

ỦY BAN NHÂN DÂN TP. HỒ CHÍ MINH  
**TRƯỜNG ĐẠI HỌC SÀI GÒN**



**Môn học: PHÁT TRIỂN PHẦN MỀM MÃ NGUỒN MỞ**

**Đề tài: Phát triển phần mềm Spotify Clone**

**Sinh viên thực hiện (Nhóm 10):**

**3121410123 - Phan Quang Duy – phanquangduyvt@gmail.com**

**3121410473 – Dương Trần Minh Thịnh – minhthinh9229@gmail.com**

**3121410413 – Phan Văn Quyền - vanquyenhtdz@gmail.com**

**Lớp: DKP1211**

**Năm học: 2025-2026**

**GVHD: ThS. Từ Lăng Phiêu**

**TP. HỒ CHÍ MINH, THÁNG 5 NĂM 2025**

# Mục lục

I. Giới thiệu.....	4
1. Mở đầu .....	4
2. Giới thiệu chung về Spotify và nhu cầu phát triển .....	4
3. Lý do chọn đề tài.....	4
4. Mục tiêu và phạm vi dự án .....	5
5. Ý nghĩa của dự án .....	5
II. Cơ sở lý thuyết .....	5
1. Các thư viện, công nghệ được sử dụng trong dự án .....	5
1.1. Ngôn ngữ lập trình .....	5
1.2. Framework .....	6
1.3. Cơ sở dữ liệu .....	6
1.4. Các công nghệ liên quan .....	6
2. Cấu trúc mã nguồn .....	6
3. Mô hình ứng dụng.....	8
4. Các tính năng được xây dựng .....	8
5. Flowchart .....	8
III. Cơ sở dữ liệu .....	13
1. Sơ đồ thực thể kết hợp (ERD – Entity Relationship Diagram) .....	13
2. Cơ sở dữ liệu mức logic .....	13
3. Mô hình dữ liệu quan hệ (RDM – Relational Data Model).....	15
4. Bảng mô tả thuộc tính vật lý của quan hệ.....	16
IV. Hình ảnh.....	20
V. Xử lý .....	26
1. Sơ đồ tuần tự (Sequence Diagram).....	26
2. Sơ đồ hoạt động (Activity Diagram) .....	32
VI. Cách cài đặt ứng dụng.....	40
1. Frontend .....	40
2. Backend.....	41
VII. Kết quả và đánh giá.....	43
VIII. Tài liệu tham khảo .....	46
LỜI KẾT.....	47

**Bảng phân công thực hiện đồ án**

MSSV	Họ và tên	Công việc phân công	Phần trăm hoàn thành
3121410123	Phan Quang Duy	Viết code giao diện phía user (phát nhạc, đăng nhập, đăng ký, tạo album, thêm nhạc vào album, thêm/xóa nhạc yêu thích)	100%
3121410473	Dương Trần Minh Thịnh	Viết các api cho các chức năng trong hệ thống	100%
3121410413	Phan Văn Quyển	Viết code giao diện phía admin (quản lý nhạc, quản quản lý album)	100%

## **Lời nói đầu**

Chúng em xin bày tỏ lòng biết ơn sâu sắc đến giảng viên hướng dẫn, thầy Từ Lăng Phiêu, người đã tận tình hướng dẫn, chia sẻ những kiến thức chuyên môn quý giá và hỗ trợ chúng em trong suốt quá trình thực hiện đồ án "Phát triển phần mềm Spotify Clone".

Dự án "Phát triển phần mềm Spotify Clone" được thực hiện với mục tiêu xây dựng một ứng dụng nghe nhạc trực tuyến mô phỏng các tính năng chính của Spotify. Thông qua dự án này, chúng em có cơ hội áp dụng những kiến thức đã học vào thực tiễn, đồng thời nâng cao kỹ năng lập trình, làm việc nhóm và quản lý dự án. Đây cũng là dịp để chúng em khám phá những thách thức thực tế trong quá trình phát triển phần mềm, từ thiết kế giao diện, xây dựng backend đến tối ưu hóa hiệu suất ứng dụng.

Chúng em hy vọng rằng báo cáo này sẽ thể hiện rõ những nỗ lực và kết quả mà chúng em đã đạt được trong quá trình thực hiện đồ án. Đồng thời, chúng em rất mong nhận được những ý kiến đóng góp quý báu từ thầy và các bạn đọc để có thể tiếp tục hoàn thiện dự án trong tương lai.

# **I. GIỚI THIỆU**

## **1. Mở đầu**

Phát triển phần mềm trong lĩnh vực công nghệ âm nhạc đã và đang trở thành một xu hướng nổi bật, đặc biệt trong bối cảnh nhu cầu giải trí trực tuyến ngày càng tăng cao. Trong số các nền tảng âm nhạc nổi tiếng, Spotify đã khẳng định vị thế của mình như một trong những dịch vụ streaming hàng đầu thế giới, với hàng triệu người dùng và kho nhạc phong phú. Việc nghiên cứu và tái tạo một phiên bản mô phỏng của Spotify, được gọi là Spotify Clone, không chỉ mang tính học thuật mà còn là cơ hội để hiểu sâu hơn về cách thức hoạt động của các ứng dụng âm nhạc hiện đại, từ giao diện người dùng đến cơ sở hạ tầng kỹ thuật. Chính vì vậy, nhóm chúng tôi đã lựa chọn đề tài này để khám phá và áp dụng kiến thức vào thực tiễn.

## **2. Giới thiệu chung về Spotify và nhu cầu phát triển**

Spotify là một nền tảng cung cấp dịch vụ nghe nhạc trực tuyến, cho phép người dùng truy cập vào một thư viện nhạc khổng lồ với các bài hát, album, và playlist từ khắp nơi trên thế giới. Tính năng nổi bật của Spotify bao gồm khả năng cá nhân hóa trải nghiệm nghe nhạc thông qua các gợi ý dựa trên sở thích người dùng, tạo playlist, và tích hợp với nhiều thiết bị. Nhận thấy tiềm năng và sự phổ biến của mô hình này, chúng tôi quyết định phát triển một dự án mô phỏng mang tên Spotify Clone, nhằm tái hiện những chức năng cốt lõi như phát nhạc, quản lý playlist, tìm kiếm bài hát, và giao diện thân thiện với người dùng. Điều này không chỉ giúp chúng tôi học hỏi từ một sản phẩm thành công mà còn tạo ra một sản phẩm thực tế để thử nghiệm và cải thiện kỹ năng lập trình.

## **3. Lý do chọn đề tài**

Việc lựa chọn phát triển Spotify Clone xuất phát từ mong muốn tìm hiểu sâu hơn về quy trình xây dựng một ứng dụng âm nhạc toàn diện. Qua đó, chúng tôi có cơ hội nghiên cứu các công nghệ hiện đại như cơ sở dữ liệu, giao diện người dùng, và thuật toán đề xuất (nếu có thể tích hợp). Đồng thời, đề tài này cho phép chúng tôi áp dụng các kiến thức đã học vào thực tế, từ việc thiết kế hệ thống, lập trình, đến kiểm thử và tối ưu hóa hiệu

suất. Ngoài ra, dự án còn mang tính thực tiễn cao, phù hợp với xu hướng phát triển các ứng dụng giải trí trên nền tảng kỹ thuật số hiện nay.

#### **4. Mục tiêu và phạm vi dự án**

Mục tiêu chính của dự án là xây dựng một ứng dụng Spotify Clone với các chức năng cơ bản như phát nhạc trực tuyến, tìm kiếm bài hát theo từ khóa, tạo và quản lý playlist cá nhân, cũng như hỗ trợ đăng nhập và đăng ký người dùng. Tuy nhiên, do hạn chế về thời gian và nguồn lực, dự án sẽ tập trung vào việc tái hiện các tính năng cốt lõi mà không tích hợp các yếu tố phức tạp như streaming nhạc trực tiếp từ nguồn bên ngoài hoặc sử dụng trí tuệ nhân tạo để gợi ý nhạc nâng cao. Phạm vi dự án sẽ giới hạn trong việc phát triển một ứng dụng cơ bản, dễ sử dụng, và có thể mở rộng trong tương lai nếu cần thiết.

#### **5. Ý nghĩa của dự án**

Dự án không chỉ là một bài tập học tập mà còn mang lại giá trị thực tiễn. Qua việc phát triển Spotify Clone, chúng tôi không chỉ nâng cao kỹ năng lập trình mà còn hiểu rõ hơn về cách các ứng dụng âm nhạc hoạt động, từ cơ sở dữ liệu lưu trữ thông tin bài hát đến cách tổ chức giao diện người dùng. Kết quả của dự án có thể được sử dụng như một nền tảng để tiếp tục phát triển thêm các tính năng mới, hoặc làm tài liệu tham khảo cho các dự án tương tự trong tương lai.

Cuối cùng, chúng tôi hy vọng rằng báo cáo này sẽ phản ánh được quá trình nỗ lực và những bài học kinh nghiệm quý báu trong suốt quá trình thực hiện. Mọi ý kiến đóng góp từ giảng viên và bạn đọc sẽ là nguồn động lực để chúng tôi hoàn thiện hơn trong các dự án tiếp theo.

## **II. CƠ SỞ LÝ THUYẾT**

### **1. Các thư viện, công nghệ được sử dụng trong dự án**

Dự án Spotify Clone được phát triển dựa trên một số công nghệ và thư viện hiện đại để đảm bảo hiệu suất và tính khả thi. Dưới đây là danh sách chi tiết:

#### **1.1. Ngôn ngữ lập trình**

- JavaScript: Sử dụng cho cả front-end để phát triển ứng dụng web. Lý do lựa chọn là tính linh hoạt, cộng đồng hỗ trợ lớn, và khả năng chạy trên nhiều nền tảng.

- HTML/CSS: Được dùng để xây dựng giao diện người dùng cơ bản, kết hợp với JavaScript để tạo trải nghiệm động.
- Python: Ngôn ngữ chính để phát triển backend, ưu điểm: dễ đọc, dễ bảo trì, có nhiều thư viện hỗ trợ, phù hợp cho phát triển web và xử lý dữ liệu.

## **1.2. Framework**

- React: Framework front-end được chọn để xây dựng giao diện người dùng tương tác cao. React giúp tái sử dụng các thành phần (component) và tối ưu hóa hiệu suất nhờ cơ chế Virtual DOM.
- Django: Framework web full-stack dùng để xây dựng ứng dụng web hoàn chỉnh. Cung cấp ORM (Object-Relational Mapping) hỗ trợ tương tác cơ sở dữ liệu. Hệ thống admin có sẵn và bảo mật tích hợp. URL routing và template engine để tạo giao diện.
- Django REST Framework: Framework xây dựng REST API cho ứng dụng backend. Hỗ trợ serialization/deserialization và authentication & permissions. ViewSets & Routers giúp tự động hóa endpoint. Filtering & Pagination tối ưu hóa dữ liệu.

## **1.3. Cơ sở dữ liệu**

- MySQL: Cơ sở dữ liệu quan hệ (RDBMS) được lựa chọn để lưu trữ thông tin liên quan đến nguồn dùng, bài hát, playlist, và album. MySQL phù hợp với dữ liệu có cấu trúc cố định và dễ mở rộng.

## **1.4. Các công nghệ liên quan**

- RESTful API: Được sử dụng để kết nối front-end và back-end, cho phép truyền dữ liệu giữa giao diện và cơ sở dữ liệu một cách hiệu quả.
- Authentication: Clerk: Sử dụng cho authentication, cung cấp giải pháp xác thực người dùng một cách hiệu quả. OAuth: Tích hợp với Google, cho phép đăng nhập qua các nền tảng xã hội một cách thuận tiện.
- Cloundinary: Là một nền tảng quản lý phương tiện đám mây (SaaS) hỗ trợ các nhà phát triển và doanh nghiệp quản lý, biến đổi, tối ưu hóa hình ảnh, video và tài sản đa phương tiện một cách hiệu quả. Cung cấp khả năng tải lên, lưu trữ, chỉnh sửa tự động bằng AI và phân phối nhanh chóng qua CDN, phù hợp với mọi thiết bị.

## **2. Cấu trúc mã nguồn**

Cấu trúc mã nguồn của dự án được tổ chức theo mô hình thư mục rõ ràng để dễ bảo trì và phát triển:

## Front-end:

- `src/`: Thư mục chứa mã nguồn chính của ứng dụng
  - `components/`: Chứa các component có thể tái sử dụng
  - `pages/`: Chứa các trang chính của ứng dụng
  - `layout/`: Chứa các component layout
  - `providers/`: Chứa các context provider
  - `stores/`: Chứa các store quản lý state
  - `lib/`: Chứa các utility functions và cấu hình
  - `types/`: Chứa các type definitions
  - `assets/`: Chứa các tài nguyên như hình ảnh, fonts
  - `main.jsx`: Entry point của ứng dụng
  - `App.jsx`: Component gốc của ứng dụng
  - `index.css`: Styles toàn cục
- `public/`: Thư mục chứa các tài nguyên tĩnh
- `node_modules/`: Thư mục chứa các thư viện đã cài đặt
- `package.json`: Quản lý dependencies và scripts
- `vite.config.js`: Cấu hình cho Vite bundler
- `jsconfig.json`: Cấu hình JavaScript
- `eslint.config.js`: Cấu hình ESLint
- `components.json`: Cấu hình components

## Back-end:

- `spotify/`
  - `settings.py`: Cấu hình chính của dự án
  - `urls.py`: Cấu hình URL chính
  - `wsgi.py` & `asgi.py`: Cấu hình server
  - `models.py`: Định nghĩa models
  - `__init__.py`: File khởi tạo Python package
- `spotify_app/`
  - `api/`: Chứa các API endpoints
  - `auth/`: Xử lý xác thực
  - `models/`: Định nghĩa models
  - `serializers/`: Serializers cho API
  - `management/`: Custom management commands
  - `migrations/`: Chứa các file migration database
  - `tests.py`: Unit tests
  - `admin.py`: Cấu hình Django admin
  - `apps.py`: Cấu hình ứng dụng



- urls.py: Cấu hình URL của ứng dụng

### 3. Mô hình ứng dụng

Ứng dụng Spotify Clone được thiết kế theo mô hình Client-Server. Các thành phần chính bao gồm:

- Client (Front-end): Giao diện người dùng được xây dựng bằng React, hiển thị danh sách bài hát, playlist, và điều khiển phát nhạc.
- Server (Back-end): Xử lý logic, quản lý API, và kết nối với cơ sở dữ liệu MySQL.
- Cơ sở dữ liệu: Lưu trữ thông tin người dùng, bài hát, và playlist.

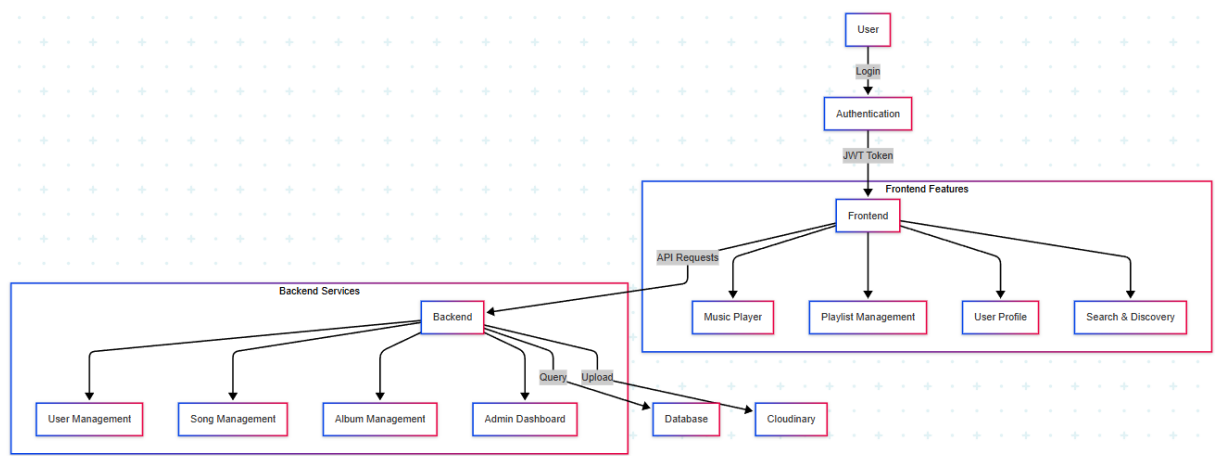
### 4. Các tính năng được xây dựng

Dự án Spotify Clone bao gồm các tính năng chính sau:

- Đăng ký/Đăng nhập: Cho phép người dùng tạo tài khoản và truy cập ứng dụng bằng tài khoản google (sử dụng Clerk Authentication).
- Tìm kiếm bài hát/album: Hỗ trợ tìm kiếm theo từ khóa, trả về danh sách kết quả phù hợp.
- Phát nhạc: Phát bài hát trực tuyến với điều khiển play/pause, next/previous.
- Tạo/Xem: Người dùng có thể tạo playlist mới, thêm/xóa bài hát.
- Quản lý thư viện nhạc cá nhân: Hiển thị danh sách bài hát yêu thích và album nhạc cá nhân.

