

HỌC VIỆN CÔNG NGHỆ Bưu Chính Viễn Thông
KHOA CÔNG NGHỆ THÔNG TIN 1



**ĐỒ ÁN
TỐT NGHIỆP ĐẠI HỌC**

ĐỀ TÀI

**Triển khai và bổ sung plugin
cho hệ thống quản trị học tập Moodle
trên máy chủ Bộ môn Khoa học máy tính**

Giảng viên hướng dẫn : TS. NGUYỄN TẤT THẮNG

Sinh viên thực hiện 1 : Nguyễn Xuân Thức – B21DCCN140

Sinh viên thực hiện 2 : Lê Đăng Hải Anh – B21DCCN147

Sinh viên thực hiện 3 : Đặng Minh Anh – B21DCCN140

Hệ : ĐẠI HỌC CHÍNH QUY

Hà Nội, tháng 12 năm 2025

LỜI CẢM ƠN

Lời đầu tiên, chúng em xin gửi lời cảm ơn chân thành và sâu sắc tới thầy TS. Nguyễn Tất Thắng, người đã trực tiếp hướng dẫn tận tình, chu đáo, chia sẻ những ý kiến và kinh nghiệm quý báu trong suốt quá trình chúng em thực tập và thực hiện đồ án tốt nghiệp này. Sự tận tâm của thầy là động lực lớn nhất cho chúng em hoàn thiện sản phẩm này.

Chúng em cũng xin gửi lời cảm ơn đến các quý Thầy, Cô trong Học viện nói chung và khoa CNTT1 nói riêng đã luôn nhiệt huyết, tận tình trong từng bài giảng và tạo điều kiện thuận lợi nhất cho chúng em trong suốt thời gian học tập và nghiên cứu tại Học Viện Công Nghệ Bưu Chính Viễn Thông. Những kiến thức nền tảng và môi trường học tập mà Thầy, Cô và Học viện đã xây dựng trong suốt những năm qua chính là hành trang vững chắc nhất để chúng em có thể hoàn thiện đồ án này cũng như sự phát triển trong sự nghiệp của chúng em sau này.

Cuối cùng, chúng con xin được gửi lời cảm ơn tới bố mẹ, bạn bè và những người thân yêu đã luôn lo lắng, động viên, ủng hộ và tạo điều kiện cho chúng con được học tập tốt. Là chỗ dựa tinh thần và những người tiếp sức cho con có được thành công trong cuộc sống.

Vì thời gian có hạn, trình độ hiểu biết của nhóm còn nhiều hạn chế cho nên trong đồ án không tránh khỏi những thiếu sót, em rất mong nhận được sự đóng góp ý kiến của tất cả các thầy, cô giáo cũng như các bạn bè để đồ án của chúng em được hoàn thiện hơn nữa.

Chúng em xin chân thành cảm ơn!

Hà Nội, tháng 12 năm 2025

Sinh viên thực hiện

...

NHẬN XÉT, ĐÁNH GIÁ, CHO ĐIỂM (Của người hướng dẫn)

Điểm: (bằng chữ:)

Đồng ý/Không đồng ý cho sinh viên bảo vệ trước hội đồng chấm đồ án tốt nghiệp?

....., ngày tháng năm 202...

CÁN BỘ GIẢNG VIÊN HƯỚNG DẪN

NHẬN XÉT, ĐÁNH GIÁ, CHO ĐIỂM (Của giáo viên phản biện)

Điểm: (bằng chữ:)

Đồng ý/Không đồng ý cho sinh viên bảo vệ trước hội đồng chấm đồ án tốt nghiệp?

....., ngày tháng năm 202...

CÁN BỘ GIẢNG VIÊN PHẢN BIỆN

MỤC LỤC

LỜI CẢM ƠN.....	i
NHẬN XÉT, ĐÁNH GIÁ, CHO ĐIỂM	ii
NHẬN XÉT, ĐÁNH GIÁ, CHO ĐIỂM	iii
MỤC LỤC	iv
BẢNG VIẾT TẮT VÀ THUẬT NGỮ.....	vii
DANH SÁCH HÌNH VẼ.....	ix
MỞ ĐẦU	1
CHƯƠNG I. CƠ SỞ LÝ THUYẾT VÀ CÔNG NGHỆ SỬ DỤNG	6
1.1. Kiến trúc Hạ tầng và Triển khai Server	6
1.1.1. Docker và Docker Compose.....	6
1.1.2. Nginx Reverse Proxy	6
1.2. Hệ quản trị nội dung học tập Moodle	6
1.2.1. Kiến trúc Module hóa (Modular Architecture).....	7
1.3. Công nghệ Đánh giá Code (Virtual Programming Lab - VPL)	7
1.3.1. Cơ chế thực thi Native (Native Execution)	7
1.4. Trí tuệ nhân tạo và Kiến trúc Microservices.....	7
1.4.1. Kiến trúc RAG (Retrieval-Augmented Generation).....	7
1.4.2. LangChain và FastAPI.....	8
1.4.3. Mô hình ngôn ngữ lớn (LLM): Qwen 2.5 3B.....	8
1.5. Công nghệ Front-end và Thiết kế Giao diện (Frontend & UI Technologies).....	8
1.5.1. Kiến trúc Template (Mustache Logic-less Templates).....	9
1.5.2. Bộ tiền xử lý CSS (SCSS/SASS).....	9
1.5.3. Framework tương thích đa nền tảng (Bootstrap & Responsive)	10
1.5.4. Công nghệ hỗ trợ tiếp cận (Web Accessibility).....	10
1.6. Kết luận Chương I.....	10
CHƯƠNG II. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG	11
2.1. Phân tích yêu cầu	11
2.1.1. Yêu cầu chức năng.....	11
2.1.2. Yêu cầu Phi chức năng (Non-functional Requirements).....	12
2.2. Phân tích một số chức năng cơ bản của Moodle.....	12

2.2.1. Chức năng dùng chung	12
2.2.2. Chức năng cho sinh viên.....	15
2.2.3. Chức năng dành cho giảng viên.....	26
2.2.4. Chức năng cho quản trị viên.....	36
2.3 Thiết kế và cài đặt plugin hỗ trợ chấm điểm sử dụng LLM – AI Grading	46
2.3.1. Yêu cầu chức năng.....	46
2.3.2. Yêu cầu phi chức năng.....	48
2.3.3. Đặc tả Use case	48
2.3.4. Phân tích hoạt động	53
2.3.5. Thiết kế cơ sở dữ liệu	57
2.3.6. Cơ chế liên kết:	58
2.4 Thiết kế Kiến trúc cơ sở hạ tầng	59
2.4.1. Mô hình kiến trúc tổng quan.....	59
2.4.2. Các thành phần chi tiết	60
2.4.3. Quy trình luồng dữ liệu (Data Flow)	61
2.4.4. Đánh giá kiến trúc.....	62
2.5. Tổng kết chương II.....	62
CHƯƠNG III. XÂY DỰNG VÀ TRIỂN KHAI HỆ THỐNG	63
3.1 Triển khai hạ tầng Server	63
3.1.1. Thiết lập môi trường và tường lửa cho hệ thống	63
3.1.2. Cài đặt Docker và Docker Compose	63
3.1.3. Triển khai Moodle LMS với Docker Compose	64
3.1.4. Cấu hình Reverse Proxy (Nginx).....	65
3.1.5. Cài đặt và cấu hình VPL JAIL SYSTEM.....	66
3.2. Phát triển Plugin AI Chấm điểm	67
3.2.1. Thiết kế hoạt động của một số chức năng cơ bản.....	67
3.2.2. Thiết kế hoạt động của Plugin AI chấm điểm	71
3.3. Tổng kết chương III	77
CHƯƠNG IV. KẾT LUẬN.....	78
4.1 Kết quả đạt được:	78
4.2 Hạn chế của hệ thống	78
4.3 Định hướng phát triển hệ thống	78

PHỤ LỤC:	80
DANH MỤC TÀI LIỆU THAM KHẢO	83

BẢNG VIẾT TẮT VÀ THUẬT NGỮ

TỪ VIẾT TẮT VÀ THUẬT NGỮ	Ý NGHĨA
CS	Computer Science (Khoa học máy tính)
LMS	Learning Management System (Hệ thống quản trị học tập)
VPL	Virtual Programming Lab (Phòng thí nghiệm lập trình)
RAG	Retrieval-Augmented Generation (Tạo sinh tăng cường truy xuất)
UI/UX	User Interface/User Experience (Giao diện người dùng / Trải nghiệm người dùng)
LLM	Large Language Model (Mô hình ngôn ngữ lớn)
SaaS	Software as a Service (Phần mềm như một dịch vụ - Mô hình phân phối phần mềm qua internet)
SSL/TLS	Secure Sockets Layer/Transport Layer Security (Lớp cổng bảo mật - Giao thức mã hóa kết nối mạng internet)
DMZ	Demilitarized Zone (Vùng phi quân sự/Vùng đệm mạng - Nơi chứa các dịch vụ công khai tách biệt với mạng nội bộ)
SSH	Secure Shell (Giao thức mạng an toàn để vận hành các dịch vụ mạng qua một mạng không an toàn)
SMTP	Simple Mail Transfer Protocol (Giao thức truyền tải thư đơn giản - Dùng để cấu hình gửi email thông báo)
API	Application Programming Interface (Giao diện lập trình ứng dụng)
RESTful API	Representational State Transfer API (Một chuẩn thiết kế API dựa trên giao thức HTTP)

OAuth2	Open Authorization 2.0 (Giao thức ủy quyền mở - Dùng để đăng nhập bằng tài khoản Microsoft/Google)
SCSS / SASS	Sassy CSS (Bộ tiền xử lý CSS - Ngôn ngữ mở rộng giúp viết mã giao diện web có cấu trúc hơn)
HTML	HyperText Markup Language (Ngôn ngữ đánh dấu siêu văn bản)
WCAG	Web Content Accessibility Guidelines (Hướng dẫn truy cập nội dung)
WSS	WebSocket Secure (Giao thức WebSocket bảo mật - Dùng cho kết nối thời gian thực như Console VPL)
XML-RPC	XML Remote Procedure Call (Giao thức gọi thủ tục từ xa sử dụng XML)

DANH SÁCH HÌNH VẼ

Hình 1.1. Sơ đồ mô tả kiến trúc RAG tiêu chuẩn.....	8
Hình 1.2. Cơ chế tách biệt Logic và Giao diện sử dụng Mustache Engine.....	9
Hình 2.1a. Use case Đăng nhập	12
Hình 2.1b. Sơ đồ Activity của Chức năng đăng nhập	13
Hình 2.2a. Use case Đăng xuất.....	14
Hình 2.2b. Sơ đồ Activity của Chức năng Đăng xuất	15
Hình 2.3a. Use case Xem danh sách các khoá học đã ghi danh.....	16
Hình 2.3b. Sơ đồ Activity của Chức năng Xem khoá học đã ghi danh.....	17
Hình 2.4a. Use case Xem bài giảng và tài liệu học tập	17
Hình 2.4b. Sơ đồ Activity của chức năng Xem bài giảng và bài tập	19
Hình 2.5a. Use case Chức năng nộp bài trực tuyến.....	19
Hình 2.5b. Sơ đồ Activity của Chức năng nộp bài	21
Hình 2.6a. Use case Chức năng Xem điểm số và phản hồi từ giảng viên.....	22
Hình 2.6b. Sơ đồ Activity của Chức năng Xem điểm số và phản hồi của giảng viên ..	23
Hình 2.7a. Use case Tham gia làm bài kiểm tra (quiz)	24
Hình 2.7b. Sơ đồ Activity của Chức năng Tham gia làm bài kiểm tra	25
Hình 2.8a. Use case Tạo mới khoá học	26
Hình 2.8b. Sơ đồ Activity của Chức năng Tạo mới khoá học	27
Hình 2.9a. Use case Chính sửa thông tin khoá học	28
Hình 2.10a. Use case Chức năng quản lý danh sách sinh viên	29
Hình 2.10b. Sơ đồ Activity của Chức năng Quản lý sinh viên	30
Hình 2.11a. Use case Quản lý tài liệu học tập.....	31
Hình 2.11b. Sơ đồ Activity của Chức năng Quản lý tài liệu học tập	32
Hình 2.12a. Use case Quản lý học tập.....	33
Hình 2.12b. Sơ đồ Activity của Chức năng Quản lý bài tập	34
Hình 2.13a. Use case Chức năng chấm điểm	35
Hình 2.13b. Sơ đồ Activity của Chức năng Chấm điểm	36
Hình 2.14a. Use case Quản lý người dùng	37
Hình 2.14b. Sơ đồ Activity của Chức năng Quản lý người dùng	39
Hình 2.15a. Use case Quản lý khoá học và danh mục khoá học	40
Hình 2.15b. Sơ đồ Activity của Chức năng Quản lý khoá học	41

Hình 2.16a. Use case Quản lý giao diện.....	42
Hình 2.16b. Sơ đồ Activity của Chức năng Quản lý Giao diện	43
Hình 2.17a. Use case Quản lý ngôn ngữ	44
Hình 2.17b. Use case Quản lý ngôn ngữ	45
Hình 2.17c. Use case Quản lý ngôn ngữ	45
Hình 2.17d. Sơ đồ Activity của Chức năng Bật/tắt lựa chọn ngôn ngữ cho người dùng	46
Hình 2.18. Biểu đồ Use case Câu hỏi chấm điểm AI.....	49
Hình 2.19. Biểu đồ Use case Gửi yêu cầu chấm điểm	50
Hình 2.20. Biểu đồ Use case Nhận và xử lý yêu cầu chấm điểm.....	52
Hình 2.21. Biểu đồ Use case Đăng tải và đồng bộ dữ liệu.....	53
Hình 2.22. Biểu đồ hoạt động Câu hỏi chức năng chấm bài	54
Hình 2.23. Biểu đồ hoạt động Gửi yêu cầu chấm bài.....	56
Hình 2.24. Biểu đồ hoạt động nhận và xử lý yêu cầu chấm điểm.....	56
Hình 2.25. Biểu đồ hoạt động Đăng tải và đồng bộ dữ liệu	57
Hình 2.26. Cơ sở dữ liệu và mối quan hệ với các thuộc tính của Moodle	59
Hình 2.27. Kiến trúc 3 tầng triển khai trên hạ tầng mạng phân tách.....	60
Hình 3.1. Biểu đồ tuần tự Câu hỏi chấm điểm AI.....	72
Hình 3.2. Biểu đồ tuần tự gửi yêu cầu chấm điểm	73
Hình 3.3. Biểu đồ tuần tự Xử lý hàng đợi và chấm điểm.....	75
Hình 3.4. Biểu đồ tuần tự Đăng tải và Đồng bộ dữ liệu.....	77

MỞ ĐẦU

1. Tổng quan tình hình nghiên cứu

1.1. Xu hướng chuyển đổi số và các nền tảng quản trị học tập (LMS)

Trong bối cảnh chuyển đổi số trong giáo dục đại học không còn là một lựa chọn mà là xu thế tất yếu. Việc ứng dụng các hệ thống quản trị học tập (Learning Management System - LMS) đã trở thành xương sống của các mô hình đào tạo hiện đại, từ Blended Learning đến E-learning hoàn toàn.

Hiện nay, thị trường LMS trên thế giới và tại Việt Nam đang có sự phân hóa mạnh mẽ giữa các nhóm giải pháp, đặt ra bài toán lựa chọn công nghệ phù hợp cho môi trường đào tạo kỹ thuật chuyên sâu:

- **Nhóm LMS thương mại (SaaS - Software as a Service):** Các đại diện tiêu biểu như **Canvas**, **Blackboard** hay **Desire2Learn**. Ưu điểm của nhóm này là giao diện hiện đại, dịch vụ hỗ trợ ổn định và tính năng quản lý học vụ chặt chẽ. Tuy nhiên, rào cản lớn nhất là chi phí bản quyền rất cao. Quan trọng hơn, với tính chất "đóng" của mã nguồn, các hệ thống này rất khó tùy biến sâu (Deep Customization) để tích hợp các công cụ đặc thù như trình biên dịch mã nguồn (Compiler) hay các module thực hành mang tính chuyên môn đối với đào tạo ngành Khoa học máy tính nói riêng và Công nghệ thông tin nói chung.
- **Nhóm nền tảng lớp học đơn giản:** Các công cụ như **Google Classroom** hay **Microsoft Teams for Education**. Mặc dù miễn phí và dễ tiếp cận, nhưng chúng thiếu các chức năng quản lý khảo thí phức tạp (ngân hàng câu hỏi, xáo đẽ, chấm điểm tự động nhiều mức độ) và không hỗ trợ khả năng lập trình mở rộng (Plugin/Add-on) mạnh mẽ.
- **Nhóm LMS mã nguồn mở (Open Source):** Nổi bật nhất là **Moodle**, Open edX. Trong đó, Moodle (Modular Object-Oriented Dynamic Learning Environment) chiếm ưu thế tuyệt đối nhờ kiến trúc module hóa linh hoạt. Cộng đồng phát triển toàn cầu cho phép Moodle sở hữu kho tàng plugin khổng lồ. Khác với Canvas hay Google Classroom, Moodle cho phép các nhà phát triển can thiệp trực tiếp vào Core System, Database và API, tạo điều kiện thuận lợi để xây dựng và phát triển các hệ thống của riêng mình để phục vụ nhu cầu đào tạo.

1.2. Nghiên cứu về hệ thống chấm điểm lập trình tự động (Automated Assessment)

Tại Việt Nam, các nghiên cứu trước đây về LMS, thường tập trung vào việc triển khai Moodle ở mức cấu hình cơ bản trên nền tảng Windows/IIS. Mặc dù đáp ứng được nhu cầu quản lý bài giảng, nhưng việc triển khai trên Windows bộc lộ nhiều hạn chế về hiệu năng và bảo mật khi tích hợp các module chấm code như Virtual Programming Lab (VPL).

Các hệ thống chấm code truyền thống thường gặp khó khăn trong việc cô lập tiến trình (Process Isolation). Nếu không có cơ chế Sandbox (hộp cát) chặt chẽ, sinh viên có thể vô tình hoặc cố ý nộp các đoạn mã độc (như fork bomb, lệnh xóa file hệ thống, hoặc

truy cập mạng trái phép) gây treo máy chủ. Các nghiên cứu hiện tại trong nước vẫn còn thiếu chiều sâu về việc xây dựng môi trường **Execution Server** (Jail Server) trên nền tảng Linux với cơ chế chroot hoặc Containerization để đảm bảo an toàn tuyệt đối.

1.3. Tầm quan trọng của trải nghiệm người dùng (UX/UI) và hạn chế của hệ thống cũ

Thực trạng các hệ thống hỗ trợ học tập cũ tại Bộ môn, điển hình là trang khmt.net, đang bộc lộ nhiều bất cập nghiêm trọng về mặt giao diện:

- **Công nghệ frontend lạc hậu:** Hệ thống cũ được xây dựng trên các công nghệ web thế hệ trước, dẫn đến giao diện rất sơ sài, đơn điệu và thiếu tính tương tác. Việc bố trí các thành phần (layout) thường cứng nhắc, thiếu các chỉ dẫn trực quan (visual cues) cần thiết, khiến sinh viên mất nhiều thời gian để tìm kiếm bài tập hoặc tài liệu.
- **Thiếu tính tương thích di động (Non-responsive):** Một hạn chế chí mạng của hệ thống cũ là không hỗ trợ chuẩn **Responsive Web Design**. Trong khi thói quen của sinh viên hiện nay là truy cập tài liệu mọi lúc mọi nơi thông qua smartphone hay máy tính bảng.
- **Thiếu đồng bộ nhận diện thương hiệu:** Giao diện cũ mang tính chất "tự phát", thiếu sự đầu tư bài bản về hệ thống màu sắc, typography và logo chuẩn của Học viện. Điều này làm giảm tính chuyên nghiệp và sự gắn kết của sinh viên với môi trường học tập số của nhà trường.

Xu hướng hiện nay của các LMS hiện đại (như Moodle 4.x trở lên) là chuyển dịch sang phong cách thiết kế *Minimalism (Tối giản)* và *Mobile-First (Ưu tiên di động)*. Mặc dù Moodle cung cấp giao diện mặc định (Boost Theme) khá hiện đại, nhưng để đáp ứng các yêu cầu đặc thù của một hệ thống chấm code (cần không gian rộng để hiển thị trình soạn thảo code, highlight cú pháp) và mang đậm bản sắc của Bộ môn KHMT, việc nghiên cứu tùy biến sâu giao diện (Child Theme Development) dựa trên SCSS và các Template engine là một hướng đi kỹ thuật quan trọng và cần thiết của đề tài.

2. Tính cấp thiết của vấn đề:

- **Thực trạng hạ tầng công nghệ hiện tại:** Hiện tại, hệ thống hỗ trợ học tập của Bộ môn Khoa học Máy tính (trang khmt.net cũ) đang đổi mới với sự xuồng cấp nghiêm trọng về mặt công nghệ. Hệ thống được xây dựng trên nền tảng cũ, thiếu các bản và bảo mật, và không tương thích tốt với các thiết bị di động hiện đại.

- **Áp lực quá tải trong công tác giảng dạy:** Với sự gia tăng nhanh chóng số lượng sinh viên ngành Công nghệ thông tin, quy mô mỗi lớp học phần thường dao động từ 60 đến 80 sinh viên. Đối với các môn học lập trình (C++, Java, Python, Web), giảng viên không chỉ chấm kết quả đúng/sai mà còn phải xem xét tư duy giải toán, trả lời câu hỏi. Việc chấm thủ công hàng trăm bài tập tự luận mỗi tuần là bất khả thi, dẫn đến tình trạng sinh viên nhận phản hồi chậm, làm giảm động lực học tập.

3. Mục tiêu đề tài:

Đề tài hướng tới việc xây dựng một hệ sinh thái học tập thông minh, toàn diện, giải quyết triệt để các hạn chế về hiệu năng và trải nghiệm người dùng của hệ thống cũ. Mục tiêu cụ thể được phân rõ dựa trên nhiệm vụ kỹ thuật chuyên sâu của từng thành viên:

3.1. Về Hạ tầng và Kiến trúc hệ thống (Nguyễn Xuân Thức)

Mục tiêu là thiết kế và triển khai một hạ tầng **kiến trúc lai (Hybrid Architecture)**, cân bằng giữa khả năng quản lý linh hoạt và hiệu năng xử lý tối đa:

- **Hiện đại hóa quy trình vận hành (DevOps):** Triển khai Web Server (Moodle Core) trên nền tảng **Docker Container**. Việc container hóa giúp chuẩn hóa môi trường phát triển và sản xuất, giải quyết vấn đề xung đột thư viện (dependency hell), đồng thời đơn giản hóa quy trình sao lưu và phục hồi hệ thống.
- **Tối ưu hóa khả năng chịu tải (High Performance):** Thiết lập **Nginx** làm Reverse Proxy để xử lý các kết nối đồng thời. Mục tiêu là đảm bảo hệ thống hoạt động mượt mà ngay cả trong các khung giờ cao điểm thi cử với lượng truy cập lớn.
- **Đảm bảo an toàn và tốc độ thực thi (Security & Speed):** Xây dựng **VPL Jail Server** chạy trực tiếp (Native) trên nền tảng Linux thay vì ảo hóa. Giải pháp này nhằm tận dụng tối đa tài nguyên phần cứng (CPU/RAM) để tăng tốc độ biên dịch code, đồng thời áp dụng các cơ chế cô lập nghiêm ngặt (chroot, overlayfs) để ngăn chặn mã độc từ bài làm sinh viên ảnh hưởng đến máy chủ.

3.2. Về Tính năng và Tích hợp thông minh (Lê Đăng Hải Anh)

Mục tiêu là tích hợp mô hình ngôn ngữ lớn (LLM) chạy cục bộ vào hệ thống để tự động hóa quy trình đánh giá các bài thi tự luận (Essay/Open-ended questions), giúp giảm tải áp lực chấm bài thủ công cho giảng viên:

- **Triển khai Mô hình Ngôn ngữ cục bộ (Self-hosted Local LLM):**
 - Thực hiện đóng gói và triển khai các mô hình mã nguồn mở (như Llama, Mistral hoặc Gemma) trên nền tảng **Docker**. Việc chạy model cục bộ (On-premise) đảm bảo tính riêng tư dữ liệu tuyệt đối cho đề thi và bài làm của sinh viên, loại bỏ sự phụ thuộc vào các dịch vụ API bên thứ ba.
 - Xây dựng API nội bộ (Internal API service) để phục vụ việc giao tiếp giữa model AI và hệ thống web, tối ưu hóa tài nguyên phần cứng để đảm bảo tốc độ phản hồi khi xử lý các đoạn văn bản dài.
- **Phát triển Plugin chấm điểm tự luận thông minh:**
 - Nghiên cứu sâu **Moodle Grading API** để xây dựng plugin chấm điểm chuyên biệt cho dạng câu hỏi tự luận (Essay Question).
 - Xây dựng thuật toán đánh giá dựa trên ngữ nghĩa (Semantic Analysis): Thay vì chỉ so khớp từ khóa đơn thuần, hệ thống sử dụng AI để "hiểu" nội dung bài làm của sinh viên, so sánh mức độ tương đồng về ý nghĩa với

đáp án mẫu hoặc barem chấm điểm (Rubric) do giảng viên cung cấp để đưa ra điểm số đề xuất.

3.3. Về Giao diện và Trải nghiệm người dùng (Đặng Minh Anh)

Mục tiêu là tái thiết kế toàn diện giao diện người dùng, tập trung vào tính thẩm mỹ và hiệu quả sử dụng:

- **Đồng bộ nhận diện thương hiệu:** Thiết kế và phát triển giao diện (Theme) mang đậm bản sắc của Học viện (PTIT), tạo môi trường học tập chính quy, chuyên nghiệp và tăng tính gắn kết cho sinh viên.
- **Tùy biến sâu về kỹ thuật Frontend:** Xây dựng Child Theme kế thừa từ kiến trúc **Boost Theme** của Moodle 4.x. Sử dụng ngôn ngữ **SCSS** để quản lý hệ thống màu sắc/bố cục và **Mustache Templates** để can thiệp sâu vào cấu trúc HTML, giúp giao diện thoát khỏi sự rập khuôn mặc định.
- **Tối ưu hóa UX đa nền tảng:** Đảm bảo hệ thống đạt chuẩn **Responsive Design**, hiển thị tối ưu trên mọi thiết bị (Laptop, Tablet, Smartphone). Đặc biệt chú trọng cải tiến giao diện trình soạn thảo code (Code Editor), bảng thông báo lỗi và dashboard theo dõi tiến độ để sinh viên dễ dàng thao tác.

4. Đối tượng và phạm vi nghiên cứu:

4.1. Đối tượng nghiên cứu

Đề tài tập trung nghiên cứu các đối tượng công nghệ và kỹ thuật sau:

- **Công nghệ Áo hóa và Hệ thống:** Kiến trúc **Docker Engine** và Docker Compose cho việc đóng gói ứng dụng Web; Các cơ chế quản lý tài nguyên (Cgroups) và cô lập tiến trình (Namespaces, Chroot) trên nhân hệ điều hành Linux phục vụ cho Jail Server.
- **Công nghệ Phát triển Plugin Moodle:** Kiến trúc Modular của Moodle, cơ chế Events/Hooks, Web Services API và cấu trúc dữ liệu nội tại của Moodle (Moodle Database Schema).
- **Công nghệ Trí tuệ nhân tạo ứng dụng:** Các kỹ thuật Prompt Engineering (kỹ thuật gợi ý) và giao thức kết nối RESTful API với các mô hình LLM (OpenAI/DeepSeek/...).
- **Công nghệ Frontend hiện đại:** Framework Bootstrap 5, bộ tiền xử lý CSS (SASS/SCSS) và ngôn ngữ template Mustache dùng trong Moodle Theme.

4.2. Phạm vi nghiên cứu

- **Phạm vi triển khai:** Hệ thống được triển khai thực nghiệm trên hạ tầng máy chủ nội bộ của trường, áp dụng mô hình lai: Web Server chạy trên Docker và Execution Server chạy Native trên cùng một node vật lý.
- **Phạm vi chức năng:** Tập trung hoàn thiện quy trình nộp bài và chấm điểm tự động cho các bài tập lập trình; tính năng phản hồi code thông minh qua AI; và

giao diện người dùng cho sinh viên/giảng viên. Các tính năng quản lý hành chính khác của trường (như thu học phí, đăng ký tín chỉ) nằm ngoài phạm vi đề tài.

- **Phạm vi dữ liệu:** Thủ nghiệm với dữ liệu mẫu và áp dụng thí điểm cho các lớp học phần Cấu trúc dữ liệu và giải thuật, Lập trình hướng đối tượng (Java/C++), Lập trình Python cơ bản.

5. Phương pháp nghiên cứu

Nhóm áp dụng phương pháp luận phát triển phần mềm linh hoạt kết hợp nghiên cứu tài liệu chuyên sâu:

- **Nghiên cứu tài liệu:** Phân tích tài liệu kỹ thuật (Developer Docs) của Moodle, tài liệu triển khai VPL Jail System của Đại học Las Palmas de Gran Canaria , và các bài báo về ứng dụng AI trong giáo dục.
- **Phân tích thiết kế hệ thống:** Sử dụng ngôn ngữ UML để mô hình hóa kiến trúc hệ thống, quy trình nghiệp vụ và cấu trúc dữ liệu, dựa trên các mẫu thiết kế (Design Patterns) trong Moodle.
- **Thực nghiệm:** Triển khai, kiểm thử và tinh chỉnh hệ thống dựa trên phản hồi thực tế từ giảng viên và sinh viên.

CHƯƠNG I. CƠ SỞ LÝ THUYẾT VÀ CÔNG NGHỆ SỬ DỤNG

Chương này trình bày các cơ sở lý thuyết và nền tảng công nghệ được sử dụng để xây dựng hệ thống. Hệ thống được thiết kế dựa trên kiến trúc hiện đại, kết hợp giữa sự ổn định của Moodle LMS, tính linh hoạt của công nghệ Container hóa (Docker) và sức mạnh của các mô hình ngôn ngữ lớn (LLM) thông qua kiến trúc Microservices.

1.1. Kiến trúc Hạ tầng và Triển khai Server

Để đảm bảo tính nhất quán của môi trường phát triển và triển khai, hệ thống sử dụng công nghệ ảo hóa ở cấp độ hệ điều hành (Containerization).

1.1.1. Docker và Docker Compose

Hệ thống Moodle và Cơ sở dữ liệu MariaDB được đóng gói và vận hành trong các container riêng biệt thông qua Docker Compose.

- Tính cô lập (Isolation):** Mỗi dịch vụ chạy trong một môi trường độc lập, ngăn chặn xung đột thư viện giữa PHP (Moodle) và các ứng dụng khác trên máy chủ.
- Docker Compose:** Đóng vai trò là công cụ điều phối (orchestration) quy mô nhỏ, quản lý việc khởi tạo, kết nối mạng nội bộ (internal network) và volume dữ liệu giữa container Web Server (Moodle) và Database Server.

1.1.2. Nginx Reverse Proxy

Thay vì để Docker container tiếp xúc trực tiếp với Internet, hệ thống sử dụng Nginx cài đặt trên máy chủ vật lý (Host machine) làm Reverse Proxy (Máy chủ ủy quyền ngược).

- Nhiệm vụ:** Nginx đóng vai trò là trạm trung chuyển, tiếp nhận các yêu cầu HTTP/HTTPS từ người dùng mạng ngoài (Internet) và điều hướng (route) vào cổng nội bộ của Docker container.
- Ưu điểm:** Tăng cường bảo mật, hỗ trợ cân bằng tải và quản lý chứng chỉ SSL tập trung.

1.2. Hệ quản trị nội dung học tập Moodle

Moodle (Modular Object-Oriented Dynamic Learning Environment) đóng vai trò là lõi (Core) của hệ thống quản lý học tập.

1.2.1. Kiến trúc Module hóa (Modular Architecture)

Moodle hoạt động dựa trên kiến trúc lõi và các plugin mở rộng. Trong đồ án này, nhóm phát triển theo hướng Local Plugin (local_aigrading).

- **Cơ chế Hook và Event:** Plugin sử dụng cơ chế *Event Observer* (như course_module_created, course_module_updated) để lắng nghe các thay đổi dữ liệu từ Core mà không cần can thiệp vào mã nguồn gốc của Moodle.
- **Database Schema:** Hệ thống sử dụng MariaDB. Các bảng dữ liệu phục vụ AI (như mdl_local_aigrading_tasks) được thiết kế riêng biệt và liên kết với bảng mdl_assign của Moodle thông qua khóa ngoại.

1.3. Công nghệ Đánh giá Code (Virtual Programming Lab - VPL)

Đối với các bài tập lập trình, hệ thống sử dụng module VPL để biên dịch và thực thi mã nguồn.

1.3.1. Cơ chế thực thi Native (Native Execution)

Khác với mô hình Jail Server tách biệt hoàn toàn, trong hệ thống này, VPL được cấu hình để chạy trực tiếp (Native) trên cùng Server vật lý chứa hạ tầng Docker.

- **Cơ chế hoạt động:** VPL Daemon (vpl-jail-system) hoạt động như một dịch vụ nền trên hệ điều hành Host. Khi sinh viên nộp bài, Moodle (trong Docker) gửi mã nguồn ra VPL Daemon trên Host để thực thi.
- **Môi trường Sandbox (Chroot):** Dù chạy trên cùng server, VPL vẫn sử dụng kỹ thuật chroot để tạo ra một hệ thống tệp tin ảo hạn chế, đảm bảo mã nguồn của sinh viên không thể truy cập hoặc phá hoại các tệp tin hệ thống hoặc can thiệp vào các Docker container đang chạy.

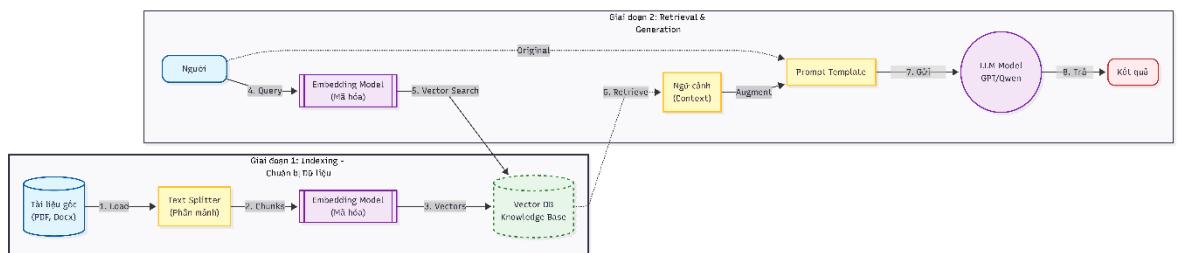
1.4. Trí tuệ nhân tạo và Kiến trúc Microservices

Hệ thống tích hợp AI không chạy trực tiếp trên nền tảng PHP của Moodle mà được tách ra thành một Service riêng biệt (Backend AI), giao tiếp với Moodle qua RESTful API. Kiến trúc này tối ưu hóa hiệu năng và tận dụng được các thư viện mạnh mẽ của Python.

