

BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP.HCM
KHOA CÔNG NGHỆ THÔNG TIN



PHAN THỊ HUỲNH THƯ
LÊ VÕ DUY TUẤN

XÂY DỰNG ỨNG DỤNG NGHE NHẠC
JT-HARMONY TRÊN MOBILE TÍCH HỢP
GỢI Ý VÀ CHIA SẺ

Ngành: Kỹ thuật phần mềm

Giảng viên hướng dẫn: ThS. Nguyễn Thị Hồng Lương

THÀNH PHỐ HỒ CHÍ MINH, THÁNG 08 NĂM 2025

BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP.HCM
KHOA CÔNG NGHỆ THÔNG TIN



PHAN THỊ HUỲNH THU'
LÊ VÕ DUY TUẤN

XÂY DỰNG ỨNG DỤNG NGHE NHẠC
JT-HARMONY TRÊN MOBILE TÍCH HỢP
GỢI Ý VÀ CHIA SẺ

Ngành: Kỹ thuật phần mềm

Giảng viên hướng dẫn: ThS. Nguyễn Thị Hồng Lương

THÀNH PHỐ HỒ CHÍ MINH, THÁNG 08 NĂM 2025

INDUSTRIAL UNIVERSITY OF HO CHI MINH CITY
FACULTY OF INFORMATION TECHNOLOGY



PHAN THI HUYNH THU
LE VO DUY TUAN

BUILDING THE JT-HARMONY MOBILE MUSIC
APPLICATION WITH INTEGRATED
RECOMMENDATION AND SHARING

Major: Software Engineering

Supervisor: Msc. Nguyen Thi Hong Luong

HO CHI MINH CITY, AUGUST 2025

ABSTRACT

Reason for choosing the topic: With the rapid growth and intense competition of digital music platforms such as Spotify, Zing MP3, Apple Music, and SoundCloud, users increasingly expect more than just high - quality music playback - they also seek personalized recommendations and social interaction features. However, most existing applications focus primarily on music streaming and pay little attention to enabling users to share cover songs, original compositions, or engage in community interactions. Therefore, this topic was chosen to develop a mobile music application that not only meets the need for music listening but also integrates intelligent recommendations powered by Machine Learning while creating a space for users to share and interact with one another.

Problems: Current music applications offer good recommendation systems but are limited in supporting users to upload personal content (covers, original compositions); There is no platform that integrates both music listening and sharing personal works along with community interaction; Music personalization has not been fully optimized according to users' moods, activities, or detailed preferences.

Methods: The system is built upon a Client-Server architecture, utilizing technologies such as React Native with Tailwind CSS for the mobile application, Next.js for the administration website, and Node.js/ExpressJS for the backend. It also incorporates Socket.IO, PostgreSQL, and an AI Recommendation System (using Content-based & Collaborative Filtering) to provide personalized experiences. The application offers users a digital music platform combined with a social networking space, where they can share cover songs and original compositions, leave comments, and communicate with other users in real-time.

LỜI CẢM ƠN

Đầu tiên, nhóm sinh viên xin gửi lời cảm ơn đến Ban Giám hiệu và quý Thầy Cô Khoa Công nghệ Thông tin, Trường Đại học Công Nghiệp Thành phố Hồ Chí Minh. Trong suốt quá trình học tập đã luôn nhẫn nại và tận tâm giảng dạy chúng em, những kiến thức nền tảng và chuyên ngành được trang bị trong suốt quá trình học tập tại trường là cơ sở quan trọng giúp chúng em thực hiện và hoàn thành đồ án này.

Đặc biệt, nhóm xin gửi lời cảm ơn chân thành đến Thạc sĩ Nguyễn Thị Hồng Lương. Trong suốt thời gian thực hiện khóa luận, cô đã luôn theo sát, định hướng phương pháp nghiên cứu và đưa ra những chỉ dẫn chuyên môn xác đáng. Những nhận xét tỉ mỉ của Cô đã giúp nhóm giải quyết được nhiều vấn đề kỹ thuật phát, và hoàn thiện các chức năng của hệ thống và cải thiện phong cách làm việc một cách tốt nhất.

Chúng em cũng xin cảm ơn quý Thầy Cô trong Hội đồng phản biện đã dành thời gian thẩm định và đóng góp ý kiến. Những nhận xét khách quan của quý Thầy Cô giúp chúng em nhận ra những hạn chế còn tồn tại trong đồ án, nhờ đó chúng em có thể củng cố thêm kiến thức và kinh nghiệm cho công việc sau này.

Mặc dù đã nỗ lực hoàn thành các yêu cầu của đề tài, nhưng do thời gian và kinh nghiệm thực tế còn hạn chế, khóa luận khó tránh khỏi những thiếu sót. Chúng em rất mong nhận được những ý kiến đóng góp từ quý Thầy Cô để đề tài có thể hoàn thiện và mang lại giá trị ứng dụng cao hơn.

Cuối cùng, chúng em xin kính chúc quý thầy cô trong khoa Công Nghệ Thông Tin nói chung, và đặc biệt cô Nguyễn Thị Hồng Lương nói riêng, luôn dồi dào sức khỏe, để có thể tiếp tục ươm mầm nên các thế hệ tiếp nối tinh thần học tập và đạt được những thành tựu xuất sắc trong tương lai.

Chúng em xin chân thành cảm ơn!

This image shows a full page of white paper designed for handwriting practice. It features 20 evenly spaced, horizontal dotted lines extending across the entire width of the page. The lines are thin and light gray, providing a guide for letter height and placement without being distracting. There is no text or other markings on the page.

CHỮ KÝ GIẢNG VIÊN

This image shows a full page of white paper with horizontal dotted lines. The lines are evenly spaced and run across the entire width of the page, providing a guide for handwriting practice. There are no margins, text, or other markings on the paper.

CHỮ KÝ GIẢNG VIÊN

This image shows a full page of white paper with horizontal dotted lines. The lines are evenly spaced and run across the width of the page, providing a guide for handwriting practice. There are no margins, text, or other markings on the page.

CHỮ KÝ GIẢNG VIÊN

MỤC LỤC

Chương 1. GIỚI THIỆU	11
1.1. Tổng quan:	11
1.2. Mục tiêu cơ bản của đề tài	12
1.3. Phạm vi đề tài	12
1.3.1. Phạm vi thực hiện	12
1.3.2. Ngoài phạm vi đề tài:	13
1.4. Phương pháp thực hiện:	13
1.4.1. Phương pháp khảo sát/ thu thập yêu cầu:	13
1.4.2. Các công nghệ sử dụng trong hệ thống:	13
1.5. Ý nghĩa khoa học và thực tiễn:	14
1.5.1. Ý nghĩa khoa học:	14
1.5.2. Ý nghĩa thực tiễn	15
1.6. Cấu trúc quyền báo cáo:	15
Chương 2. CƠ SỞ LÝ THUYẾT VÀ CÔNG NGHỆ LIÊN QUAN	17
2.1. Node.js và ExpressJS:	17
2.2. React Native:	17
2.3. Next.js:	17
2.4. PostgreSQL:	18
2.5. Socket.IO:	18
2.6. Expo:	18
2.7. TailwindCSS và Nativewind:	19
2.8. Sequelize (ORM):	19
2.9. Kiến trúc Client - Server:	20
Chương 3. PHÂN TÍCH YÊU CẦU VÀ THIẾT KẾ HỆ THỐNG	21
3.1. Khảo sát và thu thập yêu cầu:	21
3.1.1. Phương pháp thực hiện:	21
3.1.2. Kế hoạch thực hiện	21

3.1.3. Kết quả thu thập và phân tích:	22
3.2. Đặc tả yêu cầu chức năng, phi chức năng:	23
3.2.1. Yêu cầu chức năng:	23
3.2.2. Yêu cầu phi chức năng	23
3.2.3. Yêu cầu liên quan đến AI:	24
3.2.4. Quy tắc quản lý tài khoản và định danh:	24
3.2.5. Quy tắc kiểm duyệt cộng đồng:	25
3.3. Các chức năng chính của hệ thống:	25
3.3.1. Các nhóm người dùng – quyền hạn trên hệ thống	25
3.3.2. Usecase tổng quát:	30
3.3.3. Kiến trúc hệ thống:	47
3.3.4. Sơ đồ cơ sở dữ liệu (Database Diagram)	48
3.3.5. Một số giao diện chính của hệ thống:	52
Chương 4. CÀI ĐẶT VÀ TRIỂN KHAI HỆ THỐNG	58
4.1. Môi trường phát triển và công cụ:	58
4.2. Kiến trúc triển khai hệ thống:	59
4.3. Các module chức năng chính:	61
4.4. Giao diện và luồng hoạt động:	62
4.5. Tích hợp:	63
Chương 5. KIỂM THỬ VÀ ĐÁNH GIÁ HỆ THỐNG	64
5.1. Môi trường kiểm thử:	64
5.1.1. Phần cứng:	64
5.1.2. Phần mềm:	64
5.2. Chiến lược kiểm thử:	64
5.3. Danh sách các tình huống kiểm thử:	66
Chương 6. KẾ HOẠCH KHỞI NGHIỆP	70
6.1. Giới thiệu về dự án:	70
6.2. Phạm vi:	71
6.3. Lịch trình:	72
6.4. Ngân sách:	73

6.5. Quản lý nhân lực	74
6.6. Quản lý rủi ro:	76
6.7. Kế hoạch triển khai:	77
6.7.1. Giai đoạn thử nghiệm:	77
6.7.2. Giai đoạn cải tiến, mở rộng:	78
6.7.3. Giai đoạn tiếp thị:	78
Chương 7. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	79
7.1. Kết quả đạt được:	79
7.2. Đánh giá mức độ hoàn thành mục tiêu:	80
7.3. Hạn chế:	80
7.4. Hướng phát triển trong tương lai:	80
TÀI LIỆU THAM KHẢO	82
PHỤ LỤC	83
PHỤ LỤC 1: BẢNG CÂU HỎI KHẢO SÁT	83
PHỤ LỤC 2: KẾ HOẠCH THỰC HIỆN ĐỀ TÀI	89
PHỤ LỤC 3: NHẬT KÍ THỰC HIỆN ĐỀ TÀI	93

DANH MỤC CÁC HÌNH VẼ

Hình 3.1 : Use case tổng quát	30
Hình 3.2 : Sơ đồ hoạt động - UC001 Đăng ký	33
Hình 3.3 : Sơ đồ trình tự -UC001 Đăng ký	34
Hình 3.4 : Sơ đồ hoạt động UC002 - Đăng nhập	36
Hình 3.5 : Sơ đồ trình tự UC002 - Đăng nhập	38
Hình 3.6 : Sơ đồ hoạt động UC003 - Tạo danh sách phát	40
Hình 3.7 : Sơ đồ trình tự UC003 - Tạo danh sách phát	41
Hình 3.8 : Sơ đồ hoạt động UC004 - Tạo bài đăng/Cover	43
Hình 3.9 : Sơ đồ trình tự UC004 - Tạo bài đăng/Cover	44
Hình 3.10 : Sơ đồ hoạt động UC005 - Nhắn tin	46
Hình 3.11 : Sơ đồ trình tự UC005 - Nhắn tin	47
Hình 3.12 : Sơ đồ kiến trúc hệ thống JT - Harmony	47
Hình 3.13 : Sơ đồ CSDL quan hệ (bản đầy đủ)	48
Hình 3.14 : Sơ đồ CSDL (user-schema)	49
Hình 3.15 : Sơ đồ CSDL quan hệ (music schema)	50
Hình 3.16 : Sơ đồ CSDL quan hệ (social schema)	51
Hình 3.17 : Màn hình Auth Landing	53
Hình 3.18 : Màn hình đăng nhập	53
Hình 3.19 : Màn hình Đăng ký	54
Hình 3.20 : Màn hình chính	54
Hình 3.21 : Màn hình Tìm kiếm	54
Hình 3.22 : Màn hình Mạng xã hội	54
Hình 3.23 : Màn hình Profile Social	54
Hình 3.24 : Màn hình Chi tiết bài hát	54
Hình 3.25 : Màn hình Bình luận	55
Hình 3.26 : Màn hình Nhắn tin	55

Hình 3.27 : Màn hình Trang chủ quản lý	56
Hình 3.28 : Màn hình Quản lý người dùng	56
Hình 3.29 : Màn hình Quản lý bài đăng	56
Hình 3.30 : Màn hình Xem thống kê thông tin âm nhạc	57
Hình 4.1 : Mô hình triển khai front-end mobile	60
Hình 4.2 : Mô hình triển khai front-end web	60
Hình 4.3 : Mô hình triển khai back-end	61
Hình 4.4 : Sơ đồ luồng hoạt động màn hình trên mobile	62

DANH SÁCH CÁC BẢNG

Bảng 3.1 : Danh sách tác nhân mô tả.....	25
Bảng 3.2 : Danh sách chức năng và quyền hạn người dùng	26
Bảng 3.3 : Đặc tả UC001 - Đăng ký.....	31
Bảng 3.4 : Đặc tả UC002 - Đăng nhập.....	35
Bảng 3.5 : Đặc tả UC003 - Tạo danh sách phát.....	39
Bảng 3.6 : Đặc tả UC004 - Tạo bài đăng/Cover.....	41
Bảng 3.7 : Đặc tả UC005 - Nhắn tin.....	44
Bảng 4.1 : Các thành phần môi trường triển khai và công cụ.....	58
Bảng 5.1 : Bảng thông số cấu hình máy tính cá nhân.....	64
Bảng 5.2 : Bảng thông tin phần mềm.....	64
Bảng 5.3 : Bảng báo cáo kết quả kiểm thử.....	66
Bảng 6.1 : Bảng tóm tắt lịch trình dự án.....	72
Bảng 6.2 : Bảng dự kiến ngân sách dự án.....	73
Bảng 6.3 : Bảng phân công nhiệm vụ (RACI).....	74
Bảng 6.4 : Bảng quản lý rủi ro (Risk Management Table).....	76
Bảng 7.1 : Bảng đánh giá mức độ hoàn thành mục tiêu.....	80

DANH MỤC CÁC THUẬT NGỮ VIẾT TẮT

Từ viết tắt	Định nghĩa
JWT	Json Web Token: Mã hóa thông tin an toàn giữa các bên.
SQL	Structured Query Language: Ngôn ngữ truy vấn cơ sở dữ liệu.
API	Application Programming Interface: giao diện lập trình ứng dụng
CSDL/DB	Cơ sở dữ liệu/Database
UX/UI	User Interface/ User Interface: trải nghiệm người dùng/Giao diện người dùng
HTTP	HyperText Transfer Protocol: Giao thức truyền tải siêu văn bản
CI/CD	Continuous Integration/Continuous Delivery or Deployment): Tích hợp liên tục/Phân phối - Triển khai liên tục
AI	Artificial Intelligence: Trí tuệ nhân tạo
MXH	Mạng xã hội
IDE	Integrated Development Environment: Môi trường phát triển tích hợp)
RESTful	Representational State Transfer: một chuẩn thiết kế

	API
URL	Uniform Resource Locator: địa chỉ duy nhất giúp định vị và truy cập các tài nguyên trên Internet
OTP	One-Time Password : Mật khẩu dùng một lần
JSX	JavaScript XML: một phần mở rộng cú pháp cho JavaScript
ORDBMS	Object-Relational Database Management System: Hệ thống Quản lý Cơ sở Dữ liệu Quan hệ Đối tượng
ORM	Object-Relational Mapping: Ánh xạ Quan hệ Đối tượng
RACI	Responsible (Người thực hiện) - Accountable (Người chịu trách nhiệm cuối cùng/phê duyệt) - Consulted (Người được tham vấn) và Informed (Người được thông báo).
CDN	Content Delivery Network: Mạng phân phối nội dung
PM	Project Manager: Người quản lý dự án.
CTO	Chief Technology Officer: Giám đốc Công nghệ,

Chương 1. GIỚI THIỆU

1.1. Tổng quan:

Trong bối cảnh thị trường ứng dụng nghe nhạc số ngày càng phát triển và cạnh tranh gay gắt, người dùng không chỉ dừng ở nhu cầu tìm kiếm một nền tảng nghe nhạc chất lượng cao mà còn hướng đến những trải nghiệm mang tính cá nhân hóa và khả năng kết nối cộng đồng. Ứng dụng JT-Harmony được phát triển nhằm mang đến những người yêu âm nhạc một công cụ nghe nhạc toàn diện, hiện đại và dễ sử dụng, vừa đáp ứng nhu cầu cá nhân hóa, vừa khuyến khích sự tương tác và kết nối với cộng đồng những người yêu âm nhạc.

Ứng dụng JT-Harmony được phát triển trên nền tảng Android, giao diện được thiết kế thân thiện, hiện đại, trực quan, đảm bảo trải nghiệm người dùng mượt mà và tính thẩm mỹ. Tương tự như các nền tảng khác, JT-Harmony cũng bao gồm các tính năng cơ bản của một ứng dụng nghe nhạc cần có như tìm kiếm bài hát, phát/tạm dừng/chuyển bài, xem lời bài hát, tạo playlist cá nhân và có hỗ trợ nghe nhạc trực tuyến (online).

Điểm nổi bật của JT-Harmony là tính năng gợi ý nhạc cá nhân hóa, dựa trên lịch sử nghe nhạc, thể loại hoặc nghệ sĩ yêu thích, cũng như các bài hát đang thịnh hành. Điều này giúp người nghe dễ dàng khám phá ra những bài hát mới phù hợp với sở thích và nhu cầu nghe.

Bên cạnh đó ứng dụng không chỉ cho phép chia sẻ nhạc thông qua các nền tảng mạng xã hội khác (Facebook, Zalo, ...) mà còn xây dựng một nền tảng mạng xã hội bên trong ứng dụng, người dùng không chỉ được nghe những bài hát phát hành chính thức mà còn có thể đăng tải những bản cover, những sáng tác cá nhân. Việc phát triển tính năng này không chỉ giúp người dùng kết nối, tương tác với những người nghe khác, mà còn là một cơ hội cho những người có đam mê muốn bắt đầu phát triển bản thân trong lĩnh vực nghệ thuật được tiếp cận thêm một lượng khán giả.

Về mặt quản trị, hệ thống tích hợp trang quản trị (admin) cho phép quản trị viên quản lý thông tin bài hát, thể loại, nghệ sĩ; quản lý người dùng; theo dõi thống kê lượt nghe và chia sẻ; từ đó đảm bảo nội dung được kiểm soát và ứng dụng vận hành ổn định.

Về công nghệ, JT-Harmony sử dụng JWT để xác thực, kết hợp với PostgreSQL cho quản lý cơ sở dữ liệu chính và Redis cho lưu trữ kết quả các truy vấn thường xuyên, cùng các API nhạc và dịch vụ lưu trữ để đảm bảo tốc độ truy xuất nhanh, an toàn và khả năng mở rộng. Thêm vào đó, ứng dụng tích hợp Socket.IO để hỗ trợ giao tiếp thời gian thực như bình luận, thông báo và tương tác giữa người dùng.

Với mong muốn mang đến trải nghiệm giải trí cá nhân hóa và tiện ích cho cộng đồng người nghe nhạc, chúng tôi hy vọng JT-Harmony không chỉ là ứng dụng nghe nhạc mà còn là cầu nối cộng đồng, nơi người yêu nhạc có thể khám phá, chia sẻ và phát triển bản thân.

1.2. Mục tiêu cơ bản của đề tài

Đề tài nhằm mục tiêu xây dựng một ứng dụng nghe nhạc trên nền tảng di động Android, tích hợp gợi ý nhạc cá nhân hóa và chia sẻ nhạc. Giúp người dùng dễ dàng tìm kiếm bài hát mới, theo dõi các bài hát thịnh hành, nghệ sĩ yêu thích và có thể kết nối với cộng đồng những người yêu âm nhạc.

1.3. Phạm vi đề tài

1.3.1. Phạm vi thực hiện

Xây dựng ứng dụng nghe nhạc trên mobile:

- Đăng ký/ Đăng nhập.
- Nghe nhạc trực tuyến.
- Quản lý danh sách phát cá nhân.
- Tìm kiếm bài hát, nghệ sĩ, album, danh sách phát, người dùng khác.
- Gợi ý bài hát dựa trên phân tích hành vi, lịch sử nghe nhạc.

Xây dựng mạng xã hội nội bộ trong ứng dụng:

- Tạo bài đăng/cover, chia sẻ.
- Bình luận, bày tỏ cảm xúc.
- Nhắn tin 1:1 giữa người dùng (chat real-time).

Cơ sở dữ liệu: sử dụng PostgreSQL làm database chính và Redis lưu trữ bộ nhớ đệm.

Tích hợp API bên thứ 3:

- Sử dụng Spotify Web API để tham khảo metadata bài hát, nghệ sĩ, danh sách phát toàn cầu.
- Sử dụng Gemini API để xây dựng chức năng gợi ý nghe nhạc thông minh, nhằm cá nhân hóa trải nghiệm cho người dùng.

Cloud: Sử dụng Cloudinary để đăng tải, lưu trữ hình ảnh, file media.

1.3.2. Ngoài phạm vi đề tài:

Không phát triển web client cho người dùng. Chỉ xây dựng mobile client và web admin cho quản trị viên.

1.4. Phương pháp thực hiện:

1.4.1. Phương pháp khảo sát/ thu thập yêu cầu:

- *Thảo luận nhóm:* các thành viên cùng phân tích yêu cầu chức năng hệ thống, xác định phạm vi thực hiện đề tài.
- *Khảo sát so sánh:* trải nghiệm sử dụng các ứng dụng tương tự trong nền tảng nghe nhạc (Spotify, ZingMp3, Apple Music).
- *Nghiên cứu tài liệu:* đọc tài liệu tham khảo từ trang chủ các nền tảng nhạc trực tuyến tương tự, tài liệu của Spotify API và Gemini API.

1.4.2. Các công nghệ sử dụng trong hệ thống:

- *Node.js, Express.js:* Xây dựng server và cung cấp Restful API.

- *React Native + TypeScript*: Phát triển ứng dụng di động đa nền tảng, tối ưu hiệu suất và trải nghiệm người dùng.
- *Next.js*: Xây dựng website quản lý cho quản trị viên hệ thống.
- *PostgreSQL*: Lưu trữ và quản lý dữ liệu quan hệ, đảm bảo tính toàn vẹn, an toàn và dễ mở rộng.
- *GitHub*: Quản lý mã nguồn, theo dõi thay đổi và hỗ trợ cộng tác nhóm thông qua pull request và review code.
- *Cloudinary*: Lưu trữ và quản lý hình ảnh/âm thanh, hỗ trợ tối ưu hóa và phân phối nội dung nhanh chóng.
- *Socket.IO*: Hỗ trợ giao tiếp thời gian thực giữa client và server, phục vụ tính năng chat, bình luận và thông báo tức thì.
- *Redis*: Lưu trữ bộ nhớ đệm, giúp tăng tốc độ truy cập dữ liệu và giải tải cho cơ sở dữ liệu chính.

1.5. Ý nghĩa khoa học và thực tiễn:

1.5.1. Ý nghĩa khoa học:

Nghiên cứu hệ thống gợi ý: Vận dụng, kết hợp các kiến thức về hệ thống gợi ý (Recommendation Systems). Cụ thể là áp dụng các thuật toán như Lọc nội dung (Content-based Filtering) và Lọc cộng tác (Collaborative Filtering) để giải quyết vấn đề cá nhân hóa trải nghiệm người dùng trong lĩnh vực âm nhạc.

Tích hợp công nghệ thời gian thực: Đề tài nghiên cứu và tích hợp công nghệ Socket.IO vào một ứng dụng media để xây dựng các tính năng tương tác xã hội (bình luận, thông báo), góp phần tạo ra một mô hình "mạng xã hội âm nhạc".

