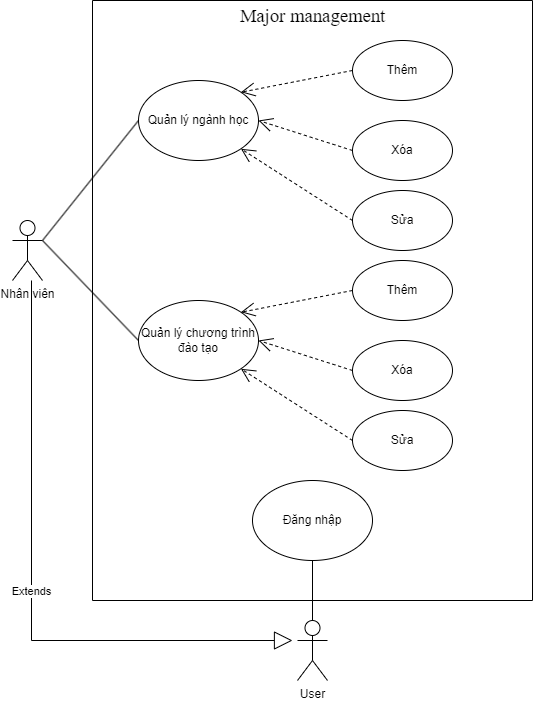
- Lưu trữ thông tin về giảng viên bao gồm tên, thông tin liên lạc, trình độ.

- Cung cấp giao diện để giảng viên nhập điểm, đánh giá sinh viên và tạo báo cáo giảng dạy.

