

**TRƯỜNG ĐẠI HỌC THỦY LỢI**  
**KHOA CÔNG NGHỆ THÔNG TIN**



# **BÀI TẬP LỚN**

**HỌC PHẦN: PHÁT TRIỂN ỨNG DỤNG CHO THIẾT BỊ DI ĐỘNG**

**ĐỀ TÀI: Xây dựng ứng dụng Nghe nhạc SoundMe**

<b>Mã Sinh Viên</b>	<b>Họ và Tên</b>	<b>Ngày Sinh</b>	<b>Lớp</b>
2151173752	Vũ Thị Yến Chi	10/10/2003	63KTPM2
2151173770	Dương Văn Hiến	22/07/2003	63KTPM2
2151173788	Nguyễn Trung Kiên	27/02/2003	63KTPM2
2151173792	Phạm Ngọc Hải Linh	20/07/2003	63KTPM2

**Hà Nội, năm 2024**

## LỜI NÓI ĐẦU

Trong thời đại công nghệ số phát triển nhanh chóng, nhu cầu giải trí và thưởng thức âm nhạc của con người ngày càng cao. Ứng dụng nghe nhạc không chỉ đơn thuần là một công cụ phát nhạc, mà còn là nơi để người dùng khám phá các thể loại âm nhạc mới, tương tác với cộng đồng yêu âm nhạc và cá nhân hóa trải nghiệm nghe nhạc của riêng mình. Trên cơ sở đó, nhóm chúng tôi đã lựa chọn đề tài "Thiết kế ứng dụng nghe nhạc" nhằm phát triển một nền tảng nghe nhạc tiện lợi, thân thiện với người dùng và tích hợp những tính năng đáp ứng nhu cầu đa dạng của người sử dụng.

Mục tiêu của dự án này là thiết kế một ứng dụng nghe nhạc "SoundMe" hiện đại, dễ sử dụng với giao diện trực quan và tính năng phong phú. Trong quá trình xây dựng ứng dụng, chúng tôi tập trung vào việc phân tích nhu cầu của người dùng, từ đó đưa ra các giải pháp thiết kế phù hợp để tạo ra một trải nghiệm âm nhạc thú vị, dễ tiếp cận và có tính cá nhân hóa cao.

Chúng tôi hy vọng rằng ứng dụng nghe nhạc này sẽ mang lại trải nghiệm âm nhạc tốt nhất, đáp ứng kỳ vọng của người dùng và góp phần vào sự phát triển của ngành công nghiệp âm nhạc kỹ thuật số.

# MỤC LỤC

<b>MỤC LỤC</b>	<b>3</b>
<b>CHƯƠNG 1. MÔ TẢ BÀI TOÁN</b>	<b>5</b>
1.1. Mục tiêu	5
1.2. Chức năng chính	5
1.3. Yêu cầu kỹ thuật	6
1.4. Quy trình hoạt động	6
<b>CHƯƠNG 2. PHÂN TÍCH YÊU CẦU VÀ THIẾT KẾ HỆ THỐNG</b>	<b>7</b>
2.1. Phân tích yêu cầu	7
2.2. Thiết kế hệ thống	10
- Lớp <b>FragmentFullPlayer</b> :	<b>11</b>
2.2.3. Thiết kế cơ sở dữ liệu	12
<b>CHƯƠNG 3. KẾT QUẢ THỰC HIỆN</b>	<b>14</b>
3.1. Công nghệ đã sử dụng	14
3.2. Tiến độ thực hiện	14
3.3. Hình ảnh sản phẩm	15



**Bảng phân chia công việc**

Thành viên	Công việc
Vũ Thị Yến Chi	Chức năng tìm kiếm bài hát và xem chi tiết bài hát
Phạm Ngọc Hải Linh	Chức năng xử lý thao tác phát nhạc
Dương Văn Hiến	Chức năng xử lý thông báo và nạp bài hát vào background
Nguyễn Trung Kiên	Chức năng lưu bài hát yêu thích và giao diện đĩa nhạc