Ứng dụng kiến trúc hiện đại: Đề tài đóng góp một kiến trúc hệ thống tham khảo kết hợp 3 nền tảng công nghệ mạnh mẽ: Next.js (cho Web Admin), React Native (cho Mobile App), và Node.js/ExpressJS (cho Backend Server) để giải quyết một bài toán nghiệp vụ hoàn chỉnh.

1.5.2. Ý nghĩa thực tiễn

Đối với người dùng: Người dùng không chỉ thụ động nghe nhạc mà còn có thể chủ động sáng tạo (đăng tải cover/sáng tác) và tương tác, kết nối với những người dùng khác có cùng sở thích.

Đối với cộng đồng sáng tạo: Tạo ra một sân chơi, một nền tảng đặc biệt cho các nghệ sĩ, người sáng tạo nghiệp dư có cơ hội chia sẻ tác phẩm của mình, tiếp cận khán giả và nhận về những phản hồi, tương tác trực tiếp.

Đối với sinh viên: Giúp sinh viên áp dụng toàn diện các kiến thức đã học (từ phân tích, thiết kế, lập trình) vào thực tế. Việc làm chủ một loạt công nghệ từ frontend (Next.js, React Native) đến backend (Node.js, PostgreSQL) và khả năng tích hợp API bên thứ 3 (Spotify Web API, Gemini API) là minh chứng rõ ràng cho năng lực chuyên môn khi tốt nghiệp.

1.6. Cấu trúc quyền báo cáo:

Nội dung của quyền báo cáo Khóa luận tốt nghiệp được trình bày trong 7 chương:

Chương 1: Giới thiệu: Trình bày tổng quan về bối cảnh và lý do chọn đề tài. Xác định mục tiêu, phạm vi nghiên cứu, các phương pháp thực hiện, ý nghĩa khoa học, thực tiễn và cấu trúc tổng thể của báo cáo.

Chương 2: Cơ sở lý thuyết và công nghệ liên quan: Trình bày các khái niệm, lý thuyết nền tảng về hệ thống gợi ý, các mô hình được sử dụng. Đồng thời, giới thiệu và phân tích các công nghệ chính được sử dụng để xây dựng hệ thống (Next.js, React Native, Node.js, PostgreSQL, Cloudinary, Socket.IO).

Chương 3: Phân tích yêu cầu và thiết kế hệ thống: Tập trung vào việc khảo sát và đặc tả các yêu cầu chức năng, phi chức năng của hệ thống. Xây dựng các mô hình phân tích (Usecase, Sơ đồ lớp) và thiết kế kiến trúc hệ thống (Kiến trúc tổng thể, thiết kế CSDL).

Chương 4: Cài đặt và triển khai hệ thống: Mô tả môi trường phát triển, kiến trúc triển khai, và trình bày chi tiết việc cài đặt các module chức năng chính. Hướng dẫn cài đặt và trình bày một số giao diện và luồng hoạt động tiêu biểu của sản phẩm.

Chương 5: Kiểm thử và đánh giá hệ thống: Trình bày chiến lược kiểm thử (Unit test, Integration test, System test). Xây dựng các kịch bản kiểm thử (test case), trình bày kết quả kiểm thử, đánh giá hiệu quả của hệ thống và nêu ra các hạn chế còn tồn tại.

Chương 6: Kế hoạch khởi nghiệp: Dựa trên sản phẩm đã phát triển, trình bày một kế hoạch dự án (Startup), bao gồm phân tích thị trường, mục tiêu, kế hoạch công việc, thời gian, nhân sự và chi phí.

Chương 7: Kết luận và hướng phát triển: Tóm tắt lại các kết quả mà đề tài đã đạt được, so sánh với mục tiêu ban đầu. Đề xuất các hướng phát triển và nâng cấp cho hệ thống trong tương lai.

Chương 2. CƠ SỞ LÝ THUYẾT VÀ CÔNG NGHỆ LIÊN QUAN

2.1. Node.js và ExpressJS:

Node.js là một runtime JavaScript mã nguồn mở, cho phép xây dựng các ứng dụng phía server một cách hiệu quả. Được xây dựng trên nền tảng V8 JavaScript Engine của Google, Node.js cung cấp khả năng xử lý bất đồng bộ và event-driven, giúp phát triển các ứng dụng có khả năng mở rộng cao. Node.js thường được sử dụng để phát triển các API, microservices và các ứng dụng thời gian thực nhờ khả năng xử lý số lượng lớn yêu cầu đồng thời mà vẫn duy trì hiệu suất cao [1].

ExpressJS (hay Express) là framework ứng dụng web phổ biến nhất của Node.js, được thiết kế để xây dựng các ứng dụng web và API. Nó thường được gọi là framework máy chủ tiêu chuẩn "de facto" cho Node.js. [2]

2.2. React Native:

React Native là một framework mã nguồn mở do Meta (Facebook) phát triển. Nó cho phép các nhà phát triển xây dựng các ứng dụng di động gốc (native) cho cả hai nền tảng Android và iOS chỉ bằng một cơ sở mã nguồn React [3]. Điểm mạnh của React Native là tái sử dụng code, tiết kiệm chi phí và thời gian, có cộng đồng lớn, UI gần với native, hiệu năng tốt nhờ cơ chế bridge kết nối giữa JavaScript và mã native. Tuy nhiên, nó vẫn có một số hạn chế như thiếu module tùy chỉnh, cần hỗ trợ từ lập trình viên native, và có thể gặp vấn đề về khả năng mở rộng trong dự án quá phức tạp.

2.3. Next.js:

Next.js là một React framework dùng để xây dựng các ứng dụng web full-stack [4]. Lập trình viên sử dụng React Components để xây dựng giao diện người dùng, và Next.js cung cấp các tính năng bổ sung và tối ưu hóa [4].

Nó tự động cấu hình các công cụ cấp thấp như bundlers và compilers, giúp lập trình viên tập trung vào việc xây dựng sản phẩm [4]. Các đặc điểm chính bao gồm:

- Rendering linh hoạt: hỗ trợ nhiều cơ chế render khác nhau
- Định tuyến dựa trên tệp (File-based Routing): Cấu trúc thư mục trong app (hoặc pages) tự động tạo ra các tuyến đường (routes).
- Tối ưu hóa hình ảnh, font chữ: Tích hợp sẵn các component giúp tự động tối ưu hóa media để cải thiện hiệu suất.

2.4. PostgreSQL:

PostgreSQL là một hệ thống quản trị cơ sở dữ liệu quan hệ và đối tượng (Object-Relational Database Management System - ORDBMS) mã nguồn mở và miễn phí [5]. Nó được biết đến với sự ổn định, khả năng mở rộng mạnh mẽ, và việc tuân thủ nghiêm ngặt các tiêu chuẩn SQL, đảm bảo tính toàn vẹn dữ liệu. PostgreSQL có khả năng xử lý các truy vấn phức tạp và khối lượng công việc lớn, khiến nó trở thành lựa chọn phổ biến cho các ứng dụng doanh nghiệp và các hệ thống đòi hỏi sự tin cậy cao.

2.5. Socket.IO:

Socket.IO là một thư viện cho phép giao tiếp hai chiều (bidirectional), độ trễ thấp và dựa trên sự kiện (event-based) giữa máy khách và máy chủ [6].

Kết nối Socket.IO có thể được thiết lập thông qua các phương thức truyền tải cấp thấp (low-level transports) khác nhau như: HTTP long-polling, WebSocket, hoặc WebTransport [6]. Socket.IO sẽ tự động chọn phương án tốt nhất tùy thuộc vào khả năng của trình duyệt và điều kiện mạng (ví dụ: một số mạng chặn kết nối WebSocket) [6].

2.6. Expo:

Expo là một nền tảng và bộ công cụ mã nguồn mở (framework) được xây dựng dựa trên React Native, giúp cho việc phát triển ứng dụng Android và iOS trở nên dễ

dàng hơn. Nền tảng này cung cấp các tính năng như định tuyến dựa trên tệp (file-based routing), một thư viện chuẩn các mô-đun gốc (native modules), và nhiều hơn nữa [7].

2.7. TailwindCSS và Nativewind:

Tailwind CSS là một framework CSS theo triết lý utility-first (ưu tiên lớp tiện ích) [8]. Thay vì cung cấp các component dựng sẵn (như btn, card), Tailwind cung cấp hàng ngàn "lớp" (class) tiện ích cấp thấp (ví dụ: pt-4 để thêm padding-top: 1rem, flex để tạo display: flex, text-red-500 để đổi màu chữ).

Lập trình viên có thể "ghép" các lớp tiện ích này trực tiếp trong HTML/JSPX để xây dựng bất kỳ thiết kế nào mà không cần rời khỏi trình soạn thảo mã nguồn hay viết CSS tùy chỉnh [8].

Nativewind là một thư viện cho phép các nhà phát triển sử dụng Tailwind CSS để tạo kiểu (style) cho các component trong React Native [9]. Nó mang triết lý utility-first của Tailwind vào môi trường di động, cho phép lập trình viên sử dụng các lớp (className) giống hệt như trên web (ví dụ: className="flex-1 p-4") thay vì viết các file StyleSheet truyền thống.

Mục tiêu của Nativewind là cung cấp trải nghiệm tạo kiểu nhất quán trên mọi nền tảng, cải thiện hiệu suất và khả năng bảo trì mã nguồn.

2.8. Sequelize (ORM):

Sequelize là một thư viện ORM (Object-Relational Mapper) dựa trên Promise dành cho Node.js, hỗ trợ nhiều hệ CSDL như Postgres, MySQL, MariaDB, và SQLite [10].

Nói đơn giản, ORM đóng vai trò là "người phiên dịch" giữa mã JavaScript (hướng đối tượng) và cơ sở dữ liệu (quan hệ). Thay vì viết các câu lệnh SQL thô (như INSERT INTO...), lập trình viên có thể định nghĩa các "Model" (mô hình) trong JavaScript (ví dụ: class User). Sau đó, họ có thể tương tác với CSDL bằng các phương thức JavaScript trực quan như User.findAll() hay user.save().

2.9. Kiến trúc Client - Server:

Một phong cách cơ bản trong kiến trúc là tách biệt chức năng kỹ thuật giữa giao diện người dùng (frontend) và phía sau (backend), được gọi là kiến trúc hai tầng (two-tier), hoặc client/server (khách/chủ) [11].

Đây là kiến trúc được trích dẫn thường xuyên nhất khi các hệ thống phân tán được thảo luận. Về mặt lịch sử, nó là kiến trúc quan trọng nhất và vẫn là kiến trúc được sử dụng rộng rãi nhất [12].

Chương 3. PHÂN TÍCH YÊU CẦU VÀ THIẾT KẾ HỆ THỐNG

3.1. Khảo sát và thu thập yêu cầu:

3.1.1. Phương pháp thực hiện:

Để xác định chính xác các yêu cầu nghiệp vụ và tính năng cần thiết cho hệ thống "JT-Harmony", nhóm đã áp dụng kết hợp hai phương pháp chính:

- *Phương pháp nghiên cứu tài liệu*: Tìm hiểu các quy trình nghiệp vụ chuẩn của một hệ thống nghe nhạc trực tuyến, các tiêu chuẩn về định dạng âm thanh và bảo mật người dùng.

- *Phương pháp quan sát và trải nghiệm thực tế*: Nhóm đã tiến hành cài đặt, sử dụng và phân tích các ứng dụng nghe nhạc hàng đầu trên thị trường hiện nay (Spotify, Zing MP3, SoundCloud) để đánh giá ưu/nhược điểm và tìm ra khoảng trống thị trường.

3.1.2. Kế hoạch thực hiện

Để đảm bảo yêu cầu hệ thống được xác định chính xác và khách quan, nhóm đã lập và thực hiện 2 kế hoạch khảo sát chi tiết sau:

Kế hoạch 1: Khảo sát hiện trạng các ứng dụng tương tự (Phương pháp Quan sát):

Mục đích: Xác định các tính năng tiêu chuẩn (phát nhạc, playlist) và tìm ra các hạn chế (gap) của các ứng dụng hiện có trên thị trường.

Đối tượng/Nơi thực hiện: Các ứng dụng: Spotify, Zing MP3, SoundCloud trên Google Play và App Store.

Người thực hiện: Nhóm sinh viên thực hiện đề tài (Phan Thị Huỳnh Thư, Lê Võ Duy Tuấn) .

Phương pháp: Cài đặt ứng dụng, đóng vai người dùng trải nghiệm, ghi chép lại các luồng nghiệp vụ và chụp ảnh màn hình các giao diện chức năng.

Kế hoạch 2: Khảo sát nhu cầu người dùng (Phương pháp Bảng câu hỏi):

Mục đích: Thu thập ý kiến về thói quen nghe nhạc, nhu cầu tương tác xã hội khi nghe nhạc, và mong muốn về tính năng chia sẻ cover/sáng tác.

Đối tượng cung cấp câu trả lời: Sinh viên trường Đại học Công nghiệp TP.HCM và các bạn trẻ yêu thích âm nhạc (độ tuổi 18-25).

Người thực hiện: Nhóm sinh viên thực hiện đề tài (Phan Thị Huỳnh Thư, Lê Võ Duy Tuấn) tạo biểu mẫu và gửi đi.

Phương pháp: Sử dụng Google Form để tạo bảng câu hỏi trắc nghiệm trực tuyến và gửi link khảo sát qua các nhóm mạng xã hội. (Chi tiết bảng câu hỏi đính kèm tại Phụ lục).

3.1.3. Kết quả thu thập và phân tích:

Dựa trên quá trình khảo sát, nhóm đã rút ra các kết quả phân tích sau:

Về tính năng nghe nhạc: Hầu hết các ứng dụng đều đáp ứng tốt nhu cầu nghe nhạc chất lượng cao, tạo playlist và nghe offline. Đây là các tính năng nền tảng bắt buộc phải có.

Về tính năng gợi ý: Spotify và YouTube Music làm rất tốt việc gợi ý dựa trên hành vi. Zing MP3 tập trung vào bảng xếp hạng (Trending). Hệ thống cần kết hợp cả hai: gợi ý theo sở thích cá nhân và cập nhật xu hướng.

Về tính năng xã hội và chia sẻ (Vấn đề tồn tại):

Các ứng dụng lớn (Spotify, Zing MP3) chủ yếu là nền tảng phát hành (Streaming), tính năng tương tác xã hội rất hạn chế (thường chỉ là chia sẻ đường link sang Facebook/Instagram). Người dùng không thể thảo luận trong ứng dụng và cũng chỉ có thể nghe nhạc từ nền tảng của ứng dụng mà không được phép đăng tải, upload nhạc của chính mình.

SoundCloud cho phép tương tác tốt nhưng giao diện chưa thực sự thân thiện với người dùng phổ thông.

Kết luận yêu cầu hệ thống: Từ phân tích trên, nhóm xác định yêu cầu cốt lõi của JT-Harmony là phải xây dựng một mô hình lai (Hybrid): Một trình phát nhạc chuyên nghiệp (như Spotify). Tích hợp sâu tính năng Mạng xã hội (như Facebook):

Cho phép upload Cover/Sáng tác, bình luận thời gian thực (Realtime), và theo dõi lẫn nhau.

3.2. Đặc tả yêu cầu chức năng, phi chức năng:

3.2.1. Yêu cầu chức năng:

Xây dựng ứng dụng đáp ứng các chức năng cơ bản sau:

- *Đăng nhập/đăng ký*: có thể đăng nhập bằng tài khoản email đã đăng ký, google hoặc facebook.
- *Chức năng nghe nhạc cơ bản*: phát nhạc, tạm dừng, phát kế, tiếp, phát bài trước, lặp lại.
- *Quản lý danh sách phát cá nhân*: thêm, xóa, sửa, thêm bài hát, xóa bài hát khỏi danh sách phát.
- *Quản lý tài khoản*: cập nhật thông tin, liên kết tài khoản, khóa tài khoản.
- *Tìm kiếm*: tìm bài hát, nghệ sĩ, album, danh sách phát, người dùng,
- *Xem thông tin*: xem thông tin bài hát, nghệ sĩ, người dùng khác.
- *Quản lý bài đăng*: tạo bài đăng, tạo bản cover, chỉnh sửa, xóa.
- *Quản lý bình luận*: tạo mới, xóa, sửa.
- *Tương tác/chia sẻ*: thích, báo cáo, chia sẻ bài đăng, bài hát, bình luận của người khác.

3.2.2. Yêu cầu phi chức năng

Hiệu năng (Performance)

- Thời gian phản hồi API trung bình dưới 200ms.
- Tính năng phát nhạc (streaming) phải mượt mà, độ trễ thấp, hỗ trợ bộ nhớ đệm (caching) để nghe khi mạng yếu.
- Hỗ trợ nhiều người dùng thao tác đồng thời mà không gây tắc nghẽn hệ thống (nhờ kiến trúc Non-blocking I/O của Node.js).

Bảo mật (Security)

- Mật khẩu người dùng phải được mã hóa một chiều (sử dụng Bcrypt) trước khi lưu vào CSDL.
- Giao tiếp giữa Client và Server phải được xác thực an toàn thông qua cơ chế JWT (JSON Web Token).
- Dữ liệu nhạy cảm và file media phải được lưu trữ bảo mật trên Cloud (Cloudinary) và CSDL (PostgreSQL).

Tính khả dụng và Tin cậy (Reliability)

- Hệ thống hoạt động ổn định 24/7.
- Cơ chế tự động kết nối lại (Reconnection) của Socket.IO giúp duy trì tính năng realtime (chat, thông báo) ngay cả khi kết nối mạng chập chờn.

Khả năng mở rộng (Scalability): Kiến trúc hệ thống được thiết kế dạng mô-đun (Client-Server), dễ dàng nâng cấp hoặc thêm mới các tính năng (như thanh toán, livestream) trong tương lai mà không ảnh hưởng đến các chức năng hiện có.

3.2.3. Yêu cầu liên quan đến AI:

Tích hợp Gemini API để phân tích và cá nhân hóa trải nghiệm người dùng, hệ thống gợi ý/đề xuất nhạc cần đáp ứng những yêu cầu sau:

- *Tính cá nhân hóa:* phân tích lịch sử nghe nhạc, danh sách yêu thích, thể loại yêu thích, phân tích cảm xúc bài hát mà người dùng đã nghe, dựa trên hoạt động, tâm trạng của người nghe để đề xuất nội dung mới.
- *Tính chính xác:* Hệ thống gợi ý phải đề xuất được nội dung có khả năng tương thích cao với sở thích của người dùng.

3.2.4. Quy tắc quản lý tài khoản và định danh:

Tính duy nhất: Mỗi địa chỉ email chỉ được phép đăng ký một tài khoản duy nhất trong hệ thống.

Xác thực: Tài khoản đăng ký mới bắt buộc phải xác thực qua email (hoặc xác thực qua Google/Facebook OAuth) để đảm bảo danh tính hợp lệ.

Trạng thái hoạt động: Tài khoản bị Quản trị viên khóa (do vi phạm) sẽ không thể đăng nhập hoặc thực hiện bất kỳ tác vụ nào cho đến khi được mở khóa.

3.2.5. Quy tắc kiểm duyệt cộng đồng:

Các bình luận hoặc bài đăng chứa từ ngữ thô tục hoặc vi phạm tiêu chuẩn cộng đồng có thể bị Quản trị viên xóa mà không cần báo trước.

Người dùng lạm dụng tính năng (spam bình luận, spam upload) sẽ bị cảnh cáo hoặc khóa tài khoản tạm thời.

3.3. Các chức năng chính của hệ thống:

3.3.1. Các nhóm người dùng – quyền hạn trên hệ thống

Hệ thống tập trung phát triển một nền tảng nghe nhạc phục vụ 2 đối tượng chính: Quản trị viên (Sử dụng trang web quản trị) và Người nghe nhạc (Sử dụng ứng dụng di động).

Bảng 3.1: Danh sách tác nhân mô tả

STT	Tác nhân	Mô tả chức năng
1	Người nghe nhạc	Có thể tìm kiếm các bài hát có trong ứng dụng, nghe nhạc, đăng bài và tương tác, giao lưu với những người dùng khác.
2	Quản trị viên	Là người quản lý trực tiếp ứng dụng, có thể xem thống kê các bài hát theo lượt chia sẻ, lượt nghe. Ngoài ra còn có thể xem xét, tiếp nhận những phản hồi từ người dùng để kiểm soát thông tin và xử lý các vi phạm cộng đồng.

Bảng 3.2: Danh sách chức năng và quyền hạn người dùng

Tên chức năng	Quản trị viên	Người nghe nhạc
Đăng ký tài khoản	Không	Có
Đăng nhập	Có	Có
Đăng xuất	Có	Có
Khôi phục mật khẩu	Không	Có
Đổi mật khẩu	Không	Có
Liên kết tài khoản	Không	Có
Cập nhật thông tin	Không	Có
Tìm kiếm bài hát	Không	Có
Xem lịch sử tìm kiếm	Không	Có
Chức năng nghe nhạc	Không	Có
Xem lịch sử nghe nhạc	Không	Có
Tạo danh sách phát	Không	Có
Đổi tên danh sách phát	Không	Có
Xóa danh sách phát	Có	Có
Thêm bài hát vào danh sách phát	Không	Có
Xóa bài hát khỏi danh sách phát	Không	Có
Quản lý danh mục yêu thích	Không	Có
Theo dõi nghệ sĩ	Không	Có
Xem thông tin nghệ sĩ	Không	Có
Chia sẻ nhạc	Không	Có
Chia sẻ bài đăng	Không	Có
Đăng bài	Không	Có
Xóa bài đăng	Có	Có
Bình luận	Không	Có
Bày tỏ cảm xúc	Không	Có
Theo dõi người khác	Không	Có
Thêm bài hát	Có	Không

Xóa bài hát	Có	Không
Cập nhật thông tin bài hát	Có	Không
Thêm nghệ sĩ	Có	Không
Xóa nghệ sĩ	Có	Không
Cập nhật thông tin nghệ sĩ	Có	Không
Cảnh cáo vi phạm người dùng	Có	Không
Khóa tài khoản người dùng	Có	Có
Xem thống kê	Có	Không

Chức năng chung:

Đăng nhập: Chức năng cho phép người sử dụng đăng nhập vào ứng dụng. Khi thực hiện đăng nhập lần đầu, ứng dụng sẽ điều hướng tới trang khảo sát để thu thập thêm thông tin cá nhân và sở thích nghe nhạc của người dùng.

Đăng xuất: Chức năng cho phép người sử dụng đăng xuất tài khoản khỏi ứng dụng.

Chức năng của người nghe nhạc:

Đăng ký tài khoản: Chức năng cho phép người nghe tạo tài khoản mới. Người dùng có thể lựa chọn hình thức đăng ký bằng email/password hoặc bằng Google, Facebook. Khi người dùng đăng ký bằng email và Google sẽ cần xác thực email. Còn đăng ký bằng Facebook thì sẽ thực hiện xác thực email sau khi đăng nhập vào ứng dụng.

Đổi mật khẩu: Người dùng có thể thực hiện đổi mật khẩu sau khi đăng nhập vào ứng dụng. Khi đổi mật khẩu, người dùng cần nhập mật khẩu cũ, mật khẩu mới và xác nhận lại mật khẩu mới.

Khôi phục mật khẩu: Chức năng cho phép người dùng khôi phục mật khẩu khi quên. Khi khôi phục, người dùng nhập email khôi phục tài khoản. Ứng dụng sẽ gửi một mã xác thực đến email đó. Sau khi nhập mã xác thực thành công, người dùng có thể đặt mật khẩu mới.

Liên kết tài khoản: Chức năng cho phép người dùng có thể liên kết các tài khoản google và facebook lại với nhau để có thể đăng nhập nhanh chóng.

Cập nhật thông tin cá nhân: Người dùng có thể thay đổi avatar, nick name, tiểu sử và vài thông tin khác.

