

**ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**



**23021558 Đoàn Minh Hoàng
23020043 Trần Quang Đỉnh
23020082 Nguyễn Quốc Huy
23020061 Trần Trung Hậu**

INSIGHT TUNE PHẦN MỀM NGHE NHẠC TÍCH HỢP AI

BÀI TẬP NHÓM BỘ MÔN PHÁT TRIỂN PHẦN MỀM DI ĐỘNG

HÀ NỘI - 2025

**ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**



**23021558 Đoàn Minh Hoàng
23020043 Trần Quang Đỉnh
23020082 Nguyễn Quốc Huy
23020061 Trần Trung Hậu**

INSIGHT TUNE PHẦN MỀM NGHE NHẠC TÍCH HỢP AI

BÀI TẬP NHÓM BỘ MÔN PHÁT TRIỂN PHẦN MỀM DI ĐỘNG

**Cán bộ hướng dẫn: TS. Nguyễn Đức Anh
ThS. Trần Mạnh Cường**

HÀ NỘI - 2025

LỜI CAM ĐOAN

Chúng tôi cam đoan toàn bộ nội dung trong báo cáo là do chúng tôi tự thực hiện. Báo cáo không sao chép từ bất kỳ nhóm nào khác hoặc sử dụng nội dung được tạo bởi trí tuệ nhân tạo (AI) một cách không hợp lệ. Nếu phát hiện có bất kỳ vi phạm nào, chúng tôi xin hoàn toàn chịu trách nhiệm.

Ngày 20 tháng 11 năm 2025

Sinh viên

**Đoàn Minh Hoàng
Trần Quang Đỉnh
Nguyễn Quốc Huy
Trần Trung Hậu**

ĐÓNG GÓP

Bảng dưới đây thể hiện công việc và nội dung báo cáo từng thành viên chịu trách nhiệm:

STT	Họ tên	Đóng góp	Công việc hoàn thành	Nội dung báo cáo
1	Nguyễn Quốc Huy	25	Xây dựng kiến trúc backend, viết mã nguồn hệ thống, triển khai hệ thống bằng nền tảng đám mây, huấn luyện mô hình gợi ý và tích hợp vào hệ thống lớn, thiết lập các ca kiểm thử	Chương 1, Chương 2(phần 1, 2, 3.1), Chương 3 (phần 1, 2), Chương 4 (phần 1, 2.2, 3), Chương 5 (phần 2)
2	Trần Quang Đỉnh	25	Viết mã nguồn Backend hệ thống, triển khai các ca kiểm thử chức năng và hiệu năng	Chương 3 (phần 3.3, 3.4), Chương 4 (phần 4.4, 4.5), Chương 6 (phần 6.2)
3	Trần Trung Hậu	25	Thiết kế giao diện màn hình và xử lý logic Frontend phát nhạc và chatbot, xây dựng hệ thống chatbot agent multi-tool, kiểm thử giao diện tìm kiếm, giao diện chatbot, hệ thống chatbot	Chương 5 (phần 5.1), Chương 6 (phần 6.2) Chương 7
4	Đoàn Minh Hoàng	25	Thiết kế giao diện và xử lý logic Frontend cho màn hình đăng nhập, đăng ký, thông tin người dùng, tìm kiếm và danh mục bài hát yêu thích, kiểm thử giao diện đăng nhập, đăng ký	Chương 4(phần 2.1), Chương 6(phần 1, phần 2)

MỤC LỤC

Lời cam đoan	
Đóng góp	i
Mục lục	iii
Danh mục hình vẽ	vii
Danh mục bảng biểu	x
Danh mục thuật ngữ và từ viết tắt	xi
CHƯƠNG 1. GIỚI THIỆU	1
1.1. Hiện trạng	2
1.2. Các hệ thống tương tự	3
1.3. Giới thiệu hệ thống nghe nhạc InsightTune	5
1.4. Cấu trúc tài liệu	6
CHƯƠNG 2. KIẾN THỨC NỀN TẢNG	8
2.1. Phân tích đặc tả yêu cầu	9
2.2. Kiến trúc hệ thống	9
2.3. Xây dựng hệ thống AI	9
2.3.1. Dịch vụ gợi ý	10
2.3.2. Trợ lý ảo	10
CHƯƠNG 3. ĐẶC TẢ YÊU CẦU	13
3.1. Thu thập và phân tích đặc tả yêu cầu	13
3.1.1. Yêu cầu người sử dụng	13
3.1.2. Yêu cầu hệ thống	13
3.2. Biểu đồ ca sử dụng	14
3.3. Đặc tả ca sử dụng	16
3.3.1. Phát bài hát	16
3.3.2. Đăng ký	17
3.3.3. Đăng nhập	18
3.3.4. Đăng xuất	19
3.3.5. Tìm kiếm bài hát	20
3.3.6. Xem bài hát gần đây	21

3.3.7. Lưu bài hát yêu thích	22
3.3.8. Sửa thông tin cá nhân	23
3.3.9. Quên mật khẩu	24
3.4. Biểu đồ hoạt động	25
3.4.1. Phát bài hát	26
3.4.2. Đăng ký	28
3.4.3. Đăng nhập	28
3.4.4. Đăng xuất	29
3.4.5. Tìm kiếm bài hát	30
3.4.6. Xem bài hát gần đây	33
3.4.7. Lưu bài hát vào yêu thích	33
3.4.8. Sửa thông tin cá nhân	34
3.4.9. Quên mật khẩu	35
CHƯƠNG 4. PHÂN TÍCH KIẾN TRÚC HỆ THỐNG	37
4.1. Yêu cầu thiết kế hệ thống	38
4.2. Kiến trúc hệ thống	38
4.2.1. Client	39
4.2.2. Server	41
4.3. Kiến trúc triển khai	45
4.4. Biểu đồ tuần tự	48
4.4.1. Chatbot	48
4.4.2. Đăng ký	49
4.4.3. Đăng nhập	51
4.4.4. Đăng xuất	52
4.4.5. Phát nhạc	53
4.4.6. Xem bài hát gần đây	54
4.4.7. Lưu bài hát yêu thích	55
4.4.8. Sửa thông tin cá nhân	56
4.4.9. Quên mật khẩu	57
4.5. Cơ sở dữ liệu	59
4.5.1. Auth database	60
4.5.2. User database	60
4.5.3. Catalog database	61

4.5.4. History database	62
4.5.5. Favorite database	62
CHƯƠNG 5. XÂY DỰNG HỆ THỐNG AI	64
5.1. Xây dựng trợ lý ảo	65
5.1.1. Bài toán đặt ra	65
5.1.2. Giải pháp đề xuất	65
5.1.3. Công nghệ và luồng hoạt động của chatbot agent	65
5.2. Xây dựng mô hình gợi ý	72
5.2.1. Giới thiệu về HAC-NSW	72
5.2.2. HAC-NSW và InsightTune	73
5.2.3. Xây dựng huấn luyện mô hình	74
CHƯƠNG 6. CÀI ĐẶT VÀ KIỂM THỬ	77
6.1. Giao diện hệ thống	77
6.1.1. Bắt đầu	77
6.1.2. Đăng ký tài khoản	79
6.1.3. Đăng nhập	83
6.1.4. Quên mật khẩu và đặt lại	85
6.1.5. Trang chủ	89
6.1.6. Trình phát nhạc	91
6.1.7. Tìm kiếm	94
6.1.8. Quản lý các bài hát yêu thích	96
6.1.9. Tài khoản và cá nhân hóa	97
6.1.10. Chatbot hỗ trợ	104
6.1.11. Một số hình ảnh của giao diện tối	106
6.2. Kiểm thử hệ thống	110
6.2.1. Công cụ kiểm thử	110
6.2.2. Thiết kế các ca kiểm thử	112
6.2.3. Thực thi các ca kiểm thử	126
6.2.4. Kiểm thử hiệu năng	138
CHƯƠNG 7. TỔNG KẾT VÀ ĐỊNH HƯỚNG TƯƠNG LAI	140
7.1. Tổng kết	140
7.1.1. Điểm cộng của hệ thống	140
7.1.2. Hạn chế của hệ thống	140

7.2. Định hướng tương lai	140
Tài liệu tham khảo	142

DANH MỤC HÌNH VẼ

Hình 3.1 Biểu đồ ca sử dụng	15
Hình 3.2 Biểu đồ hoạt động chức năng "Phát bài hát"	27
Hình 3.3 Biểu đồ hoạt động chức năng "Đăng ký"	28
Hình 3.4 Biểu đồ hoạt động chức năng "Đăng nhập"	29
Hình 3.5 Biểu đồ hoạt động chức năng "Đăng xuất"	30
Hình 3.6 Biểu đồ hoạt động chức năng "Tìm kiếm bài hát"	32
Hình 3.7 Biểu đồ hoạt động chức năng "Xem bài hát gần đây"	33
Hình 3.8 Biểu đồ hoạt động chức năng "Lưu bài hát vào yêu thích"	34
Hình 3.9 Biểu đồ hoạt động chức năng "Sửa thông tin cá nhân"	35
Hình 3.10 Biểu đồ hoạt động chức năng "Quên mật khẩu"	36
Hình 4.1 Kiến trúc client-server của hệ thống	39
Hình 4.2 Kiến trúc MVVM với Repository Pattern trên Client	41
Hình 4.3 Kiến trúc microservice chung	42
Hình 4.4 Kiến trúc server bậc cao	43
Hình 4.5 Kiến trúc 3 lớp của Spring khi xây dựng Microservice	45
Hình 4.6 Kiến trúc server khi triển khai trên AWS	47
Hình 4.7 Biểu đồ tuần tự cho quá trình chuyển đổi giọng nói thành văn bản	49
Hình 4.8 Biểu đồ tuần tự cho chatbot	50
Hình 4.9 Biểu đồ tuần tự cho chức năng đăng ký	51
Hình 4.10 Biểu đồ tuần tự cho chức năng đăng nhập	52
Hình 4.11 Biểu đồ tuần tự cho chức năng đăng xuất	53
Hình 4.12 Biểu đồ tuần tự cho chức năng phát nhạc	54
Hình 4.13 Biểu đồ tuần tự cho chức năng xem bài hát gần đây	55
Hình 4.14 Biểu đồ tuần tự cho chức năng lưu bài hát yêu thích	56
Hình 4.15 Biểu đồ tuần tự cho chức năng sửa thông tin cá nhân	57
Hình 4.16 Biểu đồ tuần tự cho chức năng quên mật khẩu	59
Hình 4.17 Cơ sở dữ liệu của AuthService	60
Hình 4.18 Cơ sở dữ liệu của UserService	61
Hình 4.19 Cơ sở dữ liệu của CatalogService	62

Hình 4.20 Cơ sở dữ liệu của HistoryService	62
Hình 4.21 Cơ sở dữ liệu của FavoriteService	63
 Hình 5.1 Luồng hoạt động của agent trong hệ thống chatbot AI	67
Hình 5.2 Luồng hoạt động của bộ nhớ trong hệ thống chatbot AI	69
Hình 5.3 Luồng hoạt động của duckduckgo tool trong hệ thống chatbot AI . .	70
Hình 5.4 Luồng hoạt động của scrape website tool trong hệ thống chatbot AI .	71
Hình 5.5 Luồng của mô hình HAC-NSW	72
Hình 5.6 Bảng biểu thể hiện tương tác trung bình khi hàm mât mát của actor cải thiện	74
Hình 5.7 Bảng biểu thể hiện kết quả thấp nhất trong người dùng và độ chênh ổn định thấp khi hàm mât mát actor cải thiện	75
Hình 5.8 Bảng biểu thể hiện kết quả trung bình tốt trên tập kiểm thử	75
Hình 5.9 Bảng biểu thể hiện độ lệch vẫn ổn định thấp trên tập kiểm thử	76
 Hình 6.1 Giao diện bắt đầu ứng dụng	79
Hình 6.2 Giao diện đăng ký tài khoản - nhập email	81
Hình 6.3 Giao diện đăng ký tài khoản - tạo mật khẩu	82
Hình 6.4 Giao diện đăng ký tài khoản - tạo tên người dùng	83
Hình 6.5 Giao diện đăng nhập	85
Hình 6.6 Giao diện quên mật khẩu - nhập email	87
Hình 6.7 Giao diện quên mật khẩu - nhập mã OTP	88
Hình 6.8 Giao diện quên mật khẩu - tạo mật khẩu mới	89
Hình 6.9 Giao diện màn hình chính	91
Hình 6.10 Giao diện trình phát nhạc	93
Hình 6.11 Giao diện phát nhạc trên thanh thông báo	94
Hình 6.12 Giao diện tìm kiếm bài hát	95
Hình 6.13 Giao diện các bài hát yêu thích	97
Hình 6.14 Giao diện thông tin tài khoản	99
Hình 6.15 Giao diện đổi ảnh đại diện	100
Hình 6.16 Giao diện cập nhật thông tin cá nhân	101
Hình 6.17 Giao diện thay đổi mật khẩu	102
Hình 6.18 Giao diện thay đổi giao diện	103
Hình 6.19 Giao diện thay đổi ngôn ngữ	104

Hình 6.20 Giao diện chatbot hỗ trợ người dùng	106
Hình 6.21 Giao diện phát nhạc ở chế độ tối	107
Hình 6.22 Giao diện tìm kiếm ở chế độ tối	108
Hình 6.23 Giao diện các bài hát yêu thích ở chế độ tối	109
Hình 6.24 Giao diện chatbot ở chế độ tối	110

DANH MỤC BẢNG BIỂU

Bảng 1.1 So sánh các hệ thống phát nhạc hiện hành	4
Bảng 3.1 Use Case: Playing a song	17
Bảng 3.2 Use Case: Đăng ký	18
Bảng 3.3 Use Case: Login	19
Bảng 3.4 Use Case: Logout	20
Bảng 3.5 Use Case: Searching Songs	21
Bảng 3.6 Use Case: History Songs	22
Bảng 3.7 Use Case: Favorite Songs	23
Bảng 3.8 Use Case: Update Personal Information	24
Bảng 3.9 Use Case: Forgot Password	25

DANH MỤC THUẬT NGỮ VÀ TỪ VIẾT TẮT

Thuật ngữ / Từ viết tắt	Ý nghĩa
Use Case	Ca sử dụng
Actor (tác nhân)	Người dùng hoặc một đối tượng bên ngoài tương tác với hệ thống
Include	Mối quan hệ bao gồm nhau giữa các Use Case
Extend	Mối quan hệ mở rộng giữa các Use Case
Pre-Condition	Tiền điều kiện
Post-Condition	Hậu điều kiện
Basic Flow	Luồng sự kiện chính
Alternative Flow	Luồng sự kiện thay thế
Exception Flow	Luồng sự kiện ngoại lệ
CRUD	Create, Read, Update, Delete
HTTPS	Hypertext Transfer Protocol
SQL	Ngôn ngữ truy vấn dữ liệu
CSDL	Cơ sở dữ liệu
LLMs	Large Language Models
RS	Recommender Systems
API	Application Programming Interface
UI	User Interface
CSRF	Cross-site Request Forgery
Tool	Công cụ
Agent	Tác nhân AI có khả năng sử dụng công cụ
Function Call-ing	Cơ chế gọi hàm/công cụ của LLM
MVVM	Model-View-ViewModel
JSON	JavaScript Object Notation

Thuật ngữ / Từ viết tắt	Ý nghĩa
Model	Mô hình dữ liệu
View	Giao diện người dùng
ViewModel	Mô hình giao diện
Repository	Kho lưu trữ dữ liệu
Pattern	Mẫu thiết kế
Microservice	Một kiến trúc mà ứng dụng cấu thành từ nhiều dịch vụ nhỏ, độc lập

CHƯƠNG 1

GIỚI THIỆU

Hiện nay, việc nghe nhạc trên điện thoại trở nên phổ biến hơn bao giờ hết. Bất kỳ ai cũng có thể biến chiếc điện thoại của mình thành một chiếc máy nghe nhạc di động ngay từ những phần mềm nghe nhạc được cài sẵn trên điện thoại. Tuy nhiên, những phần mềm nghe nhạc cổ điển yêu cầu những file âm thanh sẵn có trên thiết bị mới có thể phát được nhạc. Điều này dẫn đến những phần mềm nghe nhạc thế hệ mới sẽ tập trung vào việc phát trực tiếp các bài hát từ hệ thống lưu trữ đám mây (streaming). Tuy nhiên, các phần mềm này là những phần mềm có mã nguồn đóng và yêu cầu trả phí để sử dụng các tính năng. Nhằm tạo ra một framework mã nguồn mở, miễn phí và khả năng tùy chỉnh cao, chúng tôi giới thiệu InsightTune - một ứng dụng nghe nhạc kết hợp AI theo kiến trúc Client - Server.

1.1. Hiện trạng

Âm nhạc là điều không thể thiếu trong cuộc sống hiện đại ngày nay. Những phần mềm nghe nhạc đã ra đời từ rất lâu. Chúng chỉ đơn giản là những trình phát nhạc cơ bản, cho phép người dùng phát các file âm thanh có sẵn trên thiết bị. Tuy nhiên, với sự phát triển của công nghệ và Internet, các phần mềm nghe nhạc trực tuyến dần ra đời thay thế những phần mềm này trong kỷ nguyên của công nghệ đám mây và phi tập trung hóa.

Những phần mềm nghe nhạc trực tuyến cùng cấu trúc phi tập trung này cho phép người dùng phát nhạc tốc độ cao từ các hệ thống lưu trữ đám mây mà không cần phải tải các file âm thanh về thiết bị. Điều này giúp người dùng tiết kiệm dung lượng lưu trữ trên thiết bị cũng như dễ dàng tiếp cận với kho âm nhạc khổng lồ từ Internet. Người dùng khi đổi mặt với nhiều lựa chọn khi nghe như vậy lại cảm thấy bối rối vì không biết nên chọn sản phẩm âm nhạc nào để nghe. Để giải quyết vấn đề này, các phần mềm nghe nhạc ngày nay đang dần tích hợp các công nghệ trí tuệ nhân tạo (AI) để cá nhân hóa trải nghiệm nghe nhạc cho người dùng.

Công nghệ trí tuệ nhân tạo (AI) đã và đang được ứng dụng rộng rãi trong nhiều lĩnh vực khác nhau, bao gồm cả ngành công nghiệp âm nhạc. Trong lĩnh vực âm nhạc, AI có hai ứng dụng lớn dựa trên hai nhánh công nghệ của AI: mô hình ngôn ngữ lớn (Large Language Models - LLMs) và hệ thống gợi ý (Recommender Systems - RS). LLMs mở ra cơ hội tạo ra những Chatbot như một trợ lý ảo giúp người dùng tìm kiếm, lựa chọn và quản lý âm nhạc một cách dễ dàng. Điều này tối ưu hóa, đơn giản hóa trải nghiệm người dùng khi tương tác với phần mềm nghe nhạc. Hệ thống gợi ý (Recommender Systems) sử dụng các thuật toán học máy để phân tích hành vi nghe nhạc của người dùng, từ đó đề xuất các bài hát, nghệ sĩ hoặc danh sách phát phù hợp với sở thích cá nhân của họ. Điều này giúp người dùng khám phá âm nhạc mới một cách hiệu quả và nâng cao trải nghiệm nghe nhạc tổng thể. Những phần mềm nghe nhạc hiện nay đều cố gắng tích hợp những công nghệ này để nâng cao và cá nhân hóa trải nghiệm người dùng. Tuy nhiên, với những phần mềm sẵn có hiện nay, những người phát triển độc lập hoặc các nhóm nhỏ sẽ gặp nhiều khó khăn trong việc cải tiến và tích hợp.

Các phần mềm phổ biến hiện nay như Spotify, Apple Music, YouTube Music, Zing MP3, SoundCloud đều đã có cấu trúc rất lớn và phức tạp. Vấn đề lớn ở đây là chúng thường là phần mềm có mã nguồn đóng, và việc thêm tính năng hay tích hợp những công nghệ mới vào chúng là điều khó khăn đặc biệt đối với những nhà phát triển độc lập

hoặc các nhóm nhỏ. Do đó, cộng đồng các nhà phát triển đã cùng nhau tạo ra nhiều dự án mã nguồn mở nhằm mô phỏng lại và cung cấp các tính năng tương tự như những phần mềm nghe nhạc lớn hiện nay. Điều này lại khiến những người mới bắt đầu phát triển khi độ phức tạp của dự án mã nguồn mở này ngày càng cao và chưa sẵn có một phương thức tích hợp các công nghệ AI tiên tiến vào các dự án này. Vì vậy, chúng tôi xin giới thiệu **InsightTune**, một ứng dụng nghe nhạc mã nguồn mở tích hợp AI.

Những đóng góp chính và mục tiêu của dự án của chúng tôi khi phát triển Insight-Tune bao gồm:

- **Phát triển một ứng dụng nghe nhạc mã nguồn mở, miễn phí và dễ dàng tùy chỉnh:** front end mobile app với giao diện đơn giản dễ dàng sử dụng, xây dựng back end server theo kiến trúc Microservices như một framework mở rộng sau này.
- **Tích hợp các công nghệ AI tiên tiến:** LLMs(Xây dựng tác tử) và RS (Hệ thống gợi ý sử dụng học tăng cường) trong ứng dụng.
- **Xây dựng mẫu triển khai trên nền tảng đám mây.**
- **Cung cấp các ca kiểm thử để đảm bảo chất lượng và tính ổn định hệ thống.**

1.2. Các hệ thống tương tự

Những phần mềm nghe nhạc hiện rất đa dạng và phong phú, từ những ứng dụng phổ biến như Spotify, Apple Music, YouTube Music, Zing MP3, SoundCloud cho đến những ứng dụng mã nguồn mở như SpTube. Mỗi ứng dụng đều có những đặc điểm nổi bật và nhược điểm riêng. Để đưa cho người đọc có cái nhìn tổng quan về các hệ thống phát nhạc hiện hành, chúng tôi so sánh một số hệ thống phát nhạc phổ biến hiện nay trong bảng 1.1. Qua việc xem xét những nhược điểm này, chúng tôi nhận thấy rằng việc phát triển một dự án mã nguồn mở tích hợp AI sẽ mang lại nhiều lợi ích cho cộng đồng người dùng và các nhà phát triển. Điều này là động lực chính để chúng tôi phát triển **InsightTune**, một phần mềm với mã nguồn đơn giản, dễ dàng tùy chỉnh để làm framework ví dụ về việc tích hợp AI trong một ứng dụng nghe nhạc hiện đại.

Tên hệ thống	Đặc điểm nổi bật	Nhược điểm
Spotify	Giao diện thân thiện, đa nền tảng, hệ thống gợi ý bài hát tốt	Phần mềm mã nguồn đóng, chủ yếu hoạt động theo mô hình thuê bao, không có chat bot trợ lý ảo cá nhân
YouTube Music	Thư viện bài hát và video rất lớn, tích hợp chặt với tài khoản Google, hỗ trợ cá nhân hóa playlist	Nhiều quảng cáo ở bản miễn phí, phụ thuộc vào tài khoản Google, không phải mã nguồn mở nên khó tuỳ chỉnh
Apple Music	Chất lượng âm thanh cao, tích hợp tốt với hệ sinh thái thiết bị Apple, hỗ trợ lossless/spatial audio	Chủ yếu hỗ trợ tốt trên thiết bị Apple, yêu cầu trả phí theo tháng, phần mềm mã nguồn đóng
Zing MP3	Phổ biến tại Việt Nam, thư viện nhạc Việt phong phú, hỗ trợ nghe offline	Tập trung nhiều vào nội dung trong nước, có quảng cáo trên bản miễn phí, không hỗ trợ tuỳ chỉnh sâu hệ thống
SoundCloud	Cộng đồng nghệ sĩ độc lập lớn, dễ dàng upload và chia sẻ bản nhạc cá nhân	Chất lượng và độ ổn định nội dung không đồng đều, ít tính năng gợi ý thông minh so với các nền tảng lớn
SpoTube	Miễn phí, kết hợp được với YouTube	Giao diện chưa thân thiện, chưa có hệ thống gợi ý bài hát

Bảng 1.1. So sánh các hệ thống phát nhạc hiện hành

1.3. Giới thiệu hệ thống nghe nhạc InsightTune

Nhận xét về hiện trạng và xem xét các hệ thống hiện có trên thị trường cũng như các nguồn mở, chúng tôi nhận thấy việc cần thiết để tạo ra một hệ thống đơn giản nhưng cụ thể và chi tiết về việc tích hợp AI vào trong các hệ thống nghe nhạc hiện hành. Hệ thống này đảm bảo hoàn thiện các ca sử dụng (xem thêm tại 3), cũng như xây dựng kiến trúc dễ dàng mở rộng, triển khai 4. Các ca sử dụng cần đảm bảo độ ổn định thông qua các tầng test khác nhau, được viết chạy tự động hóa để đảm bảo khả năng tái sử dụng. Hệ thống sau khi hoàn thiện được đóng gói tốt, có sẵn khuôn mẫu cũng như cách triển khai đơn giản trên hệ thống đám mây với back end và trên nền tảng di động với front end. Chúng sau tổng hợp những vấn đề phát sinh từ hiện trạng này, tại đây xin được đề xuất và giới thiệu InsightTune, một hệ thống mã nguồn mở nghe nhạc tích hợp AI.

InsightTune hướng tới giải quyết những **vấn đề** sau:

- Hệ thống nghe nhạc mã nguồn mở chưa tích hợp AI.
- Độ phức tạp và khả năng mở rộng của các hệ thống.
- Khả năng tái sử dụng các thành phần trong hệ thống
- Công cụ cho những nhà phát triển mới thử nghiệm tích hợp AI vào hệ thống
- Khả năng triển khai đơn giản trên nền tảng đám mây như Amazon Web Services (AWS).

Những **giải pháp** chúng tôi ứng dụng trong **InsightTune** gồm có:

- Kiến trúc Microservices để xây dựng hệ thống back end, giúp dễ dàng mở rộng và tái sử dụng các thành phần.
- LLM gọi API đơn giản để xây dựng Chatbot trợ lý ảo giúp người dùng tương tác và quản lý âm nhạc.
- Hệ thống gợi ý RS để cá nhân hóa trải nghiệm nghe nhạc cho người dùng.
- Xây dựng front end mobile app với giao diện đơn giản, dễ sử dụng.
- Cung cấp các ca kiểm thử tự động để cho nhà phát triển đảm bảo độ ổn định khi mở rộng hệ thống.
- Cung cấp hướng dẫn triển khai hệ thống đơn giản trên nền tảng đám mây AWS.

Tóm lại, **InsightTune** là một hệ thống nghe nhạc mã nguồn mở tích hợp AI, được

thiết kế để giải quyết các vấn đề hiện tại trong lĩnh vực này. Nó cung cấp một framework dễ dàng mở rộng, tái sử dụng và triển khai, giúp các nhà phát triển mới có thể thử nghiệm và tích hợp các công nghệ AI tiên tiến vào hệ thống nghe nhạc.

1.4. Cấu trúc tài liệu

Để giúp người đọc dễ nắm bắt được nội dung tài liệu, chúng tôi đã tổ chức tài liệu này thành 7 chương. Trong phần này, chúng tôi sẽ giới thiệu sơ lược về từng chương trong tài liệu.

7 chương trong tài liệu bao gồm:

- Chương 1: Giới thiệu
- Chương 2: Kiến thức nền tảng
- Chương 3: Thu thập và phân tích yêu cầu
- Chương 4: Phân tích, thiết kế hệ thống
- Chương 5: Quá trình xây dựng hệ thống tích hợp AI
- Chương 6: Cài đặt và kiểm thử
- Chương 7: Tổng kết và định hướng tương lai

Mỗi chương sẽ tập trung vào một khía cạnh cụ thể của dự án, được ghi lại và tổng hợp trong suốt quá trình làm việc theo quy trình phát triển phần mềm Scrum của nhóm chúng tôi. Những kiến thức nền tảng cần có để đọc và theo dõi dự án được trình bày tại chương 2. Trong các chương 3 và 4, dự án được phân tích từ những yêu cầu nền tảng được đề ra, sau đó thiết kế kiến trúc hệ thống để đáp ứng các yêu cầu này. Đến chương 4, chúng ta có thể đã có một hệ thống nghe nhạc cơ bản. Tiếp theo, trong chương 5, chúng tôi trình bày chi tiết quá trình xây dựng hệ thống AI tích hợp trong dự án. Chương 6 tập trung vào quá trình cài đặt và kiểm thử để đảm bảo hệ thống hoạt động ổn định và đáp ứng các yêu cầu đã đề ra. Việc này được thực hiện xuyên suốt dự án nhưng để thuận tiện cho người đọc, chúng tôi sẽ tổng hợp toàn bộ ca kiểm thử trong chương này. Cuối cùng, chương 7 sẽ tổng kết lại toàn bộ dự án, những đóng góp chính và định hướng phát triển trong tương lai.

KẾT CHƯƠNG

Chúng tôi đã trình bày chi tiết hiện trạng, những giải pháp hiện có, động lực cũng như khái quát dự án **InsightTune** trong chương này. Trong các chương sau, chúng tôi sẽ chi tiết hóa những nội dung đã được đề cập để người đọc hiểu rõ hơn về dự án và giải pháp được sử dụng trong dự án.

CHƯƠNG 2

KIẾN THỨC NỀN TẢNG

Để giúp người đọc có cái nhìn rõ hơn về hệ thống, chúng tôi xin dành riêng một chương để trình bày về kiến thức nền tảng của hệ thống **InsightTune**. Chương này sẽ trình bày về những kiến thức chúng tôi đã vận dụng trong quá trình làm việc, phân tích đặc tả yêu cầu, thiết lập kiến trúc hệ thống và xây dựng hệ thống AI.

2.1. Phân tích đặc tả yêu cầu

Muốn xây dựng một hệ thống, trước hết, chúng ta cần biết rõ về những yêu cầu mà hệ thống cần phải có. Từ đó, những kiến thức về thu thập, phân tích, đặc tả yêu cầu là bắt buộc khi xây dựng hệ thống. Trong quá trình xây dựng **InsightTune**, chúng tôi đã vận dụng những kiến thức sau đây:

- Thu thập, tổng hợp yêu cầu.
- Phân tích đặc tả yêu cầu.
- Biểu diễn thông qua bảng đặc tả, biểu đồ ca sử dụng và biểu đồ hoạt động.

2.2. Kiến trúc hệ thống

Hệ thống **InsightTune** gồm rất nhiều công cụ và module khác nhau, để hiểu được kiến trúc bậc cao của hệ thống này, người đọc cần có kiến thức về:

- Xây dựng API và cân bằng tải
- Kiến trúc Client-Server
- Kiến trúc MVVC
- Kiến trúc Microservice
- Phương thức giao tiếp HTTPS và gRPC
- Kafka và xử lý sự kiện đa luồng
- Công cụ triển khai trên nền tảng AWS
- Hệ quản trị cơ sở dữ liệu và cơ sở dữ liệu
- Các cơ chế bảo mật

Những kiến thức này sẽ được nhắc lại và sử dụng trong chương 4, nơi chúng tôi sẽ trình bày và phân tích những yêu cầu, đặc điểm, hành vi của kiến trúc hệ thống **InsightTune**.

2.3. Xây dựng hệ thống AI

Hệ thống **InsightTune** có tích hợp hai dịch vụ (service) AI, hai dịch vụ này được triển khai sao cho tương thích tốt với toàn bộ hệ thống và có tính đặc trưng cao (phân biệt với những dịch vụ AI cho đa tác vụ như: Gemini hay ChatGPT). Chúng tôi xin trình bày những kiến thức nền tảng để xây dựng hiệu quả hai dịch vụ này qua hai phần riêng biệt dưới đây: dịch vụ gợi ý và trợ lý ảo.

2.3.1. Dịch vụ gợi ý

Để xây được một dịch vụ gợi ý tốt trong một thông lớn của **InsightTune**, ta cần hiểu hai miền kiến thức chính:

- Một là về kiến trúc hệ thống, để từ đó hiểu được đầu vào và đầu ra của dịch vụ cũng như phương thức giao tiếp với những tác vụ khác của hệ thống.
- Hai là về xây dựng và triển khai một mô hình gợi ý.

Những yêu cầu kiến thức về kiến trúc hệ thống đã được trình bày tại 2.2 nên ở phần này chúng tôi chỉ tập trung vào giải thích ngắn gọn những kiến thức xây dựng và triển khai mô hình gợi ý.

2.3.1.1. Xây dựng

Xây dựng mô hình gợi ý tức là chuẩn bị tập dữ liệu mẫu chia ra làm tập huấn luyện và tập kiểm tra, sau khi hoàn tất quá trình huấn luyện trên tập huấn luyện, ta thu được một mô hình hoạt động tốt trên tập kiểm tra.

Kiến thức cần có để xây dựng lại một mô hình gợi ý như chúng tôi làm đó là:

- Kiến thức về học tăng cường
- Xử lý dữ liệu
- Các thư viện học máy trong Python: Pytorch, numpy, pandas
- Công cụ đo lường và biểu diễn: wandb

2.3.1.2. Triển khai

Sau khi có được một mô hình gợi ý, chúng tôi triển khai mô hình như một dịch vụ trong hệ thống. Điều này phải tương thích với yêu cầu đầu vào và đầu ra của dịch vụ.

Những kiến thức cần có để triển khai hiệu quả là:

- Kiến thức xây dựng dịch vụ bằng FastAPI
- Thiết lập đóng gói dịch vụ bằng Docker

2.3.2. Trợ lý ảo

Chúng tôi xây dựng một hệ thống chatbot agent kết hợp với các tool:

- Duckduckgo tool: là công cụ tra cứu, tìm kiếm thông tin trên mạng.
- Scrape Website tool: là công cụ để lấy dữ liệu từ một trang web.

Hệ thống sử dụng LLM là gemini-2.5-pro làm model chính để xử lí input đầu vào và kết hợp những tool để trả lời người dùng. Dưới đây là kiến thức cần có để xây dựng

hệ thống chatbot agent này:

- Nguyên lý hoạt động của LLM và cơ chế function calling/tool use.
- Kiến thức Agent-base Chatbot.
- Thiết kế và tích hợp tools vào LLM.
- Kỹ năng viết prompt hiệu quả.
- Framework Langgraph để xây dựng stateful multi-agent workflows.
- Quản lý bộ nhớ hội thoại (memory management) với Checkpointer.

Và được triển khai thông qua FastAPI.

KẾT CHƯƠNG

Trong chương 2, chúng tôi đã liệt kê và giải thích đơn giản những khái niệm và kiến thức được vận dụng trong quá trình xây dựng hệ thống **InsightTune**. Những kiến thức này sẽ được nhắc lại và sử dụng xuyên suốt các chương sau của báo cáo. Cụ thể, tại chương 3, kiến thức về thu thập phân tích đặc tả yêu cầu sẽ được cụ thể hóa thông qua kết quả là yêu cầu, bảng biểu. Tiếp theo tại chương 4, những mô tả, phân tích về kiến trúc hệ thống sẽ được trình bày rõ ràng lấy nền tảng là kiến thức trong chương 2. Do đó, việc hiểu và nắm chắc những kiến thức nền tảng tại chương này là bắt buộc để đọc hiểu phần tiếp theo của báo cáo này.

CHƯƠNG 3

ĐẶC TẢ YÊU CẦU

3.1. Thu thập và phân tích đặc tả yêu cầu

3.1.1. Yêu cầu người sử dụng

- Có thể đăng nhập vào hệ thống an toàn.
- Có thể phát được bài hát.
- Có thể thêm bài hát yêu thích.
- Có thể đăng ký tài khoản mới.
- Có thể lấy lại mật khẩu khi quên.
- Có thể xem bài hát phát gần đây
- Có thể trò chuyện cùng chatbot
- Có thể nhận bài hát gợi ý
- Có thể sửa thông tin cá nhân

3.1.2. Yêu cầu hệ thống

3.1.2.1. Yêu cầu chức năng

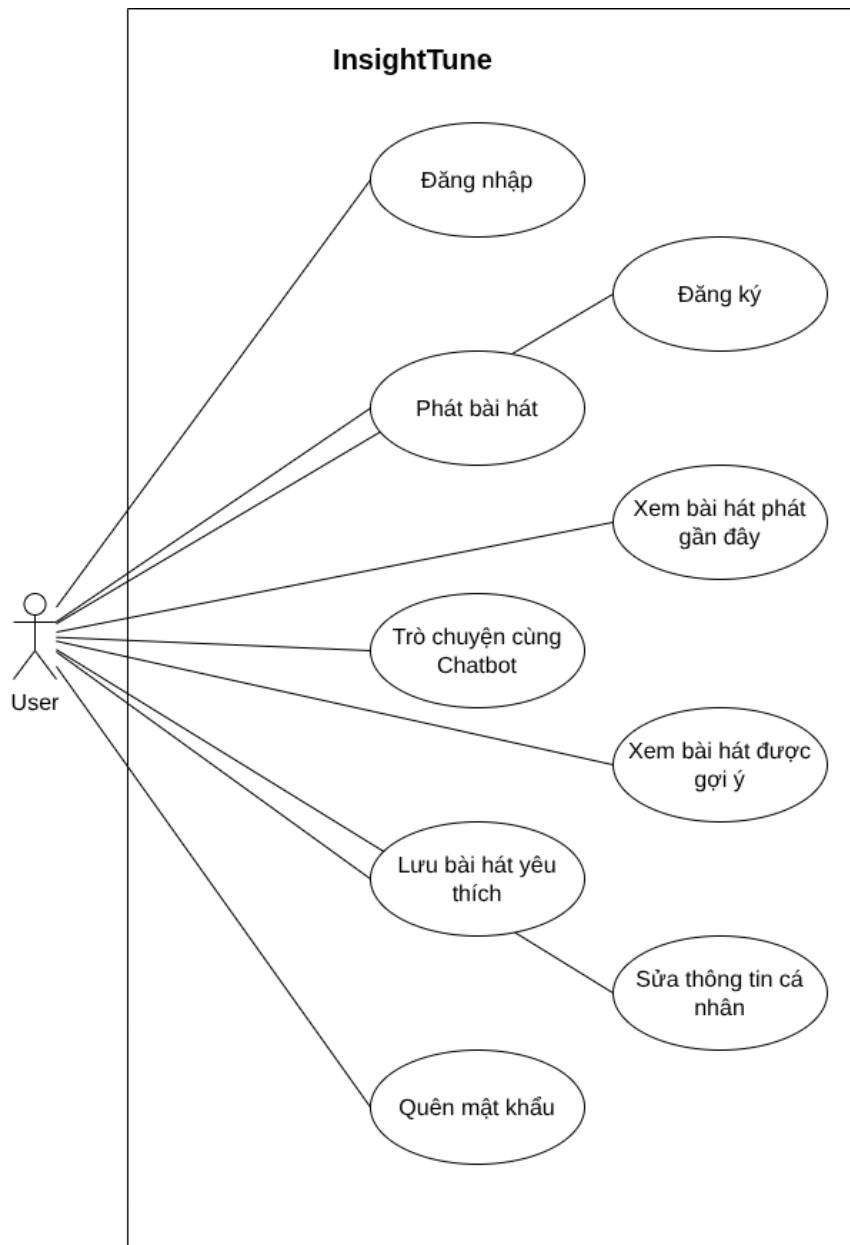
- Người dùng có thể đăng nhập bằng email và mật khẩu, thông qua Google
- Người dùng đăng ký được tài khoản mới
- Khi quên mật khẩu, người dùng có thể lấy lại
- Người dùng phát được bài hát
- Người dùng mở lại xem được những bài hát gần đây
- Người dùng trò chuyện được với chatbot và được hỗ trợ
- Người dùng sửa thông tin cá nhân khi có nhu cầu
- Người dùng nhận những bài hát được gợi ý
- Khi thích, người dùng lưu lại được bài hát

3.1.2.2. Yêu cầu phi chức năng

- Hiệu suất: Hệ thống xử lý yêu cầu hiệu quả, phản hồi nhanh, chịu tải tốt
- Bảo mật: Có cơ chế bảo mật riêng, bảo mật thông tin người dùng
- Độ chính xác: hệ thống gợi ý và chatbot có khả năng phản hồi chính xác

3.2. Biểu đồ ca sử dụng

Hệ thống gồm một tác nhân duy nhất là người dùng. Người dùng có thể thực hiện các chức năng chính như trong hình 3.1.



Hình 3.1. Biểu đồ ca sử dụng

Từ việc xem xét những ca sử dụng này, chúng tôi đã phát triển kiến trúc của hệ thống theo mô hình Client - Server, trong đó Server được thiết kế theo kiến trúc Microservices để dễ dàng mở rộng với các tính năng mới trong tương lai.

Các ca sử dụng trong biểu đồ được chia cho nhiều service khác nhau, với các phương thức giao tiếp và truyền tin cần thiết (chi tiết ở 4.2). Chúng tôi sẽ chỉ lưu ý và giải thích chi tiết những ca sử dụng quan trọng trong các phần sau của báo cáo.

3.3. Đặc tả ca sử dụng

Trong phần này, những ca sử dụng được nhắc tới trong biểu đồ ca sử dụng sẽ được mô tả chi tiết thông qua các đặc tả ca sử dụng (use case specification).

3.3.1. Phát bài hát

Ca sử dụng phát bài hát là trọng tâm của một ứng dụng nghe nhạc. Chúng tôi đã phân tích và giải thích chi tiết trong đặc tả ca sử dụng dưới đây.

Mã Use case	UC-PlaySong
Tên Use case	Playing a song
Tác nhân	Người dùng
Mô tả	Người dùng bấm vào một bản nhạc trên màn hình chính để phát bản nhạc đó.
Sự kiện kích hoạt	Người dùng bấm vào bản nhạc
Pre-condition	<ul style="list-style-type: none">• Người dùng đã đăng nhập vào app và hình ảnh các bản nhạc đã được load trên màn hình chính.• Kết nối internet ổn định.

	STT	Sự kiện	
Basic flow	1	Người dùng bấm vào hình bản nhạc trên màn hình chính	
	2	Client (mobile app) gửi ID bản nhạc người dùng bấm vào về API Gateway của Server	
	3	API Gateway gửi ID bản nhạc tới Track Service	
	4	Track Service tìm trong cơ sở dữ liệu và trả về Storage Key của bản nhạc trên Amazon Cloud	
	5	Client gọi Playing Service với Key để lấy link bản nhạc qua API Gateway	
	6	Client load giao diện phát nhạc cho người dùng	
Alternate flow: Client thoát giao diện phát nhạc	STT	Sự kiện	
	6.1.1	Client thoát giao diện phát nhạc về home của ứng dụng	
	6.1.2	Client hiện thanh điều khiển nhạc ở dưới cùng màn hình	
Post-condition	Bản nhạc được phát và người dùng có thể điều khiển được bản nhạc		
Other	Bản nhạc hiện trên màn hình chính được đảm bảo có trong database và được upload lên Object Storage Bucket		

Bảng 3.1. Use Case: Playing a song

3.3.2. Đăng ký

Ca sử dụng đăng ký cho phép người dùng tạo tài khoản để sử dụng ứng dụng.

