

**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: NGHIÊN CỨU KỸ THUẬT GỢI Ý CONTENT-BASED  
FILTERING VÀ COLLABORATIVE FILTERING VÀO XÂY DỰNG  
WEBSITE NGHE NHẠC**

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: NGHIÊN CỨU KỸ THUẬT GỢI Ý CONTENT-BASED  
FILTERING VÀ COLLABORATIVE FILTERING VÀO XÂY DỰNG  
WEBSITE NGHE NHẠC**

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**

Nghiên cứu kỹ thuật gợi ý content-based filtering và collaborative filtering vào xây dựng website nghe nhạc.

**2. Mục đích, yêu cầu**

a. Mục đích

Xây dựng website nghe nhạc, xây dựng 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 ẩn: 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

- 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 .....	8
DANH MỤC HÌNH ẢNH.....	9
CHƯƠNG 1. TỔNG QUAN .....	11
1.1 Tổng quan về đề tài.....	11
1.1.1 Lý do chọn đề tài .....	11
1.1.2 Phát biểu bài toán .....	11
1.1.3 Mô tả hoạt động nghiệp vụ.....	12
1.2 Mục tiêu và nhiệm vụ nghiên cứu.....	12
1.3 Phạm vi .....	12
1.4 Cấu trúc báo cáo thực tập tốt nghiệp.....	12
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT.....	14
2.1 Lý thuyết các công nghệ sử dụng về front-end.....	14
2.1.1 HTML.....	14
2.1.2 CSS .....	14
2.1.3 Javascript .....	14
2.1.4 Bootstrap.....	15
2.1.5 jQuery .....	15
2.1.6 Ajax .....	16
2.2 Lý thuyết công nghệ sử dụng về back-end .....	16
2.2.1 Ngôn ngữ c# .....	16
2.2.2 ASP.NET MVC.....	17
2.2.3 SQL server.....	18
2.2.4 Entity framework.....	19
2.2.5 ML.NET .....	21



CHƯƠNG 3. HỆ THỐNG GỢI Ý .....	25
3.1 Tổng quan hệ thống gợi ý .....	25
3.1.1 Machine learning .....	25
3.1.2 Tổng quan hệ thống gợi ý (Recommender systems) .....	26
3.2 Hệ thống gợi ý dựa trên nội dung (content-based systems) .....	28
3.3 Hệ thống gợi ý dựa trên lọc cộng tác (collaborative filtering) .....	30
CHƯƠNG 4. PHÂN TÍCH THIẾT KẾ HỆ THỐNG .....	37
4.1 Tổng quát bài toán .....	37
4.1.1 Phát biểu bài toán .....	37
4.1.2 Mô tả hoạt động nghiệp vụ .....	37
4.2 Sơ đồ BFD .....	38
4.3 Sơ đồ Usecase .....	39
4.4 Sơ đồ hoạt động .....	41
CHƯƠNG 5. THIẾT KẾ CHƯƠNG TRÌNH .....	56
5.1 Thiết kế cơ sở dữ liệu .....	56
5.1.1 Mô hình quan hệ thực thể (mô hình ER) .....	56
5.1.2 Biểu đồ các bảng trong cơ sở dữ liệu .....	57
5.2 Thiết kế hệ thống gợi ý .....	57
5.2.1 Ý tưởng .....	57
5.2.2 Hệ thống gợi ý dựa vào nội dung (content-based systems) .....	58
5.2.3 Hệ thống gợi ý dựa vào lọc cộng tác (collaborative systems) .....	63
5.3 Giao diện chương trình: .....	66
KẾT LUẬN .....	72
1. Kết quả đạt được .....	72
2. Vấn đề tồn đọng .....	72
PHỤ LỤC .....	73
TÀI LIỆU THAM KHẢO .....	74

## 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	SVD	Singular value decomposition	
8	NBCF	Neighborhood-based collaborative filtering	
9	ONNX	Open neural network exchange	
10	ER	Employee Relation	

## DANH MỤC HÌNH ẢNH

Hình 2. 1: Mô hình datafirst .....	19
Hình 2. 2: Mô hình codefirst .....	20
Hình 2. 3: Mô hình model first .....	20
Hình 2. 4: Cơ chế hoạt động ML.Net .....	22
Hình 3. 1: Công thức tính TF-IDF .....	29
Hình 3. 2: Ví dụ mô hình không gian .....	30
Hình 3. 3: Kỹ thuật phân rã ma trận .....	31
Hình 3. 4: Công thức tính vector w .....	31
Hình 3. 5: Mở rộng ma trận y .....	32
Hình 3. 6: Mô phỏng kỹ thuật phân rã ma trận .....	32
Hình 3. 7: Hàm mất mát .....	33
Hình 3. 8: Công thức tối ưu w .....	33
Hình 3. 9: Công thức tối ưu x .....	33
Hình 3. 10: Công thức tối ưu n .....	34
Hình 3. 11: Hàm mất mát .....	34
Hình 3. 12: Đạo hàm của hàm mất mát .....	34
Hình 3. 13: Công thức cập nhật cột cho w .....	34
Hình 3. 14: Công thức cập nhật mỗi hàng ứng với item .....	34
Hình 3. 15: Công thức cập nhật mỗi hàng sau khi biến đổi .....	35
Hình 3. 16: Công thức cập nhật mỗi hàng .....	35
Hình 3. 17: Công thức xác nhận rating của user m cho item n .....	35
Hình 3. 18: Hàm mất mát .....	36
Hình 3. 19: Hàm tối ưu w và n .....	36
Hình 4. 1: Sơ đồ BFD .....	38
Hình 4. 2: Sơ đồ Usecase admin .....	39
Hình 4. 3: Sơ đồ Usecase user .....	40
Hình 4. 4: Sơ đồ hoạt động đăng ký .....	41
Hình 4. 5: Sơ đồ hoạt động đăng nhập .....	42
Hình 4. 6: Sơ đồ hoạt động thêm bài hát .....	43
Hình 4. 7: Sơ đồ hoạt động sửa bài hát .....	43
Hình 4. 8: Sơ đồ hoạt động xóa bài hát .....	44
Hình 4. 9: Sơ đồ hoạt động tìm kiếm bài hát .....	44
Hình 4. 10: Sơ đồ hoạt động thêm ca sĩ .....	45
Hình 4. 11: Sơ đồ hoạt động sửa ca sĩ .....	45
Hình 4. 12: Sơ đồ hoạt động xóa ca sĩ .....	46
Hình 4. 13: Sơ đồ hoạt động tìm ca sĩ .....	46
Hình 4. 14: Sơ đồ hoạt động thêm thể loại .....	47
Hình 4. 15: Sơ đồ hoạt động sửa thể loại .....	47
Hình 4. 16: Sơ đồ hoạt động xóa thể loại .....	48
Hình 4. 17: Sơ đồ hoạt động thêm album .....	48
Hình 4. 18: Sơ đồ hoạt động sửa album .....	49
Hình 4. 19: Sơ đồ hoạt động xóa album .....	49
Hình 4. 20: Sơ đồ hoạt động tìm album .....	50
Hình 4. 21: Sơ đồ hoạt động thêm ca sĩ .....	50
Hình 4. 22: Sơ đồ hoạt động sửa ca sĩ .....	51

Hình 4. 23: Sơ đồ hoạt động xóa ca sĩ.....	51
Hình 4. 24: Sơ đồ hoạt động tìm nhạc sĩ .....	52
Hình 4. 25: Sơ đồ hoạt động thêm danh sách phát của user .....	52
Hình 4. 26: Sơ đồ hoạt động sửa danh sách phát của user .....	53
Hình 4. 27: Sơ đồ hoạt động xóa danh sách phát của user .....	53
Hình 4. 28: Sơ đồ hoạt động thêm nhạc vào danh sách phát của user .....	54
Hình 4. 29: Sơ đồ hoạt động xóa nhạc khỏi danh sách phát của user .....	54
Hình 4. 30: Sơ đồ hoạt động tìm kiếm của user .....	55
Hình 4. 31: Sơ đồ hoạt động nghe nhạc của user .....	55
Hình 5. 1: Mô hình ER.....	56
Hình 5. 2: Các bảng trong cơ sở dữ liệu .....	57
Hình 5. 3: Giao diện đăng ký.....	66
Hình 5. 4: Giao diện đăng nhập.....	66
Hình 5. 5: Giao diện quản lý nhạc .....	67
Hình 5. 6: Giao diện quản lý ca sĩ .....	67
Hình 5. 7: Giao diện quản lý thể loại.....	68
Hình 5. 8: Giao diện quản lý album .....	68
Hình 5. 9: Giao diện quản lý nhạc sĩ .....	69
Hình 5. 10: Giao diện trang chủ người dùng.....	69
Hình 5. 11: Giao diện phát nhạc .....	70
Hình 5. 12: Giao diện quản lý danh sách phát.....	70
Hình 5. 13: Giao diện tìm kiếm của user.....	71

# CHƯƠNG 1. TỔNG QUAN

## 1.1 Tổng quan về đề tài

### 1.1.1 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,...

### 1.1.2 Phát biểu bài toán

- Qua những thông tin tìm hiểu thực tế, ta 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.

### **1.1.3 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.

## **1.2 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.

## **1.3 Phạm vi**

- Website nghe nhạc hướng đến tất cả các đối tượng âm nhạc.

- Nghiên cứu về học máy.

## **1.4 Cấu trúc báo cáo thực tập tốt nghiệp**

- Tổng quan

- Cơ sở lý thuyết

- Hệ thống gợi ý

- Phân tích thiết kế hệ thống
- Xây dựng chương trình

## **CHƯƠNG 2. CƠ SỞ LÝ THUYẾT**

### **2.1 Lý thuyết các công nghệ sử dụng về front-end**

#### **2.1.1 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.

#### **2.1.2 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.

#### **2.1.3 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.

### **2.1.4 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. Bạn có thể tạo các giao diện responsive design cho các thiết bị khác nhau với Bootstrap.

### **2.1.5 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 bạn.

- 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.

### **2.1.6 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.

## **2.2 Lý thuyết công nghệ sử dụng về back-end**

### **2.2.1 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.

### **2.2.2 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 bạn kiểm soát đầy đủ về đánh dấu, hỗ trợ phát triển thân thiện với TDD và sử dụng các tiêu chuẩn web mới nhất.

- Mô hình MVC

- + Mô hình kiến trúc Model-View-Controller (MVC) phân tách một ứng dụng thành ba nhóm thành phần chính: Models, Views và Controllers. Mô hình này giúp đạt được sự phân tách quan tâm. Sử dụng mô hình này, các yêu cầu của người dùng được định tuyến đến một Controller chịu trách nhiệm với Model để thực hiện các hành động của người dùng và / hoặc lấy kết quả của các truy vấn. Controller chọn View để hiển thị cho người dùng và cung cấp cho nó bất kỳ dữ liệu Model nào nó yêu cầu.

- + Model Responsibility: Model trong một ứng dụng MVC đại diện cho trạng thái của ứng dụng và bất kỳ logic kinh doanh hoặc hoạt động nào cần được thực hiện bởi nó. Logic kinh doanh nên được đóng gói trong mô hình, cùng với bất kỳ logic

triển khai nào để lưu trạng thái của ứng dụng. Các view được gõ chắc chắn thường sử dụng các loại ViewModel được thiết kế để chứa dữ liệu để hiển thị trên view đó. Controller tạo và điền các trường hợp ViewModel này từ mô hình

+View Responsibility: Views chịu trách nhiệm cho việc trình bày nội dung thông qua giao diện người dùng. Chúng sử dụng Razor view engine để nhúng mã .NET vào đánh dấu HTML. Có thể có tối thiểu logic trong các view và bất kỳ logic nào trong chúng nên liên quan đến việc trình bày nội dung. Nếu bạn thấy cần phải thực hiện một lượng lớn logic trong các tệp view để hiển thị dữ liệu từ một mô hình phức tạp, hãy xem xét sử dụng View Component, ViewModel hoặc mẫu view để đơn giản hóa view.

+ Controller Responsibility: Controller là các thành phần xử lý tương tác người dùng, làm việc với mô hình và cuối cùng chọn view để hiển thị. Trong một ứng dụng MVC, view chỉ hiển thị thông tin; controller xử lý và phản hồi cho đầu vào và tương tác của người dùng. Trong mô hình MVC, controller là điểm nhập ban đầu và chịu trách nhiệm chọn loại mô hình nào để làm việc và view nào để hiển thị (do đó có tên gọi của nó - nó kiểm soát cách ứng dụng phản hồi với yêu cầu nhất định).

### **2.2.3 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. Chúng ta 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.

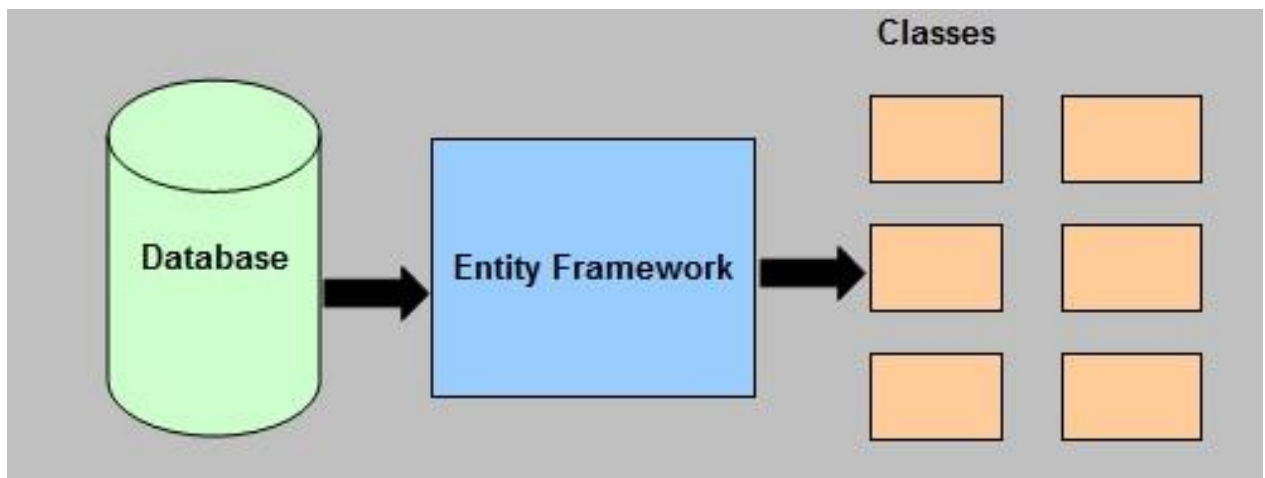
#### 2.2.4 Entity framework

- 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.

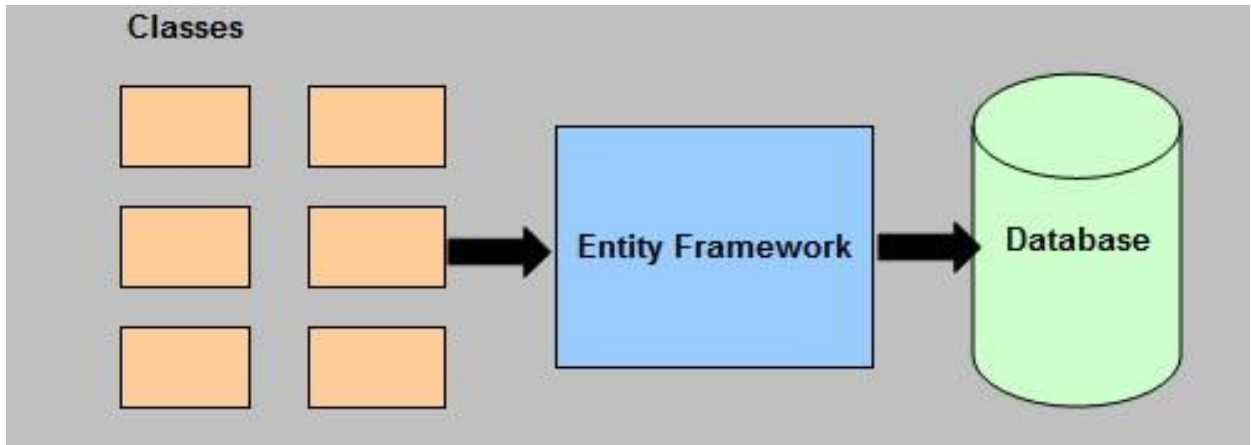
- Các tình huống dùng Entity Framework: Entity framework là rất hữu ích trong 3 tình huống sau:

- + Database First: Đầu tiên là nếu bạn đã có một CSDL hoặc muốn thiết kế CSDL trước khi làm các phần khác của ứng dụng.



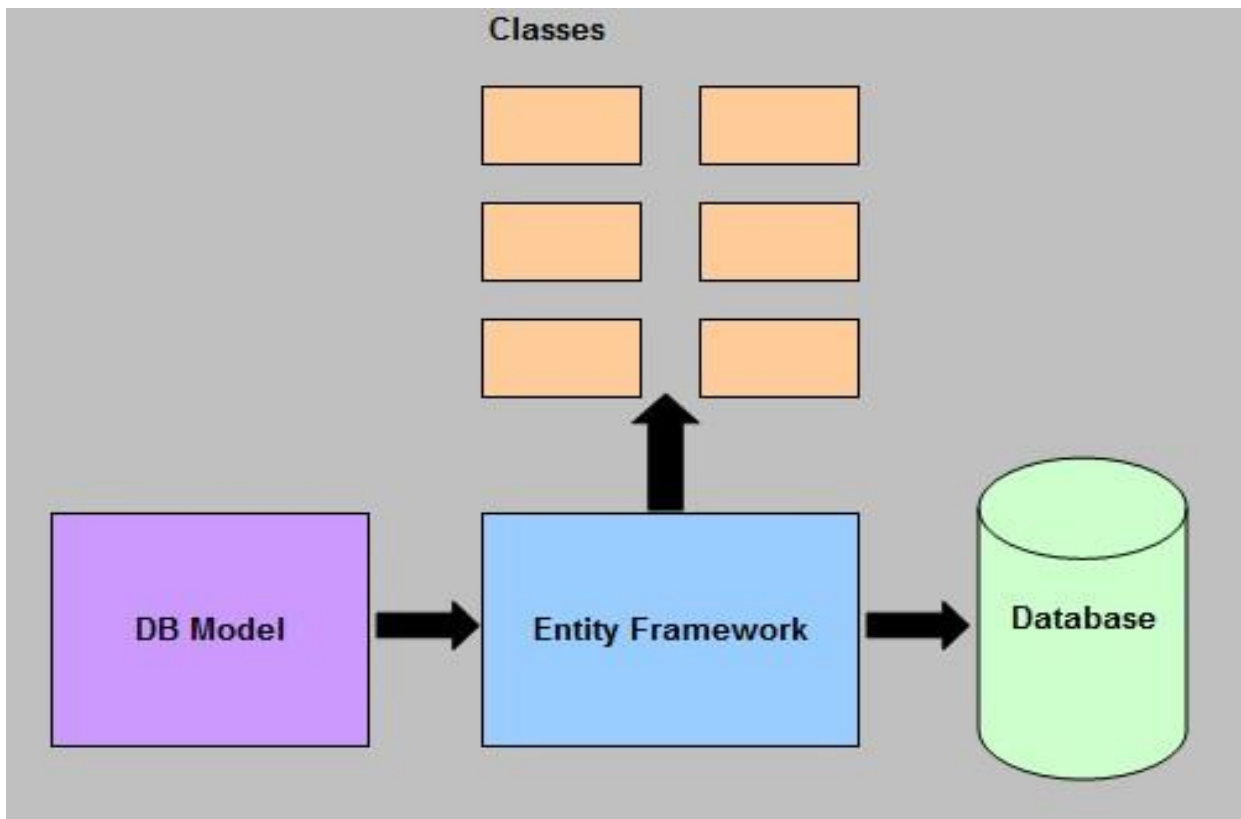
Hình 2. 1: Mô hình datafirst

+ Code First: Thứ hai là bạn muốn tập trung vào các domain class rồi mới tạo CSDL từ các domain class đó.



Hình 2. 2: Mô hình codefirst

+ Model First: Thứ ba là bạn muốn thiết kế schema của CSDL trên visual designer rồi mới tạo CSDL và các class.



