

Grade Portal Service Server

Nhóm lớp: Nhóm thầy Thuận

Giảng viên hướng dẫn: Lê Đình Thuận



Thành viên tham gia dự án

Họ và Tên	MSSV	Nhiệm vụ
Trần Đại Việt	2213951	PO
Phạm Việt Anh	2210128	DEV
Nguyễn Nhật Khoa	2211629	DEV
Nguyễn Gia Nguyên	2212303	DEV
Phạm Văn Quốc Việt	2213950	DEV
Lê Đăng Khoa	2211599	DEV

Nội dung trình bày

01. Giới thiệu về dự án

03. Phân tích yêu cầu
hệ thống

05. Hiện thực hệ thống

02. Lựa chọn quy trình
phát triển dự án

04. Thiết kế hệ thống
và phần mềm

06. Kết quả đạt được và
kế hoạch tiếp theo

Giới thiệu về dự án



Grade Portal Service là gì?

- Grade Portal là một hệ thống quản lý điểm số trực tuyến được phát triển nhằm hỗ trợ sinh viên và giảng viên tại Đại học Bách Khoa - Đại học Quốc gia TP.HCM trong việc tra cứu, quản lý và cập nhật kết quả học tập.
- Hệ thống này đáp ứng nhu cầu hiện đại hóa giáo dục, nâng cao tính minh bạch và chính xác trong việc công bố kết quả học tập, đồng thời cung cấp các tính năng như phân quyền và quản lý, cập nhật bảng điểm bằng các file CSV/Excel.

Mục tiêu của dự án

- Với Grade Portal, mục tiêu chính của dự án sẽ là mang lại lợi ích toàn diện như nhà trường hiện đại hóa quản lý, giảng viên giảm áp lực công việc và sinh viên có trải nghiệm học tập tối ưu hơn, từ đó góp phần nâng cao chất lượng giáo dục tại Đại học Bách Khoa - Đại học Quốc gia TP.HCM.
- Grade Portal đảm bảo điểm số được công bố chính xác, minh bạch nhưng vẫn giữ được tính bảo mật, tôn trọng quyền riêng tư về thông tin học tập, điểm số của sinh viên của sinh viên.
- Grade Portal còn đảm bảo việc phân quyền được rõ ràng chính xác, chỉ những đối tượng có quyền hạn nhất định mới thay đổi, cập nhật các dữ liệu nhất định trong hệ thống.

Stakeholders và các lợi ích dự án đem lại

Đối với Admin



- Hệ thống sẽ cung cấp cho Admin khả năng quản lý đa số đối tượng trong hệ thống và phân quyền một cách dễ dàng, đồng thời đảm bảo tính bảo mật trong việc quản lý thông tin của sinh viên và giảng viên.
- Đồng thời Admin trong hệ thống sẽ được đầy đủ các tính năng liên quan tới các việc cập nhật, chỉnh sửa dữ liệu trong dự án.
- Ngoài ra, chức năng Hall Of Fame sẽ cung cấp các dữ liệu thống kê chính xác về tình hình thực tiễn của nhà trường.

Stakeholders và các lợi ích dự án đem lại

Đối với giảng viên

- Giảng viên sẽ được hỗ trợ tối đa trong việc quản lý và cập nhật điểm số. Họ có thể dễ dàng tải lên và quản lý các bảng điểm từ file CSV, và hệ thống sẽ tự động cập nhật khi có thay đổi.
- Ngoài ra giảng viên được phân quyền rõ ràng từ phía Admin trong việc quản lý các đối tượng khác liên quan như lớp học và các bảng điểm tương ứng,...



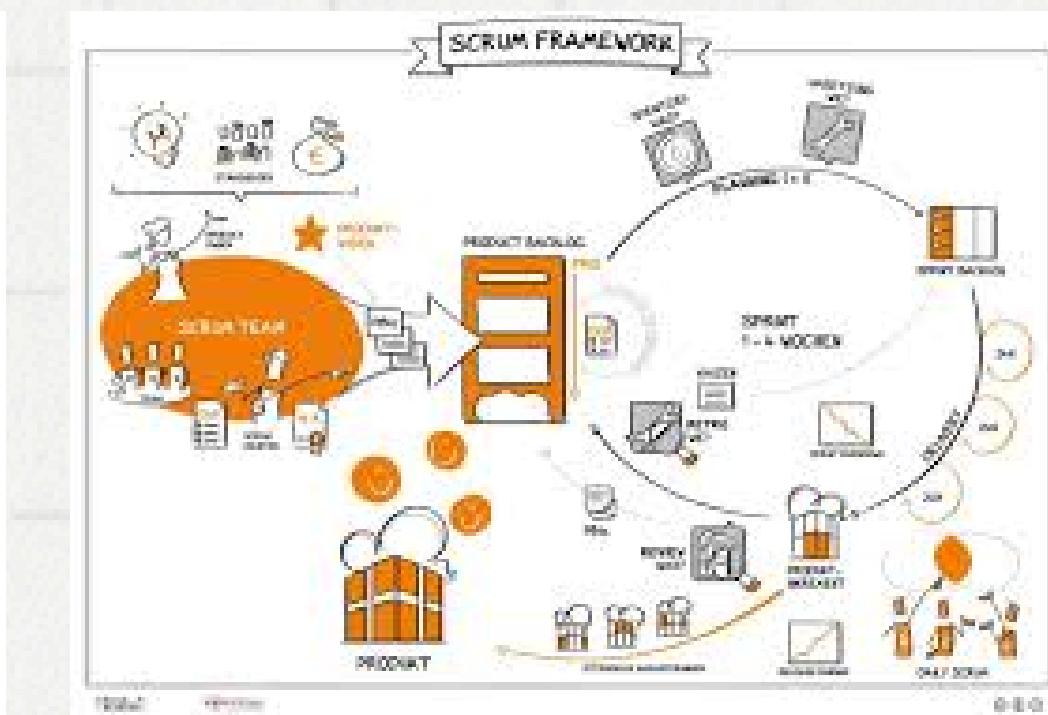
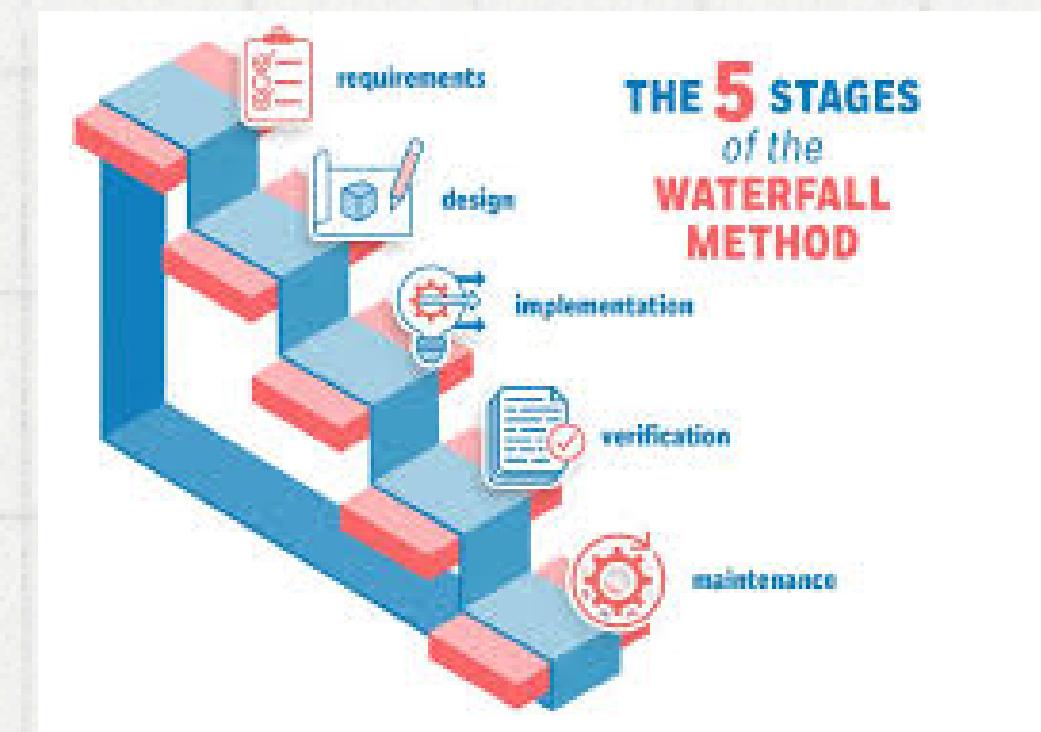
Stakeholders và các lợi ích dự án đem lại

Đối với sinh viên

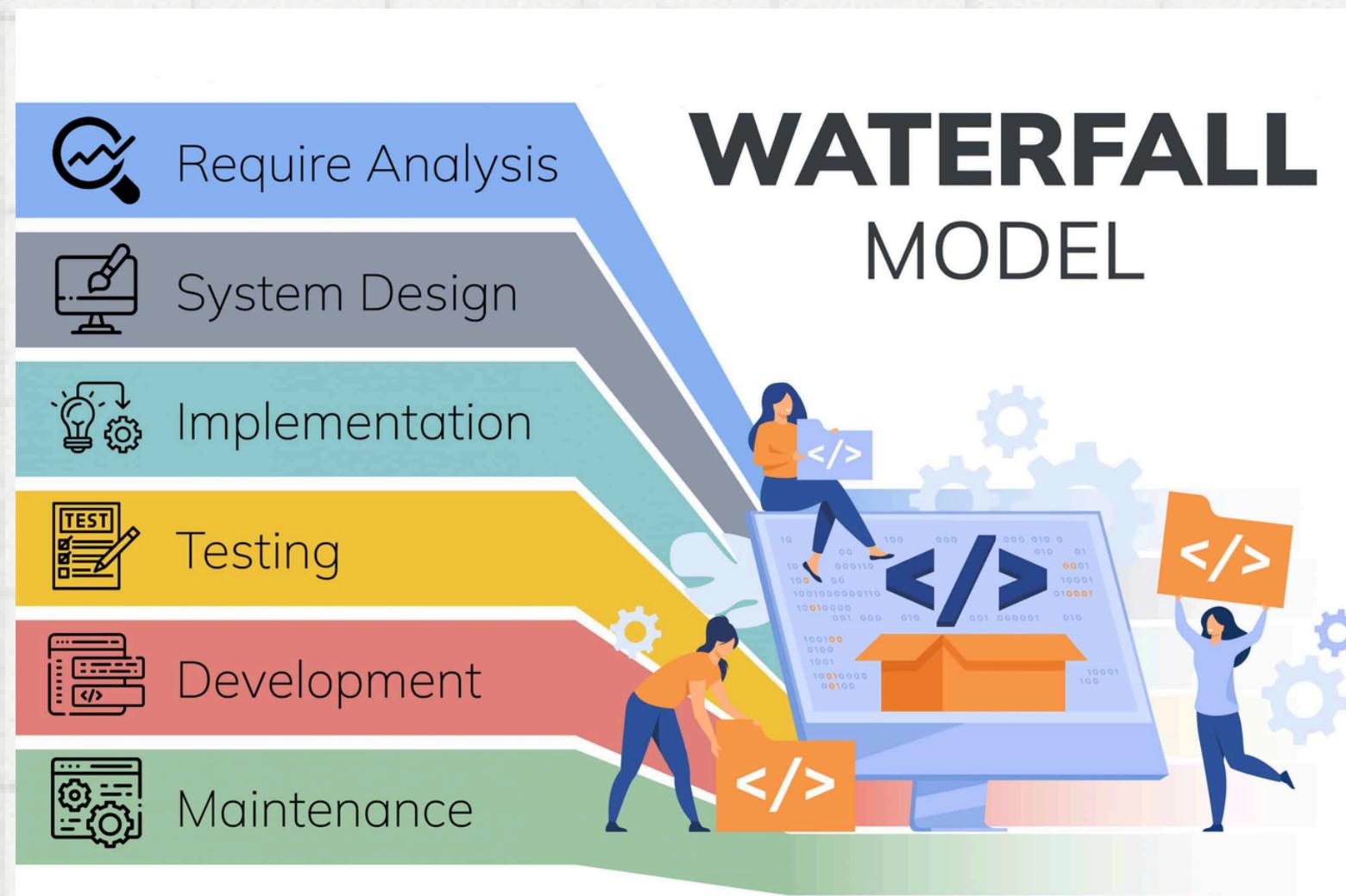


- Sinh viên sẽ có thể dễ dàng tra cứu điểm số một cách chính xác và cập nhật kịp thời trong quá trình học tập. Hệ thống giúp sinh viên theo dõi kết quả học tập của mình một cách minh bạch và rõ ràng, từ đó điều chỉnh chiến lược học tập phù hợp.
- Tính năng Hall of Fame sẽ tạo động lực cho sinh viên phấn đấu đạt thành tích cao.

