

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



BÁO CÁO ĐỒ ÁN TỐT NGHIỆP
XÂY DỰNG HỆ THỐNG LUYỆN NÓI TIẾNG ANH THEO
PHƯƠNG PHÁP SHADOWING KẾT HỢP CÔNG NGHỆ ASR, MÔ
HÌNH BERT TRONG NHẬN DẠNG GIỌNG NÓI, ĐÁNH GIÁ
TRÌNH ĐỘ CÂU TIẾNG ANH THEO CEFR

Giảng viên hướng dẫn	TH.S PHẠM THỊ MIÊN
Sinh viên thực hiện	NGUYỄN ĐĂNG QUÝ
Lớp	CQ.62.CNTT
MSSV	6251071076
Khoa	62

Tp.Hồ Chí Minh, năm 2025

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



BÁO CÁO ĐỒ ÁN TỐT NGHIỆP
XÂY DỰNG HỆ THỐNG LUYỆN NÓI TIẾNG ANH THEO
PHƯƠNG PHÁP SHADOWING KẾT HỢP CÔNG NGHỆ ASR, MÔ
HÌNH BERT TRONG NHẬN DẠNG GIỌNG NÓI, ĐÁNH GIÁ
TRÌNH ĐỘ CÂU TIẾNG ANH THEO CEFR

Giảng viên hướng dẫn	TH.S PHẠM THỊ MIÊN
Sinh viên thực hiện	NGUYỄN ĐĂNG QUÝ
Lớp	CQ.62.CNTT
MSSV	6251071076
Khoa	62

Tp.Hồ Chí Minh, năm 2025

NHIỆM VỤ THIẾT KẾ TỐT NGHIỆP
BỘ MÔN: CÔNG NGHỆ THÔNG TIN

-----***-----

Mã sinh viên: 6251071076

Họ tên SV: Nguyễn Đăng Quý

Khóa: 62

Lớp: Công nghệ thông tin

1. Tên đề tài

Xây dựng hệ thống luyện nói tiếng anh theo phương pháp shadowing kết hợp ứng dụng công nghệ ASR và nghiên cứu mô hình BERT trong việc nhận dạng giọng nói và tự động đánh giá trình độ câu tiếng anh theo CEFR

2. Mục đích, yêu cầu

a. Mục đích

- Xây dựng một ứng dụng mobile hỗ trợ luyện nói tiếng Anh qua video YouTube, kết hợp công nghệ nhận dạng giọng nói và mô hình ngôn ngữ hiện đại.
- Giúp người học luyện phát âm và phản xạ tiếng Anh thông qua phương pháp shadowing, tăng khả năng giao tiếp thực tế.
- Tận dụng nguồn nội dung phong phú từ YouTube để học tiếng Anh một cách tự nhiên, linh hoạt, không gò bó vào sách vở.
- Cung cấp trải nghiệm học tập đơn giản, trực quan, dễ sử dụng trên thiết bị di động.

b. Yêu cầu

- Cho phép người dùng nhập link hoặc ID video YouTube để hệ thống tự xử lý video.
- Tự động lấy transcript từ audio video bằng ASR (Whisper).

- Tách câu, đánh giá độ khó theo chuẩn CEFR (A1 - C2) bằng BERT/Sentence-BERT.
- Hiển thị video kèm phụ đề trực tiếp, có thể xem bản dịch và cách phát âm từng từ.
- Cho phép luyện nói từng câu: nghe – lặp lại – ghi âm – chấm điểm phát âm.
- So sánh âm thanh người dùng với bản gốc, chỉ ra lỗi phát âm và gợi ý cải thiện.
- Đảm bảo giao diện đơn giản, dễ dùng trên mobile, không yêu cầu đăng nhập phức tạp.
- Đảm bảo an toàn dữ liệu cá nhân.

3. Nội dung và phạm vi đề tài

a. Nội dung

- Xây dựng ứng dụng hỗ trợ luyện nói tiếng Anh bằng cách nhập link video YouTube, hệ thống sẽ lấy transcript, chia câu, đánh giá độ khó, cho phép người dùng luyện tập và chấm điểm phát âm.
- Kết hợp công nghệ nhận dạng giọng nói Whisper và mô hình ngôn ngữ BERT để xử lý nội dung và đánh giá trình độ CEFR.
- Cho phép người học thực hành theo phương pháp shadowing và nhận phản hồi ngay lập tức.

b. Phạm vi

- Phát triển phiên bản mobile đơn giản phục vụ người học cuối.
- Phiên bản web chỉ dùng để nhập và chia sẻ link YouTube.
- Chỉ hỗ trợ video tiếng Anh, chưa xử lý ngôn ngữ khác hoặc hội thoại dài, ngữ điệu phức tạp.
- Tập trung luyện nói từng câu đơn lẻ, chưa hỗ trợ bài hội thoại.
- Không yêu cầu hệ thống login phức tạp, tập trung vào trải nghiệm học tập.

4. Công nghệ, công cụ và ngôn ngữ lập trình

- Ngôn ngữ: Golang, Python, TypeScript, JavaScript

- Công nghệ: Whisper, BERT/Sentence-BERT, PostgreSQL, yt-dlp, Azure Translator, React Native, FastAPI, gRPC, NextJS, ReactJS
- Công cụ: Visual Studio Code, Fulltext Search, Dictionary Opensource API, Golang Echo, Gorm

5. Các kết quả chính dự kiến sẽ đạt được và ứng dụng

- Xây dựng ứng dụng học nói tiếng Anh qua video YouTube với giao diện thân thiện.
- Tích hợp công nghệ ASR để ghi âm, phân tích và chấm điểm phát âm người học.
- Áp dụng thành công mô hình BERT để đánh giá độ khó từng câu theo chuẩn CEFR.
- Cung cấp trải nghiệm học nói tự nhiên, dễ tiếp cận, phù hợp với mọi trình độ.
- Có khả năng mở rộng thêm các tính năng như luyện hội thoại, thống kê kết quả, hoặc tích hợp game hóa để tăng động lực học.
- Ứng dụng phù hợp với người tự học, giáo viên, trung tâm Anh ngữ, và học mọi lúc mọi nơi trên thiết bị di động.

6. Giáo viên và cán bộ hướng dẫn

Họ tên: ThS. PHẠM THỊ MIÊN

Đơn vị công tác: Bộ môn Công Nghệ Thông Tin – Trường Đại học Giao thông Vận tải phân hiệu TP HCM.

Điện thoại: 0961 170 638

Email: ptmien@utc2.edu.vn

Ngày tháng 03 năm 2025

Đã giao nhiệm vụ TKTN

Trưởng BM Công nghệ Thông tin

Giáo viên hướng dẫn

ThS. TRẦN PHONG NHÃ

ThS. PHẠM THỊ MIÊN

Đã nhận nhiệm vụ TKTN

Sinh viên: Nguyễn Đăng Quý

Ký tên:

Điện thoại: 0869 960 852

Email: 6251071076@st.utc2.edu.vn

LỜI CẢM ƠN

Lời đầu tiên, em xin được bày tỏ lòng biết ơn sâu sắc và lời chúc sức khỏe chân thành nhất. Nhờ sự quan tâm, chỉ dẫn và dùi dắt tận tình của Quý thầy cô trong suốt thời gian học tập, em đã có thể hoàn thành xuất sắc chương trình học và đồ án tốt nghiệp của mình.

Để hoàn thành đồ án này, bên cạnh sự nỗ lực không ngừng nghỉ của bản thân trong việc trau dồi kiến thức, em còn may mắn nhận được rất nhiều kiến thức quý báu từ sự chỉ bảo của Quý thầy cô trong suốt bốn năm qua. Đặc biệt, em xin gửi lời tri ân sâu sắc tới giảng viên ThS. Phạm Thị Miên, người đã tận tình định hướng và truyền đạt những kiến thức, kỹ năng vô giá, giúp em có đủ hành trang để hoàn thành đồ án tốt nghiệp này.

Mặc dù đã cố gắng hết sức để hoàn thiện đồ án một cách chỉnh chu nhất, em tin rằng sẽ khó tránh khỏi những thiếu sót. Em rất mong nhận được những ý kiến đánh giá và góp ý quý báu từ Quý thầy cô để em có thể rút ra bài học, tích lũy thêm kinh nghiệm cho bản thân, từ đó hoàn thiện hơn trong chặng đường sắp tới.

Sau cùng, em xin chân thành cảm ơn Quý thầy cô vì tất cả những gì thầy cô đã dành cho em trong suốt thời gian qua. Em xin kính chúc Quý thầy cô trong Bộ môn Công nghệ Thông tin, đặc biệt là cô ThS. Phạm Thị Miên, luôn dồi dào sức khỏe, tràn đầy năng lượng và gặp nhiều may mắn trong cuộc sống. Em hy vọng Quý thầy cô sẽ tiếp tục gặt hái được nhiều thành công hơn nữa, không chỉ trong sự nghiệp giảng dạy mà còn trong mọi lĩnh vực mà thầy cô theo đuổi. Những thành tựu của thầy cô chính là nguồn động viên và cảm hứng to lớn cho thế hệ sinh viên chúng em.

Em xin chân thành cảm ơn!

Tp. Hồ Chí Minh, ngày ... tháng 06 năm 2025

Sinh viên thực hiện

(Ký và ghi họ và tên)

Nguyễn Đăng Quý

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

Tp. Hồ Chí Minh, ngày tháng 06 năm 2025

Giảng viên hướng dẫn

ThS. Phạm Thị Miên

NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN

Tp. Hồ Chí Minh, ngày tháng 06 năm 2025

Giảng viên phản biện

MỤC LỤC

NHIỆM VỤ THIẾT KẾ TỐT NGHIỆP	i
LỜI CẢM ƠN	iv
NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN	v
NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN	vi
MỤC LỤC	vii
DANH MỤC CHỮ VIẾT TẮT	x
DANH MỤC HÌNH ẢNH VÀ BẢNG BIỂU	xi
TỔNG QUAN	1
1. Lý do chọn đề tài.....	1
2. Mục đích nghiên cứu.....	1
3. Phương pháp nghiên cứu.....	2
4. Phạm vi nghiên cứu.....	3
5. Kết quả dự kiến đề tài	4
6. Cấu trúc cuốn báo cáo	6
CHƯƠNG 1. CƠ SỞ LÝ THUYẾT	7
1.1. Golang (Go)	7
1.1.1. Tổng quan về ngôn ngữ lập trình Golang.....	7
1.1.2. Framework Echo	8
1.1.3. Thư viện GORM	8
1.2. Python	9
1.2.1. Tổng quan về ngôn ngữ lập trình Python	9
1.2.2. Framework FastAPI	9
1.3. Tổng quan về ngôn ngữ TypeScript.....	10
1.4. JavaScript.....	10
1.4.1. Tổng quan về ngôn ngữ JavaScript	10
1.4.2. Next.js	11
1.4.3. ReactJS	12
1.4.4. React Native	12
1.5. Các thành phần trí tuệ nhân tạo và ngôn ngữ.....	13

1.5.1. Whisper	13
1.5.2. NLP (Natural Language Processing).....	13
1.5.3. BERT	14
1.5.4. gRPC	15
1.5.5. yt-dlp	16
1.5.6. Free Dictionary API	16
1.5.7. Azure Translator và Dictionary API	17
1.6. Tổng kết	17
CHƯƠNG 2. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG	18
2.1. Phân tích hệ thống.....	18
2.2. Sơ đồ phân cấp chức năng (BFD) của hệ thống Shadowing	20
2.3. Mô hình ERD.....	21
2.4. Biểu đồ Usecase.....	22
2.4.1. Sơ đồ Usecase tổng quát	22
2.4.2. Usecase Khởi tạo ID và Nhận diện thiết bị	23
2.4.3. Usecase Xem danh sách video	24
2.4.4. Usecase Import video, xử lý transcript và Phân tích CEFR.....	25
2.4.5. Usecase Yêu thích video	26
2.4.6. Usecase bookmark.....	26
2.4.7. Usecase Hiển thị transcript và Bộ điều khiển LiveSubs	27
2.4.8. Usecase xem translation	28
2.4.9. Usecase Xem phát âm và định nghĩa từ vựng (FreeDictionary API).....	28
2.4.10.Usecase Tìm kiếm toàn văn bản.....	29
2.5. Biểu đồ tuần tự.....	30
2.5.1. Khởi tạo ID và Nhận diện thiết bị	30
2.5.2. Xem danh sách video	31
2.5.3. Import video, xử lý transcript và Phân tích CEFR	32
2.5.4. Yêu thích video	33
2.5.5. Bookmark từ vựng và câu	34
2.5.6. Hiển thị transcript và Bộ điều khiển LiveSubs	35
2.5.7. Dịch câu (Azure Translator).....	36
2.5.8. Xem phát âm và định nghĩa từ vựng (FreeDictionary API).....	36
2.5.9. Tìm kiếm toàn văn bản (Title và Description)	37
2.5.10.Ghi âm user và Chấm điểm đọc (Whisper)	38

2.6. Sơ đồ lớp	39
CHƯƠNG 3. PHÁT TRIỂN ỨNG DỤNG.....	41
3.1. Kiến trúc tổng thể hệ thống.....	41
3.1.1. Client (Ứng dụng di động)	41
3.1.2. Server chính (Backend API)	41
3.1.3. Dịch vụ xử lý ngôn ngữ (AI Microservices)	41
3.1.4. Tích hợp bên ngoài (Third-party APIs).....	42
3.1.5. Hàng đợi xử lý và tác vụ nền.....	42
3.1.6. Lưu trữ dữ liệu và tệp âm thanh	42
3.1.7. Riêng tư và bảo mật.....	42
3.2. Môi trường phát triển	42
3.3. Quá trình phát triển hệ thống	43
3.4. Phát triển Ứng dụng di động (React Native và Expo)	44
3.4.1. Cài đặt môi trường.....	44
3.4.2. Tạo project mới	44
3.4.3. Cấu trúc thư mục cơ bản	45
3.4.4. Cấu hình iOS	45
3.4.5. Giao diện Ứng dụng di động	46
3.5. Phát triển Web Interface	65
3.5.1. Khởi tạo dự án nextjs + shadcn	65
3.5.2. Cấu trúc dự án	66
3.5.3. Màn hình giao diện đóng góp video.....	66
KẾT QUẢ VÀ KIẾN NGHỊ	70
1. Kết quả đạt được	70
2. Hạn chế hiện tại.....	70
3. Kiến nghị phát triển.....	71
TÀI LIỆU THAM KHẢO	72

DANH MỤC CHỮ VIẾT TẮT

STT	Từ viết tắt	Thuật ngữ	Ý nghĩa
1	ASR	Automatic Speech Recognition	Nhận dạng giọng nói tự động
2	BERT	Bidirectional Encoder Representations from Transformers	Biểu diễn mã hóa hai chiều từ Transformer
3	CEFR	Common European Framework of Reference for Languages	Khung tham chiếu trình độ ngôn ngữ châu Âu
4	NLP	Natural Language Processing	Xử lý ngôn ngữ tự nhiên
5	API	Application Programming Interface	Giao diện lập trình ứng dụng
6	gRPC	gRPC Remote Procedure Call	Gọi thủ tục từ xa gRPC
7	ORM	Object-Relational Mapping	Ánh xạ quan hệ đối tượng
8	REST	Representational State Transfer	Truyền trạng thái biểu diễn
9	UI	User Interface	Giao diện người dùng
10	UX	User Experience	Trải nghiệm người dùng
11	SQL	Structured Query Language	Ngôn ngữ truy vấn có cấu trúc
12	DOM	Document Object Model	Mô hình đối tượng tài liệu
13	SSR	Server-Side Rendering	Kết xuất phía máy chủ
14	SSG	Static Site Generation	Tạo trang tĩnh
15	SPA	Single-Page Application	Ứng dụng trang đơn
16	CLI	Command Line Interface	Giao diện dòng lệnh
17	JWT	JSON Web Token	Mã thông báo web JSON
18	CORS	Cross-Origin Resource Sharing	Chia sẻ tài nguyên đa nguồn
19	IPA	International Phonetic Alphabet	Bảng ký hiệu phiên âm quốc tế
20	WER	Word Error Rate	Tỷ lệ lỗi từ
21	MAE	Mean Absolute Error	Sai số tuyệt đối trung bình
22	MSE	Mean Squared Error	Sai số bình phương trung bình
23	UUID	Universally Unique Identifier	Định danh duy nhất toàn cầu
24	ASGI	Asynchronous Server Gateway Interface	Giao diện cổng máy chủ bất đồng bộ
25	NER	Named Entity Recognition	Nhận diện thực thể có tên
26	RLHF	Reinforcement Learning from Human Feedback	Học tăng cường từ phản hồi con người

DANH MỤC HÌNH ẢNH VÀ BẢNG BIỂU

Bảng 2. 1. Bảng phân tích thiết kế hệ thống	17
Hình 2. 1. Sơ đồ phân cấp chức năng	20
Hình 2. 2. Sơ đồ ERD	21
Hình 2. 3. Sơ đồ Usecase tổng quát.....	22
Hình 2. 4. Sơ đồ Usecase Khởi tạo ID và Nhận diện thiết bị	23
Hình 2. 5. Sơ đồ Usecase Xem danh sách video	24
Hình 2. 6. Sơ đồ Usecase Import video, xử lý transcript và Phân tích CEFR	25
Hình 2. 7. Sơ đồ Usecase Yêu thích video.....	26
Hình 2. 8. Sơ đồ Usecase bookmark.....	26
Hình 2. 9. Sơ đồ Usecase Hiển thị transcript và Bộ điều khiển LiveSubs	27
Hình 2. 10. Sơ đồ Usecase xem translation	28
Hình 2. 11. Sơ đồ Usecase Xem phát âm và định nghĩa từ vựng	28
Hình 2. 12. Sơ đồ. Usecase Tìm kiếm toàn văn bản	29
Hình 2. 13. Biểu đồ tuần tự Khởi tạo ID và Nhận diện thiết bị.....	30
Hình 2. 14. Biểu đồ tuần tự Xem danh sách video	31
Hình 2. 15. Biểu đồ tuần tự Import video, xử lý transcript và Phân tích CEFR.....	32
Hình 2. 16. Biểu đồ tuần tự Yêu thích video	33
Hình 2. 17. Biểu đồ tuần tự Bookmark từ vựng và câu	34
Hình 2. 18. Biểu đồ tuần tự Hiển thị transcript và Bộ điều khiển LiveSubs	35
Hình 2. 19. Biểu đồ tuần tự Dịch câu (Azure Translator).....	36
Hình 2. 20. Biểu đồ tuần tự Xem phát âm và định nghĩa từ vựng.....	36
Hình 2. 21. Biểu đồ tuần tự Tìm kiếm toàn văn (Title và Description).....	37
Hình 2. 22. Biểu đồ tuần tự Ghi âm user và Chấm điểm đọc (Whisper).....	38
Hình 2. 23. Sơ đồ lớp.....	39

Hình 3. 1. Cấu trúc thư mục Ứng dụng di động	45
Hình 3. 2. Màn hình Splash	46
Hình 3. 3. Màn hình tìm kiếm.....	48
Hình 3. 4. Màn hình chính	50
Hình 3. 5. Màn hình video chi tiết	52
Hình 3. 6. Màn hình yêu thích	54
Hình 3. 7. Màn hình saved	55
Hình 3. 8. Màn hình Import video	57
Hình 3. 9. Màn hình luyện tập Shadowing	59
Hình 3. 10. Màn hình So sánh & chấm điểm tại Frontend	61
Hình 3. 11. Màn hình xem từ vựng.....	63
Hình 3. 12. Màn hình dịch câu	65
Hình 3. 13. Cấu trúc thư mục Website.....	66
Hình 3. 14. Màn hình danh sách videos.....	67
Hình 3. 15. Màn hình thêm mới video.....	68
Hình 3. 16. Màn hình video chi tiết	68
Hình 3. 17. Màn hình transcript.....	69

TỔNG QUAN

1. Lý do chọn đề tài

Trong xu hướng học tiếng Anh hiện đại, việc ứng dụng công nghệ vào luyện kỹ năng nghe – nói ngày càng phổ biến. Đề tài xây dựng ứng dụng luyện nói tiếng Anh theo phương pháp shadowing, kết hợp công nghệ ASR (Whisper) và mô hình BERT để đánh giá phát âm và trình độ CEFR. Người dùng chỉ cần nhập link YouTube, hệ thống sẽ tự động lấy transcript, tách câu, phân tích độ khó và hiển thị phụ đề kèm bản dịch. Ứng dụng hỗ trợ luyện nói từng câu: nghe – lặp lại – ghi âm – chấm điểm – góp ý lỗi. Giao diện đơn giản, dễ dùng, phù hợp với người học tự do trên thiết bị di động. Tập trung vào luyện phát âm câu đơn lẻ, chưa hỗ trợ hội thoại dài hay chức năng đăng nhập phức tạp. Hệ thống mang đến trải nghiệm học tự nhiên, linh hoạt và sát với ngữ cảnh thực tế.

2. Mục đích nghiên cứu

Lý do em chọn làm đề tài này là vì em muốn vừa học tiếng Anh, vừa làm một sản phẩm có ích, kết hợp giữa sở thích học ngôn ngữ và lập trình. Trong quá trình tự học, em thấy phương pháp shadowing rất hiệu quả để luyện phản xạ và phát âm, nhưng em lại chưa tìm được ứng dụng nào thật sự tiện, dễ dùng và phù hợp với nhu cầu của người học như em – đặc biệt là trên điện thoại.

Em nghĩ YouTube là một kho tài nguyên khổng lồ với vô vàn video tiếng Anh thực tế, rất phù hợp để luyện nói mỗi ngày. Nếu có thể tận dụng những video này, kết hợp với các mô hình ngôn ngữ hiện đại như Whisper để chuyển giọng nói thành văn bản, rồi dùng BERT để đánh giá trình độ câu theo CEFR, thì người học như em sẽ có một công cụ rất mạnh để tự học và luyện tập.

Em làm đề tài này vì em muốn tạo ra một ứng dụng đơn giản nhưng hiệu quả, để bất kỳ ai cũng có thể học shadowing mọi lúc mọi nơi, chỉ cần có một video, một chiếc điện thoại và sự kiên trì. Em tin rằng việc tự học sẽ dễ dàng hơn rất nhiều nếu có sự hỗ trợ đúng cách từ công nghệ.

3. Phương pháp nghiên cứu

Trong quá trình thực hiện đề tài, em kết hợp giữa việc nghiên cứu lý thuyết, tìm hiểu công nghệ và xây dựng hệ thống thực tế. Cụ thể, các phương pháp em sử dụng bao gồm:

3.1 Tự học và nghiên cứu tài liệu chuyên ngành

Em dành thời gian tìm hiểu các tài liệu về phương pháp shadowing, khung đánh giá CEFR, mô hình nhận dạng giọng nói (ASR – Whisper), và mô hình xử lý ngôn ngữ tự nhiên (BERT, Sentence-BERT). Việc đọc tài liệu giúp em hiểu rõ nguyên lý hoạt động, ưu điểm – hạn chế của từng công nghệ trước khi áp dụng.

3.2 Khảo sát thực tế và trải nghiệm người học

Em tham khảo nhiều ứng dụng học tiếng Anh hiện có để rút ra ưu – nhược điểm, từ đó xác định những yếu tố người học thực sự cần khi luyện nói theo phương pháp shadowing: đơn giản, phản hồi nhanh, không cần thao tác phức tạp.

3.3 Thiết kế mô hình hệ thống và chia nhỏ chức năng

Em sử dụng các sơ đồ như BFD, Usecase, ERD, Sequence diagram để phân tích và thiết kế hệ thống. Mỗi chức năng được mô tả rõ ràng, giúp việc triển khai dễ dàng và linh hoạt hơn.

3.4. Triển khai mô hình AI và xử lý ngôn ngữ

- Em sử dụng Whisper (ASR) để tạo transcript và nhận dạng giọng nói người học.
- Với mô hình ngôn ngữ, em sử dụng BERT để phân tích câu và dự đoán trình độ CEFR.
- Các mô hình này được triển khai dưới dạng microservices sử dụng Python + FastAPI.

3.5. Phát triển ứng dụng thực tế

- Em sử dụng React Native để xây dựng app di động, nơi người học luyện nói, ghi âm và nhận phản hồi.
- Backend viết bằng Golang xử lý logic và kết nối với các dịch vụ AI.
- Web interface dùng Next.js để upload video và theo dõi nội dung học.

3.6. Kiểm thử, thu thập phản hồi và cải tiến

Sau mỗi giai đoạn, em đều kiểm thử chức năng bằng tay hoặc tự động, ghi nhận lỗi và điều chỉnh. Em cũng mời bạn bè cùng trải nghiệm thử app và đóng góp ý kiến để cải tiến giao diện và hiệu quả học tập.

4. Phạm vi nghiên cứu

Trong đề tài này, em tập trung xây dựng một hệ thống hỗ trợ luyện nói tiếng Anh theo phương pháp shadowing, kết hợp với các công nghệ hiện đại như nhận dạng giọng nói và mô hình ngôn ngữ tự nhiên. Phạm vi nghiên cứu cụ thể bao gồm:

- Phát triển ứng dụng di động (React Native): hỗ trợ người dùng nhập link video YouTube, xem phụ đề tương tác, luyện nói từng câu, ghi âm giọng đọc và nhận phản hồi ngay lập tức về mức độ chính xác.
- Xây dựng phiên bản web đơn giản: hỗ trợ đóng góp video YouTube, xử lý transcript và dự đoán cấp độ CEFR cho từng câu bằng mô hình ngôn ngữ BERT.
- Nghiên cứu và ứng dụng công nghệ ASR (Automatic Speech Recognition): sử dụng mô hình Whisper của OpenAI để:
 - Tự động tạo transcript từ audio của video.
 - Chuyển giọng nói người dùng thành văn bản để so sánh và chấm điểm phát âm.
- Dự đoán độ khó của câu theo CEFR: áp dụng mô hình BERT hoặc Sentence-BERT để phân tích nội dung và gán cấp độ từ A1 đến C2.
- Chỉ hỗ trợ video tiếng Anh – chưa mở rộng sang các ngôn ngữ khác hay nội dung hội thoại dài.
- Tập trung vào luyện nói theo từng câu ngắn, đơn lẻ – chưa xử lý các đoạn hội thoại phức tạp hay giao tiếp hai chiều.
- Không yêu cầu tài khoản đăng nhập – sử dụng mã định danh thiết bị để nhận diện người dùng một cách đơn giản và bảo mật.
- Tích hợp tìm kiếm toàn văn (Full-Text Search) – giúp người dùng dễ dàng tra cứu nội dung theo từ khóa trong toàn bộ hệ thống.
- Chưa triển khai hệ thống quản trị chuyên sâu – hiện tại chưa có dashboard báo cáo hay công cụ giám sát nâng cao.