Tìm kiếm: Chức năng cho phép người dùng tìm kiếm bài hát theo tên bài hát, tên album, tên nghệ sĩ, thể loại hoặc người dùng khác trong ứng dụng.

Xem lịch sử tìm kiếm: Người dùng có thể xem lại lịch sử tìm kiếm trong vòng 7 ngày.

Chức năng nghe nhạc: Chức năng cho phép người dùng có thể phát bài hát, tạm dừng, tua tới/lùi bài hát, lặp lại, phát ngẫu nhiên, bỏ qua bài hát, xem lời bài hát và có thể nghe nhạc trực tuyến hoặc ngoại tuyến.

Xem lời bài hát: Người dùng khi đang nghe một bài hát, có thể chọn xem đầy đủ lời bài hát.

Tạo danh sách phát: Người dùng có thể tạo danh sách phát cá nhân theo nhu cầu và sở thích. Khi tạo danh sách phát, người dùng cần nhập tên danh sách phát.

Đổi tên danh sách phát: Người dùng có thể đổi tên danh sách phát khi có nhu cầu. Khi đổi tên, người dùng chỉ cần nhập tên mới của danh sách phát.

Xóa danh sách phát: Người dùng chọn chức năng “Xóa danh sách phát” và chọn “Đồng ý” khi ứng dụng yêu cầu xác nhận hành vi xóa danh sách phát.

Thêm bài hát vào danh sách phát: Chức năng cho phép thêm bài hát mới vào danh sách phát của người dùng. Người dùng có thể thêm bài hát từ màn hình đề xuất bài hát của hệ thống hoặc từ một danh sách phát, album, lựa chọn mở rộng của bài hát.

Xóa bài hát khỏi danh sách phát: Người dùng có thể xóa bài hát ra khỏi danh sách phát khi không còn nhu cầu hoặc sự yêu thích.

Quản lý danh mục yêu thích: Người dùng có thể thêm hoặc bài hát, album, danh sách phát vào danh sách yêu thích.

Theo dõi nghệ sĩ: Người dùng có thể nhấn theo dõi ca sĩ/ nhạc sĩ mà mình yêu thích.

Xem thông tin nghệ sĩ: Người dùng có thể chọn nghệ sĩ mình muốn xem thông tin và chọn xem thông tin chi tiết của nghệ sĩ đó.

Xem lịch sử nghe nhạc gần đây: Người dùng có thể xem lại lịch sử nghe nhạc gần đây trong phạm vi 7 ngày gần nhất.

Chia sẻ nhạc: Người dùng có thể chia sẻ bài hát hoặc danh sách phát yêu thích qua mạng xã hội (Facebook, Instagram, Zalo,...) hoặc thông qua link, mã QR hoặc có thể gửi cho bạn bè trong ứng dụng.

Chia sẻ bài đăng: Người dùng có thể chia sẻ bài đăng yêu thích cho bạn bè trong ứng dụng.

Đăng bài: Người dùng sử dụng mạng xã hội có thể đăng những video, bài hát mình cover, sáng tác lên mạng.

Bình luận: Người dùng có thể bình luận, tương tác với bài đăng của người dùng khác.

Bày tỏ cảm xúc: Người dùng có thể bày tỏ cảm xúc với bài hát, bài đăng và bình luận của người dùng khác.

Theo dõi người khác: Người dùng có thể nhấn theo những người dùng khác để cập nhật thông tin upload của người đó.

Chức năng của quản trị viên:

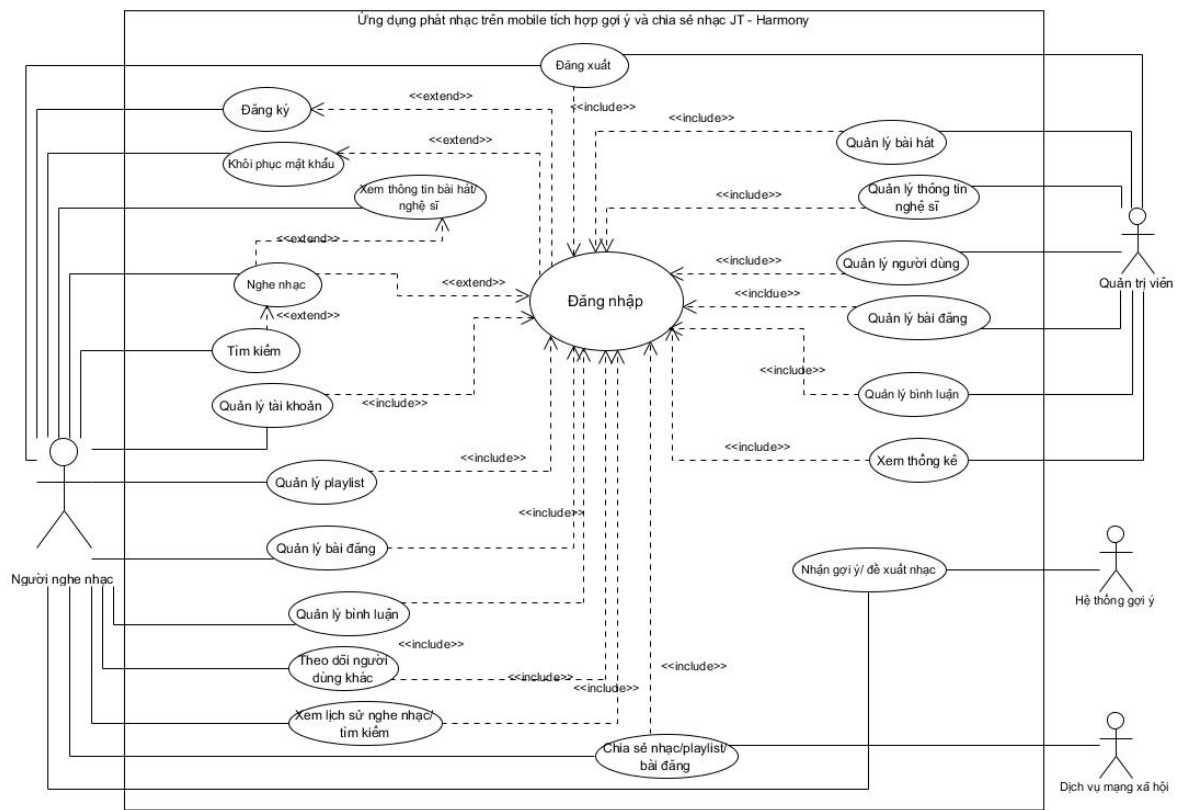
Quản lý thông tin bài hát/thể loại/nghệ sĩ: Quản trị viên được quyền thêm, xóa, sửa bài hát, thể loại.

Quản lý người dùng: Quản trị viên có thể xem xét các báo cáo, cảnh cáo hoặc khóa tài khoản khi người dùng vi phạm nguyên tắc cộng đồng.

Xem thống kê bài hát: Quản trị viên có thể xem được thống kê các bài hát về lượt nghe, lượt chia sẻ.

Quản lý bài đăng: Quản trị viên có thể xem thống kê bài đăng, thêm, xóa, sửa thông tin bài đăng đó.

3.3.2. Usecase tổng quát:



Hình 3.1: Use case tổng quát

Dựa trên sơ đồ use case tổng quát, hệ thống gồm một số use case chính sau:

ID	Tên use case	Ghi chú
UC001	Đăng ký	
UC002	Đăng nhập	
UC003	Tạo danh sách phát	
UC004	Tạo bài đăng/cover	
UC005	Nhắn tin	
UC006	Nghe nhạc	
UC007	Bình luận	
UC008	Theo dõi nghệ sĩ	
UC009	Theo dõi người dùng	
UC010	Chia sẻ bài đăng/bài hát	

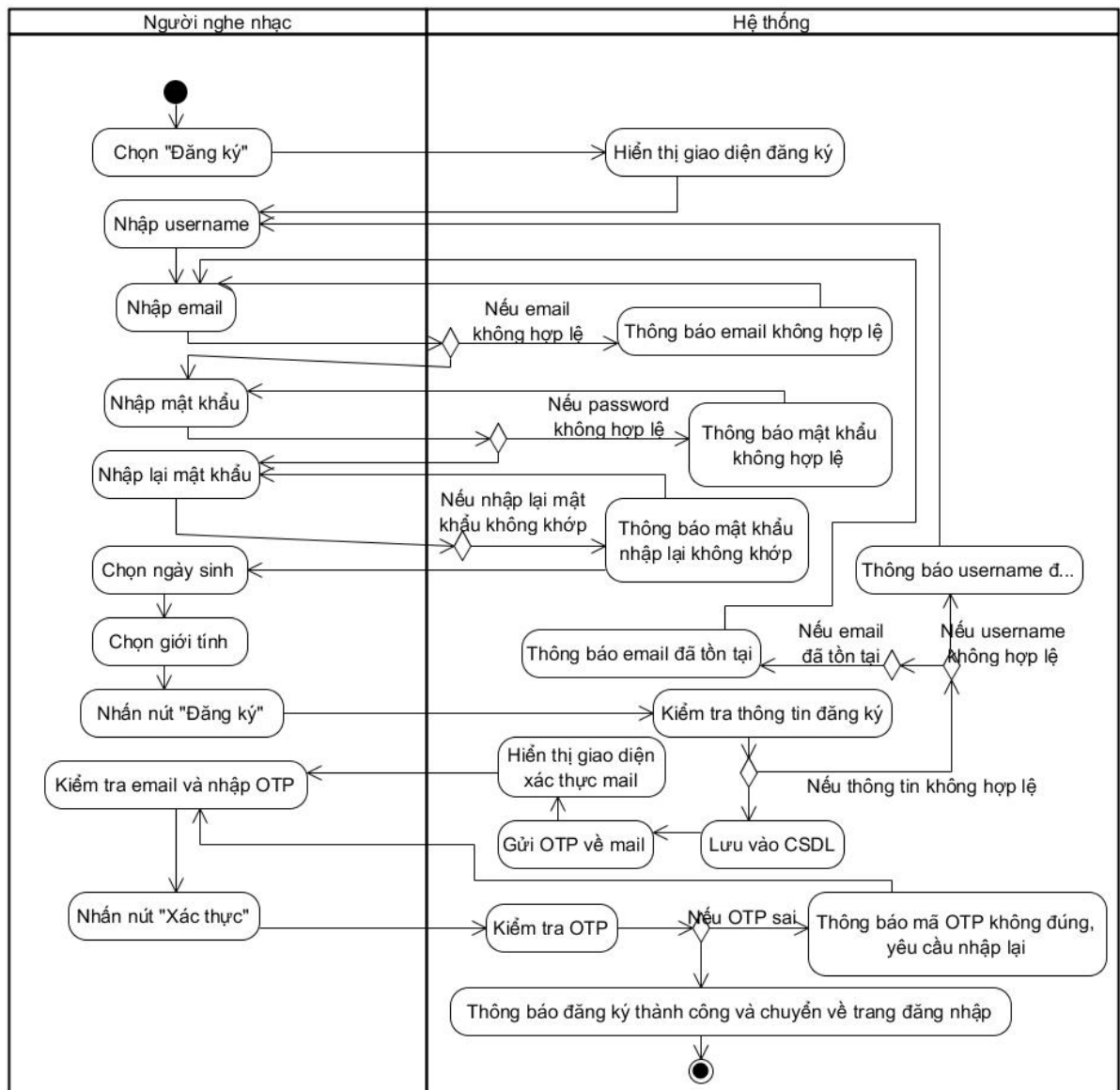
3.3.2.1. UC001 - Đăng ký:

Bảng 3.3: Đặc tả UC001 - Đăng ký

Use case: Đăng ký	
Mục đích: Đăng ký tài khoản	
Mô tả: Chức năng cho phép người nghe nhạc đăng ký tài khoản.	
Tác nhân: Người nghe nhạc	
Điều kiện trước: Nhấn vào nút Đăng ký	
Điều kiện sau: Đăng ký tài khoản thành công	
Luồng sự kiện chính (main flows)	
Tác nhân	Hệ thống
1. Chọn chức năng “Đăng ký”	2. Hiện thị giao diện đăng ký
3. Nhập username	
4. Nhập email	
5. Nhập mật khẩu	
6. Nhập lại mật khẩu	
7. Chọn ngày sinh	
8. Chọn giới tính	
9. Nhấn nút “Đăng ký”	10. Kiểm tra thông tin đăng ký
	11. Lưu thông tin vào CSDL.
	12. Gửi OTP xác thực về email đăng ký và hiện thị giao diện nhập mã OTP.
13. Kiểm tra email và nhập OTP.	
14. Nhấn nút “Xác thực”.	15. Kiểm tra OTP.
	16. Thông báo đăng ký thành công và chuyển về trang đăng nhập.
Luồng sự kiện thay thế (alternate flow)	
	4.1. Thông báo email không hợp lệ.
4.2. Quay lại bước 4.	

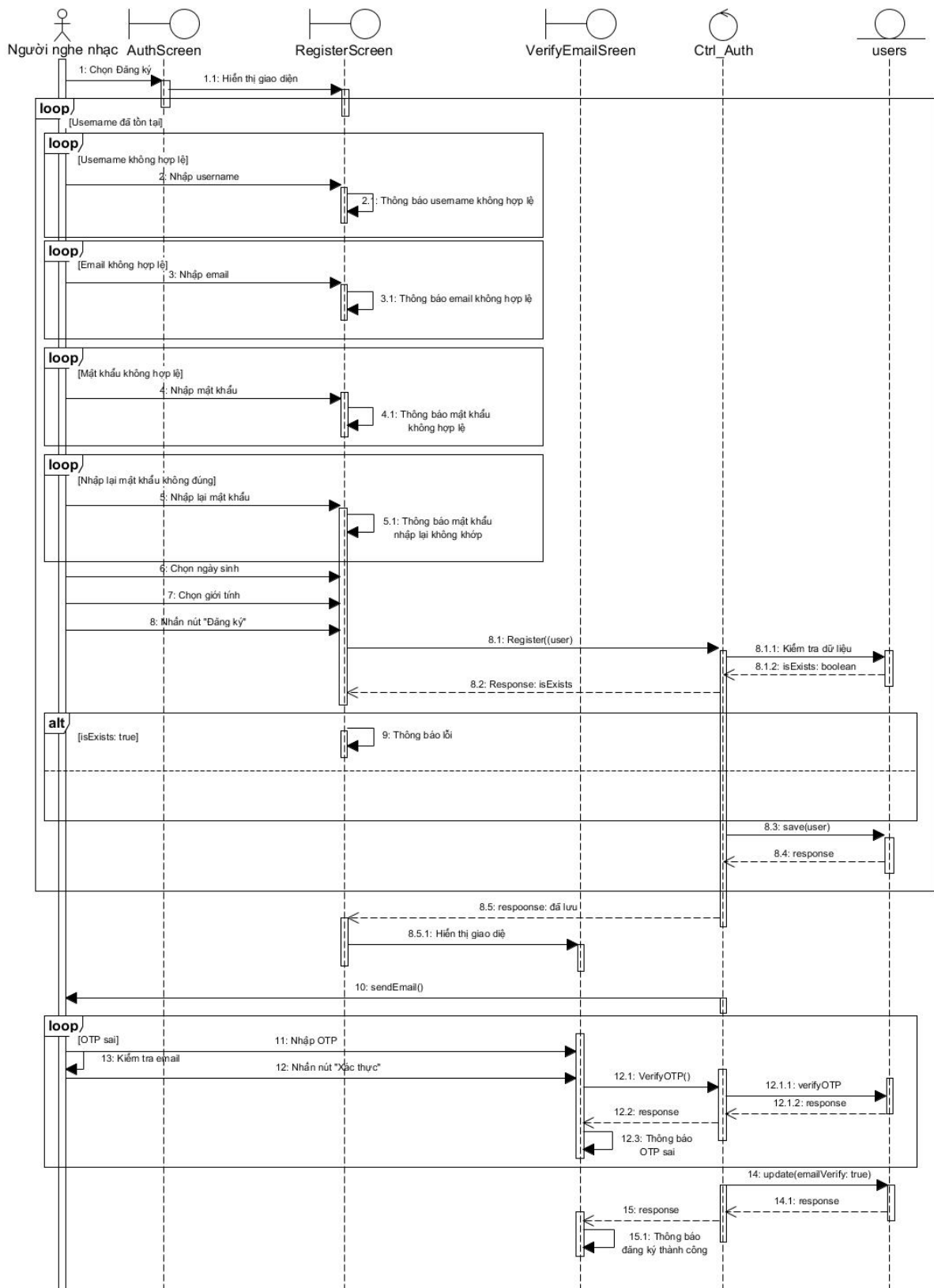
	5.1. Thông báo mật khẩu không hợp lệ.
5.2. Quay lại bước 5.	
	6.1. Thông báo mật khẩu nhập lại không khớp.
6.2. Quay lại bước 6.	
	10.1.1. Thông báo username đã tồn tại.
10.1.2. Quay lại bước 3	
	10.2.1. Thông báo email đã tồn tại trong hệ thống.
10.2.2. Quay lại bước 4.	
	15.1. Thông báo mã OTP không đúng, yêu cầu nhập lại.
15.2. Quay lại bước 13.	

Sơ đồ Activity:



Hình 3.2: Sơ đồ hoạt động - UC001 Đăng ký

Sơ đồ trình tự (sequence diagram):



Hình 3.3: Sơ đồ trình tự -UC001 Đăng ký

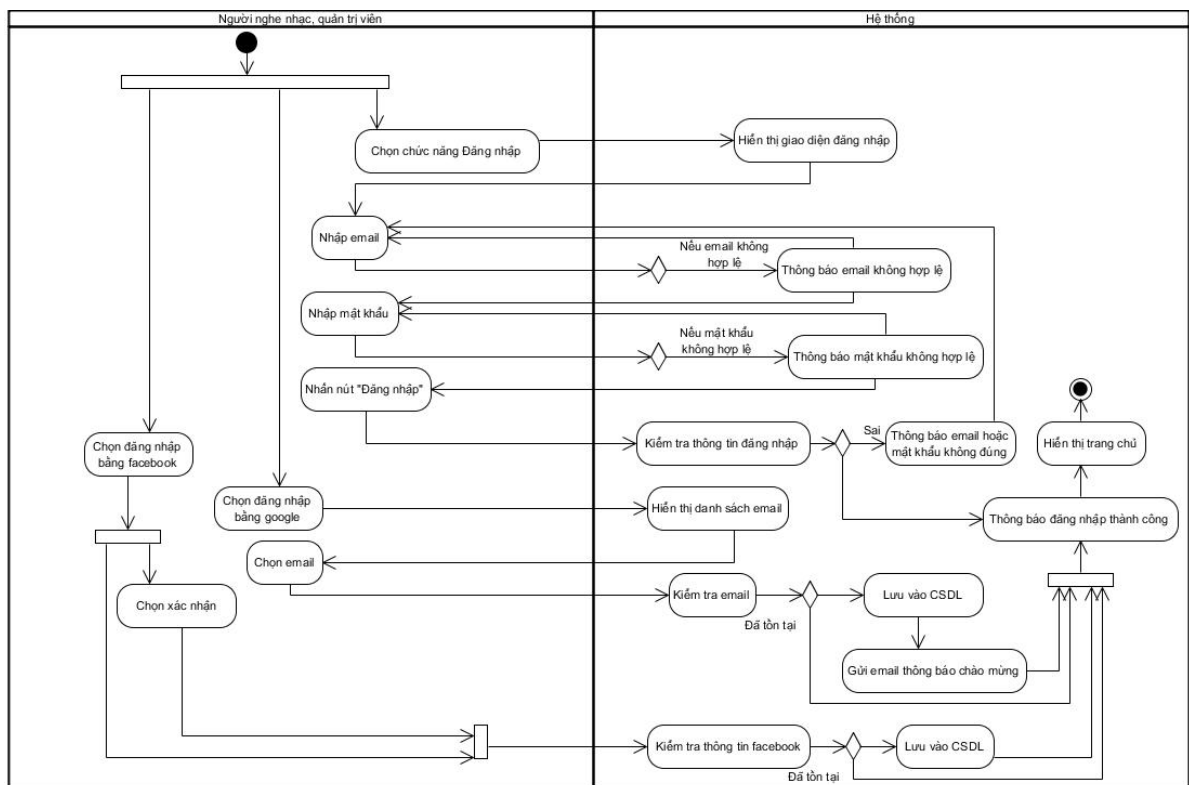
3.3.2.2. UC002 - Đăng nhập:

Bảng 3.4: Đặc tả UC002 - Đăng nhập

Use case: Đăng nhập	
Mục đích: Đăng nhập	
Mô tả: Chức năng cho phép người dùng đăng nhập vào hệ thống	
Tác nhân: Người nghe nhạc, quản trị viên	
Điều kiện trước: Có tài khoản	
Điều kiện sau: Đăng nhập thành công	
Luồng sự kiện chính (main flows)	
Tác nhân	Hệ thống
1. Chọn chức năng “Đăng nhập”	2. Hiện thị giao diện đăng nhập.
3. Nhập email	
4. Nhập password	
5. Nhấn nút “Đăng nhập”	6. Kiểm tra thông tin đăng nhập
	7. Thông báo đăng nhập thành công
	8. Hiện thị giao diện trang chủ ứng dụng.
Luồng sự kiện thay thế (alternate flow)	
1.1.1. Chọn đăng nhập bằng Google	1.1.2. Hiện thị danh sách email.
1.1.3. Chọn tài khoản	1.1.4. Hệ thống kiểm tra email, nếu email chưa tồn tại trong CSDL thì lưu và gửi mail xác nhận cho người dùng.
	1.1.5. Đi đến bước 7
1.2.1. Chọn đăng nhập bằng facebook	1.2.2 Hệ thống kiểm tra thông tin facebook, nếu chưa tồn tại thì yêu cầu xác nhận.
1.2.3 Xác nhận (nếu có yêu cầu)	1.2.4. Đi tới bước 6

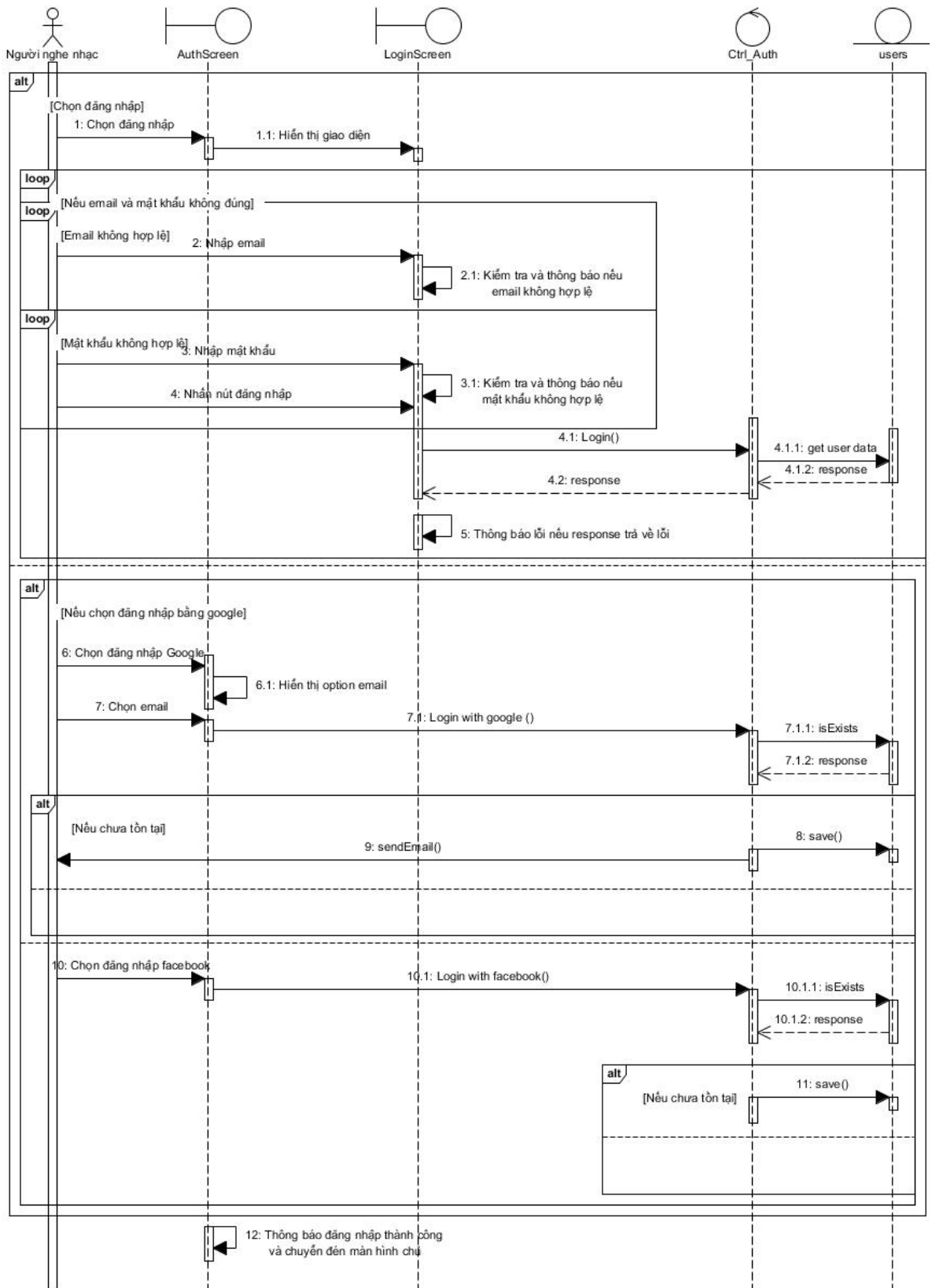
	3.3.1 Thông báo email không hợp lệ và yêu cầu nhập lại.
3.3.2. Quay lại bước 3	
	4.1. Thông báo mật khẩu không hợp lệ và hiển thị yêu cầu nhập lại.
4.2. Quay lại bước 4.	
	6.1. Thông báo email hoặc mật khẩu không khớp. Yêu cầu nhập lại.
6.2. Quay lại bước 3.	