Lựa chọn quy trình phát triển dự án



Quy trình Waterfall – Lựa chọn phù hợp cho dự án



Mô tả ngắn gọn về Waterfall:

- Quy trình phát triển phần mềm truyền thống, thực hiện tuần tự qua các giai đoạn: Yêu cầu → Thiết kế → Triển khai → Kiểm thử → Bàn giao.
- Mỗi giai đoạn được hoàn thành hoàn toàn trước khi chuyển sang giai đoạn tiếp theo.

Lý do chọn Waterfall:

- Phù hợp với các dự án có mục tiêu và yêu cầu rõ ràng từ đầu.
- Thích hợp cho bài tập lớn khi không cần thay đổi nhiều trong suốt quá trình thực hiện.

Lợi ích khi áp dụng Waterfall cho dự án

Yêu cầu ít thay đổi

- Mục tiêu bài tập đã được xác định rõ ràng ngay từ đầu (xây dựng hệ thống in ấn thông minh với các tính năng cụ thể).
- Không cần linh hoạt hoặc thay đổi liên tục như các dự án thực tế.

Thời gian và tài nguyên có giới hạn

- Bài tập lớn thường có khung thời gian cố định và nguồn lực hạn chế.
- Waterfall giúp nhóm tập trung hoàn thành từng giai đoạn một cách kỹ lưỡng và đảm bảo đúng tiến độ.

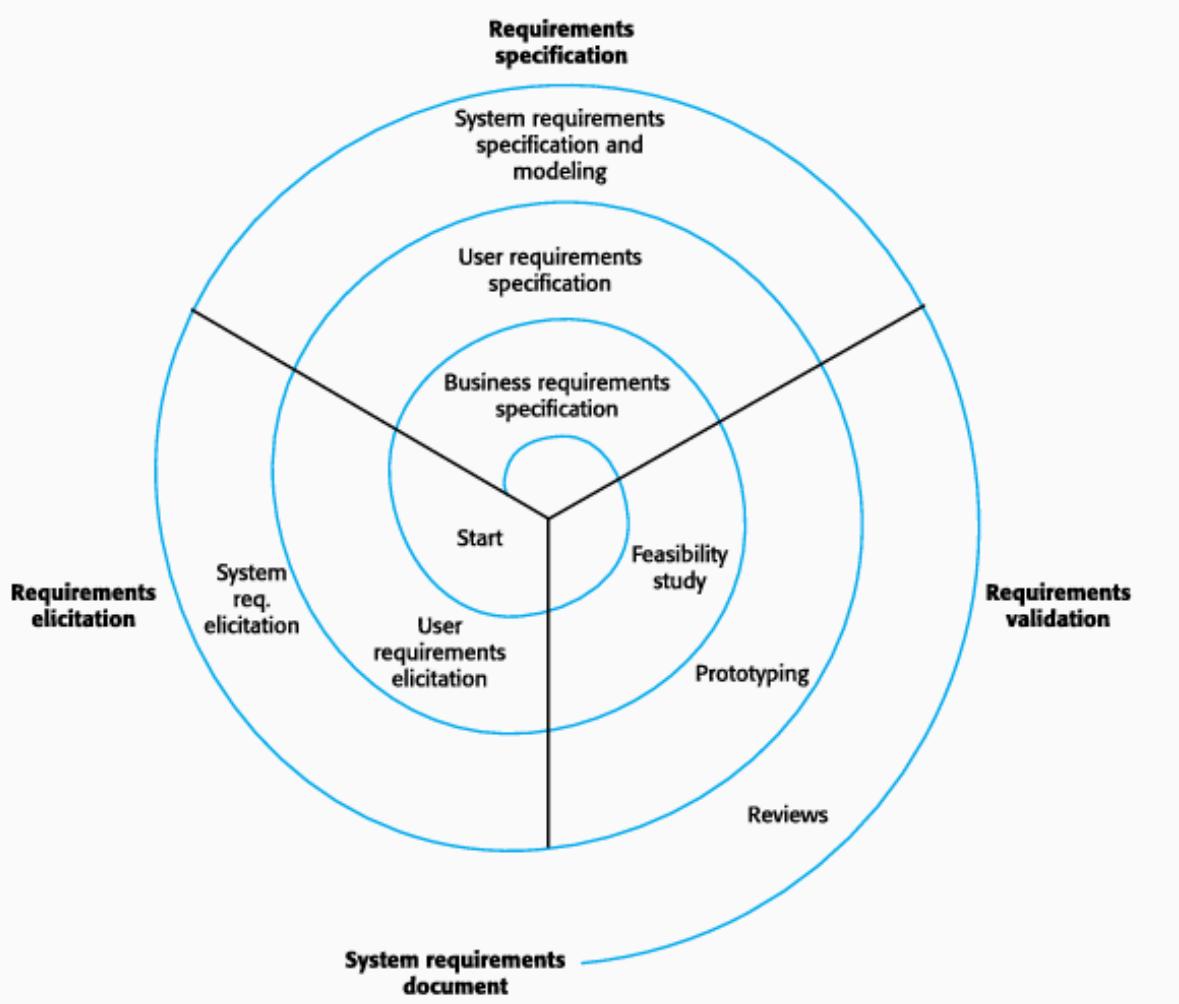
Tài liệu hóa đầy đủ

- Mô hình Waterfall khuyến khích tạo tài liệu chi tiết cho từng giai đoạn, hỗ trợ tốt cho yêu cầu nộp báo cáo cuối kỳ.

Thời gian và nội dung công việc

Phân tích yêu cầu hệ thống

A SPIRAL VIEW OF THE PROCESS



Yêu cầu chức năng

- Phải trải qua bước đăng nhập để xác thực quyền truy cập vào hệ thống thông qua Google Authentication.
- Có khả năng thêm mới và quản lý tài khoản giảng viên.
- Có khả năng tạo lớp học mới, gán sinh viên và giảng viên phụ trách vào lớp học.
- Có thể phân quyền cho giảng viên quản lý lớp học tương ứng.
- Có thể thêm hàng loạt sinh viên vào hệ thống bằng danh sách Gmail do trường cung cấp.
- Được cung cấp các hàm chức năng cho việc cập nhật và chỉnh sửa dữ liệu các đối tượng trong hệ thống.

Đối với Admin của hệ thống



Yêu cầu chức năng

Đối với giảng viên

- Phải trải qua bước đăng nhập để xác thực quyền truy cập vào hệ thống thông qua Gmail trường cấp.
- Có khả năng quản lý các lớp học được phân quyền bởi admin.
- Có thể cập nhật danh sách sinh viên vào lớp học mình quản lý.
- Có thể tải lên file CSV hoặc Excel chứa bảng điểm cho môn học.
- Hệ thống sẽ tự động giám sát và đồng bộ file CSV khi có phiên bản mới và cập nhật điểm vào cơ sở dữ liệu.



Yêu cầu chức năng

Đối với sinh viên, giảng viên



- Phải trải qua bước đăng nhập để xác thực quyền truy cập vào hệ thống thông qua Gmail trường cấp.
- Có thể tra cứu và xem điểm của mình cho tất cả các môn học đã tham gia.
- Có thể tìm kiếm điểm theo môn học, học kỳ, năm học hoặc MSSV.
- Có thể gửi phản hồi trực tiếp đến giảng viên phụ trách thông qua hệ thống (hoặc gmail được cung cấp).
- Có thể xem thông tin chi tiết về tình trạng lớp học đang tham gia và giảng viên phụ trách lớp đó.

Yêu cầu phi chức năng

Hiệu suất hoạt động (Performance)

- Hệ thống phải hỗ trợ nhiều người dùng đồng thời mà không bị giảm hiệu suất, đặc biệt là trong các thời điểm cao điểm như cuối học kỳ khi sinh viên tra cứu điểm.
- Hệ thống phải có khả năng phản hồi nhanh chóng, đặc biệt đối với các thao tác tải lên file CSV, Excel hoặc khi tra cứu thông tin điểm số.
- Hệ thống phải cung cấp phản hồi ngay lập tức về việc tải lên thành công hay thất bại của bảng điểm, cũng như khi sinh viên thực hiện tra cứu điểm.

Bảo mật (Security Requirements)

- Tất cả người dùng (Admin, giảng viên, sinh viên) đều phải xác thực bằng tài khoản Gmail trường cấp trước khi sử dụng hệ thống.
- Thông tin về điểm số, dữ liệu bảng điểm và các thông tin cá nhân liên quan đến sinh viên và giảng viên phải được bảo mật, chỉ những người có quyền truy cập mới có thể xem hoặc chỉnh sửa.
- Hệ thống phải đảm bảo an toàn cho dữ liệu CSV được tải lên và đồng bộ hóa, chỉ có giảng viên phụ trách lớp học và admin mới có quyền chỉnh sửa các bảng điểm này.

Yêu cầu phi chức năng

Toàn vẹn dữ liệu (Data Integrity)

- Mỗi yêu cầu tải lên bảng điểm và tra cứu điểm số phải ghi lại chính xác thông tin như ID sinh viên, môn học, thời gian tải lên, và điểm số tương ứng.
- Hệ thống phải đảm bảo rằng các dữ liệu bảng điểm không thể bị chỉnh sửa hoặc xóa bỏ bởi những người không có thẩm quyền.
- Mọi lịch sử và báo cáo liên quan đến việc tải lên và chỉnh sửa bảng điểm phải được duy trì toàn vẹn và bảo mật, đảm bảo rằng không có sự thay đổi dữ liệu trái phép.

Tính sẵn sàng (Availability)

- Hệ thống phải luôn sẵn sàng phục vụ giảng viên và sinh viên trong suốt thời gian học tập và giảng dạy (6h-22h mỗi ngày).
- Hệ thống phải được bảo trì định kỳ mà không làm gián đoạn hoạt động của người dùng. Các thông báo bảo trì phải được gửi trước để người dùng có thể sắp xếp công việc.
- Hệ thống phải có khả năng xử lý tình huống khẩn cấp, đảm bảo rằng dữ liệu điểm số không bị mất hoặc hư hỏng do lỗi hệ thống.

Thiết kế hệ thống và phần mềm



Thiết kế hệ thống và phần mềm

01. Usecase Diagram

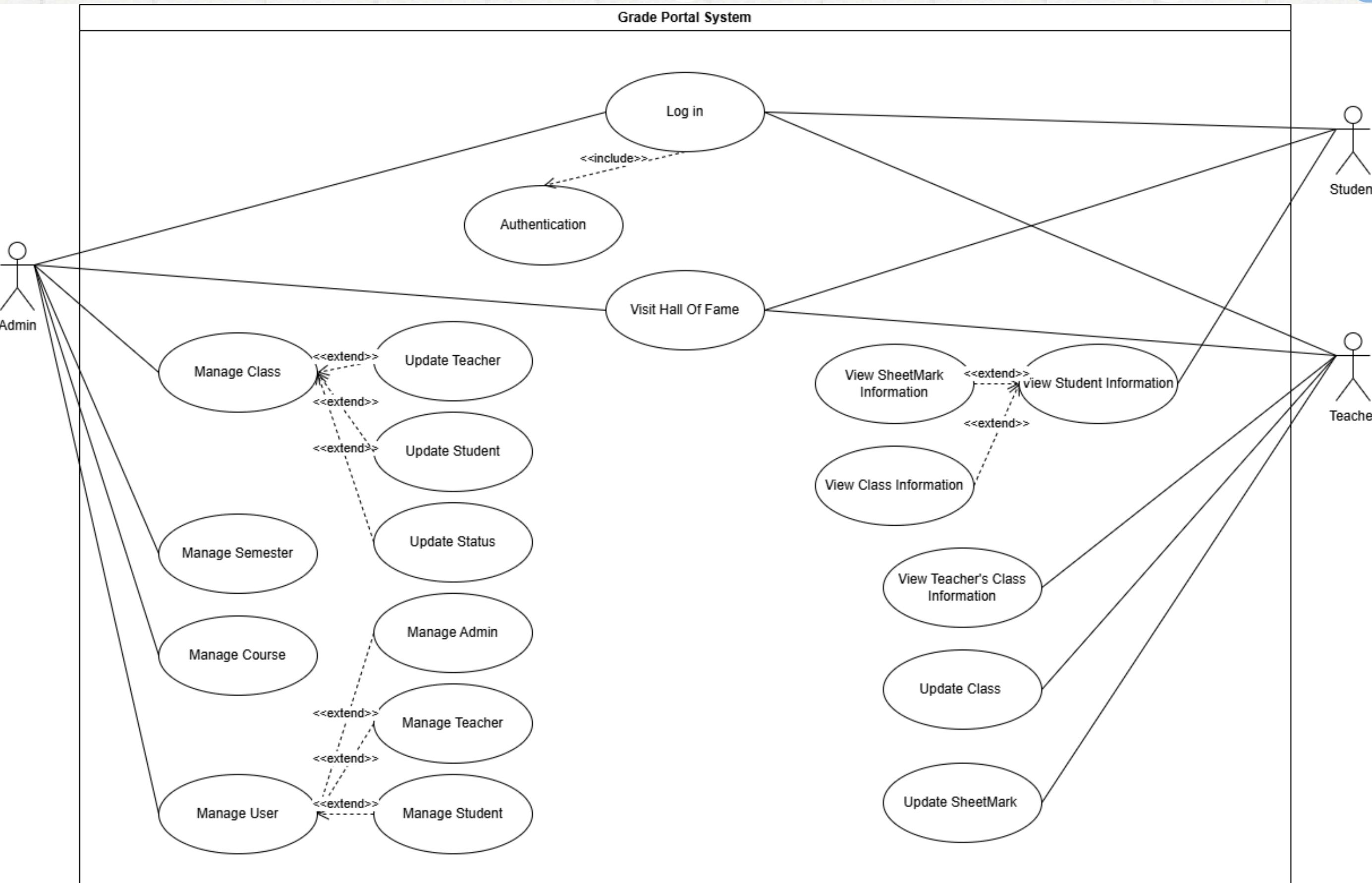
Biểu diễn sơ đồ chức năng của hệ thống. Mỗi use case mô tả chức năng mà hệ thống cần phải có xét từ góc độ người dùng.