Phạm vi nghiên cứu này cho phép em tập trung triển khai một hệ thống vừa học vừa làm thực tế, tận dụng tốt các công nghệ ASR và NLP hiện đại để giúp người học tiếng Anh luyện nói hiệu quả và chủ động hơn

5. Kết quả dự kiến đề tài

Mục tiêu chính của đề tài là xây dựng một hệ thống ứng dụng trên nền tảng di động nhằm hỗ trợ người học tiếng Anh nâng cao kỹ năng nói, đặc biệt là phát âm và phản xạ ngôn ngữ, thông qua phương pháp shadowing kết hợp với các công nghệ trí tuệ nhân tạo hiện đại như ASR (Whisper) và mô hình ngôn ngữ BERT. Ứng dụng sẽ khai thác nguồn tài nguyên video tiếng Anh thực tế từ YouTube, từ đó xử lý nội dung và cung cấp môi trường luyện tập hiệu quả, tiện lợi và mang tính cá nhân hóa.

Cụ thể, đề tài hướng đến việc hiện thực hóa các mục tiêu sau:

- Phát triển một ứng dụng mobile đơn giản, trực quan, cho phép người học luyện nói tiếng Anh theo phương pháp shadowing, với khả năng sử dụng mọi lúc, mọi nơi mà không yêu cầu đăng nhập phức tạp.
- Tích hợp công nghệ nhận dạng giọng nói tự động (ASR) – sử dụng mô hình Whisper của OpenAI để trích xuất nội dung lời thoại từ video tiếng Anh trên YouTube một cách chính xác, kể cả với tiếng nói tự nhiên.
- Áp dụng mô hình ngôn ngữ BERT hoặc Sentence-BERT để chia nhỏ các câu từ transcript và đánh giá độ khó của từng câu theo khung trình độ ngôn ngữ CEFR (A1 – C2), hỗ trợ người học chọn nội dung phù hợp với khả năng hiện tại.
- Hỗ trợ người dùng luyện nói từng câu thông qua quy trình nghe – lặp lại – ghi âm – phản hồi, từ đó nâng cao khả năng phát âm, phát hiện lỗi sai và cải thiện phát âm dựa trên so sánh với bản gốc.
- Cung cấp các công cụ hiển thị phụ đề song ngữ, phiên âm và gợi ý cách phát âm từng từ, giúp người học nắm rõ ngữ nghĩa, ngữ âm và cách dùng từ trong ngữ cảnh thực tế.
- Bảo vệ dữ liệu người dùng và đảm bảo quyền riêng tư, đặc biệt trong quá trình ghi âm và xử lý giọng nói.

- Khuyến khích người học chủ động tiếp cận tiếng Anh thông qua nội dung thực tế từ YouTube, từ đó tăng cường sự hứng thú và khả năng sử dụng tiếng Anh trong giao tiếp hàng ngày.

Thông qua hệ thống này, đề tài kỳ vọng sẽ mang đến một công cụ học tập hiệu quả, dễ tiếp cận và phù hợp với xu hướng tự học, từ đó góp phần nâng cao chất lượng học tiếng Anh trong cộng đồng người học tại Việt Nam.

Mục tiêu nghiên cứu của đề tài là tìm hiểu, phân tích và ứng dụng các công nghệ trí tuệ nhân tạo hiện đại trong việc xây dựng hệ thống hỗ trợ luyện nói tiếng Anh theo phương pháp shadowing. Cụ thể, đề tài tập trung vào việc nghiên cứu và áp dụng mô hình nhận dạng giọng nói (ASR) và mô hình ngôn ngữ (NLP) để xử lý dữ liệu giọng nói và đánh giá ngôn ngữ một cách tự động, chính xác, hiệu quả.

Các mục tiêu nghiên cứu chính bao gồm:

- Nghiên cứu phương pháp shadowing trong dạy và học tiếng Anh, phân tích hiệu quả và cách tích hợp phương pháp này vào mô hình ứng dụng công nghệ.
- Tìm hiểu và đánh giá công nghệ nhận dạng giọng nói Whisper của OpenAI, đặc biệt là khả năng trích xuất transcript chính xác từ các video YouTube có giọng nói tự nhiên, đa dạng về tốc độ và giọng điệu.
- Phân tích và ứng dụng mô hình BERT/Sentence-BERT trong đánh giá ngữ nghĩa câu tiếng Anh, từ đó xây dựng tiêu chí phân loại độ khó theo khung tham chiếu trình độ ngôn ngữ châu Âu (CEFR).
- Nghiên cứu kỹ thuật so sánh giọng nói giữa bản ghi âm của người học và mẫu gốc, qua đó đề xuất thuật toán hoặc tiêu chí đánh giá phát âm, phát hiện lỗi và cung cấp phản hồi.
- Khảo sát trải nghiệm người dùng trong môi trường học tập không chính thức trên thiết bị di động, từ đó đưa ra định hướng thiết kế giao diện đơn giản, hiệu quả và phù hợp với nhu cầu thực tế.
- Đánh giá khả năng ứng dụng mô hình ngôn ngữ và công nghệ nhận dạng giọng nói trong việc tạo ra hệ thống phản hồi tự động và đánh giá kỹ năng nói tiếng Anh một cách khách quan.

Thông qua các mục tiêu nghiên cứu trên, đề tài không chỉ tạo ra một sản phẩm ứng dụng, mà còn đóng góp giá trị nghiên cứu trong việc kết hợp giữa công nghệ ngôn ngữ tự nhiên (NLP), ASR, và giáo dục ngôn ngữ nhằm hỗ trợ người học tiếng Anh một cách hiệu quả, thông minh và thực tiễn.

6. Cấu trúc cuốn báo cáo

Báo cáo được chia thành 5 chương, cụ thể như sau:

- **Chương 1 – Mở đầu:** Trình bày tổng quan về đề tài, bao gồm lý do chọn đề tài, mục tiêu và mục đích nghiên cứu, đối tượng và phạm vi nghiên cứu, ý nghĩa thực tiễn, phương pháp nghiên cứu và cấu trúc tổng thể của hệ thống.
- **Chương 2 – Cơ sở lý thuyết và công nghệ sử dụng:** Trình bày các kiến thức nền tảng liên quan đến ngôn ngữ lập trình (Golang, Python, TypeScript...), các framework và công cụ được sử dụng (React Native, FastAPI, GORM, yt-dlp...), cùng các mô hình trí tuệ nhân tạo như Whisper (ASR), BERT, và các API bên thứ ba (Azure Translator, Free Dictionary API).
- **Chương 3 – Phân tích và thiết kế hệ thống:** Phân tích yêu cầu hệ thống, xác định các chức năng chính, xây dựng sơ đồ phân cấp chức năng (BFD), sơ đồ usecase, ERD, sơ đồ tuần tự và sơ đồ lớp nhằm mô hình hóa toàn bộ quy trình xử lý và kiến trúc hệ thống luyện nói theo phương pháp shadowing.
- **Chương 4 – Phát triển ứng dụng:** Trình bày chi tiết quá trình triển khai hệ thống bao gồm thiết kế kiến trúc backend – frontend, phát triển ứng dụng di động và phiên bản web, xây dựng các microservices AI, xử lý audio, đánh giá CEFR, luyện phát âm, cùng giao diện người dùng và chức năng tương tác hx`oc tập.
- **Chương 5 – Kết quả và kiến nghị:** Tổng kết kết quả đã đạt được, đánh giá hiệu năng hệ thống và mô hình AI, phân tích các hạn chế còn tồn tại, và đề xuất hướng phát triển trong tương lai như tối ưu mô hình CEFR, mở rộng tính năng học tập, cải tiến giao diện và mở rộng sang các ngôn ngữ khác.

CHƯƠNG 1. CƠ SỞ LÝ THUYẾT

Hệ thống luyện nói tiếng Anh theo phương pháp shadowing được xây dựng bằng sự kết hợp của nhiều ngôn ngữ lập trình, công nghệ web, framework hiện đại, cùng các mô hình trí tuệ nhân tạo mạnh mẽ phục vụ xử lý tiếng nói và ngôn ngữ. Dưới đây là phần trình bày chi tiết các công nghệ được ứng dụng trong đề tài.

1.1. Golang (Go)

1.1.1. Tổng quan về ngôn ngữ lập trình Golang

Golang, hay gọi tắt là Go, là một ngôn ngữ lập trình mã nguồn mở do Google phát triển, chính thức công bố vào năm 2009. Go được thiết kế với mục tiêu đơn giản hóa quá trình phát triển phần mềm, đồng thời vẫn giữ được hiệu năng cao tương đương với các ngôn ngữ biên dịch như C/C++. Với cú pháp gọn gàng, khả năng biên dịch nhanh, và hỗ trợ concurrency mạnh mẽ, Go trở thành lựa chọn hàng đầu trong các lĩnh vực như hệ thống backend, dịch vụ web, và microservices.

Một số đặc điểm nổi bật của Go:

- Hiệu năng cao: Go được biên dịch trực tiếp xuống mã máy, mang lại tốc độ thực thi rất nhanh.
- Concurrency nhẹ: Mô hình xử lý song song với goroutine và channel giúp lập trình song song dễ dàng và tối ưu hơn nhiều so với luồng (thread) truyền thống.
- Quản lý bộ nhớ tự động: Hệ thống garbage collector tích hợp giúp giảm thiểu lỗi rò rỉ bộ nhớ mà không làm chậm chương trình đáng kể.
- Đơn giản và dễ học: Go có cú pháp rõ ràng, thư viện chuẩn phong phú và triết lý tối giản, giúp tăng năng suất phát triển và dễ bảo trì.

Ứng dụng trong hệ thống:

- Xây dựng backend và các API xử lý logic hệ thống.
- Giao tiếp với các service AI thông qua gRPC.
- Thực hiện các thao tác dữ liệu với PostgreSQL thông qua GORM.

1.1.2. Framework Echo

Echo là một framework web được viết bằng Go, nổi bật với hiệu suất cao, nhẹ, và được tối ưu cho việc xây dựng các API RESTful. Framework này cung cấp khả năng định tuyến nhanh, hỗ trợ middleware, xử lý lỗi hiệu quả, và tích hợp tốt với các công nghệ hiện đại như JWT, CORS, gRPC...

Ưu điểm của Echo:

- Tốc độ phản hồi nhanh: Khả năng xử lý yêu cầu rất tối ưu, phù hợp với ứng dụng real-time.
- Hỗ trợ middleware: Dễ dàng tích hợp xác thực, logging, xử lý lỗi tập trung.
- Quản lý route linh hoạt: Hỗ trợ nhóm route theo module, giúp tổ chức code rõ ràng.
- Tương thích cao: Dễ kết nối với hệ sinh thái Go như gRPC, GORM, PostgreSQL...

Ứng dụng trong hệ thống:

- Tạo ra các API phục vụ frontend tương tác.
- Là điểm trung chuyển kết nối frontend với các service AI nội bộ.

1.1.3. Thư viện GORM

GORM (Go Object Relational Mapper) là thư viện ORM phổ biến nhất trong cộng đồng Golang, cho phép ánh xạ giữa struct trong Go với bảng dữ liệu trong hệ quản trị quan hệ như PostgreSQL.

Ưu điểm:

- Cú pháp rõ ràng, dễ hiểu: Giảm thiểu việc viết câu lệnh SQL thủ công.
- Tự động ánh xạ: Các struct có thể tự ánh xạ qua tag gorm, hỗ trợ cả quan hệ 1-n, n-n...
- Hỗ trợ migration: Tạo bảng, cột, hoặc cập nhật schema trực tiếp từ code.
- An toàn: Giảm thiểu lỗi SQL injection nhờ cơ chế binding dữ liệu.

Ứng dụng trong hệ thống:

- GORM hỗ trợ ánh xạ struct Go với bảng dữ liệu PostgreSQL.

- Giúp đơn giản hóa các truy vấn, tăng hiệu quả lập trình và giảm lỗi SQL.

1.2. Python

1.2.1. Tổng quan về ngôn ngữ lập trình Python

Python là một ngôn ngữ lập trình cấp cao, thông dịch, có cú pháp đơn giản, dễ học và phổ biến rộng rãi trong các lĩnh vực như web, automation, data science và đặc biệt là AI/ML. Nhờ hệ sinh thái thư viện cực kỳ phong phú và cộng đồng lớn mạnh, Python là lựa chọn lý tưởng để phát triển các mô hình học máy và xử lý ngôn ngữ tự nhiên (NLP).

Các thư viện nổi bật: PyTorch, TensorFlow, HuggingFace Transformers, spaCy, SpeechRecognition...

Ngoài ra, Python còn có:

- **Khả năng tương tác tốt với các hệ thống khác:** Nhờ thư viện như requests, aiohttp, socket,...
- **Đa dạng môi trường phát triển:** Có thể chạy trên Jupyter, Colab, môi trường server, CLI hoặc tích hợp trong API.
- **Hỗ trợ lập trình hướng đối tượng, hàm, và mô hình script:** Phù hợp với nhiều phong cách lập trình.

Ứng dụng trong hệ thống:

- Triển khai mô hình Whisper cho nhận dạng giọng nói.
- Phân tích nội dung và đánh giá độ khó câu theo chuẩn CEFR.
- Xây dựng các microservice phục vụ backend chính thông qua FastAPI.

1.2.2. Framework FastAPI

FastAPI là framework web hiện đại, nhanh, và nhẹ, chuyên dùng để xây dựng các REST API bằng Python. Nó hỗ trợ tính năng tự động tạo tài liệu OpenAPI và có performance cao nhờ chạy trên nền tảng ASGI (Asynchronous Server Gateway Interface).

Ngoài ra:

- **Hỗ trợ mô hình bất đồng bộ (async/await):** Phù hợp với xử lý I/O cường độ cao như AI service, NLP, ASR.

- Tích hợp tốt với Pydantic: Giúp xác thực và chuyển đổi dữ liệu mạnh mẽ nhờ typing.
- Tài liệu tự sinh mạnh mẽ: Có thể kiểm thử API trực tiếp từ Swagger UI và ReDoc.

Ưu điểm:

- Tốc độ xử lý nhanh: Sử dụng asyncio nên rất phù hợp với AI service cần phản hồi nhanh.
- Tự động tạo Swagger UI và docs.
- Dễ triển khai, maintain: Cú pháp gần gũi với Python hiện đại (type hint, pydantic...).

Ứng dụng trong hệ thống:

- Là nền tảng triển khai các service Whisper, BERT, CEFR NLP.
- Tự động sinh tài liệu API, hỗ trợ kiểm thử nhanh.

1.3. Tổng quan về ngôn ngữ TypeScript

TypeScript là một siêu ngôn ngữ của JavaScript, bổ sung hệ thống kiểu tĩnh. Nhờ vào khả năng kiểm tra lỗi sớm, hỗ trợ mạnh từ IDE, và khả năng mở rộng tốt, TypeScript ngày càng trở thành tiêu chuẩn trong phát triển frontend hiện đại, đặc biệt là với React hoặc Next.js.

Ưu điểm:

- Tăng khả năng tái sử dụng và bảo trì code: Nhờ typing, các component và hàm trở nên dễ đọc và ít lỗi.
- Thích hợp cho dự án lớn: Giúp giảm bug, tăng khả năng refactor an toàn.

Ứng dụng trong hệ thống:

- Phát triển frontend web với React hoặc Next.js.
- Quản lý trạng thái và logic tương tác người dùng.

1.4. JavaScript

1.4.1. Tổng quan về ngôn ngữ JavaScript

JavaScript là ngôn ngữ lập trình phổ biến nhất dùng để xây dựng các ứng dụng web động, chạy trực tiếp trong trình duyệt. JS cho phép tạo giao diện người dùng tương tác,

cập nhật nội dung theo thời gian thực, và giao tiếp với các API backend một cách linh hoạt.

Đặc điểm nổi bật:

- Lập trình hướng sự kiện (event-driven): JS cho phép xử lý sự kiện người dùng như nhấp chuột, nhập dữ liệu, v.v.
- Khả năng thao tác DOM mạnh mẽ: JS có thể thay đổi nội dung HTML và CSS trên trình duyệt mà không cần tải lại trang.
- Asynchronous programming: Hỗ trợ cơ chế bất đồng bộ thông qua callback, Promise, async/await, giúp cải thiện hiệu năng và trải nghiệm người dùng.
- Khả năng mở rộng cao: Nhờ các framework và thư viện như React, Vue, Angular, JS có thể được dùng cho cả frontend và backend (Node.js).

Ứng dụng trong hệ thống:

- Tạo giao diện người dùng thân thiện.
- Xử lý tương tác thời gian thực.
- Kết nối với các API backend để hiển thị nội dung học tập và kết quả chấm điểm.

1.4.2. Next.js

Next.js là một framework mở rộng của React hỗ trợ nhiều tính năng cao cấp như server-side rendering (SSR), static site generation (SSG), và routing tự động.

Ưu điểm:

- Tối ưu hiệu năng tải trang và SEO.
- Tích hợp frontend + backend nhẹ trong cùng một project.
- Dễ mở rộng và linh hoạt.

Ứng dụng trong hệ thống:

- Dùng để xây dựng giao diện web chính.
- Tối ưu tốc độ tải trang và SEO.
- Kết hợp frontend và backend nhẹ cho các tác vụ không cần backend riêng.

1.4.3. ReactJS

ReactJS (thường được gọi tắt là React) là một thư viện JavaScript mã nguồn mở được phát triển bởi Facebook (nay là Meta Platforms). Mục tiêu chính của React là giúp các nhà phát triển xây dựng giao diện người dùng (UI) một cách hiệu quả và linh hoạt, đặc biệt là cho các ứng dụng đơn trang (Single-Page Applications - SPAs).

Ưu điểm:

- Virtual DOM: Giúp cập nhật chỉ những phần thay đổi thay vì toàn bộ trang, nâng cao hiệu suất.
- One-way Data Binding: Dữ liệu chỉ chảy một chiều, giúp kiểm soát luồng dữ liệu rõ ràng hơn.
- Hệ sinh thái mở rộng: Dễ dàng tích hợp với Redux, Context API, hoặc kết hợp với các thư viện như Axios, Formik, React Router...

Ứng dụng trong hệ thống:

- Xây dựng giao diện học tập như hiển thị video, đoạn văn bản, ghi âm, điểm số.
- Tương tác với backend để hiển thị kết quả chấm điểm và lịch sử học tập.

1.4.4. React Native

React Native là một framework mã nguồn mở được phát triển bởi Facebook (nay là Meta Platforms). Mục tiêu chính của nó là cho phép các nhà phát triển xây dựng ứng dụng di động gốc (native apps) cho cả iOS và Android chỉ bằng một cơ sở mã (codebase) duy nhất, sử dụng JavaScript và thư viện React.

Ưu điểm:

- Phát triển đa nền tảng: Viết mã một lần và triển khai trên cả iOS và Android, giúp tiết kiệm thời gian và chi phí phát triển đáng kể.
- Hiệu suất gần như Native: Nhờ việc sử dụng các thành phần UI gốc, ứng dụng React Native mang lại trải nghiệm mượt mà, phản hồi nhanh chóng, gần giống với ứng dụng native hoàn toàn.
- Tốc độ phát triển nhanh: Tính năng Hot Reloading và Fast Refresh cho phép nhà phát triển xem ngay lập tức các thay đổi mã mà không cần biên dịch lại toàn bộ ứng dụng, tăng tốc quá trình phát triển.

- Cộng đồng lớn và hệ sinh thái phong phú: Là một framework phổ biến, React Native có một cộng đồng hỗ trợ lớn và nhiều thư viện, công cụ bên thứ ba, giúp giải quyết các vấn đề phức tạp.
- Dễ học cho người có kinh nghiệm JavaScript/React: Những nhà phát triển đã quen với JavaScript và React có thể nhanh chóng bắt đầu với React Native.

Ứng dụng trong hệ thống:

- Giúp người học luyện nói mọi lúc mọi nơi trên thiết bị di động.
- Kết nối trực tiếp với microphone, bộ nhớ, và API backend.

1.5. Các thành phần trí tuệ nhân tạo và ngôn ngữ

1.5.1. Whisper

Whisper là một mô hình nhận dạng giọng nói tự động (ASR - Automatic Speech Recognition) mạnh mẽ được phát triển bởi OpenAI. Nó được thiết kế để chuyển đổi giọng nói thành văn bản một cách chính xác và hiệu quả.

Đặc điểm kỹ thuật nổi bật:

- Đa ngôn ngữ: Hỗ trợ hơn 90 ngôn ngữ và có khả năng nhận diện ngôn ngữ đầu vào.
- Multitask Model: Không chỉ nhận dạng, Whisper còn có thể làm segmentation, transcription, translation (dịch từ audio), punctuation...
- Độ chính xác cao: Được huấn luyện từ dữ liệu thực tế, kể cả khi có nhiều nền hoặc chất lượng âm thanh kém.
- Hỗ trợ inference offline: Có thể tích hợp vào hệ thống mà không phụ thuộc cloud.

Ứng dụng trong hệ thống:

- Tạo transcript cho video YouTube không có phụ đề.
- Nhận dạng lời người học nói để phục vụ chấm điểm.

1.5.2. NLP (Natural Language Processing)

NLP là lĩnh vực của trí tuệ nhân tạo tập trung vào việc giúp máy tính hiểu và xử lý ngôn ngữ tự nhiên của con người.

Một số tác vụ phổ biến:

- Tokenization: Cắt câu thành các đơn vị nhỏ (từ, cụm từ).
- Part-of-speech tagging: Gắn nhãn từ loại cho từng từ.
- Named Entity Recognition (NER): Nhận diện thực thể như tên người, địa điểm...
- Dependency Parsing: Phân tích quan hệ giữa các từ trong câu.
- Text Similarity & Semantic Analysis: So sánh câu về mặt ngữ nghĩa.

Ứng dụng trong hệ thống:

- Phân tích cấu trúc câu, mức độ ngữ nghĩa, phục vụ đánh giá CEFR.
- So khớp câu nói người học với câu mẫu gốc.

1.5.3. BERT

BERT (Bidirectional Encoder Representations from Transformers) là một trong những mô hình ngôn ngữ sâu nhất và mạnh nhất hiện nay, được Google công bố năm 2018. Khác với các mô hình truyền thống, BERT học ngữ cảnh từ hai chiều của văn bản, giúp hiểu sâu hơn mối quan hệ giữa các từ.

Ưu điểm:

- Hiểu ngữ cảnh sâu sắc: Khả năng học song hướng giúp BERT hiểu được sự mơ hồ và ý nghĩa đa dạng của từ trong các ngữ cảnh khác nhau.
- Tính linh hoạt cao: Có thể được điều chỉnh tinh chỉnh để đạt hiệu suất cao trên nhiều nhiệm vụ NLP khác nhau mà không cần xây dựng mô hình mới từ đầu.
- Giảm nhu cầu dữ liệu gắn nhãn: Nhờ quá trình tiền huấn luyện trên lượng lớn dữ liệu không gắn nhãn, việc fine-tuning chỉ yêu cầu ít dữ liệu gắn nhãn hơn đáng kể so với các phương pháp truyền thống.
- Kết quả vượt trội: BERT đã đạt được các kết quả tiên tiến (state-of-the-art) trên nhiều bộ dữ liệu và bảng xếp hạng NLP quan trọng ngay khi ra mắt

Ứng dụng trong hệ thống:

- Tạo sentence embedding để phân tích CEFR.
- Đánh giá độ tương đồng ngữ nghĩa giữa lời nói và transcript gốc.

1.5.4. gRPC

gRPC (viết tắt của gRPC Remote Procedure Call) là một framework RPC (Remote Procedure Call) mã nguồn mở hiệu suất cao, được phát triển bởi Google. Nó cho phép các ứng dụng giao tiếp với nhau như thể chúng đang gọi một hàm hoặc phương thức cục bộ, nhưng thực tế các hàm này lại được thực thi trên một máy chủ từ xa khác.

Các kiểu giao tiếp chính trong gRPC

gRPC hỗ trợ bốn kiểu giao tiếp giữa client và server:

- Unary RPC (RPC một chiều): Kiểu giao tiếp phổ biến nhất, client gửi một yêu cầu và server trả về một phản hồi duy nhất, giống như một lời gọi hàm thông thường.
- Server-side Streaming RPC (Server gửi luồng): Client gửi một yêu cầu duy nhất, và server phản hồi bằng một luồng các thông điệp. Client đọc các thông điệp này cho đến khi không còn thông điệp nào nữa.
- Client-side Streaming RPC (Client gửi luồng): Client gửi một luồng các thông điệp tới server. Sau khi client gửi xong, server xử lý luồng thông điệp và trả về một phản hồi duy nhất.
- Bidirectional Streaming RPC (Luồng hai chiều): Cả client và server đều gửi một luồng thông điệp cho nhau một cách độc lập. Hai luồng này có thể hoạt động đồng thời và theo bất kỳ thứ tự nào.

Ưu điểm nổi bật:

- Hiệu suất cao: Nhờ Protobuf và HTTP/2, gRPC cực kỳ hiệu quả về băng thông và CPU, lý tưởng cho các dịch vụ microservices.
- Đa ngôn ngữ (Polyglot): gRPC hỗ trợ tạo mã client và server tự động (code generation) cho nhiều ngôn ngữ lập trình khác nhau (Java, Python, C++, Go, Node.js, C#, Ruby, PHP, Dart, v.v.). Điều này giúp các dịch vụ được viết bằng các ngôn ngữ khác nhau có thể dễ dàng giao tiếp.
- Thiết kế API chặt chẽ: Việc sử dụng Protobuf để định nghĩa dịch vụ và thông điệp giúp duy trì các API rõ ràng, mạnh mẽ và nhất quán.

- Tích hợp tốt với các công nghệ Cloud Native: gRPC thường được sử dụng trong kiến trúc microservices và rất phù hợp với các môi trường điện toán đám mây.
- Streaming: Khả năng hỗ trợ streaming hai chiều là một điểm mạnh lớn, hữu ích cho các ứng dụng thời gian thực hoặc cần truyền lượng lớn dữ liệu liên tục.