### 5. Flowchart

Luồng Kiến Trúc Hệ Thống



### Authentication Flow

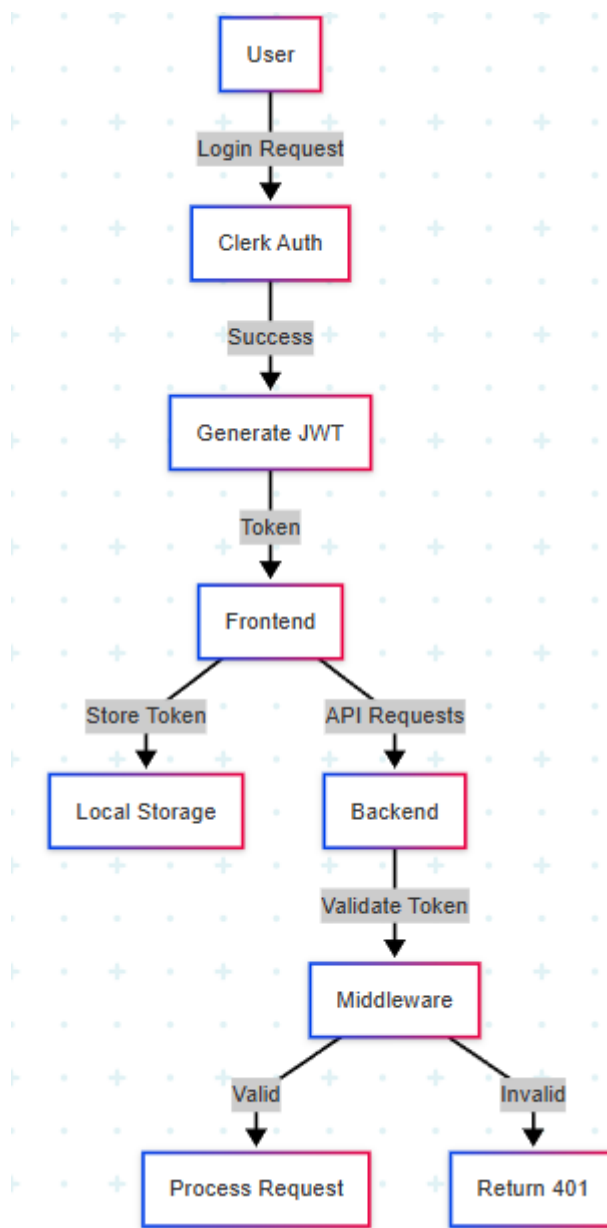
Mô tả:

## 1. Quá Trình Đăng Nhập

- Người dùng đăng nhập thông qua Clerk
- Clerk xử lý xác thực và tạo JWT token
- Token được lưu trong local storage
- Middleware Django xác thực token cho mỗi request

## 2. Phân Quyền

- Kiểm tra quyền admin dựa trên email
- Phân quyền người dùng thông thường và admin
- Bảo vệ các route API
- Xử lý token hết hạn



## Music Player Flow

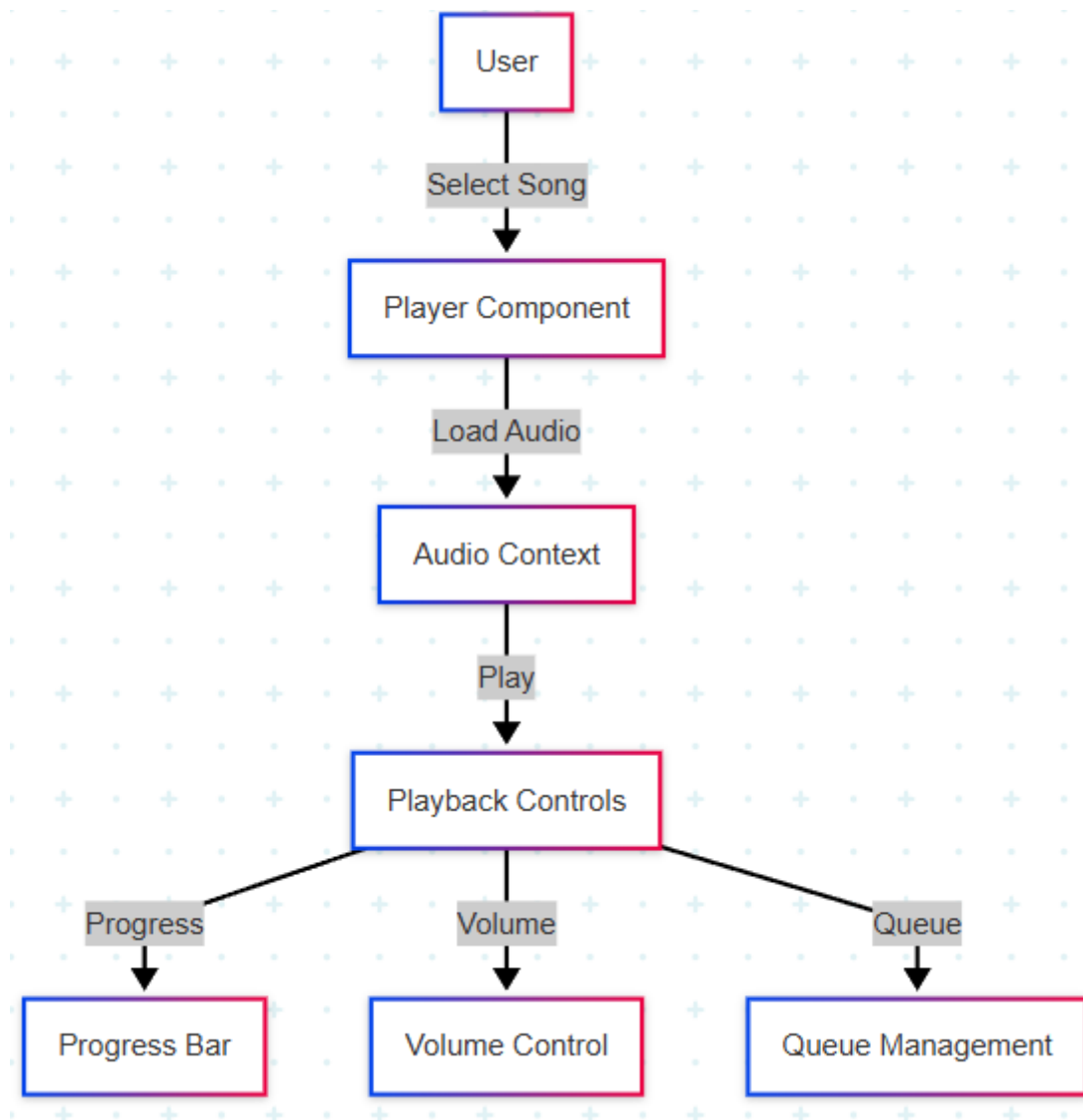
Mô tả:

## 1. Phát Nhạc

- Tải và phát file âm thanh từ Cloundinary
- Điều khiển phát/tạm dừng
- Điều chỉnh âm lượng
- Hiện thị tiến trình phát

## 2. Quản Lý Hàng Đợi

- Thêm/xóa bài hát khỏi hàng đợi
- Chuyển bài tiếp theo/trước đó
- Lưu trạng thái phát



## Playlist Management Flow

Mô tả:

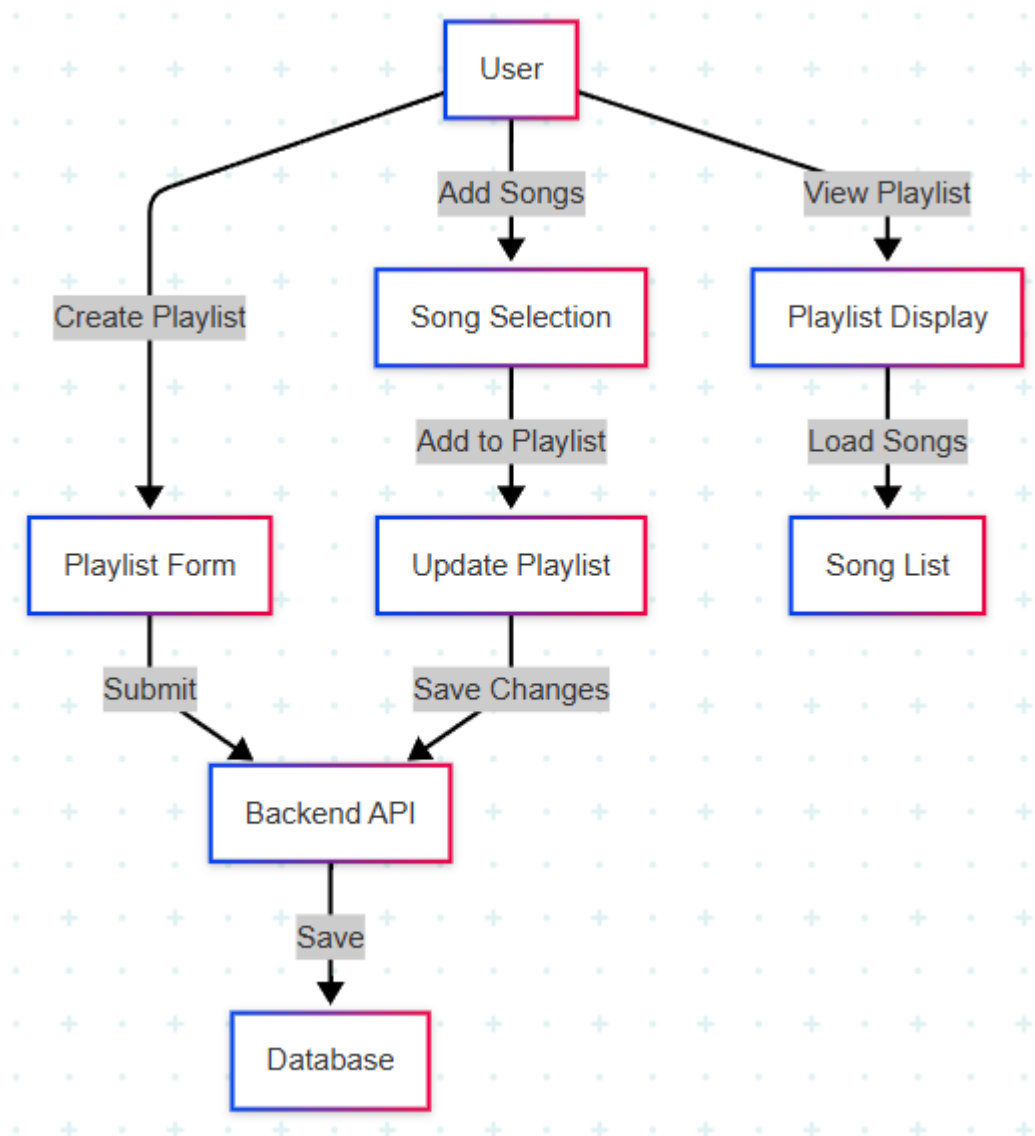
### 1. Tạo và Quản Lý Playlist

- Tạo playlist mới
- Thêm/xóa bài hát vào playlist

- Xem danh sách playlist
- Cập nhật thông tin playlist

## 2. Tương Tác với Bài Hát

- Thêm bài hát vào playlist
- Xóa bài hát khỏi playlist



## Album Management Flow

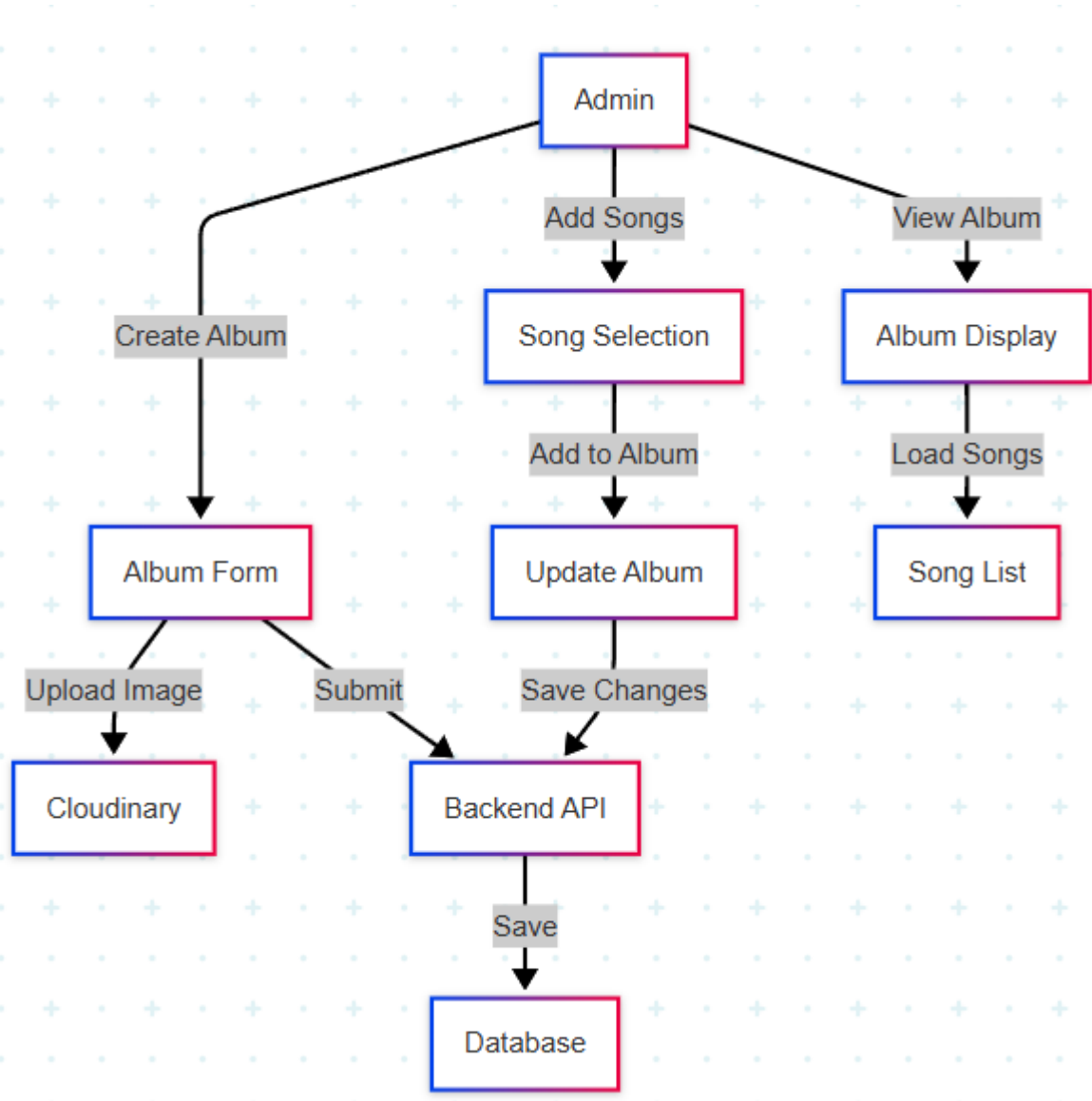
Mô tả:

### 1. Quản Lý Album (Admin)

- Tạo album mới
- Tải ảnh bìa lên Cloudinary
- Thêm thông tin album
- Quản lý bài hát trong album

### 2. Hiện Thị Album

- Xem thông tin album
- Xem danh sách bài hát
- Phát bài hát từ album



## Songs Management Flow

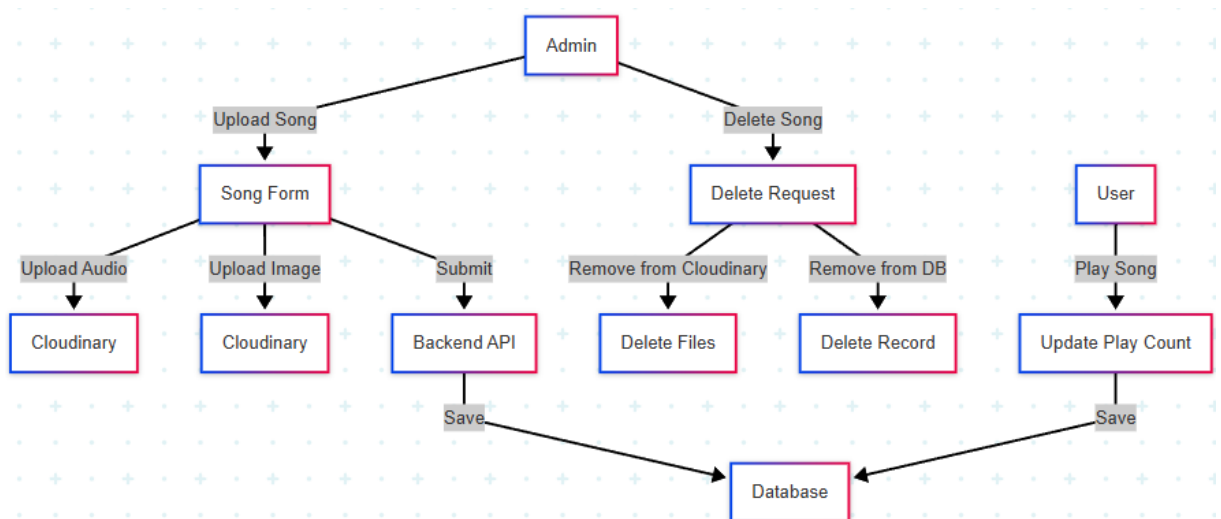
Mô tả:

### 1. Quản Lý Bài Hát (Admin)

- Tải lên bài hát mới
  - Upload file âm thanh lên Cloudinary
  - Upload ảnh bìa lên Cloudinary
  - Thêm thông tin bài hát (tên, nghệ sĩ, thời lượng)
  - Lưu thông tin vào database
- Xóa bài hát
  - Xóa file từ Cloudinary
  - Xóa thông tin từ database

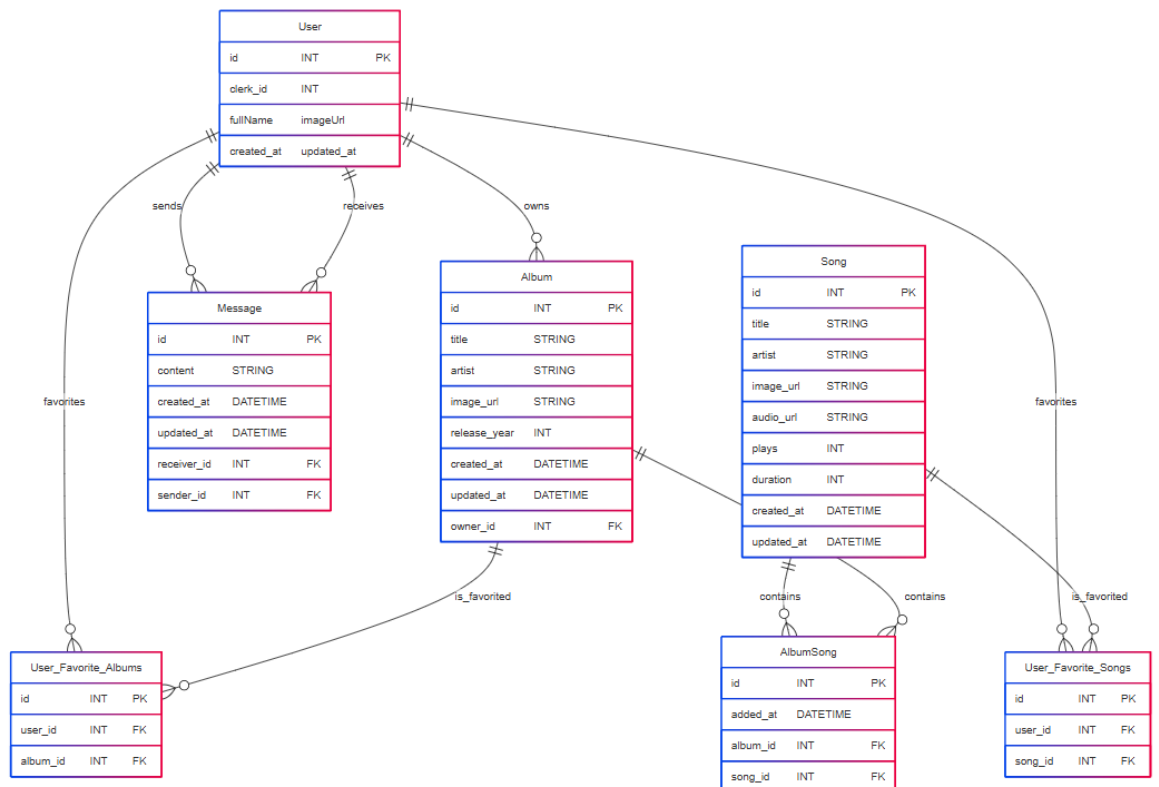
## 2. Tương Tác Người Dùng

- Phát bài hát
- Cập nhật số lượt phát
- Thêm vào playlist yêu thích



## III. CƠ SỞ DỮ LIỆU

### 1. Sơ đồ thực thể kết hợp (ERD – Entity Relationship Diagram)



### 2. Cơ sở dữ liệu mức logic

**User (Người dùng):**

- Thuộc tính:
  - id (Khóa chính): Định danh duy nhất cho người dùng.
  - clerk\_id: Mã định danh từ hệ thống xác thực bên ngoài.
  - fullName: Họ và tên người dùng.
  - imageUrl: Đường dẫn đến ảnh đại diện.
  - created\_at: Thời gian tạo tài khoản.
  - updated\_at: Thời gian cập nhật thông tin gần nhất.

### **Song (Bài hát):**

- Thuộc tính:
  - id (Khóa chính): Định danh duy nhất cho bài hát.
  - title: Tiêu đề bài hát.
  - artist: Tên nghệ sĩ.
  - image\_url: Đường dẫn đến ảnh bìa bài hát.
  - audio\_url: Đường dẫn đến tệp âm thanh.
  - plays: Số lượt phát bài hát.
  - duration: Thời lượng bài hát.
  - created\_at: Thời gian tạo bài hát.
  - updated\_at: Thời gian cập nhật thông tin gần nhất.

### **Album (Album nhạc):**

- Thuộc tính:
  - id (Khóa chính): Định danh duy nhất cho album.
  - title: Tiêu đề album.
  - artist: Tên nghệ sĩ của album.
  - image\_url: Đường dẫn đến ảnh bìa album.
  - release\_year: Năm phát hành album.
  - created\_at: Thời gian tạo album.
  - updated\_at: Thời gian cập nhật thông tin gần nhất.
  - owner\_id (Khóa ngoại): Liên kết đến User (có thể NULL nếu album không thuộc người dùng cụ thể).

### **Message (Tin nhắn):**

- Thuộc tính:
  - id (Khóa chính): Định danh duy nhất cho tin nhắn.
  - content: Nội dung tin nhắn.
  - created\_at: Thời gian gửi tin nhắn.
  - updated\_at: Thời gian cập nhật tin nhắn.





#### 4. Bảng mô tả thuộc tính vật lý của quan hệ

**Bảng User**

Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
Id	CHAR(32)	PRIMARY KEY	Định danh duy nhất cho người dùng.
clerk_id	VARCHAR(100)	UNIQUE	Mã định danh từ hệ thống xác thực bên ngoài.
fullName	VARCHAR(255)		Họ và tên đầy đủ của người dùng.
imageUrl	VARCHAR(255)		Đường dẫn đến ảnh đại diện của người dùng.
created_at	DATETIME	NOT NULL	Thời gian tạo tài khoản.
updated_at	DATETIME	NOT NULL	Thời gian cập nhật thông tin gần nhất.

**Bảng Song**

Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
Id	BIGINT	PRIMARY KEY, AUTO_INCREMENT	Định danh duy nhất cho bài hát, tự động tăng.
title	VARCHAR(255)	NOT NULL	Tiêu đề của bài hát.
artist	VARCHAR(255)	NOT NULL	Tên nghệ sĩ thể hiện bài hát.
image_url audio_url	VARCHAR(255) VARCHAR(255)		Đường dẫn đến tệp âm

			thanh của bài hát.
plays	INT	DEFAULT 0	Số lượt phát của bài hát.
duration	INT	NOT NULL	Thời lượng bài hát (tính bằng giây).
created_at	DATETIME	NOT NULL	Thời gian tạo bài hát.
updated_at	DATETIME	NOT NULL	Thời gian cập nhật thông tin gần nhất.

### **Bảng Album**

Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
id	BIGINT	PRIMARY KEY, AUTO_INCREMENT	Định danh duy nhất cho album, tự động tăng.
title	VARCHAR(255)	NOT NULL	Tiêu đề của album.
artist	VARCHAR(255)	NOT NULL	Tên nghệ sĩ của album.
image_url	VARCHAR(1024)		Đường dẫn đến ảnh bìa của album.
release_year	INT		Năm phát hành album.
created_at	DATETIME	NOT NULL	Thời gian tạo album.
updated_at	DATETIME	NOT NULL	Thời gian cập nhật

			thông tin gần nhất.
owner_id	CHAR(32)	FOREIGN KEY (User.id), NULLABLE	NULLABLE Liên kết đến User sở hữu album (có thể NULL nếu không thuộc người dùng).

**Bảng Message**

Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
id	BIGINT	PRIMARY KEY, AUTO_INCREMENT	Định danh duy nhất cho tin nhắn, tự động tăng.
content	LONGTEXT	NOT NULL	Nội dung của tin nhắn.
created_at	DATETIME	NOT NULL	Thời gian gửi tin nhắn.
updated_at	DATETIME	NOT NULL	Thời gian cập nhật tin nhắn gần nhất.
release_year	INT		Năm phát hành album.
receiver_id	INT	FOREIGN KEY (User.id)	Liên kết đến User nhận tin nhắn.
sender_id	INT	FOREIGN KEY (User.id)	Liên kết đến User gửi tin nhắn.

**Bảng User\_Favorite\_Albums**

Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
id	BIGINT	PRIMARY KEY, AUTO_INCREMENT	Định danh duy nhất cho bản ghi, tự động tăng.
user_id	CHAR(32)	FOREIGN KEY (User.id)	Liên kết đến User yêu thích album.
album_id	BIGINT	FOREIGN KEY (Album.id)	Liên kết đến Album được yêu thích.

**Bảng User\_Favorite\_Songs**

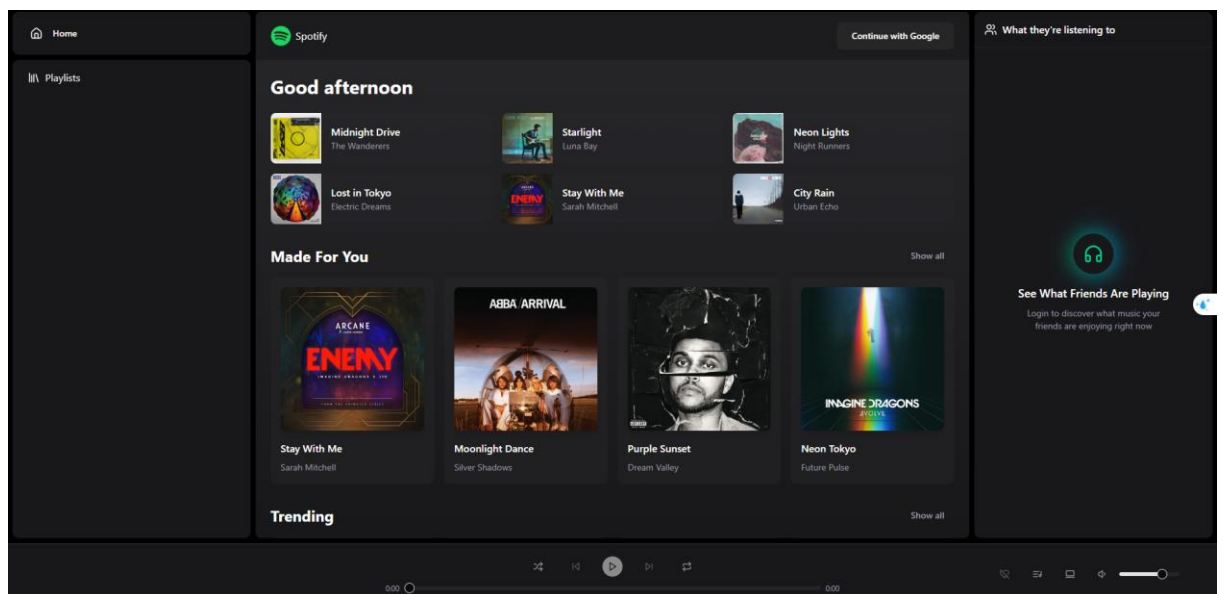
Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
id	BIGINT	PRIMARY KEY, AUTO_INCREMENT	Định danh duy nhất cho bản ghi, tự động tăng.
user_id	CHAR(32)	FOREIGN KEY (User.id)	Liên kết đến User yêu thích album.
song_id	BIGINT	FOREIGN KEY (Song.id)	Liên kết đến Song được yêu thích.

**Bảng AlbumSong**

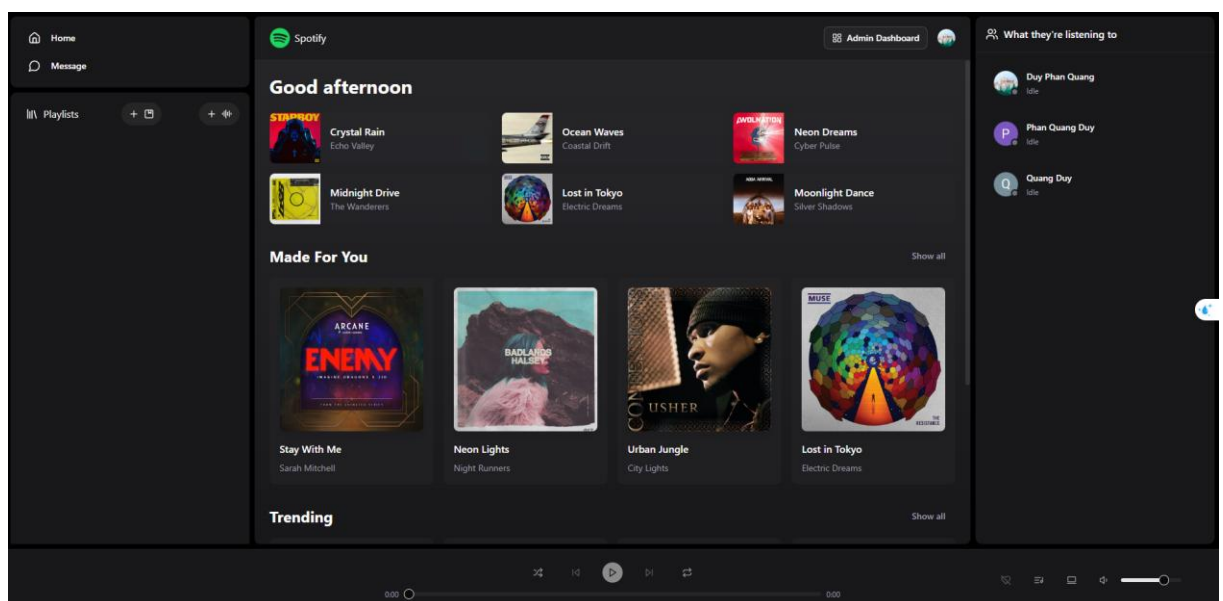
Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
id	BIGINT	PRIMARY KEY, AUTO_INCREMENT	Định danh duy nhất cho bản ghi, tự động tăng.
added_at	DATETIME	NOT NULL	Thời gian thêm bài hát

			vào album.
album_id	BIGINT	FOREIGN KEY (Album.id)	Liên kết đến Album chứa bài hát.
song_id	BIGINT	FOREIGN KEY (Song.id)	Liên kết đến Song thuộc album.

