**TRƯỜNG ĐẠI HỌC VINH**

**VIỆN KỸ THUẬT VÀ CÔNG NGHỆ**

****

**BÁO CÁO ĐỒ ÁN HỌC PHẦN**

**THIẾT BỊ DI ĐỘNG**

**ĐỀ TÀI: XÂY DỰNG PHẦN MỀM**

**APP MUSIC**

|  |  |
| --- | --- |
| **GVHD**: | Ths. Trần Xuân Hào |
| **Nhóm SVTH**: |  |
|  | 1.Phan Thanh Chi, 59K1, 18574802010169  2.Khương Văn Hải 60k1, 1957480201004  3.Trần Tài Đức 59k1, 18574802010192 |

**Nghệ An, 10/2022**

**Mục Lục**

[Danh mục hình ảnh 4](#_Toc123851660)

[LỜI MỞ ĐẦU 5](#_Toc123851661)

[LỜI CẢM ƠN 6](#_Toc123851662)

[Phân Công Nhiệm Vụ 7](#_Toc123851663)

[CHƯƠNG 1: GIỚI THIỆU BÀI TOÁN 8](#_Toc123851664)

[1.1. Tổng quan về vấn đề được nghiên cứu 8](#_Toc123851665)

[1.2. Khảo sát thực trạng 8](#_Toc123851666)

[1.3. Đánh giá 8](#_Toc123851667)

[1.4. Nhiệm vụ đồ án 9](#_Toc123851668)

[1.5. Đối tượng và phạm vi của ứng dụng 9](#_Toc123851669)

[CHƯƠNG II : PHÂN TÍCH THIẾT KẾ HỆ THỐNG 9](#_Toc123851670)

[**2.1. Tổng quan về các công nghệ** 9](#_Toc123851671)

[2.1.1. Ngôn ngữ Java 9](#_Toc123851672)

[2.2.3. Giới thiệu framework 10](#_Toc123851673)

[2.1.4. Môi trường lập trình 10](#_Toc123851674)

[2.1.5. Các thư viện hỗ trợ 11](#_Toc123851675)

[**3. Các chức năng trong music app** 12](#_Toc123851676)

[3.1. Đọc lấy dữ liệu từ máy lên bằng use-permission 12](#_Toc123851677)

[3.1.2. Định phép quyền truy cập file nhạc tại MainActivity 14](#_Toc123851678)

[3.1.3. Tìm kiếm nhạc trong App Music 15](#_Toc123851679)

[III. TRIÊN KHAI ỨNG DỤNG 18](#_Toc123851680)

[3.1.Thiết kế các Layout ứng dụng 18](#_Toc123851681)

[3.2. Thiết lập điều hướng cho Music App 22](#_Toc123851682)

[3.2.1. Thiết lập SeekBar 22](#_Toc123851683)

[3.2.2. Điều khiển shuffle và repeat 23](#_Toc123851684)

[3.2.3. Điều khiển prevButton 23](#_Toc123851685)

[3.2.4. Điều khiển nextButton 24](#_Toc123851686)

[3.3. Notification music app 25](#_Toc123851687)

[3.3.1. Tổng quan 25](#_Toc123851688)

[3.3.1. Tạo Notification 26](#_Toc123851689)

[4. Phác thảo giao diện 27](#_Toc123851690)

[LỜI KẾT 28](#_Toc123851691)

[TÀI LIỆU THAM KHẢO 29](#_Toc123851692)

# Danh mục hình ảnh

[Hình 1.Các chức năng có có trong app 12](#_Toc123853646)

[Hình 2: Giao diện tìm kiếm nhạc 16](#_Toc123853647)

[Hình 3: Layout music\_iteam 19](#_Toc123853648)

[Hình 4: Layout fragment\_song 19](#_Toc123853649)

[Hình 5: Layout fragment\_now\_play 20](#_Toc123853650)

[Hình 6: Layout album\_item 20](#_Toc123853651)

[Hình 7: Activity\_player 21](#_Toc123853652)

[Hình 8: Activity\_main 21](#_Toc123853653)

[Hình 9: Layout Activity\_album\_detail 21](#_Toc123853654)

[Hình 10: Music App Design 27](#_Toc123853655)

**LỜI MỞ ĐẦU**

**Công nghệ thông tin ngày càng có tính ứng dụng cao trong thực tế, nhất là đối với các công ty trong lĩnh vực quản lí ,kinh doanh lại càng quan trọng, thành phần khác. Công nghệ thông tin ngày càng phát triển mạnh tại các doanh nghiệp.**

**Trước kia, các công ty, các tổ nhân của từng tổ chức… Do đó, khả năng thống nhất một chuẩn không cao, đồng thời bảo mật và tính chính xác trong lưu trữ dữ liệu thấp hơn việc sử dụng một phần mềm Music App.**

**Thế mạnh khi sử dụng một phần mềm Music App cho làm thư giãn căng thẳng sau thời gian làm việc học tập. Music App cho tốt hơn, như có thể lưu trữ thông tin bài nhạc, chơi nhạc.**

**Do đó trước khi đi vào phần nội dung chính của đồ án, chúng em xin gửi lời cảm ơn chân thành tới thầy Trần Xuân Hào là giáo viên trực tiếp hướng dẫn chúng em giúp đỡ chúng em hoàn thành tốt đồ án này.**

# ****LỜI CẢM ƠN****

Đầu tiên, em xin gửi lời cảm ơn chân thành đến Trường Đại học Vinh đã đưa môn học **Android** vào chương trình giảng dạy. Đặc biệt, em xin gửi lời cảm ơn sâu sắc đến giảng viên bộ môn –**Trần Xuân Hào** đã dạy dỗ, truyền đạt những kiến thức cho em trong suốt thời gian học tập vừa qua. Trong thời gian tham gia lớp học của thầy, em đã có thêm cho mình nhiều kiến thức bổ ích, tinh thần học tập hiệu quả, nghiêm túc. Đây chắc chắn sẽ là những kiến thức quý báu, là hành trang để em có thể vững bước sau này.

Đây là một đề tài 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 nhiên, do vốn kiến thức còn nhiều hạn chế và khả năng tiếp thu thực tế còn nhiều bỡ ngỡ. Mặc dù em đã cố gắng hết sức nhưng chắc chắn bài báo cáo khó có thể tránh khỏi những thiếu sót và nhiều chỗ còn chưa chính xác, kính mong thầy xem xét và góp ý để bài đồ án của em được hoàn thiện hơn.

Em xin chân thành cảm ơn!

# Phân Công Nhiệm Vụ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Họ và Tên | Nhiệm Vụ | Ngày Làm | Tổng | Điểm |
| Phan Thanh Chi | - Thiết kế giao diện  -Code list view Music --Kết nối các thiết bị thực  -Load dữ liệu lên View  -Fix lỗi display View  -Làm Player Activity  -Làm controlMusic  -Notification Music  -NowPlayMusic  -Làm word | 10/10 – 12/10  20/11 – 1/11  5/11-15/11  20/11 – 2/12  14/12 – 18/12  22/12 – 24/12  26/12-31/12 | 60% |  |
| Khương Văn Hải | -Làm word  -frangment\_songs  -musicItems  -popup  -search  -SeekBar  -Delete songs | 10/10-23/10  29/10 – 6/11 | 20% |  |
| Trần Tài Đức | **-**làm word  **-**làm playerActivity  -Activityalbum  -albumItem  -control prev và nextButton  -Find songs | 10/10-23/10  29/10 – 12/11 | 20% |  |

# CHƯƠNG 1: GIỚI THIỆU BÀI TOÁN

## 1.1. Tổng quan về vấn đề được nghiên cứu

* Nghiên cứu về việc ứng dụng nền tảng Công Nghệ Thông Tin vào việc quản lý, thư giản trong các công việc, giải trí
* Nghiên cứu về công nghệ làm app bằng AndroidStudio, Java
* Nghiên cứu về các nền tảng để lập trình di động sử dụng ngôn ngữ Java và các công nghệ hỗ trợ khác

## 1.2. Khảo sát thực trạng

* **Người dùng**:
* Người dùng có thể sử dụng app music để thư giản sau những ngày mệt mỏi
* Người dùng sử dụng các tài nguyên nhạc.