1.4.1. Kiến trúc RAG (Retrieval-Augmented Generation)

Hệ thống sử dụng kỹ thuật RAG để nâng cao độ chính xác của AI khi chấm bài.

- Nguyên lý:** Thay vì chỉ dựa vào kiến thức đã học của mô hình, hệ thống sẽ truy xuất (Retrieve) các thông tin liên quan từ dữ liệu bên ngoài (như file đính kèm, đáp án mẫu, giáo trình) để cung cấp ngữ cảnh (Context) cho AI trước khi sinh câu trả lời (Generation).
- ChromaDB (Vector Database):** Đóng vai trò là bộ nhớ dài hạn cho AI. Các tài liệu tham khảo và lịch sử chấm bài được mã hóa thành các vector (embeddings) và lưu trữ trong ChromaDB, cho phép truy vấn ngôn ngữ nhanh chóng.



Hình 1.1. Sơ đồ mô tả kiến trúc RAG tiêu chuẩn

1.4.2. LangChain và FastAPI

- FastAPI:** Framework Python hiện đại, hiệu năng cao dùng để xây dựng API server. Nó đóng vai trò nhận request từ Moodle plugin, xử lý logic và trả về kết quả chấm điểm.
- LangChain:** Framework dùng để điều phối luồng xử lý của LLM. LangChain giúp kết nối mô hình ngôn ngữ với ChromaDB, quản lý Prompt Template và xử lý chuỗi tác vụ (Chain) phức tạp trong quy trình chấm điểm.

1.4.3. Mô hình ngôn ngữ lớn (LLM): Qwen 2.5 3B

Mô hình AI được lựa chọn là **Qwen 2.5 3B**.

- Đặc điểm:** Đây là mô hình mã nguồn mở (Open Weights) có kích thước nhỏ gọn (3 tỷ tham số), phù hợp để triển khai cục bộ (Local deployment) trên server có tài nguyên giới hạn mà vẫn đảm bảo khả năng hiểu ngôn ngữ và code tốt.
- Vai trò:** Mô hình nhận đầu vào là bài làm sinh viên + ngữ cảnh từ ChromaDB + Prompt hướng dẫn chấm, sau đó sinh ra điểm số và nhận xét chi tiết.

1.5. Công nghệ Front-end và Thiết kế Giao diện (Frontend & UI Technologies)

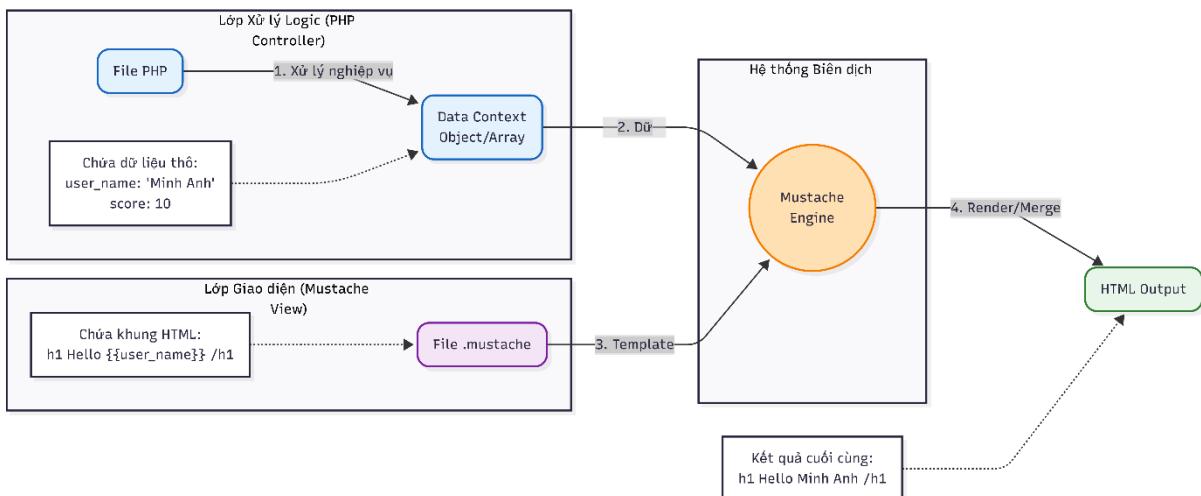
Để đáp ứng yêu cầu khắt khe về trải nghiệm người dùng (UX) và đồng bộ nhận diện thương hiệu của Học viện, nhóm không sử dụng giao diện mặc định mà phát triển

một hệ thống giao diện tùy biến sâu (Custom Theme) dựa trên các công nghệ Front-end hiện đại.

1.5.1. Kiến trúc Template (Mustache Logic-less Templates)

Hệ thống tách biệt hoàn toàn lớp giao diện (View) và lớp xử lý logic (Controller) thông qua việc sử dụng Mustache Templates.

- Cơ chế:** Các tệp tin .mustache đóng vai trò là khung xương HTML. Thay vì nhúng mã PHP trực tiếp vào HTML (gây khó khăn cho việc bảo trì), Mustache sử dụng các thẻ tag (placeholder) để hiển thị dữ liệu.
- Lợi ích:** Giúp mã nguồn sạch (clean code), dễ dàng tái sử dụng các thành phần giao diện (components) như thanh điều hướng (navbar), chân trang (footer) hay các thẻ khóa học (course cards).



Hình 1.2. Cơ chế tách biệt Logic và Giao diện sử dụng Mustache Engine

1.5.2. Bộ tiền xử lý CSS (SCSS/SASS)

Việc quản lý hàng nghìn dòng mã giao diện được thực hiện thông qua SCSS (Sassy CSS), một bản nâng cấp mạnh mẽ của CSS truyền thống.

- Biến (Variables):** Hệ thống định nghĩa bộ mã màu chuẩn của PTIT (Đỏ: #DE221A, Xanh: #061A4F) thành các biến toàn cục. Khi cần thay đổi giao diện theo mùa hoặc sự kiện, chỉ cần thay đổi giá trị biến một lần, toàn bộ website sẽ tự động cập nhật.
- Mixins & Nesting:** Sử dụng các cấu trúc lồng nhau giúp mã nguồn CSS có cấu trúc rõ ràng, dễ đọc và tối ưu hóa dung lượng khi tải trang.

1.5.3. Framework tương thích đa nền tảng (Bootstrap & Responsive)

Giao diện được xây dựng trên nền tảng Bootstrap Framework, đảm bảo tính tương thích (Responsive Design) trên mọi thiết bị.

- **Grid System:** Sử dụng hệ thống lưới 12 cột để bố trí các phần tử linh hoạt.

1.5.4. Công nghệ hỗ trợ tiếp cận (Web Accessibility)

Dựa trên tiêu chuẩn WCAG 2.1, hệ thống tích hợp bộ công cụ hỗ trợ người dùng đặc biệt (Accessibility Tools) sử dụng Javascript (AMD Modules). Cho phép người dùng tùy chỉnh cỡ chữ, độ tương phản cao (High Contrast) và phông chữ dành cho người mắc chứng khó đọc (Dyslexia Friendly Font - OpenDyslexic). Đảm bảo hệ thống học tập công bằng, dễ tiếp cận cho mọi đối tượng sinh viên.

1.6. Kết luận Chương I

Chương I đã trình bày tổng quan về kiến trúc hệ thống lai (Hybrid Architecture). Việc kết hợp Docker cho hạ tầng, Nginx cho điều hướng, VPL Native cho thực thi code truyền thống, và mô hình Microservices (FastAPI + LangChain + ChromaDB) cho AI tạo nên một giải pháp chấm điểm tự động toàn diện, vừa đảm bảo tính an toàn hệ thống vừa tận dụng được sức mạnh của các mô hình AI thế hệ mới.

CHƯƠNG II. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

Tiếp nối cơ sở lý thuyết và tổng quan về công nghệ đã được trình bày tại Chương I, Chương II sẽ tập trung đi sâu vào việc phân tích bài toán thực tế đặt ra cho hệ thống Moodle LMS tại Bộ môn Khoa học máy tính. Mục tiêu của chương này là chuyển đổi các nhu cầu nghiệp vụ của giảng viên và sinh viên thành các yêu cầu kỹ thuật cụ thể, từ đó xây dựng bản thiết kế hệ thống tối ưu.

Nội dung chương sẽ bao gồm ba phần chính: Thứ nhất là phân tích một số chức năng cơ bản của Moodle sau đó phân tích chi tiết các yêu cầu chức năng cho plugin chấm điểm tự động (AI Grading) và các yêu cầu phi chức năng về hiệu năng, bảo mật. Thứ hai là thiết kế cơ sở dữ liệu (Database Schema) để đảm bảo khả năng lưu trữ và truy xuất dữ liệu lớn cho các tác vụ AI. Cuối cùng, chương sẽ trình bày phân tích và thiết kế giao diện (Theme) nhằm đảm bảo tính thẩm mỹ và tối ưu trải nghiệm người dùng. Các bản thiết kế trong chương này đóng vai trò là "bản vẽ kỹ thuật" định hướng cho quá trình phát triển ở chương tiếp theo.

2.1. Phân tích yêu cầu

2.1.1. Yêu cầu chức năng

A. Nhóm chức năng cốt lõi (Core LMS Functions): Các chức năng này được kế thừa từ Moodle Core và tinh chỉnh lại.

- **Đăng nhập hệ thống:** Hệ thống phải hỗ trợ xác thực đa phương thức. Ngoài tài khoản cục bộ, hệ thống tích hợp đăng nhập qua **Microsoft Office 365** (tài khoản Email Edu của trường) sử dụng giao thức OAuth2. Quy trình này giúp sinh viên không phải nhớ thêm mật khẩu và đồng bộ dữ liệu người dùng tự động (Xem Phụ lục).
- **Quản lý Khóa học và Ghi danh:** Giảng viên có quyền tạo khóa học, cấu hình phương thức ghi danh (tự động hoặc thủ công). Sinh viên truy cập danh sách khóa học thông qua Dashboard cá nhân hóa.

B. Nhóm chức năng Đánh giá và AI (AI & Assessment): Đây là nhóm chức năng phát triển mới thông qua plugin local_aigrading.

- **Cấu hình AI Grading (Use case 1):** Tại giao diện cài đặt bài tập (Assignment), giảng viên có thêm tùy chọn "AI Grading". Hệ thống cho phép nhập "System Prompt" (hướng dẫn chấm chi tiết) và tải lên file "Đáp án mẫu". Dữ liệu này được lưu trữ riêng biệt để phục vụ quá trình RAG.
- **Gửi yêu cầu chấm điểm (Use case 2):** Trong danh sách bài nộp, giảng viên có thể chọn một hoặc nhiều sinh viên và nhấn nút "Gửi đi chấm AI". Hệ thống phải kiểm tra tính hợp lệ của file nộp trước khi đưa vào hàng đợi xử lý.
- **Xử lý và Phản hồi (Use case 3):** Hệ thống tự động xử lý các yêu cầu trong hàng đợi. Kết quả trả về gồm: Điểm số đề xuất (trên thang 10) và Nhận xét chi tiết (Feedback comments) giải thích lý do trừ điểm hoặc gợi ý cải thiện code.

- **Rà soát và Công bố:** Giảng viên có quyền xem điểm do AI chấm (ở trạng thái Nháp/Draft), chỉnh sửa lại nếu cần thiết, sau đó mới xác nhận lưu vào Sổ điểm (Gradebook) chính thức.

C. Nhóm chức năng Giao diện (Theme & UX):

- **Dashboard tùy biến:** Trang chủ cá nhân của sinh viên phải hiển thị rõ ràng các Deadline sắp tới, thanh tiến độ học tập của từng môn.
- **Code Editor tối ưu:** Giao diện làm bài tập lập trình (VPL) cần được tối ưu hóa không gian hiển thị, hỗ trợ highlight cú pháp (Syntax highlighting) và các phím tắt tiện dụng.

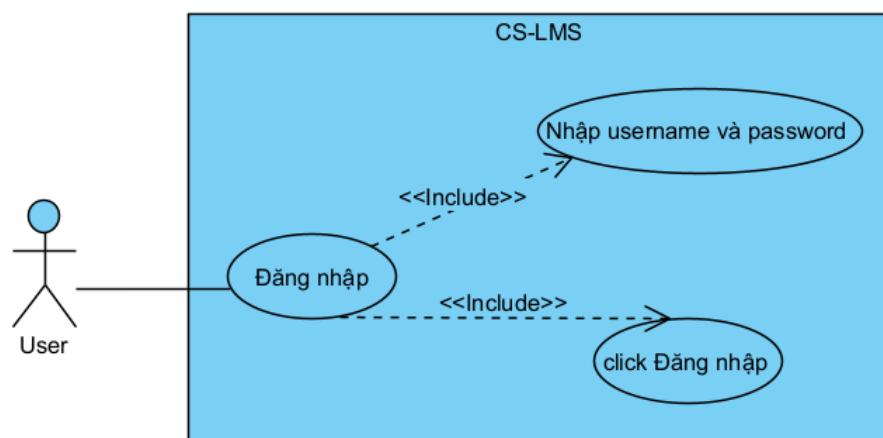
2.1.2. Yêu cầu Phi chức năng (Non-functional Requirements)

- **Hiệu năng:** Hệ thống phải đảm bảo thời gian phản hồi trang (Page Load Time) dưới 2 giây đối với các trang cơ bản. Đối với tác vụ chấm code, thời gian biên dịch và chạy không quá 5 giây/bài.
- **Tính sẵn sàng và độ tin cậy:** Cơ chế Adhoc Task đảm bảo rằng nếu AI Service bị gián đoạn tạm thời, các yêu cầu chấm điểm không bị mất đi mà sẽ được thử lại (Retry) sau đó.
- **Bảo mật:** Tuân thủ nghiêm ngặt các nguyên tắc bảo mật: Mọi đầu vào từ sinh viên (Code, Text) đều phải được lọc (Sanitization); Code thực thi phải nằm trong Jail Server; API giao tiếp nội bộ phải có Token bảo vệ.

2.2. Phân tích một số chức năng cơ bản của Moodle

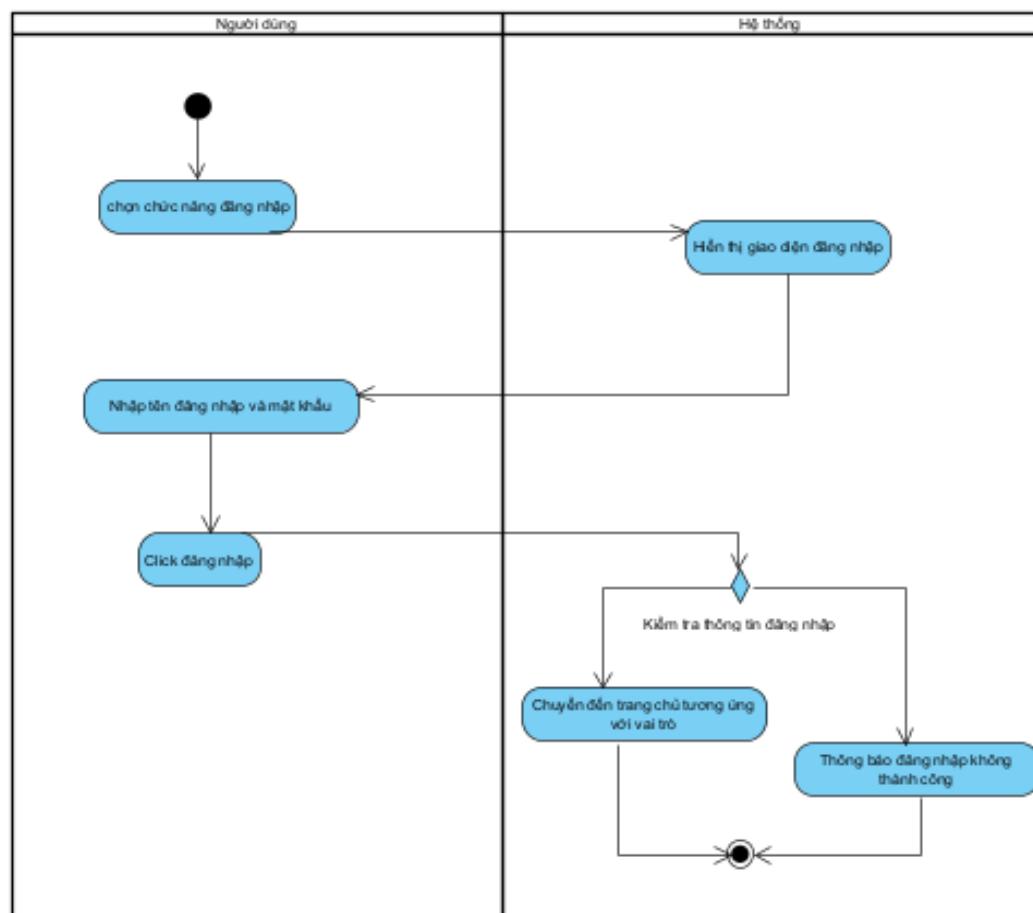
2.2.1. Chức năng dùng chung

A, Đăng nhập



Hình 2.1a. Use case Đăng nhập

Chức năng	Đăng nhập
Tác nhân	Người dùng
Tiền điều kiện	Đã có tài khoản hệ thống
Kịch bản chính	<ol style="list-style-type: none"> 1. Người dùng truy cập trang cs-lms.ptit.edu.vn 2. Hệ thống hiển thị form đăng nhập gồm: <ul style="list-style-type: none"> • Tên đăng nhập/Tài khoản • Mật khẩu 3. Người dùng nhập thông tin đăng nhập. 4. Người dùng nhấn nút Đăng nhập. 5. Hệ thống chuyển sang giao diện trang chủ
Ngoại lệ	<ol style="list-style-type: none"> 1. Người dùng nhập sai thông tin đăng nhập 2. Hệ thống hiển thị thông báo: “Tên đăng nhập hoặc mật khẩu không đúng.”



Hình 2.1b. Sơ đồ Activity của Chức năng đăng nhập

Phân tích hoạt động:

Bước 1 (Người dùng):

Người dùng (Sinh viên/Giảng viên/Quản trị viên) tác động vào ứng dụng bằng cách truy cập trang đăng nhập và chọn chức năng “Đăng nhập”.

Bước 2 (Hệ thống):

Hệ thống nhận được yêu cầu, luồng xử lý chuyển sang phía hệ thống. Hệ thống thực hiện hành động “Hiển thị giao diện đăng nhập” để người dùng nhập thông tin xác thực.

Bước 3 (Người dùng):

Sau khi giao diện hiển thị, luồng xử lý quay về phía người dùng. Người dùng thực hiện thao tác nhập liệu: “Nhập tên đăng nhập và mật khẩu”.

Bước 4 (Người dùng):

Người dùng xác nhận thao tác bằng cách “Nhấn nút ‘Đăng nhập’”.

Bước 5 (Hệ thống):

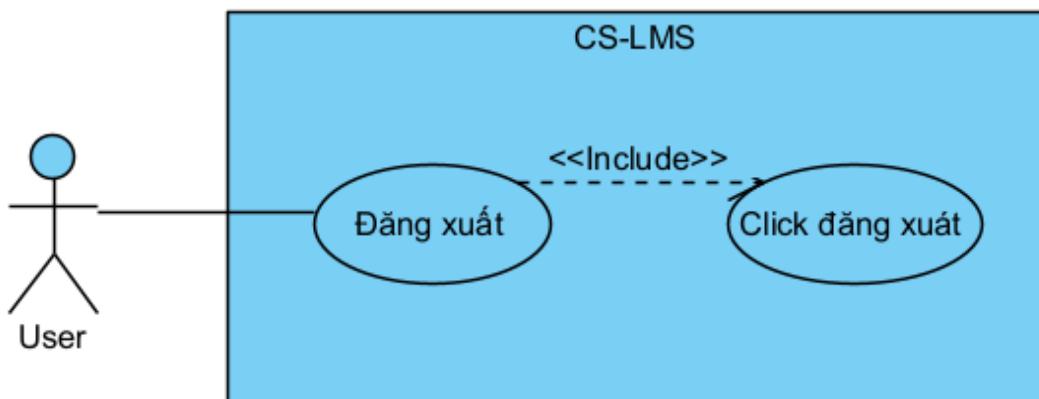
Luồng xử lý chuyển về phía hệ thống để xử lý dữ liệu. Hệ thống thực hiện hành động “Kiểm tra thông tin đăng nhập” bằng cách đối chiếu với dữ liệu trong CSDL.

Bước 6 (Hệ thống):

- Nếu thông tin hợp lệ, hệ thống thực hiện “Tạo phiên đăng nhập (session)” và “Chuyển người dùng đến trang chính tương ứng với vai trò”.

- Nếu thông tin không hợp lệ, hệ thống gửi phản hồi “Thông báo đăng nhập không thành công” cho người dùng.

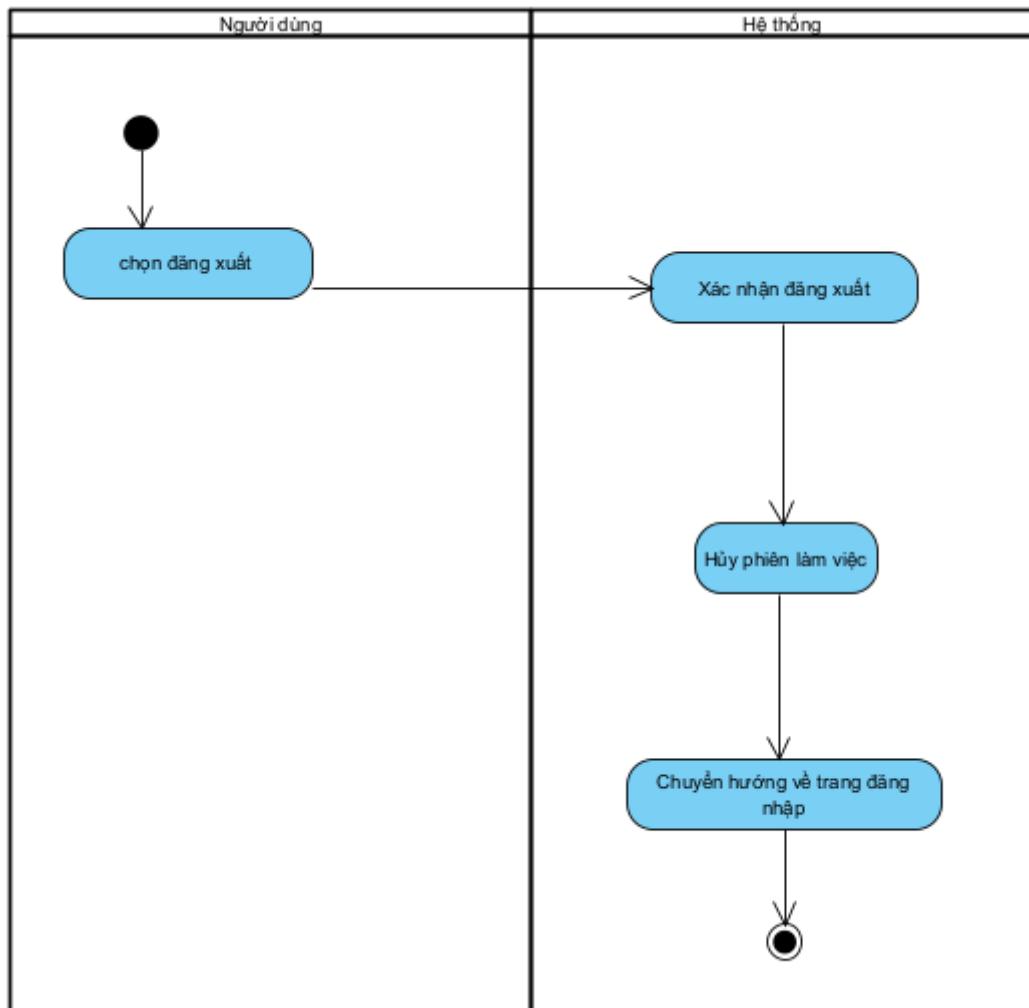
B, Đăng xuất



Hình 2.2a. Use case Đăng xuất

Chức năng	Đăng xuất
Tác nhân	Người dùng
Tiền điều kiện	Người dùng đang trong trạng thái đăng nhập

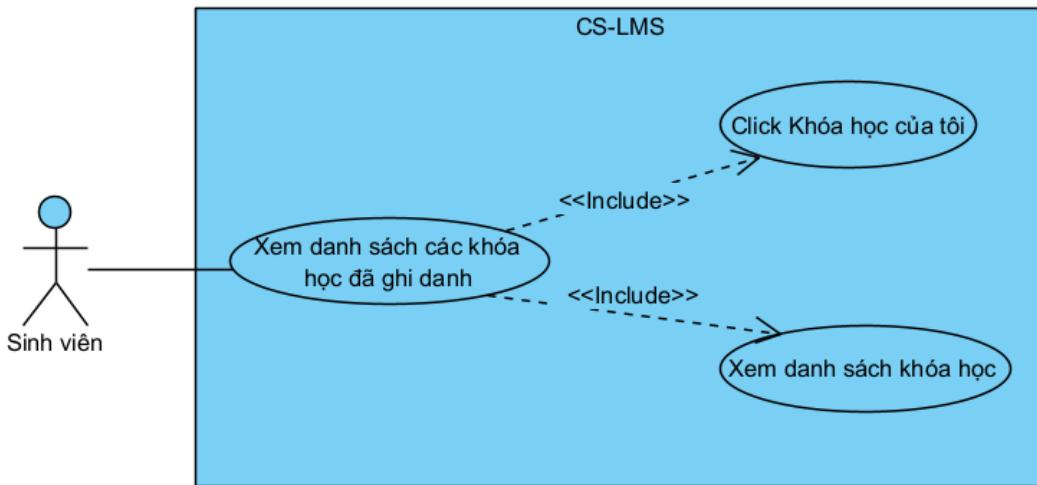
Kịch bản chính	<ol style="list-style-type: none"> Người dùng click “Đăng xuất” trên giao diện hệ thống Hệ thống thoát khỏi trạng thái đăng nhập và quay về giao diện đăng nhập
Ngoại lệ	<ol style="list-style-type: none"> Người dùng click đăng xuất Hệ thống không đăng xuất ra khỏi tài khoản



Hình 2.2b. Sơ đồ Activity của Chức năng Đăng xuất

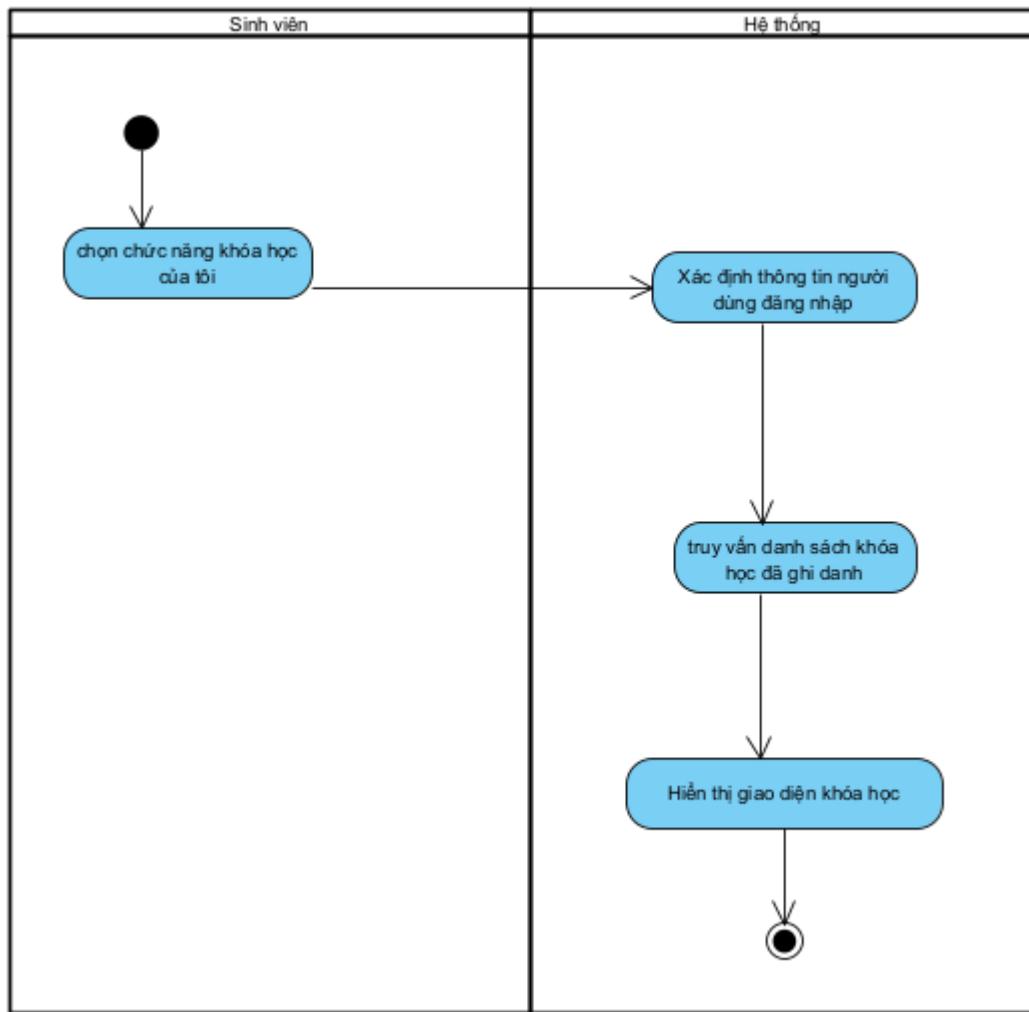
2.2.2. Chức năng cho sinh viên

A, Xem danh sách các khóa học đã ghi danh

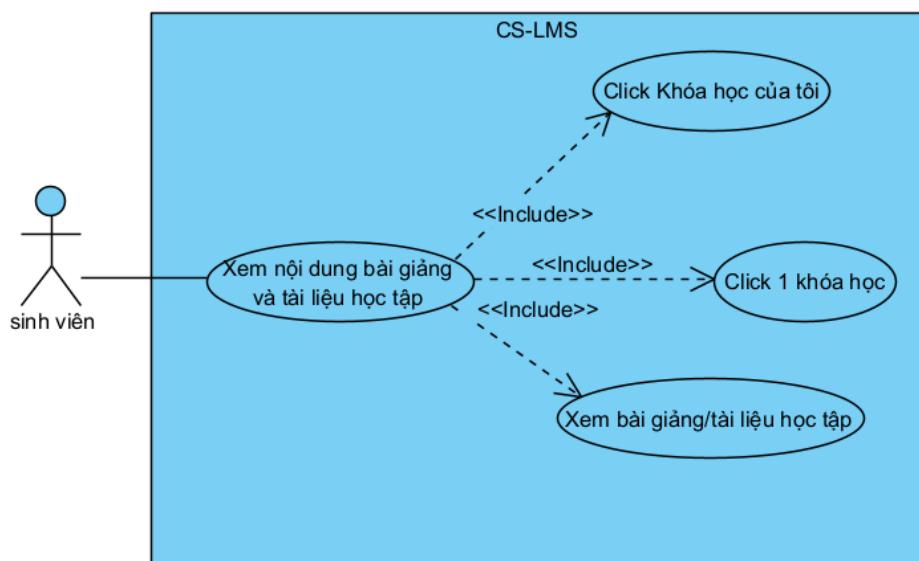


Hình 2.3a. Use case Xem danh sách các khóa học đã ghi danh

Chức năng	Xem danh sách các khóa học đã ghi danh
Tác nhân	Sinh viên
Tiền điều kiện	Đã đăng nhập vào hệ thống
Kịch bản chính	<ol style="list-style-type: none"> Sinh viên truy cập hệ thống Sinh viên chọn mục “Khóa học của tôi” trên giao diện chính. Hệ thống truy xuất danh sách các khóa học mà người dùng đã ghi danh. Hệ thống hiển thị danh sách khóa học bao gồm: <ul style="list-style-type: none"> Tên khóa học Mô tả ngắn Giảng viên phụ trách Trạng thái khóa học Người dùng chọn một khóa học để xem nội dung chi tiết.
Ngoại lệ	<ol style="list-style-type: none"> Người dùng chưa ghi danh khóa học nào → Hệ thống hiển thị thông báo: “Bạn chưa ghi danh khóa học nào.” Không thể tải danh sách khóa học → Hệ thống hiển thị thông báo: “Không thể tải dữ liệu khóa học. Vui lòng thử lại sau.”

