

TRƯỜNG ĐẠI HỌC KIẾN TRÚC ĐÀ NẴNG
KHOA CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN TỐT NGHIỆP
NGÀNH: CÔNG NGHỆ THÔNG TIN

ĐỀ TÀI:

XÂY DỰNG HỆ THỐNG HỌC NGOẠI NGỮ
TÍCH HỢP TRÍ TUỆ NHÂN TẠO

Người hướng dẫn: **TS. Nguyễn Năng Hùng Vân**
Sinh viên thực hiện: **Bùi Quý Việt**
Mã số sinh viên: **2151220253**
Lớp: **21CT5**

Đà Nẵng,/2025

TRƯỜNG ĐẠI HỌC KIẾN TRÚC ĐÀ NẴNG
KHOA CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN TỐT NGHIỆP
NGÀNH: CÔNG NGHỆ THÔNG TIN

ĐỀ TÀI:

XÂY DỰNG HỆ THỐNG HỌC NGOẠI NGỮ
TÍCH HỢP TRÍ TUỆ NHÂN TẠO

Người hướng dẫn: **TS. Nguyễn Năng Hùng Vân**
Sinh viên thực hiện: **Bùi Quý Việt**
Mã số sinh viên: **2151220253**
Lớp: **21CT5**

Đà Nẵng,/2025

{Trang trắng này dùng để dán bản Nhận xét của người phản biện, hoặc thay trang này bằng Nhận xét của người phản biện}

TÓM TẮT

Tên đề tài: Xây dựng hệ thống học ngoại ngữ tích hợp trí tuệ nhân tạo

Sinh viên thực hiện: Bùi Quý Việt

Mã số SV: 2151220253 Lớp: 21CT5

Đề tài “Xây dựng hệ thống học ngoại ngữ tích hợp trí tuệ nhân tạo” được thực hiện trong bối cảnh nhu cầu học ngoại ngữ ngày càng tăng, trong khi nhiều nền tảng hiện nay vẫn chủ yếu cung cấp nội dung tĩnh, thiếu cá nhân hóa và chưa tận dụng tốt các kỹ thuật AI hiện đại. Người học thường khó luyện phát âm, thiếu môi trường hội thoại thực tế, khó duy trì động lực và chọn nội dung phù hợp với trình độ. Do đó, việc xây dựng một hệ thống học ngoại ngữ tích hợp AI để hỗ trợ bốn kỹ năng nghe – nói – đọc – viết và cá nhân hóa lộ trình học tập là hướng tiếp cận mang tính thực tiễn.

Mục tiêu của đề tài là thiết kế và xây dựng một hệ thống học ngoại ngữ trực tuyến hỗ trợ nhiều ngôn ngữ, trong đó tiếng Anh được tổ chức theo các cấp độ từ A1 đến C2, đồng thời tích hợp các mô hình AI như nhận dạng giọng nói (Speech-to-Text), tổng hợp giọng nói (Text-to-Speech), mô hình gợi ý và mô hình chấm điểm. Hệ thống cho phép người học đăng ký tài khoản, lựa chọn ngôn ngữ, tham gia lộ trình học được tổ chức theo chủ đề – bài học – kỹ năng, luyện phát âm với phản hồi tự động, luyện hội thoại theo kịch bản mô phỏng tình huống thực tế, ôn tập từ vựng, theo dõi tiến độ học tập và tham gia bảng xếp hạng, huy hiệu để tăng động lực.

Về mặt kỹ thuật, hệ thống được xây dựng theo kiến trúc web nhiều lớp với mô hình RESTful API: backend cung cấp các API phục vụ quản lý người dùng, nội dung bài học, tiến độ và dữ liệu tương tác; frontend tiêu thụ các API này để xây dựng giao diện người dùng thân thiện, hỗ trợ đa thiết bị. Các mô-đun trí tuệ nhân tạo được tách thành các dịch vụ riêng (microservice) đảm nhiệm chuyển giọng nói thành văn bản, tổng hợp giọng nói tự nhiên, đánh giá phát âm, sinh gợi ý nội dung và hỗ trợ hội thoại. Cơ sở dữ liệu lưu trữ có cấu trúc thông tin về ngôn ngữ, bài học, kỹ năng, lịch sử làm bài, lỗi thường gặp, làm đầu vào cho các thuật toán gợi ý và phân tích. Hệ thống tích hợp cơ chế gamification (XP, badge, bảng xếp hạng, bạn bè) nhằm duy trì và tăng cường mức độ gắn bó của người học.

LỜI NÓI ĐẦU

Trong bối cảnh toàn cầu hóa và hội nhập quốc tế ngày càng sâu rộng, ngoại ngữ đã trở thành kỹ năng thiết yếu đối với mỗi cá nhân trong học tập, công việc và đời sống. Cùng với đó, sự phát triển mạnh mẽ của công nghệ thông tin và trí tuệ nhân tạo (AI) đã mở ra nhiều cơ hội đổi mới trong giáo dục ngôn ngữ. Tuy nhiên, phần lớn các nền tảng học ngoại ngữ hiện nay vẫn chủ yếu cung cấp nội dung tĩnh, chưa khai thác hiệu quả các mô hình AI để cá nhân hóa lộ trình học, hỗ trợ phát âm, luyện hội thoại và theo dõi tiến bộ một cách toàn diện.

Từ thực tiễn này, đề tài “Xây dựng hệ thống học ngoại ngữ tích hợp trí tuệ nhân tạo” được lựa chọn làm đề án tốt nghiệp, với mục tiêu xây dựng một hệ thống học ngoại ngữ trực tuyến hỗ trợ đa ngôn ngữ, trong đó tập trung triển khai lộ trình học tiếng Anh. Hệ thống tích hợp các mô hình AI như nhận dạng giọng nói, tổng hợp giọng nói, mô hình gợi ý và mô hình chấm điểm; đồng thời cung cấp các chức năng luyện đủ bốn kỹ năng nghe, nói, đọc, viết, luyện phát âm với phản hồi tự động, luyện hội thoại theo tình huống thực tế, ôn tập từ vựng bằng thuật toán lặp lại ngắt quãng, và áp dụng gamification để duy trì động lực học tập.

Về mặt học thuật, đề tài góp phần minh chứng khả năng ứng dụng các kỹ thuật AI hiện đại trong giáo dục ngoại ngữ thông qua việc thiết kế mô hình dữ liệu, xây dựng RESTful API, tổ chức kiến trúc hệ thống nhiều lớp và tích hợp các dịch vụ AI dạng microservice. Về mặt thực tiễn, sản phẩm có thể được xem như một nền tảng thử nghiệm có tính mở rộng cao, sẵn sàng nâng cấp mô hình AI, bổ sung ngôn ngữ mới hoặc mở rộng nội dung học trong tương lai.

Trong quá trình thực hiện, em nhận được sự hướng dẫn tận tình của quý thầy cô. Em xin gửi lời cảm ơn chân thành đến quý thầy cô bộ môn, đặc biệt là giảng viên hướng dẫn, đã hỗ trợ, định hướng chuyên môn và cung cấp những góp ý quý báu để em hoàn thành đề tài. Do hạn chế về kiến thức và kinh nghiệm thực tiễn, luận văn không tránh khỏi thiếu sót. Em rất mong nhận được ý kiến đóng góp của quý thầy cô để đề tài được hoàn thiện hơn.

Em xin trân thành cảm ơn!

CAM ĐOAN

Em xin cam đoan rằng tất cả các kết quả và thông tin trình bày trong đề tài này là thành quả của công việc của em dưới sự hướng dẫn của TS. Nguyễn Năng Hùng Vân. Em đã tuân thủ mọi nguyên tắc và quy định về nghiên cứu khoa học và đạo đức trong việc thực hiện khóa luận này. Tất cả các nguồn tham khảo và công trình nghiên cứu của người khác đã được em trích dẫn và trình bày một cách rõ ràng và chi tiết trong danh mục tài liệu tham khảo của khóa luận. Em đã tuân thủ nguyên tắc không sao chép, không vi phạm bản quyền, và đã tuân thủ đúng quy định về việc chỉ rõ nguồn gốc thông tin từ nguồn khác. Các kết quả đánh giá và thống kê được trình bày trong khóa luận này đều là kết quả của những thực nghiệm mà em đã tự thực hiện, và không có sự can thiệp từ bất kỳ nguồn nào khác. Em cam đoan tính trung thực và chính xác trong việc trình bày kết quả và thông tin trong khóa luận này

Đà Nẵng, ngày tháng năm 2025

Sinh viên thực hiện

Bùi Quý Việt

MỤC LỤC

TÓM TẮT

LỜI NÓI ĐẦU	i
-------------	---

CAM ĐOAN	ii
----------	----

MỤC LỤC	iii
---------	-----

DANH SÁCH CÁC BẢNG, HÌNH VẼ	vii
-----------------------------	-----

DANH SÁCH CÁC KÝ HIỆU, CHỮ VIẾT TẮT	xiii
-------------------------------------	------

MỞ ĐẦU	1
--------------	---

CHƯƠNG 1: CƠ SỞ LÝ THUYẾT	7
---------------------------------	---

1.1 Tổng quan về đề tài	7
-------------------------------	---

1.2 Giới thiệu hệ thống gợi ý [8][54]	11
---	----

1.2.1 Lọc cộng tác dựa trên vùng lân cận	12
--	----

1.2.2 Lọc theo nội dung	16
-------------------------------	----

1.3 Giới thiệu mô hình chuyển đổi văn bản thành giọng nói	18
---	----

1.3.1. Tổng quan về công nghệ chuyển đổi văn bản thành giọng nói	18
--	----

1.3.2 Giới thiệu về mô hình Piper	19
---	----

1.3.3. Dịch vụ Google Text-to-Speech	20
--	----

1.3.4. Vai trò của TTS trong học ngoại ngữ tích hợp AI	21
--	----

1.4 Giới thiệu mô hình chuyển đổi giọng nói thành văn bản	22
---	----

1.4.1 Giới thiệu về mô hình Whisper	22
---	----

1.4.2 Phân tích về mô hình Whisper	22
--	----

1.5 Tổng quan về xử lý ngôn ngữ tự nhiên	25
--	----

1.5.1 Giới thiệu về xử lý ngôn ngữ tự nhiên	25
---	----

1.5.2 Các tác vụ chính trong xử lý ngôn ngữ tự nhiên	26
--	----

1.5.3 Các Thuật Toán và Chỉ Số trong NLP và Đánh Giá Phát Âm	29
1.5.4 Ứng dụng của NLP trong giáo dục	31
1.6 Tổng quan về mô hình Ngôn ngữ Lớn.....	33
1.6.1 Giới thiệu về mô hình Ngôn ngữ Lớn	33
1.6.2 Kiến trúc kỹ thuật cơ bản của LLM.....	33
CHƯƠNG 2: PHƯƠNG PHÁP ĐỀ XUẤT XÂY DỰNG HỆ THỐNG HỌC	
NGOẠI NGỮ TÍCH HỢP TRÍ TUỆ NHÂN TẠO.....	37
2.1 Tổng quan về phương pháp tiếp cận	37
2.2 Kiến trúc xử lý thông tin	38
2.2.1. Quy trình xử lý hội thoại thông minh.....	38
2.2.2. Quy trình đánh giá phát âm	39
2.2.3 Thu thập và tạo dữ liệu	40
2.3 Tiền xử lý và phân đoạn dữ liệu	42
2.3.1 Tiền xử lý văn bản.....	42
2.3.2 Tiền xử lý dữ liệu âm thanh.....	44
2.4 So sánh và lựa chọn mô hình AI	45
2.4.1. Lựa chọn mô hình Embedding.....	45
2.4.2. Lựa chọn mô hình LLM.....	46
2.5 Thuật toán và kỹ thuật.....	47
2.5.1 Đánh giá phát âm	47
2.5.2 Tìm kiếm ngữ liệu bằng vector	49
2.5.3 Ưu tiên gợi ý dựa trên trọng số.....	49
2.5.4 Đánh giá mô hình gợi ý.....	49
2.6 Kết quả thực nghiệm	51
2.6.1 Thực nghiệm mô hình chuyển văn bản thành giọng nói.....	51
2.6.2. Thực nghiệm mô hình chấm điểm phát âm	52
2.6.3 Thực nghiệm hệ gợi ý.....	54

CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ HỆ THỐNG	57
3.1 Xác định yêu cầu	57
3.2 Phân tích yêu cầu.....	58
3.2.1. <i>Phía người học</i>	<i>58</i>
3.2.2. <i>Phía quản trị viên</i>	<i>59</i>
3.3 Mô hình hóa yêu cầu	61
3.3.1 <i>Sơ đồ usecase tổng quát.....</i>	<i>61</i>
3.3.2 <i>Chức năng luyện hội thoại, phát âm.....</i>	<i>61</i>
3.3.3 <i>Chức năng làm bài.....</i>	<i>65</i>
3.3.4 <i>Chức năng ghi danh ngôn ngữ.....</i>	<i>67</i>
3.3.5 <i>Chức năng bảng xếp hạng.....</i>	<i>69</i>
3.3.6 <i>Chức năng bạn bè.....</i>	<i>71</i>
3.3.7 <i>Chức năng nhận gợi ý học</i>	<i>75</i>
3.3.8 <i>Chức năng xem tiến độ & thống kê</i>	<i>77</i>
3.3.9 <i>Quản lý tài khoản.....</i>	<i>80</i>
3.3.10 <i>Quản lý mô hình và AI huấn luyện</i>	<i>82</i>
3.3.11 <i>Quản lý nội dung</i>	<i>84</i>
3.3.12 <i>Quản lý kịch bản.....</i>	<i>86</i>
3.4 Thiết kế cơ sở dữ liệu	88
CHƯƠNG 4 TRIỂN KHAI HỆ THỐNG	106
4.1 Công cụ và công nghệ triển khai.....	106
4.1.1 <i>Django.....</i>	<i>106</i>
4.1.2 <i>Django Rest Framework.....</i>	<i>107</i>
4.1.3 <i>Celery.....</i>	<i>107</i>
4.1.4 <i>Reactjs vite</i>	<i>108</i>
4.1.5 <i>Docker</i>	<i>108</i>

4.1.6 MinIO	109
4.1.7 Redis.....	109
4.1.8 Triển khai hệ thống.....	110
4.2 Xây dựng backend phục vụ nghiệp vụ học ngoại ngữ	111
4.2.1 Cấu trúc dự án và module.....	111
4.3 Giao diện hệ thống.....	113
KẾT LUẬN	120
TÀI LIỆU THAM KHẢO	124
PHỤ LỤC 1	

DANH SÁCH CÁC BẢNG, HÌNH VẼ

Hình 1. 1: Utility matrix dựa trên số sao một user rate cho một item [54]	13
Hình 1. 2: Mô tả User-user Collaborative Filtering. a) Utility Matrix ban đầu. b) Utility Matrix đã được chuẩn hoá. c) User similarity matrix. d) Dự đoán các (normalized) ratings còn thiếu. e) Ví dụ về cách dự đoán normalized rating của u_1 cho i_1 . f) Dự đoán các (denormalized) ratings còn thiếu. [54]	14
Hình 1. 3: Giả sử feature vector cho mỗi item được cho trong cột cuối cùng. Với mỗi user, chúng ta cần tìm một mô hình.	17
Hình 1. 4: Mô hình Biến đổi chuỗi-trình tự (seq-2-seq) được huấn luyện trên nhiều tác vụ xử lý giọng nói khác nhau. [28].....	23
Hình 1. 5: Đầu vào đầu ra dữ liệu đào tạo ASR.....	23
Hình 1. 6: Dòng suy luận & định dạng token của Whisper	24
Hình 1. 7: Minh họa cách tokenization.....	26
Hình 1. 8: Minh họa quá trình mỗi từ được gán một thẻ dựa trên vai trò ngữ pháp của nó trong câu [32].....	27
Hình 1. 9: Minh họa dependency parsing (head->dependent) [33]	28
Hình 1. 10: Ví dụ về input & output của GEC.....	29
Hình 1. 11: Dòng thời gian phát triển của các LLM tiêu biểu (GPT, Llama, Gemini...)	33
Hình 1. 12: Kiến trúc Transformer	34
Hình 2. 1: Kiến trúc RAG.....	38
Hình 2. 2: pipeline quy trình đánh giá phát âm.....	39
Hình 2. 3: Sơ đồ hoạt động tiền xử lý và nhúng văn bản	42
Hình 2. 4: Sơ đồ hoạt động của quy trình tính điểm âm thanh	44
Hình 2. 5: Biểu đồ độ trễ TTS.....	51
Hình 2. 6: Biểu đồ phân phối điểm tổng.....	53

Hình 2. 7: Biểu đồ giá trị trung bình WER và CER.....	53
Hình 2. 8: Biểu đồ mối quan hệ giữa WER và điểm phát âm	54
Hình 2. 9 : a, Biểu đồ Độ chính xác phân loại. b, Biểu đồ Precision-Recall.....	55
Hình 2. 10: Biểu đồ phân phối xác suất dự đoán.....	55
Hình 3. 1: Sơ đồ usecase tổng quát.....	61
Hình 3. 2: Sơ đồ usecase chức năng hội thoại	61
Hình 3. 3: Biểu đồ tuần tự chức năng hội thoại	63
Hình 3. 4: Sơ đồ usecase chức năng làm bài.....	65
Hình 3. 5: Sơ đồ tuần tự chức năng làm bài.....	66
Hình 3. 6: usecase chức năng ghi danh ngôn ngữ	67
Hình 3. 7: Sơ đồ tuần tự chức năng ghi danh ngôn ngữ	68
Hình 3. 8: usecase chức năng bảng xếp hạng	69
Hình 3. 9: Sơ đồ tuần tự chức năng bảng xếp hạng	70
Hình 3. 10: usecase chức năng bạn bè.....	71
Hình 3. 11: sơ đồ tuần tự chức năng quản lý bạn bè	73
Hình 3. 12: usecase chức năng nhận gợi ý học	75
Hình 3. 13: Sơ đồ tuần tự chức năng gợi ý học.....	76
Hình 3. 14: usecase chức năng xem tiến độ & thống kê	77
Hình 3. 15: Sơ đồ tuần tự xem tiến độ mà thống kê.....	79
Hình 3. 16: Usecase chức năng quản lý tài khoản	80
Hình 3. 17: Sơ đồ tuần tự quản lý tài khoản	81
Hình 3. 18: Usecase quản lý mô hình và AI huấn luyện	82
Hình 3. 19: Sơ đồ tuần tự chức năng quản lý mô hình AI và huấn luyện	83
Hình 3. 20: Usecase quản lý nội dung.....	84
Hình 3. 21: sơ đồ tuần tự chức năng quản lý nội dung.....	86
Hình 3. 22: Usecase quản lý kịch bản	86

Hình 3. 23: Usecase quản lý kịch bản	88
Hình 3. 24: Sơ đồ thực thể quan hệ	105
Hình 4.1: Logo django.....	106
Hình 4. 2: logo django rest framework.....	107
Hình 4. 3: logo celery	107
Hình 4. 4: logo vite và reactjs	108
Hình 4. 5: Logo docker	108
Hình 4. 6: logo minIO.....	109
Hình 4. 7: logo Redis	109
Hình 4. 8: Giao diện mở đầu	113
Hình 4. 9: Giao diện đăng nhập.....	113
Hình 4. 10: Giao diện đăng kí	114
Hình 4. 11: Giao diện đăng kí ngôn ngữ mới	114
Hình 4. 12: Giao diện bài học	115
Hình 4. 13: Giao diện luyện phát âm.....	115
Hình 4. 14: Giao diện luyện tập, thống kê của user	116
Hình 4. 15: Giao diện bảng xếp hạng	116
Hình 4. 16: Giao diện Nhiệm vụ và huy hiệu	117
Hình 4. 17: Giao diện hồ sơ người dùng.....	117
Hình 4. 18: Giao diện bạn bè	118
Hình 4. 19: Giao diện admin	118
Hình 4. 20: Bong bóng hội thoại	119
Bảng 1. 1: So sánh một số ứng dụng học ngoại ngữ.....	8
Bảng 1. 2: Ưu điểm, hạn chế của piperTTS.....	20
Bảng 1. 3: Ưu điểm & nhược điểm của mô hình Whisper	25
Bảng 2. 1: Loại dữ liệu trong hệ thống	40

Bảng 2. 2: So sánh mô hình embedding	46
Bảng 2. 3: So sánh các mô hình LLM cho tác vụ Chat	46
Bảng 2. 4: Độ trễ theo từng mô hình Whisper	52
Bảng 3. 1: Mô tả usecase luyện hội thoại.....	61
Bảng 3. 2: Mô tả usecase chức năng làm bài	65
Bảng 3. 3: Mô tả usecase chức năng ghi danh ngôn ngữ	67
Bảng 3. 4: Mô tả usecase chức năng bảng xếp hạng	69
Bảng 3. 5: Mô tả usecase chức năng quản lý bạn bè.....	71
Bảng 3. 6: Mô tả usecase chức năng gợi ý.....	75
Bảng 3. 7: Mô tả usecase xem tiến độ & thống kê.....	77
Bảng 3. 8: Mô tả usecase quản lý tài khoản.....	80
Bảng 3. 9: Mô tả usecase quản lý mô hình và AI huấn luyện	82
Bảng 3. 10: Mô tả usecase quản lý nội dung	84
Bảng 3. 11: Usecase quản lý kịch bản	87
Bảng 3. 12: Bảng User Lưu trữ thông tin tài khoản người dùng.	89
Bảng 3. 13: Bảng Languages Danh sách các ngôn ngữ mà hệ thống hỗ trợ giảng dạy.	90
Bảng 3.14: Bảng LanguageEnrollment Lưu trữ thông tin ghi danh của người dùng vào một ngôn ngữ cụ thể (tiến độ tổng quan).....	90
Bảng 3.15: Bảng Lesson Định nghĩa các bài học cụ thể thuộc về một chủ đề.	91
Bảng 3.16: Bảng Skill Định nghĩa các kỹ năng hoặc bài tập nhỏ trong bài học.	91
Bảng 3.17: Bảng Word Kho từ vựng của hệ thống.	92
Bảng 3.18: Bảng Misstake Lưu trữ lịch sử các lỗi sai của người dùng.....	92
Bảng 3.19: Bảng LearningInteraction chi tiết mọi tương tác học tập phân tích hành vi.	93
Bảng 3.20: Bảng Topic Định nghĩa các chủ đề lớn trong lộ trình học.....	94

Bảng 3. 21: Bảng SessionAnswer Lưu trữ câu trả lời chi tiết của người dùng trong một phiên làm bài.	94
Bảng 3. 22: Bảng RolePlayScenario Định nghĩa các kịch bản hội thoại nhập vai.	95
Bảng 3. 23: Bảng RolePlayBlock Các khối nội dung chi tiết trong kịch bản Roleplay.	96
Bảng 3. 24: Bảng PronAttempt Lưu trữ chi tiết từng lần thu âm luyện phát âm để chấm điểm AI.	97
Bảng 3. 25: Bảng TopicProgress Lưu trữ tiến độ học tập của người dùng theo từng chủ đề lớn.....	97
Bảng 3. 26: Bảng LessonSession Lưu trữ thông tin chi tiết về một phiên làm bài học (một lần người dùng vào học một bài).	98
Bảng 3. 27: Bảng UserSkillStats Lưu trữ thống kê chi tiết về độ thành thạo từng kỹ năng (Skill) của người dùng.	99
Bảng 3. 28: Bảng DailyXP Theo dõi XP kiếm được theo ngày để vẽ biểu đồ hoạt động.	99
Bảng 3. 29: Bảng Friend Lưu trữ mối quan hệ bạn bè giữa các người dùng.	100
Bảng 3. 30: Bảng LeaderboardEntry Lưu trữ dữ liệu bảng xếp hạng thi đua.	100
Bảng 3. 31: Bảng UserBadge Lưu trữ các huy hiệu mà người dùng đã đạt được.	100
Bảng 3. 32: Bảng Badge Định nghĩa các huy hiệu/thành tích trong hệ thống.....	101
Bảng 3. 33: Bảng Notification Lưu trữ thông báo gửi đến người dùng.	101
Bảng 3. 34: Bảng KnownWord Lưu trữ từ vựng người dùng đã học.	101
Bảng 3. 35: Bảng PronunciationPrompt Dữ liệu câu/từ mẫu dùng cho bài tập phát âm.	102
Bảng 3. 36: Bảng SkillQuestion Câu hỏi trắc nghiệm (Quiz) trong các bài Skill.	102
Bảng 3. 37: Bảng SkillChoice Các phương án lựa chọn cho câu hỏi trắc nghiệm.	102
Bảng 3. 38: Bảng SkillGap Lưu trữ dữ liệu cho dạng bài tập điền từ vào chỗ trống.	103
Bảng 3. 39: Bảng OrderingItem Lưu trữ các mảnh ghép cho dạng bài sắp xếp lại câu.	103

Bảng 3. 40: Bảng MatchingPair Lưu trữ các cặp từ/nghĩa cho dạng bài tập nối cột. .	103
Bảng 3. 41: Bảng ListeningPrompt Lưu trữ dữ liệu cho bài tập luyện nghe.....	104
Bảng 3. 42: Bảng ReadingContent Lưu trữ đoạn văn bản dài cho bài tập đọc hiểu...	104
Bảng 3. 43: Bảng ReadingQuestion Lưu trữ câu hỏi liên quan đến đoạn văn đọc hiểu.	104
Bảng 3. 44: Bảng WritingQuestion Lưu trữ đề bài cho kỹ năng viết.	105
Bảng 3. 45: Bảng LessonSkill Bảng trung gian định nghĩa thứ tự các kỹ năng xuất hiện trong một bài học.	105

DANH SÁCH CÁC KÝ HIỆU, CHỮ VIẾT TẮT

KÝ HIỆU	DIỄN GIẢI
u, v	Ký hiệu đại diện cho người dùng (User)
i, j	Ký hiệu đại diện cho mục/sản phẩm (Item) hoặc bài học
$sim(u, v)$	Độ tương đồng (Similarity) giữa hai người dùng hoặc hai mục
Y	Ma trận tiện ích (Utility Matrix) chứa các đánh giá (rating)
$\bar{Y}$	Giá trị rating trung bình
$\hat{y}$	Giá trị dự đoán (Predicted rating)
$N(u, i)$	Tập hợp các người dùng lân cận (neighbors) của u đã đánh giá i
θ	Tham số của mô hình học máy cần tối ưu
L	Hàm mất mát (Loss function)
λ, α	Các tham số điều chỉnh (Regularization terms)
w, b	Trọng số (weight) và hệ số bias trong mô hình tuyến tính
Q, K, V	Các ma trận Query, Key, Value trong cơ chế Attention
d_k	Kích thước (chiều) của vector Key trong công thức Attention
$\sqrt{d_k}$	Hệ số tỉ lệ (scaling factor) trong Attention để ổn định gradient
WER	Tỷ lệ lỗi từ (Word Error Rate)
CER	Tỷ lệ lỗi ký tự (Character Error Rate)
S, D, I	Số lượng phép thay thế (Substitutions), xóa (Deletions), chèn (Insertions)
N	Tổng số phần tử (từ hoặc ký tự) trong mẫu chuẩn để tính lỗi
P	Độ chính xác (Precision)
R	Độ phủ (Recall)
F_1	Chỉ số F1-score (trung bình điều hòa của Precision và Recall)
$\cap$	Phép giao của hai tập hợp (Intersection)
$\cup$	Phép hợp của hai tập hợp (Union)
q	Vector biểu diễn câu truy vấn (Query vector)
d	Vector biểu diễn tài liệu (Document vector)
U_{time}	Độ khẩn cấp theo thời gian (Review Urgency)
P_{error}	Áp lực từ lỗi sai (Error Pressure)
G_{prof}	Khoảng cách đến mức thành thạo (Proficiency Gap)
$G_{recency}$	Độ mới của việc luyện tập (Recency Gap)
Σ	Phép tổng (Summation)
$\in$	Quan hệ thuộc tập hợp (Element of)
$\forall$	Lượng từ với mọi (For all)

STT	CHỮ VIẾT TẮT	DIỄN GIẢI	Ý NGHĨA
1	AES	Automated Essay Scoring	Hệ thống chấm điểm bài luận tự động
2	AI	Artificial Intelligence	Trí tuệ nhân tạo
3	API	Application Programming Interface	Giao diện lập trình ứng dụng
4	ASR	Automatic Speech Recognition	Nhận dạng tiếng nói tự động
5	AUC	Area Under the Curve	Diện tích dưới đường cong (đánh giá mô hình)
6	BERT	Bidirectional Encoder Representations from Transformers	Mô hình biểu diễn mã hóa hai chiều từ Transformer
7	BLEU	Bilingual Evaluation Understudy	Chỉ số đánh giá chất lượng dịch máy
8	BPE	Byte Pair Encoding	Mã hóa cặp byte (kỹ thuật tách từ)
9	CEFR	Common European Framework of Reference	Khung tham chiếu trình độ ngôn ngữ chung châu Âu
10	CER	Character Error Rate	Tỷ lệ lỗi theo ký tự
11	CI/CD	Continuous Integration / Continuous Deployment	Tích hợp liên tục / Triển khai liên tục
12	CNN	Convolutional Neural Network	Mạng nơ-ron tích chập
13	CORS	Cross-Origin Resource Sharing	Chia sẻ tài nguyên liên miền
14	CSDL	Database	Cơ sở dữ liệu
15	CTC	Connectionist Temporal Classification	Phân loại thời gian kết nối (trong ASR)
16	DRF	Django REST Framework	Bộ khung phát triển REST API trên nền tảng Django

17	EFL	English as a Foreign Language	Tiếng Anh như một ngoại ngữ
18	FK	Foreign Key	Khóa ngoại (trong CSDL)
19	G2P	Grapheme-to-Phoneme	Chuyển đổi từ chữ viết sang âm vị
20	GBM	Gradient Boosting Machine	Máy học tăng cường Gradient
21	GEC	Grammatical Error Correction	Sửa lỗi ngữ pháp
22	GPT	Generative Pre-trained Transformer	Mô hình Transformer huấn luyện trước có khả năng sinh
23	GRU	Gated Recurrent Unit	Đơn vị hồi quy có cổng (biến thể của RNN)
24	HTTP	HyperText Transfer Protocol	Giao thức truyền tải siêu văn bản
25	HTTPS	HyperText Transfer Protocol Secure	Giao thức truyền tải siêu văn bản bảo mật
26	IPA	International Phonetic Alphabet	Bảng mẫu tự phiên âm quốc tế
27	JSON	JavaScript Object Notation	Định dạng dữ liệu đối tượng JavaScript
28	JWT	JSON Web Token	Mã thông báo web dạng JSON (dùng xác thực)
29	LLM	Large Language Model	Mô hình ngôn ngữ lớn
30	LMS	Learning Management System	Hệ thống quản lý học tập
31	LSTM	Long Short-Term Memory	Bộ nhớ ngắn hạn dài hạn (biến thể của RNN)
32	MFCC	Mel-Frequency Cepstral Coefficients	Hệ số Cepstral tần số Mel (đặc trưng âm thanh)
33	ML	Machine Learning	Học máy
34	MVT	Model-View-Template	Kiến trúc Mô hình - View - Template
35	NER	Named Entity Recognition	Nhận dạng thực thể có tên

36	NLG	Natural Language Generation	Sinh ngôn ngữ tự nhiên
37	NLP	Natural Language Processing	Xử lý ngôn ngữ tự nhiên
38	NLU	Natural Language Understanding	Hiểu ngôn ngữ tự nhiên
39	ONNX	Open Neural Network Exchange	Định dạng trao đổi mạng nơ-ron mở
40	ORM	Object-Relational Mapping	Ánh xạ đối tượng - quan hệ
41	PK	Primary Key	Khóa chính (trong CSDL)
42	POS	Part-of-Speech	Từ loại (trong ngữ pháp)
43	QA	Question Answering	Hệ thống hỏi đáp
44	RAG	Retrieval-Augmented Generation	Thế hệ tăng cường truy xuất (Kỹ thuật AI)
45	REST	Representational State Transfer	Kiến trúc truyền tải trạng thái đại diện
46	RNN	Recurrent Neural Network	Mạng nơ-ron hồi quy
47	ROC	Receiver Operating Characteristic	Đường cong đặc trưng hoạt động của bộ thu
48	ROUGE	Recall-Oriented Understudy for Gisting Evaluation	Chỉ số đánh giá tóm tắt văn bản
49	RS	Recommendation System	Hệ thống gợi ý
50	S3	Simple Storage Service	Dịch vụ lưu trữ đối tượng đơn giản
51	SOTA	State-of-the-Art	Hiện đại nhất / Tiên tiến nhất
52	SPA	Single Page Application	Ứng dụng đơn trang
53	SQL	Structured Query Language	Ngôn ngữ truy vấn có cấu trúc
54	SSML	Speech Synthesis Markup Language	Ngôn ngữ đánh dấu tổng hợp giọng nói
55	STT	Speech-to-Text	Chuyển đổi giọng nói thành văn bản
56	TTS	Text-to-Speech	Chuyển đổi văn bản thành giọng nói

57	UI/UX	User Interface / User Experience	Giao diện người dùng / Trải nghiệm người dùng
58	UML	Unified Modeling Language	Ngôn ngữ mô hình hóa thống nhất
59	UUID	Universally Unique Identifier	Mã định danh duy nhất toàn cầu
60	VAD	Voice Activity Detection	Phát hiện hoạt động giọng nói
61	VITS	Variational Inference with adversarial learning for TTS	Mô hình TTS suy luận biến thiên kết hợp học đối kháng
62	WER	Word Error Rate	Tỷ lệ lỗi theo từ
63	XP	Experience Points	Điểm kinh nghiệm (trong Gamification)

MỞ ĐẦU

1. Lý do chọn đề tài

Trong bối cảnh toàn cầu hóa và hội nhập quốc tế, ngoại ngữ đặc biệt là tiếng Anh ngày càng trở thành một trong những năng lực cốt lõi đối với sinh viên và người lao động. Nhu cầu giao tiếp, học tập, làm việc với đối tác quốc tế khiến việc sở hữu khả năng sử dụng tiếng Anh ở mức độ giao tiếp thực tế không còn là lợi thế đơn thuần mà đã trở thành yêu cầu gần như bắt buộc.

Tuy nhiên, các phương pháp học truyền thống như học qua giáo trình, lớp học trực tiếp, hoặc tự học thụ động vẫn bộc lộ nhiều hạn chế: ít môi trường luyện nói trong ngữ cảnh thực tế, khó được sửa lỗi phát âm kịp thời, thiếu cơ chế phản hồi tức thời và đánh giá khách quan, cũng như khó cá nhân hóa nội dung theo trình độ và tốc độ học của từng người.

Trong những năm gần đây, nhiều nghiên cứu đã chỉ ra rằng các hệ thống học ngoại ngữ được hỗ trợ công nghệ và có khả năng thích ứng (technology-assisted and adaptive language learning systems) giúp cá nhân hóa lộ trình học, điều chỉnh nội dung theo đặc điểm người học và cải thiện hiệu quả tiếp thu kiến thức so với cách học truyền thống [1].

Đặc biệt, các tổng quan nghiên cứu về ứng dụng công nghệ nhận diện giọng nói (Automatic Speech Recognition – ASR) cho thấy ASR có tiềm năng lớn trong việc hỗ trợ luyện phát âm và cải thiện kỹ năng nói của người học tiếng Anh như ngoại ngữ (EFL), thông qua việc ghi âm, nhận dạng, so sánh với mẫu chuẩn và cung cấp phản hồi tự động [2, 3, 24, 25].

Song song với đó, các nền tảng học ngoại ngữ sử dụng mô hình trò chơi hóa (gamification)[19] và tương tác linh hoạt như Duolingo đã chứng minh được hiệu quả ở mức độ nhất định trong việc hỗ trợ người học cải thiện vốn từ vựng, ngữ pháp và duy trì động lực học tập, khi được sử dụng đều đặn trong một khoảng thời gian đủ dài [4].

Tại Việt Nam, các nghiên cứu gần đây về việc sử dụng các công cụ AI trong tự học tiếng Anh cho sinh viên không chuyên cũng cho thấy sự cải thiện tích cực về kỹ năng nói và mức độ tự tin khi sử dụng tiếng Anh, nếu người học được hỗ trợ bởi các công cụ số phù hợp [5].

Xuất phát từ bối cảnh trên, đề tài “Xây dựng hệ thống học ngoại ngữ tích hợp trí tuệ nhân tạo” được lựa chọn với mong muốn ứng dụng các kỹ thuật AI hiện đại (ASR, TTS, NLP, cơ chế gợi ý) [20, 21] để xây dựng một hệ thống mẫu (prototype) hỗ trợ người học luyện phát âm, luyện phản xạ hội thoại, mở rộng từ vựng và tăng động lực học thông qua tương tác trực tiếp với hệ thống, góp phần giải quyết phần nào các hạn chế của phương pháp học ngoại ngữ truyền thống.

2. Mục tiêu đề tài

Đề tài hướng tới các mục tiêu cụ thể xây dựng một hệ thống mẫu hỗ trợ học ngoại ngữ (trọng tâm là tiếng Anh giao tiếp cơ bản) có tích hợp các mô-đun trí tuệ nhân tạo phục vụ luyện phát âm, luyện hội thoại và gợi ý học tập.

Thiết kế và triển khai mô-đun luyện phát âm dựa trên công nghệ nhận diện giọng nói (ASR) [2, 3, 24, 25], cho phép ghi nhận âm thanh người học, phân tích và đưa ra phản hồi khách quan về mức độ chính xác so với phát âm chuẩn. Phát triển mô-đun hội thoại đóng vai (roleplay) với tác nhân AI, tái hiện các tình huống giao tiếp thường gặp (nhà hàng, khách sạn, sân bay, lớp học, công sở, v.v.), hỗ trợ người học luyện phản xạ ngôn ngữ qua văn bản hoặc giọng nói.

Xây dựng cơ chế trò chơi hóa (gamification)[19] và bảng xếp hạng nhằm tăng động lực, khuyến khích người học duy trì thói quen luyện tập thường xuyên và cạnh tranh lành mạnh. Áp dụng phân tích dữ liệu học tập của người dùng để đề xuất lộ trình ôn luyện mang tính cá nhân hóa, tập trung vào các kỹ năng và nội dung mà người học còn hạn chế. Đánh giá sơ bộ tính khả thi và hiệu quả của hệ thống thông qua kiểm thử kỹ thuật và khảo sát nhóm người dùng thử nghiệm, từ đó rút ra các nhận xét, ưu điểm, hạn chế và định hướng hoàn thiện trong tương lai.

3. Phạm vi và đối tượng nghiên cứu

3.1. Đối tượng nghiên cứu

Đề tài tập trung vào việc xây dựng một hệ thống hỗ trợ học ngoại ngữ ứng dụng trí tuệ nhân tạo, hướng đến đối tượng người học là sinh viên, học sinh và người đi làm có nhu cầu tự học tiếng Anh giao tiếp cơ bản. Mục tiêu của hệ thống là giúp người học cải thiện phát âm, nâng cao kỹ năng nói và khả năng phản xạ hội thoại trong các tình huống thực tế.

Về mặt công nghệ, hệ thống tích hợp mô hình nhận diện giọng nói Whisper [22, 23] để ghi nhận và chuyển đổi lời nói thành văn bản, cho phép đối chiếu nội dung người học phát âm với kịch bản chuẩn và đánh giá mức độ chính xác. Đồng thời, công nghệ

tổng hợp giọng nói như gTTS [43] và Piper TTS [42] được sử dụng để tạo ra các giọng nói mẫu chuẩn nhằm hỗ trợ người học nghe, lặp lại và so sánh phát âm của bản thân với giọng đọc mẫu. Các biến thể về tốc độ, giọng nói và ngôn ngữ cũng được thử nghiệm để phù hợp với từng đối tượng sử dụng.

Hệ thống còn ứng dụng công nghệ xử lý ngôn ngữ tự nhiên và các mô hình ngôn ngữ lớn như Ollama [44] và Gemini [18]. Trong đó, Ollama [44] đảm nhận vai trò serve nội bộ các mô hình huấn luyện phản xạ hội thoại roleplay dựa trên tập dữ liệu có sẵn, còn Gemini [18] được tích hợp thông qua API để tạo hội thoại roleplay theo ngữ cảnh, sinh phản hồi hội thoại tự nhiên, phù hợp với trình độ người học. Các mô hình này cũng hỗ trợ đưa ra gợi ý câu trả lời, cách diễn đạt tốt hơn và phân tích lỗi sai khi người học gặp khó khăn.

Công nghệ phân tích dữ liệu và gợi ý học tập được triển khai nhằm theo dõi hành vi học tập như tần suất luyện tập, kỹ năng thường mắc lỗi, tiến độ học,... Từ đó, hệ thống đưa ra các gợi ý bài luyện phù hợp và xây dựng lộ trình cá nhân hóa. Giai đoạn đầu sử dụng các phương pháp thống kê và heuristic đơn giản, nhưng có khả năng mở rộng sang các kỹ thuật học máy khi điều kiện cho phép.

Hệ thống được phát triển với kiến trúc backend sử dụng Django [47] và Django REST Framework [48] để xây dựng API xử lý nghiệp vụ và kết nối với các mô hình AI. Giao diện người dùng được phát triển bằng ReactJS [49], hỗ trợ tương tác thời gian thực như luyện phát âm, phản xạ hội thoại và truy cập bảng xếp hạng. Các thành phần AI được thiết kế dưới dạng mô-đun riêng biệt, đảm bảo khả năng mở rộng, bảo trì và thay thế linh hoạt trong tương lai.

Dữ liệu được lưu trữ trong hệ quản trị cơ sở dữ liệu PostgreSQL, kết hợp với tiện ích mở rộng pgvector [45] để lưu vector nhúng. Hệ thống quản lý toàn bộ thông tin người dùng, cấu hình tài khoản, hồ sơ học tập, nội dung bài học, kịch bản hội thoại roleplay, lịch sử phiên luyện tập, kết quả phát âm, tiến độ học, bảng xếp hạng và log tương tác phục vụ cho việc phân tích và gợi ý. Ngoài ra, các file âm thanh được lưu trữ riêng và Redis [46] được sử dụng để tối ưu hiệu năng truy xuất trong phạm vi hệ thống mẫu.

3.2. Phạm vi nghiên cứu và triển khai

Do giới hạn về thời gian và khuôn khổ của đề án tốt nghiệp, phạm vi nghiên cứu và triển khai được giới hạn như sau: ngôn ngữ mục tiêu tập trung vào tiếng Anh ở mức độ cơ bản và giao tiếp đời sống, chưa đi sâu vào các nội dung chuyên ngành hoặc luyện

thi chứng chỉ chuẩn hóa. Về chức năng hệ thống, đề tài tập trung phát triển các mô-đun chính gồm: luyện phát âm, hội thoại roleplay với AI, cơ chế gamification và bảng xếp hạng, gợi ý học tập. Các chức năng nâng cao như hệ quản lý chương trình học lớn, thi cử, quản trị học vụ quy mô rộng, đa nền tảng di động – web ở mức sản phẩm thương mại chỉ được đề cập ở mức định hướng. Về mức độ triển khai, hệ thống được xây dựng ở mức mô hình mẫu (prototype), đủ để minh họa kiến trúc và kiểm chứng ý tưởng kỹ thuật, chưa hướng tới tối ưu hiệu năng ở quy mô rất lớn hoặc triển khai sản phẩm thương mại.

4. Phương pháp nghiên cứu

Để đạt được các mục tiêu đề ra, đề tài sử dụng kết hợp một số phương pháp nghiên cứu chủ yếu sau:

Phương pháp nghiên cứu tài liệu:

- Thu thập và nghiên cứu các bài báo khoa học, tổng quan nghiên cứu liên quan đến hệ thống học ngoại ngữ hỗ trợ công nghệ và hệ thống thích ứng [1].
- Tham khảo các nghiên cứu về ứng dụng công nghệ nhận diện giọng nói và tổng hợp giọng nói trong việc hỗ trợ luyện phát âm và kỹ năng nói tiếng Anh [2, 3].
- Khảo sát các nghiên cứu đánh giá hiệu quả của các ứng dụng học ngoại ngữ dựa trên gamification như Duolingo [4] và các nghiên cứu trong bối cảnh Việt Nam về việc sử dụng công cụ AI hỗ trợ tự học kỹ năng nói [5].

Trên cơ sở đó, xây dựng khung lý thuyết cho hệ thống: các khái niệm, mô hình tổng quát, các thành phần chức năng cần thiết, cũng như tiêu chí đánh giá hiệu quả hệ thống và mô hình AI được sử dụng.

Phương pháp nghiên cứu thực nghiệm:

- Sử dụng mô hình Whisper (ASR) [22], [23] để luyện tập phát âm Whisper nhận file âm thanh mp3 hoặc base64 kết hợp câu được kì vọng để chấm điểm phát âm. Tương tự phát âm mẫu cho người dùng sẽ được sinh ra từ gTTS [43], Piper TTS [42] để đưa ra giọng đọc mẫu.
- Hội thoại roleplay và gợi ý diễn đạt dựa trên các mô hình ngôn ngữ lớn (LLM) triển khai qua Ollama [44] và Gemini.[18]

Thử nghiệm tích hợp các công nghệ trên vào kiến trúc hệ thống (frontend, backend, dịch vụ AI, cơ sở dữ liệu), đánh giá tính tương thích, độ ổn định và hiệu năng ở quy mô người dùng thử nghiệm. Thực hiện các kịch bản sử dụng (use case) điển hình để kiểm chứng xem hệ thống có đáp ứng đúng yêu cầu chức năng và các luồng nghiệp vụ đã phân tích hay không.

Phân tích dữ liệu:

- Thiết kế cơ chế ghi nhận dữ liệu học tập và tương tác, bao gồm:
 - + Lịch sử phiên luyện tập (thời gian, kỹ năng, bài luyện).
 - + Kết quả luyện phát âm (điểm, mức độ đúng/sai, số lần thử).
 - + Kết quả tham gia hội thoại roleplay (mức hoàn thành, số lượt trao đổi, độ dài phát ngôn...).
 - + Chỉ số gamification (XP, cấp độ, vị trí trên bảng xếp hạng) [19].
- Sử dụng các phương pháp thống kê mô tả (tần suất, trung bình, phân phối điểm kinh nghiệm...) để:
 - + Đánh giá mức độ sử dụng hệ thống của người học.
 - + Phát hiện các mẫu hành vi học tập (ví dụ: kỹ năng nào thường sai, loại bài tập nào được sử dụng nhiều).

