TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN CUỐI KỲ MÔN LẬP TRÌNH ỨNG DỤNG DI ĐỘNG**

**PHÁT TRIỂN ỨNG DỤNG ÂM NHẠC**

*Người hướng dẫn*: **Thầy NGUYỄN THANH PHƯỚC**

*Người thực hiện*: **NGUYỄN THANH HÙNG – 517H0126**

**LÝ GIA BẢO – 517H0097**

**PHẠM XUÂN ĐỨC – 517H0107**

Lớp **: 17050310**

Khoá  **: 21**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2020**

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN CUỐI KỲ MÔN LẬP TRÌNH ỨNG DỤNG DI ĐỘNG**

**PHÁT TRIỂN ỨNG DỤNG ÂM NHẠC**

*Người hướng dẫn*: **Thầy NGUYỄN THANH PHƯỚC**

*Người thực hiện*: **NGUYỄN THANH HÙNG – 517H0126**

**LÝ GIA BẢO – 517H0097**

**PHẠM XUÂN ĐỨC – 517H0107**

Lớp **: 17050310**

Khoá  **: 21**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2020**

LỜI CẢM ƠN

Nhóm em xin gửi lời cảm ơn chân thành và sự tri ân sâu sắc đối với thầy NGUYỄN THANH PHƯỚC cùng với các thầy,cô trong khoa Công Nghệ Thông Tin đã tạo điều kiện cho em có đề tài về ỨNG DỤNG NGHE NHẠC . Và em cũng xin chân thành cảm ơn thầy môn đã nhiệt tình hướng dẫn để em hoàn thành tốt bài tiểu luận này.

Bài làm của nhóm em còn nhiều sai sót và một số hạn chế. Nhóm em rất mong quý Thầy Cô bỏ qua và nhận được ý kiến đóng góp quý báo Thầy Cô để nhóm em học thêm được nhiều kinh nghiệm và sẽ hoàn thành tốt hơn trong đề tài sắp tới. Nhóm em xin chân thành cảm ơn.

TÓM TẮT

Các phần mềm ứng dụng trên các thiết bị điện tử di động (hay còn gọi tắt là ứng dụng di động), là phần mềm được thiết kế để chạy trên điện thoại di động, máy tính bảng, và vô số các thiết bị điện tử di động khác.

Trong thời đại số hóa đang phát triển vô cùng mạnh mẽ, ứng dụng di động trở nên phổ biến hơn bao giờ hết, với vô số các tính năng bao phủ và hỗ trợ mạnh mẽ cho đời sống thường nhật của con người. Bên cạnh đó, nhu cầu về nghe nhạc cũng trở nên phổ biến,cần thiết, và rất quan trọng trong đời sống của con người về vấn đề thư giãn sau những ngày làm việc, học tập mệt mỏi.

**ĐỒ ÁN ĐƯỢC HOÀN THÀNH**

**TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Chúng tôi xin cam đoan đây là sản phẩm đồ án của riêng chúng tôi và được sự hướng dẫn của thầy Nguyễn Thanh Phước. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình.** Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày 25 tháng 3 năm 2020*

*Tác giả*

*(ký tên và ghi rõ họ tên)*

*Nguyễn Thanh Hùng*

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

**Phần xác nhận của GV hướng dẫn**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

**Phần đánh giá của GV chấm bài**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ

**DANH MỤC HÌNH**

Hình 1.1 Android………………………………………………………………..…9

Hình 1.2 Lịch sử về Android…………………………………………………..…11

Hình 1.3 Các phiên bảng android…………………………………………….….12

Hình 1.4 Kiến trúc Android………………………………………………………15

Hình 2.0 Giao diện trang chủ sản phẩm……………………………………….…18

Hình 3.1 Cấu hình retrofit …………………………………………………….….29

Hình 3.2 Các phương thức dùng để tương tác với server………………………...30

Hình 3.3 Cầu nối giữa retrofit và các phương thức………………………………31

Hình 3.4 Thiết kế class cho các đối tượng …………………………….……….32

Hình 3.5 Giao Diện Trang Chủ………………………………………….………..32

Hình 3.6 Thiết kế phần giao diện cho fragment playlist……………….…………33

Hình 3.7 Gắn layout vào fragment và gắn adapter vào fragment………….……..33

Hình 3.8 Thiết kế layout cho adapter playlist…………………………….………34

Hình 3.9 Cấu hình adapter cho playlist………………………………….……….34

Hình 3.10 Màn hình giao diện cho phần hiển thị các bài hát từ playlist hay album……………………………………………………………….…………………….35

Hình 3.11 Màn hình media play music …………………………………………36

Hình 3.12 Màn hình hiển thị tìm kiếm bài hát……………………………………36

Hình 3.13 Màn hình đăng nhập cho user…………………………………………37

Hình 3.14 Màn hình hiển thị tất cả các Playlist…………………………………..37

MỤC LỤC

Contents

[LỜI CẢM ƠN i](#_Toc39435251)

[TÓM TẮT ii](#_Toc39435252)

[PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN iv](#_Toc39435253)

[DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ 5](#_Toc39435254)

[MỤC LỤC 5](#_Toc39435255)

[CHƯƠNG 1: GIỚI THIỆU 7](#_Toc39435256)

[I. Tổng Quan Về Android 7](#_Toc39435257)

[1.1 Android là gì ? 7](#_Toc39435258)

[1.2 Lịch sử về android 8](#_Toc39435259)

[1.3 Các phiên bản chính của android. 11](#_Toc39435260)

[1.4 Kiến trúc Android 14](#_Toc39435261)

[1.4.1 Nhân Linux (Linux kernel) 15](#_Toc39435262)

[1.4.2. Thư viện Android (Android libraries) và Android Runtime 15](#_Toc39435263)

[1.4.3. Application Framework 16](#_Toc39435264)

[1.4.4 – Ứng dụng (applications) 16](#_Toc39435265)

[II.Giới Thiệu Của Nhóm 17](#_Toc39435266)

[CHƯƠNG 2: CÔNG NGHỆ VÀ THƯ VIỆN ĐƯỢC SỬ DỤNG 18](#_Toc39435267)

[1. Retrofit 18](#_Toc39435268)

[1.1 Retrofit là gì ? 18](#_Toc39435269)

[1.2 Sử dụng Retrofit 18](#_Toc39435270)

[2. JSON 19](#_Toc39435271)

[2.1 JSON là gì ? 19](#_Toc39435272)

[2.2 Cấu trúc của JSON. 19](#_Toc39435273)

[2.3 Thao tác với JSON trong Android 20](#_Toc39435274)

[2.3.2 Ghi JSON trong Android. 24](#_Toc39435275)

[CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ ỨNG DỤNG 28](#_Toc39435276)

[3.1 Cấu hình Service để tương tác với server 28](#_Toc39435277)

[3.2 Thiết kế trang chủ 31](#_Toc39435278)

[3.3 Thiết kế cho các activity khác 34](#_Toc39435279)

[CHƯƠNG 4 – TỔNG KẾT 37](#_Toc39435280)

[TÀI LIỆU THAM KHẢO 38](#_Toc39435281)

CHƯƠNG 1: GIỚI THIỆU

1. Tổng Quan Về Android
   1. Android là gì ?

Android là một hệ điều hành dựa trên nền tảng Linux, 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

[](https://genk.mediacdn.vn/k:2015/android-1-0-1433777469803/lich-su-va-chang-duong-phat-trien-cua-android-qua-cac-phien-ban-phan-1-thoi-ky-dau-cua-android.png)

***Hình 1.1 Android***

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.