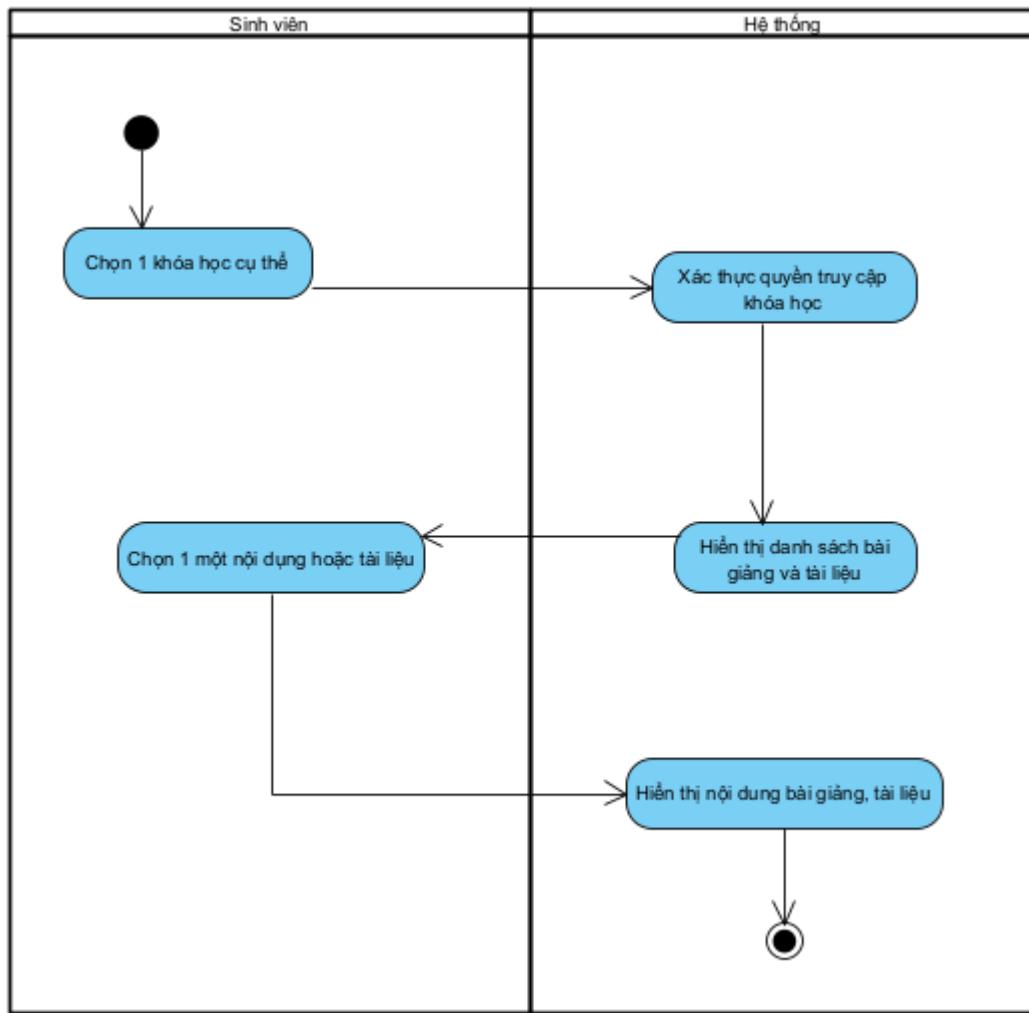


Hình 2.3b. Sơ đồ Activity của Chức năng Xem khóa học đã ghi danh
B, Xem bài giảng và tài liệu học tập

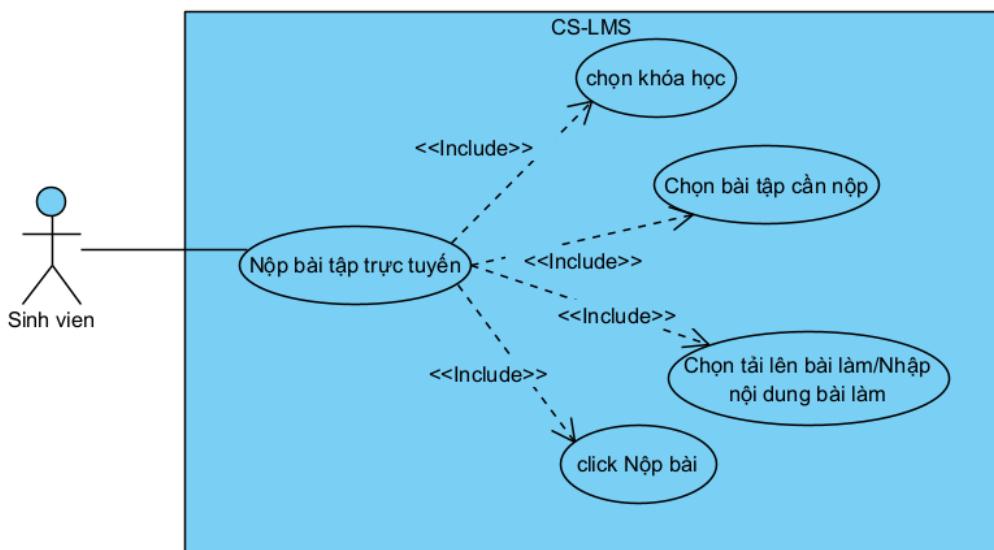


Hình 2.4a. Use case Xem bài giảng và tài liệu học tập

Chức năng	Xem bài giảng và tài liệu học tập
Tác nhân	Sinh viên
Tiền điều kiện	Đã đăng nhập vào hệ thống Đã ghi danh vào khóa học
Kịch bản chính	<ol style="list-style-type: none"> 1. Sinh viên truy cập hệ thống 2. Sinh viên chọn mục “Khóa học của tôi” trên giao diện chính. 3. Hệ thống truy xuất danh sách các khóa học mà người dùng đã ghi danh 4. Sinh viên chọn một khóa học đã ghi danh 5. Hệ thống hiển thị danh sách các chủ đề / tuần học / chương học 6. Người dùng chọn một bài giảng hoặc tài liệu học tập. Hệ thống hiển thị nội dung tương ứng, bao gồm: <ul style="list-style-type: none"> • Video bài giảng • Tài liệu học tập (PDF, Word, Slide, Link...) • Mô tả nội dung bài học
Ngoại lệ	<ol style="list-style-type: none"> 1. Trường hợp 1: Người dùng chưa ghi danh khóa học → Hệ thống hiển thị thông báo: “Bạn chưa được ghi danh vào khóa học này.” 2. Trường hợp 2: Tài liệu không tồn tại hoặc đã bị xóa → Hệ thống hiển thị thông báo: “Tài liệu hiện không khả dụng.” 3. Trường hợp 3: Người dùng không có quyền truy cập tài liệu → Hệ thống hiển thị thông báo: “Bạn không có quyền xem nội dung này.”

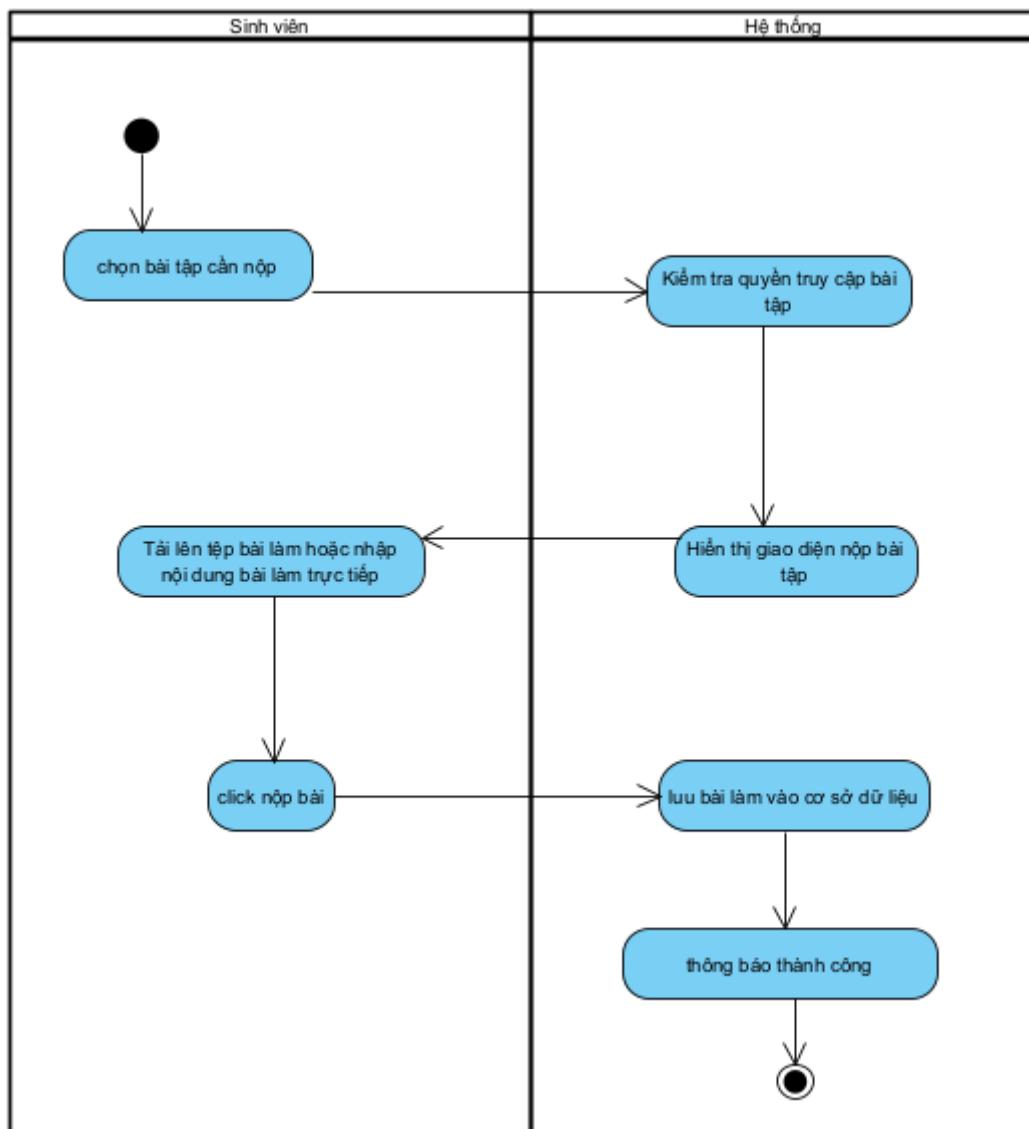


Hình 2.4b. Sơ đồ Activity của chức năng Xem bài giảng và bài tập
C, Chức năng nộp bài trực tuyến



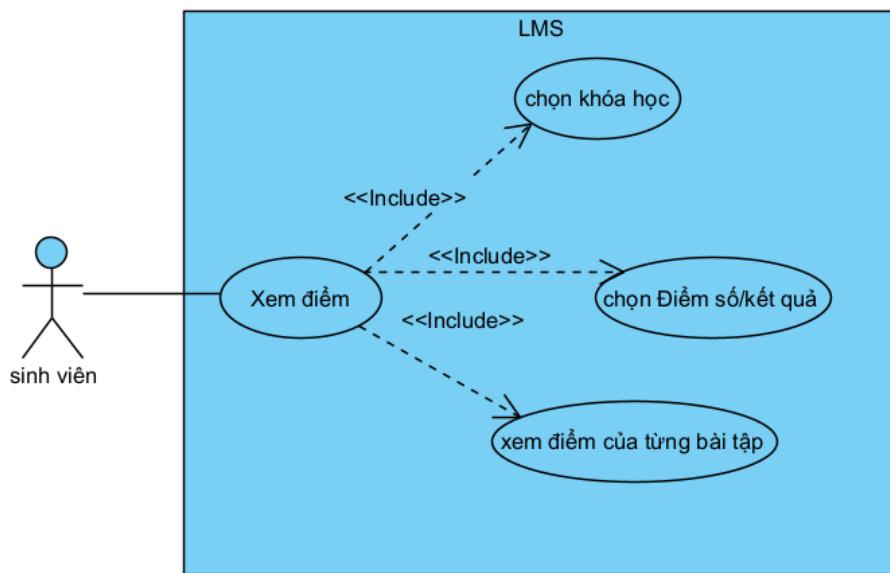
Hình 2.5a. Use case Chức năng nộp bài trực tuyến

Chức năng	Nộp bài tập trực tuyến
Tác nhân	Sinh viên
Tiền điều kiện	Đã đăng nhập vào hệ thống Sinh viên đã ghi danh vào khóa học Bài tập đã được giảng viên tạo và đang trong thời gian cho phép nộp
Kịch bản chính	<ol style="list-style-type: none"> 1. Người dùng truy cập hệ thống 2. Người dùng chọn “Khóa học của tôi” 3. Người dùng chọn khóa học tương ứng 4. Người dùng chọn bài tập cần nộp 5. Hệ thống hiển thị thông tin bài tập: <ul style="list-style-type: none"> • Tên bài tập • Mô tả yêu cầu • Hạn nộp • Hình thức nộp (tải file / nhập nội dung trực tiếp) 6. Người dùng chọn Tải lên bài làm hoặc Nhập nội dung bài làm 7. Người dùng xác nhận Nộp bài 8. Hệ thống lưu bài nộp và hiển thị thông báo: “Nộp bài thành công”
Ngoại lệ	<ol style="list-style-type: none"> 1. Trường hợp 1: Nộp bài sau thời hạn → Hệ thống hiển thị thông báo: “Đã quá hạn nộp bài. Bạn không thể nộp bài.” 2. Trường hợp 2: Định dạng tệp không hợp lệ → Hệ thống hiển thị thông báo: “Định dạng tệp không được hỗ trợ.” 3. Trường hợp 3: Kích thước tệp vượt quá giới hạn → Hệ thống hiển thị thông báo: “Dung lượng tệp vượt quá giới hạn cho phép.” 4. Trường hợp 4: Người dùng chưa ghi danh khóa học → Hệ thống hiển thị thông báo: “Bạn chưa được ghi danh vào khóa học này.”



Hình 2.5b. Sơ đồ Activity của Chức năng nộp bài

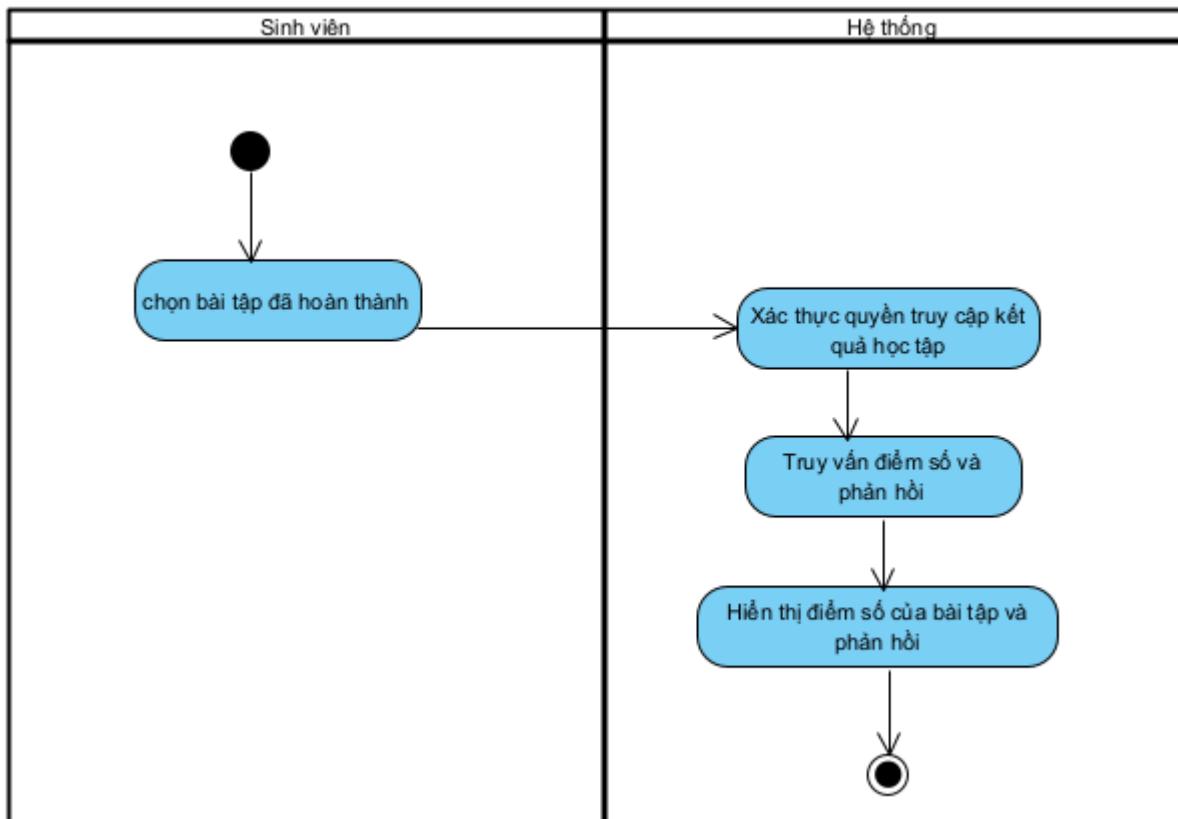
D, Chức năng xem điểm



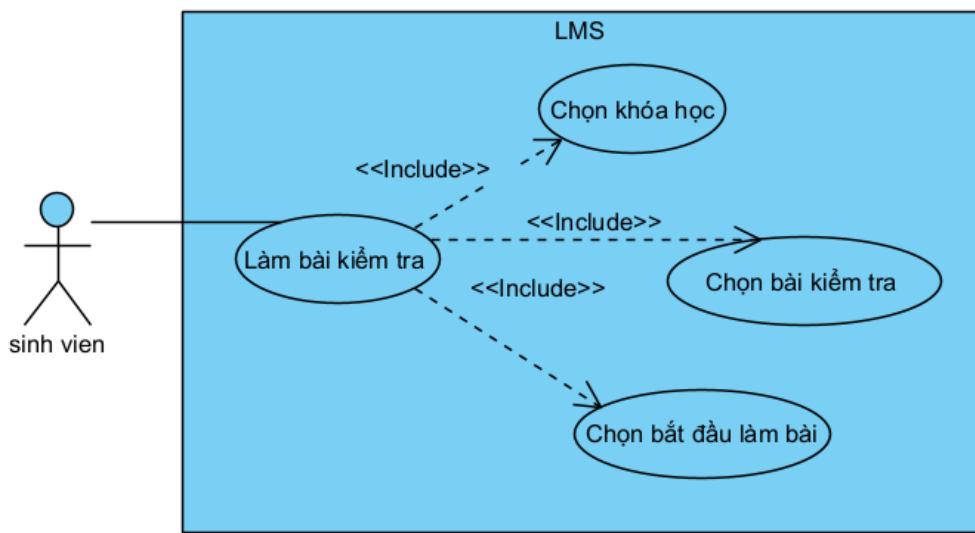
Hình 2.6a. Use case Chức năng Xem điểm số và phản hồi từ giảng viên

Chức năng	Xem điểm số và phản hồi từ giảng viên
Tác nhân	Sinh viên
Tiền điều kiện	Người dùng đã đăng nhập vào hệ thống Người dùng đã ghi danh vào khóa học Bài tập hoặc bài kiểm tra đã được giảng viên chấm điểm
Kịch bản chính	<ol style="list-style-type: none"> Người dùng truy cập hệ thống Người dùng chọn “Khóa học của tôi” Người dùng chọn khóa học cần xem điểm Người dùng chọn mục “Điểm số” hoặc “Kết quả học tập” Hệ thống hiển thị danh sách các bài đã nộp, bao gồm: <ul style="list-style-type: none"> Tên bài tập / bài kiểm tra Điểm số đạt được Nhận xét / phản hồi từ giảng viên Thời gian chấm điểm Người dùng chọn một bài cụ thể để xem chi tiết phản hồi (nếu có)
Ngoại lệ	<ol style="list-style-type: none"> Trường hợp 1: Bài tập chưa được chấm điểm → Hệ thống hiển thị thông báo: “Bài làm đang được chấm. Vui lòng quay lại sau.”

	<p>2. Trường hợp 2: Người dùng chưa ghi danh khóa học → Hệ thống hiển thị thông báo: “Bạn chưa được ghi danh vào khóa học này.”</p> <p>3. Trường hợp 3: Lỗi tải dữ liệu điểm số → Hệ thống hiển thị thông báo: “Không thể tải dữ liệu điểm. Vui lòng thử lại sau.”</p>
--	--



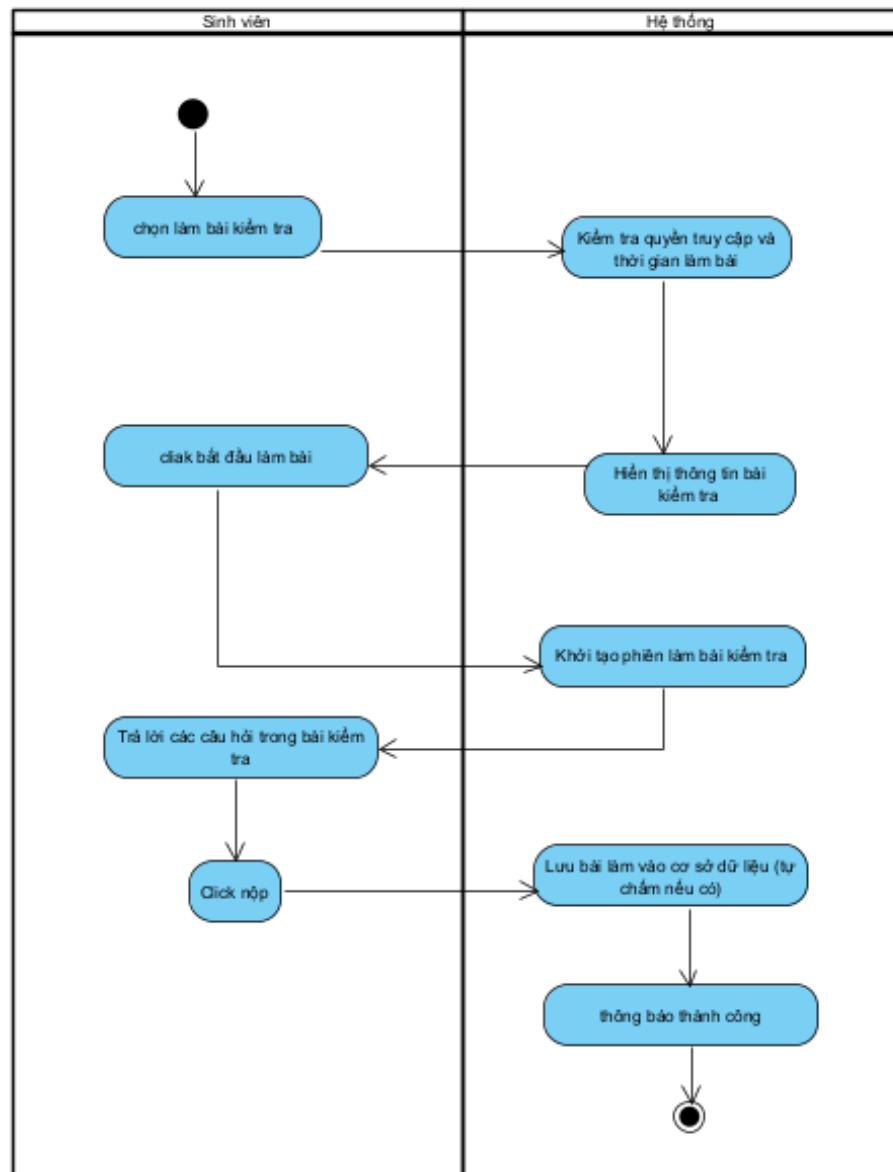
Hình 2.6b. Sơ đồ Activity của Chức năng Xem điểm số và phản hồi của giảng viên E, Tham gia làm bài kiểm tra (quiz)



Hình 2.7a. Use case Tham gia làm bài kiểm tra (quiz)

Chức năng	Tham gia làm bài kiểm tra (Quiz)
Tác nhân	Người dùng (Sinh viên)
Tiền điều kiện	Người dùng đã đăng nhập vào hệ thống Người dùng đã ghi danh vào khóa học Bài kiểm tra đang trong thời gian cho phép làm bài
Kịch bản chính	<ol style="list-style-type: none"> Người dùng truy cập hệ thống Người dùng chọn “Khóa học của tôi” Người dùng chọn khóa học có bài kiểm tra Người dùng chọn bài kiểm tra (Quiz) cần thực hiện Hệ thống hiển thị thông tin bài kiểm tra gồm: <ul style="list-style-type: none"> Tên bài kiểm tra Thời gian làm bài Số câu hỏi Số lần được phép làm Thời gian mở và đóng bài kiểm tra Người dùng nhấn “Bắt đầu làm bài” Hệ thống hiển thị các câu hỏi để người dùng trả lời Người dùng hoàn thành bài làm và nhấn “Nộp bài” Hệ thống lưu kết quả và hiển thị thông báo: Nộp bài thành công

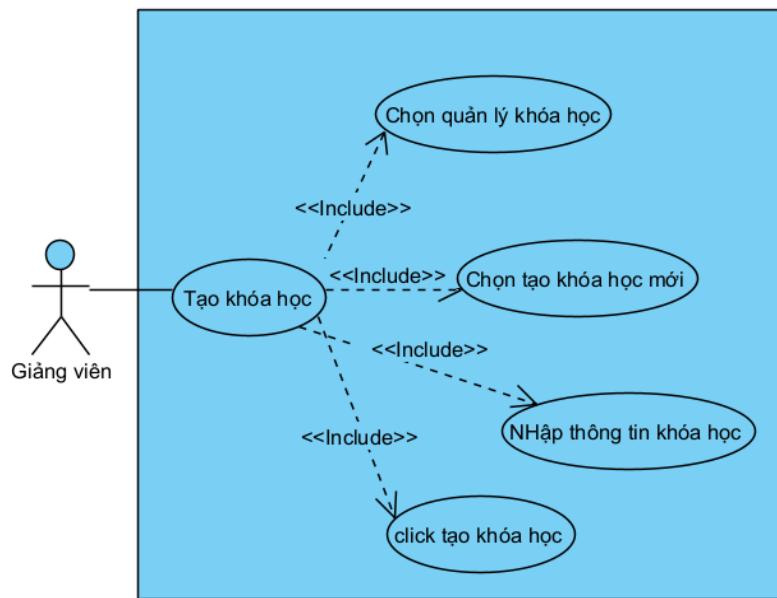
Ngoại lệ	<ol style="list-style-type: none"> 1. Trường hợp 1: Người dùng truy cập khi chưa đến thời gian làm bài → Hệ thống hiển thị thông báo: “Bài kiểm tra chưa được mở.” 2. Trường hợp 2: Người dùng truy cập khi đã hết thời gian làm bài → Hệ thống hiển thị thông báo: “Bài kiểm tra đã kết thúc.” 3. Trường hợp 3: Hết thời gian làm bài → Hệ thống tự động nộp bài và hiển thị thông báo: “Đã hết thời gian làm bài. Bài làm được nộp tự động.” 4. Trường hợp 4: Mất kết nối trong quá trình làm bài → Hệ thống hiển thị thông báo: “Mất kết nối mạng. Vui lòng kiểm tra lại.”
-----------------	---



Hình 2.7b. Sơ đồ Activity của Chức năng Tham gia làm bài kiểm tra

2.2.3. Chức năng dành cho giảng viên

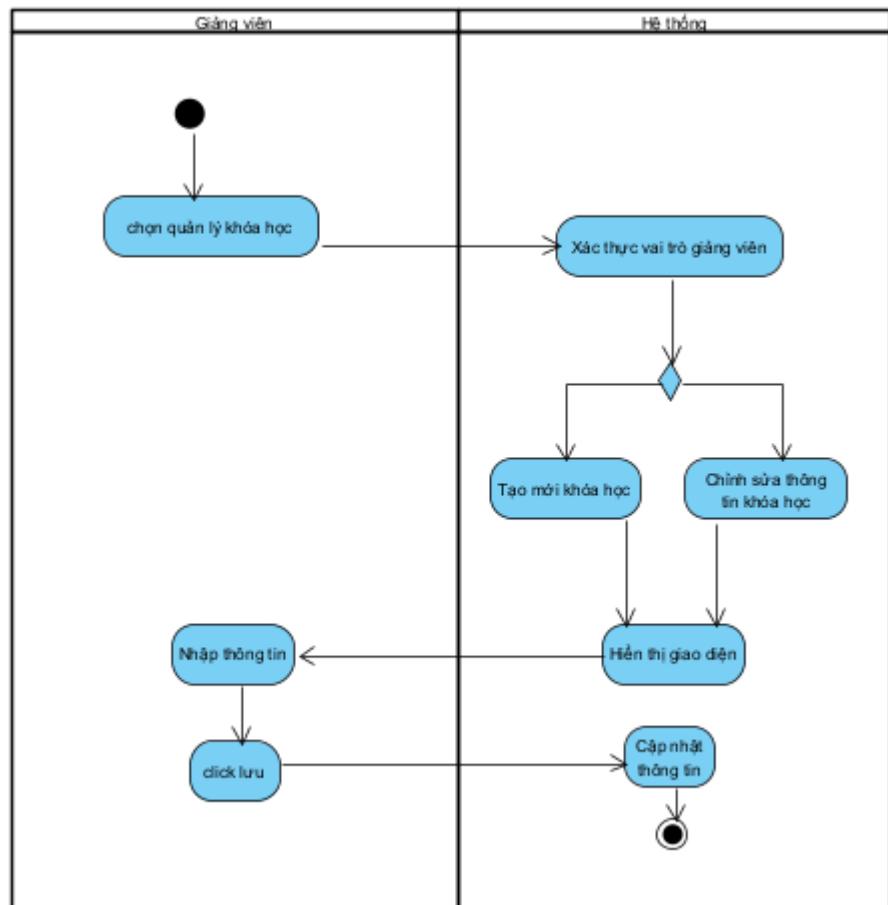
A, Tạo mới khóa học



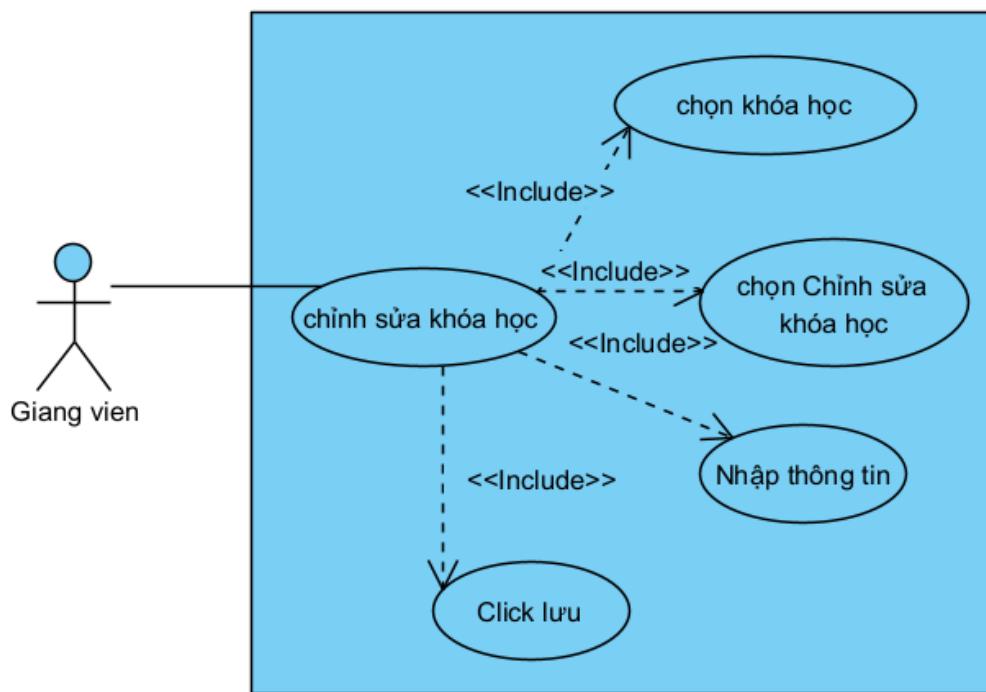
Hình 2.8a. Use case Tạo mới khóa học

Chức năng	Tạo mới khóa học
Tác nhân	Giảng viên
Tiền điều kiện	Giảng viên đã đăng nhập vào hệ thống
Kịch bản chính	<ol style="list-style-type: none"> Giảng viên truy cập hệ thống Giảng viên chọn mục “Quản lý khóa học” Giảng viên chọn chức năng “Tạo khóa học mới” Hệ thống hiển thị form tạo khóa học gồm các thông tin: <ul style="list-style-type: none"> Tên khóa học Mô tả khóa học Danh mục môn học Thời gian bắt đầu Thời gian kết thúc Trạng thái khóa học (mở/đóng) Giảng viên nhập đầy đủ thông tin cần thiết Giảng viên nhấn nút “Tạo khóa học” Hệ thống kiểm tra tính hợp lệ của dữ liệu Hệ thống lưu thông tin khóa học vào cơ sở dữ liệu Hệ thống hiển thị thông báo: “Tạo khóa học thành công.”

Ngoại lệ	<ol style="list-style-type: none"> 1. Trường hợp 1: Thiếu thông tin bắt buộc → Hệ thống hiển thị thông báo: “Vui lòng nhập đầy đủ thông tin bắt buộc.” 2. Trường hợp 2: Thông tin không hợp lệ → Hệ thống hiển thị thông báo: “Dữ liệu không hợp lệ. Vui lòng kiểm tra lại.” 3. Trường hợp 3: Giảng viên không có quyền tạo khóa học → Hệ thống hiển thị thông báo: “Bạn không có quyền tạo khóa học.”
-----------------	---



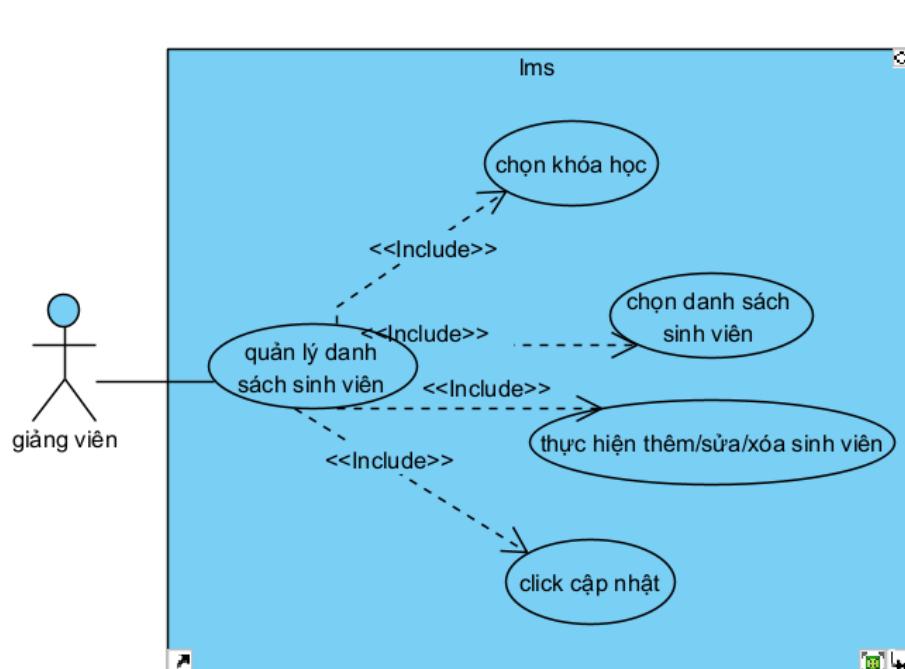
Hình 2.8b. Sơ đồ Activity của Chức năng Tạo mới khóa học
B, Chức năng chỉnh sửa khóa học

Hình 2.9a. Use case *Chỉnh sửa thông tin khóa học*

Chức năng	Chỉnh sửa thông tin khóa học
Tác nhân	Giảng viên
Tiền điều kiện	Giảng viên đã đăng nhập vào hệ thống Khóa học đã được tạo trước đó
Kịch bản chính	<ol style="list-style-type: none"> Giảng viên truy cập hệ thống Giảng viên chọn mục “Quản lý khóa học” Giảng viên chọn khóa học cần chỉnh sửa Giảng viên chọn chức năng “Chỉnh sửa khóa học” Hệ thống hiển thị form chỉnh sửa với các thông tin hiện tại của khóa học: <ul style="list-style-type: none"> Tên khóa học Mô tả khóa học Danh mục môn học Thời gian bắt đầu – kết thúc Trạng thái khóa học (mở / đóng) Giảng viên chỉnh sửa thông tin cần thay đổi Giảng viên nhấn nút “Lưu thay đổi” Hệ thống kiểm tra dữ liệu và cập nhật thông tin khóa học

	9. Hệ thống hiển thị thông báo: “Cập nhật thông tin khóa học thành công.”
Ngoại lệ	<ol style="list-style-type: none"> 1. Trường hợp 1: Thiếu thông tin bắt buộc → Hệ thống hiển thị thông báo: “Vui lòng nhập đầy đủ thông tin bắt buộc.” 2. Trường hợp 2: Dữ liệu không hợp lệ → Hệ thống hiển thị thông báo: “Dữ liệu không hợp lệ. Vui lòng kiểm tra lại.”

