Contents

[MỞ ĐẦU 3](#_Toc449271444)

[1. Lý do chọn đề tài. 3](#_Toc449271445)

[2. Mục đích nghiên cứu. 3](#_Toc449271446)

[3. Bố cục đề tài. 3](#_Toc449271447)

[5. Phương pháp. 3](#_Toc449271448)

[CHƯƠNG I: TÌM HIỂU LẬP TRÌNH ANDROID VÀ LẬP TRÌNH ĐA PHƯƠNG TIỆN MEDIA 4](#_Toc449271449)

[1. Giới thiệu về lập trình android 4](#_Toc449271450)

[1.1 Tổng quan về hệ điều hành Android 4](#_Toc449271451)

[1.2 Kiến trúc của hệ điều hành Android 5](#_Toc449271452)

[2. Các thành phần cơ bản của một ứng dụng 7](#_Toc449271453)

[2.1. Activity 7](#_Toc449271454)

[2.2. Services 9](#_Toc449271455)

[2.3. Broadcast Receivers 9](#_Toc449271456)

[2.4. Content Providers 10](#_Toc449271457)

[2.5. Một số thành phần khác. 10](#_Toc449271458)

[3. Các thành phần cơ bản trong Android 10](#_Toc449271459)

[3.1 Cấu trúc một Project Android 10](#_Toc449271460)

[3.2 Một số file quan trọng trong project Android 12](#_Toc449271461)

[4. Trình bày tổng quan về audio player 15](#_Toc449271462)

[4.1 Kiến trúc audio 15](#_Toc449271463)

[4.2 Các thư viện và API có thể sử dụng trong android.media 16](#_Toc449271464)

[5. Trình bày tổng quan về video player 19](#_Toc449271465)

[5.1 Kiến trúc 19](#_Toc449271466)

[5.2 Các thư viện và API có thể sử dụng trong android.media 21](#_Toc449271467)

[CHƯƠNG II. PHÂN TÍCH THIẾT KẾ 23](#_Toc449271468)

[1. Khảo sát mô tả bài toán 23](#_Toc449271469)

[1.1. Khảo sát 23](#_Toc449271470)

[1.2. Mục đích 23](#_Toc449271471)

[1.3. Đặc tả bài toán 23](#_Toc449271472)

[2. Phân tích hệ thống 25](#_Toc449271473)

[2.1. Mô hình hóa môi trường làm việc 25](#_Toc449271474)

[2.2. Mô hình hóa chức năng hệ thống 25](#_Toc449271475)

[3. Mô hình hóa lĩnh vực bài toán 26](#_Toc449271476)

[2.3. Phân tích các ca sử dụng 29](#_Toc449271477)

[3. Thiết kế 37](#_Toc449271478)

[3.1. Thiết kế đối tượng 37](#_Toc449271479)

[3.2. Thiết kế giao diện 40](#_Toc449271480)

[CHƯƠNG III. XÂY DỰNG ỨNG DỤNG 45](#_Toc449271481)

[1. Xây dựng giao diện 45](#_Toc449271482)

[1.1 Xây dựng giao diện cho audio 45](#_Toc449271483)

[1.2 Xây dựng một số giao diện chính cho video 49](#_Toc449271484)

[2. Các api sử dụng 56](#_Toc449271485)

[2.1 Các api được sử dụng để chạy audio 56](#_Toc449271486)

[2.2 Các api sử dụng cho chạy video 59](#_Toc449271487)

[KẾT LUẬN 65](#_Toc449271488)

[Gach dau dong liet ke cac van de dat duoc 65](#_Toc449271489)

[1. Những vấn đề đã giải quyết. 65](#_Toc449271490)

[2. Mở rộng 65](#_Toc449271491)

[3. Hướng phát triển 65](#_Toc449271492)

# MỞ ĐẦU

## 1. Lý do chọn đề tài.

Trong sự phát triển của xã hôi hiện nay, công nghệ thông tin đóng một vai trò vô cùng quan trọng trong sự phát triển đi lên, cùng với đó là sự ra đời của hang loạt các thiệt bị công nghệ. Chúng giúp thay đổi cuộc sống của chúng ta ngày càng tốt đẹp hơn. Chúng ta phải kể đến như là điện thoại di động, smart TV, máy tính, PC, laptop, máy chiếu…Trong số các thiết bị đó, điện thoại di động đang được nhiều người sử dụng nhất và bán chạy nhất trên toàn thê giới.Mỗi ngày có hang triệu thiết bị được bán ra và chúng thì luôn được cải tiến về mẫu mã và phần bên trong sao cho đáp ứng đa dạng người dung . Có rất nhiều loại điện thoại khác nhau trên thế giới hiện nay, từ các nhà sản xuất khác nhau, sau đó là các hệ điều hành khác nhau. Chúng ta phải kể đến các ông lớn hiện nay như Apple, Google, SamSung, Sony, Microsoft,.. và chúng chạy các hệ điều hành khác nhau như: Android, iOS, Window Phone… Và theo thống kê thì hệ điều hành anroid đang là hệ điều hành chiếm được đa số các thiết bị di động nhất. Do vậy các ứng dụng trên hệ điều hành này luôn được quan tâm của các lập trình viên.

Với mong muốn có cái nhìn xác thực, rõ ràng hơn để làm nên một ứng dụng chạy trên hệ điều hành này, em lựa chọn để tài “***Xây dựng ứng dụng video player trên hệ điều hành android***” làm đề tài đồ án tốt nghiệp của mình. Từ đó, giúp em trong việc có thể xây dựng thêm các phần mềm khác với mong muốn trở thành một lập trình viên giỏi trong lĩnh vực này.

## 2. Mục đích nghiên cứu.

Đề tài này nhằm mục tiêu hoàn thiện kĩ năng lập trình, tìm hiểu công nghệ mới, áp dụng và triển khai kĩ thuật.

**3. Nội dung nghiên cứu.**

Nội dung nghiên cứu trong đề tài này sẽ chỉ ra cách thức để tạo một trình đọc audio và video. Cách xử lí hoạt động của chúng.

## 3. Bố cục đề tài.

Những nội dung và kết quả tìm hiểu của đề tài được trình bày trong 3 chương như sau:

- Chương I: Tổng quan lí thuyết android và lập trình đa phương tiện media.

- Chương II: Phân tích thiết kế

- Chương III : Xây dựng ứng dụng và kết quả thu được

## 5. Phương pháp.

Phương pháp nghiên cứu lý thuyết:

Tiến hành thu thập và nghiên cứu các tài liệu có liên quan đến đề tài.

Tổng hợp các tài liệu.

Chọn lọc các tài liệu nghiên cứu để báo cáo đề tài đồ án tốt nghiệp.

Phương pháp nghiên cứu thực nghiệm:

Khảo sát thực nghiệm.

Tìm hiểu một vài nguyên lí hệ điều hành

Tìm hiểu về audio và video player trên hệ điều hành android

Ứng dụng kĩ thuật vào triển khai phần mềm.

Kiểm tra và đánh giá kết quả đạt được.

# CHƯƠNG I: TÌM HIỂU LẬP TRÌNH ANDROID VÀ LẬP TRÌNH ĐA PHƯƠNG TIỆN MEDIA

## 1. **Giới thiệu về lập trình android**

### 1.1 Tổng quan về hệ điều hành Android

Android là hệ điều hành mã nguồn mở dựa trên Linux được dùng cho các thiết bị di động như  điện thoại thông minh (smart phone) hoặc máy tính bảng (tablet computer). Android được phát triển bởi *Open Handset Alliance,*dẫn đầu bởi Google và một số công ty khác.

Android cung cấp một phương pháp tiếp cận thống nhất để phát triển ứng dụng cho các thiết bị di động, điều này có nghĩa là các lập trình viên chỉ cần pháp triển ứng dụng cho Android, và các ứng dụng của họ có thể chạy trên nhiều thiết bị Android khác nhau.

Phiên bản thử nghiệm đầu tiên của Android Software Development Kit (SDK) được phát hành vào năm 2007 bởi Google. Trong khi đó phiên bản thương mại đầu tiên Android 1.0 được phát hành vào tháng 9 năm 2008.

Ngày 27-6-2012, tại hội nghị Google I/O. Google đã công bố phiên bản Android tiếp theo, 4.1 Jelly Bean, Jelly Bean là một phiên bản được cải tiến đáng kể, với mục đích chính là cải thiện giao diện người dùng, bao gồm cả chức năng và hiệu suất.

Android cung cấp mã nguồn mở miễn phí, cùng với giấy phép sử dụng phần mềm mã nguồn mở. Google công bố hầu hết các mã nguồn mở theo giấy phép Apache 2.0, nhân Linux, theo giấy phép GNU 2.0.

***Tính năng của Android***

Android là một hệ điều hành mạnh mẽ cạnh tranh với Apple 4GS, với nhiều tính năng tuyệt vời. Một vài tính năng chính như sau:

+ Giao diện đẹp: HĐH Android cung cấp một giao diện người dùng đẹp và trực quan.

+ Kết nối: GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE, NFC and WiMAX.

+ Lưu trữ: SQLite: một cơ sở dữ liệu nhẹ, được sử dụng cho mục đích lưu trữ.

+ Media: H.263, H.264, MPEG-4 SP, AMR, AMR-WB, AAC, HE-AAC, AAC 5.1, MP3, MIDI, Ogg Vorbis, WAV, JPEG, PNG, GIF, and BMP.

+ Tin nhắn: SMS, MMS.

+ Trình duyệt Web: Cơ bản dựa trên mã nguồn mở WebKit, cùng với Chrome’s V8 JavaScript, hỗ trợ HTML5 và CSS3.

+ Cảm ứng đa điểm: Android có hỗ trợ cho cảm ứng đa điểm mà ban đầu đã được tạo sẵn trong các thiết bị cầm tay như HTC Hero.

+ Đa tác vụ: Cùng thời điểm các ứng dụng khác nhau có thể chạy đồng thời.

+ Thay đổi kích thước widgets: widgets có thể thay đổi kích thước vì vậy người dùng có thể mở rộng để hiển thị nhiều nội dung, hoặc thu nhỏ để tiết kiệm không gian.

+ Đa ngôn ngữ

+ GCM: Google Cloud Messaging (GCM) là một dịch vụ cho phép nhà phát triển gửi các thông điệp ngắn đến người dùng của họ trên thiết bị Anrdroid, mà không cần giải pháp đồng bộ độc quyền.

+ Wi-Fi: Công nghệ cho phép phát hiện và kết nối Wifi nhanh, trên một băng thông cao thông qua kết nối Peer-to-Peer.

+ Android Beam: Phổ phiến là NFC, một công nghệ cho phép chia sẻ ngay lập tức, chỉ cần chạm hai điện thoại cùng hỗ trợ NFC.

***Các ứng dụng Android***

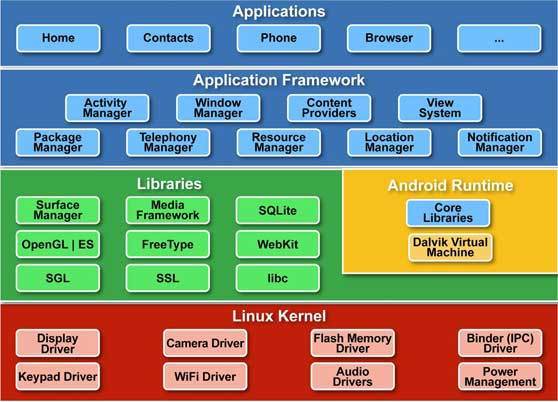
Ứng dụng Android thông thường được phát triển bằng ngôn ngữ Java, sử dụng Android Software Development Kit.

Mỗi lần phát triển, ứng dụng Anrdroid có thể đóng gói dễ dàng và bán ra thông qua Google Play hoặc Amazon Appstore.

Android được dùng cho hàng trăm triệu thiết bị di động tại hơn 190 quốc gia trên toàn thế giới. Mỗi ngày có hơn 1 triệu thiết bị Android mới được kích hoạt.

### 1.2 Kiến trúc của hệ điều hành Android

Hệ điều hành Android là một tập hợp của các thành phần phần mềm được chia thành 5 phần và 4 lớp chính:



* **Linux Kernel**

Linux Kernel là lớp thấp nhất. Nó cung cấp các chức năng cơ bản như quản lý tiến trình, quản lý bộ nhớ, quản lý thiết bị như: Camera, bàn phím, màn hình,… Ngoài ra, nó còn quản lý mạng, driver của các thiết bị, điều này gỡ bỏ sự khó khăn về giao tiếp với các thiết bị ngoại vi.

