

**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 : Ngô Văn Việt

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 TRÊN INTERNET**

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

Sinh viên : Ngô Văn Việt

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: Ngô Văn Việt

Mã SV: 1412101031

Lớp: CT1801

Ngành: Công nghệ thông tin

Tên đề tài: Xây dựng ứng dụng Android nghe nhạc trên internet

Mục lục

CHƯƠNG 1: GIỚI THIỆU HỆ ĐIỀU HÀNH ANDROID	6
1.1. Giới thiệu hệ điều hành Android	6
1.2. Lịch sử phát triển.....	7
1.3. Ứng dụng Android	8
1.4. Quản lý bộ nhớ Android	9
1.5. Lịch nâng cấp	9
1.6. Những đặc trưng của hệ điều hành Android	10
1.7. Kiến trúc trong hệ điều hành Android	11
Hình 1.1: Kiến trúc của Android.....	11
1.7.1. Nhân Linux Kernel	11
1.7.2. Thư viện	11
1.7.3. Thực thi ứng dụng Android.....	12
1.7.4. Khung ứng dụng	12
1.7.5. Tầng Ứng dụng.....	12
CHƯƠNG 2: MÔI TRƯỜNG LẬP TRÌNH ANDROID STUDIO	13
2.1. Sơ lược về Android Studio	13
2.2. Thiết lập môi trường	13
2.2.1 Cài đặt Andoid Studio	14
2.3. Cấu trúc dự án Android Studio	17
2.3.1. Tạo mới một project	17
2.3.2 Tạo một project.....	18
2.3.3. Các thành phần trong một ứng dụng Android	21
Bảng 2.1: Các thành phần trong ứng dụng Android	21
Hình 2.12: Mô hình ViewGroup	25
CHƯƠNG 3: CÁC KỸ THUẬT HỖ TRỢ.....	29
3.1. Webservice	29
3.1.1. Giới thiệu về Webservice	29
3.1.2. Đặc điểm của Webservice	29
3.1.3. Kiến trúc của Webservice.....	30
Hình 3.1: Kiến trúc của Web Service.....	31
3.1.4. Định dạng dữ liệu JSON.....	31

b) Giống nhau và khác nhau giữa JSON và XML	31
c) Cú pháp JSON	32
3.2. Dịch vụ web và kỹ thuật cơ sở dữ liệu trên internet	32
3.2.1. Hosting	33
3.2.2. Xây dựng webservice	42
3.3. MediaPlayer trong Android	44
3.3.1. Trạng thái phát nhạc	44
3.3.1. Permission.....	46
3.3.2. Callbacks.....	46
3.4. Các thư viện sử dụng thêm	47
3.5. Kỹ thuật lập trình trên Android.....	47
3.5.1. Kỹ thuật xây dựng tệp khai báo Manifest.....	47
3.5.2. Sử dụng retrofit để lấy dữ liệu từ API.....	48
Hình 3.20: Khai báo và tạo instance của retrofit	48
3.5.3. Các Activity trong ứng dụng.....	49
CHƯƠNG 4: CHƯƠNG TRÌNH THỰC NGHIỆM	55
4.1. Phát biểu bài toán.....	55
4.1.1 Các chức năng của chương trình.....	55
4.2 Sơ đồ Use-Case của chương trình.....	55
4.3 Cơ sở dữ liệu	56
4.4. Giao diện của chương trình.....	57
KẾT LUẬN.....	64
TÀI LIỆU THAM KHẢO	65

CHƯƠNG 1: GIỚI THIỆU HỆ ĐIỀU HÀNH ANDROID

1.1. Giới thiệu hệ điều hành Android

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. Android ra mắt vào năm 2007 cùng với tuyên bố thành lập Liên minh thiết bị cảm tay mở: một hiệp hội gồm các công ty phần cứng, phần mềm, và viễn thông với mục tiêu đẩy mạnh các tiêu chuẩn mở cho các thiết bị di động. Chiếc điện thoại đầu tiên chạy Android được bán vào tháng 10 năm 2008.

Android có mã nguồn mở và Google phát hành mã nguồn theo Giấy phép Apache. Chính mã nguồn mở cùng với một giấy phép không có nhiều ràng buộc đã cho phép các nhà phát triển thiết bị, mạng di động và các lập trình viên nhiệt huyết 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, bằng một loại ngôn ngữ lập trình Java có sửa đổi. Vào tháng 10 năm 2012, có khoảng 700.000 ứng dụng trên Android, và số lượt tải ứng dụng từ Google Play, cửa hàng ứng dụng chính của Android, ước tính khoảng 25 tỷ lượt.

Những yếu tố này đã giúp Android trở thành nền tảng điện thoại thông minh phổ biến nhất thế giới, vượt qua Symbian vào quý 4 năm 2010, và được các công ty công nghệ lựa chọn khi họ cần một hệ điều hành không nặng nề, có khả năng tinh chỉnh, và giá rẻ chạy trên các thiết bị công nghệ cao thay vì tạo dựng từ đầu. Kết quả là mặc dù được thiết kế để chạy trên điện thoại và máy tính bảng, Android đã xuất hiện trên TV, máy chơi game và các thiết bị điện tử khác. Bản chất mở của Android cũng khích lệ một đội ngũ đông đảo lập trình viên và những người đam mê sử dụng mã nguồn mở để tạo ra những dự án do cộng đồng quản lý. Những dự án này bổ sung các tính năng cao cấp cho những người dùng thích tìm tòi hoặc đưa Android vào các thiết bị ban đầu chạy hệ điều hành khác.

Android chiếm 75% thị phần điện thoại thông minh trên toàn thế giới vào thời điểm quý 3 năm 2012, với tổng cộng 500 triệu thiết bị đã được kích hoạt và 1,3 triệu lượt kích hoạt mỗi ngày. Sự thành công của hệ điều hành cũng khiến nó trở thành mục tiêu trong các vụ kiện liên quan đến bằng phát minh, góp mặt trong cái gọi là "cuộc chiến điện thoại thông minh" giữa các công ty công nghệ.^[1]

1.2. Lịch sử phát triển

Tổng công ty Android (Android, Inc.) được thành lập tại Palo Alto, California vào tháng 10 năm 2003 bởi Andy Rubin (đồng sáng lập công ty Danger), Rich Miner (đồng sáng lập Tổng công ty Viễn thông Wildfire), Nick Sears (từng là Phó giám đốc T-Mobile), và Chris White (trưởng 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 được vị trí và sở thích của người dùng". Dù những người thành lập và nhân viên đều là những người 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 người bạn thân của Rubin, mang cho ông 10.000 USD tiền mặt nhưng 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 thương vụ này. Vào thời điểm đó không có nhiều thông tin về Tổng công ty, nhưng nhiều người đồn đoán rằng Google dự tính tham gia thị trường điện thoại di động sau bước 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 như đối tác phần mềm, bắt 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 điện thoại di động xuất hiện trong tháng 12 năm 2006. Tin tức của BBC và Nhật báo phố Wall chú thích rằng Google muốn đưa công nghệ tìm kiếm và các ứng dụng của họ vào điện thoại di động và họ đang nỗ lực làm việc để thực hiện điều này. Các phương tiện truyền thông truyền thống lẫn online cũng viết về tin đồn rằng Google đang phát triển một thiết bị cầm tay mang thương hiệu Google. Một vài tờ báo còn nói rằng trong khi Google vẫn đang thực hiện những bản mô tả kỹ thuật chi tiết, họ đã trình diễn sản phẩm mẫu cho các nhà sản xuất điện thoại di động và nhà mạng. Tháng 9 năm 2007, InformationWeek đăng tải một nghiên cứu của Evalueserve cho biết Google đã nộp một số đơn xin cấp bằng sáng chế trong lĩnh vực điện thoại di động.

Ngày 5 tháng 11 năm 2007, Liên minh thiết bị cầm tay mở (Open Handset Alliance), một hiệp hội bao gồm nhiều công ty trong đó có Texas Instruments, Tập đoàn Broadcom,

Google, HTC, Intel, LG, Tập đoàn Marvell Technology, Motorola, Nvidia, Qualcomm, Samsung Electronics, Sprint Nextel và T-Mobile được thành lập với mục đích phát triển các tiêu chuẩn mở cho thiết bị di động. Cùng ngày, Android cũng được ra mắt với vai trò là sản phẩm đầu tiên của Liên minh, một nền tảng thiết bị di động được xây dựng trên nhân Linux phiên bản 2.6. Chiếc điện thoại chạy Android đầu tiên được bán ra là HTC Dream, phát hành ngày 22 tháng 10 năm 2008. Biểu trưng của hệ điều hành Android mới là một con rôbốt màu xanh lá cây do hãng thiết kế Irina Blok tại California vẽ.

Từ năm 2008, Android đã trải qua nhiều lần cập nhật để dần dần cải tiến hệ điều hành, bổ sung các tính năng mới và sửa các lỗi trong những lần phát hành trước. Mỗi bản nâng cấp được đặt tên lần lượt theo thứ tự bảng chữ cái, theo tên của một món ăn tráng miệng; ví dụ như phiên bản 1.5 Cupcake (bánh bông lan nhỏ có kem) tiếp nối bằng phiên bản 1.6 Donut (bánh vòng). Phiên bản mới nhất hiện nay là 9.0 với tên gọi là Android Pie.

Vào năm 2010, Google ra mắt loạt thiết bị Nexus-một dòng sản phẩm bao gồm điện thoại thông minh và máy tính bảng chạy hệ điều hành Android, do các đối tác phần cứng sản xuất. HTC đã hợp tác với Google trong chiếc điện thoại thông minh Nexus đầu tiên, Nexus One. Kể từ đó nhiều thiết bị mới hơn đã gia nhập vào dòng sản phẩm này, như điện thoại Nexus 4 và máy tính bảng Nexus 10, lần lượt do LG và Samsung sản xuất. Google xem điện thoại và máy tính bảng Nexus là những thiết bị Android chủ lực của mình, với những tính năng phần cứng và phần mềm mới nhất của Android.

1.3. Ứng dụng Android

Android có lượng ứng dụng của bên thứ ba ngày càng nhiều, được chọn lọc và đặt trên một cửa hàng ứng dụng như Google Play hay Amazon Appstore để người dùng lấy về, hoặc bằng cách tải xuống rồi cài đặt tập tin APK từ trang web khác. Các ứng dụng trên Cửa hàng Play cho phép người dùng duyệt, tải về và cập nhật các ứng dụng do Google và các nhà phát triển thứ ba phát hành. Cửa hàng Play được cài đặt sẵn trên các thiết bị thỏa mãn điều kiện tương thích của Google. Ứng dụng sẽ tự động lọc ra một danh sách các ứng dụng tương thích với thiết bị của người dùng, và nhà phát triển có thể giới hạn ứng dụng của họ chỉ dành cho những nhà mạng cố định hoặc những quốc gia cố định vì lý do kinh doanh. Nếu người dùng mua một ứng dụng mà họ cảm thấy không thích, họ được hoàn trả tiền sau 15 phút kể từ lúc tải về, và một vài nhà mạng còn có khả năng mua giúp các ứng dụng trên Google Play, sau đó tính tiền vào trong hóa đơn sử dụng hàng tháng của người dùng. Đến tháng 9 năm 2012, có hơn 675.000 ứng dụng dành cho Android, và số lượng ứng dụng tải về từ Cửa hàng Play ước tính đạt 25 tỷ.

Các ứng dụng cho Android được phát triển bằng ngôn ngữ Java sử dụng Bộ phát triển phần mềm Android (SDK). SDK bao gồm một bộ đầy đủ các công cụ dùng để phát triển, gồm có công cụ gỡ lỗi, thư viện phần mềm, bộ giả lập điện thoại dựa trên QEMU, tài liệu hướng dẫn, mã nguồn mẫu, và hướng dẫn từng bước. Môi trường phát triển tích hợp (IDE) được hỗ trợ chính thức là Eclipse sử dụng phần bổ sung Android Development Tools (ADT). Các công cụ phát triển khác cũng có sẵn, gồm có Bộ phát triển gốc dành cho các ứng dụng hoặc phần mở rộng viết bằng C hoặc C++, Google App Inventor, một môi trường đồ họa cho những nhà lập trình mới bắt đầu, và nhiều nền tảng ứng dụng web di động đa nền tảng phong phú.

1.4. Quản lý bộ nhớ Android

Vì Hệ điều hành Android được thiết kế để quản lý bộ nhớ (RAM) để giảm tối đa mức tiêu thụ điện năng, trái với hệ điều hành máy tính để bàn luôn cho rằng máy tính sẽ có nguồn điện không giới hạn. Khi một ứng dụng Android không còn được sử dụng, hệ thống sẽ tự động ngưng nó trong bộ nhớ - trong khi ứng dụng về mặt kỹ thuật vẫn "mở", những ứng dụng này sẽ không tiêu thụ bất cứ tài nguyên nào (như năng lượng pin hay năng lượng xử lý) và nằm đó cho đến khi nó được cần đến. Cách làm như vậy có lợi kép là vừa làm tăng khả năng phản hồi nói chung của thiết bị Android, vì ứng dụng không nhất phải đóng rồi mở lại từ đầu, vừa đảm bảo các ứng dụng nền không làm tiêu hao năng lượng một cách không cần thiết.

Hệ điều hành Android quản lý các ứng dụng trong bộ nhớ một cách tự động: khi bộ nhớ thấp, hệ thống sẽ bắt đầu diệt ứng dụng và tiến trình không hoạt động được một thời gian, sắp theo thời điểm cuối mà chúng được sử dụng (tức là cũ nhất sẽ bị tắt trước). Tiến trình này được thiết kế ẩn đi với người dùng, để người dùng không cần phải quản lý bộ nhớ hoặc tự tay tắt các ứng dụng. Tuy nhiên, sự che giấu này của hệ thống quản lý bộ nhớ Android đã dẫn đến sự thịnh hành của các ứng dụng tắt chương trình của bên thứ ba trên cửa hàng Google Play; những ứng dụng kiểu như vậy được cho là có hại nhiều hơn có lợi.

1.5. Lịch nâng cấp

Google đưa ra các bản nâng cấp lớn Cho Android theo chu kỳ từ 6 đến 9 tháng, mà phần lớn thiết bị đều có thể nhận được qua sóng không dây. Bản nâng cấp lớn mới nhất là Android 6.0 Marshmallow. So với các hệ điều hành cạnh tranh khác, như iOS, các bản nâng cấp Android thường mất thời gian lâu hơn để đến với các thiết bị. Với những thiết bị không thuộc dòng Nexus, các bản nâng cấp thường đến sau vài tháng kể từ khi phiên bản

được chính thức phát hành. Nguyên nhân của việc này một phần là do sự phong phú về phần cứng của các thiết bị Android, nên người ta phải mất thời gian điều chỉnh bản nâng cấp Cho phù hợp, vì mã nguồn chính thức của Google chỉ chạy được trên những thiết bị Nexus chủ lực của họ. Chuyển Android sang những phần cứng cụ thể là một quy trình tốn thời gian và công sức của các nhà sản xuất thiết bị, những người luôn ưu tiên các thiết bị mới nhất và thường bỏ rơi các thiết bị cũ hơn. Do đó, những chiếc điện thoại thông minh thế hệ cũ thường không được nâng cấp nếu nhà sản xuất quyết định rằng nó không đáng để bỏ thời gian, bất kể chiếc điện thoại đó có khả năng chạy bản nâng cấp hay không. Vấn đề này còn trầm trọng hơn khi những nhà sản xuất điều chỉnh Android để đưa giao diện và ứng dụng của họ vào, những thứ này cũng sẽ phải làm lại Cho mỗi bản nâng cấp. Sự chậm trễ còn được đóng góp bởi nhà mạng, sau khi nhận được bản nâng cấp từ nhà sản xuất, họ còn điều chỉnh thêm Cho phù hợp với nhu cầu rồi thử nghiệm kỹ lưỡng trên hệ thống mạng của họ trước khi chuyển nó đến người dùng. Việc thiếu các hỗ trợ hậu mãi của nhà sản xuất và nhà mạng đã bị những nhóm người dùng và các trang tin công nghệ chỉ trích rất nhiều. Một số người viết còn nói rằng giới công nghiệp do cái lợi về tài chính đã cố tình không nâng cấp thiết bị, vì nếu thiết bị hiện tại không cập nhật sẽ thúc đẩy việc mua thiết bị mới, một thái độ được coi là “xúc phạm”. The Guardian đã than phiền rằng phương cách phân phối bản nâng cấp trở nên phức tạp chính vì những nhà sản xuất và nhà mạng đã cố tình làm nó như thế. Vào năm 2011, Google đã hợp tác cùng một số hãng công nghiệp và ra mắt “Liên minh nâng cấp Android”, với lời hứa sẽ nâng cấp thường xuyên Cho các thiết bị trong vòng 18 tháng sau khi ra mắt. Tính đến năm 2012, người ta không còn nghe nhắc đến liên minh này nữa.

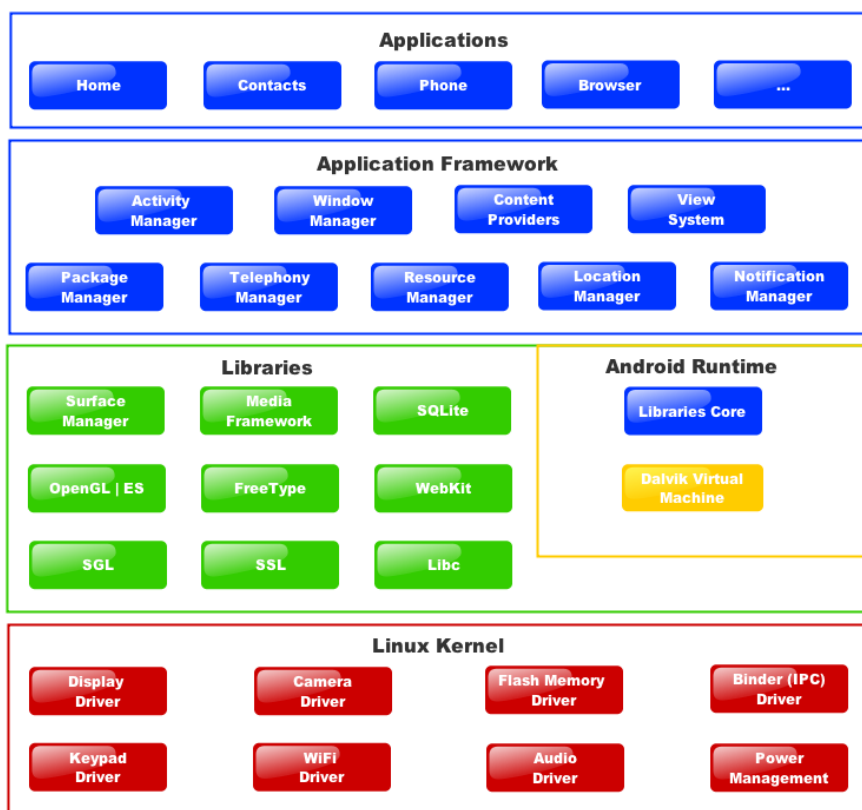
1.6. Những đặc trưng của hệ điều hành Android

- **Application framework:** cho phép sử dụng lại và thay thế các thành phần trong lập trình ứng dụng.
- **Dalvik virtual machine:** tối ưu hóa cho thiết bị di động.
- **Intergrated browser:** trình duyệt tích hợp, dựa trên cơ chế WebKit mã nguồn mở.
- **SQLite:** cơ sở dữ liệu trong môi trường di động.
- **Media support:** hỗ trợ các định dạng audio, video và hình ảnh thông dụng.
- **GSM Telephony:** mạng điện thoại di động (phụ thuộc vào phần cứng).
- **Bluetooth, EDGE, 3G, 4G và WiFi :** các chuẩn kết nối dữ liệu (phụ thuộc vào phần cứng).
- **Camera, GPS, la bàn, và gia tốc kế:** (phụ thuộc vào phần cứng).