Sự tương tác trong biểu đồ use case có thể là:

- Cách thức mà người dùng tương tác với hệ thống
- Cách thức mà hệ thống tương tác với các hệ thống khác
- Làm cơ sở cho việc phát triển, kiểm tra các bản vẽ như Class Diagram, Activity Diagram, Sequence Diagram, Component Diagram



Sơ đồ chức năng toàn hệ thống



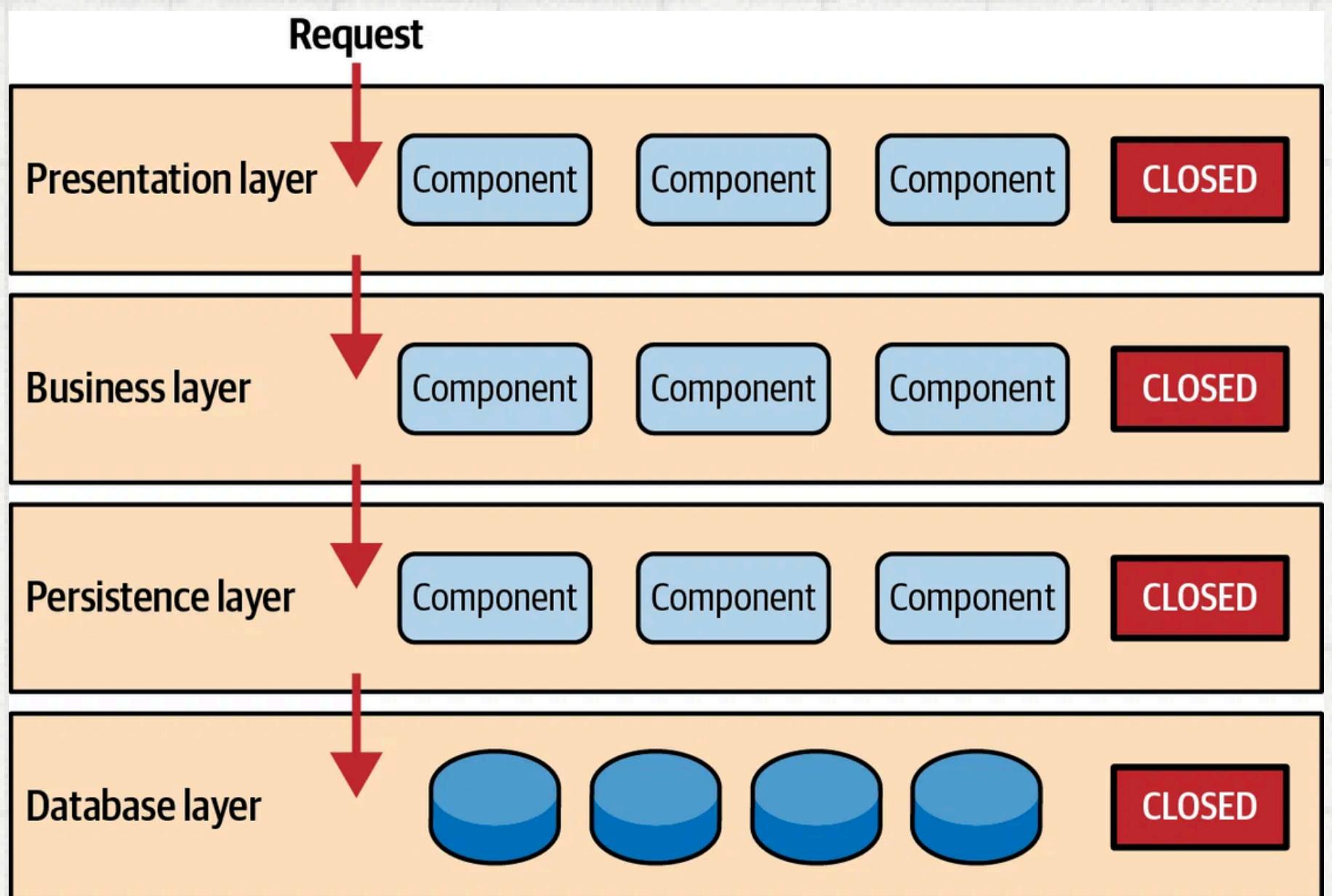
Thiết kế hệ thống và phần mềm

02. System Architecture

- Khi phát triển một hệ thống, việc xác định kiến trúc hệ thống là yếu tố vô cùng quan trọng.
- Kiến trúc hệ thống đóng vai trò là "bộ khung" tổng thể, giúp định hình cách các thành phần trong hệ thống tương tác với nhau và đảm bảo hệ thống vận hành hiệu quả, an toàn và dễ bảo trì.

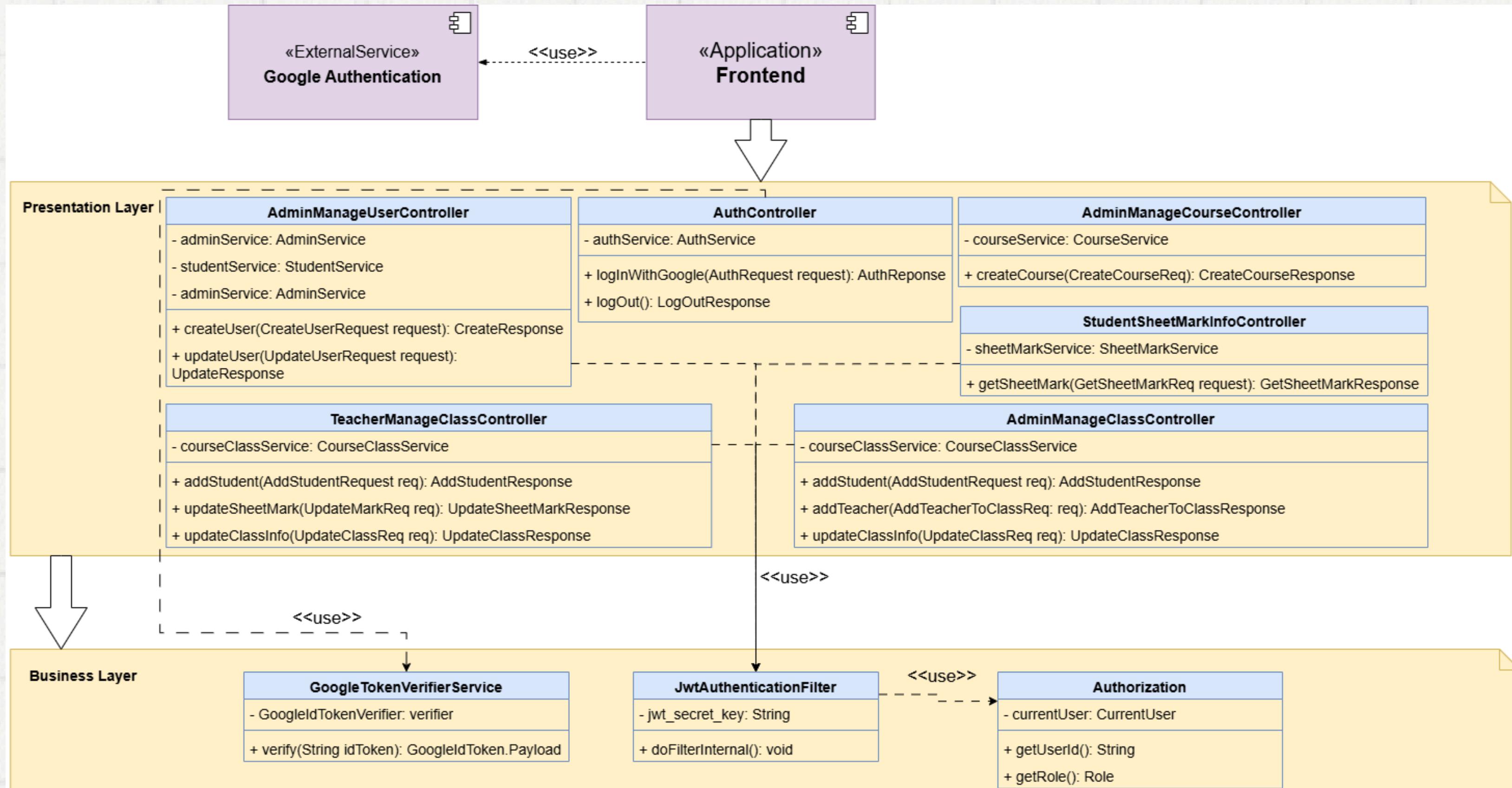


Layered Architecture

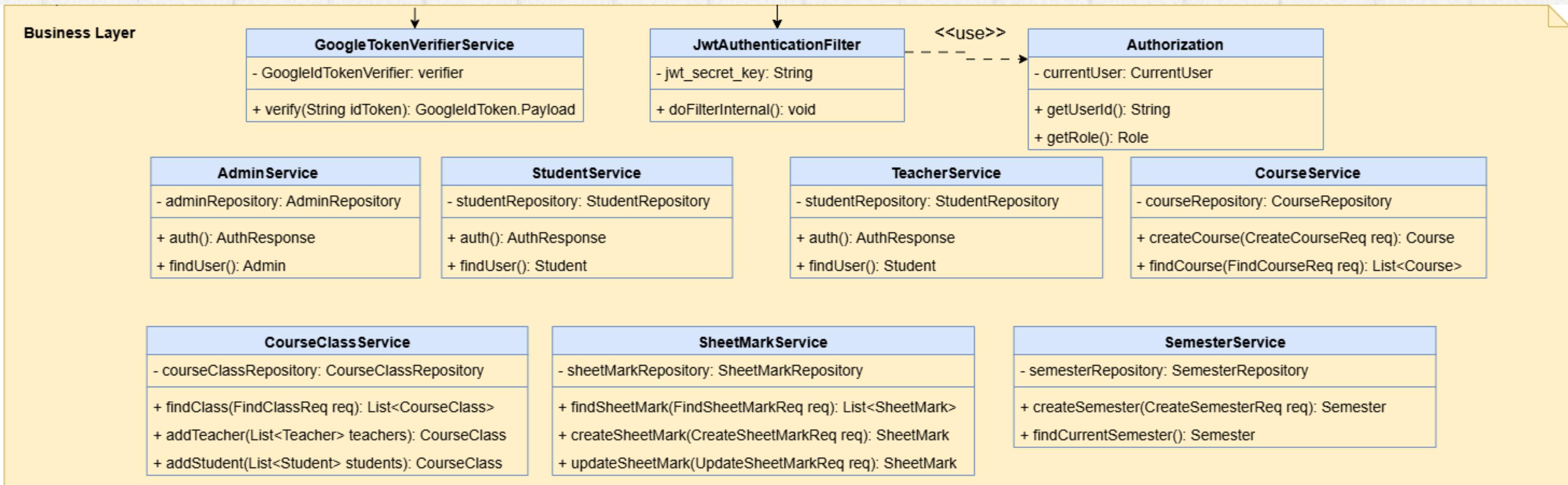


- Mặc dù đây là một kiến trúc Monolithic nên có một số hạn chế, nhưng trong trường hợp của dự án, đây vẫn là lựa chọn khả thi cho giai đoạn phát triển của hệ thống.
- Tuy nhiên, vẫn có lý do để chọn kiến trúc Monolithic ở giai đoạn này. Đầu tiên, kiến trúc monolithic dễ triển khai và phát triển nhanh hơn, đặc biệt trong giai đoạn khởi đầu của dự án. Với một đội ngũ nhỏ và khi chưa có nhiều yêu cầu phức tạp về khả năng mở rộng, việc tập trung vào một ứng dụng duy nhất giúp tiết kiệm thời gian phát triển, đơn giản hóa việc quản lý code và dễ dàng kiểm thử hệ thống.
- Đây là architecture pattern n-tier. Tất cả logic điều được move lên phía server, do đó giải quyết triệt để vấn đề về mở rộng, client bây giờ chỉ làm nhiệm vụ render mà không cần biết các thay đổi từ server ra sao.

