

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



BÁO CÁO ĐỒ ÁN TỐT NGHIỆP

**ĐỀ TÀI: XÂY DỰNG WEBSITE NGHE NHẠC TÍCH HỢP CHỨC
NĂNG GỢI Ý BÀI HÁT**

Giảng viên hướng dẫn: TS. NGUYỄN THỊ HẢI BÌNH

Sinh viên thực hiện: NGUYỄN HOÀNG HIỆP

Lớp : CQ.60.CNTT

Khoá : 60

Tp. Hồ Chí Minh, năm 2023

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



BÁO CÁO ĐỒ ÁN TỐT NGHIỆP

**ĐỀ TÀI: XÂY DỰNG WEBSITE NGHE NHẠC TÍCH HỢP CHỨC
NĂNG GỢI Ý BÀI HÁT**

Giảng viên hướng dẫn: TS. NGUYỄN THỊ HẢI BÌNH

Sinh viên thực hiện: NGUYỄN HOÀNG HIỆP

Lớp : CQ.60.CNTT

Khoá : 60

Tp. Hồ Chí Minh, năm 2023

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

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

Mã sinh viên: 6051071147 **Họ tên SV:** Nguyễn Hoàng Hiệp.....

Khóa: 60..... **Lớp:** CQ.60.CNTT.....

1. Tên đề tài

Xây dựng website nghe nhạc tích hợp chức năng gợi ý bài hát

2. Mục đích, yêu cầu

a. Mục đích

Xây dựng website nghe nhạc tích hợp hệ thống gợi ý content-based filtering và collaborative filtering để gợi ý bài hát cho người sử dụng.

b. Yêu cầu

- Yêu cầu công nghệ:

+ Về front-end: Hiểu và sử dụng HTML, CSS, javascript, thư viện jQuery, ajax, bootstrap.

+ Về back-end: Hiểu và sử dụng ngôn ngữ lập trình c# .Net MVC, database SQL server, entity framework, thư viện ML.Net.

- Yêu cầu chức năng: Xây dựng website nghe nhạc với các chức năng sau:

+ Chức năng đăng nhập: Người dùng đăng nhập hoặc tạo tài khoản.

+ Chức năng nghe nhạc: Website hiện danh sách nhạc, có thể nghe nhạc phát ngẫu nhiên, có thể nghe lặp lại một bài hát.

+ Chức năng tạo danh sách phát nhạc: Người dùng có thể tạo nhiều danh sách phát nhạc riêng.

+ Công cụ admin: Admin thực hiện thao tác CRUD về bài hát, thể loại, ca sĩ, album, nhạc sĩ.

- + Chức năng tìm kiếm: Người dùng có thể tìm kiếm nhạc, ca sĩ, album.
- + Chức năng ấn: Website sẽ ghi nhớ sự truy cập về bài hát của người dùng, tính thời lượng nghe một bài hát của người dùng. Website thực hiện gợi ý dựa trên lượt truy cập và thời lượng nghe bài hát của người dùng.
- Yêu cầu phi chức năng: Giao diện thân thiện, dễ sử dụng.
- Yêu cầu về code: Đơn giản dễ hiểu.

3. Nội dung và phạm vi đề tài

- Nội dung:
 - + Tổng quan về bài toán.
 - + Lý thuyết các công nghệ sử dụng.
 - + Hệ thống gợi ý
 - + Thiết kế hệ thống
 - + Xây dựng chương trình
 - + Kết quả thực nghiệm chức năng gợi ý bài hát
- Phạm vi đề tài:
 - + Website nghe nhạc hướng đến tất cả các đối tượng âm nhạc.
 - + Nghiên cứu về học máy.

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

- Công cụ: Hai công cụ chính là Visual studio và Microsoft SQL server management studio.
- Công nghệ và ngôn ngữ lập trình:
 - + Về front-end: HTML, CSS, javascript, jQuery, ajax, bootstrap.
 - + Về back-end: Ngôn ngữ lập trình c# .Net MVC, database SQL server, entity framework, thư viện ML.Net.

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

- Xây dựng website nghe nhạc trực tuyến đầy đủ với các yêu cầu về chức năng cơ bản.

- Xây dựng hệ thống gợi ý content-based filtering để gợi ý được các bài hát phù hợp với sở thích nghe nhạc của người dùng.
- Xây dựng hệ thống gợi ý collaborative filtering để gợi ý bài hát cho người dùng dựa trên những người có sở thích nghe tương tự.

6. Giáo viên và cán bộ hướng dẫn

Họ tên: Ts.Nguyễn Thị Hải Bình

Đơn vị công tác: Trường Đại học Công nghệ tp.Hồ Chí Minh

Điện thoại: 0339530712

Email: nth.binh@hutech.edu.vn

Ngày tháng năm 2023 Trưởng BM Công nghệ Thông tin	Đã giao nhiệm vụ TKTN Giáo viên hướng dẫn
ThS. Trần Phong Nhã	

Đã nhận nhiệm vụ TKTN

Sinh viên: Nguyễn Hoàng Hiệp

Điện thoại: 0812356946

Ký tên:

Email: 6051071147@st.utc2.edu.vn

LỜI CẢM ƠN

Để hoàn thành đề tài đồ án này trước hết em xin gửi đến quý thầy, cô **Bộ môn Công nghệ thông tin – Phân hiệu Trường Đại học Giao thông Vận tải tại Thành phố Hồ Chí Minh** lời cảm ơn chân thành vì đã truyền đạt cho em những kiến thức không chỉ từ sách vở, mà còn những kinh nghiệm quý giá từ cuộc sống trong khoảng thời gian học tập tại trường.

Đặc biệt, em xin gửi lời cảm ơn đến cô Nguyễn Thị Hải Bình, người cô đã tận tâm, hướng dẫn nhiệt tình để em có nền tảng vững chắc để thực hiện đồ án tốt nghiệp.

Vì thời gian làm đề tài đồ án có hạn cũng như hiểu biết của em còn hạn chế, em cũng đã nỗ lực hết sức để hoàn thành bài báo cáo một cách tốt nhất, nhưng chắc chắn vẫn sẽ có những thiếu sót không thể tránh khỏi. Em kính mong nhận được sự thông cảm và những ý kiến đóng góp chân thành từ quý thầy cô.

Sau cùng, em xin kính chúc Quý Thầy Cô trong **Bộ môn Công nghệ thông tin** luôn mạnh khỏe, hạnh phúc và thành công hơn nữa trong công việc cũng như trong cuộc sống.

Em xin chân thành cảm ơn!

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Tp. Hồ Chí Minh, ngày tháng năm
Giáo viên hướng dẫn

Nguyễn Thị Hải Bình

MỤC LỤC

NHIỆM VỤ THIẾT KẾ TỐT NGHIỆP	1
LỜI CẢM ƠN	4
DANH MỤC CHỮ VIẾT TẮT	9
DANH MỤC HÌNH ẢNH.....	10
TỔNG QUAN	12
Tổng quan về đề tài	12
Lý do chọn đề tài	12
Phát biểu bài toán	12
Mô tả hoạt động nghiệp vụ	13
Mục tiêu và nhiệm vụ nghiên cứu	13
Phạm vi.....	13
Cấu trúc đồ án tốt nghiệp	13
CHƯƠNG 1. CƠ SỞ LÝ THUYẾT.....	14
1.1 Lý thuyết các công nghệ sử dụng về front-end	14
1.1.1 HTML	14
1.1.2 CSS	14
1.1.3 Javascript	14
1.1.4 Bootstrap.....	15
1.1.5 jQuery	15
1.1.6 Ajax	16
1.2 Lý thuyết công nghệ sử dụng về back-end.....	16
1.2.1 Ngôn ngữ c#	16
1.2.2 ASP.NET MVC	17
1.2.3 SQL server.....	17
1.2.4 Entity framework.....	18

1.2.5 ML.NET	18
CHƯƠNG 2. HỆ THỐNG GỢI Ý	23
2.1 Tổng quan hệ thống gợi ý.....	23
2.1.1 Machine learning	23
2.1.2 Tổng quan hệ thống gợi ý (Recommender systems)	24
2.2 Hệ thống gợi ý dựa trên nội dung (content-based filtering).....	26
2.3 Hệ thống gợi ý dựa trên lọc cộng tác (collaborative filtering).....	28
CHƯƠNG 3. PHÂN TÍCH THIẾT KẾ HỆ THỐNG	34
3.1 Tổng quát bài toán.....	34
3.1.1 Phát biểu bài toán	34
3.1.2 Mô tả hoạt động nghiệp vụ.....	34
3.2 Sơ đồ BFD	35
3.3 Sơ đồ Usecase.....	36
3.4 Sơ đồ hoạt động.....	38
CHƯƠNG 4. THIẾT KẾ CHƯƠNG TRÌNH	53
4.1 Thiết kế cơ sở dữ liệu	53
4.1.1 Mô hình quan hệ thực thể (mô hình ER).....	53
4.1.2 Biểu đồ các bảng trong cơ sở dữ liệu	54
4.2 Thiết kế hệ thống gợi ý.....	54
4.2.1 Ý tưởng:.....	54
4.2.2 Kết quả thực nghiệm chức năng gợi ý content-based filtering.....	55
4.2.3 Kết quả thực nghiệm chức năng gợi ý collaborative filtering.....	59
4.3 Giao diện chương trình:.....	60
KẾT LUẬN	66
Kết quả đạt được.....	66
Vấn đề tồn đọng	66
Hướng phát triển.....	66

PHỤ LỤC	67
TÀI LIỆU THAM KHẢO	68

DANH MỤC CHỮ VIẾT TẮT

STT	Mô tả	Ý nghĩa	Ghi chú
1	CRUD	Create, read, update, delete	
2	TDD	Test-Driven-Development	
3	DOM	Document object model	
4	SEO	Search engine optimization	
5	AI	Artificial intelligence	
6	PCA	Principal component analysis	
7	NBCF	Neighborhood-based collaborative filtering	
8	ONNX	Open neural network exchange	
9	ER	Entity Relationship	

DANH MỤC HÌNH ẢNH

Hình 1. 1: Cơ chế hoạt động ML.Net	20
Hình 2. 1: Ví dụ mô hình không gian.....	27
Hình 2. 2: Kỹ thuật phân rã ma trận	29
Hình 3. 1: Sơ đồ BFD.....	35
Hình 3. 2: Sơ đồ Usecase admin.....	36
Hình 3. 3: Sơ đồ Usecase user	37
Hình 3. 4: Sơ đồ hoạt động đăng ký	38
Hình 3. 5: Sơ đồ hoạt động đăng nhập	39
Hình 3. 6: Sơ đồ hoạt động thêm bài hát	40
Hình 3. 7: Sơ đồ hoạt động sửa bài hát.....	40
Hình 3. 8: Sơ đồ hoạt động xóa bài hát	41
Hình 3. 9: Sơ đồ hoạt động tìm kiếm bài hát.....	41
Hình 3. 10: Sơ đồ hoạt động thêm ca sĩ.....	42
Hình 3. 11: Sơ đồ hoạt động sửa ca sĩ	42
Hình 3. 12: Sơ đồ hoạt động xóa ca sĩ.....	43
Hình 3. 13: Sơ đồ hoạt động tìm ca sĩ	43
Hình 3. 14: Sơ đồ hoạt động thêm thể loại	44
Hình 3. 15: Sơ đồ hoạt động sửa thể loại	44
Hình 3. 16: Sơ đồ hoạt động xóa thể loại	45
Hình 3. 17: Sơ đồ hoạt động thêm album.....	45
Hình 3. 18: Sơ đồ hoạt động sửa album	46
Hình 3. 19: Sơ đồ hoạt động xóa album	46
Hình 3. 20: Sơ đồ hoạt động tìm album	47
Hình 3. 21: Sơ đồ hoạt động thêm ca sĩ.....	47
Hình 3. 22: Sơ đồ hoạt động sửa ca sĩ	48
Hình 3. 23: Sơ đồ hoạt động xóa ca sĩ.....	48
Hình 3. 24: Sơ đồ hoạt động tìm nhạc sĩ	49
Hình 3. 25: Sơ đồ hoạt động thêm danh sách phát của user	49
Hình 3. 26: Sơ đồ hoạt động sửa danh sách phát của user	50
Hình 3. 27: Sơ đồ hoạt động xóa danh sách phát của user	50
Hình 3. 28: Sơ đồ hoạt động thêm nhạc vào danh sách phát của user	51
Hình 3. 29: Sơ đồ hoạt động xóa nhạc khỏi danh sách phát của user	51
Hình 3. 30: Sơ đồ hoạt động tìm kiếm của user	52
Hình 3. 31: Sơ đồ hoạt động nghe nhạc của user	52
Hình 4. 1: Mô hình ER.....	53
Hình 4. 2: Sơ đồ cơ sở dữ liệu	54
Hình 4. 3: Kết quả biểu diễn bài hát vào ma trận	56
Hình 4. 4: Kết quả biểu diễn bài hát sau khi chuẩn hóa	57
Hình 4. 5: Kết quả độ tương tự cosin giữa các bài hát	57
Hình 4. 6: Kết quả vector đặc trưng của người dùng	58
Hình 4. 7: Kết quả dự đoán độ yêu thích bài hát của người dùng	58
Hình 4. 8: Kết quả sau khi train mô hình bằng ML.Net.....	59
Hình 4. 9: Giao diện đăng ký.....	60

Hình 4. 10: Giao diện đăng nhập.....	60
Hình 4. 11: Giao diện quản lý nhạc.....	61
Hình 4. 12: Giao diện quản lý ca sĩ.....	61
Hình 4. 13: Giao diện quản lý thể loại.....	62
Hình 4. 14: Giao diện quản lý album.....	62
Hình 4. 15: Giao diện quản lý nhạc sĩ.....	63
Hình 4. 16: Giao diện trang chủ người dùng.....	63
Hình 4. 17: Giao diện phát nhạc.....	64
Hình 4. 18: Giao diện quản lý danh sách phát.....	64
Hình 4. 19: Giao diện tìm kiếm của user.....	65

TỔNG QUAN

Tổng quan về đề tài

Lý do chọn đề tài

- Thời kì công nghệ 4.0, khi mà cuộc sống trở nên nhộn nhịp và năng động, mọi người ngày càng quan tâm đến việc giải trí nhiều hơn. Vì vậy, một sân chơi dành cho những ai yêu thích âm nhạc là cần thiết. Một sản phẩm ‘Website nghe nhạc’ là giai đoạn cuối cùng để đưa thính giả lại gần hơn với âm nhạc.

- Tại đây, các thính giả sẽ tìm kiếm được những bài hát yêu thích của mình. Các nhạc sĩ có thể đưa những sản phẩm âm nhạc của bản thân đến với các thính giả. Mang lại cho mọi người những âm thanh, giai điệu tuyệt vời để tận hưởng những cung bậc cảm xúc khác nhau. Điều mà mọi người nhất là giới trẻ hiện nay đang quan tâm.

- Với số lượng thính giả dồi dào em hy vọng có thể góp một phần công sức của mình để phục vụ xã hội. Đó là lí do em chọn đề tài “Website nghe nhạc”.

- Phần mềm website nghe nhạc có chức năng phục vụ việc nghe nhạc. Website có thể gợi ý những bài hát phù hợp với sở thích nghe của người dùng. Chức năng của phần mềm này là quản lý thông tin liên quan đến bài hát , album, thể loại bài hát, ca sĩ,...

Phát biểu bài toán

- Qua những thông tin tìm hiểu thực tế, có thể phát biểu bài toán như sau:

+ Trong bài toán này người quản trị sẽ quản lý thông tin liên quan về bài hát, ca sĩ, album, thể loại, nhạc sĩ.

+ Người nghe nhạc có thể nghe nhạc, tạo danh sách phát nhạc. Người sử dụng có thể tìm kiếm bài hát, ca sĩ, album.

+ Người nghe nhạc và quản trị muốn thực hiện được các nhiệm vụ thì phải đăng nhập vào hệ thống theo username và password.

Mô tả hoạt động nghiệp vụ

- Đối với người quản lí (sau khi đăng nhập thành công)
- + CRUD: ca sĩ, bài hát, album, thể loại, nhạc sĩ.
- + Tra cứu thông tin về bài hát, ca sĩ, album, thể loại, nhạc sĩ.
- Đối với người dùng (sau khi đăng nhập)
- + Nghe nhạc, tạo danh sách phát nhạc
- + Tìm kiếm bài hát, ca sĩ, album.

Mục tiêu và nhiệm vụ nghiên cứu

Nghiên cứu và xây dựng website nghe nhạc với các chức năng cơ bản. Nghiên cứu xây dựng hệ thống gợi ý content-based filtering và collaborative filtering để gợi ý bài hát cho người dùng.

Phạm vi

- Website nghe nhạc hướng đến tất cả các đối tượng âm nhạc.
- Nghiên cứu về học máy.

Cấu trúc đề án tốt nghiệp

- Chương 1: Cơ sở lý thuyết
- Chương 2: Hệ thống gợi ý
- Chương 3: Phân tích thiết kế hệ thống
- Chương 4: Xây dựng chương trình

CHƯƠNG 1. CƠ SỞ LÝ THUYẾT

1.1 Lý thuyết các công nghệ sử dụng về front-end

1.1.1 HTML

- HTML có tên đầy đủ là Hypertext Markup Language nghĩa là ngôn ngữ đánh dấu siêu văn bản. HTML thường được sử dụng để tạo và cấu trúc các phần trong trang web và ứng dụng, phân chia các đoạn văn, heading, link, blockquote...

- HTML không phải là một ngôn ngữ lập trình mà chỉ là một ngôn ngữ đánh dấu. Điều này đồng nghĩa với việc HTML không thể thực hiện các chức năng “động”. Nói cách khác, HTML tương tự như phần mềm Microsoft Word, chỉ có tác dụng định dạng các thành phần có trong website.

1.1.2 CSS

- CSS là viết tắt của Cascading Style Sheets, một ngôn ngữ thiết kế đơn giản, xử lý một phần giao diện của trang web. CSS mô tả cách các phần tử HTML hiển thị trên màn hình và các phương tiện khác.

- Sử dụng CSS để kiểm soát màu chữ, cỡ chữ, kiểu chữ, khoảng cách giữa các đoạn văn bản, kích thước của các thành phần trên trang web, màu nền, thiết kế bố cục và cách trang web hiển thị trên những màn hình có kích thước khác nhau cũng như hàng loạt hiệu ứng khác.

- CSS rất hữu ích và tiện lợi. Nó có thể kiểm soát tất cả các trang trên một website.

1.1.3 Javascript

- Javascript là một ngôn ngữ lập trình kịch bản dựa vào đối tượng phát triển có sẵn hoặc tự định nghĩa. Javascript được sử dụng rộng rãi trong các ứng dụng

Website. Javascript được hỗ trợ hầu như trên tất cả các trình duyệt như Firefox, Chrome, ... trên máy tính lẫn điện thoại.

- Nhiệm vụ của Javascript là xử lý những đối tượng HTML trên trình duyệt. Nó có thể can thiệp với các hành động như thêm / xóa / sửa các thuộc tính CSS và các thẻ HTML một cách dễ dàng. Hay nói cách khác, Javascript là một ngôn ngữ lập trình trên trình duyệt ở phía client. Tuy nhiên, hiện nay với sự xuất hiện của NodeJS đã giúp cho Javascript có thể làm việc ở backend.

1.1.4 Bootstrap

Bootstrap là một front-end framework mã nguồn mở miễn phí giúp quá trình phát triển web được nhanh và dễ dàng hơn. Bootstrap bao gồm các mẫu thiết kế dựa trên HTML và CSS như typography, forms, buttons, tables, navigation, modals, image carousels... cũng như các plugin JavaScript tùy chọn. Người dùng có thể tạo các giao diện responsive design cho các thiết bị khác nhau với Bootstrap.

1.1.5 jQuery

- jQuery là một thư viện JavaScript nhẹ trên nhiều trình duyệt. jQuery đã được thiết kế để giúp điều hướng đến bất kỳ phần tử nào dễ dàng hơn hoặc thêm/gọi các trình xử lý sự kiện trên trang HTML của người dùng.

- Sử dụng jQuery, người dùng có thể dễ dàng chuyển đổi trang web trong tĩnh của mình thành một trang web động và mạnh mẽ.

- Toàn bộ logic jQuery xoay quanh điều kỳ diệu của hàm \$ (). \$ () theo mặc định, đại diện cho đối tượng jQuery. Khi được kết hợp với một bộ chọn, nó có thể đại diện cho nhiều Phần tử DOM.

1.1.6 Ajax

- AJAX là chữ viết tắt của cụm từ Asynchronous Javascript and XML. AJAX là phương thức trao đổi dữ liệu với máy chủ và cập nhật một hay nhiều phần của trang web, hoàn toàn không reload lại toàn bộ trang.

- Ajax được viết bằng Javascript chạy trên client, tức là mỗi browser sẽ chạy độc lập hoàn toàn không ảnh hưởng lẫn nhau. Về mặt kỹ thuật, nó đề cập đến việc sử dụng các đối tượng XMLHttpRequest để tương tác với một máy chủ web thông qua Javascript.

- AJAX được sử dụng để thực hiện một callback. Được dùng để thực hiện việc truy xuất dữ liệu hoặc lưu trữ dữ liệu mà không cần phải reload lại toàn bộ trang web. Với những server nhỏ thì việc này cũng tiết kiệm được băng thông cho người sử dụng. Client cần gì thì chỉ gửi dữ liệu phần đó, load lại một phần nhỏ để cập nhật thông tin chứ không load cả trang. Bằng cách này thì có thể giảm thiểu được tốc độ tải trang giúp người dùng có trải nghiệm tốt hơn. Trang web được tạo ra cũng sẽ đa dạng và động hơn.

1.2 Lý thuyết công nghệ sử dụng về back-end

1.2.1 Ngôn ngữ c#

- C Sharp hay còn được biết đến với tên gọi viết tắt C#, là dạng ngôn ngữ lập trình được phát triển và sản xuất bởi Microsoft, được xây dựng trên nền tảng của C++ và Java (2 loại ngôn ngữ mạnh nhất hiện nay), ra mắt chính thức vào 2001.

- C# – một ngôn ngữ lập trình về khoa học máy tính (computer science), là một ngôn ngữ được sử dụng để viết các chương trình phần mềm.

- C# tận dụng được hết các khả năng vượt trội mang đến cho người sử dụng sự tiện lợi, dễ dàng. C# chạy trên nền tảng của Microsoft, có sự hỗ trợ của .NET framework.

- C# phát huy được tối đa những ưu điểm tốt nhất của C và C++, có thể sử dụng để lập trình ra một ứng dụng bất kỳ và đều thực hiện được trên hầu hết các thiết bị di động.

- C# đang loại ngôn ngữ được đông đảo người dùng sử dụng và được đánh giá là loại ngôn ngữ mang tính hiện đại, trở thành loại ngôn ngữ phổ biến nhất thế giới.

1.2.2 ASP.NET MVC

- ASP.NET MVC là một framework phong phú để xây dựng các ứng dụng web bằng cách sử dụng mô hình thiết kế Model-View-Controller. ASP.NET MVC là một mã nguồn mở, có khả năng kiểm tra cao được tối ưu hóa cho việc sử dụng với ASP.NET.

- ASP.NET MVC cung cấp một cách dựa trên mô hình để xây dựng các trang web động cho phép tách rõ ràng các vấn đề. Nó cho phép người dùng kiểm soát đầy đủ về đánh dấu, hỗ trợ phát triển thân thiện với TDD và sử dụng các tiêu chuẩn web mới nhất.

1.2.3 SQL server

- SQL server là một dạng hệ quản trị cơ sở dữ liệu quan hệ (Relational Database Management System - Viết tắt là RDBMS). Nó được phát triển bởi gã khổng lồ trong làng công nghệ Microsoft vào năm 1989 và vẫn được sử dụng rộng rãi đến ngày nay.

- SQL server có chức năng chính là lưu trữ và truy xuất dữ liệu theo yêu cầu của các ứng dụng phần mềm khác. Người dùng sẽ lưu trữ dữ liệu vào đó và sử dụng các câu lệnh để tìm kiếm dữ liệu khi cần.

- SQL server sử dụng câu lệnh SQL (Transact-SQL) để trao đổi dữ liệu giữa máy khách (máy Client) và máy cài SQL Server.

1.2.4 Entity framework

- Entity Framework là một khung ORM(Object Relational Mapper) mã nguồn mở cho các ứng dụng .NET được Microsoft hỗ trợ, một phần của .NET Framework và một Open source ORM Framework.

- Entity Framework ra đời nhằm hỗ trợ sự tương tác giữa các ứng dụng trên nền tảng .NET với các cơ sở dữ liệu quan hệ. Hay, Entity Framework chính là công cụ giúp ánh xạ giữa các đối tượng trong ứng dụng, phần mềm với các bảng của một cơ sở dữ liệu quan hệ.