Cung cấp đầu vào cho các chức năng gợi ý (recommendation) [8] cơ bản, gợi ý kỹ năng bài luyện phù hợp với điểm yếu hiện tại của người học.

Đánh giá mô hình AI:

- Đối với mô hình nhận diện giọng nói (Whisper): Đánh giá chất lượng nhận dạng bằng các chỉ số thường dùng như tỷ lệ lỗi từ (Word Error Rate – WER), tỷ lệ lỗi ký tự (Character Error Rate – CER) trên một tập câu mẫu được lựa chọn phù hợp với nội dung luyện tập. So sánh kết quả nhận dạng với kịch bản chuẩn để kiểm tra mức độ phù hợp khi sử dụng vào bối cảnh luyện phát âm tiếng Anh giao tiếp cơ bản.
- Đối với mô hình tổng hợp giọng nói (gTTS, Piper TTS): Đánh giá chủ quan chất lượng âm thanh tổng hợp (độ rõ, tự nhiên, dễ nghe, tốc độ phù hợp) [42] [43] thông qua khảo sát người dùng thử nghiệm. Kiểm tra khả năng đáp ứng thời gian thực (độ trễ khi sinh giọng nói) trong quá trình người học tương tác với hệ thống.

- Đối với mô hình hội thoại (Ollama, Gemini): Đánh giá mức độ phù hợp ngữ cảnh, mạch lạc, đúng ngữ pháp và mức độ hỗ trợ học tập (giải thích lỗi, gợi ý diễn đạt tốt hơn). Kiểm tra khả năng điều chỉnh độ khó, mức độ phức tạp của hội thoại theo trình độ người học [18], [43].

5. Cấu trúc của đồ án tốt nghiệp

Nội dung báo cáo được tổ chức thành năm chương chính với các nội dung cụ thể như sau:

- Chương 1: Cơ sở lý thuyết: Trình bày tổng quan về quá trình học ngoại ngữ và những khó khăn thường gặp, giới thiệu các khái niệm cơ bản về trí tuệ nhân tạo trong giáo dục, hệ thống học ngoại ngữ có trợ giúp công nghệ, công nghệ nhận diện và tổng hợp giọng nói, xử lý ngôn ngữ tự nhiên, cũng như mô hình trò chơi hóa trong học tập.
- Chương 2: Phương pháp đề xuất Trình bày mô hình tổng thể của hệ thống học ngoại ngữ tích hợp AI mô tả các mô-đun chức năng chính (luyện phát âm, hội thoại roleplay, gamification, gợi ý lộ trình học) và cách thức áp dụng các kỹ thuật AI để giải quyết bài toán.
- Chương 3: Phân tích và thiết kế hệ thống: Phân tích yêu cầu chức năng và phi chức năng. Xây dựng các sơ đồ UML (Use Case, Activity, Sequence, Class Diagram, Component Diagram). Thiết kế cơ sở dữ liệu và kiến trúc triển khai cho hệ thống.
- Chương 4: Xây dựng hệ thống học ngoại ngữ tích hợp trí tuệ nhân tạo: Trình bày quá trình hiện thực hóa hệ thống: môi trường và công nghệ sử dụng, cấu trúc các thành phần frontend/backend/AI services, các màn hình giao diện chính. Mô tả chi tiết việc cài đặt các mô-đun luyện phát âm, hội thoại roleplay, gamification và gợi ý học tập. Chương này đồng thời giới thiệu kết quả kiểm thử và một số kịch bản sử dụng minh họa.

CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

1.1 Tổng quan về đề tài

Trong những năm gần đây, trí tuệ nhân tạo (AI) đã và đang tạo ra những thay đổi mang tính cách mạng trong lĩnh vực giáo dục ngoại ngữ. Việc tích hợp AI vào dạy và học ngoại ngữ giúp cá nhân hóa trải nghiệm học tập, cung cấp nội dung thích ứng với trình độ người học, đồng thời cho phép tương tác thời gian thực nhằm nâng cao sự tham gia và cải thiện kỹ năng lưu loát của người học [6]. Nhiều nghiên cứu chỉ ra rằng việc ứng dụng các công cụ AI trong lớp học ngoại ngữ (đặc biệt là tiếng Anh như ngôn ngữ thứ hai) đã tạo ra môi trường học tập năng động, lấy người học làm trung tâm hơn so với phương pháp truyền thống [6]. Chẳng hạn, AI có thể cung cấp phản hồi tự động về ngữ pháp và từ vựng cho người học, giúp họ tự phản ánh và điều chỉnh lỗi sai một cách kịp thời.

Sự phát triển mạnh mẽ của AI đã thúc đẩy số lượng nghiên cứu trong lĩnh vực ứng dụng AI vào học ngoại ngữ tăng nhanh đáng kể từ sau năm 2018 [7]. AI mở ra những khả năng mới trong việc hỗ trợ người học ngôn ngữ: từ các hệ thống học tập thích ứng, cơ chế phản hồi thông minh cho đến các chatbot tương tác, tất cả đều hứa hẹn cải tiến phương pháp sư phạm truyền thống [6]. Việc tích hợp AI cho phép xây dựng các hệ thống dạy kèm thông minh (Intelligent Tutoring Systems) trong lĩnh vực ngoại ngữ, giúp theo dõi tiến độ người học, tự động đánh giá câu trả lời và cung cấp gợi ý học tập phù hợp. Chẳng hạn, Duolingo một nền tảng học ngoại ngữ trực tuyến phổ biến đã áp dụng thuật toán AI để cá nhân hóa lộ trình học cho từng người dùng, thông qua bài kiểm tra xếp lớp thích ứng và cơ chế spaced repetition (lặp lại ngắt quãng) nhằm tối ưu hóa việc ghi nhớ từ vựng [8] [51]. Các nghiên cứu đã ghi nhận hiệu quả của cách tiếp cận này: AI có thể dự đoán mức độ tiếp thu từ vựng của người học dựa trên lịch sử tương tác, từ đó đưa ra bài luyện tập đúng thời điểm giúp củng cố kiến thức và duy trì động lực học tập [8]. Nhờ khả năng phân tích dữ liệu người học quy mô lớn bằng kỹ thuật học sâu, những hệ thống như Duolingo có thể liên tục thử nghiệm và tối ưu chiến lược giảng dạy, mang lại trải nghiệm học tập tốt hơn cho hàng triệu người dùng.

Bên cạnh đó, AI còn hỗ trợ tạo ra các môi trường thực hành ngôn ngữ phong phú mà trước đây khó thực hiện trong lớp học truyền thống. Ví dụ, người học có thể hội thoại với chatbot thông minh để rèn luyện kỹ năng giao tiếp, hoặc nhận phản hồi tức thì về phát âm và ngữ pháp thông qua các ứng dụng di động sử dụng AI. Những công nghệ

này không chỉ giúp người học tăng cường tương tác với ngôn ngữ mục tiêu mọi lúc, mọi nơi mà còn góp phần giảm lo lắng khi thực hành ngoại ngữ, do môi trường học tập ảo thường không có sự đánh giá khắt khe như khi giao tiếp với người thật [6]. Mặc dù vậy, việc lạm dụng công nghệ AI cũng có thể dẫn đến một số thách thức, chẳng hạn nguy cơ làm suy giảm tư duy phản biện và tính sáng tạo của người học nếu họ quá phụ thuộc vào phản hồi tự động [6]. Do đó, việc tích hợp AI trong học ngoại ngữ đòi hỏi phải có chiến lược sư phạm hợp lý, kết hợp hướng dẫn của giáo viên để đảm bảo người học sử dụng AI như một công cụ hỗ trợ chứ không thay thế hoàn toàn quá trình tư duy độc lập.

Hệ thống học ngoại ngữ tích hợp AI hứa hẹn mang lại nhiều lợi ích vượt trội: nội dung học tập được cá nhân hóa theo trình độ và sở thích của từng người, khả năng tương tác liên tục giúp duy trì động lực, và phản hồi tức thời giúp cải thiện kỹ năng ngôn ngữ hiệu quả hơn. Minh chứng thực tế cho xu hướng này là sự xuất hiện của hàng loạt ứng dụng và nền tảng học ngoại ngữ thông minh (Duolingo, ELSA Speak, v.v.) thu hút đông đảo người học trên toàn thế giới. Trong khuôn khổ đề tài này, chúng ta tập trung vào các thành phần công nghệ chính cấu thành nên hệ thống học ngoại ngữ tích hợp AI, bao gồm: hệ thống gợi ý (Recommendation System) [8] giúp đề xuất bài học phù hợp, mô hình chuyển đổi văn bản thành giọng nói (Text-to-Speech) [28] hỗ trợ phát âm và kỹ năng nghe, mô hình chuyển đổi giọng nói thành văn bản (Speech-to-Text) hỗ trợ nhận dạng tiếng nói và kiểm tra phát âm, các kỹ thuật xử lý ngôn ngữ tự nhiên (Natural Language Processing) [21] để hiểu và phân tích ngữ liệu ngôn ngữ và cuối cùng là mô hình Ngôn ngữ Lớn (Large Language Model) [15] [16] tạo nên trí tuệ hội thoại và khả năng sinh ngôn ngữ nâng cao. Các phần sau đây sẽ lần lượt trình bày cơ sở lý thuyết của những thành phần này và vai trò của chúng trong hệ thống học ngoại ngữ tích hợp AI.

Bảng 1. 1: So sánh một số ứng dụng học ngoại ngữ

Ứng dụng	Đối tượng / Mục tiêu chính	Điểm mạnh nổi bật	Hạn chế chính
Duolingo	Người học phổ thông, nhiều ngôn ngữ; tập trung A1–B1. học nhanh trên mobile	- Trò chơi hóa mạnh (streak, XP, leaderboard) giúp tăng động lực và tần suất luyện tập. - Cải thiện khá tốt từ vựng và ngữ pháp cơ bản, nội	- Luyện nói và độ trôi chảy còn hạn chế; bài tập nói chủ yếu kiểm tra phát âm đơn giản, ít hội thoại đa lượt. - Cá nhân hóa nội dung chưa sâu (chủ yếu tuyến bài

		dung phong phú, dễ tiếp cận (miễn phí).	chung, điều chỉnh độ khó ở mức hạn chế).
Babbel	Người lớn, người đi làm; muốn khóa học có cấu trúc, chú trọng giao tiếp thực tế	- Bài học được thiết kế có lộ trình rõ ràng, nhấn mạnh ngữ pháp và hội thoại thực tế trong đời sống. - Nghiên cứu thực nghiệm cho thấy cải thiện tốt từ vựng, ngữ pháp và khả năng giao tiếp trong thời gian tương đối ngắn.	- Mô hình thuê bao trả phí, rào cản với người học phổ thông. - Trò chơi hóa yếu hơn Duolingo; tính năng AI/ASR/LLM không được công bố chi tiết, mức độ hội thoại AI còn hạn chế công khai.
Memrise	Người học muốn mở rộng từ vựng nhanh, ôn tập bằng flashcard, video ngắn	- Mạnh về từ vựng, dùng lặp lại ngắt quãng (SRS) và video người bản ngữ; phù hợp ôn từ nhanh, ngắn.	- Thiếu một khóa học hoàn chỉnh về ngữ pháp, kỹ năng nói/viết, khó dùng làm chương trình chính.
Busuu	Người học muốn khóa học tương đối đầy đủ + cộng đồng sửa lỗi	- Cân bằng các kỹ năng (nghe, đọc, viết, nói) với cộng đồng người dùng hỗ trợ sửa bài, phản hồi.	- Chất lượng và tốc độ phản hồi phụ thuộc vào cộng đồng; chưa khai thác mạnh AI tự động chấm/giải thích; nội dung nâng cao còn hạn chế.
ELSA Speak	Người học tiếng Anh, tập trung luyện phát âm và intonation	- Tập trung sâu vào phát âm tiếng Anh, dùng ASR để phân tích lỗi ở mức âm/âm tiết; nhiều nghiên cứu chứng minh hiệu quả cải thiện phát âm.	- Chủ yếu cho tiếng Anh, chưa bao quát các kỹ năng khác (đọc, viết, ngữ pháp) theo lộ trình đầy đủ. - Khó sử dụng làm nền tảng học ngoại ngữ tổng thể, mà phù hợp hơn là công cụ hỗ trợ luyện phát âm.

Đề tài “Xây dựng hệ thống học ngoại ngữ tích hợp trí tuệ nhân tạo” hướng tới kết hợp có chủ đích những ưu điểm của các ứng dụng hiện có, đồng thời khắc phục một số

hạn chế theo các hướng sau: Tập trung mạnh vào kỹ năng nói và phát âm, nhưng đặt trong một hệ thống đa kỹ năng. Thay vì chỉ luyện phát âm như ELSA hoặc chỉ luyện từ vựng/ngữ pháp như nhiều ứng dụng phổ biến, hệ thống của đề tài được thiết kế để:

Kết hợp mô-đun STT (Whisper) [28] để nhận dạng tiếng nói người học, phân tích lỗi, tính toán chỉ số như Word Error Rate (WER) và Character Error Rate (CER). Mô-đun TTS (gTTS, Piper TTS) [43] [42] để cung cấp giọng nói mẫu cho mọi đơn vị nội dung (từ, câu, đoạn hội thoại). Tích hợp với bài học, kỹ năng, kịch bản hội thoại role-play và hệ thống XP/bảng xếp hạng. Nhờ vậy, kỹ năng nói và phát âm không tách rời mà được gắn liền với ngữ cảnh học thực tế, cùng với các kỹ năng nghe – đọc – từ vựng, tạo thành một hệ sinh thái học tập thống nhất.

Cá nhân hóa sâu dựa trên hệ thống gợi ý mở và dữ liệu hành vi chi tiết. Trong khi Duolingo, Babbel hay các ứng dụng khác đều có yếu tố cá nhân hóa, cơ chế này chủ yếu nằm trong hệ thống đóng và ít được mô tả chi tiết trong tài liệu khoa học [8] [14]. Đề tài hướng tới: Xây dựng mô hình dữ liệu chi tiết về người học (LessonSession, SkillSession, Mistake, UserSkillStats, XP, badge, v.v.). Thiết kế Recommendation Engine riêng (kết hợp lọc cộng tác, gợi ý dựa trên nội dung, và heuristic giáo dục) để đề xuất kỹ năng/bài học tiếp theo. Cho phép tùy biến, thử nghiệm và đánh giá các thuật toán gợi ý trong môi trường kiểm soát được (có log, có dữ liệu nền để so sánh).

Điểm mới ở đây là hệ thống không chỉ dùng gợi ý như tính năng sản phẩm, mà còn là đối tượng nghiên cứu, có thể thay đổi thuật toán, so sánh kết quả và báo cáo.

Hệ thống của đề tài lựa chọn cách tiếp cận mở:

- Sử dụng LLM (qua Ollama [44], Gemini [18]) để tạo hội thoại, giải thích lỗi, gợi ý cách diễn đạt .
- Kết hợp với RAG (Retrieval-Augmented Generation) [41] trên tập kịch bản hội thoại, cấu trúc bài học do người phát triển thiết kế, nhằm đảm bảo hội thoại bám sát nội dung giảng dạy thay vì trả lời tự do không kiểm soát.
- Cho phép thử nghiệm nhiều mô hình, cấu hình prompt, chiến lược truy hồi khác nhau, phục vụ mục tiêu nghiên cứu và tối ưu sư phạm.

Tính mới ở đây là sự kết hợp LLM + RAG + dữ liệu học ngoại ngữ được gắn nhãn (theo CEFR, chủ đề, kỹ năng) trong một hệ thống mở, thay vì chỉ sử dụng LLM như một chatbot chung chung.

1.2 Giới thiệu hệ thống gợi ý [8][54]

Các hệ thống gợi ý (Recommendation Systems – RS) hiện diện trong hầu hết các nền tảng số hiện đại mà người dùng tiếp xúc hằng ngày:

- Youtube tự động chuyển các clip liên quan đến clip bạn đang xem. Youtube cũng tự gợi ý những clip mà có thể bạn sẽ thích.
- Khi bạn mua một món hàng trên Amazon, hệ thống sẽ tự động gợi ý “Frequently bought together”, hoặc nó biết bạn có thể thích món hàng nào dựa trên lịch sử mua hàng của bạn.
- Facebook hiển thị quảng cáo những sản phẩm có liên quan đến từ khoá bạn vừa tìm kiếm.
- Facebook gợi ý kết bạn.
- Netflix tự động gợi ý phim cho người dùng.

Tất cả những ví dụ trên đều được vận hành bằng các thuật toán Machine Learning thuộc nhóm Recommender Systems. Trong các hệ thống này, hai thực thể chính là user và item. User là người dùng; item là các thực thể có thể được gợi ý (phim, bài hát, sách, bài học, người dùng khác...). Mục tiêu của hệ thống gợi ý là dự đoán mức độ quan tâm của một user đối với một item để đưa ra đề xuất phù hợp. Các Recommendation Systems thường được chia thành hai nhóm lớn:

- Content-based systems: đánh giá đặc tính của items được recommended. Ví dụ: một user xem rất nhiều các bộ phim về cảnh sát hình sự, vậy thì gợi ý một bộ phim trong cơ sở dữ liệu có chung đặc tính hình sự tới user này, ví dụ phim Người phán xử. Cách tiếp cận này yêu cầu việc sắp xếp các items vào từng nhóm hoặc đi tìm các đặc trưng của từng item. Tuy nhiên, có những items không có nhóm cụ thể và việc xác định nhóm hoặc đặc trưng của từng item đôi khi là bất khả thi.
- Collaborative filtering: hệ thống gợi ý items dựa trên sự tương quan (similarity) giữa các users và/hoặc items. Có thể hiểu rằng ở nhóm này một item được recommended tới một user dựa trên những users có hành vi tương tự. Ví dụ: users A, B, C đều thích các bài hát của Noo Phước Thịnh. Ngoài ra, hệ thống biết rằng users B, C cũng thích các bài hát của Bích Phương nhưng chưa có thông tin về việc liệu user A có thích Bích Phương hay không. Dựa trên thông tin của những users tương tự là B và C, hệ thống

có thể dự đoán rằng A cũng thích Bích Phương và gợi ý các bài hát của ca sĩ này tới A.

1.2.1 Loại cộng tác dựa trên vùng lân cận

1.2.1.1 Giới thiệu

Ý tưởng cơ bản của NBCF là xác định mức độ quan tâm của một user tới một item dựa trên các users khác gần giống với user này. Việc gần giống nhau giữa các users có thể được xác định thông qua mức độ quan tâm của các users này tới các items khác mà hệ thống đã biết. Ví dụ A, B đều thích phim Cảnh sát hình sự, tức đều rate bộ phim này 5 sao. Ta đã biết A cũng thích Người phán xử, vậy nhiều khả năng B cũng thích bộ phim này.

Hai câu hỏi quan trọng nhất trong một hệ thống Neighborhood-based Collaborative Filtering là [8][54]:

- Làm thế nào xác định được sự giống nhau giữa hai users?
- Khi đã xác định được các users gần giống nhau (similar users) rồi, làm thế nào dự đoán được mức độ quan tâm của một user lên một item?

Việc xác định mức độ quan tâm của mỗi user tới một item dựa trên mức độ quan tâm của similar users tới item đó còn được gọi là User-user collaborative filtering. Có một hướng tiếp cận khác được cho là làm việc hiệu quả hơn là Item-item collaborative filtering. Trong hướng tiếp cận này, thay vì xác định user similarities, hệ thống sẽ xác định item similarities. Từ đó, hệ thống gợi ý những items gần giống với những items mà user có mức độ quan tâm cao.

1.2.1.2 Loại cộng tác dựa trên user-user

Công việc quan trọng nhất phải làm trước tiên trong User-user Collaborative Filtering là phải xác định được sự giống nhau (similarity) giữa hai users. Dữ liệu duy nhất chúng ta có là Utility matrix Y , vậy nên sự giống nhau này phải được xác định dựa trên các cột tương ứng với hai users trong ma trận này.

	u_0	u_1	u_2	u_3	u_4	u_5	u_6
i_0	5	5	2	0	1	?	?
i_1	3	?	?	0	?	?	?
i_2	?	4	1	?	?	1	2
i_3	2	2	3	4	4	?	4
i_4	2	0	4	?	?	?	5

Hình 1. 1: Utility matrix dựa trên số sao một user rate cho một item [54]

Giả sử có các users từ u_0 đến u_6 và các items từ i_0 đến i_4 trong đó các số trong mỗi ô vuông thể hiện số sao mà mỗi user đã *rated* cho item với giá trị cao hơn thể hiện *mức độ quan tâm* cao hơn. Các dấu hỏi chấm là các giá trị mà hệ thống cần phải đi tìm. Đặt *mức độ giống nhau* của hai users u_i, u_j là $\text{sim}(u_i, u_j)$.

Quan sát đầu tiên chúng ta có thể nhận thấy là các u_0, u_1 thích i_0, i_1, i_2 và không thích i_3, i_4 cho lắm. Điều ngược lại xảy ra ở các users còn lại. Vì vậy, một *similarity function* tốt cần đảm bảo:

$$\text{sim}(u_0, u_1) > \text{sim}(u_0, u_i), \forall i > 1 \quad (1.1)$$

Từ đó, để xác định *mức độ quan tâm* của u_0 lên i_2 , chúng ta nên dựa trên hành vi của u_1 lên sản phẩm này. Rất may rằng u_1 đã thích i_2 nên hệ thống cần *recommend* i_2 cho u_0 .

Câu hỏi đặt ra là: hàm số *similarity* nào là tốt? Để đo *similarity* giữa hai users, cách thường làm là xây dựng *feature vector* cho mỗi user rồi áp dụng một hàm có khả năng đo *similarity* giữa hai vectors đó. Chú ý rằng việc xây dựng *feature vector* này khác với việc xây dựng item profile như trong Content-based Recommendation Systems. Các vectors này được xây dựng trực tiếp dựa trên Utility matrix chứ không dùng dữ liệu ngoài như item profiles. Với mỗi user, thông tin duy nhất chúng ta biết là các *ratings* mà user đó đã thực hiện, tức cột tương ứng với user đó trong Utility matrix. Tuy nhiên, khó khăn là các cột này thường có rất nhiều *missing ratings* vì mỗi user thường chỉ *rated* một số lượng rất nhỏ các items. Cách khắc phục là bằng cách nào đó, ta giúp hệ thống điền các giá trị này sao cho việc điền không làm ảnh hưởng

hiều tới sự giống nhau giữa hai vector. Việc điều này chỉ phục vụ cho việc tính *similarity* chứ không phải là *suy luận* ra giá trị cuối cùng.

Vậy mỗi dấu ‘?’ nên được thay bằng giá trị ‘0’ để hạn chế việc sai lệch quá nhiều. Điều này không thực sự tốt vì giá trị ‘0’ tương ứng với mức độ quan tâm thấp nhất. Một giá trị *an toàn* hơn là 2.5 vì nó là trung bình cộng của 0, mức thấp nhất, và 5, mức cao nhất. Tuy nhiên, giá trị này có hạn chế đối với những *users dễ tính* hoặc *khó tính*. Với các *users dễ tính*, *thích* tương ứng với 5 sao, *không thích* có thể ít sao hơn 1 chút, 3 sao chẳng hạn. Việc chọn giá trị 2.5 sẽ khiến cho các *items* còn lại là quá *negative* đối với *user* đó. Điều ngược lại xảy ra với những *user* khó tính hơn khi chỉ cho 3 sao cho các *items* họ thích và ít sao hơn cho những *items* họ không thích.

Một giá trị khả dĩ hơn cho việc này là trung bình cộng của các *ratings* mà *user* tương ứng đã thực hiện. Việc này sẽ tránh được việc *users* quá khó tính hoặc dễ tính, tức lúc nào cũng có những *items* mà một *user* thích hơn so với những *items* khác.

	u_0	u_1	u_2	u_3	u_4	u_5	u_6
i_0	5	5	2	0	1	?	?
i_1	4	?	?	0	?	2	?
i_2	?	4	1	?	?	1	1
i_3	2	2	3	4	4	?	4
i_4	2	0	4	?	?	?	5

$\bar{u}_j$	3.25	2.75	2.5	1.33	2.5	1.5	3.33
-------------	------	------	-----	------	-----	-----	------

a) Original utility matrix $\mathbf{Y}$ and mean user ratings.

	u_0	u_1	u_2	u_3	u_4	u_5	u_6
i_0	1.75	2.25	-0.5	-1.33	-1.5	0	0
i_1	0.75	0	0	-1.33	0	0.5	0
i_2	0	1.25	-1.5	0	0	-0.5	-2.33
i_3	-1.25	-0.75	0.5	2.67	1.5	0	0.67
i_4	-1.25	-2.75	1.5	0	0	0	1.67

b) Normalized utility matrix $\bar{\mathbf{Y}}$.

	u_0	u_1	u_2	u_3	u_4	u_5	u_6
u_0	1	0.83	-0.58	-0.79	-0.82	0.2	-0.38
u_1	0.83	1	-0.87	-0.40	-0.55	-0.23	-0.71
u_2	-0.58	-0.87	1	0.27	0.32	0.47	0.96
u_3	-0.79	-0.40	0.27	1	0.87	-0.29	0.18
u_4	-0.82	-0.55	0.32	0.87	1	0	0.16
u_5	0.2	-0.23	0.47	-0.29	0	1	0.56
u_6	-0.38	-0.71	0.96	0.18	0.16	0.56	1

c) User similarity matrix $\mathbf{S}$.

	u_0	u_1	u_2	u_3	u_4	u_5	u_6
i_0	1.75	2.25	-0.5	-1.33	-1.5	0.18	-0.63
i_1	0.75	0.48	-0.17	-1.33	-1.33	0.5	0.05
i_2	0.91	1.25	-1.5	-1.84	-1.78	-0.5	-2.33
i_3	-1.25	-0.75	0.5	2.67	1.5	0.59	0.67
i_4	-1.25	-2.75	1.5	1.57	1.56	1.59	1.67

d) $\hat{\mathbf{Y}}$

Predict normalized rating of u_1 on i_1 with $k = 2$

Users who rated i_1 : $\{u_0, u_3, u_5\}$

Corresponding similarities: $\{0.83, -0.40, -0.23\}$

$\Rightarrow$ most similar users: $\mathcal{N}(u_1, i_1) = \{u_0, u_5\}$

with **normalized ratings** $\{0.75, 0.5\}$

$$\Rightarrow \hat{y}_{i_1, u_1} = \frac{0.83 \cdot 0.75 + (-0.23) \cdot 0.5}{0.83 + |-0.23|} \approx 0.48$$

e) Example

	u_0	u_1	u_2	u_3	u_4	u_5	u_6
i_0	5	5	2	0	1	1.68	2.70
i_1	4	3.23	2.33	0	1.67	2	3.38
i_2	4.15	4	1	-0.5	0.71	1	1
i_3	2	2	3	4	4	2.10	4
i_4	2	0	4	2.9	4.06	3.10	5

f) Full $\hat{\mathbf{Y}}$

Hình 1. 2: Mô tả User-user Collaborative Filtering. a) Utility Matrix ban đầu. b) Utility Matrix đã được chuẩn hoá. c) User similarity matrix. d) Dự đoán các (normalized) ratings còn thiếu. e) Ví dụ về cách dự đoán normalized rating của u_1 cho i_1 . f) Dự đoán các (denormalized) ratings còn thiếu. [54]

Chuẩn hoá dữ liệu:

Hàng cuối cùng trong Hình 1.2a là giá trị trung bình của ratings cho mỗi user. Giá trị cao tương ứng với các user dễ tính và ngược lại. Khi đó, nếu tiếp tục trừ từ mỗi rating đi giá trị này và thay các giá trị chưa biết bằng 0, ta sẽ được normalized utility matrix như trong Hình 1.2b. Bước chuẩn hoá này là một bước quan trọng bởi vì:

- Việc trừ đi trung bình cộng của mỗi cột khiến trong mỗi cột có những giá trị dương và âm. Những giá trị dương tương ứng với việc user thích item, những giá trị âm tương ứng với việc user không thích item. Những giá trị bằng 0 tương ứng với việc chưa xác định được liệu user có thích item hay không.

Sau khi đã chuẩn hoá dữ liệu như trên, một vài similarity function thường được sử dụng là: Cosine Similarity đây là hàm được sử dụng nhiều nhất. Công thức tính cos của góc giữa hai vector u_1, u_2 :

$$\text{cosine similarity}(u_1, u_2) = \cos(u_1, u_2) = \frac{u_1^T u_2}{\|u_1\|_2 \cdot \|u_2\|_2} \quad (1.2)$$

Trong đó u_1, u_2 là vectors tương ứng với users 1, 2 đã được chuẩn hóa như trên. Độ similarity của hai vector là 1 số trong đoạn $[-1, 1]$. Giá trị bằng 1 thể hiện hai vector hoàn toàn similar nhau. Hàm số cos của một góc bằng 1 nghĩa là góc giữa hai vector bằng 0, tức một vector bằng tích của một số dương với vector còn lại. Giá trị cos bằng -1 thể hiện hai vector này hoàn toàn trái ngược nhau. Điều này cũng hợp lý, tức khi hành vi của hai users là hoàn toàn ngược nhau thì similarity giữa hai vector đó là thấp nhất.

Có một chú ý quan trọng ở đây là khi số lượng users lớn, ma trận S cũng rất lớn và nhiều khả năng là không có đủ bộ nhớ để lưu trữ, ngay cả khi chỉ lưu hơn một nửa số các phần tử của ma trận đối xứng này. Với các trường hợp đó, mới mỗi user, chúng ta chỉ cần tính và lưu kết quả của một hàng của similarity matrix, tương ứng với việc độ giống nhau giữa user đó và các users còn lại.

1.2.1.3 Dự đoán xếp hạng

Việc xác định mức độ quan tâm của một user lên một item dựa trên các users gần nhất (neighbor users). Khi làm việc với large-scale problems, chúng ta sẽ thấy thêm rằng phương pháp lười học K-nearest neighbors (KNN) được sử dụng rất nhiều vì tính đơn giản của nó. Tất nhiên, chúng ta không thể trực tiếp sử dụng KNN mà còn cần phải làm thêm nhiều bước trung gian nữa.

Trong Collaborative Filtering, missing rating cũng được xác định dựa trên thông tin về k neighbor users. Tất nhiên, chúng ta chỉ quan tâm tới các users đã rated

item đang xét. Predicted rating thường được xác định là trung bình có trọng số của các ratings đã chuẩn hoá. Có một điểm cần lưu ý, trong KNN, các trọng số được xác định dựa trên distance giữa 2 điểm, và các distance này là các số không âm. Trong khi đó, trong CF, các trọng số được xác định dựa trên similarity giữa hai users, những trọng số này có thể nhỏ hơn 0 như trong Hình 1.2c.

Công thức phổ biến được sử dụng để dự đoán *rating* của u cho i là:

$$\hat{y}_{i,u} = \frac{\sum_{u_j \in N(u,i)} \bar{y}_{i,u_j} \text{sim}(u,u_j)}{\sum_{u_j \in N(u,i)} |\text{sim}(u,u_j)|} \quad (1.3)$$

trong đó $N_{u,i}$ là tập hợp k users trong neighborhood (tức có similarity cao nhất) của u mà đã rated i .

Một ví dụ về việc tính normalized rating của u_1 cho i_1 được cho trong Hình 1.2e với số nearest neighbors là $k=2$. Các bước thực hiện là:

- Xác định các users đã rated i_1 , đó là u_0, u_3, u_5 .
- Xác định similarities của u_1 với các users này ta nhận được 0.83, -0.40, -0.23. Hai ($k=2$) giá trị lớn nhất là 0.83 và -0.23 tương ứng với u_0 và u_5
- Xác định các normalized ratings của u_0, u_5 cho i_1 , ta thu được hai giá trị lần lượt là 0.75 và 0.5.
- Dự đoán kết quả:

Việc quy đổi các giá trị ratings đã chuẩn hoá về thang 5 có thể được thực hiện bằng cách cộng các cột của ma trận $\hat{Y}$ với giá trị rating trung bình của mỗi *user* như đã tính trong Hình 1.2a. Việc hệ thống quyết định *recommend items* nào cho mỗi *user* có thể được xác định bằng nhiều cách khác nhau. Có thể sắp xếp *unrated items* theo thứ tự từ lớn đến bé của các *predicted ratings*, hoặc chỉ chọn các *items* có *normalized predicted ratings* dương - tương ứng với việc *user* này có nhiều khả năng thích hơn.

1.2.2 Lọc theo nội dung

1.2.2.1 Item profiles

Trong các hệ thống content-based, tức dựa trên nội dung của mỗi item, chúng ta cần xây dựng một bộ hồ sơ (profile) cho mỗi item. Profile này được biểu diễn dưới dạng toán học là một feature vector. Trong những trường hợp đơn giản, feature vector được trực tiếp trích xuất từ item.

Có rất nhiều yếu tố khác của một bài hát có thể được sử dụng. Ngoại trừ Thể loại khó định nghĩa, các yếu tố khác đều được xác định rõ ràng. Đặt các feature vector cho mỗi bài hát là x_1, x_2, x_3, x_4, x_5 . Giả sử các feature vector (ở dạng hàng) cho mỗi bài hát được cho trong hình dưới đây:

	A	B	C	D	E	F	item's feature vectors
Mưa nửa đêm	5	5	0	0	1	?	$x_1 = [0.99, 0.02]$
Cỏ úa	5	?	?	0	?	?	$x_2 = [0.91, 0.11]$
Vùng lá me bay	?	4	1	?	?	1	$x_3 = [0.95, 0.05]$
Con cò bé bé	1	1	4	4	4	?	$x_4 = [0.01, 0.99]$
Em yêu trường em	1	0	5	?	?	?	$x_5 = [0.03, 0.98]$
User's models	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	← need to optimize

Hình 1. 3: Giả sử feature vector cho mỗi item được cho trong cột cuối cùng. Với mỗi user, chúng ta cần tìm một mô hình.

Bài toán đi tìm mô hình Θ_i cho mỗi user có thể được coi là một bài toán Regression trong trường hợp ratings là một dải giá trị, hoặc bài toán Classification trong trường hợp ratings là một vài trường hợp cụ thể, như like/dislike chẳng hạn. Dữ liệu training để xây dựng mỗi mô hình Θ_i là các cặp (item profile, ratings) tương ứng với các items mà user đó đã rated. Việc điền các giá trị còn thiếu trong ma trận Utility chính là việc dự đoán đầu ra cho các unrated items khi áp dụng mô hình Θ_i lên chúng. Việc lựa chọn mô hình Regression/Classification nào tùy thuộc vào ứng dụng.

1.2.2.2 Xây dựng hàm mất mát

Giả sử rằng số users là N , số items là M , utility maxtrix được mô tả bởi ma trận Y . Thành phần ở hàng thứ m , cột thứ n của Y là mức độ quan tâm (ở đây là số sao đã rate) của user thứ n lên sản phẩm thứ m mà hệ thống đã thu thập được. Ma trận Y bị khuyết rất nhiều thành phần tương ứng với các giá trị mà hệ thống cần dự đoán. Thêm nữa, gọi R là ma trận rated or not thể hiện việc một user đã rated một item hay chưa. Cụ thể, r_{ij} bằng 1 nếu item thứ i đã được rated bởi user thứ j , bằng 0 trong trường hợp ngược lại.

Mô hình tuyến tính: Giả sử rằng ta có thể tìm được một mô hình cho mỗi user, minh họa bởi vector cột hệ số w_i và bias b_n sao cho mức độ quan tâm của một user tới một item có thể tính được bằng một hàm tuyến tính:

$$y_{mn} = x_m w_n + b_n \quad (1.4)$$

Chú ý rằng x_m là một vector hàng, w_n là một vector cột. Xét một user thứ n bất kỳ, nếu ta coi training set là tập hợp các thành phần đã được điền của y_n , ta có thể xây dựng hàm mất mát tương tự như Ridge Regression [21] như sau:

$$L_n = \frac{1}{2} \sum_{m:r_{mn}=1} (x_m w_n + b_n - y_{mn})^2 + \frac{\lambda}{2} \|w_n\|_2^2 \quad (1.5)$$

Trong đó, thành phần thứ hai là regularization term và λ là một tham số dương. Chú ý rằng regularization thường không được áp dụng lên *bias* b_n . Trong thực hành, trung bình cộng của *lỗi* thường được dùng, và *mất mát* L_n được viết lại thành:

$$L_n = \frac{1}{2s_n} \sum_{m:r_{mn}=1} (x_m w_n + b_n - y_{mn})^2 + \frac{\alpha}{2s_n} \|w_n\|_2^2 \quad (1.6)$$

Trong đó s_n là số lượng các items mà user thứ n đã rated. Nói cách khác:

$$s_n = \sum_{m=1}^M r_{mn} \quad (1.7)$$

Là tổng các phần tử trên cột thứ n của ma trận rated or not R . Vì biểu thức loss function chỉ phụ thuộc vào các items đã được rated, ta có thể rút gọn nó bằng cách đặt $\widehat{y}_n$ là sub vector của y được xây dựng bằng cách trích các thành phần khác dấu ? ở cột thứ n , tức đã được *rated* bởi user thứ n trong Utility Matrix Y . Đồng thời, đặt $\widehat{X}_n$ là sub matrix của ma trận feature X , được tạo bằng cách trích các hàng tương ứng với các items đã được *rated* bởi user thứ n . Khi đó, biểu thức hàm mất mát của mô hình cho user thứ n được viết gọn thành:

$$L_n = \frac{1}{2s_n} \left\| \widehat{X}_n w_n + b_n e_n - \widehat{y}_n \right\|_2^2 + \frac{\lambda}{2s_n} \|w_n\|_2^2 \quad (1.8)$$

Trong đó e_n là vector chứa cột s_n phần tử 1. Đây chính xác là hàm mất mát của Ridge Regression. Cặp nghiệm w_n, b_n có thể được tìm qua Stochastic Gradient Descent (SGD), hoặc Mini-batch Gradient Descent [21].

1.3 Giới thiệu mô hình chuyển đổi văn bản thành giọng nói

1.3.1. Tổng quan về công nghệ chuyển đổi văn bản thành giọng nói

Mô hình chuyển đổi văn bản thành giọng nói (TTS) là công nghệ cho phép chuyển đổi nội dung văn bản thành âm thanh giọng nói nhân tạo. Nói một cách đơn giản, TTS giúp máy tính “đọc to” văn bản bằng tiếng nói giống con người [20], [21]. Trong bối cảnh hệ thống học ngoại ngữ tích hợp AI, TTS đóng vai trò đặc biệt quan trọng trong việc hỗ trợ kỹ năng nghe và phát âm. Nhờ TTS, mọi đoạn văn, từ vựng hoặc câu hỏi thoại trong tài liệu học tập đều có thể được phát âm chuẩn, giúp người học nghe và bắt chước một cách dễ dàng, kể cả khi không có giáo viên hay người bản ngữ kèm cặp.

1.3.2 Giới thiệu về mô hình Piper

Piper TTS là một hệ thống TTS mã nguồn mở thuộc hệ sinh thái Rhasspy, được thiết kế hướng tới: Chạy cục bộ trên máy cá nhân hoặc thiết bị nhúng (Raspberry Pi, mini-PC). Yêu cầu tài nguyên thấp, tối ưu cho tốc độ suy luận. Dễ triển khai thông qua mô hình đã được chuyển sang định dạng ONNX (Open Neural Network Exchange), tận dụng ONNX Runtime.

Piper cung cấp nhiều mô hình đã được huấn luyện sẵn cho nhiều ngôn ngữ khác nhau. Người dùng có thể chọn giọng nam/nữ, tốc độ đọc, hoặc tự huấn luyện thêm mô hình mới nếu có dữ liệu giọng nói đủ lớn.

Piper triển khai theo kiến trúc TTS end-to-end dựa trên các mô hình kiểu VITS (Variational Inference with adversarial learning for end-to-end TTS) [51]. Quy trình xử lý có thể mô tả qua các bước chính:

- Tiền xử lý văn bản (Text front-end): Chuẩn hóa văn bản (xử lý số, ký hiệu, viết tắt...). Chuyển chuỗi ký tự thành chuỗi phoneme (G2P – grapheme-to-phoneme) nếu mô hình hoạt động ở mức phoneme [21]. Mã hóa thành chuỗi ID (token) dùng làm input cho mô hình.
- Mô hình TTS lõi (VITS-based acoustic model): Nhận chuỗi token/phoneme, ánh xạ sang không gian ẩn, đồng thời ước lượng phân bố biến ẩn đại diện cho đặc trưng giọng nói. Sử dụng cơ chế variational inference để học phân bố latent, kết hợp với mạng đối kháng (adversarial) nhằm tối ưu chất lượng tín hiệu âm thanh đầu ra. Thay vì tách riêng acoustic model và vocoder, VITS tích hợp hai phần trong một kiến trúc end-to-end, giúp giảm độ trễ và đơn giản pipeline.

Suy luận với ONNX Runtime: Mô hình sau khi huấn luyện được chuyển sang ONNX, nhờ đó có thể load và chạy trên nhiều nền tảng (CPU/GPU) với hiệu năng cao. Piper cung cấp CLI/API để đưa text → WAV/OGG chỉ bằng một lệnh, rất thuận lợi khi tích hợp vào backend [42].

Bảng 1. 2: Ưu điểm, hạn chế của piperTTS

Ưu điểm	Hạn chế
Hoạt động offline, không gửi dữ liệu lên cloud đảm bảo quyền riêng tư. Độ trễ thấp Mã nguồn mở, dễ tùy biến và đóng gói trong Docker để triển khai cùng hệ thống.	Số lượng giọng và ngôn ngữ ít đa dạng hơn so với các dịch vụ cloud lớn. Chất lượng/độ tự nhiên của một số voice có thể thấp hơn các voice thương mại cao cấp

Piper TTS có thể làm mô hình TTS chính cho các bài luyện nghe, roleplay, phát âm. Được sử dụng cho các tính năng cần phản hồi nhanh, ổn định, không phụ thuộc Internet.

1.3.3. Dịch vụ Google Text-to-Speech

1.3.3.1. Giới thiệu gTTS

Google Text-to-Speech là dịch vụ TTS do Google cung cấp, triển khai dưới dạng API cloud. Thư viện gTTS (Python) là một wrapper đơn giản giúp gửi text lên API này và nhận về file âm thanh (thường là MP3) [43]. Hỗ trợ nhiều ngôn ngữ và biến thể vùng miền. Cung cấp nhiều giọng đọc khác nhau. Sử dụng các mô hình neural TTS hiện đại (WaveNet, Neural2...) ở phía server, nên chất lượng giọng rất tự nhiên, trơn tru. gTTS được sử dụng như một lựa chọn bổ sung (secondary TTS engine) bên cạnh Piper. Dùng để tạo một số nội dung audio pre-generated. Là phương án dự phòng khi cần thêm giọng hoặc ngôn ngữ mà Piper chưa hỗ trợ tốt.

1.3.3.2. Cấu trúc hệ thống và cách hoạt động

Pipeline điển hình của gTTS/Google TTS trong hệ thống:

- Ứng dụng khách (client) / backend: Chuẩn bị request: text, ngôn ngữ, lựa chọn giọng, tốc độ đọc... Gửi lên Google TTS API qua HTTP.
- Tầng xử lý văn bản (text processing) phía Google: Chuẩn hóa văn bản, áp dụng các quy tắc SSML (Speech Synthesis Markup Language) nếu có (điều chỉnh tốc độ, cao độ, nhấn giọng) [36]. Mã hóa text thành chuỗi token/phoneme.

- Mô hình TTS phía server: Các mô hình neural TTS (WaveNet/Neural2) tạo mel-spectrogram và/hoặc sinh waveform. Hệ thống tối ưu cho chất lượng giọng và độ tự nhiên ở mức sản phẩm thương mại.
- Trả về kết quả: API trả về file âm thanh. Backend lưu file này hoặc stream trực tiếp cho client.

Khi tích hợp gTTS vào hệ thống Backend có thể cung cấp một endpoint nội bộ gọi gTTS để tạo file âm thanh và cache theo (text, lang, voice) tương tự Piper. Với các nội dung tĩnh (bài đọc, ví dụ, hội thoại mẫu), có thể pre-generate audio và lưu vào CSDL/bucket, tránh gọi API nhiều lần.

1.3.4. Vai trò của TTS trong học ngoại ngữ tích hợp AI

Đối với người học ngoại ngữ, kỹ năng nghe và phát âm thường là thử thách lớn do thiếu môi trường ngôn ngữ bản xứ. TTS trở thành “cầu nối” giúp người học tiếp cận âm thanh chuẩn một cách chủ động và linh hoạt:

Hỗ trợ luyện nghe: Mọi đoạn hội thoại, bài khóa, câu ví dụ trong tài liệu đều có thể được phát thành âm thanh sống động.

Hỗ trợ luyện phát âm (kết hợp với STT): Hệ thống phát câu mẫu bằng TTS gửi tới người học nói để người học bắt chước sau đó hệ thống ghi âm và phân tích. Nhờ có giọng mẫu chuẩn, người học biết “đích” phát âm cần đạt được, kết hợp các chỉ số đánh giá phát âm tự động giúp họ nhận biết rõ vị trí sai và điều chỉnh dần. Nhiều nghiên cứu cho thấy việc nghe TTS và đọc theo giúp cải thiện đáng kể độ chính xác phát âm, đặc biệt ở mức âm tiết/từ đơn [6], [28].

Nâng cao trải nghiệm học tập: Việc bài học luôn kèm audio làm nội dung sinh động hơn, giúp tăng động lực học. Tất nhiên, TTS vẫn có hạn chế: giọng tổng hợp dù tự nhiên vẫn có thể thiếu một số sắc thái cảm xúc hoặc ngữ điệu tinh tế như người bản xứ trong hội thoại thực. Vì vậy, người học không nên phụ thuộc hoàn toàn vào TTS mà nên kết hợp thêm việc nghe người thật nói (qua video, phim, podcast). Tuy vậy, xét trên tổng thể, TTS vẫn là một thành phần cực kỳ quan trọng trong hệ thống học ngoại ngữ tích hợp AI, giúp thu hẹp khoảng cách về môi trường luyện nghe–nói cho người học tự học, và sẽ càng đóng vai trò lớn hơn khi các mô hình TTS đa ngữ, giàu cảm xúc tiếp tục phát triển [5], [6], [19].