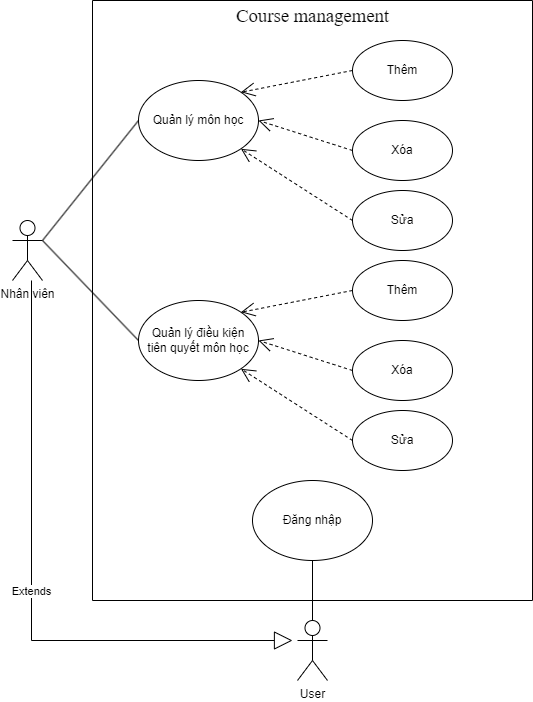
3.2. Usecase tổng quát

****

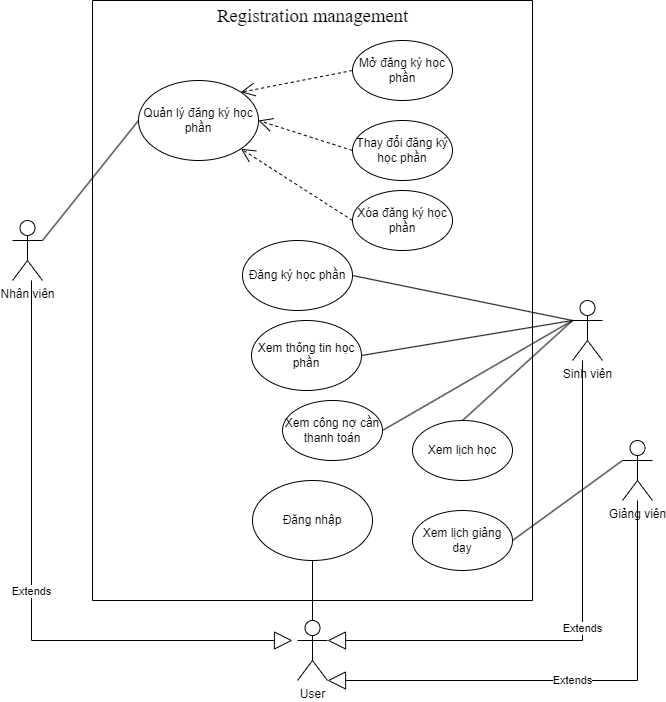
Hình 5 : Sơ đồ use-case tổng quát

****

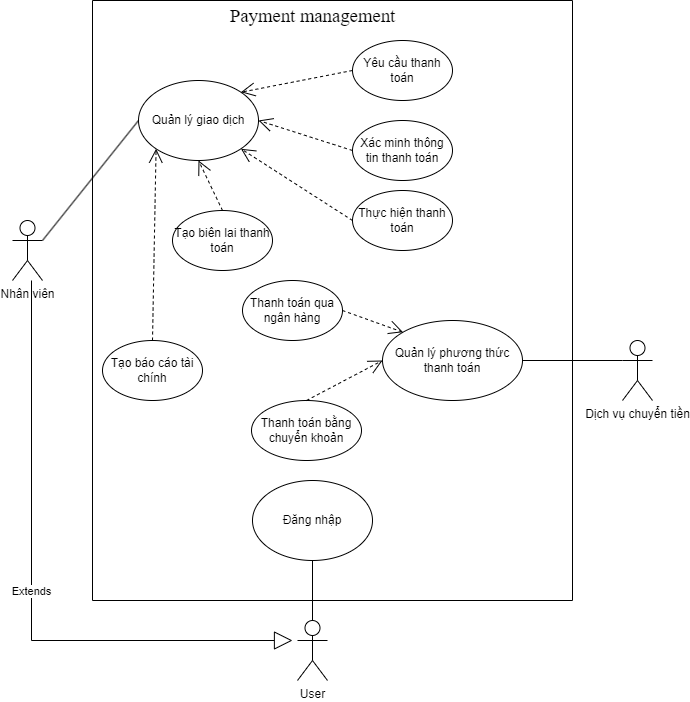
Hình 6: Sơ đồ usecase Major management

****

Hình 7: Sơ đồ usecase Course management

****

Hình 8: Sơ đồ usecase Registration management

****

Hình 9: Sơ đồ use-case Payment management

3.2. Danh sách tác nhân và mô tả

|  |  |  |  |
| --- | --- | --- | --- |
| **Tác nhân** | **Mô tả tác nhân** | **Tên chức năng** | **Mô tả chức năng** |
| **Sinh viên** | Sinh viên học tập tại trường | Đăng kí học phần | Gồm các chức năng như xem danh sách học phần đã đăng ký, đăng ký lớp học phần, hủy đăng ký lớp học phần, xem chi tiết lịch học lớp học phần |
| Đánh giá học phần | Cho phép sinh viên phản hồi về khóa học hoặc môn học sinh viên đã tham gia |
| Quản lý thông tin | Quản lí thông tin cá nhân của sinh viên đã đăng nhập vào ứng dụng |
| Xem lịch học | Cho phép sinh viên xem thông tin lịch học sau khi đăng ký thành công lớp học phần |
| Xem kết quả học tập | Cho phép sinh viên xem kết quả học tập sau khi đăng nhập vào ứng dụng |
| **Giáo viên** | Giáo Viên hướng dẫn , giảng dạy , quản lý lớp học phần của Sinh viên đăng ký | Xem lịch giảng dạy | Cho phép giáo viên xem lịch giảng dạy khi giáo viên đăng nhập vào ứng dụng |
| Nhập điểm | Giáo viên có thể nhập điểm số cho các sinh viên thuộc lớp giảng dạy của giáo viên |
| Xem tiến độ giảng dạy | Cho phép giáo viên theo dõi và đánh giá sự phát triển và hoàn thành các mục tiêu trong học tập của sinh viên |
| **Nhân viên** | Nhân Viên quản lý hệ thống | Quản lý giảng viên | Cho phép thêm , xóa, sửa thông tin của giảng viên, xem danh sách giảng viên |
| Quản lý chương trình học | Cho phép thêm, xóa, sửa thông tin chương trình học,xem danh sách chương trình học |
| Quản lý học phần | Cho phép thêm, xóa, sửa thông tin của học phần, xem danh sách học phần |
| Quản lý sinh viên | Cho phép thêm , xóa, sửa thông tin của sinh viên, xem danh sách sinh viên |

Bảng 1 : Danh sách các tác nhân và mô tả

3.3. Danh sách các tình huống

|  |  |
| --- | --- |
| **ID use case** | **Tên Use case** |
| **UC01** | UC01: Đăng ký học phần |
| **UC02** | UC02: Xem lịch học |
| **UC03** | UC03: Xem kết quả học tập |
| **UC04** | UC04: Đánh giá học phần |
| **UC05** | UC05: Quản lý thông tin cá nhân |
| **UC06** | UC06: Xem lịch giảng dạy |
| **UC07** | UC07: Nhập điểm |
| **UC08** | UC08: Xem tiến độ giảng dạy |
| **UC09** | UC09: Quản lý giảng viên |
| **UC10** | UC10: Quản lý chương trình học |
| **UC11** | UC11: Quản lý học phần |
| **UC12** | UC12: Quản lý sinh viên |