1.2.5 ML.NET

- ML.NET là thư viện máy học Mã nguồn mở và chạy cross-platform (Windows, Linux, macOS) của Microsoft. Người dùng có thể lập trình được thư viện này trên các nền tảng như Desktop, Web, hay build các Service. Nó được đánh giá là mạnh mẽ có thể làm được những gì một số thư viện khác làm được (chẳng hạn như scikit-learn viết bằng Python) và làm được những thứ mà thư viện khác không làm được. Với việc sở hữu các nền tảng công nghệ mạnh mẽ nhất, khách hàng sẵn có trải rộng khắp thế giới nên ML.NET được kỳ vọng rất lớn sẽ tạo ra được cơn sốt về công nghệ liên quan tới máy học viết bằng C#/F# và tạo ra thị trường lao động ở phân khúc này là rất khả thi.

- ML.NET hỗ trợ các bài toán nào liên quan tới Máy Học:

+ Classification: Ví dụ các bài toán về phân loại cảm xúc khách hàng tích cực hay tiêu cực từ các feedback của họ

+ Clustering : Ví dụ các bài toán về gom cụm khách hàng, giả sử có N khách hàng cần phải gom thành k cụm, các cụm này chứa các đặc trưng khác nhau của khách hàng.

+ Regression/Predict continuous values: Ví dụ các bài toán về hồi quy như dự đoán giá nhà, giá taxi từ một tập dữ liệu giao dịch trong quá khứ, hãy dự đoán giá của nó là gì khi có một số dự kiện mới.

+ Anomaly Detection: Các bài toán về phát hiện bất thường, chẳng hạn như Phát hiện các giao dịch gian lận trong ngân hàng

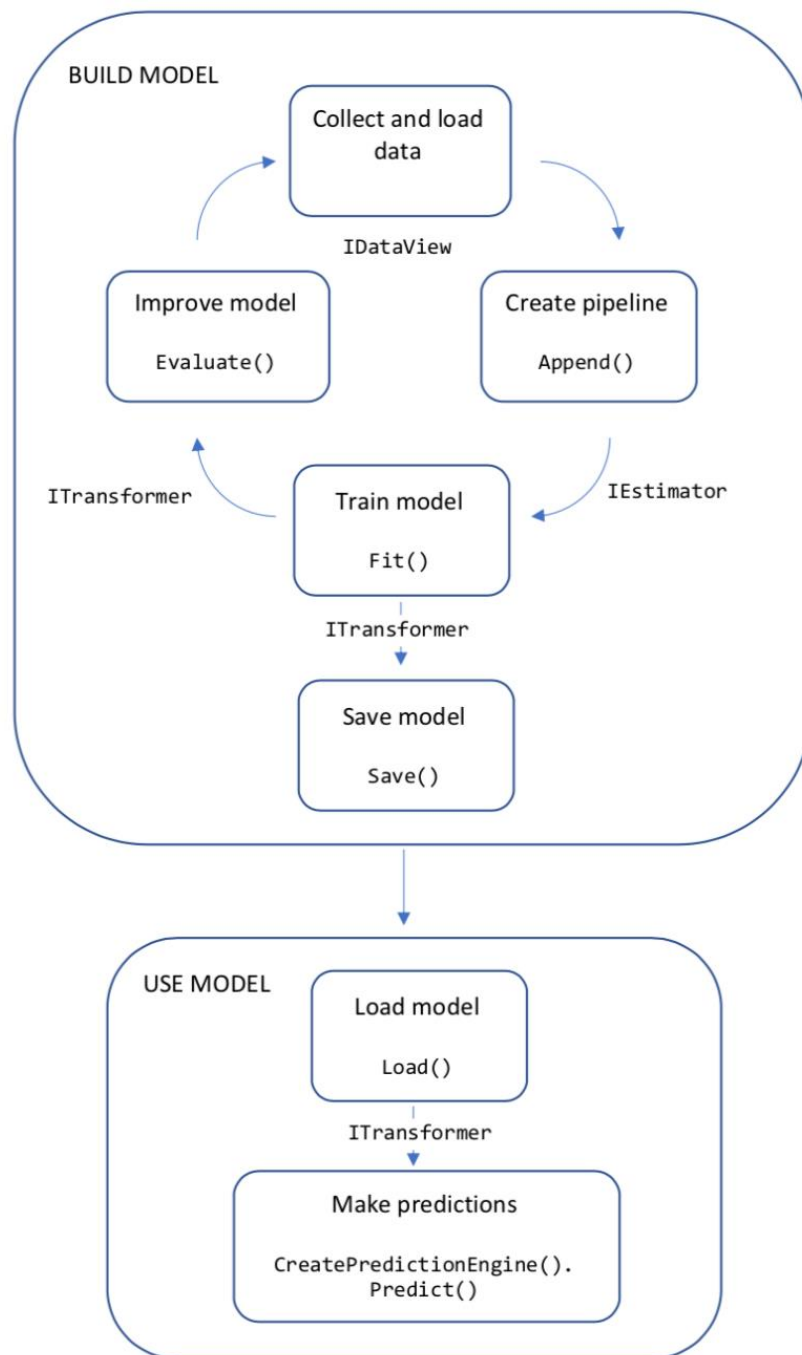
+ Recommendations: Các bài toán về khuyến nghị, ví dụ như làm sao quảng cáo được sản phẩm tới đúng khách hàng có nhu cầu, làm sao khi vào tiki thì nó gợi ý được các cuốn sách mà người mua quan tâm.

+ Time series/sequential data: các bài toán như dự báo thời tiết hay doanh số bán sản phẩm

+ Image classification: các bài toán về phân loại hình ảnh

+ Object detection: Các bài toán về Phát hiện đối tượng

- Cơ chế hoạt động của ML.NET:



Hình 1. 1: Cơ chế hoạt động ML.Net

+ Bước 1: là bước chuẩn bị dữ liệu, ở bước này tùy vào bài toán mà thu thập dữ liệu khác nhau, dữ liệu sau khi thu thập phải được làm sạch/ chuẩn hóa. Vì nguyên tắc vàng “garbage in garbage out”, dữ liệu có thể được lưu dạng text, csv, sql

server.... ứng với mỗi loại file ML.NET sẽ cung cấp các thư viện phù hợp để có thể tải dữ liệu đồng thời tự động mô hình hóa dữ liệu này thành mô hình hướng đối tượng. Ở bước 1 thường sẽ làm công đoạn tải dữ liệu xong thì làm công đoạn chia dữ liệu ban đầu thành 2 phần: Train set để huấn luyện mô hình (dùng cho bước 3), Test set để đánh giá mô hình (dùng cho bước 4). các dữ liệu được lưu vào IDataView object

+ Bước 2: Tùy vào mục đích bài toán máy học mà dùng các giải thuật khác nhau, có thể dùng Binary classification, Multiclass classification, Regression.... Bước này đại khái là chỉ định một quy trình hoạt động để trích xuất các đặc trưng và áp dụng thuật toán học máy cho phù hợp. Đối tượng tạo ra ở bước này là IEstimator

+ Bước 3: Tiến hành train mô hình bằng cách gọi phương thức Fit() của IEstimator. Kết quả của phương thức Fit() sẽ trả về một mô hình có kiểu ITransformer. Dữ liệu train là lấy Train set ở bước 1.

+ Bước 4: Sau khi train mô hình xong thì chưa có sai ngay, vì các bài toán máy học nó sẽ có kết quả dự đoán sai khác với thực tế, vấn đề là sự sai khác này có được chấp nhận hay có được tiếp tục sử dụng và tiếp tục cải tiến mô hình nữa hay không. Do đó khi train mô hình xong thì cần đánh giá mô hình này chất lượng ra sao. Lấy Test set ở bước 1 để đánh giá. Hàm đánh giá này tên là Evaluate() nó nằm trong các lớp giải thuật mà sử dụng để train mô hình, train mô hình dùng giải thuật nào thì khi đánh giá cũng dùng giải thuật đó. Ví dụ khi train dùng MulticlassClassification để train, thì khi đánh giá cũng dùng MulticlassClassification để đánh giá. Kết quả của hàm Evaluate() sẽ trả về một object XYZ Metrics. Với XYZ là giải thuật dùng để đánh giá, ví dụ dùng MulticlassClassification để đánh giá thì nó trả về MulticlassClassificationMetrics, dùng Regression để đánh giá thì nó lại trả về kết quả là RegressionMetrics... Nên sau khi gọi hàm đánh giá xong thì dựa vào các đối tượng kết quả trả về này mà quyết

định xem có dùng mô hình này được hay không (trong máy học không có mô hình sai chỉ có mô hình phù hợp hay không).

+ Bước 5: Lưu mô hình, sau khi đã đánh giá mô hình chất lượng ở bước 4 rồi thì lưu lại để lần sau chỉ tải mô hình ra sài, không phải chuẩn bị lại dữ liệu và train lại (vì các bước này rất tốn chi phí). File mô hình được lưu mặc định có đuôi .zip. Gọi phương thức `Save()` để lưu

+ Bước 6: Load mô hình đã được lưu ở bước 5.

+ Bước 7: là gọi hàm `CreatePredictionEngine().Predict()` để sử dụng mô hình nhằm tìm ra kết quả dự báo của chương trình. Truyền đối tượng `ITransformer` ở bước 6 vào sử dụng.

CHƯƠNG 2. HỆ THỐNG GỢI Ý

2.1 Tổng quan hệ thống gợi ý

2.1.1 Machine learning

-Machine learning là một nhánh của trí tuệ nhân tạo (AI) và khoa học máy tính, tập trung vào việc sử dụng dữ liệu và thuật toán để bắt chước hành động của con người, dần dần cải thiện độ chính xác của nó.

- Machine learning còn là một thành phần quan trọng của lĩnh vực khoa học dữ liệu đang phát triển. Thông qua việc sử dụng các phương pháp thống kê, các thuật toán được đào tạo để đưa ra các phân loại hoặc dự đoán và khám phá những thông tin chi tiết từ chính các dự án khai thác dữ liệu.

- Các phương pháp học máy:

+ Máy học có giám sát (Supervised learning): Đây là kỹ thuật học sử dụng cho các bài toán phân lớp (Classification). Một số thuật toán thường được sử dụng khi xây dựng bộ phân lớp này gồm có: Máy vector hỗ trợ (Support Vector Machine - SVM), cây quyết định (Decision Tree - DT), dựa trên vector trọng tâm (Centroid - based vector); hay tuyến tính bình phương nhỏ nhất (Linear Least Square Fit - LLSF).

+ Máy học không giám sát (Unsupervised learning): Đây là kỹ thuật học sử dụng cho các bài toán phân cụm, gom cụm (Clustering). Có rất nhiều thuật toán học không giám sát được ra đời và phát triển nhằm giải quyết bài toán phân cụm phục vụ khai thác hiệu quả nguồn dữ liệu chưa gán nhãn nhiều và rất đa dạng. Việc lựa chọn sử dụng thuật toán nào tùy thuộc vào dữ liệu và mục đích của từng bài toán. Trong đó có các thuật toán thường được sử dụng như: k-means, HAC (Hierarchical Agglomerative Clustering), SOM (Self-Organizing Map),...

+ Học tập bán giám sát (Semi-Supervised Learning): Học bán giám sát là một lớp của kỹ thuật học máy, sử dụng cả dữ liệu đã gán nhãn và chưa gán nhãn để huấn

luyện – điển hình là một lượng nhỏ dữ liệu có gán nhãn cùng với lượng lớn dữ liệu chưa gán nhãn. Một số thuật toán thường được sử dụng gồm có: thuật toán Cực đại kỳ vọng (EM – Expectation Maximization), SVM truyền dẫn (TSVM – Transductive Support Vector Machine), Self-training, Co-training và các phương pháp dựa trên đồ thị (graph-based).

2.1.2 Tổng quan hệ thống gợi ý (Recommender systems)

- Hệ thống gợi ý (Recommender systems hoặc Recommendation systems) là một dạng của hệ hỗ trợ ra quyết định, cung cấp giải pháp mang tính cá nhân hóa mà không phải trải qua quá trình tìm kiếm phức tạp. Hệ gợi ý học từ người dùng và gợi ý các sản phẩm tốt nhất trong số các sản phẩm phù hợp.

- Recommendation Systems là một mảng khá rộng của Machine Learning và có tuổi đời ít hơn so với Classification vì internet mới chỉ thực sự bùng nổ khoảng 10-15 năm đổ lại đây. Có hai thực thể chính trong Recommendation Systems là users và items. Users là người dùng. Items là sản phẩm, ví dụ như các bộ phim, bài hát, cuốn sách, clip, hoặc cũng có thể là các users khác trong bài toán gợi ý kết bạn. Mục đích chính của các Recommender Systems là dự đoán mức độ quan tâm của một user tới một item nào đó, qua đó có chiến lược recommend phù hợp.

- Xây dựng Utility Matrix:

+ Không có Utility matrix, gần như không thể gợi ý được sản phẩm tới người dùng, ngoài cách luôn luôn gợi ý các sản phẩm phổ biến nhất. Vì vậy, trong các Recommender Systems, việc xây dựng Utility Matrix là tối quan trọng. Tuy nhiên, việc xây dựng ma trận này thường có gặp nhiều khó khăn. Có hai hướng tiếp cận phổ biến để xác định giá trị rating cho mỗi cặp user-item trong Utility Matrix:

+ Nhờ người dùng rate sản phẩm. Amazon luôn nhờ người dùng rate các sản phẩm của họ bằng cách gửi các email nhắc nhở nhiều lần. Rất nhiều hệ thống khác cũng làm việc tương tự. Tuy nhiên, cách tiếp cận này có một vài hạn chế, vì thường

thì người dùng ít khi rate sản phẩm. Và nếu có, đó có thể là những đánh giá thiên lệch bởi những người sẵn sàng rate.

+ Hướng tiếp cận thứ hai là dựa trên hành vi của users. Nếu một người dùng mua một sản phẩm trên Amazon, xem một clip trên Youtube (có thể là nhiều lần), hay đọc một bài báo, thì có thể khẳng định rằng người dùng đó thích sản phẩm đó. Facebook cũng dựa trên việc người dùng like những nội dung nào để hiển thị newsfeed của họ những nội dung liên quan. Thường thì với cách này chỉ xây dựng được một ma trận với các thành phần là 1 và 0, với 1 thể hiện người dùng thích sản phẩm, 0 thể hiện chưa có thông tin. Trong trường hợp này, 0 không có nghĩa là thấp hơn 1, nó chỉ có nghĩa là người dùng chưa cung cấp thông tin. Có thể xây dựng ma trận với các giá trị cao hơn 1 thông qua thời gian hoặc số lượt mà người dùng xem một sản phẩm nào đó. Đôi khi, nút dislike cũng mang lại những lợi ích nhất định cho hệ thống, lúc này có thể gán giá trị -1

- Hai nhóm thuật toán chính: Các thuật toán machine learning trong hệ thống gợi ý thường được chia thành hai nhóm lớn:

+ Hệ thống dựa trên nội dung (content-based systems): nhóm thuật toán này gợi ý cho người dùng những sản phẩm tương tự như những sản phẩm mà người dùng đã có phản hồi tích cực. Hệ thống này cần xây dựng đặc trưng cho các sản phẩm sao cho những sản phẩm tương tự nhau có khoảng cách tới nhau nhỏ. Việc này khá tương tự như việc xây dựng các embedding cho các sản phẩm. Việc dự đoán cho mỗi người dùng hoàn toàn chỉ dựa trên lịch sử thông tin của người dùng đó.

+ Lọc cộng tác (collaborative filtering): nhóm thuật toán này không chỉ dựa trên thông tin về sản phẩm tương tự mà còn dựa trên hành vi của những người dùng tương tự. Ví dụ: người dùng A, B, C đều thích các bài hát của Noo Phước Thịnh. Ngoài ra, hệ thống biết rằng B, C cũng thích các bài hát của Bích Phương nhưng chưa có thông tin về việc liệu user A có thích Bích Phương hay không. Dựa trên

thông tin của những người dùng tương tự là B và C, hệ thống có thể dự đoán rằng A cũng thích Bích Phương và gợi ý các bài hát của ca sĩ này tới A.

2.2 Hệ thống gợi ý dựa trên nội dung (content-based filtering)

Các khái niệm được sử dụng trong Hệ thống dựa trên nội dung

- Tần số kỳ hạn (TF) và Tần số tài liệu nghịch đảo (IDF)

- + TF chỉ đơn giản là tần số của một từ trong tài liệu. IDF là nghịch đảo của tần số tài liệu trong toàn bộ kho tài liệu. TF-IDF được sử dụng chủ yếu vì hai lý do: Giả sử người dùng tìm kiếm "the rise of analytics" trên Google. Chắc chắn rằng "the" sẽ xảy ra thường xuyên hơn "analytics" nhưng tầm quan trọng tương đối của phân tích cao hơn quan điểm truy vấn tìm kiếm. Trong những trường hợp như vậy, trọng số TF-IDF phủ nhận ảnh hưởng của các từ tần số cao trong việc xác định tầm quan trọng của một mục.

- + Trong tính toán TF-IDF, log được sử dụng để làm giảm hiệu ứng của các từ tần số cao

- + Công thức tính tf:

$$tf(t, d) = \frac{f(t, d)}{\max\{f(w, d) : w \in d\}}$$

Trong đó:

- + $tf(t, d)$: tần suất xuất hiện của từ t trong văn bản d

- + $f(t, d)$: Số lần xuất hiện của từ t trong văn bản d

- + $\max(\{f(w, d) : w \in d\})$: Số lần xuất hiện của từ có số lần xuất hiện nhiều nhất trong văn bản d

+ Công thức tính IDF:

$$\text{idf}(t, D) = \log \frac{|D|}{|\{d \in D : t \in d\}|}$$

Trong đó:

+ $\text{idf}(t, D)$: giá trị idf của từ t trong tập văn bản

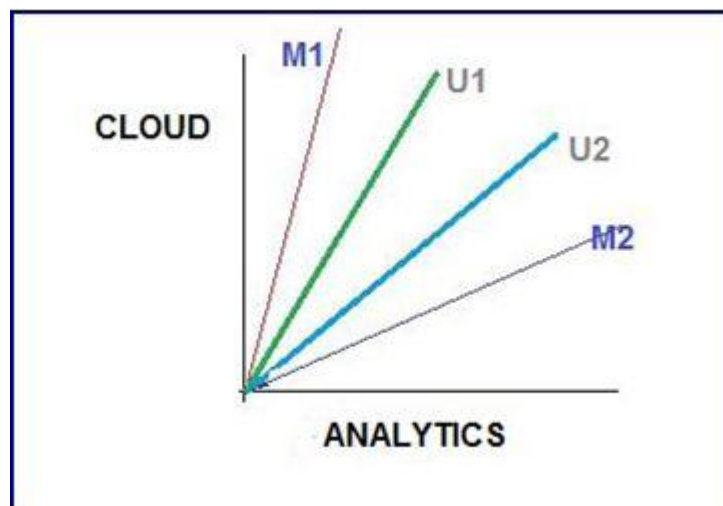
+ $|D|$: Tổng số văn bản trong tập D

+ $|\{d \in D : t \in d\}|$: thể hiện số văn bản trong tập D có chứa từ t .

- Mô hình không gian vector hoạt động:

+ Trong mô hình này, mỗi mục được lưu trữ dưới dạng vector các thuộc tính của nó (cũng là vector) trong không gian n chiều và các góc giữa các vector được tính toán để xác định sự giống nhau giữa các vector. Tiếp theo, các vector hồ sơ người dùng cũng được tạo dựa trên hành động của người dùng trên các thuộc tính trước đó của các mục và sự giống nhau giữa một mục và người dùng cũng được xác định theo cách tương tự.

+ Ví dụ:.



Hình 2. 1: Ví dụ mô hình không gian

+Hiển thị ở trên là biểu diễn 2-D của hai thuộc tính, Cloud và Analytics. M1 và M2 là tài liệu. U1 và U2 là người dùng. Tài liệu M2 thiên về Analytics hơn là Cloud trong khi M1 thiên về Cloud hơn Analytics.

+ Người dùng U1, thích các bài viết về chủ đề 'Cloud' hơn các bài viết về 'Analytics' và ngược lại cho người dùng U2. Phương pháp tính toán lượt thích / không thích / thước đo của người dùng được tính bằng cách lấy cosin của góc giữa vector hồ sơ người dùng và vector tài liệu.

+ Lý do cuối cùng đằng sau việc sử dụng cosin là giá trị của cosin sẽ tăng lên khi giá trị giảm của góc giữa đó biểu thị sự giống nhau hơn. Các vector được chuẩn hóa độ dài, sau đó chúng trở thành vector có độ dài 1, và sau đó phép tính cosin chỉ đơn giản là tích tổng của vector.

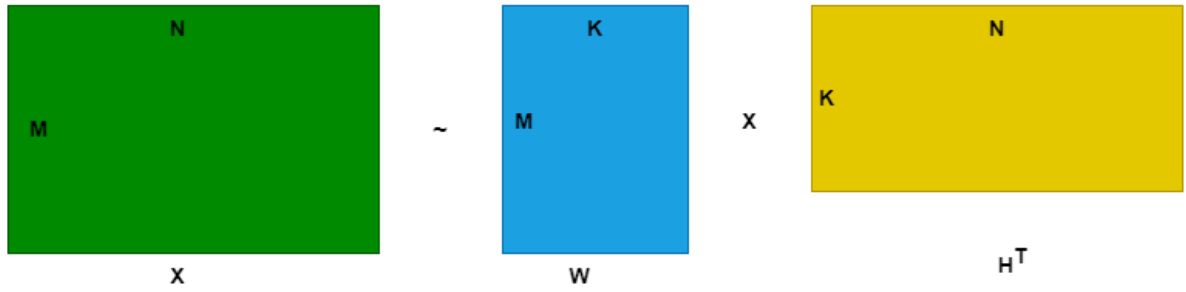
2.3 Hệ thống gợi ý dựa trên lọc cộng tác (collaborative filtering)

- Phương pháp lọc cộng tác hay hệ thống lọc cộng tác là phương pháp phân tích dữ liệu người dùng để tìm ra mối tương quan giữa các đối tượng người dùng. Lọc cộng tác hoạt động bằng cách xây dựng một cơ sở dữ liệu, lưu trữ dưới dạng ma trận người dùng (users) - sản phẩm (items) và mỗi dòng của nó là một vector.

- Website sử dụng phương pháp Matrix Factorization để thực hiện lọc cộng tác

- Matrix Factorization là một hướng tiếp cận khác của Collaborative Filtering, còn gọi là Matrix Decomposition, nghĩa là gợi ý bằng "kỹ thuật phân rã ma trận".

- Kỹ thuật phân rã ma trận là phương pháp chia một ma trận lớn X thành hai ma trận có kích thước nhỏ hơn là W và H , sao cho có thể xây dựng lại X từ hai ma trận nhỏ hơn này càng chính xác càng tốt, nghĩa là $X \sim WH^T$



Hình 2. 2: Kỹ thuật phân rã ma trận

- Có thể hiểu rằng, ý tưởng chính của Matrix Factorization là đặt items và users vào trong cùng một không gian thuộc tính ẩn. Trong đó, $W \in \mathbb{R}^{U \times K}$ là một ma trận mà mỗi dòng u là một vector bao gồm K nhân tố tiềm ẩn (latent factors) mô tả user u và $H \in \mathbb{R}^{I \times K}$ là một ma trận mà mỗi dòng i là một vector bao gồm K nhân tố tiềm ẩn mô tả cho item i .

- Áp dụng phương pháp này vào bài toán gợi ý, có x là một vector của item profile.

-Mục tiêu là tìm một vector w tương ứng với mỗi user sao cho ratings đã biết của user đó cho item (y) xấp xỉ với:

$$y \approx \mathbf{xw}$$

Mở rộng với Y là utility matrix, giả sử đã được điền hết giá trị, có:

$$\mathbf{Y} \approx \begin{bmatrix} \mathbf{x}_1 \mathbf{w}_1 & \mathbf{x}_1 \mathbf{w}_2 & \dots & \mathbf{x}_1 \mathbf{w}_N \\ \mathbf{x}_2 \mathbf{w}_1 & \mathbf{x}_2 \mathbf{w}_2 & \dots & \mathbf{x}_2 \mathbf{w}_N \\ \dots & \dots & \ddots & \dots \\ \mathbf{x}_M \mathbf{w}_1 & \mathbf{x}_M \mathbf{w}_2 & \dots & \mathbf{x}_M \mathbf{w}_N \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \dots \\ \mathbf{x}_M \end{bmatrix} [\mathbf{w}_1 \quad \mathbf{w}_2 \quad \dots \quad \mathbf{w}_N] = \mathbf{XW}$$

với M , N lần lượt là số users và số items.