## 1.3. Đánh giá

**Ưu điểm**

* Xây dựng app một cách thuận tiện, đơn giản và nhanh chóng.
* Cập nhật các thông tin về thông tin bài hát, album, ca sĩ và những bài hát mới, hot.
* Giao diện tiếp cận người dùng, thuận tiện sử dụng và đơn giản.
* Hoạt động ổn định, tính bảo mật cao.

**Nhược điểm**

* Giao diện còn đơn giản chưa được tối ưu.
* Một số tính năng còn hạn chế.
* Code còn dài chưa được tối ưu hóa và app hoạt động chưa mượt.

## 1.4. Nhiệm vụ đồ án

* Tìm hiểu các quy trình xây dựng một app
* Tìm hiểu ngôn ngữ lập trình Java

## 1.5. Đối tượng và phạm vi của ứng dụng

* Đối tượng: Tất cả mọi người

# CHƯƠNG II : PHÂN TÍCH THIẾT KẾ HỆ THỐNG

## **2.1. Tổng quan về các công nghệ**

### 2.1.1. Ngôn ngữ Java

* Java là một trong những ngôn ngữ lập trình hướng đối tượng. Nó được sử dụng trong phát triển phần mềm, trang web, game hay ứng dụng trên các thiết bị di động.
* Java được khởi đầu bởi James Gosling và bạn đồng nghiệp ở Sun MicroSystem năm 1991. Ban đầu Java được tạo ra nhằm mục đích viết phần mềm cho các sản phẩm gia dụng, và có tên là Oak.Java được phát hành năm 1994, đến năm 2010 được Oracle mua lại từ Sun MicroSystem.
* Java được tạo ra với tiêu chí “Viết (code) một lần, thực thi khắp nơi” (Write Once, Run Anywhere  – WORA). Chương trình phần mềm viết bằng Java có thể chạy trên mọi nền tảng (platform) khác nhau thông qua một môi trường thực thi với điều kiện có môi trường thực thi thích hợp hỗ trợ nền tảng đó.
* Ở cấp độ mã nguồn: Kiểu dữ liệu trong Java nhất quán cho tất cả các hệ điều hành và phần cứng khác nhau. Java có riêng một bộ thư viện để hỗ trợ vấn đề này. Chương trình viết bằng ngôn ngữ Java có thể biên dịch trên nhiều loại máy khác nhau mà không gặp lỗi.
* Ở cấp độ nhị phân: Một mã biên dịch có thể chạy trên nhiều nền tảng khác nhau mà không cần dịch lại mã nguồn. Tuy nhiên cần có Java Virtual Machine để thông dịch đoạn mã này.
* Có tính an toàn và bảo mật.

### 2.2.3. Giới thiệu framework

* **Gilde** là khung hình ảnh và phương tiện mã nguồn mở nhanh và hiệu quả, bao gồm giải mã phương tiện, bộ nhớ đệm và bộ nhớ đệm đĩa cũng như tổng hợp tài nguyên và một giao diện đơn giản và dễ sử dụng.
* **Glide** hỗ trợ tìm nạp, giải mã và hiển thị video tĩnh, hình ảnh và GIF động. **Glide** bao gồm một API linh hoạt cho phép các nhà phát triển cắm vào hầu hết mọi ngăn xếp mạng. Theo mặc định, Glide sử dụng HttpUrlConnection ngăn xếp dựa trên tùy chỉnh, nhưng cũng bao gồm các thư viện tiện ích bổ trợ cho dự án Volley của Google hoặc thư viện OkHttp của Square.Trọng tâm chính của Glide là làm cho việc cuộn bất kỳ loại danh sách hình ảnh nào mượt mà và nhanh nhất có thể, nhưng Glide cũng hiệu quả đối với hầu hết mọi trường hợp bạn cần tìm nạp, thay đổi kích thước và hiển thị hình ảnh từ xa.

### 2.1.4. Môi trường lập trình

Android Studio là môi trường tích hợp IDE\* để viết code và phát triển ứng dụng trên nền tảng hệ điều hành Android, dành cho Google Android Development được ra mắt vào ngày 16 tháng 5 năm 2013, trong sự kiện I/O 2013 của Google.

Android Studio chứa tất cả các công cụ để hỗ trợ việc phát hành ứng dụng Android như: thiết kế, kiểm tra, gỡ lỗi và cấu hình ứng dụng. Android Studio sử dụng Gradle để quản lý dự án của bạn.

IDE viết tắt của Integrated Development Environment, được gọi là “Môi trường thiết kế hợp nhất" hay "Môi trường gỡ lỗi hợp nhất" (tiếng Anh: Integrated Debugging Environment) là một loại phần mềm máy tính có công dụng giúp đỡ các lập trình viên trong việc phát triển phần mềm.

Để phát triển ứng dụng đầu tiên của mình, bạn cần tải xuống Android Studio cho nền tảng ưa thích (Windows, Mac OS X hoặc Linux) từ trang web dành cho nhà phát triển Android. Android Studio có thể hỗ trợ phát triển và thử nghiệm ứng dụng của bạn trên thiết bị thực hoặc trình giả lập.

### 2.1.5. Các thư viện hỗ trợ

**Xác định yêu cầu**

* Yêu cầu chung:
* Kiểu App Music: đơn giản, thân thiện, có menu, các chức năng control app
* Phong cách thiết kế: đơn giản
* Không sử dụng các font chữ không chuẩn, khó đọc. Nên sử dụng các font Unicode chuẩn như Arial, Tahoma và Time News Roman.
* Giao diện:
* Giao diện thân thiện với người dùng, dễ sử dụng
* Trang trí vừa phải, màu sắc dễ nhìn, ổn định
* Cho phép người dùng có thể chỉnh sửa thông tin sản phẩm và cập nhập mới.
* Cho phép người dùng có thể xóa sản phẩm.
* Tác vụ thống kê:
* Bao gồm: thống kê lượng người truy cập, thống kê đơn đặt hàng

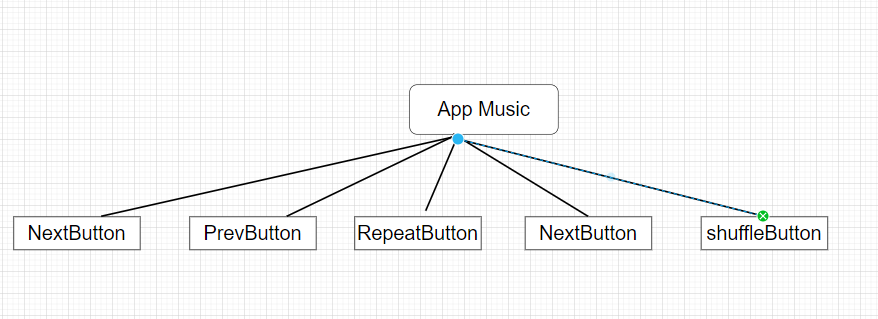
**Yêu cầu hệ thống**

* Hệ thống sử dụng hệ đều hành android kết hợp với điện thoại thông minh
* Ứng dụng đảm bảo được các yêu cầu người dùng.
* Các chức năng của hệ thống hoạt động ổn định.

**Yêu cầu chức năng**

* App Music phải có đầy các chức năng cơ bản như load nhạc, liệt kê ra các album theo nhạc sĩ, tên bài hát
* Chức năng control như playMusic, preMusic, nextMusic,repeatMusic.

## **3. Các chức năng trong music app**

****

Hình .Các chức năng có có trong app

## 3.1. Đọc lấy dữ liệu từ máy lên bằng use-permission

Cú pháp:

<uses-permission  
 android:name="android.permission.WRITE\_EXTERNAL\_STORAGE"  
 tools:ignore="ScopedStorage" />

Chỉ định một quyền hệ thống mà người dùng phải cấp để ứng dụng hoạt động chính xác. Quyền được người dùng cấp khi ứng dụng được cài đặt (trên thiết bị chạy Android 5.1 trở xuống) hoặc khi ứng dụng đang chạy (trên thiết bị chạy Android 6.0 trở lên) nền tảng cơ sở xác định tại [android.Manifest.permission](https://developer.android.com/reference/android/Manifest.permission).