Mã Use case	UC-Register
Tên Use case	Đăng ký

Tác nhân	Người dùng		
Mô tả	Người dùng đăng ký để tạo tài khoản		
Sự kiện kích hoạt	Người dùng bấm vào nút "Create An Account"		
Pre-condition	<ul style="list-style-type: none"> Giao diện đăng ký được hiển thị. Kết nối internet ổn định. 		
Basic flow	STT	Sự kiện	
	1	Người dùng nhập email, password, confirmPassword, firstName, lastName vào form đăng ký	
	2	Hệ thống kiểm tra định dạng và tính hợp lệ của các trường	
	3	Nếu hợp lệ, hệ thống gửi thông tin đến server	
	4	Server kiểm tra thông tin, nếu chưa tồn tại thì lưu thông tin và phản hồi kết quả	
Alternate flow: Các thao tác thay thế trong quá trình đăng ký	STT	Sự kiện	
	3.1	Nếu thông tin không hợp lệ, hiển thị thông báo lỗi	
	4.1	Nếu email đã tồn tại, hiển thị thông báo lỗi	
Post-condition	Người dùng có thể đăng nhập bằng thông tin vừa đăng ký		
Other	<ul style="list-style-type: none"> Email đăng ký được đảm bảo đúng định dạng và chưa từng được đăng ký trước đây. Thông tin yêu cầu được điền đầy đủ. 		

Bảng 3.2. Use Case: Đăng ký

3.3.3. Đăng nhập

Ca sử dụng đăng nhập cho phép người dùng đăng nhập vào ứng dụng để sử dụng.

Mã Use case	UC-Login												
Tên Use case	Login												
Tác nhân	Người dùng												
Mô tả	Người dùng đăng nhập để sử dụng hệ thống.												
Sự kiện kích hoạt	Người dùng bấm vào nút "Log in"												
Pre-condition	<ul style="list-style-type: none"> Giao diện đăng nhập đã hiển thị. Kết nối internet ổn định. 												
Basic flow	<table border="1"> <thead> <tr> <th>STT</th> <th>Sự kiện</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Người dùng nhập email và mật khẩu vào form</td> </tr> <tr> <td>2</td> <td>Người dùng nhấn nút "Log in"</td> </tr> <tr> <td>3</td> <td>Hệ thống gửi thông tin đến server</td> </tr> <tr> <td>4</td> <td>Server kiểm tra thông tin và phản hồi kết quả</td> </tr> <tr> <td>5</td> <td>Client load giao diện home của ứng dụng cho người dùng sử dụng</td> </tr> </tbody> </table>	STT	Sự kiện	1	Người dùng nhập email và mật khẩu vào form	2	Người dùng nhấn nút "Log in"	3	Hệ thống gửi thông tin đến server	4	Server kiểm tra thông tin và phản hồi kết quả	5	Client load giao diện home của ứng dụng cho người dùng sử dụng
STT	Sự kiện												
1	Người dùng nhập email và mật khẩu vào form												
2	Người dùng nhấn nút "Log in"												
3	Hệ thống gửi thông tin đến server												
4	Server kiểm tra thông tin và phản hồi kết quả												
5	Client load giao diện home của ứng dụng cho người dùng sử dụng												
Alternate flow: Đăng nhập cách khác	2.1 Người dùng nhấn nút "Login with Google" để đăng nhập bằng Google												
Post-condition	<ul style="list-style-type: none"> Hệ thống khởi tạo phiên làm việc hợp lệ cho người dùng. Giao diện chính được hiển thị để bắt đầu sử dụng hệ thống. 												
Other	Thông tin đăng nhập hợp lệ												

Bảng 3.3. Use Case: Login

3.3.4. Đăng xuất

Ca sử dụng đăng xuất cho phép người dùng đăng xuất tài khoản ra khỏi ứng dụng.

Mã Use case	UC-Logout
Tên Use case	Logout

Tác nhân	Người dùng		
Mô tả	Người dùng đăng xuất khỏi hệ thống.		
Sự kiện kích hoạt	Người dùng bấm vào nút "Log out"		
Pre-condition	<ul style="list-style-type: none"> • Người dùng đã đăng nhập thành công. • Kết nối internet ổn định. 		
Basic flow	STT	Sự kiện	
	1	Người dùng ấn vào "Setting"	
	2	Người dùng nhấn nút "Log out"	
	3	Hệ thống gửi thông tin đến server	
	4	Server kiểm tra thông tin và xóa token	
	5	Client load giao diện bắt đầu	
Alternate flow:	None		
Post-condition	<ul style="list-style-type: none"> • Giao diện bắt đầu được hiển thị. 		
Other	None		

Bảng 3.4. Use Case: Logout

3.3.5. Tìm kiếm bài hát

Ca sử dụng tìm kiếm bài hát cho phép người dùng có thể tìm kiếm các bài hát theo tên bài hát hoặc tên tác giả.

Mã Use case	UC-SearchSong		
Tên Use case	Seach songs		
Tác nhân	Người dùng		
Mô tả	Người dùng tìm kiếm bài hát.		
Sự kiện kích hoạt	Người dùng nhập keyword vào ô search		
Pre-condition	<ul style="list-style-type: none"> • Người dùng đã đăng nhập thành công. • Kết nối internet ổn định. 		

	STT	Sự kiện	
Basic flow	1	Người dùng ấn vào nút "Search"	
	2	Người dùng nhập tên bài hát	
	3	Hệ thống gửi tên bài hát đến server	
	4	Server kiểm tra thông tin và trả về kết quả	
	5	Client load các bài hát lên giao diện	
Alternate flow: Tìm kiếm theo tác giả	2.1	Người dùng nhập tên tác giả	
Post-condition		<ul style="list-style-type: none"> Các bài hát được hiển thị trên giao diện và người dùng có thể phát nhạc. 	
Other		None	

Bảng 3.5. Use Case: Searching Songs

3.3.6. Xem bài hát gần đây

Ca sử dụng xem bài hát gần đây cho phép người dùng có thể xem lịch sử phát bài hát.

Mã Use case	UC-HistorySong
Tên Use case	History songs
Tác nhân	Người dùng
Mô tả	Người dùng xem lịch sử bài hát.
Sự kiện kích hoạt	Người dùng vào giao diện home của ứng dụng
Pre-condition	<ul style="list-style-type: none"> Người dùng đã đăng nhập thành công. Kết nối internet ổn định.

	STT	Sự kiện
Basic flow	1	Người dùng vào giao diện home của ứng dụng
	2	Client gửi thông tin người dùng lên server để lấy lịch sử nghe
	3	Server kiểm tra thông tin và trả về kết quả
	4	Client load các bài hát server trả về lên giao diện
Alternate flow:	None	
Post-condition	<ul style="list-style-type: none"> Các bài hát được hiển thị trên giao diện và người dùng có thể phát nhạc. 	
Other	Đảm bảo người dùng đã từng phát nhạc trước đó	

Bảng 3.6. Use Case: History Songs

3.3.7. Lưu bài hát yêu thích

Ca sử dụng lưu bài hát yêu thích cho phép người dùng có thể lưu các bài hát vào mục yêu thích.

Mã Use case	UC-FavoriteSong
Tên Use case	Favorite songs
Tác nhân	Người dùng
Mô tả	Người dùng lưu bài hát yêu thích.
Sự kiện kích hoạt	Người dùng ấn nút "Favorite"
Pre-condition	<ul style="list-style-type: none"> Người dùng đã đăng nhập thành công. Người dùng đang phát nhạc và giao diện bài hát đã hiển thị. Kết nối internet ổn định.

	STT	Sự kiện	
Basic flow	1	Người dùng nhấn nút "Favorite" ở góc dưới cùng bên phải màn hình	
	2	Client gửi thông tin bài hát, người dùng lên server để thêm vào mục yêu thích	
	3	Server kiểm tra thông tin và lưu vào CSDL	
	4	Server trả về kết quả	
Alternate flow:	None		
Post-condition	<ul style="list-style-type: none"> Bài hát được thêm vào mục yêu thích. 		
Other	None		

Bảng 3.7. Use Case: Favorite Songs

3.3.8. Sửa thông tin cá nhân

Ca sử dụng sửa thông tin cá nhân cho phép người dùng có thể sửa các thông tin firstname, lastname, phone, address và mật khẩu.

Mã Use case	UC-UpdateProfile
Tên Use case	Update Profile
Tác nhân	Người dùng
Mô tả	Người dùng có thể sửa các thông tin firstname, lastname, address, phone và mật khẩu tài khoản.
Sự kiện kích hoạt	Người dùng vào trang cài đặt, ấn nút "Save" hoặc "Change Password" nếu đổi mật khẩu
Pre-condition	<ul style="list-style-type: none"> Người dùng đã đăng nhập thành công. Kết nối internet ổn định.

	STT	Sự kiện		
Basic flow	1	Người dùng nhấn vào avatar ở góc trên cùng bên phải màn hình		
	2	Người dùng chọn "Edit Profiles"		
	3	Người dùng nhập thông tin vào form đổi thông tin		
	4	Server lưu kết quả		
	5	Giao diện hiển thị thông tin đã cập nhật		
Alternate flow: Người dùng thay đổi mật khẩu	STT	Sự kiện		
	2.1.1	Người dùng chọn "Change Password"		
	3.1.2	Người dùng nhập thông tin vào form đổi mật khẩu		
	2.2.1	Người dùng nhấn vào avatar		
	3.2.2	Người dùng chọn avatar mới		
Post-condition	Thông tin người dùng được cập nhật			
Other	Giao diện hiện thông tin đã cập nhật			

Bảng 3.8. Use Case: Update Personal Information

3.3.9. Quên mật khẩu

Ca sử dụng quên mật khẩu cho phép người dùng có thể thay đổi mật khẩu khi đã quên mật khẩu cũ.

Mã Use case	UC-ForgotPassword
Tên Use case	Forgot Password
Tác nhân	Người dùng
Mô tả	Người dùng có thể sửa mật khẩu ngay cả khi quên mật khẩu cũ.
Sự kiện kích hoạt	Người dùng nhấn "Forgot Password"
Pre-condition	<ul style="list-style-type: none"> • Người dùng chưa đăng nhập. • Kết nối internet ổn định.

	STT	Sự kiện	
Basic flow	1	Người dùng nhấn vào "Forgot Password" ở trong giao diện đăng nhập	
	2	Người dùng nhập email đã dùng để đăng ký, chuyển đến giao diện xác thực otp	
	3	Server tiếp nhận email, gửi một mã otp xác nhận tới email và lưu otp vào CSDL	
	4	Người dùng nhận mã otp qua mail và nhập vào ô xác thực, chuyển đến giao diện đặt lại mật khẩu	
	5	Người dùng nhập thông tin vào form cài đặt mật khẩu mới, server tiếp nhận thông tin và thay đổi mật khẩu	
	7	Trả về thông báo thành công, chuyển đến giao diện đăng nhập	
Alternate flow: Người dùng thay đổi mật khẩu	STT	Sự kiện	
	4.1	Nếu otp không khớp, hiển thị thông báo lỗi	
Post-condition		Mật khẩu người dùng được cập nhật	
Other		Chuyển đến giao diện đăng nhập	

Bảng 3.9. Use Case: Forgot Password

3.4. Biểu đồ hoạt động

Phần Activity Diagram (Biểu đồ hoạt động) được sử dụng nhằm mô tả chi tiết về luồng hoạt động trong hệ thống, từ khi người dùng bắt đầu một hoạt động cho đến cách hệ thống phản hồi kết quả cuối cùng.

Biểu đồ này giúp nhóm phát triển hiểu rõ hơn các bước nghiệp vụ và tương tác giữa người dùng và hệ thống, đặc biệt trong các ca sử dụng (Use Case) quan trọng. Thông qua biểu đồ này, ta có thể quan sát được toàn bộ tiến trình xử lý, bao gồm các hành động mà người dùng thực hiện, các xử lý nội bộ của hệ thống, điều kiện rẽ nhánh, cũng như các kết thúc có thể xảy ra. Điều này không chỉ hỗ trợ việc phân tích yêu cầu mà còn giúp lập trình viên nắm bắt logic nghiệp vụ để triển khai chính xác hơn.

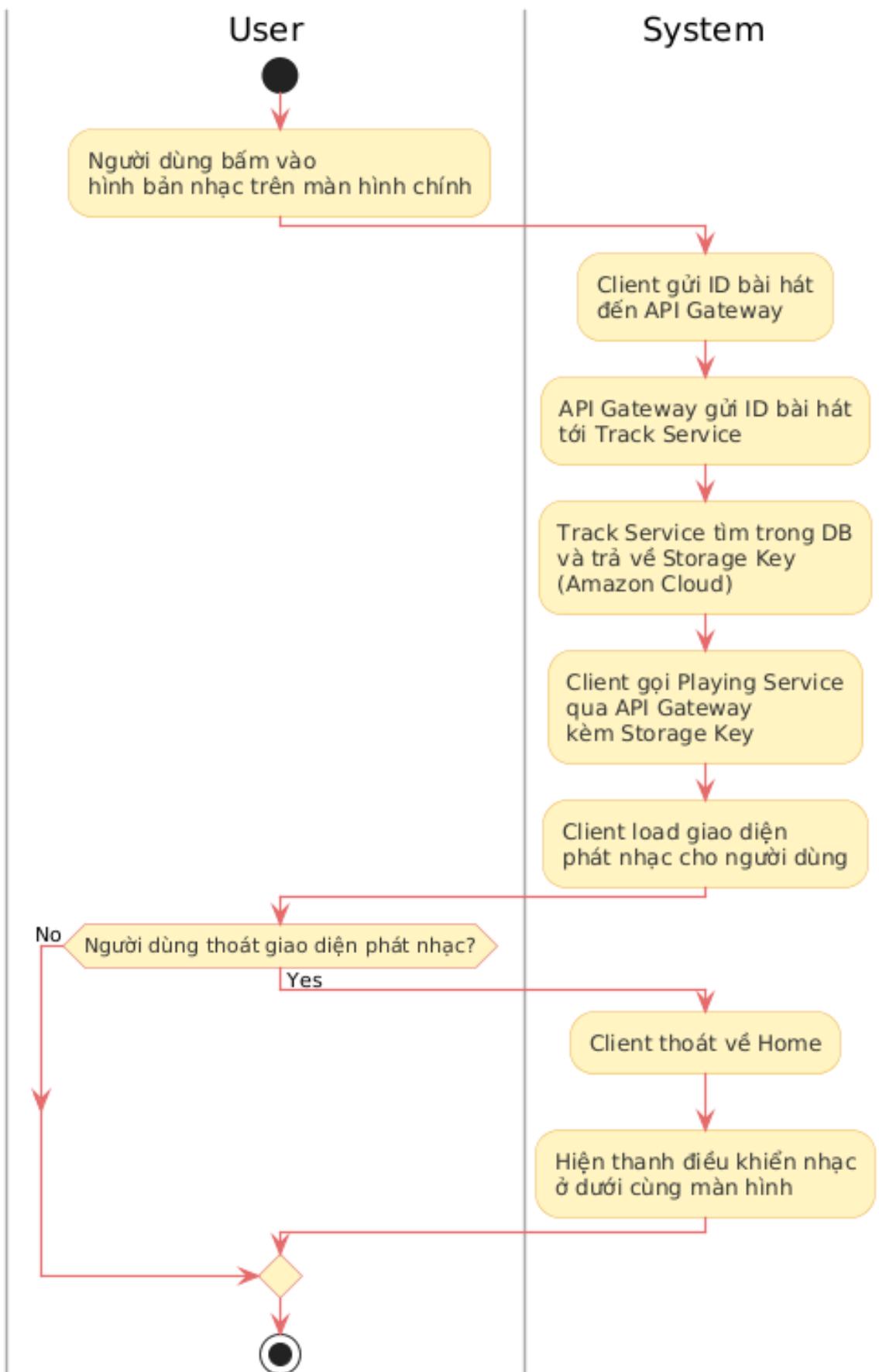
Chú thích: Decision trong Activity Diagram theo chuẩn thì được biểu diễn bằng một hình thoi, tuy nhiên do code bằng PlantUml, mặc định decision là hình lục giác nên decision trong tài liệu này được để là một hình lục giác.

3.4.1. Phát bài hát

Activity Diagram "Phát bài hát" mô tả quy trình mà người dùng thực hiện để phát một bài hát trên hệ thống. Biểu đồ này thể hiện các hoạt động chính như người dùng chọn bài nhạc để phát, hệ thống xử lý yêu cầu phát nhạc, xử lý các trường hợp lỗi và kết thúc bằng việc bài nhạc được phát thành công.

Biểu đồ giúp minh họa rõ các hoạt động giữa người dùng và hệ thống, cho phép nhìn tổng thể các bước, các quyết định quan trọng cũng như tương tác giữa các đối tượng.

Activity Diagram - Playing a Song (UC-PlaySong)

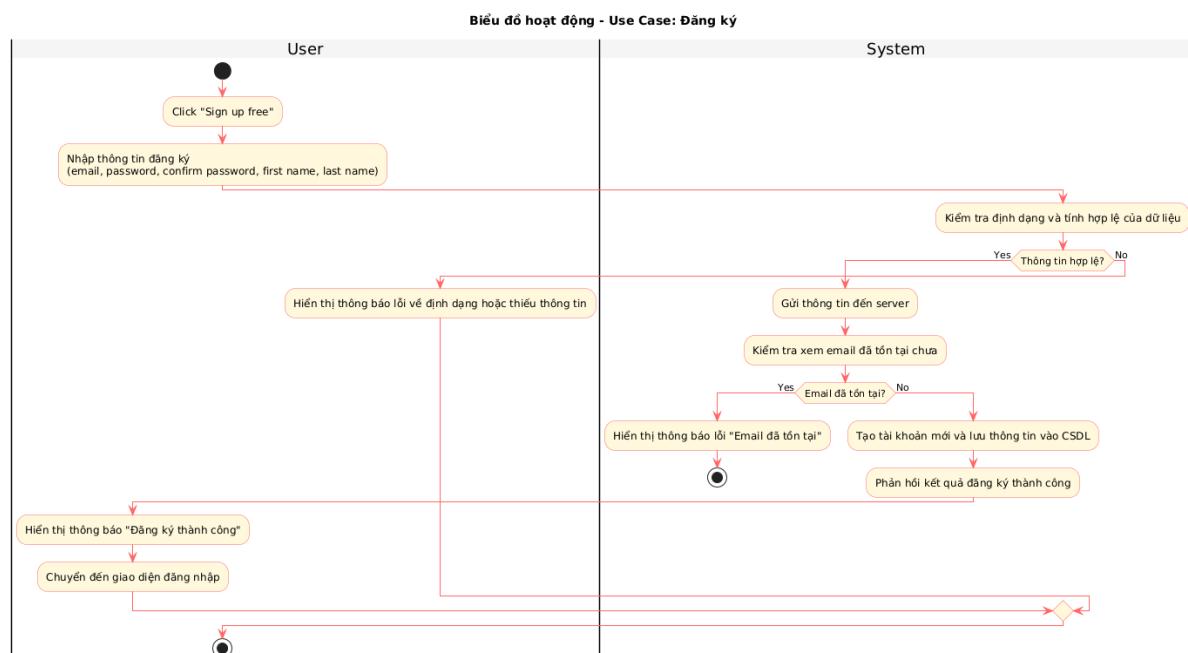


Hình 3.2. Biểu đồ hoạt động chức năng "Phát bài hát"²⁷

3.4.2. Đăng ký

Activity Diagram "Đăng ký" mô tả quy trình mà người dùng thực hiện để tạo tài khoản mới trên hệ thống. Biểu đồ này thể hiện các hoạt động chính như nhập thông tin đăng ký, kiểm tra tính hợp lệ của dữ liệu, xử lý các trường hợp lỗi và kết thúc bằng việc tạo tài khoản thành công.

Biểu đồ giúp minh họa rõ các hoạt động giữa người dùng và hệ thống, cho phép nhìn tổng thể các bước, các quyết định quan trọng cũng như tương tác giữa các đối tượng.

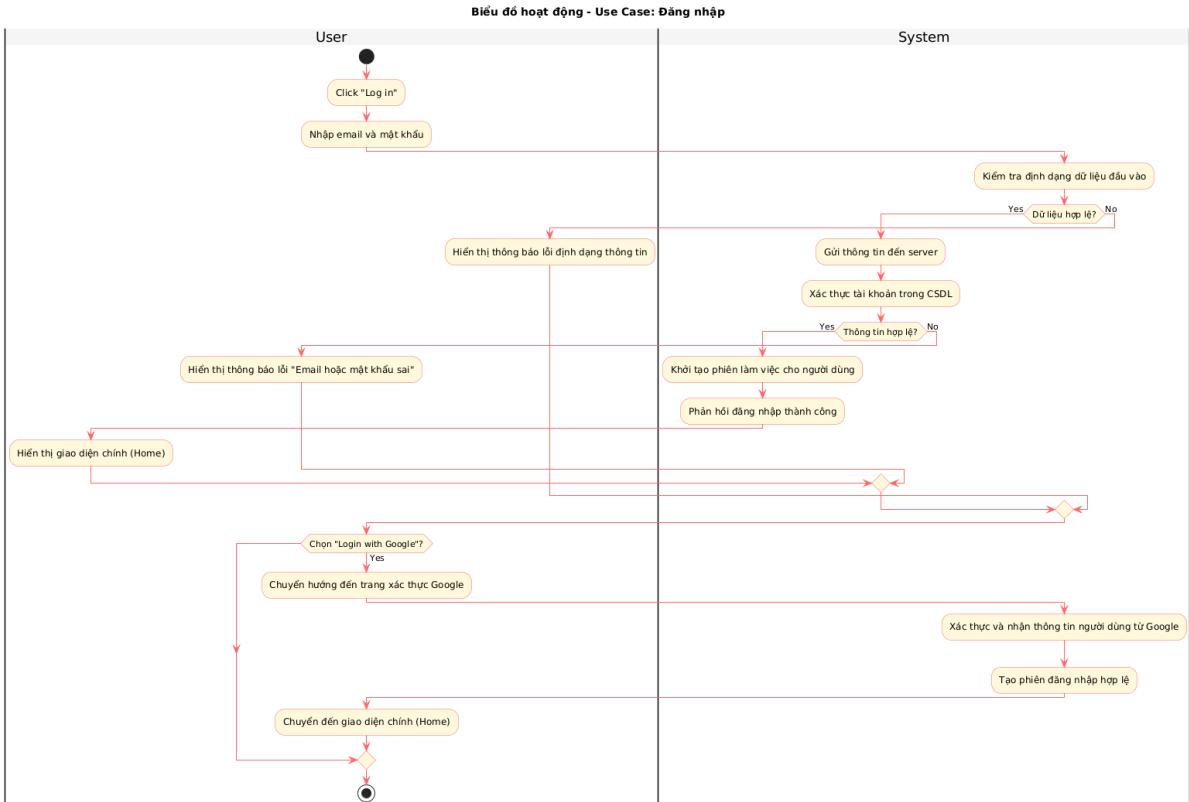


Hình 3.3. Biểu đồ hoạt động chức năng "Đăng ký"

3.4.3. Đăng nhập

Activity Diagram "Đăng nhập" mô tả quy trình mà người dùng thực hiện để đăng nhập vào hệ thống. Biểu đồ này thể hiện các hoạt động chính như nhập thông tin đăng nhập, kiểm tra tính hợp lệ của dữ liệu, xử lý các trường hợp lỗi và kết thúc bằng việc đăng nhập thành công.

Biểu đồ giúp minh họa rõ các hoạt động giữa người dùng và hệ thống, cho phép nhìn tổng thể các bước, các quyết định quan trọng cũng như tương tác giữa các đối tượng.



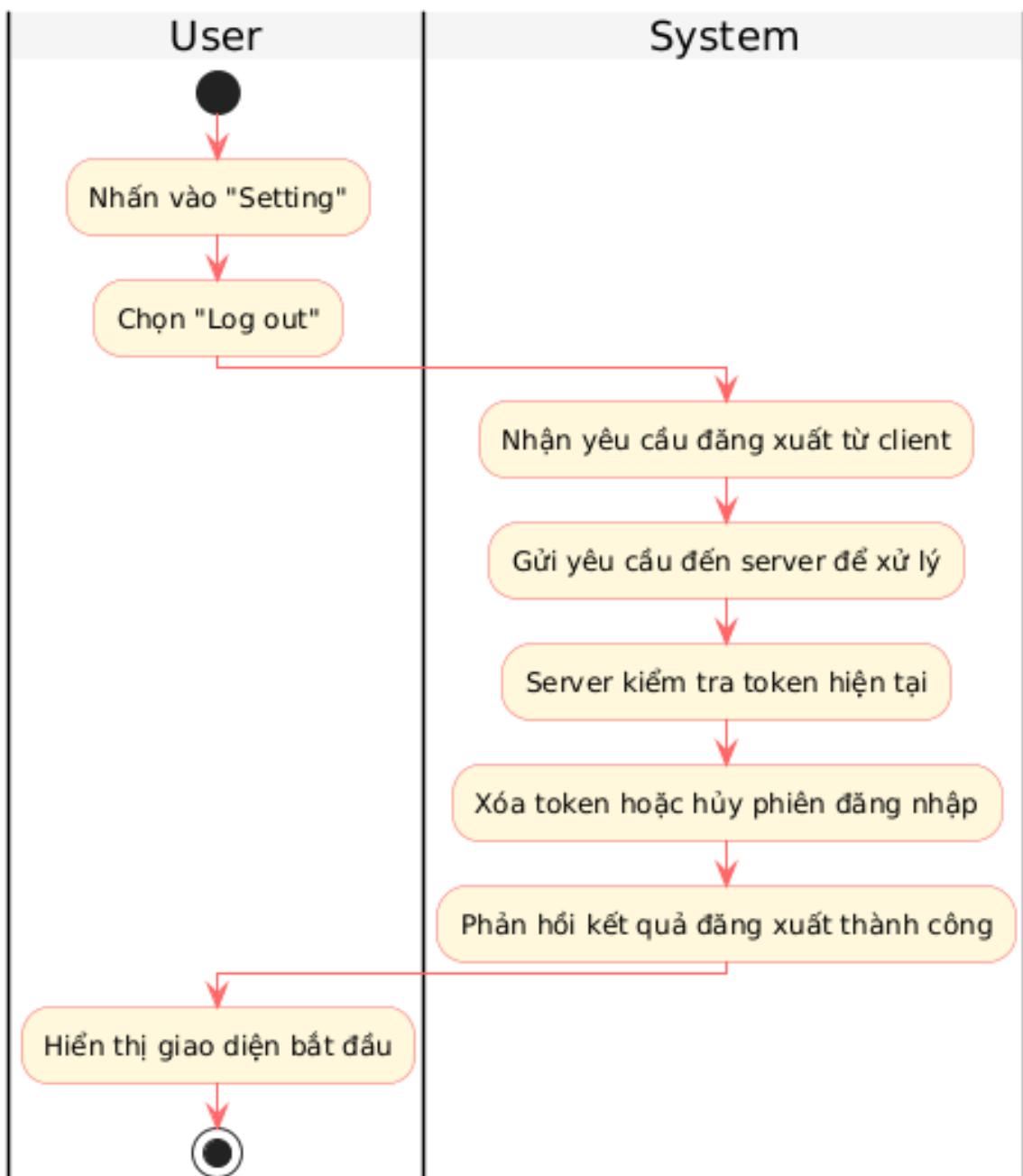
Hình 3.4. Biểu đồ hoạt động chức năng "Đăng nhập"

3.4.4. Đăng xuất

Activity Diagram "Đăng xuất" mô tả quy trình mà người dùng thực hiện để đăng xuất khỏi hệ thống. Biểu đồ này thể hiện các hoạt động chính như tương tác của người dùng với hệ thống tính hợp lệ của dữ liệu, xử lý các trường hợp lỗi và kết thúc bằng việc đăng xuất thành công.

Biểu đồ giúp minh họa rõ các hoạt động giữa người dùng và hệ thống, cho phép nhìn tổng thể các bước, các quyết định quan trọng cũng như tương tác giữa các đối tượng.

Biểu đồ hoạt động - Use Case: Đăng xuất



Hình 3.5. Biểu đồ hoạt động chức năng "Đăng xuất"

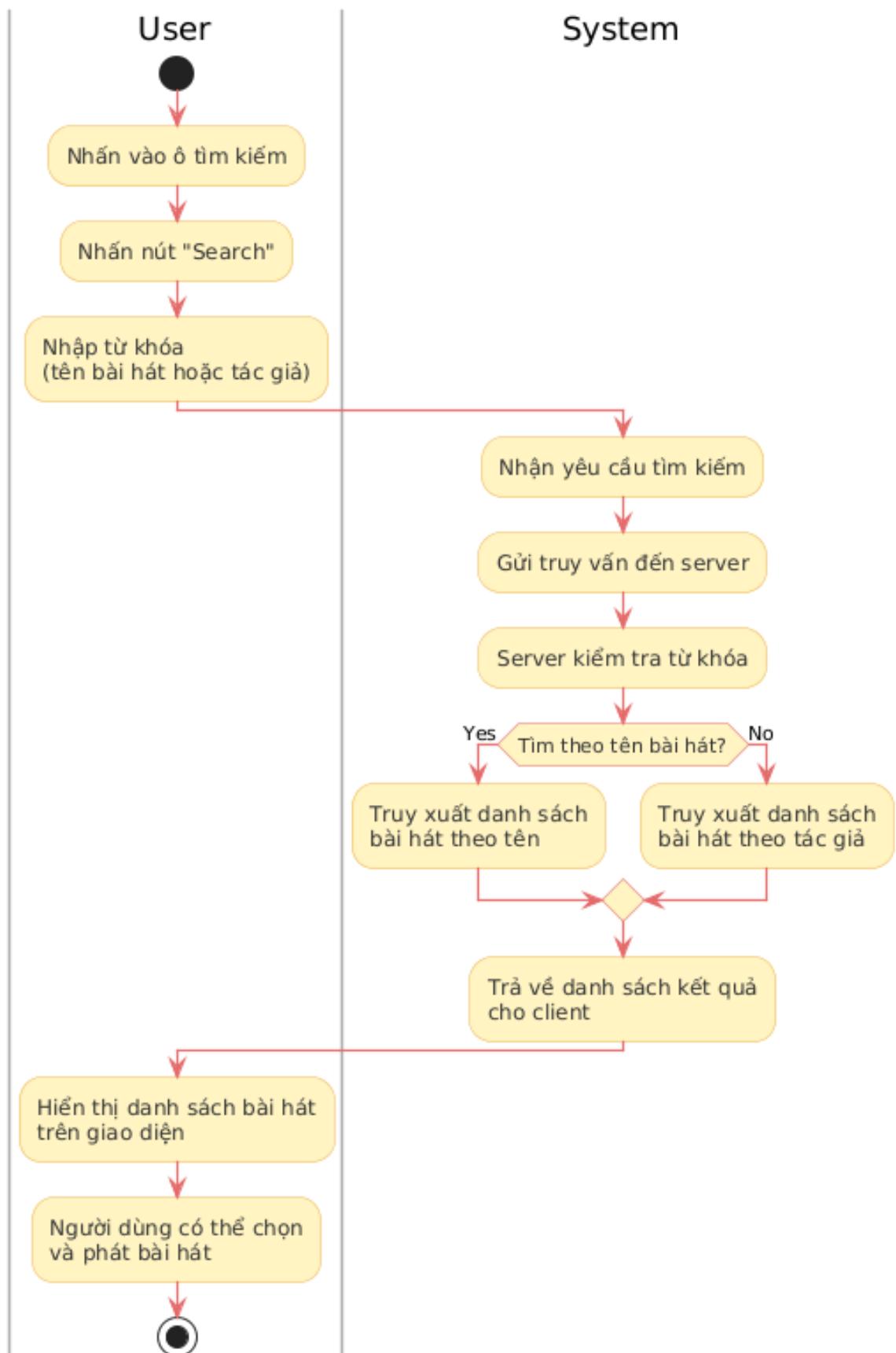
3.4.5. Tìm kiếm bài hát

Activity Diagram "Tìm kiếm bài hát" mô tả quy trình mà người dùng thực hiện để tìm kiếm bài hát muốn nghe trên hệ thống. Biểu đồ này thể hiện các hoạt động chính như cách người dùng thao tác với giao diện, nhập thông tin cần tìm kiếm, hệ thống xử lý dữ liệu, xử lý các trường hợp lỗi và kết thúc bằng việc tìm thấy một loạt danh sách bài

hát thành công.

Biểu đồ giúp minh họa rõ các hoạt động giữa người dùng và hệ thống, cho phép nhìn tổng thể các bước, các quyết định quan trọng cũng như tương tác giữa các đối tượng.

Activity Diagram - Search Song

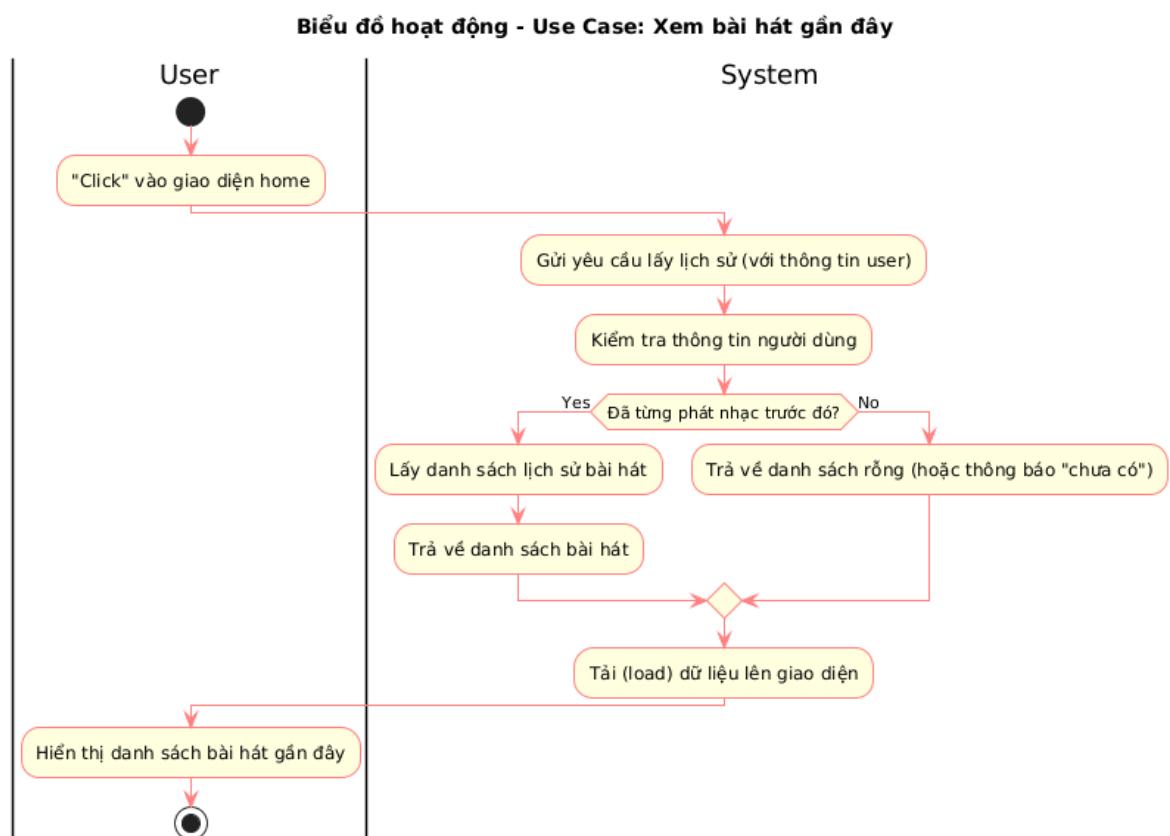


Hình 3.6. Biểu đồ hoạt động chức năng "Tìm kiếm bài hát"

3.4.6. Xem bài hát gần đây

Activity Diagram "Xem bài hát gần đây" mô tả quy trình mà người dùng thực hiện để xem danh sách các bài hát đã nghe trên hệ thống. Biểu đồ này thể hiện các hoạt động chính như cách người dùng thao tác với giao diện, hệ thống xử lý dữ liệu, xử lý các trường hợp lỗi và kết thúc bằng việc hiện lên một loạt danh sách bài hát thành công.

Biểu đồ giúp minh họa rõ các hoạt động giữa người dùng và hệ thống, cho phép nhìn tổng thể các bước, các quyết định quan trọng cũng như tương tác giữa các đối tượng.



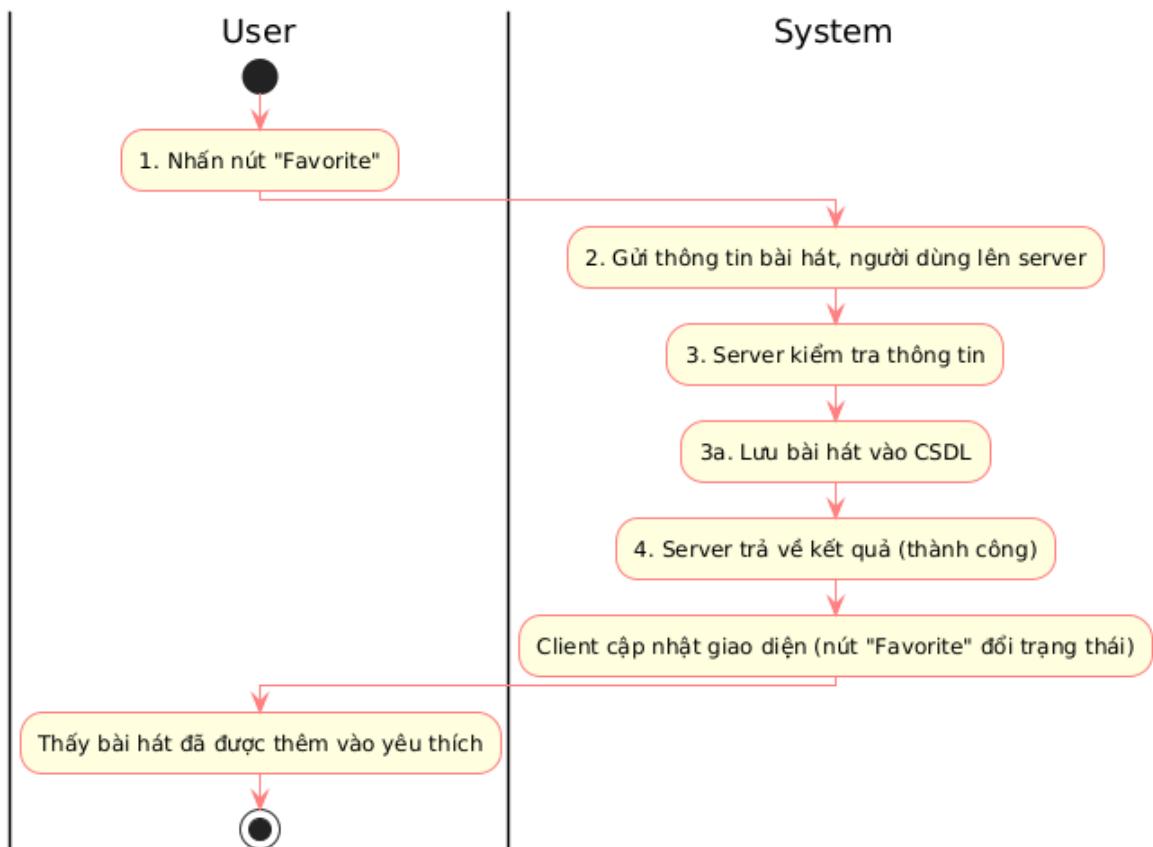
Hình 3.7. Biểu đồ hoạt động chức năng "Xem bài hát gần đây"

3.4.7. Lưu bài hát vào yêu thích

Activity Diagram "Lưu bài hát vào yêu thích" mô tả quy trình mà người dùng thực hiện để lưu bài hát vào danh sách các bài hát yêu thích trên hệ thống. Biểu đồ này thể hiện các hoạt động chính như cách người dùng thao tác với giao diện, hệ thống xử lý dữ liệu và kết thúc bằng việc lưu bài hát vào danh sách yêu thích thành công.

Biểu đồ giúp minh họa rõ các hoạt động giữa người dùng và hệ thống, cho phép nhìn tổng thể các bước, các quyết định quan trọng cũng như tương tác giữa các đối tượng.

Biểu đồ hoạt động - Use Case: Lưu bài hát yêu thích

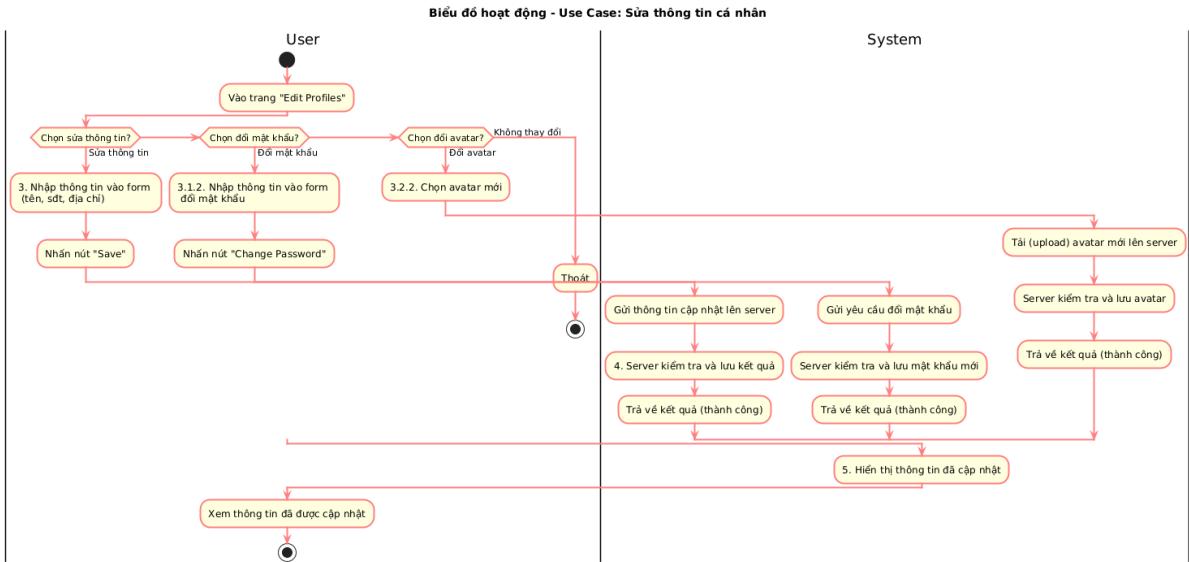


Hình 3.8. Biểu đồ hoạt động chức năng "Lưu bài hát vào yêu thích"

3.4.8. Sửa thông tin cá nhân

Activity Diagram "Sửa thông tin cá nhân" mô tả quy trình mà người dùng thực hiện để cập nhật thông tin cá nhân như firstname, lastname, mật khẩu, avatar,... Biểu đồ này thể hiện các hoạt động chính như cách người dùng thao tác với giao diện, hệ thống xử lý dữ liệu và kết thúc bằng việc thông tin người dùng được cập nhật thành công.