Ứng dụng trong hệ thống:

- Kết nối backend chính viết bằng Golang với các dịch vụ AI viết bằng Python.
- Đảm bảo tốc độ truyền dữ liệu như audio, transcript và kết quả xử lý.

1.5.5. yt-dlp

yt-dlp là một công cụ dòng lệnh (command-line program) mã nguồn mở cho phép bạn tải xuống video và âm thanh từ YouTube và hàng ngàn trang web video khác. Nó là một bản phân nhánh (fork) của youtube-dl nổi tiếng, được phát triển để khắc phục những vấn đề về bảo trì và bổ sung các tính năng mới, cải tiến liên tục.

Ưu điểm nổi bật:

- Phạm vi hỗ trợ rộng lớn: Tải xuống từ hầu hết mọi trang web video phổ biến.
- Cập nhật thường xuyên: Khắc phục nhanh chóng các vấn đề tương thích do thay đổi trang web, giúp công cụ luôn hoạt động hiệu quả.
- Tùy biến mạnh mẽ: Cung cấp quyền kiểm soát chi tiết về chất lượng, định dạng và quá trình tải xuống.
- Hiệu suất và độ tin cậy cao: Được tối ưu để tải xuống nhanh chóng và ổn định.
- Miễn phí và mã nguồn mở: Hoàn toàn miễn phí để sử dụng và đóng góp.

Ứng dụng trong hệ thống:

- Dùng để trích xuất phần audio từ video người học cung cấp.
- Kết hợp với Whisper để nhận diện nội dung audio và tạo transcript.

1.5.6. Free Dictionary API

Free Dictionary API là dịch vụ miễn phí cung cấp định nghĩa, từ loại, cách phát âm và ví dụ cho từ vựng tiếng Anh.

Ưu điểm:

- Hoàn toàn miễn phí mà không yêu cầu khóa API cho việc sử dụng cơ bản.
- Cung cấp dữ liệu toàn diện cho một từ.
- Dễ dàng tích hợp do cấu trúc REST đơn giản.

Ứng dụng trong hệ thống:

- Hiển thị nghĩa từ khi người học tra cứu trực tiếp trên giao diện.
- Kết hợp với phát âm audio để người học luyện theo.
- Không cần đăng ký, miễn phí và dễ tích hợp frontend.

1.5.7. Azure Translator và Dictionary API

Azure AI Translator là dịch vụ dịch máy thần kinh dựa trên đám mây của Microsoft. Dịch vụ này cho phép bạn dịch văn bản và tài liệu giữa hàng trăm ngôn ngữ theo thời gian thực

Ứng dụng trong hệ thống:

- Dịch câu transcript: Giúp người học hiểu nội dung trước khi luyện nói hoặc khi muốn xem bản dịch sang tiếng Việt.
- Tra nghĩa và phát âm từ vựng: Cung cấp định nghĩa, từ loại, cách phát âm (IPA và audio), ví dụ sử dụng.
- Gợi ý ngữ cảnh: Hiển thị các cụm từ tương đương, cách sử dụng trong câu.
- Cá nhân hóa nội dung: Dựa trên các từ/cụm từ người học hay tra, hệ thống gợi ý các video chứa từ tương tự.

1.6. Tổng kết

Việc lựa chọn và kết hợp các công nghệ trên giúp hệ thống vừa có hiệu năng cao, trải nghiệm người dùng mượt mà, vừa có khả năng phân tích sâu ngữ nghĩa và phát âm để hỗ trợ người học luyện nói một cách chính xác, chủ động và hiệu quả.

CHƯƠNG 2. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

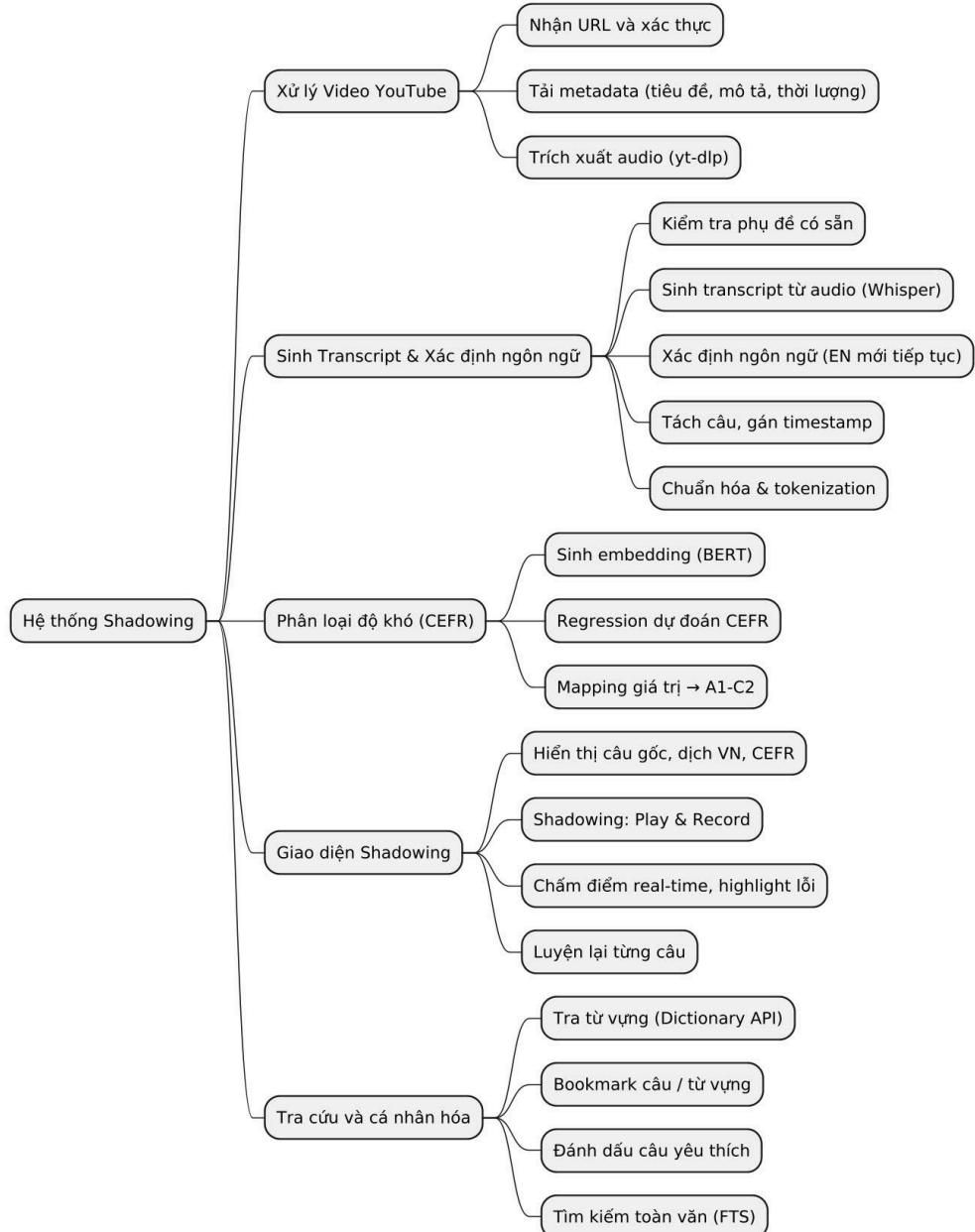
2.1. Phân tích hệ thống

Bảng 2. 1. Bảng phân tích thiết kế hệ thống

STT	Tên chức năng	Mô tả chi tiết chức năng
1	Khởi tạo ID và Nhận diện thiết bị	Khi ứng dụng chạy lần đầu, sinh một UUID/device ID và lưu cục bộ. Mỗi lần mở app, nếu chưa đăng nhập, dùng luôn ID này làm định danh user. ID được gửi kèm mọi request API để phân biệt người dùng.
2	Xem danh sách video	Gọi API lấy metadata video đã import (tiêu đề, thumbnail, độ dài, trạng thái xử lý), hiển thị dạng lưới/danh sách. Hỗ trợ phân trang hoặc lazy-load.
3	Import video, xử lý transcript và Phân tích CEFR	Người dùng dán URL YouTube hoặc chọn file để import. Hệ thống gọi yt-dlp crawl data, tải audio, kiểm tra ngôn ngữ (chỉ tiếng Anh), sau đó gọi Whisper tạo transcript kèm timestamp. Tiếp đó module phân tích CEFR sẽ đánh giá mức (A1→C2) cho từng đoạn. Lưu text, timestamp, CEFR-level và metadata import. Hiển thị tiến trình xử lý.
4	Yêu thích video	Trên mỗi item video có icon “♥”. Khi bấm, gửi API đánh dấu video là favorite. Lần sau tải danh sách sẽ ưu tiên hiển thị video đã yêu thích
5	Bookmark từ vựng và câu	Trong transcript, user chạm giữ từ hoặc câu để hiện menu “Bookmark”. Khi chọn, lưu vào bảng UserBookmarks (video_id, segment_id, type). Có màn “Bookmarks” riêng để xem lại theo bộ lọc “Từ vựng” hoặc “Câu”.
6	Hiển thị transcript và Bộ điều khiển LiveSubs	Giao diện chia hai: video player và transcript. Transcript cuộn tự động (live subtitles) theo timestamp, highlight câu đang phát. Bộ điều khiển gồm play/pause, tua nhanh, chỉnh tốc độ (0.5×–2×). Click vào đoạn text để nhảy đến timestamp tương ứng.

F7	Dịch câu (Azure Translator)	Bên cạnh mỗi đoạn transcript có nút “Dịch”. Khi bấm, gọi API Azure Translator để dịch sang tiếng Việt, hiển thị ngay bên dưới. Kết quả được cache để tái sử dụng, giảm phí dịch.
8	Xem phát âm và định nghĩa từ vựng (FreeDictionary API)	Trong transcript, khi bấm vào một từ, gọi FreeDictionary API lấy phiên âm IPA, audio play và định nghĩa (nhiều nghĩa, ví dụ minh họa). Hiển thị popup/sidebar với thông tin chi tiết. Cho phép lưu từ vào “Flashcards” để ôn tập sau.
9	Tìm kiếm toàn văn bản	Cung cấp ô tìm kiếm toàn cục: nhập từ khoá, hệ thống full-text search trên transcript của tất cả video, trả về danh sách kết quả kèm snippet có highlight. Kết quả hiển thị video, đoạn và timecode; click vào sẽ mở player tại thời điểm đó.
10	Ghi âm user và Chấm điểm đọc (Whisper)	Khi user bấm “Ghi âm” cho một đoạn, app thu âm qua mic và gửi file audio lên server. Server chạy Whisper chuyển audio thành text, trả về transcript của user. Trên client, so sánh text user với transcript gốc, highlight từ/câu sai. Cho phép user xem danh sách từ lỗi, từ thiếu, từ thừa và bấm vào từng từ để nghe cách phát âm chuẩn (từ phiên âm IPA + audio từ FreeDictionary).

2.2. Sơ đồ phân cấp chức năng (BFD) của hệ thống Shadowing

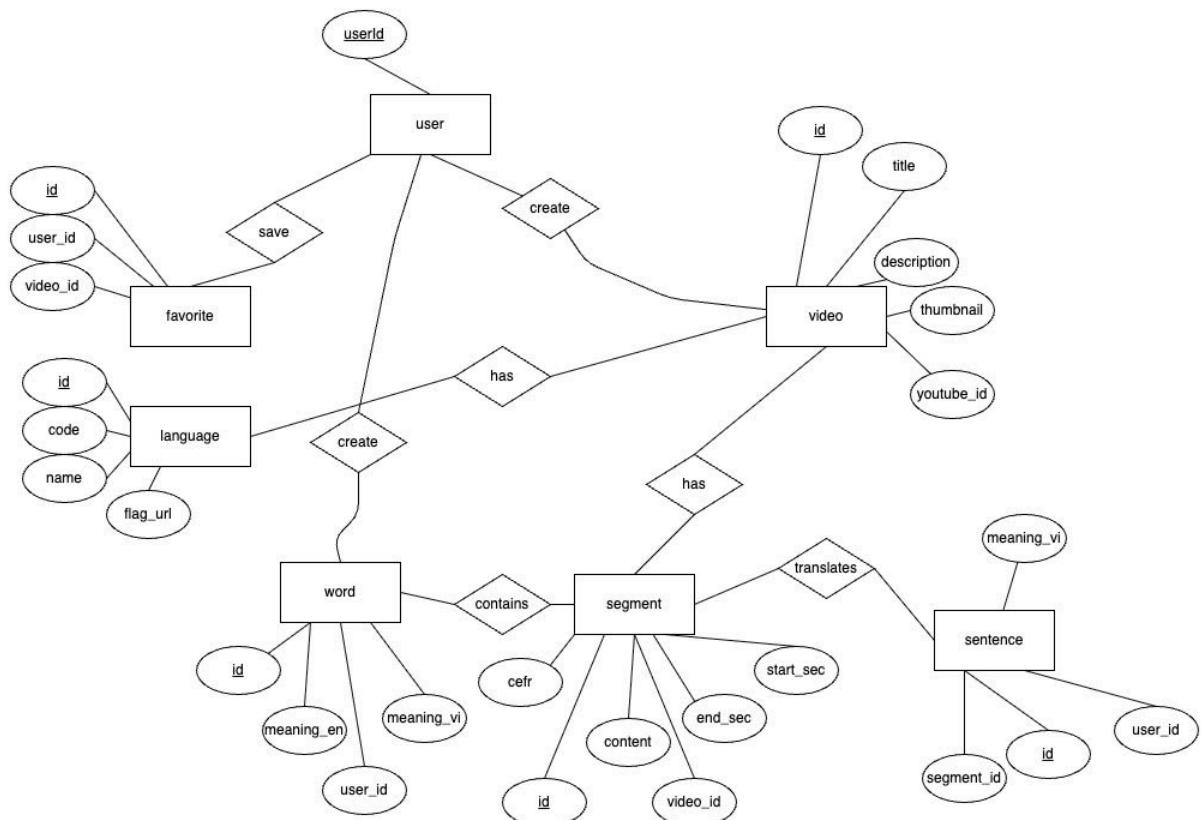


Hình 2. 1. Sơ đồ phân cấp chức năng

Sơ đồ trong ảnh mô tả cấu trúc chức năng của hệ thống Shadowing, bao gồm các nhóm chính. Đầu tiên, hệ thống thực hiện xử lý video YouTube, với các bước nhận URL và xác thực, tải metadata như tiêu đề, mô tả, thời lượng, và trích xuất audio từ video bằng công cụ yt-dlp. Tiếp theo, trong nhóm sinh transcript và xác định ngôn ngữ, hệ thống kiểm tra phụ đề có sẵn, sinh transcript từ audio bằng Whisper, xác định ngôn ngữ (hỗ trợ tiếng Anh), tách câu và gắn timestamp, chuẩn hóa và thực hiện tokenization. Sau đó, hệ thống tiến hành phân loại độ khó (CEFR) bằng cách sử dụng embedding (BERT) để dự đoán cấp độ CEFR thông qua regression, rồi mapping giá trị sang các cấp độ từ

A1 đến C2. Trong nhóm giao diện Shadowing, hệ thống hiển thị câu gốc, bản dịch tiếng Việt, và cấp độ CEFR, đồng thời hỗ trợ các chức năng như Shadowing (Play & Record), chấm điểm real-time, highlight lỗi, và luyện lại từng câu. Cuối cùng, nhóm tra cứu và cá nhân hóa cung cấp các tính năng như tra từ vựng qua Dictionary API, bookmark câu hoặc từ vựng, đánh dấu câu yêu thích, và tìm kiếm toàn văn (Full-Text Search - FTS). Sơ đồ này thể hiện rõ quy trình xử lý và các chức năng của hệ thống, từ việc phân tích video đầu vào đến hỗ trợ học tập và cá nhân hóa trải nghiệm người dùng.

2.3. Mô hình ERD



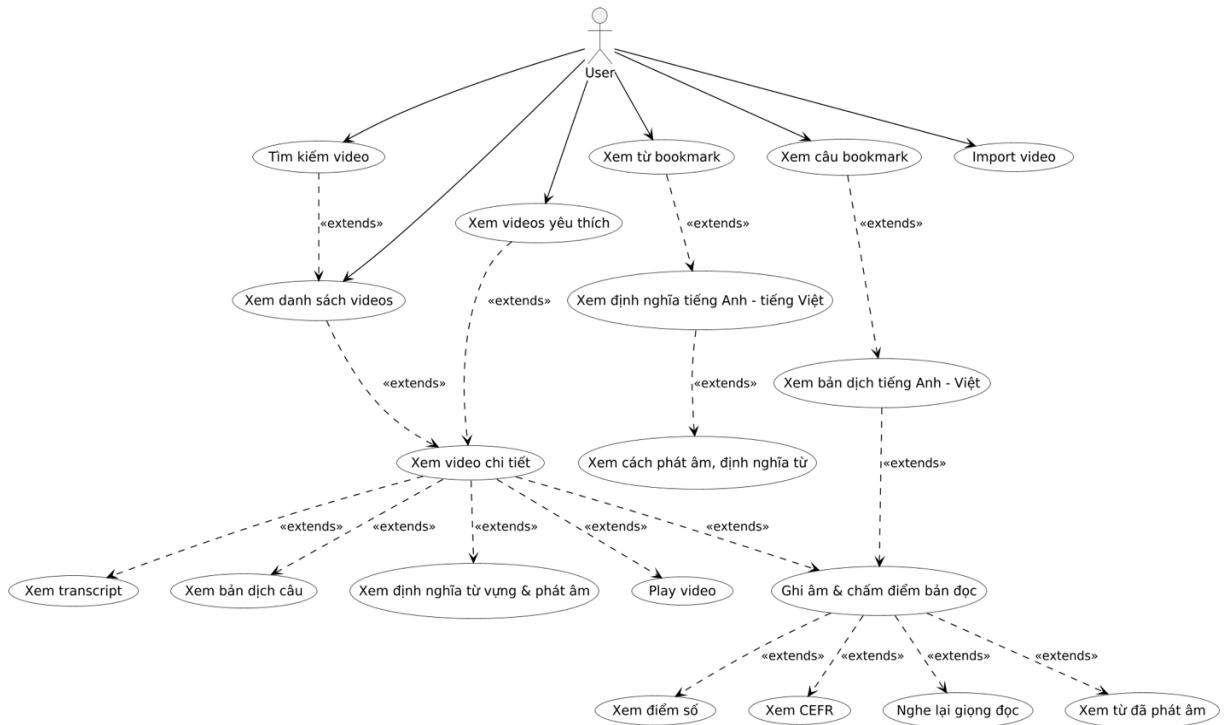
Hình 2. 2. Sơ đồ ERD

Mô hình ERD trong ảnh mô tả các thực thể và mối quan hệ trong hệ thống Shadowing. Thực thể User đại diện cho người dùng, với khả năng tạo video, lưu video yêu thích (thực thể Favorite), và tạo từ vựng (thực thể Word). Thực thể Video chứa các thông tin như tiêu đề, mô tả, thumbnail, và ID YouTube, đồng thời liên kết với ngôn ngữ (thực thể Language) và chứa các đoạn (thực thể Segment). Thực thể Segment đại diện cho các đoạn trong video, với thông tin về nội dung, thời gian bắt đầu và kết thúc, cấp độ CEFR, và liên kết với video. Mỗi đoạn có thể chứa từ vựng (thực thể Word) và dịch ra câu (thực thể Sentence). Thực thể Word lưu trữ thông tin từ vựng, bao gồm nghĩa tiếng Anh, tiếng

Việt, và liên kết với người dùng. Thực thể Sentence đại diện cho câu, với thông tin về nghĩa tiếng Việt, liên kết với người dùng và đoạn. Mô hình này thể hiện rõ các mối quan hệ giữa người dùng, video, ngôn ngữ, đoạn, từ vựng, và câu trong hệ thống.

2.4. Biểu đồ Usecase

2.4.1. Sơ đồ Usecase tổng quát

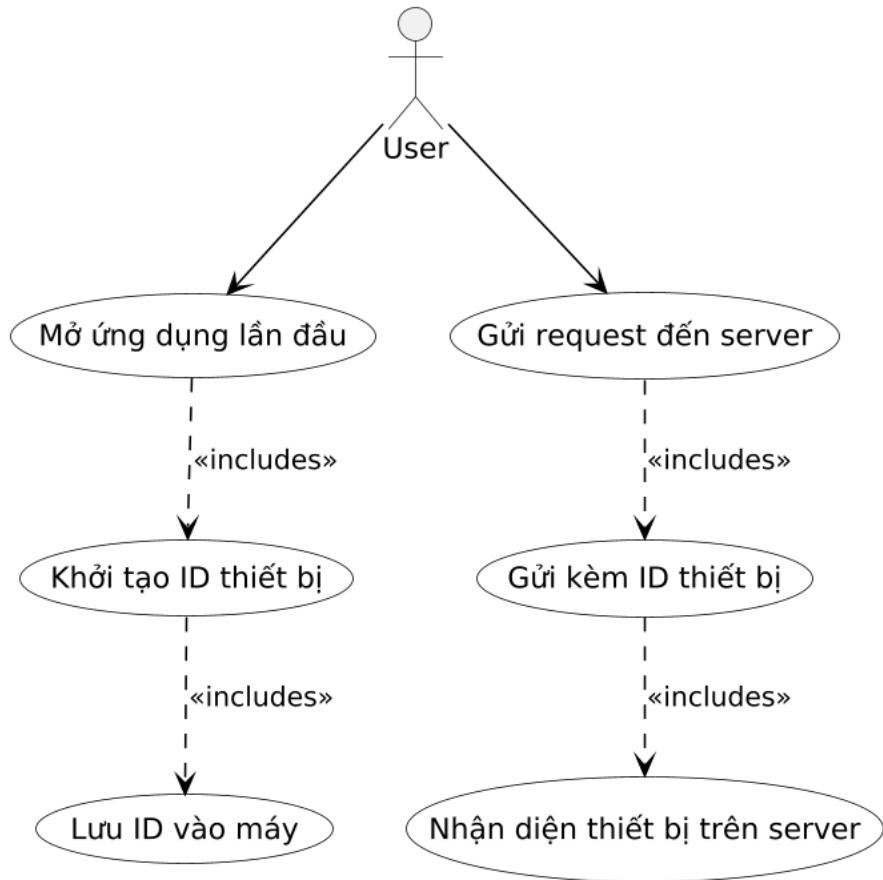


Hình 2. 3. Sơ đồ Usecase tổng quát

Sơ đồ Use Case tổng quát mô tả các chức năng chính của hệ thống dành cho người dùng. Người dùng có thể thực hiện các hành động như tìm kiếm video, xem từ bookmark, xem câu bookmark, và import video. Trong chức năng tìm kiếm video, người dùng có thể xem danh sách video, từ đó mở rộng ra xem chi tiết video. Chức năng xem video chi tiết bao gồm các hành động như xem transcript, xem bản dịch câu, xem định nghĩa từ vựng và cách phát âm, hoặc thực hiện ghi âm và chấm điểm bản đọc. Ghi âm và chấm điểm bản đọc có thể mở rộng ra các chức năng như xem điểm số, xem CEFR, nghe lại giọng đọc, và xem từ đã phát âm. Với chức năng xem từ bookmark, người dùng có thể xem định nghĩa tiếng Anh - tiếng Việt, từ đó mở rộng ra xem cách phát âm và định nghĩa từ. Tương tự, trong xem câu bookmark, người dùng có thể xem bản dịch tiếng Anh - Việt và thực hiện ghi âm chấm điểm bản đọc. Ngoài ra, người dùng có thể xem danh sách các video yêu thích, từ đó mở rộng ra xem chi tiết video. Sơ đồ này tố

chức các chức năng theo mỗi quan hệ extends, giúp thể hiện rõ các hành động chính và các chức năng mở rộng liên quan trong hệ thống.

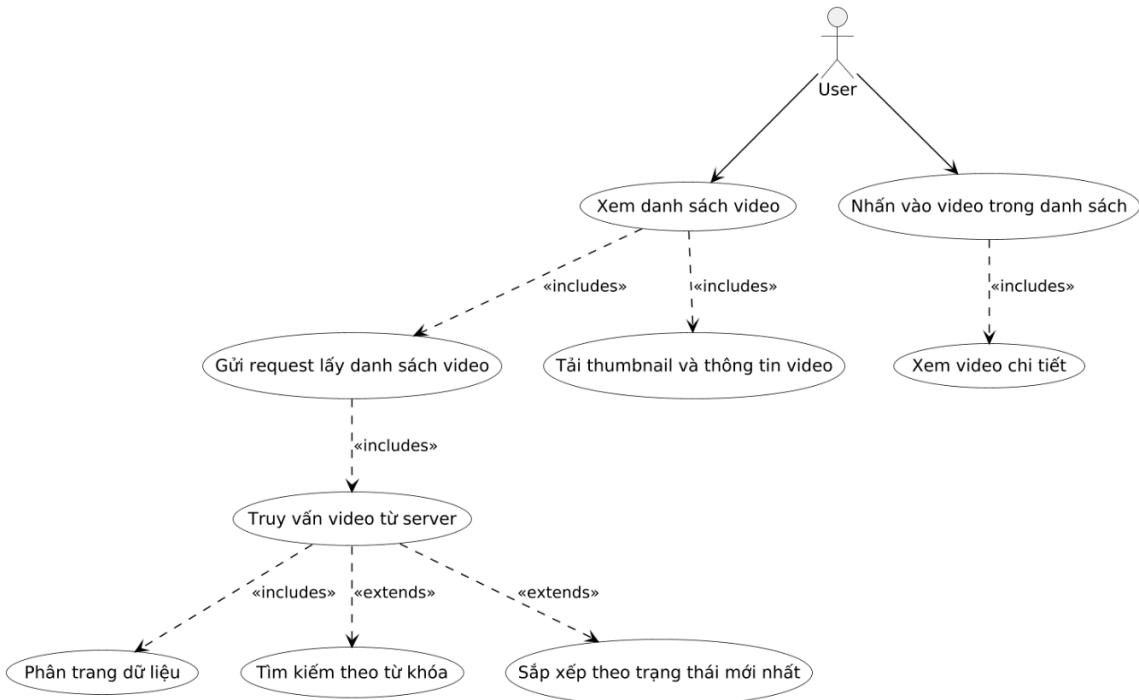
2.4.2. Usecase Khởi tạo ID và Nhận diện thiết bị



Hình 2. 4. Sơ đồ Usecase Khởi tạo ID và Nhận diện thiết bị

Sơ đồ mô tả quy trình khởi tạo ID và nhận diện thiết bị trong hệ thống. Người dùng có hai luồng chính: mở ứng dụng lần đầu và gửi request đến server. Khi người dùng mở ứng dụng lần đầu, hệ thống sẽ thực hiện các bước bao gồm khởi tạo ID thiết bị và lưu ID vào máy, trong đó các hành động này được liên kết với nhau thông qua quan hệ includes. Trong luồng gửi request đến server, hệ thống sẽ gửi kèm ID thiết bị trong request và thực hiện nhận diện thiết bị trên server, cũng thông qua quan hệ includes. Sơ đồ này thể hiện rõ cách hệ thống xử lý ID thiết bị, từ việc khởi tạo và lưu trữ trên máy đến việc gửi và nhận diện trên server, đảm bảo tính duy nhất và xác thực của thiết bị trong hệ thống.

