BỘ CÔNG THƯƠNG

TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP THÀNH PHỐ HỒ CHÍ MINH

**KHOA khoa CÔNG NGHỆ THÔNG TIN**



**MÔN: KIẾN TRÚC PHẦN MỀM**

**ĐỒ ÁN NGHIÊN CỨU**

***Đề tài:***

**HỆ THỐNG QUẢN LÝ HỌC PHẦN TÍN CHỈ**

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

**Lớp:** DHKTPM16A

**GVHD:** TS. Võ Văn Hải

Nhóm 10 – Sinh viên thực hiện

1. Trương Dương Minh Nhật – 21028411
2. Trần Quang Hiển – 20036311
3. Nguyễn Đức Tài – 2102454

Thành phố Hồ Chí Minh, 19 tháng 05 năm 2024

**Mục Lục**

[1. Phân tích nghiệp vụ 6](#_Toc167053837)

[2. Các bản vẽ thiết kế 8](#_Toc167053838)

[a. Sơ đồ use-case 8](#_Toc167053839)

[b. Đặc tả use-case 8](#_Toc167053840)

[ Danh sách đặc tả 8](#_Toc167053841)

[ Đặc tả 10](#_Toc167053842)

[c. Activity 14](#_Toc167053843)

[ UC050: 14](#_Toc167053844)

[ UC060: 15](#_Toc167053845)

[ UC011: 15](#_Toc167053847)

[ UC017: 17](#_Toc167053849)

[d. Class 18](#_Toc167053850)

[e. Package 19](#_Toc167053851)

[3. Mô tả kiến trúc được chọn cho đề tài 20](#_Toc167053852)

[a. Mô hình kiến trúc 20](#_Toc167053853)

[b. Giải thích sơ đồ 20](#_Toc167053854)

[c. Lý do chọn kiến trúc 21](#_Toc167053855)

[d. Ưu điểm và nhược điểm khi chọn kiến trúc 22](#_Toc167053856)