Sơ đồ activity:



Hình 3.4: Sơ đồ hoạt động UC002 - Đăng nhập

Sơ đồ sequence:



Hình 3.5: Sơ đồ trình tự UC002 - Đăng nhập

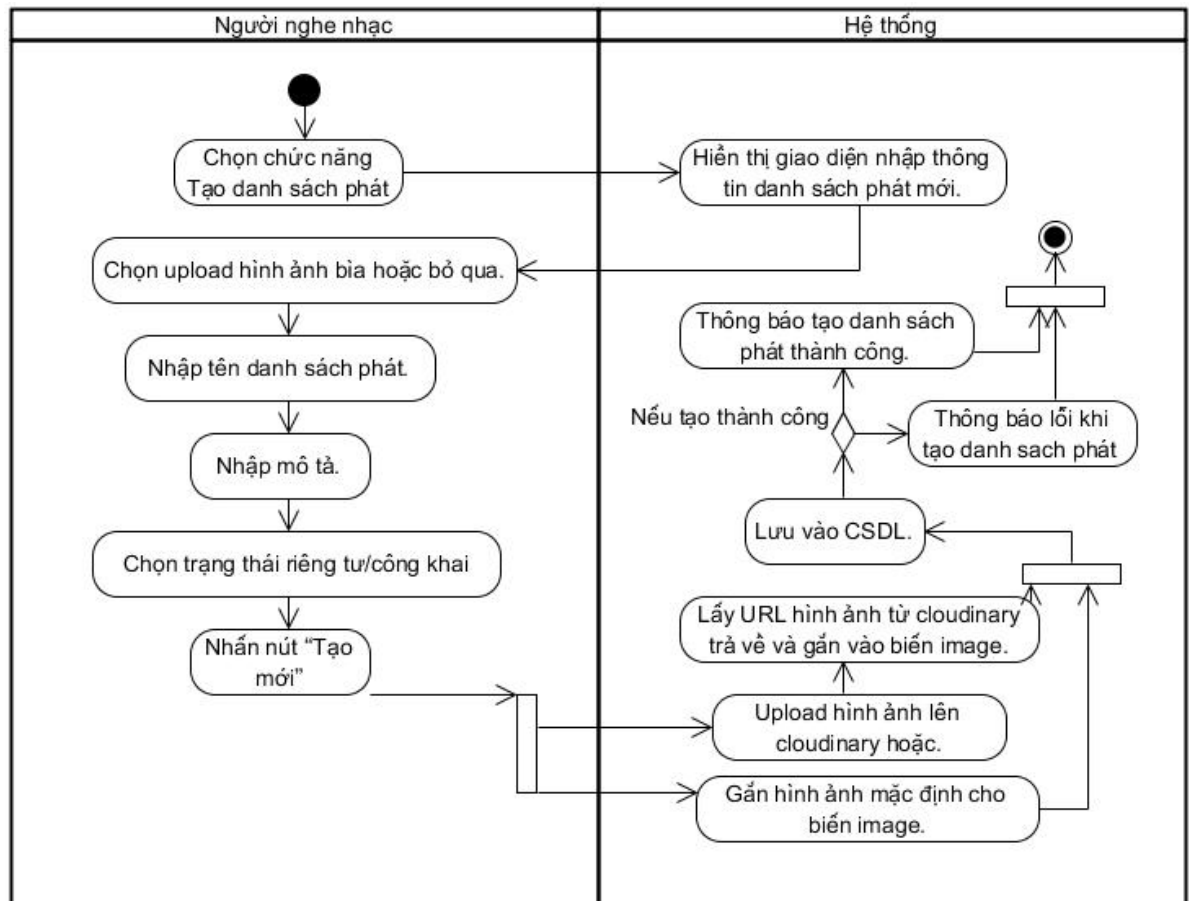
3.3.2.3. UC003 - Tạo danh sách phát:

Bảng 3.5: Đặc tả UC003 - Tạo danh sách phát

Use case: Tạo danh sách phát	
Mục đích: Tạo danh sách phát	
Mô tả: Chức năng cho phép người nghe nhạc tạo danh sách phát mới	
Tác nhân: Người nghe nhạc	
Điều kiện trước: Đã đăng nhập	
Điều kiện sau: Tạo một danh sách phát mới thành công	
Luồng sự kiện chính (main flows)	
Tác nhân	Hệ thống
1. Chọn chức năng “Tạo danh sách phát”	2. Hệ thống hiển thị giao diện nhập thông tin danh sách phát mới.
3. Chọn upload hình ảnh bìa hoặc bỏ qua.	4. Hiển thị hình ảnh lên.
5. Nhập tên danh sách phát.	
6. Nhập mô tả.	
7. Chọn trạng thái riêng tư/công khai.	
8. Nhấn nút “Tạo mới”	9. Upload hình ảnh lên clouinary hoặc.
	10. Lấy URL hình ảnh từ clouinary trả về và gắn vào biến image.
	11. Lưu vào CSDL.
	12. Thông báo tạo danh sách phát thành công.
Luồng sự kiện thay thế (alternate flow)	
	9.1. Kiểm tra không có hình ảnh được upload
	9.2. Gắn hình ảnh mặc định cho biến

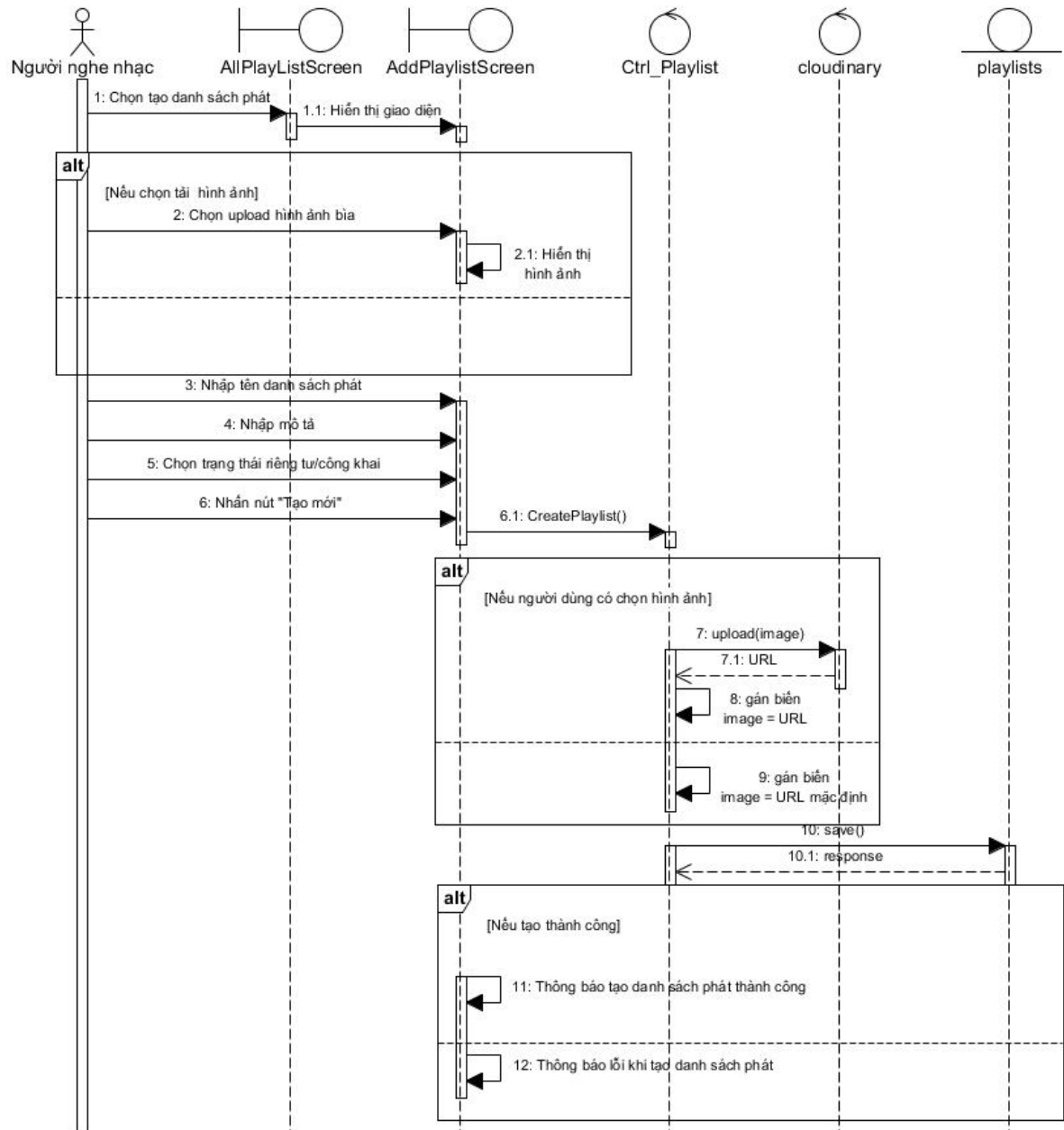
	image.
	9.3. Đi tới bước 11
	12.1 Thông báo lỗi tạo danh sách phát.

Sơ đồ activity:



Hình 3.6: Sơ đồ hoạt động UC003 - Tạo danh sách phát

Sơ đồ sequence:



Hình 3.7: Sơ đồ trình tự UC003 - Tạo danh sách phát

3.3.2.4. UC004 - Tạo bài đăng:

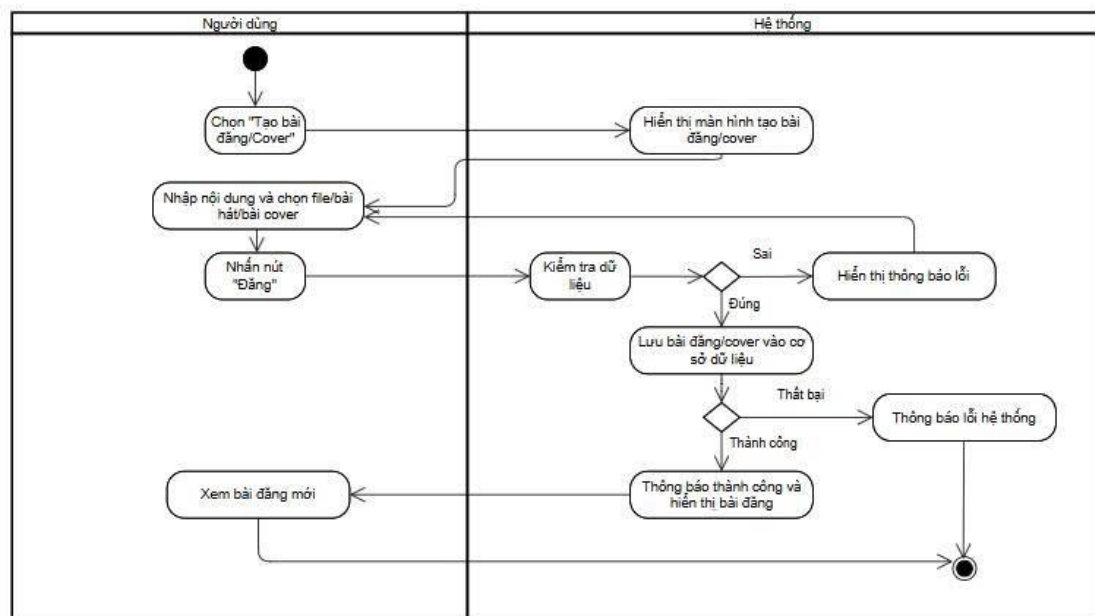
Bảng 3.6: Đặc tả UC004 - Tạo bài đăng/Cover

Use case: Tạo bài đăng/Cover
Mô tả: Cho phép người dùng chia sẻ nội dung (văn bản, hình ảnh, video) hoặc bản thu âm cover bài hát lên hệ thống.

Tác nhân: Người nghe nhạc.	
Tiền điều kiện (Pre-condition): <ul style="list-style-type: none">- Người dùng đã có tài khoản và đăng nhập vào ứng dụng.- Thiết bị có kết nối Internet.- Người dùng đang ở trang mạng xã hội.	
Hậu điều kiện (Post-condition): <ul style="list-style-type: none">- Bài đăng/cover được tạo thành công và xuất hiện trong danh sách bài đăng.- Người dùng có thể xem lại bài đăng của mình, người khác có thể xem, thích, bình luận và chia sẻ.	
Luồng sự kiện chính (Main flows):	
Tác nhân	Hệ thống
1. Người dùng chọn nút "Tạo bài đăng" hoặc "Đăng cover".	2. Hệ thống hiển thị màn hình tạo bài đăng.
3. Người dùng nhập nội dung, chọn file đính kèm (ảnh/video/âm thanh) hoặc chọn bài hát muốn cover.	
4. Người dùng nhấn nút "Đăng".	5. Hệ thống kiểm tra tính hợp lệ của dữ liệu (định dạng file, nội dung rỗng...).
	6. Hệ thống thực hiện lưu bài đăng vào cơ sở dữ liệu.
	7. Hệ thống hiển thị thông báo "Đăng bài thành công" và hiển thị bài đăng vừa tạo.
8. Người dùng xem bài đăng mới và kết thúc.	
Luồng sự kiện thay thế (Alternate flow)	
	5.1. Hệ thống hiển thị thông báo lỗi tương ứng (ví dụ: "Nội dung không

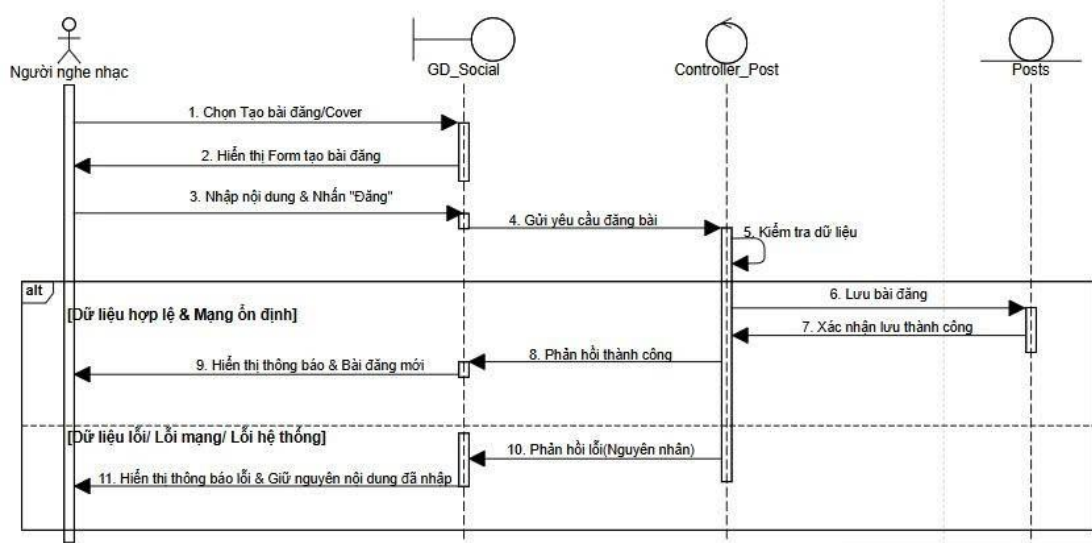
	được để trống" hoặc "File quá lớn").
5.2. Người dùng xác nhận và chỉnh sửa lại dữ liệu.	5.3. Hệ thống quay lại bước 4.
Luồng sự kiện ngoại lệ (Exception flow):	
	6.1. Hệ thống gặp lỗi khi lưu vào cơ sở dữ liệu (Database Error) hoặc mất kết nối mạng.
	6.2. Hệ thống hiển thị thông báo lỗi hệ thống: "Đăng bài thất bại, vui lòng thử lại sau".
6.3. Người dùng xác nhận và kết thúc.	

Sơ đồ activity:



Hình 3.8: Sơ đồ hoạt động UC004 - Tạo bài đăng/Cover

Sơ đồ sequence:



Hình 3.9: Sơ đồ trình tự UC004 - Tạo bài đăng/Cover

3.3.2.5. UC005 - Nhắn tin:

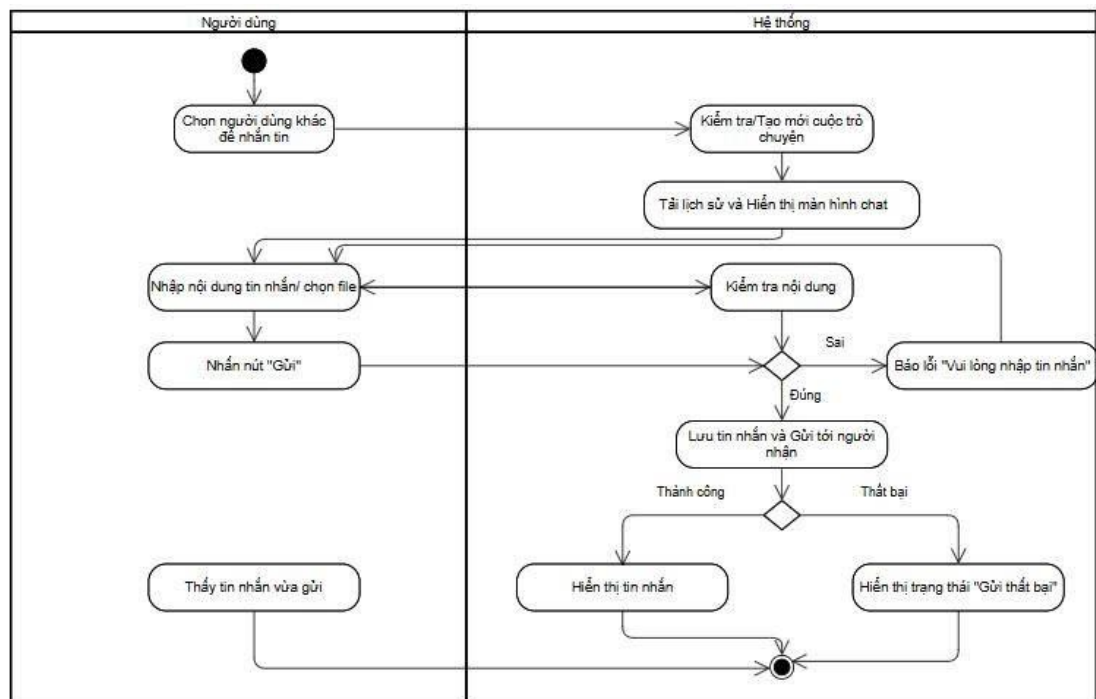
Bảng 3.7: Đặc tả UC005 - Nhắn tin

Use case: Nhắn tin	
Mô tả: Cho phép người dùng gửi và nhận tin nhắn với người dùng khác trong hệ thống để trò chuyện, trao đổi thông tin.	
Tác nhân: Người nghe nhạc.	
Tiền điều kiện (Pre-condition): <ul style="list-style-type: none"> - Người dùng đã có tài khoản và đăng nhập vào ứng dụng. - Thiết bị có kết nối Internet. - Người dùng đã chọn được người muốn trò chuyện (từ danh sách theo dõi). 	
Hậu điều kiện (Post-condition): <ul style="list-style-type: none"> - Tin nhắn mới được lưu vào cơ sở dữ liệu và hiển thị trong cuộc trò chuyện của cả hai bên. - Lịch sử cuộc trò chuyện được cập nhật. 	
Luồng sự kiện chính (main flows):	
Tác nhân	Hệ thống
1. Người dùng chọn một người dùng	2. Hệ thống kiểm tra cuộc trò chuyện

khác và nhấn nút "Nhắn tin" (hoặc chọn cuộc trò chuyện có sẵn).	giữa hai người (nếu chưa có thì tạo mới) và tải lịch sử tin nhắn cũ (nếu có).
	3. Hệ thống hiển thị màn hình chat với lịch sử tin nhắn.
4. Người dùng nhập nội dung tin nhắn vào ô nhập liệu.	
5. Người dùng nhấn nút "Gửi".	
	6. Hệ thống kiểm tra tính hợp lệ của tin nhắn (không rỗng, không vi phạm quy tắc).
	7. Hệ thống lưu tin nhắn vào cơ sở dữ liệu và gửi tin nhắn ngay lập tức tới người nhận.
	8. Hệ thống cập nhật giao diện, hiển thị tin nhắn vừa gửi vào khung chat của cả người gửi và người nhận.
9. Người dùng nhìn thấy tin nhắn mới và kết thúc.	
Luồng sự kiện thay thế (Alternate flow)	
	6.1 Hệ thống kiểm tra thấy nội dung tin nhắn rỗng.
	6.2 Hệ thống không gửi tin nhắn và hiển thị thông báo "Vui lòng nhập nội dung".
	6.3 Quay lại bước 4.
Luồng sự kiện ngoại lệ (Exception flow)	
	7.1. Hệ thống không thể lưu hoặc gửi tin nhắn đi do lỗi kết nối mạng hoặc

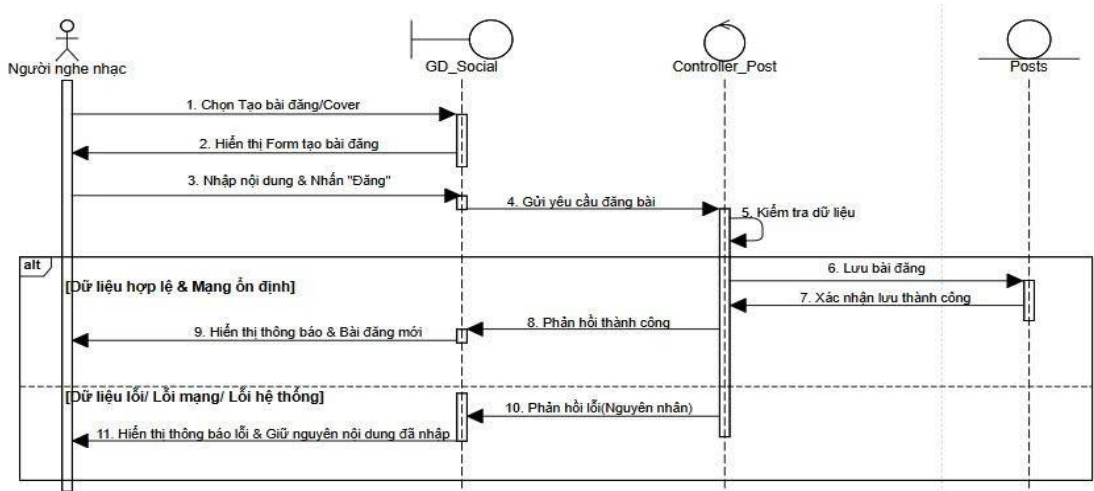
	lỗi Server.
	7.2. Hệ thống hiển thị trạng thái "Gửi thất bại".
7.3. Người dùng xác nhận và kết thúc use case.	

