

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI
PHÂN HIỆU TẠI TP. HỒ CHÍ MINH
BỘ MÔN CÔNG NGHỆ THÔNG TIN**



BÁO CÁO HỌC PHẦN PROJECT 1

ĐỀ TÀI

**XÂY DỰNG WEBSITE HỌC TỪ VỰNG TIẾNG ANH
TÍCH HỢP CHATBOT**

Giảng viên hướng dẫn : ThS. TRẦN PHONG NHÃ
Sinh viên thực hiện : HOÀNG GIA KIỆT
: BÙI TẤN PHÁT
: NGUYỄN TIẾN ĐẠT
Lớp : CÔNG NGHỆ THÔNG TIN
Khoá : 62

Tp. Hồ Chí Minh, năm 2024

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI
PHÂN HIỆU TẠI TP. HỒ CHÍ MINH
BỘ MÔN CÔNG NGHỆ THÔNG TIN**



BÁO CÁO HỌC PHẦN PROJECT 1

ĐỀ TÀI

**XÂY DỰNG WEBSITE HỌC TỪ VỰNG TIẾNG ANH
TÍCH HỢP CHATBOT**

Giảng viên hướng dẫn : ThS. TRẦN PHONG NHÃ
Sinh viên thực hiện : HOÀNG GIA KIỆT – 6251071049
: BÙI TẤN PHÁT – 6251071069
: NGUYỄN TIẾN ĐẠT – 6251071019
Lớp : CÔNG NGHỆ THÔNG TIN
Khoá : 62

Tp. Hồ Chí Minh, năm 2024

LỜI CẢM ƠN

Lời đầu tiên nhóm chúng em xin gửi lời cảm ơn sâu sắc đến thầy Trần Phong Nhã đã hỗ trợ và giúp đỡ nhóm chúng em hoàn thành học phần Project 1.

Chúng em cũng xin gửi lời cảm ơn đến Quý thầy cô Bộ môn Công nghệ thông tin Trường Đại học Giao thông Vận tải Phân hiệu tại TP. Hồ Chí Minh đã truyền đạt kiến thức cho chúng em trong 3 năm học vừa qua. Từ đó chúng em có nền tảng vững chắc để có thể ứng dụng những kiến thức đó vào dự án này.

Ngoài ra, chúng em xin gửi lời cảm ơn đến các anh, chị, bạn đã đồng hành và giúp đỡ chúng em về tài liệu. Trong quá trình thực hiện đề tài, chúng em còn gặp nhiều khó khăn, thiếu kiến thức thực tiễn nên dự án phát triển ra có thể chưa được đúng với các hệ thống thực tế hiện tại. Mong thầy có thể xem xét, đánh giá cho báo cáo của nhóm chúng em được cập nhật và chỉnh sửa tốt nhất.

Chúng em xin chân thành cảm ơn thầy./.

TP. Hồ Chí Minh, ngày ... tháng ... năm 2024

TM. Nhóm sinh viên thực hiện

Hoàng Gia Kiệt

[illegible]

Giảng viên hướng dẫn

ii

BẢNG PHÂN CÔNG CÔNG VIỆC

Người phụ trách	Công việc được giao	Đánh giá của nhóm
Hoàng Gia Kiệt	<ul style="list-style-type: none">- Lên ý tưởng đề tài, phân tích hệ thống.- Xây dựng cơ sở dữ liệu.- Phát triển hệ thống Backend.- Hỗ trợ thiết kế UI.	
Nguyễn Tiên Đạt	<ul style="list-style-type: none">- Thiết kế UI.- Phát triển Frontend.- Hỗ trợ phân tích hệ thống.	
Bùi Tấn Phát	<ul style="list-style-type: none">- Phát triển hệ thống Chatbot.- Hỗ trợ phân tích hệ thống.	

MỤC LỤC

LỜI CẢM ƠN	i
NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN.....	ii
BẢNG PHÂN CÔNG CÔNG VIỆC	iii
MỤC LỤC	iv
DANH MỤC CHỮ VIẾT TẮT.....	viii
DANH MỤC BẢNG BIỂU	x
DANH MỤC HÌNH ẢNH	xi
CHƯƠNG 1: TỔNG QUAN.....	1
1.1. Tổng quan đề tài	1
1.2. Lý do chọn đề tài.....	1
1.3. Mục tiêu của đề tài	2
1.4. Nội dung đề tài	2
1.5. Đối tượng nghiên cứu	2
1.6. Phạm vi nghiên cứu	2
1.7. Phương pháp thực hiện.....	3
1.8. Kết quả dự kiến	3
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT.....	4
2.1. Quy trình phát triển phần mềm.....	4
2.2. Quy trình phát triển phần mềm Agile và phương pháp Agile Scrum.....	5
2.2.1. Quy trình phát triển phần mềm Agile.....	5
2.2.2. Phương pháp Agile Scrum	6
2.3. Công nghệ Spring Boot	7
2.3.1. Spring Boot là gì?.....	7
2.3.2. Những tính năng nổi bật của Spring Boot.....	7
2.3.3. Kiến trúc của Spring Boot	9
2.4. Công nghệ Reactjs	11

2.4.1. Đặc điểm kỹ thuật của React.js	11
2.4.2. Ưu điểm của React.js.....	11
2.4.3. Hạn chế của React.js	12
2.5. Tổng quan về MySQL	12
2.5.1. MySQL là gì?	12
2.5.2. Những tính năng chính của MySQL	12
2.6. Kiến trúc Retrieval Augmented Generation (RAG)	13
2.6.1. Tổng quan về mô hình ngôn ngữ lớn	13
2.6.2. Tổng quan về Retrieval Augmented Generation (RAG).....	14
2.6.3. Cấu tạo của RAG.....	16
2.6.4. Ưu điểm của RAG	16
2.6.5. Nhược điểm của RAG	17
2.7. Vector Database.....	17
2.7.1. Từ cơ sở dữ liệu truyền thống đến sự ra đời của cơ sở dữ liệu vector	17
2.7.2. Cơ sở dữ liệu vector (Vector Database)	18
2.8. Ngôn ngữ lập trình Python	19
2.8.1. Giới thiệu về Python.....	19
2.8.2. Ưu điểm	19
2.8.3. Nhược điểm	20
2.9. Ngôn ngữ lập trình Java	20
2.9.1. Giới thiệu về Java.....	20
2.9.2. Ưu điểm	21
2.9.3. Nhược điểm	21
2.10. Ngôn ngữ lập trình Javascript (Js)	22
2.10.1. Giới thiệu về Javascript	22
2.10.2. Ưu điểm	23
2.10.3. Nhược điểm	23
CHƯƠNG 3: KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU	24

3.1. Khảo sát hiện trạng	24
3.1.1. Tầm quan trọng của việc học tiếng Anh.....	24
3.1.2. Nhu cầu học tiếng Anh của người Việt Nam	25
3.2. Phân tích yêu cầu.....	27
3.2.1. Yêu cầu chức năng	27
3.2.2. Yêu cầu phi chức năng	28
CHƯƠNG 4: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG	29
4.1. Sơ đồ ER	29
4.2. Sơ đồ use case tổng quát	29
4.3. Đặc tả một số use case chính.....	29
4.3.1. Chức năng đăng nhập	29
4.3.2. Chức năng trợ lý học tập	32
4.3.3. Chức năng quản lý học phần	34
4.3.4. Chức năng tìm kiếm	43
4.4. Thiết kế cơ sở dữ liệu	46
4.4.1. Xác định thực thể.....	46
4.4.2. Lược đồ quan hệ	48
4.4.3. Đặc tả các bảng dữ liệu	48
4.5. Thiết kế Chatbot	52
4.5.1. Quy trình xử lý dữ liệu	53
4.5.2. Thiết kế cuộc hội thoại	54
CHƯƠNG 5: XÂY DỰNG CHƯƠNG TRÌNH	55
5.1. Giao diện chính	55
5.2. Giao diện xử lý	59
KẾT LUẬN	68
1. Kết quả đạt được	68
2. Hạn chế	68
3. Hướng phát triển	68

PHỤ LỤC	69
TÀI LIỆU THAM KHẢO.....	70

DANH MỤC CHỮ VIẾT TẮT

STT	Viết tắt	Ý nghĩa	Diễn giải
1	RAG	Retrieval Augmented Generation	
2	API	Application Programming Interface	Giao diện lập trình ứng dụng
3	JWT	JSON Web Token	
4	UI	User Interface	Giao diện người dùng
5	HTTP	Hypertext Transfer Protocol	Giao thức truyền tải siêu văn bản
6	JSON	Javascript Object Notation	
7	CSDL	Cơ sở dữ liệu	
8	ER	Entity Relationship	Mối quan hệ thực thể
9	CRUD	Create, Read, Update và Delete	Thêm, Đọc, Cập nhật và Xóa
10	MVC	Model, View, Controller	
11	SQL	Structured Query Language	Ngôn ngữ truy vấn mang có cấu trúc
12	GPL	General Public License	Giấy phép công cộng
13	ODBC	Open Database Connectivity	
14	CPU	Central Processing Unit	Bộ xử lý trung tâm
15	I/O	Input/Output	Đầu vào/Đầu ra
16	AI	Artificial Intelligence	Trí tuệ nhân tạo
17	LLM	Large Language Model	Mô hình ngôn ngữ lớn
18	NLP	Natural Language Processing	Xử lý ngôn ngữ tự nhiên

STT	Viết tắt	Ý nghĩa	Diễn giải
19	UX	User Experience	Trải nghiệm người dùng
20	NoSQL	None Structured Query Language	Ngôn ngữ truy vấn phi cấu trúc
21	OOP	Object Oriented Programming	Lập trình hướng đối tượng
22	Web	Website	Trang web
23	ML	Machine Learning	Học máy
24	XML	Xml Extensible Markup Language	
25	DOM	Document Object Model	Mô hình Đối tượng Tài liệu
26	HTML	Hyper Text Markup Language	Ngôn ngữ đánh dấu siêu văn bản
27	JSX	JavaScript XML	
28	JVM	Java Virtual Machine	

DANH MỤC BẢNG BIỂU

Bảng 4.1. Đặc tả use case Đăng nhập.....	30
Bảng 4.2. Đặc tả use case Trợ lý học tập	33
Bảng 4.3. Đặc tả use case Quản lý học phần.....	35
Bảng 4.4. Đặc tả use case Tạo mới học phần.....	35
Bảng 4.5. Đặc tả use case Chỉnh sửa học phần	38
Bảng 4.6. Đặc tả use case Xóa học phần.....	41
Bảng 4.7. Đặc tả use case Tìm kiếm	44
Bảng 4.8. Đặc tả bảng USER	49
Bảng 4.9. Đặc tả bảng ROLE	49
Bảng 4.10. Đặc tả bảng TOPIC	50
Bảng 4.11. Đặc tả bảng WORD	50
Bảng 4.12. Đặc tả bảng TYPE	51
Bảng 4.13. Đặc tả bảng EXAM.....	51
Bảng 4.14: Đặc tả bảng TOKEN.....	51
Bảng 4.15. Đặc tả bảng QUESTION	52
Bảng 4.16. Đặc tả bảng WORD_TYPE	52

DANH MỤC HÌNH ẢNH

Hình 2.1. Chu trình phát triển phần mềm.....	4
Hình 2.2. Quy trình phát triển phần mềm trong Scrum.....	6
Hình 2.3. Các tầng trong Spring Boot.....	9
Hình 2.4. Luồng hoạt động của Spring Boot.....	10
Hình 2.5. Một vài LLM nổi tiếng.....	14
Hình 2.6. Mô hình RAG cơ bản	15
Hình 2.7. Một số Vector Database phổ biến	18
Hình 2.8. Logo ngôn ngữ lập trình Python.....	19
Hình 2.9. Logo ngôn ngữ lập trình Java.....	21
Hình 2.10. Logo ngôn ngữ lập trình Js.....	22
Hình 3.1. Các ngoại ngữ phổ biến được chọn học (Ảnh trích từ báo cáo).....	26
Hình 3.2. Các ngôn ngữ được tìm kiếm nhiều nhất trên Cốc Cốc	27
Hình 4.1. Sơ đồ ER.....	29
Hình 4.2. Use case tổng quát.....	29
Hình 4.3. Sơ đồ hoạt động use case Đăng nhập.....	31
Hình 4.4. Sơ đồ tuần tự use case Đăng nhập.....	32
Hình 4.5. Sơ đồ hoạt động use case Trợ lý học tập.....	34
Hình 4.6. Sơ đồ tuần tự use case Trợ lý học tập.....	34
Hình 4.7. Sơ đồ hoạt động use case Tạo mới học phần	36
Hình 4.8. Sơ đồ tuần tự use case Tạo mới học phần	37
Hình 4.9. Sơ đồ hoạt động use case Chỉnh sửa học phần.....	39
Hình 4.10. Sơ đồ tuần tự use case Chỉnh sửa học phần	40
Hình 4.11. Sơ đồ hoạt động use case Xóa học phần	42
Hình 4.12. Sơ đồ tuần tự use case Xóa học phần	43
Hình 4.13. Sơ đồ hoạt động use case Tìm kiếm.....	45
Hình 4.14. Sơ đồ tuần tự use case Tìm kiếm.....	46

Hình 4.15. Lược đồ quan hệ	48
Hình 4.16. Pipeline của hệ thống Chatbot.....	53
Hình 5.1. Giao diện đăng nhập.....	55
Hình 5.2. Giao diện đăng lý	55
Hình 5.3. Giao diện trang chủ Admin	56
Hình 5.4. Giao diện trang chủ người dùng	56
Hình 5.5. Giao diện danh sách học phần	57
Hình 5.6. Giao diện từ vựng	57
Hình 5.7. Giao diện trợ lý học tập	58
Hình 5.8. Giao diện nội dung học phần.....	58
Hình 5.9. Giao diện thông tin người dùng.....	59
Hình 5.10. Giao diện khóa người dùng	59
Hình 5.11. Giao diện chỉnh sửa học phần	60
Hình 5.12. Giao diện xóa học phần	60
Hình 5.13. Giao diện học.....	61
Hình 5.14. Giao diện kiểm tra	61
Hình 5.15. Giao diện kết quả kiểm tra	62
Hình 5.16. Giao diện tìm kiếm	62
Hình 5.17. Giao diện học từ vựng bằng flashcard.....	63
Hình 5.18. Giao diện từ điển	64
Hình 5.19. Giao diện thêm từ vựng của học phần.....	65
Hình 5.20. Giao diện chỉnh từ vựng của học phần.....	65
Hình 5.21. Giao diện xóa từ vựng học phần	66
Hình 5.22. Giao diện email để xác thực tài khoản	66
Hình 5.23. Giao diện email để đổi mật khẩu.....	67

CHƯƠNG 1: TỔNG QUAN

1.1. Tổng quan đề tài

Hiện nay, việc học Tiếng Anh, đặc biệt là từ vựng, đang được nhiều người quan tâm và đầu tư. Có nhiều nền tảng trực tuyến và ứng dụng hỗ trợ học từ vựng đã ra đời như:

Duolingo: Ứng dụng học ngoại ngữ thông qua trò chơi hóa và các bài luyện tập đơn giản.

Memrise: Hỗ trợ học từ vựng qua hệ thống thẻ ghi nhớ và video ngắn.

Quizlet: Cung cấp công cụ học từ vựng qua flashcards và bài kiểm tra.

Tuy nhiên, hầu hết các nền tảng này vẫn còn hạn chế về khả năng tương tác, chưa đáp ứng được nhu cầu cá nhân hóa cho người học và không tận dụng được các công nghệ AI hiện đại để nâng cao trải nghiệm học tập. Việc thiếu đi sự phản hồi tự nhiên và linh hoạt từ hệ thống khiến cho trải nghiệm học tập trở nên nhàm chán và kém hiệu quả. Hơn nữa, nhiều ứng dụng hiện tại chỉ tập trung vào việc học từ vựng một cách thụ động mà chưa kết hợp các tính năng hỗ trợ thực hành và phản hồi kịp thời.

Trong bối cảnh đó, việc xây dựng một website học từ vựng Tiếng Anh tích hợp nhiều tính năng hỗ trợ các phương pháp học tập tiên tiến như học theo chủ đề, làm bài kiểm tra và theo dõi tiến độ học tập sẽ khắc phục được các hạn chế trên. Đặc biệt, việc tích hợp tính năng chatbot giúp tạo thêm một kênh hỗ trợ học tập linh hoạt và tiện lợi. Chatbot sẽ giúp người dùng tra cứu nhanh từ vựng, cung cấp ví dụ minh họa và hỗ trợ giải đáp một số thắc mắc đơn giản trong quá trình học.

1.2. Lý do chọn đề tài

Trong xu thế toàn cầu hóa, nhu cầu học Tiếng Anh không ngừng tăng cao, đặc biệt là đối với kỹ năng từ vựng, một yếu tố then chốt giúp phát triển khả năng giao tiếp và hiểu biết ngôn ngữ. Phương pháp học từ vựng truyền thống qua sách vở hoặc ghi chép thường thiếu tính tương tác và không đủ hấp dẫn, khiến người học dễ nản chí và khó ghi nhớ lâu dài. Điều này đòi hỏi một giải pháp mới kết hợp công nghệ hiện đại để cải thiện hiệu quả học tập. Sự ra đời của các nền tảng học trực tuyến đã khắc phục phần nào vấn đề này nhưng vẫn còn hạn chế trong việc cung cấp trải nghiệm học tập linh hoạt và cá nhân hóa.

Website học từ vựng Tiếng Anh được xây dựng nhằm mang đến một môi trường học tập hiện đại, hỗ trợ người dùng học từ vựng theo chủ đề, làm bài kiểm tra và theo

dõi tiến độ học tập của mình. Việc tích hợp chatbot vào hệ thống sẽ giúp người dùng thuận tiện hơn khi cần tra cứu nghĩa của từ, tìm kiếm ví dụ minh họa hoặc các câu hỏi về ngữ pháp trong tiếng Anh. Tất cả những điều này góp phần tạo nên một công cụ học tập toàn diện và hiệu quả hơn cho người dùng.

1.3. Mục tiêu của đề tài

Xây dựng một website hỗ trợ học từ vựng Tiếng Anh với các tính năng như:

- Hỗ trợ học từ vựng theo chủ đề, làm bài kiểm tra để đánh giá kiến thức và theo dõi tiến độ học tập.
- Tính năng chatbot được tích hợp vào website nhằm hỗ trợ tra cứu nhanh và giải đáp thắc mắc trong quá trình học từ vựng và các kiến thức về tiếng Anh, giúp người học có thêm một công cụ hỗ trợ linh hoạt.
- Đảm bảo tính an toàn và bảo mật thông tin người dùng.

1.4. Nội dung đề tài

Nội dung của đề tài bao gồm các công việc chính như phân tích yêu cầu hệ thống, thiết kế giao diện người dùng và kiến trúc Backend, phát triển các tính năng học từ vựng và kiểm tra trên Website. Bên cạnh đó, việc phát triển chatbot AI tích hợp công nghệ RAG là một phần quan trọng để tạo ra tính năng hỏi đáp thông minh và luyện tập hội thoại. Ngoài ra, hệ thống thống kê và báo cáo tiến trình học tập sẽ giúp người dùng theo dõi kết quả học tập của mình một cách dễ dàng và hiệu quả.

1.5. Đối tượng nghiên cứu

Đối tượng nghiên cứu của đề tài bao gồm các phương pháp học từ vựng trực tuyến hiện có, công nghệ chatbot AI và mô hình RAG trong xử lý ngôn ngữ tự nhiên. Đề tài hướng đến việc nghiên cứu và áp dụng những công nghệ này vào việc phát triển một nền tảng học từ vựng có tính tương tác và hiệu quả cao.

1.6. Phạm vi nghiên cứu

Phạm vi của đề tài bao gồm việc phát triển một website học từ vựng với các chức năng cơ bản như học từ vựng theo chủ đề, làm bài kiểm tra và thống kê tiến trình học. Đồng thời, phạm vi nghiên cứu còn bao gồm việc phát triển và tích hợp chatbot AI có khả năng hỗ trợ hỏi đáp và luyện tập hội thoại Tiếng Anh. Đối tượng sử dụng hệ thống là học sinh, sinh viên và người đi làm có nhu cầu học từ vựng Tiếng Anh. Ngoài ra, giáo viên và phụ huynh cũng có thể sử dụng hệ thống để hỗ trợ trong quá trình học tập.

1.7. Phương pháp thực hiện

Phương pháp nghiên cứu được áp dụng trong đề tài bao gồm việc thu thập và phân tích tài liệu liên quan đến các nền tảng học từ vựng hiện có, cũng như các công nghệ AI và mô hình RAG. Đề tài sử dụng phương pháp phát triển phần mềm theo mô hình Agile để đảm bảo tính linh hoạt và hiệu quả trong quá trình phát triển. Các công cụ và ngôn ngữ lập trình như ReactJS, Node.js, Java, Spring Boot, MySQL và thư viện LlamaIndex sẽ được sử dụng để xây dựng hệ thống. Cuối cùng, hệ thống sẽ được kiểm thử qua các bước như kiểm thử chức năng, kiểm thử hiệu năng và kiểm thử bảo mật để đảm bảo chất lượng sản phẩm.

1.8. Kết quả dự kiến

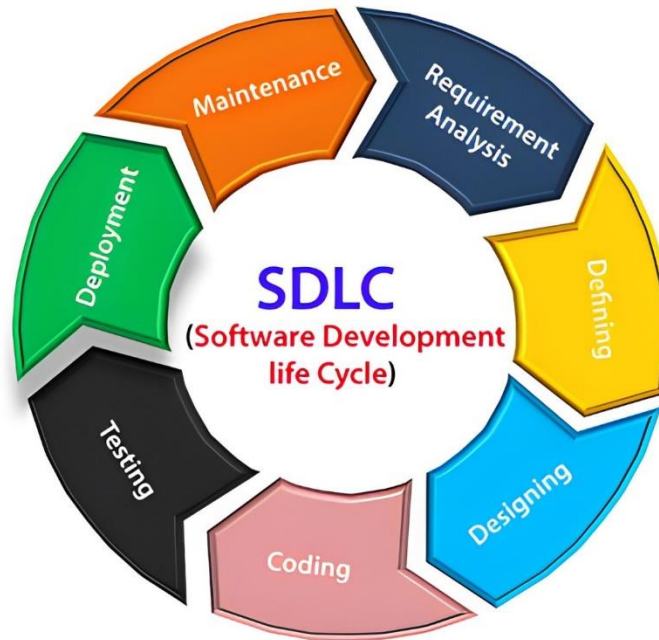
Sau khi hoàn thành, website sẽ mang lại những lợi ích như sau:

- Tiết kiệm thời gian học từ vựng. Giúp người học cải thiện khả năng từ vựng nhanh chóng, hiệu quả.
- Tăng cường khả năng tự học và thực hành Tiếng Anh. Nâng cao khả năng phản xạ và ghi nhớ từ vựng.
- Tối ưu hóa việc học từ vựng thông qua chatbot AI.
- Ứng dụng công nghệ AI vào giáo dục, góp phần chuyển đổi số trong giáo dục ngôn ngữ.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1. Quy trình phát triển phần mềm

Chu trình phát triển phần mềm là một chuỗi các bước có tổ chức nhằm đảm bảo việc xây dựng và triển khai phần mềm một cách hiệu quả và chất lượng. Gồm sự tham gia của nhà phân tích, nhà thiết kế, người phát triển và người dùng để phát triển và thực hiện một hệ thống thông tin. [1]



Hình 2.1. Chu trình phát triển phần mềm

Quy trình phát triển phần mềm bao gồm các giai đoạn cụ thể từ khi bắt đầu dự án cho đến khi hoàn thiện và bảo trì sản phẩm:

Xác định vấn đề: đề cập đến việc xác định tính khả thi của hệ thống và ý tưởng ban đầu cho dự án. Điều quan trọng nhất ở giai đoạn này là đảm bảo rằng hệ thống được đề xuất có ý nghĩa và cần thiết.

Phân tích yêu cầu: tập trung vào việc hiểu rõ các yêu cầu của hệ thống. Việc này bao gồm xác định rõ các tính năng cần thiết, phạm vi của dự án và nhu cầu cụ thể của người dùng. Kết quả của giai đoạn này là một tài liệu đặc tả yêu cầu chi tiết, đóng vai trò là cơ sở cho các giai đoạn tiếp theo.

Thiết kế: tạo ra một kế hoạch cụ thể cho việc triển khai yêu cầu đã được phân tích. Bao gồm việc thiết kế kiến trúc hệ thống, cơ sở dữ liệu và giao diện người dùng. Kiến trúc hệ thống xác định cách các thành phần trong phần mềm sẽ tương tác với nhau, còn thiết kế cơ sở dữ liệu đảm bảo việc lưu trữ và truy xuất dữ liệu hiệu quả. Đồng thời,

giao diện người dùng (UI/UX) được thiết kế để mang lại trải nghiệm thân thiện và dễ sử dụng cho người dùng cuối.

Xây dựng phần mềm: trong giai đoạn này, thực hiện viết mã cho từng thành phần của hệ thống, bao gồm cả Frontend và Backend. Frontend tập trung vào giao diện và trải nghiệm người dùng, trong khi backend xử lý logic nghiệp vụ và tương tác với cơ sở dữ liệu. Quá trình này thường được chia thành các vòng lặp nhỏ để dễ dàng kiểm soát tiến độ và điều chỉnh khi cần thiết. Thử nghiệm hệ thống: kiểm tra tính hoàn chỉnh và đúng đắn của hệ thống trước khi đưa vào sử dụng.

Kiểm thử: khi việc phát triển hoàn thành, tiến hành kiểm tra phần mềm để đảm bảo các tính năng hoạt động đúng như yêu cầu. Việc kiểm thử bao gồm kiểm thử chức năng để đảm bảo phần mềm thực hiện đúng các tác vụ, kiểm thử hiệu năng để đánh giá tốc độ và độ ổn định của hệ thống, và kiểm thử bảo mật để phát hiện và khắc phục các lỗ hổng bảo mật. Giai đoạn này giúp phát hiện lỗi và đảm bảo chất lượng phần mềm trước khi đưa vào sử dụng.

Triển khai: quá trình triển khai bao gồm việc cài đặt, cấu hình hệ thống và đảm bảo phần mềm hoạt động ổn định trên máy chủ hoặc hạ tầng đám mây. Đôi khi, việc triển khai có thể thực hiện theo từng giai đoạn hoặc toàn bộ hệ thống tùy theo chiến lược của dự án. Cuối cùng triển khai hệ thống cho người dùng cuối, bao gồm việc chuẩn bị dữ liệu và đào tạo người dùng.

Bảo trì, nâng cấp: là việc duy trì và sửa lỗi phát sinh của hệ thống sau khi nó đã được triển khai để đảm bảo hiệu suất và tính chính xác. Đồng thời, các tính năng mới có thể được bổ sung dựa trên phản hồi từ người dùng để cải thiện trải nghiệm sử dụng. Việc bảo trì liên tục giúp phần mềm luôn đáp ứng nhu cầu và thích nghi với các thay đổi trong tương lai.

2.2. Quy trình phát triển phần mềm Agile và phương pháp Agile Scrum

2.2.1. Quy trình phát triển phần mềm Agile

Agile là một phương pháp phát triển phần mềm linh hoạt, ra đời nhằm giải quyết những hạn chế của các phương pháp truyền thống như Waterfall. Agile tập trung vào việc cải tiến liên tục, phản hồi nhanh từ khách hàng và khả năng thích ứng với thay đổi trong suốt vòng đời dự án. Quy trình Agile được chia thành các chu kỳ ngắn gọi là Iteration hoặc Sprint, mỗi chu kỳ thường kéo dài từ 1 đến 4 tuần. Trong mỗi Iteration, nhóm phát triển hoàn thành một phần sản phẩm có thể hoạt động được, từ đó liên tục kiểm tra và cải tiến dựa trên phản hồi từ khách hàng và các bên liên quan. Agile khuyến khích sự cộng tác chặt chẽ giữa các thành viên trong nhóm, khách hàng và người dùng cuối, đảm bảo sản phẩm đáp ứng đúng nhu cầu thực tế. Việc chia nhỏ công việc và liên

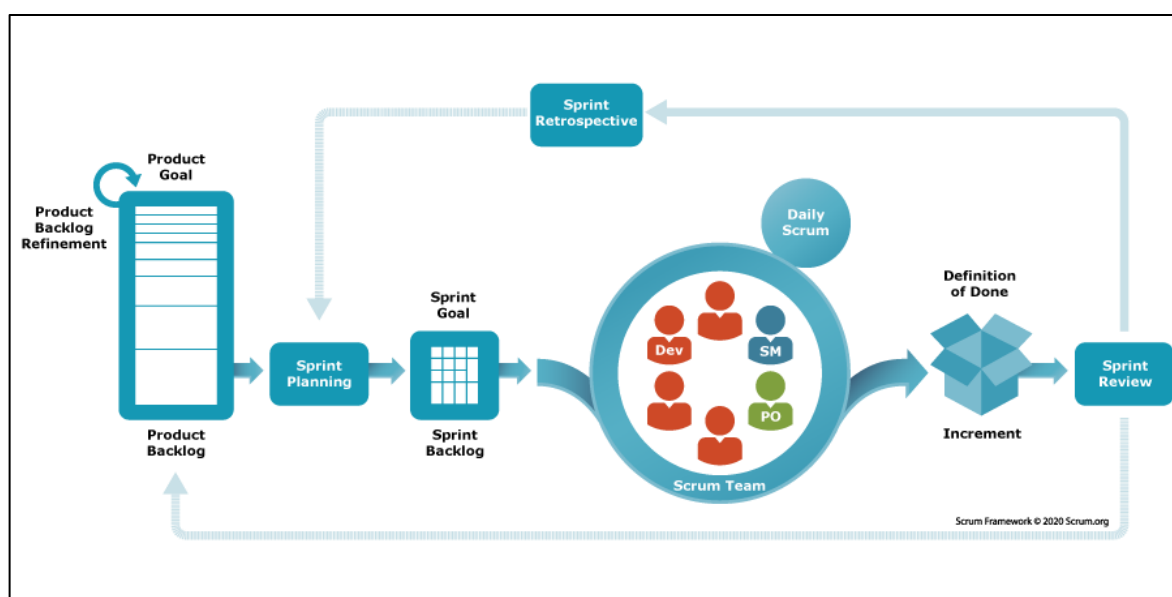
tục đánh giá giúp giảm thiểu rủi ro và đảm bảo sản phẩm cuối cùng có chất lượng cao. [2]

Agile dựa trên 4 giá trị cốt lõi từ **Tuyên ngôn Agile**: tập trung vào cá nhân và sự tương tác hơn quy trình và công cụ, phần mềm hoạt động được hơn là tài liệu đầy đủ, hợp tác với khách hàng hơn là đàm phán hợp đồng, và phản hồi nhanh với thay đổi hơn là bám sát kế hoạch cố định. Nhờ vào tính linh hoạt, Agile phù hợp với các dự án có yêu cầu không rõ ràng hoặc thường xuyên thay đổi, đặc biệt trong môi trường công nghệ phát triển nhanh.