Biểu đồ giúp minh họa rõ các hoạt động giữa người dùng và hệ thống, cho phép nhìn tổng thể các bước, các quyết định quan trọng cũng như tương tác giữa các đối tượng.

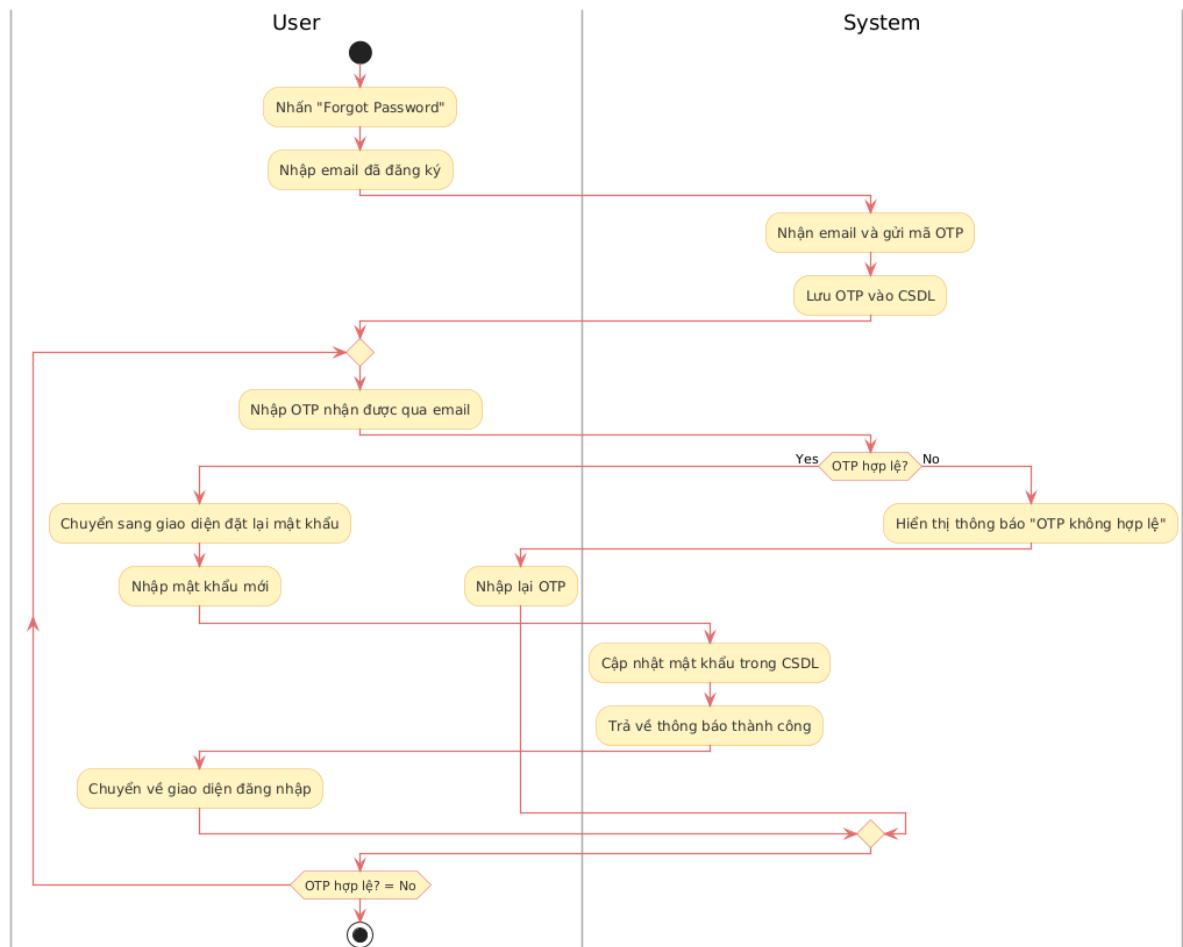


Hình 3.9. Biểu đồ hoạt động chức năng "Sửa thông tin cá nhân"

3.4.9. Quên mật khẩu

Activity Diagram "Quên mật khẩu" mô tả quy trình mà người dùng thực hiện để thay đổi mật khẩu khi không nhớ mật khẩu cũ. Biểu đồ này thể hiện các hoạt động chính như cách người dùng thao tác với giao diện, hệ thống xử lý dữ liệu và kết thúc bằng việc mật khẩu người dùng được cập nhật thành công.

Biểu đồ giúp minh họa rõ các hoạt động giữa người dùng và hệ thống, cho phép nhìn tổng thể các bước, các quyết định quan trọng cũng như tương tác giữa các đối tượng.



Hình 3.10. Biểu đồ hoạt động chức năng "Quên mật khẩu"

CHƯƠNG 4

PHÂN TÍCH KIẾN TRÚC HỆ THỐNG

Chương 4 trình bày về kiến trúc hệ thống, bao gồm yêu cầu thiết kế, thiết kế bậc cao, thiết kế cơ sở dữ liệu, luồng xử lý của hệ thống. Chúng tôi trình bày các yêu cầu thiết kế của hệ thống trong 4.1. Các giải pháp cụ thể sẽ được thể hiện trong kiến trúc tổng quát tại 4.2. Luồng chạy của hệ thống được biểu diễn qua các biểu đồ tuần tự ở 4.4. Phần 4.5 trình bày về thiết kế cơ sở dữ liệu để phù hợp với kiến trúc bậc cao của hệ thống. Kiến trúc của hệ thống khi triển khai trên AWS được mô tả trong phần 4.3. Xuyên suốt chương này, chúng tôi sẽ trình bày các đặc trưng thiết kế của hệ thống tuân theo trình tự nêu ra vấn đề, yêu cầu, giải pháp và thiết kế chi tiết ứng dụng các giải pháp đó vào hệ thống của **InsightTune**.

4.1. Yêu cầu thiết kế hệ thống

Như chúng tôi đã trình bày trong chương 1 về các vấn đề cần giải quyết, trong đó rất nhiều vấn đề sẽ được đề cập và giải quyết trong chương này thông qua các yêu cầu về thiết kế hệ thống.

Các yêu cầu được đề ra nhằm để giải quyết các **vấn đề** sau:

- Độ phức tạp và khả năng mở rộng của các hệ thống.
- Khả năng tái sử dụng các thành phần trong hệ thống.
- Khả năng triển khai đơn giản trên nền tảng đám mây AWS.

Xem xét các vấn đề này, chúng tôi thiết lập các ràng buộc cho hệ thống **InsightTune** như sau:

- (1) Về ngôn ngữ: Java Spring Boot backend, Kotlin frontend.
- (2) Về khả năng mở rộng: mở rộng theo chiều ngang, tách backend và frontend.
- (3) Về hiệu năng: chịu tải lớn.
- (4) Về bảo mật: tách phương thức bên trong và bên ngoài, giao tiếp nội bộ an toàn.
- (5) Về quan sát: ghi lỗi, theo dõi cơ sở dữ liệu và hệ thống.

Chúng tôi không trình bày chi tiết thông số của các yêu cầu trên, chi tiết các thông số và dữ liệu để đo độ đạt của thiết kế hệ thống sẽ được trình bày ở chương 6. Chương này sẽ chỉ tập trung vào tích các yêu cầu, đưa ra giải pháp trên phương diện thiết kế bậc cao của hệ thống.

4.2. Kiến trúc hệ thống

Trong phần này, chúng tôi sẽ trình bày, phân tích thiết kế hệ thống được sử dụng tuân theo các yêu cầu được trình bày trong 4.1.

Để tách biệt front end và back end, hệ thống của chúng tôi được thiết kế theo kiến trúc **Client - Server**. Trong đó Server được thiết kế theo kiến trúc **Microservice** và Client theo **MVVM**. Điều này giải quyết yêu cầu (2) giúp hệ thống trở nên linh hoạt trong việc phát triển và mở rộng, nhà phát triển có thể dễ dàng thêm các chức năng mới bằng việc thiết kế thêm các hệ thống mới.

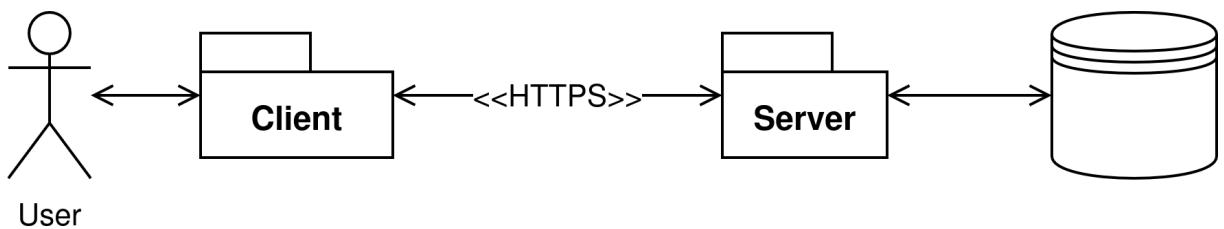
Yêu cầu (1) nhằm giúp mã nguồn trở nên đơn giản bằng cách tận dụng kiến trúc của Spring Boot. Kiến trúc **controller - service - repository** của Spring Boot sẽ được áp dụng cho từng service của hệ thống.

Việc đảm bảo hiệu năng (3) được thực hiện thông qua thiết kế API và các phương

thức cân bằng tải. Vai trò và vị trí của chúng sẽ được biểu diễn trong phần server dưới đây.

Các phương thức giao tiếp giữa các service và API cũng sẽ được nhắc tới và trình bày chi tiết nhằm đảm bảo yêu cầu (4).

Bằng việc tận dụng dịch vụ sẵn có của AWS và công cụ của Spring Boot, chúng tôi thiết kế công cụ ghi lỗi và theo dõi cơ sở dữ liệu cho yêu cầu (5).



Hình 4.1. Kiến trúc client-server của hệ thống

4.2.1. Client

Client là một thành phần chính trong kiến trúc hệ thống, nằm ở thiết bị di động của người dùng. Nhiệm vụ của Client là cung cấp giao diện để người dùng tương tác với hệ thống của ứng dụng, ghi nhận dữ liệu đầu vào và hiển thị dữ liệu của người dùng khi được yêu cầu. Không chỉ vậy, Client còn chịu trách nhiệm quản lý các trạng thái giao diện, xử lý logic và dữ liệu, giao tiếp hiệu quả với Server để đem lại trải nghiệm tốt nhất cho người dùng.

Trong các mô hình phát triển ứng dụng truyền thống, Client thường được xây dựng theo kiến trúc **MVC (Model - View - Controller)** hoặc không có kiến trúc rõ ràng. Tuy nhiên, các hệ thống này thường bị pha trộn các thành phần với nhau, ví dụ như hiển thị giao diện và xử lý logic, xử lý dữ liệu trong cùng một lớp. Điều này dẫn đến các lớp View thường trở nên quá lớn, phức tạp, phải thực hiện quá nhiều nhiệm vụ khác nhau như hiển thị giao diện, xử lý logic, gọi API, xử lý dữ liệu,..., dẫn đến các vấn đề:

- Khó khăn trong việc bảo trì và nâng cấp hệ thống
- Khó khăn trong việc mở rộng hệ thống khi cần thêm các tính năng mới.
- Khó khăn trong việc kiểm thử do các lớp quá lớn và phức tạp.
- Khó khăn trong việc tái sử dụng mã nguồn do các thành phần không được tách biệt rõ ràng.

Để giải quyết các vấn đề nêu trên, chúng tôi lựa chọn xây dựng kiến trúc Client sử

dụng mô hình **MVVM (Model - View - ViewModel)** kết hợp với **Repository Pattern** nhằm tách biệt các thành phần trong hệ thống, giúp việc phát triển, bảo trì và mở rộng hệ thống trở nên dễ dàng hơn.

Trong kiến trúc này, vai trò của các thành phần được mô tả như sau:

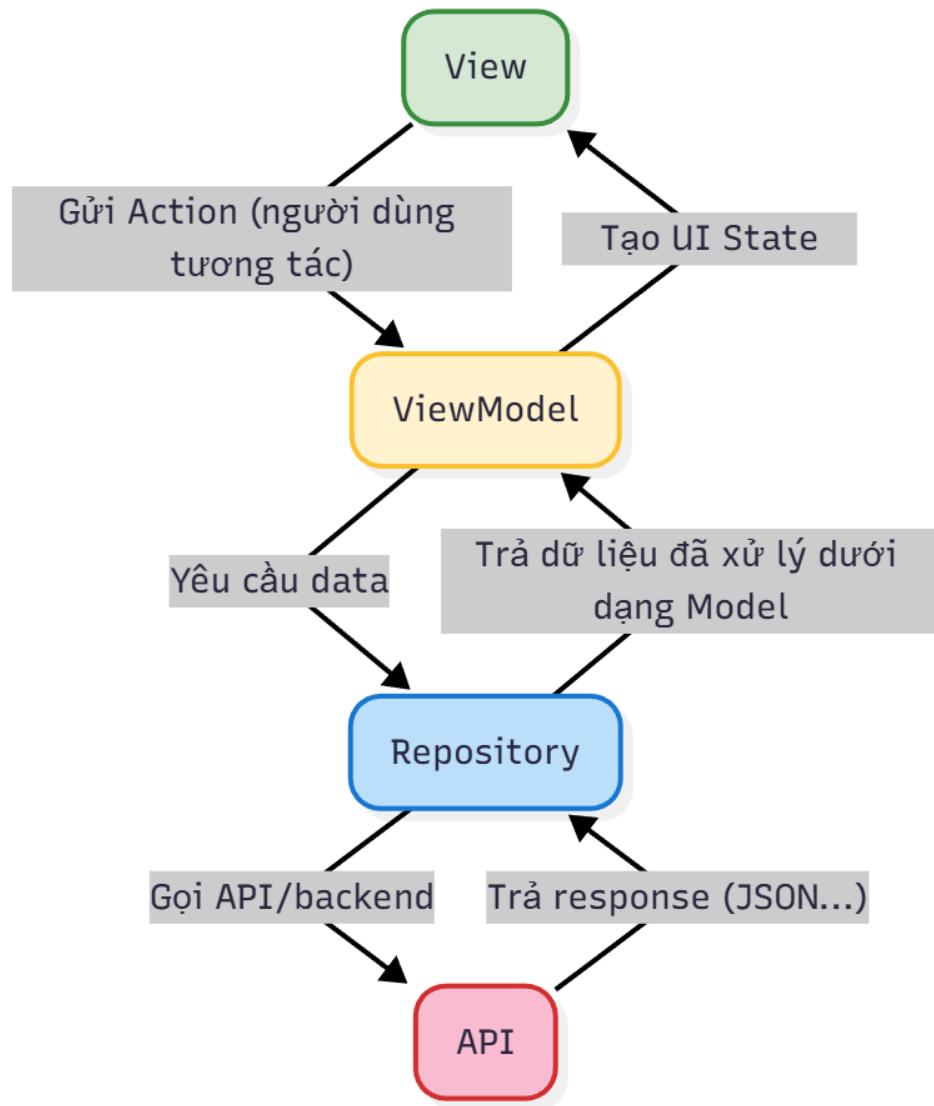
- **Model:** Chứa các lớp đối tượng biểu diễn dữ liệu trong hệ thống.
- **View:** Hiển thị giao diện và nhận các tương tác từ người dùng. View chỉ tập trung vào hiển thị giao diện và không chứa logic xử lý dữ liệu. Trong kiến trúc này, View được xây dựng bằng Jetpack Compose, một thư viện UI hiện đại của Android giúp xây dựng giao diện người dùng một cách nhanh chóng và hiệu quả.
- **ViewModel:** ViewModel nhận yêu cầu, tương tác của người dùng từ View và gửi yêu cầu, nhận dữ liệu từ Repository. ViewModel chịu trách nhiệm xử lý logic, xử lý dữ liệu, quản lý trạng thái và cung cấp dữ liệu cho View.
- **Repository:** Là lớp trung gian giữa ViewModel và API Server. Repository chịu trách nhiệm thực hiện các thao tác gọi API, đồng thời có thể nhận dữ liệu từ Server dưới dạng JSON và xử lý dữ liệu trước khi trả về cho ViewModel.

Khi người dùng tương tác với giao diện-**View** (ví dụ ấn nút đăng nhập) thì **View** sẽ gửi yêu cầu đến **ViewModel**. **ViewModel** sẽ xử lý yêu cầu này, gọi đến các phương thức trong **Repository**. **Repository** gọi API đến Server để lấy hoặc nhận dữ liệu. API sẽ trả về dữ liệu cho **Repository** dưới dạng JSON. **Repository** nhận, có thể xử lý dữ liệu thành các đối tượng **Model** và trả về cho **ViewModel**. **ViewModel** tiếp tục xử lý dữ liệu nếu cần và cập nhật lại trạng thái của giao diện mà **View** đang quan sát, sau đó **View** sẽ tự động cập nhật giao diện dựa trên trạng thái mới từ **ViewModel**.

Lợi ích của việc sử dụng kiến trúc **MVVM** kết hợp với **Repository Pattern** bao gồm:

- Tách biệt rõ ràng các thành phần trong hệ thống, phân chia trách nhiệm rõ ràng giữa các lớp, giúp loại bỏ sự phụ thuộc chồng chéo giữa các thành phần.
- Mã nguồn trở nên linh hoạt và dễ dàng thay đổi.
- Tăng khả năng tái sử dụng mã nguồn, vì các thành phần được tách biệt rõ ràng và có thể sử dụng lại trong các phần khác của hệ thống.
- Giúp kiểm thử dễ dàng hơn vì các thành phần nhỏ hơn và có trách nhiệm rõ ràng hơn.
- Dễ dàng quản lý trạng thái và dữ liệu trong ứng dụng.

Việc áp dụng kiến trúc **MVVM** kết hợp với **Repository Pattern** giúp giải quyết các vấn đề trong phát triển ứng dụng truyền thống, đồng thời nâng cao chất lượng và hiệu quả của quá trình phát triển phần mềm, xây dựng nền tảng Client vững chắc, linh hoạt để dễ dàng bảo trì và tiếp tục mở rộng trong tương lai.



Hình 4.2. Kiến trúc MVVM với Repository Pattern trên Client

4.2.2. Server

Phần trước chúng tôi đã diễn giải và phân tích Client, ở phần này chúng tôi xin nói về thành phần quan trọng còn lại trong kiến trúc Client - Server, Server.

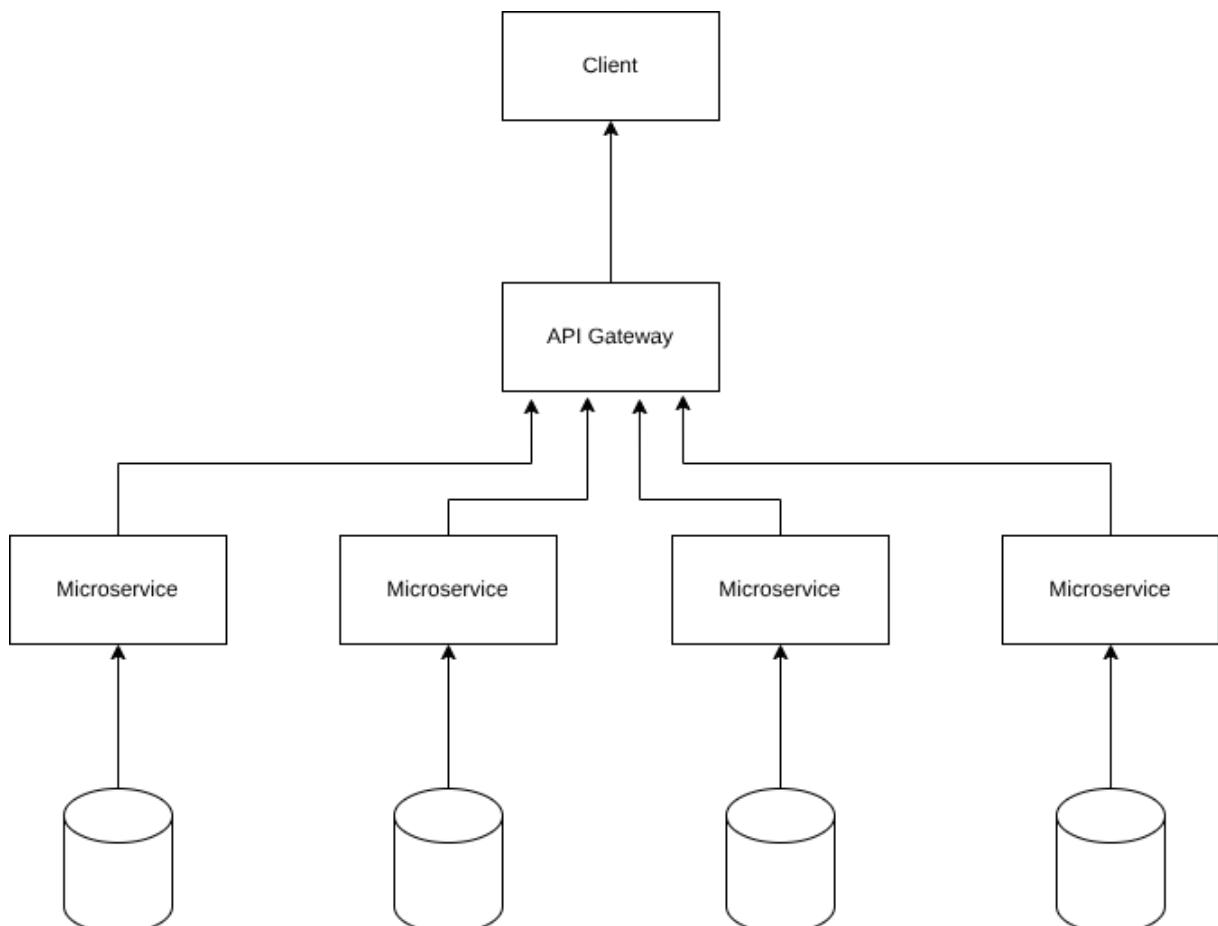
Trong kiến trúc Client - Server, Server có vai trò cốt lõi như sau:

- Là cửa ngõ tiếp nhận và phản hồi yêu cầu
- Thực thi nghiệp vụ (Business Logic)

- Truy cập cơ sở dữ liệu và bảo toàn nhất quán dữ liệu
- Quản lý phiên và bảo mật xác thực danh tính
- Kiểm soát lưu lượng
- Ghi lại (logging) hoạt động, trạng thái của hệ thống

So với Client yêu cầu dung lượng nhẹ, khả năng dễ triển khai tương thích với nhiều loại sản phẩm. Server cần có khả năng mở rộng cao khi hệ thống sẽ ngày càng phức tạp trong tương lai. Với các mô hình monolithic truyền thống, việc triển khai và xây dựng sẽ nhanh chóng dễ dàng giai đoạn đầu. Tuy nhiên, càng về sau, khi hệ thống trở nên đồ sộ và bắt đầu cần mở rộng, monolithic lại cho thấy những hạn chế của mình trong khả năng mở rộng và tái sử dụng code.

Do đó, để đáp ứng yêu cầu (2) đưa ra về khả năng mở rộng, chúng tôi đã lựa chọn kiến trúc microservice. Một kiến trúc chia hệ thống thành nhiều dịch vụ nhỏ, độc lập triển khai. Kiến trúc chung của microservice có thể minh họa qua hình 4.3 sau đây.

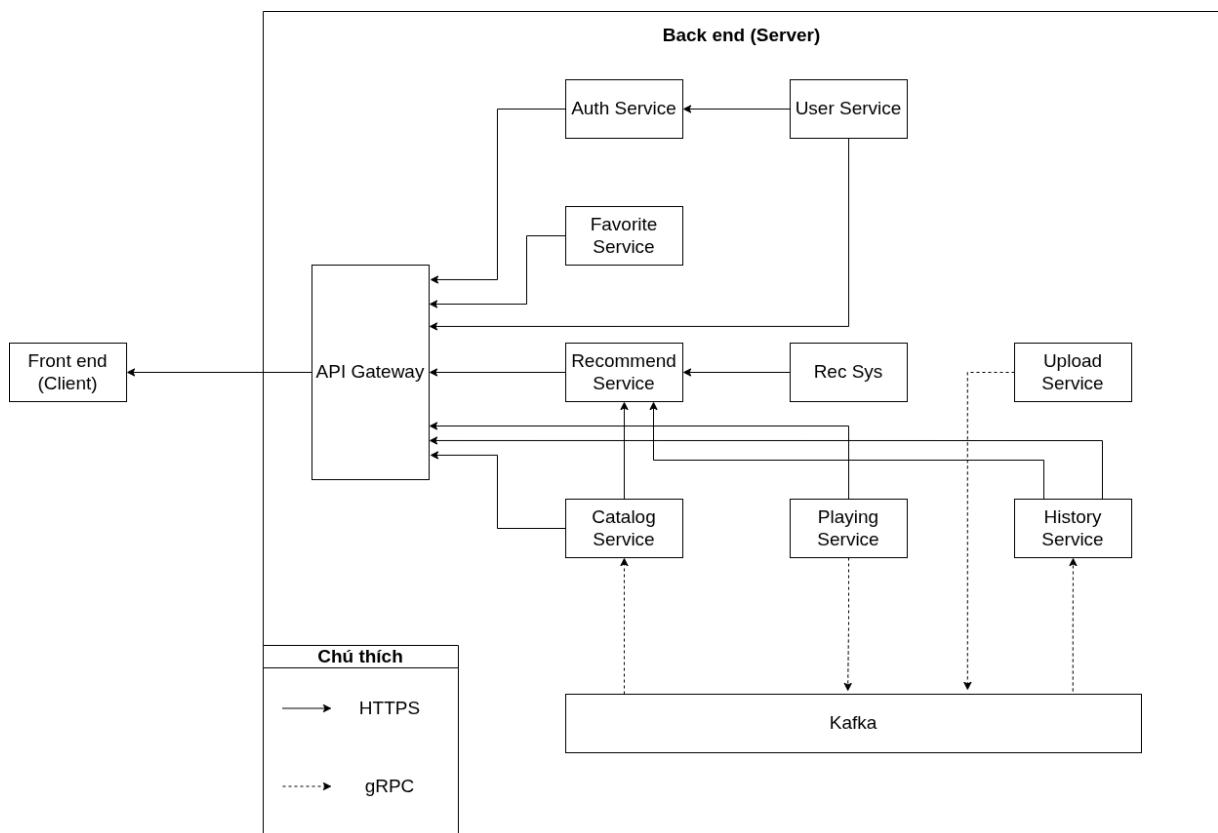


Hình 4.3. Kiến trúc microservice chung

Ở trong kiến trúc microservice điểm đặc biệt nằm ở các dịch vụ nhỏ của nó. Mỗi dịch vụ (service) phải thỏa mãn các yêu cầu cố định, tạo nên sự linh hoạt và khả năng mở rộng cao đặc trưng của kiến trúc này. Các yêu cầu này có thể tóm gọn lại như sau:

- Có miền nghiệp vụ rõ ràng (bounded context).
- Tự quản lý dữ liệu riêng (database-per-service).
- Giao tiếp qua API (REST, gRPC,...) hoặc broker (kafka, rabbitMQ).
- Có vòng đời phát triển và triển khai tách biệt với các dịch vụ khác.

Sau khi cân nhắc các ca sử dụng như trong chương 3 và những công cụ hiện có. Chúng tôi đã ứng dụng kiến trúc microservice này vào hệ thống **InsightTune** mà vẫn giữ nguyên tắc thủ các yêu cầu khi triển khai từng service. Chúng tôi xin trình bày kiến trúc bậc cao của server như sơ đồ 4.4 dưới đây.



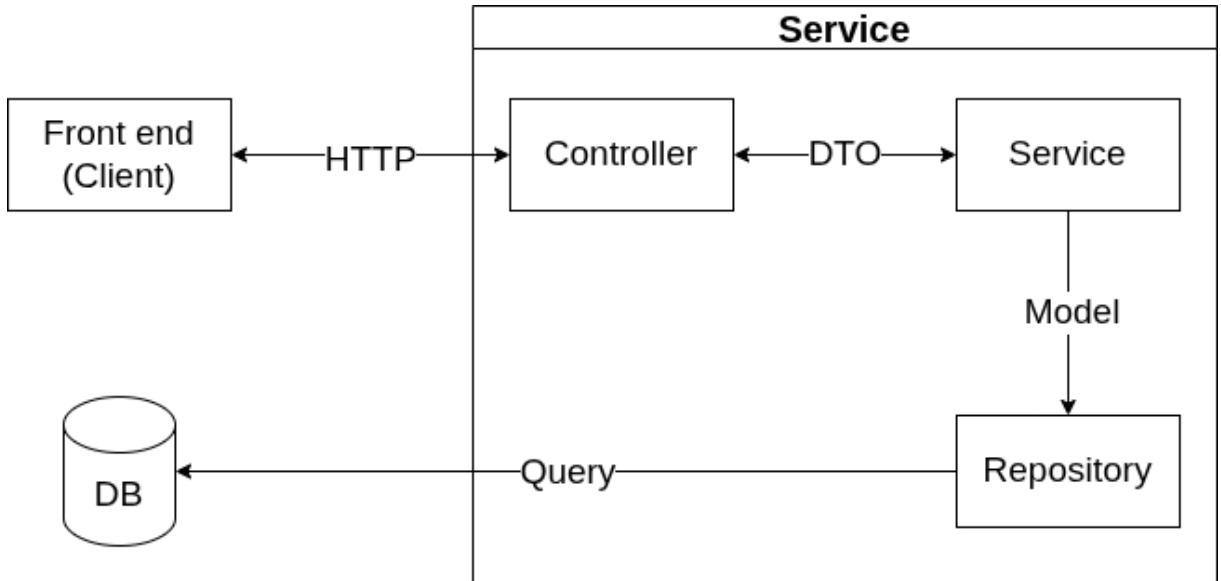
Hình 4.4. Kiến trúc server bậc cao

Sơ đồ này biểu thị các dịch vụ có trong hệ thống và phương thức giao tiếp của chúng. Mỗi dịch vụ đều có chức năng riêng cũng như dữ liệu riêng của mình tách biệt cùng các dịch vụ khác trong server. Chúng được cố định phương thức giao tiếp truyền tin đồng bộ qua REST và bất đồng bộ thông qua Kafka. Những dịch vụ trong server của

hệ thống **InsightTune** gồm có:

- AuthService: phụ trách xác thực, quản lý phiên, đảm nhiệm cơ chế bảo mật chính của server
- UserService: lưu thông tin riêng của người dùng
- CatalogService: quản lý thông tin bài hát, ca sĩ, album ca nhạc
- FavoriteService: quản lý bài hát yêu thích của người dùng
- RecSys: hệ thống gợi ý, nhận lịch sử người dùng và trả về danh sách bài hát đề xuất, quản lý đặc trưng của bài hát
- RecommendService: tổng hợp lịch sử, danh sách ứng cử viên gửi cho RecSys để nhận đề xuất
- HistoryService: quản lý lịch sử phát nhạc của người dùng
- PlayingService: tương tác với kho lưu trữ nhạc để lấy ảnh và file phát nhạc được streaming trực tiếp
- UploadService: dịch vụ riêng không tương tác với bên ngoài để tải bài hát lên kho lưu trữ
- API Gateway: dịch vụ quản lý các tuyến đường truy cập từ ngoài mạng vào nội bộ mạng của server, có khả năng cân bằng tải khi nhiều người dùng gọi tới

Mỗi dịch vụ này khi thiết kế luôn tuân theo các yêu cầu đặt ra cho một dịch vụ trong kiến trúc microservice. Với việc sử dụng SpringBoot Framework cho mã nguồn server, chúng tôi đã vận dụng luồng xử lý, kiến trúc bậc cao của Spring cho mỗi dịch vụ. Kiến trúc 3 lớp **controller - service - repository** của Spring có thể được biểu diễn như trong hình 4.5.



Hình 4.5. Kiến trúc 3 lớp của Spring khi xây dựng Microservice

Trong kiến trúc 3 lớp này, mỗi lớp có vai trò và nhiệm vụ riêng biệt. Cụ thể:

- Controller: Nhận yêu cầu HTTP và trả lời lại frontend
- Service: phục trách nghiệp vụ, bao gồm chuyển đổi và xử lý giữa data transfers object (đối tượng chuyển dữ liệu, tương tác tại controller) và domain model (mô hình miền, phục vụ giao tiếp với database)
- Repository: Quản lý tương tác với cơ sở dữ liệu, thực hiện các truy phàn

Tuân thủ theo thiết kế này, chúng tôi đã tạo ra được một thiết kế bậc cao microservice hoàn chỉnh với các dịch vụ con được mô hình hóa tốt, riêng biệt, nghiệp vụ riêng rõ ràng, có bộ code tách biệt không trùng lặp nhau và có khả năng tái sử dụng cao cho Server của hệ thống **InsightTune**. Tuy nhiên để triển khai trên được một nền tảng đám mây như Amazon Web Service, chúng tôi thấy rằng cần phải có một sơ đồ rõ ràng và kế hoạch cũng như các công cụ được liệt kê cẩn thận để các dịch vụ nói riêng cũng như toàn bộ hệ thống **InsightTune** nói chung hoạt động hiệu quả. Điều này sẽ được chúng tôi trình bày kỹ hơn tại phần 4.3.

4.3. Kiến trúc triển khai

Khác với kiến trúc monolithic dễ dàng triển khai khi toàn bộ hệ thống được đóng gói vào một gói duy nhất, microservice lại yêu cầu nhiều công đoạn và quy trình phức tạp hơn. Kiến trúc microservice gồm nhiều dịch vụ con với các phương thức giao tiếp và truyền tải thông tin giữa các hệ thống khiến chi phí và công sức để triển khai hệ thống

một cách hiệu quả sẽ cao hơn. Điều này đặt ra yêu cầu cần có một bản mẫu thiết kế kiến trúc với chi tiết các công cụ và dịch vụ cần sử dụng khi triển khai.

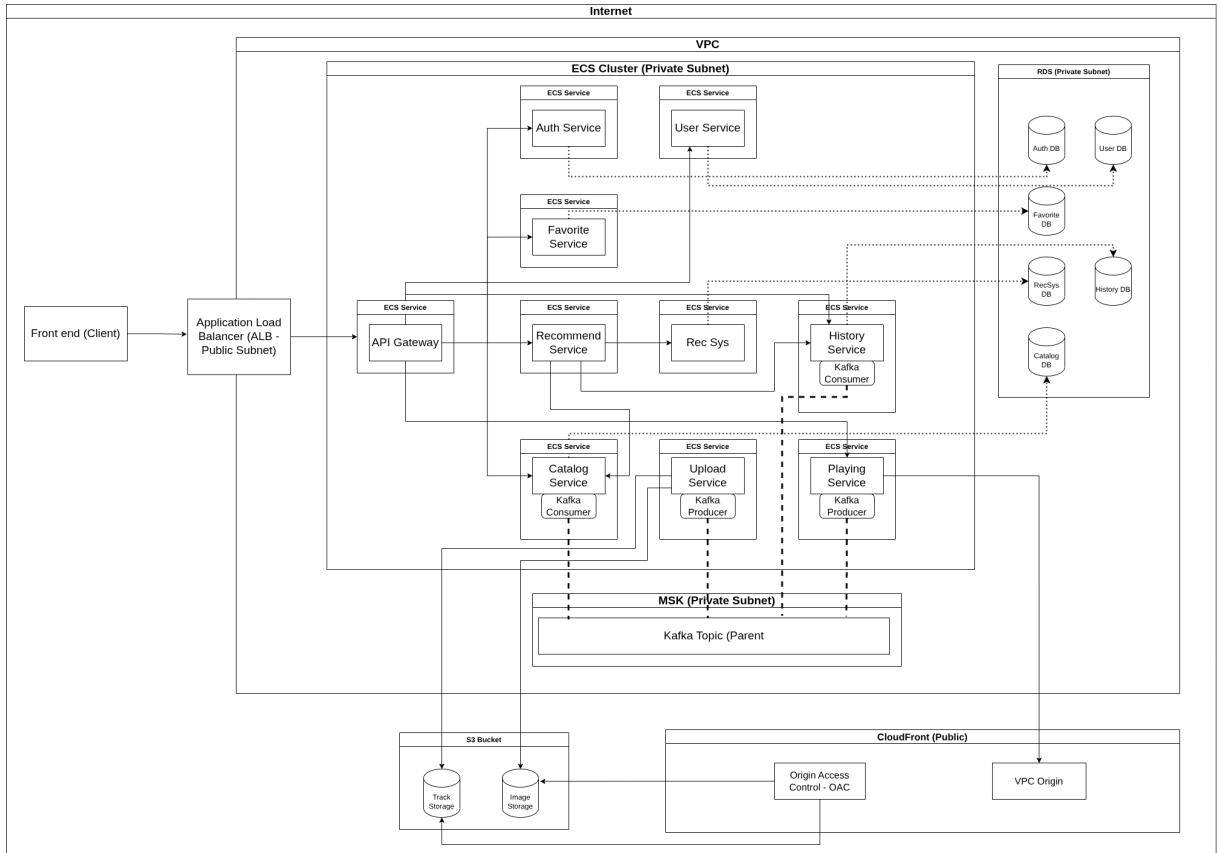
Từ đó, để triển khai hệ thống **InsightTune** tốt, chúng tôi đã cân nhắc các công cụ và dịch vụ sẵn có của Amazon Web Service và thiết kế nên kiến trúc server khi triển khai trên AWS như hình 4.6.

Với dung lượng lưu trữ yêu cầu cao cho file bài hát và ảnh của những bản nhạc, tạo một cơ sở dữ liệu dữ liệu để lưu trữ và sử dụng khi triển khai là không thực tế. Do đó, việc triển khai một bộ lưu trữ đối tượng (object storage) như S3 Bucket là bắt buộc đối với hệ thống yêu cầu dung lượng lưu trữ cho một file lớn như **InsightTune**. Bằng cách này, hệ thống có khả năng triển khai hiệu quả với chi phí thấp hơn so với chỉ sử dụng cơ sở dữ liệu. Giải được vấn đề lưu trữ file bài hát ảnh hưởng trực tiếp tới khả năng mở rộng trong yêu cầu (2), chúng tôi tiếp tục hoàn thành hai yêu cầu (4), (3).

Để thực hiện yêu cầu (4) về bảo mật, chúng tôi tạo ra một mạng đám mây riêng tư ảo (virtual private cloud). VPC bao gồm các mạng con riêng tư (private subnet) phục vụ cho: ECS Cluster (trình quản lý tài nguyên), RDS (triển khai của cơ sở dữ liệu quan hệ), MSK (triển khai của kafka). ECS Cluster quản lý nhiều dịch vụ khác nhau được tạo bởi ECS, mỗi dịch vụ này đều được định nghĩa bởi định nghĩa tác vụ (task definition) và cấu hình bằng một máy ảo EC2. Tất cả các phương thức giao tiếp đều thực hiện nội bộ tách biệt với mạng công cộng (public internet). Đối với giao tiếp với front end, server sẽ tập trung định tuyến một cửa API sau đó kết nối với một cân bằng tải định nghĩa trên một mạng con public của VPC. Application Load Balancer định tuyến đường đi cho các kết nối từ bên ngoài VPC vào bên trong server và cân bằng tải cho những yêu cầu này giúp tăng tính ổn định và hiệu suất của hệ thống.

Việc triển khai ALB đã một phần giải quyết vấn đề (3). Tuy nhiên, để một trình nghe nhạc hoạt động hiệu quả hơn, chúng tôi đã sử dụng CloudFront, một dịch vụ mạng vận chuyển nội dung (content delivery network) của AWS với cơ chế hoạt động là: phân phối, lưu trữ dữ liệu thông qua các điểm truy cập mạng trên toàn thế giới (được gọi là edge location). Điều này tối ưu hóa khả năng và hiệu suất streaming của trình nghe nhạc giúp tăng cao trải nghiệm người dùng. Để giao tiếp với S3 Bucket và VPC đang được bảo mật hoàn toàn được thực thi hai phương thức lần lượt là Origin Access Controll (OAC) và VPC Origin của CloudFront. Đơn giản là hai phương thức này sẽ tạo ra key cá biệt trong CloudFront sau đó, thông qua việc cấp quyền trong S3Bucket và VPC, CloudFront

sẽ có khả năng tương tự như một người dùng được cấp quyền truy cập mạng riêng.



Hình 4.6. Kiến trúc server khi triển khai trên AWS

Biểu đồ 4.6 chỉ là kiến trúc thượng tầng không thể hiện được hết những công cụ và dịch vụ được sử dụng khi triển khai. Để người đọc rõ hơn về triển khai hạ tầng chúng tôi xin liệt kê những dịch vụ được sử dụng để thuận tiện cho người đọc tìm hiểu, tra cứu thêm.