# CHƯƠNG 1. MÔ TẢ BÀI TOÁN

## 1.1. Mục tiêu

- Xây dựng một ứng dụng nghe nhạc trên điện thoại di động có kết nối internet, giúp người dùng có thể nghe nhạc ở mọi lúc mọi nơi.
- Ứng dụng nghe nhạc không có quảng cáo, thư viện nhạc được mua bản quyền từ những đơn vị sản xuất nhạc uy tín hoặc có thể được lấy từ những bản thu âm chất lượng do chúng em sản xuất ra.
- Ứng dụng giúp mọi người xả stress sau những giờ làm việc học tập mệt mỏi.

## 1.2. Chức năng chính

- Phát nhạc:
  - + Với tư cách là một người dùng, tôi muốn phát các bài hát từ bộ sưu tập cá nhân của mình để tôi có thể nghe nhạc bất cứ lúc nào, ở bất cứ đâu.
  - + Với tư cách là một người dùng, tôi muốn có thể phát, tạm dừng, tiếp tục và dừng bài hát, để tôi có thể điều khiển quá trình phát nhạc theo ý muốn.
- Tìm kiếm bài hát:
  - + Với tư cách là một người dùng, tôi muốn tìm kiếm bài hát theo tên hoặc thể loại, để tôi có thể nhanh chóng tìm thấy bài hát mình muốn.
- Phân loại bài hát:
  - + Với tư cách là một người dùng, tôi muốn lọc và sắp xếp thư viện nhạc của mình theo tên, nghệ sĩ, album hoặc thể loại, để tôi có thể dễ dàng quản lý và tìm kiếm bài hát.
- Phát nhạc trong nền:
  - + Với tư cách là một người dùng, tôi muốn tiếp tục nghe nhạc ngay cả khi tôi thoát khỏi ứng dụng, để tôi có thể vừa nghe nhạc vừa làm các công việc khác trên điện thoại.
- Điều khiển nhạc trên thông báo:
  - + Với tư cách là một người dùng, tôi muốn có thể điều khiển phát nhạc trực tiếp từ thông báo, để tôi có thể nhanh chóng tạm dừng, phát, bỏ qua hoặc quay lại bài hát trước đó mà không cần mở ứng dụng.
- Danh sách phát yêu thích:

- + Với tư cách là một người dùng, tôi muốn tạo và quản lý các danh sách phát các bài hát yêu thích của mình, để tôi có thể dễ dàng truy cập chúng sau này.
- Chuyển bài hát:
  - + Với tư cách là một người dùng, tôi muốn bỏ qua sang bài hát tiếp theo hoặc quay lại bài hát trước đó, để tôi có thể nhanh chóng thay đổi bài hát.
- Tua bài hát:
  - + Với tư cách là một người dùng, tôi muốn tua đến một điểm cụ thể trong bài hát, để tôi có thể bỏ qua đến phần mình yêu thích.
- Hiện thị bìa nhạc:
  - + Với tư cách là một người dùng, tôi muốn xem bìa album của bài hát đang phát, để tôi có trải nghiệm nghe nhạc sống động hơn.

### **1.3. Yêu cầu kỹ thuật**

- Phát triển trên nền tảng Android, sử dụng Android Java, dùng Firebase Realtime Database để lưu thông tin bài hát, Firebase Authentication để đăng nhập bằng tài khoản gmail, Firebase Storage lưu file nhạc, ảnh bìa nhạc.

### **1.4. Quy trình hoạt động**

- Người dùng đăng nhập vào ứng dụng, tìm kiếm bài hát muốn nghe chọn bài hát muốn phát. Nếu yêu thích bài hát đó có thể thêm bài hát vào danh sách phát yêu thích.