Tên quyền. Đây có thể là quyền do ứng dụng xác định bằng phần tử [<permission>](https://developer.android.com/guide/topics/manifest/permission-element), quyền do ứng dụng khác xác định hoặc một trong các quyền tiêu chuẩn của hệ thống ["android.permission.CAMERA"](https://developer.android.com/reference/android/Manifest.permission#CAMERA) hoặc ["android.permission.READ\_CONTACTS"](https://developer.android.com/reference/android/Manifest.permission#READ_CONTACTS)). Trong những ví dụ này, tên quyền thường bao gồm cả tên gói dưới dạng tiền tố. Cấp API cao nhất mà hệ thống sẽ cấp quyền này cho ứng dụng. Thao tác đặt thuộc tính này sẽ hữu ích nếu quyền mà ứng dụng yêu cầu không còn cần thiết kể từ khi bắt đầu ở một cấp API cụ thể. Bắt đầu từ Android 4.4 (API cấp 19), ứng quyền [WRITE\_EXTERNAL\_STORAGE](https://developer.android.com/reference/android/Manifest.permission#WRITE_EXTERNAL_STORAGE) nữa khi muốn ghi vào các thư mục dành riêng cho chính ứng dụng đó trên bộ nhớ ngoài (các thư mục do [getExternalFilesDir()](https://developer.android.com/reference/android/content/Context#getExternalFilesDir(java.lang.String)) cung cấp). Tuy nhiên, bắt buộc phải yêu cầu quyền này đối với API cấp 18 trở xuống.

3.1.1. Khởi tại các đối tượng trong AppMusic

Chúng ta khởi tạo đối tượng thông qua file MusicFiles.java để khởi tọa đối tượng bài hát bằng hướng đối tượng qua các phương thức get và set của đối tượng để tạo path, title, artist, album,duaration, id thông tin từ bài hát.

*public class* MusicFiles {  
 *private* String path;  
 *private* String title;  
 *private* String artist;  
 *private* String album;  
 *private* String duration;  
 *private* String id;  
  
 *public* MusicFiles(String path, String title, String artist, String album, String duration, String id) {  
 *this*.path = path;  
 *this*.title = title;  
 *this*.artist = artist;  
 *this*.album = album;  
 *this*.duration = duration;  
 *this*.id = id;

*public* String getPath() {  
 *return* path;  
}  
  
*public void* setPath(String path) {  
 *this*.path = path;  
}  
  
*public* String getTitle() {  
 *return* title;  
}  
  
*public void* setTitle(String title) {  
 *this*.title = title;  
}  
  
*public* String getArtist() {  
 *return* artist;  
}  
  
*public void* setArtist(String artist) {  
 *this*.artist = artist;  
}  
  
*public* String getAlbum() {  
 *return* album;  
}  
  
*public void* setAlbum(String album) {  
 *this*.album = album;  
}  
  
*public* String getDuration() {  
 *return* duration;  
}  
  
*public void* setDuration(String duration) {  
 *this*.duration = duration;  
}  
  
*public* String getId() {  
 *return* id;  
}  
  
*public void* setId(String id) {  
 *this*.id = id;  
}

## 3.1.2. Định phép quyền truy cập file nhạc tại MainActivity

Tại đây chúng ta gọi hàm permission() gồm có các phương thức checkSelfPermission trong onCreate dùng để kiểm tra truyền truy cập đến thư mục nhạc trong ứng dụng và đọc từ Manifest thông qua phương thức WRITE\_EXTERNAL\_STORAGE để chúng ta có thể đọc và từ thiết bị lên trong điều kiện nếu không có nhạc trong thiết bị hoặc có nhạc trong thiết bị.

*private void* permission(){  
 *if*(ContextCompat.*checkSelfPermission*(getApplicationContext(), Manifest.permission.*WRITE\_EXTERNAL\_STORAGE*) != PackageManager.*PERMISSION\_GRANTED*)  
 {  
 ActivityCompat.*requestPermissions*(MainActivity.*this*, *new* String[]{Manifest.permission.*WRITE\_EXTERNAL\_STORAGE*}, *REQUEST\_CODE*);  
 }*else*{  
  
 *musicFiles* = *getAllAudio*(*this*);  
 initViewPager();  
 }  
}

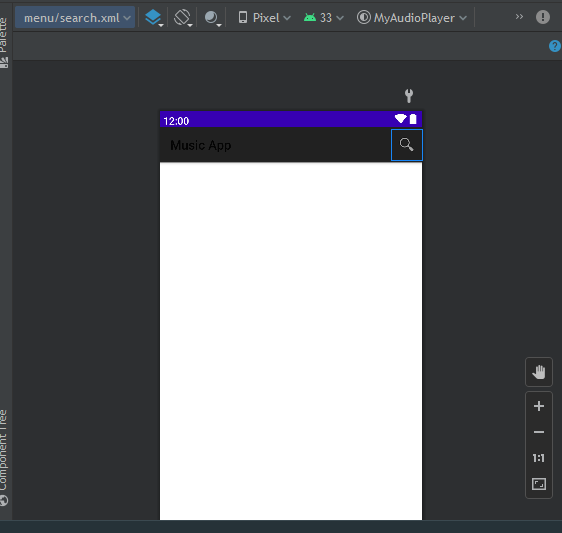
Sau đó tạo một ArrayList để lấy từng đối tượng trong file nhạc vào ứng dụng bằng tham số MusicFiles đã gọi trước đó và gọi nó thông qua các phương thức *EXTERNAL\_CONTENT\_URI* bằng *MediaStore*để trỏ đến từng đối tượng trong một bài hát.

*public static* ArrayList<MusicFiles> getAllAudio(Context context)  
{  
 ArrayList<String> duplicate = *new* ArrayList<>();  
 ArrayList<MusicFiles> tempAudioList = *new* ArrayList<>();  
 Uri uri = MediaStore.Audio.Media.*EXTERNAL\_CONTENT\_URI*;  
 String[] projection = {  
 MediaStore.Audio.Media.*ALBUM*,  
 MediaStore.Audio.Media.*TITLE*,  
 MediaStore.Audio.Media.*DURATION*,  
 MediaStore.Audio.Media.*DATA*,MediaStore.Audio.Media.*ARTIST*,  
 MediaStore.Audio.Media.*\_ID* };

## 3.1.3. Tìm kiếm nhạc trong App Music

Tại thư mục res -> menu chúng ta sẽ thiết kế thanh search bài nhạc tên file search.xml

<item  
 *android:title*="@string/search"  
 *android:id*="@+id/search\_option"  
 *android:icon*="@android:drawable/ic\_menu\_search"  
 *app:showAsAction*="ifRoom"  
 *app:actionViewClass*="androidx.appcompat.widget.SearchView" />