Những dịch vụ chúng tôi sử dụng tóm gọn như sau:

- CloudFormation: công cụ triển khai hạ tầng bằng code, sử dụng một khuôn mẫu để tạo ngăn xếp chứa tài nguyên và tự động xử lý phụ thuộc
- EC2: một máy ảo, có khả năng tùy chỉnh bộ xử lý và dung lượng bộ nhớ đệm, dùng để triển khai container của docker.
- ECS: kết hợp khi tạo FargateService trên AWS giúp tự động quản lý máy ảo, đảm bảo triển khai ứng dụng, quản lý tài nguyên, tiết kiệm chi phí hiệu quả trên nhiều máy (điều này đặc biệt có ý nghĩa với hệ thống có kiến trúc gồm nhiều dịch vụ tách biệt đồng nghĩa nhiều container khi triển khai như microservice)
- Application Load Balancer (ALB): trình định tuyến theo đường đi cho HTTP/HTTPS,

có khả năng tự động cân bằng tải

- IAM: Quản lý quyền truy cập trong nội bộ mạng riêng tư
- Kafka (MSK): Triển khai của kafka - một môi giới sự kiện dùng vận chuyển thông tin các sự kiện bất đồng bộ giữa các dịch vụ
- CloudWatch Logs: Sử dụng cho ghi nhận sự kiện (logging), phục vụ cho việc sửa lỗi cũng như giải quyết yêu cầu (5) cho hệ thống.
- RDS: Triển khai cơ sở dữ liệu quan hệ
- CloudMap: kết hợp với Route 53 để tạo hệ miền, mỗi dịch vụ tự đăng ký vào CloudMap giúp các dịch vụ tự khám phá trong nội bộ
- S3: Lưu trữ đối tượng (Object storage) phục vụ việc lưu ảnh và file mp3 của bài hát.
- Secret Manager: Quản lý các chìa khóa bí mật một cách hiệu quả, tự sinh và tự gán cho các dịch vụ cần
- CloudFront: trình phân phối nội dung phân tán thông qua các điểm truy cập mạng (edge location)

Trên đây là những dịch vụ được sử dụng làm nền tảng trong kiến trúc bậc cao của hệ thống khi triển khai trên hạ tầng AWS. Việc trình bày về kiến trúc thương tầng làm rõ về việc tương tác của các dịch vụ trong hệ thống ở mức bậc cao. Để hiểu rõ về luồng xử lý của hệ thống khi thực hiện các tác vụ cụ thể, chúng tôi trình bày về luồng xử lý của hệ thống thông qua các biểu đồ tuần tự trong phần 4.4 tiết theo.

4.4. Biểu đồ tuần tự

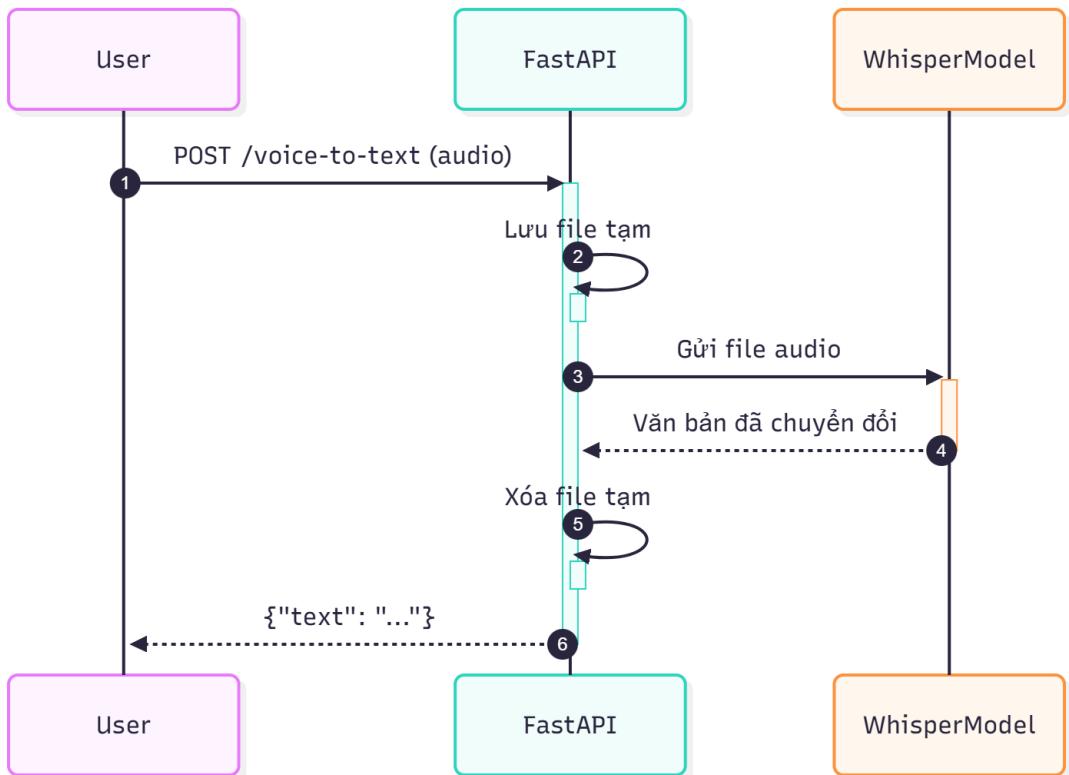
4.4.1. Chatbot

Quá trình hoạt động của chatbot được mô tả như sau. Khi người dùng gửi câu hỏi thì sẽ có 2 loại dữ liệu nhận vào là văn bản và âm thanh.

Đối với dữ liệu là âm thanh, hệ thống sẽ chuyển đổi âm thanh thành văn bản và trả ngược về cho ứng dụng. Sau đó, ứng dụng sẽ trả về cho ứng dụng là văn bản đã được chuyển đổi từ giọng nói. Sau đó, ứng dụng sẽ gọi API của chatbot với đầu vào là dạng văn bản.

Hình 4.7 mô tả quá trình này

Còn đối với dữ liệu là văn bản, người dùng sẽ gọi trực tiếp API của chatbot với đầu vào là văn bản. Và sau khi nhận được yêu cầu từ ứng dụng, chatbot sẽ lấy dữ liệu cũ (nếu có) theo luồng được định danh đối với mỗi chuỗi đầu vào. Nếu chưa có dữ liệu cũ, chatbot sẽ tạo một luồng mới để lưu trữ dữ liệu dùng cho sau này. Sau đó, chatbot sẽ



Hình 4.7. Biểu đồ tuần tự cho quá trình chuyển đổi giọng nói thành văn bản

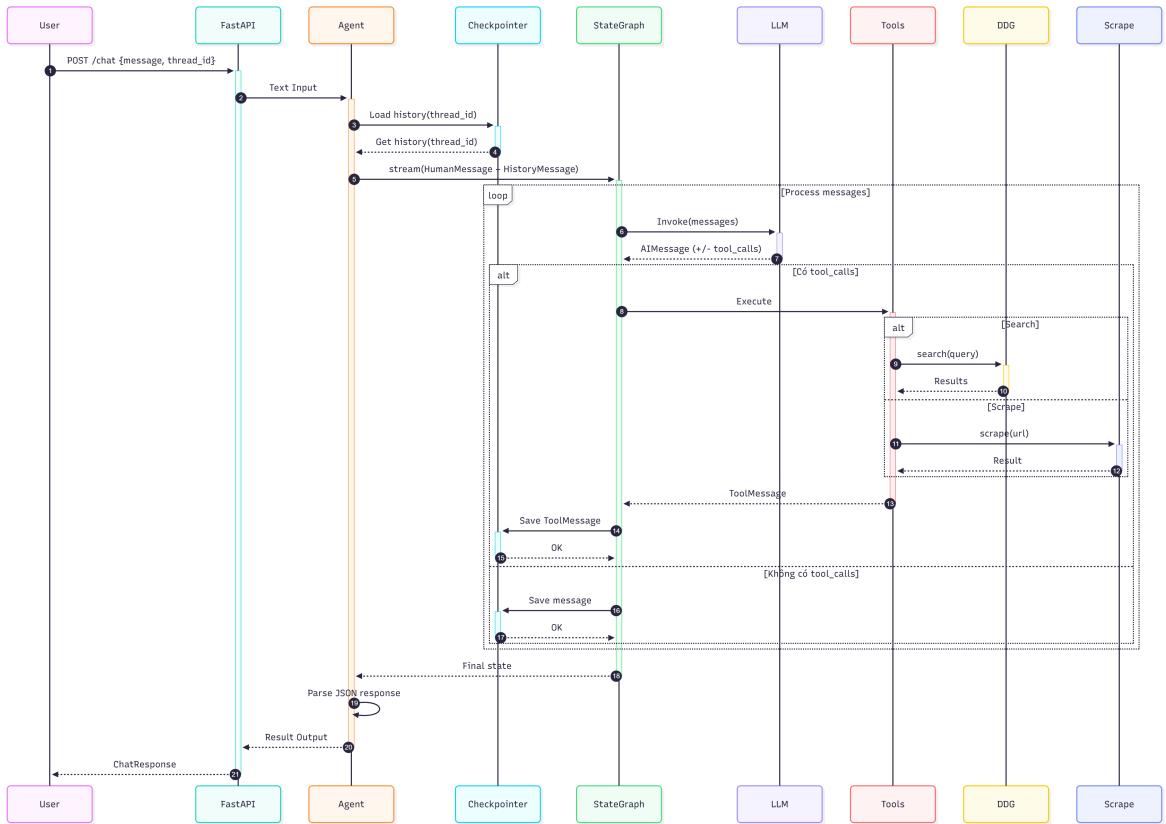
tổng hợp cả dữ liệu cũ theo luồng và dữ liệu mới từ người dùng để tạo thành một đầu vào hoàn chỉnh. Tiếp theo, chatbot sẽ đưa chuỗi đã được tổng hợp nào tạo thành một yêu cầu hoàn chỉnh và gửi đến LLM để xử lý. Lúc này LLM sẽ phân tích và xử lý yêu cầu. Nếu cần phải dùng đến các tool hỗ trợ, LLM sẽ có thể gọi các tool như:

- Duckduckgo: để tìm kiếm các trang web liên quan đến dữ liệu mà LLM yêu cầu.
- ScrapeWebsite: để trích xuất dữ liệu từ các trang web mà LLM muốn lấy thông tin.

Sau khi có được đầy đủ các thông tin cần thiết để LLM sẽ tổng hợp và tạo ra câu trả lời cuối cùng. Cuối cùng, chatbot sẽ lưu lại dữ liệu mới vào luồng tương ứng và có thể xử lý dữ liệu về dạng văn bản hoàn chỉnh, rõ ràng rồi trả về cho ứng dụng dữ liệu cuối cùng. Hình 4.8 mô tả quá trình này.

4.4.2. Đăng ký

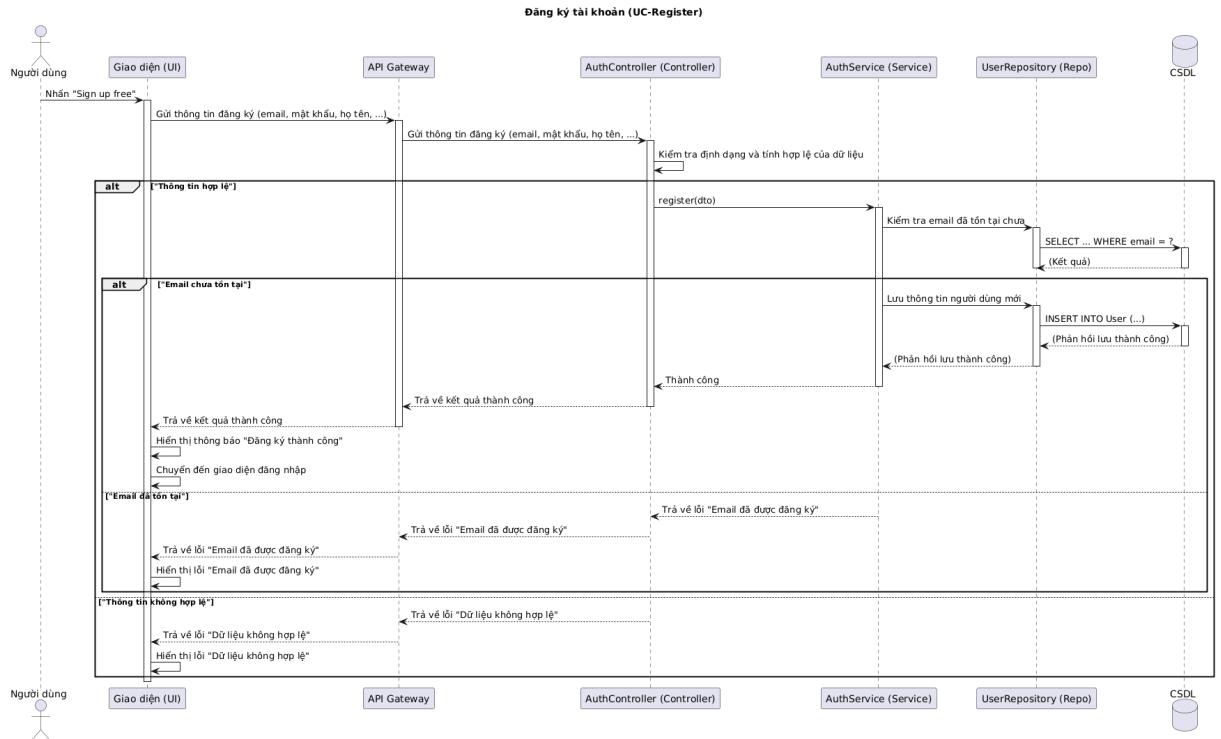
- Người dùng nhấn "Sign up for free", giao diện hiển thị form đăng ký.
- Gửi thông tin đăng ký qua API Gateway tới Controller, kiểm tra định dạng và tính hợp lệ của dữ liệu.
- Nếu thông tin hợp lệ, gửi thông tin cho Service kiểm tra email đã tồn tại chưa, nếu



Hình 4.8. Biểu đồ tuần tự cho chatbot

không hiển thị thông báo lỗi.

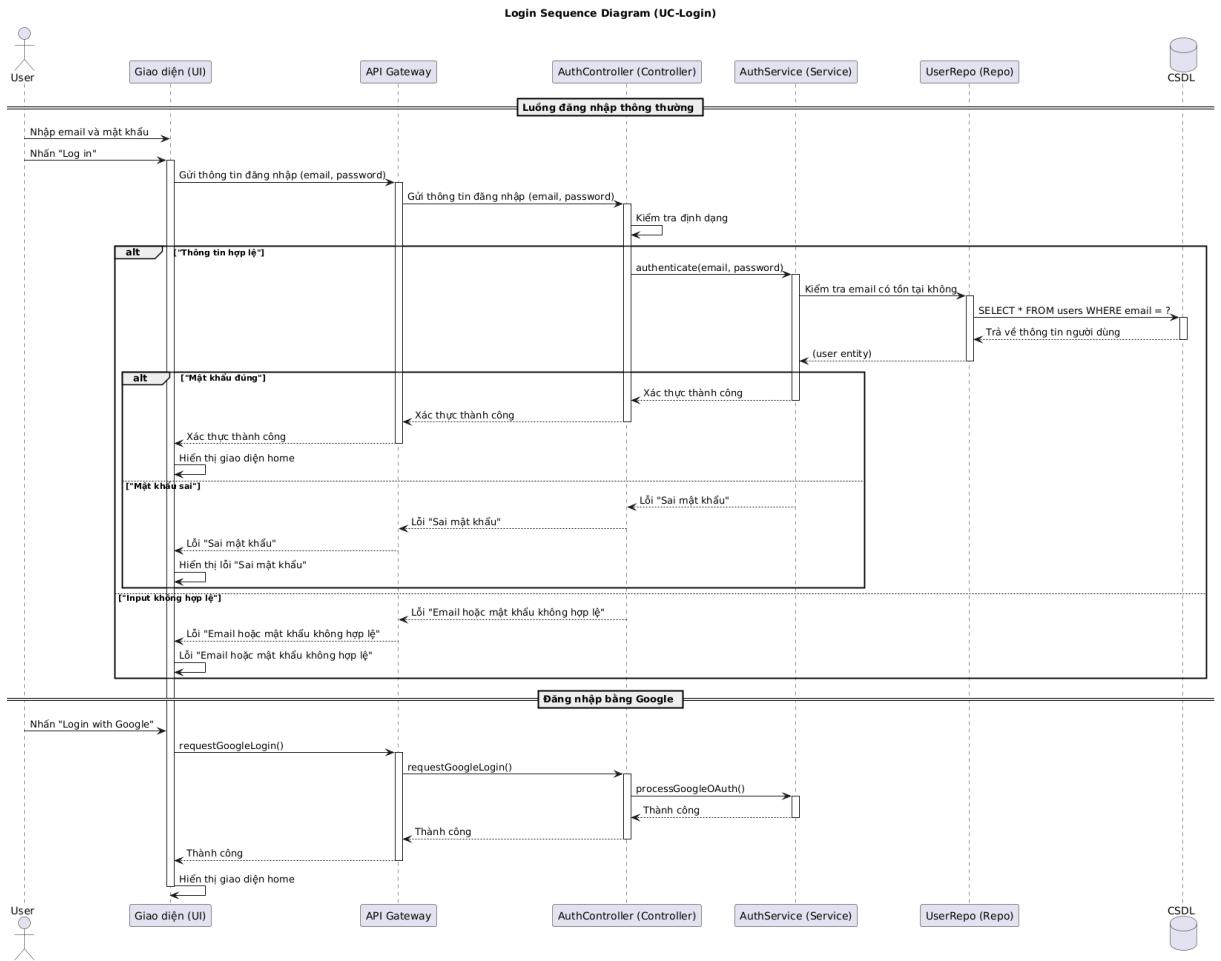
- Nếu email chưa tồn tại, gửi thông tin qua Repo, lưu vào CSDL, nếu không hiển thị thông báo lỗi.
- Chuyển đến giao diện đăng nhập.



Hình 4.9. Biểu đồ tuần tự cho chức năng đăng ký

4.4.3. Đăng nhập

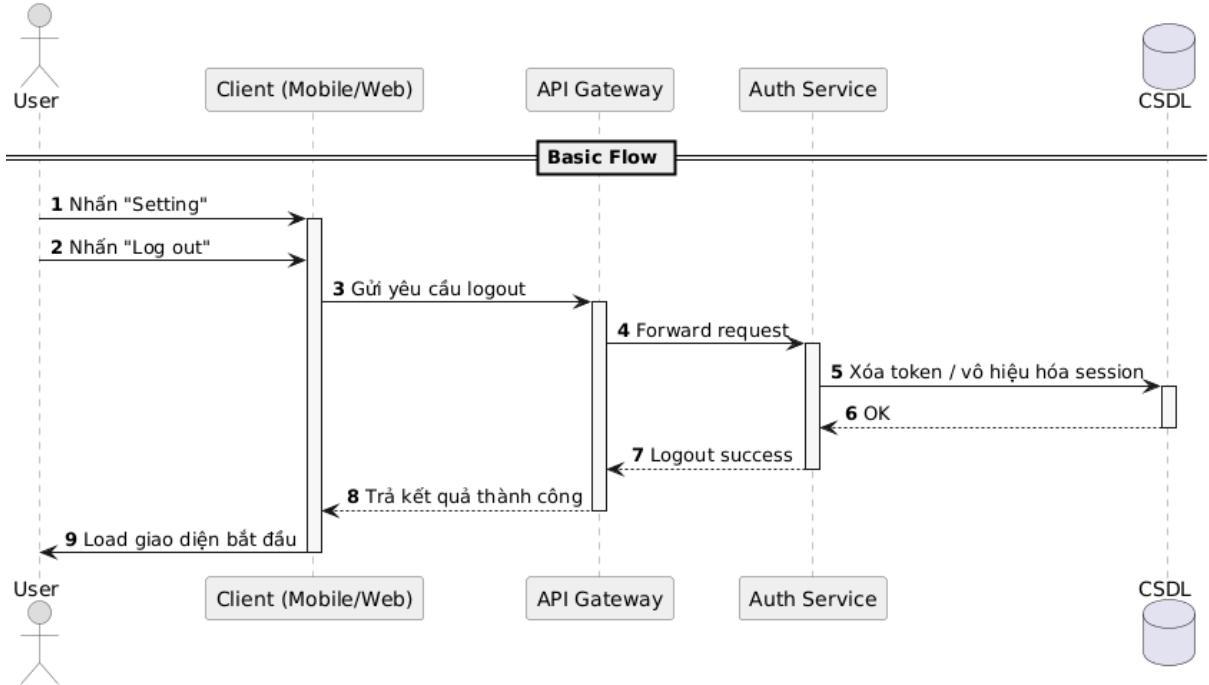
- Người dùng nhấn "Login" và nhập email và mật khẩu hoặc nhấn "Login with Google".
- Gửi thông tin đăng nhập qua API Gateway tới Controller, kiểm tra định dạng.
- Nếu thông tin hợp lệ, gửi cho Service kiểm tra email đã tồn tại trong CSDL chưa, nếu không thông báo lỗi.
- Nếu email tồn tại, kiểm tra thông tin mật khẩu.
- Nếu mật khẩu đúng, thông báo thành công, giao diện chuyển sang home, nếu không thông báo lỗi.



Hình 4.10. Biểu đồ tuần tự cho chức năng đăng nhập

4.4.4. Đăng xuất

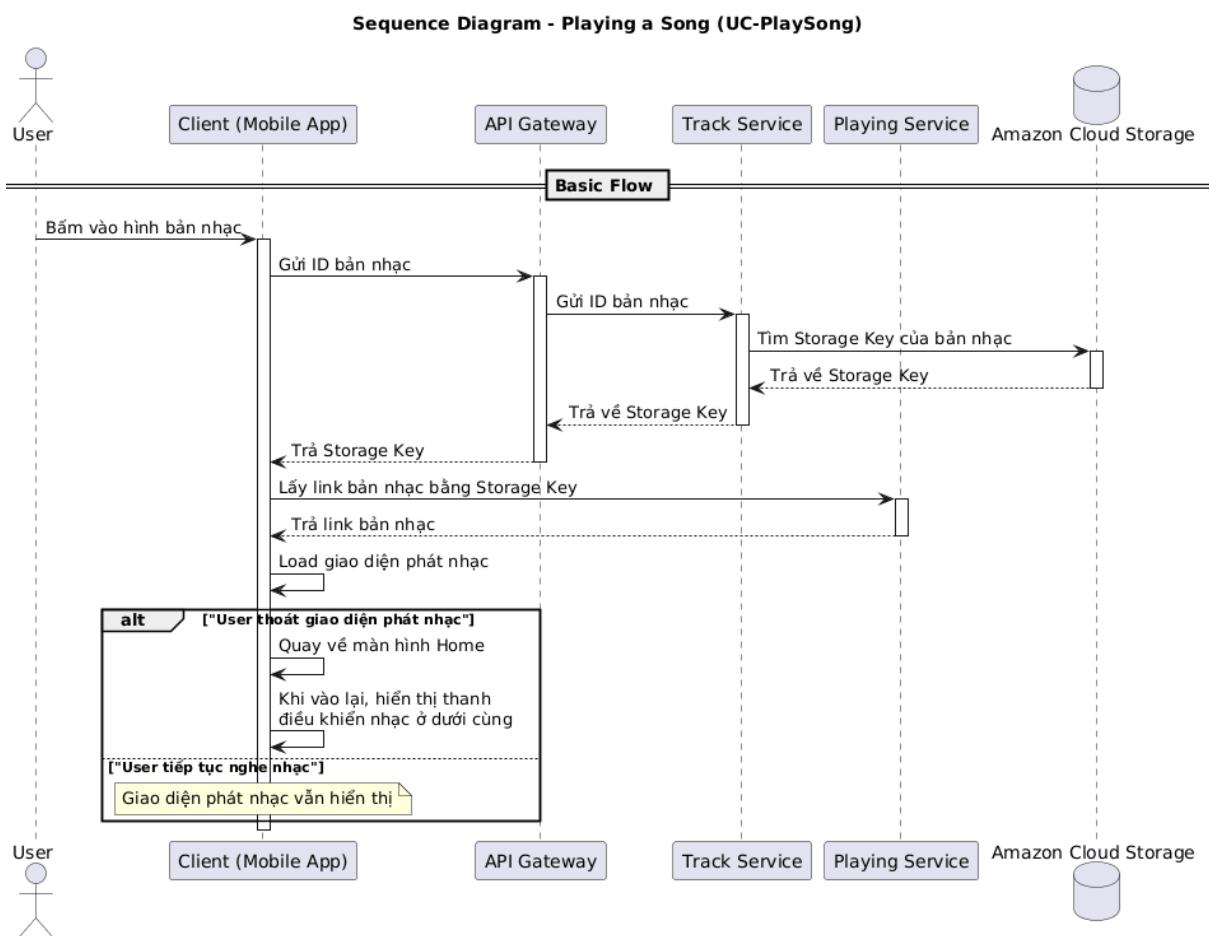
- Người dùng nhấn "Logout".
- Client gửi token người dùng qua API Gateway tới Controller.
- Nếu thông tin hợp lệ, gửi cho Service để xóa token từ CSDL.
- Trả về kết quả thành công, Client chuyển đến giao diện đăng nhập.



Hình 4.11. Biểu đồ tuần tự cho chức năng đăng xuất

4.4.5. Phát nhạc

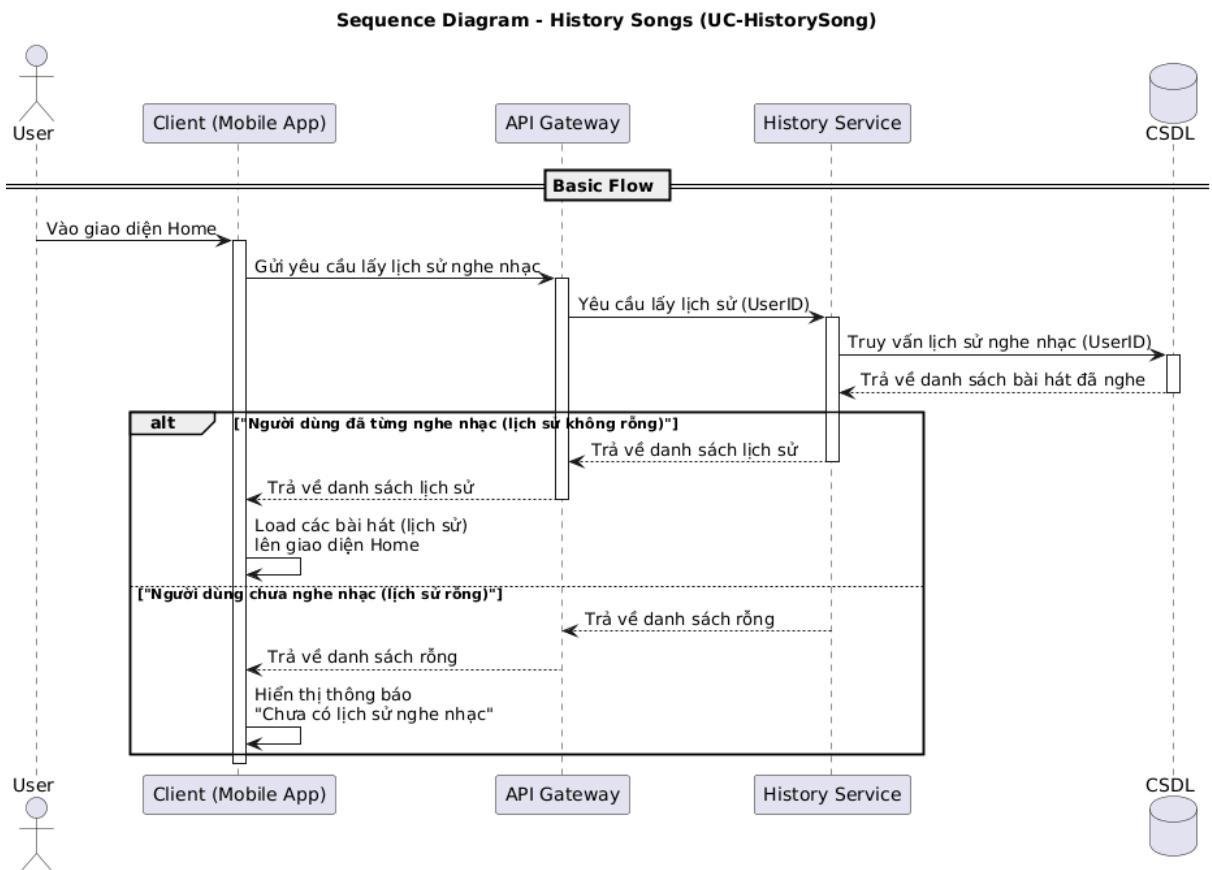
- Người dùng nhấn vào hình ảnh bài nhạc, client gửi qua API Gateway tới cho Service.
- Service tìm Storage Key của bài nhạc và gửi lại cho Client qua API Gateway.
- Client gửi Storage Key cho Service, Service trả về link bài nhạc.
- Client load giao diện phát nhạc và hiển thị cho người dùng.
- Người dùng thoát giao diện phát nhạc thì hiển thị giao diện home, khi vào lại hiển thị thanh điều khiển ở bên dưới.



Hình 4.12. Biểu đồ tuần tự cho chức năng phát nhạc

4.4.6. Xem bài hát gần đây

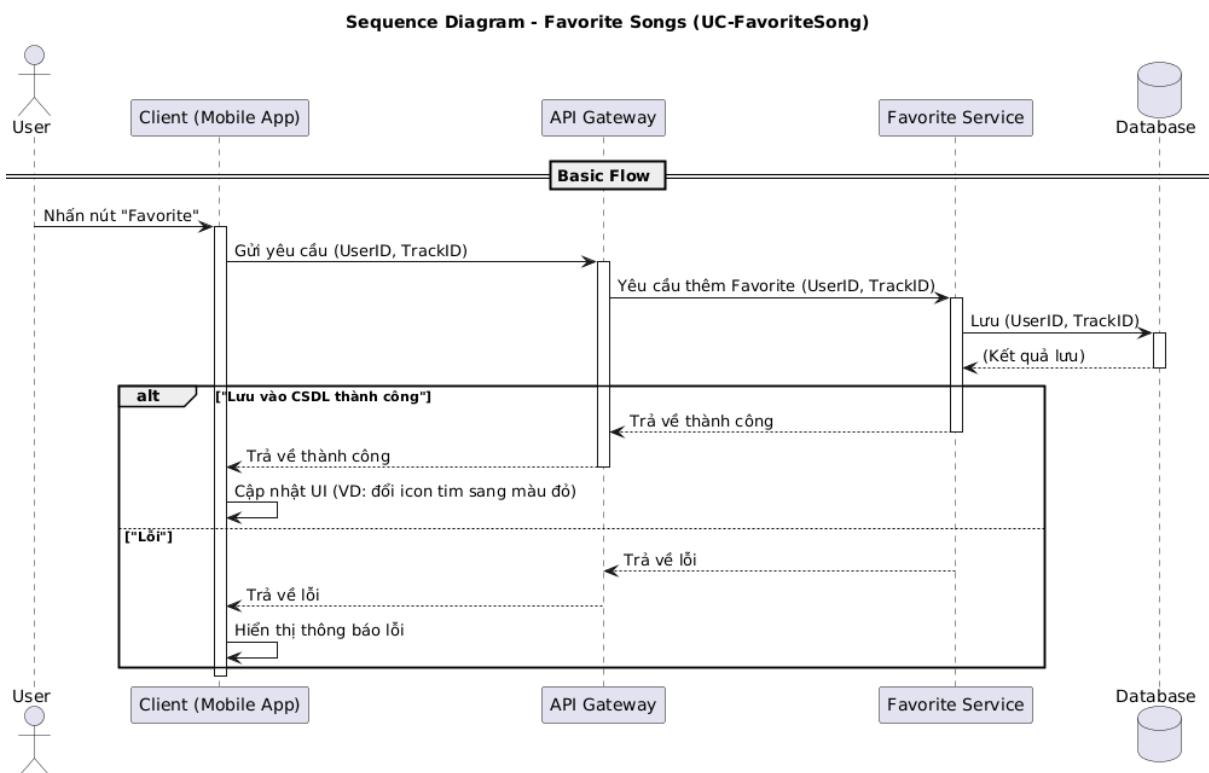
- Người dùng vào giao diện home, Client gửi yêu cầu lấy lịch sử nghe nhạc qua API Gateway tới Service.
- Service truy vấn và nhận lịch sử nghe nhạc từ CSDL.
- Nếu người dùng đã từng nghe nhạc, Service trả về danh sách lịch sử nghe nhạc qua API Gateway cho Client.
- Client load các bài hát trong lịch sử nghe nhạc lên giao diện home.
- Nếu người dùng chưa nghe nhạc, Service trả về danh sách rỗng, Client hiển thị thông báo "Chưa có lịch sử nghe nhạc"



Hình 4.13. Biểu đồ tuần tự cho chức năng xem bài hát gần đây

4.4.7. Lưu bài hát yêu thích

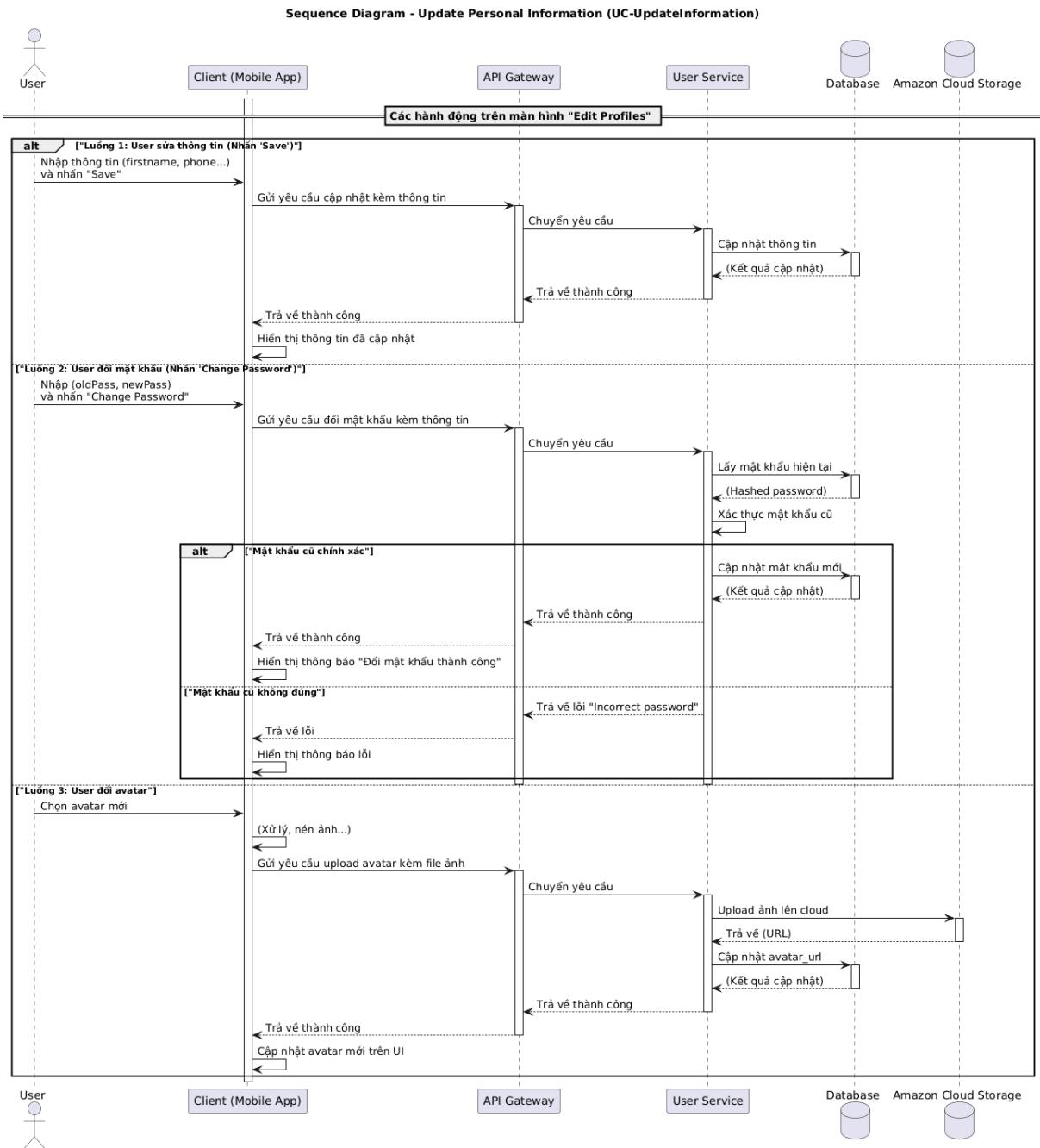
- Người dùng nhấn nút "Favorite", Client gửi yêu cầu qua API Gateway cho Service.
- Service gửi truy vấn tới CSDL để lưu thông tin.
- Nếu lưu vào CSDL thành công, Service trả về thành công, Client cập nhật UI.
- Nếu lưu vào CSDL không thành công, Service trả về lỗi, Client hiển thị thông báo lỗi.



Hình 4.14. Biểu đồ tuần tự cho chức năng lưu bài hát yêu thích

4.4.8. Sửa thông tin cá nhân

- Người dùng nhập thông tin vào form thông tin cá nhân/mật khẩu/avatar, Client gửi thông tin qua API Gateway tới Service.
- Service gửi truy vấn tới CSDL để lưu thông tin.
- Nếu lưu vào CSDL thành công, Service trả về thành công, Client cập nhật UI.
- Nếu lưu vào CSDL không thành công, Service trả về lỗi, Client hiển thị thông báo lỗi.

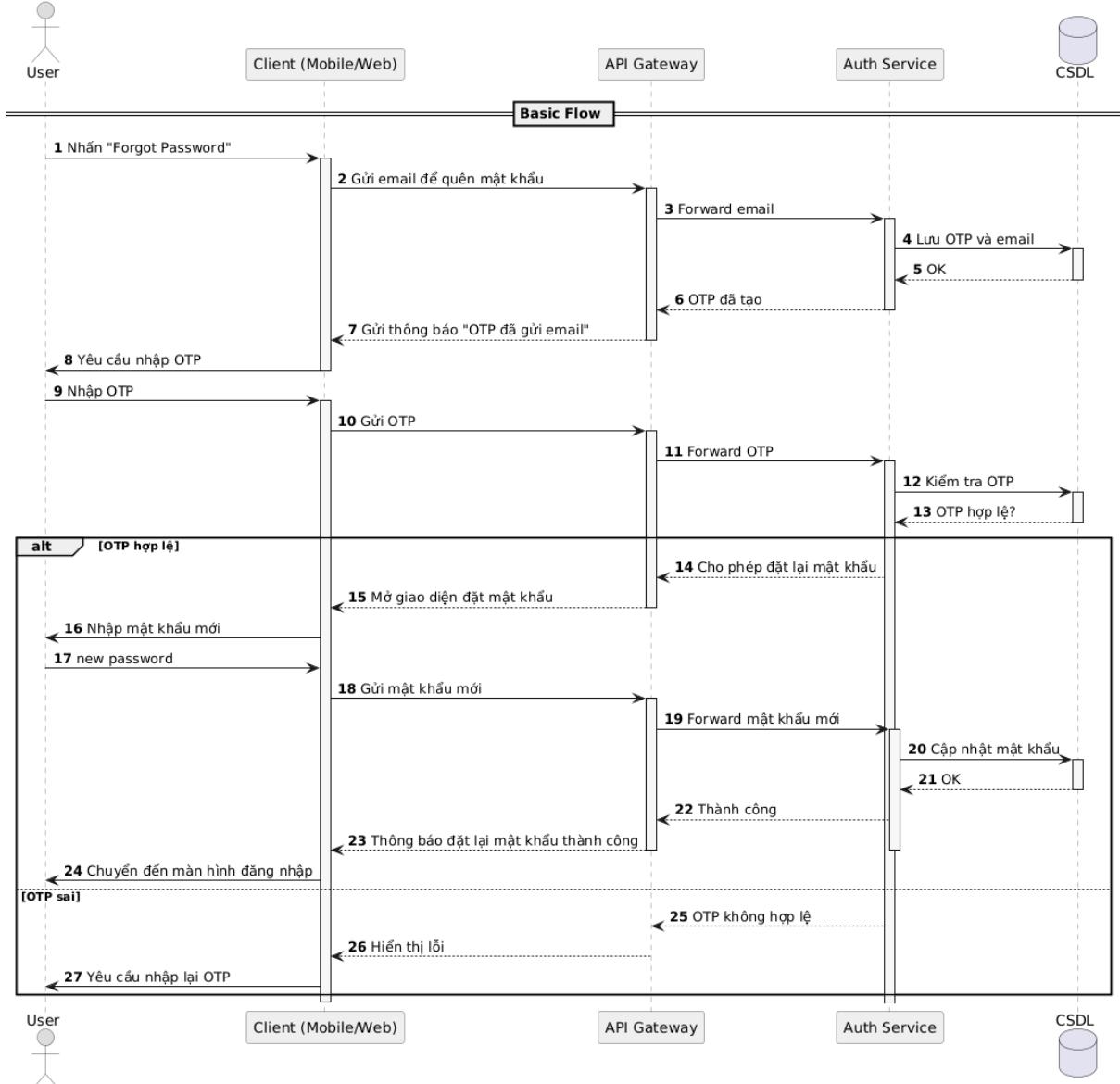


Hình 4.15. Biểu đồ tuần tự cho chức năng sửa thông tin cá nhân

4.4.9. Quên mật khẩu

- Người dùng nhập email vào form, Client gửi qua API Gateway tới Service.
- Service tạo otp gửi tới email người dùng và gửi truy vấn tới CSDL để lưu otp.
- Nếu lưu vào CSDL thành công, Service trả về thành công, Client chuyển sang giao diện xác thực otp.
- Người dùng xác thực otp, Client chuyển otp cho Service.

- Nếu otp hợp lệ, Service trả về thành công, Client chuyển đến giao diện cài đặt mật khẩu mới.
- Người dùng nhập mật khẩu mới vào form, Client gửi cho Service lưu vào CSDL và trả về thành công.



