

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

KHOA CÔNG NGHỆ THÔNG TIN I



## BÁO CÁO ĐỒ ÁN TỐT NGHIỆP

ĐỀ TÀI:

Thiết kế và triển khai hệ thống Web mô phỏng lớp học ảo tích hợp  
AI Agent

Giảng viên hướng dẫn : PGS. TS. Trần Đình Quê  
Sinh viên thực hiện : Phạm Huy Thái – B21DCAT174  
Đỗ Quang Tuấn – B21DCVT443  
Phạm Thanh Tùng - B21DCCN773  
Bùi Trung Dũng - B21DCVT139

Lớp : E21CNPM04  
Khóa : 2021 - 2026  
Hệ : Đại học chính quy

Hà Nội – 2025

## LỜI CẢM ƠN

Để đạt được kết quả học tập như ngày hôm nay, em xin bày tỏ lòng biết ơn sâu sắc tới toàn thể quý Thầy Cô Khoa Công Nghệ Thông Tin 1 – Học viện Công Nghệ Bưu Chính Viễn Thông nói riêng và quý Thầy Cô của Học viện nói chung. Quý Thầy Cô đã tận tâm truyền đạt cho em không chỉ những kiến thức chuyên môn quý báu, mà còn chia sẻ nhiều kinh nghiệm thực tiễn, giúp em có được nền tảng vững chắc để học tập và rèn luyện.

Đặc biệt, em xin gửi lời tri ân chân thành đến Thầy **PGS.TS. Trần Đình Quế**, người đã trực tiếp hướng dẫn, chỉ bảo tận tình và định hướng cho em trong suốt quá trình thực hiện đồ án. Những chỉ dẫn và sự tận tâm của Thầy đã giúp em định hình rõ ràng con đường nghiên cứu, giải quyết được những khó khăn gặp phải và hoàn thiện đồ án này một cách trọn vẹn hơn.

Bên cạnh đó, em cũng xin gửi lời cảm ơn sâu sắc đến gia đình và những người bạn thân thiết – những người luôn ở bên, động viên, tạo điều kiện tốt nhất và tiếp thêm cho em nghị lực vượt qua khó khăn.

Đồ án được thực hiện trong khoảng thời gian 3 tháng, là sự kết hợp giữa kiến thức đã học và quá trình tìm hiểu, nghiên cứu về lĩnh vực xử lý ngôn ngữ tự nhiên trong Công nghệ Thông tin. Đây là bước khởi đầu để em rèn luyện khả năng nghiên cứu, tư duy độc lập và tiếp cận sâu hơn với thực tiễn. Tuy nhiên, do hạn chế về thời gian và kiến thức, chắc chắn đồ án còn nhiều thiếu sót. Em rất mong nhận được sự góp ý chân thành từ quý Thầy Cô và các bạn để hoàn thiện hơn trong tương lai.

Em xin trân trọng gửi lời cảm ơn và lòng biết ơn sâu sắc nhất!

Hà Nội, tháng 12 năm 2025

# MỤC LỤC

DANH MỤC HÌNH ẢNH .....	7
DANH MỤC BẢNG BIỂU .....	9
DANH MỤC TỪ VIẾT TẮT .....	10
LỜI MỞ ĐẦU .....	12
Bối cảnh xã hội .....	12
Lý do chọn đề tài. ....	12
Nội dung thực hiện. ....	13
Cấu trúc đồ án.....	14
CHƯƠNG 1: TỔNG QUAN VỀ HỆ THỐNG WEB MÔ PHỎNG LỚP HỌC ẢO TÍCH HỢP AI AGENT .....	15
1. Tổng quan về hệ thống giáo dục và ứng dụng AI trong giáo dục .....	15
1.1. Tổng quan về các hình thức giáo dục hiện nay .....	15
1.2. Tổng quan về các hệ thống MOOC hiện nay .....	16
1.3. Xu hướng nổi bật của việc ứng dụng AI trong giáo dục .....	18
2. Kiến trúc hệ thống vi dịch vụ .....	19
3. Các công nghệ phát triển ứng dụng .....	21
3.1. Spring Framework .....	21
3.2. NestJS Framework.....	21
3.3. FastAPI Framework.....	22
3.4. Kong Gateway .....	22
3.5. NextJS Framework .....	23
4. Hệ quản trị cơ sở dữ liệu PostgreSQL .....	24
5. Message Broker RabbitMQ .....	24
6. Kết luận.....	25
CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG WEB MÔ PHỎNG LỚP HỌC ẢO TÍCH HỢP AI AGENT .....	26
1. Phân tích .....	26
1.1. Use Case Diagram .....	26
a. Usecase tổng quát .....	26
b. Tương tác với Teaching Agent Use Case Diagram.....	28
c. Làm quiz Use Case Diagram .....	30
d. Xem video bài giảng Use Case Diagram.....	32

e. Đăng ký khóa học .....	34
1.2. Activity Diagram .....	36
a. Đăng nhập .....	36
b. Tương tác với Teaching Agent.....	37
c. Quiz.....	38
1.3. Sequence Diagram.....	39
a. Người dùng đăng ký khóa học.....	39
b. Người dùng tương tác với Teaching Agent.....	39
c. Người dùng làm bài tập .....	40
d. Người dùng yêu cầu hệ thống phân tích đánh giá năng lực .....	40
e. Người dùng yêu cầu tạo MindMap.....	41
1.4. Biểu đồ lớp toàn hệ thống.....	41
a. Xác định các lớp của hệ thống.....	41
b. Xác định mối quan hệ giữa các thực thể .....	41
1.5. Phân rã hệ thống theo miền chức năng.....	43
1.6. Biểu đồ chi tiết từng dịch vụ .....	43
a. Biểu đồ lớp pha phân tích Auth-service .....	43
b. Biểu đồ lớp pha phân tích Course-service.....	44
2. Pha thiết kế .....	45
2.1. Thiết kế biểu đồ gói .....	45
2.2. Thiết kế biểu đồ lớp thực thể.....	50
2.3. Thiết kế cơ sở dữ liệu .....	50
2.4. Thiết kế kiến trúc microservice .....	51
2.5. Thiết kế chi tiết từng dịch vụ.....	53
a. Auth-service + User-service .....	53
b. Course-service .....	61
3. Kết luận.....	70
<b>CHƯƠNG 3: XÂY DỰNG HỆ THỐNG TRỢ GIẢNG ẢO AI AGENT .....</b>	<b>71</b>
1. Tổng quan về hệ thống AI Agent hỗ trợ học tập .....	71
1.1. Giới thiệu chung và Kiến trúc tổng thể .....	71
1.2. Mô tả Luồng hoạt động (System Workflow) .....	71
1.3. Chi tiết các Tính năng cốt lõi .....	72
a. Phân hệ Hội thoại Thông minh (Smart Interactive Chat) .....	72
b. Phân hệ Trực quan hóa Kiến thức (Automated Mindmap Generation) .....	73

c. Phân hệ Đánh giá và Phân tích (Progress Analyzer) .....	73
2. Quy trình xây dựng pipeline xử lý dữ liệu (Data Engineering) .....	74
2.1. Đặc tả và Phân tích Dữ liệu đầu vào .....	75
a. Cấu trúc dữ liệu thô .....	75
b. Hạn chế của phương pháp truyền thống .....	76
2.2. Quy trình Tiền xử lý và Chuẩn hóa (Preprocessing) .....	76
a. Trích xuất Metadata thông minh .....	76
b. Pipeline Làm sạch văn bản (Text Cleaning) .....	77
2.3. Chiến lược Phân đoạn Ngữ nghĩa (Semantic Chunking Strategy) .....	78
2.4. Mã hóa Vector và Lưu trữ Cơ sở tri thức (Vector Store) .....	79
a. Embedding .....	79
b. Metadata Enrichment (Làm giàu dữ liệu) .....	80
c. Lưu trữ (Storage) .....	80
2.5. Tổng kết quy trình .....	80
3. Xây dựng pipeline truy xuất và tối ưu hóa tìm kiếm (Retrieval Pipeline) .....	80
3.1. Chiến lược Tìm kiếm Lai (Hybrid Search Strategy) .....	81
3.2. Cơ chế Lọc Metadata (Metadata Filtering) .....	82
3.3. Kỹ thuật Re-ranking (Sắp xếp lại kết quả) .....	82
3.4. Quản lý Cửa sổ Ngữ cảnh (Context Window Management) .....	84
3.5. Tổng kết Pipeline Truy xuất .....	85
4. Thiết kế và hiện thực hóa kiến trúc agent hướng đồ thị .....	86
4.1. Tổng quan về LangGraph và Cơ sở lựa chọn Công nghệ .....	86
a. Giới thiệu về LangGraph .....	87
b. Hạn chế của Kiến trúc Chuỗi (Chain Architecture) truyền thống .....	87
4.2. Định nghĩa Trạng thái Hệ thống (Agent State Schema) .....	88
4.3. Thiết kế Chi tiết Các Nút Xử lý (Nodes Implementation) .....	90
a. Intent Node (Bộ phân loại ý định) .....	90
b. Metadata Node (Bộ làm giàu ngữ cảnh) .....	90
c. Retrieve Node (Bộ truy xuất kiến thức) .....	91
4.4. Cơ chế Điều hướng và Luồng Điều khiển (Routing Logic) .....	91
4.5. Kỹ thuật Prompt Engineering Chuyên sâu .....	92
4.6. Tích hợp Công cụ Mở rộng (Tool Calling Strategy) .....	92
a. Mindmap Generator Tool (Công cụ vẽ sơ đồ tư duy) .....	93
b. Learning Analyzer Tool (Công cụ phân tích học tập) .....	94

4.7. Tổng kết Kiến trúc Hệ thống .....	95
5. Thiết kế api và hiện thực hóa triển khai .....	96
5.1. Kiến trúc Đóng gói và Thiết kế API (API Design) .....	96
a. Cấu trúc Dự án và Phân tầng (Project Structure) .....	96
b. Đặc tả Chi tiết các Endpoint (API Specification).....	96
5.2. Hạ tầng Triển khai (Deployment Infrastructure).....	98
a. Tầng Dữ liệu (Data Layer) - Supabase (PostgreSQL & PgVector) .....	98
b. Tầng Ứng dụng (Compute Layer) - Railway (Docker Container) .....	98
5.3. Kết luận về Triển khai .....	99
6. Thực nghiệm và Đánh giá hiệu quả AI Agent.....	100
6.1. Phương pháp và Tiêu chí đánh giá .....	100
6.2. Các kịch bản kiểm thử chi tiết .....	100
7. Kết luận.....	105
<b>CHƯƠNG 4: HỆ THỐNG WEB MÔ PHỎNG LỚP HỌC ẢO TÍCH HỢP AI AGENT</b>	
.....	106
1. Cài đặt và triển khai hệ thống.....	106
1.1. Kiến trúc triển khai tổng thể .....	106
1.2. Quy trình triển khai chi tiết .....	106
a. Triển khai Cơ sở dữ liệu trên Supabase.....	106
b. Triển khai các Microservices trên Railway .....	106
c. Triển khai Frontend trên Vercel .....	107
2. Giao diện website.....	108
a. Giao diện trang chủ.....	108
b. Giao diện đăng nhập / đăng ký .....	108
c. Giao diện Dashboard. ....	109
d. Giao diện danh mục khóa học. ....	110
e. Giao diện báo cáo tiến trình học tập. ....	110
f. Giao diện hồ sơ người dùng.....	111
g. Giao diện các khóa học đã đăng ký.....	111
h. Giao diện xem chi tiết khóa học. ....	112
i. Giao diện lớp học chính của lớp học ảo. ....	112
k. Giao diện tham gia làm bài kiểm tra cuối bài học.....	116
l. Giao diện xem đánh giá năng lực cá nhân.....	118
3. Kết luận.....	118

KẾT LUẬN .....	119
TÀI LIỆU THAM KHẢO .....	121

# DANH MỤC HÌNH ẢNH

Hình 1: Kiến trúc vi dịch vụ .....	20
Hình 2: Usecase tổng quát .....	26
Hình 3: Biểu đồ use case chức năng hỏi đáp Teaching Agent .....	28
Hình 4: Biểu đồ use case chức năng làm quiz.....	30
Hình 5: Biểu đồ use case chức năng xem video bài giảng .....	32
Hình 6: Biểu đồ use case chức năng đăng ký khóa học .....	34
Hình 7: Biểu đồ hoạt động “Đăng nhập”.....	36
Hình 8: Biểu đồ hoạt động “Tương tác với Agent”.....	37
Hình 9: Biểu đồ hoạt động “Làm quiz”.....	38
Hình 10: Biểu đồ tuần tự “ Đăng ký khóa học” .....	39
Hình 11: Biểu đồ tuần tự “Teaching Agent” .....	39
Hình 12: Biểu đồ tuần tự “Làm quiz” .....	40
Hình 13: Biểu đồ tuần tự “Đánh giá năng lực người dùng” .....	40
Hình 14: Biểu đồ tuần tự “Tạo Mindmap” .....	41
Hình 15: Biểu đồ lớp thực thể pha phân tích.....	42
Hình 16: Biểu đồ lớp pha phân tích Auth-service .....	43
Hình 17: Biểu đồ lớp pha phân tích Course-service.....	44
Hình 18: Thiết kế tổng thể hệ thống .....	45
Hình 19 : Cấu hình chi tiết Auth Service và các public routes trong Kong Gateway.....	46
Hình 20 : Cấu hình chi tiết AI Service và các protected routes trong Kong Gateway (yêu cầu JWT validation). .....	47
Hình 21 : Cấu hình JWT plugin bảo vệ Course Service trong Kong Gateway (áp dụng cho tất cả protected routes).....	47
Hình 22 : Cấu hình CORS plugin toàn cục .....	48
Hình 23: Các Message Pattern .....	49
Hình 24: Biểu đồ lớp thực thể pha thiết kế.....	50
Hình 25: Biểu đồ cơ sở dữ liệu toàn hệ thống .....	51
Hình 26: Biểu đồ tổng quan toàn hệ thống .....	52
Hình 27: Biểu đồ gói pha thiết kế Auth-service và User-service .....	54
Hình 28: Auth Controller.....	55
Hình 29: User Controller .....	56
Hình 30: Code AuthServiceImpl .....	58
Hình 31: Code UserServiceImpl .....	59

Hình 32: Biểu đồ lớp pha thiết kế Auth-service và User-service.....	60
Hình 33: Lược đồ cơ sở dữ liệu Auth-service và User-service .....	60
Hình 34: Biểu đồ gói pha thiết kế Course Service .....	61
Hình 35: Course Controller .....	62
Hình 36: Lesson Controller .....	63
Hình 37: Code Coursevice.....	64
Hình 38: Code Lessonservice .....	66
Hình 39: Biểu đồ lớp pha thiết kế Course-service .....	68
Hình 40: Lược đồ cơ sở dữ liệu Course-service.....	69
Hình 41: Data Processing Pipeline .....	75
Hình 42: Sơ đồ Retrieval pipeline .....	81
Hình 43: Kiến trúc của AI Agent .....	86
Hình 44: LangGraph general flow.....	86
Hình 45: Giao diện trang chủ .....	108
Hình 46: Giao diện đăng nhập.....	108
Hình 47: Giao diện đăng ký.....	109
Hình 48: Giao diện Dashboard .....	109
Hình 49: Giao diện danh mục khóa học .....	110
Hình 50: Giao diện báo cáo tiến trình học tập.....	110
Hình 51: Giao diện hồ sơ người dùng .....	111
Hình 52: Giao diện các khóa học đã đăng ký .....	111
Hình 53: Giao diện xem chi tiết khóa học .....	112
Hình 54: Giao diện xem bài giảng và danh sách bài học .....	112
Hình 55: Chat tương tác hỏi và trả lời với Agent .....	113
Hình 56: Phụ đề bài giảng .....	114
Hình 57: Xem sơ đồ tư duy bài học .....	115
Hình 58: Hiển thị sơ đồ tư duy toàn màn hình .....	116
Hình 59: Giao diện Trước khi làm bài .....	116
Hình 60: Giao diện Bắt đầu làm bài .....	117
Hình 61: Giao diện Hoàn thành bài kiểm tra.....	117
Hình 62: Giao diện Xem kết quả .....	118
Hình 63: Giao diện Xem đánh giá năng lực .....	118

## **DANH MỤC BẢNG BIỂU**

Bảng 1 : Tác nhân và chức năng của hệ thống .....	26
Bảng 2 : Mô tả các use case hệ thống.....	27
Bảng 3 : Kịch bản chức năng hỏi đáp Teaching Agent.....	30
Bảng 4 : Kịch bản chức năng làm quiz.....	31
Bảng 5 : Kịch bản chức năng xem video bài giảng.....	33
Bảng 6 : Kịch bản chức năng đăng ký khóa học .....	35
Bảng 7 : Phân rã thành các miền chức năng.....	43
Bảng 8 : Các thành phần hệ thống.....	53
Bảng 9 : Trường thông tin cốt lõi của Agent State.....	89

# DANH MỤC TỪ VIẾT TẮT

STT	Từ viết tắt	Định nghĩa	Dịch/Ghi chú
1	ACID	Atomicity, Consistency, Isolation, Durability	Các thuộc tính đảm bảo tính toàn vẹn giao dịch trong cơ sở dữ liệu.
2	AI	Trí tuệ nhân tạo (Artificial Intelligence)	Được dùng xuyên suốt, đặc biệt là Generative AI (Trí tuệ nhân tạo tạo sinh).
3	ASGI	Asynchronous Server Gateway Interface	Giao diện cổng máy chủ bất đồng bộ (trong FastAPI).
4	CAGR	Compound Annual Growth Rate	Tốc độ Tăng trưởng Kép Hàng năm (trong báo cáo thị trường AI giáo dục).
5	CoT	Chain-of-Thought	Kỹ thuật suy luận từng bước (trong Prompt Engineering).
6	CNTT	Công nghệ Thông tin	Dùng trong ngữ cảnh Việt Nam.
7	DI	Dependency Injection	Tiêm phụ thuộc (trong NestJS và Spring).
8	DTO	Data Transfer Object	Đối tượng truyền dữ liệu (trong API Design).
9	EdTech	(Educational Technology)	Công nghệ Giáo dục, dùng trong bối cảnh chuyển đổi số.
10	HNSW	Hierarchical Navigable Small World	Thuật toán tìm kiếm trong pgVector.
11	ISR	Incremental Static Regeneration	Tái tạo tĩnh tăng dần (trong NextJS).
12	JWT	JSON Web Token	Dùng trong xác thực (ngữ cảnh API).
13	LLM	Large Language Model	Mô hình Ngôn ngữ Lớn (như GPT-4o).
14	LMS	Learning Management System	Hệ thống quản lý học tập.
15	MOOC	Massive Open Online Courses	Khóa học Trực tuyến Mở Rộng Quy Mô.

16	NLP	Natural Language Processing	Xử lý ngôn ngữ tự nhiên (đề cập gián tiếp trong AI).
17	OOP	Object-Oriented Programming	Lập trình hướng đối tượng (trong NestJS).
18	ORDBMS	Object-Relational Database Management System	Hệ quản trị cơ sở dữ liệu quan hệ đối tượng (cho PostgreSQL).
19	PaaS	Platform as a Service	Nền tảng như một dịch vụ (trong triển khai Railway).
20	RAG	Retrieval-Augmented Generation	Sinh tạo tăng cường truy xuất (kiến trúc cốt lõi của AI Agent).
21	SSG	Static Site Generation	Sinh tĩnh trang (trong NextJS).
22	SSR	Server-Side Rendering	Kết xuất phía máy chủ (trong NextJS).
23	UNESCO	(United Nations Educational, Scientific and Cultural Organization)	Tổ chức Liên Hợp Quốc về Giáo dục, Khoa học và Văn hóa (đề cập trong báo cáo).

# LỜI MỞ ĐẦU

## Bối cảnh xã hội

Trong bối cảnh cuộc Cách mạng Công nghiệp 4.0, chuyển đổi số trong giáo dục (EdTech) đã không còn là một lựa chọn mà trở thành một xu thế tất yếu. Đặc biệt, sự bùng nổ của Trí tuệ nhân tạo tạo sinh (Generative AI) trong giai đoạn 2023-2025 đã làm thay đổi căn bản cách thức dạy và học, chuyển dịch từ mô hình "truyền thụ một chiều" sang "cá nhân hóa trải nghiệm học tập" (Personalized Learning).

Thị trường và Xu hướng toàn cầu Theo báo cáo từ *Precedence Research*, quy mô thị trường AI trong giáo dục toàn cầu dự kiến sẽ đạt mức tăng trưởng ấn tượng, với dự báo đạt 112,30 tỷ USD vào năm 2034, phản ánh nhu cầu cấp thiết trong việc áp dụng công nghệ để nâng cao chất lượng đào tạo [1]. Các tổ chức quốc tế như UNESCO cũng đã ban hành các hướng dẫn toàn cầu về Generative AI trong giáo dục và nghiên cứu, nhấn mạnh việc sử dụng AI để hỗ trợ tư duy bậc cao và khả năng sáng tạo của người học, thay vì chỉ đơn thuần là tự động hóa các tác vụ cơ bản [2].

Vấn đề của các lớp học ảo truyền thống Mặc dù các hệ thống quản lý học tập (LMS) và lớp học ảo hiện tại đã giải quyết được bài toán về không gian và thời gian, chúng vẫn tồn tại một hạn chế lớn: thiếu sự tương tác cá nhân hóa. Giáo viên trong một lớp học ảo khó có thể theo sát và giải đáp thắc mắc cho từng học sinh theo thời gian thực (Real-time), dẫn đến việc đánh giá năng lực người học thường bị chậm trễ hoặc thiếu chính xác. Vấn đề này gợi nhớ đến "Vấn đề 2 Sigma của Bloom" (Bloom's 2 Sigma Problem), trong đó khẳng định rằng học sinh được kèm cặp 1-1 sẽ có kết quả tốt hơn 98% so với học sinh trong lớp học truyền thống.

Sự trỗi dậy của AI Tutor và Agentic Workflow Năm 2024 chứng kiến sự chuyển dịch từ các Chatbot hỏi đáp đơn giản sang các AI Agents có khả năng tự chủ (autonomous). Các hệ thống này không chỉ trả lời câu hỏi mà còn có khả năng suy luận, lập kế hoạch và thực hiện các tác vụ phức tạp như một gia sư thực thụ. Việc ứng dụng AI để phân tích dữ liệu học tập giúp tạo ra các lộ trình cải thiện năng lực cụ thể cho từng cá nhân, điều mà các phương pháp truyền thống khó thực hiện trên quy mô lớn [3].

Tại Việt Nam, chiến lược chuyển đổi số quốc gia cũng đang khuyến khích mạnh mẽ việc ứng dụng các công nghệ mới như AI, Big Data vào trường học để xây dựng mô hình giáo dục thông minh. Do đó, việc nghiên cứu và triển khai các hệ thống lớp học ảo tích hợp AI Agent là hoàn toàn phù hợp với bối cảnh thực tiễn và nhu cầu xã hội hiện nay.

## Lý do chọn đề tài.

Từ những phân tích về bối cảnh xã hội và hạn chế công nghệ nêu trên, đề tài "Thiết kế và triển khai hệ thống Web mô phỏng lớp học ảo tích hợp AI Agent" được lựa chọn thực hiện nhằm giải quyết bài toán cấp thiết: Cá nhân hóa giáo dục trên quy mô lớn thông qua AI.

Lý do chọn đề tài này xuất phát từ ba động lực chính:

Thứ nhất, yêu cầu về một "người thầy ảo" thông minh hơn: Các chatbot hiện tại thường trả lời rập khuôn và thiếu ngữ cảnh. Đề tài hướng tới việc xây dựng một AI Companion thực thụ, sử dụng kiến trúc Agentic RAG điều phối bởi LangGraph. Khác với các chuỗi xử lý tuyến tính, kiến trúc này cho phép Agent có tư duy vòng lặp (cyclic graphs), tự suy luận, kiểm tra và quyết định hành động tiếp theo, mang lại trải nghiệm tương tác tự nhiên và sâu sắc như một gia sư người thật.

Thứ hai, yêu cầu về đánh giá năng lực chính xác và tức thời: Trong giáo dục trực tuyến, việc đánh giá học sinh thường bị trễ (sau các bài kiểm tra). Đề tài đề xuất giải pháp sử dụng GPT-4o để phân tích liên tục hội thoại và hành vi học tập, từ đó đưa ra đánh giá năng lực (Competency Assessment) và lộ trình cải thiện ngay lập tức, lập đầy khoáng trống mà các hệ thống LMS truyền thống chưa làm được.

Thứ ba, ứng dụng công nghệ để tối ưu hóa việc tổng hợp kiến thức: Học sinh thường gặp khó khăn trong việc hệ thống hóa bài học. Bằng việc kết hợp pgvector cho tìm kiếm ngữ nghĩa và khả năng tóm tắt của LLM, hệ thống có khả năng tự động trích xuất các khái niệm cốt lõi (core concepts) để tạo Mindmap, giúp người học nắm bắt trọng tâm bài giảng một cách trực quan và logic nhất.

Tóm lại, việc thực hiện đề tài này không chỉ có ý nghĩa khoa học trong việc ứng dụng các công nghệ AI tiên tiến (LangGraph, Vector DB, LLM) mà còn mang lại giá trị thực tiễn cao, góp phần nâng cao chất lượng dạy và học trong kỷ nguyên số.

## Nội dung thực hiện.

Nghiên cứu được khởi đầu bằng việc phân tích toàn diện các yêu cầu chức năng của Hệ thống Lớp học ảo tích hợp AI Agent. Các chức năng trọng yếu được xác định bao gồm: quản lý ghi danh khóa học, môi trường học tập tương tác, trợ giảng ảo hỗ trợ cá nhân hóa, hệ thống theo dõi tiến độ học tập và cơ chế đánh giá năng lực cuối khóa.

### Thiết kế kiến trúc hệ thống.

Hệ thống được xây dựng dựa trên kiến trúc Microservices, nhằm đảm bảo tính độc lập giữa các dịch vụ và khả năng mở rộng linh hoạt (scalability) theo nhu cầu tái.

- **Tầng Backend:** Là sự kết hợp giữa hiệu năng của Spring Boot và tính linh hoạt của NestJS Framework, đảm bảo khả năng xử lý dữ liệu phức tạp.
- **Tầng Frontend:** Được phát triển trên nền tảng NextJS, tối ưu hóa hiệu suất hiển thị và nâng cao trải nghiệm người dùng (UX).

### Triển khai và phát triển hệ thống

Quá trình phát triển tập trung vào các phân hệ (module) cốt lõi gồm: Dịch vụ Khóa học (Course Service), Dịch vụ Người dùng (User Service) và trọng tâm là Dịch vụ AI Agent.

- Cơ chế hoạt động của Agent: Sử dụng LangGraph trong vai trò bộ điều phối (Orchestrator), cho phép quản lý luồng hành vi của Agent một cách thông minh.
- Xử lý dữ liệu học tập: Thay vì sử dụng các cơ sở tri thức tĩnh, hệ thống vận hành dựa trên cơ chế truy xuất thông tin từ dữ liệu ngữ cảnh (Contextual Data). Dữ liệu này được trích xuất và chuẩn hóa trực tiếp từ các bản ghi (Transcript) bài giảng. Phương pháp này giúp đảm bảo các phản hồi của Agent luôn được khu trú (grounding) chính xác vào nội dung bài học cụ thể, loại bỏ hiện tượng sai lệch thông tin.

### Thực nghiệm và Đánh giá hiệu quả

Hiệu năng của AI Agent được kiểm chứng chủ yếu thông qua phương pháp kiểm thử hộp đen (Black-box testing). Nhóm nghiên cứu đã thiết lập bộ kịch bản kiểm thử (Test scenarios) bao gồm các truy vấn người dùng (user queries) với đa dạng ngữ cảnh để đánh giá độ chính xác và khả năng phản hồi của Agent.

Kết quả: Nghiên cứu đã hoàn thiện một hệ thống Lớp học ảo tích hợp AI Agent đảm bảo các tiêu chuẩn về an toàn thông tin và khả năng mở rộng. Hệ thống đáp ứng thành công các mục tiêu đề ra: tối ưu hóa thời gian phản hồi hỗ trợ và nâng cao mức độ hài lòng của người học.

### Cấu trúc đồ án.

Đồ án bao gồm 4 chương:

- Chương 1: Tổng quan về hệ thống web mô phỏng lớp học ảo tích hợp AI Agent.
- Chương 2: Phân tích và thiết kế hệ thống web mô phỏng lớp học ảo tích hợp AI Agent.
- Chương 3: Xây dựng hệ thống trợ giảng ảo AI Agent.
- Chương 4: Hệ thống web mô phỏng lớp học ảo tích hợp AI Agent.

# **CHƯƠNG 1: TỔNG QUAN VỀ HỆ THỐNG WEB MÔ PHỎNG LỚP HỌC ẢO TÍCH HỢP AI AGENT**

## **1. Tổng quan về hệ thống giáo dục và ứng dụng AI trong giáo dục**

“Cách mạng Công nghiệp 4.0” đang diễn ra tại nhiều nước phát triển trên thế giới. Nó mang đến cho nhân loại cơ hội để thay đổi bộ mặt các nền kinh tế, ảnh hưởng lên nhiều mặt của mọi lĩnh vực tồn tại trên thế giới. Ngành giáo dục cũng như vậy, quá trình tiến lên “Cách mạng Công nghiệp 4.0”, cùng với sự phát triển không ngừng nghỉ của Trí tuệ nhân tạo (AI) đã làm thay đổi hoàn toàn cách dạy và học hiện nay. Cơ hội và thách thức có thể nói là rất lớn nhưng nó sẽ trở thành xu thế mới.

### **1.1. Tổng quan về các hình thức giáo dục hiện nay**

Giáo dục, ở cấp độ cơ bản, là một quá trình xã hội có tính chất truyền thừa, hệ thống hóa và lĩnh hội các kiến thức, kỹ năng, thói quen và giá trị văn hóa từ thế hệ này sang thế hệ khác. Quá trình này được thực hiện thông qua các hình thức có tổ chức như đào tạo, nghiên cứu và giảng dạy, nhằm phát triển năng lực cá nhân và đóng góp vào sự tiến bộ của xã hội [4].

Các hình thức giáo dục hiện nay:

#### **- Giáo dục Truyền thống**

- Định nghĩa: Đây là hình thức dạy và học trực tiếp, là mô hình chủ đạo trong hệ thống giáo dục chính quy [4].
- Đặc điểm: Yêu cầu người học và người dạy phải cùng hiện diện tại một địa điểm (ví dụ: lớp học, giảng đường) trong cùng một khoảng thời gian theo lịch trình cố định (đồng bộ).
- Ưu điểm: Thúc đẩy sự tương tác trực tiếp giữa người dạy và người học, tạo ra môi trường học tập xã hội hóa cao, giúp dễ dàng kiểm soát kỷ luật và đánh giá tức thời.
- Vai trò: Tiếp tục là nền tảng cho các cấp học từ phổ thông đến đại học, đặc biệt quan trọng trong việc truyền đạt các môn học yêu cầu thực hành và tương tác phức tạp.

#### **- E-Learning**

- Định nghĩa: Là một hệ thống học tập linh hoạt, cho phép giảng viên và học viên tham gia vào quá trình dạy-học, quản lý nội dung và đánh giá trên các nền tảng kỹ thuật số (máy tính, máy tính bảng, điện thoại thông minh) [4].
- Hoạt động: Giảng viên có thể truyền đạt nội dung thông qua bài giảng trực tiếp đồng bộ hoặc lưu trữ tài liệu học tập (video, hình ảnh, âm thanh, bài tập) trên hệ thống. Học viên có thể truy cập nội dung trực tuyến hoặc ngoại tuyến, tham gia các diễn đàn thảo luận, trao đổi với đồng nghiệp và thực hiện các bài kiểm tra đánh giá tự động.
- Ý nghĩa: E-Learning đóng vai trò then chốt trong việc phá vỡ rào cản không gian và thời gian, tối ưu hóa tính linh hoạt và cá nhân hóa lộ trình học tập.

#### **- Đào tạo từ xa**

- Định nghĩa: Đào tạo từ xa là một chương trình học tập có cấu trúc hoàn chỉnh, trong đó người dạy và người học không cần phải xuất hiện cùng lúc và cùng một địa điểm trong suốt quá trình học tập [4].
- Quan hệ với E-Learning: Đào tạo từ xa hiện đại thường sử dụng E-Learning làm công cụ truyền tải và tương tác chính (dựa trên Internet, được gọi là Online Distance Learning). Tuy nhiên, về mặt khái niệm, Đào tạo từ xa là một phương thức tổ chức chương trình học (cung cấp bằng cấp, chứng chỉ) trong khi E-Learning là phương tiện kỹ thuật số được sử dụng để thực hiện phương thức đó.
- Mục tiêu: Cung cấp cơ hội tiếp cận giáo dục chất lượng cao cho những người bị hạn chế về địa lý, công việc hoặc thời gian, đồng thời mở rộng quy mô đào tạo.

## 1.2. Tổng quan về các hệ thống MOOC hiện nay

Sự phát triển mạnh mẽ của Công nghệ Thông tin (CNTT), đặc biệt là sự gia tăng đột biến của người dùng Internet trên toàn cầu [5], đã thúc đẩy một sự chuyển mình đáng kể trong ngành Giáo dục. Mô hình giảng dạy truyền thống được đổi mới thông qua việc chuyển đổi các bài giảng tĩnh thành tài liệu điện tử đa phương tiện có tính trực quan cao. CNTT và Internet đã mở rộng hình thức học tập từ tập trung sang phân tán, cho phép người học tiếp cận kiến thức thông qua các kênh giáo dục trực tuyến (ví dụ: Udacity, YouTube EDU, iTunes U) bằng thiết bị có kết nối mạng. Điều này không chỉ cung cấp nguồn học liệu mở phong phú mà còn đa dạng hóa lựa chọn trong việc truyền bá và tiếp nhận tri thức. Đặc biệt, việc sử dụng Internet làm cầu nối đã thúc đẩy sự phát triển của các giải pháp giáo dục-đào tạo trực tuyến, từ Đào tạo từ xa, E-learning đến mô hình phổ biến nhất hiện nay là Khóa học Trực tuyến Mở Rộng Quy Mô (MOOCs - Massive Open Online Courses). MOOCs hiện được xem là xu hướng chủ đạo của giáo dục số hiện đại, mang lại khả năng tiếp cận giáo dục chất lượng cao trên diện rộng [6].

Đào tạo từ xa (Distance Learning) đã trải qua một quá trình phát triển dài và sâu rộng kể từ khi xuất hiện vào thế kỷ XIX, khi hình thức chủ yếu chỉ là gửi tài liệu học tập qua đường bưu điện. Bước sang thế kỷ XX, sự hỗ trợ của các phương tiện truyền thông đại chúng như radio và truyền hình đã mở rộng phạm vi và khả năng tiếp cận của mô hình này. Đến cuối thế kỷ XX, sự ra đời và phổ cập của Internet tạo nên một bước ngoặt căn bản, chuyển đổi mô hình đào tạo từ xa truyền thống sang E-learning, nơi tài liệu và hoạt động học tập được phân phối chủ yếu thông qua các hệ thống mạng máy tính. Trong giai đoạn đầu, E-learning chủ yếu mang tính truyền tải một chiều. Tuy nhiên, sự phát triển của Web 2.0 đã thúc đẩy việc tích hợp các tính năng tương tác như diễn đàn, trò chuyện trực tuyến, cùng với sự ra đời của các Hệ thống Quản lý Học tập (LMS), từ đó tăng cường đáng kể mức độ tương tác giữa người học và nội dung học tập. Hệ sinh thái tài nguyên học tập cũng trở nên phong phú hơn, bao gồm các bài giảng dạng video, tài liệu tải về và nhiều khóa học mở được cung cấp thông qua các nền tảng nổi tiếng như MIT OpenCourseWare, Khan Academy, YouTube và các chương trình đào tạo cấp tín chỉ trực tuyến (ví dụ: Open Learning Initiative). Những thay đổi mang tính cách mạng này đã đặt nền móng cho sự hình thành và phát triển của MOOCs, một xu hướng giáo dục trực tuyến hiện đại có quy mô lớn và tính mở cao [6].

Khóa học Trực tuyến Mở Rộng Quy Mô (MOOCs - Massive Open Online Courses) là một mô hình giáo dục số được định nghĩa là các khóa học được thiết kế cho quy mô người tham dự lớn, có thể truy cập toàn diện và miễn phí cho bất kỳ ai có kết nối Internet, không yêu cầu chứng thực trình độ đầu vào [7]. Các chữ cái trong từ viết tắt MOOC phản ánh bốn đặc trưng cốt lõi của mô hình này:

- **M (Massive - Quy mô lớn):** Chỉ ra khả năng phục vụ một số lượng học viên vượt trội, thường lên đến hàng trăm nghìn người, cao hơn đáng kể so với các khóa học trực tuyến hoặc truyền thống.
- **O (Open - Tính Mở):** Đặc tính này thể hiện trên nhiều khía cạnh:
  - Mở về tiếp cận: Không giới hạn người tham gia, không yêu cầu bằng cấp hay điều kiện tiên quyết.
  - Mở về tài nguyên: Tài liệu học tập luôn sẵn có và hoàn toàn miễn phí.
  - Mở về bản quyền: Nội dung khóa học thường cho phép tái sử dụng hợp pháp cho mục đích giáo dục và giảng dạy.
- **O (Online - Trực tuyến):** Quá trình dạy và học được tiến hành hoàn toàn qua môi trường Internet, loại bỏ sự phụ thuộc vào vị trí vật lý hoặc thời gian cố định, chỉ yêu cầu kết nối mạng ổn định.
- **C (Course - Khóa học):** Đề cập đến một chương trình học tập có cấu trúc hoàn chỉnh, được thiết kế đặc biệt cho hình thức trực tuyến. Một khóa học MOOC bao gồm đầy đủ: tài nguyên học tập, cơ chế tương tác đa chiều (giảng viên – học viên và học viên – học viên), cùng với các hoạt động đánh giá (nhiệm vụ, bài tập, kiểm tra, phản hồi) nhằm đảm bảo mục tiêu học tập.

Mặc dù mô hình MOOC (Massive Open Online Courses) mang lại lợi ích rõ rệt về tính linh hoạt về thời gian và địa điểm, tạo điều kiện tiếp cận tri thức đại chúng, nó vẫn phải đổi mới với một số thách thức có hữu liên quan đến hiệu quả học tập và công nhận giá trị.