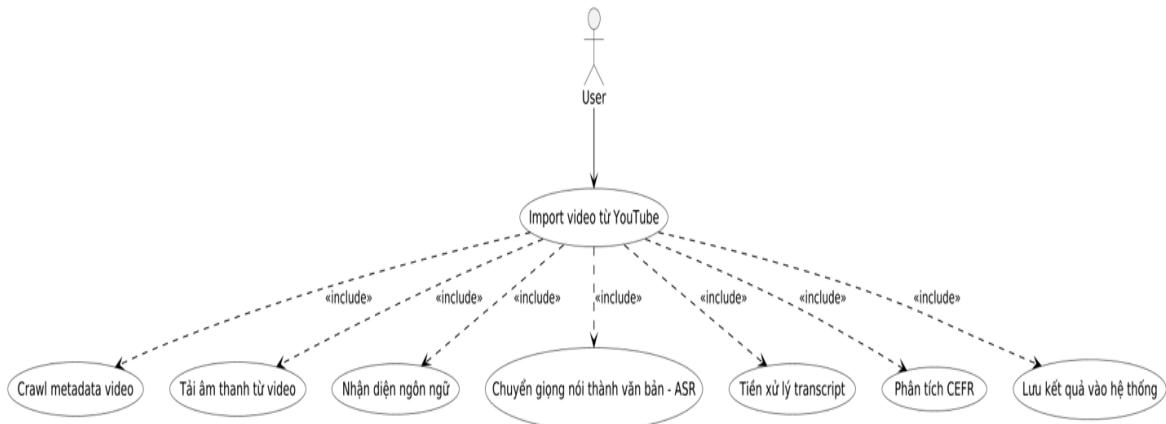
2.4.3. Usecase Xem danh sách video



Hình 2. 5. Sơ đồ Usecase Xem danh sách video

Sơ đồ Use Case "Xem danh sách video" mô tả quy trình người dùng tương tác với hệ thống để xem và quản lý danh sách video. Người dùng có thể thực hiện hành động xem danh sách video, trong đó hệ thống sẽ gửi request để lấy danh sách video từ server và tải thumbnail cùng thông tin video. Quá trình này bao gồm các bước như truy vấn video từ server, với các chức năng mở rộng như phân trang dữ liệu, tìm kiếm theo từ khóa, và sắp xếp theo trạng thái mới nhất để tối ưu hóa việc hiển thị danh sách. Ngoài ra, người dùng có thể nhấn vào video trong danh sách để xem chi tiết video, trong đó hệ thống sẽ cung cấp thông tin đầy đủ về video được chọn. Sơ đồ này sử dụng các mối quan hệ includes và extends để thể hiện rõ các bước xử lý và các chức năng mở rộng, đảm bảo trải nghiệm người dùng linh hoạt và hiệu quả khi tương tác với danh sách video.

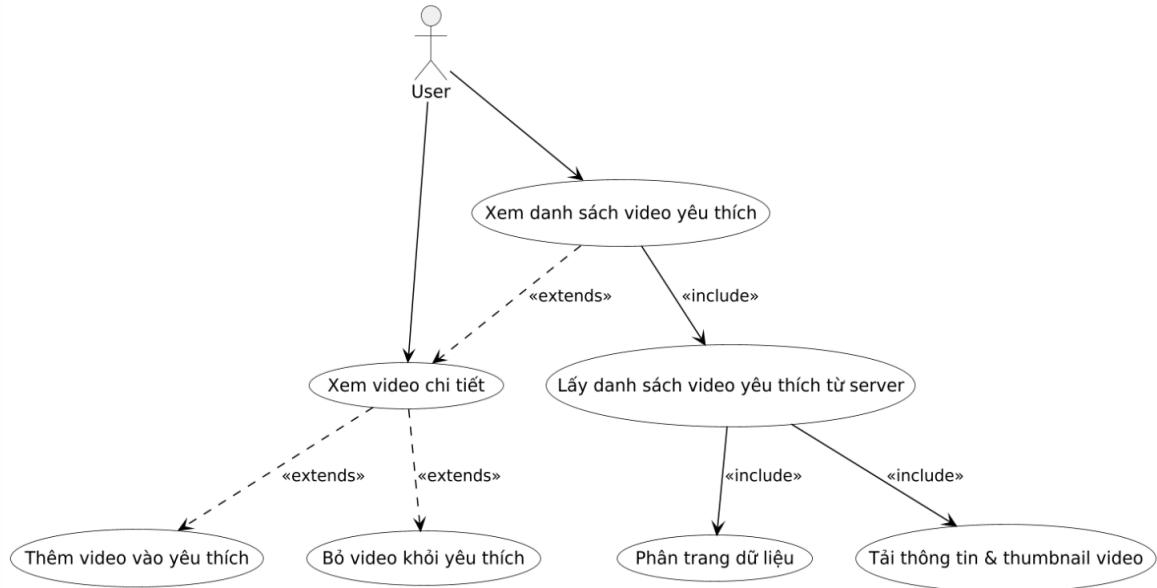
2.4.4. Usecase Import video, xử lý transcript và Phân tích CEFR



Hình 2. 6. Sơ đồ Use Case "Import video, xử lý transcript và phân tích CEFR"

Sơ đồ Use Case "Import video, xử lý transcript và phân tích CEFR" mô tả quy trình người dùng tương tác với hệ thống để nhập video từ YouTube và thực hiện các bước xử lý dữ liệu. Khi người dùng thực hiện hành động Import video từ YouTube, hệ thống sẽ bao gồm các bước xử lý chính như crawl metadata video để thu thập thông tin cơ bản về video, tải âm thanh từ video, và nhận diện ngôn ngữ của nội dung. Sau đó, hệ thống thực hiện chuyển giọng nói thành văn bản (ASR), tiếp tục với tiến xử lý transcript để chuẩn hóa nội dung. Tiếp theo, hệ thống thực hiện phân tích CEFR để đánh giá độ khó của nội dung theo chuẩn CEFR. Cuối cùng, kết quả được lưu vào hệ thống để phục vụ cho các chức năng khác như hiển thị và tra cứu. Sơ đồ này sử dụng quan hệ includes để thể hiện các bước xử lý liên kết chặt chẽ, đảm bảo quy trình nhập và phân tích video diễn ra một cách toàn diện và hiệu quả.

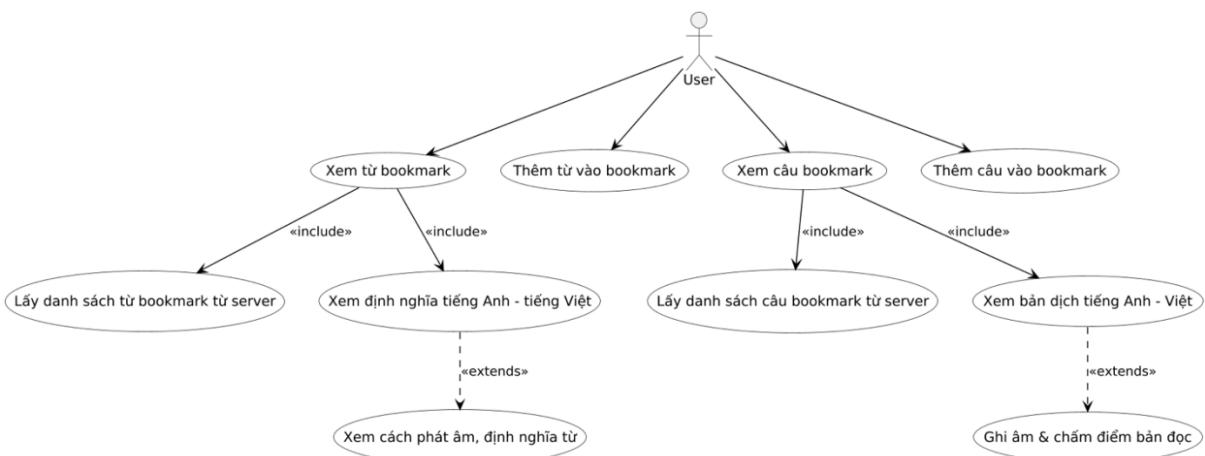
2.4.5. Usecase Yêu thích video



Hình 2. 7. Sơ đồ Usecase Yêu thích video

Sơ đồ UseCase mô tả quy trình người dùng tương tác với hệ thống để quản lý danh sách video yêu thích. Người dùng có thể thực hiện hành động xem danh sách video yêu thích, trong đó hệ thống sẽ gửi request để lấy danh sách video yêu thích từ server. Quá trình này bao gồm các bước như phân trang dữ liệu để hiển thị danh sách một cách hợp lý và tải thông tin & thumbnail video để cung cấp thông tin trực quan cho người dùng. Ngoài ra, người dùng có thể xem video chi tiết, từ đó mở rộng ra các hành động như thêm video vào yêu thích hoặc bỏ video khỏi yêu thích. Sơ đồ này sử dụng các mối quan hệ includes và extends để thể hiện rõ các bước xử lý và các chức năng mở rộng, đảm bảo người dùng có trải nghiệm linh hoạt và hiệu quả khi quản lý danh sách video yêu thích.

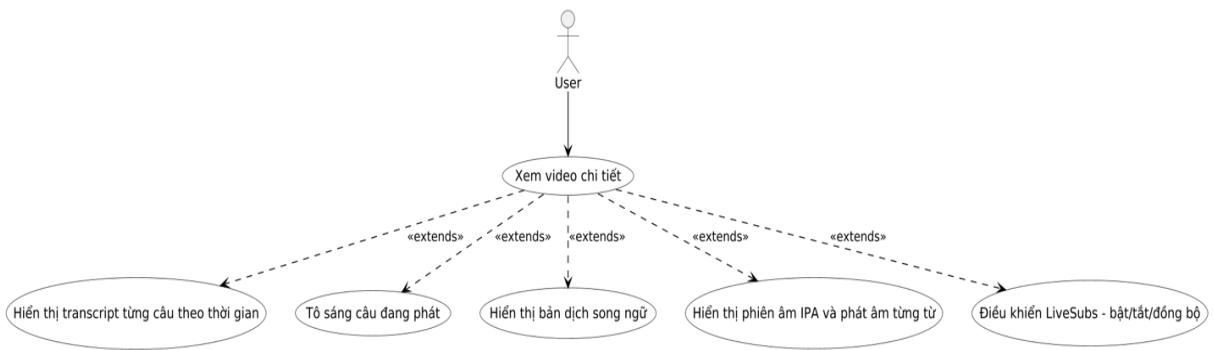
2.4.6. Usecase bookmark



Hình 2. 8. Sơ đồ Usecase bookmark

Sơ đồ Use Case "Bookmark" mô tả quy trình người dùng tương tác với hệ thống để quản lý từ và câu được đánh dấu. Người dùng có thể thực hiện các hành động xem từ bookmark và xem câu bookmark, trong đó hệ thống sẽ gửi request để lấy danh sách từ bookmark từ server hoặc lấy danh sách câu bookmark từ server. Khi xem từ bookmark, người dùng có thể mở rộng ra các chức năng như xem định nghĩa tiếng Anh - tiếng Việt và xem cách phát âm, định nghĩa từ. Tương tự, khi xem câu bookmark, người dùng có thể mở rộng ra các chức năng như xem bản dịch tiếng Anh - Việt và ghi âm & chấm điểm bản đọc. Ngoài ra, người dùng có thể thực hiện các hành động thêm từ vào bookmark hoặc thêm câu vào bookmark để lưu trữ nội dung yêu thích. Sơ đồ này sử dụng các mối quan hệ includes và extends để thể hiện rõ các bước xử lý và các chức năng mở rộng, giúp người dùng dễ dàng quản lý và tra cứu nội dung đã bookmark.

2.4.7. Usecase Hiển thị transcript và Bộ điều khiển LiveSubs

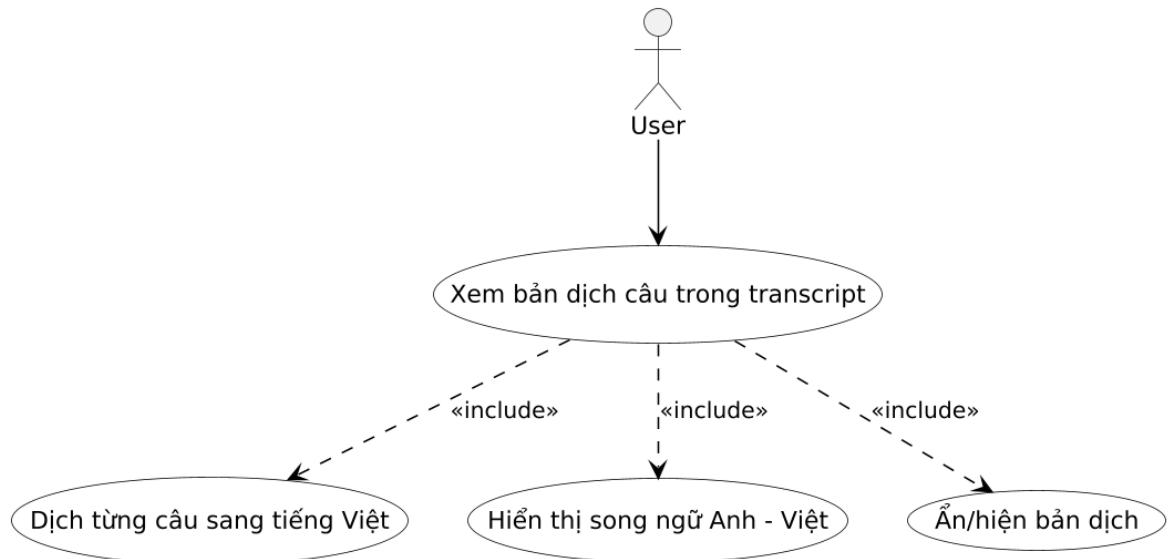


Hình 2. 9. Sơ đồ Usecase Hiển thị transcript và Bộ điều khiển LiveSubs

Sơ đồ Use Case "Hiển thị transcript và Bộ điều khiển LiveSubs" mô tả quy trình người dùng tương tác với hệ thống để xem chi tiết video. Khi thực hiện hành động xem video chi tiết, hệ thống cung cấp các chức năng mở rộng như hiển thị transcript từng câu theo thời gian, giúp người dùng theo dõi nội dung video một cách trực quan. Ngoài ra, hệ thống hỗ trợ tô sáng câu đang phát, giúp người dùng dễ dàng nhận biết câu hiện tại trong video. Người dùng cũng có thể sử dụng chức năng hiển thị bản dịch song ngữ để xem nội dung video bằng cả tiếng Anh và tiếng Việt, hoặc hiển thị phiên âm IPA và phát âm từng từ để hỗ trợ học phát âm. Cuối cùng, hệ thống cung cấp bộ điều khiển LiveSubs, cho phép người dùng bật, tắt, hoặc đồng bộ phụ đề theo thời gian thực. Sơ đồ này sử

dụng các mối quan hệ extends để thể hiện rõ các chức năng mở rộng, giúp người dùng có trải nghiệm toàn diện và linh hoạt khi xem video chi tiết.

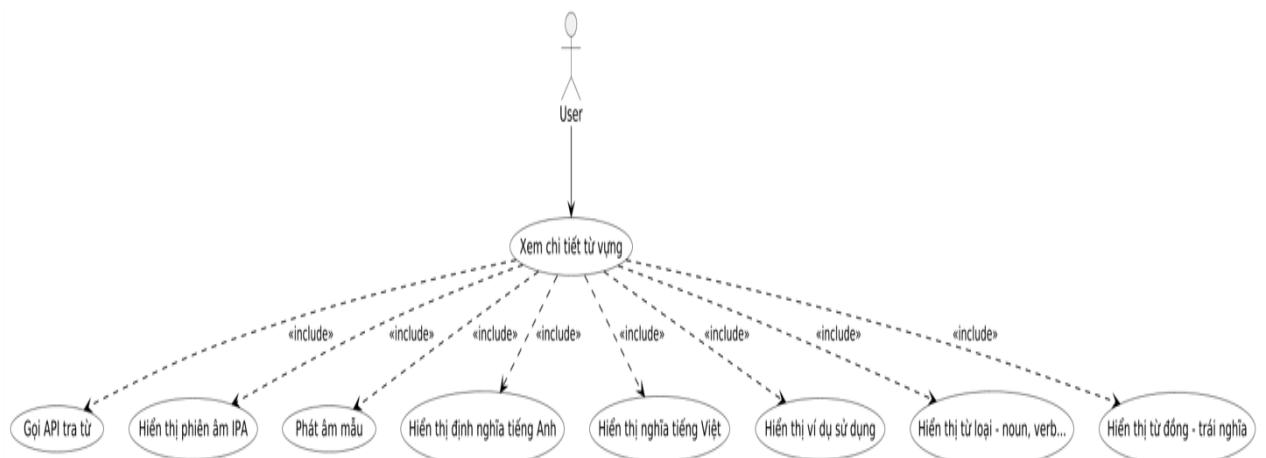
2.4.8. Usecase xem translation



Hình 2. 10. Sơ đồ Usecase xem translation

Sơ đồ mô tả quy trình người dùng tương tác với hệ thống để xem nội dung dịch của từng câu trong video. Khi thực hiện hành động xem bản dịch câu trong transcript, hệ thống sẽ bao gồm các chức năng chính như dịch từng câu sang tiếng Việt, giúp người dùng hiểu rõ nội dung câu. Ngoài ra, hệ thống hỗ trợ hiển thị song ngữ Anh - Việt, cho phép người dùng xem đồng thời cả bản gốc và bản dịch để tiện so sánh. Người dùng cũng có thể sử dụng chức năng ẩn/hiện bản dịch để tùy chỉnh giao diện theo nhu cầu. Sơ đồ này sử dụng quan hệ includes để thể hiện các bước xử lý liên kết chặt chẽ, đảm bảo trải nghiệm người dùng linh hoạt và hiệu quả khi xem bản dịch trong transcript.

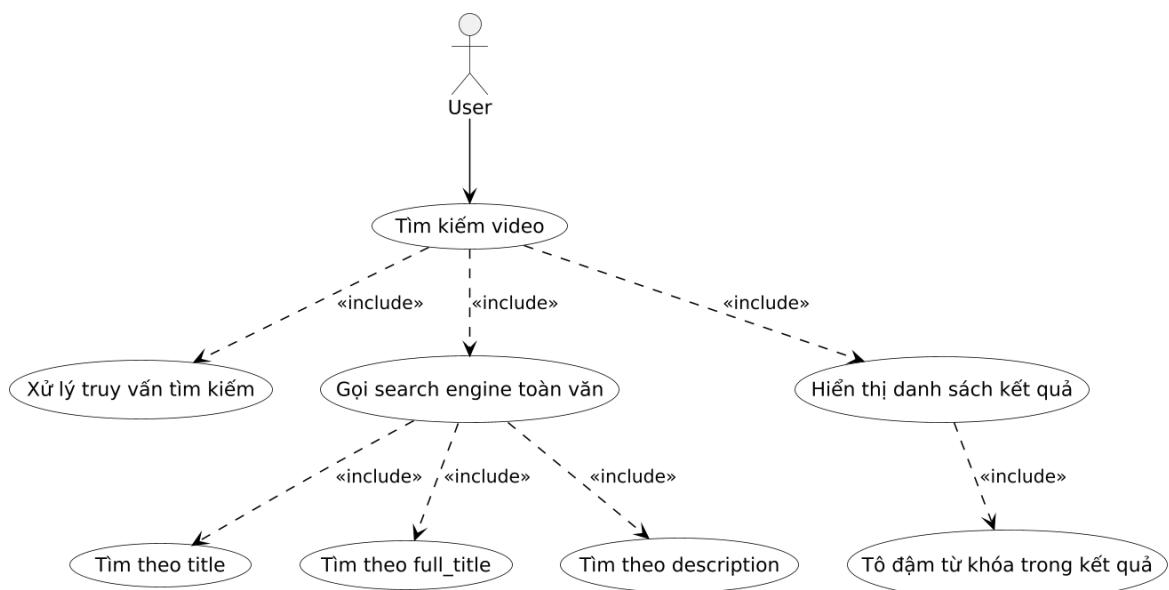
2.4.9. Usecase Xem phát âm và định nghĩa từ vựng (FreeDictionary API)



Hình 2. 11. Sơ đồ Usecase Xem phát âm và định nghĩa từ vựng

Sơ đồ mô tả quy trình người dùng tương tác với hệ thống để tra cứu chi tiết từ vựng thông qua FreeDictionary API. Khi thực hiện hành động xem chi tiết từ vựng, hệ thống sẽ gọi API để tra cứu thông tin từ vựng, bao gồm các chức năng như hiển thị phiên âm IPA và phát âm mẫu, giúp người dùng học cách phát âm chính xác. Ngoài ra, hệ thống cung cấp các thông tin chi tiết như hiển thị định nghĩa tiếng Anh, hiển thị định nghĩa tiếng Việt, và hiển thị ví dụ sử dụng để người dùng hiểu rõ ngữ nghĩa và cách dùng từ. Hệ thống cũng hỗ trợ hiển thị từ loại (noun, verb, etc.) và hiển thị từ đồng - trái nghĩa, giúp người dùng mở rộng vốn từ vựng. Sơ đồ sử dụng quan hệ includes để thể hiện các bước xử lý liên kết chặt chẽ, đảm bảo cung cấp thông tin từ vựng đầy đủ và chính xác thông qua FreeDictionary API.

2.4.10. Usecase Tìm kiếm toàn văn bản

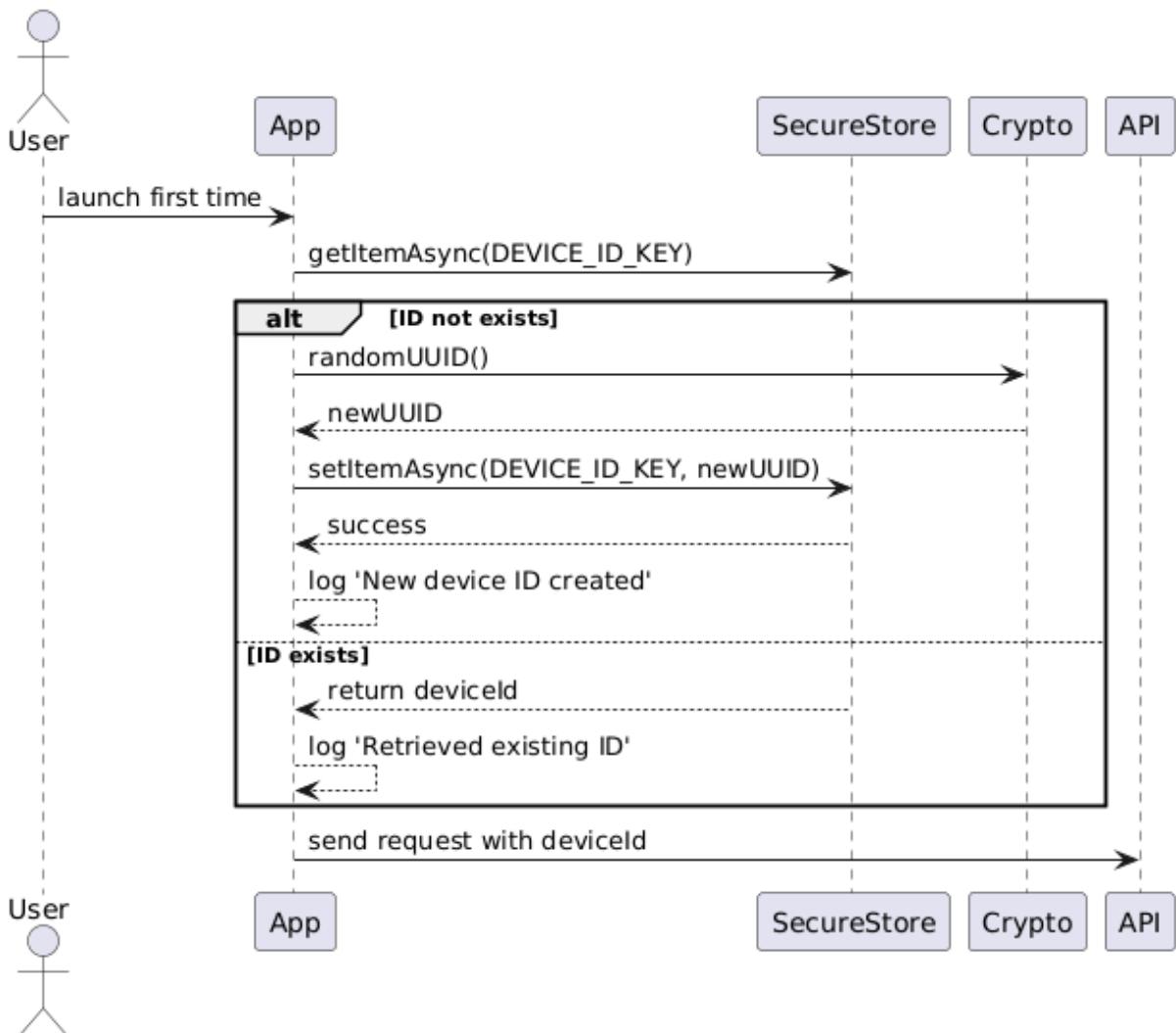


Hình 2. 12. Sơ đồ. Usecase Tìm kiếm toàn văn bản

Sơ đồ mô tả quy trình người dùng tương tác với hệ thống để tìm kiếm video dựa trên nội dung toàn văn. Khi thực hiện hành động tìm kiếm video, hệ thống sẽ bao gồm các bước như xử lý truy vấn tìm kiếm để phân tích và chuẩn hóa từ khóa tìm kiếm của người dùng. Sau đó, hệ thống sẽ gọi search engine toàn văn, cho phép tìm kiếm theo các tiêu chí như title, full_title, và description để đảm bảo kết quả tìm kiếm chính xác và đầy đủ. Kết quả tìm kiếm sẽ được hiển thị thông qua chức năng hiển thị danh sách kết quả, trong đó hệ thống hỗ trợ tô đậm từ khóa trong kết quả để người dùng dễ dàng nhận diện nội dung liên quan. Sơ đồ sử dụng quan hệ includes để thể hiện các bước xử lý liên kết chặt chẽ, đảm bảo trải nghiệm tìm kiếm toàn diện và hiệu quả.