2.2.2. Phương pháp Agile Scrum

Scrum là một trong những framework phổ biến nhất trong Agile, được sử dụng để quản lý và phát triển phần mềm theo hướng lặp lại và gia tăng. Scrum giúp các nhóm làm việc tự tổ chức, cộng tác hiệu quả và thích nghi nhanh với thay đổi. Quy trình Scrum chia dự án thành các chu kỳ làm việc ngắn gọi là Sprint, mỗi Sprint thường kéo dài từ 1 đến 4 tuần và có mục tiêu rõ ràng để hoàn thành một phần tính năng cụ thể của sản phẩm. [3]

Scrum hoạt động dựa trên ba vai trò chính: Product Owner, Scrum Master và Development Team. Product Owner chịu trách nhiệm xác định và ưu tiên các tính năng trong Product Backlog – danh sách các yêu cầu và nhiệm vụ cần thực hiện. Scrum Master đảm bảo quy trình Scrum diễn ra suôn sẻ, giúp loại bỏ các rào cản và hỗ trợ nhóm phát triển. Development Team là những người trực tiếp thực hiện các công việc lập trình, kiểm thử và triển khai.



Hình 2.2. Quy trình phát triển phần mềm trong Scrum

Quy trình Scrum bắt đầu với **Sprint Planning** – cuộc họp lập kế hoạch để xác định các công việc cần thực hiện trong Sprint. Trong suốt Sprint, nhóm họp ngắn hàng ngày (**Daily Stand-up**) để cập nhật tiến độ và giải quyết các vấn đề phát sinh. Cuối mỗi Sprint, nhóm tổ chức **Sprint Review** để trình bày kết quả công việc và nhận phản hồi từ khách hàng, sau đó thực hiện **Sprint Retrospective** để đánh giá và cải tiến quy trình làm việc cho Sprint tiếp theo.

Scrum giúp tăng cường tính minh bạch, khả năng thích ứng và cải tiến liên tục, nhờ đó nhóm phát triển có thể nhanh chóng điều chỉnh theo nhu cầu của khách hàng và thị trường. Việc chia nhỏ dự án thành các Sprint giúp giảm thiểu rủi ro và đảm bảo mỗi phần của sản phẩm đều được hoàn thiện một cách chất lượng.

2.3. Công nghệ Spring Boot

2.3.1. Spring Boot là gì?

Spring Boot là một Framework Java mã nguồn mở được thiết kế để đơn giản việc phát triển ứng dụng web dựa trên ngôn ngữ nền tảng là Java [4]. Nó được tạo ra dựa trên Framework Spring và cung cấp một hướng tiếp cận cấu hình tiện lợi hơn, cho phép các nhà phát triển có thể nhanh chóng tạo ra các ứng dụng sẵn sàng trên môi trường production mà không tốn nhiều thời gian để cấu hình.

Mục đích chính của Spring Boot là loại bỏ việc mã bị trùng lặp và những cấu hình phức tạp mà các nhà phát triển thường phải đối mặt trong khi xây dựng các ứng dụng Web với Framework Spring. Nó đi kèm với một tập hợp các giá trị mặc định có ý kiến và khả năng cấu hình tự động, giúp giảm thời gian và công sức cần thiết để thiết lập ứng dụng Web Spring.

2.3.2. Những tính năng nổi bật của Spring Boot

2.3.2.1. Cấu hình dễ dàng (Easy Configuration)

Spring Boot tự động định cấu hình các thành phần khác nhau dựa trên các phụ thuộc có trong đường dẫn lớp. Nó loại bỏ nhu cầu cấu hình rõ ràng và giảm thời gian phát triển cần thiết để thiết lập một ứng dụng web.

2.3.2.2. Phụ thuộc khởi đầu (Starter Dependencies)

Spring Boot cung cấp nhiều loại phụ thuộc "khởi động (Starter)". Đây là các gói phụ thuộc được cấu hình sẵn. Những phần khởi đầu này đơn giản hóa việc quản lý phần phụ thuộc bằng cách cung cấp một tập hợp các phần phụ thuộc cho các chức năng cụ thể như truy cập dữ liệu (Data access), bảo mật (Security), kiểm tra (Testing), ...

2.3.2.3. Máy chủ nhúng (Embedded Servers)

Spring Boot đi kèm với các máy chủ Web nhúng như Tomcat, Jetty, Netty và Undertow. Các máy chủ này cho phép các nhà phát triển chạy ứng dụng của họ dưới dạng tệp JAR hoặc WAR độc lập mà không cần triển khai đến máy chủ bên ngoài. Nó đơn giản hóa quá trình triển khai và làm cho việc phân phối và chạy các ứng dụng Spring Boot trở nên liền mạch.

2.3.2.4. Công cụ giám sát Spring Boot (Spring Boot Actuator)

Spring Boot Actuator đi kèm với một tính năng để giám sát ứng dụng. Nó hiển thị một tập hợp các điểm cuối cho phép các nhà phát triển thu thập số liệu (Gather metrics), theo dõi tình trạng sức khỏe (Health status), xem nhật ký (View logs), thực hiện kết xuất luồng và hơn thế nữa.

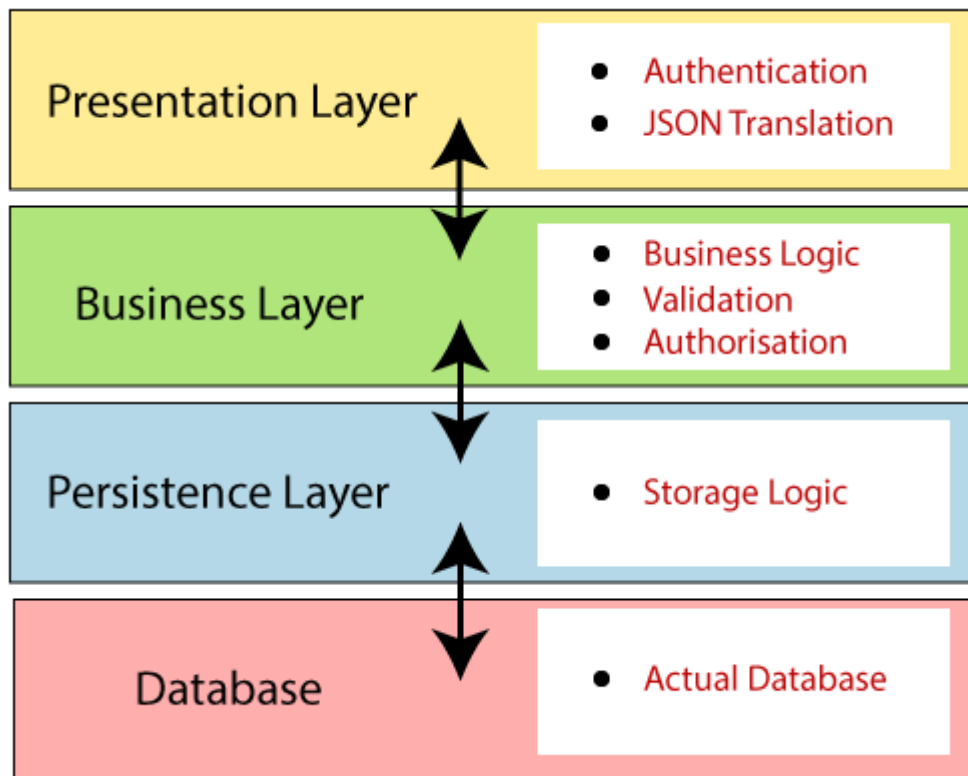
Spring Boot Actuator đơn giản hóa các khía cạnh hoạt động của ứng dụng của bạn và cho phép giám sát hiệu quả ứng dụng trong quá trình sản xuất.

2.3.2.5. Công cụ dành cho nhà phát triển (Developer Tool)

Spring Boot cung cấp một loạt công cụ thân thiện với nhà phát triển giúp nâng cao năng suất trong quá trình phát triển. Nó đi kèm với các tính năng như tự động khởi động lại ứng dụng, tải lại trực tiếp (Live reloading) các thay đổi, hỗ trợ gỡ lỗi nâng cao, ... Công cụ này giúp hợp lý hóa quy trình làm việc của nhà phát triển và giảm hơn nữa thời gian cần thiết để các thay đổi mã phản ánh trong ứng dụng đang chạy.

2.3.3. Kiến trúc của Spring Boot

2.3.3.1. Các tầng trong Spring Boot



Hình 2.3. Các tầng trong Spring Boot

a) Tầng trình bày (Presentation Layer)

Đây là tầng trên cùng của kiến trúc Spring Boot. Nó bao gồm các lượt xem, tức là phần front-end của ứng dụng.

Xử lý các yêu cầu HTTP và thực hiện xác thực

Chịu trách nhiệm chuyển đổi tham số của trường JSON thành đối tượng Java và ngược lại. Khi nó thực hiện xác thực yêu cầu, nó sẽ chuyển nó sang tầng tiếp theo – tầng nghiệp vụ. [5]

b) Tầng nghiệp vụ (Business Layer)

Tầng nghiệp vụ chứa tất cả logic dành riêng cho ứng dụng đang được phát triển. Nó tương tác với cả lớp trình bày (Presentation Layer) và tầng duy trì (Persistence Layer) để thực hiện trách nhiệm của mình.

Chịu trách nhiệm thực hiện xác nhận, thực hiện ủy quyền và xử lý các quy tắc và logic nghiệp vụ và nó bao gồm các lớp dịch vụ.

c) Tầng duy trì (Persistence Layer)

Tầng duy trì chịu trách nhiệm về tất cả logic lưu trữ bao gồm CSDL và bất kỳ hình thức lưu trữ dữ liệu nào khác. Nó chịu trách nhiệm chứa logic lưu trữ, tìm nạp các đối tượng và dịch chúng thành các hàng trong CSDL (và ngược lại).

Tầng này tương đương với giao diện kho lưu trữ và các truy vấn cơ sở dữ liệu được viết bên trong giao diện này. Tầng Persistence là Tầng duy nhất giao tiếp với Tầng Business và Tầng Database.

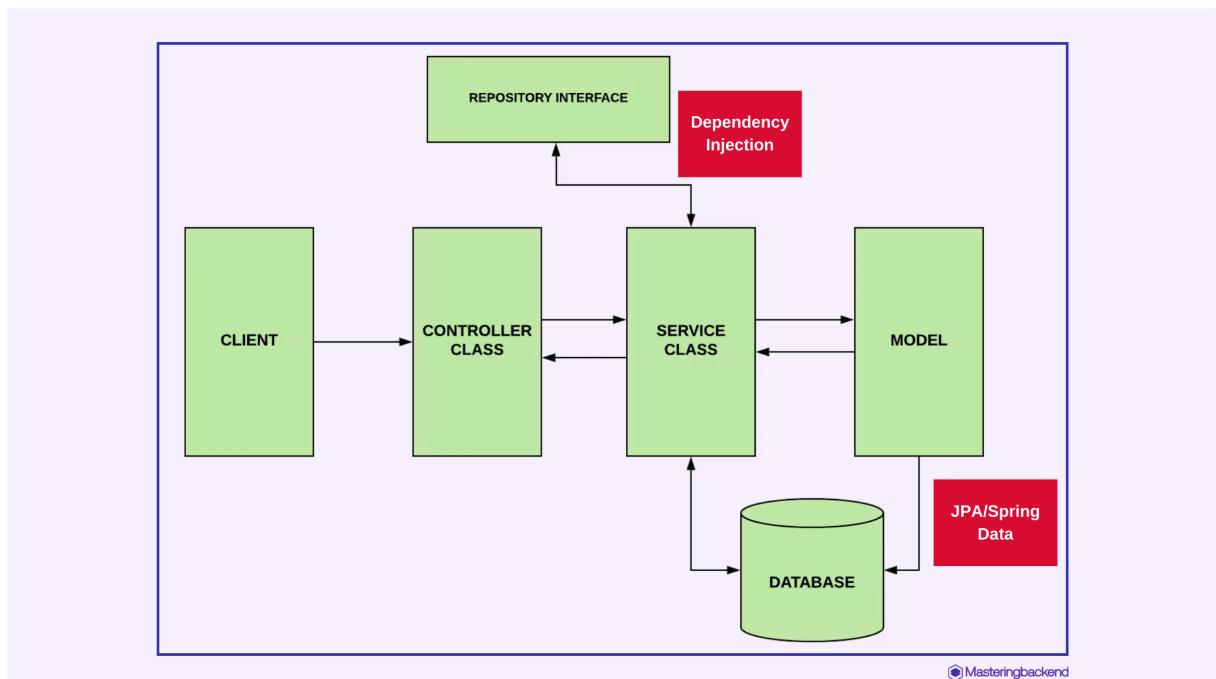
d) Tầng cơ sở dữ liệu (Database Layer)

Tầng cơ sở dữ liệu chứa tất cả các CSDL như: MySQL, MongoDB, Postgres... đúng như tên gọi của nó.

Chịu trách nhiệm thực hiện các hoạt động cơ sở dữ liệu (chủ yếu là các hoạt động CRUD) và đó là cơ sở dữ liệu thực tế mà chúng quyết định sử dụng để xây dựng ứng dụng của mình.

2.3.3.2. Luồng hoạt động của Spring Boot

Spring Boot được xây dựng dựa trên Framework Spring, do đó Spring Boot được cấu trúc bằng cách sử dụng Pattern phổ biến là MVC (Model, View, Controller) nhưng có sửa đổi một chút.



Hình 2.4. Luồng hoạt động của Spring Boot

Ví dụ về luồng hoạt động của Spring Boot từ khi một yêu cầu (Request) đến và đến khi phản hồi (Response) được trả về:

- Đầu tiên, máy khách (Client) tạo một yêu cầu HTTP sử dụng phương thức GET, PUT, POST, DELETE...
- Lớp Controller nhận yêu cầu HTTP.
- Controller hiểu loại yêu cầu nào thì sẽ xử lý yêu cầu đó.
- Nếu cần, Controller sẽ gọi lớp Service.
- Lớp Service sẽ xử lý logic nghiệp vụ. Nó thực hiện điều này trên dữ liệu từ cơ sở dữ liệu.
- Nếu mọi xử lý đã xong, nó sẽ trả về phản hồi dưới dạng trang JSP hoặc JSON.

2.4. Công nghệ Reactjs

React.js là một thư viện JavaScript mã nguồn mở được phát triển bởi Facebook vào năm 2013. React được thiết kế nhằm xây dựng các giao diện người dùng (UI) linh hoạt, có hiệu năng cao và dễ bảo trì. [6]

2.4.1. Đặc điểm kỹ thuật của React.js

Virtual DOM (Document Object Model): React sử dụng một cơ chế gọi là Virtual DOM để theo dõi sự thay đổi trong giao diện. Thay vì cập nhật trực tiếp DOM thực, React so sánh trạng thái mới với trạng thái cũ và chỉ thay đổi những phần cần thiết. Điều này cải thiện hiệu suất đáng kể so với việc thao tác DOM trực tiếp.

Component-Based Architecture (Kiến trúc dựa trên thành phần): React phân chia ứng dụng thành các thành phần nhỏ, độc lập, có thể tái sử dụng. Điều này giúp mã nguồn dễ quản lý và dễ mở rộng.

Unidirectional Data Flow (Luồng dữ liệu một chiều): Dữ liệu trong React được truyền từ trên xuống thông qua các thành phần, tạo ra một quy trình dữ liệu dễ kiểm soát và giúp phát hiện lỗi nhanh chóng.

JSX (JavaScript XML): React sử dụng cú pháp JSX cho phép kết hợp mã JavaScript với HTML, giúp việc viết mã giao diện trở nên tự nhiên và dễ đọc.

2.4.2. Ưu điểm của React.js

Hiệu năng cao: sử dụng cơ chế Virtual DOM giúp giảm thiểu chi phí cập nhật giao diện.

Tái sử dụng mã nguồn: Các component có thể được tái sử dụng nhiều lần, giúp tiết kiệm thời gian phát triển.

Hệ sinh thái phong phú: Có rất nhiều thư viện và công cụ hỗ trợ React như Redux, React Router, Ant Design.

Tính năng cập nhật và bảo trì tốt: Kiến trúc component đảm bảo rằng các phần của ứng dụng có thể được cập nhật độc lập.

2.4.3. Hạn chế của React.js

Reactjs là một thư viện UI: chỉ xử lý giao diện, cần kết hợp thêm các thư viện khác như React Router, Redux để xây dựng ứng dụng hoàn chỉnh.

Yêu cầu học JSX: JSX là một cú pháp mới, có thể gây khó khăn cho người mới bắt đầu.

2.5. Tổng quan về MySQL

2.5.1. MySQL là gì?

MySQL là một mã nguồn mở, hệ thống quản lý cơ sở dữ liệu quan hệ - Relational Database Management System (RDBMS) có hiệu quả trong việc lưu trữ, sắp xếp và truy xuất thông tin. Được phát triển bởi Oracle và dựa trên Ngôn ngữ truy vấn có cấu trúc - Structured Query Language (SQL), được sử dụng phổ biến trong các ứng dụng Web, cổng mua sắm và ứng dụng dữ liệu lớn (Big Data). [7]

2.5.2. Những tính năng chính của MySQL

2.5.2.1. Mã nguồn mở (Open source)

MySQL là phần mềm nguồn mở cho phép người dùng tải xuống MySQL và bắt đầu sử dụng nó mà không mất bất kỳ khoản phí nào. Nó có Giấy phép công cộng ((GPL) GNU. Nó mang lại sự tự do cho các dự án nguồn mở và làm cho MySQL trở nên phù hợp đối với các doanh nghiệp quan tâm đến ngân sách phát triển.

2.5.2.2. Hỗ trợ nhiều nền tảng (Supported Platforms)

MySQL hỗ trợ tất cả các nền tảng hệ điều hành chính bao gồm Oracle Linux, Solaris, Ubuntu, SUSE, Debian, Windows và macOS. Nó cũng hỗ trợ các ngôn ngữ và trình điều khiển phổ biến như Perl, Ruby, Go, Rust, C, C++, C# và ODBC.

2.5.2.3. Cơ sở dữ liệu quan hệ (Relational database)

Dữ liệu trong MySQL được sắp xếp theo mô hình quan hệ và được lưu trữ trong các bảng, mỗi hàng được liên kết với một bảng ghi liên quan. Các phương pháp tiếp cận có cấu trúc đảm bảo rằng dữ liệu này là thống nhất, do đó giúp dễ dàng truy xuất và xử lý thông tin.

2.5.2.4. Ngôn ngữ truy vấn có cấu trúc (SQL)

Ngôn ngữ chính được MySQL sử dụng để giao tiếp với CSDL là SQL. Người dùng có thể tạo, đọc, cập nhật hoặc xóa dữ liệu bằng các lệnh SQL để thực hiện công việc phát triển và quản CSDL.

2.5.2.5. Khả năng mở rộng (Scalability)

MySQL phù hợp cho các ứng dụng cơ sở dữ liệu quy mô nhỏ. Chúng ta có thể mở rộng quy mô theo chiều dọc (thêm nhiều tài nguyên hơn như CPU, Bộ nhớ, Đĩa và I/O) và định cấu hình các cụm để chia tỷ lệ theo chiều ngang cho các yêu cầu ứng dụng của bạn. Các công ty như Facebook và X (Twitter) sử dụng hệ thống cơ sở dữ liệu MySQL để xử lý khối lượng công việc của người dùng ở quy mô lớn.

2.5.2.6. Tính linh hoạt (Flexibility)

MySQL cho phép các nhà phát triển sử dụng SQL truyền thống hoặc NoSQL. Nó cũng hỗ trợ dữ liệu quan hệ và tài liệu JSON trong cùng một CSDL cũng như trong cùng một ứng dụng.

2.5.2.7. Hiệu suất (Performance)

Hiệu suất CSDL là điều cần thiết khi chọn một hệ thống quản lý cơ sở dữ liệu. MySQL bao gồm tối ưu hóa lưu trữ dữ liệu, tăng khả năng mở rộng, truy xuất dữ liệu nâng cao... Nó cung cấp nhiều cơ chế khác nhau cho các chuyên gia CSDL để đảm bảo hiệu suất được tối ưu hóa nhằm tối đa hóa hiệu suất hệ thống.

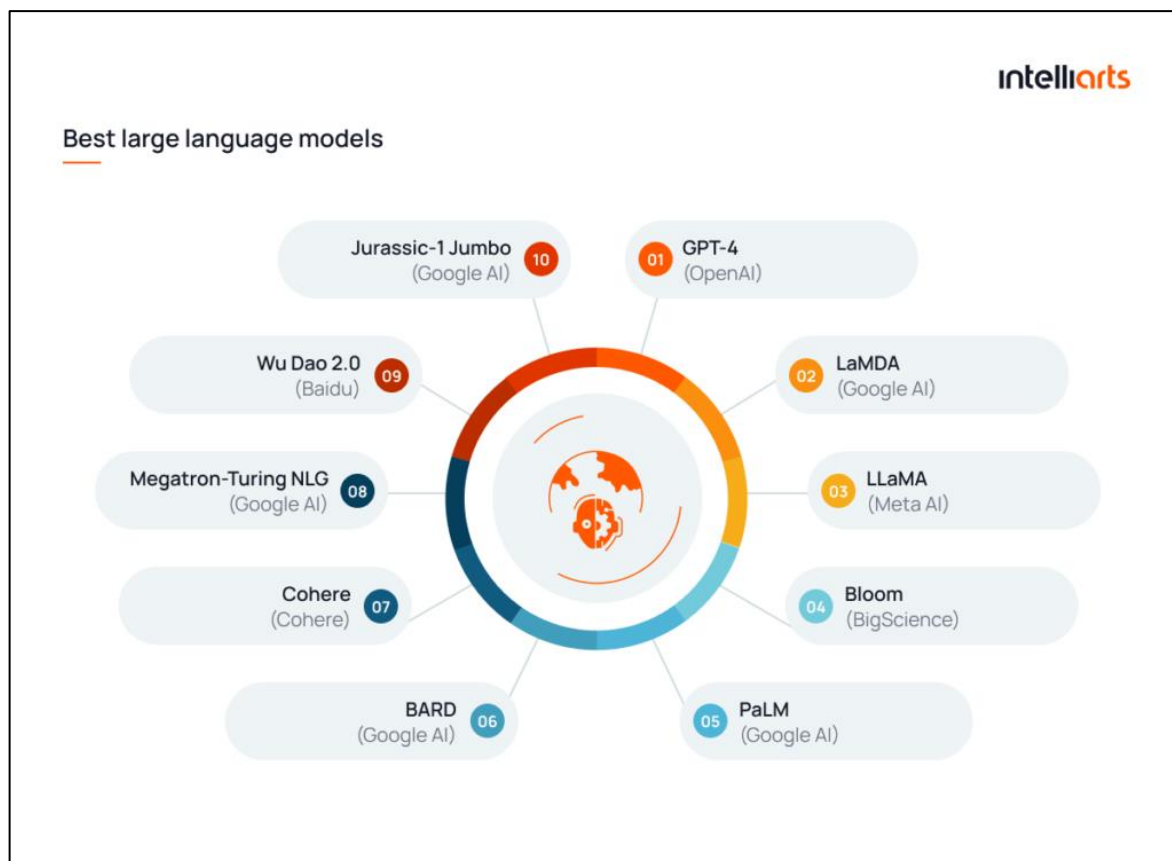
2.5.2.8. Cộng đồng và hỗ trợ (Community and support)

MySQL có cộng đồng người dùng và nhà phát triển cơ sở dữ liệu hoạt động tích cực lớn. Chúng ta có thể sử dụng diễn đàn MySQL để thảo luận về các chủ đề và vấn đề nhằm giúp chúng ta khắc phục sự cố.

2.6. Kiến trúc Retrieval Augmented Generation (RAG)

2.6.1. Tổng quan về mô hình ngôn ngữ lớn

Với sự phát triển mạnh mẽ của lĩnh vực trí tuệ nhân tạo (AI), các mô hình ngôn ngữ lớn (Large Language Models – LLM) như GPT-3, GPT-4, GPT-4o, Gemini, Llama, Claude và các mô hình tương tự đã trở thành những công cụ mạnh mẽ trong việc xử lý ngôn ngữ tự nhiên (Natural Language Processing – NLP). Những mô hình này được huấn luyện trên khối lượng dữ liệu khổng lồ, từ nhiều nguồn văn bản khác nhau như sách, bài báo, trang web, và nội dung trực tuyến. Nhờ vào kiến trúc Transformer và khả năng học từ rất nhiều ngữ cảnh, các LLM có thể tạo ra các văn bản, trả lời câu hỏi, tóm tắt văn bản và thực hiện nhiều tác vụ NLP phức tạp mức độ tự nhiên như con người.



Hình 2.5. Một vài LLM nổi tiếng

Tuy nhiên, các mô hình LLM gặp phải một số hạn chế lớn như:

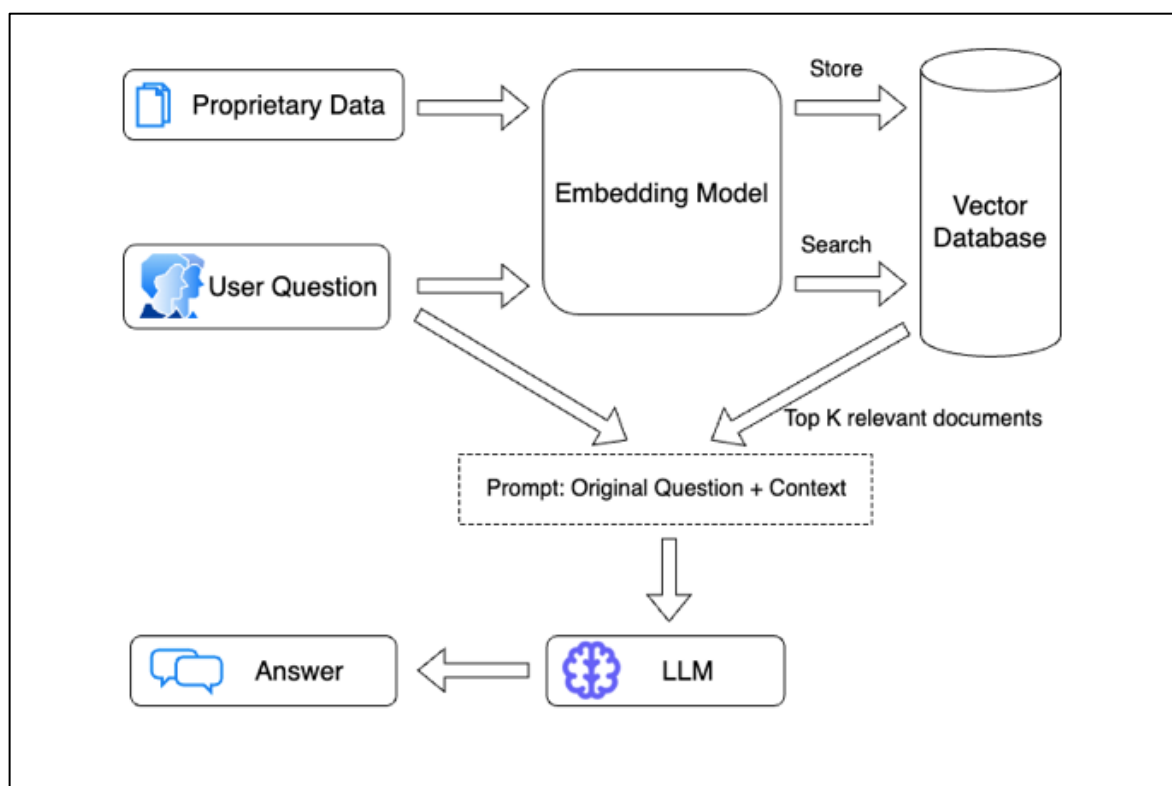
Kiến thức tĩnh và lỗi thời: Các mô hình LLM không thể cập nhật kiến thức sau khi được huấn luyện. Kiến thức của chúng chỉ bao gồm những gì đã học được trong dữ liệu huấn luyện và không được cập nhật thường xuyên. Do đó, các sự kiện, thông tin hoặc kiến thức mới xuất hiện sau thời điểm huấn luyện sẽ không được phản ánh trong câu trả lời của mô hình. Điều này dẫn đến việc cung cấp thông tin lỗi thời hoặc không chính xác khi người dùng hỏi về các sự kiện mới.