* **Libraries**

Phía trên Linux Kernel là tập hợp các bộ thư viện mã nguồn mở WebKit, bộ thư viện nổi tiếng libc, cơ sở dữ liệu SQLite hữu ích cho việc lưu trữ và chia sẻ dữ liệu, bộ thư viện thể phát, ghi âm về âm thanh, hoặc video. Thư viện SSL chịu trách nhiệm cho bảo mật Internet.

* **Android Runtime**

Đây là thành phần thứ 3 trong cấu trúc, thuộc về lớp 2 tính từ dưới lên. Phần này cung cấp một thành phần quan trọng gọi là Dalvik Virtual Machine là một máy ảo Java đặt biệt, được thiết kế tối ưu cho Android.

Máy ảo Dalvik sử dụng các tính năng cốt lõi của Linux như quản lý bộ nhớ, đa luồng, mà thực chất là bên trong ngôn ngữ Java. Máy ảo Dalvik cho phép tất cả các ứng dụng Android chạy trong tiến trình riêng của nó.

Android Runtime cũng cung cấp bộ thư viện cốt lõi, cho phép các lập trình viên Android sử dụng để viết các ứng dụng Android.

* **Application Framework**

Lớp Application Framework cung cấp nhiều dịch vụ cấp cao hơn cho các ứng dụng trong các class Java. Các lập trình viên cũng được phép sử dụng các dịch vụ này trong các ứng dụng của họ.

* **Applications**

Bạn sẽ thấy tất cả các ứng dụng Android ở lớp trên cùng. Ứng dụng bạn viết sẽ được cài đặt vào lớp này.

## 2. Các thành phần cơ bản của một ứng dụng

Thành phần ứng dụng là khối kiến trúc cơ bản của ứng dụng Android. Các thành phần này được liên kết lỏng lẻo bởi các tập tin kê khai ứng dụng: **AndroidManifest.xml**, mô tả mỗi thành phần của ứng dụng và cách chúng tương tác.

4 thành phần chính được sử dụng trong ứng dụng Android:

**Activities**: Ra lệnh cho giao diện và xử lý tương tác của người dùng với màn hình smartPhone.

**Services**: Là thành phần chạy phía sau, chạy bên trong của mỗi ứng dụng.

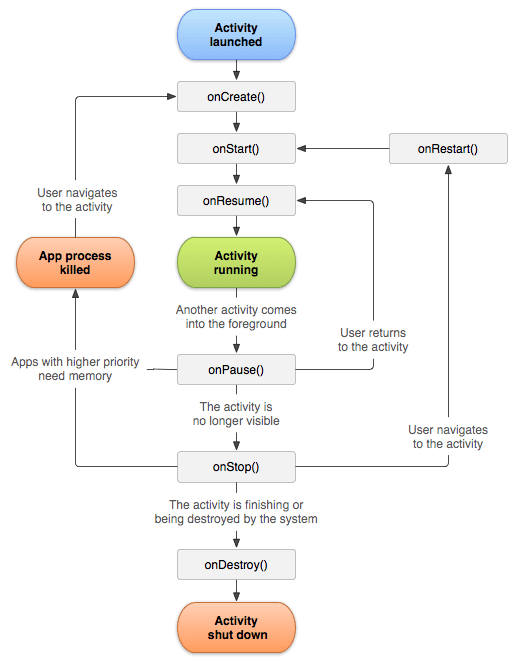
**Broadcast Receivers**: Xử lý việc giao tiếp giữa HĐH Android và ứng dụng.

**Content Providers**: Xử lý dữ liệu và  các vấn đề về quản lý cơ sở dữ liệu.

### 2.1. Activity

Một **Activity** đại diện cho một màn hình với một giao diện người dùng. VD: Một ứng dụng Email sẽ có một **Activity** để hiển thị tất cả các Email, một **Activity** khác để để soạn Email, một**Activity** khác để đọc Email. Nếu một ứng dụng có nhiều hơn 1 Activity thì một trong số chúng sẽ được đánh dấu là **Activity** hiển thị đầu tiên khi ứng dụng khởi chạy.

Nếu bạn đã từng lập trình với ngôn ngữ C, C++, Java thì bạn cũng biết rằng chương trình bắt đầu chạy từ hàm **main()**, theo cách tương tự thì **Android** cũng khởi đầu bởi 1 **activity** được đánh dấu là khởi đầu bằng cách gọi phương thức **onCreate()**. Có một chuỗi các phương thức được gọi để bắt đầu một **activity** và một chuỗi các phương thức khác để hủy 1 **activity**, như hình bên dưới:



**Class Activity** định nghĩa nhiều phương thức cho các sự kiện, bạn không cần phải thực thi tất cả các phương thức này, tuy nhiên việc hiểu rõ và áp dụng chúng vào ứng dụng của bạn khá quan trọng, giúp cho ứng dụng của bạn có các hành vi như mong đợi của người dùng.

* **onCreate()**: Nó là phương thức đầu tiên được gọi dùng để tạo một **activity** vào lần đầu tiên **activity** được gọi.
* **onStart()**: Sẽ được gọi khi nó hiện hữu với người dùng.
* **onResume()**: Sẽ được gọi khi người dùng tương tác với các ứng dụng.
* **onPause()**: Tạm dừng một **activity**, không nhận dữ liệu do người dùng nhập vào và không thế thực thi lệnh nào. Phương thức này được gọi khi **activity** hiện tại đang được tạm dừng, và **activity** trước đó đang được tiếp tục.
* **onStop()**: Được gọi khi một activity đã không được nhìn thấy trong thời gian dài.
* **onDestroy()**: Được gọi trước khi hệ thống hủy **activity**.
* **onRestart()**: Được gọi khi **activity** cần được dùng trở lại sau khi bị gọi **onStop()**;

Một Activity là một Class con của Class **Activities**:

|  |  |
| --- | --- |
|  | public class MainActivity extends Activity {  } |

### 2.2. Services

Một Service là một thành phần được chạy bên trong nền để xử lý các công việc trong thời gian dài. Một ứng dụng nghe nhạc có thể phát nhạc, trong khi đó người dùng đang ở giao diện của ứng dụng khác. Hoặc ứng dụng download có thể tải dữ liệu trên mạng về máy mà không ngăn chặn người dùng tương tác với các ứng dụng khác.

Một Service là một Class con của Class **Service**:

|  |  |
| --- | --- |
|  | public class MyService extends Service {   } |

### 2.3. Broadcast Receivers

Broadcast Receivers chỉ đơn giản là xử lý và phát các thông điệp từ các ứng dụng khác hoặc từ hệ thống. Ví dụ: các ứng dụng có thể bắt đầu phát thông điệp đến các ứng dụng khác để cho biết rằng một số dữ liệu đã được tải thành công xuống thiết bị và sẵn sàng cho việc sử dụng, Broadcast Receivers sẽ đảm nhận việc thông báo vào đưa ra những hành động thích hợp.

Một Broadcast Receiver được thực thi như là một class con của class **BroadcastReceiver** và mỗi thông điệp là một Intent.

|  |  |
| --- | --- |
|  | public class MyReceiver extends BroadcastReceiver {  } |

### 2.4. Content Providers

Một Content Providers chứa các thành phần cung cấp dữ liệu từ một ứng dụng đến một ứng dụng khác theo yêu cầu. Yêu cầu đó được sử lý bằng các phương thức của class ContentResolver. Dữ liệu được lưu trữ trong file hệ thống, trong cơ sở dữ liệu hoặc ở một nơi nào khác.

Một Content Providers được thực thi như là class con của class **ContentProvider** và phải thực thi theo tiêu chuẩn trong bộ các API để cho các ứng dụng khác có thể giao tiếp.

|  |  |
| --- | --- |
|  | public class MyContentProvider extends ContentProvider {  } |

Chúng ta sẽ đi vào chi tiết các thành phần này trong các phần sau.

### 2.5. Một số thành phần khác.

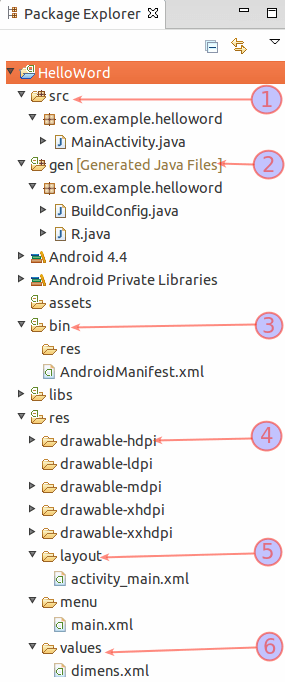
Ngoài các thành phần chính nêu ở trên còn có một số thành phần khác. Các thành phần này sẽ bổ sung, tạo mối liên hệ logic giữa chúng.

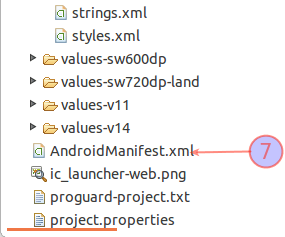
* **Fragments**: Đại diện cho một hành vi hoặc một phần của giao diện người dùng trong một Activity.
* **Views**: Thành phần giao diện được vẽ trên màn hình bao gồm các button, textView, list,…
* **Layouts**: Phân chia, định dạng màn hình, và sự thể hiện các Views.
* **Intents**: Đóng gói các thông điệp.
* **Resources**: Các phần tử bên ngoài: chuỗi, hằng số, hình ảnh, …
* **Manifest**: Tập tin cấu hình của các ứng dụng.

## 3. Các thành phần cơ bản trong Android

### 3.1 Cấu trúc một Project Android

Trước khi chạy ứng dụng, nên dành vài phút để tìm hiểu về các thư mục và các File bên trong HelloWord Project.





* **(1) src**: Nơi chứa các file .java trong Project. Mặc định, nó sẽ chứ file MainActivity.java
* **(2) gen**: Nơi chứa các file .R trong Project. Một tập tin được trình biên dịch tạo ra để tham chiếu đến các tài nguyên trong Project, chúng ta không nên thay đổi file này.
* **(3) bin**: Nơi chứa file cài đặt .apk.
* **(4) res/drawable-hdpi**: Đây là thư mục chứa các đối tương đồ họa.
* **(5) res/layout**: Đây là thư mục chức các tập tin để xác định giao diện người dùng.
* **(6) values**: Đây là thư mục chứa các file .xml khác nhau. Nó bao gồm tập hợp các tài nguyên như: string, colors, …
* **(7) AndroidManifest.xml**: Đây là tập tin khai báo, dùng để mô tả các đặc trưng cơ bản của ứng dụng, và xác định mỗi thành phần của nó.

### 3.2 Một số file quan trọng trong project Android

#### **3.2.1 MainActivity.java**

Đây là code mặc định của file MainActivity.java

|  |  |
| --- | --- |
|  | package com.example.helloword;  import android.os.Bundle;  import android.app.Activity;  import android.view.Menu;  public class MainActivity extends Activity {      @Override      protected void onCreate(Bundle savedInstanceState) {          super.onCreate(savedInstanceState);          setContentView(R.layout.activity\_main);      }      @Override      public boolean onCreateOptionsMenu(Menu menu) {          // Inflate the menu; this adds items to the action bar if it is present.          getMenuInflater().inflate(R.menu.main, menu);          return true;      }  } |

**R.layout.activity\_main**: trỏ đến file **activity\_main.xml** được đặt trong thư mục **res/layout**.

Phương thức **onCreate()** là một trong số các phương thức để nạp một activity.

#### **3.2.2 AndroidManifest.xml**

Bất cứ thành phần nào được triển khai như một phần trong ứng dụng, phải được khai báo trong tập tin:  AndroidManifest.xml. File này giống như một interface giữa HĐH Android và ứng dụng của bạn. Vì vậy nếu thành phần nào bạn dùng mà không khai báo trong file này, thì HĐH sẽ không xem xét đến chúng.