- **Thách thức về Duy trì động lực và tỷ lệ hoàn thành:** Tính tự chủ cao trong học tập đồng thời là một con dao hai lưỡi. Việc thiếu cấu trúc giám sát và môi trường tương tác trực tiếp theo kiểu truyền thống thường dẫn đến sự suy giảm động lực và sự tập trung của người học. Điều này được phản ánh rõ qua tỷ lệ bỏ học (attrition rate) cao, khiến số lượng học viên thực sự hoàn thành khóa học thấp hơn nhiều so với tổng số lượng đăng ký ban đầu [6].
- **Thách thức về công nhận giá trị:** Giá trị học thuật và chuyên môn của chứng chỉ MOOC vẫn chưa được công nhận rộng rãi tương đương với các khóa học chính quy truyền thống. Tuy nhiên, để cải thiện tình trạng này, các nhà cung cấp MOOC hàng đầu và các đối tác đại học đang nỗ lực chuyển dịch bằng cách phát triển các chương trình cấp chứng chỉ và bằng cấp chính thức, ví dụ như chương trình Professional Education và MicroMasters của edX, hay chương trình Thạc sĩ Khoa học Máy tính Trực tuyến OMS CS của Udacity [6].
- **Hạn chế trong Ứng dụng Công nghệ Hỗ trợ cá nhân hóa:** Một hạn chế đáng kể khác là sự thiếu vắng các giải pháp cá nhân hóa và hỗ trợ học tập quy mô lớn. Mặc dù Trí tuệ Nhân tạo đã phát triển mạnh mẽ và có tiềm năng to lớn trong việc cung cấp phản hồi tức thời, xây dựng lộ trình học tập thích ứng, và dự đoán nguy cơ bỏ

học, việc tích hợp AI để hỗ trợ người học trên phần lớn các hệ thống MOOC vẫn còn ở giai đoạn sơ khai. Hiện tại, chỉ một số ít các tổ chức tiên phong (như các nền tảng lớn liên kết với các trường đại học hàng đầu) mới bắt đầu chuyển dịch, đầu tư vào việc sử dụng AI để nâng cao trải nghiệm và tỷ lệ hoàn thành khóa học.

### 1.3. Xu hướng nổi bật của việc ứng dụng AI trong giáo dục

Trong kỷ nguyên đương đại, sự tích hợp của Trí tuệ Nhân tạo (AI) đã tạo ra một bước chuyển mình căn bản trong mô hình giáo dục toàn cầu, định hình lại đáng kể cả phương pháp giảng dạy và trải nghiệm học tập. Công nghệ AI không chỉ giới hạn ở việc tự động hóa các quy trình hành chính và giáo vụ cơ bản, mà còn mở rộng sang khả năng thiết kế tài liệu học tập đa phương tiện có tính tương tác cao. Đáng chú ý nhất, AI đóng vai trò thiết yếu trong việc thực hiện cá nhân hóa chương trình học một cách tinh vi, cho phép nội dung và tốc độ giảng dạy được điều chỉnh linh hoạt theo nhu cầu và tốc độ tiếp thu riêng biệt của từng người học. Sự can thiệp này thúc đẩy một môi trường học tập chủ động hơn, khơi gợi tính tự chủ trong học tập và làm tăng sự hứng thú của người học đối với quá trình lĩnh hội tri thức [3].

Ứng dụng Trí tuệ Nhân tạo trong lĩnh vực giáo dục được định nghĩa là việc khai thác các công nghệ tiên tiến như học máy, xử lý ngôn ngữ tự nhiên, nhận diện mẫu và phân tích dữ liệu lớn để tối ưu hóa quá trình giảng dạy, học tập, cùng với việc phát triển tài liệu và chương trình giáo dục. Các ứng dụng của AI ngày càng đa dạng, bao gồm việc phân tích dữ liệu chuyên sâu nhằm cung cấp nhận định chính xác và xây dựng lộ trình đào tạo cá nhân hóa; đồng thời tự động hóa các tác vụ hành chính cơ bản của giảng viên, và hỗ trợ thiết kế các tài liệu giảng dạy số hóa có tính tương tác cao. Thông qua những vai trò này, AI đã góp phần nâng cao chất lượng giáo dục và tạo ra một môi trường học tập linh hoạt và thích ứng với nhu cầu riêng biệt của từng người học, từ đó củng cố vị thế của AI như một công cụ hỗ trợ đắc lực trong hoạt động sư phạm [3].

Sự hội tụ giữa Trí tuệ Nhân tạo (AI) và các nền tảng học tập đã tạo ra sự chuyển dịch chiến lược trong ngành giáo dục, đặc biệt rõ rệt trong lĩnh vực đào tạo trực tuyến (e-learning) và việc tích hợp vào các hệ thống Khóa học Trực tuyến Đại chúng Mở (MOOCs) hiện tại và mới. Theo Global Market Insights Inc., quy mô thị trường AI trong Giáo dục đã đạt 4 tỷ USD vào năm 2022 và được dự báo sẽ tăng trưởng với Tốc độ Tăng trưởng Kép Hàng năm (CAGR) trên 10% trong giai đoạn 2023–2032, chủ yếu nhờ vào nhu cầu ngày càng tăng đối với xu hướng học tập cá nhân hóa [8]. Hơn nữa, 50% các cơ sở giáo dục trên toàn cầu đã ứng dụng AI để tối ưu hóa quy trình hành chính, góp phần nâng cao hiệu quả giáo dục cho người học [9]. Trên quy mô khu vực, Bắc Mỹ hiện đang thống trị thị trường toàn cầu với tỷ trọng doanh thu 36%, tiếp theo là Châu Âu và Châu Á – Thái Bình Dương, trong đó Mỹ và Ấn Độ là các quốc gia dẫn đầu về Công nghệ Giáo dục (EdTech) [9]. Đáng chú ý, Việt Nam đã được xếp vào top 10 quốc gia có tốc độ tăng trưởng EdTech nhanh nhất thế giới và là một trong top 3 quốc gia EdTech tiềm năng nhất khu vực Đông Nam Á (cùng với Singapore và Indonesia), đóng góp 20% tổng quy mô EdTech toàn khu vực, với sự hiện diện của nhiều đơn vị tiêu biểu như CodeGym, Azota, Teky, Clevai, và Prep [9]. Từ đó có thể thấy xu hướng ứng dụng AI nói riêng, Công nghệ

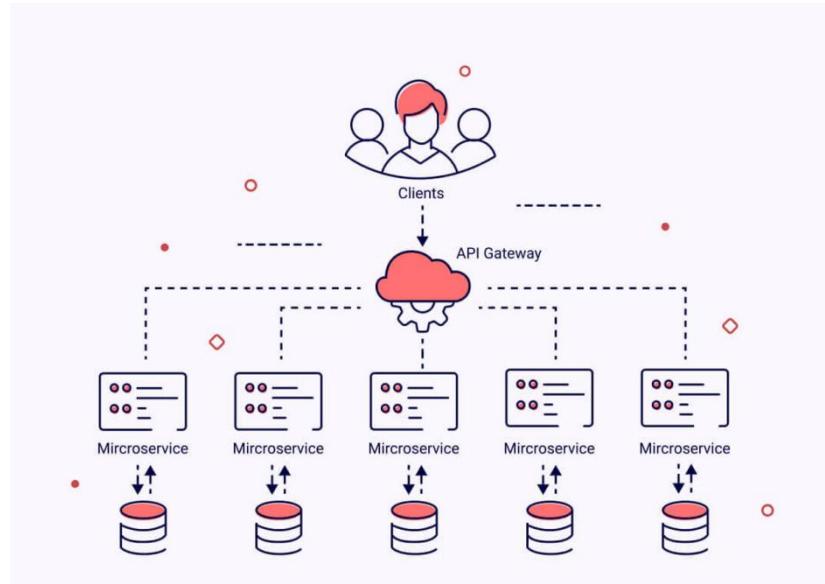
thông tin nói chung vào giáo dục đã là xu hướng và chắc chắn sẽ được triển khai ứng dụng mạnh mẽ hơn vào hiện tại cũng như tương lai.

Một trong những ứng dụng nổi bật của Trí tuệ Nhân tạo (AI) trong lĩnh vực giáo dục là sự triển khai của Chatbots AI. Nhờ vào sự tiến bộ của các thuật toán xử lý ngôn ngữ tự nhiên (NLP) và học máy (machine learning), Chatbots AI đã trở thành một trợ lý học tập 24/7 hiệu quả, cung cấp hỗ trợ tức thời, linh hoạt về thời gian và địa điểm cho người học, ngay cả khi không có sự can thiệp trực tiếp của giáo viên. Điều này có ý nghĩa quan trọng trong việc thúc đẩy tính tự chủ và hiệu quả của quá trình tự học [3]. Tuy nhiên, các hệ thống Chatbot truyền thống thường bộc lộ những hạn chế cố hữu về tính đơn điệu, thiếu sự linh hoạt trong tương tác, và đặc biệt là hạn chế về khả năng lập kế hoạch (planning) hoặc suy luận chuyên sâu (deep reasoning), dẫn đến việc thiếu khả năng cung cấp các phản hồi có chất lượng cao và độ chính xác tối ưu. Do đó, sự xuất hiện và phát triển mạnh mẽ của Tác nhân AI (AI Agent) đã được nhìn nhận là một giải pháp tiềm năng để khắc phục các giới hạn này. AI Agent được thiết kế với khả năng lập kế hoạch hành động rõ ràng, thực hiện suy luận đa tầng (multi-step reasoning) và tích hợp linh hoạt các công cụ bên ngoài, từ đó nâng cao đáng kể mức độ hỗ trợ và sự tinh vi trong trải nghiệm học tập của người dùng.

Thực trạng phân tích cho thấy một khoảng trống chiến lược trong các hệ thống giáo dục tích hợp AI và các nền tảng MOOC (Khóa học Trực tuyến Đại chúng Mô) hiện nay: Tính đa dụng của AI trong hệ thống không chỉ giới hạn ở chức năng Chatbots đơn thuần. Nhận thức rõ sự thiếu sót này đã tạo ra động lực nghiên cứu nhằm phát triển một Hệ thống Web Mô phỏng Lớp học Ảo ứng dụng AI Agent. Giải pháp được đề xuất này tận dụng sức mạnh của Mô hình Ngôn ngữ Lớn (LLM) tiên tiến, cụ thể là GPT-4o, kết hợp với công nghệ pgVector để lưu trữ và truy xuất vector nhúng hiệu quả. Mục tiêu chính là nhằm khắc phục triệt để các hạn chế về tương tác và khả năng suy luận của các hệ thống MOOC sử dụng AI hiện tại, từ đó cung cấp một môi trường học tập toàn diện và thông minh hơn.

## 2. Kiến trúc hệ thống vi dịch vụ

Kiến trúc Vi dịch vụ (Microservices Architecture) là một mô hình thiết kế phần mềm hiện đại, trong đó một ứng dụng được xây dựng như một tập hợp các dịch vụ nhỏ, độc lập, có khả năng triển khai độc lập. Mỗi dịch vụ chạy trong quy trình riêng của mình, quản lý một tập hợp chức năng nghiệp vụ cụ thể, và giao tiếp với các dịch vụ khác thông qua các giao thức nhẹ, thường là HTTP/REST hoặc hàng đợi tin nhắn (message queues).



Hình 1: Kiến trúc vi dịch vụ

Đặc trưng cốt lõi:

- Phân rã: Ứng dụng được phân tách thành các dịch vụ nhỏ xoay quanh các khả năng nghiệp vụ.
- Độc lập về Công nghệ: Mỗi dịch vụ có thể được phát triển bằng ngôn ngữ lập trình, framework, và cơ sở dữ liệu tối ưu nhất cho chức năng cụ thể đó.
- Triển khai Độc lập: Các dịch vụ có thể được xây dựng, thử nghiệm và triển khai một cách độc lập mà không cần phải triển khai lại toàn bộ ứng dụng.
- Khả năng Phục hồi: Sự thất bại của một dịch vụ sẽ ít có khả năng lan truyền và làm sập toàn bộ hệ thống.
- Quản trị Phi tập trung: Giảm thiểu sự phụ thuộc vào một tiêu chuẩn công nghệ duy nhất, cho phép các nhóm phát triển đưa ra quyết định cục bộ nhanh chóng hơn.

Lý do sử dụng:

- Tính linh hoạt và khả năng mở rộng: Vi dịch vụ cho phép mở rộng theo chiều ngang (horizontal scaling) chỉ các dịch vụ có nhu cầu tải cao, thay vì phải mở rộng toàn bộ ứng dụng.
- Tốc độ phát triển và triển khai: Với mô hình phân tách, các nhóm nhỏ, chuyên trách có thể phát triển, kiểm thử và triển khai từng dịch vụ độc lập và đồng thời.
- Phân bổ rủi ro và tính bền vững: Việc phân tách giúp cô lập sự cố. Nếu một dịch vụ gặp lỗi, các dịch vụ khác vẫn hoạt động bình thường, đảm bảo tính khả dụng cao của hệ thống.
- Tự do công nghệ: Các nhóm phát triển có thể lựa chọn công nghệ (ngôn ngữ lập trình, cơ sở dữ liệu) phù hợp nhất cho yêu cầu cụ thể của từng dịch vụ, thay vì bị ràng buộc bởi một bộ công nghệ duy nhất của kiến trúc đơn khối.
- Tổ chức đội ngũ: Kiến trúc Vi dịch vụ thường đi đôi với nguyên tắc "You Build It, You Run It" và phù hợp với Quy tắc của Conway, nơi các đội ngũ nhỏ, đa chức

năng sở hữu toàn bộ vòng đời của dịch vụ, thúc đẩy trách nhiệm và tốc độ ra quyết định.

### 3. Các công nghệ phát triển ứng dụng

#### 3.1. Spring Framework

Spring Framework là một khuôn khổ phát triển ứng dụng mã nguồn mở dẫn đầu trong hệ sinh thái Java. Nó cung cấp một hệ thống module toàn diện, được thiết kế để đơn giản hóa quá trình xây dựng các ứng dụng cấp doanh nghiệp có tính mô-đun hóa và khả năng mở rộng cao.

Đặc trưng cốt lõi:

- Hệ sinh thái toàn diện: Spring không chỉ là một framework mà là một tập hợp các dự án (như Spring Boot, Spring Security, Spring Data) giải quyết hầu hết các vấn đề kỹ thuật hiện đại.
- Sử dụng ngôn ngữ Java: Tận dụng sự mạnh mẽ, ổn định, và bảo mật vốn có của nền tảng Java/JVM, vốn là tiêu chuẩn cho các hệ thống doanh nghiệp cấp cao.
- Tính Hướng module: Khung công tác được xây dựng theo kiến trúc module, cho phép nhà phát triển chỉ chọn các thành phần cần thiết, giảm thiểu dung lượng ứng dụng và tăng hiệu suất.

Lý do sử dụng: Việc lựa chọn Spring Framework cho các dịch vụ cốt lõi như Xác thực và Người dùng là một quyết định kỹ thuật dựa trên các yêu cầu nghiêm ngặt về tính bảo mật, ổn định, tin cậy, và khả năng mở rộng không thể bị thỏa hiệp mà Java và Spring cung cấp.

#### 3.2. NestJS Framework

NestJS là một framework phát triển ứng dụng phía máy chủ hiệu suất cao, được xây dựng dựa trên Node.js và TypeScript. Nó được thiết kế với mục đích giúp các nhà phát triển xây dựng các ứng dụng phía máy chủ có khả năng mở rộng, dễ bảo trì và có kiến trúc rõ ràng bằng cách áp dụng các mô hình thiết kế nổi tiếng.

Đặc trưng cốt lõi:

- Kiến trúc hướng Module: NestJS áp dụng các mô hình kiến trúc đã được chứng minh trong ngành (như Angular), tổ chức mã nguồn thành các Modules, Controllers, và Providers (Services). Điều này tạo ra sự phân tách rõ ràng (separation of concerns).
- Sử dụng TypeScript: Tận dụng TypeScript để cung cấp kiểu dữ liệu tĩnh (static typing), giúp giảm thiểu lỗi runtime và nâng cao khả năng tái cấu trúc (refactoring) của mã nguồn, đặc biệt quan trọng cho các dự án phức tạp.

- Tuân thủ Thiết kế Hướng Đối tượng (OOP): Hỗ trợ đầy đủ các nguyên lý OOP (Kế thừa, Đóng gói, Đa hình), giúp xây dựng mã nguồn có tính tái sử dụng và dễ kiểm thử cao.
- Dependency Injection (DI): Áp dụng mô hình DI (tương tự Spring) để quản lý sự phụ thuộc giữa các thành phần, từ đó cải thiện tính mô-đun và khả năng kiểm thử đơn vị (unit testing).

Lý do sử dụng: NestJS cung cấp một khuôn khổ có cấu trúc cao và hiệu suất vượt trội nhờ vào Node.js và TypeScript. Điều này lý tưởng cho Course Service, nơi mà việc quản lý logic nghiệp vụ phức tạp cần sự rõ ràng về kiến trúc, tính bền vững trong bảo trì và khả năng xử lý đồng thời cao để phục vụ người học.

### 3.3. FastAPI Framework

FastAPI là một web framework hiện đại, hiệu suất cao dành cho việc xây dựng các API bằng Python, dựa trên các tiêu chuẩn mở như OpenAPI (trước đây là Swagger) và JSON Schema. Nó được thiết kế đặc biệt để đơn giản hóa quá trình tạo ra các API RESTful mạnh mẽ và nhanh chóng, lý tưởng cho việc triển khai các mô hình học máy và AI.

Đặc trưng cốt lõi:

- Hiệu suất vượt trội: FastAPI đạt hiệu suất ngang bằng với Node.js và Go, chủ yếu nhờ vào việc sử dụng ASGI (Asynchronous Server Gateway Interface) và chạy trên các máy chủ bất đồng bộ như Uvicorn. Đây là yếu tố then chốt cho các tác vụ tính toán nặng của AI.
- Hỗ trợ bất đồng bộ: Khả năng xử lý các tác vụ I/O và tính toán một cách bất đồng bộ thông qua cú pháp `async/await` của Python, giúp tối ưu hóa việc sử dụng tài nguyên và duy trì độ trễ thấp (low latency) khi xử lý nhiều yêu cầu đồng thời.
- Kiểm tra kiểu dữ liệu tự động: Sử dụng thư viện Pydantic để xác định, kiểm tra và tuân tự hóa dữ liệu đầu vào và đầu ra. Điều này đảm bảo tính toàn vẹn dữ liệu và cung cấp các thông báo lỗi rõ ràng.
- Tài liệu API Tự động: Tự động tạo tài liệu API tương tác theo chuẩn OpenAPI, bao gồm giao diện người dùng Swagger UI và Redoc. Điều này hỗ trợ mạnh mẽ cho việc tích hợp và khám phá dịch vụ.

Lý do sử dụng: FastAPI là lựa chọn tối ưu về mặt kỹ thuật cho AI Service vì nó kết hợp hiệu suất vượt trội cần thiết cho các tác vụ tính toán nặng với sự tương thích hoàn hảo với hệ sinh thái AI/ML của Python. Điều này cho phép dịch vụ này xử lý các yêu cầu mô hình phức tạp một cách nhanh chóng, ổn định và đáng tin cậy.

### 3.4. Kong Gateway

Kong Gateway là một nền tảng API Gateway mã nguồn mở, nhẹ, có khả năng mở rộng cao, được thiết kế để quản lý, định tuyến và bảo mật lưu lượng truy cập API cho các kiến trúc microservices và hybrid. Nó hoạt động như một điểm nhập duy nhất (Single Entry Point) cho tất cả các dịch vụ backend.

Đặc trưng cốt lõi:

- Kiến trúc Lớp Proxy: Kong được triển khai giữa máy khách (client) và các dịch vụ backend (vì dịch vụ). Nó chịu trách nhiệm xử lý các yêu cầu ban đầu, chuyển tiếp chúng đến các dịch vụ đích, và trả về phản hồi một cách minh bạch.
- Khả năng Mở rộng Cao: Được xây dựng trên nền tảng Nginx và chạy trên Luajit/OpenResty (sử dụng ngôn ngữ Lua), Kong đạt được hiệu suất I/O cao và khả năng xử lý lượng lớn lưu lượng truy cập đồng thời.
- Hệ thống Plugin Module: Tính năng cốt lõi của Kong được mở rộng thông qua một hệ thống plugin mạnh mẽ. Các plugin cho phép thực hiện các chính sách xuyên suốt (cross-cutting concerns) như bảo mật, giám sát, và giới hạn tốc độ mà không cần sửa đổi mã dịch vụ backend.
- Khả năng Quản lý API: Cung cấp các công cụ để quản lý các dịch vụ (Service), các tuyến đường (Route), Người tiêu dùng (Consumer), và các Chứng chỉ SSL/TLS.

Lý do sử dụng: Kong Gateway đóng vai trò là cổng kiểm soát chiến lược, cho phép hệ thống vi dịch vụ của bạn đạt được tính bảo mật, khả năng quản lý, và hiệu suất cần thiết. Nó trừu tượng hóa sự phức tạp của kiến trúc phân tán khỏi máy khách và cho phép các nhóm phát triển thực hiện các chính sách xuyên suốt một cách nhất quán và tập trung.

### 3.5. NextJS Framework

Next.js là một framework phát triển ứng dụng web mã nguồn mở được xây dựng trên React, cung cấp một giải pháp toàn diện để xây dựng các giao diện người dùng có khả năng mở rộng, hiệu suất cao và được tối ưu hóa cho công cụ tìm kiếm (SEO). Nó là một "Meta-Framework" giúp tiêu chuẩn hóa và tăng cường khả năng của React.

Đặc trưng cốt lõi:

- Render đa dạng: Hỗ trợ nhiều chiến lược render khác nhau: Server-Side Rendering (SSR), Static Site Generation (SSG), và Incremental Static Regeneration (ISR). Điều này cho phép tối ưu hóa hiệu suất và SEO cho từng trang cụ thể.
- Tối ưu hóa Hiệu suất: Tích hợp sẵn các tính năng tối ưu như chia nhỏ mã nguồn tự động (Automatic Code Splitting), tải trước (pre-fetching) thông minh, và tối ưu hóa hình ảnh. Điều này giúp cải thiện đáng kể Core Web Vitals và trải nghiệm người dùng (UX).
- Phát triển Tốc độ cao: Cung cấp cấu hình mặc định (Zero-config) và tính năng làm mới nhanh chóng (Fast Refresh), giúp tăng tốc độ phát triển và cải thiện sự hài lòng của lập trình viên.
- Routing Dựa trên File: Cấu trúc định tuyến (routing) được quản lý tự động thông qua cấu trúc thư mục/tệp, đơn giản hóa việc quản lý đường dẫn và trạng thái ứng dụng.

Lý do sử dụng: Next.js là lựa chọn chiến lược và kỹ thuật vượt trội cho Giao diện Người dùng, bởi vì nó cung cấp một phương tiện để xây dựng các ứng dụng web nhanh, SEO-

friendly, và có khả năng mở rộng dựa trên React. Điều này đảm bảo người dùng có được trải nghiệm truy cập tốt nhất khi tương tác với dữ liệu được xử lý bởi các dịch vụ backend phức tạp của hệ thống.

#### 4. Hệ quản trị cơ sở dữ liệu PostgreSQL

PostgreSQL là một Hệ quản trị Cơ sở dữ liệu Quan hệ Đối tượng (Object-Relational Database Management System - ORDBMS) mã nguồn mở, nổi tiếng với sự mạnh mẽ, khả năng mở rộng, tuân thủ tiêu chuẩn SQL và các tính năng tiên tiến ở cấp độ doanh nghiệp. Nó được đánh giá cao về tính toàn vẹn dữ liệu và độ tin cậy.

Đặc trưng cốt lõi:

- Tuân thủ ACID: Đảm bảo các thuộc tính Atomicity, Consistency, Isolation, và Durability (ACID) nghiêm ngặt, tối quan trọng cho các ứng dụng yêu cầu tính toàn vẹn giao dịch cao.
- Khả năng Mở rộng: Hỗ trợ hiệu quả cả mở rộng theo chiều dọc và các kỹ thuật mở rộng theo chiều ngang thông qua các tính năng như Replication và Partitioning.
- Hỗ trợ SQL và Tính năng Đối tượng: Ngoài việc tuân thủ các tiêu chuẩn SQL, PostgreSQL còn tích hợp các tính năng hướng đối tượng (ví dụ: kế thừa bảng, các kiểu dữ liệu phức tạp), giúp xử lý các mối quan hệ dữ liệu phức tạp hơn.
- Kiểu dữ liệu Đa dạng: Hỗ trợ nhiều kiểu dữ liệu nâng cao ngoài các kiểu quan hệ truyền thống, bao gồm cả các kiểu dữ liệu phi cấu trúc như JSON/JSONB và XML, cũng như kiểu dữ liệu Geospatial (PostGIS).

Lý do sử dụng: PostgreSQL cung cấp một nền tảng cơ sở dữ liệu mạnh mẽ, đáng tin cậy, và linh hoạt, đóng vai trò là xương sống dữ liệu cho kiến trúc vi dịch vụ của bạn. Khả năng hỗ trợ ACID nghiêm ngặt và các tính năng tiên tiến như JSONB và pgVector làm cho nó trở thành lựa chọn lý tưởng để đáp ứng cả yêu cầu về tính toàn vẹn của các giao dịch cốt lõi lẫn nhu cầu xử lý dữ liệu vector phức tạp của dịch vụ AI.

#### 5. Message Broker RabbitMQ

RabbitMQ là một Message Broker (Bộ môi giới tin nhắn) mã nguồn mở, được triển khai dựa trên giao thức Advanced Message Queuing Protocol (AMQP). Nó hoạt động như một nền tảng trung gian, cho phép các ứng dụng (microservices) gửi và nhận thông điệp một cách bất đồng bộ (asynchronously), đảm bảo rằng các dịch vụ được tách biệt (decouple) về mặt thời gian và không gian.

Đặc trưng cốt lõi:

- Đảm bảo Gửi/Nhận (Reliability): Hỗ trợ các cơ chế xác nhận (acknowledgement) và giữ lại tin nhắn (message persistence), đảm bảo không có thông điệp quan trọng nào bị mất.
- Tách biệt (Decoupling): Cho phép các microservice như AI Service và Course Service được phát triển và triển khai độc lập.

- Hỗ trợ Định tuyến linh hoạt: Cung cấp nhiều loại Exchange cho phép định tuyến thông điệp dựa trên các tiêu chí phức tạp.
- Khả năng Mở rộng (Scalability): Dễ dàng thêm các Consumer mới để xử lý tin nhắn, từ đó phân tán tải và tăng khả năng xử lý.

Lý do sử dụng: RabbitMQ được chọn để làm Message Broker, tạo nên xương sống giao tiếp bất đồng bộ và tin cậy cho kiến trúc microservice. Nó đảm bảo sự tách biệt (decoupling) hoàn toàn giữa AI Service và Course Service. Cụ thể, RabbitMQ giúp Course Service đẩy các dữ liệu quan trọng như transcript bài giảng, kết quả quiz và lịch sử chat dưới dạng sự kiện (events) tới AI Service một cách hiệu quả, tự động kích hoạt các pipeline xử lý dữ liệu (Data Processing và Retrieval), đồng thời duy trì tính nhất quán và khả năng mở rộng của toàn hệ thống.

## 6. Kết luận.

Chương 1 đã hoàn thành việc phân tích tổng quan về bối cảnh chuyển đổi số trong giáo dục (EdTech) và vai trò chiến lược của Trí tuệ Nhân tạo (Generative AI), chỉ rõ xu hướng tất yếu là Cá nhân hóa trải nghiệm học tập. Tuy nhiên, chương này cũng chỉ ra hạn chế có hữu của các hệ thống Lớp học ảo và MOOC truyền thống là thiếu sự tương tác cá nhân hóa và đánh giá tức thời, dẫn đến tỷ lệ bỏ học cao. Để giải quyết vấn đề này, đề tài đã đề xuất xây dựng Hệ thống Web mô phỏng lớp học ảo tích hợp AI Agent. Kiến trúc hệ thống được thiết kế dựa trên Kiến trúc Microservices để đảm bảo khả năng mở rộng và lõi thông minh được xây dựng bằng kiến trúc Agentic RAG điều phối bởi Lang Graph, sử dụng mô hình tiên tiến như GPT-4o và cơ sở dữ liệu vector pgVector. Sự kết hợp công nghệ này cho phép Agent có khả năng suy luận, cung cấp phản hồi sâu sắc và đưa ra đánh giá nồng lực chính xác, từ đó tạo ra một AI Companion thực thụ, đáp ứng hiệu quả nhu cầu giáo dục trong kỷ nguyên số.

# CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THÔNG WEB MÔ PHỎNG LỚP HỌC ẢO TÍCH HỢP AI AGENT

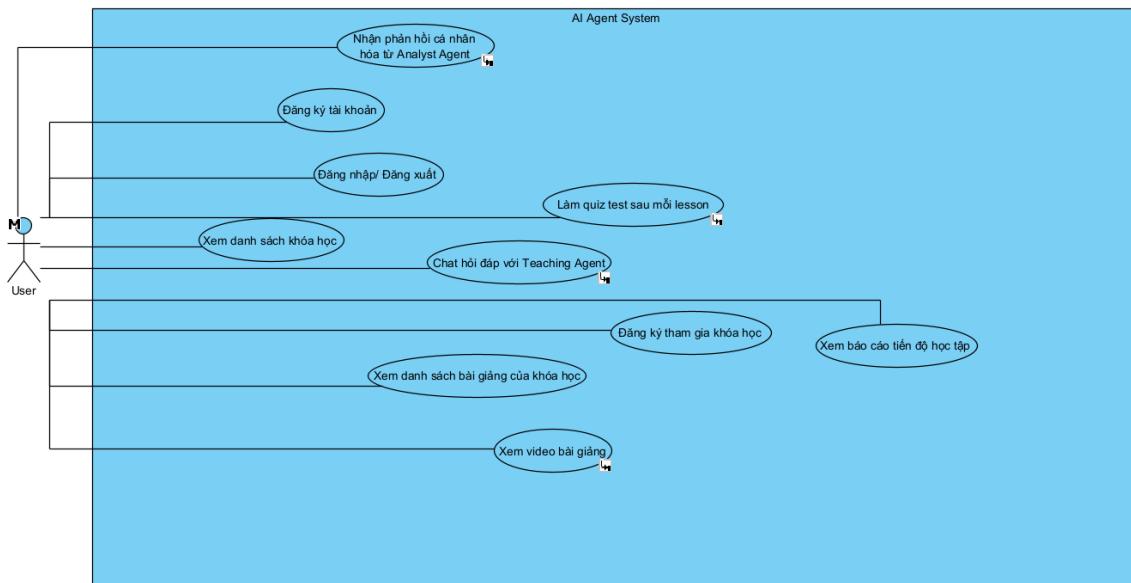
## 1. Pha phân tích

STT	Tác nhân	Hoạt động chính
1	Người dùng (User)	<ul style="list-style-type: none"> <li>- Đăng ký tài khoản và đăng nhập để sử dụng các chức năng của hệ thống.</li> <li>- Cập nhật thông tin cá nhân</li> <li>- Xem lại thông tin cá nhân</li> <li>- Chọn khóa học và đăng ký khóa học</li> <li>- Chọn bài học và xem video bài giảng</li> <li>- Trò chuyện với AI Agent về bài học.</li> <li>- Làm bài quiz của AI</li> <li>- Yêu cầu AI vẽ MindMap</li> </ul>

Bảng 1 : Tác nhân và chức năng của hệ thống

### 1.1. Use Case Diagram

#### a. Usecase tổng quát



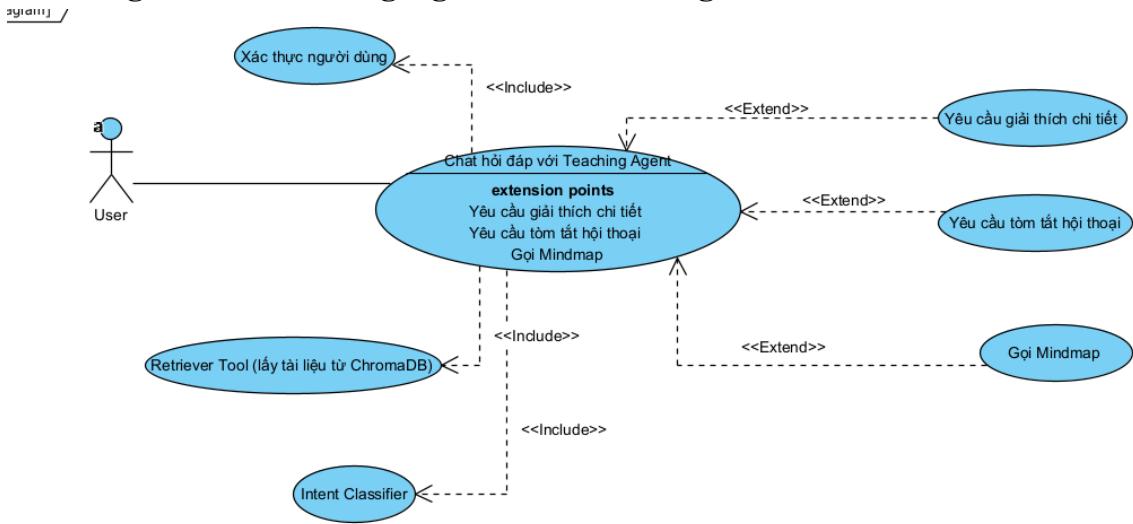
Hình 2: Usecase tổng quát

Mô tả use case:

Use case	Mô tả
Đăng ký	Cho phép người dùng tạo tài khoản mới trong hệ thống học tập trực tuyến.
Đăng nhập/ Đăng xuất	Cho phép học viên truy cập vào hệ thống bằng tài khoản đã có và đăng xuất an toàn.
Xem danh sách khóa học	Cho phép học viên xem danh sách các khóa học có sẵn, bao gồm thông tin mô tả và mức độ khó.
Xem báo cáo tiến độ học tập	Cho phép học viên xem báo cáo chi tiết về tiến độ học tập, bao gồm phần trăm hoàn thành, điểm số và khuyến nghị cải thiện.
Xem danh sách bài giảng của khóa học	Cho phép học viên xem danh sách các bài giảng trong khóa học đã đăng ký, với thông tin thời lượng và nội dung tóm tắt.
Tương tác với Teaching Agent	Cho phép học viên tương tác thời gian thực với AI Teaching Agent để hỏi đáp, giải thích khái niệm và hỗ trợ học tập.
Xem video bài giảng	Cho phép học viên xem video bài giảng với tính năng tốc độ điều chỉnh phù hợp.
Làm quiz test sau mỗi lesson	Cho phép học viên thực hiện bài kiểm tra quiz sau mỗi bài học, nhận điểm số ngay lập tức.

Bảng 2 : Mô tả các use case hệ thống

## b. Tương tác với Teaching Agent Use Case Diagram



Hình 3: Biểu đồ use case chức năng hỏi đáp Teaching Agent

- Kịch bản chức năng:

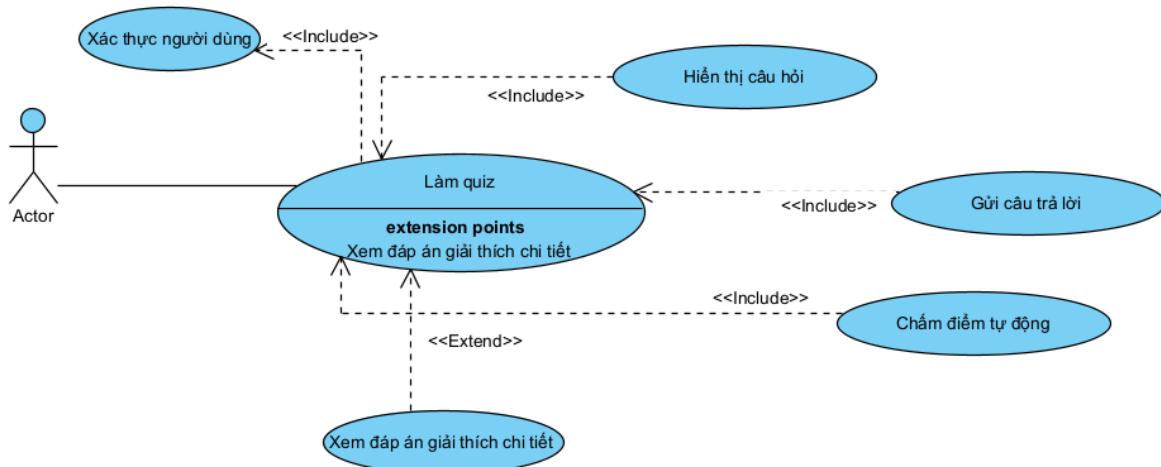
Tên kịch bản	Tương tác với Agent
Tác nhân	Người dùng (User)
Tên điều kiện	<ul style="list-style-type: none"> <li>- Người dùng đã đăng nhập thành công vào hệ thống.</li> <li>- Hệ thống đang ở chế độ tương tác với Teaching Agent (giao diện chat sẵn sàng).</li> <li>- Cơ sở dữ liệu pgVector đã được tải dữ liệu kiến thức giảng dạy.</li> </ul>
Kịch bản chính	<ol style="list-style-type: none"> <li>1. Người dùng gửi câu hỏi hoặc yêu cầu học tập (ví dụ: "Số có 5 chữ số là gì?") thông qua giao diện Chatbot. Hệ thống tiếp nhận và khởi tạo phiên xử lý mới.</li> <li>2. Hệ thống kích hoạt module Intent Classifier để phân tích ngữ nghĩa và xác định loại yêu cầu (ví dụ: giải thích khái niệm, giải bài tập, hay vẽ sơ đồ).</li> <li>3. Dựa trên kết quả phân loại, module Retriever thực hiện truy vấn vector (Vector Search) trên cơ sở tri thức (Knowledge Base) để trích xuất các phân đoạn kiến thức (Context Chunks) có độ tương đồng cao nhất.</li> <li>4. Teaching Agent tổng hợp ngữ cảnh vừa tìm được cùng với lịch sử hội thoại, sử dụng Mô hình Ngôn ngữ Lớn (LLM) để sinh câu trả lời sự phạm phù hợp với trình độ học sinh.</li> </ol>

	<p>5. Hệ thống hiển thị câu trả lời hoàn chỉnh trên giao diện người dùng.</p> <p>6. Nếu câu trả lời chứa các điểm mở rộng (Extension Points) như yêu cầu xem chi tiết hoặc tạo trực quan hóa, hệ thống tự động đề xuất các hành động tiếp theo (ví dụ: nút "Vẽ sơ đồ tư duy").</p> <p>7. Nếu người dùng chọn xem mindmap. Hệ thống truy vấn dữ liệu bài học được lưu trữ trong cơ sở dữ liệu và thực hiện tạo một sơ đồ tư duy tổng hợp nội dung chính của bài học một cách có cấu trúc và hiển thị lên cho người dùng xem.</p> <p>8. Người dùng tiếp tục tương tác. Hệ thống đồng bộ toàn bộ ngữ cảnh vào Bộ nhớ Bền vững (Persistent Memory) để phục vụ các lượt truy vấn sau.</p> <p>9. Khi người dùng yêu cầu đánh giá hoặc hệ thống phát hiện kết thúc phiên học (hoặc đạt tiến độ xem video &gt;90%).</p> <p>10. Xử lý: Hệ thống kích hoạt module Session Analyzer. Module này tổng hợp toàn bộ lịch sử hội thoại (Chat History) và kết quả làm bài kiểm tra cuối bài học của người dùng (Quiz Attempt) để đánh giá năng lực người dùng.</p> <p>11. Hệ thống trả về bảng đánh giá năng lực (Proficiency Report), bao gồm: Kiến thức đã đạt, Điểm cần cải thiện, và Lộ trình đề xuất tiếp theo (Next-step Recommendation).</p>
Kịch bản ngoại lệ	<p>2.1. Hệ thống nhận diện câu hỏi mơ hồ, thiếu ngữ cảnh hoặc đa nghĩa (Ambiguous Query).</p> <p>2.1.1. Hệ thống tạm dừng quy trình xử lý chính, kích hoạt cơ chế đàm thoại làm rõ (Clarification Dialog). Agent phản hồi yêu cầu người dùng cung cấp thêm thông tin cụ thể trước khi quay lại Bước 1.</p> <p>3.1. Module Retriever trả về kết quả rỗng hoặc độ tương đồng thấp dưới ngưỡng cho phép (Low Confidence Score), tức không tìm thấy tri thức phù hợp trong CSDL.</p> <p>3.1.1. Hệ thống thông báo trạng thái "Knowledge Gap" (ví dụ: "Không tìm thấy thông tin phù hợp trong bài học này") và gợi ý các từ khóa hoặc chủ đề liên quan để người dùng thử lại.</p> <p>4.1. Xảy ra lỗi trong quá trình suy luận của LLM (LLM Inference Error) hoặc lỗi Time-out.</p> <p>4.1.1. Hệ thống bắt lỗi (Exception Catching), hiển thị thông</p>

	<p>báo sự cố kỹ thuật thân thiện và đề xuất người dùng thực hiện lại yêu cầu sau ít phút.</p> <p>7.1. Người dùng kích hoạt tính năng vẽ mindmap nhưng dữ liệu đầu vào (Context) quá ít hoặc không có cấu trúc phân cấp rõ ràng để tạo sơ đồ.</p> <p>7.1.1. Hệ thống từ chối yêu cầu tạo hình ảnh, thông báo lý do cụ thể (Insufficient Data) và hướng dẫn người dùng đặt câu hỏi chi tiết hơn để nạp đủ ngữ cảnh cần thiết.</p>
--	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Bảng 3 : Kịch bản chức năng hỏi đáp Teaching Agent

### c. Làm quiz Use Case Diagram



Hình 4: Biểu đồ use case chức năng làm quiz

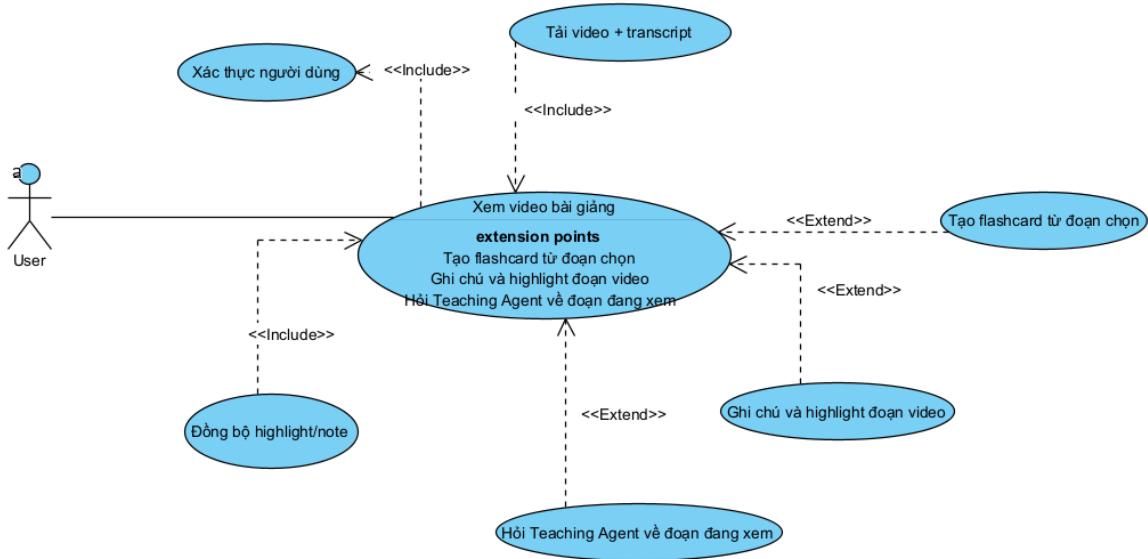
- Kịch bản chức năng:

Tên kịch bản	Làm Quiz
Tác nhân	Người dùng (User)
Tên điều kiện	<ul style="list-style-type: none"> <li>- Người dùng đã đăng nhập thành công vào hệ thống.</li> <li>- Hệ thống đang ở chế độ luyện tập (Practice mode), với Practice Agent sẵn sàng.</li> <li>- Cơ sở dữ liệu bài quiz đã được tải, bao gồm các bài kiểm tra liên quan đến kiến thức giảng dạy.</li> </ul>
Kịch bản chính	<ol style="list-style-type: none"> <li>1. Người dùng chọn chức năng "Làm Quiz" từ giao</li> </ol>

	diện chính. 2. Hệ thống kích hoạt Xác thực người dùng để xác minh quyền truy cập. 3. Hệ thống hiển thị danh sách bài quiz và gợi ý bài phù hợp dựa trên lịch sử học tập . 4. Người dùng chọn bài quiz và hệ thống hiển thị giao diện làm quiz . 5. Người dùng trả lời các câu hỏi trong bài quiz. 6. Người dùng hoàn thành và click "Gửi bài". 7. Hệ thống xử lý bài nộp và chấm điểm tự động . 8. Hệ thống hiển thị kết quả quiz, bao gồm điểm số và phản hồi. 9. Hệ thống lưu kết quả vào hồ sơ người dùng và quay về giao diện chính.
Kịch bản ngoại lệ	2.1. Xác thực thất bại (ví dụ: quyền không đủ). 2.1.1. Hệ thống thông báo lỗi "Không có quyền truy cập" và quay về trang chủ. 3.1. Không có bài quiz phù hợp (danh sách rỗng). 3.1.1. Hệ thống gợi ý chuyển sang chế độ học khác hoặc tạo bài quiz mới. 6.1. Người dùng gửi bài chưa hoàn thành (thiếu câu trả lời). 6.1.1. Hệ thống cảnh báo "Vui lòng hoàn thành tất cả câu hỏi" và quay lại bước 5. 7.1. Lỗi chấm điểm (ví dụ: ván đè kết nối cơ sở dữ liệu). 7.1.1. Hệ thống thông báo "Lỗi xử lý, vui lòng thử lại" và lưu bài tạm thời để nộp sau.