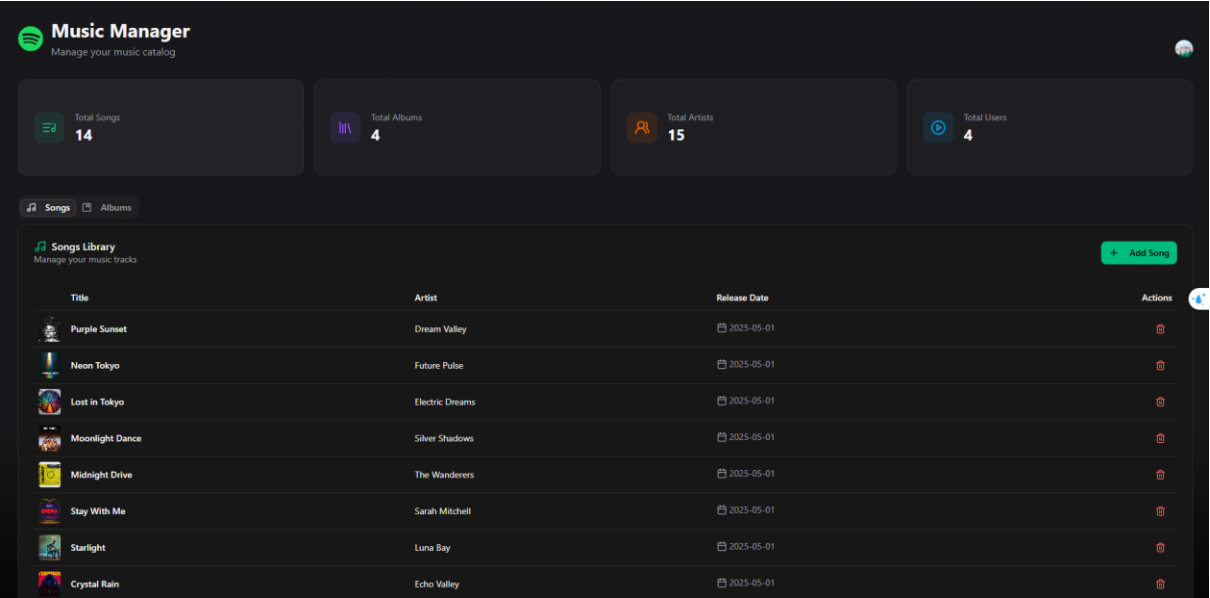
#### IV. HÌNH ẢNH



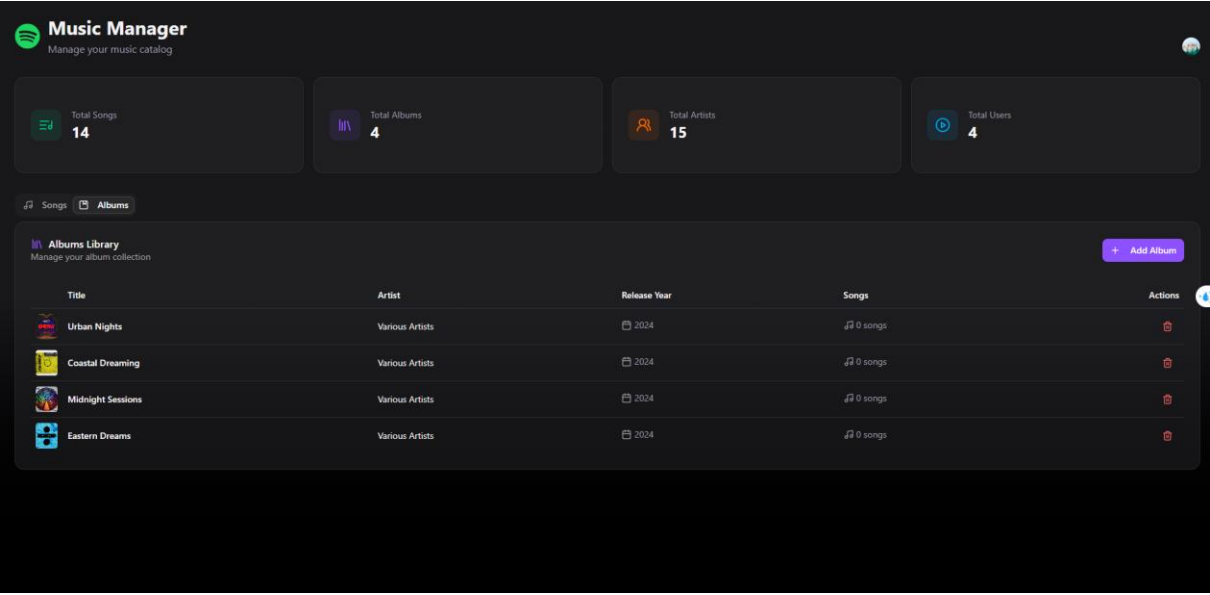
Hình 1: Giao diện trang chủ chưa đăng nhập



Hình 2: Giao diện trang chủ sau khi đăng nhập tài khoản admin



Hình 3: Giao diện quản lý nhạc



Hình 4: Giao diện quản lý album

Add New Song

×

Add a new song to your music library

↑

Upload artwork

Choose File

Audio File

Choose Audio File

Title

Artist

Duration (seconds)

0

Album (Optional)

Select album ▾

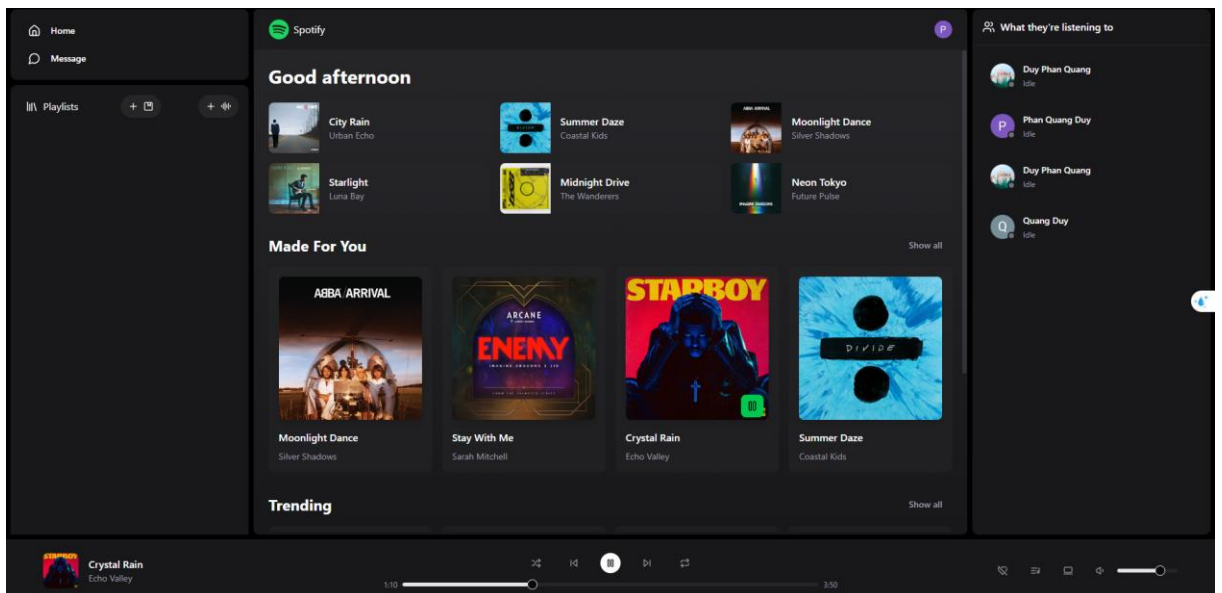
Cancel

Add Song

Hình 5: Giao diện thêm nhạc cho admin







Hình 8: Giao diện người ở trạng thái đang phát nhạc

Add New Album

×

Add a new album to your collection

Upload album artwork

Choose File

Album Title

Enter album title

Artist

Enter artist name

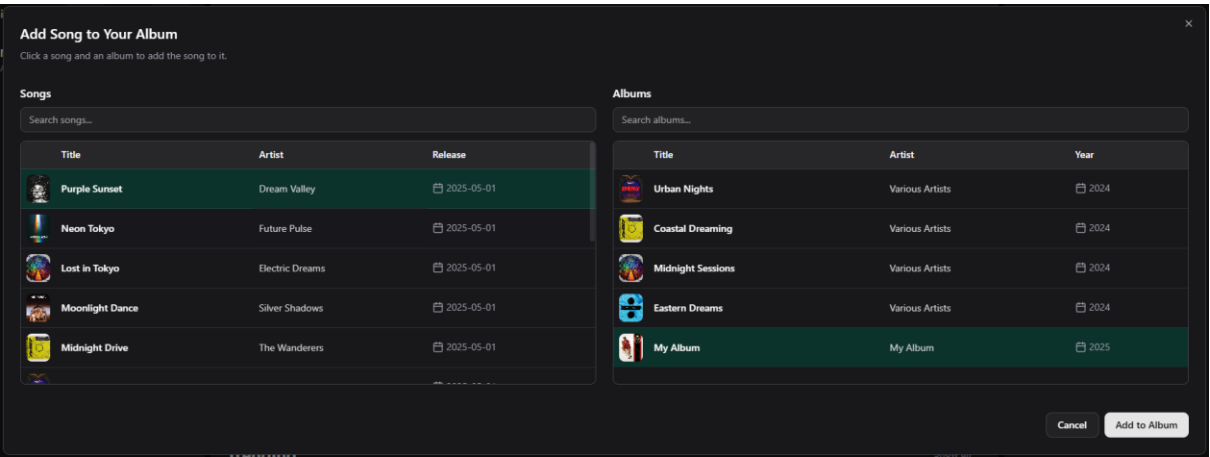
Release Year

2025

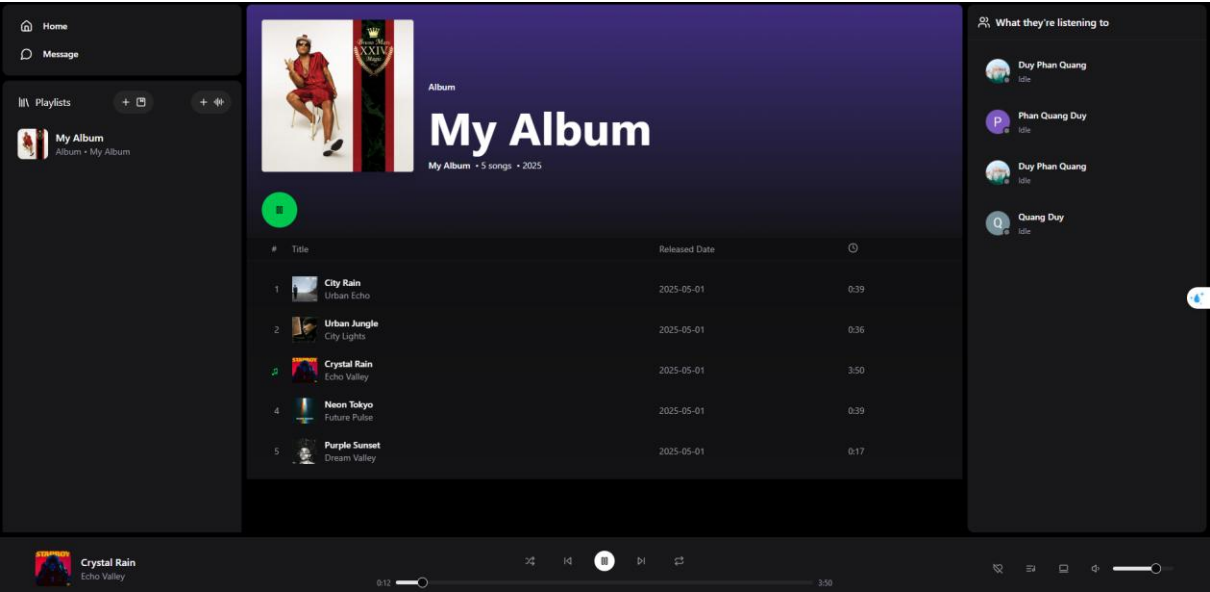
Cancel

Add Album

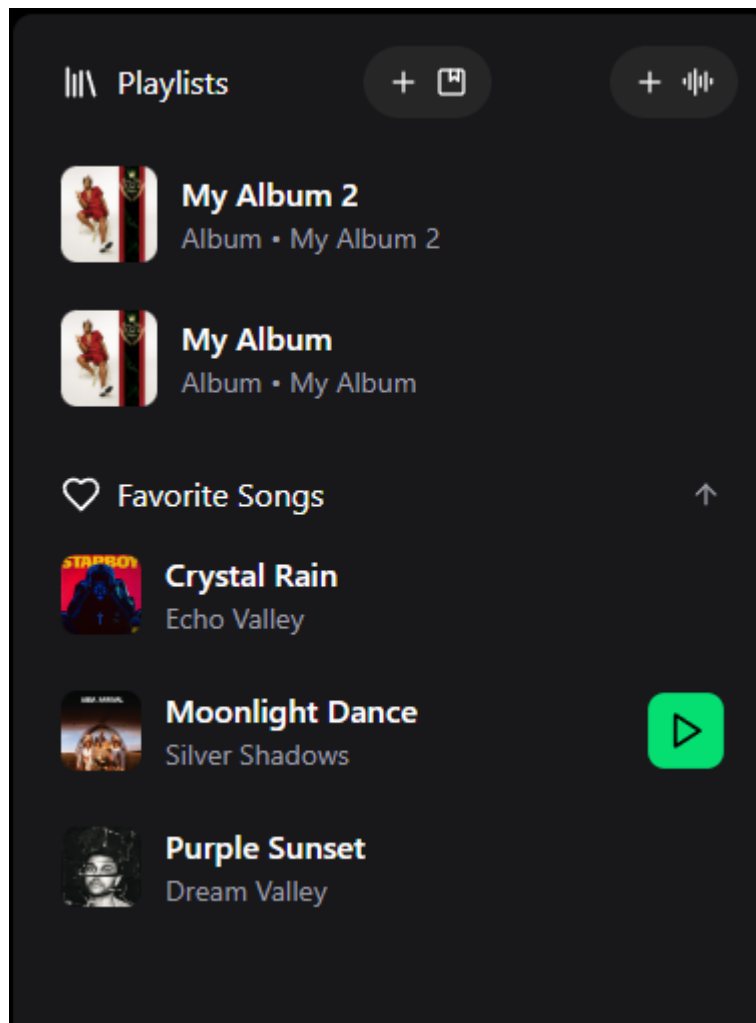
Hình 9: Giao diện tạo album mới cho user



