

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**  
**TRƯỜNG ĐẠI HỌC SƯ PHẠM TP.HCM**  
**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO ĐỒ ÁN**  
**PHÁT TRIỂN ỨNG DỤNG TRÊN THIẾT BỊ DI ĐỘNG**  
**ỨNG DỤNG NGHE NHẠC MP3 OFFLINE**

❖ Giảng viên hướng dẫn: **TS. NGUYỄN ĐỖ THÁI NGUYÊN**

❖ Sinh viên thực hiện:

❖ Phan Ngọc Tánh .....46.01.104.163

❖ Phạm Đức Quý .....46.01.104.149

❖ Phạm Nguyễn Hoàng Kha.....46.01.104.075

❖ Lê Minh Triệu .....46.01.104.198

❖ Lê Đức Thi .....46.01.104.173

**TP. Hồ Chí Minh, ngày 10 tháng 5 năm 2022**

# MỤC LỤC

LỜI MỞ ĐẦU.....	1
DANH MỤC CÁC CHỮ VIẾT TẮT .....	2
DANH MỤC HÌNH.....	3
DANH MỤC BẢNG.....	4
CHƯƠNG 1: GIỚI THIỆU ỨNG DỤNG MỤC TIÊU VÀ CÁC TÍNH NĂNG.....	5
1.Giới thiệu ứng dụng .....	5
2. Mục tiêu và các tính năng.....	5
2.1. Tạo nó bằng một ứng dụng có tính năng đơn giản và chạy trơn tru .....	6
2.2. Hỗ trợ điều khiển bằng cử chỉ .....	7
2.3. Hỗ trợ tìm kiếm nhanh, tích hợp chế độ tối (dark mode) .....	8
CHƯƠNG 2: THIẾT KẾ HỆ THỐNG VÀ LƯỢC ĐỒ .....	9
1. Site Map .....	9
2. Use Case Diagram ( sơ đồ Use Case).....	10
3. Activity Diagram.....	13
3.1. Gợi ý (Feature).....	13
3.2. Thư viện (Library).....	14
3.3. Cài đặt (Setting) .....	15
4. System Wireframe (Bố cục hệ thống).....	16
CHƯƠNG 3:NGUỒN CODE VÀ CÁC THÀNH PHẦN TRONG CODE.....	21
1.Quyền ứng dụng .....	21
2. Thư mục Java .....	23
□ Lấy danh sách bài hát gán lên listview .....	23
□ Bắt và xử lý sự kiện .....	24
3. Thư mục Res.....	31
3.1. Gói Drawable .....	31
3.2. Gói Layout.....	32
3.3. Gói Mipmap. ....	38
3.4. Gói Values .....	38
3.5. Animato .....	41
4. Tập Grade Scripts .....	43
TỔNG KẾT .....	44
TÀI LIỆU THAM KHẢO .....	45

## LỜI MỞ ĐẦU

🌈 Đầu tiên, nhóm em xin gửi lời cảm đến khoa **Công nghệ thông tin** nói riêng và hội đồng lãnh đạo **Trường Đại học Sư Phạm TP.HCM** nói chung đã luôn lắng nghe và tạo điều kiện học tập tốt nhất cho cá nhân, nhóm và cũng như toàn thể sinh viên trường.

🌈 Tiếp đến xin dành lời cảm ơn sâu sắc đến **Thầy- ThS. Nguyễn Đỗ Thái Nguyên**, thầy đã truyền đạt những kiến thức thực tế vừa quý báu và vô cùng bổ ích cho nhóm em và các bạn cùng lớp trong suốt thời gian học tập vừa qua. Qua môn học Đồ án chuyên ngành do thầy phụ trách, nhóm em đã có thêm cho mình nhiều kiến thức thực tế quan trọng, tinh thần học tập hiệu quả. Khi chưa học môn này nhóm em còn chưa có những suy nghĩ sâu xa và chưa có định hướng một cách cụ thể, sợ khi tốt nghiệp không có việc làm phải đi làm việc trái ngành trái nghề hay làm công việc đúng ngành nghề nhưng lại không xứng đáng với quá trình, công sức bỏ ra để học đại học với tấm bằng cử nhân.

🌈 Sau khi học xong môn này em và nhóm đã có sự thay đổi trong tư tưởng và nhận thức của mình, tư duy và cách suy nghĩ của nhóm đã tiến triển rõ rệt và càng tăng thêm quyết tâm trở thành một kỹ sư tài năng khiến các nhà tuyển dụng phải tranh giành mình chứ mình không phải đi xin việc nữa. Bộ môn Phát triển ứng dụng di động là môn học thú vị, vô cùng bổ ích và có tính thực tế cao. Đảm bảo cung cấp đủ kiến thức, gắn liền với nhu cầu thực tiễn của sinh viên.

🌈 Tuy, do điều kiện chung của xã hội và dịch bệnh Covid-19 làm cho môn học phải học online, và cũng là lần đầu học online nên cũng có một chút khó khăn nhưng thầy và các bạn học đã cùng nhau đóng góp để môn học thành công tốt đẹp. Mặc dù nhóm đã cố gắng hết sức nhưng chắc chắn bài đồ án không thể tránh khỏi những thiếu sót và nhiều chỗ còn chưa chính xác, kính mong thầy xem xét và góp ý để báo cáo đồ án của nhóm được hoàn thiện hơn.

**DANH MỤC CÁC CHỮ VIẾT TẮT**

<b>TỪ VIẾT TẮT</b>	<b>CỤM TỪ ĐẦY ĐỦ</b>
<b>App</b>	<b>Applications</b>
<b>CPU</b>	<b>Central processing unit</b>
<b>MP3</b>	<b>MPEG 1 Audio Layer 3</b>
<b>UI</b>	<b>User interface</b>
<b>XML</b>	<b>Full extensible markup language</b>
<b>APK</b>	<b>Android Application Package</b>
<b>SDK</b>	<b>Software Development Kit</b>

## DANH MỤC HÌNH

Hình 1 Giới thiệu giao diện.....	6
Hình 2 Giới thiệu giao diện phát nhạc.....	7
Hình 3 Giới thiệu giao diện tìm kiếm và chế độ tối ( dark mode ).....	8
Hình 4 Site Map .....	9
Hình 5 Use Case .....	10
Hình 6 Activity Diagram 1 .....	13
Hình 7 Activity Diagram 2 .....	14
Hình 8 Activity Diagram 3 .....	15
Hình 9 Giao diện logo lần đầu khởi động.....	16
Hình 10 Giao diện truy cập bộ nhớ.....	17
Hình 11 Giao diện thư viện và cách sắp xếp .....	18
Hình 12 Giao diện khi 1 bài hát được phát và các tùy chọn.....	19
Hình 13 Giao diện cài đặt .....	20
Hình 14 Android Manifest.xml .....	22
Hình 15 Gói Drawable.....	31
Hình 16 Gói Layout .....	32
Hình 17 item_feature_playlist.xml.....	33
Hình 18 feature_tab_body.xml.....	34
Hình 19 screen_song_tab.xml.....	35
Hình 20 item_single_playlist.xml .....	36
Hình 21 now_playing.xml .....	36
Hình 22 menu_botton_sheet_layout.xml .....	37
Hình 23 screen_setting_tab.xml .....	37
Hình 24 Gói MipMap .....	38
Hình 25 Strings.xml.....	39
Hình 26 Styles.xml .....	40
Hình 27 Accorodion.....	41
Hình 28 Card_flip .....	42
Hình 29 Grade Scripts.....	43

**DANH MỤC BẢNG**

<b>Bảng 1 Use Case – Thư Viện (ID01) .....</b>	<b>11</b>
<b>Bảng 2 Use Case – Cài Đặt (ID02) .....</b>	<b>12</b>
<b>Bảng 3 Use Case – Gọi Ý (ID03) .....</b>	<b>13</b>

## CHƯƠNG 1: GIỚI THIỆU ỨNG DỤNG MỤC TIÊU VÀ CÁC TÍNH NĂNG

### 1. Giới thiệu ứng dụng

Trong xã hội hiện đại, con người sống với nhịp sống gấp gáp, áp lực thường xuyên hiện hữu trong cuộc sống. Do điện thoại di động được sử dụng rộng rãi nên âm nhạc đã trở thành món ăn tinh thần thiết yếu hàng ngày, bên trong điện thoại di động của mọi người phải có máy nghe nhạc. Một ứng dụng như máy nghe nhạc MP3 được sử dụng để cân bằng giữa căng thẳng và hạnh phúc. Nó đồng hành với mọi người mọi lúc, mọi nơi và bất cứ nơi đâu chẳng hạn như khi mọi người đi xe buýt và tập thể dục.

Ứng dụng nghe nhạc MP3 trên điện thoại di động được thiết kế để cho phép người dùng nghe nhạc một cách thuận tiện và thoải mái hơn mà không bị hạn chế quá nhiều. Hơn nữa, nó có thể phát nhạc đúng cách mà không bị can thiệp bởi quảng cáo và ngoại tuyến.

Do nhiều chủ đầu tư nhận ra rằng các đô thị hiện đại đang sống trong tình trạng căng thẳng, họ đã nắm bắt được cơ hội thương mại, do đó nhiều ứng dụng tương tự đã xuất hiện trên thị trường. Các ứng dụng này có giao diện và tính năng dễ sử dụng giúp trải nghiệm người dùng tốt hơn.

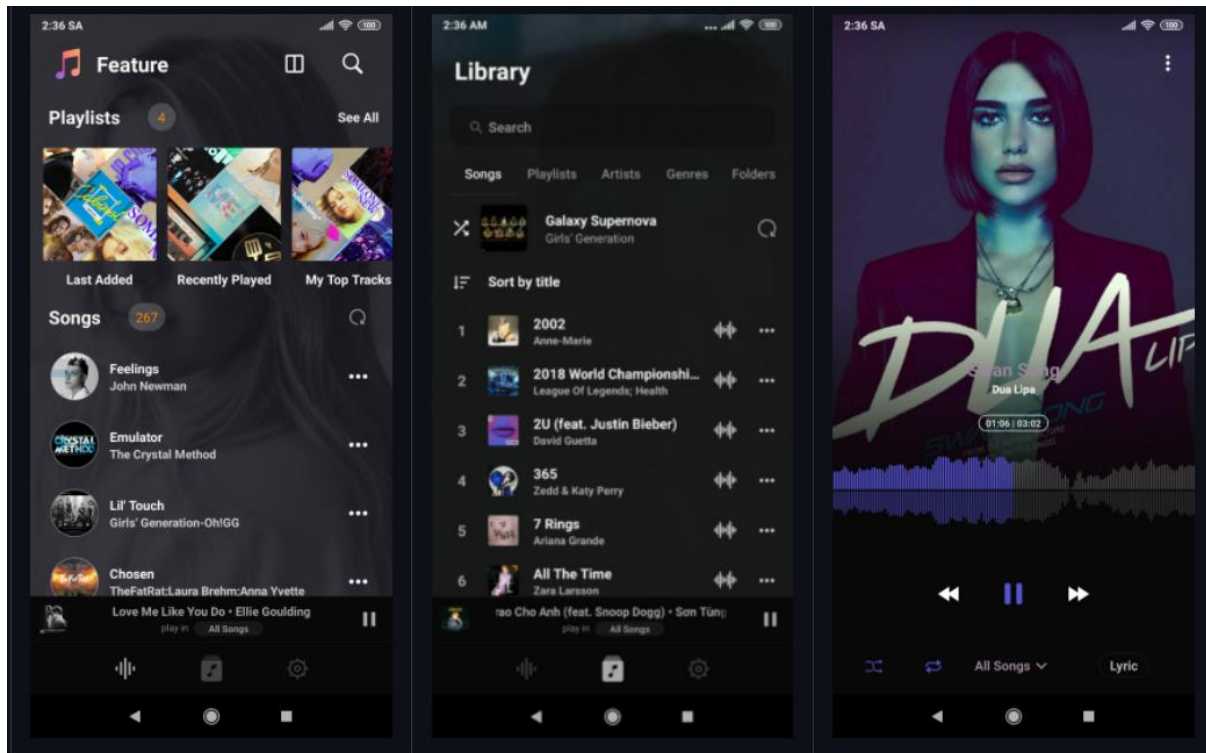
Tuy nhiên, những máy nghe nhạc hiện tại này theo đuổi một cách mù quáng về ngoài sang trọng và các tính năng không lồ, dẫn đến tỷ lệ sử dụng điện thoại di động của người dùng, chẳng hạn như CPU và bộ nhớ cao. Trong khi, đối với hầu hết người dùng bình thường, những loại tính năng khổng lồ và nhiều tính năng này là vô nghĩa. Do đó, dự án này được thiết kế để dành riêng cho máy nghe nhạc MP3 dựa trên nền tảng điện thoại di động Android nhằm tối ưu hóa hiệu suất và đơn giản hóa để đáp ứng nhu cầu của người dùng.

### 2. Mục tiêu và các tính năng

*\*Mục tiêu của đồ án này là đề xuất sự phát triển của android:*

## 2.1. Tạo nó bằng một ứng dụng có tính năng đơn giản và chạy trơn tru

Bằng cách sử dụng máy nghe nhạc mp3 này sẽ làm cho người dùng cảm thấy thoải mái và thư giãn vì nó sẽ chú ý hơn đến các tính năng thường được người dùng sử dụng, loại trừ một số tính năng ít sử dụng chiếm nhiều bộ vi xử lý hệ thống, giúp máy nghe nhạc nhẹ, đơn giản mà còn có các tính năng cơ bản mạnh mẽ.



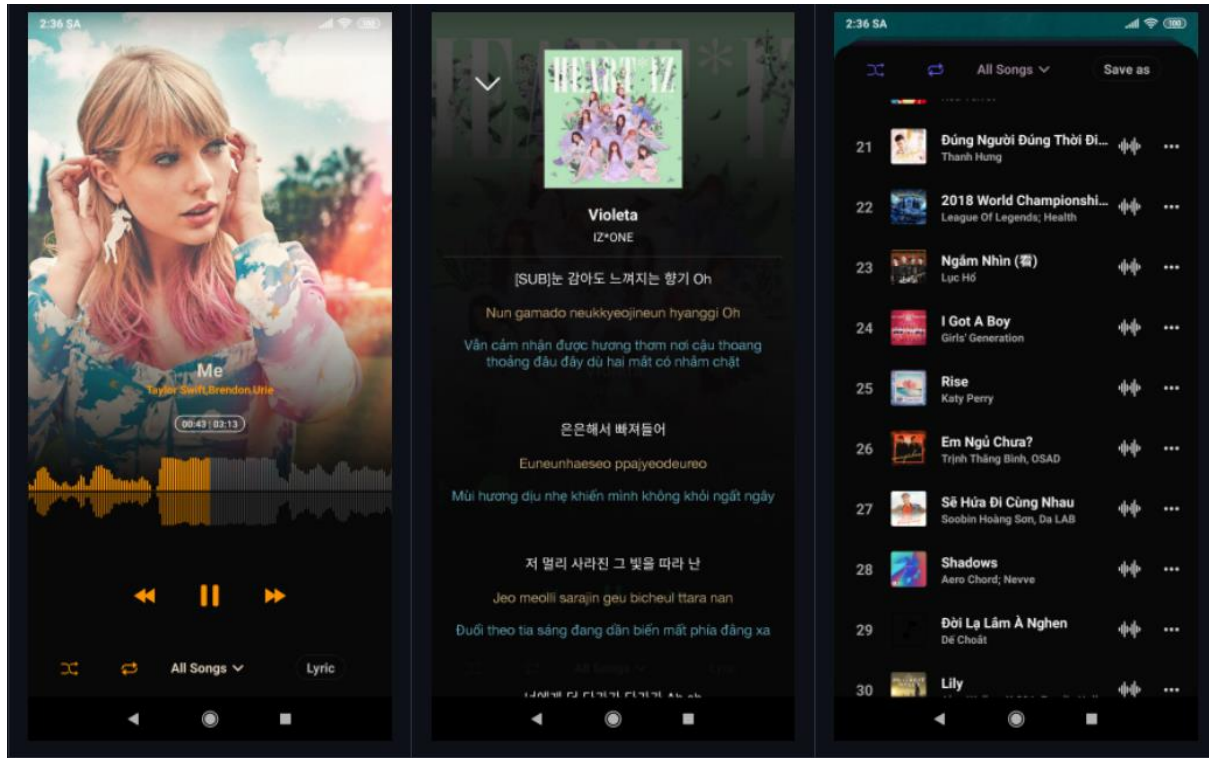
Hình 1 Giới thiệu giao diện



## 2.2. Hỗ trợ điều khiển bằng cử chỉ

Máy nghe nhạc MP3 sẽ bổ sung thêm tính năng kích hoạt bằng cử chỉ để người dùng sử dụng dễ dàng hơn cũng như bớt phụ thuộc vào các nút cảm ứng.