- **Môi trường phát triển phong phú:** bao gồm thiết bị mô phỏng, công cụ cho việc dò tìm lỗi, bộ nhớ và định hình hiệu năng và một plugin cho Android Studio.[4]

1.7. Kiến trúc trong hệ điều hành Android

Hệ điều hành Android có thể coi như một ngăn xếp chứa các thành phần của phần mềm, được chia làm các phần như sau:



Hình 1.1: Kiến trúc của Android

1.7.1. Nhân Linux Kernel

Ở dưới cùng của các lớp là Linux-Linux 2.6. Nhân Linux cung cấp chức năng cơ bản như hệ thống quản lý tiến trình, quản lý bộ nhớ, quản lý thiết bị như máy ảnh, bàn phím, màn hình hiển thị, vv...

1.7.2. Thư viện

Phần đầu của nhân Linux có một tập hợp các thư viện bao gồm cả mã nguồn mở trình duyệt web WebKit, các thư viện phổ biến, cơ sở dữ liệu SQLite có thể được coi như một kho lưu trữ hữu ích cho việc lưu trữ và chia sẻ dữ liệu ứng dụng, thư viện để chạy các ứng dụng như ghi âm và video, SSL thư viện chịu trách nhiệm về an ninh Internet, vv...

1.7.3. Thực thi ứng dụng Android

Đây là phần thứ ba của kiến trúc và có sẵn trên lớp thứ hai từ dưới lên. Phần này cung cấp một thành phần quan trọng được gọi là Dalvik (có thể đã thay đổi theo phiên bản) còn được gọi là máy ảo. Máy ảo là một loại máy Java được thiết kế đặc biệt và tối ưu hóa cho Android.

Máy ảo Dalvik sử dụng các tính năng cốt lõi Linux như quản lý bộ nhớ và đa luồng, nội tại trong ngôn ngữ Java. Máy ảo Dalvik cho phép tất cả các ứng dụng Android chạy trong tiến trình riêng của nó, với trường hợp riêng của các máy ảo Dalvik.

Thực thi Android cũng cung cấp một tập hợp các thư viện lõi cho phép các nhà phát triển ứng dụng Android sử dụng ngôn ngữ lập trình Java để viết các ứng dụng của mình.

1.7.4. Khung ứng dụng

Khung ứng dụng cung cấp nhiều dịch vụ cấp cao hơn cho các ứng dụng dưới dạng các lớp Java. Nhà phát triển ứng dụng được phép sử dụng các dịch vụ này trong các ứng dụng của họ.

1.7.5. Tầng Ứng dụng

Người dùng sẽ tìm thấy tất cả các ứng dụng Android ở lớp trên cùng. Người dùng sẽ viết ứng dụng và các ứng dụng đó phải được cài đặt trên lớp này.

CHƯƠNG 2: MÔI TRƯỜNG LẬP TRÌNH ANDROID STUDIO

2.1. Sơ lược về Android Studio

Google cung cấp một công cụ phát triển ứng dụng Android trên Website chính thức dựa trên nền tảng IntelliJ IDEA gọi là Android Studio. Android studio dựa vào IntelliJ IDEA, là một IDE tốt cho nhất Java hiện nay. Do đó Android Studio sẽ là môi trường phát triển ứng dụng tốt nhất cho ứng dụng Android.[5]

2.2. Thiết lập môi trường

Hai thành phần cơ bản nhất mà chúng ta cần phải có để lập trình Android là Bộ phát triển Java(Java Development Kit) và Bộ phát triển phần mềm(Software Development Kit).Bộ phát triển Java dùng để tạo ra môi trường thực thi máy ảo cho hệ điều hành mà chúng ta đang sử dụng. Bộ phát triển phần mềm chứa các phiên bản Android, các hàm API cần thiết, mã nguồn minh họa cũng như các công cụ hỗ trợ lập trình khác. Mỗi khi Google ra phiên bản Android mới thì Bộ phát triển phần mềm cũng được cập nhật tương ứng.

- Cài đặt Java

Để cài đặt Bộ phát triển Java(JDK), ta cần truy cập vào trang Oracle JDK và tải phiên bản mới nhất để tăng tính ổn định và tận dụng được nhiều tính năng hỗ trợ tốt nhất.



Hình 2.1 Cài đặt JDK

2.2.1 Cài đặt Android Studio

a) Yêu cầu phần cứng máy tính

- Microsoft® Windows® 10/8/7 (32 or 64-bit)
- 4 GB RAM trở lên.
- 400 MB hard disk space + ít nhất 1GB cho Android SDK, emulator system images và caches
- Độ phân giải tối thiểu 1280 x 800
- Java Development Kit

b) Phần mềm android studio

- Vào đường dẫn: "<http://developer.android.com/sdk/index.html>"
- Để download bản mới nhất và tiến hành cài đặt click như hình:

androidstudio

Android Studio provides the fastest tools for building apps on every type of Android device.

DOWNLOAD ANDROID STUDIO

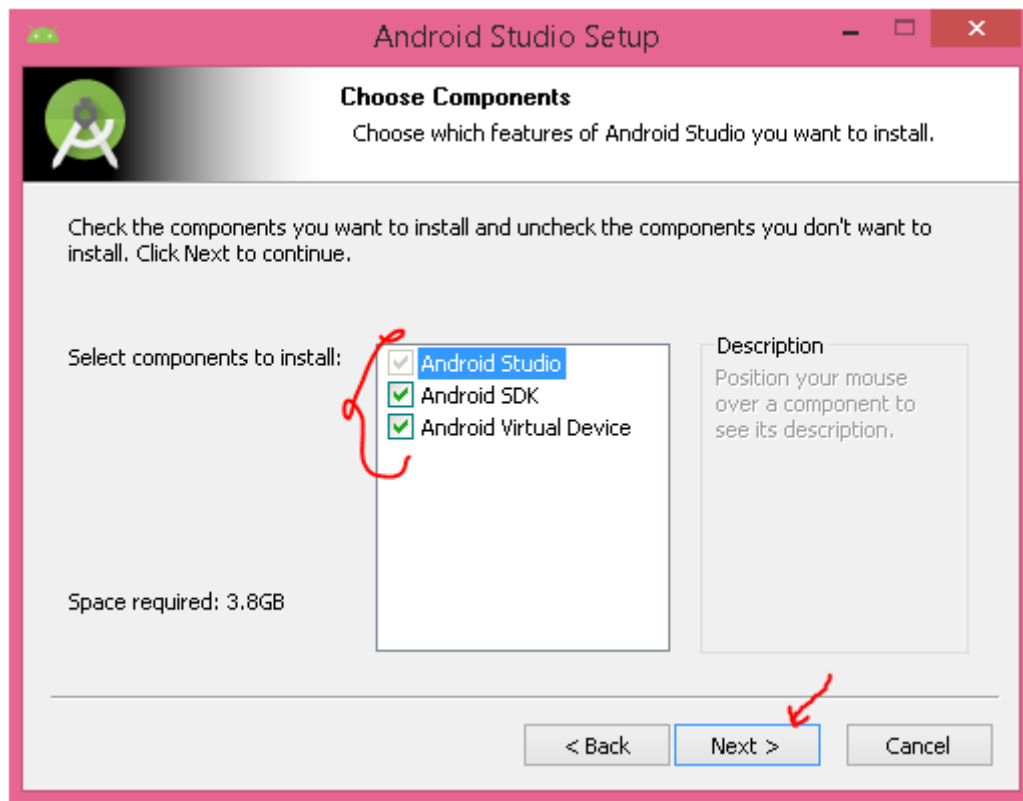
3.1.4 for Windows 64-bit (790 MB)

DOWNLOAD OPTIONS

RELEASE NOTES

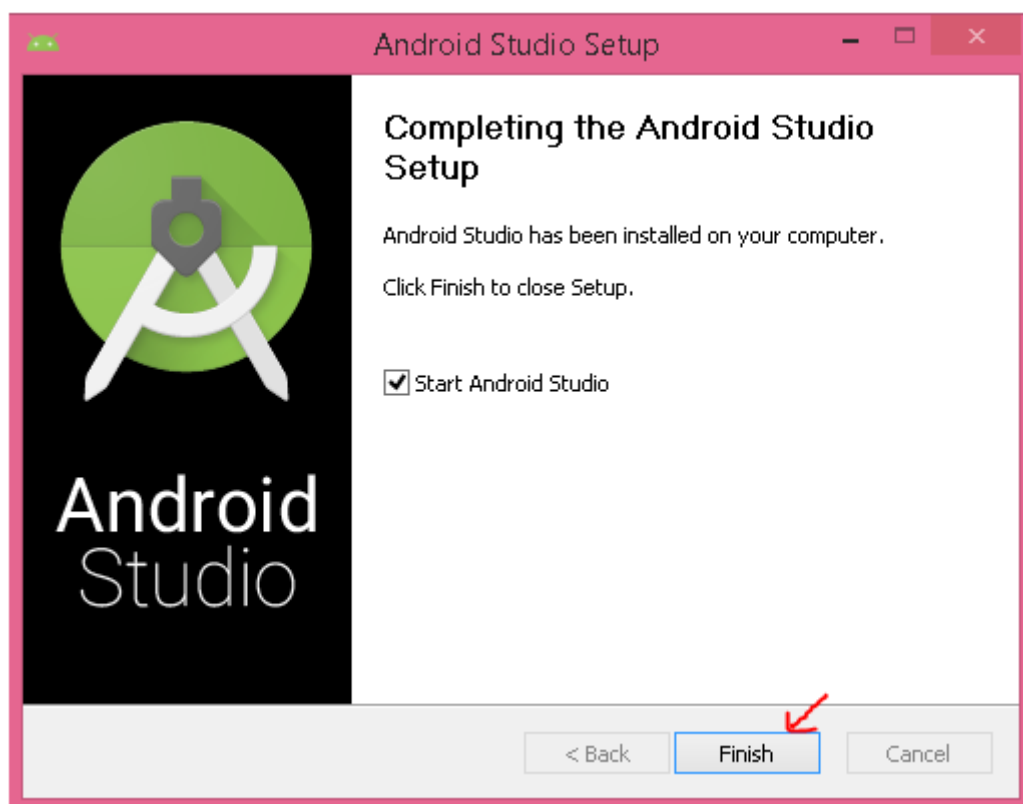
Hình 2.2: Download android studio

- Khi cài đặt chú ý chọn cả SDK và trình giả lập thiết bị android ảo như hình:



Hình 2.3: Cài đặt SDK

- Tiếp tục chọn Next và Agree cho đến khi hoàn tất.



Hình 2.4: Kết thúc cài đặt

- Màn hình khởi động Android Studio

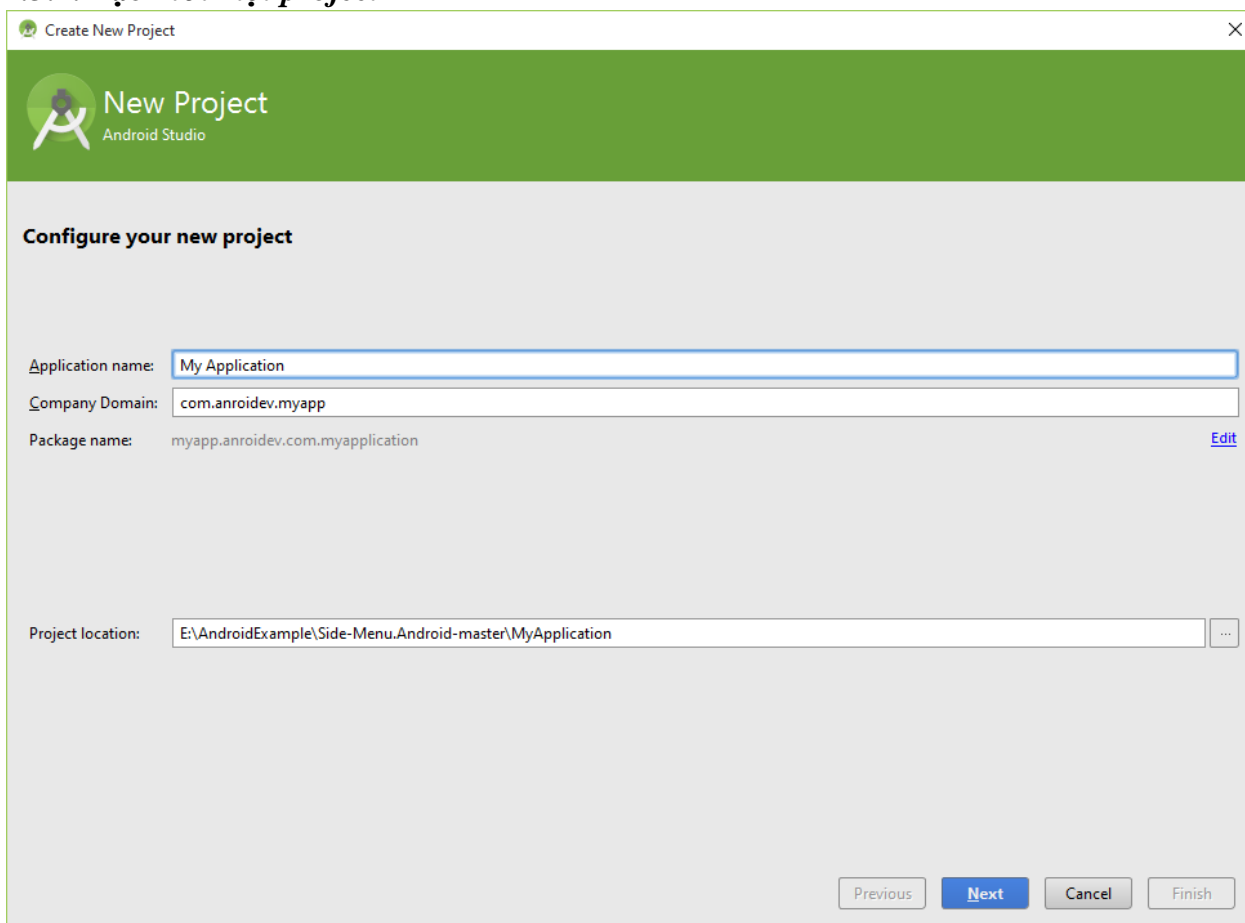


Hình 2.5: Màn hình khởi động android studio

- Chúng ta có thể sử dụng trực tiếp máy ảo trên android studio hoặc kết nối trực tiếp máy điện thoại có hệ điều hành Android để chạy chương trình thực nghiệm.

2.3. Cấu trúc dự án Android Studio

2.3.1. Tạo mới một project

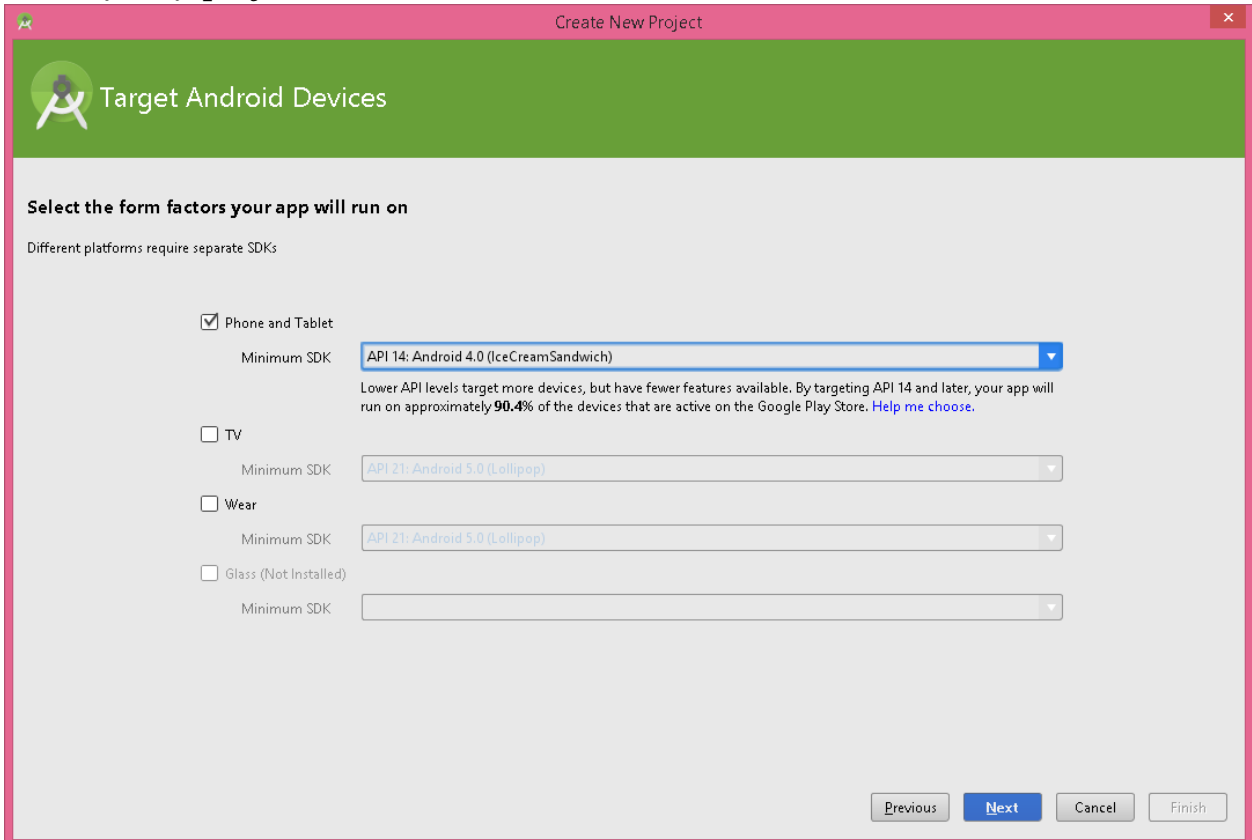


Hình 2.6: Tạo mới project

- Application Name: Tên ứng dụng muốn đặt
- Company Domain: Tên domain công ty, thường được dùng để kết hợp với tên Application để tạo thành Package (chú ý viết thường hết và có ít nhất 1 dấu chấm).
- Package name: Nó sẽ tự động nối ngược Company Domain với Application name.
- Project location: Là nơi lưu trữ ứng dụng.