Bảng 2: Danh sách các tình huống

3.4 Tình huống hoạt động

|  |  |
| --- | --- |
| **− Tên use case:** UC00\_Đăng nhập | |
| **− Mô tả sơ lược:** Sinh viên , giáo viên và nhân viên quản lý hệ thống được cung cấp tài khoản của trường | |
| **− Actor chính:** Sinh viên , giáo viên và nhân viên quản lý hệ thống | |
| **− Actor phụ:** | |
| **− Tiền điều kiện (Pre - condition):** Người dùng đã có tài khoản do trường cấp | |
| **− Hậu điều kiện (Post-condition):** | |
| **− Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Người dùng chọn chức năng đăng nhập | 2. Hệ thống hiển thị giao diện nhập thông tin tài khoản |
| 3. Người dùng nhập tài khoản gồm email và mật khẩu |  |
| 4. Chọn nút đăng nhập | 5. Hệ thống kiểm tra thông tin nhập vào |
|  | 6. Hệ thống chuyển hướng người dùng vào giao diện trang chủ. |
| **− Luồng sự kiện thay thế (alternate flow):** | |
|  | 6.1. Hệ thống thông báo thông tin tài khoản đăng nhập không đúng |
|  | 6.2. Hệ thống quay lại bước 3 |

Bảng 3 : UC00\_Đăng nhập

**Activity diagram:**

A diagram of a diagram

Description automatically generated

Hình 10: UC00\_Activity

**Sequence diagram:**

**A diagram of a diagram

Description automatically generated**

Hình 11: UC00\_Sequence

|  |  |
| --- | --- |
| **− Tên use case:** UC01\_Đăng ký học phần | |
| **− Mô tả sơ lược:** Sinh viên thực hiện việc đăng ký các học phần cho kỳ học mới thông qua hệ thống quản lý đào tạo của trường đại học. | |
| **− Actor chính:**Sinh Viên | |
| **− Actor phụ:** Hệ thống quản lý đào tạo | |
| **− Tiền điều kiện (Pre-condition):** Sinh viên đã đăng nhập vào hệ thống quản lý đào tạo và thông tin học phần cho kỳ học mới đã được cập nhật trên hệ thống. | |
| **− Hậu điều kiện (Post-condition):** Học phần được đăng ký thành công sẽ được lưu lại trong hệ thống và hiển thị trong lịch học của sinh viên. Sinh viên nhận được xác nhận về các học phần đã đăng ký. | |
| **− Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Đăng nhập vào hệ thống |  |
|  | 1. xác thực thông tin đăng nhập của sinh viên |
| 1. chọn chức năng đăng ky học phần |  |
|  | 1. Hiển thị danh sách các học phần có sẵn chi kỳ học mới |
| 1. Chọn các học phần muốn đăng ký và thêm vào danh sách đăng ký |  |
|  | 1. Kiểm tra điều kiện đăng ký( như số tín chỉ tối đa, học phần tiên quyết, số lượng chỗ còn lại trong lớp học phần,..) |
|  | 1. nếu các điều kiện đăng ký hợp lệ, lưu thông tin đăng ký của sinh viên |
|  | 1. Gửi thông báo xác nhận đăng ký thành công cho sinh viên |
| 1. Kiểm tra lại thông tin các học phần đã đăng ký và hoàn tất quy trình |  |
| **− Luồng sự kiện thay thế (alternate flow):** | |
|  | 6.1 Nếu điều kiện đăng ký không hợp lệ (vượt quá số tín chỉ cho phép, chưa học học phần tiên quyết, hết chỗ, v.v.), hệ thống sẽ thông báo lỗi và yêu cầu điều chỉnh lại danh sách học phần đăng ký. |