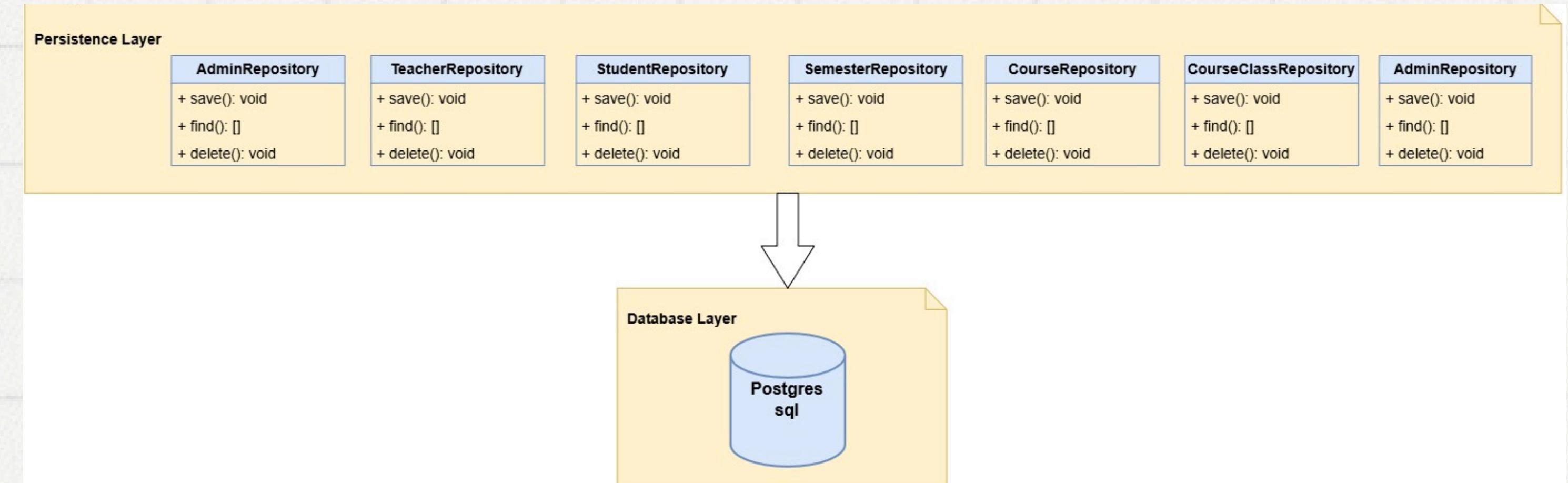
Presentation Layer



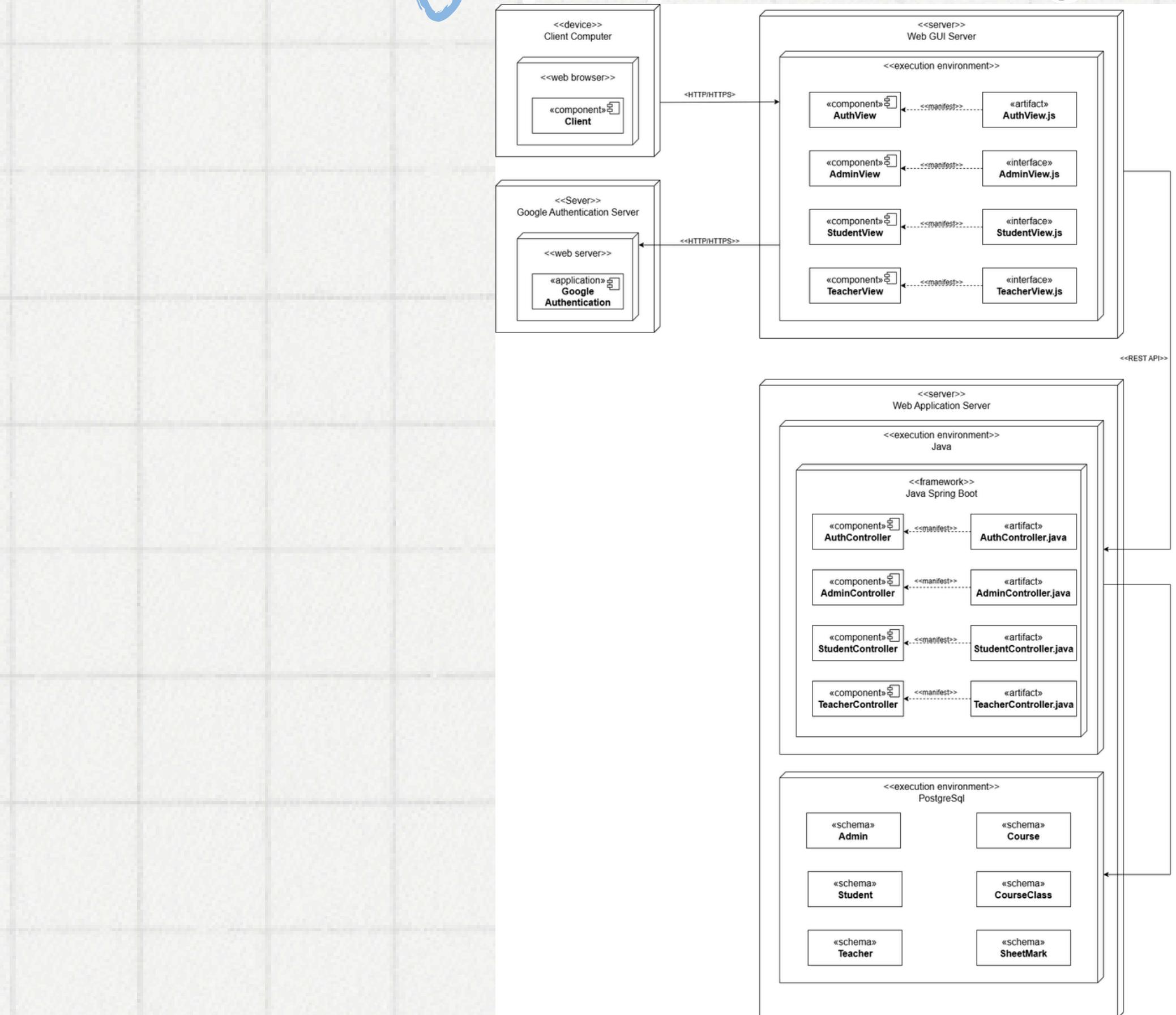
Business Layer



Persistence And Data Layer



Deployment Diagram



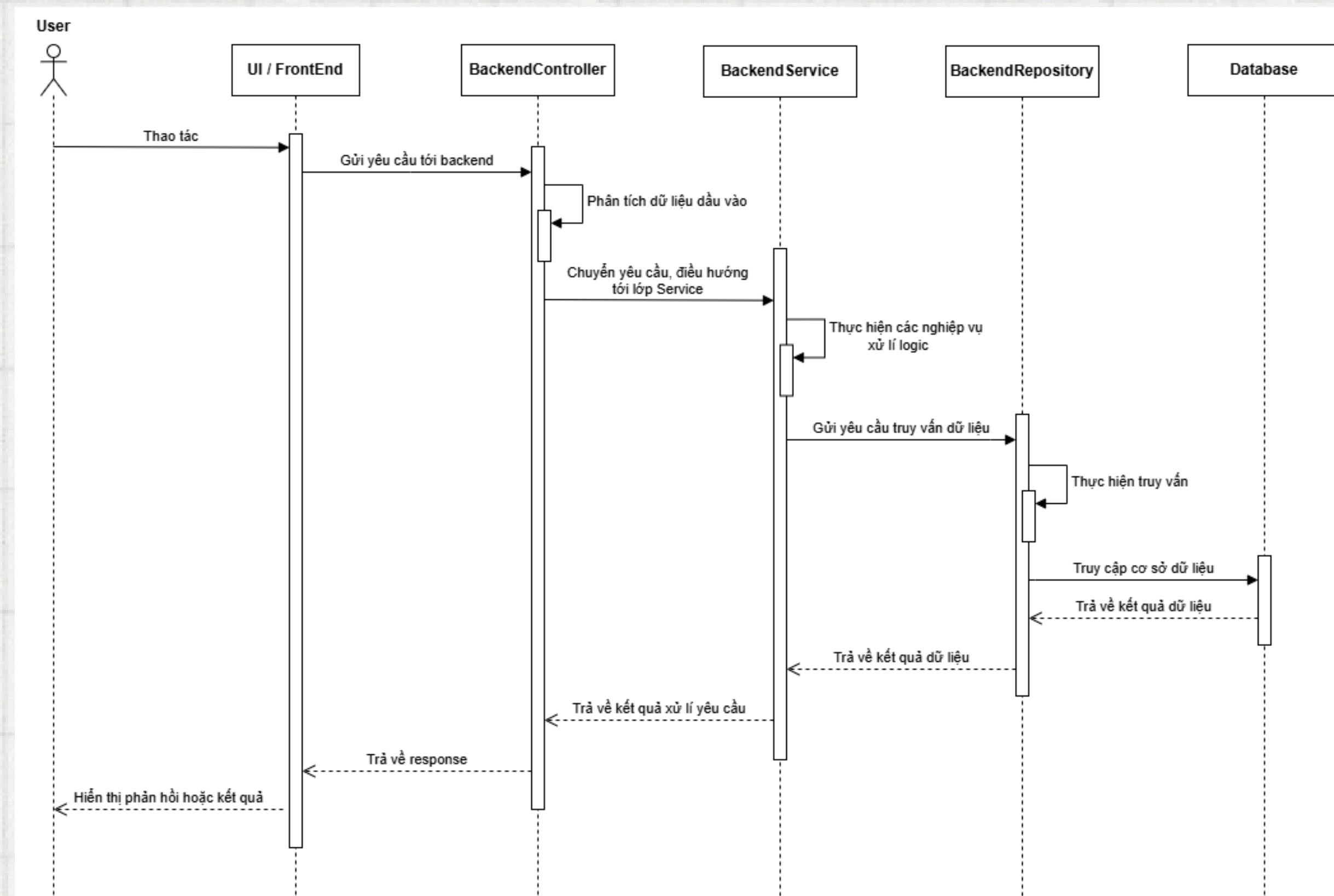
Thiết kế hệ thống và phần mềm

03. Activity and Sequence Diagram

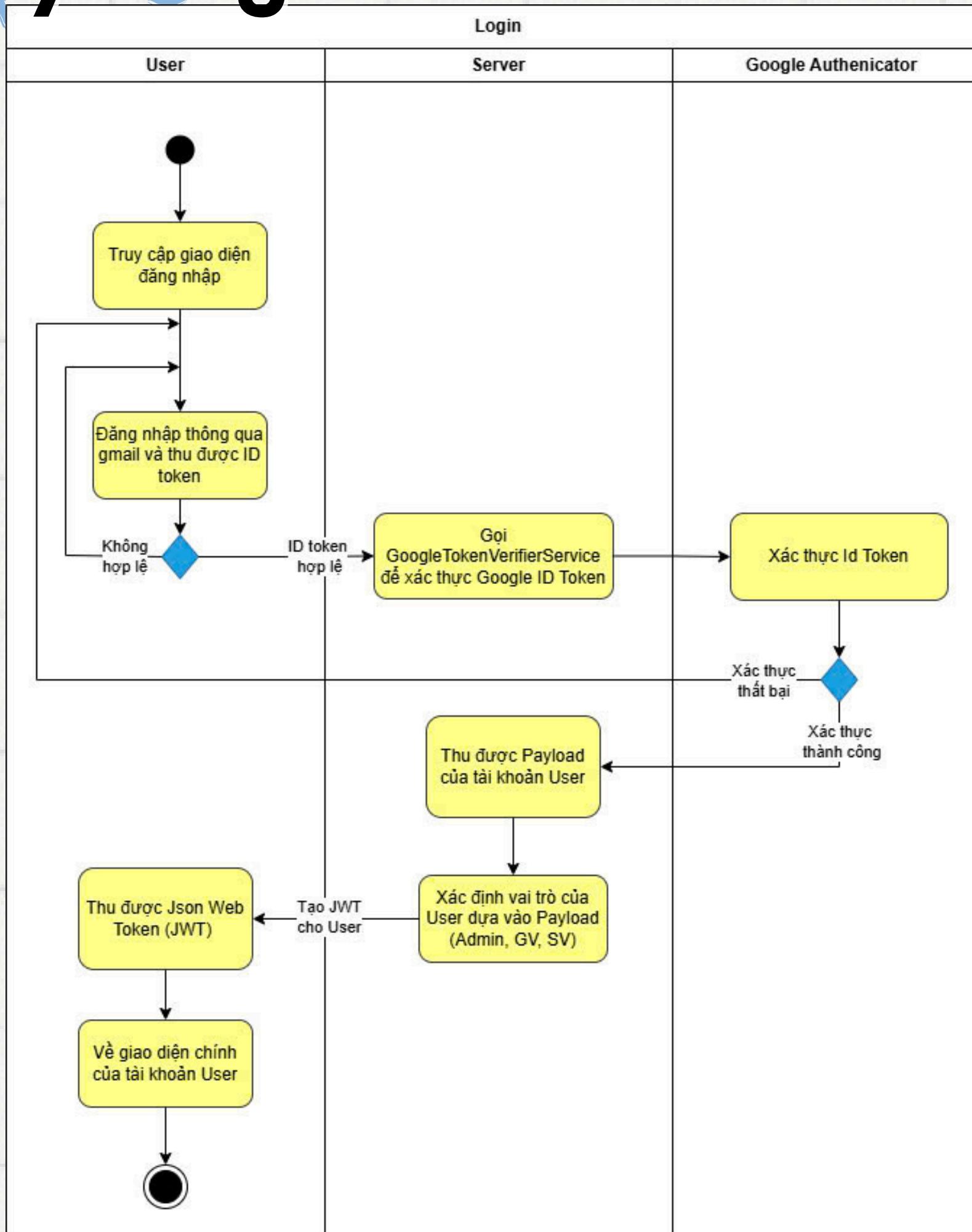
- Activity Diagram là một loại sơ đồ thuộc nhóm Behavioral Diagrams trong UML (Unified Modeling Language), được sử dụng để biểu diễn luồng công việc (workflow) hoặc quy trình nghiệp vụ (business process) trong hệ thống.
- Sequence Diagram là một loại sơ đồ trong UML dùng để biểu diễn luồng tương tác (interaction flow) giữa các đối tượng trong hệ thống theo thứ tự thời gian, tập trung vào cách các thành phần (actors, hệ thống con, hoặc đối tượng) giao tiếp với nhau để hoàn thành một chức năng cụ thể.