Sau đó nhập Next để tiếp tục.

2.3.2 Tạo một project.

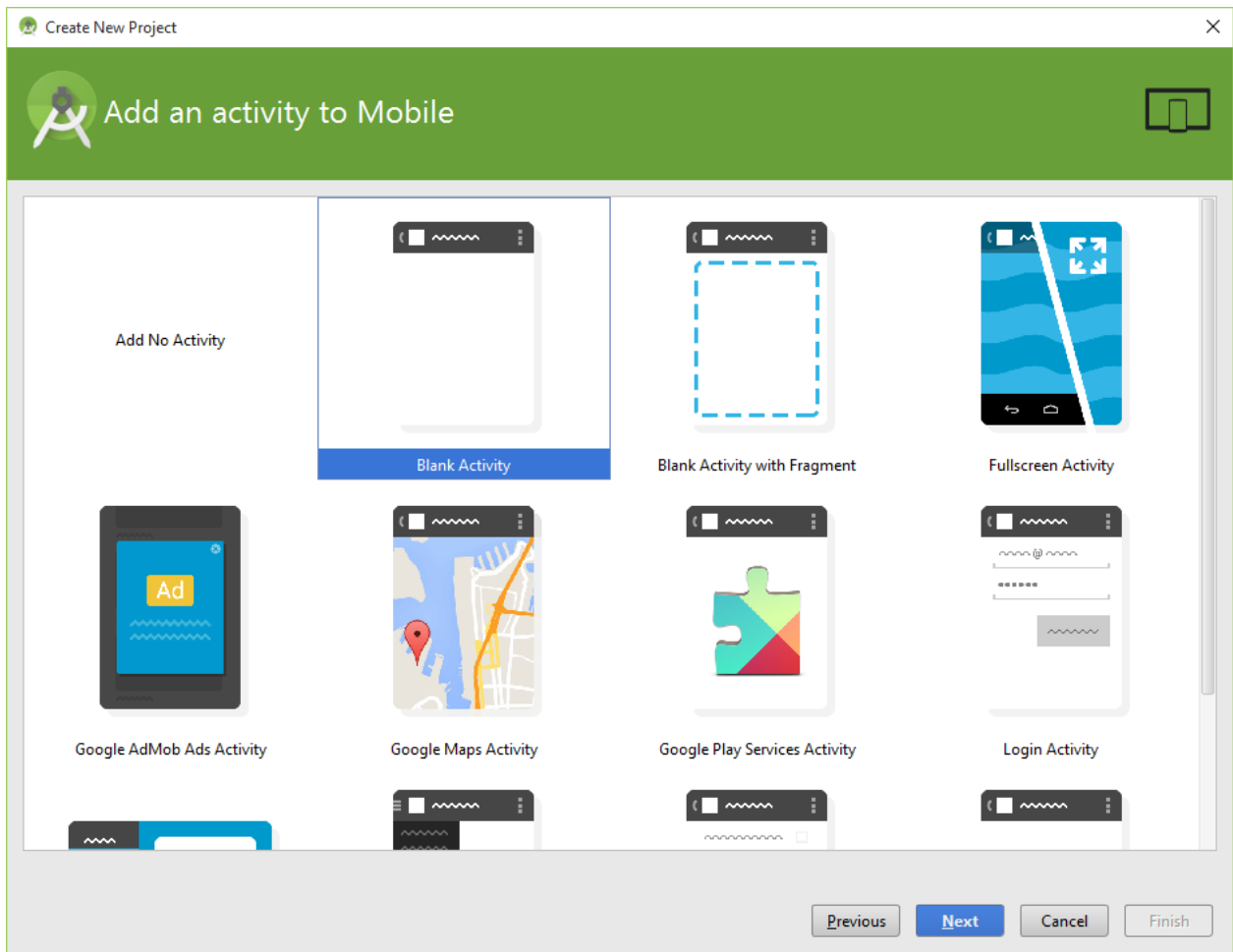


Hình 2.7: Lựa chọn thiết bị android và phiên bản SDK tối thiểu

Ở hộp thoại trên cho phép ta lựa chọn là ứng dụng sẽ được viết cho những thiết bị nào (Phone and Tablet, TV, Wear).

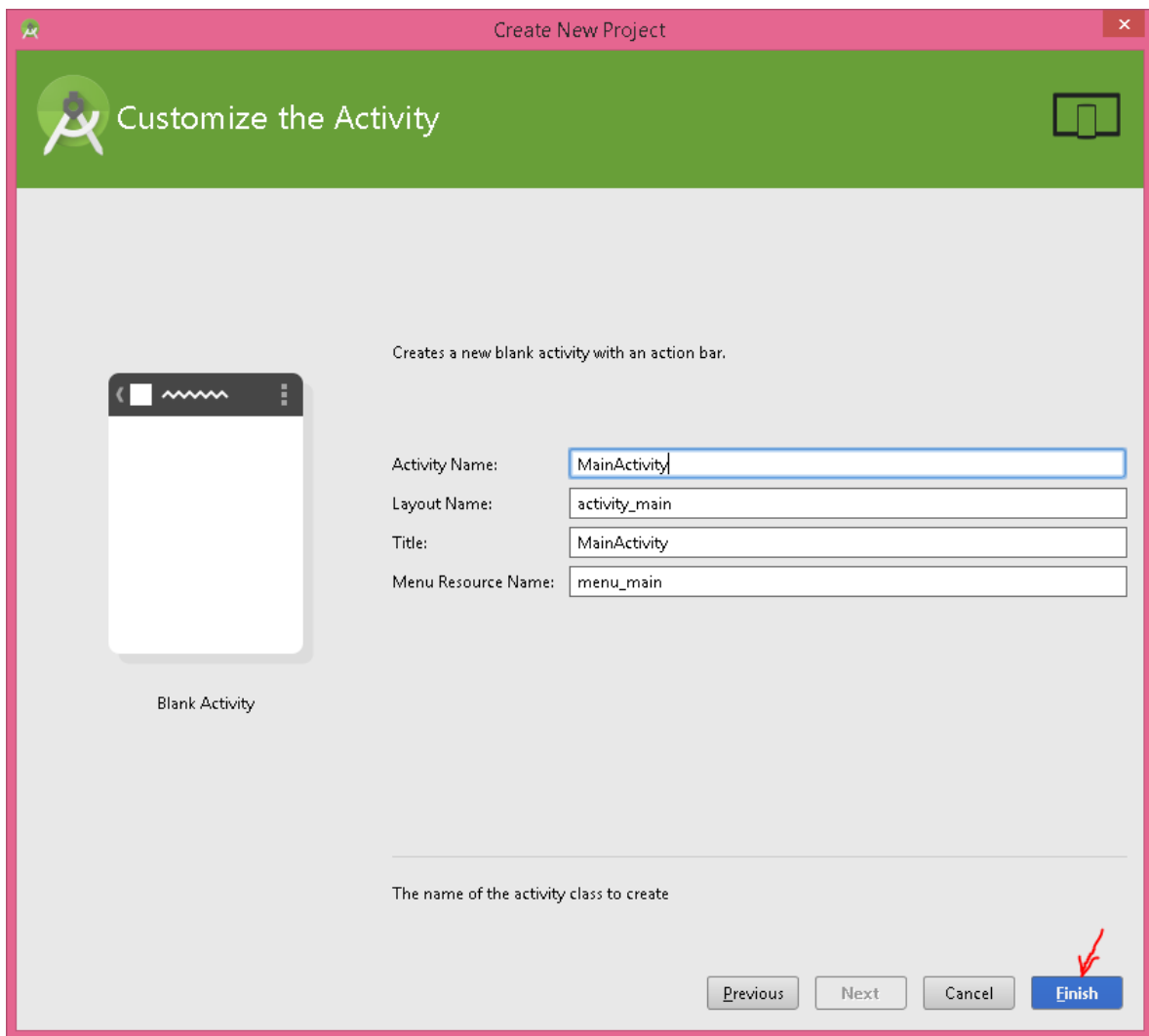
Ở mục Minimum SDK, quy định phiên bản android tối thiểu để chạy ứng dụng.

Màn hình này hiển thị cho phép chọn loại Activity mặc định.



Hình 2.8: Màn hình hiển thị Activity mặc định

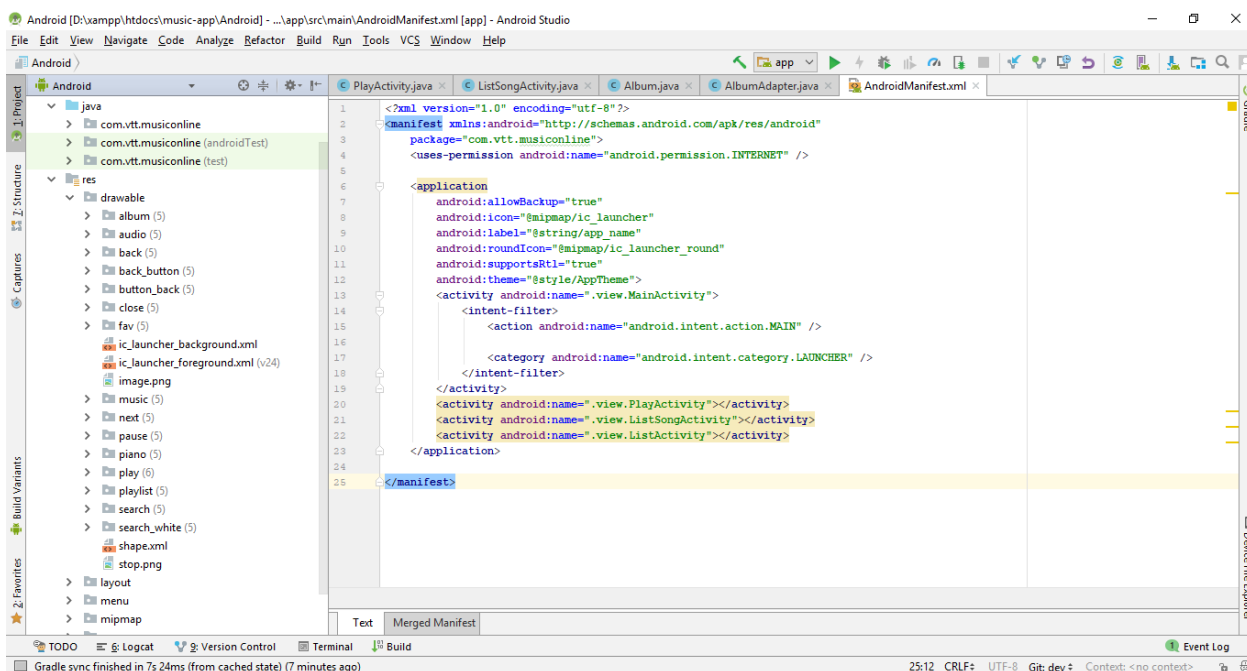
Chọn Blank Activity rồi bấm Next:



Hình 2.9: Blank Activity

- Activity Name: Tên class Activity (java) để ta viết mã lệnh
- Layout Name: Tên file XML làm giao diện cho Activity Name.
- Title: Tiêu đề hiển thị khi kích hoạt Activity trên thiết bị.
- Menu Resource Name: Tên file xml để tạo menu cho phần mềm.

Sau khi cấu hình xong, bấm Finish, Màn hình Build Gradle project hiển thị:



Hình 2.10: Màn hình làm việc của Android Studio

2.3.3. Các thành phần trong một ứng dụng Android

Thành phần ứng dụng là các khối cơ bản để xây dựng một ứng dụng Android. Các thành phần này được liên kết với các ứng dụng bởi tập tin AndroidManifest.xml, tập tin AndroidManifest.xml mô tả mỗi thành phần của ứng dụng và cách chúng tương tác với nhau.

Có 7 thành phần chính có thể được sử dụng trong một ứng dụng Android:

Bảng 2.1: Các thành phần trong ứng dụng Android

Thành phần	Đặc tả
Activities	Gọi giao diện người dùng và xử lý các tương tác người dùng với màn hình điện thoại.
Services	Xử lý nền kết hợp với một ứng dụng.
Broadcast Receivers	Xử lý thông tin liên lạc giữa hệ điều hành Android và các ứng dụng.
Content Providers	Xử lý dữ liệu và các vấn đề quản lý cơ sở dữ

	liệu.
View	Nơi hiển thị giao diện người dùng.
Intent	Cung cấp khả năng giao tiếp giữa các màn hình với nhau.
Tập tin khai báo (Manifest File)	Khai báo các thành phần của ứng dụng.

a) *Activities*

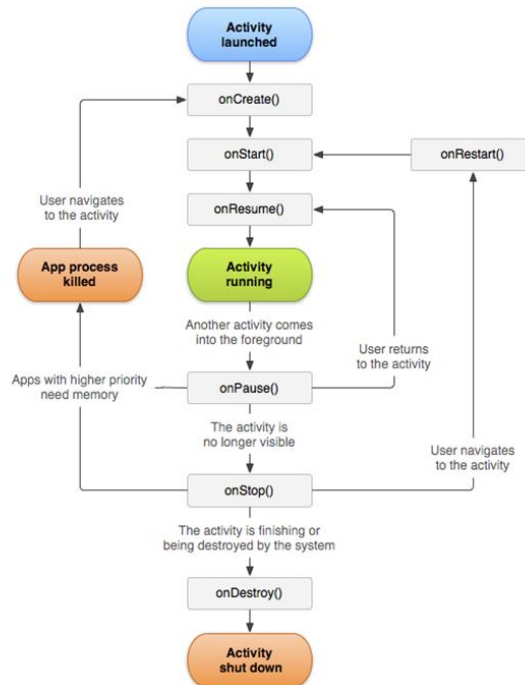
Một *Activities* đại diện cho một màn hình duy nhất với một giao diện người dùng. Ví dụ, một ứng dụng email có thể có một hoạt động cho thấy một danh sách các email mới, một hoạt động để soạn một email, và một hoạt động để đọc email. Nếu một ứng dụng có nhiều hơn một hoạt động, sau đó một trong số chúng được đánh dấu là hoạt động được hiển thị khi ứng dụng được khởi chạy.

Một *Activities* được thực hiện như một lớp con của lớp *Activity* như sau:

```
public class MainActivity extends Activity {...}
```

Với mỗi *Activity* thường vòng đời có 3 trạng thái sau:

- **Running** (đang kích hoạt)
- **Paused** (tạm dừng)
- **Stopped** (dừng – không phải Destroyed)



Hình 2.11: Vòng đời của ứng dụng Android

1, Running (đang kích hoạt): Khi màn hình là Foreground(Activity nằm trên cùng ứng dụng và cho phép người sử dụng tương tác).

2, Paused (tạm dừng) : Activity bị mất focus nhưng mà vẫn nhìn thấy được Activity này (Ví dụ bạn mở một Activity mới lên dưới dạng Dialog). Trường hợp này nó vẫn có khả năng bị hệ thống tự động Destroy trong tình huống bộ nhớ quá ít.

3, Stopped (dừng – không phải Destroyed): Activity mất focus và không nhìn thấy được (ví dụ bạn mở một Activity mới lên mà Full màn hình chẳng hạn). Trong trường hợp này nó có thể bị hệ thống Destroy trong bất kỳ tình huống nào.

Như vậy cả **Paused** hay **Stopped** đều có khả năng bị **Destroyed** (hủy) khi bộ nhớ cần cho việc khác ưu tiên hơn.

a) Services

Một Services là một thành phần chạy trong nền để thực hiện các hoạt động lâu dài. Ví dụ, một dịch vụ có thể chơi nhạc ở chế độ nền trong khi người dùng đang ở một ứng dụng khác nhau, hoặc nó có thể lấy dữ liệu qua mạng mà không ngăn chặn người dùng tương tác với một hoạt động.

Một Service được thực hiện như một lớp con của lớp Service như sau:

```
public class MyService extends Service { ... }
```

i. Broadcast Receivers

Broadcast Receivers chỉ đơn giản là phản ứng để phát các tín hiệu từ các ứng dụng khác hoặc từ hệ thống. Ví dụ, các ứng dụng cũng có thể bắt đầu chương trình phát tín hiệu để cho các ứng dụng khác biết rằng một số dữ liệu đã được tải về điện thoại và sẵn sàng cho họ sử dụng.

Một máy thu phát tín hiệu được thực hiện như một lớp con của BroadcastReceiver lớp và mỗi tín hiệu được phát đi như một đối tượng Intent.

```
public class MyReceiver extends BroadcastReceiver { ... }
```

ii. Content Providers

Content Providers cung cấp nội dung dữ liệu từ một ứng dụng khác theo yêu cầu. Yêu cầu đó được xử lý bằng các phương thức (methods) của lớp

ContentResolver. Dữ liệu có thể được lưu trữ trong hệ thống tập tin, cơ sở dữ liệu (database) hoặc ở một nơi hoàn toàn khác.

Một Content Providers được thực hiện như một lớp con của ContentProvider lớp và phải thực hiện một bộ tiêu chuẩn API cho phép các ứng dụng khác để thực hiện các giao dịch.