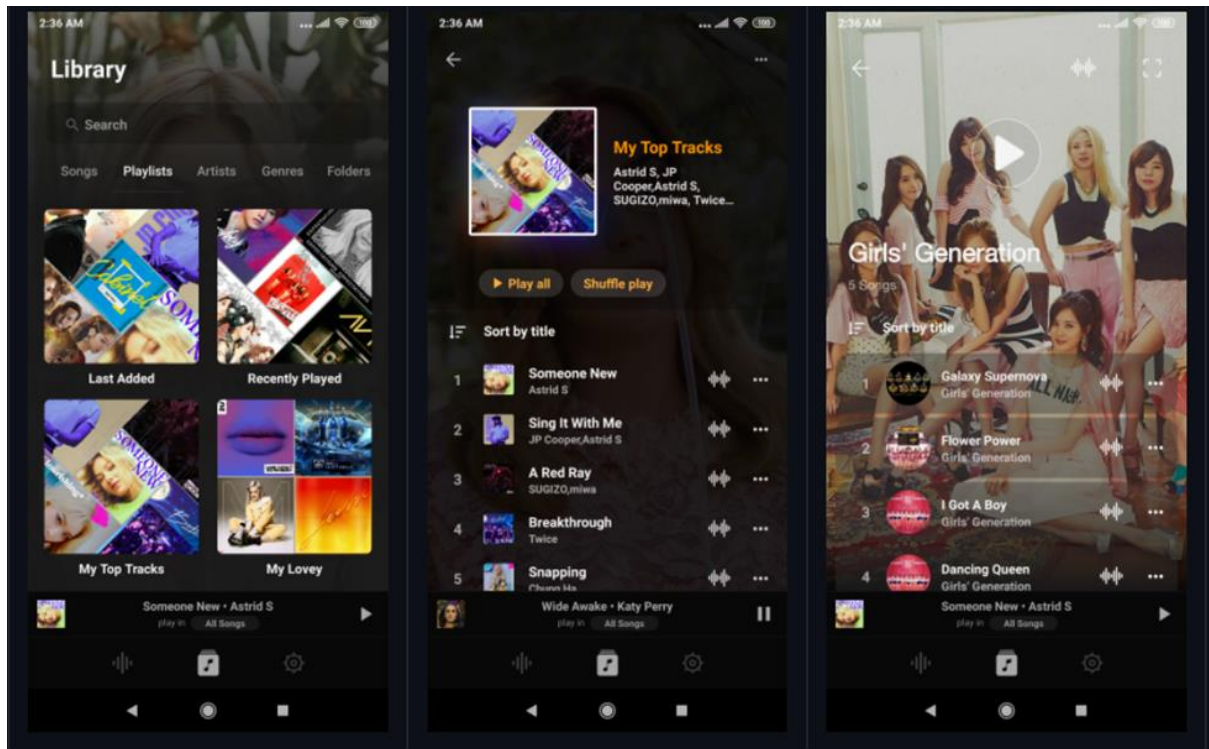
**Ví dụ:** người dùng có thể bỏ qua các bài hát tiếp theo hoặc trước đó bằng cách chỉ cần vuốt sang trái và phải trên bất kỳ vị trí nào của màn hình trong giao diện phát.



Hình 2 Giới thiệu giao diện phát nhạc

### 2.3. Hỗ trợ tìm kiếm nhanh, tích hợp chế độ tối (dark mode)

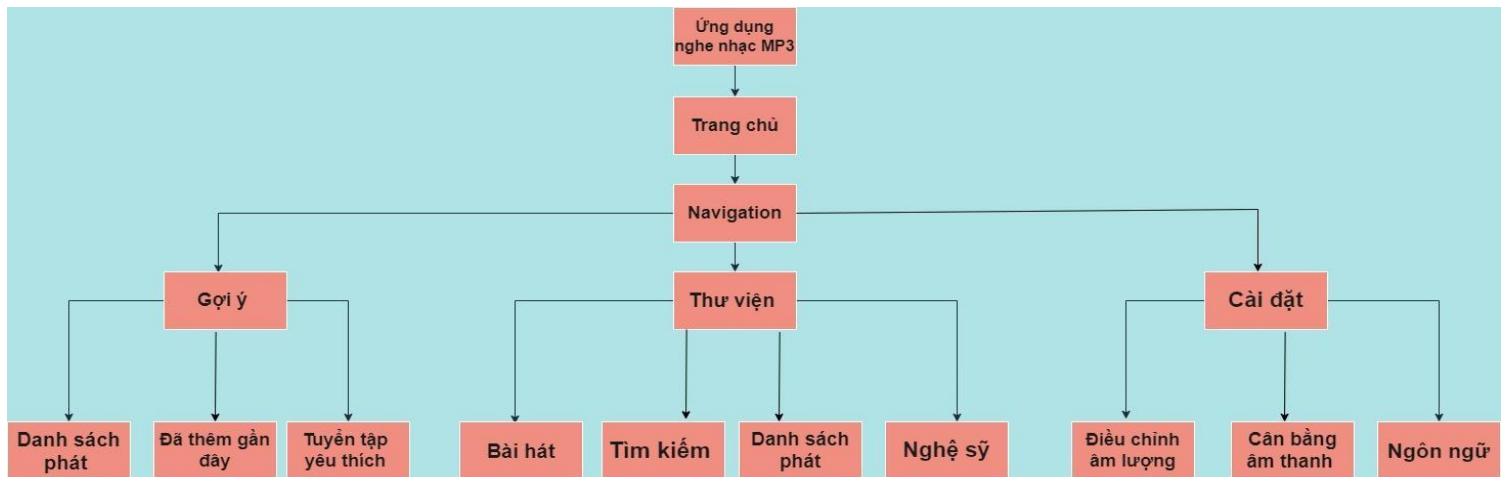
Việc thiếu thanh tìm kiếm trong danh sách nhạc là không thể chấp nhận được. Do đó, trình phát nhạc mp3 sẽ sử dụng thanh tìm kiếm cũng như cuộn nhanh bằng bảng chữ cái ở bên phải màn hình, cho phép người dùng lọc nhanh hàng trăm bài hát để tìm ra bài mà người dùng muốn phát.



Hình 3 Giới thiệu giao diện tìm kiếm và chế độ tối ( dark mode )

## CHƯƠNG 2: THIẾT KẾ HỆ THỐNG VÀ LƯỢC ĐỒ

### 1. Site Map



**Hình 4 Site Map**

- Sơ đồ trang trong dự án này là một phương pháp để hiển thị thông tin trên bố cục của tổng thể ứng dụng nghe nhạc MP3.

- Theo hình 1.1 ở trên, nó cho thấy rằng dự án bao gồm bên trái thanh điều hướng (navigation) gồm có “Gợi ý”, trung tâm thanh điều hướng là “Thư viện” nơi chứa bài hát, và cuối cùng bên phải thanh điều hướng là “Cài đặt” giúp cho người dùng chỉnh sửa nếu cần thiết.

- Trong trang "Thư viện", người dùng có thể chọn một bài hát từ danh sách phát để phát và sau đó chuyển sang giao diện "Đang phát nhạc". Trên trang này, người dùng được phép chuyển sang chế độ điều khiển để phát bài hát. Gồm các chức năng tiêu biểu sau:

▲ Chức năng Shuffle: Khi người dùng chọn nút Shuffle, bài hát ngẫu nhiên trong danh sách sẽ được phát khi kết thúc bài hát hiện tại.

▲ Chức năng Previous: Khi người dùng chọn nút previous, bài hát phía trước của bài hát hiện tại trong danh sách bài hát sẽ được phát.

▲ Chức năng Play: Khi người dùng chọn nút play, một bài hát trong danh sách bài hát sẽ được phát.

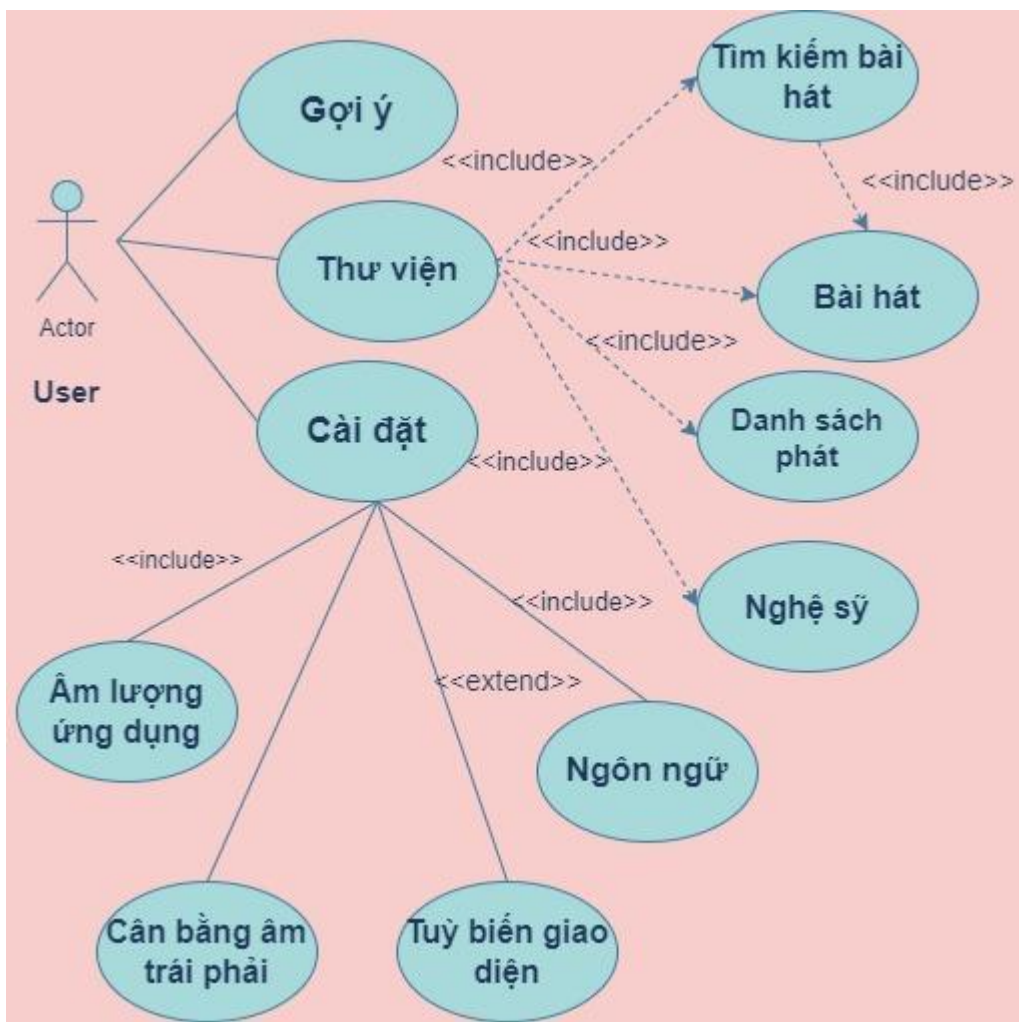
▲ Chức năng Pause: Khi người dùng chọn nút pause, bài hát sẽ tạm dừng. – Chức năng Stop: Khi người dùng chọn nút stop, bài hát sẽ dừng lại.

▲ Chức năng Next: Khi người dùng chọn nút next, bài hát phía sau của bài hát hiện tại trong danh sách bài hát sẽ được phát.

▲ Chức năng Repeat: Khi người dùng chọn nút Repeat, các bài hát trong danh sách bài hát sẽ được phát tuần tự cho đến bài hát cuối cùng trong danh sách

- Tiếp theo, trong phần “Gợi ý” hay còn gọi là Feature, người dùng có thể xem những bài hát được thêm gần đây, bài hát được nghe gần đây nhất, cũng như những album yêu thích. Ngoài ra, người dùng có thể thay đổi bố cục thành màu tối trong “Chế độ ban đêm”.
- Người dùng muốn chấm dứt ứng dụng có thể chỉ cần nhấp vào “Thoát”

## 2. Use Case Diagram ( sơ đồ Use Case)



**Hình 5 Use Case**

**\*Đặc tả Use Case:**

<b>Tên Use Case:</b> Thư viện <b>ID (01)</b>
<b>Actor:</b> Người dùng (user)
<b>Mô tả:</b> người dùng có thể nghe nhạc trên ứng dụng này.
<b>Hướng dẫn ( mẹo):</b> 1.Người dùng chọn bài hát bất kì hoặc bài hát nào đó trong danh sách phát. 2.Người dùng nhấp vào nút phát để bắt đầu.
<b>Điều kiện tiên quyết:</b> 1.Người dùng nên nhập bài hát từ thiết bị di động ( thiết bị nghe nhạc) của họ và ứng dụng nghe nhạc 2.Người dùng nên thêm bài hát vào danh sách nghe nhạc.
<b>Dòng sự kiện chính:</b> 1.Người dùng chọn 1 bài hát bất kì hoặc bài hát trong danh sách phát. 2.Bài hát được phát cho đến khi kết thúc.
<b>Dòng sự kiện phụ:</b> ứng dụng sẽ hiển thị thông báo không có bài hát nào được phát.

**Bảng 1 Use Case – Thư Viện (ID01)**

<b>Tên Use Case:</b> Cài đặt <b>ID 02)</b>
<b>Actor:</b> Người dùng (user)
<b>Mô tả:</b> người dùng có thể thay đổi ngôn ngữ, giao diện, âm lượng.
<b>Hướng dẫn ( mẹo):</b> 1.Người dùng chọn vào cài đặt ( hình bánh răng).

2. Người dùng nhấp vào các cài đặt mà lựa chọn.
<b>Điều kiện tiên quyết:</b> 1. Người dùng bắt buộc phải mở ứng dụng. 2. Người dùng phải nhập vô phần cài đặt và lựa chọn các quyền.
<b>Dòng sự kiện chính:</b> 1. Người dùng chọn bất kì 1 cài đặt 2. Nhấp vô để hoàn tất áp dụng.
<b>Dòng sự kiện phụ:</b> ứng dụng sẽ hiển thị trạng thái, giao diện, các hành động sau khi thay đổi.

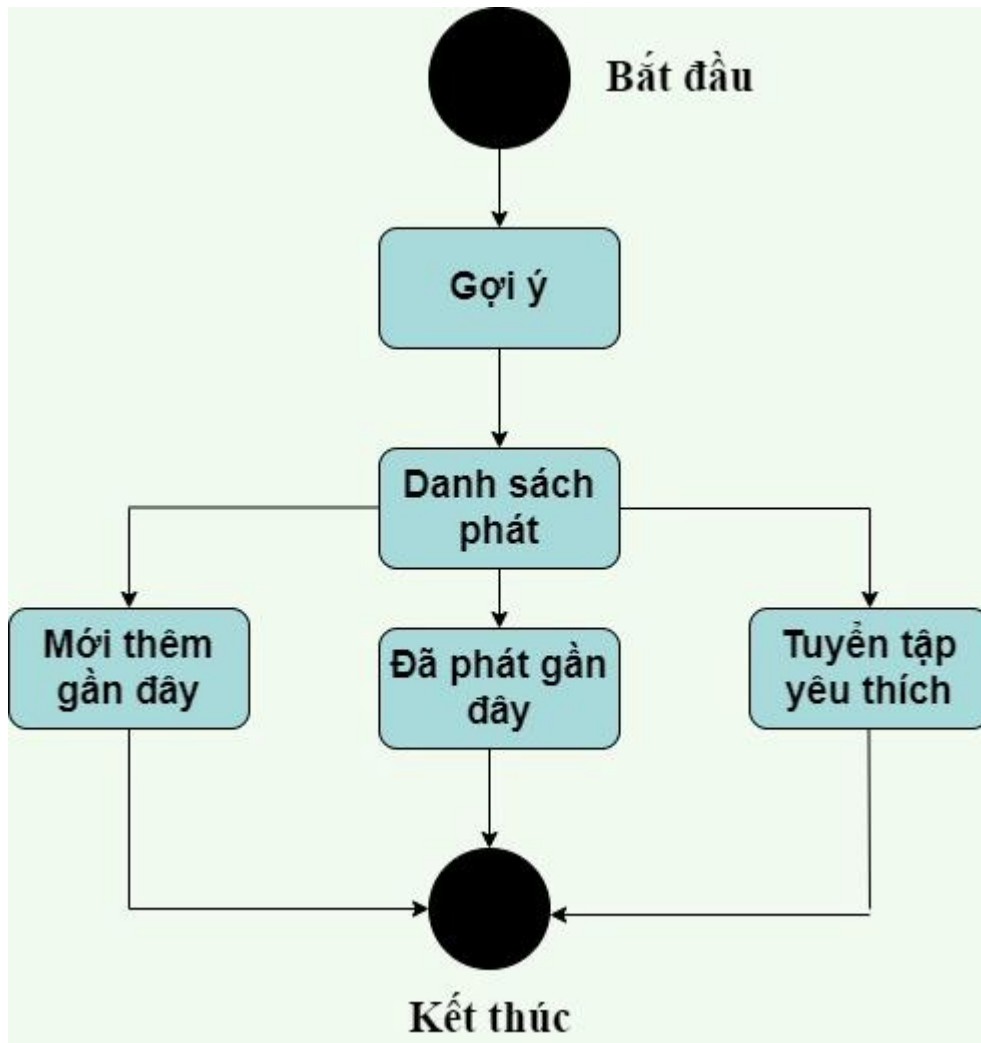
Bảng 2 Use Case – Cài Đặt (ID02)

