**Case Study: Đăng ký môn học**

1. **Phân tích:**

Thực hiện mô hình hóa dịch vụ REST cho quy trình “Đăng ký môn học”. Quy trình này bao gồm:

* Kiêm tra lớp học phần có còn slot hay không?
* Kiểm tra lớp học phần được chọn của sinh viên có bị trùng lịch học với lớp học phần đã đăng ký.
* Phê duyệt đăng ký của sinh viên.

A diagram of a flowchart

Description automatically generated

1, Phân tách quy trình thành các hành động chi tiết:

* Quy trình “Đăng ký môn học” được chia ra thành các hành động chi tiết sau:
  + Bắt đầu quy trình đăng ký
  + Nhận dữ liệu chi tiết về lịch học
  + Sinh viên chọn vào lịch học theo nhu cầu
  + Xác minh có còn slot hay không?
  + Nếu không còn slot thì gửi thông báo từ chối
  + Xác minh từ chối thủ công
  + Xác minh lịch học có trùng hay không?
  + Nếu lịch học trùng thì gửi thông báo từ chối và kết thúc chương trình
  + Xác minh từ chối thủ công
  + Nếu lịch học không trùng thì xác minh thành công
  + Xác minh sự chấp nhận thủ công
  + Gửi chấp nhận
  + Ghi lại việc đăng ký trong cở sở dữ liệu
  + Kết thúc chương trình

2,Lọc bỏ các hành động không phù hợp:

* Các hành động không phù hợp
  + Xác minh từ chối thủ công
  + Xác minh sự chấp nhận thủ công
* Quy trình “Đăng ký mông học” được chia ra thành các hành động chi tiết sau:
  + Bắt đầu quy trình đăng ký
  + Nhận dữ liệu chi tiết về lịch học
  + Sinh viên chọn vào lịch học theo nhu cầu
  + Xác minh có còn slot hay không?
  + Nếu không còn slot thì gửi thông báo từ chối
  + Xác minh lịch học có trùng hay không?
  + Nếu lịch học trùng thì gửi thông báo từ chối và kết thúc chương trình
  + Nếu lịch học không trùng thì xác minh thành công
  + Gửi chấp nhận
  + Ghi lại việc đăng ký trong cở sở dữ liệu
  + Kết thúc chương trình

3, Xác định ứng viên dịch vụ thực thể

Các hành động không bất khả tri

* + Bắt đầu quy trình đăng ký
  + Xác minh có còn slot hay không?
  + Nếu không còn slot thì gửi thông báo từ chối
  + Xác minh lịch học có trùng hay không?
  + Nếu lịch học trùng thì gửi thông báo từ chối và kết thúc chương trình

Các hành động bất khả tri

* + Nhận dữ liệu chi tiết về lịch học
  + Sinh viên chọn vào lịch học theo nhu cầu
  + Nếu lịch học không trùng thì xác minh thành công
  + Gửi chấp nhận
  + Ghi lại việc đăng ký trong cở sở dữ liệu
  + Kết thúc chương trình

Các hành động bất khả tri được phân loại là các ứng viên năng lực dịch vụ sơ bộ và được nhóm lại tương ứng vào các ứng viên năng lực dịch vụ:

* Ứng viên dịch vụ Lớp học phần (Class):
* GetDetail
* CheckCapacity
* CheckDuplicate
* Ứng viên dịch vụ Lớp học phần đã chọn (SelectedClass):
* SaveClass
* DeleteClass
* SaveSchedule
* GetDetail
* Ứng viên dịch vụ Sinh viên (Student):
* GetDetail.
* Ứng viên dịch vụ Môn học (Subject):
* GetDetail.

4, Xác định logic quy trình cụ thể:

Các hành động không bất khả tri

* + Bắt đầu quy trình đăng ký
  + Xác minh có còn slot hay không?
  + Nếu không còn slot thì gửi thông báo từ chối
  + Xác minh lịch học có trùng hay không?
  + Nếu lịch học trùng thì gửi thông báo từ chối và kết thúc chương trình

Trong số các hành động này , hành động đầu tiên được coi là ứng cử viên năng lực dịch vụ là tạo nên cơ sở của một ứng viên dịch vụ tác vụ được gọi là “Đăng ký môn học” . Các hành động còn lại được xác định là logic nội bộ trong dịch vụ tác vụ này :

Register Subject :

* Start

  Ứng cử viên Register Subject với một năng lực dịch vụ duy nhất Start .

5, Xác định tài nguyên:

Sau khi xem xét các yêu cầu xử lý của khả năng dịch vụ đã được xác định , các tài nguyên tiềm năng sau đây được xem xét :

/Register Subject Process/

/Class/

/SelectedClass/

/Student/

/Subject/

Thiết lập một số ánh xạ sơ bộ giữa các tài nguyên và thực thể kinh doanh được xác định

|  |  |
| --- | --- |
| **Entity** | **Resource** |
| Class | /Class/ |
| SelectedClass | /SelectedClass/ |
| Student | /Student/ |
| Subject | /Subject/ |

