

**BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG**

---



ISO 9001:2015

**ĐỒ ÁN TỐT NGHIỆP**

**NGÀNH: CÔNG NGHỆ THÔNG TIN**

**Sinh viên : Đoàn Hoàng Hải**  
**Giảng viên hướng dẫn: ThS. Phùng Anh Tuấn**

**HẢI PHÒNG - 2018**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG**

-----

**XÂY DỰNG ỨNG DỤNG ANDROID NGHE NHẠC OFFLINE**

**ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY  
NGÀNH: CÔNG NGHỆ THÔNG TIN**

**Sinh viên : Đoàn Hoàng Hải  
Giảng viên hướng dẫn: ThS. Phùng Anh Tuấn**

**HẢI PHÒNG - 2018**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**  
**TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG**

---

**NHIỆM VỤ ĐỀ TÀI TỐT NGHIỆP**

Sinh viên: Đoàn Hoàng Hải

Mã SV: 1412101004

Lớp: CT1802

Ngành: Công nghệ thông tin

Tên đề tài: Xây dựng ứng dụng Android nghe nhạc offline

## MỤC LỤC

LỜI MỞ ĐẦU.....	7
LỜI CẢM ƠN.....	8
CHƯƠNG 1: HỆ ĐIỀU HÀNH ANDROID.....	9
1.1. Giới thiệu về hệ điều hành Android.....	9
1.2. Kiến trúc cơ bản của hệ điều hành Android.....	12
1.2.1. Nhân Linux.....	12
1.2.2. Thư viện.....	12
1.2.3. Thực thi.....	13
1.2.4. Nền tảng Android.....	13
1.2.5. Tầng ứng dụng.....	14
1.3. Giao diện hệ điều hành Android.....	14
1.3.1. Phiên bản Android 1.5.....	16
1.3.2. Phiên bản Android 1.6.....	16
1.3.3. Phiên bản Android 2.0 - 2.1.....	16
1.3.4. Phiên bản Android 2.2 - 2.2.3.....	17
1.3.5. Phiên bản Android 2.3 - 2.3.7.....	17
1.3.6. Phiên bản Android 3.0 - 3.2.6.....	18
1.3.7. Phiên bản Android 4.0 - 4.0.4.....	18
1.3.8. Phiên bản Android 4.1 - 4.3.1.....	18
1.3.9. Phiên bản Android 4.4 - 4.4.4.....	19
1.3.10. Phiên bản Android 5.0 - 5.1.1.....	20
1.3.11. Phiên bản Android 6.0 - 6.0.1.....	20
1.3.12. Phiên bản Android 7.0 - 7.1.2.....	21
1.3.13. Phiên bản Android 8.0 - 8.1.....	22
CHƯƠNG 2: MÔI TRƯỜNG PHÁT TRIỂN ỨNG DỤNG ANDROID STUDIO.....	23
2.1. Giới thiệu ứng dụng Android Studio.....	23
2.2. Cài đặt môi trường lập trình Android.....	23
2.2.1. Cài đặt JAVA JDK.....	23
2.2.2 Cài đặt Android Studio.....	24
2.2.3 Máy ảo Android Genymotion.....	28

2.3. Thành phần trong một ANDROID PROJECT.....	29
2.3.1. Tập cấu hình Android.....	29
2.3.2. Thư mục Java.....	31
2.3.3. Thư mục Res.....	31
2.3.4. Tập Grade Scripts .....	32
2.4. Thành phần giao diện.....	32
2.4.1. View group.....	32
2.4.2. View.....	33
2.5. Vòng đời ứng dụng android.....	35
2.6. Intent.....	36
2.6.1. Khái niệm .....	36
2.6.2. Các loại Intent.....	37
2.7. Share preferences.....	40
2.7.1. Khái niệm .....	40
2.7.2. Cách sử dụng .....	40
2.8. Hiệu ứng trong android .....	41
2.8.1. Hiệu ứng cơ bản.....	41
2.8.2. Cách sử dụng .....	41
<b>CHƯƠNG 3: TỔNG QUAN VỀ PHÁT NHẠC TRONG ANDROID .....</b>	<b>43</b>
3.1. Giới thiệu file nhạc số .....	43
3.2. Một số định dạng file nhạc số phổ biến.....	44
3.2.1. Định dạng file nhạc số MP3.....	44
3.2.2. Một số định dạng file nhạc số khác .....	49
3.3. Chỉnh sửa thông tin file nhạc.....	51
3.3.1. Tải về và cài đặt.....	52
3.3.2. Một số thao tác cơ bản trong mp3tag .....	52
3.3. Phát nhạc trong android .....	57
3.3.1. Xin quyền truy cập bộ nhớ ngoài .....	57
3.3.2. Thư viện phát nhạc MediaPlayer.....	57
<b>CHƯƠNG 4: CHƯƠNG TRÌNH THỰC NGHIỆM.....</b>	<b>59</b>
4.1. Phát biểu bài toán .....	59
4.2. Thiết kế chức năng hệ thống.....	59

4.2.1. Sơ đồ chức năng.....	59
4.2.2. Xây dựng chương trình.....	59
4.3. Giao diện chương trình .....	67
TỔNG KẾT .....	68
TÀI LIỆU THAM KHẢO .....	71

## LỜI MỞ ĐẦU

Hiện nay Công nghệ thông tin vô cùng phát triển thì mọi người đều sử dụng máy vi tính hoặc điện thoại di động để làm việc và giải trí. Do đó việc xây dựng các ứng dụng cho điện thoại di động đang là một ngành công nghiệp mới đầy tiềm năng và hứa hẹn nhiều sự phát triển vượt bậc của ngành khoa học kỹ thuật.

Phần mềm, ứng dụng cho điện thoại di động hiện nay rất đa dạng và phong phú trên các hệ điều hành di động. Các hệ điều hành J2ME, Adroid, 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 Adroid 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 hiện nay. Adroid đã nhanh chóng là đối thủ cạnh tranh mạnh mẽ với các hệ điều hành trước đó và đang là hệ điều hành di động của tương lai và được nhiều người ưa chuộng nhất.

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 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.

## LỜI CẢM ƠN

Em xin gửi lời cảm ơn chân thành nhất đến quý thầy cô Trường Đại Học Dân Lập Hải Phòng, những người đã dìu dắt em tận tình, đã truyền đạt cho em những kiến thức và bài học quý báu trong suốt thời gian em theo học tại trường.

Em xin trân trọng gửi lời cảm ơn đến tất cả các thầy cô trong khoa Công Nghệ Thông Tin, đặc biệt là thầy giáo ThS. Phùng Anh Tuấn, thầy đã tận tình hướng dẫn và giúp đỡ em trong suốt quá trình làm tốt nghiệp. Với sự chỉ bảo của thầy, em đã có những định hướng tốt trong việc triển khai và thực hiện các yêu cầu trong quá trình làm đồ án tốt nghiệp.

Em xin cảm ơn những người thân và gia đình đã quan tâm, động viên và luôn tạo cho em những điều kiện tốt nhất trong suốt quá trình học tập và làm tốt nghiệp.

Ngoài ra, em cũng xin gửi lời cảm ơn tới tất cả bạn bè, đặc biệt là các bạn trong lớp CT1802 đã luôn gắn bó, cùng học tập và giúp đỡ em trong những năm qua và trong suốt quá trình thực hiện đồ án này.

Em xin chân thành cảm ơn!

*Hải Phòng, ngày 01 tháng 08 năm 2018*

Sinh viên

**Đoàn Hoàng Hải**



## CHƯƠNG 1: HỆ ĐIỀU HÀNH ANDROID

### 1.1. Giới thiệu về hệ điều hành Android

Android là một hệ điều hành dựa trên nền tảng Linux[1], được thiết kế dành cho các thiết bị di động có màn hình cảm ứng như điện thoại thông minh và máy tính bảng. Ban đầu, Android được phát triển bởi Tổng công ty Android, với sự hỗ trợ tài chính từ Google và sau này được chính Google mua lại vào năm 2005.

Chính mã nguồn mở của Android cùng với tính không ràng buộc nhiều đã cho phép các nhà phát triển thiết bị di động và các lập trình viên được điều chỉnh và phân phối Android một cách tự do. Ngoài ra, Android còn có một cộng đồng lập trình viên đông đảo chuyên viết các ứng dụng để mở rộng chức năng của thiết bị.

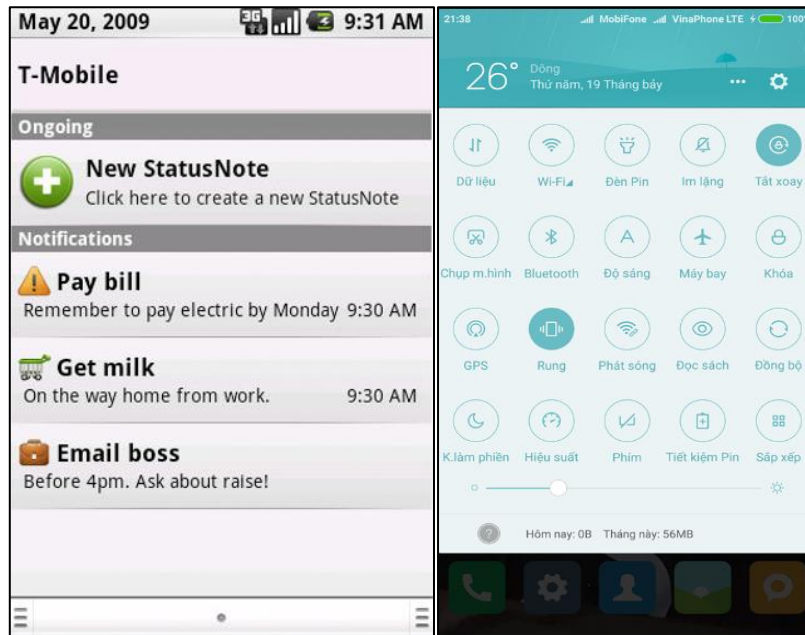
Nhờ yếu tố mở, dễ dàng tinh chỉnh cùng sự phát triển nhanh chóng đã khiến hệ điều hành này dần trở nên phổ biến, kết quả là mặc dù được thiết kế để chạy trên điện thoại và máy tính bảng nhưng giờ đây Android đã xuất hiện trên các smart TV, máy chơi game và một số thiết bị điện tử khác.

Android bắt đầu với bản beta đầu tiên vào tháng 11 năm 2007 và phiên bản thương mại đầu tiên, Android 1.0, được phát hành vào tháng 9 năm 2008. Kể từ tháng 4 năm 2009, phiên bản Android được phát triển, đặt tên theo chủ đề bánh kẹo và phát hành theo thứ tự bảng chữ cái: Cupcake, Donut, Eclair, Froyo, Gingerbread, Honeycomb, Ice Cream Sandwich, Jelly Bean, Kitkat, Lollipop, Marshmallow, Nougat, và bây giờ là Oreo.

Kỷ nguyên của Android chính thức bắt đầu vào ngày 22 tháng 10 năm 2008, khi chiếc điện thoại T-Mobile G1 bắt đầu được bán ra tại Mỹ. Vào thời gian đầu, rất nhiều tính năng cơ bản bị thiếu sót như: bàn phím ảo, cảm ứng đa điểm và tính năng mua ứng dụng vẫn chưa xuất hiện. Tuy nhiên, một số tính năng cũng như giao diện đặc sản của hệ điều hành này đã khởi nguồn từ chiếc G1 và trở thành những yếu tố không thể thiếu trên Android sau này.

Sự phát triển của hệ điều hành Android:

- **Thanh thông báo vuốt từ trên xuống (Notification bar):** Ngay từ những ngày đầu tiên của Android, thanh thông báo này đã đánh dấu một bước quan trọng mà trước đây chưa hề có hệ điều hành nào làm được - đưa tất cả thông tin tin nhắn, tin thoại hoặc các cuộc gọi nhớ chỉ với thao tác vuốt xuống.



Hình 1.1.1. Thanh thông báo ở phiên bản thời kỳ đầu(trái) so với phiên bản Android 8.1(phải)

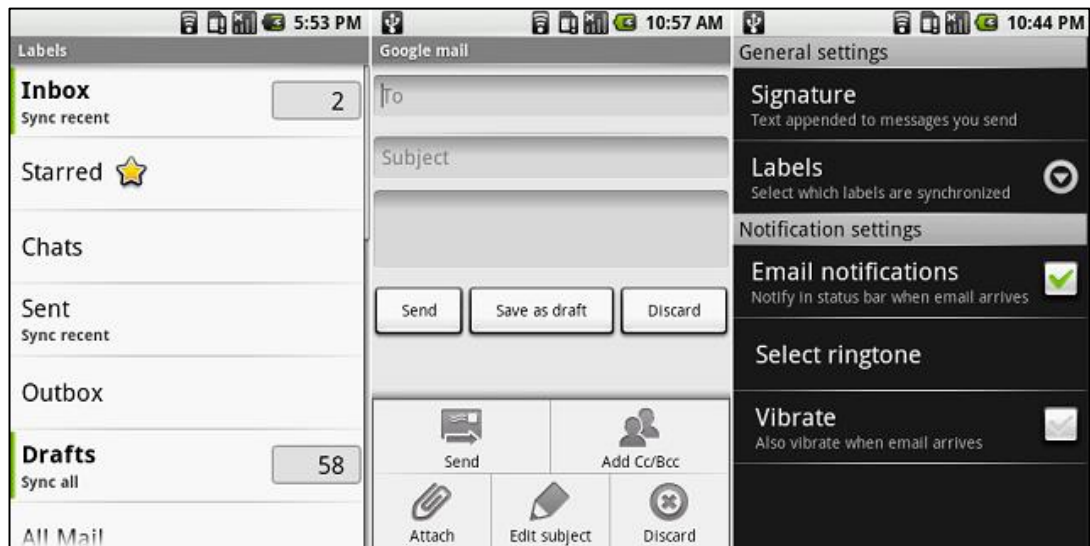
- Màn hình chính (Home Screen) và các widget: Một điểm khác biệt giữa Android so với các hệ điều hành khác là phần màn hình chính của mình. Bên cạnh việc thay đổi được hình nền, Android còn cho phép người dùng tùy biến màn hình chính của mình với nhiều widgets kèm theo, chẳng hạn như đồng hồ, lịch, trình nghe nhạc, đưa các icon ứng dụng ra ngoài hoặc thậm chí có thể can thiệp sâu hơn để thay đổi toàn bộ giao diện màn hình Home Screen này.



Hình 1.1.2. Màn hình chính của Android 1.0(trái) so với Android 8.1(phải)

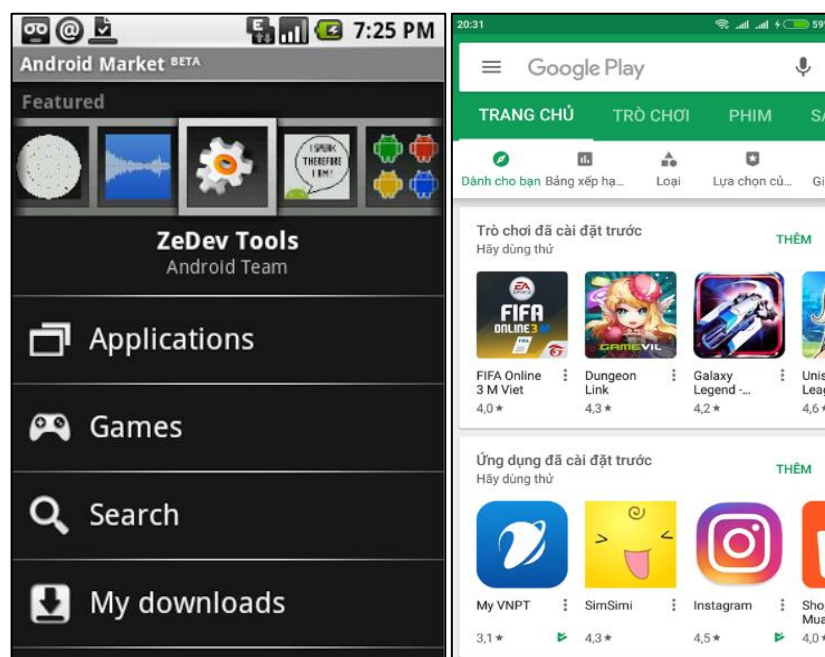
- Đồng bộ và tích hợp chặt chẽ với Gmail: Vào thời điểm điện thoại G1 được bán ra, Gmail đã hỗ trợ giao thức POP và IMAP để tích hợp với các trình email trên di động. Tuy nhiên, lúc bấy giờ không có bất kì sản phẩm nào có

thể hỗ trợ được hoàn toàn những tính năng ưu việt này của Gmail. Mãi cho đến khi Android 1.0 xuất hiện, vấn đề này đã được khắc phục và G1 trở thành chiếc điện thoại mang lại trải nghiệm Gmail tốt nhất trên thị trường lúc bấy giờ.



Hình 1.1.3. Gmail trên Android thời kỳ đầu

- Kho ứng dụng Android Market: Thật khó có thể tưởng tượng một chiếc smartphone mà không hề có kho ứng dụng, nhưng vào thời điểm Android mới ra mắt, gần như không có bất kì điện thoại nào có kho ứng dụng nào được tích hợp và chính Android đã mở đầu cuộc cách mạng ứng dụng di động này. Android Market trên G1 thời bấy giờ có rất ít ứng dụng và giao diện cực kỳ đơn giản, hơn nữa tính năng mua ứng dụng trên phiên bản này vẫn chưa được xuất hiện mãi cho đến năm sau - những vấn đề này dễ hiểu vì thời điểm này Android chỉ mới được khai sinh nên mọi thứ còn khá thô sơ.

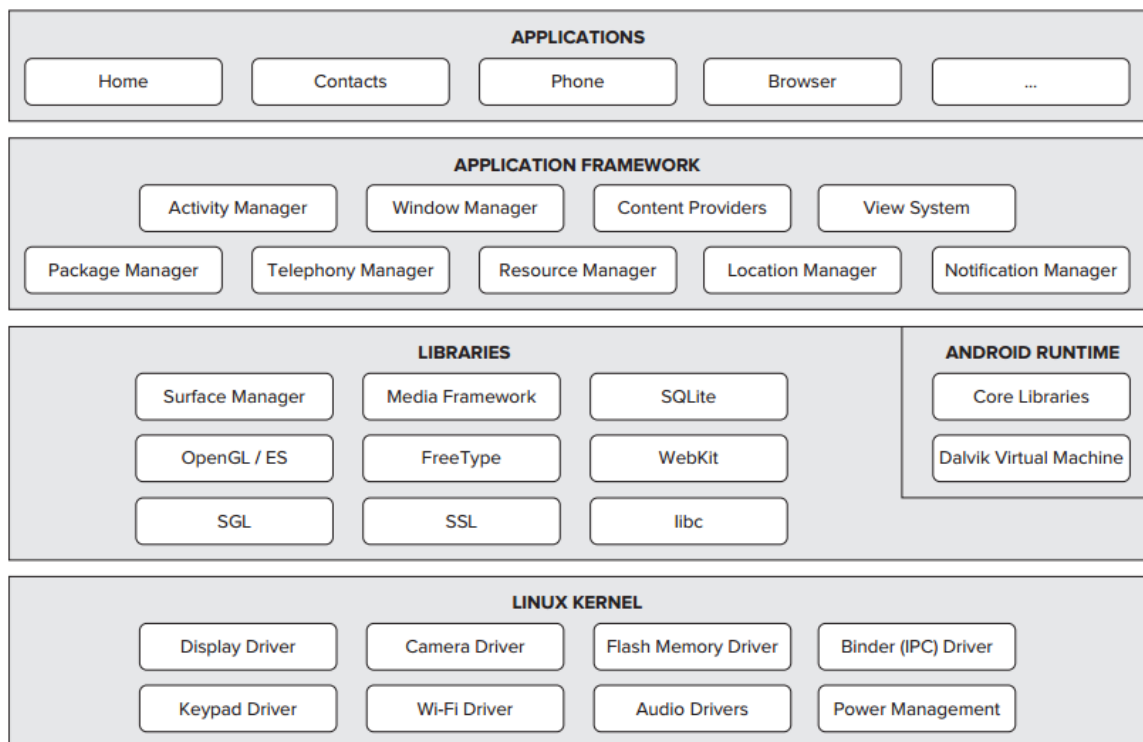


Hình 1.1.4. Hình ảnh Store của Android 1.0(trái) và Android 8.1(phải)

- Giao diện: Google đã phát triển giao diện Android phiên bản 1.0 với sự hỗ trợ từ TAT, viết tắt từ The Astonishing Tribe, một công ty thiết kế tương tác của Thụy Điển. Dấu ấn rõ ràng nhất mà TAT để lại trên phiên bản Android từ phiên bản 1.0 cho đến 2.2 chính là widget đồng hồ kim nằm ngoài Home Screen tuy đơn giản nhưng rất đẹp mắt. Công ty này sau đó ngừng hợp tác với Google và bị RIM mua lại để tập trung phát triển sản phẩm Blackberry cũng như nền tảng BBX sau này.

## 1.2. Kiến trúc cơ bản của hệ điều hành Android

Android gồm năm phần chính sau được chứa trong bốn lớp:



Hình 1.2.1. Kiến trúc hệ điều hành Android

### 1.2.1. Nhân Linux

Android dựa trên Linux phiên bản 2.6 cho hệ thống dịch vụ cốt lõi như security, memory management, process management, network stack, and driver model. Kernel Linux hoạt động như một lớp trừu tượng hóa giữa phần cứng và phần còn lại của phần mềm stack.

### 1.2.2. Thư viện

Android bao gồm một tập hợp các thư viện C/C++ được sử dụng bởi nhiều thành phần khác nhau trong hệ thống Android. Điều này được thể hiện thông qua nền tảng ứng dụng Android. Một số các thư viện cơ bản được liệt kê dưới đây:

- Hệ thống thư viện C: một BSD có nguồn gốc từ hệ thống thư viện tiêu chuẩn C (libc), điều chỉnh để nhúng vào các thiết bị dựa trên Linux.

- Thư viện Media - dựa trên PacketVideo's OpenCORE; các thư viện hỗ trợ phát lại và ghi âm của âm thanh phổ biến và các định dạng video, cũng như các tập tin hình ảnh tĩnh, bao gồm cả MPEG4, H.264, MP3, AAC, AMR, JPG, and PNG.
- Bề mặt quản lý - Quản lý việc truy xuất vào hệ thống hiển thị.
- LibWebCore - một công cụ trình duyệt web hiện đại mà quyền hạn cả hai trình duyệt web Android và xem web nhúng.
- SGL - Đồ họa 2D cơ bản của máy.
- Thư viện 3D - một thực hiện dựa vào OpenGL ES 1.0 APIs; các thư viện sử dụng phần cứng tăng tốc 3D (nếu có), tối ưu hóa cao rasterizer phần mềm 3D.
- FreeType - vẽ phông chữ bitmap và vector.