1.4 Giới thiệu mô hình chuyển đổi giọng nói thành văn bản

1.4.1 Giới thiệu về mô hình Whisper

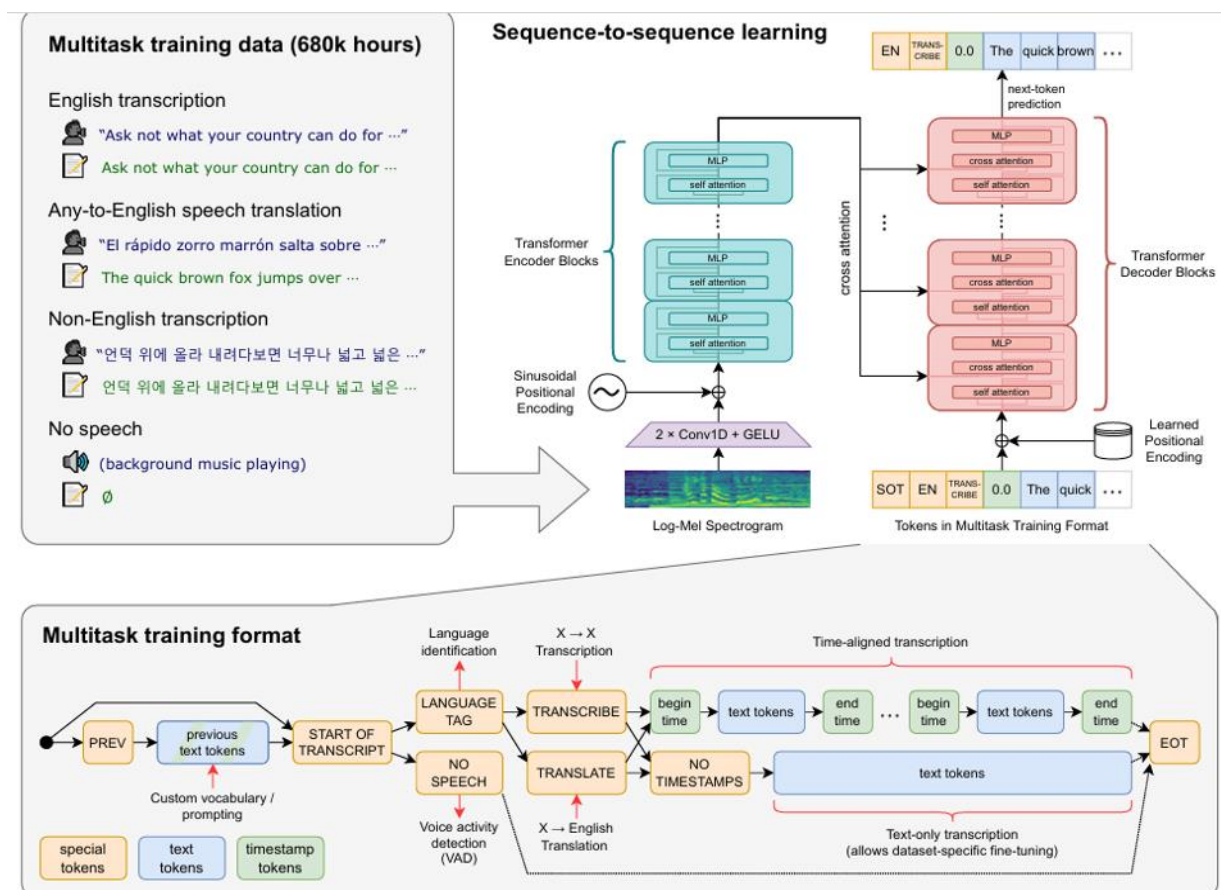
Chuyển đổi giọng nói thành văn bản (STT), hay nhận dạng giọng nói tự động (Automatic Speech Recognition - ASR), là công nghệ cho phép máy tính chuyển đổi tiếng nói của con người thành văn bản. Hệ thống STT tiếp nhận tín hiệu âm thanh và tạo ra đầu ra là chuỗi ký tự hoặc từ tương ứng. Mục tiêu cuối cùng của STT là đạt độ chính xác tương đương con người trong việc hiểu lời nói, mở đường cho các ứng dụng như trợ lý ảo, điều khiển bằng giọng nói, nhập liệu bằng giọng nói... [2], [3], [24].

Whisper mô hình nhận dạng tiếng nói end-to-end và là một mô hình STT nổi bật nhờ huấn luyện trên 680 nghìn giờ dữ liệu đa ngôn ngữ và đa nhiệm, hướng đến tính robust (bền vững) trên nhiều điều kiện dữ liệu khác nhau dựa trên kiến trúc Transformer encoder-decoder cổ điển [26], [27]. Whisper nổi bật ở hỗ trợ đa ngôn ngữ (~99 ngôn ngữ) và đa nhiệm (chuyển giọng nói thành văn bản, dịch sang tiếng Anh, nhận dạng ngôn ngữ, dấu mốc thời gian, phát hiện không có lời thoại) trong một hệ thống duy nhất.

1.4.2 Phân tích về mô hình Whisper

Đầu vào (Input): Whisper tiếp nhận âm thanh dưới dạng các đoạn dài tối đa 30 giây. Âm thanh thô (waveform) được xử lý sơ bộ thành đặc trưng tần số: cụ thể, tín hiệu được resample về 16 kHz, sau đó tính spectrogram Mel với cửa sổ 25 ms, bước 10 ms và 80 kênh Mel-frequency (80 chiều). Kết quả là một ma trận đặc trưng kích thước $80 \times N$ (với N là số frame thời gian) biểu diễn năng lượng theo thời gian và tần số. Tiếp theo, Whisper chuẩn hóa biên độ spectrogram về khoảng $[-1, 1]$ (để trung bình ~ 0) nhằm tránh sai lệch do thang âm lượng [21], [28]. Những đặc trưng log-Mel này là đầu vào chính thức cho bộ mã hóa (encoder) của mô hình. Mỗi đoạn âm thanh dài 30 giây tương ứng khoảng $N \approx 3000$ frame (30s/10ms), qua khối CNN giảm còn $\sim 1/2$ (xem phần Encoder). Như vậy, chuỗi đầu vào cho Transformer encoder có độ dài khoảng vài nghìn bước (phụ thuộc thời lượng thực tế, tối đa ~ 3000 bước cho 30s).

Đầu ra (Output): Đầu ra của Whisper là chuỗi token văn bản đại diện cho nội dung lời nói (transcript hoặc bản dịch). Khác với một số mô hình chỉ xuất ra ký tự rời rạc, Whisper xuất ra token từ vựng nhờ sử dụng BPE tokenizer nghĩa là mỗi token có thể là một phần của từ hoặc ký tự đặc biệt. Đặc biệt, định dạng chuỗi đầu ra của Whisper được thiết kế để tích hợp nhiều nhiệm vụ. Trước khi các từ của câu thoại được sinh, mô hình sẽ sinh ra một loạt các token đặc biệt (ngôn ngữ, nhiệm vụ, thời gian) để thiết lập ngữ cảnh nhiệm vụ.



Hình 1. 4: Mô hình Biến đổi chuỗi-trình tự (seq-2-seq) được huấn luyện trên nhiều tác vụ xử lý giọng nói khác nhau. [28]

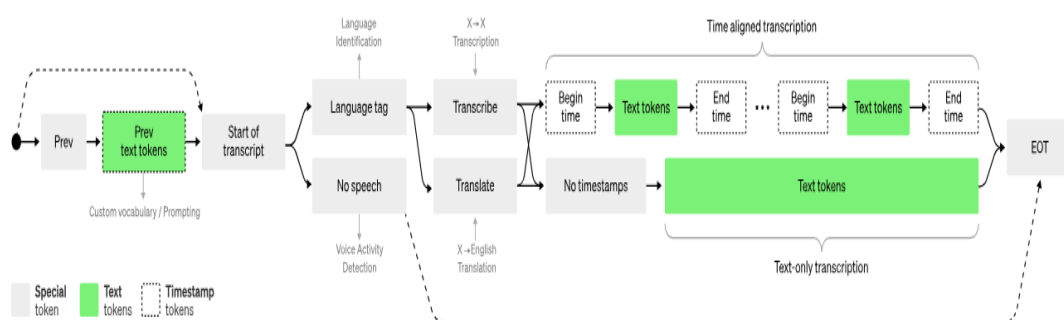


Hình 1. 5: Đầu vào đầu ra dữ liệu đào tạo ASR

Whisper sử dụng tokenizer ở cả hai loại đầu vào với văn bản (để sinh đầu ra) và audio (cho encoder).

Mã hóa văn bản (Text tokenizer): Whisper có một vocab chung cho tất cả 99 ngôn ngữ, bao gồm cả các token đặc biệt: token nhiệm vụ transcribe, translate, token nospeech (báo hiệu không có tiếng nói) và tập các token timestamp_x (x đại diện cho các mốc thời gian lượng tử 0.02s). Tất cả những token này được xem như các từ trong từ điển của decoder. Cách tiếp cận này cho phép decoder của Whisper sinh đầu ra đa dạng mà không cần thay đổi cấu trúc mô hình khi thêm nhiệm vụ.

Mã hóa audio đầu vào âm thanh được mã hóa thông qua mạng neural. Cụ thể, bộ mã hóa (encoder) của Whisper sẽ xử lý chuỗi frame spectrogram để trích xuất embedding liên tục. Một bước tiền xử lý quan trọng là CNN stem – mô hình có một tầng CNN gồm 2 lớp tích chập (kernel size 3) áp dụng lên spectrogram, với lớp thứ hai stride=2 để giảm độ dài chuỗi một nửa. Mục đích là giảm bớt độ dài chuỗi thời gian (ví dụ từ ~3000 frame/30s xuống ~1500 bước) giúp Transformer encoder hoạt động hiệu quả hơn. Sau khối CNN, Whisper thêm position encoding dạng sin vào chuỗi đặc trưng (theo trục thời gian), rồi đưa qua các khối Transformer encoder. Như vậy, ta có thể coi mỗi frame âm thanh sau CNN tương ứng với một “token âm” chứa thông tin đặc trưng âm thanh ở thời điểm đó, và vị trí tương ứng của nó được mã hóa qua sinusoidal positional encoding.



Hình 1. 6: Dòng suy luận & định dạng token của Whisper

WER (Word Error Rate) là thước đo chính để đánh giá độ chính xác STT, tính bằng tỷ lệ % số từ bị nhận dạng sai trên tổng từ. CER (Character Error Rate) tương tự nhưng tính theo ký tự, phù hợp cho ngôn ngữ dán chữ hoặc đánh giá chi tiết hơn. Whisper tổng quát hóa tốt hơn khi gặp điều kiện mới (tiếng ồn, giọng địa phương, hội thoại nhiều người, v.v). Đối với ngôn ngữ khác tiếng Anh, Whisper cung cấp năng lực zero-shot đáng kể. Thử nghiệm trên bộ FLEURS [30] (102 ngôn ngữ) cho thấy WER giảm dần khi tăng dữ liệu huấn luyện của ngôn ngữ đó – cụ thể, mỗi khi lượng dữ liệu cho ngôn ngữ tăng 16 lần, WER giảm khoảng một nửa. Tuy nhiên, hiệu quả giữa các tiếng không đồng đều: những ngôn ngữ có chữ viết đặc biệt thường có WER cao hơn dự đoán do tokenizer BPE kém hiệu quả hoặc dữ liệu huấn luyện hạn chế.

Bảng 1. 3: Ưu điểm & nhược điểm của mô hình Whisper

Ưu điểm	Nhược điểm
Độ chính xác tốt trên tiếng Anh sạch ($\approx 2.5\%$ WER test-clean), dùng ngay không cần fine-tune	Mô hình lớn dẫn tới độ trễ cao, đặc biệt trên CPU/mobile
Robust hơn nhiều mô hình truyền thống khi gặp ồn, accent, hội thoại tự nhiên	Không streaming gốc (xử lý theo block $\sim 30s$) làm cho khó đáp ứng real-time
Đa ngôn ngữ (~ 99) & đa nhiệm (transcribe/translate/timestamps) trong một mô hình	Xử lý audio dài cần cắt ghép với heuristic có thể lặp/hallucination ở ca khó
Tự thêm dấu câu/chữ hoa và timestamp để transcript dễ đọc	Không SOTA trên mọi benchmark/ngôn ngữ, một số tiếng hiếm còn hạn chế
Mã nguồn mở, nhiều kích cỡ model, hệ sinh thái phong phú (whisper.cpp, faster-whisper)	Cần GPU để đạt tốc độ và chất lượng cao on-device khó với bản lớn
Có thể chạy offline	Tokenizer/dữ liệu chưa tối ưu cho một số ngôn ngữ $\rightarrow$ CER/WER cao hơn

1.5 Tổng quan về xử lý ngôn ngữ tự nhiên

1.5.1 Giới thiệu về xử lý ngôn ngữ tự nhiên

Xử lý ngôn ngữ tự nhiên là một nhánh của trí tuệ nhân tạo tập trung vào việc giúp máy tính hiểu, phân tích và sinh ngôn ngữ của con người. Trong bối cảnh giáo dục ngoại ngữ, NLP giữ vai trò trung tâm khi biến dữ liệu giao tiếp tự nhiên (văn bản hoặc lời nói) thành tín hiệu khả dụng để chấm điểm, gợi ý luyện tập và đối thoại tương tác theo thời gian thực. Nhờ vậy, hệ thống không chỉ phát hiện và giải thích lỗi ngữ pháp, mà còn cá nhân hoá phản hồi theo trình độ và mục tiêu học tập của từng người học.

Về mặt chức năng, NLP trước hết hỗ trợ phân tích cú pháp và ngữ pháp: tách từ (Tokenization), gán thẻ từ loại (Part-of-Speech), nhận diện phụ thuộc cú pháp (Dependency Parsing) để phát hiện các lỗi hòa hợp chủ-vị, thì, mạo từ hay trật tự từ và gợi ý cách sửa có ngữ cảnh. Trên lớp này là bài toán phát hiện và sửa lỗi ngữ pháp (GEC). Đồng thời, mô-đun nhận diện ý định giúp xác định mục đích của câu nói hỏi đường, gợi món, giới thiệu bản thân... để kích hoạt kịch bản hội thoại phù hợp, còn mô-đun hiểu ngữ nghĩa và nhận diện thực thể trích rút các thông tin như ngày giờ, địa điểm,

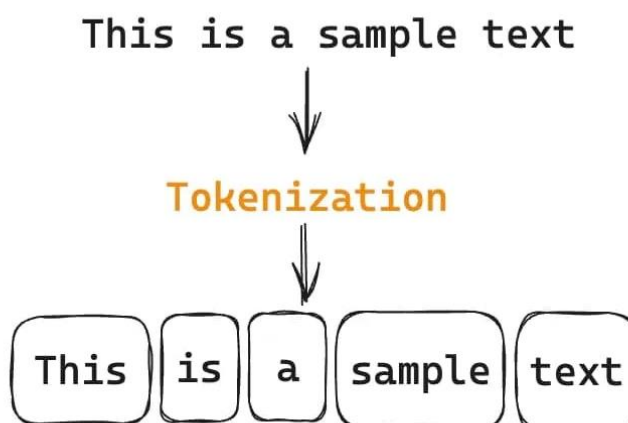
tên riêng nhằm tránh hiểu sai ngữ cảnh. Ở chiều sinh, NLG tạo ra câu hỏi luyện tập, đoạn hội thoại hoặc phản hồi giải thích chứa quy tắc và ví dụ đối chiếu.

1.5.2 Các tác vụ chính trong xử lý ngôn ngữ tự nhiên

1.5.2.1 Tách từ

Chia đoạn văn bản thành các đơn vị nhỏ gọi là *token* (thường là từ hoặc cụm từ).

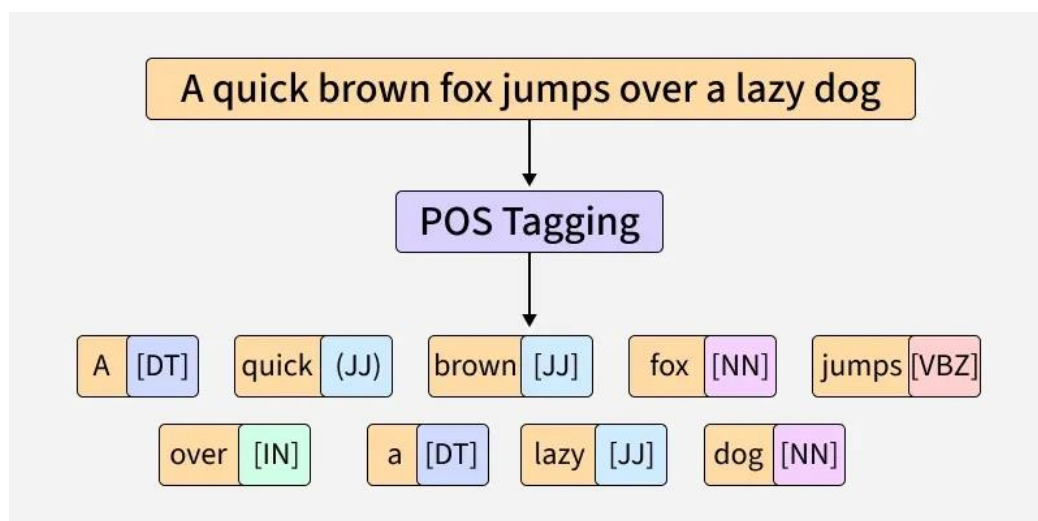
Tokenization là một bước trong nền tảng trong quy trình NLP định hình toàn bộ quy trình làm việc. Nó liên quan đến việc chia chuỗi hoặc văn bản thành một danh sách các đơn vị nhỏ hơn được gọi là mã thông báo (tokens). Việc sử dụng Tokenizer để phân đoạn dữ liệu phi cấu trúc và văn bản ngôn ngữ tự nhiên thành các phần riêng biệt, coi chúng là các yếu tố khác nhau. Tokens là từ hoặc từ phụ trong ngữ cảnh xử lý ngôn ngữ tự nhiên [31].



Hình 1. 7: Minh họa cách tokenization

1.5.1.2. Gắn thẻ từ loại

Gắn thẻ từ loại là một trong những nhiệm vụ cơ bản của NLP trong đó mỗi từ trong đó mỗi từ sẽ được gán một danh mục ngữ pháp như danh từ, động từ, tính từ hoặc trạng từ. Quá trình này giúp máy móc hiểu cấu trúc và ý nghĩa của câu bằng cách xác định vai trò của các từ và mối quan hệ của chúng [32]. Nó đóng một vai trò quan trọng trong các ứng dụng NLP khác nhau bao gồm dịch máy, phân tích cảm xúc và truy xuất thông tin, bằng cách thu hẹp khoảng cách giữa ngôn ngữ con người và hiểu biết của máy.



Hình 1. 8: Minh họa quá trình mỗi từ được gán một thẻ dựa trên vai trò ngữ pháp của nó trong câu [32]

Quy trình thực hiện Để thực hiện việc gán thẻ từ loại một cách chính xác, hệ thống thường tuân theo một quy trình xử lý tuần tự bao gồm các bước sau:

- Tokenization [31] : Văn bản đầu vào được chia thành các mã thông báo riêng lẻ (từ hoặc từ phụ), bước này là cần thiết để phân tích sâu hơn.
- Tiền xử lý: Văn bản được làm sạch như chuyển sang chữ thường và loại bỏ các ký tự đặc biệt, để nâng cao độ chính xác.
- Tải mô hình ngôn ngữ: Các công cụ như NLTK hoặc SpaCy sử dụng các mô hình ngôn ngữ được đào tạo trước để hiểu các quy tắc ngữ pháp của ngôn ngữ, các mô hình này đã được đào tạo trên các tập dữ liệu lớn.
- Phân tích ngôn ngữ: Cấu trúc của câu được phân tích để hiểu vai trò của từng từ trong ngữ cảnh.
- Gán thẻ POS: Mỗi từ được gán một nhãn thành phần lời nói dựa trên vai trò của nó trong câu và ngữ cảnh do các từ xung quanh cung cấp.
- Đánh giá: Kết quả được kiểm tra độ chính xác. Nếu có bất kỳ sai sót hoặc phân loại sai nào, chúng sẽ được sửa chữa.

1.5.1.3 Phân tích cú pháp phụ thuộc [33]

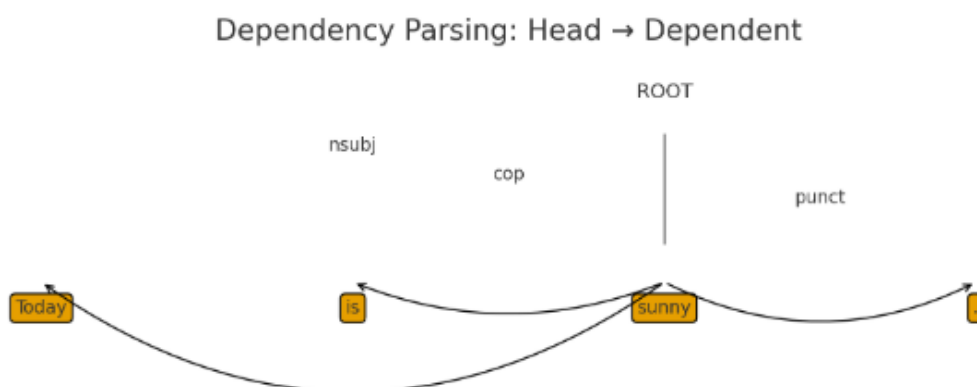
Phân tích cú pháp phụ thuộc (Dependency Parsing) là kỹ thuật xác định cấu trúc ngữ pháp của câu thông qua việc thiết lập các mối quan hệ phụ thuộc hai ngôi giữa các

từ. Khác với phân tích cú pháp thành phần (Constituency Parsing) tập trung vào các cụm từ, Dependency Parsing tập trung vào mối liên kết trực tiếp giữa từ với từ [32].

Trong cấu trúc này, mỗi quan hệ được xác định giữa hai thành phần:

- Từ chủ dẫn (Head): Từ đóng vai trò trung tâm, quy định tính chất ngữ pháp của cụm.
- Từ phụ thuộc (Dependent): Từ bổ nghĩa hoặc thay đổi ý nghĩa cho từ chủ dẫn.

Mỗi quan hệ này thường được biểu diễn dưới dạng các cạnh có hướng (từ Head đến Dependent) kèm theo nhãn mô tả loại quan hệ ngữ pháp (ví dụ: chủ ngữ (nsubj), tân ngữ (obj), bổ ngữ (amod)). Kết quả đầu ra là một cây hoặc đồ thị phụ thuộc, giúp máy tính hiểu sâu hơn về ngữ nghĩa và cấu trúc tuyến tính của câu.



Hình 1. 9: Minh họa dependency parsing (head->dependent) [33]

Vai trò và Ứng dụng Kỹ thuật này đóng vai trò nền tảng trong nhiều bài toán NLP hiện đại:

- Hiểu ngôn ngữ tự nhiên (NLU): Giúp máy xác định ai làm gì, tác động lên đối tượng nào.
- Dịch máy (Machine Translation): Cải thiện chất lượng dịch bằng cách giữ nguyên cấu trúc ngữ nghĩa giữa các ngôn ngữ có trật tự từ khác nhau.
- Hệ thống Hỏi-Đáp (QA) và Tóm tắt văn bản: Hỗ trợ trích xuất thông tin chính xác dựa trên vai trò của các thực thể trong câu.

Thách thức Mặc dù hiệu quả, Dependency Parsing đối mặt với nhiều trở ngại:

- Phụ thuộc khoảng cách xa (Long-distance dependencies): Khi từ chủ dẫn và từ phụ thuộc nằm cách xa nhau trong câu, việc thiết lập liên kết trở nên phức tạp.
- Đa nghĩa cú pháp (Syntactic Ambiguity): Một câu có thể có nhiều cách phân tích hợp lý về mặt ngữ pháp nhưng chỉ có một ý nghĩa đúng trong ngữ cảnh.
- Cấu trúc đặc biệt: Các thành ngữ hoặc cấu trúc câu đảo ngữ gây khó khăn cho việc gán nhãn quan hệ chính xác.

1.5.1.4. Nhận dạng thực thể có tên [32]

Nhận dạng thực thể được đặt tên (NER) trong NLP tập trung vào việc xác định và phân loại thông tin quan trọng được gọi là thực thể trong văn bản. Các thực thể này có thể là tên của người, địa điểm, tổ chức, ngày tháng, v.v. Nó giúp chuyển đổi văn bản phi cấu trúc thành thông tin có cấu trúc, giúp thực hiện các tác vụ như tóm tắt văn bản, tạo biểu đồ tri thức và trả lời câu hỏi. NER giúp phát hiện thông tin cụ thể và sắp xếp nó thành các danh mục được xác định trước. Nó đóng một vai trò quan trọng trong việc nâng cao các tác vụ NLP khác như Part-of-Speech và parsing [32].

1.5.1.5 Phát hiện và sửa lỗi ngữ pháp

Sửa lỗi ngữ pháp (Grammatical Error Correction - GEC) là nhiệm vụ sửa các loại lỗi khác nhau trong văn bản như lỗi chính tả, dấu câu, ngữ pháp và lựa chọn từ. GEC thường được xây dựng như một nhiệm vụ sửa câu. Một hệ thống GEC lấy một câu có khả năng sai làm đầu vào và dự kiến sẽ chuyển đổi nó thành phiên bản sửa chữa của nó [36].

Input (Erroneous)	Output (Corrected)
She see Tom is caught by policeman in park at last night.	She saw Tom caught by a policeman in the park last night.

Hình 1. 10: Ví dụ về input & output của GEC

1.5.3 Các Thuật Toán và Chỉ Số trong NLP và Đánh Giá Phát Âm

Cosine Similarity (Độ tương đồng cosin)

Cosine similarity đo mức độ tương tự giữa hai vector bằng cosin của góc giữa chúng. Công thức tính như sau:

$$\text{Cosine similarity} = \frac{(A \cdot B)}{(\|A\| \times \|B\|)} \quad (1.9)$$

Trong đó :

- $A \cdot B$ là tích vô hướng hai vector
- $\|A\|$ và $\|B\|$ là độ dài (chuẩn Euclid) của từng vector.

Giá trị cosin nằm trong khoảng $[-1, 1]$, nếu các vector biểu diễn cùng hướng thì $\cosin = 1$ (cực kỳ giống nhau), vuông góc thì $= 0$, và ngược hướng thì $= -1$

Trong xử lý ngôn ngữ tự nhiên (NLP), Cosine similarity được áp dụng rộng rãi để so sánh độ giống nhau về ngữ nghĩa của văn bản. Văn bản thường được biểu diễn dưới dạng vector (ví dụ: vector từ vựng theo mô hình bag-of-words, TF-IDF hoặc embedding). Khi đó, độ tương đồng cosin giữa các vector văn bản phản ánh mức độ hai đoạn văn có nội dung chủ đề giống nhau, bất chấp độ dài khác nhau của chúng [26]. Nhờ tính chất chuẩn hóa theo độ dài, cosine similarity là thước đo phổ biến để so sánh tài liệu, truy vấn tìm kiếm với văn bản, phát hiện ý nghĩa tương đồng giữa từ/ngữ trong không gian nhúng, v.v. với chi phí tính toán thấp.

Độ tương đồng từ vựng (Lexical Similarity): Độ tương đồng từ vựng đánh giá mức độ trùng lặp hay khác biệt về các token (từ hoặc ký tự) giữa hai chuỗi văn bản. Khác với cosin (dựa trên không gian vector ngữ nghĩa), các phép đo lexical similarity hoạt động trực tiếp trên chuỗi ký tự/từ. Dưới đây là một số chỉ số và thuật toán phổ biến:

Jaccard similarity (Chỉ số Jaccard)[40]: Đây là thước đo dựa trên tập hợp từ chung. Chỉ số Jaccard giữa hai tập từ A và B được định nghĩa là tỷ lệ giữa kích thước giao và hợp của chúng:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad (1.10)$$

Tức là số lượng token chung chia cho tổng số token duy nhất xuất hiện trong cả hai đoạn. Giá trị Jaccard nằm trong $[0, 1]$: bằng 0 nếu hai tập không có token nào chung, và bằng 1 nếu hai tập trùng nhau hoàn toàn [27]. Trong NLP, chỉ số Jaccard được dùng để so sánh độ giống nhau của tài liệu dựa trên từ chung (ví dụ phát hiện văn bản trùng lặp, đo độ tương tự tiêu đề, v.v.).

Độ trùng lặp token và F1-score trên từ: Một cách khác để đo độ tương đồng trên tập từ là sử dụng Precision, Recall và F1-score – thường dùng trong đánh giá mô hình. Ta coi một văn bản làm danh sách dự đoán và văn bản kia là danh sách tham chiếu. Khi đó:

Precision (Độ chính xác - P): Chỉ số này đo lường tỷ lệ các từ trong văn bản dự đoán thực sự xuất hiện trong văn bản tham chiếu. Nó trả lời cho câu hỏi: "Bao nhiêu phần nội dung máy tạo ra là đúng?".

$$P = \frac{|T_{\text{pred}} \cap T_{\text{ref}}|}{|T_{\text{pred}}|} \quad (1.11)$$

Recall (Độ phủ - R): Chỉ số này đo lường tỷ lệ các từ trong văn bản tham chiếu đã được văn bản dự đoán bao phủ. Nó trả lời cho câu hỏi: "Văn bản dự đoán đã khôi phục được bao nhiêu phần nội dung của văn bản gốc?".

$$R = \frac{|T_{\text{pred}} \cap T_{\text{ref}}|}{|T_{\text{ref}}|} \quad (1.12)$$

F1-score: Vì Precision và Recall thường có sự đánh đổi (trade-off) – tăng cái này có thể làm giảm cái kia – nên chỉ số F1-score được sử dụng như một thước đo trung bình điều hòa (harmonic mean) giữa P và R, giúp đánh giá tổng quan độ tương đồng.

$$F1 = 2 \times \frac{P \times R}{P + R} \quad (1.13)$$

Trong đó:

- T_{pred} là tập hợp các token (từ) xuất hiện trong văn bản dự đoán.
- T_{ref} là tập hợp các token (từ) xuất hiện trong văn bản tham chiếu.
- $|T|$ biểu thị tổng số lượng token trong tập hợp T.
- $|T_{\text{pred}} \cap T_{\text{ref}}|$ biểu thị số lượng token xuất hiện đồng thời trong cả hai văn bản (số token chung).

1.5.4 Ứng dụng của NLP trong giáo dục

1.5.4.1 Nhận dạng giọng nói và phát âm

Hai công nghệ nền tảng hỗ trợ trực tiếp kỹ năng nghe – nói là STT và TTS. Nhận diện giọng nói (STT/ASR) chuyển lời nói của người học thành văn bản để kiểm tra phát âm, ngữ điệu, tốc độ và ngắt nghỉ. Các kiến trúc hiện đại như wav2vec [24] hay Conformer [25] cho phép nhận dạng streaming, giảm độ trễ trong hội thoại. Ở chiều ngược lại, tổng hợp giọng nói (TTS) biến văn bản thành âm thanh tự nhiên, có thể điều chỉnh tốc độ, cường độ, cao độ và sử dụng ngôn ngữ đánh dấu tổng hợp giọng nói (SSML) là ngôn ngữ đánh dấu dựa trên XML mà bạn có thể sử dụng để tinh chỉnh các thuộc tính đầu ra văn bản thành giọng nói như cao độ, phát âm, tốc độ nói, âm lượng, v.v. Nó cung cấp cho bạn nhiều quyền kiểm soát và linh hoạt hơn so với nhập văn bản

thuần túy [36] để nhấn mạnh/hạ giọng đúng chỗ, giúp người học tham chiếu nhịp điệu, trọng âm và ngữ điệu chuẩn.

Một phiên luyện nói–nghe điển hình diễn ra như sau: người học phát âm, STT chuyển âm thanh thành văn bản, văn bản được chuẩn hoá (tokenization) trước khi so khớp với câu đích hoặc ý định mục tiêu. Hệ thống phân tích cú pháp–ngữ pháp, nhận diện lỗi và đồng thời chấm điểm phát âm bằng căn chỉnh âm vị để chỉ ra âm khó/nhầm lẫn. Ngay tức thì, NLG sinh phản hồi ngắn gọn gồm báo lỗi, gợi ý sửa và ví dụ đúng. Nếu cần TTS phát lại mẫu chuẩn. Cuối cùng, kết quả được ghi log để mô-đun gợi ý đề xuất bài/đoạn hội thoại tiếp theo phù hợp điểm yếu và sở thích. Trạng thái tiến độ (XP, streak, kỹ năng con) cũng được cập nhật theo tuần/tháng.

Trong triển khai thực tế, hệ thống đối mặt với thách thức về đa dạng accent và điều kiện thu âm, đòi hỏi dữ liệu phong phú và kỹ thuật thích nghi miền nguy cơ hiểu sai ngữ cảnh được giảm bằng ràng buộc ý định và tri thức chủ đề, phản hồi phải mang tính sư phạm — không chỉ “đúng/sai” mà còn giải thích “vì sao sai” và “sửa thế nào”. Về vận hành, các thành phần streaming phải duy trì độ trễ thấp để đối thoại tự nhiên, đồng thời tuân thủ nghiêm ngặt yêu cầu ẩn danh hoá, mã hoá và lưu trữ tối thiểu nhằm bảo vệ dữ liệu người học.

1.5.4.2 Phản hồi tự động và chấm điểm.

Các hệ thống chấm điểm bài luận tự động (Automated Essay Scoring – AES) sử dụng NLP để đánh giá độ mạch lạc, ngữ pháp, từ vựng của bài viết. Chúng cung cấp nhận xét chi tiết, nhất quán và khách quan, giúp người học cải thiện chất lượng viết [6], [17].

Nghiên cứu cho thấy những hệ thống này có độ chính xác chấm điểm tương đương người, và việc phản hồi gần như tức thì cho phép học viên liên tục chỉnh sửa và nâng cao bài viết của mình.

1.5.4.3 Trợ lý hội thoại và mô phỏng đối thoại

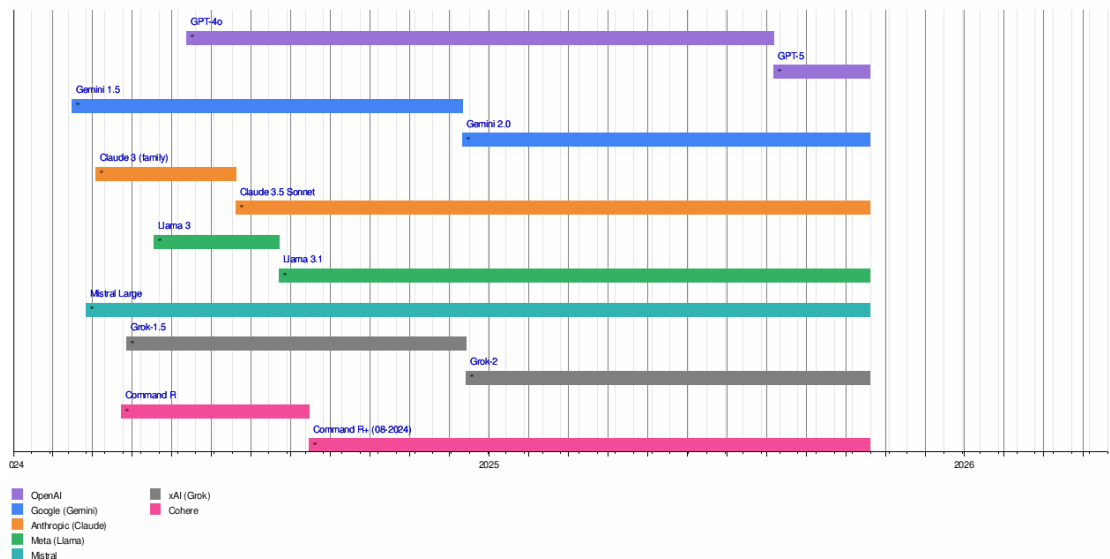
NLP cho phép tạo ra các chatbot hoặc đối tượng hội thoại ảo để người học thực hành giao tiếp [53]. Một số ứng dụng học tiếng Anh tích hợp chatbot đóng vai người bản ngữ, giúp học viên rèn luyện hội thoại ở nhiều tình huống khác nhau. Những chatbot này sử dụng mô hình ngôn ngữ để hiểu câu hỏi của học viên và phản hồi một cách tự nhiên. Chúng có thể nhận diện ý định của người học (intent) – ví dụ phân biệt liệu học viên đang hỏi thông tin, yêu cầu giải thích từ vựng hay chỉ chào hỏi xã giao – từ đó đưa

ra câu trả lời phù hợp. Nhờ NLP, các mô phỏng hội thoại trở nên sống động và thực tế hơn, giúp người học tự tin giao tiếp bằng ngoại ngữ.

1.6 Tổng quan về mô hình Ngôn ngữ Lớn

1.6.1 Giới thiệu về mô hình Ngôn ngữ Lớn

LLM là các mô hình học sâu có quy mô cực lớn (hàng tỷ tham số) được huấn luyện trên lượng dữ liệu văn bản khổng lồ thông qua phương pháp học tự giám sát (self-supervised learning) [15]. Về bản chất, LLM là mô hình xác suất học cách dự đoán từ (token) tiếp theo trong chuỗi dựa trên ngữ cảnh trước đó. Tuy nhiên, nhờ kích thước tham số khổng lồ và được tiếp xúc với kho ngữ liệu bao quát tri thức nhân loại, LLM có khả năng thực hiện đa dạng các nhiệm vụ NLP (như tóm tắt, dịch thuật, trả lời câu hỏi) mà không cần huấn luyện lại chuyên biệt cho từng tác vụ [15], [16].



Hình 1. 11: Dòng thời gian phát triển của các LLM tiêu biểu (GPT, Llama, Gemini...)

1.6.2 Kiến trúc kỹ thuật cơ bản của LLM

1.6.2.1 Cơ chế Attention

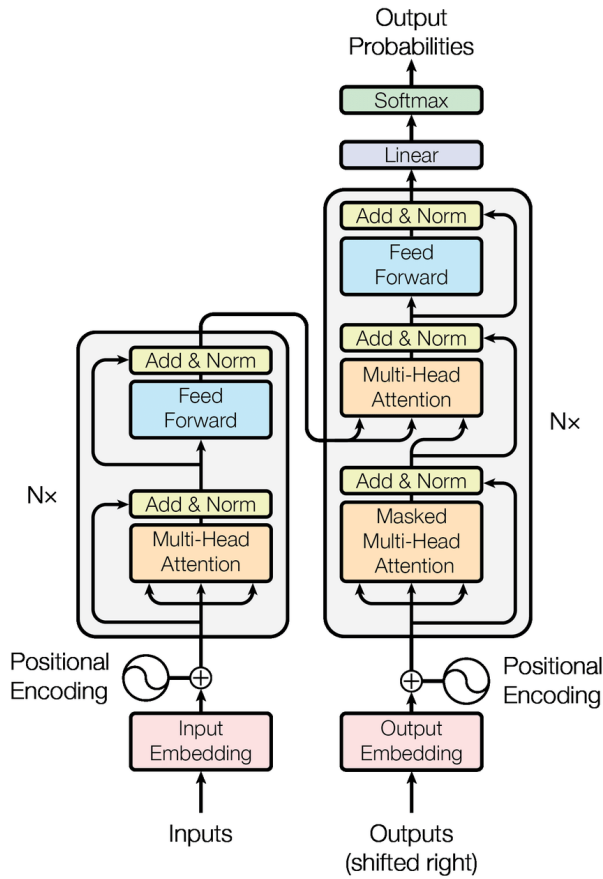
Cơ chế Attention là trái tim của các mô hình ngôn ngữ hiện đại, cho phép mô hình xác định mức độ quan trọng của từng từ trong câu đầu vào đối với từ đang được xử lý hoặc dự đoán. Khác với các mô hình tuần tự cũ, Attention tạo kết nối trực tiếp giữa mọi vị trí từ, giúp xử lý hiệu quả các mối quan hệ phụ thuộc xa (long-range dependencies) [26].

Trong kiến trúc Transformer, cơ chế này được triển khai dưới dạng Multi-head Self-attention. Về mặt toán học, Attention được mô tả như một hàm ánh xạ truy vấn (Q-Query) và một tập hợp các cặp khóa-giá trị (K - Key, V - Value) tới đầu ra:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (1.14)$$

Trong đó, trọng số chú ý được tính toán dựa trên độ tương đồng giữa Q và K, sau đó dùng để tổng hợp thông tin từ V. Việc sử dụng "đa đầu" (multi-head) cho phép mô hình học được nhiều khía cạnh ngữ nghĩa khác nhau (như quan hệ cú pháp, ngữ nghĩa, đồng tham chiếu) song song trong cùng một lớp [26].

1.6.2.2 Kiến trúc Encoder-Decoder



Hình 1. 12: Kiến trúc Transformer

Đây là kiến trúc nền tảng cho các mô hình chuyển đổi chuỗi sang chuỗi (sequence-to-sequence) [27]. Kiến trúc này gồm hai thành phần chính:

- Encoder (Bên trái - Bộ mã hóa): Tiếp nhận chuỗi đầu vào và chuyển đổi nó thành một biểu diễn vector ngữ nghĩa (context vector).

- + Input Embedding: Chuyển đổi các từ (token) đầu vào thành các vector số học.
- + Positional Encoding (Mã hóa vị trí): Vì Transformer xử lý toàn bộ câu cùng lúc (song song) chứ không tuần tự như RNN, nó không biết từ nào đứng trước, từ nào đứng sau. Thành phần này cộng thêm thông tin vị trí vào vector để mô hình biết thứ tự của từ.
- + Multi-Head Attention (Chú ý đa đầu): Đây là cơ chế Self-Attention. Nó cho phép mỗi từ trong câu "nhìn" vào tất cả các từ khác trong cùng câu đó để hiểu mối quan hệ (ví dụ: từ "nó" ám chỉ danh từ nào phía trước).
- + Add & Norm: Kết nối tắt (residual connection) cộng với chuẩn hóa lớp (Layer Normalization) giúp quá trình huấn luyện ổn định và sâu hơn.
- + Feed Forward: Mạng nơ-ron truyền thẳng để xử lý thông tin sau khi đã qua lớp Attention.
- Decoder (Bên phải - Bộ giải mã): Sử dụng biểu diễn từ Encoder để sinh ra chuỗi đầu ra mong muốn từng bước một.
 - + Output Embedding (Shifted Right): Đầu vào của Decoder là chuỗi đích đã sinh ra ở bước trước đó.
 - + Masked Multi-Head Attention: Đây là điểm khác biệt quan trọng. Cơ chế Masked (Che) đảm bảo rằng khi dự đoán một từ tại vị trí t , mô hình chỉ được phép nhìn thấy các từ từ vị trí 1 đến $t-1$. Nó che đi các từ trong tương lai để mô hình không "nhìn trộm" đáp án.
 - + Multi-Head Attention (Lớp giữa): Đây thường gọi là Cross-Attention.
 - Nó lấy Query (Q) từ tầng dưới của Decoder (thông tin từ đã sinh ra).
 - Nó lấy Key (K) và Value (V) từ đầu ra của khối Encoder bên trái.
 - Ý nghĩa: Bước này giúp Decoder "nhìn lại" câu gốc bên Encoder để quyết định xem cần tập trung vào từ nào để dịch hoặc sinh từ tiếp theo cho chính xác.
 - + Linear & Softmax: Lớp cuối cùng chuyển đổi vector đầu ra thành xác suất để chọn từ tiếp theo trong từ điển.

Tóm tắt luồng hoạt động:

- Dữ liệu đi vào bên trái (Encoder) được mã hóa thành vector ngữ cảnh (chứa K và V).
- Dữ liệu mục tiêu (đã sinh ra trước đó) đi vào bên phải (Decoder) qua lớp Masked Attention.
- Hai luồng này gặp nhau ở lớp Cross-Attention giữa Decoder.
- Cuối cùng, Decoder sinh ra từ tiếp theo qua lớp Linear/Softmax.

Ban đầu, kiến trúc này được xây dựng dựa trên mạng RNN hoặc LSTM [27]. Tuy nhiên, sự ra đời của Transformer [26] đã thay thế hoàn toàn RNN nhờ khả năng huấn luyện song song.

- Transformer Encoder: Gồm các lớp self-attention hai chiều (bidirectional), tối ưu cho việc hiểu ngữ cảnh toàn cục (Ví dụ: BERT).
- Transformer Decoder: Gồm các lớp masked self-attention (chỉ nhìn thấy quá khứ) và cross-attention (nhìn vào output của Encoder), tối ưu cho việc sinh văn bản (Ví dụ: GPT chỉ sử dụng phần Decoder [15]).

Việc hiểu rõ kiến trúc này giúp lựa chọn mô hình phù hợp: Encoder mạnh về phân loại và hiểu (NLU), Decoder mạnh về sáng tạo và sinh ngữ (NLG), trong khi kiến trúc Encoder-Decoder đầy đủ (như T5, BART) phù hợp cho các tác vụ chuyển đổi như dịch máy hay tóm tắt văn bản.

CHƯƠNG 2: PHƯƠNG PHÁP ĐỀ XUẤT XÂY DỰNG HỆ THỐNG HỌC NGOẠI NGỮ TÍCH HỢP TRÍ TUỆ NHÂN TẠO

2.1 Tổng quan về phương pháp tiếp cận

Đề tài đề xuất một phương pháp tiếp cận tích hợp toàn diện các mô hình Trí tuệ nhân tạo (AI) tiên tiến vào một chu trình học tập khép kín. Mục tiêu cốt lõi của phương pháp này là kiến tạo một môi trường mô phỏng giao tiếp thực tế, nơi người học không chỉ tiếp nhận kiến thức thụ động mà còn tương tác trực tiếp qua giọng nói và văn bản với độ phản hồi gần thời gian thực (near real-time latency).