Hình : Giao diện tìm kiếm nhạc

Để thực thực thi tìm kiếm nhạc tại file MainActivity chúng ta tạo phương thức **onCreateOptionMenu** chúng ta có thể khai báo các mục cho menu tùy chọn từ lớp con của bạn [Activity](https://developer.android.com/reference/android/app/Activity) hoặc [Fragment](https://developer.android.com/reference/android/app/Fragment)lớp con. Nếu cả hoạt động và (các) đoạn của bạn khai báo các mục cho menu tùy chọn, chúng sẽ được kết hợp trong giao diện người dùng. Các mục của hoạt động xuất hiện đầu tiên, tiếp theo là các mục của từng phân đoạn theo thứ tự mà mỗi phân đoạn được thêm vào hoạt động. Nếu cần, chúng ta có thể sắp xếp lại thứ tự các mục menu với android:orderInCategory thuộc tính trong mỗi mục <item>bạn cần di chuyển.Để chỉ định menu tùy chọn cho một hoạt động,hãy nghi đè [onCreateOptionsMenu()](https://developer.android.com/reference/android/app/Activity#onCreateOptionsMenu(android.view.Menu))(các đoạn cung cấp lệnh [onCreateOptionsMenu()](https://developer.android.com/reference/android/app/Fragment#onCreateOptionsMenu(android.view.Menu,%20android.view.MenuInflater))gọi lại của riêng chúng). [**MenuInflater**](https://developer.android.com/reference/android/view/MenuInflater) cho phép chúng ta menu ngữ cảnh từ [tài nguyên menu](https://developer.android.com/guide/topics/resources/menu-resource) . Các tham số của phương thức gọi lại bao gồm đối tượng mà [View](https://developer.android.com/reference/android/view/View) người dùng đã chọn và [**ContextMenu**.**ContextMenuInfo**](https://developer.android.com/reference/android/view/ContextMenu.ContextMenuInfo) đối tượng cung cấp thông tin bổ sung về mục đã chọn. Nếu hoạt động của bạn có một số chế độ xem mà mỗi chế độ xem cung cấp một menu ngữ cảnh khác nhau, thì bạn có thể sử dụng các tham số này để xác định menu ngữ cảnh nào sẽ tăng cao. Tạo một phương thức truy vấn query song – find songs query method onQueryOptionsSubmit

*public boolean* onCreateOptionsMenu(*Menu* menu) {  
 getMenuInflater().inflate(R.menu.*search*, menu);  
 *MenuItem* menuItem = menu.findItem(R.id.*search\_option*);  
 SearchView searchView = (SearchView) menuItem.getActionView();  
 searchView.setOnQueryTextListener(*this*);  
 *return super*.onCreateOptionsMenu(menu);  
}

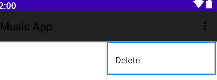
@Override  
*public boolean* onQueryTextSubmit(String query) {  
 *return false*;  
}

Ta tạo hàm **onQueryTextChange** để tạo điều kiện tìm kiếm trong với input nhập vào và trả về List danh sách bài hát

*public boolean* onQueryTextChange(String newText) {  
 String userInput = newText.toLowerCase();  
 ArrayList<MusicFiles> myFlies = *new* ArrayList<>();  
 *for* (MusicFiles song : *musicFiles*){  
 *if* (song.getTitle().toLowerCase().contains(userInput));  
 {  
 myFlies.add(song);  
 }  
 }  
 SongsFragment.*musicAdapter*.updateList(myFlies);  
 *return true*;  
}

3.1.4. Xóa file nhạc trong ứng dụng

Ta sẽ tạo giao diện popup.xml bên trong tại menu để tạo phần tương tác xóa bài hát trong ứng dụng trong music app.

<menu *xmlns:android*="http://schemas.android.com/apk/res/android"  
 *xmlns:app*="http://schemas.android.com/apk/res-auto">  
 <item  
 *android:title*="@string/delete"  
 *android:id*="@+id/delete"  
 *app:showAsAction*="never" />  
</menu>

Chúng ta sẽ tạo phương thức deleteFile để xóa nhạc với các phương thức và các biến tạo như **Uri, File, Snackbar** với các method **getContenResolve**r và notifyItemRemoved để lấy vị trí position index của bài hát và tiến hành xóa và gọi hiện thị bằng phương thức **setOnClickListener** gọi đến popup và hiện thị giao diện.

*private void* deleteFile(*int* position, View v){  
 Uri contentUri = ContentUris.*withAppendedId*(MediaStore.Audio.Media.*EXTERNAL\_CONTENT\_URI*, Long.*parseLong*(*mFiles*.get(position).getId()));  
  
 File file = *new* File(*mFiles*.get(position).getPath());  
 *boolean* deleted = file.delete();  
 *if* (deleted){  
 mContext.getContentResolver().delete(contentUri, *null*, *null*);  
 *mFiles*.remove(position);  
 notifyItemRemoved(position);  
 notifyItemRangeChanged(position, *mFiles*.size());  
 Snackbar.*make*( v, " File deleted: ", Snackbar.*LENGTH\_LONG*)  
 .show();  
 }*else*{  
 Snackbar.*make*( v, " Cant not be deleted: ", Snackbar.*LENGTH\_LONG*)  
 .show();  
 }  
}

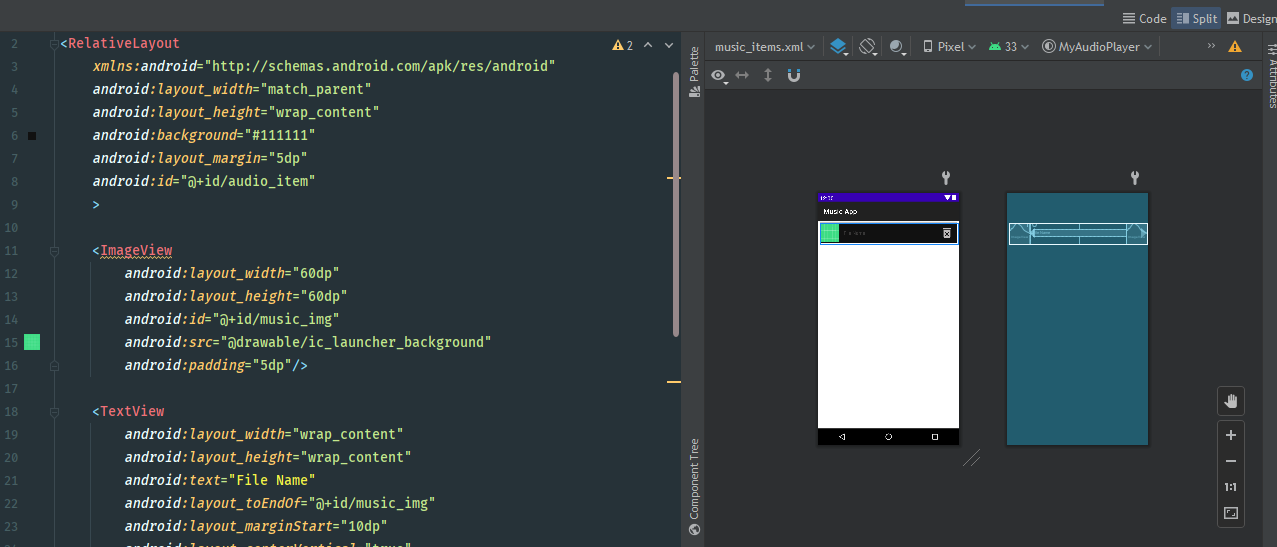
holder.menuMore.setOnClickListener(*new* View.OnClickListener() {  
 @Override  
 *public void* onClick(*final* View v) {  
 PopupMenu popupMenu = *new* PopupMenu(mContext, v);  
 popupMenu.getMenuInflater().inflate(R.menu.*popup*, popupMenu.getMenu());  
 popupMenu.show();  
 popupMenu.setOnMenuItemClickListener((item) ->{  
 *switch* (item.getItemId()){  
 *case* R.id.*delete*:  
 Toast.*makeText*(mContext, " delete clicked!", Toast.*LENGTH\_SHORT*).show();  
 deleteFile(position, v);  
 *break*;