**SQLite** một công cụ cơ sở dữ liệu quan hệ mạnh mẽ và nhẹ có sẵn cho tất cả các ứng dụng.

### 1.2.3. Thực thi

Android bao gồm một tập hợp các thư viện cơ bản mà cung cấp hầu hết các chức năng có sẵn trong các thư viện lõi của ngôn ngữ lập trình Java. Tất cả các ứng dụng Android đều chạy trong tiến trình riêng. Máy ảo Dalvik đã được viết để cho một thiết bị có thể chạy nhiều máy ảo hiệu quả. Các VM Dalvik thực thi các tập tin thực thi Dalvik (dex). Định dạng được tối ưu hóa cho bộ nhớ tối thiểu. VM là dựa trên register-based, và chạy các lớp đã được biên dịch bởi một trình biên dịch Java để chuyển đổi thành các định dạng dex. Các VM Dalvik dựa vào nhân Linux cho các chức năng cơ bản như luồng và quản lý bộ nhớ thấp.

### 1.2.4. Nền tảng Android

Bằng cách cung cấp một nền tảng phát triển mở, Android cung cấp cho các nhà phát triển khả năng xây dựng các ứng dụng cực kỳ phong phú và sáng tạo. Nhà phát triển được tự do tận dụng các thiết bị phần cứng, thông tin địa điểm truy cập, các dịch vụ chạy nền, thiết lập hệ thống báo động, thêm các thông báo để các thanh trạng thái, và nhiều, nhiều hơn nữa. Nhà phát triển có thể truy cập vào các API cùng một khuôn khổ được sử dụng bởi các ứng dụng lõi. Các kiến trúc ứng dụng được thiết kế để đơn giản hóa việc sử dụng lại các thành phần; bất kỳ ứng dụng có thể xuất bản khả năng của và ứng dụng nào khác sau đó có thể sử dụng những khả năng (có thể hạn chế bảo mật được thực thi bởi khuôn khổ). Cơ chế này cho phép các thành phần tương tự sẽ được thay thế bởi người sử dụng.

Cơ bản tất cả các ứng dụng là một bộ các dịch vụ và các hệ thống, bao gồm:

- Một tập hợp rất nhiều các View có khả năng kế thừa lẫn nhau dùng để thiết kế phần giao diện ứng dụng như: gridview, tableview, linearlayout

- Một “Content Provider” cho phép các ứng dụng có thể truy xuất dữ liệu từ các ứng dụng khác (chẳng hạn như Contacts) hoặc là chia sẻ dữ liệu giữa các ứng dụng đó.
- Một “Resource Manager” cung cấp truy xuất tới các tài nguyên không phải là mã nguồn, chẳng hạn như: localized strings, graphics, and layout files.
- Một “Notification Manager” cho phép tất cả các ứng dụng hiển thị các custom alerts trong status bar. Activity Manager được dùng để quản lý chu trình sống của ứng dụng và điều hướng các activity.

### 1.2.5. Tầng ứng dụng

Tầng ứng dụng (Application) là tầng giao tiếp với người dùng với các thiết bị Android như Danh bạ, tin nhắn, trò chơi, tiện ích tính toán, trình duyệt... Mọi ứng dụng viết đều nằm trên tầng này.

## 1.3. Giao diện hệ điều hành Android

Giao diện người dùng của Android dựa trên nguyên tắc tác động trực tiếp, sử dụng cảm ứng chạm tương tự như những động tác ngoài đời thực như vuốt, chạm, kéo giãn và thu lại để xử lý các đối tượng trên màn hình. Sự phản ứng với tác động của người dùng diễn ra gần như ngay lập tức, nhằm tạo ra giao diện cảm ứng mượt mà, thường dùng tính năng rung của thiết bị để tạo phản hồi rung cho người dùng. Những thiết bị phần cứng bên trong như gia tốc kế, con quay hồi chuyển và cảm biến khoảng cách được một số ứng dụng sử dụng để phản hồi một số hành động khác của người dùng, ví dụ như điều chỉnh màn hình từ chế độ hiển thị dọc sang chế độ hiển thị ngang tùy theo vị trí của thiết bị, hoặc cho phép người dùng lái xe đua bằng xoay thiết bị, giống như đang điều khiển vô-lăng.

Các thiết bị Android sau khi khởi động sẽ hiển thị màn hình chính, điểm khởi đầu với các thông tin chính trên thiết bị, tương tự như khái niệm desktop (bàn làm việc) trên máy tính để bàn. Màn hình chính Android thường gồm nhiều biểu tượng (icon) và tiện ích (widget); biểu tượng ứng dụng sẽ mở ứng dụng tương ứng, còn tiện ích hiển thị những nội dung sống động, cập nhật tự động như dự báo thời tiết, hộp thư của người dùng, hoặc những mẩu tin thời sự ngay trên màn hình chính. Màn hình chính có thể gồm nhiều trang xem được bằng cách vuốt ra trước hoặc sau, mặc dù giao diện màn hình chính của Android có thể tùy chỉnh ở mức cao, cho phép người dùng tự do sắp đặt hình dáng cũng như hành vi của thiết bị theo sở thích. Những ứng dụng do các hãng thứ ba có trên Google Play và các kho ứng dụng khác còn cho phép người dùng thay đổi "chủ đề" của màn hình chính, thậm chí bắt chước hình dáng của hệ điều hành khác như Windows Phone chẳng hạn. Phần lớn những nhà sản xuất, và một số nhà mạng, thực hiện thay đổi hình dáng và hành vi của các thiết bị Android của họ để phân biệt với các hãng cạnh tranh.

Ở phía trên cùng màn hình là thanh trạng thái, hiển thị thông tin về thiết bị và tình trạng kết nối. Thanh trạng thái này có thể "kéo" xuống để xem màn hình thông báo gồm thông tin quan trọng hoặc cập nhật của các ứng dụng, như email hay tin nhắn SMS mới nhận, mà không làm gián đoạn hoặc khiến người dùng cảm thấy bất tiện. Trong các phiên bản đời đầu, người dùng có thể nhấn vào thông báo để mở ra ứng dụng tương ứng, về sau này các thông tin cập nhật được bổ sung thêm tính năng, như có khả năng lập tức gọi ngược lại khi có cuộc gọi nhỡ mà không cần phải mở ứng dụng gọi điện ra. Thông báo sẽ luôn nằm đó cho đến khi người dùng đã đọc hoặc xóa nó đi.

#### 1.4. Phiên bản Android

Phiên bản	Tên	Ngày phát hành
Android 1.5	Cupcake	27/4/2009
Android 1.6	Donut	15/9/2009
Android 2.0 - 2.1	Eclair	26/9/2009 (phát hành lần đầu)
Android 2.2 - 2.2.3	Froyo	20/5/2010 (phát hành lần đầu)
Android 2.3 - 2.3.7	Gingerbread	6/12/2010 (phát hành lần đầu)
Android 3.0 - 3.2.6	Honeycomb	22/2/2011 (phát hành lần đầu)
Android 4.0 - 4.0.4	Ice Cream Sandwich	18/10/2011 (phát hành lần đầu)
Android 4.1 - 4.3.1	Jelly Bean	9/7/2012 (phát hành lần đầu)
Android 4.4 - 4.4.4	KitKat	31/10/2013 (phát hành lần đầu)
Android 5.0 - 5.1.1	Lollipop	12/11/2014 (phát hành lần đầu)
Android 6.0 - 6.0.1	Marshmallow	5/10/2015 (phát hành lần đầu)
Android 7.0 - 7.1.2	Nougat	22/8/2016 (phát hành lần đầu)
Android 8.0 - 8.1	Oreo	21/8/2017 (phát hành lần đầu)

Hình 1.4.1. Lịch sử ra đời của các phiên bản android

### 1.3.1. Phiên bản Android 1.5

Android 1.5 **Cupcake** có lẽ có vai trò cực kì quan trọng trong quá trình trưởng thành của Android khi nó bổ sung cho hệ điều hành này những tính năng nổi bật giúp nó cạnh tranh với các nền tảng đối thủ khác[2]. Đây cũng là bản Android đầu tiên được Google gọi tên theo các món đồ ăn với chữ cái bắt đầu được xếp theo thứ tự alphabet. Về mặt giao diện, Android 1.5 không có nhiều điểm thay đổi so với người tiền nhiệm của mình. Google chỉ điểm thêm vài điểm để làm giao diện trông bóng bẩy, mượt mà hơn một tí, chẳng hạn như widget tìm kiếm có độ trong suốt nhẹ, biểu tượng app drawer có một số hoa văn nhỏ mới, v.v. Nói chung, giao diện không phải là một điểm nhấn của Android 1.5 mà người ta quan tâm nhiều hơn đến các tính năng mới mà nó mang lại.

### 1.3.2. Phiên bản Android 1.6

Phiên bản **Donut** này, mặc dù chỉ thêm có 0.1 vào mã số của Android 1.5 nhưng nó cũng mang lại nhiều cải tiến đáng giá. Một vài điểm trong giao diện được cải thiện, vài tính năng nhỏ được thêm vào, cuối cùng là hỗ trợ cho mạng CDMA. Bổ sung tính năng có thể chạy trên nhiều độ phân giải và tỉ lệ màn hình khác nhau, cho phép những thiết bị có nhiều độ phân giải hơn là 320 x 480. Hiện nay, chúng ta có những chiếc smartphone Android chạy ở độ phân giải QVGA, HVGA, WVGA, FWVGA, qHD, và 720p. Vài chiếc máy tính bảng còn đạt mức 1920 x 1080 nữa.

### 1.3.3. Phiên bản Android 2.0 - 2.1

Đầu tháng 9 năm 2009, Android 2.0 **Eclair** đã được ra mắt trên chiếc Motorola Droid cùng các tính năng mới:

Hỗ trợ nhiều tài khoản người dùng: lần đầu tiên, nhiều tài khoản Google có thể được đăng nhập trên cùng một thiết bị chạy Android. Tài khoản Microsoft Exchange cũng được hỗ trợ trong Eclair. Người dùng có thể duyệt qua danh bạ, email của từng tài khoản. Google giờ đây cho phép những nhà phát triển bên thứ ba tích hợp dịch vụ của họ vào trong mục Account này, đồng thời hỗ trợ tự động đồng bộ hóa.

- Quick Contact: khi chạm vào một số liên lạc nào đó, có một menu nhỏ sẽ xuất hiện để bạn tương tác theo nhiều cách: gửi email, gọi điện, nhắn tin và hơn thế nữa. Miễn là nơi nào có biểu tượng contact xuất hiện là nơi đó có thể dùng Quick Contact. Sau này có thêm Twitter, Facebook và nhiều dịch vụ đồng bộ khác cũng tích hợp tính năng riêng của mình vào Quick Contact.
- Cải tiến bàn phím ảo: Giống với chiếc G1, Droid/Milestone có một bàn phím QWERTY đầy đủ dạng trượt ngang nhưng Google cho phép người dùng sử dụng thêm bàn phím ảo mà hãng đã thiết kế lại. Mặc dù tính năng cảm ứng vẫn còn một điều xa vời với Android vào lúc đó (trình duyệt, bản đồ và cả các phần mềm khác không hề dùng được tính năng hai ngón tay để phóng to, thu



nhỏ,...) nhưng Eclair lại sử dụng dữ liệu multitouch từ bàn phím để xác định điểm chạm thứ hai trong lúc người dùng gõ nhanh. Độ chính xác của bàn phím ảo trên Android 2.0 nhờ đó đã được cải tiến đáng kể.

- Trình duyệt mới: Mặc dù chưa hỗ trợ cảm ứng đa điểm nhưng trình duyệt trên Android 2.0 cũng có nhiều điểm nâng cấp đáng khen. Google đã hỗ trợ HTML5, bao gồm luôn khả năng phát video ở chế độ toàn màn hình. Hộp địa chỉ kết hợp với thanh tìm kiếm (giống với trình duyệt Chrome) cũng lần đầu xuất hiện trên Android. Để bù lại cho tính năng cảm ứng đa điểm, trình duyệt mới hỗ trợ chạm hai lần để phóng to nội dung trên màn hình, kèm theo đó là hai nút Zoom in, Zoom out.
- Giao diện mới: các biểu tượng giờ đây đã đẹp hơn, sang trọng hơn, gọn gàng hơn nhiều so với trước. Widget cũng được thiết kế mới để tương thích với độ phân giải cao trên Droid. Giao diện menu cũng đẹp hơn nữa.

Android 2.1 ra mắt chủ yếu để sửa lỗi và thêm hàm API để lập trình viên can thiệp sâu hơn vào hệ thống nhưng nó đã hỗ trợ thêm vài tính năng lý thú như Live Wallpaper, chuyển giọng nói thành văn bản và một màn hình khóa mới. Android 2.1 được đánh dấu bởi sự ra đời của chiếc Nexus One do HTC sản xuất, chiếc điện thoại Nexus đầu tiên chạy Android chuẩn của Google.

#### **1.3.4. Phiên bản Android 2.2 - 2.2.3**

Android 2.2 **Froyo** được ra mắt trong năm 2010. Nexus One là chiếc điện thoại đầu tiên được nâng cấp lên Android 2.2, sớm hơn nhiều so với tất cả các hãng khác. Giao diện màn hình chính đã được thay đổi, từ 3 màn hình chính tăng lên thành 5 màn hình. Google đã có nhiều cố gắng để giao diện Android được vui và đẹp hơn, dễ dùng hơn, bắt kịp phần nào với giao diện của bên thứ ba như HTC Sense chẳng hạn.

Froyo còn có một trình xem ảnh mới với khả năng hiển thị hình ảnh 3D: nghiêng máy thì dãy ảnh sẽ nghiêng theo, đồng thời nó mang trong mình nhiều hiệu ứng chuyển động đẹp mắt khác.

Một trong những thay đổi nổi bật ở phiên bản Froyo là tính năng USB tethering và Wi-Fi hotspot. Các thay đổi khác là hỗ trợ dịch vụ Cloud to Device Messaging (C2DM), cho phép đẩy thông báo. Ngoài ra còn cải thiện tốc độ ứng dụng, được hiện thực thông qua biên dịch JIT. Nhiều tính năng bảo mật mới cũng xuất hiện.

#### **1.3.5. Phiên bản Android 2.3 - 2.3.7**

Khoảng nửa năm sau khi Froyo xuất hiện, Google đã trở lại với bản Android 2.3 **Gingerbread**. Google giới thiệu nó với nhiều tính năng mới, tập trung vào việc phát triển game, đa phương tiện và phương thức truyền thông mới. Bên cạnh đó, chiếc Nexus S do Samsung sản xuất cũng xuất hiện với vai trò là người kế nhiệm cho Nexus One. Nexus S có vài thay đổi nhỏ so với người anh em Galaxy S.

Giao diện Gingerbread được tinh chỉnh rất nhiều, giúp người dùng làm chủ dễ dàng hơn, dễ sử dụng hơn, và tiết kiệm năng lượng hơn. Một bộ màu đơn giản với màu nền đen khiến cho thanh thông báo, menu và các thành phần giao diện khác nổi bật lên. Những cải thiện trong menu và thiết lập giúp người dùng điều hướng dễ dàng hơn.

Gingerbread sử dụng nhân Linux phiên bản 2.6.35.

### **1.3.6. Phiên bản Android 3.0 - 3.2.6**

**Honeycomb** là phiên bản Android dành riêng cho máy tính bảng, và sản phẩm đầu tiên dùng hệ điều hành này Motorola Xoom. Mặc dù Android 3.0 không có nhiều dấu ấn đặc biệt trên thị trường nhưng nó là nền tảng cho Android 4.0 với các tính năng như. Ngoài ra, Honeycomb còn hỗ trợ cho việc bố cục ứng dụng theo nhiều cột để hướng đến việc hỗ trợ máy tính bảng tốt hơn.

### **1.3.7. Phiên bản Android 4.0 - 4.0.4**

Cuối năm 2011, Google chính thức giới thiệu điện thoại Galaxy Nexus, thiết bị đầu tiên trên thị trường sử dụng Android 4.0 **Ice Cream Sandwich**. Có thể nói Android 4.0 là sự thay đổi lớn nhất trong lịch sử phát triển của Android tính đến ngày viết bài này. Android 4.0 hỗ trợ một bộ font mới tên là Roboto được cho là tối ưu hóa để dùng trên các màn hình độ phân giải càng ngày càng cao hơn, đồng thời để hiển thị được nhiều thông tin hơn trên màn hình.

Hệ thống thông báo (Notification) già nua của Android đã được làm mới hoàn toàn, đẹp hơn, tiện dụng hơn, nhất là tính năng trượt ngang để xóa từng thông báo riêng lẻ. Tương tự như vậy cho tính năng Recent Apps và cả trình duyệt của máy. Bàn phím cũng được làm mới với khả năng tự động sửa lỗi cao hơn, việc sao chép, cắt dán chữ và nội dung cũng tốt hơn bao giờ hết.

Đây cũng là lần đầu tiên Google hợp nhất hệ điều hành dành cho smartphone và cho máy tính bảng vào làm một.

Android 4.0 nhận được nhiều đánh giá tích cực từ các nhà phê bình, họ khen ngợi giao diện hệ điều hành được tân trang, gọn gàng hơn so với các phiên bản trước, cùng với các cải thiện về hiệu suất và tính năng. Tuy nhiên, các nhà phê bình vẫn cảm thấy các ứng dụng gốc của Android 4.0 vẫn còn thua kém về chất lượng và tính năng so với các ứng dụng tương đương của bên thứ ba, và một số tính năng mới của hệ điều hành, đặc biệt là "mở khóa bằng khuôn mặt", không cần thiết lắm.

### **1.3.8. Phiên bản Android 4.1 - 4.3.1**

Android **Jelly Bean** đánh dấu sự ra đời với chiếc máy tính bảng Nexus 7 do Asus sản xuất. Về giao diện, không có nhiều thay đổi so với Android 4.0, vẫn là màn hình chính với thanh dock bên dưới quen thuộc. Sự xuất hiện của Google Now cho thấy rằng Google đã bắt đầu bước chân vào việc cạnh tranh với Siri.

Trên Android 4.1 bạn có thể tìm kiếm bằng giọng nói và các kết quả trả không chỉ đơn giản là những dòng tìm kiếm nữa mà nó được thiết kế theo dạng thẻ đồ họa, thông minh hơn, trực quan hơn. Không chỉ tìm kiếm theo yêu cầu. Nhưng có lẽ quan trọng hơn hết của Jelly Bean không phải là về giao diện hay ứng dụng mới mà về Project Butter giúp mang lại độ mượt chưa từng có cho Android.

Ngày 30 tháng 10 năm 2012, Google chính thức tuyên bố cập nhật hệ điều hành Android của hãng lên phiên bản 4.2 và vẫn giữ nguyên tên gọi "Jelly Bean". Được hãng gọi là "một hương vị mới của Jelly Bean", Android 4.2 mang trong mình nhiều tính năng mới như hỗ trợ Miracast, bàn phím có thể nhập liệu bằng cách vẽ các đường nét từ kí tự này đến kí tự khác, chế độ chụp ảnh toàn cảnh Photo Sphere, ứng dụng Gmail mới và còn rất nhiều thứ khác nữa.

Một cải tiến lớn và quan trọng của Android 4.2 đó là việc hỗ trợ nhiều tài khoản người dùng trên máy tính bảng để có thể dễ dàng chia sẻ trong gia đình hoặc nơi làm việc. Mỗi tài khoản sẽ có dữ liệu app của riêng họ. Ví dụ, người dùng A cài game Angry Birds, người này chơi và đã ghi được một số điểm cũng như lên được màn cao hơn. Khi người B sử dụng máy bằng tài khoản khác, game này vẫn hiện diện trên máy nhưng dưới dạng một bản cài mới, không có sẵn data của người A.

Lại thêm một thế hệ Jelly Bean nữa và lần này là Android 4.3. Ngày 24/7/2013, Google đã chính thức ra mắt hệ điều hành này song song với chiếc Nexus 7 (2013). Đây là phiên bản Android mới nhất đang có mặt trên thị trường và đi kèm những tính năng mới như hỗ trợ kết nối Bluetooth Smart, bộ API OpenGL ES 3.0, bổ sung tính năng sử dụng Wi-Fi để định vị ngay cả khi người dùng tắt kết nối này đi cùng nhiều thay đổi lớn nhỏ khác. Bên cạnh tính năng Multi User của Android 4.2 như đã nói ở trên, Google bổ sung thêm một tính năng mới cho Android 4.3 đó là Restricted Profile. Mỗi thành viên sẽ có một "hồ sơ" riêng của mình và chỉ có thể làm được những gì được chỉ định trong "hồ sơ". Ví dụ, cha mẹ có thể khóa tính năng in-app purchase của một ứng dụng nào đó để ngăn con trẻ vô tình mua hàng trăm USD. Ngoài ra, khi áp dụng vào môi trường doanh nghiệp thì nó sẽ cho phép người quản trị phân quyền cho nhân viên dễ dàng hơn.

### **1.3.9. Phiên bản Android 4.4 - 4.4.4**

Google tiết lộ thêm rằng "mục tiêu của chúng tôi với Android **KitKat** đó là mang trải nghiệm Android đáng kinh ngạc đến cho mọi người". Android Kitkat xuất hiện đầu tiên trên chiếc smartphone Nexus 5. Những cải tiến trên Android Kitkat bao gồm Chế độ toàn màn hình - Immersive Mode, Hiệu ứng chuyển cảnh màn hình - Transition Manager, Storage Access Framework, Chromium WebView, NFC, Cổng hồng ngoại - Infrared Blasters ...

Giao diện: Tông màu chủ đạo của font chữ là xanh và đen được sử dụng trên Android Jelly Bean được Google thay đổi bằng tông màu đen và ghi trên Android 4.4 Kitkat vừa ra mắt, thể hiện rõ ở các dòng chữ trên thanh thông báo Notification. Bên cạnh đó, Android Kitkat cũng có bộ biểu tượng, folder mới, giao diện nhìn chung được làm phẳng bớt, loại bỏ các chi tiết đổ bóng và thêm hiệu ứng trong suốt vào nhiều phần, như ba phim ảo bên dưới hay thanh thông báo ở phía trên.

Dù là phiên bản mới, Android 4.4 Kitkat lại không đòi hỏi cấu hình phần cứng mạnh hơn, thậm chí hỗ trợ tốt cả những thiết bị cũ với phần cứng không cao, như có RAM chỉ đạt dung lượng 512 MB. Hệ điều hành mới được Google tối ưu khả năng hoạt động, cho hiệu năng cao hơn tới 1,6 lần phiên bản trước.