<b>Tên Use Case:</b> Gợi ý (ID03)
<b>Actor:</b> Người dùng (user)
<b>Mô tả:</b> người dùng có thể xem thanh tiến trình, các bài hát gần đây, các bài hát mới được thêm vào
<b>Điều kiện tiên quyết:</b> 1. Người dùng nên nhập bài hát từ thiết bị di động ( thiết bị nghe nhạc) của họ và ứng dụng nghe nhạc 2. Người dùng đã nghe hoặc đã tạo danh sách phát
<b>Dòng sự kiện chính:</b> 1. Người dùng sẽ chọn các bài hát gần đây đã nghe 2. Bài hát được phát cho đến khi kết thúc.
<b>Dòng sự kiện phụ:</b> ứng dụng sẽ hiển thị thông báo không có bài hát nào đã nghe gần đây, hoặc không có danh sách phát nào được thêm mới

Bảng 3 Use Case – Gọi Ý (ID03)

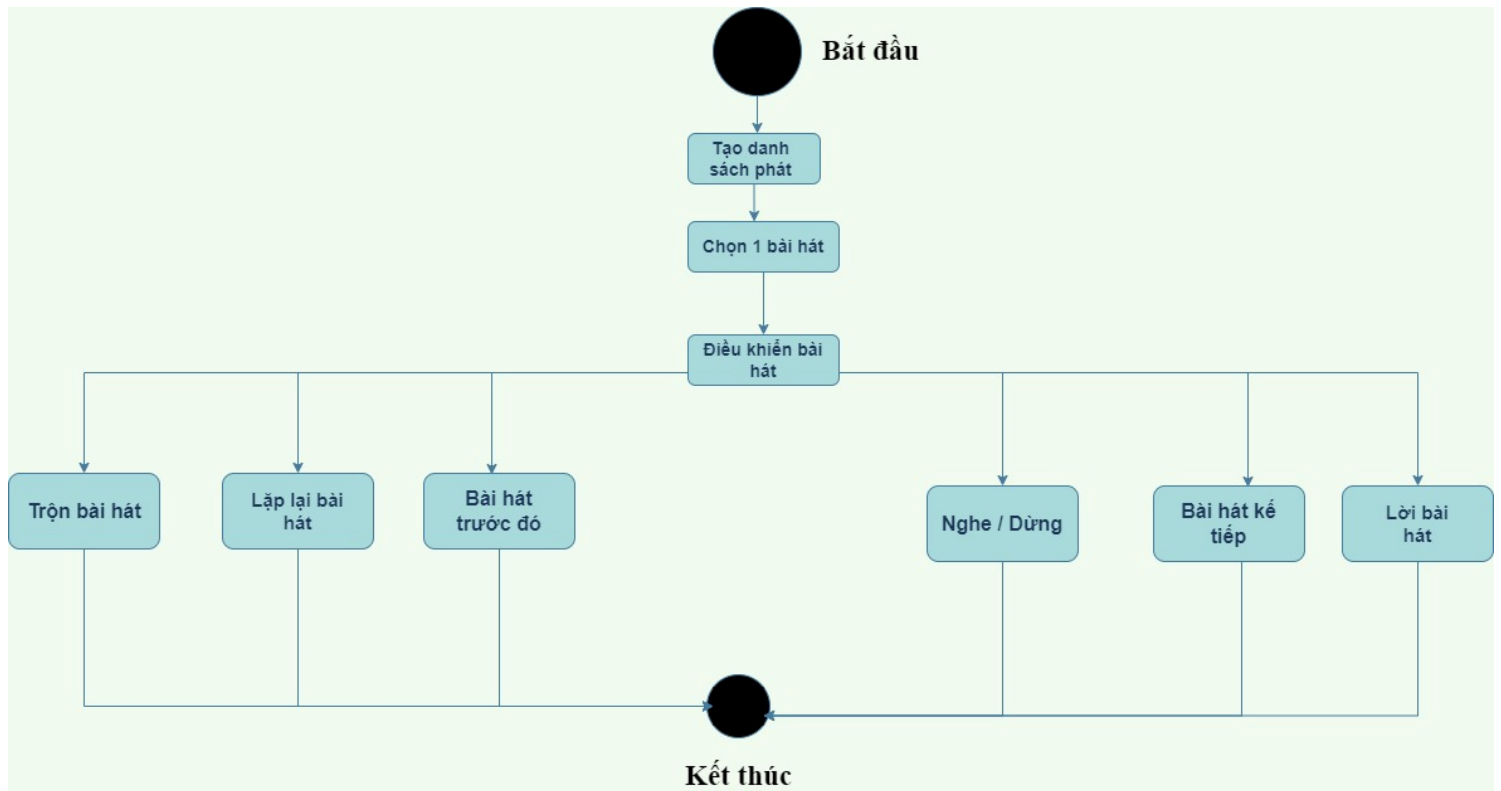
3. Activity Diagram

## 3.1. Gọi ý (Feature)



Hình 6 Activity Diagram 1

### 3.2. Thư viện (Library)



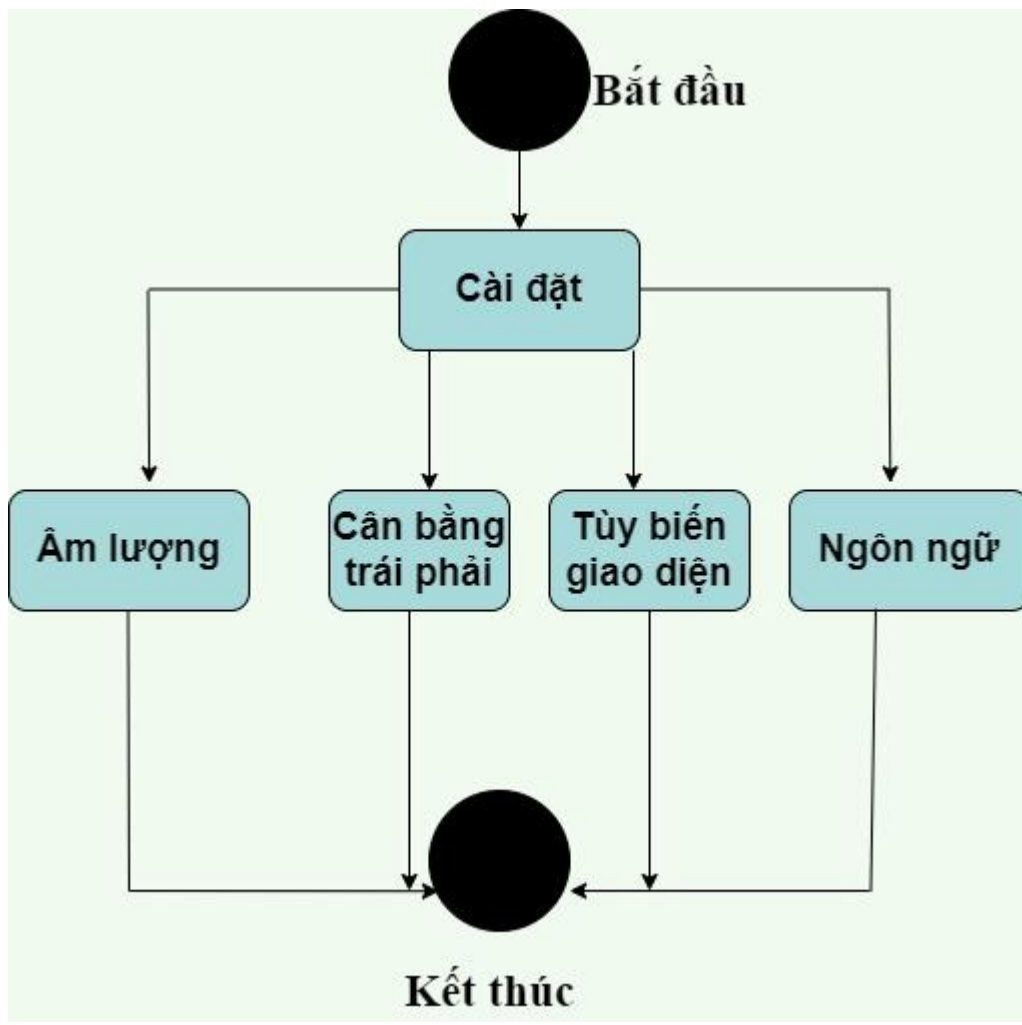
**Hình 7 Activity Diagram 2**

- Người dùng khởi động ứng dụng và sau đó nó sẽ chuyển đến trang chủ là "Thư viện trang nhạc". Ứng dụng sẽ đọc các bài hát cục bộ của thiết bị và tạo danh sách phát.

- Sau đó, người dùng chọn bài hát và phát, trang sẽ chuyển sang chơi. Giao diện âm nhạc, nơi người dùng được phép điều khiển âm nhạc. Chế độ chơi là được phép kiểm soát các lệnh phát nhạc là phát ngẫu nhiên, chu kỳ đơn và phát lại bình thường. Hơn nữa, người dùng cũng có thể chuyển sang bài hát tiếp theo, quay lại bài hát trước đó bài hát và phát hoặc tạm dừng nhạc hiện tại để điều khiển nhạc. Cuối cùng, người dùng có thể bật chế độ trình điều khiển để phát bài hát.