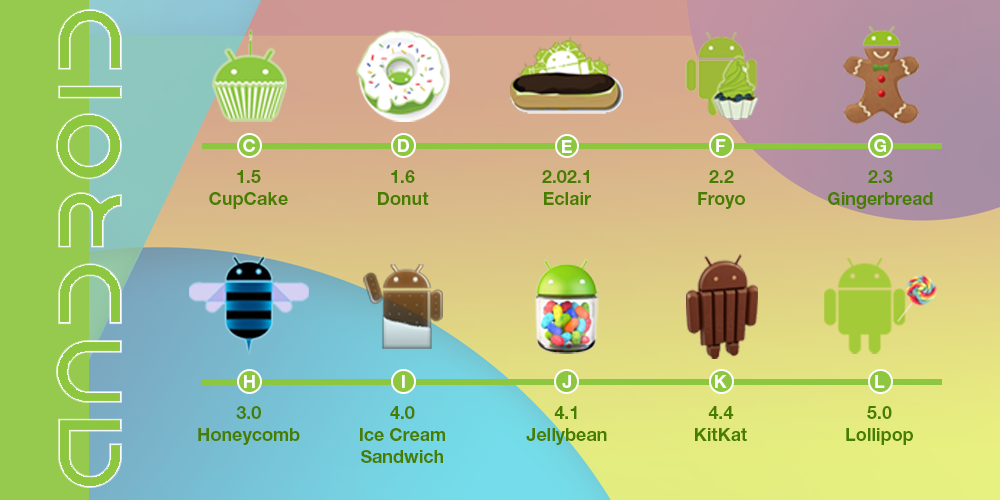
Cho đến hiện tại, Google đã bán ra 200.000 thiết bị Android mỗi ngày và Play Store của họ vượt mức 90.000 ứng dụng. Nhờ yếu tố mở, dễ dàng tinh chỉnh cùng sự phát triển nhanh chóng đã khiến hệ điều hành này dần trở nên phổ biến, kết quả là mặc dù được thiết kế để chạy trên điện thoại và máy tính bảng nhưng giờ đây Android đã xuất hiện trên các smart TV, máy chơi game và một số thiết bị điện tử khác.

* 1. Lịch sử về android

Tổng công ty Android (Android, Inc.) đƣợc thành lập tại Palo Alto, California vào tháng 10 năm 2003 bởi Andy Rubin (đồng sáng lập công ty Danger),[20] Rich Miner (đồng sáng lập Tổng công ty Viễn thông Wildfire), Nick Sears (từng là Phó giám đốc T-Mobile), và Chris White (trƣởng thiết kế và giao diện tại WebTV) để phát triển, theo lời của Rubin, "các thiết bị di động thông minh hơn có thể biết đƣợc vị trí và sở thích của ngƣời dùng". Dù những ngƣời thành lập và nhân viên đều là những ngƣời có tiếng tăm, Tổng công ty Android hoạt động một cách âm thầm, chỉ tiết lộ rằng họ đang làm phần mềm dành cho điện thoại di động. Trong năm đó, Rubin hết kinh phí. Steve Perlman, một ngƣời bạn thân của Rubin, mang cho ông 10.000 USD tiền mặt nhƣng từ chối tham gia vào công ty. Google mua lại Tổng công ty Android vào ngày 17 tháng 8 năm 2005, biến nó thành một bộ phận trực thuộc Google. Những nhân viên của chủ chốt của Tổng công ty Android, gồm Rubin, Miner và White, vẫn tiếp tục ở lại công ty làm việc sau thƣơng vụ này. Vào thời điểm đó không có nhiều thông tin về Tổng công ty, nhƣng nhiều ngƣời đồn đoán rằng Google dự tính tham gia thị trƣờng điện thoại di động sau bƣớc đi này. Tại Google, nhóm do Rubin đứng đầu đã phát triển một nền tảng thiết bị di động phát triển trên nền nhân Linux. Google quảng bá nền tảng này cho các nhà sản xuất điện thoại và các nhà mạng với lời hứa sẽ cung cấp một hệ thống uyển chuyển và có khả năng nâng cấp. Google đã liên hệ với hàng loạt hãng phần cứng cũng nhƣ đối tác phần mềm, bắn tin cho các nhà mạng rằng họ sẵn sàng hợp tác với các cấp độ khác nhau. Ngày càng nhiều suy đoán rằng Google sẽ tham gia thị trường. 

***Hình 1.2 Lịch sử về Android***

* 1. Các phiên bản chính của android.



*Hình 1.3 Các phiên bảng android*

**Phiên bản 1.0:** Hệ điều hành Android thời gian đầu ra mắt mang những đặc điểm, tính năng thú vị phải kể đến như:

Thanh thông báo kéo từ trên xuống cho phép người dùng xem nhanh các thông tin ngày giờ, tin nhắn, cuộc gọi…,

Màn hình chính và Widget: màn hình chính gồm các biểu tượng chương trình người dùng hay truy cập và các Widget là các ứng dụng nhỏ trên màn hình chính, hoạt động và cung cấp thông tin liên tục.

Tích hợp chặt chẽ với Gmail

**Phiên bản 1.1:** Phiên bản cập nhật đầu tiên của hệ điều hành Android giúp tinh chỉnh và sửa các lỗi gặp phải trên phiên bản 1.0. Tính năng cập nhật phần mềm từ động qua OTA (Over The Air) được bổ sung và được xem như là một cải tiến đáng giá vì các hệ điều hành di động trước đó đều phải nhờ tới một chiếc máy tính để thực hiện việc này.

**Phiên bản 1.5 Cupcake :** Android 1.5 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. 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.

**Phiên bản 1.6 Donut:** Phiên bản bánh 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.

**Phiên bản 2.0 và 2.1 Éclair:** 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.

**Phiên bản 2.2 Froyo:** Android 2.2 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.

**Phiên bản 2.3 Gingerbread:** Khoảng nửa năm sau khi FroYo xuất hiện, Google đã trở lại với bản Android 2.3. 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.

**Phiên bản 3.0 Honeycumb:** 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.

**Phiên bản 4.0 Ice Cream Sandwich:** 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. Đây cũng là lần đầu tiên Google hợp nhất hệ điều hành dành cho smartphone và cho máy tính bảng vào làm một.

**Phiên bản 4.1 Jelly Bean:** 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.

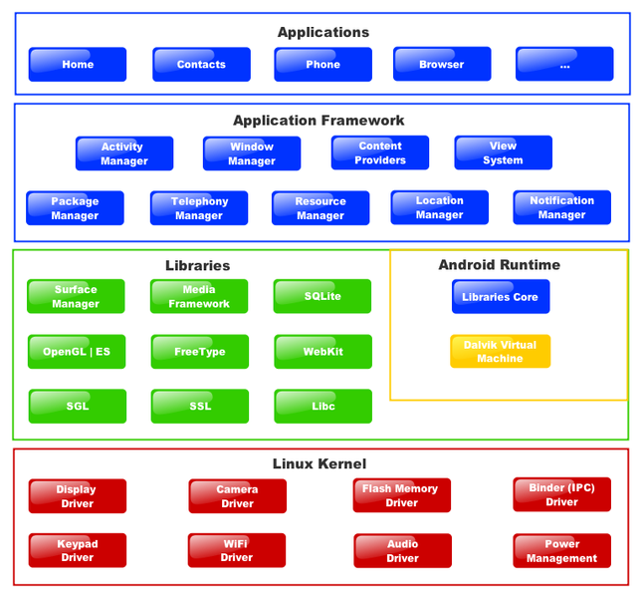
**Phiên bản 4.4 Kikat:** Dù là phiên bản mới, Android 4.4 Kitkat lại không đòi hỏi cấu hình phần cứng mạnh hơn, thậm chí hỗ trợ tốt cả những thiết bị cũ với phần cứng không cao, như có RAM chỉ đạt dung lượng 512 MB. Hệ điều hành mới được Google tối ưu khả năng hoạt động, cho hiệu năng cao hơn tới 1,6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

**Phiên bản 5.0 Lollipop:** 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 Thiết kế "Material" mới: Ngôn ngữ đồ họa mới của Android được lấy cảm hứng từ thiên nhiên, vật lý và phong cách đậm, đổ bóng giống như in ấn. Nói cách khác, đó là một thiết kế dựa trên những đặc tính của giấy.