Bảng 4 : UC01 - Đăng ký học phần

**Activity diagram:**

A diagram with blue text

Description automatically generated

Hình 1*2*: UC01\_Activity

**Sequence diagram:** **A screenshot of a computer

Description automatically generated**

Hình 13: UC01\_Sequence

|  |  |
| --- | --- |
| **− Tên use case:** UC02\_Đánh giá học phần | |
| **− Mô tả sơ lược:** Sinh viên thực hiện việc đăng ký các học phần cho kỳ học mới thông qua hệ thống quản lý đào tạo của trường đại học. | |
| **− Actor chính:**Sinh Viên | |
| **− Actor phụ:** Hệ thống quản lý đào tạo | |
| **− Tiền điều kiện (Pre-condition):** Sinh viên đã hoàn thành các học phần trong kỳ học và đã đăng nhập vào hệ thống quản lý đào tạo. | |
| **− Hậu điều kiện (Post-condition):** Đánh giá của sinh viên được lưu lại trong hệ thống và có thể được sử dụng để cải thiện chất lượng giảng dạy của các học phần. | |
| **− Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Đăng nhập vào hệ thống |  |
|  | 1. Xác thực thông tin đăng nhập của sinh viên |
| 1. Chọn chức năng "Đánh giá học phần". |  |
|  | 1. Hiển thị danh sách các học phần mà sinh viên đã tham gia trong kỳ học. |
| 1. Chọn học phần muốn đánh giá. |  |
| 1. Điền vào bảng đánh giá với các tiêu chí cụ thể (giảng viên, nội dung học phần, phương pháp giảng dạy, cơ sở vật chất, v.v.). |  |
|  | 1. Gửi đánh giá |
|  | 1. Lưu đánh giá của sinh viên vào hệ thống |
| 1. Nhận thông báo xác nhận đánh giá đã được lưu thành công |  |
| **− Luồng sự kiện thay thế (alternate flow):** | |
|  | 7.1 Nếu có lỗi xảy ra trong quá trình gửi đánh giá (như mất kết nối, lỗi hệ thống), hệ thống sẽ thông báo lỗi và yêu cầu sinh viên thử lại sau. |

Bảng 5 : UC02 – Đánh giá học phần

Activity Diagram :

A diagram with blue rectangles and black dots

Description automatically generated

Hình 14: UC02\_Activity

|  |  |
| --- | --- |
| **− Tên use case:** UC03\_Quản lý thông tin | |
| **− Mô tả sơ lược:** Sinh viên cập nhật và quản lý thông tin cá nhân của mình thông qua hệ thống quản lý đào tạo của trường đại học. | |
| **− Actor chính:**Sinh Viên | |
| **− Actor phụ:** Hệ thống quản lý đào tạo | |
| **− Tiền điều kiện (Pre-condition):** Sinh viên đã đăng nhập vào hệ thống quản lý đào tạo. | |
| **− Hậu điều kiện (Post-condition):**Thông tin cá nhân của sinh viên được cập nhật và lưu lại trong hệ thống. | |
| **− Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Đăng nhập vào hệ thống |  |
|  | 1. Xác thực thông tin đăng nhập của sinh viên |
| 1. Chọn chức năng "Quản lý thông tin cá nhân". |  |
|  | 1. Hiển thị thông tin các nhân hiện tại của sinh viên |
| 1. Cập nhật thông tin cá nhân( như địa chỉ , số điện thoại, email) |  |
|  | 1. Xác nhận và lưu thông tin đăng nhập |
|  | 1. Gửi thông báo xác nhận thông tin đã được cập nhật thành công |
| **− Luồng sự kiện thay thế (alternate flow):** | |
|  | 6.1 Nếu có lỗi xảy ra trong quá trình lưu thông tin (như mất kết nối, lỗi hệ thống), hệ thống sẽ thông báo lỗi và yêu cầu sinh viên thử lại sau. |

Bảng 6 : UC03 – Quản lý thông tin

Activity Diagram :

A screenshot of a diagram

Description automatically generated

Hình 15: UC03\_Activity

|  |  |
| --- | --- |
| **− Tên use case:** UC04\_Xem Lịch học | |
| **− Mô tả sơ lược:** Sinh viên xem lịch học của mình cho kỳ học hiện tại thông qua hệ thống quản lý đào tạo của trường đại học. | |
| **− Actor chính:**Sinh Viên | |
| **− Actor phụ:** Hệ thống quản lý đào tạo | |
| **− Tiền điều kiện (Pre-condition):** Sinh viên đã đăng nhập vào hệ thống quản lý đào tạo và đã đăng ký các học phần cho kỳ học hiện tại. | |
| **− Hậu điều kiện (Post-condition):**Lịch học của sinh viên được hiển thị trên màn hình. | |
| **− Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Đăng nhập vào hệ thống |  |
|  | 1. Xác thực thông tin đăng nhập của sinh viên |
| 1. Chọn chức năng "Xem lịch học". |  |
|  | 1. Hiển thị lịch học của sinh viên cho kỳ học hiện tại |
| 1. Kiểm tra lịch học và các thông tin liên quan(thời gian, phòng học, giảng viên,...) |  |
| **− Luồng sự kiện thay thế (alternate flow):** | |
|  |  |