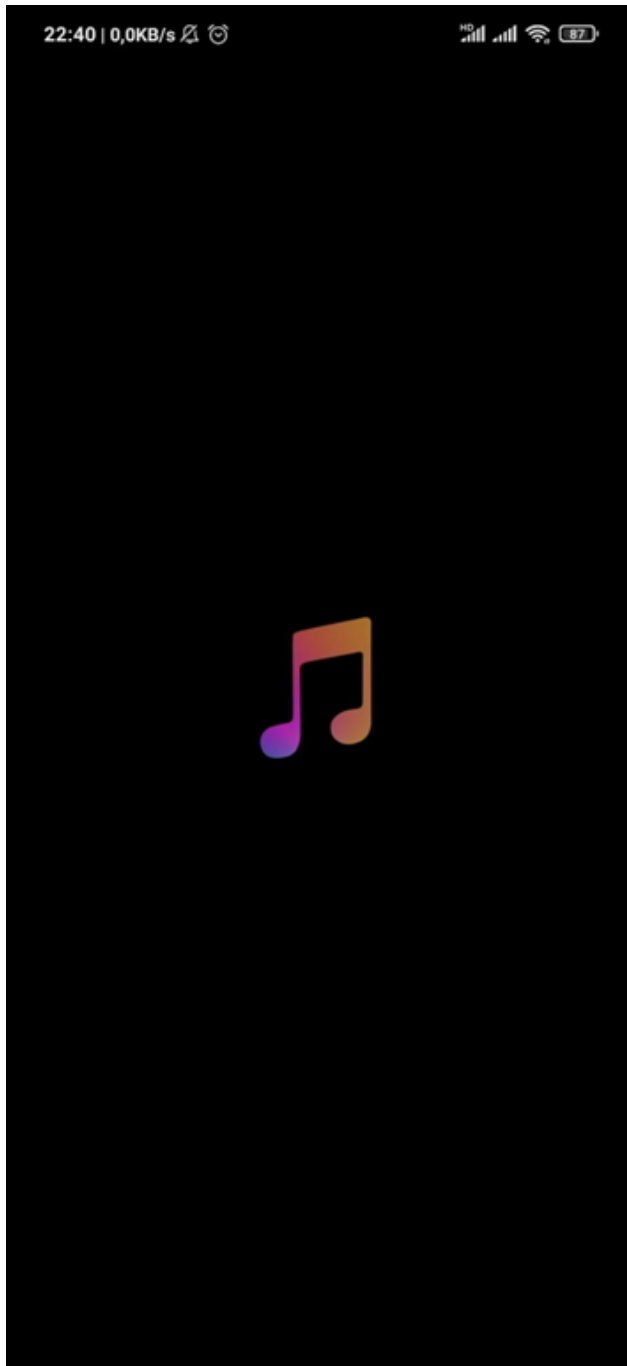


### 3.3. Cài đặt (Setting)



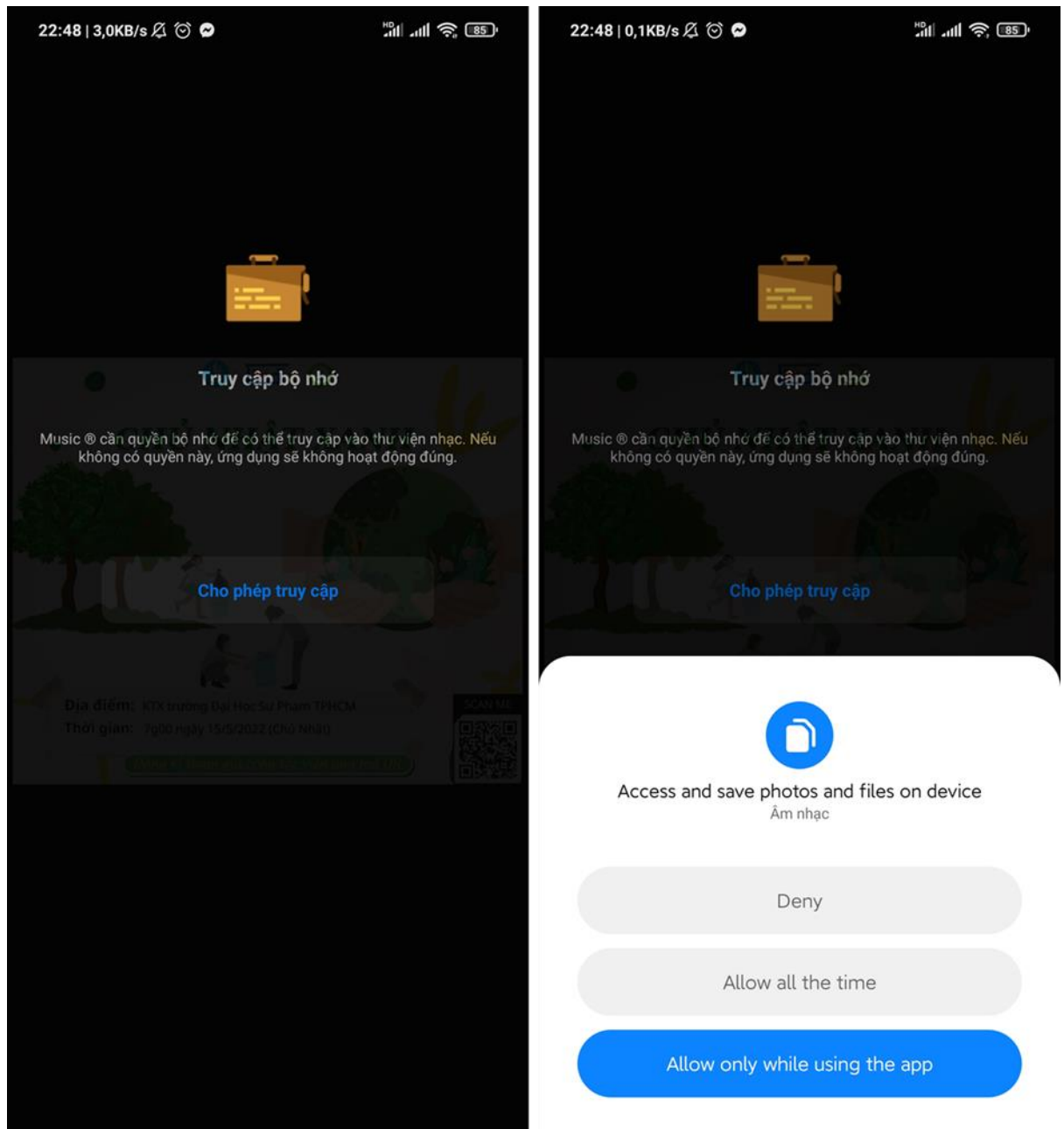
Hình 8 Activity Diagram 3

#### 4. System Wireframe (Bố cục hệ thống)



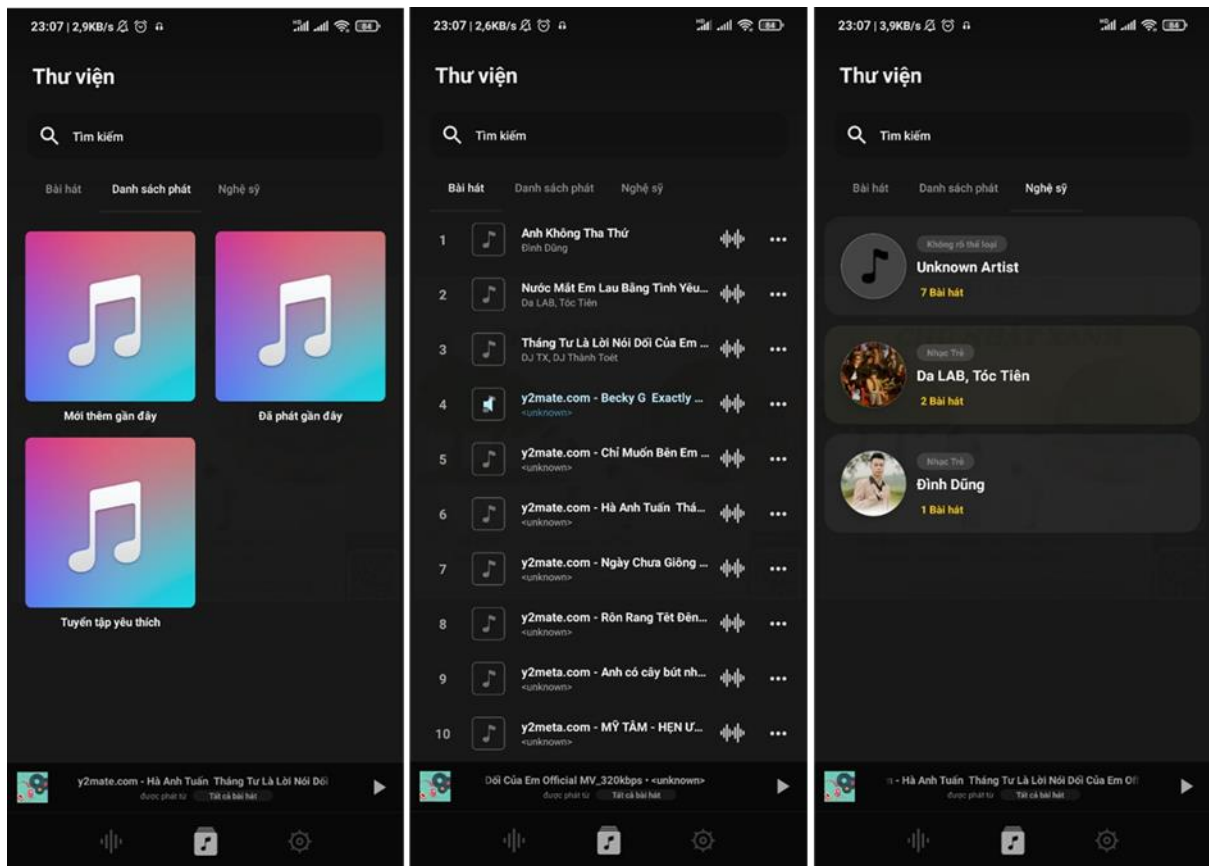
- **Hình 9** mô tả: Khi người dùng lần đầu tiên khởi động ứng dụng, sẽ hiển thị biểu tượng logo của ứng dụng

**Hình 9** Giao diện logo lần đầu khởi động



**Hình 10** Giao diện truy cập bộ nhớ

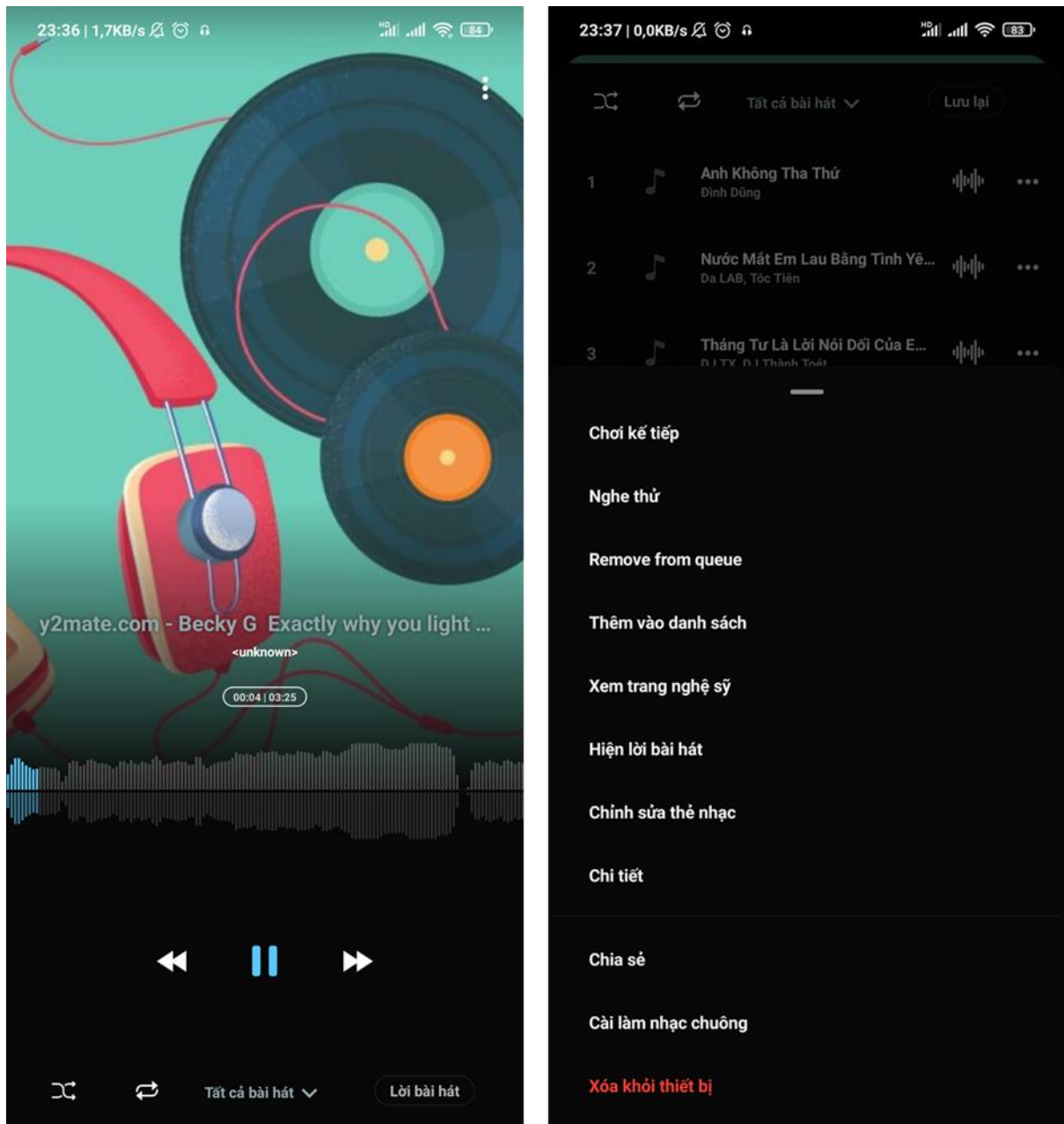
- **Hình 10** : Tiếp theo đó ứng dụng sẽ hỏi quyền truy cập bộ nhớ ( chỉ áp dụng lần đầu tiên ). Và lưu ý: bạn cho phép ứng dụng thì mới chạy được.



Hình 11 Giao diện thư viện và cách sắp xếp

Trong 3 hình trên là đầu tiên là công cụ tìm kiếm bài hát. Thư viện được chia thành 3 phần:

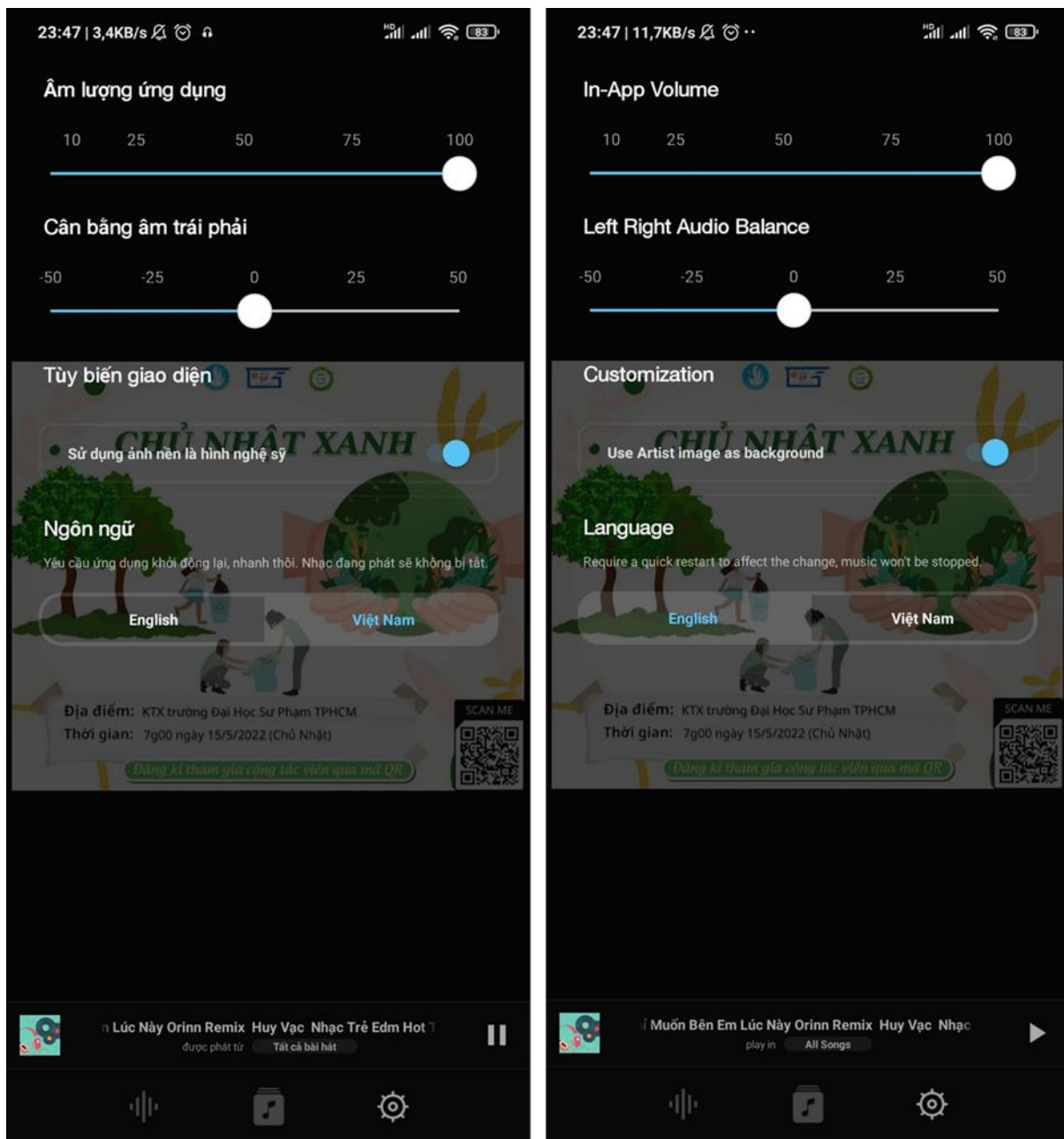
- + Bài hát: thứ tự và tên các bài hát
- + Danh sách phát: cho biết danh sách phát do người dùng tạo
- + Nghệ sĩ: hiển thị tên nghệ sĩ, tác giả kèm theo ảnh bài hát



**Hình 12 Giao diện khi 1 bài hát được phát và các tùy chọn**

- Khi bài hát bắt đầu phát, đĩa ở giữa màn hình sẽ bắt đầu xoay và nút ở dưới cùng chính giữa sẽ được cập nhật thành nút tạm dừng. Nút tạm dừng sẽ được cập nhật thành nút phát nếu bài hát ngừng phát. Người dùng có thể kiểm soát tiến trình của bài hát bằng cách kéo thanh tiến trình.

- Kéo thanh tiến trình ở dưới ta được có thể sử dụng các tính năng như nghe bài kế tiếp, nghe thử, thêm, xóa, sửa bài hát, và đặc biệt có thể cài làm nhạc chuông điện thoại.



Hình 13 Giao diện cài đặt

Trong 2 hình trên có thể tùy chỉnh âm lượng ứng dụng cũng như cân bằng âm trái và phải. Đặc biệt có thể chuyển đổi ngôn ngữ linh hoạt từ Tiếng Việt sang Tiếng Anh và ngược lại.

## CHƯƠNG 3: NGUỒN CODE VÀ CÁC THÀNH PHẦN TRONG CODE

### 1. Quyền ứng dụng

#### AndroidManifest.xml:

Nó đặt tên gói Java cho ứng dụng. Tên gói đóng vai trò như một mã nhận diện duy nhất cho ứng dụng.

❖ Nó mô tả các thành phần của ứng dụng - hoạt động, dịch vụ, hàm nhận quảng bá, và trình cung cấp nội dung mà ứng dụng được soạn bởi. Nó đặt tên các lớp triển khai từng thành phần và công bố các khả năng của chúng (ví dụ, những tin nhắn Intent mà chúng có thể xử lý). Những khai báo này cho phép hệ thống AndroidManifest 30 Android biết các thành phần là gì và chúng có thể được khởi chạy trong những điều kiện nào.

❖ Nó xác định những tiến trình nào sẽ lưu trữ các thành phần ứng dụng.

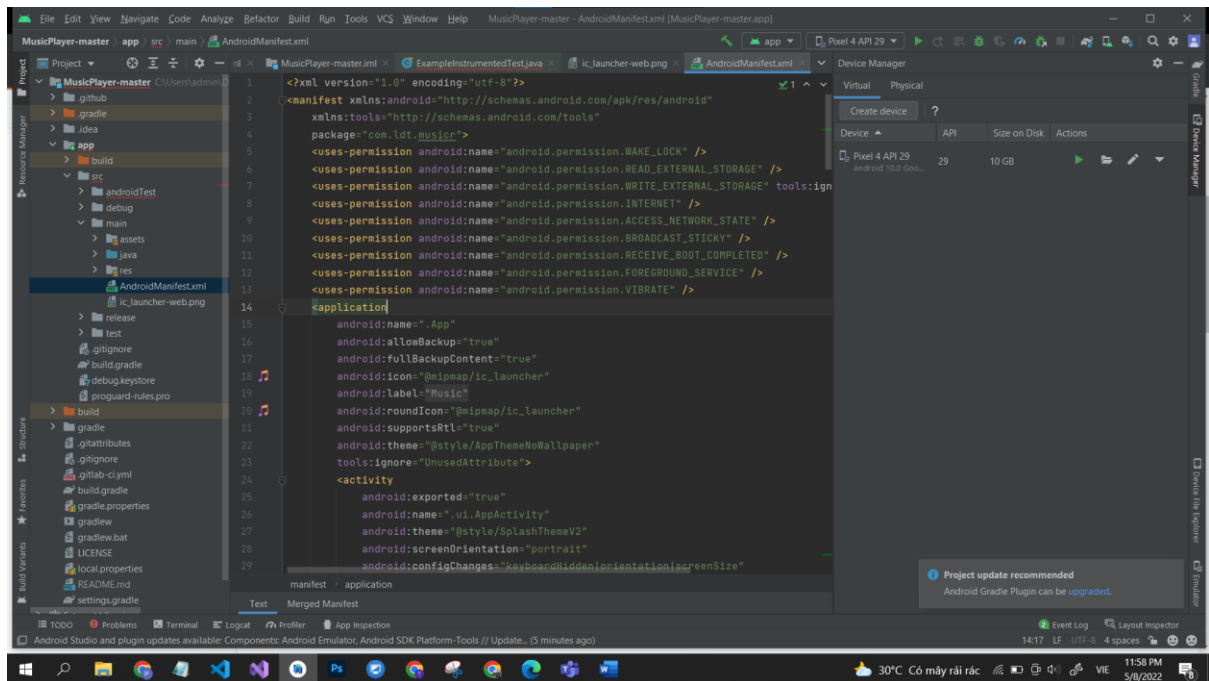
❖ Nó khai báo các quyền mà ứng dụng phải có để truy cập các phần được bảo vệ của API và tương tác với các ứng dụng khác.

❖ Nó cũng khai báo các quyền mà ứng dụng khác phải có để tương tác với các thành phần của ứng dụng.

❖ Nó liệt kê các lớp Instrumentation cung cấp tính năng tạo hồ sơ và các thông tin khác khi ứng dụng đang chạy. Những khai báo này chỉ xuất hiện trong bản kê khai khi ứng dụng đang được phát triển và thử nghiệm; chúng bị loại bỏ trước khi ứng dụng được công bố.

❖ Nó khai báo mức tối thiểu của API Android mà ứng dụng yêu cầu.

❖ Nó liệt kê các thư viện mà ứng dụng phải được liên kết với.