Phương pháp đề xuất được xây dựng dựa trên hai trụ cột kỹ thuật chính: (1) Hội thoại thông minh theo ngữ cảnh (Contextual Roleplay) và (2) Đánh giá phát âm tự động (Automated Pronunciation Assessment). Để hiện thực hóa hai trụ cột này, hệ thống phối hợp bốn nhóm công nghệ lõi:

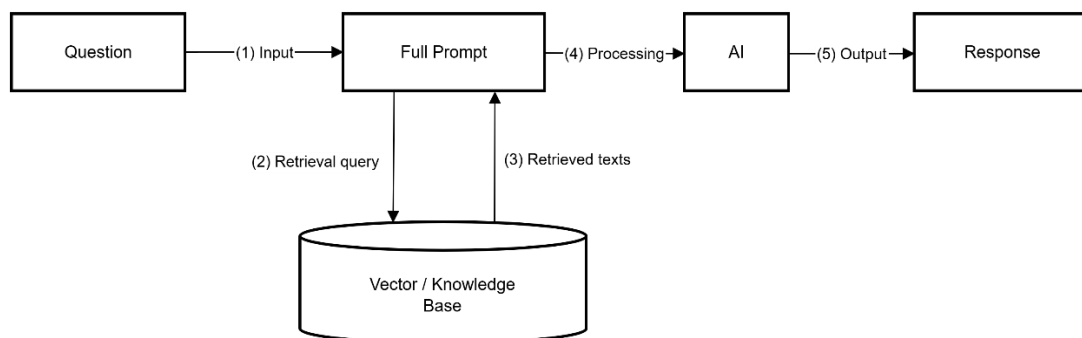
- Kỹ thuật truy xuất dữ liệu tăng cường (RAG - Retrieval-Augmented Generation): Đảm bảo nội dung hội thoại của AI luôn bám sát kịch bản giáo dục, khắc phục nhược điểm "ảo giác" (hallucination) của các mô hình ngôn ngữ lớn.
- Mô hình ngôn ngữ lớn (LLM): Đóng vai trò là động cơ sinh ngữ tự nhiên, cho phép AI nhập vai nhân vật linh hoạt và xử lý đa dạng các tình huống giao tiếp.
- Công nghệ nhận dạng giọng nói (ASR): Chuyển đổi tín hiệu âm thanh của người học thành dữ liệu văn bản để phân tích nội dung và phát hiện lỗi sai.
- Công nghệ tổng hợp giọng nói (TTS): Cung cấp mẫu phát âm chuẩn và phản hồi bằng giọng nói tự nhiên, giúp tăng cường kỹ năng nghe cho người học.

Sự kết hợp chặt chẽ giữa các thành phần này cho phép hệ thống không chỉ sửa lỗi ngữ pháp hay từ vựng đơn thuần mà còn đánh giá được mức độ trôi chảy, ngữ điệu và sự phù hợp về ngữ nghĩa của người học trong từng bối cảnh cụ thể.

2.2 Kiến trúc xử lý thông tin

2.2.1. Quy trình xử lý hội thoại thông minh

Quy trình xử lý hội thoại được thiết kế nhằm giải quyết bài toán cân bằng giữa sự linh hoạt trong giao tiếp tự nhiên và tính chính xác của nội dung bài học. Hệ thống vận hành dựa trên kiến trúc RAG, bắt đầu ngay khi người dùng khởi tạo một bài học hoặc gửi một tin nhắn.



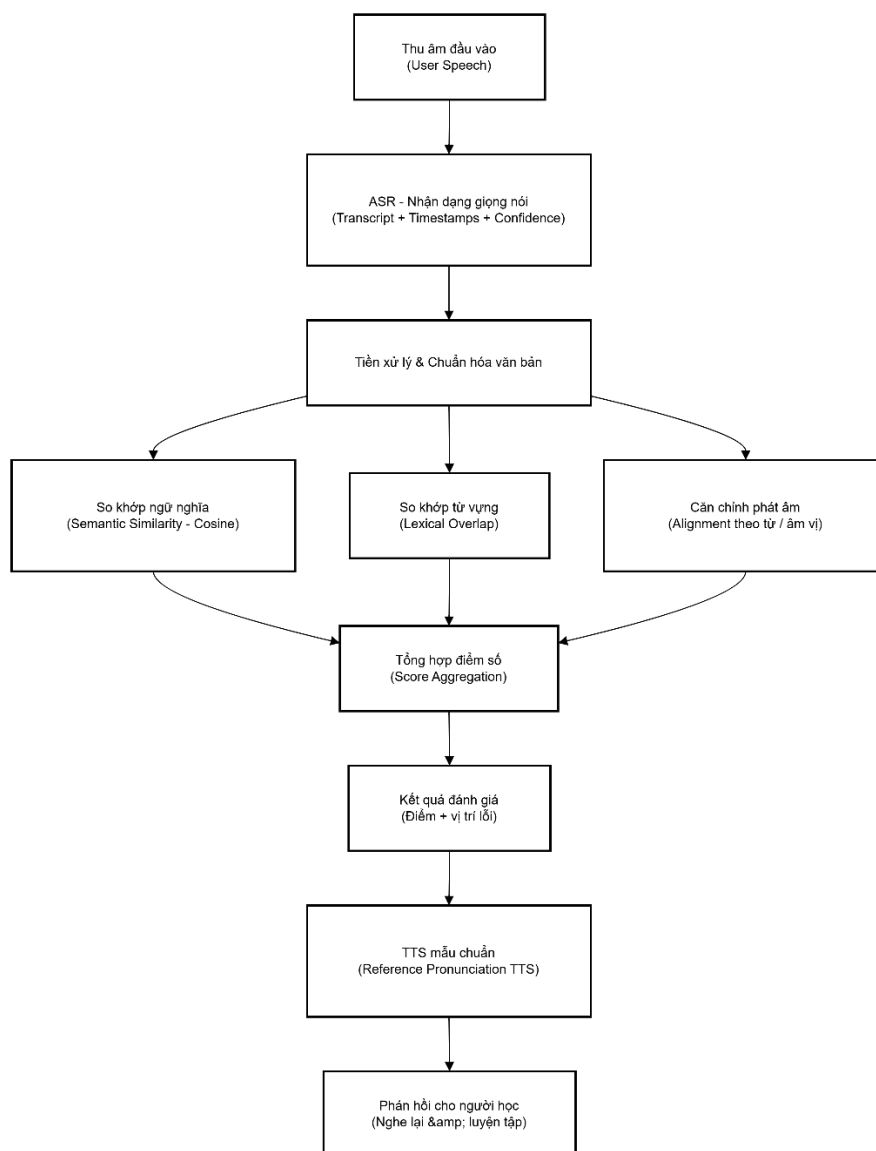
Hình 2. 1: Kiến trúc RAG

Kiến trúc RAG kết hợp hai thành phần cốt lõi: truy xuất tri thức (retrieval) và mô hình sinh ngôn ngữ (generation)[41]. Khi người dùng gửi một câu hỏi, hệ thống không truyền trực tiếp đầu vào này vào mô hình AI, mà trước hết xây dựng một prompt sơ khởi chứa nội dung câu hỏi và các ràng buộc hội thoại.

Từ prompt này, hệ thống tạo một truy vấn nhúng (retrieval query) để tìm kiếm thông tin liên quan trong Vector / Knowledge Base. Cơ sở tri thức này lưu trữ các kịch bản hội thoại, từ vựng trọng tâm và mẫu câu được mã hóa bằng vector. Các tài liệu có độ tương đồng cao nhất được trả về dưới dạng retrieved texts và được bổ sung vào prompt, tạo thành full prompt.

Full prompt, gồm cả câu hỏi của người dùng và tri thức được truy xuất, được đưa vào mô hình AI để sinh phản hồi. Nhờ việc cung cấp đúng ngữ cảnh và thông tin chuyên biệt, RAG giảm hiện tượng mô hình “bịa” thông tin, nâng cao độ chính xác và giúp phản hồi bám sát nội dung bài học. Phản hồi cuối cùng được gửi trả cho người dùng, hoàn thiện một chu trình tương tác.

2.2.2. Quy trình đánh giá phát âm



Hình 2. 2: pipeline quy trình đánh giá phát âm

Quy trình đánh giá phát âm tập trung vào việc lượng hóa kỹ năng nói thông qua cơ chế so sánh đối chiếu đa chiều giữa đầu vào của người học và kịch bản mẫu.

Quá trình bắt đầu bằng việc Thu thập và nhận dạng. Khi người học thực hiện phát âm, tín hiệu âm thanh được thu thập và gửi đến mô hình ASR [2] [3] [21]. Tại đây, âm thanh được chuyển đổi thành văn bản (transcript) kèm theo các thông tin chi tiết về mốc thời gian (timestamp) của từng từ và độ tin cậy (confidence score).

Bản phiên âm này sau đó được đưa vào quy trình So khớp và Chấm điểm. Hệ thống không chỉ so sánh chuỗi ký tự đơn thuần mà sử dụng thuật toán Cosine Similarity [38]

để đánh giá độ tương đồng về mặt ngữ nghĩa (Semantic Similarity). Điều này cho phép hệ thống chấp nhận các câu trả lời có cách diễn đạt khác với văn bản mẫu nhưng vẫn đúng về ý nghĩa. Song song đó, các chỉ số trùng lặp từ vựng (Lexical Overlap) [40] được tính toán để kiểm tra độ chính xác của từ ngữ.

Kết quả từ quá trình này được tổng hợp để đưa ra điểm số cuối cùng. Đồng thời, hệ thống sử dụng thông tin căn chỉnh (alignment) để chỉ ra các từ bị phát âm sai hoặc bỏ sót, và kích hoạt mô hình TTS [20], để cung cấp âm thanh mẫu giúp người học thực hành lại ngay lập tức.

2.2.3 Thu thập và tạo dữ liệu

Dữ liệu là thành phần trung tâm trong thiết kế và vận hành hệ thống học ngoại ngữ. Quá trình thu thập và tạo dữ liệu bao gồm nhiều nguồn và phương pháp, trải rộng trên các module: từ bài học lý thuyết, kỹ năng luyện tập, từ vựng, đến hội thoại nhập vai (Roleplay). Dữ liệu được chuẩn hóa, gắn nhãn và lưu trữ có tổ chức để phục vụ nhiều mục tiêu: cá nhân hóa học tập, huấn luyện mô hình AI, và phân tích hành vi người học.

Bảng 2. 1: Loại dữ liệu trong hệ thống

Loại dữ liệu	Mô tả
Bài học	Cấu trúc theo topic, cấp độ CEFR [50], có tiêu đề, nội dung bài, và mục tiêu.
Kỹ năng	Bao gồm các dạng luyện tập như trắc nghiệm, điền chỗ trống, nói, viết...
Câu hỏi	Đa dạng định dạng: lựa chọn, sắp xếp, nghe-hiểu, phát âm, v.v.
Từ vựng	Bao gồm từ/cụm từ, định nghĩa, ví dụ, âm thanh, loại từ, tag phân loại.
Hội thoại	Các kịch bản hội thoại thực tế, gán vai trò nhân vật và mục tiêu giao tiếp
Dữ liệu người dùng	Lịch sử học, câu trả lời, lỗi sai, tiến độ, điểm kỹ năng.
Tương tác học tập	Dữ liệu hành vi: thời gian phản hồi, số lần lặp lại, tương tác với AI.

Phương pháp thu thập dữ liệu:

- Nhập liệu có kiểm soát (curated input): Sử dụng hệ thống admin hoặc các API để nhập nội dung bài học, kỹ năng và kịch bản hội thoại
- Trích xuất từ tài nguyên sẵn có: Sử dụng các tài nguyên công cộng (dữ liệu CEFR, từ điển mở, kịch bản hội thoại mẫu) [17] để sử dụng cho train AI Roleplay
- Tự động sinh nội dung (semi-auto generation): Hệ thống có thể tự sinh Lesson cho từng Topic dựa trên mẫu hoặc AI (qua endpoint /auto-generate-lessons). Với Roleplay, có thể tạo câu hỏi warm-up, từ vựng liên quan, hoặc gợi ý lời thoại AI bằng LLM [15], [16], [18].
- Ghi nhận từ người học (user-generated): Hệ thống lưu lại tương tác thật như câu trả lời, lỗi sai, phát âm, nhật ký trả lời câu hỏi. Dữ liệu này là đầu vào để phân tích tiến độ học và điều chỉnh nội dung phù hợp.

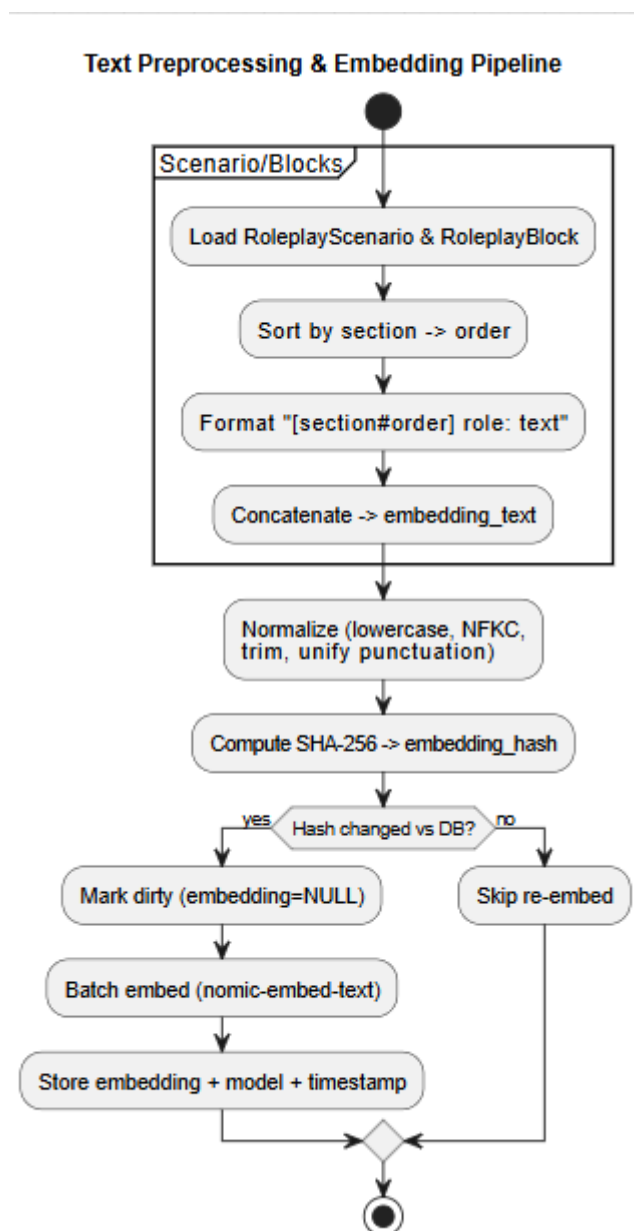
Chuẩn hóa và nhúng ngữ nghĩa (embedding) gồm toàn bộ dữ liệu ngôn ngữ đều được chuẩn hóa và sinh vector embedding phục vụ:

- Tìm kiếm ngữ nghĩa (semantic search) trong các block hội thoại đoạn từ vựng, hoặc ngữ cảnh trả lời.
- Đánh giá nội dung người học sinh ra bằng cách so sánh câu nói với câu mẫu qua cosine similarity.
- Gợi ý AI phù hợp ngữ cảnh thông qua RAG (retrieval-augmented generation) [41] như với Gemini/Ollama [41], [44].

Gắn nhãn và kiểm tra chất lượng: Dữ liệu được gắn tag theo chủ đề, loại kỹ năng và cấp độ CEFR (A1-B2). Các câu hỏi, kỹ năng, kịch bản đều được kiểm duyệt qua workflow để đảm bảo đầy đủ thông tin, phù hợp độ khó, đúng chuẩn ngữ pháp/ngữ cảnh. Các script và bộ kiểm định API cho phép kiểm tra lượt thoại và logic flow.

2.3 Tiền xử lý và phân đoạn dữ liệu

2.3.1 Tiền xử lý văn bản



Hình 2. 3: Sơ đồ hoạt động tiền xử lý và nhúng văn bản

Hệ thống tiến hành tiền xử lý văn bản cho nội dung kịch bản trước khi thực hiện nhúng embedding. Mỗi kịch bản được cấu thành từ nhiều khối hội thoại. Mỗi khối chứa các thuộc tính như section (loại nội dung, ví dụ: background, instruction, dialogue, ...), order (thứ tự trong loại nội dung đó), role (vai người nói: giáo viên, học viên A/B, người dẫn chuyện, v.v.) và text (nội dung thoại). Trước hết, hệ thống phân đoạn kịch bản thành các khối như vậy để đảm bảo cấu trúc hội thoại rõ ràng. Các khối được sắp xếp theo thứ tự ưu tiên dựa trên loại nội dung (section) và thứ tự xuất hiện (order). Nhờ đó, các phần

như Bối cảnh (background), Hướng dẫn (instruction) sẽ xuất hiện trước, sau đó đến Đối thoại (dialogue) theo đúng trật tự logic của kịch bản.

Tiếp theo, nội dung của toàn bộ kịch bản được chuẩn hóa định dạng và tổng hợp thành một trường văn bản duy nhất gọi là `embedding_text`. Quá trình chuẩn hóa bao gồm việc loại bỏ khoảng trắng thừa, ký tự xuống dòng không cần thiết và ghép nối các đoạn thoại theo định dạng thống nhất. Cụ thể, mỗi khối `RoleplayBlock` được chuyển thành một dòng văn bản theo mẫu:

`"[<section>#<order>] <role>: <text>"`.

Ví dụ, một khối đối thoại có section “dialogue” thứ tự 3, vai giáo viên, nội dung “Hello class” sẽ được định dạng thành:

`"[dialogue#3] teacher: Hello class"`

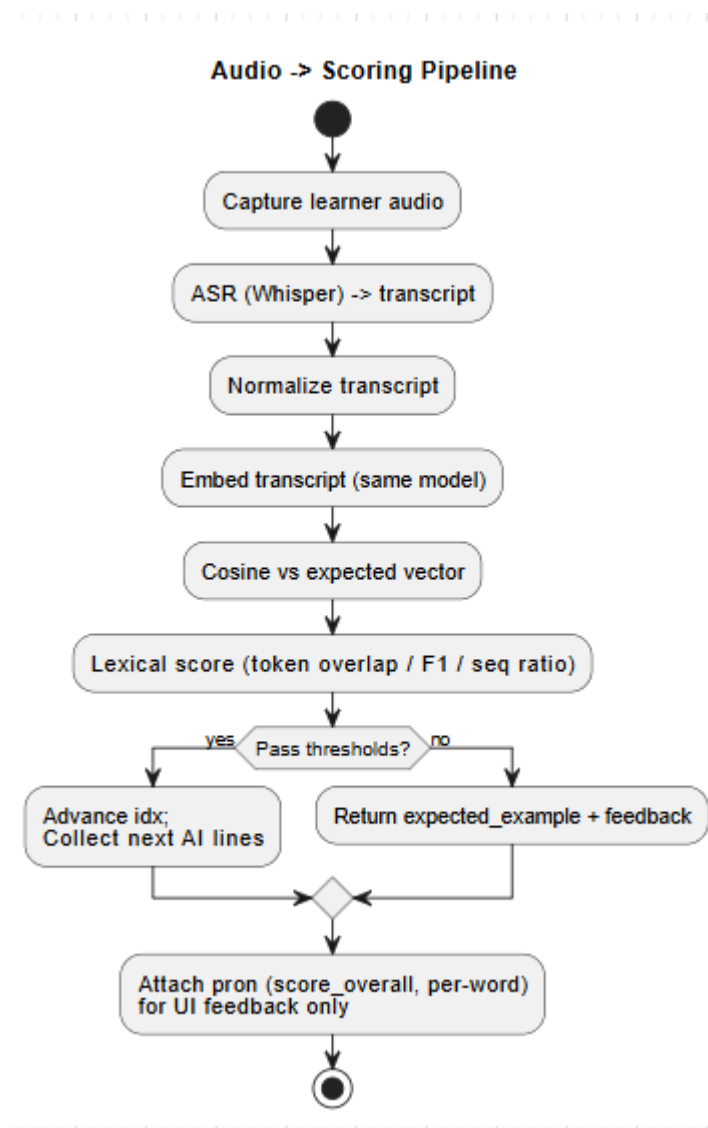
Tất cả các dòng này được ghép lại theo đúng trình tự sắp xếp để hình thành `embedding_text` hoàn chỉnh của kịch bản. Việc chuẩn hóa văn bản trước khi embedding giúp đảm bảo nội dung đầu vào cho mô hình nhúng là nhất quán, loại bỏ nhiều định dạng, và duy trì ngữ cảnh (thông qua việc ghi chú section và vai nói).

Sau khi có `embedding_text`, hệ thống tính toán một hàm băm (SHA-256) đại diện cho nội dung này. Mục đích của `embedding_hash` này là theo dõi phiên bản nội dung đã nhúng nếu người soạn bài học chỉnh sửa kịch bản (thay đổi câu thoại, thêm/bớt khối, v.v.), hàm băm mới sẽ khác với giá trị cũ lưu trong cơ sở dữ liệu. Khi đó, hệ thống đánh dấu kịch bản là “cần embedding lại” bằng cách đặt trường embedding về None (rỗng) và cập nhật `embedding_text` và `embedding_hash` mới. Nhằm tự động xác định các đối tượng “bẩn” (dirty) – tức là nội dung đã thay đổi so với lần nhúng trước. Những đối tượng này sẽ được đưa vào pipeline nhúng lại.

Quy trình nhúng embedding sẽ lấy danh sách các kịch bản cần xử lý, mặc định là tất cả `RoleplayScenario` trong hệ thống, sau đó lọc ra các kịch bản dirty (trừ khi có yêu cầu nhúng tất cả thông qua tham số force). Với mỗi kịch bản đủ điều kiện, hệ thống sử dụng mô hình nhúng được cấu hình sẵn để chuyển `embedding_text` thành vector embedding. Quá trình này thực hiện theo lô (batch) để tối ưu hiệu suất: các `embedding_text` được gửi đến mô hình theo từng nhóm và thu về các vector tương ứng. Sau khi nhận kết quả, hệ thống cập nhật trở lại từng đối tượng kịch bản với vector embedding mới, đồng thời ghi nhận tên mô hình đã dùng và mốc thời gian cập nhật. Quy trình tương tự cũng áp dụng cho từng `RoleplayBlock` nếu hệ thống yêu cầu nhúng vector cho các khối riêng lẻ. Kết quả của bước tiền xử lý văn bản là mỗi kịch bản và

mỗi khối thoại đều có một vector embedding đại diện cho nội dung của nó, sẵn sàng cho các bước tìm kiếm và so khớp ngữ nghĩa sau này.

2.3.2 Tiền xử lý dữ liệu âm thanh



Hình 2. 4: Sơ đồ hoạt động của quy trình tính điểm âm thanh

Đối với dữ liệu âm thanh, hệ thống tập trung vào giọng nói phản hồi từ người học. Khi người học nói (phát âm câu thoại), âm thanh này được thu và chuyển qua bước nhận dạng giọng nói để lấy bản phiên âm (transcript). Cụ thể, hệ thống sử dụng mô hình Whisper (một mô hình nhận dạng tiếng nói mạnh mẽ của OpenAI) để tự động chuyển âm thanh thành văn bản [1], [2], [28]. Kết quả là lời nói của học viên được chuyển thành câu chữ, giúp hệ thống có thể so sánh với câu thoại mẫu mong đợi trong kịch bản.

Sau khi có transcript, hệ thống thực hiện đánh giá mức độ khớp giữa câu người học nói và câu thoại mẫu. Việc so sánh này được thực hiện dựa trên hai tiêu chí chính:

- Độ tương đồng ngữ nghĩa, tính bằng cosine similarity giữa vector embedding của câu người học và câu mẫu [38], [39], sử dụng cùng mô hình embedding như Sentence-BERT [22].
- Độ tương đồng từ vựng (lexical), được tính bằng các chỉ số như tỉ lệ từ chung, F1-score, token overlap dựa trên các phương pháp so khớp chuỗi và NLP cổ điển [31]–[35].

Kết quả từ hai chỉ số được kết hợp để đưa ra quyết định pass/fail. Một số ngưỡng điển hình:

- Nếu cosine similarity vượt 0.78 thì được xem là đạt.
- Nếu cosine hơi thấp nhưng điểm lexical cao hơn 0.82 vẫn có thể chấp nhận.
- Nếu lexical cực cao (trên 0.88) dù cosine thấp vẫn có thể đạt.

Ngược lại, nếu cả hai chỉ số đều thấp, hệ thống sẽ đánh dấu lượt nói chưa đạt và yêu cầu thử lại. Cơ chế kết hợp linh hoạt hai tiêu chí dựa trên embedding [22], cosine [38], lexical metrics [31]–[35] giúp chấm điểm công bằng và phù hợp cho môi trường học.

Ngoài transcript và các điểm so khớp nội dung, hệ thống còn thu dữ liệu về phát âm (pron). Các mô-đun ASR hoặc đánh giá phát âm cung cấp cấu trúc dữ liệu chứa điểm tổng quan, điểm từng từ, và chi tiết các lỗi âm vị. Tuy nhiên, các chỉ số này chỉ dùng để phản hồi cho người học, không dùng làm điểm chấm chính thức. Việc phân tách đánh giá nội dung (semantic + lexical) và đánh giá phát âm giúp đảm bảo trọng tâm sư phạm: ưu tiên nói đúng ý trước, sau đó mới cải thiện phát âm chi tiết, theo hướng tiếp cận của các nghiên cứu về đánh giá kỹ năng nói EFL [1], [2], [3].

2.4 So sánh và lựa chọn mô hình AI

Để đảm bảo hệ thống đạt được sự cân bằng tối ưu giữa hiệu năng, chi phí và khả năng đáp ứng thời gian thực, đề tài đã tiến hành khảo sát và so sánh các mô hình phổ biến hiện nay trên bốn tác vụ chính: Embedding, Sinh ngữ (LLM), Nhận dạng giọng nói (STT) và Tổng hợp giọng nói (TTS).

2.4.1. Lựa chọn mô hình Embedding

Trong kiến trúc RAG [41], chất lượng của vector embedding quyết định độ chính xác của việc truy xuất ngữ cảnh. Đề tài đã xem xét ba ứng viên tiềm năng:

Bảng 2. 2: So sánh mô hình embedding

Mô hình	Loại hình	Kích thước Vector	Cửa sổ ngữ cảnh	Hiệu năng (MTEB)	Khả năng triển khai
OpenAI text-embedding-3-small	Cloud API (Trả phí)	1536 chiều	8191 tokens	Rất cao	Phụ thuộc Internet
all-MiniLM-L6-v2	Local (Mã nguồn mở)	384 chiều	512 tokens	Trung bình	Chạy tốt trên CPU
nomic-embed-text	Local (Mã nguồn mở)	768 chiều	8192 tokens	Cao	Chạy tốt trên CPU/GPU

Hệ thống lựa chọn nomic-embed-text:latest (triển khai qua Ollama [44]). Mô hình này khắc phục được điểm yếu về độ dài ngữ cảnh của các mô hình BERT truyền thống (như MiniLM), đồng thời duy trì chất lượng embedding ổn định tương tự các mô hình dựa trên Sentence-BERT [22]. Khả năng triển khai cục bộ giúp tiết kiệm chi phí vận hành và đảm bảo bảo mật dữ liệu.

2.4.2. Lựa chọn mô hình LLM.

Đối với tác vụ đóng vai (Roleplay) và sinh phản hồi trong môi trường giáo dục, hệ thống đòi hỏi một mô hình có sự cân bằng hoàn hảo giữa ba yếu tố:

- Tốc độ suy luận (Inference Speed) để đảm bảo hội thoại thời gian thực.
- Khả năng hiểu ngữ cảnh dài (Long Context) để ghi nhớ toàn bộ lịch sử bài học
- Chi phí vận hành tối ưu.

Đề tài đã tiến hành so sánh ba mô hình đại diện cho các xu hướng hiện nay:

Bảng 2. 3: So sánh các mô hình LLM cho tác vụ Chat

Mô hình	GPT-4o (OpenAI)	Llama 3.1 8B (Local)	Gemini 2.5 Flash (Google)
Tốc độ phản hồi (Latency)	Trung bình (~80 tokens/s)	Phụ thuộc phần cứng (GPU)	Rất nhanh (>150 tokens/s)
Chi phí vận hành	Cao	Chi phí phần cứng/điện năng	Rất thấp

Cửa sổ ngữ cảnh (Context)	128K tokens	128K tokens	1M+ tokens
Khả năng Multimodal	Xuất sắc (Native)	Hạn chế (Cần module riêng)	Xuất sắc (Native Audio/Video)

Hệ thống sử dụng mô hình Gemini 2.5 Flash [18] với độ trễ thấp nhất (Lowest Latency): Trong ứng dụng luyện giao tiếp, việc phản hồi tức thì là yếu tố sống còn để duy trì cảm giác "hội thoại tự nhiên". Gemini 2.5 Flash được thiết kế tối ưu cho throughput cao, giúp giảm thời gian chờ đợi (Time to First Token - TTFT) xuống mức tối thiểu. Cửa sổ ngữ cảnh vượt trội: Với khả năng xử lý hơn 1 triệu tokens, hệ thống có thể nạp toàn bộ kịch bản, từ vựng, và lịch sử hội thoại dài của người học vào ngữ cảnh mà không cần áp dụng các kỹ thuật cắt gọt (truncation) phức tạp, giúp AI "nhớ" bài học tốt hơn hẳn Llama 3 [26] hay GPT-4o [16]. Hiệu quả chi phí: So với GPT-4o, Gemini 2.5 Flash cung cấp chất lượng sinh ngữ đủ tốt cho mức độ CEFR A1-B2 [50] với mức chi phí chỉ bằng một phần nhỏ, cho phép mở rộng quy mô người dùng lớn mà không gặp áp lực về ngân sách.

2.5 Thuật toán và kỹ thuật

2.5.1 Đánh giá phát âm

Đánh giá phát âm là một thành phần quan trọng trong hệ thống học ngoại ngữ tích hợp AI, cho phép hệ thống xác định mức độ chính xác, tự nhiên và rõ ràng của lời nói so với câu mẫu. Các phương pháp đánh giá hiện nay thường kết hợp nhiều kỹ thuật NLP và ASR như cosine similarity, lexical similarity, dynamic alignment và WER/CER để tạo điểm số tổng hợp phản ánh toàn diện chất lượng phát âm [21], [28], [35].

Ngoài Cosine similarity được trình bày trong công thức (1.2) (1.9) là phép đo thường dùng trong xử lý ngôn ngữ tự nhiên để đánh giá mức độ tương đồng giữa hai câu hoặc đoạn văn, được biểu diễn bằng vector ngữ nghĩa [22], [38], [39].

2.5.1.1 SequenceMatcher

$$Ratio = \frac{2 \cdot M}{T} \quad (2.1)$$

Với M là số lượng phần tử khớp và T là tổng số phần tử.

Phương pháp này hữu ích trong bối cảnh đánh giá phát âm khi một số lỗi nhỏ (ví dụ kéo dài âm, thiếu âm cuối) làm Whisper thay đổi một phần ký tự [28], [35].

2.5.1.2 Đo độ tương đồng từ vựng – F1 Score

Để đánh giá mức độ trùng khớp từ vựng giữa câu người học nói và câu mẫu, ta sử dụng F1-score dựa trên Precision và Recall [54] với công thức:

$$\text{Precision} = \frac{|A \cap B|}{|B|} \quad (2.2)$$

$$\text{Recall} = \frac{|A \cap B|}{|A|} \quad (2.3)$$

$$F_1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (2.4)$$

Trong đó:

- A: tập hợp các token (từ) của câu mẫu.
- B: tập hợp các token trong câu nói của người học.
- $|A \cap B|$: số lượng token trùng nhau giữa A và B.
- F_1 : trung bình điều hòa giữa Precision và Recall.

2.5.1.3 Loại Kết Quả Tốt Dựa Trên Ngưỡng Kép

Hệ thống sử dụng ngưỡng kép để xác định người học đã đạt hay chưa dựa trên Cosine và Lexical Similarity [38], [39]:

Biểu thức điều kiện hợp lệ:

$$\text{Đạt} \Leftrightarrow \begin{cases} (\text{Cosine} \geq 0.78) \\ (\text{Cosine} \geq 0.72 \wedge \text{Lexical} \geq 0.82) \\ (\text{Lexical} \geq 0.88) \end{cases} \quad (2.5)$$

2.5.1.4 Word Error Rate và Character Error Rate

Trong các hệ thống nhận diện giọng nói (STT – Speech to Text), độ chính xác được đánh giá bằng hai chỉ số phổ biến là Word Error Rate (WER) và Character Error Rate (CER). Đây là các phép đo khoảng cách chỉnh sửa (edit distance) giữa câu gốc và câu được nhận dạng [30], [35].

Công thức WER:

$$\text{WER} = \frac{S + D + I}{N} \quad (2.6)$$

Trong đó:

- S: số từ sai (substitutions) – bị nhận nhầm thành từ khác.
- D: số từ bị thiếu (deletions) – từ gốc không được nhận ra.
- I: số từ thừa (insertions) – hệ thống nhận sai thêm từ không tồn tại.
- N: tổng số từ trong câu gốc (reference).

WER thể hiện tỷ lệ sai lệch ở mức từ. Giá trị càng thấp thì độ chính xác càng cao. Mức WER chấp nhận được tùy thuộc ứng dụng, thường < 20% với các hệ thống mạnh.

Công thức CER :

$$CER = \frac{S_c + D_c + I_c}{N_c} \quad (2.7)$$

- S_c : Số ký tự bị thay thế sai (Substitutions).
- D_c : Số ký tự bị xóa (Deletions).
- I_c : Số ký tự bị chèn thừa (Insertions).
- N_c : Tổng số ký tự trong văn bản gốc (Number of characters).

2.5.2 Tìm kiếm ngữ liệu bằng vector

Tìm đoạn văn phù hợp từ kho dữ liệu dựa trên truy vấn của người học được mô hình hóa bằng truy vấn vector.

$$\arg \max_i \cos(\vec{q}, \vec{d}_i) \quad (2.8)$$

Trong đó với $\vec{q}$ là vector câu hỏi và $\vec{d}_i$ là các vector tài liệu.

2.5.3 Ưu tiên gợi ý dựa trên trọng số

Khi đề xuất bài luyện hoặc kỹ năng tiếp theo, hệ thống áp dụng phương pháp xếp hạng heuristic [51], [58].

$$S = w_1 M + w_2 T + w_3 D \quad (2.9)$$

Trong đó M tỷ lệ lỗi của người học ở kỹ năng này, T số ngày kể từ lần gần nhất luyện kỹ năng đó và D độ khó của kỹ năng. w_1, w_2, w_3 các trọng số do hệ thống điều chỉnh để tối ưu hóa khuyến nghị. Heuristic scoring thường được kết hợp với spaced repetition theo đường cong quên lãng [51], [59].

2.5.4 Đánh giá mô hình gợi ý

Hệ thống gợi ý bài học cá nhân hóa được xây dựng dựa trên kiến trúc lai (Hybrid Approach), kết hợp giữa Mô hình Học máy (Machine Learning) để dự đoán khả năng

hoàn thành bài học và Thuật toán Heuristic để tối ưu hóa lộ trình học theo phương pháp lặp lại ngắt quãng (Spaced Repetition) [52], [58].

2.5.4.1 Mô hình dự đoán khả năng hoàn thành

Mô hình dự đoán khả năng hoàn thành (Completion Prediction Model) Hệ thống sử dụng thuật toán Gradient Boosting Classifier (GBM). Đây là một kỹ thuật học máy tổ hợp (ensemble learning), xây dựng một mô hình dự đoán mạnh từ tập hợp các mô hình yếu thường là cây quyết định có độ sâu thấp. Mô hình dự đoán xác suất người học có thể hoàn thành bài học, ký hiệu:

$$P(y = 1 | X)$$

Trong đó đầu vào (Features) mô hình được huấn luyện dựa trên các đặc trưng hành vi học tập X được trích xuất từ lịch sử người dùng:

- X_1 : Tỷ lệ lỗi (Mistake Rate).
- X_2 : Tổng số lượt tương tác (Interaction Count).
- X_3 : Điểm trung bình (Average Score).
- $y \in \{0, 1\}$ là nhãn thực tế ($y = 1$ tương ứng với người học hoàn thành bài học, $y = 0$ nếu không hoàn thành).

2.5.4.2 Hàm mục tiêu của mô hình

Gradient Boosting tối ưu hàm mất mát Logarithmic Loss trên toàn bộ tập dữ liệu. Hàm mất mát được định nghĩa như sau:

$$L(y, p) = -[y \log(p) + (1 - y) \log(1 - p)] \quad (2.10)$$

Trong đó p là xác suất dự đoán của mô hình.

Mục tiêu là tìm bộ tham số tối ưu:

$$\theta^* = \arg \min_{\theta} \frac{1}{N} \sum_{i=1}^N L(y_i, p_{\theta}(x_i)) \quad (2.11)$$

2.5.4.3 Thuật toán xếp hạng ưu tiên

Thuật toán xếp hạng ưu tiên (Heuristic Ranking Strategy) Kết quả dự đoán của mô hình ML Sau khi mô hình dự đoán xác suất hoàn thành $p = P(y = 1 | X)$ được kết hợp với thuật toán xếp hạng dựa trên luật (Rule-based) để đảm bảo yếu tố sư phạm. Điểm ưu tiên *Score* của một bài học i được tính theo công thức tuyến tính có trọng số:

$$Score = w_{urg} \cdot U_{time} + w_{err} \cdot P_{error} + w_{prof} \cdot G_{prof} + w_{rec} \cdot G_{recency} \quad (2.12)$$

Trong đó:

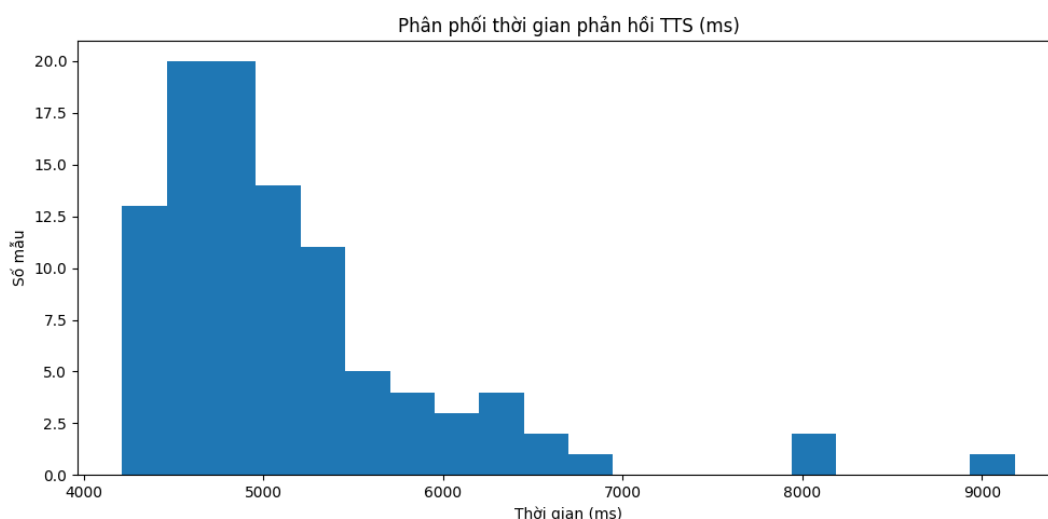
- U_{time} : Độ khẩn cấp theo thời gian (Review Urgency).
- P_{error} : Áp lực từ lỗi sai (Error Pressure).
- G_{prof} : Khoảng cách đến mức thành thạo (Proficiency Gap).
- $G_{recency}$: Độ mới của việc luyện tập (Recency Gap).
- w : Các trọng số tương ứng.

Chiến lược heuristic này được thiết kế dựa trên các nghiên cứu về hành vi học tập và cơ chế tạo động lực trong hệ thống e-learning [23], cũng như nguyên lý của Spaced Repetition được trình bày trong lý thuyết đường cong quên lãng của Ebbinghaus [59]. Việc kết hợp khả năng dự đoán của mô hình GBM với quy tắc xếp hạng heuristic cho phép hệ thống xây dựng lộ trình học cá nhân hóa, đúng thời điểm và phù hợp năng lực người học.

2.6 Kết quả thực nghiệm

Các thực nghiệm trong mục này nhằm đánh giá hiệu năng và chất lượng của hai mô-đun quan trọng trong hệ thống học phát âm và hệ gợi ý:

2.6.1 Thực nghiệm mô hình chuyển văn bản thành giọng nói



Hình 2. 5: Biểu đồ độ trễ TTS

Biểu đồ thể hiện sự phân bố thời gian phản hồi của mô-đun TTS khi sinh giọng nói từ 100 câu trong tập dữ liệu thực nghiệm. Đa số các mẫu có độ trễ nằm trong khoảng 4.5–5.5 giây, trong khi phần đuôi bên phải (right tail) xuất hiện một số mẫu có độ trễ cao hơn, lên đến 7–9 giây. Kết quả này cho thấy:

- Piper TTS duy trì độ trễ ổn định ở mức 4–6 giây với phần lớn câu.
- Những điểm có độ trễ cao xuất hiện khi văn bản dài hơn hoặc mô-đun phải xử lý thêm (ví dụ: spell-out ký tự, lưu file chậm, hệ thống có ít CPU rảnh).
- Phân phối lệch phải (right-skewed) là đặc trưng phổ biến trong bài toán TTS, đặc biệt khi chạy trên CPU theo dạng real-time synthesis.

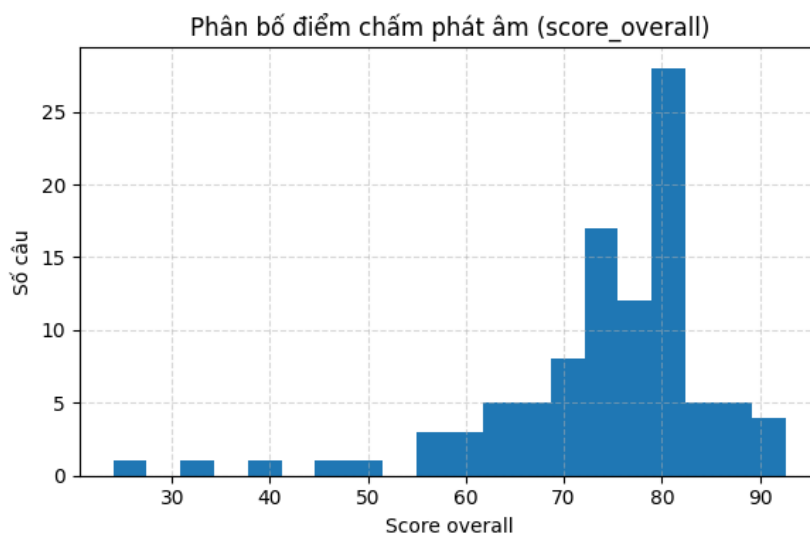
Độ trễ trung bình ở mức thấp giúp hệ thống đảm bảo khả năng phản hồi nhanh trong bài luyện phát âm và hội thoại.

2.6.2. Thực nghiệm mô hình chấm điểm phát âm

Bảng 2. 4: Độ trễ theo từng mô hình Whisper

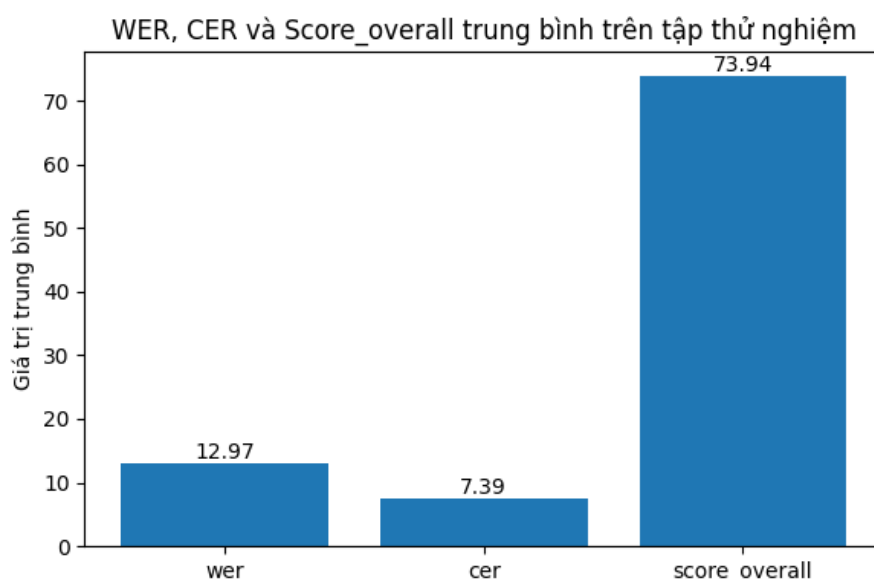
Whisper model	CPU thời gian (avg)	Ghi chú
tiny.en	700–900 ms	Nhanh nhất nhưng độ chính xác thấp hơn
base	1.9–2.4 s	Tối ưu nhất cho hệ thống
small	5–8 s	Chính xác hơn nhưng chậm đáng kể
medium	20–40 s	Không phù hợp cho real-time khi không có GPU

Thực nghiệm sử dụng 100 tệp âm thanh đầu tiên của tập LibriSpeech ASR (split: other/test), định dạng FLAC 16kHz. Tín hiệu đầu vào được chuẩn hóa bằng FFMPEG để đảm bảo đồng nhất trước khi đưa vào mô hình. Whisper thực hiện nhận dạng tiếng nói và sinh văn bản dự đoán kèm logits, là nền tảng để hệ thống tiến hành so khớp với câu mẫu. Quá trình alignment động (dynamic programming) được sử dụng để xác định lỗi thay thế, thiếu, thừa, từ đó tính các chỉ số WER và CER theo công thức (2.6) (2.7). Bên cạnh mức độ khớp từ vựng, hệ thống còn đánh giá trung bình confidence, thời lượng đọc và tốc độ đọc (syllables per second), cho phép mô hình chấm điểm không chỉ dựa vào lỗi từ vựng mà còn dựa trên độ rõ ràng và tính tự nhiên của phát âm. Điểm cuối cùng (score_overall) được tính trên thang 0–100 dựa trên tổ hợp có trọng số các đặc trưng này.