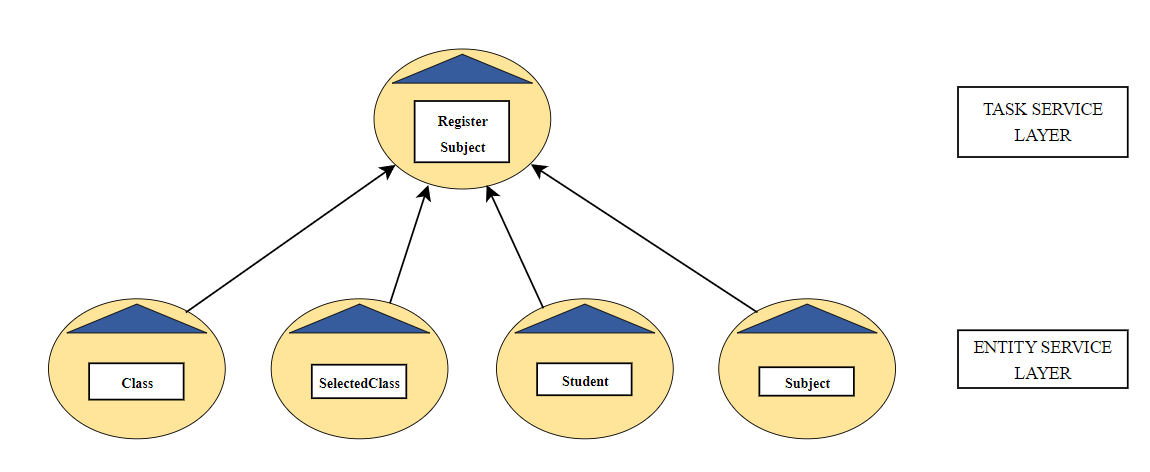
6, Liên kết năng lực dịch vụ với tài nguyên và các phương thức :

* Ứng viên dịch vụ Lớp học phần (Class):
* GetDetail(GET + /Class/)
* CheckCapacity(POST +/Class/)
* CheckDuplicate(POST + /Class/)
* Ứng viên dịch vụ Lớp học phần đã chọn (SelectedClass):
* SaveClass (POST + /SelectedClass/)
* DeleteClass (POST +/SelectedClass/)
* SaveSchedule(POST + /SelectedClass/)
* GetClass(GET +/SelectedClass/)
* Ứng viên dịch vụ Sinh viên (Student):
* GetDetail (GET + /Student/)
* Ứng viên dịch vụ Môn học (Subject):
* GetDetail (GET + /Subject/)
* (Tác vụ)Register Subject :
* Start (POST + /RegisterSubject/)

8, Xác định ứng viên tổ hợp dịch vụ

Kết quả phân tích các kịch bản ghép dịch vụ liên quan đến quá trình đăng ký môn học :

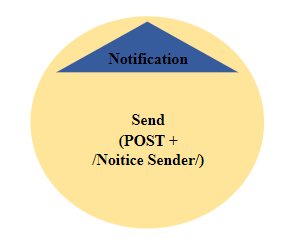
Dịch vụ tác vụ Register Subject triệu gọi các dịch vụ thực Class, Selected Class, Student, Subject đồng thời kết hợp với dịch vụ Announce để đưa ra thông báo chấp nhận hoặc từ chối.



9, Phân tích các yêu cầu xử lý:

* **Các chức năng mang tính chất tiện ích:**
* Gửi thông báo từ chối
* Gửi thông báo trùng lịch học
* Gửi thông báo sinh viên trong lớp đã đủ
* Gửi thông báo thành công
* **Các hành động có yêu cầu xử lý quan trọng và chuyên biệt:**
  + Xác minh có còn slot hay không?
  + Xác minh lịch học có trùng hay không?

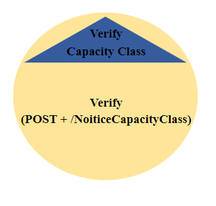
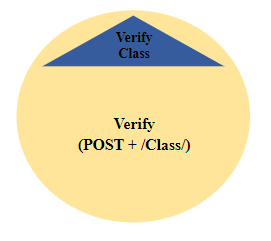
10, Xác định các ứng viên dịch vụ tiện ích (và các nguồn lực và phương pháp liên kết)



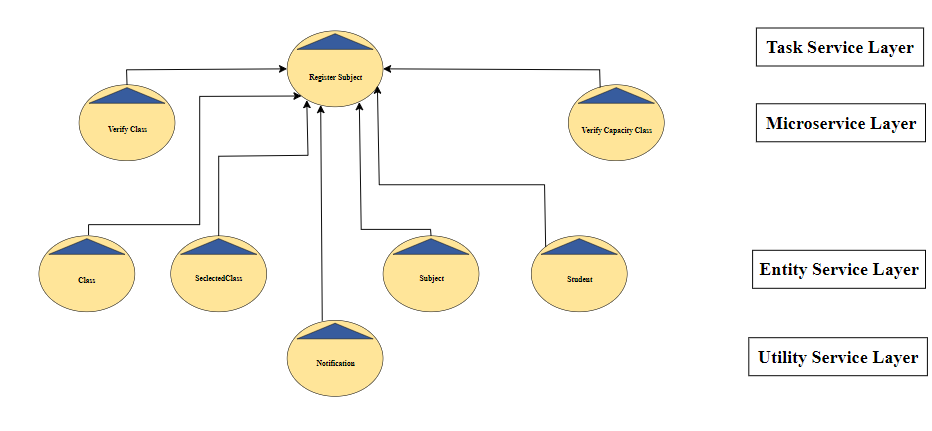
**Ứng viên dịch vụ thông báo**

11, Xác định các ứng viên microservice (liên kết tài nguyên và phương thức )

* **Để cô lập việc xử lý hành động “**Kiểm tra xem lớp học phần có còn slot hay không**” => đề xuất một ứng viên microservice có tên “Verify Capacity Class” và hành động “**Kiểm tra lớp học phần mới được chọn có lịch học trùng với lịch học của lớp học phần đã chọn**” => đề xuất một ứng viên có tên “Verify Class”.**

12, Cấu trúc phân cấp các ứng viên dịch vụ sau khi  kết hợp và mở rộng :



Biểu đồ tuần tự