Đây là code mặc định của AndroidManifest.xml:

|  |  |
| --- | --- |
|  | <?xml version="1.0" encoding="utf-8"?>  <manifest xmlns:android="<http://schemas.android.com/apk/res/android>"      package="com.example.helloword"      android:versionCode="1"      android:versionName="1.0" >      <uses-sdk          android:minSdkVersion="8"          android:targetSdkVersion="18" />        <application          android:allowBackup="true"          android:icon="@drawable/ic\_launcher"          android:label="@string/app\_name"          android:theme="@style/AppTheme" >          <activity              android:name="com.example.helloword.MainActivity"              android:label="@string/app\_name" >              <intent-filter>                  <action android:name="android.intent.action.MAIN" />                  <category android:name="android.intent.category.LAUNCHER" />              </intent-filter>          </activity>      </application>  </manifest> |

**<application>…</application>**: Thẻ bao quanh các thành phần có liên quan trong ứng dụng. Thuộc tính **Android:icon**: Sẽ trỏ đến các icon có sẵn trong thư mục**res/drawable-hdpi**. Ứng dụng này sẽ sử dụng hình **ic\_launcher** là icon.

**<activity>…</activity>**: Thẻ này được sử dụng để nêu chi tiết về một activity. Thuộc tính **android:name** được sử dụng để mô tả chi tiết về tên class của một activity.  Thuộc tính**android:label** xác định một chuỗi ký tự được sử dụng là nhãn cho activity đó. Nếu trong ứng dụng của bạn có nhiều Activities thì bạn sẽ phải tạo nhiều thẻ **<activity>**.

**<action />**: với name **android.intent.action.MAIN** để cho Android biết rằng activity này sẽ được load lên đầu tiên khi chạy ứng dụng.

**<category />**: với name**android.intent.category.LAUNCHER** để cho Android biết ứng dụng có thể được chạy từ một Icon trên thiết bị.

**@string**: sẽ được chỉ định đến file **strings.xml**, **@string/app\_name** sẽ được dẫn đến chuỗi app\_name nó là **HelloWord**.

Sau đây là một số thẻ khác bạn có thể sử dụng trong file **AndroidManifest.xml**.

**<activity>** là thẻ dành cho các activities.

**<service>** là thẻ dành cho các services.

**<receiver>** là thẻ dành cho các broadcast receivers.

**<provider>** là thẻ dành cho các content providers.

#### **3.2.3 strings.xml**

Đây là file được đặt trong thư mục**res/values** chứa tất cả các text mà ứng dụng của bạn sử dụng. Ví dụ: tên của các buttons, các labels, các text mặc định bạn nên để trong file này.

Code mặc định của file strings.xml:

|  |  |
| --- | --- |
|  | <?xml version="1.0" encoding="utf-8"?>  <resources>      <string name="app\_name">HelloWord</string>      <string name="action\_settings">Settings</string>      <string name="hello\_world">Hello world!</string>  </resources> |

#### **3.2.4 R.java**

R.java tạo ra mối liên kết giữa các**file activity**và các**file resource** (VD: như giữa file**MainActivity.java** và file **strings.xml**). File này được tạo ra tự động và bạn không nên thay đổi nội dung trong file này.

#### **3.2.5 activity\_main.xml**

**activity\_main.xml** là một file layout nằm trong thư mục **res/layout**. File này sẽ được tham chiếu bởi ứng dụng khi tạo ra giao diện trên màn hình. Bạn sẽ phải thay đổi nội dung file này rất nhiều trong khi tạo giao diện cho ứng dụng.

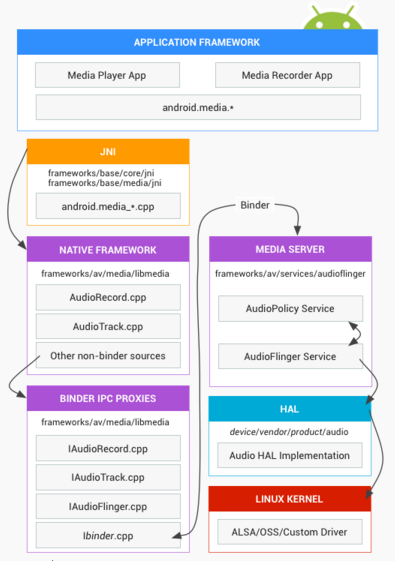
Đây code mặc định của file activity\_main.xml

|  |  |
| --- | --- |
|  | <RelativeLayout xmlns:android="<http://schemas.android.com/apk/res/android>"      xmlns:tools="<http://schemas.android.com/tools>"      android:layout\_width="match\_parent"      android:layout\_height="match\_parent"      tools:context=".MainActivity" >      <TextView          android:layout\_width="wrap\_content"          android:layout\_height="wrap\_content"          android:text="@string/hello\_world" />  </RelativeLayout> |

## 4. Trình bày tổng quan về audio player

### 4.1 Kiến trúc audio

Kiến trúc audio android định nghĩa các thành phần cơ bản và các điểm truy cập có liên quan tới mã nguồn được thực hiện như sơ dồ sau:



* **Application framework (Tầng framwork)**

Tầng này chứa các mã ứng dụng bao gồm các API trong android.media dùng để tương tác với các phần cứng âm thanh. Bên trong, các mã này sẽ gọi tương ứng tới các thành phần tương ứng trong JNI để truy cập vào các mã nguồn tương tác với các phần cứng âm thanh.

* **JNI**

Mã JNI kết hợp với android.media gọi mã nguồn cấp thấp hơn để truy cập vào phần cứng âm thanh. JNI nắm trong frameworks/base/core/jni/ and frameworks/base/media/jni .

* **Native framwork( Tầng native)**

Tầng cơ bản cung cấp các mã nguồn tương đương cơ bản nhất trong gói android.media. gọi Binder IPC proxy để truy cập các dịch vụ âm thanh cụ thể của máy chủ phương tiện truyền thông. Các mã nguồn cơ bản nằm trong frameworks/av/media/libmedia

* **Binder IPC proxy**

Tạo điều kiện giao tiếp qua biên tiến trình. Proxy sẽ được nằm trong frameworks/av/media/libmedia vả bắt đầu với chữ cái “I”.

* **Media server (Máy chủ phương tiện truyền thông)**

Các máy chủ phương tiện truyền thông có các dịch vụ âm thanh, nó là các mã thực tế tương tác với HAL (Hardware Abstraction Layer) của ban. Chúng nằm trong frameworks/av/services/audioflinger

* **HAL**

Định nghĩa các giao diện tiêu chuẩn để các dịch vụ âm thanh gọi tới và thực hiện và bạn phải thực thi phần cứng âm thanh của bạn một cách chính xác. Các giao diện âm thanh được đặt tại hardware/libhardware/include/hardware.

* **Kernel driver (điều khiên nhân)**

Các trình điều khiển âm thanh tương tác với phần cứng của bạn và thực hiện HAL. Bạn có thể sử dụng Advanced Linux Sound Architecture (ALSA), Open Sound System (OSS), hoặc một trình điều khiển tùy chỉnh (HAL là driver-agnostic).

### 4.2 Các thư viện và API có thể sử dụng trong android.media

Để làm việc với audio chúng ta sẽ sử dụng một số thư viện mà android hỗ trợ trong android.media như sau:

**Lớp MediaPlayer**: Lớp android.media.MediaPlayer là được sử dụng để điều kiển những tâp tin như audio hoặc video.

Chúng có chứa một số phương thức được liệt kê sau đây:

|  |  |
| --- | --- |
| **Phương thức** | **Mô tả** |
| public void setDataSource(String path) | bộ nguồn dữ liệu (đường dẫn file hoặc http url) để sử dụng. |
| public void prepare() | chuẩn bị các cầu thủ để phát lại đồng bộ. |
| public void start() | bắt đầu hoặc tiếp tục lại phát lại. |
| public void stop() | dừng phát lại. |
| public void pause() | tạm dừng phát lại. |
| public boolean isPlaying() | kiểm tra nếu máy nghe nhạc phương tiện truyền thông đang phát. |
| public void seekTo(int millis) | tìm cách thời gian quy định trong mili giây. |
| public void setLooping(boolean looping) | đặt máy trong vòng lặp hoặc không lặp. |
| public boolean isLooping() | kiểm tra nếu người chơi được lặp hoặc không lặp. |
| public void selectTrack(int index) | chọn một ca khúc cho các chỉ số xác định. |
| public int getCurrentPosition() | trả về vị trí phát hiện. |
| public int getDuration() | trả về thời gian của tập tin. |
| public void setVolume(float leftVolume,float rightVolume) | đặt âm lượng trên máy nghe nhạc này. |

**Lớp MediaMetadataRetriever:** cung cấp các interface để lấy thông tin về frame và dữ liệu từ đầu vào là tệp tin media.

Bảng sau liệt kê một số dữ liệu lấy được khi gọi lớp MediaMetadataRetriever.

|  |  |  |
| --- | --- | --- |
| int | [METADATA\_KEY\_ALBUM](http://developer.android.com/reference/android/media/MediaMetadataRetriever.html#METADATA_KEY_ALBUM) | Thông tin về tiêu đề album nguồn |
| int | [METADATA\_KEY\_ALBUMARTIST](http://developer.android.com/reference/android/media/MediaMetadataRetriever.html#METADATA_KEY_ALBUMARTIST) | Thông tin người thể hiện liên quan tới tệp tin nguồn |
| int | [METADATA\_KEY\_DATE](http://developer.android.com/reference/android/media/MediaMetadataRetriever.html#METADATA_KEY_DATE) | Dữ liệu về ngày cập nhật của tệp tin media |
| int | [METADATA\_KEY\_DURATION](http://developer.android.com/reference/android/media/MediaMetadataRetriever.html#METADATA_KEY_DURATION) | Dẽ liệu về thời gian của tệp tin media |
| int | [METADATA\_KEY\_GENRE](http://developer.android.com/reference/android/media/MediaMetadataRetriever.html#METADATA_KEY_GENRE) | Dữ liệu về thể loại media |

Các phương thức có thể sử dụng khi dùng như sau:

|  |  |
| --- | --- |
| [String](http://developer.android.com/reference/java/lang/String.html) | [extractMetadata](http://developer.android.com/reference/android/media/MediaMetadataRetriever.html#extractMetadata(int))(int keyCode)  Gọi hàm này sau setDataSource() để giải mã các key dữ liệu |
| [Bitmap](http://developer.android.com/reference/android/graphics/Bitmap.html) | [getFrameAtTime](http://developer.android.com/reference/android/media/MediaMetadataRetriever.html#getFrameAtTime(long))(long timeUs)  gọi hàm này sau setDataSource() để lấy frame tại thời điểm t. |
| [Bitmap](http://developer.android.com/reference/android/graphics/Bitmap.html) | [getFrameAtTime](http://developer.android.com/reference/android/media/MediaMetadataRetriever.html#getFrameAtTime())()  gọi hàm này sau setDataSource() để lấy frame tại thời điểm mặc định đầu tiên |
| void | [release](http://developer.android.com/reference/android/media/MediaMetadataRetriever.html#release())()  Gọi hàm này sau khi đã hoàn thành một đối tượng |
| void | [setDataSource](http://developer.android.com/reference/android/media/MediaMetadataRetriever.html#setDataSource(java.lang.String))([String](http://developer.android.com/reference/java/lang/String.html) path)  Đặt file nguồn để chạy |
| void | [setDataSource](http://developer.android.com/reference/android/media/MediaMetadataRetriever.html#setDataSource(android.content.Context, android.net.Uri))([Context](http://developer.android.com/reference/android/content/Context.html) context, [Uri](http://developer.android.com/reference/android/net/Uri.html) uri)  Đặt file nguồn như là một content uri |

**Interface OnCompletionListener**: tạo sự kiện lắng nghe khi audio hoàn thành

|  |  |
| --- | --- |
| abstract void | [onCompletion](http://developer.android.com/reference/android/media/MediaPlayer.OnCompletionListener.html#onCompletion(android.media.MediaPlayer))([MediaPlayer](http://developer.android.com/reference/android/media/MediaPlayer.html) mp)  Được gọi khi file media chạy hết |

**Interface OnErrorListener**: định nghĩa cho việc gọi lại để xử lí khi mà lỗi xảy ra trong một quá trình không đồng bộ(những lỗi khác sẽ được ném ra ngoại lệ).

|  |  |
| --- | --- |
| abstract boolean | [onError](http://developer.android.com/reference/android/media/MediaPlayer.OnErrorListener.html#onError(android.media.MediaPlayer, int, int))([MediaPlayer](http://developer.android.com/reference/android/media/MediaPlayer.html) mp, int what, int extra)  Được gọi để biểu thị một lỗi xảy ra |

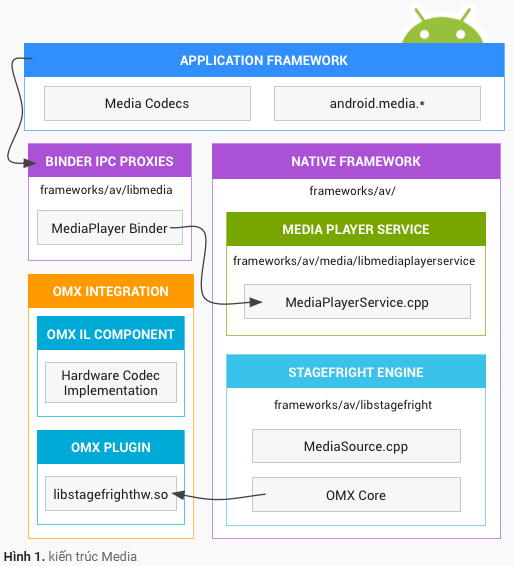
## 5. Trình bày tổng quan về video player

Android cung cấp một cơ chế phát video media ở cấp bản địa gọi là Stagefright mà đi kèm bằng các codec dựa trên phần mềm cho nhiều định dạng media phổ biến. tính năng Stagefright cho âm thanh và phát lại video bao gồm tích hợp với các bộ codec OpenMAX, quản lý phiên, dựng hình thời gian đồng bộ, kiểm soát vận chuyển, và DRM.