Hình 2. 3: Mô hình model first

### 2.2.5 ML.NET

- ML.NET là thư viện máy học Mã nguồn mở và chạy cross-platform (Windows, Linux, macOS) của Microsoft. Ta 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

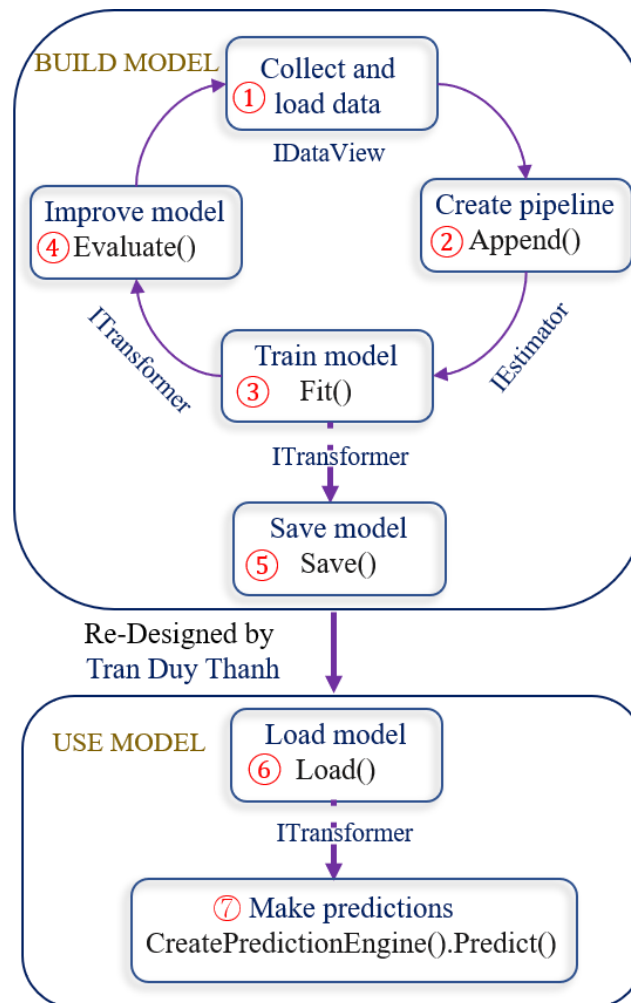
+ Clustering : Ví dụ các bài toán về gom cụm khách hàng, giả sử có N khách hàng ta cần phải gom thành k cụm, các cụm này chứa các đặc trưng khác nhau của khách hàng.

+ 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 này 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 2. 4: Cơ chế hoạt động ML.Net

+ Bước 1: là bước chuẩn bị dữ liệu, ở bước này tùy vào bài toán mà ta thu thập dữ liệu khác nhau, dữ liệu sau khi thu thập phải được làm sạch/ chuẩn hóa. Vì nguyên tắc vàng các bạn phải nhớ “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 ta 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 ta làm luôn 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 ta 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 (thường là vậy), vì thường 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 không? có được tiếp tục sử dụng và tiếp tục cải tiến mô hình nữa hay không. Do đó khi train mô hình xong thì ta cần đánh giá mô hình này chất lượng ra sao. Ta sẽ lấy Test set ở bước 1 để đánh giá. Hàm đánh giá này tên là Evaluate() nó nằm trong các lớp giải thuật mà ta 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 mà ta 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 mà ta 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 ta 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 thôi, 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 . ta 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 3. HỆ THỐNG GỌI Ý

### 3.1 Tổng quan hệ thống gọi ý

#### 3.1.1 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.

- Bài toán của machine learning thường được chia làm hai loại là dự đoán (prediction) và phân loại (classification). Các bài toán dự đoán thường là giá nhà, giá xe,..., còn các bài toán phân loại thường là nhận diện chữ viết tay, đồ vật, ...

- Các phương pháp Machine learning:

+ Supervised machine learning: Supervised learning còn được gọi là máy học có giám sát, được định nghĩa bằng cách sử dụng các tập dữ liệu được gắn nhãn để huấn luyện các thuật toán phân loại dữ liệu hoặc dự đoán kết quả một cách chính xác. Khi dữ liệu đầu vào được đưa vào mô hình, mô hình sẽ điều chỉnh trọng lượng của nó cho đến khi nó được lắp một cách thích hợp. Điều này xảy ra như một phần của quá trình xác nhận chéo để đảm bảo rằng mô hình tránh trạng bị quá nhiều hoặc trạng bị thiếu thông tin.

+ Supervised machine learning giúp các tổ chức giải quyết nhiều vấn đề trong thế giới thực trên quy mô lớn, chẳng hạn như phân loại thư rác trong một thư mục riêng biệt từ hộp thư đến của bạn. Một số phương pháp được sử dụng trong Supervised machine learning bao gồm: logistic regression, neural networks, linear regression, naive bayes, random forest, và support vector machine (SVM)

+ Unsupervised machine learning: Unsupervised machine learning là phương pháp sử dụng các thuật toán máy học để phân tích và phân cụm các tập dữ liệu không được gán nhãn. Không cần sự can thiệp của con người, các thuật toán này có thể phát hiện ra các mẫu hoặc nhóm dữ liệu ẩn. Khả năng phát hiện ra những điểm tương đồng và khác biệt trong thông tin của phương pháp này khiến nó trở nên lý tưởng cho việc phân tích dữ liệu khám phá, chiến lược bán chéo (cross-sell), phân khúc khách hàng cũng như nhận dạng hình ảnh và mẫu. Unsupervised machine learning cũng được sử dụng để giảm số lượng các tính năng trong một mô hình thông qua quá trình giảm kích thước. Phân tích thành phần chính (PCA) và phân tích giá trị đơn lẻ (SVD) là hai cách tiếp cận phổ biến cho việc này. Các thuật toán khác được sử dụng trong học tập không giám sát bao gồm: k-means clustering, neural networks, và probabilistic clustering methods.

- Semi-supervised learning: Semi-supervised learning cung cấp một phương pháp hiệu quả giữa học tập có giám sát và không giám sát. Trong quá trình đào tạo, nó sử dụng một tập dữ liệu có nhãn nhỏ hơn để hướng dẫn phân loại và trích xuất tính năng từ một tập dữ liệu lớn hơn, không được gán nhãn. Phương pháp Semi-supervised learning có thể giải quyết vấn đề không có đủ dữ liệu được gán nhãn cho thuật toán học có giám sát. Nó cũng hữu ích nếu quá tốn kém để gán nhãn đủ dữ liệu.

### **3.1.2 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 bạn like những nội dung nào để hiển thị newsfeed của bạn những nội dung liên quan. Bạn càng đam mê facebook, facebook càng được hưởng lợi, thế nên nó luôn mang tới bạn những thông tin mà khả năng cao là bạn muốn đọc. Thường thì với cách này, ta 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. Chúng ta cũng 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.

### **3.2 Hệ thống gợi ý dựa trên nội dung (content-based systems)**

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 chúng ta 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. Ví dụ: TF = 3 vs TF = 4 khác rất nhiều so với TF = 10 so với TF = 1000. Nói cách khác, mức độ liên quan của một từ trong tài liệu không thể được đo bằng số lượng thô đơn giản và do đó phương trình dưới đây:

$$w_{t,d} = \begin{cases} 1 + \log_{10} tf_{t,d}, & \text{if } tf_{t,d} > 0 \\ 0, & \text{otherwise} \end{cases}$$

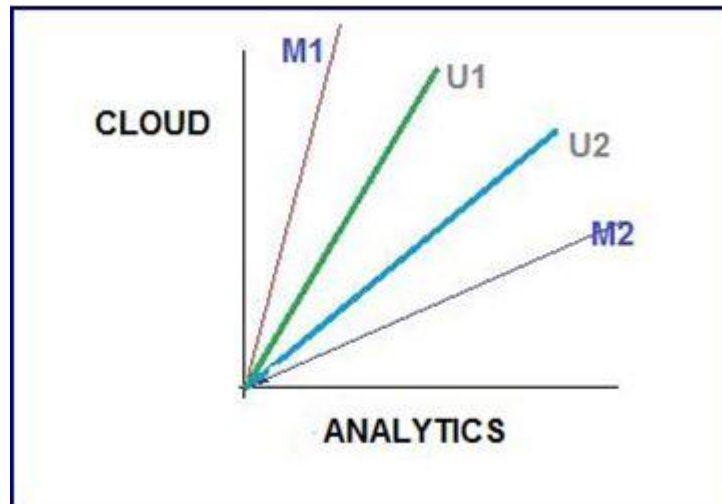
Term Frequency	Weighted Term Frequency
0	0
10	2
1000	4

Hình 3. 1: Công thức tính TF-IDF

- 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 3. 2: 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.

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

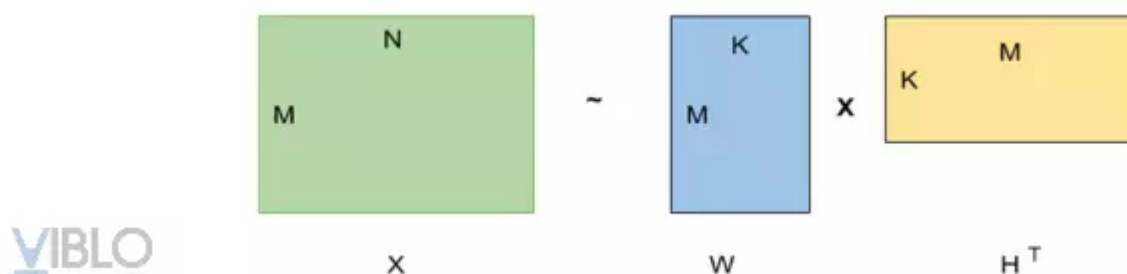
- Phương pháp lọc cộng tác hay hệ thống lọc cộng tác là phương pháp phân tích dữ liệu người dùng để tìm ra mối tương quan giữa các đối tượng người dùng. Lọc cộng tác hoạt động bằng cách xây dựng một cơ sở dữ liệu, lưu trữ dưới dạng ma trận người dùng (users) - sản phẩm (items) và mỗi dòng của nó là một 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 gọi 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 ta 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 3. 3: Kỹ thuật phân rã ma trận

- Ta 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,  $W \in \mathbb{R}^{|\mathcal{U}| \times K}$  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  $H \in \mathbb{R}^{|\mathcal{I}| \times K}$  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, chúng ta có  $x$  là một vector của item profile.

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

$$y \approx xw \quad (2.10)$$

Hình 3. 4: Công thức tính vector  $w$

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

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

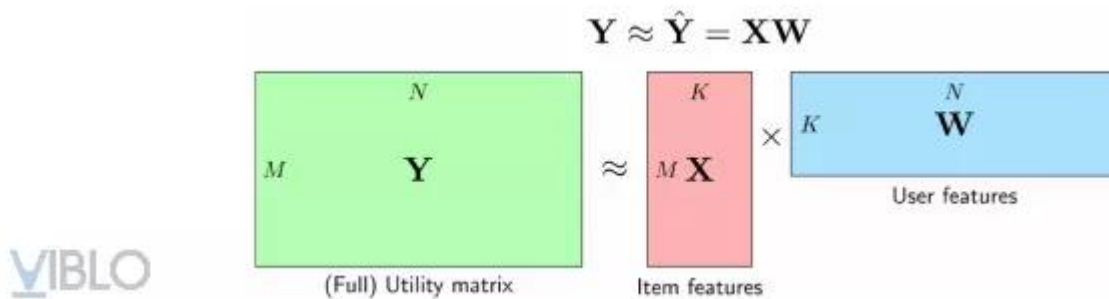
Hình 3. 5: Mở rộng ma trận y

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 ta 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 X và W, trong đó X là ma trận của toàn bộ các item profiles, mỗi hàng tương ứng với một item. Còn W là ma trận của toàn bộ user models (các mô hình của users), mỗi cột tương ứng với một user. Chúng ta sẽ cố gắng xấp xỉ utility matrix  $Y \in \mathbb{R}^{M \times N}$  bằng tích của hai ma trận con là  $X \in \mathbb{R}^{M \times K}$  và  $W \in \mathbb{R}^{K \times N}$ .

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



Hình 3. 6: Mô phỏng kỹ thuật phân ra ma trận

- 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, chúng ta sẽ xét hàm mất mát không có cả bias và biến tối ưu cho X và W:

$$\mathcal{L}_{(X,W)} = \frac{1}{2s} \sum_{n=1}^N \sum_{m:r_{mn}} (y_{mn} - x_m w_n)^2 + \frac{\lambda}{2} (\|X\|_F^2 + \|W\|_F^2) \quad (2.11)$$

Hình 3. 7: Hàm mất mát

- 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 tác dụng giúp tránh overfitting.

- Tiếp theo, chúng ta sẽ tối ưu X và 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 X, việc tối ưu W được đưa về hàm:

$$\mathcal{L}_{(W)} = \frac{1}{2s} \sum_{n=1}^N \sum_{m:r_{mn}} (y_{mn} - x_m w_n)^2 + \frac{\lambda}{2} \|W\|_F^2 \quad (2.12)$$

Hình 3. 8: Công thức tối ưu w

+ Khi cố định W, việc tối ưu X được đưa về tối ưu hàm:

$$\mathcal{L}_{(X)} = \frac{1}{2s} \sum_{n=1}^N \sum_{m:r_{mn}} (y_{mn} - x_m w_n)^2 + \frac{\lambda}{2} \|X\|_F^2 \quad (2.13)$$

Hình 3. 9: Công thức tối ưu x

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

+ Ta có thể thấy rằng, bài toán tối ưu  $W$  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  $W$ .

$$\mathcal{L}(w_n) = \frac{1}{2s} \sum_{m:r_{mn}} (y_{mn} - x_m w_n)^2 + \frac{\lambda}{2} \|w_n\|_2^2 \quad (2.14)$$

Hình 3. 10: Công thức tối ưu  $n$

+ Vì biểu thức  $(y_{mn} - x_m w_n)^2$  chỉ phụ thuộc vào các items đã được user đang xét đánh giá, nên ta 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:

$$\mathcal{L}(w_n) = \frac{1}{2s} \|\hat{y}_{mn} - \hat{X}_n w_n\|^2 + \frac{\lambda}{2} \|w_n\|_2^2 \quad (2.15)$$

Hình 3. 11: Hàm mất mát

+ Và đạo hàm của nó là:

$$\frac{\partial \mathcal{L}(w_n)}{\partial w_n} = -\frac{1}{s} \hat{X}_n^T (\hat{y}_n - \hat{X}_n w_n) + \lambda w_n \quad (2.16)$$

Hình 3. 12: Đạo hàm của hàm mất mát

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

$$w_n = w_n - \eta \left( \frac{1}{s} \hat{X}_n^T (\hat{y}_n - \hat{X}_n w_n) + \lambda w_n \right) \quad (2.17)$$

Hình 3. 13: Công thức cập nhật cột cho  $w$

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

$$\mathcal{L}(x_m) = \frac{1}{2s} \sum_{n:r_{mn}} (y_{mn} - x_m w_n)^2 + \frac{\lambda}{2} \|x_m\|_2^2 \quad (2.18)$$

Hình 3. 14: Công thức cập nhật mỗi hàng ứng với item

+ Đặt  $W_m$  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  $y_m$ . Công thức trên sẽ trở thành:

$$\mathcal{L}_{(x_m)} = \frac{1}{2s} \|\hat{y}^m - x_m \widehat{W}_m\|_2^2 + \frac{\lambda}{2} \|x_m\|_2^2 \quad (2.19)$$

Hình 3. 15: Công thức cập nhật mỗi hàng sau khi biến đổi

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

$$x_n = x_n - \eta \left( \frac{1}{s} (\hat{y}^m - x_m \widehat{W}_m) \widehat{W}_m^T + \lambda w x_m \right) \quad (2.19)$$

Hình 3. 16: Công thức cập nhật mỗi hàng

+ 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, ta sẽ 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, ta sẽ xét hàm mất mát với **bias** và **hệ số tối ưu cho  $X$  và  $W$** .

+ Như trong NBCF, chúng ta có bước chuẩn hóa ma trận để tránh sự thiên lệch do sự khó tính hay dễ tính khác nhau giữa các users. Với MF, ta 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  $X$  và  $W$ .

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

$$y_{mn} \approx x_m y_n + b_m + d_n + \mu \quad (2.20)$$

Hình 3. 17: Công thức xác nhận rating của user  $m$  cho item  $n$

với  $b_m, d_n, \mu$  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:

$$\mathcal{L}(X, W, b, d) = \frac{1}{2s} \sum_{n=1}^N \sum_{m:r_{mn}} (x_m w_n + b_m + d_n + \mu - y_{mn})^2 + \frac{\lambda}{2} (\|X\|_F^2 + \|W\|_F^2 + \|b\|_F^2 + \|d\|_F^2) \quad (2.21)$$

Hình 3. 18: Hàm mất mát

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

$$w_n = w_n - \eta \frac{1}{s} X_n (X_n^T w_n + b_n + d_n - y_{nn}) + \lambda w_n \quad (2.22)$$

$$b_n = b_n - \eta \frac{1}{s} \mathbf{1}^T (X_n^T w_n + b_n + d_n - y_{nn}) \quad (2.23)$$

$$x_m = x_m - \eta \frac{1}{s} W_m (W_m^T x_m + b_n + d_n - y_{mn}) + \lambda x_m \quad (2.24)$$

$$d_m = d_m - \eta \frac{1}{s} \mathbf{1}^T (W_m^T x_m + b_n + d_n - y_{mn}) \quad (2.25)$$

Hình 3. 19: Hàm tối ưu w và n

+ Cuối cùng, ta sẽ thu được các ma trận X, b, W, d, từ đó dự đoán các ratings chưa biết.

## CHƯƠNG 4. PHÂN TÍCH THIẾT KẾ HỆ THỐNG

### 4.1 Tổng quát bài toán

#### 4.1.1 Phát biểu bài toán

- Qua những thông tin tìm hiểu thực tế, ta 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.