Hình 2. 6: Biểu đồ phân phối điểm tổng

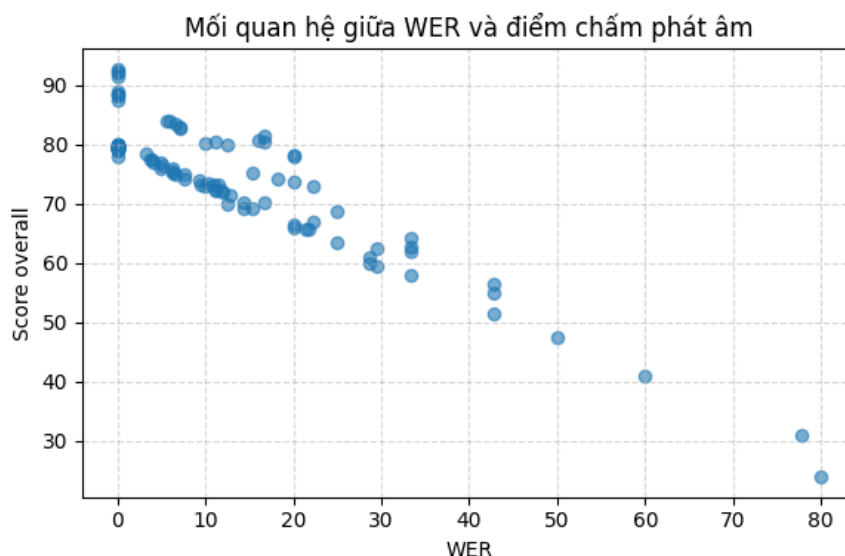
Phân phối điểm tổng của 100 mẫu. Phần lớn mẫu tập trung trong khoảng 70–85 điểm, phản ánh khả năng nhận dạng tốt của Whisper-base trên dữ liệu có chất lượng cao. Những mẫu có điểm thấp chủ yếu xuất phát từ tín hiệu mờ, nhiễu, hoặc phát âm không rõ ràng, khiến mô hình gặp khó khăn trong việc xác định chính xác chuỗi âm vị. Phân phối điểm này phù hợp với nhận định chung trong các nghiên cứu về ASR trên dữ liệu chuẩn hóa [2], [3], [21], [28].



Hình 2. 7: Biểu đồ giá trị trung bình WER và CER

Trình bày giá trị trung bình của WER, CER và score_overall. WER trung bình đạt 12.97% và CER đạt 7.39%, cho thấy độ sai lệch thấp trên tập dữ liệu kiểm thử. Điểm

trung bình đạt 73.94, phản ánh sự kết hợp hợp lý giữa các đặc trưng âm học và ngôn ngữ, thay vì chỉ dựa vào số lỗi nhận dạng.



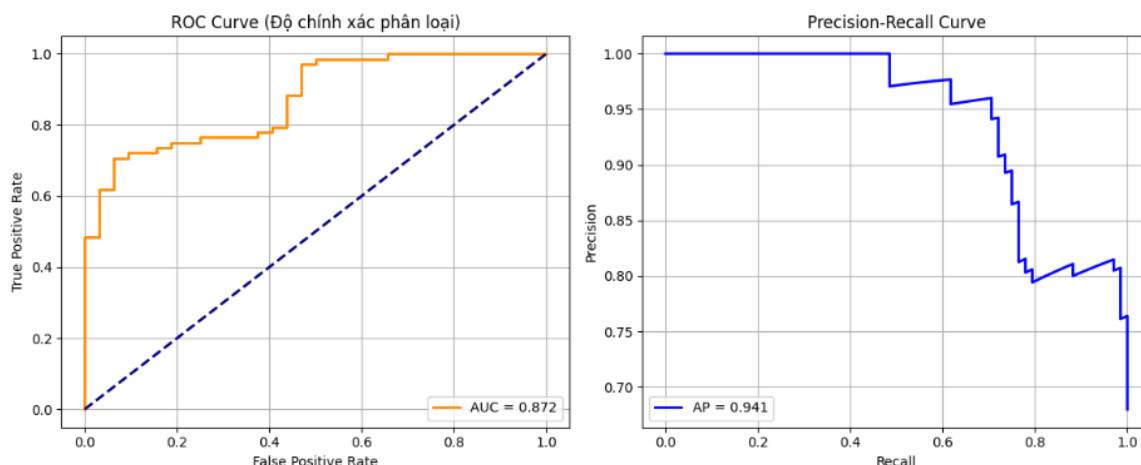
Hình 2. 8: Biểu đồ mối quan hệ giữa WER và điểm phát âm

Mối tương quan nghịch rõ rệt: WER càng thấp, điểm chấm phát âm càng cao, và điểm giảm dần khi WER tăng. Các mẫu có WER vượt 40–60% thường có điểm dưới 50, cho thấy hệ thống nhận dạng khó tái tạo chính xác nội dung người nói. Tuy vậy, một số điểm ngoại lệ vẫn xuất hiện do mô hình đánh giá thêm các yếu tố khác như tốc độ nói bất thường, độ dài nguyên âm, hoặc confidence thấp ở những đoạn khó nghe. Điều này chứng minh hệ thống chấm điểm không phụ thuộc duy nhất vào sai số nhận dạng mà sử dụng bộ tiêu chí đa chiều để đánh giá tổng quan chất lượng phát âm.

2.6.3 Thực nghiệm hệ ngoại ý

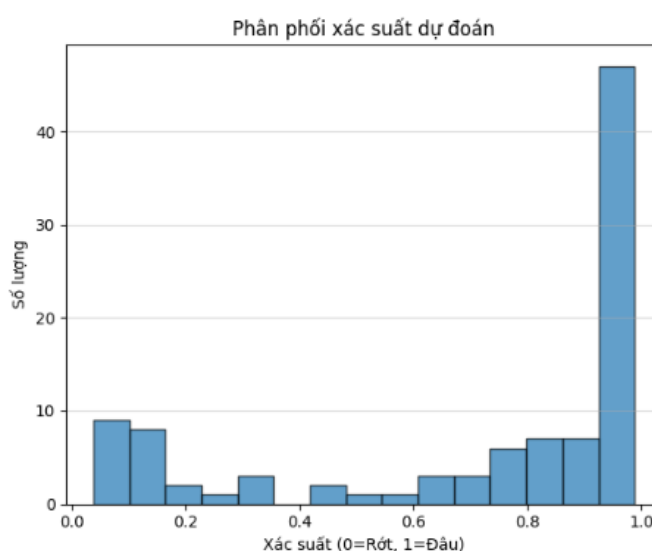
Quá trình thực nghiệm được tiến hành nhằm kiểm chứng khả năng hoạt động của kiến trúc lai (Hybrid Architecture) trong việc đưa ra các gợi ý bài học phù hợp. Thực nghiệm được chia thành hai giai đoạn: (1) Đánh giá độ chính xác của mô hình dự đoán khả năng hoàn thành (Gradient Boosting) và (2) Kiểm thử logic xếp hạng dựa trên luật (Heuristic Ranking) với các kịch bản người dùng giả định.

Dữ liệu đầu vào thực nghiệm được tiến hành trên bộ dữ liệu mẫu gồm 500 bản ghi ($N = 500$) chia theo tỉ lệ 80% train và 20% validation, mô phỏng lịch sử học tập của người dùng.



Hình 2. 9 : a, Biểu đồ Độ chính xác phân loại. b, Biểu đồ Precision-Recall

Sự tin cậy của mô hình được khẳng định qua các chỉ số đo lường tiêu chuẩn. Trên đường cong ROC (Receiver Operating Characteristic) hình 1.13a, mô hình đạt diện tích dưới đường cong (AUC) là 0.872, phản ánh khả năng phân tách tốt giữa hai lớp người dùng "Hoàn thành" và "Không đạt". Đặc biệt, biểu đồ Precision-Recall hình 1.13b ghi nhận chỉ số Average Precision (AP) lên tới 0.941. Kết quả này cho thấy mô hình duy trì được độ chuẩn xác (Precision) rất cao ngay cả khi độ phủ (Recall) tăng lên, đồng nghĩa với việc hệ thống có thể nhận diện chính xác hầu hết các người dùng tiềm năng sẽ hoàn thành bài học mà ít khi đưa ra báo động giả. Sự kết hợp giữa khả năng phân loại dứt khoát và độ chính xác cao tạo nền tảng vững chắc để hệ thống gợi ý đưa ra các chiến lược học tập phù hợp (như "Thử thách" hoặc "Ôn tập") ở giai đoạn xếp hạng tiếp theo.



Hình 2. 10: Biểu đồ phân phối xác suất dự đoán

Kết quả thực nghiệm trên tập dữ liệu kiểm thử cho thấy mô hình Gradient Boosting đạt hiệu năng cao và có khả năng đưa ra các quyết định dứt khoát. Cụ thể, biểu đồ phân phối xác suất dự đoán thể hiện một xu hướng phân cực rõ rệt (bimodal distribution). Phần lớn các dự đoán của mô hình tập trung ở hai ngưỡng xác suất cực trị: nhóm có nguy cơ rớt cao tập trung ở khoảng dưới 0.2 và nhóm có khả năng hoàn thành mục tiêu tập trung dày đặc ở ngưỡng trên 0.9. Đáng chú ý, vùng "không chắc chắn" (khoảng xác suất 0.4 - 0.6) chứa rất ít điểm dữ liệu, chứng tỏ các đặc trưng đầu vào có tính phân loại mạnh, giúp mô hình hiếm khi rơi vào trạng thái lưỡng lự khi phân lớp.

CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

3.1 Xác định yêu cầu

Nghiên cứu quy trình và luồng hoạt động của hệ thống: Tiến hành phân tích và nghiên cứu các quy trình hoạt động cơ bản của một hệ thống hỗ trợ học ngoại ngữ trực tuyến tích hợp trí tuệ nhân tạo, bao gồm: đăng ký tài khoản, lựa chọn ngôn ngữ, tham gia lộ trình học (theo chủ đề, bài học, kỹ năng), luyện phát âm, luyện hội thoại, ôn tập từ vựng, theo dõi tiến độ và nhận gợi ý học tập cá nhân hóa. Việc nghiên cứu này giúp đảm bảo hiểu rõ cấu trúc tổng thể, cách thức vận hành và các yêu cầu chức năng, phi chức năng cần thiết của hệ thống.

Nghiên cứu nhu cầu và yêu cầu người dùng: Tìm hiểu nhu cầu của hai nhóm người dùng chính là người học và quản trị viên hệ thống đối với một nền tảng học ngoại ngữ tích hợp trí tuệ nhân tạo, nhằm nâng cao hiệu quả tự học và tối ưu vận hành hệ thống:

- Người học cần một môi trường học ngoại ngữ linh hoạt, có lộ trình rõ ràng, nhiều dạng bài luyện nghe – nói – đọc – viết, hiển thị lỗi sai và gợi ý nội dung ôn tập phù hợp.
- Admin cần công cụ để quản lý người dùng, quản lý nội dung học (ngôn ngữ, chủ đề, bài học, kỹ năng, từ vựng, kịch bản hội thoại), giám sát hoạt động học tập và cấu hình các mô hình AI.

Xây dựng các chức năng từ yêu cầu người dùng: Từ nhu cầu của người học và admin, hệ thống được xây dựng với các nhóm chức năng chính sau:

- Lộ trình học và khóa học ngoại ngữ: Xây dựng lộ trình học theo ngôn ngữ và cấp độ (A1–C2), tổ chức nội dung theo chủ đề – bài học – kỹ năng. Người học có thể đăng ký ngôn ngữ muốn học, truy cập tài liệu, thực hiện bài tập và được hệ thống ghi nhận tiến độ, điểm số và thời gian học.
- Bài học và kỹ năng (Lesson – Skill): Thiết kế các bài học với nhiều dạng kỹ năng: nghe (listening), nói (speaking), đọc (reading), viết (writing), từ vựng (vocabulary), ngữ pháp (grammar) và các dạng bài tập trắc nghiệm/ghép đôi/điền khuyết/sắp xếp câu,... Hệ thống chấm điểm tự động, cập nhật tiến độ kỹ năng và xác định điểm mạnh – yếu của người học.

- Cung cấp chức năng luyện phát âm, trong đó người học: Nghe câu/từ mẫu bằng dịch vụ tổng hợp giọng nói. Ghi âm giọng nói của mình, hệ thống dùng nhận dạng giọng nói và mô hình đánh giá phát âm để so sánh với phát âm chuẩn. Nhận điểm số, nhận diện lỗi phát âm, gợi ý cải thiện.
- Ôn tập từ vựng với lặp lại ngắt quãng: Quản lý danh sách từ vựng đã học của người dùng, áp dụng thuật toán lặp lại ngắt quãng để xác định thời điểm ôn tập tiếp theo. Người học nhận danh sách từ cần ôn trong ngày, làm bài kiểm tra nhanh, hệ thống cập nhật mức độ ghi nhớ của từng từ và điều chỉnh lịch ôn tương ứng.
- Luyện hội thoại với AI: Cho phép người học tham gia các kịch bản hội thoại theo bối cảnh thực tế (nhà hàng, khách sạn, hỏi đường, phỏng vấn, ...). Người học có thể tương tác với AI bằng văn bản hoặc giọng nói, hệ thống đánh giá mức độ hoàn thành hội thoại, góp ý về từ vựng, ngữ pháp và độ tự nhiên.
- Công cụ tìm kiếm nội dung học: Xây dựng công cụ tìm kiếm nâng cao cho phép người học tìm bài học, kỹ năng, từ vựng, kịch bản hội thoại theo nhiều tiêu chí ngôn ngữ, cấp độ, chủ đề, từ khóa,...
- Thông báo và nhắc nhở học tập: Cho phép hệ thống gửi thông báo tới người học: nhắc lịch học, nhắc ôn từ vựng đến hạn, thông báo hoàn thành bài học, đạt huy hiệu mới, cập nhật bảng xếp hạng, hoặc thông tin về sự kiện, tính năng mới.
- Gamification và tương tác xã hội: Tích hợp các yếu tố game hóa như XP, huy hiệu (badge), bảng xếp hạng, bạn bè để tăng động lực học tập, khuyến khích học liên tục và tạo môi trường thi đua giữa các người học.

3.2 Phân tích yêu cầu

3.2.1. Phía người học

- Đăng ký và đăng nhập hệ thống: Người học tạo tài khoản và đăng nhập vào hệ thống bằng email và mật khẩu (hoặc các phương thức xác thực khác). Hệ thống cần hỗ trợ xác thực an toàn, cho phép người học đổi mật khẩu, khôi phục mật khẩu khi cần thiết.

- Người học có thể chọn ngôn ngữ và lộ trình học: Xem danh sách các ngôn ngữ được hỗ trợ. Đăng ký một hoặc nhiều ngôn ngữ muốn học. Xem lộ trình học theo cấp độ và các chủ đề tương ứng.
- Xem và tham gia bài học/kỹ năng: Người học có thể tìm kiếm, duyệt và tham gia các bài học/kỹ năng trong lộ trình. Xem mô tả bài học, mục tiêu, thời lượng, cấp độ. Thực hiện các dạng bài tập (trắc nghiệm, ghép đôi, điền khuyết, sắp xếp câu, nghe hiểu, đọc hiểu, viết,...). Sau khi hoàn thành, hệ thống tự động chấm điểm, cộng XP và cập nhật trạng thái hoàn thành của bài học.
- Ôn tập từ vựng: Người học được cung cấp danh sách từ vựng cần ôn trong ngày theo lịch ôn do thuật toán lặp lại ngắt quãng quyết định: Thực hiện các bài ôn tập nhanh (flashcard, trắc nghiệm, điền nghĩa,...). Mỗi câu trả lời được hệ thống ghi nhận (đúng/sai, mức độ dễ/khó) để tính toán thời điểm ôn tiếp theo, đảm bảo tối ưu hóa khả năng ghi nhớ.
- Luyện phát âm và hội thoại: Người học chọn câu/từ vựng, kịch bản roleplay theo chủ đề, bối cảnh, cấp độ cần luyện. Hệ thống phát audio mẫu, người học nghe và ghi âm lại. Hệ thống sử dụng STT và mô hình đánh giá phát âm để phân tích phần ghi âm, so sánh với chuẩn, trả về điểm số và gợi ý chỉnh sửa. Lịch sử luyện phát âm và lỗi phát âm được lưu để phục vụ phân tích và gợi ý luyện tập sau này. Đối với chế độ luyện tập hội thoại AI sinh câu trả lời tự nhiên bám sát theo chủ đề.
- Hệ thống hiển thị các thống kê như: tổng XP, số kỹ năng đã hoàn thành, kỹ năng còn yếu, biểu đồ XP theo ngày/tuần.
- Tham gia bảng xếp hạng và nhận thông báo: Xem bảng xếp hạng theo bạn bè, theo ngôn ngữ, theo tuần hoặc tháng. Nhận thông báo về bài tập mới, nhắc nhở học, điểm số, huy hiệu mới, lời mời kết bạn sự kiện trong hệ thống.

3.2.2. Phía quản trị viên

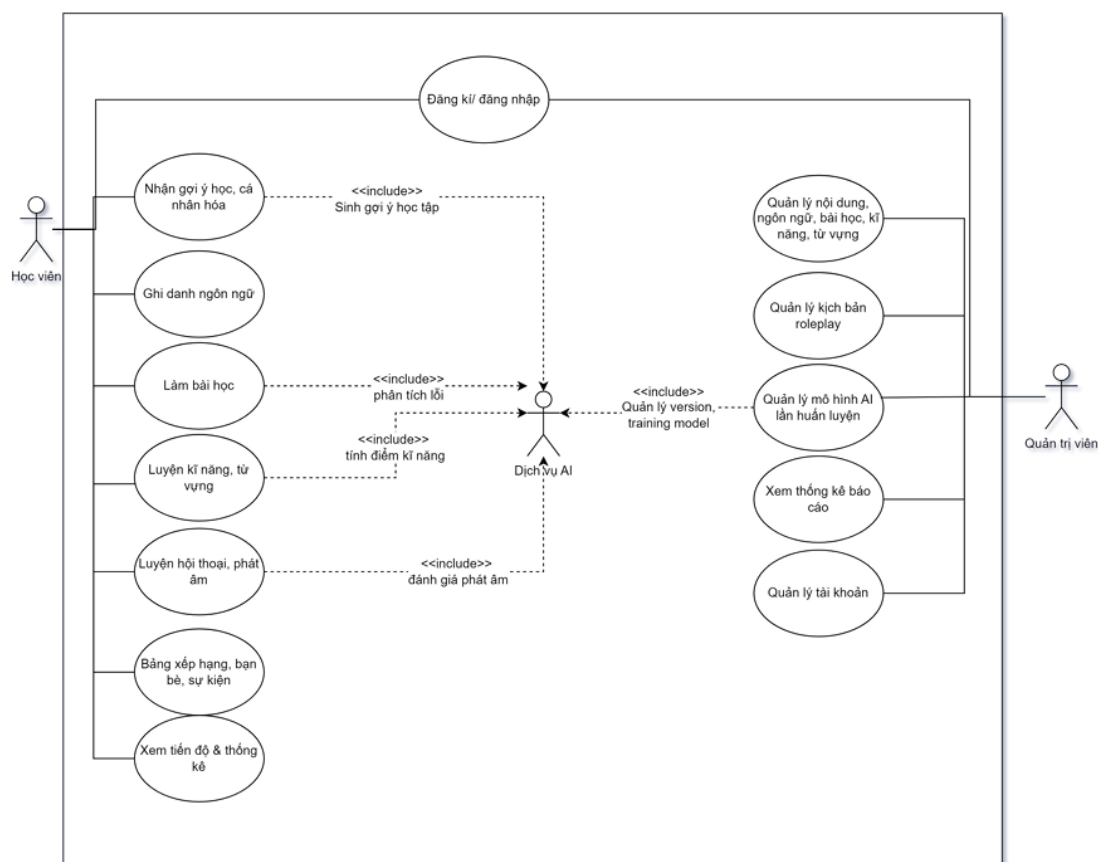
- Quản lý người dùng hệ thống: cho phép thực hiện đầy đủ các nghiệp vụ quản lý và vận hành nền tảng. Admin có thể quản lý người dùng thông qua việc xem danh sách tài khoản, tìm kiếm theo email, tên hoặc trạng thái hoạt động, đồng thời có quyền khóa hoặc mở khóa tài khoản trong trường hợp người dùng vi phạm chính sách sử dụng hệ thống. Bên cạnh đó, hệ thống hỗ

trợ chức năng phân quyền linh, bảo đảm mỗi nhóm người dùng chỉ được phép sử dụng các chức năng phù hợp với vai trò của mình.

- Quản lý nội dung học tập: admin có thể thực hiện đầy đủ các thao tác CRUD, bao gồm tạo mới, xem chi tiết, cập nhật và xóa các đối tượng như ngôn ngữ, chủ đề, bài học, kỹ năng, câu hỏi, câu trả lời, từ vựng, ví dụ minh họa, kịch bản hội thoại và các tệp âm thanh mẫu. Mỗi nội dung có thể được gắn cấp độ khó (A1–C2), gắn tag và các thông tin metadata như ngữ cảnh, tình huống sử dụng, giúp việc tìm kiếm, phân loại và tái sử dụng nội dung trở nên dễ dàng và có hệ thống. Ở mức kỹ thuật, quản trị viên còn có thể cấu hình và cập nhật các mô hình trí tuệ nhân tạo như STT, TTS, mô hình gợi ý hay mô hình chấm điểm (lựa chọn phiên bản, điều chỉnh một số tham số cơ bản được cho phép), đồng thời giám sát trạng thái hoạt động của các dịch vụ AI (online/offline), theo dõi log lỗi và hiệu năng để kịp thời xử lý sự cố.
- Giám sát hoạt động học tập: Xem thống kê tổng thể, số người học, số phiên học, số bài hoàn thành, tỷ lệ sử dụng các chức năng AI (luyện phát âm, roleplay, ôn từ,...). Theo dõi các chỉ số quan trọng hỗ trợ điều chỉnh nội dung, cấu hình hoặc mở rộng tài nguyên (máy chủ, dịch vụ AI).
- Quản lý hệ thống gamification và thông báo: Cấu hình các loại huy hiệu (badge), điều kiện đạt, cơ chế cộng XP, quy tắc streak. Cấu hình mẫu thông báo, lịch gửi thông báo tự động (nhắc học, nhắc ôn từ, sự kiện đặc biệt).

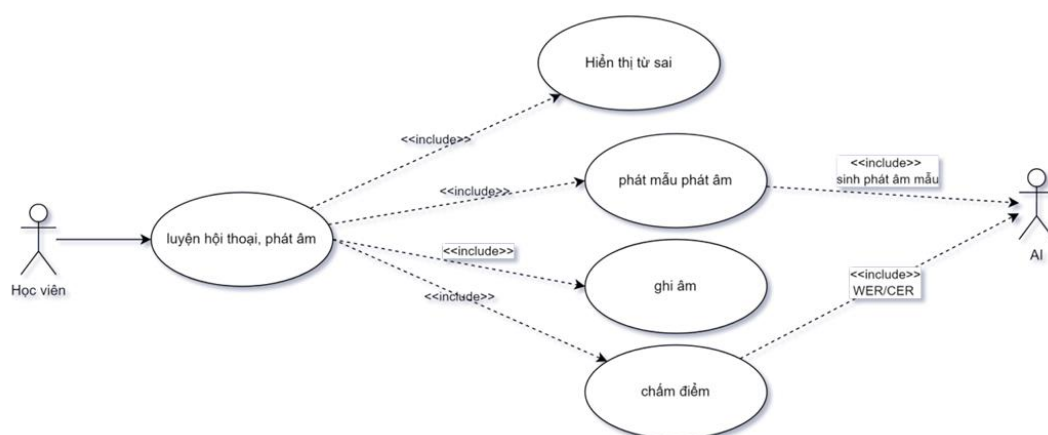
3.3 Mô hình hóa yêu cầu

3.3.1 Sơ đồ usecase tổng quát



Hình 3. 1: Sơ đồ usecase tổng quát

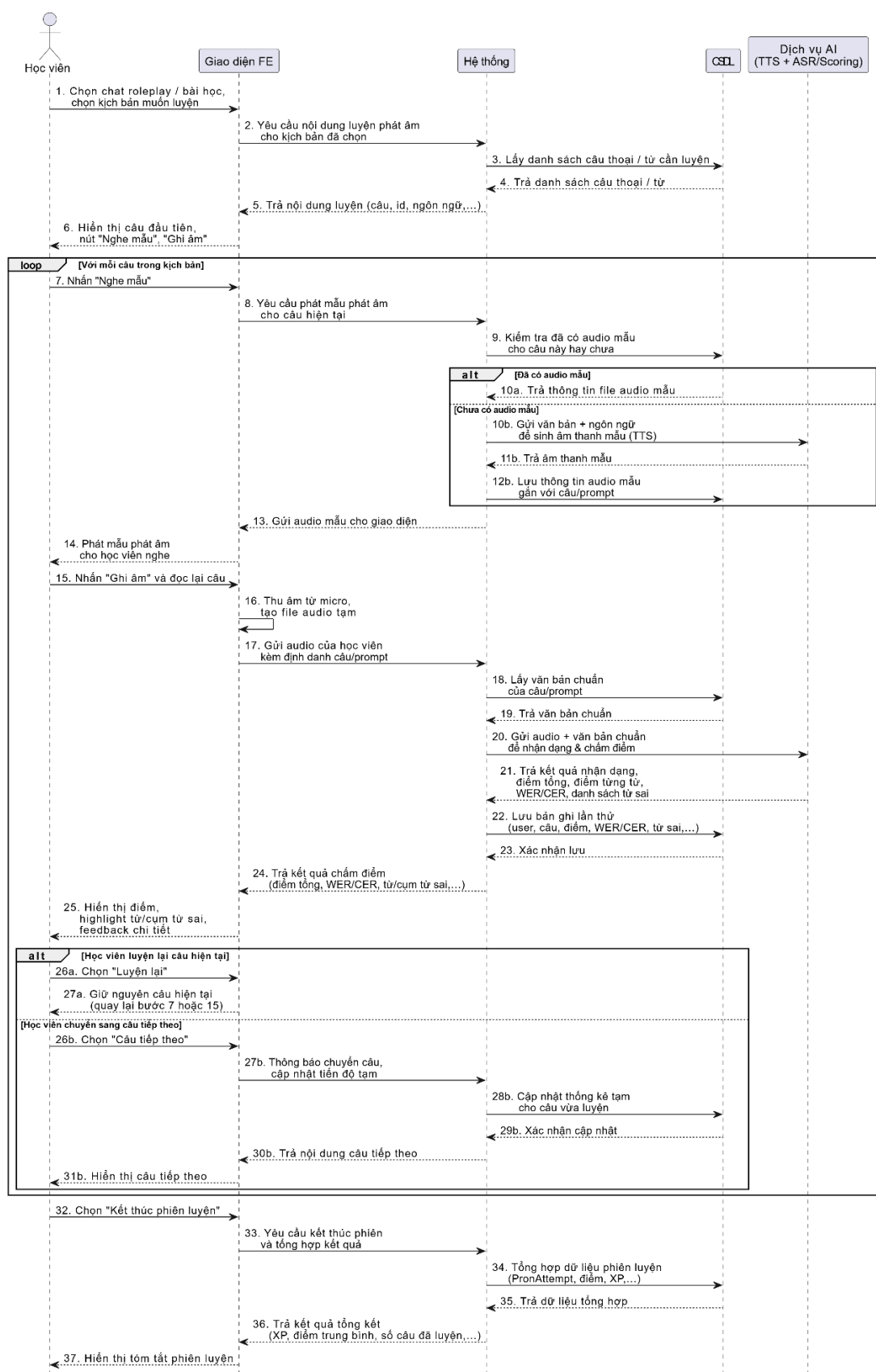
3.3.2 Chức năng luyện hội thoại, phát âm



Hình 3. 2: Sơ đồ usecase chức năng hội thoại

Bảng 3. 1: Mô tả usecase luyện hội thoại

Tên usecase	Hội thoại
Tác nhân	Học viên
Điều kiện đầu vào	Học viên đã đăng nhập hệ thống. Học viên đã chọn ngôn ngữ và bài học/kịch bản có hỗ trợ luyện hội thoại, phát âm. Thiết bị có micro hoạt động, trình duyệt/ứng dụng đã cấp quyền ghi âm. Dịch vụ AI đang sẵn sàng.
Kết quả đầu ra	Hệ thống hiển thị được các từ/cụm từ sai, điểm phát âm tổng thể và gợi ý chỉnh sửa cho học viên. Cập nhật thống kê kỹ năng nói/phát âm và XP
Mô tả	Chức năng cho phép học viên luyện nói theo câu thoại hoặc từ/cụm từ cho trước. Hệ thống phát mẫu phát âm bằng AI, học viên ghi âm lại, AI chấm điểm chất lượng phát âm và hiển thị các từ phát âm sai để học viên sửa và luyện lại.
Luồng xử lý chính	1. Học viên mở chat roleplay hoặc học trong bài học. Chọn kịch bản muốn luyện. 2. Hệ thống tải nội dung cần luyện. Lấy danh sách câu thoại hoặc từ/cụm từ tương ứng với ngữ cảnh đã chọn. 3. Phát mẫu phát âm. Hệ thống gửi văn bản câu/từ hiện tại sang dịch vụ AI TTS. AI sinh audio mẫu và hệ thống phát cho học viên nghe. 4. Ghi âm phát âm của học viên. Học viên bấm nút “Ghi âm” và đọc lại câu/từ theo mẫu. Hệ thống thu âm và lưu tạm file audio. 5. Hệ thống gửi audio của học viên cùng câu chuẩn sang AI để chấm điểm. Hệ thống nhận kết quả và lưu thành một bản ghi lần thử. 6. Hiển thị từ sai, điểm tổng, highlight các từ/cụm từ được phát hiện là sai hoặc chưa chuẩn. 7. Lặp lại hoặc chuyển sang câu tiếp theo. Học viên luyện lại câu hiện tại (quay lại bước 3–6), hoặc chuyển sang câu/cụm từ tiếp theo trong kịch bản hội thoại. 8. Kết thúc phiên luyện



Hình 3. 3: Biểu đồ tuần tự chức năng hội thoại

Khi học viên thực hiện chức năng hội thoại thì các bước xử lý chi tiết diễn ra như sau:

(1) Học viên mở chức năng chat roleplay hoặc bài học có tích hợp luyện nói và chọn kịch bản/câu thoại muốn luyện.

(2-6) Giao diện gửi yêu cầu lên hệ thống, hệ thống truy vấn CSDL, lấy danh sách câu thoại hoặc từ/cụm từ theo kịch bản đã chọn và trả về cho giao diện để hiển thị câu đầu tiên cùng nút “Nghe mẫu”, “Ghi âm”.

(7-14) Khi học viên nhấn “Nghe mẫu”, hệ thống kiểm tra có sẵn audio mẫu hay chưa, nếu chưa thì gọi AI TTS sinh audio, lưu cache và gửi audio mẫu về FE để phát cho học viên nghe.

(15-17) Học viên nhấn “Ghi âm” và đọc lại câu. Giao diện thu âm từ micro, tạo file audio tạm và gửi kèm định danh câu/prompt lên hệ thống để chấm điểm.

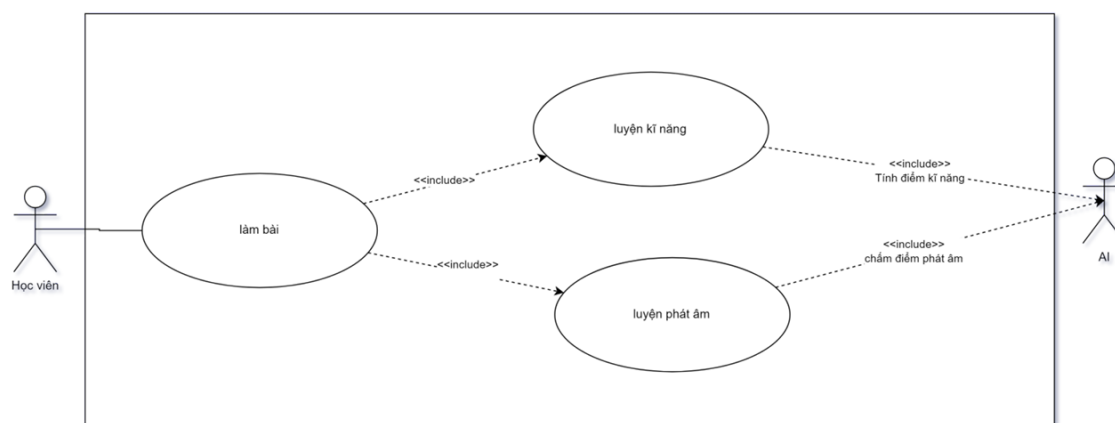
(18-23) Hệ thống lấy câu chuẩn từ CSDL, gửi audio + text chuẩn sang AI để nhận dạng và chấm điểm. AI trả về điểm tổng, điểm từng từ/âm, WER/CER và danh sách từ sai, hệ thống lưu lại bản ghi lần thử vào CSDL.

(24-25) Hệ thống trả kết quả chấm điểm về giao diện, giao diện hiển thị điểm tổng, các chỉ số đánh giá và highlight những từ/cụm từ phát âm sai hoặc chưa chuẩn để học viên quan sát và sửa lỗi.

(26a–31b) Học viên sẽ phải “Luyện lại” nếu phát âm chưa đạt được mục tiêu của câu. Khi chuyển câu, hệ thống cập nhật thống kê tạm và tải nội dung câu tiếp theo cho giao diện hiển thị, quy trình lặp lại từ bước 3–6 cho tới khi học viên dừng.

(32-37) Khi học viên chọn “Kết thúc phiên luyện”, hệ thống tổng hợp dữ liệu toàn bộ phiên (các lần thử, điểm, số câu đã luyện, XP...), cập nhật tiến độ/kỹ năng phát âm và gửi kết quả tổng kết để giao diện hiển thị màn hình tóm tắt phiên luyện.

3.3.3 Chức năng làm bài

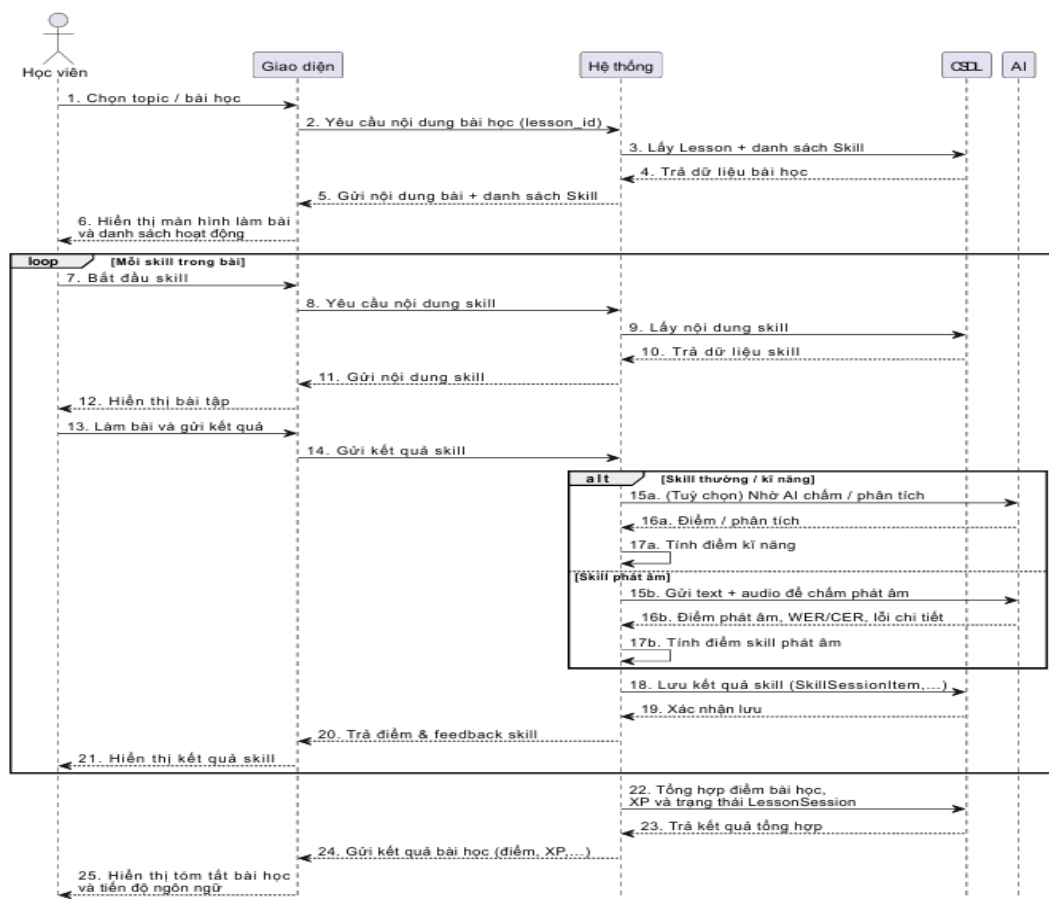


Hình 3. 4: Sơ đồ usecase chức năng làm bài

Bảng 3. 2: Mô tả usecase chức năng làm bài

Tên usecase	Làm bài
Tác nhân	Học viên
Điều kiện đầu vào	- Tài khoản đã đăng ký/đăng nhập. - Học viên đã ghi danh vào ngôn ngữ tương ứng. - Bài học được chọn
Kết quả đầu ra	- Kết quả hoàn thành bài học: điểm từng skill, điểm tổng, XP. - Các bản ghi của bài học được lưu trong CSDL (trạng thái hoàn thành, thời gian, điểm).
Mô tả	Học viên chọn một bài học thuộc chủ đề/ngôn ngữ đã ghi danh. Hệ thống tải nội dung bài, dẫn dắt học viên lần lượt thực hiện các skill trong bài (skill kĩ năng và skill phát âm), chấm điểm từng skill và cuối cùng tổng hợp điểm, XP và cập nhật tiến độ học tập.
Luồng xử lý chính	1. Học viên chọn chủ đề ngôn ngữ và chọn bài học muốn làm. 2. Hệ thống tải nội dung bài học và danh sách skill từ CSDL, hiển thị giao diện làm bài. 3. Học viên lần lượt làm các skill trong bài với skill thường, hệ thống chấm điểm kĩ năng, với skill phát âm.

	4. Sau mỗi skill, hệ thống lưu kết quả. 5. Khi hoàn thành toàn bộ skill, hệ thống tổng hợp điểm, tính XP, cập nhật và hiển thị tóm tắt kết quả bài học.
--	---



Hình 3. 5: Sơ đồ tuần tự chức năng làm bài

Khi học viên thực hiện chức năng làm bài thì các bước xử lý chi tiết diễn ra như sau:

(1–2) Học viên chọn chủ đề/ngôn ngữ và bài học muốn làm trên giao diện, giao diện gửi yêu cầu lên hệ thống để lấy nội dung bài học tương ứng với bài đã chọn.

(3–6) Hệ thống truy vấn CSDL, lấy thông tin bài học và danh sách các skill trong bài, trả dữ liệu về cho giao diện để hiển thị màn hình làm bài và danh sách skill/hoạt động cho học viên.

(7–14) Với mỗi skill, học viên bấm bắt đầu skill, giao diện yêu cầu hệ thống tải nội dung chi tiết của skill, hiển thị bài tập (câu hỏi, dạng bài, giao diện ghi âm nếu là phát

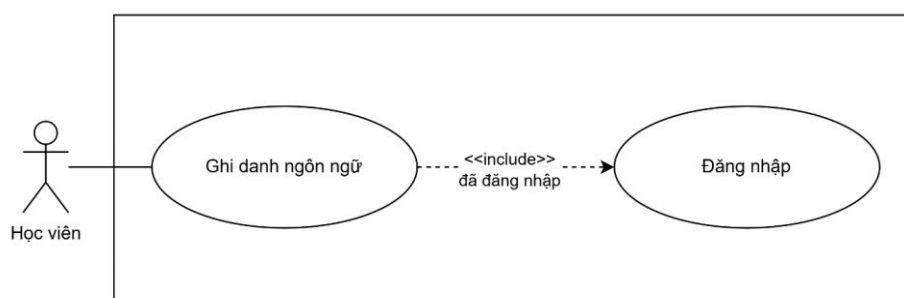
âm), sau đó học viên thực hiện bài tập và gửi kết quả (đáp án, lựa chọn, audio...) lên hệ thống.

(15a–17b) Hệ thống chấm điểm skill: với skill thường, hệ thống (có thể) gọi AI hỗ trợ phân tích rồi tính điểm kỹ năng; với skill phát âm, hệ thống gửi text + audio sang AI để chấm phát âm (tính điểm tổng, WER/CER, lỗi phát âm) và quy đổi thành điểm cho skill phát âm.

(18–21) Sau mỗi skill, hệ thống lưu kết quả skill (điểm, thời gian, thông tin chi tiết) vào CSDL, trả điểm và feedback về cho giao diện, giao diện hiển thị kết quả skill và cho phép học viên chuyển sang skill tiếp theo, vòng này lặp lại cho tới khi hoàn thành toàn bộ skill trong bài.

(22–25) Khi học viên hoàn thành tất cả các skill, hệ thống tổng hợp điểm bài học, tính XP, cập nhật LessonSession/tiến độ học tập và gửi kết quả tổng kết để giao diện hiển thị màn hình tóm tắt kết quả bài học (điểm, XP, trạng thái hoàn thành).

3.3.4 Chức năng ghi danh ngôn ngữ

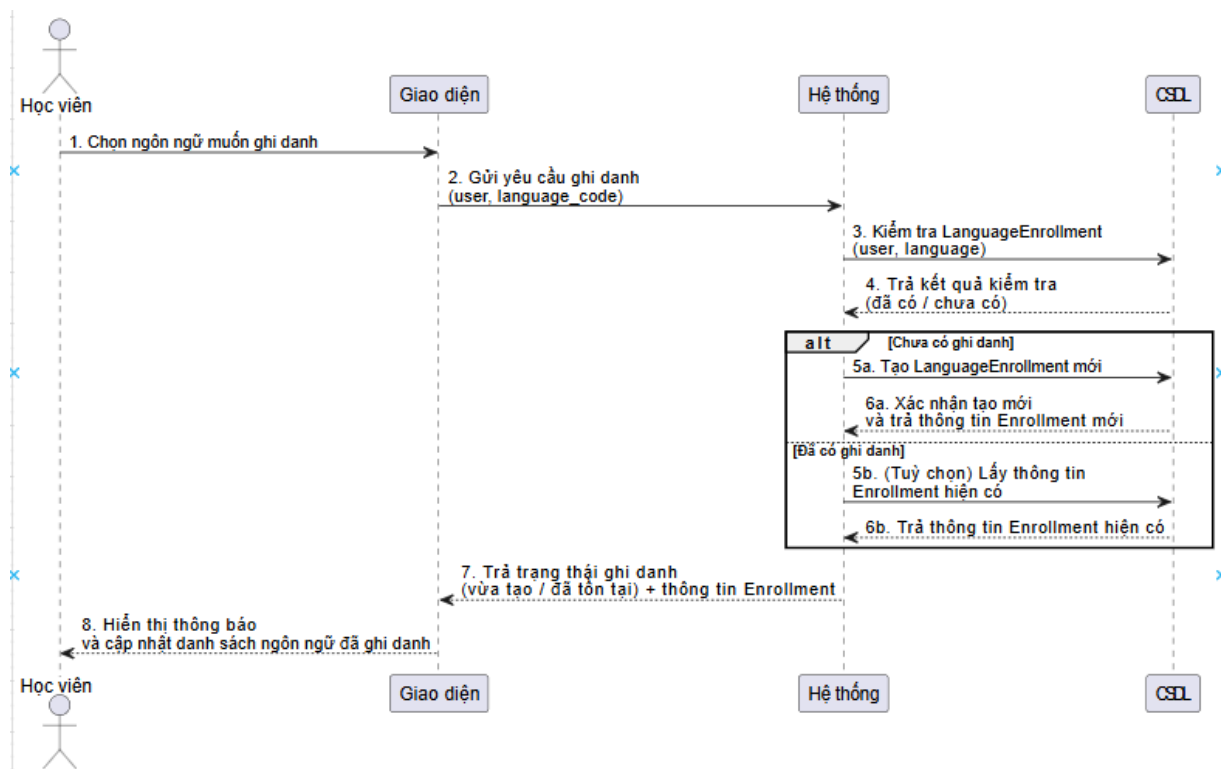


Hình 3. 6: usecase chức năng ghi danh ngôn ngữ

Bảng 3. 3: Mô tả usecase chức năng ghi danh ngôn ngữ

Tên usecase	Ghi danh ngôn ngữ
Tác nhân	Học viên
Điều kiện đầu vào	- Tài khoản đã đăng nhập. - Ngôn ngữ muốn ghi danh.
Kết quả đầu ra	Bản ghi mới trạng thái ghi danh (thành công, đã tồn tại, lỗi).
Mô tả	Học viên chọn một ngôn ngữ để thêm vào danh sách ngôn ngữ đang học. Hệ thống kiểm tra trong CSDL: nếu học viên chưa ghi danh ngôn ngữ đó thì tạo mới một bản ghi, nếu đã ghi danh

	rồi thì không tạo mới, không cập nhật dữ liệu, chỉ trả về thông tin rằng ngôn ngữ này đã được đăng kí.
Luồng xử lý chính	1. Học viên chọn ngôn ngữ muốn ghi danh. 2. Hệ thống kiểm tra trong CSDL xem đã tồn tại bản đăng kí ngôn ngữ cho (học viên, ngôn ngữ) hay chưa. 3. Nếu chưa có, hệ thống tạo mới bản ghi. Nếu đã có, hệ thống giữ nguyên dữ liệu hiện có. 4. Hệ thống trả về cho giao diện trạng thái ghi danh để hiển thị cho học viên.