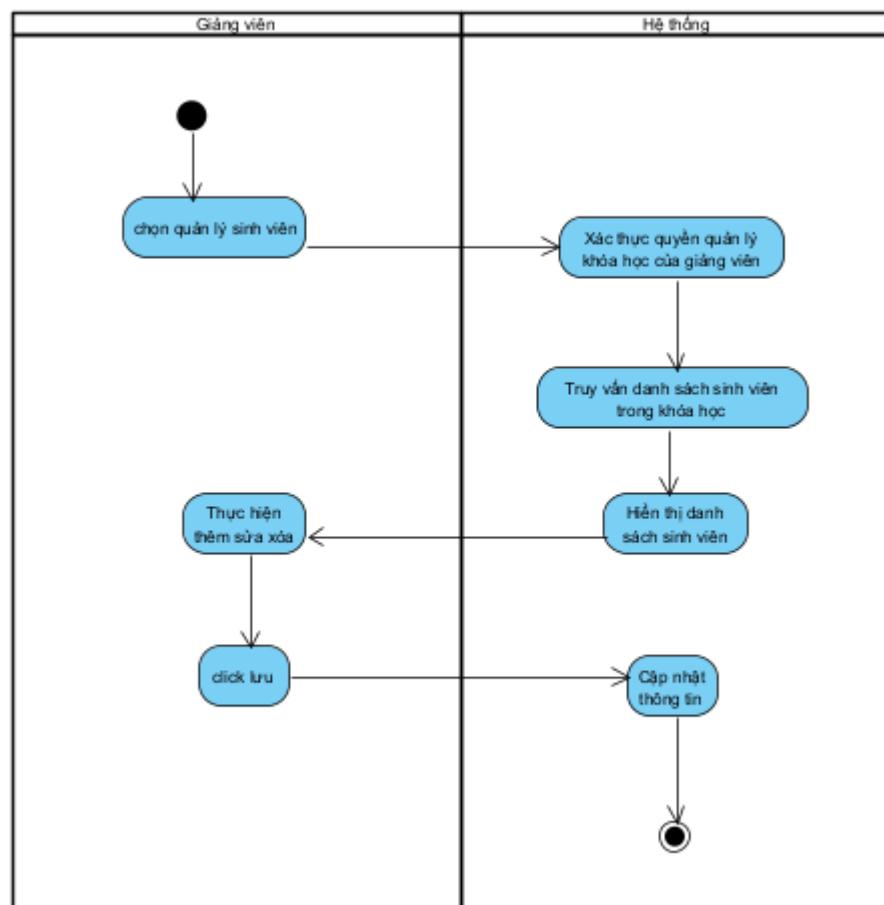
C, Chức năng quản lý danh sách sinh viên



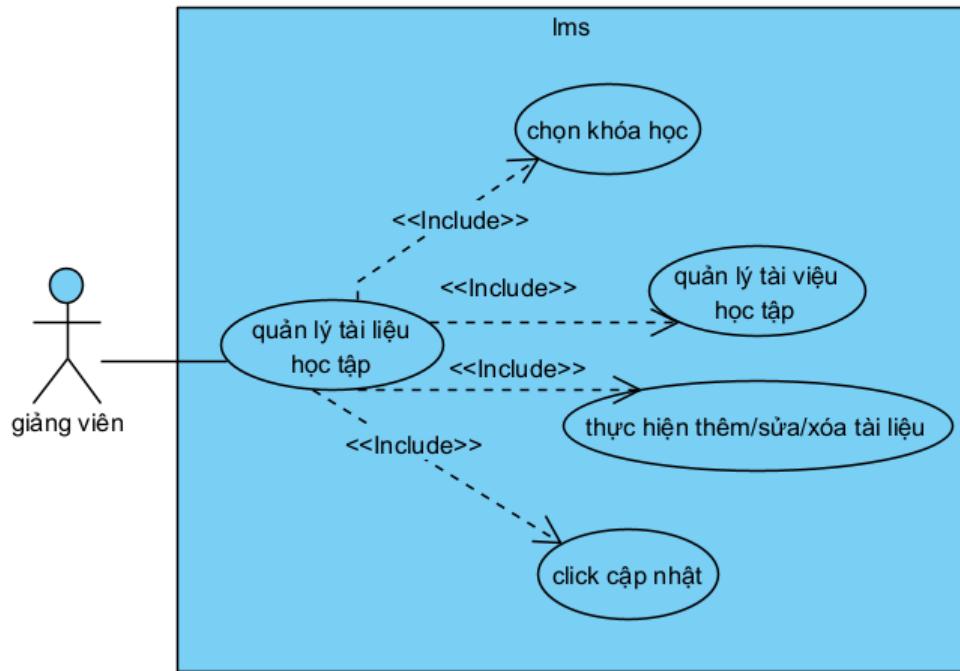
Hình 2.10a. Use case Chức năng quản lý danh sách sinh viên

Chức năng	Quản lý danh sách sinh viên trong khóa học
Tác nhân	Giảng viên
Tiền điều kiện	Giảng viên đã đăng nhập vào hệ thống Khóa học đã được tạo và đang hoạt động
Kịch bản chính	<ol style="list-style-type: none"> 1. Giảng viên truy cập hệ thống 2. Giảng viên chọn mục “Quản lý khóa học” 3. Giảng viên chọn khóa học cần quản lý 4. Giảng viên chọn chức năng “Danh sách sinh viên” 5. Hệ thống hiển thị danh sách sinh viên đã ghi danh, bao gồm: <ul style="list-style-type: none"> • Mã sinh viên • Họ và tên

	<ul style="list-style-type: none"> • Email • Trạng thái tham gia khóa học <p>6. Giảng viên thực hiện các thao tác quản lý:</p> <ul style="list-style-type: none"> • Thêm sinh viên vào khóa học • Xóa sinh viên khỏi khóa học • Xem thông tin chi tiết sinh viên <p>7. Hệ thống cập nhật danh sách sinh viên tương ứng</p> <p>8. Hệ thống hiển thị thông báo: “Cập nhật danh sách sinh viên thành công.”</p>
Ngoại lệ	<ol style="list-style-type: none"> 1. Trường hợp 1: Sinh viên đã tồn tại trong khóa học → Hệ thống hiển thị thông báo: “Sinh viên đã được ghi danh trong khóa học.” 2. Trường hợp 2: Sinh viên không tồn tại trong hệ thống → Hệ thống hiển thị thông báo: “Không tìm thấy sinh viên trong hệ thống.”



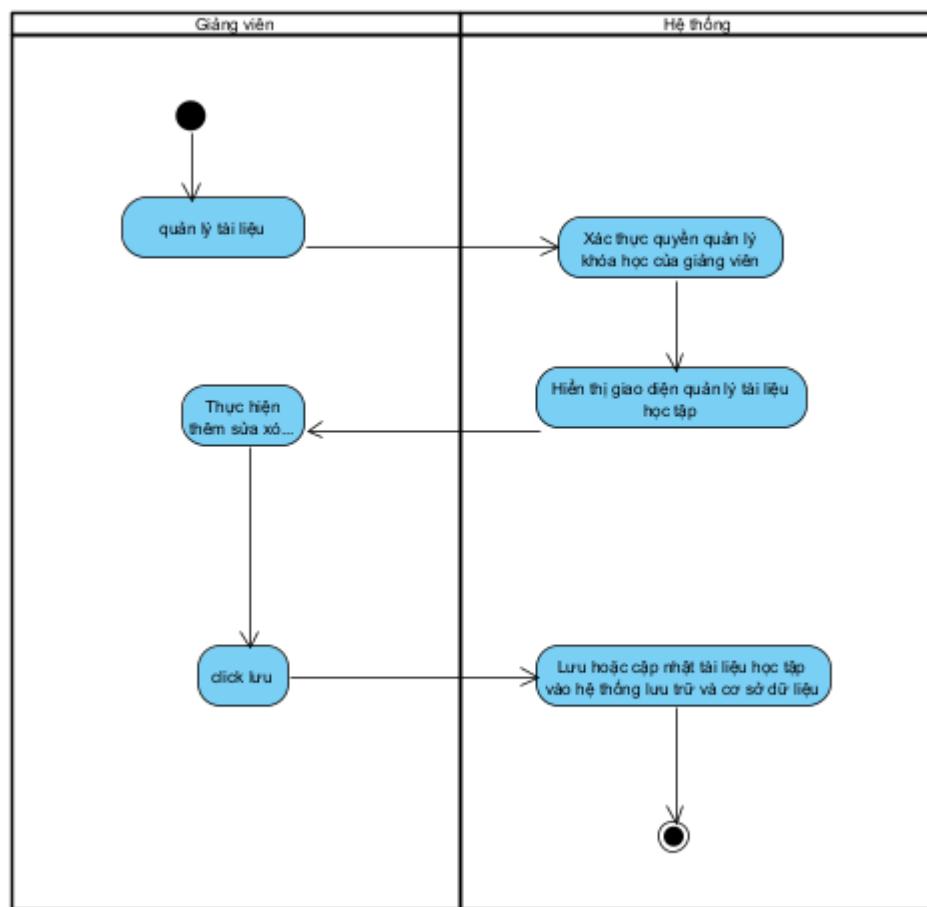
Hình 2.10b. Sơ đồ Activity của Chức năng Quản lý sinh viên
D, Quản lý tài liệu học tập



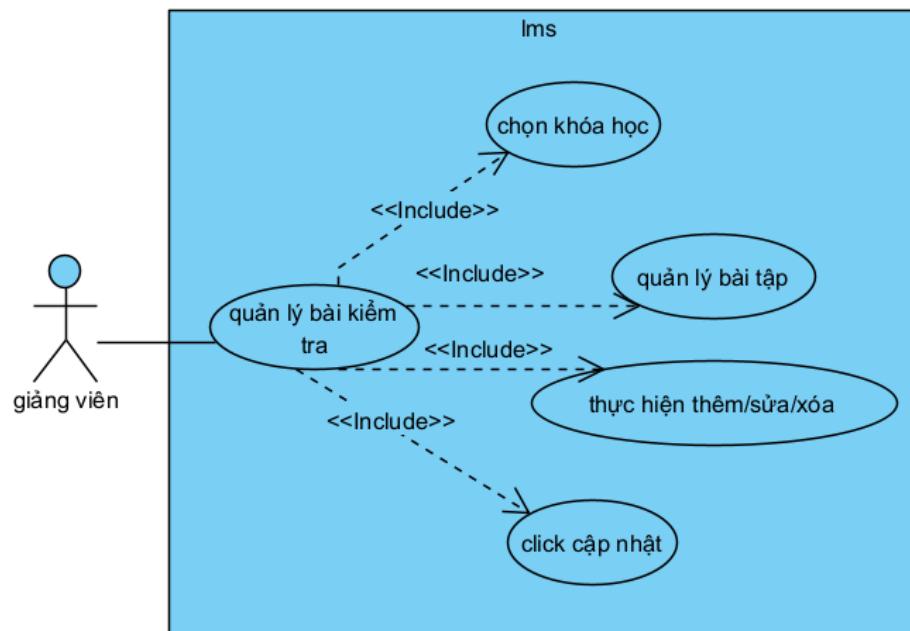
Hình 2.11a. Use case Quản lý tài liệu học tập

Chức năng	Quản lý tài liệu học tập
Tác nhân	Giảng viên
Tiền điều kiện	Giảng viên đã đăng nhập vào hệ thống Khóa học đã được tạo và đang hoạt động
Kịch bản chính	<ol style="list-style-type: none"> Giảng viên truy cập hệ thống Giảng viên chọn mục “Quản lý khóa học” Giảng viên chọn khóa học cần quản lý Giảng viên chọn chức năng “Quản lý tài liệu học tập” Hệ thống hiển thị danh sách tài liệu hiện có, bao gồm: <ul style="list-style-type: none"> Tên tài liệu Loại tài liệu (PDF, Word, Slide, Video, Link, ...) Ngày tải lên Trạng thái hiển thị Giảng viên thực hiện các thao tác: <ul style="list-style-type: none"> Thêm tài liệu mới Chỉnh sửa thông tin tài liệu Xóa tài liệu Bật / tắt hiển thị tài liệu cho sinh viên Hệ thống cập nhật dữ liệu tài liệu học tập

	8. Hệ thống hiển thị thông báo: “Cập nhật tài liệu học tập thành công.”
Ngoại lệ	<ol style="list-style-type: none"> 1. Trường hợp 1: Tệp tải lên không đúng định dạng → Hệ thống hiển thị thông báo: “Định dạng tệp không được hỗ trợ.” 2. Trường hợp 2: Dung lượng tệp vượt quá giới hạn → Hệ thống hiển thị thông báo: “Dung lượng tệp vượt quá giới hạn cho phép.”



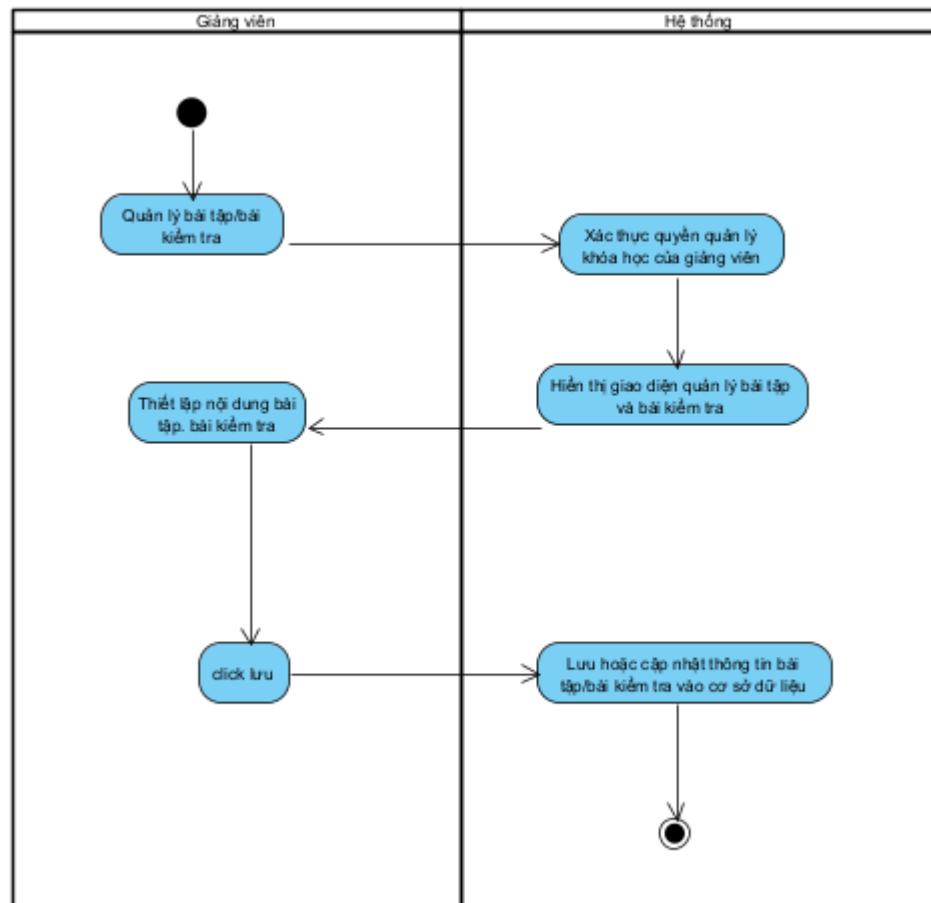
Hình 2.11b. Sơ đồ Activity của Chức năng Quản lý tài liệu học tập E, Quản lý bài tập



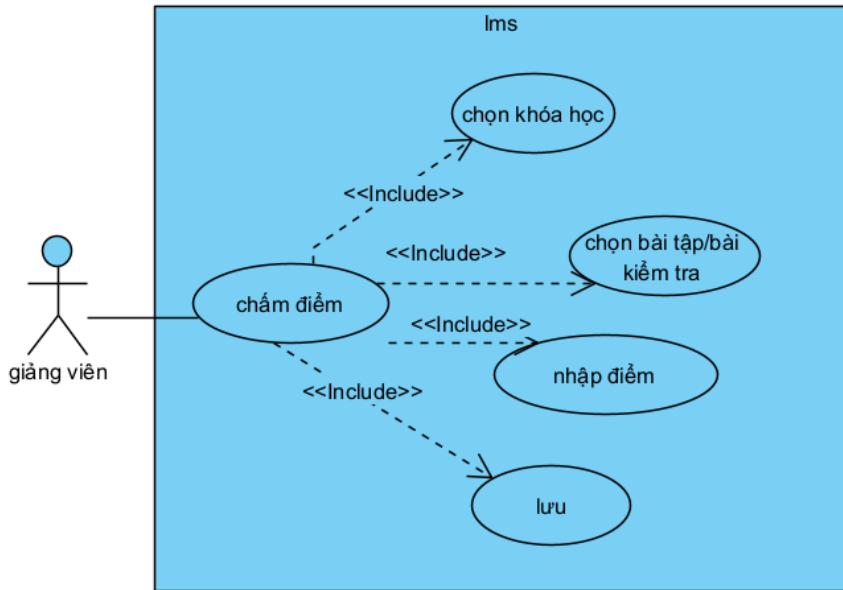
Hình 2.12a. Use case Quản lý học tập

Chức năng	Quản lý học tập
Tác nhân	Giảng viên
Tiền điều kiện	Giảng viên đã đăng nhập vào hệ thống Khóa học đã được tạo và đang hoạt động
Kịch bản chính	<ol style="list-style-type: none"> Giảng viên truy cập hệ thống Giảng viên chọn mục “Quản lý khóa học” Giảng viên chọn khóa học cần quản lý Giảng viên chọn chức năng “Quản lý bài tập / bài kiểm tra” Hệ thống hiển thị danh sách bài tập và bài kiểm tra hiện có, bao gồm: <ul style="list-style-type: none"> Tên bài Loại (Bài tập/Bài kiểm tra) Thời gian mở – đóng Trạng thái Giảng viên thực hiện các thao tác: <ul style="list-style-type: none"> Tạo bài tập/bài kiểm tra mới Chỉnh sửa nội dung bài tập/bài kiểm tra Xóa bài tập/bài kiểm tra Thiết lập thời gian làm bài, số lần làm bài Hệ thống lưu thay đổi và cập nhật dữ liệu

	8. Hệ thống hiển thị thông báo: “Cập nhật bài tập/bài kiểm tra thành công.”
Ngoại lệ	<ol style="list-style-type: none"> Trường hợp 1: Thiếu thông tin bắt buộc khi tạo bài → Hệ thống hiển thị thông báo: “Vui lòng nhập đầy đủ thông tin bài tập / bài kiểm tra” Trường hợp 2: Thời gian cấu hình không hợp lệ → Hệ thống hiển thị thông báo: “Thời gian mở – đóng không hợp lệ.



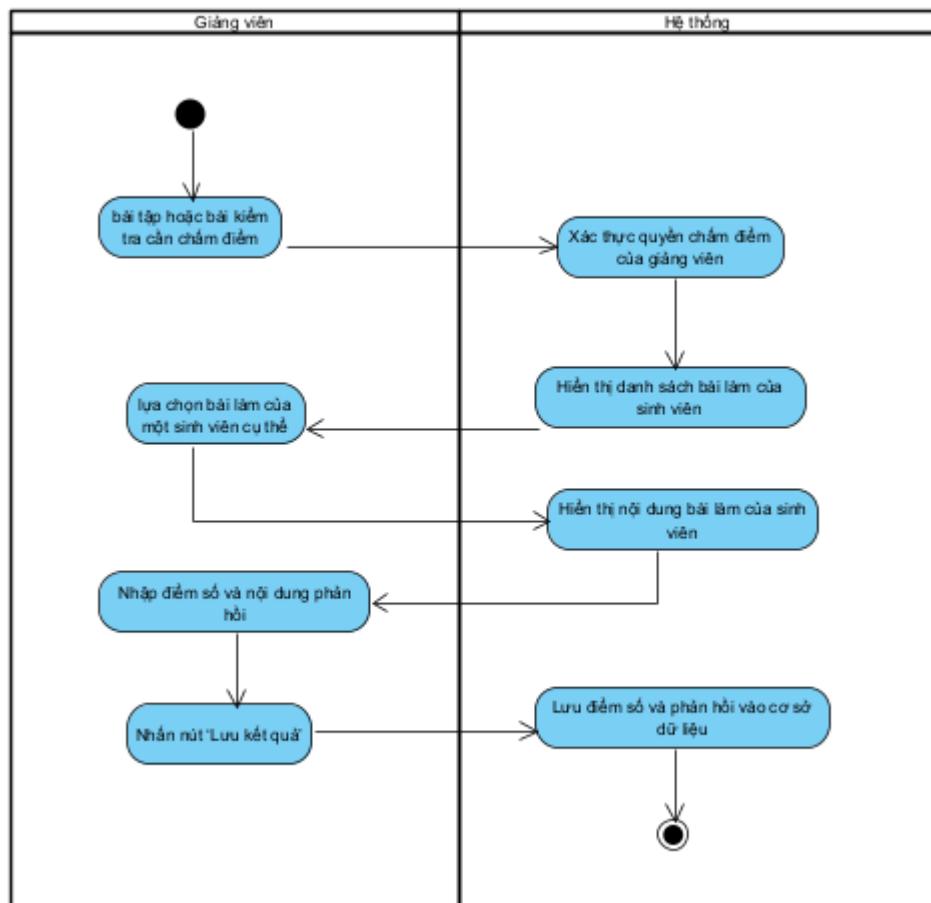
Hình 2.12b. Sơ đồ Activity của Chức năng Quản lý bài tập F, Chức năng chấm điểm



Hình 2.13a. Use case Chức năng chấm điểm

Chức năng	Chấm điểm bài tập và bài kiểm tra
Tác nhân	Giảng viên
Tiền điều kiện	Giảng viên đã đăng nhập vào hệ thống Giảng viên có quyền quản lý khóa học Bài tập / bài kiểm tra đã được sinh viên nộp
Kịch bản chính	<ol style="list-style-type: none"> Giảng viên truy cập hệ thống Giảng viên chọn “Quản lý khóa học” Giảng viên chọn khóa học cần chấm điểm Giảng viên chọn “Bài tập / Bài kiểm tra” cần chấm Hệ thống hiển thị danh sách sinh viên đã nộp bài, bao gồm: <ul style="list-style-type: none"> Họ và tên sinh viên Thời gian nộp bài Trạng thái bài nộp Giảng viên chọn một bài nộp để chấm điểm Hệ thống hiển thị nội dung bài làm của sinh viên Giảng viên nhập: <ul style="list-style-type: none"> Điểm số Nhận xét / phản hồi Giảng viên nhấn “Lưu kết quả” Hệ thống cập nhật điểm và phản hồi cho sinh viên Hệ thống hiển thị thông báo “Chấm điểm thành công.”

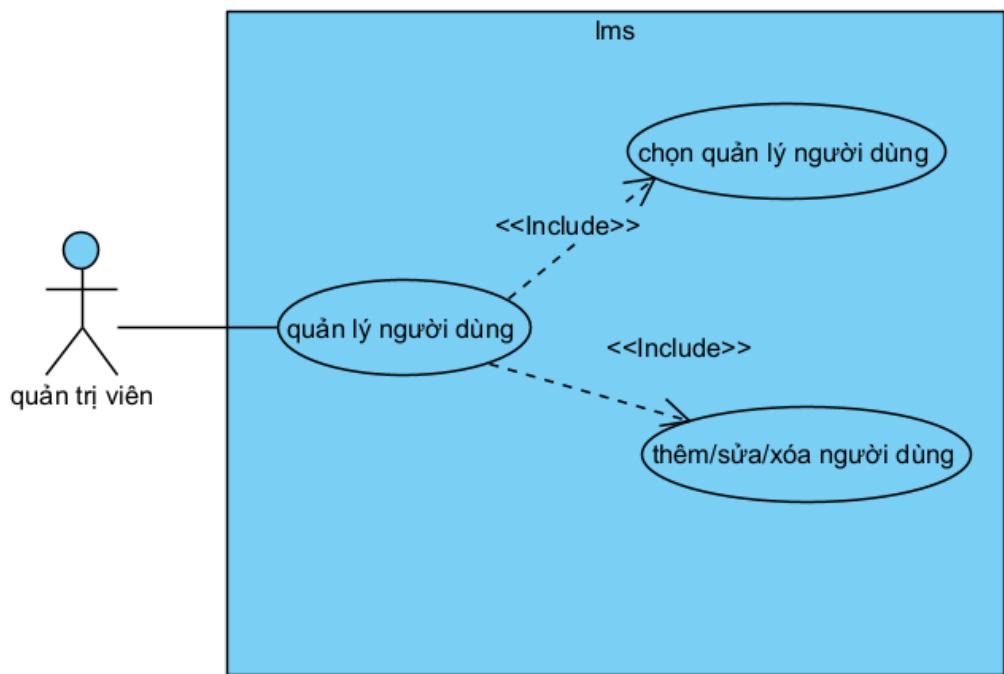
Ngoại lệ	<ol style="list-style-type: none"> 1. Trường hợp 1: Bài nộp chưa tồn tại → Hệ thống hiển thị thông báo: “Không tìm thấy bài nộp của sinh viên.” 2. Trường hợp 2: Điểm nhập không hợp lệ → Hệ thống hiển thị thông báo: “Điểm số không hợp lệ.” 3. Trường hợp 3: Giảng viên không có quyền chấm bài → Hệ thống hiển thị thông báo: “Bạn không có quyền chấm bài này.”
-----------------	---



Hình 2.13b. Sơ đồ Activity của Chức năng Chấm điểm

2.2.4. Chức năng cho quản trị viên

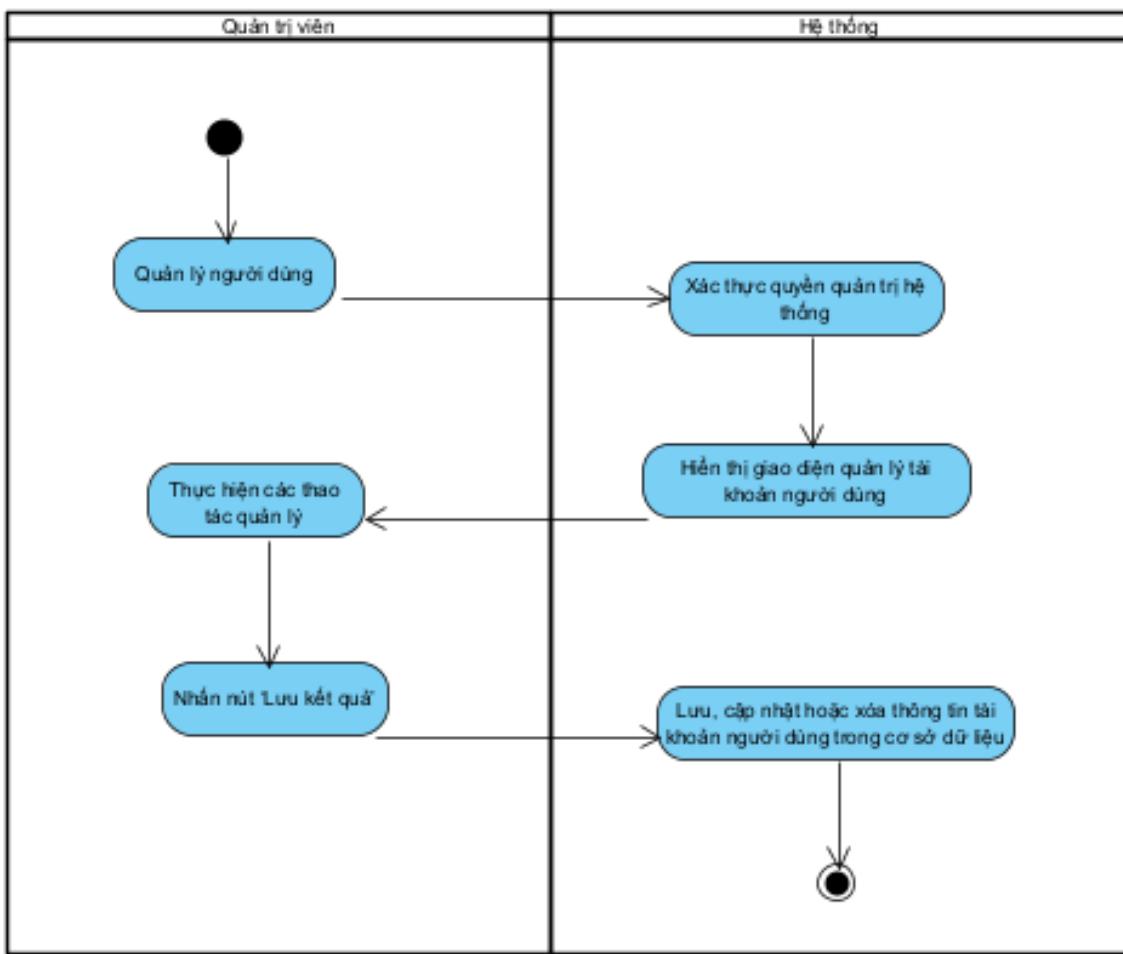
A, Quản lý người dùng



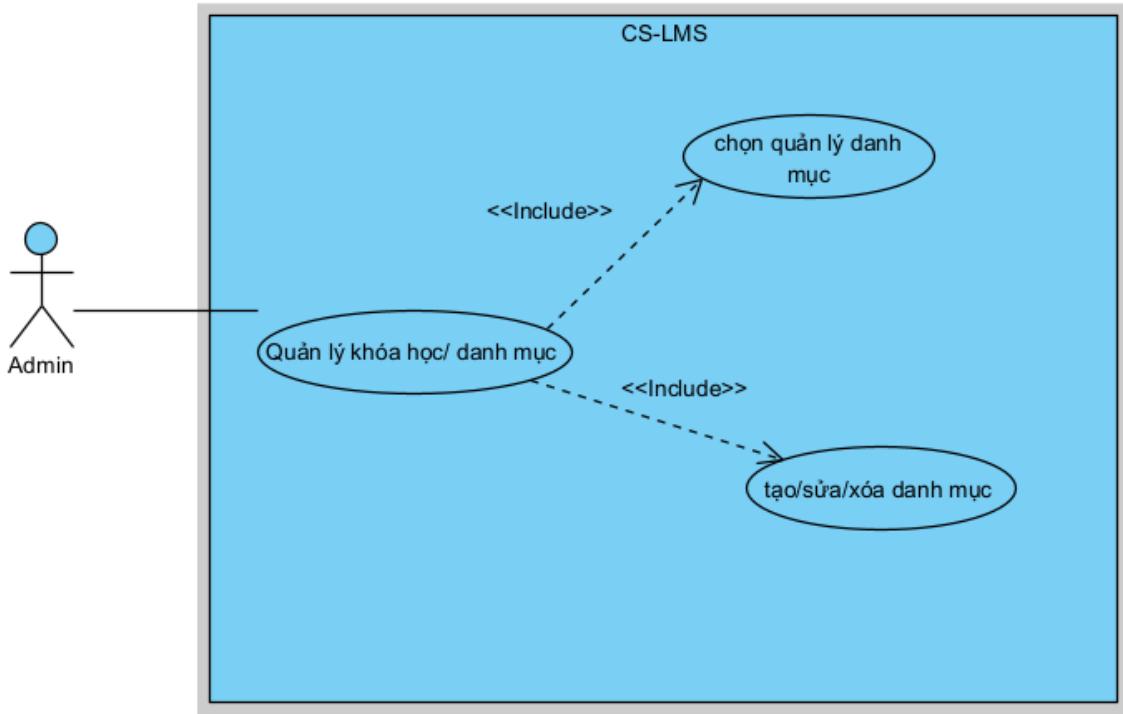
Hình 2.14a. Use case Quản lý người dùng

Chức năng	Quản lý người dùng
Tác nhân	Quản trị viên
Tiền điều kiện	Quản trị viên đã đăng nhập vào hệ thống
Kịch bản chính	<ol style="list-style-type: none"> Quản trị viên truy cập hệ thống Quản trị viên chọn mục “Quản lý người dùng” Hệ thống hiển thị danh sách người dùng, bao gồm: <ul style="list-style-type: none"> Tên người dùng Vai trò (Sinh viên / Giảng viên / Quản trị viên) Email Trạng thái tài khoản Quản trị viên thực hiện một trong các thao tác: <ul style="list-style-type: none"> Thêm tài khoản mới Chỉnh sửa thông tin tài khoản Xóa tài khoản người dùng
	Chi tiết thao tác a, Thêm tài khoản người dùng 1. Quản trị viên chọn “Thêm người dùng”

	<p>2. Hệ thống hiển thị form nhập thông tin:</p> <ul style="list-style-type: none"> o Họ và tên o Tên đăng nhập o Mật khẩu o Email o Vai trò (Sinh viên / Giảng viên / Quản trị viên) <p>3. Quản trị viên nhấn “Lưu”</p> <p>4. Hệ thống kiểm tra dữ liệu và tạo tài khoản mới “Tạo tài khoản thành công.”</p>
	<p>b, Chính sửa tài khoản người dùng</p> <ol style="list-style-type: none"> 1. Quản trị viên chọn người dùng cần chỉnh sửa 2. Hệ thống hiển thị thông tin chi tiết người dùng 3. Quản trị viên chỉnh sửa thông tin cần thiết 4. Nhấn “Cập nhật” 5. Hệ thống lưu thay đổi và hiển thị thông báo: “Cập nhật tài khoản thành công.”
	<p>c. Xóa tài khoản người dùng</p> <ol style="list-style-type: none"> 1. Quản trị viên chọn người dùng cần xóa 2. Hệ thống hiển thị hộp thoại xác nhận xóa 3. Quản trị viên xác nhận thao tác 4. Hệ thống xóa tài khoản khỏi hệ thống, Thông báo hiển thị: “Xóa tài khoản thành công.”
Ngoại lệ	<ol style="list-style-type: none"> 1. Trường hợp 1: Thiếu thông tin bắt buộc → Hệ thống hiển thị thông báo “Vui lòng nhập đầy đủ thông tin bắt buộc” 2. Trường hợp 2: Tài khoản đã tồn tại → Hệ thống hiển thị thông báo: “Tài khoản đã tồn tại trong hệ thống” 3. Trường hợp 3: Quản trị viên không có quyền thao tác → Hệ thống hiển thị thông báo: “Bạn không có quyền thực hiện chức năng này.”