# Phân tích nghiệp vụ

* Nhà trường yêu cầu tạo một phần mềm website có thể quản lý vấn đề đăng ký học phần tín chỉ cho các sinh viên, nhà trường muốn một phần mềm website có thể chịu tải được số lượng đông sinh viên khi tham gia đăng ký học phần cùng một lúc khi thời điểm đăng ký học phần mở, cùng lúc đó nhà trường cần một giao diện đơn giản cũng như đầy đủ các chức năng cần thiết để sử dụng. Nhà trường muốn có hệ thống sao cho sinh viên, giảng viên và giáo vụ khoa đều có thể sử dụng.
* Đối với sinh viên, các sinh viên muốn có thể sử dụng hệ thống này thì trước hết phải thực hiện đăng nhập tài khoản với mã số sinh viên và mật khẩu đã được nhà trường cung cấp lúc sinh viên tham gia học chính thức tại trường, sinh viên có thực hiện các dịch vụ của hệ thống cung cấp như: xem thông tin sinh viên hoặc điều chỉnh thay đổi thông tin nếu cần thiết, sinh viên có thể xem được điểm các môn học mà đã đăng ký theo kỳ học cũng như xem thời khoá biểu để biết lịch học, lịch thi. Khi đến mùa đăng ký học phần, sinh viên sẽ thực hiện đăng ký học phần các môn học do chính giáo vụ khoa mở cho sinh viên đăng ký, sinh viên thực hiện chọn năm học cũng như kỳ học để hiển thị danh sách các môn học có thể đăng ký được do giáo vụ khoa them, sinh viên chọn một môn học để hiển thị danh sách các lớp có sẵn để sinh viên có thể chọn lớp đúng theo ý muốn để học, mỗi lớp học phần đều luôn luôn có 1 lớp lý thuyết và trong trường hợp môn đó có thực hành sẽ có lớp thực hành riêng cũng như sẽ có nhiều ca lớp thực hành để sinh viên có thể lựa chọn ca sao cho phù hợp và đăng ký, khi nhấn nút đăng ký, sinh viên sẽ được gửi một thư email để xác nhận rằng việc mà mình đăng ký sẽ chấp nhận hoàn toàn và không thể nào huỷ, sau khi sinh viên đã thực hiện chấp nhận đăng ký thông qua xác nhận email, sinh viên sẽ nhận được giấy thông báo tiền học phí của môn mà sinh viên đã đăng ký. Ngoài ra nhà trường muốn cung cấp chức năng xét tốt nghiệp cho sinh viên nào đủ yêu cầu số lượng tín chỉ đã học để có thể thực hiện xét tốt nghiệp.
* Đối với giảng viên, giảng viên có thể xem hay cập nhật thông tin cá nhân của bản than, ngoài ra giảng viên có thể xem được thời khoá biểu để biết được lịch dạy cũng như sinh viên có thể thực hiện nhập điểm của sinh viên của mình đang dạy, việc nhập điểm của sinh viên chỉ dành cho giảng viên dạy lý thuyết nhập, đối với giảng viên thực hành sẽ không thể nhập điểm, thay vào đó vào gần cuối đến thời điểm nhập điểm thì giảng viên thực hành có thể gửi bảng điểm hay bảng đánh giá đến giảng viên lý thuyết để giảng viên lý thuyết có thể dựa vào đó có thể cho điểm phù hợp.
* Đối với giáo vụ, ngoài việc giáo vụ khoa có thể xem thông tin cũng như chỉnh được thông tin bản than thì giáo vụ khoa có thể thực hiện quản lý đăng ký lớp học phần, giáo vụ thể thực hiện them một môn học vào kỳ học sắp tới để có thể chuẩn bị them các lớp học lý thuyết và thực hành. Trong quá trình thêm môn học, giáo vụ cần chú ý vấn đề xếp thời gian, lịch thời khoá biểu sao cho phù hợp tránh vấn đề bị trùng lịch với nhau, giáo vụ khoa là người sẽ quyết định giáo viên sẽ dạy môn nào trong lớp học phần nào.

# Các bản vẽ thiết kế

## Sơ đồ use-case

A diagram of a diagram

Description automatically generated

## Đặc tả use-case

### Danh sách đặc tả

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Tên Use case | Mô tả ngắn gọn Use case | Chức năng | Ghi chú |
| **UC001** | **Xem lịch học sinh viên** | Xem thông tin chi tết lịch học | Xem chi tiết lịch học |  |
| **UC002** | **Xem thông tin cá nhân** | Xem thông tin chi tiết cá nhân của sinh viên | Xem chi tiết thông tin sinh viên |  |
| **UC003** | **Cập nhật thông tin sinh viên** | Cập nhật thông tin mới của sinh viên | Cập nhật thông tin sinh viên |  |
| **UC004** | **Xem thông tin kết quả học tập** | Xem thông tin chi tiết kết quả học tập sinh viên | Xem chi tiết kết quả học tập |  |
| **UC005** | **Đăng ký học phần của sinh viên** | Đăng ký học phần của sinh viên khi đến thời điểm đăng ký môn học | Đăng ký học phần sinh viên |  |
| **UC006** | **Đăng ký tốt nghiệp** | Đăng ký tốt nghiệp sau khi sinh viên đủ điều kiện | **Đăng ký tốt nghiệp** |  |
| **UC007** | **Đăng nhập tài khoản** | Đăng nhập vào hệ thống bằng tài khoản | Xác định quyền trên hệ thống |  |
| **UC008** | **Xem thông tin cá nhân giáo vụ** | Xem thông tin chi tiết cá nhân của giáo vụ | Xem thông tin cá nhân giáo vụ |  |
| **UC009** | **Cập nhật thông tin cá nhân giáo vụ** | Cập nhật thông tin cá nhân giáo vụ | **Cập nhật thông tin cá nhân giáo vụ** |  |
| **UC010** | **Thêm môn học vào kỳ tới** | Giáo vụ sẽ them môn học vào kỳ tới để mở | **Thêm môn học phần của giáo vụ** |  |
| **UC011** | **Thêm lớp học phần lý thuyết và thực hành** | Giáo vụ sẽ them lớp học phần vào môn học, trong lớp học phần giáo vụ sẽ them lớp học lý thuyết và thực hành | **Thêm lớp học phần của giáo vụ** |  |
| **UC012** | **Cập nhật trạng thái lớp học phần** | Giáo vụ cập nhật trạng thái lớp học phần | **Cập nhật trạng thái lớp học phần** |  |
| **UC013** | **Xem thông tin cá nhân giáo viên** | Xem thông tin chi tiết cá nhân giáo viên | **Xem thông tin cá nhân giáo viên** |  |
| **UC014** | **Cập nhật thông tin cá nhân giáo viên** | Cập nhật thông tin chi tiết cá nhân giáo viên | **Cập nhật thông tin cá nhân giáo viên** |  |
| **UC015** | **Xem thông tin lịch dạy giáo viên** | Xem thông tin lịch học của giáo viên | **Xem thông tin lịch dạy giáo viên** |  |
| **UC016** | **Xem thông tin danh sách lớp** | Xem thông tin danh sách lớp học do giáo viên quản | **Xem thông tin danh sách lớp** |  |
| **UC017** | **Nhập điểm điểm của sinh viên** | Nhập điểm của sinh viên do giáo viên nhập | **Nhập điểm điểm của sinh viên** |  |