Hiện tượng ảo giác (Hallucination) hay đưa ra thông tin không chính xác: Một vấn đề lớn của các LLM là khả năng tự tạo ra thông tin không chính xác hoặc vô nghĩa (hallucination). Khi được hỏi về những điều mà chúng không chắc chắn hoặc không biết (không có trong dữ liệu huấn luyện), LLM thường sẽ đưa ra câu trả lời được tạo sinh một cách tự tin, nhưng lại không có căn cứ thực tế.

2.6.2. Tổng quan về Retrieval Augmented Generation (RAG)

Retrieval-Augmented Generation (RAG) là một kiến trúc kết hợp giữa hai thành phần: truy xuất thông tin (retrieval) và tạo sinh ngôn ngữ (generation). RAG hoạt động bằng cách tìm kiếm thông tin liên quan từ một tập dữ liệu bên ngoài trước khi sử dụng mô hình ngôn ngữ lớn để tạo ra câu trả lời dựa trên thông tin đã truy xuất. Nhờ cơ chế

này, RAG không chỉ dựa vào kiến thức đã được huấn luyện mà còn tận dụng thông tin cập nhật và cụ thể từ các nguồn bên ngoài để cải thiện chất lượng câu trả lời, giảm thiểu sai lệch và cung cấp câu trả lời cập nhật dựa trên dữ liệu từ bên ngoài.



Hình 2.6. Mô hình RAG cơ bản

RAG đã khắc phục được những vấn đề của các LLM:

Cung cấp thông tin cập nhật và chính xác: Nhờ khả năng truy xuất thông tin từ nguồn dữ liệu bên ngoài, RAG có thể trả lời các câu hỏi dựa trên thông tin mới nhất, ngay cả khi thông tin đó không có trong dữ liệu huấn luyện ban đầu của mô hình. Điều này giúp tránh tình trạng cung cấp thông tin lỗi thời.

Giảm thiểu tình trạng sinh thông tin sai lệch: Bằng cách tham khảo trực tiếp các đoạn văn bản liên quan từ cơ sở dữ liệu, RAG giúp mô hình tạo câu trả lời dựa trên bằng chứng thực tế, giảm thiểu khả năng sinh ra thông tin không có căn cứ (hallucination).

Mở rộng phạm vi kiến thức: RAG không bị giới hạn bởi khả năng lưu trữ của LLM. Khi gặp câu hỏi yêu cầu thông tin chi tiết hoặc chuyên sâu, RAG có thể truy xuất từ cơ sở dữ liệu lớn để cung cấp câu trả lời chính xác và cụ thể hơn.

Không cần tái huấn luyện mô hình thường xuyên: Thay vì phải tái huấn luyện LLM mỗi khi cập nhật kiến thức mới, RAG chỉ cần cập nhật cơ sở dữ liệu hoặc tài liệu

tham chiếu. Điều này tiết kiệm chi phí và thời gian so với việc liên tục tái huấn luyện một mô hình lớn.

Nâng cao độ tin cậy và minh bạch: Khi cần thiết, RAG có thể cung cấp nguồn tham chiếu hoặc trích dẫn từ các đoạn văn bản được truy xuất, giúp người dùng dễ dàng kiểm tra và xác minh thông tin.

Tuy vậy nhưng RAG không thay thế các mô hình LLM mà bổ sung thêm khả năng truy xuất dữ liệu từ bên ngoài, giúp cải thiện độ chính xác, tính cập nhật và khả năng cung cấp thông tin cụ thể. Với sự kết hợp này, RAG trở thành một giải pháp hiệu quả cho các ứng dụng chatbot, hệ thống hỏi đáp và các dịch vụ xử lý ngôn ngữ tự nhiên có sử dụng thông tin cơ mật.

2.6.3. Cấu tạo của RAG

Kiến trúc của RAG gồm 4 phần chính: Bộ chuyển đổi tri thức thành vector, bộ truy xuất thông tin (Retriever), bộ tạo văn bản (Generator) và cơ chế kết hợp.

Bộ chuyển đổi tri thức thành vector: Bộ phận này có nhiệm vụ chuyển đổi các dữ liệu tri thức thành các vector và lưu trữ chúng vào một vector database.

Bộ truy xuất thông tin (Retriever): Bộ phận này có nhiệm vụ tìm kiếm và lấy ra các tài liệu hoặc đoạn văn bản liên quan từ một tập dữ liệu lớn (ví dụ: cơ sở kiến thức, kho dữ liệu văn bản). Sử dụng các kỹ thuật như embedding và phương pháp tìm kiếm gần nhất (Nearest Neighbor Search) để tìm ra thông tin phù hợp nhất với câu hỏi.

Bộ tạo văn bản (Generator): Thường là một mô hình ngôn ngữ lớn (ví dụ: GPT, Llama...), sử dụng thông tin do bộ truy xuất cung cấp để tạo ra câu trả lời mạch lạc và phù hợp với ngữ cảnh.

Cơ chế kết hợp: Kết hợp kết quả từ bộ truy xuất và bộ tạo để tối ưu hóa câu trả lời nhằm đảm bảo rằng thông tin được tạo ra có độ chính xác cao và phản ánh đúng dữ liệu tìm được.

2.6.4. Ưu điểm của RAG

Tăng cường độ chính xác: RAG giúp cung cấp thông tin chính xác và chi tiết hơn nhờ khả năng lấy dữ liệu từ nguồn bên ngoài thay vì chỉ dựa vào kiến thức nội bộ của mô hình ngôn ngữ.

Khả năng cập nhật thông tin: Mô hình có thể cung cấp câu trả lời dựa trên dữ liệu mới nhất mà không cần phải tái huấn luyện toàn bộ mô hình.

Giảm thiểu sai lệch: Bằng cách dựa vào thông tin truy xuất được, RAG giảm thiểu nguy cơ đưa ra thông tin không chính xác hoặc phóng đại.

Hiệu quả trong các câu hỏi cụ thể: Phù hợp với các ứng dụng đòi hỏi câu trả lời chi tiết và dựa trên kiến thức chuyên biệt.

2.6.5. Nhược điểm của RAG

Độ trễ cao: Quá trình truy xuất thông tin và sau đó tạo câu trả lời có thể mất nhiều thời gian hơn so với các mô hình tạo văn bản đơn thuần.

Phụ thuộc vào chất lượng dữ liệu: Nếu nguồn dữ liệu không chính xác hoặc không đầy đủ, kết quả truy xuất và câu trả lời sẽ bị ảnh hưởng.

Phức tạp trong triển khai: Việc tích hợp và tối ưu hóa cả hai thành phần (truy xuất và tạo sinh) đòi hỏi nhiều công sức và kỹ thuật.

Khả năng kết hợp thông tin sai lệch: Mặc dù có cơ chế truy xuất, nhưng bộ tạo văn bản vẫn có thể diễn giải sai thông tin được tìm thấy.

2.7. Vector Database

2.7.1. Từ cơ sở dữ liệu truyền thống đến sự ra đời của cơ sở dữ liệu vector

Trong thời đại big data, thông tin tồn tại dưới nhiều hình thức khác nhau và được chia thành hai loại chính:

Thông tin có cấu trúc là dữ liệu được tổ chức và định dạng theo một mô hình cụ thể, dễ dàng lưu trữ, truy cập và phân tích bằng các hệ thống quản lý cơ sở dữ liệu truyền thống như SQL hoặc NoSQL. Dữ liệu có cấu trúc thường được biểu diễn dưới dạng bảng với các hàng (record) và cột (field), trong đó mỗi cột chứa một loại dữ liệu cụ thể. Các trường dữ liệu trong thông tin có cấu trúc được xác định rõ ràng về kiểu dữ liệu và mối quan hệ giữa chúng nếu có.

Thông tin phi cấu trúc là dữ liệu không tuân theo một mô hình hoặc định dạng cụ thể, không dễ dàng tổ chức trong bảng hay cơ sở dữ liệu truyền thống. Loại dữ liệu này thường có dạng văn bản tự do, hình ảnh, âm thanh, video hoặc các tài liệu có độ phức tạp cao.

Các cơ sở dữ liệu truyền thống như SQL và NoSQL gặp phải một số thách thức lớn khi làm việc với dữ liệu phi cấu trúc. Những dữ liệu phi cấu trúc như văn bản, hình ảnh và âm thanh thường được chuyển thành vector có hàng trăm hoặc hàng nghìn chiều. Các cơ sở dữ liệu truyền thống không được thiết kế để lưu trữ và tìm kiếm hiệu quả trong không gian có độ chiều cao như vậy, dẫn đến hiệu năng tìm kiếm chậm và kém hiệu quả. Ngoài ra, các cơ sở dữ liệu truyền thống chỉ hỗ trợ tìm kiếm chính xác hoặc tìm kiếm theo từ khóa. Chúng không thể thực hiện tìm kiếm tương đồng dẫn đến việc không tối ưu trong việc tìm kiếm và khai thác thông tin.

Những thách thức trên đòi hỏi một giải pháp mới, từ đó dẫn đến sự ra đời của Vector Database, giúp lưu trữ và tìm kiếm vector hiệu quả, nhanh chóng, và đáp ứng được các ứng dụng AI và học máy trong thời đại dữ liệu lớn.

2.7.2. Cơ sở dữ liệu vector (Vector Database)

Vector Database (cơ sở dữ liệu vector) là một loại cơ sở dữ liệu được thiết kế đặc biệt để lưu trữ, quản lý và tìm kiếm các vector dữ liệu có độ chiều cao. Các vector này thường là kết quả của việc chuyển đổi dữ liệu phi cấu trúc như văn bản, hình ảnh, âm thanh thành các biểu diễn số học bằng các mô hình nhúng (embedding model) – một loại mô hình học máy. Mỗi vector biểu diễn các đặc trưng hoặc ý nghĩa của dữ liệu dưới dạng không gian nhiều chiều, cho phép tìm kiếm các vector có độ tương đồng gần nhau trong không gian đó. [8]

Vector Database hỗ trợ các chức năng tìm kiếm tương đồng như và tìm kiếm theo ngữ nghĩa. Điều này giúp nó trở thành công cụ lý tưởng cho các ứng dụng như tìm kiếm văn bản theo ngữ nghĩa, tìm kiếm hình ảnh, hệ thống đề xuất, chatbot sử dụng kiến trúc Retrieval-Augmented Generation (RAG), và các ứng dụng AI khác. Thay vì tìm kiếm dữ liệu dựa trên từ khóa hoặc cấu trúc truyền thống, Vector Database tìm kiếm thông qua việc so sánh khoảng cách giữa các vector (ví dụ: khoảng cách Euclidean hoặc cosine similarity).



Hình 2.7. Một số Vector Database phổ biến

Một số Vector Database phổ biến bao gồm FAISS (Facebook AI Similarity Search), Milvus, Pinecone, Qdrant, Weaviate... Những cơ sở dữ liệu này được tối ưu hóa cho khả năng tìm kiếm nhanh và hiệu quả trong các tập dữ liệu lớn, hỗ trợ xử lý thời gian thực. Vector Database giúp giảm đáng kể thời gian tìm kiếm và nâng cao độ chính xác khi làm việc với dữ liệu có tính chất phi cấu trúc và phức tạp. Trong thời đại AI và học máy phát triển mạnh mẽ, Vector Database đóng vai trò quan trọng trong việc xây

dựng các hệ thống thông minh có khả năng xử lý thông tin theo ngữ nghĩa và ý nghĩa sâu sắc hơn.

2.8. Ngôn ngữ lập trình Python

2.8.1. Giới thiệu về Python

Python là một ngôn ngữ lập trình bậc cao, thông dịch, và đa mục đích, được tạo ra bởi Guido van Rossum và ra mắt lần đầu vào năm 1991. Python nổi tiếng với cú pháp đơn giản, dễ học, và dễ đọc, giúp lập trình viên phát triển phần mềm một cách nhanh chóng và hiệu quả. Python được thiết kế với triết lý "Code rõ ràng hơn là code phức tạp", khiến nó trở thành lựa chọn lý tưởng cho người mới bắt đầu cũng như các lập trình viên giàu kinh nghiệm. [9]



Hình 2.8. Logo ngôn ngữ lập trình Python

Python hỗ trợ nhiều paradigm lập trình, bao gồm lập trình hướng đối tượng (OOP), lập trình hàm, và lập trình thủ tục. Ngôn ngữ này đi kèm với một thư viện chuẩn phong phú (Standard Library) cung cấp nhiều module và hàm có sẵn, giúp giải quyết các tác vụ phổ biến như xử lý chuỗi, làm việc với file, kết nối mạng và quản lý cơ sở dữ liệu. Python cũng có một hệ sinh thái rộng lớn với các thư viện và framework mạnh mẽ như NumPy, Pandas, TensorFlow, Django, và Flask, hỗ trợ phát triển từ khoa học dữ liệu, trí tuệ nhân tạo đến phát triển web và tự động hóa.

Python hoạt động trên nhiều hệ điều hành như Windows, macOS, Linux, và là một ngôn ngữ mã nguồn mở, được cộng đồng phát triển rộng khắp hỗ trợ. Python được sử dụng trong nhiều lĩnh vực như phát triển phần mềm, phân tích dữ liệu, học máy (ML), trí tuệ nhân tạo (AI), tự động hóa, và xử lý ngôn ngữ tự nhiên (NLP). Nhờ tính linh hoạt, mạnh mẽ và khả năng mở rộng, Python đã trở thành một trong những ngôn ngữ lập trình phổ biến nhất hiện nay.

2.8.2. Ưu điểm

Cú pháp đơn giản, dễ học: Cú pháp gần gũi với ngôn ngữ tự nhiên, dễ đọc và viết.

Thư viện phong phú: Hệ sinh thái với nhiều thư viện và framework hỗ trợ đa lĩnh vực (AI, ML, Web, dữ liệu).

Đa nền tảng: Chạy trên hầu hết các hệ điều hành như Windows, macOS, và Linux.

Lập trình đa mục đích: Hỗ trợ nhiều paradigms (hướng đối tượng, hàm, thủ tục).

Cộng đồng lớn: Cộng đồng mạnh mẽ và tài liệu hỗ trợ dồi dào.

Khả năng mở rộng: Có thể tích hợp với các ngôn ngữ khác như C/C++ để tăng hiệu suất.

2.8.3. Nhược điểm

Hiệu suất thấp: Chạy chậm hơn các ngôn ngữ biên dịch như C++ hay Java do là ngôn ngữ thông dịch.

Quản lý bộ nhớ kém: Tiêu tốn nhiều bộ nhớ, không thích hợp cho các ứng dụng cần tối ưu cao.

GIL (Global Interpreter Lock): Giới hạn khả năng xử lý đa luồng trong một số tác vụ.

Hạn chế trong lập trình di động: Không phổ biến trong phát triển ứng dụng di động.

Không phù hợp cho ứng dụng thời gian thực: Thiếu tốc độ xử lý cần thiết cho các hệ thống thời gian thực.

2.9. Ngôn ngữ lập trình Java

2.9.1. Giới thiệu về Java

Java là một ngôn ngữ lập trình hướng đối tượng và đa năng, được phát triển bởi Sun Microsystems do James Gosling khởi xướng vào năm 1995. Hiện tại, Java thuộc sở hữu của Oracle. Ngôn ngữ Java được thiết kế với khẩu hiệu nổi tiếng "Write Once, Run Anywhere" (WORA), nghĩa là các chương trình viết bằng Java có thể chạy trên mọi nền tảng có Java Virtual Machine (JVM). Java được sử dụng rộng rãi trong phát triển phần mềm, ứng dụng web, ứng dụng di động (Android), hệ thống nhúng, và các hệ thống doanh nghiệp. [10]



Hình 2.9. Logo ngôn ngữ lập trình Java

Một số đặc điểm chính của Java như:

Độc lập nền tảng: Java sử dụng JVM để thực thi, giúp chương trình có thể chạy trên mọi hệ điều hành.

Hướng đối tượng: Mọi thứ trong Java được coi là một đối tượng, giúp tăng tính linh hoạt và dễ dàng bảo trì.

Được biên dịch thành bytecode: Mã nguồn Java được biên dịch thành bytecode, có thể thực thi trên JVM thay vì trực tiếp trên phần cứng.

2.9.2. Ưu điểm

Độc lập nền tảng (Platform Independence): Một trong những ưu điểm lớn nhất của Java là khả năng chạy trên nhiều hệ điều hành nhờ JVM.

Hướng đối tượng: Java tuân thủ chặt chẽ các nguyên tắc lập trình hướng đối tượng, giúp việc thiết kế và bảo trì dễ dàng hơn.

Quản lý bộ nhớ tự động: Cơ chế Garbage Collection (thu gom rác) của Java giúp lập trình viên giảm thiểu lỗi khi quản lý bộ nhớ.

Tính ổn định cao: Java thường được sử dụng trong các hệ thống lớn nhờ sự ổn định và khả năng mở rộng tốt.

Thư viện phong phú: Java có sẵn nhiều thư viện hỗ trợ từ các tính năng cơ bản đến các tính năng phức tạp. [11]

2.9.3. Nhược điểm

Độ phức tạp của mã nguồn: Mặc dù mạnh mẽ, Java yêu cầu viết mã dài dòng hơn và cấu trúc phức tạp hơn so với các ngôn ngữ hiện đại như Python hay Kotlin.

Phụ thuộc vào JVM: Chương trình Java cần JVM để chạy, có thể gây ra các vấn đề tương thích phiên bản giữa các hệ điều hành.

Không tối ưu cho phát triển UI hiện đại: Mặc dù có JavaFX và Swing, Java không phải là lựa chọn phổ biến cho giao diện người dùng hiện đại, thay vào đó các công cụ như HTML5 hay React được ưa chuộng hơn.

2.10. Ngôn ngữ lập trình Javascript (Js)

2.10.1. Giới thiệu về Javascript

JavaScript là một ngôn ngữ lập trình phổ biến và mạnh mẽ được phát triển lần đầu vào năm 1995 bởi Brendan Eich tại Hãng truyền thông Netscape với cái tên đầu tiên Mocha, rồi sau đó đổi tên thành LiveScript, và cuối cùng thành JavaScript. Ban đầu, JavaScript được thiết kế để chạy trên trình duyệt và cung cấp khả năng tương tác cho các trang web, như xử lý sự kiện người dùng hoặc thay đổi nội dung động trên giao diện. Ngày nay, JavaScript đã vượt ra khỏi trình duyệt nhờ vào các nền tảng như Node.js, cho phép nó được sử dụng cả trên cả máy chủ server. Đây là một ngôn ngữ kịch bản dựa với các đặc điểm như kiểu động, hướng đối tượng theo mô hình prototype, và hỗ trợ lập trình không đồng bộ qua cơ chế callback, promise hoặc async/await. [12]



Hình 2.10. Logo ngôn ngữ lập trình Js

JavaScript có một số đặc điểm nổi bật như:

Hướng sự kiện và lập trình bất đồng bộ: Hỗ trợ callback, promise và async/await giúp xử lý các tác vụ không đồng bộ hiệu quả.

Lập trình động: Cho phép thay đổi kiểu dữ liệu và thêm thuộc tính vào đối tượng trong thời gian chạy.

Không phụ thuộc nền tảng: JavaScript có thể chạy trên mọi trình duyệt hiện đại và trên máy chủ (Node.js).

Hỗ trợ đa mô hình: Bao gồm lập trình hướng đối tượng, lập trình hàm và lập trình dựa trên sự kiện.

2.10.2. Ưu điểm

Tính phổ biến và hỗ trợ rộng rãi: Hầu hết các trình duyệt hiện đại đều hỗ trợ JavaScript mà không cần cài đặt thêm. Điều này giúp JavaScript trở thành ngôn ngữ lập trình được sử dụng phổ biến nhất trên thế giới.

Tính năng động và linh hoạt: JavaScript có khả năng thực thi mã ngay trên trình duyệt, giúp tạo ra các trang web tương tác và phản hồi nhanh. Ngoài ra, kiểu dữ liệu động giúp tăng tốc quá trình phát triển ứng dụng.

Hệ sinh thái phong phú: JavaScript có một hệ sinh thái thư viện và framework khổng lồ như React, Angular, Vue.js, và Express.js, hỗ trợ từ việc xây dựng giao diện người dùng đến phát triển backend.

Khả năng đa nền tảng: JavaScript có thể được sử dụng để xây dựng ứng dụng web, ứng dụng di động (React Native), ứng dụng desktop (Electron), và thậm chí cả các ứng dụng IoT.

2.10.3. Nhược điểm

Thiếu kiểm tra kiểu chặt chẽ: Do là ngôn ngữ kiểu động, JavaScript không kiểm tra kiểu dữ liệu nghiêm ngặt, dễ dẫn đến lỗi runtime. Điều này thường được khắc phục một phần bằng cách sử dụng TypeScript.

Hiệu năng không cao như ngôn ngữ biên dịch: Mặc dù V8 engine đã cải thiện hiệu năng, JavaScript vẫn không thể so sánh với các ngôn ngữ biên dịch như C++ hoặc Java trong các ứng dụng đòi hỏi tính toán nặng hay hiệu suất cao.

Vấn đề bảo mật: JavaScript dễ bị khai thác cho các cuộc tấn công như Cross-Site Scripting (XSS) hoặc Cross-Site Request Forgery (CSRF) nếu không được quản lý đúng cách.

CHƯƠNG 3: KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU

3.1. Khảo sát hiện trạng

3.1.1. Tầm quan trọng của việc học tiếng Anh

Tiếng Anh được xem là ngôn ngữ toàn cầu và được sử dụng rộng rãi nhất trên thế giới. Đây cũng là ngôn ngữ chung trong các hoạt động giao lưu văn hóa, du lịch, và ngoại giao quốc tế. Tầm quan trọng của việc học Tiếng Anh ngày nay không chỉ nằm ở khả năng giao tiếp mà còn ở việc mở ra cánh cửa tiếp cận tri thức, cơ hội nghề nghiệp, và trải nghiệm văn hóa toàn cầu.

Tiếng Anh là ngôn ngữ của khoa học và công nghệ. Phần lớn các tài liệu khoa học, nghiên cứu và thông tin công nghệ đều sử dụng Tiếng Anh, khiến nó trở thành ngôn ngữ không thể thiếu trong việc cập nhật tri thức. Hơn nữa, các công ty công nghệ lớn cũng dùng Tiếng Anh làm ngôn ngữ chính, giúp người học tiếp cận với các lĩnh vực tiên tiến và xu hướng trên thế giới.

Mặc khác, trong bối cảnh toàn cầu hóa, Tiếng Anh mang lại lợi thế lớn trong việc tìm kiếm việc làm tại các công ty đa quốc gia cũng như cơ hội làm việc ở nước ngoài. Đây là kỹ năng quan trọng để thăng tiến trong sự nghiệp và mở rộng tầm nhìn nghề nghiệp.

Ngoài ra, việc biết tiếng Anh còn phục vụ cho nhu cầu học tập, giải trí và du lịch. Tiếng Anh là ngôn ngữ của nhiều nền tảng giáo dục trực tuyến, sách, và khóa học chuyên môn hàng đầu như Coursera, edX, và Udemy. Hơn nữa, đa số nội dung giải trí quốc tế, bao gồm phim, nhạc, và trò chơi, đều được sản xuất bằng Tiếng Anh. Điều này làm tăng trải nghiệm học tập và giải trí khi biết ngôn ngữ này. Trong du lịch quốc tế, Tiếng Anh là ngôn ngữ phổ biến giúp du khách giao tiếp dễ dàng và tận hưởng trọn vẹn hành trình của mình. Từ đó giúp hiểu biết thêm về các nền văn hóa khác. Việc tiếp cận đa dạng các quan điểm và ý tưởng thông qua Tiếng Anh giúp mở rộng tư duy và khả năng phân tích vấn đề.

Từ những luận điểm trên, có thể kết luận rằng tiếng Anh là chìa khóa mở ra cơ hội trong học tập, công việc, và cuộc sống hiện đại. Từ vai trò là ngôn ngữ giao tiếp toàn cầu đến việc hỗ trợ tiếp cận tri thức, công nghệ, và văn hóa, Tiếng Anh giúp con người kết nối và hòa nhập với thế giới. Việc thành thạo Tiếng Anh không chỉ mang lại lợi thế cạnh tranh trong sự nghiệp mà còn nâng cao tư duy, mở rộng hiểu biết và khám phá những trải nghiệm mới. Đầu tư vào việc học Tiếng Anh là một bước đi cần thiết để phát triển bản thân và tận dụng tối đa các cơ hội mà thế giới hiện đại mang lại.

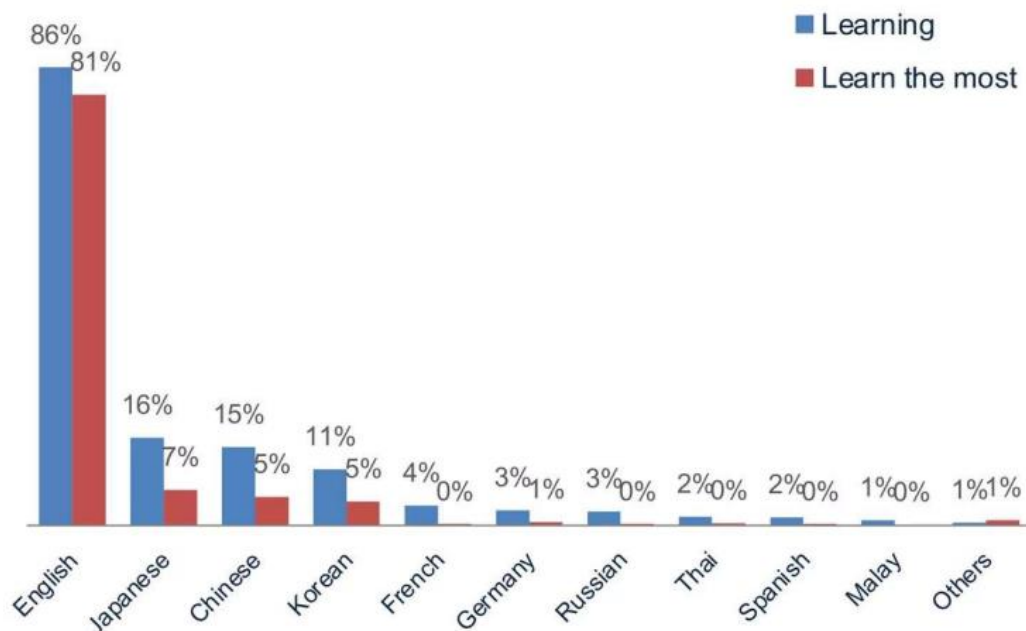
3.1.2. Nhu cầu học tiếng Anh của người Việt Nam

Trong những năm trở lại đây, nhu cầu học ngoại ngữ của người Việt Nam ngày càng tăng cao và phổ biến ở mọi độ tuổi. Trong đó, tiếng Anh là ngôn ngữ được lựa chọn học nhiều nhất so với các ngoại ngữ khác như tiếng Trung, tiếng Nhật, tiếng Hàn...

Từ các số liệu trong “Báo cáo về thói quen học ngoại ngữ của người Việt Nam” của Q&Me tháng 6 năm 2021, tiếng Anh là ngôn ngữ được học nhiều nhất, chiếm 86% tổng số người tham gia, trong đó 81% số người xác định rằng tiếng Anh là ngôn ngữ họ học nhiều nhất. Đây là sự cách biệt lớn so với các ngôn ngữ khác, với tiếng Nhật chiếm 16%, tiếng Trung 15% và tiếng Hàn 11%. Các ngôn ngữ khác như tiếng Pháp, tiếng Đức, tiếng Nga, tiếng Thái, và tiếng Tây Ban Nha chỉ chiếm tỷ lệ dưới 5%, cho thấy sự ưu tiên rõ rệt đối với việc học tiếng Anh đối với người Việt Nam. Phần lớn người học (37%) dành từ 1 đến 3 giờ mỗi tuần cho việc học tiếng Anh và 20% dành từ 3 đến 5 giờ. Trong đó, tự học là phương pháp được sử dụng phổ biến nhất cho việc học tiếng Anh (50%), tiếp theo là sử dụng các khóa học trực tuyến miễn phí (36%). Đối với các ngôn ngữ như tiếng Nhật, tiếng Hàn và tiếng Trung, việc tham gia các trung tâm ngôn ngữ được ưu tiên hơn. Sự phụ thuộc vào các trung tâm ngôn ngữ có thể do thiếu nguồn tài liệu miễn phí hoặc yêu cầu hướng dẫn cụ thể đối với các ngôn ngữ phức tạp. Từ đó cho thấy, tiếng Anh là ngôn ngữ dễ học và dễ tiếp cận nhất hiện nay. [13]

Popular foreign language

English is the most learned language with 86%, followed by Japanese 16%, Chinese 15% and Korean 11%.