Các tính năng khác cũng được cải tiến như: thiết kế báo nhắc hoàn toàn mới, hỗ trợ chip 64 bit và cải thiện thời lượng pin.

* 1. Kiến trúc Android



*Hình 1.4 Kiến trúc Android*

****1.4.1 Nhân Linux (Linux kernel)****

Nằm ở tầng dưới cùng là lớp Linux kernal – Linux 3.6 với khoảng 115 patches. Lớp này liên hệ với phần cứng và nó chứa tất cả driver phần cứng cần thiết như camera, bàn phím, màn hình…

1.4.****2. Thư viện Android (Android libraries) và Android Runtime****

Phía trên tầng Linux kernel là tầng libraries, chứa những thư viện hỗ trợ. Một số có thể kể đến như là bộ máy trình duyệt web mã nguồn mở WebKit, thư viện libc, cơ sở dữ liệu SQLite tiện lợi cho việc lưu trữ và chia sẻ dữ liệu, thư viện hỗ trợ thu phát âm thanh và video, thư viện SSL giúp bảo mật mạng…

Tầng này chứa tất cả các thư viện Java, được viết đặc biệt cho Android như các thư viện framework, các thư viện xây dựng giao diện, đồ họa và cơ sở dữ liệu. Dưới đây là một số thư viện quan trọng mà các lập trình viên nên biết:

android.app − Cung cấp quyền truy cập tới mô hình ứng dụng và là nền tảng của mọi ứng dụng Android applications.

android.content − Cho phép truy cập nội dung, phát hành và tin nhắn giữa các ứng dụng và các thành phần trong một ứng dung.

android.database −  Được dùng để truy cập vào dữ liệu được đưa ra bởi bộ phận cung cấp nội dung, bao gồm các lớp quản lý cơ sở dữ liệu SQLite.

android.opengl − Cho phép tương tác với thư viện đồ họa OpenGL ES 3D.

android.os − Giúp ứng dụng truy cập những dịch vụ cơ bản của hệ điều hành bao gồm tin nhắn, dịch vụ hệ thống và liên lạc nội bộ (inter-process communication).

android.text − Được dùng để vẽ và thao tác văn bản trên màn hình.

android.view − Các khối xây dựng cơ bản của giao diện người dùng.

android.widget − Một tập hợp rất nhiều thành phần giao diện được xây dựng sẵn như nút bấm (button), nhãn (label), danh sách hiển thị (list views), quản lý bố cục (layout managers)…

android.webkit − Tập hợp các lớp (classes) cho phép trình duyệt web được nhúng vào ứng dụng.

Trong tầng này còn có một phần không kém phần quan trọng là Android runtime, bây giờ chúng ta sẽ tìm hiểu các thư viện C/C++ trong tầng này.

Android Runtime chứa Dalvik Virtual Machine (DVM) – một biến thể của Java Virtual Machine, đặc biêt thiết kế và tối ưu hóa cho Android. DVM giúp mỗi ứng dụng Android chạy trong chính tiến trình (process) của nó với một đại diện (instance) của DVM.

Ngoài ra, Android Runtime cũng chứa tập hợp các thư viện quan trong cho phép người lập trình viết ứng dụng sử dụng ngôn ngữ lập trình Java.

****1.4.3. Application Framework****

Application Framework cung cấp nhiều dịch vụ cấp cao dưới dạng các lớp viết bằng Java (Java classes) Lập trình viên được phép sử dụng các lớp này để tạo ra các ứng dụng.

Android framework chứa các dịch vụ quan trọng như:

Activity Manager − Quản lý tất cả các phần của vòng đời (lifecycle) ứng dụng và các hoạt động ngăn xếp (activity stack).

Content Providers − Cho phép ứng dụng phát hành và chia sẻ dữ liệu với ứng dụng khác.

Resource Manager − Cho phép truy cập tới những tài nguyên không phải là mã nguồn như chuỗi, cài đặt màu, bố cục giao diện.

Notifications Manager − Giúp ứng dụng hiển thị thông báo và nhắc nhở người dùng.

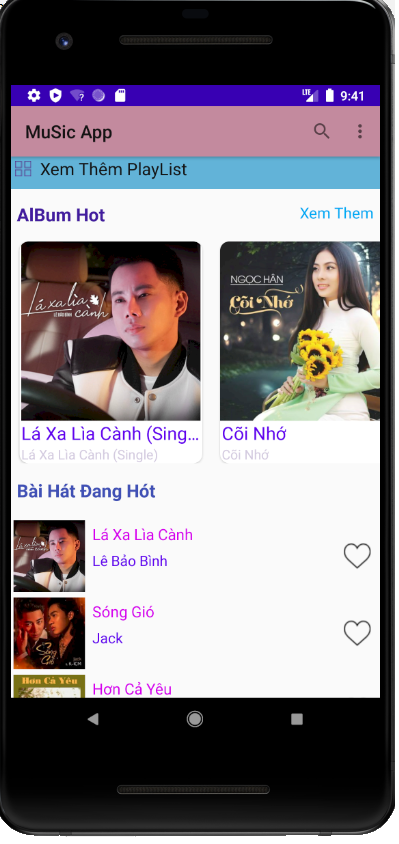
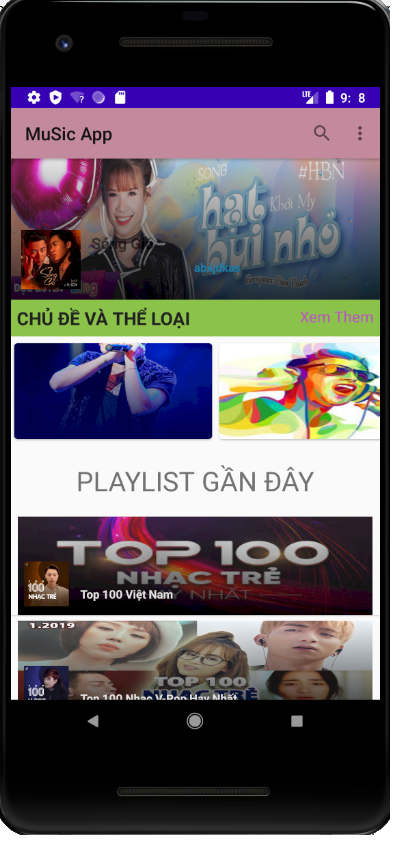
View System − Một tập hợp mở rộng giúp tạo giao diện người dùng.