### **1.3.10. Phiên bản Android 5.0 - 5.1.1**

Sau hơn một năm Android 4.0 KitKat ra mắt, Google đã chính thức trình làng phiên bản hậu duệ mới nhất với tên gọi Android **Lollipop**, được đánh giá là hệ điều hành có những thay đổi rõ rệt và đáng kể nhất trong lịch sử Android từ trước đến nay.

Một trong những thay đổi lớn nhất trong Lollipop là được thiết kế lại giao diện người dùng được xây dựng dựa trên ngôn ngữ thiết kế gọi là Material Design. Những thay đổi khác bao gồm cải tiến thanh thông báo, có thể truy cập từ màn hình khóa và hiển thị trong ứng dụng như banner trên màn hình. Google còn thay đổi bên trong nền tảng này, với Android Runtime (ART) chính thức thay thế cho Dalvik để cải thiện hiệu năng ứng dụng, và tối ưu hóa việc sử dụng pin, gọi nội bộ là Project Volta.

### **1.3.11. Phiên bản Android 6.0 - 6.0.1**

Vào ngày 5 tháng 10 năm 2015 thì Google đã giới thiệu đến người dùng phiên bản Android kế tiếp với mã hiệu **Marshmallow**. Với những thay đổi đợt cập nhật này tuy nhỏ nhưng thiên về xu hướng hoàn thiện tương tác người dùng, cho một trải nghiệm tuyệt vời hơn.

Về mặt giao diện người dùng, màn hình chính có vẻ có khá ít thay đổi so với phiên bản trước ngoại trừ logo Google, tuy nhiên khi bạn vào trong Menu ứng dụng bạn có thể thấy khá nhiều thay đổi. Các ứng dụng được bố trí và cuộn xuống theo chiều dọc, ở phía trên cùng là thanh tìm kiếm và có một điểm thú vị là máy sẽ tự động gợi ý cho bạn bốn ứng dụng để sử dụng tùy theo từng địa điểm mà bạn đang ở.

Tính năng mới Now on Tap: khi ở màn hình của bất kỳ ứng dụng (ví dụ trình duyệt web) đang hiển thị thông tin, bạn có thể giữ nút home và máy sẽ bắt đầu "quét" những thông tin đang được hiển thị trên màn hình, gửi về Google và Google sẽ phản hồi lại cho bạn những thông tin, gợi ý có ích liên quan. Việc tích hợp Google theo cách này nâng trải nghiệm người dùng lên một tầm cao mới.

Tiết kiệm năng lượng hơn: Khi ở trạng thái không sử dụng, điện thoại Android đốt pin khá nhiều nếu so sánh với các điện thoại hệ điều hành khác. Google đã cho ra

đời chế độ Doze để giải quyết vấn đề về pin chờ điện thoại này. Khi máy bạn không được sử dụng, chế độ Doze sẽ được kích hoạt và sẽ hoạt động hơi giống như khi bạn bật chế độ Airplane trên điện thoại vậy: không kết nối, không thông báo, không có bất kỳ việc chạy ứng dụng nào diễn ra trong quá trình này để đảm bảo cho máy ở trạng thái tiết kiệm năng lượng tối đa có thể. Chỉ một số thông báo quan trọng như cuộc gọi, báo thức, tin nhắn là có thể hiển thị trong trạng thái Doze này.

Ngoài ra Android 6.0 cũng mang lại cho người dùng một số tính năng mới như: sử dụng thẻ SD cắm ngoài như bộ nhớ trong hay chính thức hỗ trợ nhận diện vân tay và Google Pay.

### **1.3.12. Phiên bản Android 7.0 - 7.1.2**

Android 7.0 **Nougat** được Google tung ra vào ngày 22 tháng 8 năm 2016 và những thiết bị Nexus sẽ là những chiếc smartphone đầu tiên được cập nhật phiên bản mới này. Tên gọi chính thức của phiên bản Android 7.0 đã được Google xác nhận, đó chính là một loại kẹo truyền thống có tên Nougat.

Android 7.0 hỗ trợ một số tính năng mới đáng kể như:

- Hỗ trợ Menu chuyển đổi nhanh giữa các cài đặt hệ thống: Một menu hamburger (biểu tượng dấu ba gạch nằm ngang) đã được thêm vào Cài đặt của hệ thống. Ví dụ như khi bạn đang trong phần cài đặt Bluetooth bạn có thể nhanh chóng chuyển đến các cài đặt khác của hệ thống bằng cách sử dụng Menu bên trái này.
- Trả lời nhanh tin nhắn từ thanh thông báo: Trong những bổ sung mới, thì đây là tính năng mới mẻ và khá giống với hệ điều hành iOS của Apple. Tin nhắn có thể được trả lời nhanh từ thông báo của chính nó. Với tính năng này người dùng không cần phải rời khỏi ứng dụng hiện tại để trả lời một tin nhắn hoặc thậm chí là mở khóa điện thoại.
- Chế độ chia đôi màn hình: Cuối cùng thì Google đã thêm tính năng vào Android 7.0, đây là tính năng mà bạn thường thấy trên các máy tính bảng, các dòng điện thoại của Samsung hay LG. Người dùng có thể chia màn hình thành hai phần, sử dụng hai ứng dụng song song cùng một lúc. Và bạn còn có thể tùy chọn kích thước cửa sổ cho mỗi ứng dụng bằng cách kéo thanh màu đen giữa hai ứng dụng sang trái hoặc phải, lên hoặc xuống.
- Trung tâm thông báo được làm mới: Trung tâm thông báo đã được google thiết kế lại. Các biểu tượng, phím tắt trên thanh thông báo được thu gọn và nổi bật hơn, bạn có thể mở rộng các phím tắt bằng phím mũi tên phía bên trái góc trên màn hình. Các thông báo chung từ một ứng dụng bây giờ đã được tối ưu gộp lại với nhau thành một dòng duy nhất và bạn có thể mở rộng bằng

cách sử dụng nút mũi tên hoặc thao tác với hai ngón tay. Giúp tối ưu hóa không gian trong trung tâm thông báo.

Ngoài ra còn một số tính năng đáng chú ý như: hỗ trợ chế độ thực tế ảo với VR, tích hợp chế độ tiết kiệm dữ liệu, chế độ tiết kiệm pin Doze được cải tiến.

### **1.3.13. Phiên bản Android 8.0 - 8.1**

Oreo" (tên mã phát triển là Android O) là phiên bản lớn thứ tám của hệ điều hành di động Android. Nó được phát hành lần đầu dưới dạng một phiên bản alpha xem trước cho nhà phát triển vào ngày 21 tháng 3 năm 2017. Bản xem trước thứ hai được phát hành ngày 17 tháng 5 năm 2017, được coi là phiên bản beta, và phiên bản xem trước thứ ba được phát hành ngày 8 tháng 6 năm 2017 với phần API được hoàn thiện. Vào ngày 24 tháng 7 năm 2017, một bản xem trước thứ tư được phát hành bao gồm những tính năng hệ thống cuối cùng cùng với những sửa lỗi và cải tiến mới nhất. Oreo được chính thức phát hành công khai vào ngày 21 tháng 8 năm 2017. Sau đó là Android 8.1 vào ngày 5 tháng 12 năm 2017. Oreo có một số cập nhật mới, nhiều điểm cải tiến so với bản tiền nhiệm như sideload (cài ứng dụng không có trên Store), giới hạn dữ liệu của ứng dụng nền, thông báo cho các ứng dụng chạy ở chế độ nền, Picture-in-picture, tự động điền mật khẩu trong ứng dụng, cải thiện thời lượng pin, Project Treble...

## CHƯƠNG 2: MÔI TRƯỜNG PHÁT TRIỂN ỨNG DỤNG ANDROID STUDIO

### 2.1. Giới thiệu ứng dụng Android Studio

Android Studio là một nền tảng IDE<sup>1</sup> (integrated development environment) dùng để phát triển các ứng dụng android, được Google release vào khoảng đầu năm 2015 thay thế cho bản Eclipse cũ. Android Studio được phát triển dựa trên IntelliJ IDEA Community Edition - công cụ lập trình tốt nhất cho java, giúp cho các lập trình viên tạo ứng dụng, thực hiện các thay đổi một cách dễ dàng, bên cạnh đó có thể xem trước trong thời gian thực và thiết kế giao diện đẹp hơn trước. Tiếng Việt cũng đã được tích hợp trong Android Studio. Đặc biệt, Android Studio cho phép người dùng Import Project từ Eclipse sang và logic lập trình cũng tương tự.

### 2.2. Cài đặt môi trường lập trình Android

#### 2.2.1. Cài đặt JAVA JDK

- Bước 1: Tải file cài đặt tại đường link

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

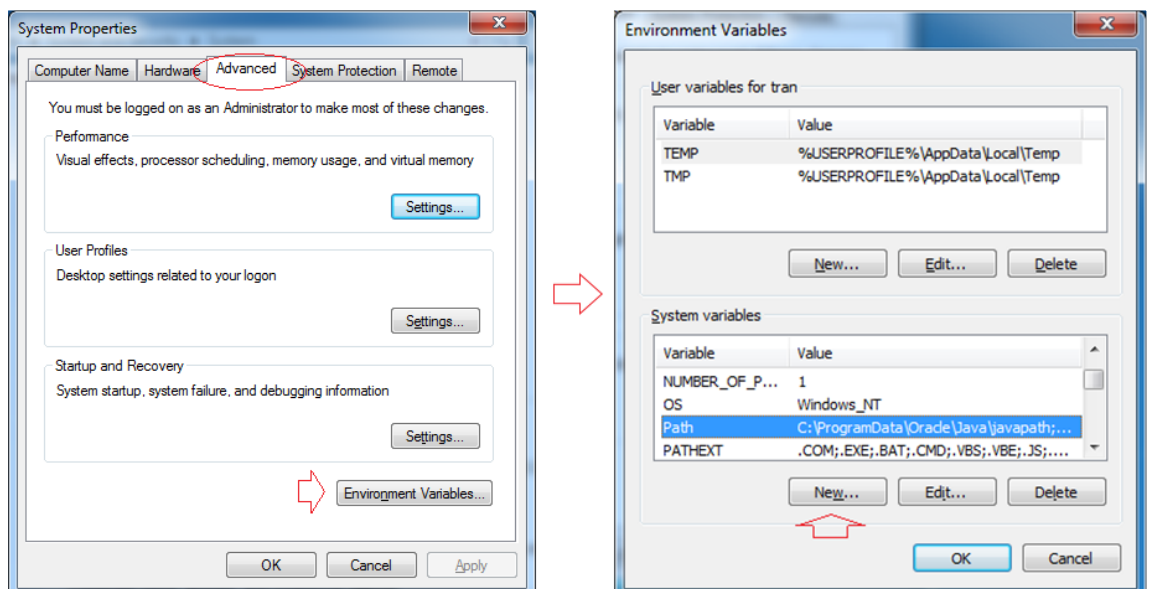
- Lưu ý: Chọn phiên bản tương ứng với hệ điều hành đúng với máy đang sử dụng.

- Bước 2: Mở file cài đặt “jdk-\*.exe” để tiến hành cài đặt

- Bước 3: Cấu hình biến môi trường cho Java

Việc này không bắt buộc, nhưng nếu trên máy tính của cài đặt nhiều phiên bản Java, thì việc cấu hình là cần thiết để xác định phiên bản java nào mặc định được sử dụng.

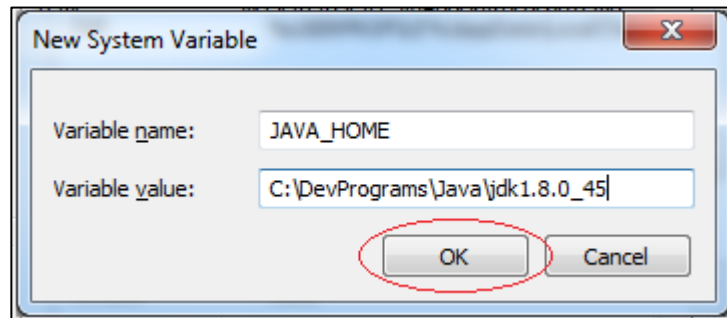
- Trên Desktop, nhấn phải chuột vào Computer, chọn **Properties > Advanced system settings**.



Hình 2.2.1.1. Thêm đường dẫn mới tới thư mục JDK

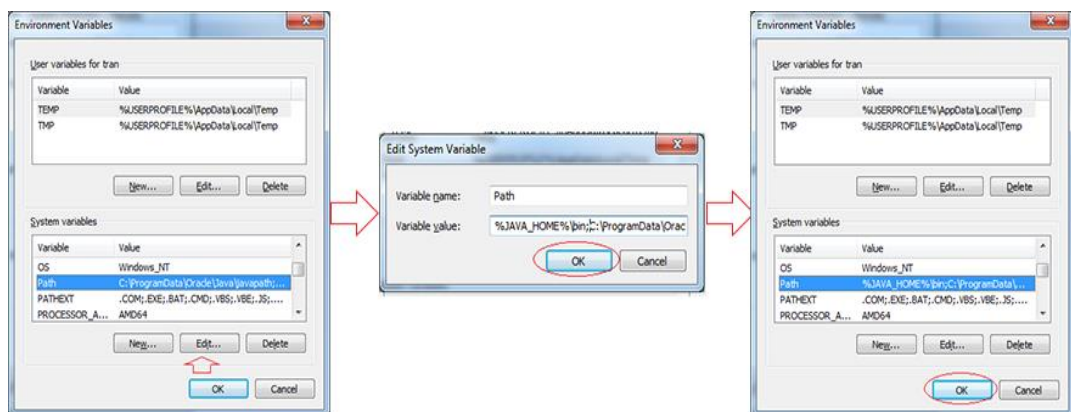
<sup>1</sup> Môi trường lập trình phát triển tích hợp

- Nhập vào đường dẫn tới thư mục JDK.



Hình 2.2.1.2. Nhập vào đường dẫn tới thư mục JDK

- Tiếp theo đổi tên path là đã cài đặt xong Java JDK



Hình 2.2.1.3. Đổi tên path

## 2.2.2 Cài đặt Android Studio

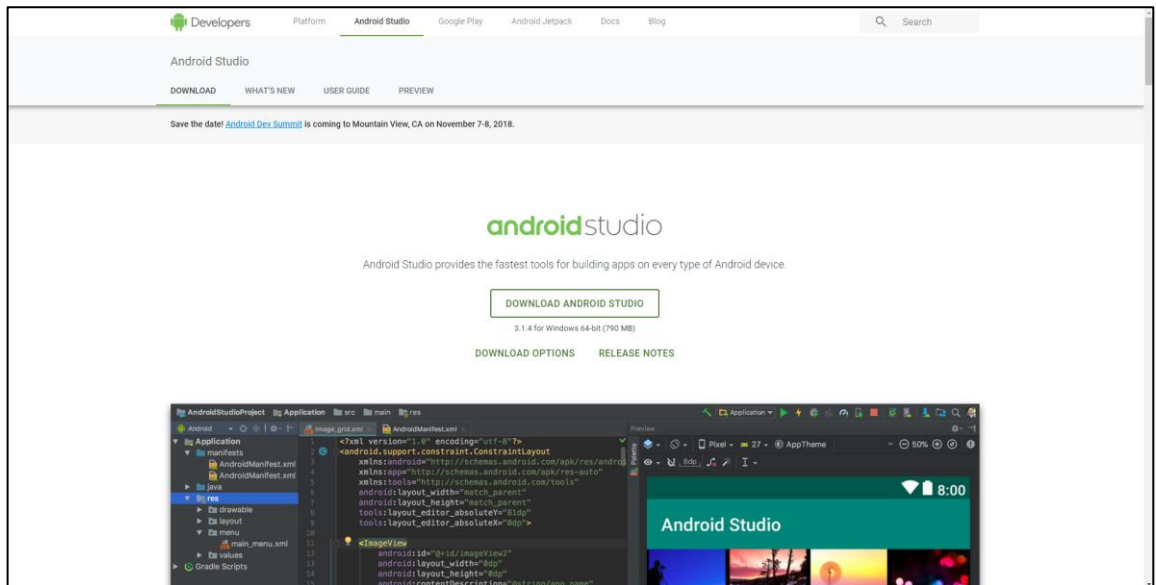
### a. Cấu hình yêu cầu:

- Microsoft® Windows® 10/8/7/Vista (32 or 64-bit)
- Tối thiểu 4 GB RAM, cấu hình đề nghị: 8 GB RAM
- Ổ cứng trống ít nhất : 20Gb
- Độ phân giải tối thiểu 1280 x 800
- Java Development Kit (JDK) 7 trở lên
- Lựa chọn thêm cho accelerated emulator: Intel® processor with support for Intel® VT-x, Intel® EM64T (Intel® 64), and Execute Disable (XD) Bit functionality

### b. Tải file cài đặt

Download file cài đặt: <https://developer.android.com/studio>

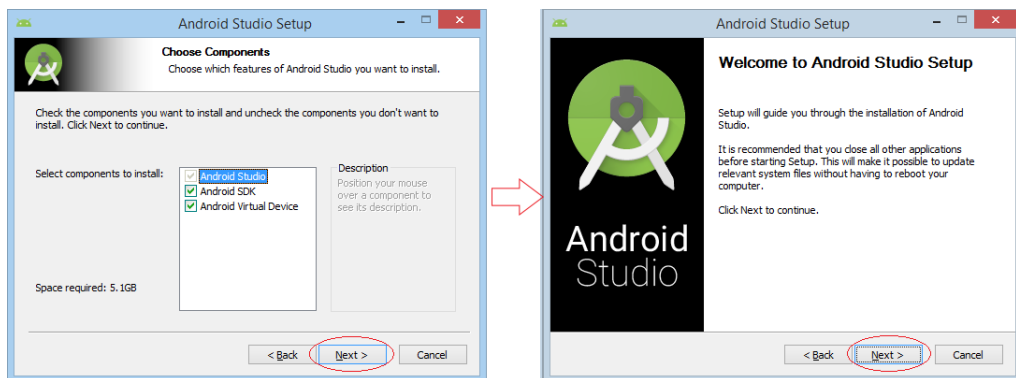




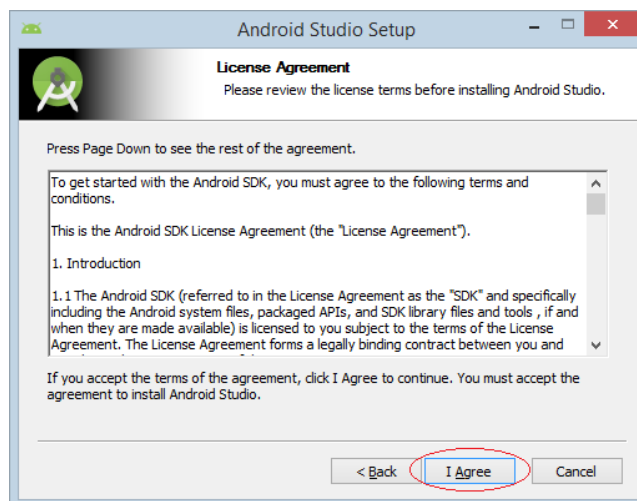
Hình 2.2.2.1. Giao diện trang web để tải file cài đặt

### c. Cách cài đặt

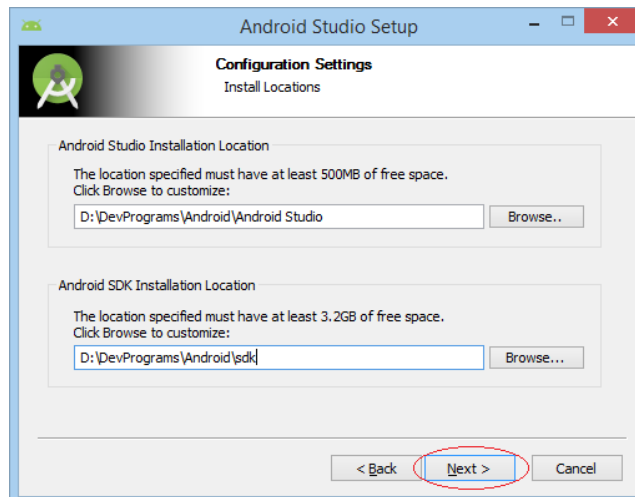
- Bước 1: Mở file cài đặt và ấn Next



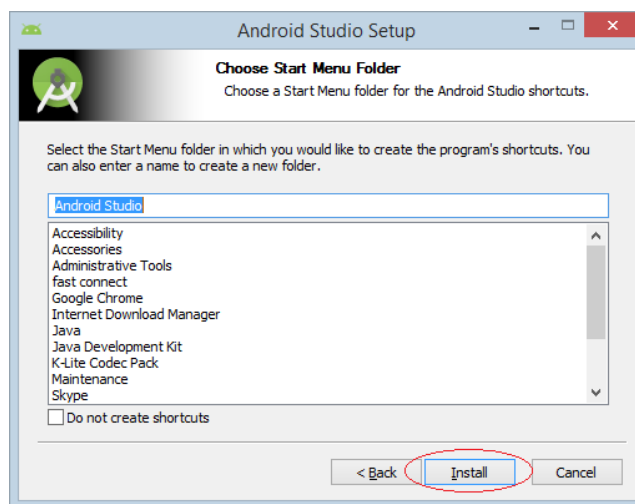
- Bước 2: Chọn “I agree” để xác nhận, đồng ý với các điều khoản và tiếp tục.



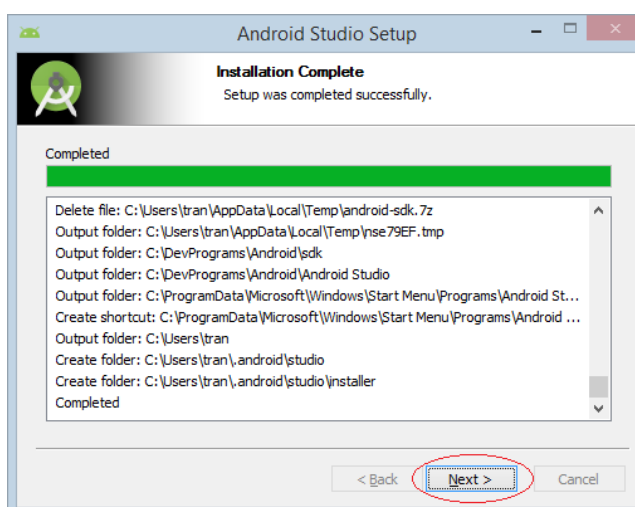
- Bước 3: Chọn nơi cài đặt Android Studio và Android SDK



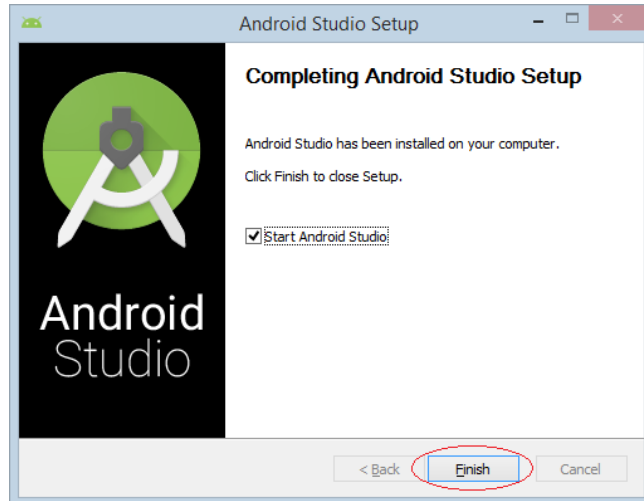
- Bước 4: Chọn Start Menu Folder. Ở bước này, có thể tích tạo short cut cho ứng dụng hoặc không. Bấm Install để bắt đầu cài đặt.