Hình 3.1. Các ngoại ngữ phổ biến được chọn học (Ảnh trích từ báo cáo)

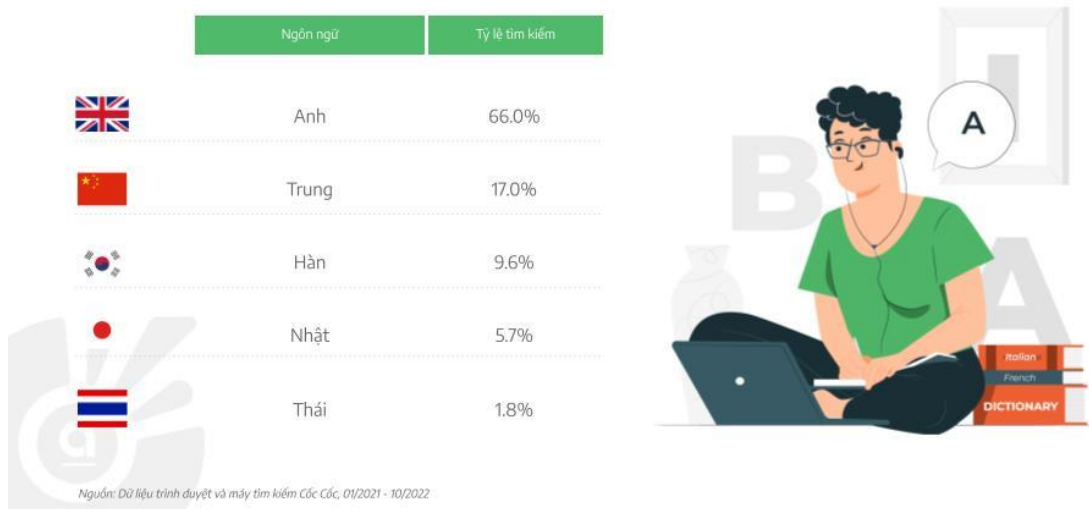
Ngoài ra, khi được hỏi về lý do học ngoại ngữ, 60% người tham gia khảo sát cho biết họ học để "đạt được công việc tốt". Đây là động lực mạnh mẽ nhất, theo sau là việc ngôn ngữ cần thiết trong công việc (49%). Những lý do liên quan đến giao tiếp, như mong muốn giao tiếp với người nước ngoài (44%) hoặc du lịch (32%), cũng được đề cập, nhưng ít phổ biến hơn. Điều này nhấn mạnh rằng việc học ngoại ngữ không chỉ để thỏa mãn sở thích cá nhân mà còn mang tính thiết yếu trong môi trường lao động toàn cầu hóa.

Theo “Báo cáo thị trường – Nhu cầu học tiếng Anh của người lớn” của Cốc Cốc ngày 15 tháng 11 năm 2022, trong số 5 ngôn ngữ được tìm kiếm nhiều trên Cốc Cốc thì Tiếng Anh là ngôn ngữ có lượng tìm kiếm nhiều nhất, chiếm 66% các ngôn ngữ được tìm kiếm. Trong đó, lượng tìm kiếm tiếng Anh cho sinh viên nhiều gấp 1,6 lần lượng tìm kiếm tiếng Anh cho người đi làm. Người dùng tìm kiếm những kiến thức tiếng Anh nhằm hai mục đích chính là giao tiếp và phục vụ cho công việc. Về hình thức học, đa số người dùng sẽ sử dụng các kênh online như các website, các phần mềm, ứng dụng... để học tập và rèn luyện kỹ năng tiếng Anh của mình. Trong đó từ vựng có lượng cầu cao nhất, cứ 4 người thì có 3 người có nhu cầu học từ vựng. [14]

Ngôn ngữ được người dùng quan tâm



Trong số 5 ngôn ngữ được tìm kiếm nhiều trên Cốc Cốc thì **Tiếng Anh** là **ngôn ngữ có lượng tìm kiếm nhiều nhất**, chiếm 66% các ngôn ngữ được tìm kiếm. Theo sau đó là các ngôn ngữ khu vực Châu Á: Trung, Hàn, Nhật, Thái.



Hình 3.2. Các ngôn ngữ được tìm kiếm nhiều nhất trên Cốc Cốc

Từ những số liệu khảo sát nêu trên cho thấy tiếng Anh giữ vai trọng, là ngôn ngữ được học nhiều nhất ở Việt Nam nhờ vào tính thông dụng của nó. Trong quá trình học tiếng Anh, việc nắm vững từ vựng hay cải thiện vốn từ rất quan trọng vì từ vựng là động lực thúc đẩy sự tiến bộ các kỹ năng nghe – nói – đọc – viết của người học. Đa số người học tiếng Anh sẽ lựa chọn phương pháp tự học và sử dụng tài nguyên trực tuyến miễn phí. Các phương pháp học từ vựng truyền thống như ghi chép, học qua sách giáo khoa, và tham gia các lớp học đã bộc lộ một số hạn chế như thiếu tính tương tác và khó duy trì động lực học tập. Hiện nay, các giải pháp công nghệ, bao gồm ứng dụng học từ vựng và chatbot hỗ trợ, đã và đang nhận được nhiều sự quan tâm. Tuy nhiên, chưa có nhiều nền tảng kết hợp hiệu quả giữa việc học từ vựng và các tính năng tương tác thông minh.

Dự án "Xây dựng Website học từ vựng Tiếng Anh tích hợp Chatbot" ra đời nhằm khắc phục những hạn chế nêu trên. Website sẽ tập trung vào việc cung cấp các phương pháp học từ vựng hiệu quả, cá nhân hóa trải nghiệm học tập đồng thời tích hợp chatbot như một công cụ hỗ trợ nhằm tăng cường tính tương tác và tiện lợi cho người học.

3.2. Phân tích yêu cầu

3.2.1. Yêu cầu chức năng

- Cho phép người dùng đăng nhập vào hệ thống.
- Cho phép người dùng cập nhật thông tin cá nhân.
- Cho phép người dùng tạo, truy cập, học và ôn luyện các nội dung, tài nguyên trên hệ thống.

- Cho phép người quản lý (admin) quản lý người dùng: xem, chặn.

3.2.2. Yêu cầu phi chức năng

Yêu cầu về chất lượng

- Tính cập nhật – phát triển:

- + Dễ dàng nâng cấp hoặc thêm các tính năng tiện ích.
- + Bảo trì hệ thống.

- Tính tiện dụng:

- + Có các đặc tả và hướng dẫn rõ ràng, dễ dàng thao tác
- + Hoạt động ổn định, đáng tin cậy

- Tính tương thích:

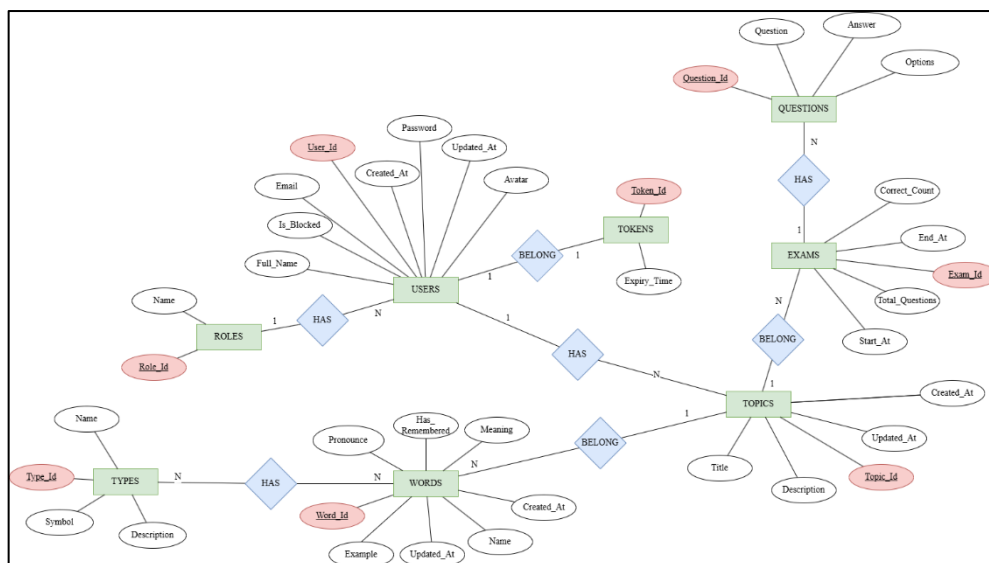
- + Hoạt động tốt trên tất cả các phương tiện phổ biến hiện nay.

Yêu cầu về giao diện

- Thân thiện, dễ sử dụng
- Hiển thị đầy đủ và chi tiết các thông tin, tính nhất quán.

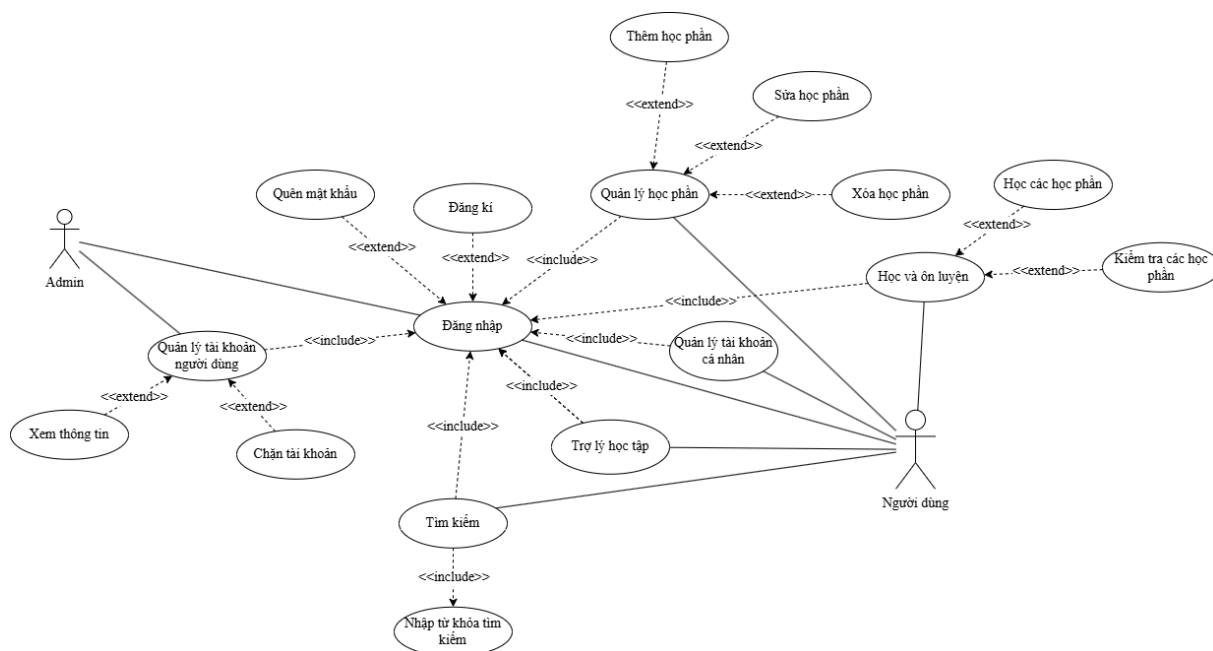
CHƯƠNG 4: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

4.1. Sơ đồ ER



Hình 4.1. Sơ đồ ER

4.2. Sơ đồ use case tổng quát



Hình 4.2. Use case tổng quát

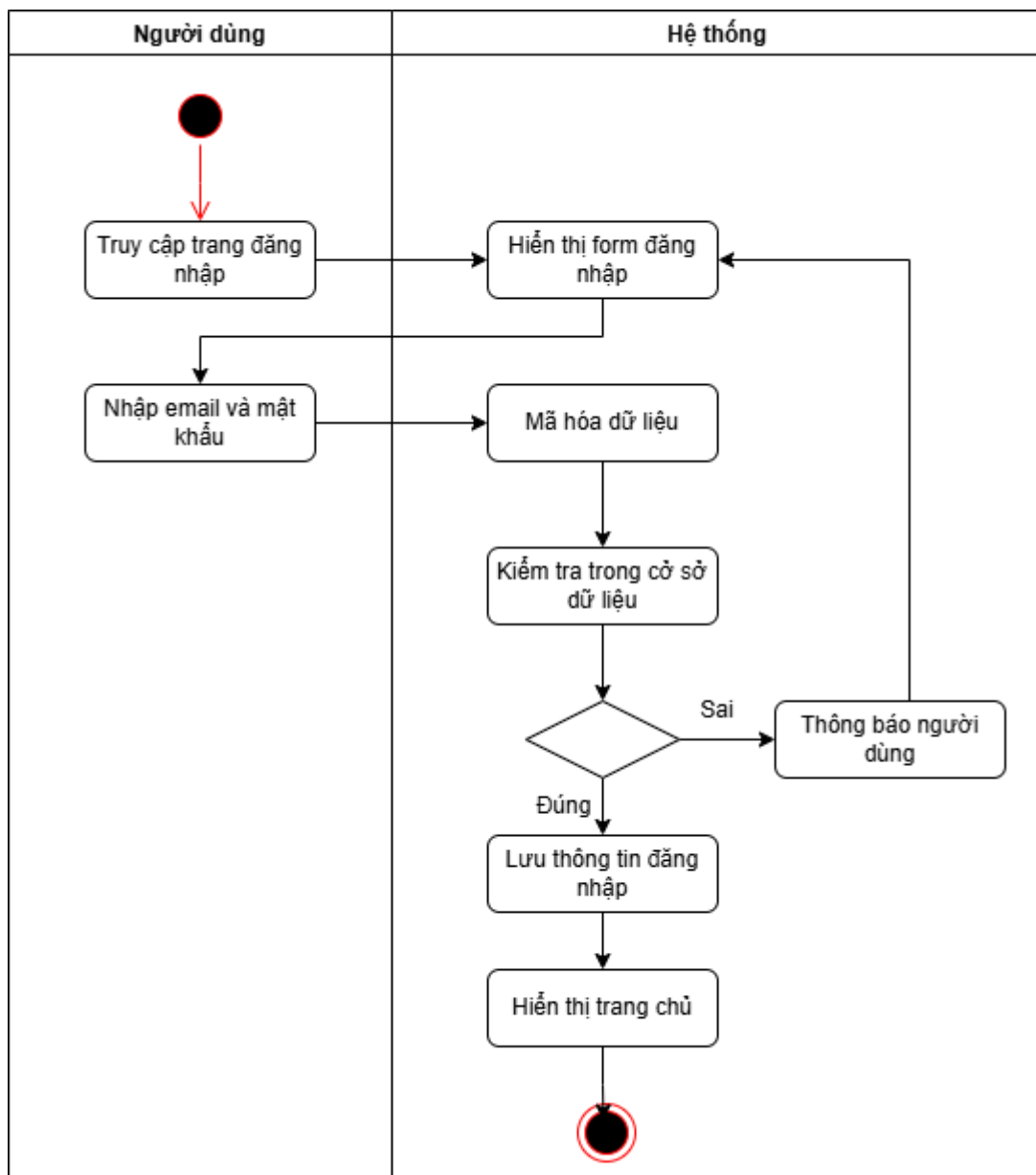
4.3. Đặc tả một số use case chính

4.3.1. Chức năng đăng nhập

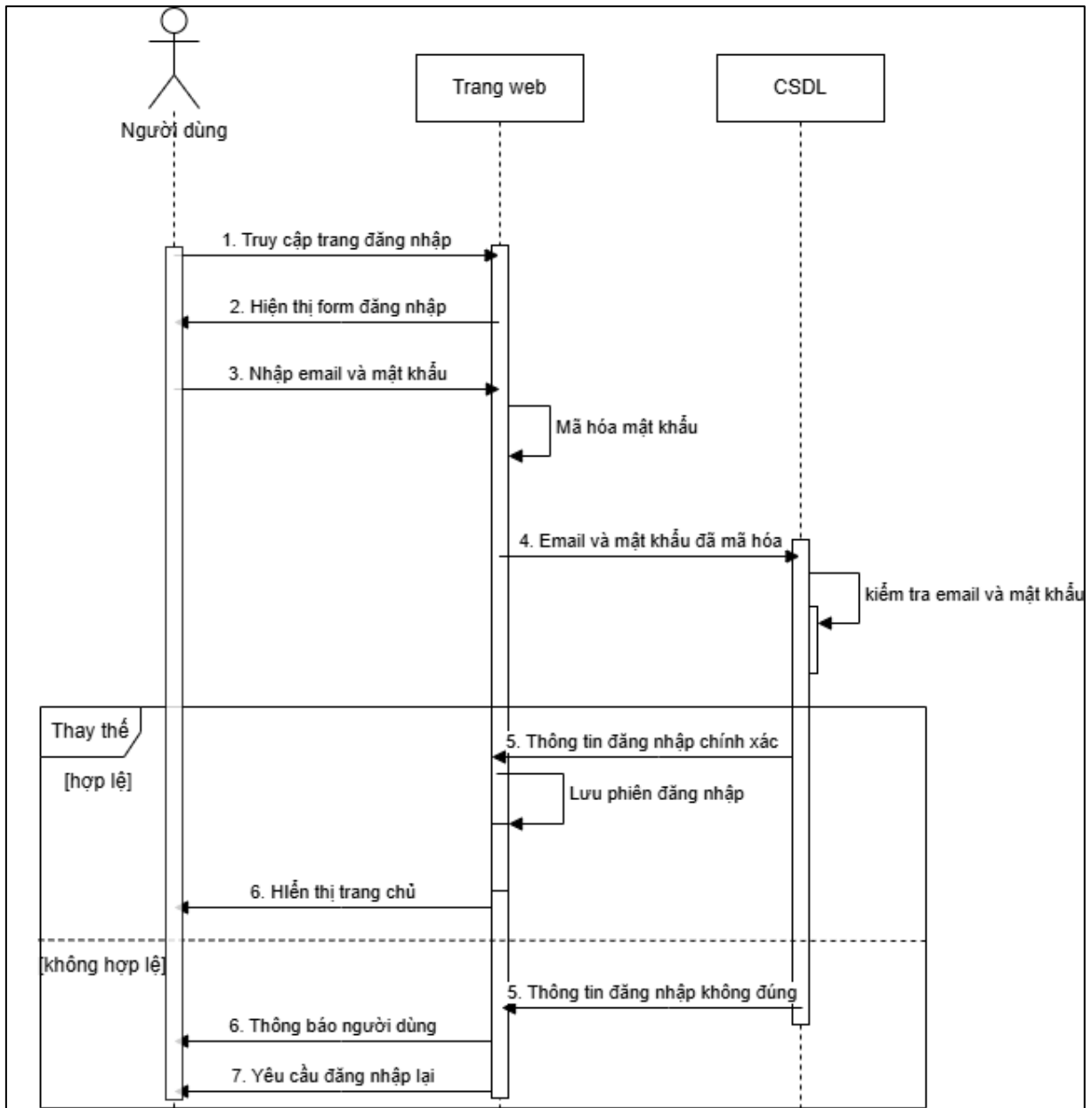
Tên Use case	Đăng nhập
--------------	-----------

Tên tác nhân	Admin, Người dùng
Điều kiện tiên quyết	Đã đăng xuất khỏi hệ thống
Đảm bảo tối thiểu	Bỏ thông tin đăng nhập và quay về trạng thái ban đầu
Đảm bảo thành công	Đăng nhập thành công vào hệ thống
Kích hoạt	Truy cập vào hệ thống
Chuỗi sự kiện chính	<ul style="list-style-type: none"> - Nhập email và mật khẩu. - Hệ thống mã hóa mật khẩu và đối chiếu với cơ sở dữ liệu. - Nếu thông tin chính xác, người dùng sẽ được chuyển đến trang chủ của hệ thống. - Ca sử dụng kết thúc
Ngoại lệ	<ul style="list-style-type: none"> - Đăng nhập thất bại - Hệ thống thông báo sai thông tin đăng nhập - Người dùng đăng nhập lại hệ thống.

Bảng 4.1. Đặc tả use case Đăng nhập



Hình 4.3. Sơ đồ hoạt động use case Đăng nhập



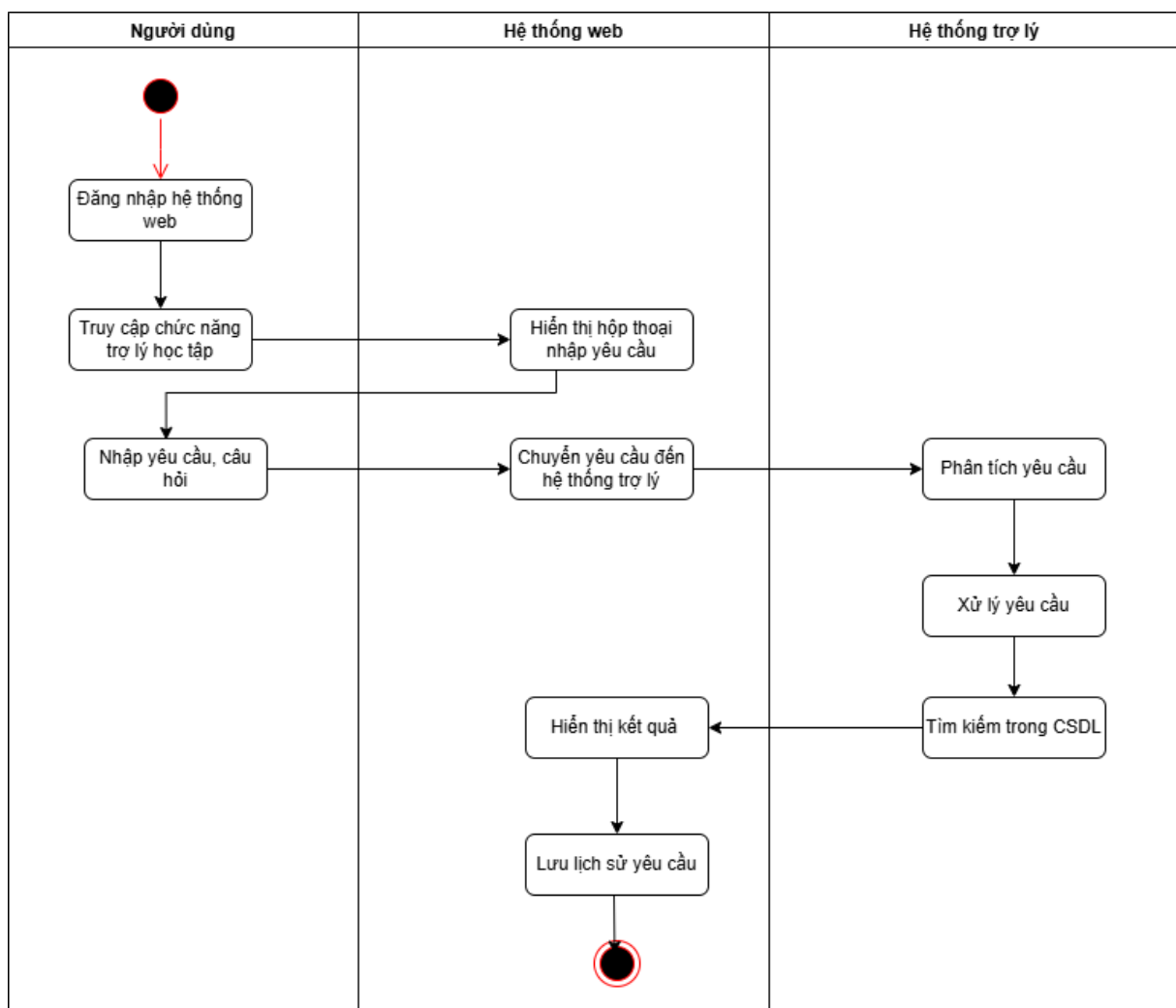
Hình 4.4. Sơ đồ tuần tự use case Đăng nhập

4.3.2. Chức năng trợ lý học tập

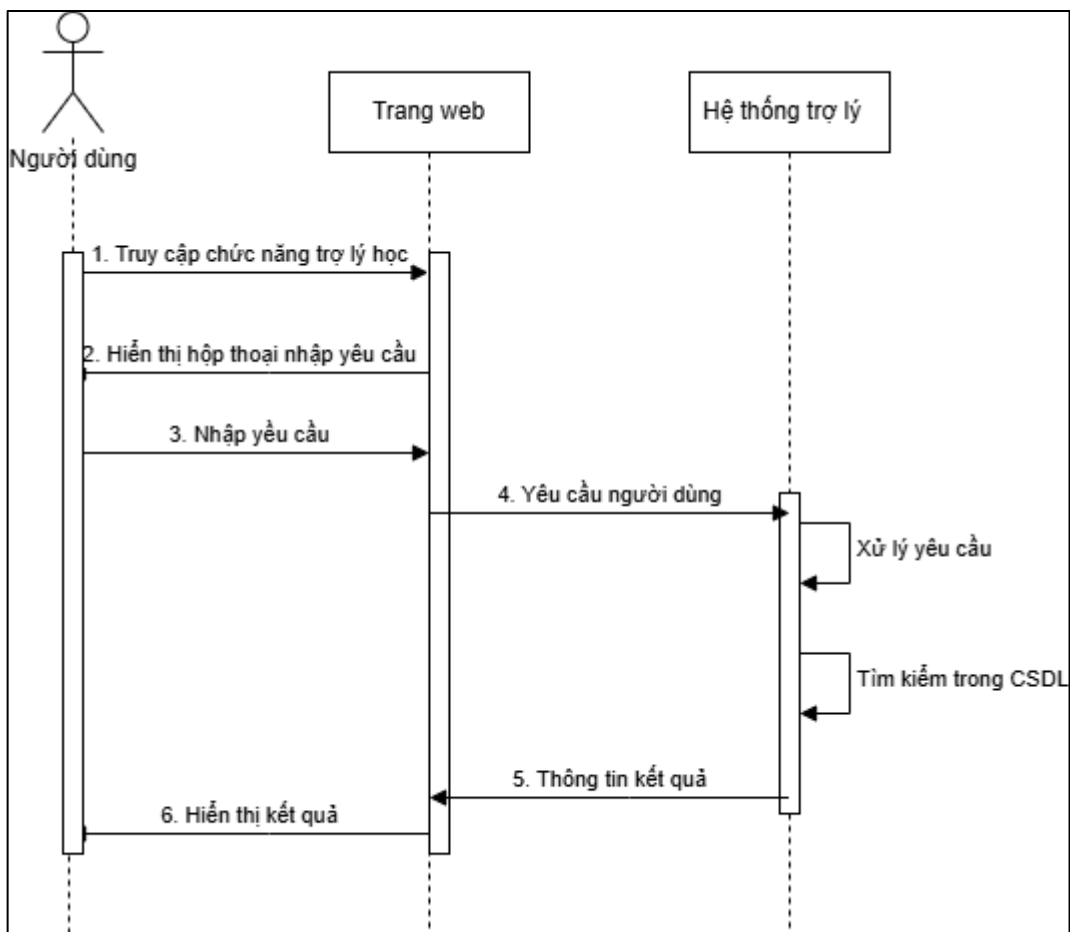
Tên Use case	Trợ lý học tập
Tên tác nhân	Người dùng
Điều kiện tiên quyết	<ul style="list-style-type: none"> - Đã đăng nhập vào hệ thống - Chức năng Trợ lý học tập sẵn sàng hoạt động. - Cơ sở dữ liệu từ vựng, ngữ pháp được tích hợp đầy đủ.
Đảm bảo tối thiểu	<ul style="list-style-type: none"> - Hệ thống nhận diện và phân tích đúng yêu cầu cơ bản của người dùng. - Phản hồi nhanh đối với các câu hỏi đơn giản

	- Giao diện thân thiện, dễ sử dụng
Đảm bảo thành công	<ul style="list-style-type: none"> - Người dùng nhận được phản hồi chính xác, dễ hiểu, và phù hợp với trình độ. - Các yêu cầu được lưu trữ vào lịch sử
Kích hoạt	Người dùng truy cập vào chức năng Trợ lý học tập trên giao diện chính và nhập câu hỏi/yêu cầu
Chuỗi sự kiện chính	<ul style="list-style-type: none"> - Nhập yêu cầu, câu hỏi - Hệ thống xử lý yêu cầu - Phản hồi đáp án cho người dùng - Kết thúc ca sử dụng
Ngoại lệ	<ul style="list-style-type: none"> - Đăng nhập thất bại - Hệ thống trợ lý lỗi - Yêu cầu không rõ ràng - Dữ liệu không tồn tại

Bảng 4.2. Đặc tả use case Trợ lý học tập



Hình 4.5. Sơ đồ hoạt động use case Trợ lý học tập



Hình 4.6. Sơ đồ tuần tự use case Trợ lý học tập

4.3.3. Chức năng quản lý học phần

Tên Use case	Quản lý học phần
Tên tác nhân	Người dùng
Điều kiện tiên quyết	Đã đăng nhập vào hệ thống
Đảm bảo tối thiểu	<ul style="list-style-type: none"> - Người dùng có thể thêm, sửa, xóa, và xem danh sách các học phần. - Hệ thống kiểm tra tính hợp lệ của thông tin nhập. - Hệ thống hiển thị thông báo lỗi nếu dữ liệu không hợp lệ hoặc không thể thực hiện thao tác.
Đảm bảo thành công	<ul style="list-style-type: none"> - Học phần được thêm mới, cập nhật, hoặc xóa thành công và lưu lại trong cơ sở dữ liệu. - Danh sách học phần được hiển thị rõ ràng, chính xác. - Người dùng nhận được thông báo xác nhận khi thực hiện các thao tác.