Hình 3. 7: Sơ đồ tuần tự chức năng ghi danh ngôn ngữ

Khi học viên thực hiện chức năng ghi danh ngôn ngữ thì các bước xử lý chi tiết diễn ra như sau:

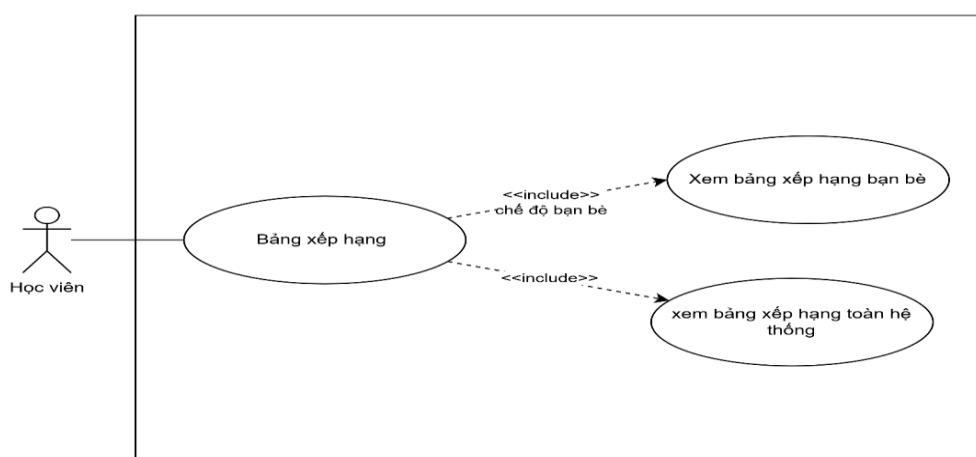
(1) Học viên mở màn hình quản lý ngôn ngữ và chọn ngôn ngữ muốn ghi danh (ví dụ tiếng Anh, tiếng Nhật).

(2–4) Giao diện gửi yêu cầu ghi danh lên hệ thống kèm theo thông tin tài khoản và mã ngôn ngữ. Hệ thống truy vấn CSDL để kiểm tra xem đã tồn tại bản ghi cho cặp (học viên, ngôn ngữ) hay chưa và nhận kết quả kiểm tra.

(5a–6a / 5b–6b) Nếu chưa có ghi danh, hệ thống tạo mới bản ghi trong CSDL và nhận xác nhận tạo mới. nếu đã có ghi danh thì hệ thống không tạo mới, không chỉnh sửa dữ liệu, chỉ lấy lại thông tin hiện có.

(7–8) Sau khi xử lý xong, hệ thống trả trạng thái ghi danh (tạo mới thành công hoặc đã được đăng kí từ trước) cùng thông tin ngôn ngữ đã đăng kí về cho giao diện, giao diện hiển thị thông báo phù hợp và cập nhật danh sách ngôn ngữ đã ghi danh của học viên.

3.3.5 Chức năng bảng xếp hạng

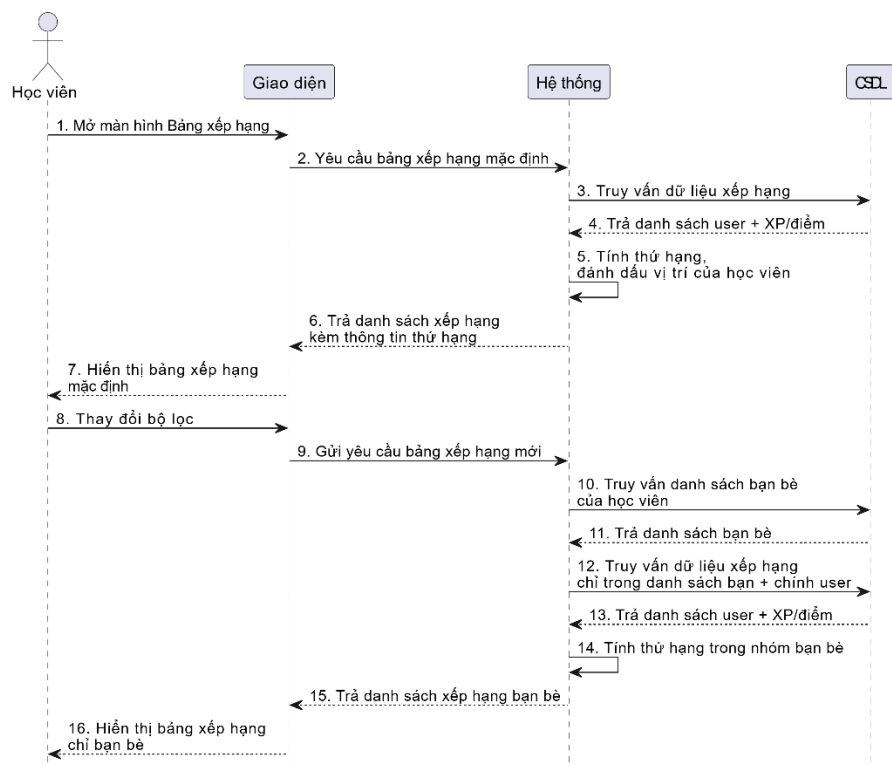


Hình 3. 8: usecase chức năng bảng xếp hạng

Bảng 3. 4: Mô tả usecase chức năng bảng xếp hạng

Tên usecase	Bảng xếp hạng
Tác nhân	Học viên
Điều kiện đầu vào	- Tài khoản đã đăng nhập. - Thông tin ngôn ngữ và phạm vi - Kiểu xếp hạng (toàn hệ thống / bạn bè), khoảng thời gian (ngày/tuần/tháng), ngôn ngữ hoặc toàn bộ ngôn ngữ.
Kết quả đầu ra	Danh sách bảng xếp hạng (thông tin user, điểm/XP, vị trí xếp hạng), vị trí hiện tại của học viên, dữ liệu được hiển thị trên giao diện.

Mô tả	Học viên mở chức năng bảng xếp hạng để xem vị trí của mình so với những người dùng khác. Hệ thống có thể hiển thị xếp hạng toàn hệ thống hoặc xếp hạng trong phạm vi bạn bè và cho phép lọc theo khoảng thời gian, ngôn ngữ.
Luồng xử lý chính	1. Học viên mở màn hình bảng xếp hạng. 2. Hệ thống tải bảng xếp hạng mặc định và hiển thị cho học viên. 3. Học viên có thể thay đổi bộ lọc. 4. Hệ thống truy vấn lại dữ liệu theo điều kiện mới, sắp xếp và trả về danh sách xếp hạng đã lọc. 5. Giao diện cập nhật bảng xếp hạng và highlight vị trí hiện tại của học viên.



Hình 3. 9: Sơ đồ tuần tự chức năng bảng xếp hạng

Khi học viên thực hiện chức năng bảng xếp hạng thì các bước xử lý chi tiết diễn ra như sau:

(1) Học viên mở màn hình Bảng xếp hạng trên giao diện.

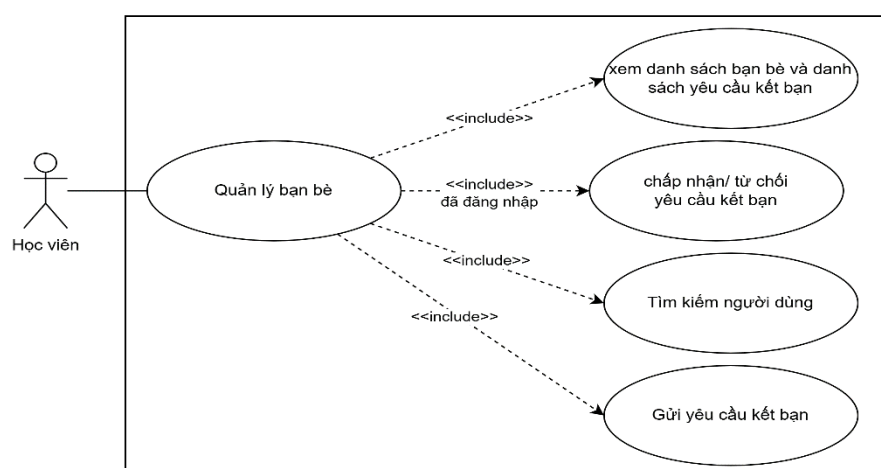
(2–7) Giao diện gửi yêu cầu lên hệ thống để lấy bảng xếp hạng mặc định. Hệ thống truy vấn CSDL, lấy danh sách người dùng và XP/điểm tương ứng, tính toán thứ hạng và đánh dấu vị trí của học viên, sau đó trả danh sách xếp hạng về cho giao diện để hiển thị.

(8–9) Khi học viên thay đổi bộ lọc giao diện gửi yêu cầu mới kèm tham số bộ lọc cho hệ thống.

(10–13) Hệ thống truy vấn CSDL để lấy danh sách bạn bè của học viên, sau đó truy vấn dữ liệu xếp hạng nhưng chỉ trong tập người dùng là bạn bè + chính học viên, rồi nhận lại danh sách user và XP/điểm tương ứng.

(14–16) Hệ thống tính lại thứ hạng trong phạm vi bạn bè, gửi danh sách xếp hạng mới về cho giao diện; giao diện cập nhật và hiển thị bảng xếp hạng theo bộ lọc.

3.3.6 Chức năng bạn bè

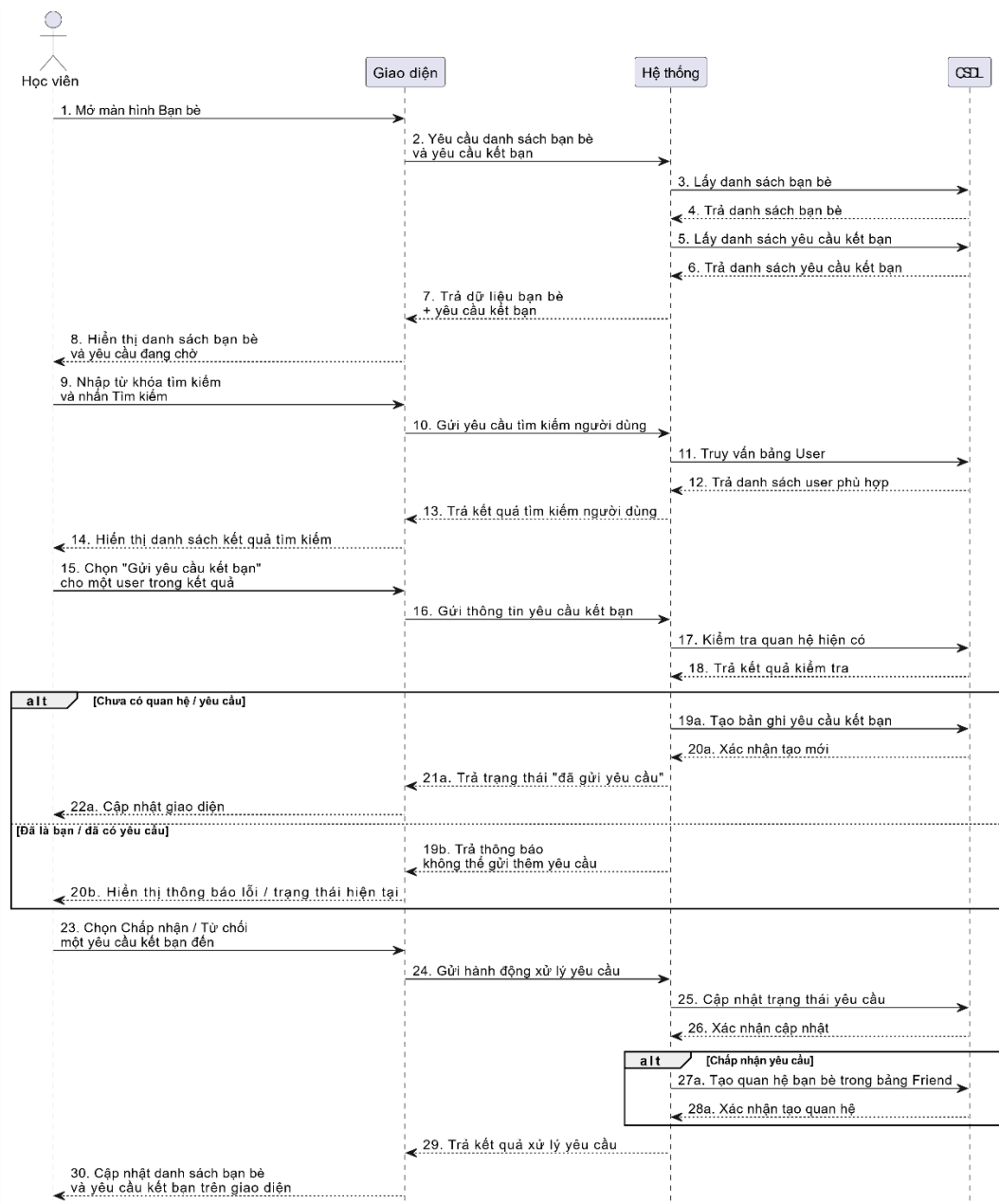


Hình 3. 10: usecase chức năng bạn bè

Bảng 3. 5: Mô tả usecase chức năng quản lý bạn bè

Tên usecase	Quản lý bạn bè
Tác nhân	Học viên
Điều kiện đầu vào	- Tài khoản đã đăng nhập. - Thông tin user mục tiêu khi gửi hay duyệt yêu cầu kết bạn và trạng thái hành động (gửi, chấp nhận, từ chối). - Khi tìm kiếm: từ khóa tìm kiếm theo email

Kết quả đầu ra	- Danh sách bạn bè hiện tại. - Danh sách yêu cầu kết bạn đến/đi và trạng thái tương ứng. - Kết quả tìm kiếm người dùng (danh sách user phù hợp). - Trạng thái đã gửi yêu cầu, không thể gửi, chấp nhận/từ chối.
Mô tả	Chức năng cho phép học viên quản lý quan hệ bạn bè trong hệ thống: xem danh sách bạn bè và các yêu cầu kết bạn đang chờ; tìm kiếm người dùng khác để kết bạn; gửi yêu cầu kết bạn; chấp nhận hoặc từ chối yêu cầu kết bạn đến. Hệ thống lưu trữ và cập nhật các quan hệ bạn bè, đồng thời cung cấp dữ liệu để các chức năng khác như bảng xếp hạng theo bạn bè sử dụng.
Luồng xử lý chính	1. Học viên mở màn hình bạn bè để xem danh sách bạn bè và các yêu cầu kết bạn đang chờ. 2. Hệ thống lấy danh sách bạn bè và các yêu cầu kết bạn rồi hiển thị cho học viên. 3. Khi học viên muốn kết bạn với người mới, học viên sử dụng chức năng tìm kiếm để tìm người dùng theo từ khóa và trả danh sách kết quả phù hợp. 4. Từ kết quả tìm kiếm, học viên chọn người cần kết bạn, hệ thống kiểm tra quan hệ hiện tại, nếu chưa có quan hệ/yêu cầu thì tạo bản ghi yêu cầu kết bạn, nếu đã có thì trả thông báo tương ứng. 5. Khi học viên chấp nhận hoặc từ chối một yêu cầu kết bạn đến, hệ thống cập nhật, nếu chấp nhận thì tạo thêm bản ghi quan hệ bạn bè, sau đó cập nhật danh sách bạn bè và danh sách yêu cầu kết bạn trên giao diện.



Hình 3. 11: sơ đồ tuần tự chức năng quản lý bạn bè

Khi học viên thực hiện chức năng bạn bè thì các bước xử lý chi tiết diễn ra như sau:

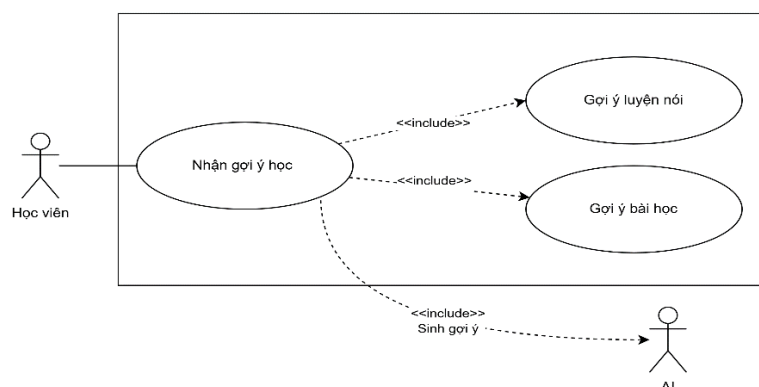
(1–8) Học viên mở màn hình Bạn bè trên giao diện để quản lý quan hệ bạn bè. Giao diện gửi yêu cầu lên hệ thống để lấy danh sách bạn bè hiện tại và các yêu cầu kết bạn đang chờ. Hệ thống truy vấn CSDL hai lần: lần đầu lấy các quan hệ đã được chấp nhận (danh sách bạn bè), lần sau lấy các yêu cầu kết bạn ở trạng thái đang chờ (incoming và outgoing). Sau khi tổng hợp dữ liệu, hệ thống trả về cho giao diện, giao diện hiển thị danh sách bạn bè cùng các yêu cầu kết bạn đang chờ xử lý.

(9–14) Khi học viên muốn kết bạn với người mới, học viên nhập từ khóa (tên, email, username...) vào ô tìm kiếm và nhấn Tìm kiếm. Giao diện gửi yêu cầu tìm kiếm người dùng lên hệ thống. Hệ thống sử dụng từ khóa này để truy vấn bảng User trong CSDL theo các điều kiện phù hợp (ví dụ tìm theo tên hiển thị, username hoặc email) và nhận lại danh sách người dùng phù hợp. Danh sách kết quả tìm kiếm được gửi về cho giao diện, giao diện hiển thị danh sách người dùng tìm được, kèm theo nút “Gửi yêu cầu kết bạn” cho từng người.

(15–22) Từ danh sách kết quả tìm kiếm (hoặc các màn hình khác như bảng xếp hạng/hồ sơ), khi học viên chọn “Gửi yêu cầu kết bạn” tới một người dùng, giao diện gửi thông tin yêu cầu kết bạn lên hệ thống. Hệ thống kiểm tra trong CSDL xem giữa hai tài khoản đã là bạn, đã có yêu cầu kết bạn pending hoặc đã từng từ chối trước đó hay chưa. Nếu chưa có quan hệ hoặc yêu cầu đang chờ, hệ thống tạo mới một bản ghi yêu cầu kết bạn ở trạng thái pending và CSDL xác nhận lưu; hệ thống trả về trạng thái “đã gửi yêu cầu” để giao diện cập nhật nút hiển thị (ví dụ chuyển thành “Đã gửi”). Nếu đã tồn tại quan hệ bạn bè hoặc yêu cầu kết bạn phù hợp, hệ thống không tạo mới, mà trả thông báo không thể gửi thêm yêu cầu; giao diện hiển thị thông báo hoặc giữ nguyên trạng thái nút để học viên biết.

(23–30) Khi học viên xử lý một yêu cầu kết bạn đến (chọn Chấp nhận hoặc Từ chối), giao diện gửi hành động tương ứng (accept/reject) cùng mã yêu cầu lên hệ thống. Hệ thống cập nhật trạng thái bản ghi yêu cầu kết bạn trong CSDL (accepted hoặc rejected) và nhận xác nhận. Nếu hành động là chấp nhận, hệ thống tạo thêm bản ghi quan hệ bạn bè (accepted) giữa hai tài khoản trong bảng Friend, CSDL xác nhận lưu. Sau khi xử lý xong, hệ thống trả kết quả về cho giao diện; giao diện cập nhật lại danh sách bạn bè (thêm bạn mới nếu chấp nhận) và loại bỏ yêu cầu đã xử lý khỏi danh sách yêu cầu kết bạn đang chờ, đồng thời hiển thị trạng thái mới cho học viên.

3.3.7 Chức năng nhận gợi ý học

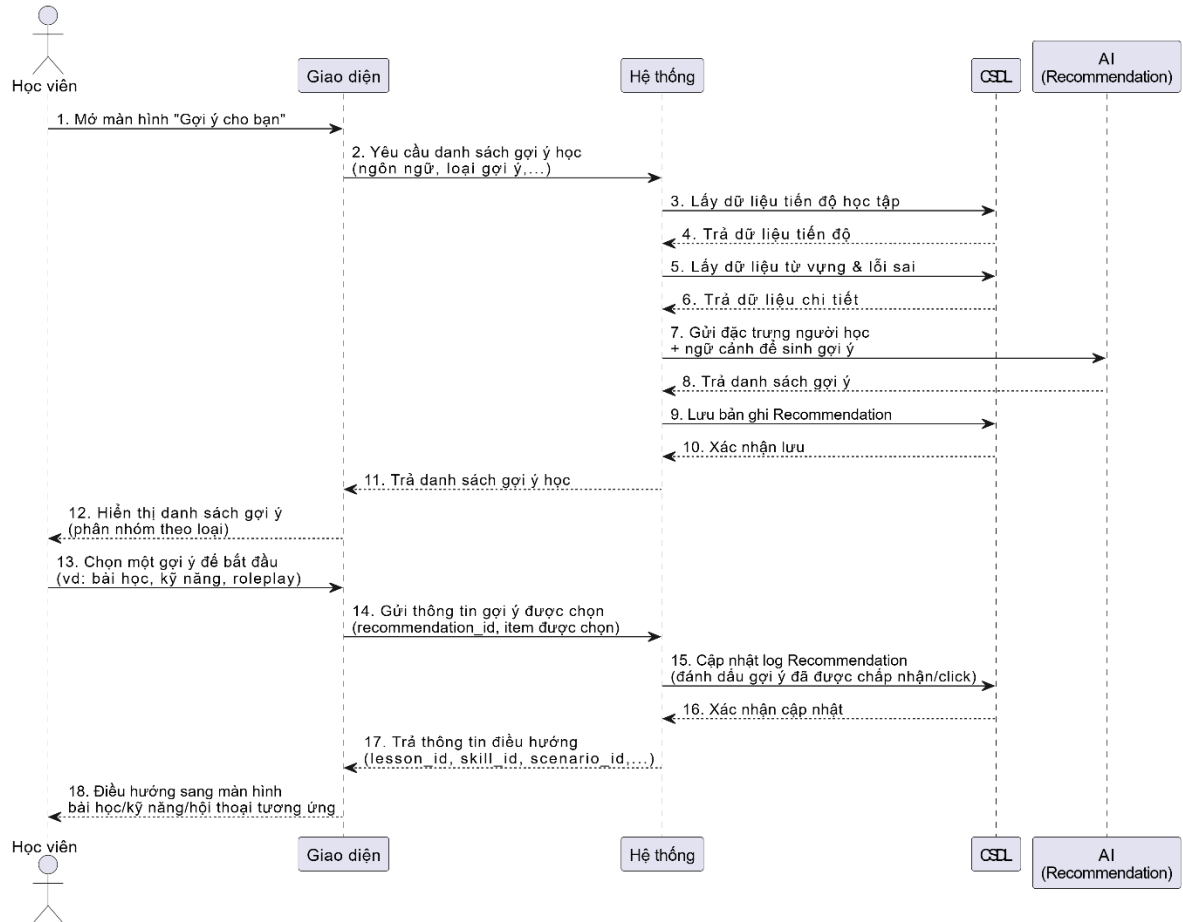


Hình 3. 12: usecase chức năng nhận gợi ý học

Bảng 3. 6: Mô tả usecase chức năng gợi ý

Tên usecase	Nhận gợi ý học
Tác nhân	Học viên
Điều kiện đầu vào	Tài khoản đã đăng nhập ngữ cảnh hiện tại (ngôn ngữ đang học, cấp độ, bài học hiện tại). tùy chọn loại gợi ý
Kết quả đầu ra	Danh sách gợi ý học cá nhân hóa
Mô tả	Học viên mở chức năng gợi ý học để xem những đề xuất bài học, kỹ năng, từ vựng hoặc hoạt động luyện nói phù hợp với trình độ và lịch sử học tập của mình. Hệ thống sử dụng dữ liệu tiến độ, lỗi sai và tương tác học tập làm đầu vào cho mô hình AI Recommendation, sau đó trả lại danh sách gợi ý ưu tiên cao nhất để học viên lựa chọn.
Luồng xử lý chính	1. Học viên mở màn hình “Gợi ý cho bạn” hoặc khu vực gợi ý trên trang chính. 2. Hệ thống thu thập dữ liệu tiến độ học của học viên từ CSDL. 3. Hệ thống gửi đặc trưng người học sang dịch vụ AI Recommendation và nhận lại danh sách các mục gợi ý 4. Hệ thống lưu lại các gợi ý được sinh và trả danh sách gợi ý về cho giao diện.

5. Giao diện hiển thị các gợi ý, khi học viên chọn một gợi ý, hệ thống ghi nhận việc “chấp nhận gợi ý” và điều hướng học viên tới bài học/kỹ năng/tính năng tương ứng.



Hình 3. 13: Sơ đồ tuần tự chức năng gợi ý học

Khi học viên thực hiện chức năng nhận gợi ý học cá nhân hóa thì các bước xử lý chi tiết diễn ra như sau:

(1–2) Học viên mở màn hình “Gợi ý cho bạn” hoặc khu vực gợi ý trên giao diện; giao diện gửi yêu cầu lên hệ thống để lấy danh sách gợi ý học phù hợp với ngôn ngữ/loại gợi ý hiện tại.

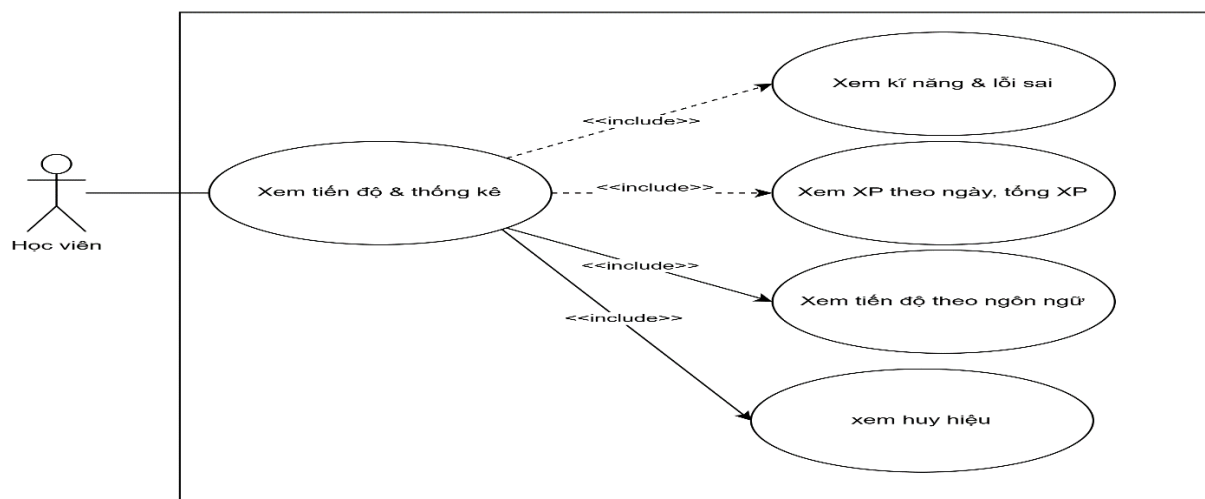
(3–6) Hệ thống truy vấn CSDL để thu thập dữ liệu tiến độ học của học viên, bao gồm các bản ghi ghi danh ngôn ngữ, các phiên bài học và kỹ năng, từ vựng đã biết, lỗi sai thường gặp và lịch sử tương tác học tập; dữ liệu này được tổng hợp thành đặc trưng đầu vào cho mô hình gợi ý.

(7–8) Hệ thống gửi đặc trưng người học và ngữ cảnh hiện tại sang dịch vụ AI Recommendation; dịch vụ AI xử lý và trả về danh sách gợi ý học, gồm các mục như bài học và kỹ năng nên học tiếp, từ vựng cần ôn lại, kịch bản hội thoại hoặc bài luyện phát âm phù hợp, kèm theo điểm ưu tiên.

(9–11) Hệ thống lưu lại bản ghi Recommendation (log) vào CSDL để phục vụ phân tích sau này, sau đó trả danh sách gợi ý học về cho giao diện. Giao diện hiển thị các gợi ý, có thể nhóm theo loại.

(13–18) Khi học viên chọn một gợi ý cụ thể giao diện gửi thông tin gợi ý được chọn lên hệ thống; hệ thống cập nhật log Recommendation để đánh dấu gợi ý đã được chấp nhận/click, sau đó trả về thông tin điều hướng. Giao diện điều hướng học viên sang màn hình học tương ứng để bắt đầu thực hiện gợi ý đó.

3.3.8 Chức năng xem tiến độ & thống kê

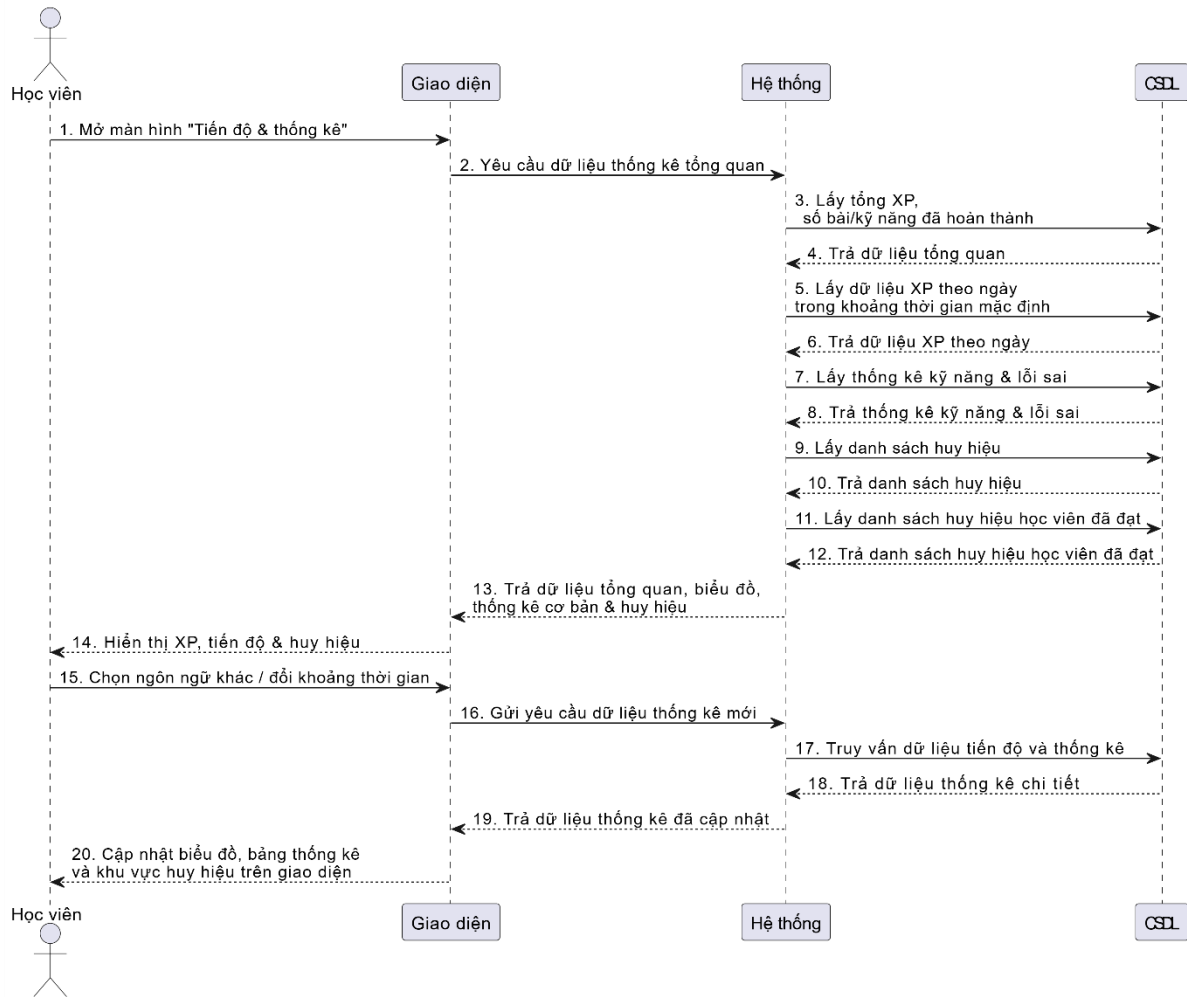


Hình 3. 14: usecase chức năng xem tiến độ & thống kê

Bảng 3. 7: Mô tả usecase xem tiến độ & thống kê

Tên usecase	Xem tiến độ & thống kê
Tác nhân	Học viên
Điều kiện đầu vào	Tài khoản đã đăng nhập, tùy chọn bộ lọc: ngôn ngữ, khoảng thời gian (ngày/tuần/tháng), loại thống kê (XP, bài đã hoàn thành, kỹ năng, lỗi sai, huy hiệu).
Kết quả đầu ra	Các chỉ số tiến độ học tập: tổng XP, số bài/kỹ năng đã hoàn thành, tiến độ theo từng ngôn ngữ, biểu đồ XP theo ngày, thống

	kê kỹ năng & lỗi sai, danh sách huy hiệu đã đạt và tiến độ đạt các huy hiệu tiếp theo.
Mô tả	Chức năng cho phép học viên xem bức tranh tổng quan về quá trình học: XP tích lũy, số bài/kỹ năng đã hoàn thành, tiến độ theo từng ngôn ngữ, biểu đồ XP theo thời gian và thống kê chi tiết về kỹ năng, lỗi sai. Đồng thời, học viên có thể xem các huy hiệu đã đạt được và điều kiện để đạt những huy hiệu tiếp theo.
Luồng xử lý chính	1. Học viên mở màn hình “Tiến độ & thống kê”. 2. Hệ thống lấy dữ liệu tổng quan dữ liệu XP theo ngày, thống kê kỹ năng & lỗi sai, cùng với danh sách huy hiệu định nghĩa và huy hiệu mà học viên đã đạt, sau đó hiển thị các chỉ số và danh sách huy hiệu. . 3. Hệ thống truy vấn và tổng hợp lại dữ liệu tương ứng với bộ lọc mới (bao gồm cập nhật tiến độ huy hiệu nếu hiển thị), rồi trả về để giao diện cập nhật biểu đồ, bảng thống kê và trạng thái huy hiệu.



Hình 3. 15: Sơ đồ tuần tự xem tiến độ mà thống kê

Khi học viên thực hiện chức năng xem tiến độ & thống kê thì các bước xử lý chi tiết diễn ra như sau:

(1–2) Học viên mở màn hình “Tiến độ & thống kê” trên giao diện; giao diện gửi yêu cầu lên hệ thống để lấy dữ liệu thống kê tổng quan theo ngôn ngữ và khoảng thời gian mặc định.

(3–8) Hệ thống truy vấn CSDL để lấy các chỉ số tổng quan như tổng XP, số bài học/kỹ năng đã hoàn thành, đồng thời lấy dữ liệu XP theo ngày trong khoảng thời gian mặc định và một số thống kê kỹ năng, lỗi sai cơ bản. Các dữ liệu này được tổng hợp để phục vụ phần biểu đồ và các thẻ chỉ số trong màn hình tiến độ.

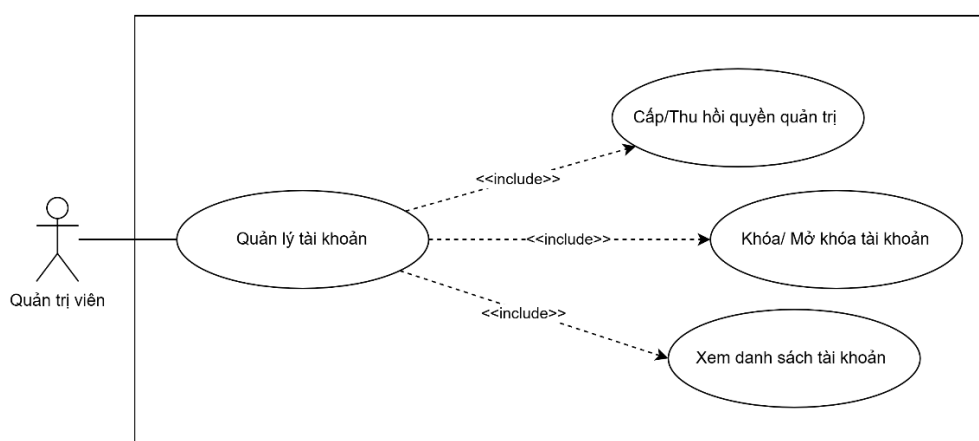
(9–12) Tiếp theo, hệ thống truy vấn bảng Badge để lấy danh sách tất cả huy hiệu đang được định nghĩa trong hệ thống và truy vấn bảng UserBadge để lấy danh sách huy hiệu mà học viên đã đạt. Từ hai nguồn dữ liệu này, hệ thống có thể tính toán thêm thông

tin hiển thị như: huy hiệu đã đạt, huy hiệu đang tiến triển và phần trăm tiến độ đạt các huy hiệu tiếp theo.

(13–14) Hệ thống trả về cho giao diện toàn bộ dữ liệu tổng quan, biểu đồ XP, thống kê kỹ năng/lỗi sai cơ bản cùng danh sách huy hiệu và trạng thái đạt/tiến độ. Giao diện hiển thị cho học viên các chỉ số XP, số bài/kỹ năng đã hoàn thành, biểu đồ XP theo ngày, đồng thời hiển thị khu vực huy hiệu với các huy hiệu đã đạt và điều kiện, tiến độ cho các huy hiệu chưa đạt.

(15–20) Khi học viên chọn ngôn ngữ khác, thay đổi khoảng thời gian hoặc chuyển tab thống kê, giao diện gửi yêu cầu mới kèm bộ lọc hiện tại lên hệ thống. Hệ thống truy vấn lại CSDL để lấy dữ liệu tiến độ và thống kê chi tiết tương ứng với bộ lọc/tab mới. Sau khi tổng hợp xong, hệ thống trả về dữ liệu đã cập nhật cho giao diện, giao diện cập nhật lại biểu đồ, bảng thống kê và khu vực huy hiệu để phản ánh chính xác tiến độ học tập và trạng thái huy hiệu của học viên theo lựa chọn hiện tại.

3.3.9 Quản lý tài khoản

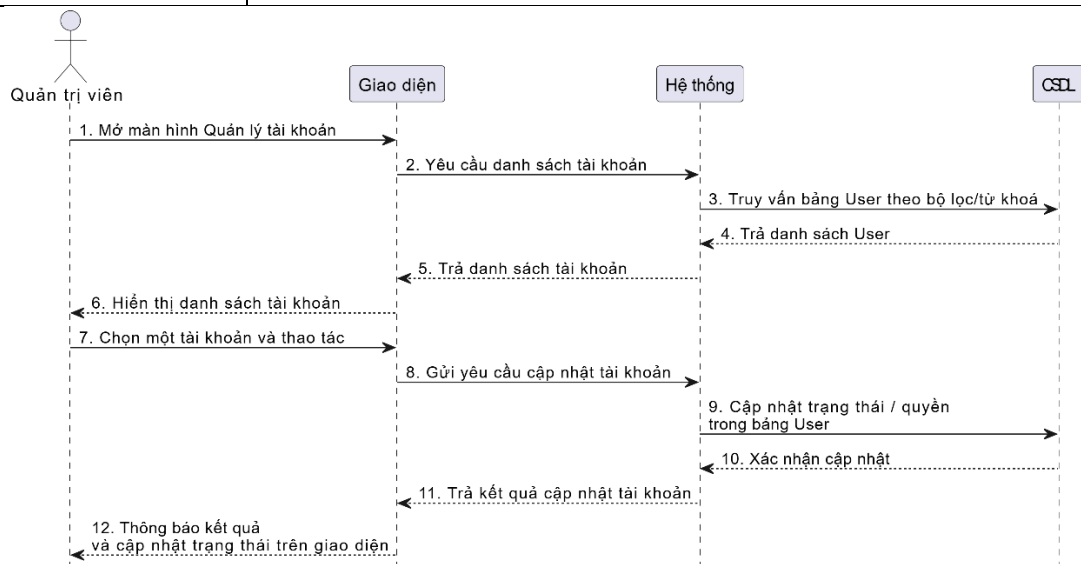


Hình 3. 16: Usecase chức năng quản lý tài khoản

Bảng 3. 8: Mô tả usecase quản lý tài khoản

Tên usecase	Quản lý tài khoản
Tác nhân	Quản trị viên
Điều kiện đầu vào	Tài khoản admin, bộ lọc tìm kiếm user thao tác quản trị.
Kết quả đầu ra	Danh sách tài khoản được hiển thị; trạng thái tài khoản được cập nhật, cờ quyền admin được cập nhật.

Mô tả	Quản trị viên dùng chức năng này để xem danh sách tài khoản, tìm kiếm người dùng, khoá/mở khoá tài khoản vi phạm, và cấp/thu hồi quyền quản trị cho những người dùng đủ thẩm quyền.
Luồng xử lý chính	1. Quản trị viên mở màn hình Quản lý tài khoản, hệ thống hiển thị danh sách tài khoản và cho phép tìm kiếm. 2. Quản trị viên chọn một tài khoản và thực hiện thao tác khoá/mở khoá hoặc cấp/thu hồi quyền admin. 3. Hệ thống cập nhật trạng thái hoặc quyền của tài khoản trong CSDL. 4. Giao diện hiển thị thông báo và cập nhật lại thông tin tài khoản tương ứng.



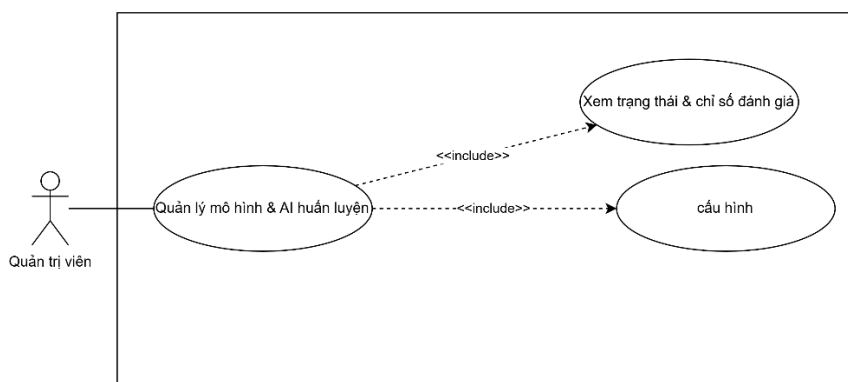
Hình 3. 17: Sơ đồ tuần tự quản lý tài khoản

Khi quản trị viên thực hiện chức năng quản lý tài khoản thì các bước xử lý chi tiết diễn ra như sau:

(1–6) Quản trị viên mở màn hình “Quản lý tài khoản” trên giao diện admin. Giao diện gửi yêu cầu lên hệ thống để lấy danh sách tài khoản theo bộ lọc mặc định (hoặc kèm từ khoá nếu có). Hệ thống truy vấn bảng User trong CSDL, nhận danh sách tài khoản phù hợp và trả về cho giao diện; giao diện hiển thị danh sách tài khoản để quản trị viên xem và lựa chọn.

(7–12) Khi quản trị viên chọn một tài khoản và thực hiện thao tác quản trị (Khoá/Mở khoá hoặc Cấp/Thu hồi quyền admin), giao diện gửi yêu cầu cập nhật tài khoản lên hệ thống. Hệ thống cập nhật trạng thái hoặc quyền của tài khoản tương ứng trong CSDL và nhận xác nhận; sau đó hệ thống trả kết quả cập nhật về cho giao diện, giao diện hiển thị thông báo kết quả và cập nhật lại trạng thái tài khoản (active/locked, có hoặc không có quyền admin) trên màn hình.

3.3.10 Quản lý mô hình và AI huấn luyện

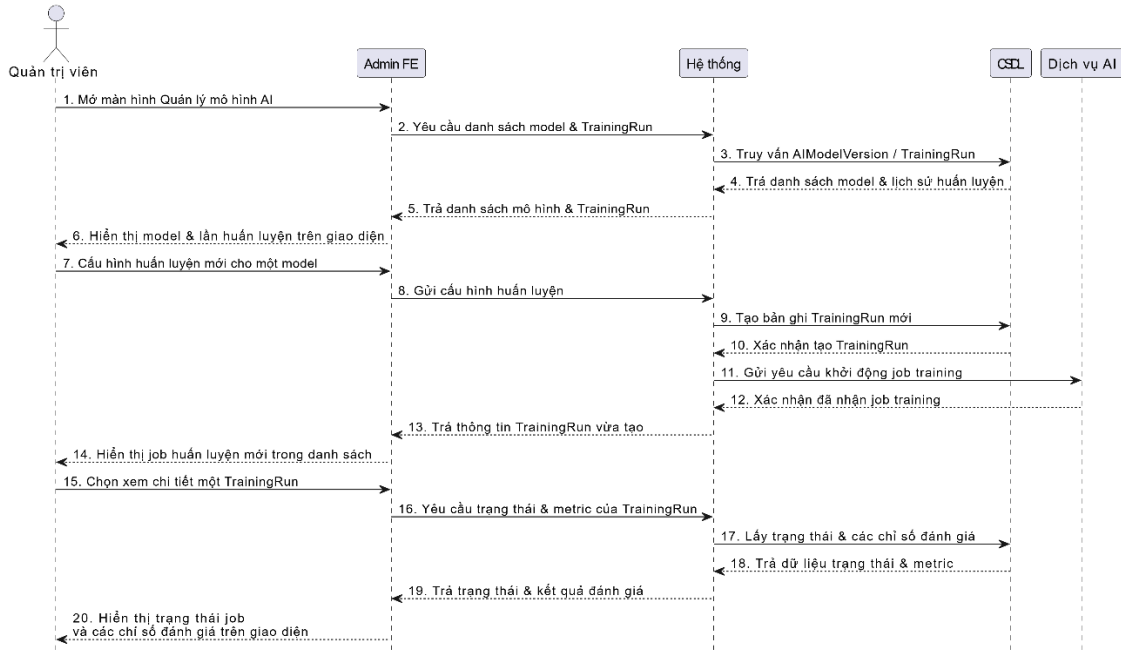


Hình 3. 18: Usecase quản lý mô hình và AI huấn luyện

Bảng 3. 9: Mô tả usecase quản lý mô hình và AI huấn luyện

Tên usecase	Quản lý tài khoản
Tác nhân	Quản trị viên
Điều kiện đầu vào	Tài khoản admin, bộ lọc tìm kiếm user thao tác quản trị.
Kết quả đầu ra	Danh sách tài khoản được hiển thị; trạng thái tài khoản được cập nhật, cờ quyền admin được cập nhật.
Mô tả	Quản trị viên dùng chức năng này để xem danh sách tài khoản, tìm kiếm người dùng, khoá/mở khoá tài khoản vi phạm, và cấp/thu hồi quyền quản trị cho những người dùng đủ thẩm quyền.
Luồng xử lý chính	1. Quản trị viên mở màn hình Quản lý tài khoản, hệ thống hiển thị danh sách tài khoản và cho phép tìm kiếm. 2. Quản trị viên chọn một tài khoản và thực hiện thao tác khoá/mở khoá hoặc cấp/thu hồi quyền admin.