## CHƯƠNG 2. PHÂN TÍCH YÊU CẦU VÀ THIẾT KẾ HỆ THỐNG

### 2.1. Phân tích yêu cầu

#### Xác định người dùng:

- Người dùng cuối: Bất kỳ ai muốn sử dụng ứng dụng **SoundMe** để nghe nhạc. Họ có thể là học sinh, sinh viên, nhân viên văn phòng hoặc người yêu thích âm nhạc.
- Mức độ hiểu biết về công nghệ của người dùng có thể khác nhau, do đó ứng dụng cần dễ sử dụng và có giao diện thân thiện với mọi đối tượng người dùng.

#### Thu thập yêu cầu

Các chức năng chính được xác định dựa trên nhu cầu nghe nhạc của người dùng:

- Phát nhạc:
  - + Với tư cách là một người dùng, tôi muốn phát các bài hát từ bộ sưu tập cá nhân của mình để tôi có thể nghe nhạc bất cứ lúc nào, ở bất cứ đâu.
  - + Với tư cách là một người dùng, tôi muốn có thể phát, tạm dừng, tiếp tục và dừng bài hát, để tôi có thể điều khiển quá trình phát nhạc theo ý muốn.
- Tìm kiếm bài hát:
  - + Với tư cách là một người dùng, tôi muốn tìm kiếm bài hát theo tên hoặc thể loại, để tôi có thể nhanh chóng tìm thấy bài hát mình muốn.
- Phân loại bài hát:
  - + Với tư cách là một người dùng, tôi muốn lọc và sắp xếp thư viện nhạc của mình theo tên, nghệ sĩ, album hoặc thể loại, để tôi có thể dễ dàng quản lý và tìm kiếm bài hát.
- Phát nhạc trong nền:
  - + Với tư cách là một người dùng, tôi muốn tiếp tục nghe nhạc ngay cả khi tôi thoát khỏi ứng dụng, để tôi có thể vừa nghe nhạc vừa làm các công việc khác trên điện thoại.
- Điều khiển nhạc trên thông báo:
  - + Với tư cách là một người dùng, tôi muốn có thể điều khiển phát nhạc trực tiếp từ thông báo, để tôi có thể nhanh chóng tạm dừng,



phát, bỏ qua hoặc quay lại bài hát trước đó mà không cần mở ứng dụng.

- Danh sách phát yêu thích:
  - + Với tư cách là một người dùng, tôi muốn tạo và quản lý các danh sách phát các bài hát yêu thích của mình, để tôi có thể dễ dàng truy cập chúng sau này.
- Chuyển bài hát:
  - + Với tư cách là một người dùng, tôi muốn bỏ qua sang bài hát tiếp theo hoặc quay lại bài hát trước đó, để tôi có thể nhanh chóng thay đổi bài hát.
- Tua bài hát:
  - + Với tư cách là một người dùng, tôi muốn tua đến một điểm cụ thể trong bài hát, để tôi có thể bỏ qua đến phần mình yêu thích.
- Hiện thị bài nhạc:
  - + Với tư cách là một người dùng, tôi muốn xem bìa album của bài hát đang phát, để tôi có trải nghiệm nghe nhạc sống động hơn.

## Phân tích yêu cầu:

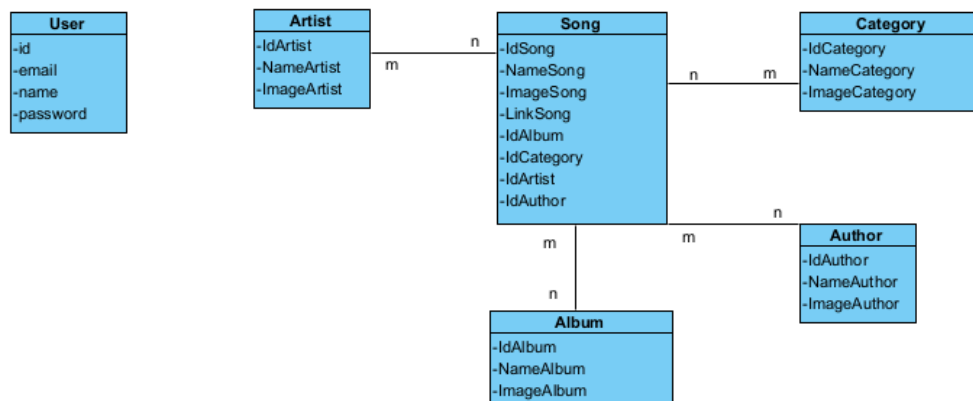
### - Mô hình hóa tương tác:

uc [Tổng quát]



*Biểu đồ Use Case tổng quát*

### - Mô hình đối tượng



*Biểu đồ lớp*

## 2.2. Thiết kế hệ thống

- Kiến trúc hệ thống:
  - + Client: Ứng dụng di động (Android).
  - + Server: Firebase xử lý các yêu cầu từ client.
  - + Database: Firebase Realtime Database lưu trữ thông tin bài hát, ca sĩ, nhạc sĩ, thể loại...
  - + Firebase Storage: lưu trữ các file nhạc, ảnh bìa của bài hát.
- Mô hình thiết kế:
  - + MVC (Model-View-Controller):
- Model:
  - Song: Class ứng với cấu trúc json Song trong Firebase realtime database
  - Artist: Class ứng với cấu trúc json Artist trong Firebase realtime database
  - Category: Class ứng với cấu trúc json Category trong Firebase realtime database
  - Author: Class ứng với cấu trúc json Author trong Firebase realtime database
  - Album: Class ứng với cấu trúc json Album trong Firebase realtime database
- View:
  - ActivitySignInBinding: Bao gồm các thuộc tính ánh xạ đến các thành phần UI trong file activity\_sign\_in.xml.
  - ActivitySignUpBinding: Bao gồm các thuộc tính ánh xạ đến các thành phần UI trong file activity\_sign\_up.xml.
  - ActivityForgotPasswordBinding: Bao gồm các thuộc tính ánh xạ đến các thành phần UI trong file activity\_sign\_in.xml.
  - ActivitySplashScreenBinding: Bao gồm các thuộc tính ánh xạ đến các thành phần UI trong file activity\_splash.xml.
  - FragmentDiscoverBinding: Bao gồm các thuộc tính ánh xạ đến các thành phần UI trong file fragment\_discover.xml.
  - FragmentSearchBinding: Bao gồm các thuộc tính ánh xạ đến các thành phần UI trong file fragment\_search.xml.
  - FragmentLibraryBinding: Bao gồm các thuộc tính ánh xạ đến các thành phần UI trong file fragment\_library.xml.
  - FragmentFullPlayerBinding: Bao gồm các thuộc tính ánh xạ đến các thành phần UI trong file fragment\_full\_player.xml.

- FragmentDetailBinding: Bao gồm các thuộc tính ánh xạ đến các thành phần UI trong file fragment\_detail.xml.
- Controller
  - SplashScreenActivity
  - SignInActivity
  - SignUpActivity
  - ForgotPasswordActivity
  - FragmentDiscover
  - FragmentSearch
  - FragmentFullPlayer

Dựa trên kiến trúc MVC đã học, ta có thể xác định được các lớp sau:

- Lớp FragmentDiscover:
  - + Thuộc tính:
    - FragmentDiscoverBinding mFragmentHomeBinding;
    - List<Category> mListCategory: danh sách thể loại bài hát.
    - List<Artist> mListArtist: danh sách ca sĩ.
    - List<Song> mListSong: danh sách bài hát
  - + Phương thức:
    - getListCategoryFromFirebase: Lấy danh sách thể loại từ firebase.
    - displayListCategory: Hiện danh sách các thể loại.
    - getListArtistFromFirebase: Lấy danh sách ca sĩ từ firebase.
    - displayListArtist: Hiện danh sách các thể loại.
    - getListSongFromFirebase: Lấy danh sách bài hát từ firebase.
    - displayListBannerSongs: Hiện danh sách banner bài hát.
- Lớp FragmentSearch:
  - + Thuộc tính:
    - FragmentSearchBinding mFragmentSearchBinding;
    - List<Song> mListSong: danh sách bài hát.
    - SongAdapter mSongAdapter: quản lý danh sách bài hát
  - + Phương thức:
    - initUI: hiển thị danh sách bài hát ở recyclerview.
    - initListner: xử lý sự kiện nhập ký tự vào EditText và click icon search
    - getListSongFromFirebase: lấy danh sách bài hát theo ký tự vừa nhập vào EditText.
- Lớp FragmentFullPlayer:
  - + Thuộc tính:

- FragmentFullPlayerBinding mFragmentPlaySongBinding
- Timer mTimer: dùng để lấy thời gian.
- int mAction: dùng để lưu trữ hành động nhạc được gửi qua Intent.
- final BroadcastReceiver mBroadcastReceiver: dùng để nhận các broadcast trong ứng dụng

+ Phương thức:

- initControl: khởi tạo các điều khiển.
- showInforSong: hiển thị thông tin bài hát.
- handleMusicAction: xử lý các hành động liên quan đến điều khiển nhạc.

### 2.2.3. Thiết kế cơ sở dữ liệu

Bảng 1: Bảng “User”: chứa dữ liệu người dùng

Tên trường	Kiểu dữ liệu	Khóa	Mô tả
id	int	PK	Mã định danh người dùng
email	varchar		Địa chỉ email
name	varchar		Tên người dùng
password	varchar		Mật khẩu người dùng

Bảng 2: Bảng “Artist”: Chứa dữ liệu thông tin nghệ sĩ

Tên trường	Kiểu dữ liệu	Khóa	Mô tả
idArtist	int	PK	Mã định danh nghệ sĩ
nameArtist	varchar		Tên nghệ sĩ
imageArtist	varchar		Ảnh nghệ sĩ

Bảng 3: Bảng “Album”: chứa dữ liệu thông tin Album

Tên trường	Kiểu dữ liệu	Khóa	Mô tả
idAlbum	int	PK	Mã album

nameAlbum	varchar		Tên album
imageAlbum	varchar		Ảnh album

Bảng 4: Bảng “Category”: Chứa dữ liệu thể loại

Tên trường	Kiểu dữ liệu	Khóa	Mô tả
idCategory	int	PK	Mã thể loại
nameCategory	varchar		Tên thể loại
imageCategory	varchar		Ảnh thể loại

Bảng 5: Bảng “Author”: Chứa dữ liệu thông tin tác giả

Tên trường	Kiểu dữ liệu	Khóa	Mô tả
idAuthor	int	PK	Mã định danh tác giả
nameAuthor	varchar		Tên tác giả
imageAuthor	varchar		Ảnh tác giả

Bảng 6: Bảng “Song”: Chứa dữ liệu thông tin bài hát

Tên trường	Kiểu dữ liệu	Khóa	Mô tả
idSong	int	PK	Mã bài hát
nameSong	varchar		Tên bài hát
imageSong	varchar		Ảnh bài hát
linkSong	varchar		Đường dẫn bài hát
idAlbum	int	FK	Mã album
idCategory	int	FK	Mã thể loại
idArtist	int	FK	Mã định danh nghệ sĩ
idAuthor	int	FK	Mã định danh tác giả