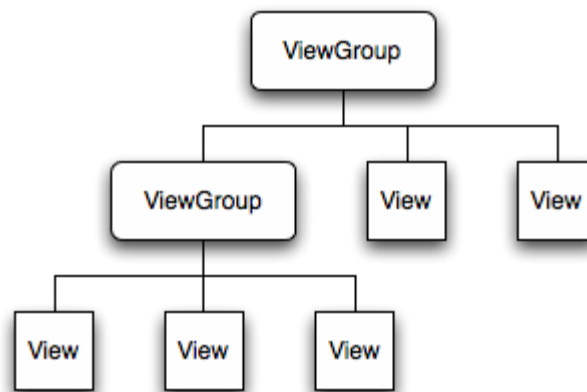
```
public class MyContentProvider extends ContentProvider { ... }
```


iii. View

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. Mỗi một kiểu là một con của class View và 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 phân cấp như hình dưới. Một màn hình là một tập hợp các Layout và các widget được bố trí có thứ tự. Để thể hiện một màn hình thì trong hàm onCreate của mỗi Activity cần phải được gọi một hàm là setContentView(R.layout.main), hàm này sẽ load giao diện từ file XML lên để phân tích thành mã bytecode.



Hình 2.12: Mô hình ViewGroup

iv. Intent

Content Provider được kích hoạt khi chúng được gọi từ một ContentResolver. Ba thành phần khác (hoạt động, dịch vụ và bộ nhận quảng bá) được kích hoạt bởi thông điệp không đồng bộ từ các Intent. Một Intent là một đối tượng có kiểu Intent chứa nội dung của thông điệp. Với các hoạt động và dịch vụ, nó gọi tên hành động được yêu cầu và xác định URI của dữ liệu tác động tới ở giữa. Ví dụ, nó có thể truyền tải một yêu cầu cho một hoạt động hiển thị một ảnh cho người dùng hay cho phép người dùng sửa văn bản. Với bộ nhận quảng bá, đối tượng Intent gọi tên của hành động được thông báo. Ví dụ, bộ nhận quảng bá có thể thông báo các phần nó quan tâm là nút chụp ảnh đã được bấm.

Có vài phương thức cho việc kích hoạt mỗi thành phần:

Một hoạt động được khởi chạy thông qua một đối tượng Intent `Context.startActivity()` hay `Activity.startActivityForResult()`. Hoạt động đáp lại có thể theo dõi Intent được tạo ra đó bằng phương thức `getIntent()` và cập nhật thông qua phương thức `setIntent(Intent)`. Android gọi phương thức `onNewIntent()` để bỏ qua các Intent đến trước nó.

Một hoạt động thường bắt đầu hoạt động khác. Nếu nó muốn trả lại kết quả hoạt động nó đã khởi chạy, nó sẽ gọi phương thức: `startActivityForResult()` thay cho phương thức `startActivity()`. Ví dụ, nếu nó khởi chạy một hoạt động mà cho phép người dùng lấy một ảnh, nó có thể muốn lấy kết quả của ảnh được chọn. Kết quả được trả về trong một đối tượng Intent thông qua phương thức `onActivityResult()`.

Một dịch vụ được bắt đầu thông qua một đối tượng Intent là `Context.startService()`. Android gọi phương thức `onStart()` của dịch vụ và thông qua đối tượng Intent của nó.

Tương tự, một Intent có thể thông qua `Context.bindService()` để thiết lập một kết nối liên tục giữa các thành phần và dịch vụ đích. Dịch vụ nhận đối tượng Intent qua lời gọi `onBind()` (nếu dịch vụ chưa được chạy, `bindService()` có thể chọn bắt đầu nó). Cho ví dụ, một hoạt động có thể thiết lập kết nối với dịch vụ chơi nhạc để cập ở phần trước để nó có thể cung cấp cho người dùng giao diện sử dụng để điều khiển chơi lại. Hoạt động sẽ gọi `bindService` để thiết lập kết nối và sau đó gọi phương thức đã định nghĩa bởi dịch vụ để áp dụng chơi lại ca khúc.

Một ứng dụng có thể khởi tạo một quảng bá thông qua đối tượng Intent bằng phương thức như: `Context.setBroadcast()`, `Context.setOrderedBroadcast()`,

và `Context.sendStickyBroadcast()`. Android chuyển những Intent tới tất cả các bộ nhận quảng bá nào quan tâm bằng việc gọi phương thức `onReceive()` của nó.

v. Tập tin khai báo

Trước khi có thể khởi chạy một ứng dụng thành phần, nó phải xem ứng dụng bao gồm những thành phần nào. Và đóng gói các thành phần đó lại vào trong Android (tập tin `.apk` chứa các mã nguồn, tập tin và tài nguyên).

Tập tin này có cấu trúc của tập tin XML và luôn có tên là `AndroidManifest.xml` trong mọi ứng dụng. Tập tin này thực hiện một số chức năng như thêm và khai báo các thành phần của ứng dụng, tên các thư viện ứng dụng cần liên kết tới (ngoài thư viện chuẩn của Android) và xác định các quyền cho ứng dụng.

Tuy nhiên, nhiệm vụ chính của tập tin khai báo là khai báo các thành phần của ứng dụng. Một hoạt động có thể được khai báo như sau:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest . . . >
<application . . . >
<activity android:name="com.example.project.FreneticActivity"
android:icon="@drawable/small_pic.png"
android:label="@string/freneticLabel"
. . . >
</activity>
. . .
</application>
</manifest>
```

Thuộc tính “name” của phần tử <activity> là tên các lớp con lớp Activity đã được cài đặt, thuộc tính “icon” và “label” trỏ đến tập tin tài nguyên chứa biểu tượng và nhãn được hiển thị cho người dùng.

Các thành phần khác được khai báo theo cách tương tự: <service> dịch vụ, <receiver> bộ nhận quảng bá và <provider> content provider. Các hoạt động, dịch vụ và content provider có thể cùng được khai báo trong tập tin khai báo hoặc có thể được tạo tự động trong mã (như đối tượng BroadcastReceiver) và được đăng ký với hệ thống bằng cách gọi Context.registerReceiver().

CHƯƠNG 3: CÁC KỸ THUẬT HỖ TRỢ

3.1. Webservice

3.1.1. Giới thiệu về Webservice

Theo định nghĩa của W3C (World Wide Web Consortium), dịch vụ Web là một hệ thống phần mềm được thiết kế để hỗ trợ khả năng tương tác giữa các ứng dụng trên các máy tính khác nhau thông qua mạng Internet, giao diện chung và sự gắn kết của nó được mô tả bằng XML. Dịch vụ Web là tài nguyên phần mềm có thể xác định bằng địa chỉ URL, thực hiện các chức năng và đưa ra các thông tin người dùng yêu cầu. Một dịch vụ Web được tạo nên bằng cách lấy các chức năng và đóng gói chúng sao cho các ứng dụng khác dễ dàng nhìn thấy và có thể truy cập đến những dịch vụ mà nó thực hiện, đồng thời có thể yêu cầu thông tin từ dịch vụ Web khác. Nó bao gồm các mô đun độc lập cho hoạt động của khách hàng và doanh nghiệp và bản thân nó được thực thi trên server.

Trước hết, có thể nói rằng ứng dụng cơ bản của dịch vụ Web là tích hợp các hệ thống và là một trong những hoạt động chính khi phát triển hệ thống. Trong hệ thống này, các ứng dụng cần được tích hợp với cơ sở dữ liệu và các ứng dụng khác, người sử dụng sẽ giao tiếp với cơ sở dữ liệu để tiến hành phân tích và lấy dữ liệu. Trong thời gian gần đây, việc phát triển mạnh mẽ của thương mại điện tử và B2B cũng đòi hỏi các hệ thống phải có khả năng tích hợp với cơ sở dữ liệu của các đối tác kinh doanh (nghĩa là tương tác với hệ thống bên ngoài – bên cạnh tương tác với các thành phần bên trong của hệ thống trong doanh nghiệp).[6]

3.1.2. Đặc điểm của Webservice

Theo định nghĩa của W3C (World Wide Web Consortium), dịch vụ Web là một hệ thống phần mềm được thiết kế để hỗ trợ khả năng tương tác giữa các ứng dụng trên các máy tính khác nhau thông qua mạng Internet, giao diện chung và sự gắn kết của nó được mô tả bằng XML. Dịch vụ Web là tài nguyên phần mềm có thể xác định bằng địa chỉ URL, thực hiện các chức năng và đưa ra các thông tin người dùng yêu cầu. Một dịch vụ Web được tạo nên bằng cách lấy các chức năng và đóng gói chúng sao cho các ứng dụng khác dễ dàng nhìn thấy và có thể truy cập đến những dịch vụ mà nó thực hiện, đồng thời có thể yêu cầu thông tin từ dịch vụ Web khác. Nó bao gồm các mô đun độc lập cho hoạt động của khách hàng và doanh nghiệp và bản thân nó được thực thi trên server.

Trước hết, có thể nói rằng ứng dụng cơ bản của dịch vụ Web là tích hợp các hệ thống và là một trong những hoạt động chính khi phát triển hệ thống. Trong hệ thống này,

các ứng dụng cần được tích hợp với cơ sở dữ liệu và các ứng dụng khác, người sử dụng sẽ giao tiếp với cơ sở dữ liệu để tiến hành phân tích và lấy dữ liệu. Trong thời gian gần đây, việc phát triển mạnh mẽ của thương mại điện tử và B2B cũng đòi hỏi các hệ thống phải có khả năng tích hợp với cơ sở dữ liệu của các đối tác kinh doanh (nghĩa là tương tác với hệ thống bên ngoài – bên cạnh tương tác với các thành phần bên trong của hệ thống trong doanh nghiệp).

3.1.3. Kiến trúc của Webservice

Dịch vụ Web có 3 chuẩn chính:

- SOAP (Simple Object Access Protocol): cách mà Web Service sử dụng để truyền tải dữ liệu.

- WSDL (Web Service Description Language): dùng để mô tả các thông tin cần thiết của một web service. Nó được tổ chức thành 1 file có cấu trúc được thống nhất, không phân biệt ngôn ngữ sử dụng.

- UDDI (Universal Description, Discovery, and Integration). UDDI được sử dụng để đăng ký và khám phá dịch vụ Web đã được miêu tả cụ thể trong WSDL. Giao tác UDDI sử dụng SOAP để nói chuyện với UDDI server, sau đó các ứng dụng SOAP yêu cầu một dịch vụ Web. Các thông điệp SOAP được gửi đi chính xác bởi HTTP và TCP/IP.

Webservice có 4 thành phần chính:

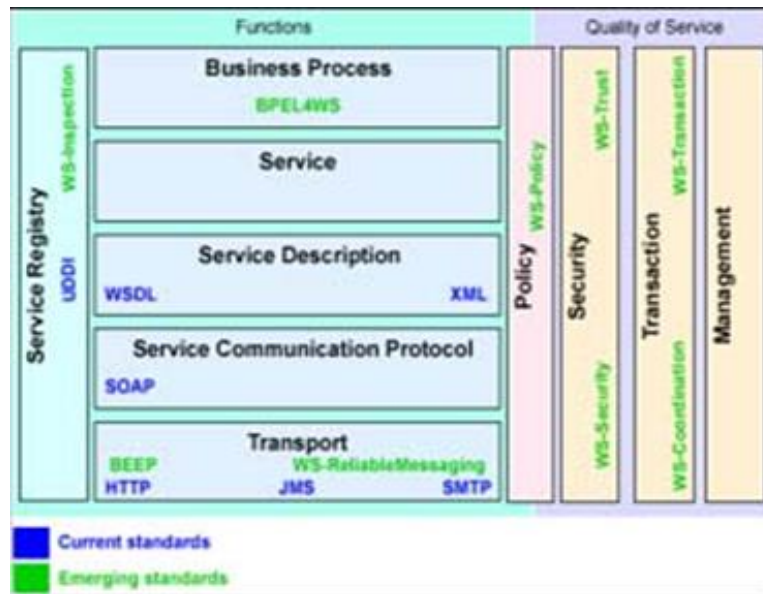
Dịch vụ vận chuyển (Service Transport): có nhiệm vụ truyền thông điệp giữa các ứng dụng mạng, bao gồm những giao thức như HTTP, SMTP, FTP, JSM và gần đây nhất là giao thức thay đổi khối mở rộng (Blocks Extensible Exchange Protocol- BEEP).

Thông điệp XML: có nhiệm vụ giải mã các thông điệp theo định dạng XML để có thể hiểu được ở mức ứng dụng tương tác với người dùng. Hiện tại, những giao thức thực hiện nhiệm vụ này là XML-RPC, SOAP và REST.

Mô tả dịch vụ: được sử dụng để miêu tả các giao diện chung cho một dịch vụ Web cụ thể. WSDL thường được sử dụng cho mục đích này, nó là một ngôn ngữ mô tả giao tiếp và thực thi dựa trên XML. Dịch vụ Web sẽ sử dụng ngôn ngữ này để truyền tham số và các loại dữ liệu cho các thao tác và chức năng mà dịch vụ Web cung cấp.

Khám phá dịch vụ: tập trung dịch vụ vào trong một nơi được đăng ký, từ đó giúp một dịch vụ Web có thể dễ dàng khám phá ra những dịch vụ nào đã có trên mạng, tốt hơn trong việc tìm kiếm những dịch vụ khác để tương tác. Một dịch vụ Web cũng phải tiến

hành đăng ký để các dịch vụ khác có thể truy cập và giao tiếp. Hiện tại, UDDI API thường được sử dụng để thực hiện công việc này.



Hình 3.1: Kiến trúc của Web Service

Trong đó, tầng giao thức tương tác dịch vụ (Service Communication Protocol) với công nghệ chuẩn là SOAP. SOAP là giao thức nằm giữa tầng vận chuyển và tầng mô tả thông tin về dịch vụ, cho phép người dùng triệu gọi một dịch vụ từ xa thông qua một thông điệp XML. Ngoài ra, để các dịch vụ có tính an toàn, toàn vẹn và bảo mật thông tin, trong kiến trúc dịch vụ Web, người dùng có thêm các tầng Policy, Security, Transaction, Management.

3.1.4. Định dạng dữ liệu JSON

a) Khái niệm JSON

- JSON là một kiểu định dạng dữ liệu trong đó sử dụng văn bản thuần túy, định dạng JSON sử dụng các cặp key - value để dữ liệu sử dụng.

b) Giống nhau và khác nhau giữa JSON và XML

Giống nhau:

- Đều là văn bản tron (không có định dạng (màu sắc, cỡ chữ,...)).
- Đều là “tự mô tả” (người dùng có thể hiểu được).
- Đều là phân cấp (có cấu trúc cây).
- Đều có thể được phân tích cú pháp (parse) bởi JavaScript.
- Dữ liệu JSON và XML có thể được truyền đi bằng AJAX.

Khác nhau:

- JSON không có thẻ kết thúc.
- JSON ngắn hơn.
- JSON nhanh hơn để đọc và ghi.
- JSON có thể được phân tích cú pháp bằng hàm dựng sẵn trong JavaScript là eval().
- JSON sử dụng mảng (Array).
- JSON không dùng các từ reserve.

c) Cú pháp JSON