Bảng 4 : Kịch bản chức năng làm quiz

#### d. Xem video bài giảng Use Case Diagram



Hình 5: Biểu đồ use case chức năng xem video bài giảng

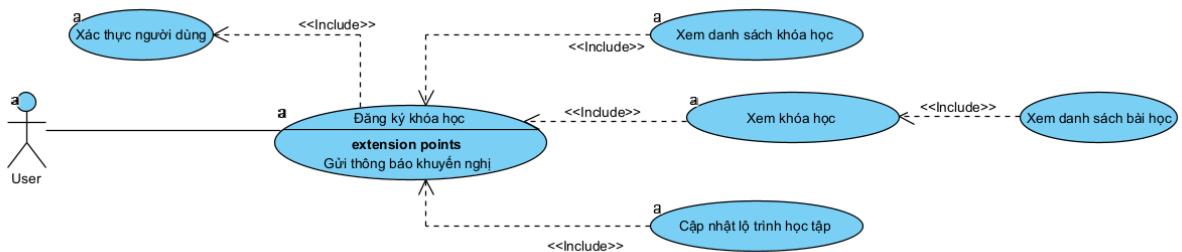
- Kịch bản chức năng:

Tên kịch bản	Xem video bài giảng
Tác nhân	Người dùng (User)
Tên điều kiện	<ul style="list-style-type: none"> <li>- Người dùng đã đăng nhập thành công vào hệ thống.</li> <li>- Hệ thống đang ở chế độ học tập video (Video Learning mode), với Teaching Agent sẵn sàng hỗ trợ.</li> <li>- Thư viện video bài giảng đã được tải, bao gồm metadata như transcript và thời lượng.</li> </ul>
Kịch bản chính	<ol style="list-style-type: none"> <li>1. Người dùng chọn video bài giảng từ danh sách gợi ý hoặc tìm kiếm (extend: Đọc transcript nếu cần preview).</li> <li>2. Hệ thống kích hoạt Xác thực người dùng để xác minh quyền truy cập nội dung.</li> <li>3. Hệ thống tải video và transcript đồng bộ.</li> <li>4. Hệ thống hiển thị giao diện xem video với các công cụ hỗ trợ (player, timeline, note panel).</li> <li>5. Người dùng phát video và theo dõi nội dung.</li> <li>6. Hệ thống tự động đồng bộ highlight hoặc note nếu có từ phiên trước (extend: Highlight/note).</li> <li>7. Nếu người dùng tạm dừng tại đoạn cụ thể, hệ thống kích hoạt extension points.</li> </ol>

	<p>8. Người dùng có thể hỏi Teaching Agent về nội dung đang xem.</p> <p>9. Video kết thúc, hệ thống hiển thị tóm tắt và gợi ý quiz liên quan, lưu tiến độ xem vào hồ sơ.</p>
Kịch bản ngoại lệ	<p>2.1. Xác thực thất bại (ví dụ: quyền truy cập nội dung premium).</p> <p>2.1.1. Hệ thống thông báo "Nội dung yêu cầu nâng cấp tài khoản" và quay về danh sách video.</p> <p>3.1. Lỗi tải video/transcript (kết nối chậm hoặc file hỏng).</p> <p>3.1.1. Hệ thống hiển thị thông báo "Đang tải lại, vui lòng chờ" và thử tải sau 10 giây, nếu thất bại thì gợi ý video thay thế.</p> <p>7.1. Người dùng tạm dừng nhưng không kích hoạt extension (ví dụ: thoát đột ngột).</p> <p>7.1.1. Hệ thống lưu vị trí tạm dừng và gửi nhắc nhở tiếp tục qua thông báo.</p> <p>9.1. Lỗi lưu tiến độ (vấn đề cơ sở dữ liệu).</p> <p>9.1.1. Hệ thống thông báo "Tiến độ sẽ được đồng bộ sau" và lưu cục bộ tạm thời.</p>

Bảng 5 : Kịch bản chức năng xem video bài giảng

## e. Đăng ký khóa học



Hình 6: Biểu đồ use case chức năng đăng ký khóa học

- Kịch bản chức năng:

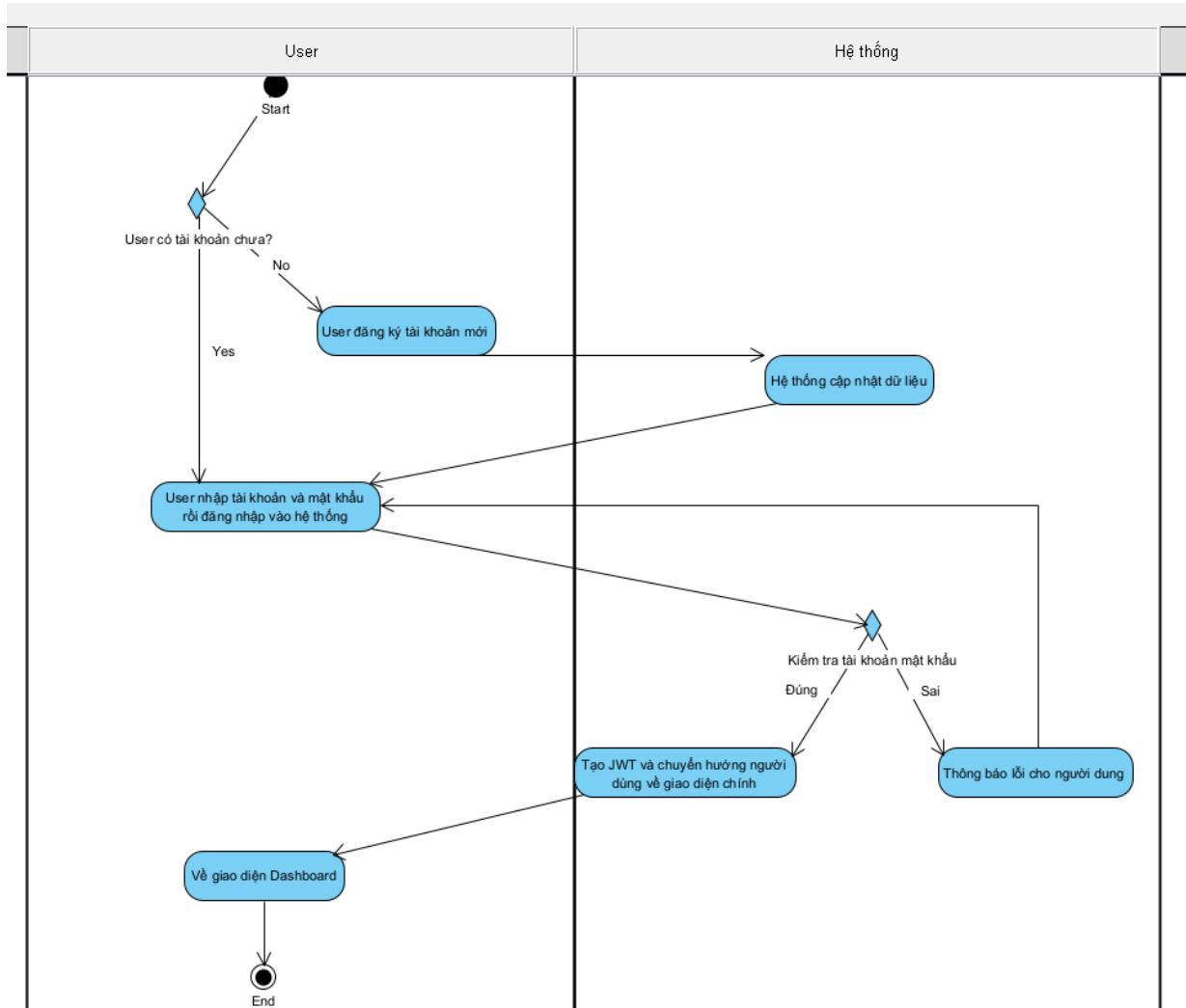
Tên kịch bản	Đăng ký khóa học
Tác nhân	Người dùng (User)
Tên điều kiện	<ul style="list-style-type: none"> <li>- Người dùng đã truy cập giao diện chính của hệ thống.</li> <li>- Hệ thống đang ở chế độ đăng ký khóa học, với danh sách khóa học sẵn sàng tải.</li> <li>- Người dùng có tài khoản hợp lệ (chưa đăng nhập thì sẽ redirect đến đăng nhập).</li> </ul>
Kịch bản chính	<ol style="list-style-type: none"> <li>1. Người dùng chọn chức năng "Đăng ký khóa học" từ giao diện chính.</li> <li>2. Hệ thống kích hoạt Xác thực người dùng để xác minh quyền truy cập.</li> <li>3. Hệ thống hiển thị danh sách khóa học có sẵn.</li> <li>4. Người dùng chọn khóa học mong muốn từ danh sách.</li> <li>5. Hệ thống kiểm tra điều kiện khóa học (ví dụ: còn slot, mở đăng ký) và hiển thị chi tiết bài học.</li> <li>6. Người dùng xác nhận đăng ký bằng cách click "Đăng ký ngay".</li> <li>7. Hệ thống xử lý đăng ký (lưu enrollment, cập nhật slot khóa học).</li> <li>8. Hệ thống cập nhật tiến độ học tập ban đầu cho người dùng.</li> <li>9. Hệ thống hiển thị thông báo "Đăng ký thành công" và redirect đến giao diện khóa học đã đăng ký.</li> </ol>
Kịch bản ngoại lệ	<ol style="list-style-type: none"> <li>2.1. Xác thực thất bại (ví dụ: chưa đăng nhập hoặc token hết hạn).</li> </ol>

	<p>hạn).</p> <p>2.1.1. Hệ thống redirect đến trang đăng nhập và thông báo "Vui lòng đăng nhập để tiếp tục".</p> <p>4.1. Không có khóa học nào phù hợp (danh sách rỗng).</p> <p>4.1.1. Hệ thống hiển thị thông báo "Hiện chưa có khóa học, vui lòng quay lại sau" và gợi ý tìm kiếm nâng cao.</p> <p>6.1. Khóa học không đủ điều kiện (đầy slot hoặc đóng đăng ký).</p> <p>6.1.1. Hệ thống thông báo lỗi cụ thể (ví dụ: "Khóa học đã đầy, vui lòng chọn khóa khác") và quay lại bước 3.</p> <p>7.1. Lỗi xử lý đăng ký (vấn đề kết nối DB).</p> <p>7.1.1. Hệ thống rollback thay đổi, thông báo "Lỗi hệ thống, vui lòng thử lại sau" và lưu log để admin kiểm tra.</p>
--	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Bảng 6 : Kích bản chúc năng đăng ký khóa học

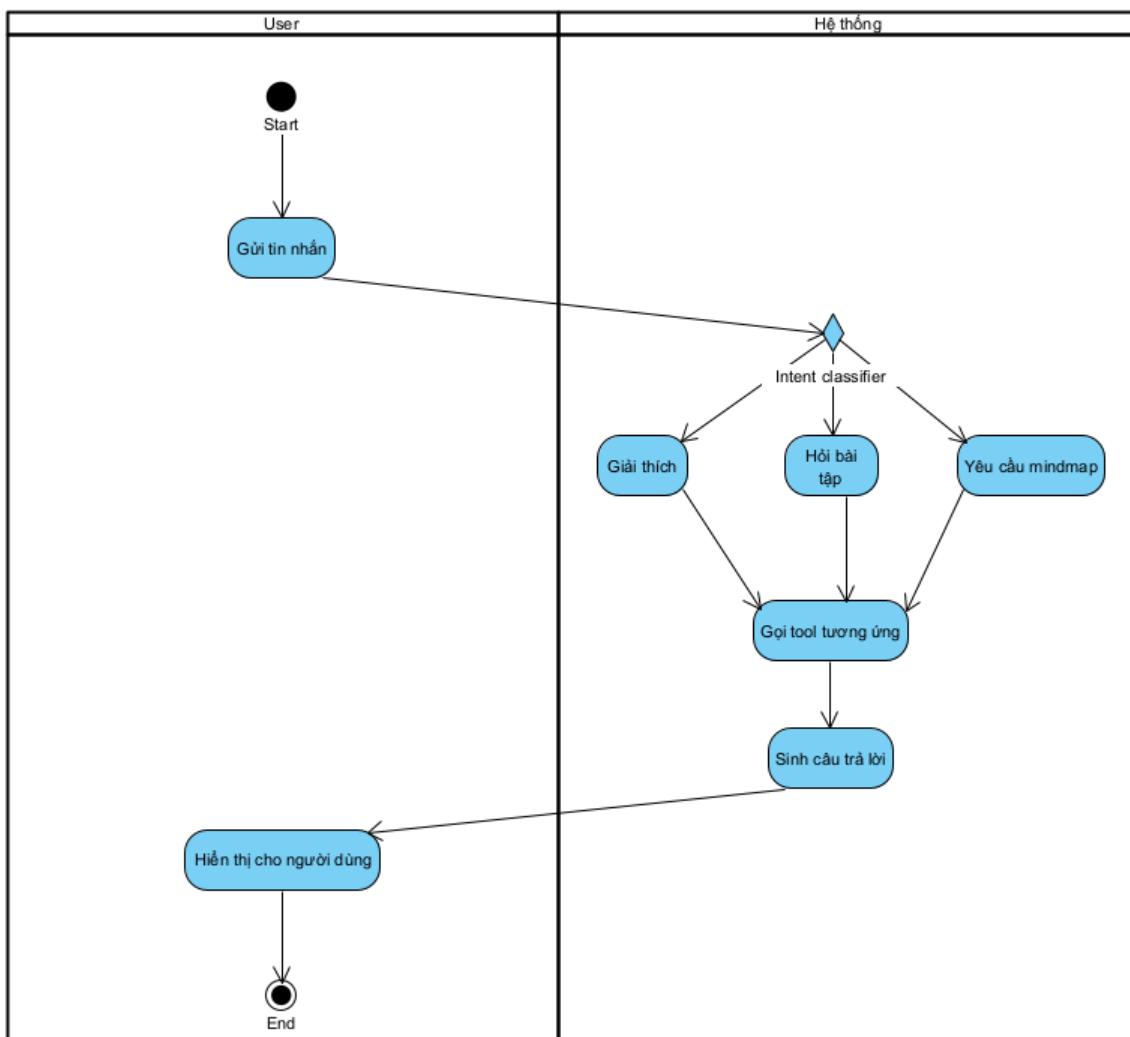
## 1.2. Activity Diagram

### a. Đăng nhập



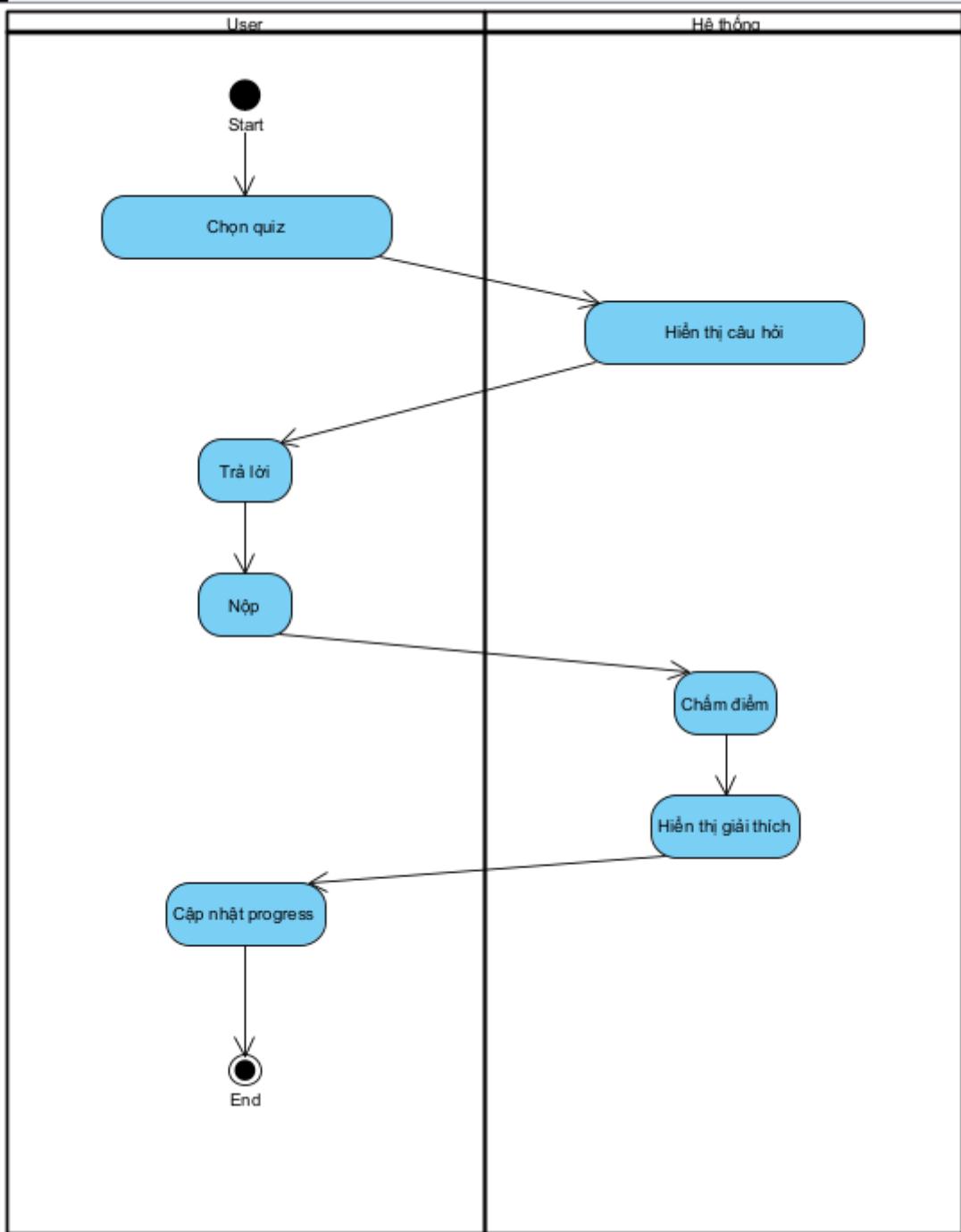
Hình 7: Biểu đồ hoạt động “Đăng nhập”

## b.Tương tác với Teaching Agent



Hình 8: Biểu đồ hoạt động “Tương tác với Agent”

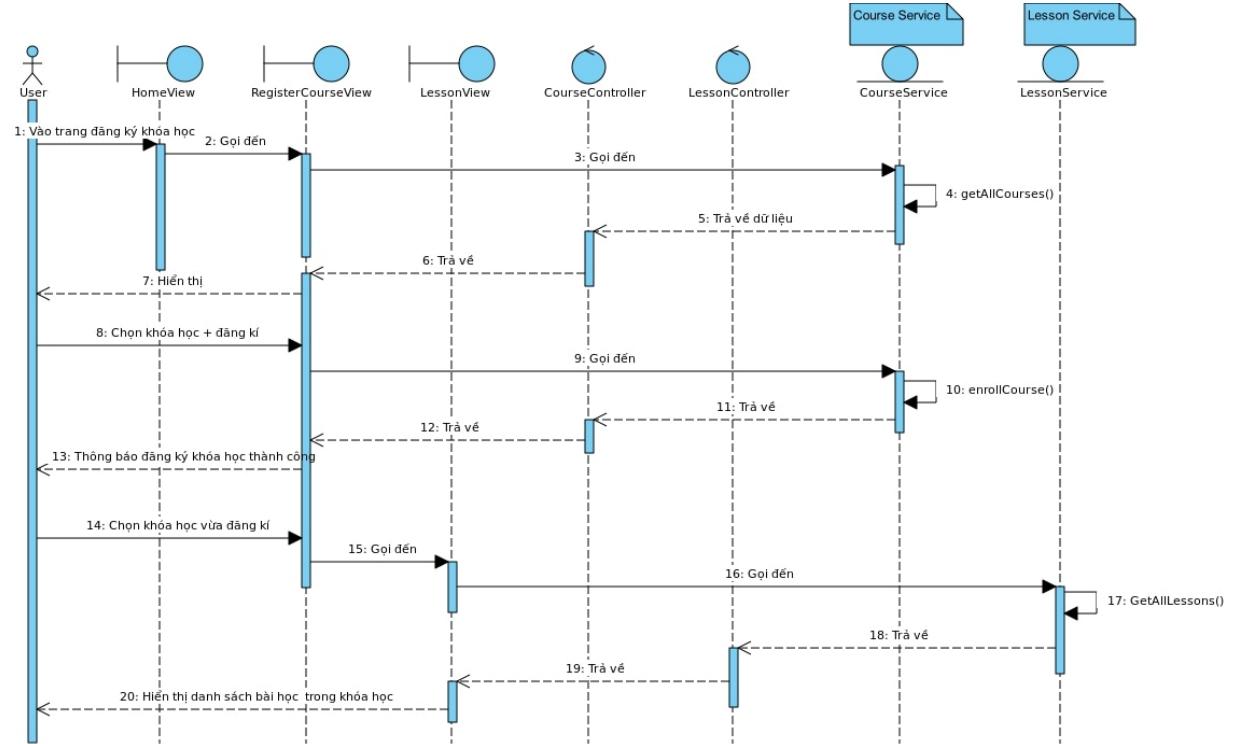
### c.Quiz



Hình 9: Biểu đồ hoạt động “Làm quiz”

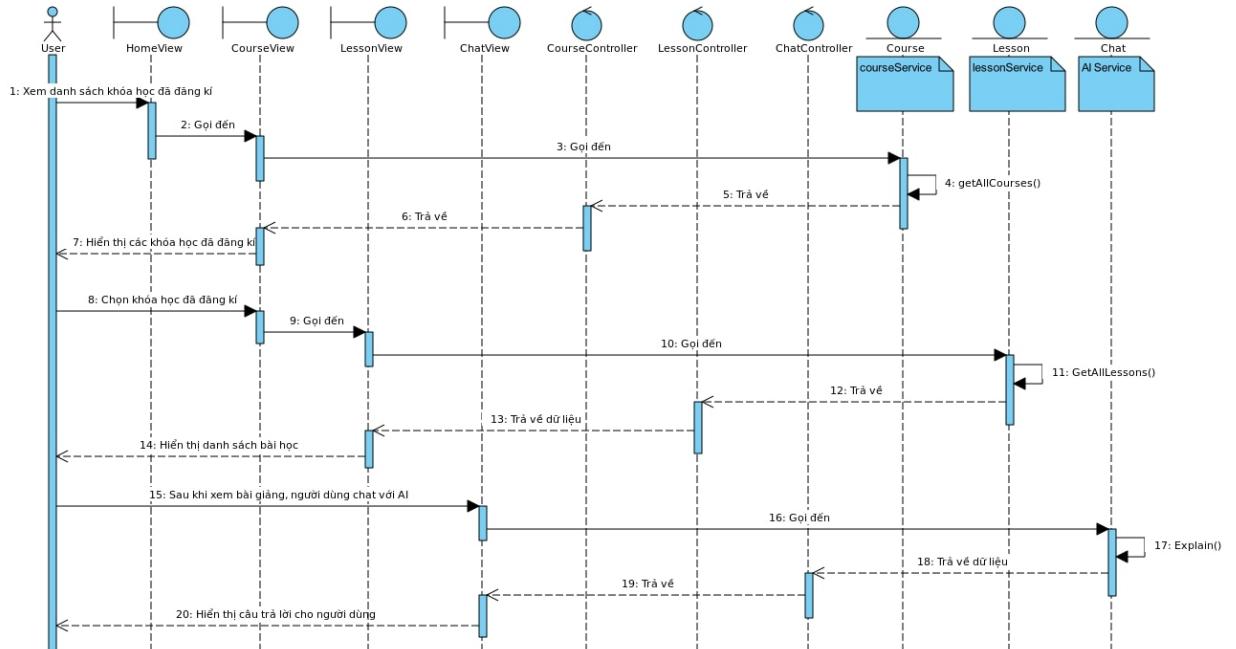
### 1.3. Sequence Diagram

#### a. Người dùng đăng ký khóa học



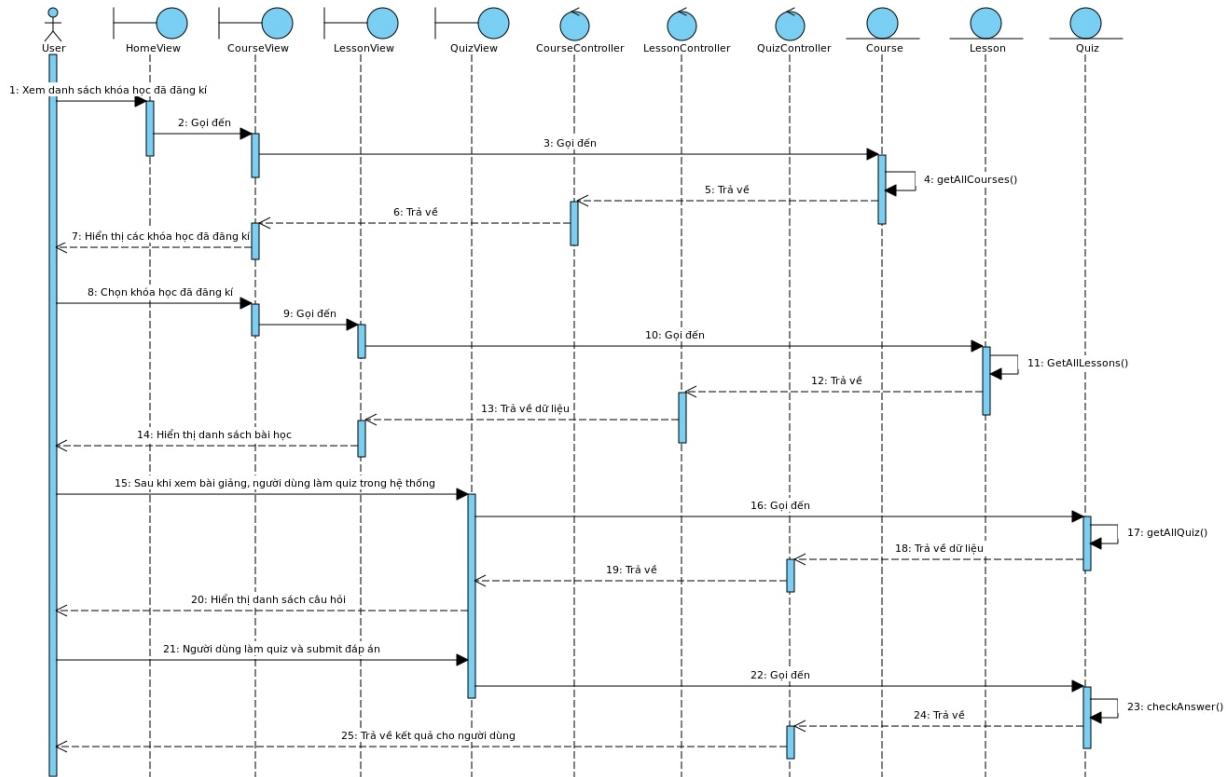
Hình 10: Biểu đồ tuần tự “Đăng ký khóa học”

#### b. Người dùng tương tác với Teaching Agent



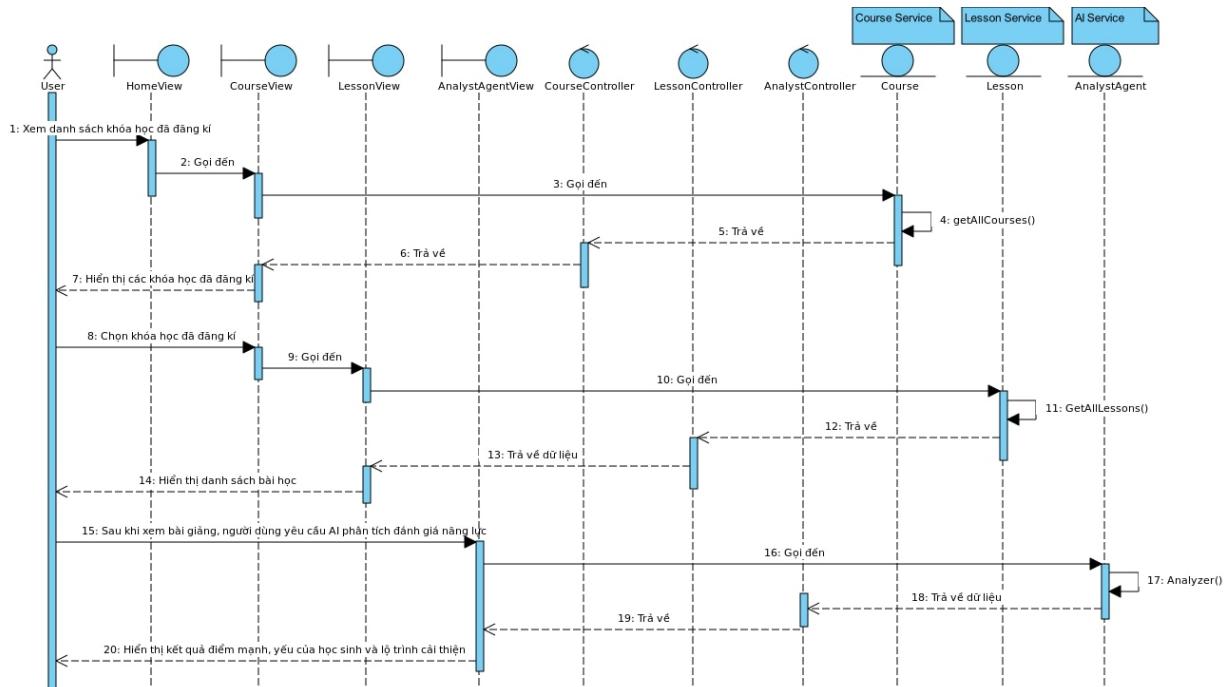
Hình 11: Biểu đồ tuần tự “Teaching Agent”

### c. Người dùng làm bài tập



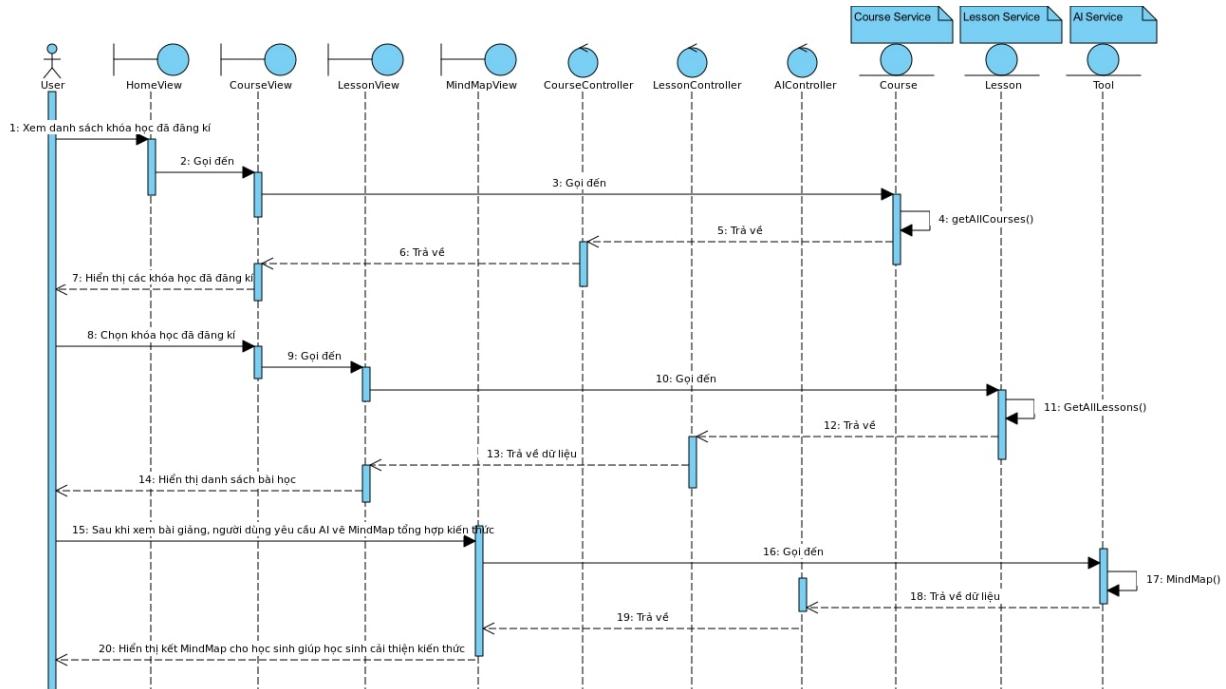
Hình 12: Biểu đồ tuần tự “Làm quiz”

### d. Người dùng yêu cầu hệ thống phân tích đánh giá năng lực



Hình 13: Biểu đồ tuần tự “Đánh giá năng lực người dùng”

## e. Người dùng yêu cầu tạo MindMap



Hình 14: Biểu đồ tuần tự “Tạo Mindmap”

## 1.4. Biểu đồ lớp toàn hệ thống

### a. Xác định các lớp của hệ thống

Để xác định các lớp của hệ thống, ta sử dụng kỹ thuật trích danh từ trong các use case và kịch bản. Các danh từ thu được là:

*Người dùng, hệ thống, bài học, khóa học, khóa học đã được đăng ký, kiểm tra, câu hỏi, đáp án, tiến trình, lần thử, đoạn chat, phụ đề, AI agent.*

Loại bỏ các danh từ nằm ngoài phạm vi mục đích của hệ thống, các danh từ trùng lặp và các danh từ làm thuộc tính của các lớp, ta thu được các danh từ làm lớp thực thể sau:

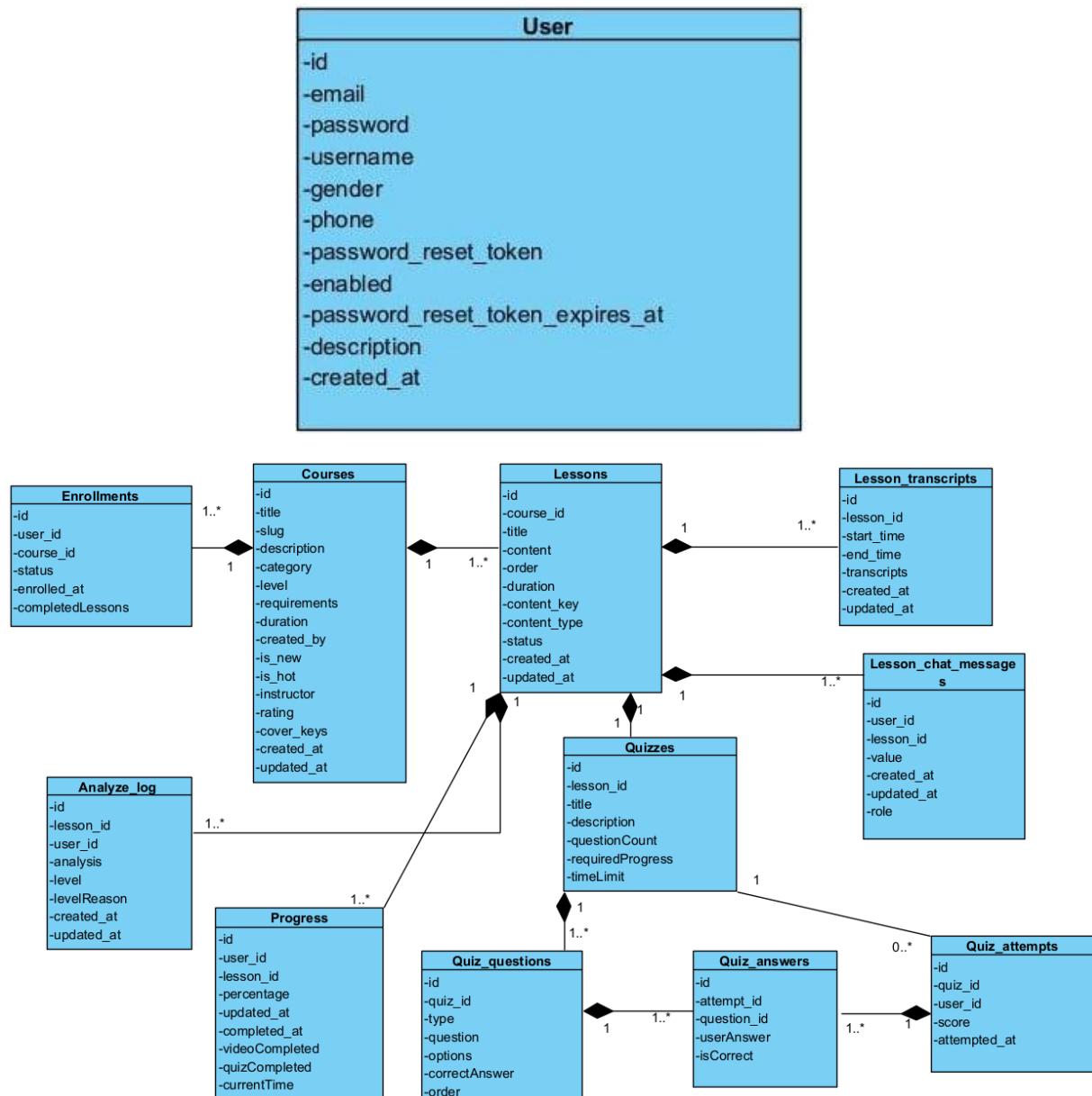
Những lớp xác định bao gồm: Người dùng (User), khóa học (Course), bài học (Lesson), khóa học được đăng ký (Enrollment), kiểm tra (Quiz), câu hỏi (Quiz Question), đáp án (Quiz Answer), lần thử (Quiz Attempt), tiến trình (Progress), đoạn chat (Lesson Chat\_message), phụ đề (Lesson Transcript).