### Đặc tả

|  |  |
| --- | --- |
| **Use case: UC-005\_Đăng ký học phần sinh viên** | |
| Actor | Sinh viên |
| Mô tả: | Use case thực hiện việc thêm học phần cho sinh viên |
| Điều kiện trước: | * Đăng nhập hệ thống thành công với quyền sinh viên * Chọn chức năng đăng ký học phần |
| Điều kiện sau: | Lưu vào trong CSDL và thông báo đăng ký thành công. |
| **Luồng sự kiện chính (Basic flows)** | |
| Actor: | System: |
| 1. Sinh viên chọn chức năng “Đăng ký học phần” |  |
|  | 2. Hệ thống hiển thị form đăng ký học phần |
| 3. Sinh viên chọn học kì và chọn nút radio học mới hoặc chọn nút radio học cải thiện |  |
|  | 4. Hệ thống hiển thị danh sách dữ liệu các môn học vào table |
| 6. Sinh viên chọn môn học dòng dữ liệu trên table |  |
|  | 7. hệ thống hiển thị các lớp có thể đăng ký học phần |
| 8. Sinh viên chọn một lớp học cần, chọn nhóm thực hành(nếu có) và chọn nút đăng ký |  |
|  | 9. hệ thống kiểm tra điều cần thiết để đăng ký học phần |
|  | 10. hệ thống thông báo đăng ký thành công |
| 11. sinh viên xác nhận và thoát |  |
| Luồng sự kiện phụ (Alternative Flows): | |
|  | 9.1. hệ thống thông báo lớp học đã đầy hoặc không đủ điều kiện |
| 9.2 Sinh viên xác nhận và quay trở lại bước 8 |  |

|  |  |
| --- | --- |
| **Use case: UC-006\_Đăng ký tốt nghiệp** | |
| Actor | Sinh viên |
| Mô tả: | Use case thực hiện việc đăng ký tốt nghiệp |
| Điều kiện trước: | * Đăng nhập hệ thống thành công với quyền sinh viên * Chọn chức năng đăng ký tốt nghiệp |
| Điều kiện sau: | Lưu vào trong CSDL và thông báo đăng ký thành công. |
| **Luồng sự kiện chính (Basic flows)** | |
| Actor: | System: |
| 1. Sinh viên chọn chức năng “Đăng ký tốt nghiệp” |  |
|  | 2. Hệ thống hiển thị form đăng ký thành công |
| 3. sinh viên điền thông tin vào form và bấm xác nhận |  |
|  | 4. Hệ thống xử lý các thông tin trong form qua điều kiện |
|  | 5. Hệ thống lưu vào PDF và csdl |
|  | 6. hệ thống hiển thị thông tin đơn đăng ký |
| 7. sinh viên xác nhận và thoát |  |
| Luồng sự kiện phụ (Alternative Flows): | |
|  | 4.1 Hệ thống thông báo thông tin bị sai lệch |
| 4.2 Sinh viên quay lại bước 3 |  |