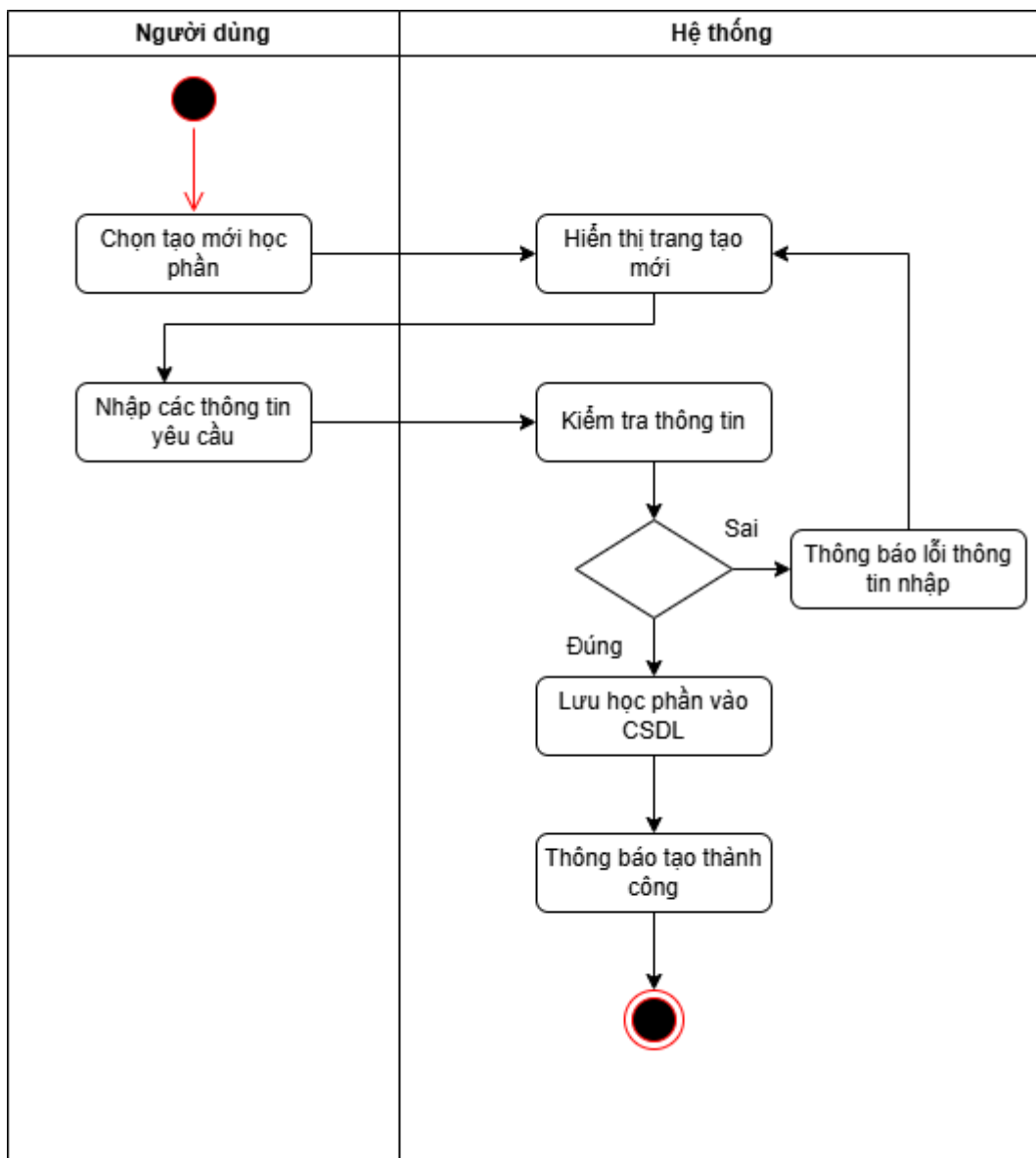
Kích hoạt	Đăng nhập vào hệ thống, chọn vào mục học phần
Chuỗi sự kiện chính	<ul style="list-style-type: none"> - Đăng nhập vào hệ thống - Chọn mục học phần - Hệ thống hiện những học phần hiện có - Chọn các lựa chọn - Ca sử dụng kết thúc
Ngoại lệ	<ul style="list-style-type: none"> - Đăng nhập thất bại - Không tìm thấy học phần

Bảng 4.3. Đặc tả use case Quản lý học phần

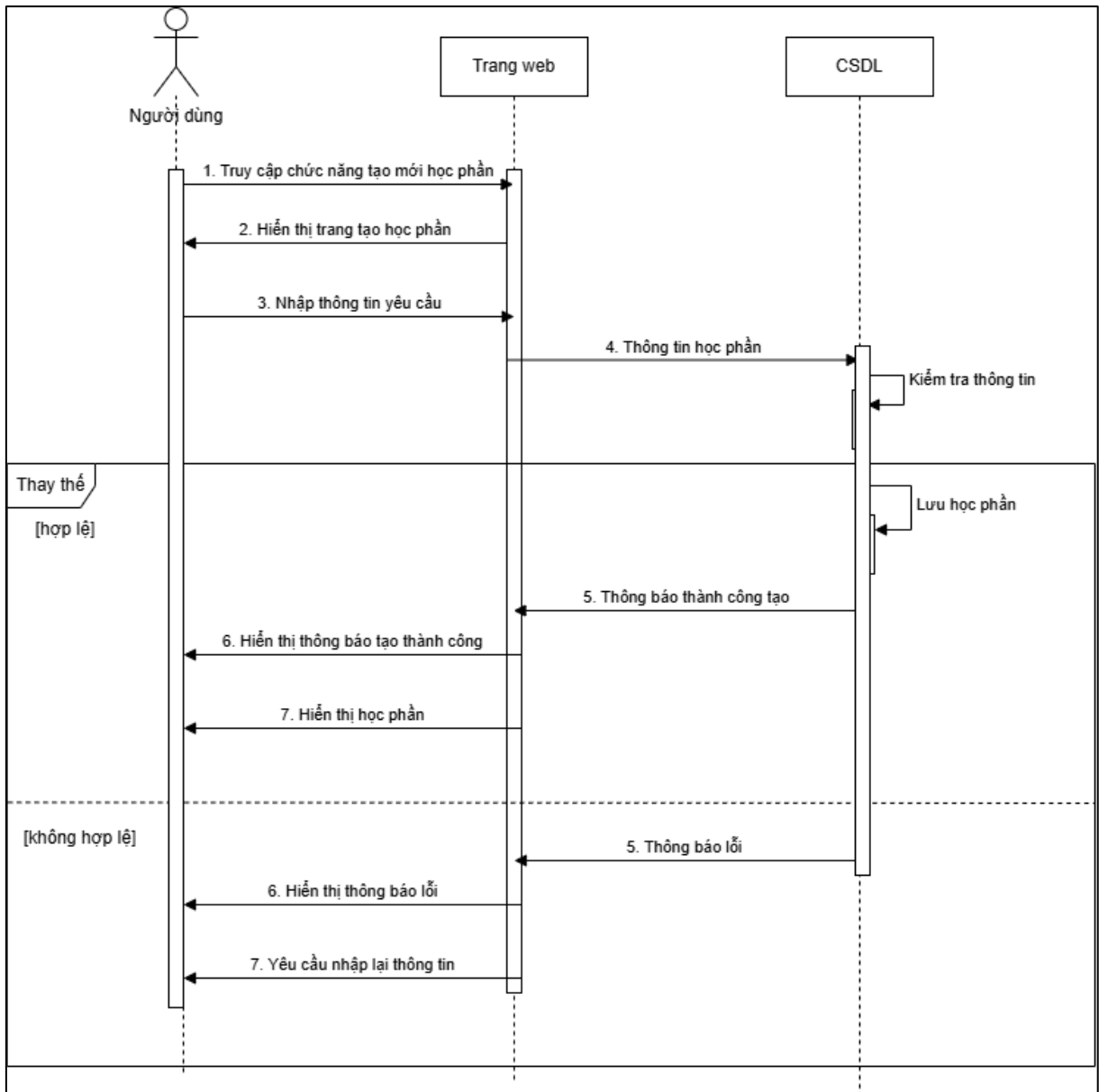
4.3.3.1. Chức năng tạo mới học phần

Tên Use case	Tạo học phần
Tên tác nhân	Người dùng
Điều kiện tiên quyết	Đã đăng nhập vào hệ thống
Đảm bảo tối thiểu	
Đảm bảo thành công	<ul style="list-style-type: none"> - Học phần mới được tạo với đầy đủ thông tin chính xác. - Học phần xuất hiện trong danh sách các học phần trên giao diện của hệ thống. - Dữ liệu học phần sẵn sàng để người dùng học viên truy cập.
Kích hoạt	<ul style="list-style-type: none"> - Đăng nhập vào hệ thống - Chọn vào mục học phần - Chọn thêm học phần
Chuỗi sự kiện chính	<ul style="list-style-type: none"> - Đăng nhập vào hệ thống - Chọn mục học phần - Hệ thống hiện những học phần hiện có - Chọn mục tạo học phần - Điền các thông tin, nội dung học phần - Ca sử dụng kết thúc
Ngoại lệ	<ul style="list-style-type: none"> - Đăng nhập thất bại - Thông tin hoặc nội dung học phần không đúng.

Bảng 4.4. Đặc tả use case Tạo mới học phần



Hình 4.7. Sơ đồ hoạt động use case Tạo mới học phần



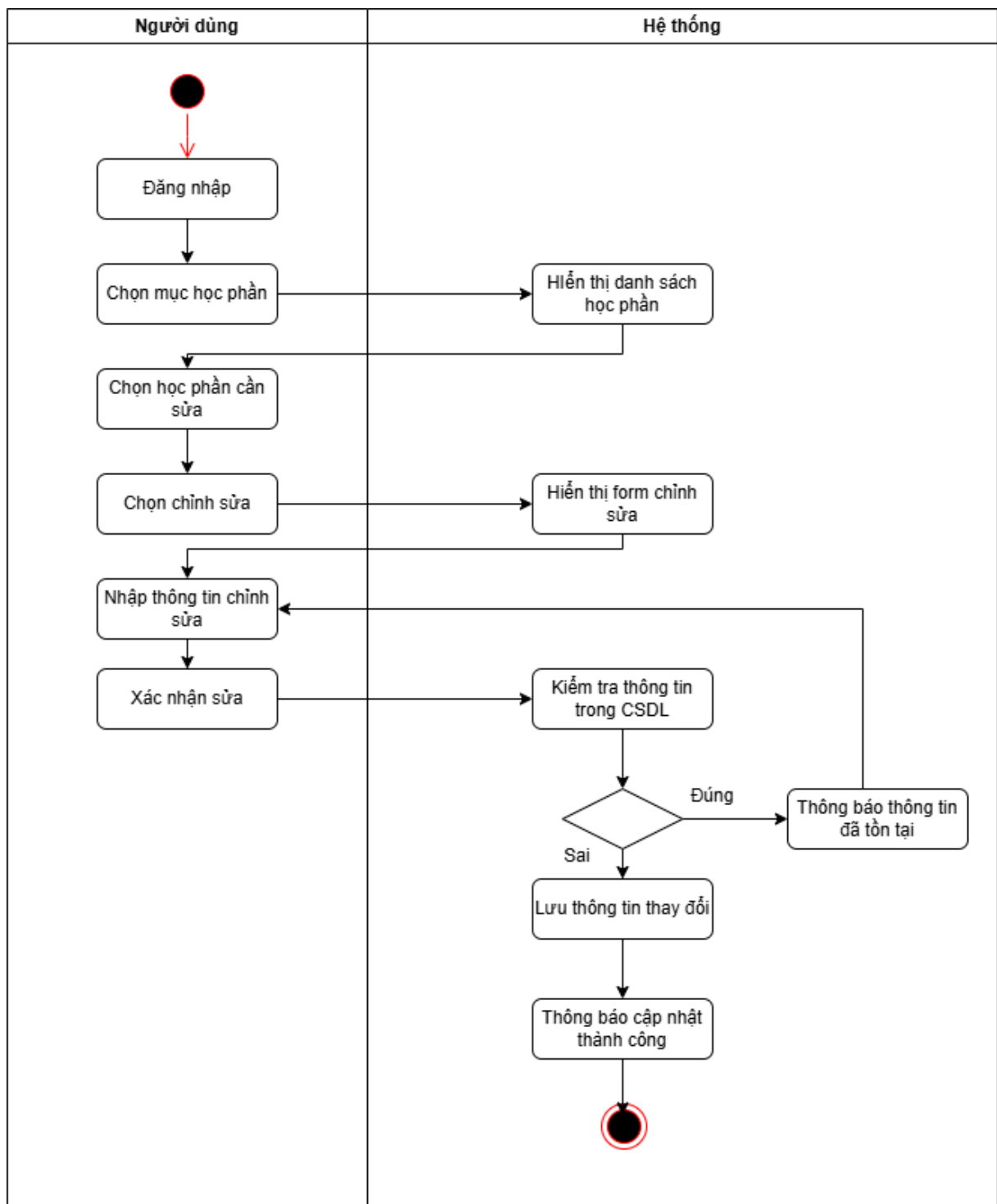
Hình 4.8. Sơ đồ tuần tự use case Tạo mới học phần

4.3.3.2. Chức năng chỉnh sửa học phần

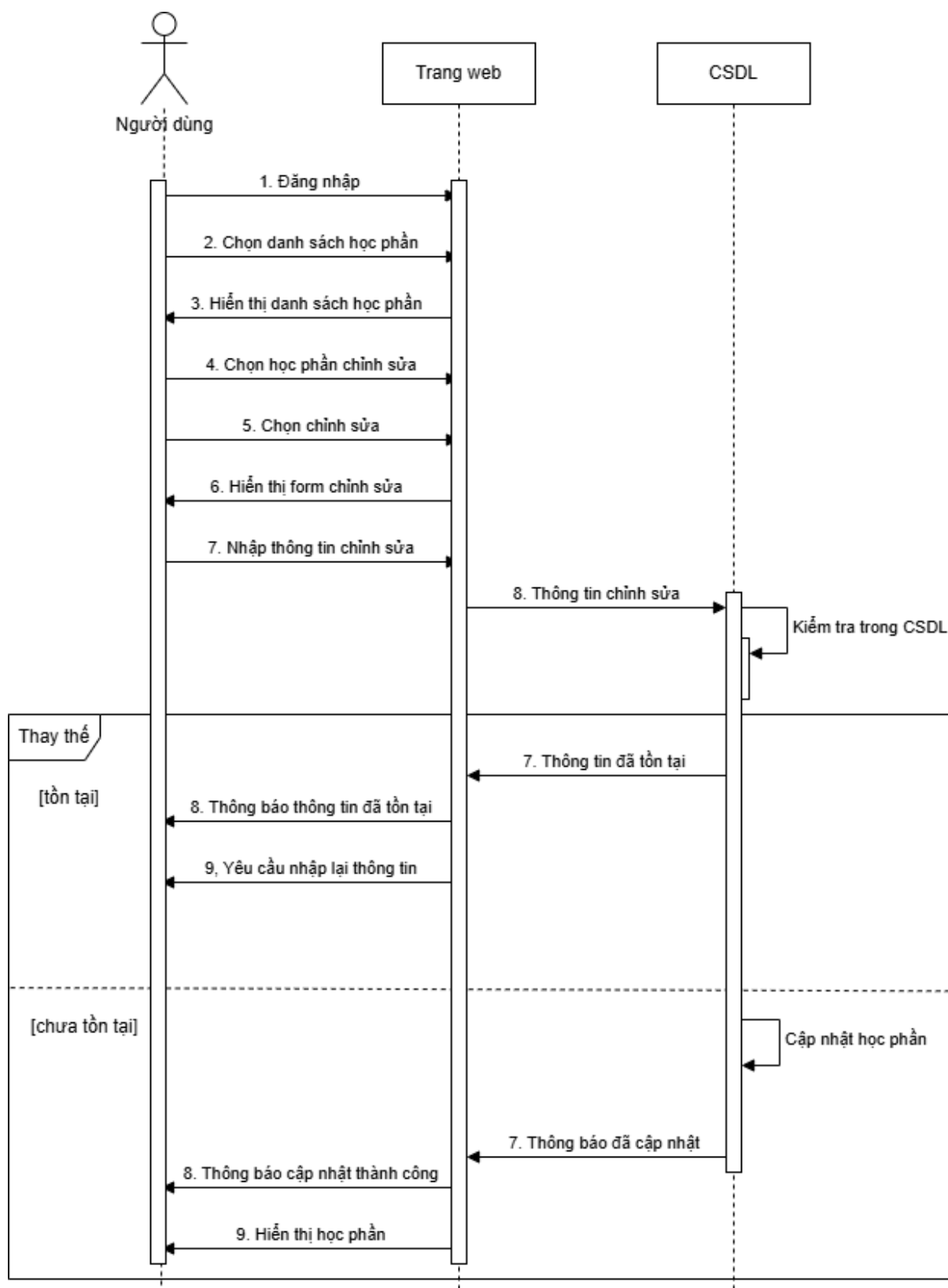
Tên Use case	Chỉnh sửa học phần
Tên tác nhân	Người dùng
Điều kiện tiên quyết	Đã đăng nhập vào hệ thống
Đảm bảo tối thiểu	Tồn tại học phần của người dùng trong cơ sở dữ liệu
Đảm bảo thành công	- Học phần được chỉnh sửa và cập nhật thay đổi.
Kích hoạt	- Đăng nhập vào hệ thống - Chọn vào mục học phần

	<ul style="list-style-type: none"> - Chọn học phần cần sửa - Chọn nút ‘Sửa’
Chuỗi sự kiện chính	<ul style="list-style-type: none"> - Đăng nhập vào hệ thống - Chọn vào mục học phần - Chọn học phần cần sửa - Chọn nút ‘Sửa’ - Nhập thông tin chỉnh sửa - Nhấn ‘Lưu’ - Ca sử dụng kết thúc
Ngoại lệ	<ul style="list-style-type: none"> - Đăng nhập thất bại - Người dùng thoát ra bất kỳ lúc nào. - Dữ liệu không đúng định dạng hoặc thiếu thông tin. - Lỗi kết nối mạng

Bảng 4.5. Đặc tả use case Chỉnh sửa học phần



Hình 4.9. Sơ đồ hoạt động use case Chỉnh sửa học phần



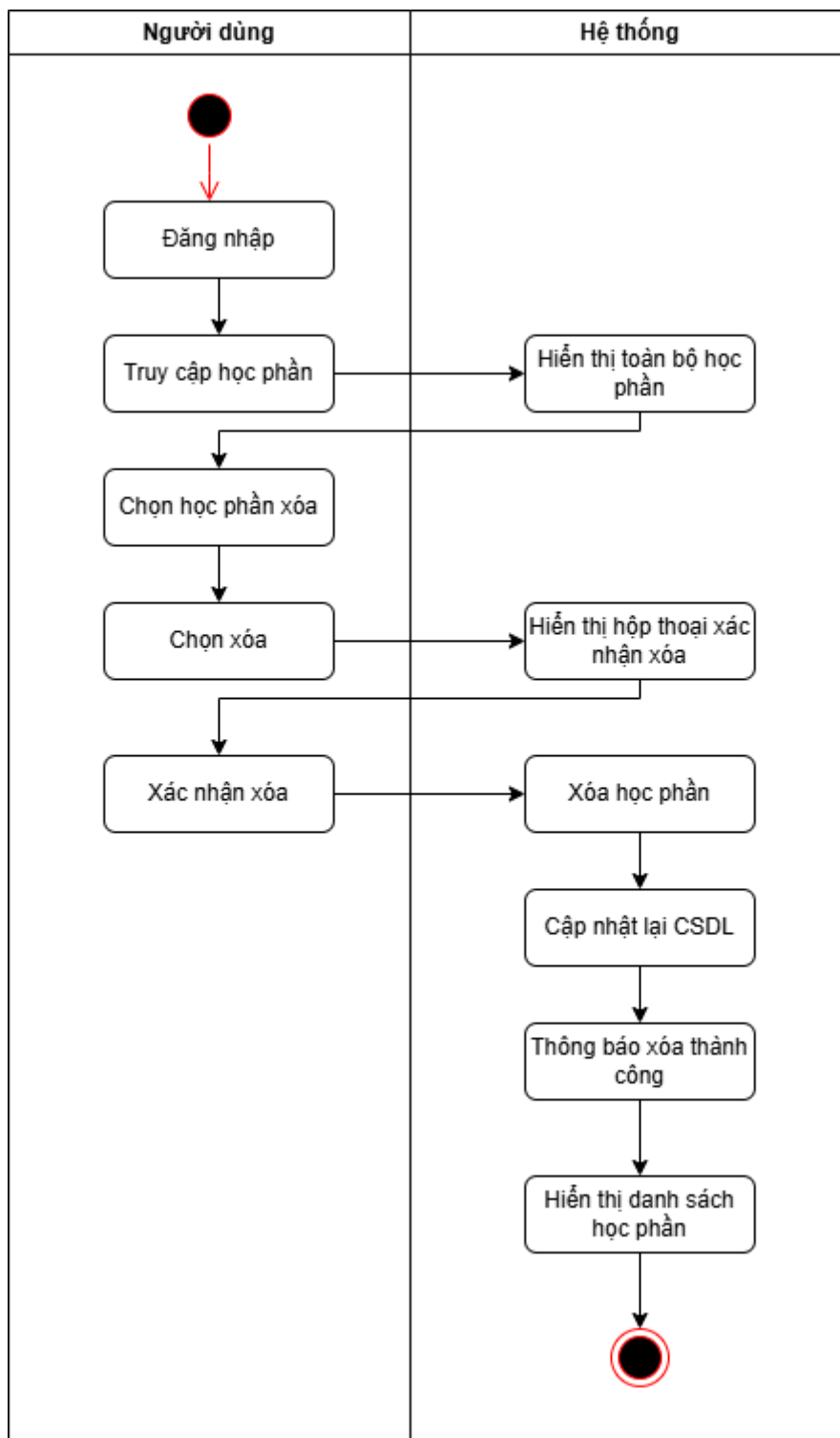
Hình 4.10. Sơ đồ tuần tự use case Chỉnh sửa học phần

4.3.3.3. Chức năng xóa học phần

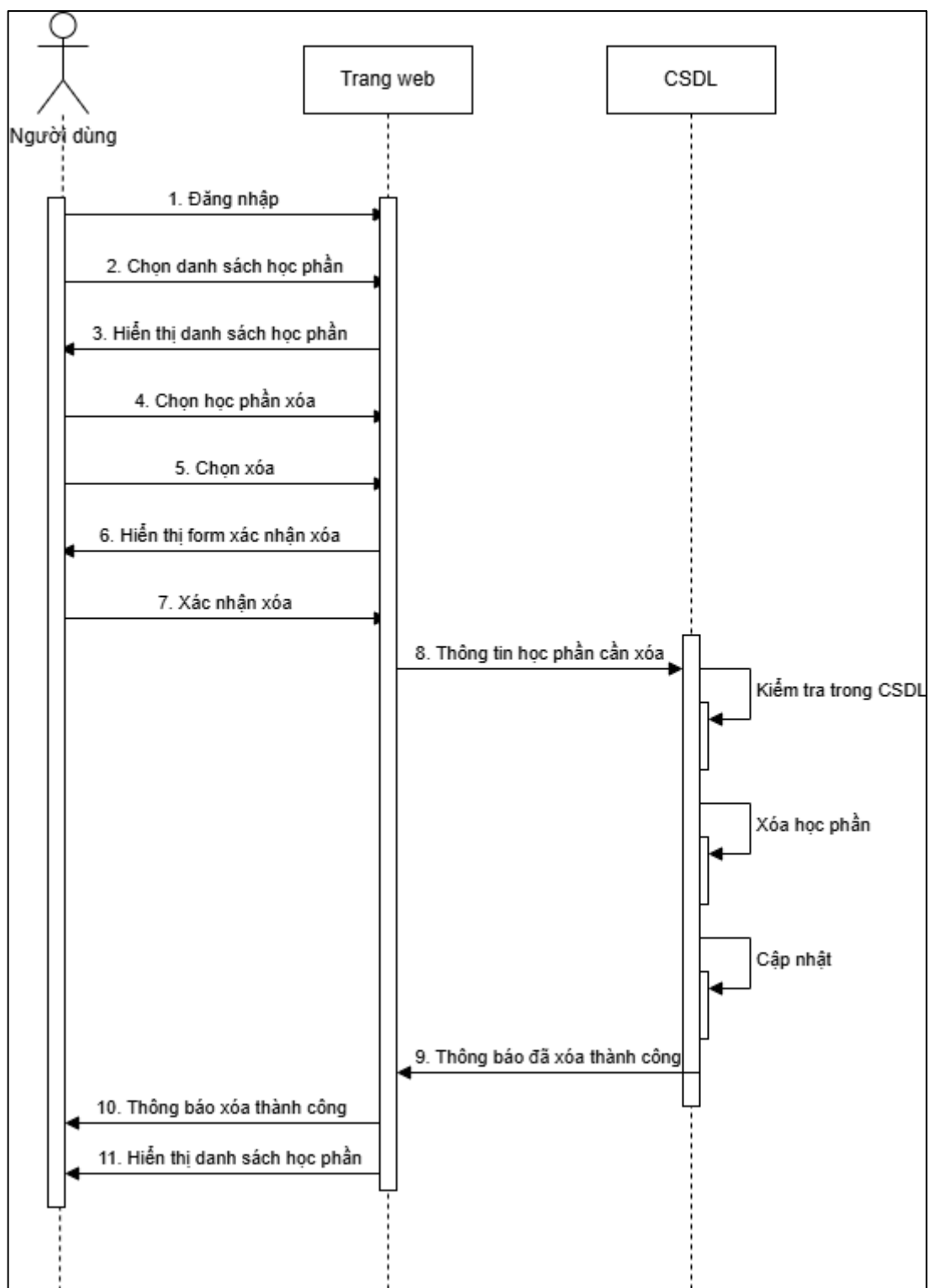
Tên Use case	Xóa học phần
--------------	--------------

Tên tác nhân	Người dùng
Điều kiện tiên quyết	Đã đăng nhập vào hệ thống
Đảm bảo tối thiểu	Tồn tại học phần của người dùng trong cơ sở dữ liệu
Đảm bảo thành công	<ul style="list-style-type: none"> - Học phần được xóa thành công. - Cập nhật lại danh sách học phần
Kích hoạt	<ul style="list-style-type: none"> - Đăng nhập vào hệ thống - Chọn vào mục học phần - Chọn học phần cần xóa - Chọn nút “Xóa”
Chuỗi sự kiện chính	<ul style="list-style-type: none"> - Đăng nhập vào hệ thống - Chọn vào mục học phần - Chọn học phần cần sửa - Chọn nút ‘Xóa’ - Xác nhận xóa - Ca sử dụng kết thúc
Ngoại lệ	<ul style="list-style-type: none"> - Đăng nhập thất bại - Người dùng thoát ra bất kỳ lúc nào. - Không tồn tại học phần cần xóa - Lỗi kết nối mạng