Hình 10: Giao diện thêm nhạc mới vào album cho user



Hình 11: Giao diện album của user

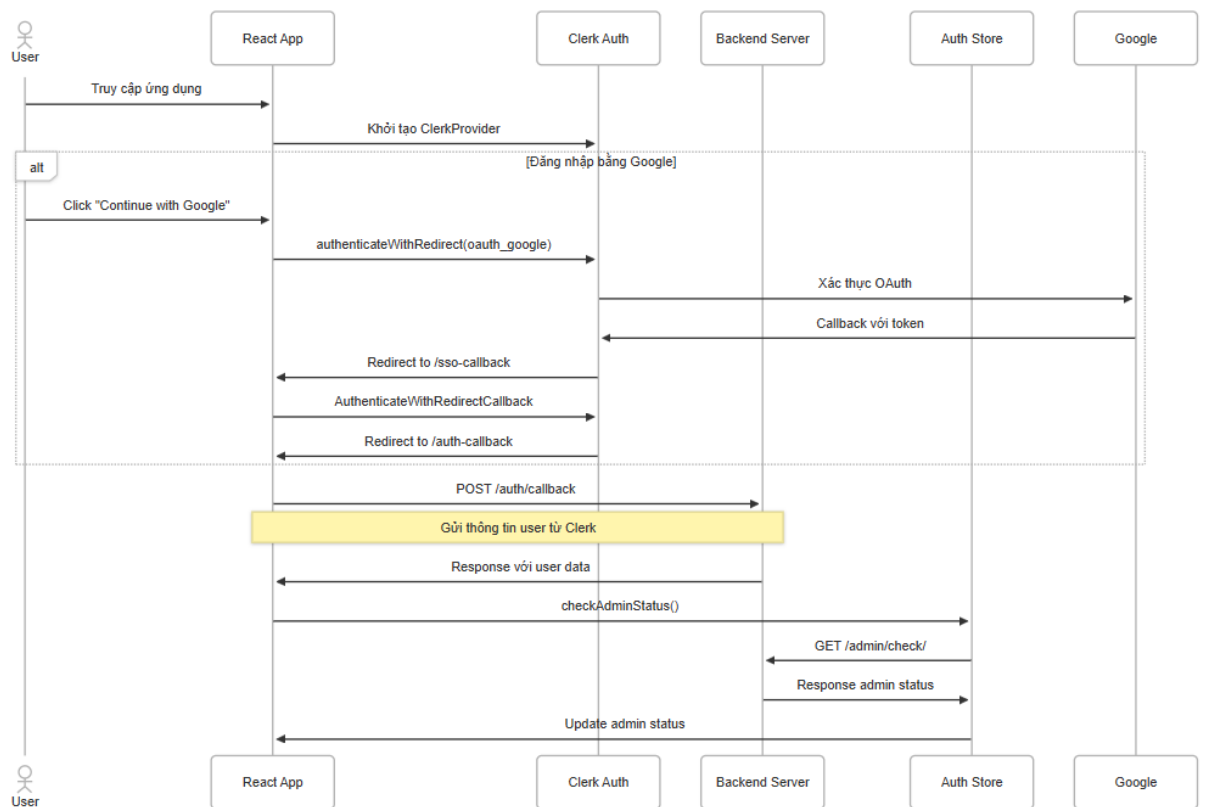


**Hình 12: Giao diện danh sách album và danh sách nhạc yêu thích của user**

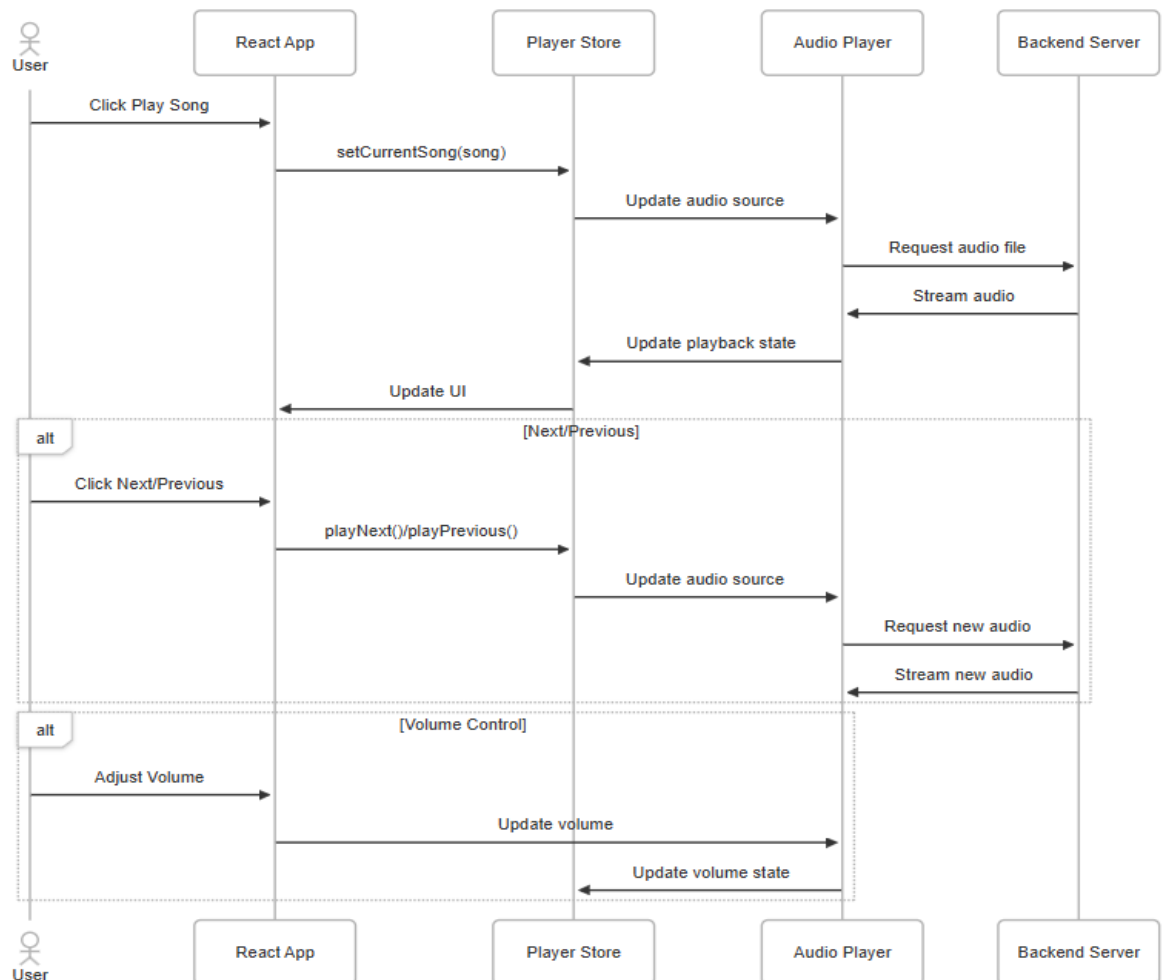
## **V. XỬ LÝ**

### **1. Sơ đồ tuần tự (Sequence Diagram)**

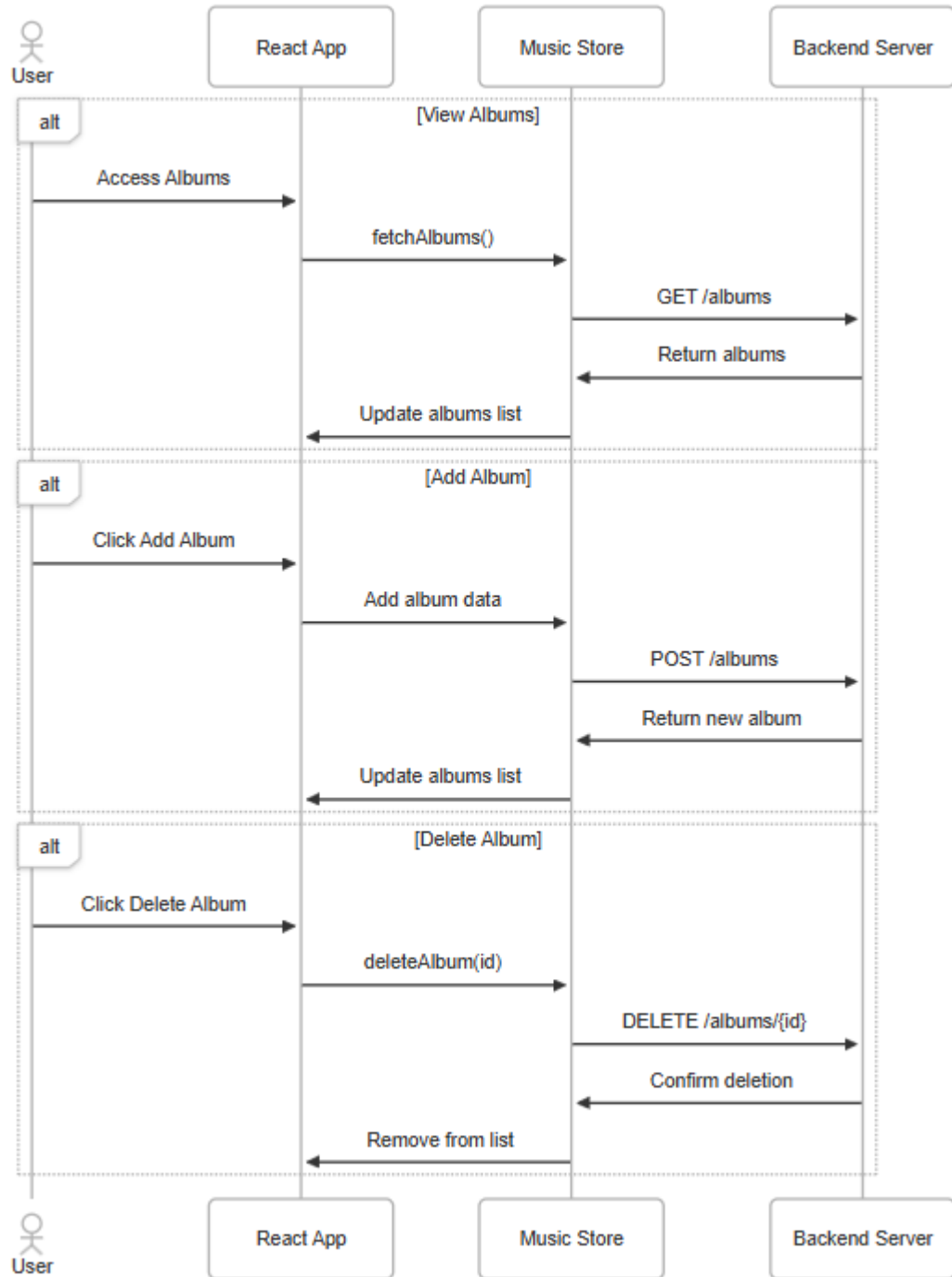
**Xác thực người dùng**



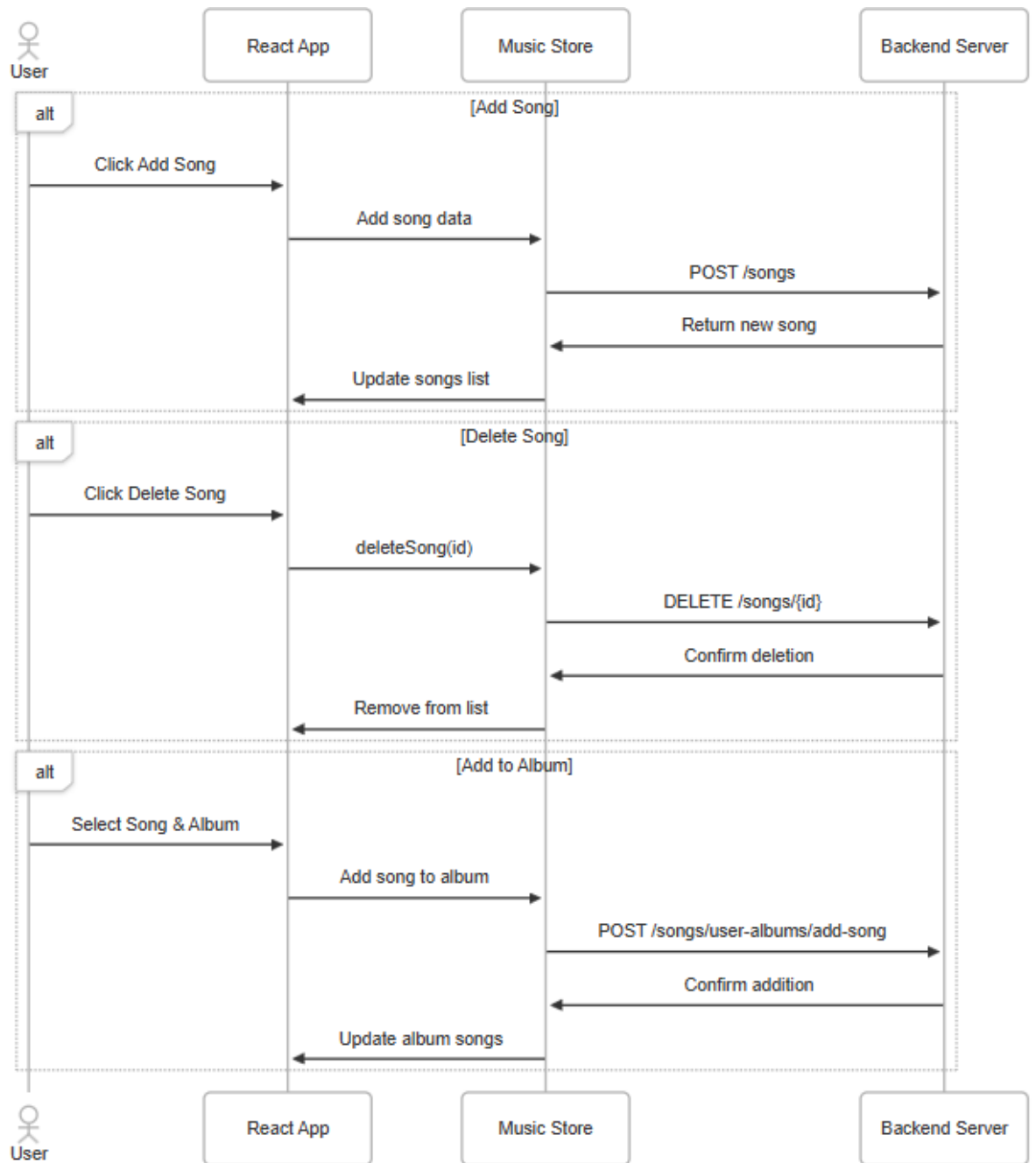
## Phát nhạc



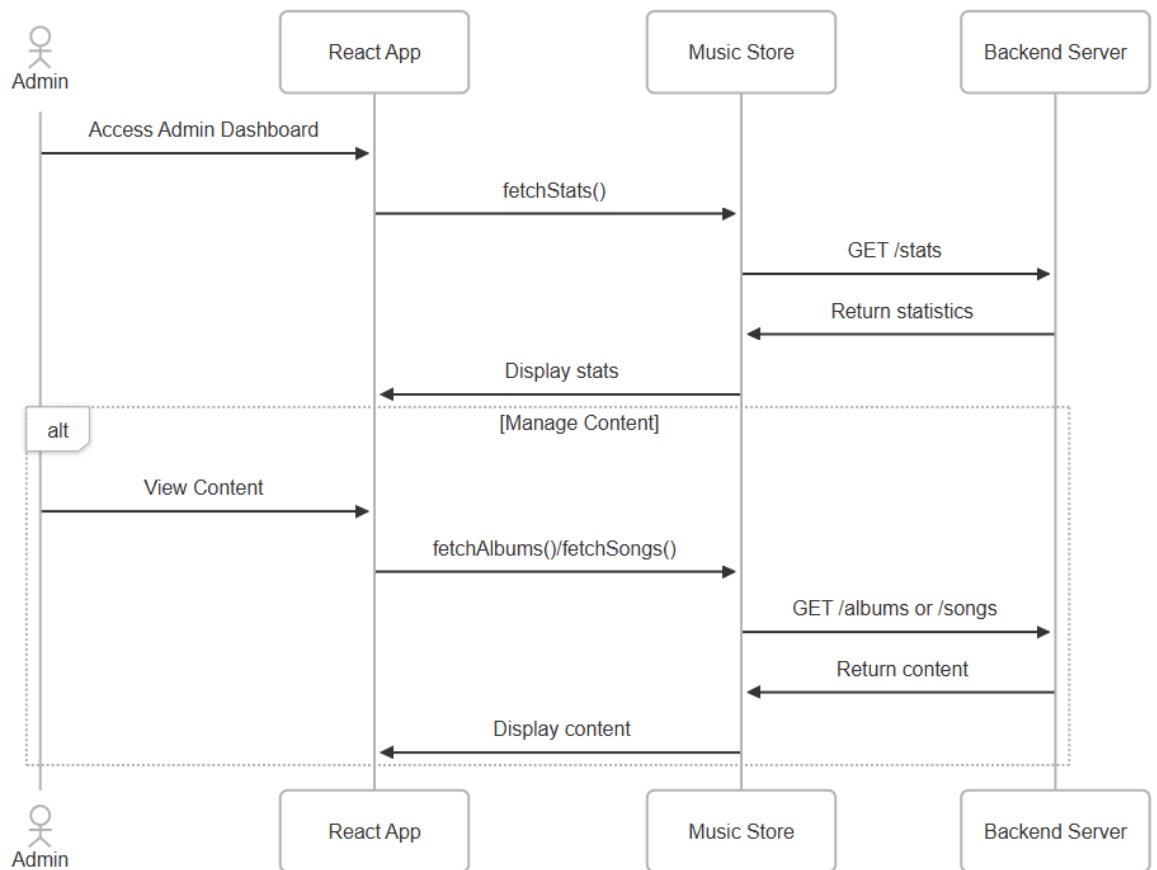
## Quản lý album (admin)