### b. Xác định mối quan hệ giữa các thực thể

- Một khóa học thì chứa nhiều bài học → Quan hệ Course - Lesson là 1 - n.
- Một khóa học có thể được nhiều người dùng đăng ký → Quan hệ Course - Enrollment là 1 - n.
- Một khóa học có nhiều phụ đề với các khoảng thời gian khác nhau → Quan hệ Lesson - Lesson Transcript là 1 - n.
- Một bài học có nhiều đoạn tin nhắn chat → Quan hệ Lesson - Lesson Chat\_message là 1 - n.

- Mỗi người dùng có bản ghi riêng cho mỗi bài học → Quan hệ Lesson - Progress là 1 - n
- Mỗi bài học thì có một bài kiểm tra đi kèm → Quan hệ Lesson - Quiz là 1 - 1.
- Một bài kiểm tra chứa nhiều câu hỏi → Quan hệ Quiz - Quiz Question là 1 - n.
- Nhiều người dùng có thể làm bài kiểm tra → Quan hệ Quiz - Quiz Answer là 1 - n.
- Một câu hỏi có thể được nhiều người dùng trả lời trong các lượt làm bài khác nhau → Quan hệ Quiz Question - Quiz Answer là 1 - n.
- Mỗi lượt làm bài chứa nhiều câu trả lời (một câu trả lời cho mỗi câu hỏi) → Quan hệ Quiz Attempt - Quiz Answer là 1 - n.

### c. Biểu đồ trích lớp thực thể



Hình 15: Biểu đồ trích lớp thực thể pha phân tích

## 1.5. Phân rã hệ thống theo miền chức năng

Dựa trên kết quả phân tích nghiệp vụ ở các mục trước, hệ thống được chia thành các miền chức năng nhằm xác định ranh giới nghiệp vụ và phạm vi xử lý của từng nhóm chức năng. Việc phân rã này giúp làm rõ cấu trúc logic của hệ thống, đồng thời tạo tiền đề để chuyển đổi các miền chức năng thành các dịch vụ độc lập trong kiến trúc microservice ở giai đoạn thiết kế.

Quá trình phân rã được thực hiện dựa trên các tiêu chí:

- Tính độc lập nghiệp vụ.
- Khả năng triển khai và mở rộng riêng biệt.
- Mức độ liên kết giữa các chức năng và mức độ phụ thuộc dữ liệu.
- Ranh giới tự nhiên của các thực thể trong mô hình domain.

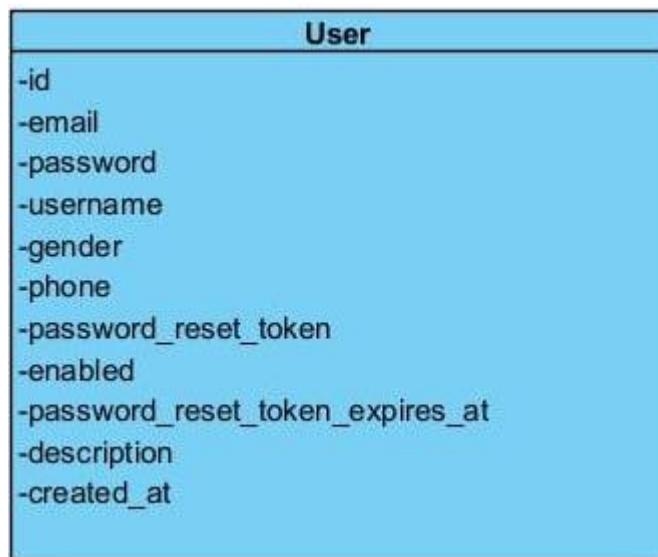
Hệ thống được phân rã thành các miền chức năng sau:

STT	Miền chức năng	Mô tả
1	Auth	Quản lý liên quan đến tài khoản như đăng ký, đăng nhập, thông tin người dùng.
2	Course	Quản lý các thông tin liên quan đến khóa học như danh sách khóa học, đăng ký khóa học, quản lý bài học trong từng khóa học.
3	AI Agent	Quản lý liên quan đến AI Agent như trả lời câu hỏi người dùng, Tldraw, đánh giá user qua kết quả quiz

Bảng 7 : Phân rã thành các miền chức năng

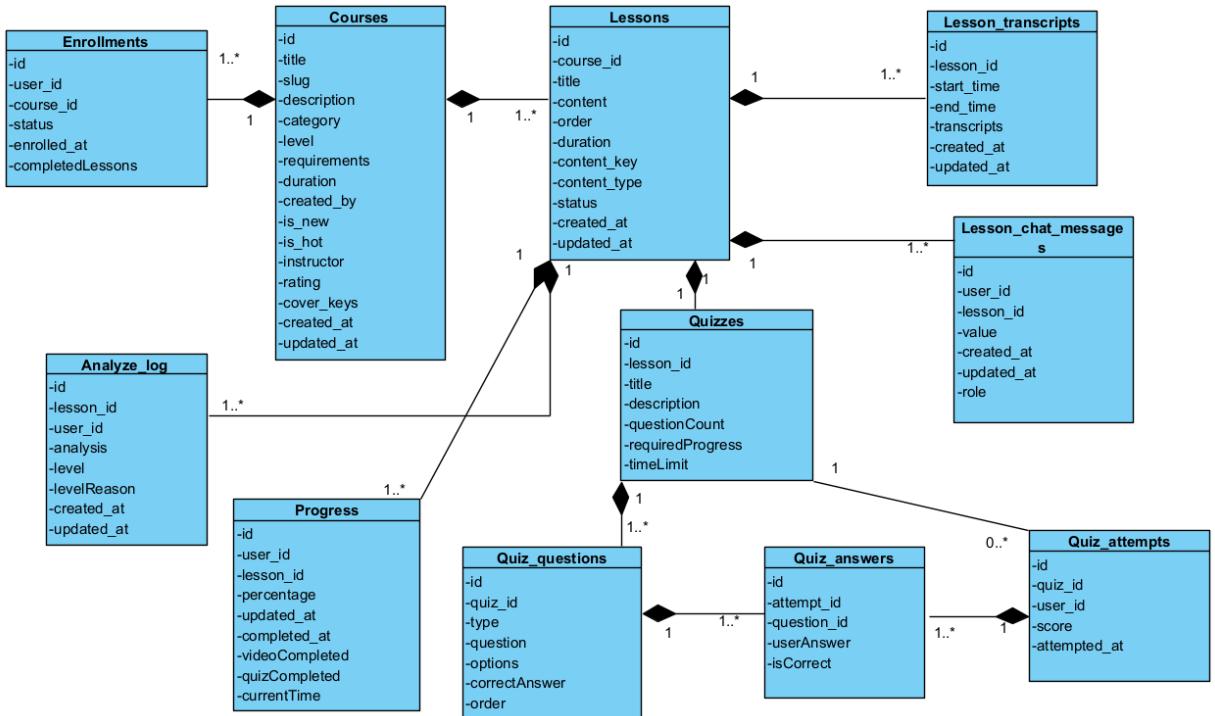
## 1.6. Biểu đồ chi tiết từng dịch vụ

### a. Biểu đồ lớp pha phân tích Auth-service



Hình 16: Biểu đồ lớp pha phân tích Auth-service

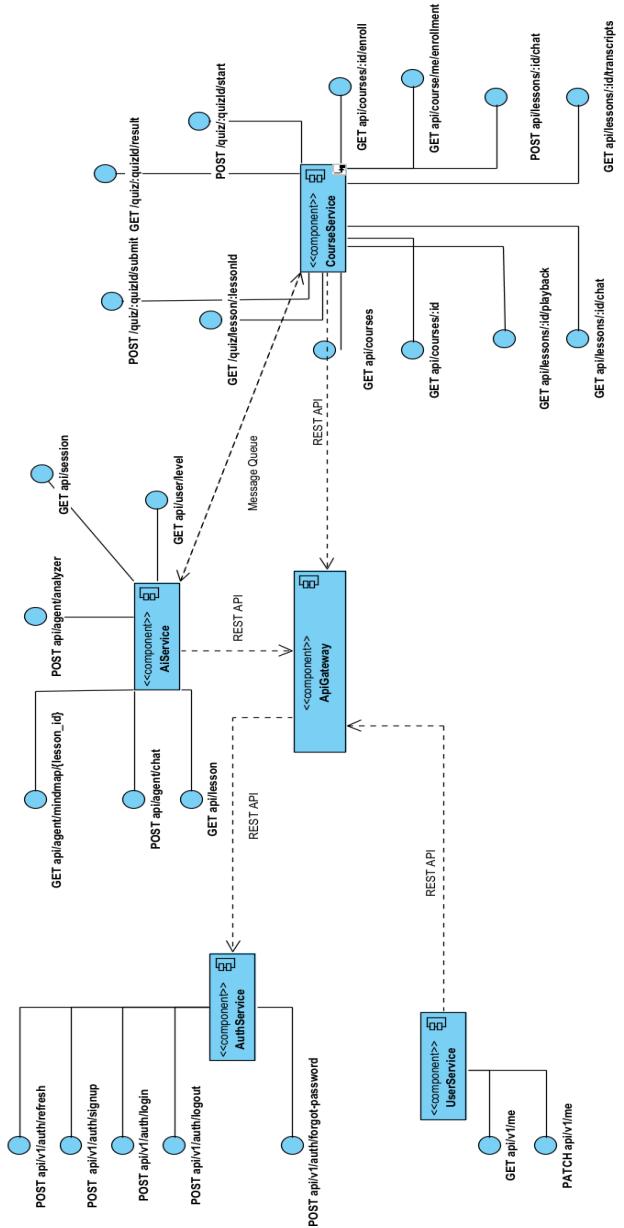
## b. Biểu đồ lớp pha phân tích Course-service



Hình 17: Biểu đồ lớp pha phân tích Course-service

## 2. Pha thiết kế

## 2.1. Thiết kế biểu đồ gói



Hình 18: Thiết kế tổng thể hệ thống

- Dựa trên thiết kế tổng thể hệ thống, do nhu cầu client cần giao tiếp thống nhất với nhiều microservice độc lập (Auth Service, Course Service, Lesson Service và AI Service) mà vẫn đảm bảo tuân thủ chặt chẽ nguyên tắc của kiến trúc vi dịch vụ (Microservice) – tránh client gọi trực tiếp đến các service backend và tập trung hoá các cross-cutting concerns như routing, authentication, CORS – Kong được lựa chọn làm API Gateway để làm điểm vào duy nhất cho toàn bộ request từ client. Để đạt được tính bảo mật và quản lý tập trung cao, hệ thống khai báo các service tương ứng với URL upstream thực tế của từng microservice (deploy trên Railway), đồng thời định nghĩa rõ ràng các route với path prefix riêng biệt: public routes dành cho Auth Service (không yêu cầu JWT, bao gồm login, signup, refresh, forgot/reset password, logout và OAuth2 flow) và protected routes cho

các service còn lại (Course, Lesson, AI và endpoint /api/v1/me). Các protected route/service được bảo vệ bằng JWT plugin (validate claim "iss" và "exp" dựa trên secret từ Auth Service), trong khi CORS plugin được áp dụng toàn cục để chỉ cho phép origin từ domain production và localhost, hỗ trợ đầy đủ methods, header Authorization/Content-Type, credentials và preflight cache 3600 giây. Nhờ đó, mọi request từ client đều đi qua Gateway để được routing chính xác, validate token tập trung và xử lý CORS thống nhất trước khi forward đến microservice tương ứng, đảm bảo an toàn, dễ maintain và không expose trực tiếp URL backend.

```
---  
  name: auth-service  
  url: https://doan2025-production.up.railway.app  
  tags: ["auth"]  
  routes:  
    # PUBLIC routes (Không JWT)  
    - name: auth-public-routes  
      paths:  
        - /api/v1/auth/login  
        - /api/v1/auth/signup  
        - /api/v1/auth/refresh  
        - /api/v1/auth/forgot-password  
        - /api/v1/auth/reset-password  
        - /api/v1/auth/logout  
        - /api/v1/auth/oauth2/exchange  
        - /oauth2/authorization/google  
        - /login/oauth2/code/google
```

Hình 19 : Cấu hình chi tiết Auth Service và các public routes trong Kong Gateway

```
#-----
# AI Service
#-----
- name: ai-service
  url: https://virtual-classroom-ai-agent-production.up.railway.app
  tags: [ "ai" ]
  routes:
    - name: ai-routes
      paths:
        - /api/v1/ai
        - /api/agent/chat
        - /api/agent/mindmap
        - /api/agent/analyzer
        - /api/session
        - /api/user/level
```

Hình 20 : Cấu hình chi tiết AI Service và các protected routes trong Kong Gateway (yêu cầu JWT validation).

```
#-----
# JWT CHO course-service
#-----
- name: jwt
  service: course-service
  tags: [ "jwt-plugin", "courses", "protected" ]
  config:
    key_claim_name: "iss"
    claims_to_verify: [ "exp" ]
    run_on_preflight: false
```

Hình 21 : Cấu hình JWT plugin bảo vệ Course Service trong Kong Gateway (áp dụng cho tất cả protected routes).

```

- name: cors
  tags: [ "cors-plugin" ]
  config:
    origins:
      - "https://ai-agent-virtual-classroom-fe.vercel.app"
      - "http://localhost:3000"
    methods: [ "GET", "POST", "PUT", "DELETE", "PATCH", "OPTIONS" ]
    headers: [ "Authorization", "Content-Type" ]
    credentials: true
    max_age: 3600
    preflight_continue: false

```

*Hình 22 : Cấu hình CORS plugin toàn cục.*

- Dựa trên thiết kế tổng thể hệ thống, do nhu cầu sử dụng dữ liệu được quản lý tập trung bên phía Course Service của AI Service tuy nhiên vẫn cần đảm bảo tuân thủ chặt chẽ nguyên tắc của thiết kế vi dịch vụ (Microservice), RabbitMQ được lựa chọn sử dụng như một Message Broker để điều phối các thông điệp qua lại giữa 2 service này để luân chuyển dữ liệu cần thiết từ Course Service cho AI Service. Để giao tiếp đạt được độ chính xác cao, mỗi service cần chuẩn bị 2 luồng message queue: ai\_to\_course (queue\_out đối với AI Service và queue\_in với Course Service) và course\_to\_ai (queue\_in đối với AI Service và queue\_out với Course Service) và được kết nối với nhau thông qua một RABBITMQ\_URL thống nhất. Khi bắt đầu một luồng giao tiếp giữa 2 dịch vụ, một bên (thường là AI Service) sẽ gửi ra một message có chứa pattern thống nhất được chuẩn bị sẵn từ Course Service vào queue\_out (ai\_to\_course) của AI Service, Course Service sẽ thực hiện truy vấn và gửi lại dữ liệu tương ứng với pattern trong queue\_in (ai\_to\_course) của Course Service nhận được vào queue\_out (course\_to\_ai) cho AI Service. AI Service nhận dữ liệu trong queue\_in (course\_to\_ai) để thực thi các tác vụ liên quan.

Các pattern được set up bên phía Course Service:

```

    /**
     * Get quiz data for AI service
     */
    @MessagePattern('GET_QUIZ_DATA')
    async getQuizData(...)

}

/**
 * Save analysis log from AI service (Fire-and-Forget)
 */
@MessagePattern('SAVE_ANALYSIS_LOG')
async saveAnalysisLog(...)

}

/**
 * Get quiz attempts for AI service (RPC)
 */
@MessagePattern('GET_QUIZ_ATTEMPTS')
async getQuizAttempts(...)

}

/**
 * Get analysis history for AI service (RPC)
 */
@MessagePattern('GET_ANALYSIS_HISTORY')
async getAnalysisHistory(...)

}

/**
 * Get lesson transcript for AI context (RPC)
 */
@MessagePattern('GET_LESSON_TRANSCRIPT')
async getLessonTranscript(...)

}

/**
 * Get full lesson data for AI Vector DB sync (RPC)
 * AI Service calls this after receiving lesson.updated event
 */
@MessagePattern('GET_LESSON')
async getLesson(...)

}

```

```

    /**
     * Handle SAVE_CHAT_MESSAGES from AI Service
     * Saves user and AI messages in a single transaction
     */
    @MessagePattern('SAVE_CHAT_MESSAGES')
    async saveChatMessages(...)

}

/**
 * Get chat history for a lesson and user
 */
@MessagePattern('GET_CHAT_HISTORY')
async getChatHistory(...)

}

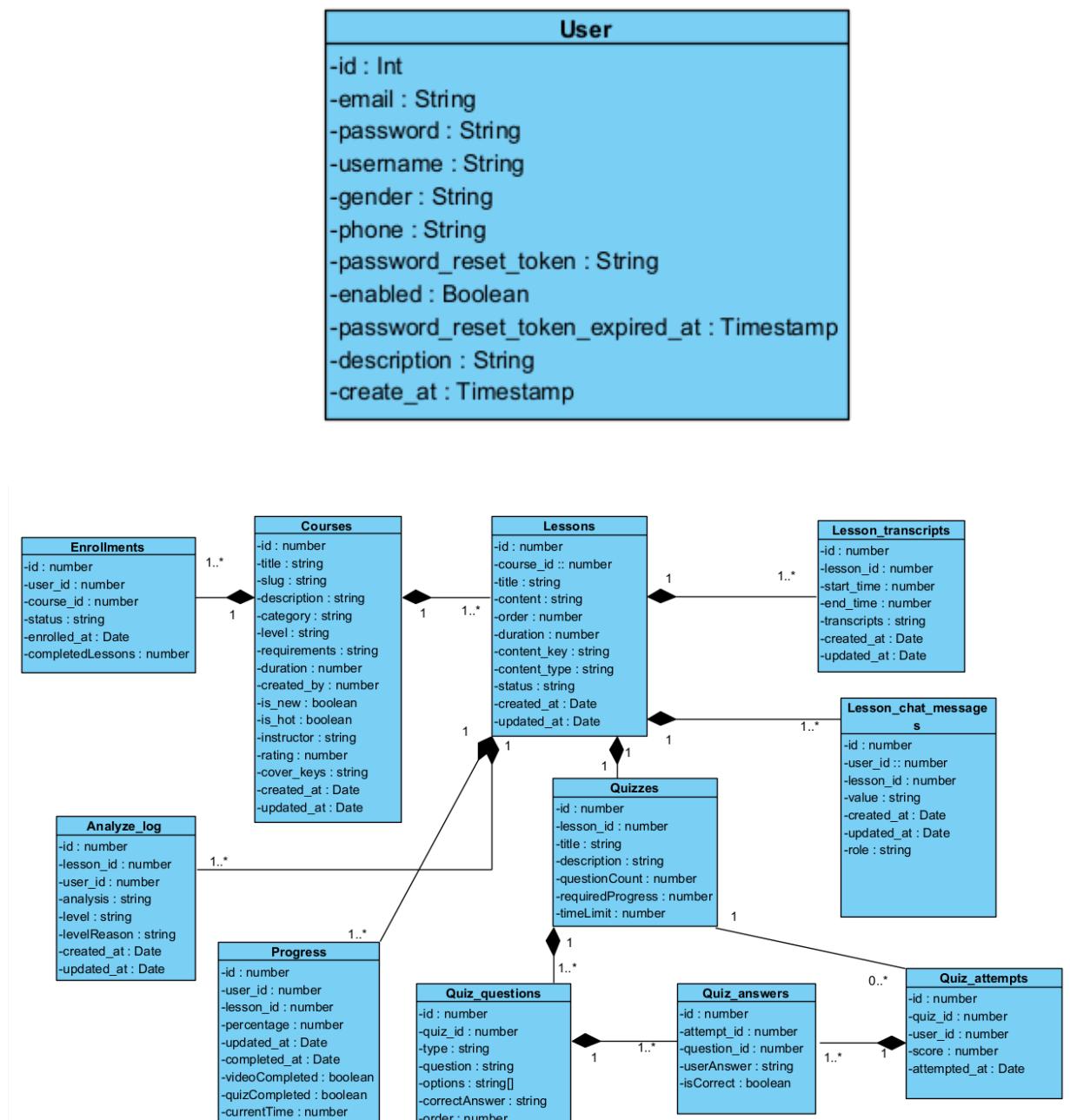
```

Hình 23: Các Message Pattern

## 2.2. Thiết kế biểu đồ lớp thực thể

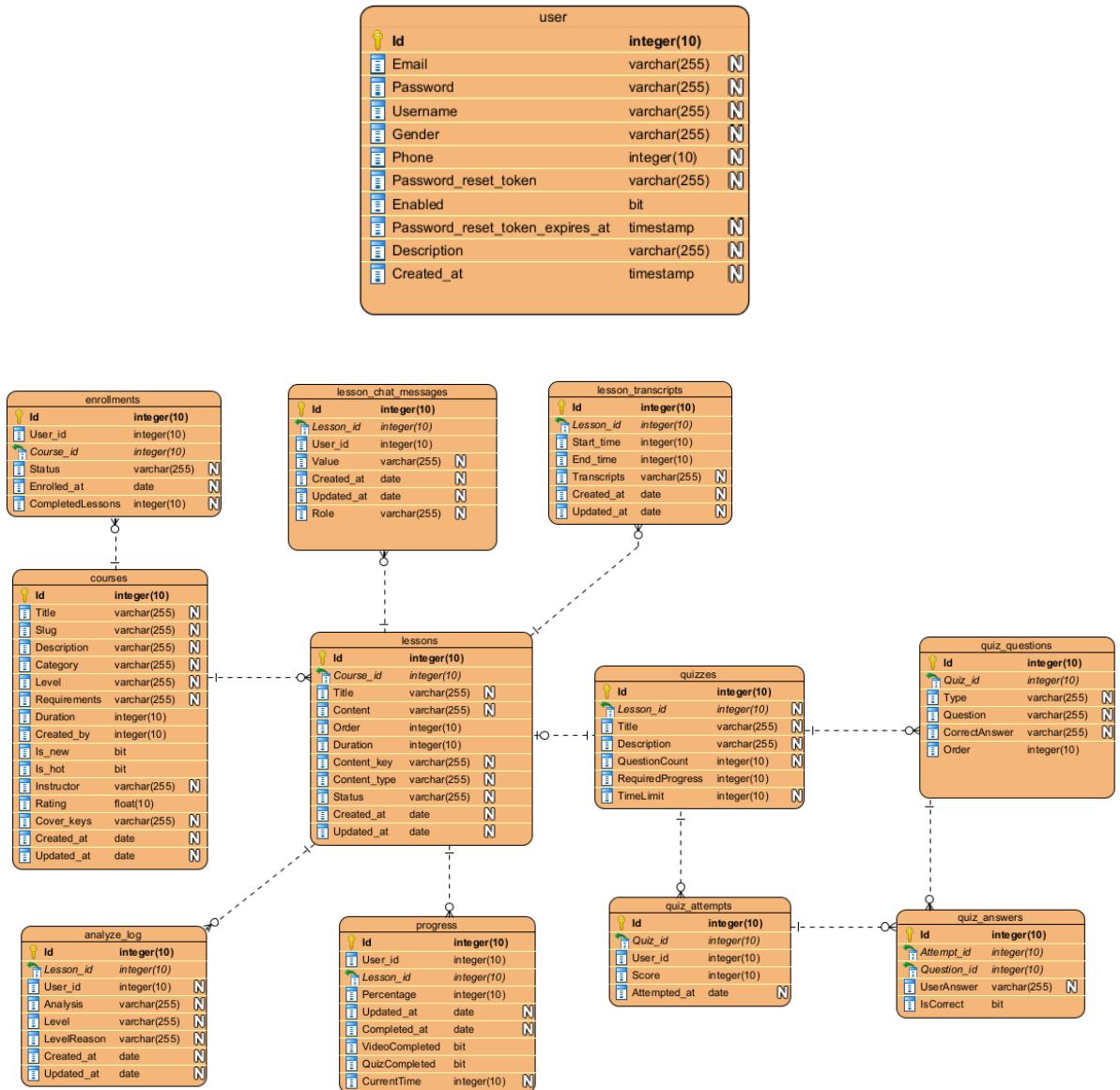
Các lớp đều bổ sung thêm thuộc tính id.

Thuộc tính của các lớp được bổ sung thuộc tính theo kiểu của ngôn ngữ lập trình TypeScript



Hình 24: Biểu đồ lớp thực thể pha thiết kế

## 2.3. Thiết kế cơ sở dữ liệu

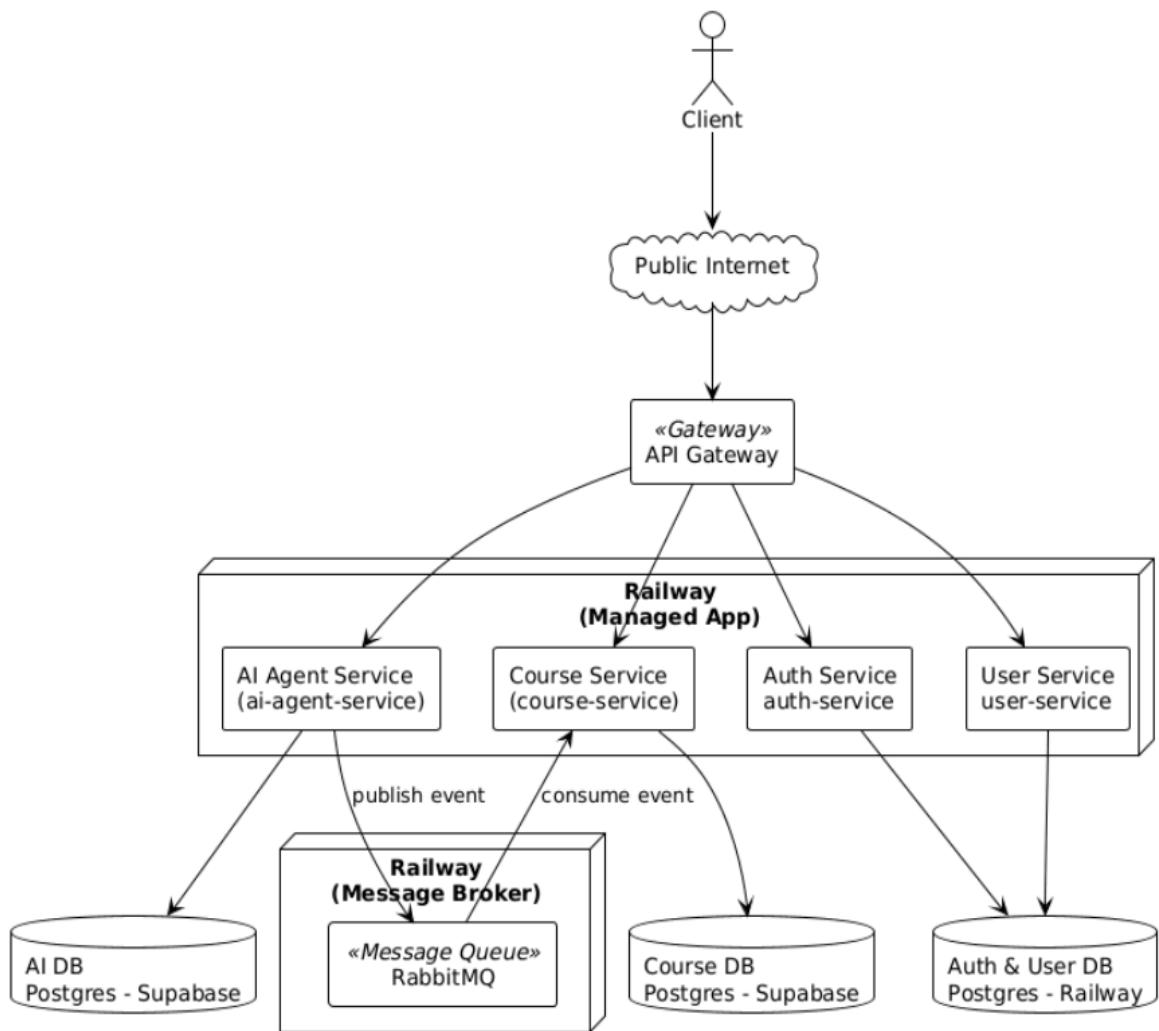


Hình 25: Biểu đồ cơ sở dữ liệu toàn hệ thống

## 2.4. Thiết kế kiến trúc microservice

Hệ thống được xây dựng dựa trên kiến trúc Microservice gồm 5 service:

- Auth-service
- Course-service
- AI Agent-service
- API Gateway



Hình 26: Biểu đồ tổng quan toàn hệ thống

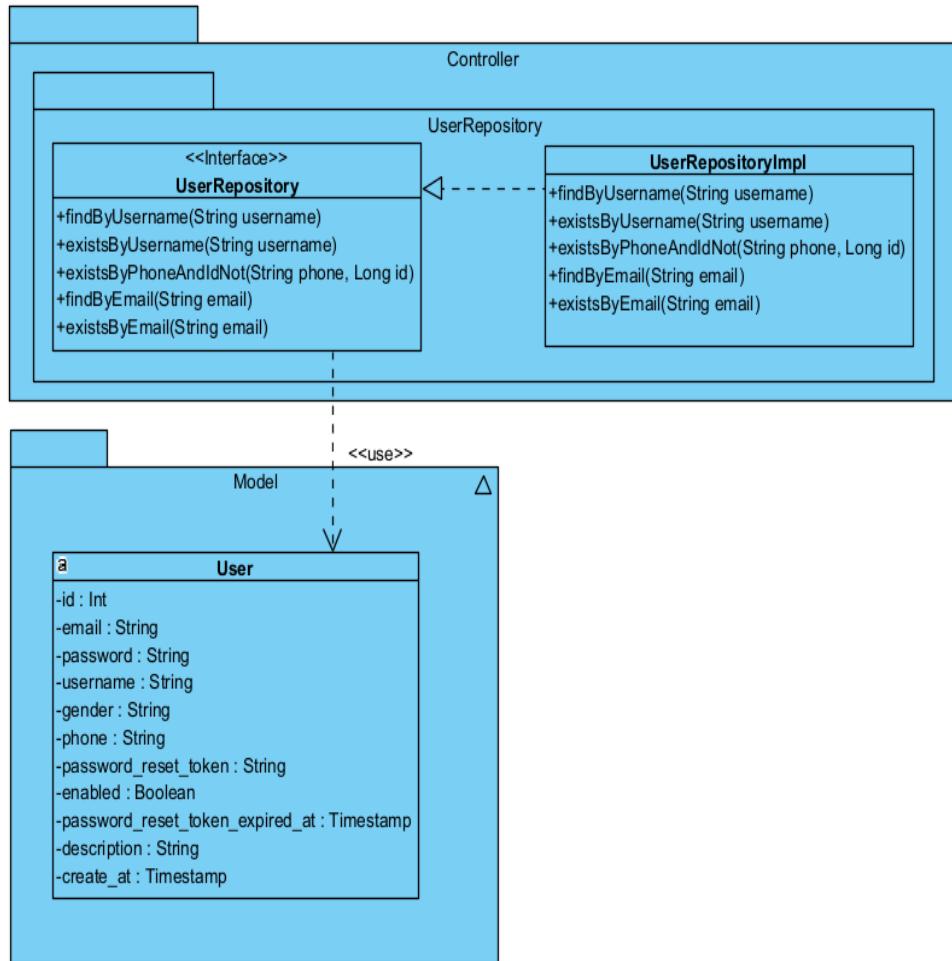
STT	Tên	Mô tả
1	Auth-service	Quản lý xác thực và bảo mật tài khoản người dùng, bao gồm đăng ký, đăng nhập, phát hành và làm mới JWT, đăng xuất, quên mật khẩu và xác thực OAuth2.
2	User-service	Quản lý thông tin hồ sơ người dùng sau khi đã xác thực, bao gồm xem và cập nhật thông tin cá nhân như số điện thoại, giới tính, mô tả hồ sơ.
2	Course-service	Quản lý các thông tin liên quan đến khóa học như danh sách khóa học, đăng ký khóa học, quản lý bài học trong từng khóa học.
3	AI Agent-service	Quản lý liên quan đến AI Agent như trả lời câu hỏi người dùng, Tldraw, đánh giá user qua lịch sử tương tác.
4	API Gateway	Quản lý xác thực các yêu cầu từ client, thực hiện định tuyến đến các service tương ứng.

Bảng 8 : Các thành phần hệ thống

## 2.5. Thiết kế chi tiết từng dịch vụ

### a. Auth-service + User-service

- Biểu đồ gói pha thiết kế



Hình 27: Biểu đồ gói pha thiết kế Auth-service và User-service

- Code tương ứng biểu đồ gói:

```

@RestController no usages
@RequestMapping("/api/v1/auth")
@RequiredArgsConstructor
public class AuthController {
    private final AuthService authService;

    @PostMapping("/signup") no usages
    public ResponseEntity<ApiResponse<LoginResponse>> signup(@Valid @RequestBody SignupRequest req) {...}

    @PostMapping("/login") no usages
    public ResponseEntity<ApiResponse<LoginResponse>> login(@Valid @RequestBody LoginRequest req) {...}

    @PostMapping("/refresh") no usages
    public ResponseEntity<ApiResponse<TokenPair>> refresh(@Valid @RequestBody TokenRequest req) {...}

    @PostMapping("/forgot-password") no usages
    public ResponseEntity<ApiResponse<Void>> forgotPassword(@Valid @RequestBody ForgotPasswordRequest req)

    @PostMapping("/reset-password") no usages
    public ResponseEntity<ApiResponse<Void>> resetPassword(@Valid @RequestBody ResetPasswordRequest req)

    @PostMapping("/logout") no usages
    public ResponseEntity<ApiResponse<Void>> logout(@Valid @RequestBody TokenRequest req) {...}

    @PostMapping("/oauth2/exchange") no usages
    public ResponseEntity<ApiResponse<LoginResponse>> exchangeOAuth2Code(@RequestParam String code) {...}
}

```

Hình 28: Auth Controller

Đây là **AuthController** – lớp RestController chính trong Auth Service (Spring Boot), chịu trách nhiệm xử lý các request HTTP từ client (qua API Gateway) liên quan đến xác thực người dùng, đăng ký, quên/đặt lại mật khẩu, refresh token, logout và OAuth2 flow.

Cấu trúc chính:

- `@RestController` và `@RequestMapping("/api/v1/auth")`: Định nghĩa base path cho tất cả endpoint là `/api/v1/auth/*`.
- `@RequiredArgsConstructor`: Tự động inject dependency `AuthService`.
- Tất cả method đều trả về `ResponseEntity<ApiResponse<T>>` → đóng gói response thống nhất (bao gồm status, message, data) theo chuẩn REST API của dự án.

Các endpoint chính (public – không yêu cầu JWT):

- POST /signup: Xử lý đăng ký tài khoản mới (nhận `SignupRequest` → validate bằng `@Valid` → gọi `authService.signup()` → trả `LoginResponse` chứa token nếu thành công).
- POST /login: Xử lý đăng nhập thường (username/password) (nhận `LoginRequest` → validate → gọi `authService.login()` → trả `LoginResponse` chứa access + refresh token).

- POST /refresh: Làm mới access token (nhận TokenRequest chưa refresh token → validate → gọi authService.refresh() → trả TokenPair mới).
- POST /forgot-password: Yêu cầu quên mật khẩu (nhận ForgotPasswordRequest thường chưa email → gọi authService.forgotPassword() → gửi email reset, trả Void).
- POST /reset-password: Đặt lại mật khẩu mới (nhận ResetPasswordRequest chưa token + new password → validate → gọi authService.resetPassword() → trả Void).
- POST /logout: Đăng xuất (nhận TokenRequest → gọi authService.logout() → invalidate refresh token hoặc cleanup, trả Void).
- POST /oauth2/exchange: Endpoint nội bộ hỗ trợ OAuth2 login (nhận authorization code từ provider như Google → gọi authService.exchangeOAuth2Code() → đổi lấy token và trả LoginResponse).