- Bước 5: Chờ hệ thống cài đặt các package cần thiết cho tới khi hoàn thành và bấm Next để tiếp tục.



- Bước 6: Bấm Finish và click Start Android Studio để khởi động phần mềm.



## 2.2.3 Máy ảo Android Genymotion

### a. Yêu cầu:

- Card đồ họa OpenGL 2.0.
- CPU hỗ trợ VT-x hoặc AMD-V và kích hoạt thiết lập BIOS.
- RAM: Tối thiểu 1 GB.
- Dung lượng trống của ổ cứng: Ít nhất là 2GB để cài đặt Genymotion và các máy ảo chạy Genymotion (đây chỉ là mức yêu cầu tối thiểu, bởi nếu sử dụng nhiều máy ảo cùng lúc và có nhiều ứng dụng, phần mềm cài đặt thì dung lượng trống có thể được yêu cầu nhiều hơn gấp 4 lần).
- Đảm bảo có kết nối Internet
- Độ phân giải màn hình: ít nhất 1024 x 768 pixel
- Oracle **VirtualBox 4.1** trở lên.
- Ngoài ra, người dùng cần có một tài khoản Genymotion để có thể sử dụng.

### b. Tải file cài đặt:

**Download:** <https://www.genymotion.com/fun-zone/>

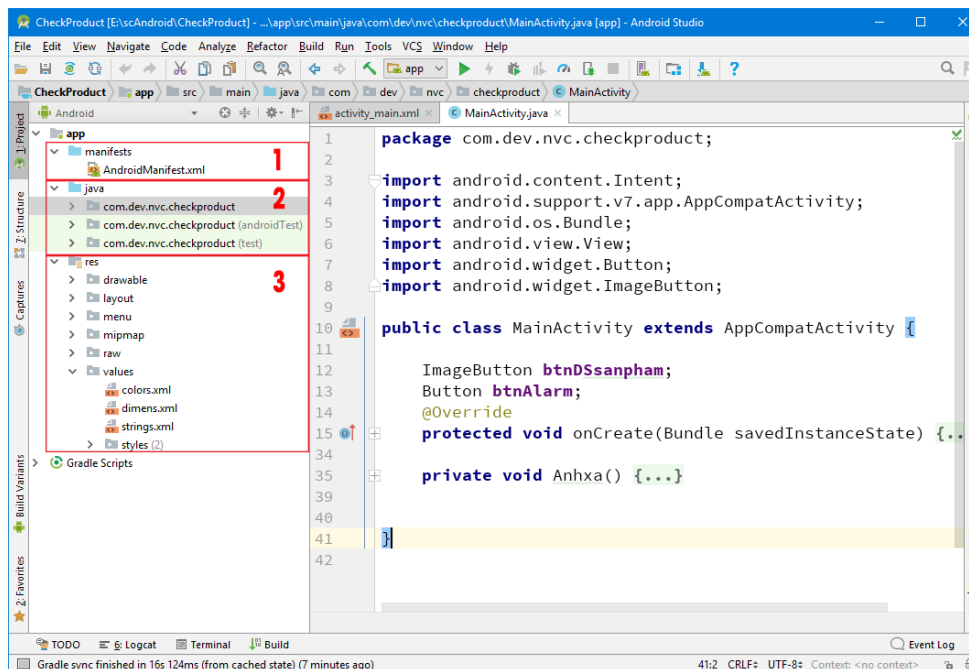
### c. Cách cài đặt:

- Bước 1: Tìm tới vị trí lưu file vừa tải về đầu tiên, sau đó click đúp chuột trái vào file đó để bắt đầu tiến hành. Giao diện đầu tiên sẽ cho lựa chọn ngôn ngữ để sử dụng.
- Bước 2: Tùy theo ý muốn người dùng mà có thể **Next** ngay và cài đặt Genymotion ngay tại vị trí được gợi ý hoặc chọn **Browse** để tìm vị trí khác.
- Bước 3: Tiếp tục **Next**.
- Bước 4: Chọn mục **Create a desktop shortcut** để tạo biểu tượng cho phần mềm này trên máy tính, sau đó **Next** tiếp. Chọn vào **Install** để thực hiện cài đặt.
- Bước 5: Rất có thể Genymotion sẽ tự động cài thêm một số phần mềm hỗ trợ, đồng thời hỏi có muốn cài đặt VirtualBox hay không? Nếu máy tính của đã có sẵn phần mềm VirtualBox mới nhất rồi thì thôi, còn nếu không, chọn **Yes** để cài đặt tiếp.
- Bước 6: Tới đây, ngoài cửa sổ cài đặt Genymotion (sẽ tạm thời bị dừng lại), sẽ thấy có thêm một giao diện khác để tiến hành cài đặt VirtualBox.
- Bước 7: Sẽ không có thêm bất cứ phần mềm hay thao tác thiết lập nào khác, nên các có thể yên tâm **Next** tới khi kết thúc > **Next** tiếp nào > Click chọn **Yes**.
- Bước 8: **Install**.
- Bước 9: Bỏ dấu tick ở **Start Oracle WM VirtualBox** và chọn vào **Finish**.

- Bước 10: Lúc này cửa sổ cài đặt cuối cùng của Genymotion xuất hiện, **Finish** để kết thúc quá trình.

## 2.3. Thành phần trong một dự án ANDROID

Bao bọc một Project android là thư mục app và phía dưới là ba thành phần chính của một project trong Android Studio:



Hình 2.3.1. Cấu trúc một project trong Android Studio

### 2.3.1. Tập cấu hình Android<sup>1</sup>

#### a. Giới thiệu

Trong bất kì một project Android nào khi tạo ra đều có một file AndroidManifest.xml, file này được dùng để định nghĩa các screen sử dụng, các permission cũng như các theme cho ứng dụng. Đồng thời nó cũng chứa thông tin về phiên bản SDK cũng như main activity sẽ chạy đầu tiên. File này được tự động sinh ra khi tạo một Android project. Trong file manifest bao giờ cũng có 3 thành phần chính đó là: application, permission và version. Hay nói cách khác đây là file dùng để config những thuộc tính cho ứng dụng của mà khi ứng dụng khởi chạy hệ điều hành có thể hiểu được và xử lí.

#### b. Tác dụng của AndroidManifest

- 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

<sup>1</sup> AndroidManifest

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.

### Cấu trúc tệp AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest>
  <uses-permission /> <permission /> <permission-tree
/> <permission-group />
  <instrumentation />
  <uses-sdk /> <uses-configuration /> <uses-feature />
  <supports-screens />
  <compatible-screens />
  <supports-gl-texture />
  <application>
    <activity>
      <intent-filter> <action /> <category /> <data /> </intent-
filter>
      <meta-data />
    </activity>
    <activity-alias>
      <intent-filter> . . . </intent-filter>
      <meta-data />
    </activity-alias>
    <service>
      <intent-filter> . . . </intent-filter>
      <meta-data/>
    </service>
    <receiver>
      <intent-filter> . . . </intent-filter>
      <meta-data />
    </receiver>
    <provider>
      <grant-uri-permission />
      <meta-data />
      <path-permission />
    </provider>
    <uses-library />
  </application>
</manifest>
```

### c. Một số thẻ trong **AndroidManifest.xml**

`<uses-permission>...</uses-permission/>`: đây là cú pháp lệnh để xin cấp một quyền gì đó khi ứng dụng tương tác với dữ liệu nào đó mà google không cho phép dùng tùy ý.

`<activity>...</activity>`: đây là thẻ để khai báo các activity trong ứng dụng và các thuộc tính của activity đó, nếu như không khai báo thì khi khởi chạy ứng dụng sẽ lỗi ngay.

`<service>...</service>`: đây là một dịch vụ chạy ngầm trên ứng dụng của kể cả khi ứng dụng đã tắt đi và khi sử dụng phải khai báo chúng trong **AndroidMainifest**.

`<receiver>...</receiver>`: đây là khai báo khi sử dụng Broadcast Receiver<sup>1</sup>, cái này là để lắng nghe các sự kiện thay đổi của hệ thống như khi bắt wifi, khi sạc pin...

`<uses-sdk>...</uses-sdk>`: Thẻ xác định phiên bản SDK

#### 2.3.2. Thư mục **Java**

Đây chính là nơi chứa các gói<sup>2</sup> của dự án, có thể tạo các gói ở đây và bên trong là các class.

#### 2.3.3. Thư mục **Res**

**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.

**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.

**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.

**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.

---

<sup>1</sup> Một trong các lớp của Android

<sup>2</sup> Package

- **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.
- **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.

#### 2.3.4. Tập Grade Scripts

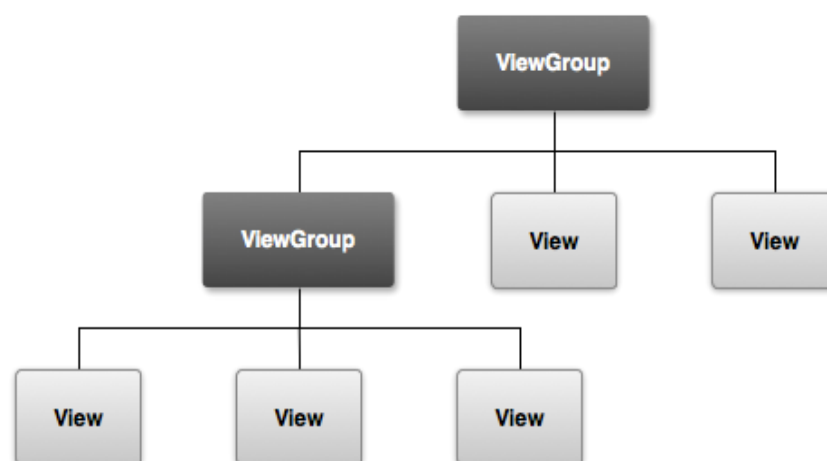
**Build.grade** là file để thiết lập các thuộc tính cho dự án android như: phiên bản SDK, Version ứng dụng, package, thêm thư viện ngoài...

### 2.4. Thành phần giao diện

#### 2.4.1. View group

##### a. Khái niệm

Một ViewGroup là một đối tượng được sử dụng để chứa các đối tượng View và ViewGroup khác để tổ chức và kiểm soát layout của một màn hình. Các đối tượng ViewGroup được sử dụng cho việc tạo ra một hệ thống phân cấp của các đối tượng View (xem bên dưới) do đó có thể tạo các layout phức tạp hơn.



Hình 2.4.1.1. Sơ đồ phân cấp các thành phần giao diện



## **b. Một số view group cơ bản**

- **LinearLayout**: Tồn tại để hiển thị các phần tử theo một thứ tự xếp chồng lên nhau theo chiều ngang hoặc chiều dọc. **LinearLayout** cũng có thể được sử dụng để gán **weight** cho các phần tử **View** con để các phần tử được cách khoảng trên màn hình theo tỉ lệ tương ứng với nhau.
- **RelativeLayout**: Lớp con này của **ViewGroup** cho phép hiển thị các phần tử trên màn hình tương đối với nhau, cung cấp nhiều tính linh hoạt hơn và tự do trong cách layout của xuất hiện so với **LinearLayout**.
- **FrameLayout**: Được thiết kế để hiển thị một **View** con tại một thời điểm, **FrameLayout** vẽ các phần tử trong một ngăn xếp và cung cấp một cách đơn giản để hiển thị một phần tử trên các kích cỡ màn hình khác nhau.
- **ScrollView**: Một lớp mở rộng của **FrameLayout**, lớp **ScrollView** xử lý việc cuộn các đối tượng con của nó trên màn hình.
- **RecyclerView**: Lớp **RecyclerView** là một lớp con của **ViewGroup**, nó liên quan đến các lớp **ListView** và **GridView** và nó được cung cấp bởi Google thông qua thư viện hỗ trợ **RecyclerView** cho các phiên bản **Android** cũ hơn. Lớp **RecyclerView** đòi hỏi việc sử dụng các mẫu thiết kế view holder để tái sử dụng phần tử một cách có hiệu quả và nó hỗ trợ việc sử dụng một **LayoutManager**, một thành phần trang trí, và một phần tử động để làm cho thành phần này vô cùng linh hoạt và đơn giản.
- **CoordinatorLayout**: Được thêm gần đây vào thư viện hỗ trợ thiết kế, lớp **CoordinatorLayout** sử dụng một đối tượng **Behavior** để xác định cách các phần tử **View** con sẽ được sắp xếp và di chuyển khi người dùng tương tác với ứng dụng của .

### **2.4.2. View**

#### **a. Khái niệm**

Trong một ứng dụng **Android**, giao diện người dùng được xây dựng từ các đối tượng **View** và **ViewGroup**. Có nhiều kiểu **View** và **ViewGroup**. Tất cả các kiểu đó được gọi là các **Widget**. Tất cả mọi widget đều có chung các thuộc tính cơ bản như là cách trình bày vị trí, background, kích thước, lề,... Tất cả những thuộc tính chung này được thể hiện hết ở trong đối tượng **View**. Trong **Android Platform**, các screen luôn được bố trí theo một kiểu cấu trúc. Một screen là một tập hợp các **Layout** và các widget được bố trí có thứ tự.

#### **b. Một số view cơ bản trong android**

<b>Tên widget</b>	<b>Chức năng</b>
<b>TextView</b>	Cho phép người dùng hiển thị một đoạn văn bản lên màn hình mà không cho phép người dùng sửa nó.
<b>EditText</b>	Cho phép người dùng nhập, xóa, sửa một đoạn văn bản vào trong đó. Có nhiều dạng EditText khác nhau như: Plaintext, Person Name, Password, Email, Phone,...
<b>Button</b>	Dùng để thiết lập các sự kiện khi người dùng thao tác với nó.
<b>ImageButton</b>	Là dạng nút bấm nhưng có thể chèn thêm hình ảnh vào để giao diện thêm sinh động, trực quan hơn.
<b>CheckBox</b>	Một dạng nút bấm đặc biệt chỉ có hai trạng thái là check và uncheck.
<b>ToggleButton</b>	Có thể xem nó như một Checkbox và có kèm theo hiệu ứng ánh sáng bật/tắt thể hiện trạng thái Check/Uncheck.
<b>RadioButton</b>	Chọn một trong các Radio
<b>Spinner</b>	Là view thể hiện quá trình xử lý diễn ra bên dưới ứng dụng, tạo cảm giác thực cho người dùng. Có hai dạng progressBar là Large và Horizontal
<b>ListView</b>	Chỉ cho phép chọn một trong nhóm lựa chọn.
<b>GridView</b>	Là view được hiển thị dưới dạng lưới gồm nhiều item con bên trong và ta có thể tùy chỉnh nội dung cũng như các đối tượng nằm bên trong item một cách tùy ý.
<b>ViewFlipper</b>	Cho phép định nghĩa một tập hợp nhiều View nhưng chỉ có một View hiển thị tại một thời điểm và hỗ trợ hiệu ứng chuyển đổi giữa các View
<b>QuickContactBage</b>	Hiển thị hình ảnh gắn liền với một đối tượng bao gồm số điện thoại, tên, email đồng thời hỗ trợ việc gọi, nhắn tin sms, email hay tin nhắn tức thời (IM - instant message).
<b>ImageView</b>	Hiển thị hình ảnh
<b>Switch</b>	Tồn tại hai trạng thái đóng mở
<b>ProgressBar</b>	Thể hiện tiến trình, mức độ

### c. Thuộc tích cơ bản widget

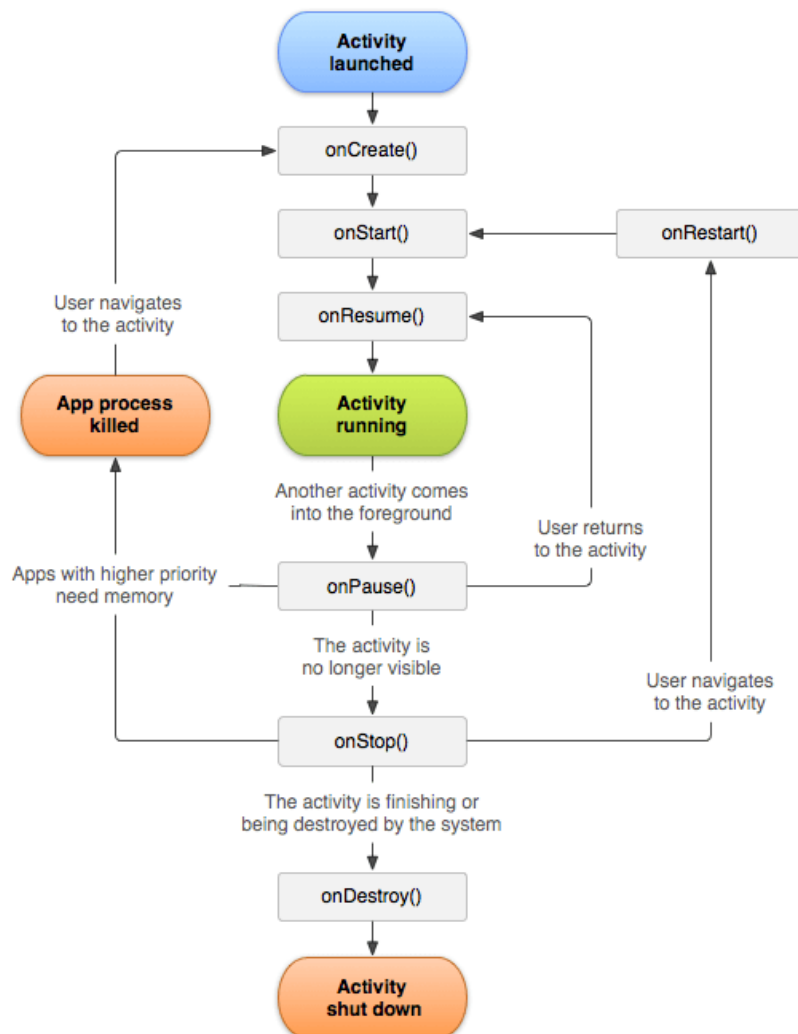
<b>Tên thuộc tính</b>	<b>Công dụng</b>
<b>layout_width</b>	Chiều rộng của control (Tính theo trục Ox từ góc trên bên trái qua phải). Có ba thuộc tính: wrap_content (Co dẫn theo nội dung control), match_parent (kích thước chiều ngang bằng kích thước đối tượng chứa nó)
<b>layout_height</b>	Chiều cao của control (Tính theo trục Oy từ góc trên bên trái xuống dưới). Có ba thuộc tính như <b>layout:width</b>
<b>text</b>	Nội dung hiển thị
<b>textcolor</b>	Màu chữ
<b>background</b>	Màu nền của view
<b>id</b>	Định danh của view
<b>gravity</b>	Căn lề cho nội dung của view
<b>textStyle</b>	Thiết lập kiểu dáng chữ: đậm, in nghiêng, gạch chân
<b>textSize</b>	Thiết lập cỡ chữ
<b>fontFamily</b>	Thiết lập họ phông chữ
<b>inputType</b>	Thiết lập kiểu nhập dữ liệu
<b>hint</b>	Đoạn văn bản gợi ý cho người dùng biết về chức năng hay ràng buộc gì đó...

## 2.5. Vòng đời ứng dụng android

- Các Activity trong hệ thống được quản lý như một ngăn xếp activity (activity stack). Khi một activity mới bắt đầu nó được đặt lên đầu của ngăn xếp và trở thành Running Activity (activity đang chạy), đồng thời activity trước đó sẽ nằm ngay phía dưới trong ngăn xếp đó, và sẽ không trở nên visible (nhìn thấy) cho đến khi activity ở trên thoát ra khỏi ngăn xếp.
- Một Activity gồm bốn trạng thái chính:
  - Nếu activity ở phía trên của màn hình (hay ở trên cùng của ngăn xếp), thì nó đang ở trạng thái active (hoạt động) / running (đang chạy). Ví dụ khi ta cần gọi điện thì activity bấm số đó đang ở trạng thái active.
  - Nếu activity không thể tương tác nhưng vẫn nhìn thấy (khi mà bị che bởi một activity khác nhưng người dùng vẫn có thể nhìn thấy nó ở phía sau) thì activity này đang ở trạng thái paused (tạm dừng). Khi ở trạng thái này activity có thể bị xóa bỏ bởi hệ thống khi thiết bị thiếu bộ nhớ. Ví dụ khi có một activity khác dạng dialog hiện lên chỉ che đi một phần của activity hiện tại thì activity vào trạng thái paused.
  - Nếu activity hoàn toàn bị che khuất bởi activity khác thì nó đang ở trạng thái stopped (đã dừng). Activity này vẫn giữ được tất cả trạng thái và thông

tin, nhưng không còn hiển thị với người dùng và thường xuyên bị xóa bỏ bởi hệ thống khi thiếu bộ nhớ. Ví dụ khi ta tắt màn hình thì khi đó activity vào trạng thái stopped.

- Nếu activity ở trạng thái paused (tạm dừng) hay stopped (đã dừng), hệ thống có thể xóa bỏ activity đó khỏi bộ nhớ bằng cách yêu cầu nó tự kết thúc hoặc xóa bỏ tiến trình của nó. Khi activity đó hiển thị lại với người dùng thì sẽ được khởi tạo lại và khôi phục lại trạng thái trước đó.



Hình 2.5.1. Vòng đời của một Activity

## 2.6. Lớp Intent

### 2.6.1. Khái niệm

**Intent** là một thành phần quan trọng trong android. **Intent** cho phép các thành phần ứng dụng có thể yêu cầu các hàm từ các thành phần ứng dụng android khác.

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

Một đối tượng **Intent** có thể chứa dữ liệu thông qua một đối tượng của lớp **Bundle**. Dữ liệu này có thể được sử dụng bởi các thành phần tiếp nhận.

## 2.6.2. Các loại Intent

Android hỗ trợ hai loại Intent là Intent tường minh (explicit) và Intent không tường minh (implicit).

Một ứng dụng có thể xác định thành phần mục tiêu một cách trực tiếp vào Intent (mục tiêu được yêu cầu là rõ ràng) hoặc yêu cầu hệ thống Android đánh giá các thành phần đã đăng ký trên dữ liệu đích để chọn ra một cái để gửi yêu cầu đến (Intent không tường minh).

**Intent tường minh (Explicit intents):** Là những ý định (intent) chỉ định rõ ràng tên của các thành phần mục tiêu để xử lý; trong đó, trường mục tiêu (tùy chọn) được sét một giá trị cụ thể thông qua các phương thức *setComponent()* hoặc *setClass()*.

**Ví dụ tạo Intents tường minh:**