****1.4.4 – Ứng dụng (applications)****

Ở tầng trên cùng là các ứng dụng Android đi kèm với hệ điều hành như Contacts Books, Browser, Games…

II.Giới Thiệu Của Nhóm

Project của chúng em là về việc tạo một app để nghe nhạc, app này được thiết kế dựa trên app nghe nhạc chuyên nghiệp Zing Mp3. App nhạc của chúng em còn rất nhiều thiếu xót,nhưng cũng đã hoàn thành cơ bản các chức năng cơ bản của một app nhac thông thường như nghe nhạc,đăng nhập để tương tác với bài hát…

** **

***Hình 2.0 Giao diện trang chủ sản phẩm.***

CHƯƠNG 2: CÔNG NGHỆ VÀ THƯ VIỆN ĐƯỢC SỬ DỤNG

1. Retrofit
   1. Retrofit là gì ?

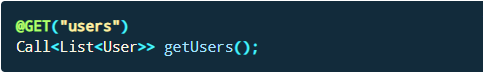
Retrofit là REST Client cho Java và Android. Nó giúp việc lấy và tải JSON ( hoặc dữ liệu có cấu trúc khác ) tương dối dễ dàng thông qua dịch vụ web dựa trên REST. Trong Retrofit, cấu hình trình chuyển đổi nào được sử dụng để tuaàn tự hóa dữ liệu. Thông thường đối với JSON thì sử dụng GSON, nhưng có thể thêm các trình chuyển đổi tùy chỉnh để xử lý các XML hoặc các giao thức khác. Retrofit sử dụng thư viện OkHttp cho các yêu cầu của HTTP.

* 1. Sử dụng Retrofit

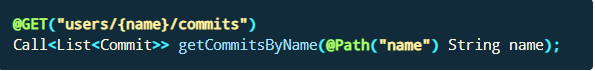
Để làm việc với Retrofit, về cơ bản cần ba lớp sau:

* Lớp mô hình được sử dụng làm mô hình JSON.
* Các giao diện xác định các hoạt động HTTP
* Lớp Retrofit.Builder. Trường hợp sử dụng giao diện và API Builder để cho phép xác định điểm cuối URL cho các hoạt động HTTP.

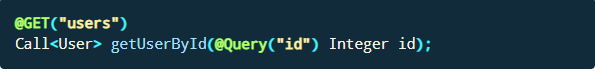
Mỗi phương thức của một giao diện đại diện cho một lệnh gọi API. Nó phải có một chú thích HTTP (***GET***, ***POST***,v.v) để xác định các loại yêu cầu và các URL có liên quan. Giá trị trả về kết thúc phản hồi trong ***Call***  đối tượng và loại kết quả mong đợi.



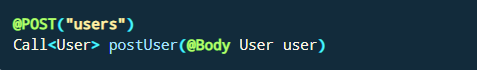
Sử dụng các khối thay thế và tham số truy vấn để điều chỉnh URL. Một khối thay thế được thêm vào URL tương đối với {}. Với sự trợ giúp của @Path chú thích trên tham số phương thức, giá trị của tham số đố được liên kết với khối thay thế cụ thể.



Các tham số truy vấn được thêm vào với @Query chú thích treen một tham số phương thức. Chúng được tự động thêm vào cuối URL.



Các @Body chú thích trên một số phương pháp bảo Retrofit để sử dụng đối tượng là cơ quan yêu cầu cho cuộc gọi.



1. **JSON**

### **2.1 JSON là gì ?**

JSON là tên viết tắt của Javascript Oject Notation. Là một cú pháp để lưu trữ và chuyển đổi dữ liệu và thuận tiện hơn XML.

**2.2 Cấu trúc của JSON.**

Cấu trúc của JSON được bắt đầu bằng kí tự ‘{’ và kết thúc bằng kí tự ‘}’.

Trong {} sẽ định nghĩa những object.

Ví dụ muốn định nghĩa một object Person có name là “ Nguyen Thanh Hung”, age là 22, address là ĐắkLắk thì chuỗi JSON sẽ được biể như sau:

**{name:"Nguyen Thanh Hung", age:22, address:" ĐắkLắk "}**

Một ví dụ nữa biểu diễn các danh sách employess:

|  |  |
| --- | --- |
| 3  4  5 | **{"employees":[**  **{"firstName":"John", "lastName":"Doe"},**  **{"firstName":"Anna", "lastName":"Smith"},**  **{"firstName":"Peter", "lastName":"Jones"}**  **]}** |

So sánh với cấu trúc XML:

**<employees>**

**<employee>**

**<firstName>John</firstName> <lastName>Doe</lastName>**

**</employee>**

**<employee>**

**<firstName>Anna</firstName> <lastName>Smith</lastName>**

**</employee>**

**<employee>**

**<firstName>Peter</firstName> <lastName>Jones</lastName>**

**</employee>**

**</employees>**

Chúng ta thấy rằng cùng biểu diễn một nội dung như nhau nhưng JSON ngắn gọn và xúc tích hơn XML. Và so sánh về tốc độ Parse thì JSON nhanh hơn rất nhiều so với XML.

**2.3 Thao tác với JSON trong Android**

Trong Android các thư viện dùng để thao tác với JSON nằm trong package org.json.

**2.3.1 Đọc Json trong Android**

**{"widget": {**

**"debug": "on",**

**"window": {**

**"title": "Sample Konfabulator Widget",**

**"name": "main\_window",**

**"width": 500,**

**"height": 500**

**},**

**"image": {**

**"src": "Images/Sun.png",**

**"name": "sun1",**

**"hOffset": 250,**

**"vOffset": 250,**

**"alignment": "center"**

**},**

**"text": {**

**"data": "Click Here",**

**"size": 36,**

**"style": "bold",**

**"name": "text1",**

**"hOffset": 250,**

**"vOffset": 100,**

**"alignment": "center",**

**"onMouseUp": "sun1.opacity = (sun1.opacity / 100) \* 90;"**

**}**

**}}**

Có tất cả 4 Object đó là:

* widget: có thuộc tính là debug giá trị on và 3 thuộc tính khác là 3 object window, image, text
* window: có các thuộc tính là title, name, width, height
* image: có các thuộc tính là src, name, hOffset, vOffset, alignment.
* text: có các thuộc tính là data, size, style, name,  hOffset, vOffset, alignment, onMouseUp.

Các bước parse như sau :

**Bước 1**: Chúng ta tạo đối tượng JSONObject với đối số truyền vào là chuổi JSON trên.

**String json = "{\"widget\": {\n" +**

**"    \"debug\": \"on\",\n" +**

**"    \"window\": {\n" +**

**"        \"title\": \"Sample Konfabulator Widget\",\n" +**

**"        \"name\": \"main\_window\",\n" +**

**"        \"width\": 500,\n" +**

**"        \"height\": 500\n" +**

**"    },\n" +**

**"    \"image\": { \n" +**

**"        \"src\": \"Images/Sun.png\",\n" +**

**"        \"name\": \"sun1\",\n" +**

**"        \"hOffset\": 250,\n" +**

**"        \"vOffset\": 250,\n" +**

**"        \"alignment\": \"center\"\n" +**

**"    },\n" +**

**"    \"text\": {\n" +**

**"        \"data\": \"Click Here\",\n" +**

**"        \"size\": 36,\n" +**

**"        \"style\": \"bold\",\n" +**

**"        \"name\": \"text1\",\n" +**

**"        \"hOffset\": 250,\n" +**

**"        \"vOffset\": 100,\n" +**

**"        \"alignment\": \"center\",\n" +**

**"        \"onMouseUp\": \"sun1.opacity = (sun1.opacity / 100) \* 90;\"\n" +**

**"    }\n" +**

**"}}";**

**try {**

**JSONObject rootObject = new JSONObject(json);**

**} catch (JSONException e) {**

**e.printStackTrace();**

**}**

**Bước 2**: Lấy ra JSONObject widget.

**JSONObject widgetObject = rootObject.getJSONObject("widget");**

**Bước 3**: Lấy ra tất các thuộc tính. Nếu là 1 Object thì phải tạo JSONObject và lấy thuộc tính bởi các phương thức.

* int getInt(String name);
* double getDouble(String name);
* boolean getBoolean(String name);
* String getString(String name);
* long getLong(String name);
* Object get(String name);
* Lấy thuộc tính debug