- Lưu ý, do X là được xây dựng dựa trên thông tin mô tả item và quá trình xây dựng này độc lập với quá trình đi tìm hệ số phù hợp cho mỗi user nên việc xây dựng

item profile đóng vai trò quan trọng và có thể ảnh hưởng trực tiếp đến hiệu năng của mô hình. Thêm nữa, việc xây dựng mô hình riêng lẻ cho mỗi user dẫn đến kết quả chưa thực sự tốt vì không khai thác được đặc điểm giống nhau giữa các user.

- Giả sử rằng không cần xây dựng trước các item profile mà ma trận này có thể huấn luyện đồng thời với ma trận trọng số, hay nói cách khác bài toán này là bài toán tối ưu các ma trận \mathbf{X} và \mathbf{W} , trong đó \mathbf{X} là ma trận của toàn bộ các item profiles, mỗi hàng tương ứng với một item. Còn \mathbf{W} là ma trận của toàn bộ user models (các mô hình của users), mỗi cột tương ứng với một user. Xấp xỉ utility matrix $\mathbf{Y} \in \mathbb{R}^{M \times N}$ bằng tích của hai ma trận con là $\mathbf{X} \in \mathbb{R}^{M \times K}$ và $\mathbf{W} \in \mathbb{R}^{K \times N}$.

- K được chọn thường nhỏ hơn rất nhiều so với M và N , và cả hai ma trận \mathbf{X} và \mathbf{W} đều phải có bậc (rank) không được vượt quá K .

- Xây dựng và tối ưu hàm mất mát: Quy trình xây dựng và tối ưu hàm mất mát như sau:

- Hàm mất mát: Đầu tiên, xét hàm mất mát không có cả bias và biến tối ưu cho \mathbf{X} và \mathbf{W} :

$$\mathcal{L}(\mathbf{X}, \mathbf{W}) = \frac{1}{2s} \sum_{n=1}^N \sum_{m:r_{mn}=1} (y_{mn} - \mathbf{x}_m \mathbf{w}_n)^2 + \frac{\lambda}{2} (\|\mathbf{X}\|_F^2 + \|\mathbf{W}\|_F^2)$$

- Trong đó, $r_{mn} = 1$ nếu item thứ m đã được đánh giá bởi user thứ n , $\|\cdot\|$ là căn bậc hai của tổng bình phương tất cả các phần tử của ma trận, s là toàn bộ số ratings đã có. Thành phần thứ nhất chính là trung bình sai số của mô hình. Thành phần thứ hai trong hàm mất mát phía, có tách dựng giúp tránh overfitting.

- Tiếp theo sẽ tối ưu \mathbf{X} và \mathbf{W} bằng cách cố định từng ma trận và tối ưu ma trận còn lại cho tới khi hội tụ.

-Tối ưu hàm mất mát

+ Khi cố định \mathbf{X} , việc tối ưu \mathbf{W} được đưa về hàm:

$$\mathcal{L}(\mathbf{W}) = \frac{1}{2s} \sum_{n=1}^N \sum_{m:r_{mn}=1} (y_{mn} - \mathbf{x}_m \mathbf{w}_n)^2 + \frac{\lambda}{2} \|\mathbf{W}\|_F^2$$

+ Khi cố định \mathbf{W} , việc tối ưu \mathbf{X} được đưa về tối ưu hàm:

$$\mathcal{L}(\mathbf{X}) = \frac{1}{2s} \sum_{n=1}^N \sum_{m:r_{mn}=1} (y_{mn} - \mathbf{x}_m \mathbf{w}_n)^2 + \frac{\lambda}{2} \|\mathbf{X}\|_F^2$$

+ Bài toán này sẽ được tối ưu bằng Gradient Descent.

+ Có thể thấy rằng, bài toán tối ưu \mathbf{W} có thể được tách thành N bài toán nhỏ (N là số lượng users), mỗi bài toán tương ứng với việc đi tối ưu một cột của ma trận \mathbf{W} .

$$\mathcal{L}(\mathbf{w}_n) = \frac{1}{2s} \sum_{m:r_{mn}=1} (y_{mn} - \mathbf{x}_m \mathbf{w}_n)^2 + \frac{\lambda}{2} \|\mathbf{w}_n\|_2^2$$

+ Vì biểu thức $(y_{mn} - \mathbf{x}_m \mathbf{w}_n)^2$ chỉ phụ thuộc vào các items đã được user đang xét đánh giá, nên có thể đơn giản nó bằng cách sử dụng sub matrix, là matrix chỉ chứa các giá trị ratings đã biết. Khi đó, hàm mất mát có dạng:

$$\mathcal{L}(\mathbf{w}_n) = \frac{1}{2s} \|\hat{\mathbf{y}}_n - \hat{\mathbf{X}}_n \mathbf{w}_n\|^2 + \frac{\lambda}{2} \|\mathbf{w}_n\|_2^2$$

+ Và đạo hàm của nó là:

$$\frac{\partial \mathcal{L}(\mathbf{w}_n)}{\partial \mathbf{w}_n} = -\frac{1}{s} \hat{\mathbf{X}}_n^T (\hat{\mathbf{y}}_n - \hat{\mathbf{X}}_n \mathbf{w}_n) + \lambda \mathbf{w}_n$$

+ Vậy công thức cập nhật cho các cột của W là:

$$\mathbf{w}_n = \mathbf{w}_n - \eta \left(-\frac{1}{s} \hat{\mathbf{X}}_n^T (\hat{\mathbf{y}}_n - \hat{\mathbf{X}}_n \mathbf{w}_n) + \lambda \mathbf{w}_n \right)$$

+ Tương tự với X, mỗi hàng tương ứng với một item sẽ được tìm bằng cách tối ưu:

$$\mathcal{L}(\mathbf{x}_m) = \frac{1}{2s} \sum_{n:r_{mn}=1} (y_{mn} - \mathbf{x}_m \mathbf{w}_n)^2 + \frac{\lambda}{2} \|\mathbf{x}_m\|_2^2$$

+ Đặt \mathbf{W}_m là ma trận được tạo bằng các cột của W tương ứng với các users đã đánh giá items đó và sử dụng submatrix Y tương ứng là \mathbf{y}_m . Công thức trên sẽ trở thành:

$$\mathcal{L}(\mathbf{x}_m) = \frac{1}{2s} \|\hat{\mathbf{y}}^m - \mathbf{x}_m \hat{\mathbf{W}}_m\|_2^2 + \frac{\lambda}{2} \|\mathbf{x}_m\|_2^2$$

+ Vậy công thức cập nhật cho mỗi hàng của X là:

$$\mathbf{x}_m = \mathbf{x}_m - \eta \left(-\frac{1}{s} (\hat{\mathbf{y}}^m - \mathbf{x}_m \hat{\mathbf{W}}_m) \hat{\mathbf{W}}_m^T + \lambda \mathbf{x}_m \right)$$

+ Sau khi cố định X, tính W và ngược lại, cố định W và tính X cho đến khi các ma trận này hội tụ, thu được ma trận X và W cần tìm. Từ đó, dự đoán các giá trị ratings chưa biết.

+ Ngoài phương pháp trên, để tăng độ chính xác của thuật toán này, xét hàm mất mát với bias và hệ số tối ưu cho X và W.

+ Như trong NBCF, bước chuẩn hóa ma trận để tránh sự thiên lệch do sự khó tính hay dễ tính khác nhau giữa các users. Với MF có thể chuẩn không chuẩn hóa mà sử dụng trực tiếp các giá trị ratings ban đầu, bằng cách tối ưu các biases cùng lúc với \mathbf{X} và \mathbf{W} .

+ Trong trường hợp này, ratings của user m cho item n được xác định bởi công thức:

$$y_{mn} \approx \mathbf{x}_m \mathbf{w}_n + b_m + d_n + \mu$$

với b_m, d_n, μ lần lượt là bias của item m , user n và ratings là ratings trung bình của toàn bộ các ratings.

+ Và hàm mất mát trong trường hợp này có dạng:

$$\begin{aligned} \mathcal{L}(\mathbf{X}, \mathbf{W}, \mathbf{b}, \mathbf{d}) = & \frac{1}{2s} \sum_{n=1}^N \sum_{m:r_{mn}=1} (\mathbf{x}_m \mathbf{w}_n + b_m + d_n + \mu - y_{mn})^2 + \\ & + \frac{\lambda}{2} (\|\mathbf{X}\|_F^2 + \|\mathbf{W}\|_F^2 + \|\mathbf{b}\|_2^2 + \|\mathbf{d}\|_2^2) \end{aligned}$$

+ Tiến hành cố định \mathbf{X}, \mathbf{b} và tối ưu \mathbf{W}, \mathbf{d} và ngược lại, cố định \mathbf{W}, \mathbf{d} và tối ưu \mathbf{X}, \mathbf{b} , theo các công thức:

$$\begin{aligned} \mathbf{w}_n &= \mathbf{w}_n - \eta \frac{1}{s} \mathbf{X}_n (\mathbf{X}_n^T \mathbf{w}_n + b_n + d_n - y_{mn}) + \lambda \mathbf{w}_n \\ b_n &= b_n - \eta \frac{1}{s} \mathbf{1}^T (\mathbf{X}_n^T \mathbf{w}_n + b_n + d_n - y_{mn}) \\ \mathbf{x}_m &= \mathbf{x}_m - \eta \frac{1}{s} \mathbf{W}_m (\mathbf{W}_m^T \mathbf{x}_m + b_n + d_n - y_{mn}) + \lambda \mathbf{x}_m \\ d_m &= d_m - \eta \frac{1}{s} \mathbf{1}^T (\mathbf{W}_m^T \mathbf{x}_m + b_n + d_n - y_{mn}) \end{aligned}$$

+ Cuối cùng thu được các ma trận $\mathbf{X}, \mathbf{b}, \mathbf{W}, \mathbf{d}$, từ đó dự đoán các ratings chưa biết.

CHƯƠNG 3. PHÂN TÍCH THIẾT KẾ HỆ THỐNG

3.1 Tổng quát bài toán

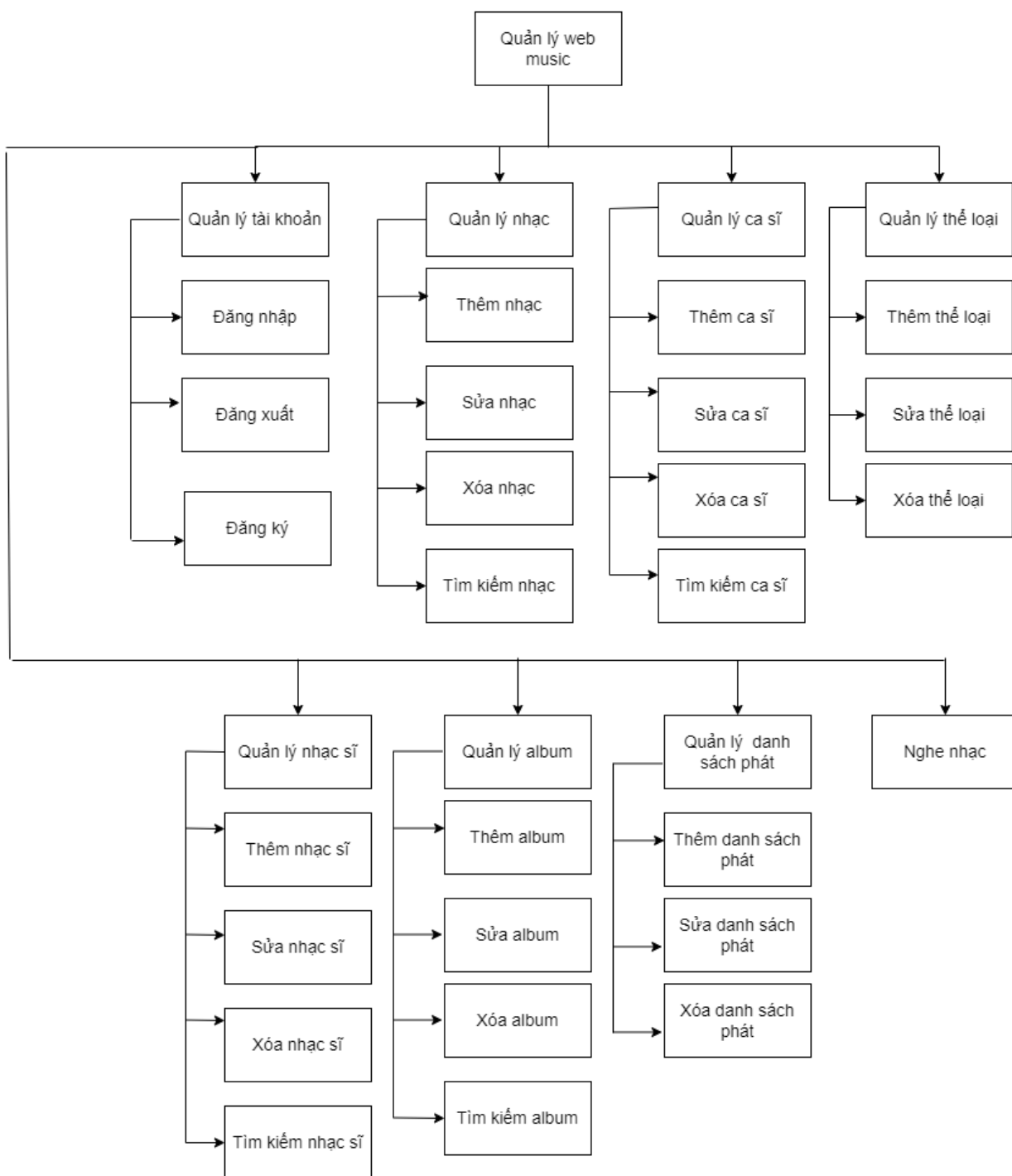
3.1.1 Phát biểu bài toán

- Qua những thông tin tìm hiểu thực tế, có thể phát biểu bài toán như sau:
 - + Trong bài toán này người quản trị sẽ quản lý thông tin liên quan về bài hát, ca sĩ, album, thể loại, nhạc sĩ
 - + Người nghe nhạc có thể nghe nhạc, tạo danh sách phát nhạc. Người sử dụng có thể tìm kiếm bài hát, ca sĩ, album.
 - + Người nghe nhạc và quản trị muốn thực hiện được các nhiệm vụ thì phải đăng nhập vào hệ thống theo username và password.

3.1.2 Mô tả hoạt động nghiệp vụ

- Đối với người quản lí (sau khi đăng nhập thành công)
 - + CRUD: ca sĩ, bài hát, album, thể loại, nhạc sĩ.
 - + Tra cứu thông tin về bài hát, ca sĩ, album, thể loại, nhạc sĩ.
- Đối với người dùng (sau khi đăng nhập)
 - + Nghe nhạc, tạo danh sách phát nhạc
 - + Tìm kiếm bài hát, ca sĩ, album.

3.2 Sơ đồ BFD



Hình 3. 1: Sơ đồ BFD

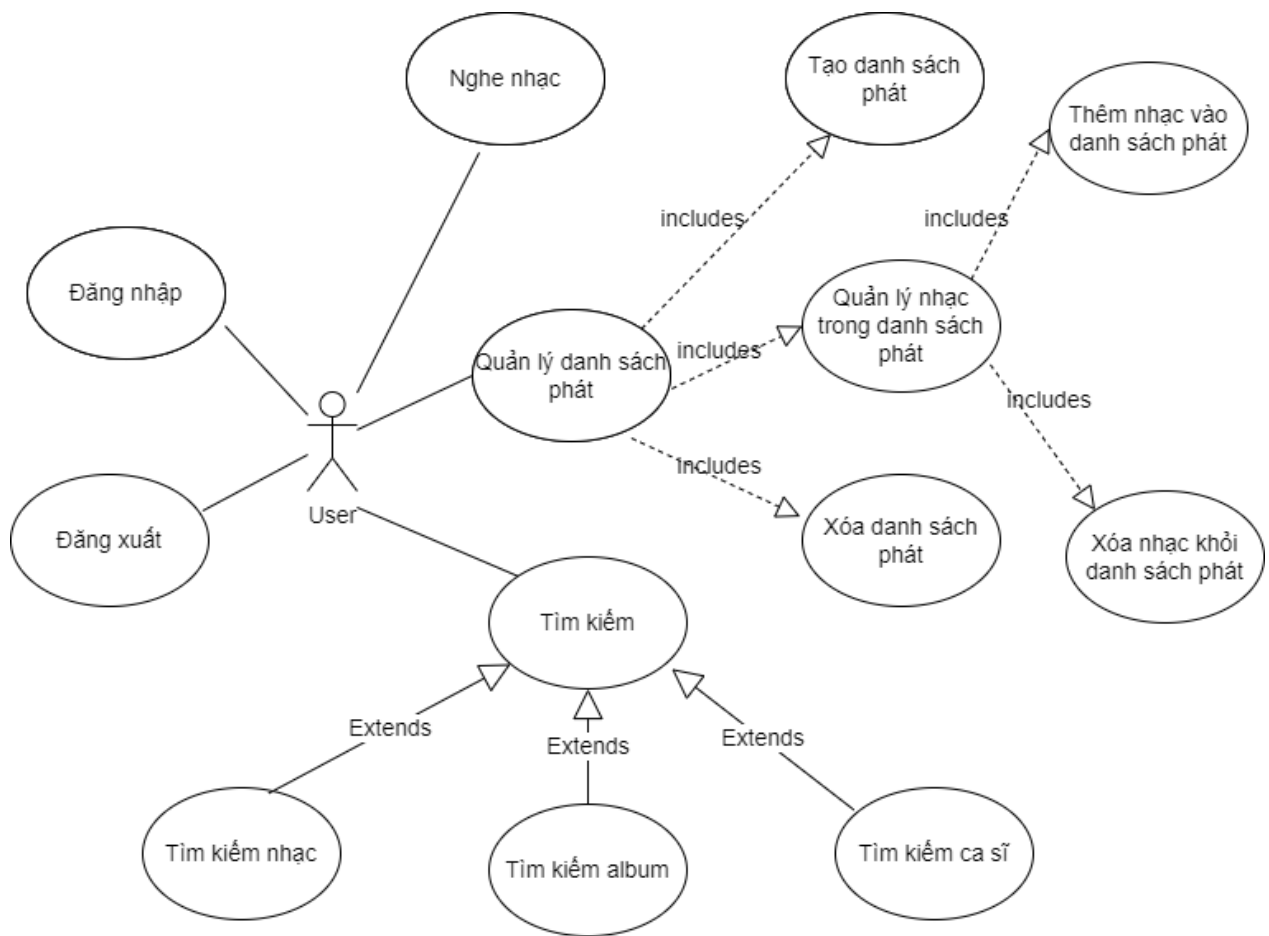
3.3 Sơ đồ Usecase

-Sơ đồ Usecase của Admin



Hình 3. 2: Sơ đồ Usecase admin

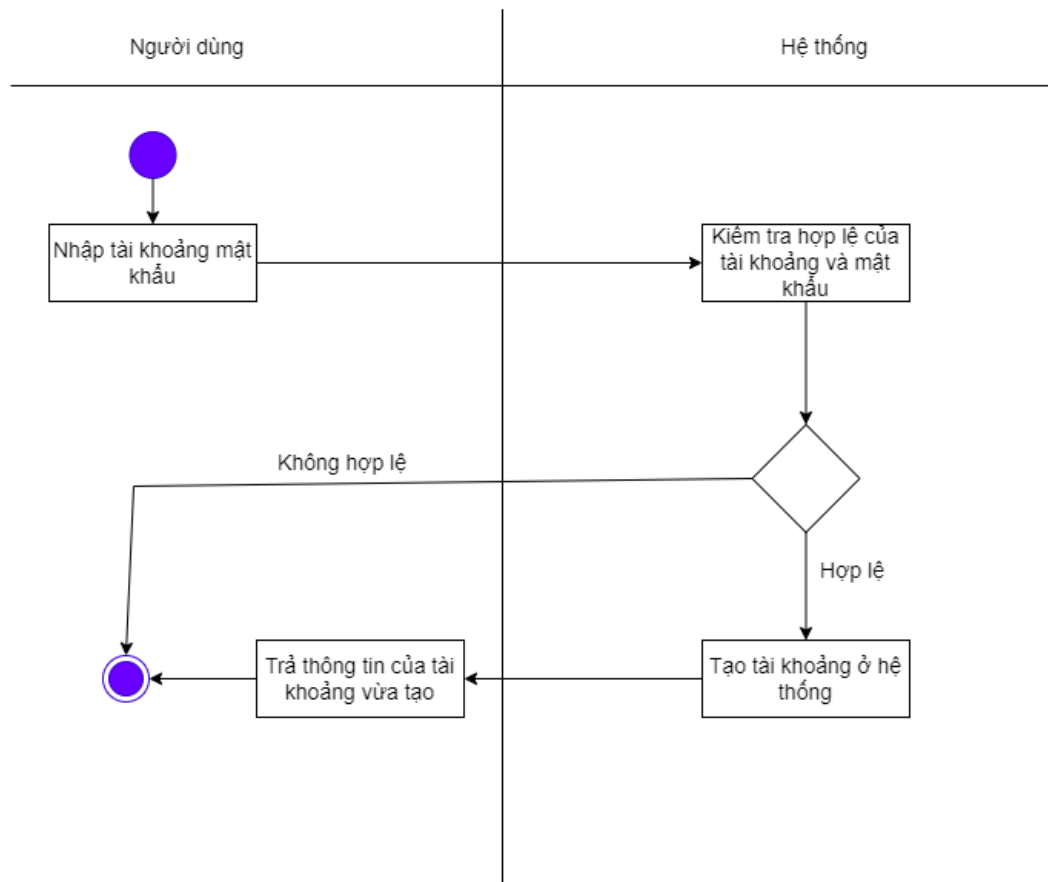
- Sơ đồ Usecase của user:



Hình 3. 3: Sơ đồ Usecase user

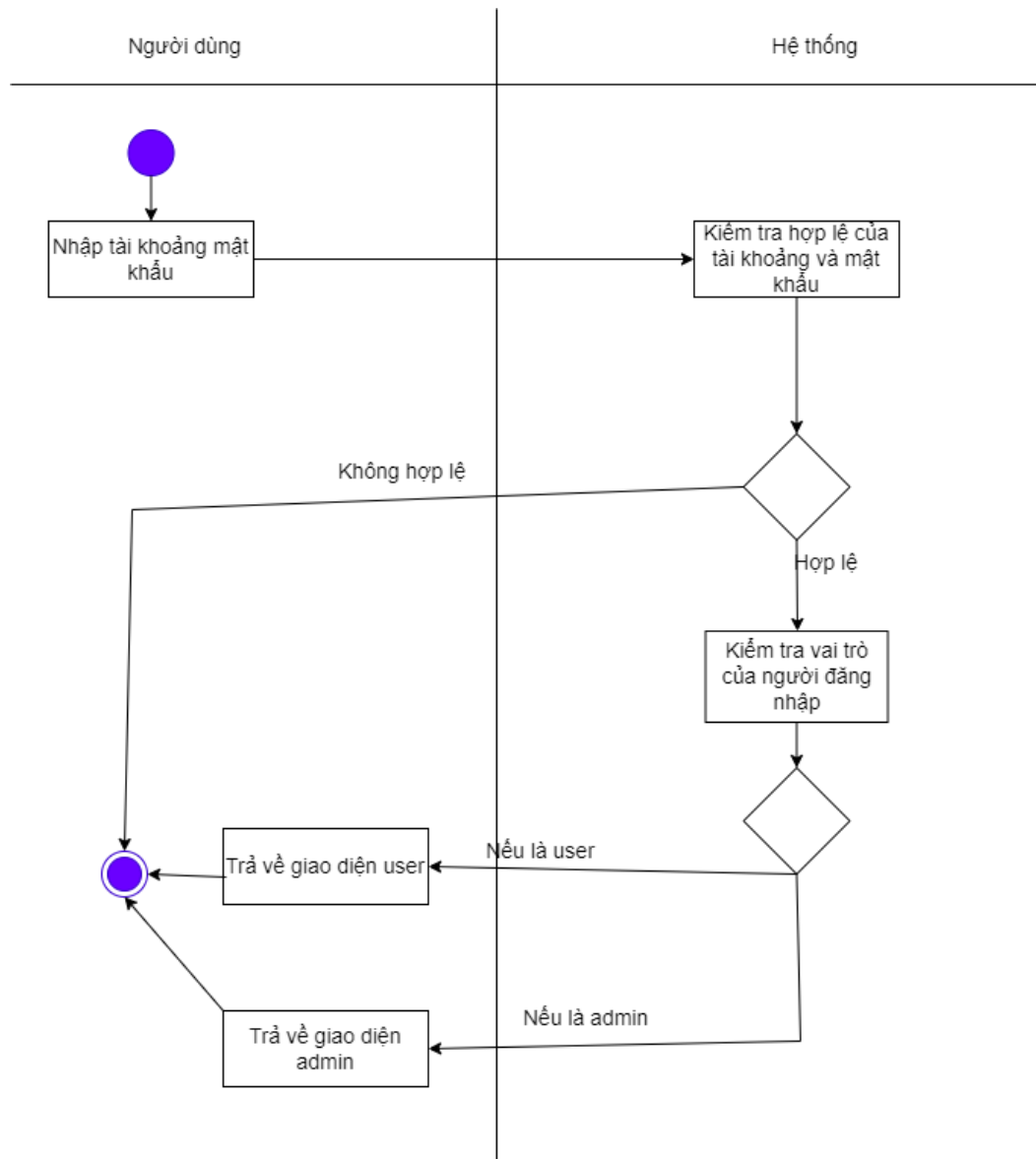
3.4 Sơ đồ hoạt động

- Quy trình đăng ký:



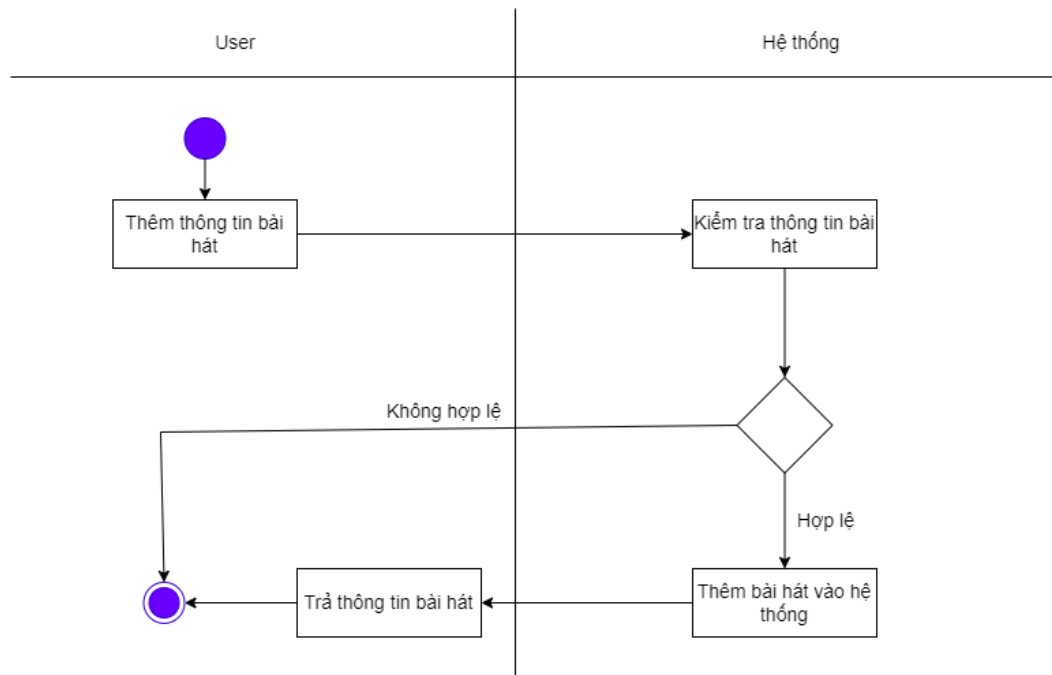
Hình 3. 4: Sơ đồ hoạt động đăng ký

- Quy trình đăng nhập:



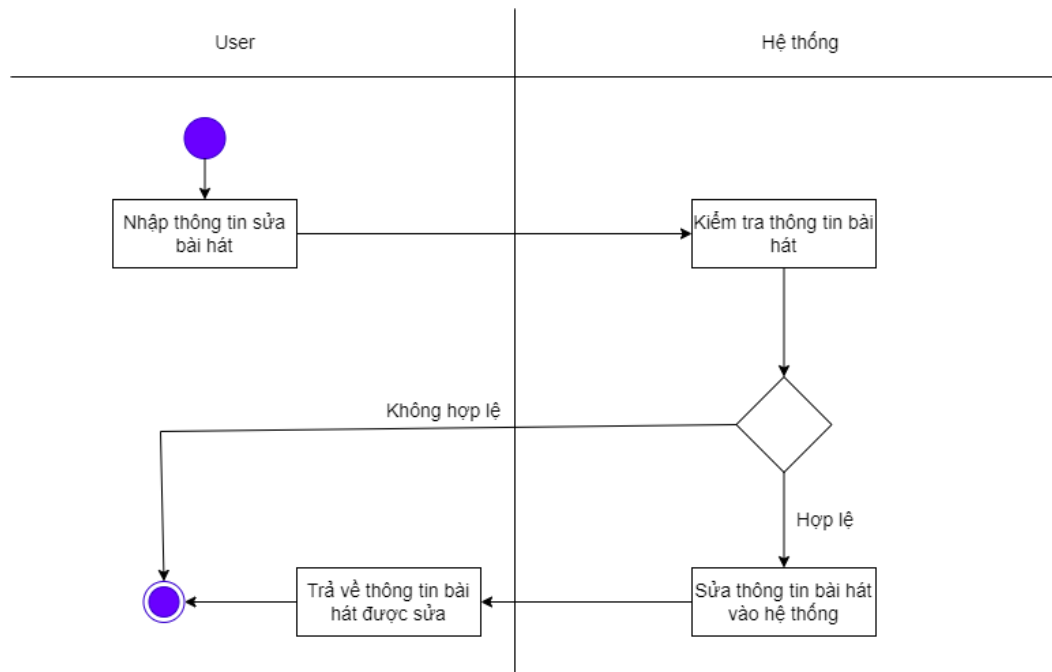
Hình 3. 5: Sơ đồ hoạt động đăng nhập

- Quy trình thêm bài hát:



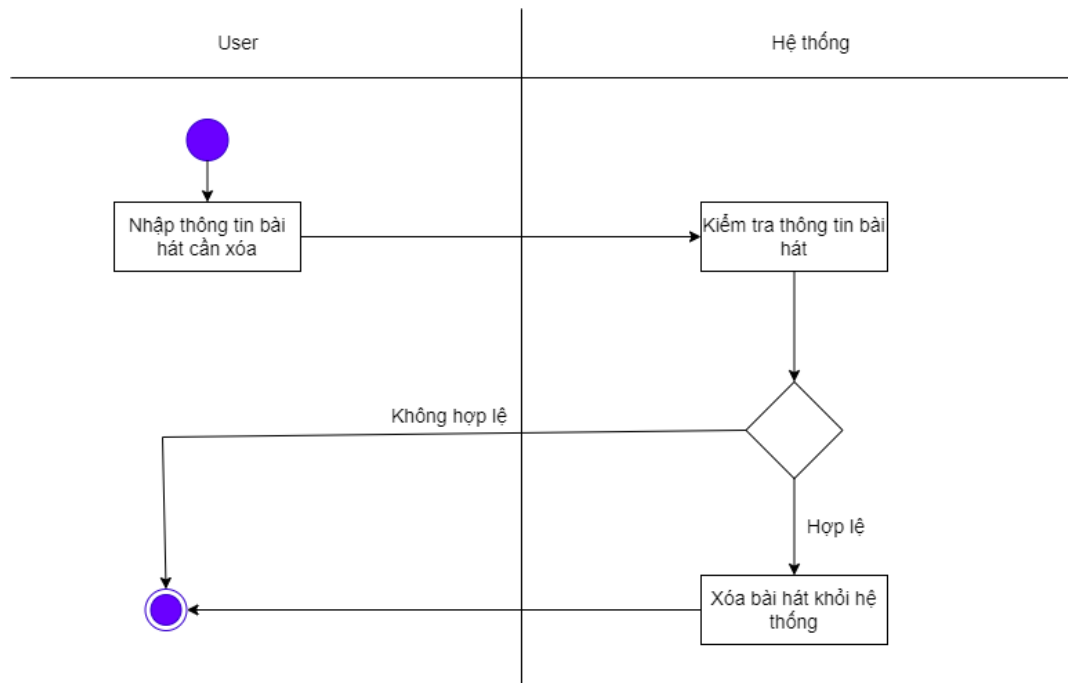
Hình 3. 6: Sơ đồ hoạt động thêm bài hát

- Quy trình sửa bài hát:



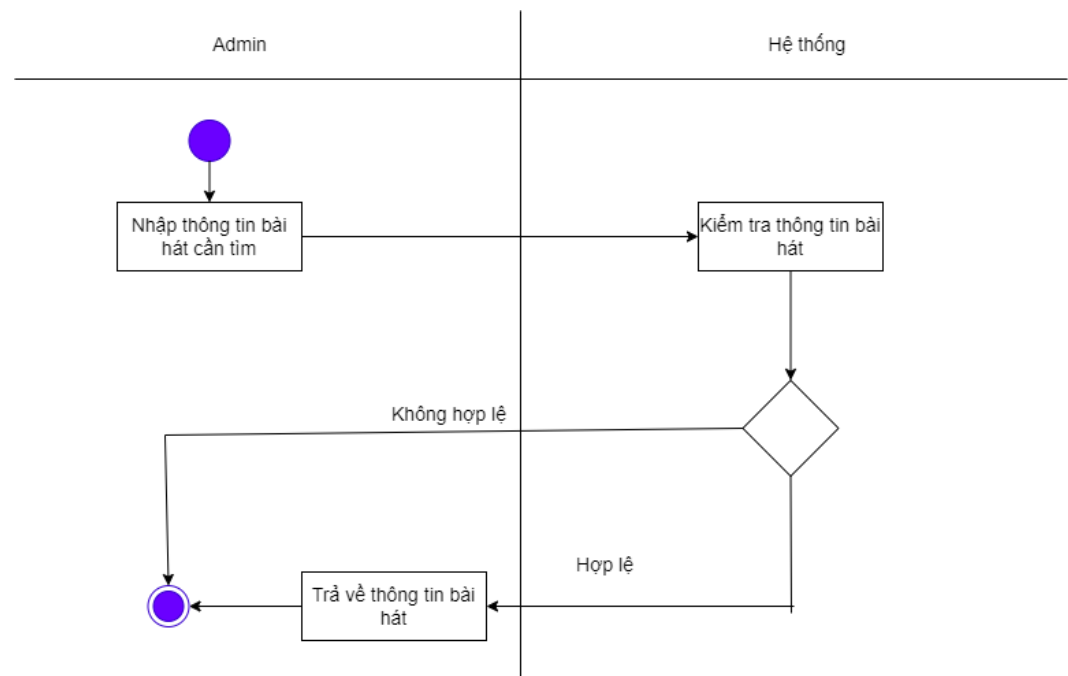
Hình 3. 7: Sơ đồ hoạt động sửa bài hát

- Quy trình xóa bài hát:



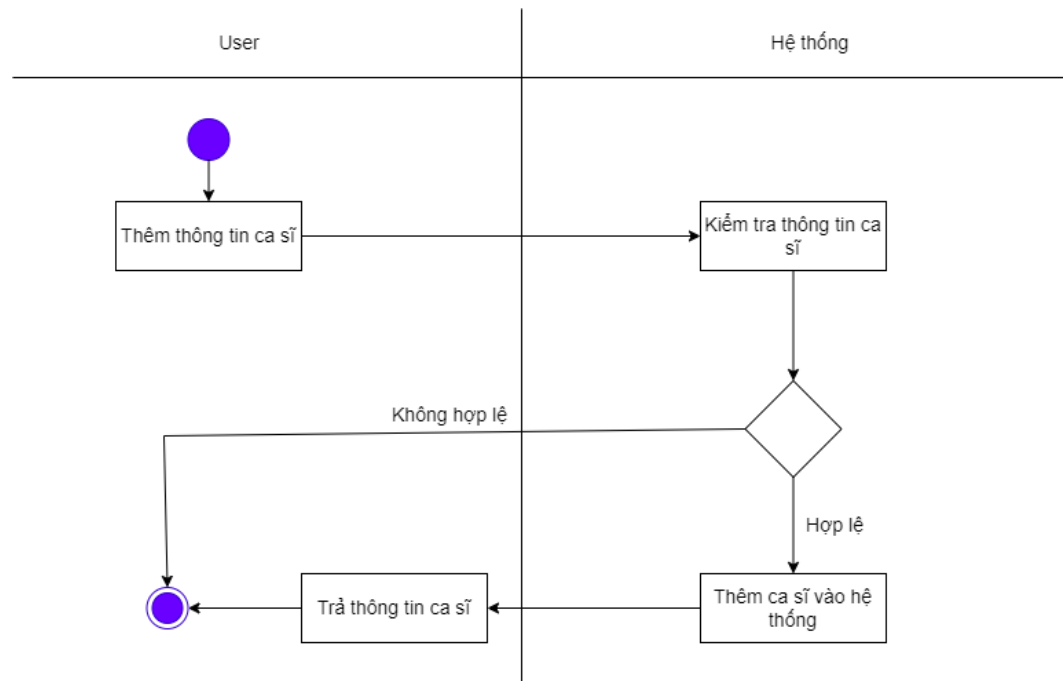
Hình 3. 8: Sơ đồ hoạt động xóa bài hát

- Quy trình tìm kiếm bài hát:



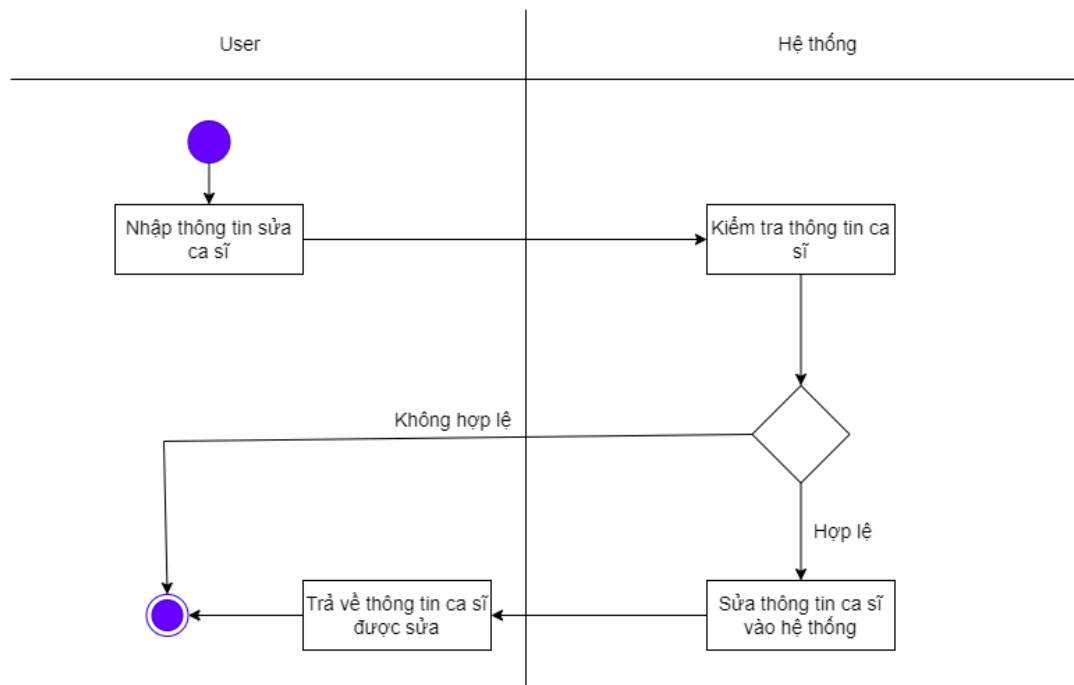
Hình 3. 9: Sơ đồ hoạt động tìm kiếm bài hát

-Quy trình thêm ca sĩ:



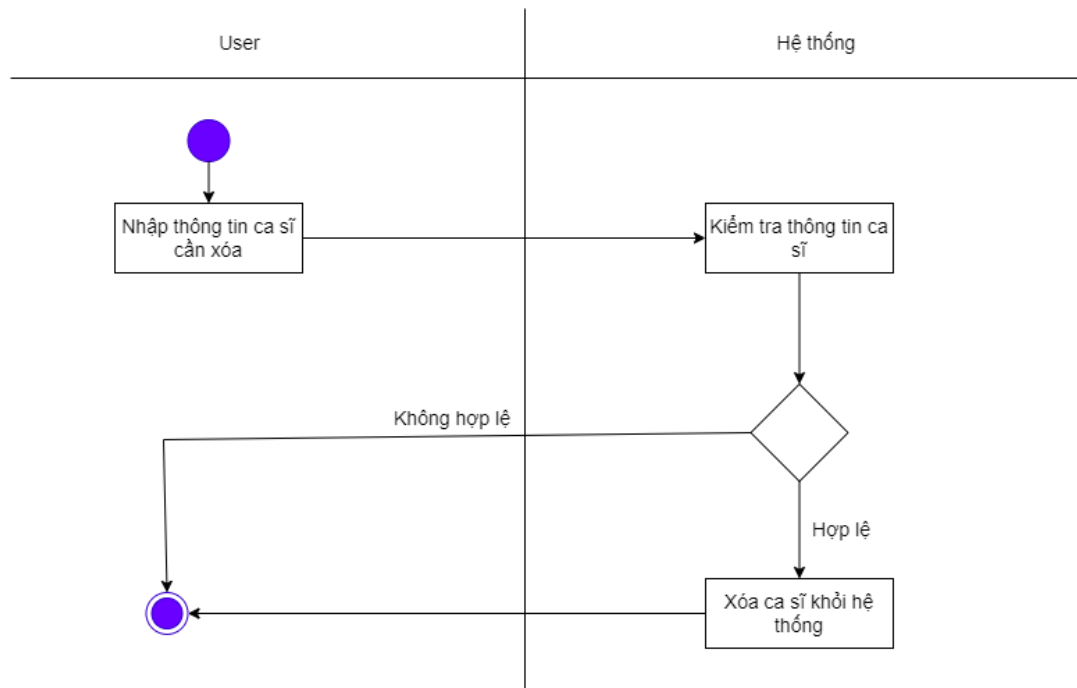
Hình 3. 10: Sơ đồ hoạt động thêm ca sĩ

- Quy trình sửa ca sĩ:



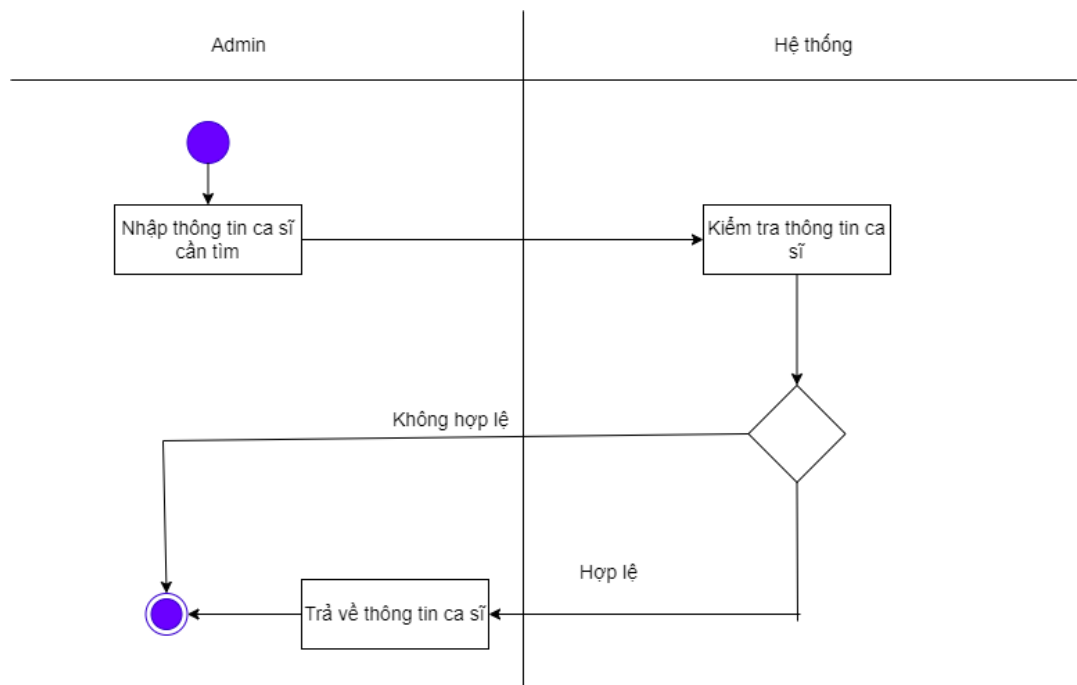
Hình 3. 11: Sơ đồ hoạt động sửa ca sĩ

- Quy trình xóa ca sĩ:



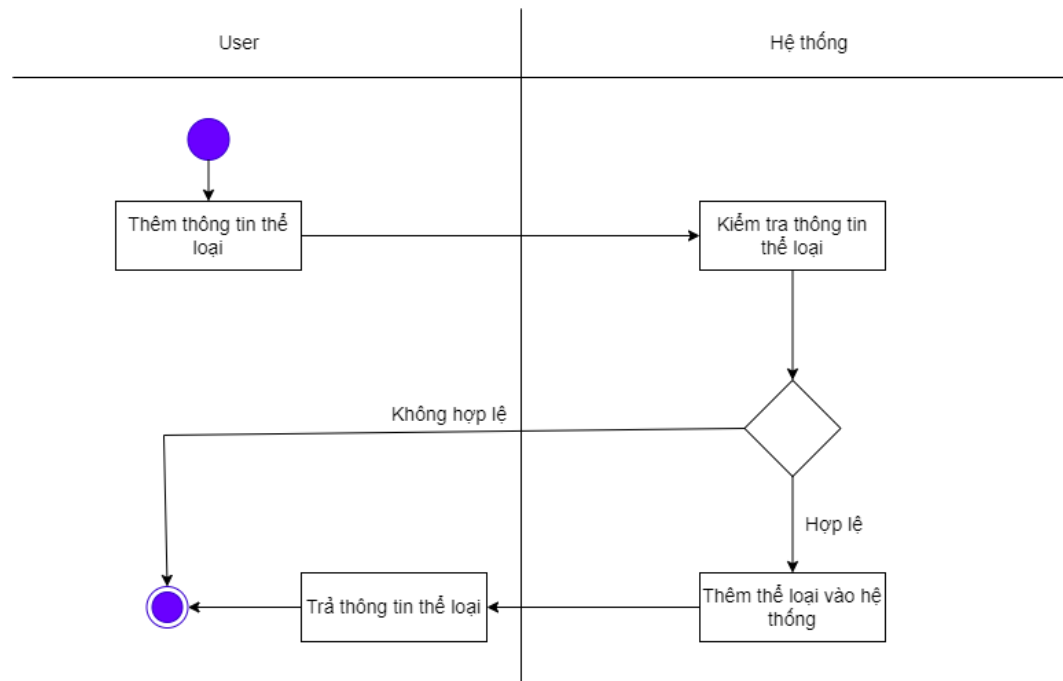
Hình 3. 12: Sơ đồ hoạt động xóa ca sĩ

- Quy trình tìm ca sĩ:



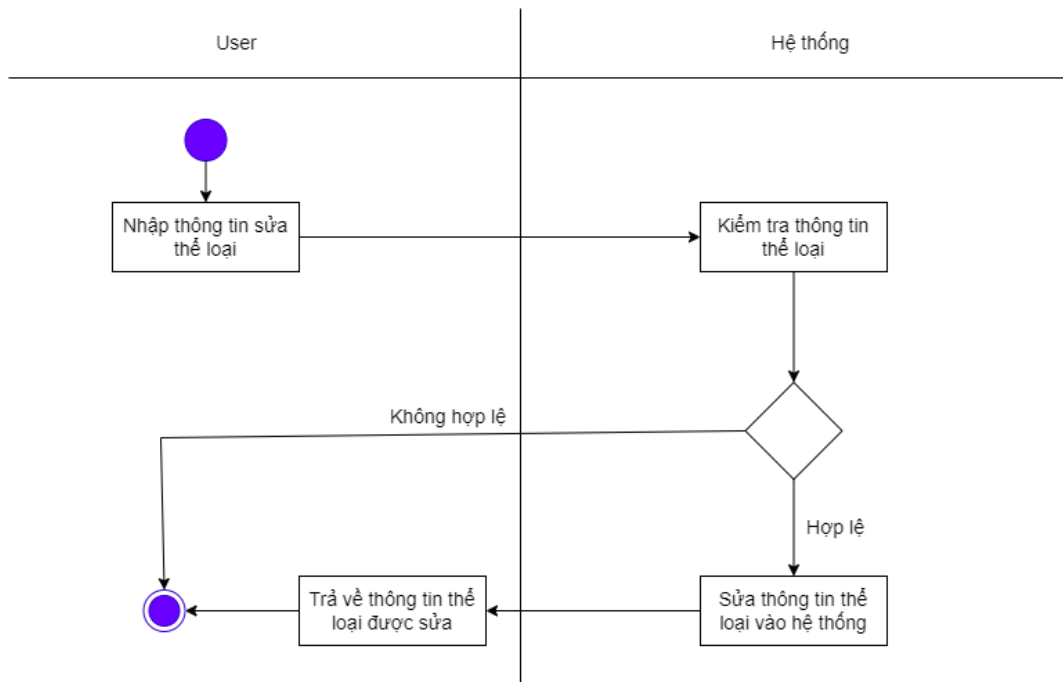
Hình 3. 13: Sơ đồ hoạt động tìm ca sĩ

- Quy trình thêm thẻ loại:



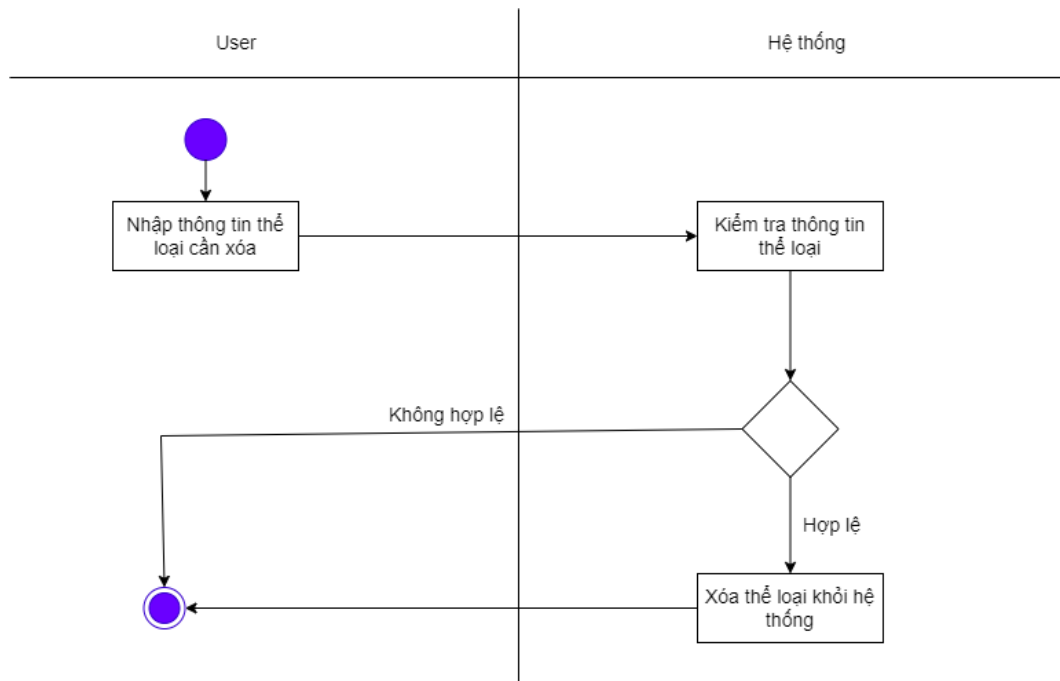
Hình 3. 14: Sơ đồ hoạt động thêm thẻ loại

-Quy trình sửa thẻ loại:



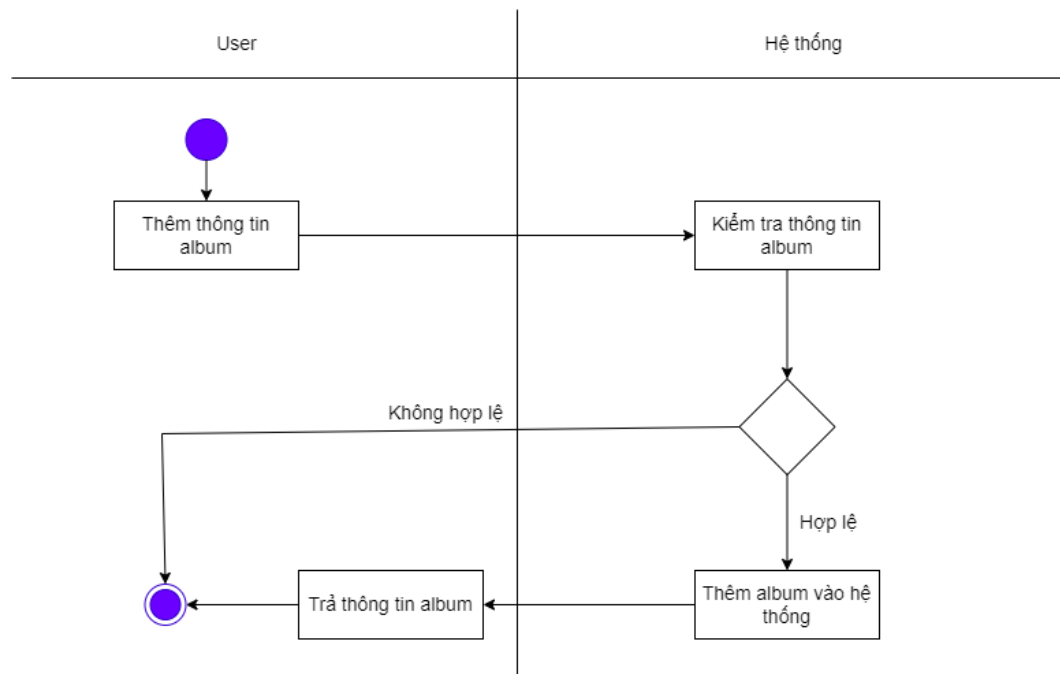
Hình 3. 15: Sơ đồ hoạt động sửa thẻ loại

- Quy trình xóa thể loại:



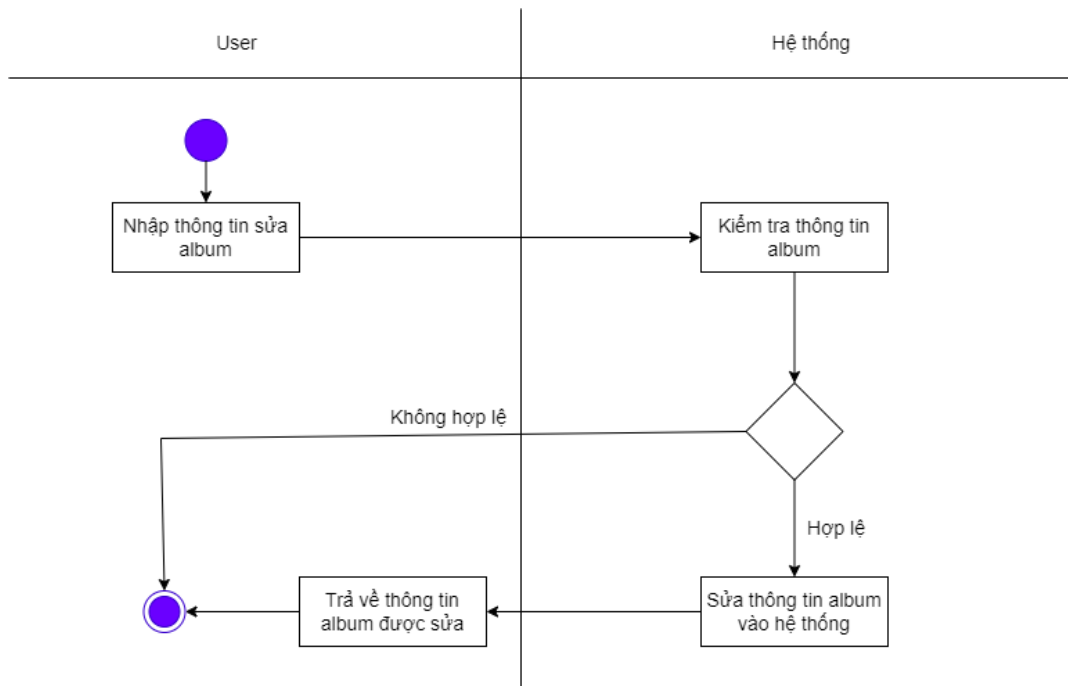
Hình 3. 16: Sơ đồ hoạt động xóa thể loại

- Quy trình thêm album:



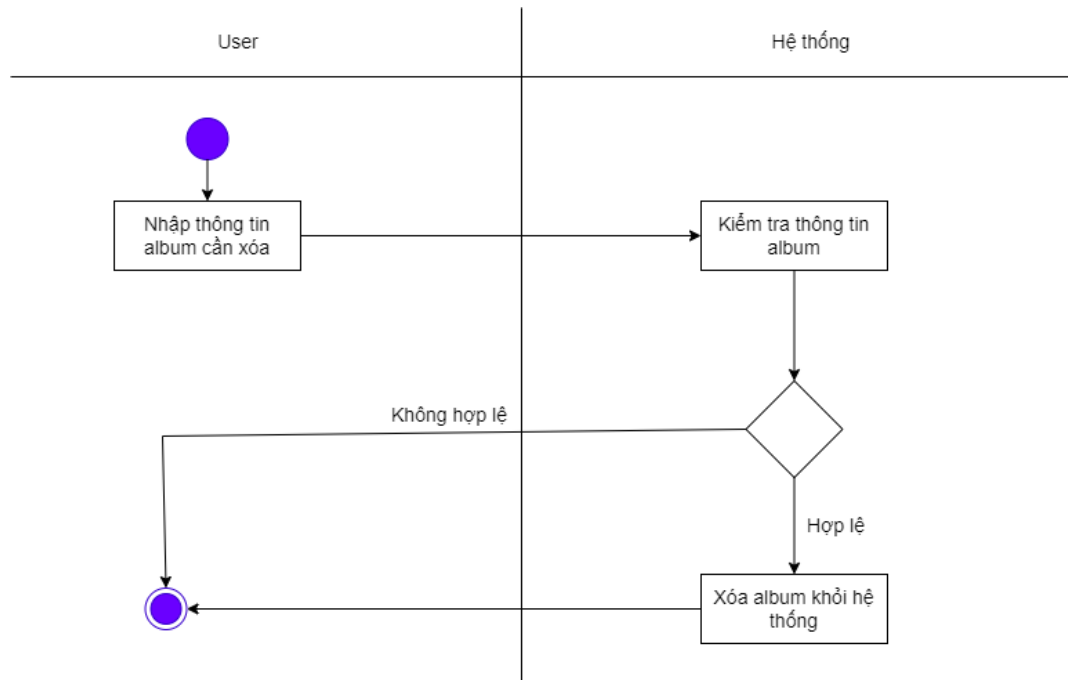
Hình 3. 17: Sơ đồ hoạt động thêm album

- Quy trình sửa album:



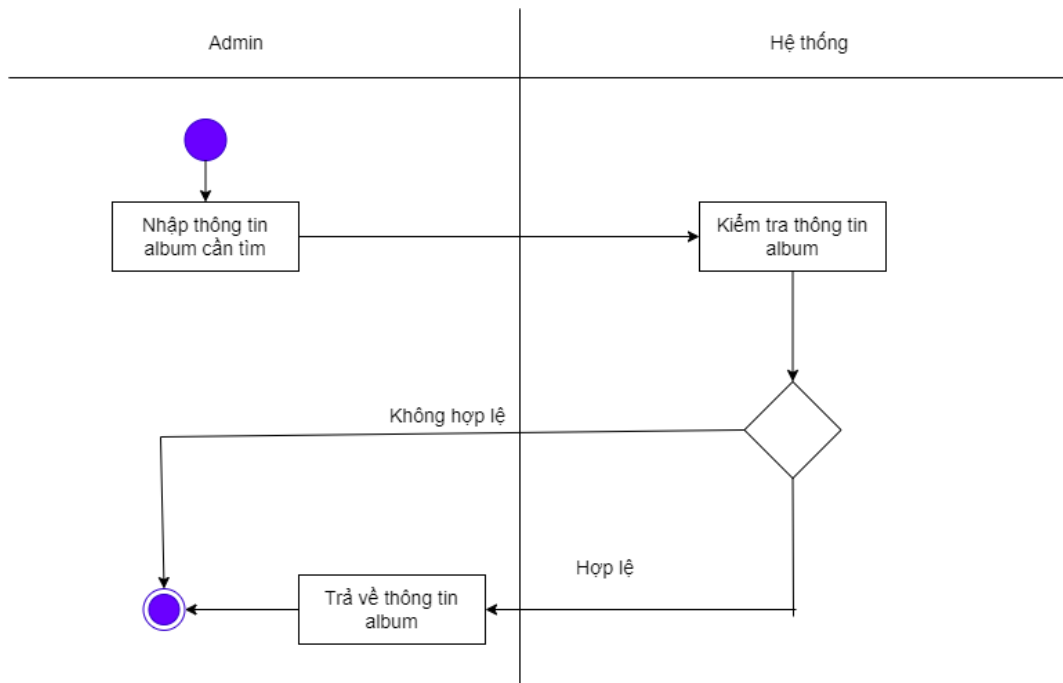
Hình 3. 18: Sơ đồ hoạt động sửa album

- Quy trình xóa album:



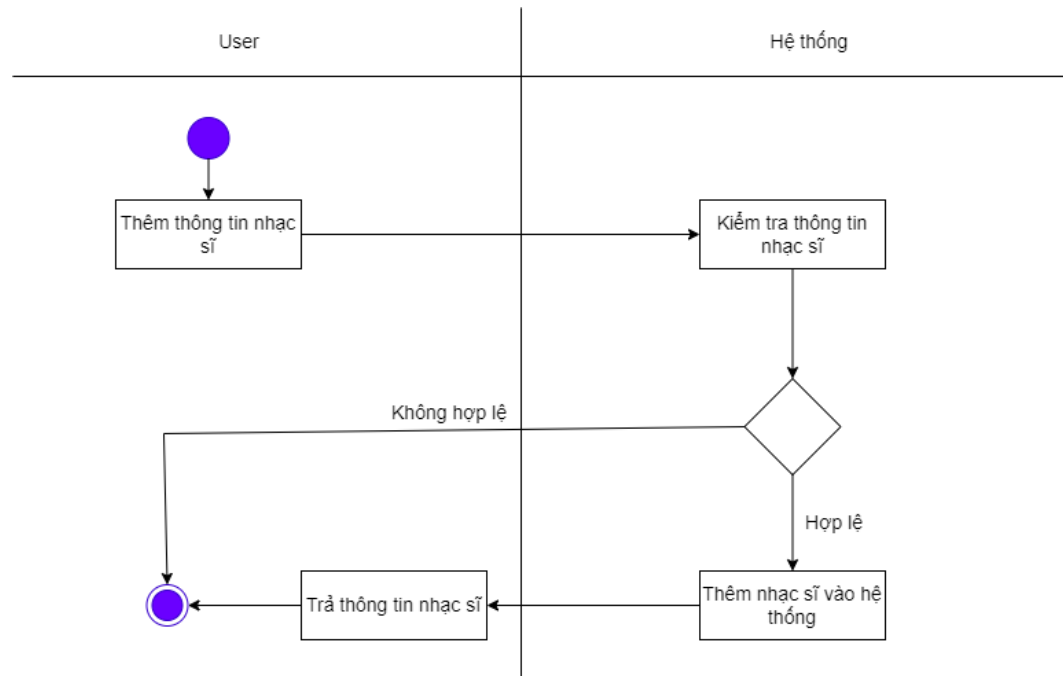
Hình 3. 19: Sơ đồ hoạt động xóa album

- Quy trình tìm album:



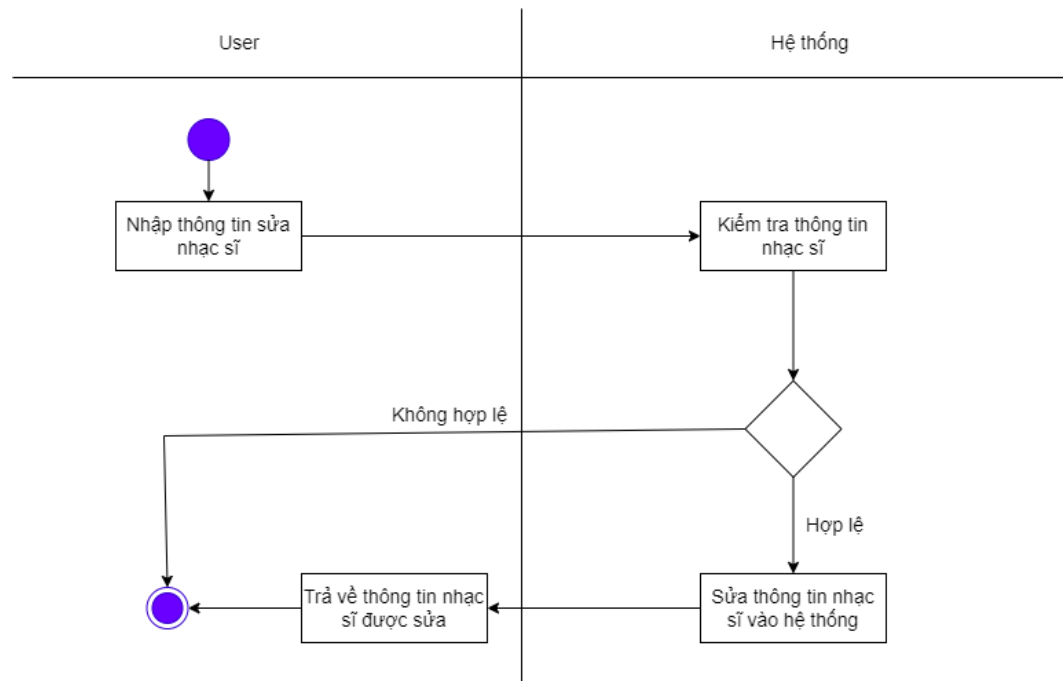
Hình 3. 20: Sơ đồ hoạt động tìm album

- Quy trình thêm nhạc sĩ:



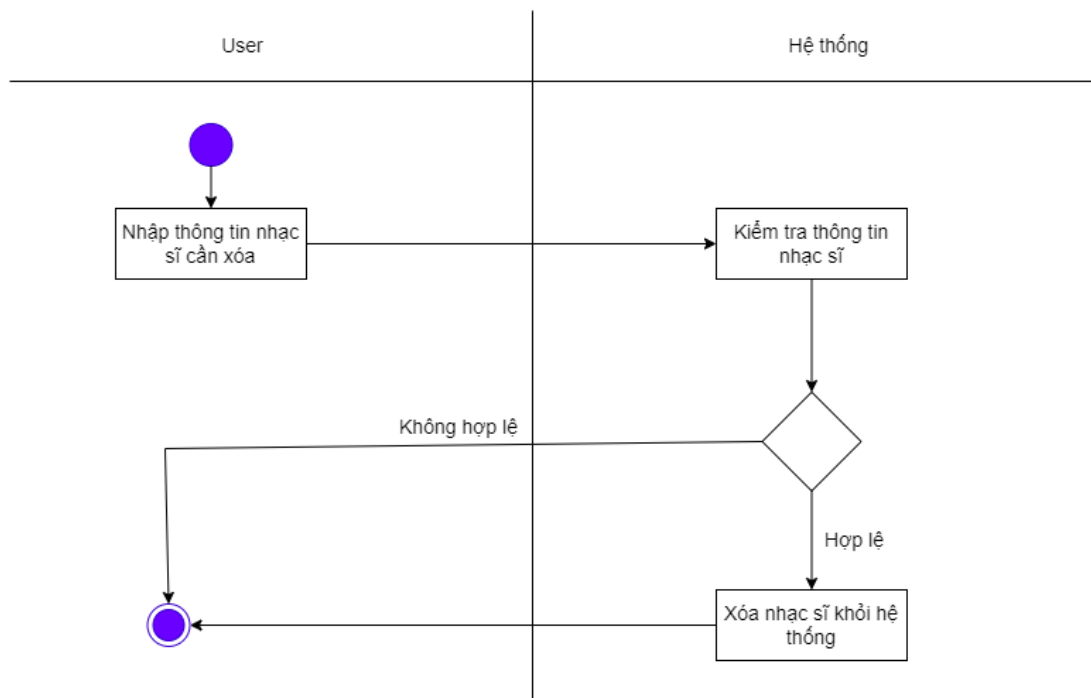
Hình 3. 21: Sơ đồ hoạt động thêm ca sĩ

- Quy trình sửa nhạc sĩ:



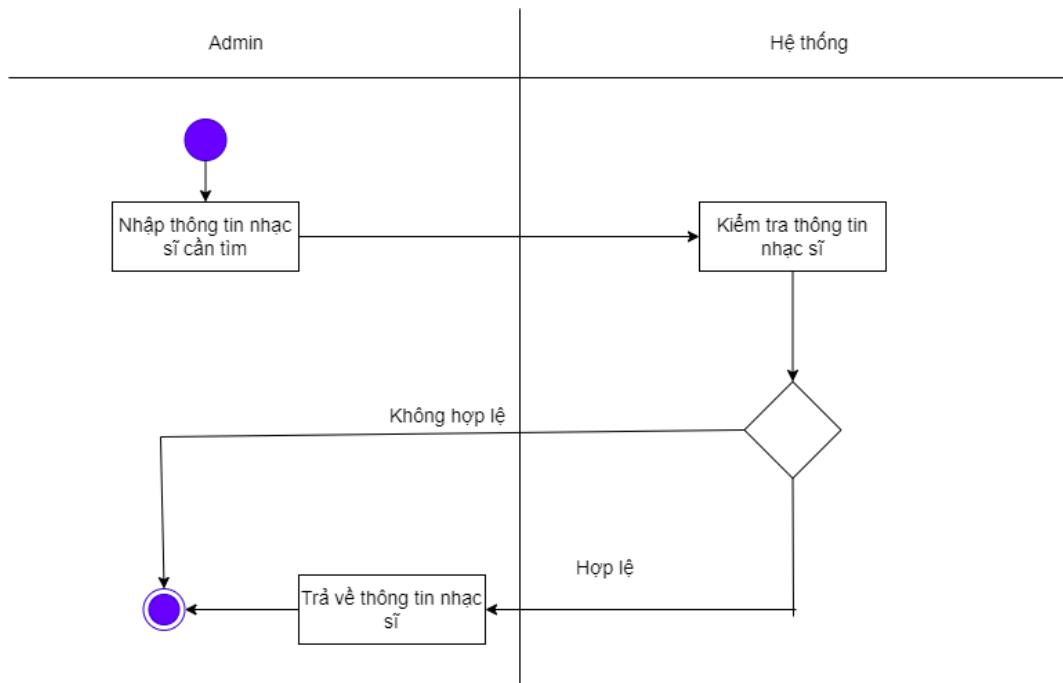
Hình 3. 22: Sơ đồ hoạt động sửa ca sĩ

-Quy trình xóa nhạc sĩ:



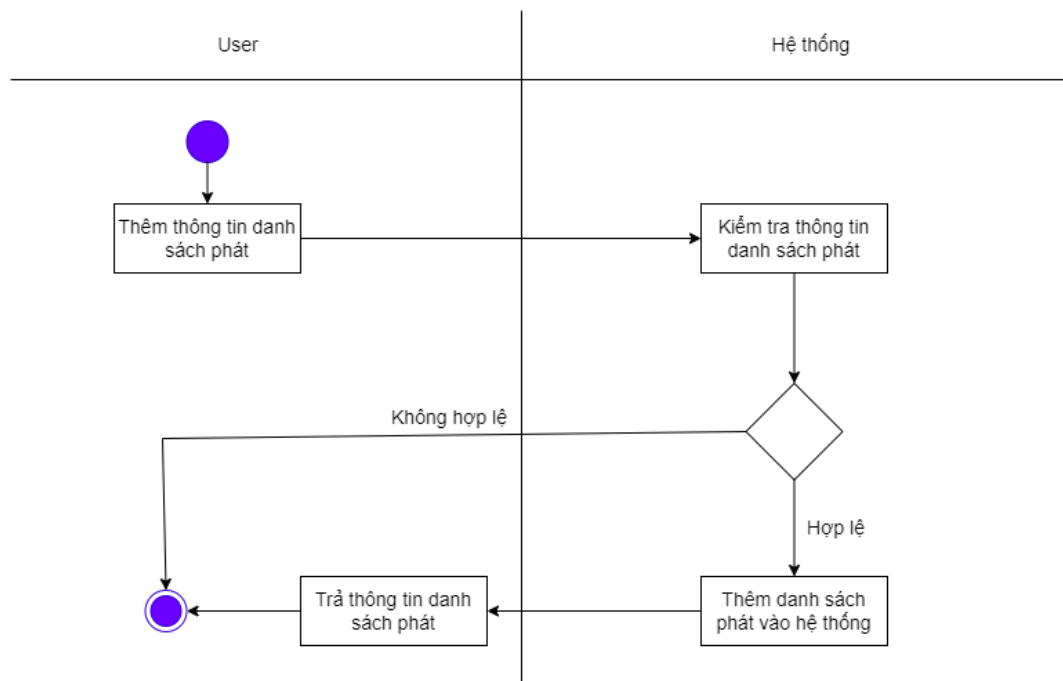
Hình 3. 23: Sơ đồ hoạt động xóa ca sĩ

- Quy trình tìm nhạc sĩ:



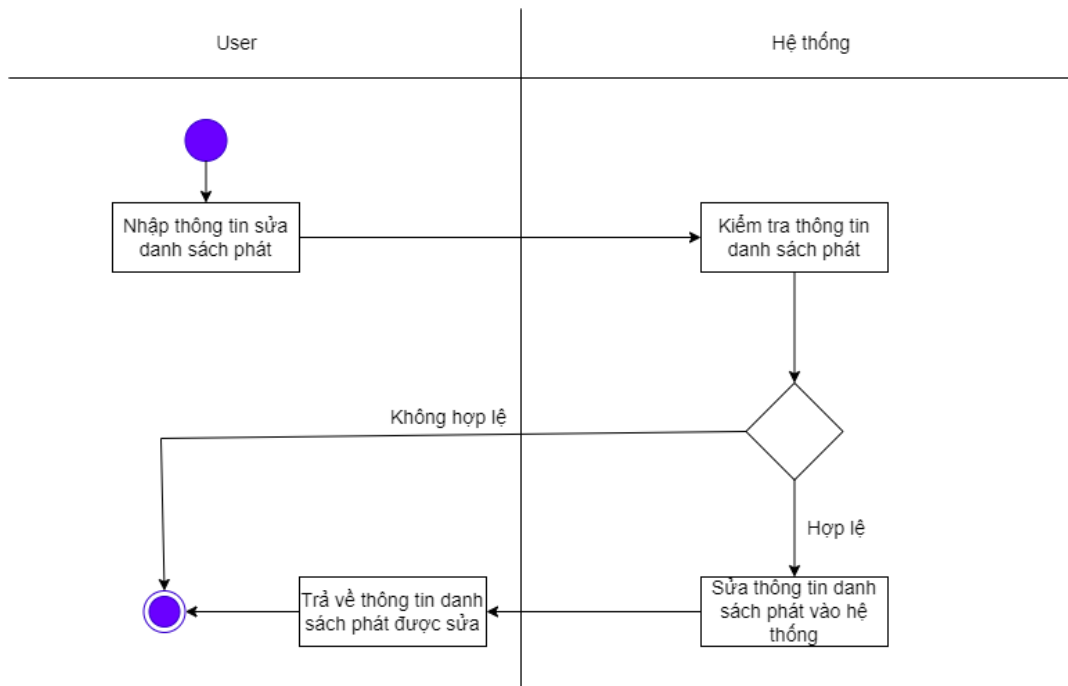
Hình 3. 24: Sơ đồ hoạt động tìm nhạc sĩ

- Quy trình thêm danh sách phát của user:



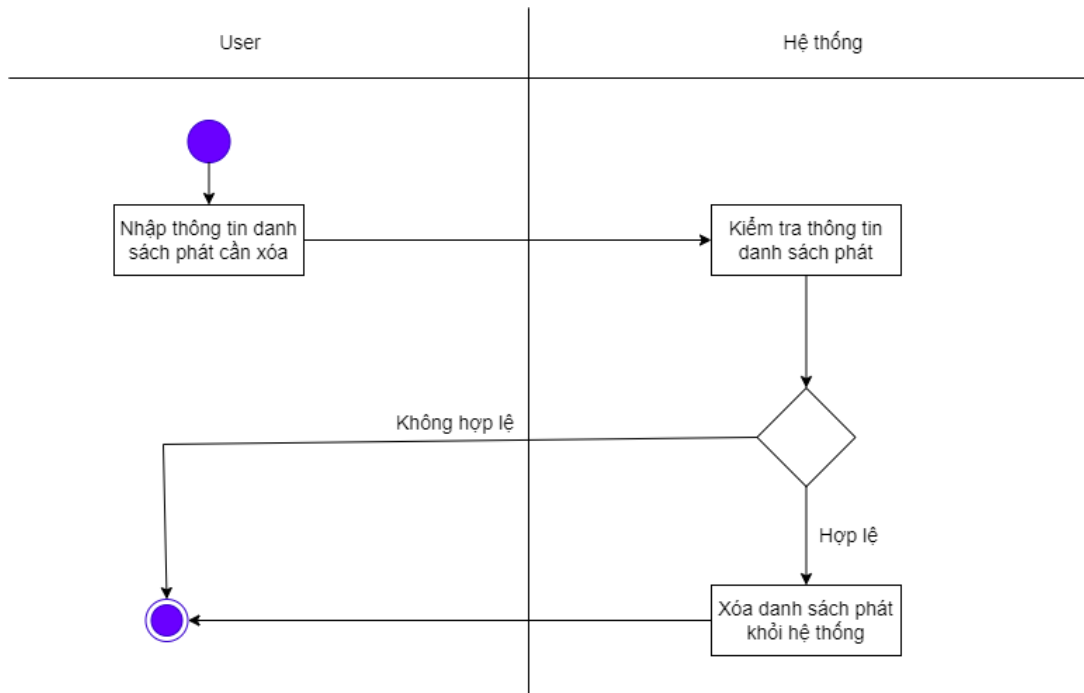
Hình 3. 25: Sơ đồ hoạt động thêm danh sách phát của user

- Quy trình sửa danh sách phát của user:



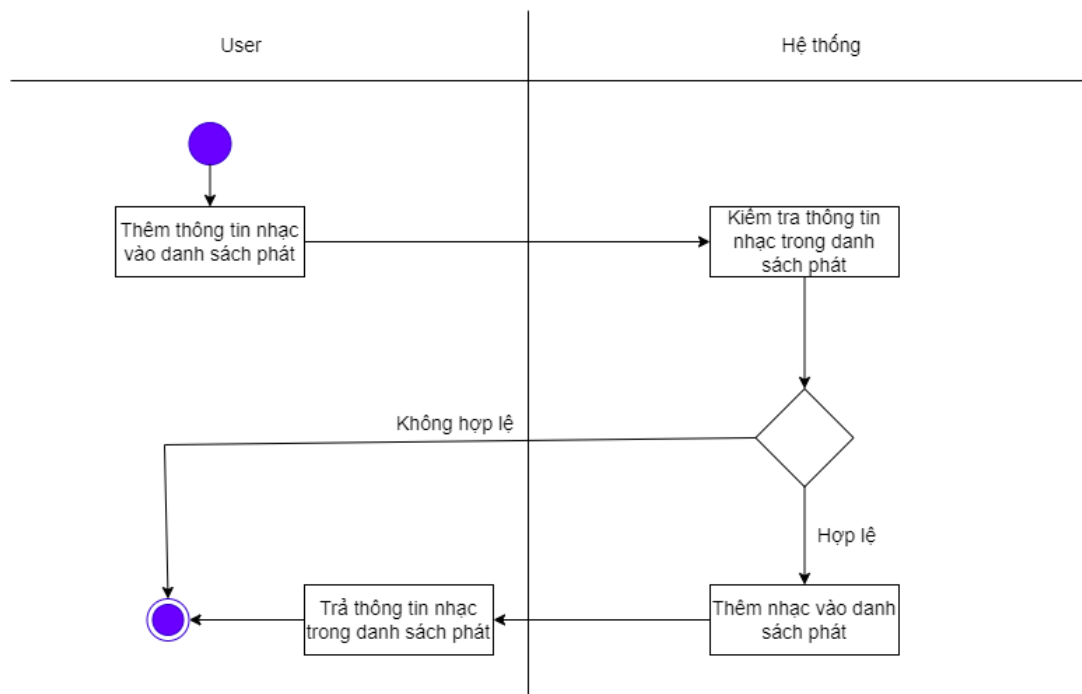
Hình 3. 26: Sơ đồ hoạt động sửa danh sách phát của user

-Quy trình xóa danh sách phát của user:



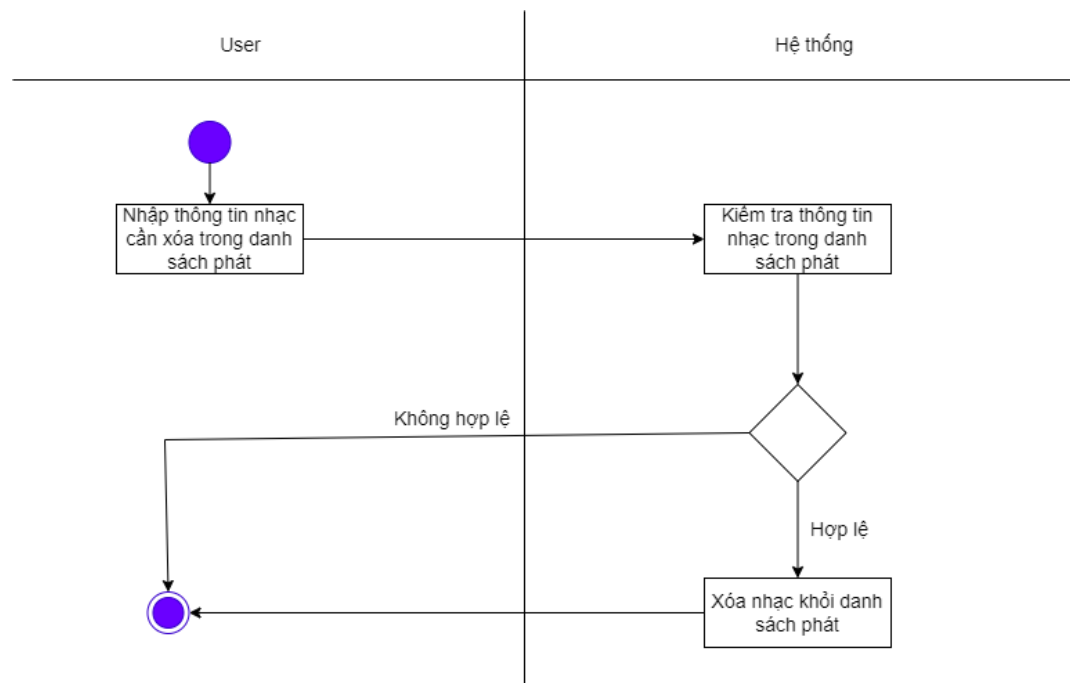
Hình 3. 27: Sơ đồ hoạt động xóa danh sách phát của user

-Quy trình thêm nhạc vào danh sách phát của user:



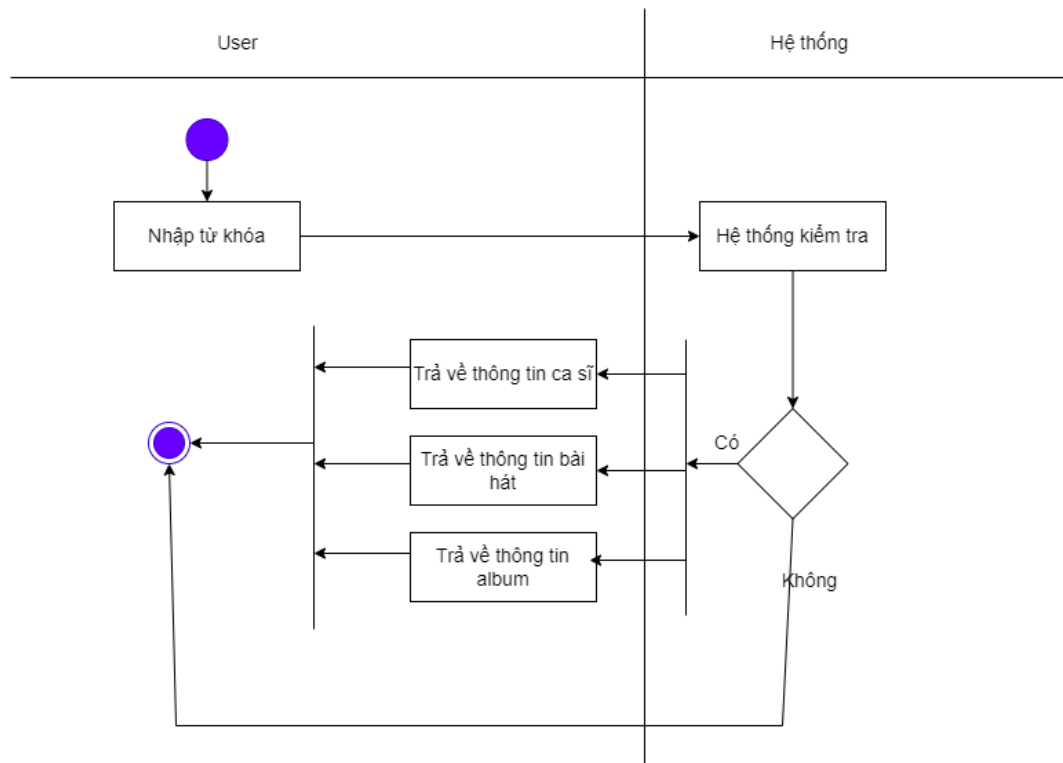
Hình 3. 28: Sơ đồ hoạt động thêm nhạc vào danh sách phát của user

-Quy trình xóa nhạc khỏi danh sách phát của user:



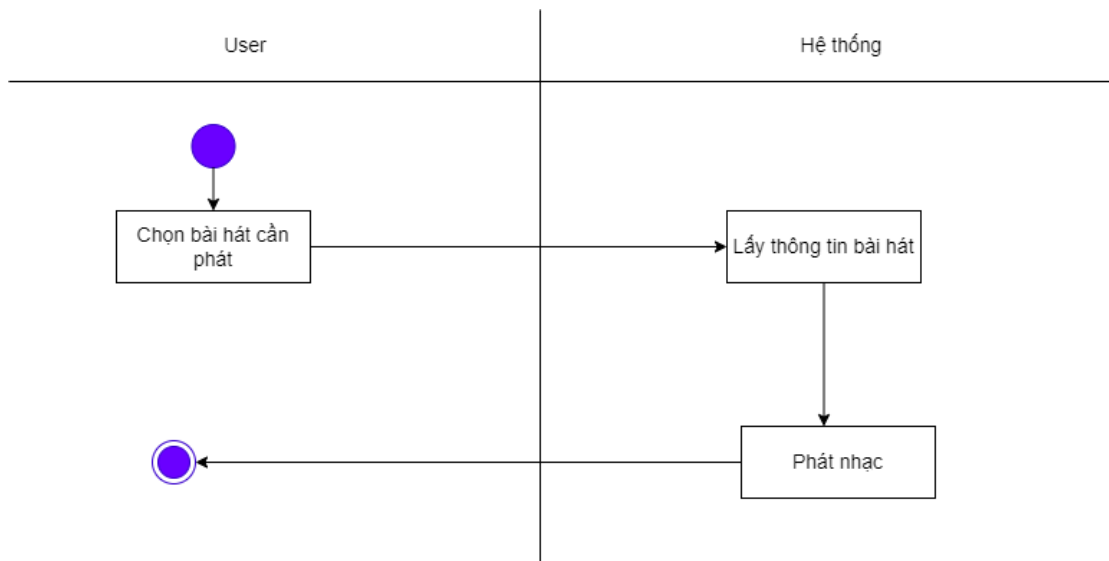
Hình 3. 29: Sơ đồ hoạt động xóa nhạc khỏi danh sách phát của user

- Quy trình tìm nhạc, album, ca sĩ của user:



Hình 3. 30: Sơ đồ hoạt động tìm kiếm của user

- Quy trình nghe nhạc của user:



Hình 3. 31: Sơ đồ hoạt động nghe nhạc của user

CHƯƠNG 4. THIẾT KẾ CHƯƠNG TRÌNH

4.1 Thiết kế cơ sở dữ liệu

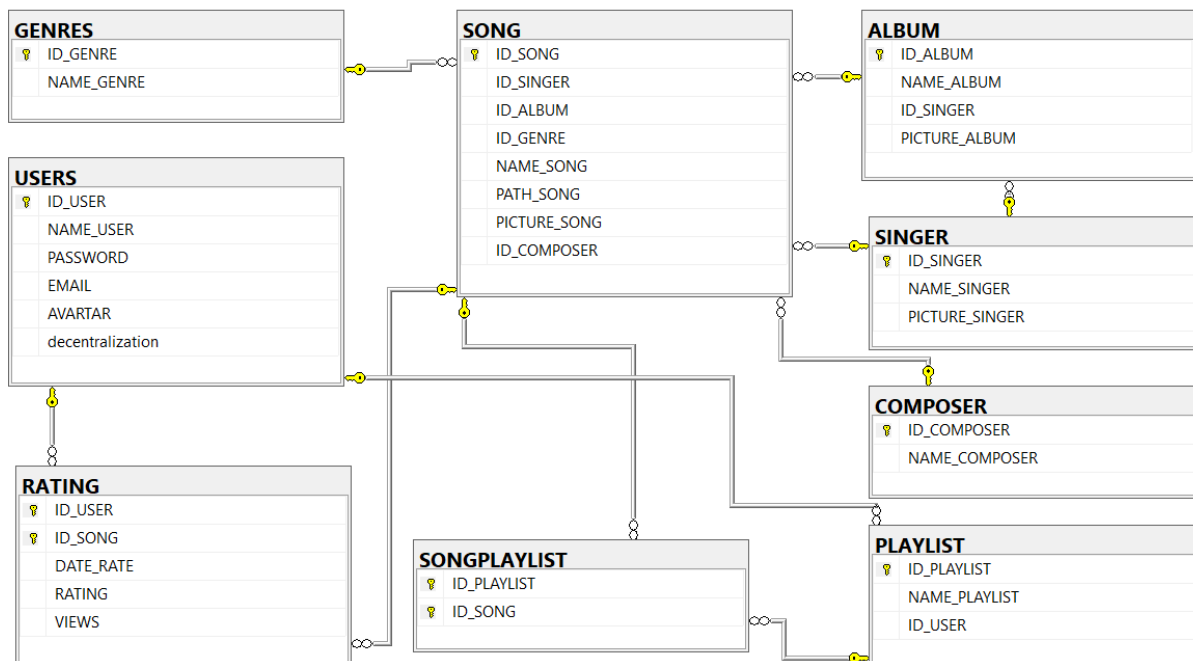
4.1.1 Mô hình quan hệ thực thể (mô hình ER)



Hình 4. 1: Mô hình ER

4.1.2 Biểu đồ các bảng trong cơ sở dữ liệu

Từ dữ liệu mô hình thực thể quan hệ ER, xây dựng các bảng với khóa chính và khóa ngoại



Hình 4. 2: Sơ đồ cơ sở dữ liệu

4.2 Thiết kế hệ thống gợi ý

4.2.1 Ý tưởng:

Gợi ý content-based filtering:

+ Trong bài toán này, hệ thống gợi ý dựa trên nội dung được xây dựng dựa vào ba thuộc tính của bài hát (item) là ca sĩ, thể loại, tác giả. Xây dựng ma trận đề tính toán. Mỗi cột lần lượt là giá trị của thể loại, ca sĩ, tác giả. Mỗi hàng là biểu diễn của mỗi bài hát. Sử dụng phương pháp tạo biểu diễn nhị phân. Giá trị ở các cột của mỗi hàng bằng 1 nếu bài hát có giá trị một trong ba thuộc tính (ca sĩ, thể loại, tác giả), bằng 0 nếu không có giá trị. Tại mỗi cột của mỗi hàng có giá trị bằng 1, thực hiện chuẩn hóa bằng công thức $1/\sqrt{3}$. Sau khi chuẩn hóa, các hàng là vector đặc trưng

của từng bài hát (item). Thực hiện tính độ tương đồng giữa các bài hát (cụ thể trong bài toán này là bài hát đầu tiên với các bài hát còn lại) bằng công thức cosin

$$\text{Độ tương đồng cosin} = \text{tích vô hướng} / \text{tích độ dài}$$

+ Giá trị cosin càng lớn thì độ tương đồng hai bài hát càng lớn. Ngược lại, giá trị càng nhỏ thì hai bài hát không tương đồng.

+ Hệ thống ghi nhớ các bài hát mà người dùng truy cập và tính độ ưa thích của người dùng với bài hát dựa vào thời gian nghe một bài hát của người đó. Ví dụ người dùng nghe bài hát ‘xuân thì’ thì mức độ ưa thích được tính bằng công thức:

$$\text{Độ ưa thích} = \text{thời lượng nghe bài hát} / \text{thời lượng bài hát}$$

+ Từ giá trị ưa thích của người dùng, có thể tìm được vector ưa thích của người dùng. Giả sử người dùng thích n bài hát, bài hát i với mức ưa thích là a, thì độ ưa thích của thuộc tính w_i được tính bằng $w_i = a_i * \text{giá trị chuẩn hóa của thuộc tính đó}$, tương tự có n bài hát thì thực hiện cộng w_i của n bài hát sẽ có được độ ưa thích của thuộc tính w_n . Các thuộc tính còn lại được tính tương tự. Cuối cùng thu được vector độ ưa thích của user.

- Gợi ý collaborative filtering: Sử dụng thư viện ML.Net để train mô hình, dữ liệu đưa vào là ID_USER, ID_SONG, RATING. Dữ liệu lấy gia là các giá trị RATING được dự đoán bởi mô hình.

4.2.2 Kết quả thực nghiệm chức năng gợi ý content-based filtering

- Tiến hành thực nghiệm 5 bài hát:

+ Mưa hồng (id: 143, ca sĩ: Lê Uyên, thể loại: nhạc Trịnh, tác giả: Trịnh Công Sơn), người dùng đã nghe và có độ yêu thích là 4.

+ Kiếp tha hương (id: 138, ca sĩ: Khánh Ly, thể loại: nhạc Trịnh, tác giả: Trịnh Công Sơn), người dùng đã nghe và có độ yêu thích là 5.

+ Phôi pha (id: 144, ca sĩ: Tuấn ngọc, thể loại: nhạc Trịnh, tác giả: Trịnh Công Sơn) , người dùng đã nghe và có độ yêu thích là 4.

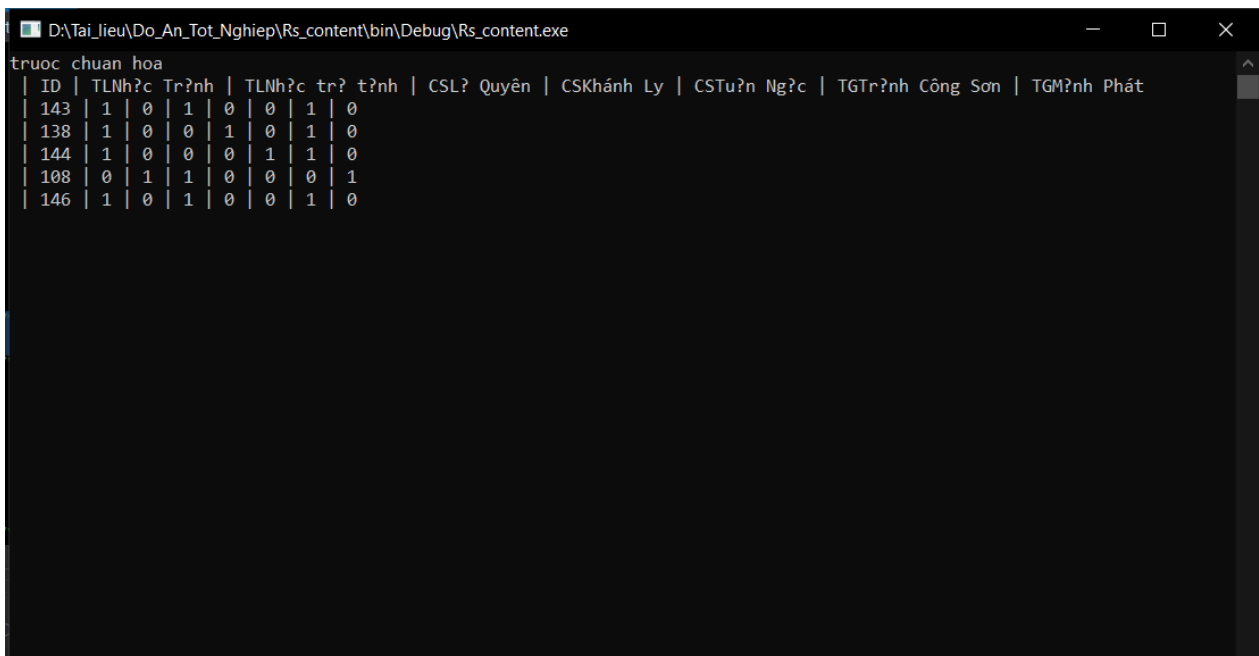
+ Hoa nở về đêm (id: 108, ca sĩ: Lệ Quyên, thể loại: nhạc trữ tình, tác giả: Mạnh phát), người dùng chưa nghe.