```
//Màn hình 1:
//cách 1:
//Sử dụng Bundle
Intent intent1 = new Intent(this, ManHinh2Activity.class);
Bundle bundle = intent1.getExtras();
bundle.putString("khoa1", "Giá trị 1");
bundle.putString("khoa1", "Giá trị 2");
//cách 2:
//Sử dụng Bundle
Intent intent2 = new Intent(this, ManHinh2Activity.class);
Bundle bundle2 = new Bundle();
bundle2.putString("khoa1", "Giá trị 1");
bundle2.putString("khoa2", "Giá trị 2");
intent2.putExtra(bundle2);
//-----
//màn hình 2:
Intent intent = this.getIntent();
String str_giatri1 = intent.getStringExtra("khoa1");
String str_giatri2 = intent.getStringExtra("khoa2");
```

**Intent không tường minh (Implicit Intents):** Là những ý định (intent) không chỉ định rõ một mục tiêu thành phần, nhưng bao gồm đầy đủ thông tin cho hệ thống để xác định các thành phần có sẵn là tốt nhất để chạy cho mục đích đó. Hãy xem xét một ứng dụng liệt kê các nhà hàng có sẵn ở gần . Khi bấm vào một tùy chọn nhà hàng cụ thể, ứng dụng sẽ hỏi một ứng dụng khác để hiển thị các tuyến đường đến nhà hàng đó. Để đạt được điều này, nó có thể gửi một ý định rõ ràng trực tiếp đến các ứng dụng **Google Maps**, hoặc gửi ý định ngầm, ý định sẽ được chuyển giao cho bất kỳ ứng dụng nào cung cấp các tính năng bản đồ (map) (chẳng hạn, **Yahoo Maps**).

### Ví dụ Intents không tường minh:

```
//Một Intent không tường minh, yêu cầu xem một URL
//Một đường dẫn URL vốn là một thứ được xem trên trình duyệt
Intent intent = new Intent(Intent.ACTION_VIEW,
Uri.parse("//google.com"));
startActivity(intent);
```

### c. Truyền dữ liệu giữa các activities

- Truyền dữ liệu đến thành phần đích

Để truyền dữ liệu cho intent, ta dùng phương thức **putExtras()**. Extras là **một cặp key/value**. key luôn luôn là kiểu string. value có thể sử dụng kiểu dữ liệu nguyên thủy hoặc đối tượng của String, Bundle, ....

Thành phần tiếp nhận có thể lấy lại được đối tượng intent thông qua hàm **getIntent()**. Để lấy ra được dữ liệu, tùy thuộc vào kiểu dữ liệu truyền đi, sử dụng các phương thức **getStringExtra()**, **getIntExtra()**;

Ví dụ:

```
Intent intent = new Intent(MainActivity.this,
Main2Activity.class);
Intent.putExtra("name", "My name");
Intent intent = getIntent();
String hoTen = intent.getStringExtra("name");
```

Ta có thể sử dụng đối tượng **Bundle**. Đóng gói tất cả dữ liệu vào trong Bundle xong truyền Bundle cho Intent, ví dụ:

```
Intent mIntent = new Intent(MainActivity.this,
Main2Activity.class);
Bundle mBundle = new Bundle();
mBundle.putString(key, value);
mIntent.putExtras(mBundle);
-----
String value = getIntent().getExtras().getString(key);
```

- Sử dụng Intent chia sẻ

Có rất nhiều ứng dụng android cho phép chia sẻ dữ liệu với những người khác, ví dụ: facebook, G+, Gmail... có thể gửi dữ liệu tới một vài thành phần nào đó. Ví dụ dưới đây sẽ mô tả việc sử dụng intent như vậy.

```
Intent intent = new Intent(Intent.ACTION_SEND);
intent.setType("text/plain");
intent.putExtra(android.content.Intent.EXTRA_TEXT, "News for
you!");
startActivity(intent);
```

- Lấy lại kết quả sử dụng từ sub-activity

Một activity có thể được đóng lại thông qua button back trên điện thoại. Trong trường hợp đó, hàm **finish()** sẽ được thực thi. Nếu activity đã được khởi chạy cùng với hàm **startActivity(intent)**, người gọi không cần thiết phải có kết quả hoặc phản hồi từ activity mà có thể close ngay lập tức.

Nếu start một activity cùng với hàm **startActivityForResult()**, như vậy là mong muốn có phản hồi từ **sub-activity**. Khi một sub-activity kết thúc, hàm **onActivityResult()** trên sub-activity sẽ được gọi và có thể thực hiện hành động dựa trên kết quả.

Trong hàm **startActivityForResult()**, có thể xác định hành động sẽ khởi chạy. Hành động này sẽ mong muốn nhận về kết quả. Một hành động đã được khởi chạy cũng có thể thiết lập một đoạn mã code mà người gọi có thể sử dụng để xác định hành động sẽ được cancel hoặc not.

Sub-activity sử dụng hàm **finish()** để khởi tạo một intent mới và truyền dữ liệu cho nó. Nó cũng thiết lập kết quả thông qua hàm **setResult**

```
public void onClick(View view) {
    Intent i = new Intent(this,ActivityTow.class);
    i.putExtra("value1", "This value one for activityTow");
    i.putExtra("Value2", "This value two ActivityTwo");
    // set the request code to any code you like,
    // you can identify the callback via this code
    startActivityForResult(i, REQUEST_CODE);
}
```

Nếu sử dụng hàm **startActivityForResult()**, thì sau khi start activity sẽ gọi đến sub-activity

Nếu hàm sub-activity kết thúc, nó sẽ gửi dữ liệu đến cái đã gọi nó thông qua intent. điều này được xử lý trên hàm **finish()**

```
@Override
public void finish() {
    // Prepare data intent
    Intent data = new Intent();
    data.putExtra("returnKey1", "Swinging on a star. ");
    data.putExtra("returnKey2", "You could be better then you
are. ");
    // Activity finished ok, return the data
    setResult(RESULT_OK, data);
    super.finish();
}
```

Một sub-activity kết thúc, hàm `onActivityResult()` trong activity cha sẽ được gọi

```
@Override
protected void onActivityResult(int requestCode, int resultCode,
Intent data) {
    if (resultCode == RESULT_OK && requestCode == REQUEST_CODE)
    {
        if (data.hasExtra("returnKey1")) {
            Toast.makeText(this,
            data.getExtras().getString("returnKey1"),
            Toast.LENGTH_SHORT).show();
        }
    }
}
```

## 2.7. Share preferences

### 2.7.1. Khái niệm

Đây là một class **Interface** cho phép lưu trữ và đọc dữ liệu với bằng các cặp key và value và nó được lưu dưới dạng một file xml, dữ liệu nó có thể lưu là ở dạng nguyên thủy như: int, float, string, boolean, long. Dữ liệu của Shared Preferences sẽ được lưu ở trong ứng dụng android luôn chính vì thế nếu các xóa ứng dụng đi hoặc là xóa dữ liệu app thì dữ liệu này sẽ hoàn toàn bị biến mất.

### 2.7.2. Cách sử dụng

#### a. Lưu dữ liệu xuống Shared Preferences

Đầu tiên, để lưu dữ liệu xuống thì sẽ khởi tạo một đối tượng (biến) thuộc kiểu **Shared Preferences**, với cú pháp như sau:

```
SharedPreferences sharedPreferences = getSharedPreferences
(SHARED_PREFERENCES_NAME, Context.MODE_PRIVATE );
```

**SHARED\_PREFERENCES\_NAME**: là tên của file **Shared Preferences**.

**Context.MODE\_PRIVATE**: đây là một chế độ bảo mật dữ liệu trong android, khi để như vậy có nghĩa là chỉ cho ứng dụng hiện tại truy cập vào file Shared Preferences này thôi mà không một ứng dụng nào có quyền truy cập vào được.

Tạo đối tượng Editor từ biến `sharedPreferences` đã tạo ở trên, mục đích của thằng này là để có thể mở file **shared\_preferences\_name.xml** ra và đưa dữ liệu vào, cú pháp như sau:

```
SharedPreferences.Editor editor = sharedPreferences.edit();
```

Đưa dữ liệu vào:

```
editor.putX(String key,value);
```

Sau khi đã đặt dữ liệu xong thì sẽ gọi hàm **commit()** hoặc **apply()** để xác nhận những thay đổi.



## b. Cách lấy dữ liệu từ Shared Preferences

Đây là cách lấy ra tất cả các dữ liệu mà ở trên đã lưu:

```
SharedPreferences sharedPreferences =
getSharedPreferences(SHARED_PREFERENCES_NAME,
Context.MODE_PRIVATE);

boolean isFirtsLauncher =
sharedPreferences.getBoolean("IS_FIRTS_LAUNCHER", true);

int highScore = sharedPreferences.getInt("HIGH_SCORE", 0);

float mark = sharedPreferences.getFloat("MARK", 0.0f);

long downloadProgress =
sharedPreferences.getLong("DOWNLOAD_PROGRESS", 0);

String name = sharedPreferences.getString("NAME", "");
```

## c. Cách xóa dữ liệu đã lưu xuống Shared Preferences

Xoá đi dữ liệu nào đó đã lưu trong Shared Preferences thì chỉ cần gọi lệnh:

```
editor.remove(String key);
```

Xoá đi hết tất cả giá trị đã lưu thì bạn dùng cú pháp:

```
editor.clear();
```

Gọi editor.apply() hoặc commit() để lưu thay đổi.

## 2.8. Hiệu ứng trong android

### 2.8.1. Hiệu ứng cơ bản

- a. **Fade in:** Là hiệu ứng làm mờ đối tượng.
- b. **Fade out:** Là hiệu ứng làm mờ dần đối tượng.
- c. **Blink:** Là hiệu ứng làm nhấp nháy đối tượng.
- d. **Zoom in:** Là hiệu ứng phóng to đối tượng.
- e. **Zoom out:** Là hiệu ứng thu nhỏ đối tượng.
- f. **Rotate:** Là hiệu ứng xoay đối tượng.
- g. **Move:** Là hiệu ứng dịch chuyển đối tượng.
- h. **Side up:** Là hiệu ứng trượt đối tượng lên trên.
- i. **Side down:** Là hiệu ứng trượt đối tượng xuống dưới.
- j. **Sequential animation:** Lặp lại một trong các hiệu ứng trên.
- k. **Together animation:** Xảy ra đồng thời hai hoặc nhiều trong các hiệu ứng trên.

### 2.8.2. Cách sử dụng

- Bước 1: Tạo tập tin xml định nghĩa hiệu ứng  
(Tập tin này được đặt trong thư mục **anim** dưới thư mục **res** (res ⇒ anim ⇒ xml))
- Bước 2: Nạp hiệu ứng

Tại activity tạo một đối tượng của lớp Animation và nạp tập tin xml sử dụng phương thức **loadAnimation()** của lớp **AnimationUtils**

– Bước 3: Thiết lập sự kiện (Tùy chọn)

Nếu bạn muốn xử lý các sự kiện như start, end và repeat, bạn phải cài đặt giao diện AnimationListener cho activity của bạn. Lưu ý bước này là tùy chọn, bạn có thể bỏ qua nếu không cần thiết.

– Bước 4: Bắt đầu hiệu ứng

Chúng ta có thể bắt đầu một animation bất cứ khi nào mình muốn bằng cách gọi phương thức **startAnimation()** trên bất kỳ thành phần UI nào mà chúng ta muốn thiết lập animation.

## CHƯƠNG 3: TỔNG QUAN VỀ PHÁT NHẠC TRONG ANDROID

### 3.1. Giới thiệu file nhạc số

Nhạc số (digital music), được hiểu là các bản nhạc được lưu trữ dưới dạng số trong các thiết bị nghe nhìn như máy vi tính, điện thoại di động, các thiết bị nghe nhạc cầm tay... Và là vì lưu trữ dưới dạng số, nên nhạc số còn có thể được sao chép, truyền tải trực tiếp giữa các thiết bị số hay qua mạng máy tính.

Các bản nhạc số thường được thu âm trong các studio với khá nhiều thiết bị phức tạp. Nhưng về bản chất, thu âm là quá trình chuyển đổi sóng âm thanh thành tín hiệu số. Như đã biết, “âm thanh thực” là các sóng cơ học, có dạng hình sin tuần hoàn liên tục (analog), trong khi “âm thanh số” chỉ là những xung điện tử rời rạc (digital). Do đó, bằng những mẫu rời rạc, âm thanh số chỉ có thể mô phỏng một cách gần giống nhất với âm thanh thực tế. Việc mô phỏng đó được đặc trưng bởi các thông số sau:

- **Sample:** Là thành phần nhỏ nhất của bản nhạc số. Để có các xung điện tử rời rạc, cần phải tiến hành rất nhiều lần lấy mẫu. Mỗi mẫu gọi là một sample - là giá trị biên độ của tần số sóng âm tại thời điểm lấy mẫu. Càng lấy nhiều mẫu, tín hiệu số thu được càng chính xác hơn.
- **Sample Rate:** (Sampling Rate, Sampling Frequency): Là số lần lấy mẫu trên một giây, có đơn vị Hz. Một bản nhạc có sample rate là 44100 Hz thì mỗi giây nhạc sẽ được lấy mẫu 44100 lần.
- **BitDepth:** Để lưu lại dưới dạng số, mỗi mẫu được biểu diễn bằng một lượng bit dữ liệu nhất định, gọi là BitDepth. Các bản nhạc hiện nay thường có BitDepth là 16 bits, 24 bits... BitDepth càng lớn âm thanh càng sắc nét, trung thực nên nó còn được gọi là Resolution (độ nét).
- **Channel:** Bằng các thuật toán, tín hiệu số sẽ được tách ra thành nhiều kênh (Channel) sao cho khi nghe bằng hệ thống loa thích hợp sẽ có cảm giác như khi đang nghe nhạc trong không gian thực tế.

Từ 4 thông số cơ bản trên, ta không những biết được chất lượng mà còn có thể tính được dung lượng của bản nhạc. Ví dụ một phút của bản nhạc có: Sample rate = 44100 Hz, BitDepth = 16 bits = 2 bytes, Channel = 2 kênh sẽ có dung lượng:  $44100 \text{ đọt lấy mẫu} \times 2 \text{ bytes} \times 60 \text{ giây} \times 2 \text{ kênh} = 10.584.000 \text{ bytes}$ , tức khoảng 10.1 MB

- **BitRate:** Là thông số thu gọn, đại diện cơ bản cho các thuộc tính trên. Bitrate có đơn vị Kbps (Kilobits per second) - dung lượng (tính theo bit) của âm thanh số trên một giây. Với Bitrate, ta có thể xác định nhanh chóng dung lượng cũng như phần nào chất lượng của bản nhạc. Một phút nhạc 128 kbps có dung lượng khoảng 1 MB và bản nhạc 320 kbps thì chắc chắn sẽ hay hơn bản nhạc 128 kbps.

### 3.2. Một số định dạng file nhạc số phổ biến

Sau quá trình thu âm, ta được một file nhạc wav có chất lượng nguyên gốc nhưng dung lượng rất lớn, khoảng 10 MB cho mỗi phút nhạc. Bởi vậy, để tiện việc lưu trữ hay chia sẻ, người ta phải nén các bản nhạc lại dưới các định dạng. Mỗi định dạng ứng với một thuật toán nén nhất định và tỉ lệ nén cũng như chất lượng sau khi nén của bản nhạc cũng khác nhau. Có hai cách nén chính là: nén có mất (lossy compression) tạo ra các file nhạc mp3, wma, ogg... và nén không mất (lossless compression) để tạo các file nhạc flac, ape... Khi nén có mất (rip hoặc convert nhạc), chương trình nén sẽ cắt bớt đi những dải tần số âm thanh nhất định (thường là dải tần trên 20 Khz, theo đặc điểm về khả năng nghe của tai người), từ đó giảm được dung lượng bản nhạc. Nhưng cái phải trả giá là chất lượng âm thanh sẽ giảm đi. Do vậy, càng giảm ít thì định dạng nhạc hay thuật toán mã hóa càng tốt.

Theo nhiều đánh giá thì định dạng ogg, wma cho chất lượng tốt hơn mp3 với cùng một dung lượng. Và việc chuyển đổi qua lại các định dạng cũng làm giảm chất lượng bản nhạc.

#### 3.2.1. Định dạng file nhạc số MP3

MP3 là một dạng file đã được nén bằng cách nén dữ liệu có tổn hao (lossy)[6]. Nó là một dạng âm thanh được mã hóa PCM pulse-code modulation và có dung lượng nhỏ hơn rất nhiều so với dữ liệu ban đầu do nó bỏ đi những phần âm thanh được cho là không quan trọng trong khoảng nghe được của con người, tương tự như cách nén JPEG dành cho hình ảnh.

##### a. Đặc điểm

Tên của định dạng mp3 bắt nguồn từ "MPEG-1, layer 3", còn được gọi chính thức hơn là ISO/IEC 11172 - 3 lớp ba. Những tập tin theo dạng này được lưu với phần mở rộng tên .mp3. Đôi khi những tập tin theo tiêu chuẩn MPEG-2, layer 3 cũng sử dụng phần mở rộng này.

Có rất nhiều kỹ thuật đã được dùng trong chuẩn nén MP3 để xác định phần nào nên bỏ đi, trong đó có tâm thần âm học psychoacoustic. Dữ liệu MP3 có thể được tạo ra với nhiều bitrate khác nhau để có thể dễ dàng chọn lựa giữa chất lượng cao hay cỡ tập tin đầu ra nhỏ.

Cốt lõi của kỹ thuật nén MP3 là một dạng biến đổi phức để chuyển tín hiệu sóng ngang theo thời gian thành tín hiệu dựa theo tần số:

- Có 32 băng tần phép lọc cầu phương đa âm.
- Có 36 hay 12 nhánh MDCT; kích cỡ có thể được định cho từng băng tần con từ 0 đến 1, từ 2 đến 31.
- Đặt ký hiệu aliasing để làm giảm cỡ tập tin sau nén.

MP3 vòm, một dạng MP3 hỗ trợ 5+1 kênh cho âm thanh vòm, được giới thiệu vào tháng 12 năm 2004. MP3 vòm có tính tương thích ngược với chuẩn MP3 trước đây, và kích cỡ file sau nén cũng tương tự.

Sử dụng trong MP3 của một thuật toán nén được thiết kế để làm giảm đáng kể số lượng dữ liệu cần thiết để đại diện cho âm thanh ghi âm và vẫn còn âm thanh như một sự sao chép trung thành của âm thanh không nén ban đầu cho hầu hết các thính giả. Một tập tin MP3 được tạo ra bằng cách sử dụng các thiết lập của 128 kbit / s sẽ cho kết quả trong một tập tin đó là khoảng 1/11 kích thước lưu ý của file CD tạo ra từ nguồn âm thanh ban đầu. Một tập tin MP3 cũng có thể được xây dựng với tốc độ bit cao hơn hoặc thấp hơn, với chất lượng kết quả cao hơn hoặc thấp hơn.

Nén hoạt động bằng cách làm giảm độ chính xác của các bộ phận nhất định của âm thanh được coi là vượt quá khả năng giải quyết thính giác của hầu hết mọi người. Phương pháp này thường được gọi là mã hóa tri giác sử dụng mô hình tâm lý học để loại bỏ hoặc làm giảm độ chính xác của các thành phần này ít nghe điều trần của con người, và sau đó ghi lại các thông tin còn lại một cách hiệu quả.

## **b. Lịch sử phát triển**

Lossy MP3 - âm thanh nén dữ liệu thuật toán có lợi thế của một giới hạn nhận thức của người nghe gọi là nhĩ mất nạ. Năm 1894, nhà vật lý người Mỹ Alfred Marshall Mayer đã báo cáo rằng một giai điệu có thể được trả lại không nghe được bởi một âm thanh tần số thấp hơn. Năm 1959, Richard Ehmer mô tả một bộ hoàn chỉnh các đường cong thính giác liên quan đến hiện tượng này. Ernst Terhardt, tạo ra một thuật toán mô tả mất nạ thính giác với độ chính xác cao. Công việc này được thêm vào một loạt các báo cáo từ các tác giả trở lại Fletcher.

Những người tiên nhiệm của MP3 "mã hoá tối ưu trong miền tần số" (OCF), và Perceptual chuyển đổi Coding (PXF). Hai codec, cùng với khối chuyển đổi đóng góp từ Thomson-Brandt, đã được sáp nhập vào một codec được gọi là ASPEC, đã được gửi sang MPEG, và giành được sự cạnh tranh chất lượng, nhưng bị nhầm lẫn từ chối là quá phức tạp để thực hiện. Là người đầu tiên thực tế thực hiện của một coder cảm nhận âm thanh (OCF) trong phần cứng (phần cứng Krasner là quá cồng kềnh và chậm cho sử dụng thực tế), là một thực hiện biến đổi của tâm lý học dựa trên Motorola 56.000 chip DSP.

Là một tiến sĩ tại Đại học Erlangen-Nuremberg, Đức. Karlheinz Brandenburg đã bắt đầu làm việc nén nhạc kỹ thuật số đầu những năm 1980, tập trung vào cách người ta cảm nhận âm nhạc. "Diner Tom" - bài hát của Suzanne Vega là bài hát đầu tiên được sử dụng bởi Karlheinz Brandenburg để phát triển các bản MP3. Brandenburg đã thông qua các bài hát cho mục đích thử nghiệm, nghe nó mỗi lần tinh chỉnh trong các

chương trình, đảm bảo nó không gây ảnh hưởng xấu đến sự tinh tế trong tiếng hát của Vega.

Nén hiệu quả của bộ mã hóa thường được định nghĩa bởi tỷ lệ bit, bởi vì tỉ lệ nén phụ thuộc vào độ sâu bit và tỷ lệ lấy mẫu của tín hiệu đầu vào. Tuy nhiên, họ có thể sử dụng Compact Disc (CD) qua các thông số như tài liệu tham khảo (44,1 kHz, 2 kênh 16 bit cho mỗi kênh hoặc  $2 \times 16$  bit), hoặc đôi khi Digital Audio Tape (DAT) các thông số (48 kHz,  $2 \times 16$  bit). Tỷ lệ nén với tài liệu tham khảo thứ hai cao hơn, chứng tỏ vấn đề với việc sử dụng tỉ lệ nén hạn chế cho các bộ mã hóa lossy.

Karlheinz Brandenburg sử dụng ghi đĩa CD của bài hát "Tom Diner" để đánh giá và tinh chỉnh các thuật toán nén MP3 Suzanne Vega. Bài hát này được chọn vì tính chất gần như đơn âm và nội dung quang phổ rộng, làm cho nó dễ dàng hơn để nghe không hoàn hảo trong định dạng nén trong quá trình phát lại.

Ngày 7 tháng 7 năm 1994, Hiệp hội Fraunhofer phát hành các bộ mã hóa MP3 phần mềm đầu tiên được gọi là Isenc. Các phần mở rộng tên tập tin mp3 đã được lựa chọn bởi nhóm Fraunhofer vào 14 tháng 7 năm 1995 (trước đây, các tập tin đã được đặt tên ".Bit"). Với thời gian thực phần mềm đầu tiên máy nghe nhạc MP3 WinPlay3 (phát hành ngày 09 Tháng 9 năm 1995), nhiều người đã có thể để mã hóa và phát lại các tập tin MP3 trên máy tính của họ.

Trong nửa thứ hai của năm 1994, các tập tin MP3 bắt đầu lan rộng trên Internet. Sự phổ biến của MP3 bắt đầu tăng lên nhanh chóng với sự ra đời của máy nghe nhạc Winamp Nullsoft, phát hành vào năm 1997. Năm 1998, máy nghe nhạc âm thanh kỹ thuật số MPMAN, được phát triển bởi hệ thống thông tin Saehan có trụ sở tại Seoul, Hàn Quốc, đã được phát hành và Rio PMP300 được bán sau đó vào năm 1998, bắt chước những nỗ lực đàn áp pháp lý của RIA.

Trong tháng 11 năm 1997, mp3.com trang web được cung cấp hàng ngàn bài nhạc MP3 được tạo ra bởi các nghệ sĩ độc lập miễn phí. Kích thước nhỏ của các tập tin MP3 cho phép chia sẻ phổ biến rộng rãi từ đĩa CD.

### **c. Thiết kế của mp3**

#### **– Mã hóa âm thanh**

Các tiêu chuẩn MPEG-1 không bao gồm một đặc điểm kỹ thuật chính xác cho một bộ mã hóa MP3. Thực hiện các tiêu chuẩn có nghĩa vụ phải đưa ra các thuật toán của họ phù hợp để loại bỏ các bộ phận thông tin từ đầu vào âm thanh. Kết quả là, có rất nhiều bộ mã hóa MP3 khác nhau có sẵn, mỗi file có một chất lượng khác nhau. Vì vậy nó rất dễ dàng cho một người sử dụng tiềm năng của các bộ mã hóa để nghiên cứu các lựa chọn tốt nhất. Một bộ mã hóa là thành thạo mã hóa ở mức bit cao hơn (như LAME) là không nhất thiết phải là tốt ở mức bit thấp hơn.

Trong quá trình mã hóa, 576 mẫu miền thời gian được ghi lại và được chuyển đến 576 mẫu miền tần số. Nếu có là một thoáng qua, mẫu 192 được lấy thay vì 576. Điều này được thực hiện để hạn chế sự lây lan thời gian của tiếng ồn quantization kèm theo thoáng qua.

– Giải mã âm thanh

Giải mã, mặt khác, là cần thận định nghĩa trong chuẩn. Hầu hết các bộ giải mã là "bitstream tuân thủ", có nghĩa là sản lượng giải nén mà họ sản xuất từ một tập tin MP3 trong một mức độ quy định, như đầu ra quy định toán học trong các tài liệu ISO / IEC tiêu chuẩn cao (ISO / IEC 11.172-3). Vì vậy, so sánh các bộ giải mã thường dựa trên tính toán hiệu quả( ví dụ, bao nhiêu bộ nhớ hoặc CPU thời gian họ sử dụng trong quá trình giải mã).

– Chất lượng âm thanh

Khi thực hiện mã hóa âm thanh, chẳng hạn như việc tạo ra một tập tin MP3. Thông thường, người sáng tạo được cho phép để thiết lập một tỷ lệ bit, trong đó xác định các tập tin có thể sử dụng bao nhiêu kilobits mỗi giây của âm thanh.

Bên cạnh tỷ lệ bit của một phần mã hóa âm thanh, chất lượng của các tập tin MP3 cũng phụ thuộc vào chất lượng của các bộ mã hóa riêng của mình, và những khó khăn của các tín hiệu được mã hóa. Như là tiêu chuẩn MP3, mã hóa khác nhau có thể tính năng chất lượng hoàn toàn khác nhau, ngay cả với tốc độ bit giống hệt nhau. Chất lượng phụ thuộc vào sự lựa chọn của các thông số mã hóa và mã hóa.

Loại đơn giản nhất của tập tin MP3 sử dụng một tỷ lệ bit cho toàn bộ tập tin này được gọi là tốc độ không đổi (CBR) mã hóa. Sử dụng một tốc độ không đổi làm cho mã hóa đơn giản và nhanh hơn. Tuy nhiên, nó cũng có thể tạo ra các tập tin mà tốc độ bit thay đổi trong suốt các tập tin. Chúng được gọi là Variable Bit Rate (VBR). Trong bất kỳ đoạn âm thanh nào, một số phần sẽ dễ dàng để nén hơn nhiều, chẳng hạn như sự im lặng hoặc âm nhạc chỉ có chứa một vài công cụ, trong khi những người khác sẽ khó khăn hơn để nén. Vì vậy, chất lượng tổng thể của tập tin có thể được tăng lên bằng cách sử dụng một tỷ lệ bit thấp hơn cho những đoạn ít phức tạp hơn và một tỷ lệ bit cao hơn cho các bộ phận phức tạp hơn. Với một số bộ mã hóa, nó có thể chỉ định dạng một chất lượng nhất định, và các bộ mã hóa sẽ thay đổi tốc độ bit cho phù hợp. Người dùng biết một "thiết lập chất lượng" cụ thể mà trong suốt đối với đôi tai của họ có thể sử dụng giá trị này khi mã hóa tất cả các âm nhạc của họ, và nói chung không cần phải lo lắng về việc thực hiện các bài kiểm tra cá nhân khi lắng nghe trên mỗi tác phẩm âm nhạc để xác định tỷ lệ bit chính xác.

Chất lượng cảm nhận có thể bị ảnh hưởng bởi môi trường âm nhạc (môi trường xung quanh tiếng ồn), sự chú ý của người nghe, và đào tạo người biết lắng nghe và

trong nhiều trường hợp thiết bị âm thanh nghe (chẳng hạn như card âm thanh, loa và tai nghe).

Một thử nghiệm cho sinh viên mới của trường Đại học Stanford Music. Giáo sư Jonathan Berger cho thấy sinh viên ưu tiên cho chất lượng âm nhạc MP3 đã tăng mỗi năm.