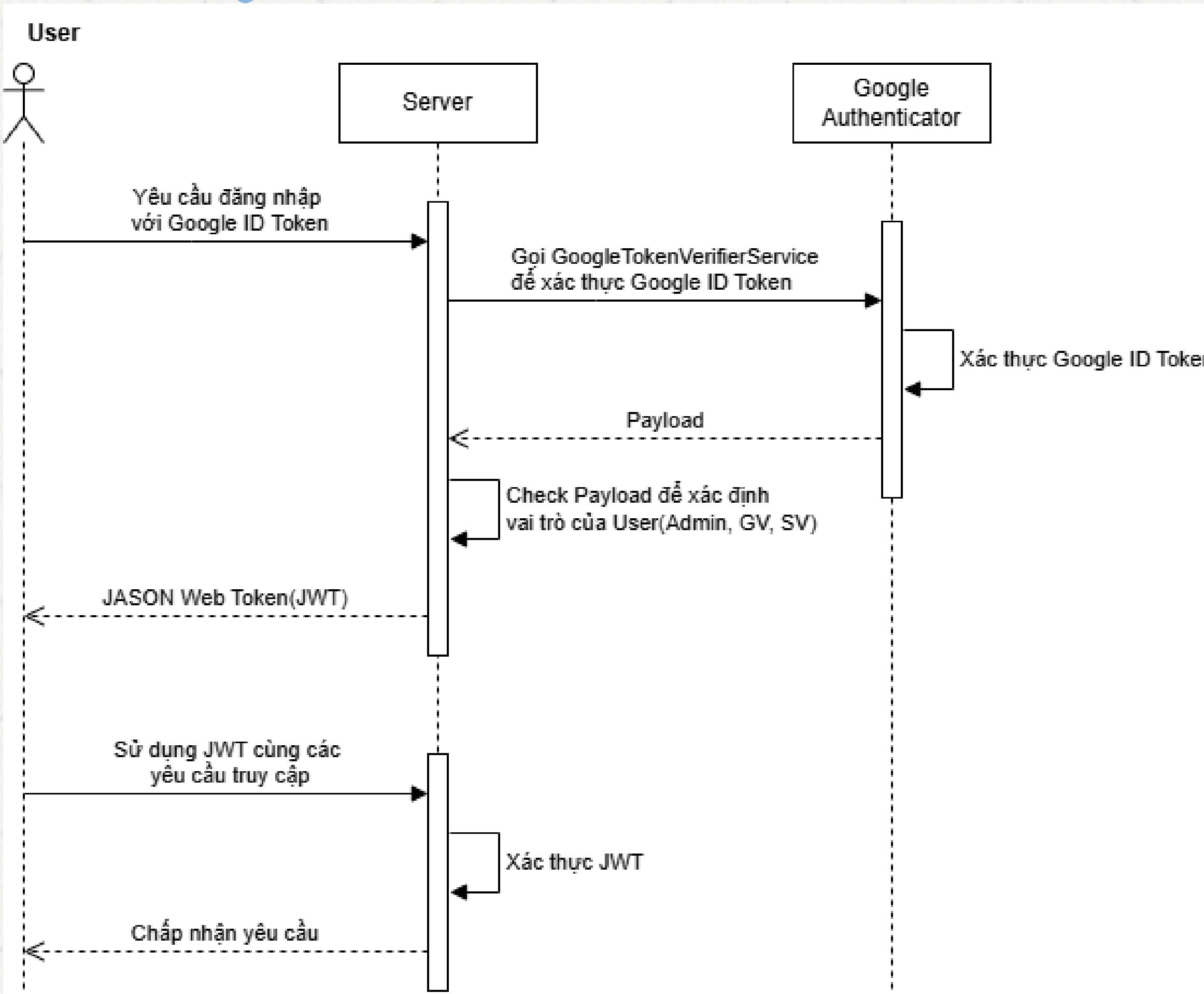
Sequence Diagram - The whole system



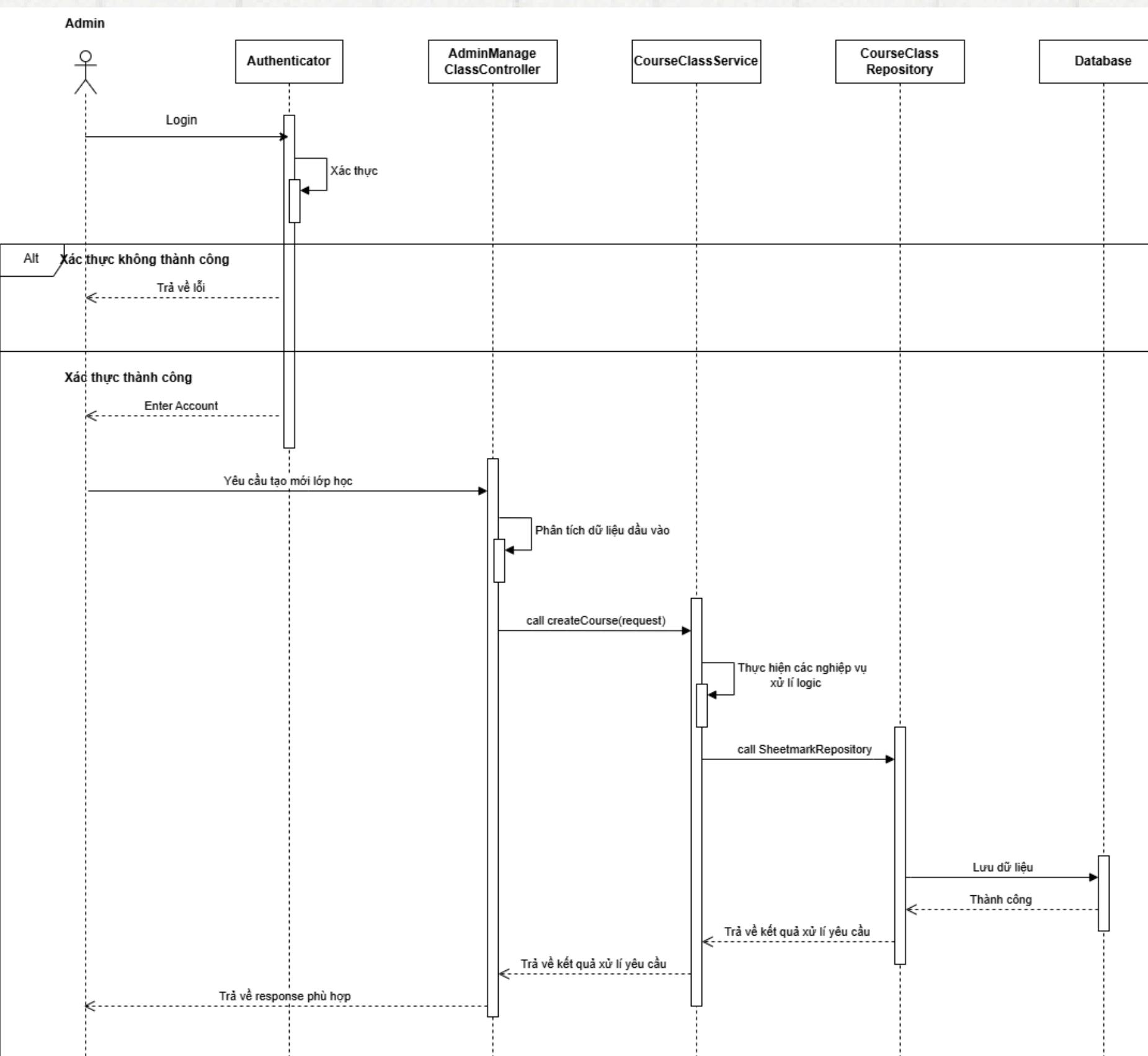
Activity Diagram - Authentication



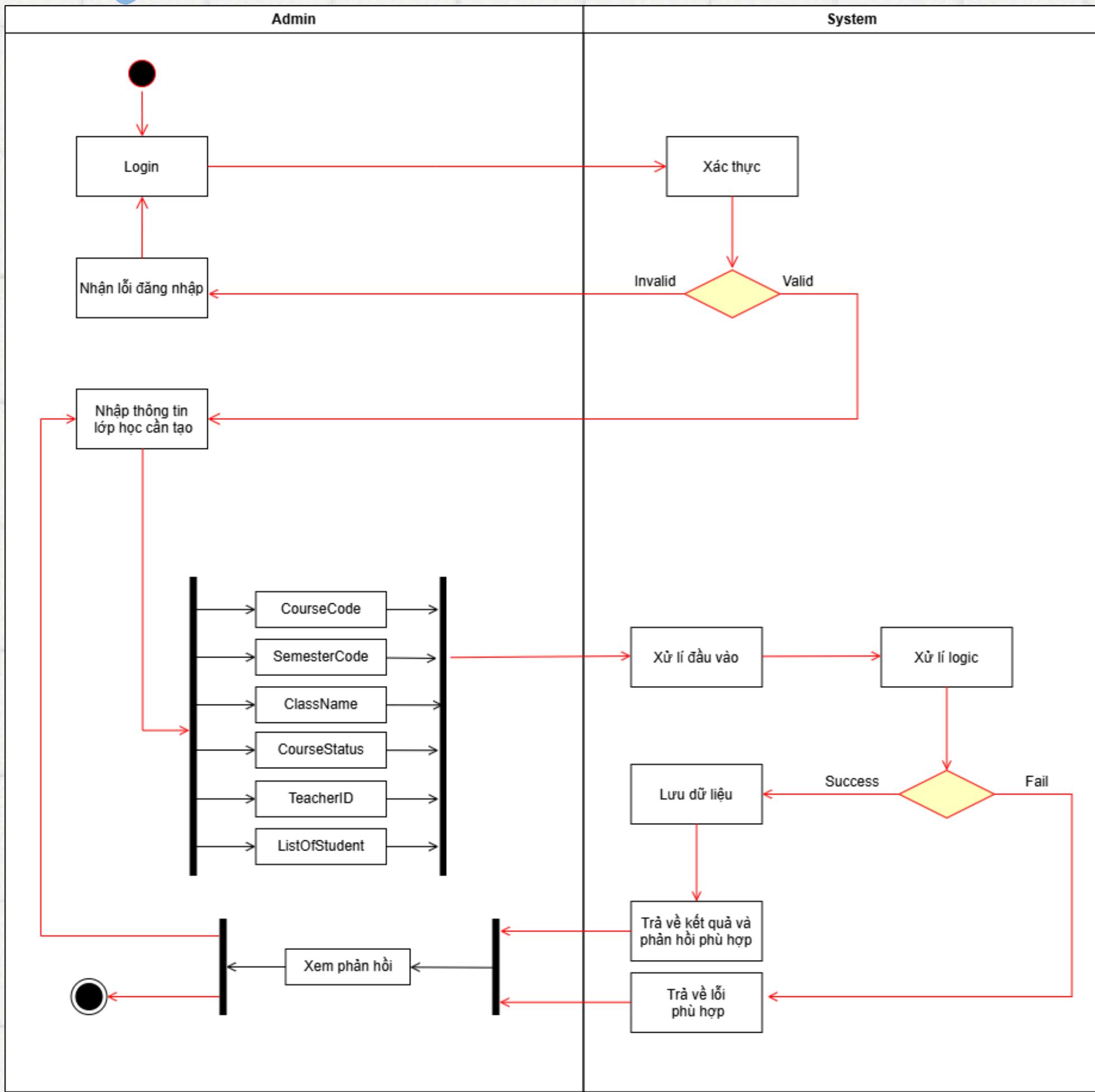
Sequence Diagram - Authentication