Bảng 7 : UC04 – Xem lịch học

A screenshot of a computer screen

Description automatically generated

Hình 16: UC04\_Activity

|  |  |
| --- | --- |
| **− Tên use case:** UC05\_Xem kết quả học tập | |
| **− Mô tả sơ lược:** Sinh viên xem kết quả học tập của mình cho các kỳ học đã qua thông qua hệ thống quản lý đào tạo của trường đại học. | |
| **− Actor chính:**Sinh Viên | |
| **− Actor phụ:** Hệ thống quản lý đào tạo | |
| **− Tiền điều kiện (Pre-condition):** Sinh viên đã đăng nhập vào hệ thống quản lý đào tạo và đã hoàn thành các học phần có kết quả học tập. | |
| **− Hậu điều kiện (Post-condition):** Kết quả học tập của sinh viên được hiển thị trên màn hình | |
| **− Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Đăng nhập vào hệ thống |  |
|  | 1. Xác thực thông tin đăng nhập của sinh viên |
| 1. Chọn chức năng "Xem kết quả học tập". |  |
|  | 1. Hiển thị danh sách các kỳ học tập và các học phần đã hoàn thành |
| 1. Chọn kỳ học hoặc học phần muốn xem kết quả |  |
|  | 1. Hiển thị kết quả học tập chi tiết cho kỳ học hoặc học phần đó (bao gồm điểm số, xếp loại, nhận xét của giảng viên nếu có). |
| **− Luồng sự kiện thay thế (alternate flow):** | |
|  |  |

Bảng 8 : UC05 – Xem kết quả học tập

A diagram of a system

Description automatically generated

Hình 17: UC05\_Activity

|  |  |
| --- | --- |
| **− Tên use case:** UC06\_Xem lịch giảng dạy | |
| **− Mô tả sơ lược:** Cho phép giáo viên xem lịch giảng dạy của mình khi đăng nhập vào hệ thống quản lý đào tạo | |
| **− Actor chính:** giáo viên | |
| **− Actor phụ:** Hệ thống quản lý đào tạo | |
| **− Tiền điều kiện (Pre-condition):** giáo viên đã đăng nhập vào hệ thống quản lý đào tạo | |
| **− Hậu điều kiện (Post-condition):** Lịch giảng dạy của giáo viên được hiển thị trên màn hình | |
| **− Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Đăng nhập vào hệ thống |  |
|  | 1. Xác thực thông tin đăng nhập của giáo viên |
| 1. Chọn chức năng "Xem lịch giảng dạy". |  |
|  | 1. Hiển thị lịch giảng dạy của giáo viên cho kỳ học hiện tại |
| 1. Kiểm tra lịch giảng dạy và các thông tin liên quan( thời gian, phòng học, lớp học,...) |  |
| **− Luồng sự kiện thay thế (alternate flow):** | |
|  |  |

Bảng 9 : UC06 – Xem lịch giảng dạy

**A diagram with blue text and black dots

Description automatically generated**

Hình 18: UC06\_Activity

|  |  |
| --- | --- |
| **− Tên use case:** UC07\_Nhập điểm | |
| **− Mô tả sơ lược:** Giáo viên có thể nhập điểm số cho các sinh viên thuộc lớp giảng dạy của mình thông qua hệ thống quản lý đào tạo. | |
| **− Actor chính:** giáo viên | |
| **− Actor phụ:** Hệ thống quản lý đào tạo | |
| **− Tiền điều kiện (Pre-condition):** Giáo viên đã đăng nhập vào hệ thống quản lý đào tạo và đã có danh sách lớp giảng dạy của mình. | |
| **− Hậu điều kiện (Post-condition):** Điểm số của sinh viên được lưu lại trong hệ thống và sinh viên có thể xem được kết quả. | |
| **− Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Đăng nhập vào hệ thống |  |
|  | 1. Xác thực thông tin đăng nhập của giáo viên |
| 1. Chọn chức năng "Nhập điểm". |  |
|  | 1. Hiển thị danh sách các lớp giảng dạy của giáo viên |
| 1. Chọn lớp học cần nhập điểm |  |
|  | 1. Hiển thị danh sách sinh viên trong lớp học đó |
| 1. Nhập điểm cho từng sinh viên |  |
|  | 1. Xác nhận và lưu điểm số |
|  | 1. Gửi thông báo xác nhận điểm số đã được lưu thành công |
| **− Luồng sự kiện thay thế (alternate flow):** | |
|  |  |

Bảng 10 : UC07 – Nhập điểm

|  |  |
| --- | --- |
| **− Tên use case:** UC08\_Xem tiến độ giảng dạy | |
| **− Mô tả sơ lược:** Cho phép giáo viên theo dõi và đánh giá sự phát triển và hoàn thành các mục tiêu trong học tập của sinh viên. | |
| **− Actor chính:** giáo viên | |
| **− Actor phụ:** Hệ thống quản lý đào tạo | |
| **− Tiền điều kiện (Pre-condition):** Giáo viên đã đăng nhập vào hệ thống quản lý đào tạo và đã có danh sách lớp giảng dạy của mình.. | |
| **− Hậu điều kiện (Post-condition):** Giáo viên có thể theo dõi tiến độ giảng dạy và đánh giá sự phát triển của sinh viên. | |
| **− Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Đăng nhập vào hệ thống |  |
|  | 1. Xác thực thông tin đăng nhập của giáo viên |
| 1. Chọn chức năng "Xem tiến độ giảng dạy". |  |
|  | 1. Hiển thị danh sách các lớp giảng dạy của giáo viên |
| 1. Chọn lớp học cần xem tiến độ |  |
|  | 1. Hiển thị tiến độ giảng dạy của lớp học đó( bao gồm các bài kiểm tra thường kỳ, điểm số, nhận xét của giáo viên,...) |
| 1. Đánh giá sự phát triển của sinh viên dựa trên các tiêu chí cụ thể |  |
| **− Luồng sự kiện thay thế (alternate flow):** | |
|  |  |