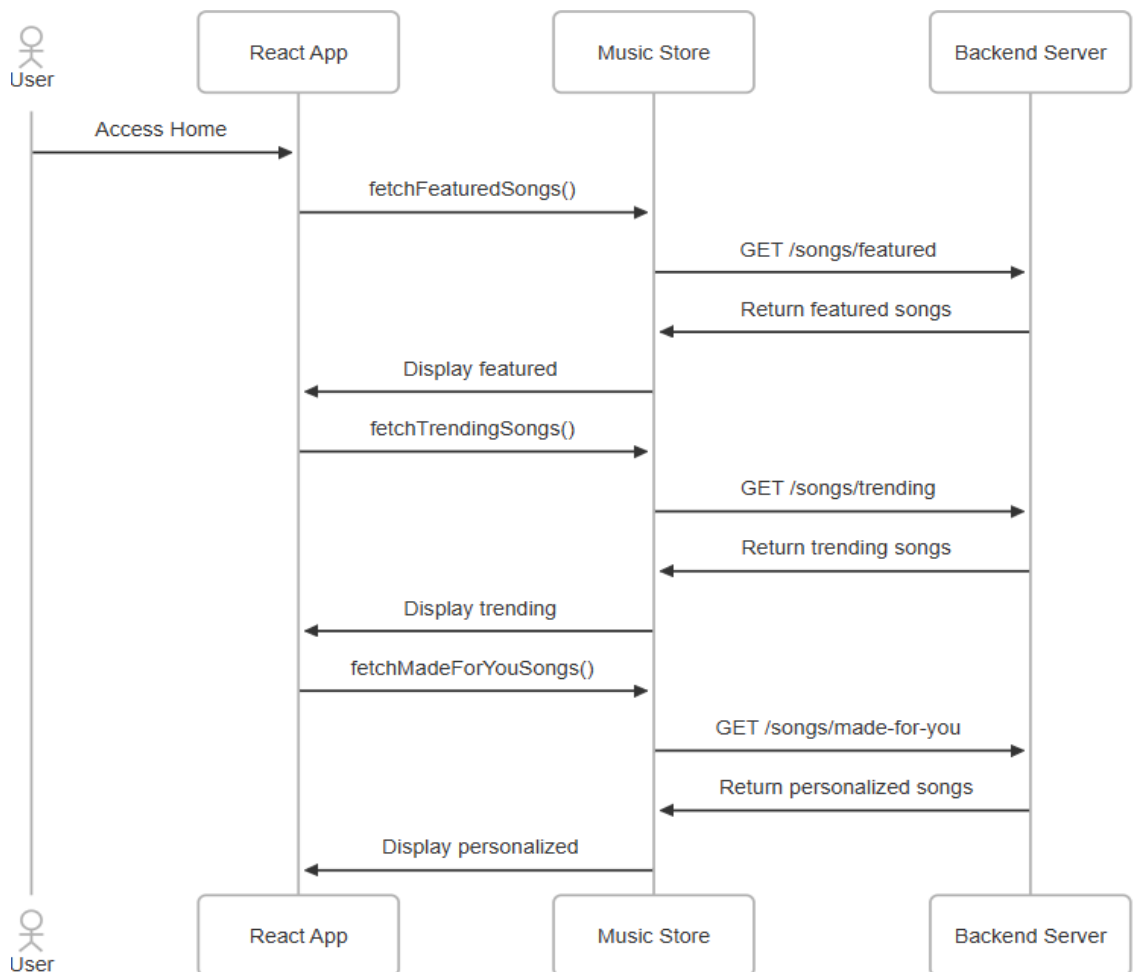
## Quản lý nhạc (admin)



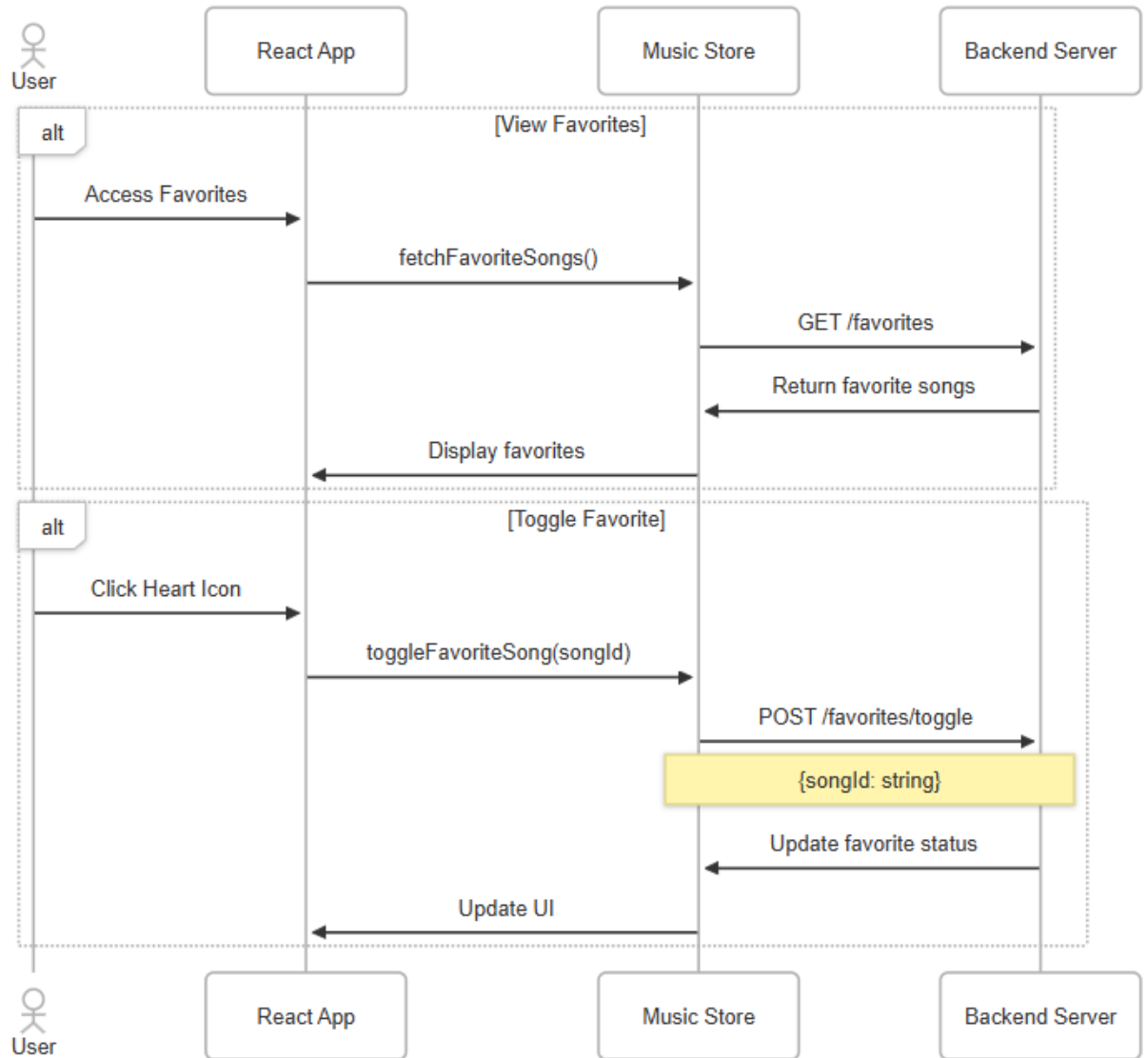
## Admin Dashboard



## Danh sách FeatureSong

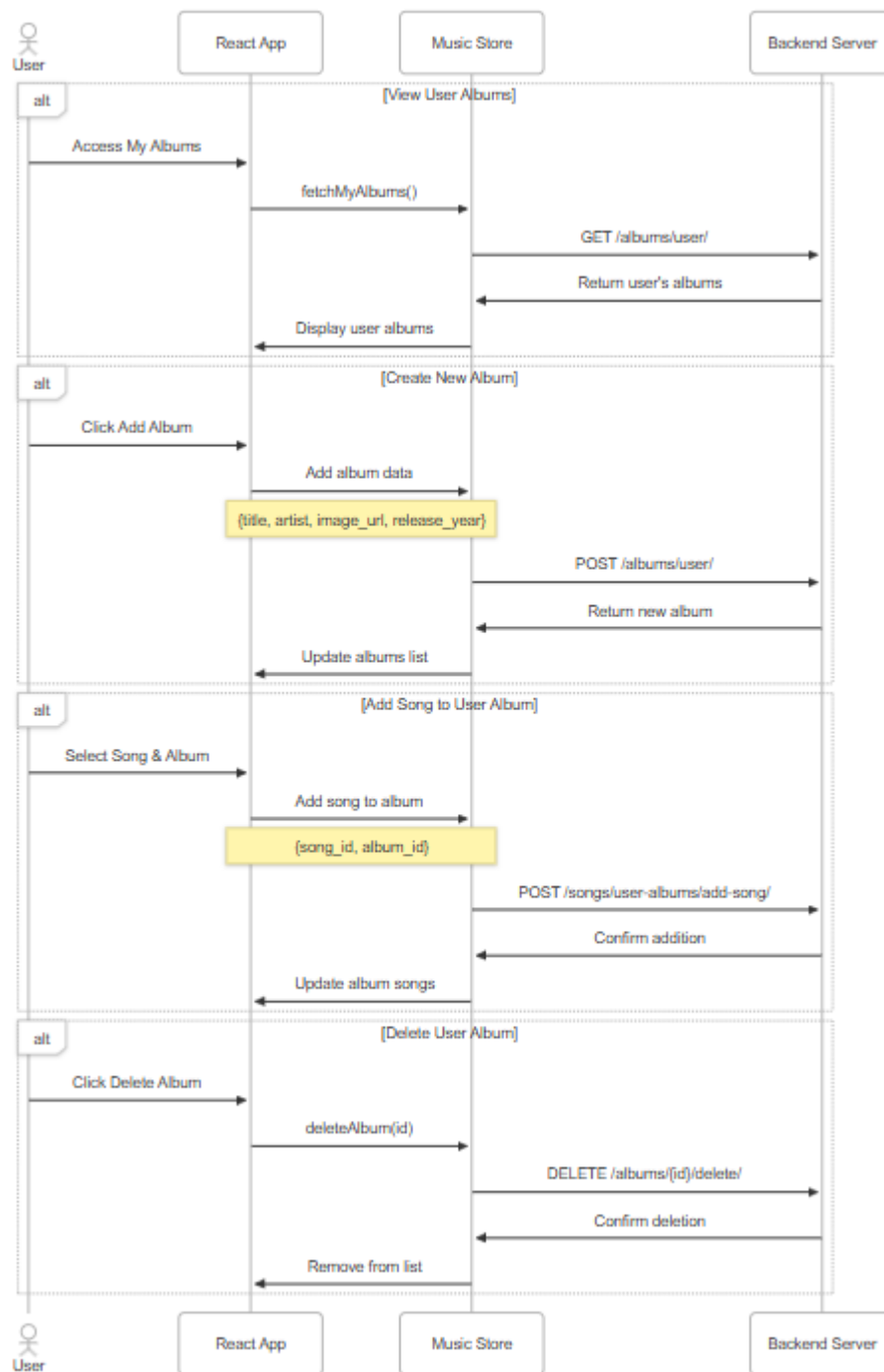


## Thêm/xóa nhạc yêu thích



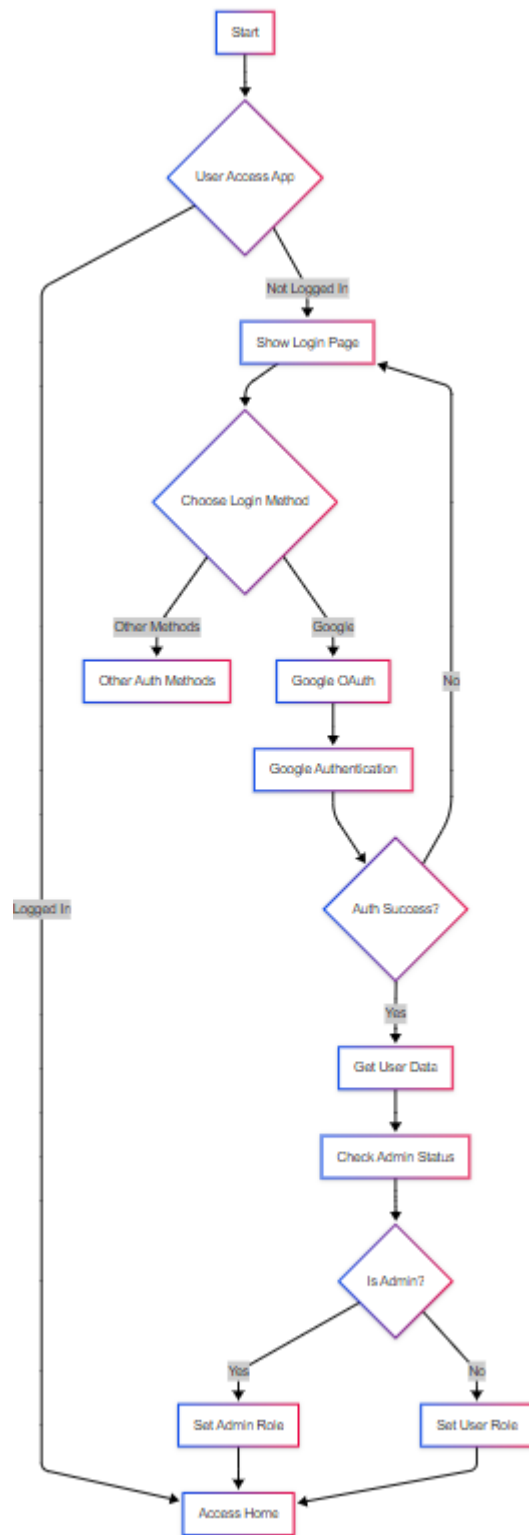
## Thêm album (user)