Bảng 4.6. Đặc tả use case Xóa học phần



Hình 4.11. Sơ đồ hoạt động use case Xóa học phần



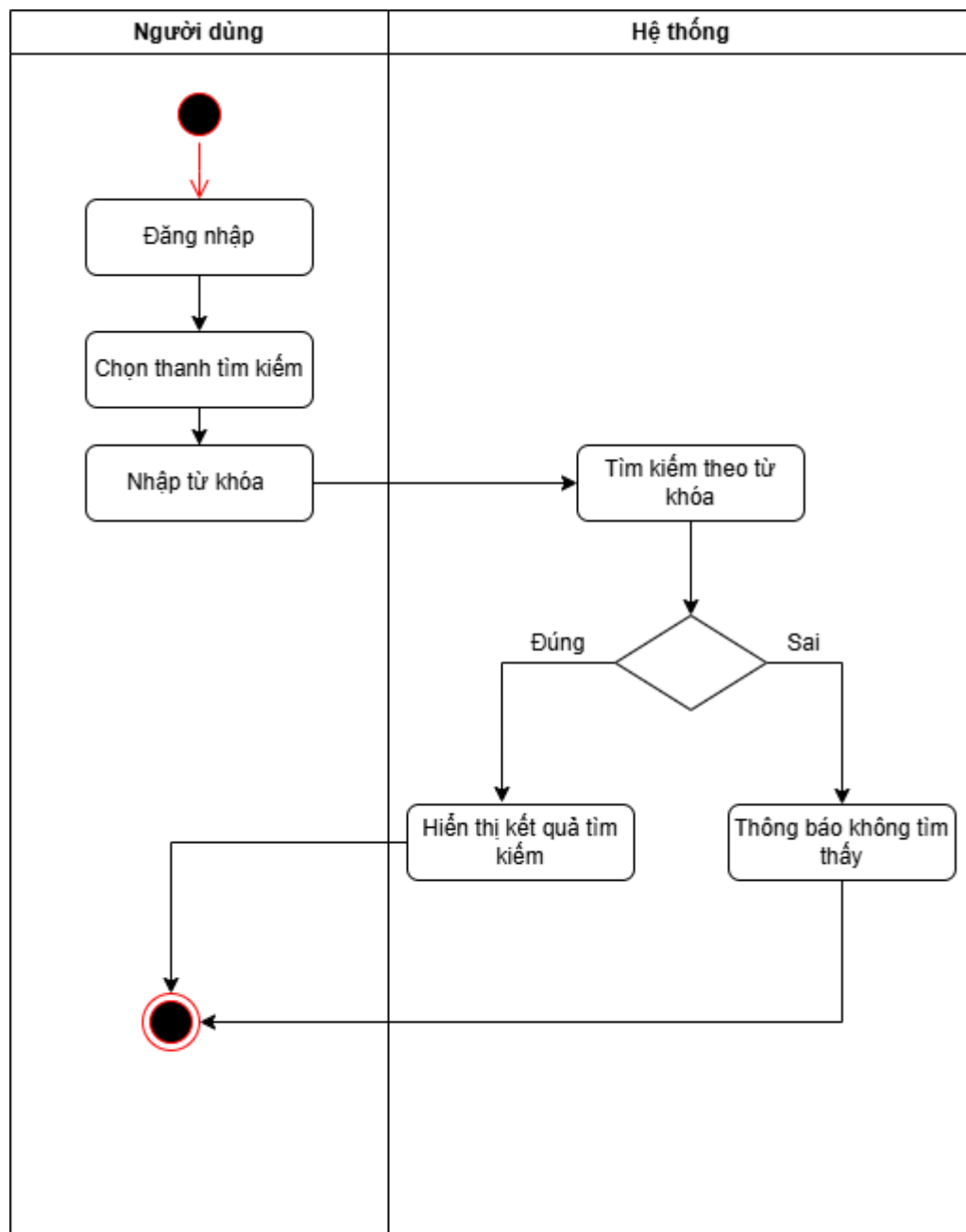
Hình 4.12. Sơ đồ tuần tự use case Xóa học phần

4.3.4. Chức năng tìm kiếm

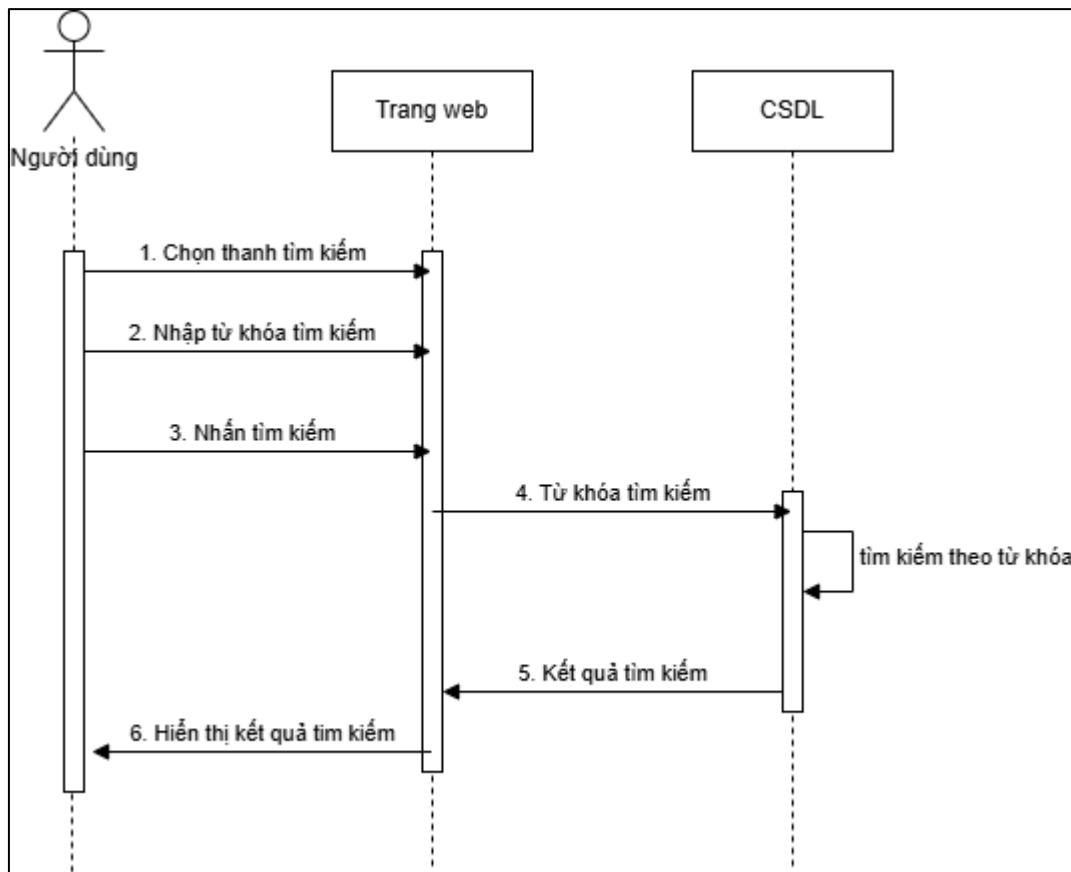
Tên Use case	Tìm kiếm
Tên tác nhân	Người dùng

Điều kiện tiên quyết	Đã đăng nhập vào hệ thống
Đảm bảo tối thiểu	
Đảm bảo thành công	- Hiện thị kết quả tìm kiếm
Kích hoạt	- Đăng nhập vào hệ thống - Chọn vào thanh tìm kiếm
Chuỗi sự kiện chính	- Đăng nhập vào hệ thống - Chọn thanh tìm kiếm - Nhập từ khóa tìm kiếm - Nhấn 'Tìm kiếm' - Ca sử dụng kết thúc
Ngoại lệ	- Đăng nhập thất bại - Người dùng thoát ra bất kỳ lúc nào. - Lỗi kết nối mạng

Bảng 4.7. Đặc tả use case Tìm kiếm



Hình 4.13. Sơ đồ hoạt động use case Tìm kiếm



Hình 4.14. Sơ đồ tuần tự use case Tìm kiếm

4.4. Thiết kế cơ sở dữ liệu

4.4.1. Xác định thực thể

4.4.1.1. Thực thể USER (Người dùng)

- User_Id: Mã người dùng.
- Email: Tên đăng nhập của người dùng.
- Full_Name: Họ và tên người dùng.
- Avatar: Ảnh đại diện của người dùng.
- Block: Xác định người dùng có bị khoá tài khoản hay không.
- Active: Xác định tài khoản đã kích hoạt chưa.
- Password: Mật khẩu tài khoản.
- Created_At: Thời gian người dùng được tạo.
- Updated_At: Thời gian người dùng được cập nhật.

4.4.1.2. Thực thể *ROLE* (Vai trò)

- Role_Id: Mã vai trò.
- Name: Tên của vai trò.

4.4.1.3. Thực thể *TOPIC* (Chủ đề)

- Topic_Id: Mã chủ đề.
- Title: Tiêu đề của chủ đề.
- Description: Mô tả của chủ đề.
- Created_At: Thời gian chủ đề được tạo.
- Updated_At: Thời gian chủ đề được cập nhật.

4.4.1.4. Thực thể *WORD* (Từ vựng)

- Word_Id: Mã từ vựng.
- Name: Tên của từ vựng.
- Pronounce: Phát âm của từ.
- Meaning: Nghĩa của từ.
- Example: Ví dụ về từ.
- Has_Remembered: Xác định từ này đã ghi nhớ hay chưa
- Created_At: Thời gian từ vựng được tạo.
- Updated_At: Thời gian từ vựng được cập nhật.

4.4.1.5. Thực thể *TYPE* (Loại từ)

- Type_Id: Mã loại từ.
- Name: Tên của loại từ
- Description: Mô tả của loại từ.
- Symbol: Kí hiệu của loại từ

4.4.1.6. Thực thể *EXAM* (Kiểm tra)

- Exam_Id: Mã kiểm tra
- Total_Questions: Số lượng câu hỏi.
- Start_At: Thời gian bắt đầu.
- End_At: Thời gian kết thúc.

- Correct_Count: số lượng câu đúng.

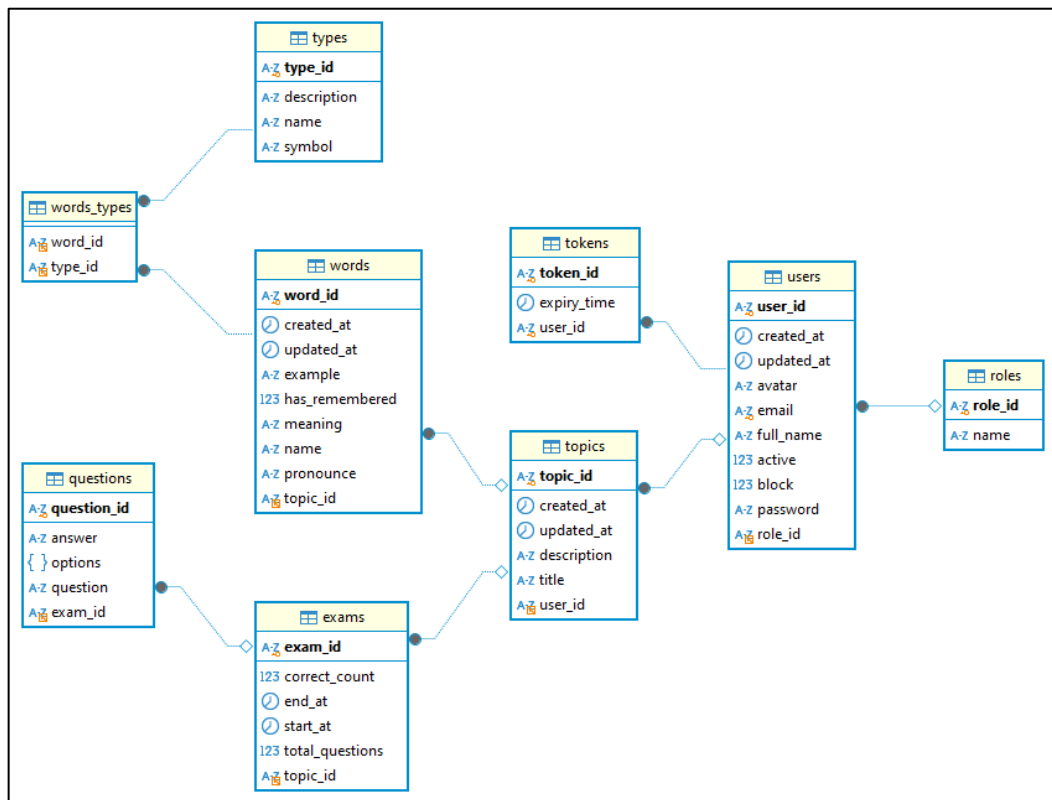
4.4.1.7. Thực thể TOKEN

- Token_Id: Mã của token.
- Expiry_Long: Thời gian token hết hạn.

4.4.1.8. Thực thể QUESTION (Câu hỏi)

- Question_Id: Mã câu hỏi.
- Question: Câu hỏi.
- Options: Các đáp án lựa chọn.
- Answer: Đáp án.

4.4.2. Lược đồ quan hệ



Hình 4.15. Lược đồ quan hệ

4.4.3. Đặc tả các bảng dữ liệu

4.4.3.1. Bảng USER

STT	Tên trường	Kiểu dữ liệu	Khoá	Diễn giải
1	User_Id	Varchar	Chính	Mã người dùng

STT	Tên trường	Kiểu dữ liệu	Khoá	Diễn giải
2	Created_At	Datetime		Thời gian tạo
3	Updated_At	Datetime		Thời gian cập nhật
4	Avatar	Varchar		Ảnh đại diện
5	Email	Varchar		Thư điện tử
6	Full_Name	Varchar		Họ và tên
7	Block	Bit		Khoá tài khoản
8	Password	Varchar		Mật khẩu
9	Active	Bit		Kích hoạt tài khoản
10	Role_Id	Varchar	Ngoại	Mã vai trò

Bảng 4.8. Đặc tả bảng USER

4.4.3.2. Bảng ROLE

STT	Tên trường	Kiểu dữ liệu	Khoá	Diễn giải
1	Role_Id	Varchar	Chính	Mã vai trò
2	Name	Varchar		Tên vai trò

Bảng 4.9. Đặc tả bảng ROLE

4.4.3.3. Bảng TOPIC

STT	Tên trường	Kiểu dữ liệu	Khoá	Diễn giải
1	Topic_Id	Varchar	Chính	Mã chủ đề
2	Created_At	Datetime		Thời gian tạo
3	Updated_At	Datetime		Thời gian cập nhật
4	Description	Text		Mô tả
5	Title	Varchar		Tiêu đề
6	User_Id	Varchar	Ngoại	Mã người dùng

Bảng 4.10. Đặc tả bảng TOPIC

4.4.3.4. Bảng WORD

STT	Tên trường	Kiểu dữ liệu	Khoá	Diễn giải
1	Word_Id	Varchar	Chính	Mã từ vựng
2	Created_At	Datetime		Thời gian tạo
3	Updated_At	Datetime		Thời gian cập nhật
4	Example	Text		Ví dụ
5	Has_Remembered	Bit		Trạng thái ghi nhớ từ vựng
6	Meaning	Varchar		Nghĩa của từ
7	Name	Varchar		Tên từ vựng
8	Pronounce	Varchar		Cách phát âm
9	Topic_Id	Varchar	Ngoại	Mã chủ đề

Bảng 4.11. Đặc tả bảng WORD

4.4.3.5. Bảng TYPE

STT	Tên trường	Kiểu dữ liệu	Khoá	Diễn giải
1	Type_Id	Varchar	Chính	Mã loại từ
2	Description	Text		Mô tả
3	Name	Varchar		Tên loại từ
4	Symbol	Varchar		Ký hiệu

Bảng 4.12. Đặc tả bảng TYPE

4.4.3.6. Bảng EXAM

STT	Tên trường	Kiểu dữ liệu	Khoá	Diễn giải
1	Quizz_Id	Varchar	Chính	Mã kiểm tra
2	Start_At	Datetime		Thời gian bắt đầu
3	End_At	Datetime		Thời gian kết thúc
4	Correct_Count	Integer		Số lượng câu đúng
5	Total_Questions	Integer		Số lượng câu hỏi
6	Topic_Id	Varchar	Ngoại	Mã chủ đề

Bảng 4.13. Đặc tả bảng EXAM

4.4.3.7. Bảng TOKEN

STT	Tên trường	Kiểu dữ liệu	Khoá	Diễn giải
1	Token_Id	Varchar	Chính	Mã token
2	Expiry_Time	Datetime		Thời gian token hết hạn

Bảng 4.14: Đặc tả bảng TOKEN

4.4.3.8. Bảng QUESTION

STT	Tên trường	Kiểu dữ liệu	Khoá	Diễn giải
1	Question_Id	Varchar	Chính	Mã câu hỏi
2	Answer	Varchar		Đáp án
3	Options	Json		Các đáp án lựa chọn
4	Question	Varchar		Câu hỏi
5	Exam_Id	Varchar	Ngoại	Mã kiểm tra

Bảng 4.15. Đặc tả bảng QUESTION

4.4.3.9. Bảng WORD_TYPE

STT	Tên trường	Kiểu dữ liệu	Khoá	Diễn giải
1	Word_Id	Varchar	Chính	Mã từ vựng
2	Type_Id	Varchar	Chính	Mã loại từ

Bảng 4.16. Đặc tả bảng WORD_TYPE

4.5. Thiết kế Chatbot

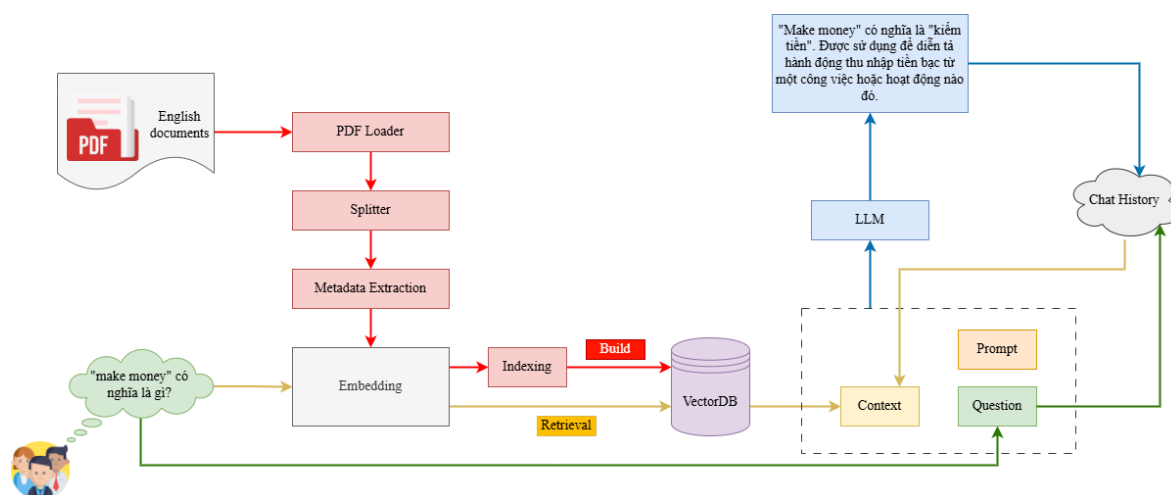
Trong dự án này, chatbot được tích hợp nhằm mang lại khả năng tương tác, hỗ trợ học tiếng Anh một cách thông minh và tiện lợi cho người dùng website. Hệ thống chatbot sẽ được xây dựng bằng thư viện LlamaIndex [15] dựa trên mô hình RAG (Retrieval-Augmented Generation) với các thành phần chính sau:

Document: một đơn vị dữ liệu lớn, chứa nội dung văn bản mà hệ thống xử lý. Nó có thể là một bài viết, một đoạn văn dài, một tài liệu PDF, hoặc bất kỳ dạng văn bản nào được con người tạo ra. Document thường là nguồn dữ liệu đầu vào, được chia nhỏ hoặc trích xuất thông tin để sử dụng trong các tác vụ NLP, như phân loại văn bản, tóm tắt, hoặc trả lời câu hỏi.

Node: một đơn vị dữ liệu nhỏ hơn được trích xuất hoặc phân chia từ document. Nó thường đại diện cho các đoạn văn ngắn, câu, hoặc cụm từ cụ thể, đóng vai trò là thành phần cơ bản để xử lý và phân tích. Các node thường được dùng trong các thuật toán NLP để tạo cấu trúc chỉ mục, biểu diễn vector, hoặc tạo mối liên kết ngữ nghĩa giữa các phần trong tài liệu.

Index: cấu trúc dữ liệu được tạo ra để lưu trữ, tổ chức và quản lý các document hoặc node trong một hệ thống NLP. Mục đích của index là tối ưu hóa việc truy xuất thông tin, cho phép các mô hình NLP nhanh chóng tìm kiếm và lấy thông tin liên quan. Có nhiều loại index như keyword-based index, vector-based index, hoặc hierarchical index, được thiết kế dựa trên nhu cầu cụ thể của ứng dụng NLP.

Sơ đồ phía dưới miêu tả kiến trúc và quy trình xử lý của hệ thống chatbot trong dự án:



Hình 4.16. Pipeline của hệ thống Chatbot

Từ sơ đồ bên trên, việc thiết kế hệ thống chatbot sẽ được chia thành hai phần chính: thiết kế quy trình xử lý dữ liệu và thiết kế cuộc hội thoại.

4.5.1. Quy trình xử lý dữ liệu

Quy trình xử lý dữ liệu bắt đầu từ kho dữ liệu thô lưu trữ các tài liệu học tiếng Anh dưới định dạng .pdf. Các tài liệu này sẽ được đọc và xử lý qua các công cụ PDF loader để tạo ra các đối tượng Document từ dữ liệu đầu vào.

Sau khi các đối tượng Documents được tạo, chúng được đưa qua bộ Splitter để chia nhỏ thành các Node. Quá trình này nhằm chia nhỏ các dữ liệu lớn để dễ dàng quản lý và xử lý hơn.

Tiếp theo, việc trích xuất Metadata được thực hiện trên các Node và tích hợp vào chúng. Metadata có thể là bản tóm tắt nội dung của Node hoặc chủ đề của nội dung dữ liệu trong Node... Ngoài ra, Metadata có thể chứa cả thông tin của các Node liền trước và sau Node hiện tại nhằm duy trì nội dung cũng như ngữ cảnh của Node, điều này giúp quá trình truy xuất và xử lý thông tin sau này trở nên hiệu quả hơn.

Sau khi đã có Metadata, các Nodes sẽ được embedding bằng các mô hình Embedding được cung cấp sẵn, chuyển đổi nội dung trong Node thành định dạng Vector để dễ dàng tìm kiếm và so sánh.

Cuối cùng, quá trình tạo Index sẽ được thực hiện trên các Node đã được embedding và lưu trữ vào Vector Database.

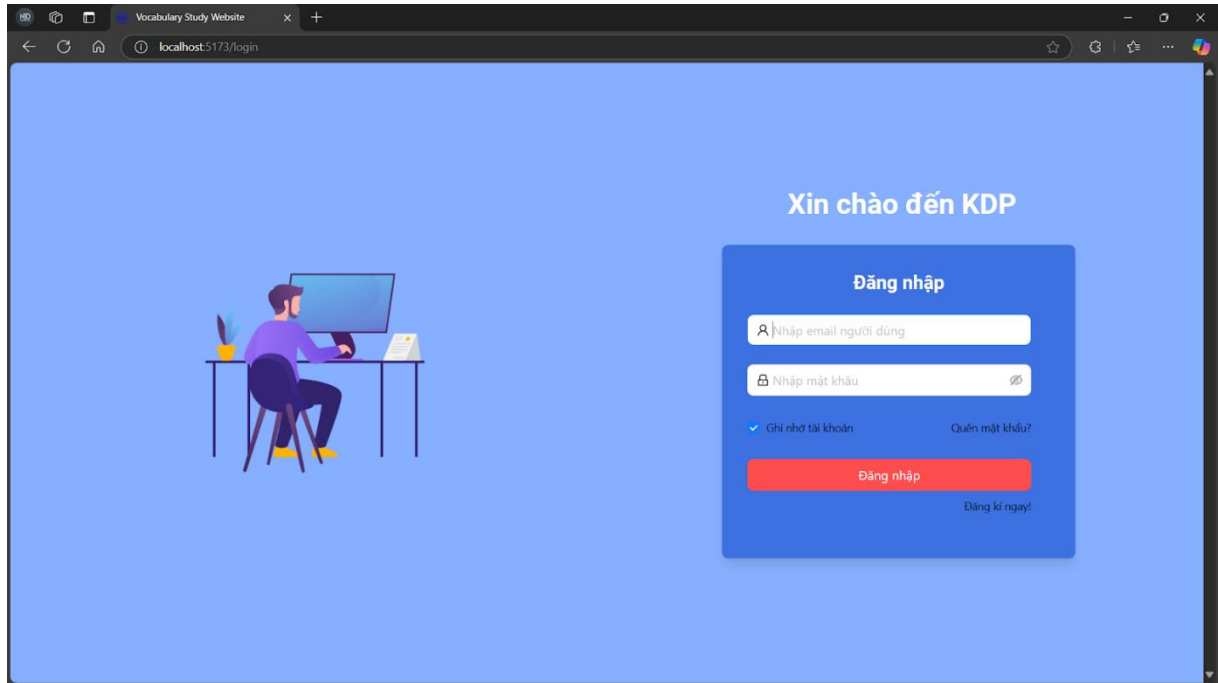
4.5.2. Thiết kế cuộc hội thoại

Quá trình xử lý cuộc hội thoại sẽ được xử lý theo trình tự như sau:

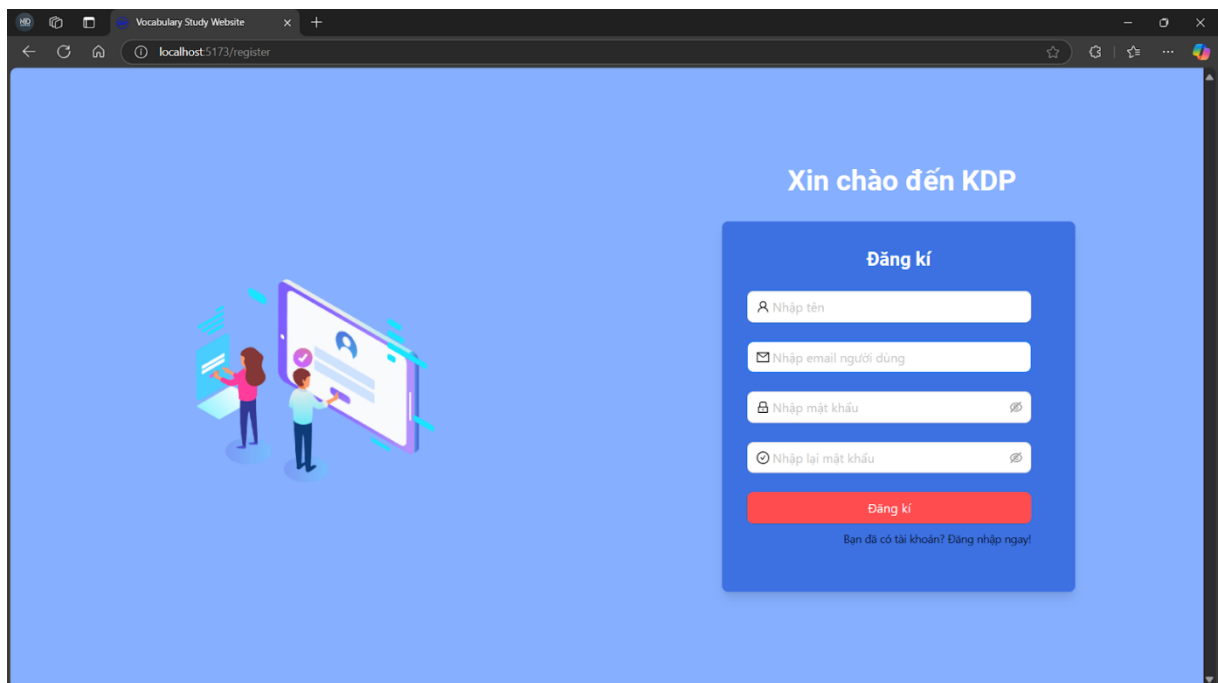
- Câu hỏi của người dùng (Question) sẽ được chuyển đổi thành định dạng Vector bằng mô hình Embedding và sau đó được so sánh với các Vector dữ liệu trong Vector Database (tính khoảng cách hoặc tính độ tương đồng) để truy xuất ra các nội dung liên quan đến câu hỏi.
- Các nội dung liên quan được truy xuất sẽ được kết hợp với lịch sử cuộc trò chuyện (nếu có) để tạo ra ngữ cảnh (Context).
- Thiết lập Prompt để hướng dẫn cách mô hình LLM xử lý và tạo ra câu trả lời.
- Cuối cùng, Context, Question và Prompt sẽ được đưa vào bộ tạo văn bản hay mô hình LLM để tối ưu hóa câu trả lời, tạo ra câu trả lời mạch lạc và phù hợp với ngữ cảnh.