# III. TRIỂN KHAI ỨNG DỤNG

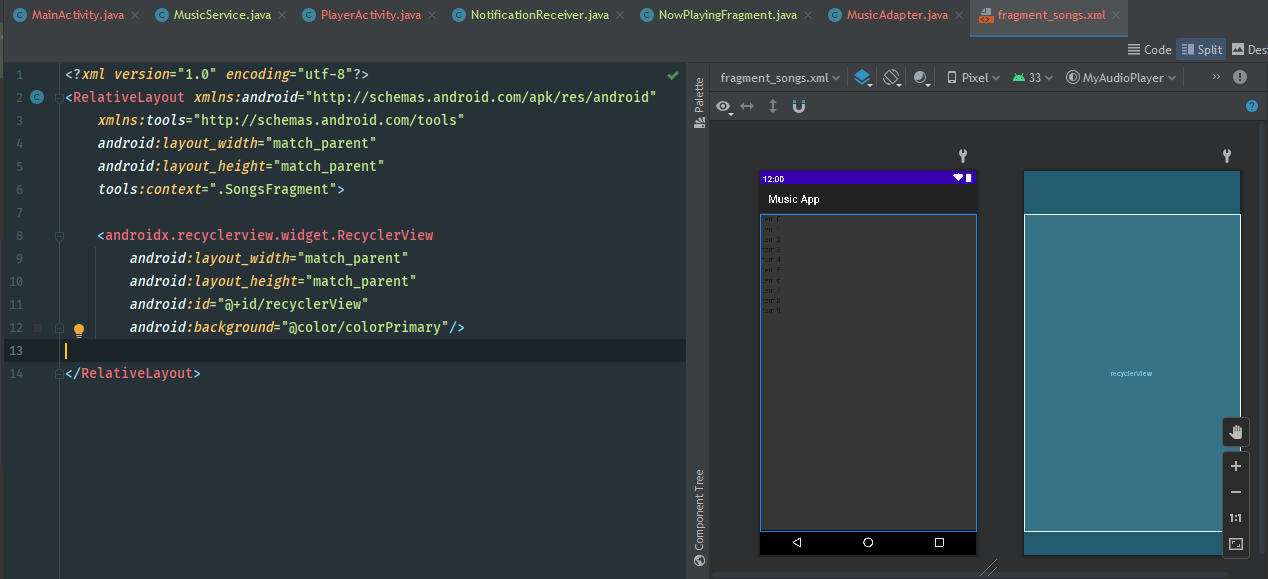
## 3.1.Thiết kế các Layout ứng dụng

Bố cục xác định cấu trúc cho giao diện người dùng trong ứng dụng của bạn, chẳng hạn như trong một [hoạt động](https://developer.android.com/guide/components/activities) . Tất cả các thành phần trong bố cục được xây dựng bằng cách sử dụng hệ thống phân cấp [View](https://developer.android.com/reference/android/view/View)và [ViewGroup](https://developer.android.com/reference/android/view/ViewGroup)đối tượng.

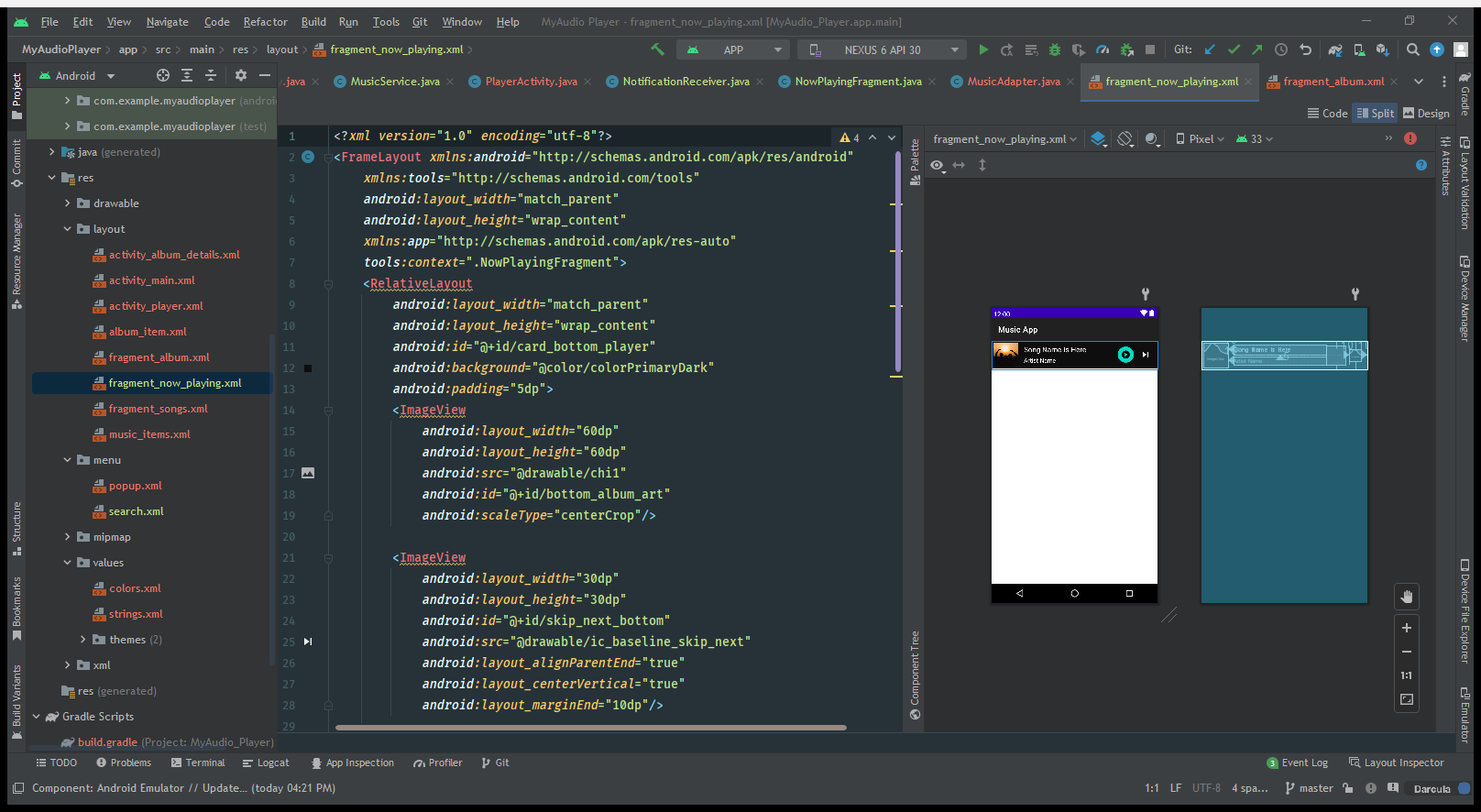
Thiết lập giao diện bằng những thành phần trong ứng dụng bằng các loại layout: Relative, ImageView,FrameLayout, FloatingActionButton…



