**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP THỰC PHẨM THÀNH PHỐ HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

**🙢🕮🙠**



**BÁO CÁO ĐỒ ÁN**

**MÔN HỌC: LẬP TRÌNH DI ĐỘNG**

**ĐỀ TÀI: APP NGHE NHẠC OFFLINE**

**GVHD: Ths. VŨ ĐÌNH ÁI**

**Sinh viên thực hiện:**

Lê Thị Phương Mai 2001160278

Võ Thị Ngọc Tuyền 2001160397

Nguyễn Trần Thúy Vy 2001150165

THÁNG 12 NĂM 2019

**MỤC LỤC**

[**LỜI MỞ ĐẦU** 2](#_Toc25433844)

[**1** **Tổng quan về Android** 3](#_Toc25433845)

[1.1 Giới thiệu hệ điều hành Android 3](#_Toc25433846)

[1.2 Ưu điểm và khuyết điểm 5](#_Toc25433847)

[1.3 Giới thiệu về Android Studio và Android SDK 9](#_Toc25433848)

[**2** **Triển khai đề tài** 10](#_Toc25433849)

[2.1 Về app nghe nhạc 10](#_Toc25433850)

[2.2 Mục tiêu đề tài 11](#_Toc25433851)

[2.3 Cơ sở lý thuyết 11](#_Toc25433852)

[2.3.1 Intent 11](#_Toc25433853)

[2.3.2 Dialog 12](#_Toc25433854)

[2.3.3 Permissions (quyền): 13](#_Toc25433855)

[2.3.4 ListView 14](#_Toc25433856)

[2.3.5 Media player 15](#_Toc25433857)

[**3** **Thiết kế và mô tả giao diện** 18](#_Toc25433858)

[3.1 Màn hình ảnh chờ 18](#_Toc25433859)

[3.2 Chức năng hỏi quyền truy cập 19](#_Toc25433860)

[3.3 Chứ năng đọc toàn bộ file mp3 trong bộ nhớ trong 20](#_Toc25433861)

[3.4 Dialog xuất hiện khi người dùng bấm nút back trên điện thoại 21](#_Toc25433862)

[3.5 Khi người dùng ấn vào 1 trong những bài hát ở listview thì sẽ chuyển qua layout khác và chơi bản nhạc đó. 22](#_Toc25433863)

[3.6 Button phát ngẫu nhiên (Shuffle) 23](#_Toc25433864)

[3.7 Button phay/pause, previous, next. 24](#_Toc25433865)

[3.8 Hẹn giờ khi đang phát nhạc 25](#_Toc25433866)

[3.9 Xử lý sự kiện khi ấn nút back về giao diện listView hoặc người dùng ấn nút back trên điện thoại…………………………………………………………………………….……………………………………………………………….26](#_Toc25433867)

**Tổng kết**………………………………………………………………………….……………………………………………………………….27

**Tài liệu tham khảo**……………………………………………………….…………………………………………………………………28

LỜI MỞ ĐẦU

Trong những năm gần đây, nền tảng công nghệ thông tin phát triển mạnh mẽ cùng với đó là sự ra đời của các thiết bị di động như máy tính bảng, điện thoại, smartphone… ngày càng được ưa chuộng và sử dụng nhiều hơn. Do đó, việc xây dựng các ứng dụng trên điện thoại di động là một ngành công nghiệp mới đầy tiềm năng và hứa hẹn nhiều sự phát triển vượt bậc của ngành khoa học kỹ thuật.

Các ứng dụng dựa trên nền tảng là hệ điều hành cho thiết bị di động ngày càng đa dạng và phong phú hơn, đáp ứng được ngày càng nhiều các yêu cầu thực tế của người dùng. Các hệ điều hành J2ME, Android, IOS, Hybrid, Web bases Mobile Application đã rất phát triển trên thị trường truyền thông di động.

Trong vài năm trở lại đây, hệ điều hành Android ra đời với sự kế thừa những ưu việt của các hệ điều hành ra đời trước và sự kết hợp của nhiều công nghệ tiên tiến nhất. Android đang là hệ điều hành di động của tương lai và được nhiều người ưa chuộng. Việc xây dựng ứng dụng cho điện thoại di động chạy hệ điều hành Android hiện nay đang rất mới và rất phát triển Việt Nam.

Ngày nay, với sự phát triển nhanh chóng của xã hội, nhu cầu giải trí thông qua điện thoại di động ngày càng phổ biến. Vì vậy, nhóm em đã chọn đề tài **“Xây dựng ứng dụng android nghe nhạc offline”** với mục đích nghiên cứu, tìm hiểu về ứng dụng nghe nhạc trên android để có thể đáp ứng nhu cầu giải trí đó, giúp cho mọi người có thể thư giãn thông qua ứng dụng một cách dễ dàng.

# **Tổng quan về Android**

## Giới thiệu hệ điều hành Android

* Hệ điều hành Android đang chiếm hơn 80% thị trường thiết bị di động hiện đại ngày nay. Nhu cầu sử dụng ứng dụng di động cũng ngày càng cao. Nhu cầu việc làm về lập trình trên hệ điều hành mobile phổ biến nhất thế giới này cũng vì thế mà tăng mạnh do tính mở và dễ tiếp cận của nó.
* Android là hệ điều hành giành cho thiết bị di động, được phát triển bởi Google và ngày càng trở nên phổ biến với các hãng điện thoại.
* Android được xây dựng trên nhân Linux và được phân phối miễn phí. Mọi ứng dụng Android điều được viết trên cùng một tập API, thế nên không có sự phân biệt giữa các ứng dụng mặc định và các ứng dụng bên thứ ba. Người dùng hoàn toàn có thể thay thế mọi ứng dụng mặc định bằng mọi ứng dụng ưa thích của mình, thậm chí ngay cả màn hình thực hiện cuộc gọi và màn hình nhà (home screen).
* Các nhà phát triển viết ứng dụng Android dựa trên ngôn ngữ Java. Sự ra mắt của Android vào ngày 5 tháng 11 năm 2007 gắn với sự thành lập của liên minh cầm tay mã nguồn mở, bao gồm 78 công ty phần cứng, phần mềm và viễn thông nhằm mục đích tạo nên một chuẩn mở cho điện thoại di động trong tương lai.
* Hiện Android là một thương hiệu của Google. Có khả năng tùy biến rất cao và có thể chạy trên nhiều thiết bị, nhiều kiến trúc vi xử lý (ARM / x86). Tính đến nay, Android đã có các phiên bản (kèm tên mã) lần lượt là:
* Android 1.5 Cupcake
* Android 1.6 Donut
* Android 2.1 Eclair
* Android 2.2 Froyo
* Android 2.3 Gingerbread
* Android 3.2 Honeycomb – Phiên bản Android đầu tiên được thiết kế cho máy tính bảng.
* Android 4.0 Ice Cream Sandwich
* Android 4.1 Jelly Bean
* Android 4.2 Jelly Bean
* Android 4.3 Jelly Bean
* Android 4.4 KitKat
* Android 5.0 Lollipop
* Android 6.0 Marshmallow
* Android 7.0 Nougat
* Về kiến trúc của hệ điều hành Android, chúng ta có thể xem qua sơ đồ sau:



* **Tầng Applications:** Là tầng chứa các ứng dụng Danh bạ, Gọi điện, Trình duyệt, Nghe nhạc,… các ứng dụng này thường mua máy về chúng ta đã có sẵn rồi.
* **Tầng Framework:** Là tầng chứa các API để làm việc với hệ điều hành như lấy thông tin danh bạ, quản lý các Activity (Activity là gì thì giờ chúng ta chưa cần quan tâm, các bài sau sẽ giải thích kĩ), quản lý địa điểm, quản lý các View (cũng chưa cần quan tâm).
* **Tầng Libraries:** Chứa các thư viện, API gần như là cốt lõi của Android, bao gồm bộ quản lý bề mặt cảm ứng (Surface Manager), OpenGL (phục vụ cho việc dựng đồ họa phức tạp),…
* **Tầng Android Runtime:** Chứa các thư viện lõi của Android và máy ảo Dalvik Virtual Machine (từ Android 4 trở lên chúng ta có thêm máy ảo ART).
* **Tầng Kernel**: Là nhân lõi của hệ điều hành, chứa các tập lệnh, driver giao tiếp giữa phần cứng và phần mềm của Android.

Trong quá trình làm việc, chúng ta sẽ gần như chỉ làm việc với tầng xanh lam (Applications và Application Framework) và xanh lá (Libraries). Chương trình Android được viết bằng ngôn ngữ Java và được máy ảo DVM / ART trong mỗi thiết bị Android biên dịch ra mã máy.