|  |  |
| --- | --- |
| **Use case: UC-011\_ Thêm lớp học phần lý thuyết và thực hành** | |
| Actor | Giáo vụ |
| Mô tả: | Use case thực hiện việc thêm lớp học phần của giáo vụ |
| Điều kiện trước: | * Đăng nhập hệ thống thành công với quyền giáo vụ |
| Điều kiện sau: | Lưu vào trong CSDL và thông báo thêm thành công. |
| **Luồng sự kiện chính (Basic flows)** | |
| Actor: | System: |
| 1. Giáo vụ chọn chức năng “Thêm lớp học phần của giáo vụ” |  |
|  | 2. Hệ thống hiển thị form mở lớp học phần |
| 3. Giáo vụ chọn học kì, chọn môn học và nhấn nút thêm lớp học phần |  |
|  | 4. hệ thống thêm dữ liệu vào bảng thông tin lớp học phần |
| 5. Giáo vụ chọn dòng dữ liệu của bảng thông tin lớp học phần và nhấn nút thêm lớp học phần |  |
|  | 5. Hệ thống hiển thị model form đăng ký lớp học phần |
| 6. Giáo vụ điền thông tin về lớp học phần lý thuyết |  |
| 7. Giáo vụ điền thông tin về lớp học phần thực hành và nhấn nút thêm lớp thực hành (nếu có) |  |
|  | 8. Hệ thống thêm dữ liệu vào bảng thông tin chi tiết lớp học thực hành |
| 9. Giáo vụ nhấn nút lưu |  |
|  | 10. hệ thống kiểm tra so sánh dữ liệu trong CSDL |
|  | 11. hệ thống thêm dữ liệu lớp học lý thuyết và lớp học thực hành vào bảng thông tin học phần |
|  | 11. Hệ thống thông báo mở lớp thành công |
| 11. Giáo vụ xác nhận và thoát |  |
| Luồng sự kiện phụ (Alternative Flows): | |
|  | 10.1 Hệ thống thông báo học phần đã có sẵn |
| 10.2 Giáo vụ quay về bước 6 |  |

|  |  |
| --- | --- |
| **Use case: UC-017\_Nhập điểm điểm cho sinh viên** | |
| Actor | Giáo viên |
| Mô tả: | Use case thực hiện nhập điểm sinh viên |
| Điều kiện trước: | * Đăng nhập hệ thống thành công với chức vụ sinh viên * Chọn chức năng xem lịch học |
| Điều kiện sau: | Lưu vào trong CSDL và thông báo cập nhật thành công |
| **Luồng sự kiện chính (Basic flows)** | |
| Actor: | System: |
| 1. Giáo viên chọn chức năng “Xem thông tin danh sách lớp” |  |
| 2. Giáo viên chọn học kì và chọn lớp học |  |
|  | 3. hệ thống hiển thị bảng danh sách thông tin sinh viên |
| 4. Giáo viên chọn nút nhập điểm của sinh viên |  |
|  | 5. hệ thống hiển thị model form nhập điểm cho sinh viên |
| 6. giáo viên nhập điểm sinh viên và bấm nút xác nhận |  |
|  | 7. hệ thống kiểm tra dữ liệu |
|  | 8.hệ thống thêm dữ liệu vào CSDL và thông báo cập nhật thành công |
| 9. Giáo viên xác nhận thành công và thoát |  |
| Luồng sự kiện phụ (Alternative Flows): | |
|  | 7.1 hệ thống thông báo cập nhật thất bại |
| 7.2 giáo viên quay trở lại bước 6 |  |

## Activity

### UC050:

A diagram of a diagram

Description automatically generated

* + - UC060:

A diagram of a flowchart

Description automatically generated

* + - UC011:

A screenshot of a computer

Description automatically generated

* + - UC017:

A diagram of a flowchart

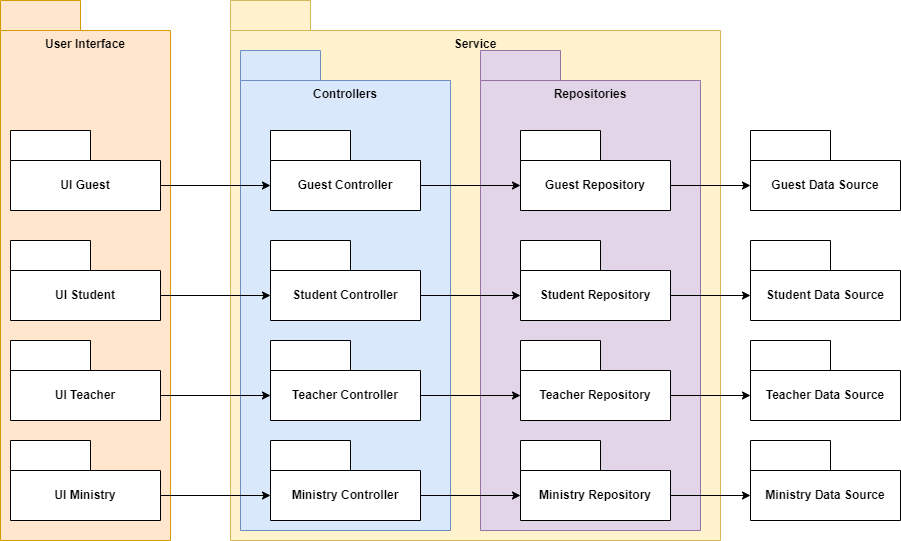
Description automatically generated

## Class

A screenshot of a computer

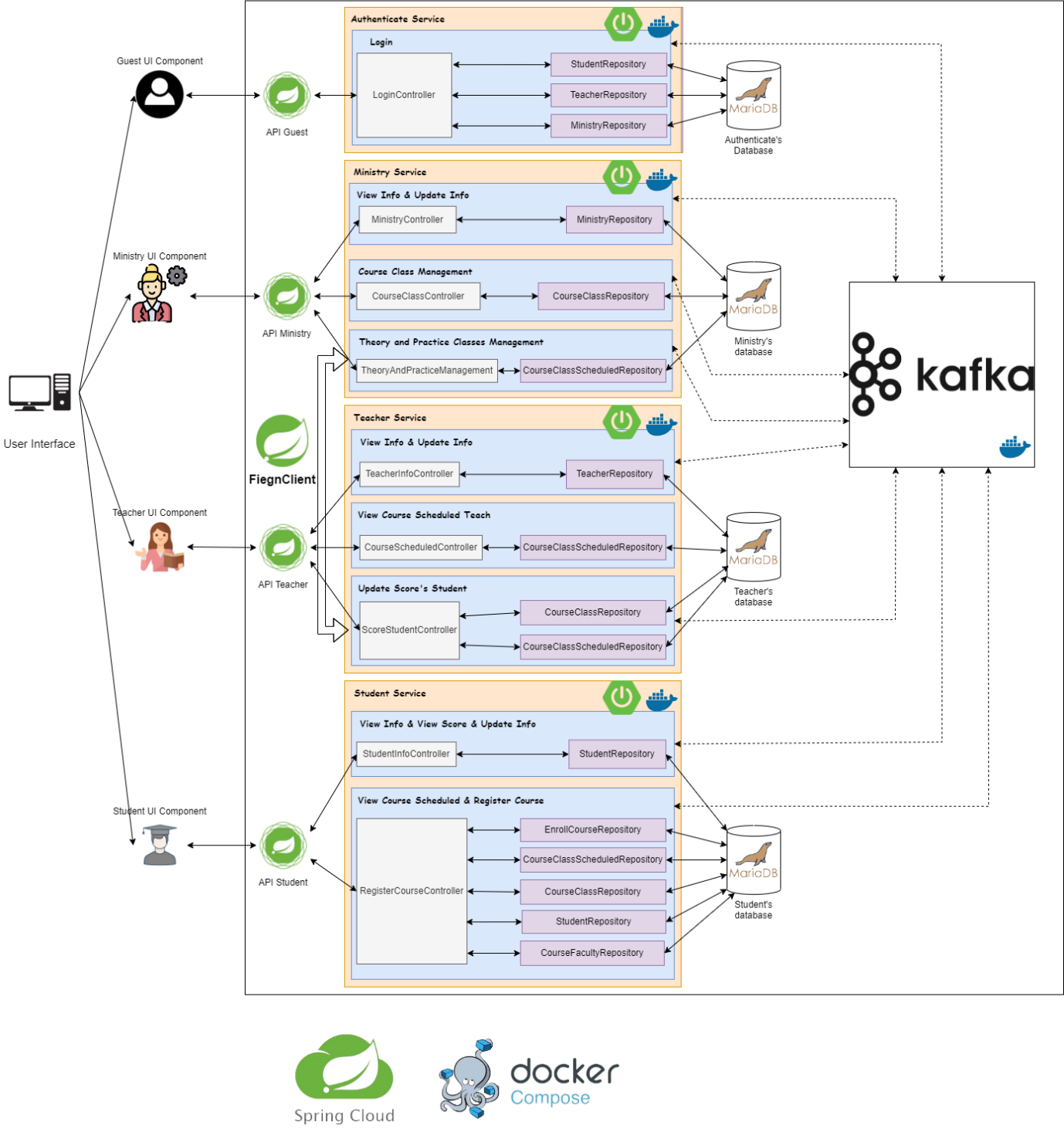
Description automatically generated

## Package



# Mô tả kiến trúc được chọn cho đề tài

## Mô hình kiến trúc



## Giải thích sơ đồ

* Từ phía người dùng “User Interface” họ có thể sẽ thuộc nhiều actors khác nhau, trước khi thực hiện các chức năng của các actors: Student, Teacher hay Ministry. Các người dùng, họ sẽ là những người “Guest” truy cập vào trang đăng nhập thông qua "Guest UI Component”, khi thực hiện đăng nhập, “Guest” sẽ gửi một request đến server để xử lý, server nhận và xử lý trả ngược lại cho “Guest”.
* Khi đăng nhập thành công thì UI sẽ được hiển thị trang chủ ứng với actors mà người dùng đó đăng nhập thông qua “Role”.
* Trong service “Authenticate”, chỉ có 1 service con duy nhất là “Login” chủ yếu để thực hiện xử lý các request liên quan đến đăng nhập.
* Trong service “Student”, có 5 service con (3 service nhỏ và 2 service lớn) trong Student, đối với 3 service nhỏ là: “View Info”, “Update Info” và “View Score” (“Xem thông tin”, “Cập nhật thông tin” và “Xem