Ngoài ra, Stagefright hỗ trợ tích hợp với các bộ codec phần cứng tùy chỉnh mà bạn cung cấp. Nó thực sự không phải là một HAL để thực hiện cho các định dạng tuỳ chỉnh, nhưng để cung cấp một đường dẫn phần cứng để mã hóa và giải mã phương tiện truyền thông, bạn phải thực hiện mã hóa dựa trên phần cứng của bạn như là một thành phần OpenMax IL (Integration Layer).

### 5.1 Kiến trúc

Sơ đồ dưới đây cho thấy cách ứng dụng phương tiện truyền thông tương tác với Android khuôn khổ đa phương tiện bản địa.



* **Application Framework**

Ở cấp khung ứng dụng là mã của ứng dụng, trong đó sử dụng các [android.media](http://developer.android.com/reference/android/media/package-summary.html) API để tương tác với các phần cứng đa phương tiện.

* **Binder IPC**

Các proxy Binder IPC tạo điều kiện giao tiếp qua biên tiến trình. Chúng nằm trong frameworks/av/media/libmedia và bắt đầu bằng chữ "I”.

* **Native Multimedia**

Ở cấp độ bản xứ, Android cung cấp một khuôn khổ đa phương tiện mà sử dụng động cơ Stagefright cho âm thanh và video ghi âm và phát lại. Stagefright đi kèm với một danh sách mặc định của phần mềm codec hỗ trợ và bạn có thể thực hiện mã hóa phần cứng của riêng bạn bằng cách sử dụng các tiêu chuẩn lớp hội nhập OpenMax. Để biết chi tiết thực hiện nhiều hơn, xem MediaPlayer khác nhau và các thành phần Stagefright nằm trong frameworks/av/media.

* **OpenMAX Integration Layer (IL)**

Các OpenMAX IL cung cấp một cách tiêu chuẩn hóa cho Stagefright để nhận biết và sử dụng codec đa phương tiện dựa trên phần cứng tùy chỉnh được gọi là thành phần. Bạn phải cung cấp một plugin OpenMAX trong dạng một thư viện chia sẻ mang tên libstagefrighthw.so . Plugin này liên kết các thành phần mã hóa tùy chỉnh của bạn để Stagefright. Codec tùy chỉnh của bạn phải được thực hiện theo tiêu chuẩn thành phần OpenMAX IL..

### 5.2 Các thư viện và API có thể sử dụng trong android.media

Do việc chạy video là sự kết hợp của audio và tích hợp hình frame trong quá trình chạy Để làm việc với video chúng ta cũng sử dụng một số thư viện mà android hỗ trợ trong android.media như đã trình bày ở phần trên, chúng ta cần kết hợp thêm bộ giải mã hình frame để hiển thị lên thiết bị.Để làm được điều này chúng ta sử dung thư viện android.widget.VideoView.

***Lớp VideoView***: Lớp android.widget.VideoView cung cấp phương pháp để chơi và điều khiển máy video. Các phương thức thường được sử dụng trong lớp VideoView như sau:

|  |  |
| --- | --- |
| **Phương thức** | **Mô tả** |
| public void setMediaController(MediaController controller) | thiết lập điều khiển đa phương tiện để hiển thị video. |
| public void setVideoURI (Uri uri) | thiết lập đường dẫn URI của tập tin video. |
| public void start() | bắt đầu hiển thị video. |
| public void stopPlayback() | ngừng phát lại. |
| public void pause() | tạm dừng phát lại. |
| public void suspend() | đình chỉ việc phát lại. |
| public void resume() | lại tiếp tục được phát ra. |
| public void seekTo(int millis) | tìm cách thời gian quy định trong mili giây. |

***Lớp MediaController*** .Lớp ***MediaController***  sẽ tạo ra một điều khiển mặc định và đặt chúng bên trên nổi lên window ứng dụng. Đặc biệt, điều khiển sẽ nổi lên trên khung nhìn xác định bởi setAnchorView(). Cửa sổ này sẽ mất đi sau chừng 3s và xuất hiện lại khi người dùng chạm vào anchor view. Nó có chứa các điều khiển đa phương tiện như play/pause, trước đó, tiếp theo, nhanh về phía trước, tua lại vv

Một số phương thức có thể dùng theo bảng sau:

|  |  |
| --- | --- |
| void | [hide](http://developer.android.com/reference/android/widget/MediaController.html#hide())()  rời màn điều khiển từ màn hình |
| boolean | [isShowing](http://developer.android.com/reference/android/widget/MediaController.html#isShowing())() |
| boolean | [onTouchEvent](http://developer.android.com/reference/android/widget/MediaController.html#onTouchEvent(android.view.MotionEvent))([MotionEvent](http://developer.android.com/reference/android/view/MotionEvent.html) event)  thực thi phương thức này khi có sự tác động của tay vào màn hình |
| void | [setEnabled](http://developer.android.com/reference/android/widget/MediaController.html#setEnabled(boolean))(boolean enabled)  Đặt trạng thái khả dụng cho view |
| void | [setMediaPlayer](http://developer.android.com/reference/android/widget/MediaController.html#setMediaPlayer(android.widget.MediaController.MediaPlayerControl))([MediaController.MediaPlayerControl](http://developer.android.com/reference/android/widget/MediaController.MediaPlayerControl.html) player) |
| void | [setPrevNextListeners](http://developer.android.com/reference/android/widget/MediaController.html#setPrevNextListeners(android.view.View.OnClickListener, android.view.View.OnClickListener))([View.OnClickListener](http://developer.android.com/reference/android/view/View.OnClickListener.html) next, [View.OnClickListener](http://developer.android.com/reference/android/view/View.OnClickListener.html) prev) |
| void | [show](http://developer.android.com/reference/android/widget/MediaController.html#show(int))(int timeout)  hiện controller lên màn hình theo thời gian timeout |

Các định dạng được hỗ trợ:

Với audio: hiện tại đang hỗ trợ đọc các định dạng MP3,MP4,WMA.

Với video: hỗ trợ các định dạng: MP4,3GP,WMV;

Vậy có những đặc điểm gì mới. Có các đọc điểm chung như việc hiển thị audio, video,tìm kiếm, xem chi tiết xóa… các tác vụ thông thường của một trình phát.

Hướng tới giao diện dễ dàng cho người dùng, em đã thiết kế giao diện sao cho có thể dễ dàng với người dùng nhất để không tốn nhiều công đoạn trong quá trình thực hiện.

Trong tab video: có rất nhiều phần mềm đọc video hiện nay có nhiều tính năng hay và hỗ trợ nhiều định dạng. trong này em cũng đã tích hợp được một số tính năng như: popup player, đọc video online .

Ngoài ra còn có khả năng chia sẻ qua bluetooth, qua facebook.

# CHƯƠNG II. PHÂN TÍCH THIẾT KẾ

## 1. Khảo sát mô tả bài toán

### 1.1. Khảo sát

Ngày nay, sự phát triển về nhu cầu sở hữu các thiết bị kỹ thuật số mà trong đó thiết bị di động có thị phần khá lớn. Sự đòi hỏi về mẫu mã, chất lượng dịch vụ mà đặc biệt là tính năng của chiếc điện thoại, các phần mềm tiện ích đi kèm đã kéo theo sự phát triển của các Hệ điều hành đẻ các nhà phát triển ứng dụng có thể thực hiện các ý tưởng của mình. Các hệ điều hành phổ biến như: Android, Windows Phone, iOS…Cùng với tốc độ phát triển đó là những tiến bộ vượt bậc về tốc độ xử lý. Nhờ đó lập trình các ứng dụng cho loại thiết bị này tăng lên nhanh chóng.

Với mong muốn học tập và nghiên cứu thêm nhiều vấn đề mới. Em xin làm phần mềm media player để học hỏi và tích lũy kiến thức cho các dự án về android sắp tới.b2. Mô tả bài toán

### 1.2. Mục đích

Bài toán “Xây dựng ứng dụng media player hỗ trợ trình đọc video và audio, giúp cho đọc video, audio trở nên dễ dàng hơn. Người dùng có thể học tập, giải trí thông qua các video, audio mà họ có.

### 1.3. Đặc tả bài toán

Hiện nay việc giải trí trên các thiết bị di động là vô cùng lớn. Các ứng dụng giải trí xoay quanh game, video,audio clip rất nhiều.Có nhiều định dạng của các nguồn khác nhau. Có nhiều loại clip, video khác nhau. Do vậy, việc tạo ra một phần mềm hỗ trợ việc đọc các video,audio là cần thiết. Mặc dù có nhiều phần mềm hỗ trợ, nhưng em cũng mong muốn có thể tạo ra một phần mềm cho riêng mình. Nó vừa giúp em học tập thêm kiến thức mới vừa có sản phẩm cho riêng mình. Nâng cao kiến thức cho các sản phẩm tương lai sau này.

**Ứng dụng có các chức năng sau:**

+ Quản lí các video, hiển thị thumbnail cho người dùng, quản lí audio, hiển thị list cho người dùng.

+ Cho phép đọc video, audio có các tác vụ như: tạm dừng, chuyển tiếp, lùi lại;

+ Cho phép chia sẻ video qua bluetooth,facebook;

+ Cho phép tìm kiếm video,xóa video, hiển thị chi tiết thông tin về video;

+ Cho phép trình phát video dạng pop up .

+ Cho phép cut video theo mong muốn người dùng

+ Cho phép phát video online dựa theo tìm kiếm từ khóa online tựa youtube.

* + 1. **Chức năng của hệ thống:**

**Quản lí video**

*Mô tả chức năng:*

Chức năng này cho phép ứng dụng quản lí các video có trong thiết bị của bạn. Ứng dụng sẽ load các video trong thiết bị của bạn và hiển thị chúng dưới dạng thumbnail, cho người dùng có cái nhìn sơ bộ về video của họ, thêm nữa nó còn hiển thị tên video cùng với hình ảnh của video. Ngoài ra có các chức năng như tìm kiếm video, xóa video, xem chi tiết về video cũng được tích hợp.

**Trình đọc video**

Cũng như những phần mềm đọc video khác, nó cũng có các tác vụ như tạm dừng, chuyển tiếp, quay lui, chơi video bạn đang trình chiếu.

**Chia sẻ video**

Khi bạn và người bạn của bạn cùng xem video mà thấy hay , tự dưng muốn lấy về mà không có mạng hay phương tiện khác để sao chép. Thật may khi đó bạn có thể dùng trình chia sẻ video qua bluetooth của ứng dụng để chia sẻ cho bạn bè của bạn một cách hữu ích. Vì hầu hết các điện thoại đều có hỗ trợ bluetooth. Hay khi bạn muốn upload video đó lên trên facebook cho mọi người xem cùng, bạn cũng có thể sử dụng tính năng upload video lên facebook.( Điều kiện, thiết bị của bạn có hỗ trợ phần mềm facebook).