Ý nghĩa trong thiết kế hệ thống:

- Controller này là tầng presentation của Auth Service, chỉ tập trung nhận request, validate input, gọi service layer và format response → tuân thủ clean architecture (không chứa business logic).
- Tất cả endpoint đều là public routes (không cần JWT) → phù hợp với cấu hình Kong Gateway.
- Hỗ trợ đầy đủ flow xác thực hiện đại: JWT-based (access + refresh), forgot/reset password qua email, logout an toàn và OAuth2 với Google.
- Request từ client/frontend sẽ gọi qua Gateway → Kong forward đến Auth Service mà không cần token.

```

@RestController no usages
@RequestMapping("/api/v1/me")
@RequiredArgsConstructor
public class UserController {
    private final UserService userService;

    @GetMapping no usages
    @PreAuthorize("isAuthenticated()")
    public ResponseEntity<ApiResponse<ProfileResponse>> getMe(Authentication auth) {...}

    @PatchMapping no usages
    @PreAuthorize("isAuthenticated()")
    public ResponseEntity<ApiResponse<ProfileResponse>> updateMe(Authentication auth,
        @Valid @RequestBody ProfileUpdateReq req) {...}
}

```

Hình 29: User Controller

Đây là **UserController** – lớp RestController trong Auth Service (hoặc User Service riêng nếu tách module), chịu trách nhiệm xử lý các request HTTP từ client (qua API Gateway) liên quan đến thông tin profile cá nhân của người dùng đã đăng nhập (endpoint /api/v1/me).

Cấu trúc chính:

- `@RestController` và `@RequestMapping("/api/v1/me")`: Định nghĩa base path cho các endpoint profile là /api/v1/me.
- `@RequiredArgsConstructor`: Tự động inject dependency UserService (lớp service xử lý nghiệp vụ profile: get và update).
- Tất cả method đều trả về `ResponseEntity<ApiResponse<T>>` → đóng gói response thống nhất (status, message, data) theo chuẩn REST API của dự án.
- Bảo mật: Sử dụng `@PreAuthorize("isAuthenticated()")` từ Spring Security → chỉ cho phép user đã authenticated (có JWT hợp lệ) mới truy cập được các endpoint này.

Các endpoint chính (protected – yêu cầu JWT):

- GET /api/v1/me: Lấy thông tin profile của user hiện tại (nhận Authentication auth từ Spring Security context → gọi userService.getProfile() → trả ProfileResponse chứa thông tin user như username, email, phone, gender, description...).
- PATCH /api/v1/me: Cập nhật profile (nhận Authentication auth + ProfileUpdateReq req → validate bằng `@Valid` → gọi userService.updateProfile() → trả ProfileResponse đã cập nhật).

Ý nghĩa trong thiết kế hệ thống:

- Controller này là tầng presentation cho nghiệp vụ profile, chỉ xử lý request/response và gọi service layer → tuân thủ clean architecture.
- Các endpoint đều là protected routes (yêu cầu JWT) → khác biệt rõ ràng với các public routes trong AuthController (login/signup/refresh...).
- Bảo mật được thực hiện hai lớp:
  - Tầng Gateway: Kong validate JWT trước khi forward request (như cấu hình me-route với JWT plugin ở phần trước).
  - Tầng Service: Spring Security `@PreAuthorize` kiểm tra lại authentication.
- Client gọi qua Gateway (ví dụ: <https://gateway-domain/api/v1/me>) → Kong check token → forward đến Auth/User Service → Spring Security extract user từ JWT và inject vào method.

```

27     @Service no usages
28     @RequiredArgsConstructor
29     @Slf4j
30     public class AuthServiceImpl implements AuthService {
31         private final UserRepository userRepo;
32         private final RefreshTokenRepository rtRepo;
33         private final PasswordEncoder encoder;
34         private final JwtService jwtService;
35         private final EmailService emailService;
36
37         @Value("${jwt.refresh-exp-days:7}") private long refreshExpDays; 2
38
39         @Override 1 usage
40         @Transactional
41     ↗ @ >     public LoginResult signup(SignupRequest req) {...}
53
54         @Override 1 usage
55         @Transactional
56     ↗ @ >     ⚡ public LoginResult login(LoginRequest req) {...}
77
78     @ >     private LoginResult generateTokensAndLoginResult(User user) {...}
89
90         @Override 1 usage
91         @Transactional
92     ↗ >     public TokenPair refresh(String refreshPlain) {...}
114
128
129
130     ↗ >     @Override 1 usage
131         @Transactional
132     ↗ >     public void forgotPassword(String email) {...}
141
142
143
144     ↗ >     @Override 1 usage
145         @Transactional
146     ↗ >     public void resetPassword(String token, String newPassword) {...}
161
162
163
164     ↗ >     @Override 1 usage
165         @Transactional(readOnly = true)
166     ↗ >     public LoginResult getOAuth2TokenByCode(String code) {...}
172
173
174
175     ↗ >     @Override 1 usage
176         @Transactional
177     ↗ >     public void deleteOAuth2Code(String code) { OAuth2LoginSuccessHandler.removeCode(code); }
178     }
179
180

```

Hình 30: Code AuthServiceImpl

Đây là code khung của AuthServiceImpl – lớp implement chính của AuthService trong Spring Boot, là service layer chịu trách nhiệm xử lý toàn bộ nghiệp vụ liên quan đến xác thực (authentication), đăng ký (registration), quản lý token và quên/đặt lại mật khẩu.

- Các dependency chính:

- UserRepository, RefreshTokenRepository: Dùng Spring Data JPA để tương tác với database (lưu trữ và truy vấn thông tin user và refresh token).
  - PasswordEncoder: Mã hóa mật khẩu.
  - JwtService: Tạo và xử lý JWT access token / refresh token.
  - EmailService: Gửi email (dùng cho chức năng forgot password, gửi link reset).
- Các chức năng chính:
    - signup(): Xử lý đăng ký người dùng mới (nhận SignupRequest, tạo user, mã hóa password, có thể gửi email xác nhận).
    - login(): Xử lý đăng nhập (nhận LoginRequest, xác thực credential, trả về LoginResult chứa token).
    - generateTokensAndLoginResult(): Phương thức hỗ trợ tạo cặp access + refresh token và đóng gói kết quả login.
    - refresh(): Làm mới access token dựa trên refresh token hợp lệ.
    - forgotPassword(): Xử lý yêu cầu quên mật khẩu.
    - resetPassword(): Đặt lại mật khẩu mới dựa trên token reset.
    - getAuth2TokenByCode() và deleteOAuth2Code(): Hỗ trợ OAuth2 login.

```

14     @Service no usages
15     @RequiredArgsConstructor
16     public class UserServiceImpl implements UserService {
17         private final UserRepository userRepo;
18         private final PasswordEncoder encoder;
19
20         @Override no usages
21         @Transactional(readOnly = true)
22         <@> public ProfileResponse getProfile(String username) {...}
23
24         @Override no usages
25         @Transactional
26         <@> public ProfileResponse updateProfile(String username, ProfileUpdateReq req) {...}
27
28         @> private ProfileResponse toDto(User u) {...}
29
30     }
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

```

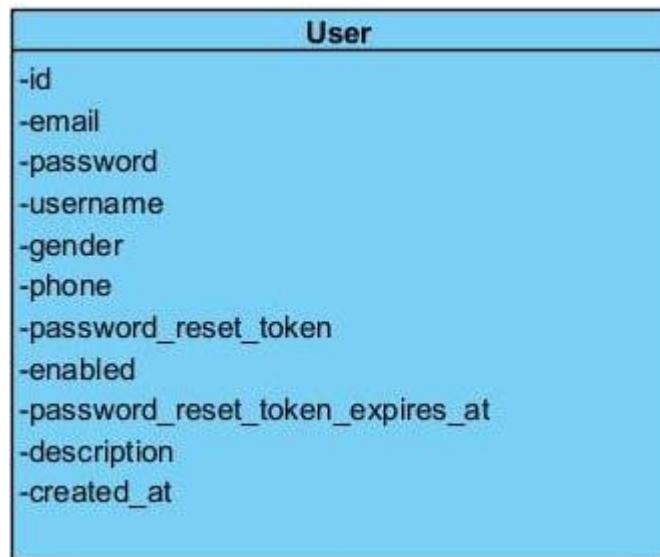
Hình 31: Code UserServiceImpl

Đây là code khung của UserServiceImpl – lớp implement chính của UserService trong Spring Boot, là service layer chịu trách nhiệm xử lý nghiệp vụ liên quan đến thông tin profile người dùng (lấy và cập nhật).

- Các dependency chính:
  - **UserRepository**: Dùng Spring Data JPA để tương tác với database (truy vấn và cập nhật entity User).

- **PasswordEncoder**: Mã hóa mật khẩu (dùng khi update profile có thay đổi password).
- **Các chức năng chính:**
  - **getProfile()**: Lấy thông tin profile của người dùng theo username.
  - **updateProfile()**: Cập nhật thông tin profile.
  - **toDto()**: Phương thức hỗ trợ chuyển đổi entity User sang DTO để trả về client, tránh expose entity trực tiếp.

Biểu đồ lớp pha thiết kế



Hình 32: Biểu đồ lớp pha thiết kế Auth-service và User-service

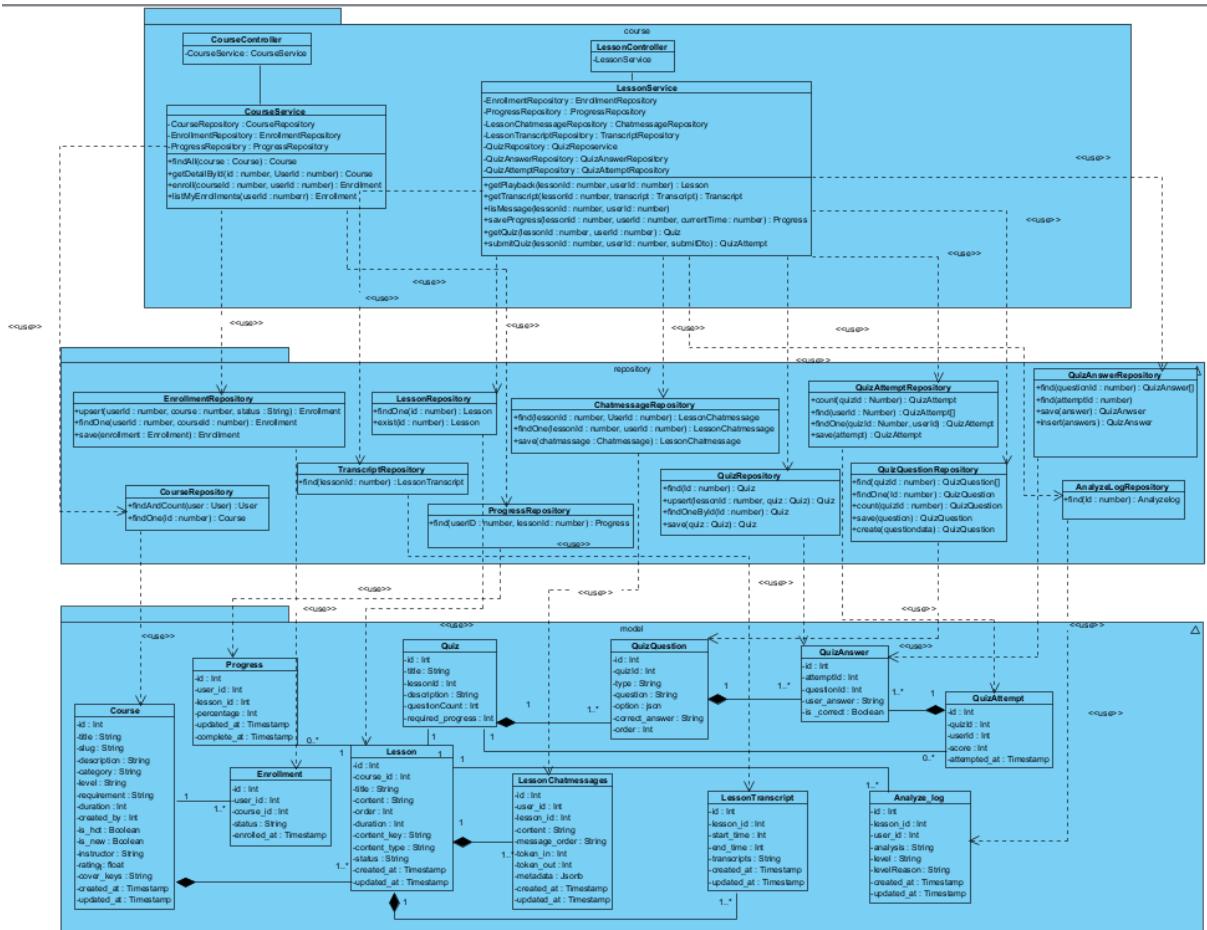
Lược đồ cơ sở dữ liệu

user		
<b>Id</b>	integer(10)	
Email	varchar(255)	N
Password	varchar(255)	N
Username	varchar(255)	N
Gender	varchar(255)	N
Phone	integer(10)	N
Password_reset_token	varchar(255)	N
Enabled	bit	
Password_reset_token_expires_at	timestamp	N
Description	varchar(255)	N
Created_at	timestamp	N

Hình 33: Lược đồ cơ sở dữ liệu Auth-service và User-service

### b. Course-service

- #### - Biểu đồ gói pha thiết kế



Hình 34: Biểu đồ gói pha thiết kế Course Service

- Code tương ứng biểu đồ gói:

```

    @Get(':id/details')
    @ApiOperation({ summary: 'Get course details (with progress if enrolled)' })
    @ApiOkResponse({ ... })
)
getDetailsById( ... )

@Post(':id/enroll')
@ApiOperation({ summary: 'Enroll current user to a course' })
@ApiOkResponse({ ... })
)
@ApiNotFoundResponse({ ... })
)
@ApiUnauthorizedResponse({ description: 'Thiếu/invalid Bearer token' })
enroll(@Param('id', ParseIntPipe) id: number, @CurrentUser() user: any) { ... }

@Post(':id/cancel-enrollment')
@ApiOperation({ summary: 'Cancel enrollment of current user' })
@ApiOkResponse({ ... })
)
@ApiNotFoundResponse({ ... })
)
@ApiUnauthorizedResponse({ description: 'Thiếu/invalid Bearer token' })
cancel(@Param('id', ParseIntPipe) id: number, @CurrentUser() user: any) { ... }

@Get('me/enrollments')
@ApiOperation({ summary: 'List my enrolled courses' })
@ApiOkResponse({ ... })
)
@ApiUnauthorizedResponse({ description: 'Thiếu/invalid Bearer token' })
myEnrollments(@CurrentUser() user: any) { ... }

```

*Hình 35: Course Controller*

Course Controller quản lý các API liên quan đến khóa học, yêu cầu xác thực JWT. Các chức năng chính:

- GET /courses/categories – Lấy danh sách danh mục khóa học.
- GET /courses – Lấy danh sách khóa học (có phân trang, filter).
- GET /courses/:id/details – Lấy chi tiết khóa học (kèm tiến độ nếu đã đăng ký).
- POST /courses/:id/enroll – Đăng ký khóa học.
- GET /courses/me/enrollments – Lấy danh sách khóa học đã đăng ký.

```

    @Get(':id/playback')
    @ApiOperation({ ... })
    @ApiOkResponse({ ... })
    getPlayback(@Param('id', ParseIntPipe) id: number, @CurrentUser() user: any) { ... }

    @Get(':id/transcripts')
    @ApiOperation({ summary: 'Get transcripts from time (seconds)' })
    @ApiOkResponse({ ... })
    getTranscripts( ... )

    @Get(':id/chat')
    @ApiOperation({ ... })
    @ApiOkResponse({ ... })
    listChat(@Param('id', ParseIntPipe) id: number, @CurrentUser() user: any) {
      return this.lessonService.listMessages(id, user.userId);
    }
  }
}

```

```

  @Get(':id/quiz')
  @ApiOperation({ ... })
  @ApiOkResponse({ ... })
  getQuiz(@Param('id', ParseIntPipe) id: number, @CurrentUser() user: any) {
    return this.lessonService.getQuiz(id, user.userId);
  }

  @Post(':id/quiz/submit')
  @ApiOperation({ ... })
  @ApiBody({ ... })
  @ApiOkResponse({ ... })
  submitQuiz( ... )

  @Get(':id/quiz/history')
  @ApiOperation({ ... })
  @ApiOkResponse({ ... })
  getQuizHistory( ... )
}

}

```

Hình 36: Lesson Controller

Lesson Controller quản lý các API liên quan đến bài học và quiz, yêu cầu xác thực JWT. Các chức năng chính:

- GET /lessons/:id/playback – Lấy thông tin phát video (URL video + danh sách bài học sidebar).

- GET /lessons/:id/transcripts – Lấy transcript (phụ đề) theo thời gian.
- GET /lessons/:id/chat – Lấy lịch sử chat của user trong bài học.
- GET /lessons/:id/quiz – Lấy câu hỏi quiz (yêu cầu đã xem ≥90% video).
- POST /lessons/:id/quiz/submit – Nộp bài quiz và tính điểm.
- GET /lessons/:id/quiz/history – Lấy lịch sử làm quiz.

```
@Injectable()
export class CourseService {
  constructor(
    @InjectRepository(Course) private readonly courseRepo: Repository<Course>,
    @InjectRepository(Enrollment)
    private readonly enrollmentRepo: Repository<Enrollment>,
    @InjectRepository(Progress)
    private readonly progressRepo: Repository<Progress>,
    @InjectRepository(Lesson) private readonly lessonRepo: Repository<Lesson>,
    private readonly s3Media: S3MediaService,
  ) {}

  async findAll(filter: FilterCourseDto) { ... }

  async getDetailsById(id: number, userId?: any) { ... }

  async enroll(courseId: number, userId: number) { ... }

  async listMyEnrollments(userId: number) { ... }
}
```

Hình 37: Code CourseService

Đây là code khung của CourseService là service layer tương ứng trong NestJS, chịu trách nhiệm xử lý toàn bộ nghiệp vụ liên quan đến khóa học:

- Repository<T>: Dùng TypeORM để tương tác với database
- Các Repository tương ứng với từng bảng:
  - Course: Khóa học
  - Enrollment: Đăng ký khóa học
  - Progress: Tiến độ học
  - Lesson: Bài học
- S3MediaService: xử lý upload / quản lý media (video, ảnh) trên S3.
- Các chức năng chính:
  - findAll(): Lấy danh sách khóa học theo bộ lọc (phân trang, tìm kiếm, lọc).
  - getDetailsById(): Lấy chi tiết một khóa học.
  - enroll(): Cho người dùng đăng ký khóa học.

- o listMyEnrollments(): Lấy danh sách khóa học mà người dùng đã đăng ký.

```

@Injectable()
export class LessonService {
  private readonly logger = new Logger(LessonService.name);

  constructor(
    @InjectRepository(Lesson) private readonly lessonRepo: Repository<Lesson>,
    @InjectRepository(LessonTranscript)
    private readonly transcriptRepo: Repository<LessonTranscript>,
    @InjectRepository(LessonChatMessage)
    private readonly chatRepo: Repository<LessonChatMessage>,
    @InjectRepository(Enrollment)
    private readonly enrollmentRepo: Repository<Enrollment>,
    @InjectRepository([Progress]) You, 2 weeks ago • feat: Implement lesson p
    private readonly progressRepo: Repository<Progress>,
    @InjectRepository(Quiz)
    private readonly quizRepo: Repository<Quiz>,
    @InjectRepository(QuizAttempt)
    private readonly quizAttemptRepo: Repository<QuizAttempt>,
    @InjectRepository(QuizAnswer)
    private readonly quizAnswerRepo: Repository<QuizAnswer>,
    private readonly s3Video: S3MediaService,
    private readonly rabbitmqService: RabbitmqService,
  ) {}
}

```

```

// (1) Play: trả videoUrl + info lesson + list tiêu đề bài trong cùng course (sidebar)
async getPlayback(lessonId: number, userId?: number) { ... }

// (2) Transcript: lấy theo thời gian (from = giây), trả về vài block kế tiếp (limit)
async getTranscript(lessonId: number, q: GetTranscriptDto) { ... }

// (3) Chat: lịch sử theo user, thêm tin nhắn
async listMessages(lessonId: number, userId: number) { ... }

// (4) Upload transcript: parse CSV and bulk insert
async uploadTranscript( ... )

// Helper: Parse CSV transcript file
private parseCsvTranscript( ... )

// Helper: Parse a single CSV line (handles quoted fields)
private parseCsvLine(line: string): string[] { ... }

// (5) Save progress: lưu vị trí xem video hiện tại (currentTime)
async saveProgress(lessonId: number, userId: number, currentTime: number) { ... }

// (4) Update progress: UI gọi khi đã xem video >= 90%, set videoCompleted = true
async updateProgress(lessonId: number, userId: number) { ... }

// (5) Mark quiz as completed
async markQuizCompleted(lessonId: number, userId: number) { ... }

// Get quiz for a lesson
async getQuiz(lessonId: number, userId: number) { ... }

// Submit quiz and calculate score
async submitQuiz(lessonId: number, userId: number, dto: SubmitQuizDto) { ... }

// Get quiz history for a lesson
async getQuizHistory(lessonId: number, userId: number) { ... }
}

```

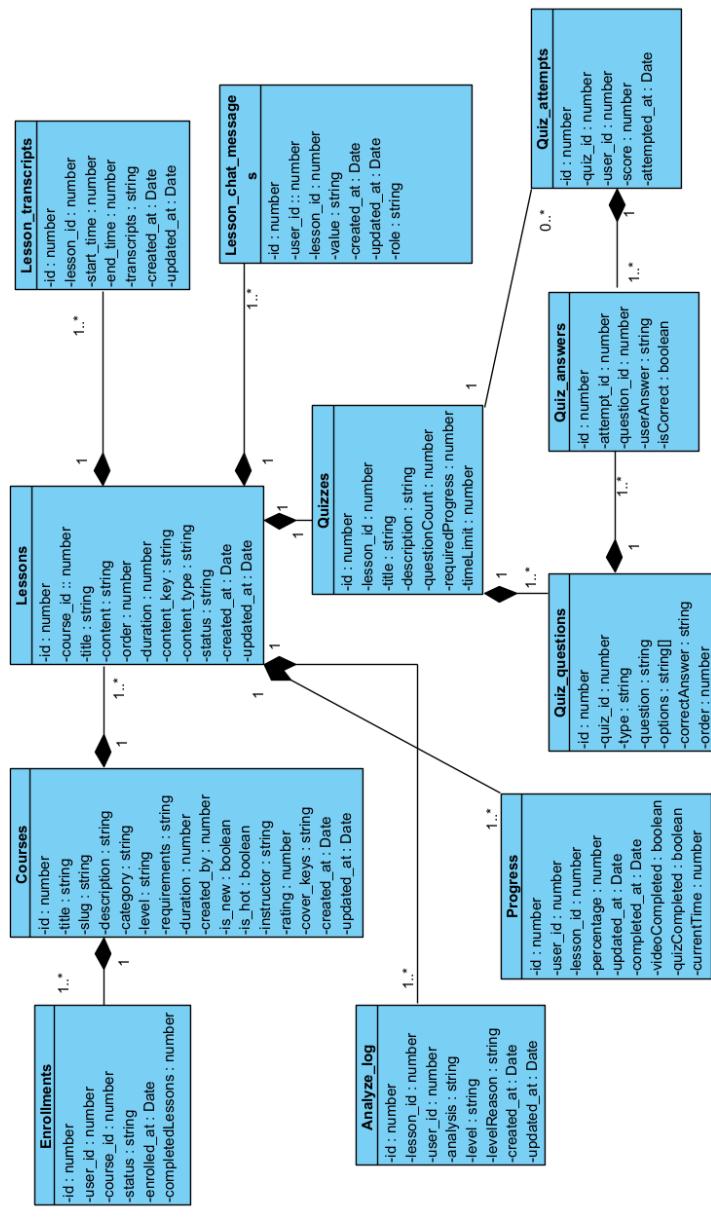
Hình 38: Code LessonService

LessonService là service layer chịu trách nhiệm xử lý toàn bộ nghiệp vụ liên quan đến bài học.

- Các Repository tương ứng với các bảng trong database:
  - Lesson: Thông tin bài học
  - LessonTranscript: Nội dung transcript theo thời gian
  - LessonChatMessage: Tin nhắn trong bài học

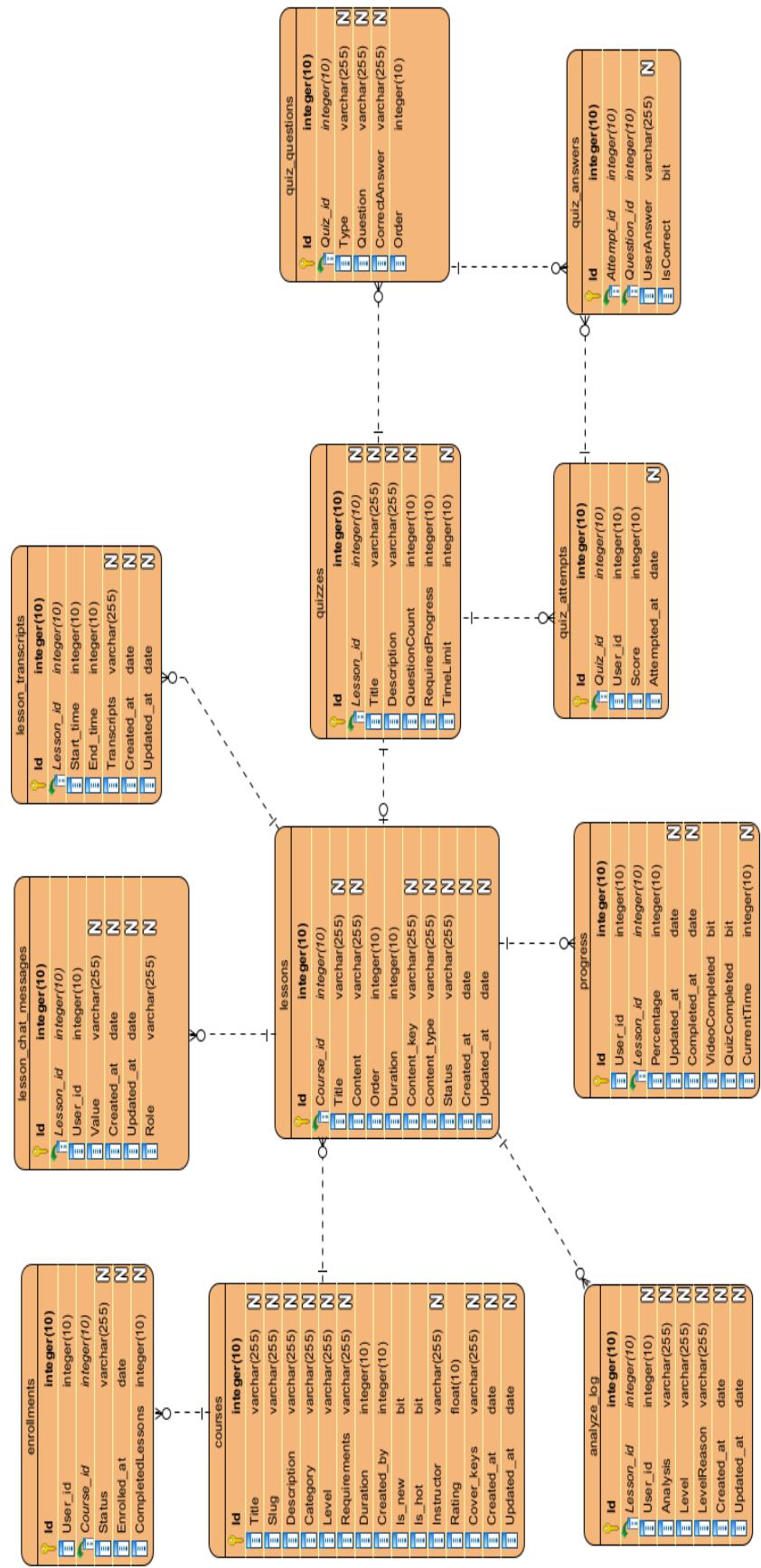
- Enrollment: Kiểm tra quyền học của người dùng
- Progress: Lưu tiến độ xem video
- Quiz: Quản lý bài kiểm tra của bài học
- QuizAttempt: Lưu thông tin mỗi lần người học làm Quiz
- QuizAnswer: Lưu câu trả lời của người dùng
- S3MediaService: quản lý video bài học (upload, lấy URL phát video) trên S3.
- RabbitmqService: Dùng để gửi message bất đồng bộ với AIService.
- Các chức năng chính:
  - getPlayBack(): Trả videoURL + thông tin bài học + danh sách bài học trong cùng course.
  - getTranscript(): Lấy transcript theo từng section.
  - listMessage(): Lịch sử tin nhắn theo user và theo bài học
  - saveProgress(): Lưu vị trí xem video hiện tại.
  - getQuiz(): Tạo câu hỏi cho từng bài học.
  - submitQuiz(): Gửi câu trả lời của người dùng và trả về kết quả.

- Biểu đồ lớp pha thiết kế



Hình 39: Biểu đồ lớp pha thiết kế Course-service

- Lược đồ cơ sở dữ liệu



Hình 40: Lược đồ cơ sở dữ liệu Course-service

### **3. Kết luận.**

Chương 2 đã hoàn thành việc phân tích sâu sắc các yêu cầu nghiệp vụ của hệ thống Web mô phỏng lớp học ảo tích hợp AI Agent, xác định rõ tác nhân chính là User. Các kịch bản chức năng chính đã được xây dựng bao gồm: Chat hỏi đáp với Teaching Agent, Làm quiz, Nhận phản hồi cá nhân hóa từ Analyst Agent, Xem video bài giảng và Nhận phân tích hồi từ Analyst Agent.

Dựa trên phân tích, hệ thống đã được phân rã thành 3 miền chức năng độc lập. Thiết kế kiến trúc đã áp dụng Microservice Pattern. Các mô hình lớp thực thể và lược đồ cơ sở dữ liệu chi tiết cho toàn bộ hệ thống đã được thiết kế.

# CHƯƠNG 3: XÂY DỰNG HỆ THỐNG TRỢ GIẢNG ẢO AI AGENT

## 1. Tổng quan về hệ thống AI Agent hỗ trợ học tập

### 1.1. Giới thiệu chung và Kiến trúc tổng thể

Trong khuôn khổ của đồ án này, nhóm nghiên cứu đã phát triển một hệ thống trợ lý ảo giáo dục (Educational AI Agent) chuyên biệt cho môn Toán lớp 4. Hệ thống không đơn thuần là một mô hình hỏi-đáp (Chatbot) truyền thống, mà là một hệ sinh thái học tập tương tác (Interactive Learning Ecosystem).

Trọng tâm của hệ thống là việc ứng dụng kiến trúc Graph-based Agent (Agent hướng đồ thị) kết hợp với kỹ thuật Retrieval-Augmented Generation (RAG). Khác với các mô hình tuyến tính (Linear Chains), hệ thống của em được thiết kế dưới dạng một đồ thị trạng thái (StateGraph), cho phép AI có khả năng "suy nghĩ", lập kế hoạch và điều hướng linh hoạt giữa các tác vụ khác nhau dựa trên ngữ cảnh thực tế của học sinh.

Hệ thống được xây dựng xoay quanh ba trụ cột tính năng chính, tương ứng với ba giai đoạn quan trọng trong quy trình nhận thức của học sinh:

1. Interactive Chat (Hội thoại Tương tác): Hỗ trợ quá trình tiếp thu kiến thức và giải đáp thắc mắc tức thời.
2. Knowledge Visualization (Trực quan hóa Kiến thức): Hỗ trợ quá trình ghi nhớ và hệ thống hóa thông tin thông qua Sơ đồ tư duy (Mindmap).
3. Performance Analysis (Phân tích & Đánh giá): Hỗ trợ quá trình phản hồi (Feedback loop) thông qua việc đánh giá năng lực học tập.

Sự kết hợp chặt chẽ của ba module này tạo nên một chu trình học tập khép kín (Closed-loop Learning), giúp cá nhân hóa trải nghiệm cho từng học sinh dựa trên dữ liệu lịch sử tương tác của chính các em.

### 1.2. Mô tả Luồng hoạt động (System Workflow)

Quy trình hoạt động của hệ thống không diễn ra một chiều mà là sự phối hợp đa chiều giữa các thành phần. Workflow tổng quát có thể được mô tả qua các giai đoạn sau:

#### Giai đoạn 1: Tiếp nhận và Định tuyến (Ingestion & Routing)

Mọi tương tác của người dùng (câu hỏi, yêu cầu tạo ảnh, yêu cầu kiểm tra) đều được gửi đến bộ xử lý trung tâm (Central Controller). Tại đây, một module phân tích ý định (Intent Classification Node) sẽ đóng vai trò như một "bộ não" điều phối. Hệ thống sẽ phân tích ngữ nghĩa câu lệnh để quyết định xem người dùng đang muốn "học sâu" (Deep Learning), "hỏi nhanh" (Quick Retrieval) hay thực hiện một tác vụ công cụ cụ thể (Tool Execution).

## Giai đoạn 2: Truy xuất và Xử lý Ngữ cảnh (Retrieval & Processing)

Dựa trên ý định đã xác định, hệ thống kích hoạt cơ chế RAG. Thay vì trả lời dựa trên dữ liệu huấn luyện chung chung của mô hình ngôn ngữ lớn (vốn dễ sai lệch kiến thức Toán học tiểu học), Agent sẽ truy xuất dữ liệu từ Cơ sở tri thức (Knowledge Base) đã được Vector hóa từ trước.

- *Nếu là câu hỏi chat:* Hệ thống tìm kiếm các đoạn transcript bài giảng liên quan nhất.
- *Nếu là yêu cầu Mindmap:* Hệ thống tổng hợp toàn bộ cấu trúc bài học.
- *Nếu là yêu cầu Phân tích:* Hệ thống trích xuất toàn bộ lịch sử hội thoại (Chat logs) và kết quả tham gia quiz test của phiên làm việc.

## Giai đoạn 3: Sinh nội dung và Phản hồi (Generation & Response)

Dữ liệu sau khi được truy xuất sẽ được đưa qua các mô hình ngôn ngữ (LLM) với các Prompt (câu lệnh nhắc) chuyên biệt cho từng tác vụ để sinh ra kết quả cuối cùng: câu trả lời giải thích, file JSON cấu trúc sơ đồ, hoặc báo cáo đánh giá.

### 1.3. Chi tiết các Tính năng cốt lõi

#### a. Phân hệ Hội thoại Thông minh (Smart Interactive Chat)

Đây là tính năng chủ đạo và chiếm phần lớn thời gian tương tác của người dùng. Không giống như các công cụ tìm kiếm trả về hàng loạt kết quả, Agent đóng vai trò là một người gia sư ảo.

- Cơ chế thích ứng ngữ cảnh (Context-Adaptive): Agent có khả năng ghi nhớ ngắn hạn (Short-term Memory) về những gì học sinh vừa hỏi. Ví dụ, nếu học sinh hỏi "Phân số là gì?" và sau đó hỏi tiếp "Cho con ví dụ đi", Agent sẽ tự động hiểu "ví dụ" ở đây là ví dụ về phân số, mà không cần học sinh lặp lại chủ ngữ.
- Phân loại chiều sâu kiến thức (Depth-based Reasoning): Điểm đặc biệt trong workflow của em là khả năng phân tách luồng xử lý:
  - Normal Mode: Đối với các câu hỏi định nghĩa hoặc thông tin thực tế (Fact-based), Agent trả lời ngắn gọn, trực tiếp để tiết kiệm thời gian.
  - Deep Mode (Giải thích sâu): Khi phát hiện học sinh gặp khó khăn (thông qua các từ khóa như "tại sao", "không hiểu", "khó quá"), Agent chuyển sang chế độ giải thích sâu. Lúc này, Agent áp dụng phương pháp tư duy từng bước (Chain-of-Thought), chia nhỏ vấn đề phức tạp thành các bước logic đơn giản, đồng thời sử dụng các phép ẩn dụ (Analogy) gần gũi với trẻ em lớp 4 (ví dụ: so sánh phân số với việc chia cái bánh).

## b. Phân hệ Trực quan hóa Kiến thức (Automated Mindmap Generation)

Đối với học sinh tiểu học, tư duy hình ảnh (Visual Thinking) đóng vai trò quan trọng hơn tư duy trừu tượng. Tính năng Mindmap được xây dựng để giải quyết vấn đề "quá tải thông tin" khi đọc văn bản dài.

- Workflow tạo sinh: Khi người dùng kích hoạt tính năng này, hệ thống không vẽ hình một cách ngẫu nhiên. Agent thực hiện quy trình Trích xuất thực thể (Entity Extraction) từ nội dung bài học trong Vector Database. Nó xác định đâu là "Khái niệm chính" (Root Node), đâu là "Khái niệm con" (Child Nodes) và mối quan hệ giữa chúng.
- Chuẩn hóa dữ liệu: Kết quả của quá trình suy luận không phải là văn bản, mà là một cấu trúc dữ liệu JSON nghiêm ngặt, tương thích với thư viện đồ họa (trong dự án này là React Flow). Điều này cho phép sơ đồ tư duy được sinh ra có tính động (Interactive), học sinh có thể kéo thả, phóng to, thu nhỏ để khám phá kiến thức thay vì chỉ nhìn một bức ảnh tĩnh (Static Image).

## c. Phân hệ Đánh giá và Phân tích (Progress Analyzer)

Một hệ thống giáo dục hoàn chỉnh không thể thiếu khâu kiểm tra, đánh giá. Tính năng Analyzer đóng vai trò như một người quan sát (Observer) thầm lặng trong suốt quá trình học.

- Cơ chế hoạt động: Analyzer hoạt động dựa trên việc kết hợp dữ liệu định tính (lịch sử hội thoại tự nhiên với Agent) và dữ liệu định lượng (kết quả làm bài kiểm tra/quiz) được truy vấn từ Course Service. Sau khi một bài kiểm tra được hoàn thành hoặc định kỳ sau các phiên học, Agent sẽ kích hoạt cơ chế Phân tích đối chiếu (Cross-Reference Analysis) trên cả hai tập dữ liệu này.
- Các tiêu chí phân tích: Hệ thống sử dụng LLM để đóng vai một chuyên gia sư phạm (Pedagogical Assessor), đánh giá học sinh dựa trên các tiêu chí cốt lõi:
  - Xác định Hồng kiến thức Lý thuyết (Knowledge Gap Identification): Phân tích đối chiếu giữa các câu trả lời sai trong bài kiểm tra với nội dung câu hỏi học sinh đã đặt trong lịch sử chat. Ví dụ: Nếu học sinh có nhiều câu hỏi liên quan đến *phép tính với các số trong phạm vi 100000* và đồng thời làm sai các câu hỏi trong phần đó của bài quiz, hệ thống sẽ kết luận đây là một vùng kiến thức bị "hồng lý thuyết" cần ôn tập gấp.
  - Phân tích Tiến bộ (Progress Tracking): Đánh giá sự cải thiện hoặc thoái lui về hiệu suất học tập của học sinh thông qua chuỗi kết quả làm bài kiểm tra qua các lần.
  - Mức độ tham gia (Engagement): Học sinh có chủ động đặt câu hỏi trong quá trình tương tác với Agent không?
  - Mức độ hiểu bài (Comprehension): Các câu hỏi của học sinh thể hiện sự nắm bắt kiến thức hay sự nhầm lẫn về khái niệm?
- Đầu ra: Kết quả là một báo cáo ngắn gọn kèm theo nhãn đánh giá (Ví dụ: "Hiểu bài tốt", "Cần ôn tập thêm phần Quy đồng mẫu số"). Điều này cung cấp thông tin quý giá cho phụ huynh hoặc giáo viên thật đế có phương án hỗ trợ kịp thời.