điểm”) của mỗi cá nhân sinh viên, các chức năng này phụ để hỗ trợ cho sinh viên dễ dàng theo dõi. Còn 2 service lớn như: “View Course Scheduled” và “Register Course” (“Xem danh sách lớp học phần” và “Đăng ký môn học”) hai chức năng chủ yếu để sinh viên thực hiện đăng ký học phần tín chỉ.
* Trong service “Ministry”, có 5 chức năng: “View Info / Update Info”, “Course Class Management”, “Theory and Practice Class Management” (hay “Xem / Cập nhật”, “Quản lý các môn học trong kỳ”, “Quản lý lớp học lý thuyết và thực hành”).
* Trong service “Teacher”, có 5 chức năng: “View Info / Update Info”, “View Course Scheduled” và “Update Score Student” (hay “Xem / Cập nhật thông tin”, “Xem lịch thời khoả biểu dạy”, “Cập nhật điểm số sinh viên”). Trong trường hợp này đối với chức năng “Cập nhật điểm số sinh viên” (Update Score Student), sẽ phụ thuộc vào service “Ministry” của chức năng “Quản lý lớp học phần lý thuyết và thực hành” thông qua một thư viện hỗ trợ là **FeignClient**.
* Mỗi service sẽ kết nối với cơ sở dữ liệu **MariaDB** chính service đó sử dụng. Chính vì mỗi service sẽ sử dụng các database khác nhau nhưng có thể chung bảng cho nên việc đòi hỏi đồng bộ dữ liệu khá quan trọng, nên chúng em quyết định sử dụng công nghệ **Apache Kafka** để thực hiện vấn đề **gửi** và **nhận** các message giữa các services với nhau.