+ Sóng về đâu (id: 146, ca sĩ: Lệ Quyên, thể loại: nhạc Trịnh, tác giả: Trịnh Công Sơn), người dùng chưa nghe.

Gợi ý các bài hát tương tự

- Bài toán: Tính độ tương đồng của bài hát “Mưa hồng” với các bài hát còn lại.

+ Biểu diễn các bài hát vào ma trận với các cột lần lượt là tên ca sĩ, tên thể loại, tên tác giả, các hàng lần lượt là các bài hát thu được ma trận:



ID	TLNh?c Tr?nh	TLNh?c tr? t?nh	CSL? Quyên	CSKhánh Ly	CSTu?n Ng?c	TGTr?nh Công Sơn	TGM?nh Phát
143	1	0	1	0	0	1	0
138	1	0	0	1	0	1	0
144	1	0	0	0	1	1	0
108	0	1	1	0	0	0	1
146	1	0	1	0	0	1	0

Hình 4. 3: Kết quả biểu diễn bài hát vào ma trận

+ Tiếp theo, thực hiện chuẩn hoá và thu được ma trận chứa các dòng là các vector bài hát:

```

D:\Tai_lieu\Do_An_Tot_Nghiep\Rs_content\bin\Debug\Rs_content.exe
trước chuẩn hóa
| ID | TLNh?c | Tr?nh | TLNh?c | tr? | t?nh | CSL? | Quyên | CSKhánh Ly | CSTu?n | Ng?c | TGTr?nh | Công Sơn | TGM?nh | Phát
| 143 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0
| 138 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0
| 144 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0
| 108 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1
| 146 | 1 | 0 | 1 | 0 | 0 | 1 | 0
sau chuẩn hóa
| 143 | 0.577350269189626 | 0 | 0.577350269189626 | 0 | 0 | 0.577350269189626 | 0
| 138 | 0.577350269189626 | 0 | 0 | 0.577350269189626 | 0 | 0.577350269189626 | 0
| 144 | 0.577350269189626 | 0 | 0 | 0 | 0.577350269189626 | 0.577350269189626 | 0
| 108 | 0 | 0.577350269189626 | 0.577350269189626 | 0 | 0 | 0 | 0.577350269189626
| 146 | 0.577350269189626 | 0 | 0.577350269189626 | 0 | 0 | 0.577350269189626 | 0

```

Hình 4. 4: Kết quả biểu diễn bài hát sau khi chuẩn hóa

+ Cuối cùng là tính độ tương đồng của các bài hát:

```

D:\Tai_lieu\Do_An_Tot_Nghiep\Rs_content\bin\Debug\Rs_content.exe
trước chuẩn hóa
| ID | TLNh?c | Tr?nh | TLNh?c | tr? | t?nh | CSL? | Quyên | CSKhánh Ly | CSTu?n | Ng?c | TGTr?nh | Công Sơn | TGM?nh | Phát
| 143 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0
| 138 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0
| 144 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0
| 108 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1
| 146 | 1 | 0 | 1 | 0 | 0 | 1 | 0
sau chuẩn hóa
| 143 | 0.577350269189626 | 0 | 0.577350269189626 | 0 | 0 | 0.577350269189626 | 0
| 138 | 0.577350269189626 | 0 | 0 | 0.577350269189626 | 0 | 0.577350269189626 | 0
| 144 | 0.577350269189626 | 0 | 0 | 0 | 0.577350269189626 | 0.577350269189626 | 0
| 108 | 0 | 0.577350269189626 | 0.577350269189626 | 0 | 0 | 0 | 0.577350269189626
| 146 | 0.577350269189626 | 0 | 0.577350269189626 | 0 | 0 | 0.577350269189626 | 0
Tính cosin
do tương đồng bài 1 với bài 2 là: 0.666666666666667
do tương đồng bài 1 với bài 3 là: 0.666666666666667
do tương đồng bài 1 với bài 4 là: 0.333333333333333
do tương đồng bài 1 với bài 5 là: 1

```

Hình 4. 5: Kết quả độ tương tự cosin giữa các bài hát

- Gợi ý các bài hát dựa vào sở thích nghe của người dùng

+ Biểu diễn các bài hát mà người dùng thích vào ma trận và chuẩn hóa như phần gợi ý tương tự, sau đó tính vector đặc trưng của người dùng (User profile).

```

D:\Tai_lieu\Do_An_Tot_Nghiep\Rs_content\bin\Debug\Rs_content.exe
trước chuẩn hóa
| ID | TLNh?c tr? t?nh | TLNh?c Tr?nh | CSKhánh Ly | CSL? Quyên | CSTu?n Ng?c | TGTr?nh Công Sơn | TGM?nh Phát
| 138 | 0 | 1 | 1 | 0 | 0 | 1 | 0
| 143 | 0 | 1 | 0 | 1 | 0 | 1 | 0
| 144 | 0 | 1 | 0 | 0 | 1 | 1 | 0
sau chuẩn hóa
| 138 | 0 | 0.577350269189626 | 0.577350269189626 | 0 | 0 | 0.577350269189626 | 0
| 143 | 0 | 0.577350269189626 | 0 | 0.577350269189626 | 0 | 0.577350269189626 | 0
| 144 | 0 | 0.577350269189626 | 0 | 0 | 0.577350269189626 | 0.577350269189626 | 0
User profile
| 0 | 0 | 7.50555349946513 | 2.88675134594813 | 2.3094010767585 | 2.3094010767585 | 7.50555349946513 | 0

```

Hình 4. 6: Kết quả vector đặc trưng của người dùng

+ Tiếp theo, đưa các bài hát mà người dùng chưa nghe vào ma trận vào chuẩn hóa, sau đó tính toán kết hợp với vector người dùng, cuối cùng dự đoán được độ yêu thích của người dùng đối với bài hát chưa nghe.

```

D:\Tai_lieu\Do_An_Tot_Nghiep\Rs_content\bin\Debug\Rs_content.exe
trước chuẩn hóa
| ID | TLNh?c tr? t?nh | TLNh?c Tr?nh | CSKhánh Ly | CSL? Quyên | CSTu?n Ng?c | TGTr?nh Công Sơn | TGM?nh Phát
| 138 | 0 | 1 | 1 | 0 | 0 | 1 | 0
| 143 | 0 | 1 | 0 | 1 | 0 | 1 | 0
| 144 | 0 | 1 | 0 | 0 | 1 | 1 | 0
sau chuẩn hóa
| 138 | 0 | 0.577350269189626 | 0.577350269189626 | 0 | 0 | 0.577350269189626 | 0
| 143 | 0 | 0.577350269189626 | 0 | 0.577350269189626 | 0 | 0.577350269189626 | 0
| 144 | 0 | 0.577350269189626 | 0 | 0 | 0.577350269189626 | 0.577350269189626 | 0
User profile
| 0 | 0 | 7.50555349946513 | 2.88675134594813 | 2.3094010767585 | 2.3094010767585 | 7.50555349946513 | 0
trước chuẩn hóa
| ID | TLNh?c tr? t?nh | TLNh?c Tr?nh | CSKhánh Ly | CSL? Quyên | CSTu?n Ng?c | TGTr?nh Công Sơn | TGM?nh Phát
| 108 | 1 | 0 | 0 | 1 | 0 | 0 | 1
| 146 | 0 | 1 | 0 | 1 | 0 | 1 | 0
sau chuẩn hóa
| 108 | 0.577350269189626 | 0 | 0 | 0.577350269189626 | 0 | 0 | 0.577350269189626
| 146 | 0 | 0.577350269189626 | 0 | 0.577350269189626 | 0 | 0.577350269189626 | 0
Tính độ ưa thích
Giá trị yêu thích bài i 4 là: 1.33333333333333
Giá trị yêu thích bài i 5 là: 10

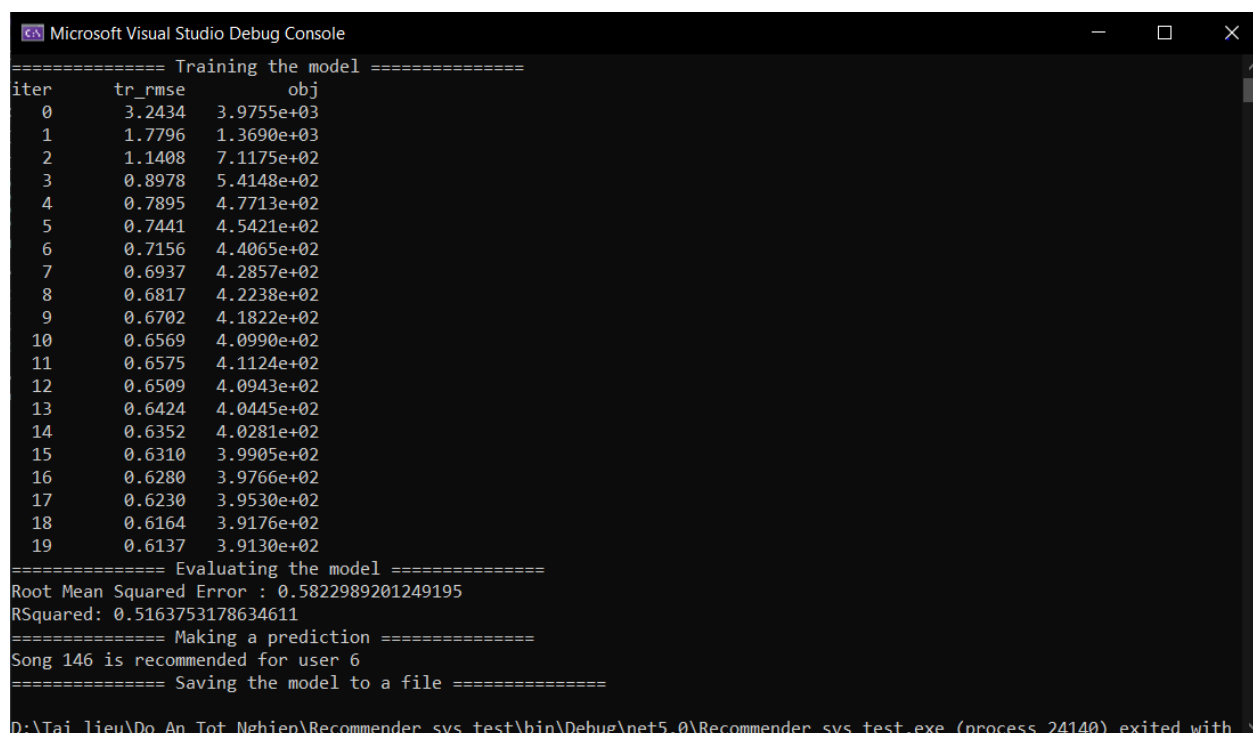
```

Hình 4. 7: Kết quả dự đoán độ yêu thích bài hát của người dùng

4.2.3 Kết quả thực nghiệm chức năng gợi ý collaborative filtering

Tiến hành train tập dữ liệu thu được kết quả:

- Phép đo sau 20 vòng lặp và hội tụ ngày càng gần 0.
- RMS (root of mean squares error) được sử dụng để đo lường sự khác biệt giữa các giá trị dự đoán của mô hình và các giá trị quan sát được của tập thử nghiệm. Nó càng thấp thì mô hình càng tốt.
- Rsquared cho biết mức độ phù hợp của dữ liệu với mô hình có giá trị từ 0 đến 1. Giá trị 0 có nghĩa là dữ liệu ngẫu nhiên, giá trị bằng 1 có nghĩa là mô hình khớp chính xác với dữ liệu.
- Kết quả dự đoán bài hát có ID_SONG là 146 cho người dùng có ID_USER là 6

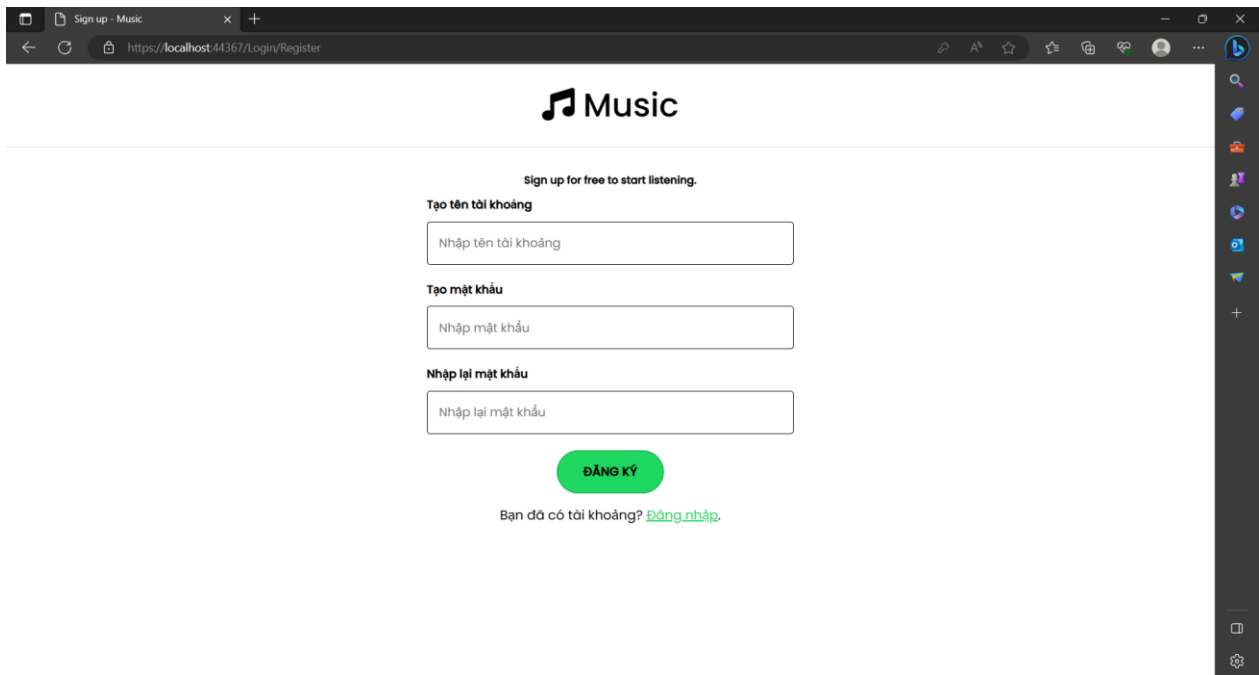


```
Microsoft Visual Studio Debug Console
===== Training the model =====
iter    tr_rmse    obj
0       3.2434    3.9755e+03
1       1.7796    1.3690e+03
2       1.1408    7.1175e+02
3       0.8978    5.4148e+02
4       0.7895    4.7713e+02
5       0.7441    4.5421e+02
6       0.7156    4.4065e+02
7       0.6937    4.2857e+02
8       0.6817    4.2238e+02
9       0.6702    4.1822e+02
10      0.6569    4.0990e+02
11      0.6575    4.1124e+02
12      0.6509    4.0943e+02
13      0.6424    4.0445e+02
14      0.6352    4.0281e+02
15      0.6310    3.9905e+02
16      0.6280    3.9766e+02
17      0.6230    3.9530e+02
18      0.6164    3.9176e+02
19      0.6137    3.9130e+02
===== Evaluating the model =====
Root Mean Squared Error : 0.5822989201249195
RSquared: 0.5163753178634611
===== Making a prediction =====
Song 146 is recommended for user 6
===== Saving the model to a file =====
D:\Tai_lieu\Do_An_Tot_Nghiep\Recommender_sys_test\bin\Debug\net5.0\Recommender_sys_test.exe (process 24140) exited with
```

Hình 4. 8: Kết quả sau khi train mô hình bằng ML.Net

4.3 Giao diện chương trình:

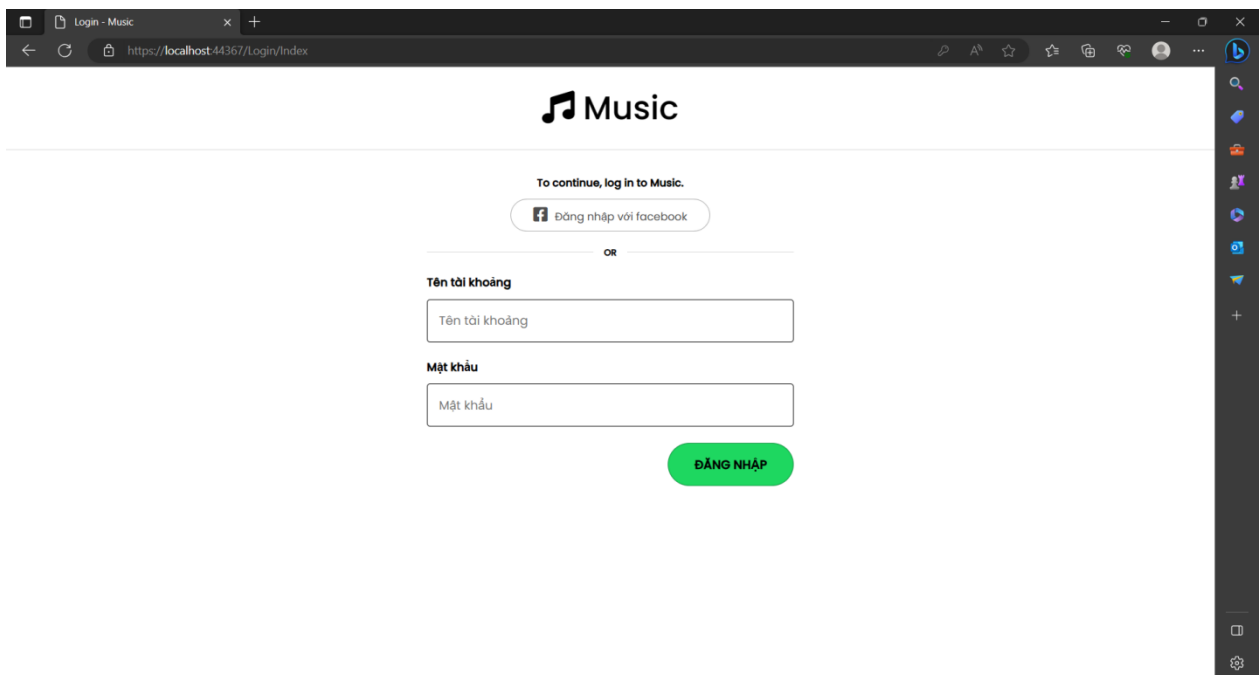
- Giao diện đăng ký:



The screenshot shows a web browser window with the address bar displaying "https://localhost:44367/Login/Register". The page title is "Sign up - Music". The main content area features the "Music" logo at the top, followed by the text "Sign up for free to start listening." Below this, there are three input fields: "Tạo tên tài khoản" (Create account name) with placeholder text "Nhập tên tài khoản", "Tạo mật khẩu" (Create password) with placeholder text "Nhập mật khẩu", and "Nhập lại mật khẩu" (Repeat password) with placeholder text "Nhập lại mật khẩu". A green button labeled "ĐĂNG KÝ" (Register) is positioned below the input fields. At the bottom, there is a link that says "Bạn đã có tài khoản? [Đăng nhập.](#)" (Do you have an account? [Log in.](#)).

Hình 4. 9: Giao diện đăng ký

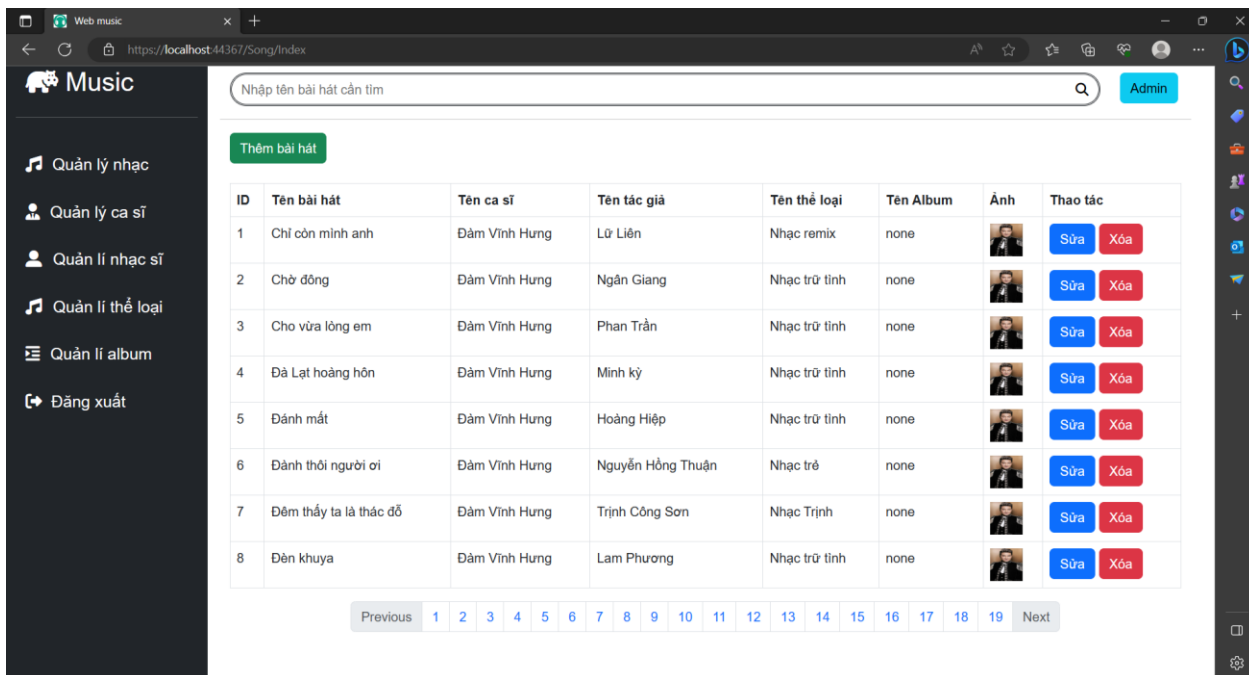
- Giao diện đăng nhập



The screenshot shows a web browser window with the address bar displaying "https://localhost:44367/Login/Index". The page title is "Login - Music". The main content area features the "Music" logo at the top, followed by the text "To continue, log in to Music." Below this, there is a button with the Facebook logo and the text "Đăng nhập với facebook" (Log in with facebook). A horizontal line with the text "OR" is positioned below the Facebook button. Below the line, there are two input fields: "Tên tài khoản" (Account name) with placeholder text "Tên tài khoản" and "Mật khẩu" (Password) with placeholder text "Mật khẩu". A green button labeled "ĐĂNG NHẬP" (Log in) is positioned below the input fields.