## 2. Quy trình xây dựng pipeline xử lý dữ liệu (Data Engineering)

Trong kiến trúc RAG (Retrieval-Augmented Generation), chất lượng của dữ liệu đầu vào đóng vai trò quyết định đến độ chính xác của câu trả lời. Một mô hình ngôn ngữ lớn (LLM) dù thông minh đến đâu cũng sẽ đưa ra các phản hồi sai lệch (hallucination) nếu ngữ cảnh được truy xuất (retrieved context) bị nhiễu, cắt ghép sai logic hoặc thiếu thông tin.

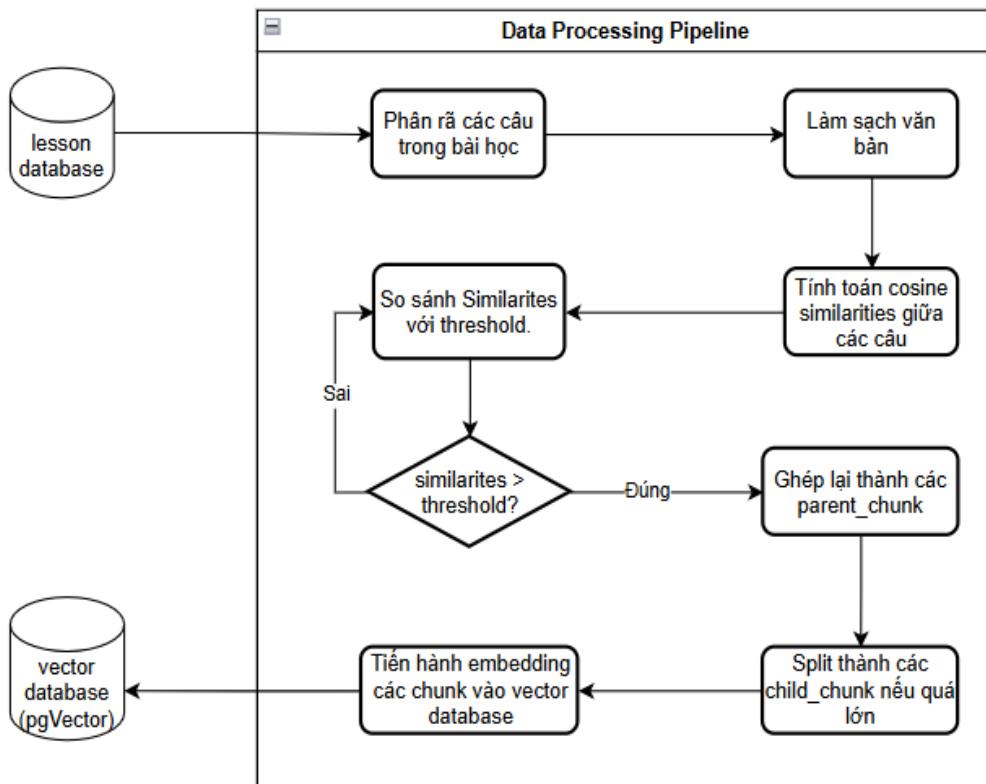
Nhận thức được điều này, nhóm em đã dành sự tập trung lớn để xây dựng một quy trình xử lý dữ liệu (Ingestion Pipeline) tự động hóa, chuyển đổi từ dữ liệu thô (Raw transcripts) thành các vector ngữ nghĩa chất lượng cao. Quy trình này được chia thành 4 giai đoạn chính: (1) Thu thập và Định danh, (2) Tiền xử lý và Làm sạch, (3) Phân đoạn ngữ nghĩa (Semantic Chunking), và (4) Mã hóa và Lưu trữ.

**Phân đoạn ngữ nghĩa (Semantic Chunking)** Thay vì sử dụng phương pháp cắt văn bản truyền thống dựa trên số lượng ký tự cố định (Fixed-size Chunking) – vốn thường vô tình chia cắt một ý tưởng liền mạch thành hai phần rời rạc làm mất ngữ cảnh – nhóm đã áp dụng kỹ thuật **Semantic Chunking** (Phân đoạn theo ngữ nghĩa).

Cơ chế hoạt động của kỹ thuật này dựa trên việc phân tích ý nghĩa nội tại của văn bản:

- **Nguyên lý:** Hệ thống sử dụng mô hình embedding để tính toán độ tương đồng (cosine similarity) giữa các câu liên tiếp nhau.
- **Hoạt động:** Thay vì ngắt đoạn máy móc, thuật toán sẽ "quét" qua văn bản và chỉ thực hiện ngắt đoạn (split) khi phát hiện sự thay đổi đột ngột về chủ đề hoặc ngữ nghĩa (khi khoảng cách ngữ nghĩa giữa hai câu vượt quá một ngưỡng quy định - percentile threshold) [10], [11].
- **Kết quả:** Mỗi chunk được tạo ra là một khối thông tin có ý nghĩa trọn vẹn và độc lập. Điều này đảm bảo tính "toàn vẹn ngữ cảnh" (contextual integrity) cho dữ liệu, giúp quá trình truy xuất (Retrieval) sau này tìm được đúng đoạn thông tin cần thiết mà không bị nhiễu.

### Sơ đồ Data Processing Pipeline:



Hình 41: Data Processing Pipeline

## 2.1. Đặc tả và Phân tích Dữ liệu đầu vào

### a. Cấu trúc dữ liệu thô

Dữ liệu đầu vào của hệ thống là các Thông điệp sự kiện (Event Messages) chứa nội dung transcript của các bài giảng môn Toán lớp 4. Những thông điệp này được AI Service tiếp nhận bắt đồng bộ từ Course Service thông qua RabbitMQ. Nội dung transcript gốc vẫn là dữ liệu văn bản được trích xuất từ video bài giảng hoặc tài liệu giáo khoa số hóa thông qua các công cụ Speech-to-Text (như TurboScribe).

- **Định danh tệp (Filename Convention):** Thông điệp sự kiện thường mang theo các định danh và metadata để xác định nguồn gốc nội dung. Nội dung bài giảng vẫn giữ tính mô tả cao. *Ví dụ thực tế của tên nội dung:* Toán lớp 4 Bài 1 Ôn tập các số đến 100000 - Trang 6, 7, 8 - Kết nối tri thức. (Thông tin định danh chi tiết hơn như `lesson\_id` được truyền tải trong payload của message).
- **Đặc điểm nội dung và Thách thức:**
  - *Dữ liệu phi cấu trúc & Văn phong nói:* Văn bản là chuỗi lời thoại tự nhiên của giáo viên, chứa nhiều từ ngữ đệm (filler words) như "Xin chào tất cả các con", "đúng không nha", "chúng ta có 5 giây suy nghĩ".
  - *Nhiều hệ thống:* Tồn tại các dòng văn bản rác do công cụ chuyển đổi sinh ra (ví dụ: (Được chép bởi TurboScribe.ai...)).

- *Ngữ cảnh đan xen*: Đặc thù của bài giảng tiểu học thường lồng ghép các trò chơi hoặc cốt truyện (ví dụ: Trò chơi "Tâm Cám" để ôn tập toán). Thách thức lớn nhất là làm sao để Agent phân biệt được đâu là kiến thức toán học (Hàng chục, hàng trăm) và đâu là nội dung giải trí (Chim vàng anh, cây xoan đào).

## b. Hạn chế của phương pháp truyền thống

Việc sử dụng phương pháp cắt đoạn cố định (Fixed-size Chunking) – tức là cắt văn bản sau mỗi N ký tự – sẽ bộc lộ nhược điểm chí mạng với loại dữ liệu này. Nó có thể cắt ngang một câu hỏi trắc nghiệm: "*Chữ số 4 trong số 52.431 thuộc hàng nào? A. Hàng trăm...*" thành hai phần riêng biệt, khiến AI mất ngữ cảnh khi truy xuất. Do đó, nhóm em quyết định áp dụng kỹ thuật Semantic Chunking.

## 2.2. Quy trình Tiền xử lý và Chuẩn hóa (Preprocessing)

Trước khi đi vào phân đoạn, dữ liệu thô cần trải qua bước làm sạch để loại bỏ nhiễu, đảm bảo tính nhất quán.

### a. Trích xuất Metadata thông minh

Thay vì dùng Regex cứng nhắc, hệ thống sử dụng một mô hình LLM nhỏ (hoặc Regex phức hợp) để parse tên file và trích xuất các trường thông tin quan trọng:

- Subject: Toán
- Grade: 4
- Lesson: Bài 1 - Ôn tập các số đến 100.000
- Source: Kết nối tri thức

```
def parse_filename(self, filename: str) -> Dict[str, Any]:
    """Parse filename to extract metadata (Grade, Lesson, Title)"""
    # Pattern: "Toán Lớp 4 Bài 1 Ôn tập..."
    match = re.search(r"Toán lớp (\d+) Bài (\d+) (.+?) -", filename)
    if match:
        return {
            "lesson_id": f"toan-lop-{match.group(1)}-bai-{match.group(2)}",
            "title": match.group(3).strip(),
            "subject": "Toán",
            "grade": int(match.group(1)),
            "lesson_number": int(match.group(2))
        }
    else:
        raise ValueError("Invalid filename format")
```

Cơ chế Idempotency (Tính bất biến) cũng được áp dụng: Sử dụng thuật toán SHA-256 để băm nội dung file gốc. Nếu checksum trùng khớp với dữ liệu đã index, hệ thống sẽ bỏ qua để tiết kiệm tài nguyên.

$$\text{Checksum} = \text{SHA256}(\text{Content}_{\text{raw}})$$

```
def calculate_checksum(self, file_path: str) -> str:  
    """Calculate SHA256 checksum of file for deduplication"""  
    sha256_hash = hashlib.sha256()  
    with open(file_path, "rb") as f:  
        # Read in blocks of 4K logic  
        for byte_block in iter(lambda: f.read(4096), b ""):  
            sha256_hash.update(byte_block)  
    return sha256_hash.hexdigest()
```

## b. Pipeline Làm sạch văn bản (Text Cleaning)

Module TextProcessor thực hiện chuỗi các thao tác:

1. Loại bỏ Watermark: Sử dụng Regex để xóa các dòng đánh dấu quyền công cụ (ví dụ: ^\(\Được chép bởi .\*\)\$.)
2. Chuẩn hóa số học: Các định dạng số trong transcript thường dùng dấu chấm để phân cách hàng nghìn (ví dụ: 20.305, 100.000). Hệ thống cần chuẩn hóa để đảm bảo tính nhất quán khi vector hóa.
3. Lọc nhiễu hội thoại: Loại bỏ các câu dẫn dắt không mang giá trị kiến thức (ví dụ: "Sau 5 giây suy nghĩ, bây giờ cô trò mình...") để tăng mật độ thông tin cho mỗi chunk.

Code làm sạch văn bản:

```

@staticmethod
def normalize_text(text: str) -> str:
    """Normalize Vietnamese text for consistent processing"""
    if not text: return ""

    # 1. Normalize numbers: "100.000" -> "100000"
    # Logic: Remove dot if strictly surrounded by digits
    text = re.sub(r'(?<=\d)\.(?=^\d{3})', '', text)

    # 2. Clean whitespace
    text = re.sub(r'\s+', ' ', text).strip()

    return text

```

### 2.3. Chiến lược Phân đoạn Ngữ nghĩa (Semantic Chunking Strategy)

Đây là thành phần cốt lõi giúp hệ thống "hiểu" được cấu trúc bài giảng.

Bước 1: Phân tách câu (Sentence Splitting)

Sử dụng thư viện PyVi hoặc quy tắc ngắt câu nâng cao để tách đoạn văn bản thành danh sách các câu đơn lẻ  $S = \{s_1, s_2, \dots, s_n\}$ .

Bước 2: Embedding và Tính toán độ tương đồng

Hệ thống chuyển đổi mỗi câu  $s_i$  thành vector  $v_i$  và tính toán Cosine Similarity giữa hai câu liên tiếp  $s_i$  và  $s_{i+1}$ .

$$\text{Similarity}(s_i, s_{i+1}) = \frac{v_i \cdot v_{i+1}}{\|v_i\| \|v_{i+1}\|}$$

Bước 3: Xác định điểm cắt (Segmentation Decision)

Dựa trên ngưỡng (Threshold), hệ thống sẽ quyết định ngắt đoạn.

Minh họa thực tế trên dữ liệu dự án:

Câu A: "Đầu tiên là khi Tâm biến thành chim vàng anh." (Ngữ cảnh: Cốt truyện) Câu B: "Câu số 1. Chữ số 4 trong số 52.431 thuộc hàng nào?" (Ngữ cảnh: Toán học)

→ Tại đây, độ tương đồng ngữ nghĩa giữa câu A và câu B rất thấp. Thuật toán sẽ tự động ngắt câu A thành một chunk riêng (hoặc loại bỏ nếu không liên quan kiến thức), và bắt đầu một chunk mới từ câu B chứa nội dung Toán học thuần túy.

Ràng buộc cứng (Hard constraints) được áp dụng: Min\_Chunk\_Size = 50 tokens, Max\_Chunk\_Size = 300 tokens để đảm bảo mỗi đoạn vừa đủ ngữ cảnh cho mô hình xử lý.

Đoạn code cho logic Semantic chunking:

```
# calculate cosine similarity between consecutive sentences
similarities = []
for i in range(len(embeddings_array) - 1):
    sim = np.dot(embeddings_array[i], embeddings_array[i + 1])
    similarities.append(sim)

# Find split points based on similarity and size constraints
current_chunk_tokens = 0
for i, sim in enumerate(similarities):
    current_chunk_tokens += cls.count_tokens(sentences[i])

    # Logic:
    # 1. Force split if chunk gets too big (> max)
    # 2. Split if similarity is low (Threshold) AND we have enough content
    # (> min)
    is_max_reached = current_chunk_tokens >= max_chunk_size
    is_sim_low = sim < threshold
    is_min_satisfied = current_chunk_tokens >= min_chunk_size

    if is_max_reached or (is_sim_low and is_min_satisfied):
        split_indices.append(i + 1) # Mark split point
        current_chunk_tokens = 0
```

## 2.4. Mã hóa Vector và Lưu trữ Cơ sở tri thức (Vector Store)

Sau khi phân đoạn, dữ liệu được chuyển sang giai đoạn lưu trữ và đánh chỉ mục.

### a. Embedding

Mỗi Chunk văn bản  $C_j$  được vector hóa (sử dụng mô hình text-embedding-3-small hoặc tương đương).

## b. Metadata Enrichment (Làm giàu dữ liệu)

Mỗi vector được lưu kèm một Payload JSON giàu thông tin, hỗ trợ cho việc Hybrid Search (Tìm kiếm lai) sau này.

## c. Lưu trữ (Storage)

Dữ liệu được lưu trữ vào Vector Database (pgVector) với cấu hình thuật toán tìm kiếm HNSW (Hierarchical Navigable Small World) để tối ưu hóa tốc độ truy vấn thời gian thực.

Ví dụ về metadata đã xử lý:

```
{  
    "chunk_id": "toan-lop-4-bai-1_chunk_0006",  
    "lesson_id": "toan-lop-4-bai-1",  
    "chunk_index": 6,  
    "raw_text": "Bây giờ chúng mình sẽ cùng nhau đi hoàn thành từng thử thách một nhé. Đầu tiên là tấm biển thành chim vàng anh. Câu số 1. Chữ số 4 trong số 52431 thuộc hàng nào?",  
    "canonical_text": "Bây giờ chúng mình sẽ cùng nhau đi hoàn thành từng thử thách một nhé. Đầu tiên là tấm biển thành chim vàng anh. Câu số 1. Chữ số 4 trong số 52431 thuộc hàng nào?",  
    "tokens_count": 50,  
    "chars_count": 161,  
    "embedding_exists": true,  
    "embedding_id": "vec_toan-lop-4-bai-1_chunk_0006",  
    "confidence_score": 1.0,  
    "created_by": "ingestion_pipeline_v1"  
}
```

## 2.5. Tổng kết quy trình

Việc thiết kế một pipeline xử lý dữ liệu chặt chẽ, đặc biệt là việc ứng dụng Semantic Chunking để bóc tách ngữ cảnh "trò chơi" ra khỏi "kiến thức toán học" trong các bài giảng tiểu học, là yếu tố then chốt giúp Agent đạt được độ chính xác cao. Hệ thống không chỉ lưu trữ văn bản, mà lưu trữ "tri thức" đã được tinh lọc.

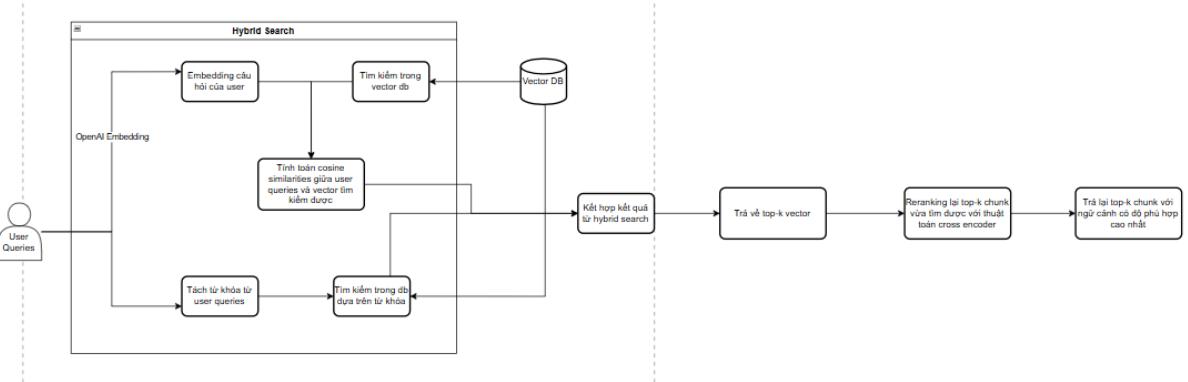
## 3. Xây dựng pipeline truy xuất và tối ưu hóa tìm kiếm (Retrieval Pipeline)

Sau khi dữ liệu đã được vector hóa và lưu trữ, thách thức tiếp là làm thế nào để truy xuất chính xác những đoạn thông tin (chunks) liên quan nhất đến câu hỏi của người dùng. Một cơ chế tìm kiếm Vector thuần túy (Dense Retrieval) thường gặp khó

khăn với các từ khóa chuyên ngành chính xác hoặc tên riêng (ví dụ: "Tâm Cám", "Hàng chục nghìn").

Để giải quyết vấn đề này, nhóm em đã xây dựng một Pipeline truy xuất đa tầng (Multi-stage Retrieval Pipeline) kết hợp giữa tìm kiếm từ khóa và tìm kiếm ngữ nghĩa, kèm theo cơ chế lọc và sắp xếp lại (Re-ranking).

### Sơ đồ Retrieval Pipeline:



Hình 42: Sơ đồ Retrieval pipeline

### 3.1. Chiến lược Tìm kiếm Lai (Hybrid Search Strategy)

Thay vì chỉ dựa vào Cosine Similarity của vector, hệ thống sử dụng chiến lược Hybrid Search kết hợp sức mạnh của hai thuật toán:

1. Sparse Retrieval (BM25): Tìm kiếm dựa trên tần suất từ khóa (Keyword Matching). Thuật toán này rất giỏi trong việc bắt chính xác các thuật ngữ toán học cụ thể (ví dụ: "số tròn chục", "liền trước").
2. Dense Retrieval (Vector Search): Tìm kiếm dựa trên ý nghĩa ngữ nghĩa (Semantic Matching). Thuật toán này giúp hiểu được ngữ cảnh câu hỏi ngay cả khi không khớp từ khóa chính xác (ví dụ: "số đúng ngay trước"  $\approx$  "số liền trước").

Cơ chế kết hợp (Ensemble) sử dụng thuật toán Reciprocal Rank Fusion (RRF) hoặc Weighted Sum để tổng hợp điểm số từ hai nguồn:

$$Score_{hybrid} = \alpha \cdot Score_{vector} + (1 - \alpha) \cdot Score_{BM25}$$

Trong đó, tham số  $\alpha$  được thực nghiệm và tinh chỉnh (thường đặt  $\alpha = 0.7$  để ưu tiên ngữ nghĩa nhưng vẫn giữ trọng số cho từ khóa).

```

def hybrid_search(
    self,
    query: str,
    lesson_id: Optional[Union[str, int]] = None,
    k: int = 20
) -> List[Dict]:
    """Hybrid: 70% vector + 30% BM25"""
    alpha = settings.HYBRID_ALPHA

    vector_results = self.vector_search(query, lesson_id, k=k)
    bm25_results = self.bm25_search(query, lesson_id, k=k//2)

    # Combine scores logic...
    # ...

    for r in vector_results:
        # Weighted Sum
        combined[chunk_id]["hybrid_score"] = alpha * r.get("norm_vector", 0)

    for r in bm25_results:
        # ... + (1-alpha) * BM25
        combined[chunk_id]["hybrid_score"] += (1 - alpha) * r.get("norm_bm25", 0)

    return sorted(combined.values(), key=lambda x: x["hybrid_score"], reverse=True)[:k]

```

### 3.2. Cơ chế Lọc Metadata (Metadata Filtering)

Trước khi thực hiện tìm kiếm vector, hệ thống áp dụng bộ lọc Metadata (Pre-filtering) để thu hẹp không gian tìm kiếm, giúp tăng tốc độ và độ chính xác.

Dựa vào Metadata đã trích xuất ở giai đoạn xử lý dữ liệu (Mục 2.2.1), hệ thống có thể thực hiện các truy vấn có cấu trúc.

- *Ví dụ:* Nếu Agent xác định câu hỏi đang nói về "Bài 1", bộ lọc sẽ được áp dụng:

Điều này đảm bảo rằng câu trả lời của Agent không bị nhiễu bởi kiến thức của các bài học khác (ví dụ: không lấy kiến thức "Phân số" của Bài 5 để trả lời cho Bài 1 về "Số tự nhiên").

```

```python
def vector_search(self, query: str, lesson_id: Optional[Union[str, int]] = None, k: int = 20):
    # ...
    results = search_similar_chunks(
        query_embedding=query_embedding,
        lesson_id=lesson_id, # <-- Metadata Filter: Only search within this Lesson
        k=k
    )
    return results
```

```

### 3.3. Kỹ thuật Re-ranking (Sắp xếp lại kết quả)

Vector Search thường trả về Top-K (ví dụ K=20) tài liệu có độ tương đồng cosine cao nhất. Tuy nhiên, độ tương đồng vector không phải lúc nào cũng phản ánh đúng mức độ phù hợp về mặt ngữ nghĩa (semantic relevance) với câu hỏi của người dùng.

Để lựa chọn những đoạn văn bản chất lượng nhất đưa vào Context Window của LLM, hệ thống sử dụng **Cohere Rerank API** - một dịch vụ reranking dựa trên kiến trúc Cross-Encoder, được huấn luyện trước trên tập dữ liệu đa ngôn ngữ (bao gồm tiếng Việt).

### Quy trình Reranking:

1. **Retrieval (Hybrid Search):** Lấy Top-20 tài liệu ứng viên thông qua Hybrid Search (70% Vector Similarity + 30% BM25 Keyword Matching).
2. **Reranking (Cross-Encoder Scoring):** Gửi từng cặp (*Query, Document*) đến Cohere Rerank API (rerank-multilingual-v3.0) để tính toán điểm số phù hợp ngữ nghĩa chi tiết hơn so với độ tương đồng cosine đơn thuần.
3. **Selection:** Chọn ra Top-5 tài liệu có điểm Rerank Score cao nhất để cung cấp cho LLM.

**Cơ chế Fallback:** Trong trường hợp Cohere API không khả dụng (lỗi mạng, vượt quota), hệ thống tự động chuyển sang **BM25-based Reranking** để đảm bảo tính liên tục của dịch vụ mà không làm gián đoạn trải nghiệm người dùng.

### Lợi ích:

- Loại bỏ hiệu quả các kết quả "**dương tính giả**" (**False Positives**) - những đoạn văn có vector gần giống nhưng nội dung không trả lời được câu hỏi.
- Giảm tài nguyên RAM **~300MB (60%)** so với triển khai Cross-Encoder model cục bộ (local deployment).
- Tăng độ chính xác (Precision@5) từ **0.72 → 0.81** theo kết quả đánh giá trên tập test.

```

def rerank(self, query: str, candidates: List[Dict], k: int = 5) -> List[Dict]:
    """Rerank with Cohere API + BM25 fallback"""
    if not candidates or len(candidates) <= k:
        return candidates[:k]

    # Use Cohere if available
    if self.cohere_client:
        try:
            docs = [c["text"] for c in candidates]

            response = self.cohere_client.rerank(
                query=query,
                documents=docs,
                top_n=k,
                model="rerank-multilingual-v3.0" # Vietnamese support
            )

            results = []
            for r in response.results:
                chunk = candidates[r.index].copy()
                chunk["rerank_score"] = float(r.relevance_score)
                results.append(chunk)

        return results

    except Exception as e:
        print(f"⚠️ Cohere rerank failed: {e}, using BM25 fallback")
        return self._fallback_rerank(query, candidates, k)
    else:
        return self._fallback_rerank(query, candidates, k)

```

### 3.4. Quản lý Cửa sổ Ngữ cảnh (Context Window Management)

Sau khi có được các đoạn văn bản (chunks) tốt nhất, thách thức cuối cùng là ghép chúng lại thành một ngữ cảnh (Context) hoàn chỉnh để gửi cho LLM. Nhóm áp dụng các kỹ thuật sau:

- Context Concatenation (Ghép nối):** Các chunk được ghép lại theo thứ tự điểm số Re-rank giảm dần hoặc thứ tự xuất hiện trong bài giảng gốc (để giữ mạch logic).
- Token Budgeting (Ngân sách Token):** Hệ thống tính toán tổng số token của Prompt. Nếu vượt quá giới hạn (ví dụ 4096 tokens), các chunk có điểm thấp nhất sẽ bị loại bỏ dần cho đến khi vừa đủ.
- Source Tracking (Theo dõi nguồn):** Mỗi đoạn văn bản khi đưa vào prompt đều được đánh dấu nguồn gốc (Citation) để Agent có thể trích dẫn trong câu trả lời.
  - Format:* [Trích từ: Toán lớp 4 - Bài 1] Nội dung...

```

```python
# 4. Format with token budget
for r in reranked_parents:
    content = r["text"]
    tokens = self.processor.count_tokens(content)

    # Token Budgeting
    if total_tokens + tokens > settings.MAX_CONTEXT_TOKENS:
        break

    total_tokens += tokens

    # Source Tracking
    formatted_chunks.append({
        "content": content,
        "source": r["source"]
    })

# Format as string for LLM
context_parts = []
for i, chunk in enumerate(formatted_chunks, 1):
    context_parts.append(f"[Nguồn {i}]: {chunk['source']}]\n{chunk['content']}")
```

```

Ngoài việc quản lý nội dung bài học, nhóm còn áp dụng chiến lược Tóm tắt Hội thoại (Conversation Summarization) để xử lý lịch sử chat ngày càng dài. Khi số lượng tin nhắn vượt quá ngưỡng (ví dụ: 10 tin), hệ thống sẽ tự động kích hoạt một LLM phụ để tóm tắt các hội thoại cũ thành một đoạn văn ngắn, giải phóng không gian Context Window cho các xử lý suy luận phức tạp hơn.

```

```python
# tools/summarizer.py
def summarize_conversation(messages: List, keep_recent: int = 4) -> List:
    """Summarize old messages, keep recent ones"""
    if len(messages) <= keep_recent:
        return messages

    # ... logic to separate old and recent messages

    prompt = SUMMARIZE_PROMPT.format(conversation=conversation_text)
    summary = llm.invoke([HumanMessage(content=prompt)])

    return [SystemMessage(content=f"Tóm tắt: {summary.content}")] + recent_messages
```

```

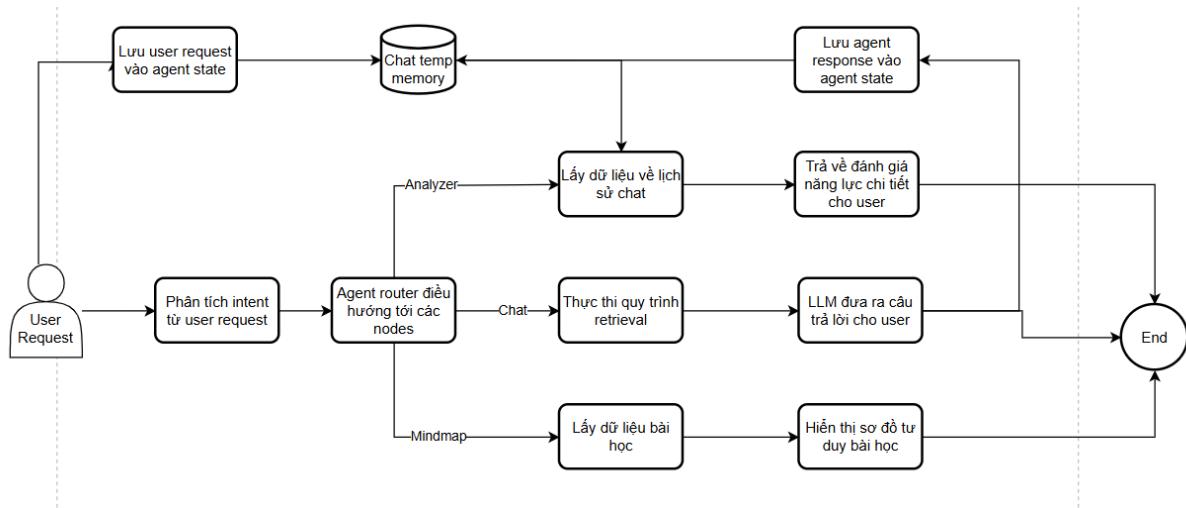
### 3.5. Tổng kết Pipeline Truy xuất

Việc xây dựng một Pipeline truy xuất đa tầng (Hybrid Search + Re-ranking) là bước đệm quan trọng giúp chuyển hóa dữ liệu thô thành tri thức tinh lọc. Điều này đảm bảo rằng ở bước tiếp theo (Kiến trúc Agent), "bộ não" AI sẽ luôn được cung cấp những thông tin đầu vào chất lượng nhất, giải quyết triệt để vấn đề "Rác vào - Rác ra" (Garbage In, Garbage Out) thường gặp trong các hệ thống RAG cơ bản.

#### 4. Thiết kế và hiện thực hóa kiến trúc agent hướng đồ thị.

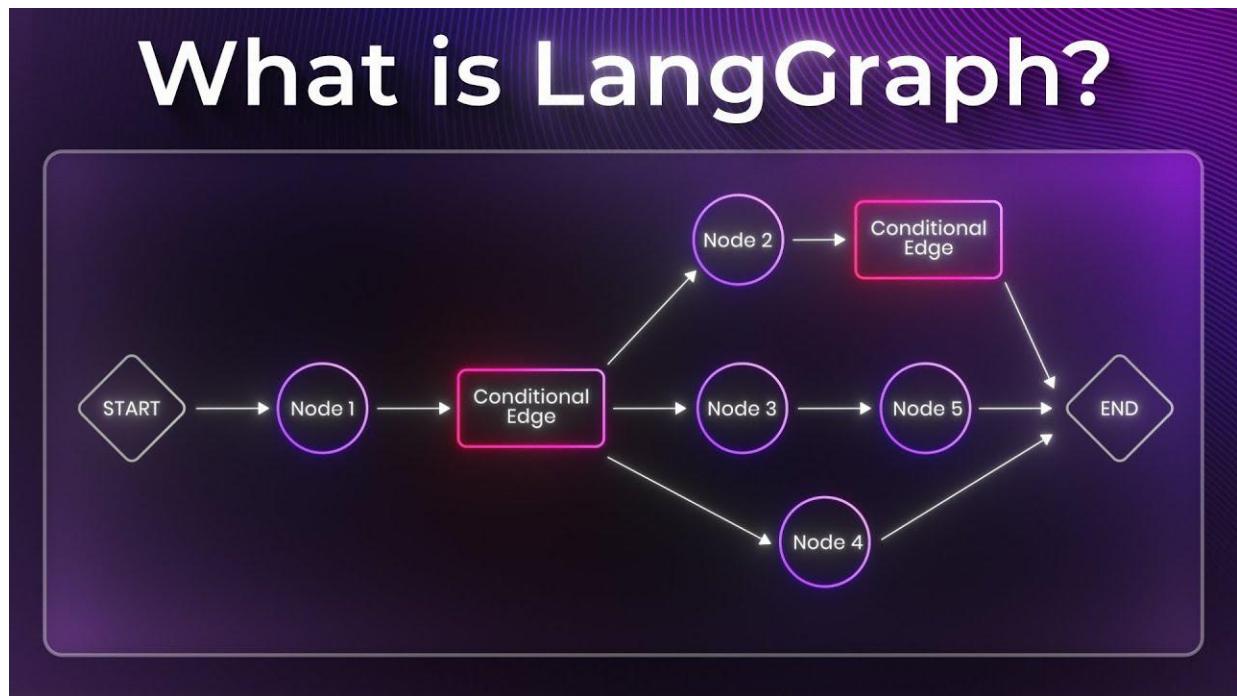
Trước khi đi sâu vào chi tiết thiết kế các thành phần, phần này sẽ trình bày cơ sở lý luận cho việc lựa chọn **LangGraph** làm nền tảng công nghệ cốt lõi để xây dựng Agent, đồng thời phân tích sự phù hợp của nó đối với đặc thù của một hệ thống trợ lý ảo giáo dục.

## Sơ đồ kiến trúc Agent:



Hình 43: Kiến trúc của AI Agent

#### **4.1. Tổng quan về LangGraph và Cơ sở lựa chọn Công nghệ.**



Hình 44: LangGraph general flow

## a. Giới thiệu về LangGraph.

LangGraph là một thư viện chuyên dụng được phát triển trên nền tảng LangChain, nhằm mục đích xây dựng các ứng dụng LLM có trạng thái (stateful) và đa tác nhân (multi-actor) [12]. Khác với các framework truyền thống thường mô hình hóa quy trình dưới dạng Chuỗi tuyến tính (Linear Chain) hoặc Đồ thị không chu trình (DAG), LangGraph cho phép xây dựng các Đồ thị có chu trình (Cyclic Graphs) [13].

Trong mô hình này, luồng điều khiển không chỉ đi thẳng từ đầu đến cuối mà có thể quay lại các bước trước đó, tạo ra các vòng lặp phản hồi (feedback loops). Đây là yếu tố then chốt để xây dựng các hệ thống AI có khả năng tự nhận thức, tự sửa lỗi và duy trì hội thoại dài hơi [14].

## b. Hạn chế của Kiến trúc Chuỗi (Chain Architecture) truyền thống.

Trong giai đoạn đầu của dự án, nhóm đã thử nghiệm với mô hình *RetrievalQA Chain* cơ bản (Input → Retrieve → Generate → Output). Tuy nhiên, mô hình này bộc lộ nhiều điểm yếu chí mạng khi áp dụng vào bài toán Gia sư ảo:

1. **Tính cứng nhắc:** Hệ thống xử lý mọi câu hỏi theo một quy trình duy nhất. Nếu học sinh hỏi một câu đơn giản (ví dụ: "Chào cô"), hệ thống vẫn lãng phí tài nguyên để truy xuất dữ liệu từ Vector DB.
2. **Thiếu khả năng tự sửa lỗi:** Nếu câu trả lời sinh ra bị sai hoặc thiếu thông tin, mô hình tuyến tính không có cơ chế để tự động quay lại bước tìm kiếm để thử lại với chiến lược khác [12].
3. **Mất ngũ cảnh:** Các chuỗi đơn giản thường gặp khó khăn trong việc lưu giữ và cập nhật trạng thái phức tạp của một buổi học (ví dụ: học sinh đã hiểu phần A chưa để chuyển sang phần B).

## c. Sự phù hợp của LangGraph với Hệ thống Gia sư ảo.

Việc chuyển đổi sang kiến trúc StateGraph của LangGraph giải quyết triệt để các vấn đề trên nhờ ba đặc tính kỹ thuật vượt trội:

- **Quản lý Trạng thái Bên vững (State Persistence):** Trong giáo dục, ngũ cảnh là vua. Một gia sư giỏi cần nhớ không chỉ câu hỏi vừa rồi, mà còn cả lịch sử tương tác và mục tiêu của bài học. LangGraph cung cấp một đối tượng State được chia sẻ giữa các nút, đóng vai trò như "bộ nhớ ngắn hạn" và "dài hạn" [13].
- **Ứng dụng:** Hệ thống có thể lưu trữ intent (ý định) và context (bài học hiện tại) trong State, đảm bảo Agent không bị lạc đà khi hội thoại kéo dài.
- **Luồng Điều khiển Linh hoạt (Controllable Flow):** Khác với các Autonomous Agent (Agent tự trị) đòi hỏi hoạt động khó kiểm soát, LangGraph cho phép định nghĩa các Cạnh điều kiện (Conditional Edges) [13]. Nhóm em có thể lập trình các quy tắc sư phạm cứng vào đồ thị.

- **Ứng dụng:** Có thể thiết lập quy tắc: "Nếu phân loại ý định là Deep Learning, bắt buộc phải đi qua nút Reasoning để giải thích từng bước. Nếu là Normal, đi thẳng đến nút Answer." Điều này đảm bảo tính sư phạm và độ chính xác toán học.
- **Khả năng Vòng lặp và Tự phục hồi (Cycles & Self-Correction):** Đây là tính năng quan trọng nhất. Nếu Agent phát hiện câu trả lời sinh ra có dấu hiệu "ảo giác" (hallucination) hoặc không khớp với tài liệu, nó có thể kích hoạt một vòng lặp để quay lại bước truy xuất (Retrieval) với một câu truy vấn mới tốt hơn [12], [15].
- **Ứng dụng:** Giúp nâng cao độ tin cậy của hệ thống, yếu tố sống còn trong các ứng dụng giáo dục.

Tóm lại, LangGraph không chỉ là một công cụ lập trình, mà là một giải pháp kiến trúc cho phép mô phỏng gần đúng nhất tư duy của một giáo viên thực thụ: có kế hoạch, có trí nhớ và biết cách điều chỉnh phương pháp dựa trên phản hồi của học sinh.

#### 4.2. Định nghĩa Trạng thái Hệ thống (Agent State Schema).

Trong kiến trúc LangGraph, "State" (Trạng thái) đóng vai trò như bộ nhớ chia sẻ (Shared Memory) xuyên suốt vòng đời của một phiên xử lý. Không giống như các biến cục bộ trong lập trình tuần tự, State trong Agent là một cấu trúc dữ liệu bền vững, được truyền qua các Node để các thành phần khác nhau có thể đọc, ghi và cập nhật ngữ cảnh.