Sơ đồ activity:



Hình 3.10: Sơ đồ hoạt động UC005 - Nhắn tin

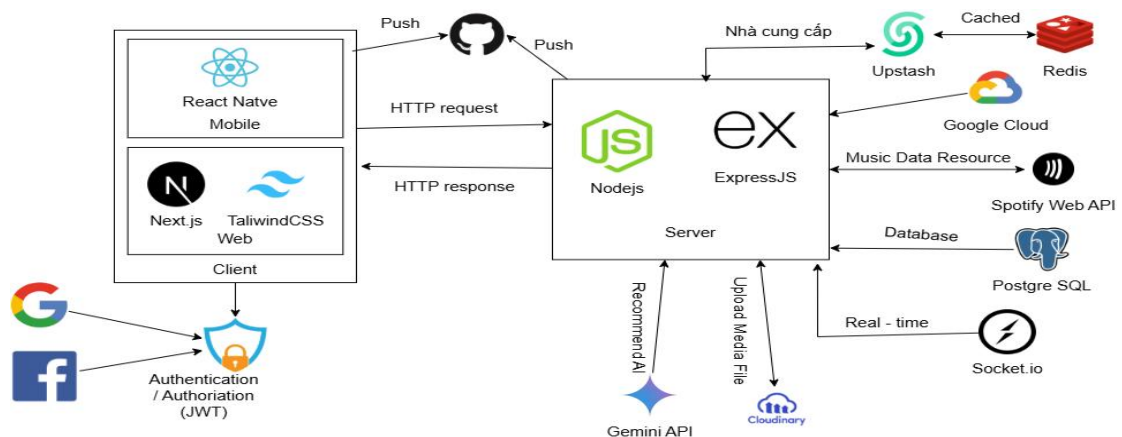
Sơ đồ sequence:



Hình 3.11: Sơ đồ trình tự UC005 - Nhấn tin

3.3.3. Kiến trúc hệ thống:

Hệ thống JT - Harmony sử dụng mô hình Client - Server với HTTP request và response để giao tiếp giữa các thành phần client là giao diện mobile (ReactNative) và giao diện web (Next.js) với server (Node.js/ ExpressJS). Người sử dụng đăng nhập vào hệ thống bằng tài khoản đã đăng ký với email hoặc sử dụng tài khoản Google, Facebook. Nhận JWT ở và lưu trữ token trong cookie (web) và header.Authorization (mobile) để thực hiện các yêu cầu mà không cần đăng nhập lại. Server kết nối đến PostgreSQL để lưu trữ dữ liệu và thực hiện truy vấn (Hình 3.12).

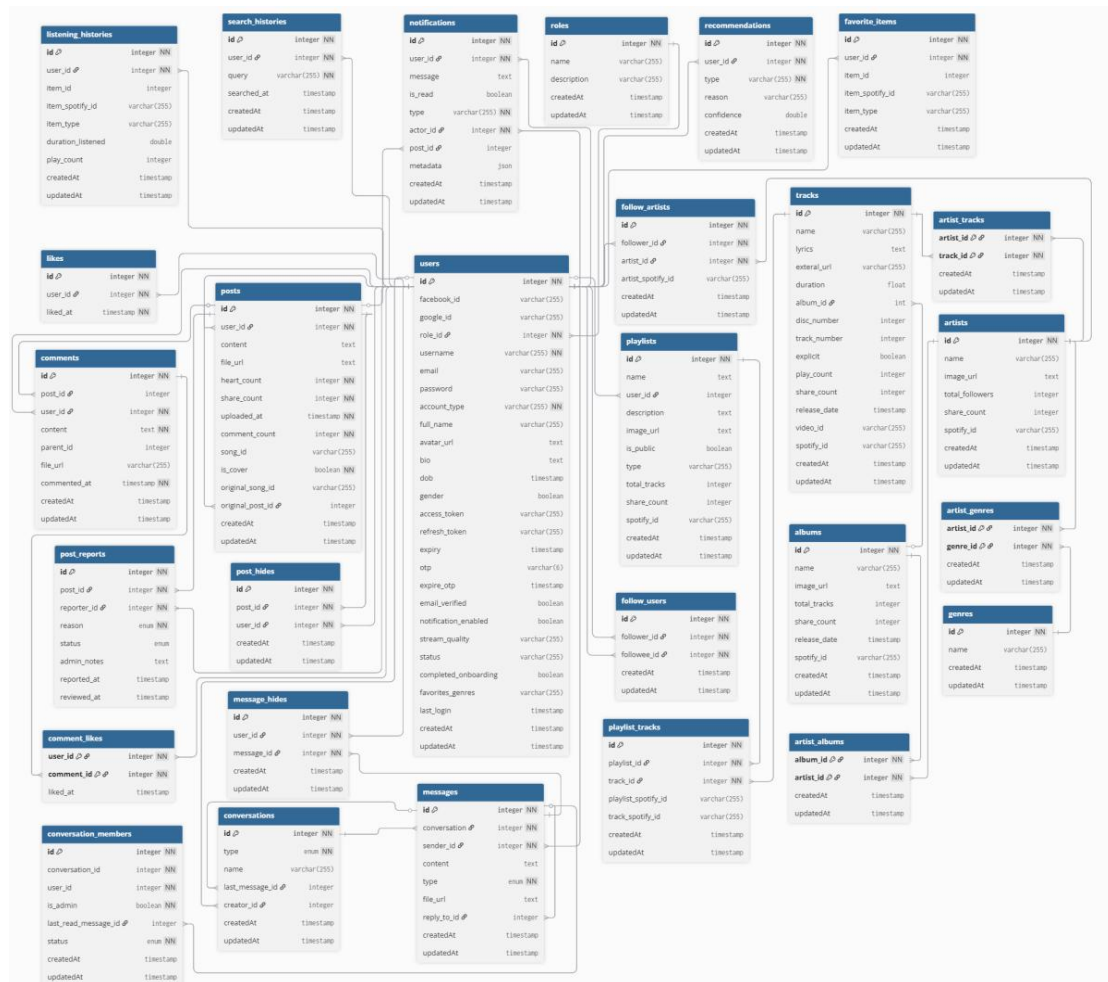


Hình 3.12: Sơ đồ kiến trúc hệ thống JT - Harmony

3.3.4. Sơ đồ cơ sở dữ liệu (Database Diagram)

Hệ thống JT - Harmony sử dụng hệ quản trị CSDL PostgreSQL làm database chính, sơ đồ CSDL quan hệ được thiết kế với 28 bảng, trong đó có 13 bảng chính, 12 bảng phụ và 3 bảng trung gian (Hình 3.13).

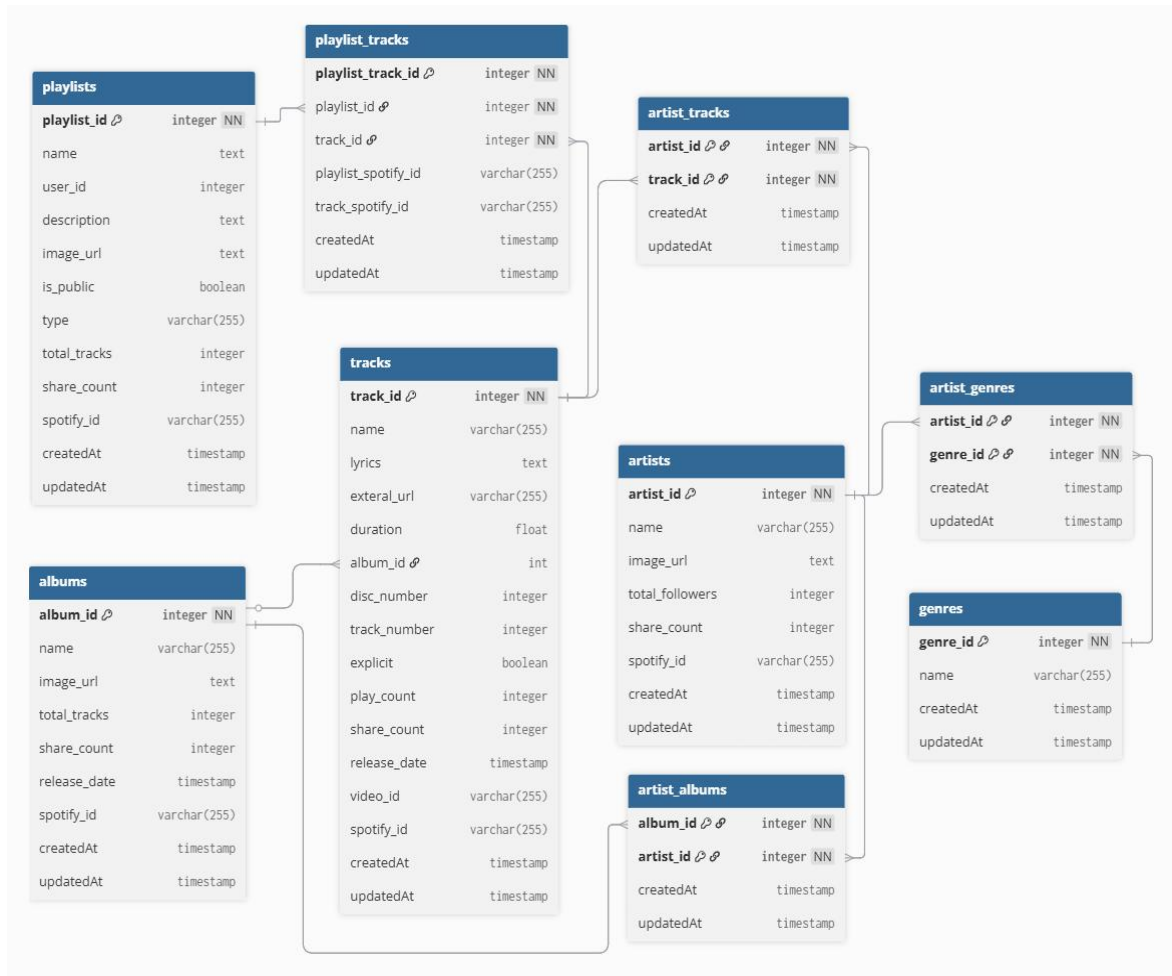
Sơ đồ được chia tách rõ ràng để xây dựng 2 chức năng cốt lõi của hệ thống là sơ đồ các bảng quản lý thông tin nghe nhạc người dùng (3.14) và các bảng quản lý mạng xã hội (Hình 3.15)



Hình 3.13: Sơ đồ CSDL quan hệ (bản đầy đủ)



Hình 3.14: Sơ đồ CSDL (user-schema)



Hình 3.15: Sơ đồ CSDL quan hệ (music schema)



Hình 3.16: Sơ đồ CSDL quan hệ (social schema)

Các quan hệ giữa các bảng được ràng buộc với nhau thông qua các khóa ngoại là id có kiểu dữ liệu là integer. Dưới đây là mô tả một số mối quan hệ cốt lõi trong hệ thống.

Quan hệ một - nhiều (One - to - many)

Role - User: một vai trò có thể có nhiều người dùng, một người dùng chỉ có một vai trò

User - Post: một người dùng có thể có nhiều bài đăng, một bài đăng thuộc sở hữu của một người.

User - Comment: một người dùng có thể có nhiều bình luận, một bình luận thuộc sở hữu của một người.

User - Playlist: một người dùng có thể tạo ra nhiều danh sách phát, một danh sách phát thuộc sở hữu của chỉ một người.

User - Notification: một người dùng có thể có nhiều thông báo, một thông báo chỉ gửi đến một người.

User - Message: một người dùng có thể nhắn nhiều tin nhắn, một tin nhắn chỉ do một người gửi đi.

User - Recommendation: một người dùng sẽ nhận được nhiều đề xuất cá nhân hóa, một đề xuất chỉ dành riêng cho một người.

Post - Comment: một bài đăng có thể có nhiều bình luận, một bình luận chỉ phản hồi về một bài đăng.

Conversation - Message: một cuộc trò chuyện sẽ có nhiều tin nhắn, một tin nhắn chỉ nằm trong một cuộc trò chuyện duy nhất.

Quan hệ nhiều - nhiều (Many - to - many)

User - User: một người dùng có thể theo dõi nhiều người dùng và người dùng có thể được nhiều người dùng khác theo dõi thông qua bảng follow_users.

User - Artist: một người dùng có thể theo dõi nhiều nghệ sĩ và nghệ sĩ thì có nhiều người theo dõi thông qua bảng follow_artists.

User - Track: một người dùng có thể nghe nhiều bài hát và một bài hát có thể được nghe bởi nhiều người dùng thông qua bảng listening_histories. Người dùng cũng có thể thêm nhiều bài hát vào danh sách yêu thích và một bài hát cũng được yêu thích bởi nhiều người thông qua bảng favorite_items.

Track - Artist: một bài hát được biểu diễn bởi một hoặc nhiều nghệ sĩ, một nghệ sĩ có thể biểu diễn nhiều bài hát thông qua bảng artist_tracks.

Artist - Album: một nghệ sĩ có thể sở hữu nhiều album, và một album có thể thuộc sở hữu của nhiều nghệ sĩ thông qua bảng artist_albums.

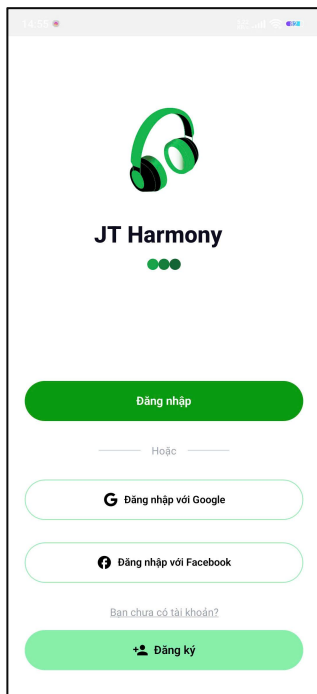
Conversation - User: một cuộc trò chuyện có thể có nhiều người dùng, một người dùng có thể tham gia nhiều cuộc trò chuyện khác nhau.

3.3.5. Một số giao diện chính của hệ thống:

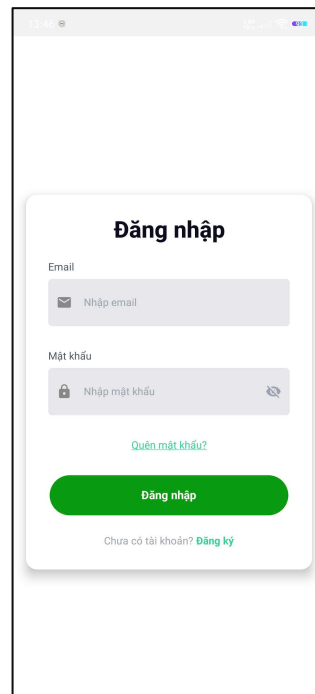
Giao diện hệ thống JT -Harmony được xây dựng trên hai nền tảng là mobile cho người nghe nhạc và website cho quản trị viên. Màn hình chính trên nền tảng mobile hiển thị những danh sách bài hát, danh sách phát, nghệ sĩ và album đề xuất cho người dùng. Điều đặc biệt là những thông tin trên đều được hệ thống tự động đề

xuất dựa trên phân tích hành vi thói quen và sở thích của người nghe nhạc (Hình 3.18).

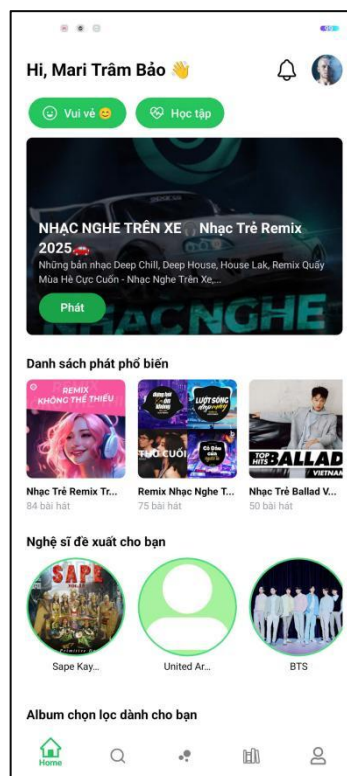
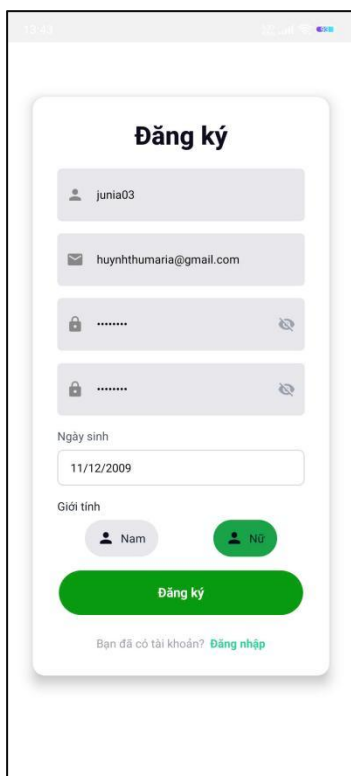
Một số giao diện trên giao diện mobile