#### 4.1.2 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.

## 4.2 Sơ đồ BFD



Hình 4. 1: Sơ đồ BFD



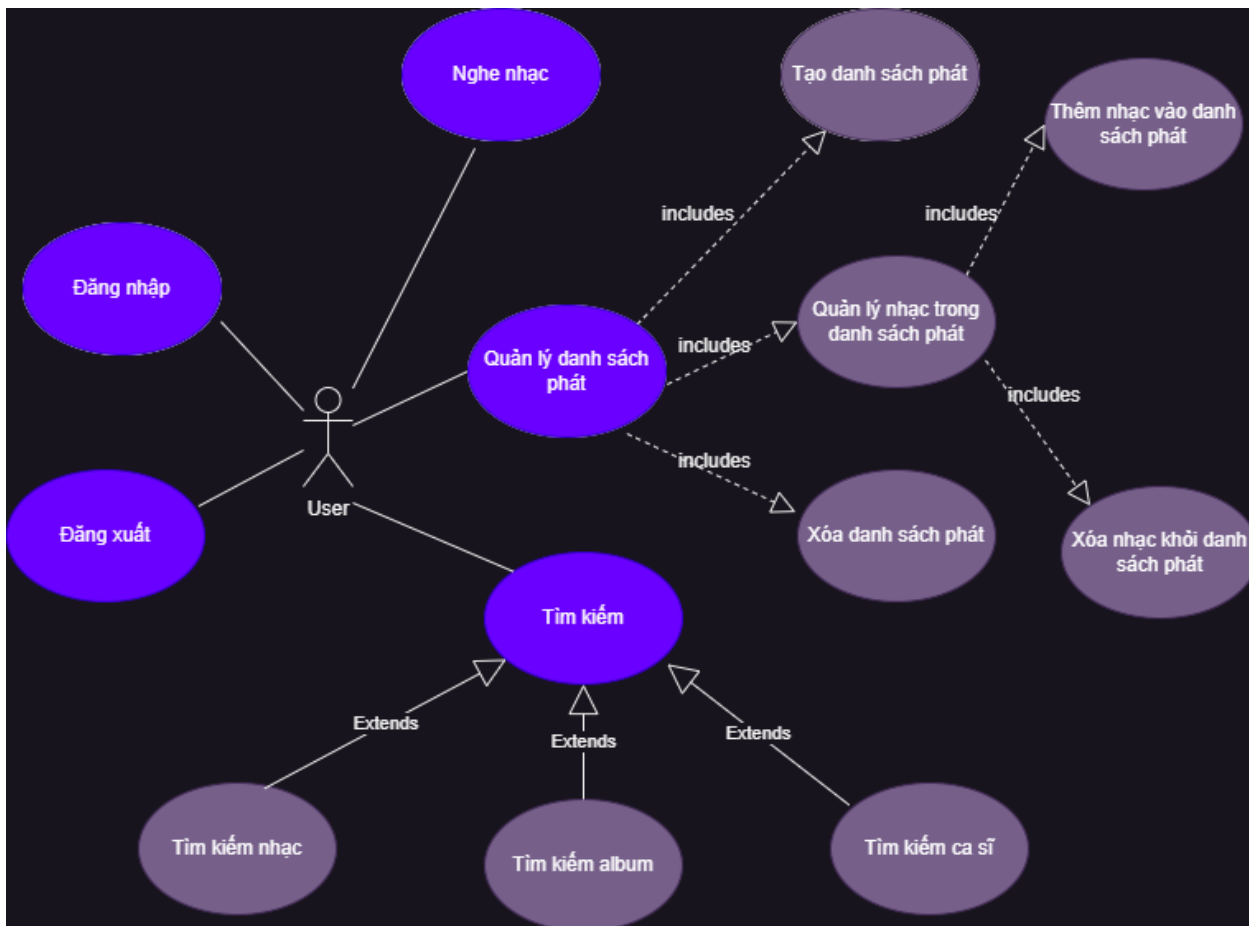
### 4.3 Sơ đồ Usecase

-Sơ đồ Usecase của Admin



Hình 4. 2: Sơ đồ Usecase admin

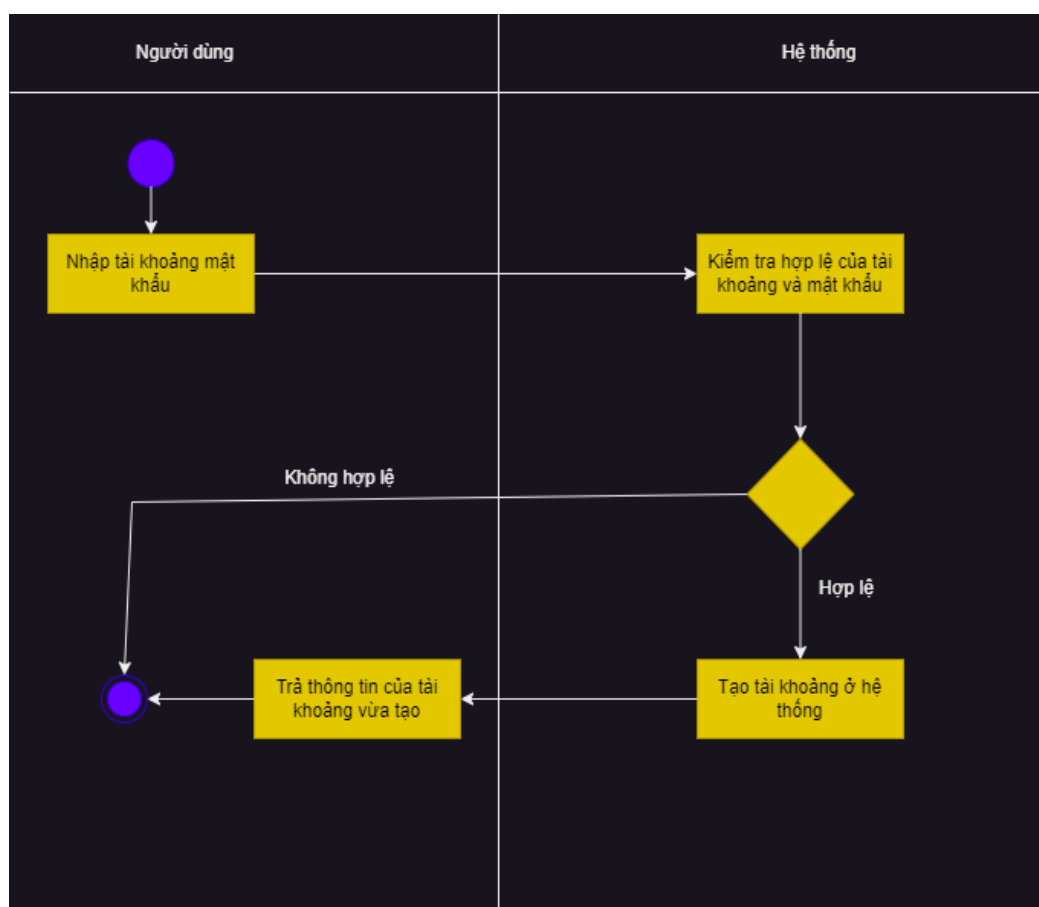
- Sơ đồ Usecase của user:



Hình 4. 3: Sơ đồ Usecase user

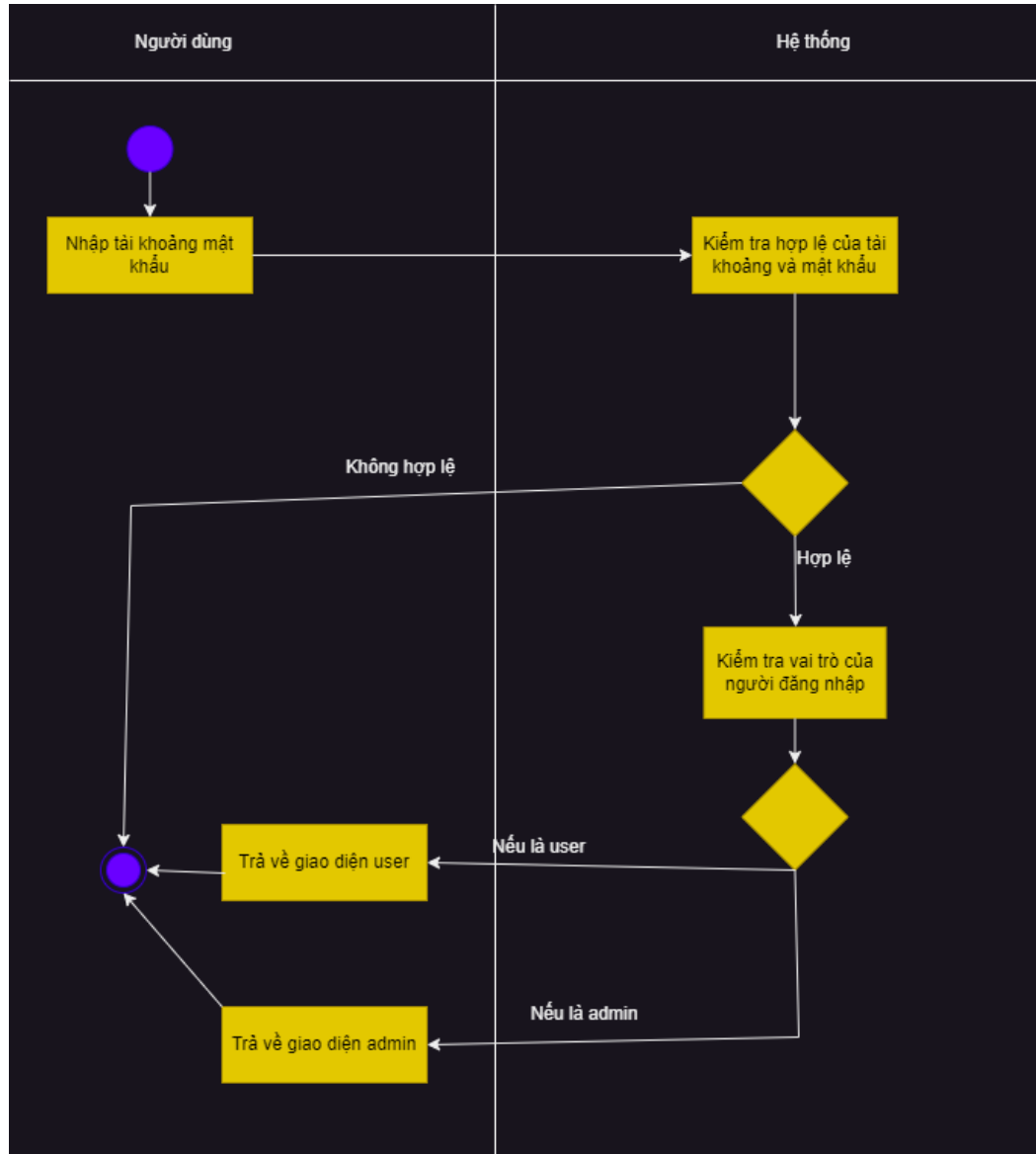
## 4.4 Sơ đồ hoạt động

- Quy trình đăng ký:



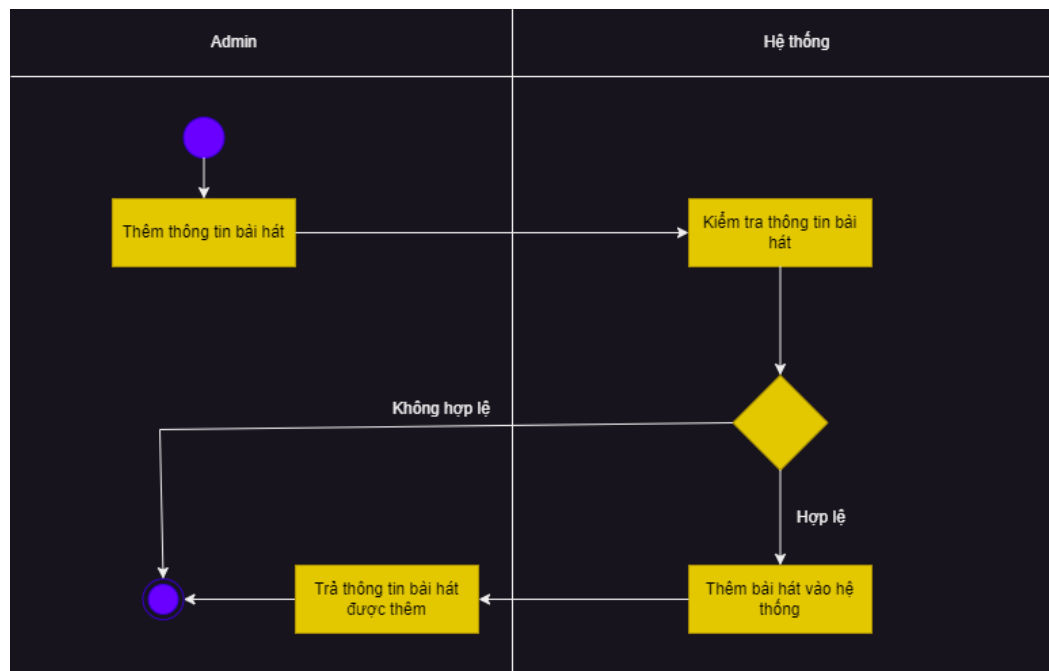
Hình 4. 4: Sơ đồ hoạt động đăng ký

- Quy trình đăng nhập:



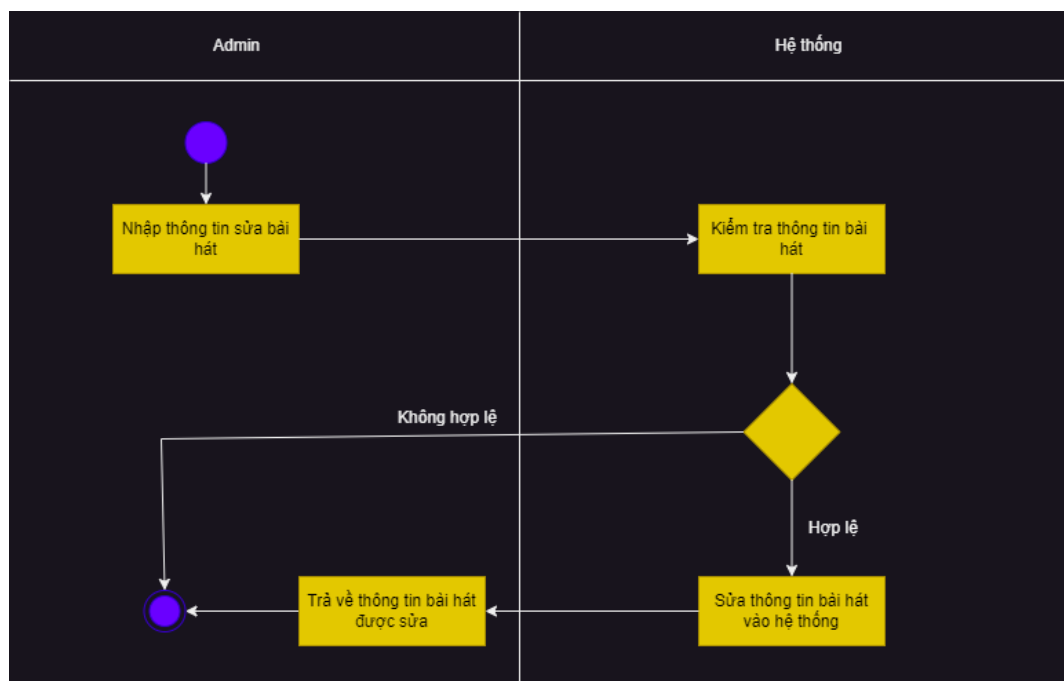
Hình 4. 5: Sơ đồ hoạt động đăng nhập

- Quy trình thêm bài hát:



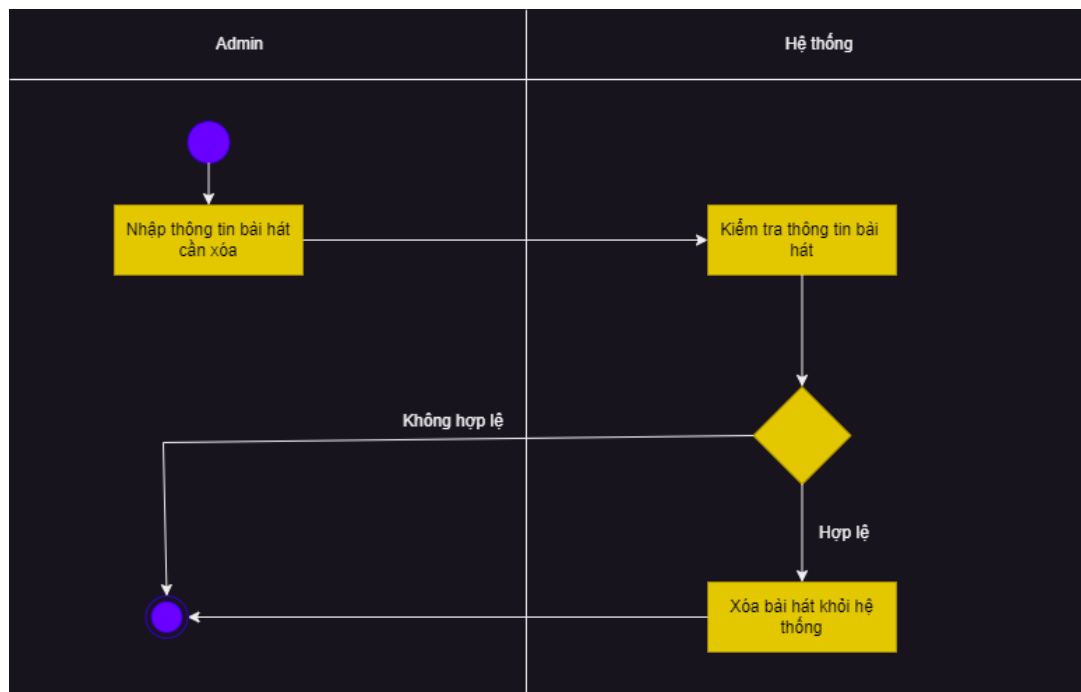
Hình 4. 6: Sơ đồ hoạt động thêm bài hát

- Quy trình sửa bài hát:



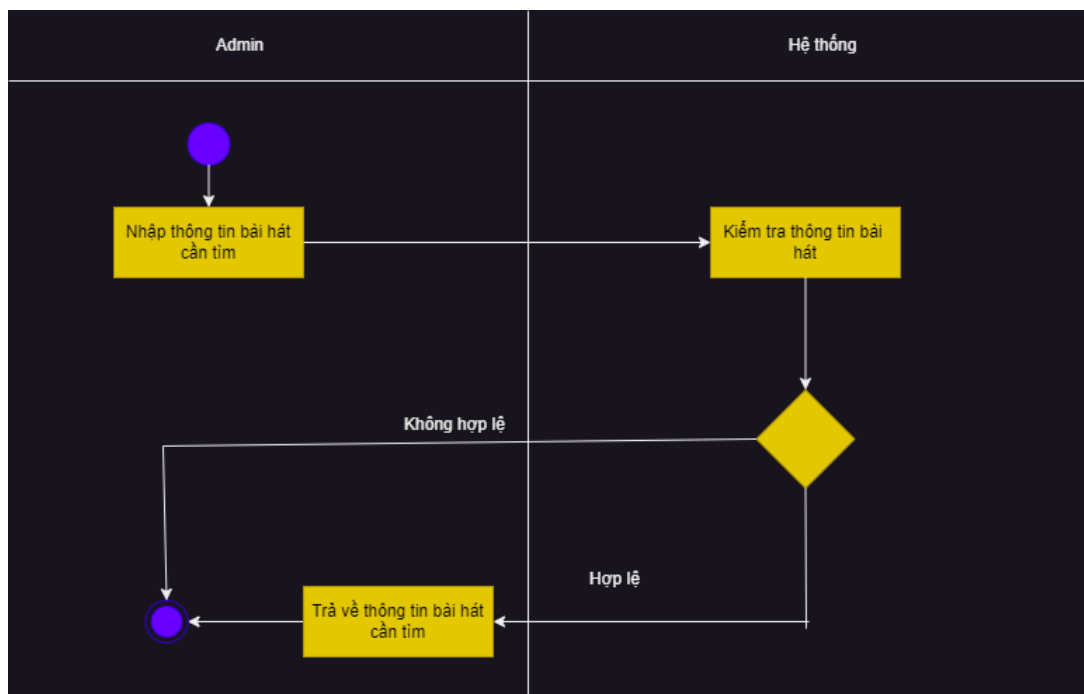
Hình 4. 7: Sơ đồ hoạt động sửa bài hát

- Quy trình xóa bài hát:



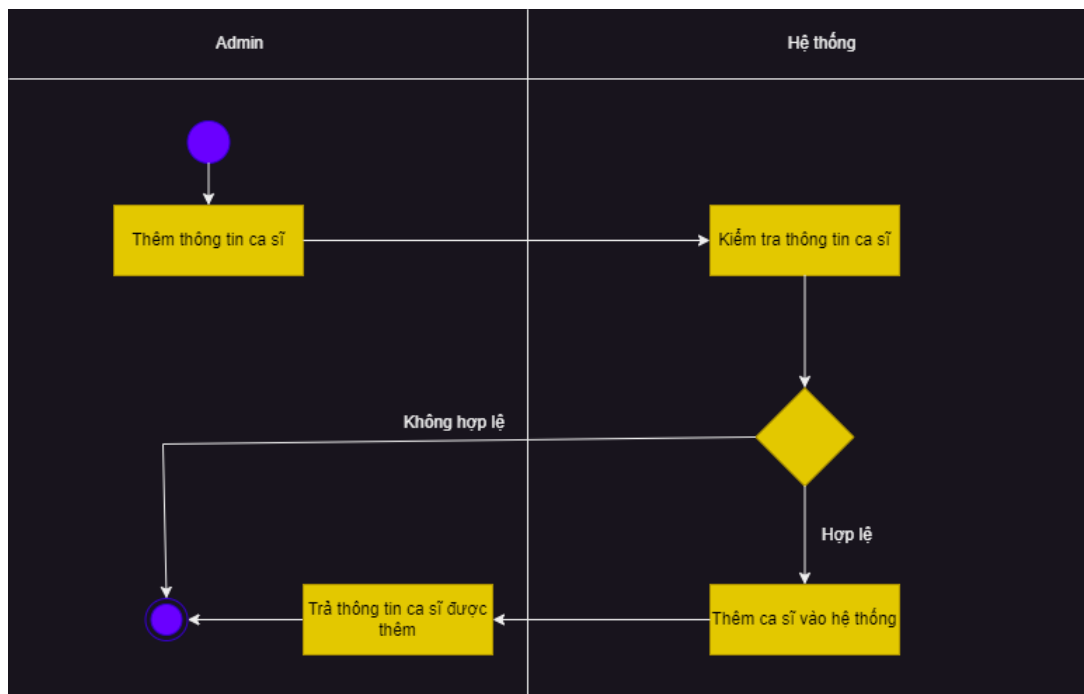
Hình 4. 8: Sơ đồ hoạt động xóa bài hát

- Quy trình tìm kiếm bài hát:



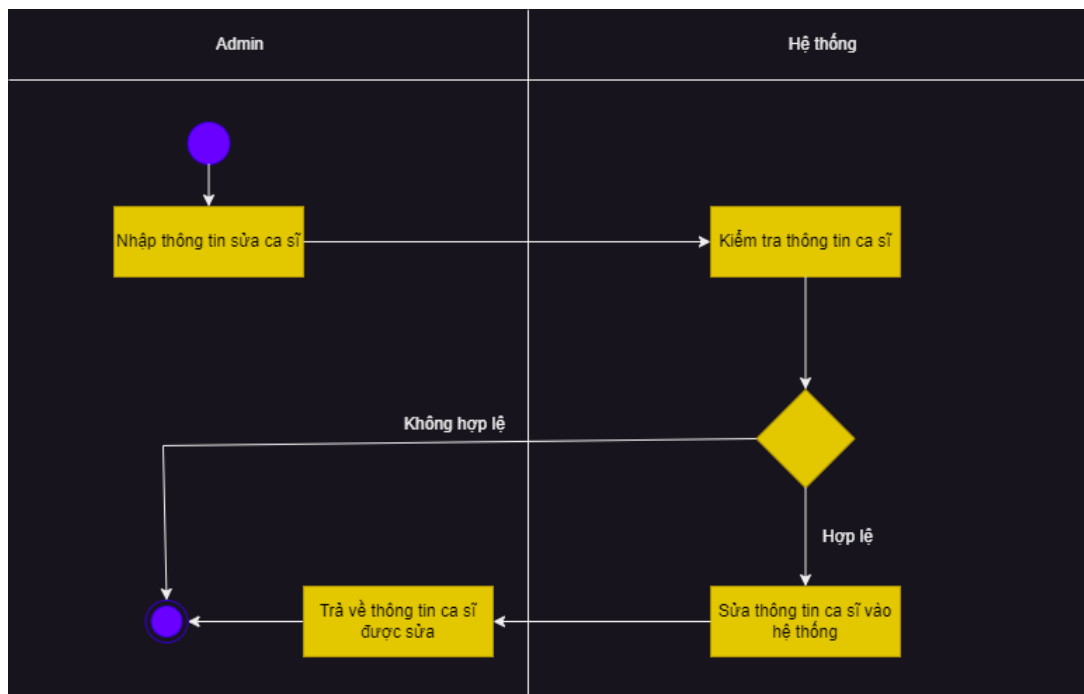
Hình 4. 9: Sơ đồ hoạt động tìm kiếm bài hát

-Quy trình thêm ca sĩ:



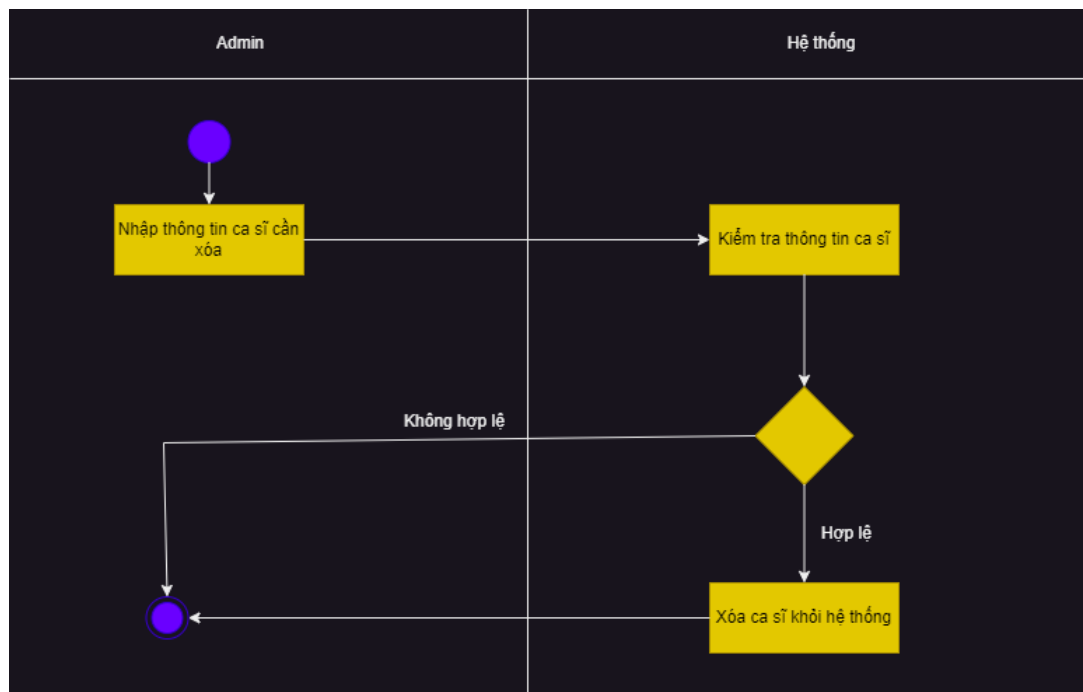
Hình 4. 10: Sơ đồ hoạt động thêm ca sĩ

- Quy trình sửa ca sĩ:



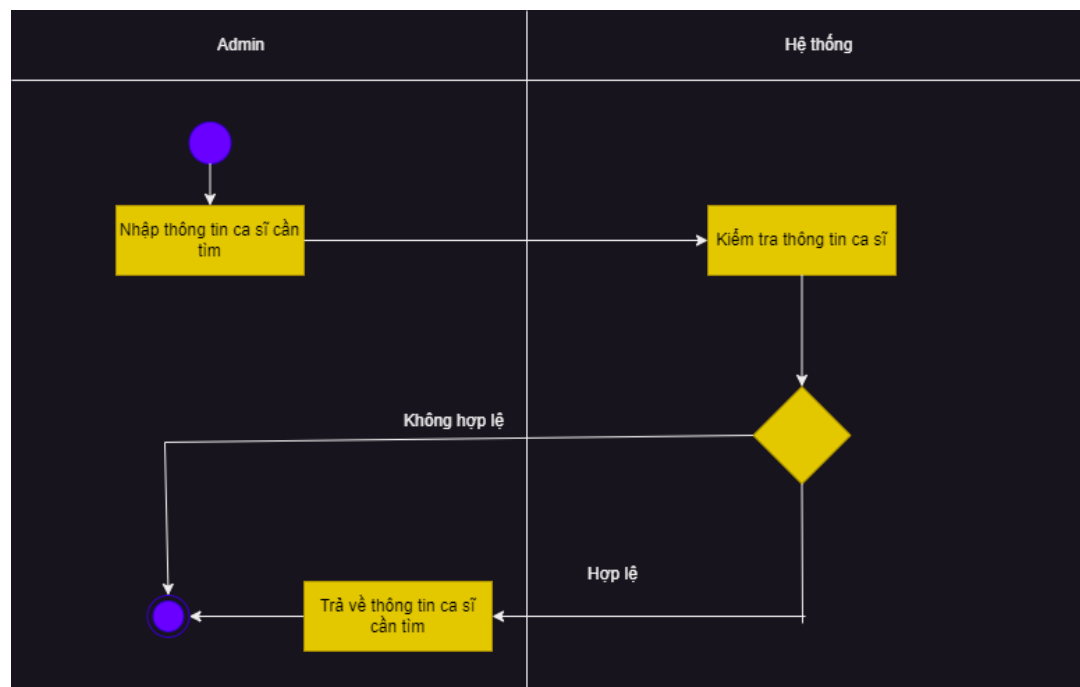
Hình 4. 11: Sơ đồ hoạt động sửa ca sĩ

- Quy trình xóa ca sĩ:



Hình 4. 12: Sơ đồ hoạt động xóa ca sĩ

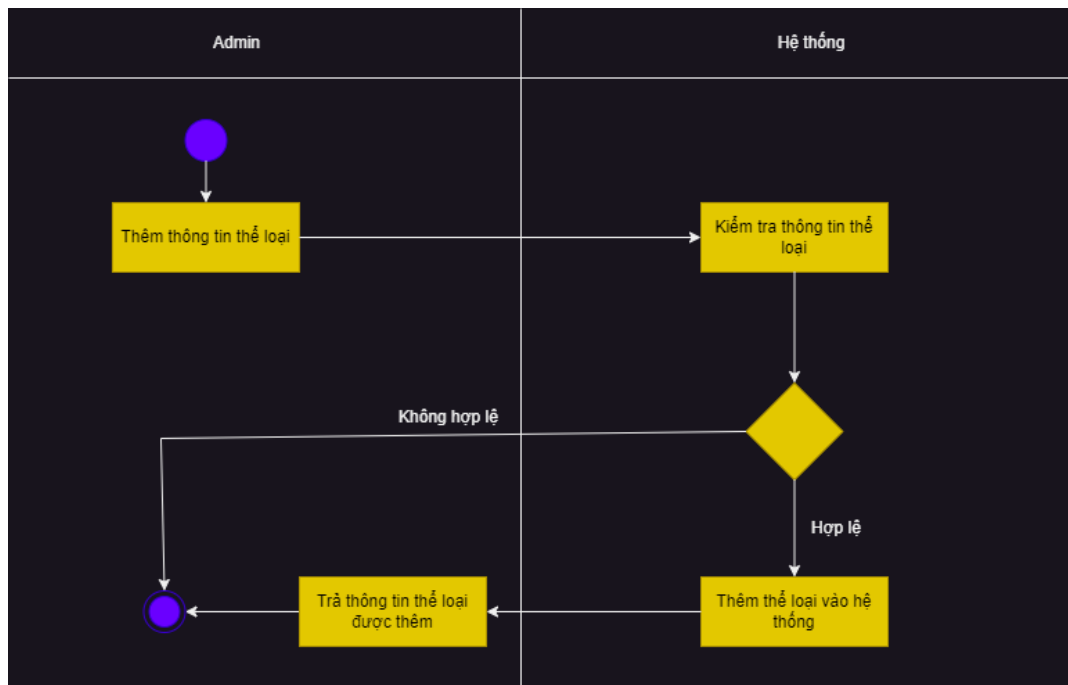
- Quy trình tìm ca sĩ:



Hình 4. 13: Sơ đồ hoạt động tìm ca sĩ

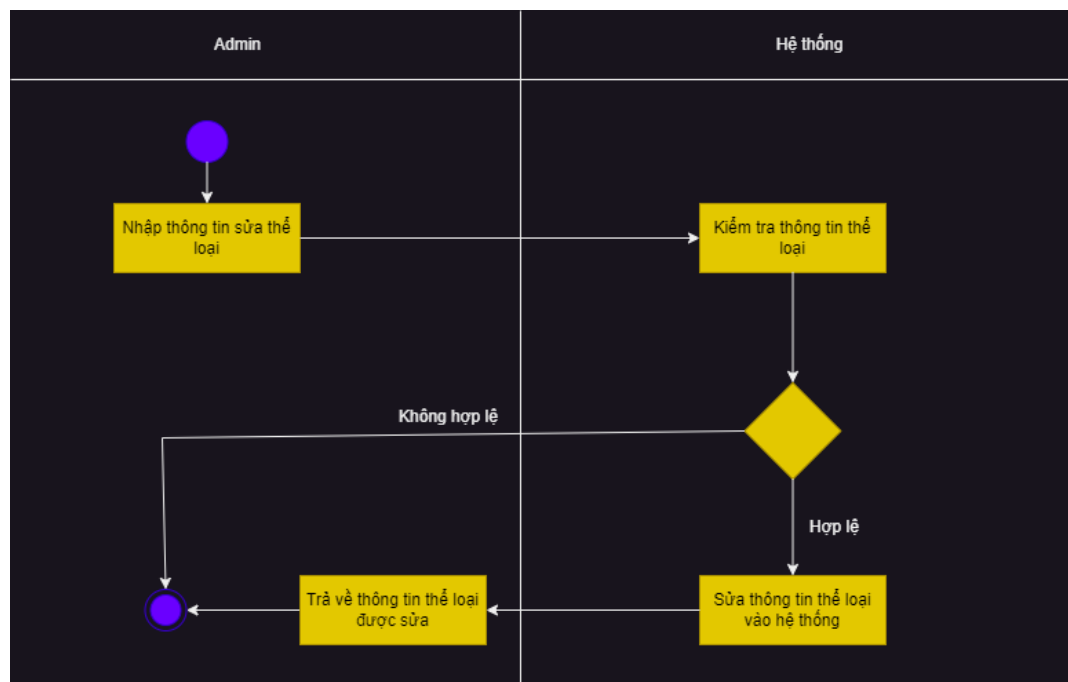


- Quy trình thêm thẻ loại:



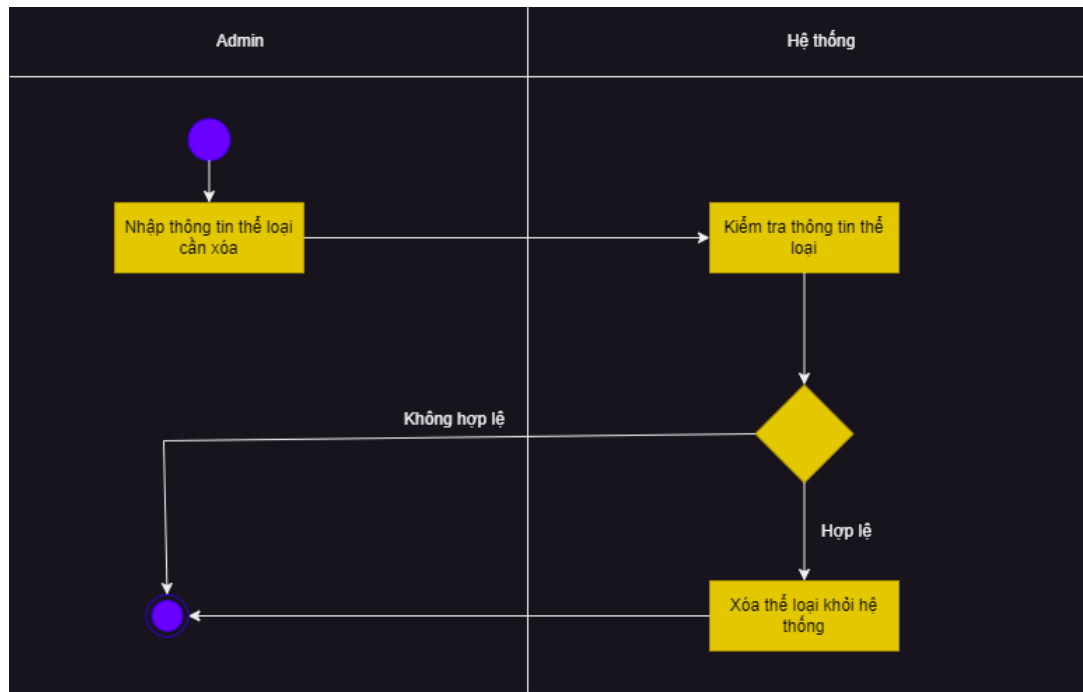
Hình 4. 14: Sơ đồ hoạt động thêm thẻ loại

-Quy trình sửa thẻ loại:



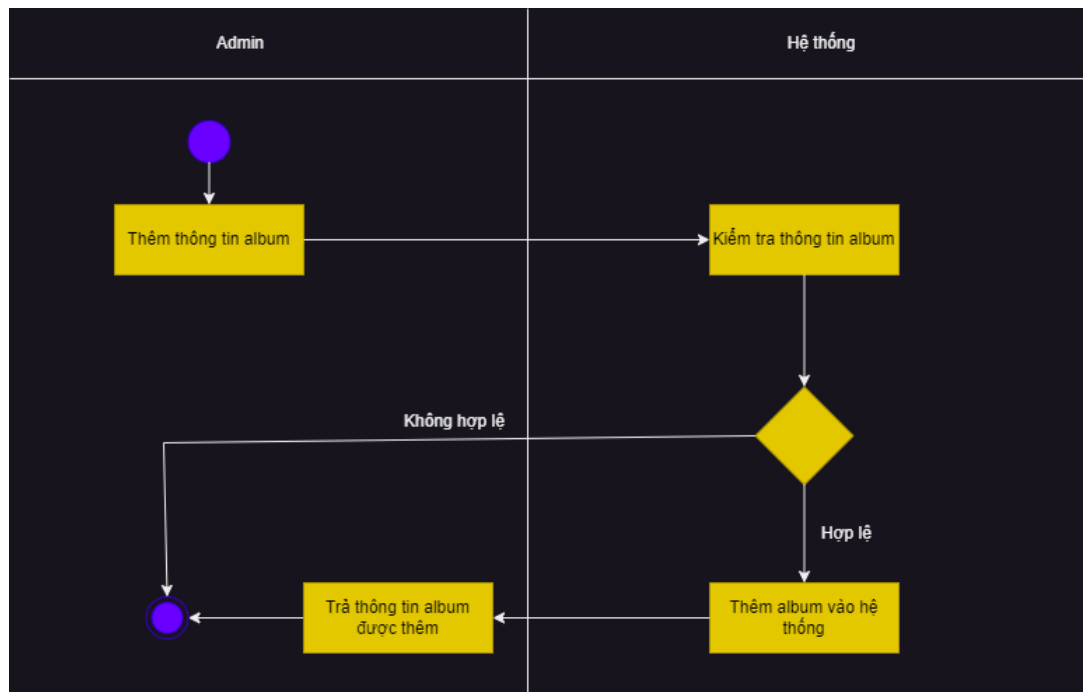
Hình 4. 15: Sơ đồ hoạt động sửa thẻ loại

- Quy trình xóa thể loại:



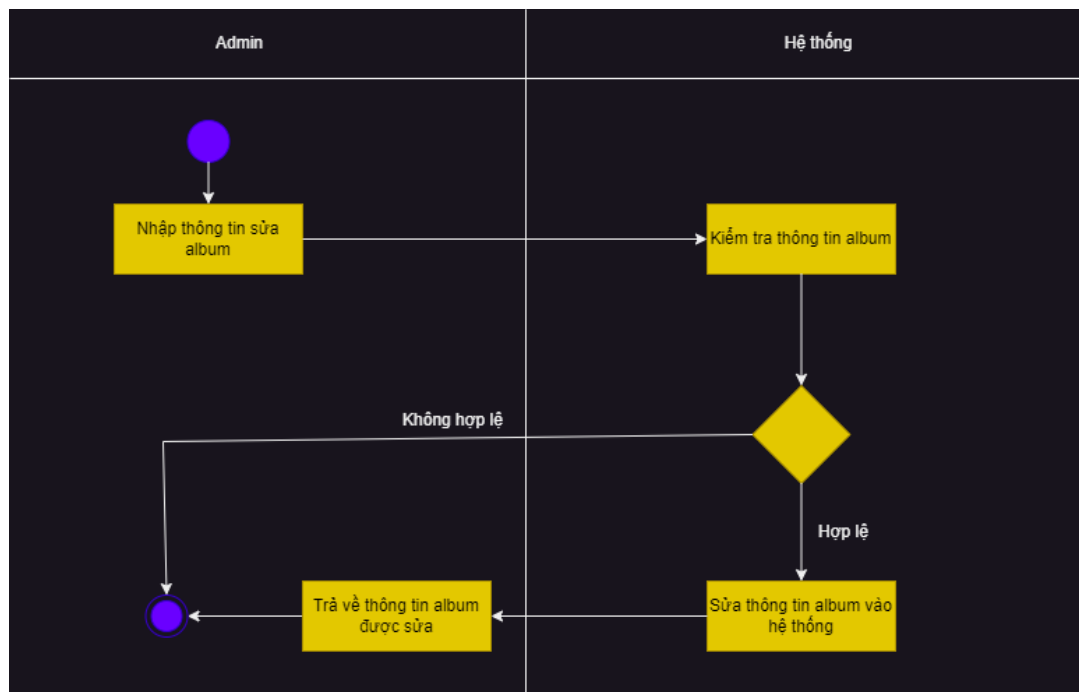
Hình 4. 16: Sơ đồ hoạt động xóa thể loại

- Quy trình thêm album:



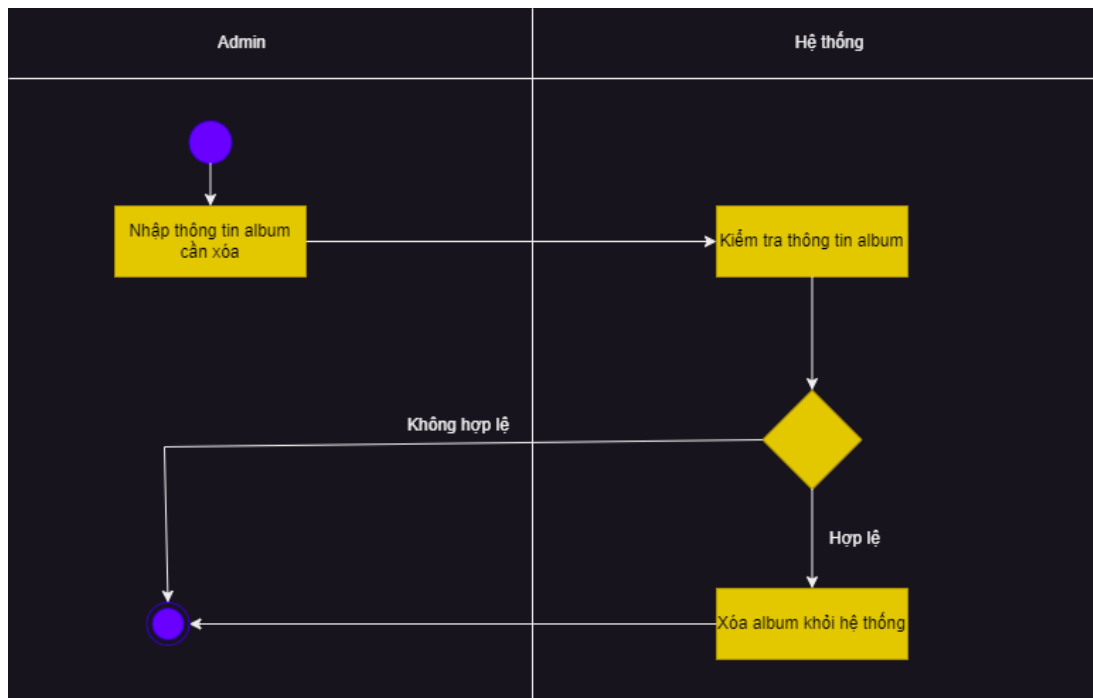
Hình 4. 17: Sơ đồ hoạt động thêm album

- Quy trình sửa album:



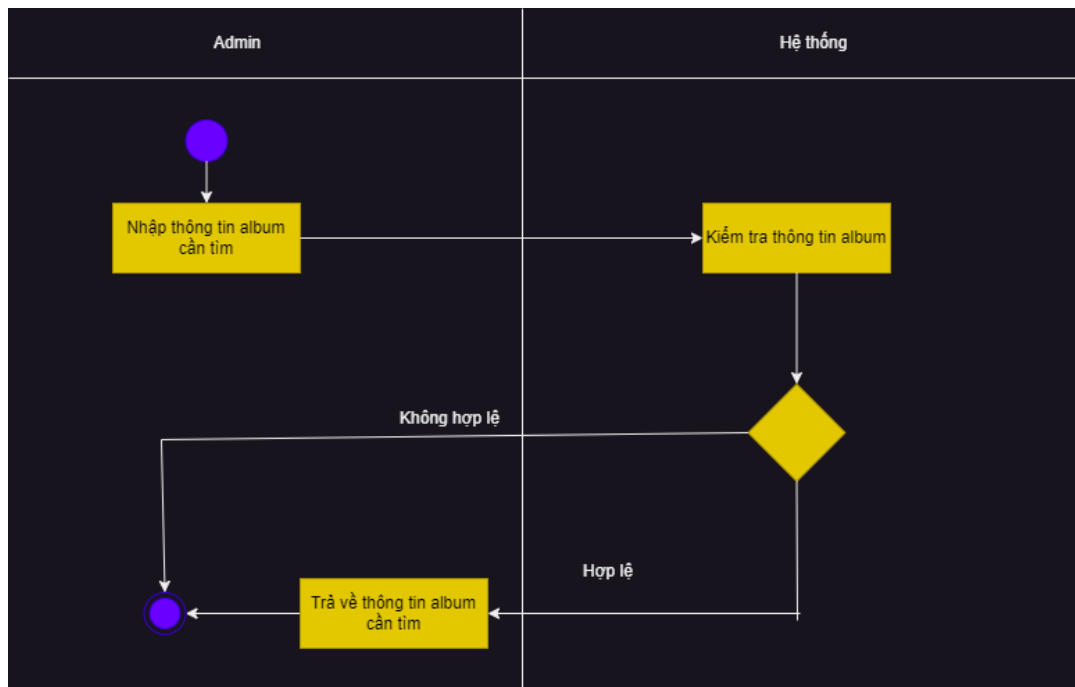
Hình 4. 18: Sơ đồ hoạt động sửa album

- Quy trình xóa album:



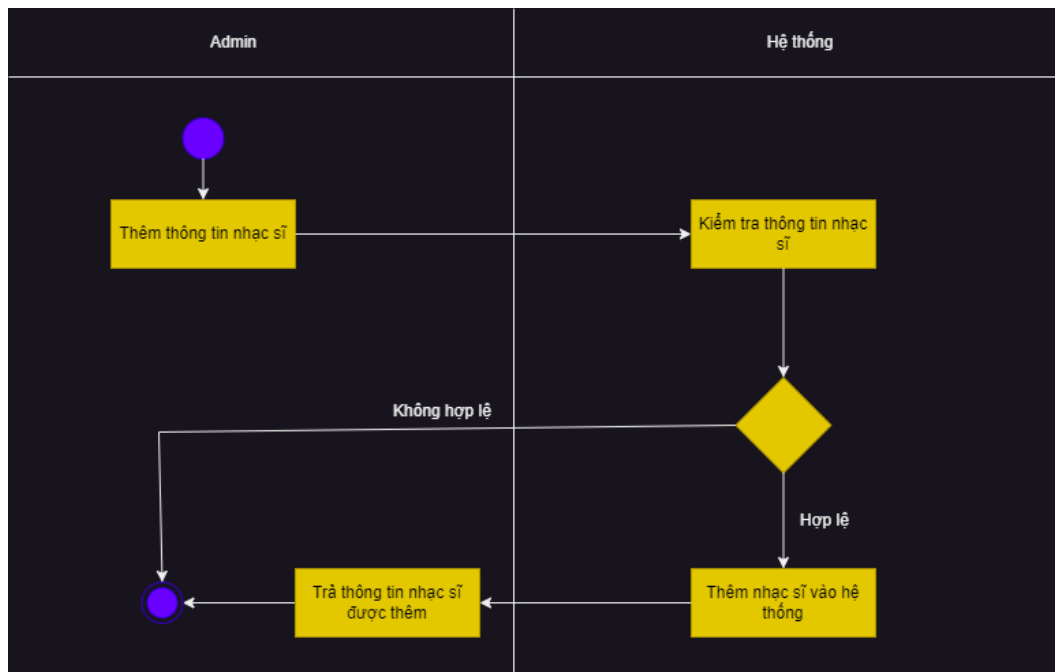
Hình 4. 19: Sơ đồ hoạt động xóa album

- Quy trình tìm album:



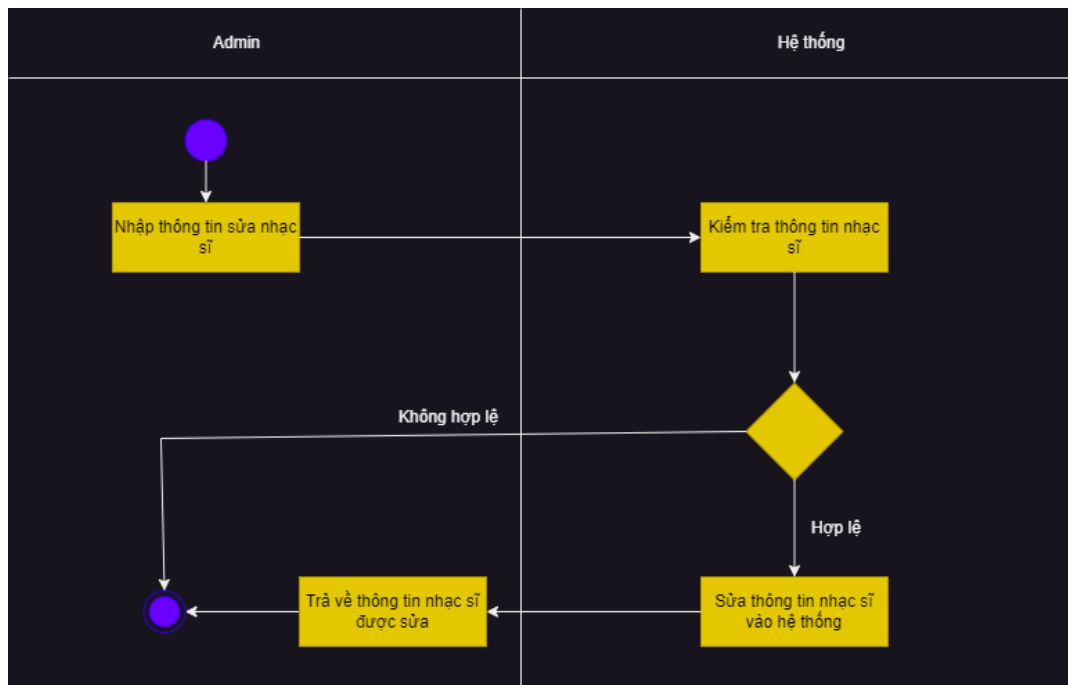
Hình 4. 20: Sơ đồ hoạt động tìm album

- Quy trình thêm nhạc sĩ:



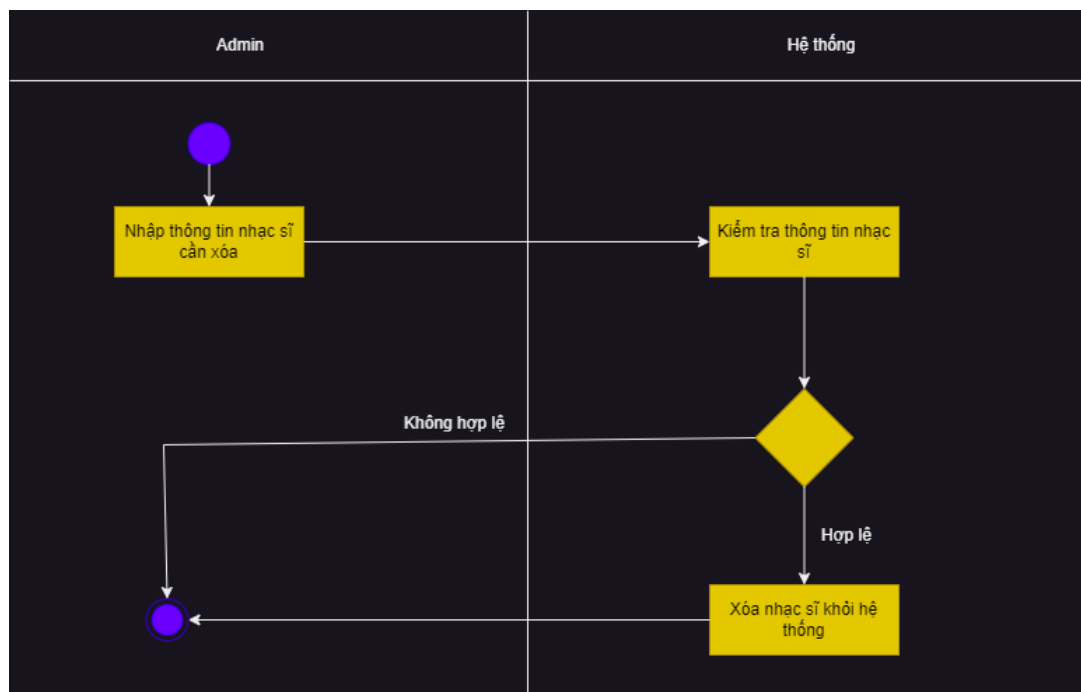
Hình 4. 21: Sơ đồ hoạt động thêm ca sĩ

- Quy trình sửa nhạc sĩ:



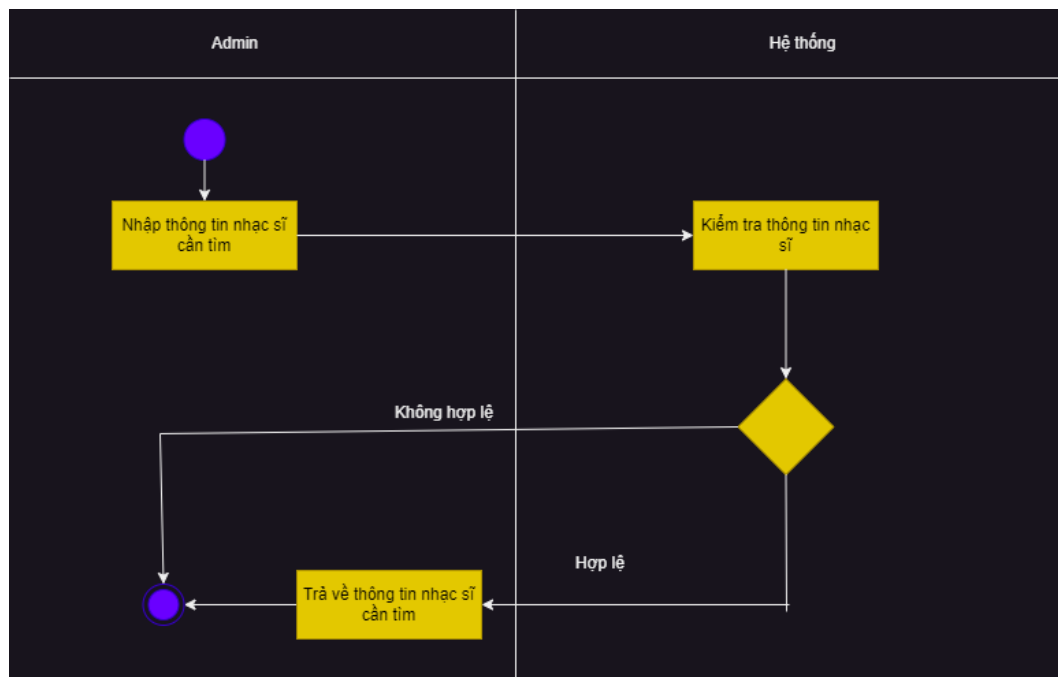
Hình 4. 22: Sơ đồ hoạt động sửa ca sĩ

-Quy trình xóa nhạc sĩ:



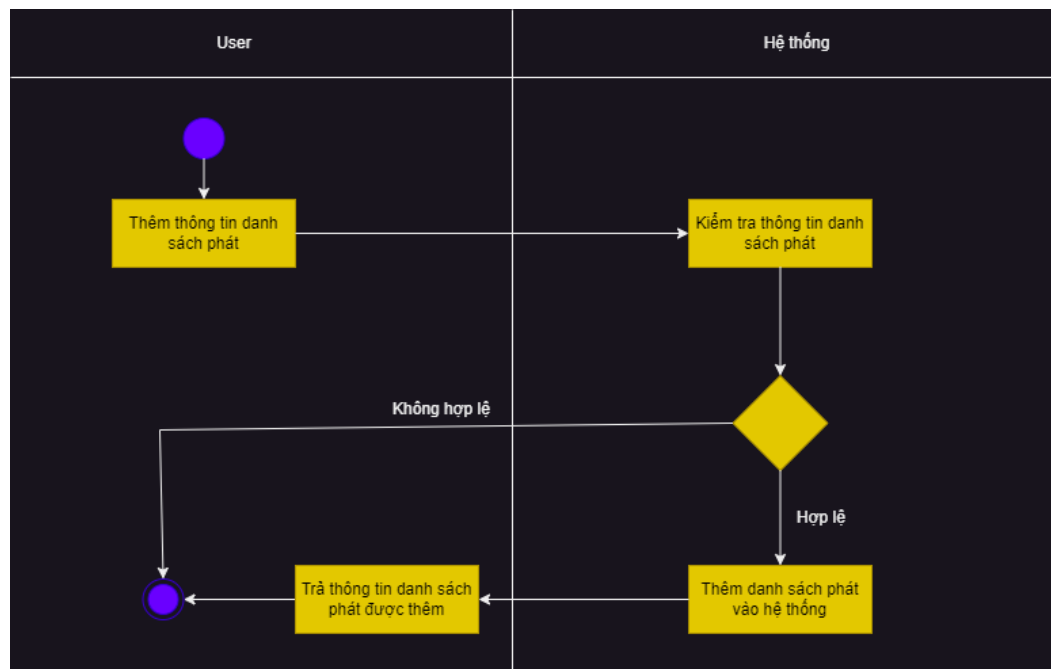
Hình 4. 23: Sơ đồ hoạt động xóa ca sĩ

- Quy trình tìm nhạc sĩ:



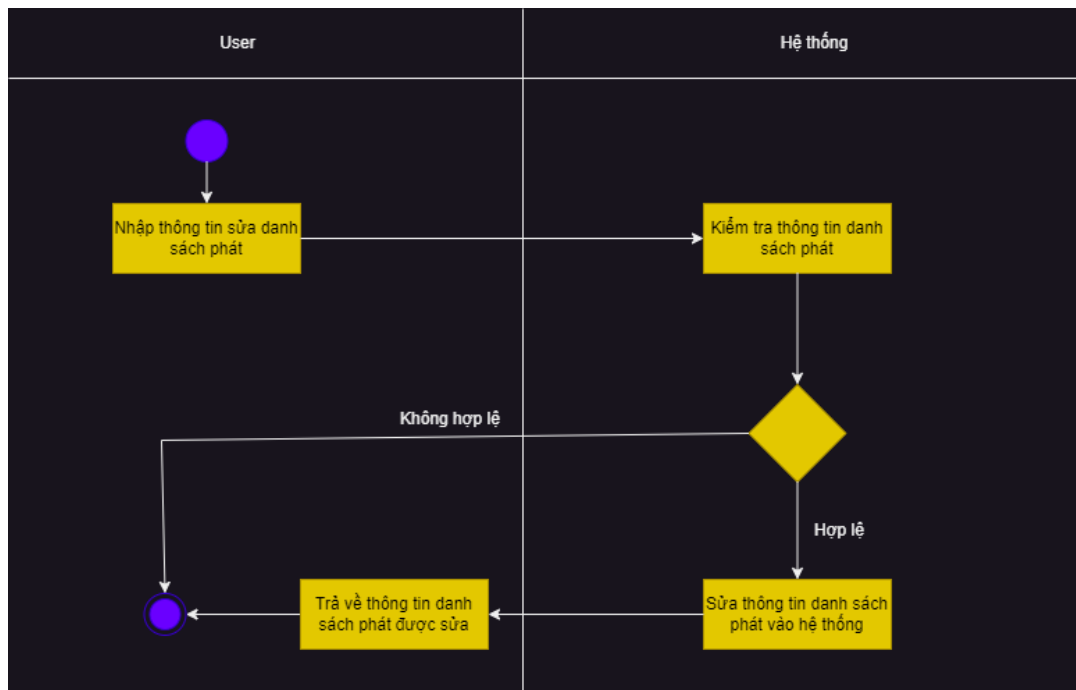
Hình 4. 24: Sơ đồ hoạt động tìm nhạc sĩ

- Quy trình thêm danh sách phát của user:



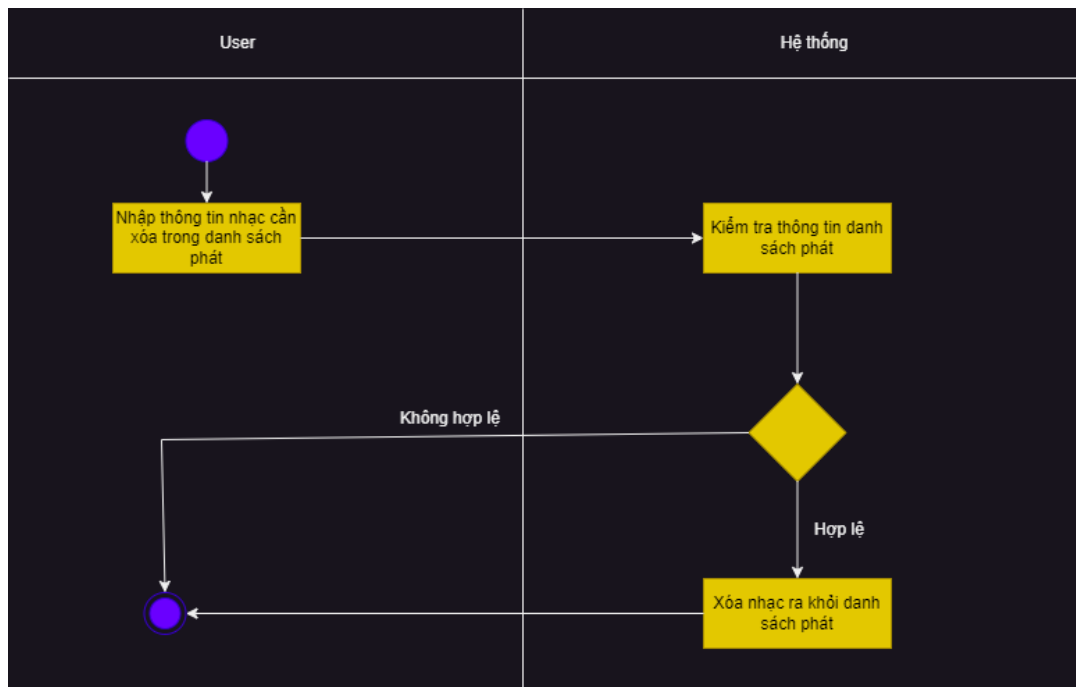
Hình 4. 25: Sơ đồ hoạt động thêm danh sách phát của user

- Quy trình sửa danh sách phát của user:



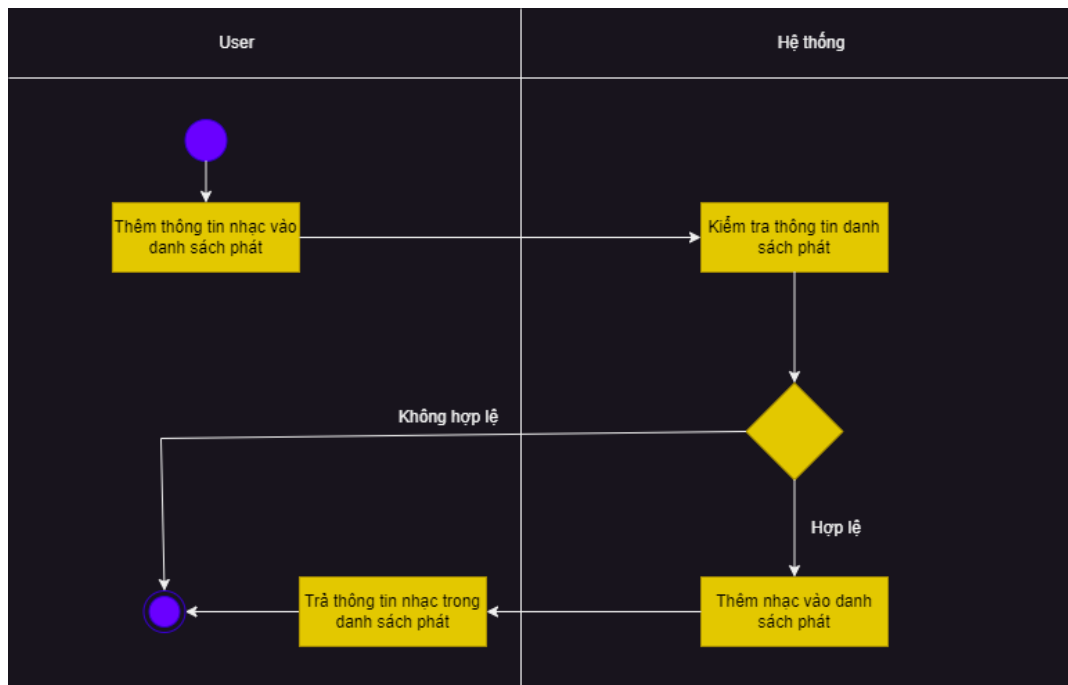
Hình 4. 26: Sơ đồ hoạt động sửa danh sách phát của user

-Quy trình xóa danh sách phát của user:



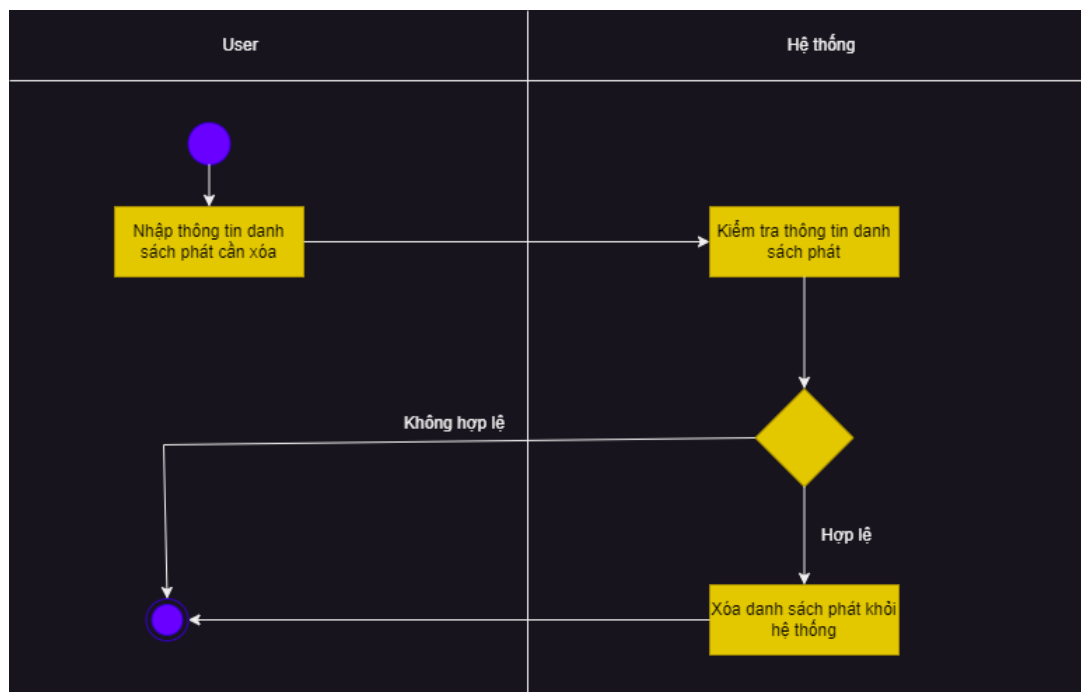
Hình 4. 27: Sơ đồ hoạt động xóa danh sách phát của user

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



Hình 4. 28: Sơ đồ hoạt động thêm nhạc vào danh sách phát của user

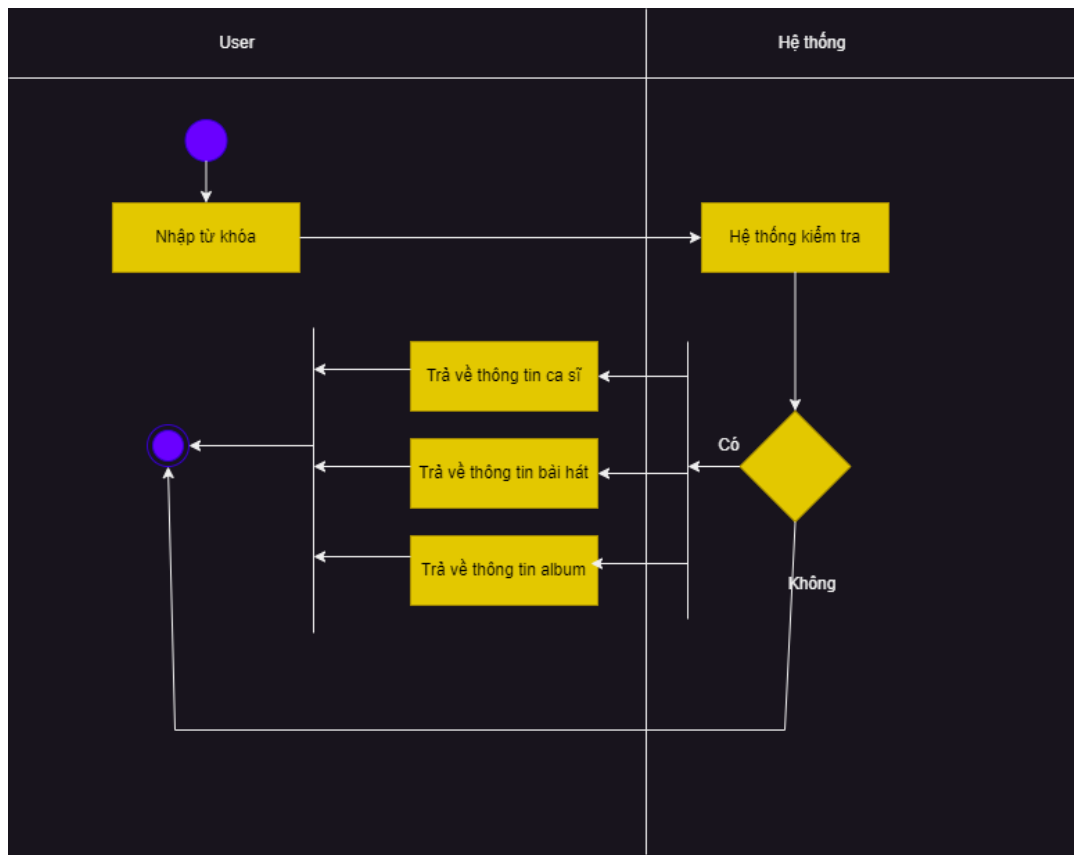
-Quy trình xóa nhạc khỏi danh sách phát của user:



Hình 4. 29: Sơ đồ hoạt động xóa nhạc khỏi danh sách phát của user

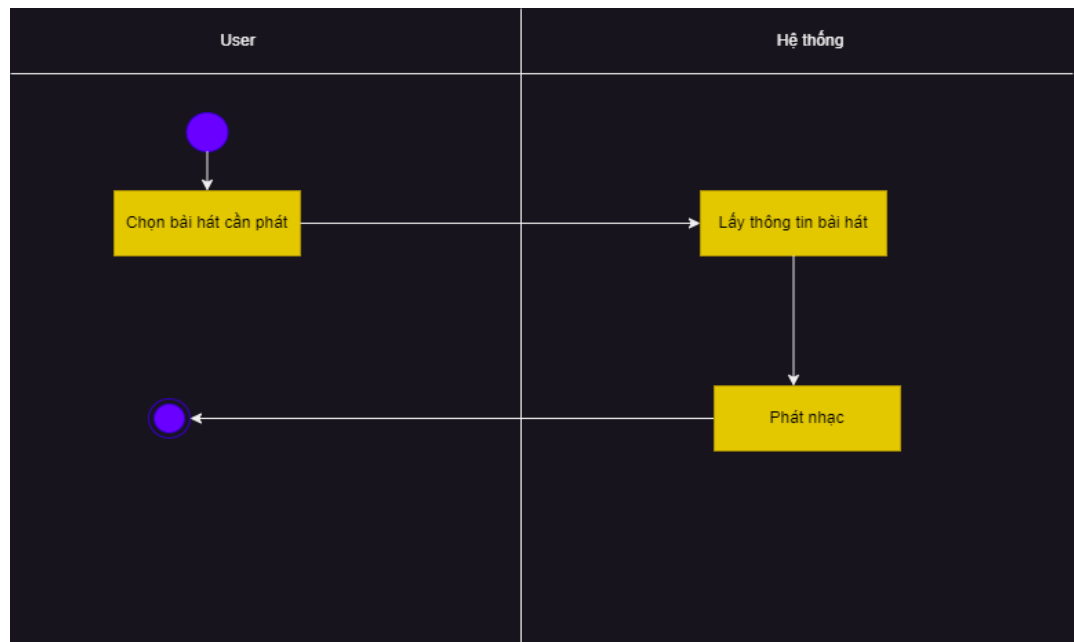


- Quy trình tìm nhạc, album, ca sĩ của user:



Hình 4. 30: Sơ đồ hoạt động tìm kiếm của user

- Quy trình nghe nhạc của user:



Hình 4. 31: Sơ đồ hoạt động nghe nhạc của user

## CHƯƠNG 5. THIẾT KẾ CHƯƠNG TRÌNH

### 5.1 Thiết kế cơ sở dữ liệu

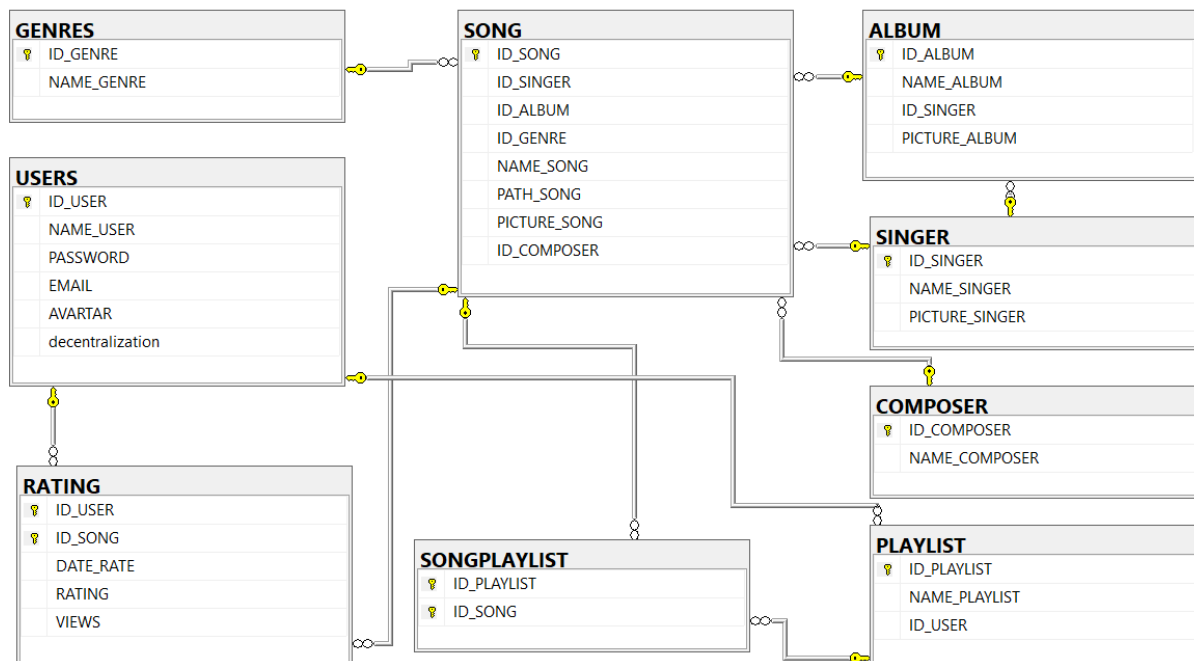
#### 5.1.1 Mô hình quan hệ thực thể (mô hình ER)



Hình 5. 1: Mô hình ER

### 5.1.2 Biểu đồ các bảng trong cơ sở dữ liệu

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



Hình 5. 2: Các bảng trong cơ sở dữ liệu

## 5.2 Thiết kế hệ thống gợi ý

### 5.2.1 Ý tưởng

- 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. Ta xây dựng ma trận để 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, ta thực hiện chuẩn hóa bằng công thức  $1/\sqrt{3}$ . Sau khi chuẩn hóa, ta có các hàng là vector đặc trưng của từng bài hát (item). Ta 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

*Độ tương đồng cosin = tích vô hướng / 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:

*Độ ưa thích = thời lượng nghe bài hát / thời lượng bài hát*

- Từ giá trị ưa thích của người dùng, ta 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  $w_i = a_i \cdot \text{giá trị chuẩn hóa của thuộc tính đó}$ , tương tự có n bài hát thì ta thực hiện ta cộng  $w_i$  của n bài hát ta sẽ có được độ ưa thích của thuộc tính  $w_n$ . Các thuộc tính còn lại được tính tương tự ta được vector độ ưa thích của user.

### 5.2.2 Hệ thống gợi ý dựa vào nội dung (content-based systems)

- Khai báo các giá trị cần sử dụng:

```
//ma trận
private static DataTable matric = new DataTable();
//tên cột
private static string[] NameColumn;
//danh sách dữ liệu
private static List<CustomSong> lstCusSong;
//số cột
static int countColumn = 0;
//số hàng
static int countRow = 0;
//mảng chứa giá trị đánh giá của người dùng
private static double[] rating ;
// mảng chứa giá trị là vector người dùng
private static double[] userProfile;
```

-Hàm lấy danh sách đánh giá của người dùng:

```
//Lấy danh sách bài hát được người dùng đánh giá
static List<SONG> getSongRating(int id )
{
    //Gia su user id=2
    Web_musicEntities conn = new Web_musicEntities();
    List<SONG> lstSong = new List<SONG>();
    List<RATING> lstRating1 = conn.RATINGs.ToList();
    List<RATING> lstRating2 = new List<RATING>();
    foreach (RATING a in lstRating1)
    {
        if (a.ID_USER == id)
        {
            lstRating2.Add(a);
            SONG song = conn.SONGs.Find(a.ID_SONG);
            lstSong.Add(song);
        }
    }
    rating = new double[lstRating2.Count];
    for(int i=0; i<lstRating2.Count; i++)
    {
        rating[i] =Convert.ToDouble( lstRating2[i].RATING1);
    }
    return lstSong;
}
```

- Hàm lấy danh sách bài hát, dữ liệu cột và hàng:

```
//Hàm lấy dữ liệu từ database và thêm column
static void getData(List<SONG> lstSong)
{
    //lấy danh sách
    Web_musicEntities conn = new Web_musicEntities();
    List<GENRE> lstGenre = conn.GENRES.ToList();
    List<COMPOSER> lstComposer = conn.COMPOSERs.ToList();
    List<SINGER> lstSinger = conn.SINGERs.ToList();
    //xử lý dữ liệu bài hát
    lstCusSong = new List<CustomSong>();
    foreach(SONG a in lstSong)
    {
        CustomSong cus = new CustomSong();
        cus.id = a.ID_SONG;
        cus.name = a.NAME_SONG;
        cus.image = a.PICTURE_SONG;
        cus.song = a.PATH_SONG;
        if (a.ID_ALBUM != null)
        {
            cus.idalbum = Convert.ToInt32(a.ID_ALBUM);
            cus.tenalbum = conn.ALBUMs.Find(a.ID_ALBUM).NAME_ALBUM;
        }
        else
        {
            cus.tenalbum = "none";
        }
    }
}
```

```

        cus.idcasi = Convert.ToInt32(a.ID_SINGER);
        cus.idtheloai = Convert.ToInt32(a.ID_GENRE);
        cus.tencasi = "CS"+conn.SINGERS.Find(a.ID_SINGER).NAME_SINGER;
        cus.tentheloai = "TL"+ conn.GENRES.Find(a.ID_GENRE).NAME_GENRE;
        cus.idtacgia = Convert.ToInt32(a.ID_COMPOSER);
        cus.tentacgia = "TG"+ conn.COMPOSERS.Find(a.ID_COMPOSER).NAME_COMPOSER;
        lstCusSong.Add(cus);
    }
    List<string> nameComposer = new List<string>();
    foreach (COMPOSER a in lstComposer)
    {
        string name = a.NAME_COMPOSER;
        nameComposer.Add(name);
    }
    List<string> nameGenre = new List<string>();
    foreach (GENRE a in lstGenre)
    {
        string name = a.NAME_GENRE;
        nameGenre.Add(name);
    }
    List<string> nameSinger = new List<string>();
    foreach (SINGER a in lstSinger)
    {
        string name = a.NAME_SINGER;
        nameSinger.Add(name);
    }
    //Thực hiện tính số cột, số hàng, tên cột
    nameComposer = KtraTrung(nameComposer);
    nameGenre = KtraTrung(nameGenre);
    nameSinger = KtraTrung(nameSinger);
    countColumn = nameSinger.Count() + nameGenre.Count + nameComposer.Count()
+ 1;

    userProfile = new double[countColumn];
    countRow = lstCusSong.Count();
    NameColumn = new string[countColumn];
    NameColumn[0] = "ID";
    int i = 1;
    foreach(string a in nameGenre)
    {
        NameColumn[i] = "TL" + a;
        i = i + 1;
    }
    foreach (string a in nameSinger)
    {
        NameColumn[i] = "CS" + a;
        i = i + 1;
    }
    foreach (string a in nameComposer)
    {
        NameColumn[i] = "TG" + a;
        i = i + 1;
    }
}

```

-Hàm kiểm tra trùng lặp của các cột:

```
// hàm kiểm tra trùng của cột matrix
static List<String> KtraTrung(List<String> lst)
{
    List<String> newlist = new List<string>();
    newlist = lst.Distinct().ToList();
    return newlist;
}
```

-Khởi tạo ma trận với ba hàm:

```
// Hàm tạo cột cho ma trận
static void CreateColumnMatric(string nameColumn)
{
    DataColumn dtColumn = new DataColumn();
    dtColumn.DataType = typeof(double);
    dtColumn.ColumnName = nameColumn;
    matrix.Columns.Add(dtColumn);
}
// hàm gán cột cho ma trận
static void addColumn()
{
    for(int i=0; i<countColumn; i++)
    {
        CreateColumnMatric(NameColumn[i]);
    }
}
// Hàm thêm dòng cho ma trận
static void addRow()
{
    foreach(CustomSong a in lstCusSong)
    {
        DataRow row = matrix.NewRow();
        row["ID"] = a.id;
        for(int i=1; i<countColumn; i++)
        {
            if(matrix.Columns[i].ColumnName.Equals(a.tencasi) ||
matrix.Columns[i].ColumnName.Equals(a.tentacgia) ||
matrix.Columns[i].ColumnName.Equals(a.tentheloai))
            {
                row[NameColumn[i]] = 1;
            }
            else
            {
                row[NameColumn[i]] = 0;
            }
        }
        matrix.Rows.Add(row);
    }
}
```

-Hàm chuẩn hóa các thuộc tính:

```
//Hàm chuẩn hóa các giá trị thuộc tính
static void Chuanhoa()
{
    for(int j=0; j<countRow; j++)
    {
        int Total = 0;
        for (int i = 1; i < countColumn; i++)
        {
            if (Convert.ToInt32(matric.Rows[j].ItemArray[i]) == 1)
            {
                Total = Total + 1;
            }
        }
        for (int i = 1; i < countColumn; i++)
        {
            if (Convert.ToInt32(matric.Rows[j].ItemArray[i]) == 1)
            {
                double Gia_tri_da_chuan_hoa = Convert.ToDouble(1 /
Math.Sqrt(Total));

                matric.Rows[j][NameColumn[i]] = Gia_tri_da_chuan_hoa;
                matric.AcceptChanges();
            }
        }
    }
}
```