#### – Tốc độ Bit

Một số tốc độ bit được quy định cụ thể trong MPEG-1 Audio III Lớp tiêu chuẩn: 32, 40, 48, 56, 64, 80, 96, 112, 128, 160, 192, 224, 256 và 320 kbit / s, với tần số lấy mẫu có sẵn 32, 44 và 48 kHz. MPEG-2 Audio Layer III cho phép bit tỷ lệ 8, 16, 24, 32, 40, 48, 56, 64, 80, 96, 112, 128, 144, 160 kbit / s với lấy mẫu tần số 16, 22 và 24 kHz. MPEG-2.5 Audio Layer III được giới hạn tốc độ bit của 8, 16, 24, 32, 40, 48, 56 và 64 kbit / s với tần số lấy mẫu 8, 11.025, và 12 kHz. Bởi vì định lý Nyquist / Shannon, tần số sinh sản luôn luôn là một nửa của tần số lấy mẫu, vì vậy 8 kHz lấy mẫu tỷ lệ giới hạn tần số tối đa 4 kHz, trong khi 48 kHz tỷ lệ lấy mẫu tối đa giới hạn MP3 tái tạo âm thanh đến 24 kHz.

Một tỷ lệ mẫu là 44,1 kHz hầu như luôn luôn được sử dụng, bởi vì điều này cũng được sử dụng cho đĩa CD âm thanh, nguồn chính được sử dụng để tạo ra các tập tin MP3. Một loạt lớn hơn tốc độ bit được sử dụng trên Internet. Tỷ lệ 128 kbit / s được sử dụng phổ biến, một tỉ lệ nén 11:01, cung cấp đầy đủ chất lượng âm thanh trong một không gian tương đối nhỏ. Khi Internet băng thông và kích thước ổ đĩa cứng đã tăng lên, tốc độ bit cao hơn lên đến 320 kbit / s được phổ biến rộng rãi.

Âm thanh không nén được lưu trữ trên một đĩa CD âm thanh có tốc độ bit 1,411.2 kbit / s, lưu ý để bitrate 128, 160 và 192 kbit / s đại diện cho tỷ lệ nén khoảng 11:01, 9:1 và 07:01 tương ứng.

Phi tiêu chuẩn tốc độ bit lên đến 640 kbit / s có thể đạt được với bộ mã hóa LAME. Theo tiêu chuẩn ISO, bộ giải mã chỉ yêu cầu để có thể giải mã các dòng lên tới 320 kbit / s.

Các phần mềm chỉ có thể sử dụng một bitrate thống nhất trên tất cả các khung hình trong một tập tin MP3.

#### – VBR

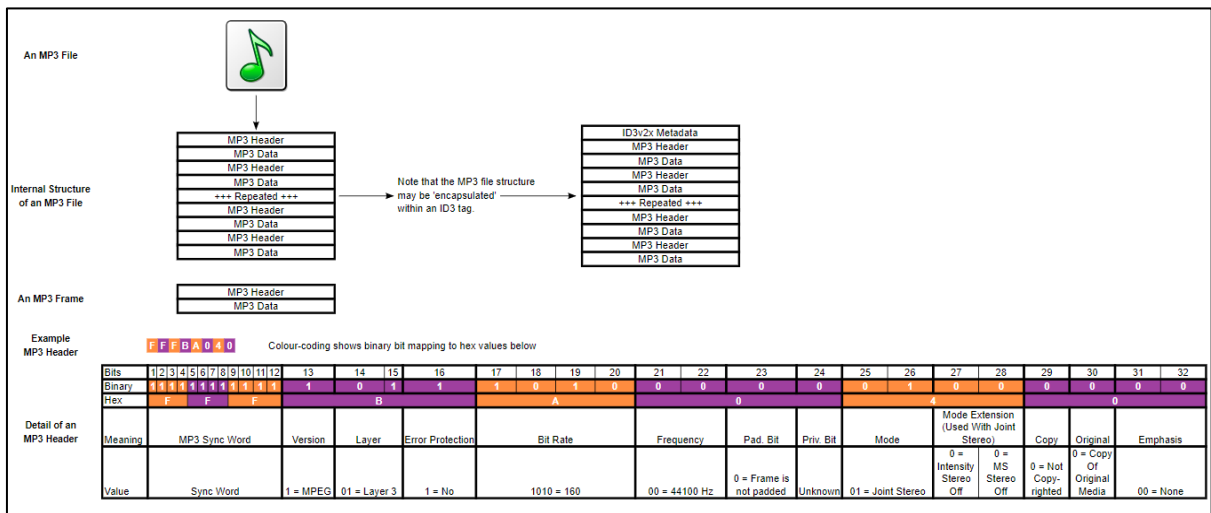
MP3 encoder phức tạp hơn có thể sản xuất âm thanh Variable Bit Rate. MPEG có thể sử dụng bitrate chuyển đổi trên cơ sở mỗi khung hình, nhưng chỉ có lớp III giải mã phải hỗ trợ nó. VBR được sử dụng khi mục tiêu là để đạt được một mức cố định của chất lượng. Kích thước tập tin cuối cùng của một mã hóa VBR khó dự đoán hơn so với bitrate liên tục. Bitrate trung bình là VBR thực hiện như là một thỏa hiệp giữa hai bitrate được phép thay đổi về chất lượng ổn định hơn, cho kích thước file dự đoán. Mặc dù một bộ giải mã MP3 phải hỗ trợ VBR là tiêu chuẩn phù hợp. Lớp III âm



thanh cũng có thể sử dụng một "hộp chứa bit", một khung một phần đầy đủ khả năng nắm giữ một phần của dữ liệu âm thanh của khung hình kế tiếp, cho phép thay đổi tạm thời trong bitrate hiệu quả, ngay cả trong một dòng bitrate không đổi.

– Cấu trúc tập tin

Một tập tin MP3 được tạo thành từ nhiều khung hình MP3, trong đó bao gồm một tiêu đề và một khối dữ liệu. Chuỗi các khung này được gọi là một dòng tiêu. Các khối dữ liệu MP3 chứa các thông tin âm thanh (nén) về tần số và biên độ. Đồ thị cho thấy Header MP3 bao gồm một từ đồng bộ, được sử dụng để xác định sự khởi đầu của một khung hợp lệ. Sau này, các giá trị sẽ khác nhau, tùy thuộc vào các tập tin MP3. ISO / IEC 11.172-3 xác định phạm vi của các giá trị cho mỗi phần của tiêu đề cùng với các đặc điểm kỹ thuật của tiêu đề. Hầu hết các tập tin MP3 hôm nay chứa ID3 siêu dữ liệu, trước hoặc sau các khung MP3.



Hình 3.2.1.1. cấu trúc của một tập MP3

– ID3 và các thẻ khác

Một "tag" trong một tập tin âm thanh là một phần của tập tin có chứa siêu dữ liệu chẳng hạn như tiêu đề, nghệ sĩ, album, số lượng theo dõi hoặc các thông tin khác về nội dung của tập tin. Các tiêu chuẩn MP3 không xác định các định dạng thẻ cho các tập tin MP3, và cũng không có một định dạng container tiêu chuẩn có thể hỗ trợ siêu dữ liệu và loại trừ nhu cầu cho các thẻ.

Tuy nhiên, một số thực tế tiêu chuẩn cho các định dạng tag tồn tại. Năm 2010, phổ biến nhất là ID3v1 và ID3v2, và gần đây hơn là APEv2. Các thẻ này thường được nhúng ở đầu hoặc cuối của các tập tin MP3, tách biệt với khung dữ liệu MP3 thực tế.

**3.2.2. Một số định dạng file nhạc số khác**

**a. Định dạng eAAC+**

eAAC+ được cải tiến từ ACC, được phát triển để nén các tập tin media kỹ thuật số chất lượng cao[5]. eAAC + sử dụng công nghệ nén tiên tiến hơn so với MP3, có

nghĩa là có thể có được chất lượng âm thanh tương tự từ một tập tin nhỏ hơn nhiều (hoặc cách khác, một chất lượng cao từ các tập tin có kích thước tương tự).

### **b. Định dạng FLAC**

**FLAC (Free Lossless Audio Codec)** là một định dạng nén âm thanh lossless (Định dạng âm thanh nén không mất dữ liệu). Một trong những lợi thế của định dạng file này là nó làm giảm kích thước tập tin của một chương trình âm thanh từ 30% đến 40% dung lượng của File âm thanh gốc (số lượng không gian lưu trữ phải mất trên một đĩa hoặc thiết bị khác) mà không ảnh hưởng đến chất lượng âm thanh.

### **c. Định dạng PCM**

**PCM (Pulse Code Modulation)** là định dạng âm thanh tiêu chuẩn cho đĩa CD. Âm thanh PCM là một hệ thống lưu trữ không nén dữ liệu (lossless) nên file có dung lượng lớn, làm chiếm nhiều diện tích ổ cứng của bạn.

### **d. Định dạng WMA**

**WMA (Windows Media Audio)** là chuẩn nén âm thanh được Microsoft sản xuất, dùng cho Windows Media Player, chất lượng tương đương MP3 với bit rate thấp hơn một nửa (và vì vậy dung lượng tập tin cũng giảm một nửa). Đây cũng là công nghệ nén nhạc độc quyền của công nghệ Windows Media. Ngày nay định dạng loại WMA đang là đối thủ cạnh tranh với MP3, Real Audio và AAC mặc dù những thiết bị cho loại này còn khá ít.

### **e. Định dạng WAV**

Các tập tin âm thanh **WAV (Waveform)** tương tự như âm thanh PCM và có thể được mã hóa như âm thanh lossless và WAV cũng tương tự như các file âm thanh AIFF được sử dụng trên các máy tính Mac.

### **f. Định dạng MIDI**

Nhạc thường (bao gồm tiếng đàn, tiếng sáo, tiếng các bộ gõ và cả tiếng hát nữa) đều tồn tại dưới dạng sóng âm thanh với bản chất cơ học: Một luồng sóng âm sẽ làm rung động không khí quanh nó, truyền đi trong không gian, rồi đập vào tai người nghe, làm rung động màng nhĩ, khiến cho người ấy nghe được âm thanh đó. Để ghi lại, lưu lại một sóng âm thường thì người ta sử dụng kỹ thuật tương tự (Analog), biến một sóng âm bản chất cơ học thành sóng điện từ, với những định dạng quen thuộc như .wav, .cda, .mp3 v.v...

Nhạc **Midi (Musical Instrument Digital Interface)** không dùng kỹ thuật tương tự (Analog), mà dùng kỹ thuật số (Digital) để lưu lại âm thanh. Mỗi âm thanh của các nhạc cụ khác nhau được gán cho một chuỗi ký tự số nhị nguyên tương ứng (chỉ bao gồm hai chữ số 0 và 1) chẳng hạn như 010101, hoặc 101010,... Như vậy một chuỗi âm thanh sẽ được ghi lại như một... chuỗi số.

Ở công cụ nghe, một quá trình ngược sẽ được thực thi: Chuỗi số sẽ được biến đổi, hoán cải ngược lại thành chuỗi âm thanh. Vì thế Nhạc Midi còn được gọi bằng những tên khác như nhạc điện tử, hay gọn hơn nữa là nhạc số.

Vì đã được tiêu chuẩn hóa nên nhạc Midi chơi rất chính xác và rất hay, rất lạ tai. Một lợi ích quan trọng hơn nữa là file nhạc Midi chiếm dụng rất ít không gian. Một bài hát định dạng wav 40 Mb, định dạng Mp3 khoảng 4 Mb, thì một file Midi tương ứng chỉ mất khoảng 40 Kb (ít hơn Mp3 một trăm lần, và ít hơn Wav một ngàn lần).

#### **g. Định dạng OGG**

**OGG** là một file audio nén sử dụng chuẩn nén Ogg Vorbis miễn phí tương tự chuẩn MP3 nhưng cho chất lượng tốt hơn với kích cỡ file tương đương. File chứa siêu dữ liệu của bài hát bao gồm thông tin nhạc sỹ và bài hát.

#### **h. Định dạng AC-3**

Khi chuyển từ công nghệ thu thanh thường sang kỹ thuật số, âm thanh được chứa trong những file có kích thước lớn, trong khi đó một đĩa CD chỉ cho 70 phút âm nhạc, công ty Dolby đã phát triển cách số hóa âm thanh mới (digital audio coding). **AC-3** là tên gọi của công nghệ này đòi thứ ba cho phép số hóa (coding) âm thanh với tần số thấp hơn (lower sample rate) nhưng vẫn bảo đảm được chất lượng âm thanh. Vì vậy dung lượng các file âm thanh AC-3 giảm thiểu đáng kể. Ngoài ra công nghệ này còn hỗ trợ giảm nhiễu âm thanh hiệu quả hơn hẳn bằng cách chia âm thanh ra các dải tần nhỏ khác nhau sau đó mới lọc nhiễu từng phần.

#### **j. Định dạng AMR**

**AMR** là một định dạng âm thanh nén thường được dùng cho ghi âm, nhạc chuông. Đây cũng là định dạng âm thanh được sử dụng khi đàm thoại trên qua điện thoại, được sử dụng làm codec âm thanh đàm thoại tiêu chuẩn bởi 3GPP vào tháng 10 năm 1998.

### **3.3. Chỉnh sửa thông tin file nhạc**

Khi sưu tầm các file nhạc từ nhiều nguồn khác nhau các thông tin về file nhạc như: tên bài hát, tên ca sĩ, tên tác giả, ảnh ca sĩ có thể chưa chính xác. Để chuẩn lại những thông tin này người dùng có thể sử dụng phần mềm Mp3Tag để chỉnh sửa lại. Phần mềm Mp3Tag là chương trình nhỏ gọn được thiết kế với các tính năng ưu việt nhằm hỗ trợ người sử dụng tạo và thêm các thẻ thông tin cho file nhạc MP3 để quản lý các thông tin đó được hợp lý. Phần mềm Mp3Tag có khả năng tạo và thêm những phần thông tin còn thiếu trong thẻ MP3 của người sử dụng như: tiêu đề, nghệ sỹ, album, năm, thể loại, đánh giá hay phê bình... Công cụ này giúp người dùng có được những thông tin đầy đủ nhất về bài hát nằm trong bộ sưu tập của mình cũng như cần thiết cho công việc sắp xếp và phân loại các bài hát theo nhiều tiêu chí.

### 3.3.1. Tải về và cài đặt

- Bước 1:

Link download: <https://www.mp3tag.de/en/download.html>

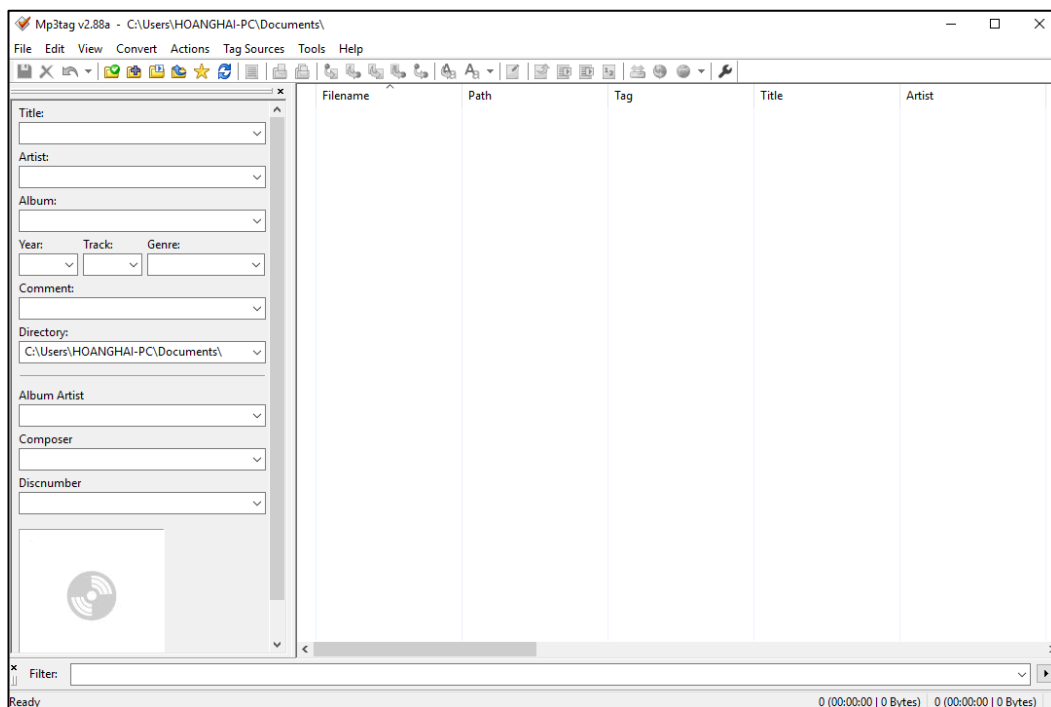
- Bước 2:

Mở file cài đặt mp3tagsetup.exe để tiến hành cài đặt.



Hình 3.3.1.1. Giao diện khi chạy file cài đặt

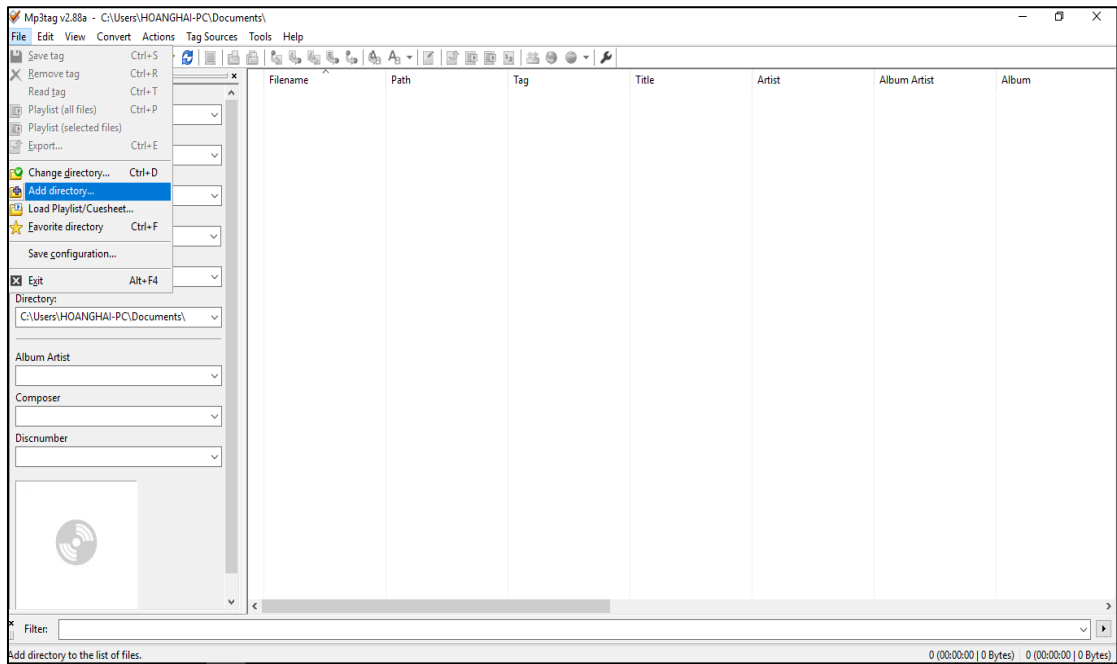
- Bước 3: Khi cài đặt hoàn tất, chạy chương trình mp3tag



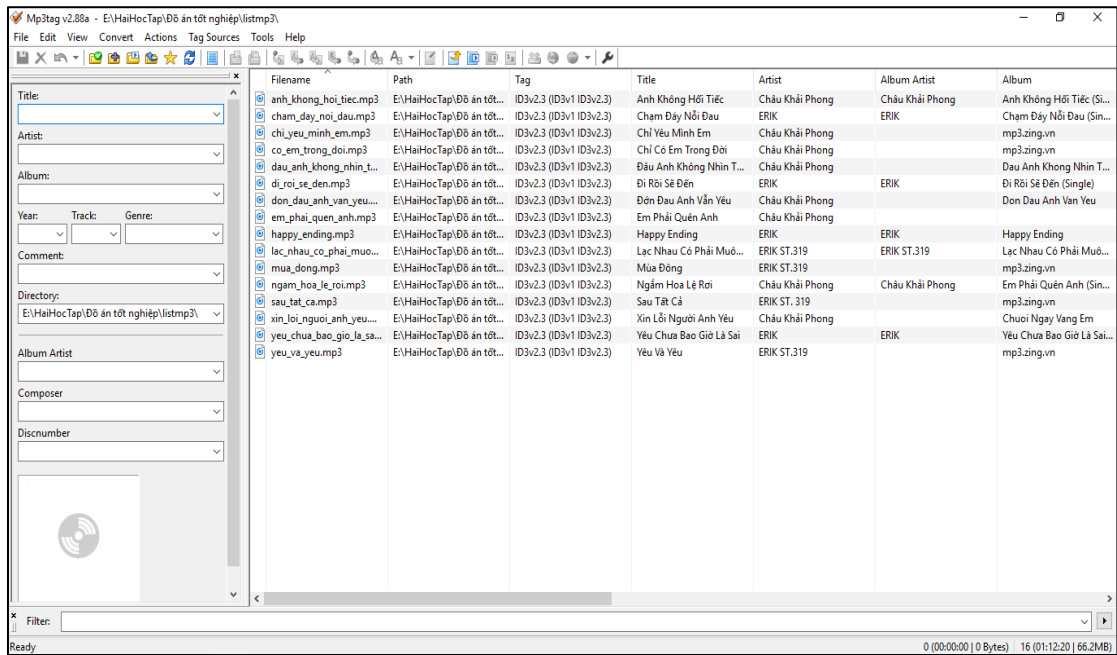
Hình 3.3.1.2. Giao diện chương trình

### 3.3.2. Một số thao tác cơ bản trong mp3tag

- Bước 1: Chọn **File -> Add directory** và chọn đến thư mục chứa nhạc.

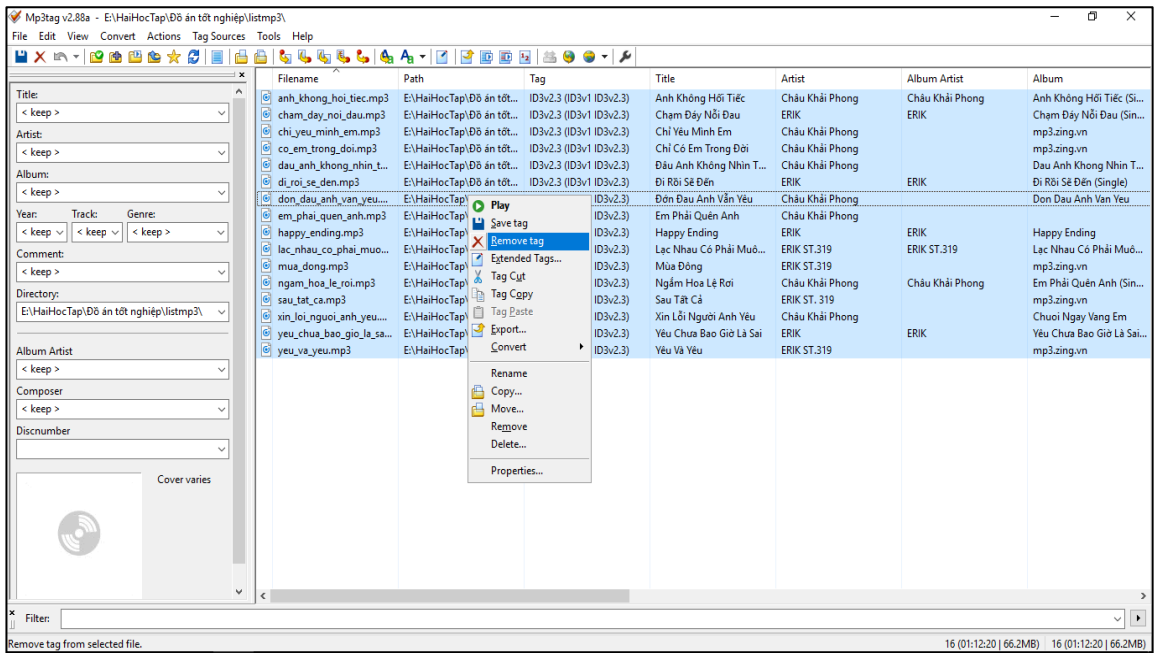


Hình 3.3.2.1. Thao tác chọn đến thư mục chứa nhạc

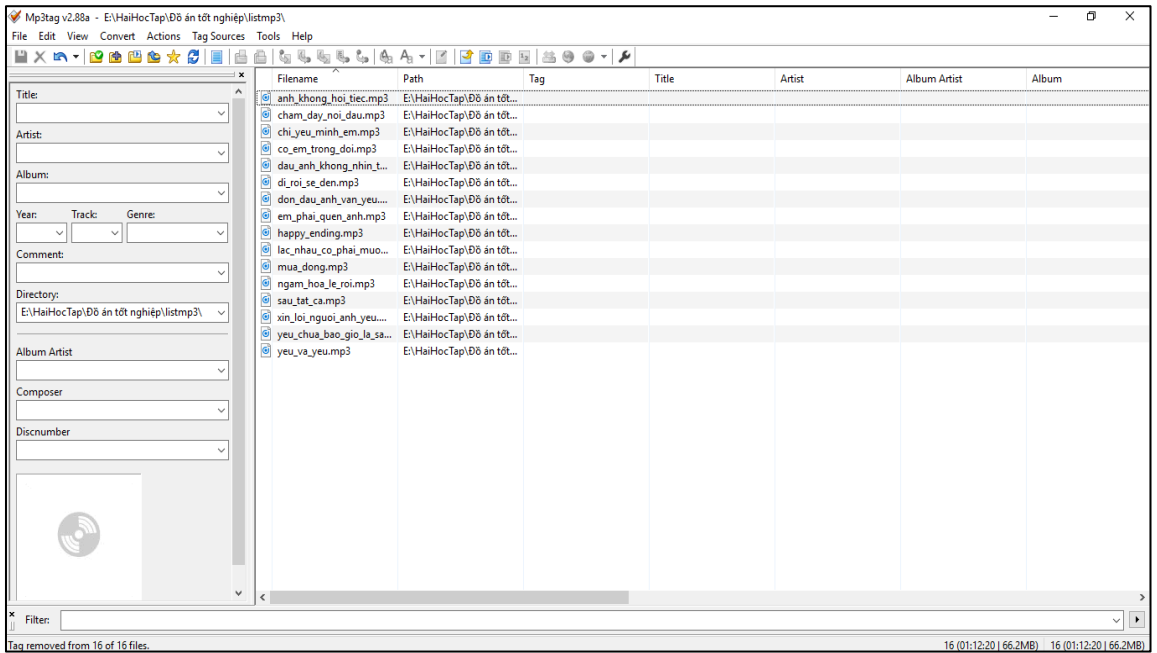


Hình 3.3.2.2. Kết quả sau khi chọn đến thư mục chứa nhạc

- Bước 2: Xóa các tag không cần thiết để làm sạch các file nhạc.  
Chọn (bôi đen) tất cả các file nhạc, chọn chuột phải và chọn **Remove tag**.



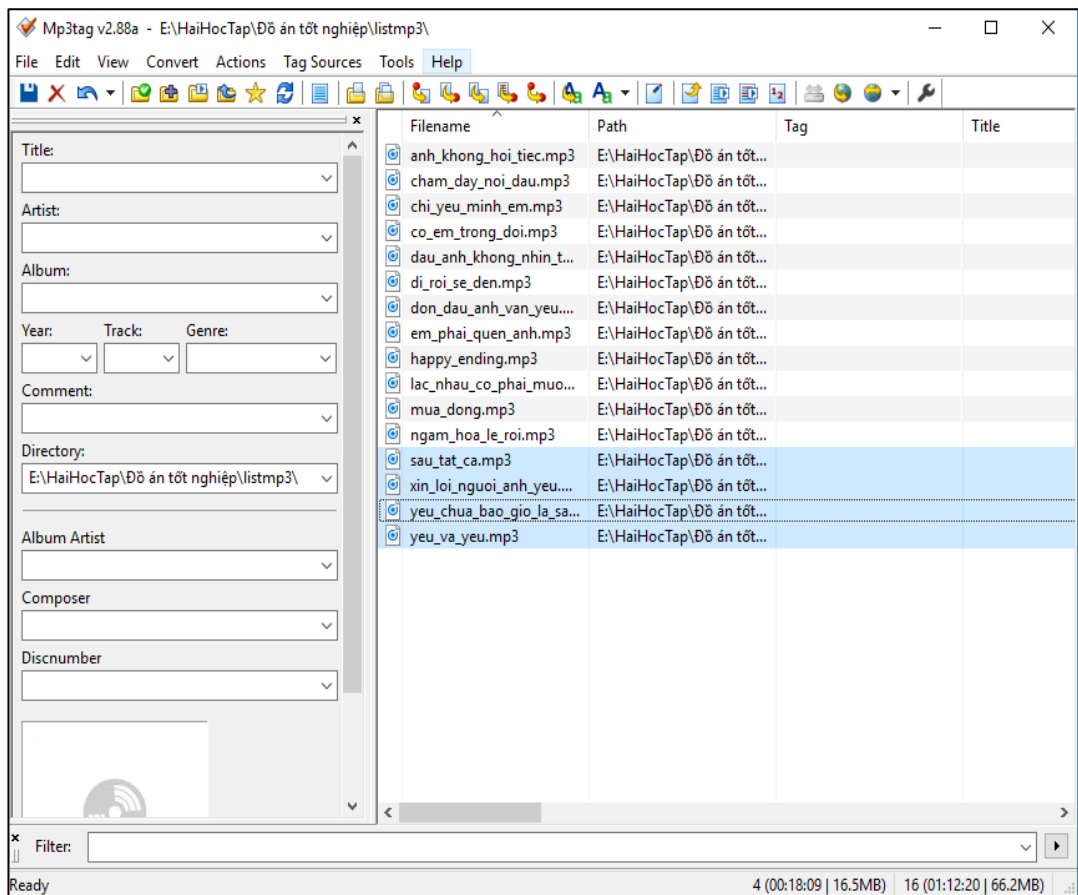
Hình 3.3.2.3. Thao tác chọn Remove tag



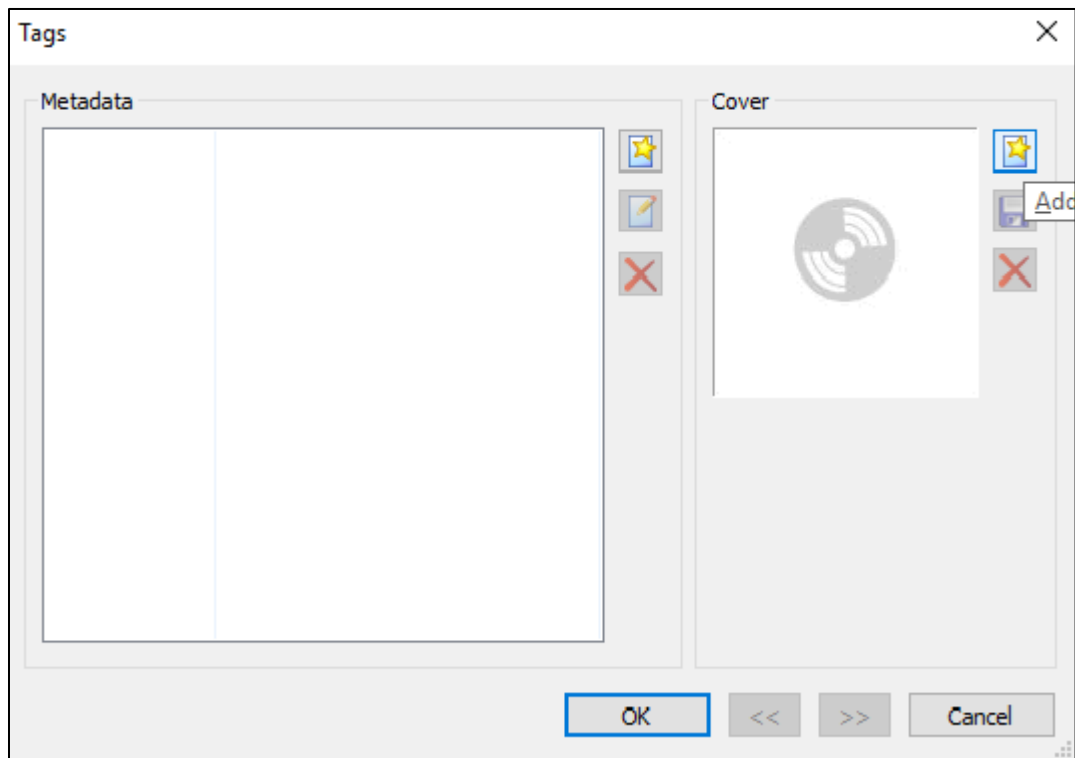
Hình 3.3.2.4. Kết quả sau khi đã xóa tag

– Bước 3: Thêm tag cho file nhạc.

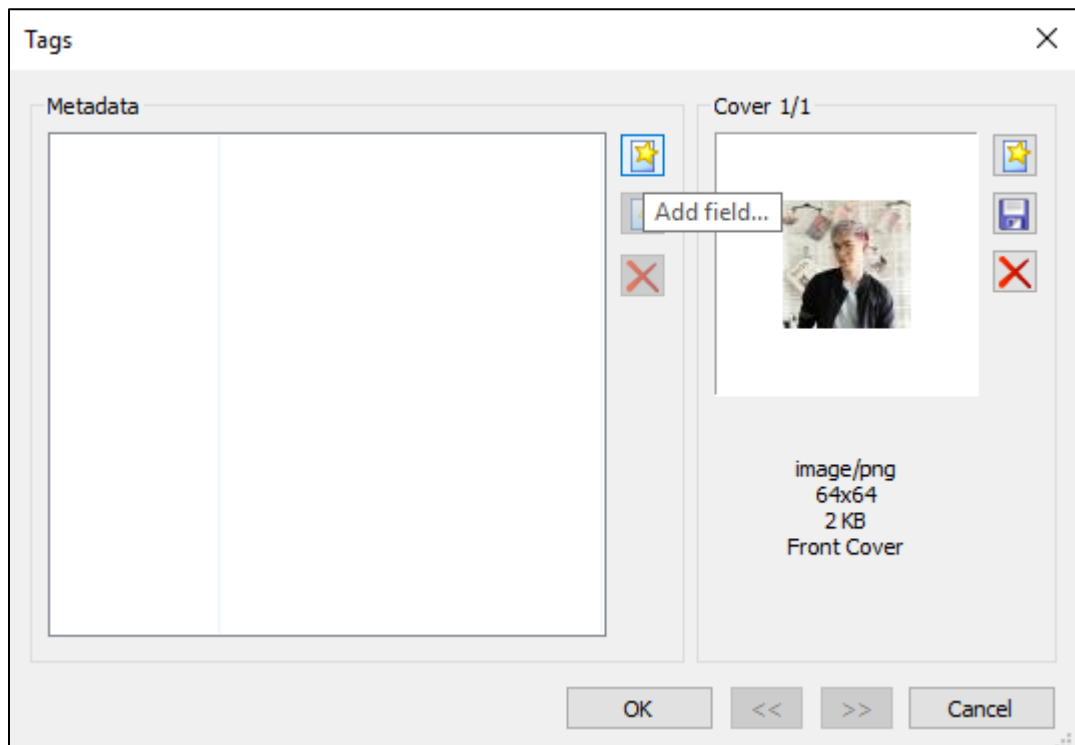
Chọn (bôi đen) tất cả các file nhạc muốn tạo thành một album, nhấn chuột phải và chọn **Extended tags**.



Hình 3.3.2.5. Kết quả khi bôi đen các file nhạc muốn tạo thành album  
 Xuất hiện hộp thoại **Tags**, chọn biểu tượng hình ngôi sao bên **Add cover** để thêm ảnh bìa cho album

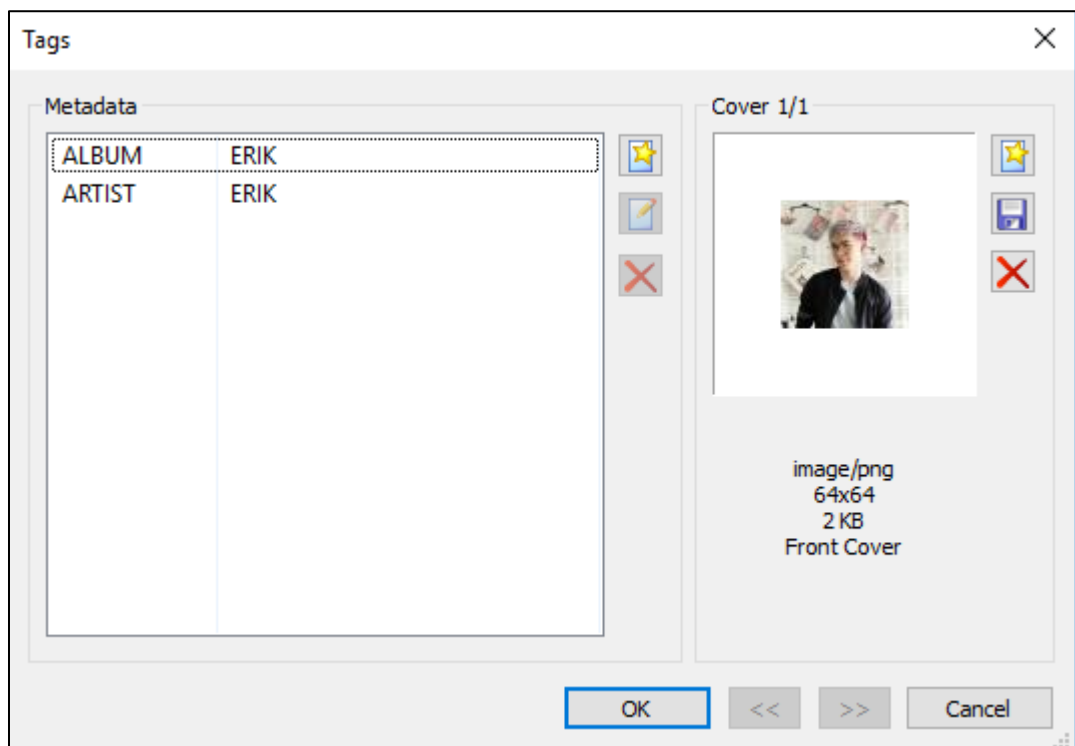


Hình 3.3.2.6. Biểu tượng của nút add cover để thêm ảnh  
 Nhấn chọn biểu tượng hình ngôi sao ở bên **Add field** để thêm tag cho file nhạc.



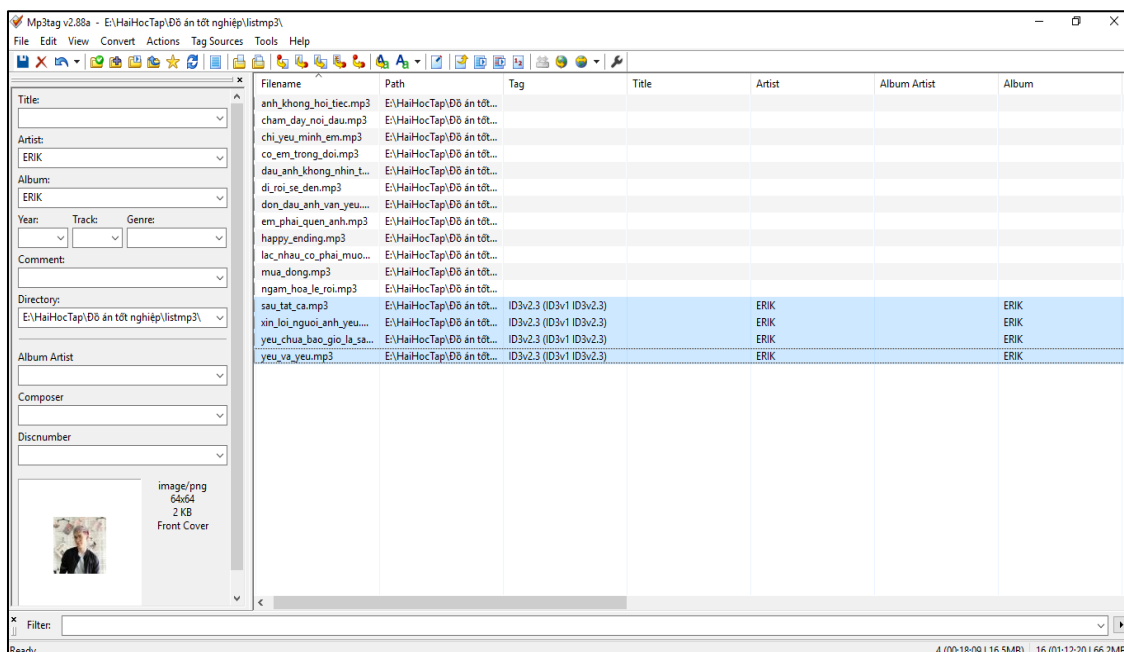
Hình 3.3.2.7. Biểu tượng add field để thêm tag

Thêm hai tag là tên album và tên ca sĩ.



Hình 3.3.2.8. Kết quả sau khi thêm hai tag ALBUM và ARTIST





Hình 3.3.2.9. Kết quả cuối cùng sau khi đã thêm tag tên album và tên ca sỹ

### 3.3. Phát nhạc trong android

#### 3.3.1. Xin quyền truy cập bộ nhớ ngoài

Dù ghi dữ liệu hay đọc dữ liệu thì cũng phải xin cấp quyền đọc hay ghi trên bộ nhớ trong. Chính vì vậy mà đầu tiên chúng ta sẽ đi xin cung cấp quyền trước bằng cách khai báo vào trong file AndroidManifest.xml như sau:

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

#### 3.3.2. Thư viện phát nhạc MediaPlayer

Android cung cấp nhiều cách để điều khiển tập tin audio hoặc video[3][4]. Một trong những cách này là sử dụng lớp **MediaPlayer**

##### a. Các phương thức

- **create():** Phương thức này giúp chúng ta tạo một thể hiện của lớp **MediaPlayer**.
- **start():** Nhạc sẽ được chơi từ khi bắt đầu.
- **pause():** Nhạc sẽ được phát từ chỗ nó bị dừng chứ không phải được phát từ đầu.
- **reset():** Để bắt đầu âm nhạc ngay từ đầu.
- **isPlaying():** Phương thức này chỉ trả về true/false cho biết bài hát được chơi hay không.
- **seekTo (position):** Phương thức này có một số nguyên, và di chuyển bài hát đến giây phút đặc biệt.

- **getCurrentDuration():** Phương thức này trả về vị trí hiện tại của bài hát trong mili giây.
- **getDuration():** Phương thức này trả về tổng thời gian của bài hát trong mili giây.
- **release():** Phương thức này phát hành bất kỳ tài nguyên gắn với đối tượng MediaPlayer.
- **setVolume(float leftVolume, float rightVolume):** Phương thức này đặt âm lượng lên xuống.
- **setDataSource(FileDescriptor fd):** Phương thức này tập hợp các nguồn dữ liệu của tập tin âm thanh / video.
- **selectTrack(int index):** Phương thức này có một số nguyên, và chọn bài hát từ danh sách trên chỉ số đặc biệt.
- **getTrackInfo():** Phương thức này trả về một mảng các thông tin theo dõi.

#### b. Các hàm callback

- **setOnPreparedListener(OnPreparedListener):** Lắng nghe sự kiện Prepared xong của media để chuẩn bị cho việc play.
- **setOnVideoSizeChangedListener(OnVideoSizeChangedListener):** Lắng nghe sự kiện thay đổi size của media trong lúc play video.
- **setOnSeekCompleteListener(OnSeekCompleteListener):** Lắng nghe sự kiện khi di chuyển seekbar hỗ trợ bởi MediaPlayer
- **setOnCompletionListener(OnCompletionListener):** Lắng nghe sự kiện khi play xong một media, sử dụng để có thể quy định việc tiếp theo nó sẽ làm là play lại hay play bài hát khác.
- **setOnBufferingUpdateListener(OnBufferingUpdateListener):** Lắng nghe sự kiện thay đổi của bộ đệm khi play media online
- **setOnInfoListener(OnInfoListener):** Lắng nghe sự kiện khi có thông tin hoặc cảnh báo
- **setOnErrorListener(OnErrorListener):** Lắng nghe sự kiện nếu xảy ra lỗi khi play media

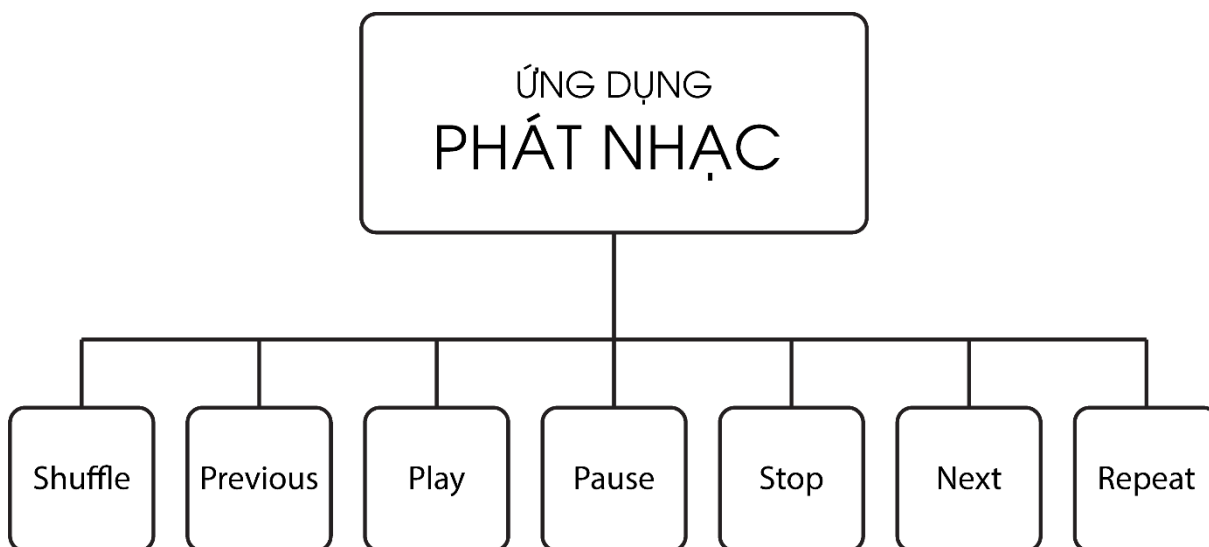
## CHƯƠNG 4: CHƯƠNG TRÌNH THỰC NGHIỆM

### 4.1. Phát biểu bài toán

Nghe nhạc trên điện thoại là một trong các nhu cầu giải trí phổ biến của người dùng, nó giúp thư giãn và làm việc hiệu quả hơn. Chương trình nghe nhạc thường được tích hợp sẵn trên điện thoại. Tuy nhiên việc tự xây dựng ứng dụng phát nhạc cá nhân giúp cho người dùng có thể tùy biến giao diện máy phát nhạc, hiển thị thêm trên giao diện các thông tin về bài hát họ yêu thích, chọn lựa bài hát cần nghe một cách dễ dàng, nhanh chóng... Xuất phát từ nhu cầu trên, em sẽ

### 4.2. Thiết kế chức năng hệ thống

#### 4.2.1. Sơ đồ chức năng



Hình 4.2.1.1. Sơ đồ chức năng

- 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.

#### 4.2.2. Xây dựng chương trình

##### a. Xây dựng giao diện

Giao diện chương trình bao 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.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/list_item"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
    <ImageView
        android:id="@+id/ivAnhcasi"
        android:layout_width="64dp"
        android:layout_height="64dp"
        android:src="@mipmap/ic_launcher"/>
    <TextView
        android:textStyle="bold"
        android:layout_marginTop="10dp"
        android:layout_toRightOf="@id/ivAnhcasi"
        android:id="@+id/tv_title"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Tên bài hát:"
        android:textColor="#000"
        android:textSize="20sp" />
    <TextView
        android:layout_toRightOf="@id/ivAnhcasi"
        android:layout_below="@id/tv_title"
        android:id="@+id/tv_artist"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Tên ca sỹ:"
        android:textColor="#424242"
        android:textSize="16sp" />
</RelativeLayout>
```