## **CHƯƠNG 3. KẾT QUẢ THỰC HIỆN**

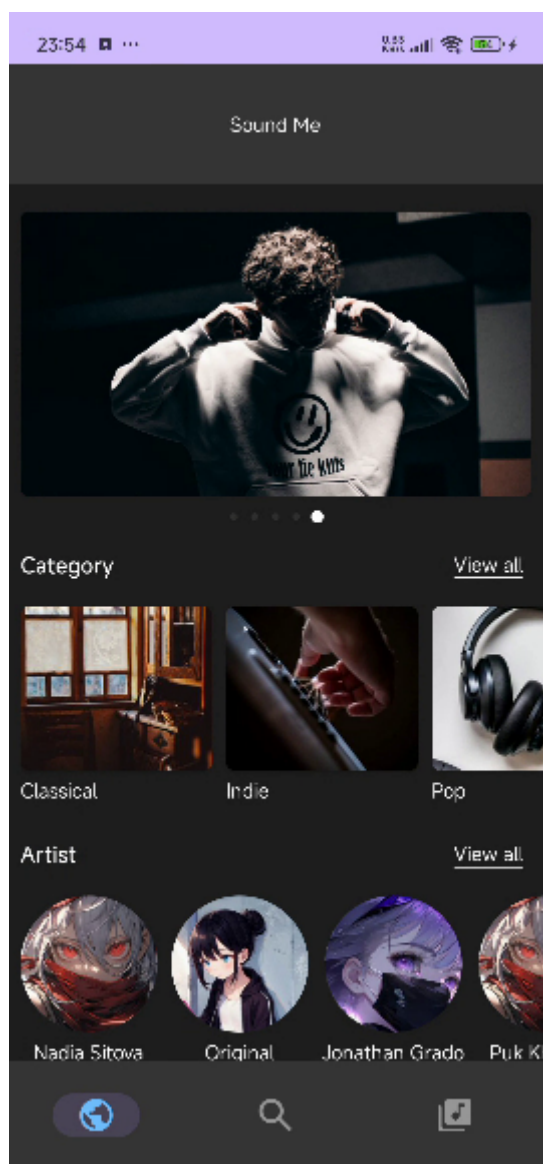
### **3.1. Công nghệ đã sử dụng**

- Ngôn ngữ lập trình: Java
- Công cụ: Android studio
- Thư viện:
  - Gson: dùng chuyển đổi qua lại giữa các đối tượng Java và JSON.
  - Circleindicator: dùng để tạo chỉ báo dưới dạng hình tròn.
  - Glide: dùng để tải và hiển thị ảnh từ URL, URI, tài nguyên nội bộ, và nhiều nguồn khác vào các ImageView một cách hiệu quả
  - Firebase Author: dùng để xác thực người dùng.
  - Firebase Realtime Database: dùng để lưu thông tin bài hát, thể loại, ca sĩ, ...
  - CircleImageView: dùng để hiển thị hình ảnh trong một ImageView dạng hình tròn.

### **3.2. Tiến độ thực hiện**

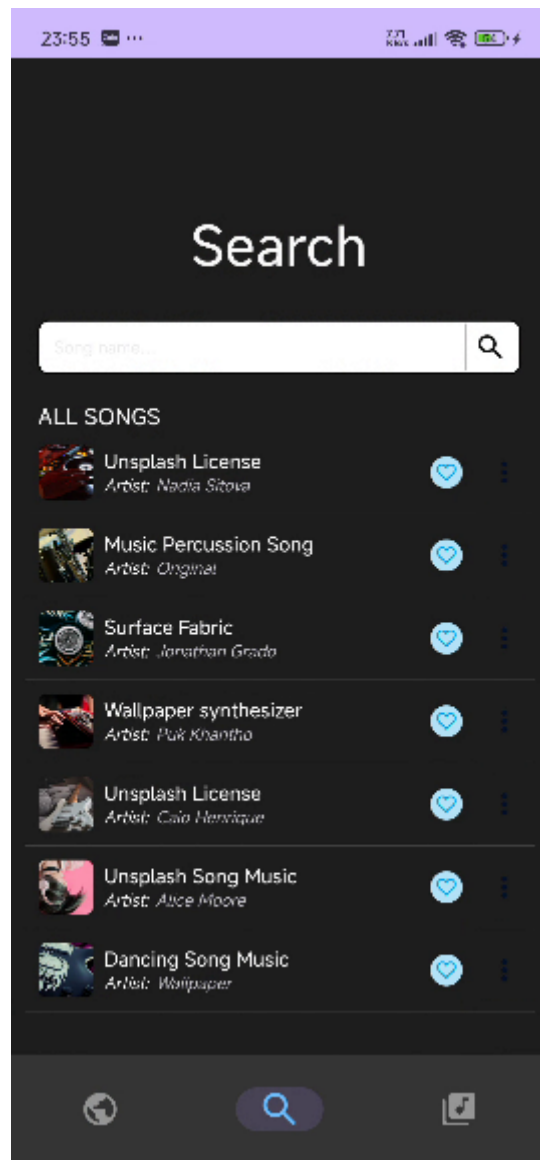
**Link github tới dự án: [https://github.com/linhsalah/CSE441\\_PROJECT](https://github.com/linhsalah/CSE441_PROJECT)**