**Cho trình phát video dạng pop-up**

Khi bạn đang xem một bộ phim hay mà có tin nhắn hay một tác vụ khác, bạn không muốn bỏ lỡ mà muốn xem cả thì có thể dùng tác vụ này, nó giúp bạn có thể vừa đọc tin nhắn vừa xem mà không bị gián đoạn

**Cho phép cắt video theo mong muốn**

Khi video của bạn quá dài hay bạn chỉ muốn cắt một phần nào đo để chia sẻ cho bạn bè có thể thông qua bluetooth hay facebook, khi đó trình ứng dụng hỗ trợ việc cut video theo mong muốn của bạn để có được nội dung mong muốn.

**Phát video online dựa theo việc tìm kiếm từ khóa.**

Hiện nay do sự phát triển của công nghệ mạng, đa số hầu hết mọi người đều có thể kết nối với mạng internet cực kì dễ dàng, việc xem các video online là rất phổ biến. Với mong muốn hỗ trợ việc này, em có tích hợp bên trong ứng dụng tính năng xem video online dựa theo tìm kiếm từ khóa, một dang kiểu như youtube. Tính năng này hỗ trợ người dùng khi có kết nội mạng thì có thể xem được các video online mà họ muốn xem.

**Quản lí audio**

Cho phép hiển thị danh sách audio,hiển thị thông tin chi tiết audio.

**Trình phát audio**

Cũng như những phần mềm đọc video khác, nó cũng có các tác vụ như tạm dừng, chuyển tiếp, quay lui, chơi audio bạn đang trình chiếu.

**Chia sẻ qua bluetooth**

cũng như video người dùng có thể chia sẻ qua bluetooth, tuy nhiên chưa thể chia sẻ qua facebook do facebook chỉ hỗ trợ định dạng video.

## 2. Phân tích hệ thống

### 2.1. Mô hình hóa môi trường làm việc

**Tác nhân**

**Người dùng :** Tác nhân này sẽ sử dụng ứng dụng vào mỗi lần mở ứng dụng. Tác nhân này có các quyền như là :

### 2.2. Mô hình hóa chức năng hệ thống

#### **2.2.1. Các ca sử dụng**

|  |  |
| --- | --- |
| Tác nhân | Ca sử dụng |
| Người dùng | Xem danh sách video |
| Xem chi tiết video, xóa, tìm kiếm |
| Chia sẻ video |
| Xem video( dạng thông thường, pop-up) |
| Xem video, tạm dừng , chuyển tiếp, quay lui |
| Cut video mong muốn |
| Xem video online |
| Xem danh sách audio |
| Phát audio( tạm dừng chuyển tiếp, quay lui) |
| Chia sẻ audio |

#### **2.2.2 Biểu đồ ca sử dụng**



*Hình: Biểu đồ ca sử dụng*

3. Mô hình hóa lĩnh vực bài toán



*Hình: Biểu đồ lớp lĩnh vực cho video*



Hình : biểu đồ lớp lĩnh vực cho audio



Hình : biểu diễn lớp lĩnh vực chung cho main activity

### 2.3. Phân tích các ca sử dụng

#### **2.3.1. Đặc tả ca sử dụng**

1. Xem danh sách video

Tên ca sử dụng: Xem danh sách video

Tóm tắt ca sử dụng: Ca sử dụng hiển thị các video sẽ được load từ thiết bị của bạn.

Tác nhân: Người sử dụng

|  |  |
| --- | --- |
| Người sử dụng | Hệ thống |
| 1. 1. Bấm vào icon của ứng dụng | 1. 2. Hiển thị màn hình chính của giao diện |
| 1. 3. Người dùng chuyển tab sang Video | 1. 4. Hiển thị danh sách các video được load từ thiết bị |

2. Xem video dạng thông thường

Tên ca sử dụng: Xem video

Tóm tắt ca sử dụng: Ca sử dụng hiển thị video mà người dùng đã lựa chọn để trình chiếu.

Tác nhân: Người sử dụng

|  |  |
| --- | --- |
| Người sử dụng | Hệ thống |
| 1. 1. Chọn video cần xem từ danh sách | 1. 2. Hiển thị video được lựa chọn và trình chiếu |
| Điều kiện | Người dùng đã vào ứng dụng |

3. Xóa video

Tên ca sử dụng: Xóa video

Tóm tắt ca sử dụng: Để thực hiện xóa video.

Tác nhân: Người sử dụng

|  |  |
| --- | --- |
| Người sử dụng | Hệ thống |
| 1. 1. Từ giao diện chi tiết danh sách các video,người dùng longclick vào một video lựa chọn | 1. 2. Hiển thị context menu để xác nhận việc lựa chọn |
| 1. 3. Người dùng chọn xóa | 1. 4. Hiện dialog xác nhận xóa |
| 5.1 Người dùng xác nhận xóa  5.2 Cancel lựa chọn | 6.1 Xóa video xác nhận và quay lại màn hình danh sách video  6.2 Quay lại màn hình danh sách video  UC bắt đầu lại |

4. Xem chi tiết video

Tên ca sử dụng: Xem chi tiết video

Tóm tắt ca sử dụng: dùng để xem chi tiết về video

Tác nhân: Người sử dụng

|  |  |
| --- | --- |
| Người sử dụng | Hệ thống |
| 1. 1. Từ giao diện chi tiết danh sách các video,người dùng longclick vào một video lựa chọn | 1. 2. Hiển thị context menu để xác nhận việc lựa chọn |
| 1. 3. Người dùng chọn detail | 1. 4. Hiện chi tiết vê video |

1. Share video

Tên ca sử dụng: Share video

Tóm tắt ca sử dụng: dùng để share video qua bluetooth,facebook

Tác nhân: Người sử dụng

|  |  |
| --- | --- |
| Người sử dụng | Hệ thống |
| 1. 1. Từ giao diện chi tiết danh sách các video,người dùng longclick vào một video lựa chọn | 1. 2. Hiển thị context menu để xác nhận việc lựa chọn |
| 1. 3. Người dùng chọn share | 1. 4. Hiện danh sách lựa chọn video cần share |
| 5. Người dùng tích vào video cần share | 1. Hiển thị danh sách video có thể share |
| * 1. Click vào một video cần share   2. Check button all | 8.1 Video đó được tích chọn  8.2 Tất cả các video được chọn |
| 9.1 Người dùng click vào button share   * 1. click vào cancel | 10.1 Màn hình hiển thị content provider cho phép lựa chọn các ứng dụng cho phép share. Thực hiện (11)  10.2 hủy tác vụ , UC bắt đầu lại, quay lại màn hình danh sách video. |
| 11. Người dùng chọn 1 ứng dụng cho phép share  11.1 click bluetooth  11.2 click facebook | 11.1.1 Bluetooth được mở tiến hành tìm thiết bị, kết nối  11.1.2 Chưa mở yêu cầu mở và thực hiện như 11.1.1  11.2 Mở ứng dụng facbook trạng thái status và get video được tải lên. Người dùng tiến hành share |

6.Cut video

Tên ca sử dụng: Cut video

Tóm tắt ca sử dụng: dùng để cut video theo mong muốn người dùng.

Tác nhân: Người sử dụng

|  |  |
| --- | --- |
| Người sử dụng | Hệ thống |
| 1. 1. Từ giao diện player video, người dùng click vào button trimmer video | 1. 2. Hiển thị tùy chọn cho phép cut |
| 1. 3.1. Người dùng chọn start 2. 3.2 Người dùng chọn end 3. 3.3 Người dùng chọn reset   3.4 người dùng chọn cut | 4.1 Get vị trí video hiện tại hiển thị lên textview, thời gian bắt đầu.  4.2 Get vị trí video hiện tại và hiển thị lên textview, thời gian kết thúc.  4.3 Reset thời gian đã định  4.4 Get thời gian đã định từ textview và tiến hành trim video |
| Điều kiện | Ứng dụng đang ở màn hình player video |

1. Xem video online

|  |  |
| --- | --- |
| Người sử dụng | Hệ thống |
| 1. Từ giao diện list video, người dùng click vào button nhỏ ở góc dưới màn hình | 2. Hiển thị một activity mới |
| 1. Người dùng click vào thanh bên trên và gõ vào nội dung tìm kiếm | 1. Hiển thị danh sách các video được tìm thấy dựa trên từ khóa |
| 1. Người dùng click vào một video tùy chọn | * 1. Nếu kết nối mạng thành công Hiển thị player activity cho phép trình chiếu video được chọn   2. Nếu không thì hiển thị thông báo chưa kết nối mạng. |
| Điều kiện | Ứng dụng đang ở màn hình list video.  Ứng dụng phải được kết nối mạng |

1. Share audio

Tên ca sử dụng: Share audio

Tóm tắt ca sử dụng: dùng để share audio qua bluetooth

Tác nhân: Người sử dụng

|  |  |
| --- | --- |
| Người sử dụng | Hệ thống |
| 1. Từ giao diện chi tiết danh sách các audio,người dùng tick vào một audio lựa chọn qua radio tick  Người dùng click vào nút share trên thanh actionbar | 2. Hiển thị activity mới chứa danh sách audio cho phép share |
| 3. Người dùng chọn share (có thể check all trước) | 4. Tiến hành share |

1. Play audio

Tên ca sử dụng: Share audio

Tóm tắt ca sử dụng: dùng để share audio qua bluetooth

Tác nhân: Người sử dụng

|  |  |
| --- | --- |
| Người sử dụng | Hệ thống |
| 1. Từ giao diện chi tiết danh sách các audio,người dùng click vao button play | 2. Play audio tương ứng |

1. Xem chi tiết audio

Tên ca sử dụng: Xem chi tiết audio

Tóm tắt ca sử dụng: dùng để xem thông tin chi tiết về audio

Tác nhân: Người sử dụng

|  |  |
| --- | --- |
| Người sử dụng | Hệ thống |
| 1. Từ giao diện chi tiết danh sách các audio,người dùng click vao button có icon chữ i | 2. Hiển thị activity chứa thông tin về audio tương ứng |

#### **2.3.2. Biểu đồ lớp tham gia ca sử dụng**

1. Xem danh sách video

* Lớp biên: W\_XemDSVideo hiển thị các video được load.
* Điều khiển: GridAdapter điều khiển các chức năng.
* Thực thể tham gia UC: VideoInfo lưu thông tin về danh sách các video,gridAdapter.



2. Xem video

* Lớp biên: W\_XemVideo màn hình hiển thị video trình chiếu phát
* Lớp điều khiển: C\_ XemVideo điều khiển các hoạt động
  + Lớp thực thể tham gia UC: VideoInfo : thông tin video, PlayerActivity: giữ thực thể.



3. Xóa video

* Lớp biên: W\_XoaVideo thông báo video muốn xóa.
* Lớp điều khiển: C\_ XoaVideo điều khiển các hoạt động.
* Lớp thực thể: VideoInfo lưu thông tin về nội dung video sẽ xóa,Dialog.



4. Xem chi tiết

* Lớp biên: W\_XemChiTiet chi tiết về video
* Lớp điều khiển: C\_ XemChiTiet điều khiển các hoạt động xếp lớp
* Lớp thực thể: VideoInfo lưu thông tin video, Dialog.



5. Share Video

* Lớp biên: W\_shareVideo
* Lớp điều khiển: C\_ shareVideo điều khiển các hoạt động
* Lớp thực thể: VideoInfo,ShareVideoActivity .



1. Trim video

Lớp biên : W\_CutVideo

Lớp điều khiển: C\_Cutvideo

Lớp thực thể: VideoPlayerState,VideoTrimmerService,VideoTrimmer,FileUtil



1. Xem video online

Lớp biên: W\_XemVideo

Lớp điều khiển: C\_XemVideo

Lớp thực thể: VideoItem,YoutubeConnector,SearchActivity,PlayerActivity



1. Share Audo. Việc share tương tự như video với định dạng là audio
2. Play audio

Lớp biên: W\_PlayAudio

Lớp điều khiển: C\_Playaudio

Lớp thực thể: AudioDataModel,AudioMainActivity.