	3. Hệ thống cập nhật trạng thái hoặc quyền của tài khoản trong CSDL. 4. Giao diện hiển thị thông báo và cập nhật lại thông tin tài khoản tương ứng.
--	---



Hình 3. 19: Sơ đồ tuần tự chức năng quản lý mô hình AI và huấn luyện

Khi quản trị viên thực hiện chức năng quản lý mô hình AI và các lần huấn luyện thì các bước xử lý chi tiết diễn ra như sau:

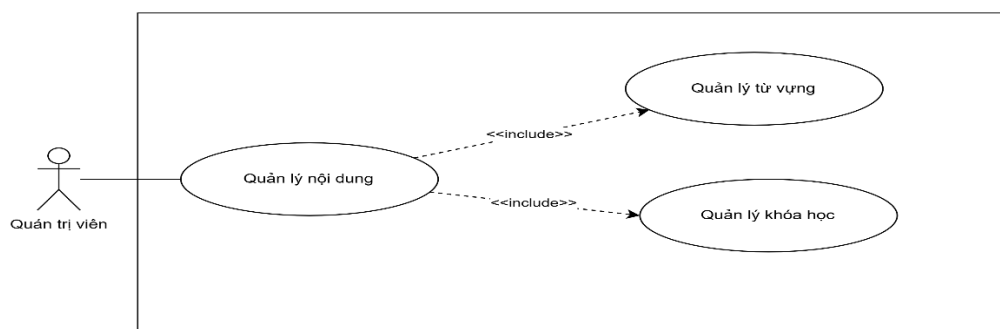
(1–6) Quản trị viên mở màn hình “Quản lý mô hình AI” trên giao diện admin. Giao diện gửi yêu cầu lên hệ thống để lấy danh sách các mô hình và các TrainingRun đã ghi nhận. Hệ thống truy vấn CSDL (bảng AIModelVersion và TrainingRun), tổng hợp dữ liệu và trả về cho giao diện; giao diện hiển thị danh sách model và lịch sử các lần huấn luyện tương ứng để quản trị viên theo dõi.

(7–14) Khi quản trị viên cấu hình một lần huấn luyện mới cho một mô hình (chọn model, nhập tham số, chọn tập dữ liệu...), giao diện gửi cấu hình huấn luyện lên hệ thống. Hệ thống tạo một bản ghi TrainingRun mới trong CSDL với trạng thái ban đầu (ví dụ queued) và nhận xác nhận, sau đó gửi yêu cầu khởi động job training sang dịch vụ AI và nhận xác nhận đã nhận job; hệ thống trả lại thông tin TrainingRun cho giao diện, giao diện hiển thị job huấn luyện mới trong danh sách để quản trị viên theo dõi.

(15–20) Khi quản trị viên muốn xem trạng thái hoặc kết quả đánh giá của một TrainingRun, giao diện gửi yêu cầu truy vấn trạng thái và metric lên hệ thống. Hệ thống

truy vấn CSDL để lấy trạng thái hiện tại và các chỉ số đánh giá (nếu đã có) của lần huấn luyện đó, sau đó trả dữ liệu trạng thái và metric về cho giao diện; giao diện hiển thị trạng thái job (đang chạy, đã hoàn thành, lỗi) cùng các chỉ số đánh giá để quản trị viên đánh giá chất lượng mô hình và ra quyết định sử dụng phiên bản phù hợp.

3.3.11 Quản lý nội dung

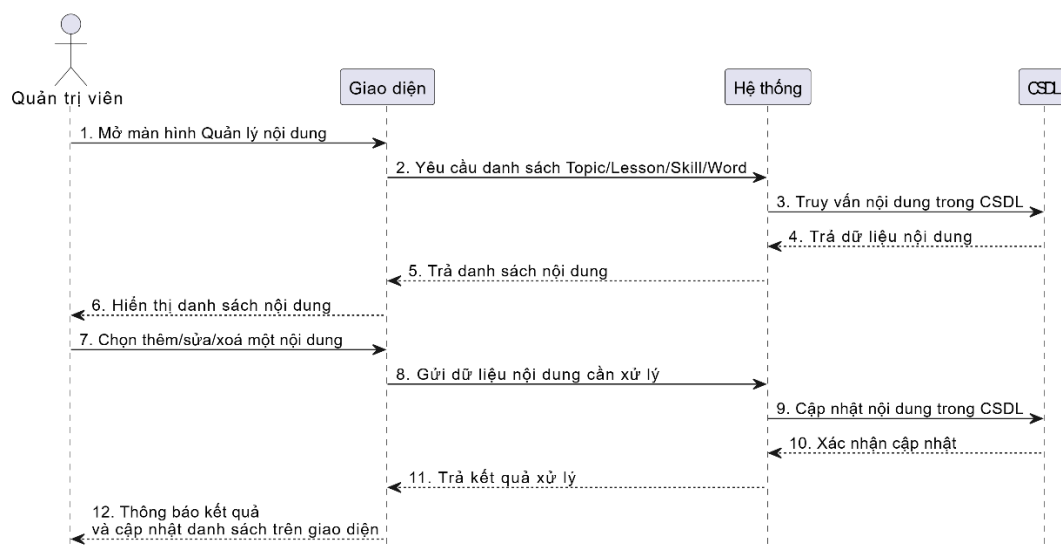


Hình 3. 20: Usecase quản lý nội dung

Bảng 3. 10: Mô tả usecase quản lý nội dung

Tên usecase	Quản lý nội dung (bài học, kỹ năng, từ vựng)
Tác nhân	Quản trị viên
Điều kiện đầu vào	- Quản trị viên đã đăng nhập hệ thống và có quyền quản lý nội dung. - Hệ thống đã có sẵn ít nhất một ngôn ngữ / chủ đề để gắn nội dung (Language, Topic).
Kết quả đầu ra	- Nội dung (Topic, Lesson, Skill, từ vựng, câu hỏi, kịch bản liên quan) được tạo mới, chỉnh sửa hoặc xóa theo yêu cầu. - Cấu trúc lộ trình học được cập nhật (danh sách bài học, danh sách skill trong bài, từ vựng gắn với bài/skill). - Danh sách nội dung hiển thị cho học viên được cập nhật đúng và nhất quán.
Mô tả	Use case này cho phép quản trị viên quản lý toàn bộ nội dung học tập của hệ thống, bao gồm tạo mới, chỉnh sửa, vô hiệu hóa hoặc xóa các Topic, Lesson, Skill và từ vựng liên quan. Thông qua đó, admin có thể thiết kế lộ trình học, điều chỉnh nội dung,

	cập nhật hoặc mở rộng kho bài học mà không cần can thiệp trực tiếp vào cơ sở dữ liệu.
Luồng xử lý chính	1. Quản trị viên đăng nhập vào hệ thống và truy cập khu vực quản trị nội dung. 2. Admin chọn loại nội dung cần thao tác (chủ đề, bài học, kỹ năng, từ vựng). 3. Hệ thống tải và hiển thị danh sách nội dung hiện có kèm các thao tác: thêm mới, chỉnh sửa, vô hiệu hóa/xóa. 4. Admin chọn một thao tác cụ thể: tạo mới bản ghi hoặc mở bản ghi hiện có để chỉnh sửa / thay đổi trạng thái hiển thị. 5. Admin nhập hoặc cập nhật thông tin chi tiết (ví dụ: tên, mô tả, ngôn ngữ, level, danh sách skill trong bài, danh sách từ vựng gắn với bài/skill) và xác nhận lưu. 6. Hệ thống kiểm tra tính hợp lệ dữ liệu, ghi thay đổi vào cơ sở dữ liệu và cập nhật quan hệ liên quan (ví dụ: LessonSkill, liên kết Topic-Lesson, liên kết Skill-nội dung chi tiết). 7. Hệ thống trả về thông báo thành công và hiển thị lại danh sách nội dung đã được cập nhật để admin tiếp tục thao tác.



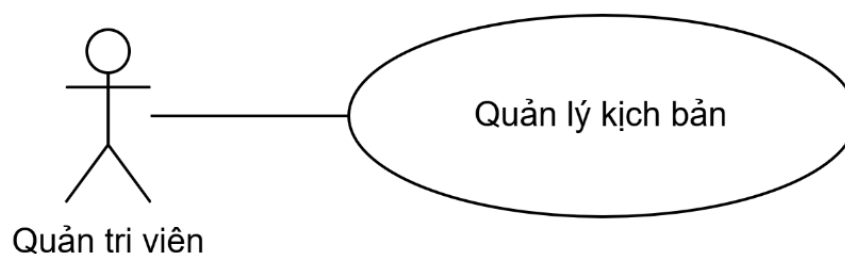
Hình 3. 21: sơ đồ tuần tự chức năng quản lý nội dung

Khi quản trị viên thực hiện chức năng quản lý nội dung (bài học, kỹ năng, từ vựng) thì các bước xử lý chi tiết diễn ra như sau:

(1–6) Quản trị viên mở màn hình “Quản lý nội dung” trên giao diện admin. Giao diện gửi yêu cầu lên hệ thống để lấy danh sách Topic/Lesson/Skill/Word hiện có. Hệ thống truy vấn CSDL, nhận dữ liệu nội dung và trả lại cho giao diện; giao diện hiển thị danh sách nội dung để quản trị viên xem và lựa chọn đối tượng cần chỉnh sửa.

(7–12) Khi quản trị viên thực hiện thao tác thêm mới, chỉnh sửa hoặc xoá một nội dung (ví dụ một Lesson, một Skill hoặc một từ vựng), giao diện gửi dữ liệu nội dung cần xử lý lên hệ thống. Hệ thống cập nhật bản ghi tương ứng trong CSDL và nhận xác nhận; sau đó hệ thống trả kết quả xử lý về cho giao diện, giao diện hiển thị thông báo thành công/thất bại và cập nhật lại danh sách nội dung trên màn hình để phản ánh thay đổi mới nhất.

3.3.12 Quản lý kịch bản



Hình 3. 22: Usecase quản lý kịch bản

Bảng 3. 11: Usecase quản lý kịch bản

Tên usecase	Quản lý kịch bản
Tác nhân	Quản trị viên
Điều kiện đầu vào	Tài khoản quản trị viên đã đăng nhập, có quyền truy cập module quản trị nội dung/roleplay.
Kết quả đầu ra	Danh sách kịch bản hội thoại (RoleplayScenario) và các block hội thoại (RoleplayBlock) được tạo mới, chỉnh sửa hoặc xóa; dữ liệu kịch bản được lưu nhất quán trong CSDL và sẵn sàng cho chức năng luyện hội thoại, luyện phát âm.
Mô tả	Use case cho phép quản trị viên quản lý toàn bộ kịch bản hội thoại roleplay của hệ thống. Quản trị viên có thể xem danh sách kịch bản, mở chi tiết từng kịch bản để xem các câu thoại, sau đó thêm, sửa, xóa các block hội thoại và gắn audio mẫu/TTS cho từng câu. Hệ thống đảm bảo thứ tự block, ngôn ngữ và liên kết với Scenario được lưu.
Luồng xử lý chính	1. Quản trị viên mở màn hình “Quản lý kịch bản roleplay”, hệ thống hiển thị danh sách các kịch bản (Scenario) hiện có. 2. Quản trị viên chọn một kịch bản để xem chi tiết, hệ thống tải thông tin Scenario và danh sách block hội thoại tương ứng và hiển thị trên giao diện. 3. Quản trị viên thực hiện các thao tác thêm mới, chỉnh sửa hoặc xóa các block hội thoại (câu nói, nhân vật, thứ tự, loại block, audio mẫu). 4. Hệ thống kiểm tra dữ liệu, cập nhật các bản ghi Scenario/Block, trả kết quả cho giao diện và hiển thị lại kịch bản roleplay sau khi chỉnh sửa.



Hình 3. 23: Usecase quản lý kịch bản

Khi quản trị viên thực hiện chức năng quản lý kịch bản roleplay thì các bước xử lý chi tiết diễn ra như sau:

(1–6) Quản trị viên mở màn hình “Quản lý Roleplay” trên giao diện admin. Giao diện gửi yêu cầu lên hệ thống để lấy danh sách các RoleplayScenario hiện có. Hệ thống truy vấn CSDL, nhận danh sách Scenario và trả về cho giao diện; giao diện hiển thị danh sách kịch bản roleplay để quản trị viên chọn kịch bản cần thao tác.

(7–12) Khi quản trị viên chọn một Scenario cụ thể, giao diện gửi yêu cầu lên hệ thống để lấy chi tiết Scenario và các RoleplayBlock thuộc kịch bản đó. Hệ thống truy vấn CSDL, lấy thông tin Scenario và danh sách Block hội thoại rồi trả về cho giao diện; giao diện hiển thị danh sách các câu thoại/block theo thứ tự để quản trị viên quan sát và chỉnh sửa.

(13–18) Khi quản trị viên thêm, sửa hoặc xóa các Block hội thoại, giao diện gửi danh sách Block đã chỉnh sửa lên hệ thống. Hệ thống cập nhật các bản ghi RoleplayBlock tương ứng trong CSDL và nhận xác nhận, sau đó hệ thống trả kết quả chỉnh sửa về cho giao diện, giao diện hiển thị thông báo kết quả và cập nhật lại kịch bản roleplay đã chỉnh sửa để quản trị viên kiểm tra.

3.4 Thiết kế cơ sở dữ liệu

Để xây dựng các chức năng cơ bản của hệ thống học ngoại ngữ. Mô tả chi tiết các bảng lưu trữ thông tin như sau:

Bảng 3. 12: Bảng User Lưu trữ thông tin tài khoản người dùng.

Tên trường	Kiểu dữ liệu	Mô tả	Khóa
id	INT	Khóa chính tự tăng của người dùng	PK
username	VARCHAR	Tên đăng nhập của người dùng, là duy nhất	UNIQUE
email	VARCHAR	Địa chỉ email dùng để đăng ký / đăng nhập	UNIQUE
password	VARCHAR	Mật khẩu đã được băm (không lưu dạng plain text)	
avatar	VARCHAR	Đường dẫn (URL) ảnh đại diện của người dùng	
bio	TEXT	Thông tin giới thiệu ngắn về người dùng	
verify_code	CHAR(6)	Mã xác thực/OTP dùng để kích hoạt hoặc xác minh tài khoản	
last_active	DATETIME	Thời điểm người dùng hoạt động gần nhất	
first_name	VARCHAR	Tên hoặc họ tên đệm của người dùng (nếu hệ thống có sử dụng)	
last_name	VARCHAR	Họ hoặc phần còn lại của họ tên người dùng	
is_active	BOOLEAN	Cờ cho biết tài khoản có đang được kích hoạt hay không	
is_staff	BOOLEAN	Cờ cho biết tài khoản có quyền truy cập trang quản trị (admin)	
is_superuser	BOOLEAN	Cờ cho biết tài khoản có toàn quyền trong hệ thống	
date_joined	DATETIME	Thời điểm tài khoản được tạo trong hệ thống	
last_login	DATETIME	Thời điểm người dùng đăng nhập hệ thống gần nhất	

Bảng 3. 13: Bảng Languages Danh sách các ngôn ngữ mà hệ thống hỗ trợ giảng dạy.

Tên trường	Kiểu dữ liệu	Mô tả	Khóa
id	INT	Khóa chính tự tăng	PK
name	VARCHAR	Trường name của Language	
abbreviation	VARCHAR	Kí hiệu của ngôn ngữ chỉ định (en, vi,..)	UNIQUE
native_name	VARCHAR	Tên gốc của ngôn ngữ	
direction	VARCHAR	Hướng viết văn bản (LTR, RTL)	
created_at	TIMESTAMP	Thời điểm bản ghi được tạo	

Bảng 3.14: Bảng LanguageEnrollment Lưu trữ thông tin ghi danh của người dùng vào một ngôn ngữ cụ thể (tiến độ tổng quan).

Tên trường	Kiểu dữ liệu	Mô tả	Khóa
id	INT	Khóa chính định danh một bản ghi ghi danh ngôn ngữ	PK
user	INT	Người dùng được ghi danh vào ngôn ngữ này	FK User
language	INT	Ngôn ngữ mà người dùng đang học	FK Language
level	INT	Cấp độ hiện tại của người học trong ngôn ngữ (theo thang nội bộ)	
total_xp	INT	Tổng số điểm kinh nghiệm (XP) đã tích lũy cho ngôn ngữ này	
streak_days	INT	Số ngày học liên tiếp hiện tại (streak)	
last_practiced	DATETIME	Lần cuối cùng người học luyện tập ngôn ngữ này	
created_at	DATETIME	Thời điểm người dùng bắt đầu ghi danh ngôn ngữ	

Bảng 3.15: Bảng Lesson Định nghĩa các bài học cụ thể thuộc về một chủ đề.

Tên trường	Kiểu dữ liệu	Mô tả	Khóa
id	INT	Khóa chính định danh một bài học	PK
topic	INT	Chủ đề mà bài học này thuộc về	FK Topic
title	VARCHAR	Tiêu đề bài học	
content	JSON	Nội dung/mô tả cấu trúc bài học	
order	INT	Thứ tự hiển thị bài học trong chủ đề	
xp_reward	INT	Tổng số XP dự kiến nhận khi hoàn thành bài học	
duration_seconds	INT	Thời lượng ước tính của bài học	

Bảng 3.16: Bảng Skill Định nghĩa các kỹ năng hoặc bài tập nhỏ trong bài học.

Tên trường	Kiểu dữ liệu	Mô tả	Khóa
id	INT	Khóa chính định danh một kỹ năng (skill)	PK
title	VARCHAR	Tiêu đề năng	
type	VARCHAR	Loại kỹ năng	
xp_reward	INT	Số điểm XP khi hoàn thành	
duration_seconds	INT	Thời lượng ước tính để hoàn thành skill	
difficulty	SMALLINT	Độ khó tương đối của kỹ năng	
title_i18n	JSON	Tiêu đề kỹ năng đa ngôn ngữ	
description_i18n	JSON	Mô tả chi tiết kỹ năng đa ngôn ngữ	
language_code	VARCHAR	Mã ngôn ngữ nội dung chính của skill	
tags	JSON	Danh sách nhãn gắn với skill	
is_active	BOOLEAN	Trạng thái	

Bảng 3.17: Bảng Word Kho từ vựng của hệ thống.

Tên trường	Kiểu dữ liệu	Mô tả	Khóa
id	INT	Khóa chính từ vựng	PK
language	INT	Ngôn ngữ mà từ vựng thuộc về	FK Language
text	VARCHAR	Chuỗi gốc của từ vựng	UNIQUE
normalized	VARCHAR	Dạng chuẩn hóa	
part_of_speech	VARCHAR	Từ loại: noun, verb, adj,...	
created_at	DATETIME	Thời điểm tạo từ	

Bảng 3.18: Bảng Misstake Lưu trữ lịch sử các lỗi sai của người dùng.

Tên trường	Kiểu dữ liệu	Mô tả	Khóa
id	INT	Khóa chính bản ghi lỗi	PK
skill	INT	Kỹ năng liên quan đến lỗi	FK Skill
user	INT	Người dùng mắc lỗi	FK User
enrollment	INT	Ghi danh ngôn ngữ	FK LanguageEnrollment
interaction	INT	Bản ghi LearningInteraction	FK LearningInteraction
lesson	INT	Bài học liên quan đến lỗi	FK Lesson
word	INT	Từ vựng liên quan đến lỗi	FK Word
source	VARCHAR	Loại lỗi	
description	TEXT	Mô tả ngắn về lỗi	
error_detail	JSON	Chi tiết cấu trúc	
score	FLOAT	Điểm đánh giá lỗi	
confidence	FLOAT	Độ tin cậy từ mô hình AI khi phát hiện lỗi	
timestamp	DATETIME	Thời điểm phát sinh/ghi nhận lỗi	

Bảng 3.19: Bảng LearningInteraction chi tiết mọi tương tác học tập phân tích hành vi.

Tên trường	Kiểu dữ liệu	Mô tả	Khóa
id	INT	Khóa chính log tương tác học tập	PK
user	INT	Người dùng thực hiện hành động	FK User
enrollment	INT	Ghi danh ngôn ngữ tại thời điểm tương tác	FK LanguageEnrollment
lesson	INT	Bài học liên quan	FK Lesson
word	INT	Từ vựng liên quan	FK Word
skill	INT	Kỹ năng liên quan đến tương tác	FK Skill
action	VARCHAR	Loại hành động	
value	FLOAT	Giá trị đo được: điểm, tỉ lệ đúng,...	
success	BOOLEAN	Tương tác thành công hay thất bại	
duration_seconds	INT	Thời lượng tương tác	
xp_earned	INT	XP kiếm được từ tương tác này	
meta	JSON	Thông tin chi tiết khác	
created_at	DATETIME	Thời điểm ghi nhận tương tác	

Bảng 3.20: Bảng Topic Định nghĩa các chủ đề lớn trong lộ trình học.

Tên trường	Kiểu dữ liệu	Mô tả	Khóa
id	INT	Khóa chính định danh một chủ đề (topic) học	PK
language	INT	Ngôn ngữ mà chủ đề này thuộc về	FK Language
slug	VARCHAR	Mã định danh dạng URL	UNIQUE
title	VARCHAR	Tên chủ đề hiển thị	
description	TEXT	Mô tả	
order	INT	Thứ tự hiển thị chủ đề	
golden	BOOLEAN	Đánh dấu quan trọng/đề xuất	
created_at	DATETIME	Thời điểm tạo chủ đề	
embedding	VECTOR(768)	Vector biểu diễn chủ đề dùng cho tìm kiếm/gợi ý (RAG/AI)	
embedding_text	TEXT	Đoạn văn bản được dùng để sinh embedding cho chủ đề	
embedding_updated_at	DATETIME	Lần cuối cùng embedding của chủ đề được cập nhật	
embedding_hash	VARCHAR(64)	Hash của embedding_text/phiên bản để tránh tính lại không cần thiết	
embedding_model	VARCHAR	Tên mô hình embedding đã sử dụng	

Bảng 3. 21: Bảng SessionAnswer Lưu trữ câu trả lời chi tiết của người dùng trong một phiên làm bài.

Tên trường	Kiểu dữ liệu	Mô tả	Khóa
id	INT	Khóa chính	PK
session	INT	Phiên làm bài mà câu trả lời này thuộc về	FK LessonSession

skill	INT	Kỹ năng tương ứng với câu hỏi này	FK Skill
question_id	VARCHAR	Định danh câu hỏi	
is_correct	BOOLEAN	Học viên trả lời đúng hay sai	
user_answer	TEXT	Nội dung câu trả lời của học viên	
expected	TEXT	Đáp án hoặc gợi ý đáp án chuẩn	
meta	JSON	Metadata bổ sung:	
created_at	DATETIME	Thời điểm lưu câu trả lời	

Bảng 3. 22: Bảng RolePlayScenario Định nghĩa các kịch bản hội thoại nhập vai.

Tên trường	Kiểu dữ liệu	Mô tả	Khóa
id	UUID	Khóa chính dạng UUID cho kịch bản hội thoại	PK
slug	VARCHAR	Mã định danh ngắn, duy nhất cho kịch bản	UNIQUE
title	VARCHAR	Tiêu đề kịch bản hội thoại	
description	TEXT	Mô tả chi tiết bối cảnh/nội dung kịch bản	
level	CHAR(2)	Cấp CEFR mục tiêu (A1, A2, B1, ...)	
order	INT	Thứ tự sắp xếp kịch bản trong danh sách	
tags	JSON	Danh sách nhãn cho kịch bản	
skill_tags	JSON	Nhãn liên kết với các skill	
is_active	BOOLEAN	Cờ bật/tắt kịch bản trong hệ thống	
created_at	DATETIME	Thời điểm tạo kịch bản	
updated_at	DATETIME	Thời điểm cập nhật gần nhất	
embedding	VECTOR(768)	Vector embedding dùng cho tìm kiếm/gợi ý hội thoại	
embedding_text	TEXT	Văn bản nguồn để sinh embedding	
embedding_updated_at	DATETIME	Lần cuối cập nhật embedding	

embedding_hash	VARCHAR	Hash nội dung embedding_text để tránh tính lại	
embedding_model	VARCHAR	Tên mô hình embedding sử dụng	

Bảng 3. 23: Bảng RolePlayBlock Các khối nội dung chi tiết trong kịch bản Roleplay.

Tên trường	Kiểu dữ liệu	Mô tả	Khóa
id	UUID	Khóa chính của một block trong kịch bản hội thoại	PK
scenario	UUID	Kịch bản hội thoại mà block này thuộc về	FK RoleplayScenario
section	VARCHAR	Loại block	
order	INT	Thứ tự trong kịch bản	UNIQUE
role	VARCHAR	Vai trò nói	
text	TEXT	Nội dung câu thoại / chỉ dẫn / từ vựng	
extra	JSON	Thông tin bổ sung	
audio_key	VARCHAR	Khóa audio	
tts_voice	VARCHAR	Tên giọng TTS gợi ý dùng để đọc block này	
lang_hint	VARCHAR	Gợi ý ngôn ngữ của block	
is_active	BOOLEAN	Bật/tắt block trong kịch bản	
created_at	DATETIME	Thời điểm tạo block	
updated_at	DATETIME	Thời điểm chỉnh sửa gần nhất	
embedding	VECTOR(768)	Embedding cho block	

embedding_text	TEXT	Văn bản nguồn để sinh embedding	
embedding_updated_at	DATETIME	Lần cuối cập nhật embedding	
embedding_hash	VARCHAR	Hash nội dung embedding_text	
embedding_model	VARCHAR	Tên mô hình embedding	

Bảng 3. 24: Bảng PronAttempt Lưu trữ chi tiết từng lần thu âm luyện phát âm để chấm điểm AI.

Tên trường	Kiểu dữ liệu	Mô tả	Khóa
id	INT	Khóa chính tự tăng	PK
session	INT	Phiên luyện tập kỹ năng chứa lần thử này	FK SkillSession
prompt_id	INT	Câu/Từ mẫu được yêu cầu đọc	FK PronunciationPrompt
expected_text	TEXT	Văn bản chuẩn cần đọc	
recognized	TEXT	Văn bản AI nhận dạng được	
score_overall	FLOAT	Điểm chấm tổng quát	
audio_path	VARCHAR	Đường dẫn file âm thanh người dùng đã thu	
details	JSON	Chi tiết lỗi (WER, CER, độ tin cậy...)	

Bảng 3. 25: Bảng TopicProgress Lưu trữ tiến độ học tập của người dùng theo từng chủ đề lớn.

Tên trường	Kiểu dữ liệu	Mô tả	Khóa
id	INT	Khóa chính tự tăng	PK

enrollment	INT	Ghi danh ngôn ngữ liên quan	FK LanguageEnrollment
topic	INT	Chủ đề đang học	FK Topic
highest_completed_order	INT	Thứ tự bài học cao nhất đã hoàn thành trong chủ đề	
completed	BOOLEAN	Đánh dấu đã hoàn thành toàn bộ chủ đề	
xp	INT	Tổng XP đạt được trong chủ đề này	
last_seen	DATETIME	Thời gian truy cập chủ đề gần nhất	

Bảng 3. 26: Bảng LessonSession Lưu trữ thông tin chi tiết về một phiên làm bài học (một lần người dùng vào học một bài).

Tên trường	Kiểu dữ liệu	Mô tả	Khóa
id	INT	Khóa chính tự tăng	PK
session_id	VARCHAR	Mã định danh duy nhất (UUID) cho session	UNIQUE
user	INT	Người dùng thực hiện phiên học	FK User
lesson	INT	Bài học đang thực hiện	FK Lesson
enrollment	INT	Ghi danh ngôn ngữ liên quan	FK LanguageEnrollment
status	VARCHAR	Trạng thái	
started_at	DATETIME	Thời điểm bắt đầu	
completed_at	DATETIME	Thời điểm hoàn thành	
correct_answers	INT	Số câu trả lời đúng	

xp_earned	INT	Số XP kiếm được trong phiên này	
perfect_lesson	BOOLEAN	Cờ đánh dấu bài học hoàn hảo (không sai câu nào)	

Bảng 3. 27: Bảng UserSkillStats Lưu trữ thống kê chi tiết về độ thành thạo từng kỹ năng (Skill) của người dùng.

Tên trường	Kiểu dữ liệu	Mô tả	Khóa
id	INT	Khóa chính tự tăng	PK
enrollment	INT	Ghi danh ngôn ngữ liên quan	FK LanguageEnrollment
skill	INT	Kỹ năng được theo dõi	FK Skill
level	INT	Cấp độ thành thạo hiện tại (Crown level 0-5)	
proficiency_score	FLOAT	Điểm thành thạo (0.0 - 100.0)	
needs_review	BOOLEAN	Cờ báo hiệu kỹ năng này cần ôn tập lại	
status	VARCHAR	Trạng thái	
last_practiced	DATETIME	Lần cuối luyện tập kỹ năng này	

Bảng 3. 28: Bảng DailyXP Theo dõi XP kiếm được theo ngày để vẽ biểu đồ hoạt động.

Tên trường	Kiểu dữ liệu	Mô tả	Khóa
id	INT	Khóa chính tự tăng	PK
user	INT	Người dùng	FK User
date	DATE	Ngày ghi nhận	
xp	INT	Tổng XP kiếm được trong ngày đó	

Bảng 3. 29: Bảng Friend Lưu trữ mối quan hệ bạn bè giữa các người dùng.

Tên trường	Kiểu dữ liệu	Mô tả	Khóa
id	INT	Khóa chính tự tăng	PK
from_user	INT	Người gửi lời mời kết bạn	FK User
to_user	INT	Người nhận lời mời	FK User
accepted	BOOLEAN	Trạng thái chấp nhận	
created_at	DATETIME	Thời điểm gửi lời mời	

Bảng 3. 30: Bảng LeaderboardEntry Lưu trữ dữ liệu bảng xếp hạng thi đua.

Tên trường	Kiểu dữ liệu	Mô tả	Khóa
id	INT	Khóa chính tự tăng	PK
user	INT	Người dùng trên bảng xếp hạng	FK User
language	INT	Ngôn ngữ xếp hạng	FK Language
date	DATE	Ngày/Tuần/Tháng tính bảng xếp hạng	
rank	INT	Thứ hạng hiện tại	
xp	INT	Số XP dùng để xếp hạng	

Bảng 3. 31: Bảng UserBadge Lưu trữ các huy hiệu mà người dùng đã đạt được.

Tên trường	Kiểu dữ liệu	Mô tả	Khóa
id	INT	Khóa chính tự tăng	PK
user	INT	Người dùng đạt huy hiệu	FK User
badge	INT	Huy hiệu đạt được	FK Badge
awarded_at	DATETIME	Thời điểm nhận huy hiệu	
progress	INT	Tiến độ hiện tại (nếu huy hiệu cần tích lũy)	

Bảng 3. 32: Bảng Badge Định nghĩa các huy hiệu/thành tích trong hệ thống.

Tên trường	Kiểu dữ liệu	Mô tả	Khóa
id	INT	Khóa chính tự tăng	PK
slug	VARCHAR	Mã định danh duy nhất của huy hiệu	UNIQUE
name	VARCHAR	Tên hiển thị của huy hiệu	
description	TEXT	Mô tả cách đạt được huy hiệu	
criteria	JSON	Điều kiện logic để mở khóa	
icon	VARCHAR	Đường dẫn icon huy hiệu	

Bảng 3. 33: Bảng Notification Lưu trữ thông báo gửi đến người dùng.

Tên trường	Kiểu dữ liệu	Mô tả	Khóa
id	INT	Khóa chính tự tăng	PK
user	INT	Người nhận thông báo	FK User
type	VARCHAR	Loại thông báo	
title	VARCHAR	Tiêu đề thông báo	
body	TEXT	Nội dung thông báo	
read_at	DATETIME	Thời điểm người dùng đã xem	

Bảng 3. 34: Bảng KnownWord Lưu trữ từ vựng người dùng đã học.

Tên trường	Kiểu dữ liệu	Mô tả	Khóa
id	INT	Khóa chính tự tăng	PK
enrollment	INT	Ghi danh ngôn ngữ liên quan	FK LanguageEnrollment
word	INT	Từ vựng đang theo dõi	FK Word
status	VARCHAR	Trạng thái	

next_review	DATETIME	Thời điểm cần ôn tập tiếp theo (thuật toán SM-2)	
score	FLOAT	Điểm đánh giá mức độ thuộc từ (0-100)	
ease_factor	FLOAT	Hệ số ghi nhớ (dùng cho thuật toán lặp lại)	

Bảng 3. 35: Bảng PronunciationPrompt Dữ liệu câu/từ mẫu dùng cho bài tập phát âm.

Tên trường	Kiểu dữ liệu	Mô tả	Khóa
id	INT	Khóa chính tự tăng	PK
skill	INT	Thuộc về kỹ năng nào	FK Skill
word	VARCHAR	Từ hoặc câu cần phát âm	
phonemes	VARCHAR	Phiên âm IPA (nếu có)	
tts_file	VARCHAR	Đường dẫn file âm thanh mẫu (TTS sinh ra)	

Bảng 3. 36: Bảng SkillQuestion Câu hỏi trắc nghiệm (Quiz) trong các bài Skill.

Tên trường	Kiểu dữ liệu	Mô tả	Khóa
id	INT	Khóa chính tự tăng	PK
skill	INT	Thuộc về kỹ năng nào	FK Skill
question_text	TEXT	Nội dung câu hỏi	

Bảng 3. 37: Bảng SkillChoice Các phương án lựa chọn cho câu hỏi trắc nghiệm.

Tên trường	Kiểu dữ liệu	Mô tả	Khóa
id	INT	Khóa chính tự tăng	PK
question	INT	Thuộc về câu hỏi nào	FK SkillQuestion
text	VARCHAR	Nội dung phương án	
is_correct	BOOLEAN	Là đáp án đúng hay sai	

Bảng 3. 38: Bảng SkillGap Lưu trữ dữ liệu cho dạng bài tập điền từ vào chỗ trống.

Tên trường	Kiểu dữ liệu	Mô tả	Khóa
id	INT	Khóa chính tự tăng	PK
skill	INT	Thuộc về kỹ năng nào	FK Skill
text	TEXT	Câu văn chứa chỗ trống (placeholder)	
answer	VARCHAR	Đáp án đúng cần điền	

Bảng 3. 39: Bảng OrderingItem Lưu trữ các mảnh ghép cho dạng bài sắp xếp lại câu.

Tên trường	Kiểu dữ liệu	Mô tả	Khóa
id	INT	Khóa chính tự tăng	PK
skill	INT	Thuộc về kỹ năng nào	FK Skill
text	VARCHAR	Nội dung của mảnh ghép	
order_index	INT	Vị trí đúng trong câu hoàn chỉnh	

Bảng 3. 40: Bảng MatchingPair Lưu trữ các cặp từ/nghĩa cho dạng bài tập nối cột.

Tên trường	Kiểu dữ liệu	Mô tả	Khóa
id	INT	Khóa chính tự tăng	PK
skill	INT	Thuộc về kỹ năng nào	FK Skill
left_text	VARCHAR	Nội dung cột trái	
right_text	VARCHAR	Nội dung cột phải tương ứng (đáp án)	
left_text_i18n	JSON	Đa ngôn ngữ cho cột trái (nếu có)	

Bảng 3. 41: Bảng ListeningPrompt Lưu trữ dữ liệu cho bài tập luyện nghe.

Tên trường	Kiểu dữ liệu	Mô tả	Khóa
id	INT	Khóa chính tự tăng	PK
skill	INT	Thuộc về kỹ năng nào	FK Skill
audio_url	URL	Đường dẫn file âm thanh đề bài	
question_text	VARCHAR	Câu hỏi hiển thị (nếu có)	
answer	VARCHAR	Đáp án văn bản của bài nghe	

Bảng 3. 42: Bảng ReadingContent Lưu trữ đoạn văn bản dài cho bài tập đọc hiểu.

Tên trường	Kiểu dữ liệu	Mô tả	Khóa
id	INT	Khóa chính tự tăng	PK
skill	INT	Thuộc về kỹ năng nào	FK Skill (1-1)
passage	TEXT	Nội dung đoạn văn cần đọc	

Bảng 3. 43: Bảng ReadingQuestion Lưu trữ câu hỏi liên quan đến đoạn văn đọc hiểu.

Tên trường	Kiểu dữ liệu	Mô tả	Khóa
id	INT	Khóa chính tự tăng	PK
skill	INT	Thuộc về kỹ năng đọc nào	FK Skill
question_text	TEXT	Nội dung câu hỏi	
answer	VARCHAR	Đáp án đúng	

CHƯƠNG 4 TRIỂN KHAI HỆ THỐNG

4.1 Công cụ và công nghệ triển khai

4.1.1 Django



Hình 4.1: Logo django

Django một framework lập trình web bậc cao, mã nguồn mở được viết bằng ngôn ngữ Python, nổi tiếng với triết lý "batteries-included" (tích hợp sẵn mọi thứ cần thiết). Ra đời với mục tiêu giúp các lập trình viên xây dựng ứng dụng web nhanh chóng, an toàn và dễ bảo trì, Django tuân theo mô hình kiến trúc MVT (Model-View-Template). Điểm mạnh của Django nằm ở khả năng bảo mật vượt trội, tự động ngăn chặn các lỗ hổng phổ biến như SQL Injection hay Cross-site Scripting (XSS), cùng với hệ thống quản trị (Admin Interface) được sinh ra tự động, giúp tiết kiệm đáng kể thời gian phát triển các tính năng quản lý nội dung. Trong dự án này, Django được lựa chọn làm nền tảng cốt lõi cho phía Backend, chịu trách nhiệm quản lý logic nghiệp vụ phức tạp và tương tác với cơ sở dữ liệu. Nhờ vào hệ thống ORM (Object-Relational Mapping) mạnh mẽ, việc thao tác với dữ liệu quan hệ trở nên trực quan thông qua các đối tượng Python thay vì phải viết các câu lệnh SQL thô. Hệ thống cũng tận dụng các middleware của Django để xử lý xác thực người dùng, quản lý session và đảm bảo tính toàn vẹn của dữ liệu trước khi được xử lý bởi các tầng tiếp theo.

4.1.2 Django Rest Framework



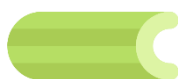
Hình 4. 2: logo django rest framework

Django Rest Framework (DRF) là nền tảng xây dựng API tiêu chuẩn trong hệ sinh thái Django, được lựa chọn làm lõi backend nhằm đảm bảo tốc độ phát triển, độ tin cậy và khả năng mở rộng. DRF cung cấp cơ chế serialization mạnh mẽ, quản lý xác thực (authentication) linh hoạt, phân quyền (permission) chi tiết, và hỗ trợ tốt cho mô hình microservices.

Trong hệ thống, DRF đảm nhiệm: Xây dựng API cho toàn bộ nghiệp vụ: bài học, kỹ năng, bài tập, roleplay, thống kê học tập, gamification, XP, leaderboard. Cung cấp endpoint RESTful cho frontend React và các dịch vụ AI truy cập. Tích hợp tốt với Postgres, Redis, Celery và MinIO để xử lý các tác vụ xử lý dữ liệu lớn, streaming hoặc upload file. Hỗ trợ WebSocket thông qua ASGI layer cho các chức năng thời gian thực (realtime leaderboard, roleplay chat).

DRF phù hợp với yêu cầu mở rộng data model phức tạp của hệ thống học ngoại ngữ (Lesson, Skill, UserSkillStats, LearningInteraction...).

4.1.3 Celery

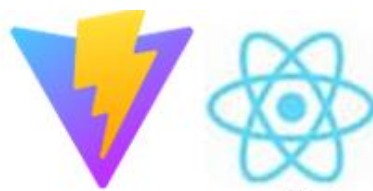


Hình 4. 3: logo celery

Celery là một thư viện xử lý tác vụ bất đồng bộ, cho phép thực thi các công việc nặng, chạy nền hoặc chạy theo lịch mà không làm chậm hệ thống chính. Trong hệ thống học ngoại ngữ, Celery được dùng để xử lý audio bằng Whisper (ASR), sinh giọng bằng gTTS hoặc Piper (TTS), và điều phối pipeline chấm điểm phát âm gồm chuẩn hóa âm thanh, nhận dạng, so sánh và tính điểm.

Celery cũng đảm nhiệm cập nhật gamification và leaderboard, gửi email hoặc thông báo nhắc học, và thực hiện các tác vụ định kỳ thông qua Celery Beat như thống kê, làm mới dữ liệu và dọn rác. Ngoài ra, hệ thống sử dụng Celery cho xử lý batch và các quy trình huấn luyện offline. Việc tích hợp Celery giúp hệ thống vận hành nhanh hơn, mượt hơn và dễ mở rộng.

4.1.4 Reactjs vite



Hình 4. 4: logo vite và reactjs

Frontend được xây dựng bằng React kết hợp với Vite nhằm tối ưu hóa hiệu năng và trải nghiệm người dùng. Vite sử dụng cơ chế native ESBuild, cho tốc độ khởi động dự án, hot reload và bundling nhanh hơn.

Trong hệ thống: React đảm nhận UI/UX cho các luồng chính. Vite đảm bảo tốc độ phát triển nhanh, thời gian phản hồi thấp, phù hợp ứng dụng giàu tương tác (SPA). Khả năng kết hợp React Query, Redux Toolkit giúp quản lý trạng thái học tập, cache dữ liệu và tối ưu API calls. Dễ đóng gói vào Docker Container để triển khai thống nhất với backend. React + Vite là lựa chọn phù hợp cho nền tảng học ngoại ngữ vì yêu cầu nhiều tương tác thời gian thực, streaming âm thanh STT/TTS, và giao diện động.

4.1.5 Docker



Hình 4. 5: Logo docker

Docker được sử dụng để đóng gói toàn bộ kiến trúc microservices và đảm bảo môi trường chạy đồng nhất giữa các môi trường phát triển, staging và production. Trong hệ thống, Docker orchestration bao gồm: Backend Django/DRF container. Frontend React container. MinIO, Redis, Postgres, Nginx, Celery worker, Celery beat, và các dịch vụ

AI (Whisper STT, Piper TTS, Ollama LLM). Docker Compose dùng để khai báo các service, volume, network.

Việc container hóa giúp: Tách biệt từng dịch vụ để dễ nâng cấp, thay thế hoặc scale ngang. Đảm bảo tính reproducible: code chạy giống nhau trên mọi máy. Tăng tính bảo trì dự án và giảm rủi ro lệ thuộc môi trường.

4.1.6 MinIO



Hình 4. 6: logo minIO

MinIO được sử dụng như kho lưu trữ đối tượng (object storage) thay thế cho Amazon S3 nhưng triển khai cục bộ. MinIO hỗ trợ giao thức S3 API, dễ tích hợp với Django-Storages. MinIO lưu trữ toàn bộ audio: file TTS, STT upload, user recordings, voice assets, roleplay audio, vocabulary audio. Lưu file ảnh minh họa, avatar, tài liệu.

Tối ưu cho dữ liệu phi cấu trúc dung lượng lớn. Dễ scale khi hệ thống tăng lượng học viên. Việc dùng MinIO giúp tách layer lưu trữ khỏi backend, giảm tải cho Postgres và phù hợp với kiến trúc microservices.

4.1.7 Redis



Hình 4. 7: logo Redis

Redis là hệ thống lưu trữ key-value trong bộ nhớ có tốc độ cao, phù hợp cho caching và các tác vụ hàng nghìn request/giây. Cache API responses (đặc biệt: Word definitions, Lessons, Leaderboard ranking, KnownWords). Lưu session WebSocket và realtime state. Ngăn nghẽn hiệu năng khi truy cập dữ liệu lặp lại. Training mô hình gợi

ý. Convert file audio/video. Redis tối ưu rõ rệt cho các nghiệp vụ cần tốc độ cao và trạng thái tạm thời.

4.1.8 Triển khai hệ thống

4.1.8.1 Giới thiệu về Vercel và ngrok

a. Vercel

Vercel là nền tảng triển khai (deployment platform) tối ưu cho các ứng dụng frontend, đặc biệt là các framework hiện đại như React, Vite, Next.js và Svelte. Vercel cung cấp các đặc điểm nổi bật:

- Hỗ trợ tốt cho ứng dụng SPA (Single Page Application).
- Build và deploy tự động mỗi khi có commit mới lên GitHub nhờ tích hợp CI/CD sẵn.
- CDN toàn cầu giúp tăng tốc độ truy cập.
- Cung cấp HTTPS mặc định, không cần cấu hình chứng chỉ SSL.

Tự động tạo preview deployment cho từng pull request, hỗ trợ kiểm thử giao diện nhanh chóng.

Vercel được sử dụng để triển khai phần React + Vite, đảm bảo giao diện người dùng luôn hoạt động trực tuyến, phục vụ kiểm thử chức năng và trải nghiệm toàn bộ hệ thống.

b. Ngrok

Ngrok là công cụ tạo đường hầm bảo mật từ internet tới máy cục bộ, cho phép expose một dịch vụ backend đang chạy local ra ngoài thông qua domain công khai (public URL) dạng HTTPS. Một số ưu điểm:

- Tạo URL HTTPS ngay lập tức, không cần cấu hình server hay hosting.
- Giúp frontend gọi backend local từ môi trường online.
- Dễ thiết lập trong giai đoạn phát triển, chỉ cần một câu lệnh để khởi chạy tunnel.
- Cung cấp dashboard theo dõi request tại <http://127.0.0.1:4040>, hỗ trợ debug hiệu quả.

Trong quá trình thực hiện đồ án, backend Django (REST API) chưa được triển khai lên server thật mà chạy cục bộ. Việc dùng ngrok giúp backend “xuất hiện” trên internet, tạo điều kiện cho frontend trên Vercel kết nối và kiểm thử liền mạch.