CHƯƠNG 5: XÂY DỰNG CHƯƠNG TRÌNH

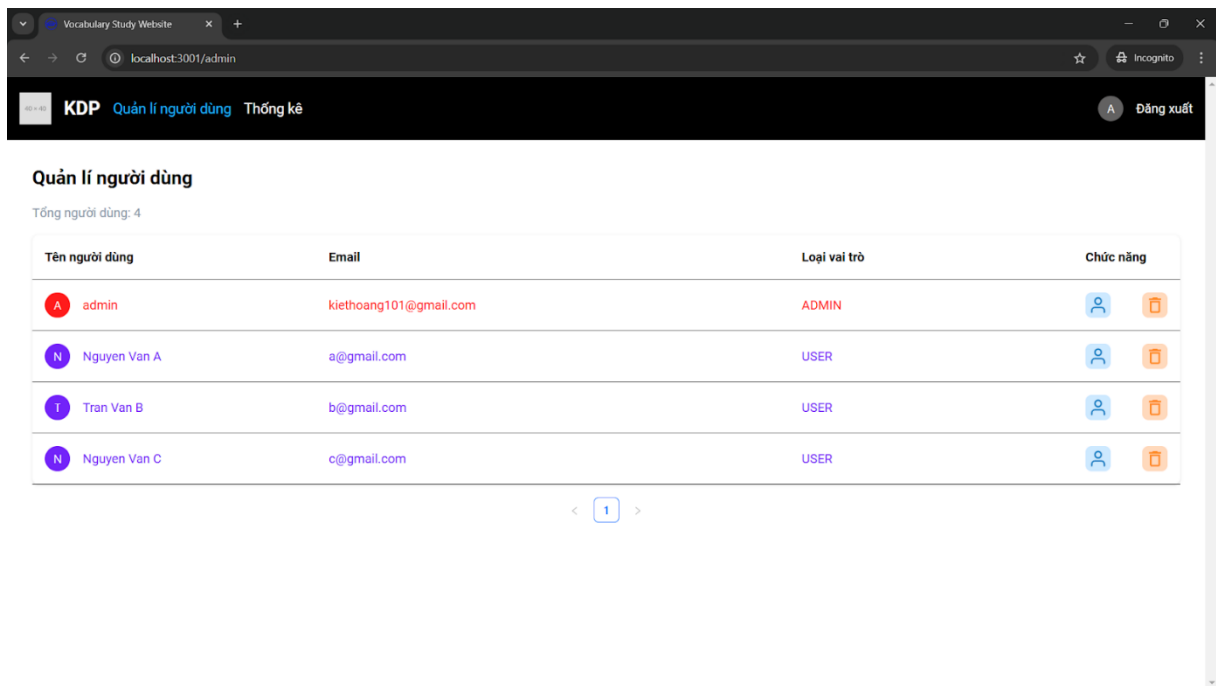
5.1. Giao diện chính



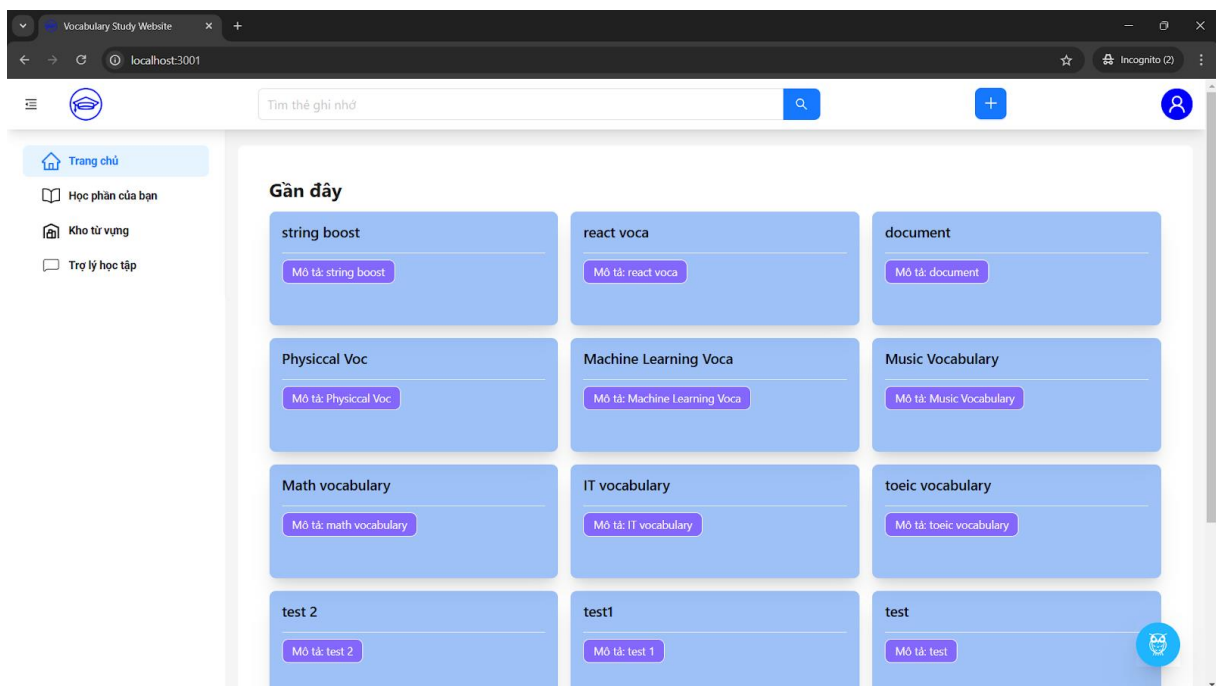
Hình 5.1. Giao diện đăng nhập



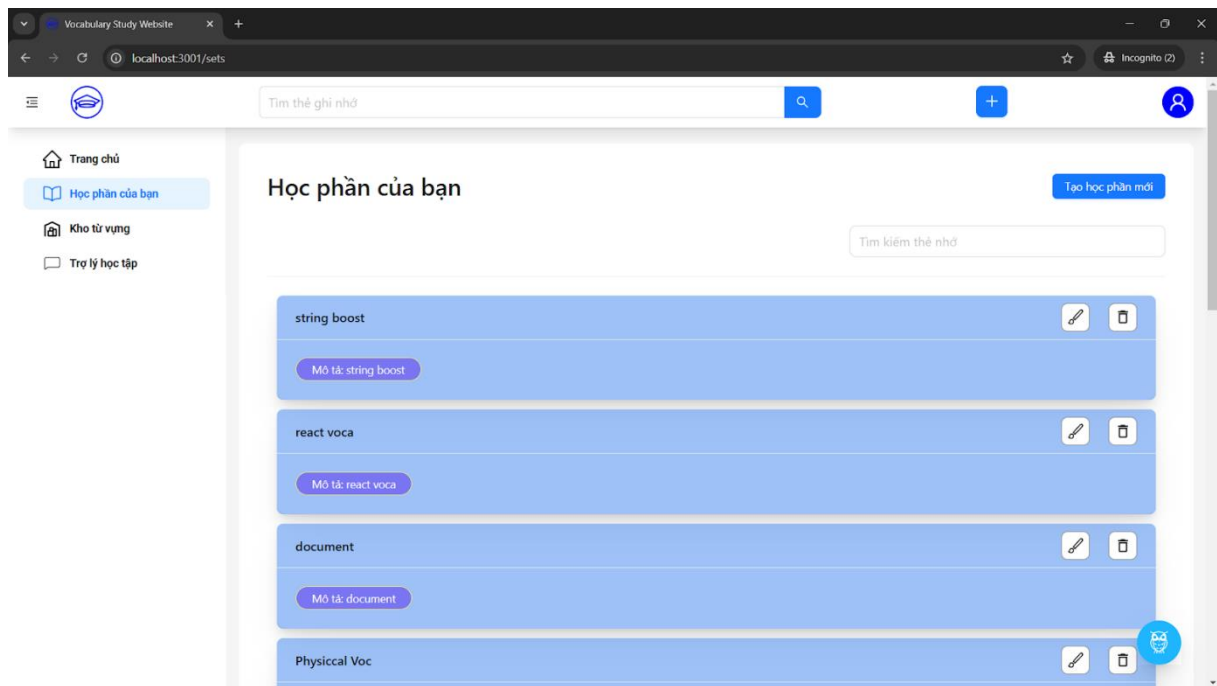
Hình 5.2. Giao diện đăng lý



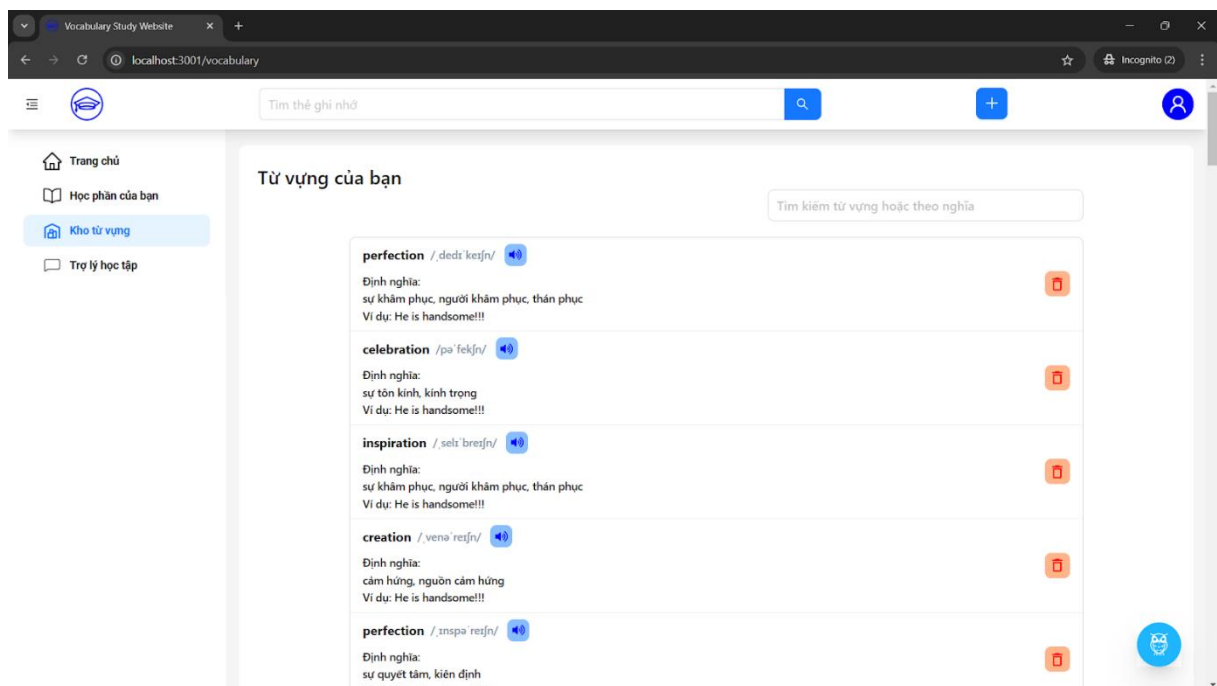
Hình 5.3. Giao diện trang chủ Admin



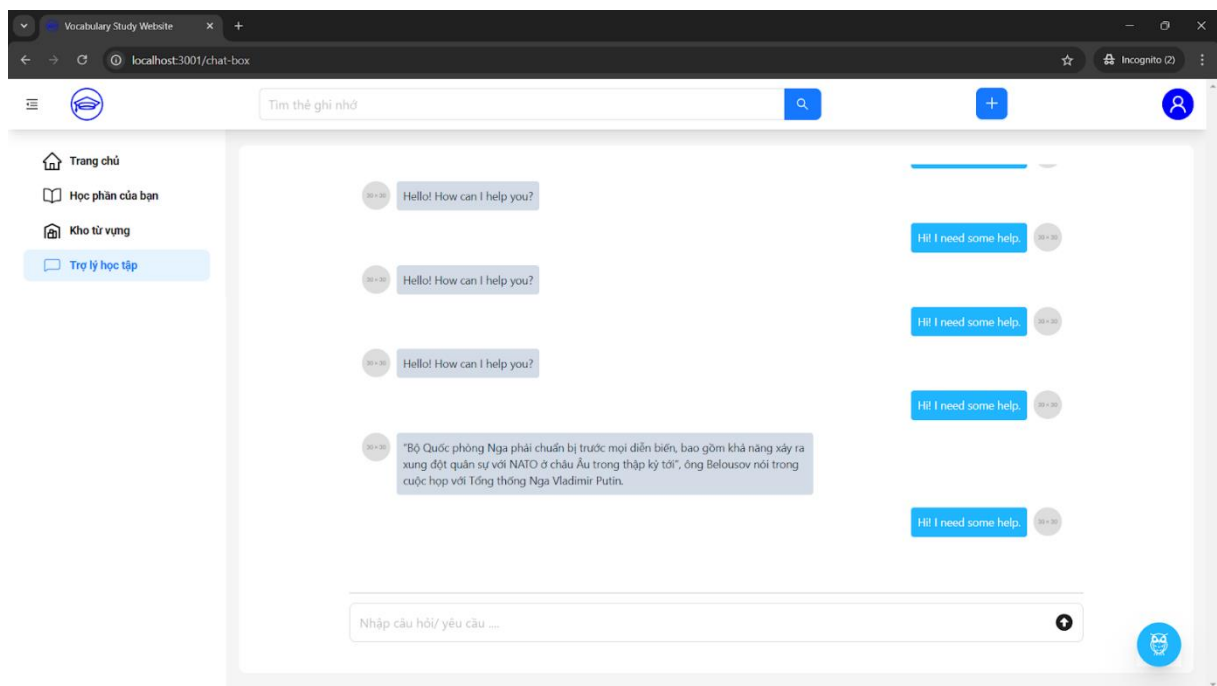
Hình 5.4. Giao diện trang chủ người dùng



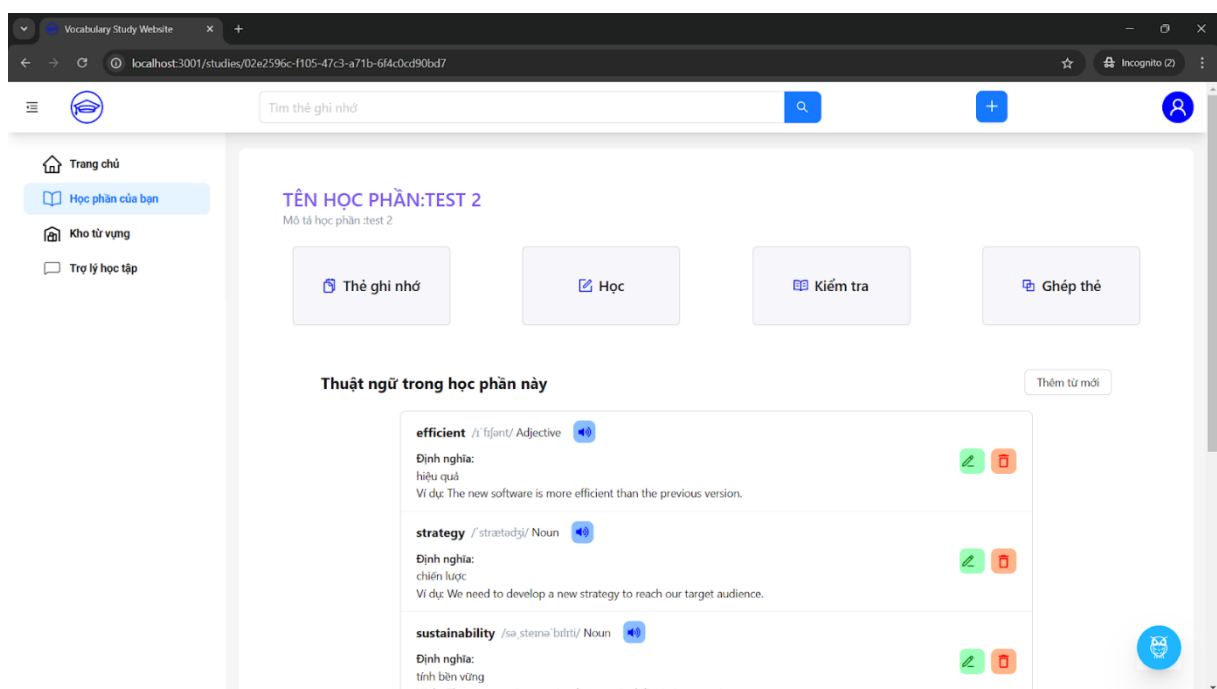
Hình 5.5. Giao diện danh sách học phần



Hình 5.6. Giao diện từ vựng

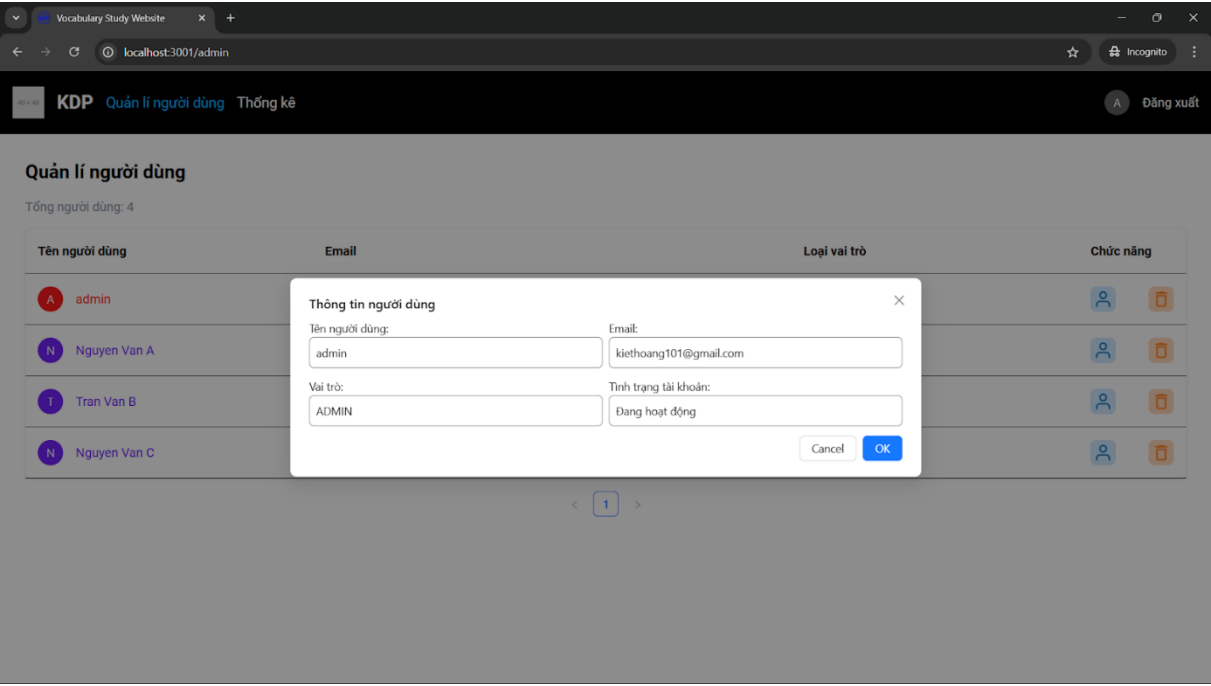


Hình 5.7. Giao diện trợ lý học tập

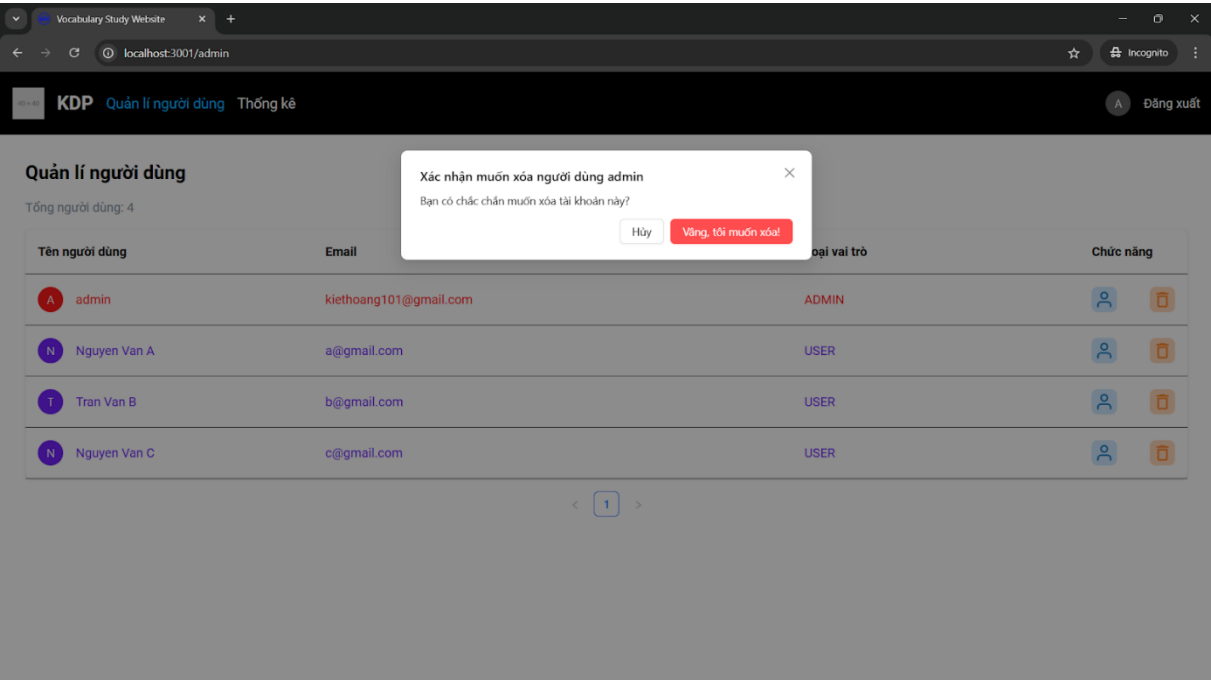


Hình 5.8. Giao diện nội dung học phần

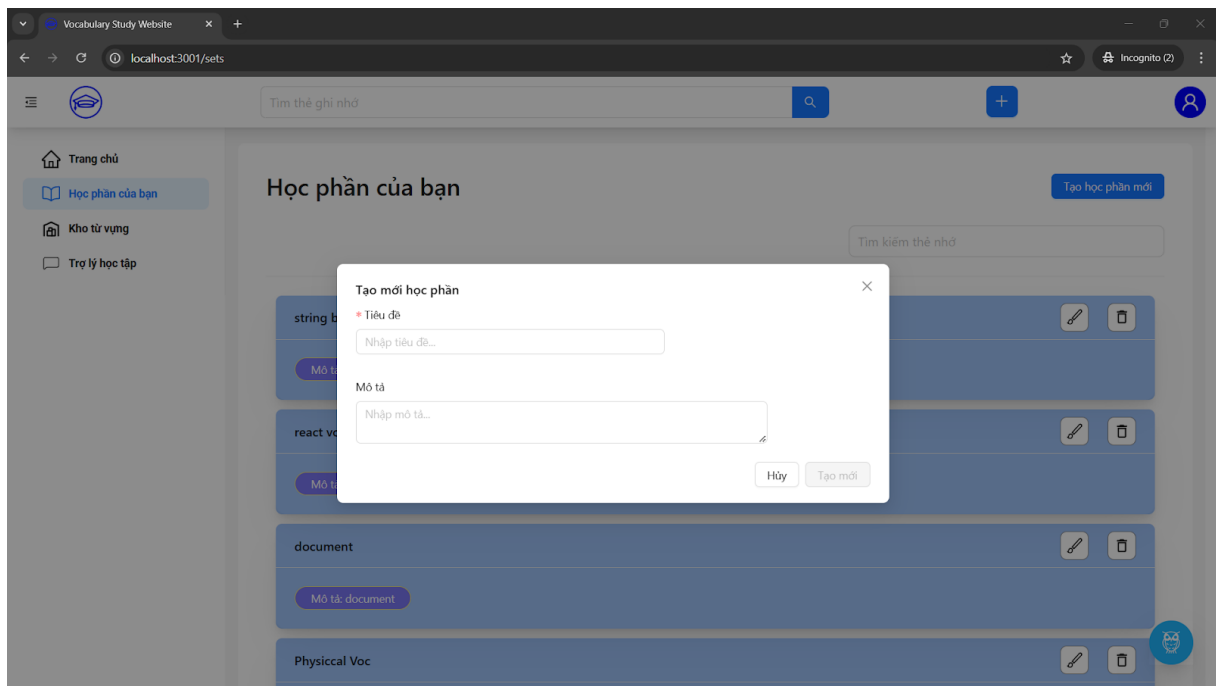
5.2. Giao diện xử lý



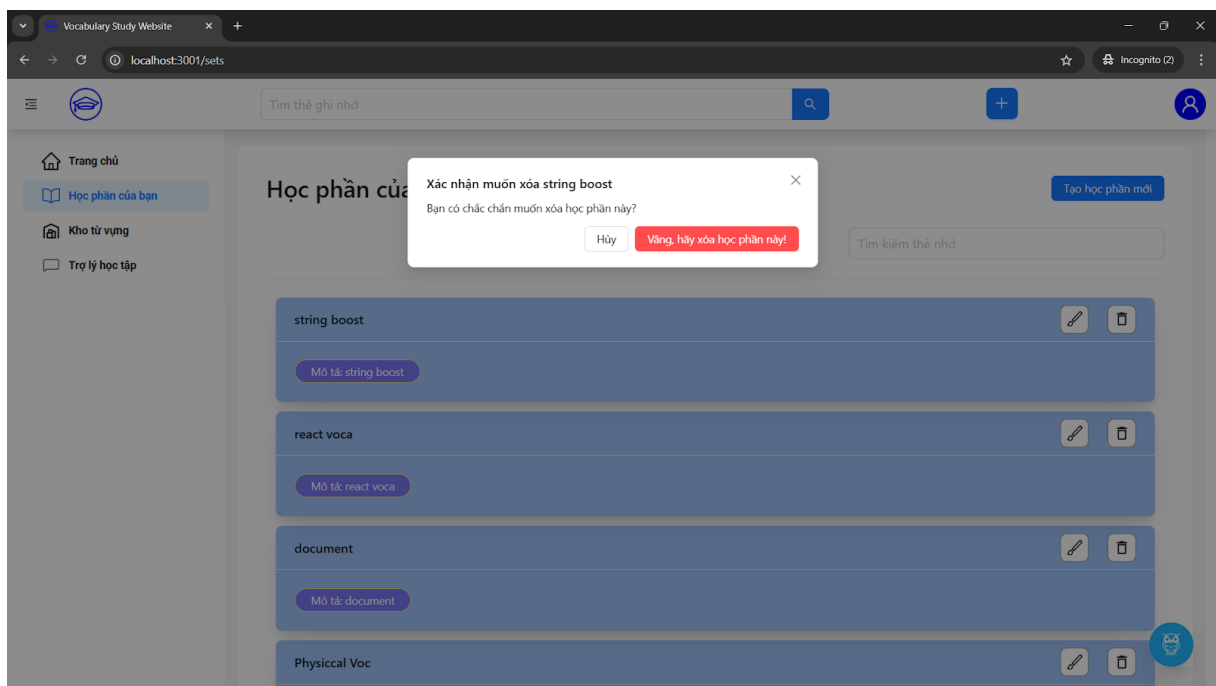
Hình 5.9. Giao diện thông tin người dùng