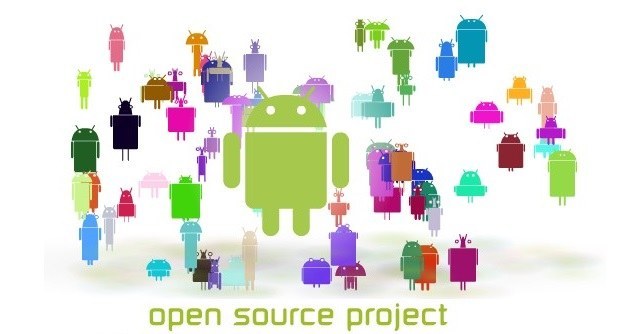
* Google công bố hầu hết các mã nguồn của Android theo bản cấp Apache.



Hình 1: Giao diện màn hình android

## Ưu điểm và khuyết điểm

* **Ưu điểm**
* Hệ điều hành mở:

Điều làm cho Android được chú ý nhất chính vì nó là mã nguồn mở. Android đã thu hút rất nhiều lập trình viên và nhà phát triển từ tất cả các tầng lớp xã hội khi Google phân phối miễn phí Android và các đoạn mã có bản quyền khác cho bất cứ ai muốn tham gia phát triển phần mềm cho Google. Vì thế, trong một thời gian ngắn, các lập trình viên cũng như các nhà phát triển có thể nhanh chóng đưa ra phiên bản Android đươc tùy chỉnh của riêng mình.

* Đa nhiệm:

Ngay từ phiên bản Android 1.0 đã có nhiều ứng dụng chạy cùng lúc(các ứng dụng hệ thống hoặc các ứng dụng từ Android Marketplace). Người dùng Android được hưởng lợi rất nhiều từ sự khác biệt này, vì họ có thể nhận thông báo, nghe nhạc và thậm chí ghi dữ liệu GPS mà không cần giữ ứng dụng luôn mở.

* Giá cả hợp lý:

Mức giá để sở hữu một thiết bị là vấn đề mà không ít người quan tâm đến. Với các thiết bị Android, trong khoảng từ 1 – 20 triệu đồng, bạn sẽ có muôn vàn những lựa chọn, từ những thiết bị bình dân vừa túi tiền đến những thiết bị cao cấp sở hữu vô số những tính năng hữu ích. Trên thị trường hiện nay có rất nhiều hãng điện tử như Samsung, LG, Sony, Lenovo, Oppo, Huawei … tung ra hàng loạt các sản phẩm đủ mọi kích cỡ, đủ mọi mức giá vì vậy người dùng sẽ có nhiều sự để lựa chọn hơn so với các thiết bị sử dụng hệ điều hành khác.

* Kho ứng dụng Google Play store đồ sộ:

Trước đây, khi mới ra mắt kho ứng dụng Google Play chỉ vỏn vẹn có một số ứng dụng. Nhưng đến nay, kho ứng dụng này đã đạt đến hơn 650.000 ứng dụng, trong đó có rất nhiều ứng dụng miễn phí mà vô cùng tiện dụng. Đây cũng là lý do để giải thích cho sự phát triển chóng mặt của hệ điều hành Android trong thế giới di động. Nếu bạn sử dụng một thiết bị Android, chắc hẳn bạn sẽ không phải bận tâm vì “thiếu ứng dụng để cài” mà cái bạn sẽ bận tâm đó là “không đủ bộ nhớ để cài”.



* Dễ sử dụng

Có lẽ hệ điều hành Android chưa chắc đã dễ sử dụng hơn các hệ điều hành khác (điều này tùy thuộc vào ý kiến chủ quan của mỗi người), nhưng vì nó đã quá phổ biến trên các thiết bị di động, nên chắc hẳn bạn không cảm thấy lạ lẫm gì với nó nữa. Bạn sẽ có 2 môi trường để làm việc trên các thiết bị Android, đó là màn hình chính (HomeScreen) và menu chính. Màn hình chủ giống như một ngôi nhà của bạn, gồm có các bức tường xung quanh mà bạn có thể thõa sức trang trí, treo đồng hồ, dán lịch, đặt máy nghe nhạc … Menu chính thì giống như trong hộc bàn của bạn, có đầy đủ những thứ bạn cần như: máy nhắn tin, máy nghe nhạc, điện thoại, máy chụp hình …

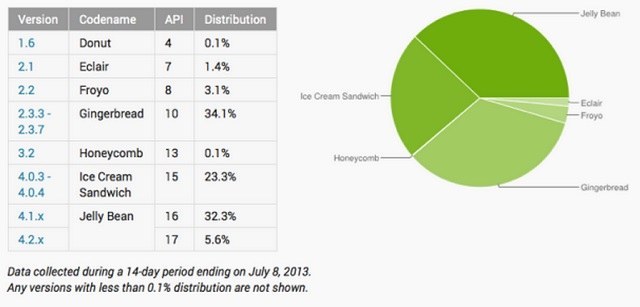


* **Khuyết điểm**
* Khả năng dọn rác kém

Sau một thời gian sử dụng thiết bị Android, các dữ liệu của ứng dụng đã được lưu trong bộ nhớ đệm, mà máy không có quá trình tự động để dọn dẹp chúng đi. Những dữ liệu này nếu không được dọn dẹp sẽ vẫn nằm ở đó và chiếm bộ nhớ của máy, đặc biệt là bộ nhớ RAM, vì vậy nếu sử dụng các thiết bị có bộ nhớ RAM thấp bạn sẽ thường xuyên gặp phải tình trạng giật, lag khi đang sử dụng. Tuy nhiên, nhược điểm này cũng có cách để khắc phục, nếu máy không tự “quét nhà” thì bạn sẽ phải “quét” bằng cách cài đặt các ứng dụng dọn dẹp như: History Eraser, Android System Cleaner, Clean MasterNhưng vấn đề giật, lag chỉ gặp ở những máy có cấu hình thấp, đối với những máy tầm trung hoặc cao cấp hiện nay thì hiện tượng đó rất hiếm.

* Quá nhiều phiên bản dẫn đến việc phân mảnh

“Đem con bỏ chợ” là những gì mà đa phần người dùng than phiền Google khi sử dụng hệ điều hành Android phiên bản cũ và không được hỗ trợ nâng cấp lên phiên bản mới. Do tốc độ phát triển nhanh, kèm theo việc nâng cấp phiên bản thường xuyên dẫn đến việc có rất nhiều phiên bản Android trên thị trường. Thêm vào đó là rất nhiều hãng sử dụng hệ điều hành Android mà không thể nào hỗ trợ nâng cấp hết cho tất cả các thiết bị. Một phần cũng dễ hiểu vì họ luôn chạy theo và tập trung phát triển cho cái mới nên việc bỏ quên các thiết bị cũ là không hề tránh khỏi.



* Bảo mật

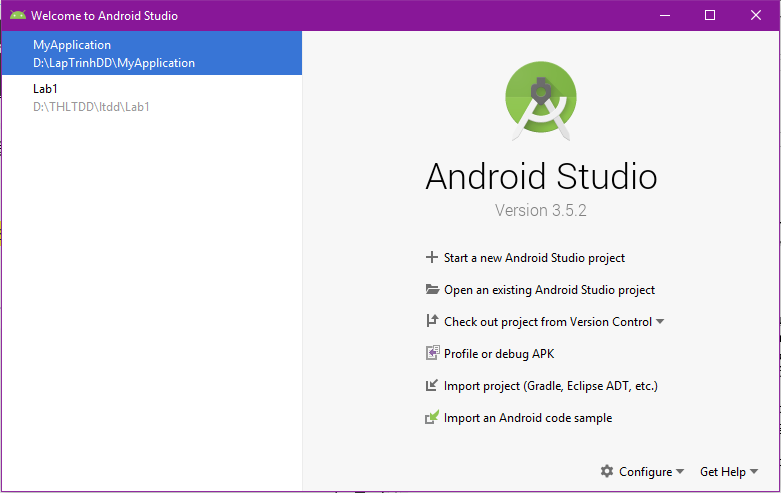
Do đặc điểm là mã nguồn mở, nên đây là điều kiện để những người có ý đồ xấu tìm hiểu và khai thác lỗ hổng bảo mật trên thiết bị Android. Ví dụ: trước đây có một ứng dụng tên là iCalendar là một ứng dụng lịch báo thông thường với những icon và hiệu ứng lạ mắt, nhưng ẩn bên dưới là những đoạn mã độc dùng để “móc túi” người dùng, những đoạn mã độc này sẽ tự động gửi tin nhắn về tổng đài đã được định sẵn khi chạm 5 lần trên màn hình mà người dùng không hề hay biết. Sau khi vụ việc được phát hiện thì ứng dụng đó đã bị gỡ bỏ khỏi Google Play. Dần dần qua các phiên bản, Google cũng đã cải thiện khả năng bảo mật, nhưng “liệu có còn tồn tại những lỗ hổng nữa không?” thì vẫn chưa thể trả lời chính xác được.