1. Xem chi tiết audio

Lớp biên: W\_XemChiTiet

Lớp điều khiển: C\_XemChiTiet

Lớp thực thể: AudiModelData, AudioInformaton.



## 3. Thiết kế

### 3.1. Thiết kế đối tượng

*VideoInfo*

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Giải thích** | **Ghi chú** |
| Path | đờng dẫn |  |
| Type | kiểu của video | Khóa chính |
| Name | Tên video | Khóa chính |
| Time | thời gian video được tạo |  |
| Resolution | Phân giải video |  |
| Duration | độ dài video |  |
| Thumbnail | ảnh thumbnail |  |

*SharevideoActivity*

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Giải thích** | **Ghi chú** |
| cbAll | Check all video |  |
| tvView | hiển thị số video được lựa chọn |  |
| grview | Grid view hiển thị video |  |
| btnShare | Button share |  |
| btnCancel | Button cancel |  |

*PlayerActivity*

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Giải thích** | **Ghi chú** |
| videoview | View hiển thị video |  |
| layout |  |  |
| tvVolum | điều chỉnh volume |  |
| tvBrighless | điều chỉnh độ sáng |  |
| position | vị trí hiện tại video |  |
| path | đường dẫn video |  |
| imgScreen | Image thu nhỏ |  |

*PopUpService*

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Giải thích** | **Ghi chú** |
| popuplayout | Khung pop up |  |
| imgFullScreen | trở lại full screen |  |
| imgTurnOff | Tắt hoàn toàn |  |
| width | độ rộng khung nhìn |  |
| height | độ cao của khung nhìn |  |

DiscoverVideo

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Giải thích** | **Ghi chú** |
| filelist | Danh sách file |  |
| loading | để load khi đọc file |  |

PopUpVideoSerVice

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Giải thích** | **Ghi chú** |
| popupLayout | Để hiển thị trạng thái service |  |
| imageFullScreen | Button trở lại màn hình full |  |
| imageTurnOff | Button tắt hoàn toàn service |  |
| position | Get vị trí hiện tại của video |  |
| name | Get tên video |  |
| url | lấy đường dẫn |  |

PlayVideoOnlineActivity

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Giải thích** | **Ghi chú** |
| playerView | Dùng để hiển thị video |  |
| RECOVERY\_DIALOG\_REQUEST | Yêu cầu chỉ định |  |

VideoOnlineActivity

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Giải thích** | **Ghi chú** |
| searchInput | Dùng để nhập từ khóa tìm kiếm |  |
| videoFound | List hiển thị tìm thấy |  |
| searchResult | Danh sách video tìm thấy |  |

VideoItem

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Giải thích** | **Ghi chú** |
| title | Tiêu đề của video |  |
| description | Mô tả ngắn |  |
| thumbnailURL | Thumbnail hiển thị |  |
| id | Id của video |  |

AudioDataModel

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Giải thích** | **Ghi chú** |
| fileName | Name của audio |  |
| fileSize | Kích thước của audio |  |
| filePath | đường dẫn audio |  |
| fileDuration | độ dài audio |  |
| selected | kiểm tra đã được chọn chưa |  |

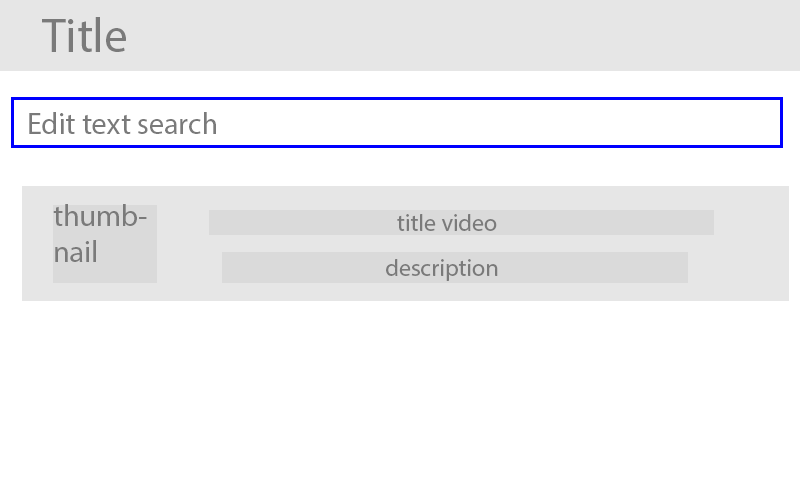
AudioListAdapter

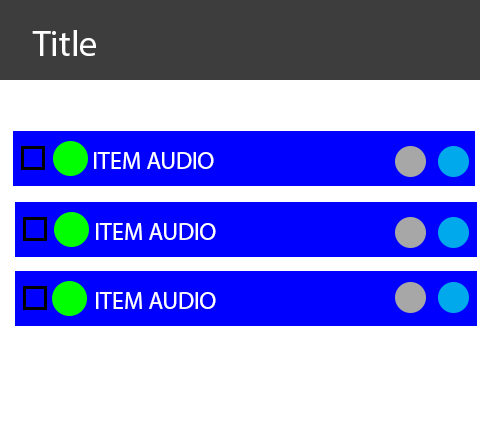
|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Giải thích** | **Ghi chú** |
| selectedFileList | Phần tử thứ mấy được chọn |  |
| audioFileList | Danh sách audio |  |

### 3.2. Thiết kế giao diện

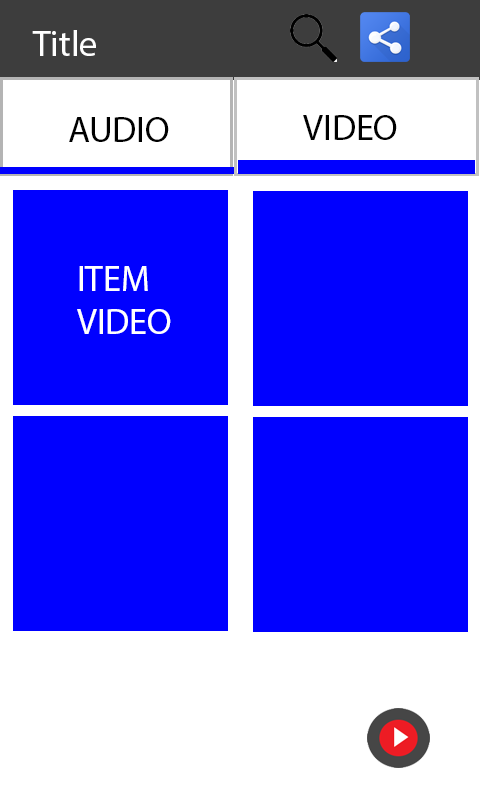
một vài thiết kế cho giao diện của ứng dụng

Màn hình cho video online

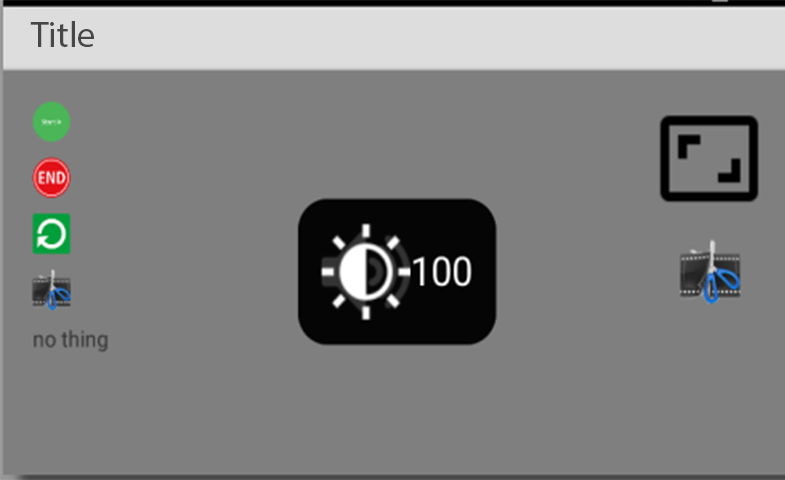




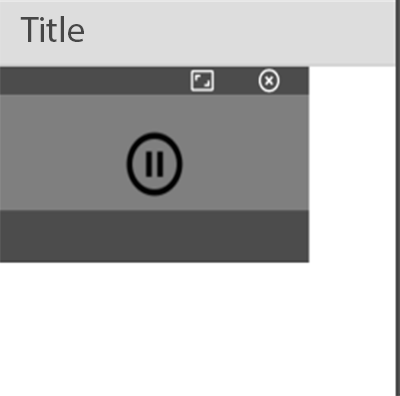
Màn hình cho các item audio khi load



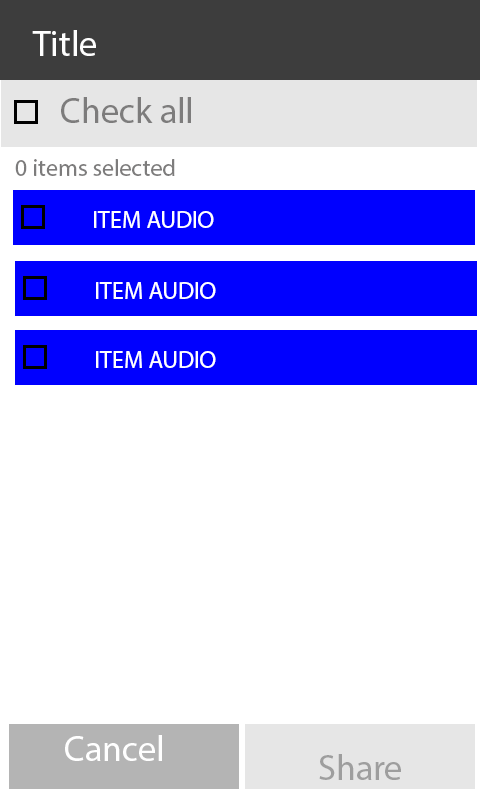
Màn hình cho video main activity



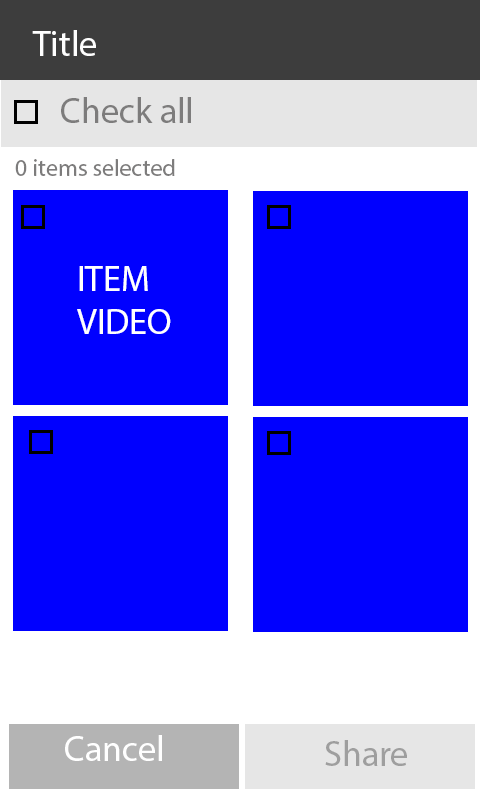
Màn hình cho player activity



Màn hình cho popup video



Màn hình share audio activity



Màn hình share video activity

# CHƯƠNG III. XÂY DỰNG ỨNG DỤNG

## Xây dựng giao diện

Trong chương trình , em sẽ xét tới một số giao diện chính tác vụ trong chương trình. Nội dung trình bày dưới đây sẽ là việc thiết kế design giao diện.Thường chúng ta sẽ dùng XML để thiết kế, và chúng được đặt trong folder res/layout.

### 1.1 Xây dựng giao diện cho audio

#### 1.1.1 Màn hình giao diện chính cho thành phần audio

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:padding="5dp" >

<CheckBox

android:id="@+id/row\_audio\_list\_item\_cbox\_select"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_centerVertical="true" />

<ImageView

android:id="@+id/row\_audio\_list\_item\_iv\_img"

android:layout\_width="25dp"

android:layout\_height="25dp"

android:layout\_centerVertical="true"

android:layout\_toRightOf="@id/row\_audio\_list\_item\_cbox\_select"

android:src="@drawable/btn\_play" />

<LinearLayout

android:id="@+id/linear0"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_centerInParent="true"

android:layout\_toRightOf="@id/row\_audio\_list\_item\_iv\_img"

android:orientation="horizontal" >

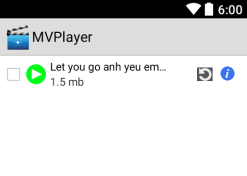
<LinearLayout

android:id="@+id/linear1"

android:layout\_width="150dp"

…….