Hình 14 Android Manifest.xml

Khi khởi động, ứng dụng cần được cấp những quyền sau:

```
<uses-permission android:name="android.permission.WAKE_LOCK" />
```

- ❖ **WAKE-LOCK:** là một cơ chế để giữ cho thiết bị luôn bật. Nó được sử dụng chẳng hạn khi bạn cần làm mọi việc ngay cả khi thiết bị có vẻ như đang ở chế độ ngủ, chẳng hạn như tải tệp xuống từ internet. Wakelocks không bao giờ được sử dụng trừ khi bạn thực sự cần chúng. Nguyên nhân là do chúng tốn nhiều pin hơn và nếu bạn gặp lỗi mà không phát hành chúng khi cần thiết, ứng dụng của bạn sẽ tiếp tục tốn pin của thiết bị. Thậm chí còn có các ứng dụng để phát hiện các ứng dụng có vấn đề như vậy (như "trình phát hiện wakelock").

```
<uses-permission  
android:name="android.permission.READ_EXTERNAL_STORAGE" />
```

```
<uses-permission  
android:name="android.permission.WRITE_EXTERNAL_STORAGE"  
tools:ignore="ScopedStorage" />
```

- ❖ **READ EXTERNAL STORAGE:** là quyền đọc lưu trữ của bộ nhớ.
- ❖ **WRITE EXTERNAL STORAGE:** là quyền được viết trên bộ nhớ  
=> Nhờ có 2 quyền này mà ứng dụng có thể nghe nhạc (đọc-ghi dữ liệu trên điện thoại của bạn)



```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission
android:name="android.permission.ACCESS_NETWORK_STATE" />
```

- ❖ INTERNET và ACCESS NETWORK STATE: là quyền được truy cập mạng Internet và trạng thái Internet.

```
<uses-permission
android:name="android.permission.BROADCAST_STICKY"/>
<uses-permission
android:name="android.permission.RECEIVE_BOOT_COMPLETED" />
<uses-permission
android:name="android.permission.FOREGROUND_SERVICE"/>
<uses-permission android:name="android.permission.VIBRATE" />
```

- ❖ đây là khai báo khi sử dụng Broadcast Reciver, cái này là để lắng nghe các sự kiện thay đổi của hệ thống như khi bắt wifi, khi sạc pin.

## 2. Thư mục Java

Đây chính là nơi chứa các gói của dự án, có thể tạo các gói ở đây và bên trong là các class

### \* Lấy danh sách bài hát gắn lên listview

- ✓ Mô tả Truy cập vào thư mục “Mymusic” trong bộ nhớ trong. Sau đó lọc các file nhạc có đuôi “.mp3
- ✓ Mã nguồn

```
String path =
Environment.getExternalStorageDirectory().getAbsolutePath()
+ "/Mymusic";
File file = new File(path);
File[] files = file.listFiles();

for (int i = 0; i < files.length; i++) {
    String s = files[i].getName();
    if(s.endsWith(".mp3")) {
        MediaMetadataRetriever mmr = new
MediaMetadataRetriever();
        String pathbh = files[i].getAbsolutePath();
        mmr.setDataSource(pathbh);
        String artist =
mmr.extractMetadata(MediaMetadataRetriever.METADATA_KEY_ARTIST);
        String title =
mmr.extractMetadata(MediaMetadataRetriever.METADATA_KEY_TITLE);
        byte[] anhcasi = mmr.getEmbeddedPicture();
        danh sach bai hat.add(new Song(title, artist ,pathbh, anhcasi));
    }
}
adapter.notifyDataSetChanged();
}
```

### \* **Bắt và xử lý sự kiện**

#### ✓ Sự kiện chạm vào listview

- Mô tả: Khi chọn bài hát nào trên listview thì bài hát đó sẽ được phát.
- Mã nguồn

```
lvDSBaiHat.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> adapterView, View view,
int i, long l) {
        index = i;
        if(status == PLAY || status == PAUSE){
            mediaPlayer.stop();
            status = STOP;
        }
        runSong();
        mediaPlayer.start();
        status = PLAY;
        ivPlay.setImageResource(R.drawable.ic_pause_black_24dp);
    }
});
```

## ✓ Nút Play

- Mô tả: Kiểm tra nếu có bài hát đang chạy (status = PLAY) thì cho tạm dừng bài hát, đổi icon pause thành icon play[7]. Còn nếu chưa có bài hát nào đang được chọn, thì sẽ phát bài hát đầu tiên trong danh sách bài hát, sau đó đổi icon play thành icon pause.
- Mã nguồn

```
ivPlay.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if(status == STOP) {
            runSong();
            mediaPlayer.start();
            status = PLAY;

            Toast.makeText(getApplicationContext(), "Play...", Toast.LENGTH_SHORT)
                .show();
        } else if(status == PLAY){
            mediaPlayer.pause();
            status = PAUSE;

            ivPlay.setImageResource(R.drawable.ic_pause_black_24dp);

            Toast.makeText(getApplicationContext(), "Pause...", Toast.LENGTH_SHORT)
                .show();
        } else if(status == PAUSE){
            mediaPlayer.start();
            status = PLAY;

            ivPlay.setImageResource(R.drawable.ic_play_arrow_black_24dp);

            Toast.makeText(getApplicationContext(), "Play...", Toast.LENGTH_SHORT)
                .show();
        }
    }
});
```

## ✓ Nút Next

- Mô tả  
Kiểm tra nếu có bài hát đang phát thì sẽ cho dừng bài hát đó lại. Sau đó kiểm tra chỉ số bài hát hiện tại, nếu nhỏ hơn tổng số bài hát trừ đi một (`danhsachbaihat.size()-1`) thì sẽ phát bài hát có chỉ số hiện tại cộng một (`index+=1`). Còn ngược lại sẽ phát bài hát đầu tiên.
- Mã nguồn

```
ivNext.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if(status == PLAY){
            mediaPlayer.stop();
            status = STOP;
        }
        if (index < danhsachbaihat.size()-1){
            index += 1;
        }else{
            index = 0;
        }
        runSong();
        mediaPlayer.start();
        status = PLAY;
    }
});
```

## ✓ Nút Previous

- Mô tả  
Kiểm tra nếu có bài hát đang phát thì sẽ cho dừng bài hát đó lại. Sau đó kiểm tra chỉ số bài hát hiện tại, nếu lớn hơn không (`index>0`) thì sẽ phát bài hát có chỉ số hiện tại trừ một (`index-=1`). Ngược lại sẽ phát bài hát cuối danh sách
- Mã nguồn

```

ivPrevious.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if (index > 0){
            index -=1 ;
        }else{
            index = danhSachbaihat.size()-1;
        }
        if(status == PLAY){
            mediaPlayer.stop();
            status = STOP;
        }
        runSong();
        mediaPlayer.start();
        status = PLAY;
    }
});

```

### ✓ Nút Stop

- Mô tả  
Nếu có bài hát đang phát thì dừng bài hát lại, sau đó đổi icon nút play thành icon play, đặt tên bài hát, tên ca sỹ, thời gian thực và thời gian tổng của bài hát về mặc định như ban đầu.
- Mã nguồn

```

ivStop.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if(status == PLAY || status ==PAUSE){
            mediaPlayer.stop();
            mediaPlayer.release();
            status = STOP;
        }
        ivPlay.setImageResource(R.drawable.ic_play_arrow_black_24dp);
        txtSong.setText("Tên bài hát");
        txtSinger.setText("Tên ca sĩ");
        stopSong();
    }
});

```

✓ Nút Shuffle

- Mô tả

Kiểm tra trạng thái Shuffle đang được bật thì sẽ chạy một bài hát ngẫu nhiên trong danh sách bài hát khi kết thúc bài hát hiện tại

- Mã nguồn

```
ivShuffle.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        isShuffle = !isShuffle;
        final Handler handler = new Handler();
        handler.postDelayed(new Runnable() {
            @Override
            public void run() {
                if (isShuffle) {
                    mediaPlayer.setOnCompletionListener(new
MediaPlayer.OnCompletionListener() {
                        @Override
                        public void onCompletion(MediaPlayer
mPlayer) {
                            if (status == PLAY) {
                                mediaPlayer.stop();
                                Random rand = new Random();
                                index =
rand.nextInt((danh sach bai hat.size() - 1) - 0 + 1) + 0;
                                runSong();
                                mediaPlayer.start();
                                status = PLAY;
                            }
                        }
                    });
                    handler.postDelayed(this, 500);
                }
            }, 500);
        }
    });
});
```

✓ Nút Repeat

- Mô tả

Kiểm tra nếu trạng thái Repeat đang bật thì sẽ cho chạy bài hát tiếp theo cho đến hết danh sách, còn nếu trạng thái không bật thì sẽ dừng lại khi kết thúc bài hát hiện tại.

- Mã nguồn

```
ivRepeat.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        isRepeat = !isRepeat;
        if (isRepeat) {

ivRepeat.setImageResource(R.drawable.ic_repeat_one_black_24dp);
        } else {
ivRepeat.setImageResource(R.drawable.ic_repeat_black_24dp);
        }
        final Handler handler = new Handler();
        handler.postDelayed(new Runnable() {
            @Override
            public void run() {
                if (isRepeat) {
                    mediaPlayer.setOnCompletionListener(new
MediaPlayer.OnCompletionListener() {
                        @Override
                        public void onCompletion(MediaPlayer
mPlayer) {
                            if (status == PLAY) {
                                mediaPlayer.stop();
                                index = index + 1;
                                if (index == danh sach bai hat.size())
{
                                    index = 0;
                                }
                                runSong();
                                mediaPlayer.start();
                                status = PLAY;
ivPlay.setImageResource(R.drawable.ic_play_arrow_black_24dp);
                                }
                            }
                        });
                    handler.postDelayed(this, 500);
                }
            }, 500);
        }
    });
});
```

### ✓ Thanh Seekbar

- Mô tả  
Diễn tả thời gian thực của bài hát Khi kéo thanh seekbar đến vị trí nào thì tiến trình của bài hát sẽ ở tiếp tục ở vị trí đó.
- Mã nguồn

```
seekBarTime.setOnSeekBarChangeListener(new
SeekBar.OnSeekBarChangeListener() {
    @Override
    public void onProgressChanged(SearchBar seekBar, int i, boolean
b) {
    }
    @Override
    public void onStartTrackingTouch(SearchBar seekBar) {
    }
    @Override
    public void onStopTrackingTouch(SearchBar seekBar) {
        mediaPlayer.seek(seekBarTime.getProgress());
    }
});
```

### ✓ Cập nhập tổng thời gian của bài hát

- Mô tả  
Định dạng thời gian tổng của bài hát để có thể hiển thị phút và giây.
- Mã nguồn

```
private void setTimeTotal(){
    SimpleDateFormat DinhDang = new SimpleDateFormat("mm:ss");
    txtTimeTotal.setText(DinhDang.format(mediaPlayer.getTimeTotal()));
    seekBarTime.setMax(mediaPlayer.getTimeTotal());
}
```

### ✓ Cập nhập thời gian thực của bài hát

- Mô tả  
Định dạng thời gian thực của bài hát để có thể hiển thị phút và giây
- Mã nguồn

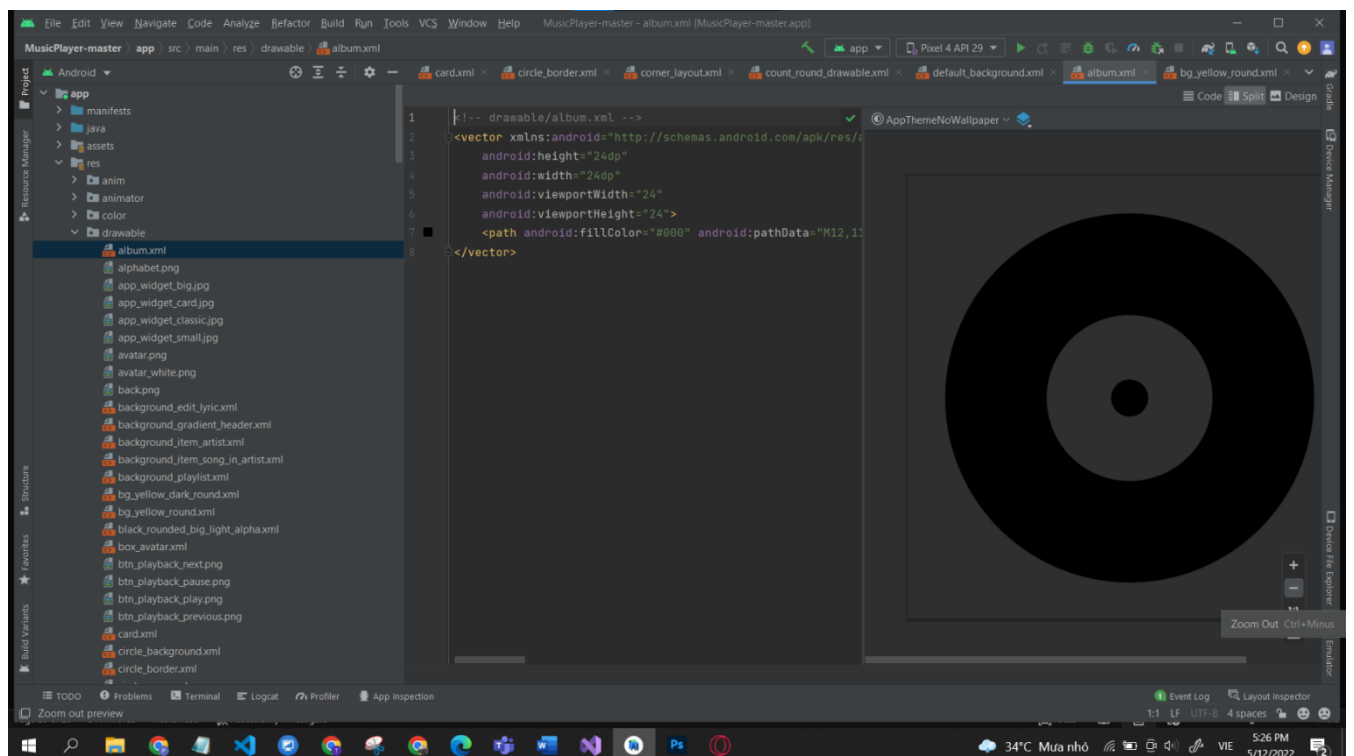


```
private void updateTimeSong(){
    final Handler handler = new Handler();
    handler.postDelayed(new Runnable() {
        @Override
        public void run() {
            SimpleDateFormat dinhDang = new
SimpleDateFormat("mm:ss");
txtTimeCurrent.setText(dinhDang.format(mediaPlayer.getTimeCurrent()
));
            seekBarTime.setProgress(mediaPlayer.getTimeCurrent());
            handler.postDelayed(this, 500);
        }
    }, 100);
}
```


### 3. Thư mục Res

#### 3.1. Gói Drawable:

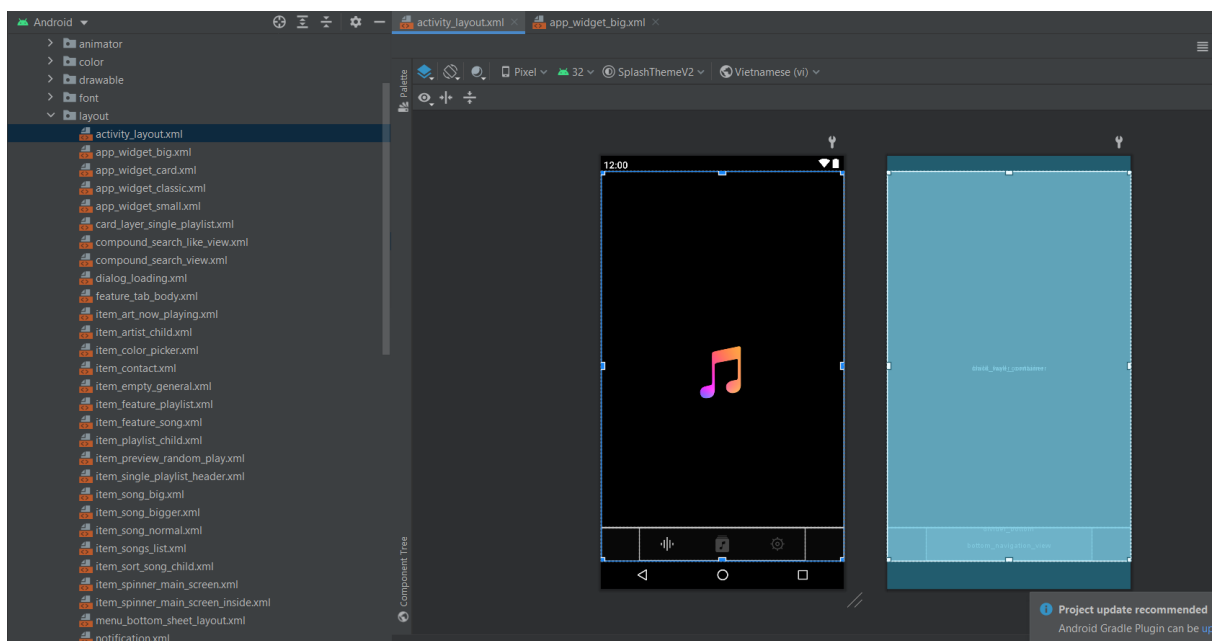
Đây chính là thư mục chứa các file hình ảnh, config xml... trong dự án android. Ví dụ như muốn ứng dụng sử dụng một hình ảnh nào đó là background thì ảnh đó sẽ bỏ vào thư mục này. Hoặc muốn điều chỉnh một nút button khi click vào màu xanh còn khi không click vào màu trắng thì sẽ config trong file xml và lưu vào trong này.