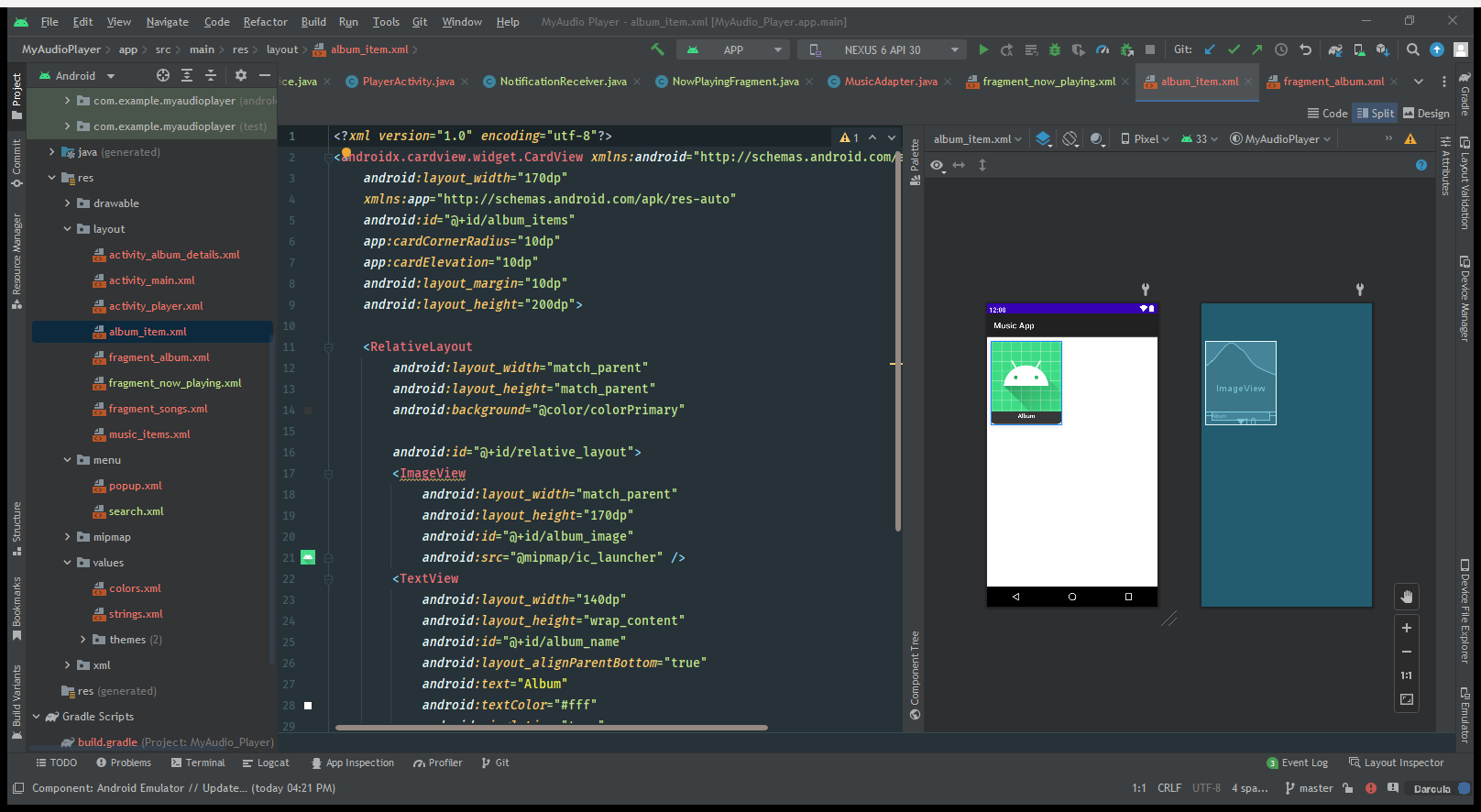
Hình : Layout music\_iteam



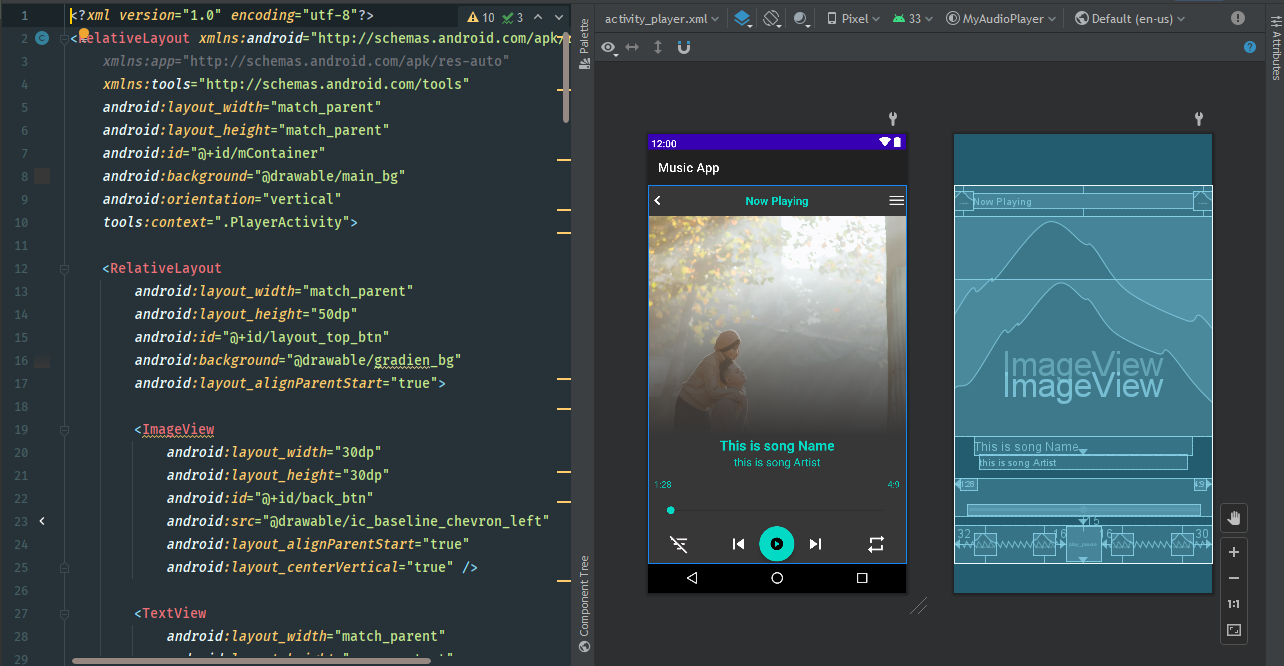
Hình : Layout fragment\_song



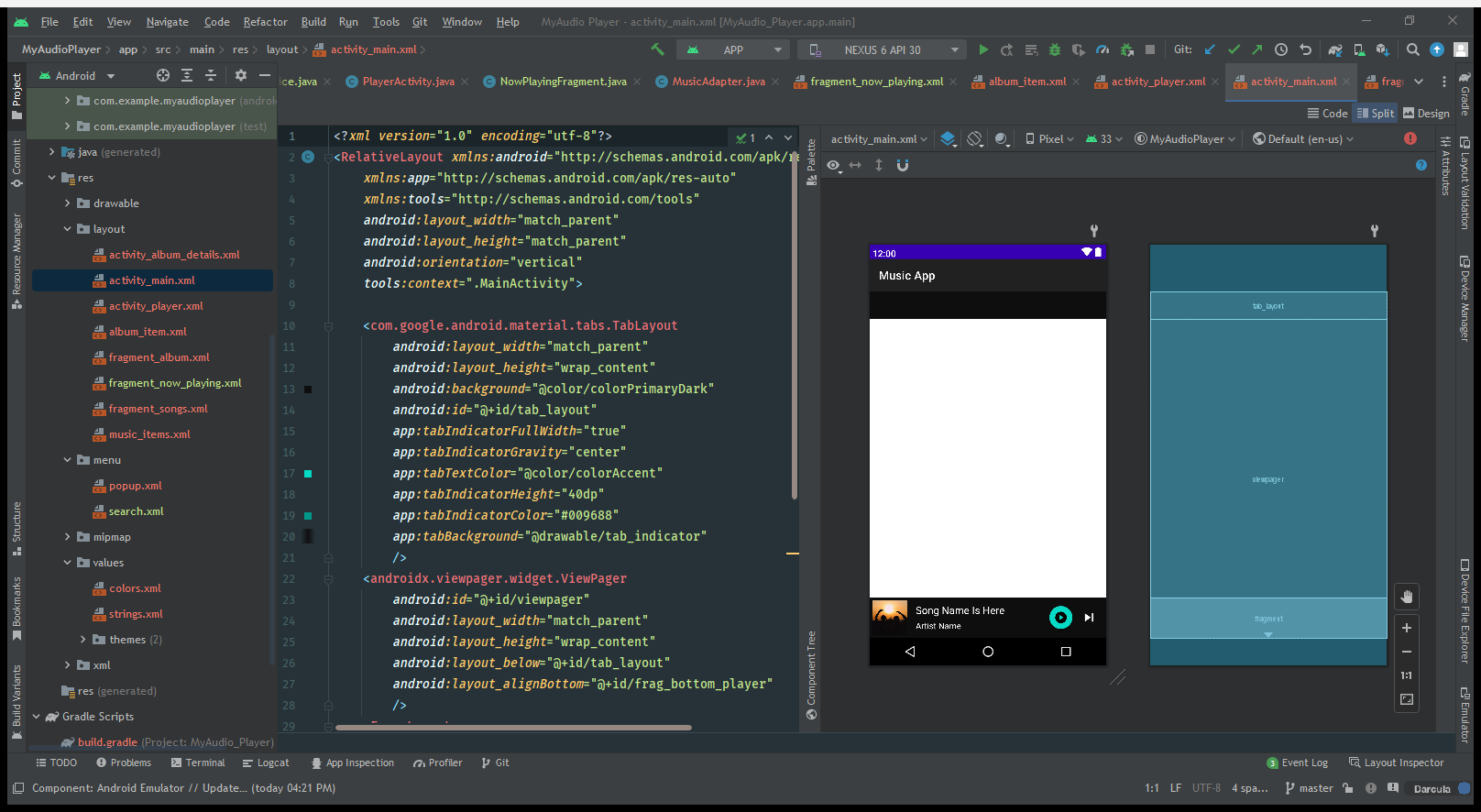
Hình : Layout fragment\_now\_play



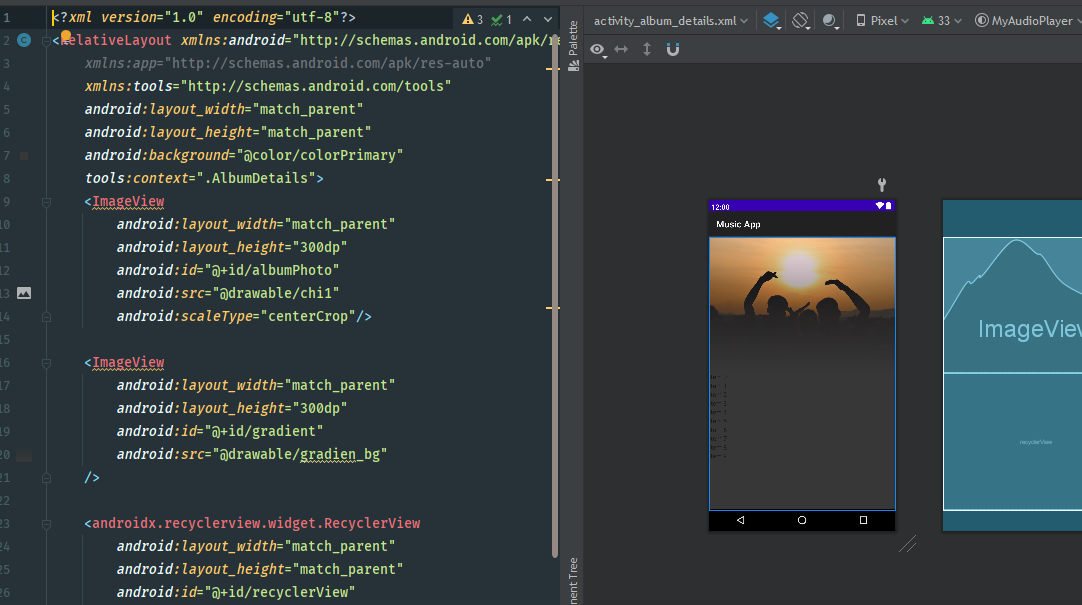
Hình : Layout album\_item



Hình : Activity\_player



Hình : Activity\_main



Hình : Layout Activity\_album\_detail

## 3.2. Thiết lập điều hướng cho Music App

### 3.2.1. Thiết lập SeekBar

Một **SeekBar** là một lớp mở rộng từ **ProgressBar** nó có thêm một cái cần gạt. Người dùng có thể chạm vào cần gạt và kéo sang trái hoặc phải để thiết lập giá trị của tiến trình (progress) hoặc sử dụng các phím mũi tên để di chuyển cần gạt. Có thể chạm vào thanh bên trái hoặc bên phải cần gạt để di chuyển nó, mặc dù không được khuyến khích.

**Một số phương thức thường dùng trong SeekBar**

Để nhận biết tiến trình đang thay đổi trong **SeekBar**chúng ta cần thực thi ba phương thức trừu tượng:

**1. public void onProgressChanged (SeekBar seekBar, int progresValue, boolean fromUser)**

 Phương thức này sẽ được gọi khi người dùng thay đổi tiến trình của thanh tìm kiếm.

**2. public void onStartTrackingTouch(SeekBar seekBar)**

 Phương thức nà được thực hiện khi người dùng bắt đầu thay đổi  tiến trình của thanh tìm kiếm

**3. public void onStopTrackingTouch(SeekBar seekBar)**

 Phương thức này được thực hiện khi người sử dụng kết thúc thay đổi tiến trình của tìm kiếm bar.

**4. getMax() - Trả về giá trị lớn nhất của SeekBar**

Chúng ta có thể lấy giá trị lớn nhất của **SeekBar**, phương thức này trả về một số nguyên. Ví dụ lấy giá trị lớn nhất của **SeekBar.**

se.setOnSeekBarChangeListener(*new* SeekBar.OnSeekBarChangeListener() {  
 @Override  
 *public void* onProgressChanged(SeekBar seekBar, *int* progress, *boolean* fromUser) {  
 *//change progress  
 if* (musicService != *null* && fromUser){  
 musicService.seekTo(progress \* 1000);  
 }  
 }

### 3.2.2. Điều khiển shuffle và repeat

shuffleBtn.setOnClickListener(v -> {  
 *if* (*shuffleBoolean*){  
 *shuffleBoolean* = *false*;  
 shuffleBtn.setImageResource(R.drawable.*ic\_shuffle\_off*);  
 }*else*{  
 *shuffleBoolean* = *true*;  
 shuffleBtn.setImageResource(R.drawable.*ic\_baseline\_shuffle\_on*);  
 }  
});  
repeatBtn.setOnClickListener(*new* View.OnClickListener() {  
 @Override  
 *public void* onClick(View view) {  
 *if* (*repeatBoolean*){  
 *repeatBoolean* = *false*;  
 repeatBtn.setImageResource(R.drawable.*ic\_baseline\_repeat\_of*);  
 }*else*{  
 *repeatBoolean* = *true*;  
 repeatBtn.setImageResource(R.drawable.*ic\_baseline\_repeat\_on*);  
 }  
 }  
});

### 3.2.3. Điều khiển prevButton

*public void* prevBtnClicked() {  