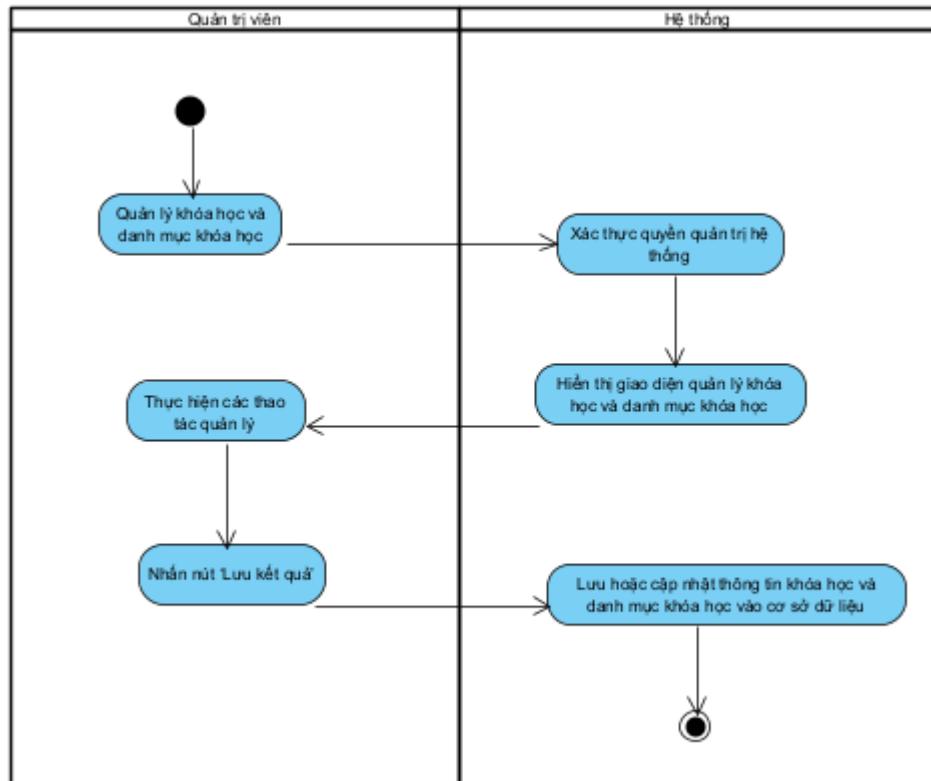
Hình 2.14b. Sơ đồ Activity của Chức năng Quản lý người dùng
B, Quản lý khóa học và danh mục khóa học



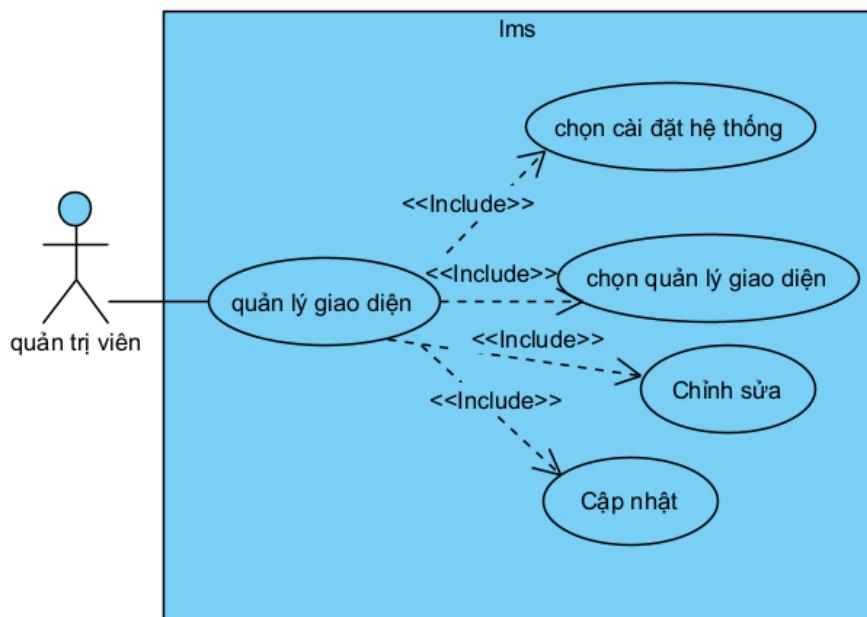
Hình 2.15a. Use case Quản lý khóa học và danh mục khóa học

Chức năng	Quản lý khóa học/danh mục
Tác nhân	Quản trị viên
Tiền điều kiện	Quản trị viên đã đăng nhập vào hệ thống Giảng viên có quyền quản lý khóa học/danh mục
Kịch bản chính	<ol style="list-style-type: none"> Quản trị viên truy cập hệ thống Quản trị viên chọn mục “Cài đặt hệ thống” Quản trị viên chọn “Quản lý khóa học/danh mục” Hệ thống hiển thị danh sách các danh mục khóa học hiện có, bao gồm: <ul style="list-style-type: none"> ○ Tên danh mục ○ Mô tả ○ Số lượng khóa học thuộc danh mục ○ Trạng thái hiển thị Quản trị viên thực hiện các thao tác: <ul style="list-style-type: none"> ○ Thêm danh mục mới ○ Chỉnh sửa thông tin danh mục ○ Xóa danh mục

	<ul style="list-style-type: none"> ○ Sắp xếp thứ tự danh mục 6. Hệ thống cập nhật dữ liệu danh mục khóa học 7. Hệ thống hiển thị thông báo: “Cập nhật danh mục khóa học thành công”
Ngoại lệ	<ol style="list-style-type: none"> 1. Trường hợp 1: Danh mục đã tồn tại → Hệ thống hiển thị thông báo: “Danh mục khóa học đã tồn tại.” 2. Trường hợp 2: Không thể xóa danh mục đang chứa khóa học → Hệ thống hiển thị thông báo: “Không thể xóa danh mục đang có khóa học.” 3. Trường hợp 3: Quản trị viên không có quyền thực hiện → Hệ thống hiển thị thông báo: “Bạn không có quyền quản lý danh mục khóa học.”



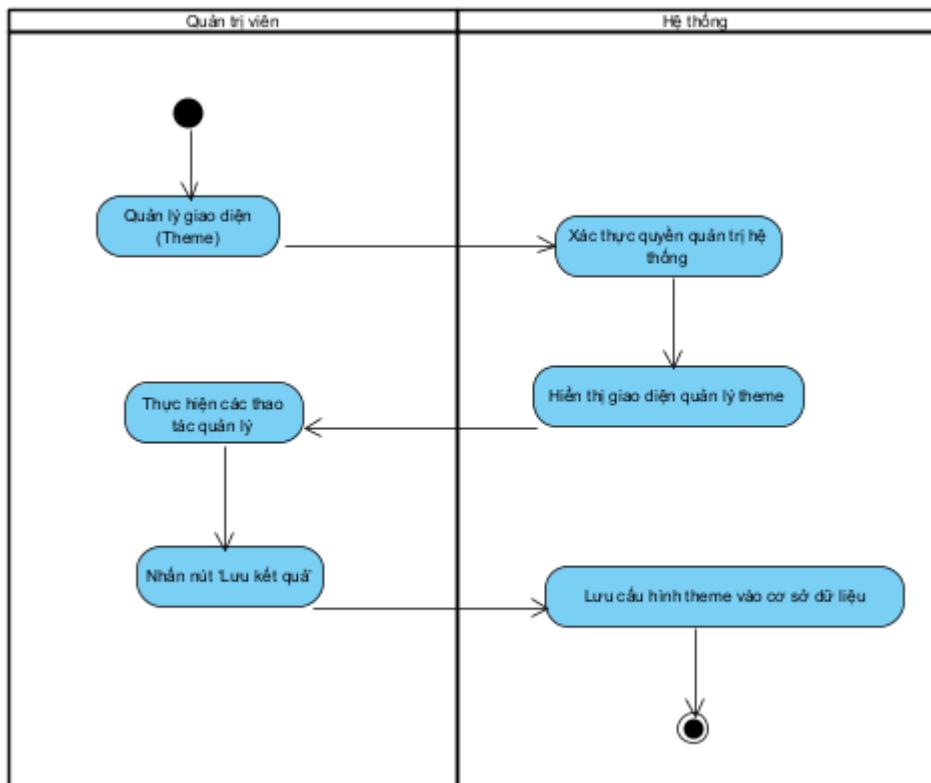
Hình 2.15b. Sơ đồ Activity của Chức năng Quản lý khóa học
C, Quản lý giao diện



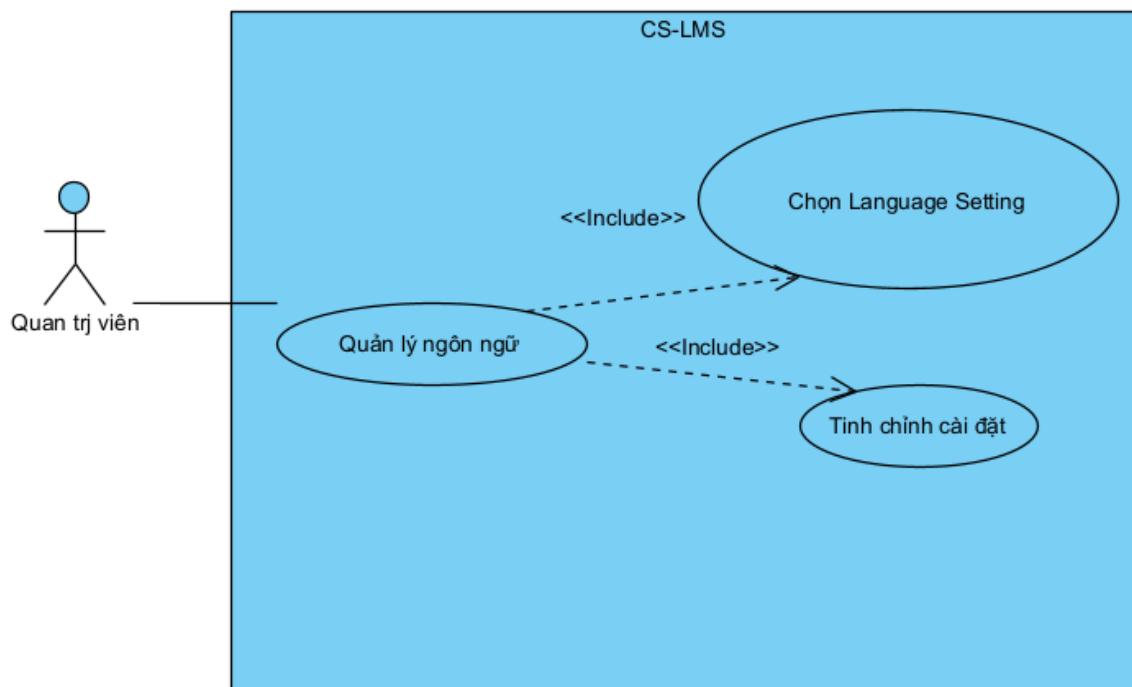
Hình 2.16a. Use case Quản lý giao diện

Chức năng	Quản lý giao diện
Tác nhân	Quản trị viên
Tiền điều kiện	Quản trị viên đã đăng nhập vào hệ thống
Kịch bản chính	<ol style="list-style-type: none"> 1. Quản trị viên truy cập hệ thống 2. Quản trị viên chọn mục “Cài đặt hệ thống” 3. Quản trị viên chọn chức năng “Quản lý giao diện (Theme)” 4. Hệ thống hiển thị danh sách các giao diện hiện có, bao gồm: <ul style="list-style-type: none"> • Tên giao diện • Mô tả • Trạng thái (đang sử dụng / chưa sử dụng) 5. Quản trị viên thực hiện các thao tác <ul style="list-style-type: none"> • Chọn giao diện làm giao diện mặc định • Cài đặt giao diện • Kích hoạt / vô hiệu hóa giao diện 6. Quản trị viên lưu thay đổi 7. Hệ thống cập nhật giao diện mới cho toàn bộ hệ thống 8. Hệ thống hiển thị thông báo: “Cập nhật giao diện thành công.”

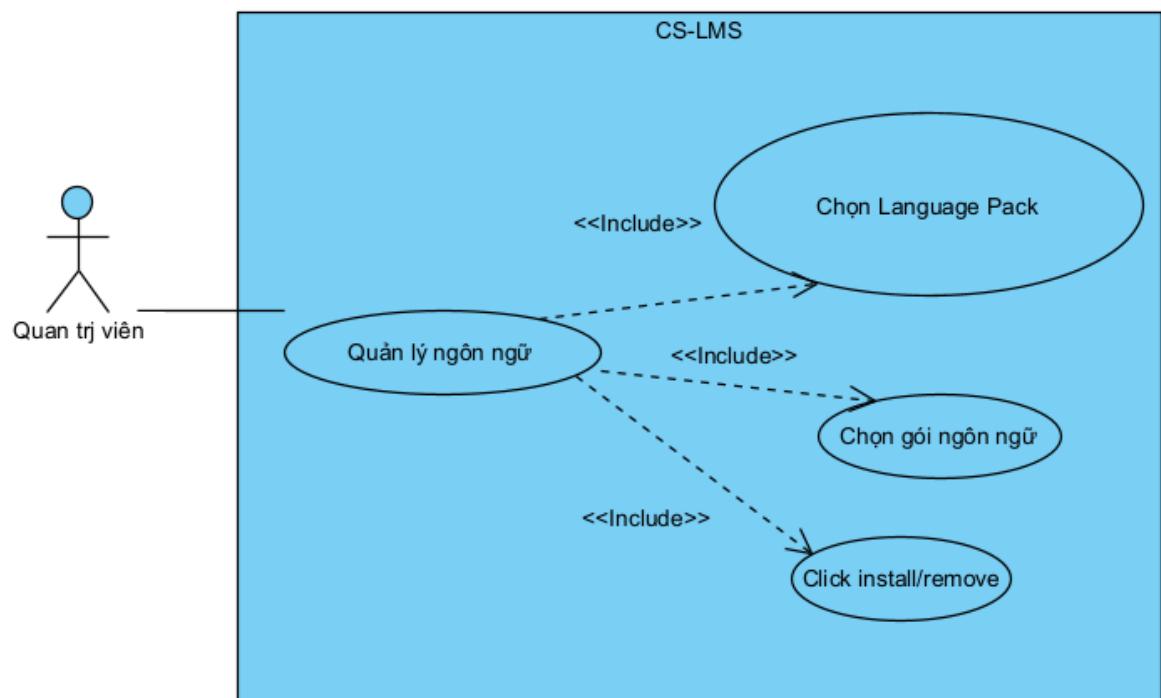
Ngoại lệ	<ol style="list-style-type: none"> 1. Trường hợp 1: Giao diện không hợp lệ hoặc bị lỗi → Hệ thống hiển thị thông báo: “Không thể áp dụng giao diện này.” 2. Trường hợp 2: Quản trị viên không có quyền thay đổi giao diện → Hệ thống hiển thị thông báo: “Bạn không có quyền quản lý giao diện.” 3. Trường hợp 3: Lỗi trong quá trình cập nhật giao diện → Hệ thống hiển thị thông báo: “Cập nhật giao diện thất bại.”
-----------------	---



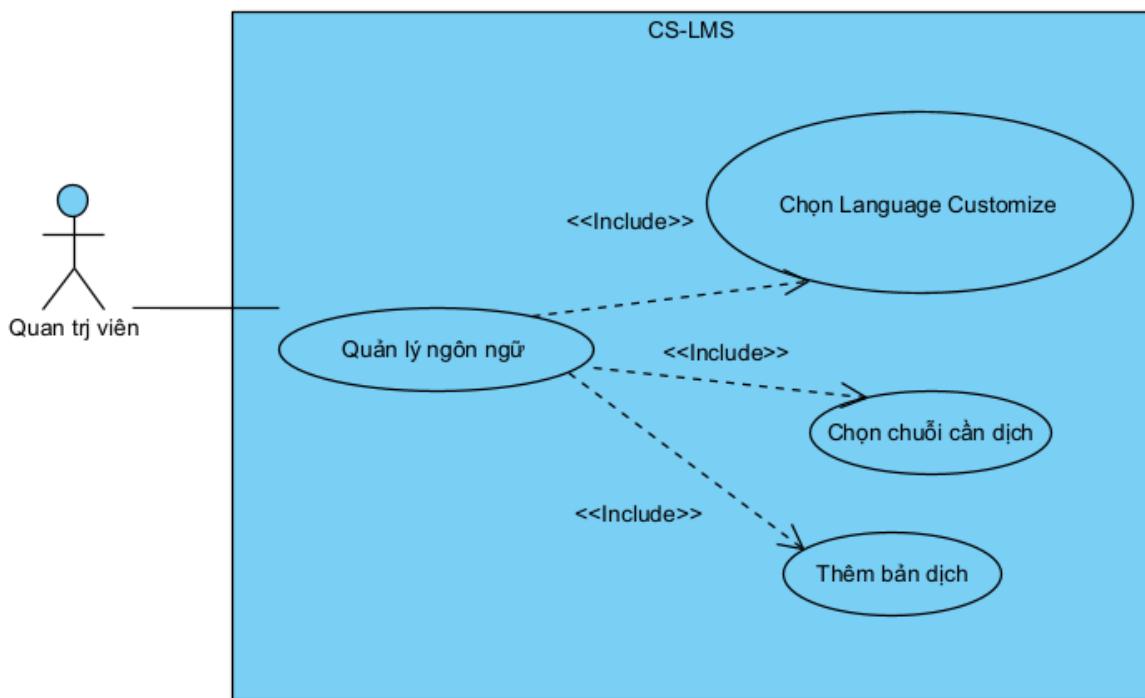
Hình 2.16b. Sơ đồ Activity của Chức năng Quản lý Giao diện
D, Quản lý ngôn ngữ



Hình 2.17a. Use case Quản lý ngôn ngữ



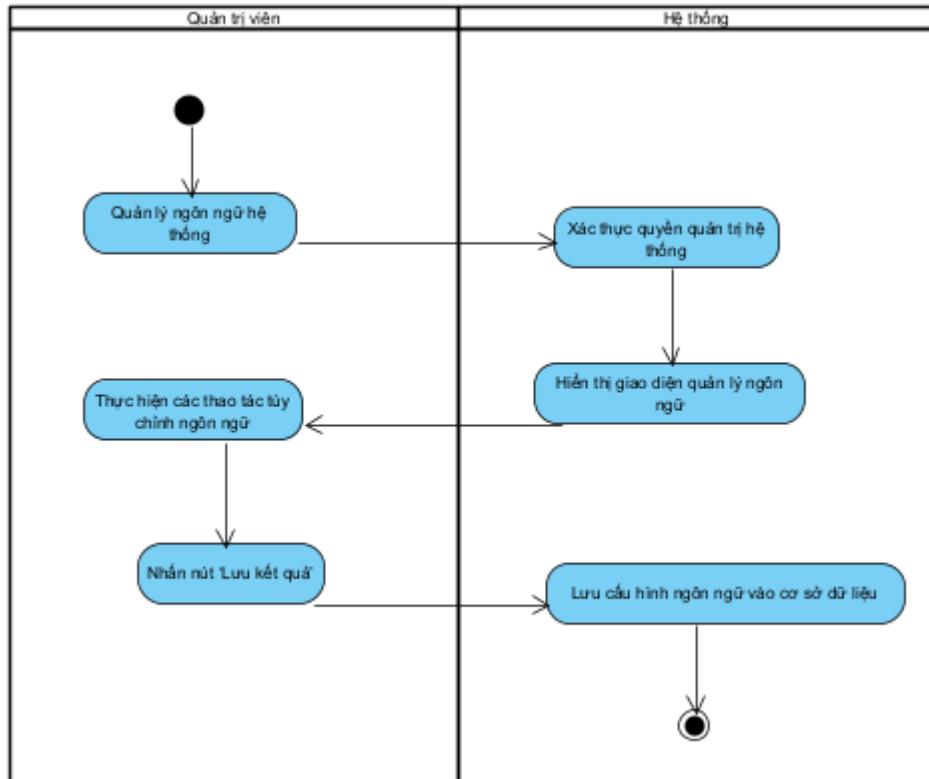
Hình 2.17b. Use case Quản lý ngôn ngữ



Hình 2.17c. Use case Quản lý ngôn ngữ

Chức năng	Quản lý ngôn ngữ
Tác nhân	Quản trị viên
Tiền điều kiện	Quản trị viên đã đăng nhập vào hệ thống
Kịch bản chính	<p>1. Quản trị viên truy cập hệ thống</p> <p>2. Quản trị viên chọn mục “Cài đặt hệ thống”</p> <p>3. Quản trị viên chọn chức năng Language Pack/Language Customize/ Language Setting</p> <ul style="list-style-type: none"> • Language Setting: <ul style="list-style-type: none"> • Quản trị viên chọn tích vào các cài đặt muốn đặt • Click lưu thay đổi • Language Customize: <ul style="list-style-type: none"> • Quản trị viên chọn load string • Quản trị viên chọn vào chuỗi cần dịch • Nhập bản dịch

	<ul style="list-style-type: none"> • Click lưu thay đổi • Language Pack: <ul style="list-style-type: none"> • Quản trị viên chọn vào 1 pack ngôn ngữ • Chọn install/remove để thêm hoặc xóa pack ngôn ngữ
Ngoại lệ	



Hình 2.17d. Sơ đồ Activity của Chức năng Bật/tắt lựa chọn ngôn ngữ cho người dùng

2.3 Thiết kế và cài đặt plugin hỗ trợ chấm điểm sử dụng LLM – AI Grading

AI Grading là plugin đóng vai trò một trợ lý ảo, giúp giáo viên để xuất điểm số và lời nhận xét chi tiết cho bài làm của sinh viên dựa trên đáp án mẫu hoặc tiêu chí chấm điểm (Rubric). Giảng viên cấu hình đề bài, các lựa chọn khi chấm bài sử dụng AI (đáp án mẫu, hướng dẫn,...). Khi giảng viên chọn các bài làm và gửi cho AI chấm, hệ thống sẽ tự động tổng hợp và xử lý dữ liệu, gửi đến LLM (API endpoint được phát triển riêng trên server). Kết quả trả về điểm + nhận xét được lưu ở trạng thái “Chờ duyệt”. Giáo viên xem xét, chỉnh sửa, đánh giá và công bố điểm chính thức.

2.3.1. Yêu cầu chức năng

Các tác nhân (Actor): Giảng viên, Hệ thống (Plugin, Cron, API)

A. Đối với Giảng viên (Teacher)

1. Cấu hình tham số chấm:
 - Giảng viên phải nhập được hướng dẫn chấm.
 - Giảng viên cung cấp đáp án mẫu (nếu có).
2. Chọn và Gửi yêu cầu chấm:
 - Giao diện danh sách bài nộp (Submission list) cần có checkbox để chọn nhiều sinh viên.
 - Nút chức năng "Gửi đi chấm AI".
 - Hệ thống hiển thị trạng thái của từng bài
3. Giao diện Rà soát:
 - Hiển thị song song: Bài làm sinh viên - Kết quả AI đề xuất (Điểm + Nhận xét)
- Ô nhập điểm chính thức.
 - Cho phép Giảng viên sửa lại Điểm và Nhận xét của AI.
 - Nút Lưu thay đổi (lưu lại kết quả chấm, nhận xét của giảng viên).
4. Lưu điểm vào Gradebook
 - Có nút Xác nhận và Cập nhật vào Sổ điểm

B. Đối với Hệ thống (Plugin Backend & Cron)

1. Quản lý hàng đợi:
 - Lưu trữ trạng thái các yêu cầu vào bảng riêng.
2. Cơ chế Cronjob:
 - Quét bảng Queue để tìm các record PENDING.
 - Chuẩn bị dữ liệu: Lấy nội dung bài làm (Online text) hoặc link file đính kèm (File submission).
 - Gửi request đến FastAPI (POST).
 - Cập nhật trạng thái thành PROCESSING sau khi nhận mã 202 từ FastAPI.
3. Xử lý Callback:
 - Cung cấp một endpoint (URL) để FastAPI gọi lại.
 - Nhận dữ liệu JSON (Điểm, nhận xét) từ FastAPI.

- Xác thực request (bảo mật).
- Lưu kết quả vào bảng kết quả tạm và cập nhật trạng thái Queue thành AI DONE.

2.3.2. Yêu cầu phi chức năng

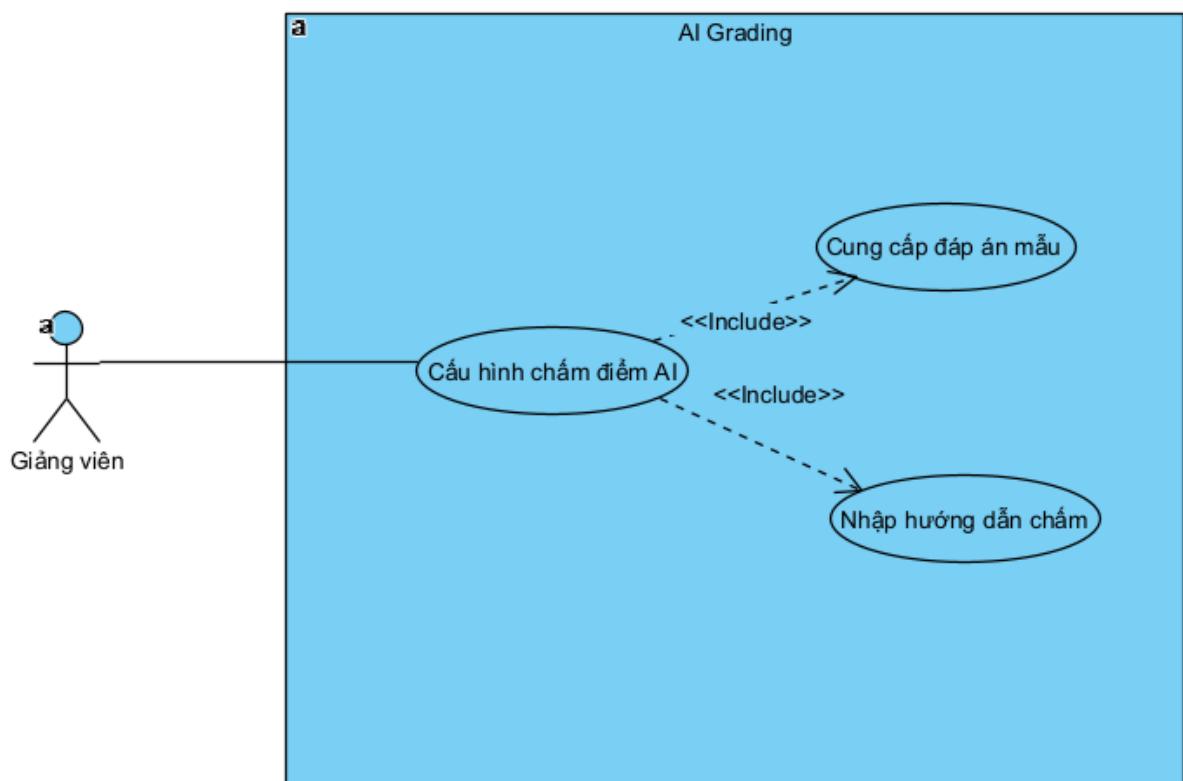
- Hiệu năng (Performance) & Tính bất đồng bộ:
 - Plugin tuyệt đối không được chờ phản hồi từ LLM trong luồng chính (PHP process). Mọi giao tiếp xử lý nặng phải qua Cron hoặc Asynchronous Request.
 - Cronjob chạy tần suất cao (1 lần /phút) để giảm độ trễ trải nghiệm.
- Khả năng mở rộng (Scalability): FastAPI cần có khả năng xử lý hàng đợi nội bộ để tránh quá tải nếu Cron Moodle đẩy sang 100 bài cùng lúc.
- Bảo mật (Security):
 - Callback URL: Phải có cơ chế xác thực để tránh việc kẻ xấu giả mạo FastAPI gửi điểm giả vờ.
 - Dữ liệu: Không gửi thông tin cá nhân nhạy cảm của sinh viên sang LLM
- Trải nghiệm người dùng (UX): Cần có thông báo khi quá trình chấm hoàn tất.

2.3.3. Đặc tả Use case

Use case 1: Cấu hình chức năng chấm điểm

Tác nhân	Giảng viên
Tiền điều kiện	<ul style="list-style-type: none"> • Giáo viên đã đăng nhập thành công vào hệ thống. • Bài tập (Assignment) đã được tạo. • Giáo viên đang ở trong giao diện quản lý bài tập đó.
Kịch bản chính	<ol style="list-style-type: none"> 1. Giáo viên chọn chức năng "Cấu hình chấm điểm AI" tại bài tập cần thiết lập. 2. Hệ thống hiển thị giao diện Cấu hình chấm điểm AI gồm các trường hướng dẫn chấm, đáp án mẫu (online text), đăng tải đáp án mẫu (upload file) 3. Giáo viên nhập nội dung vào trường "Hướng dẫn chấm"

	4. Giáo viên nhập nội dung vào trường Đáp án mẫu hoặc đăng tải file (nếu có) 5. Giáo viên nhấn nút “Lưu cấu hình” 6. Hệ thống lưu dữ liệu và thông báo “Cấu hình thành công”
Hậu điều kiện	<ul style="list-style-type: none"> Cấu hình chấm điểm được lưu vào cơ sở dữ liệu. Hệ thống sẵn sàng chấm bài dựa trên hướng dẫn và đáp án mới cập nhật.
Ngoại lệ	

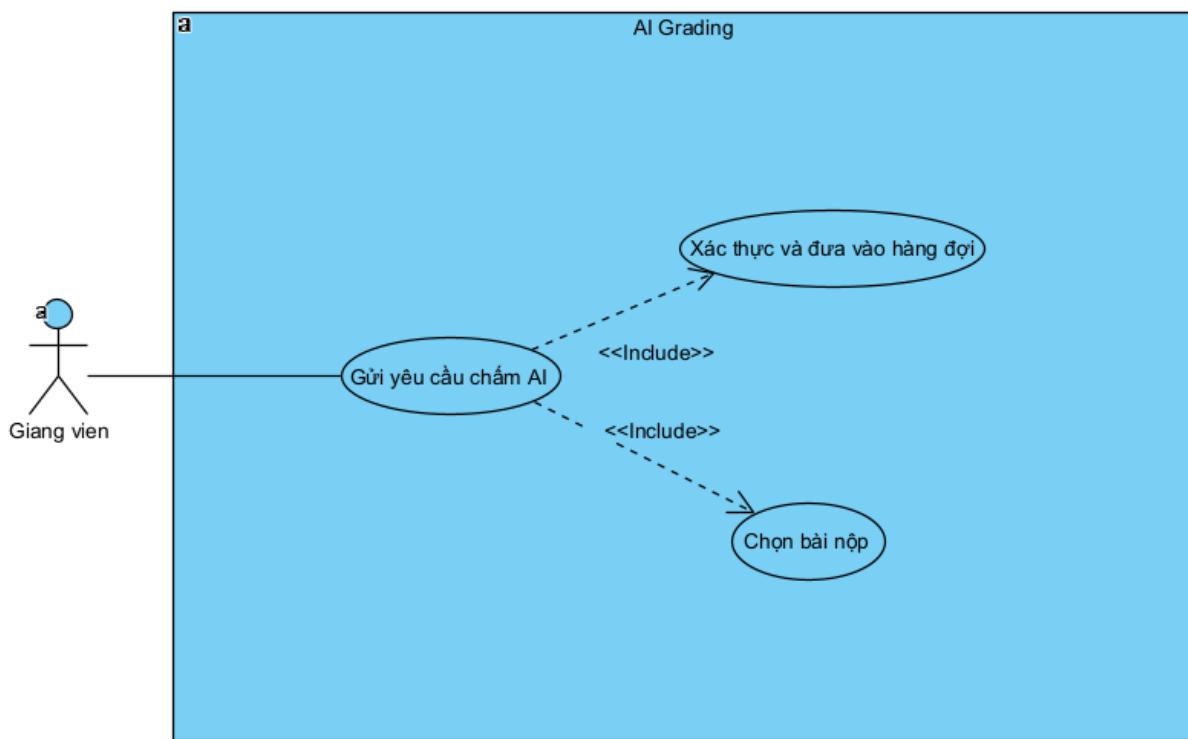


Hình 2.18. Biểu đồ Use case Cấu hình chấm điểm AI

Use case 2: Gửi yêu cầu chấm điểm từ plugin AI Grading

Tác nhân	Giảng viên
Tiền điều kiện	Sinh viên nộp bài, giảng viên cấu hình chức năng chấm điểm
Kịch bản chính	1. Giảng viên truy cập trang chấm bài của Assignment.

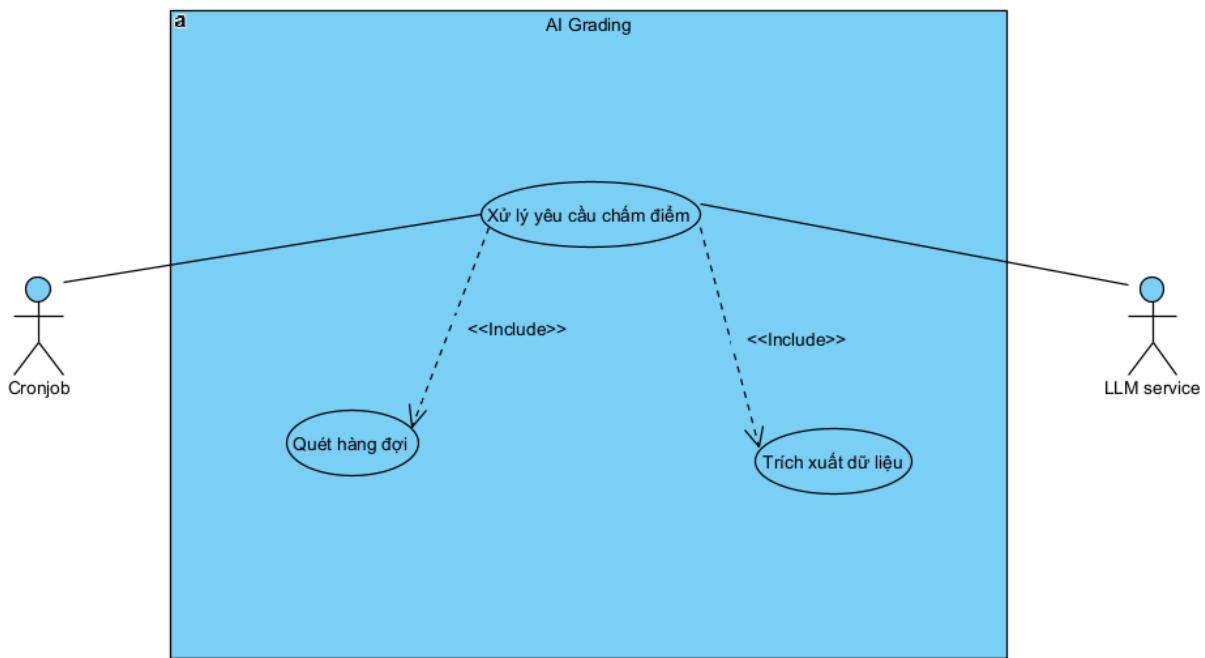
	<p>2. Giảng viên tích chọn các bài nộp cụ thể cần chấm (hoặc chọn tất cả danh sách).</p> <p>3. Giảng viên nhấn nút "Gửi đi chấm AI".</p> <p>4. Hệ thống kiểm tra tính hợp lệ của các bài nộp được chọn (ví dụ: file có tồn tại).</p> <p>5. Hệ thống ghi nhận yêu cầu, chuyển trạng thái các bài nộp này sang "Đang chờ xử lý" (In Queue).</p> <p>6. Hệ thống hiển thị thông báo xác nhận số lượng bài đã được đưa vào hàng đợi thành công.</p>
Hậu điều kiện	Các bài nộp chuyển sang trạng thái “Processing”
Ngoại lệ	A1: Chưa chọn bài nộp: Nếu giảng viên nhấn gửi mà chưa tích chọn bài nào -> Hệ thống hiển thị thông báo yêu cầu chọn ít nhất một bài.