Hình 4.16. Biểu đồ tuần tự cho chức năng quên mật khẩu

4.5. Cơ sở dữ liệu

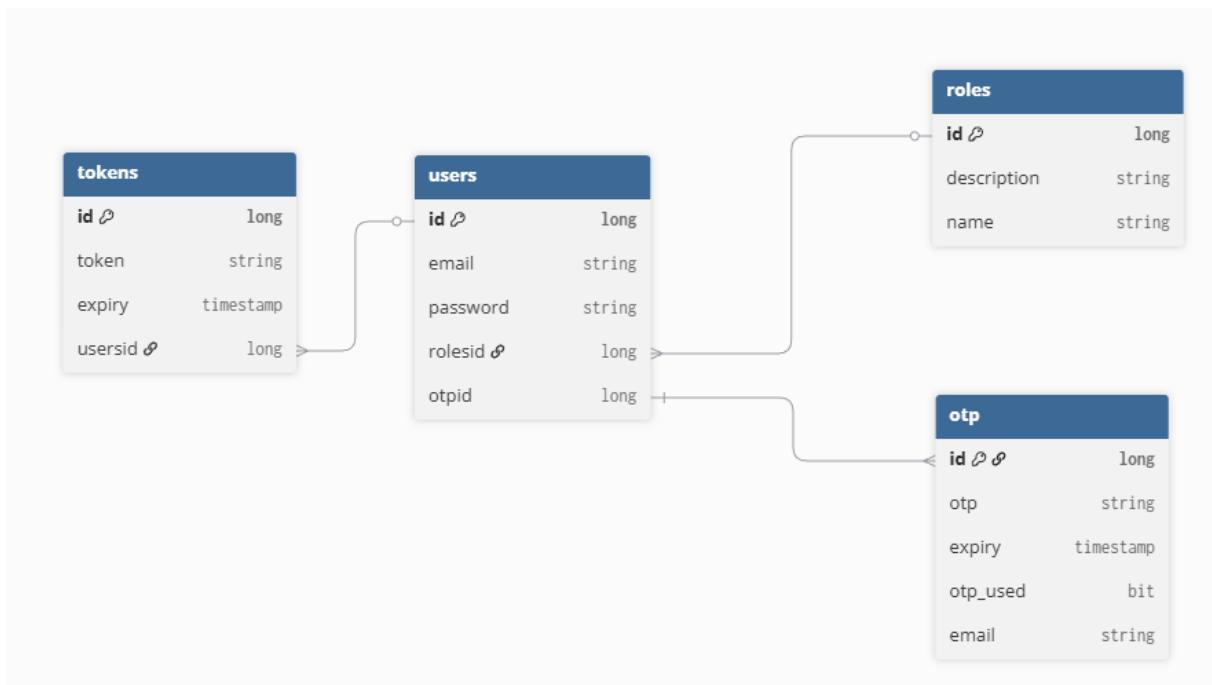
Trong kiến trúc Microservice, việc thiết kế các dịch vụ có nghiệp vụ riêng quan trọng vì điều này ảnh hưởng trực tiếp tới việc thiết kế cơ sở dữ liệu cho từng dịch vụ. Mỗi dịch vụ trong kiến trúc microservice sở hữu một cơ sở dữ liệu riêng không dùng chung với các dịch vụ khác. Thông tin và dữ liệu cần truy cập bởi nhiều dịch vụ vẫn thuộc một dịch vụ cố định và có phương thức giao tiếp truyền tin được thiết lập cho các dịch vụ có nhu cầu sử dụng. Với việc phân tách nghiệp vụ cho các dịch vụ, chúng tôi đã thiết kế cơ sở dữ liệu cho từng dịch vụ một cách riêng biệt. Trong phần này, chúng tôi sẽ phân tích cụ thể các thiết kế cơ sở dữ liệu trong hệ thống **InsightTune**.

4.5.1. Auth database

AuthService sử dụng cơ sở dữ liệu gồm các bảng chính:

- **users** dùng để lưu thông tin đăng ký/nhập người dùng.
- **roles** lưu vai trò và quyền hạn.
- **tokens** quản lý token xác thực.
- **otp** lưu mã OTP dùng cho xác thực quên mật khẩu.

Các bảng này có quan hệ với nhau để hỗ trợ phân quyền và quản lý xác thực an toàn.



Hình 4.17. Cơ sở dữ liệu của AuthService

4.5.2. User database

UserService sử dụng cơ sở dữ liệu gồm bảng chính **users** dùng để lưu thông tin cá nhân của người dùng, bao gồm: email, firstname, lastname, address, phone, role và avatar. Những thông tin này đóng vai trò quan trọng trong việc cá nhân hóa hệ thống nhằm nâng cao trải nghiệm người dùng.

users	
id ⚡	long
email	string
firstname	string
lastname	string
address	string
phone	string
role	string
avatar	string

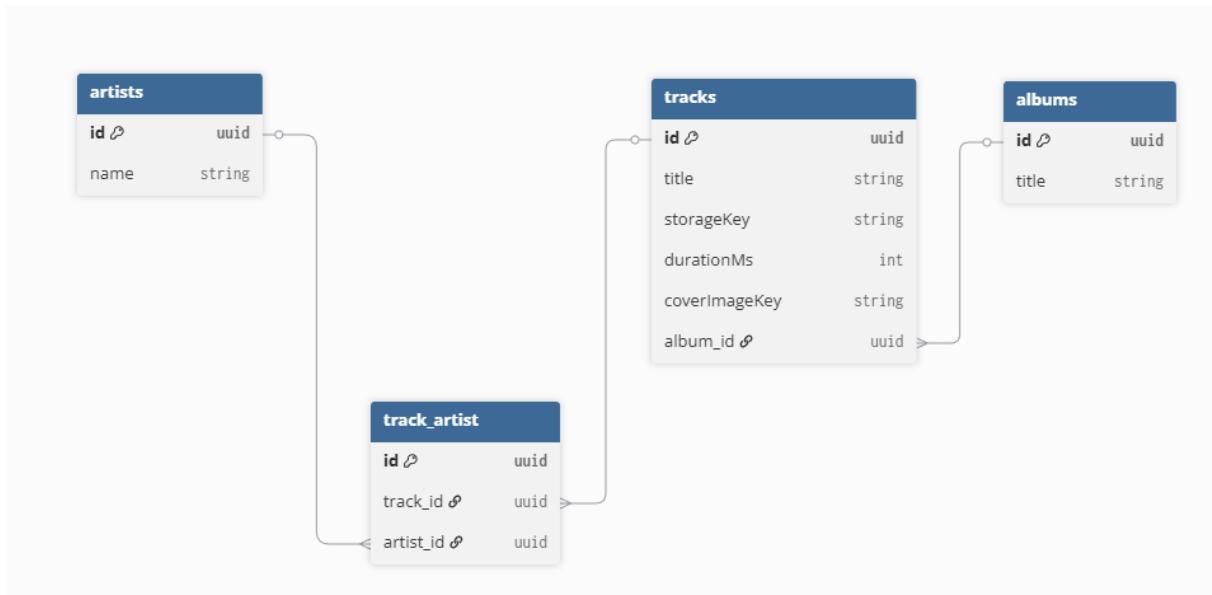
Hình 4.18. Cơ sở dữ liệu của UserService

4.5.3. Catalog database

CatalogService sử dụng cơ sở dữ liệu gồm các bảng chính:

- **tracks**: lưu thông tin dữ liệu bài hát (title, storageKey, coverImageKey, durationMs) và tham chiếu tới bảng albums.
- **tracks** lưu thông tin dữ liệu bài hát: title, storageKey, coverImageKey , durationMs lưu vị trí người dùng nghe và được tham chiếu tới bảng albums.
- **albums** lưu thông tin các danh sách bài hát.
- **track_artist** bảng phụ lưu thông tin tham chiếu giữa artists và tracks.

Các bảng này đóng vai trò thiết yếu trong hệ thống, là nơi lưu trữ dữ liệu cần thiết nhất cho hệ thống nghe nhạc trực tuyến.



Hình 4.19. Cơ sở dữ liệu của CatalogService

4.5.4. History database

HistoryService sử dụng cơ sở dữ liệu gồm các bảng chính: **histories** dùng để lưu lịch sử và thời gian về các bài hát mà người dùng đã phát, **search_histories** lưu lịch sử tìm kiếm của người dùng. Những thông tin này đóng vai trò bổ trợ cho hệ thống, giúp nâng cao trải nghiệm người dùng.



Hình 4.20. Cơ sở dữ liệu của HistoryService

4.5.5. Favorite database

FavoriteService sử dụng cơ sở dữ liệu gồm bảng chính **favorite** dùng để lưu các bài hát yêu thích của người dùng, bao gồm songId và email nhằm xác định ai đã thêm bài

hát nào vào yêu thích. Những thông tin này đóng vai trò bổ trợ cho hệ thống, giúp nâng cao trải nghiệm người dùng.

favorites	
id	 long
songId	uuid
email	string
created_at	timestamp

Hình 4.21. Cơ sở dữ liệu của FavoriteService

CHƯƠNG 5
XÂY DỰNG HỆ THỐNG AI

5.1. Xây dựng trợ lý ảo

5.1.1. Bài toán đặt ra

Nhận thấy việc phát triển của trí tuệ nhân tạo (AI) và các mô hình ngôn ngữ lớn (LLM) đang ngày càng phát triển và mang lại rất nhiều tiện ích cho người dùng. Và đồng thời khi dùng các ứng dụng âm nhạc hiện nay thì chúng tôi nhận ra được một vấn đề rằng người dùng thường sẽ hay thắc mắc hoặc tò mò về các thông tin liên quan đến bài hát, ca sĩ hoặc muốn biết xu hướng âm nhạc hiện nay ra sao. Và việc đang nghe nhạc mà lại phải thoát ra để sử dụng một ứng dụng khác để tìm kiếm thông tin khá là bất tiện và làm gián đoạn trải nghiệm người dùng.

Và hiện nay cũng có một vài ứng dụng đã nhận thấy được vấn đề này và dần tích hợp chatbot vào ứng dụng phát nhạc nhưng chatbot của họ thường chỉ trả lời dựa trên kiến thức được lưu trữ trong cơ sở dữ liệu có sẵn và không tự cập nhật những thông tin mới nhất từ trên mạng. Đây cũng chính là một bất tiện lớn trong việc tối ưu hóa trải nghiệm người dùng bởi người dùng không được cập nhật thông tin một cách chính xác và mới nhất. Do đó chính tôi đã phát triển ra một hệ thống chatbot agent có khả năng tìm kiếm thông tin trên mạng để có thể trả lời người dùng một cách chính xác và bắt kịp thời đại nhanh nhất, không bị giới hạn bởi lượng kiến thức tĩnh.

5.1.2. Giải pháp đề xuất

Chúng tôi xây dựng một hệ thống chatbot agent kết hợp với các tool tìm kiếm và lấy thông tin từ trên mạng để cung cấp dữ liệu cho agent với model chính mà chúng tôi sử dụng là gemini-2.5-pro. Điểm mạnh của chatbot của chúng tôi chính là khả năng tìm kiếm thông tin từ trên mạng, có thể tự động cập nhật những tri thức mới nhất để trả lời cho người dùng mà không bị thiếu kiến thức hay bị sai lệch thông tin so với các hệ thống chatbot truyền thống dùng các model tĩnh kết hợp RAG (retrieval-augmented generation) để cung cấp ngữ cảnh cho LLM. Ngoài ra chúng tôi cũng tự làm thêm tính năng phát nhạc khi người dùng yêu cầu hoặc có mong muốn nghe nhạc trong app của mình và có thể yêu cầu bằng giọng nói hoặc nhập yêu cầu bằng văn bản mà không cần phải thao tác thủ công.

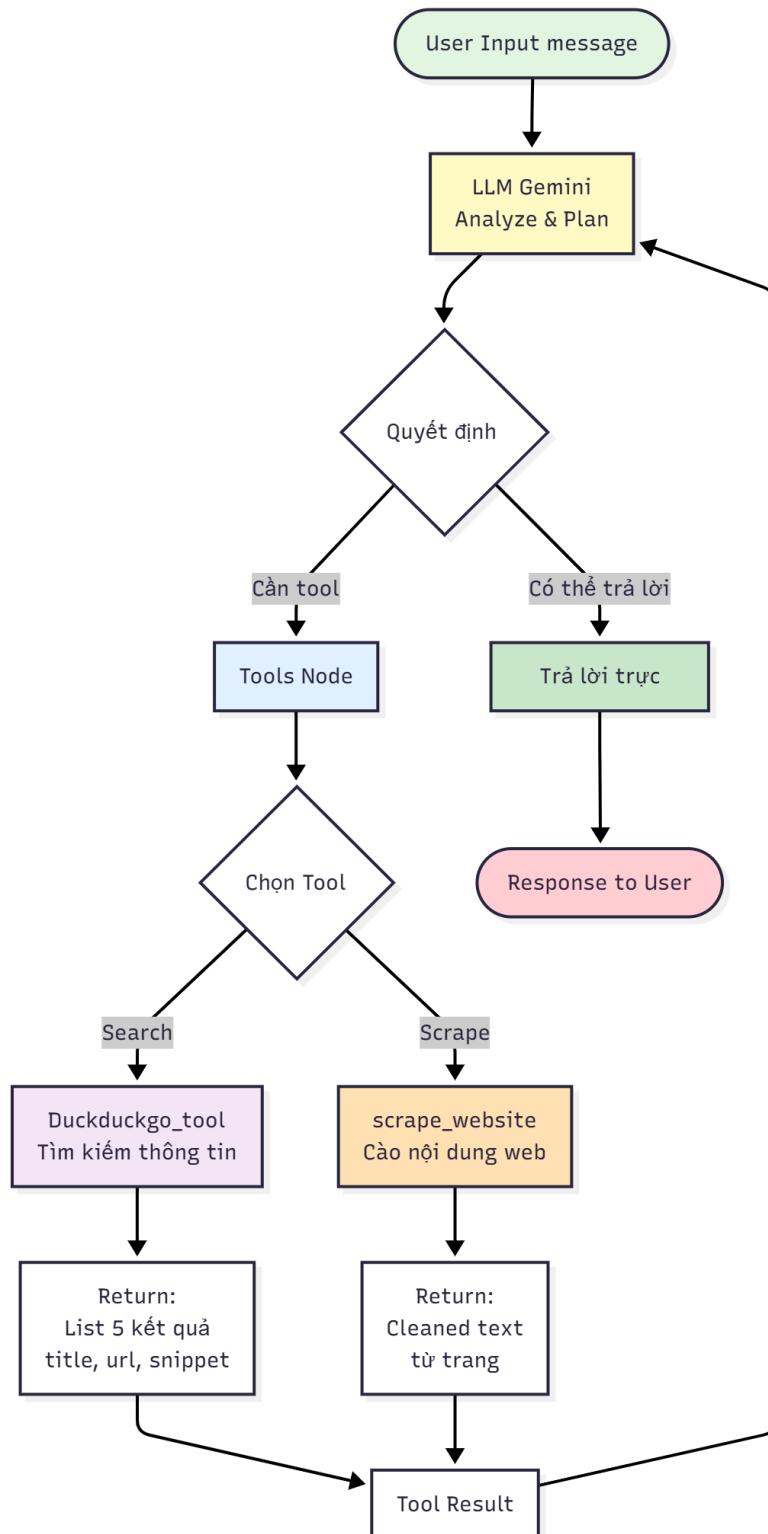
5.1.3. Công nghệ và luồng hoạt động của chatbot agent

Chatbot agent được xây dựng chủ yếu bằng framework mạnh mẽ cho agent là Langgraph và một số các thư viện cần thiết khác để phục vụ cho việc triển khai hệ thống dễ dàng hơn. Langgraph giúp chúng tôi xây dựng chatbot agent theo dạng đồ thị (graph) với

các node đại diện cho các thành phần như tool hay agent. Ngoài ra việc dùng Langgraph giúp chúng tôi có thể dễ dàng mở rộng hơn vì nó tối ưu và dễ dàng phát triển để mở rộng thành hệ thống multi-agent trong tương lai.

5.1.3.1. Luồng hoạt động chính

Trong phần này sẽ mô tả luồng hoạt động cơ bản của hệ thống chatbot agent mà chúng tôi xây dựng. Sơ đồ luồng hoạt động chính của agent được thể hiện trong hình 5.1 dưới đây:



Hình 5.1. Luồng hoạt động của agent trong hệ thống chatbot AI

Khi người dùng nhập input đầu vào, hệ thống sẽ gửi input này tới chatbot thông qua FastAPI. Chatbot sẽ xử lý input này và quyết định xem việc nên dùng tools nào phù hợp.

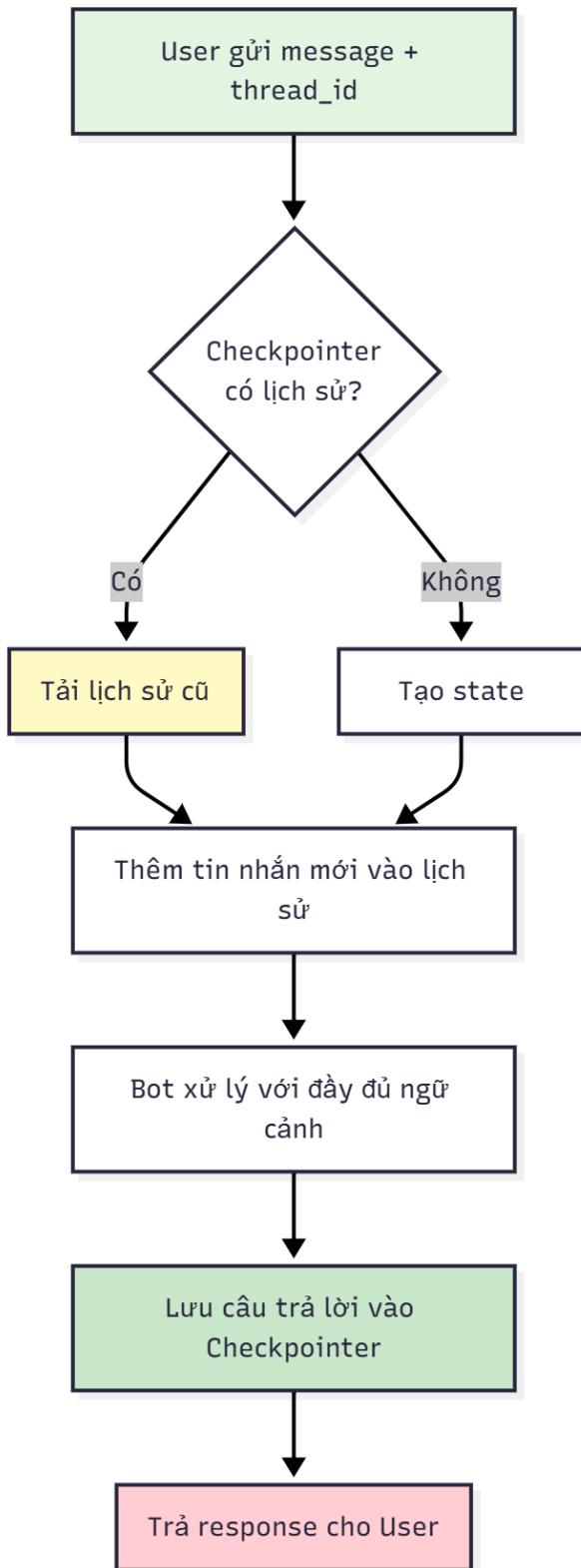
Khi cần tra cứu thêm thông tin này từ trên mạng, chatbot agent sẽ gọi function calling để gọi đến tool Duckduckgo Search và trả về các link liên quan đến kiến thức mà chatbot cần.

Đến đây, chatbot vẫn chưa có lượng thông tin cần thiết để trả lời bởi vì mới chỉ lấy được các link liên quan. Do đó, chatbot sẽ cần đến tool là Scape Website để lấy dữ liệu từ trang web về ở dạng html.

Lúc này tool Scape Website sẽ trả về input mong muốn cho chatbot. Và rồi chatbot có thông tin và từ đó dựa trên những thông tin ấy trả lời cho người dùng.

5.1.3.2. Quản lý bộ nhớ của Chatbot

Ở đây chúng tôi quản lý bộ nhớ của chatbot thông qua việc sử dụng LangGraph Checkpointer. Hình 5.2 mô tả luồng hoạt động của bộ nhớ trong hệ thống chatbot agent của chúng tôi



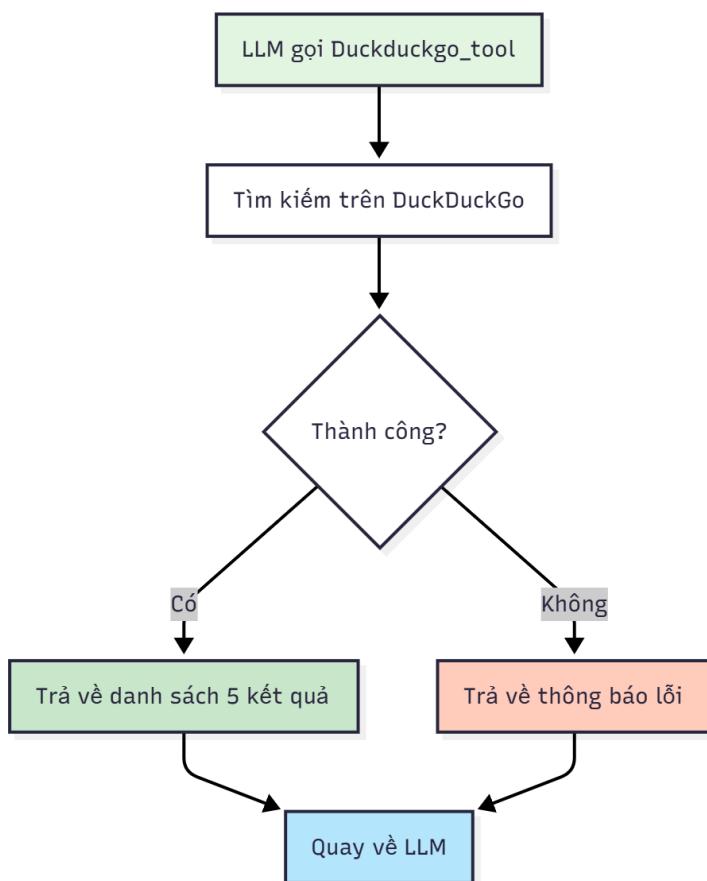
Hình 5.2. Luồng hoạt động của bộ nhớ trong hệ thống chatbot AI

Mỗi cuộc trò chuyện với người dùng sẽ được định danh trong một luồng đã được định danh riêng biệt, và Checkpointer sẽ lưu trữ toàn bộ lịch sử tin nhắn của từng luồng

đã được định danh. Và mỗi khi truy vấn, chatbot sẽ lấy dữ liệu theo luồng đó và cung cấp thông tin cho LLM. Điều này làm Chatbot duy trì được ngữ cảnh và cá nhân hóa với mỗi lần chat.

5.1.3.3. Duckduckgo Tool

Duckduckgo là một công cụ tìm kiếm, cho phép bạn tra cứu thông tin mà không bị theo dõi hay thu thập dữ liệu cá nhân. Chatbot sẽ gửi yêu cầu tìm kiếm đến Duckduckgo thông qua chức năng function calling. Duckduckgo sẽ nhận đầu vào là một chuỗi kí tự tìm kiếm và rồi sẽ truy vấn để tìm ra 5 link liên quan nhất và trả về cho agent. Hình 5.3 mô tả luồng hoạt động của duckduckgo tool trong hệ thống chatbot agent mà chúng tôi phát triển

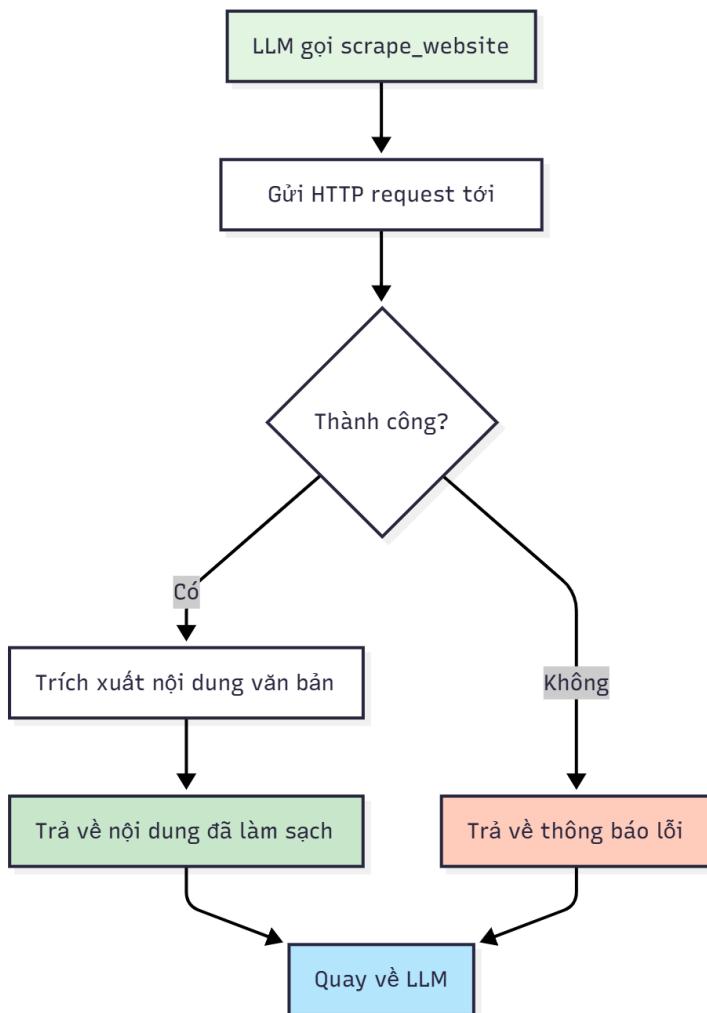


Hình 5.3. Luồng hoạt động của duckduckgo tool trong hệ thống chatbot AI

Khi LLM quyết định cần tìm kiếm, gọi Duckduckgo tool kèm theo dữ liệu cần truy vấn. Sau đó tool Duckduckgo sẽ trả về tối đa 5 link liên quan nhất đến truy vấn của LLM. Còn nếu bị lỗi LLM sẽ trả về thông báo lỗi để LLM biết và xử lý, có thể gọi lại tool hoặc đưa ra thông báo bị lỗi cho người dùng.

5.1.3.4. Scrape Website Tool

Scrape Website là một tool được xây dựng để lấy dữ liệu từ các trang web dựa trên đường dẫn mà LLM cung cấp khi được gọi thông qua function calling. Tool này giúp chatbot có thể chủ động lấy thông tin từ các trang web mà tool Duckduckgo tìm được để lấy dữ liệu và trả về cho LLM nhằm cung cấp câu trả lời chính xác nhất cho người dùng. Hình 5.4 mô tả luồng hoạt động của scrape website tool trong hệ thống chatbot agent mà chúng tôi phát triển



Hình 5.4. Luồng hoạt động của scrape website tool trong hệ thống chatbot AI

Tool Scrape Website sẽ nhận đầu vào là một đường dẫn từ LLM trả về khi dùng duckduckgo tool để tìm kiếm. Tool sẽ lấy toàn bộ nội dung website ở dạng văn bản và trả về toàn bộ chúng. Lúc này văn bản còn nhiều dữ liệu dư thừa vì ở dạng html, nên chúng tôi sẽ sử dụng framework BeautifulSoup để lọc dữ liệu và trả về chuỗi chỉ chứa nội dung của trang website.

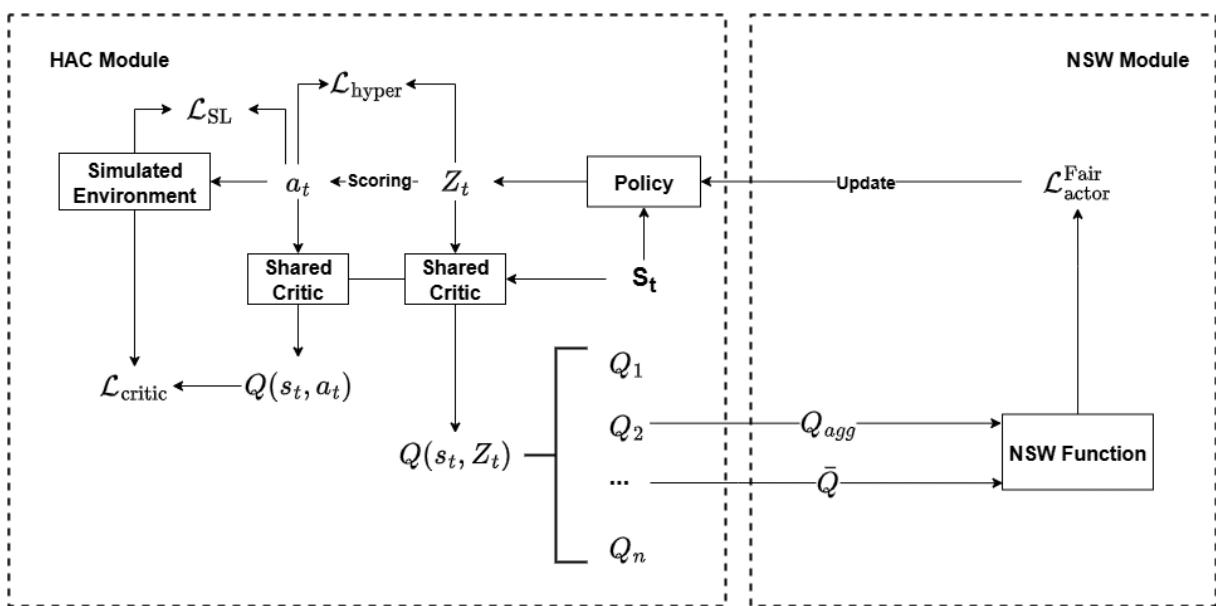
5.1.3.5. Tính năng phát nhạc

Để thêm phần trải nghiệm người dùng, chúng tôi đã xây dựng thêm tính năng phát nhạc trong ứng dụng chatbot của mình. Người dùng có thể yêu cầu phát nhạc bằng cách nhập yêu cầu với chatbot hoặc sử dụng giọng nói để ra lệnh phát nhạc. Đối với phát nhạc bằng giọng nói, chúng tôi dùng model Whisper của OpenAI để chuyển đổi giọng nói thành văn bản rồi gửi văn bản này đến Chatbot để chatbot phân tích và xử lý yêu cầu phát nhạc. Cuối cùng trả về cho ứng dụng biết rằng đây là yêu cầu phát nhạc và ứng dụng sẽ xử lý phần logic phát nhạc. Chúng tôi phải sử dụng kỹ thuật prompt để hướng dẫn chatbot cho đầu ra theo đúng định dạng mà ứng dụng có thể nhận biết được và phát nhạc đúng yêu cầu của người dùng.

5.2. Xây dựng mô hình gợi ý

5.2.1. Giới thiệu về HAC-NSW

Hyper Actor Critic Framework with Nash Social Welfare (**HAC-NSW**) là một mô hình học tăng cường phục vụ cho hệ thống gợi ý nền tảng chuỗi sự kiện, hệ thống nhận đầu vào một chuỗi thông tin (ví dụ lịch sử âm nhạc của người dùng), sau đó sẽ trả về một tập các gợi ý (ví dụ năm bài hát được gợi ý). Luồng xử lý của mô hình này được biểu diễn trong biểu đồ 5.5 dưới đây.



Hình 5.5. Luồng của mô hình HAC-NSW

HAC-NSW chính là mở rộng của **Hyper Actor Critic** Framework, một framework mạnh mẽ cho gợi ý với không gian gợi ý lớn (large item space). Nó hoạt động dựa trên

cơ chế nén không gian dữ liệu lớn lại thành **miền ẩn** (latent space) thông qua một phép chiếu từ không gian bậc cao với số chiều lớn xuống một không gian bậc thấp có số chiều nhỏ hơn. Mô hình tìm kiếm những gợi ý trên miền này sau đó phục dựng những gợi ý trong không gian gợi ý ban đầu qua một phép chiếu ngược của phép chiếu ban đầu. Bản chất của mô hình HAC nằm ở việc học hai phép chiếu này một cách hiệu quả. Vấn đề chính nảy sinh từ việc chính sách tối ưu sau quá trình học tăng cường lựa chọn quyết sách tối ưu trên không gian ẩn chứ không phải không gian ban đầu. Vậy làm sao để tìm kiếm hiệu quả trên miền ẩn và đảm bảo sự đúng đắn của phép chiếu (nếu phép chiếu sai thì chính sách cũng sai theo)?

Câu lời của HAC là thêm một hàm măt măt thể hiện sự **thống nhất** giữa hai phép chiếu ngược và xuôi kèm theo nhiễu Gaussian làm tăng hiệu quả việc **tìm kiếm** trên miền ẩn. HAC ban đầu sinh ra như có thể thấy, chính là HAC Module trong hình 5.5. Nhưng một vấn đề khác nảy sinh trong quá trình quan sát cách hoạt động của mô hình này, việc tối ưu hóa trên miền ẩn gây nên sự măt cân bằng lớn giữa nhiều người dùng (có người sẽ nhận kết quả gợi ý rất tốt trong khi số khác lại không). Đó là vì HAC thiếu đi một module có khả năng đem thông tin của sự cân bằng giữa những người dùng vào để tối ưu hóa chính sách.

Điều này dẫn tới động lực khai sinh ra HAC-NSW. NSW chính là một hàm thuộc họ Social Welfare, hàm này có khả năng tích hợp tốt và linh hoạt, quan trọng nhất là nó bao hàm trực tiếp thông tin về sự **cân bằng** trong hàm. Từ đó tích hợp NSW vào HAC làm tăng tính cân bằng giữa các người dùng của hệ thống.

5.2.2. HAC-NSW và InsightTune

Nhận thấy sự cân bằng này là rất quan trọng trong một trình duyệt âm nhạc nhiều người dùng, và phù hợp với mục đích tích hợp AI một cách hiệu quả, chúng tôi đã lựa chọn mô hình HAC-NSW làm mô hình chính cho hệ thống **InsightTune**

Để xây dựng hiệu quả mô hình gợi ý, chúng tôi đã thu thập dữ liệu từ nhiều nguồn khác nhau trên mạng. Vì mục đích chính của **InsightTune** là làm một ví dụ, khuôn mẫu tích hợp AI một cách quả nêu cần thiết phải có một hệ thống gợi ý hoàn chỉnh tương thích cao. Dữ liệu hiện tại của hệ thống **InsightTune** chưa đủ khả năng cho thấy toàn bộ tiềm năng của mô hình. Do đó, trong báo cáo này, chúng sẽ tập trung phân tích quá trình xây dựng và kết quả kiểm thử trên tập dữ liệu lớn để tăng sức thuyết phục so với sử dụng chỉ một vài người dùng sẵn có trong hệ thống.

5.2.3. Xây dựng huấn luyện mô hình

Trong mục này, chúng tôi sẽ tập trung trình bày quá trình xây dựng huấn luyện mô hình HAC-NSW thích hợp với hệ thống. Điều này yêu cầu phải có một bộ dữ liệu âm nhạc đầy đủ, có các đặc điểm rõ ràng và một quá trình huấn luyện kiểm thử kỹ càng.

5.2.3.1. Dữ liệu

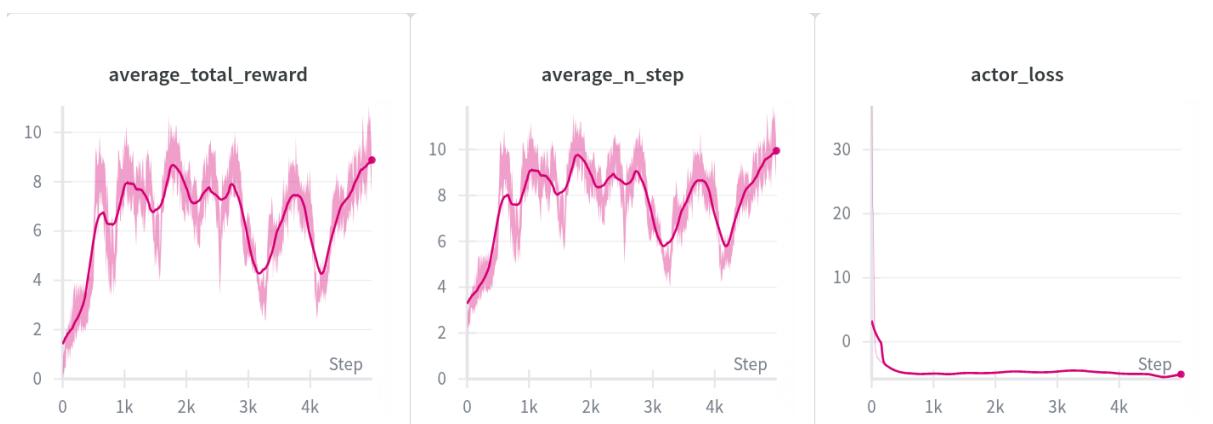
Dữ liệu là điều rất quan trọng với một mô hình học tăng cường. Chúng tôi thu thập dữ liệu bằng cách kết hợp hai bộ dữ liệu lớn. Đó là Spotify Million Playlist Dataset (MPD) và Spotify 1.2M+ Songs. Cụ thể quá trình này đã được ghi lại trên Kaggle. Trong báo cáo này chúng tôi chỉ trình bày ngắn gọn lại như sau:

- Lấy dữ liệu từ hai bộ dữ liệu về
- Loại bỏ các mẫu không trùng nhau giữa hai bộ dữ liệu
- Biến đổi lấy mẫu để tạo thành các lịch sử tương tác
- Sử dụng dữ liệu này để cho quá trình huấn luyện

5.2.3.2. Huấn luyện

Đầu tiên, chúng tôi tạo ra một mô hình giả lập lại môi trường thực tế. Mô hình này được xây dựng trên mô hình Transformer cơ bản, có độ nhiễu nhất định. Sau đó mô hình HAC-NSW sẽ tương tác với mô hình này để học được chính sách tối ưu như tương tác với môi trường thực tế.

Quá trình huấn luyện của chúng được thể hiện qua các bảng biểu sau đây.

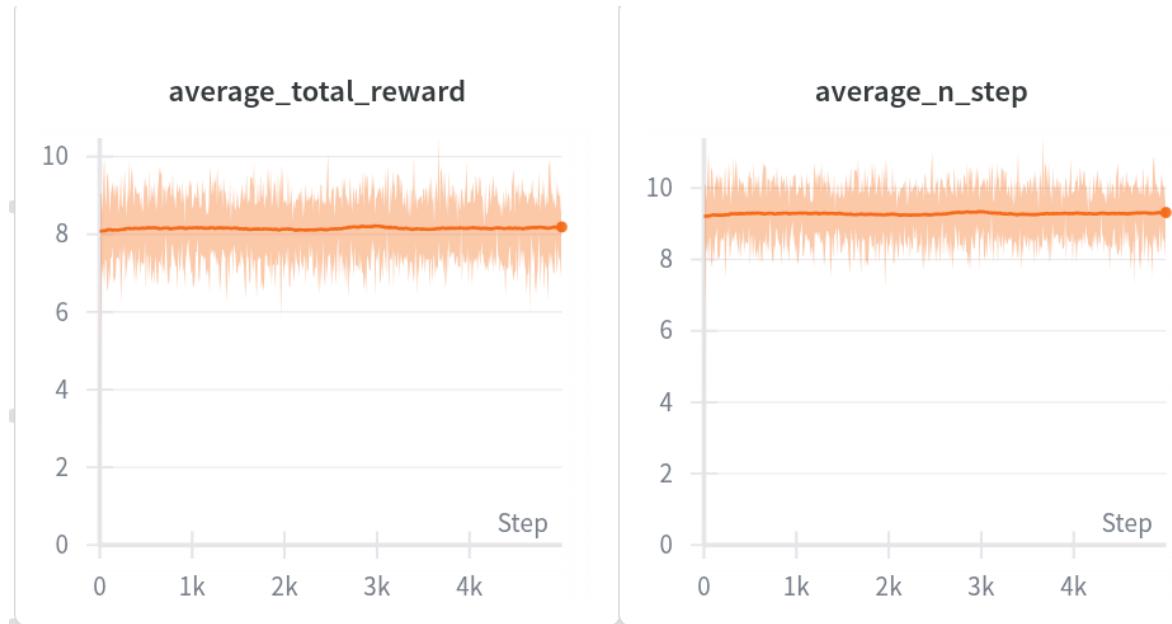


Hình 5.6. Bảng biểu thể hiện tương tác trung bình khi hàm mất mát của actor cải thiện

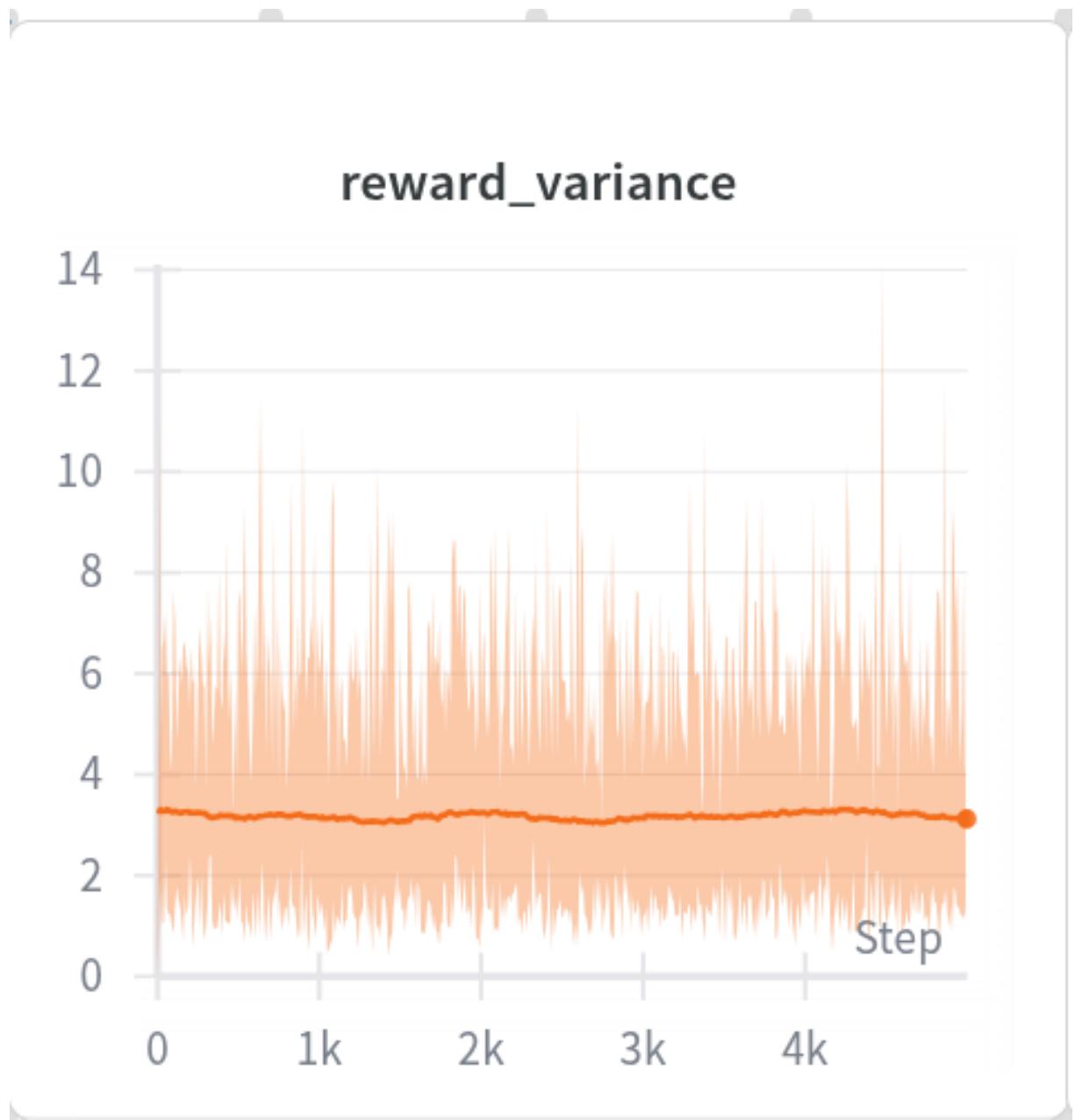


Hình 5.7. Bảng biểu thể hiện kết quả thấp nhất trong người dùng và độ chênh ổn định thấp khi hàm mất mát actor cải thiện

Mô hình cũng thể hiện sự tương tác tốt trên tập kiểm thử với kết quả như sau.