2.5. Biểu đồ tuần tự

2.5.1. Khởi tạo ID và Nhận diện thiết bị

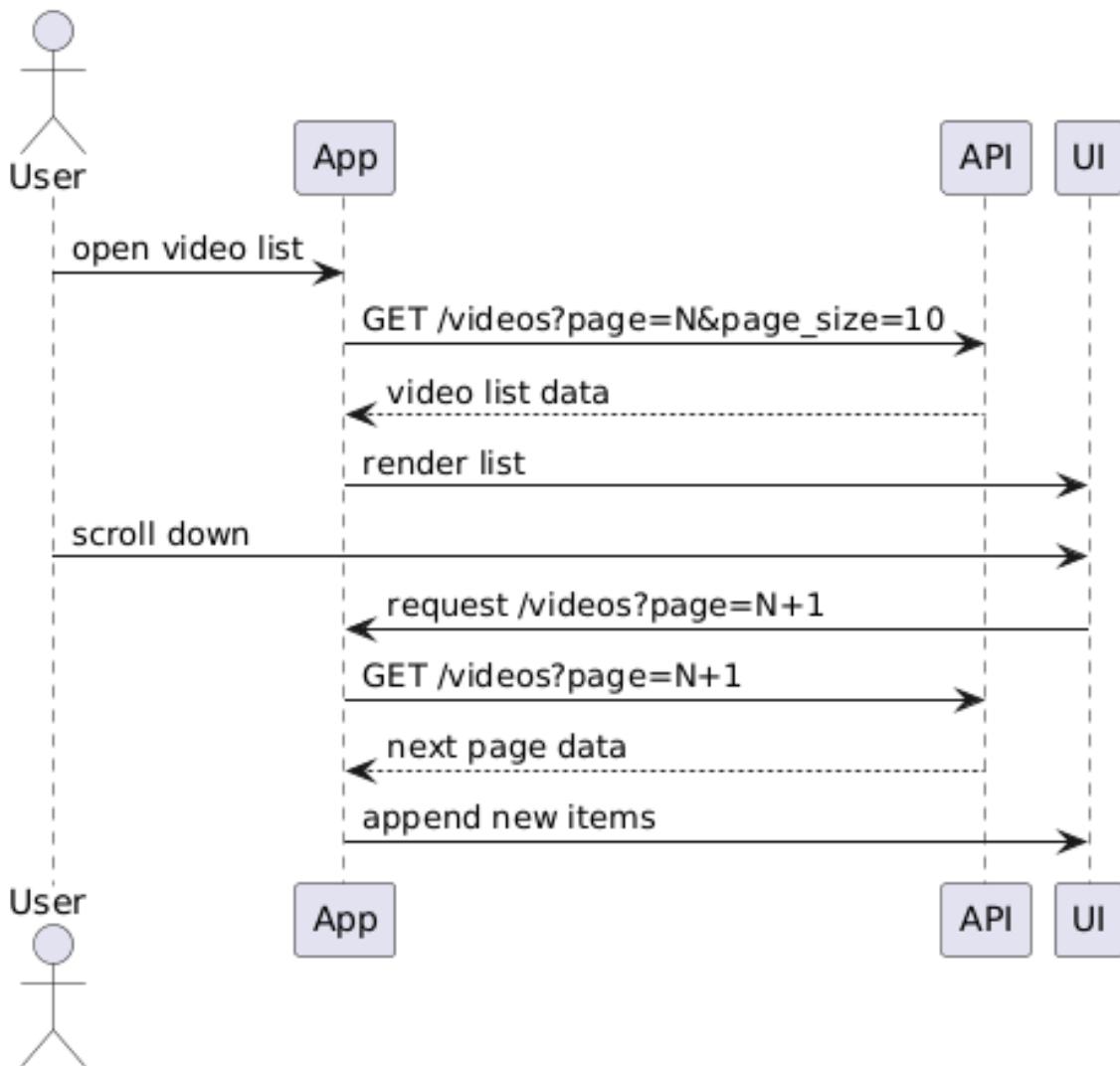


Hình 2. 13. Biểu đồ tuần tự Khởi tạo ID và Nhận diện thiết bị

Sơ đồ tuần tự "Khởi tạo ID và Nhận diện thiết bị" mô tả quy trình hệ thống xử lý khi người dùng khởi chạy ứng dụng lần đầu. Đầu tiên, ứng dụng kiểm tra xem DEVICE_ID_KEY đã tồn tại trong SecureStore hay chưa bằng cách gọi hàm `getItemAsync(DEVICE_ID_KEY)`. Nếu ID chưa tồn tại, hệ thống sẽ tạo một UUID mới thông qua hàm `randomUUID()` và lưu UUID này vào SecureStore bằng cách gọi `setItemAsync(DEVICE_ID_KEY, newUUID)`. Sau khi lưu thành công, hệ thống ghi log với nội dung "New device ID created". Ngược lại, nếu ID đã tồn tại, hệ thống sẽ lấy ID từ SecureStore và trả về giá trị deviceId, đồng thời ghi log "Retrieved existing ID". Cuối cùng, ứng dụng sử dụng deviceId để gửi yêu cầu đến API nhằm nhận diện thiết bị. Quy

trình này đảm bảo rằng mỗi thiết bị được gán một ID duy nhất hoặc sử dụng lại ID đã tồn tại, giúp hệ thống nhận diện thiết bị một cách chính xác và an toàn.

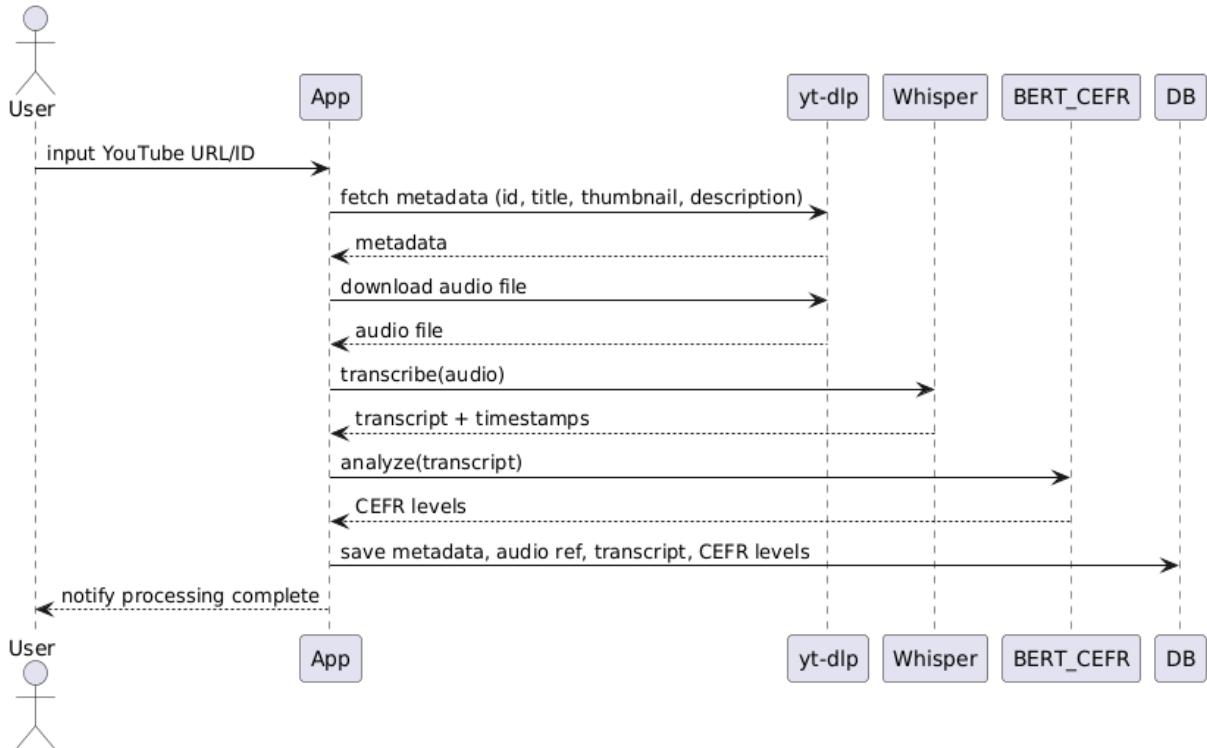
2.5.2. Xem danh sách video



Hình 2. 14. Biểu đồ tuần tự Xem danh sách video

Sơ đồ mô tả quy trình người dùng tương tác với hệ thống để hiển thị danh sách video. Khi người dùng mở danh sách video, ứng dụng gửi yêu cầu GET /videos?page=N&page_size=10 đến API để lấy dữ liệu của trang đầu tiên. API trả về danh sách video, sau đó ứng dụng hiển thị danh sách này trên giao diện người dùng (UI). Khi người dùng cuộn xuống để xem thêm video, ứng dụng gửi yêu cầu tiếp theo GET /videos?page=N+1 đến API để lấy dữ liệu của trang kế tiếp. API trả về dữ liệu của trang mới, và ứng dụng thêm các mục mới vào danh sách hiện tại trên UI. Quy trình này lặp lại mỗi khi người dùng cuộn xuống, đảm bảo danh sách video được tải động và hiển thị liên tục.

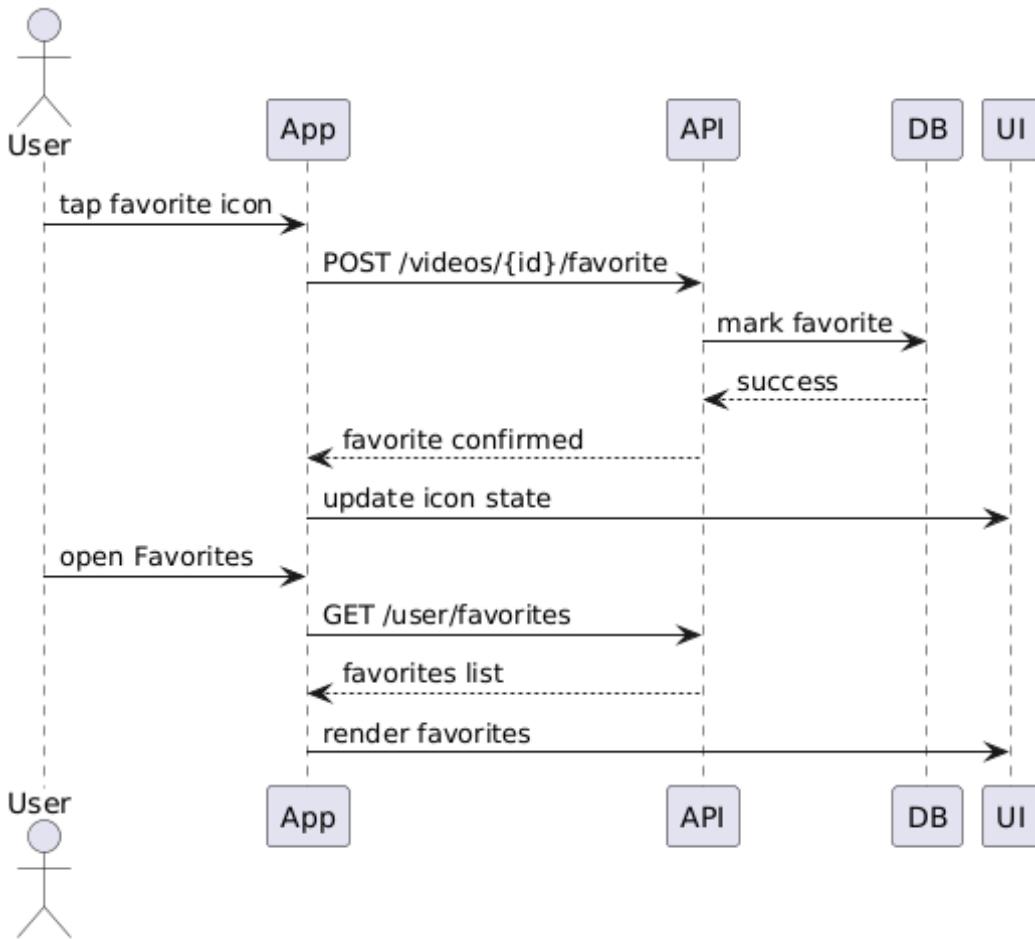
2.5.3. Import video, xử lý transcript và Phân tích CEFR



Hình 2. 15. Biểu đồ tuần tự Import video, xử lý transcript và Phân tích CEFR

Sơ đồ tuần tự "Import video, xử lý transcript và Phân tích CEFR" mô tả quy trình hệ thống xử lý khi người dùng nhập URL hoặc ID của video YouTube. Đầu tiên, ứng dụng gửi yêu cầu đến công cụ yt-dlp để lấy metadata của video, bao gồm ID, tiêu đề, thumbnail, và mô tả. Sau khi nhận được metadata, ứng dụng tiếp tục sử dụng yt-dlp để tải file âm thanh của video. File âm thanh này được gửi đến công cụ Whisper để thực hiện chuyển đổi giọng nói thành văn bản (transcribe), kèm theo các timestamp. Tiếp theo, văn bản transcript được gửi đến mô hình BERT_CEFR để phân tích và đánh giá mức độ CEFR (Common European Framework of Reference for Languages). Kết quả phân tích bao gồm các cấp độ CEFR của nội dung video. Cuối cùng, ứng dụng lưu metadata, file âm thanh, transcript, và các cấp độ CEFR vào cơ sở dữ liệu (DB). Sau khi hoàn tất, hệ thống thông báo cho người dùng rằng quá trình xử lý đã hoàn thành. Quy trình này đảm bảo việc nhập video, xử lý transcript, và phân tích CEFR diễn ra một cách tự động và hiệu quả.

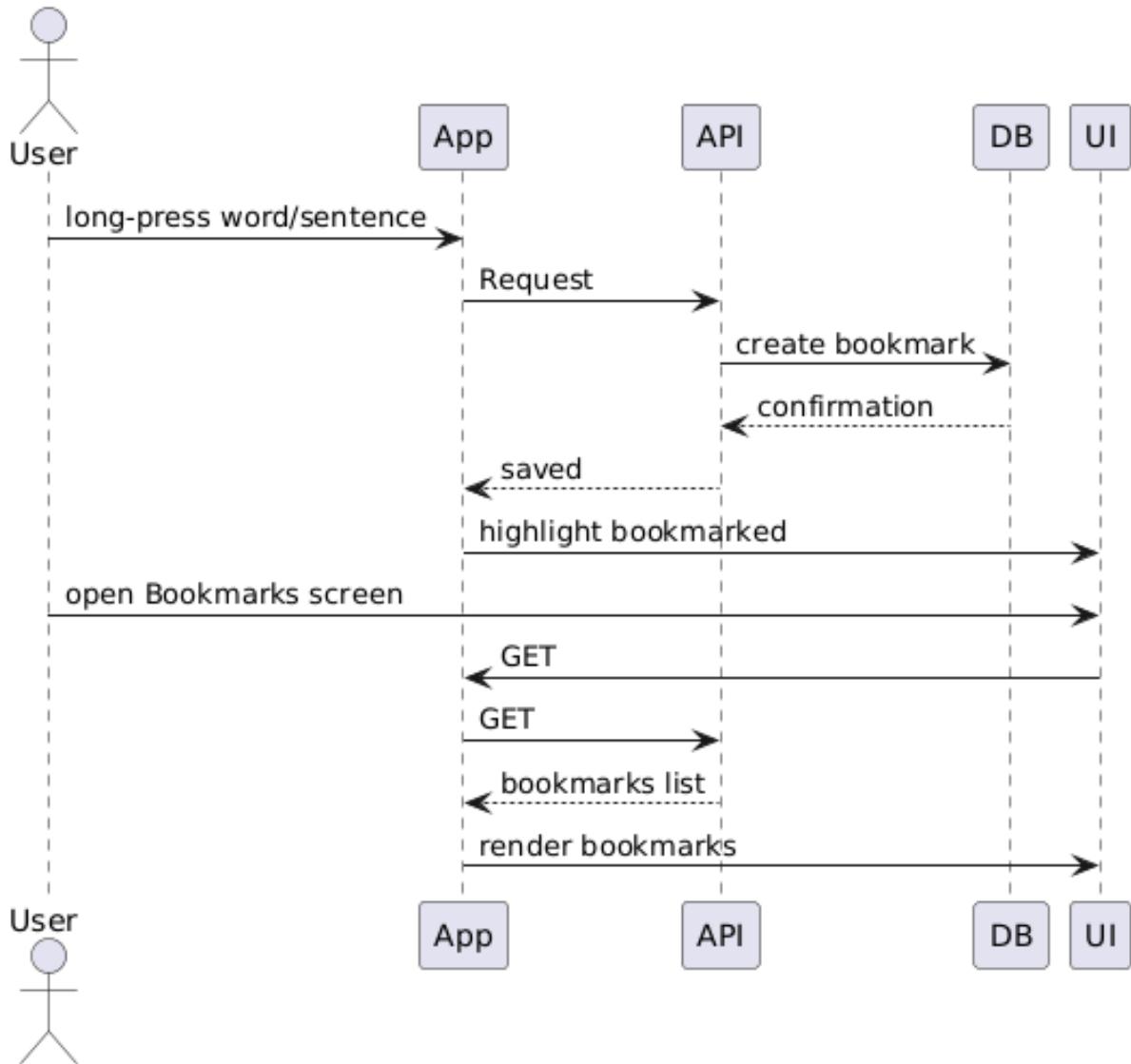
2.5.4. Yêu thích video



Hình 2. 16. Biểu đồ tuần tự Yêu thích video

Sơ đồ mô tả quy trình người dùng tương tác với hệ thống để đánh dấu video yêu thích và xem danh sách video đã yêu thích. Khi người dùng nhấn vào biểu tượng yêu thích trên một video, ứng dụng gửi yêu cầu POST `/videos/{id}/favorite` đến API để đánh dấu video đó là yêu thích. API cập nhật trạng thái yêu thích trong cơ sở dữ liệu (DB) và trả về phản hồi thành công. Sau đó, ứng dụng cập nhật trạng thái biểu tượng yêu thích trên giao diện người dùng (UI). Khi người dùng mở danh sách video yêu thích, ứng dụng gửi yêu cầu GET `/user/favorites` đến API để lấy danh sách các video đã được đánh dấu yêu thích. API truy vấn cơ sở dữ liệu để lấy danh sách video và trả về cho ứng dụng. Cuối cùng, ứng dụng hiển thị danh sách video yêu thích trên giao diện người dùng. Quy trình này đảm bảo việc đánh dấu và quản lý video yêu thích diễn ra một cách mượt mà và hiệu quả.

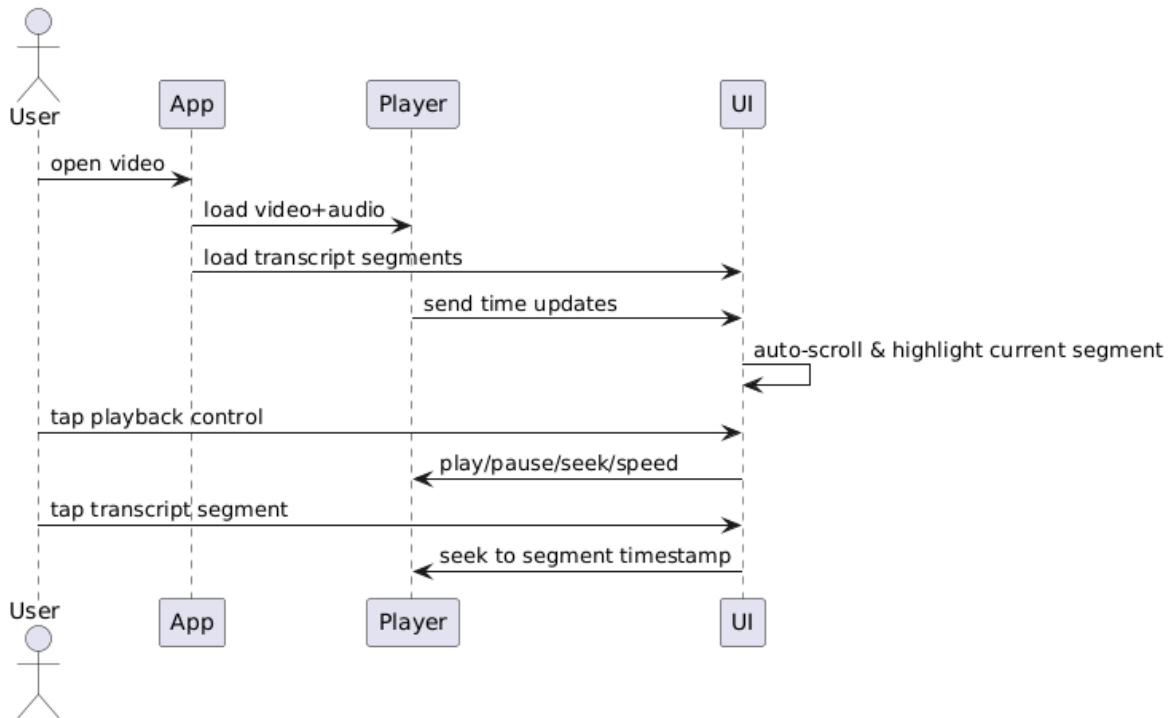
2.5.5. Bookmark từ vựng và câu



Hình 2. 17. Biểu đồ tuần tự Bookmark từ vựng và câu

Sơ đồ tuần tự "Bookmark từ vựng và câu" mô tả quy trình người dùng tương tác với hệ thống để đánh dấu từ hoặc câu và quản lý danh sách bookmark. Khi người dùng nhấn giữ (long-press) một từ hoặc câu, ứng dụng gửi yêu cầu đến API để tạo bookmark. API xử lý yêu cầu, lưu thông tin bookmark vào cơ sở dữ liệu (DB), và trả về xác nhận rằng bookmark đã được tạo thành công. Sau đó, ứng dụng làm nổi bật từ hoặc câu đã được đánh dấu trên giao diện người dùng (UI). Khi người dùng mở màn hình danh sách bookmark, ứng dụng gửi yêu cầu GET đến API để lấy danh sách các từ hoặc câu đã được đánh dấu. API truy vấn cơ sở dữ liệu để lấy danh sách bookmark và trả về cho ứng dụng. Cuối cùng, ứng dụng hiển thị danh sách bookmark trên giao diện người dùng. Quy trình này đảm bảo việc tạo và quản lý bookmark diễn ra thuận tiện và hiệu quả.

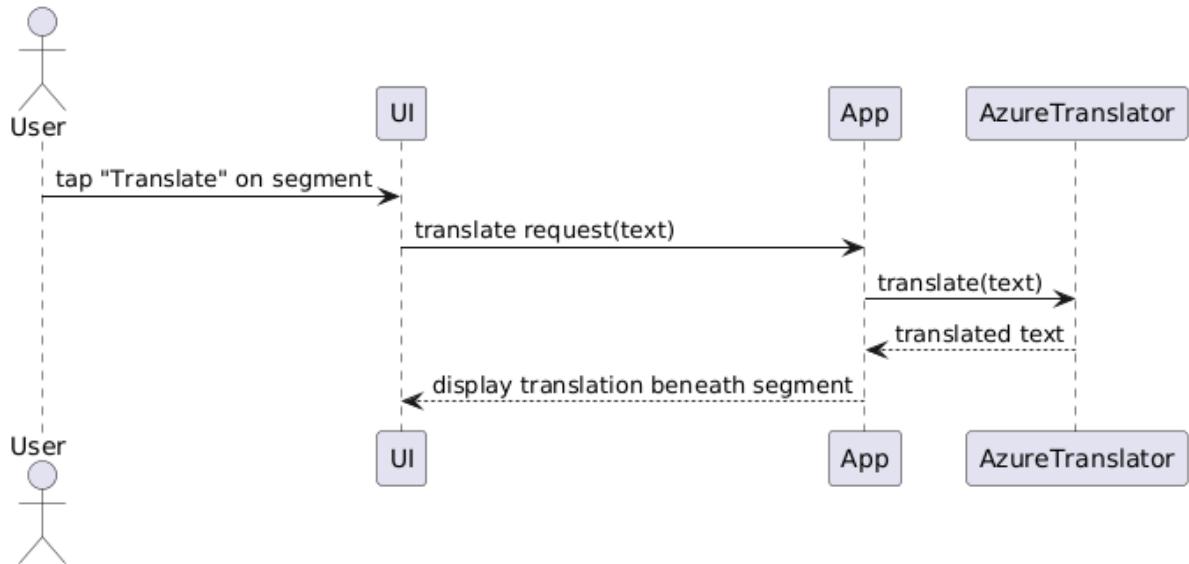
2.5.6. Hiển thị transcript và Bộ điều khiển LiveSubs



Hình 2. 18. Biểu đồ tuần tự Hiển thị transcript và Bộ điều khiển LiveSubs

Sơ mô tả quy trình người dùng tương tác với hệ thống để xem video kèm transcript và điều khiển phụ đề. Khi người dùng mở video, ứng dụng tải nội dung video và audio, đồng thời tải các đoạn transcript tương ứng. Trong quá trình phát video, trình phát (Player) gửi các cập nhật thời gian đến giao diện người dùng (UI), giúp hệ thống tự động cuộn và làm nổi bật đoạn transcript hiện tại. Người dùng có thể tương tác với các điều khiển phát lại, như phát/dừng, tua, hoặc thay đổi tốc độ phát, thông qua giao diện. Ngoài ra, khi người dùng nhấn vào một đoạn transcript, hệ thống sẽ điều chỉnh trình phát để tua đến timestamp tương ứng của đoạn đó. Quy trình này đảm bảo sự đồng bộ giữa video, audio, và transcript, mang lại trải nghiệm học tập và theo dõi nội dung hiệu quả.