Và kết quả thu được



#### 1.1.2 Màn hình giao diện cho main audio

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".AudioFileListActivity" >

<RelativeLayout

android:id="@+id/activity\_audio\_file\_list\_rl\_play\_progress"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:layout\_alignParentBottom="true"

android:background="#1F1F1F"

android:gravity="center\_vertical"

android:padding="10dp"

android:visibility="visible" >

<TextView

android:id="@+id/activity\_audio\_file\_list\_tv\_progress"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentLeft="true"

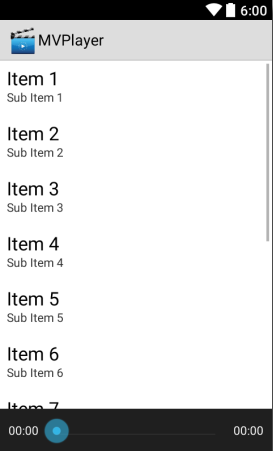
android:layout\_centerVertical="true"

android:text="00:00"

android:textColor="@android:color/white" />

……………….

Và kết quả :



#### 1.1.3 Màn hình chia sẻ data của audio

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical">

<LinearLayout

android:id="@+id/headLinear1"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_alignParentTop="true"

android:orientation="horizontal">

<CheckBox

android:id="@+id/cb\_checkall\_audio"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:button="@drawable/custom\_checkbox"/>

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:gravity="center"

android:text="Check All"

android:textSize="20sp"

android:textStyle="bold" />

</LinearLayout>

<TextView

android:id="@+id/tv\_selected\_audio"

android:layout\_marginTop="5dp"

android:layout\_marginBottom="5dp"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="5dp"

android:text="0 selected"

android:layout\_below="@+id/headLinear1"/>

<ListView

android:id="@+id/gv\_audio"

android:layout\_above="@+id/layoutButton1"

android:layout\_below="@+id/tv\_selected\_audio"

android:background="#e9e7e7"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:horizontalSpacing="5dp"

android:verticalSpacing="5dp"/>

<LinearLayout

android:id="@+id/layoutButton1"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal"

android:layout\_marginTop="5dp"

android:layout\_alignParentBottom="true">

<Button

android:id="@+id/btn\_cancel\_audio"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

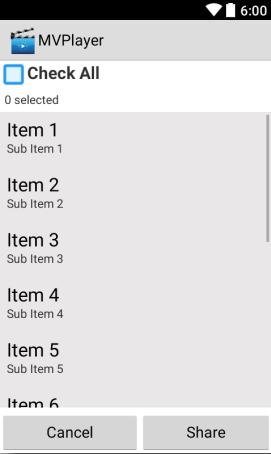
android:text="Cancel"/>

<Button

android:id="@+id/btn\_Share\_audio"

………………………….

Và kêt quả thu được



### 1.2 Xây dựng một số giao diện cho video

#### 1.2.1 Giao diện cho các item video

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout

xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:background="@drawable/borderimage">

<ImageView

android:id="@+id/thumbnail"

android:layout\_width="180dp"

android:layout\_height="180dp"

android:scaleType="centerCrop"

/>

<RelativeLayout

android:layout\_width="180dp"

android:layout\_height="80dp"

android:layout\_marginTop="100dp"

android:orientation="vertical"

android:background="#66000000">

<TextView

android:id="@+id/tv\_videoname"

android:text="CAM01324"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:maxLines="1"

android:layout\_marginRight="20dp"

android:layout\_marginLeft="20dp"

android:ellipsize="end"

android:textSize="20sp"

android:textStyle="bold"

android:textColor="@android:color/white"

android:layout\_centerInParent="true"/>

<TextView

android:id="@+id/tv\_duration"

android:text="12:09"

android:layout\_below="@+id/tv\_videoname"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

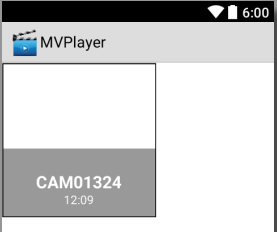
android:layout\_centerHorizontal="true"

android:textColor="@android:color/white"/>

</RelativeLayout>

</RelativeLayout>

Và kết quả thu được



#### 1.2.2 Giao diện khi share video

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

tools:context="com.tuankhac.video.ShareVideoActivity">

<LinearLayout

android:id="@+id/headLinear"

android:background="@android:color/darker\_gray"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_alignParentTop="true"

android:padding="10dp"

android:orientation="horizontal">

<CheckBox

android:id="@+id/cb\_checkall"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:button="@drawable/custom\_checkbox"/>

<LinearLayout

android:id="@+id/layoutButton"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal"

android:layout\_marginTop="5dp"

android:layout\_alignParentBottom="true">

<Button

android:id="@+id/btn\_cancel"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

android:text="Cancel"/>

<Button

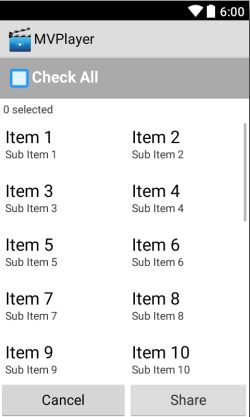
android:id="@+id/btn\_Share"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

…………….

Và kết quả thu được



#### 1.2.3 Giao diện main activity video

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:layout\_margin="5dp"

android:orientation="horizontal"

tools:context="com.tuankhac.activity.MainActivity" >

<GridView

android:id="@+id/gv\_video"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:horizontalSpacing="5dp"

android:numColumns="2"

android:verticalSpacing="5dp" >

</GridView>

<ImageButton

android:id="@+id/watch\_like\_youtube"

android:layout\_width="40dp"

android:layout\_height="40dp"

android:layout\_alignParentBottom="true"

android:layout\_alignParentRight="true"

android:layout\_marginBottom="25dp"

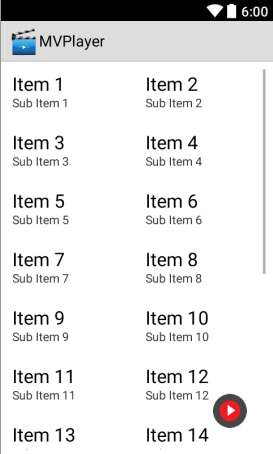
android:layout\_marginRight="25dp"

android:background="@mipmap/watch\_like\_youtube"/>

…………………….

</RelativeLayout>

Và kết quả thu được



#### 1.2.4 Giao diện player video

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context="com.tuankhac.video.PlayerActivity" >

<android.widget.VideoView

android:id="@+id/myvideoview"

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:layout\_alignParentBottom="true"

android:layout\_alignParentEnd="true"

android:layout\_alignParentStart="true"

android:layout\_alignParentTop="true"

android:visibility="visible" />

<LinearLayout

android:id="@+id/layout\_volum"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_centerInParent="true"

android:background="@drawable/borderlayout"

android:orientation="horizontal" >

<ImageView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:contentDescription="@string/desc"

android:src="@mipmap/volum" />

<TextView

android:id="@+id/tv\_volum"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_vertical"

android:text="@string/text\_view"

android:textColor="#FFFFFF"

android:textSize="25sp" />

</LinearLayout>

<LinearLayout

android:id="@+id/layout\_brightnes"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

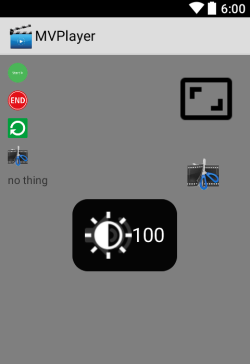
android:layout\_centerInParent="true"

android:background="@drawable/borderlayout"

android:orientation="horizontal" >

………………………….

Và kết quả thu được



#### 1.2.5 Giao diện khi popup video

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout

xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="250dp"

android:layout\_height="140dp"

android:id="@+id/layout\_fa">

<VideoView

android:id="@+id/popup\_player"

android:layout\_centerInParent="true"

android:layout\_width="250dp"

android:layout\_height="140dp" />

<RelativeLayout

android:id="@+id/layout\_top"

android:layout\_width="250dp"

android:layout\_height="20dp"

android:background="#66000000"

android:orientation="horizontal"

android:layout\_alignParentTop="true">

<ImageView

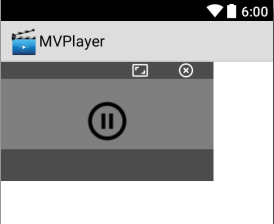
android:id="@+id/img\_fullscreen"

android:src="@mipmap/fullscreen"

android:layout\_width="wrap\_content"

……………………………

Và kết quả thu được:



## 2. Các api sử dụng

### 2.1 Các api được sử dụng để chạy audio

Trong lớp MediaPlayer dùng cách phương thức sau

|  |  |
| --- | --- |
| **Phương thức** | **Mô tả** |
| public void setDataSource(String path) | bộ nguồn dữ liệu (đường dẫn file hoặc http url) để sử dụng. |
| public void prepare() | chuẩn bị để phát lại đồng bộ. |
| public void start() | bắt đầu hoặc tiếp tục lại phát lại. |
| public void stop() | dừng phát lại. |
| public boolean isPlaying() | kiểm tra nếu máy nghe nhạc phương tiện truyền thông đang phát. |
| public void seekTo(int millis) | tìm cách thời gian quy định trong mili giây. |
| public int getCurrentPosition() | trả về vị trí phát hiện. |
| public int getDuration() | trả về thời gian của tập tin. |
| public void release() | loại bỏ chơi khi không muốn chơi tiếp |

#### **a. SetDatasource cho audio**

Code mô tả ví dụ một phương thức .

mp.setDataSource(AudioMainActivity.this, uri);

mp.start();

#### **b. Lấy thông tin audio**

Ta sẽ sử dụng phương thức extractMetadata(Key muốn extract ) để lấy các thông tin của audio. Code mô tả

String mDuration = retrive

.extractMetadata(MediaMetadataRetriever.METADATA\_KEY\_DURATION);

Dùng BroadcastReceiver để update giao diện trong quá trình chạy

BroadcastReceiver mReceiver = new BroadcastReceiver() {

@Override

public void onReceive(Context context, Intent intent) {

if (intent.getExtras() != null) {

int currentProgress = intent.getIntExtra("play\_position", 0);

tvTimeReached.setText(mediaTime((long) currentProgress));

sbPlayProgress.setProgress(currentProgress);

}

}

};

#### **c. Đồng bộ khi chạy**

Để dùng tiến trình ta sẽ implement interface Runnable và gọi như sau:

new Thread(this).start();

@Override

public void run() {

int currentPosition = 0;

int total = mp.getDuration();

while (mp.isPlaying() && currentPosition < total) {

currentPosition = mp.getCurrentPosition();

// broadcast the current progress

callBroadcast(currentPosition);

if (isSearching) {

mp.stop();

AudioListAdapter.isPlaying[AudioListAdapter.mPosition] = false;

}

}

}

đoạn code trên sẽ thực hiện call back để update lại trạng thái giao diện.