## Lý do chọn kiến trúc

* Theo đề tài này, nhóm chúng em quyết định sử dụng kiến trúc: Micro-services kết hợp với Micro-frontend, Layered Style và Event-Driven. Vì:
  + Trong đề tài này, nhóm chúng em quyết định bổ sung thêm các nghiệp vụ cũng như các thành phần khác: giáo vụ và giảng viên dẫn đến trong đề tài môn học của nhóm chúng em có 3 actors.
  + Dựa vào sự phân chia các actors như thế nhóm chúng em quyết định chia thành 3 services khác nhau, độc lập lẫn nhau (nhưng thực ra có service này phụ thuộc vào service khác), mỗi một service sẽ đảm nhiệm các chức năng, service con trong chính service của actor đó.
  + Trong một service thì nhóm chúng em sẽ quyết định kết hợp một kiến trúc chính là Layered Style (đa tầng) trong việc phát triển các service con trong một service cha.
  + Khi xây dựng kiến trúc nhất là phần cơ sở dữ liệu do các bảng sẽ có ở nhiều cơ sở dữ liệu khác nhau dẫn đến việc trùng lặp và đòi hỏi sự đồng bộ dữ liệu với nhau giữa các cơ sở dữ liệu cho nên nhóm chúng em sử dụng kiến trúc Event-Driven để có thể thực hiện đồng bộ dữ liệu thông qua vấn đề bắt event và gửi message cho các service khác thông qua Apache Kafka.

## Ưu điểm và nhược điểm khi chọn kiến trúc

* Ưu điểm:
  + Việc sử dụng kiến trúc này giúp cho nhóm chúng em có thể dễ dàng bảo trì và nâng cấp service mà không bị ảnh hưởng đến các service khác, giả sử nhóm chúng em muốn nâng cấp, thay đổi hoặc tuỳ chỉnh, chỉnh sửa trong service Student thì các service khác như: Teacher hay Ministry sẽ không bị ảnh hưởng nhưng vẫn chạy bình thường.
  + Mỗi một service là 1 instance (máy chủ) cho nên khi phân tán ở nhiều máy tính khác nhau (hay được hiểu là các service sẽ chạy trên 1 máy) thì dễ dàng một máy tính chạy trên instance sẽ giải quyết hết mức tối đa hiệu năng của máy tính đó trên riêng 1 service đó nếu service đó có nhiều sự request từ phía client gửi đến.
* Nhược điểm:
  + Chính việc tách nhiều service trên nhiều máy tính khác nhau thì đòi hỏi các máy tính phải duy trì được băng thông mạng sao cho ổn định để dễ dàng kết nối, nhận và phản hồi lại dữ liệu cũng như là kết nối các service khác. Việc băng thông mạng không ổn định sẽ là nguyên nhân chính service đó sẽ bị sụp đổ.
  + Đối với cơ sở dữ liệu, việc tách các service này dẫn đến ở một bảng nhưng sẽ có ở nhiều database khác nhau ở mỗi service khác nhau, chính vì vậy đòi hỏi phải có sự đồng bộ, và chính vấn đề này nhóm chúng em quyết định hi sinh vấn đề trùng lặp dữ liệu để đổi lấy tính hiệu năng cao thông qua các service chạy trên 1 máy tính.