Hình 2.19. Biểu đồ Use case Gửi yêu cầu chấm điểm

Use case 3: Nhận và xử lý yêu cầu chấm điểm từ moodle

Tác nhân	Dịch vụ AI (Tác nhân hệ thống)
Tiền điều kiện	<ul style="list-style-type: none"> Có các bài nộp ở trạng thái "Đang chờ xử lý" trong hàng đợi. Dịch vụ AI đang hoạt động và sẵn sàng kết nối.
Kịch bản chính	<ol style="list-style-type: none"> Hệ thống quét và lấy danh sách các yêu cầu chấm điểm đang nằm trong hàng đợi. Hệ thống trích xuất dữ liệu cần thiết (file bài làm, nội dung câu trả lời, đáp án mẫu, hướng dẫn chấm). Hệ thống chuyển giao gói dữ liệu này sang Dịch vụ AI. Dịch vụ AI gửi tín hiệu xác nhận đã tiếp nhận yêu cầu thành công (để Hệ thống biết không cần gửi lại). Dịch vụ AI thực hiện phân tích nội dung bài làm, tạo ngữ cảnh và chấm điểm dựa trên tiêu chí. Sau khi xử lý xong, Dịch vụ AI gửi trả kết quả (điểm đề xuất, lời nhận xét) ngược lại cho Hệ thống. Hệ thống nhận kết quả và lưu vào khu vực dữ liệu tham khảo (tách biệt hoàn toàn với bảng điểm chính thức của Assignment). Hệ thống cập nhật trạng thái của bài nộp sang "Đã có kết quả AI".
Hậu điều kiện	<ul style="list-style-type: none"> Bài nộp có điểm đề xuất và nhận xét từ AI. Trạng thái bài nộp sẵn sàng để Giảng viên rà soát.
Ngoại lệ	A1: Chưa chọn bài nộp: Nếu giảng viên nhấn gửi mà chưa tích chọn bài nào -> Hệ thống hiển thị thông báo yêu cầu chọn ít nhất một bài.

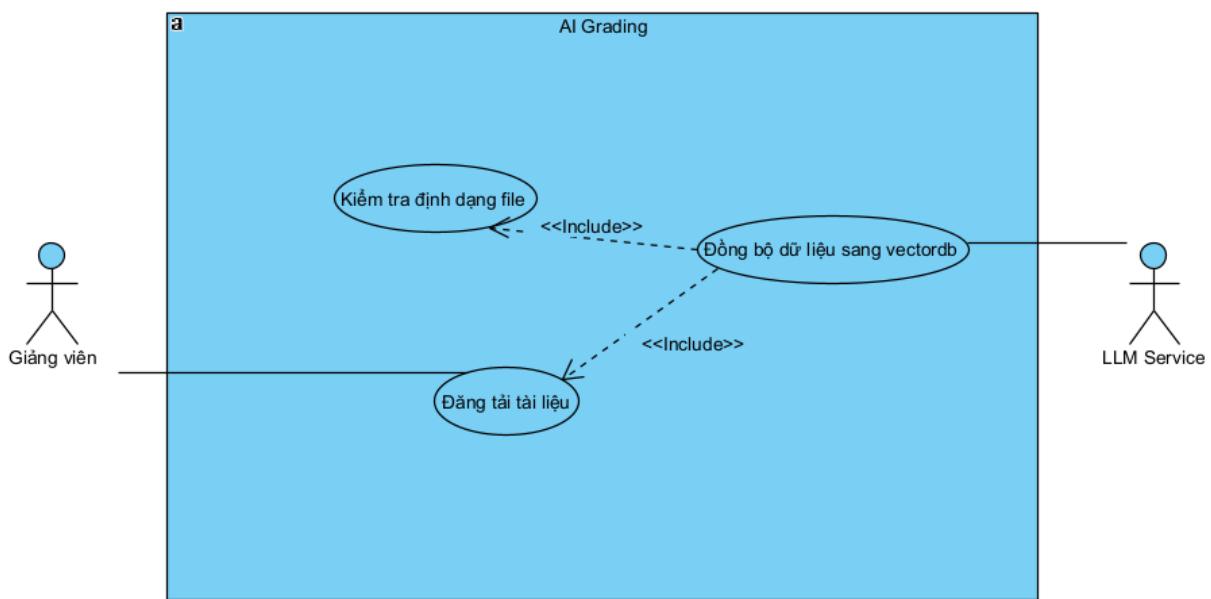


Hình 2.20. Biểu đồ Use case Nhận và xử lý yêu cầu chấm điểm

Use case 4: Đăng tải và đồng bộ tài liệu

Tác nhân	Giảng viên, Dịch vụ AI (Tác nhân hệ thống)
Tiền điều kiện	Giảng viên đã đăng nhập và có quyền chỉnh sửa (Editing Teacher) trong khóa học.
Kịch bản chính	<ol style="list-style-type: none"> Giảng viên truy cập vào khóa học và bật chế độ chỉnh sửa. Giảng viên thực hiện chức năng "Thêm tài nguyên" của Moodle để tải lên một tài liệu mới (ví dụ: File PDF, Word, Giáo trình). Moodle Core thực hiện lưu trữ file vào hệ thống tập tin (File System) và hiển thị trên giao diện khóa học. Plugin tự động phát hiện sự kiện có tài liệu mới vừa được tạo thành công. Plugin trích xuất thông tin định danh của tài liệu và gửi yêu cầu đồng bộ sang Backend AI. Backend AI tiếp nhận thông tin, thực hiện đọc nội dung file từ đường dẫn được cung cấp.

	7. Backend AI xử lý nội dung và lưu trữ dưới dạng vector vào Cơ sở dữ liệu Vector gắn với ngũ cành khóa học đó.
Hậu điều kiện	<ul style="list-style-type: none"> Tài liệu hiển thị cho sinh viên xem bình thường trên Moodle. Nội dung tài liệu đã được đánh chỉ mục (Indexed) trong Vector DB, sẵn sàng để AI truy xuất (Retrieve) khi trả lời câu hỏi hoặc chấm điểm.
Ngoại lệ	A1: Định dạng file không hỗ trợ: Giảng viên upload file ảnh hoặc video (mà hệ thống chưa hỗ trợ OCR/Transcribe).



Hình 2.21. Biểu đồ Use case Đăng tải và đồng bộ dữ liệu

2.3.4. Phân tích hoạt động

Use case 1: Cấu hình chức năng chấm điểm

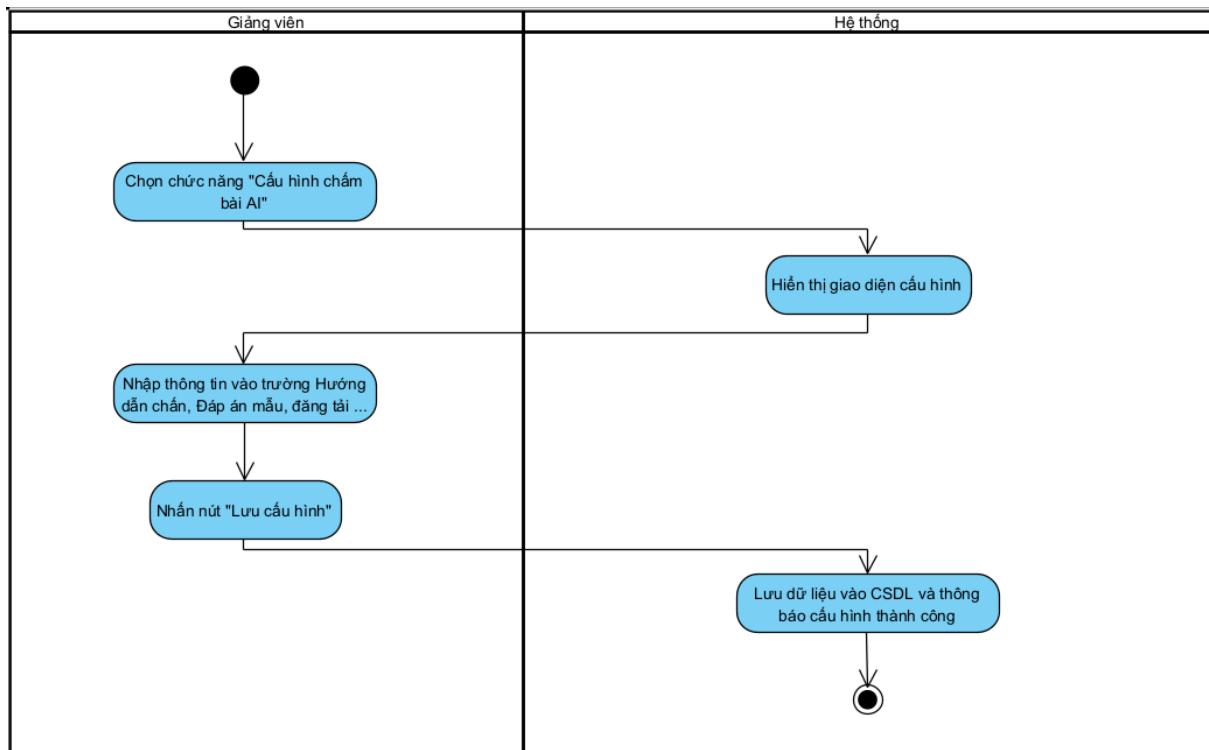
Bước 1 (Giảng viên): Giảng viên tác động vào ứng dụng bằng cách chọn chức năng "Cấu hình chấm bài AI".

Bước 2 (Hệ thống): Nhận được yêu cầu, luồng xử lý chuyển sang hệ thống. Hệ thống thực hiện hành động "Hiển thị giao diện cấu hình" để người dùng tương tác.

Bước 3 (Giảng viên): Sau khi giao diện hiện ra, luồng xử lý quay về phía giảng viên. Giảng viên thực hiện thao tác nhập liệu: "Nhập thông tin vào trường Hướng dẫn chấm, Đáp án mẫu, đăng tải...".

Bước 4 (Giảng viên): Sau khi hoàn tất nhập liệu, Giảng viên xác nhận thao tác bằng cách "Nhấn nút 'Lưu câu hình'".

Bước 5 (Hệ thống): Luồng xử lý chuyển về phía hệ thống để xử lý dữ liệu. Hệ thống thực hiện đồng thời việc "Lưu dữ liệu vào CSDL" và gửi phản hồi "thông báo câu hình thành công" cho người dùng.



*Hình 2.22. Biểu đồ hoạt động Câu hình chức năng chấm bài
Use case 2: Gửi yêu cầu chấm điểm từ AI Grading*

Bước 1: Khởi tạo và Truy cập:

- Quy trình bắt đầu từ phía Giảng viên (Nút khởi đầu - Start Node).
- Giảng viên thực hiện hành động "Truy cập vào trang chấm điểm".

Bước 2: Chọn bài nộp: Giảng viên tiếp tục thực hiện hành động "Tích chọn các bài làm để chấm". Tại đây, giảng viên có thể chọn từng bài cụ thể hoặc chọn hàng loạt.

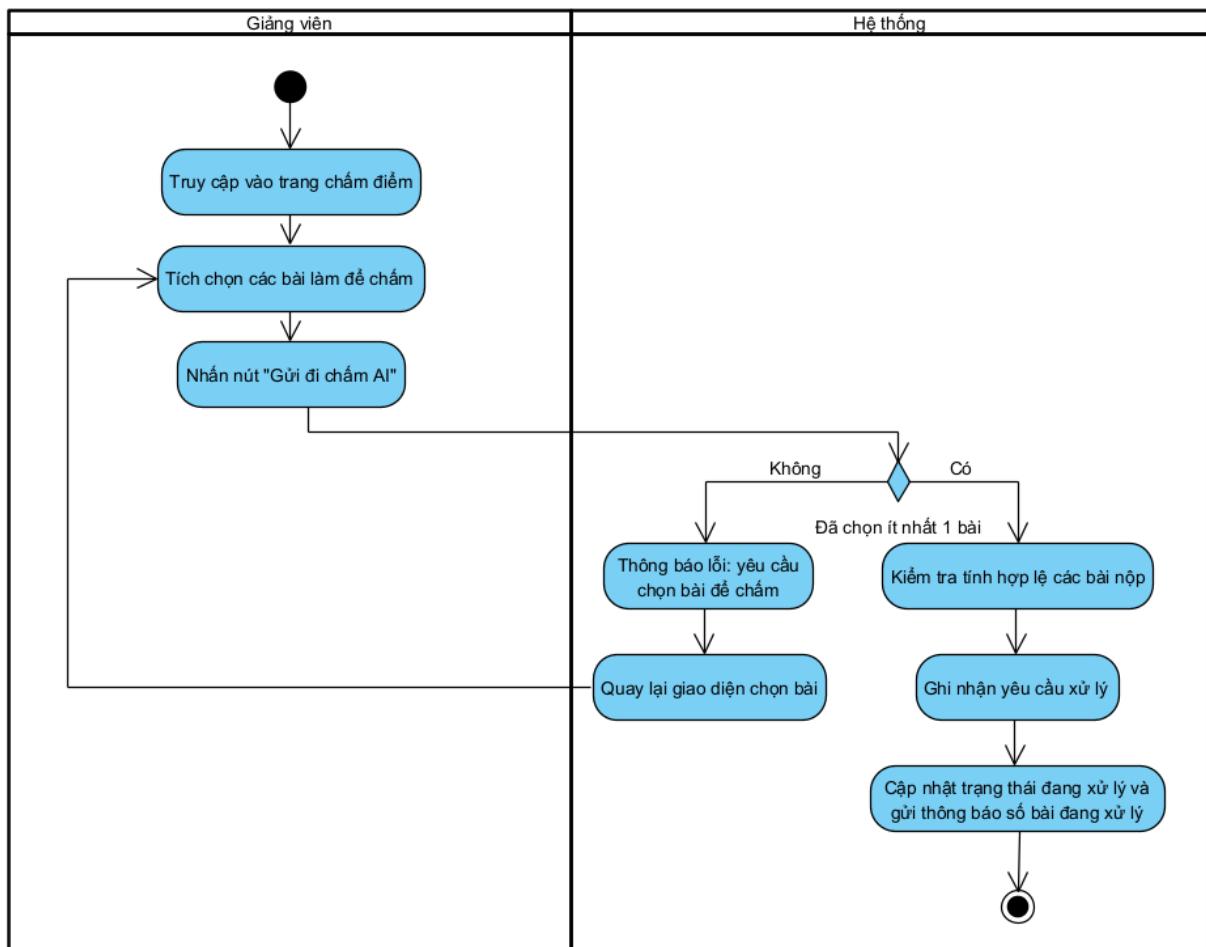
Bước 3: Gửi yêu cầu:

- Sau khi chọn xong, Giảng viên thực hiện hành động "Nhấn nút 'Gửi đi chấm AI'".
- Luồng điều khiển chuyển sang làn (swimlane) của Hệ thống.

Bước 4: Kiểm tra điều kiện (Decision Node): Hệ thống thực hiện kiểm tra điều kiện: "Đã chọn ít nhất 1 bài?". Tại đây quy trình rẽ nhánh:

- Trường hợp A: Chưa chọn bài nào (Nhánh "Không")
 - Hệ thống thực hiện hành động: "Thông báo lỗi: yêu cầu chọn bài để chấm".
 - Hệ thống điều hướng: "Quay lại giao diện chọn bài".
 - Luồng điều khiển quay ngược lại (Loop) về bước "Tích chọn các bài làm để chấm" ở phía Giảng viên để thực hiện lại.
- Trường hợp B: Đã chọn bài hợp lệ (Nhánh "Có")
 - Hệ thống tiến hành "Kiểm tra tính hợp lệ các bài nộp" (ví dụ: file tồn tại, đúng định dạng).
 - Sau đó, hệ thống thực hiện "Ghi nhận yêu cầu xử lý" vào cơ sở dữ liệu.
 - Cuối cùng, hệ thống thực hiện "Cập nhật trạng thái đang xử lý và gửi thông báo số bài đang xử lý" cho giảng viên biết.

Bước 5: Kết thúc: Quy trình kết thúc tại nút kết thúc (Final Node) ở phía Hệ thống sau khi thông báo thành công.



Hình 2.23. Biểu đồ hoạt động Gửi yêu cầu chấm bài
Use case 3: Nhận và xử lý yêu cầu chấm điểm từ moodle

Bước 1: Tại phân hệ Moodle, hệ thống thực hiện quét hàng đợi và lấy danh sách các yêu cầu chấm điểm cần xử lý.

Bước 2: Moodle thực hiện trích xuất dữ liệu cần thiết từ các yêu cầu vừa lấy được.

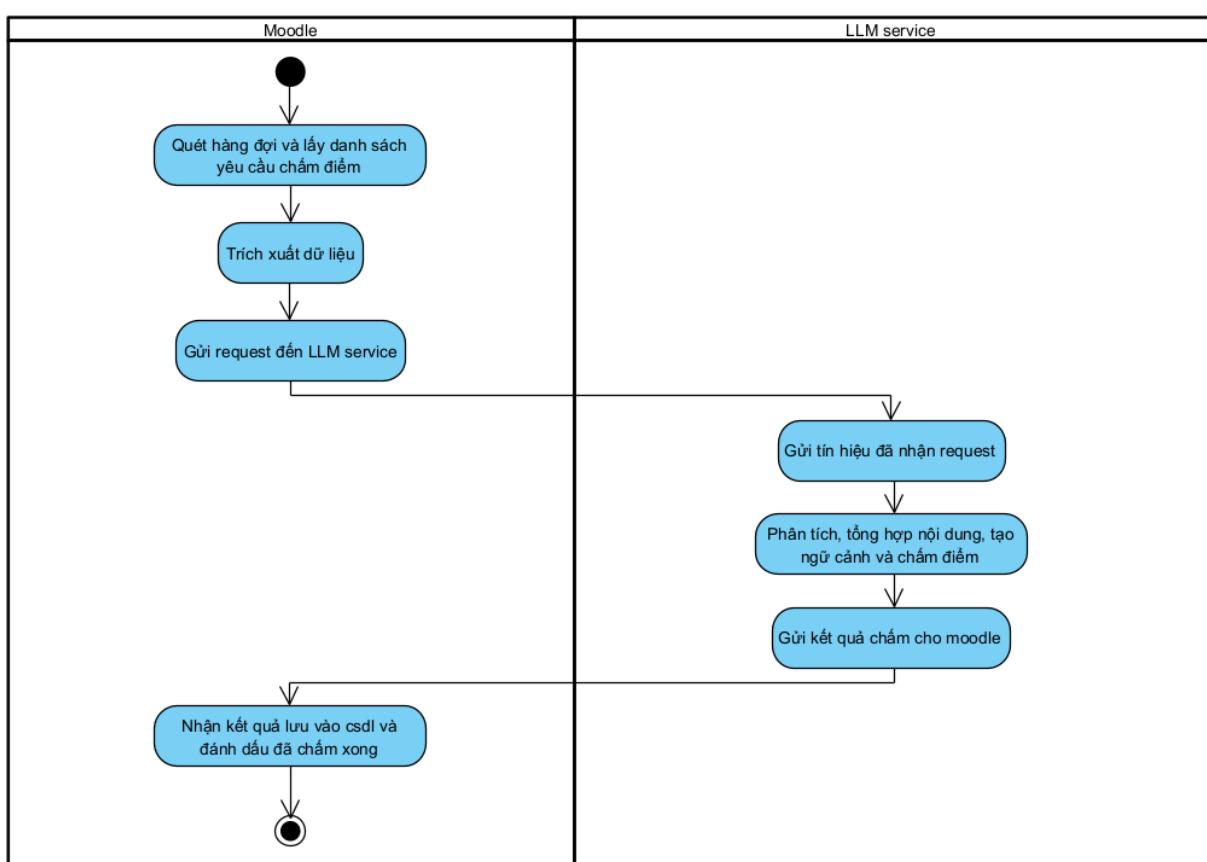
Bước 3: Moodle gửi request (yêu cầu) chứa dữ liệu đến LLM service.

Bước 4: Tại phân hệ LLM service, hệ thống gửi tín hiệu xác nhận đã nhận request từ Moodle.

Bước 5: LLM service thực hiện phân tích, tổng hợp nội dung, tạo ngữ cảnh và tiến hành chấm điểm.

Bước 6: LLM service gửi kết quả chấm điểm ngược lại cho Moodle.

Bước 7: Moodle nhận kết quả, lưu vào cơ sở dữ liệu (CSDL) và đánh dấu trạng thái đã chấm xong.



Hình 2.24. Biểu đồ hoạt động nhận và xử lý yêu cầu chấm điểm
Use case 4: Đăng tải và đồng bộ tài liệu

Bước 1: Giảng viên truy cập vào khóa học trên hệ thống và bật chế độ chỉnh sửa.

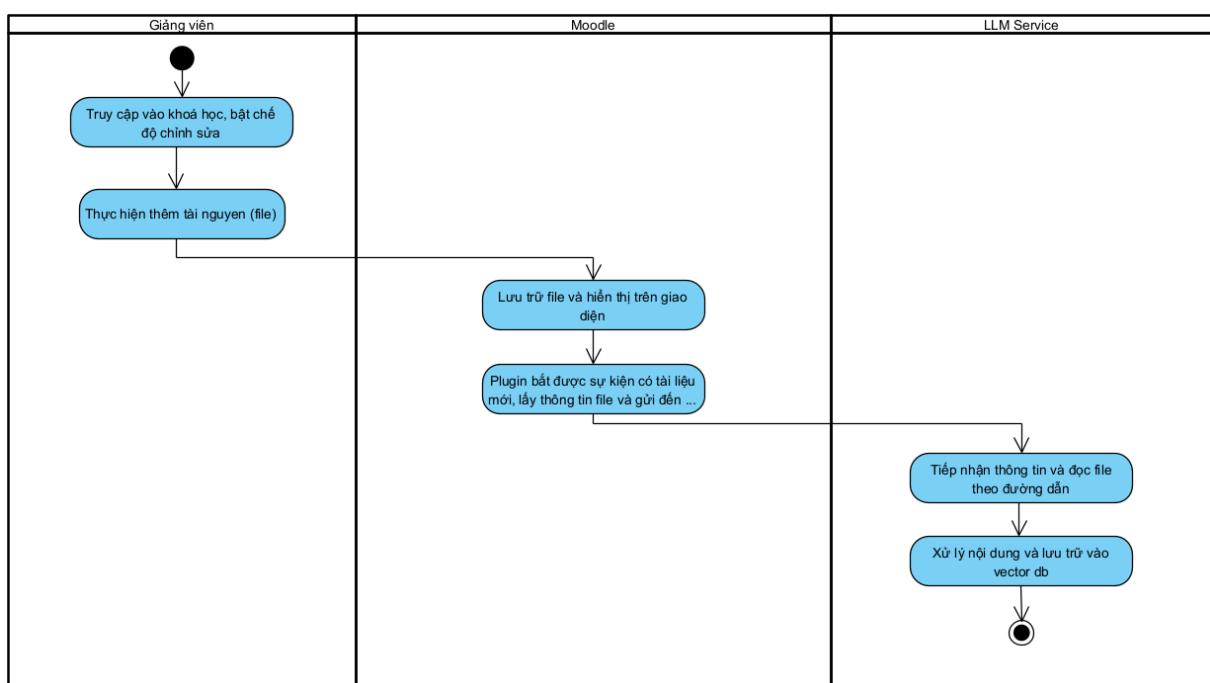
Bước 2: Giảng viên thực hiện chức năng thêm tài nguyên mới (tải file lên).

Bước 3: Hệ thống Moodle lưu trữ file và hiển thị file đó trên giao diện khóa học.

Bước 4: Plugin trên Moodle bắt được sự kiện có tài liệu mới, thực hiện lấy thông tin file và gửi yêu cầu đến LLM Service.

Bước 5: LLM Service tiếp nhận thông tin và thực hiện đọc nội dung file theo đường dẫn được cung cấp.

Bước 6: LLM Service xử lý nội dung văn bản và lưu trữ dữ liệu vào cơ sở dữ liệu vector (Vector DB).



Hình 2.25. Biểu đồ hoạt động Đăng tải và đồng bộ dữ liệu

2.3.5. Thiết kế cơ sở dữ liệu

1. Phân chia kiến trúc: Cơ sở dữ liệu được tổ chức thành 2 phân hệ tách biệt nhưng liên kết chặt chẽ qua các khóa ngoại:

- Phân hệ Moodle Core (Nền tảng): Chứa các bảng gốc của hệ thống LMS, quản lý dữ liệu người dùng và bài tập cơ bản.
- Phân hệ Local Plugin (Mở rộng): Chứa các bảng chuyên biệt cho tính năng AI, quản lý câu hình chấm và kết quả xử lý.

2. Thành phần chi tiết:

A. Nhóm Moodle Core (Dữ liệu nguồn)

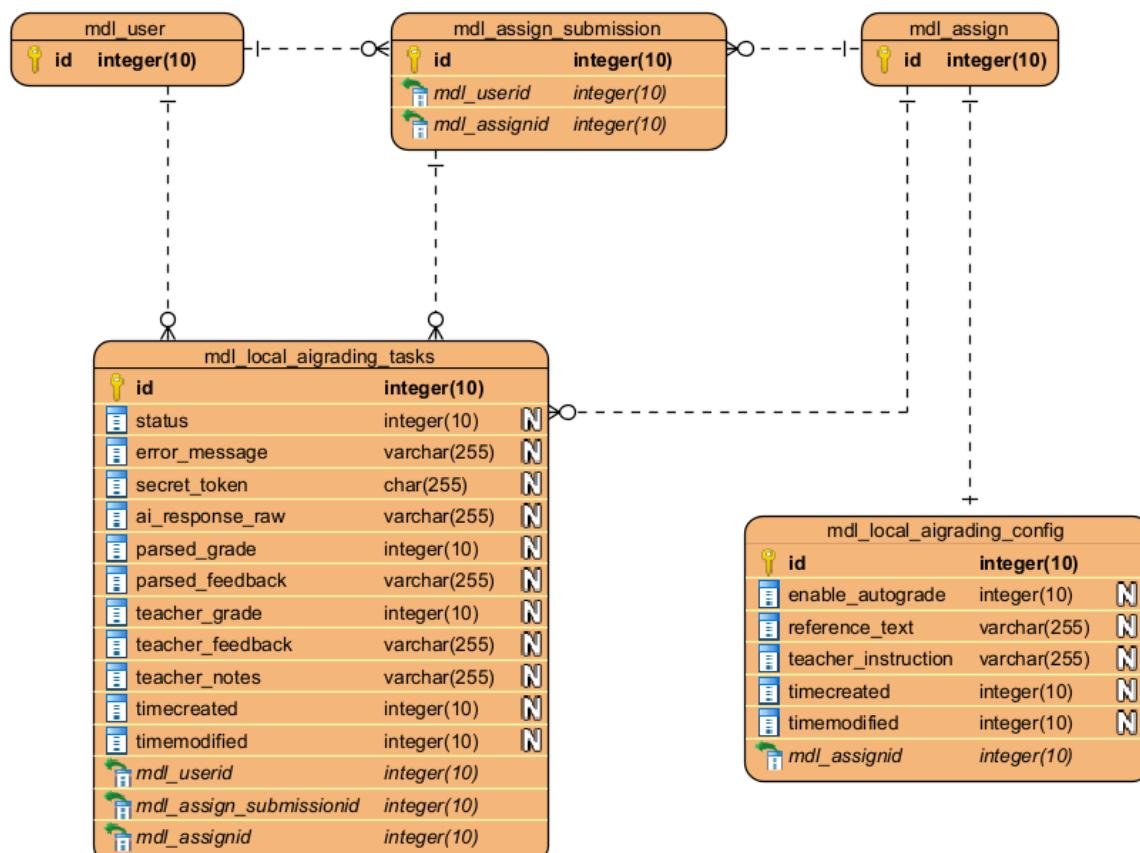
- mdl_user: Quản lý danh tính người dùng (sinh viên nộp bài, giáo viên chấm bài).
 - mdl_assign: Định nghĩa đối tượng bài tập (tên, mô tả, hạn nộp).
 - mdl_assign_submission: Lưu trữ các lượt nộp bài cụ thể của sinh viên (file, text).

B. Nhóm Local Plugin (Dữ liệu nghiệp vụ AI)

- mdl_local_aigrading_config:
 - Lưu thiết lập chấm điểm riêng cho từng Assignment.
 - Chứa: Trạng thái bật/tắt, đáp án mẫu, hướng dẫn chấm (prompt) cho AI.
 - mdl_local_aigrading_tasks:
 - Lưu trạng thái hàng đợi (Pending, Processing, Done, Error).
 - Chứa kết quả chấm: Điểm số, nhận xét từ AI và Token bảo mật.
 - Chứa thông tin hậu kiểm: Điểm và nhận xét do giáo viên chỉnh sửa lại (nếu có).

2.3.6. Cơ chế liên kết:

Các bảng Plugin kết nối với Core thông qua Khóa ngoại: assignmentid, submissionid, userid.



Hình 2.26. Cơ sở dữ liệu và mối quan hệ với các thuộc tính của Moodle

Bảng 1: mdl_local_aigrading_config

Bảng này lưu trữ cấu hình chấm cho từng Assignment.

Tên trường	Kiểu dữ liệu	Mô tả
id	BIGINT(10)	Khóa chính.
assignmentid	BIGINT(10)	Khóa ngoại trỏ đến bảng mdl_assign. Định danh bài tập.
prompt	TEXT	Hướng dẫn chấm (System Prompt) mà giảng viên thiết lập.
sample_answer	LONGTEXT	Nội dung đáp án mẫu (nếu nhập text).
file_ids	VARCHAR(255)	Danh sách ID file đính kèm (dùng File API của Moodle).
timecreated	BIGINT(10)	Thời gian tạo cấu hình.
timemodified	BIGINT(10)	Thời gian cập nhật gần nhất.

Bảng 2: mdl_local_aigrading_tasks

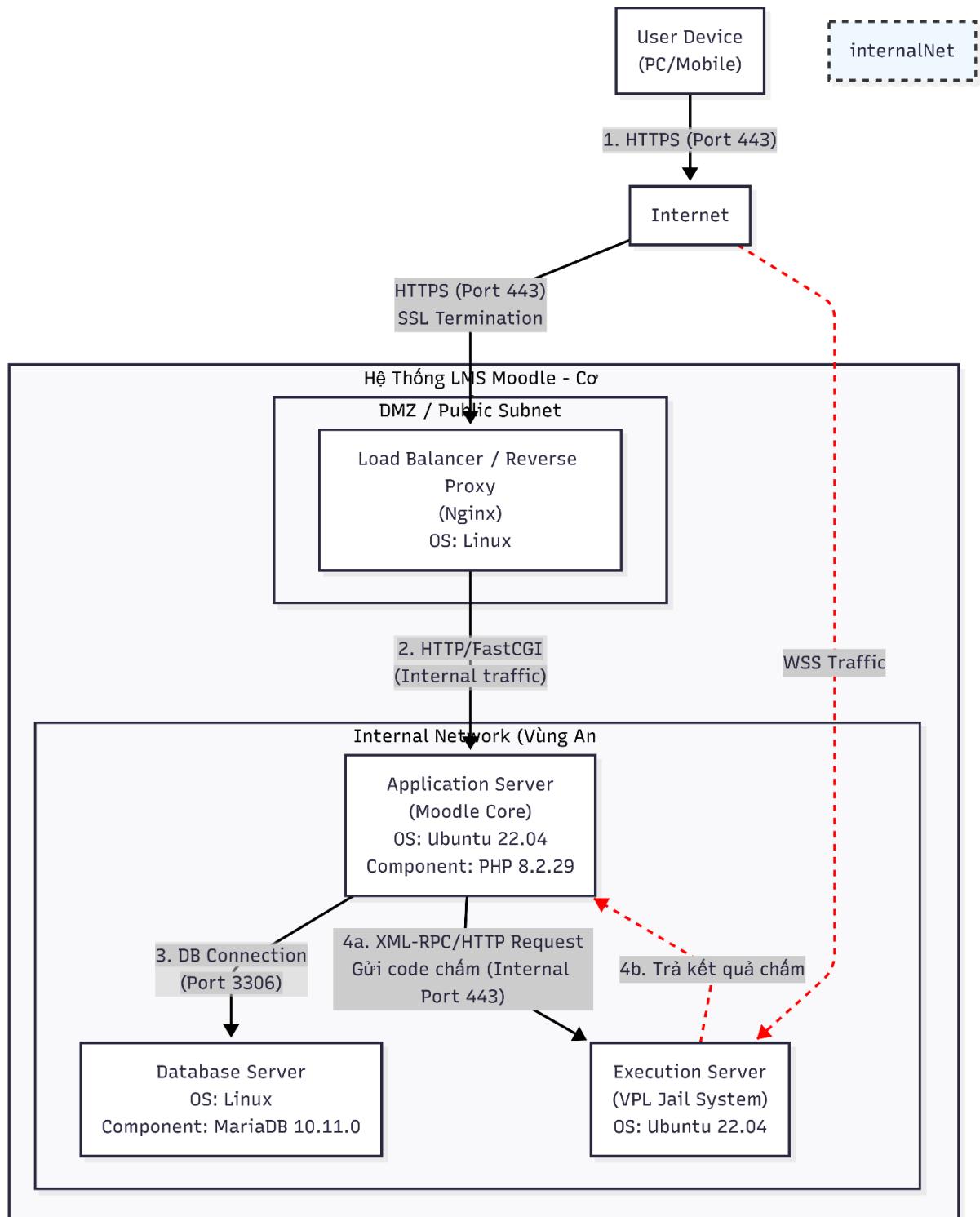
Đây là bảng quan trọng nhất, đóng vai trò hàng đợi xử lý và lưu trữ kết quả tạm thời.