```
1 {
2   "Employee" :[
3     {
4       "id":"01",
5       "name":"Nguyen Van A",
6       "salary":"1000"
7     },
8     {
9       "id":"02",
10      "name":"Nguyen Van B",
11      "salary":"2000"},
12    {
13      "id":"03",
14      "name":"Nguyen Van C",
15      "salary":"3000"
16    }
17  ]
18 }
```

Hình 3.2: Cú pháp JSON

Các quy luật cú pháp JSON: Cú pháp JSON là một tập hợp con của cú pháp đối tượng JavaScript.

- Dữ liệu là các cặp tên/giá trị (name/values).
- Dữ liệu được phân cách bằng dấu phẩy.
- Đối tượng nằm trong cặp ngoặc nhọn '{ }'.
- Dấu ngoặc vuông giữ mảng '[]'.

3.2. Dịch vụ web và kỹ thuật cơ sở dữ liệu trên internet

Trong thời đại của xã hội công nghệ thông tin và nền kinh tế tri thức, mọi hoạt động của các cá nhân, tổ chức, doanh nghiệp đều mong muốn đạt hiệu quả cao, giành được thắng

lợi trong sự cạnh tranh gay gắt thì đòi hỏi phải có những phương pháp để có thể cung cấp, trao đổi những thông tin, tri thức cần thiết một cách nhanh chính xác, thuận tiện và dễ dàng. Chính vì vậy việc tạo lập xây dựng lưu trữ thông tin, cơ sở dữ liệu chung trên môi trường Internet là một việc vô cùng cần thiết để các cá nhân, tổ chức, doanh nghiệp dễ dàng làm việc, thao tác với cơ sở dữ liệu mọi lúc, mọi nơi, mọi thiết bị khi có Internet thay vì thao tác với cơ sở dữ liệu cục bộ trên máy tính tại một vị trí.

Ưu điểm của việc tạo lập cơ sở dữ liệu trên Internet:

- Cấu trúc phân tán dữ liệu thích hợp cho bản chất phân tán của nhiều người dùng.
- Dữ liệu được chia sẻ trên mạng nhưng vẫn cho phép quản trị dữ liệu địa phương (dữ liệu đặt tại mỗi trạm).
- Dữ liệu có tính tin cậy cao
- Dữ liệu có tính sẵn sàng cao.
- Hiệu năng của hệ thống được nâng cao hơn.
- Cho phép mở rộng các cá nhân, tổ chức, doanh nghiệp một cách linh hoạt.

Nhược điểm:

- Việc thiết kế tạo lập cơ sở dữ liệu phức tạp hơn.
- Chi phí cao hơn.
- Đảm bảo an ninh khó khăn hơn.
- Đảm bảo tính nhất quán dữ liệu khó hơn.

3.2.1. Hosting

Hosting là dịch vụ lưu trữ dữ và chia sẻ liệu trực tuyến, là không gian trên máy chủ có cài đặt các dịch vụ Internet như world wide web (www), truyền file (FTP), Mail... , ta có thể chứa nội dung trang web hay dữ liệu trên không gian đó.

a) Các loại hosting:

- Shared hosting: là một dịch vụ lưu trữ rất nhiều các trang web trên một máy chủ kết nối Internet. Mỗi trang web có phân vùng riêng của mình. Dịch vụ này là một lựa chọn kinh tế cho nhiều người chia sẻ tổng chi phí bảo trì thuê máy chủ. - Free web hosting: là một dịch vụ lưu trữ miễn phí, thường được quảng cáo hỗ trợ. Free Hosting thường sẽ cung cấp một tên miền phụ (yoursite.example.com) hoặc một thư mục (www.example.com/~ Yourname) hoặc ta có thể sử dụng tên miền của chính mình và cần một vài điều kiện kèm theo. Ngược lại, dịch vụ thu phí thường sẽ cung cấp một tên miền cấp thứ hai cùng với các máy chủ (www.tenmien.com).

Nhiều máy chủ miễn phí không cho phép sử dụng tên miền riêng. Hosting miễn phí bị một số nước chặn không cho sử dụng như Trung Quốc.

- Reseller hosting: là một hình thức lưu trữ của máy chủ web mà chủ sở hữu tài khoản có khả năng sử dụng tài khoản của mình để phân bổ lại ổ cứng lưu trữ và băng thông để lưu trữ các trang web thay mặt cho bên thứ ba. Các đại lý mua một phần không gian trên máy chủ sau đó họ bán cho khách hàng thu lợi nhuận.
- Email hosting: là một dịch vụ thư điện tử đặc biệt khác với các dịch vụ email miễn phí hỗ trợ email hay webmail miễn phí. Doanh nghiệp thường chạy các dịch vụ lưu trữ thư điện tử riêng (Email hosting) theo tên miền của họ để tăng uy tín và chứng thực các thông điệp mà họ gửi đi. Email hosting cho phép tùy chỉnh cấu hình và số lượng lớn các tài khoản.
- File hosting: là dịch vụ lưu trữ tập tin trực tuyến, được thiết kế đặc biệt để lưu trữ các nội dung tĩnh, điển hình là các tập tin lớn mà không phải là các trang web. Thông thường họ cho phép truy cập qua giao thức FTP được tối ưu hóa phục vụ cho nhiều người sử dụng.
- Windows hosting: là một dịch vụ lưu trữ, cho rất nhiều các trang web trên một máy chủ chạy hệ điều hành Windows Sever kết nối Internet. Mỗi trang web có phân vùng riêng của mình thường sử dụng các phần mềm chia hosting như Hosting Controller, Plesk...
- Linux hosting: là một dịch vụ lưu trữ rất nhiều các trang web trên một máy chủ chạy hệ điều hành Linux kết nối Internet. Mỗi trang web có phân vùng riêng của mình thường sử dụng các phần mềm chia hosting như Cpanel, Direct Admin...
- VPS Hosting: một máy chủ riêng ảo (tiếng anh: Virtual Private Server), hay VPS là một phương pháp phân vùng một máy chủ vật lý thành máy tính nhiều máy chủ ảo, mỗi máy chủ đã có khả năng của riêng của mình chạy trên máy tính dành riêng. Mỗi máy chủ ảo riêng của nó có thể chạy full-fledged hệ điều hành, và mỗi máy chủ độc lập có thể được khởi động lại.

b) Các thông số cần biết trong hosting

Hệ điều hành (OS) của máy chủ: hiện tại có hai loại OS thông dụng là Linux và Windows.

- Hosting Linux: là Hosting chuyên hỗ trợ ngôn ngữ lập trình PHP, Joomla, các mã nguồn mở...
- Hosting Windows: Hosting Windows chuyên hỗ trợ về ngôn ngữ lập trình ASP, ASP.Net, HTML vì các Ngôn ngữ này, chạy chuyên trên Hosting Windows, do

vậy khi load Web sẽ hỗ trợ tốt hơn, Hosting Windows có hỗ trợ ngôn ngữ PHP, nhưng chủ yếu là hỗ trợ chính là ASP ...

- Dung lượng: Bộ nhớ lưu trữ cho phép bạn tải file lên host.
- Băng thông: Bandwidth (băng thông) là thông số chỉ dung lượng thông tin tối đa mà website được lưu chuyển qua lại mỗi tháng.
- PHP: Phiên bản php hỗ trợ.
- Max file: Số lượng file tối đa có thể upload lên host.
- RAM: Bộ nhớ đệm.
- Addon domain: Số lượng domain ta có thể trở tới hosting
- Subdomain: Số lượng tên miền phụ có thể tạo ra cho mỗi tên miền.
- Park domain: Số lượng tên miền có thể parking.
- Email accounts: Số lượng email đi kèm với hosting.
- FTP accounts: Số lượng FTP account bạn có thể tạo và dùng nó upload dữ liệu lên hosting.

c) Đăng ký mua và sử dụng hosting.

Tùy vào mục đích sử dụng của mỗi cá nhân, tổ chức và doanh nghiệp có thể chọn các hình thức đăng ký mua, thuê hosting khác nhau tại rất nhiều các nhà cung cấp hosting tại Việt Nam như:

- Mắt Bão – ODS
- PA Việt Nam
- DIGISTAR
- Nhân Hòa
- Viettel
- Tenten
- FPT
- VDC
- Vdata
- Vhost

Hiện nay cũng có một số hosting miễn phí phục vụ cho quá trình học tập nghiên cứu như:

- HOSTINGER.VN - 000WEBHOST.COM
- ATSPACE.COM
- AWARDSPACE.COM
- HOSTBUDDY.COM

d. Các bước đăng ký sử dụng hosting

VD: Đăng ký tại "000webhost.com".

- Đến địa chỉ: " https://members.000webhost.com/signup"
- Điền đầy đủ các thông tin yêu cầu như hình:

The image shows a registration form titled "Order Free Web Hosting" with the following fields and annotations:

- Domain Selection:** Two radio buttons. The second one is selected: "I will choose your free subdomain (recommended)". Below it, a text input field contains "ten-mien-thu-cap" and ".net16.net".
- Name:** A text input field containing "John Doe". A red arrow points to it with the label "họ và tên".
- Email:** A text input field containing "john.doe@example.com". A red arrow points to it with the label "email".
- Password:** A text input field. A red arrow points to it with the label "mật khẩu".
- Repeat password:** A text input field. A red arrow points to it with the label "lập lại mật khẩu".
- Facebook:** A button that says "Like us on Facebook". A red arrow points to it with the label "Nhấp chuột để đồng ý điều khoản sử dụng dịch vụ".
- Terms of Service:** A checkbox labeled "I agree to Terms Of Service". A red arrow points to it with the label "Nhấp chuột vào để xác nhận chống spam".
- reCAPTCHA:** A box containing a checkbox labeled "Tôi không phải là người máy" and the reCAPTCHA logo. A red arrow points to the checkbox.
- Create My Account:** A button at the bottom. A red arrow points to it with the label "Nhấp chuột vào để tạo tài khoản".

Hình 3.3: Đăng kí hosting

- Sau khi điền đầy đủ thông tin, nhấp chuột vào "Create my Account", nếu thành công ta sẽ nhận được trang thông báo như bên dưới:

List of your accounts

Account thietkewebsitemienphi.comlu.com with username a7984047 has been created! (account will be setup on server in about 1 minute).

» Domain	» Status	» Action
thietkewebsitemienph...	Waiting for email confirmation..	--

[Create new account](#)

Hình 3.4: Đăng kí thành công

- Đăng nhập email kiểm tra hộp thư đến để xác nhận đăng ký thành công.

Confirm your registration Inbox x

000webhost.com <no-reply@000webhost.com> 11:13 PM (6 minutes ago) ☆
to me ▾

Dear [REDACTED]

Welcome to the best free hosting provider [000webhost.com](#) !

To confirm your registration please click this link: <https://members.000webhost.com/email-confirm/15015068/7UsRCXhPW283f6ELrGy61qKcettl8eewSKcx4QoY>
If clicking the link does not work, please copy and paste the URL into your browser instead.

Sincerely,
[000webhost.com](#) team

nhập chuột vào liên kết để xác nhận

Hình 3.5: Kiểm tra mail xác nhận đăng kí

Sau khi nhấp chuột vào liên kết, ta sẽ được đưa đến trang tài khoản trên 000webhost.com. Ta có thể thay đổi mật khẩu nếu muốn.

List of your accounts

» Domain	» Status	» Action
thietkewebsitemienph...	Active	Go to CPanel

[Create new account](#)

tên miền thứ cấp đã đăng ký *Bảng điều khiển hosting*

» Customer Details	
Name	chung huynh
Email	cong1.com@gmail.com
Password	Change <i>← đổi mật khẩu</i>
Registration	2015-11-20 16:13
Account Status	Active <i>← trạng thái</i>
Last Login From	116.106.72.106

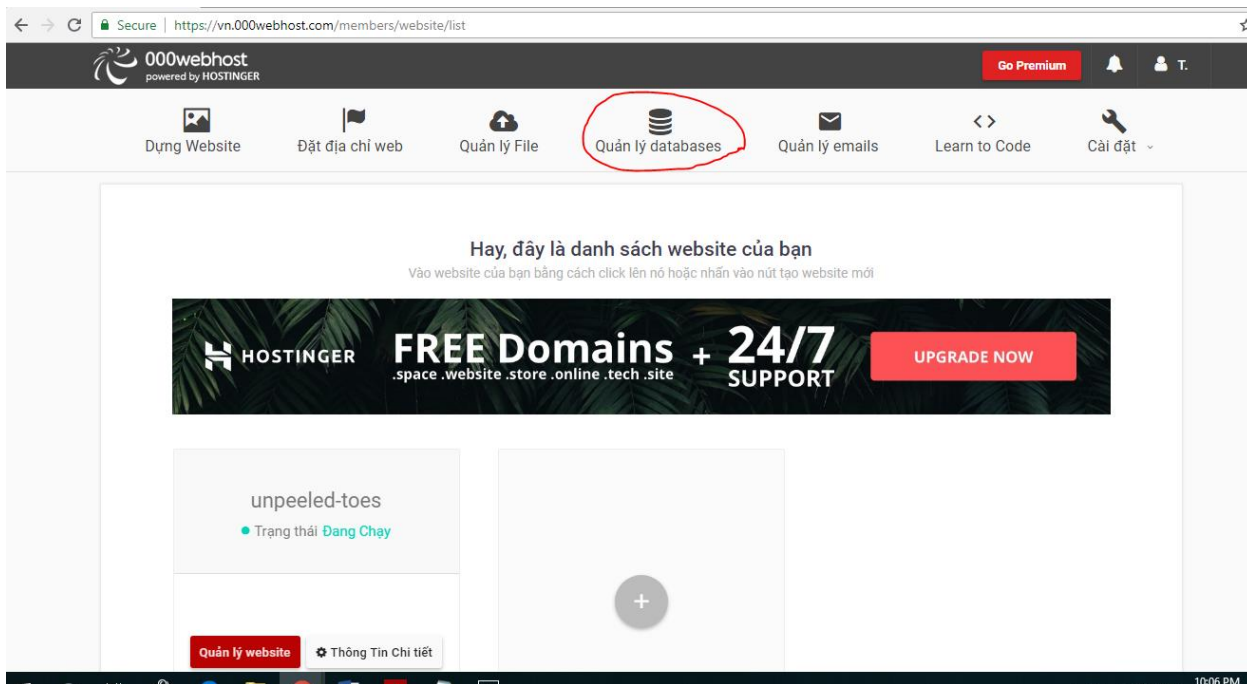
» Premium Hosting	
[REDACTED]	

Hình 3.5: Thông tin tài khoản

e) Tạo cơ sở dữ liệu trên hosting

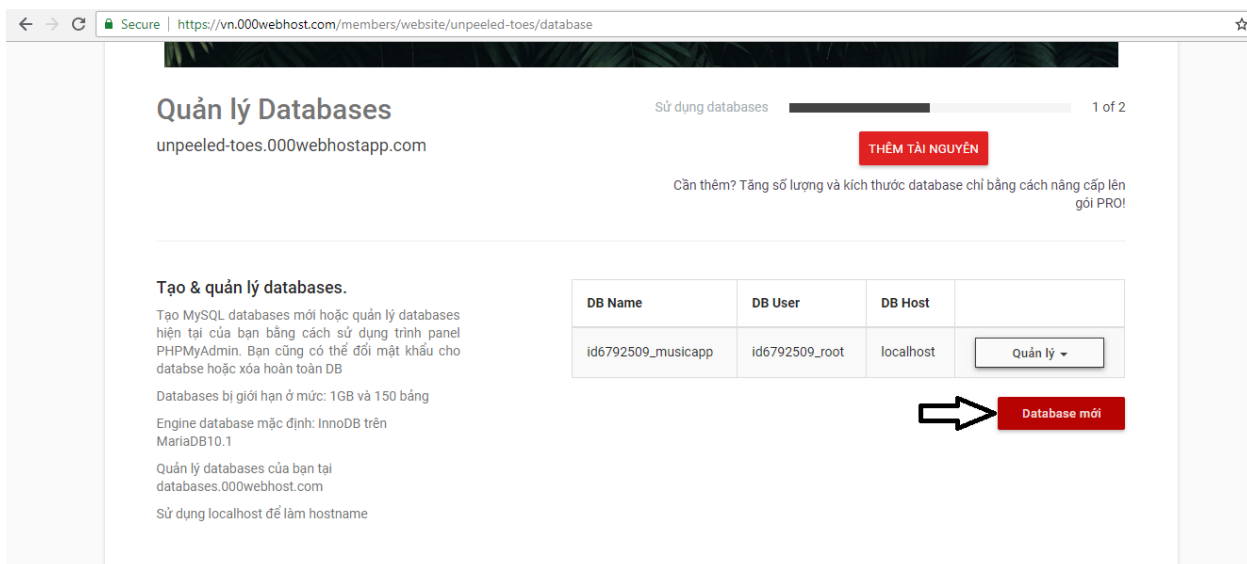
Đăng nhập vào 000webhost.com bằng tài khoản đã đăng ký:

Click vào: Quản lý Database



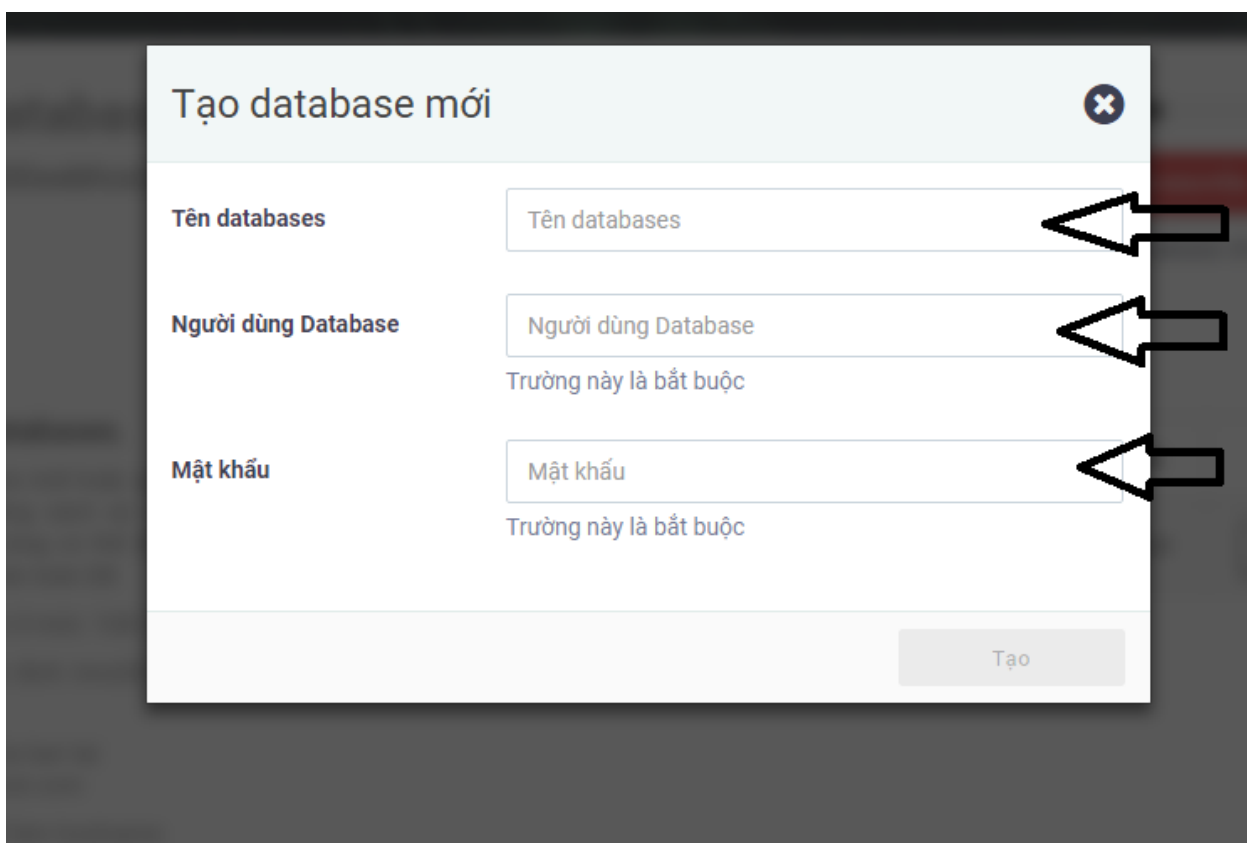
Hình 3.6: Quản lý database

Tiếp theo chọn New database để tạo mới



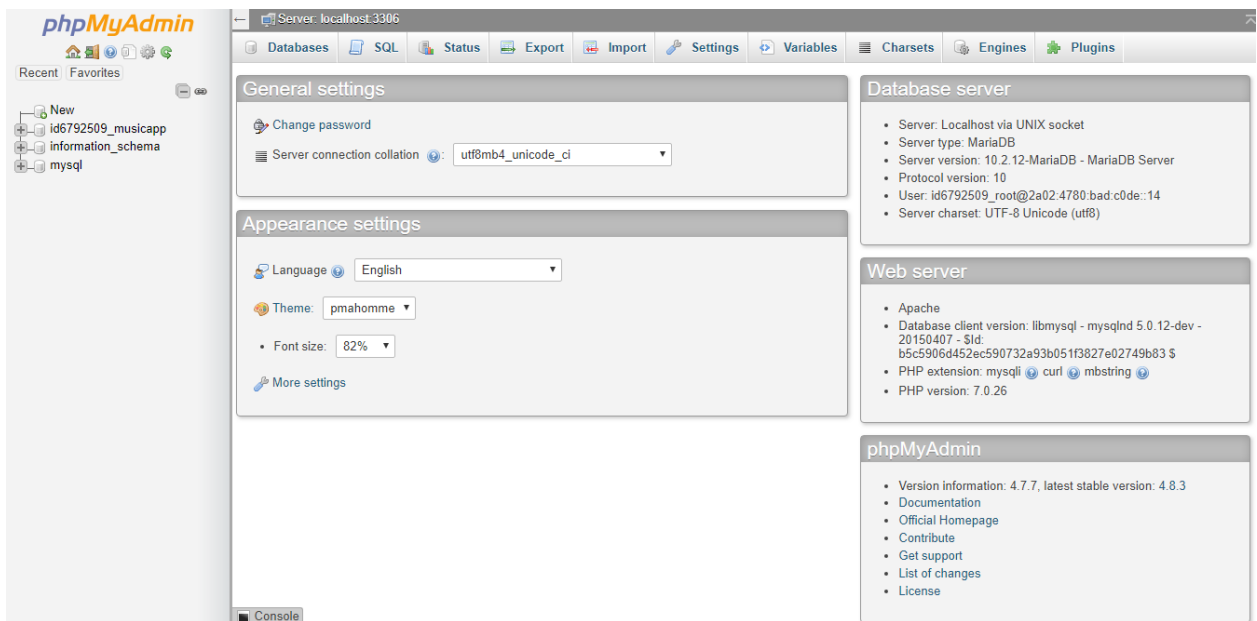
Hình 3.7: Tạo database mới

Nhập thông tin cho database cần tạo vào các ô như trong ảnh:



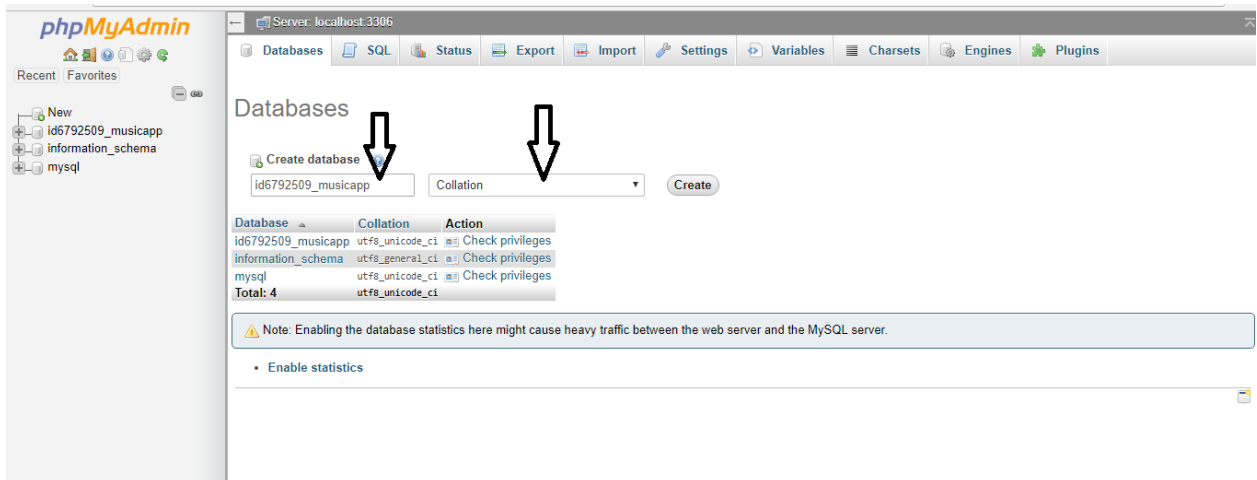
Hình 3.8: Điền thông tin database

Sau khi tạo thành công ta có giao diện mới như sau:



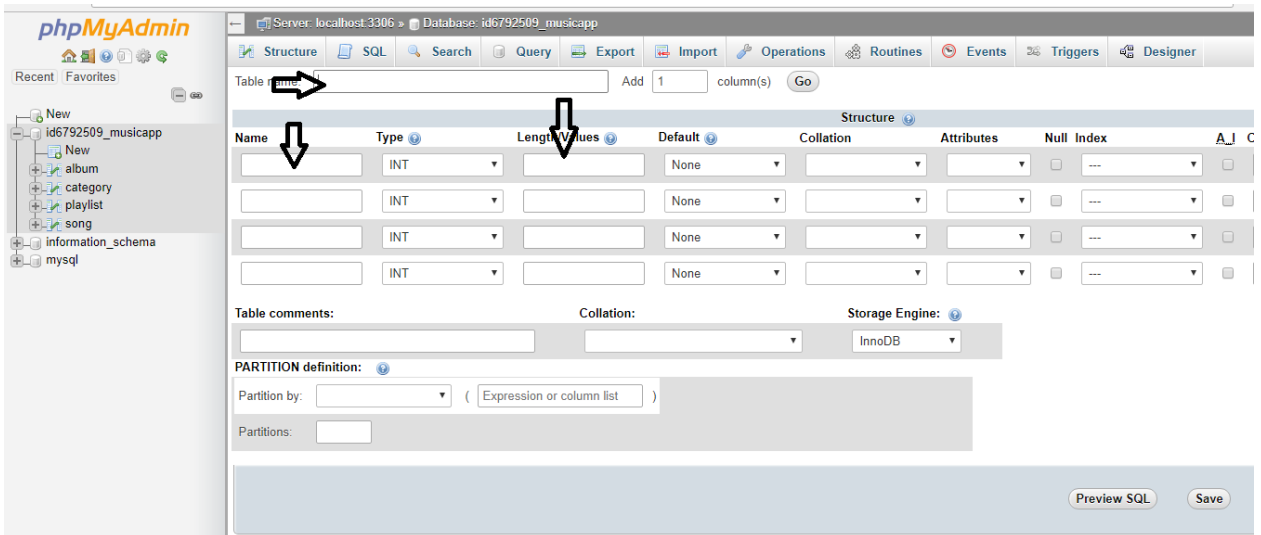
Hình 3.9: Giao diện quản lý database

Click vào New và điền thông tin còn trống sau đó click vào Create để tiến hành tạo database mới như hình:



Hình 3.10: Tạo database mới

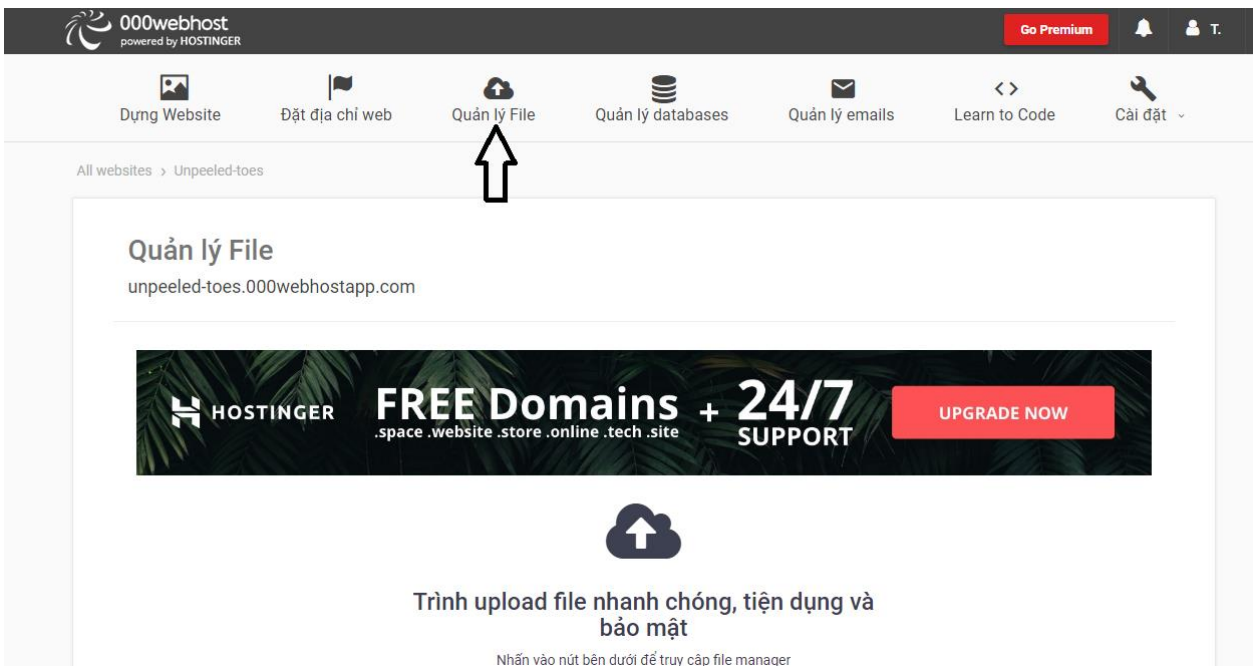
Nhập thông số cho bảng dữ liệu muốn tạo như: tên bảng tên hàng tên cột số hàng số cột, kiểu dữ liệu của hàng cột trong bảng... sau đó ấn Save để hoàn tất quá trình tạo bảng mới database.



Hình 3.11: Điền thông tin database

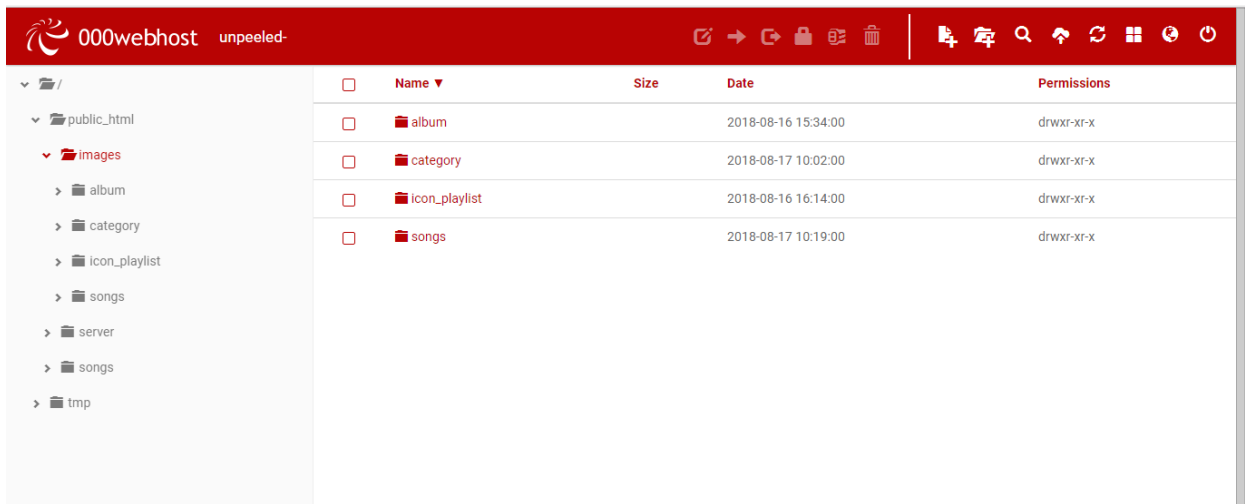
f) Upload cơ sở dữ liệu lên hosting

Click vào Quản lý files:



Hình 3.12: Quản lý file

Tại đây chúng ta sẽ upload file lên:



Hình 3.13: Upload file

3.2.2. Xây dựng webservice

Trên thực tế ứng dụng trên Android muốn trao đổi thông tin được với cơ sở dữ liệu trên Internet thì phải thông qua Webservice. Vì vậy việc xây dựng Webservice là rất quan trọng cho việc viết chương trình ứng dụng Android với cơ sở dữ liệu Internet. Muốn xây dựng được Webservice ta cần nắm được kỹ năng lập trình backend ở đây sử dụng PHP và các kỹ thuật kết nối liên quan đến CSDL.

Ta tiến hành tạo Webservice theo các bước sau:

a) *Xây dựng chức năng kết nối CSDL với Android cho Webservice.*