**String debug = widgetObject.getString("debug");**

* Lấy thuộc tính window

**JSONObject windowObject = widgetObject.getJSONObject("window");**

**String window\_title = windowObject.getString("title");**

**String window\_name = windowObject.getString("name");**

**int window\_height = windowObject.getInt("height");**

**int window\_width = windowObject.getInt("width");**

* Lấy thuộc tính image

**JSONObject imageObject = widgetObject.getJSONObject("image");**

**String image\_src = imageObject.getString("src");**

**String image\_name = imageObject.getString("name");**

**int image\_hOffset = imageObject.getInt("hOffset");**

**int image\_vOffset = imageObject.getInt("vOffset");**

**String image\_alignment = imageObject.getString("alignment");**

* Lấy thuộc tính text

**JSONObject textObject = widgetObject.getJSONObject("text");**

**String text\_data = textObject.getString("data");**

**String text\_size = textObject.getString("size");**

**String text\_style = textObject.getString("style");**

**String text\_name = textObject.getString("name");**

**int text\_hOffset = textObject.getInt("hOffset");**

**int text\_vOffset = textObject.getInt("vOffset");**

**String text\_alignment = textObject.getString("alignment");**

**String text\_on\_mouseup = textObject.getString("onMouseUp");**

Lấy về danh sách các Object thông qua phương thức getJSONArray( String name). Phương thức này trả về một JSONArray. Chúng ta sẽ qua từng phần tử trong danh sách và lấy ra các giá trị của các thuộc tính

**2.3.2 Ghi JSON trong Android.**

Trước tiên tạo một class City như sau:

**public class City {**

**private String mName;**

**private LatLng mPos;**

**public City(String name, LatLng pos){**

**mName = name;**

**mPos = pos;**

**}**

**public String getName() {**

**return mName;**

**}**

**public LatLng getPos() {**

**return mPos;**

**}**

**public void setName(String name) {**

**this.mName = name;**

**}**

**public void setPos(LatLng pos) {**

**this.mPos = pos;**

**}**

**}**

Override lại phương thức toString của class City để chuyển thành chuổi Json như sau:

**@Override**

**public String toString() {**

**JSONObject jsonCity = new JSONObject();**

**JSONObject jsonLatLng = new JSONObject();**

**try {**

**jsonLatLng.put("lat", mPos.getLat());**

**jsonLatLng.put("lng", mPos.getLng());**

**jsonCity.put("name", mName);**

**jsonCity.put("latlng", jsonLatLng);**

**} catch (JSONException e) {**

**e.printStackTrace();**

}

return jsonCity.toString();

}

|  |  |
| --- | --- |
| 1  2 | City city = new City("Ho Chi Minh", new LatLng(10.887469,106.790446));  Log.d("JsonWorking", "onCreate: " + city.toString()); |

Và được kết quả :

**D/JsonWorking:onCreate: {"latlng":{"lng":106.790446,"lat":10.887469},"name":"Ho Chi Minh"}**

Việc đọc hay ghi JSON bằng thư viện có sẵn trong Android khá là dài dòng. Nhưng vì thế, chúng ta sẽ hiểu được các cơ chế của việc sử dụng JSON như thế nào.

1. **Thư viện Picasso**
   1. **Picasso là gì ?**

Picasso là một thư viện Android mã nguồn mở rất phổ biến, dùng để load hoặc hiển thị hình ảnh trong ứng dụng Android.

Lưu ý rằng Picasso sử dụng OkHttp (một thư viện network từ cùng một nhà phát triển) để load các hình ảnh qua internet.

* 1. **Tại sao sử dụng Picasso?**

Tự phát triển chức năng load và hiển thị hình ảnh của riêng bạn trong Java hoặc Kotlin là một nỗi đau thực sự: bạn phải quan tâm đến caching, decoding, quản lý kết nối mạng, luồng, xử lý exception và hơn thế nữa. Picasso là một thư viện dễ sử dụng, được tổ chức tốt, tài liệu đầy đủ và được testing kỹ lưỡng, giúp bạn tiết kiệm rất nhiều thời gian quý báu.

Dưới đây là nhiều lỗi phổ biến của việc tải và hiển thị hình ảnh trên Android được Picasso xử lý, dựa theo các tài liệu chính thức:

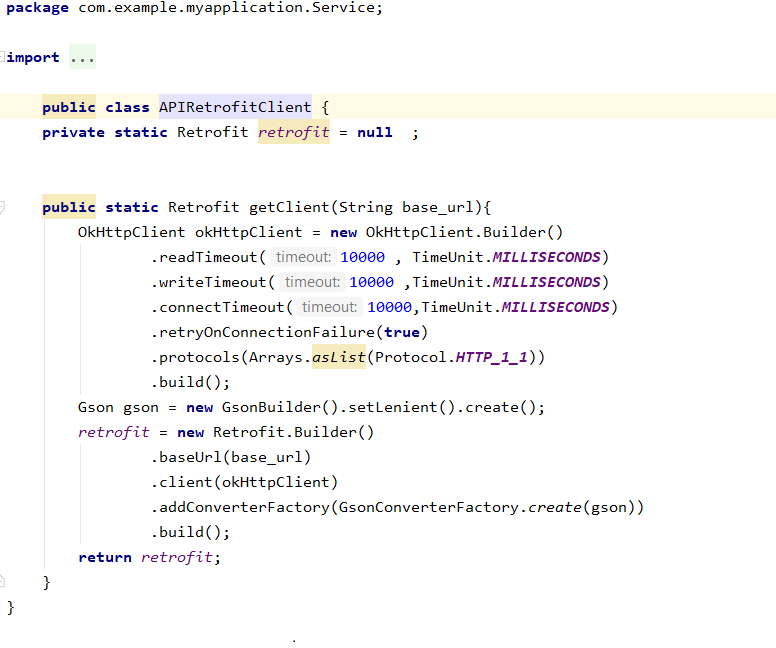
* Xử lý việc tái sử dụng ImageView trong adapter.
* Hiển thị hình ảnh phức tạp mà chỉ sử dụng bộ nhớ tối thiểu.
* Bộ nhớ đệm và bộ nhớ đĩa tự động.

Thêm hình ảnh vào ứng dụng của bạn có thể làm cho ứng dụng Android của bạn trở nên sống động. Vì vậy, trong hướng dẫn này, chúng ta sẽ tìm hiểu về Picasso 2 bằng cách xây dựng một ứng dụng thư viện hình ảnh đơn giản. Nó sẽ tải hình ảnh qua internet và hiển thị chúng dưới dạng hình thu nhỏ trong RecyclerView và khi người dùng nhấp vào hình ảnh, nó sẽ mở Activity chi tiết chứa hình ảnh lớn hơn.

CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ ỨNG DỤNG

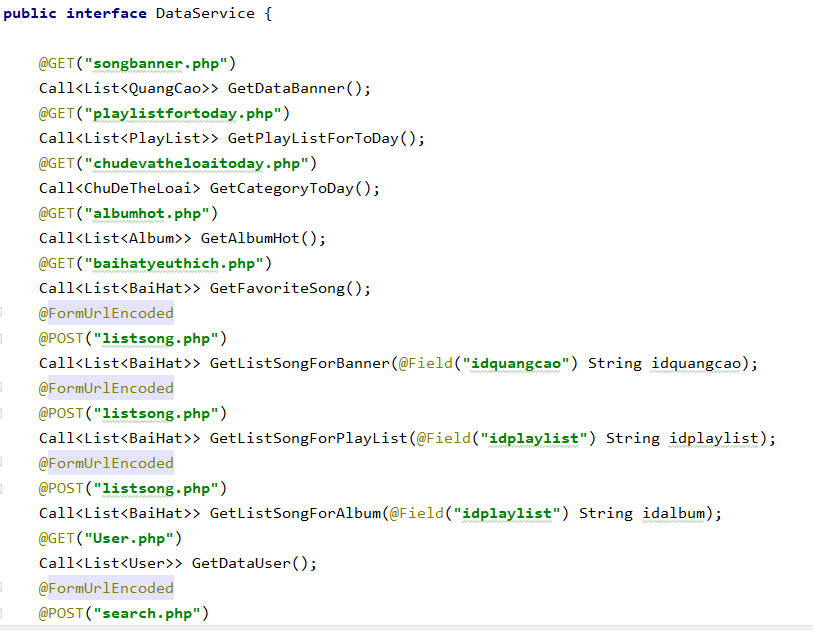
3.1 Cấu hình Service để tương tác với server

Chúng ta tiến hành cấu hình retrofit 2

****

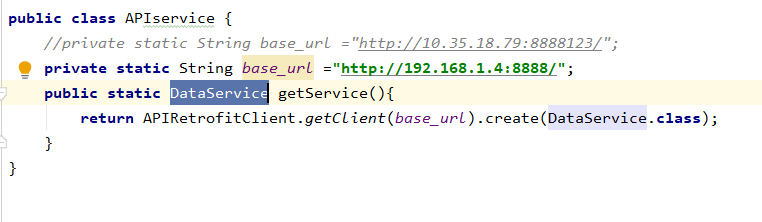
***Hình 3.1 Cấu hình retrofit***

-Sau đó ta khởi tạo các phương thức để tương tác với phần dữ liệu của server

****

***Hình 3.2 Các phương thức dùng để tương tác với server***

-Sau đó ta tiến hành tạo một phương thức để kết nối phần retrofit và các phương thức này lại

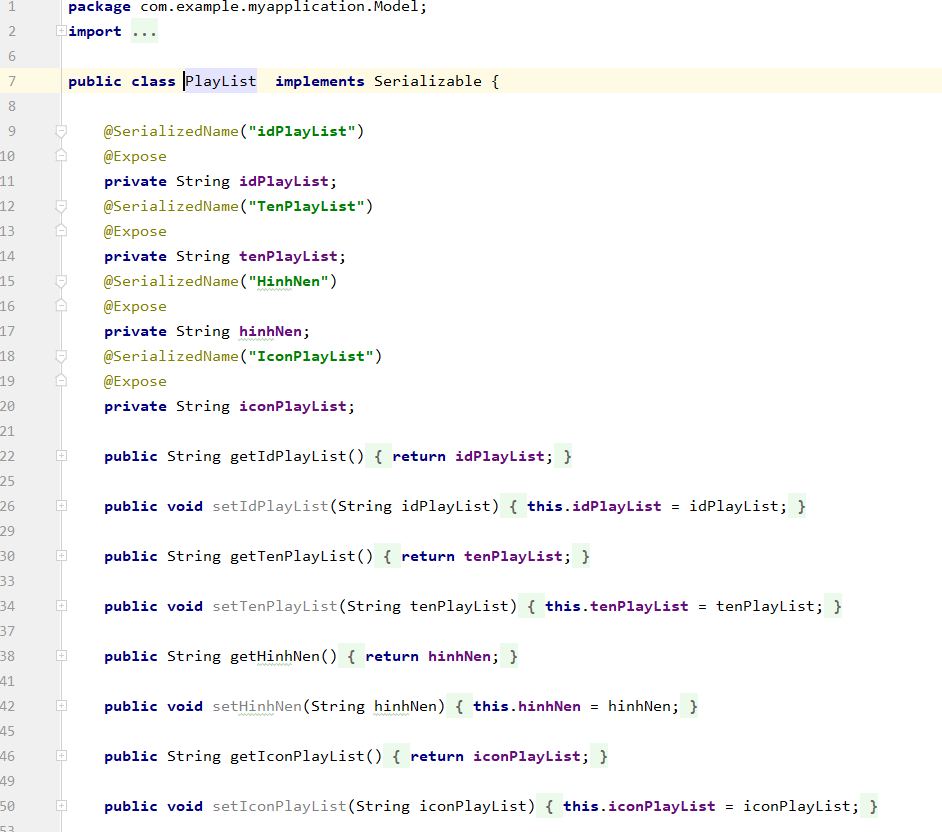
****

***Hình 3.3 Cầu nối giữa retrofit và các phương thức***

Ta tiến hành thiết kế các API để tương tác với server

Và cũng từ đó ta xây dựng được các model class cho các đối tượng





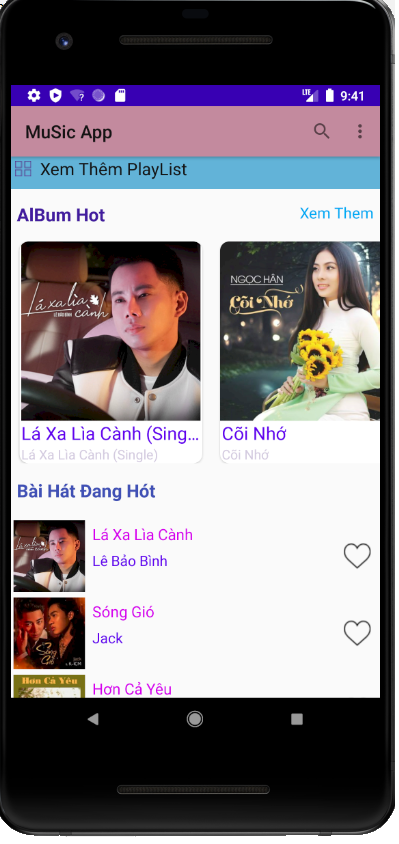
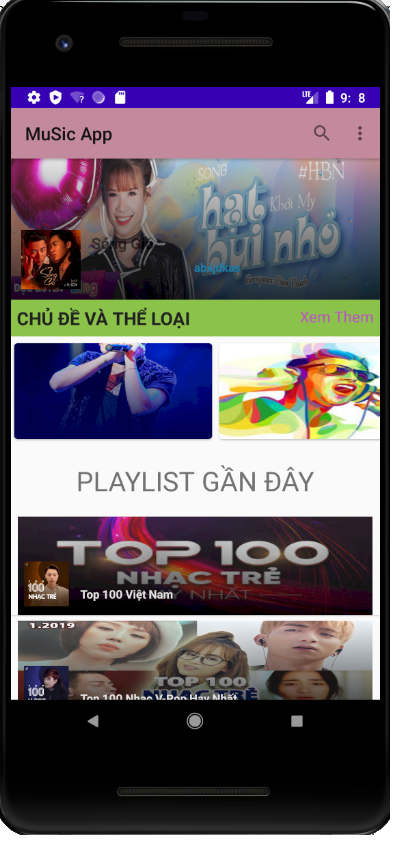
***Hình 3.4 Thiết kế class cho các đối tượng***