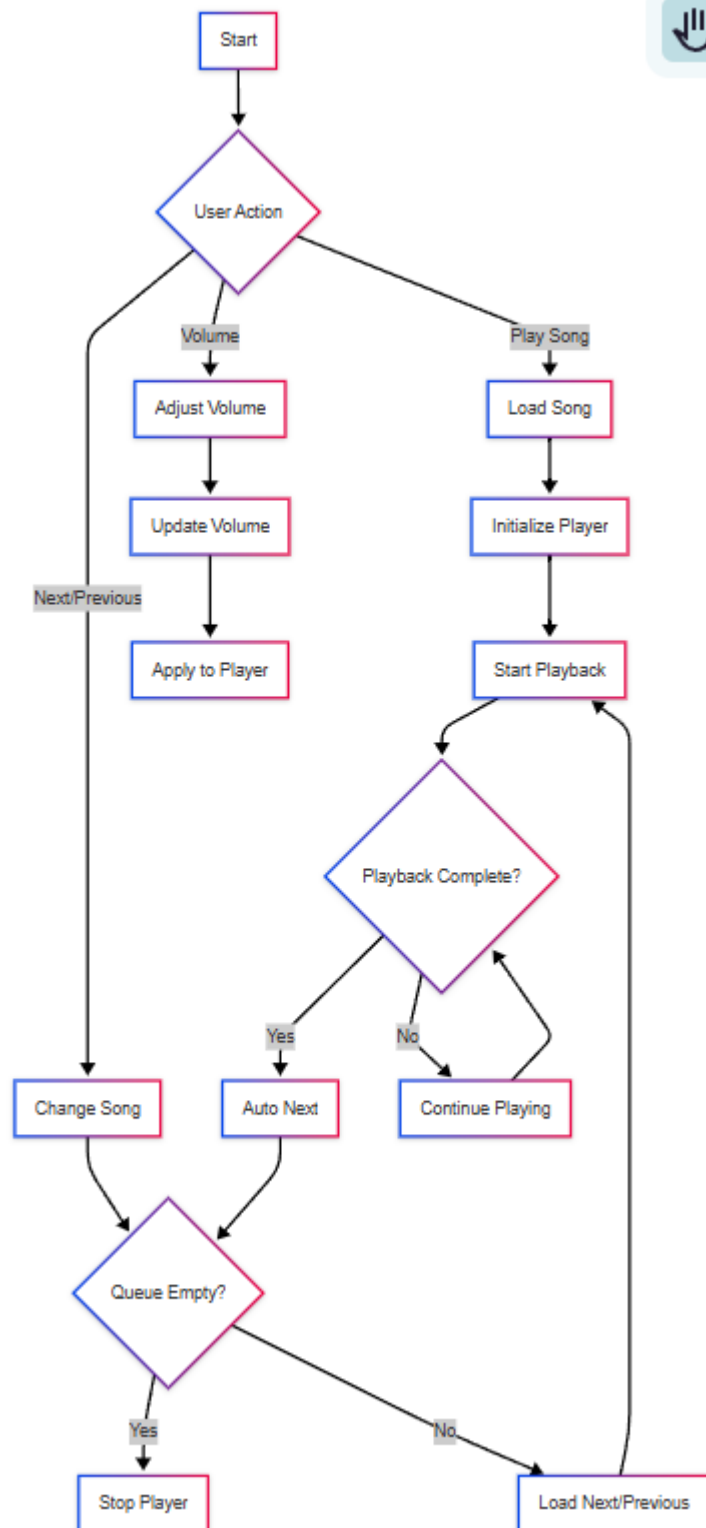


## 2. Sơ đồ hoạt động (Activity Diagram)

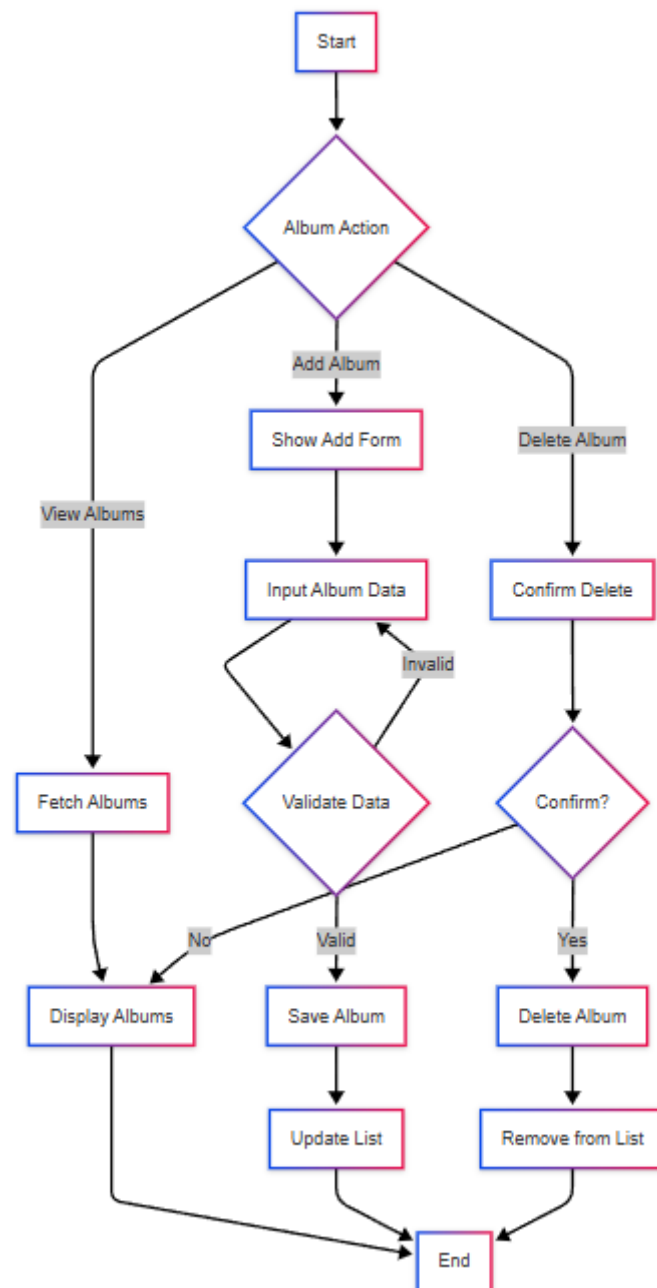
### Xác thực người dùng



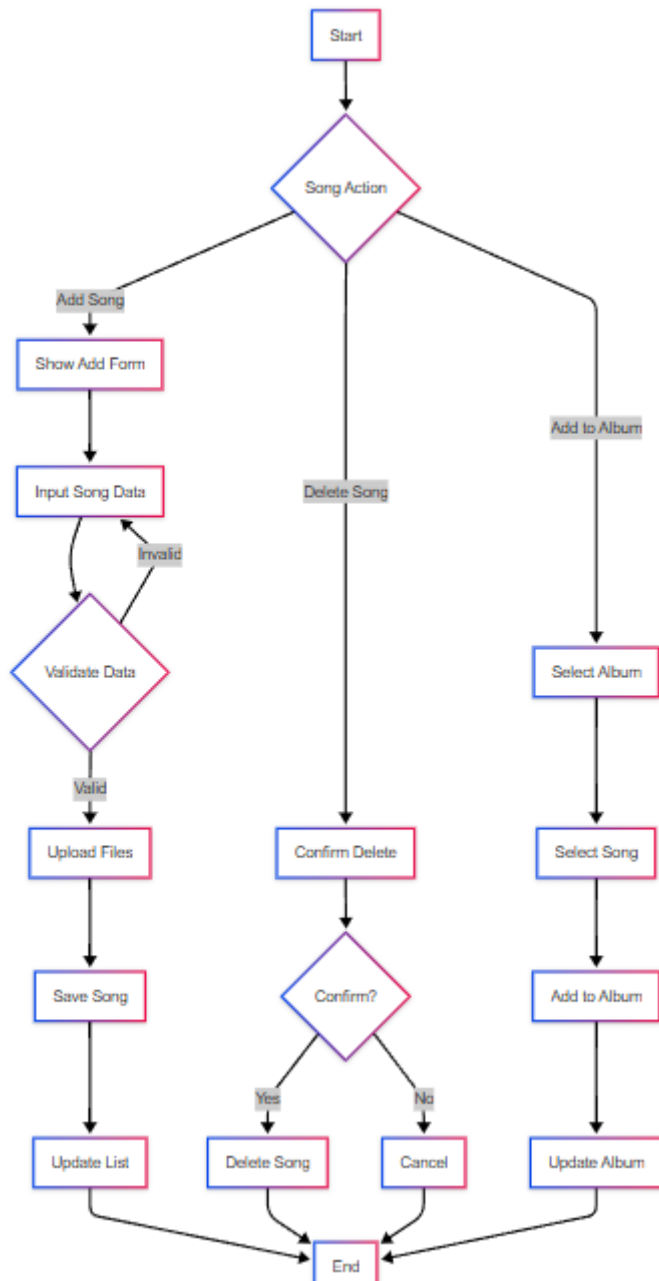
**Phát nhạc**



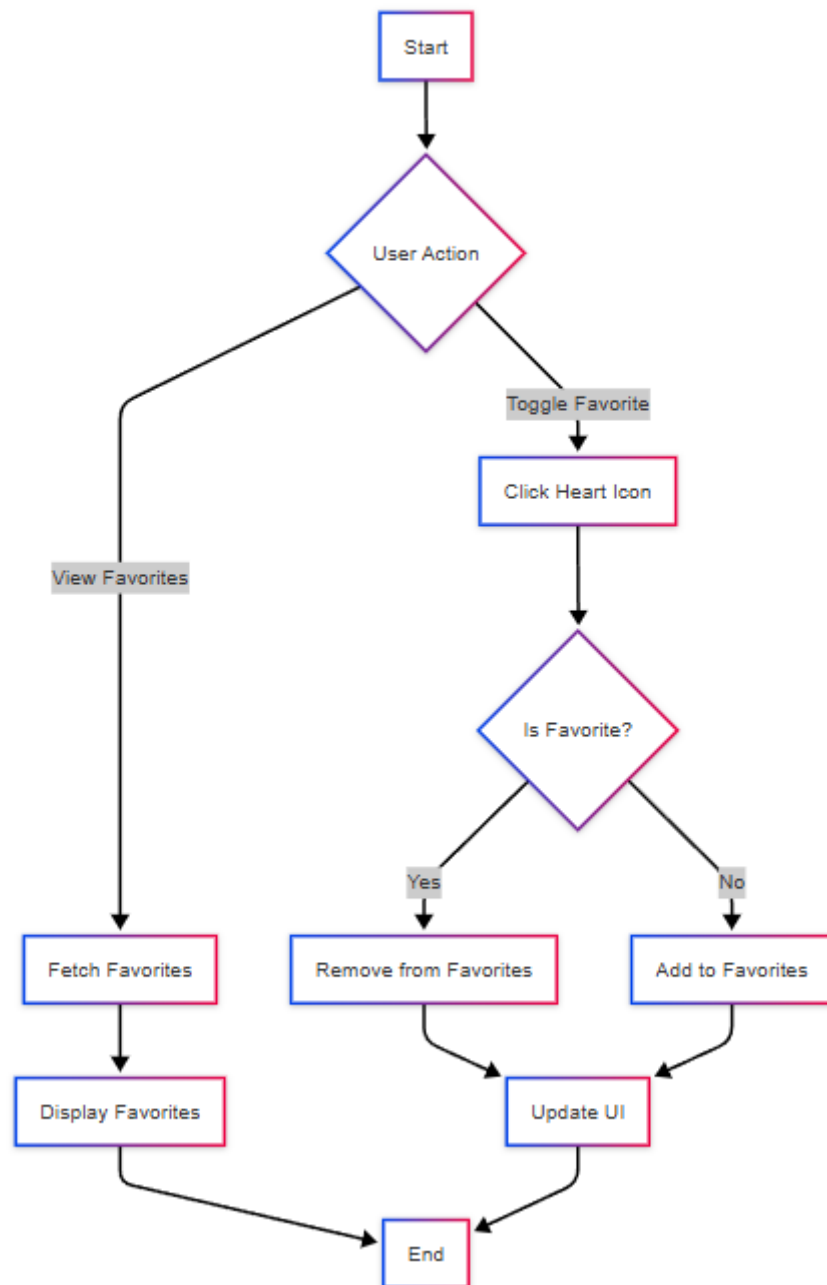
**Thêm album (admin)**



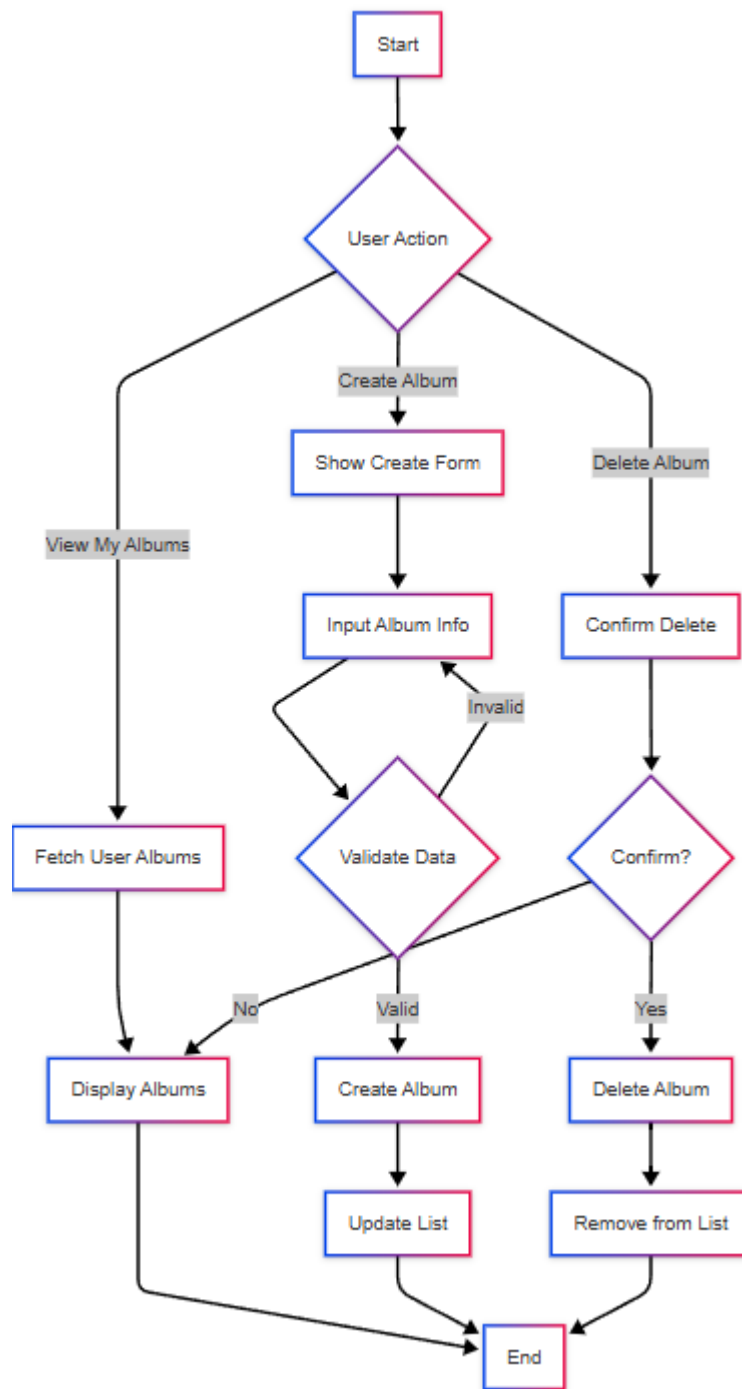
**Thêm nhạc (admin)**



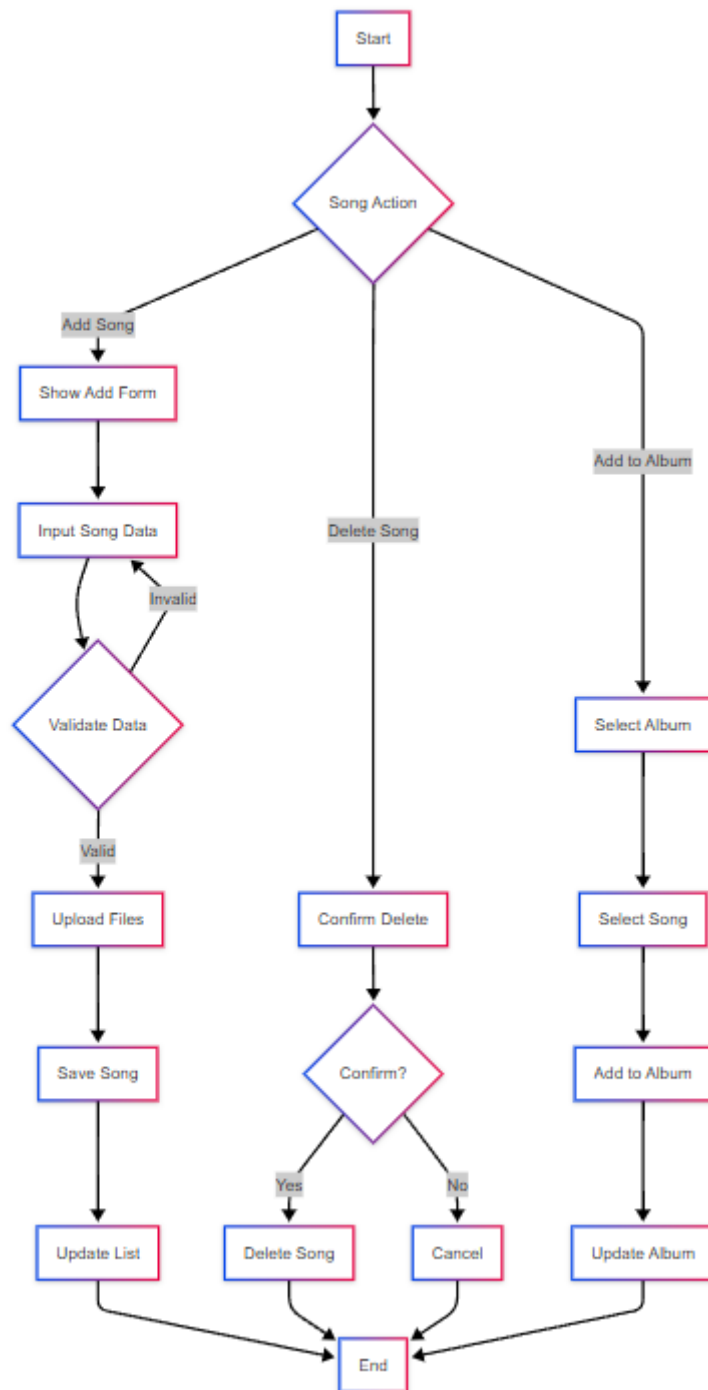
**Thêm/xóa nhạc yêu thích**



**Tạo album (user)**

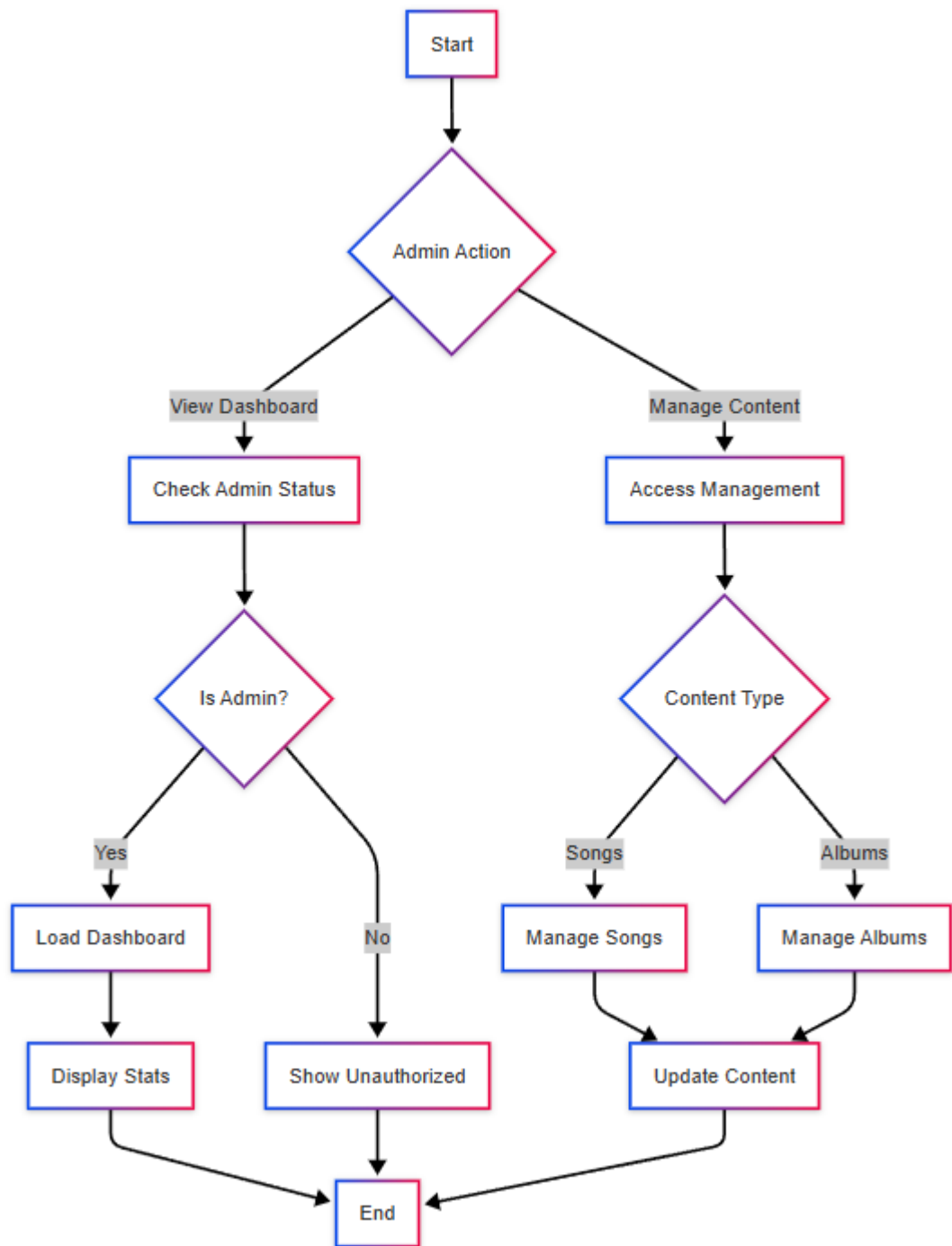


**Thêm nhạc vào album (user)**



## Admin Dashbroard





## VI. CÁCH CÀI ĐẶT ỨNG DỤNG.

### 1. Frontend

#### Yêu cầu hệ thống

Trước khi bắt đầu, hãy đảm bảo bạn đã cài đặt:

- Node.js (phiên bản 14.0.0 trở lên)
- npm hoặc yarn
- Git

- Trình duyệt web hiện đại (Chrome, Firefox, Safari, hoặc Edge)

### **Cài đặt**

1. Clone repository:

```
git clone https://github.com/QDuyPhan/spotify-clone.git
```

```
cd spotify-clone
```

2. Cài đặt các dependencies:

```
npm install
```

*# hoặc*

```
yarn install
```

3. Cấu hình môi trường:

Tạo file .env trong thư mục gốc và thêm các biến môi trường cần thiết:

```
VITE_CLERK_PUBLISHABLE_KEY=your_clerk_publishable_key
```

4. Khởi động máy chủ phát triển:

```
npm run dev
```

*# hoặc*

```
yarn dev
```

Ứng dụng sẽ có sẵn tại <http://localhost:5173>

## **2. Backend**

### **Yêu cầu hệ thống**

- Python 3.10+
- MySQL 8.0+

### **Cài đặt**

Cài đặt gói phụ thuộc:

```
pip install -r requirements.txt
```

Hoặc cài từng gói riêng lẻ:

```
sudo apt install python3-dev default-libmysqlclient-dev build-essential
```

```
pkg-config
```

```
pip install djangorestframework-simplejwt
```

```
pip install PyJWT
```

```
pip install mysqlclient
```

```
pip install django-cloudinary-storage cloudinary
```

Biến môi trường

Tạo file .env.local và thêm các biến sau:

```
CLERK_PUBLISHABLE_KEY="pk_test_..."
```

CLERK\_JWKS\_URL="https://your-app.clerk.accounts.dev/.well-known/jwks.json"

ADMIN\_EMAIL="youremailhere@gmail.com"

### **Cách lấy thông tin Clerk:**

- Đăng ký tại [clerk.dev](https://clerk.dev)
- Tạo ứng dụng mới → API Keys để lấy Publishable Key
- Bật ID Token trong phần JWT Templates
- Copy link JWKS có đuôi như sau: /.well-known/jwks.json

### **Cấu hình Database**

Bước 1: Tạo database trong MySQL

```
CREATE DATABASE spotify_clone;
```

-> Mở MySQL terminal hoặc dùng công cụ như phpMyAdmin/MySQL Workbench để thực hiện lệnh này.

Bước 2: Cấu hình kết nối database trong Django

Trong file settings.py, cập nhật cấu hình như sau:

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.mysql',  
        'NAME': 'spotify_clone',  
        'USER': 'tên_user',  
        'PASSWORD': 'mật_khẩu',  
        'HOST': 'localhost',  
        'PORT': '3306',  
    }  
}
```

Bước 3: Chạy migrate để tạo bảng trong database

```
python manage.py makemigrations
```

```
python manage.py migrate
```

Lệnh này sẽ tạo các bảng trong database đã cấu hình ở bước trên.

Bước 4: Seed dữ liệu vào database

File seeddata.py đã được viết sẵn trong thư

mục your\_app/management/commands/.

```
python manage.py seeddata
```

Lệnh trên sẽ thêm các bài hát và album mẫu vào database.

Bước 5: Kiểm tra dữ liệu trong MySQL

USE spotify\_clone;

SELECT \* FROM spotify\_app\_song;

SELECT \* FROM spotify\_app\_album;

Nếu bạn thấy dữ liệu trong bảng song và album thì bạn đã seed thành công!

## VII. KẾT QUẢ VÀ ĐÁNH GIÁ

Ưu điểm:

- **Chức năng toàn diện:** Ứng dụng cung cấp một bộ chức năng toàn diện, bao gồm các tính năng thiết yếu như đăng ký/đăng nhập, tìm kiếm, phát nhạc, quản lý playlist và thư viện nhạc cá nhân. Điều này đảm bảo rằng người dùng có thể thực hiện hầu hết các tác vụ nghe nhạc cơ bản mà họ mong đợi từ một ứng dụng phát nhạc trực tuyến.
- **Giao diện thân thiện:** Giao diện người dùng được thiết kế trực quan và dễ sử dụng, giúp người dùng ở mọi trình độ kỹ thuật có thể nhanh chóng làm quen và tận hưởng các tính năng của ứng dụng. Thiết kế này tập trung vào trải nghiệm người dùng, giảm thiểu sự phức tạp và tối đa hóa sự hài lòng.
- **Quản lý hiệu quả:** Ứng dụng cung cấp các công cụ quản lý mạnh mẽ cho phép quản trị viên dễ dàng duy trì và cập nhật thư viện nhạc. Điều này đặc biệt quan trọng đối với các ứng dụng âm nhạc, nơi nội dung thường xuyên thay đổi và cần được cập nhật thường xuyên.