-Hàm tính cosin của bài hát đầu tiên với các bài hát còn lại

```
// Hàm tính giá trị cosin
static void Cosin()
{
    double[] gia_tri_cosin = new double[countRow];
    for (int i = 1; i < countRow; i++)
    {
        for (int j = 1; j < countColumn; j++)
        {
            double giatri1 = Convert.ToDouble(matric.Rows[0].ItemArray[j]);
            double giatri2 = Convert.ToDouble(matric.Rows[i].ItemArray[j]);
            gia_tri_cosin[i] = gia_tri_cosin[i] + Convert.ToDouble(giatri1 *
giatri2);
        }
    }
    for (int i = 0; i < countRow; i++)
    {
        Console.WriteLine(gia_tri_cosin[i]);
    }
}
```



## - Hàm tính vector của người dùng

```
// Hàm tính vector người dùng
static void AddValueUserProfile()
{
    {
        for (int i = 1; i < userProfile.Length; i++)
        {
            double giatri = 0;
            int df = 0;
            for (int j = 0; j < rating.Length ; j++)
            {
                {
                    double giatri1 = Convert.ToDouble(rating[j]);
                    double giatri2 =
Convert.ToDouble(matric.Rows[j].ItemArray.GetValue(i));
                    giatri = giatri + Convert.ToDouble(giatri1 * giatri2);
                    if (giatri2 != 0)
                    {
                        df = df + 1;
                    }
                }
            }
            userProfile[i] = giatri;
        }
    }
}
```

## -Hàm tính độ ưa thích của người dùng

```
// Tính độ ưa thích của người dùng với bài hát
static void TinhDoUaThich()
{
    for (int i = 0; i < rating.Length; i++)
    {
        double giatri = 0;
        for (int j = 1; j < userProfile.Length; j++)
        {
            giatri = giatri +
(Convert.ToDouble(matric.Rows[i].ItemArray.GetValue(j)) * userProfile[j]);
        }
        rating[i] = giatri;
    }
}
```

### 5.2.3 Hệ thống gợi ý dựa vào lọc cộng tác (collaborative systems)

#### - Sử dụng ML.Net để train mô hình và dự đoán

- Hàm lấy dữ liệu:

```
static (IDataView training, IDataView test) LoadData(MLContext mlContext)
{
    DatabaseLoader loader = mlContext.Data.CreateDatabaseLoader<ModelInput>();
    DatabaseSource dbSource1 = new DatabaseSource(SqlClientFactory.Instance,
connectionString, sqlCommand1);
    DatabaseSource dbSource2 = new DatabaseSource(SqlClientFactory.Instance,
connectionString, sqlCommand1);
    var data = loader.Load(dbSource1);
    DataOperationsCatalog.TrainTestData dataSlipt =
mlContext.Data.TrainTestSplit(data, testFraction: 0.2);
    IDataView trainingDataView = dataSlipt.TrainSet;
    IDataView testDataView = dataSlipt.TestSet;

    return (trainingDataView, testDataView);
}
```

- Hàm build mô hình:

```
static ITransformer BuildAndTrainModel(MLContext mlContext, IDataView
trainingDataView)
{
    IEstimator<ITransformer> estimator =
mlContext.Transforms.Conversion.MapValueToKey(outputColumnName: "userIdEncoded",
inputColumnName: "ID_USER")
    .Append(mlContext.Transforms.Conversion.MapValueToKey(outputColumnName:
"movieIdEncoded", inputColumnName: "ID_SONG"));
    var options = new MatrixFactorizationTrainer.Options
    {
        MatrixColumnIndexColumnName = "userIdEncoded",
        MatrixRowIndexColumnName = "movieIdEncoded",
        LabelColumnName = "RATING",
        NumberOfIterations = 20,
        ApproximationRank = 100
    };

    var trainerEstimator =
estimator.Append(mlContext.Recommendation().Trainers.MatrixFactorization(options));
    Console.WriteLine("===== Training the model =====");
    ITransformer model = trainerEstimator.Fit(trainingDataView);

    return model;
}
```

## -Hàm đánh giá mô hình

```
static void EvaluateModel(MLContext mlContext, IDataView testDataView,
ITransformer model)
{
    Console.WriteLine("===== Evaluating the model =====");
    var prediction = model.Transform(testDataView);
    var metrics = mlContext.Regression.Evaluate(prediction, labelColumnName:
"RATING", scoreColumnName: "Score");
    Console.WriteLine("Root Mean Squared Error : " +
metrics.RootMeanSquaredError.ToString());
    Console.WriteLine("RSquared: " + metrics.RSquared.ToString());
}
```

## -Hàm sử dụng mô hình:

```
static void UseModelForSinglePrediction(MLContext mlContext, ITransformer
model)
{
    Console.WriteLine("===== Making a prediction =====");
    var predictionEngine = mlContext.Model.CreatePredictionEngine<MovieRating,
MovieRatingPrediction>(model);
    var testInput = new MovieRating { ID_USER = 6, ID_SONG = 7 };

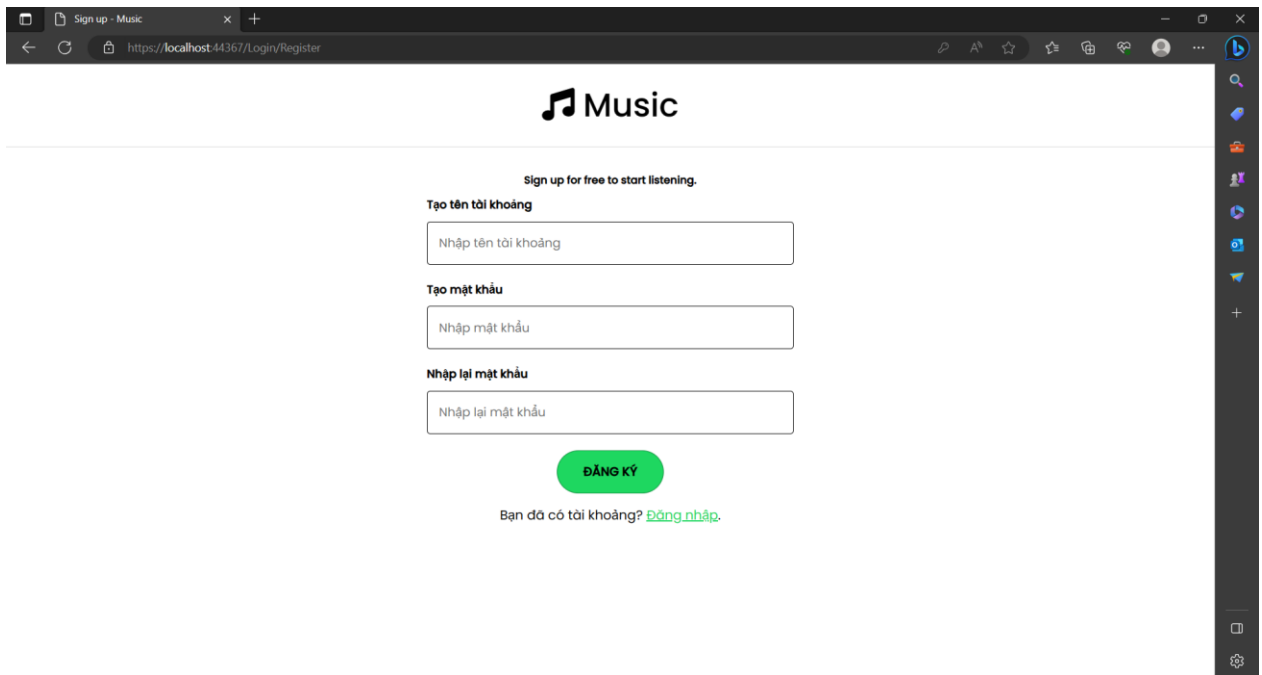
    var movieRatingPrediction = predictionEngine.Predict(testInput);
    if (Math.Round(movieRatingPrediction.Score, 1) > 3.5)
    {
        Console.WriteLine("Movie " + testInput.ID_SONG + " is recommended for
```

## - Hàm lưu mô hình:

```
static void SaveModel(MLContext mlContext, DataViewSchema
trainingDataViewSchema, ITransformer model)
{
    var modelPath = Path.Combine(Environment.CurrentDirectory, "Data",
"MovieRecommenderModel.zip");
    Console.WriteLine("===== Saving the model to a file
=====");
    mlContext.Model.Save(model, trainingDataViewSchema, modelPath);
}
```

## 5.3 Giao diện chương trình:

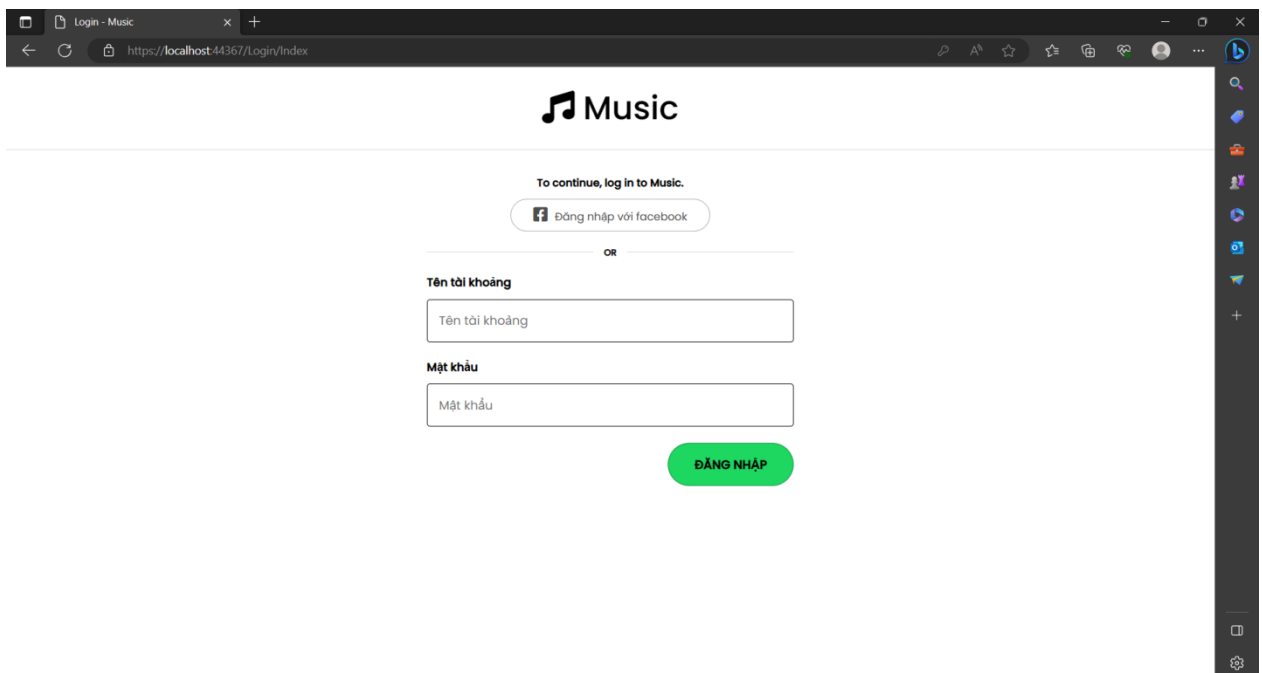
- Giao diện đăng ký:



The screenshot shows a web browser window with the address bar displaying "https://localhost:44367/Login/Register". The page features the "Music" logo at the top. Below the logo, the text "Sign up for free to start listening." is displayed. The registration form includes three input fields: "Tạo tên tài khoản" (Create account name), "Tạo mật khẩu" (Create password), and "Nhập lại mật khẩu" (Re-enter password). Each field has a placeholder text "Nhập tên tài khoản", "Nhập mật khẩu", and "Nhập lại mật khẩu" respectively. A green "ĐĂNG KÝ" (Register) button is positioned below the fields. At the bottom, there is a link "Bạn đã có tài khoản? [Đăng nhập.](#)" (Do you have an account? [Log in.](#)).

Hình 5. 3: Giao diện đăng ký

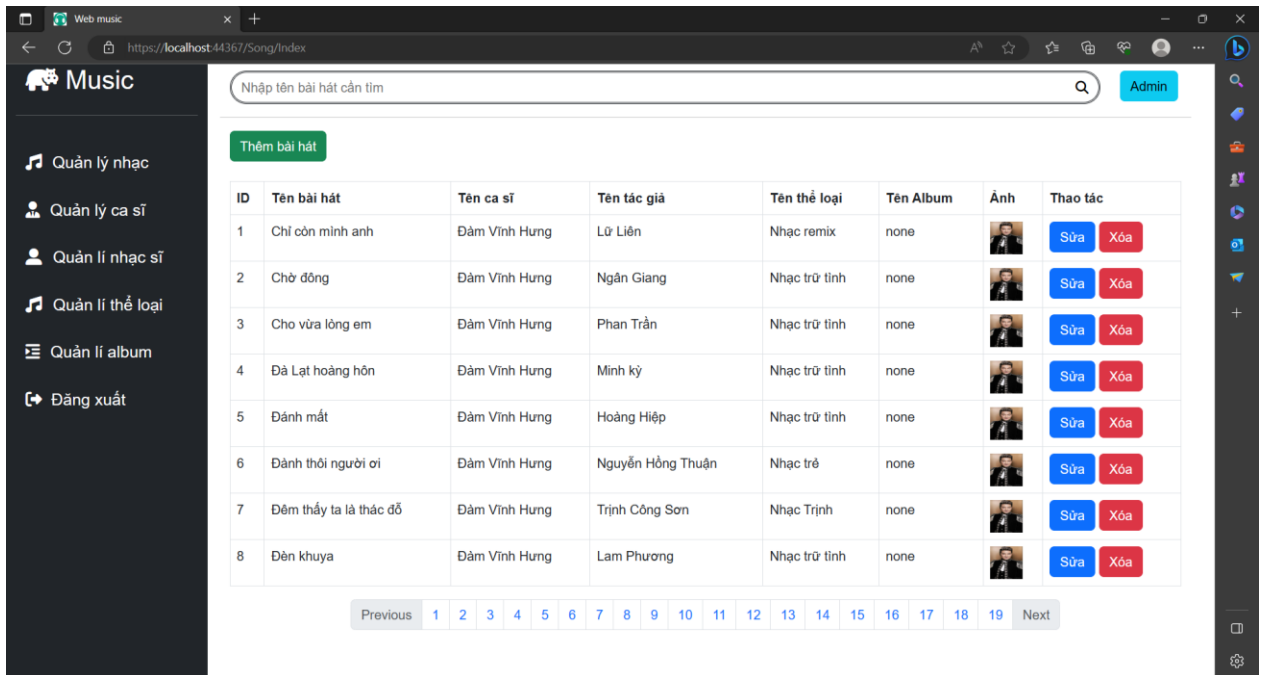
- Giao diện đăng nhập



The screenshot shows a web browser window with the address bar displaying "https://localhost:44367/Login/Index". The page features the "Music" logo at the top. Below the logo, the text "To continue, log in to Music." is displayed. There is a button with a Facebook icon and the text "Đăng nhập với facebook" (Log in with facebook). Below this, the word "OR" is centered. The login form includes two input fields: "Tên tài khoản" (Account name) and "Mật khẩu" (Password). Each field has a placeholder text "Tên tài khoản" and "Mật khẩu" respectively. A green "ĐĂNG NHẬP" (Log in) button is positioned below the fields.