3.2 Thiết kế trang chủ

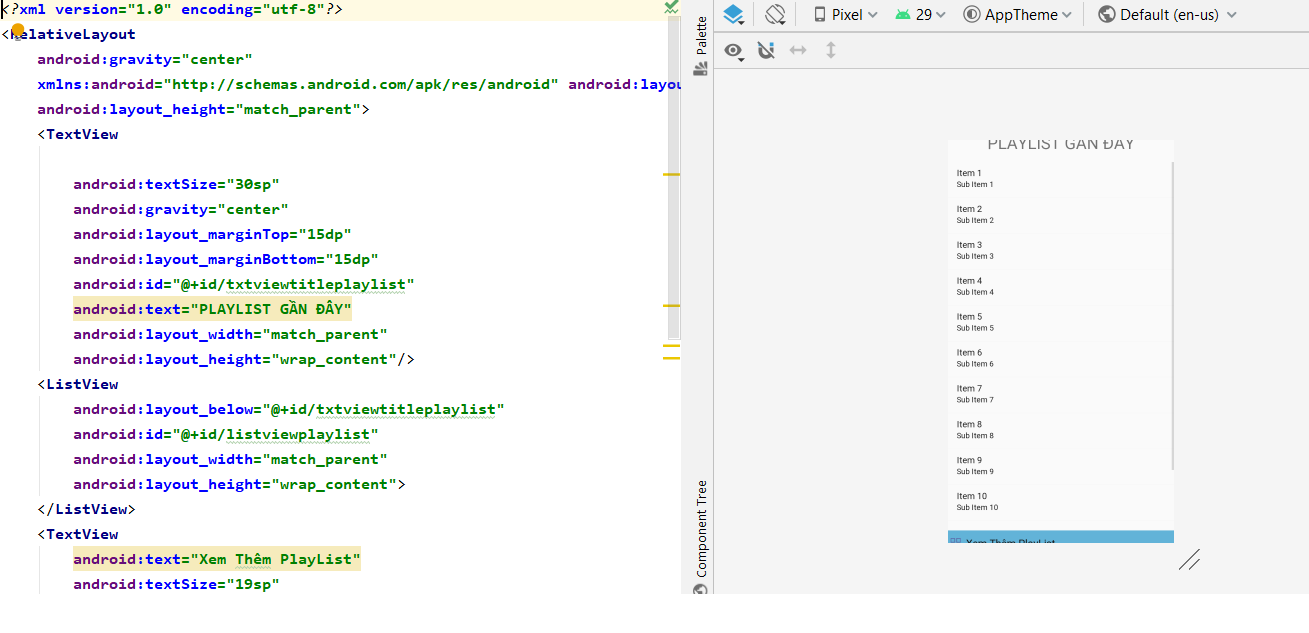
Trang chủ là 1 fragment được gắn lên MainActivity

Trong trang chủ có chứa nhiều fragment như fragment của banner,playlist hay album…

Ta tiến hành lấy dữ liệu từ server về trên mỗi fragment trên

****

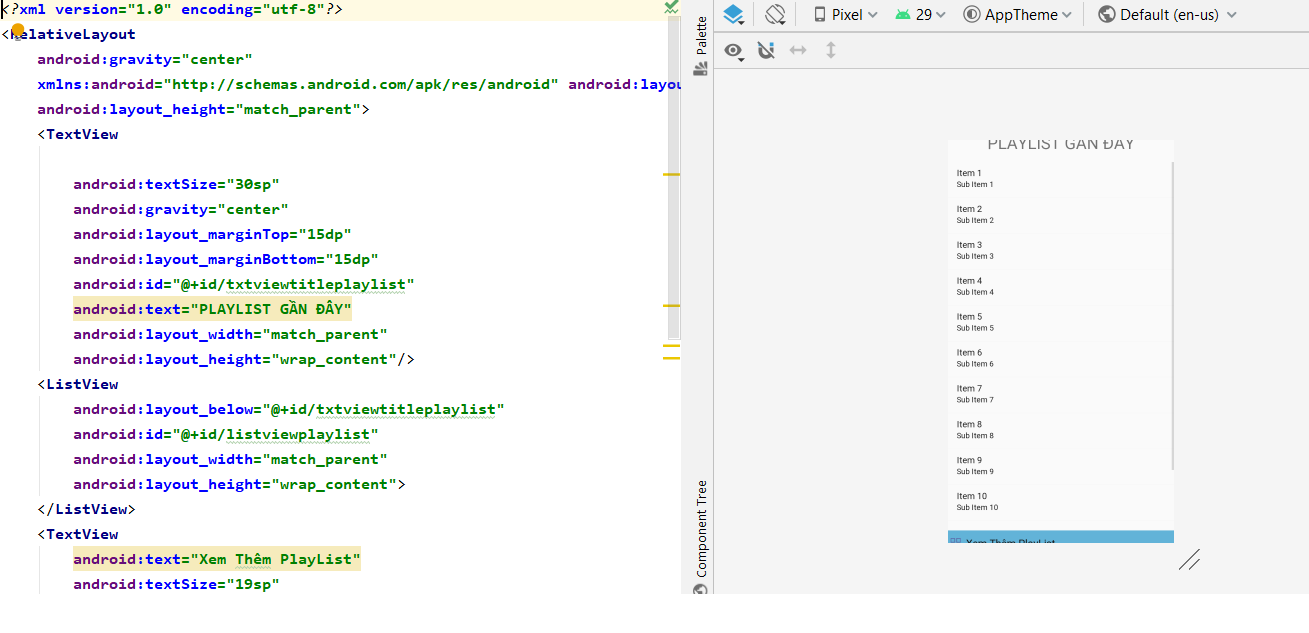
***Hình 3.5 Giao Diện Trang Chủ***



***Hình 3.6 Thiết kế phần giao diện cho fragment playlist***



***Hình 3.7 Gắn layout vào fragment và gắn adapter vào fragment***



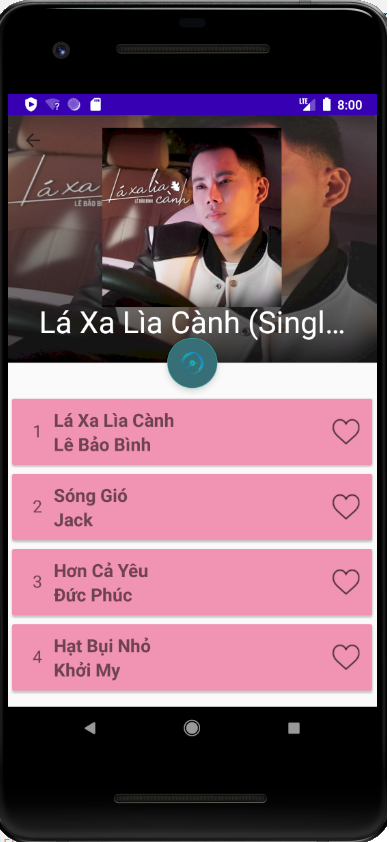
***Hình 3.8 Thiết kế layout cho adapter playlist***