 *if* (musicService.isPlaying()){  
 musicService.stop();  
 musicService.release();  
 *if* (*shuffleBoolean* && !*repeatBoolean*){  
 position = getRandom(*listSongs*.size() - 1);  
 }*else if* (!*shuffleBoolean* && !*repeatBoolean*) {  
 position = ((position - 1) < 0 ? (*listSongs*.size() - 1) : (position -1));  
 }  
 *uri* = Uri.*parse*(*listSongs*.get(position).getPath());  
 *// mediaPlayer = MediaPlayer.create(getApplicationContext(), uri);* musicService.createMediaPlayer(position);  
 metaData(*uri*);  
 song\_name.setText(*listSongs*.get(position).getTitle());  
 artist\_name.setText(*listSongs*.get(position).getArtist());  
 seekBar.setMax(musicService.getDuration() / 1000);

}

*private void* prevThreadBtn() {  
 prevThread = *new* Thread(){  
 @Override  
 *public void* run(){  
 *super*.run();  
 prevBtn.setOnClickListener(*new* View.OnClickListener() {  
 @Override  
 *public void* onClick(View view) {  
 prevBtnClicked();  
 }  
 });  
 }  
 };  
 prevThread.start();  
}

## 3.2.4. Điều khiển nextButton

*public void* nextThreadBtn() {  
 nextThread = *new* Thread(){  
 @Override  
 *public void* run(){  
 *super*.run();  
 nextBtn.setOnClickListener(view -> nextBtnClicked());  
 }  
 };  
 nextThread.start();  
}

*private void* nextBtnClicked() {  
 *if* (musicService.isPlaying()){  
 musicService.stop();  
 musicService.release();  
 *//change when btnClick with shuffle and repeat  
 if* (*shuffleBoolean* && !*repeatBoolean*){  
 position = getRandom(*listSongs*.size() - 1);  
 }*else if* (!*shuffleBoolean* && !*repeatBoolean*) {  
 position = ((position + 1) % *listSongs*.size());  
 }  
 *//else position will be position  
 uri* = Uri.*parse*(*listSongs*.get(position).getPath());  
 *//musicService = MediaPlayer.create(getApplicationContext(), uri);* musicService.createMediaPlayer(position);  
 metaData(*uri*);  
 song\_name.setText(*listSongs*.get(position).getTitle());  
 artist\_name.setText(*listSongs*.get(position).getArtist());  
 seekBar.setMax(musicService.getDuration() / 1000);  
 PlayerActivity.*this*.runOnUiThread(*new* Runnable() {  
 @Override  
 *public void* run() {  
 *if* (musicService != *null*){  
 *int* mCurrenPosition = musicService.getCurrentPosition() /1000;  
 seekBar.setProgress(mCurrenPosition);  
  
 }  
 handler.postDelayed(*this*,1000);  
 }  
 });

## 3.3. Notification music app

### 3.3.1. Tổng quan

Notification là một thông điệp mà Android hiển thị bên ngoài giao diện ứng dụng của bạn để cung cấp cho người dùng lời nhắc, tin nhắn, hay bất kỳ thông tin gì từ ứng dụng của bạn. Người dùng có thể nhấn vào notification để mở ứng dụng, hoặc thực hiện một hành động trực tiếp trên notification như gửi tin nhắn.