* Kho ứng dụng nhiều dẫn đến việc khó lựa chọn

Thật không dễ dàng để tìm kiểm một ứng dụng phù hợp trong kho lưu trữ khổng lồ Google Play Store. Khi bạn gõ một từ khóa tìm kiếm trên Play Store tiếp đó sẽ xuất hiện một danh sách thật dài các ứng dụng có cùng chức năng. Vì vậy việc lựa chọn không hề dễ dàng.Tuy nhiên, khuyết điểm này cũng có cách để khắc phục. Nếu bạn có thời gian, bạn có thể tự chọn lọc ứng dụng mình thích vì việc cài đặt và gỡ ra vô cùng dễ dàng, bạn chỉ việc kéo ứng dụng vào sọt rác là có thể gỡ nó ra nhanh chóng. Nếu không có thời gian bạn có thể tham khảo trên các diễn đàn công nghệ, có rất nhiều đề xuất về các ứng dụng hữu ích, chất lượng đã được “kiểm duyệt” bởi một số người dùng.

## Giới thiệu về Android Studio và Android SDK

* Tháng 5 năm 2013, Google công bố Android Studio, một môi trường phát triển ứng dụng tích hợp (IDE) dành riêng cho Android, mã nguồn mở, dựa trên IDE Java IntelliJ của hãng JetBrains (đối thủ với Eclipse và Netbeans, vốn khá quen thuộc với dân lập trình Java).
* Android Studio chạy trên Windows, Mac và Linux, nhằm thay thế cho Eclipse Android Development Tool (ADT) vốn được sử dụng làm IDE chính trong các năm trước đó.
* **Một số tính năng nổi bật:**
  + Bộ công cụ build ứng dụng dựa trên Gradle (thay vì Maven).
  + Chức năng dò và sửa lỗi nhanh, hướng Android.
  + Công cụ chỉnh sửa màn hình dạng kéo thả tiện lợi.
  + Các wizard tích hợp nhằm giúp lập trình viên tạo ứng dụng từ mẫu có sẵn.
* Tích hợp Google Cloud Platform, dễ dàng tích hợp với Google Cloud Messaging và App Engine của Google.
* Và đi kèm với Android Studio là Android SDK. Nếu Android Studio là trình soạn thảo code (IDE) thì Android SDK là bộ tổng hợp các công cụ để build app, các bản mẫu máy ảo Android (sử dụng để test app) cần thiết để làm ra một ứng dụng Android hoàn chỉnh.
* Giao diện màn hình đầu của Android Studio:

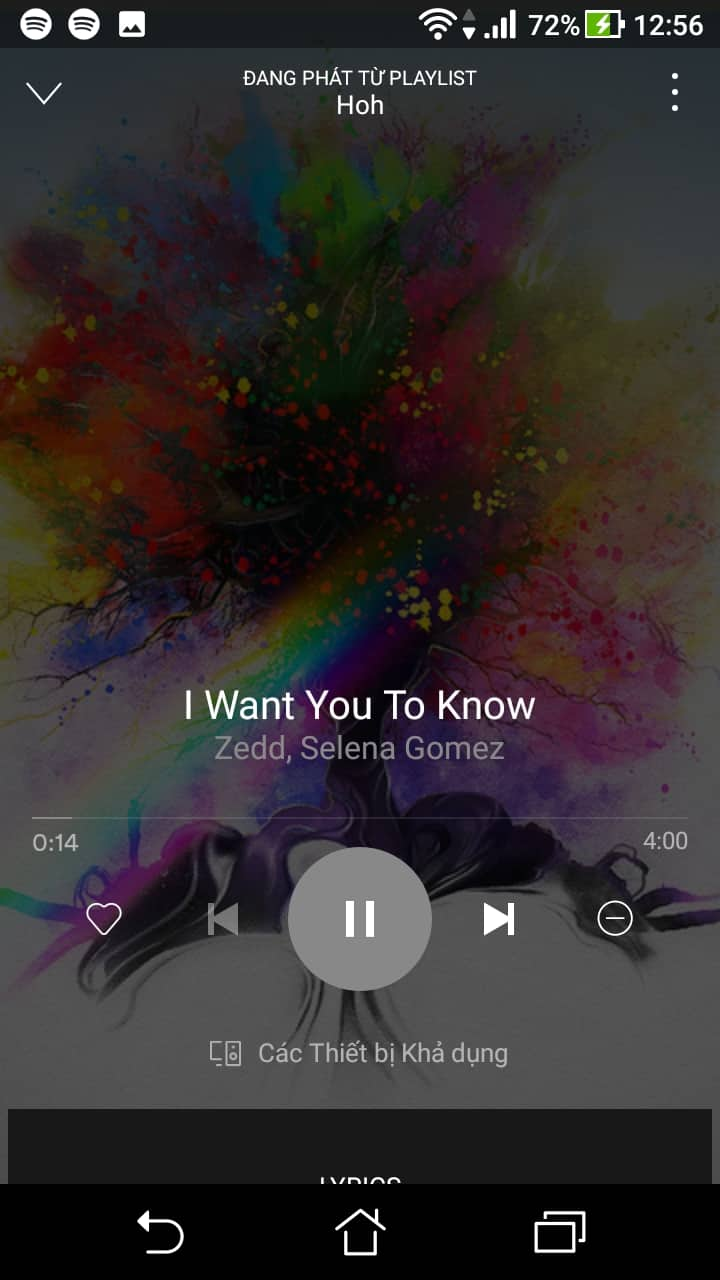


Hình 2: Giao diện màn hình Android Studio

# **Triển khai đề tài**

## Về app nghe nhạc

* Âm nhạc là một bộ môn nghệ thuật dùng âm thanh để diễn đạt. Các yếu tố chính của nó là cao độ (điều chỉnh giai điệu), nhịp điệu (và các khái niệm liên quan của nó: tempo, tốc độ), âm điệu, và những phẩm chất âm thanh của âm sắc và kết cấu bản nhạc. Là âm thanh thanh nhạc hoặc công cụ âm thanh (hoặc cả hai) kết hợp theo cách như vậy để tạo ra vẻ đẹp của hình thức, sự hài hòa và biểu hiện cảm xúc.
* Kể từ khi ra đời đến nay, nhạc trực tuyến đã phát triển một cách nhanh chóng và là một trong những lựa chọn hàng đầu cho việc nghe nhạc. Rất nhiều dịch vụ nhạc trực tuyến cho phép bạn nghe nhạc miễn phí , điều này giúp ích cho việc ngăn chặn vi phạm bản quyền và sự cho phép từ các nghệ sĩ.
* Dưới đây là một số ứng dụng nghe nhạc hiện nay:



Hình 3: Spotify Hình 4: Zing MP3

## Mục tiêu đề tài

* Ứng dụng với chức năng chính: nghe nhạc, tủy chỉnh bài hát đang chạy ( next, previous, pause), tùy chỉnh âm lượng, hẹn giờ.
* Ứng dụng đơn giản, dễ sử dụng.
* Chạy được nhiều máy trên hệ điều hành android.
* Không bị giật lag, chiếm ít tài nguyên.

## Cơ sở lý thuyết

### Intent

* Là những tin nhắn không đồng bộ cho phép các thành phần ứng dụng yêu cầu chức năng từ các thành phần Android khác.
* **Intents** cho phép bạn tương tác với các thành phần từ các ứng dụng như nhau cũng như với các thành phần đóng gói trong một ứng dụng khác.

**VD**:một Activity có thể gọi một Activity bên ngoài để chụp ảnh.

* **Intent** là đối tượng của class **android.content.Intent**.
* Mã của bạn có thể gửi **Intent** vào hệ thống Android với chỉ định thành phần mục tiêu gửi đến.

**VD:** thông qua phương thức ***startActivity()***, bạn có thể định nghĩa một **Intent** sử dụng để gọi chạy một **Activity** khác. Tại **Activity** mục tiêu, thông qua phương thức ***startActivity()*** bạn có thể xác định được ý định của người gửi đến để bắt đầu (start) **Activity** này.

* Một **Intent** có thể chứa dữ liệu thông qua một **Bundle**. Dữ liệu này có thể được sử dụng bởi các thành phần tiếp nhận.
* Các thuộc tính của **Intent**:
* **Action**: là hành động được thực hiện

**VD : ACTION\_VIEW, ACTION\_MAIN**

* **Data:** là dữ liệu sẽ được xử lý trong action, thường được diễn tả là một Uri (Uniform Resource Identifier).