Tên trường	Kiểu dữ liệu	Mô tả
id	BIGINT(10)	Khóa chính.
assignmentid	BIGINT(10)	Khóa ngoại trỏ đến mdl_assign.
submissionid	BIGINT(10)	Khóa ngoại trỏ đến mdl_assign_submission. Định danh bài nộp cụ thể.
userid	BIGINT(10)	Khóa ngoại trỏ đến mdl_user.
status	TINYINT(2)	Trạng thái xử lý: 0 (Pending), 1 (Processing), 2 (Done), 3 (Error).
ai_score	DECIMAL(5,2)	Điểm số do AI đề xuất (Chưa phải điểm chính thức).
ai_feedback	LONGTEXT	Nhận xét chi tiết từ AI (HTML format).
teacher_override	TINYINT(1)	Cờ đánh dấu: 0 (AI chấm), 1 (Giảng viên đã sửa).
request_token	VARCHAR(64)	Token bảo mật ngẫu nhiên sinh ra cho mỗi request, dùng để xác thực Callback.
error_log	TEXT	Ghi lại thông báo lỗi nếu quá trình chấm thất bại.

2.4 Thiết kế Kiến trúc cơ sở hạ tầng

2.4.1. Mô hình kiến trúc tổng quan

Hệ thống quản trị học tập (LMS) được thiết kế dựa trên mô hình phân lớp, tách biệt rõ ràng giữa vùng công khai (DMZ/Public Subnet) và vùng nội bộ (Internal Network) để đảm bảo an toàn thông tin. Kiến trúc bao gồm các thành phần chính: thiết bị người dùng, bộ cân bằng tải (Reverse Proxy), máy chủ ứng dụng, máy chủ cơ sở dữ liệu và máy chủ thực thi mã nguồn.



Hình 2.27. Kiến trúc 3 tầng triển khai trên hạ tầng mạng phân tán

2.4.2. Các thành phần chi tiết

Dựa trên sơ đồ triển khai tại Hình 2.27, hệ thống bao gồm các khối chức năng sau:

1. Phía người dùng (User Device):

- Bao gồm máy tính cá nhân (PC) hoặc thiết bị di động (Mobile).
- Người dùng truy cập hệ thống thông qua trình duyệt web sử dụng giao thức bảo mật **HTTPS (Port 443)**.

2. Phân vùng DMZ / Public Subnet (Vùng đệm):

- Đây là lớp bảo vệ đầu tiên của hệ thống, tiếp nhận mọi yêu cầu từ Internet.
- **Thành phần:** Load Balancer / Reverse Proxy sử dụng **Nginx** chạy trên nền tảng Linux.
- **Chức năng:**
 - **SSL Termination:** Đảm nhận việc giải mã chứng chỉ SSL/TLS, giúp giảm tải cho các server phía sau.
 - Điều hướng các yêu cầu HTTP/HTTPS vào máy chủ ứng dụng bên trong.

3. Phân vùng Internal Network (Vùng an toàn):

Đây là vùng mạng nội bộ, không tiếp xúc trực tiếp với Internet công cộng để đảm bảo bảo mật dữ liệu. Bao gồm 3 máy chủ chính:

- a. **Application Server (Máy chủ Ứng dụng):**
 - **Môi trường:** Hệ điều hành Ubuntu 22.04.
 - **Thành phần cốt lõi:** Moodle Core chạy trên nền tảng **PHP 8.2.29**.
 - **Nhiệm vụ:** Xử lý toàn bộ nghiệp vụ Logic của hệ thống LMS, quản lý người dùng, khóa học và giao diện.
- b. **Database Server (Máy chủ Cơ sở dữ liệu):**
 - **Môi trường:** Hệ điều hành Linux.
 - **Thành phần:** Hệ quản trị cơ sở dữ liệu **MariaDB 10.11.0**.
 - **Nhiệm vụ:** Lưu trữ bền vững dữ liệu hệ thống. Chỉ cho phép kết nối từ Application Server qua cổng **3306**.
- c. **Execution Server (Máy chủ Thực thi - VPL Jail System):**
 - **Môi trường:** Hệ điều hành Ubuntu 22.04.
 - **Nhiệm vụ:** Cung cấp môi trường Sandbox (cô lập) để biên dịch và chạy các đoạn mã nguồn do sinh viên nộp. Đây là thành phần quan trọng để ngăn chặn mã độc ảnh hưởng đến hệ thống chính.

2.4.3. Quy trình luồng dữ liệu (Data Flow)

Theo sơ đồ, luồng hoạt động của hệ thống diễn ra theo các bước sau:

1. Bước 1 - Truy cập: Người dùng gửi yêu cầu HTTPS (Port 443) từ Internet tới hệ thống.
2. Bước 2 - Điều hướng: Nginx tại vùng DMZ tiếp nhận, xử lý SSL và chuyển tiếp yêu cầu dưới dạng HTTP/FastCGI vào Application Server trong vùng nội bộ.
3. Bước 3 - Truy xuất dữ liệu: Application Server kết nối tới Database Server qua cổng 3306 để lấy thông tin hoặc lưu trữ bài làm.
4. Bước 4 - Quy trình chấm điểm (VPL Integration):
 - (4a) Gửi code: Khi có yêu cầu chấm bài, Application Server gửi mã nguồn sang Execution Server thông qua giao thức XML-RPC/HTTP (Internal Port).
 - (4b) Trả kết quả: Sau khi biên dịch và chạy test case trong môi trường Jail, Execution Server trả kết quả điểm số và log lỗi về cho Application Server.

5. **Luồng WSS Traffic (Websocket):** Một kết nối đặc biệt (đường kẻ đứt nét đứt) cho phép thiết lập giao thức WebSocket bảo mật (WSS) từ trình duyệt người dùng, đi qua Proxy và kết nối tới Execution Server để phục vụ tính năng Console tương tác thời gian thực (Interactive Console).

2.4.4. Đánh giá kiến trúc

Mô hình này đảm bảo tính an toàn cao nhờ việc đặt Cơ sở dữ liệu và Mã nguồn ứng dụng trong vùng mạng nội bộ (InternalNet), được che chắn bởi Nginx Proxy. Đồng thời, việc tách rời máy chủ thực thi (Execution Server) giúp hệ thống không bị quá tải hoặc rủi ro bảo mật khi sinh viên chạy các đoạn mã lập trình phức tạp.

2.5. Tổng kết chương II

Chương II đã hoàn thành việc phân tích toàn diện và xây dựng các bản thiết kế kỹ thuật chi tiết cho hệ thống quản trị học tập Moodle tích hợp chấm điểm AI. Cụ thể, nhóm thực hiện đã đạt được các kết quả sau:

1. **Về phân tích yêu cầu:** Đã xác định rõ các yêu cầu chức năng và phi chức năng của hệ thống, từ đó xây dựng các biểu đồ Use case và đặc tả quy trình nghiệp vụ chi tiết cho các tác nhân: Sinh viên, Giảng viên và Quản trị viên. Đặc biệt, quy trình chấm điểm tự động sử dụng AI đã được mô hình hóa rõ ràng thông qua biểu đồ hoạt động, đảm bảo tính khả thi khi triển khai.
2. **Về thiết kế dữ liệu:** Đã thiết kế thành công lược đồ cơ sở dữ liệu với sự phân tách rõ ràng giữa phân hệ Moodle Core và phân hệ Local Plugin, đảm bảo tính toàn vẹn dữ liệu và khả năng mở rộng khi tích hợp các bảng lưu trữ kết quả chấm AI và cấu hình bài tập.
3. **Về kiến trúc hệ thống:** Đã xây dựng mô hình kiến trúc 3 tầng (3-Tier Architecture) triển khai trên nền tảng Docker Container. Thiết kế này giải quyết triệt để bài toán về hiệu năng và an toàn thông tin thông qua việc sử dụng Nginx làm Reverse Proxy tại vùng DMZ và cài đặt máy chủ thực thi (VPL Jail Server) trong vùng mạng nội bộ.

Các bản thiết kế về kiến trúc, cơ sở dữ liệu và giao diện được trình bày trong chương này đóng vai trò là "bản vẽ kỹ thuật" nền tảng, định hướng chính xác cho quá trình hiện thực hóa và triển khai hệ thống sẽ được trình bày chi tiết trong chương sau.

CHƯƠNG III. XÂY DỰNG VÀ TRIỂN KHAI HỆ THỐNG

3.1 Triển khai hạ tầng Server

Các bước dưới đây mô tả chi tiết quy trình thiết lập môi trường máy chủ, cài đặt các công cụ container hóa và triển khai mã nguồn Moodle LMS, hệ thống chấm code VPL.

Lưu ý: Hệ thống sử dụng các câu lệnh cũng như các file cấu hình tương tự sẽ khác đôi chút với bản báo cáo này để tránh lộ cấu hình hệ thống phòng ngừa kẻ gian tấn công.

3.1.1. Thiết lập môi trường và tường lửa cho hệ thống.

Để đảm bảo an ninh cho máy chủ nhưng vẫn cho phép các dịch vụ web hoạt động, hệ thống sử dụng **iptables** để quản lý các luồng truy cập.

Bước 1: Cập nhật hệ thống Trước khi cài đặt, thực hiện cập nhật các gói phần mềm:

```
sudo apt-get update && sudo apt-get upgrade -y
```

Bước 2: Cấu hình iptables Mở các cổng (port) cần thiết cho SSH, Web Server (HTTP/HTTPS) và các cổng dịch vụ của Moodle (8080, 8443, 22).

1. Cho phép kết nối SSH (Cổng 22 - Quan trọng để không bị mất kết nối):
sudo iptables -A INPUT -p tcp --dport 22 -j ACCEPT

2. Cho phép kết nối Web cơ bản (HTTP - 80, HTTPS - 443):
sudo iptables -A INPUT -p tcp --dport 80 -j ACCEPT
sudo iptables -A INPUT -p tcp --dport 443 -j ACCEPT

3. Cho phép các cổng dịch vụ container (81, 8080, 8443):
sudo iptables -A INPUT -p tcp --dport 81 -j ACCEPT
sudo iptables -A INPUT -p tcp --dport 8080 -j ACCEPT
sudo iptables -A INPUT -p tcp --dport 8443 -j ACCEPT

4. Lưu cấu hình iptables (Để đảm bảo cấu hình không mất khi khởi động lại server):
sudo apt-get install iptables-persistent
sudo netfilter-persistent save
sudo netfilter-persistent reload

5. Kiểm tra lại trạng thái các quy tắc đã thêm:
sudo iptables -L -n -v

Bước 3: Cài đặt công cụ hỗ trợ Cài đặt curl để tải các gói cài đặt từ internet:

```
sudo apt install curl -y
```

3.1.2. Cài đặt Docker và Docker Compose

Hệ thống sử dụng Docker để ảo hóa ứng dụng. Quá trình cài đặt được thực hiện tự động thông qua script chính hãng.

Bước 1: Tạo script cài đặt tự động Tạo file install_docker.sh với nội dung sau:

```
nano install_docker.sh
```

Dán nội dung cài đặt:

```
#!/bin/bash
apt update && apt upgrade -y
# Tải và cài đặt Docker Engine
curl -fsSL https://get.docker.com -o get-docker.sh && sudo sh get-
docker.sh
# Tải và cài đặt Docker Compose phiên bản v2.39.3
curl https://github.com/docker/compose/releases/download/v2.39.3/docker-
compose-linux-x86_64 -o /usr/local/bin/docker-compose
# Cấp quyền thực thi
chmod +x /usr/local/bin/docker-compose
apt-get install docker-compose-plugin
```

-SL

Bước 2: Thực thi script Cấp quyền và chạy script để cài đặt:

```
chmod +x install_docker.sh
./install_docker.sh
```

Sau khi chạy xong, kiểm tra phiên bản để xác nhận cài đặt thành công:

```
docker --version
docker-compose --version
```

3.1.3. Triển khai Moodle LMS với Docker Compose

Moodle được triển khai kết hợp với cơ sở dữ liệu MariaDB thông qua file cấu hình docker-compose.yml.

Bước 1: Chuẩn bị thư mục dự án

```
sudo mkdir -p /var/www/html/moodle
cd /var/www/html/moodle
```

Bước 2: Tạo file cấu hình Docker Compose Tạo file docker-compose.yml:

```
nano docker-compose.yml
```

Nội dung file cấu hình mẫu:

```
version: '3'
services:
  mariadb:
    image: docker.io/bitnami/mariadb:latest
    ports:
      - '3306:3306'
    environment:
      - ALLOW_EMPTY_PASSWORD=yes # Lưu ý: Chỉ dùng cho môi trường Dev/Test
      - MARIADB_USER=bn_moodle
      - MARIADB_DATABASE=bitnami_moodle
```

```

    - MARIADB_CHARACTER_SET=utf8mb4
    - MARIADB_COLLATE=utf8mb4_unicode_ci
  volumes:
    - 'mariadb_data:/bitnami/mariadb'

moodle:
  image: docker.io/bitnami/moodle:5.0
  ports:
    - '8080:8080'
    - '8443:8443'
  environment:
    - MOODLE_DATABASE_HOST=mariadb
    - MOODLE_DATABASE_PORT_NUMBER=3306
    - MOODLE_DATABASE_USER=bn_moodle
    - MOODLE_DATABASE_NAME=bitnami_moodle
    - ALLOW_EMPTY_PASSWORD=yes # Lưu ý: Chỉ dùng cho môi trường
Dev/Test
  volumes:
    - 'moodle_data:/bitnami/moodle'
    - 'moodledata_data:/bitnami/moodledata'
  depends_on:
    - mariadb

volumes:
  mariadb_data:
    driver: local
  moodle_data:
    driver: local
  moodledata_data:
    driver: local

```

Bước 3: Khởi chạy hệ thống Tại thư mục chứa file docker-compose.yml, chạy lệnh:

```
docker-compose up -d
```

Lệnh này sẽ tải các image cần thiết và khởi động các container ngầm (detached mode).

3.1.4. Cấu hình Reverse Proxy (Nginx)

Lưu ý: Phần này thiết lập Nginx đóng vai trò là cổng Web Server chính (Port 80/443), tuy nhiên việc kích hoạt SSL yêu cầu tên miền thực tế.

Nginx

```

server {
  listen 80;
  server_name cs-lms.ptit.edu.vn;

  location / {
    proxy_pass http://localhost:8080;
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-Proto $scheme;
  }
}

```

}

Hiện tại, hệ thống có thể truy cập trực tiếp qua địa chỉ IP của máy chủ với cổng 8080: http://<IP_Server>:8080.

3.1.5. Cài đặt và cấu hình VPL JAIL SYSTEM

Phần này mô tả quy trình cài đặt VPL Jail System chạy trực tiếp trên nền tảng hệ điều hành. Hệ thống được cấu hình chạy ở chế độ **HTTP (Unsecured)** trên cổng **8081** và hoạt động phía sau **Nginx Reverse Proxy** (nơi đảm nhận nhiệm vụ mã hóa SSL).

Bước 1: Chuẩn bị môi trường và Thư viện

```
sudo apt-get update  
sudo apt-get install build-essential g++ make -y
```

(Lưu ý: Ở chế độ này, không bắt buộc phải cài gói libssl-dev nếu không dùng tính năng mã hóa nội bộ của VPL, nhưng khuyến nghị vẫn nên cài để đảm bảo đầy đủ thư viện hệ thống).

Bước 2: Tải và Cài đặt VPL Jail System

```
cd /tmp  
wget https://vpl.dis.ulpgc.es/releases/vpl-jail-system-3.0.1.tar.gz  
tar -xvf vpl-jail-system-3.0.1.tar.gz  
cd vpl-jail-system-3.0.1  
sudo ./install-vpl-jail-system.sh
```

Bước 3: Cấu hình Dịch vụ (Quan trọng): Vì sử dụng Nginx làm Proxy, ta sẽ tắt tính năng Secure Port của VPL và chuyển cổng hoạt động sang 8081.

- **Chỉnh sửa file cấu hình** Mở file cấu hình chính:

```
sudo nano /etc/vpl/vpl-jail-system.conf
```

- **Cập nhật thông số** Hãy sửa nội dung file cấu hình giống hệt bên dưới (Chú ý các dòng có dấu # là dòng đã bị vô hiệu hóa):

```
# --- Cấu hình Mạng ---  
# Đổi cổng mặc định sang 8081  
PORT=8081  
  
# Tắt chế độ bảo mật (Vì đã có Nginx lo SSL)  
# SECURE_PORT=443 <-- Comment dòng này  
  
# --- Cấu hình URL ---  
URLPATH=/  
  
# --- Cấu hình Tài nguyên (Tùy chọn) ---  
# UPL_MAX_MEMORY=512Mb
```

Lưu file (Ctrl+O) và thoát (Ctrl+X).

Bước 4: Cấu hình Tường lửa (iptables)

Cần mở cổng 8081 để Nginx có thể kết nối vào, và dải cổng Console để sinh viên tương tác.

```
# 1. Cho phép Nginx kết nối vào cổng dịch vụ 8081  
sudo iptables -A INPUT -p tcp --dport 8081 -j ACCEPT  
  
# 2. Cho phép dải cổng Console (50000-50100)  
# Lưu ý: Các kết nối WebSocket console cũng cần đi qua firewall  
sudo iptables -A INPUT -p tcp --dport 50000:50100 -j ACCEPT  
  
# 3. Lưu cấu hình  
sudo netfilter-persistent save  
sudo netfilter-persistent reload
```

Bước 5: Khởi động lại dịch vụ

```
sudo service vpl-jail-system restart  
sudo service vpl-jail-system status
```

Nếu thấy trạng thái **Active: active (running)** là thành công.

Bước 6: Cấu hình tích hợp trên Nginx: Để Nginx có thể chuyển tiếp các yêu cầu từ <https://cs-lms.ptit.edu.vn/vpl> (hoặc một subdomain) về cổng 8081, cấu hình block server của Nginx cần có đoạn sau:

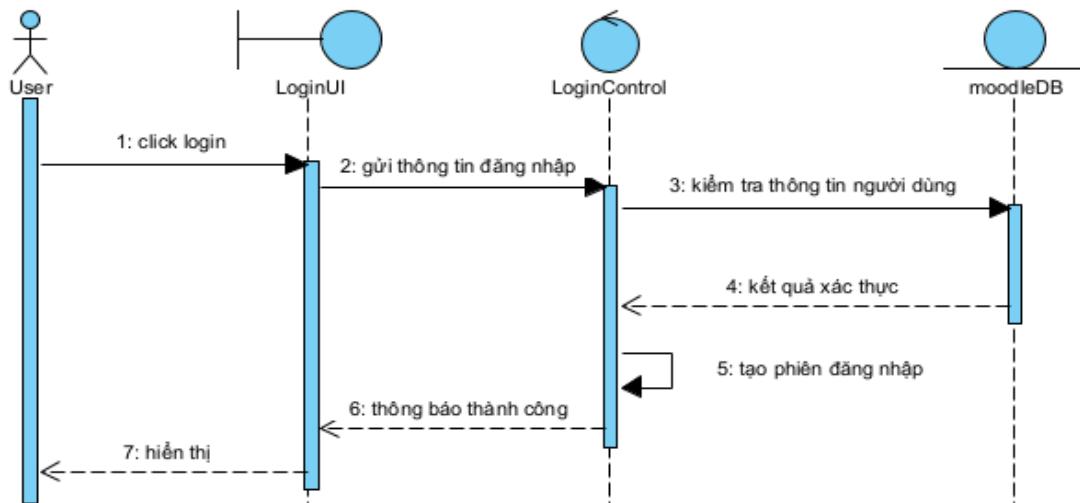
```
Nginx  
# Ví dụ cấu hình trong file /etc/nginx/sites-available/default  
location /vpl/ {  
    proxy_pass http://localhost:8081/;  
  
    # Cấu hình quan trọng cho WebSocket (Cần thiết để chạy Console  
    # VPL)  
    proxy_http_version 1.1;  
    proxy_set_header Upgrade $http_upgrade;  
    proxy_set_header Connection "upgrade";  
  
    proxy_set_header Host $host;  
    proxy_set_header X-Real-IP $remote_addr;  
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;  
}
```

3.2. Phát triển Plugin AI Chấm điểm

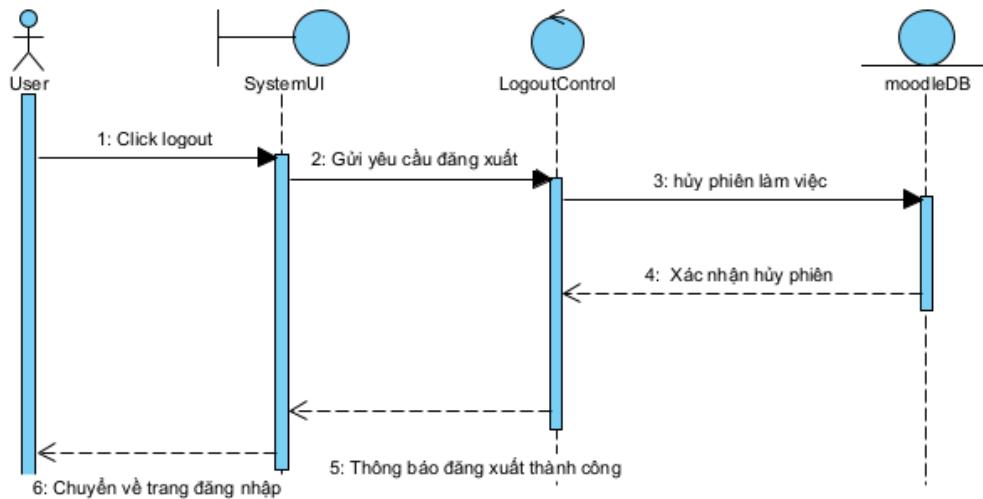
3.2.1. Thiết kế hoạt động của một số chức năng cơ bản

Trước khi đi vào thiết kế hoạt động của Plugin AI Chấm điểm, Nhóm có trình bày thiết kế hoạt động của một số chức năng cơ bản của Moodle bằng Biểu đồ tuần tự:

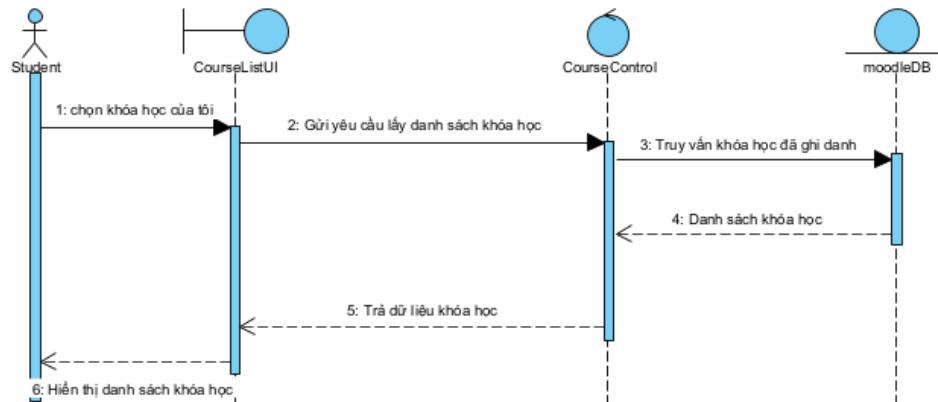
- Biểu đồ tuần tự Chức năng đăng nhập



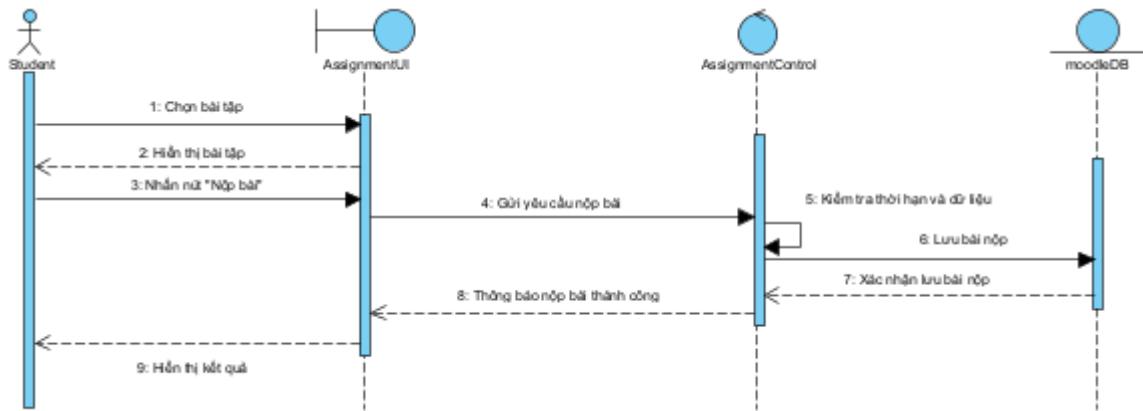
- Biểu đồ tuần tự Chức năng đăng xuất:



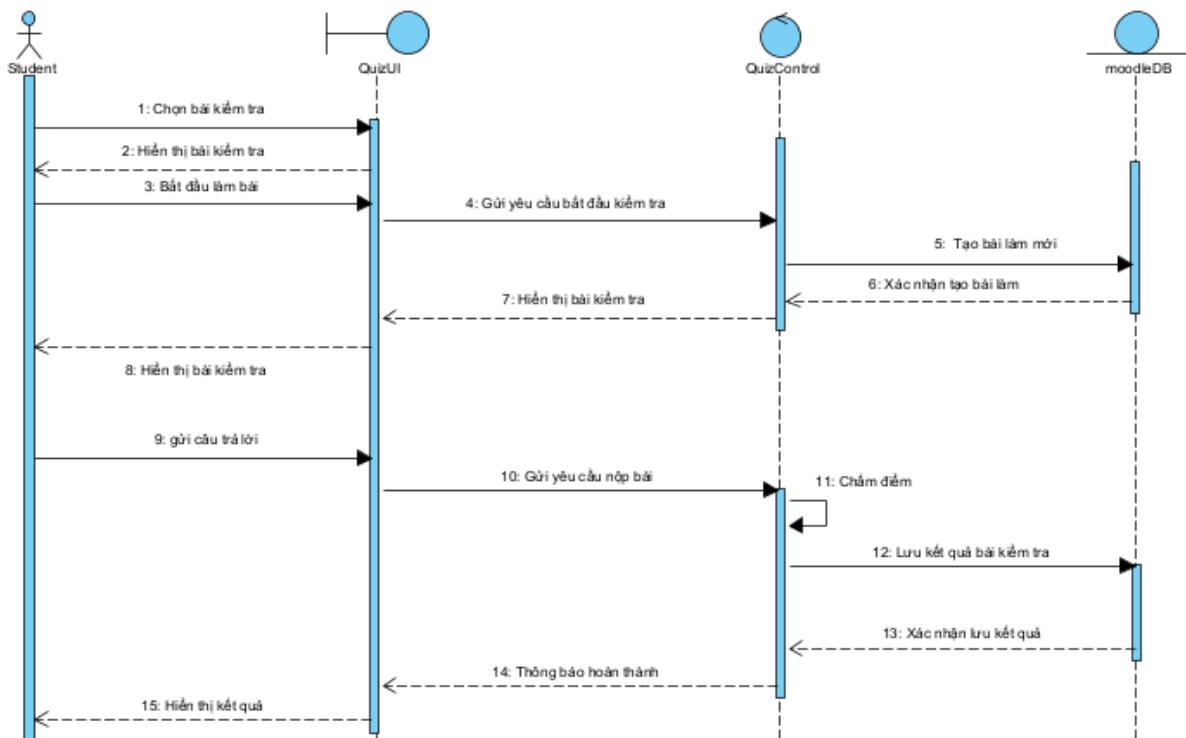
- Biểu đồ tuần tự Chức năng Xem danh sách khóa học đã tham gia:



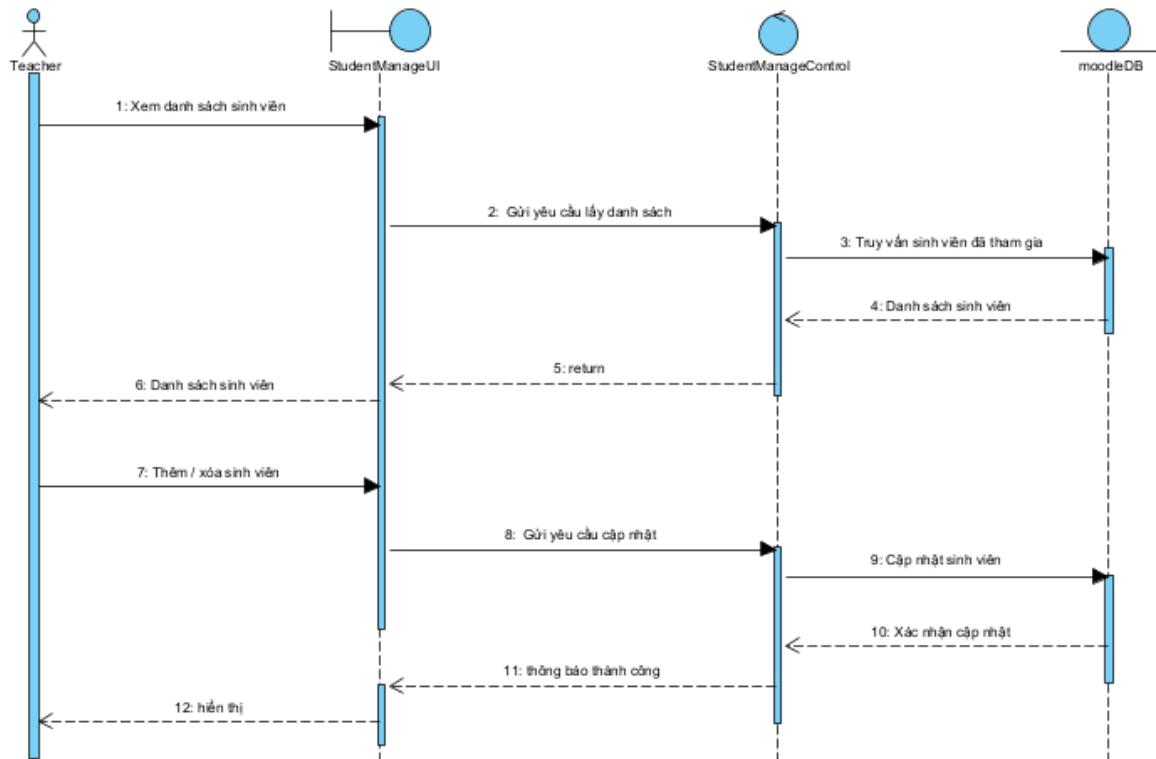
- Biểu đồ tuần tự Chức năng Xem bài giảng và tài liệu:



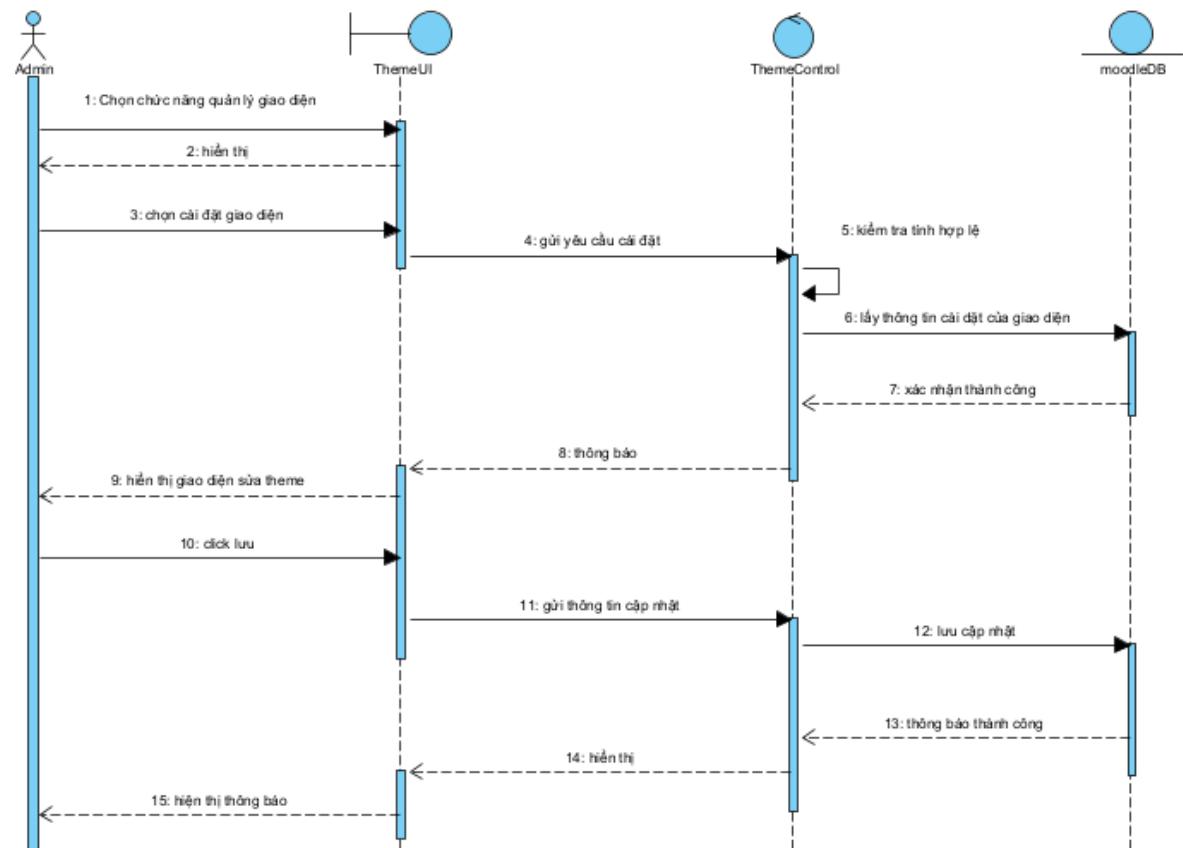
- Biểu đồ tuần tự Chức năng Tham gia làm bài kiểm tra:



- Biểu đồ tuần tự Chức năng Quản lý sinh viên trong khoá học:



- Biểu đồ tuần tự Sửa giao diện của Moodle:



3.2.2. Thiết kế hoạt động của Plugin AI chấm điểm

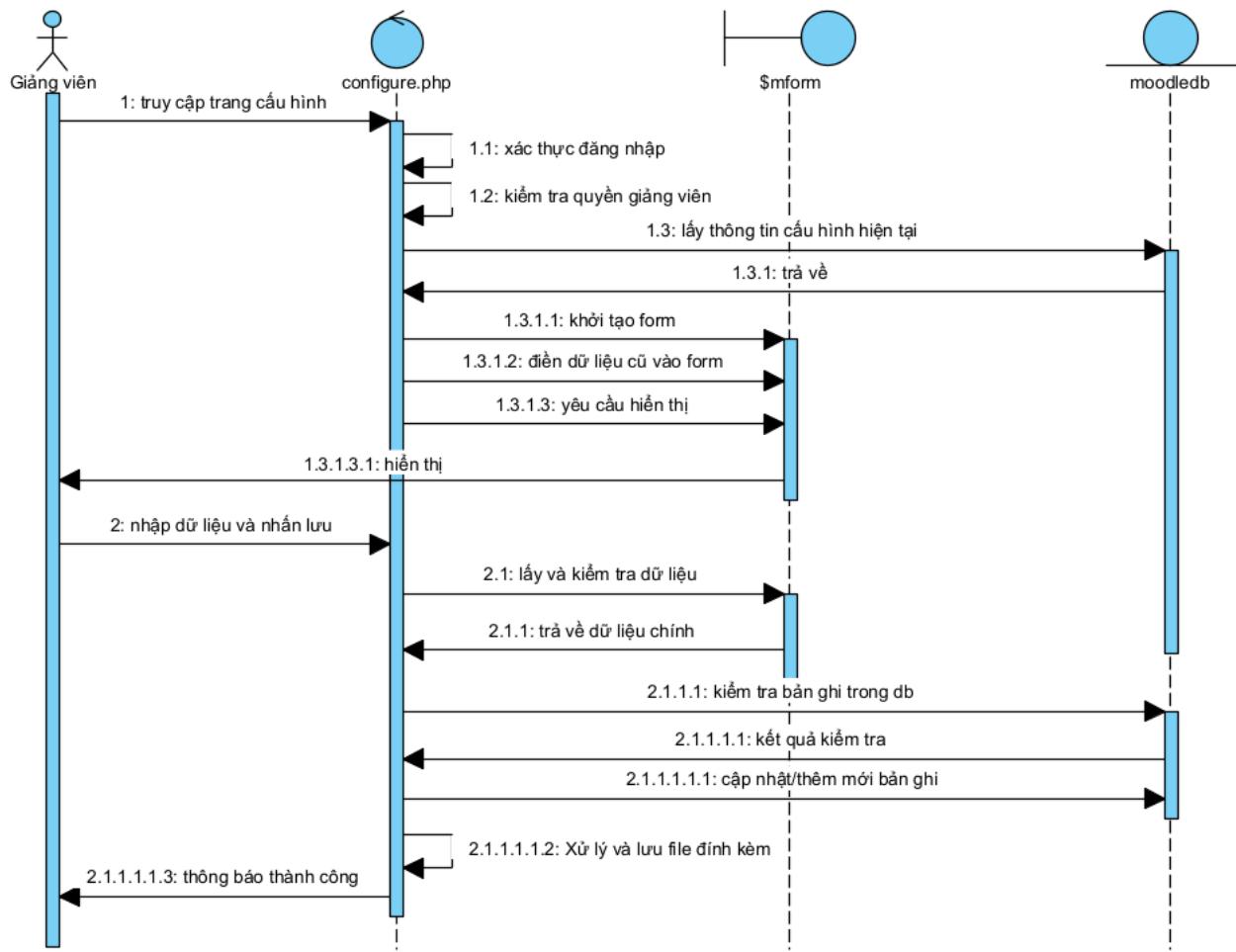
Use case 1: Cấu hình chức năng chấm điểm

Kịch bản hoạt động:

1. Teacher gửi yêu cầu GET tới local/aigrading/configure.php (kèm tham số assignid, cmid).
2. local/aigrading/configure.php gọi phương thức require_login() để xác thực session người dùng.
3. local/aigrading/configure.php gọi phương thức require_capability('moodle/course:manageactivities') để kiểm tra quyền hạn giảng viên.
4. local/aigrading/configure.php gọi \$DB -> get_record('local_aigrading_config') để lấy thông tin cấu hình hiện tại từ cơ sở dữ liệu.
5. \$DB trả về kết quả truy vấn (đối tượng \$config nếu có, hoặc false nếu chưa có).
6. local/aigrading/configure.php khởi tạo đối tượng \$mform bằng lệnh new config_form().
7. local/aigrading/configure.php gọi \$mform -> set_data(\$default_data) để điền dữ liệu cũ (nếu có) vào form.
8. local/aigrading/configure.php gọi \$mform -> display() để yêu cầu render giao diện.
9. \$mform hiển thị mã HTML của giao diện cấu hình cho Teacher.
10. Teacher nhập dữ liệu (Prompt, file mẫu) và thực hiện submit form (POST request) tới local/aigrading/configure.php.
11. local/aigrading/configure.php gọi \$mform -> get_data() để lấy và kiểm tra tính hợp lệ của dữ liệu vừa submit.
12. local/aigrading/configure.php gọi \$DB -> get_record('local_aigrading_config') một lần nữa để xác định bản ghi đã tồn tại hay chưa.