Hình 5.8. Bảng biểu thể hiện kết quả trung bình tốt trên tập kiểm thử



Hình 5.9. Bảng biểu thể hiện độ lệch vẫn ổn định thấp trên tập kiểm thử

Điều này chứng minh chúng đã huấn luyện được mô hình đủ ổn định để triển khai trên một hệ thống lớn.

CHƯƠNG 6

CÀI ĐẶT VÀ KIỂM THỬ

6.1. Giao diện hệ thống

Giao diện của hệ thống được thiết kế thân thiện, dễ sử dụng và phù hợp với đại đa số người dùng. Giao diện được xây dựng bằng Jetpack Compose, một thư viện hiện đại của Android giúp xây dựng giao diện người dùng một cách nhanh chóng và hiệu quả.

Giao diện của ứng dụng được tổ chức theo các chức năng chính, bao gồm:

- Màn hình bắt đầu
- Màn hình đăng ký tài khoản
- Màn hình đăng nhập
- Màn hình trang chủ
- Màn hình trình phát nhạc
- Màn hình tìm kiếm và bài hát yêu thích
- Màn hình thông tin người dùng
- Màn hình chatbot hỗ trợ người dùng

Các thành phần giao diện đều đảm bảo tính nhất quán về màu sắc, kiểu chữ và bố cục, giúp người dùng dễ dàng làm quen và sử dụng ứng dụng. Giao diện cũng được tối ưu hóa cho các thiết bị di động với các kích thước màn hình khác nhau, đảm bảo trải nghiệm người dùng tốt nhất trên mọi thiết bị.

Sau đây, chúng tôi sẽ trình bày hình ảnh giao diện của ứng dụng trên điện thoại, bao gồm các màn hình bắt đầu, đăng ký, đăng nhập, trang chủ, trình phát nhạc, tìm kiếm, hồ sơ người dùng và chatbot hỗ trợ người dùng.

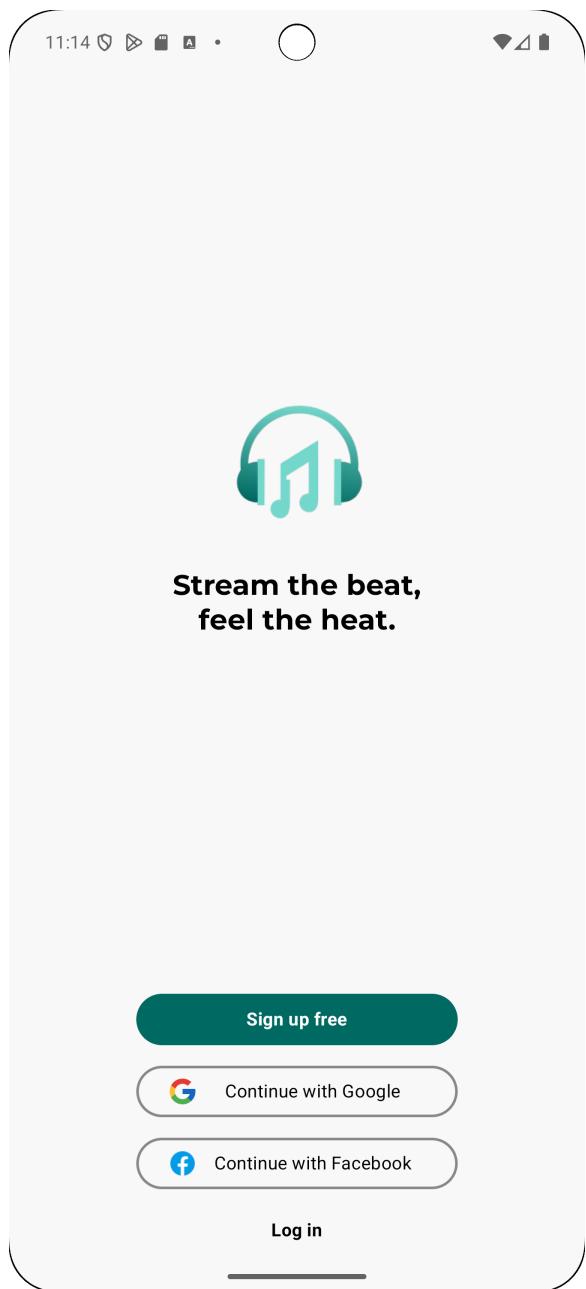
6.1.1. Bắt đầu

6.1.1.1. Mô tả chức năng

Màn hình bắt đầu là màn hình người dùng sẽ thấy khi lần đầu mở ứng dụng. Tại đây, người dùng sẽ nhìn thấy logo của ứng dụng, cùng với các nút bấm để đăng kí, đăng nhập bằng Google, đăng nhập bằng Facebook hoặc đăng nhập thông thường.

6.1.1.2. Thành phần chính

- **Logo ứng dụng:** Hiển thị logo của ứng dụng ở vị trí trung tâm của màn hình.
- **Slogan:** Slogan của ứng dụng được đặt ngay dưới logo, nhằm truyền tải thông điệp của chúng tôi đến người dùng.
- **Nút đăng kí:** Nút này cho phép người dùng chuyển đến màn hình đăng kí tài khoản mới.
- **Nút đăng nhập bằng Google:** Nút này cho phép người dùng đăng nhập nhanh bằng tài khoản Google của họ.
- **Nút đăng nhập bằng Facebook:** Nút này cho phép người dùng đăng nhập nhanh bằng tài khoản Facebook của họ (đang trong quá trình phát triển).
- **Nút đăng nhập thông thường:** Nút này cho phép người dùng chuyển đến màn hình đăng nhập thông thường với tên đăng nhập và mật khẩu mà người dùng đã tạo ở bước đăng kí.



Hình 6.1. Giao diện bắt đầu ứng dụng

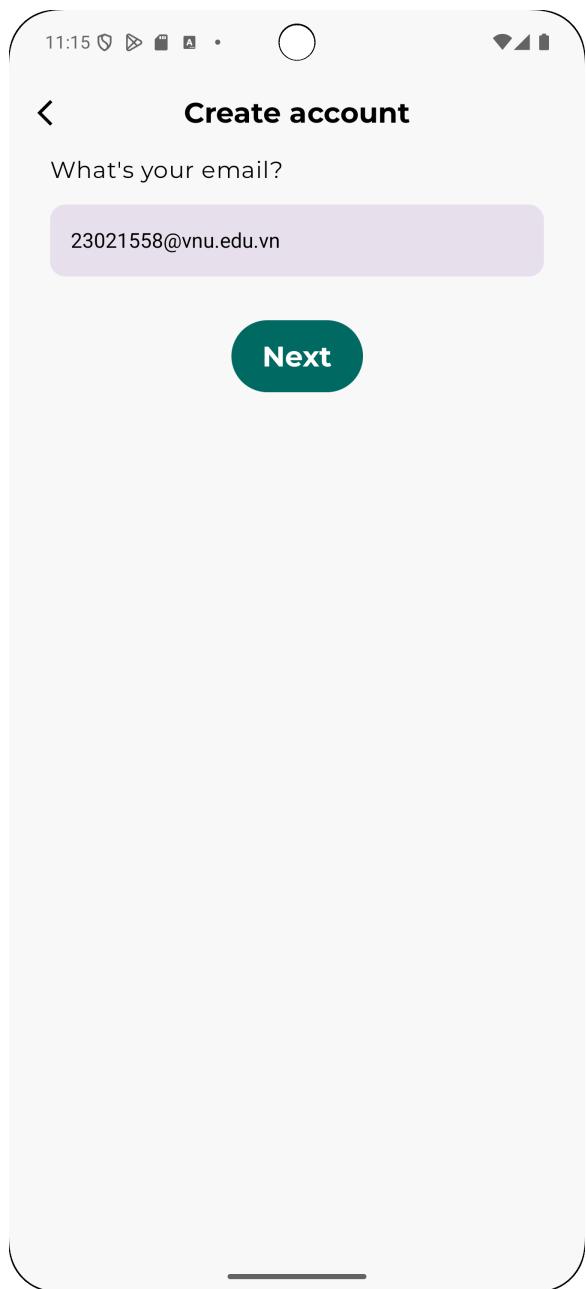
6.1.2. Đăng ký tài khoản

6.1.2.1. Mô tả chức năng

Các màn hình đăng ký tài khoản cho phép người dùng tạo tài khoản mới để sử dụng ứng dụng. Quá trình đăng ký bao gồm các bước nhập email, tạo mật khẩu và tạo tên người dùng. Mỗi bước đều có giao diện riêng biệt để người dùng dễ dàng thực hiện, người dùng có thể tiếp tục sang bước tiếp theo bằng nút "Next" hoặc quay lại bước trước đó bằng cách ấn biểu tượng mũi tên quay lại ở góc trên trái màn hình.

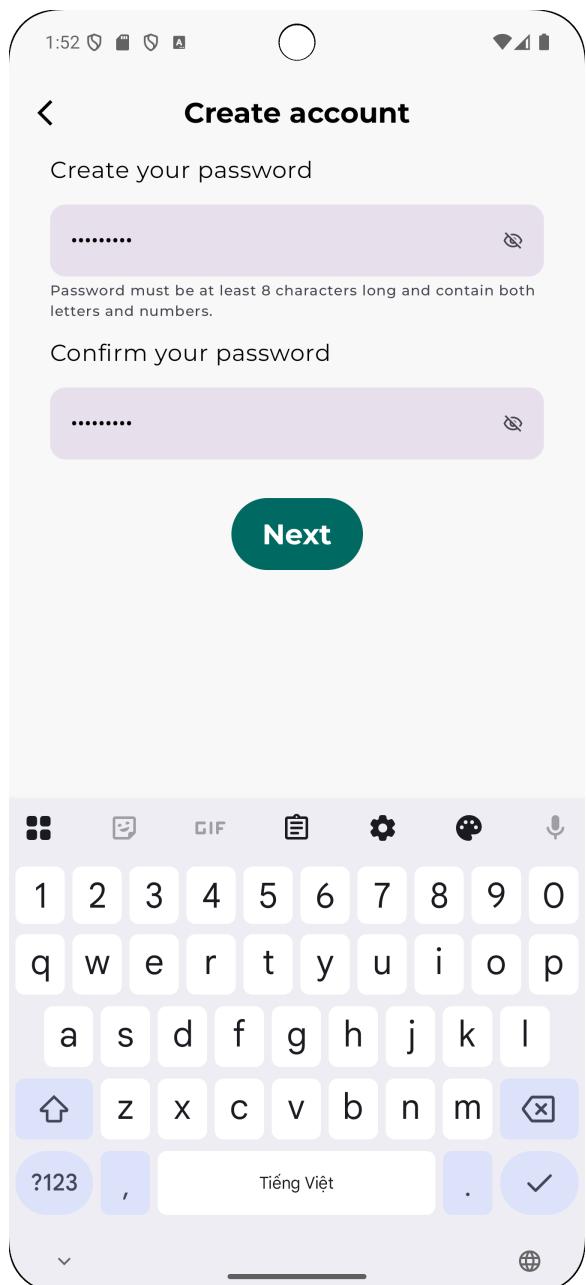
6.1.2.2. Thành phần chính

- **Trường nhập email:** Người dùng nhập địa chỉ email hợp lệ để đăng ký tài khoản mới.
- **Trường nhập mật khẩu:** Người dùng tạo mật khẩu cho tài khoản của mình, bao gồm hai trường nhập mật khẩu và xác nhận mật khẩu với yêu cầu mật khẩu phải có từ 8 ký tự và bao gồm cả chữ và chữ số.
- **Trường nhập tên người dùng:** Người dùng đặt tên cho tài khoản của mình với họ và tên.
- **Nút Next:** Nút này cho phép người dùng tiếp tục sang bước tiếp theo trong quá trình đăng ký tài khoản.
- **Biểu tượng quay lại:** Cho phép người dùng quay lại bước trước đó trong quá trình đăng ký tài khoản.



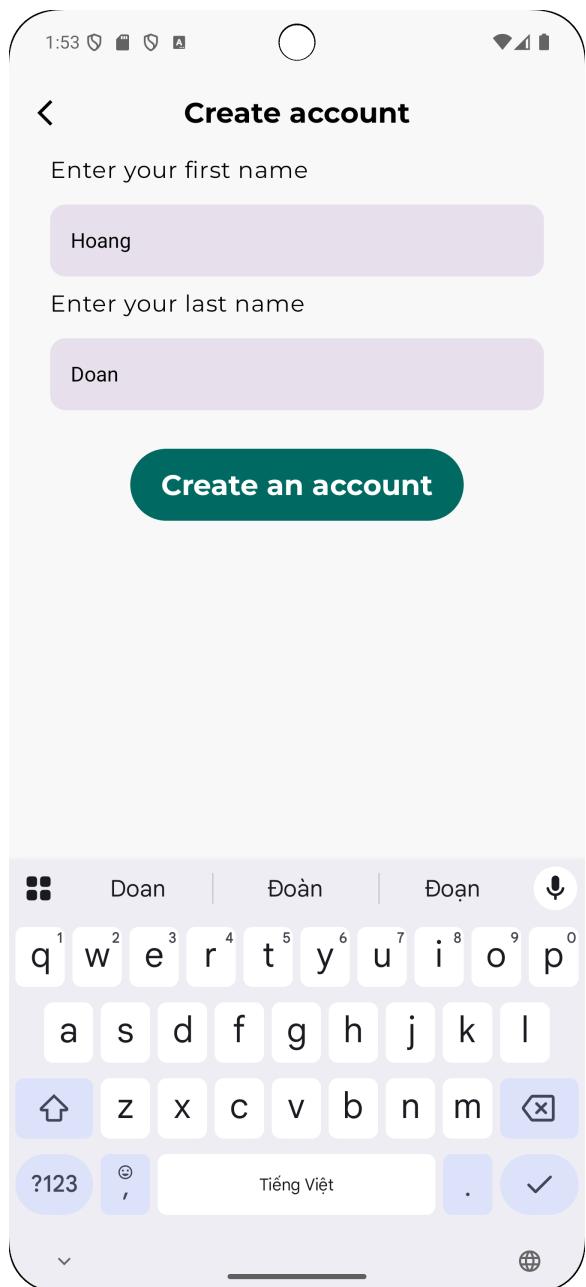
Hình 6.2. Giao diện đăng ký tài khoản - nhập email

Người dùng nhập địa chỉ email hợp lệ để đăng ký tài khoản mới.



Hình 6.3. Giao diện đăng ký tài khoản - tạo mật khẩu

Người dùng tạo mật khẩu cho tài khoản của mình, gồm hai bước nhập mật khẩu và xác nhận mật khẩu với yêu cầu mật khẩu phải có từ 8 kí tự và bao gồm cả chữ và chữ số.



Hình 6.4. Giao diện đăng ký tài khoản - tạo tên người dùng

Người dùng đặt tên cho tài khoản của mình với họ và tên, người dùng có thể đổi họ và tên của tài khoản sau khi đăng nhập vào ứng dụng sau.

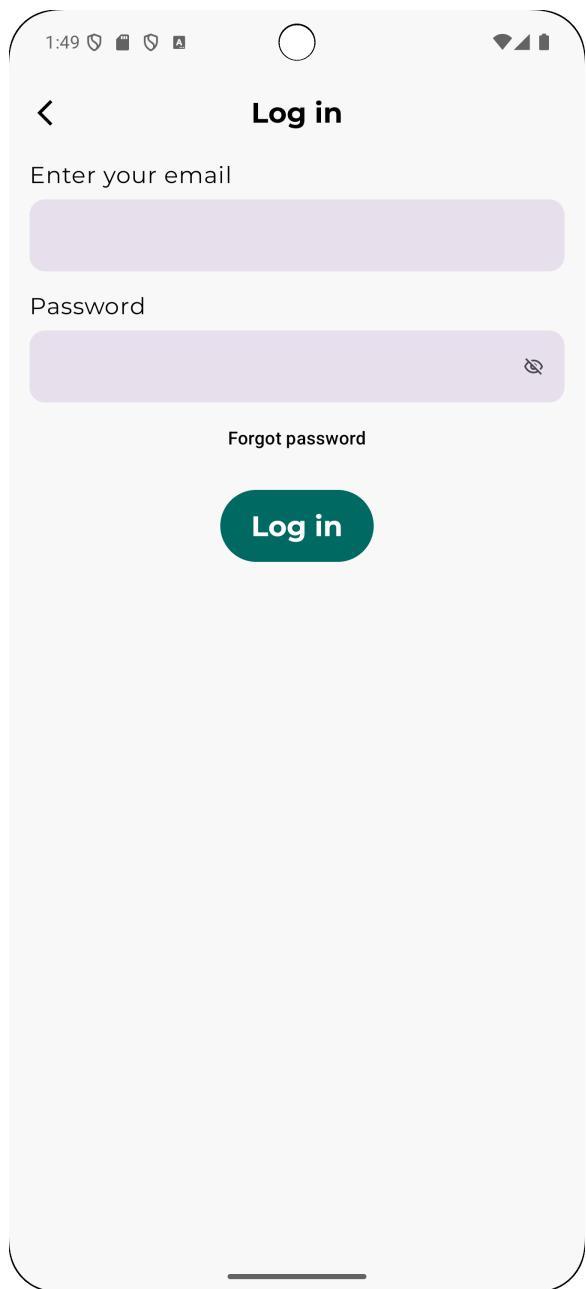
6.1.3. Đăng nhập

6.1.3.1. Mô tả chức năng

Màn hình đăng nhập cho phép người dùng đăng nhập vào tài khoản của họ bằng cách nhập email và mật khẩu đã đăng ký trước đó. Ngoài ra, nếu người dùng quên mật khẩu, họ có thể chọn tùy chọn "Quên mật khẩu" để đặt lại mật khẩu thông qua email.

6.1.3.2. Thành phần chính

- **Trường nhập email:** Người dùng nhập địa chỉ email đã đăng ký để đăng nhập vào tài khoản của mình.
- **Trường nhập mật khẩu:** Người dùng nhập mật khẩu tương ứng với tài khoản của họ.
- **Nút đăng nhập:** Nút này cho phép người dùng đăng nhập vào tài khoản của mình sau khi đã nhập đúng email và mật khẩu.
- **Tùy chọn quên mật khẩu:** Nếu người dùng quên mật khẩu, họ có thể chọn tùy chọn này để bắt đầu quá trình đặt lại mật khẩu thông qua email.
- **Biểu tượng quay lại:** Cho phép người dùng quay lại màn hình bắt đầu của ứng dụng.



Hình 6.5. Giao diện đăng nhập

Người dùng nhập email và mật khẩu đã đăng ký để đăng nhập vào ứng dụng. Nếu quên mật khẩu, người dùng có thể chọn tùy chọn "Quên mật khẩu" để đặt lại mật khẩu.

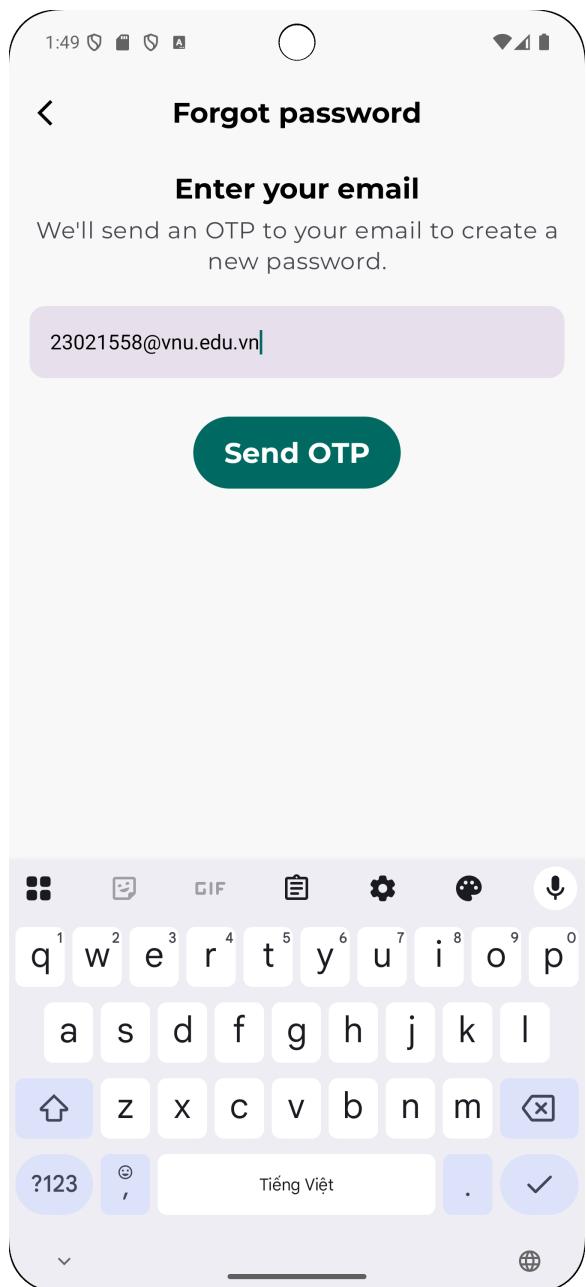
6.1.4. Quên mật khẩu và đặt lại

6.1.4.1. Mô tả chức năng

Các màn hình sau cho phép người dùng đặt lại mật khẩu khi họ quên mật khẩu đăng nhập. Quá trình đặt lại mật khẩu bao gồm ba bước: nhập email đã đăng ký, nhập mã OTP được gửi từ chúng tôi qua email đã đăng ký và tạo lại mật khẩu mới. Mỗi bước đều có giao diện riêng biệt để người dùng dễ dàng thực hiện.

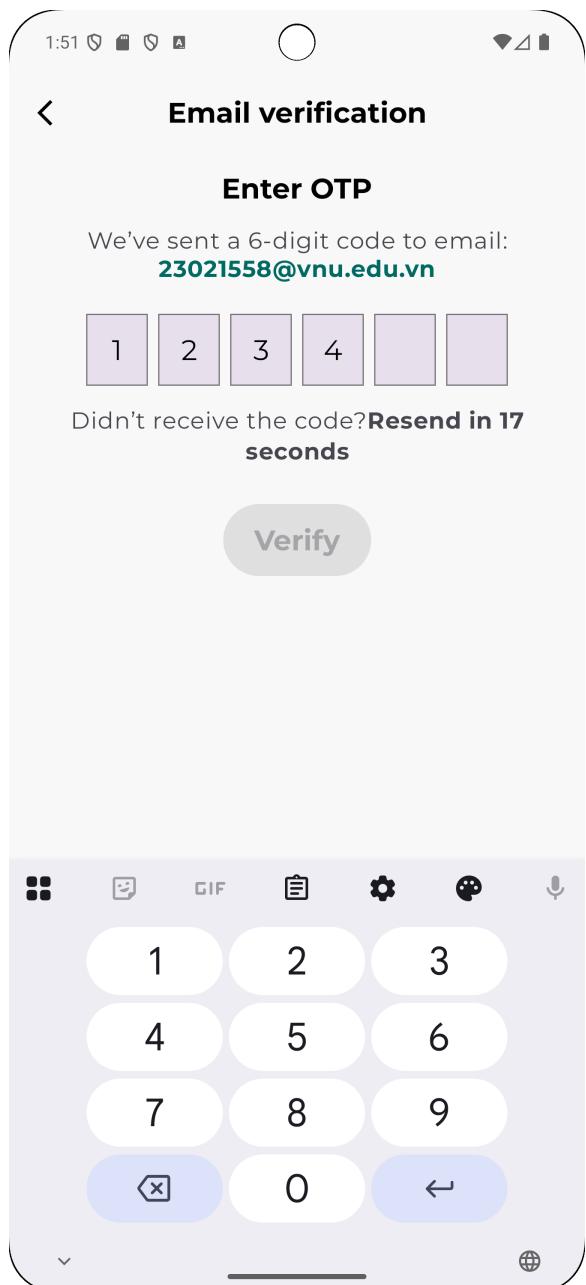
6.1.4.2. Thành phần chính

- **Trường nhập email:** Người dùng nhập địa chỉ email đã đăng ký để nhận mã OTP.
- **Nút gửi mã OTP:** Cho phép người dùng gửi yêu cầu nhận mã OTP qua email đã đăng ký.
- **Trường nhập mã OTP:** Người dùng nhập mã OTP đã nhận được qua email để xác thực và tiếp tục quá trình tạo mật khẩu mới.
- **Nút gửi lại mã:** Nếu người dùng không nhận được mã OTP, họ có thể ấn nút này để yêu cầu gửi lại mã.
- **Nút xác nhận mã OTP:** Cho phép người dùng xác nhận mã OTP đã nhập để tiếp tục sang bước tạo mật khẩu mới.
- **Trường nhập mật khẩu mới:** Người dùng tạo mật khẩu mới cho tài khoản của mình, bao gồm hai trường nhập mật khẩu và xác nhận mật khẩu với yêu cầu mật khẩu phải có từ 8 kí tự và bao gồm cả chữ và chữ số.
- **Nút đặt lại mật khẩu:** Nút này cho phép người dùng hoàn tất quá trình đặt lại mật khẩu sau khi đã nhập đúng mã OTP và mật khẩu mới.
- **Biểu tượng quay lại:** Các màn hình đều có biểu tượng quay lại cho phép người dùng quay lại bước trước đó.



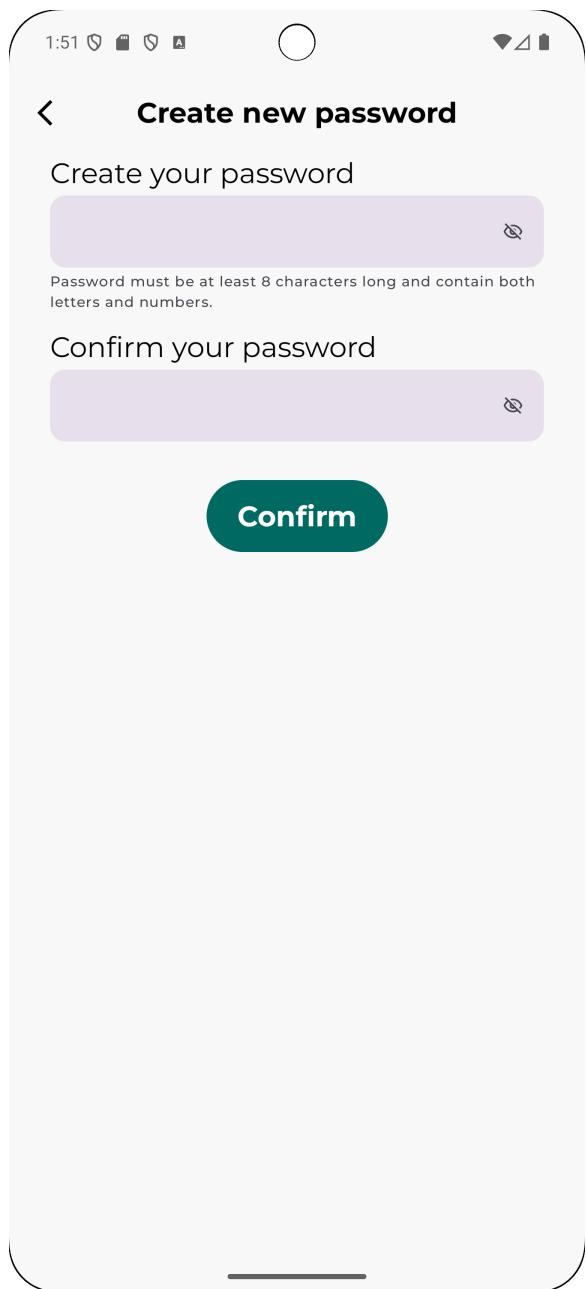
Hình 6.6. Giao diện quên mật khẩu - nhập email

Người dùng nhập email đã đăng ký, chúng tôi sẽ gửi mã OTP về email của người dùng để tạo lại mật khẩu mới.



Hình 6.7. Giao diện quên mật khẩu - nhập mã OTP

Người dùng nhập mã OTP đã nhận được qua email để xác thực và tiếp tục quá trình tạo mật khẩu mới, nếu không nhận được mã OTP, người dùng có thể ấn nút gửi lại mã.



Hình 6.8. Giao diện quên mật khẩu - tạo mật khẩu mới

Người dùng tạo mật khẩu mới cho tài khoản của mình, gồm hai bước nhập mật khẩu và xác nhận mật khẩu với yêu cầu mật khẩu phải có từ 8 ký tự và bao gồm cả chữ và chữ số.

6.1.5. Trang chủ

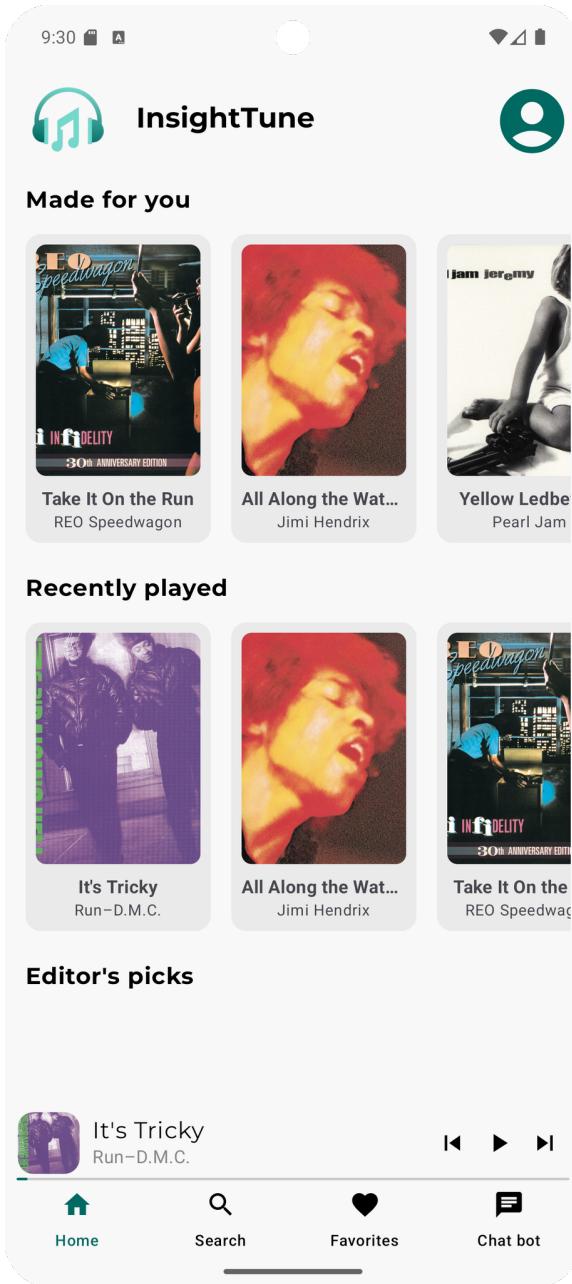
6.1.5.1. Mô tả chức năng

Màn hình chính là nơi người dùng có thể khám phá và truy cập nhanh vào các bài hát được gợi ý dựa trên thói quen nghe nhạc của họ, cũng như các bài hát mới nghe gần đây và các gợi ý từ nhà phát triển. Từ màn hình chính, người dùng cũng có thể dễ dàng

truy cập vào thông tin cá nhân qua biểu tượng người dùng và chuyển sang các tab khác như tìm kiếm, danh sách các bài hát yêu thích và chatbot hỗ trợ thông qua thanh điều hướng bên dưới.

6.1.5.2. Thành phần chính

- **Logo và tên ứng dụng:** Hiển thị logo và tên của ứng dụng ở phía trên cùng của màn hình.
- **Biểu tượng người dùng:** Nằm ở góc trên bên phải màn hình, cho phép người dùng truy cập vào thông tin cá nhân và cài đặt tài khoản.
- **Khu vực gợi ý bài hát:** Hiển thị các bài hát được gợi ý dựa trên thói quen nghe nhạc của người dùng.
- **Khu vực bài hát mới nghe gần đây:** Hiển thị các bài hát mà người dùng đã nghe gần đây để họ có thể dễ dàng tiếp tục thưởng thức.
- **Khu vực bài hát gợi ý từ nhà phát triển:** Hiển thị các bài hát được lựa chọn và gợi ý bởi nhà phát triển ứng dụng.
- **Thanh nhạc đang phát:** Hiển thị bài hát hiện tại đang được phát, cho phép người dùng nhanh chóng truy cập vào trình phát nhạc bằng cách nhấn vào thanh này.
- **Thanh điều hướng:** Cho phép người dùng chuyển đổi nhanh giữa các tab chính của ứng dụng, bao gồm Trang chủ, Tìm kiếm, Yêu thích và Chatbot



Hình 6.9. Giao diện màn hình chính

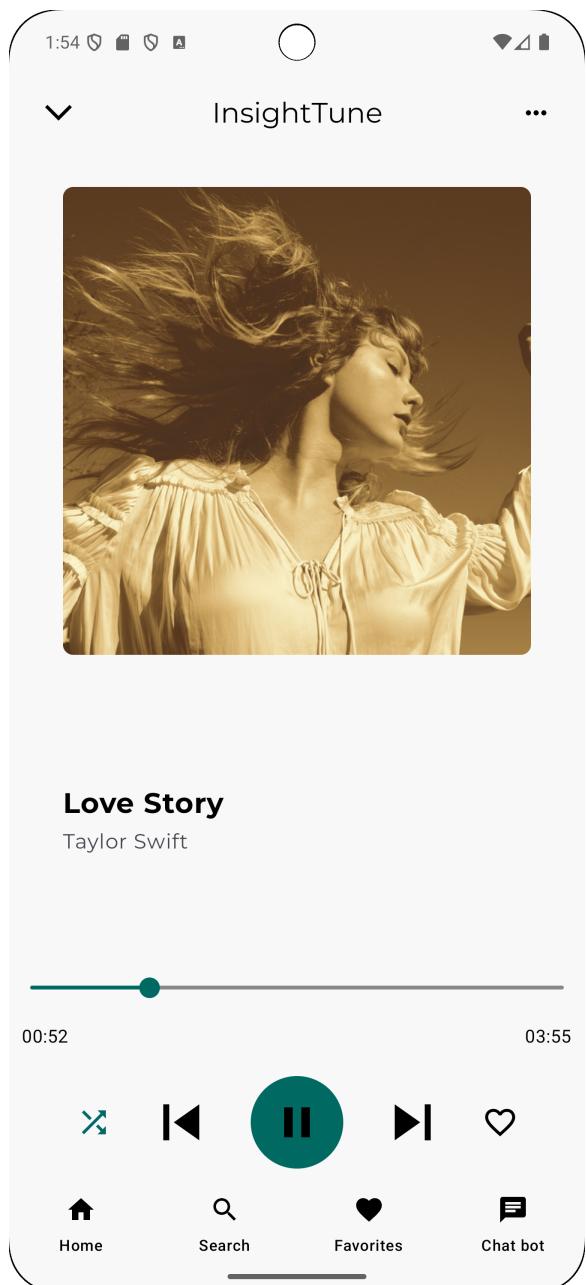
6.1.6. Trình phát nhạc

6.1.6.1. Mô tả chức năng

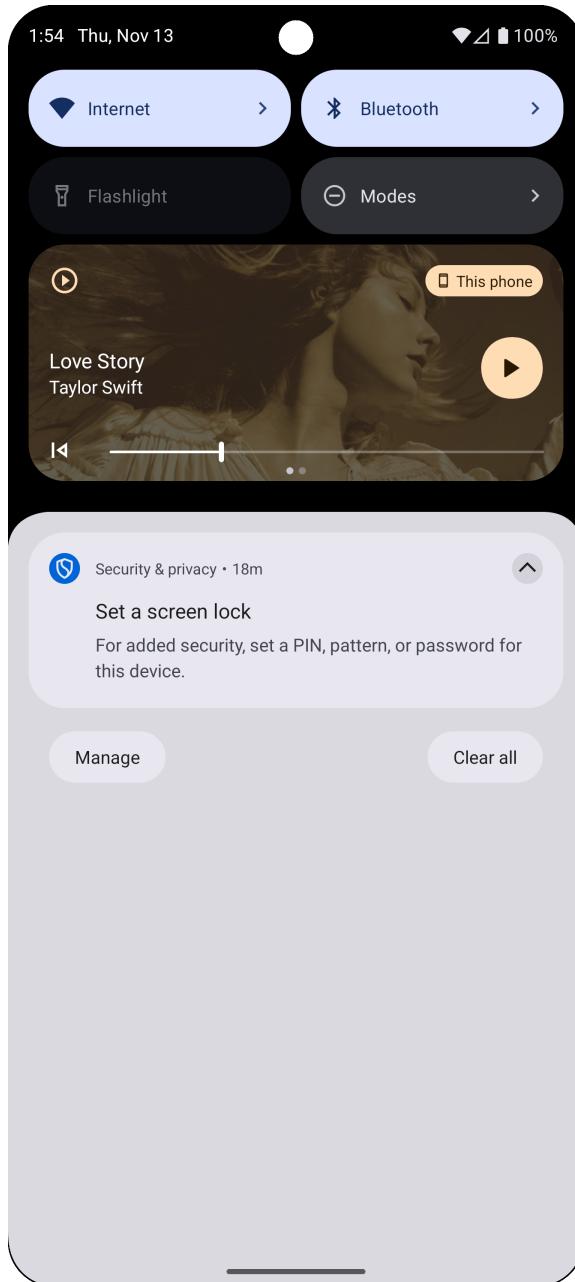
Đây là màn hình phát nhạc chính của ứng dụng, nơi người dùng có thể điều khiển việc phát nhạc và xem thông tin chi tiết về bài hát đang được phát. Người dùng có thể xem được tên bài hát, tên nghệ sĩ thể hiện cũng như ảnh bìa của bài hát. Ngoài ra, người dùng có thể sử dụng các chức năng điều khiển phát nhạc như phát/tạm dừng, chuyển bài tiếp, phát lại bài trước, tua thời gian bài hát, thêm vào mục yêu thích, chức năng phát ngẫu nhiên hoặc phát lặp lại.

6.1.6.2. Thành phần chính

- **Ảnh bìa bài hát:** Hiển thị ảnh bìa của bài hát đang được phát.
- **Tên bài hát và nghệ sĩ:** Hiển thị tên bài hát và tên nghệ sĩ thể hiện ngay dưới ảnh bìa.
- **Thanh tiến trình bài hát:** Hiển thị tiến trình phát của bài hát, cho phép người dùng tua nhanh hoặc tua lại bằng cách nhấn một vị trí thời gian mong muốn trên thanh này.
- **Nút chơi nhạc:** Nút này cho phép người dùng phát hoặc tạm dừng bài hát hiện tại.
- **Nút chuyển bài tiếp:** Bao gồm nút chuyển sang bài hát tiếp theo.
- **Nút chuyển bài trước:** Bao gồm nút quay lại bài hát trước đó.
- **Nút phát ngẫu nhiên:** Cho phép người dùng bật qua lại giữa chế độ phát ngẫu nhiên và phát lặp lại, với nút sáng màu tức đang phát nhạc ngẫu nhiên.
- **Nút thêm vào yêu thích:** Cho phép người dùng thêm bài hát hiện tại vào danh sách các bài hát yêu thích của họ.
- **Nút thu nhỏ:** Cho phép người dùng thu nhỏ trình phát nhạc về thanh nhạc đang phát ở dưới để truy cập vào các chức năng khác của ứng dụng.



Hình 6.10. Giao diện trình phát nhạc



Hình 6.11. Giao diện phát nhạc trên thanh thông báo

Người dùng có thể tiếp tục nghe nhạc khi ứng dụng đang chạy nền cũng như tạm dừng hoặc phát nhạc.

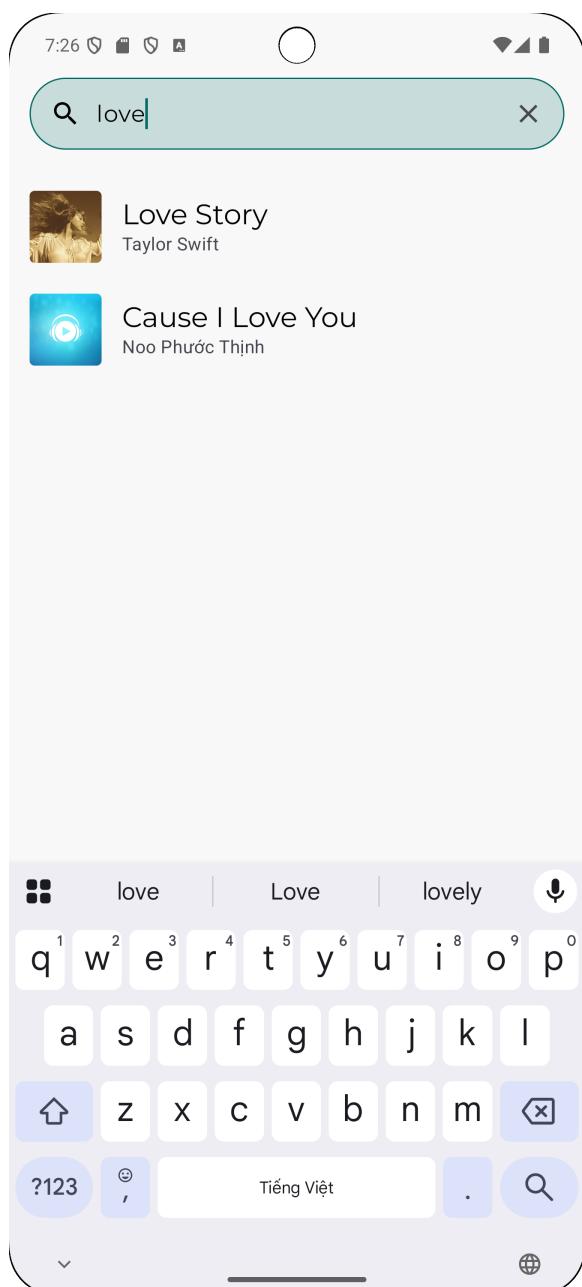
6.1.7. Tìm kiếm

6.1.7.1. Mô tả chức năng

Màn hình tìm kiếm cho phép người dùng tìm kiếm bài hát theo tên bài hát hoặc theo tên các nghệ sĩ thể hiện. Kết quả tìm kiếm sẽ hiển thị danh sách các bài hát phù hợp với từ khóa tìm kiếm mà người dùng đã nhập. Người dùng có thể phát nhạc trực tiếp từ các kết quả tìm kiếm này.