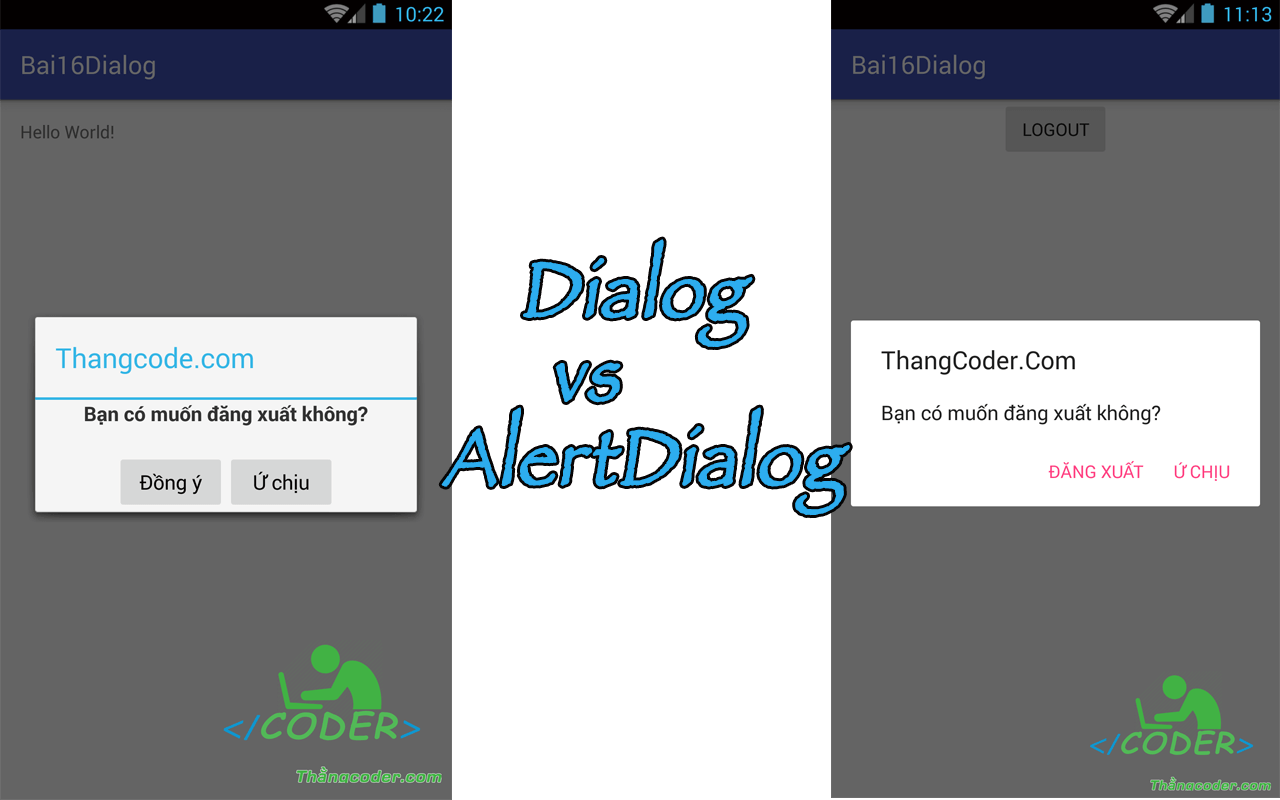
**VD**: **ACTION\_VIEW** content://contacts/people/1 - Hiển thị thông tin về người với mã danh 1.

**ACTION\_DIAL** content://contacts/people/1 - Hiển thị màn hình gọi đến người với mã danh 1.

**ACTION\_DIAL** tel:123 - Hiển thị màn hình gọi với số gọi là 123.

* Ngoài ra còn có một số thuộc tính ta có thể thêm vào **Intent:**
* **Category:** bổ sung thêm thông tin cho action của Intent. VD: **CATEGORY\_LAUNCHER** thông báo sẽ thêm vào Launcher như là một ứng dụng top-level.
* **Type**: chỉ rõ kiểu của data.
* **Component**: chỉ rõ thành phần sẽ nhận và xử lý **Intent**. Khi thuộc tính này được xác định thì các thuộc tính khác sẽ trở thành thuộc tính phụ.
* **Extras**: mang theo đối tượng **Bundle** chứa các giá trị bổ sung.
* **Intent** có thể sử dụng để :
* Bắt đầu một Activity.
* Bắt đầu một Activity con .
* Bắt đầu một dịch vụ (Service)
* **Các loại Intent phổ biến:**
* **Explicit Intents (Intent tường minh)**: **Intent** đã được xác định thuộc tính component, nghĩa là đã chỉ rõ thành phần sẽ nhận và xử lý intent. Thông thường **Intent** dạng này sẽ không bổ sung thêm các thuộc tính khác như action, data. **Explicit Intent** thương được sử dụng để khởi chạy các activity trong cùng 1 ứng dụng.
* **Implicit Intents (Intent không tường minh)**: **Intent** không chỉ rõ component xử lý, thay vào đó nó bổ sung thông tin trong các thuộc tính. Khi **Intent** được gửi đi, hệ thống sẽ dựa vào những thông tin này để quyết định component nào thích hợp nhất để xử lý nó.

### Dialog

Dialog dung để show lên một thông báo trên màn hình, và thông báo này sẽ che mờ màn hình, bạn cũng có thể xử lý sự kiện trên Dialog này.

Dialog chính trong Android đó là **Dialog** và **AlertDialog**:

* **Dialog**:

Đây là loại Dialog mà bạn phải tạo giao diện cho nó bằng xml chứ mặc định nó không có giao diện ngoài cái title, hay nói cách khác bạn phải custom nó từ đầu tới cuối, không giống AlertDialog bạn sẽ thêm được title, message, button…

* + - **dialog = new Dialog(MainActivity.this)**: đây là khởi tạo dialog
    - **dialog.setTitle(“Thangcode.com”):** ở đây bạn sẽ đặt tên tiêu đề cho Dialog nhé
    - **dialog.setContentView(R.layout.dialog):** đây chính là dòng quan trọng và quyết định đây, chắc bạn không thể nào quên phương thức setContentView() mà bất kì activity nào cũng có để tạo giao diện đó chứ.Ở đây có nghĩa là chúng ta sẽ set giao diện vào dialog, giao diện này bạn đã định nghĩa ở file layout dialog.xml trước đó rồi nhé.
    - **dialog.show():** tất cả đã xong thì bây giờ bạn chỉ cần gọi phương thức show() để dialog hiển thị lên là được.
* **AlertDialog**:

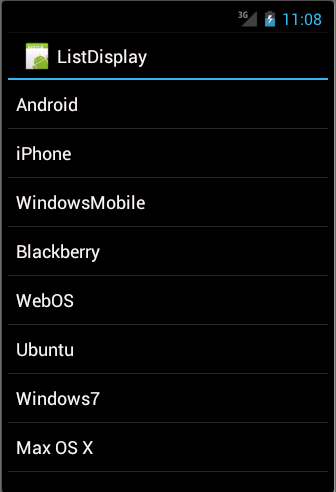
**AlertDialog** là một kiểu Dialog hay nó cách khác nó là lớp con kế thừa từ Dialog ra, nhưng có thể hiển thị nhiều các Button bên dưới mà chúng ta không cần phải định nghĩa layout cho nó.Ngoài ra bạn muốn hiển thị tiêu đề, nội dung bên trong thì **AlertDialog** đều hỗ trợ bạn nhanh chóng.

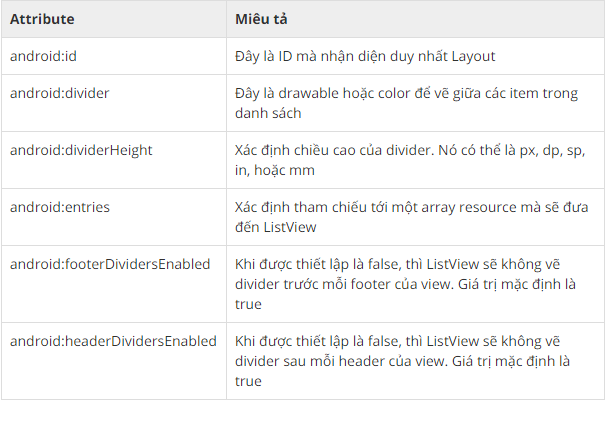
* + - **AlertDialog.Builder builder = new AlertDialog.Builder(this):** đây là hàm khởi tạo một AlertDialog.Builder, thằng này có tác dụng vẽ ra giao diện cho AlertDialog, mặc định hàm trên nó sẽ sử dụng theme dialog mặc định khi show và trong video bên dưới mình sẽ nói rõ hơn chỗ này, giúp bạn tuỳ chỉnh theme cho đẹp hơn.
    - **builder.setTitle():** ở đây là set tiêu đề cho AlertDialog, cái này tương tự như Dialog không có gì khác cả.
    - **builder.setMessage():** đây là set nội dung bên trong cho Alert, thằng Dialog thì không có phương thức này nhé.
    - **builder.setCancelable():** đây là thuộc tính nếu set false thì khi show dialog lên người dùng click ra bên ngoài dialog thì nó vẫn không bị mất, nếu set true thì sẽ mất khi click vào bất kì đâu ngoài dialog.