Bảng 11 : UC08 – Xem tiến độ giảng dạy

|  |  |
| --- | --- |
| **− Tên use case:** UC09\_Quản lý giảng viên | |
| **− Mô tả sơ lược:** Cho phép nhân viên quản lý thông tin của giảng viên, bao gồm thêm, xóa, sửa thông tin và xem danh sách giảng viên. | |
| **− Actor chính:** Nhân viên | |
| **− Actor phụ:** Hệ thống quản lý đào tạo | |
| **− Tiền điều kiện (Pre-condition):** Nhân viên đã đăng nhập vào hệ thống quản lý đào tạo. | |
| **− Hậu điều kiện (Post-condition):** Thông tin giảng viên được cập nhật và lưu lại trong hệ thống. | |
| **− Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Đăng nhập vào hệ thống |  |
|  | 1. Xác thực thông tin đăng nhập của nhân viên |
| 1. Chọn chức năng "Quản lý nhân viên". |  |
|  | 1. Hiển thị danh sách giảng viên hiện tại |
| 1. Thực hiện các chức năng quản lý:   - Thêm giảng viên: nhập thông tin giảng viên mới và lưu vào hệ thống  - Sửa giảng viên: chọn giảng viên cần sửa thông tin, cập nhật thông tin và lưu lại  - Xóa giảng viên: chọn giảng viên cần xóa và xác nhận xóa |  |
|  | 1. Gửi thông báo xác nhận thông tin giảng viên đã được cập nhật thành công |
| **− Luồng sự kiện thay thế (alternate flow):** | |
|  |  |

Bảng 12 : UC09 – Quản lý giảng viên

|  |  |
| --- | --- |
| **− Tên use case:** UC010\_Quản lý chương trình học | |
| **− Mô tả sơ lược:** Cho phép nhân viên quản lý thông tin chương trình học, bao gồm thêm, xóa, sửa thông tin và xem danh sách chương trình học. | |
| **− Actor chính:** Nhân viên | |
| **− Actor phụ:** Hệ thống quản lý đào tạo | |
| **− Tiền điều kiện (Pre-condition):** Nhân viên đã đăng nhập vào hệ thống quản lý đào tạo. | |
| **− Hậu điều kiện (Post-condition):** Thông tin chương trình học được cập nhật và lưu lại trong hệ thống. | |
| **− Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Đăng nhập vào hệ thống |  |
|  | 1. Xác thực thông tin đăng nhập của nhân viên |
| 1. Chọn chức năng "Quản lý nhân viên". |  |
|  | 1. Hiển thị danh sách giảng viên hiện tại |
| 1. Thực hiện các chức năng quản lý:   - Thêm giảng viên: nhập thông tin giảng viên mới và lưu vào hệ thống  - Sửa giảng viên: chọn giảng viên cần sửa thông tin, cập nhật thông tin và lưu lại  - Xóa giảng viên: chọn giảng viên cần xóa và xác nhận xóa |  |
|  | 1. Gửi thông báo xác nhận thông tin giảng viên đã được cập nhật thành công |
| **− Luồng sự kiện thay thế (alternate flow):** | |
|  |  |

Bảng 13 : UC10 – Quản lý chương trình học

|  |  |
| --- | --- |
| **− Tên use case:** UC011\_Quản lí học phần | |
| **− Mô tả sơ lược:** Cho phép nhân viên quản lý thông tin học phần, bao gồm thêm, xóa, sửa thông tin và xem danh sách học phần. | |
| **− Actor chính:** Nhân viên | |
| **− Actor phụ:** Hệ thống quản lý đào tạo | |
| **− Tiền điều kiện (Pre-condition):** Nhân viên đã đăng nhập vào hệ thống quản lý đào tạo. | |
| **− Hậu điều kiện (Post-condition):** Thông tin học phần được cập nhật và lưu lại trong hệ thống. | |
| **− Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Đăng nhập vào hệ thống |  |
|  | 1. Xác thực thông tin đăng nhập của nhân viên |
| 1. Chọn chức năng "Quản lý học phần". |  |
|  | 1. Hiển thị danh sách học phần hiện tại |
| 1. Thực hiện các chức năng quản lý:   - Thêm học phần: nhập thông tin giảng viên mới và lưu vào hệ thống  - Sửa học phần: chọn học phần cần sửa thông tin, cập nhật thông tin và lưu lại  - Xóa học phần: chọn học phần cần xóa và xác nhận xóa |  |
|  | 1. Gửi thông báo xác nhận thông tin học phần đã được cập nhật thành công |
| **− Luồng sự kiện thay thế (alternate flow):** | |
|  |  |