```
<?php
$host_name = "localhost";
$username = "id6792509_root";
$password = "root123";
$database_name = "id6792509_musicapp";

$conn = mysqli_connect($host_name, $username, $password, $database_name);
mysqli_set_charset($conn, charset: "utf8");
if ($conn) {
} else {
    echo "Not connect to database!";
}
```

Hình 3.14: Xây dựng code kết nối cơ sở dữ liệu

b) Xây dựng chức năng truy vấn CSDL và xem CSDL cho Webservice rồi chuyển sang dạng mã JSON.

```
<?php
require "connect.php";

class ListSong {
    function ListSong($id_song, $id_album, $id_category, $id_playlist, $name_song, $image_song, $singer_song, $link_song)
    {
        $this->id_song = $id_song;
        $this->id_album = $id_album;
        $this->id_category = $id_category;
        $this->id_playlist = $id_playlist;
        $this->name_song = $name_song;
        $this->image_song = $image_song;
        $this->singer_song = $singer_song;
        $this->link_song = $link_song;
    }
}

$arr_list_song = array();
$query = "SELECT * FROM song ";

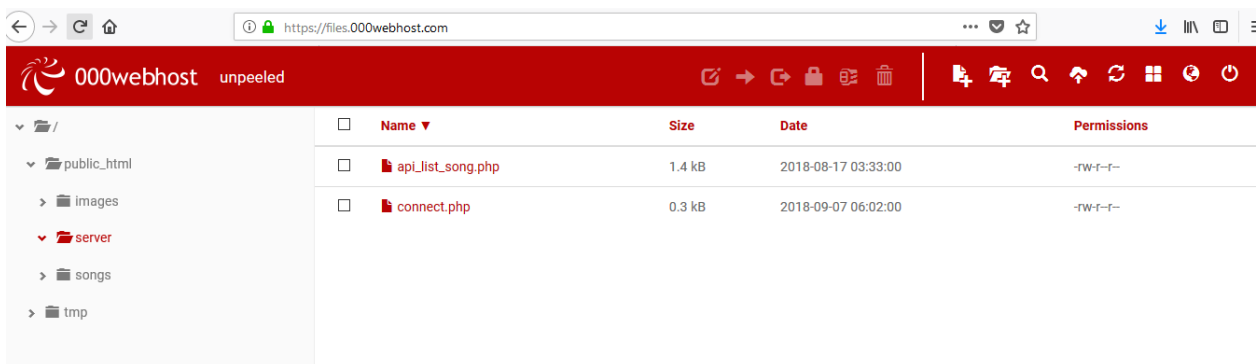
$data_songs = mysqli_query($conn, $query);

while ($row = mysqli_fetch_assoc($data_songs)) {
    array_push($arr_list_song, new ListSong($row['id_song'], $row['id_album'], $row['id_category'], $row['id_playlist'], $row['name_s
}