### 3.3. Hình ảnh sản phẩm

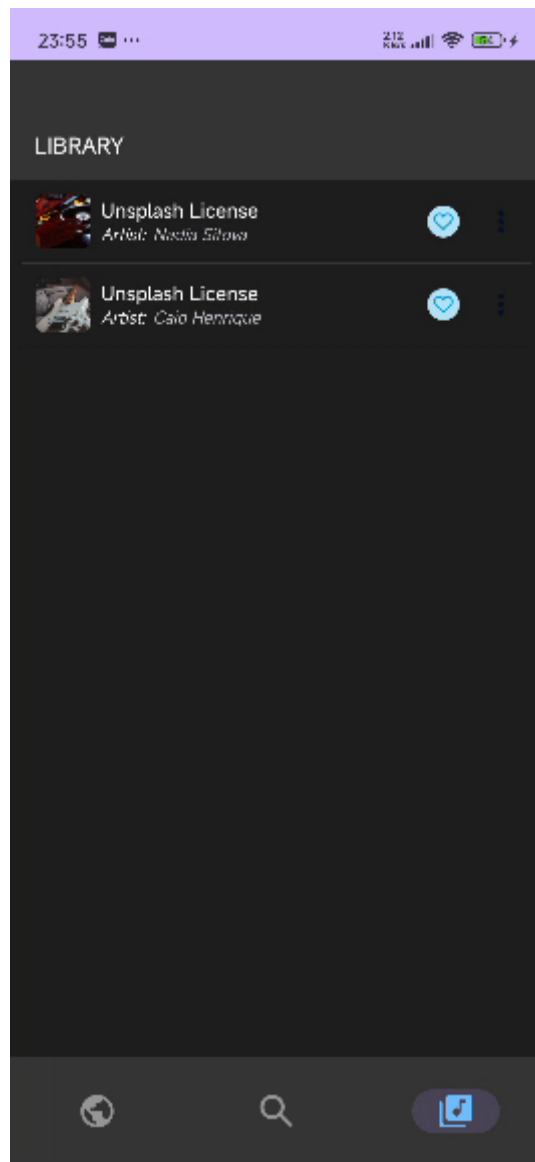


*Giao diện trang chủ*

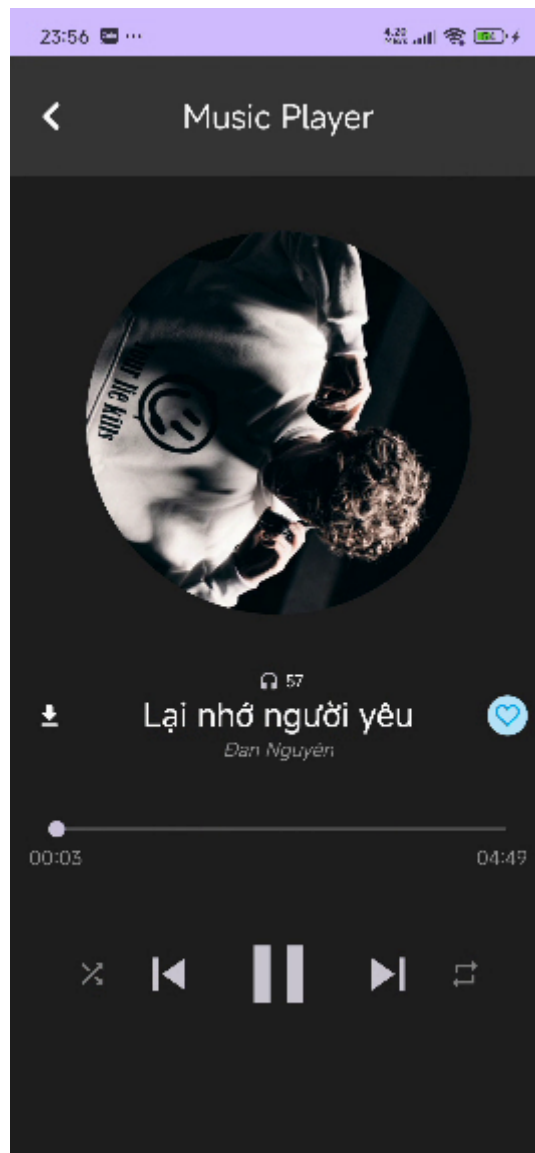