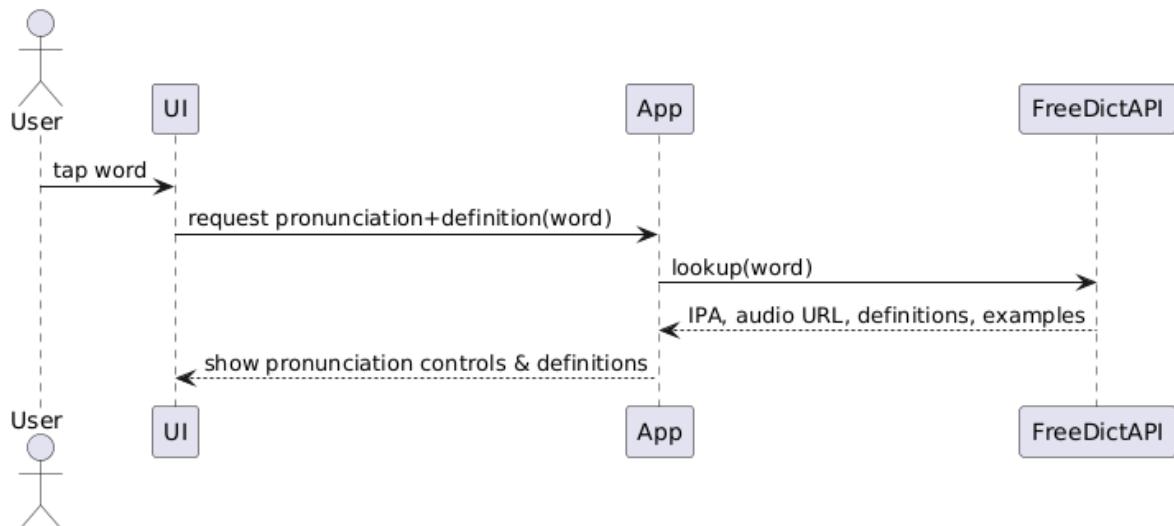
2.5.7. Dịch câu (Azure Translator)



Hình 2. 19. Biểu đồ tuần tự Dịch câu (Azure Translator)

Sơ đồ mô tả quy trình người dùng tương tác với hệ thống để dịch một đoạn văn bản. Khi người dùng nhấn vào nút "Translate" trên một đoạn transcript, giao diện người dùng (UI) gửi yêu cầu dịch văn bản đến ứng dụng. Ứng dụng tiếp tục gửi yêu cầu đến dịch vụ Azure Translator với nội dung văn bản cần dịch. Azure Translator xử lý yêu cầu và trả về văn bản đã được dịch. Sau khi nhận được kết quả, ứng dụng hiển thị bản dịch bên dưới đoạn transcript tương ứng trên giao diện người dùng (UI). Quy trình này đảm bảo việc dịch câu diễn ra nhanh chóng và chính xác, mang lại trải nghiệm tiện lợi cho người dùng.

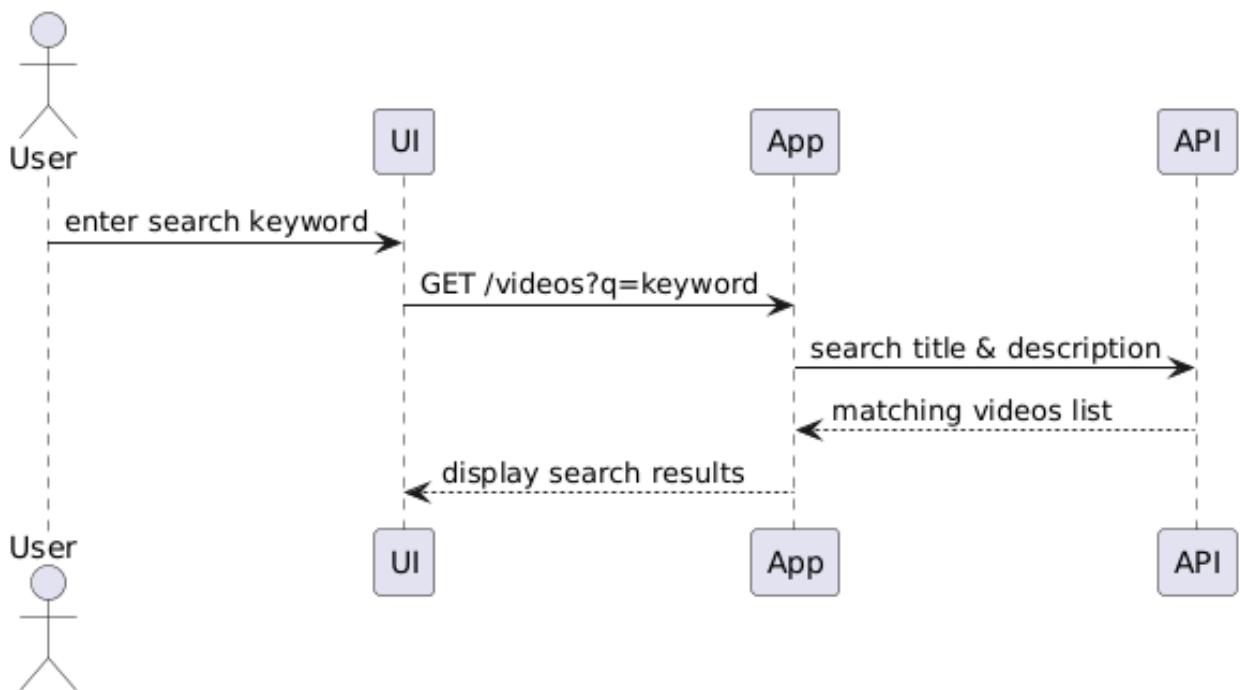
2.5.8. Xem phát âm và định nghĩa từ vựng (FreeDictionary API)



Hình 2. 20. Biểu đồ tuần tự Xem phát âm và định nghĩa từ vựng

Sơ đồ mô tả quy trình người dùng tra cứu thông tin chi tiết về một từ. Khi người dùng nhấn vào một từ trên giao diện người dùng (UI), ứng dụng gửi yêu cầu đến FreeDictionary API để tra cứu thông tin từ vựng. API xử lý yêu cầu và trả về các dữ liệu bao gồm phiên âm IPA, URL file âm thanh phát âm, định nghĩa từ, và các ví dụ sử dụng. Sau khi nhận được dữ liệu từ API, ứng dụng hiển thị các thông tin này trên giao diện người dùng, bao gồm các điều khiển phát âm và phần định nghĩa từ. Quy trình này giúp người dùng dễ dàng tra cứu và học từ vựng một cách trực quan và hiệu quả.

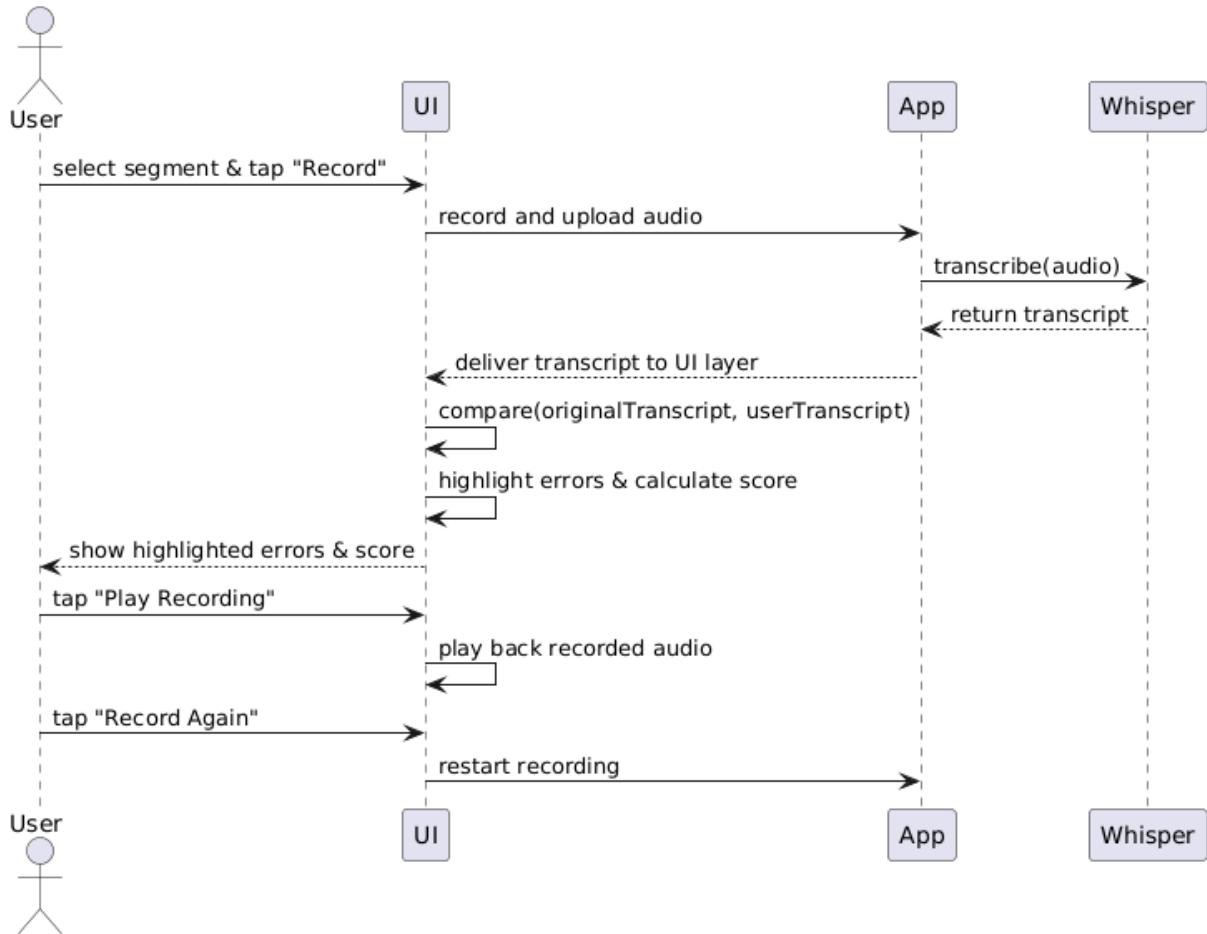
2.5.9. Tìm kiếm toàn văn bản (Title và Description)



Hình 2. 21. Biểu đồ tuần tự Tìm kiếm toàn văn bản (Title và Description)

Sơ đồ tuần tự "Tìm kiếm toàn văn bản (Title và Description)" mô tả quy trình người dùng tìm kiếm video dựa trên từ khóa. Khi người dùng nhập từ khóa tìm kiếm trên giao diện người dùng (UI), ứng dụng gửi yêu cầu GET /videos?q=keyword đến API. API xử lý yêu cầu bằng cách tìm kiếm từ khóa trong tiêu đề (title) và mô tả (description) của các video. Sau khi tìm được danh sách các video phù hợp, API trả về danh sách này cho ứng dụng. Ứng dụng sau đó hiển thị kết quả tìm kiếm trên giao diện người dùng (UI). Quy trình này đảm bảo việc tìm kiếm diễn ra nhanh chóng và chính xác, giúp người dùng dễ dàng tìm thấy nội dung mong muốn.

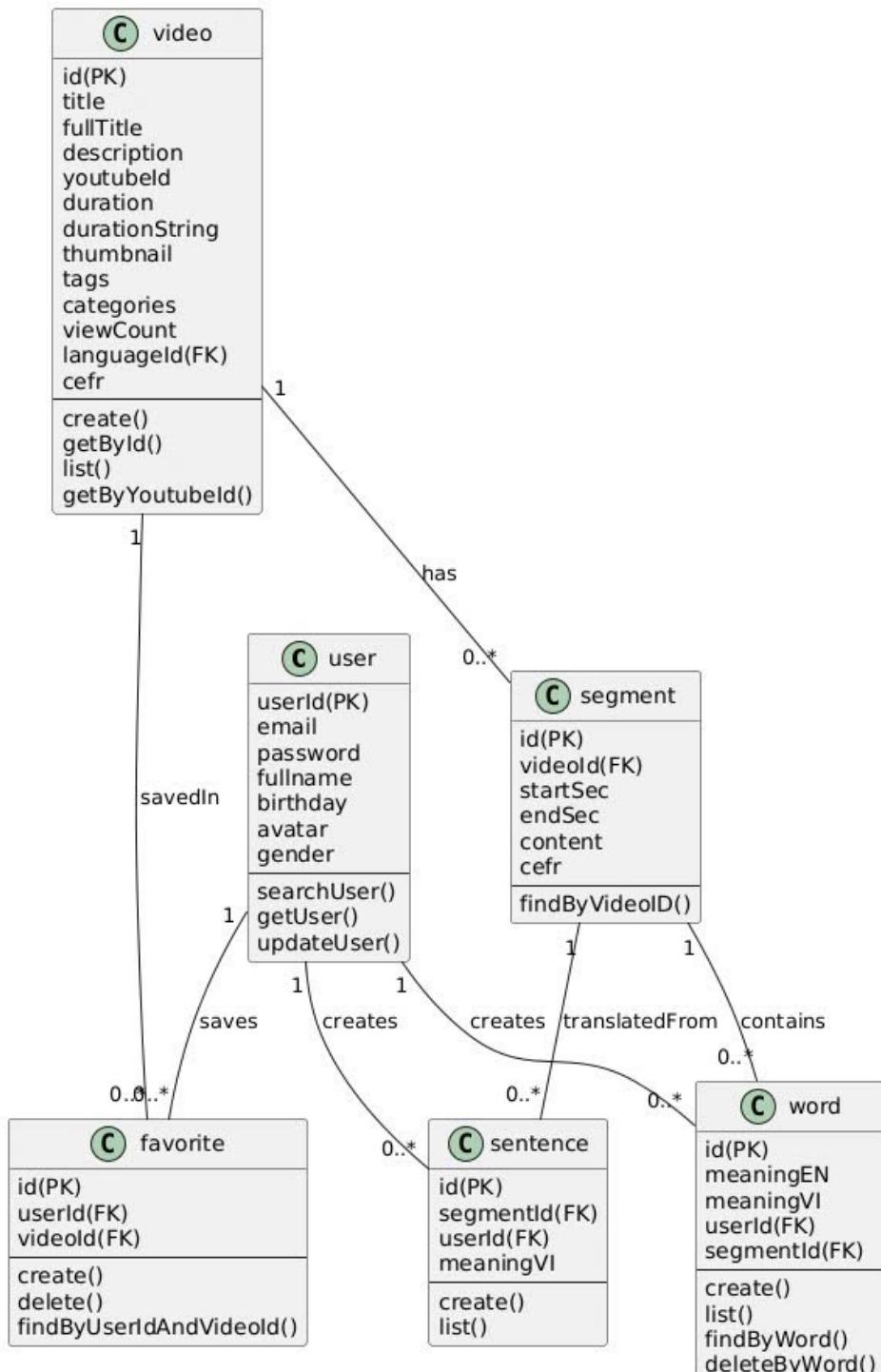
2.5.10.Ghi âm user và Chấm điểm đọc (Whisper)



Hình 2. 22. Biểu đồ tuần tự Ghi âm user và Chấm điểm đọc (Whisper)

Sơ đồ tuần tự "Ghi âm user và Chấm điểm đọc (Whisper)" mô tả quy trình người dùng ghi âm và nhận đánh giá về khả năng đọc. Khi người dùng chọn một đoạn transcript và nhấn "Record", ứng dụng bắt đầu ghi âm và tải file âm thanh lên. File âm thanh này được gửi đến công cụ Whisper để chuyển đổi giọng nói thành văn bản (transcribe). Whisper trả về transcript của người dùng. Ứng dụng so sánh transcript của người dùng với transcript gốc, xác định các lỗi và tính điểm dựa trên mức độ chính xác. Kết quả, bao gồm các lỗi được làm nổi bật và điểm số, được hiển thị trên giao diện người dùng (UI). Người dùng có thể nhấn "Play Recording" để nghe lại bản ghi âm hoặc nhấn "Record Again" để ghi âm lại. Quy trình này giúp người dùng cải thiện khả năng đọc và phát âm thông qua phản hồi chi tiết và trực quan.

2.6. Sơ đồ lớp



Hình 2. 23. Sơ đồ lớp

Sơ đồ Class Diagram mô tả cấu trúc dữ liệu và mối quan hệ giữa các lớp trong hệ thống. Lớp Video chứa thông tin về video và có mối quan hệ 1-N với lớp Segment, đại diện cho các đoạn nội dung của video. Lớp User lưu thông tin người dùng, cho phép họ lưu video yêu thích thông qua lớp Favorite và tạo các câu dịch từ nội dung video thông

qua lớp Sentence. Lớp Segment liên kết với lớp Word, đại diện cho các từ vựng trong đoạn, và lớp Sentence, đại diện cho các câu dịch từ đoạn. Mỗi từ (Word) và câu (Sentence) đều có thể được tra cứu hoặc quản lý thông qua các phương thức tương ứng. Sơ đồ này thể hiện cách các lớp tương tác để hỗ trợ các chức năng như quản lý video, lưu yêu thích, dịch nội dung, và tra cứu từ vựng.

CHƯƠNG 3. PHÁT TRIỂN ỨNG DỤNG

3.1. Kiến trúc tổng thể hệ thống

Hệ thống học tiếng Anh theo phương pháp shadowing được thiết kế theo mô hình client-server kết hợp microservice, giúp đảm bảo hiệu năng, khả năng mở rộng và dễ bảo trì. Hệ thống bao gồm 4 thành phần chính

3.1.1. Client (Ứng dụng di động)

- Ứng dụng được phát triển bằng React Native, hướng đến trải nghiệm học tập đơn giản, tập trung, phù hợp trên thiết bị di động.
- Các chức năng chính gồm:
 - Nhập link hoặc ID video YouTube để bắt đầu phân tích
 - Xem video kèm phụ đề trực tiếp theo từng câu
 - Xem bản dịch và hướng dẫn phát âm từng từ
 - Ghi âm giọng đọc và nhận phản hồi điểm số ngay lập tức
 - Nhận góp ý chi tiết cho từng từ phát âm sai

3.1.2. Server chính (Backend API)

- Được phát triển bằng GoLang, đây là thành phần trung tâm chịu trách nhiệm xử lý nghiệp vụ và kết nối giữa client và các dịch vụ AI.
- Chức năng:
 - Cung cấp API cho ứng dụng mobile
 - Quản lý người dùng, phiên học, video, và dữ liệu transcript
 - Phối hợp pipeline xử lý audio qua hàng đợi công việc
 - Truy xuất và ghi dữ liệu vào hệ quản trị cơ sở dữ liệu (PostgreSQL)

3.1.3. Dịch vụ xử lý ngôn ngữ (AI Microservices)

Các tác vụ chuyên sâu về ngôn ngữ được xử lý qua những dịch vụ độc lập theo mô hình microservice, đảm bảo tính phân tách và dễ nâng cấp:

- YouTube Processor: Tải audio và lấy metadata từ video YouTube
- Language Detector: Kiểm tra ngôn ngữ chính của audio
- ASR Service: Chuyển âm thanh thành văn bản (sử dụng mô hình như Whisper)
- Sentence Splitter: Chia transcript thành từng câu và định thời gian
- CEFR Estimator: Ước lượng trình độ tiếng Anh theo khung CEFR
- Pronunciation Evaluator: So sánh âm thanh người học với bản gốc để tính điểm và đưa ra góp ý

Các dịch vụ này có thể triển khai bằng Python và giao tiếp với backend thông qua các API nội bộ (REST hoặc gRPC).

3.1.4. Tích hợp bên ngoài (Third-party APIs)

Hệ thống sử dụng các API bên thứ ba nhằm hỗ trợ cho quá trình xử lý:

- Azure translator: Dịch từng câu transcript sang tiếng Việt
- Free Dictionary: Lấy phiên âm IPA và hướng dẫn đọc chuẩn

3.1.5. Hàng đợi xử lý và tác vụ nền

Các thao tác tốn thời gian như ASR, phân tích CEFR, xử lý giọng nói được xử lý bất đồng bộ bằng hàng đợi công việc, đảm bảo trải nghiệm người dùng luôn nhanh chóng.

3.1.6. Lưu trữ dữ liệu và tệp âm thanh

PostgreSQL: Quản lý người dùng, video, transcript, ...

3.1.7. Riêng tư và bảo mật

- Ứng dụng không yêu cầu đăng ký để sử dụng các chức năng cơ bản, hạn chế thu thập dữ liệu cá nhân
- Dữ liệu người dùng và âm thanh được mã hóa trong quá trình truyền tải và có thể tự động xóa sau khi xử lý nếu người dùng không lưu lại

3.2. Môi trường phát triển

Hệ thống được phát triển chủ yếu trên các nền tảng macOS và Ubuntu Linux, với các công cụ hiện đại phục vụ cho lập trình ứng dụng di động, backend, và mô hình học sâu. Các công cụ chính bao gồm:

- Visual Studio Code: Môi trường phát triển đa năng, hỗ trợ lập trình frontend bằng React Native, backend Golang, và script xử lý AI.
- Git: Công cụ quản lý mã nguồn phân tán, tích hợp với GitHub để phối hợp làm việc nhóm và kiểm soát phiên bản.
- Postman: Hỗ trợ kiểm thử API giữa frontend và backend trong quá trình phát triển.
- Google Colab / Jupyter Notebook: Môi trường huấn luyện và kiểm thử các mô hình học sâu như Whisper, fastText, CEFR classification.
- pgAdmin / TablePlus: Công cụ trực quan để quản lý và kiểm tra cơ sở dữ liệu PostgreSQL.
- Docker: Dùng để container hóa các dịch vụ AI và backend, đảm bảo triển khai dễ dàng và nhất quán giữa các môi trường.
- Xcode Simulator: Kiểm thử giao diện người dùng trên nhiều thiết bị di động khác nhau.

3.3. Quá trình phát triển hệ thống

Hệ thống được phát triển theo quy trình mô-đun hóa theo chức năng, chia thành từng giai đoạn nhỏ tương ứng với các tính năng chính. Mỗi giai đoạn đều được kiểm thử thủ công và tự động để đảm bảo độ ổn định và khả năng mở rộng về sau.

Quá trình phát triển gồm các bước chính:

- Phân tích yêu cầu người dùng: Tập trung vào nhu cầu học tiếng Anh thực tế, đặc biệt là luyện nghe - nói phản xạ (shadowing), phù hợp với người học phổ thông đến nâng cao.
- Thiết kế kiến trúc hệ thống: Lựa chọn mô hình client-server, kết hợp microservices xử lý AI để tăng tính mở rộng và hiệu suất.
- Xây dựng từng mô-đun chức năng:
 - Module nhập video YouTube và xử lý audio
 - Tích hợp ASR (Whisper) để tạo transcript
 - Phân tách câu và đồng bộ phụ đề

- Đánh giá CEFR và phát âm người dùng
- Xây dựng UI đơn giản, tối ưu trên mobile
- Huấn luyện và kiểm thử mô hình AI:
 - Sử dụng Google Colab để thu thập dữ liệu, huấn luyện mô hình phân loại CEFR từ transcript
 - Kiểm thử kết quả ASR và đánh giá phát âm qua nhiều mẫu giọng nói khác nhau
- Kiểm thử và cải tiến:
 - Kiểm thử giao diện và API bằng tay và thông qua script tự động
 - Sử dụng bộ test đơn vị (unit test) và tích hợp (integration test) cho backend
 - Triển khai phiên bản beta nội bộ để lấy phản hồi từ người dùng thực tế
- Tối ưu hiệu năng và trải nghiệm người dùng
 - Rút gọn quy trình xử lý backend
 - Tối ưu kích thước file audio và thời gian phản hồi đánh giá phát âm

3.4. Phát triển Ứng dụng di động (React Native và Expo)

Trong dự án Shadowing, Expo được sử dụng để tăng tốc quá trình xây dựng ứng dụng React Native. pnpm được lựa chọn để quản lý gói nhờ tốc độ cài đặt nhanh và khả năng đảm bảo tính nhất quán phiên bản. Do trọng tâm chính là trải nghiệm người dùng iOS, việc phát triển và kiểm thử chủ yếu diễn ra trên các thiết bị hoặc simulator iOS.

3.4.1. Cài đặt môi trường

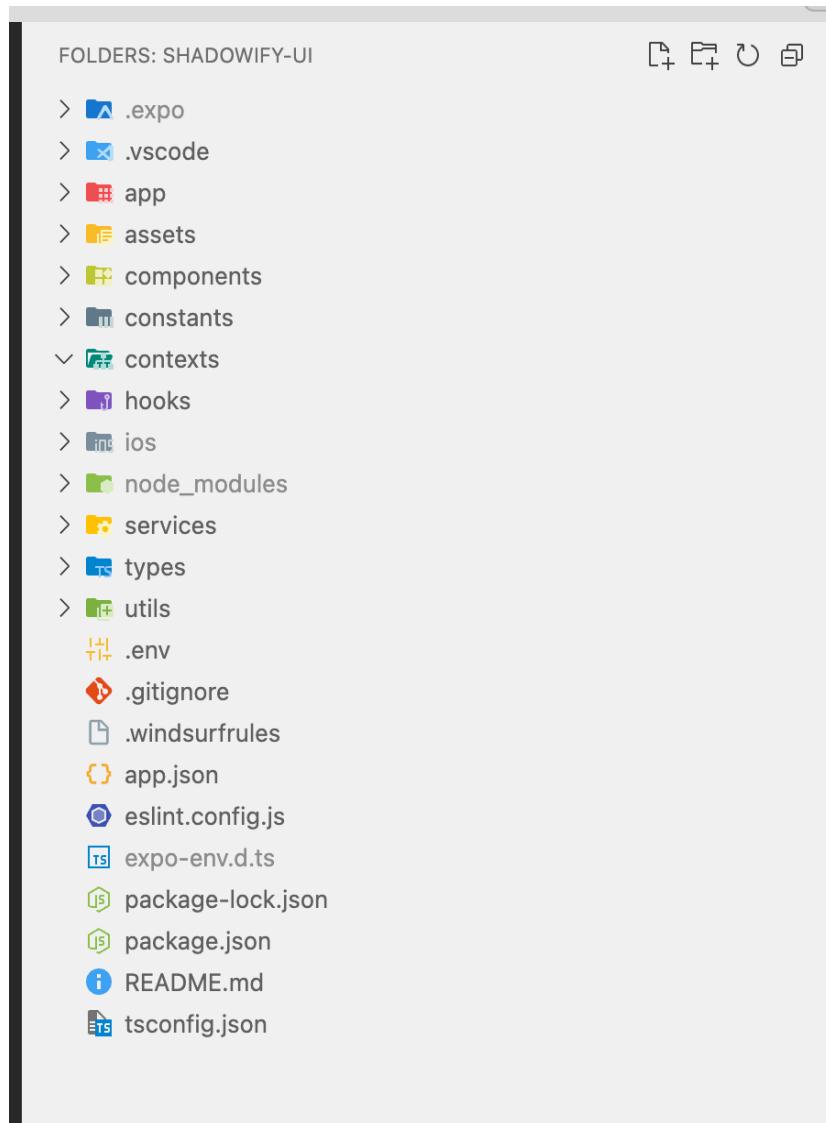
- Cài Node.js ($\geq 16.x$) và pnpm: npm install -g pnpm
- Cài Expo CLI qua pnpm: pnpm add -g expo-cli
- Cài Xcode trên macOS và đảm bảo đã cài đặt Xcode Command Line Tools (dùng để chạy simulator và build app).