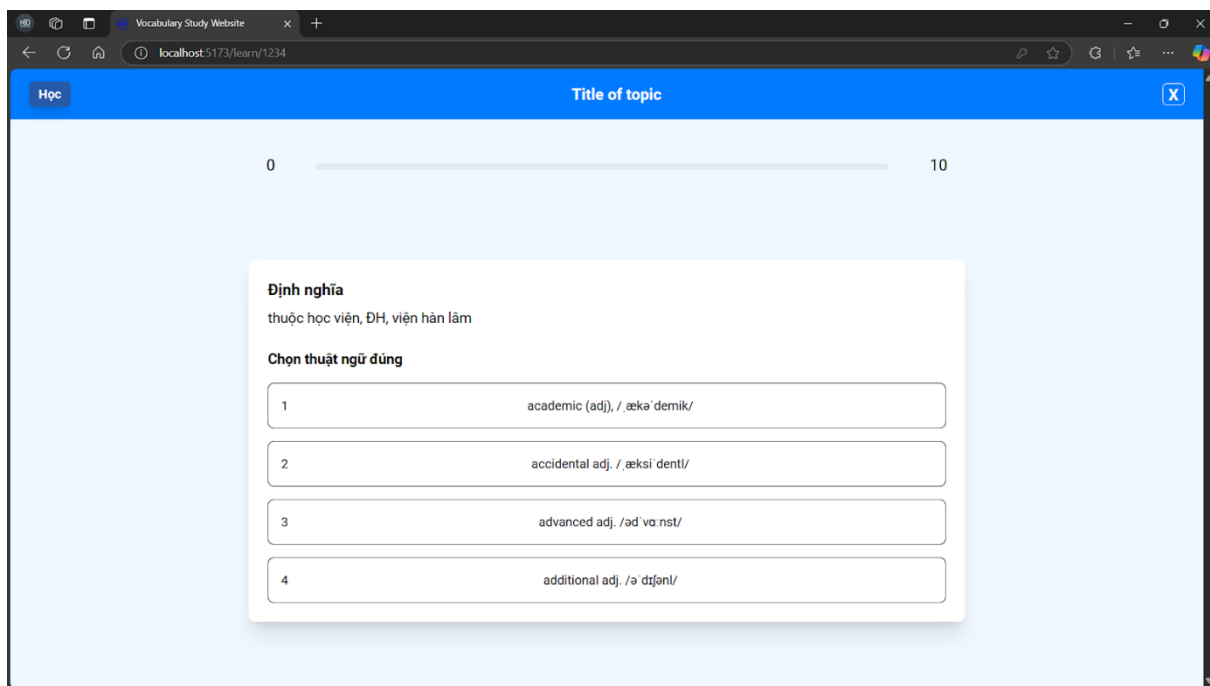
Hình 5.10. Giao diện khóa người dùng



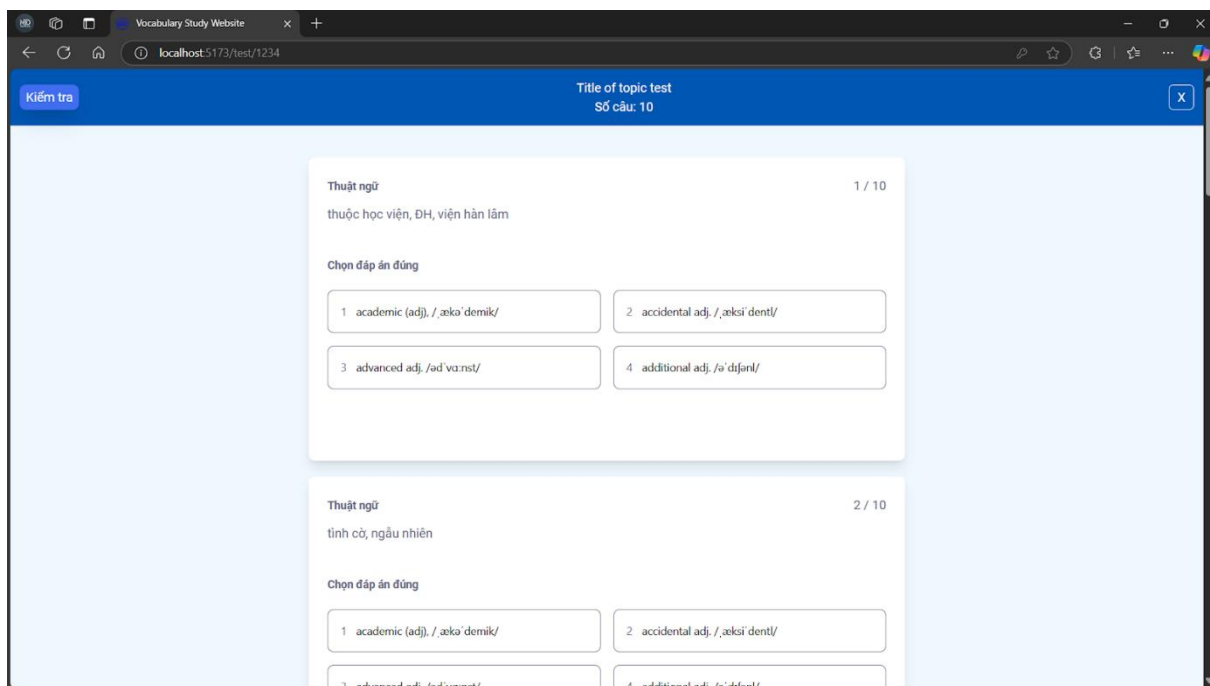
Hình 5.11. Giao diện chỉnh sửa học phần



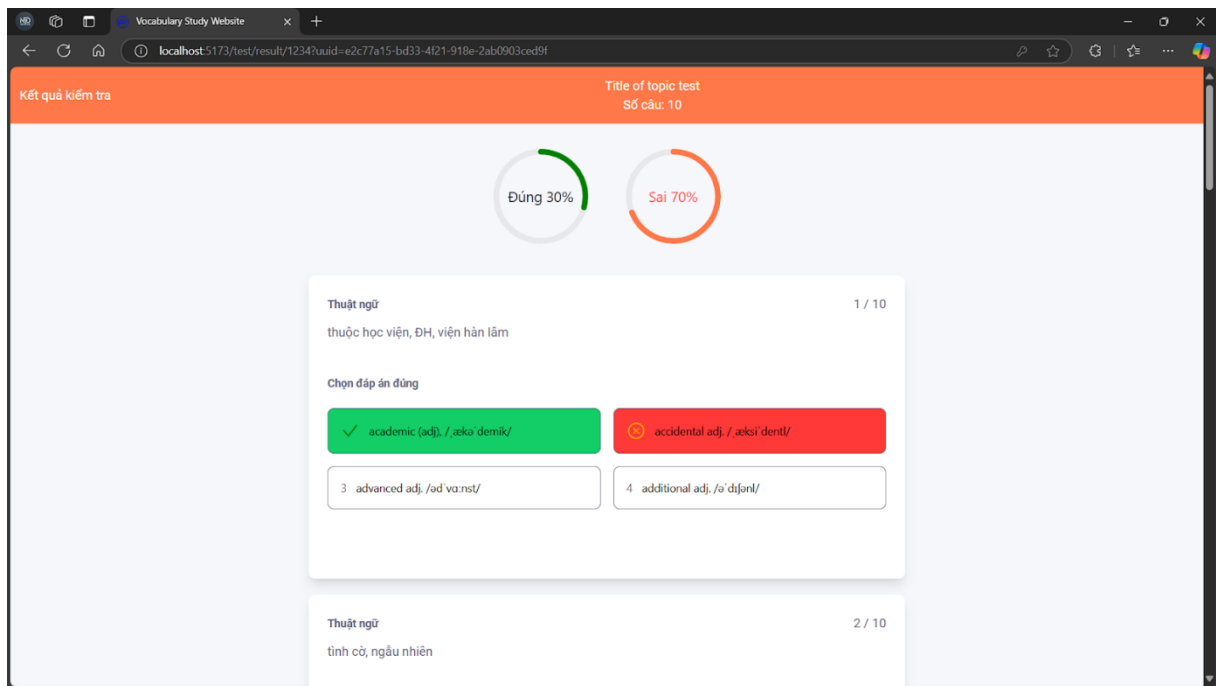
Hình 5.12. Giao diện xóa học phần



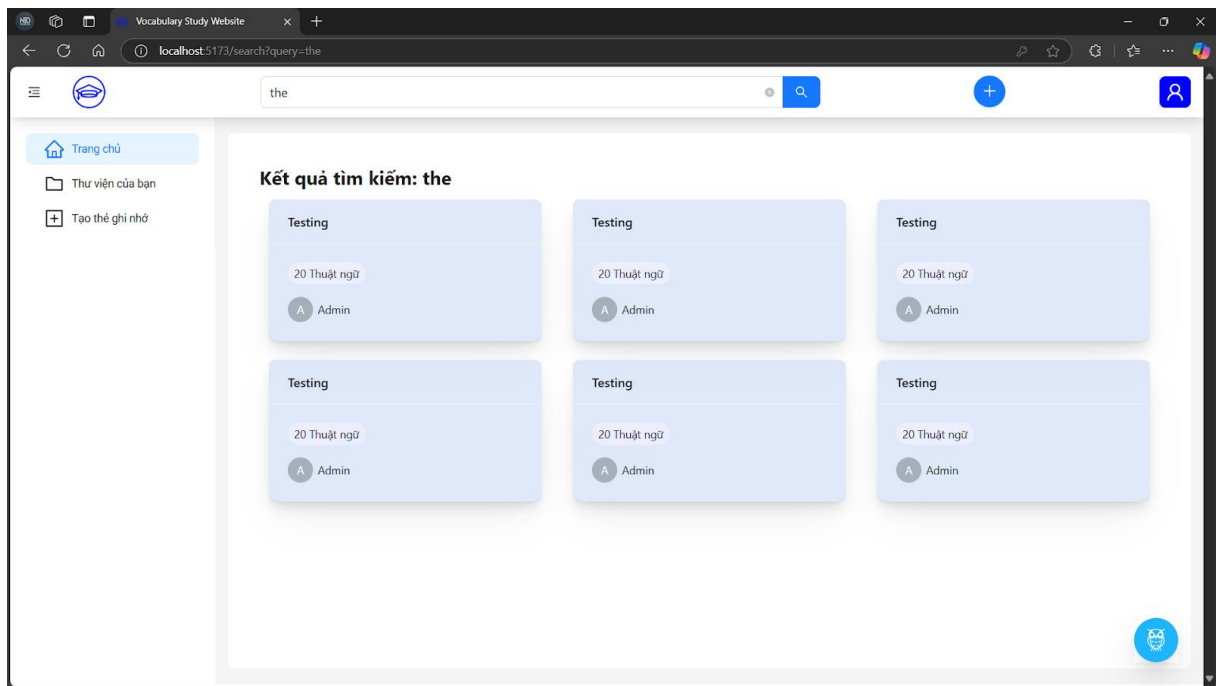
Hình 5.13. Giao diện học



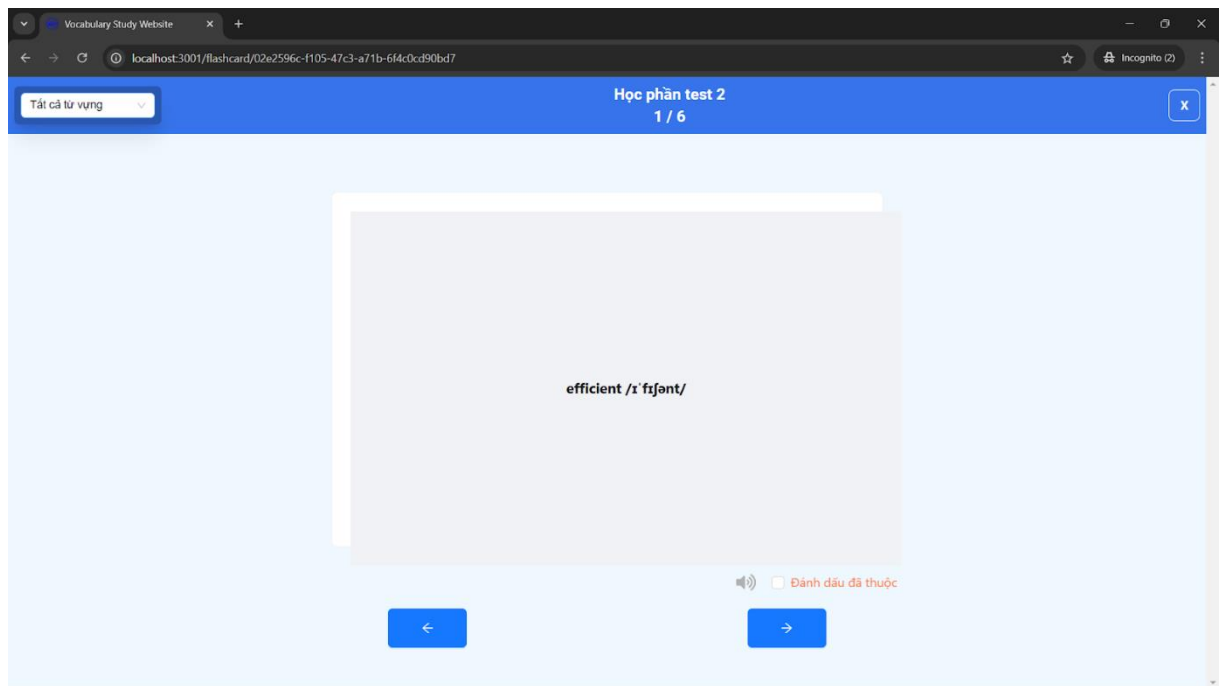
Hình 5.14. Giao diện kiểm tra



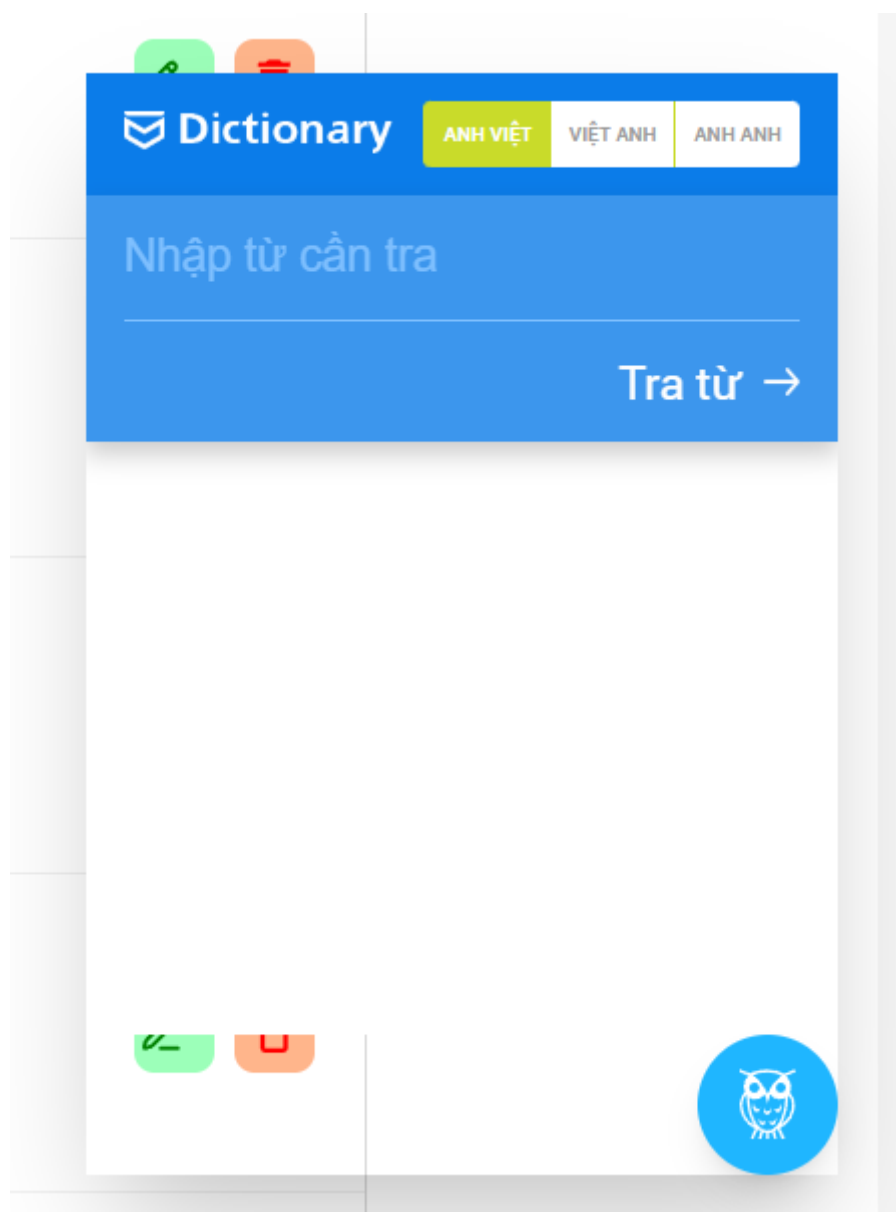
Hình 5.15. Giao diện kết quả kiểm tra



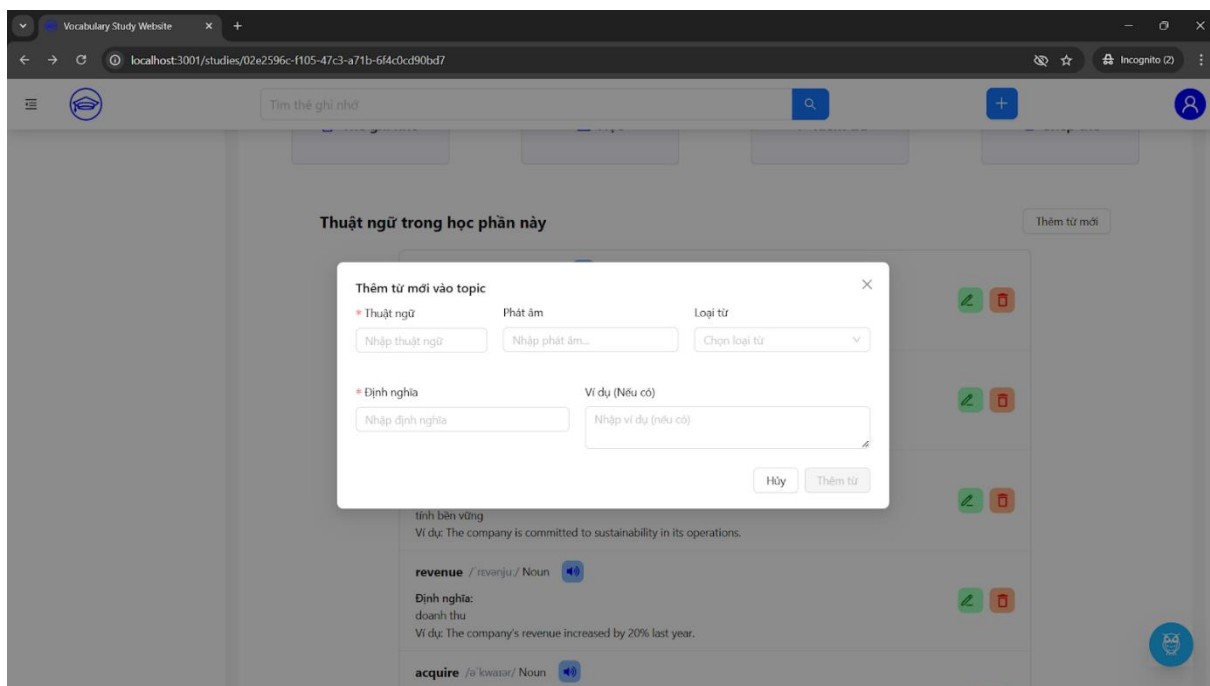
Hình 5.16. Giao diện tìm kiếm



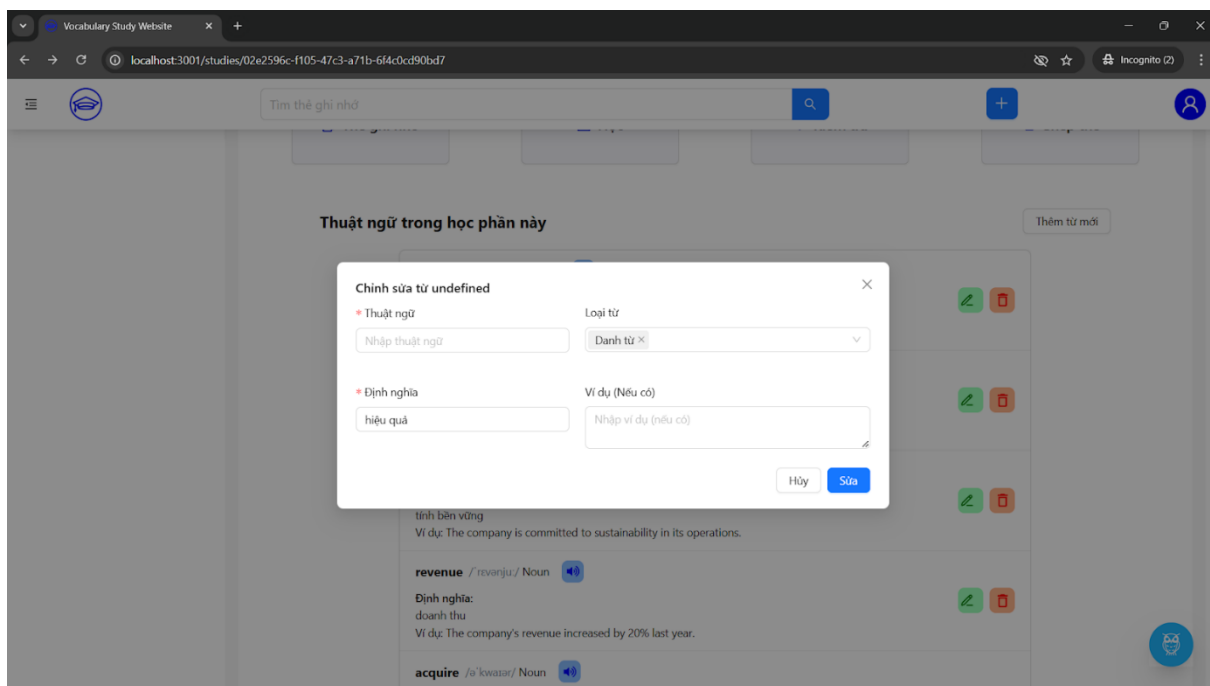
Hình 5.17. Giao diện học từ vựng bằng flashcard



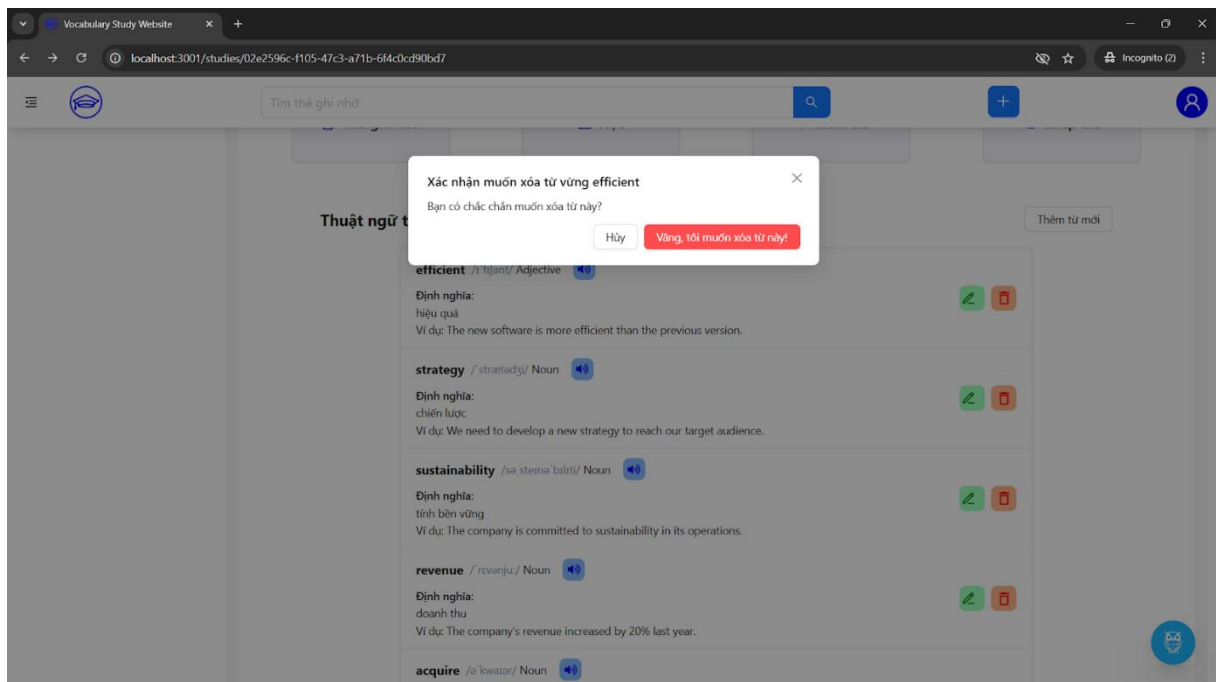
Hình 5.18. Giao diện từ điển



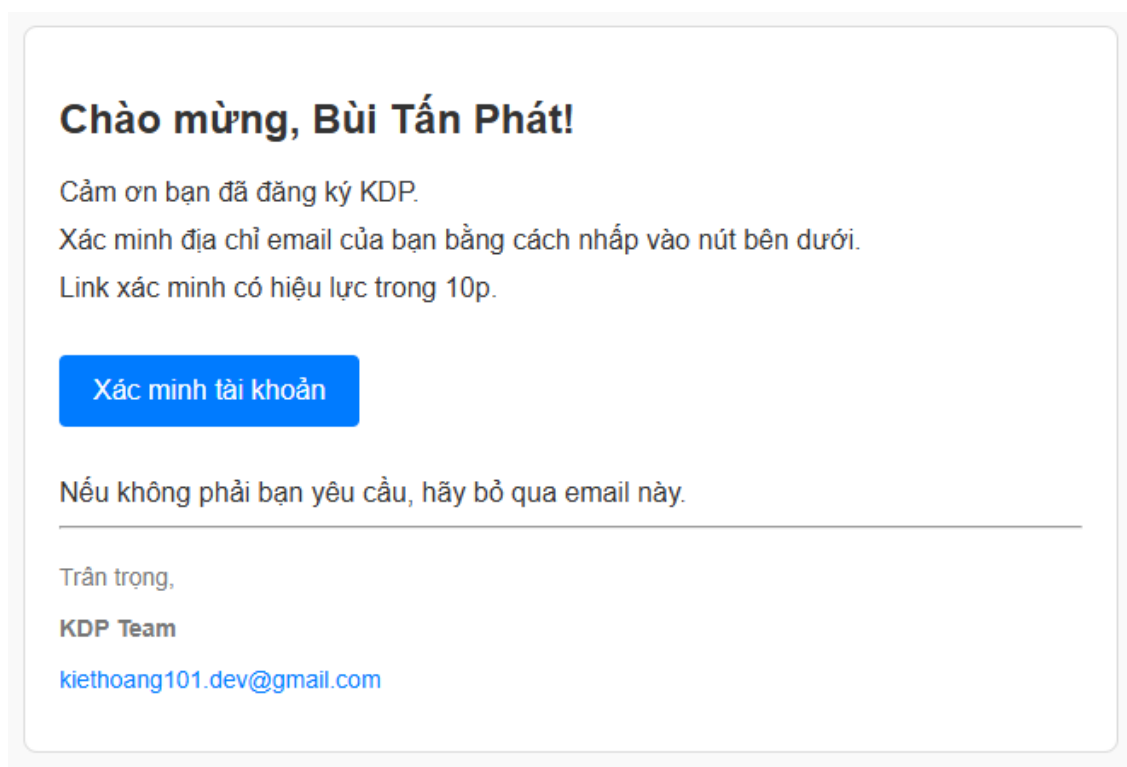
Hình 5.19. Giao diện thêm từ vựng của học phần



Hình 5.20. Giao diện chỉnh từ vựng của học phần



Hình 5.21. Giao diện xóa từ vựng học phần



Hình 5.22. Giao diện email để xác thực tài khoản

Chào, Bùi Tấn Phát!

Ai đó đã yêu cầu đổi mật khẩu của bạn. Nhấp vào nút bên dưới để đổi mật khẩu.

Chú ý: Có hiệu lực trong 30p.

[Đổi mật khẩu](#)

Nếu bạn không yêu cầu điều này, vui lòng bỏ qua email. Mật khẩu của bạn sẽ được an toàn và không bị thay đổi.

Trân trọng,

KDP Team

kiethoang101.dev@gmail.com

Hình 5.23. Giao diện email để đổi mật khẩu

KẾT LUẬN

1. Kết quả đạt được

Dự án “Xây dựng website học từ vựng tiếng Anh tích hợp chatbot” đã hoàn thành các mục tiêu chính đề ra.

Website cung cấp giao diện thân thiện và dễ sử dụng, cho phép người dùng học từ vựng thông qua các học phần mà người dùng đã tạo, tích hợp các chức năng tương tác nhằm tăng cường ghi nhớ.

Chức năng chatbot hỗ trợ người học bằng cách giải đáp thắc mắc về từ vựng và ngữ pháp tiếng anh.

2. Hạn chế

Hệ thống chưa được kiểm tra và deploy trên môi trường thực tế.

Chatbot hiện chỉ hỗ trợ các câu hỏi cơ bản và chưa xử lý tốt các tình huống phức tạp.

Hệ thống cần tối ưu hóa thêm về tốc độ phản hồi và khả năng xử lý khi có nhiều người dùng truy cập đồng thời.

Chưa hỗ trợ nhiều phương pháp học khác nhau.

3. Hướng phát triển

Hệ thống chatbot sẽ được nâng cấp để xử lý ngôn ngữ tự nhiên hiệu quả hơn, hỗ trợ các tình huống giao tiếp phức tạp hơn.

Dự án cũng sẽ tích hợp thêm các tính năng như thống kê tiến độ học tập.

Chức năng đăng nhập sẽ hỗ trợ đăng nhập bằng nhiều phương thức khác nhau (Google, Facebook...).

Deploy website lên môi trường thực tế.

PHỤ LỤC

Phụ lục 1: Mã nguồn (Source code)

- **Backend:** https://github.com/K1ethoang/BE_Learn-Vocabulary_KDP
- **Frontend:** https://github.com/K1ethoang/FE_Learn-Vocabulary_KDP
- **Chatbot:** https://github.com/krphat/RAG_with_LlamaIndex

Phụ lục 2: Hướng dẫn cài đặt

- **Cài đặt Frontend:** https://github.com/K1ethoang/FE_Learn-Vocabulary_KDP/blob/main/README.md
- **Cài đặt Backend:** https://github.com/K1ethoang/BE_Learn-Vocabulary_KDP/blob/main/README.md
- **Cài đặt Chatbot:** https://github.com/krphat/RAG_with_LlamaIndex/blob/main/README.md

TÀI LIỆU THAM KHẢO

- [1] [Online]. Available: <https://fptshop.com.vn/tin-tuc/danh-gia/quy-trinh-phat-trien-phan-mem-172036>.
- [2] [Online]. Available: <https://fmit.vn/tin-tuc/agile-la-gi>.
- [3] [Online]. Available: <https://200lab.io/blog/agile-la-gi-scrum-la-gi>.
- [4] Masteringbackend, "Masteringbackend," [Online]. Available: <https://masteringbackend.com/posts/spring-boot#spring-boot-the-framework>. [Accessed 10 2024].
- [5] javatpoint, "javatpoint," [Online]. Available: <https://www.javatpoint.com/spring-boot-architecture>. [Accessed 10 2024].
- [6] [Online]. Available: <https://www.simplilearn.com/tutorials/reactjs-tutorial/what-is-reactjs>.
- [7] Quest, "Quest," [Online]. Available: <https://www.quest.com/learn/what-is-my-sql.aspx>. [Accessed 11 2024].
- [8] [Online]. Available: <https://www.ibm.com/think/topics/vector-database>.
- [9] [Online]. Available: [https://en.wikipedia.org/wiki/Python_\(programming_language\)](https://en.wikipedia.org/wiki/Python_(programming_language)).
- [10] "java," [Online]. Available: https://www.java.com/en/download/help/whatis_java.html.
- [11] [Online]. Available: <https://funix.edu.vn/hoi-dap-cntt/uu-diem-vanhuoc-diem-cua-ngon-ngu-lap-trinh-java-cac-ung-dung-ngon-ngu-java>.
- [12] "amazon," [Online]. Available: <https://aws.amazon.com/what-is/javascript>.

- [13] Q. V. M. Research, "Vietnam language learning behaviors," 2021.
- [14] "Báo cáo thị trường - Nhu cầu học tiếng Anh của người lớn," 2022.
- [15] "LlamaIndex," [Online]. Available: <https://www.llamaindex.ai>.