4.1.8.2 Mô hình triển khai

Việc triển khai được thiết kế dựa trên đặc thù của từng thành phần trong hệ thống:

- Frontend (React + Vite): Được triển khai lên Vercel để giao diện luôn sẵn sàng trực tuyến, đảm bảo người dùng/nhóm kiểm thử có thể truy cập từ bất kỳ thiết bị nào mà không cần thiết lập máy cục bộ. Vercel đồng thời đảm nhiệm vai trò CI/CD cho frontend, tự động build và deploy mỗi khi cập nhật mã nguồn.
- Backend (Django REST Framework): Do mục tiêu đồ án chỉ cần môi trường thử nghiệm, backend được chạy trên máy lập trình viên (local). Ngrok được sử dụng để tạo URL HTTPS công khai, cho phép frontend trên Vercel gọi API qua mạng. Điều này đảm bảo các chức năng đăng ký, đăng nhập, luyện phát âm, xử lý âm thanh, hội thoại AI và cơ chế gợi ý hoạt động thông suốt.

Cách kết hợp này đáp ứng yêu cầu kiểm thử end-to-end cho toàn bộ chức năng của hệ thống, đồng thời không phát sinh chi phí vận hành backend trên môi trường cloud.

4.2 Xây dựng backend phục vụ nghiệp vụ học ngoại ngữ

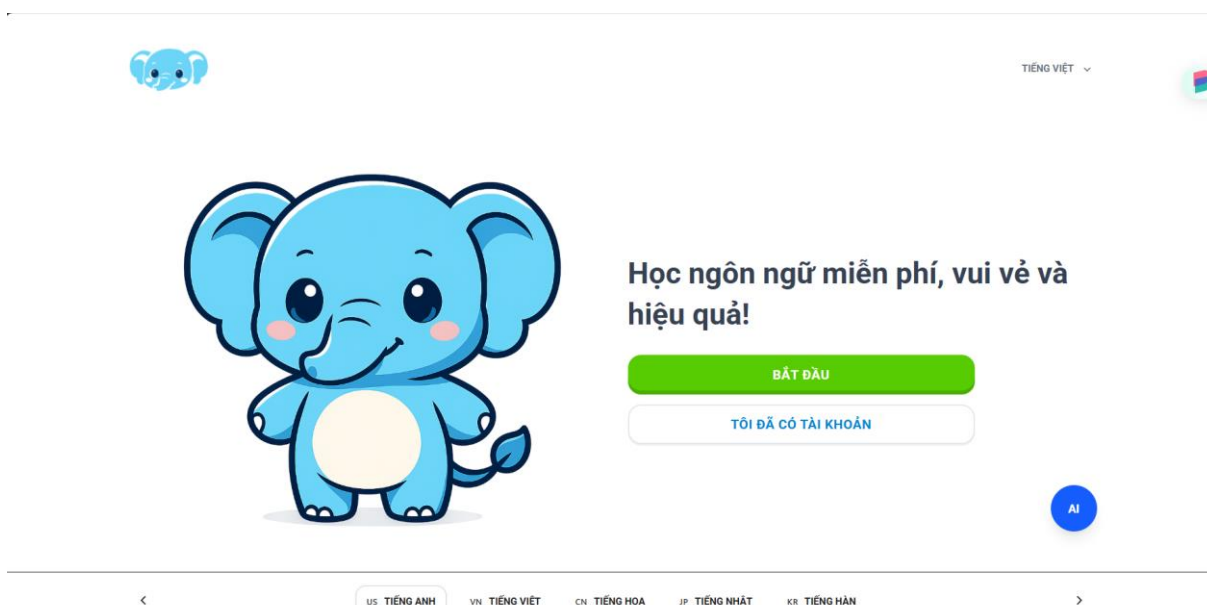
4.2.1 Cấu trúc dự án và module

Ứng dụng backend được xây dựng với Django/DRF, gồm nhiều *app/module* tương ứng chức năng nghiệp vụ riêng:

- users: Chịu trách nhiệm toàn bộ vòng đời tài khoản: đăng ký, đăng nhập, xác thực, lưu trữ thông tin hồ sơ và quản lý các cài đặt cá nhân của người dùng (ngôn ngữ giao diện, thông báo, độ khó, v.v.).
- languages: Quản lý danh sách ngôn ngữ hỗ trợ và việc ghi danh của người học vào từng ngôn ngữ (LanguageEnrollment), đồng thời tổ chức lộ trình học theo Topic, Lesson và lưu tiến độ từng chủ đề qua TopicProgress.
- learning: Tổ chức toàn bộ phần “làm bài” và luyện kỹ năng: định nghĩa Skill và các loại bài tập, quản lý phiên làm bài (LessonSession, SkillSession), lưu câu trả lời (SessionAnswer) và thống kê mức độ thành thạo kỹ năng của người học trong UserSkillStats.

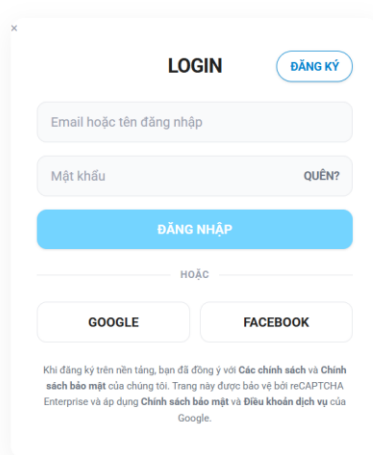
- **speech**: Đảm nhiệm chức năng luyện phát âm: quản lý các câu/từ cần luyện (PronunciationPrompt), gọi dịch vụ STT/TTS để sinh mẫu và chấm điểm, sau đó lưu lại kết quả chấm điểm phát âm trong PronAttempt.
- **roleplay**: Quản lý các kịch bản hội thoại (RoleplayScenario) và từng lượt thoại/khối nội dung (RoleplayBlock), cung cấp API cho chức năng chat hội thoại theo kịch bản và tích hợp với AI để sinh câu trả lời hoặc gợi ý trong bối cảnh hội thoại.
- **vocab**: Đóng vai trò kho từ vựng trung tâm: lưu Word cho từng ngôn ngữ, trạng thái của người dùng với từng từ qua KnownWord, và các bản dịch/giải nghĩa trong Translation để phục vụ ôn tập từ vựng.
- **social**: Hiện thực lớp mạng xã hội và gamification của hệ thống: quản lý quan hệ bạn bè (Friend), sự kiện/ lịch học (CalendarEvent), bảng xếp hạng (LeaderboardEntry), hệ thống huy hiệu (Badge, UserBadge) và các thông báo gửi tới người học (Notification).
- **tracking**: Ghi nhận chi tiết hành vi học tập và kết quả của người học: log từng tương tác học (LearningInteraction), lưu các lỗi cụ thể (Mistake) và các sự kiện XP (XPEvent, DailyXP), làm dữ liệu đầu vào cho thống kê và cho khối AI gợi ý.
- **ai_recommend**: Phụ trách toàn bộ phần gợi ý học cá nhân hóa: quản lý phiên bản mô hình AI (AIModelVersion), các lần huấn luyện (TrainingRun), snapshot đặc trưng đầu vào (FeatureSnapshot), danh sách gợi ý được sinh ra (Recommendation) và phản hồi thực tế của người dùng (FeedbackLoop) để cải thiện mô hình theo thời gian.
- Cách tổ chức trên cho phép làm việc độc lập trên từng phần chức năng, dễ dàng quản lý migration cho cơ sở dữ liệu, viết và chạy kiểm thử riêng cho từng module, đồng thời tái sử dụng và mở rộng code một cách linh hoạt khi hệ thống phát triển.

4.3 Giao diện hệ thống



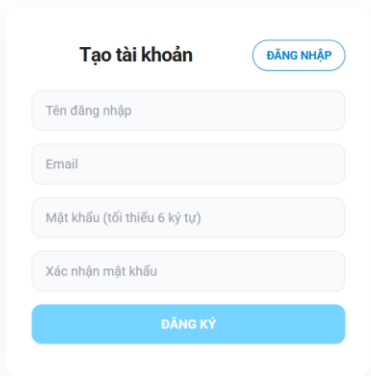
Hình 4. 8: Giao diện mở đầu

- Nút “BẮT ĐẦU”: chuyển người dùng tới trang đăng ký.
- Nút “TÔI ĐÃ CÓ TÀI KHOẢN”: dẫn tới trang đăng nhập.
- Cho phép chọn ngôn ngữ giao diện (Tiếng Việt, Tiếng Anh...).
- Dải chọn nhanh ngôn ngữ học (EN, VI, CN, JP, KR...).



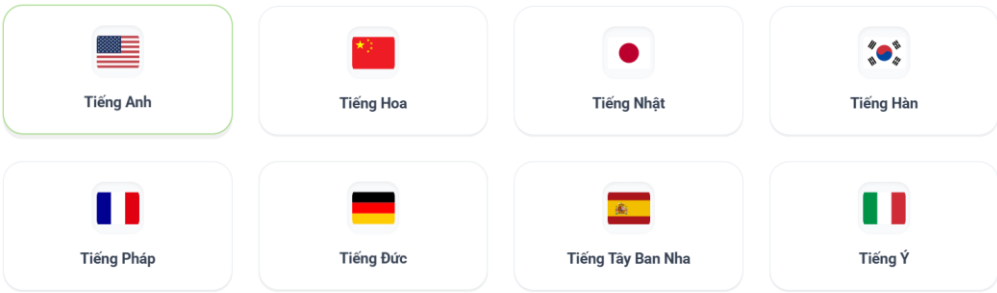
Hình 4. 9: Giao diện đăng nhập

Nhập tài khoản mật khẩu để đăng nhập vào hệ thống.



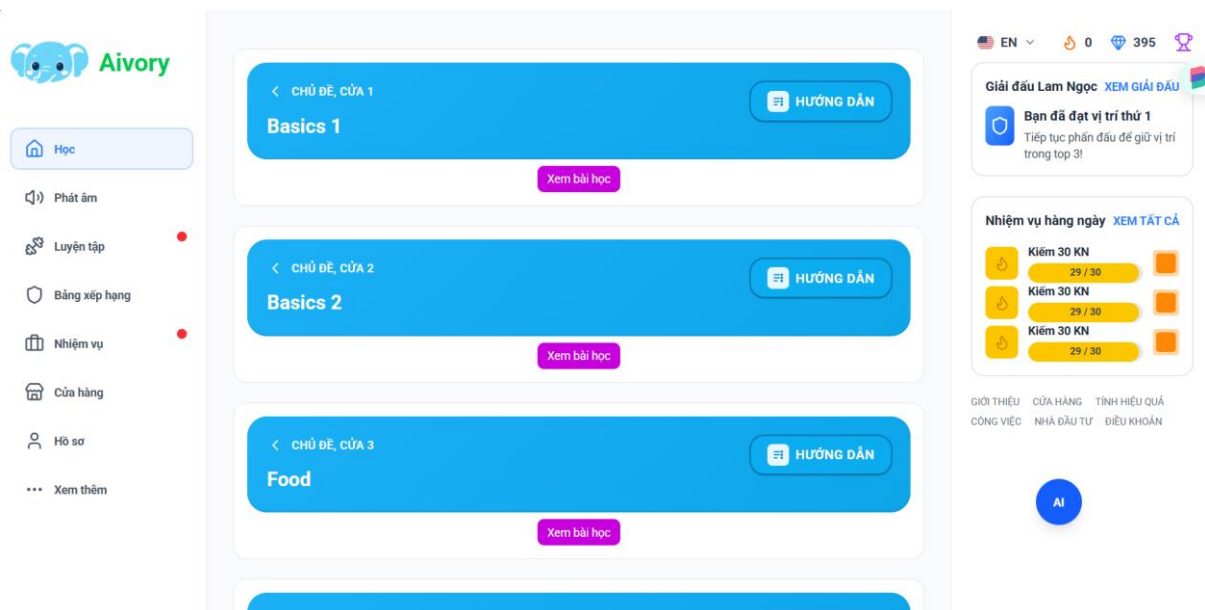
Hình 4. 10: Giao diện đăng kí

Nhập tên đăng kí tài khoản email và mật khẩu để đăng kí vào hệ thống



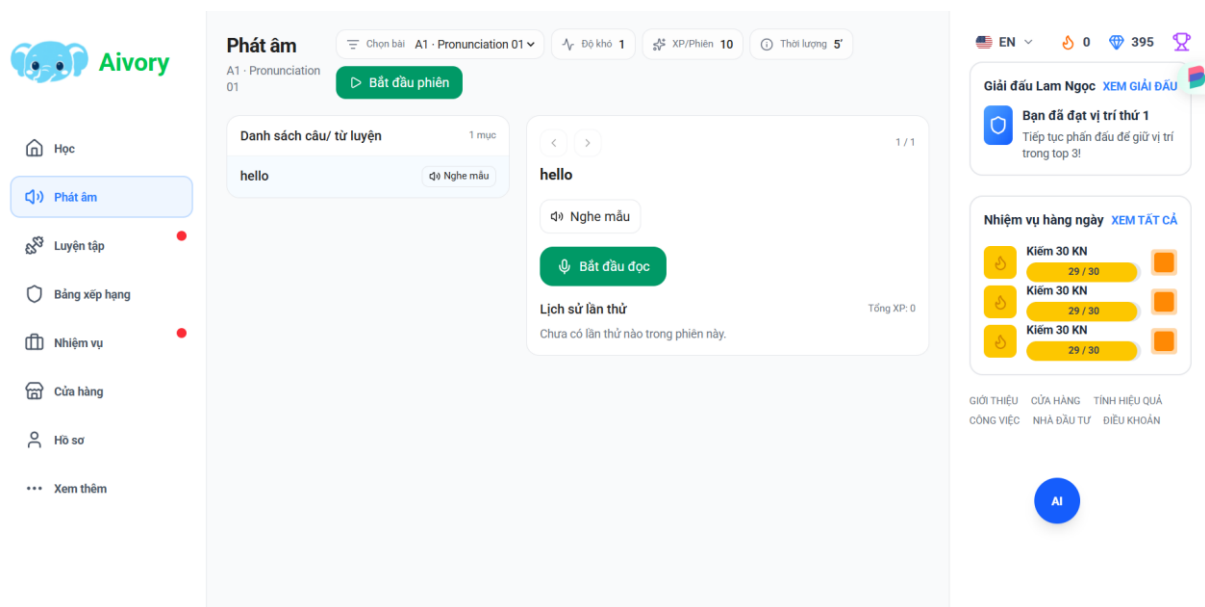
Hình 4. 11: Giao diện đăng kí ngôn ngữ mới

Chọn 1 ngôn ngữ bất kì để bắt đầu vào học.



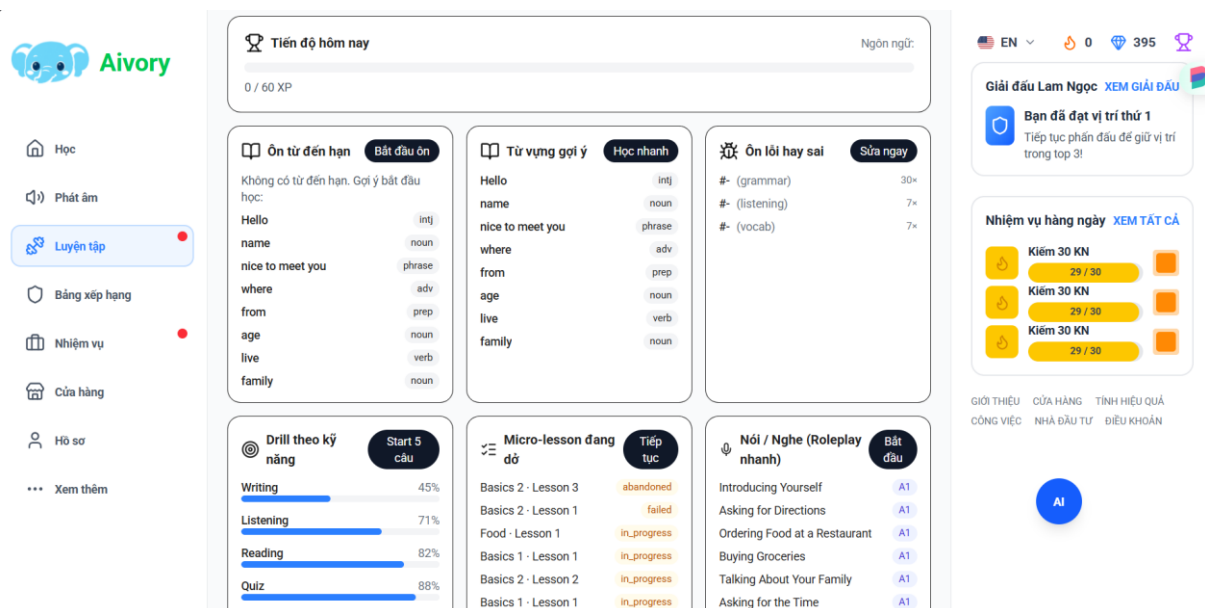
Hình 4. 12: Giao diện bài học

Trang học phân theo chủ đề có thể chọn bài học đầu tiên của các chủ đề khác nhau các bài học sau sẽ được khóa lại hoàn thành bài học trước sẽ mở khóa bài học sau.



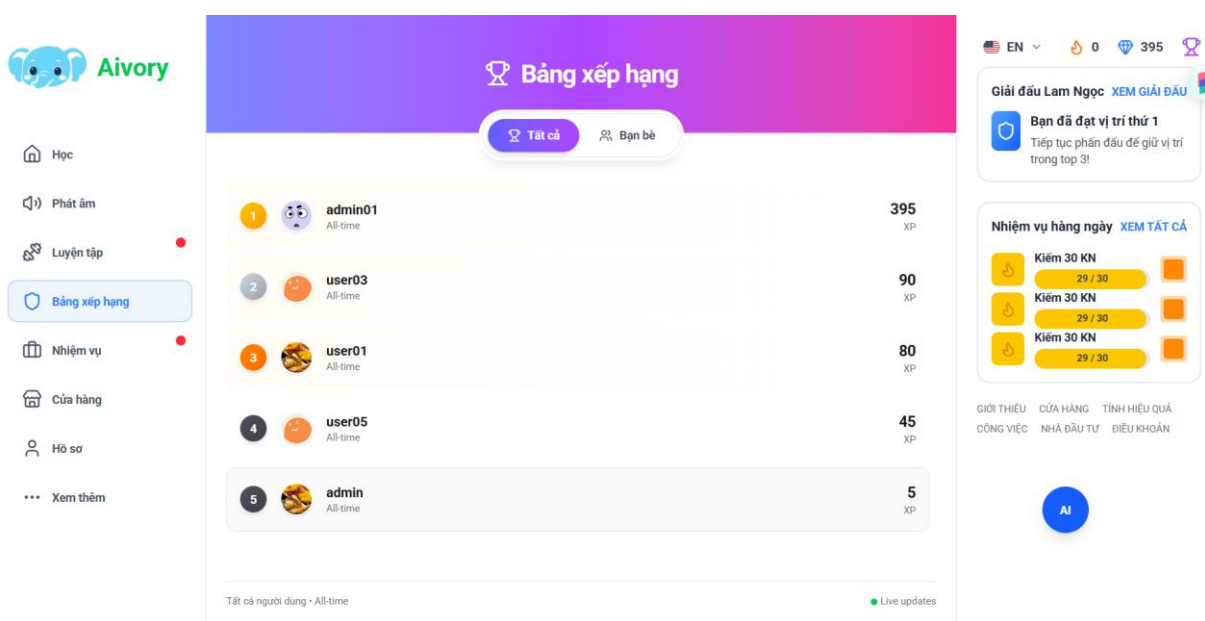
Hình 4. 13: Giao diện luyện phát âm

Hệ thống nhận giọng nói của bạn để chấm điểm phát âm và cho phép người dùng nghe phát âm mẫu kèm danh sách các bài học.



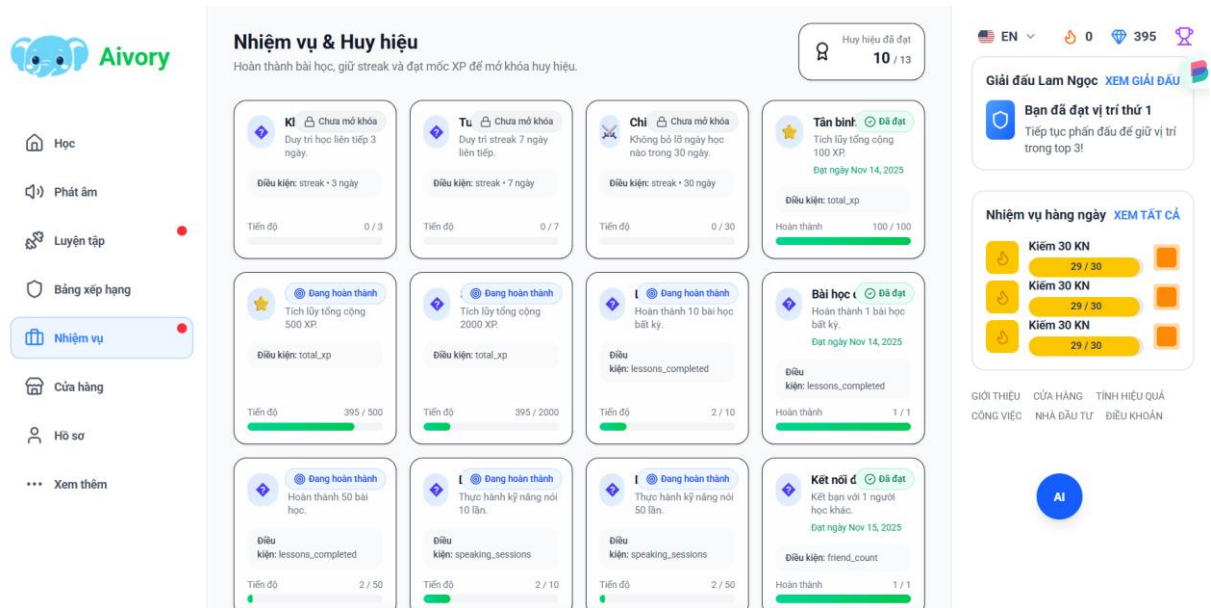
Hình 4. 14: Giao diện luyện tập, thống kê của user

Thống kê điểm kinh nghiệm của ngày hôm nay, những từ sai, kỹ năng sai, phiên học đã học.



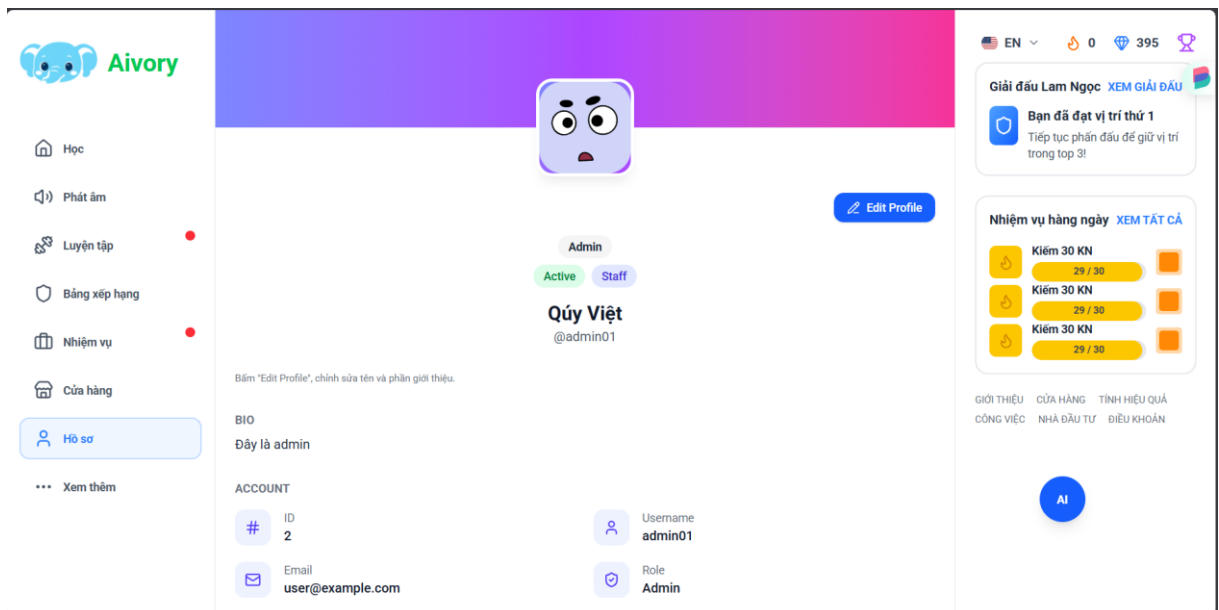
Hình 4. 15: Giao diện bảng xếp hạng

Cập nhật thời gian thực bảng xếp hạng tất cả và bảng xếp hạng bạn bè theo điểm kinh nghiệm tính theo toàn bộ kinh nghiệm các ngôn ngữ đã học.



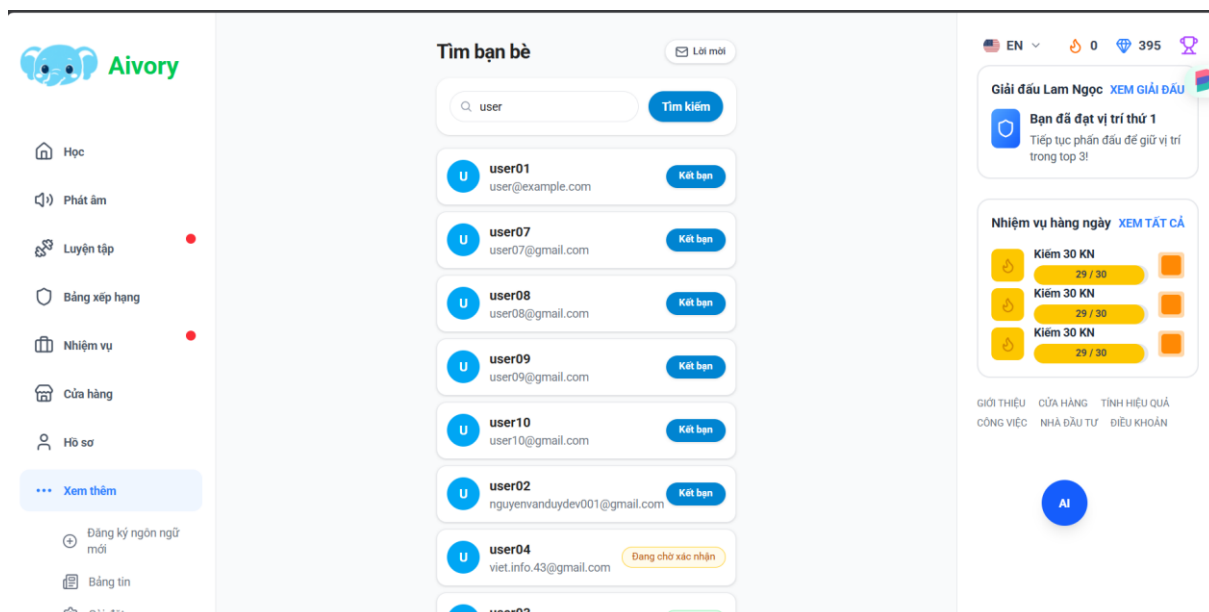
Hình 4. 16: Giao diện Nhiệm vụ và huy hiệu

Tính nhiệm vụ hoàn thành của người dùng để trao huy hiệu.



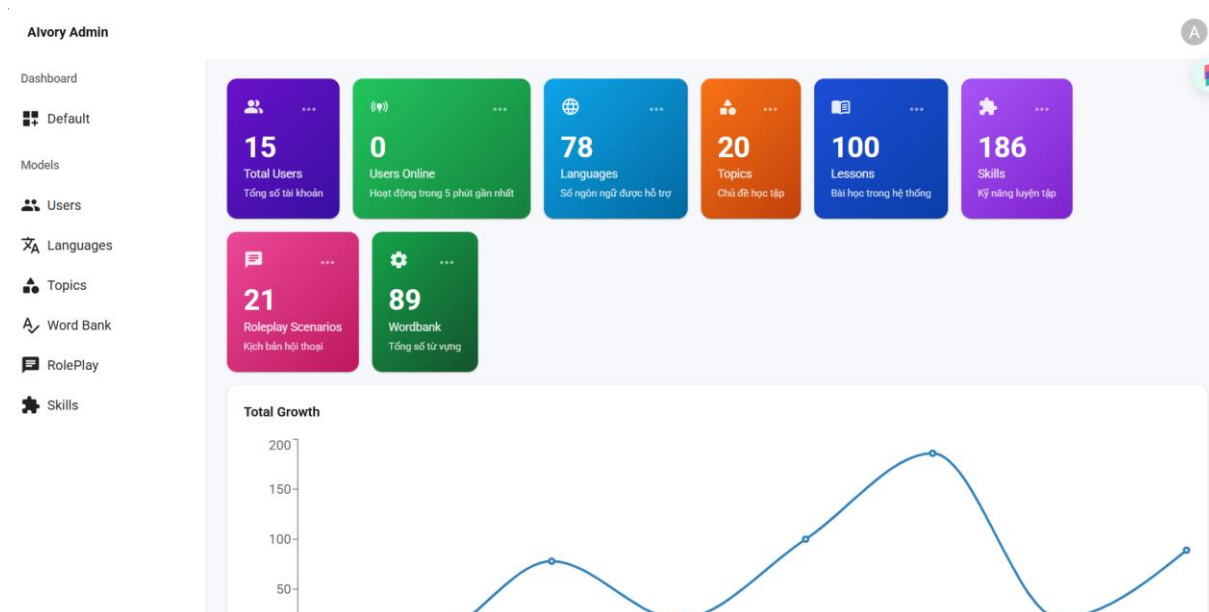
Hình 4. 17: Giao diện hồ sơ người dùng

Thông tin của người dùng đã đăng nhập hệ thống kèm chức năng chỉnh sửa.



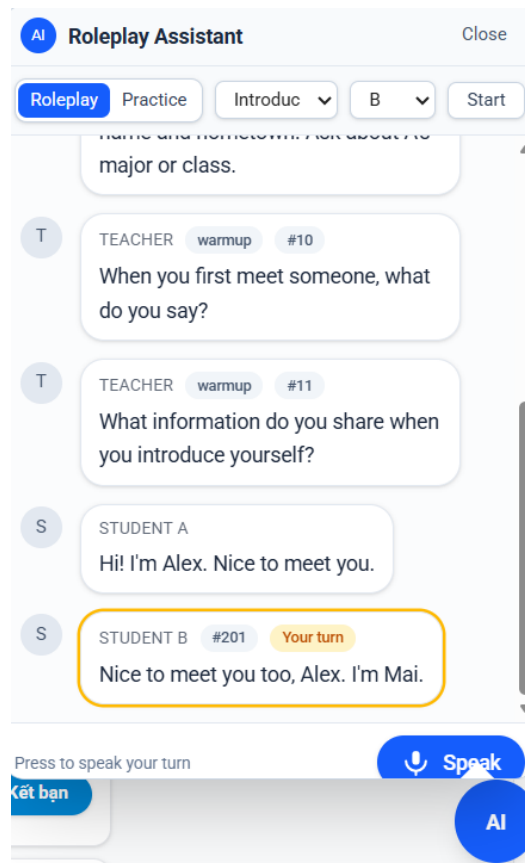
Hình 4. 18: Giao diện bạn bè

Kết nối bạn bè gửi lời mời kết bạn, tìm kiếm, xem yêu cầu kết bạn.



Hình 4. 19: Giao diện admin

Bao gồm quản lý các nội dung học, quản lý người dùng, kịch bản hội thoại, ...



Hình 4. 20: Bong bóng hội thoại

Luyện nghe-nói với chatbot roleplay theo 2 chế độ luyện tập hoặc theo kịch bản sẵn có.

KẾT LUẬN

1. Kết quả đạt được

1.1. Về mặt mô hình và kiến trúc hệ thống

Đề tài đã xây dựng thành công một kiến trúc tổng thể cho hệ thống học ngoại ngữ ứng dụng AI với các thành phần hoạt động rõ ràng, có khả năng mở rộng và dễ dàng tích hợp công nghệ mới. Hệ thống bao gồm:

- Mô-đun luyện phát âm thông minh, ứng dụng mô hình Whisper cho nhận dạng giọng nói, tính toán các chỉ số WER, CER, thời gian phát âm và đánh giá theo từng từ, từng âm vị.
- Mô-đun hội thoại roleplay AI, cho phép người học tương tác với mô hình ngôn ngữ lớn (LLM) theo bối cảnh thực tế như phỏng vấn, nhà hàng, sân bay, mua sắm. AI đóng vai trò bạn hội thoại, đưa tình huống, phản hồi theo ngữ cảnh, sửa lỗi và gợi ý diễn đạt tốt hơn.
- Hệ thống đề xuất nội dung học tập, sử dụng vector embedding, truy vấn tương tự (vector search) và hành vi người học để đưa ra gợi ý bài học, từ vựng, hội thoại phù hợp.
- Gamification đa tầng, bao gồm nhiệm vụ hằng ngày, bảng xếp hạng, hệ thống điểm kinh nghiệm (XP), cấp độ (level), huy hiệu, giúp tăng động lực và duy trì thời gian học.
- Hệ quản trị backend trên nền Django REST Framework, tích hợp Redis, PostgreSQL, MinIO, Celery, giúp xử lý dữ liệu, quản lý người dùng, xử lý âm thanh và tương tác với mô hình AI.
- Frontend xây dựng bằng ReactJS Vite, tối ưu tốc độ tải trang, hỗ trợ đa màn hình, đảm bảo trải nghiệm mượt mà và trực quan.

1.2. Về mặt thực nghiệm AI

Đề tài đã triển khai và đánh giá nhiều thí nghiệm phục vụ bài toán học ngoại ngữ:

- Thực nghiệm TTS (gTTS, Piper)
 - Đánh giá chất lượng âm thanh, độ tự nhiên, tốc độ sinh giọng.
 - Từ đó chọn mô hình phù hợp cho luyện nghe nâng cao.

- Thực nghiệm ASR (Whisper)
 - Đo WER, CER trong nhiều điều kiện: tốc độ nói khác nhau, có nền nhiễu, giọng địa phương.
 - So sánh giữa các phiên bản Whisper (tiny, base, small).
- Thực nghiệm hội thoại AI
 - Đánh giá độ phù hợp ngữ cảnh.
 - Mức độ duy trì logic hội thoại.
 - Khả năng sửa lỗi và gợi ý diễn đạt.
- Thực nghiệm gợi ý học tập
 - Kiểm tra vector similarity giữa nội dung bài học và lịch sử người học.
 - Đánh giá khả năng cá nhân hóa.

Các thực nghiệm cho thấy nền tảng đề xuất hoạt động ổn định và phù hợp để mở rộng thành sản phẩm thực tế.

1.3. Về mặt dữ liệu và nội dung học tập

Hệ thống đã xây dựng được một kho dữ liệu gồm:

- Từ vựng theo chủ đề.
- Bài học theo cấu trúc CEFR cơ bản (A1–A2).
- Hội thoại kịch bản roleplay.
- Ghi nhận lỗi phát âm, lỗi từ vựng, thống kê hành vi học.

Kho dữ liệu này đóng vai trò trung tâm cho các mô-đun gợi ý và RAG.

1.4. Về tính khả thi và tiềm năng ứng dụng

Kết quả thực nghiệm cho thấy:

- Hệ thống có thể hoạt động hoàn chỉnh từ đầu đến cuối (end-to-end).
- Các mô-đun AI mang lại giá trị rõ rệt, đặc biệt trong luyện phát âm và hội thoại roleplay.
- Kiến trúc microservice hóa có khả năng mở rộng cho hàng ngàn người dùng đồng thời nếu triển khai thực tế.

Đây là nền tảng quan trọng để phát triển hệ thống thành một sản phẩm hoàn chỉnh.

2. Hạn chế của đề tài

Dù đạt được nhiều kết quả đáng ghi nhận, đề tài vẫn tồn tại một số hạn chế xuất phát từ thời gian, nguồn lực và điều kiện triển khai:

2.1. Giới hạn về nội dung và phạm vi ngôn ngữ

- Nội dung bài học mới dừng ở mức cơ bản (A1–A2).
- Chưa tích hợp nhiều ngôn ngữ khác ngoài tiếng Anh.
- Chưa có ngân hàng bài tập đa dạng như nghe – nói – đọc – viết đầy đủ.

2.2. Giới hạn về mô hình AI

- Whisper chưa được fine-tune cho giọng người Việt nói tiếng Anh, nên độ chính xác chưa tối ưu.
- Hội thoại roleplay dựa hoàn toàn vào LLM, đôi khi trả lời không đúng mục tiêu sư phạm khi thiếu RAG chuyên biệt.
- TTS (gTTS, Piper) chưa đạt độ tự nhiên như các mô hình tiên tiến như VITS, FastSpeech2.

2.3. Hiệu năng và tối ưu hóa

- Việc xử lý âm thanh thời gian thực còn chậm khi không chạy GPU.
- Tốc độ phản hồi của LLM phụ thuộc vào mạng và tài nguyên máy chủ.
- Chưa triển khai cân bằng tải hoặc hệ thống queue cho tác vụ nặng.

2.4. Trải nghiệm người dùng

- Hiện chưa có bản ứng dụng di động (Android/iOS).

3. Hướng phát triển

Để hệ thống hoàn thiện và hướng đến sử dụng thực tế, một số hướng phát triển quan trọng bao gồm:

3.1. Mở rộng nội dung học

- Bổ sung đầy đủ bài học theo CEFR từ A1 đến C1.
- Xây dựng bộ bài tập nghe, đọc, viết nâng cao.
- Tích hợp nhiều ngôn ngữ: Hàn, Trung, Nhật.ss

3.2. Nâng cấp và tối ưu mô hình AI

- Fine-tune Whisper với tập dữ liệu người Việt.
- Tích hợp TTS tiên tiến: VITS, FastSpeech2, Coqui AI.
- Bổ sung RAG cho roleplay để AI bám sát giáo trình.
- Triển khai recommendation system nâng cao như BERT4Rec, SASRec.

3.3. Tối ưu hiệu năng hệ thống

- Chạy mô hình ASR/TTS/LLM qua GPU server.
- Hoàn thiện cơ chế monitoring, logging, caching.

3.4. Mở rộng nền tảng sử dụng

- Phát triển ứng dụng di động đa nền tảng (React Native hoặc Flutter).
- Tích hợp chế độ học ngoại tuyến (offline mode) cho bài tập từ vựng.
- Tích hợp WebRTC hoặc streaming để luyện nói trực tiếp theo thời gian thực.

3.5. Tích hợp phân tích dữ liệu học tập

- Phân tích lỗi phát âm theo âm vị.
- Mô hình dự đoán khả năng bỏ học (dropout prediction).

3.6. Hướng tới thương mại hóa

- Xây dựng mô-đun thanh toán.
- Tạo gói tài khoản cá nhân, nhóm, doanh nghiệp hoặc trường học.
- Xây dựng mô hình kinh doanh Freemium hoặc Subscription.

TÀI LIỆU THAM KHẢO

- [1] P. Kaur, H. Kumar, and S. Kaushal, “Technology-assisted language learning adaptive systems: A comprehensive review,” *International Journal of Cognitive Computing in Engineering*, vol. 4, pp. 301–313, Jun. 2023. [Paper]
- [2] Y. Liu, F. B. Ab Rahman, and F. B. Mohamad Zain, “A systematic literature review of research on automatic speech recognition in EFL pronunciation,” *Cogent Education*, vol. 12, no. 1, 2025. [Paper]
- [3] R. Shadiev and J. Liu, “Review of research on applications of speech recognition technology to assist language learning,” *ReCALL*, vol. 35, no. 1, pp. 74–88, Jan. 2023. [Paper]
- [4] R. Vesselinov and J. Grego, *Duolingo Effectiveness Study: Final Report*, City University of New York, 2012. [Paper] [PDF]
- [5] T. T. T. Nguyen, “Enhancing non-English major students’ speaking skills...,” *TNU Journal of Science and Technology*, vol. 230, no. 04, pp. 75–83, 2025. [Paper]
- [6] M. He, B. N. Abbasi, and J. He, “AI-driven language learning in higher education...,” *Humanities and Social Sciences Communications*, vol. 12, art. 1525, 2025. [Paper]
- [7] Y. Doğan and T. Talan, “Artificial intelligence in foreign language learning: A bibliometric analysis,” *Journal of Pedagogical Research*, vol. 9, no. 2, 2025. [Paper]
- [8] S. Souabi et al., “Recommendation systems on e-learning and social learning,” *Electronic Journal of e-Learning*, vol. 19, no. 5, 2021. [Paper]
- [9] B. Smith, “The effectiveness of Duolingo’s Spanish course...,” *CALICO Journal*, vol. 41, no. 2, 2024. [Paper]
- [10] J. Loewen et al., “The effectiveness of app-based language instruction...,” *Foreign Language Annals*, vol. 53, no. 2, 2020. [Paper]
- [11] F. Ahmed and S. Riaz, “A comparative analysis of effectiveness...,” *Qurtuba Journal of Social Sciences*, vol. 16, no. 1, 2025. [Paper]
- [12] G. F. Baker and E. Jensen, “Investigating mobile-assisted language learning apps...,” *Journal of Curriculum and Teaching*, vol. 13, no. 2, 2024. [Paper]

- [13] S. Sunandau, “A comparative analysis of Duolingo, Babbel, and ELSA Speak...,” *Language in India*, vol. 24, no. 10, 2024. [Paper]
- [14] P. T. T. Tiên, “The situation of using ELSA Speak application...,” *Tạp chí Khoa học Giáo dục*, 2024. [Paper]
- [15] T. B. Brown et al., “Language models are few-shot learners,” *NeurIPS* 2020. [Paper]
- [16] OpenAI, “GPT-4 technical report,” arXiv:2303.08774, 2023. [Paper]
- [17] C. K. Lo et al., “Exploring the application of ChatGPT in ESL/EFL education...,” *Smart Learning Environments*, vol. 11, art. 50, 2024. [Paper]
- [18] R. Anil et al., “Gemini: a family of highly capable multimodal models,” arXiv:2312.11805, 2023. [Paper]
- [19] S. Deterding et al., “From game design elements to gamefulness...,” *MindTrek Academic Conf.*, 2011. [Paper]
- [20] J. Shen et al., “Natural TTS synthesis by conditioning WaveNet...,” *ICASSP* 2018. [Paper]
- [21] D. Jurafsky and J. Martin, *Speech and Language Processing*, 3rd ed., Draft, 2023. [Book] [PDF]
- [22] A. Reimers and I. Gurevych, “Sentence-BERT: Sentence embeddings using Siamese BERT networks,” *EMNLP*, 2019. [Paper]
- [23] A. S. Cordova and M. Lepper, “Intrinsic motivation and the role of interactive simulations and role-playing...,” *Journal of Educational Psychology*, vol. 88, no. 2, pp. 329–337, 1996. [Paper]
- [24] A. Baeovski et al., “wav2vec 2.0: A framework for self-supervised learning...,” arXiv:2006.11477. [Paper]
- [25] A. Gulati et al., “Conformer: Convolution-augmented Transformer for Speech Recognition,” arXiv:2005.08100. [Paper]
- [26] A. Vaswani et al., “Attention Is All You Need,” *NeurIPS*, 2017. [Paper]
- [27] I. Sutskever et al., “Sequence-to-Sequence Learning with Neural Networks,” *NeurIPS*, 2014. [Paper]
- [28] OpenAI, “Whisper paper,” available at: <https://cdn.openai.com/papers/whisper.pdf>. [Paper] [PDF]

- [29] OpenAI, “Giới thiệu Whisper (Introducing Whisper)”. [Website]
- [30] “FLEURS dataset paper,” arXiv:2205.12446. [Paper]
- [31] GeeksforGeeks, “Tokenization in NLP”. [Website]
- [32] GeeksforGeeks, “POS Tagging”. [Website]
- [33] GeeksforGeeks, “Constituency & Dependency Parsing”. [Website]
- [34] GeeksforGeeks, “Introduction to Parsers”. [Website]
- [35] NLP-progress, “Grammatical Error Correction”. [Website]
- [36] Microsoft Learn, “SSML Overview”. [Website]
- [37] “Free English Lessons by CEFR levels”. [Website]
- [38] IBM, “What is Cosine Similarity?”. [Website]
- [39] GeeksforGeeks, “Cosine Similarity”. [Website]
- [40] “Jaccard Index,” arXiv:2110.09619. [Paper]
- [41] “Retrieval-Augmented Generation: A Survey,” arXiv:2402.19473. [Paper]
- [42] Rhasspy Project, “Piper TTS”. [Website]
- [43] Google Cloud, “Google Text-to-Speech API”. [Website]
- [44] Ollama, “Ollama Documentation” (ollama.com). [Website]
- [45] GitHub, “pgvector Documentation”. [Website]
- [46] Redis.io, “Redis Documentation”. [Website]
- [47] Django Project, “Django Documentation”. [Website]
- [48] Django REST Framework, “DRF Documentation”. [Website]
- [49] React.dev, “React Documentation”. [Website]
- [50] CEFR Website, “CEFR Guidelines”. [Website]
- [51] P. Wozniak, “Theoretical aspects of spaced repetition in learning,” SuperMemo, 1990. [Paper]
- [52] B. Woolf, *Building Intelligent Tutoring Systems*, Morgan Kaufmann, 2009. [Book]
- [53] R. Winkler and M. Söllner, “Unleashing the potential of chatbots in education,” Electr. Markets, 2021. [Paper]
- [54] Vũ Hữu Tiệp, “Machine learning cơ bản”. [Website] [PDF]

- [55] Sarwar, B. et al., “Item-Based Collaborative Filtering Recommendation Algorithms,” WWW’01. [Paper] [PDF]
- [56] Koren, Y., Bell, R., Volinsky, C., “Matrix Factorization Techniques for Recommender Systems,” IEEE Computer, 2009. [Paper]
- [57] Pazzani, M., Billsus, D., “Content-Based Recommendation Systems,” in *The Adaptive Web*, Springer, 2007. [Paper]
- [58] Burke, R., “Hybrid Recommender Systems: Survey and Experiments,” 2002; and “Hybrid Web Recommender Systems,” in *The Adaptive Web*, 2007. [Paper]
- [59] Murre, J. & Dros, J., “Replication and Analysis of Ebbinghaus’ Forgetting Curve,” PLoS ONE, 2015; with meta-analysis by Cepeda et al., Psychological Bulletin, 2006. [Paper]

PHỤ LỤC 1

PHỤ LỤC 2

{nếu có, bold, size 14}

{Font: Time New Roman; thường; cỡ chữ: 12; dẫn dòng: 1,3; căn lề: justified}