3.4.2. Tạo project mới

- Chạy lệnh khởi tạo với template blank managed workflow:
expo init shadowing-mobile --template blank
- Chọn ngôn ngữ “JavaScript” hoặc “TypeScript” tuỳ sở thích nhóm.

- Trong file package.json, đổi "packageManager" thành "pnpm@<version>" để mọi thành viên dùng pnpm.

3.4.3. Cấu trúc thư mục cơ bản



Hình 3. 1. Cấu trúc thư mục Ứng dụng di động

3.4.4. Cấu hình iOS

- Trong app.json đặt ios.bundleIdentifier và ios.buildNumber phù hợp.
- Sử dụng Expo Go trên iOS Simulator hoặc thiết bị thật: expo start --ios
- Mã QR và link deep linking của Expo Go cho phép test nhanh trên iPhone.

Cài đặt dependencies cần thiết

- Dùng pnpm để thêm các thư viện: pnpm install
- Chạy và debug

- Mở Xcode Simulator (iOS 14+) và expo start để live-reload code.
- Sử dụng React Native Debugger hoặc Flipper để inspect network, redux, và console logs.

3.4.5. Giao diện Ứng dụng di động

3.4.5.1. Màn hình Splash



Hình 3. 2. Màn hình Splash

Khi người dùng mở ứng dụng lần đầu trên thiết bị, hệ thống sẽ tự động khởi tạo một mã định danh duy nhất (Device ID) và lưu cục bộ trên máy. Mã này được dùng trong tất cả các tương tác tiếp theo để định danh người dùng ẩn danh, phục vụ theo dõi tiến trình học và đồng bộ dữ liệu.

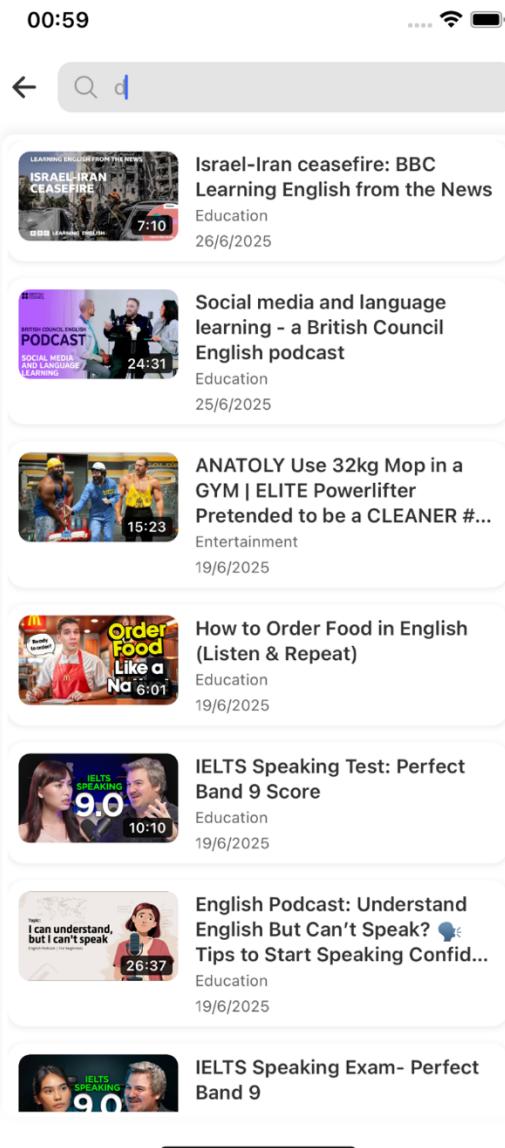
- Tác nhân: Người dùng
- Dữ liệu đầu vào: Không có (ứng dụng tự xử lý khi khởi động lần đầu)
- Hệ thống xử lý:
 - Kiểm tra xem thiết bị đã có Device ID được lưu cục bộ chưa.
 - Nếu chưa có, sinh một UUID (Universally Unique Identifier).
 - Lưu Device ID vào bộ nhớ cục bộ của thiết bị (local storage / secure storage).
- Dữ liệu đầu ra: Device ID được lưu trên thiết bị để dùng cho các lần gửi dữ liệu sau.

3.4.5.2. Màn hình tìm kiếm

Giao diện tìm kiếm toàn văn bản cho phép người dùng nhập từ khóa để tìm kiếm các video liên quan đến chủ đề mong muốn. Hệ thống sẽ quét toàn bộ nội dung bao gồm tiêu đề, mô tả và tiêu đề đầy đủ để trả về kết quả phù hợp, giúp người học dễ dàng khám phá nội dung phù hợp với mục tiêu học tập.

- Tác nhân: Người dùng
- Dữ liệu đầu vào: Từ khóa (text query) do người dùng nhập vào thanh tìm kiếm
- Hệ thống xử lý:
 - Khi người dùng nhập từ khóa và nhấn tìm kiếm:
 - Hệ thống xử lý từ khóa (lọc bỏ khoảng trắng dư thừa, ký tự đặc biệt).
 - Gửi truy vấn toàn văn (full-text search) lên hệ thống backend.
 - Tìm kiếm trong các trường:
 - Tiêu đề video (title)
 - Tiêu đề đầy đủ (full_title)
 - Mô tả video (description)
 - Nhận danh sách kết quả phù hợp theo mức độ liên quan.
 - Tô đậm (highlight) từ khóa trong kết quả để dễ nhận biết.
 - Dữ liệu đầu ra:

- Danh sách video khớp với từ khóa tìm kiếm.
- Hiển thị tiêu đề, mô tả rút gọn, thumbnail video.
- Từ khóa được highlight trong phần kết quả.
- Nếu không có kết quả, hiển thị thông báo "No videos found".

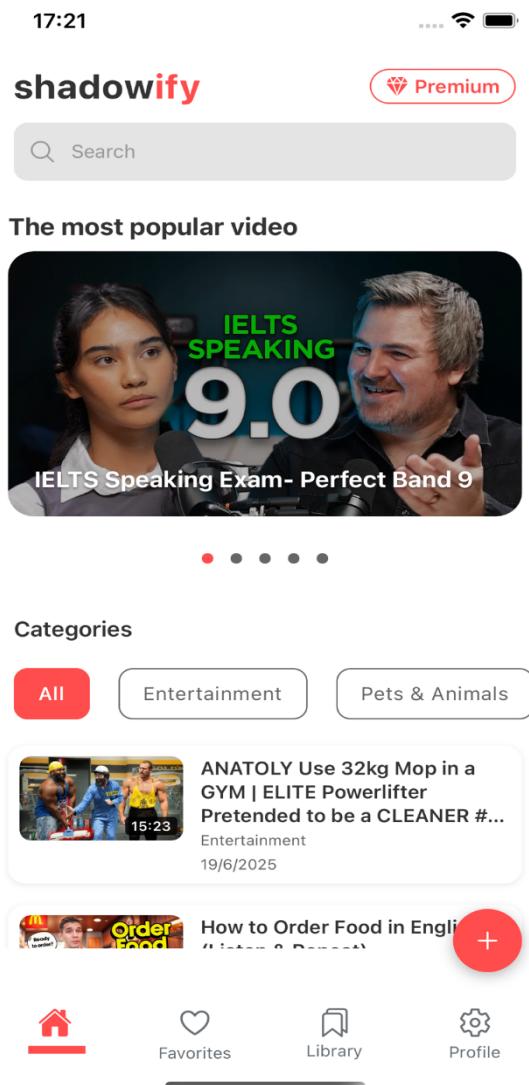


Hình 3. 3. Màn hình tìm kiếm

3.4.5.3. Màn hình chính

Sau khi người dùng mở ứng dụng lần đầu và bấm vào nút “Get Started”, hệ thống sẽ chuyển hướng đến giao diện chính. Màn hình chính cung cấp lối vào nhanh đến các nội dung học phổ biến, được cá nhân hóa theo người dùng, giúp người học bắt đầu ngay mà không cần cấu hình phức tạp.

- Tác nhân: Người dùng
- Dữ liệu đầu vào: Không có (người dùng chỉ bấm “Get Started”)
- Hệ thống xử lý:
 - Ghi nhận sự kiện “Get Started” để chuyển hướng người dùng đến màn hình chính.
 - Truy vấn và hiển thị 5 video phổ biến nhất dựa trên lượt xem tổng hợp từ người dùng khác (global popularity).
 - Tiếp theo là phần hiển thị danh sách chuyên mục (categories) như: Giao tiếp, Kinh doanh, Du lịch, Học thuật, v.v.
 - Bên dưới, hiển thị danh sách tất cả video theo phân trang dạng cuộn vô hạn (infinity scroll).
- Dữ liệu đầu ra:
 - Giao diện hiển thị 5 video phổ biến.
 - Danh sách chuyên mục học tập.
 - Danh sách video được phân trang và tải dần khi người dùng cuộn xuống.



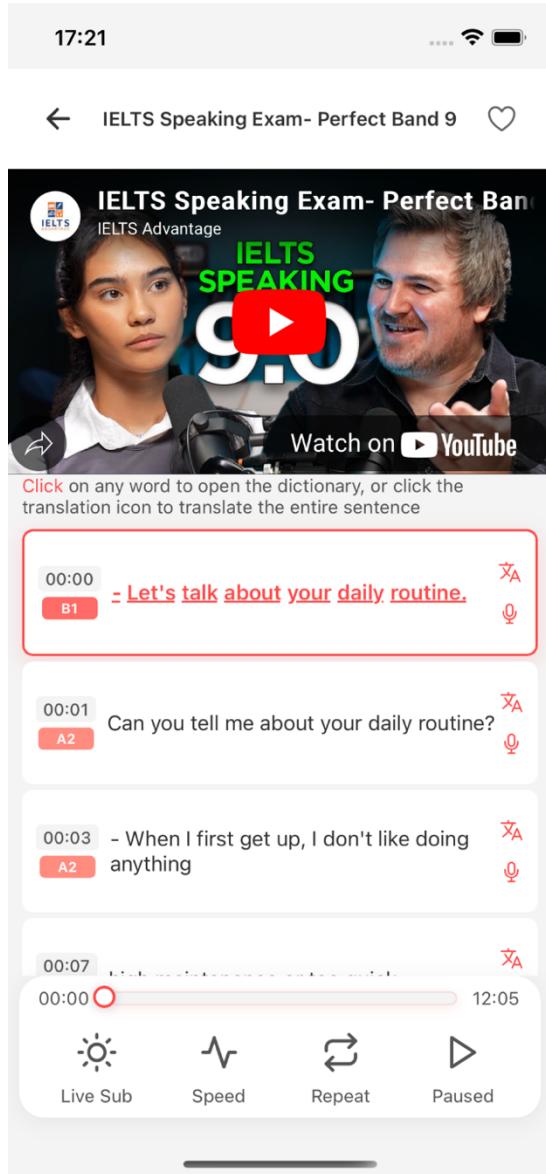
Hình 3. 4. Màn hình chính

3.4.5.4. Màn hình video chi tiết

Khi người dùng bấm vào một video bất kỳ trong danh sách, hệ thống sẽ chuyển đến màn hình chi tiết video, nơi cung cấp trải nghiệm học tập trực quan thông qua việc kết hợp xem video và theo dõi phụ đề tương tác theo thời gian thực.

- Tác nhân: Người dùng
- Dữ liệu đầu vào: Video ID (YouTube ID)
- Hệ thống xử lý:
 - Tải và nhúng iframe video từ YouTube dựa trên YouTube ID của video được chọn.

- Truy vấn và hiển thị danh sách các segment transcript được chia theo từng đoạn thời gian của video.
- Tích hợp bộ điều khiển LiveSub với các chức năng sau:
 - Hiển thị phụ đề chạy theo thời gian (highlight theo dòng đang phát).
 - Tùy chỉnh tốc độ phát video (0.5x, 1x, 1.5x, v.v.).
 - Chế độ repeat cho phép lặp lại câu hiện tại hoặc đoạn transcript đang phát.
 - Đồng bộ hóa giữa transcript và tiến độ video theo thời gian thực.
- Dữ liệu đầu ra:
 - Iframe video YouTube hoạt động.
 - Danh sách transcript chia đoạn hiển thị bên dưới hoặc bên cạnh.
 - Giao diện điều khiển LiveSub hoạt động đồng bộ với video và transcript.



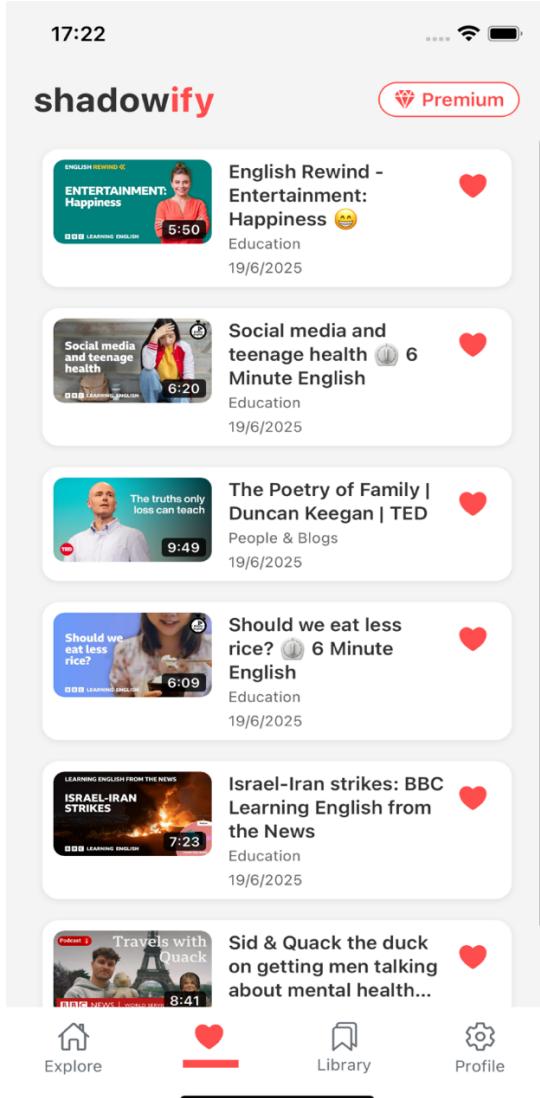
Hình 3. 5. Màn hình video chi tiết

3.4.5.5. Màn hình yêu thích

Màn hình Import video cho phép người dùng nhập một URL hoặc ID YouTube để hệ thống tự động lấy thông tin và phân tích video, phục vụ cho quá trình học tập. Giao diện đơn giản, trực quan, chỉ yêu cầu người dùng cung cấp một trường duy nhất.

- Tác nhân: Người dùng
- Dữ liệu đầu vào: YouTube URL hoặc YouTube ID
- Hệ thống xử lý:
 - Kiểm tra định dạng hợp lệ của URL hoặc ID.
 - Nếu hợp lệ, hệ thống tiến hành:

- Crawl metadata video (tiêu đề, mô tả, thời lượng...).
- Tải âm thanh từ video.
- Chuyển giọng nói thành văn bản (ASR).
- Tiền xử lý transcript: tách câu, làm sạch dữ liệu.
- Phân tích CEFR dựa trên nội dung transcript.
- Lưu toàn bộ kết quả vào hệ thống.
 - Nếu URL/ID không hợp lệ, hiển thị thông báo lỗi.
- Dữ liệu đầu ra:
 - Thông báo thành công khi import.
 - Video được đưa vào danh sách quản lý hoặc xử lý tiếp.
 - Hoặc hiển thị lỗi nếu URL không hợp lệ video không phải tiếng anh.



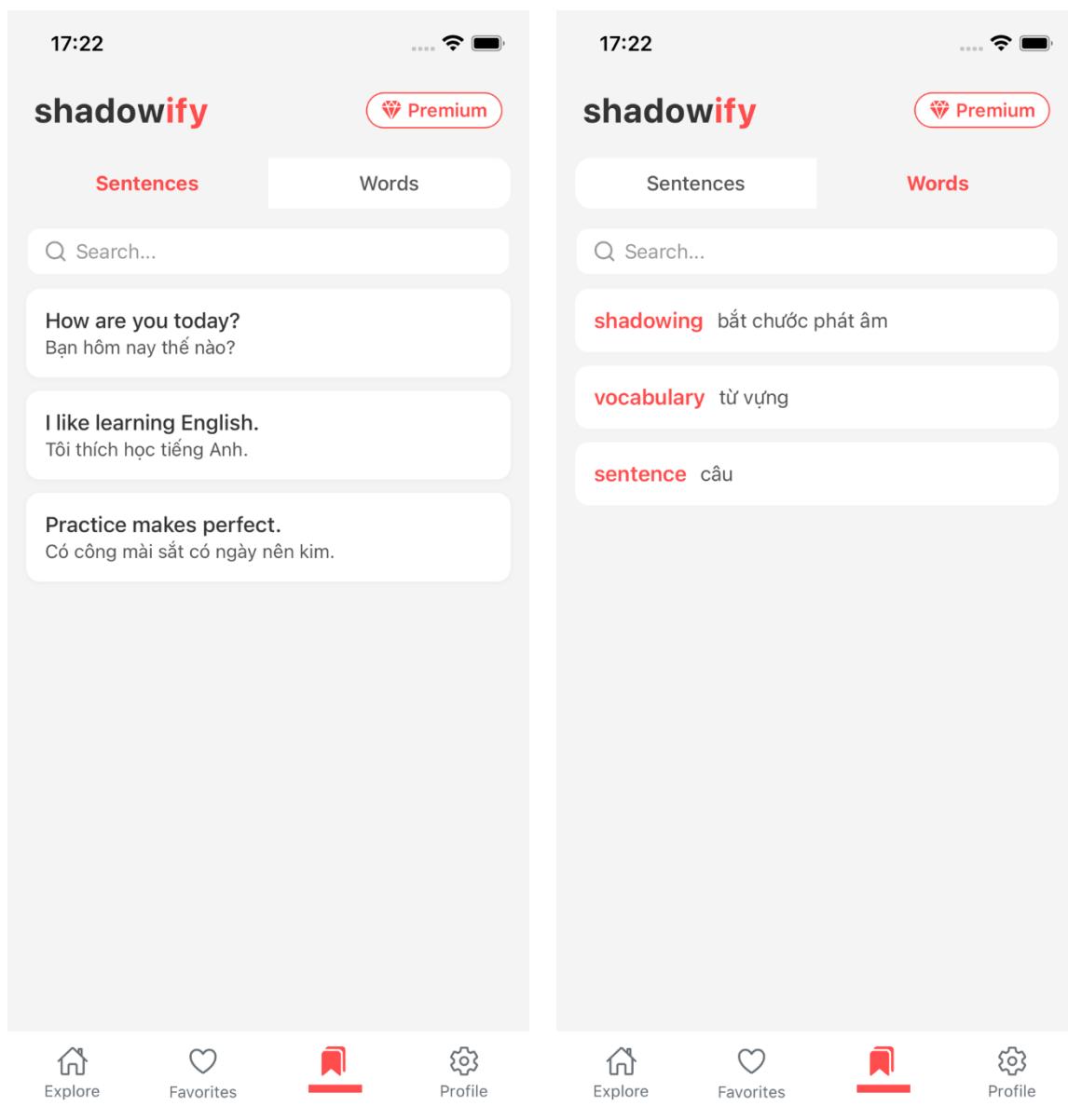
Hình 3. 6. Màn hình yêu thích

3.4.5.6. Màn hình saved

Màn hình yêu thích cho phép người dùng xem lại các video mà họ đã đánh dấu là yêu thích trong quá trình sử dụng ứng dụng. Danh sách này giúp người học nhanh chóng truy cập lại các nội dung quan trọng hoặc phù hợp với sở thích cá nhân.

- Tác nhân: Người dùng
- Dữ liệu đầu vào: Không có đầu vào trực tiếp (màn hình tự động truy xuất danh sách yêu thích từ hệ thống)
- Hệ thống xử lý:
 - Truy xuất danh sách video mà người dùng đã gán trạng thái “yêu thích” từ trước.
 - Hiển thị danh sách video yêu thích theo dạng lưới hoặc danh sách.

- Mỗi video có hiển thị icon “❤” (đã đánh dấu yêu thích).
- Cho phép người dùng bỏ yêu thích bằng cách nhấn lại vào icon trái tim.
- Dữ liệu đầu ra:
 - Danh sách video yêu thích của người dùng
 - Cập nhật trạng thái khi người dùng thêm hoặc bỏ yêu thích

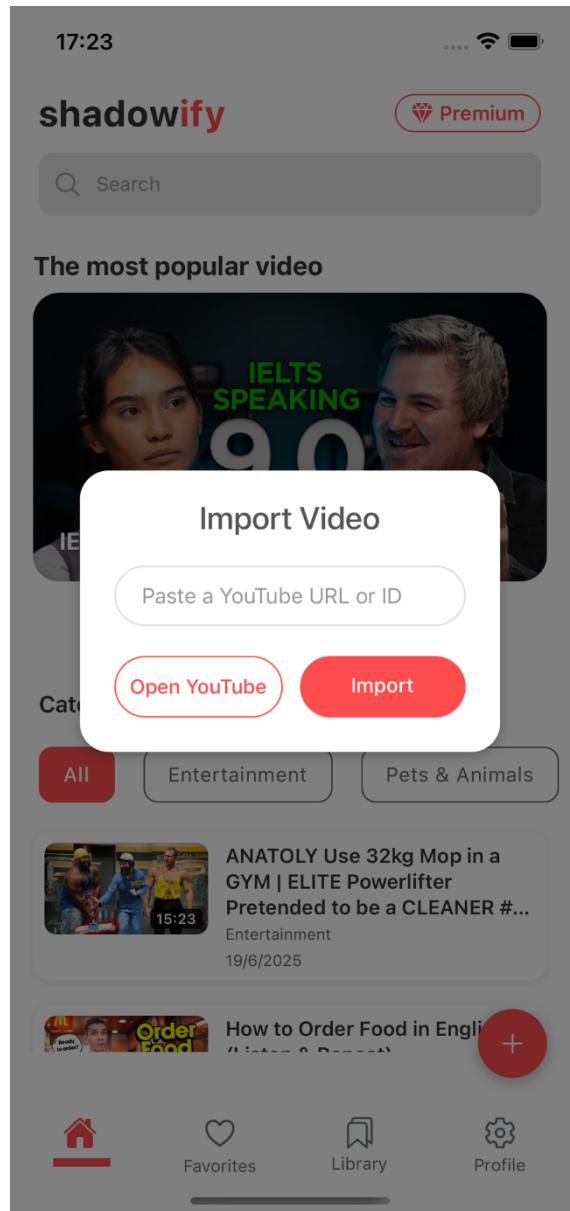


Hình 3. 7. Màn hình saved

3.4.5.7. Màn hình Import video

Màn hình Import video cho phép người dùng nhập một URL hoặc ID YouTube để hệ thống tự động lấy thông tin và phân tích video, phục vụ cho quá trình học tập. Giao diện đơn giản, trực quan, chỉ yêu cầu người dùng cung cấp một trường duy nhất.

- Tác nhân: Người dùng
- Dữ liệu đầu vào: YouTube URL hoặc YouTube ID
- Hệ thống xử lý:
 - Kiểm tra định dạng hợp lệ của URL hoặc ID.
 - Nếu hợp lệ, hệ thống tiến hành:
 - Crawl metadata video (tiêu đề, mô tả, thời lượng...).
 - Tải âm thanh từ video.
 - Chuyển giọng nói thành văn bản (ASR).
 - Tiên xử lý transcript: tách câu, làm sạch dữ liệu.
 - Phân tích CEFR dựa trên nội dung transcript.
 - Lưu toàn bộ kết quả vào hệ thống.
 - Nếu URL/ID không hợp lệ, hiển thị thông báo lỗi.
- Dữ liệu đầu ra:
 - Thông báo thành công khi import.
 - Video được đưa vào danh sách quản lý hoặc xử lý tiếp.
 - Hoặc hiển thị lỗi nếu URL không hợp lệ video không phải tiếng anh



Hình 3. 8. Màn hình Import video

3.4.5.8. Màn hình luyện tập Shadowing

Màn hình luyện tập Shadowing là nơi người dùng có thể rèn luyện kỹ năng nói bằng cách nhại lại từng câu trong video. Ứng dụng phát lại từng đoạn video (segment) tương ứng với transcript và mức độ CEFR, cho phép người học luyện nói theo đúng nhịp và ngữ điệu.

- Tác nhân: Người dùng
- Dữ liệu đầu vào: Segment transcript kèm CEFR từ video đã chọn
- Hệ thống xử lý:

- Phát lại đoạn video tương ứng với segment được chọn (có thể tự động lặp lại – repeat).
- Hiển thị transcript câu gốc cùng mức CEFR (ví dụ: A2, B1...).
- Cho phép người dùng nhấn nút “Ghi âm” để bắt đầu luyện tập.
- Hệ thống ghi lại giọng nói của người dùng và xử lý như sau:
 - Chuyển giọng nói thành văn bản (ASR – automatic speech recognition).
 - So sánh đoạn text của người dùng với bản gốc (gợi ý sai lệch, thiếu từ, phát âm sai...).
 - Tính điểm phát âm tổng thể (có thể theo phần trăm hoặc thang điểm ABC).
- Giao diện cung cấp tùy chọn:
 - Thử lại (Retry) để ghi âm lại nếu chưa đạt.
 - Nghe lại giọng mình để tự đánh giá phát âm.
- Dữ liệu đầu ra:
 - Transcript câu gốc + transcript người dùng
 - Highlight các lỗi sai/thiếu từ
 - Điểm số phát âm
 - Giao diện cho phép nghe lại và luyện tập lại nhiều lần



Shadowing Practice



Original Transcript

A2



From BBC Learning English, this is Learning English from the News.