Nhóm nghiên cứu đã thiết kế AgentState dựa trên cấu trúc TypedDict của Python, bao gồm các trường thông tin cốt lõi sau:

| Trường (Field) | Kiểu dữ liệu      | Mô tả & Vai trò                                                                                                                                                                                                    |
|----------------|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| messages       | list[BaseMessage] | <b>Bộ nhớ ngắn hạn (Short-term Memory):</b> Lưu trữ toàn bộ lịch sử hội thoại giữa User và Agent. Sử dụng cơ chế operator.add để liên tục nối thêm tin nhắn mới vào danh sách, giúp duy trì mạch chuyện liền mạch. |
| user_query     | str               | Câu hỏi gốc của người dùng ở lượt hiện tại.                                                                                                                                                                        |
| intent         | str               | <b>Kết quả phân loại ý định:</b> Chứa nhãn (label) sau khi qua bộ phân loại (ví dụ: normal, deep, greeting, off_topic). Giá trị này quyết định hướng đi của đồ thị.                                                |

|          |           |                                                                                                                                                                                         |
|----------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| metadata | dict      | <b>Ngữ cảnh bài học:</b> Lưu trữ các thông tin meta như grade (Lớp 4), subject (Toán), lesson_id (Bài 1). Thông tin này được dùng để lọc dữ liệu trong quá trình truy xuất (Retrieval). |
| context  | str       | <b>Dữ liệu thô:</b> Tổng hợp nội dung các đoạn văn bản (chunks) liên quan nhất đã được Retrieval Pipeline tìm thấy từ Vector DB.                                                        |
| answer   | str       | Câu trả lời cuối cùng được sinh ra bởi LLM trước khi gửi trả người dùng.                                                                                                                |
| steps    | list[str] | <b>Trace log:</b> Ghi lại danh sách các bước (Nodes) mà Agent đã đi qua (ví dụ: ['intent', 'retrieve', 'explain']). Trường này phục vụ cho việc debug và tối ưu hóa luồng đi.           |

Bảng 9 : Trường thông tin cốt lõi của Agent State

Hiện thực hóa bằng mã nguồn (Python):

```
class AgentState(MessagesState):
    """Agent state with context and metadata"""
    context: str = ""
    intent: str = "" # normal or deep
    current_query: str = ""
    lesson_id: Optional[Union[str, int]] = None
    metadata: dict = {}
    conversation_history: str = "" # Recent conversation for
    context
    thread_id: str = ""
```

Để đảm bảo tính liên tục của trải nghiệm người dùng, trạng thái này không chỉ tồn tại trong bộ nhớ RAM mà được bền vững hóa (Persisted) vào cơ sở dữ liệu PostgreSQL thông qua cơ chế 'Checkpointer'. Điều này cho phép người dùng có thể tải lại trang hoặc quay lại phiên học sau nhiều ngày mà vẫn giữ nguyên ngữ cảnh.

```

```python
# core/memory.py
class SessionMemory:
    """Manage session data with PostgreSQL persistence"""
    def update_session(self, thread_id: str, data: Dict[str, Any], persist: bool = True):
        # Update cache ...
        # Persist to database
        if persist:
            create_or_update_session(
                thread_id=thread_id,
                messages=messages,
                # ...
            )
```

```

### 4.3. Thiết kế Chi tiết Các Nút Xử lý (Nodes Implementation)

Hệ thống được mô đun hóa thành 5 Node chức năng chuyên biệt. Mỗi Node là một hàm độc lập, nhận vào State hiện tại, thực hiện xử lý logic và trả về bản cập nhật cho State.

#### a. Intent Node (Bộ phân loại ý định)

Đây là "cửa ngõ" đầu tiên của hệ thống. Thay vì lãng phí tài nguyên để truy xuất dữ liệu cho mọi câu hỏi, Node này sử dụng một mô hình LLM nhỏ, tốc độ cao (như Gemini Flash) để phân tích ngữ nghĩa và xác định người dùng thực sự muốn gì.

- **Input:** user\_query và messages (lịch sử chat gần nhất).
- **Kỹ thuật: Few-shot Prompting** (Cung cấp ví dụ mẫu trong prompt).
- **Logic phân loại:**
  - deep: Các câu hỏi yêu cầu giải thích cơ chế, nguyên lý hoặc hướng dẫn giải bài tập (VD: "Tại sao...", "Làm thế nào...", "Giảng lại cho con...").
  - normal: Các câu hỏi tra cứu định nghĩa, công thức hoặc sự thật hiển nhiên (VD: "Phân số là gì?", "1 km bằng bao nhiêu m?").
  - greeting: Các câu chào hỏi xã giao không cần truy xuất kiến thức.

#### b. Metadata Node (Bộ làm giàu ngữ cảnh)

- **Chức năng:** Xác định chính xác ngữ cảnh bài học mà người dùng đang tham gia.
- **Logic:** Node này truy vấn cơ sở dữ liệu quan hệ (PostgreSQL) dựa trên lesson\_id hiện tại của phiên học để lấy thông tin như: Tên bài học, Mục tiêu cần đạt, Các từ khóa trọng tâm.
- **Tác dụng:** Giúp Agent "hiểu" ngầm ngữ cảnh. Ví dụ: Khi đang học bài "Phân số", nếu học sinh hỏi "Số ở trên gọi là gì?", Agent sẽ tự động hiểu là "Tử số" nhờ có metadata của bài học, thay vì trả lời chung chung.

### c. Retrieve Node (Bộ truy xuất kiến thức)

Đây là điểm kết nối với Pipeline Truy xuất (đã trình bày ở Mục 3.2). Node này thực hiện chuỗi hành động:

1. **Query Rewriting:** Viết lại câu hỏi dựa trên lịch sử chat (ví dụ: "Nó là gì?" -> "Phân số là gì?").
2. **Hybrid Search:** Gọi Vector DB để tìm kiếm các chunk văn bản phù hợp nhất, sử dụng bộ lọc metadata để giới hạn phạm vi tìm kiếm trong bài học hiện tại.
3. **Context Update:** Cập nhật nội dung tìm được vào trường context của State.

### d. Generation Nodes (Các bộ não tạo sinh)

Để tối ưu hóa trải nghiệm người dùng, nhóm không sử dụng một prompt chung cho tất cả trường hợp mà tách thành hai nhánh xử lý riêng biệt:

- **Answer Node (Chế độ Trả lời nhanh):**
  - o **Kích hoạt khi:** intent == 'normal'.
  - o **Nhiệm vụ:** Đóng vai trò như một từ điển sống.
  - o **Chiến lược:** Trả lời ngắn gọn, trực tiếp vào vấn đề, trích dẫn chính xác định nghĩa từ tài liệu. Không lan man giải thích dông dài.
- **Explain Node (Chế độ Giảng giải sâu):**
  - o **Kích hoạt khi:** intent == 'deep'.
  - o **Nhiệm vụ:** Đóng vai trò là một giáo viên sư phạm.
  - o **Chiến lược:** Sử dụng kỹ thuật Chain-of-Thought (CoT) để kích hoạt khả năng suy luận từng bước. Prompt yêu cầu mô hình không được đưa ra đáp án ngay mà phải:
    - Bước 1: Phân tích khó khăn của học sinh.
    - Bước 2: Tìm ví dụ ẩn dụ (Analogy) gần gũi với đói sống (bánh kẹo, hoa quả).
    - Bước 3: Hướng dẫn giải quyết vấn đề từng bước một.

## 4.4. Cơ chế Điều hướng và Luồng Điều khiển (Routing Logic)

Sự linh hoạt và thông minh của Agent nằm ở các Conditional Edges (Cạnh điều kiện). Thay vì luồng đi cứng nhắc, hệ thống sử dụng một hàm Router (Bộ định tuyến) để quyết định bước tiếp theo dựa trên trạng thái hiện tại.

### Thuật toán Router (Mã giả):

```
def route_intent(state: AgentState) -> Literal["answer", "explain"]:  
    """Route by intent"""  
    return "explain" if state.get("intent") == "deep" else "answer"
```

Trong LangGraph, logic này được cấu hình như sau:

```

# Conditional routing by intent
workflow.add_conditional_edges(
    "retrieve",
    route_intent,
    {"answer": "answer", "explain": "explain"}
)

```

Cơ chế này đảm bảo tài nguyên tính toán được phân bổ hợp lý: những câu hỏi đơn giản được xử lý nhanh gọn, trong khi những vấn đề phức tạp nhận được sự "đầu tư" suy luận sâu sắc hơn.

#### 4.5. Kỹ thuật Prompt Engineering Chuyên sâu

Để Agent thực sự mang "linh hồn" của một giáo viên Toán tiểu học chứ không phải một cỗ máy vô cảm, nhóm đã áp dụng kỹ thuật Persona Prompting kết hợp với In-Context Learning cực kỳ chi tiết.

Cấu trúc System Prompt cho Explain Node:

```

{teacher_role}

**TASK**: Provide a **DETAILED**, **STEP-BY-STEP** explanation suitable for a Grade {grade} student.

**LESSON INFO**:
- Subject: {subject} | Grade: {grade} | Topic: {topic}
-----
**LESSON CONTEXT**:
{context}
-----

**STUDENT QUESTION**: "{question}"

{accuracy_constraints}

**PLANNING INSTRUCTION (Internal Thought)**:
Before generating the response, think about:
1. **Barrier Analysis**: What specific concept is confusing the student?
2. **Analogy/Example**: What real-world example (e.g., candy, money, toys) fits this concept?
3. **Structure**: How to break this down into simple steps (Step 1, Step 2...)?

**RESPONSE GUIDELINES**:
- Start with encouragement.
- Use bullet points or numbered steps for clarity.
- Use the analogy thought of in the planning phase.
- End with a "Check for Understanding" question (e.g., "Con thấy chỗ này dễ hiểu hơn chưa?").

**OUTPUT**: (Vietnamese, "Cô - Con" style)

```

#### 4.6. Tích hợp Công cụ Mở rộng (Tool Calling Strategy)

Ngoài khả năng hội thoại, Agent còn được trang bị "tay chân" để thực hiện các hành động cụ thể thông qua cơ chế Tool Calling. Các công cụ này được định nghĩa dưới dạng Function Schema và được LLM tự động gọi khi cần thiết.

### a. Mindmap Generator Tool (Công cụ vẽ sơ đồ tư duy)

Để hỗ trợ phương pháp học tập trực quan (Visual Learning), nhóm phát triển công cụ tự động tạo sơ đồ tư duy từ nội dung bài học.

#### Cơ chế hoạt động:

- **Context Retrieval:** Hệ thống truy xuất nội dung cốt lõi của bài học từ Vector DB (Top-5 chunks quan trọng nhất).
- **Structural Extraction:** Sử dụng một LLM mạnh (GPT-4o) ở chế độ `JSON Mode` để phân tích văn bản thành cấu trúc cây (Root -> Main Ideas -> Sub-branches).
- **Graph Rendering:** Dữ liệu JSON trả về được chuẩn hóa theo định dạng Nodes/Edges của thư viện `React Flow` để Frontend có thể vẽ trực tiếp.

#### Mã nguồn thực thi (Python):

```
```python
def generate_mindmap_json(topic: str, lesson_id: Union[str, int] = None) -> dict:
    # 1. Get lesson context
    context = get_context(topic, k=5, lesson_id=lesson_id)

    # 2. Format prompt with context
    prompt = format_prompt(
        SYSTEM_PROMPTS["mindmap"],
        context=context,
        topic=topic
    )

    # 3. Request LLM to generate strict JSON
    llm = ChatOpenAI(model="gpt-4o", model_kwargs={"response_format": {"type": "json_object"}})
    response = llm.invoke([HumanMessage(content=prompt)])

    # 4. Parse returning JSON
    return json.loads(response.content)
```

```

#### System Prompt (Mindmap):

```
```text
TASK: Generate React Flow mindmap JSON.

LESSON CONTEXT:
{context}

TOPIC: {topic}

Return ONLY valid JSON with nodes and edges. Root at top, main branches below.
```

```

## b. Learning Analyzer Tool (Công cụ phân tích học tập)

Công cụ này đóng vai trò như một "Giáo viên chủ nhiệm" ảo, hoạt động ngầm để theo dõi và cung cấp đánh giá chuyên sâu, khách quan về sự tiến bộ học tập của học sinh.

- **Cơ chế hoạt động:** Analyzer được kích hoạt để tiến hành Phân tích đối chiếu đa nguồn (Multi-Source Cross-Analysis), thay vì chỉ dựa trên một nguồn dữ liệu duy nhất.
  - **Thu thập Dữ liệu:** Hệ thống truy vấn và tổng hợp các dữ liệu quan trọng từ Course Service, bao gồm: Lịch sử hội thoại (Chat History) và Kết quả làm bài kiểm tra (Quiz Results) của người dùng.
  - **Educational Assessment:** Prompt được thiết kế sử dụng Persona "Assessor" (Người đánh giá) để phân tích, đối chiếu và tổng hợp các thông tin từ hai nguồn dữ liệu trên.
- **Các Tiêu chí Đánh giá Nâng cao:** Hệ thống sử dụng LLM để đưa ra các đánh giá dựa trên tiêu chí định tính và định lượng mới:
  - **Định tính & Chuyên sâu (Qualitative & In-depth):**
    - **Xác định Lỗ hổng Kiến thức Cốt lõi (Core Knowledge Gap Identification):** Đây là chức năng cốt lõi, thực hiện đối chiếu lỗi sai trong bài kiểm tra với các chủ đề mà học sinh đã hỏi trong lịch sử chat log. Nếu học sinh hỏi nhiều về một nội dung (chứng tỏ có thắc mắc lý thuyết) và sau đó thất bại trong bài quiz liên quan, hệ thống sẽ kết luận đây là một lỗ hổng lý thuyết thực sự.
    - **Nhận xét Học thuật:** Đánh giá điểm mạnh, điểm yếu dựa trên nội dung tương tác của học sinh.
  - **Định lượng (Quantitative):**
    - **Phân tích Tiến bộ (Progress Tracking):** Phân tích sự thay đổi trong hiệu suất học tập của học sinh thông qua chuỗi kết quả làm bài kiểm tra qua các lần (ví dụ: làm lại quiz, quiz từ bài học tiếp theo).
    - **Mức độ Chuyên cần:** Đếm số lượt tương tác để xếp loại mức độ tham gia học tập (Tốt/Trung bình/Cần cải thiện).

### Mã nguồn thực thi (Python):

```

```python
def analyze_session(conversation_history: str, lesson_id: Union[str, int] = None) -> dict:
    # 1. Fetch Transcript & Context
    # ...

    # 2. Analyze with Assessor Persona
    prompt = format_prompt(
        SYSTEM_PROMPTS["analyzer"],
        conversation_history=conversation_history,
        # ...
    )
    analysis = llm.invoke([HumanMessage(content=prompt)]).content

    # 3. Rule-based Level Assessment
    messages_count = conversation_history.count("\n") // 2
    if messages_count >= 10:
        level = "Tốt"
    elif messages_count >= 5:
        level = "Trung bình"
    else:
        level = "Cần cải thiện"

    return {
        "analysis": analysis,
        "level": level
    }
```

```

### System Prompt (Mindmap):

```

```text
You are an Objective Educational Assessor.
TASK: Evaluate the learning session.

CHAT HISTORY:
{conversation_history}

OUTPUT FORMAT (Vietnamese):
1. Kiến thức đã học: ...
2. Điểm mạnh: ...
3. Cần cải thiện: ...
4. Lời khuyên: ...
```

```

## 4.7. Tổng kết Kiến trúc Hệ thống

Kiến trúc Agent hướng đồ thị (Graph-based Architecture) với nền tảng LangGraph, kết hợp cùng chiến lược Prompt Engineering chuyên sâu và cơ chế Tool Calling linh hoạt, đã tạo nên một hệ thống trợ lý ảo giáo dục vượt trội. Hệ thống không chỉ có khả năng trả lời chính xác nhờ RAG, mà còn sở hữu "trí tuệ sư phạm": biết phân loại nhu cầu, biết

điều chỉnh phương pháp giải thích, và biết sử dụng công cụ hỗ trợ trực quan. Đây là bước tiến quan trọng, chuyển dịch từ mô hình Chatbot hỏi-đáp thụ động sang mô hình AI Companion (Người đồng hành) chủ động trong giáo dục.

## 5. Thiết kế API và hiện thực hóa triển khai

Để chuyển đổi các mô hình AI từ môi trường thử nghiệm sang một sản phẩm thực tế có khả năng phục vụ người dùng cuối, nhóm đã xây dựng một kiến trúc hệ thống phân tán, tách biệt giữa tầng xử lý (Compute Layer) và tầng dữ liệu (Data Layer).

Phần này trình bày chi tiết về thiết kế chuẩn hóa API và quy trình triển khai thực tế trên hạ tầng Cloud (Railway và Supabase) theo phương thức thủ công để đảm bảo kiểm soát chặt chẽ quá trình vận hành.

### 5.1. Kiến trúc Đóng gói và Thiết kế API (API Design)

Hệ thống Backend (Agent Service) được phát triển bằng **FastAPI**, tuân thủ mô hình **Clean Architecture** để đảm bảo tính độc lập của logic nghiệp vụ và dễ dàng bảo trì.

#### a. Cấu trúc Dự án và Phân tầng (Project Structure)

Mã nguồn được tổ chức khoa học, chia thành các module chuyên biệt:

- **app/core/**: Chứa logic "bộ não" của Agent (LangGraph nodes, State schema). Đây là tầng quan trọng nhất, chứa các prompt và logic điều hướng.
- **app/services/**: Tầng dịch vụ hạ tầng (Infrastructure Services), bao gồm logic RAG (gọi Vector DB) và Ingestion (xử lý dữ liệu).
- **app/api/**: Tầng giao tiếp (Presentation Layer), định nghĩa các Router và xử lý HTTP Request/Response trong app.py.
- **app/models/**: Định nghĩa các DTO (Data Transfer Object) bằng Pydantic để validate dữ liệu chặt chẽ (ví dụ: ChatRequest, MindmapRequest).

#### b. Đặc tả Chi tiết các Endpoint (API Specification)

Dưới đây là thiết kế chi tiết (Input/Output) đã được hiện thực hóa, đảm bảo tính nhất quán khi tích hợp với Frontend.

- **Endpoint Hội thoại Thông minh (POST /api/agent/chat)**

Đây là điểm tiếp nhận chính, xử lý luồng RAG và hội thoại.

#### Request Body (Chat Request)

```

```json
{
  "userMessage": "Làm sao để quy đồng mẫu số?",
  "lessonId": 2
}
```

```

- (Lưu ý: user\_id được xác thực qua JWT Header, thread\_id được sinh tự động để quản lý phiên)
- **Logic xử lý:**
  - Context Loading:** Tải lịch sử chat từ PostgreSQL (Supabase) dựa trên thread\_id.
  - Graph Execution:** Kích hoạt LangGraph. Agent tự động định tuyến (Route) sang node giải thích (Explain Node) do phát hiện câu hỏi "Làm sao".
  - Response Generation:** Trả về kết quả kèm metadata phân loại ý định.

### Response Body (ChatData):

```

```json
{
  "reply": "Để quy đồng mẫu số, con hãy làm theo 3 bước sau...",
  "intent": "deep",
  "threadId": "user_123_session"
}
```

```

- **Endpoint Tạo Sơ đồ Tư duy (POST /api/lessons/mindmap)**
- **Cơ chế:** Endpoint này không vẽ hình ảnh tĩnh (JPG/PNG) mà trả về cấu trúc dữ liệu đồ thị JSON để Frontend (sử dụng thư viện React Flow) tự render. Điều này giúp sơ đồ có tính tương tác cao.

### Request Body (MindmapRequest):

```

```json
{
  "lessonId": 2
}
```

```

(Hệ thống tự động truy vấn nội dung bài học từ ID để trích xuất Topic)

### Response Structure (Graph JSON):

```

```json
{
  "mindmap": {
    "nodes": [
      {"id": "1", "data": {"label": "Phân số"}, "position": {"x": 0, "y": 0}},
      {"id": "2", "data": {"label": "Tử số"}, "position": {"x": 100, "y": 100}}
    ],
    "edges": [
      {"source": "1", "target": "2"}
    ],
    "topic": "Phân số"
  }
}
```

```

## 5.2. Hạ tầng Triển khai (Deployment Infrastructure)

Nhóm áp dụng chiến lược triển khai trên nền tảng PaaS (Platform as a Service) để tối ưu chi phí và tận dụng sức mạnh của các dịch vụ quản lý sẵn (Managed Services).

### a. Tầng Dữ liệu (Data Layer) - Supabase (PostgreSQL & PgVector)

**Supabase** được lựa chọn làm cơ sở dữ liệu chính nhờ khả năng hỗ trợ mạnh mẽ cho AI thông qua extension pgvector. Quá trình thiết lập được thực hiện thủ công để đảm bảo cấu hình chính xác:

- Khởi tạo Project:** Tạo mới project trên Dashboard Supabase, thiết lập password và region (Singapore) để giảm độ trễ.
- Kích hoạt Extension:** Chạy câu lệnh SQL trực tiếp trên Supabase SQL Editor để kích hoạt pgvector:

```
CREATE EXTENSION IF NOT EXISTS vector;
```

- Thiết kế Schema:**
  - Tạo bảng lesson\_embeddings để lưu trữ vector (kích thước 1536 chiều, tương thích model embedding).
  - Tạo bảng chat\_history để lưu trữ trạng thái hội thoại của LangGraph.
- Lấy Connection String:** Cấu hình chuỗi kết nối (URI) để tích hợp vào Backend.

### b. Tầng Ứng dụng (Compute Layer) - Railway (Docker Container)

Service API (Backend) được triển khai lên **Railway** thông qua quy trình đóng gói Docker thủ công, đảm bảo môi trường nhất quán giữa Local và Cloud.

- Đóng gói Ứng dụng (Dockerization):** Nhóm đã xây dựng Dockerfile tối ưu dung lượng:

```
# Sử dụng Base Image python 3.9 slim để tối ưu kích thước
```

```

FROM python:3.9-slim
# Thiết lập thư mục làm việc
WORKDIR /app
# Copy và cài đặt các thư viện phụ thuộc
COPY requirements.txt .
RUN pip install --no-cache-dir -r requirements.txt
# Copy toàn bộ mã nguồn vào container
COPY ..
# Mở port và khởi chạy ứng dụng với Uvicorn
CMD sh -c "uvicorn app:app --host 0.0.0.0 --port ${PORT:-8000}"

```

```

**Minh chứng (`Dockerfile`):**
```
FROM python:3.9-slim
WORKDIR /app
COPY requirements.txt .
RUN pip install --no-cache-dir -r requirements.txt
COPY ..
CMD sh -c "uvicorn app:app --host 0.0.0.0 --port ${PORT:-8000}"
```

```

- **Quy trình Triển khai Thủ công lên Railway:**
- **Bước 1:** Kết nối tài khoản Railway với GitHub Repository chứa mã nguồn dự án.
- **Bước 2 - Cấu hình Môi trường (Environment Variables):** Thiết lập thủ công các biến môi trường nhạy cảm trên giao diện Dashboard của Railway, bao gồm:
  - OPENAI\_API\_KEY: Key truy cập LLM.
  - SUPABASE\_URL: Đường dẫn kết nối Database.
  - SUPABASE\_KEY: Key xác thực.
- **Bước 3 - Build & Deploy:** Kích hoạt quá trình Build từ nhánh main. Railway tự động đọc Dockerfile, cài đặt dependencies và khởi chạy container.
- **Bước 4 - Domain Mapping:** Gán domain công khai (ví dụ: <https://agent-api-production.up.railway.app>) để Frontend có thể gọi API.

### 5.3. Kết luận về Triển khai

Việc kết hợp triển khai thủ công trên **Railway** (cho tầng xử lý) và **Supabase** (cho tầng dữ liệu) mang lại sự cân bằng giữa tính linh hoạt và độ ổn định.

- **Supabase** giải quyết bài toán lưu trữ vector phức tạp mà không cần tự quản lý server database vật lý.

- **Railway** giúp đơn giản hóa việc vận hành container, tự động xử lý HTTPS và cân bằng tải. Hệ thống sau khi triển khai đã hoạt động ổn định, phản hồi nhanh (latency trung bình < 1.5s cho các tác vụ RAG) và sẵn sàng tích hợp vào ứng dụng thực tế.

## 6. Thực nghiệm và Đánh giá hiệu quả AI Agent

Để đảm bảo AI Agent hoạt động đúng theo các yêu cầu nghiệp vụ đã đề ra và mang lại trải nghiệm học tập hiệu quả cho học sinh, nhóm nghiên cứu đã tiến hành quy trình đánh giá dựa trên phương pháp **Kiểm thử hộp đen (Black-box testing)**. Phương pháp này tập trung vào việc kiểm chứng đầu ra của Agent dựa trên các đầu vào (Input) đa dạng, mô phỏng các tình huống thực tế mà học sinh lớp 4 có thể gặp phải.

### 6.1. Phương pháp và Tiêu chí đánh giá

Quá trình đánh giá được thực hiện thông qua việc xây dựng bộ **Kịch bản kiểm thử (Test Scenarios)** bao phủ các chức năng cốt lõi của Agent. Các phản hồi của hệ thống được chấm điểm dựa trên 4 tiêu chí định lượng và định tính sau:

1. **Độ chính xác kiến thức (Accuracy):** Câu trả lời có đúng về mặt Toán học không? Có bị ảo giác (hallucination) hay sai lệch so với sách giáo khoa không?
2. **Khả năng hiểu ngữ cảnh (Context Awareness):** Agent có hiểu đúng ý định (Intent) của người dùng và duy trì được mạch hội thoại (Conversation Memory) không?
3. **Tính sư phạm (Pedagogical Quality):** Văn phong có phù hợp với học sinh tiểu học không? Có sử dụng phương pháp giải thích từng bước (Step-by-step) khi cần thiết không?
4. **Độ trễ phản hồi (Latency):** Thời gian sinh câu trả lời có nằm trong ngưỡng chấp nhận được (dưới 3-5 giây) để đảm bảo trải nghiệm người dùng không?

### 6.2. Các kịch bản kiểm thử chi tiết

Nhóm nghiên cứu đã thực hiện kiểm thử trên tập dữ liệu bài học: "Toán lớp 4 - Ôn tập các số đến 100.000". Dưới đây là các kịch bản tiêu biểu:

#### Kịch bản 1: Hỏi đáp kiến thức cơ bản (Normal Retrieval)

- **Mục tiêu:** Kiểm tra khả năng truy xuất thông tin định nghĩa/khai niệm từ Vector Database.
- **Input:** "Hàng chục nghìn là gì hả cô?"
- **Kỳ vọng:** Agent trả lời ngắn gọn, trích dẫn định nghĩa từ bài học, không giải thích lan man.
- **Kết quả thực tế:** "À, hàng chục nghìn là một trong những hàng số trong một số có nhiều chữ số. Khi con nhìn vào một số, hàng chục nghìn là vị trí thứ năm từ phải sang trái. Ví dụ, trong số 52431, chữ số 5 nằm ở hàng chục nghìn. Con hiểu chưa nào? Nếu có gì thắc mắc, con cứ hỏi cô nhé!"

- **Đánh giá:** Đạt (Pass). Agent truy xuất đúng chunk kiến thức và trả lời đúng trọng tâm.

### Minh chứng:

The screenshot shows a conversation in a messaging app. The user asks: "Hàng chục nghìn là gì hà cô?". The AI responds: "À, hàng chục nghìn là một trong những hàng số trong một số có nhiều chữ số. Khi con nhìn vào một số, hàng chục nghìn là vị trí thứ năm từ phải sang trái. Ví dụ, trong số 52431, chữ số 5 nằm ở hàng chục nghìn. Con hiểu chưa nào? Nếu có gì thắc mắc, con cứ hỏi cô nhé!".

Type your message...

Voice

### Kịch bản 2: Giải thích chuyên sâu và Tư duy từng bước (Deep Reasoning)

- **Mục tiêu:** Kiểm tra khả năng kích hoạt *Explain Node* và kỹ thuật *Chain-of-Thought*.
- **Input:** "Con không hiểu tại sao số 4 lại đứng ở đây? Cô giải thích kỹ hơn được không?"
- **Kỳ vọng:** Agent nhận diện Intent là 'deep', chuyển sang chế độ giải thích chi tiết, sử dụng ví dụ minh họa.
  - **Kết quả thực tế:**

"À, con thắc mắc tại sao số 5 lại đứng ở hàng chục nghìn đúng không?  
Cô sẽ giải thích kỹ hơn cho con nhé!  
1. Hiểu về vị trí của các chữ số:

- Khi con nhìn vào một số có nhiều chữ số, mỗi chữ số sẽ có một vị trí riêng, gọi là "hàng".
- Từ phải sang trái, các hàng lần lượt là: hàng đơn vị, hàng chục, hàng trăm, hàng nghìn, và hàng chục nghìn.

## 2. Ví dụ cụ thể:

- Lấy ví dụ số 52431. Con hãy đếm từ phải sang trái:
  - Số 1 ở hàng đơn vị.
  - Số 3 ở hàng chục.
  - Số 4 ở hàng trăm.
  - Số 2 ở hàng nghìn.
  - Số 5 ở hàng chục nghìn.

## 3. Tại sao số 5 ở hàng chục nghìn?:

- Vì số 5 là chữ số thứ năm từ phải sang trái trong số 52431. Do đó, nó nằm ở hàng chục nghìn.

## 4. Tưởng tượng thực tế:

- Hãy tưởng tượng con có 5 túi kẹo, mỗi túi có 10.000 viên kẹo. Vậy tổng cộng con có 50.000 viên kẹo. Số 5 đại diện cho 5 túi kẹo, và mỗi túi là 10.000 viên, nên nó nằm ở hàng chục nghìn.

Con thấy chỗ này dễ hiểu hơn chưa? Nếu còn thắc mắc gì, con cứ hỏi cô nhé!"

- **Đánh giá:** Đạt (Pass). Agent sử dụng giọng văn khuyến khích, chia nhỏ vấn đề và có sử dụng phép ẩn dụ (xếp hàng).

## Minh chứng:

 Con không hiểu tại sao số 5 lại đứng ở đây? Cô giải thích kỹ hơn được không?

À, con thắc mắc tại sao số 5 lại đứng ở hàng chục nghìn đúng không?  
Cô sẽ giải thích kỹ hơn cho con nhé!

**1. Hiểu về vị trí của các chữ số:**

- Khi con nhìn vào một số có nhiều chữ số, mỗi chữ số sẽ có một vị trí riêng, gọi là "hàng".
- Từ phải sang trái, các hàng lần lượt là: hàng đơn vị, hàng chục, hàng trăm, hàng nghìn, và hàng chục nghìn.

**2. Ví dụ cụ thể:**

- Lấy ví dụ số 52431. Con hãy đếm từ phải sang trái:
  - Số 1 ở hàng đơn vị.
  - Số 3 ở hàng chục.
  - Số 4 ở hàng trăm.

Type your message...

### Kịch bản 3: Duy trì ngữ cảnh hội thoại (Conversation Memory)

- **Mục tiêu:** Kiểm tra khả năng nhớ ngắn hạn (Short-term memory) của LangGraph.
- **Input (Lượt 1):** "Số 100.000 có mấy chữ số 0?"
- **Output (Lượt 1):** "À, con hỏi số 100.000 có mấy chữ số 0 đúng không? Số 100.000 có tổng cộng 5 chữ số 0. Cụ thể là: 1 chữ số 1 và 4 chữ số 0 đứng sau. Con thấy dễ hiểu không? Nếu còn thắc mắc gì, con cứ hỏi cô nhé!"
- **Input (Lượt 2):** "Thế còn số 10.000 thì sao?" (Người dùng không lặp lại câu hỏi "có mấy chữ số 0").
- **Kỳ vọng:** Agent hiểu "thì sao" nghĩa là hỏi về số lượng chữ số 0 của số 10.000.
- **Kết quả thực tế:**

"À, con hỏi về số 10.000 đúng không? Cô sẽ giải thích cho con nhé!

1. Hiểu về số 10.000:

- Số 10.000 có tổng cộng 5 chữ số.
- Cụ thể là: 1 chữ số 1 và 4 chữ số 0 đứng sau.

2. Vị trí của các chữ số:

- Số 0 ở hàng đơn vị.
- Số 0 ở hàng chục.
- Số 0 ở hàng trăm.
- Số 0 ở hàng nghìn.
- Số 1 ở hàng chục nghìn.

3. Tưởng tượng thực tế:

- Hãy tưởng tượng con có 1 túi kẹo, mỗi túi có 10.000 viên kẹo. Vậy tổng cộng con có 10.000 viên kẹo. Số 1 đại diện cho 1 túi kẹo, và mỗi túi là 10.000 viên, nên nó nằm ở hàng chục nghìn.

Con thấy dễ hiểu hơn chưa? Nếu còn thắc mắc gì, con cứ hỏi cô nhé!"

- **Đánh giá:** Đạt (Pass). Hệ thống LangGraph đã lưu trữ trạng thái hội thoại và hiểu đúng ngữ cảnh ẩn.

**Minh chứng:**



Thế còn số 10.000 thì sao?

À, con hỏi về số 10.000 đúng không? Cô sẽ giải thích cho con nhé!

**1. Hiểu về số 10.000:**

- Số 10.000 có tổng cộng 5 chữ số.
- Cụ thể là: 1 chữ số 1 và 4 chữ số 0 đứng sau.

**2. Vị trí của các chữ số:**

- Số 0 ở hàng đơn vị.
- Số 0 ở hàng chục.
- Số 0 ở hàng trăm.
- Số 0 ở hàng nghìn.
- Số 1 ở hàng chục nghìn.

**3. Tưởng tượng thực tế:**

- Hãy tưởng tượng con có 1 túi kẹo, mỗi túi có 10.000 viên kẹo. Vậy tổng cộng con có 10.000 viên kẹo. Số 1 đại diện cho 1 túi kẹo, và

Type your message...

Voice



#### Kịch bản 4: Xử lý câu hỏi ngoài phạm vi (Out-of-Scope Handling)

- **Mục tiêu:** Kiểm tra tính an toàn và khả năng định hướng người dùng (Guardrails).
- **Input:** "Cô ơi, dạy con cách chơi game Liên Minh Huyền Thoại đi."
- **Kỳ vọng:** Agent từ chối khéo léo và hướng học sinh quay lại bài học Toán.
- **Kết quả thực tế:**  
"À, con hỏi về cách chơi game Liên Minh Huyền Thoại đúng không? Nhưng mà cô là trợ giảng Toán, nên cô sẽ giúp con học Toán thật tốt nhé!"
- **Đánh giá:** Đạt (Pass). Agent không bị ảo giác bịa ra cách chơi game, giữ đúng vai trò (Persona) giáo viên.

**Minh chứng:**



Cô ơi, dạy con cách chơi game Liên Minh Huyền Thoại đi.

À, con hỏi về cách chơi game Liên Minh Huyền Thoại đúng không?  
Nhưng mà cô là trợ giảng Toán, nên cô sẽ giúp con học Toán thật tốt  
nhé! Nếu con có câu hỏi nào về Toán, đặc biệt là về các số đến  
100.000, con cứ hỏi cô nhé!

Bây giờ, cô sẽ giúp con ôn lại một chút về các số đến 100.000 nhé:

**1. Hiểu về số 10.000:**

- Số 10.000 có tổng cộng 5 chữ số.
- Cụ thể là: 1 chữ số 1 và 4 chữ số 0 đứng sau.

**2. Vị trí của các chữ số:**

- Số 0 ở hàng đơn vị.
- Số 0 ở hàng chục.
- Số 0 ở hàng trăm.
- Số 0 ở hàng nghìn.

Type your message...

Voice



## 7. Kết luận.

Chương 3 đã trình bày một cách toàn diện quy trình thiết kế và hiện thực hóa hệ thống "Trợ lý ảo hỗ trợ học tập". Đây là phần trọng tâm nhất của đồ án, nơi kết tinh các giải pháp kỹ thuật tiên tiến để giải quyết bài toán cá nhân hóa giáo dục.

Tóm lại, những kết quả trên đã chứng minh tính khả thi và ưu việt của giải pháp tích hợp AI Agent dựa trên kiến trúc hướng đồ thị và cơ chế truy xuất ngữ nghĩa đa tầng. Các thành tựu này là nền tảng vững chắc để chuyển sang giai đoạn hiện thực hóa và triển khai hệ thống Web mô phỏng lớp học ảo, nội dung sẽ được trình bày chi tiết trong Chương 4.

# CHƯƠNG 4: HỆ THỐNG WEB MÔ PHỎNG LỚP HỌC ẢO TÍCH HỢP AI AGENT

## 1. Cài đặt và triển khai hệ thống.

Để đảm bảo hệ thống hoạt động ổn định, có khả năng mở rộng và dễ dàng bảo trì, nhóm nghiên cứu đã lựa chọn mô hình triển khai phân tán trên nền tảng Điện toán đám mây (Cloud Computing). Kiến trúc hạ tầng được tách biệt rõ ràng giữa ba tầng: Giao diện (Frontend), Dịch vụ ứng dụng (Backend Services) và Cơ sở dữ liệu (Database), tận dụng sức mạnh của các nhà cung cấp dịch vụ PaaS (Platform-as-a-Service) hàng đầu hiện nay.

### 1.1. Kiến trúc triển khai tổng thể

Hệ thống được triển khai theo mô hình **Multi-Cloud**, tích hợp các dịch vụ sau:

- **Tầng Frontend (Client-side):** Triển khai trên **Vercel** để tận dụng mạng lưới phân phối nội dung (Edge Network) toàn cầu, giúp tối ưu hóa tốc độ tải trang cho người dùng cuối.
- **Tầng Backend (Server-side):** Triển khai toàn bộ hệ thống Microservices (Auth, Course, AI Agent) trên **Railway**, sử dụng công nghệ Containerization (Docker) để đảm bảo môi trường đồng nhất.
- **Tầng Dữ liệu (Data persistence):** Sử dụng **Supabase** (nền tảng dựa trên PostgreSQL) để quản lý cơ sở dữ liệu quan hệ và dữ liệu Vector (pgvector) cho AI.

### 1.2. Quy trình triển khai chi tiết

#### a. Triển khai Cơ sở dữ liệu trên Supabase

Supabase được lựa chọn làm xương sống dữ liệu cho toàn bộ hệ thống nhờ khả năng cung cấp PostgreSQL được quản lý sẵn (Managed Postgres) và hỗ trợ mạnh mẽ cho các ứng dụng AI.

##### • Cấu hình hạ tầng:

- **Region:** Singapore (ap-southeast-1) để giảm thiểu độ trễ (latency) đường truyền về Việt Nam.
- **Extensions:** Kích hoạt pgvector để hỗ trợ lưu trữ và truy vấn vector embedding cho tính năng RAG của AI Agent.
- **Kết nối:** Sử dụng Connection Pooling (Supervisor) để quản lý hiệu quả hàng nghìn kết nối đồng thời từ các microservice trên Railway, tránh hiện tượng nghẽn cổ chai.

#### b. Triển khai các Microservices trên Railway

Railway đóng vai trò là môi trường tính toán (Compute Environment) tập trung cho toàn bộ logic nghiệp vụ phía server.

- **Phương pháp đóng gói (Containerization):**
  - Mỗi dịch vụ (Auth Service, Course Service, AI Agent Service) được đóng gói thành các Docker Image riêng biệt thông qua Dockerfile. Điều này giúp cô lập môi trường, tránh xung đột thư viện (Dependency Conflict) giữa các service viết bằng Java (Spring Boot), NodeJS (NestJS) và Python (FastAPI).
- **Quy trình CI/CD:**
  - Thiết lập cơ chế **Git Trigger**: Mỗi khi có commit mới vào nhánh main trên GitHub, Railway sẽ tự động kích hoạt quy trình Build -> Test -> Deploy.
  - **Biến môi trường (Environment Variables)**: Các thông tin nhạy cảm (Database URL, OpenAI API Key, JWT Secret) được quản lý tập trung và mã hóa trên Dashboard của Railway, không lưu trữ trực tiếp trong mã nguồn (Hard-code).
- **Giao tiếp nội bộ (Internal Networking):**
  - Các service giao tiếp với nhau thông qua mạng nội bộ riêng (Private Network) của Railway, đảm bảo an toàn bảo mật và tốc độ truyền tải cao nhất, không lộ API nội bộ ra internet công cộng.