- **Hiệu suất ổn định:** Việc lựa chọn và tích hợp các công nghệ hiện đại như React, Django, MySQL và Cloudinary đã góp phần vào sự ổn định và hiệu quả của ứng dụng. Các công nghệ này được biết đến với khả năng mở rộng, hiệu suất và độ tin cậy, đảm bảo rằng ứng dụng có thể xử lý tải người dùng và cung cấp trải nghiệm mượt mà.
- **Bảo mật và tiện lợi:** Hệ thống xác thực người dùng được triển khai thông qua Clerk và OAuth, cung cấp một cách an toàn và tiện lợi để người dùng tạo và truy cập tài khoản. Các công nghệ xác thực này được công nhận rộng rãi về các tính năng bảo mật mạnh mẽ và dễ sử dụng.
- **Cấu trúc mã nguồn tốt:** Cấu trúc mã nguồn của dự án được tổ chức rõ ràng và logic, giúp bảo trì, gỡ lỗi và mở rộng trong tương lai dễ dàng hơn. Cấu trúc mã nguồn tốt là điều cần thiết cho bất kỳ dự án

phần mềm nào, vì nó ảnh hưởng trực tiếp đến khả năng bảo trì, khả năng đọc và khả năng mở rộng của dự án.

Nhược điểm:

- **Thiếu tính năng tương tác xã hội:** Một trong những hạn chế đáng chú ý của ứng dụng là thiếu tính năng chat giữa người dùng. Tương tác xã hội là một khía cạnh quan trọng của nhiều ứng dụng âm nhạc hiện đại, cho phép người dùng kết nối, chia sẻ nhạc và thảo luận về âm nhạc. Việc thiếu tính năng này có thể hạn chế mức độ tương tác và gắn kết của người dùng.
- **Gợi ý nhạc nâng cao:** Ứng dụng hiện tại chưa có các thuật toán gợi ý bài hát nâng cao dựa trên sở thích của người dùng. Các đề xuất được cá nhân hóa đóng một vai trò quan trọng trong việc giúp người dùng khám phá âm nhạc mới và duy trì sự tương tác. Việc triển khai các thuật toán như vậy có thể nâng cao đáng kể trải nghiệm người dùng.
- **Hạn chế về nguồn nhạc:** Ứng dụng hiện tại không hỗ trợ phát nhạc trực tiếp từ các nguồn bên ngoài. Hạn chế này có thể giới hạn phạm vi lựa chọn âm nhạc có sẵn cho người dùng.
- **Khả năng mở rộng hiệu suất:** Mặc dù ứng dụng hiện tại hoạt động tốt, nhưng hiệu suất của nó có thể bị ảnh hưởng khi số lượng người dùng tăng đáng kể. Để đảm bảo rằng ứng dụng có thể xử lý tải người dùng ngày càng tăng, cần phải thực hiện các tối ưu hóa, chẳng hạn như bộ nhớ đệm và cân bằng tải.

Hướng phát triển trong tương lai:

- **Tương tác xã hội:** Ưu tiên hàng đầu là bổ sung tính năng chat để cho phép người dùng kết nối, chia sẻ nhạc và thảo luận về sở thích âm nhạc của họ. Điều này sẽ tạo ra một cộng đồng xung quanh ứng dụng và tăng cường sự tương tác của người dùng.
- **Gợi ý âm nhạc thông minh:** Tích hợp các thuật toán gợi ý nhạc thông minh để cung cấp các đề xuất được cá nhân hóa cho người dùng. Điều này sẽ giúp người dùng khám phá âm nhạc mới phù hợp với sở thích của họ và cải thiện mức độ tương tác tổng thể của ứng dụng.
- **Tối ưu hóa hiệu suất:** Thực hiện các kỹ thuật bộ nhớ đệm và cân bằng tải để cải thiện khả năng mở rộng và hiệu suất của ứng dụng. Điều này

sẽ đảm bảo rằng ứng dụng có thể xử lý số lượng lớn người dùng và duy trì thời gian phản hồi nhanh.

- **Mở rộng thư viện nhạc:** Mở rộng thư viện nhạc có sẵn trên nền tảng để cung cấp cho người dùng nhiều lựa chọn hơn. Điều này có thể bao gồm việc cấp phép nhạc từ nhiều nguồn khác nhau hoặc cho phép người dùng tải lên nhạc của riêng họ.
- **Hỗ trợ Streaming từ nhiều nguồn:** Khám phá các tùy chọn để hỗ trợ phát nhạc trực tiếp từ các nguồn bên ngoài, chẳng hạn như các dịch vụ phát nhạc trực tuyến khác hoặc máy chủ của người dùng. Điều này sẽ cung cấp cho người dùng quyền truy cập vào nhiều lựa chọn âm nhạc hơn và tăng tính linh hoạt của ứng dụng.

## **VIII. TÀI LIỆU THAM KHẢO**

<https://react.dev/learn>

<https://docs.djangoproject.com/en/5.2/>

<https://clerk.com/docs/quickstarts/react>

<https://ui.shadcn.com/docs/components/dropdown-menu>

<https://www.w3schools.com/js/>

<https://www.w3schools.com/django/index.php>

<https://www.w3schools.com/react/default.asp>

## **LỜI KẾT**

Sau khi trình bày toàn bộ nội dung của đề án, chúng em xin chân thành cảm ơn giảng viên hướng dẫn đã dành thời gian quý báu để theo dõi và đánh giá bài báo cáo của nhóm chúng em. Trong quá trình thực hiện, dù đã cố gắng hết sức, chúng em vẫn có thể không tránh khỏi một số sai sót hay hạn chế. Vì vậy, chúng em rất mong nhận được những góp ý, hướng dẫn quý báu từ Thầy, để từ đó có thể tiếp tục hoàn thiện và nâng cao chất lượng trang web trong tương lai. Một lần nữa, xin chân thành cảm ơn Thầy vì sự hỗ trợ và chỉ dẫn tận tình của Thầy trong suốt quá trình thực hiện đề án này