⌚ Free for Today: 5 mins remaining



Record

Hình 3. 9. Màn hình luyện tập Shadowing

3.4.5.9. So sánh & chấm điểm tại Frontend

Cho phép người dùng ghi âm giọng đọc của mình, gửi lên backend để chuyển thành văn bản bằng mô hình Whisper. Hệ thống so sánh văn bản thu được với câu chuẩn, tính điểm độ chính xác và hiển thị kết quả gồm: điểm số, câu chuẩn, câu người đọc, và highlight các từ sai. Người dùng có thể nghe lại hoặc ghi âm lại nếu muốn.

- Tác nhân: Người dùng
- Dữ liệu đầu vào: Audio ghi âm từ người dùng (đoạn câu đọc)
- Hệ thống xử lý:
 - Gửi audio đến backend.

- Backend sử dụng Whisper để nhận dạng giọng nói và chuyển đổi thành text.
- Backend so sánh văn bản chuyển đổi với câu chuẩn của segment hiện tại:
 - Xác định các từ khớp, các từ sai hoặc thiếu.
 - Tính điểm số độ chính xác (%).
- Backend trả về dữ liệu so sánh (bao gồm text nhận dạng, danh sách từ sai, điểm số).
- Frontend hiển thị màn hình so sánh kết quả:
 - Văn bản gốc (câu chuẩn).
 - Văn bản người dùng đọc.
 - Điểm số tổng thể.
 - Các từ sai được làm nổi bật (highlight).
 - Tùy chọn nghe lại audio đã ghi.
 - Tùy chọn ghi âm lại.
- Dữ liệu đầu ra:
 - Giao diện modal hoặc màn hình hiển thị:
 - Câu chuẩn.
 - Văn bản người dùng đọc.
 - Điểm số đánh giá.
 - Danh sách từ sai hoặc khác biệt.
 - Nút nghe lại, ghi lại.
 - Nút đóng.

← Shadowing Practice


Original Transcript

A2



hello and welcome to 6 minute english from
bbc learning english i'm beth

Your Speech

hello and welcome to sick minutes in this
from bbc learning in it i'm pat

⌚ Free for Today: 5 mins remaining

Record Again

62%
Accuracy

My Record

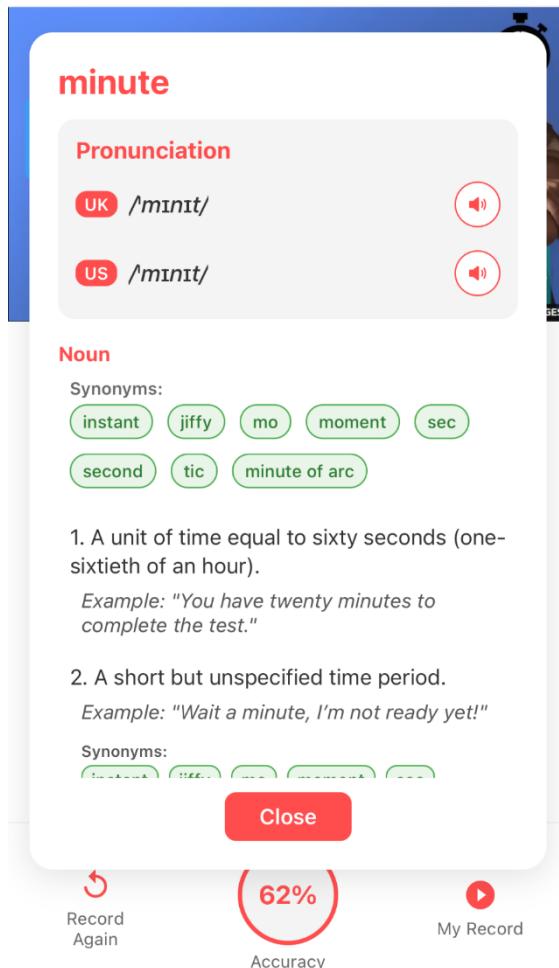
Hình 3. 10. Màn hình So sánh & chấm điểm tại Frontend

3.4.5.10. Xem từ vựng

Giao diện xem định nghĩa từ vựng được kích hoạt khi người dùng nhấp vào một từ bất kỳ trong transcript, trong danh sách từ vựng bookmark, hoặc khi tra từ trực tiếp. Giao diện này hiển thị dưới dạng popup/modal giúp người học nắm bắt nhanh nghĩa, phát âm và cách sử dụng của từ vựng đó.

- Tác nhân: Người dùng
- Dữ liệu đầu vào: Từ tiếng Anh được chọn
- Hệ thống xử lý:

- Gửi truy vấn đến FreeDictionary API hoặc hệ thống từ điển nội bộ để lấy dữ liệu của từ.
- Khi nhận được dữ liệu, hệ thống hiển thị:
 - Từ vựng gốc (English word).
 - Phiên âm IPA (ví dụ: /'præktɪs/).
 - Nút phát âm với giọng Anh và Mỹ.
 - Định nghĩa tiếng Anh chia theo từ loại (noun, verb, adjective...).
 - Nghĩa tiếng Việt (nếu có dữ liệu hỗ trợ).
 - Ví dụ sử dụng trong câu.
- Giao diện có thể hỗ trợ swipe qua lại nếu người dùng tra nhiều từ liên tiếp.
- Có nút “Close”
- Dữ liệu đầu ra:
 - Giao diện modal hiển thị thông tin chi tiết của từ vựng.
 - Tùy chọn phát âm, xem nghĩa, ví dụ, từ loại.

← Shadowing Practice


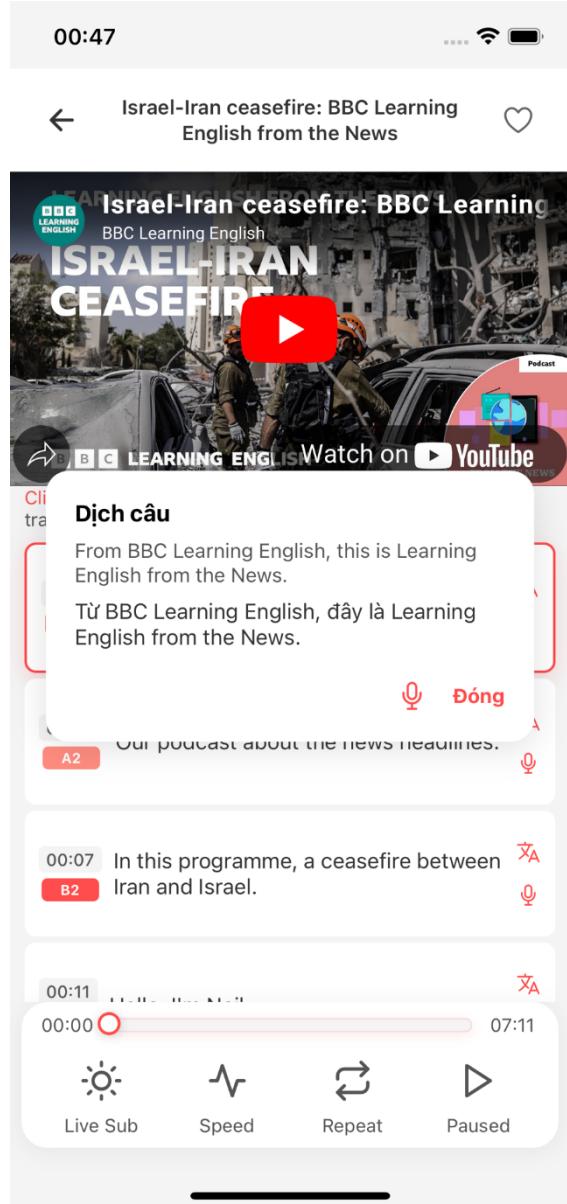
Hình 3. 11. Màn hình xem từ vựng

3.4.5.11. Màn hình dịch câu

Trong khi xem video chi tiết, người dùng có thể nhấp vào bất kỳ đoạn transcript nào để xem bản dịch song ngữ tương ứng. Tính năng này giúp người học hiểu sâu từng câu, từ đó cải thiện khả năng nghe – hiểu và dịch ngược.

- Tác nhân: Người dùng
- Dữ liệu đầu vào: Câu transcript được chọn (segment cụ thể của video)
- Hệ thống xử lý:
 - Nhận diện đoạn transcript mà người dùng đã chọn.

- Truy xuất bản dịch tiếng Việt tương ứng đã được xử lý từ trước hoặc gửi yêu cầu dịch nếu chưa có.
- Hiển thị giao diện dịch bao gồm:
 - Câu gốc tiếng Anh.
 - Bản dịch tiếng Việt.
 - Giao diện song ngữ (song song).
 - Tùy chọn ẩn/hiện bản dịch.
- Dữ liệu đầu ra:
 - Giao diện hiển thị câu tiếng Anh và bản dịch tiếng Việt.
 - Câu được đồng bộ highlight nếu video đang phát.



Hình 3. 12. Màn hình dịch câu

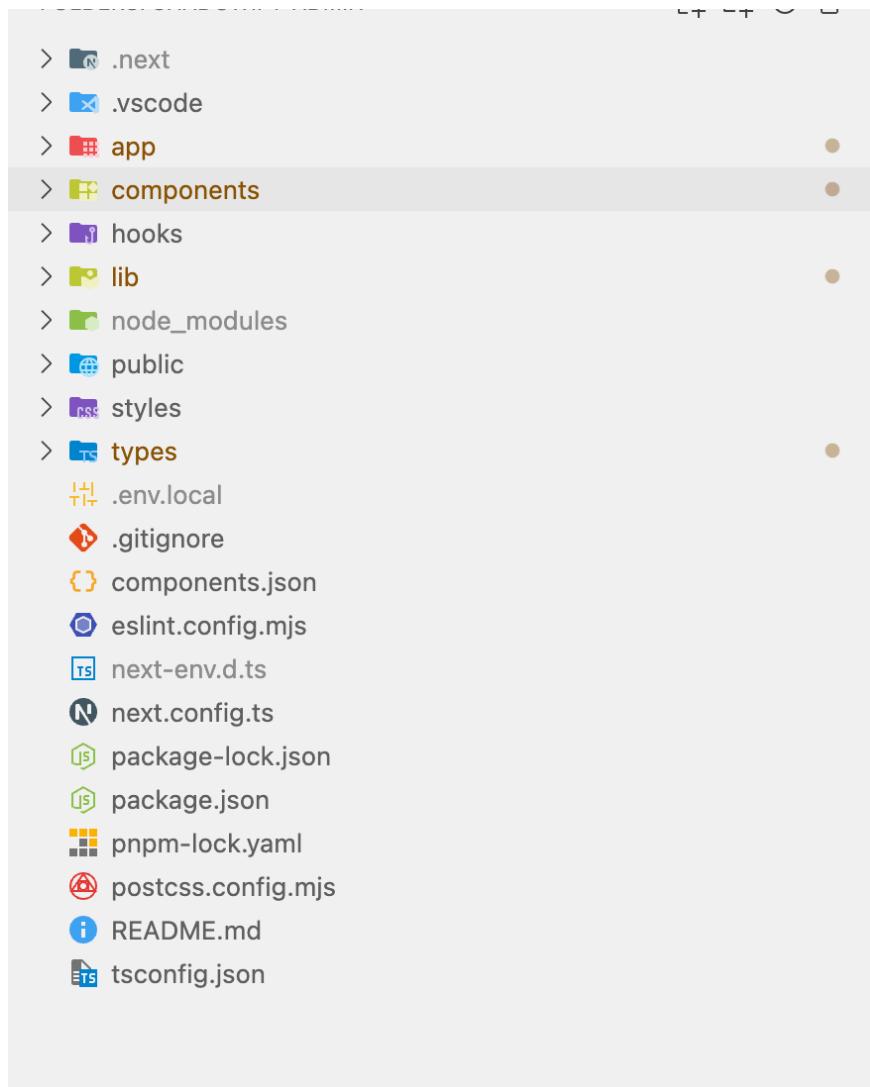
3.5. Phát triển Web Interface

3.5.1. Khởi tạo dự án nextjs + shadcn

Để xây dựng giao diện web nhập link YouTube và danh sách video, dự án sử dụng Next.js – framework React hỗ trợ server-side rendering và static generation – kết hợp thư viện shadcn/ui để nhanh chóng có bộ component UI hiện đại, linh hoạt.

```
pnpm dlx shadcn@latest init
```

3.5.2. Cấu trúc dự án



Hình 3. 13. Cấu trúc thư mục Website

3.5.3. Màn hình giao diện đóng góp video

Giao diện web đóng góp video là một cổng công khai cho phép bất kỳ người dùng nào (kể cả không đăng nhập) gửi video từ YouTube vào hệ thống. Mục tiêu là mở rộng kho dữ liệu học tập thông qua cộng đồng, đồng thời cho phép người đóng góp xem trước bản phân tích nội dung trước khi xác nhận gửi.

- Tác nhân: Người dùng (không cần đăng nhập)
- Dữ liệu đầu vào: URL hoặc ID của video YouTube
- Hệ thống xử lý:
 - Khi người dùng nhập URL hoặc ID YouTube và nhấn "Phân tích":
 - Kiểm tra định dạng video hợp lệ.

- Thực hiện:
 - Crawl metadata video (tiêu đề, mô tả, thời lượng).
 - Tải âm thanh, thực hiện chuyển giọng nói thành văn bản (ASR).
 - Phân tích CEFR level cho từng câu transcript.
- Hiển thị kết quả phân tích dưới dạng:
 - Tiêu đề, mô tả, thumbnail video
 - Transcript chia theo từng đoạn/câu
 - CEFR Level ứng với mỗi câu
- Người dùng có thể xem trước toàn bộ thông tin và nhấn nút “Đóng góp” để gửi vào hệ thống.
- Dữ liệu đầu ra:
 - Preview kết quả phân tích: video, transcript, CEFR
 - Thông báo xác nhận sau khi đóng góp thành công

3.5.3.1. Màn hình danh sách videos

Thumbnail	Title	Duration	Category	Updated At
	ANATOLY Use 32kg Mop in a GYM ELITE Powerlifter Pretended to be a CLEANER #45	15:23	Entertainment	6/19/2025
	How to Order Food in English (Listen & Repeat)	6:01	Education	6/19/2025
	IELTS Speaking Test: Perfect Band 9 Score	10:10	Education	6/19/2025
	English Podcast: Understand English But Can't Speak? Tips to Start Speaking Confidently!	26:37	Education	6/19/2025
	IELTS Speaking Exam- Perfect Band 9	12:04	Education	6/19/2025
	Emma Watson Once Mistook Jimmy Fallon for Jimmy Kimmel	4:36	Comedy	6/19/2025
	Anne Hathaway Forgets The Princess Diaries and The Devil Wears Prada Details, Chats	10:23	Comedy	6/19/2025

Hình 3. 14. Màn hình danh sách videos

3.5.3.2. Màn hình thêm mới video

The screenshot shows the Shadowify Admin interface. On the left, there's a sidebar with a user icon, 'Shadowify Enterprise', and navigation links for 'Platform' and 'Videos'. The main area is titled 'Shadowify Admin' and shows a 'Videos' dashboard with a search bar and a button to 'Add New'. Below this is a table listing several videos, each with a thumbnail, title, duration, category, and update date. A modal window titled 'Add New Video' is open in the center, containing a text input field with placeholder text 'Enter YouTube video ID or URL' and a 'Submit' button.

Hình 3. 15. Màn hình thêm mới video

3.5.3.3. Màn hình video chi tiết

This screenshot shows a detailed view of a video titled 'IELTS Speaking Exam- Perfect Band 9'. The top navigation bar includes 'Dashboard', 'Videos', and a back link 'Back to Videos'. The main content area has tabs for 'Details' and 'Transcript', with 'Details' selected. It displays the video title, ID, and a large thumbnail image featuring two people speaking. To the right is a 'Video Details' panel with fields for Duration (12:04), Created (6/19/2025), Updated (6/19/2025), and YouTube ID (E4iUiRBVUa4). Below this is a 'Preview' panel showing a smaller video player with the same content and a 'Watch on YouTube' link.

Hình 3. 16. Màn hình video chi tiết

3.5.3.4. Màn hình transcript

The screenshot shows the Shadowify Admin interface. On the left, there's a sidebar with the 'Shadowify Enterprise' logo, 'Platform', and a 'Videos' section. The main area has a header 'Shadowify Admin' with a back arrow and 'Dashboard > Videos > IELTS Speaking Exam-...'. Below the header, there are two tabs: 'Details' and 'Transcript', with 'Transcript' being active. The transcript section displays 235 segments found, each with a timestamp, duration, and text. The segments are color-coded: B1 (yellow), A2 (light blue), A2 (light blue), A2 (light blue), A1 (green), and A2 (light blue). To the right of the transcript is a 'Video Details' box showing Duration (12:04), Created (6/19/2025), Updated (6/19/2025), and YouTube ID (E4iUiRBVUa4). At the bottom right is a 'Preview' box showing a thumbnail of the video with the text 'IELTS SPEAKING' and a 'Watch on YouTube' button.

Hình 3. 17. Màn hình transcript

KẾT QUẢ VÀ KIẾN NGHỊ

1. Kết quả đạt được

- Về mặt kỹ thuật, hệ thống đã hoàn thiện quy trình xử lý video từ YouTube một cách tự động, tích hợp công cụ yt-dlp trong Golang để tải metadata và trích xuất audio với thời gian xử lý trung bình dưới 15 giây cho mỗi video 1 phút. Phần audio được chuyển đổi thành transcript bằng cách gọi Whisper CLI từ Go, cho độ chính xác trên 90% với tốc độ ~45 giây cho mỗi phút âm thanh.
- Hệ thống đánh giá trình độ CEFR được xây dựng bằng Python và FastAPI, hỗ trợ xử lý hàng trăm câu chỉ trong dưới một giây, đáp ứng yêu cầu latency thấp. Mô hình học máy đạt hiệu quả đáng kể: trong tập test 1.460 câu, sai số trung bình (MAE) là 0.27 trên thang 1–6, độ chính xác khoảng 84%, và F1-score đạt 83.8%. Mô hình phân biệt tốt các cấp độ từ A2 đến B2, dù vẫn còn hạn chế ở A1 và C2.
- Về khả năng triển khai, hệ thống đã được container hóa với Docker, cho phép mở rộng linh hoạt các dịch vụ như VideoService và CEFRService. Việc kết nối cơ sở dữ liệu PostgreSQL qua GORM cũng đảm bảo tính ổn định, hỗ trợ migration và backup hiệu quả.
- Ở mặt chức năng, người học có thể luyện tập kỹ năng nghe – nói với phản hồi gần như thời gian thực (dưới 200ms), bao gồm cả đánh giá phát âm, highlight từ sai và điểm tổng quan. Các tính năng như tra từ điển, dịch câu, lưu từ/câu yêu thích và tìm kiếm toàn văn đều hoạt động ổn định, giao diện thân thiện và truy vấn nhanh chóng (thời gian trung bình dưới 50ms cho các truy vấn phức tạp).

2. Hạn chế hiện tại

- Tuy hệ thống đã vận hành ổn định ở quy mô nhỏ, nhưng vẫn chưa kiểm thử tải ở mức hàng trăm đến hàng nghìn người dùng đồng thời. Điều này có thể gây nghẽn cổ chai, đặc biệt tại VideoService và Whisper khi xử lý đồng loạt.
- Nguồn dữ liệu huấn luyện CEFR hiện còn hạn chế (khoảng 3.000–5.000 câu), phần lớn tập trung ở mức độ A1–B1, dẫn đến recall thấp cho A1 và C2 (~39% và ~43%). Hệ thống cũng gặp khó khăn trong việc xử lý những câu dài, phức hợp hoặc mang tính chuyên ngành do mô hình embedding chưa đủ khả năng hiểu ngữ cảnh sâu.

- Tính năng quản trị còn thủ công, chưa có dashboard trực quan; các báo cáo hiện vẫn thực hiện qua truy vấn SQL hoặc xuất file. Giao diện người dùng mới chỉ hỗ trợ song ngữ Việt–Anh và thiếu các thành phần tương tác như biểu đồ, infographic, hay tùy chọn đa ngôn ngữ khác để phục vụ thị trường rộng hơn.

3. Kiến nghị phát triển

- Để nâng cao hiệu năng, cần thực hiện kiểm thử chịu tải từ 100 đến 1.000 người dùng đồng thời, đồng thời triển khai auto-scaling cho các dịch vụ quan trọng trên nền Kubernetes. Ngoài ra, việc sử dụng bộ nhớ đệm (Redis) để lưu kết quả đã xử lý và tối ưu truy vấn cơ sở dữ liệu sẽ giúp cải thiện tốc độ phản hồi và khả năng mở rộng.
- Về AI, cần thu thập thêm dữ liệu đa dạng, đặc biệt ở mức B2–C2 và từ các nguồn chuyên ngành. Việc áp dụng các mô hình tiên tiến như mBERT hoặc XLM-RoBERTa, kết hợp học chuyển tiếp (transfer learning), và phương pháp ensemble sẽ nâng cao độ chính xác. Ngoài ra, áp dụng Reinforcement Learning từ phản hồi người dùng sẽ cải thiện thuật toán chấm điểm phát âm.
- Về trải nghiệm người học, nên phát triển dashboard cá nhân hóa với thống kê, biểu đồ trực quan, và đề xuất học tập dựa trên tiến độ CEFR và từ vựng đã lưu. Giao diện cũng cần hỗ trợ thêm các ngôn ngữ như Trung Quốc, Tây Ban Nha và khả năng tùy chỉnh theo vùng địa lý.
- Cuối cùng, hệ thống quản trị nên được tự động hóa với dashboard quản trị (Metabase, Grafana) theo dõi thời gian thực, cùng quy trình gửi báo cáo và cảnh báo tự động qua email hoặc Slack. Việc tích hợp AI kiểm duyệt nội dung cũng sẽ giảm thiểu gánh nặng thủ công trong việc điều phối cộng đồng.

TÀI LIỆU THAM KHẢO

- [1]. "Alan A. A. Donovan, Brian W. Kernighan" (2015), "The Go Programming Language", "Addison-Wesley", "Boston, MA, USA".
- [2]. "William Kennedy, Brian Ketelsen, Erik St. Martin" (2014), "Go in Action", "Manning Publications", "Shelter Island, NY, USA".
- [3]. "Eric Matthes" (2019), "Python Crash Course, 2nd Edition: A Hands-On, Project-Based Introduction to Programming", "No Starch Press", "San Francisco, CA, USA".
- [4]. "Luciano Ramalho" (2015), "Fluent Python: Clear, Concise, and Effective Programming", "O'Reilly Media", "Sebastopol, CA, USA".
- [5]. "Marijn Haverbeke" (2018), "Eloquent JavaScript, 3rd Edition: A Modern Introduction to Programming", "No Starch Press", "San Francisco, CA, USA".
- [6]. "Douglas Crockford" (2008), "JavaScript: The Good Parts", "O'Reilly Media", "Sebastopol, CA, USA".
- [7]. "Boris Cherny" (2019), "Programming TypeScript: Making Your JavaScript Applications Scale", "O'Reilly Media", "Sebastopol, CA, USA".
- [8]. "Basarat Ali Syed" (N/A), "TypeScript Deep Dive", "N/A", "N/A". "Stanley B. Lippman et.al." (2012), "C++ Primer, 5th Edition", "Addison-Wesley", "Boston, MA, USA".
- [9]. "Bjarne Stroustrup" (2013), "The C++ Programming Language, 4th Edition", "Addison-Wesley", "Boston, MA, USA".
- [10]. "Alex Banks, Eve Porcello" (2020), "Learning React: Modern Patterns for Developing React Apps", "O'Reilly Media", "Sebastopol, CA, USA".
- [11]. "Robin Wieruch" (2020), "The Road to React: Your journey to master plain React.js", "self-published", "Berlin, Germany".
- [12]. "Houssein Djirdeh et.al." (2019), "Fullstack React Native: Create beautiful mobile apps with JavaScript and React Native", "Fullstack.io", "San Francisco, CA, USA".
- [13]. "Nader Dabit" (2019), "React Native in Action", "Manning Publications", "Shelter Island, NY, USA".

- [14]. "Regina O. Obe, Leo S. Hsu" (2017), "PostgreSQL: Up and Running, 3rd Edition", "O'Reilly Media", "Sebastopol, CA, USA".
- [15]. "Hans-Jürgen Schönig" (2019), "Mastering PostgreSQL 11", "Packt Publishing", "Birmingham, UK".
- [16]. "Nigel Poulton" (2019), "Docker Deep Dive", "self-published", "UK".
- [17]. "Jeff Nickoloff, Stephen Kuenzli" (2019), "Docker in Action, 2nd Edition", "Manning Publications", "Shelter Island, NY, USA".
- [18]. "Devlin, J. et.al." (2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", "Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)", "4171-4186", doi: 10.18653/v1/N19-1423.
- [19]. "Wolf, T. et.al." (2020). "Transformers: State-of-the-Art Natural Language Processing", "Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations", "38-45", doi: 10.18653/v1/2020.emnlp-demos.6.
- [20]. "Honnibal, M., Montani, S." (2017). "spaCy: Industrial-strength Natural Language Processing in Python", "Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations", "213-218", doi: 10.18653/v1/D17-2003.
- [21]. "Bird, S., Klein, E., Loper, E." (2009). "Natural Language Processing with Python", "O'Reilly Media".
- [22]. "Radford, A. et.al." (2023). "Robust Speech Recognition via Large-Scale Weak Supervision", "arXiv preprint arXiv:2212.04356".
- [23]. "Geertzen, J., Van Hout, R." (2018). "Towards Automated CEFR-based Spanish Language Proficiency Assessment", "Proceedings of the Twelfth International Conference on Language Resources and Evaluation (LREC 2018)", "1729-1736".
- [24]. "Forsgren, E., Ljung, M." (2020). "Continuous Integration and Continuous Delivery with GitHub Actions", "Blekinge Institute of Technology".

- [25]. "Kumar, R., Gupta, A." (2018). "Microservices Architecture with Docker and Docker Compose", "International Journal of Computer Applications", "182(29)", "1-6", doi: 10.5120/ijca2018917822.
- [26]. "Gupta, A., Gupta, V." (2020). "Automating Kubernetes Deployments using Helm Charts", "International Journal of Advanced Research in Computer Science", "11(3)", "1-5".