## b. Tạo lớp ListViewAdapter

- Mô tả

Lấy dữ liệu của từng đối tượng trong danh sách bài hát rồi gán từng thuộc tính của đối tượng đó vào trong layout item\_list.xml.

- Mã nguồn

```
public class ListViewAdapter extends BaseAdapter {
    private ArrayList<Song> danhSachbaihat;
    private Context context;
```

```

    public ListViewAdapter(Context context, ArrayList<Song>
danhsachbaihat) {
        this.danhsachbaihat = danhsachbaihat;
        this.context = context;
    }
    public View getView(int position, View convertView, ViewGroup
parent) {
        Holder holder;
        if (convertView == null) {
            holder = new Holder();
            LayoutInflater inflater = (LayoutInflater)
context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
            convertView =
inflater.inflate(R.layout.item_list, null);
            holder.tvTitle = (TextView)
convertView.findViewById(R.id.tv_title);
            holder.tvArtist = (TextView)
convertView.findViewById(R.id.tv_artist);
            holder.ivAnhCaSi = (ImageView)
convertView.findViewById(R.id.ivAnhcasi);
            convertView.setTag(holder);
        }else{
            holder = (Holder) convertView.getTag();
        }
        Song song = getItem(position);
        holder.tvTitle.setText(song.title);
        holder.tvArtist.setText(song.artist);
        Bitmap bm = BitmapFactory.decodeByteArray(song.anhcasy, 0,
song.anhcasy.length);
        holder.ivAnhCaSi.setImageBitmap(bm);
        return convertView;
    }
    private class Holder {
        TextView tvTitle;
        TextView tvArtist;
        ImageView ivAnhCaSi;
    }
}

```

### c. Lấy danh sách bài hát gán 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();
}

```

#### d. 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 sỹ");
        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((danhsachbaihat.size() - 1) - 0 + 1) + 0;
                runSong();
                mediaPlayer.start();
                status = PLAY;
            }
            ivPlay.setImageResource(R.drawable.ic_play_arrow_black_24dp);
        }
    });
    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 == danhsachbaihat.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(SeekBar seekBar, int i, boolean
b) {
    }
    @Override
    public void onStartTrackingTouch(SeekBar seekBar) {
    }
    @Override
    public void onStopTrackingTouch(SeekBar 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.

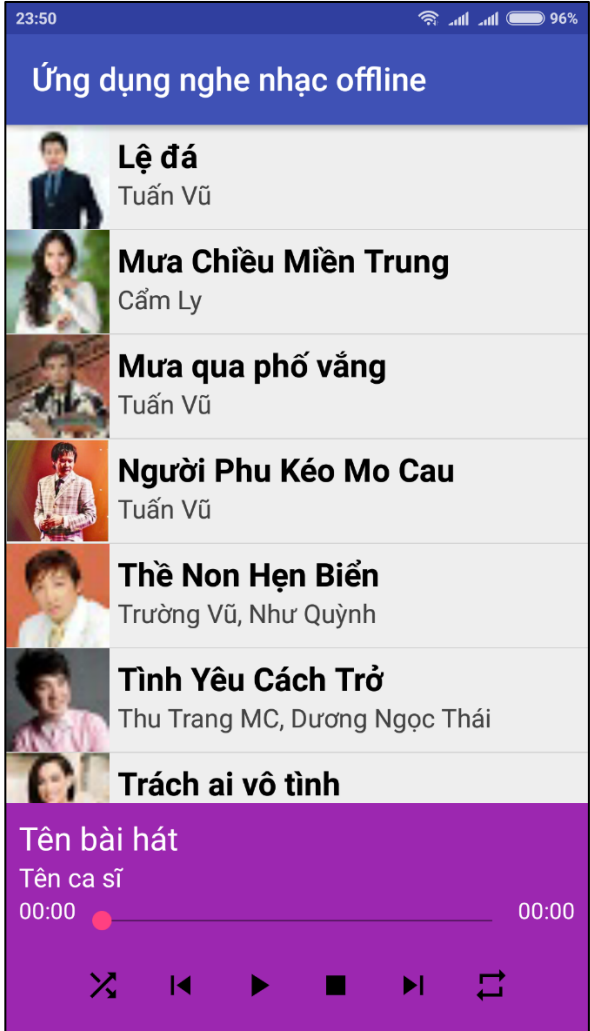
- Mã nguồn

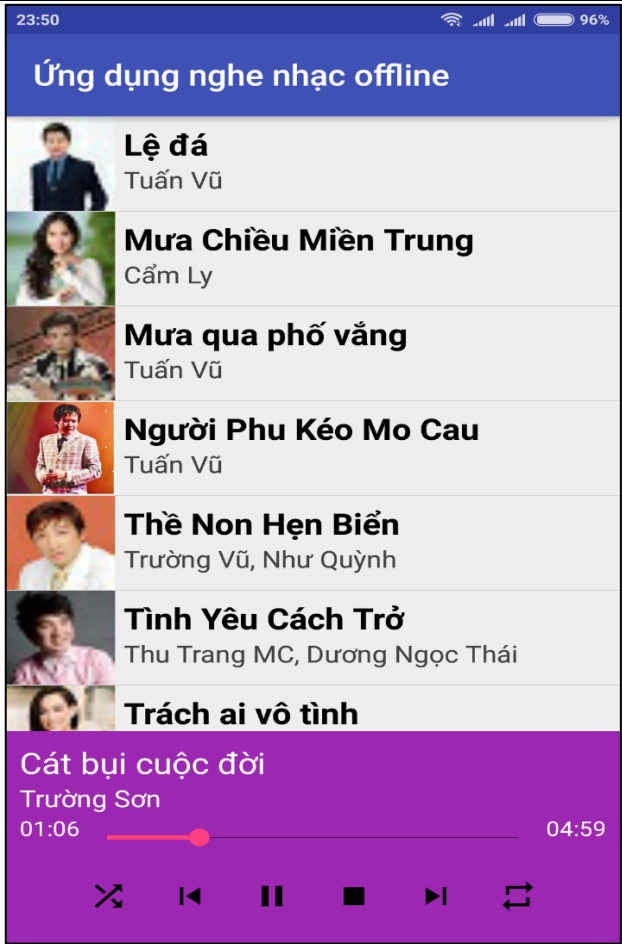
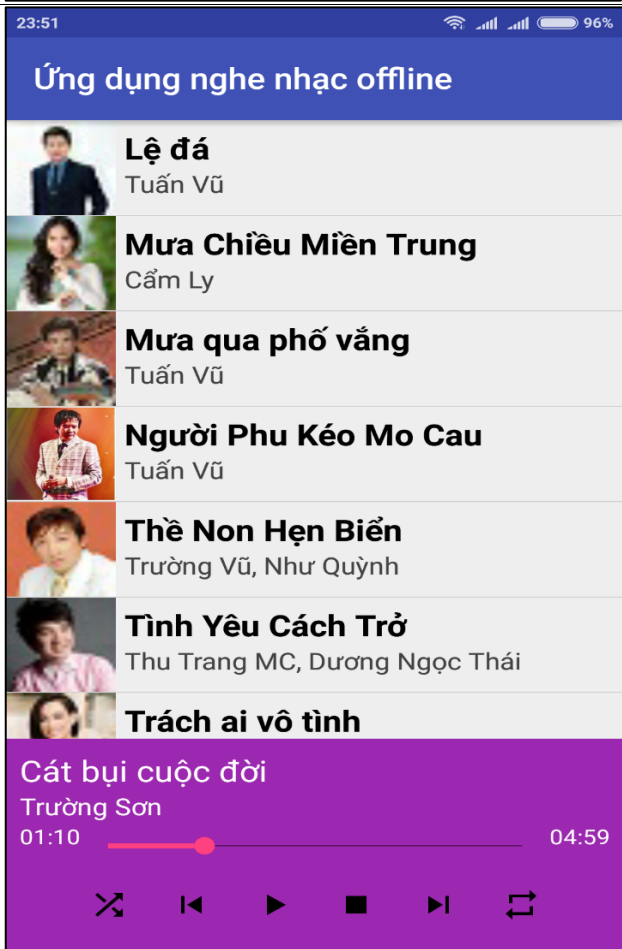
```

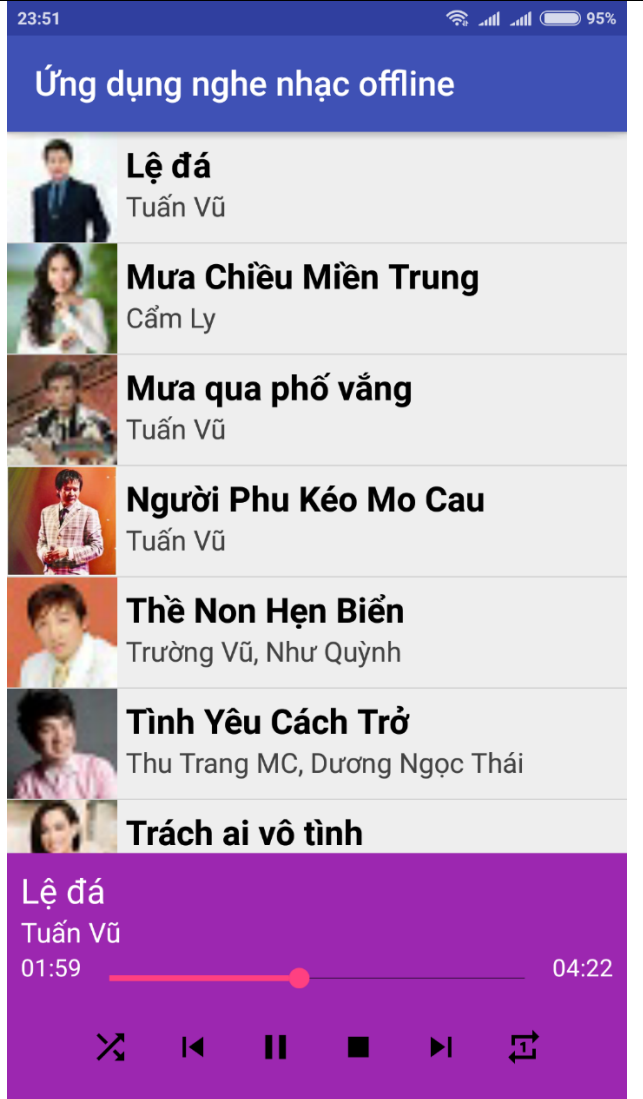
private void updateTimeSong(){
    final Handler handler = new Handler();
    handler.postDelayed(new Runnable() {
        @Override
        public void run() {
            SimpleDateFormat dinhDang = new
SimpleDateFormat("mm:ss");
txtTimeCurrent.setText(dinhDang.format(mediaPlayer.getTimeCurrent(
)));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

```

### 4.3. Giao diện chương trình

Mô tả	Hình ảnh
<p>Màn hình chương trình khi khởi động:</p> <ul style="list-style-type: none"> <li>- Hiển thị list danh sách bài hát.</li> <li>- Mỗi một bài hát sẽ hiển thị: <ul style="list-style-type: none"> <li>• Ảnh ca sỹ</li> <li>• Tên bài hát</li> <li>• Tên ca sỹ</li> </ul> </li> <li>- Hiển thị các nút chức năng của trương trình.</li> </ul>	

Mô tả	Hình ảnh
<p>Giao diện chương trình khi phát một bài hát:</p> <ul style="list-style-type: none"> <li>- Hiển thị tên bài hát, tên ca sỹ.</li> <li>- Hiển thị tổng thời gian và thời gian thực của bài hát.</li> <li>- Thanh Seekbar thể hiện thời gian thực của bài hát</li> <li>- Nút Play sẽ hiển thị icon pause.</li> </ul>	
<p>Giao diện chương trình khi tạm dừng một bài hát:</p> <ul style="list-style-type: none"> <li>- Nút Play sẽ hiển thị icon play.</li> </ul>	

Mô tả	Hình ảnh
<p>Giao diện chương trình khi bật nút Repeat:</p> <ul style="list-style-type: none"> <li>Nút Repeat sẽ đổi icon khác</li> </ul>	

## TỔNG KẾT

Trong đồ án này 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 trúc 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 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 các thầy cô giáo 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

### ❖ Tham khảo trực tuyến

- [1] <https://www.thegioididong.com/tin-tuc/he-dieu-hanh-android-qua-cac-phien-ban-cap-nhat-592544>
- [2] <https://tinhte.vn/threads/tim-hieu-qua-trinh-phat-trien-cua-android-qua-cac-phien-ban.1367201/>
- [3] <https://viblo.asia/p/tim-hieu-mediaplayer-trong-android>
- [4] <http://giasutinhoc.vn/phat-trien-app-android/media-player-trong-android>
- [5] <https://www.thegioididong.com/hoi-dap/cac-dinh-dang-video-va-am-thanh-pho-bien-hien-nay>
- [6] <https://vi.wikipedia.org/wiki/MP3>
- [7] Kênh youtube của Khoa Phạm