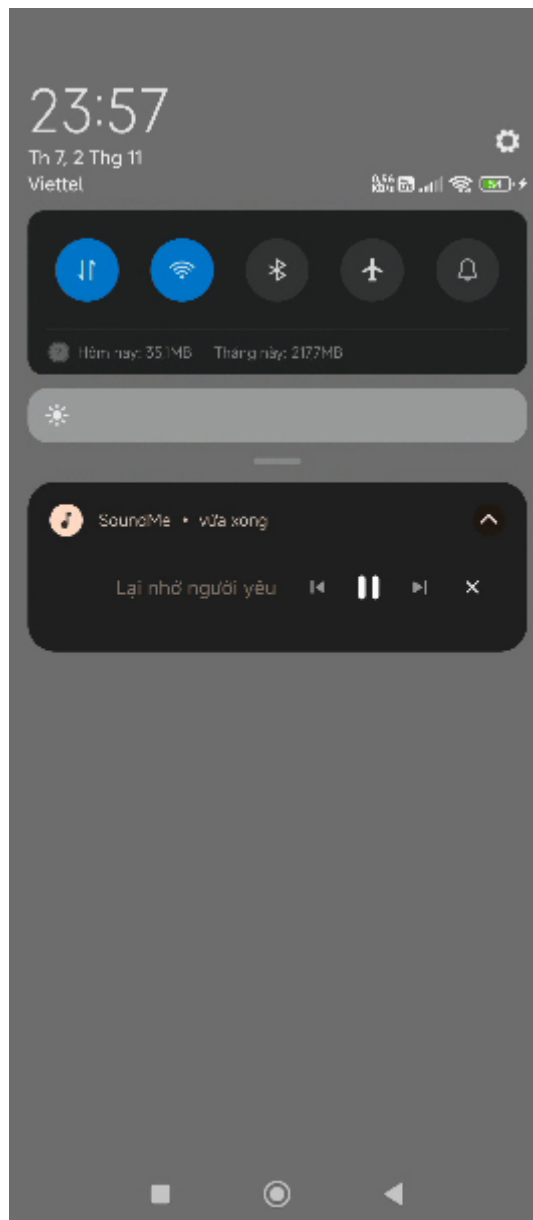
*giao diện trang tìm kiếm*



*giao diện trang bài hát yêu thích*



*giao diện trang phát nhạc*



*giao diện thông báo phát nhạc*