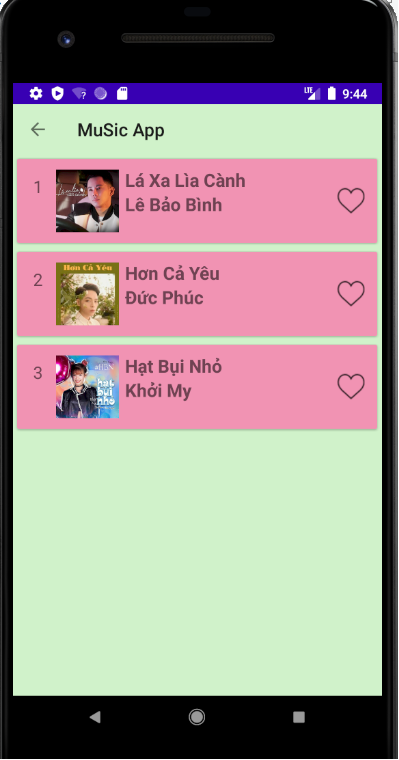
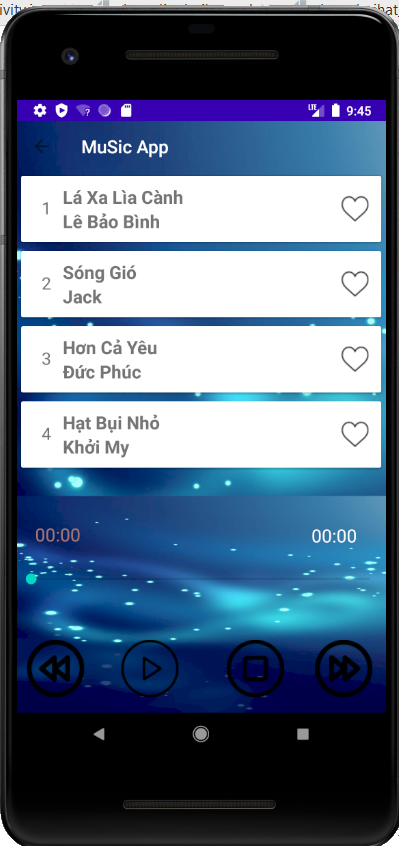
***Hình 3.9 Cấu hình adapter cho playlist***

3.3 Thiết kế cho các activity khác

-Tương tự như vậy ta tiến hành thiết kế cho các activity khác như activity đăng nhập,media play music,activity hiển thị các bài hát hay playlist

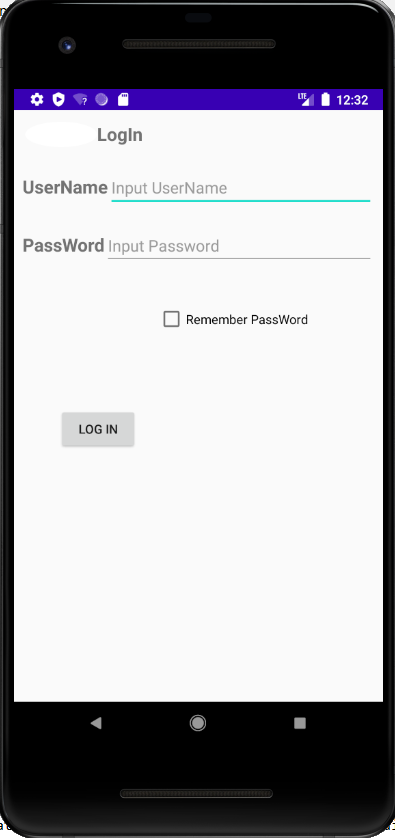
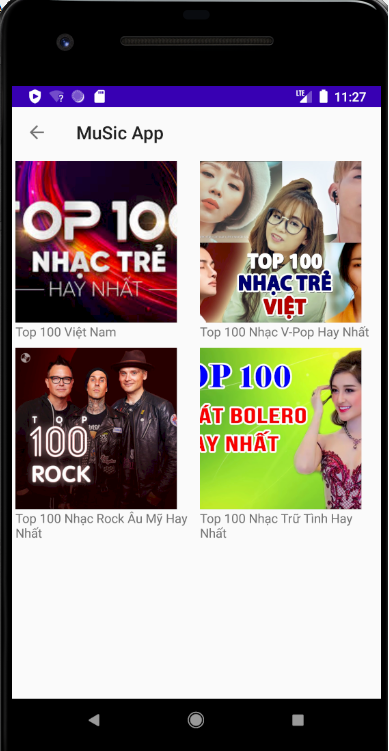


***Hình 3.10 Màn hình giao diện cho phần hiển thị các bài hát từ playlist hay album…***

 ***Hình 3.11 Màn hình media play music* *Hình 3.12 Màn hình hiển thị tìm kiếm bài hát***

-Sau khi người dung đăng nhập người dùng có thể chọn yêu thích với các bài hát cũng như tương tác với bài hát.Và người dung cũng có thể đăng xuất

**-**Nếu người dung muốn xem tất cả các playlist thì có thể nhấn vào Xem Thêm PlayList để có thể chọn PlayList mà mình yêu thích

****

***Hình 3.13 Màn hình đăng nhập cho user Hình 3.14 Màn hình hiển thị tất cả các Playlist***

CHƯƠNG 4 – TỔNG KẾT

4.KẾT QUẢ VÀ HƯỚNG PHÁT TRIỂN

4.1. SO SÁNH VỚI ỨNG DỤNG NGHE CỦA NHÓM

A. Kết quả đạt được:

* Giao diện hiển thị khá đẹp, dễ dùng và dễ tiếp cận.
* Người dùng có thể tương tác với bài hát
* Người dùng có thể nghe nhạc theo chủ đề ,playlist,thể loại…

B. Những điểm hạn chế:

* Giao diện còn khá đơn sơ chưa thu hút người dùng(hình ảnh cố định, chua có hình ảnh minh họa cho ca sĩ, bài hát,…).
* Một số tính năng như chia sẻ, tải nhạc,bình luận… vẫn chưa được hoàn thiện nên vẫn chưa áp dụng được cho sản phẩm.

4.2 HƯỚNG PHÁT TRIỂN.

* Phát triển sản phẩm hoàn chỉnh và tải lên Google Play.
* Cải thiện giao diện bằng cách thêm hình ảnh của ca sĩ, bài hát, cũng như cải thiện thêm về listview, áp dụng tablayout để nhìn chuyên nghiệp hơn.
* Thêm một số chức năng về video, M/V ca nhạc cho người dùng.
* Thêm tính năng thích, chia sẻ, tải nhạc,bình luận….

# **TÀI LIỆU THAM KHẢO**

[**https://genk.vn/mobile/lich-su-va-chang-duong-phat-trien-cua-android-qua-cac-phien-ban-phan-1-thoi-ky-dau-cua-android-20150608204249527.chn**](https://genk.vn/mobile/lich-su-va-chang-duong-phat-trien-cua-android-qua-cac-phien-ban-phan-1-thoi-ky-dau-cua-android-20150608204249527.chn)

[**https://fptshop.com.vn/tin-tuc/danh-gia/lich-su-phat-trien-cua-android-he-dieu-hanh-di-dong-pho-bien-nhat-the-gioi-55019**](https://fptshop.com.vn/tin-tuc/danh-gia/lich-su-phat-trien-cua-android-he-dieu-hanh-di-dong-pho-bien-nhat-the-gioi-55019)

[**https://www.vogella.com/tutorials/Retrofit/article.html**](https://www.vogella.com/tutorials/Retrofit/article.html)

[**https://viblo.asia/p/su-dung-thu-vien-picasso-trong-ung-dung-android-E375zRkd5GW**](https://viblo.asia/p/su-dung-thu-vien-picasso-trong-ung-dung-android-E375zRkd5GW)