Hình 15 Gói Drawable

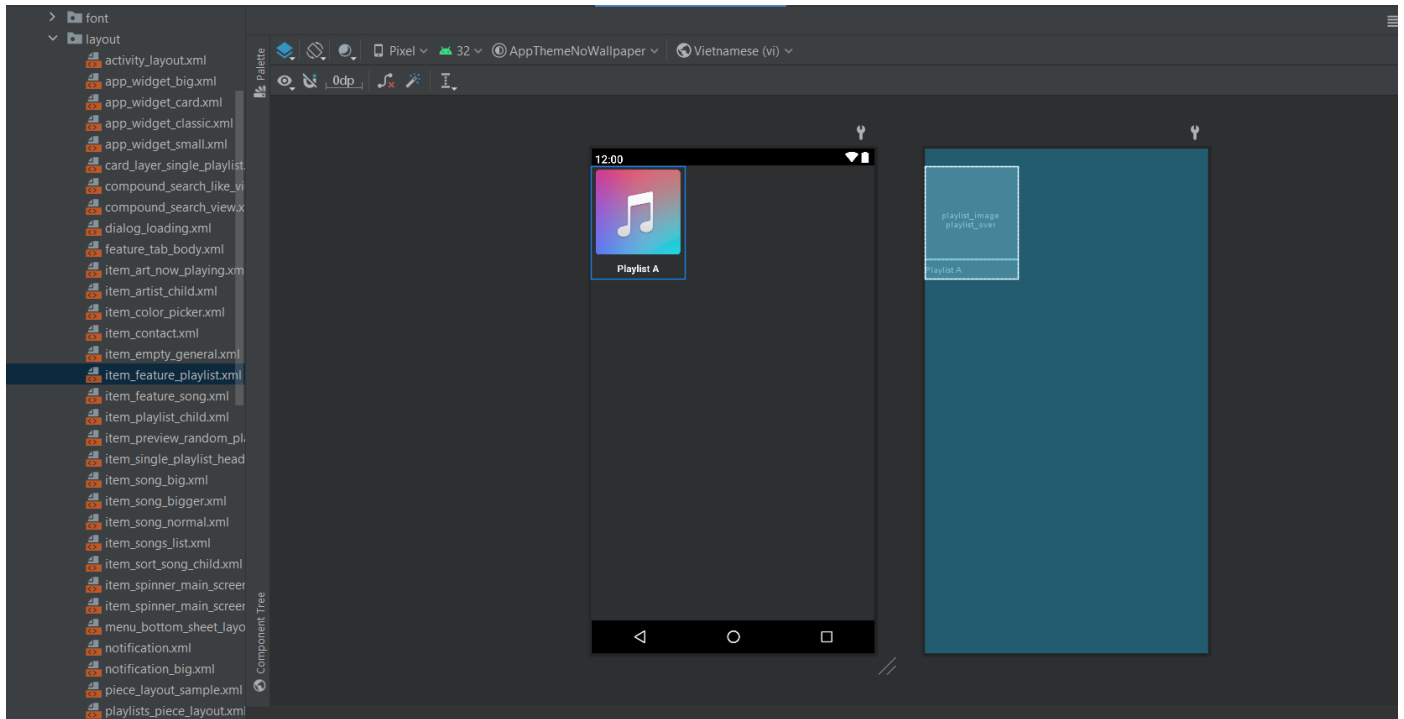
 **3.2. Gói Layout:** Đây chính là thư mục lưu các file xml về giao diện của các màn hình ứng dụng của. Ở trên phần số một có các package lưu các class, các class này sẽ kết nối với các file xml trong thư mục layout nào để tạo nên một màn hình có giao diện cho người dùng thao tác. Giao diện gồm:

- Listview: Hiển thị danh sách bài hát
- Textview: Hiển thị tên bài hát, tên ca sỹ, tổng thời gian bài hát và thời gian thực của bài hát.
- Imageview: Hiển thị các nút sự kiện (play, next, previous, stop, repeat, shuffle).
- Seekbar: Thanh thiết lập tiến trình của bài hát
- Tạo một file item\_list.xml trong thư mục layout. File xml này để thể hiện tiêu đề của bài hát. Dùng một textview để hiển thị tên bài hát, một textview để hiển thị tên ca sỹ, một imageview để hiển thị ảnh của ca sỹ.
- Trong file **activity\_layout.xml** thể hiện bố cục, giao diện khi hoạt động của người dùng bấm vào ứng dụng.



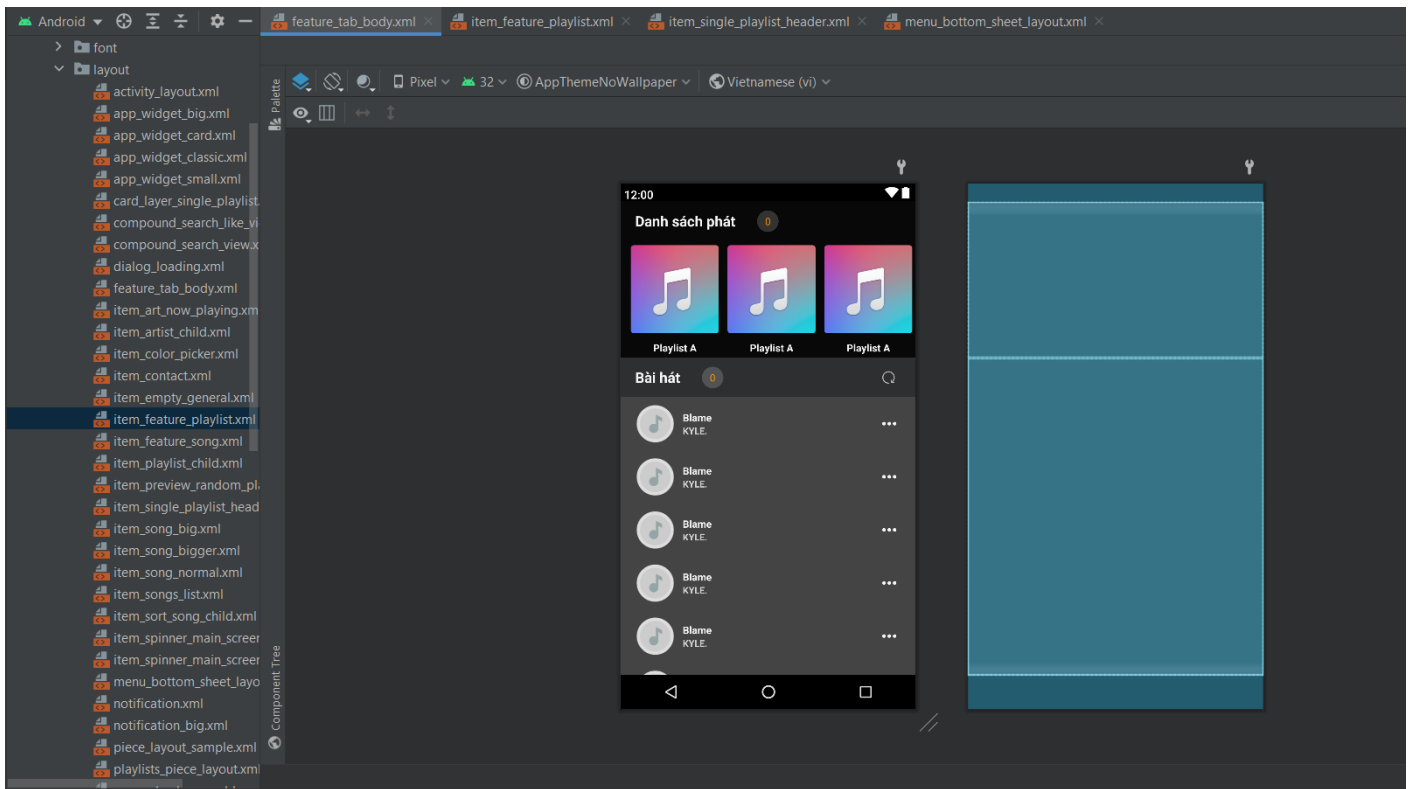
**Hình 16 Gói Layout**

- Tiếp đến file **item\_feature\_playlist.xml** thể hiện các mục danh sách phát



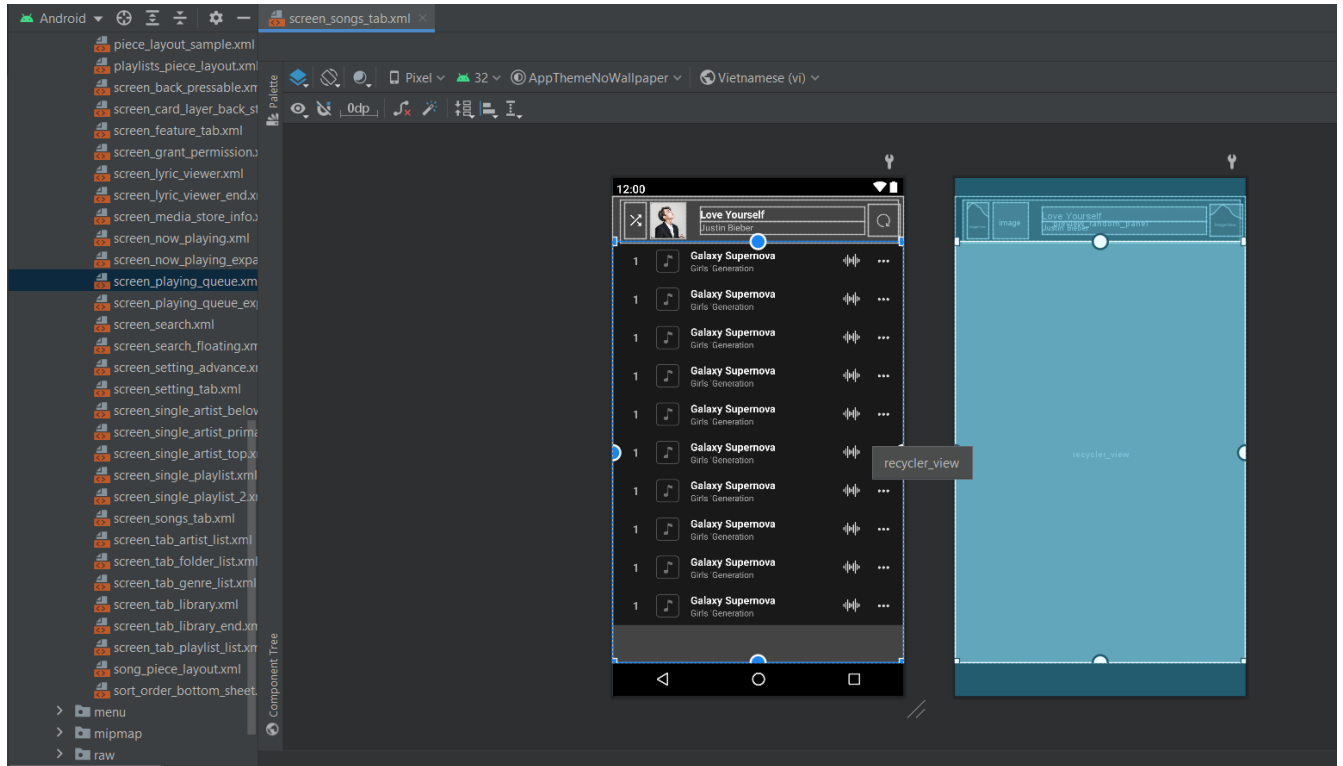
Hình 17 item\_feature\_playlist.xml

- Tiếp đến **feature\_tab\_body.xml** phần thân của các Gợi ý (feature)



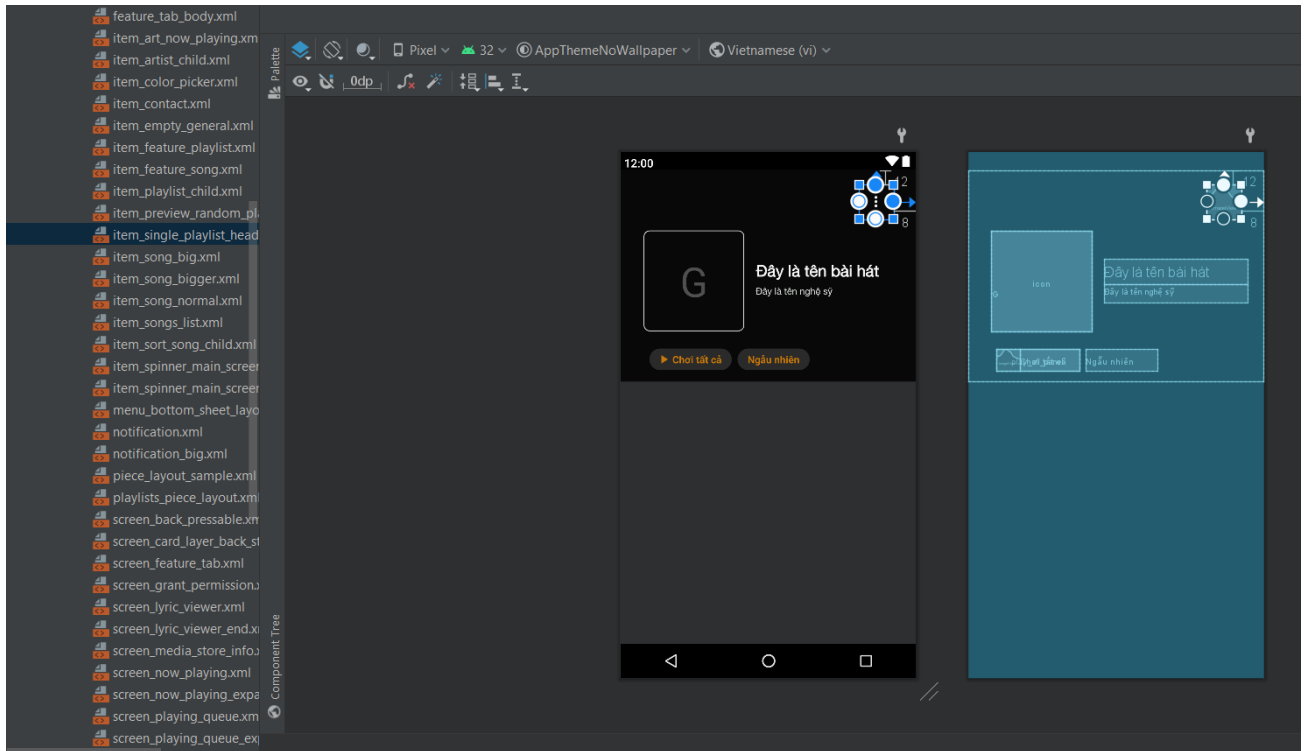
Hình 18 feature\_tab\_body.xml

- **screen\_song\_tab.xml** thể hiện giao diện sắp xếp các bài hát



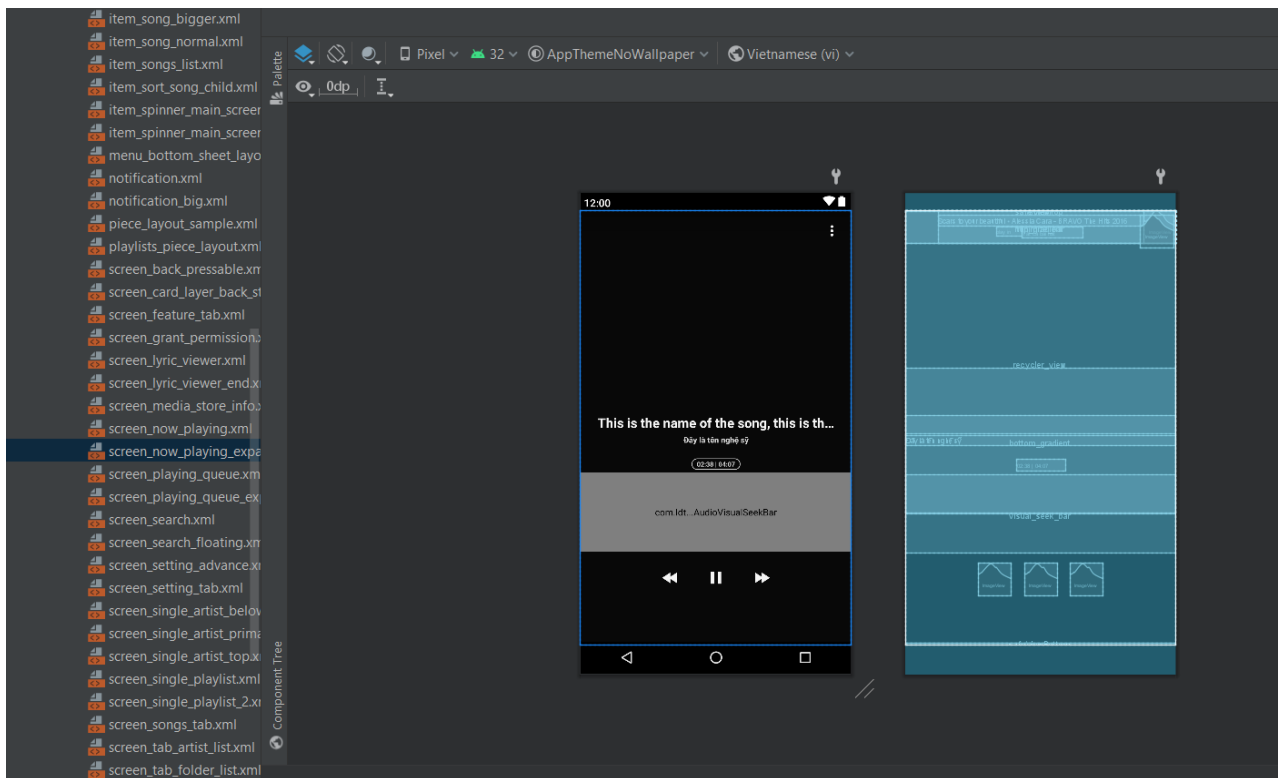
Hình 19 screen\_song\_tab.xml

- **item\_single\_playlist.xml** : giao diện các bài nhạc nằm trong danh sách phát.