[Rẽ nhánh xử lý] Dựa vào kết quả kiểm tra ở bước 12:

- Trường hợp 1 (Đã có): local/aigrading/configure.php gọi \$DB -> update_record('local_aigrading_config') để cập nhật dữ liệu.
 - Trường hợp 2 (Chưa có): local/aigrading/configure.php gọi \$DB -> insert_record('local_aigrading_config') để thêm mới dữ liệu.
13. local/aigrading/configure.php gọi phương thức file_save_draft_area_files() để xử lý lưu file đính kèm từ vùng nhập.
 14. local/aigrading/configure.php gọi phương thức redirect() để tải lại trang và hiển thị thông báo "Lưu thành công".

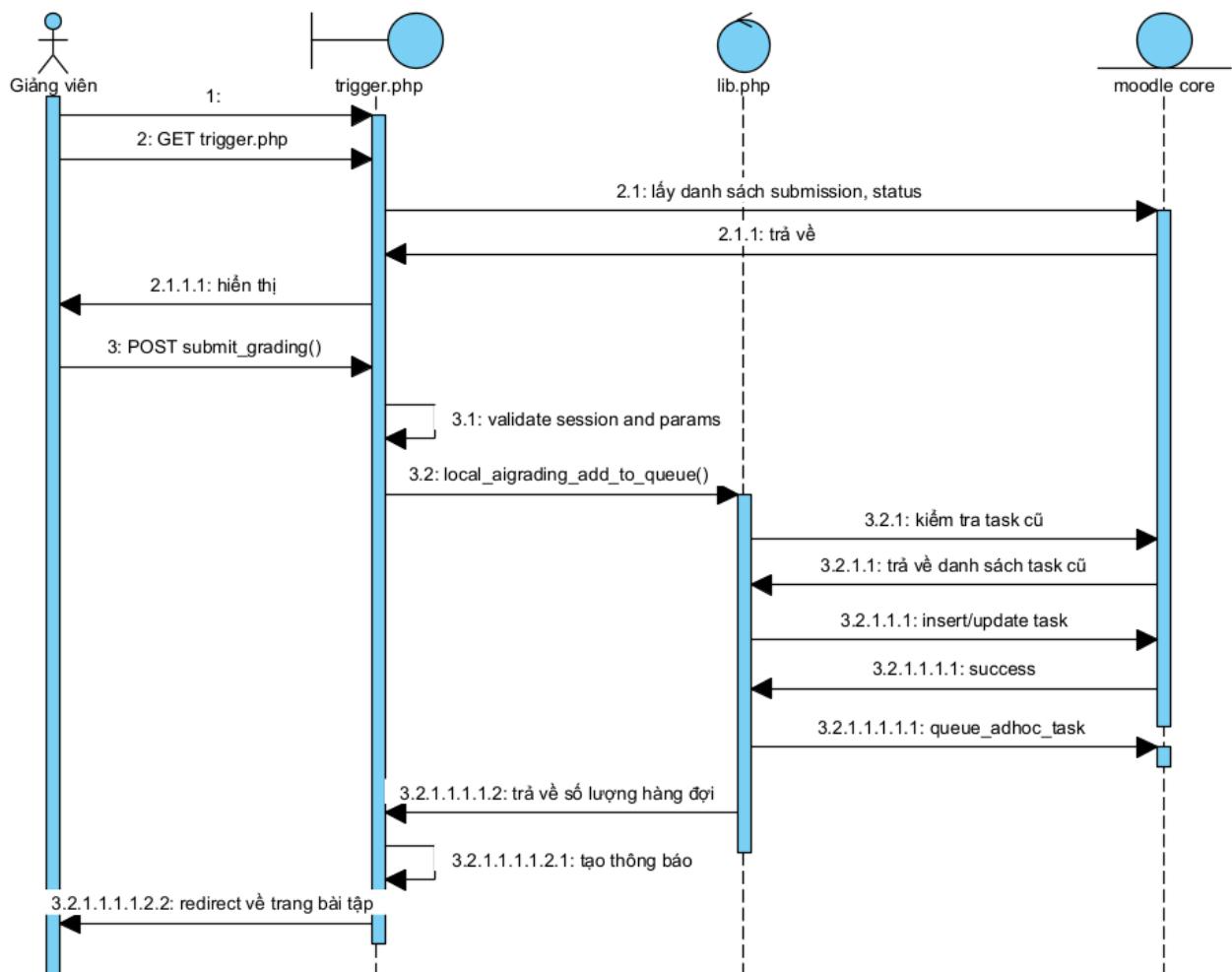


Hình 3.1. Biểu đồ tuần tự Câu hình chấm điểm AI

Use case 2: Gửi yêu cầu chấm điểm từ AI Grading

Kịch bản hoạt động:

1. Giảng viên truy cập trang chấm điểm trigger.php.
2. trigger.php (Boundary) tải danh sách sinh viên từ Moodle Core (Entity) để hiển thị giao diện chọn bài.
3. Giảng viên tích chọn các bài nộp và nhấn nút "Gửi đi chấm AI" (Submit Form).
4. trigger.php (Boundary) nhận yêu cầu POST, kiểm tra hợp lệ (Session key, danh sách bài chọn).
5. trigger.php gọi hàm điều khiển local_aigrading_add_to_queue nằm trong lib.php (Control).
6. lib.php (Control) truy vấn Moodle Core (Entity) để kiểm tra trạng thái các bài nộp hiện tại (tránh trùng lặp hoặc xử lý timeout).
7. lib.php gửi lệnh xuống Moodle Core (Entity) để INSERT hoặc UPDATE các bản ghi vào bảng local_aigrading_tasks với trạng thái status = 0 (In Queue/Pending).
8. lib.php gửi lệnh xuống Moodle Core (System) để lên lịch chạy task (\core\task\manager::queue_adhoc_task), báo hiệu cho hệ thống biết có việc cần làm.
9. lib.php trả về kết quả số lượng đã queue cho trigger.php.
10. trigger.php hiển thị thông báo thành công (Notify) và điều hướng (Redirect) Giảng viên quay lại trang danh sách bài tập Assignment.



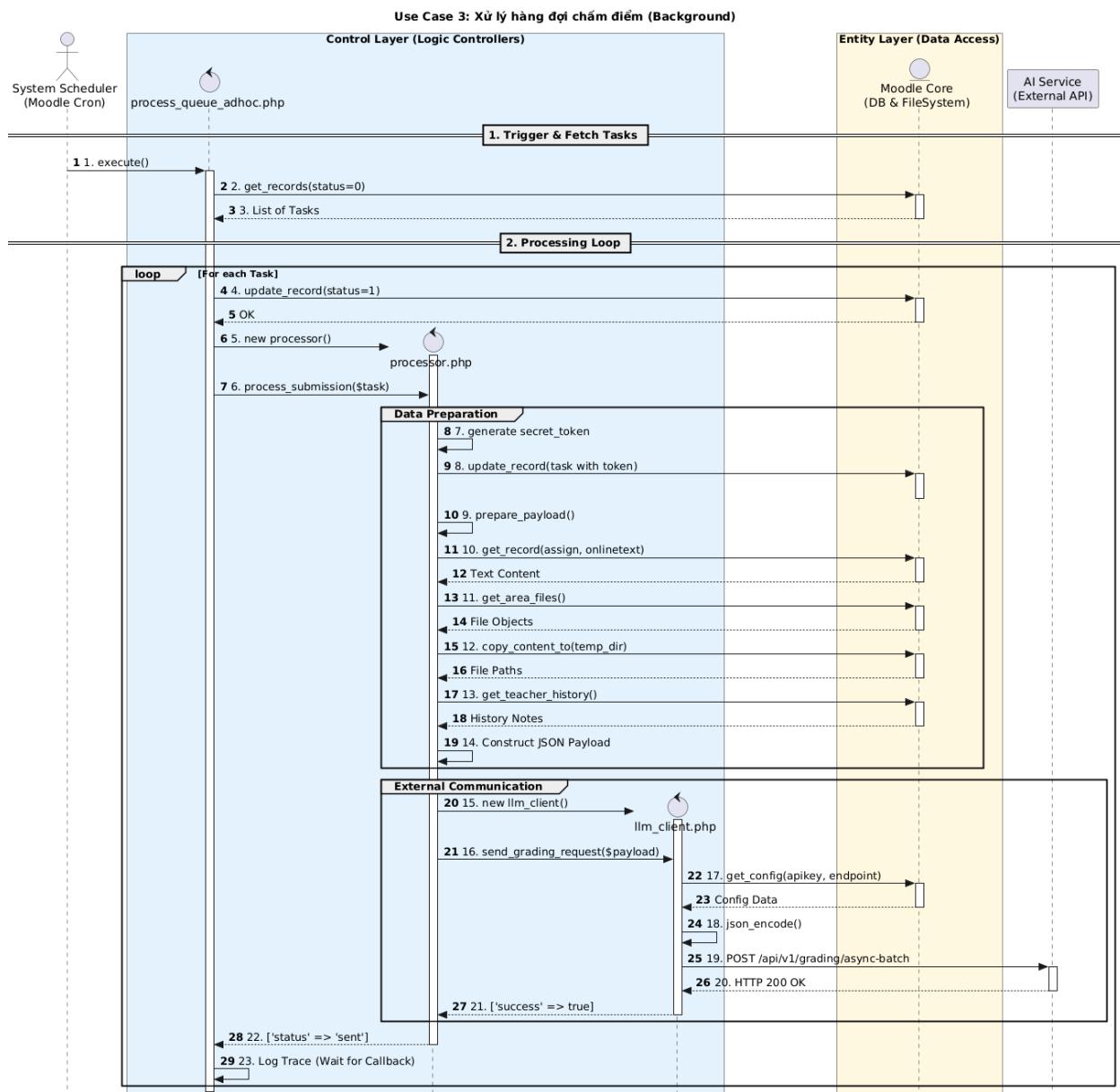
Hình 3.2. Biểu đồ tuần tự gửi yêu cầu chấm điểm

Use case 3: Nhận và xử lý yêu cầu chấm điểm từ moodle

Kịch bản hoạt động:

1. Moodle Cron (Hệ thống) gửi tín hiệu kích hoạt phương thức `execute()` của `process_queue_adhoc.php`.
2. `process_queue_adhoc.php` gửi truy vấn đến Moodle Core (Database) để lấy danh sách các task có `status = 0` (Pending).
3. Moodle Core trả về danh sách các đối tượng task cho `process_queue_adhoc.php`.
4. `process_queue_adhoc.php` gửi lệnh cập nhật `status = 1` (Processing) cho task hiện tại xuống Moodle Core.
5. `process_queue_adhoc.php` khởi tạo một đối tượng mới từ lớp `processor.php`.
6. `process_queue_adhoc.php` gọi phương thức `process_submission($task)` của `processor.php`.
7. `processor.php` tạo chuỗi `secret_token` ngẫu nhiên.
8. `processor.php` gửi lệnh cập nhật `secret_token` vào task trong Moodle Core.
9. `processor.php` gọi phương thức nội bộ `prepare_payload()`.
10. `processor.php` truy vấn Moodle Core để lấy nội dung bài tập (`intro`) và bài làm sinh viên (`onlinetext`).

11. `processor.php` yêu cầu Moodle Core (File API) cung cấp danh sách file đính kèm.
12. `processor.php` thực hiện lệnh `copy_content_to` thông qua Moodle Core để sao chép file ra thư mục tạm (`local_aigrading_temp`).
13. `processor.php` truy vấn Moodle Core để lấy lịch sử chấm của giáo viên (`get_teacher_history`).
14. `processor.php` đóng gói toàn bộ dữ liệu thành mảng `$payload`.
15. `processor.php` khởi tạo đối tượng từ lớp `l1m_client.php`.
16. `processor.php` gọi phương thức `send_grading_request($payload)` của `l1m_client.php`.
17. `l1m_client.php` lấy cấu hình API (URL, Key) từ Moodle Core.
18. `l1m_client.php` mã hóa payload thành JSON.
19. `l1m_client.php` gửi POST request tới AI Service (External).
20. AI Service trả về mã HTTP 200 OK (xác nhận đã nhận).
21. `l1m_client.php` trả về kết quả thành công (`['success' => true]`) cho `processor.php`.
22. `processor.php` trả về trạng thái (`['status' => 'sent']`) cho `process_queue_adhoc.php`.
23. `process_queue_adhoc.php` ghi log (`mtrace`) xác nhận việc gửi đã hoàn tất.



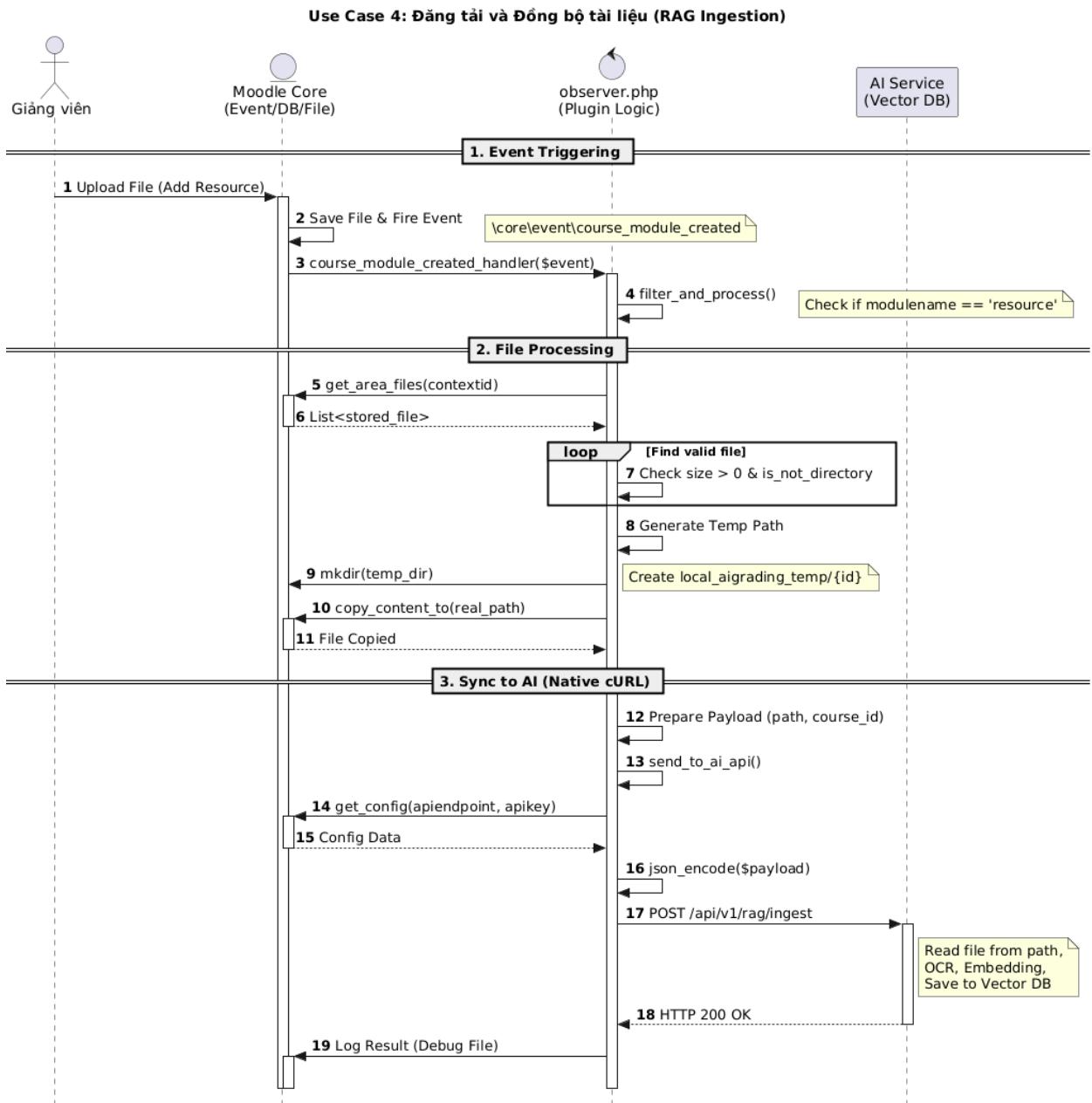
Hình 3.3. Biểu đồ tuần tự Xử lý hàng đợi và chấm điểm

Use case 4: Đăng tải và đồng bộ tài liệu

Kịch bản hoạt động:

1. Giảng viên thực hiện hành động tải file lên khóa học (*Upload Resource*).
2. Moodle Core lưu file và kích hoạt sự kiện `\core\event\course_module_created`.
3. Moodle Core gọi phương thức `course_module_created_handler($event)` của `observer.php`.
4. `observer.php` gọi phương thức nội bộ `process_log()` để ghi log bắt đầu sự kiện.
5. `observer.php` kiểm tra dữ liệu sự kiện: Nếu `modulename != 'resource'`, quy trình kết thúc.
6. `observer.php` lấy `contextid` và `courseid` từ dữ liệu sự kiện.
7. `observer.php` gọi `get_file_storage()` và yêu cầu Moodle Core lấy danh sách file (`get_area_files`) thuộc context đó.
8. Moodle Core trả về danh sách các đối tượng file (`stored_file`).

9. *observer.php* lặp qua danh sách để tìm file hợp lệ (không phải thư mục, dung lượng > 0).
10. *observer.php* tạo đường dẫn thư mục tạm (*local_aigrading_temp/{request_id}*).
11. *observer.php* yêu cầu hệ thống (PHP *mkdir*) tạo thư mục vật lý nếu chưa tồn tại.
12. *observer.php* làm sạch tên file bằng hàm *clean_param*.
13. *observer.php* gọi phương thức *copy_content_to(\$real_path)* của đối tượng file để sao chép nội dung ra thư mục tạm.
14. *observer.php* ghi log xác nhận file đã sẵn sàng (*FILE READY*).
15. *observer.php* chuẩn bị mảng dữ liệu *\$payload* (gồm đường dẫn file, *course_id*, *mime_type*).
16. *observer.php* gọi phương thức nội bộ *send_to_ai_api(\$payload)*.
17. *observer.php* truy vấn Moodle Core (Config) để lấy *apiendpoint* và *apikey*.
18. *observer.php* mã hóa payload thành JSON.
19. *observer.php* khởi tạo *curl* (Native PHP) với cấu hình bỏ qua SSL.
20. *observer.php* gửi POST request tới AI Service (*/api/v1/rag/ingest*).
21. AI Service xử lý và trả về HTTP Status Code (ví dụ: 200).
22. *observer.php* ghi log kết quả phản hồi (API RESPONSE CODE) vào file log debug.



Hình 3.4. Biểu đồ tuần tự Đăng tải và Đồng bộ dữ liệu

3.3. Tổng kết chương III

Chương III đã trình bày chi tiết quy trình hiện thực hóa hệ thống từ các bản thiết kế đã được xây dựng. Nhóm thực hiện đã triển khai thành công hạ tầng mạng phân tách lớp (3-Tier Architecture) sử dụng công nghệ Docker Container, đảm bảo sự ổn định cho Moodle Core và tính an toàn cho máy chủ chấm code VPL Jail System.

Đặc biệt, quy trình phát triển Plugin chấm điểm AI (local_aigrading) đã được mô tả tường minh thông qua các biểu đồ tuần tự, làm rõ cơ chế giao tiếp không đồng bộ giữa Moodle PHP, Cron Task và AI Service (FastAPI). Việc tích hợp thành công các thành phần này không chỉ giải quyết bài toán chấm điểm tự động mà còn đảm bảo hiệu năng hệ thống khi xử lý đồng thời nhiều yêu cầu.

CHƯƠNG IV. KẾT LUẬN

4.1 Kết quả đạt được:

Sau quá trình nghiên cứu và triển khai, đề tài "Triển khai và bổ sung plugin cho hệ thống quản trị học tập Moodle trên máy chủ Bộ môn Khoa học máy tính" đã đạt được những kết quả quan trọng sau:

- **Về mặt hạ tầng:** Đã xây dựng thành công hệ thống LMS dựa trên kiến trúc Microservices lai (Hybrid), kết hợp Docker cho Web Server và môi trường Native Linux cho Execution Server, giải quyết triệt để vấn đề xung đột tài nguyên và bảo mật của hệ thống cũ.
- **Về mặt tính năng:** Đã phát triển và tích hợp thành công plugin local_aigrading. Hệ thống có khả năng tự động chấm điểm và đưa ra nhận xét chi tiết (Feedback) cho các bài tập tự luận dựa trên công nghệ RAG và mô hình ngôn ngữ lớn Qwen 2.5, giúp giảm tải đáng kể khối lượng công việc cho giảng viên.
- **Về trải nghiệm người dùng:** Đã tùy biến giao diện Moodle (Child Theme) mang đặc trưng thương hiệu của Học viện, tối ưu hóa giao diện soạn thảo code (Code Editor) và bảng điều khiển (Dashboard), giúp sinh viên dễ dàng tiếp cận và thao tác trên mọi thiết bị.

4.2 Hạn chế của hệ thống

Bên cạnh những kết quả đạt được, hệ thống vẫn còn tồn tại một số hạn chế cần khắc phục:

- **Tốc độ phản hồi của AI:** Do triển khai mô hình LLM cục bộ (On-premise) trên phần cứng giới hạn, tốc độ sinh phản hồi (Generation speed) còn chậm khi gặp các bài làm quá dài hoặc khi số lượng request đồng thời tăng đột biến.
- **Độ chính xác của ngữ cảnh (Context):** Kỹ thuật RAG hiện tại mới chỉ xử lý tốt các tài liệu dạng văn bản (Text/PDF). Với các đề bài chứa hình ảnh hoặc công thức toán học phức tạp, khả năng "hiểu" của AI còn hạn chế.

4.3 Định hướng phát triển hệ thống

Để nâng cao chất lượng và khả năng ứng dụng thực tiễn, nhóm đề xuất các hướng phát triển trong tương lai:

- **Nâng cấp phần cứng và Mô hình:** Trang bị thêm GPU chuyên dụng để triển khai các mô hình lớn hơn (như Qwen 7B hoặc Llama 3 8B) nhằm cải thiện tốc độ và độ chính xác trong suy luận.

- **Tối ưu hóa RAG:** Tích hợp thêm module OCR (Nhận dạng quang học) để AI có thể đọc hiểu đề bài dạng ảnh, và sử dụng Re-ranking (Sắp xếp lại) để cải thiện chất lượng dữ liệu đầu vào cho AI.
- **Phát triển Mobile App:** Xây dựng ứng dụng di động riêng biệt (Mobile App) kết nối với API của hệ thống, hỗ trợ thông báo đẩy (Push Notification) và làm bài tập trắc nghiệm nhanh.
- **Mở rộng phạm vi chấm:** Nghiên cứu áp dụng AI để chấm các dạng bài tập khác như báo cáo thực tập, tiểu luận chuyên ngành thay vì chỉ giới hạn ở các bài tập lập trình và câu hỏi ngắn.

PHỤ LỤC:

CÁU HÌNH MAIL SERVER VÀ TÍCH HỢP ĐĂNG NHẬP (OAUTH2)

Phần này mô tả quy trình thiết lập dịch vụ gửi thư điện tử (SMTP) để phục vụ tính năng "Quên mật khẩu" và cấu hình xác thực OAuth2 cho phép sinh viên/giảng viên đăng nhập bằng tài khoản email Edu (@ptit.edu.vn) mà không cần tạo tài khoản thủ công.

Phụ lục 1: Cấu hình SMTP (Outgoing Mail Configuration)

Moodle cần một máy chủ SMTP để gửi email thông báo và email xác nhận thay đổi mật khẩu. Trong phạm vi đồ án, chúng tôi sử dụng **Google SMTP Server** (Gmail) vì tính ổn định và miễn phí.

Bước 1: Chuẩn bị tài khoản Email gửi

- Do chính sách bảo mật mới của Google, không thể sử dụng mật khẩu đăng nhập thông thường. Cần tạo **App Password (Mật khẩu ứng dụng)**.
- Truy cập Google Account > Security > 2-Step Verification > App passwords. Tạo một mật khẩu mới và lưu lại chuỗi 16 ký tự này.

Bước 2: Cấu hình trong Moodle

1. Đăng nhập quyền Admin.
2. Truy cập: Site administration (Quản trị hệ thống) > Server (Máy chủ) > Outgoing mail configuration (Cấu hình thư đi).
3. Điền các thông số sau:

Tham số	Giá trị cấu hình	Giải thích
SMTP hosts	smtp.gmail.com:587	Máy chủ Google và cổng TLS
SMTP security	TLS	Giao thức bảo mật
SMTP auth type	LOGIN	Kiểu xác thực
SMTP username	your_email@gmail.com	Email dùng để gửi thông báo
SMTP password	xxxx xxxx xxxx xxxx	App Password 16 ký tự (Không dùng pass thường)
No-reply address	noreply@cs-lms.ptit.edu.vn	Email hiển thị ở mục người gửi

Bước 3: Kiểm tra thử nghiệm

- Tại trang cấu hình SMTP, kéo xuống dưới cùng tìm link "**Test outgoing mail configuration**".
- Nhập email cá nhân và nhấn **Send a test email**.
- *Kết quả:* Nếu nhận được thông báo "Pass", cấu hình thành công.

Phụ lục 2: Cấu hình tích hợp đăng nhập bằng Microsoft Office 365 (Mail edu)

Phụ lục này hướng dẫn chi tiết quy trình đăng ký ứng dụng trên Microsoft Azure và cấu hình Moodle để cho phép sinh viên/giảng viên đăng nhập bằng tài khoản email trường (@ptit.edu.vn) thông qua giao thức OAuth 2.0.

P2.1. Đăng ký ứng dụng trên Microsoft Azure Portal

Để Moodle có thể "nói chuyện" với hệ thống xác thực của Microsoft, ta cần tạo một "App registration" trên nền tảng đám mây của Microsoft.

Bước 1: Truy cập Azure Portal

- Truy cập: <https://portal.azure.com/>

- Đăng nhập bằng tài khoản Microsoft (Email sinh viên PTIT của bạn).

Bước 2: Tạo ứng dụng mới

1. Tại thanh tìm kiếm, gõ "App registrations" và chọn dịch vụ này.
2. Nhấn nút + New registration.
3. Điền thông tin:
 - o **Name:** PTIT LMS Login (hoặc tên tùy ý).
 - o **Supported account types:** Chọn dòng "Accounts in any organizational directory (Any Microsoft Entra ID tenant - Multitenant)" (Để chấp nhận cả mail trường và các tổ chức khác nếu cần).
 - o **Redirect URI (Quan trọng):**
 - Chọn nền tảng: **Web**.
 - Nhập URL callback của Moodle.
Ví dụ: <https://cs-lms.ptit.edu.vn/admin/oauth2callback.php>
4. Nhấn **Register**.

Bước 3: Lấy thông tin xác thực (Client ID & Secret)

Sau khi tạo xong, hệ thống chuyển đến trang Overview.

1. **Copy Application (client) ID:** Lưu lại chuỗi mã này (Đây là *Client ID*).
2. Trên menu bên trái, chọn **Certificates & secrets**.
3. Chọn thẻ **Client secrets** > Bấm + New client secret.
 - o Description: MoodleKey.
 - o Expires: Chọn thời hạn (ví dụ: 12 months).
 - o Bấm **Add**.
4. **QUAN TRỌNG:** Copy ngay chuỗi ở cột **Value** (Đây là *Client Secret*). *Lưu ý: Nếu tải lại trang, chuỗi này sẽ bị ẩn vĩnh viễn.*

P2.2. Cấu hình Dịch vụ OAuth 2 trên Moodle

Quay trở lại trang quản trị Moodle để khai báo thông tin vừa lấy được.

Bước 1: Tạo dịch vụ Microsoft

1. Đăng nhập Admin Moodle.
2. Truy cập: **Site administration > Server > OAuth 2 services**.
3. Nhấn nút **Create new Microsoft service**. (*Lưu ý: Chọn đúng Microsoft, không chọn Google*).

Bước 2: Điền thông số kết nối

- **Name:** Đăng nhập bằng Email PTIT.
- **Client ID:** (*Dán Application ID lấy ở bước C.1*).
- **Client secret:** (*Dán Client Secret Value lấy ở bước C.1*).
- **Show on login page:** Đánh dấu tích (✓).
- Nhấn **Save changes**.

Bước 3: Giới hạn tên miền (Chỉ cho phép mail trường): Để ngăn chặn người dùng sử dụng mail Outlook cá nhân (@outlook.com) đăng nhập vào hệ thống:

1. Tại danh sách dịch vụ OAuth 2, nhấn biểu tượng **Configure (Hình bánh răng)** cạnh dịch vụ vừa tạo.
2. Tìm mục **Login domains**.
3. Nhập: ptit.edu.vn.
4. Nhấn **Save changes**.

P2.3. Đồng bộ dữ liệu người dùng (Data Mapping): Moodle cần biết trường dữ liệu nào của Microsoft tương ứng với Username, Email...

1. Tại trang **OAuth 2 services**, nhấn vào biểu tượng **Edit mappings (Hình mũi tên 2 chiều)** của dịch vụ Microsoft.
2. Đảm bảo các trường được map như sau (Mặc định template Microsoft đã làm tốt, nhưng cần kiểm tra lại):

Trường bên ngoài (External field)	Trường nội bộ (Internal field)
givenName	firstname
surname	lastname
userPrincipalName	email
userPrincipalName	username

DANH MỤC TÀI LIỆU THAM KHẢO

Tài liệu, giáo trình:

- [1] PGS.TS Trần Đình Quế, Phân tích và thiết kế hệ thống thông tin, Học viện Công nghệ Bưu chính Viễn thông, 2014.
- [2] J. Rodríguez-del-Pino, E. Rubio-Royo, and Z. J. Hernández-Figueroa, "Virtual Programming Lab: The execution server," *XVIII Jornadas de Enseñanza Universitaria de la Informática*, 2012.

Trang web:

- [3] Moodle HQ, "Moodle Developer Documentation," [Online].
Available: <https://docs.moodle.org/dev/>.
- [4] ULPGC, "VPL Jail System 3.0.1 Service Documentation," [Online].
Available: <https://vpl.dis.ulpgc.es/documentation/vpl-jail-system-3.0.1/>.
- [5] Docker Inc., "Docker Documentation," [Online].
Available: <https://docs.docker.com/>.
- [6] Qwen Team, "Qwen2.5: A Party of Foundation Models," *Alibaba Cloud*, Sep. 2024. [Online]. Available: <https://qwenlm.github.io/>
- [7] LangChain AI, "Introduction to LangChain," [Online].
Available: https://python.langchain.com/docs/get_started/introduction.
- [8] P. Lewis et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," *NeurIPS*, 2020. [10] FastAPI, "FastAPI Web Framework Documentation," [Online].
Available: <https://fastapi.tiangolo.com/>.