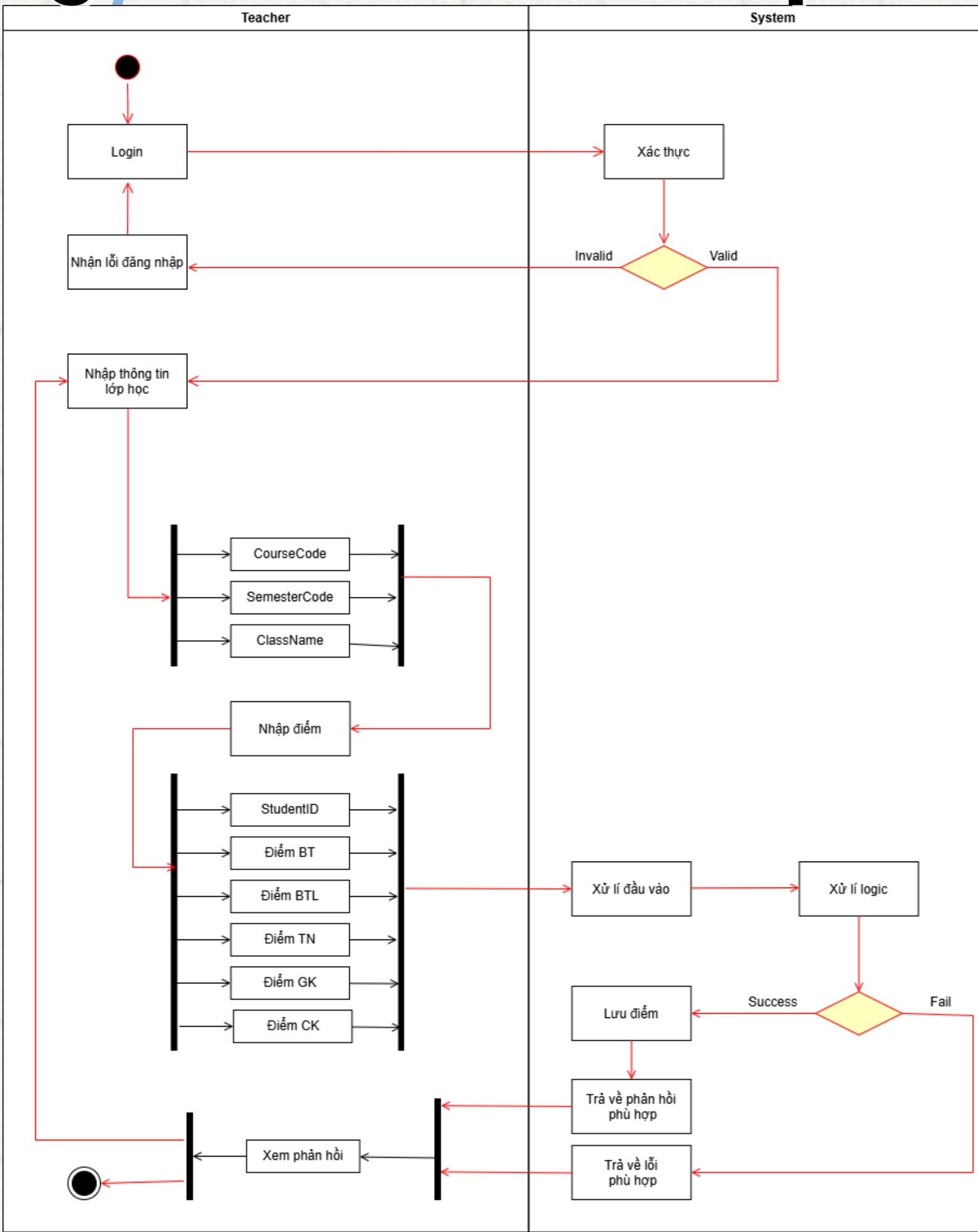
Sequence Diagram - Admin Create Class



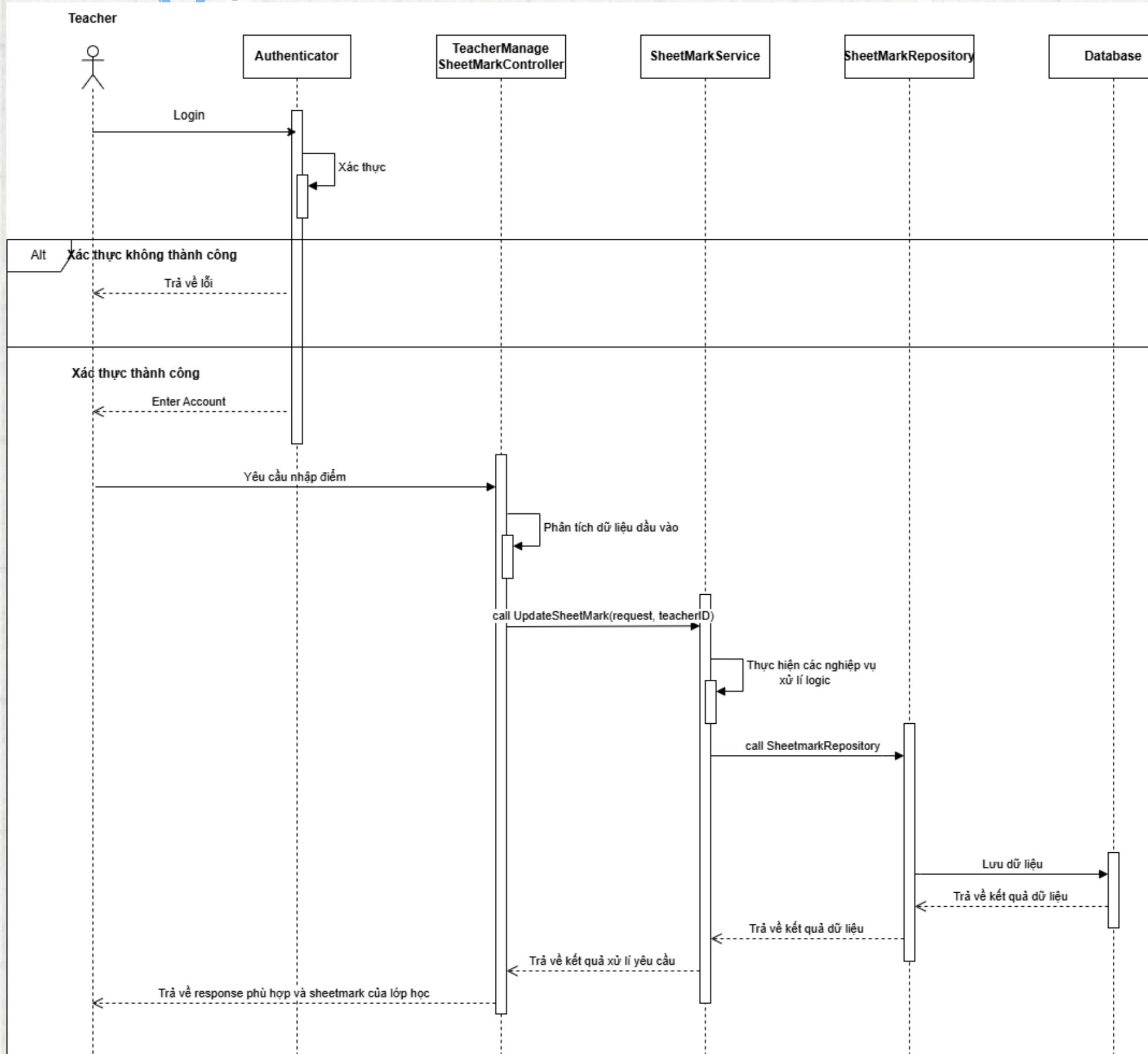
Activity Diagram - Admin Create Class



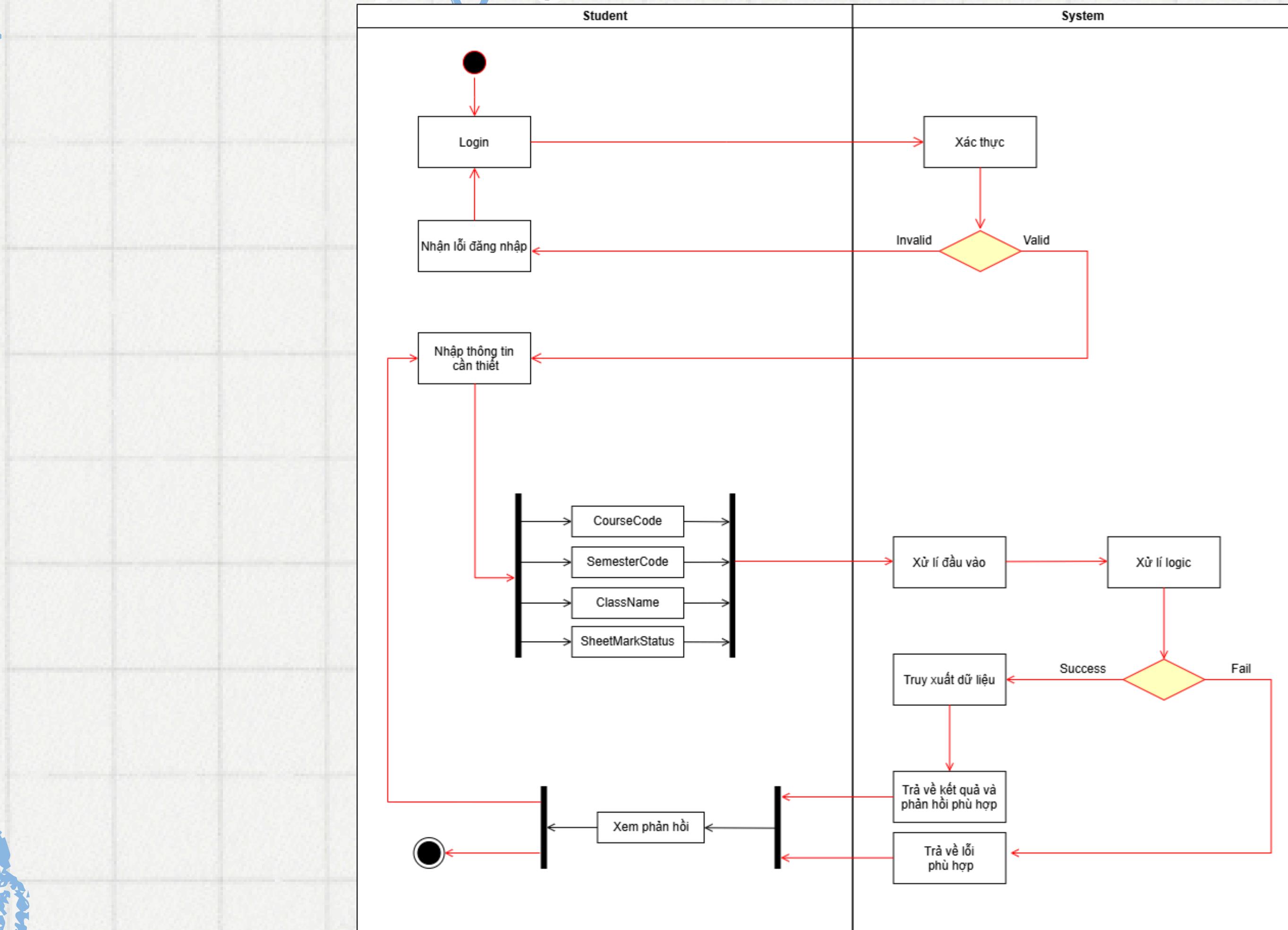
Activity Diagram - Teacher Update Sheetmark