6.1.7.2. Thành phần chính

- **Thanh tìm kiếm:** Người dùng nhập từ khóa tên bài hát hoặc tên nghệ sĩ để tìm kiếm bài hát mong muốn.
- **Kết quả tìm kiếm:** Hiển thị danh sách các bài hát phù hợp với từ khóa tìm kiếm mà người dùng đã nhập. Mỗi mục trong danh sách bao gồm tên bài hát, tên nghệ sĩ và ảnh bìa của bài hát.



Hình 6.12. Giao diện tìm kiếm bài hát

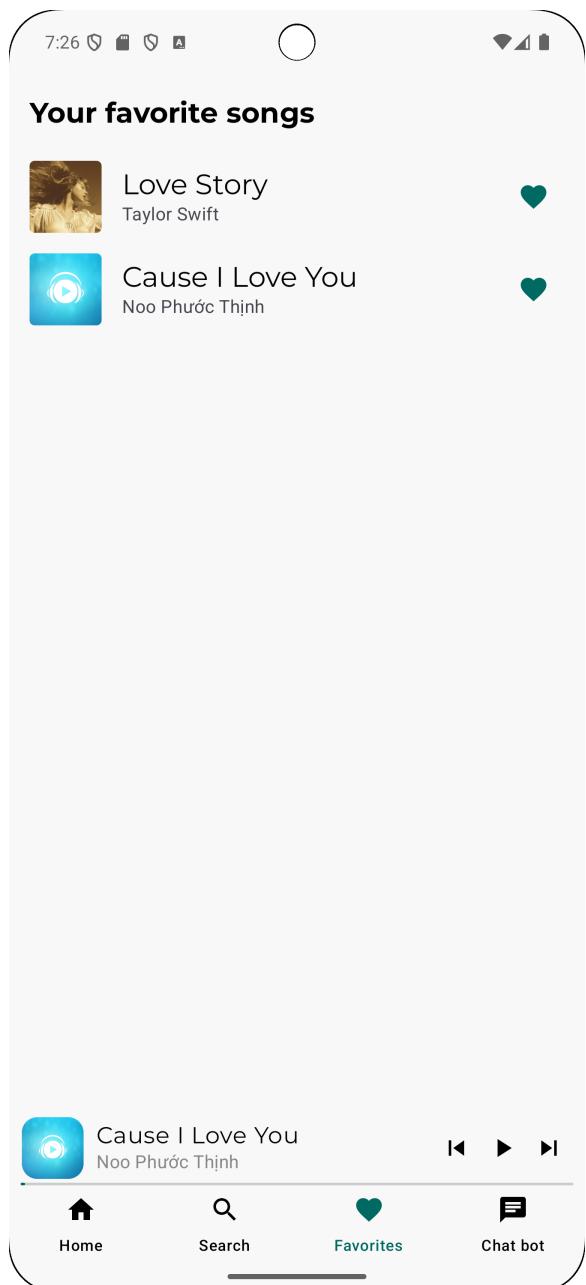
6.1.8. Quản lý các bài hát yêu thích

6.1.8.1. Mô tả chức năng

Màn hình các bài hát yêu thích hiển thị danh sách các bài hát mà người dùng đã đánh dấu yêu thích. Người dùng có thể truy cập và phát các bài hát yêu thích từ danh sách này. Người dùng cũng có thể xóa bài hát khỏi danh sách yêu thích bằng cách nhấn vào biểu tượng trái tim.

6.1.8.2. Thành phần chính

- **Danh sách bài hát yêu thích:** Hiển thị các bài hát mà người dùng đã đánh dấu yêu thích. Mỗi mục trong danh sách bao gồm tên bài hát, tên nghệ sĩ và ảnh bìa của bài hát.
- **Biểu tượng trái tim:** Cho phép người dùng xóa bài hát khỏi danh sách yêu thích bằng cách nhấn vào biểu tượng này.



Hình 6.13. Giao diện các bài hát yêu thích

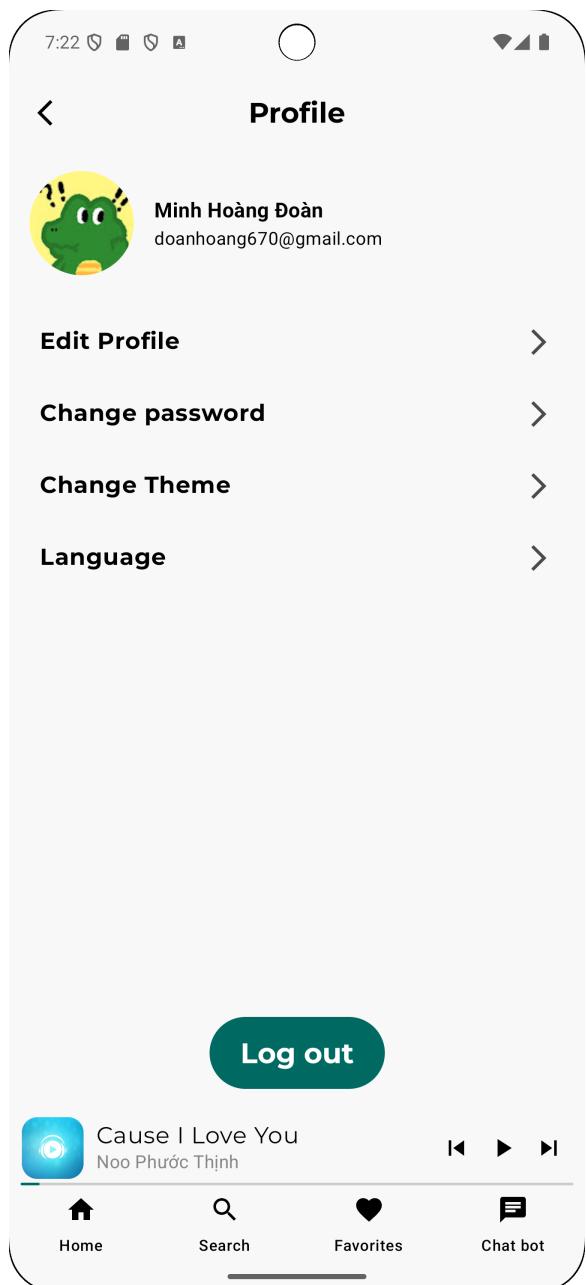
6.1.9. Tài khoản và cá nhân hóa

6.1.9.1. Mô tả chức năng

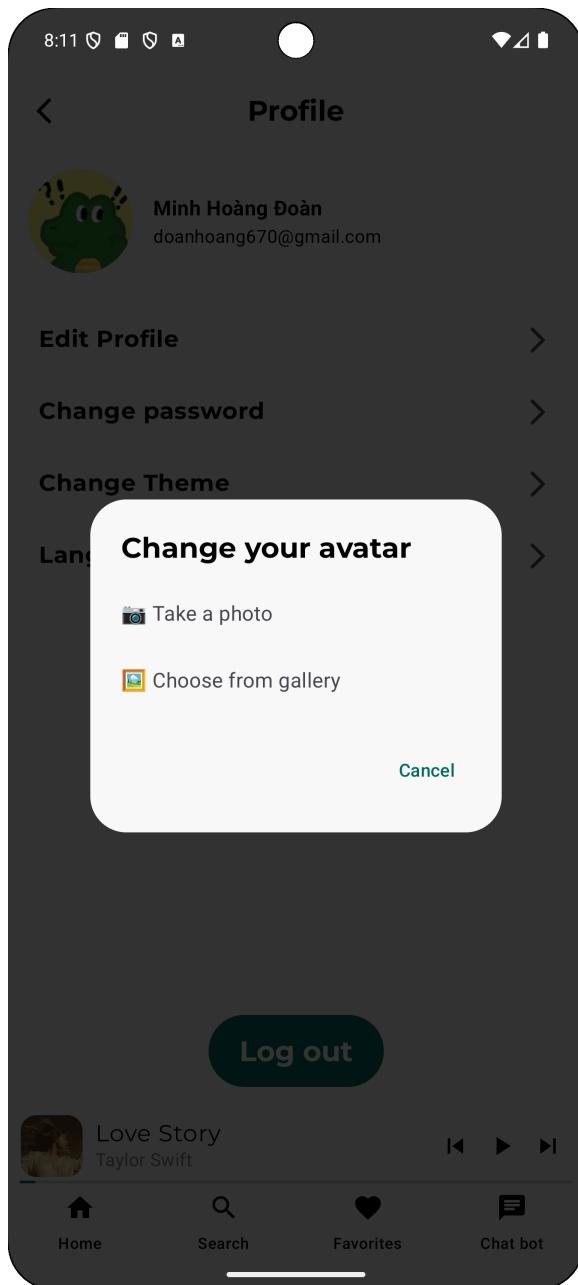
Màn hình thông tin tài khoản hiển thị các thông tin cá nhân của người dùng bao gồm ảnh đại diện, tên người dùng và email. Người dùng có thể chỉnh sửa thông tin cá nhân, thay đổi mật khẩu, thay đổi giao diện và ngôn ngữ của ứng dụng. Ngoài ra, người dùng cũng có thể đăng xuất khỏi tài khoản.

6.1.9.2. Thành phần chính

- **Ảnh đại diện:** Hiển thị ảnh đại diện hiện tại của người dùng, người dùng có thể nhấp vào để thay đổi ảnh đại diện.
- **Tên người dùng:** Hiển thị tên người dùng hiện tại.
- **Email:** Hiển thị địa chỉ email đã đăng ký của người dùng.
- **Nút chỉnh sửa thông tin cá nhân:** Cho phép người dùng cập nhật thông tin cá nhân như họ và tên, số điện thoại và địa chỉ.
- **Nút thay đổi mật khẩu:** Cho phép người dùng thay đổi mật khẩu hiện tại của mình.
- **Nút thay đổi giao diện:** Cho phép người dùng thay đổi giao diện của ứng dụng giữa các chế độ sáng, tối hoặc theo chế độ của hệ thống.
- **Nút thay đổi ngôn ngữ:** Cho phép người dùng thay đổi ngôn ngữ hiển thị của ứng dụng giữa tiếng Việt và tiếng Anh.
- **Nút đăng xuất:** Cho phép người dùng đăng xuất khỏi tài khoản hiện tại và quay lại màn hình bắt đầu.
- **Biểu tượng quay lại:** Cho phép người dùng quay lại màn hình chính.

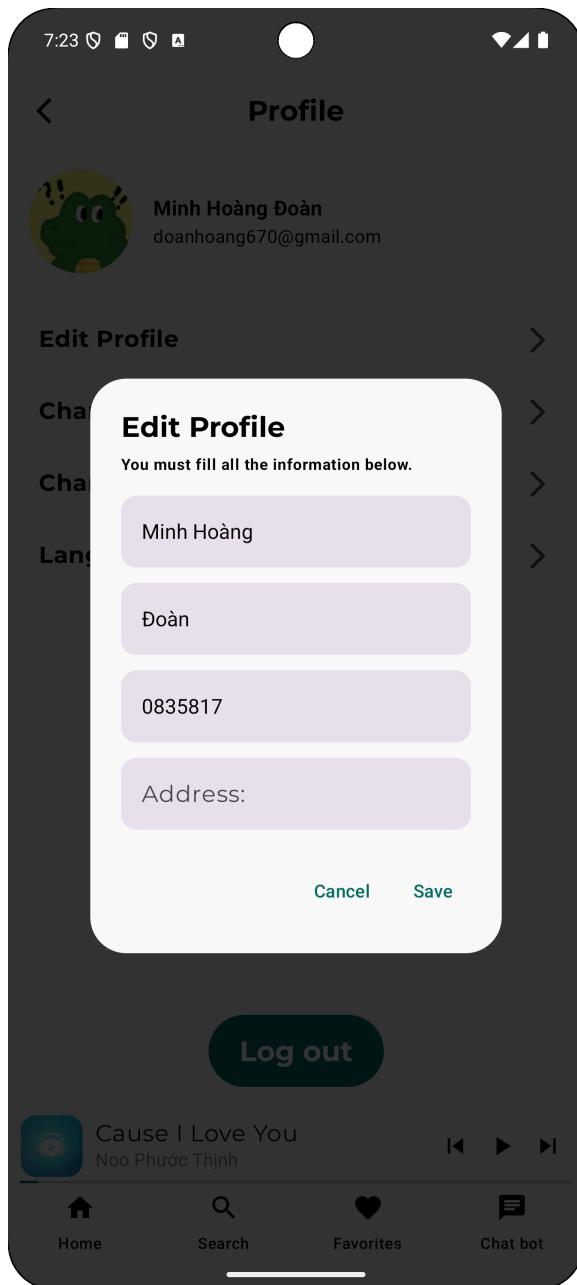


Hình 6.14. Giao diện thông tin tài khoản



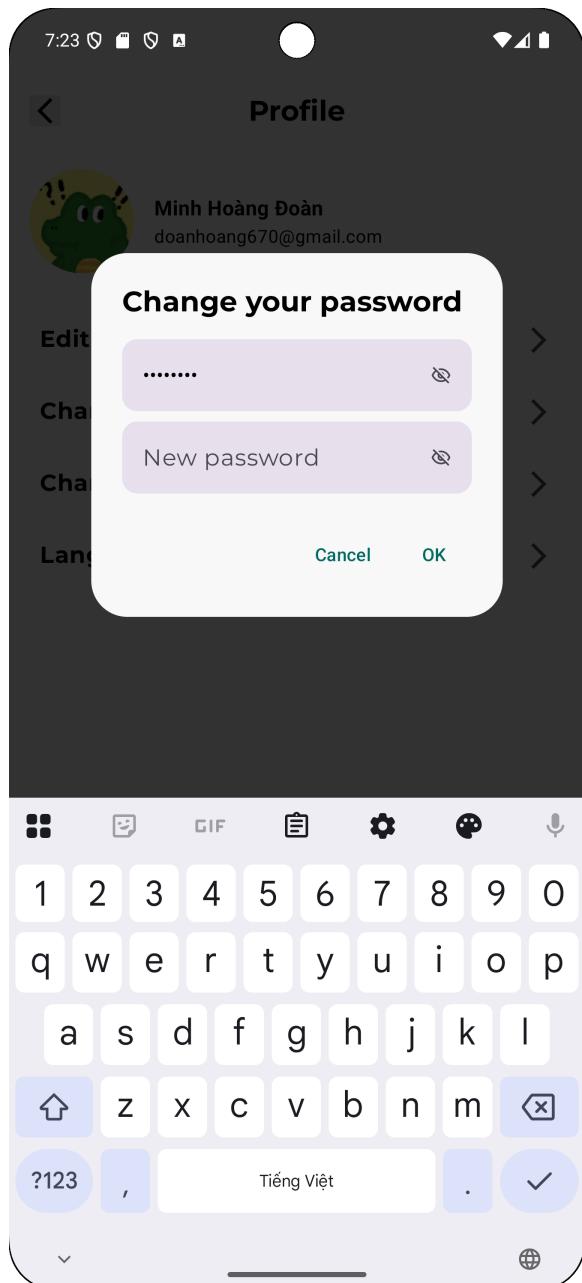
Hình 6.15. Giao diện đổi ảnh đại diện

Giao diện cho phép người dùng thay đổi ảnh đại diện của mình bằng cách chọn ảnh từ thư viện ảnh của điện thoại hoặc chụp ảnh mới trực tiếp bằng camera.



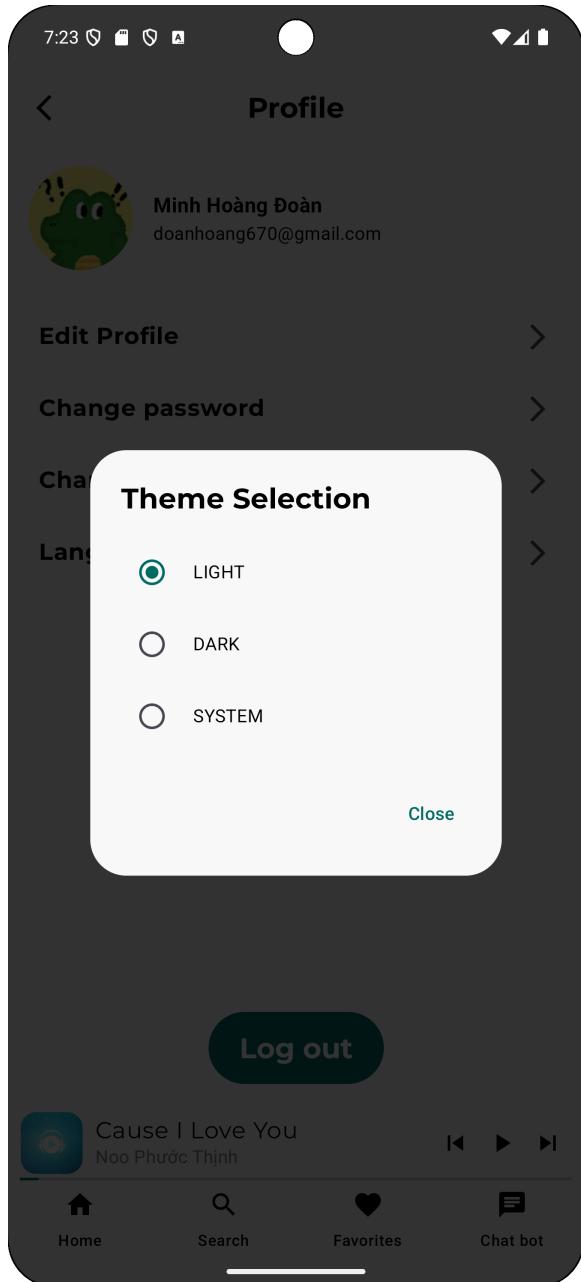
Hình 6.16. Giao diện cập nhật thông tin cá nhân

Giao diện cho phép người dùng cập nhật thông tin cá nhân của mình bao gồm họ và tên, số điện thoại cũng như địa chỉ của mình. Người dùng có thể lưu lại thông tin mới hoặc hủy.



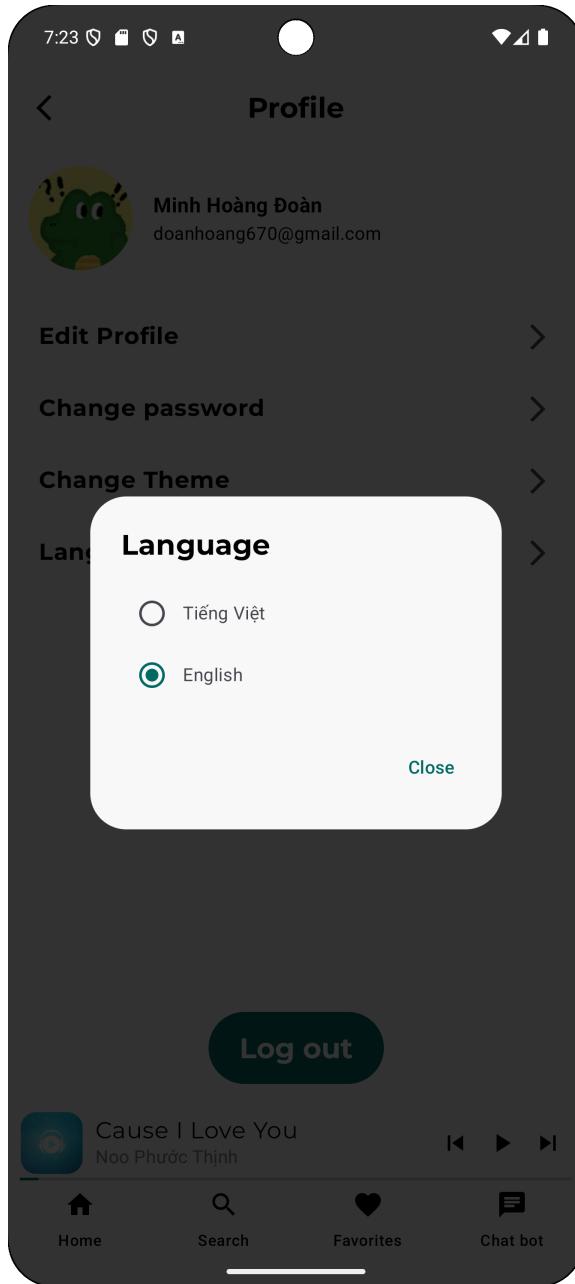
Hình 6.17. Giao diện thay đổi mật khẩu

Giao diện cho phép người dùng thay đổi mật khẩu của mình bằng cách nhập mật khẩu hiện tại và mật khẩu mới, yêu cầu mật khẩu hiện tại phải trùng khớp. Người dùng có thể lưu lại mật khẩu mới hoặc hủy.



Hình 6.18. Giao diện thay đổi giao diện

Giao diện cho phép người dùng thay đổi giao diện của ứng dụng giữa các chế độ sáng, tối hoặc theo chế độ của hệ thống. Người dùng có thể chọn một trong các chế độ và áp dụng thay đổi ngay lập tức.



Hình 6.19. Giao diện thay đổi ngôn ngữ

Giao diện cho phép người dùng thay đổi ngôn ngữ hiển thị của ứng dụng giữa các ngôn ngữ được hỗ trợ như tiếng Việt và tiếng Anh. Người dùng có thể chọn ngôn ngữ mong muốn và áp dụng thay đổi ngay lập tức.

6.1.10. Chatbot hỗ trợ

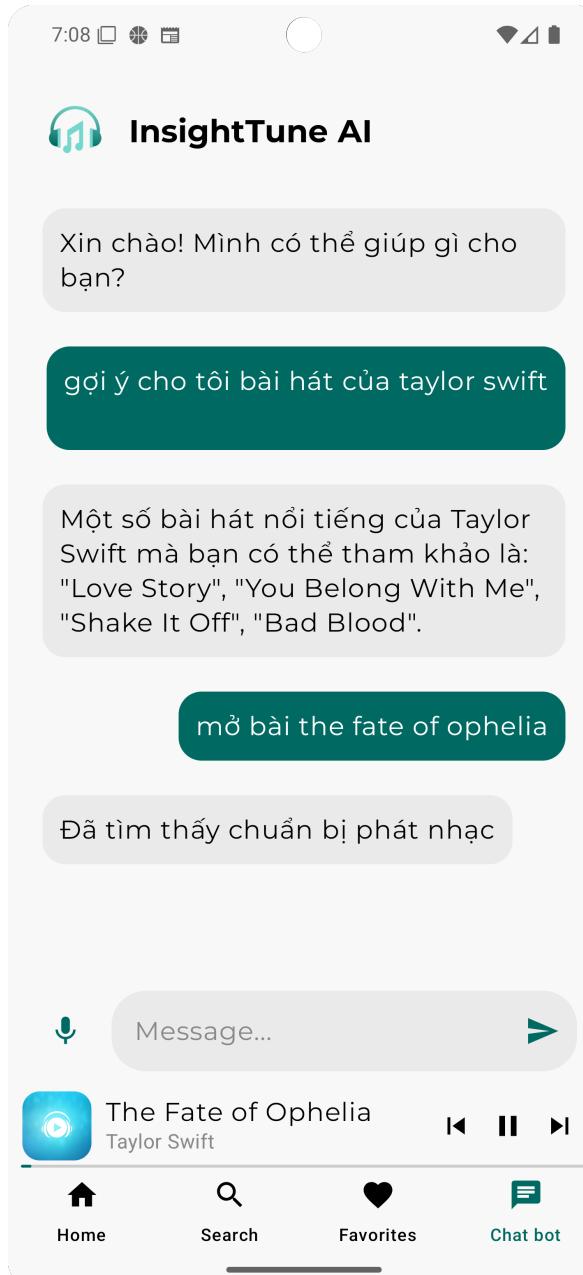
6.1.10.1. Mô tả chức năng

Giao diện chatbot sẽ hỗ trợ người dùng trong việc giải đáp các thắc mắc liên quan đến âm nhạc, tìm kiếm hoặc gợi ý bài hát. Người dùng cũng có thể yêu cầu AI mở một bài hát có sẵn trong thư viện của ứng dụng. Ngoài ra, người dùng cũng có thể nhập liệu

bằng giọng nói để đưa ra yêu cầu với chatbot.

6.1.10.2. Thành phần chính

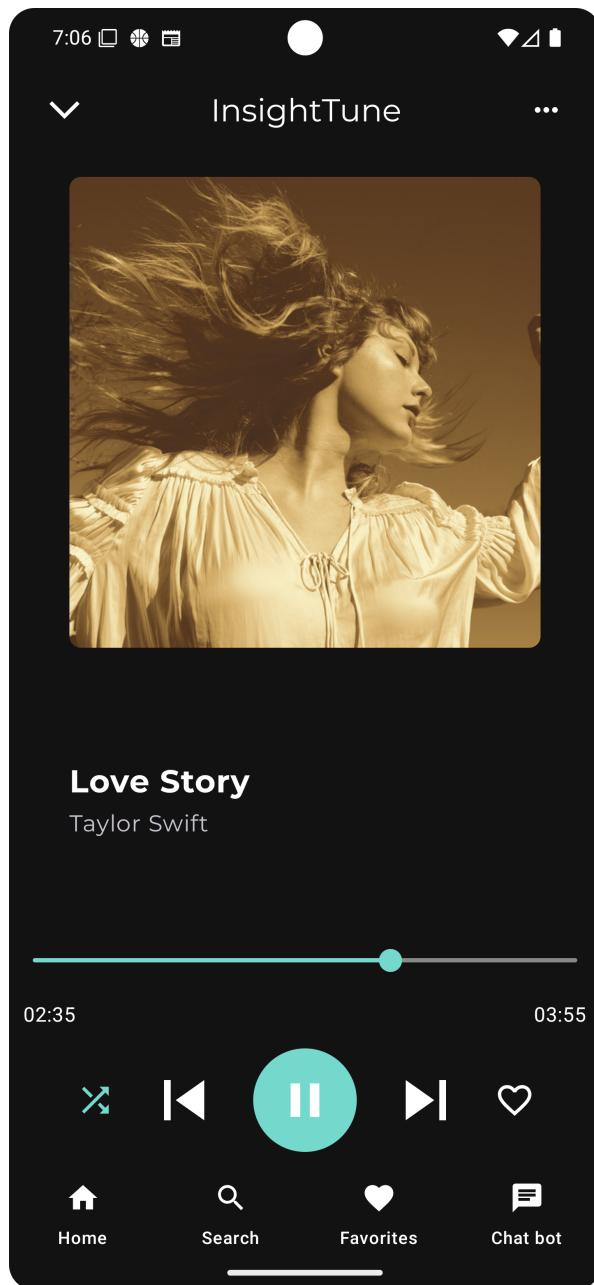
- **Khu vực hiển thị cuộc trò chuyện:** Hiển thị các tin nhắn giữa người dùng và chatbot.
- **Trường nhập tin nhắn:** Người dùng nhập tin nhắn và yêu cầu của mình để gửi đến chatbot.
- **Nút gửi tin nhắn:** Cho phép người dùng gửi tin nhắn đã nhập đến chatbot.
- **Nút ghi âm giọng nói:** Cho phép người dùng ghi âm giọng nói để gửi yêu cầu đến chatbot thay vì nhập liệu bằng văn bản. Người dùng ấn lần đầu để bắt đầu ghi âm và ấn lần nữa để kết thúc ghi âm và gửi yêu cầu.



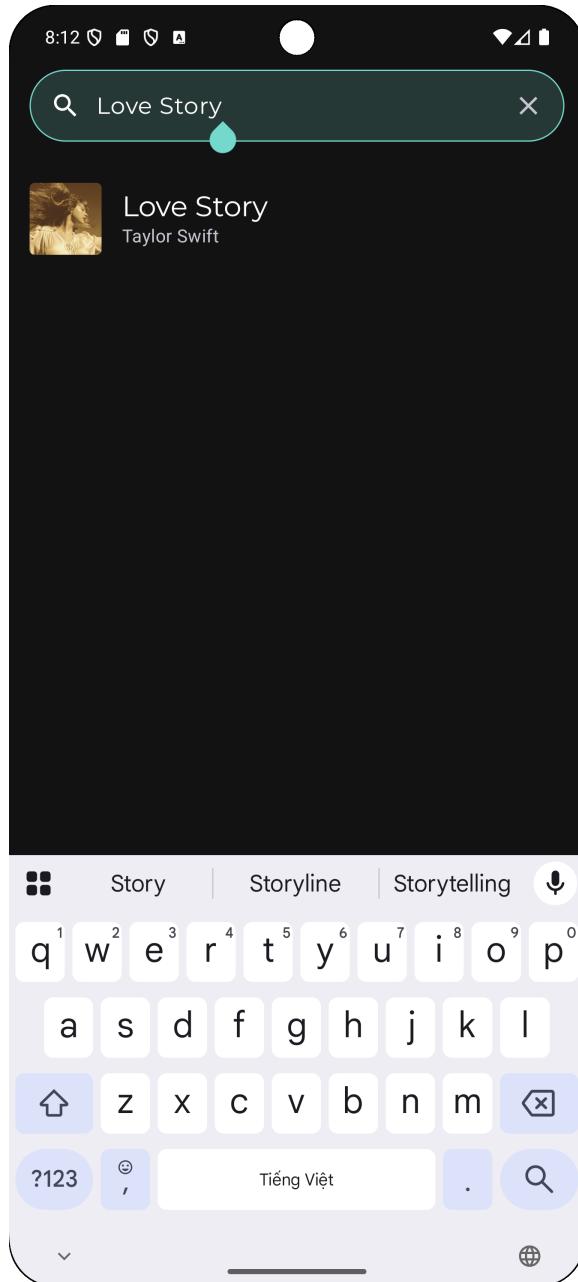
Hình 6.20. Giao diện chatbot hỗ trợ người dùng

6.1.11. Một số hình ảnh của giao diện tối

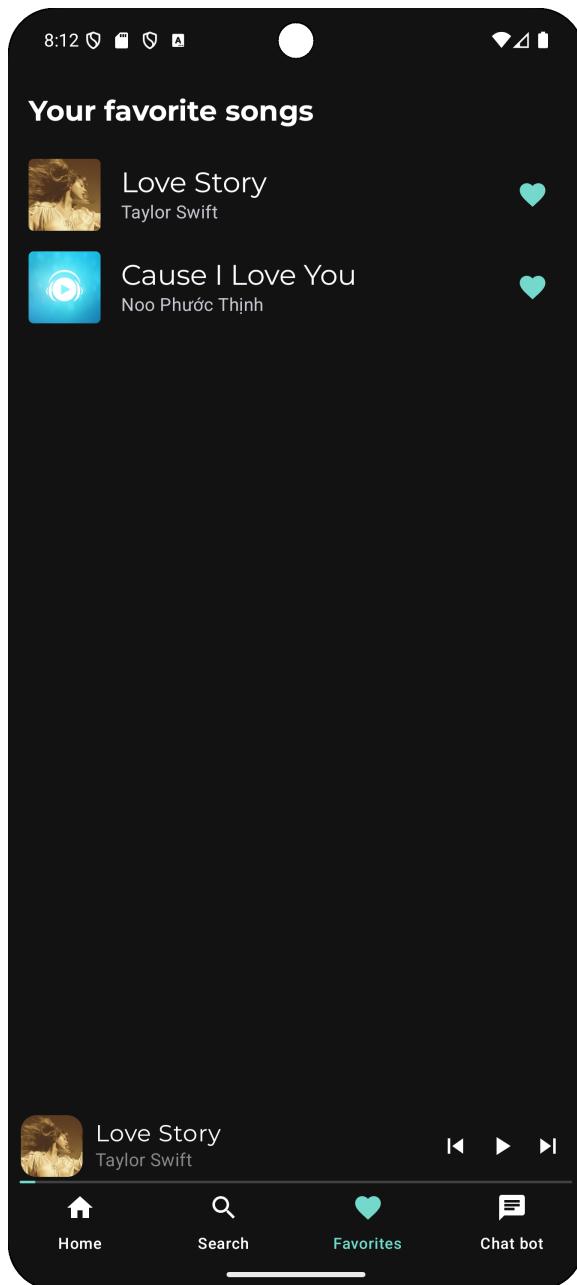
Sau đây là một số hình ảnh minh họa về giao diện tối của ứng dụng nghe nhạc mà chúng tôi đã thiết kế và triển khai:



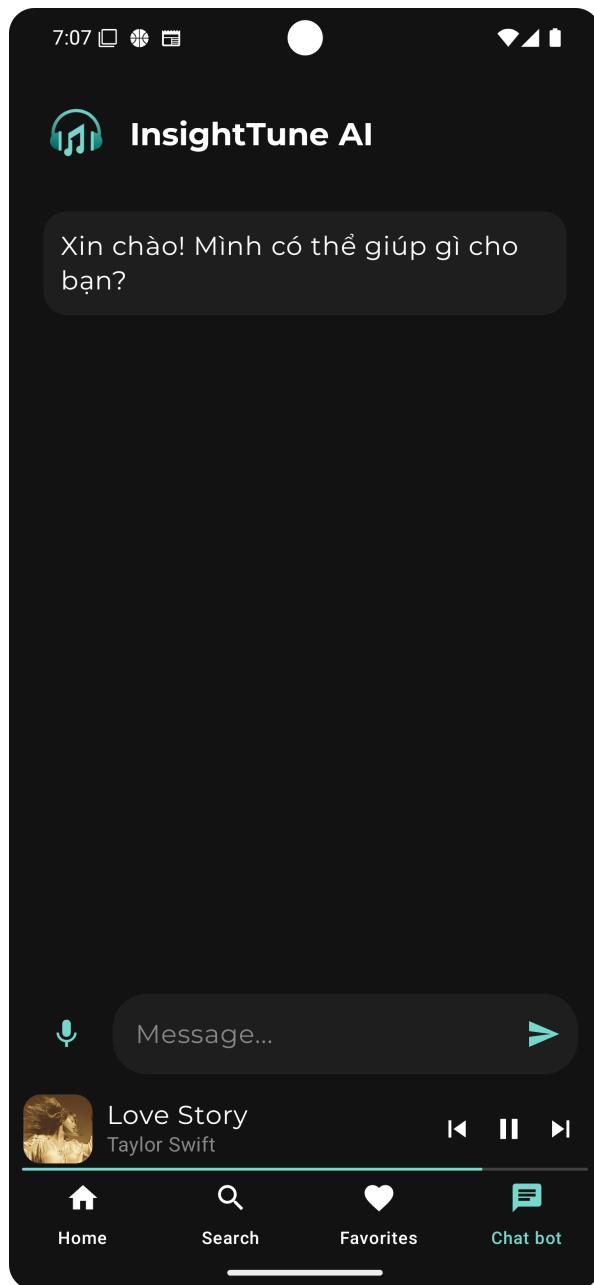
Hình 6.21. Giao diện phát nhạc ở chế độ tối



Hình 6.22. Giao diện tìm kiếm ở chế độ tối



Hình 6.23. Giao diện các bài hát yêu thích ở chế độ tối



Hình 6.24. Giao diện chatbot ở chế độ tối

6.2. Kiểm thử hệ thống

6.2.1. Công cụ kiểm thử

Để đảm bảo chất lượng và tính ổn định của hệ thống nghe nhạc trực tuyến Insight-Tune, yêu cầu đặt ra là phải xây dựng bộ kiểm thử, đảm bảo trải nghiệm tốt nhất cho người dùng khi sử dụng hệ thống. Kiểm thử hệ thống được thực hiện nhằm đánh giá được toàn bộ chức năng, đảm bảo rằng tất cả các thành phần cũng như module được tích hợp trong hệ thống hoạt động đồng bộ, nhất quán, đúng theo yêu cầu đặt ra. Kiểm thử hệ thống không chỉ là kiểm thử từng chức năng riêng lẻ mà còn chú trọng đến sự tương

tác của các thành phần trong hệ thống, dữ liệu luân chuyển giữa các module cũng như đáp ứng được các trường hợp có thể xảy ra trong thực tế.

Để thực hiện kiểm thử hệ thống, nhóm sử dụng một số công cụ nhằm kiểm tra chức năng, hiệu năng, tính ổn định và độ tin cậy của ứng dụng:

- Kiểm thử đơn vị (Unit Test) với JUnit
 - Mục đích: Đảm bảo logic chính xác của từng thành phần mã nguồn bên phía Backend.
 - Áp dụng: Sử dụng Junit để xây dựng các ca kiểm thử tự động cho các logic nghiệp vụ quan trọng như đăng nhập, mã hóa mật khẩu... và các thao tác CRUD với CSDL.
- Đo lường Độ bao phủ mã (Code Coverage) với JaCoCo:
 - Mục đích: Đánh giá chất lượng và mức độ đầy đủ của bộ Unit Test.
 - Áp dụng: JaCoCo được tích hợp cùng JUnit để tạo báo cáo trực quan, cho biết tỷ lệ phần trăm mã nguồn (lệnh, nhánh, dòng) đã được kiểm thử. Điều này giúp nhóm phát hiện các khu vực mã "chết" hoặc chưa được kiểm thử.
- Kiểm thử API với Postman
 - Mục đích: Xác minh rằng các giao tiếp giữa Client (ứng dụng) và Server hoạt động đúng như thiết kế.
 - Áp dụng: Sử dụng Postman để gửi các HTTP request (GET, POST, PUT, DELETE) đến các API của hệ thống.
- Kiểm thử Hiệu năng với JMeter:
 - Mục đích: Đánh giá khả năng chịu tải và độ ổn định của hệ thống khi có nhiều người dùng truy cập đồng thời.
 - Áp dụng: JMeter được dùng để giả lập kịch bản nhiều người dùng cùng lúc thực hiện các hành động như đăng nhập, phát nhạc, và tìm kiếm. Kết quả kiểm thử giúp đo lường thời gian phản hồi của server và tìm ra các "nút thắt cổ chai" (bottleneck) về hiệu năng.

Bên cạnh kiểm thử hệ thống, nhóm cũng tiến hành kiểm thử giao diện người dùng để đảm bảo trải nghiệm người dùng mượt mà và tốt nhất. Để thực hiện kiểm thử giao diện, chúng tôi sử dụng một số công cụ như sau:

- Kiểm thử giao diện người dùng với Jetpack Compose Test:
 - Mục đích: Đảm bảo giao diện người dùng với Jetpack Compose sẽ hiển thị

chính xác và hoạt động đúng như mong đợi.

- Áp dụng: Sử dụng createComposeRule để kiểm thử các giao diện không có trạng thái, các hàm onNodeWithTag và onNodeWithText để tìm các thành phần giao diện, các hàm assert để xác nhận giao diện hiển thị đúng.

- Kiểm thử đơn vị (Unit Test) với JUnit4:

- Mục đích: Đảm bảo các hàm, lớp và logic hoạt động chính xác.
- Áp dụng: Sử dụng các chú thích @Test, @Before, @Rule để định nghĩa các ca kiểm thử và thực thi các ca kiểm thử.