Bảng 14 : UC11 – Quản lý học phần

|  |  |
| --- | --- |
| **− Tên use case:** UC012\_Quản lý sinh viên | |
| **− Mô tả sơ lược:** Cho phép nhân viên quản lý thông tin sinh viên, bao gồm thêm, xóa, sửa thông tin và xem danh sách sinh viên. | |
| **− Actor chính:** Nhân viên | |
| **− Actor phụ:** Hệ thống quản lý đào tạo | |
| **− Tiền điều kiện (Pre-condition):** Nhân viên đã đăng nhập vào hệ thống quản lý đào tạo. | |
| **− Hậu điều kiện (Post-condition):** Thông tin sinh viên được cập nhật và lưu lại trong hệ thống. | |
| **− Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Đăng nhập vào hệ thống |  |
|  | 1. Xác thực thông tin đăng nhập của nhân viên |
| 1. Chọn chức năng "Quản lý sinh viên". |  |
|  | 1. Hiển thị danh sách sinh viên hiện tại |
| 1. Thực hiện các chức năng quản lý:   - Thêm sinh viên: nhập thông tin giảng viên mới và lưu vào hệ thống  - Sửa sinh viên: chọn sinh viên cần sửa thông tin, cập nhật thông tin và lưu lại  - Xóa sinh viên: chọn sinh viên cần xóa và xác nhận xóa |  |
|  | 1. Gửi thông báo xác nhận thông tin sinh viên đã được cập nhật thành công |
| **− Luồng sự kiện thay thế (alternate flow):** | |
|  |  |

Bảng 15 : UC012\_Quản lý sinh viên

3.5. Data view

A screenshot of a computer program

Description automatically generated

Hình 19: ER Diagram

3.6. Các mẫu thiết kế và cơ chế chung

3.6.1. API Gateway

**Mô tả**: API Gateway là một điểm truy cập duy nhất cho tất cả các service, cung cấp một cách để quản lý, bảo mật và giám sát các yêu cầu tới các service.

**Lợi ích**:

* Đơn giản hóa giao tiếp giữa client và các service.

**Cách hoạt động:**

A diagram of a software application

Description automatically generated

Hình 20 : Luồng hoạt động của API Gateway

* Client gửi HTTP request tới API Gateway.
* Yêu cầu được xử lý bởi Authenticate filter. Authenticate filter thực hiện các nhiệm vụ như kiểm tra tính hợp lệ (validation), xác thực (authentication). Sau khi authenticate filter xử lý xong, request được chuyển đến route filter.
* Route filter xác định service đích mà request cần được chuyển đến (ví dụ như service A hoặc service B). Router filter có thể sử dụng các quy tắc định tuyến (routing rules) dựa trên URL, headers hoặc thông tin khác trong request để quyết định đích đến.
* Sau khi định tuyến route filter chuyển yêu cầu đến service tương ứng.
* Yêu cầu được gửi đến service đích và service này tiến hành xử lý request. Sau đó gửi phản hồi đến API Gateway.
* API Gateway nhận phản hồi từ service đích và chuyển tiếp phản hồi này đến client.

A diagram of a service register

Description automatically generated

Hình 21 : Luồng đăng nhập qua API Gateway

* Client gửi yêu cầu đăng nhập đến API Gateway kèm theo dữ liệu account.
* API Gateway nhận request và chuyển tiếp đến identity-service qua endpoint /api/token.
* Identity service nhận request và kiểm tra thông tin tài khoản. Nếu tài khoản tồn tại trong hệ thống thì sẽ tạo token và phản hồi lại API Gateway.
* API Gateway nhận phản hồi từ identity service và chuyển lại phản hồi này tới client
* Client dùng token này để thực hiện xác thực cho những request lần sau.

3.6.2. Service Discovery

**Mô tả** : Cho phép các service tự động tìm thấy nhau thông qua một registry.  
**Lợi ích**:  
- Giảm thiểu sự phụ thuộc vào cấu hình tĩnh.  
- Tăng khả năng mở rộng và linh hoạt của hệ thống.