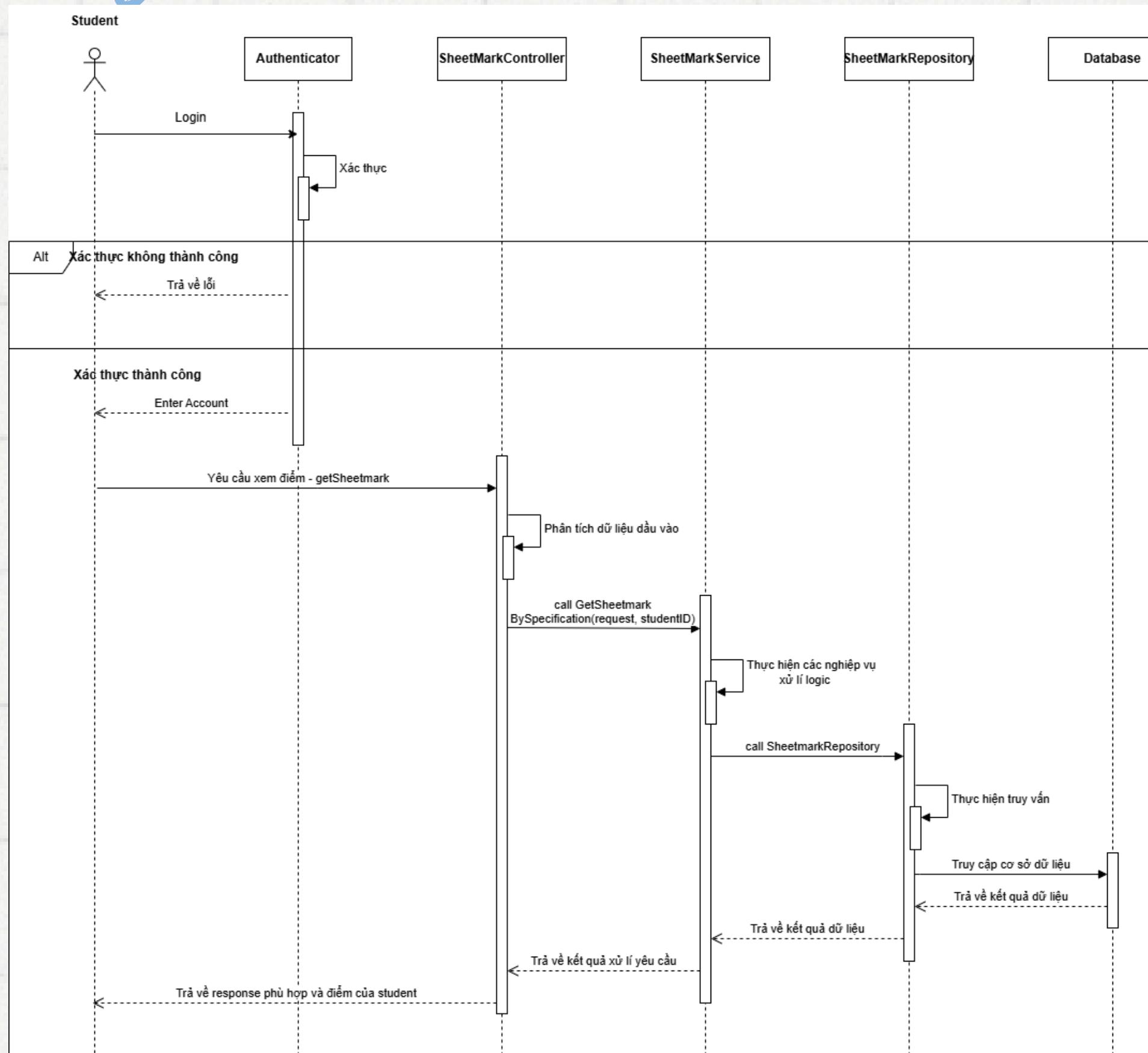
Sequence Diagram - Teacher Update Sheetmark



Activity Diagram – Student Get Sheetmark



Sequence Diagram - Student Get Sheetmark



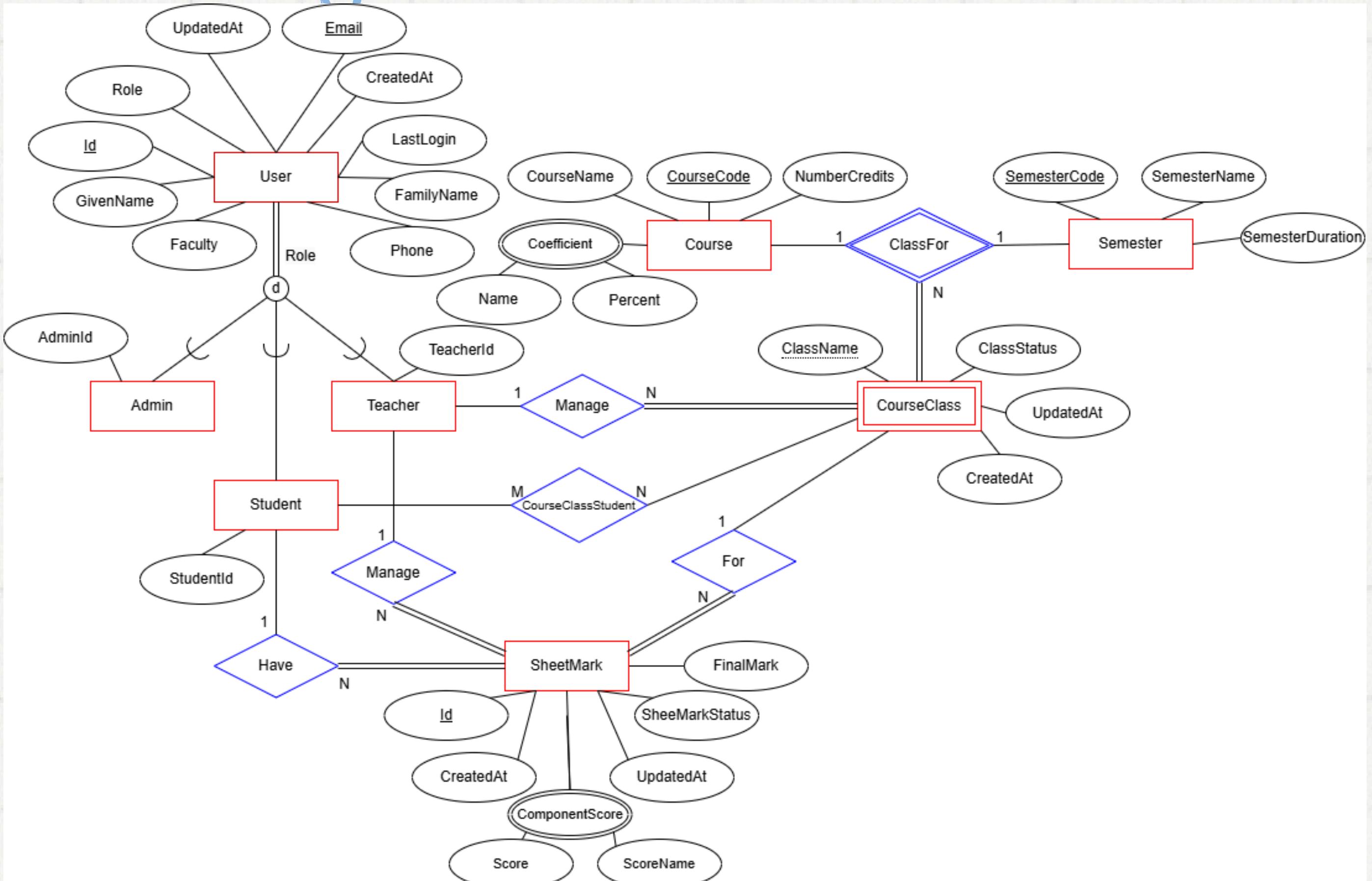
Thiết kế hệ thống và phần mềm

04. Database approach

- Đối với các yêu cầu của hệ thống , không khó để nhận ra, các hành động liên quan tới việc truy xuất, xử lý và cập nhập dữ liệu là rất quan trọng và cần sự chính xác cao. Và đó cũng chính là một trong những lí do kiến trúc phân tầng được chọn để hiện thực hệ thống.
- Nhóm chúng em sẽ sử dụng cơ sở dữ liệu quan hệ Postgres để lưu trữ dữ liệu.



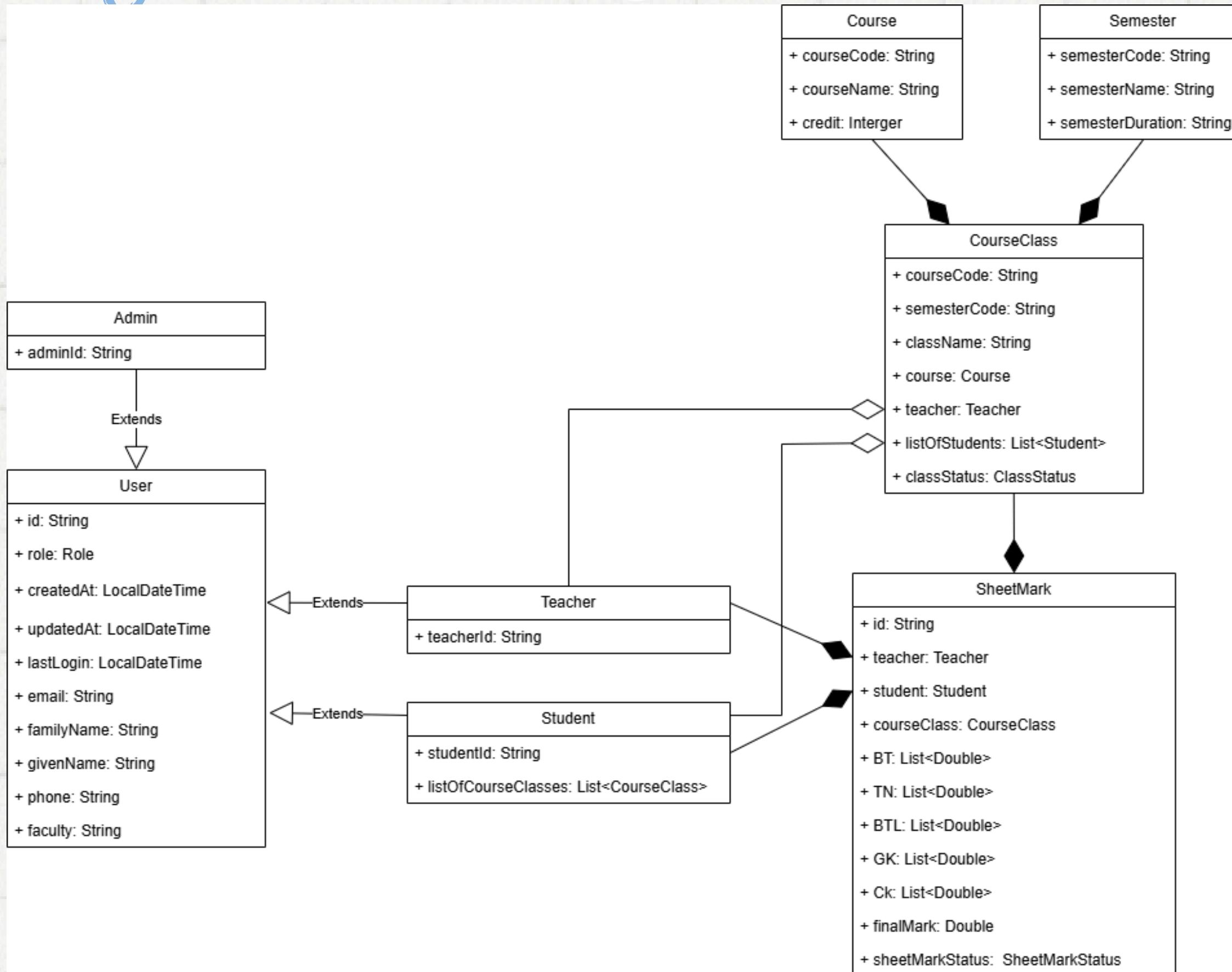
EERD



Relational Data Model



Class Diagram





Hiện thực hệ thống

Tổ chức và quản lý Code trong mã nguồn

```
|--- .mvn  
|   |--- wrapper      % Cấu hình Maven Wrapper  
|--- .vscode        % Cấu hình cho VSCode  
|--- docs          % Thư mục chứa tài liệu dự án  
|   |--- api-document    % Tài liệu API  
|   |--- database-design  % Thiết kế CSDL  
|   |--- deploy-guide     % Hướng dẫn triển khai  
|   |--- git-flow         % Quy trình làm việc với Git  
|   |--- system-architecture% Kiến trúc hệ thống  
|--- reports        % Báo cáo dự án  
|   |--- mainReport      % Báo cáo chính  
|   |--- weeklyReport    % Báo cáo hàng tuần  
|       |--- meeting_minute_report % Biên bản họp  
|       |--- weekly_result    % Kết quả hàng tuần  
|           |--- img_10_27_24 % Hình ảnh minh họa  
|--- src            % Mã nguồn chính  
|   |--- main  
|   |   |--- java  
|   |   |   |--- com  
|   |   |   |--- hcmut  
|       |--- gradeportal  
|           |--- config      % Cấu hình hệ thống  
|           |--- controller   % Các controller xử lý API  
|               |--- admin  
|                   |--- client_auth  
|                   |--- hall_of_fame  
|                   |--- health_check  
|                   |--- init_data  
|                   |--- student  
|                   |--- teacher  
|           |--- database    % Thao tác cơ sở dữ liệu
```