echo json_encode($arr_list_song);
```

Hình 3.15: Code truy vấn cơ sở dữ liệu

c) Upload Webservice lên hosting sau đó lấy đường dẫn để chạy



Hình 3.16: Upload file lên host

https://unpeeled-toes.000webhostapp.com/server/api_list_song.php

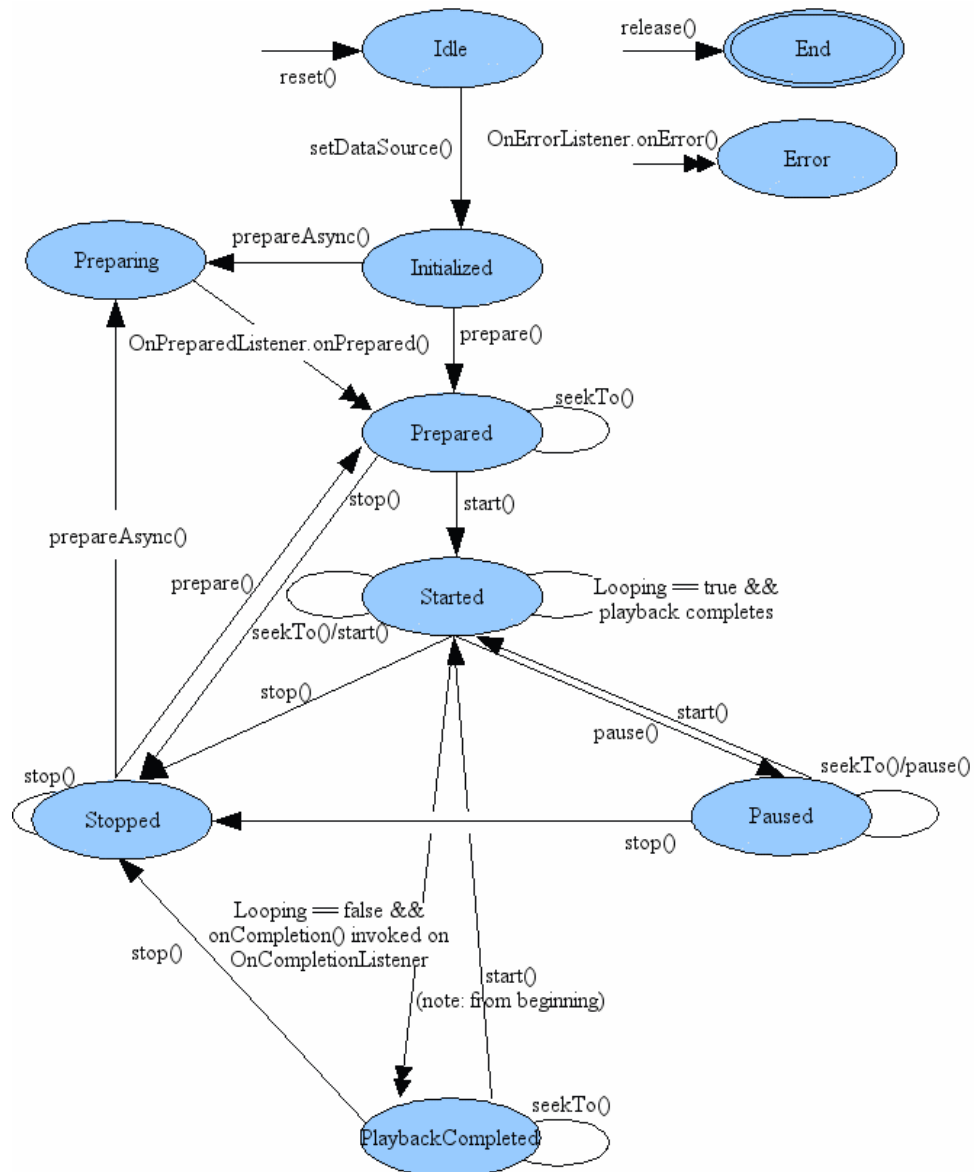
Như vậy ta đã tạo dựng xong một Webservice với các chức năng cơ bản như: kết nối, truy vấn lấy dữ liệu về xem dữ liệu trên CSDL internet.

3.3. MediaPlayer trong Android

Media Player trong Android là một framework hỗ trợ để play những định dạng như video, nhạc, ảnh trong ứng dụng của mình. MediaPlayer có thể hỗ trợ việc play offline hoặc là online tùy chỉnh nhờ vào những API MediaPlayer.[2]

3.3.1. Trạng thái phát nhạc

Dưới đây là trạng thái của MediaPlayer sử dụng để có thể hỗ trợ việc play các media một cách tốt nhất:



Hình 3.17: Trạng thái của MediaPlayer

Chúng ta có thể thấy được các trạng thái sau là các trạng thái mà media có thể ở đó:

- **Idle** : Trạng thái ban đầu của Media Player. Khi muốn trở về trạng thái ban đầu của MediaPlayer thì có thể sử dụng phương thức **reset()**.
- **Initialized**: trạng thái đã khởi tạo được dữ liệu bên trong Media Player bằng phương thức **setDataSource(Uri uri)**. Việc truyền vào Uri này có thể là ở local hoặc là ở server đều có thể play được.
- **End**: Trạng thái kết thúc MediaPlayer khi hàm **release()** được gọi. Đến đây MediaPlayer hoàn toàn bị hủy bỏ khi không sử dụng nữa
- **Error**: Trạng thái lỗi nếu việc play có vấn đề gì đó với Uri hoặc việc tương tác với nhạc trên internet gặp vấn đề. Lắng nghe listener **OnErrorListener** để xử lý khi không thể play nhạc.
- **Prepared**: Trạng thái đã được chuẩn bị để sẵn sàng để có thể play hoặc là trạng thái khi stop hoặc là **seekTo()** đến khoảng nào đó trong toàn bộ độ dài của media. Việc đến được trạng thái này chúng ta phải trải qua hàm đồng bộ **prepareAsync()** hoặc là không đồng bộ **prepare()**. Nếu gọi hàm đồng bộ **prepareAsync()** thì sẽ có 1 listener được trả về nếu mà quá trình chuẩn bị thành công thì sẵn sàng để play. Chỗ này có thể gửi sự kiện chuẩn bị xong ra ngoài giao diện để cập nhật được giao diện phù hợp.
- **Started**: Trạng thái khi mà media đang được chạy, lúc này có thể sử dụng **pause()** để tạm dừng nhạc hoặc **stop()** để dừng hẳn nhạc của media. Gọi **isPlaying()** để biết được media có đang được chạy không.
- **Pause**: Khi media đang chạy, gọi hàm **pause()** để dừng media khi đang phát nhạc. Lúc này trạng thái media đang tạm dừng. Có thể dừng hẳn bằng hàm **stop()** hoặc **start()** để chạy media. Thường dùng cho nút play/pause gọi hàm này để thực hiện việc quản lý media.
- **Stop**: Khi đang ở trạng thái dừng hẳn, có thể chạy lại media bằng **prepareAsync()** hoặc là **prepare()** để có thể trở về trạng thái Prepared và chuẩn bị để play lại từ đầu.
- **Playback Complete**: Trạng thái mà chạy xong media sẽ nhảy vào đây, có thể lắng nghe sự kiện **OnCompletionListener** của nó để có thể thực hiện chuyển bài hoặc là vẫn play bài đó với các trường hợp của loop media.

Ngoài ra còn hàm **seekTo()** để có thể chuyển đến phần mà bạn chọn khi play media. Hàm này có thể được gọi ở nhiều trạng thái khác nhau nên phải chú ý tới từng trạng thái của Media khi gọi hàm này. Việc cho biết chạy đến đâu của tổng số thời gian được trả về thông qua hàm **getCurrentPosition()**.

3.3.1. Permission

Có một số trường hợp sử dụng permission WAKE_LOCK để giữ màn hình ở trạng thái on, ví dụ khi sử dụng với play video.

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

Nếu ứng dụng có sử dụng đến media trên mạng internet thì cần cấp quyền INTERNET để có thể tải được về và chạy.

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

3.3.2. Callbacks

Những callback trả về cho chúng ta những trạng thái và giá trị của Media Player:

- `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

3.4. Các thư viện sử dụng thêm

- Glide 4.7.1: Thư viện hỗ trợ load ảnh từ url trên Android.
- Gson 2.8.5: Thư viện java cho phép người sử dụng có thể chuyển đổi từ một đối tượng Java sang JSON và cũng có thể chuyển đổi từ một đối tượng JSON sang java.
- Retrofit 2.4.0: là một **HTTP client type-safe** cho Android và Java. Retrofit giúp dễ dàng kết nối đến một dịch vụ REST trên web bằng cách chuyển đổi API thành Java Interface.
- NoInternetDialog: 1.1.3 kiểm tra mạng di động

3.5. Kỹ thuật lập trình trên Android

3.5.1. Kỹ thuật xây dựng tệp khai báo Manifest

Khai báo các Activity trong ứng dụng và các permission liên quan (INTERNET)

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.vtt.musiconline">

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

    <application
        android:hardwareAccelerated="true"
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="Music Online"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme"
        tools:replace="android:theme">
        <activity android:name=".view.ListSongActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".view.PlayActivity" />
    </application>

</manifest>
```

Hình 3.18: Khai báo Activity trong ứng dụng

3.5.2. Sử dụng retrofit để lấy dữ liệu từ API

a) Sử dụng Annotations

Annotations để mô tả yêu cầu HTTP và hỗ trợ tham số các URL và truyền tham số truy vấn

Mỗi phương thức phải có Annotation HTTP cung cấp request method và URL. Ở ứng dụng này, sử dụng annotations có sẵn là: @GET

```
public interface MusicApi {  
    @GET("server/api_list_song.php")  
    Call<List<ListSong>> listSongAll(  
    );  
}
```

Hình 3.19: Annotation trong Android

b) Tạo instance Retrofit và xử lý kết quả trả về:

Khai báo và tạo instance của retrofit với domain API:

```
Retrofit retrofit = new Retrofit.Builder()  
    .baseUrl("https://unpeeled-toes.000webhostapp.com/")  
    .client(client)  
    .addConverterFactory(GsonConverterFactory.create(gson))  
    .build();  
musicApi = retrofit.create(MusicApi.class);
```

Hình 3.20: Khai báo và tạo instance của retrofit

Thực hiện Request đến API và xử lý kết quả trả về, ở đây là xử lý API listSong, trả về List các bài hát:

```
try {
    ApiManager apiManager = ApiManager.getInstant(context);
    Response<List<ListSong>> response = apiManager.listSongAll().execute();
    if(response.isSuccessful()){
        if(callback!=null){
            callback.onSuccess(response.body());
        }
    }else{
        if(callback!=null){
            callback.onFailure(response.code(), response.message());
        }
    }
} catch (IOException e) {
    e.printStackTrace();
}
return null;
}
```

Hình 3.21: Thực hiện Request đến API

3.5.3. Các Activity trong ứng dụng

Ứng dụng gồm 2 activity chính:

a) *ListSongActivity*:

- Load dữ liệu listSong và đổ dữ liệu vào recyclerview:

```

// lấy dữ liệu bài hát từ api
private void getAllSong() {
    runOnUiThread() -> CircleProgressBar.getInstance(ListSongActivity.this).showProgress();
    (new PostListSongAllTask( context: ListSongActivity.this,
        new PostListSongAllTask.Listener() {
            @Override
            public void onSuccess(List<ListSong> listSongs) {
                runOnUiThread() -> {
                    try {
                        runOnUiThread() -> CircleProgressBar.getInstance(ListSongActivity.this).dismissProgress();
                        songList = listSongs;
                        //truyền list bài hát lấy từ api về vào adapter của listview
                        songAdapter = new ListSongAdapter( activity: ListSongActivity.this, songList, listSong -> {
                            //Truyền bài hát sang màn phát nhạc khi click vào bài hát
                            runOnUiThread() -> CircleProgressBar.getInstance(ListSongActivity.this).showProgress();
                            Intent intent = new Intent( packageContext: ListSongActivity.this, PlayActivity.class);
                            intentList.clear();
                            intentList.add(listSong);
                            Gson gson = new Gson();
                            String playSearch = gson.toJson(intentList);
                            intent.putExtra( name: "listSongPlay", playSearch);
                            startActivity(intent);
                        });
                        // gán adapter vào listview để hiển thị các bài hát
                        recyclerViewListSong.setAdapter(songAdapter);
                    }
                }
            }
        });
}

```

Hình 3.22: Lấy dữ liệu listSong

- ListSongActivity này gồm 2 nhiệm vụ:
 - Tìm kiếm bài hát (nếu click vào khung tìm kiếm ở MainActivity)

```

editTextSearch.addTextChangedListener(new TextWatcher() {
    @Override
    public void beforeTextChanged(CharSequence s, int start, int count, int after) {
    }

    @Override
    public void onTextChanged(CharSequence s, int start, int before, int count) {
        filter(s.toString());
    }

    @Override
    public void afterTextChanged(Editable s) {
    }
});

private void filter(String text) {
    filteredList.clear();
    for (ListSong item : songList) {
        if (item.getNameSong().toLowerCase().contains(text.toLowerCase())) {
            filteredList.add(item);
        }
    }
    if(text.equals("")){
        filteredList.clear();
    }
    songAdapter.filterList(filteredList);
}

```

Hình 3.23: Xử lý hàm filter để lọc bài hát

- Hiển thị bài hát
 - Dùng RecyclerView để hiển thị danh sách bài hát

```

songList = listSongs;
songAdapter = new ListSongAdapter( activity: ListSongActivity.this, songList, (listSong) -> {
    runOnUiThread() -> CircleProgressBar.getInstance(ListSongActivity.this).showProgress();
    Intent intent = new Intent( packageContext: ListSongActivity.this, PlayActivity.class);
    intentList.clear();
    intentList.add(listSong);
    Gson gson = new Gson();
    String songPlayList = gson.toJson(intentList);
    intent.putExtra( name: "ListSongPlay", songPlayList);
    startActivity(intent);
});
recyclerViewListSong.setAdapter(songAdapter);

```

Hình 3.24: RecyclerView để hiển thị danh sách bài hát

b) PlayActivity

Nhận dữ liệu bài hát từ ListSongActivity và phát bài hát với MediaPlayer

- Sử dụng Glide để load ảnh bài hát:

```

Glide.with( activity: this)
    .load(songList.get(position).getImageSong())
    .into(imgSong);

```

Hình 3.25: Glide để load ảnh bài hát

- Khởi tạo MediaPlayer với URL bài hát và phát :

```

private void PlaySongs(String linkSong) {
    try {
        // Media Player
        mediaPlayer = new MediaPlayer();
        mediaPlayer.setAudioStreamType(AudioManager.STREAM_MUSIC);
        mediaPlayer.setOnCompletionListener(mediaPlayer -> {
            mediaPlayer.stop();
            mediaPlayer.reset();
        });
        mediaPlayer.setDataSource(linkSong);
        mediaPlayer.prepareAsync();
    } catch (IOException e) {
        e.printStackTrace();
    }
    mediaPlayer.setOnPreparedListener(mediaPlayer1 -> {
        mediaPlayer.start();
        timeSong();
        updateTime();
    });
}

```

Hình 3.26: Khởi tạo MediaPlayer

- Dừng bài hát khi tắt PlayActivity:

```

@Override
protected void onDestroy() {
    super.onDestroy();
    mediaPlayer.stop();
    songList.clear();
}

```

Hình 3.27: Code dừng bài hát

- Xử lý thanh SeekBar chạy theo thời gian bài hát bằng hàm seekTo:

```
positionBar.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.seekTo(seekBar.getProgress());  
    }  
});  
}
```

Hình 3.28: Code xử lý thanh SeekBar

- Xử lý event khi ấn nút phát ngẫu nhiên:

Khi click phát ngẫu nhiên sẽ không thể phát lặp lại bài hát và bài hát tiếp theo sẽ random.

```
btnSuffle.setOnClickListener(view -> {  
    if(!checkrandom){  
        if(repeat){  
            repeat = false;  
            btnRefresh.setBackgroundResource(R.drawable.refresh_gray);  
            btnSuffle.setBackgroundResource(R.drawable.suffle);  
        }  
        btnSuffle.setBackgroundResource(R.drawable.suffle);  
        checkrandom = true;  
    } else {  
        btnSuffle.setBackgroundResource(R.drawable.suffle_gray);  
        checkrandom = false;  
    }  
});
```

Hình 3.29: Code cho bài hát chạy ngẫu nhiên

- Xử lý event khi ấn nút lặp lại:

Khi click lặp lại sẽ không thể phát ngẫu nhiên bài hát và bài hát tiếp theo sẽ lặp lại

```

btnRefresh.setOnClickListener(view -> {
    if(!repeat){
        if(checkrandom){
            checkrandom = false;
            btnRefresh.setBackgroundResource(R.drawable.refresh);
            btnSuffle.setBackgroundResource(R.drawable.suffle_gray);
        }
        btnRefresh.setBackgroundResource(R.drawable.refresh);
        repeat = true;
    } else {
        btnRefresh.setBackgroundResource(R.drawable.refresh_gray);
        repeat = false;
    }
});

```

Hình 3.30: Xử lý event khi ấn nút lặp lại

- Xử lý event khi ấn nút play:

Khi click play MediaPlayer sẽ pause khi bài hát đang phát và start khi bài hát đang tạm dừng

```

btnPlay.setOnClickListener(view -> {
    if(mediaPlayer.isPlaying()){
        mediaPlayer.pause();
        btnPlay.setBackgroundResource(R.drawable.play);
    } else {
        mediaPlayer.start();
        btnPlay.setBackgroundResource(R.drawable.pause);
    }
});

```

Hình 3.31: Xử lý event khi ấn nút play

- Xử lý event khi ấn nút next và back:

Chuyển tới URL bài hát tiếp theo hoặc trước đó, đồng thời Load lại các thông tin bài hát lên view

```

btnNext.setOnClickListener(view -> {
    if(songList.size() > 0){
        if(mediaPlayer.isPlaying() || mediaPlayer != null){
            mediaPlayer.stop();
            mediaPlayer.release();
            mediaPlayer = null;
        }
        if(position < songList.size()){
            btnPlay.setBackgroundResource(R.drawable.pause);
            position++;
            if(repeat){
                if(position == 0){
                    position = songList.size();
                }
                position -- 1;
            }
            if(checkrandom){
                Random random = new Random();
                int iRandom = random.nextInt(songList.size());
                if(iRandom == position){
                    position = iRandom-1;
                }
                position = iRandom;
            }
        }
        if (position > songList.size()-1){
            position = 0;
        }
        Glide.with( activity: this)
            .load(songList.get(position).getImageSong())
            .into(imgSong);
        tvName.setText(songList.get(position).getNameSong());
        tvNameGroup.setText(songList.get(position).getNameSong());
        btnPlay.setBackgroundResource(R.drawable.pause);
        PlaySongs(songList.get(0).getLinkSong());
        updateTime();
    }
}

btnPre.setClickable(false);
btnNext.setClickable(false);
Handler handler = new Handler();
handler.postDelayed(() -> {
    btnPre.setClickable(true);
    btnNext.setClickable(true);
}, delayMills: 5000);
});

```

Hình 3.32: Xử lý event khi ấn nút next và back

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 nó thường chỉ được phát được các file bài hát lưu trữ cục bộ trên điện thoại. Việc lưu trữ kho nhạc cục bộ có thể dẫn đến: tốn dụng lượng bộ nhớ, các bài hát nhanh chóng trở nên cũ và không muốn nghe... Việc xây dựng ứng dụng nghe nhạc online giúp người dùng có thể nghe thoải mái các bài hát mới liên tục và không phải băn khoăn về không gian lưu trữ.

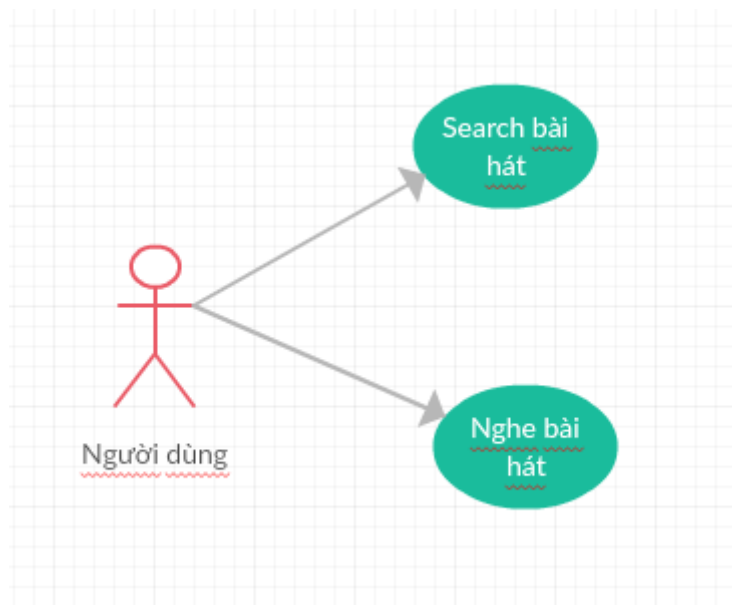
4.1.1 Các chức năng của chương trình

Chương trình xây dựng sẽ đảm bảo các chức năng chính sau:

- Hiển thị các bài hát, ca sĩ.
- Tìm kiếm bài hát
- Phát bài hát theo bài hát tìm kiếm

4.2 Sơ đồ Use-Case của chương trình

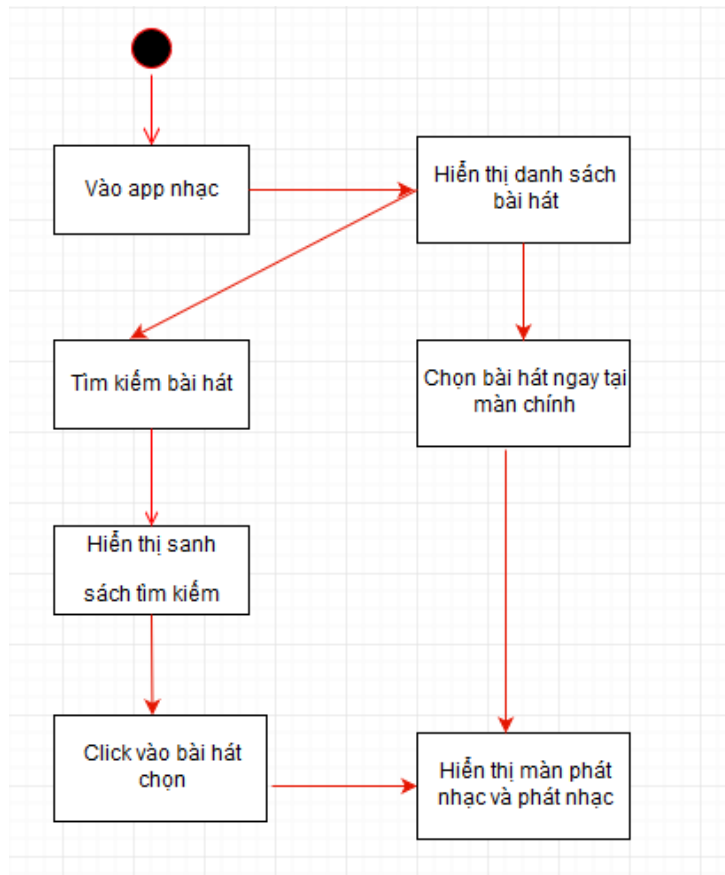
- Sơ đồ Use-Case tổng quát



Hình 4.1: Use-Case tổng quát

Mô tả: người dùng truy cập app có thể search bài hát theo ý thích của mình và nghe bài hát.