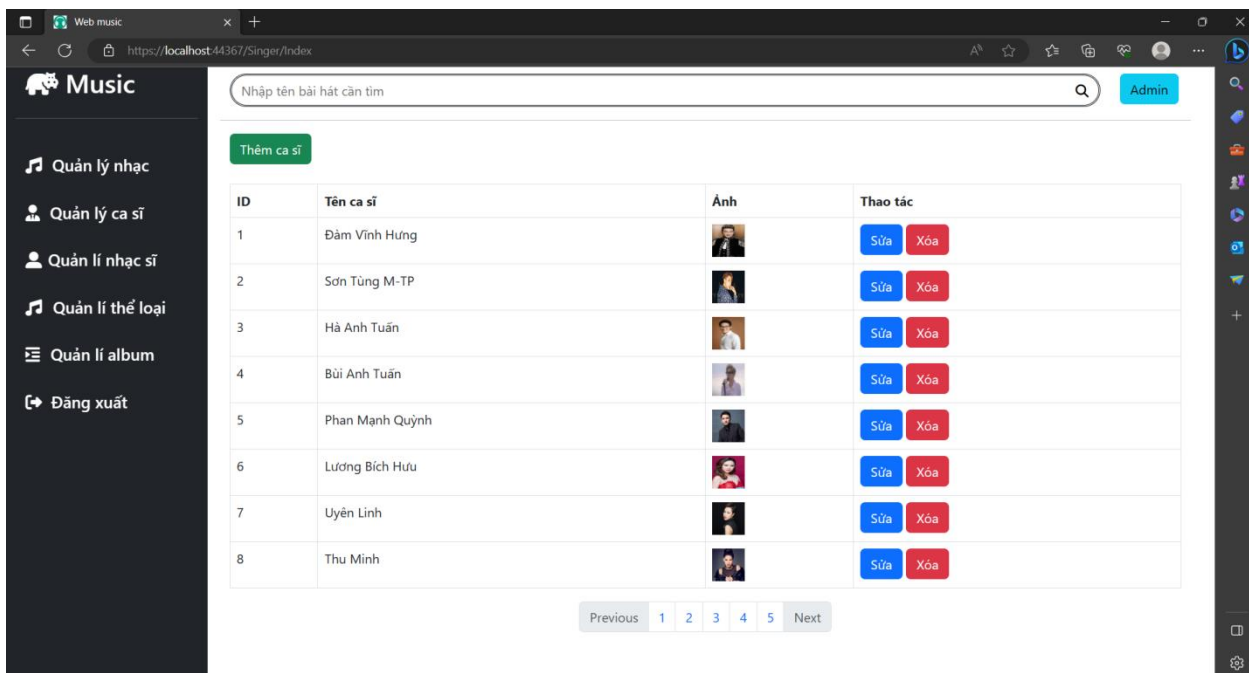
Hình 4. 10: Giao diện đăng nhập

- Giao diện quản lý bài hát:



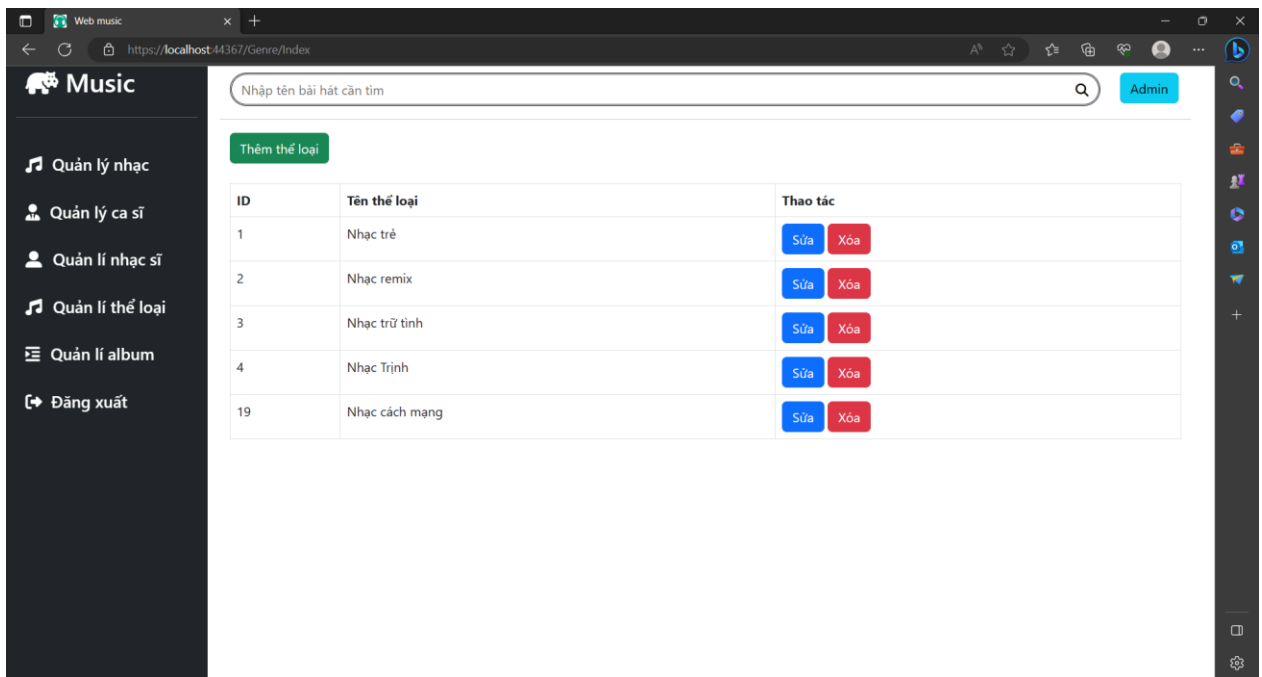
Hình 4. 11: Giao diện quản lý nhạc

- Giao diện quản lý ca sĩ:



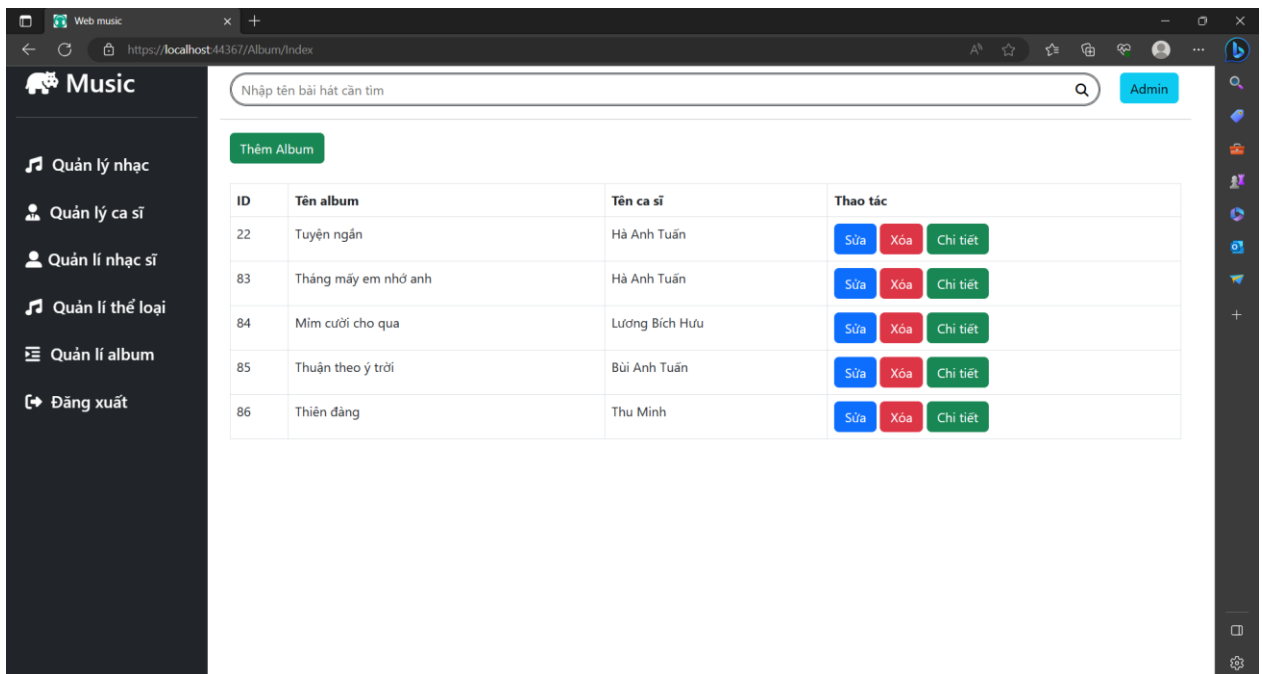
Hình 4. 12: Giao diện quản lý ca sĩ

- Giao diện quản lý thể loại:



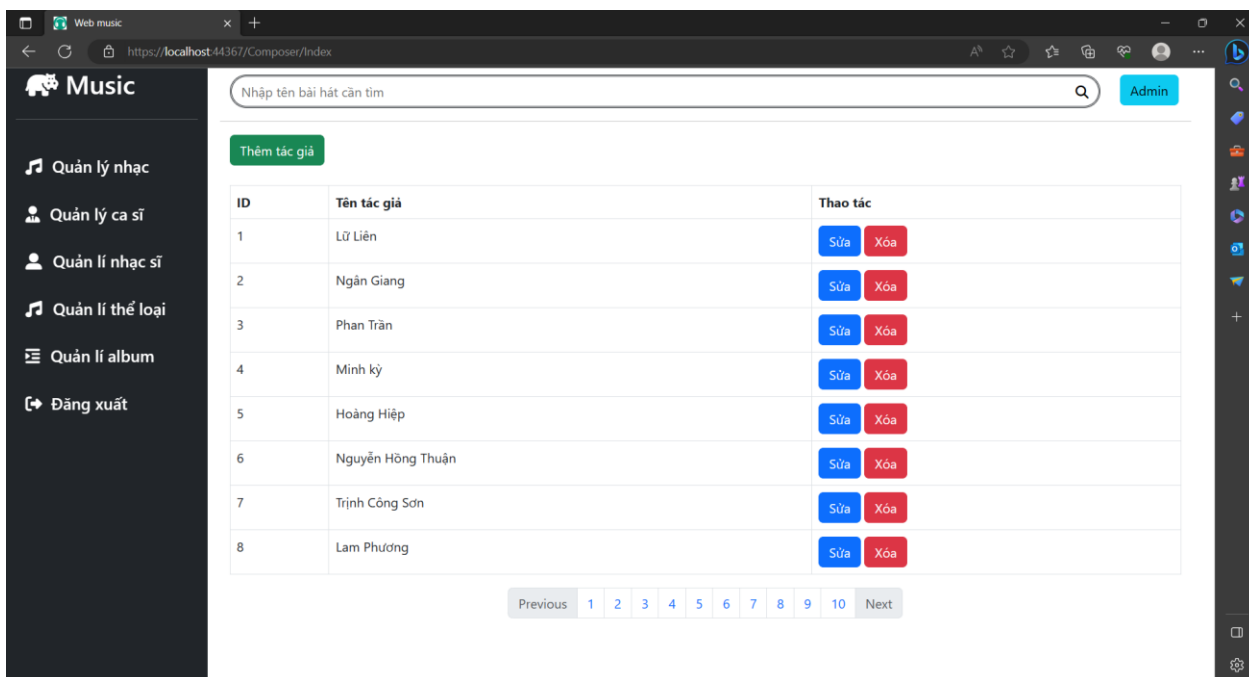
Hình 4. 13: Giao diện quản lý thể loại

- Giao diện quản lý album:



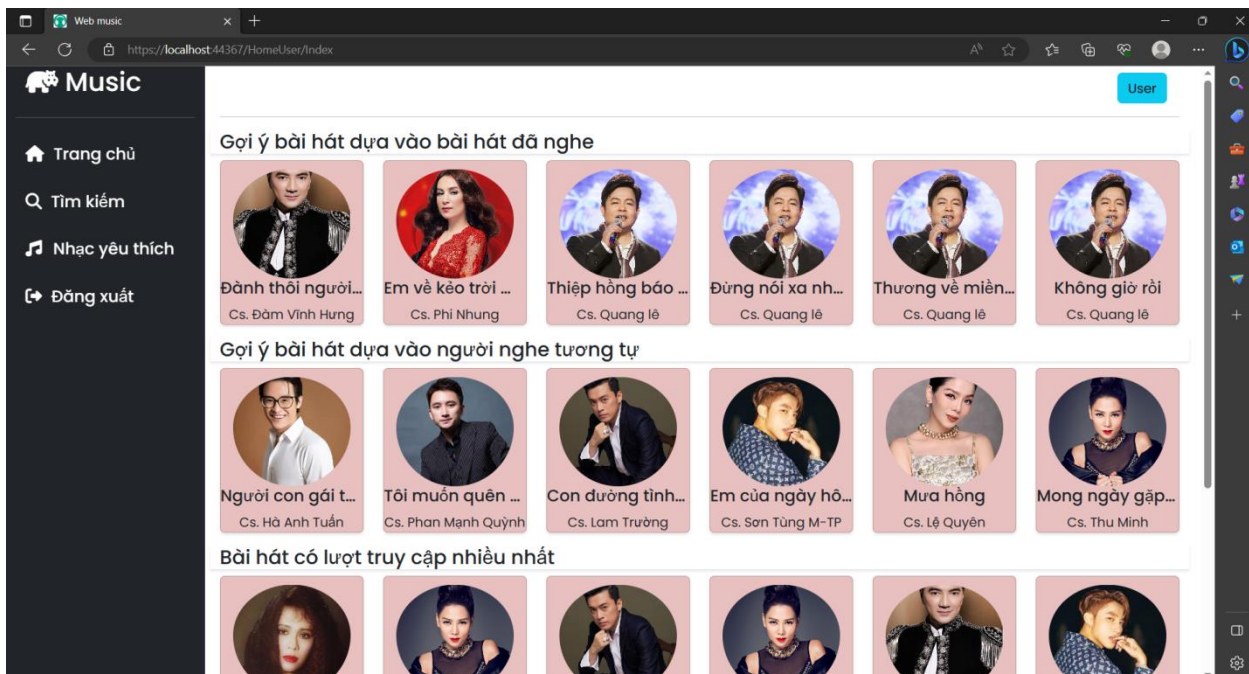
Hình 4. 14: Giao diện quản lý album

- Giao diện quản lý nhạc sĩ:



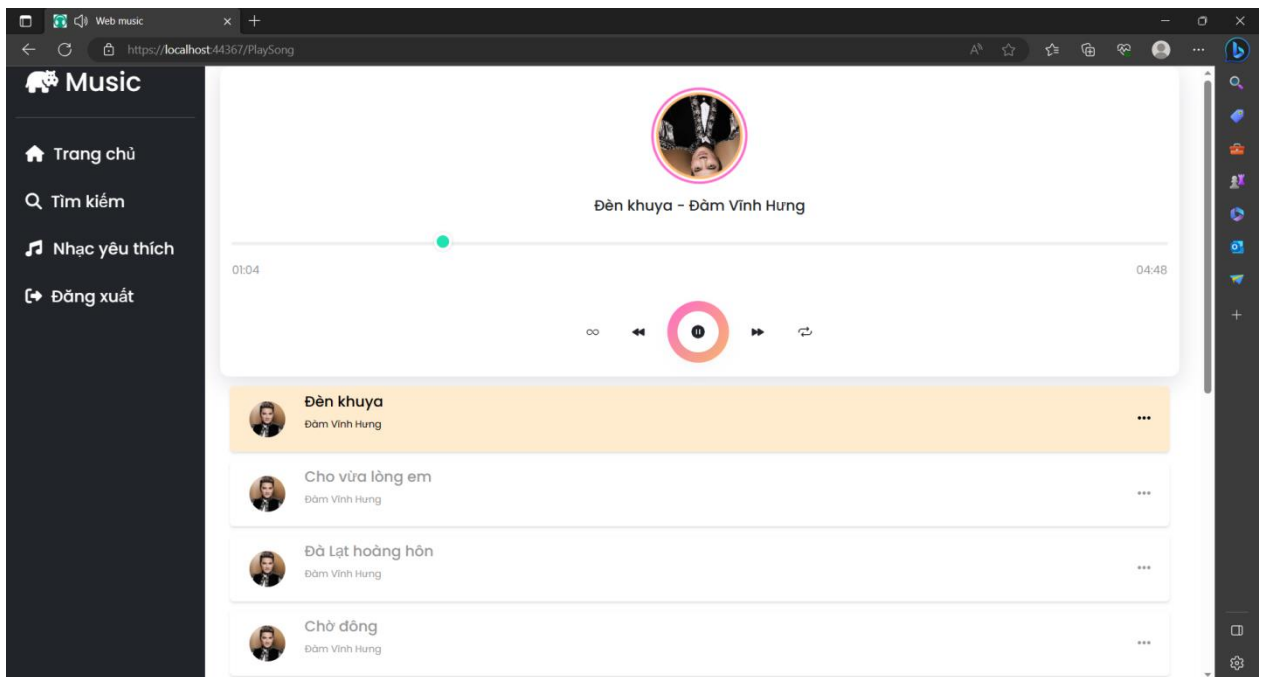
Hình 4. 15: Giao diện quản lý nhạc sĩ

- Giao diện trang chủ người dùng:



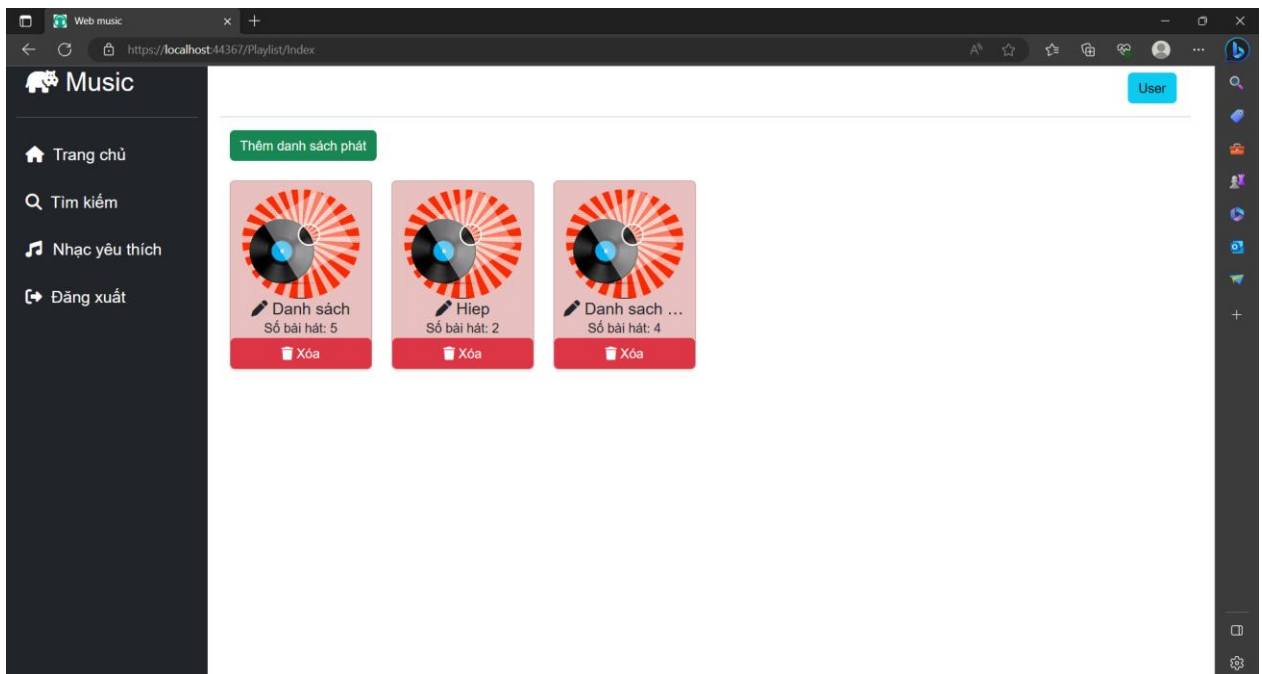
Hình 4. 16: Giao diện trang chủ người dùng

- Giao diện phát nhạc:



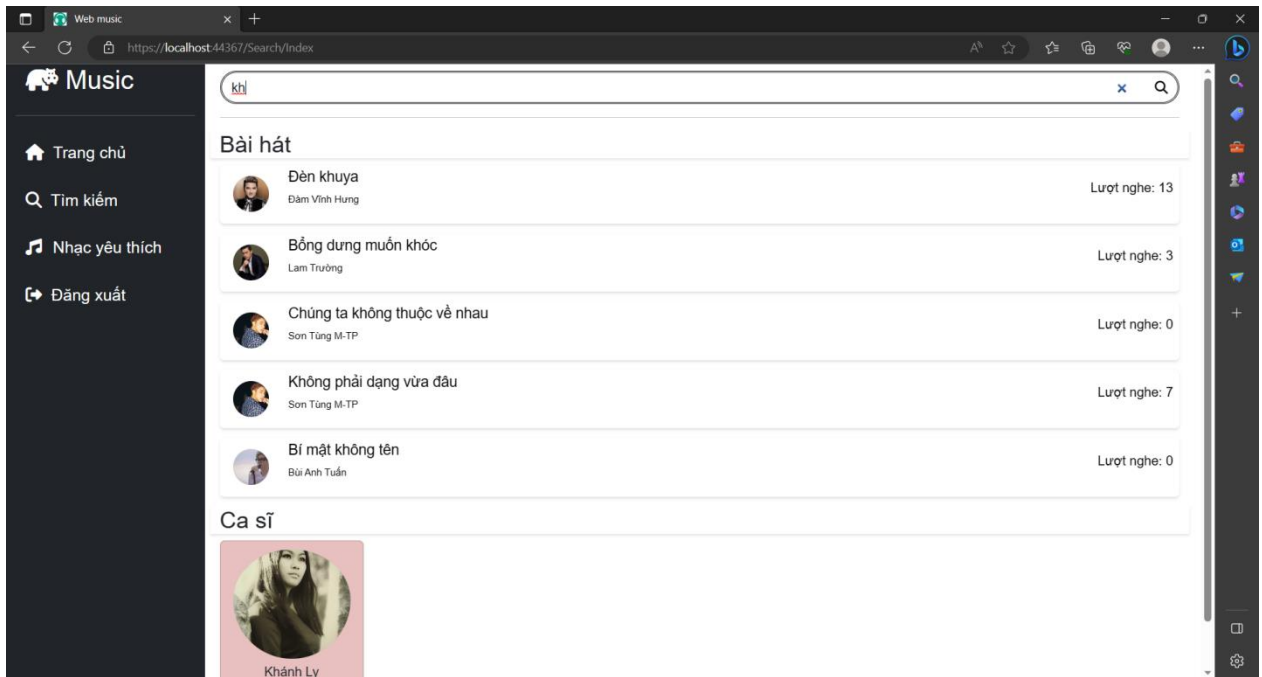
Hình 4. 17: Giao diện phát nhạc

- Giao diện quản lý danh sách phát:



Hình 4. 18: Giao diện quản lý danh sách phát

- Giao diện tìm kiếm:



Hình 4. 19: Giao diện tìm kiếm của user

KẾT LUẬN

Kết quả đạt được

Hoàn thành đề tài với các yêu cầu đặt ra. Trong quá trình thực hiện, em học hỏi được nhiều kiến thức mới và củng cố kiến thức cũ. Xây dựng thành công các chức năng cơ bản của website nghe nhạc để phục vụ nhu cầu giải trí âm nhạc cho người dùng. Về hệ thống gợi ý, xây dựng cơ bản thành công hệ thống gợi ý theo hai nhóm là content-based filtering và collaborative filtering. Về kỹ thuật gợi ý content-based filtering giúp hệ thống khai thác dữ liệu về những bài hát của người dùng đã nghe và đưa ra gợi ý. Về kỹ thuật collaborative hệ thống khai thác dữ liệu hành vi của người dùng và đưa ra gợi ý.

Vấn đề tồn đọng

Vì giới hạn về thời gian và khả năng thực hiện nên vẫn còn vài vấn đề tồn đọng

- Các chức năng của website chỉ phục vụ cơ bản nhu cầu nghe nhạc và tìm kiếm nhạc.
- Dữ liệu của bài hát và người dùng chưa nhiều, thiếu thực tế nên hệ thống gợi ý collaborative filtering thiếu chính xác trong vài trường hợp.

Hướng phát triển

Trong tương lai, em sẽ cố gắng khắc phục những hạn chế trên. Ngoài ra, em muốn nghiên cứu thêm dữ liệu và thuật toán để dự đoán được tính cách, xu hướng nghe nhạc của người sử dụng.

PHỤ LỤC

Phụ lục 1: hướng dẫn cài đặt

Link github: https://github.com/hiep1512001/DATN_WebMusic_Recommender_system

TÀI LIỆU THAM KHẢO

- [1]. Phạm Thị Miên, *Giáo trình phân tích thiết kế hướng đối tượng*, Đại học Giao thông vận tải phân hiệu tại tp.HCM.
- [2]. Phạm Thị Miên, *Giáo trình phân tích thiết kế hệ thống*, Đại học Giao thông vận tải phân hiệu tại tp.HCM.
- [3]. [Online] “Thiết kế website nghe nhạc online” Available: “<https://bizfly.vn/techblog/thiet-ke-website-nghe-nhac.html>” [Accessed 15/4/2023]
- [4]. [Online] “Giới thiệu nền tảng máy học ML.Net” Available: “<https://topdev.vn/blog/gioi-thieu-nen-tang-may-hoc-ml-net-cua-microsoft-bai-1/>” [Accessed 20/5/2023]
- [5]. [Online] ”Content-based Recommendation System” Available: “<https://machinelearningcoban.com/2017/05/17/contentbasedrecommendersys/>” [Accessed 16/5/2023]
- [6]. [Online] “Matrix Factorization Collaborative Filtering” Available: “<https://machinelearningcoban.com/2017/05/31/matrixfactorization/>” [Accessed 16/5/2023]
- [7]. [Online] “Building a Music Recommendation Engine” Available: “<https://www.section.io/engineering-education/building-spotify-recommendation-engine/>” [23/5/2023]
- [8]. [Online] “Build A Product Recommender Using C# and ML.NET Machine Learning” Available: “<https://mdfarragher.medium.com/build-a-product-recommender-using-c-and-ml-net-machine-learning-ab890b802d25>” [Accessed 24/3/2023]
- [9]. [Online] “Machine Learning with ML.NET – Recommendation Systems” Available: “<https://rubikscore.net/2021/03/15/machine-learning-with-ml-net-recommendation-systems/>” [Accessed 18/5/2023]
- [10]. [Online] “Beginners Guide to learn about Content Based Recommender Engines” Available: “<https://www.analyticsvidhya.com/blog/2015/08/beginners-guide-learn-content-based-recommender-systems/>” [Accessed 17/5/2023]