Để bắt đầu, ta cần đặt nội dung và kênh của thông báo bằng cách sử dụng một NotificationCompat. Builderđối tượng. Ví dụ sau đây cho thấy cách tạo thông báo với nội dung sau:

* Một biểu tượng nhỏ, được đặt bởi setSmallIcon(). Đây là nội dung duy nhất mà người dùng có thể nhìn thấy được yêu cầu.
* Một tiêu đề, được đặt bởi setContentTitle().
* Văn bản nội dung, được đặt bởi setContentText().
* Mức độ ưu tiên của thông báo, được đặt bởi setPriority().
* Mức độ ưu tiên xác định mức độ xâm nhập của thông báo trên Android 7.1 trở xuống. (Đối với Android 8.0 trở lên, thay vào đó, ta phải đặt tầm quan trọng của kênh—được hiển thị trong phần tiếp theo.

Nếu muốn thông báo của mình dài hơn, ta có thể bật thông báo có thể mở rộng bằng cách thêm mẫu kiểu có phần mở rộng setStyle(). Ví dụ: đoạn mã sau tạo vùng văn bản lớn hơn:

var builder = NotificationCompat.Builder(this, CHANNEL\_ID)

.setSmallIcon(R.drawable.notification\_icon)

.setContentTitle("My notification")

.setContentText("Much longer text that cannot fit one line...")

.setStyle(NotificationCompat.BigTextStyle()

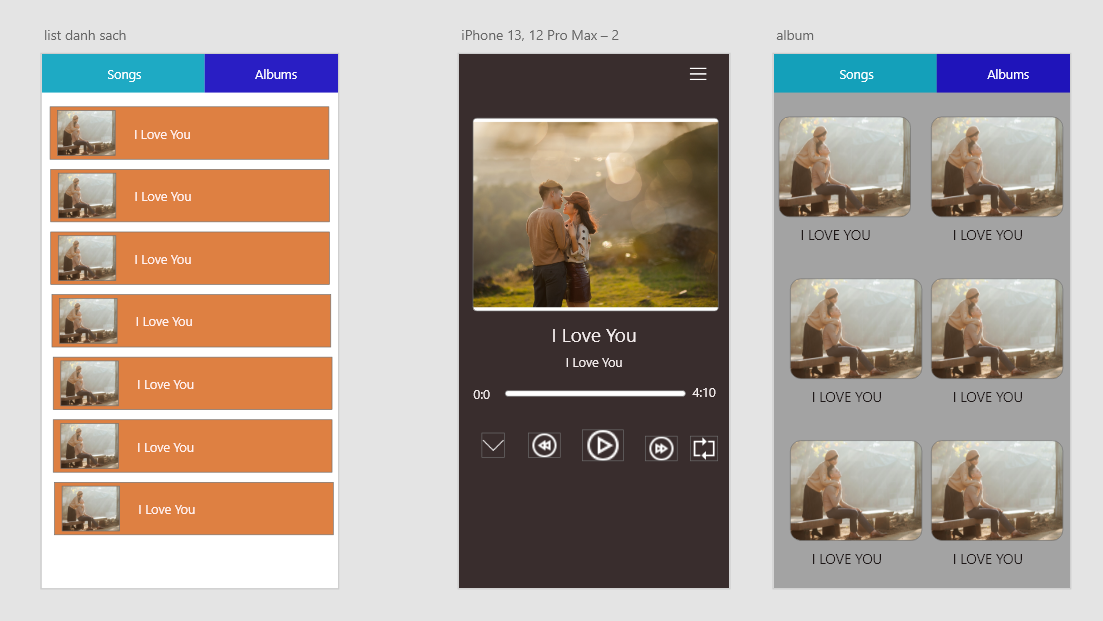
.bigText("Much longer text that cannot fit one line..."))

.setPriority(NotificationCompat.PRIORITY\_DEFAULT)

### 3.3.1. Tạo Notification

*private void* createNotificationChannel() {  
 *if* (Build.VERSION.*SDK\_INT* >= Build.VERSION\_CODES.*O*){  
 NotificationChannel channel1 =  
 *new* NotificationChannel(*CHANNEL\_ID\_1*, "Channel(1)", NotificationManager.*IMPORTANCE\_HIGH*);  
 channel1.setDescription("Channel 1 Desc...");  
  
 NotificationChannel channel2 =  
 *new* NotificationChannel(*CHANNEL\_ID\_2*, "Channel(2)", NotificationManager.*IMPORTANCE\_HIGH*);  
 channel2.setDescription("Channel 2 Desc...");  
  
 NotificationManager notificationManager =  
 getSystemService(NotificationManager.*class*);  
 notificationManager.createNotificationChannel(channel1);  
 notificationManager.createNotificationChannel(channel2);

## 4. Phác thảo giao diện



Hình : Music App Design

# LỜI KẾT

Học phần lập trình cho thiết bị di động là một môn học có rất nhiều kiến thức để học. Các kiến thức từ nền tảng tới nâng cao. Môn học có lượng kiến thức rộng lớn. Trong thời gian quá trình theo học môn học này thì chúng em thấy được rằng để lập trình tốt và thành thạo để tự thiết kế cho riêng mình một ứng dụng đẹp và bắt mắt là chưa đủ, cần phải rèn luyện thêm và tìm hiểu thêm để trang trải cho chúng em một kến thức nền vững chắc. Đặc biệt, học phần này chúng em có nền tảng lập trình android và cũng là tiền đề cho sau này tiếp cận các công nghệ, các famework hỗ trợ tốt hơn trong việc lập trình được các doanh nghiệp sử dụng.

Sản phẩm của nhóm chúng em chưa được hoàn thiện, và có một số lỗi thiếu sót và giao diện còn khá đơn giản. Qua môn học này, chúng em đã nắm vững được một lượng kiến thức cơ bản để dựng nên một app, một ứng dụng trên thiết bị android. Do vậy cần phải mỗi bản thân trong nhóm cần rèn luyện và thực hành nhiều hơn để sau này có thể phát triển hệ thống, hạn chế được các lỗi trong app.

Chúng em chân thành cám ơn thầy Trần Xuân Hào đã chia sẻ những kinh nghiệm quý báu để được hoàn thiện đồ án này.

# TÀI LIỆU THAM KHẢO

[1] https://developer.android.com/guide/components/services

[2] <https://www.youtube.com/watch?v=1D1Jo1sLBMo&t=2406s>

[3] <https://www.youtube.com/watch?v=oipB6dUgoSg>

[4] B. q. v. X. b. Đ. học, Lập trình android cơ bản

[5] T. t. đ. t. c. n. K. Phạm, "Trung tâm đào tạo công nghệ Khoa Phạm," [Online]. https://www.youtube.com/watch?v=P60kcSaeFmg&list=PL5uqQAwS\_KDjAgLGiaCakwJV1f4vRnTLS.