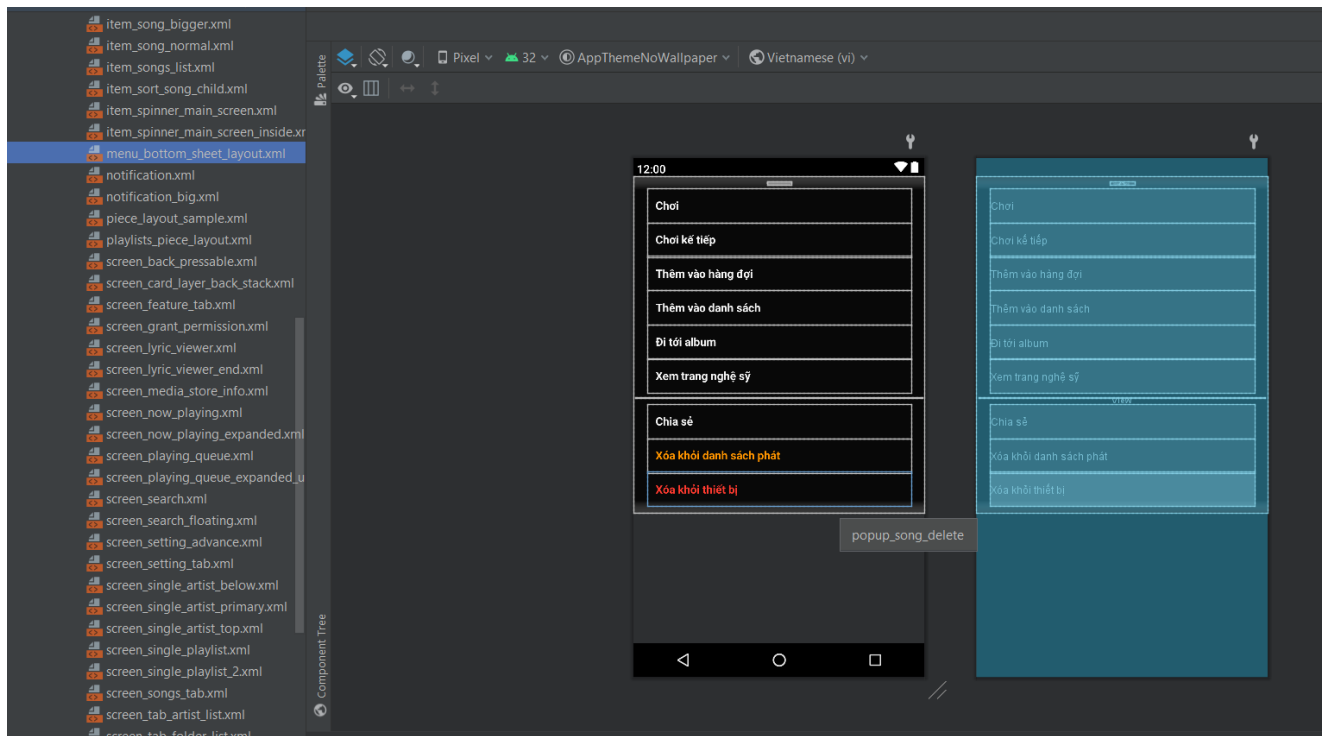
Hình 20 item\_single\_playlist.xml

- **now\_playing.xml**



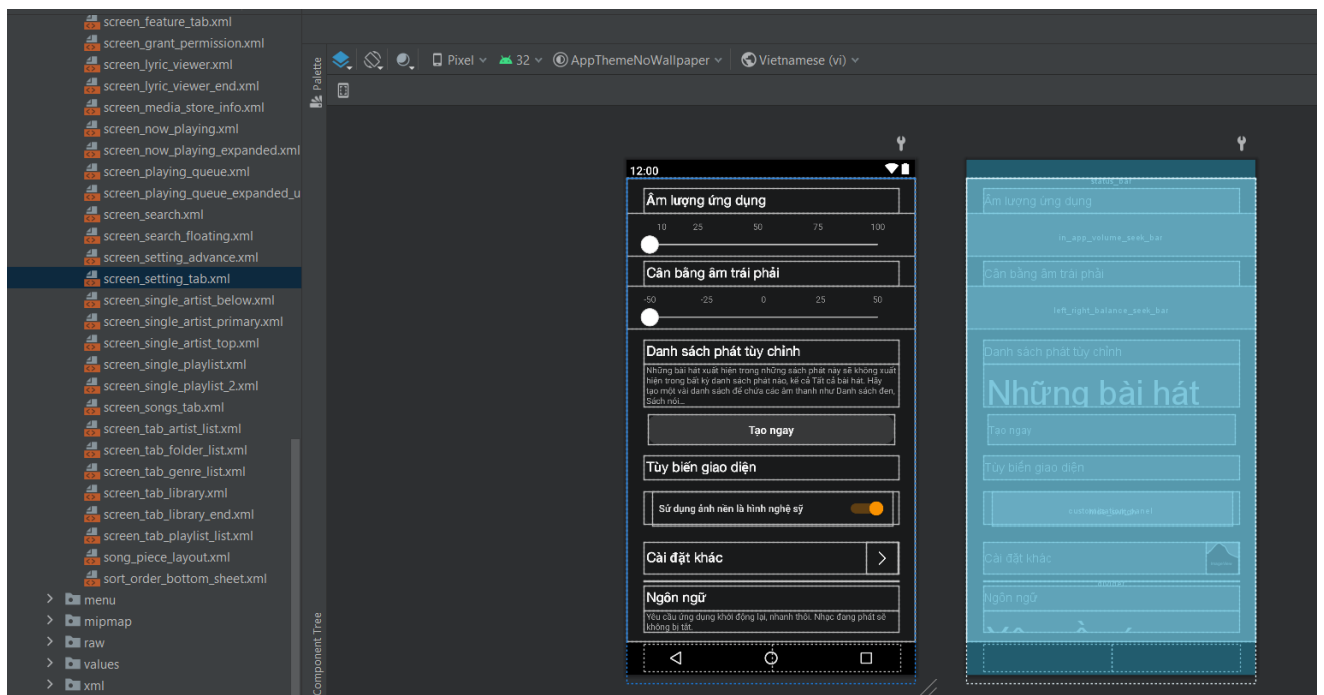
Hình 21 now\_playing.xml

- menu\_bottom\_sheet\_layout.xml




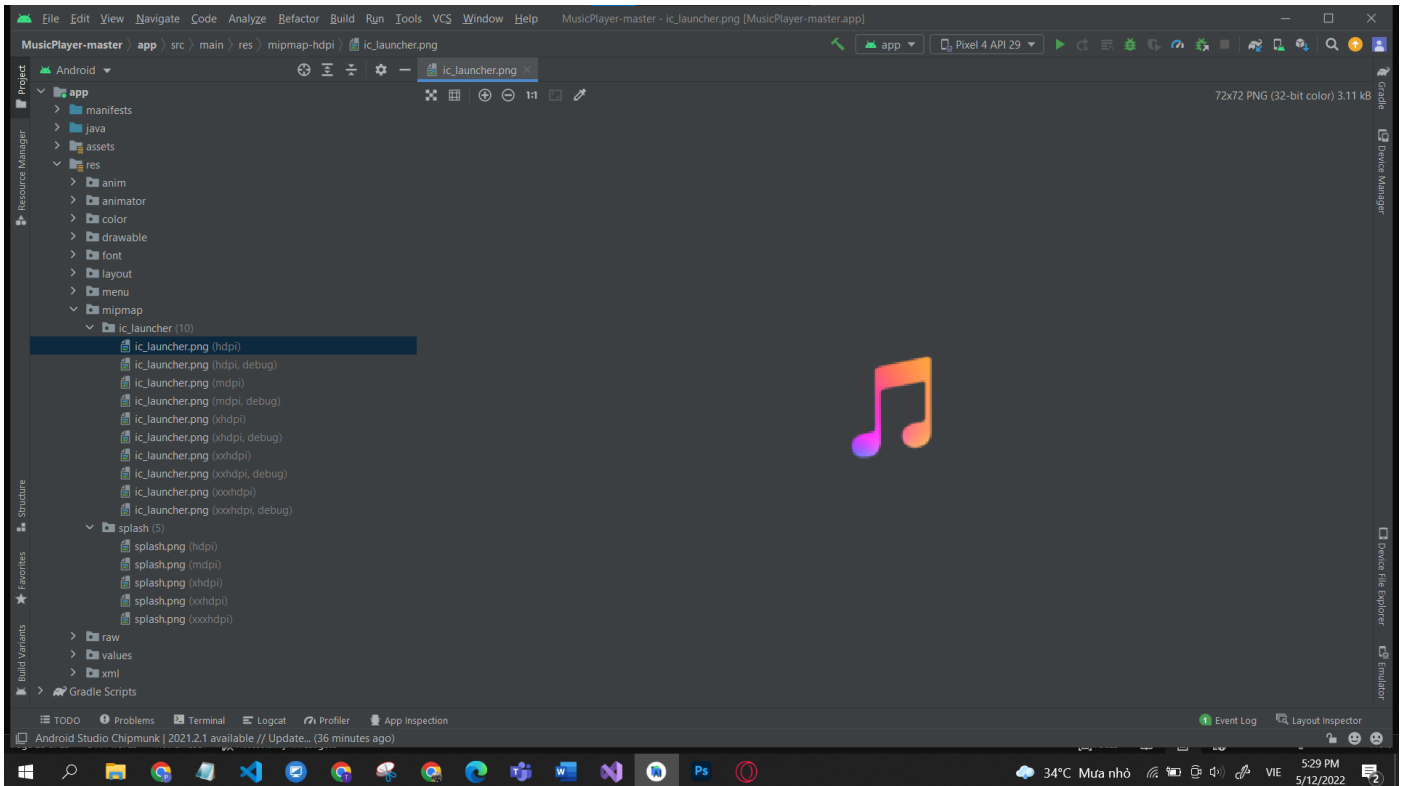
Hình 22 menu\_bottom\_sheet\_layout.xml

- screen\_setting\_tab.xml



Hình 23 screen\_setting\_tab.xml

 **3.3. Gói Mipmap:** Đây là thư mục mà sẽ chứa ảnh logo ứng dụng chúng ta, lúc này có nói là các file hình ảnh sẽ được chứa trong thư mục drawable nhưng ngoại lệ ảnh logo thì chứa trong thư mục mipmap này cho chuẩn.



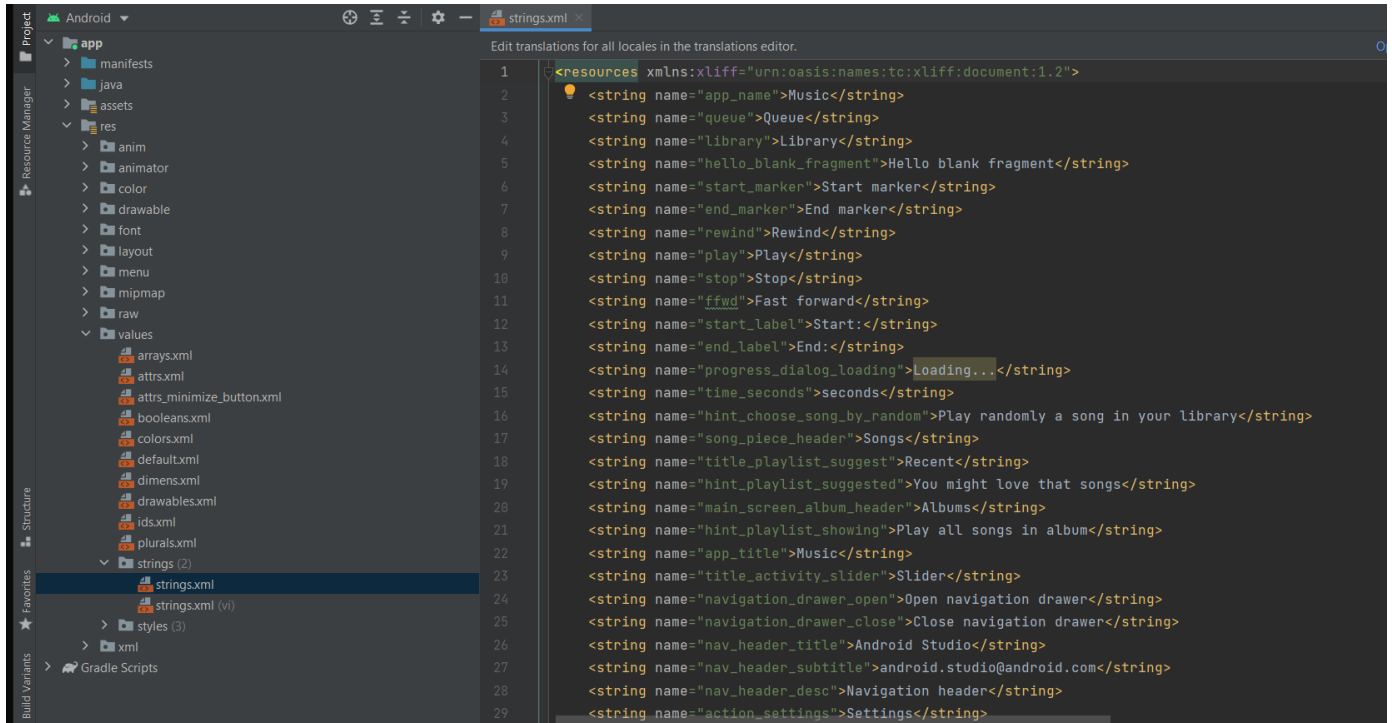
Hình 24 Gói MipMap

 **3.4. Gói Values:** Sẽ có rất nhiều file ở bên trong như sau:

- **color.xml:** đây là file định nghĩa các mã màu trong dự án android, khi sử dụng màu nào chỉ cần gọi tên mã màu đã định nghĩa trong đây ra là xong
- **dimens.xml:** đây là file mà sẽ định nghĩa ra các kích thước như cỡ chữ, chiều cao, chiều rộng các view.
- **strings.xml:** đây là file định nghĩa các đoạn văn bản trong ứng dụng Android của ví dụ như có một đoạn văn bản mà sử dụng đi sử dụng lại trong các màn hình khác nhau, khi set cứng ở nhiều

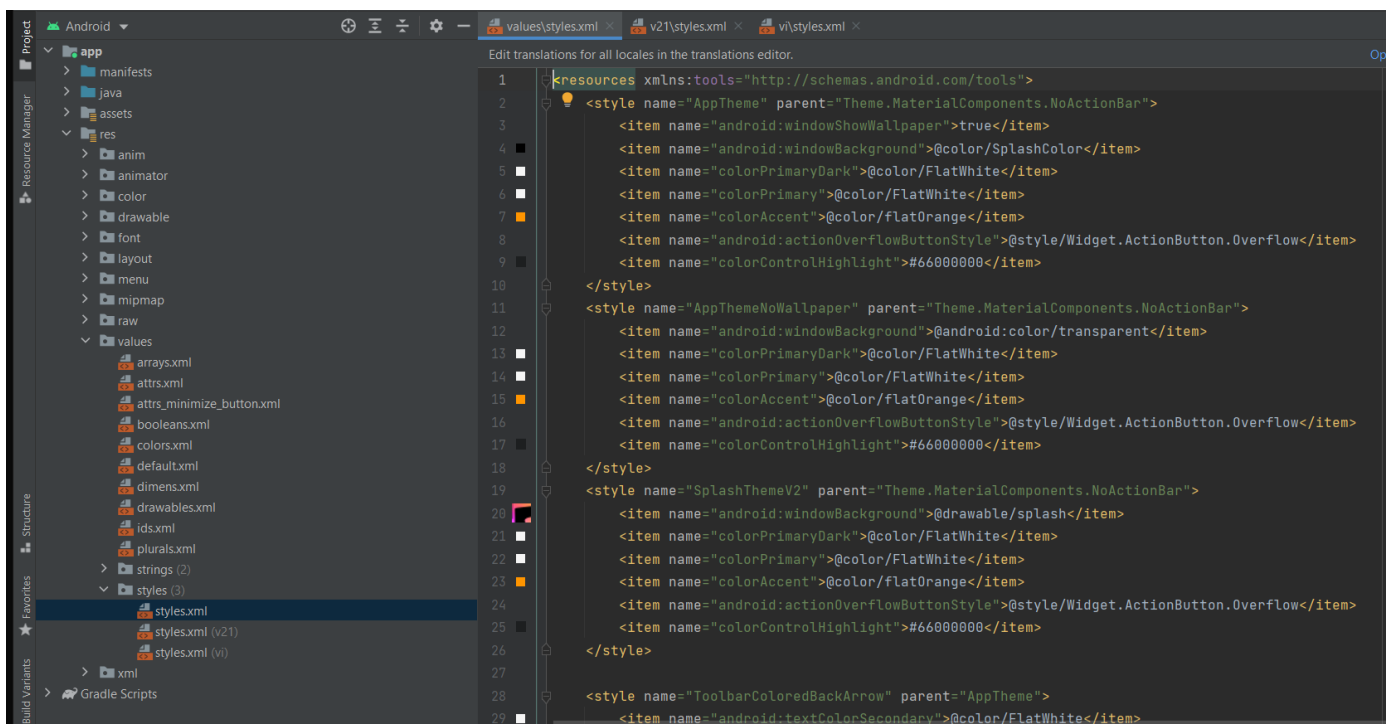


nơi thì khi cần chỉnh sửa thì phải tìm hết tất cả và sửa lại. Bây giờ định nghĩa đoạn văn bản đó trong đây và khi dùng thì gọi ra sử dụng và sau này chỉnh sửa chỉ cần sửa trong đây là xong, nó sẽ apply tất cả mọi nơi.