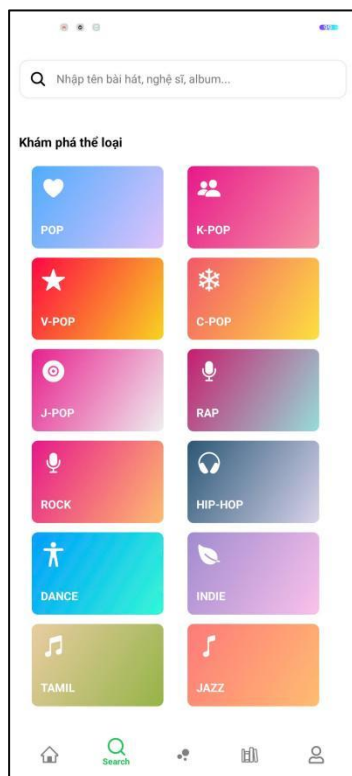
Hình 3.17: Màn hình Auth Landing



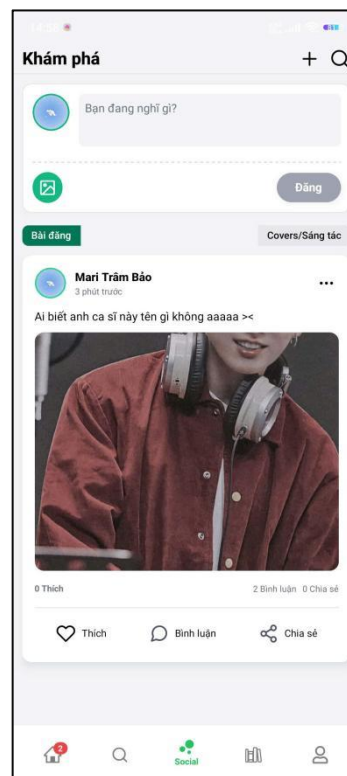
Hình 3.18: Màn hình đăng nhập



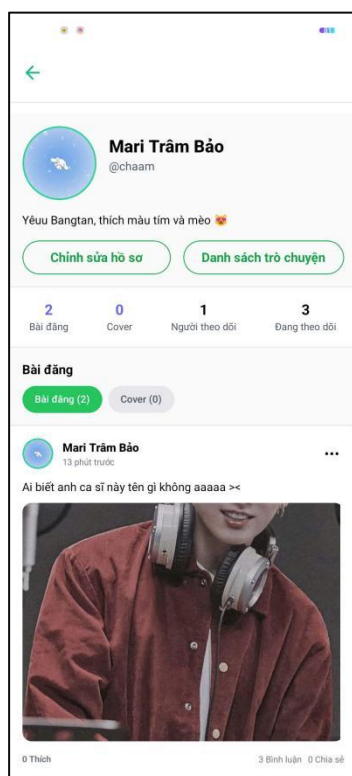
Hình 3.19: Màn hình Đăng ký



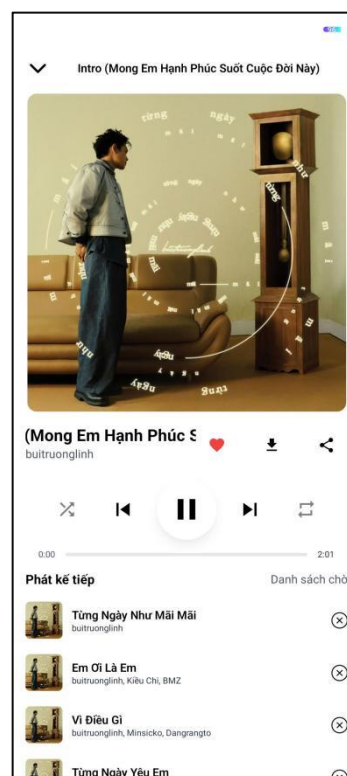
Hình 3.20: Màn hình chính



Hình 3.21: Màn hình Tìm kiếm



Hình 3.22: Màn hình Mạng xã hội

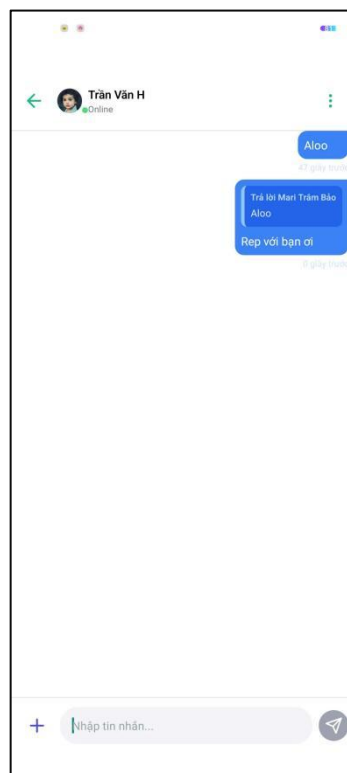


Hình 3.23: Màn hình Profile Social

Hình 3.24: Màn hình Chi tiết bài hát



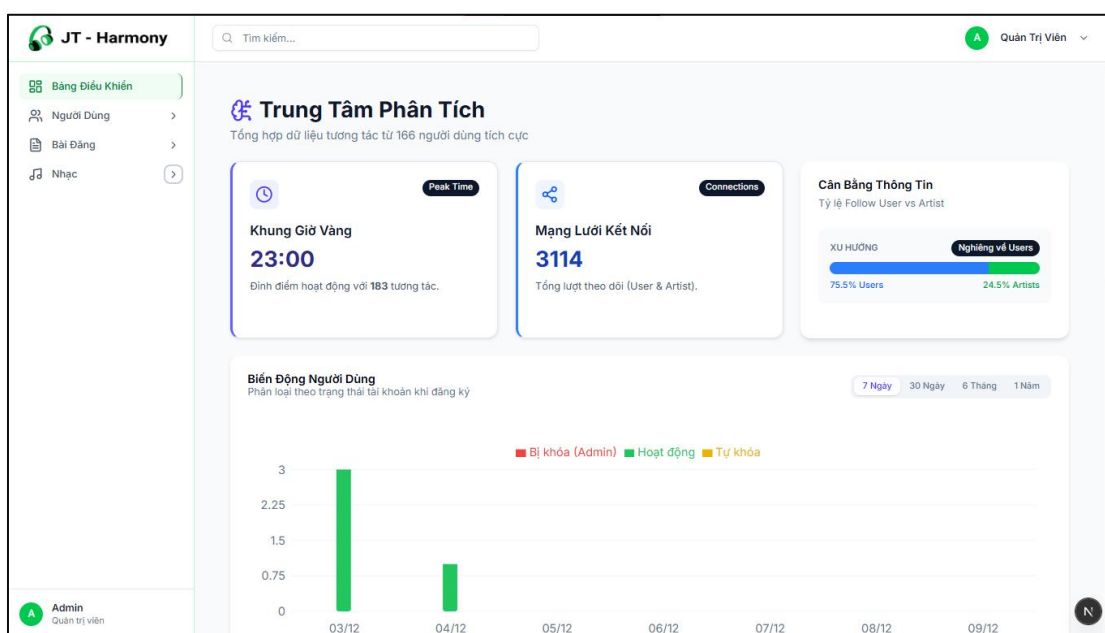
Hình 3.25: Màn hình Bình luận



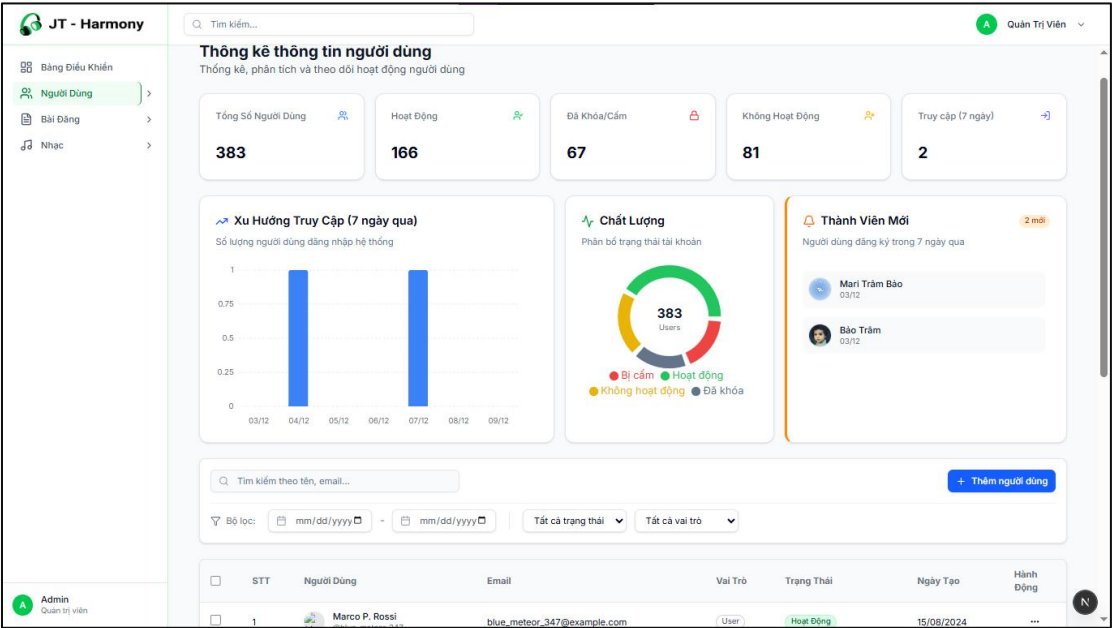
Hình 3.26: Màn hình Nhắn tin

Trang chủ của web quản trị viên được hiển thị đầy đủ thông tin phân tích và thống kê về người dùng, trạng thái và biến diễn số liệu dưới dạng biểu đồ để người quản trị xem được bao quát thông tin trong toàn hệ thống (Hình 3.26).

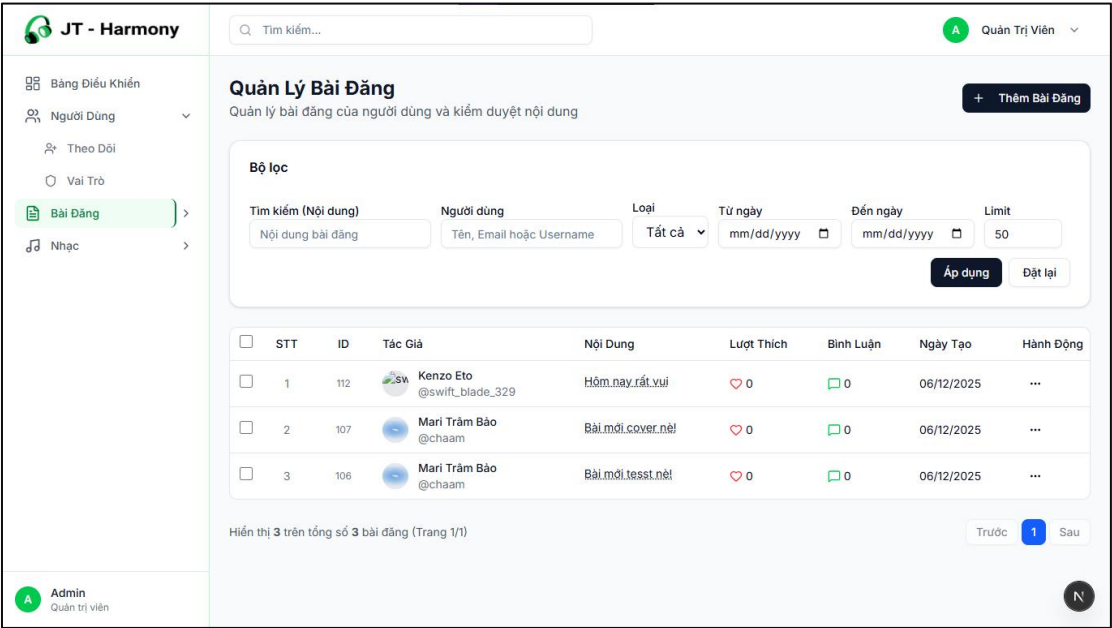
Một số giao diện trên giao diện web



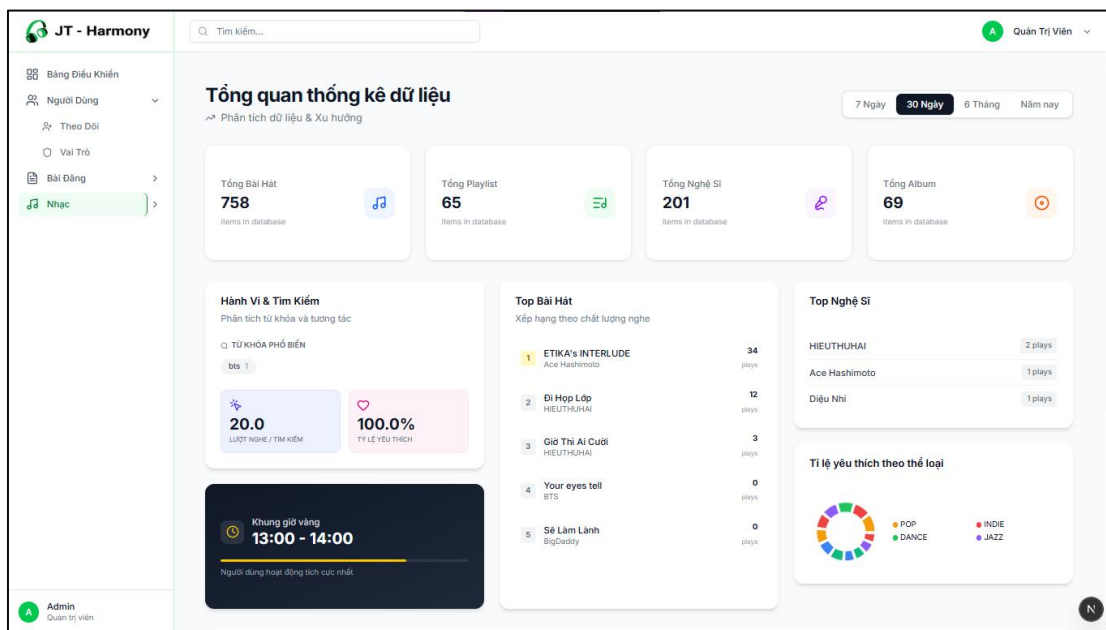
Hình 3.27: Màn hình Trang chủ quản lý



Hình 3.28: Màn hình Quản lý người dùng



Hình 3.29: Màn hình Quản lý bài đăng



Hình 3.30: Màn hình Xem thống kê thông tin âm nhạc

Chương 4. CÀI ĐẶT VÀ TRIỂN KHAI HỆ THỐNG

4.1. Môi trường phát triển và công cụ:

Hệ thống JT - Harmony được phát triển trên hai nền tảng mobile và web sử dụng các môi trường và công cụ phổ biến:

Bảng 4.1: Các thành phần môi trường triển khai và công cụ

Danh mục	Thành phần	Mô tả
Hệ điều hành	Window 10: 22H2	Môi trường phát triển và triển khai hệ thống
Ngôn ngữ lập trình	JavaScript	JavaScript là ngôn ngữ chính được sử dụng cho toàn bộ dự án (Backend, Frontend và Mobile).
Môi trường thực thi	Node: 22.13.0	Môi trường runtime Backend để thực thi mã JavaScript và chạy Express.js Server.
Framework back-end	Express.js: 4.17.1,	Framework Node.js được sử dụng để xây dựng API RESTful cho hệ thống.
Framework front-end	React Native: 0.81.4,	Framework React Full-stack để phát triển giao diện Web (hỗ trợ SSR/SSG và API Routes).
Framework mobile	Next.js: 16.0.1	Framework dùng để phát triển ứng dụng di động trên nền tảng Android.
CSDL	PostgreSQL: 17.2, Redis: ^5.9.0	PostgreSQL là hệ quản trị CSDL quan hệ chính, dùng để lưu trữ

		dữ liệu hệ thống. Redis là kho lưu trữ dữ liệu trong Cache, mục đích lưu trữ kết quả của những truy vấn thường được truy cập.
IDE	Visual Studio Code: 1.106.3	Trình soạn thảo mã nguồn.
Version Controller	Git: 2.47.1.windows.2	Công cụ quản lý mã nguồn.

4.2. Kiến trúc triển khai hệ thống:

Hệ thống nghe nhạc JT - Harmony được thiết kế với kiến trúc tách biệt rõ ràng giữa Client (front-end) và Server (back-end). Sử dụng các công nghệ hiện đại để đảm bảo hiệu suất, bảo mật và tăng khả năng mở rộng.

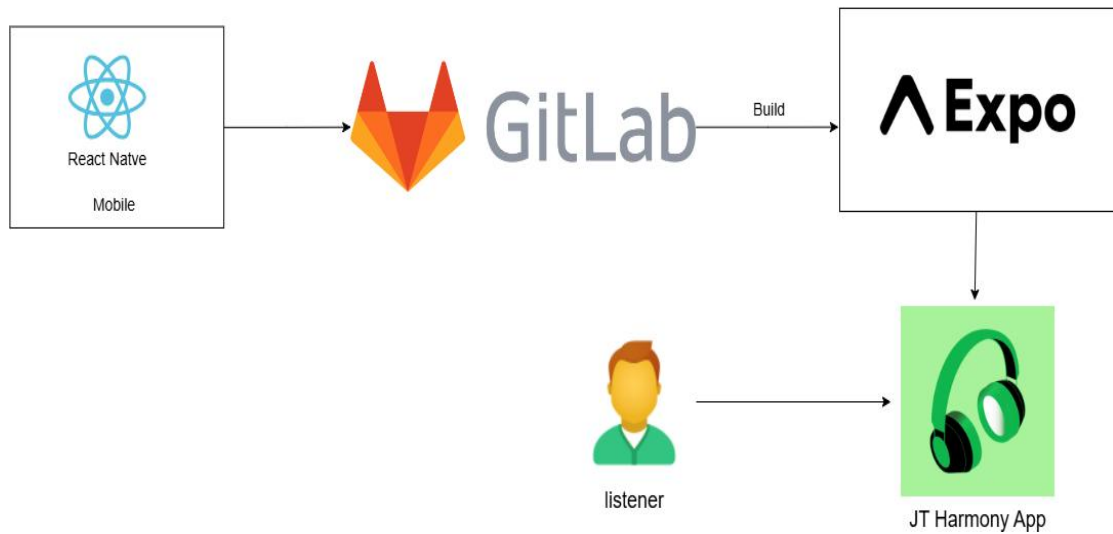
Client

Giao diện người dùng được xây dựng bằng ReactNative và NativeWind trên mobile, Next.js và TailwindCSS, giúp tạo ra một nền tảng ứng dụng web linh hoạt. JWT được sử dụng để tăng cường bảo mật và xác thực người dùng.

Server

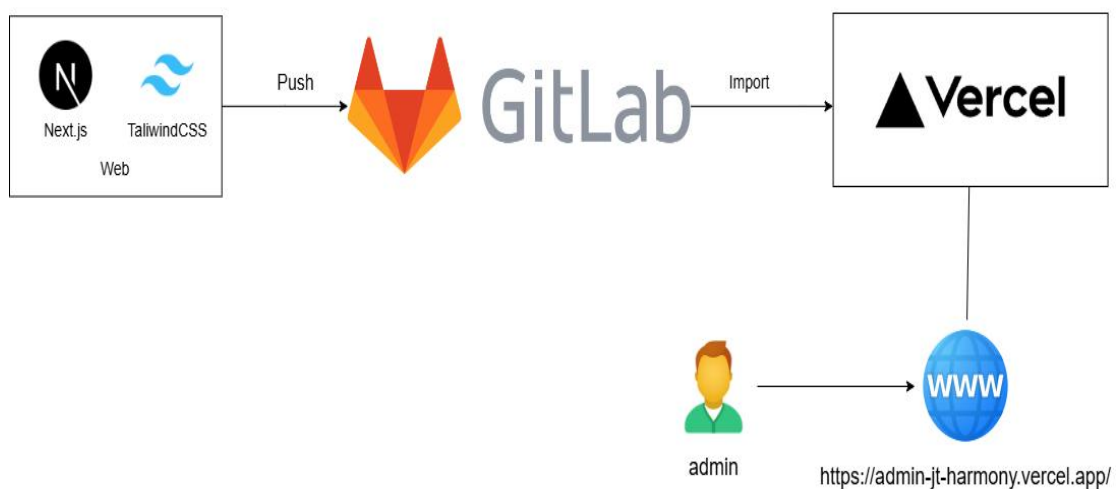
NodeJS và ExpressJS xây dựng các RESTful API, xử lý các yêu cầu và logic nghiệp vụ. Sử dụng PostgreSQL lưu trữ dữ liệu, phục vụ các yêu cầu lưu trữ thông tin âm nhạc như bài hát, thông tin nghệ sĩ, album và các thông tin người dùng. Socket.IO hỗ trợ tương tác thời gian thực giữa những người dùng với nhau.

Control Version

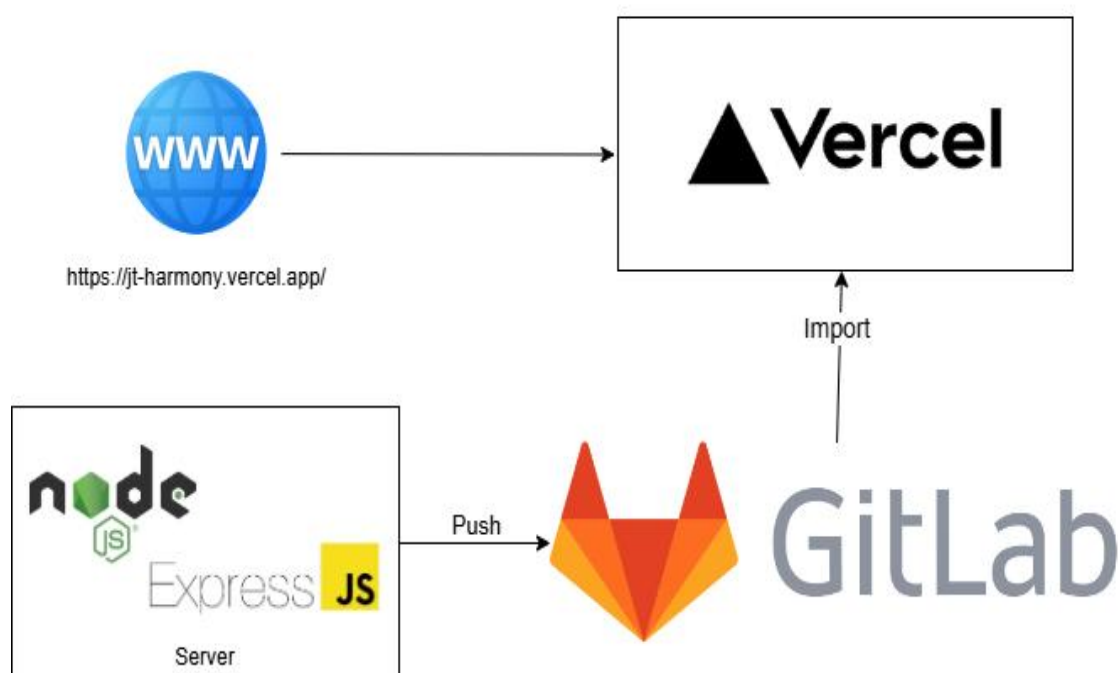


Hình 4.1: Mô hình triển khai front-end mobile

Control Version



Hình 4.2: Mô hình triển khai front-end web



Hình 4.3: Mô hình triển khai back-end

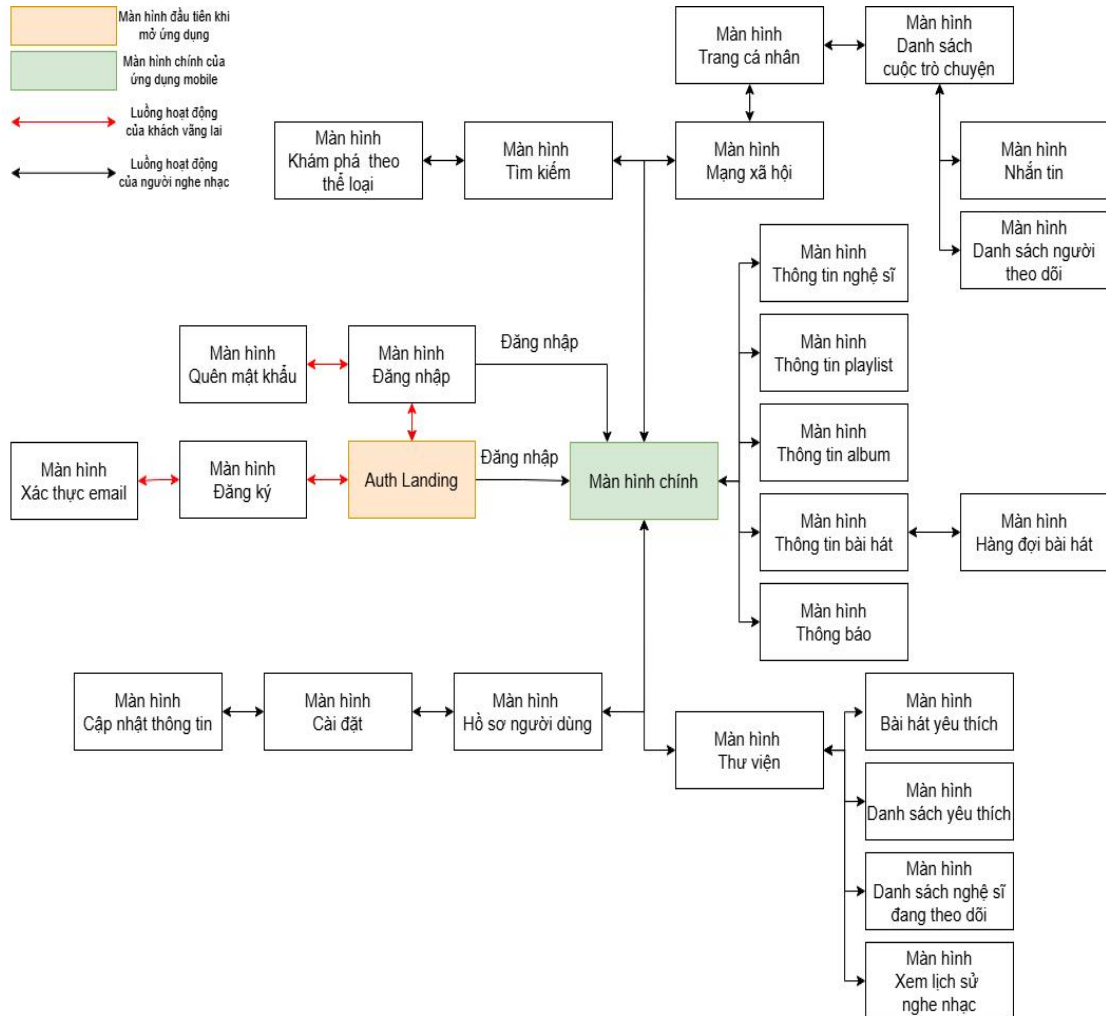
4.3. Các module chức năng chính:

Hệ thống đã triển khai hoàn thiện các phân hệ chức năng sau:

- *Module Xác thực & Người dùng*: Cung cấp API đăng ký, đăng nhập (bao gồm OAuth Google/Microsoft), cấp phát JWT token, và quản lý thông tin hồ sơ người dùng. Middleware xác thực đảm bảo an toàn cho các API quan trọng.
- *Module Nhạc & Playlist*: Quản lý toàn bộ vòng đời của bài hát, album, nghệ sĩ và playlist. Hỗ trợ các API cho việc tìm kiếm, thêm vào yêu thích, và ghi nhận lịch sử nghe nhạc.
- *Module Tương tác xã hội*: Bao gồm chức năng đăng bài viết (Posts), bình luận và Chat realtime. Hệ thống sử dụng Socket.IO để đẩy thông báo ngay lập tức khi có tương tác mới.
- *Module Upload & Media*: Tích hợp Multer và Cloudinary để xử lý việc tải lên và lưu trữ an toàn các tệp hình ảnh (avatar, cover) và âm thanh từ người dùng.

4.4. Giao diện và luồng hoạt động:

Ứng dụng JT - Harmony được xây dựng dựa trên sự thiết lập mối liên kết chặt chẽ giữa các màn hình, tạo trải nghiệm liền mạch và dễ dàng sử dụng cho người dùng. Luồng hoạt động của các màn hình chính trên ứng dụng JT - Harmony được trình bày cụ thể qua



Hình 4.4: Sơ đồ luồng hoạt động màn hình trên mobile

Người dùng khi chưa đăng nhập vào ứng dụng sẽ được gọi chung là khách vãng lai. Nhóm người dùng này chỉ có thể truy cập vào bộ các màn hình Auth Screen. Người nghe nhạc đã đăng nhập vào ứng dụng được phép thao tác trên toàn bộ màn hình mà không bị giới hạn quyền truy cập.

Ứng dụng được xây dựng sử dụng Expo Router làm cơ chế định tuyến dựa trên cấu trúc tập tin. Điều này cho phép ứng dụng được tổ chức và phân tách rõ ràng

hai luồng điều hướng riêng biệt: nhóm màn hình xác thực dành cho khách vãng lai và nhóm màn hình chính dành cho người nghe nhạc đã đăng nhập, đảm bảo tính bảo mật và logic luồng hoạt động.