### Permissions (quyền):

* Permission được dùng để giới hạn quyền truy xuất vào chương trình mà dữ liệu trên thiết bị. Những giới hạn này được áp đặt để bảo vệ dữ liệu nhạy cảm và các chương trình có thể gây lỗi hoặc cố ý dùng sai của người dùng có kinh nghiệm.
* Mỗi quyền được xác định bởi 1 chuỗi duy nhất. Thường mỗi chuỗi xác định duy nhất 1 quyền. Dưới đây là một số quyền được mô tả bởi sẳn Android.
  + - * + android.permission.READ\_EXTERNAL\_STORAGE
        + android.permission.READ\_OWNER\_DATA
        + android.permission.SET\_WALLPAPER
        + android.permission.DEVICE\_POWER
* Mỗi tính năng của ứng dụng có thể được bảo vệ bằng một quyền.
* Nếu ứng dụng cần truy xuất vào một tính năng được bảo vệ bằng quyền truy xuất, nó phải mô tả quyền cần để truy xuất trong tập tin manifest bằng thẻ <uses-permission>.
* Khi ứng dụng được cài đặt vào thiết bị, trình cài đặt sẽ kiểm tra xem có thể cấp quyền được yêu cầu bằng cách kiểm tra chứng chỉ xác thực của ứng dụng, nếu không cấp quyền được, trình cài đặt có thể yêu cầu xác nhận của người dùng.
* Nếu đã được ủy quyền, ứng dụng có thể truy xuất vào các tính năng được bảo vệ, nếu không được ủy quyền, khi ứng dụng sẽ không truy cập được các tính năng được bảo vệ mà không có bất kỳ thông báo nào với người dùng.
* Ứng dụng có thể bảo vệ các component của nó (activities, services, broadcast receivers và content providers) bằng quyền.
* Nó có thể sử dụng bất kỳ quyền nào đã được mô tả sẳn bởi Android hoặc được khai báo trong ứng dụng khác.
* Library:
* Mỗi ứng dụng sẽ cần liên kết với những thư viện có sẳn của Android, chứa các gói cơ bản cho việc biên dịch ứng dụng.
* Tuy nhiên, nếu ứng dụng sử dụng những package nằm ngoài thư viện có sẳn của Android, bạn cần phải liên yêu cần được liên kết với chúng. Thẻ <uses-library> trong tập tin manifest sẽ thực hiện công việc này.

### ListView

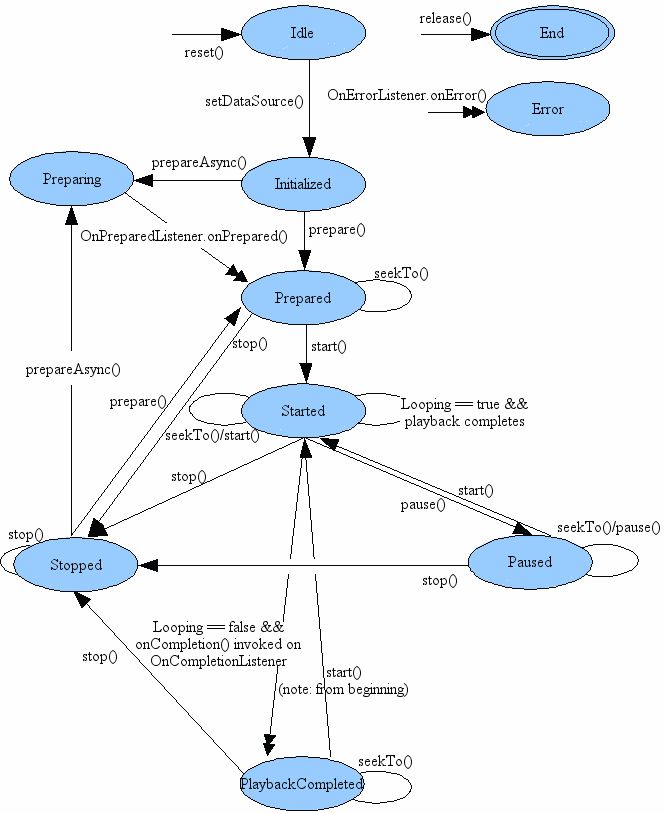
* ListView trong Android là một view mà nhóm một số item và hiển thị chúng trong danh sách theo chiều dọc có thể scroll. Các item của danh sách này được tự động chèn vào danh sách bởi sử dụng một Adapter mà kéo nội dung từ một nguồn như một mảng hoặc cơ sở dữ liệu.
* Một Adapter thực sự là cầu nối giữa các thành phần UI và nguồn dữ liệu mà điền dữ liệu vào trong thành phần UI. Adapter giữ dữ liệu và gửi dữ liệu tới Adapter View, từ đó view có thể lấy dữ liệu từ Adapter View đó và hiển thị dữ liệu trên các view khác nhau ở dạng Spinner, List View, Grid View, …
* ListViewvà GridView là các lớp con của AdapterView . Gắn kết chúng với mộtAdapter, mà thu nhận dữ liệu từ một nguồn ngoại vi và tạo một View để biểu diễn mỗi entry.
* Android cung cấp một số lớp con hữu ích của lớp Adapter để thu lấy các kiểu dữ liệu khác nhau và xây dựng các view cho một AdapterView (ví dụ ListView hoặc GridView). Các Adapter phổ biến là ArrayAdapter, Base Adapter, CursorAdapter, SimpleCursorAdapter, SpinnerAdapter và WrapperListAdapter.
* Các thuộc tính cơ bản của ListView



* ArrayAdapter trong Android
* Bạn có thể sử dụng Adapter này khi nguồn dữ liệu là một mảng. Theo mặc định, ArrayAdapter tạo một view cho mỗi item trong mảng bằng việc gọi toString() trên mỗi item và đặt nội dung vào trong một TextView. Giả sử bạn có một mảng các chuỗi và muốn hiển thị trong một ListView, bạn khởi tạo một ArrayAdapter mới sử dụng một Constructor để xác định Layout cho mỗi chuỗi và mảng chuỗi.

### Media player

* Các lớp sau được sử dụng để phát nhạc và video trong Android :
* **MediaPlayer:** Lớp này là API chính để phát âm thanh và video.
* **AudioManager:** Lớp này quản lý nguồn âm thanh và đầu ra âm thanh trên thiết bị (loa, tai nghe).
* Sơ đồ sau đây cho thấy vòng đời và trạng thái của đối tượng MediaPlayer, các hình bầu dục đại diện cho các trạng thái mà một đối tượng MediaPlayer có thể nằm trong. Các mũi tên biểu diễn các phương thức điều khiển chuyển tiếp trạng thái đối tượng MediaPlayer.
* Có hai loại mũi tên:
* Các mũi tên với một đầu mũi tên đơn đại diện cho việc gọi phương thức đồng bộ
* Các mũi tên với một đầu mũi tên kép đại diện cho việc gọi phương thức không đồng bộ