Hình 25 Strings.xml

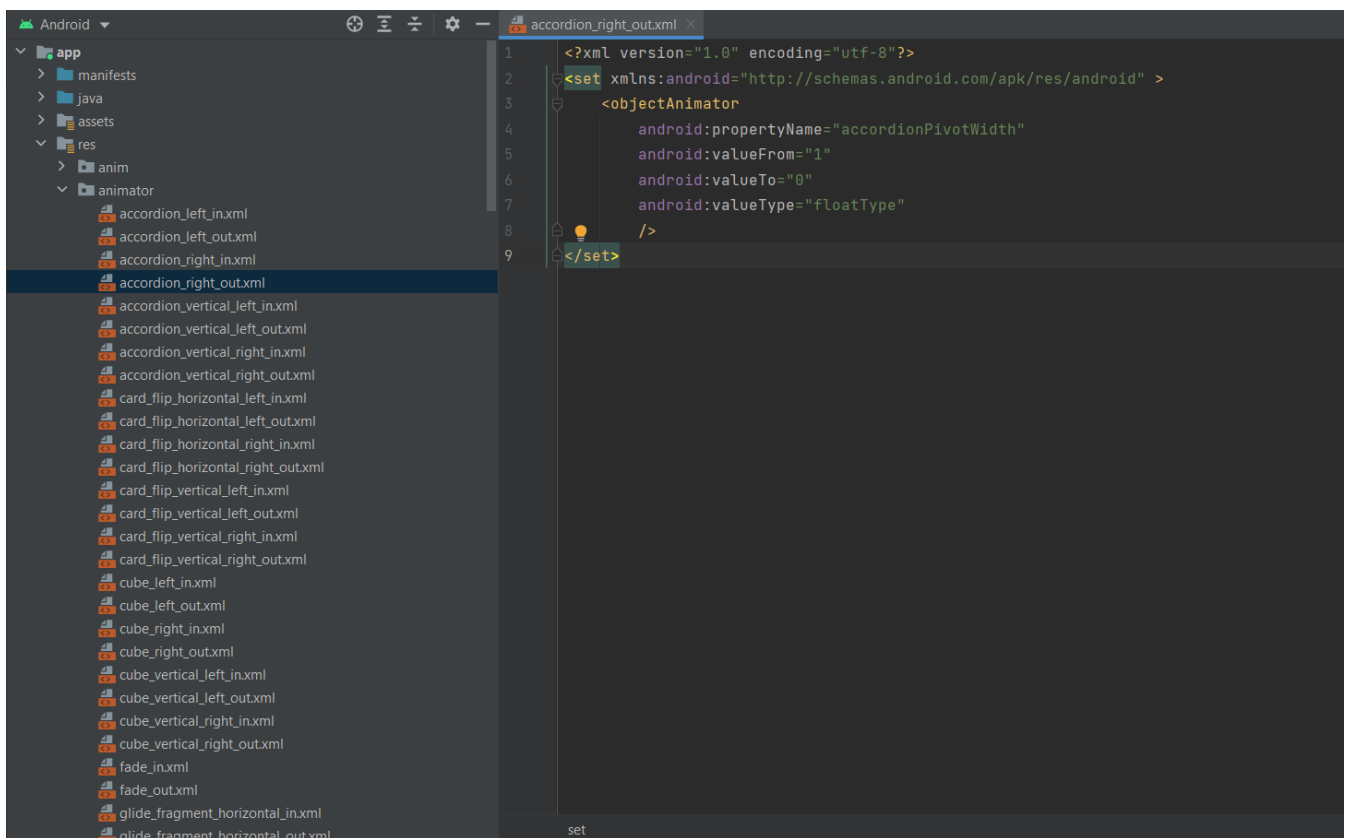
- **styles.xml:** đây chính là nơi định nghĩa các giao diện của các file layout trong thư mục layout đã nói phía trên. Kiểu như thế này nhé, muốn chỉnh một nút Button chiều cao 10dp, chiều rộng 10dp, màu xanh... và lại sử dụng kiểu thiết kế này ở năm màn hình khác nhau. Không thể mỗi màn hình lại định nghĩa lại như thế sẽ làm duplicate code (lặp lại) và sẽ không tối ưu tí nào cả. Thay vào đó chỉ cần định nghĩa một file giao diện như trên và ở mỗi màn hình chỉ cần gọi là xong.




Hình 26 Styles.xml

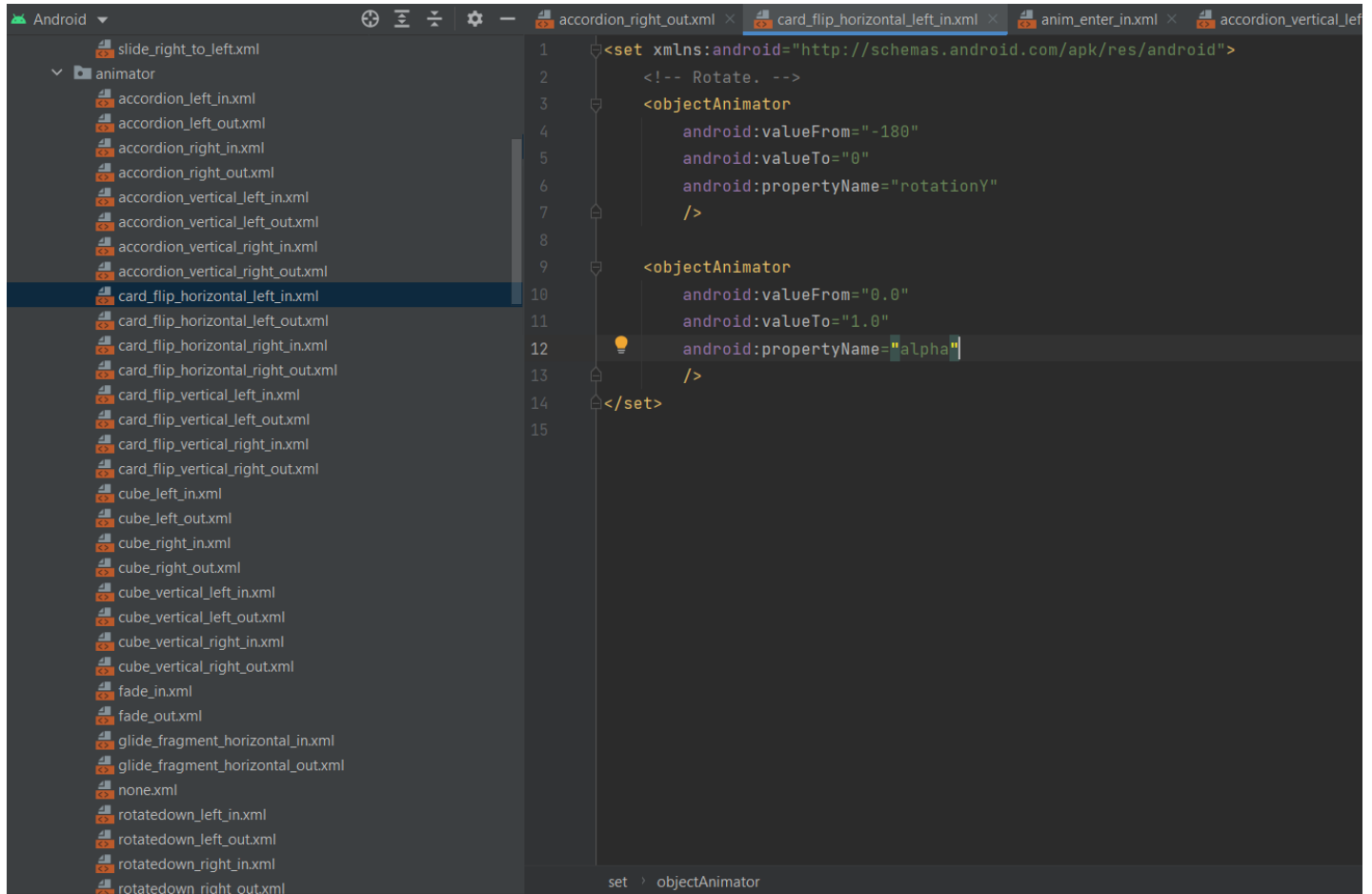
## 3.5. Animato

**3.5.1 Accordion:** là một danh sách được sắp xếp theo chiều dọc hay chiều ngang có chức năng để hiển thị hay ẩn nội dung. Nếu bạn muốn xem lại thì có thể tham khảo ở đây nhé Accordion. Nó giúp chúng ta có thể tiết kiệm diện tích trang và hiển thị những thông tin chính, khi người dùng muốn xem có thể nhấn vào để hiện nội dung của thông tin đó mà không cần phải điều hướng trang nhiều lần.



Hình 27 Accorodion

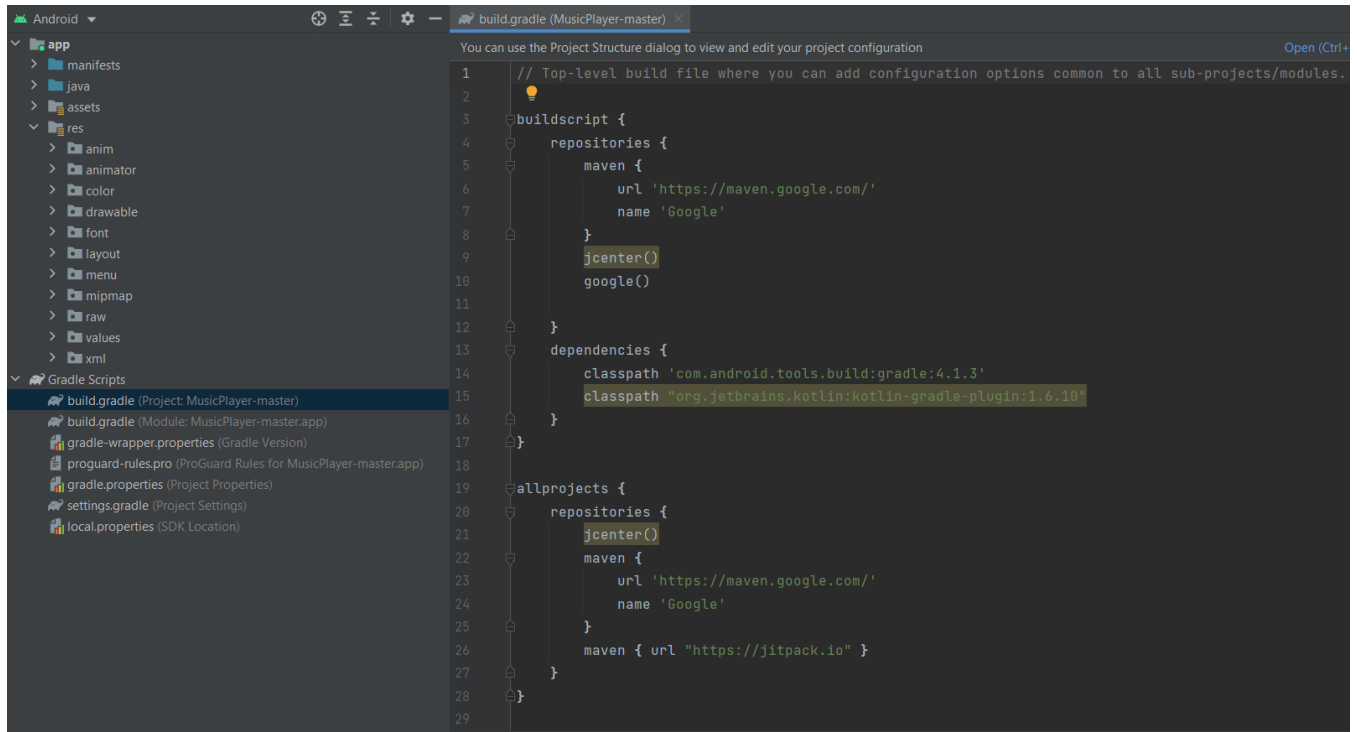
 **3.5.2 Card flip:** là những hoạt ảnh chuyển động, gồm có **Card\_flip\_horizontal\_left\_in/ out**, **Card\_flip\_horizontal\_right\_in/ out**,... là những hoạt ảnh lật ngang bên trái, bên phải, trong và ngoài



**Hình 28 Card\_flip**


#### 4. Tập Grade Scripts

**Build.gradle** là file để thiết lập các thuộc tính cho dự án android như: phiên bản SDK, Vesion ứng dụng, package, thêm thư viện ngoài...



**Hình 29** Grade Scripts

## TỔNG KẾT

 Trong đề án này nhóm đã nghiên cứu, tìm hiểu về ứng dụng phát nhạc trên Android. Đề án đã thực hiện những nhiệm vụ sau:

✓ Tìm hiểu lịch sử và cấu trúc của hệ điều hành Android. Cách mô hình hệ thống ứng dụng, cũng như sơ đồ Use Case mô tả quá trình thực hành của người dùng.

✓ Nắm bắt quy trình làm một phần mềm trên mobile. Tìm hiểu kỹ thuật phát nhạc trên Android.

✓ Kỹ thuật truy cập file trên bộ nhớ ngoài của điện thoại Android. Bước đầu xây dựng thành công ứng dụng Android nghe nhạc offline. Tuy nhiên, do thời gian và khả năng có hạn, nên em chưa đi sâu tìm hiểu được thêm về ứng dụng, vì vậy giao diện ứng dụng vẫn còn sơ sài, xử lý code trên môi trường lập trình Android Studio còn chưa tốt, chương trình vẫn còn nhiều thiết sót

✱ Rất mong thầy và các bạn tận tình giúp đỡ để chương trình ngày càng được hoàn thiện hơn. Trong tương lai, em mong muốn ứng dụng sẽ được phát triển về mọi mặt (giao diện, chức năng,...) và sẽ trở thành một chương trình nghe nhạc online phổ biến.

## TÀI LIỆU THAM KHẢO

1. Guru99.com. (2019). Mobile App Performance Testing: CheckList, Tools (Android & iOS). [online] Available at: <https://www.guru99.com/mobile-app-performance-testingstrategy-tools.html#1> [Accessed ngày 8 tháng 5. 2022]
2. Play.google.com. (2022). [online] Available at: <https://play.google.com/store/apps/details?id=fm.xiami.main&hl=en> [Accessed 26 Jun. 2019].
3. Sohail, A. (2019). Agile Project Management: Understanding and Benefits - DevCrew I/O. [online] DevCrew I/O. Available at: <https://devcrew.io/2017/08/03/agile-basics/> [Accessed ngày 8 tháng 5. 2022].
4. Google Developer Android (Kotlin & Java) <https://developer.android.com/courses/pathways/android-basics-kotlin-two>