4.5. Tích hợp:

Hệ thống đã tích hợp thành công các dịch vụ bên thứ ba: **Cloudinary** cho lưu trữ media, **Google/Facebook OAuth** cho xác thực nhanh, và **Nodemailer** cho dịch vụ email thông báo.

Chương 5. KIỂM THỬ VÀ ĐÁNH GIÁ HỆ THỐNG

5.1. Môi trường kiểm thử:

Yêu cầu: Máy tính cá nhân có kết nối mạng LAN

5.1.1. Phần cứng:

Bảng 5.1: Bảng thông số cấu hình máy tính cá nhân

CPU	RAM	HDD/SSD
11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz 2.42 GHz	16 GB	477 GB

5.1.2. Phần mềm:

Bảng 5.2: Bảng thông tin phần mềm

Tên phần mềm	Phiên bản	Loại
Postman	11.74.5	Công cụ test API
Visual Studio Code	1.106.3	IDE cho ReactJS

5.2. Chiến lược kiểm thử:

Để đảm bảo hệ thống nghe nhạc JT-Harmony hoạt động ổn định, chính xác và đáp ứng nhu cầu người sử dụng, nhóm đã áp dụng chiến lược kiểm thử từ mức độ thấp nhất là code đến mức độ cao nhất là người dùng. Các công cụ hỗ trợ bao gồm Visual Studio Code và Postman.

Kiểm thử đơn vị (Unit Test)

- *Mục tiêu:* Kiểm tra tính đúng đắn của các hàm xử lý logic riêng lẻ, đảm bảo các module con hoạt động đúng thiết kế trước khi ghép nối.
- *Phạm vi:* Tập trung vào các hàm tiện ích, các hàm xử lý logic ở phía server.
- *Công cụ:* Sử dụng trình debug của Visual Studio Code.

Kiểm thử tích hợp (Integration Test)

- *Mục tiêu:* Kiểm tra phản hồi HTTP và dữ liệu trả về có đúng với tài liệu thiết kế hay không.
- *Phạm vi:* Gửi request HTTP đến server và tập trung vào các phản hồi của API.
- *Công cụ:* Sử dụng công cụ kiểm thử API Postman.

Kiểm thử quy trình (Process Test)

- *Mục tiêu:* Đảm bảo luồng hoạt động của hệ thống suôn sẻ khi người dùng thao tác qua nhiều chức năng và màn hình.
- *Quy trình mẫu:*
 - + Đăng ký: Người dùng chưa có tài khoản -> Thực hiện đăng ký -> Hệ thống gửi email kèm mã otp đến cho người dùng -> Người dùng xác thực email thành công -> Hệ thống quay lại trang đăng nhập -> Người dùng thực hiện Đăng nhập bằng tài khoản vừa tạo -> Đăng nhập thành công và chuyển vào trang chủ.
 - + Đăng bài: Người dùng đăng nhập -> Mở trang mạng xã hội -> Chọn đăng bài -> Hệ thống thông báo đăng bài thành công -> Bài đăng hiển thị trên trang chính của mạng xã hội.

Kiểm thử tính hiệu quả/chính xác

- *Mục tiêu:* Đảm bảo hệ thống phản hồi nhanh và dữ liệu được lưu trữ chính xác.
- *Phương pháp:* Quan sát thời gian phản hồi (Response time) trên Postman và kiểm tra dữ liệu thực tế được ghi vào Database sau khi thao tác.

5.3. Danh sách các tình huống kiểm thử:

Bảng 5.3: Bảng báo cáo kết quả kiểm thử

BÁO CÁO KẾT QUẢ KIỂM THỬ								
Họ tên người thực hiện:				Passed:			13	
Phan Thị Huỳnh Thư				Fail:			0	
Lớp: DHKTPM17A				Tổng số test case:			13	
Mã số sinh viên: 21019021								
STT	Ngày thực hiện	Test Case ID	Chức năng	Mô tả test case	Quy trình kiểm tra	Kết quả mong muốn	Kết quả thực tế	Status (pass/fail)
1	29/11/2025	TC001	Đăng nhập	Không nhập email	Email: "" Mật khẩu: "12345678"	Yêu cầu nhập đầy đủ thông tin	Yêu cầu nhập đầy đủ thông tin	pass
2	29/11/2025	TC002	Đăng nhập	Nhập email sai định dạng	Email: "abca@" Mật khẩu: "12345678"	Thông báo email sai định dạng	Thông báo email sai định dạng	pass
3	29/11/2025	TC003	Đăng nhập	Nhập sai mật khẩu	Email: "abc@gmail.com" Mật khẩu: "12345679"	Thông báo sai email hoặc mật khẩu không đúng	Thông báo sai email hoặc mật khẩu không đúng	pass
4	29/11/2025	TC004	Đăng nhập	Nhập đúng	Email: "abc@gmail"	Đăng nhập thành công	Đăng nhập thành công	pass

				email và mật khẩu	.com” Mật khẩu: “12345678”	và chuyển vào màn hình chính	và chuyển vào màn hình chính	
5	29/11/2025	TC005	Đăng ký	Không nhập username	Username: “” Email: “abc@gmail.com” Mật khẩu: “12345678” Nhập lại mật khẩu: “12345678”	Thông báo username là bắt buộc	Thông báo username là bắt buộc	pass
6	29/11/2025	TC006	Đăng ký	Không nhập email	Username: “user003” Email: “” Mật khẩu: “12345678” Nhập lại mật khẩu: “12345678”	Thông báo email là bắt buộc	Thông báo email là bắt buộc	pass
7	29/11/2025	TC007	Đăng ký	Chiều dài mật khẩu < 8 ký tự hoặc > 12 ký tự	Username: “user003” Email: “abc@gmail.com” Mật khẩu: “1234567” Nhập lại	Thông báo mật khẩu từ 8 - 12 ký tự	Thông báo mật khẩu từ 8 - 12 ký tự	pass

					mật khẩu: “1234567”			
8	29/11 /2025	TC008	Đăng ký	Nhập lại mật khẩu không trùng khớp	Username: “user003” Email: “abc@gmail .com” Mật khẩu: “12345678” Nhập lại mật khẩu: “123456789 ”	Thông báo mật khẩu nhập lại không khớp	Thông báo mật khẩu nhập lại không khớp	pass
9	29/11 /2025	TC009	Đăng ký	Username đã tồn tại trong CSDL	Username: “user001” Email: “abc@gmail .com” Mật khẩu: “12345678” Nhập lại mật khẩu: “12345678”	Thông báo username đã tồn tại	Thông báo username đã tồn tại	pass
10	29/11 /2025	TC010	Đăng ký	Email đã tồn tại trong CSDL	Username: “user003” Email: “abc@gmail .com” Mật khẩu:	Thông báo email đã tồn tại	Thông báo email đã tồn tại	pass

					“12345678” Nhập lại mật khẩu: “12345678”			
11	29/11 /2025	TC011	Đăng ký	Tạo tài khoản thành công	Username: “user003” Email: “abc@gmail .com” Mật khẩu: “12345678” Nhập lại mật khẩu: “12345678”	Gửi OTP xác thực đến email và chuyển vào trang xác thực email.	Gửi OTP xác thực đến email và chuyển vào trang xác thực email.	pass
12	29/11 /2025	TC012	Tạo danh sách phát	Không nhập tên	Name: “” Description: “”	Thông báo tên là bắt buộc	Thông báo tên là bắt buộc	pass
13	29/11 /2025	TC013	Tạo danh sách phát	Tạo danh sách phát thành công	Name: “Yêu đời” Description: “”	Thông báo tạo danh sách phát thành công và hiển thị	Thông báo tạo danh sách phát thành công và hiển thị	pass

Chương 6. KẾ HOẠCH KHỞI NGHIỆP

6.1. Giới thiệu về dự án:

Tên dự án: Xây dựng ứng dụng nghe nhạc JT - Harmony mobile tích hợp gợi ý và chia sẻ nhạc.

Mục tiêu dự án

Ngắn hạn: 6 tháng đầu

- Phát triển một ứng dụng mobile nghe nhạc hiện đại, tích hợp gợi ý cá nhân hóa và chức năng chia sẻ trên mạng xã hội.
- Triển khai AI recommendation đạt độ chính xác gợi ý $\geq 70\%$ dựa trên lịch sử nghe và sở thích.
- Cung cấp trải nghiệm nghe nhạc toàn diện (tìm kiếm, phát, tạo playlist, offline) với giao diện UX/UI tối ưu.
- Tạo điều kiện tiếp cận sinh viên các Trường Đại học trải nghiệm trực tiếp, đặc biệt là sinh viên Trường Đại học Công nghiệp Thành phố Hồ Chí Minh.
- Đạt 5000 lượt tải xuống và 2000 người dùng hoạt động hàng tháng tại các thành phố khu vực miền Nam.

Dài hạn: 3 - 5 năm

- Nâng cấp hệ thống, tăng cường bảo mật, mở rộng lưu trữ và bổ sung thêm chức năng hệ thống.
- Mở rộng phạm vi thị trường ra cả nước, đặt nền móng cho tiếp cận thị trường Đông Nam á trong tương lai.
- Triển khai mô hình Freemium, cung cấp miễn phí các dịch vụ, tính năng cơ bản và trả phí cho các tính năng nâng cao.
- Đạt 100.000 lượt tải xuống và 50.000 người dùng hoạt động hàng tháng trong phạm vi cả nước.

Các bên tham gia

- *Người lãnh đạo dự án:* Phan Thị Huỳnh Thu

- *Đội ngũ phát triển dự án*: bao gồm các thành viên nhóm dự án và người quản lý dự án.

- *Người dùng cuối*: người nghe nhạc.

6.2. Phạm vi:

Phạm vi chức năng

- *Người nghe nhạc*: Đăng nhập/Đăng ký (Social Auth), nghe nhạc, tìm kiếm, tạo danh sách phát, quản lý thông tin cá nhân. Bên cạnh đó là xây dựng các chức năng cho mạng xã hội nội bộ cho phép người sử dụng upload cover/video, bình luận/tương tác thời gian thực (SocketIO), chia sẻ qua MXH (link).

- *Quản trị viên*: Quản lý nội dung/thống kê qua dashboard (Node.js/Express API).

- *Tính năng nâng cao của hệ thống*: Tích hợp Gemini API để tạo ra đề xuất thông minh nhằm mang lại trải nghiệm cá nhân hóa cho người dùng.

Công nghệ

- *Backend*: Node.js/Express cho API, SocketIO cho realtime, JWT, Social login (google sign-in, facebook sign-in)

- *Frontend*: React Native cho mobile, Next.js cho web quản trị.

- *Database*: PostgreSQL cho dữ liệu chính (users, songs, playlists), Redis cho lưu trữ bộ nhớ đệm các request thường gặp.

- *AI*: tích hợp Gemini API để hiện thực recommendation cho người dùng.

Kiến trúc: Client-Server, với React Native client kết nối API và SocketIO.

Giới hạn

- Không hỗ trợ nghe nhạc offline.

- Không tích hợp thanh toán hoặc subscription trong giai đoạn đầu.

- Chưa có cơ chế bảo vệ bản quyền cho sáng tác cá nhân của người dùng.

6.3. Lịch trình:

Kế hoạch khởi nghiệp dự kiến hoàn thành trong 6 tháng.

Bảng 6.1: Bảng tóm tắt kế hoạch dự án

Giai đoạn	Thời gian	Mục tiêu chính	Đầu ra
Nghiên cứu và Phân tích	1 - 2 tuần	- Khảo sát nhu cầu người nghe nhạc và các nền tảng tương tự (Spotify, Zing MP3, Apple Music).	- Báo cáo nhu cầu thị trường, phân tích và xác định điểm mạnh/yếu của đối thủ - Tìm được nhà cung cấp dữ liệu hỗ trợ xây dựng chức năng nghe nhạc.
		Tìm kiếm nguồn dữ liệu nhạc uy tín, bản quyền.	
Thiết kế kiến trúc và ứng dụng	1 tháng	Xây dựng giao diện UI/UX bằng Figma.	- Bộ giao diện hoàn chỉnh. - Công nghệ Next.js cho web admin, React Native cho mobile, Node.js cho backend, PostgreSQL để lưu trữ database chính và Redis để lưu trữ bộ nhớ đệm, Socket.IO để tương tác thời gian thực, ... - Sơ đồ kiến trúc Client Server.
		Xác định sử dụng kiến trúc Client - Server và lựa chọn các công nghệ.	
Phát triển hệ thống	2 - 3 tháng	Xây dựng Server.	- Hệ thống với mô hình Server - Client hoàn chỉnh.
		Phát triển ứng dụng mobile (Frontend) và Website Admin.	

		Gắn API kết nối frontend và backend.	- Ứng dụng có tính năng nghe nhạc, mạng xã hội chia sẻ và tính năng nâng cao Đề xuất dựa trên hành vi, sở thích, lịch sử của người dùng.
		Huấn luyện và Tích hợp thuật toán AI đưa ra các đề xuất cá nhân hóa trải nghiệm.	
Kiểm thử và tối ưu hóa	1 - 2 tuần	Kiểm thử tích hợp hệ thống	Báo cáo kiểm thử: sửa lỗi, tối ưu hiệu năng và bảo mật.
Triển khai và ra mắt sản phẩm	1 tháng	Ra mắt bản Beta chính thức trên Google Play.	Website và ứng dụng hoạt động ổn định. Thu thập phản hồi từ 50 người dùng tích cực đầu tiên để ưu tiên tính năng cho phiên bản chính thức.
		Thực hiện tiếp thị tại các Trường Đại học nhằm tiếp cận nhóm người dùng trẻ tuổi.	
		Thu thập phản hồi từ người sử dụng và cải tiến sản phẩm.	
		Thực hiện tiếp thị qua Google Ads, Youtube, Facebook/Tiktok Ads nhằm tiếp cận người dùng nhanh chóng.	

6.4. Ngân sách:

Bảng 6.2: Bảng dự kiến ngân sách dự án

Danh mục	Ngân sách (VND)	Chi tiết sử dụng
Nhân sự (15 người)	3.600.000.000	Lương cho: 4 Dev Mobile (React Native) 2 Dev Web (ReactJS) 3 Dev Backend (Node.js/Microservice)

		1 Chuyên gia AI/Data 1 Designer 1 Marketing 1 CTO 1 Project Manager 1 Tester
Cơ sở hạ tầng (máy chủ, công cụ phát triển)	80.000.000	Thuê máy chủ (google cloud) và database. Mua công cụ phát triển (figma, github premium, CI/CD tools).
Chi phí bản quyền nội dung	50.000.000	Chi phí chi trả cho việc mua bản quyền data từ các nền tảng cung cấp nhạc.
Tiếp thị và quảng bá	27.000.000	Chi phí quảng cáo google, youtube, facebook/tiktok ads trong giai đoạn ra mắt.
Dự phòng	30.000.000	Dự phòng cho lỗi kỹ thuật phức tạp hoặc chi phí huấn luyện ai phát sinh.
Tổng ngân sách dự kiến	3.787.000.000	

6.5. Quản lý nhân lực

Bảng 6.3: Bảng phân công nhiệm vụ (RACI)

Hoạt động	Chịu trách nhiệm (R)	Thực hiện (A)	Tham vấn (C)	Thông báo (I)
Giai đoạn 1: Nghiên cứu và Phân tích				
Khảo sát thị trường, đối thủ, nhu cầu người sử	Marketing	Marketing	PM, Chuyên gia AI	CTO

dụng.				
Tìm kiếm nguồn dữ liệu nhạc uy tín, bản quyền.	Chuyên gia AI/Data	Chuyên gia AI/Data	Marketing, 2 Dev Backend	CTO, PM
Giai đoạn 2: Thiết kế kiến trúc và ứng dụng				
Xây dựng giao diện UI/UX bằng Figma.	Designer	Designer	PM, Marketing	CTO
Xác định sử dụng kiến trúc Client - Server và lựa chọn các công nghệ.	CTO	3 Dev Backend	PM, Tester	CTO
Giai đoạn 3: Phát triển hệ thống				
Xây dựng Server	CTO	3 Dev Backend	Chuyên gia AI, Tester	PM
Lập trình frontend	PM	4 Dev Mobile, 2 Dev Web	Designer, Tester	CTO
Huấn luyện và tích hợp ai	Chuyên gia AI/Data	Chuyên gia AI, 3 Dev Backend	CTO, PM	Chuyên gia AI
Giai đoạn 4: Kiểm thử và tối ưu hóa				
Kiểm thử hệ thống	Tester	Tester	PM, CTO	3 Dev Backend, 4 Dev Mobile, 2 Dev Web
Giai đoạn 5: Triển khai và ra mắt sản phẩm				

Xây dựng chiến dịch quảng bá (youtube, ads), tiếp cận sinh viên trong khu vực.	Marketing	Marketing	PM, Designer	CTO
Ra mắt bản beta & thu thập phản hồi	PM	Marketing, Tester	CTO	Toàn bộ Dev Team

Chú thích phân vai trò chính

- *CTO*: Chịu trách nhiệm chính về kiến trúc và hiệu suất của Backend Microservices.
- *PM*: Chịu trách nhiệm điều phối toàn bộ dự án và đặc biệt là kết quả của Frontend (Dev Mobile/Web).
- *Chuyên gia AI*: Chịu trách nhiệm chính về chất lượng của dữ liệu (Kho công thức) và thuật toán AI.
- *Marketing*: Chịu trách nhiệm về nghiên cứu thị trường và các hoạt động tiếp thị/thu thập phản hồi.
- *Tester*: Chịu trách nhiệm thực hiện các loại kiểm thử, nhưng cần tham vấn chuyên gia AI cho kiểm thử lỗi AI.

6.6. Quản lý rủi ro:

Bảng 6.4: Bảng quản lý rủi ro (Risk Management Table)

Rủi ro	Nguyên nhân	Giải pháp
Chậm tiến độ	Thiếu kinh nghiệm, debug lâu.	Chia nhỏ task, đặt deadline tuần.
Hiệu suất database kém	Dữ liệu lớn, chưa tối ưu index.	Thêm index từ đầu, dùng EXPLAIN để tối ưu query, test với 10.000 bản ghi.
Dữ liệu phân tích gợi ý không chính	Dữ liệu huấn luyện thiếu hoặc thuật toán sai.	Điều chỉnh prompt hoặc thay đổi model gen AI.

xác		
Bug trong app	Tích hợp API lỗi.	Test trên máy ảo (Android), dùng log để debug.
Bug trong AI model	Dữ liệu train thiếu.	Sử dụng dữ liệu mẫu, validate model với cross-validation; ưu tiên content-based filtering đơn giản trước.
Hiệu suất realtime kém (SocketIO/notifications)	Kết nối không ổn định trên mobile.	Sử dụng SocketIO với fallback (polling), test trên thiết bị thực tế; fallback Firebase Realtime DB nếu cần.

6.7. Kế hoạch triển khai:

6.7.1. Giai đoạn thử nghiệm:

Thời gian: Tháng 12/2025.

Mục tiêu: Thu thập phản hồi và đánh giá hiệu quả hoạt động của chức năng Gợi ý cá nhân hóa.

Hoạt động

- *Tiến hành kiểm thử toàn hệ thống lần cuối trước khi ra mắt:* Đánh giá hiệu suất của hệ thống khi kiểm thử trên môi trường thực tế.

- *Tổ chức sự kiện ra mắt sản phẩm Beta:* Hợp tác với Trường Đại học Công Nghiệp Thành phố Hồ Chí Minh để tổ chức sự kiện giới thiệu sản phẩm đến sinh viên và các doanh nghiệp tham gia thử nghiệm. Đăng tải trên các nền tảng truyền thông để tiếp cận người dùng.

- *Thu thập phản hồi từ người sử dụng*: Gửi bản khảo sát trực tuyến trên nền tảng hệ thống để thu thập đánh giá về nền tảng âm nhạc, đặc biệt về trải nghiệm cá nhân hóa của người dùng. Lập nhóm cho giai đoạn cải tiến, mở rộng.

6.7.2. Giai đoạn cải tiến, mở rộng:

Thời gian: tháng 1 - 3/2026

Mục tiêu: Nâng cao trải nghiệm của người sử dụng dựa trên phản hồi thực tế đã thu thập, tiến hành nâng cấp, cải tiến và mở rộng thêm các tính năng khắc phục từ bản lỗi Beta, cải thiện tính năng Gợi ý cá nhân hóa.

Hoạt động

- *Phân tích phản hồi đã thu thập*: Tổng hợp ý kiến từ khảo sát và các nền tảng truyền thông. Phân loại phản hồi thành các nhóm: hiệu suất, chức năng, giao diện hoặc trải nghiệm.

- *Lên kế hoạch cải tiến*: Nhóm giai đoạn thực hiện họp nhóm, lập kế hoạch nâng cấp hệ thống.

- *Nâng cấp và kiểm thử*: Thêm các tính năng bổ sung và thực hiện kiểm thử với nhóm người dùng nội bộ trước khi phát hành phiên bản chính thức.

6.7.3. Giai đoạn tiếp thị:

Thời gian: tháng 5 - 7/2026

Mục tiêu: Tiếp cận được lượng lớn người sử dụng, tăng độ nhận diện thương hiệu và đạt được sự công nhận của thị trường, các chuyên gia. Chuyển đổi sang mô hình Freemium.

Hoạt động

- *Thực hiện tiếp thị số*: chạy Google Ads, Facebook Ads, Tiktok Ads nhắm tới giới trẻ tại các thành phố lớn.

- *Hợp tác chiến lược*: Hợp tác với các tổ chức giáo dục như Trường Đại học, Trường Cao Đẳng; Các tổ chức thương mại như các trung tâm thương mại, cửa hàng,...

Chương 7. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

7.1. Kết quả đạt được:

Sau quá trình nghiên cứu và phát triển, đề tài "Xây dựng ứng dụng nghe nhạc trên mobile tích hợp gợi ý và chia sẻ nhạc" đã hoàn thành và đạt được các kết quả quan trọng sau:

Về sản phẩm

- Xây dựng thành công hệ thống JT-Harmony hoàn chỉnh bao gồm: Ứng dụng di động (Mobile App) cho người nghe nhạc, trang Web quản trị (Admin Dashboard) và hệ thống Máy chủ (Backend Server).

- Ứng dụng hoạt động ổn định trên nền tảng Android với giao diện hiện đại, trải nghiệm người dùng mượt mà.

Về chức năng

- *Nghe nhạc*: Đáp ứng tốt các nhu cầu tìm kiếm, phát nhạc trực tuyến, tạo playlist cá nhân và nghe nhạc ngoại tuyến.

- *Xã hội*: Hiện thực hóa thành công các tính năng tương tác thời gian thực (Real-time) như Chat, Bình luận, Đăng cover và Thông báo tức thì nhờ công nghệ Socket.IO.

- *Gợi ý thông minh*: Tích hợp thành công mô hình gợi ý lai (Hybrid Recommendation) sử dụng Gemini AI, giúp cá nhân hóa trải nghiệm nghe nhạc cho người dùng.

Về kỹ thuật

Nhóm đã làm chủ được các công nghệ hiện đại: React Native (Expo) cho mobile, Next.js cho web, Node.js/Express cho backend và PostgreSQL cho cơ sở dữ liệu.

Hệ thống đạt hiệu năng tốt với thời gian phản hồi API trung bình dưới 200ms và khả năng chịu tải ổn định.