**Cách hoạt động:**

A diagram of a service register

Description automatically generated

Hình 22 : Luồng hoạt động của service registry

* Khi một service mới được triển khai hoặc khởi chạy, nó sẽ gửi thông tin về bản thân mình bao gồm tên service, địa chỉ IP, port, và các thông tin khác đến service registry.
* Service registry sẽ lưu trữ thông tin này. Thông tin sẽ được cập nhật định kỳ hoặc khi có sự thay đổi trong hạ tầng dịch vụ.
* Khi 1 service cần tương tác với service khác, nó sẽ truy vấn service registry để tìm kiếm thông tin về service đó.
* Khi service đã tìm kiếm được thông tin về service cần tương tác nó sẽ nhận thông tin về địa chỉ IP, port về các instance service có sẵn và cách kết nối với chúng.
* Load balancer sẽ sử dụng thông tin này để phân phối tải giữa các instance service, giúp tối đa hiệu suất của hệ thống.
* Client service sẽ kết nối và gửi yêu cầu đến một trong các instance service được chọn thông qua load balancer.
* Sau đó client service sẽ nhận được phản hồi từ service đó và tiếp tục quá trình xử lý.

3.6.3. Circuit Breaker

**Mô tả**: Ngăn ngừa các lỗi liên tục và bảo vệ hệ thống khỏi các dịch vụ bị lỗi bằng cách ngắt kết nối khi phát hiện các lỗi quá nhiều.

**Lợi ích**:

- Tăng cường độ tin cậy của hệ thống.

- Giảm tải cho các dịch vụ đang gặp sự cố.

**Cách hoạt động:**

**A diagram of a speaker

Description automatically generated**

Hình 23 : Luồng hoạt động của circuit breaker

**- Closed State:** Trong trạng thái bình thường, tất cả các yêu cầu đều được phép đi qua.

- **Open State:** Khi số lượng lỗi vượt qua một ngưỡng nhất định, Circuit Breaker chuyển sang trạng thái mở và chặn tất cả các yêu cầu tiếp theo để tránh làm tình trạng tồi tệ hơn.

- **Half-Open State:** Sau một khoảng thời gian chờ, Circuit Breaker cho phép một số ít yêu cầu đi qua để kiểm tra xem dịch vụ đã phục hồi hay chưa. Nếu các yêu cầu này thành công, nó chuyển lại về trạng thái đóng; nếu không, nó trở lại trạng thái mở.

Chương 4: Kết luận

Thiết kế kiến trúc hệ thống quản lý lớp học tín chỉ sử dụng kiến trúc dựa trên dịch vụ đã được xây dựng với mục tiêu tạo ra một hệ thống hiệu quả, dễ bảo trì và có khả năng mở rộng. Hệ thống được chia thành nhiều dịch vụ độc lập, mỗi dịch vụ chịu trách nhiệm cho một chức năng cụ thể như quản lý sinh viên, quản lý khóa học, đăng ký lớp học.

Trong quá trình phát triển kiến trúc, chúng em đã đưa ra các quyết định thiết kế quan trọng sau:

* **Phân chia dịch vụ**: Các chức năng chính của hệ thống được phân chia thành các dịch vụ riêng biệt để giảm thiểu sự phụ thuộc lẫn nhau và tăng cường tính linh hoạt.
* **Quản lý dữ liệu tập trung**: Sử dụng một cơ sở dữ liệu tập trung giúp đảm bảo tính nhất quán và toàn vẹn của dữ liệu.
* **Bảo mật**: Áp dụng các biện pháp bảo mật như xác thực và phân quyền để đảm bảo an toàn cho dữ liệu và hệ thống.

Thiết kế kiến trúc service-based sử dụng WebClient mang lại nhiều lợi ích cho hệ thống quản lý lớp học tín chỉ:

* **Khả năng mở rộng**: Hệ thống dễ dàng mở rộng bằng cách thêm hoặc nâng cấp các dịch vụ một cách độc lập.
* **Dễ bảo trì**: Các dịch vụ riêng biệt giúp việc bảo trì và cập nhật trở nên dễ dàng hơn mà không ảnh hưởng đến toàn bộ hệ thống.
* **Tính nhất quán của dữ liệu**: Quản lý dữ liệu tập trung giúp duy trì tính nhất quán và toàn vẹn.

**Rủi ro** : Khi số lượng dịch vụ tăng lên, việc quản lý sẽ trở nên phức tạp hơn, lúc này cần nghĩ đến việc chia nhỏ database ra. Thông thường sẽ là 3-12 service cho 1 database.

TÀI LIỆU THAM KHẢO

[1] Richards, M., & Ford, N. (2020). Fundamentals of Software Architecture: An Engineering Approach. O'Reilly Media.

[2] Richards, M. (n.d.). Software Architecture Monday. Developer to Architect. Retrieved May 19, 2024, from <https://www.developertoarchitect.com/lessons/>

[3] Fowler, M. (n.d.). CircuitBreaker. Martin Fowler. Retrieved May 19, 2024, from https://martinfowler.com/bliki/CircuitBreaker.html