6.2.2. Thiết kế các ca kiểm thử

Tên ca kiểm thử	Mô tả	Đầu vào	Kết quả mong đợi
Đăng ký	Kiểm thử chức năng đăng ký thành công	firstname: “tran”, lastname: “dinh”, email: “23020043@vnu.edu.vn”, password: “123456”, confirmPassword: “123456”	Tài khoản mới thêm vào CSDL, chuyển hướng đến trang đăng nhập
Đăng ký-trùng email	Kiểm thử đăng ký khi tài khoản đã tồn tại	firstname: “tran”, lastname: “dinh”, email: “23020043@vnu.edu.vn”, password: “123456”, confirmPassword: “123456”	Thông báo lỗi: “Email already exists”
Đăng ký-xác nhận mật khẩu sai	Kiểm thử password khác confirmPassword	firstname: “tran”, lastname: “dinh”, email: “23020044@vnu.edu.vn”, password: “123456”, confirmPassword: “12345677”	Thông báo lỗi: “Passwords do not match”
Đăng ký-thiếu firstname	Kiểm thử đăng ký thiếu firstname	firstname: “”, lastname: “dinh”, email: “23020044@vnu.edu.vn”, password: “1234567”, confirmPassword: “1234567”	Thông báo lỗi: “Field must not be blank”

Tên ca kiểm thử	Mô tả	Đầu vào	Kết quả mong đợi
Đăng ký-thiếu lastname	Kiểm thử đăng ký thiếu lastname	firstname: "tran", lastname: "", email: "23020044@vnu.edu.vn", password: "1234567", confirmPassword: "1234567"	Thông báo lỗi: "Field must not be blank"
Đăng ký-thiếu email	Kiểm thử đăng ký thiếu email	firstname: "tran", lastname: "dinh", email: "", password: "1234567", confirmPassword: "1234567"	Thông báo lỗi: "Field must not be blank"
Đăng ký-thiếu password	Kiểm thử đăng ký thiếu password	firstname: "tran", lastname: "dinh", email: "23020044@vnu.edu.vn", password: "", confirmPassword: "1234567"	Thông báo lỗi: "Field must not be blank"
Đăng ký-thiếu confirmPassword	Kiểm thử đăng ký thiếu confirmPassword	firstname: "tran", lastname: "dinh", email: "23020044@vnu.edu.vn", password: "1234567", confirmPassword: ""	Thông báo lỗi: "Field must not be blank"
Đăng nhập-sai mật khẩu	Kiểm thử sai mật khẩu	email: "23020043@vnu.edu.vn", password: "12345677"	Thông báo lỗi: "Incorrect password"
Đăng nhập-không tìm thấy email	Kiểm thử email chưa được đăng ký	email: "23020044@gmail.com", password: "123456"	Thông báo lỗi: "Email not found"
Đăng nhập-Google	Kiểm thử đăng nhập bằng Google thành công	Tài khoản email của người dùng	Đăng nhập thành công, chuyển hướng đến trang home

Tên ca kiểm thử	Mô tả	Đầu vào	Kết quả mong đợi
Đăng nhập	Kiểm thử đăng nhập thành công	email: “23020043@vnu.edu.vn”, password: “123456”	Đăng nhập thành công, chuyển hướng đến trang home
Đăng xuất	Kiểm thử chức năng đăng xuất thành công		Đăng xuất thành công, chuyển hướng đến trang đăng nhập
Quên mật khẩu-gửi otp	Kiểm thử chức năng gửi otp khi quên mật khẩu thành công	email: “23020043@vnu.edu.vn”	Hệ thống gửi OTP: “7184920” đến email người dùng, chuyển đến giao diện xác nhận OTP
Quên mật khẩu-xác nhận otp sai	Kiểm thử chức năng xác nhận otp quên mật khẩu không đúng	OTP: “7184921”	Thông báo lỗi: “Invalid OTP”
Quên mật khẩu-xác nhận otp	Kiểm thử chức năng xác nhận otp quên mật khẩu thành công	OTP: “7184920”	Hệ thống xác nhận OTP thành công, chuyển đến giao diện tạo mật khẩu mới
Quên mật khẩu-tạo mật khẩu mới	Kiểm thử chức năng xác nhận otp quên mật khẩu thành công	newPassword: “5zz2n8ce”, confirmPassword: “5zz2n8ce”	Tạo mật khẩu mới thành công, chuyển đến giao diện đăng nhập
Tìm kiếm bài hát	Kiểm thử tìm kiếm theo keyword	keyword: “Taylor”	Hiển thị danh sách bài hát liên quan đến keyword

Tên ca kiểm thử	Mô tả	Đầu vào	Kết quả mong đợi
Sửa thông tin cá nhân-mật khẩu	Kiểm thử đổi mật khẩu thành công	oldPassword: “123456”, newPassword: “123456789”	Đổi mật khẩu thành công
Sửa thông tin cá nhân-sai mật khẩu	Kiểm thử đổi mật khẩu khi điền sai mật khẩu cũ	oldPassword: “12345689”, newPassword: “12345678”	Thông báo lỗi “Incorrect password”
Sửa thông tin cá nhân-avatar	Kiểm thử đổi avatar thành công	Một bức ảnh chọn từ máy hoặc chụp trực tiếp	Avatar cập nhật thành công
Sửa thông tin cá nhân	Kiểm thử đổi thông tin cá nhân của người dùng thành công	firstname: “tran”, lastname: “dinh”, address: “Hanoi”, phone: “0398880776”	Đổi thông tin thành công, giao diện hiện thông tin vừa cập nhật
Sửa thông tin cá nhân-thiếu firstname	Kiểm thử đổi thông tin cá nhân của người dùng khi thiếu firstname	firstname: “”, lastname: “dinh”, address: “Hanoi”, phone: “0398880776”	Thông báo lỗi: “Field must not be blank”
Sửa thông tin cá nhân-thiếu lastname	Kiểm thử đổi thông tin cá nhân của người dùng khi thiếu lastname	firstname: “tran”, lastname: “”, address: “Hanoi”, phone: “0398880776”	Thông báo lỗi: “Field must not be blank”

Tên ca kiểm thử	Mô tả	Đầu vào	Kết quả mong đợi
Sửa thông tin cá nhân-thiếu address	Kiểm thử đổi thông tin cá nhân của người dùng khi thiếu address	firstname: "tran", lastname: "dinh", address: "", phone: "0398880776"	Thông báo lỗi: "Field must not be blank"
Sửa thông tin cá nhân-thiếu phone	Kiểm thử đổi thông tin cá nhân của người dùng khi thiếu phone	firstname: "tran", lastname: "dinh", address: "hanoi", phone: ""	Thông báo lỗi: "Field must not be blank"
Phát bài hát	Kiểm thử phát bài hát thành công	Người dùng chọn bài hát muốn nghe	Chuyển sang giao diện bài hát, bài hát bắt đầu phát nhạc
Lưu bài hát yêu thích	Kiểm thử lưu bài hát khỏi danh sách yêu thích	Người dùng đang ở trong giao diện bài hát, chọn nút yêu thích	Bài hát được thêm vào danh sách yêu thích
Xóa bài hát yêu thích	Kiểm thử xóa bài hát khỏi danh sách yêu thích	Người dùng đang ở trong giao diện danh sách bài hát yêu thích, chọn bài hát xóa khỏi yêu thích	Bài hát được xóa khỏi danh sách yêu thích
Xem bài hát gần đây	Kiểm thử xem bài hát người dùng đã nghe	Người dùng đăng nhập thành công	Danh sách bài hát đã nghe gần đây hiện trên giao diện home
Giao diện đăng ký người dùng	Kiểm thử giao diện đăng ký người dùng thành công	Người dùng đăng ký email: "23021558@vnu.edu.vn", password: "12345678h", confirmPassword: "12345678h", firstname: "Hoang", lastname: "Doan"	Màn hình chuyển về màn hình bắt đầu, hiển thị Toast thông báo "register-successfully"

Tên ca kiểm thử	Mô tả	Đầu vào	Kết quả mong đợi
Giao diện đăng nhập người dùng	Kiểm thử giao diện đăng nhập người dùng thành công	Người dùng đăng nhập email: “23021558@vnu.edu.vn”, password: “12345678h”	Màn hình chuyển đến trang chủ, hiển thị Toast “Login successfully!”
Giao diện đăng nhập người dùng - sai mật khẩu	Kiểm thử giao diện đăng nhập người dùng với mật khẩu sai	Người dùng đăng nhập email: “23021558@vnu.edu.vn”, password: “1234567”	Hiển thị thông báo lỗi “Incorrect email or password”
Điều hướng giao diện đăng ký người dùng	Kiểm thử điều hướng giữa các giao diện đăng ký	Người dùng nhấn vào nút “Next” trên giao diện đăng kí email hoặc mật khẩu để chuyển sang giao diện tiếp theo và ấn biểu tượng “Back” để quay lại giao diện trước đó	Giao diện chuyển sang giao diện tiếp theo hoặc quay lại giao diện trước đó tương ứng
Giao diện quên mật khẩu - nhập email	Kiểm thử giao diện quên mật khẩu khi nhập email	Người dùng nhập email: “23021558@vnu.edu.vn” và ấn nút “Gửi mã OTP”	Màn hình chuyển sang giao diện nhập mã OTP, hiển thị thông báo gửi mã OTP thành công
Giao diện quên mật khẩu - nhập OTP	Kiểm thử giao diện quên mật khẩu khi nhập mã OTP	Người dùng nhập mã OTP đúng: “123456” và ấn nút xác nhận	Màn hình chuyển sang giao diện tạo mật khẩu mới
Giao diện quên mật khẩu - tạo mật khẩu mới	Kiểm thử giao diện quên mật khẩu khi tạo mật khẩu mới	Người dùng nhập mật khẩu mới: “12345678h”, confirm-Password: “12345678h” và ấn nút “Xác nhận”	Màn hình chuyển về màn hình bắt đầu, hiển thị Toast “Change password successfully”

Tên ca kiếm thử	Mô tả	Đầu vào	Kết quả mong đợi
Giao diện màn hình chính	Kiểm thử giao diện màn hình chính hiển thị đúng các bài hát được đề xuất	Người dùng đăng nhập thành công	Giao diện hiển thị đúng các bài hát được đề xuất
Giao diện tìm kiếm bài hát	Kiểm thử giao diện tìm kiếm bài hát	Người dùng nhập từ khóa “Taylor” vào thanh tìm kiếm	Giao diện hiển thị danh sách các bài hát liên quan đến từ khóa “Taylor”, cụ thể là các bài hát “Love Story”, “The Fate of Ophelia”
Giao diện tìm kiếm bài hát - không có kết quả	Kiểm thử giao diện tìm kiếm bài hát khi không có kết quả	Người dùng nhập từ khóa “abcdefghijklmноп” vào thanh tìm kiếm	Giao diện hiển thị thông báo “Không tìm thấy kết quả”
Giao diện phát nhạc	Kiểm thử giao diện phát bài hát	Người dùng chọn bài hát “Love Story”	Giao diện chuyển sang giao diện phát bài hát, hiển thị đúng thông tin bài hát “Love Story”, bài hát bắt đầu phát
Nút tạm dừng	Kiểm thử nút tạm dừng	Người dùng nhấn nút “Pause” trên giao diện đang phát bài “Love Story”	Bài hát được tạm dừng, nút ấn cũng được thay đổi trạng thái tương ứng thành nút “Play”
Nút chuyển bài tiếp	Kiểm thử nút bài hát tiếp theo	Người dùng nhấn nút bài hát tiếp theo trên giao diện phát bài “Love Story”	Giao diện chuyển sang bài hát tiếp theo “The Fate of Ophelia”, bài hát bắt đầu phát

Tên ca kiếm thử	Mô tả	Đầu vào	Kết quả mong đợi
Nút chuyển bài trước	Kiểm thử nút chuyển bài trước	Người dùng nhấn nút bài hát trước trên giao diện phát bài “The Fate of Ophelia”	Giao diện chuyển sang bài hát trước “Love Story”, bài hát bắt đầu phát
Nút thêm vào mục yêu thích	Kiểm thử nút thêm bài hát vào mục yêu thích	Người dùng nhấn nút yêu thích trên giao diện phát bài “Love Story”	Bài hát “Love Story” được thêm vào mục yêu thích, nút yêu thích chuyển sang có màu
Nút phát ngẫu nhiên/lặp lại	Kiểm thử nút phát ngẫu nhiên/lặp lại	Người dùng nhấn nút phát ngẫu nhiên/lặp lại trên giao diện phát bài “Love Story”	Chế độ phát lặp lại được kích hoạt do mặc định là ngẫu nhiên, nút ấn cũng được thay đổi thành không màu
Giao diện các bài hát yêu thích	Kiểm thử giao diện danh sách bài hát yêu thích hiển thị đúng	Người dùng truy cập vào mục yêu thích từ thanh điều hướng	Giao diện hiển thị đúng danh sách các bài hát yêu thích của người dùng
Xóa bài hát khỏi danh sách yêu thích	Kiểm thử chức năng xóa bài hát khỏi danh sách yêu thích	Người dùng truy cập giao diện yêu thích, ấn biểu tượng trái tim bên cạnh bài “Love Story”	Bài hát “Love Story” được xóa khỏi danh sách yêu thích, giao diện cập nhật lại danh sách yêu thích với bài “Love Story” đã được xóa
Bài hát đã nghe gần đây	Kiểm thử giao diện bài hát đã nghe gần đây	Người dùng nghe các bài hát theo thứ tự “Love Story”, “The Fate of Ophelia” và “Từng là”	Trên màn hình chính xuất hiện dòng “Recently played” với 3 bài hát vừa nghe

Tên ca kiểm thử	Mô tả	Đầu vào	Kết quả mong đợi
Giao diện thông tin cá nhân của người dùng	Kiểm thử giao diện thông tin cá nhân hiển thị đúng	Người dùng truy cập vào mục thông tin cá nhân từ màn hình chính	Giao diện hiển thị đúng thông tin cá nhân của người dùng, bao gồm tên, ảnh đại diện và email cũng như các nút chức năng đổi thông tin, đổi mật khẩu, đổi giao diện, đổi ngôn ngữ và đăng xuất
Đăng xuất người dùng	Kiểm thử chức năng đăng xuất người dùng	Người dùng nhấn nút đăng xuất từ mục thông tin cá nhân	Người dùng được đăng xuất khỏi hệ thống, màn hình chuyển về màn hình bắt đầu
Đổi ảnh đại diện - Bước 1	Kiểm thử chức năng đổi ảnh đại diện	Người dùng nhấn vào ảnh đại diện hiện tại	Màn hình hiện ra cho phép người dùng chọn ảnh từ thư viện hoặc chụp một ảnh mới từ camera
Đổi ảnh đại diện - Bước 2	Kiểm thử chức năng đổi ảnh đại diện	Người dùng chọn một ảnh từ thư viện hoặc chụp một ảnh mới từ camera	Ảnh đại diện của người dùng được cập nhật thành công với ảnh vừa chọn/chụp
Cập nhật thông tin cá nhân - thành công	Kiểm thử chức năng cập nhật thông tin cá nhân	Người dùng nhập đầy đủ các thông tin bao gồm Họ: "Doan", Tên: "Hoang", số điện thoại: "012345678", địa chỉ: "Nam Dinh"	Thông tin cá nhân của người dùng được cập nhật thành công

Tên ca kiểm thử	Mô tả	Đầu vào	Kết quả mong đợi
Cập nhật thông tin cá nhân - không thành công	Kiểm thử chức năng cập nhật thông tin cá nhân khi người dùng nhập thiếu thông tin	Người dùng chỉ nhập Họ: “Doan”, Tên: “Hoang”, để trống số điện thoại và địa chỉ	Hiển thị thông báo lỗi yêu cầu người dùng nhập đầy đủ thông tin
Đổi mật khẩu - thành công	Kiểm thử chức năng đổi mật khẩu	Người dùng ấn vào nút đổi mật khẩu, nhập mật khẩu cũ “12345678h”, mật khẩu mới “1234567890h”	Hiển thị thông báo đổi mật khẩu thành công
Đổi giao diện	Kiểm thử chức năng đổi giao diện	Người dùng ấn vào nút đổi giao diện từ mục thông tin cá nhân, ấn vào “Dark”	Giao diện của ứng dụng được chuyển đổi thành chế độ tối
Đổi ngôn ngữ	Kiểm thử chức năng đổi ngôn ngữ	Người dùng ấn vào nút đổi ngôn ngữ từ mục thông tin cá nhân, ấn vào “Tiếng Việt”	Ngôn ngữ của ứng dụng được chuyển đổi thành tiếng Việt
Giao diện ban đầu của chatbot	Kiểm thử giao diện ban đầu của chatbot	Người dùng mở giao diện chatbot	Giao diện hiển thị đúng các thành phần: tiêu đề, nút đóng, khu vực hiển thị tin nhắn, ô nhập tin nhắn và nút gửi tin nhắn
Gửi tin nhắn đến chatbot	Kiểm thử chức năng gửi tin nhắn đến chatbot	Người dùng nhập tin nhắn “Hello” vào ô nhập tin nhắn và ấn nút gửi	Tin nhắn của người dùng được hiển thị trong khu vực hiển thị tin nhắn, đồng thời nhận được phản hồi từ chatbot

Tên ca kiểm thử	Mô tả	Đầu vào	Kết quả mong đợi
Ngăn chặn gửi tin nhắn trống	Kiểm thử chức năng ngăn chặn gửi tin nhắn trống đến chatbot	Người dùng để trống ô nhập tin nhắn và ấn nút gửi	Hệ thống không gửi tin nhắn trống, không có tin nhắn mới được hiển thị trong khu vực hiển thị tin nhắn
Chỉ báo chatbot đang trả lời	Kiểm thử chức năng hiển thị chỉ báo chatbot đang trả lời	Người dùng gửi tin nhắn đến chatbot	Hiển thị chỉ báo chatbot đang trả lời trong khu vực hiển thị tin nhắn cho đến khi nhận được phản hồi từ chatbot
Tự động cuộn khu vực hiển thị tin nhắn	Kiểm thử chức năng tự động cuộn khu vực hiển thị tin nhắn khi có tin nhắn mới	Người dùng gửi nhiều tin nhắn liên tiếp đến chatbot	Khu vực hiển thị tin nhắn tự động cuộn xuống để hiển thị tin nhắn mới nhất
Quy trình cấp quyền truy cập microphone	Kiểm thử quy trình cấp quyền truy cập microphone cho chatbot	Người dùng mở giao diện chatbot và ấn vào biểu tượng microphone	Hệ thống yêu cầu người dùng cấp quyền truy cập microphone, người dùng có thể chấp nhận hoặc từ chối
Nhập tin nhắn bằng giọng nói	Kiểm thử chức năng nhập tin nhắn bằng giọng nói đến chatbot	Người dùng cấp quyền truy cập microphone, ấn vào biểu tượng microphone và nói “Hello”	Tin nhắn “Hello” được hiển thị trong khu vực hiển thị tin nhắn, đồng thời nhận được phản hồi từ chatbot

Tên ca kiểm thử	Mô tả	Đầu vào	Kết quả mong đợi
Nhập tin nhắn bằng giọng nói khi không có quyền truy cập microphone	Kiểm thử chức năng nhập tin nhắn bằng giọng nói khi không có quyền truy cập microphone	Người dùng từ chối cấp quyền truy cập microphone, ấn vào biểu tượng microphone và nói “Hello”	Hệ thống không nhận tin nhắn giọng nói, không có tin nhắn mới được hiển thị trong khu vực hiển thị tin nhắn
Nhập tin nhắn bằng giọng nói nhưng không nói gì	Kiểm thử chức năng nhập tin nhắn bằng giọng nói khi người dùng không nói gì	Người dùng cấp quyền truy cập microphone, ấn vào biểu tượng microphone nhưng không nói gì	Hệ thống không nhận tin nhắn giọng nói, không có tin nhắn mới được hiển thị trong khu vực hiển thị tin nhắn
Nhập input thông qua POST request của chatbot	Kiểm thử chức năng nhập input thông qua POST request của chatbot	POST /chat messages:Chào bạn, thread-id: tc01	type: reply, reply: Chào bạn! Tôi có thể giúp gì?
Nhập lời bài hát tiếng Việt	Kiểm thử chức năng nhập lời bài hát tiếng Việt	POST Tìm lời bài hát Lâu không cười	type: reply, reply: Lời bài hát Lâu không cười:\n...”
Nhập lời bài hát tiếng Anh	Kiểm thử chức năng nhập lời bài hát tiếng Anh	POST Find lyrics Love Story	type: reply, reply: Lyrics of Love Story:\n...

Tên ca kiểm thử	Mô tả	Đầu vào	Kết quả mong đợi
Phát bài hát thông qua chatbot	Kiểm thử chức năng phát bài hát thông qua chatbot	type: playMusic, song-name: Ngày mai em đi	type: playMusic, song-name: Ngày mai em đi
Gọi ý bài hát thông qua chatbot	Kiểm thử chức năng gọi ý bài hát thông qua chatbot	POST Gợi ý bài hát cho tôi	type: reply, reply: 1... 2... 3...
Thông tin tác giả thông qua chatbot	Kiểm thử chức năng cung cấp thông tin tác giả thông qua chatbot	POST Ai là tác giả của bài hát Từng là?	type: reply, reply: Tác giả của bài hát Từng là là...
Ghi nhớ thread-id trong chatbot	Kiểm thử chức năng ghi nhớ thread-id trong chatbot	POST /chat messages: Chào bạn, tôi tên Hậu, thread-id: tc01 tiếp theo POST /chat messages: Tôi tên là gì?, thread-id: tc01	type: reply, reply: Bạn tên là Hậu,..
Tách biệt thread-id trong chatbot	Kiểm thử chức năng tách biệt thread-id trong chatbot	POST /chat messages: Chào bạn, tôi tên Hậu, thread-id: tc01 tiếp theo POST /chat messages: Tôi tên là gì?, thread-id: tc02	type: reply, reply: Tôi không biết bạn tên gì,...
Kiểm tra lỗi với DuckDuckGo	Kiểm thử chức năng tìm kiếm thông tin với DuckDuckGo thông qua chatbot	Mock ddgs ném exception	Log phải có ERROR Duckduckgo-tool.-run ... và Connection timed out

Tên ca kiểm thử	Mô tả	Đầu vào	Kết quả mong đợi
Kiểm tra lỗi với ScrapeWebsite	Kiểm thử chức năng lấy thông tin với ScrapeWebsite thông qua chatbot	Gọi URL 404	Log có hiện ERROR scrape-website ... 404 Client Error: Not Found for url:
Kiểm tra luồng song-song trong chatbot	Kiểm thử chức năng chạy song-song trong chatbot	10 POST /chat khác thread-id	10 response type: reply, reply: Xin chào user-i!
Xử lý ngoại phạm vi của chatbot	Kiểm thử chức năng xử lí nội dung ngoài phạm vi trong chatbot	POST /chat messages: Hãy kể cho tôi một câu chuyện cười về lập trình viên, thread-id: tc01	type: reply, reply: Xin lỗi, tôi không thể giúp với yêu cầu đó.
Xử lý giọng nói với model Whisper	Kiểm thử chức năng xử lí giọng nói với model Whisper	Gửi file âm thanh có giọng nói “Xin chào” với định dạng .m4a	Xin chào
Xử lý giọng nói không có giọng nói với model Whisper	Kiểm thử chức năng xử lí giọng nói không có giọng nói với model Whisper	Gửi file âm thanh trống với định dạng .m4a	””

Tên ca kiểm thử	Mô tả	Đầu vào	Kết quả mong đợi
Xử lý giọng nói với định dạng không hỗ trợ với model Whisper	Kiểm thử chức năng xử lý giọng nói với định dạng không hỗ trợ với model Whisper	Gửi file âm thanh định dạng .txt	Error: Unsupported audio format.

6.2.3. Thực thi các ca kiểm thử

STT	Tên ca kiểm thử	Ngày kiểm thử	Người kiểm thử	Kết quả thực tế	Trạng thái
1	Đăng ký	25/10/2025	Trần Quang Đỉnh	Tài khoản mới thêm vào CSDL, chuyển hướng đến trang đăng nhập	Đạt
2	Đăng ký-trùng email	25/10/2025	Trần Quang Đỉnh	Thông báo lỗi: “Email already exists”	Đạt
3	Đăng ký-xác nhận mật khẩu sai	25/10/2025	Trần Quang Đỉnh	Thông báo lỗi: “Passwords do not match”	Đạt
4	Đăng ký-thiếu firstname	25/10/2025	Trần Quang Đỉnh	Thông báo lỗi: “Field must not be blank”	Đạt
5	Đăng ký-thiếu lastname	25/10/2025	Trần Quang Đỉnh	Thông báo lỗi: “Field must not be blank”	Đạt
6	Đăng ký-thiếu email	25/10/2025	Trần Quang Đỉnh	Thông báo lỗi: “Field must not be blank”	Đạt
7	Đăng ký-thiếu password	25/10/2025	Trần Quang Đỉnh	Thông báo lỗi: “Field must not be blank”	Đạt

STT	Tên ca kiểm thử	Ngày kiểm thử	Người kiểm thử	Kết quả thực tế	Trạng thái
8	Đăng ký-thiếu confirmPass-word	25/10/2025	Trần Quang Đỉnh	Thông báo lỗi: “Field must not be blank”	Đạt
9	Đăng nhập-sai mật khẩu	25/10/2025	Trần Quang Đỉnh	Thông báo lỗi: “Incorrect password”	Đạt
10	Đăng nhập-không tìm thấy email	25/10/2025	Trần Quang Đỉnh	Thông báo lỗi: “Email not found”	Đạt
11	Đăng nhập-Google	25/10/2025	Trần Quang Đỉnh	Đăng nhập thành công, chuyển hướng đến trang home	Đạt
12	Đăng nhập	25/10/2025	Trần Quang Đỉnh	Đăng nhập thành công, chuyển hướng đến trang home	Đạt
13	Đăng xuất	25/10/2025	Trần Quang Đỉnh	Đăng xuất thành công, chuyển hướng đến trang đăng nhập	Đạt
14	Quên mật khẩu-gửi otp	25/10/2025	Trần Quang Đỉnh	Hệ thống gửi OTP (vd “7184920”), chuyển đến giao diện xác nhận OTP	Đạt
15	Quên mật khẩu-xác nhận otp sai	25/10/2025	Trần Quang Đỉnh	Thông báo lỗi: “Invalid OTP”	Đạt
16	Quên mật khẩu-xác nhận otp	25/10/2025	Trần Quang Đỉnh	Xác nhận OTP thành công, chuyển đến giao diện tạo mật khẩu mới	Đạt

STT	Tên ca kiểm thử	Ngày kiểm thử	Người kiểm thử	Kết quả thực tế	Trạng thái
17	Quên mật khẩu-tạo mật khẩu mới	25/10/2025	Trần Quang Đỉnh	Tạo mật khẩu mới thành công, chuyển đến giao diện đăng nhập	Đạt
18	Tìm kiếm bài hát	25/10/2025	Trần Quang Đỉnh	Hiển thị danh sách bài hát liên quan đến keyword	Đạt
19	Sửa thông tin cá nhân-mật khẩu	25/10/2025	Trần Quang Đỉnh	Đổi mật khẩu thành công	Đạt
20	Sửa thông tin cá nhân-sai mật khẩu	25/10/2025	Trần Quang Đỉnh	Thông báo lỗi “Incorrect password”	Đạt
21	Sửa thông tin cá nhân-avatar	25/10/2025	Trần Quang Đỉnh	Avatar cập nhật thành công	Đạt
22	Sửa thông tin cá nhân	25/10/2025	Trần Quang Đỉnh	Đổi thông tin thành công, giao diện hiện thông tin vừa cập nhật	Đạt
23	Sửa thông tin cá nhân-thiếu first-name	25/10/2025	Trần Quang Đỉnh	Thông báo lỗi: “Field must not be blank”	Đạt
24	Sửa thông tin cá nhân-thiếu last-name	25/10/2025	Trần Quang Đỉnh	Thông báo lỗi: “Field must not be blank”	Đạt
25	Sửa thông tin cá nhân-thiếu address	25/10/2025	Trần Quang Đỉnh	Thông báo lỗi: “Field must not be blank”	Đạt
26	Sửa thông tin cá nhân-thiếu phone	25/10/2025	Trần Quang Đỉnh	Thông báo lỗi: “Field must not be blank”	Đạt

STT	Tên ca kiểm thử	Ngày kiểm thử	Người kiểm thử	Kết quả thực tế	Trạng thái
27	Phát bài hát	26/10/2025	Nguyễn Quốc Huy	Chuyển sang giao diện bài hát, bài hát bắt đầu phát nhạc	Đạt
28	Lưu bài hát yêu thích	26/10/2025	Nguyễn Quốc Huy	Bài hát được thêm vào danh sách yêu thích	Đạt
29	Xóa bài hát yêu thích	26/10/2025	Nguyễn Quốc Huy	Bài hát được xóa khỏi danh sách yêu thích	Đạt
30	Xem bài hát gần đây	26/10/2025	Nguyễn Quốc Huy	Danh sách bài hát đã nghe gần đây hiện trên giao diện home	Đạt
31	Giao diện đăng kí người dùng	03/11/2025	Đoàn Minh Hoàng	Màn hình chuyển về màn hình bắt đầu, hiển thị Toast thông báo “register-successfully”	Đạt
32	Giao diện đăng nhập người dùng	03/11/2025	Đoàn Minh Hoàng	Màn hình chuyển đến trang chủ, hiển thị Toast “Login successfully!”	Đạt
33	Giao diện đăng nhập người dùng - sai mật khẩu	03/11/2025	Đoàn Minh Hoàng	Hiển thị thông báo lỗi “Incorrect email or password”	Đạt
34	Điều hướng giao diện đăng kí người dùng	03/11/2025	Đoàn Minh Hoàng	Giao diện chuyển sang giao diện tiếp theo hoặc quay lại giao diện trước đó tương ứng	Đạt

STT	Tên ca kiểm thử	Ngày kiểm thử	Người kiểm thử	Kết quả thực tế	Trạng thái
35	Giao diện quên mật khẩu - nhập email	03/11/2025	Đoàn Minh Hoàng	Màn hình chuyển sang giao diện nhập mã OTP, hiển thị thông báo gửi mã OTP thành công	Đạt
36	Giao diện quên mật khẩu - nhập OTP	03/11/2025	Đoàn Minh Hoàng	Màn hình chuyển sang giao diện tạo mật khẩu mới	Đạt
37	Giao diện quên mật khẩu - tạo mật khẩu mới	03/11/2025	Đoàn Minh Hoàng	Màn hình chuyển về màn hình bắt đầu, hiển thị Toast thông báo “Change password successfully!”	Đạt
38	Giao diện màn hình chính	03/11/2025	Đoàn Minh Hoàng	Giao diện hiển thị đúng các bài hát được đề xuất	Đạt
39	Giao diện tìm kiếm bài hát	03/11/2025	Đoàn Minh Hoàng	Giao diện hiển thị danh sách các bài hát liên quan đến từ khóa “Taylor”, cụ thể là các bài hát “Love Story”, “The Fate of Ophelia”	Đạt
40	Giao diện tìm kiếm bài hát - không có kết quả	03/11/2025	Đoàn Minh Hoàng	Giao diện hiển thị thông báo “Không tìm thấy kết quả”	Đạt

STT	Tên ca kiểm thử	Ngày kiểm thử	Người kiểm thử	Kết quả thực tế	Trạng thái
41	Giao diện phát nhạc	05/11/2025	Trần Trung Hậu	Giao diện chuyển sang giao diện phát bài hát, hiển thị đúng thông tin bài hát “Love Story”, bài hát bắt đầu phát	Đạt
42	Nút tạm dừng	05/11/2025	Trần Trung Hậu	Bài hát được tạm dừng, nút ấn cũng được thay đổi trạng thái tương ứng thành nút “Play”	Đạt
43	Nút chuyển bài tiếp	05/11/2025	Trần Trung Hậu	Giao diện chuyển sang bài hát tiếp theo “The Fate of Ophelia”, bài hát bắt đầu phát	Đạt
44	Nút chuyển bài trước	05/11/2025	Trần Trung Hậu	Giao diện chuyển sang bài hát trước “Love Story”, bài hát bắt đầu phát	Đạt
45	Nút thêm vào mục yêu thích	04/11/2025	Đoàn Minh Hoàng	Bài hát “Love Story” được thêm vào mục yêu thích, nút yêu thích chuyển sang có màu	Đạt

STT	Tên ca kiểm thử	Ngày kiểm thử	Người kiểm thử	Kết quả thực tế	Trạng thái
46	Nút phát ngẫu nhiên/lắp lại	04/11/2025	Đoàn Minh Hoàng	Chế độ phát lặp lại được kích hoạt do mặc định là ngẫu nhiên, nút ấn cũng được thay đổi thành không màu	Đạt
47	Giao diện các bài hát yêu thích	04/11/2025	Đoàn Minh Hoàng	Giao diện hiển thị đúng danh sách các bài hát yêu thích của người dùng	Đạt
48	Xóa bài hát khỏi danh sách yêu thích	04/11/2025	Đoàn Minh Hoàng	Bài hát “Love Story” được xóa khỏi danh sách yêu thích, giao diện cập nhật lại danh sách yêu thích với bài “Love Story” đã được xóa	Đạt
49	Bài hát đã nghe gần đây	05/11/2025	Đoàn Minh Hoàng	Trên màn hình chính xuất hiện dòng “Recently played” với 3 bài hát “Love Story”, “The Fate of Ophelia”, “Từng là”	Đạt

STT	Tên ca kiểm thử	Ngày kiểm thử	Người kiểm thử	Kết quả thực tế	Trạng thái
50	Giao diện thông tin cá nhân của người dùng	08/11/2025	Đoàn Minh Hoàng	Giao diện hiển thị đúng thông tin cá nhân của người dùng, bao gồm tên, ảnh đại diện và email cũng như các nút chức năng đổi thông tin, đổi mật khẩu, đổi giao diện, đổi ngôn ngữ và đăng xuất	Đạt
51	Đăng xuất người dùng	03/11/2025	Đoàn Minh Hoàng	Người dùng được đăng xuất khỏi hệ thống, màn hình chuyển về màn hình bắt đầu	Đạt
52	Đổi ảnh đại diện - Bước 1	08/11/2025	Đoàn Minh Hoàng	Màn hình hiện ra cho phép người dùng chọn ảnh từ thư viện hoặc chụp một ảnh mới từ camera	Đạt
53	Đổi ảnh đại diện - Bước 2	08/11/2025	Đoàn Minh Hoàng	Ảnh đại diện của người dùng được cập nhật thành công với ảnh vừa chọn/chụp	Đạt
54	Cập nhật thông tin cá nhân - thành công	08/11/2025	Đoàn Minh Hoàng	Thông tin cá nhân của người dùng được cập nhật thành công	Đạt

STT	Tên ca kiểm thử	Ngày kiểm thử	Người kiểm thử	Kết quả thực tế	Trạng thái
55	Cập nhật thông tin cá nhân - không thành công	08/11/2025	Đoàn Minh Hoàng	Hiển thị thông báo lỗi yêu cầu người dùng nhập đầy đủ thông tin	Đạt
56	Đổi mật khẩu - thành công	08/11/2025	Đoàn Minh Hoàng	Hiển thị thông báo đổi mật khẩu thành công	Đạt
57	Đổi giao diện	08/11/2025	Đoàn Minh Hoàng	Giao diện của ứng dụng được chuyển đổi thành chế độ tối	Đạt
58	Đổi ngôn ngữ	08/11/2025	Đoàn Minh Hoàng	Ngôn ngữ của ứng dụng được chuyển đổi thành tiếng Việt	Đạt
59	Giao diện ban đầu của chatbot	10/11/2025	Trần Trung Hậu	Giao diện hiển thị đúng các thành phần: tiêu đề, nút đóng, khu vực hiển thị tin nhắn, ô nhập tin nhắn và nút gửi tin nhắn	Đạt
60	Gửi tin nhắn đến chatbot	10/11/2025	Trần Trung Hậu	Tin nhắn của người dùng được hiển thị trong khu vực hiển thị tin nhắn, đồng thời nhận được phản hồi từ chatbot	Đạt
61	Ngăn chặn gửi tin nhắn trống	10/11/2025	Trần Trung Hậu	Hệ thống không gửi tin nhắn trống, không có tin nhắn mới được hiển thị trong khu vực hiển thị tin nhắn	Đạt

STT	Tên ca kiểm thử	Ngày kiểm thử	Người kiểm thử	Kết quả thực tế	Trạng thái
62	Chỉ báo chatbot đang trả lời	10/11/2025	Trần Trung Hậu	Hiển thị chỉ báo chatbot đang trả lời trong khu vực hiển thị tin nhắn cho đến khi nhận được phản hồi từ chatbot	Đạt
63	Tự động cuộn khu vực hiển thị tin nhắn	10/11/2025	Trần Trung Hậu	Khu vực hiển thị tin nhắn tự động cuộn xuống để hiển thị tin nhắn mới nhất	Đạt
64	Quy trình cấp quyền truy cập microphone	10/11/2025	Trần Trung Hậu	Hệ thống yêu cầu người dùng cấp quyền truy cập microphone, người dùng có thể chấp nhận hoặc từ chối	Đạt
65	Nhập tin nhắn bằng giọng nói	10/11/2025	Trần Trung Hậu	Tin nhắn “Hello” được hiển thị trong khu vực hiển thị tin nhắn, đồng thời nhận được phản hồi từ chatbot	Đạt
66	Nhập tin nhắn bằng giọng nói khi không có quyền truy cập microphone	10/11/2025	Trần Trung Hậu	Hệ thống không ghi nhận tin nhắn giọng nói, không có tin nhắn mới được hiển thị trong khu vực hiển thị tin nhắn	Đạt

STT	Tên ca kiểm thử	Ngày kiểm thử	Người kiểm thử	Kết quả thực tế	Trạng thái
67	Nhập tin nhắn bằng giọng nói nhưng không nói gì	10/11/2025	Trần Trung Hậu	Hệ thống không ghi nhận tin nhắn giọng nói, không có tin nhắn mới được hiển thị trong khu vực hiển thị tin nhắn	Đạt
68	Nhập tin nhắn bằng giọng nói với định dạng không hỗ trợ	10/11/2025	Trần Trung Hậu	Hệ thống hiển thị thông báo lỗi định dạng âm thanh không được hỗ trợ	Đạt
69	Nhập input thông qua POST request của chatbot	12/11/2025	Trần Trung Hậu	Nhận được phản hồi type: reply, reply: Chào bạn! Tôi có thể giúp gì?	Đạt
70	Nhập lời bài hát tiếng Việt	12/11/2025	Trần Trung Hậu	Nhận được phản hồi type: reply, reply: Lời bài hát Lâu không cười:...	Đạt
71	Nhập lời bài hát tiếng Anh	12/11/2025	Trần Trung Hậu	Nhận được phản hồi type: reply, reply: Lyrics of Love Story:...	Đạt
72	Phát bài hát thông qua chatbot	12/11/2025	Trần Trung Hậu	Nhận được phản hồi type: playMusic, song name: Ngày mai em đi	Đạt
73	Gợi ý bài hát thông qua chatbot	12/11/2025	Trần Trung Hậu	Nhận được phản hồi type: reply, reply: "1... 2... 3..."	Đạt

STT	Tên ca kiểm thử	Ngày kiểm thử	Người kiểm thử	Kết quả thực tế	Trạng thái
74	Thông tin tác giả thông qua chatbot	12/11/2025	Trần Trung Hậu	Nhận được phản hồi type: reply, reply: Tác giả của bài hát Từng là là...	Đạt
75	Ghi nhớ thread id trong chatbot	12/11/2025	Trần Trung Hậu	Nhận được phản hồi type: reply, reply: Bạn tên là Hậu,..	Đạt
76	Tách biệt thread id trong chatbot	12/11/2025	Trần Trung Hậu	Nhận được phản hồi type: reply, reply: Tôi không biết bạn tên gì,...	Đạt
77	Kiểm tra lỗi với DuckDuckGo	12/11/2025	Trần Trung Hậu	Log có hiện ERROR Duckduckgo-tool.-run ... và Connection timed out	Đạt
78	Kiểm tra lỗi với ScrapeWebsite	12/11/2025	Trần Trung Hậu	Log có hiện ER-ROR scrape-website ... 404 Client Error: Not Found for url:	Đạt
79	Kiểm tra luồng song song trong chatbot	12/11/2025	Trần Trung Hậu	Nhận được 10 response type: reply, reply: Xin chào user i!	Đạt
80	Xử lý ngoài phạm vi của chatbot	12/11/2025	Trần Trung Hậu	Nhận được phản hồi type: reply, reply: Xin lỗi, tôi không thể giúp với yêu cầu đó.	Đạt
81	Xử lý giọng nói với model Whisper	13/11/2025	Trần Trung Hậu	Nhận được phản hồi Xin chào	Đạt

STT	Tên ca kiểm thử	Ngày kiểm thử	Người kiểm thử	Kết quả thực tế	Trạng thái
82	Xử lý giọng nói không có giọng nói với model Whisper	13/11/2025	Trần Trung Hậu	Nhận được phản hồi rõng	Đạt
83	Xử lý giọng nói với định dạng không hỗ trợ với model Whisper	13/11/2025	Trần Trung Hậu	Nhận được phản hồi Error: Unsupported audio format.	Đạt

6.2.4. Kiểm thử hiệu năng

Đánh giá hiệu năng là bước quan trọng nhằm xác định mức độ đáp ứng, độ ổn định và khả năng chịu tải của hệ thống dưới các điều kiện sử dụng khác nhau. Mục tiêu của quá trình kiểm thử hiệu năng là đảm bảo ứng dụng duy trì thời gian phản hồi tốt, hoạt động ổn định và không xảy ra lỗi khi số lượng người dùng tăng hoặc khi hệ thống thực hiện nhiều tác vụ đồng thời.

Trong dự án này, nhóm sử dụng **Apache JMeter** – công cụ mã nguồn mở phổ biến cho kiểm thử tải và đo lường hiệu năng – để:

- Mô phỏng lượng người dùng truy cập.
- Thực thi các kịch bản nghiệp vụ.
- Thu thập số liệu thực nghiệm.

Jmeter cho phép tạo các *test plan* linh hoạt, gửi nhiều loại yêu cầu (HTTP, API, database, ...) và quan sát các chỉ số quan trọng:

- **Thời gian phản hồi:** đo tốc độ hệ thống trả lời yêu cầu.
- **Tốc độ xử lý (Throughput):** số lượng yêu cầu xử lý trên giây.
- **Tỷ lệ lỗi:** phần trăm các yêu cầu thất bại.

Hệ thống được triển khai trên cấu hình kiểm thử như sau:

- **Thread group:** JMeter khởi tạo 100 người dùng ảo trong vòng 5 giây (tương đương khoảng 20 users/giây). Mỗi người dùng thực hiện 1 vòng lặp kịch bản để mô phỏng lượng lớn yêu cầu đồng thời.
- **Test Plan:** tập trung kiểm thử các API REST của hệ thống.

- **Thành phần hỗ trợ:** HTTP Header Manager, Response Assertion, Post Processor
 - JSON Extractor, Listener.

Các API được kiểm thử:

- **Xác thực và người dùng:**
 - POST /auth/register
 - POST /auth/login
 - PUT /user/changePassword
 - GET /auth/refresh
 - PUT /users
 - GET /users
 - POST /auth/logout
- **Quản lý tracks:**
 - GET /tracks
 - GET /tracks/search?keyword=Taylor
 - GET /tracks/next/33333333-3333-3333-3333-333333333333
- **Lịch sử và yêu thích:**
 - GET /history/search
 - POST /favorites/add
 - GET /favorites
 - POST /favorites/delete

Kết quả kiểm thử:

- Thời gian phản hồi tối đa (Max Response Time): **< 1.9 giây**
- Tỉ lệ lỗi (Error / Fail): **0%**
- Tốc độ xử lý (Throughput): **166 requests/giây**

Đánh giá: Ứng dụng hoàn toàn đáp ứng yêu cầu tải của bài kiểm thử và phù hợp cho triển khai thực tế.

CHƯƠNG 7

TỔNG KẾT VÀ ĐỊNH HƯỚNG TƯƠNG LAI

7.1. Tổng kết

Ứng dụng âm nhạc InsightTune được tích hợp công nghệ trí tuệ nhân tạo với giao diện thiết kế đơn giản, cá nhân hóa trải nghiệm của người dùng.

Hệ thống cung cấp đầy đủ các tính năng cơ bản của một ứng dụng nghe nhạc hiện đại, đồng thời tích hợp các công nghệ AI tiên tiến như Chatbot trợ lý ảo và hệ thống gợi ý bài hát, giúp người dùng có một trải nghiệm nghe nhạc tuyệt vời hơn.

7.1.1. Điểm cộng của hệ thống

- Giao diện thân thiện người dùng phổ thông.
- Hệ thống gợi ý bài hát cá nhân hóa dựa trên sở thích người dùng.
- Chatbot trợ lý ảo có khả năng giải đáp các thắc mắc, cung cấp thông tin liên quan đến âm nhạc. Người dùng có thể ra lệnh cho Chatbot phát nhạc bằng giọng nói hoặc nhập yêu cầu.
- Kiến trúc Microservices giúp hệ thống dễ bảo trì và mở rộng trong tương lai.
- Hệ thống được triển khai trên nền tảng đám mây, đảm bảo tính sẵn sàng và khả năng mở rộng.

7.1.2. Hạn chế của hệ thống

- Chatbot đôi khi trả lời chưa chính xác, hoặc không tự gọi tool.
- Hệ thống gợi ý chưa thực sự cá nhân hóa sâu.
- Ứng dụng mới chỉ là dạng di động, chưa có phiên bản web hoặc desktop.
- Giao diện người dùng còn chưa tối đa hóa trải nghiệm, thiết kế chưa rõ ràng.
- Hệ thống chưa có cơ chế caching

7.2. Định hướng tương lai

- Phát triển thêm phiên bản web và ứng dụng trên desktop.
- Hệ thống gợi ý được tối ưu hơn.
- Thêm cơ chế caching cho hệ thống.

- Giao diện đơn giản, tập trung vào trải nghiệm người dùng.
- Phát triển thêm nhiều tính năng như thêm lời bài hát kèm theo bản dịch (nếu cần), ghim album,..
- Có thể phát triển AI theo hướng mix nhạc tự động dựa vào các bài hát được cung cấp.
- Chatbot có thể chia ra thành nhiều cuộc hội thoại riêng biệt để tránh nhầm lẫn thông tin.

TÀI LIỆU THAM KHẢO