Tổ chức và quản lý Code trong mã nguồn

```
|   |   |   |--- dtos          % Data Transfer Objects
|   |   |   |   |--- admin
|   |   |   |   |   |--- request
|   |   |   |   |   |--- response
|   |   |   |   |--- auth
|   |   |   |   |--- course
|   |   |   |   |--- sheetMark
|   |   |   |   |--- teacher
|   |   |--- entities      % Định nghĩa các entity
|   |   |   |   |--- enums
|   |   |   |   |--- idsOfEntities
|   |   |--- repositories  % Repository làm việc với CSDL
|   |   |--- security      % Bảo mật JWT
|   |   |--- service        % Logic nghiệp vụ
|   |   |--- specification  % Query Specification
|   |   |--- validation      % Validation dữ liệu
|   |--- resources
|   |   |--- baseData       % Dữ liệu nền
|   |   |--- static          % File tĩnh
|   |   |--- templates       % Template giao diện
|--- test                         % Mã nguồn test
|--- target                        % Thư mục build của Maven
|--- pom.xml                       % File cấu hình Maven
```

Code Style

- Mã nguồn được tổ chức theo các quy tắc chuẩn như sắp xếp import theo nhóm (Java chuẩn, thư viện bên thứ ba, module nội bộ) và loại bỏ import dư thừa.
- Tên biến sử dụng camelCase, class dùng PascalCase, và hằng số viết in hoa có gạch dưới (ví dụ: MAX_CONNECTIONS).
- Sử dụng các thư viện hỗ trợ để format mã được thống nhất.
- Phân chia trách nhiệm rõ ràng giữa repository và service theo nguyên tắc SOLID. CRUD được tách biệt logic, comment mô tả ngắn gọn từng bước xử lý, giúp mã nguồn dễ đọc, bảo trì và sẵn sàng mở rộng trong tương lai.



Quản lý phiên bản (Version Control)



grade-portal-be-java Public

main 17 Branches 0 Tags

VietTranDai Merge pull request #59 from dath-241/staging 8fd3060 - 3 hours ago 94 Commits

File	Description	Time
.mvn/wrapper	first commit	2 months ago
docs	update document and report	last month
reports	update document and report	last month
src	update api	3 hours ago
.dockerignore	merge staging	3 weeks ago
.gitattributes	config docker	2 weeks ago
.gitignore	config docker	3 weeks ago
.sops.yaml	first commit	2 months ago
Dockerfile	config docker	3 weeks ago
README.md	end feature/01-get-post-request-admin-manage-semester	2 months ago
dev.enc	update file dev.enc	last week
docker-compose.yml	config docker	2 weeks ago
example.env	first commit	2 months ago
hcmut.png	first commit	2 months ago
pom.xml	fix composite key for course class and update api for init dat...	3 days ago
prod.enc	update file dev.enc	last week

Code About

a2-grade.thuanle.me/

Readme Activity Custom properties 0 stars 1 watching 0 forks Report repository

Releases No releases published Create a new release

Packages No packages published Publish your first package

Contributors 5

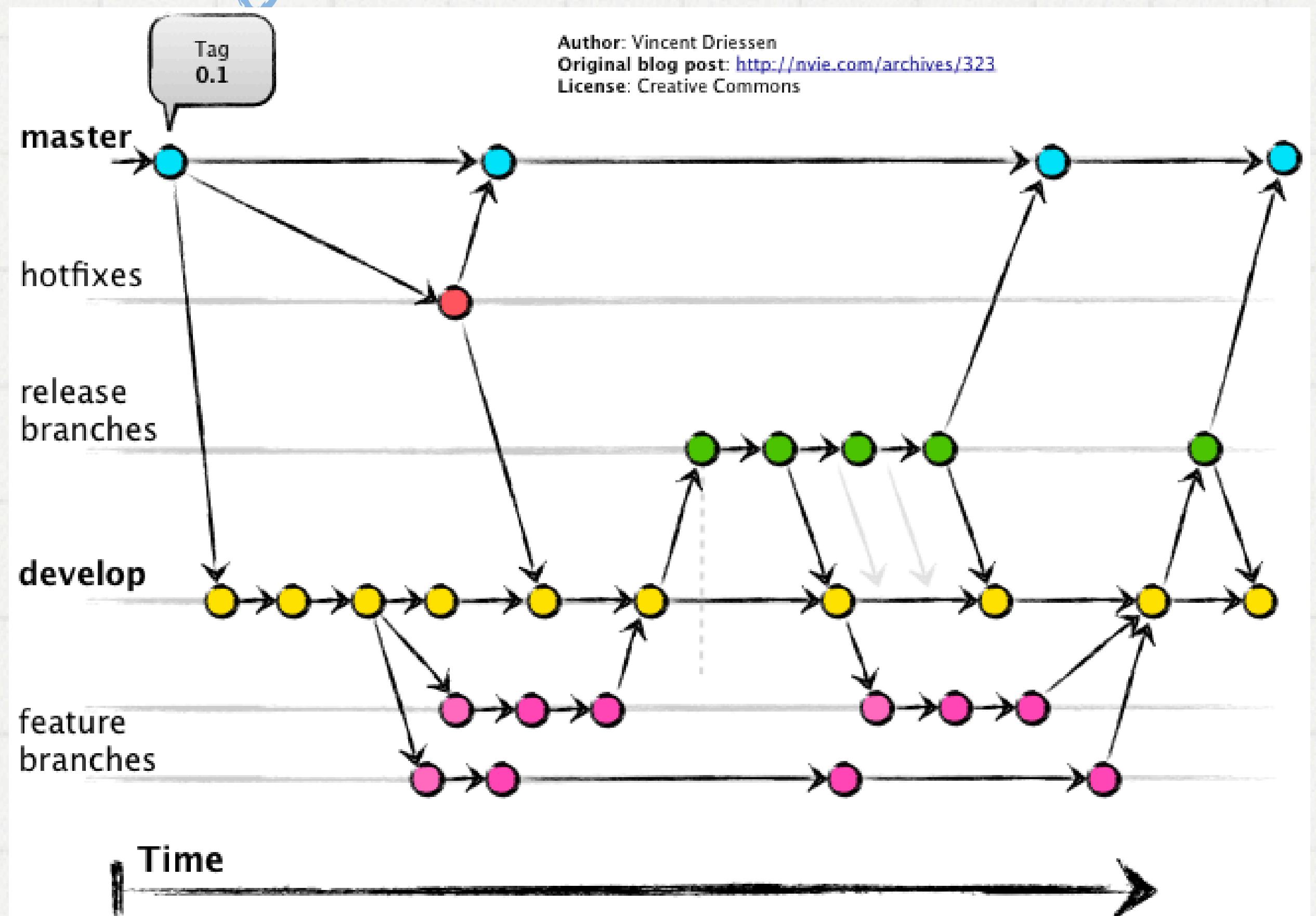
Languages Java 98.4% Python 1.5% Dockerfile 0.1%

Suggested workflows Based on your tech stack

Publish Docker Container Configure

Build, test and push Docker image to GitHub Packages.

Git Flow



Mã hóa file .env

☞ Vấn đề về file .env

Trong các dự án phần mềm, file `.env` chứa các biến môi trường quan trọng như:

- Thông tin cấu hình hệ thống.
- Khóa bảo mật.
- Đường dẫn kết nối đến cơ sở dữ liệu.

Việc để lộ file `.env` lên repository có thể dẫn đến rủi ro bảo mật nghiêm trọng, chẳng hạn như:

- Lộ thông tin kết nối cơ sở dữ liệu.
- Khóa API bị sử dụng trái phép.
- Tấn công vào hệ thống qua các thông tin nhạy cảm.

Giải pháp: Mã hóa file .env

Để bảo vệ file `.env`, nhóm áp dụng các công cụ mã hóa như **SOPS** kết hợp với **Age** để mã hóa các file chứa thông tin nhạy cảm.

Kết quả đạt được và kế hoạch tiếp theo



Tổng kết quá trình hiện thực

- Dự án đã hoàn thành kịp tiến độ, đáp ứng các mục tiêu cơ bản đặt ra ban đầu.
- Hệ thống backend được thiết kế và hiện thực đảm bảo hiệu suất, bảo mật, và khả năng mở rộng, sẵn sàng để tích hợp với các hệ thống frontend trong tương lai.
- Tuy nhiên vẫn còn tính năng chưa hiện thực kịp như cung cấp các Api hỗ trợ việc xác thực cho các nhóm Telegram, hay cung cấp khả năng tự động cập nhật bảng điểm dựa trên các Url dẫn đến File điểm tương ứng của từng giáo viên.

Những khó khăn gặp phải

Trong quá trình thực hiện dự án, nhóm đã gặp phải một số khó khăn như sau:

- Kiến thức về các công cụ và thư viện backend (như Spring Boot,...) yêu cầu nhiều thời gian để tìm hiểu và làm quen, đặc biệt là với các thành viên chưa có nhiều kinh nghiệm lập trình backend.
- Việc tích hợp Google Authentication và xây dựng cơ chế xác thực JWT đòi hỏi phải nắm rõ các nguyên tắc về bảo mật và ủy quyền, điều này gây khó khăn trong giai đoạn đầu.
- Quá trình kiểm tra và debug hệ thống backend, đặc biệt là với các yêu cầu liên quan đến bảo mật dữ liệu.
- Phối hợp và phân chia công việc giữa các thành viên để đảm bảo tiến độ và chất lượng của hệ thống.

Demo kết quả hiện thực



POSTMAN

ENVIRONMENT Environment LAYOUT Double Column LANGUAGE cURL - cURL

GRADE PORTAL API DOCUMENT

Introduction

- > Admin
- > Teacher
- > Student
- > Hall-of-fame
- > Auth For Student - Teacher

Grade Portal Api Document

GradePortal API là giao diện lập trình ứng dụng của hệ thống quản lý điểm và lớp học, cho phép các nhà phát triển tích hợp và tương tác với dữ liệu của hệ thống một cách dễ dàng và hiệu quả.

Tài liệu này cung cấp thông tin chi tiết về các endpoint, phương thức HTTP, và các tham số cần thiết để thực hiện các thao tác như quản lý khóa học, lớp học, điểm số, giáo viên và sinh viên.

Các API được xây dựng theo chuẩn RESTful, giúp đảm bảo khả năng mở rộng và tương thích cao trong các ứng dụng hiện đại. Mỗi endpoint đi kèm với mô tả chi tiết về chức năng, yêu cầu đầu vào, và ví dụ về phản hồi trả về.

Các endpoint chính bao gồm:

- /admin
 - /auth: Quản lý việc Log In, Log Out, lấy thông tin cá nhân đối với người dùng với role Admin.
 - /manage-user: Các API giúp Admin quản lý người dùng trong hệ thống.
 - /manage-course: Các API giúp Admin quản lý khóa học trong hệ thống.
 - /manage-class: Các API giúp Admin quản lý các lớp học được mở ra trong hệ thống.
 - /manage-semester: Các API giúp Admin quản lý học kỳ có trong hệ thống.
- /student
 - /auth: Quản lý việc Log In, Log Out, lấy thông tin cá nhân đối với người dùng với role Student.
 - /class-info: Các API giúp Student truy xuất những thông tin lớp học có trong hệ thống.
 - /sheetmark-info: Các API giúp Student truy xuất những thông tin về bảng điểm trong hệ thống.
 - /teacher-info: Các API giúp Student truy xuất những thông tin giáo viên dạy học có trong hệ thống.
- /teacher
 - /auth: Quản lý việc Log In, Log Out, lấy thông tin cá nhân đối với người dùng với role Teacher.
 - /manage-sheetmark: Các API giúp Teacher quản lý các bảng điểm được phân quyền trong hệ thống.
 - /manage-class: Các API giúp Teacher quản lý các lớp học được phân quyền trong hệ thống.
 - /student-info: Các API giúp Teacher truy xuất thông tin sinh viên cần thiết.

**Thank you
very much!**