7.2. Đánh giá mức độ hoàn thành mục tiêu:

So sánh với các mục tiêu đề ra ban đầu tại Chương 1, nhóm tự đánh giá mức độ hoàn thành như sau:

Bảng 7.1: Bảng đánh giá mức độ hoàn thành mục tiêu

Mục tiêu ban đầu	Mức độ hoàn thành	Ghi chú
Xây dựng hệ thống nghe nhạc đa nền tảng	100%	Hoàn thiện Mobile App chính trên Android cho client và Web Admin.
Tích hợp tính năng Mạng xã hội (Upload, Chat)	100%	Đã triển khai đầy đủ và hoạt động realtime.
Xây dựng hệ thống gợi ý nhạc cá nhân hóa	80%	Đã hoạt động tốt, tuy nhiên độ chính xác cần thêm dữ liệu người dùng thực tế để cải thiện.
Triển khai thực tế và kiểm thử	100%	Đã thực hiện Unit Test, System Test.

7.3. Hạn chế:

Bên cạnh những kết quả đạt được, đề tài vẫn còn một số hạn chế:

- *Vấn đề bản quyền*: Chưa có cơ chế tự động kiểm soát bản quyền nhạc do người dùng tải lên (Content ID).
- *Quy mô dữ liệu*: Kho nhạc hiện tại còn phụ thuộc nhiều vào API bên thứ ba, chưa tự chủ hoàn toàn về nội dung.
- *Tính năng Offline*: Chưa có cơ chế mã hóa DRM để bảo vệ file nhạc tải về khỏi việc sao chép trái phép.

7.4. Hướng phát triển trong tương lai:

Để phát triển JT-Harmony thành một sản phẩm thương mại hoàn chỉnh trong tương lai, nhóm đề xuất các hướng nâng cấp sau:

Về công nghệ

- *Nâng cấp AI*: Hoàn thiện lại mô hình gợi ý với tập dữ liệu lớn hơn (Big Data) để tăng độ chính xác. Áp dụng Deep Learning để phân tích tín hiệu âm thanh (audio analysis) nhằm gợi ý dựa trên giai điệu.

- *Hạ tầng*: Triển khai Auto-scaling (Tự động mở rộng) và CDN (Mạng phân phối nội dung) để phục vụ lượng người dùng lớn hơn.

Về chương trình

- Tích hợp tính năng chạy chữ Karaoke theo thời gian thực.
- Kiếm tiền (Monetization): Tích hợp cổng thanh toán để hỗ trợ các gói Premium và tính năng Donate (ủng hộ) cho nghệ sĩ.
- Cải thiện các lớp bảo mật để bảo vệ dữ liệu người dùng.
- Tối ưu hóa giao diện người dùng (UI/UX) để nâng cao trải nghiệm.
- Mở rộng hỗ trợ đa ngôn ngữ

TÀI LIỆU THAM KHẢO

- [1]. Node.js Foundation, "About this documentation", Node.js v25.x API Reference Documentation. [About this documentation | Node.js v25.2.0 Documentation](#). [Truy cập ngày: 16-11-2025].
- [2]. W3Schools, "Node.js Express Intro", W3Schools.com. https://www.w3schools.com/nodejs/nodejs_express.asp. [Truy cập ngày: 16-11-2025].
- [3]. Meta, "React Native - Create native apps for Android and iOS using React", Reactnative.dev. <https://reactnative.dev/>. [Truy cập ngày: 16-11-2025].
- [4]. Vercel, "Introduction to Next.js", Next.js Documentation. <https://nextjs.org/docs>. [Truy cập ngày: 16-11-2025].
- [5]. The PostgreSQL Global Development Group, "What Is PostgreSQL?", PostgreSQL 16 Documentation. <https://www.postgresql.org/docs/current/intro-what-is.html>. [Truy cập ngày: 16-11-2025].
- [6]. Socket.IO, "Introduction to Socket.IO", Socket.IO Documentatio. <https://socket.io/docs/v4/>. [Truy cập ngày: 16-11-2025].
- [7]. Expo Team, "Introduction to Expo", Expo Documentation. <https://docs.expo.dev/get-started/introduction/>. [Truy cập ngày: 16-11-2025].
- [8]. Tailwind Labs, "Tailwind CSS Documentation", v2.tailwindcss.com. <https://v2.tailwindcss.com/docs/>. [Truy cập ngày: 16-11-2025].
- [9]. NativeWind, "What is NativeWind?", NativeWind Documentation. <https://www.nativewind.dev/docs>. [Truy cập ngày: 16-11-2025].
- [10]. Sequelize Team, "Sequelize - Tutorials and Guides", Sequelize Documentation (v6). <https://sequelize.org/docs/v6/>. [Truy cập ngày: 16-11-2025].
- [11]. Neal Ford và Mark Richards, *Fundamentals of Software Architecture*, O'Reilly Media, Inc., 2020
- [12]. G. Coulouris, J. Dollimore, T. Kindberg, and G. Blair, *Distributed Systems: Concepts and Design*. Pearson Education, Inc., 2012.

PHỤ LỤC

PHỤ LỤC 1: BẢNG CÂU HỎI KHẢO SÁT

STT	Câu hỏi	Loại câu hỏi	Lựa chọn câu trả lời
PHẦN 1: THÔNG TIN CƠ BẢN VÀ THÓI QUEN SỬ DỤNG (Sàng lọc đối tượng)			
1	Bạn thuộc nhóm độ tuổi nào?	Trắc nghiệm - chọn 1	A. Dưới 18 B. 18 - 24 C. 25 - 34 D. Trên 35
2	Tần suất nghe nhạc trên thiết bị di động của bạn?	Trắc nghiệm - chọn 1	A. Hàng ngày (Trên 2 tiếng/ngày) B. Thường xuyên (3-4 lần/tuần) C. thỉnh thoảng D. Hiếm khi
3	Bạn đang sử dụng chủ yếu ứng dụng nghe nhạc nào? (Có thể chọn nhiều)	Trắc nghiệm -- chọn nhiều	A. Spotify B. Zing MP3 C. YouTube Music D. SoundCloud E. Nhaccuatui
PHẦN 2: ĐÁNH GIÁ TÍNH NĂNG CỐT LÕI & HỆ THỐNG GỢI Ý (Khảo sát nhu cầu tiếp cận nhạc)			
4	Với bạn, tính năng nào là BẮT BUỘC PHẢI CÓ ở một ứng dụng nghe nhạc? (Đánh giá mức độ quan trọng 1-5)	Xếp hạng	- Chất lượng âm thanh cao (Lossless/ High Quality). - Chế độ nghe Offline (Tải nhạc) - Tạo và quản lý Playlist cá

			nhân. - Hiện thị lời bài hát (Lyrics).
5	Bạn thường tìm kiếm bài hát mới theo cách nào?	Trắc nghiệm - chọn nhiều	A. Dựa vào Bảng xếp hạng/Top Trending (như Zing MP3) B. Dựa vào Gợi ý tự động theo sở thích/tâm trạng (như Spotify) C. Tìm kiếm thủ công
6	Bạn đánh giá thế nào về khả năng gợi ý nhạc của ứng dụng bạn đang dùng?	Trắc nghiệm - chọn 1	A. Rất tệ, toàn gợi ý bài không liên quan B. Tạm ổn nhưng chưa hiểu rõ gu của tôi C. Rất tốt, nhưng đôi khi tôi muốn biết thêm nhạc đang Hot (Trending) D. Tuyệt vời
7	Nếu có một ứng dụng kết hợp được cả Gợi ý AI thông minh (như Spotify) VÀ cập nhật Xu hướng thịnh hành (như Zing MP3), bạn có hứng thú không?	Trắc nghiệm - chọn 1	A. Rất hứng thú B. Có thể thử C. Không quan tâm lắm
PHẦN 3: TƯƠNG TÁC XÃ HỘI & CHIA SẺ (Tìm kiếm khuyết điểm của các App lớn hiện tại)			
8	Khi nghe một bài hát hay, bạn thường làm gì để chia sẻ cảm xúc?	Trắc nghiệm - chọn nhiều	A. Copy link và đăng lên Facebook/Instagram Story. B. Gửi tin nhắn riêng cho

			bạn bè qua Messenger/Zalo. C. Muốn bình luận ngay tại bài hát nhưng ứng dụng không hỗ trợ/hỗ trợ kém. D. Chỉ nghe một mình,
9	Bạn cảm thấy thế nào về tính năng tương tác (comment, chat) trên Spotify hoặc Zing MP3 hiện tại?	Trắc nghiệm - chọn 1	A. Rất hạn chế, gần như không có không gian thảo luận. B. Có nhưng tẻ nhạt, ít người dùng. C. Đủ dùng. D. Không quan tâm
10	Bạn có muốn vừa nghe nhạc vừa bình luận trực tiếp (Realtime comment) tại đúng đoạn nhạc cao trào (timestamp) giống như trên SoundCloud/Video livestream không?	Trắc nghiệm - chọn 1	A. Rất muốn, điều đó rất thú vị. B. Muốn thử. C. Không cần thiết
11	Bạn có thường xuyên nghe các bản nhạc Cover, Remix hoặc nhạc tự sáng tác (Indie/Underground) không?	Trắc nghiệm - chọn 1	A. Rất thường xuyên. B. thỉnh thoảng. C. Không bao giờ
12	Bạn đã từng sử dụng SoundCloud chưa? Nếu có, điều gì khiến bạn	Trắc nghiệm - chọn nhiều	A. Chưa từng sử dụng. B. Giao diện khó dùng, không thân thiện với người

	KHÔNG sử dụng nó làm ứng dụng nghe nhạc chính?		dùng phổ thông. C. Chất lượng nhạc không đồng đều. D. Kho nhạc chính thống (US-UK, Kpop...) bị thiếu hụt. E. Lý do khác...
PHẦN 4: NHU CẦU ĐĂNG TẢI			
13	Bạn (hoặc bạn bè của bạn) có nhu cầu tự đăng tải các bản thu âm/cover của chính mình lên nền tảng nghe nhạc không?	Trắc nghiệm - chọn 1	A. Có, tôi rất muốn chia sẻ giọng hát/sáng tác của mình. B. Có, bạn bè tôi có nhu cầu này. C. Không có nhu cầu.
14	Việc đăng tải nhạc lên các nền tảng lớn như Spotify/Zing MP3 hiện nay đối với bạn là?	Trắc nghiệm - chọn 1	A. Rất khó khăn, thủ tục phức tạp, phải thông qua nhà phân phối. B. Khá rắc rối.. C. Dễ dàng. D. Không rõ quy trình
15	Bạn có muốn một tính năng "Newsfeed Âm nhạc" - nơi bạn có thể lướt xem các bài hát/cover mới từ bạn bè và nghệ sĩ mình theo dõi (giống như lướt	Trắc nghiệm - chọn 1	A. Chắc chắn có. B. Có thể sẽ dùng. C. Không thích

	Facebook/TikTok) không?		
16	Bạn đánh giá thế nào về ý tưởng: "Một ứng dụng có trình phát nhạc chuyên nghiệp như Spotify nhưng cho phép đăng tải, bình luận và tương tác tự do như Mạng xã hội"?	Trắc nghiệm - chọn 1	A. Đây chính là ứng dụng tôi đang tìm kiếm. B. Ý tưởng rất hay, giải quyết được nhiều vấn đề. C. Bình thường. D. Không khả thi
Đánh giá mức độ quan trọng của các tính năng đề xuất cho JT-Harmony			
17	Tìm kiếm thông minh (theo lời bài hát, nghệ sĩ, tâm trạng)	Xếp hạng	Thang điểm 1-5
18	Upload nhạc Cover/Sáng tác trực tiếp từ điện thoại	Xếp hạng	Thang điểm 1-5
19	Theo dõi (Follow) người dùng khác và Nghệ sĩ	Xếp hạng	Thang điểm 1-5
20	Chat riêng tư (Private Message) với bạn bè trong app	Xếp hạng	Thang điểm 1-5
21	Thông báo (Notification) khi người dùng tương tác với bạn	Xếp hạng	Thang điểm 1-5
22	Giao diện Hiện đại	Xếp hạng	Thang điểm 1-5
23	Khả năng hoạt động mượt mà trên Mobile	Xếp hạng	Thang điểm 1-5

24	Bạn có sẵn sàng chuyển sang sử dụng thử nghiệm ứng dụng JT-Harmony nếu nó đáp ứng được các yếu tố trên không?	Trắc nghiệm - chọn 1	A. Chắc chắn sẽ dùng. B. Sẽ dùng thử xem sao. C. Cần xem xét thêm. D. Không.
25	Bạn có đóng góp ý kiến gì thêm về tính năng mà bạn mong muốn ở một ứng dụng nghe nhạc lý tưởng không?	Trả lời tự do	

PHỤ LỤC 2: KẾ HOẠCH THỰC HIỆN ĐỀ TÀI

Tuần	Từ ngày	Đến ngày	Nội dung công việc
1	11/08/2025	17/08/2025	<ul style="list-style-type: none"> - Nhận đề tài và phân tích công việc: Đăng ký nhóm - thành viên. - Phân tích, xác định các chức năng, phi chức năng (Backend, Frontend). - Thiết kế các UI/UX trên Figma. - Thiết kế các sơ đồ: usecase, sequence diagram, class diagram,...
2	18/08/2025	24/08/2025	<ul style="list-style-type: none"> - Chọn lựa và thiết kế database: Chính sửa, hoàn thiện các sơ đồ. - Chọn và cấu hình môi trường phát triển (NodeJS, IntelliJ,...). - Chọn cơ sở dữ liệu (PostgreSQL,...). - Thiết kế database dựa trên Class Diagram và Use Case Diagram. - Viết script SQL để tạo bảng và dữ liệu mẫu.
3	25/08/2025	31/08/2025	<ul style="list-style-type: none"> - Phát triển frontend: Tạo project, cài đặt các môi trường phát triển đã chọn. - Tạo layout và UI cho các trang chính. - Viết code frontend dùng React Native, kết nối API mockup để test giao diện tạm thời. - Thu thập phản hồi và điều chỉnh.
4	01/09/2025	07/09/2025	<ul style="list-style-type: none"> - Hệ thống người dùng (User Management): Thiết lập cấu trúc backend cơ bản. - Xây dựng chức năng đăng ký, đăng nhập người dùng. - Tạo mô hình người dùng và vai trò. - Thiết lập xác thực người dùng. - Viết unit test và kiểm thử API bằng Postman.
5	08/09/2025	14/09/2025	<ul style="list-style-type: none"> - Quản lý bài hát (Song Management): Tạo mô hình bài hát, nghệ sĩ và album. - Xây dựng chức năng lấy danh sách bài hát. - Xây dựng chức năng tìm kiếm bài hát theo tên. - Xây dựng chức năng xem chi tiết bài hát. - Quản lý hình ảnh và thông tin bài hát.

			- Kiểm thử các chức năng liên quan đến bài hát.
6	15/09/2025	21/09/2025	<ul style="list-style-type: none"> - Quản lý playlist cá nhân: Tạo mô hình playlist và danh sách phát. - Xây dựng chức năng tạo playlist mới. - Xây dựng chức năng xem danh sách playlist. - Xây dựng chức năng chỉnh sửa playlist (đổi tên, xóa). - Xây dựng chức năng thêm/bỏ bài hát khỏi playlist. - Phân quyền sở hữu playlist (chỉ chủ sở hữu được sửa). - Kiểm thử toàn bộ chức năng playlist.
7	22/09/2025	28/09/2025	<ul style="list-style-type: none"> - Tìm kiếm bài hát (cơ bản): Mở rộng chức năng tìm kiếm theo tên bài hát, tên nghệ sĩ,... - Hiện thị kết quả tìm kiếm theo độ phù hợp. - Phân trang kết quả tìm kiếm. - Gợi ý từ khóa tìm kiếm cơ bản. - Kiểm thử chức năng tìm kiếm.
8	29/09/2025	05/10/2025	<ul style="list-style-type: none"> - Lịch sử nghe nhạc và bài hát yêu thích: Tạo mô hình lịch sử nghe nhạc. - Ghi nhận hành vi nghe nhạc của người dùng. - Xây dựng chức năng xem lịch sử nghe gần nhất. - Tạo mô hình bài hát yêu thích (thích/thả tim). - Xây dựng chức năng thêm/bỏ bài hát khỏi yêu thích. - Hiện thị danh sách bài hát yêu thích. - Kiểm thử chức năng lịch sử và yêu thích.
9	06/10/2025	12/10/2025	<ul style="list-style-type: none"> - Chia sẻ nhạc (cơ bản): Tạo mô hình chia sẻ bài hát và playlist. - Xây dựng chức năng chia sẻ bài hát qua liên kết. - Xây dựng chức năng chia sẻ playlist cho người dùng khác. - Hiện thị danh sách nội dung đã được chia sẻ. - Xử lý quyền truy cập nội dung được chia sẻ.

			<ul style="list-style-type: none"> - Ghi nhận lượt chia sẻ. - Kiểm thử chức năng chia sẻ.
10	13/10/2025	19/10/2025	<ul style="list-style-type: none"> - Kết nối Frontend và Backend: Đồng bộ API với giao diện người dùng. - Kết nối chức năng đăng nhập/đăng ký với frontend. - Hiển thị danh sách bài hát từ backend lên giao diện. - Hiển thị và quản lý playlist trên frontend. - Đồng bộ trạng thái yêu thích và lịch sử nghe. - Xử lý lỗi kết nối và phản hồi người dùng. - Kiểm thử tích hợp toàn bộ chức năng cơ bản.
11	20/10/2025	26/10/2025	<ul style="list-style-type: none"> - Gợi ý nhạc cá nhân hóa: Thu thập dữ liệu hành vi người dùng (nghe, yêu thích). - Phân tích sở thích dựa trên lịch sử nghe. - Xây dựng cơ chế gợi ý theo thể loại yêu thích. - Xây dựng gợi ý theo nghệ sĩ thường nghe. - Hiển thị danh sách bài hát được gợi ý. - Cập nhật gợi ý theo thời gian thực (khi có hành vi mới). - Kiểm thử độ chính xác của gợi ý.
12	27/10/2025	02/11/2025	<ul style="list-style-type: none"> - Thông báo cá nhân: Tạo mô hình thông báo người dùng. - Gửi thông báo khi có bài hát mới từ nghệ sĩ yêu thích. - Gửi thông báo khi có người chia sẻ nội dung với mình. - Hiển thị danh sách thông báo chưa đọc. - Đánh dấu thông báo đã đọc. - Quản lý cài đặt thông báo. - Kiểm thử chức năng thông báo.
13	03/11/2025	09/11/2025	<ul style="list-style-type: none"> - Tối ưu và kiểm thử toàn hệ thống: Kiểm thử toàn bộ các chức năng đã phát triển. - Phát hiện và sửa lỗi (bug) trong quy trình xử lý. - Tối ưu hiệu suất phản hồi của API. - Kiểm tra tính nhất quán dữ liệu giữa các module. - Đảm bảo tính bảo mật cơ bản (xác thực, phân quyền).

			<ul style="list-style-type: none"> - Chuẩn bị dữ liệu mẫu cho demo. - Phát triển thêm các chức năng nâng cao. - Tổng hợp kết quả kiểm thử và báo cáo lỗi.
14	10/11/2025	16/11/2025	<ul style="list-style-type: none"> - Chỉnh sửa và hoàn thiện: Điều chỉnh giao diện và chức năng theo phản hồi. - Viết báo cáo phân tích, thiết kế, triển khai hệ thống. - Tổng hợp các tài liệu đã có. - Chuẩn bị demo cho giảng viên và nhóm.
15	17/11/2025	23/11/2025	<ul style="list-style-type: none"> - Viết báo cáo và chuẩn bị bảo vệ: Viết phần kết luận và đề xuất phát triển. - Tạo slide thuyết trình. - Kiểm tra toàn bộ hệ thống trước khi bảo vệ. - Chuẩn bị câu trả lời cho các câu hỏi thường gặp. - In poster.

PHỤ LỤC 3: NHẬT KÍ THỰC HIỆN ĐỀ TÀI

STT	Tuần	Nội dung công việc
1	11/08/2025 - 17/08/2025	<ul style="list-style-type: none">- Họp nhóm, bàn bạc mục tiêu cụ thể, xác định công việc cần làm.- Lên kế hoạch thực hiện đề tài.- Xác định và tìm hiểu các công nghệ sử dụng: React native, Nodejs/ExpressJs, PostgreSQL, SocketIO,...
2	18/08/2025 - 24/08/2025	<ul style="list-style-type: none">- Phân tích yêu cầu chức năng, phi chức năng.- Xác định actor, danh sách chức năng chính.- Thiết kế database diagram, sơ đồ use case tổng quát.
3	25/08/2025 - 31/08/2025	<ul style="list-style-type: none">- Chỉnh sửa database diagram.- Khởi tạo back end, api CRUD cơ bản.- Tìm kiếm/thiết kế UI.
4	01/09/2025 - 07/09/2025	<ul style="list-style-type: none">- Xây dựng api authen, user.- Chỉnh sửa database diagram.- Phân tích cấu trúc source meta Spotify Web API.
5	08/09/2025 - 14/09/2025	<ul style="list-style-type: none">- Kiểm thử API với post-man.- Code front-end mobile.- Chỉnh sửa usecase tổng quát và database diagram,
6	15/09/2025 - 22/09/2025	<ul style="list-style-type: none">- Xây dựng API cho quản lý nhạc, kết nối với API của Spotify.- Tích hợp API Authen vào mobile.
7	23/09/2025 - 29/09/2025	<ul style="list-style-type: none">- Xây dựng API cho quản lý bài đăng.- Tích hợp API quản lý nhạc vào mobile.- Chỉnh sửa database diagram.
8	30/09/10/2025 -	<ul style="list-style-type: none">- Cập nhật các tính năng đăng nhập, đăng ký, bỏ

	06/10/2025	<p>sung thêm tính năng liên kết tài khoản.</p> <ul style="list-style-type: none"> - Hoàn thiện các chức năng cập nhật, xác thực thông tin cá nhân. - Chỉnh sửa database và các API quản lý nhạc.
9	07/10/2025 - 13/10/2025	<ul style="list-style-type: none"> - Hiện thực các chức năng nghe, phát, tìm kiếm nhạc, xem thông tin. - Tích hợp socket.io vào hệ thống. - Xây dựng API cho các tính năng mạng xã hội. - Cập nhật tài liệu.
10	14/10/2025 - 20/10/2025	<ul style="list-style-type: none"> - Gắn API mạng xã hội vào mobile: đăng bài/cover, bình luận. Like, share cơ bản. - Hiện thực các chức năng quản lý danh sách phát và danh mục yêu thích, lịch sử nghe nhạc. - Tích hợp Gemini API vào back-end và tiến hành training thông qua prompt.
11	21/10/2025 - 27/10/2025	<ul style="list-style-type: none"> - Xây dựng chức năng đề xuất, gợi ý dựa trên hành vi và lịch sử yêu thích của người dùng lên mobile. - Xây dựng logic khách vắng lai trên mobile. - Hiện thực chức năng chat real-time, thời người người dùng, nghệ sĩ.
12	28/10/2025 - 03/11/2025	<ul style="list-style-type: none"> - Xây dựng API thông báo. - Tích hợp Redis vào back-end để cải thiện tốc độ truy vấn đối với các request thường xuyên và có data lớn. - Code front-end web admin. - Chỉnh sửa tài liệu.
13	04/11/2025 - 10/11/2025	<ul style="list-style-type: none"> - Hiện thực chức năng Đăng nhập trên web admin, các chức năng quản lý, thống kê thông tin. - Xây dựng tính năng bình luận theo thời lượng bài

		hát. - Chỉnh sửa tài liệu.
14	11/11/2025 - 17/11/2025	- Xây dựng tính năng khảo sát cho người dùng lần đầu đăng nhập trên mobile. - Test deploy web và server trên vercel.
15	18/11/2025 - 24/11/2025	- Thực hiện kiểm thử toàn hệ thống lần cuối trước khi deploy chính thức. - Deploy mobile, web và server. - Cập nhật tài liệu
16	25/11/2025 - 01/12/2025	- Hoàn thiện tài liệu báo cáo. - Làm slide báo cáo. - Làm video.