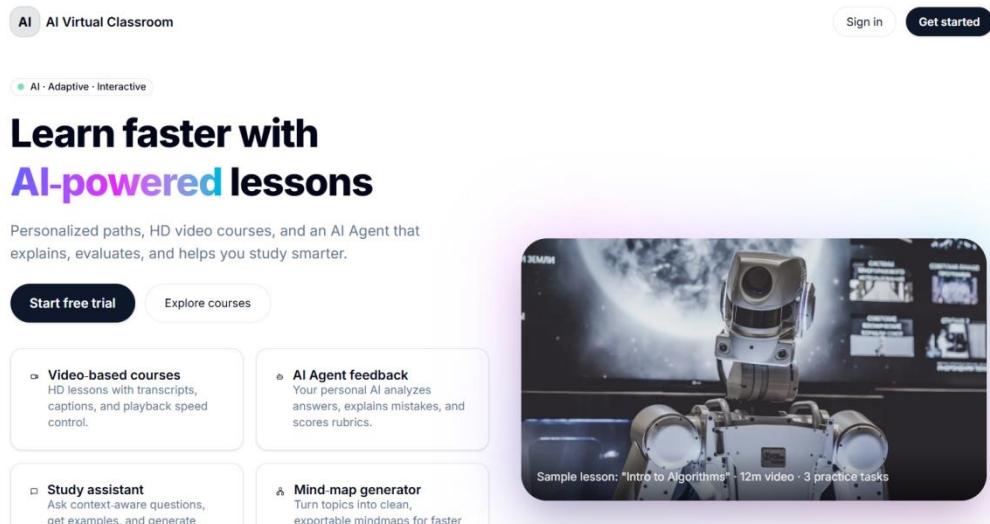
### c. Triển khai Frontend trên Vercel

Vercel là nền tảng tối ưu nhất cho Next.js, framework được sử dụng để xây dựng giao diện người dùng của hệ thống.

- **Tối ưu hóa hiệu năng:**
  - Sử dụng tính năng **Static Site Generation (SSG)** và **Incremental Static Regeneration (ISR)** của Next.js để cache các trang nội dung tĩnh (như danh sách khóa học, bài đọc) ngay tại Edge Server. Điều này giúp giảm tải cho Backend và tăng tốc độ phản hồi trang gần như tức thì.
  - **Image Optimization**: Các hình ảnh bài giảng, avatar người dùng được Vercel tự động nén và chuyển đổi định dạng (WebP/AVIF) để tối ưu băng thông.
- **Tên miền và Bảo mật (Domain & Security):**
  - Hệ thống được gán tên miền chính thức (Custom Domain).
  - Vercel tự động cung cấp và gia hạn chứng chỉ bảo mật **SSL/TLS (HTTPS)**, đảm bảo toàn bộ dữ liệu truyền tải giữa người dùng và hệ thống đều được mã hóa an toàn.

## 2. Giao diện website.

### a. Giao diện trang chủ.

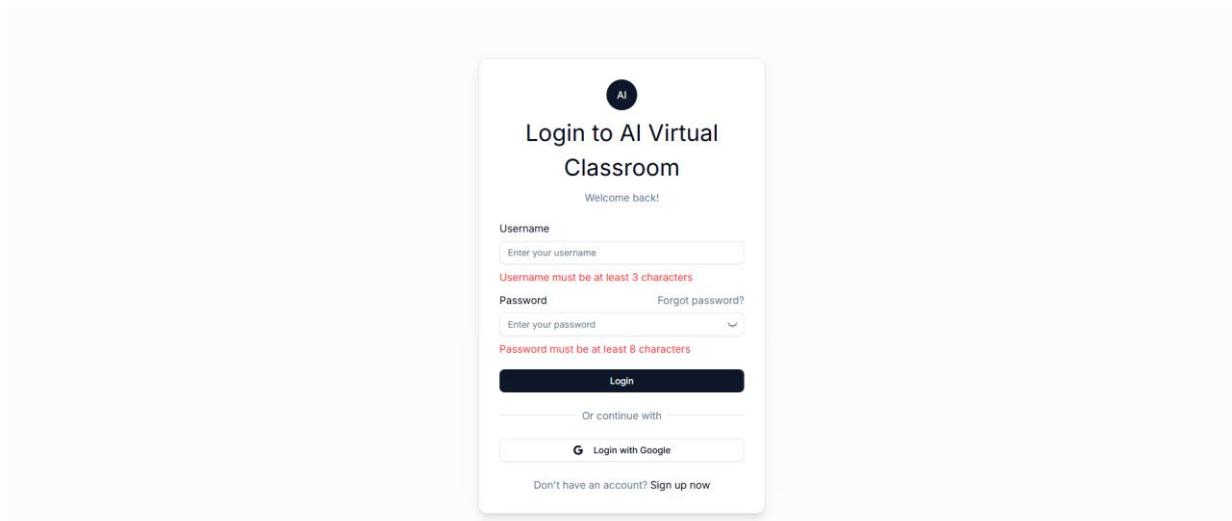


Hình 45: Giao diện trang chủ

- Ngay khi truy cập vào website hệ thống, người dùng sẽ nhìn thấy giao diện trang chủ được thiết kế với màu sắc hài hòa, bắt mắt. Những tính năng chính của hệ thống được hiển thị khái quát trên giao diện trang chủ này. Click vào button Sign in hoặc Get started, người dùng sẽ được điều phổi tới giao diện đăng ký, đăng nhập.

### b. Giao diện đăng nhập / đăng ký.

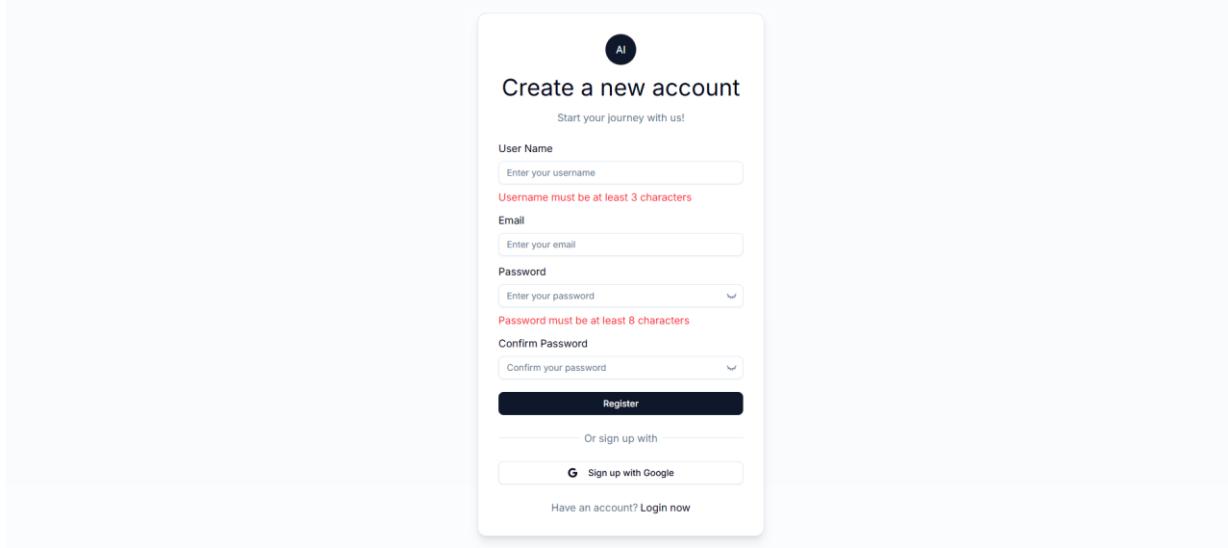
- Giao diện đăng nhập.



Hình 46: Giao diện đăng nhập

- Đến với giao diện đăng nhập, nếu người dùng đã thực hiện tạo tài khoản, giao diện này sẽ được hiển thị ra với yêu cầu nhập các trường Username (Username phải chứa ít nhất 3 kí tự) và Password (Phải chứa ít nhất 8 kí tự). Giao diện được thiết kế hiện đại, bắt mắt và giản dị. Ngay khi ấn Login, người dùng sẽ được điều hướng tới giao diện Dashboard.

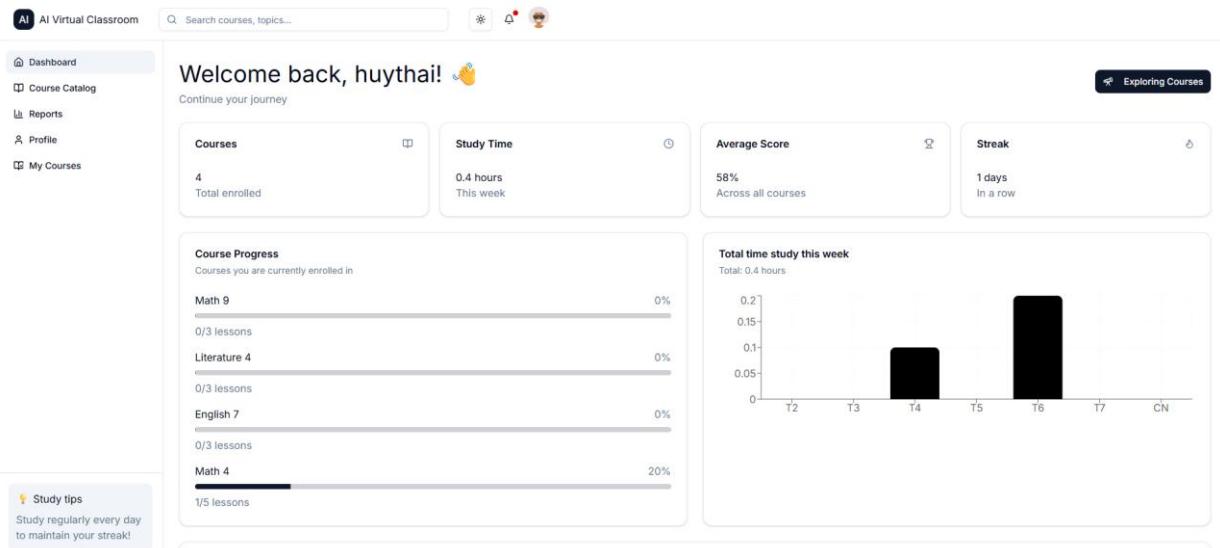
- Giao diện đăng ký.



*Hình 47: Giao diện đăng ký*

- Trong trường hợp người dùng lần đầu tiên truy cập vào hệ thống, người dùng sẽ phải thực hiện đăng ký một tài khoản mới với các trường Username (Chứa ít nhất 3 kí tự), Email, Password (Chứa ít nhất 8 kí tự) và Confirm Password (Dùng để xác nhận lại password đã tạo). Ngay khi ấn Register, thông tin tài khoản của người dùng sẽ được lưu trữ vào trong cơ sở dữ liệu, đồng thời, giao diện Dashboard sẽ hiển thị cho người dùng.

### c. Giao diện Dashboard.



*Hình 48: Giao diện Dashboard*

- Sau khi thực hiện đăng nhập / đăng ký, giao diện chính của hệ thống được hiển thị lên với mặc định khởi đầu được hiển thị cho người dùng chính là giao diện Dashboard. Tại giao diện này, người dùng sẽ quan sát được một số thông tin mang tính thống kê về quá trình học của bản thân. Các thông tin được hiển thị bao gồm Course (Tổng số khóa học hiện hành trên hệ thống), Study Time (Tổng số giờ học của người dùng trong tuần này),

Average Score (Điểm trung bình của các bài kiểm tra tương ứng với các bài học), Streak (Số ngày liên tiếp mà người dùng tham gia học). Ngoài ra còn một số thống kê về tiến trình học của người dùng.

#### d. Giao diện danh mục khóa học.

The screenshot shows the 'Course Catalog' section of the AI Virtual Classroom. On the left sidebar, 'Course Catalog' is selected. The main area displays a grid of course cards. Each card includes the course title, level, category, and a small image. A 'Study tips' box at the bottom left encourages regular study to maintain a streak.

| Course Title         | Level    | Category           |
|----------------------|----------|--------------------|
| Civic Education 8    | Advanced | Civic Education    |
| Physical Education 9 | Medium   | Physical Education |
| Music 5              | Basic    | Music              |

Hình 49: Giao diện danh mục khóa học

- Để đăng ký khóa học hay xem các khóa học hiện có trên hệ thống, người dùng truy cập giao diện Course Catalog. Giao diện này hiển thị cho người dùng toàn bộ các khóa học hiện có trên hệ thống. Người dùng có thể sử dụng tính năng tìm kiếm khóa học theo từ khóa, lọc khóa học theo Category, Level, v.v. Mỗi khóa học sẽ hiển thị một button View Course Detail để người dùng có thể xem chi tiết khóa học.

#### e. Giao diện báo cáo tiến trình học tập.

The screenshot shows the 'Progress Report' section of the AI Virtual Classroom. On the left sidebar, 'Reports' is selected. The main area displays various performance metrics and charts. A 'Study tips' box at the bottom left encourages regular study to maintain a streak.

| Metric            | Value     | Comparison                       |
|-------------------|-----------|----------------------------------|
| Total Time        | 0.7 hours | +0.7 hours compared to last week |
| Lessons Completed | 1 lessons | +1 lessons compared to last week |
| Average Score     | 58%       | +58% compared to last week       |
| Current Streak    | 1 days    | Record: 1 days                   |

**Study Time by Day:**

| Day | Study Time (approx.) |
|-----|----------------------|
| T7  | 0.00                 |
| CN  | 0.00                 |
| T2  | 0.00                 |
| T3  | 0.00                 |
| T4  | 0.10                 |
| T5  | 0.00                 |
| T6  | 0.20                 |

**Lessons Completed:**

| Day | Lessons Completed (approx.) |
|-----|-----------------------------|
| T7  | 0.00                        |
| CN  | 0.00                        |
| T2  | 0.00                        |
| T3  | 0.00                        |
| T4  | 0.00                        |
| T5  | 0.00                        |
| T6  | 4.00                        |

Hình 50: Giao diện báo cáo tiến trình học tập

- Đến với giao diện báo cáo tiến trình học tập, ở đây các thông số thống kê về tiến trình học tập của người dùng được hiển thị trên giao diện. Các thông số này giúp người dùng

năm bắt và kiểm soát được quá trình học tập của bản thân từ đó có những sự điều chỉnh tương ứng.

## f. Giao diện hồ sơ người dùng.

Hình 51: Giao diện hồ sơ người dùng

- Người dùng có thể chủ động quản lý thông tin tài khoản của mình thông qua giao diện hồ sơ người dùng. Giao diện này bao gồm ảnh đại diện và các thông tin thống kê nổi bật của người dùng. Bên cạnh đó, các thông tin về tài khoản như Username, Email, Gender, Phone và Description cũng được hiển thị, người dùng có thể chủ động thay đổi, chỉnh sửa các thông tin này qua Edit button.

## g. Giao diện các khóa học đã đăng ký.

Hình 52: Giao diện các khóa học đã đăng ký

- Các khóa học mà người dùng đã đăng ký được hiển thị tại giao diện My Course với button Continue Learning cho phép người dùng tiếp tục tham gia các khóa học mà người dùng đã đăng ký.

## h. Giao diện xem chi tiết khóa học.

The screenshot shows the AI Virtual Classroom interface. On the left, there's a sidebar with links to Dashboard, Course Catalog, Reports, My Courses, and Settings. A 'Study tips' box encourages regular study. The main area displays the course catalog for 'Civic Education 8'. It features a large title 'CONG DAN 8' with a cartoon illustration of four students. Below the title, there are buttons for 'Advanced', 'Civic Education', and 'New'. To the right, there's a section titled 'Start Learning now!' with a 'Enroll in Course' button. Another section lists 'Course Includes' with items like high-quality video lessons, practice exercises, AI Chat Agent support, and completion certificates. A 'Author' section shows a profile for 'Dk1utea', described as an expert in Civic Education.

Hình 53: Giao diện xem chi tiết khóa học

- Để xem chi tiết khóa học, người dùng có thể ấn vào button View Course Detail tại giao diện Course Catalog hay Continue Learning tại giao diện My Course. Cả hai lựa chọn này đều sẽ điều hướng người dùng tới giao diện chi tiết khóa học. Tại giao diện này, các thông tin nổi bật của khóa học sẽ được hiển thị lên cho người dùng như Author, Lessons, v.v. Khi user lựa chọn Enroll in course, thông tin đăng ký khóa học của người dùng sẽ được lưu lại trong cơ sở dữ liệu, đồng thời người dùng sẽ được điều hướng tới giao diện lớp học chính của lớp học ảo.

## i. Giao diện lớp học chính của lớp học ảo.

- Xem bài giảng và danh sách bài học.

The screenshot shows the AI Virtual Classroom interface. The top navigation bar includes 'AI Virtual Classroom', a search bar, and user icons. The main content area shows the course catalog for 'Math 4' under 'On tap cac so den 100000'. On the left, there's a math lesson titled 'Số có năm chữ số' with a 3D block diagram of numbers. On the right, there's a transcript and a chat agent interface. A 'Lesson List' panel on the right lists four lessons: 'Lesson 1: On tap cac so den 100000' (20:59), 'Lesson 2: On tap cac phep tinh trong pham vi 100000' (16:49), 'Lesson 3: So chan so le' (38:33), and 'Lesson 4: Bieu thuc chua chu' (20:33). Each lesson has a 'Take a quiz test' button.

Hình 54: Giao diện xem bài giảng và danh sách bài học

- Đến với giao diện chính của lớp học, video bài giảng được hiển thị cho người dùng tham gia học. Các tính năng tương tự so với các hệ thống trình phát video hiện hành như

start, pause, tăng giảm âm lượng, cài đặt tốc độ phát, phóng to, thu nhỏ màn hình video. Người dùng phải xem hết 90% thời lượng bài giảng mới có thể sử dụng tua nhanh, tua ngược. Để tham gia các bài học tiếp theo, người dùng có thể nhấn vào button Next Lesson để tham gia bài học tiếp theo. Bên cạnh giao diện phát video bài giảng, người dùng có thể lựa chọn xem danh sách bài học, xem phụ đề bài giảng, chat tương tác với trợ giảng ảo AI Agent hay xem sơ đồ tư duy bài học (Mindmap).

- Chat tương tác hỏi và trả lời với Agent.

Hình 55: Chat tương tác hỏi và trả lời với Agent

- Hệ thống tích hợp một trợ giảng ảo đóng vai trò giải đáp các thắc mắc về kiến thức của môn học và bài học tương ứng từ người dùng. Người dùng có thể trò chuyện, đặt câu hỏi để giải đáp thắc mắc của mình về bài học và nhận được lời giải thích chi tiết, dễ hiểu và thân thiện nhất đến từ trợ giảng ảo.
- Phụ đề bài giảng.

 Find in transcripts...

00:09 - 00:19

Xin chào tất cả các con học sinh thân mến, chào mừng các con đã đến với khóa học bài giảng toán lớp 4, bộ sách mang tên Kết nối tri thức. Hôm

00:19 - 00:28

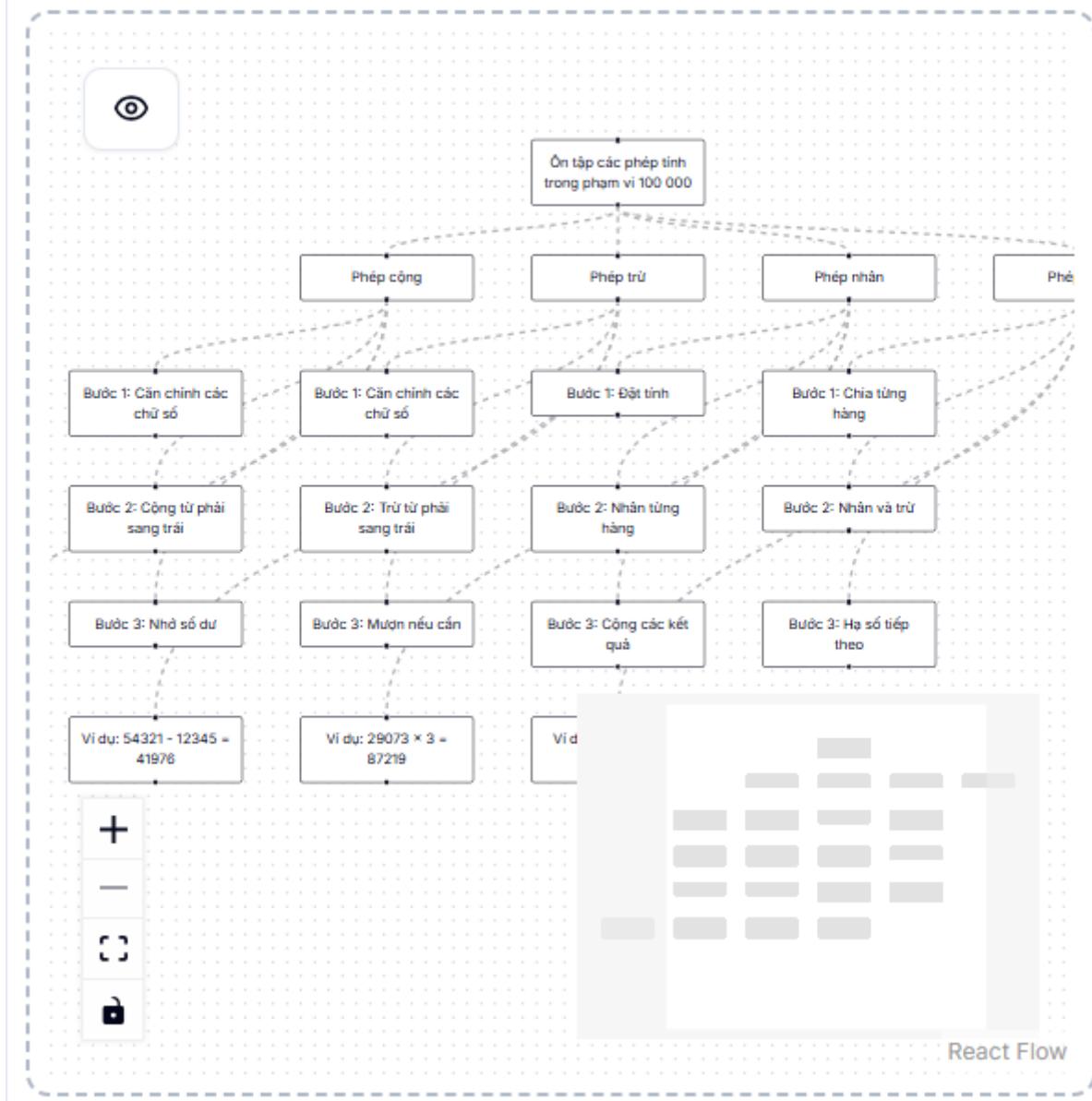
nay cô và các con sẽ cùng nhau bước vào chủ đề đầu tiên trong chương trình, đó là chủ đề 1 mang tên Ôn tập và bổ sung.

00:30 - 00:40

Và bài học ngày hôm nay của chúng ta chính là bài số 1, Ôn tập các số đến 100.000. Chúng mình đã cùng nhau học

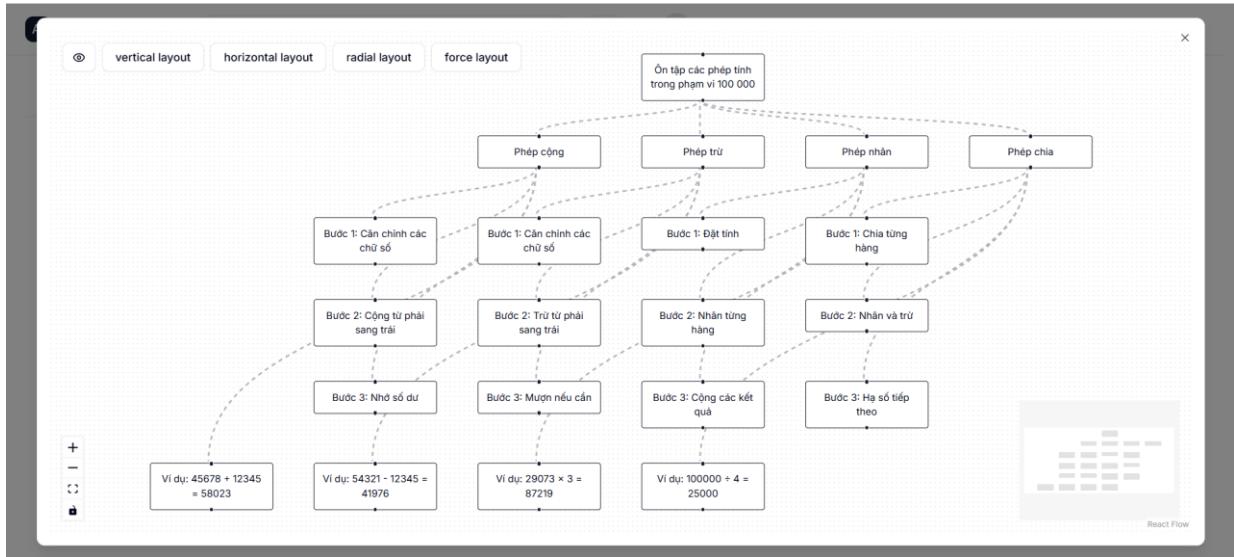
#### *Hình 56: Phụ đề bài giảng*

- Khi người dùng lựa chọn xem phụ đề bài giảng, hệ thống sẽ hiển thị phụ đề bài giảng với độ chính xác cao cho người dùng quan sát.
- Xem sơ đồ tư duy bài học.



Hình 57: Xem sơ đồ tư duy bài học

- Trong học tập, sơ đồ tư duy đóng vai trò vô cùng quan trọng trong việc tiếp thu kiến thức từ bài học của người học. AI Agent đóng vai trò sinh sơ đồ tư duy một cách thông minh, có hệ thống và trực quan, sinh động cho người dùng. Tại giao diện này, người dùng có thể thỏa sức kéo thả các node lý thuyết để dễ dàng hình dung được bài giảng.
- Hiển thị sơ đồ tư duy toàn màn hình.



Hình 58: Hiển thị sơ đồ tư duy toàn màn hình

- Ngoài ra, người dùng còn có thể xem tổng quan sơ đồ tư duy bài học trong một không gian lớn hơn. Bên cạnh đó, người dùng còn có thể lựa chọn các bộ cục hiển thị khác nhau của sơ đồ tư duy bài học, thực hiện kéo thả, điều chỉnh các node trong sơ đồ tư duy, phóng to, thu nhỏ sơ đồ tư duy, v.v.

### k. Giao diện tham gia làm bài kiểm tra cuối bài học.

- Sau khi tiến độ xem video bài giảng đạt 90%, người dùng có thể tham gia tính năng làm bài kiểm tra cuối mỗi bài học.
- Trước khi làm bài.

Hình 59: Giao diện Trước khi làm bài

- Tại giao diện chính của lớp học, khi người dùng nhấn vào button “Take a quiz test” tại giao diện xem danh sách bài học, người dùng sẽ được điều hướng sang giao diện làm bài kiểm tra. Tại đây, giao diện trước khi làm bài sẽ hiển thị thông tin bài kiểm tra bao gồm

tên bài kiểm tra, số câu hỏi và thời gian làm bài. Sau khi đọc xong hướng dẫn làm bài, người dùng lựa chọn Start Quiz để bắt đầu tham gia bài thi.

- Bắt đầu làm bài.

The screenshot shows a quiz interface titled "Bài kiểm tra: Ôn tập các phép tính trong phạm vi 100000". It displays a single question: "Kết quả của  $10.000 + 5.000$  là?". There are four options: 15.000, 50.000, 10.500, and 20.000. The user has selected the first option. Navigation buttons "Previous" and "Next" are visible at the bottom.

Hình 60: Giao diện Bắt đầu làm bài

- Sau khi bắt đầu làm bài, các câu hỏi trong bài kiểm tra sẽ hiển thị. Người dùng lần lượt trả lời các câu hỏi trong bài kiểm tra, có thể tự do lựa chọn Next hoặc Previous question để làm bài. Các câu hỏi được thiết kế đa dạng các loại như Multiple choices, True / False, v.v.

- Hoàn thành bài kiểm tra.

The screenshot shows a confirmation message: "Quiz Completed!". It states: "You have successfully completed the quiz 'Bài kiểm tra: Ôn tập các phép tính trong phạm vi 100000'. Results will be available shortly." A "View Results" button is visible at the bottom.

Hình 61: Giao diện Hoàn thành bài kiểm tra

- Sau khi hoàn thành trả lời các câu hỏi, giao diện thông báo hoàn thành bài kiểm tra được hiển thị cho người dùng. Người dùng có thể lựa chọn xem kết quả bài làm.

- Xem kết quả.

The screenshot shows a quiz interface with the following details:

- Status:** Passed
- CORRECT ANSWERS:** 5 / 6
- STATUS:** Excellent!
- Score:** 83%
- Review Answers:**
  - Q1. Kết quả của  $10.000 + 5.000$  là? Your answer: 15.000
  - Q2. Phép cộng có tính chất giao hoán (đổi chỗ). Your answer: True
- Retry Quiz** button

Hình 62: Giao diện Xem kết quả

- Kết quả bài kiểm tra sẽ được hiển thị cho người dùng. Người dùng sẽ quan sát được số điểm, số câu trả lời đúng, đáp án đúng của từng câu trả lời. Để làm lại bài quiz, người dùng lựa chọn Retry Quiz để tham gia làm lại bài kiểm tra này.

## I. Giao diện xem đánh giá năng lực cá nhân.

The message content is as follows:

1. **Đánh giá chung:** Chúc mừng Con đã hoàn thành bài học với thành tích xuất sắc và hiểu biết tốt về các phép tính trong phạm vi 100000. Điểm số 100% trong bài kiểm tra thể hiện sự nỗ lực và sự hiểu biết của Con!

2. **Phân tích chi tiết:**

- Điểm mạnh: Con đã hoàn thành bài kiểm tra với điểm số tuyệt vời là 100%! Điều này cho thấy sự chăm chỉ và kiến thức tốt của Con về các phép tính.
- Vấn đề cần khắc phục: Không có lỗi sai trong bài kiểm tra, tuy nhiên, hãy tiếp tục ôn tập và thực hành thêm để củng cố kiến thức và kỹ năng tính toán của mình.
- Lời khuyên cụ thể: Hãy tiếp tục giữ vững tinh thần học tập và thực hành thường xuyên. Cố gắng áp dụng kiến thức đã học vào các bài tập thực tế để phát triển kỹ năng tính toán của mình. Hãy hỏi Cố nếu cần sự hỗ trợ hoặc giải đáp thêm về bất kỳ vấn đề nào liên quan đến toán học. Chúc Con tiếp tục thành công trong hành trình học tập!

**Level:** Khá  
**Reason:** Đã hoàn thành 90% bài giảng và nắm vững lý thuyết cơ bản

OK button

Hình 63: Giao diện Xem đánh giá năng lực

- Sau khi kết thúc bài học và hoàn thành xong bài kiểm tra, người dùng có thể quan sát được đánh giá khách quan về năng lực của bản thân như điểm mạnh, điểm cần cải thiện, lời khuyên cụ thể đến từ Agent.

## 3. Kết luận.

Chương 4 đã hoàn thành việc trình bày chi tiết về quá trình Cài đặt, Triển khai Hệ thống và Giao diện Website của đồ án "Thiết kế và triển khai hệ thống Web mô phỏng lớp học ảo tích hợp AI Agent".

Tóm lại, Chương 4 là minh chứng cho việc chuyển đổi thành công từ giai đoạn thiết kế lý thuyết (Chương 2) và xây dựng nền tảng AI Agent (Chương 3) sang một hệ thống website hoàn chỉnh, sẵn sàng cho việc kiểm thử và ứng dụng thực tiễn.

## KẾT LUẬN

Đồ án "Thiết kế và triển khai hệ thống Web mô phỏng lớp học ảo tích hợp AI Agent" đã được hoàn thành, đáp ứng thành công mục tiêu giải quyết bài toán cấp thiết về cá nhân hóa trải nghiệm học tập trên quy mô lớn, vốn là hạn chế cốt lõi của các hệ thống E-Learning và MOOC truyền thống.

Nghiên cứu đã tiến hành phân tích toàn diện bối cảnh xã hội, xu hướng chuyển đổi số trong giáo dục (EdTech) và ứng dụng Trí tuệ Nhân tạo (AI), đặc biệt là sự trỗi dậy của các AI Agent tự chủ. Trên cơ sở đó, nhóm nghiên cứu đã thiết kế và triển khai một hệ thống lớp học ảo tiên tiến, tích hợp thành công AI Agent làm trợ giảng ảo cá nhân hóa.

Đồ án đã đạt được các đóng góp quan trọng, cả về mặt kỹ thuật và ứng dụng thực tiễn:

- **Kiến trúc hệ thống:** Hệ thống được xây dựng trên **kiến trúc Microservices** hiện đại, sử dụng sự kết hợp giữa Spring Boot/NestJS cho Backend và NextJS cho Frontend, đảm bảo tính độc lập, khả năng mở rộng (scalability) và hiệu năng xử lý cao.
- **Thiết kế AI Agent tiên tiến:** Phát triển một **AI Companion thực thụ** thay vì Chatbot đơn thuần. Agent sử dụng **kiến trúc Agentic RAG** điều phối bởi **LangGraph**, cho phép thực hiện tư duy vòng lặp, suy luận đa tầng và lập kế hoạch hành động, mang lại trải nghiệm tương tác tự nhiên và sâu sắc như một gia sư 1-1.
- **Giải pháp cá nhân hóa cốt lõi:**
  - Đảm bảo độ chính xác (grounding) của phản hồi Agent bằng cách truy xuất thông tin từ **dữ liệu ngữ cảnh** (Contextual Data) được chuẩn hóa từ bản ghi (Transcript) bài giảng, loại bỏ hiện tượng sai lệch thông tin.
  - Đề xuất cơ chế **Dánh giá Năng lực (Competency Assessment)** chính xác và tức thời thông qua việc sử dụng GPT-4o để phân tích liên tục hội thoại và hành vi học tập của người học.
  - Tối ưu hóa tổng hợp kiến thức bằng cách kết hợp **pgVector** và Mô hình Ngôn ngữ Lớn (LLM) để tự động trích xuất các khái niệm cốt lõi, hỗ trợ tạo **Mindmap** trực quan.

Hệ thống đã hoàn thiện và đạt được các mục tiêu đề ra (tối ưu hóa thời gian phản hồi, nâng cao mức độ hài lòng). Tuy nhiên, do giới hạn về thời gian và tài nguyên, đồ án vẫn còn một số hạn chế cần được cải thiện trong các phiên bản phát triển tiếp theo:

- **Phạm vi tính năng AI Agent ban đầu:** Các tính năng cốt lõi của AI Agent trong đồ án hiện tại tập trung vào tương tác cơ bản (hội thoại, tạo Mindmap và đánh giá tiến trình), chưa được mở rộng để tích hợp sâu với các công cụ bên ngoài (External Tools) nhằm thực hiện các tác vụ phức tạp (ví dụ: tự động tạo bài tập, mô phỏng phòng thí nghiệm ảo).
- **Mức độ chuyên sâu của đánh giá năng lực:** Mô hình Dánh giá Năng lực (Competency Assessment) hiện tại vẫn còn ở mức cơ bản, cần được tăng cường

để có thể phân tích sâu hơn hành vi học tập phi cấu trúc (unstructured learning behavior) nhằm dự đoán nguy cơ bỏ học (attrition rate) và điều chỉnh lộ trình học thích ứng (Adaptive Learning Path) một cách chủ động.

- **Phụ thuộc vào LLM thương mại:** Hệ thống vẫn đang phụ thuộc nhiều vào các API thương mại của Mô hình Ngôn ngữ Lớn (LLM) như GPT-4o, dẫn đến nguy cơ về chi phí và tính ổn định. Đò án chưa kịp nghiên cứu và áp dụng các kỹ thuật tinh chỉnh (Fine-tuning) trên các mô hình mã nguồn mở (Open-source LLMs) để giảm sự phụ thuộc này.

#### Hướng phát triển trong tương lai bao gồm:

- **Mở rộng tính năng Agent:** Tích hợp các công cụ bên ngoài (External Tools) sâu hơn để Agent có khả năng thực hiện các tác vụ phức tạp như tự động tạo bài tập, mô phỏng phòng thí nghiệm ảo hoặc tương tác với các LMS khác.
- **Tăng cường đánh giá:** Xây dựng một mô hình đánh giá học thuật chuyên sâu hơn dựa trên hành vi học tập phi cấu trúc (unstructured learning behavior) để dự đoán nguy cơ bỏ học (attrition rate) và điều chỉnh lộ trình học thích ứng (Adaptive Learning Path) một cách chủ động.
- **Tối ưu hóa chi phí LLM:** Nghiên cứu và áp dụng các kỹ thuật tinh chỉnh (Fine-tuning) trên các mô hình ngôn ngữ mã nguồn mở (Open-source LLMs) để giảm sự phụ thuộc vào các API thương mại (như GPT-4o) mà vẫn duy trì chất lượng phản hồi cao.

Đò án này đã chứng minh tính khả thi và hiệu quả của việc ứng dụng các công nghệ AI Agent tiên tiến vào lĩnh vực giáo dục, mở ra triển vọng xây dựng một môi trường học tập thông minh, cá nhân hóa và có khả năng mở rộng cao trong tương lai.

# TÀI LIỆU THAM KHẢO

- [1] P. Research, “AI in Education Market Size, Share and Trends 2025 to 2034,” Precedence Research, 2025.
- [2] UNESCO, “Guidance for generative AI in education and research,” *UNESCO*, 2023.
- [3] N. M. Hải, “Khám phá lợi ích và xu hướng nổi bật của việc ứng dụng AI trong giáo dục,” *VNPT AI*, 28 2 2025.
- [4] T. Tun, “MOOC là gì?,” *VIBLO ASIA*, 23 10 2020.
- [5] C. B. d. Leusse, “2018 Global Internet Report – Take our survey on the Future of the Internet,” Internet Society, 2018.
- [6] Hoàng Thị Minh Anh, Hoàng Anh Tuấn, Phạm Ngọc Dương, Nguyễn Hoàng Giang, “Mô hình MOOC - Xu hướng phát triển giáo dục số hiện đại,” *TẠP CHÍ KHOA HỌC GIÁO DỤC VIỆT NAM*, 2019.
- [7] OpenupEd, “Definition Massive Open Online Courses (MOOCs),” *OpenupEd*, 12 2015.
- [8] G. M. Insights, “AI in Education Market Size,” AI in Education Market Size, 2023.
- [9] E. Agency, “Vietnam EdTech White Paper 2024,” 2024.
- [10] G. Kamradt, “GitHub,” Full Stack Retrieval, [Trực tuyến]. Available: <https://github.com/FullStackRetrieval-com/RetrievalTutorials.git>.
- [11] Manh Hung Nguyen, Dinh Que Tran, “A SEMANTIC SIMILARITY MEASURE,” *Southeast Asian Journal of Sciences*, pp. 63-75, 2014.
- [12] LangChain, “LangGraph,” *LangChain*, 17 1 2024.
- [13] LangChain, LangGraph, “LangGraph overview,” *LangChain docs*.
- [14] L. Weng, “LLM Powered Autonomous Agents,” *Lil'Log*, 2023.
- [15] T. Batch, “AI agentic workflows,” *The Batch*, 27 3 2024.