- Sơ đồ hoạt động của app



Hình 4.2 Sơ đồ hoạt động của app

4.3 Cơ sở dữ liệu

Bảng bài hát

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	id_song	int(11)		No	None		AUTO_INCREMENT	Change Drop Primary Unique Index More
<input type="checkbox"/>	2	id_album	varchar(255)	utf8_unicode_ci	No	None			Change Drop Primary Unique Index More
<input type="checkbox"/>	3	id_category	varchar(255)	utf8_unicode_ci	No	None			Change Drop Primary Unique Index More
<input type="checkbox"/>	4	id_playlist	varchar(255)	utf8_unicode_ci	No	None			Change Drop Primary Unique Index More
<input type="checkbox"/>	5	name_song	varchar(255)	utf8_unicode_ci	No	None			Change Drop Primary Unique Index More
<input type="checkbox"/>	6	image_song	text	utf8_unicode_ci	No	None			Change Drop Primary Unique Index More
<input type="checkbox"/>	7	singer_song	varchar(255)	utf8_unicode_ci	No	None			Change Drop Primary Unique Index More
<input type="checkbox"/>	8	link_song	text	utf8_unicode_ci	No	None			Change Drop Primary Unique Index More

Hình 4.3: Bảng bài hát

Showing rows 0 - 16 (17 total. Query took 0.0092 seconds.)

SELECT * FROM "song"

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

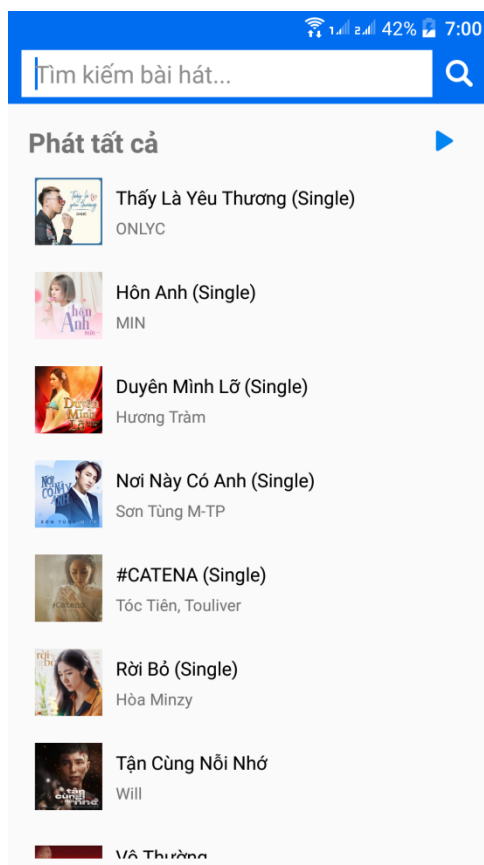
	id_song	id_album	id_category	id_playlist	name_song	image_song	singer_song	link_song
<input type="checkbox"/> Edit Copy Delete	1	1	1	2	Thầy Là Yêu Thương (Single)	https://unpeeled-toes.000webhostapp.com/images/alb...	ONLYC	https://unpeeled-toes.000webhostapp.co
<input type="checkbox"/> Edit Copy Delete	2	2	2	2	Hồn Anh (Single)	https://unpeeled-toes.000webhostapp.com/images/alb...	MIN	https://unpeeled-toes.000webhostapp.co
<input type="checkbox"/> Edit Copy Delete	3	3	1,3	1,2	Duyên Minh Lữ (Single)	https://unpeeled-toes.000webhostapp.com/images/alb...	Hương Trâm	https://unpeeled-toes.000webhostapp.co
<input type="checkbox"/> Edit Copy Delete	4	4	1	1	Nơi Đây Có Anh (Single)	https://unpeeled-toes.000webhostapp.com/images/alb...	Sơn Tùng M-TP	https://unpeeled-toes.000webhostapp.co
<input type="checkbox"/> Edit Copy Delete	5	5	1	1	#CATENA (Single)	https://unpeeled-toes.000webhostapp.com/images/alb...	Tóc Tiên, Touliver	https://unpeeled-toes.000webhostapp.co
<input type="checkbox"/> Edit Copy Delete	6	6	1	1	Rời Bỏ (Single)	https://unpeeled-toes.000webhostapp.com/images/alb...	Hòa Minzy	https://unpeeled-toes.000webhostapp.co
<input type="checkbox"/> Edit Copy Delete	7		3	2	Tân Cùng Nhớ Nhớ	https://unpeeled-toes.000webhostapp.com/images/son...	Will	https://unpeeled-toes.000webhostapp.co
<input type="checkbox"/> Edit Copy Delete	8		4,1	2,1	Vô Thường	https://unpeeled-toes.000webhostapp.com/images/son...	Karik, Orange	https://unpeeled-toes.000webhostapp.co
<input type="checkbox"/> Edit Copy Delete	9		3	3,4	Đẹp Nhất Là Em	https://unpeeled-toes.000webhostapp.com/images/son...	Soobin Hoàng Sơn, Jiyeon	https://unpeeled-toes.000webhostapp.co
<input type="checkbox"/> Edit Copy Delete	10		3	5	We Wish You A Merry Christmas	https://unpeeled-toes.000webhostapp.com/images/son...		https://unpeeled-toes.000webhostapp.co W...

Hình 4.4: Bảng insert dữ liệu bài hát

4.4. Giao diện của chương trình

Chương trình hiển thị danh sách bài hát từ webservice, cho phép tìm kiếm bài hát yêu thích và nghe bài hát người dùng chọn.

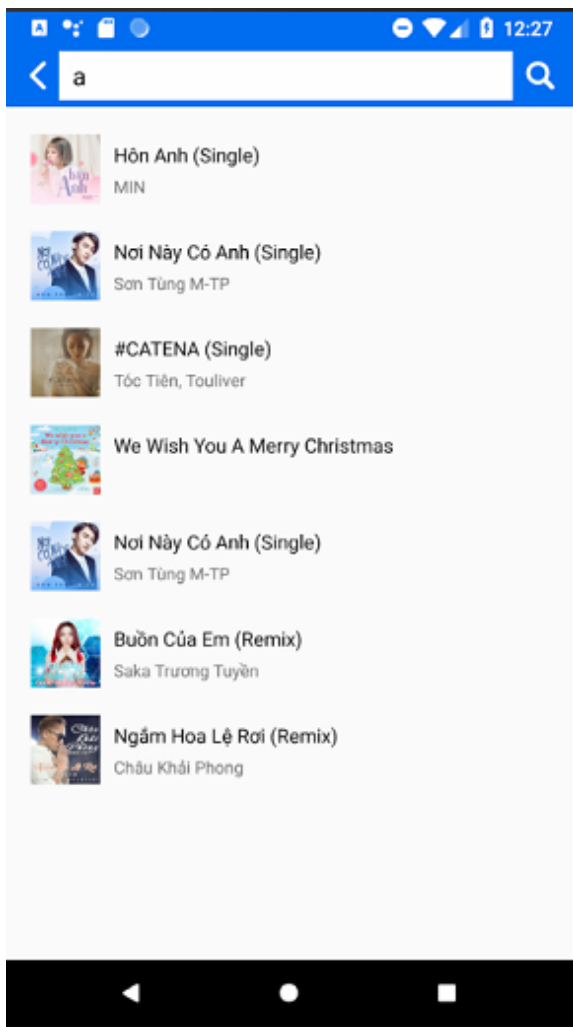
- Giao diện Màn hình chính chương trình phát nhạc: lựa chọn bài hát tùy thích để phát nhạc



Hình 4.5 : Giao diện màn hình chính

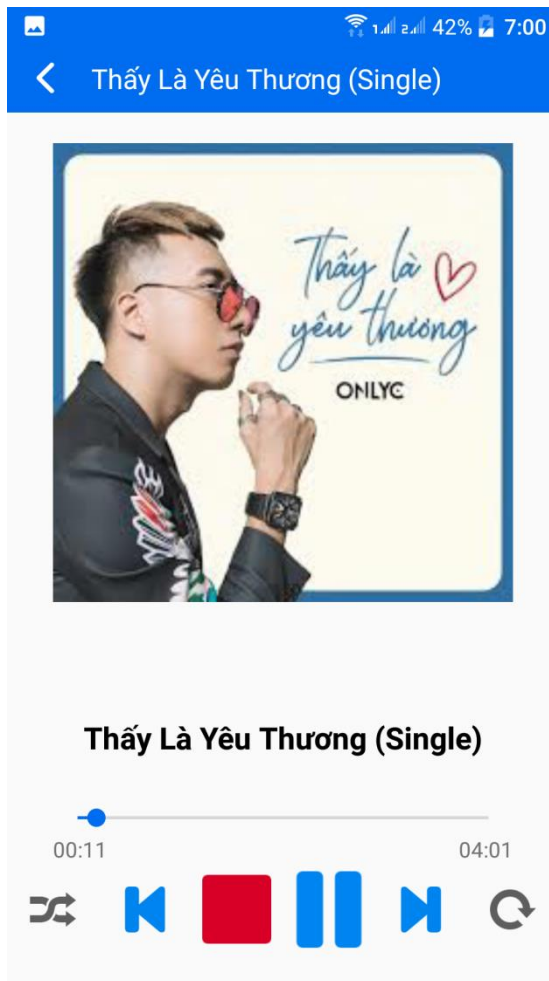
- Giao diện tìm kiếm bài hát: Người dùng gõ vào ô tìm kiếm ứng dụng sẽ hiển thị ra danh sách bài hát gần đúng với từ khóa người dùng gõ vào

-



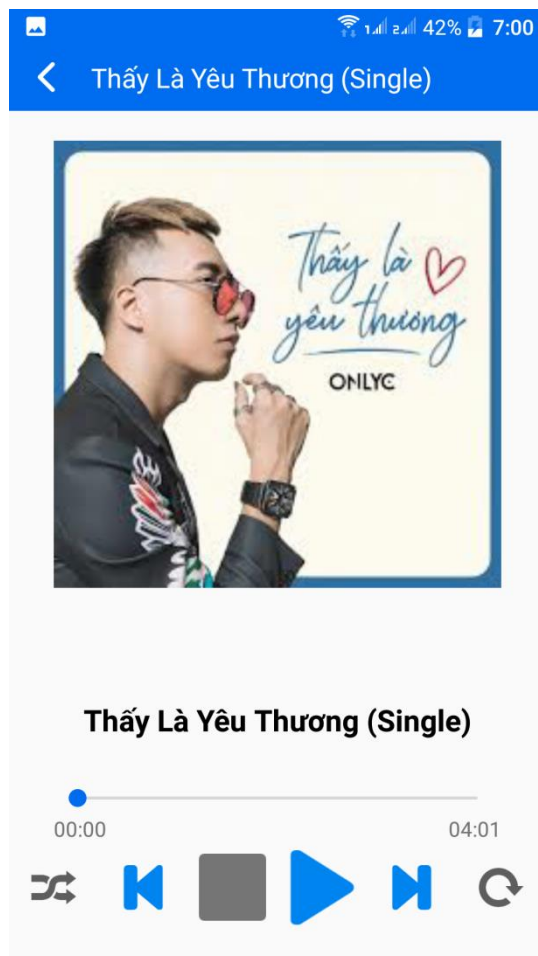
Hình 4.5: Màn hình tìm kiếm bài hát

- Khi chọn vào bài hát sẽ hiển thị ra màn hình phát nhạc



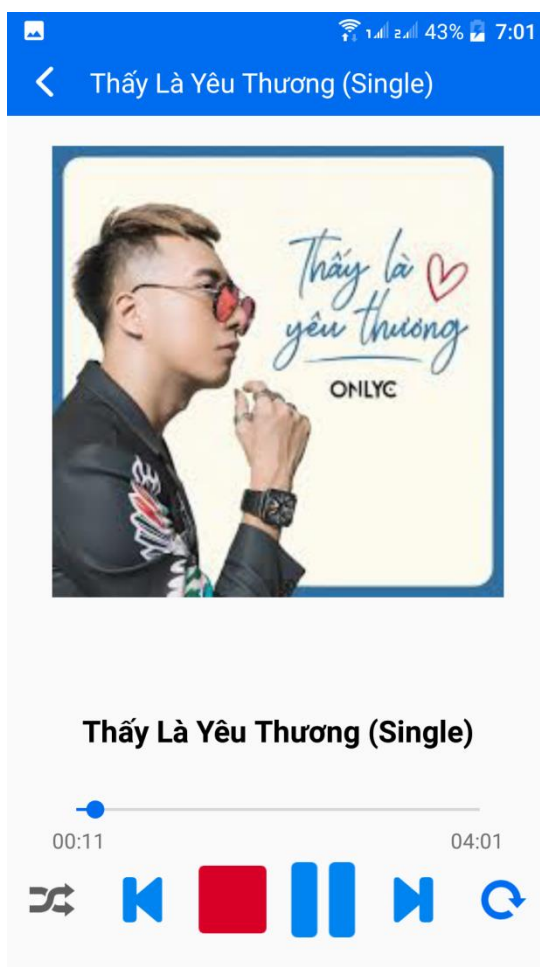
Hình 4.6: Màn hình phát nhạc

- Chức năng dừng phát: Khi chọn chức năng dừng phát màu đỏ bài hát sẽ dừng phát



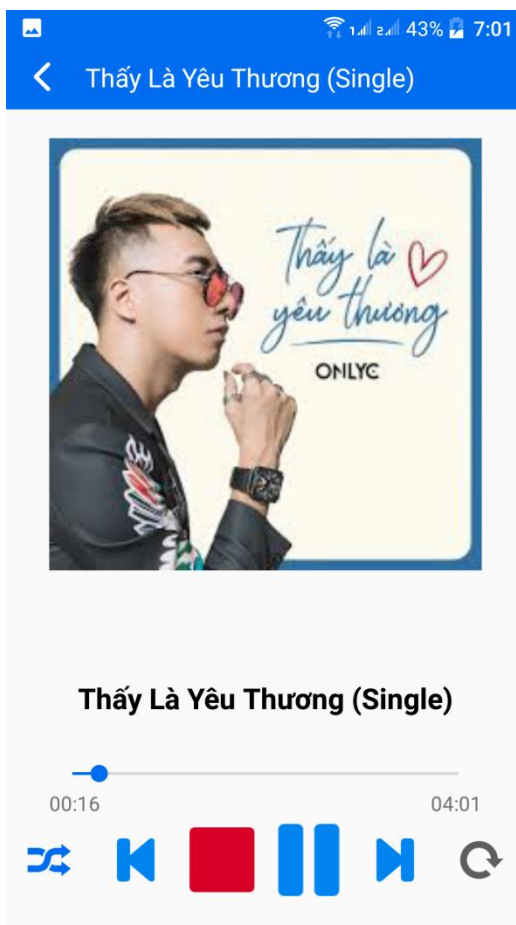
Hình 4.8: Màn hình khi dừng phát nhạc

- Chức năng lặp lại một bài hát đang phát: Chọn vào nút Lặp lại bên phải



- Hình 4.9: Giao diện lặp lại bài hát đang phát

- Chức năng Phát ngẫu nhiên một bài hát: Chọn vào nút phát ngẫu nhiên bên trái



- 4.9.1: Chức năng phát ngẫu nhiên

KẾT LUẬN

Sau một thời gian tìm hiểu đề tài “Xây dựng ứng dụng Android nghe nhạc trên internet” em đã thực hiện được nội dung đề tài của theo yêu cầu đặt ra. Với mục đích là tìm hiểu về hệ điều hành Android, hiểu biết về ngôn ngữ lập trình mới cho điện thoại di động Android và tìm hiểu các ứng dụng của hệ điều hành Android trong thế giới di động ngày nay. Thông qua việc tìm hiểu và nắm bắt lý thuyết về hệ điều hành Android, bước đầu đã xây dựng thành công chương trình thực nghiệm với các chức năng cơ bản.

Trong quá trình tìm hiểu lý thuyết về hệ điều hành Android và xây dựng ứng dụng nghe nhạc online em đã thu được kết quả sau:

- Hiểu và sử dụng được các lớp đối tượng, phương thức hỗ trợ lập trình trên nền hệ điều hành Android.
- Hiểu được định dạng dữ liệu và cơ chế làm việc với JSON.
- Hiểu được các thành phần của Webservice.
- Xây dựng thành công ứng dụng Demo với các chức năng nghe nhạc trực tuyến, tìm kiếm bài hát và hiển thị bài hát theo các mục.
- Đóng gói ứng dụng thành file .apk cho phép cài đặt trên điện thoại Android.

Do thời gian có hạn cùng với kiến thức của bản thân còn hạn chế nên em chưa tìm hiểu sâu về các kiến thức liên quan tới hệ điều hành Android nên chương trình vẫn còn rất nhiều hạn chế, em sẽ cố gắng phát triển thêm để chương trình thân thiện, dễ sử dụng và nhiều chức năng hơn.

TÀI LIỆU THAM KHẢO

- [1]. [https://vi.wikipedia.org/wiki/Android_\(hệ_điều_hành\)](https://vi.wikipedia.org/wiki/Android_(hệ_điều_hành))
 - [2]. <http://laptrinhandroid.vn/media-player-trong-android>
 - [3]. <http://khoapham.vn/KhoaPhamTraining/android/snippet/>
 - [4]. <https://techtalk.vn/kien-truc-cua-he-dieu-hanh-android.html>
 - [5]. <https://o7planning.org/vi/10405/huong-dan-cai-dat-va-cau-hinh-android-studio>
 - [6]. <https://vovanhai.wordpress.com/web-services/giới-thiệu-về-web-service/>
- Tác giả: *Võ Văn Hải*