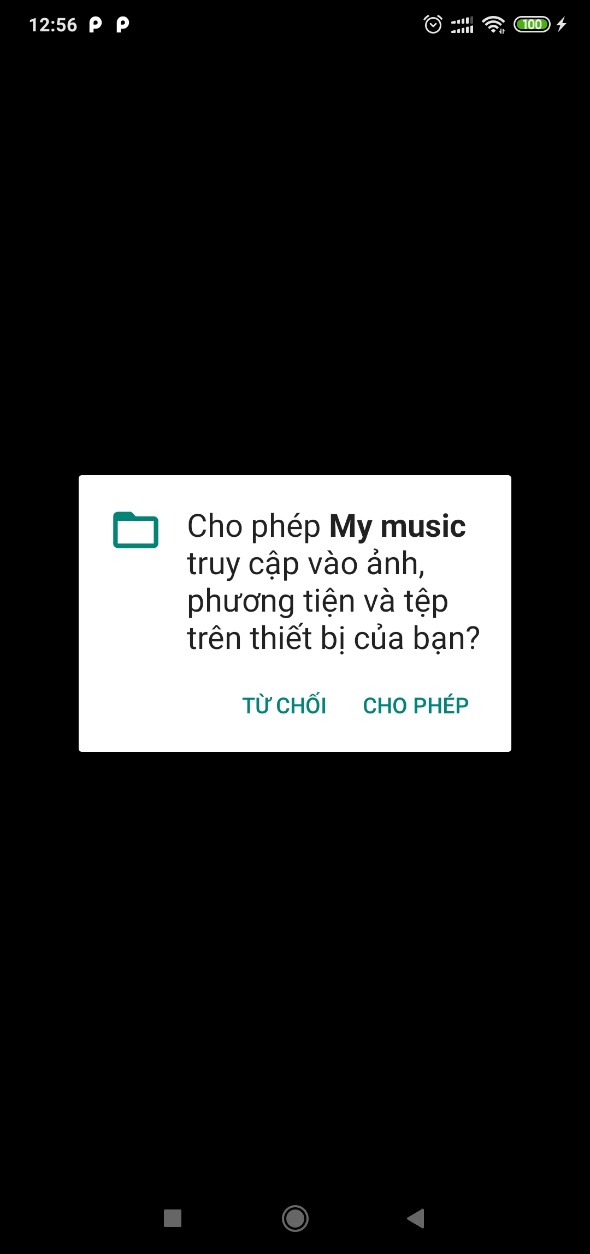
* Để bắt đầu phát media, **start()** phải được gọi. Sau khi **start()** trả về thành công, đối tượng MediaPlayer đang ở trạng thái **Started**. Phương thức **isPlaying()** có thể được gọi để kiểm tra xem đối tượng MediaPlayer có ở trạng thái **Started** hay không.
* Gọi **start()** không ảnh hưởng đến đối tượng MediaPlayer đã ở trạng thái **Started**.
* Player có thể được tạm dừng và dừng lại, và vị trí đang phát hiện tại có thể được điều chỉnh. Có thể tạm dừng thông qua **pause()**. Khi **pause()** trả về, đối tượng MediaPlayer nhập trạng thái **Paused**.
* Lưu ý rằng quá trình chuyển đổi từ trạng thái **Started** sang trạng thái **Paused** và ngược lại xảy ra không đồng bộ. Có thể mất một thời gian trước khi trạng thái được cập nhật tkhi gọi đến **isPlaying()**, và nó có thể là một vài giây trong trường hợp nội dung được truyền trực tuyến.
* Gọi **start()** để tiếp tục phát lại đối tượng MediaPlayer bị tạm dừng và vị trí phát được tiếp tục giống với vị trí đã tạm dừng. Khi **start()** trả về, đối tượng MediaPlayer bị tạm dừng quay lại trạng thái **Started**.
* Gọi **pause()** không có hiệu lực đối với đối tượng MediaPlayer đã ở trạng thái **Paused**.
* Gọi **stop()** để dừng phát và khiến MediaPlayer ở trạng thái **Started**, **Paused**, **Prepared** hoặc **PlaybackCompleted** để vào trạng thái **Stopped**.
* Khi ở trạng thái **Stopped**, không thể bắt đầu phát cho đến khi **prepare()** hoặc **prepareAsync()** được gọi để đặt đối tượng MediaPlayer về trạng thái **Started**.
* Gọi **stop()** không có hiệu lực đối với đối tượng MediaPlayer đã ở trạng thái **Stopped**.
* Vị trí phát có thể điều chỉnh bằng cách gọi **seekTo(long, int)**
* Chú ý **seekTo(long, int)** có thể được gọi khi Media Player ở những trạng thái **Prepared**, **Paused**and **PlaybackCompleted**
* Hơn nữa vị trí hiện tại có thể được lấy ra bằng phương thức **getCurrentPosition()** , phương thức hữu ích nếu bạn muốn sử dụng tiến trình phát media.

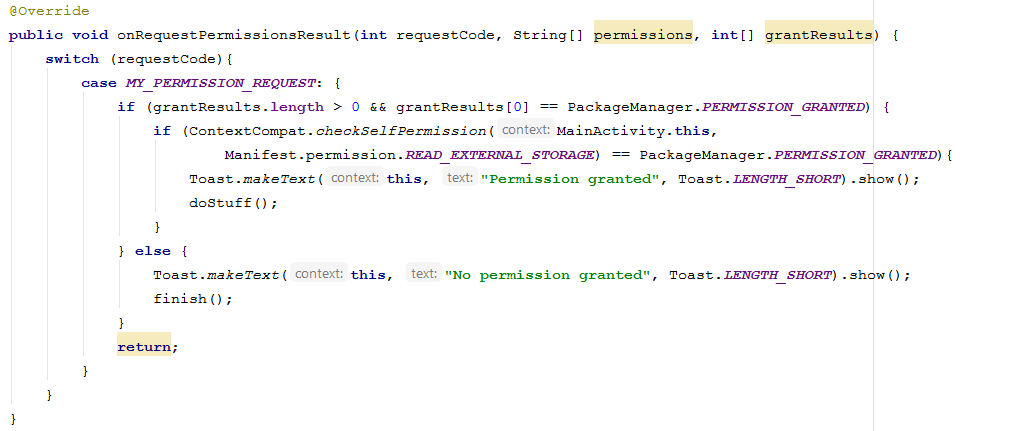
# **Thiết kế và mô tả giao diện**

## Màn hình ảnh chờ

* Khi vừa mở, ứng dụng sẽ load hình ảnh chờ lên trước.

## Chức năng hỏi quyền truy cập

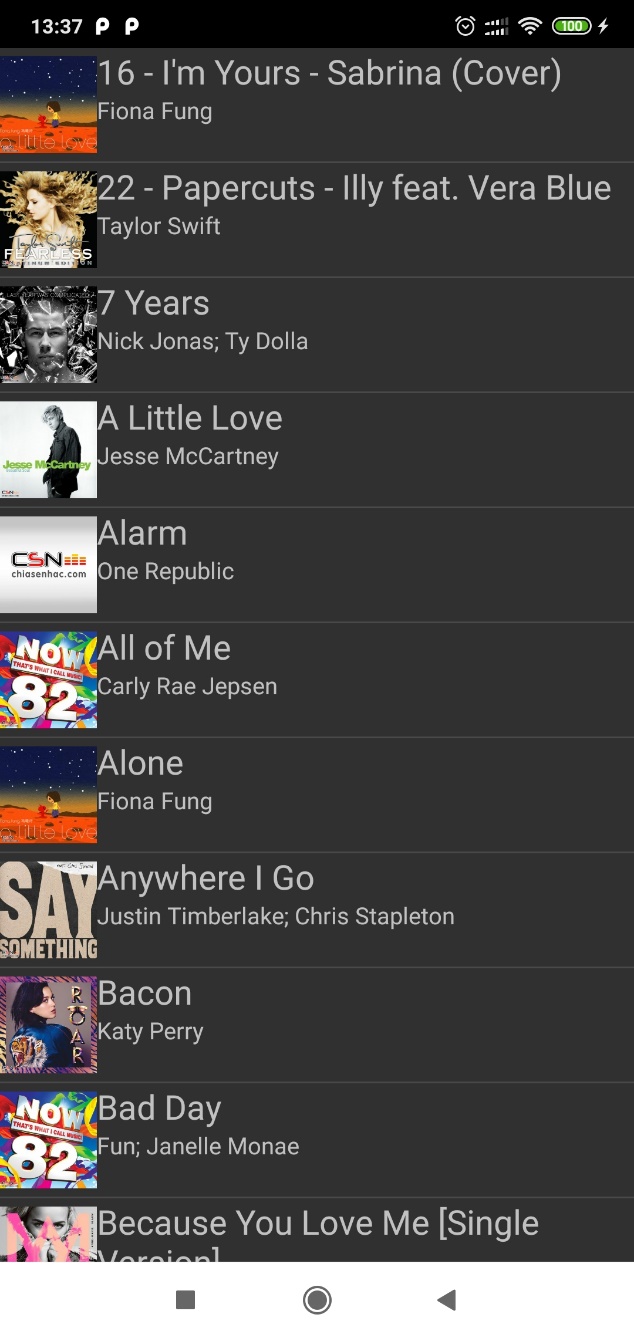
* ****Sau hình ảnh chờ xuất hiện, ứng dựng sẽ yêu cầu quyền đọc bộ nhớ.

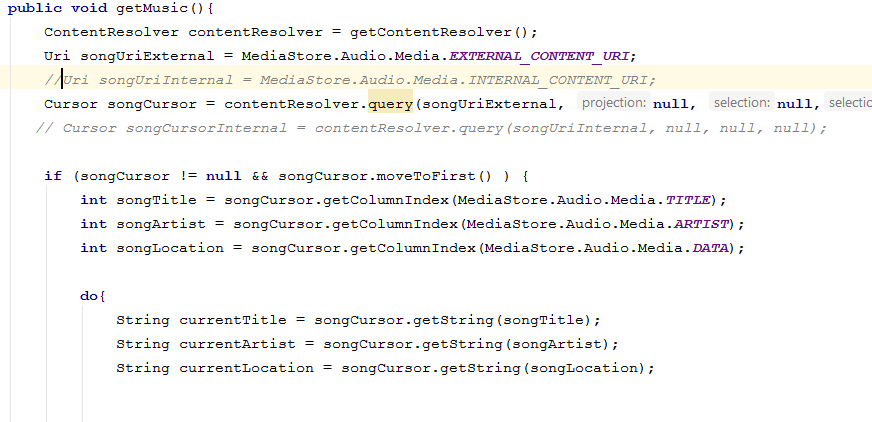


* Nếu cấp quyền ok thì ứng dụng sẽ đọc tất cả các file Audio trong bộ nhớ ngoài và hiển thị lên một listview.