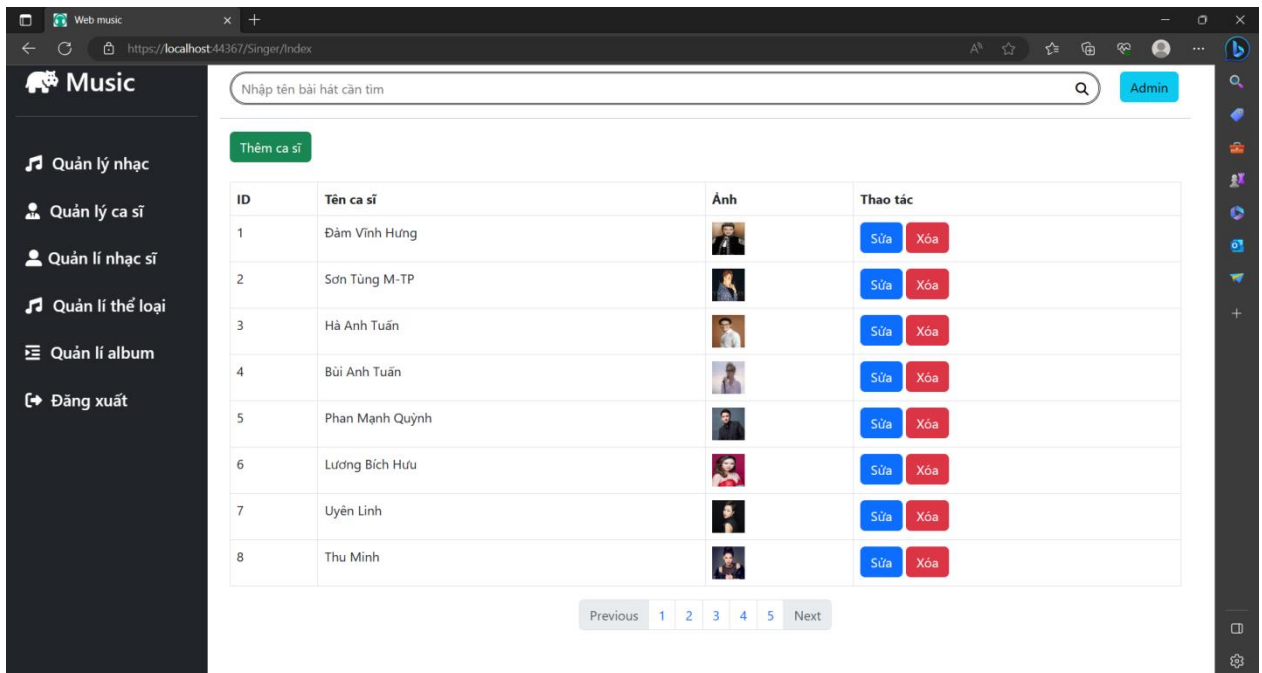
Hình 5. 4: Giao diện đăng nhập

## - Giao diện quản lý bài hát:



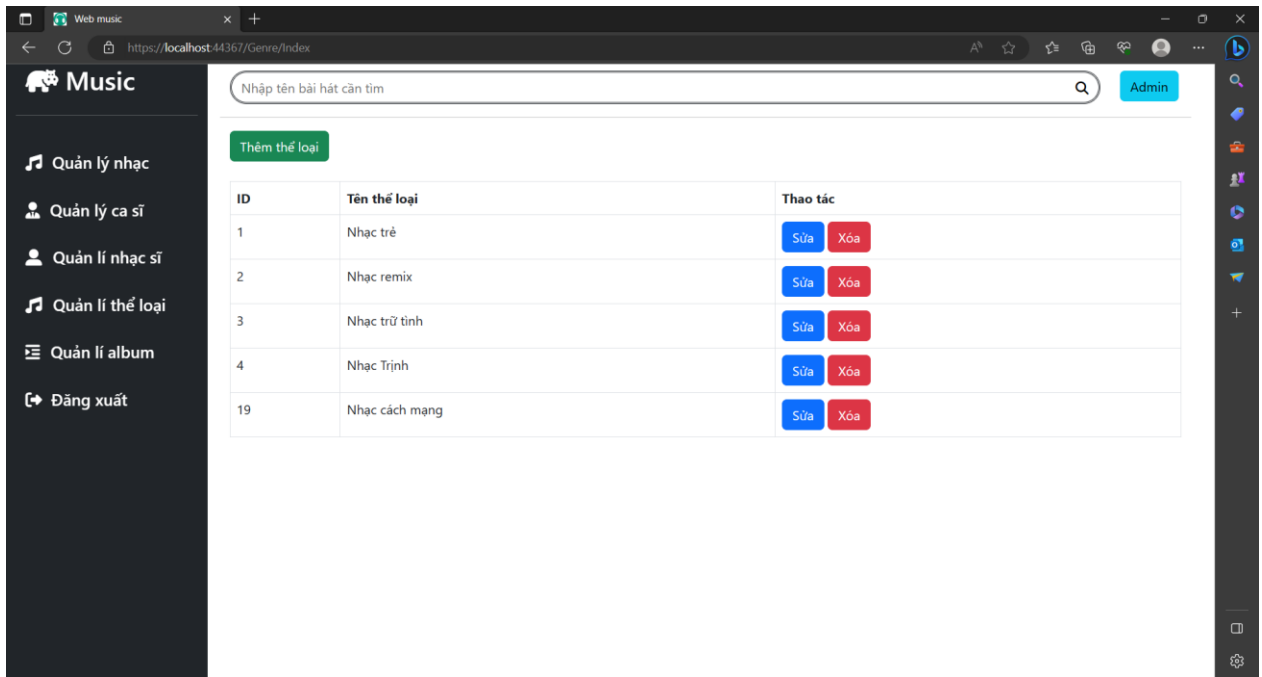
Hình 5. 5: Giao diện quản lý nhạc

## - Giao diện quản lý ca sĩ:



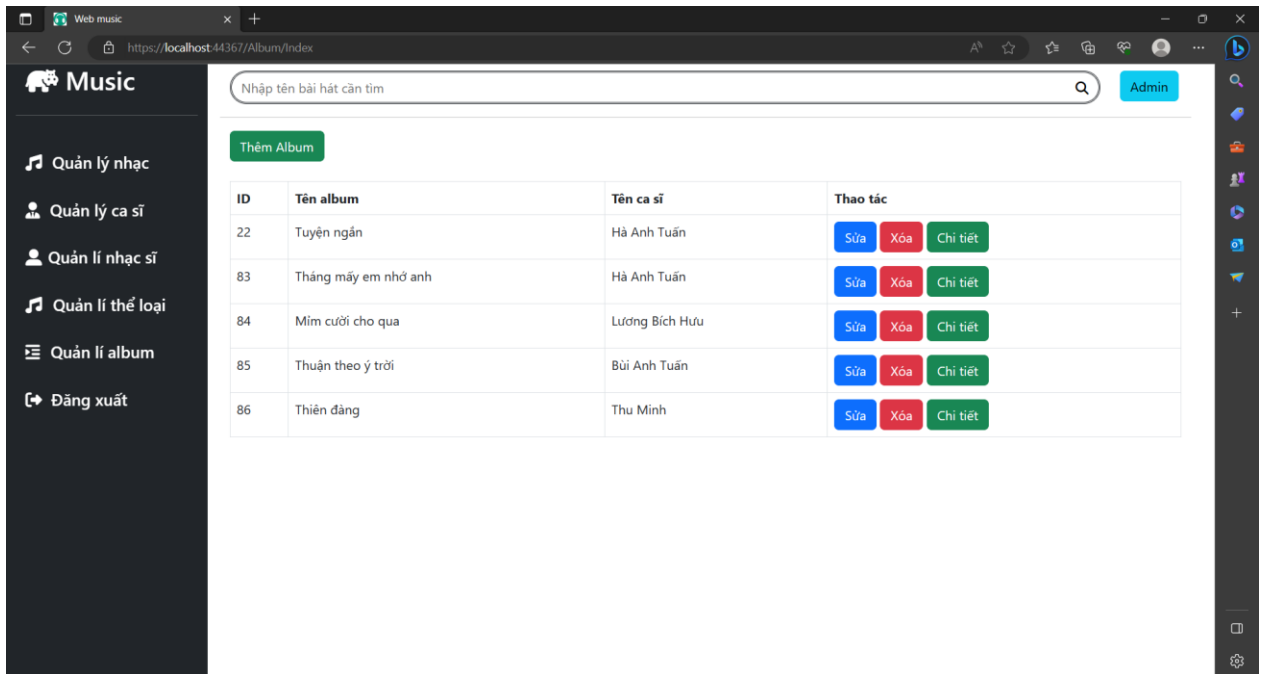
Hình 5. 6: Giao diện quản lý ca sĩ

- Giao diện quản lý thể loại:



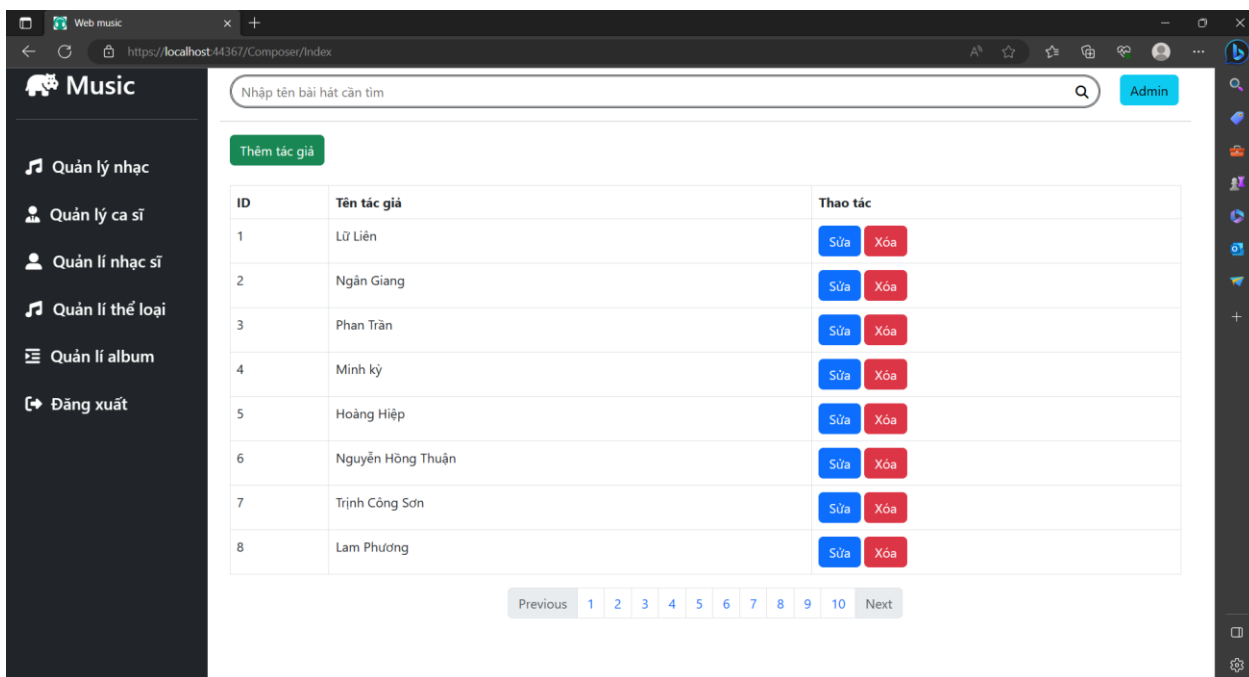
Hình 5. 7: Giao diện quản lý thể loại

- Giao diện quản lý album:



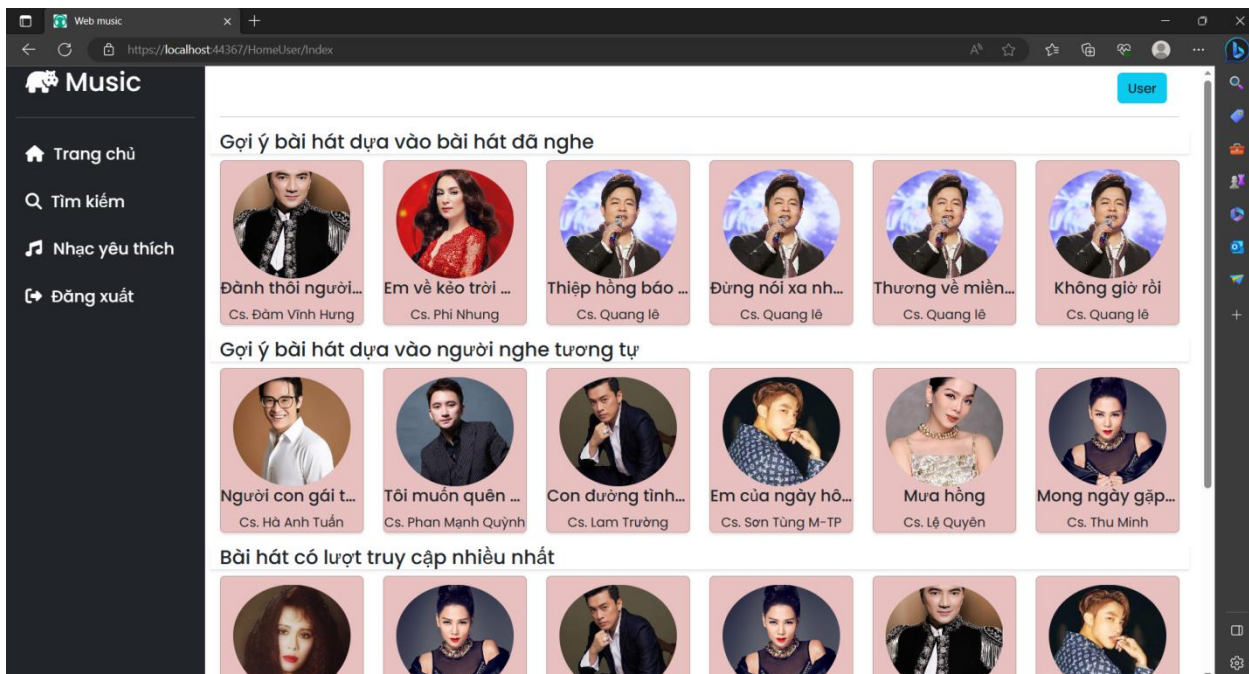
Hình 5. 8: Giao diện quản lý album

- Giao diện quản lý nhạc sĩ:



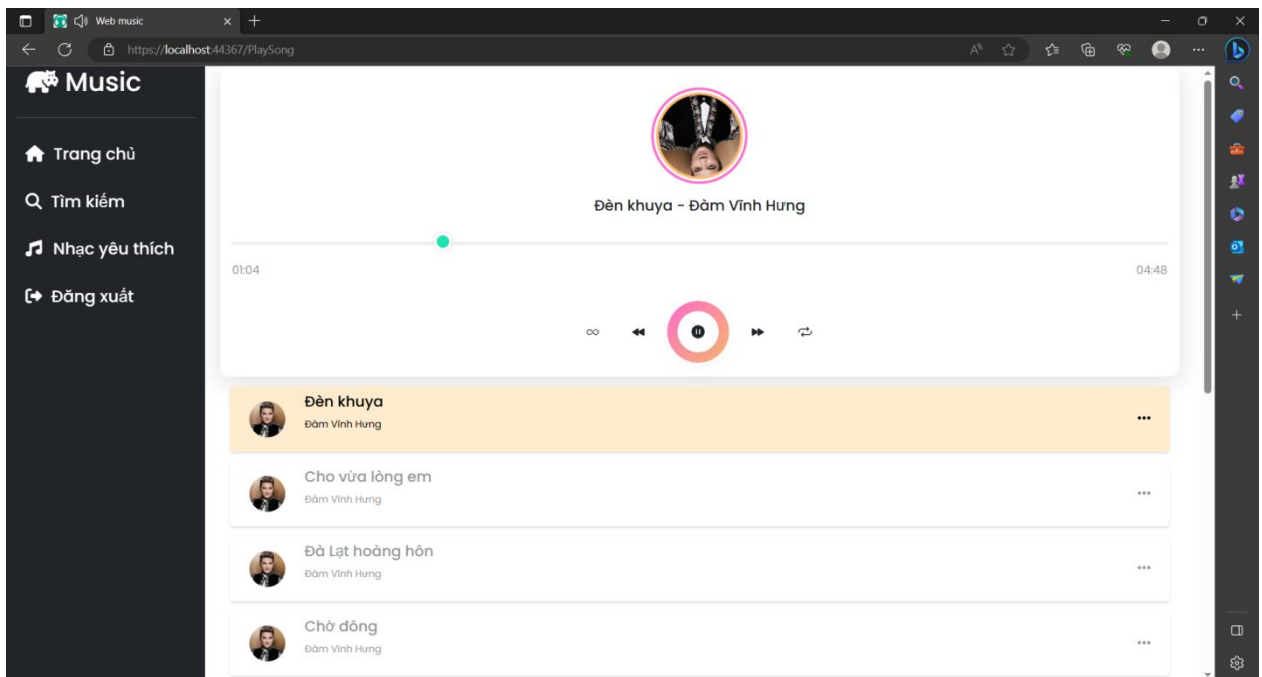
Hình 5. 9: Giao diện quản lý nhạc sĩ

- Giao diện trang chủ người dùng:



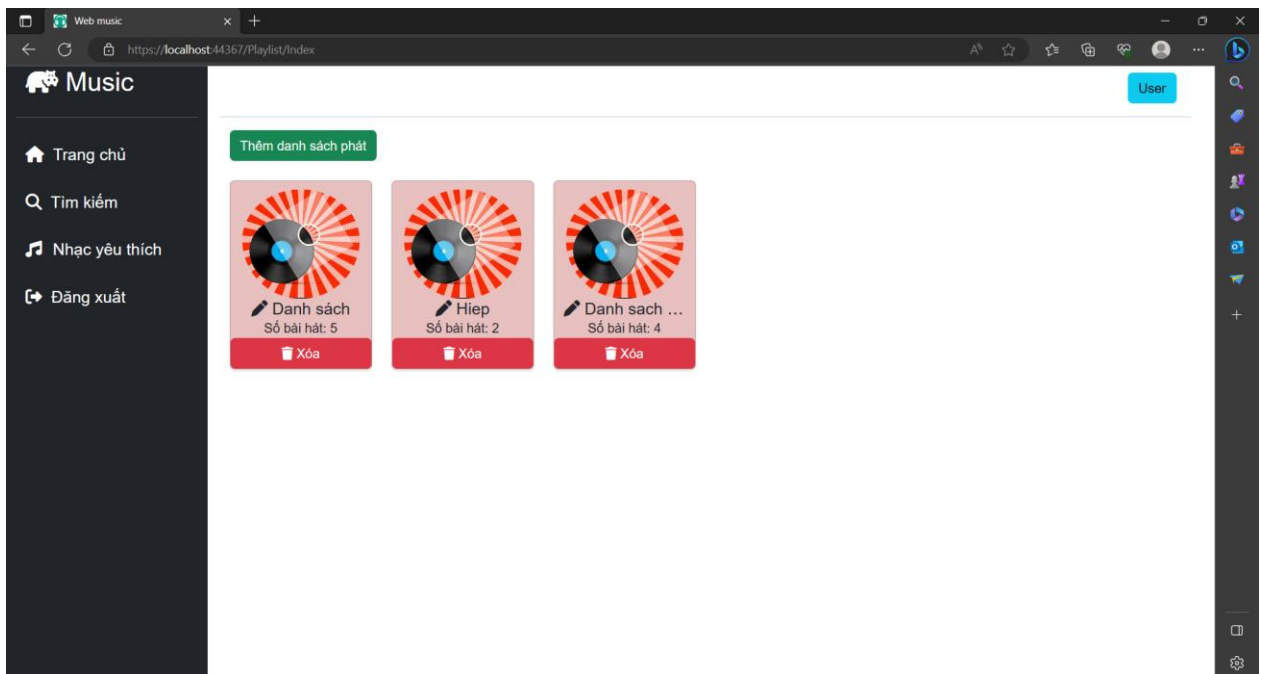
Hình 5. 10: Giao diện trang chủ người dùng

- Giao diện phát nhạc:



Hình 5. 11: Giao diện phát nhạc

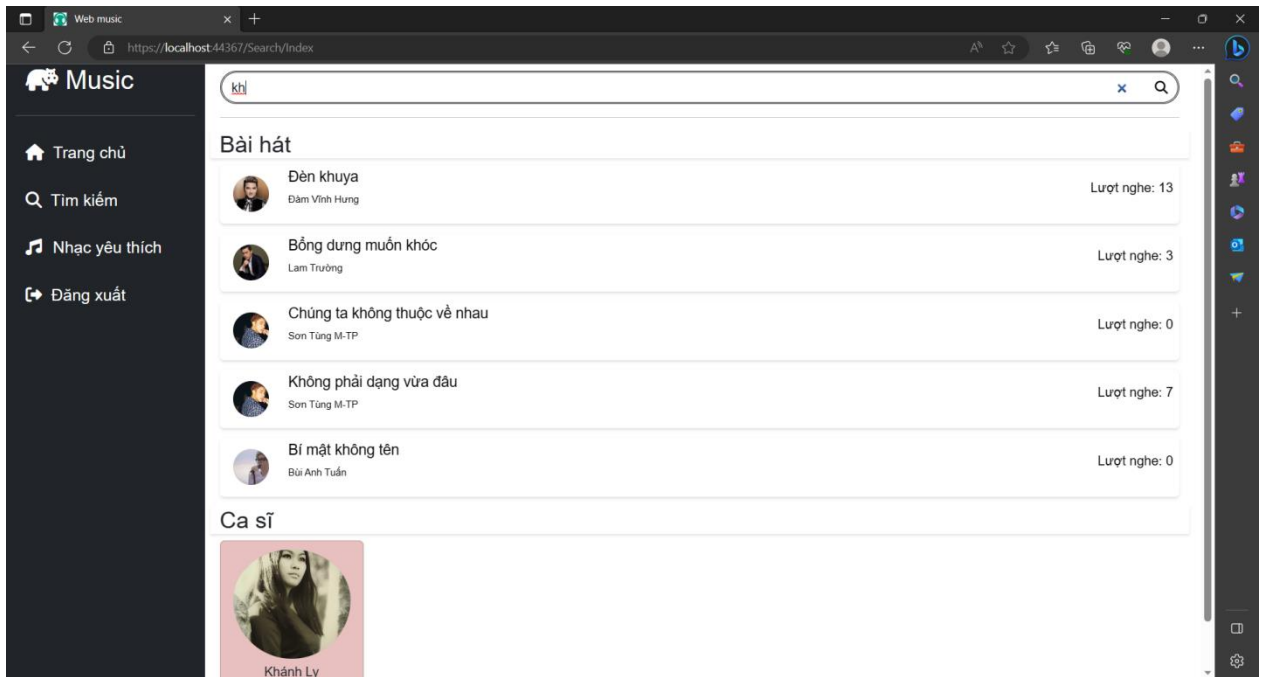
- Giao diện quản lý danh sách phát:



Hình 5. 12: Giao diện quản lý danh sách phát



- Giao diện tìm kiếm:



Hình 5. 13: Giao diện tìm kiếm của user

## KẾT LUẬN

### 1. Kết quả đạt được

Xây dựng thành công website nghe nhạc với các chức năng cơ bản, giao diện thân thiện với người dùng. Xây dựng hệ thống gợi ý bài hát cho người sử dụng. Hiểu được các công nghệ. Hiểu được machine learning và xây dựng cơ bản thành công hệ thống gợi ý bài hát cho người dùng.

### 2. Vấn đề tồn đọng

- Website còn ít chức năng.
- Do nguồn dữ liệu cung cấp chưa phù hợp, còn hạn chế về mặt thực tế nên mô hình gợi ý dựa trên lọc cộng tác hoạt động chưa thực sự chính xác.

## **PHỤ LỤC**

Phụ lục 1: hướng dẫn cài đặt

Link github: [https://github.com/hiep1512001/Do\\_An\\_Tot\\_Ngiep](https://github.com/hiep1512001/Do_An_Tot_Ngiep)

## TÀI LIỆU THAM KHẢO

- [1]. Giáo trình phân tích thiết kế hướng đối tượng của cô Phạm Thị Miên
- [2]. Giáo trình phân tích thiết kế hệ thống của cô Phạm Thị Miên
- [3]. [W3Schools Online Web Tutorials](#)
- [4]. [Get started with Bootstrap · Bootstrap v5.3 \(getbootstrap.com\)](#)
- [5]. [Machine Learning cơ bản \(machinelearningcoban.com\)](#) content-based
- [6]. [Tutorial: Build a movie recommender - matrix factorization - ML.NET | Microsoft Learn](#)
- [7]. [Giới thiệu nền tảng máy học ML.NET của Microsoft - Bài 1 | TopDev](#)
- [8]. [Building a Music Recommendation Engine | Engineering Education \(EngEd\) Program | Section](#)
- [9]. [Build A Product Recommender Using C# and ML.NET Machine Learning | by Mark Farragher | Medium](#)
- [10]. [Machine Learning with ML.NET - Recommendation Systems \(rubikscore.net\)](#)
- [11]. [How to Build a Content-Based Recommender System For Your Product - OfferZen](#)
- [12]. [Machine Learning cơ bản \(machinelearningcoban.com\)](#) matrix factorization
- [13]. [Beginners Guide to learn about Content Based Recommender Engine \(analyticsvidhya.com\)](#)
- [14]. [Building A Music Recommendation System Like Spotify | Music Recommender System | Intellipaat - YouTube](#)
- [15]. [Thiết kế website nghe nhạc online, tải nhạc trực tuyến ấn tượng \(bizfly.vn\)](#)
- [16]. [CRUD trong ASP.NET MVC sử dụng AJAX và Bootstrap \(luyencode.net\)](#)