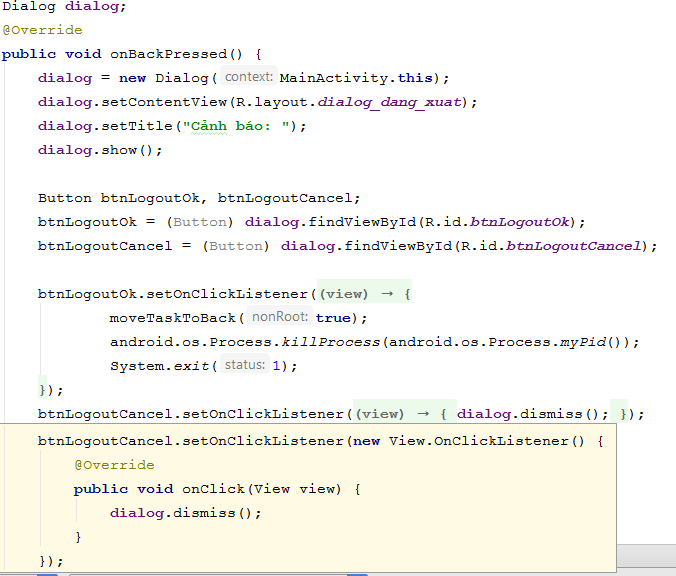
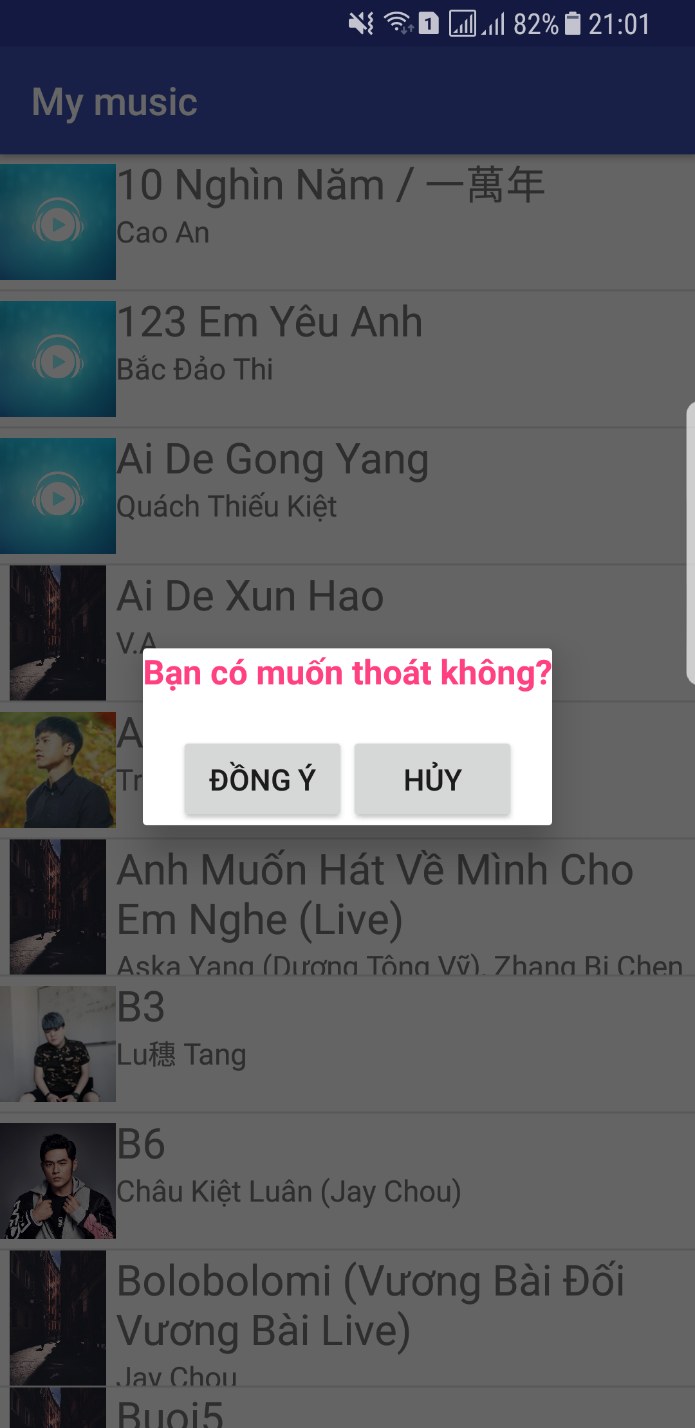
## Chứ năng đọc toàn bộ file mp3 trong bộ nhớ trong

* Đọc tất cả audio file trong Media.

****

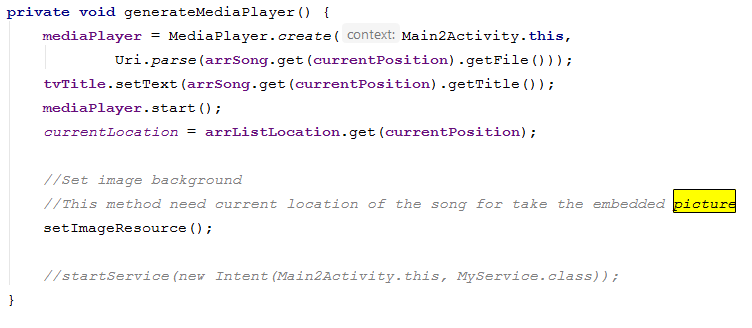
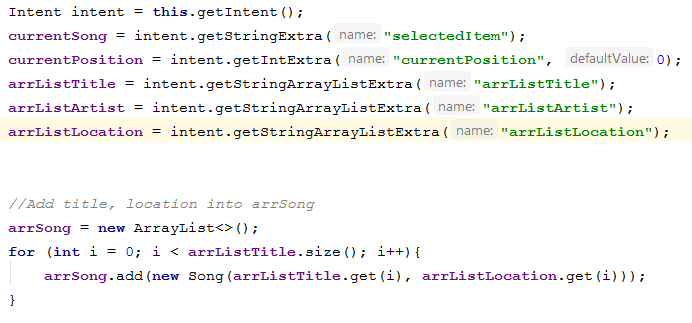
****

## Dialog xuất hiện khi người dùng bấm nút back trên điện thoại

****

## Khi người dùng ấn vào 1 trong những bài hát ở listview thì sẽ chuyển qua layout khác và chơi bản nhạc đó.

* Sử dụng intent để truyền các giá trị: currnetPosition (vị trí của bài hát trong arraySong)
* Từ vị trí, ta lấy đường dẫn thư mục, tên bài hát… và sau đó phát nhạc.

********

## Button phát ngẫu nhiên (Shuffle)

* Khi đang chơi nhạc, ấn chọn nút Shuffle thì bài hát ngẫu nhiên trong danh sách sẽ được phát khi kết thúc bài hát hiện tại, còn khi click lại thì sẽ phát nhạc theo thứ tự.

****

## Button phay/pause, previous, next.

* Khi đang chơi nhạc, ấn nút play thì nhạc đó sẽ dừng (mediaPlayer.stop()), nếu click lại thì sẽ tiếp tục chơi bản nhạc đó (mediaPlayer.start())
* Khi đang chơi nhạc, nếu click nút next thì sẽ dừng bài hát hiện tại (mediaPlayer.stop()), sau đó sẽ chơi bản nhạc tiếp theo trong danh sách.
* Mỗi lần click thì vị trí hiện tại trong danh sách ( currentPosition) sẽ tăng lên 1.
* Nếu bài hát hiện tại là bài hát cuối cùng trong danh sách thì sẽ bắt đầu lại, chơi bản nhạc đầu danh sách.

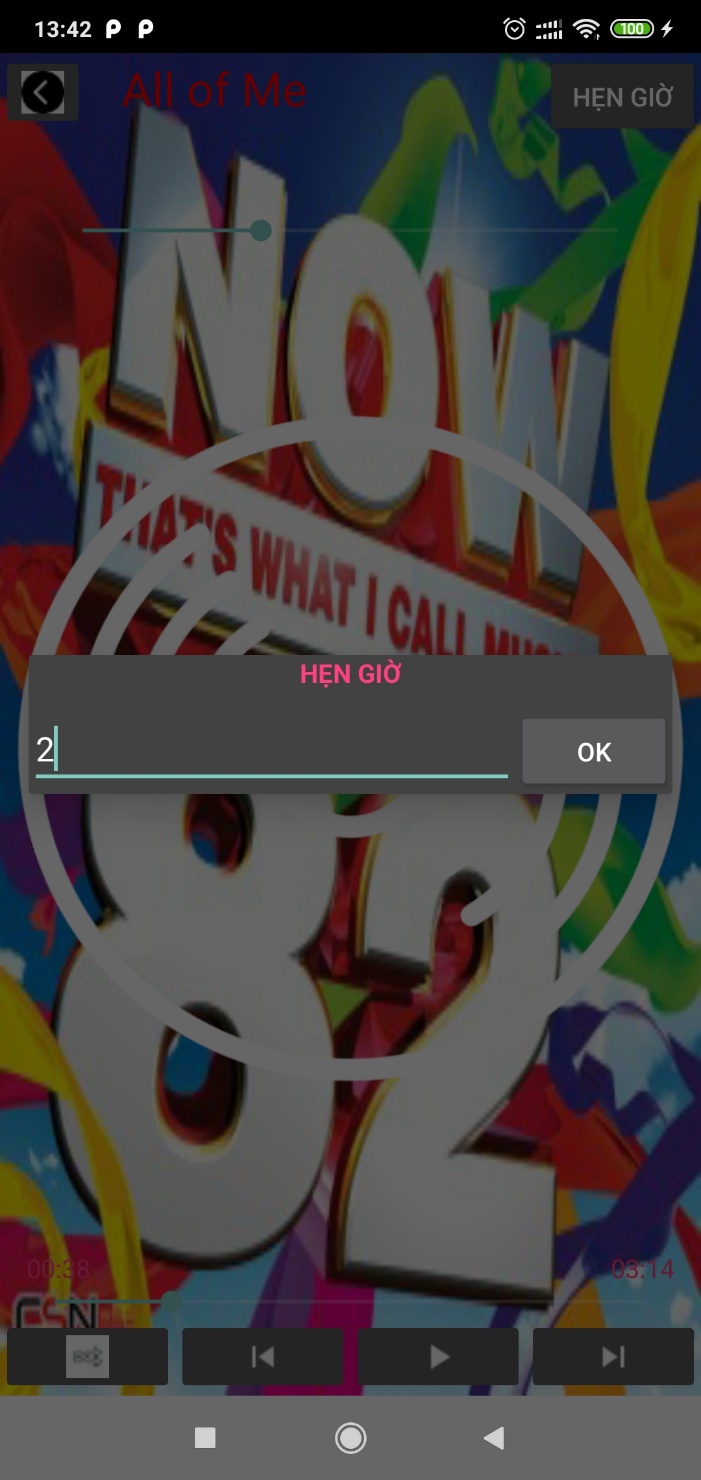
**currentPosition**++;  
**if** (**currentPosition** > **arrSong**.size() -1 ){  
 **currentPosition** = 0;  
}  
**if** (**mediaPlayer**.isPlaying()){  
 **mediaPlayer**.stop();  
}  
**if** (**mediaPlayer**.isPlaying() == **false**){  
 **btnPlay**.setImageResource(R.drawable.***iconpause***);  
}  
generateMediaPlayer();

* Nút previous.
* Tương tự như nút next, chỉ khác là currentPosition sẽ trừ đi 1.

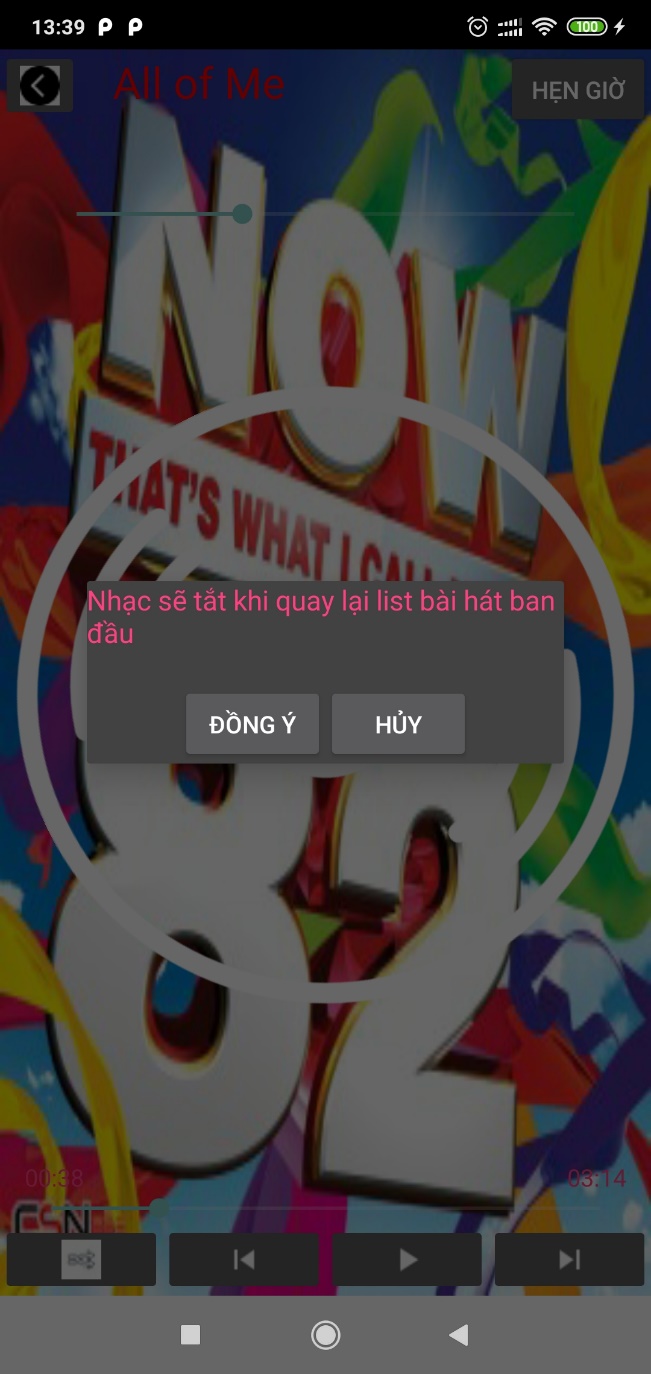
**currentPosition**--;  
**if** (**currentPosition** < 0 ){  
 **currentPosition** = **arrSong**.size() -1 ;  
}  
**if** (**mediaPlayer**.isPlaying()){  
 **mediaPlayer**.stop();  
}  
**if** (**mediaPlayer**.isPlaying() == **false**){  
 **btnPlay**.setImageResource(R.drawable.***iconpause***);  
}  
generateMediaPlayer();

## Hẹn giờ khi đang phát nhạc

* Khi nhấn vào button hẹn giờ sẽ xuất hiện dialog cho người dùng nhập số phút cần phát nhạc, sau khi hết thời gian hẹn giờ thì nhạc sẽ tự động ngưng phát.



## Xử lý sự kiện khi ấn nút back về giao diện listView hoặc người dùng ấn nút back trên điện thoại.

****

* Hàm showDialog dùng để khai báo và hiển thị dialog ra màn hình chính.
* Nếu người dùng ấn ok, sẽ dừng bài hát đang phát và quay về màn hình chính để chọn bài khác.

Tổng kết

Trong đồ án này chúng em đã nghiên cứu, tìm hiểu về ứng dụng phát nhạc trên Android. Đồ án đã thực hiện những nhiệm vụ sau:

* Tìm hiểu lịch sử và cấu trsuc của hệ điều hành Android.
* 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 thnahf công ứng dụng Android nghe nhạc offline.

Tuy nhiên, do thời gian và khả năng có hạn, nên chúng em chưa đi sâu tì hiểu thêm về ứng dụng, vì vậy giao diện ứng dụng vẫn còn sơ sài, xử lý code chưa tốt, ứng dụng còn nhiều thiếu sót. Rất mong thầy và các bạn giúp đỡ để ứng dụng 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 ứng dụng nghe nhạc online phổ biến.

TÀI LIỆU THAM KHẢO

* 1. <https://tinhte.vn/threads/tim-hieu-qua-trinh-phat-trien-cua-android-qua-cac-phien-ban.1367201/>
  2. <https://viblo.asia/p/tim-hieu-mediaplayer-trong-android-ORNZq99bZ0n>
  3. <https://www.thegioididong.com/hoi-dap/cac-dinh-dang-video-va-am-thanh-pho-bien-hien-nay-740243>
  4. Kênh youtube của Khoa Phạm