#### **d. Tương tác seekbar**

để tương tác với seekbar thực hiện các tác vụ như chuyển tiếp quay lui audio ta sẽ dùng đoạn code sau:

sbPlayProgress.setOnSeekBarChangeListener(**new** OnSeekBarChangeListener() {

@Override

**public** **void** onStopTrackingTouch(SeekBar seekBar) {

**if** (!mp.isPlaying()) seekBar.setProgress(mp.getCurrentPosition());

}

@Override

**public** **void** onStartTrackingTouch(SeekBar seekBar) {

}

@Override

**public** **void** onProgressChanged(SeekBar seekBar,

**int** progress, **boolean** fromUser) {

**if** (fromUser && mp.isPlaying()) {

mp.seekTo(progress);

}

}

});

#### **e. Update giao diện**

Mỗi khi có sự tương tác của người dùng với seekbar cần cập nhật lại trạng thái giao diện ta dùng broastcast receiver

BroadcastReceiver mReceiver = **new** BroadcastReceiver() {

@Override

**public** **void** onReceive(Context context, Intent intent) {

**if** (intent.getExtras() != **null**) {

**int** currentProgress = intent.getIntExtra("play\_position", 0);

tvTimeReached.setText(*mediaTime*((**long**) currentProgress));

sbPlayProgress.setProgress(currentProgress);

}

}

};

#### **f. Tạo sự kiện cho nút bấm bất kì**

Để tạo sự kiện nút bấm ta dùng phương thức sau

holder.checkBox.setOnClickListener(new OnClickListener() {

@Override

public void onClick(View v) {

int chkPos = (Integer) v.getTag();

audioFileList.get(chkPos).setSelected((((CheckBox)v).isChecked()));

if (selectedFileList.contains(String.valueOf(chkPos))) {

……

}

}

### 2.2 Các api sử dụng cho chạy video

#### **a. Load video**

Để load các video từ thẻ nhớ lên ta dùng như sau:

Phương thức đệ qui dùng để lấy file

private ArrayList<File> getAllFile(File dir) {

File[] files = dir.listFiles();

for (File file : files) {

if (file.isDirectory() && !file.isHidden()) {

getAllFile(file);

} else if (file.getName().endsWith(".mp4")

|| file.getName().endsWith(".wmv")

|| file.getName().endsWith(".3gp"))

filelist.add(file);

if (isCancelled())

break;

}

return filelist;

}

Tiến trình chạy nền để đọc video

@Override

protected ArrayList<VideoInfo> doInBackground(File... params) {

ArrayList<VideoInfo> arrVideo = new ArrayList<VideoInfo>();

getAllFile(params[0]);

……

}

Update giao diện trong quá trình load video

@Override

protected void onPostExecute(ArrayList<VideoInfo> videoInfos) {

VideoMainActivity.videoInfos.clear();

VideoMainActivity.videoInfos.addAll(videoInfos);

VideoMainActivity.gridAdapter.notifyDataSetChanged();

loading.cancel();

}

}

Video hoạt động dựa trên audio và kết hợp hình ảnh do vậy nó có thể kế thừa các lớp và phương thức như audio đã kể trên. Sau đây là một số chi tiết khác

VideoView: dùng hiển thị video trong quá trình chạy.

#### **b. Chạy âm thanh của video**

Ta cần chạy và đồng bộ nó với audio để cho phù hợp. đoạn code sau mô tả

mediaController.setMediaPlayer(videoView);

videoView.setMediaController(mediaController);

videoView.setVideoPath(path);

videoView.setOnPreparedListener(new MediaPlayer.OnPreparedListener() {

@Override

public void onPrepared(MediaPlayer mediaPlayer) {

}

});

videoView.requestFocus();

videoView.seekTo(position);

videoView.start();

#### **c. Tương tác video**

Video cũng chứa các phương thức cho việc tạm dừng chuyển tiếp, lùi lại…Ví dụ

videoView.pause();

#### **d. Popup video**

Để gọi popupvideo ta sẽ kế thừa từ một service thay vì một activity.

Trong onCreate() của service ta sẽ khởi tạo một layout dùng để người dùng tương tác với nó. Code như sau

**final** WindowManager.LayoutParams params = **new** WindowManager.LayoutParams(

WindowManager.LayoutParams.***WRAP\_CONTENT***,

WindowManager.LayoutParams.***WRAP\_CONTENT***,

WindowManager.LayoutParams.***TYPE\_PHONE***,

WindowManager.LayoutParams.***FLAG\_NOT\_FOCUSABLE***,

PixelFormat.***TRANSLUCENT***);

params.height = 280;

params.width = 500;

params.gravity = Gravity.***TOP*** | Gravity.***LEFT***;

params.x = 100;

params.y = 100;

windowManager.addView(popupLayout, params);

#### **e. Tương tác layout popup**

Người dùng muốn di chuyển layout trên ta sẽ sử dụng :

popupLayout.setOnTouchListener(**new** View.OnTouchListener() {

**private** **int** initialX;

**private** **int** initialY;

**private** **float** initialTouchX;

**private** **float** initialTouchY;

@Override

**public** **boolean** onTouch(View v, MotionEvent event) {

**switch** (event.getAction()) {

**case** MotionEvent.***ACTION\_DOWN***:

initialX = params.x;

initialY = params.y;

initialTouchX = event.getRawX();

initialTouchY = event.getRawY();

**return** **true**;

**case** MotionEvent.***ACTION\_UP***:

layoutTop.setVisibility(View.***VISIBLE***);

tvVideoName.setVisibility(View.***VISIBLE***);

btnPlay.setVisibility(View.***VISIBLE***);

inVisibleView();

**return** **true**;

**case** MotionEvent.***ACTION\_MOVE***:

params.x = initialX

+ (**int**) (event.getRawX() - initialTouchX);

params.y = initialY

+ (**int**) (event.getRawY() - initialTouchY);

windowManager.updateViewLayout(popupLayout, params);

**return** **true**;

}

**return** **false**;

}

});

#### **f. Cut video**

Để có thể cắt video mong muốn ta dùng phương thức sau:

public int trim\_(String inputFileName, String outFileName, int start,

int duration) {

return trim(inputFileName, outFileName, start, duration);

}

Native method được gọi cơ bản như sau:

JNIEXPORT jint JNICALL Java\_net\_video\_trimmer\_natives\_VideoTrimmer\_trim(JNIEnv \*env,

jclass someclass, jstring inputFile, jstring outFile, jint startTime,

jint length) {

log\_message("Starting to cut");

int numberOfArgs = 12;

char\*\* arguments = calloc(numberOfArgs, sizeof(char\*));

char start[5], duration[5];

const char \*in, \*out;

itoa(startTime, start);

itoa(length, duration);

in = (\*env)->GetStringUTFChars(env, inputFile, 0);

out = (\*env)->GetStringUTFChars(env, outFile, 0);

arguments[0] = "ffmpeg";

arguments[1] = "-i";

arguments[2] = in;

arguments[3] = "-ss";

arguments[4] = start;

arguments[5] = "-t";

arguments[6] = duration;

arguments[7] = "-vcodec";

arguments[8] = "copy";

arguments[9] = "-acodec";

arguments[10] = "copy";

// arguments[11] = "-strict";

// arguments[12] = "experimental";

// arguments[13] = "-ab";

// arguments[14] = "12k";

arguments[11] = out;

int i;

for (i = 0; i < numberOfArgs; i++) {

log\_message(arguments[i]);

}

log\_message("Printed all");

ffmpeg\_main(numberOfArgs, arguments);

log\_message("Finished cutting");

free(arguments);

(\*env)->ReleaseStringUTFChars(env, inputFile, in);

(\*env)->ReleaseStringUTFChars(env, outFile, out);

return 0;

}

Phương thức này trả về một native method để thực hiện quá trình cắt video như ta đã chỉ định trước đó.

#### **g. Share video cũng như audio**

Khi muốn share một media file ta thực hiện như sau:

Intent intentShare = **new** Intent();

intentShare.setAction(Intent.***ACTION\_SEND***);

intentShare.putParcelableArrayListExtra(Intent.***EXTRA\_STREAM***,uris);

intentShare.setType("video/\*");

startActivity(Intent.*createChooser*(intentShare, "Share video through..."));

đoạn trên sẽ khởi tạo một intent và kiểu action\_send và ta có thể truyền theo mong muốn .

#### **h. Thay đổi âm lượng hay ánh sáng video**

Quá trình phát video ứng dụng, người dùng mong muốn thay đổi độ sáng hay âm lượng có thể sử dụng thông qua code sau:

@Override

**public** **boolean** onTouchEvent(MotionEvent event) {

imgScreen.setVisibility(View.***VISIBLE***);

btn\_trim\_start.setVisibility(View.***VISIBLE***);

**if** (mGestureDetector.onTouchEvent(event)) {

**return** **true**;

}

**switch** (event.getAction() & MotionEvent.***ACTION\_MASK***) {

**case** MotionEvent.***ACTION\_UP***:

layoutBrightness.setVisibility(View.***GONE***);

layoutVolum.setVisibility(View.***GONE***);

inVisibleView();

endGesture();

**break**;

**case** MotionEvent.***ACTION\_DOWN***:

layoutBrightness.setVisibility(View.***GONE***);

layoutVolum.setVisibility(View.***GONE***);

inVisibleView();

endGesture();

**break**;

}

**return** **super**.onTouchEvent(event);

}

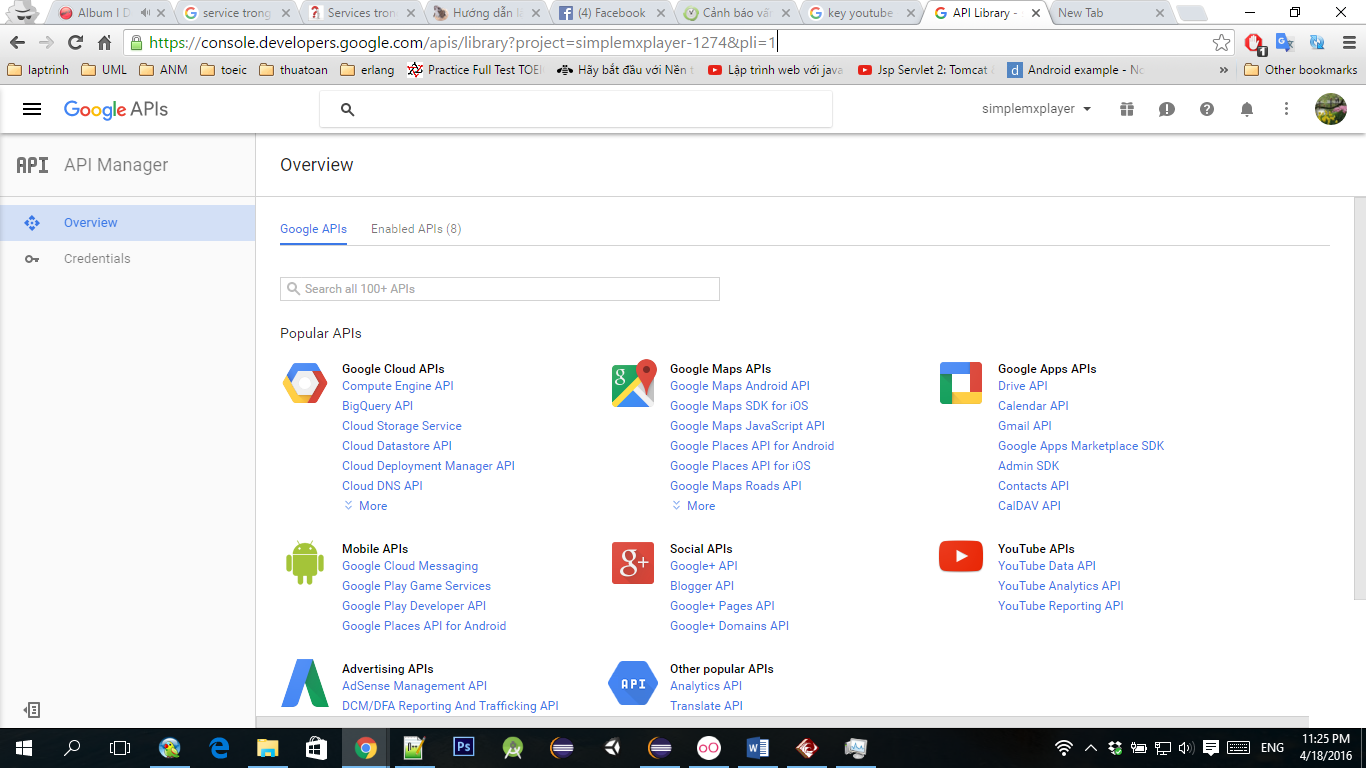
#### **i. Xem video online**

Ứng dụng cho phép người dùng có thể xem trực tiếp video ngay trên mạng. Theo như cách nhiều người gọi thì nó là youtube client. Tức là ta sẽ có thể xem trực tiếp video trên thiết bị tựa như youtube khi thực hiện tác vụ. Chúng sẽ trả về một danh sách video được tìm kiếm theo từ khóa và ta có thể xem các video mà đã tìm được. để làm được vậy chúng ta cần phải đăng kí một key id cho ứng dụng và kết nối tới youtube thông qua key đó. Một số code

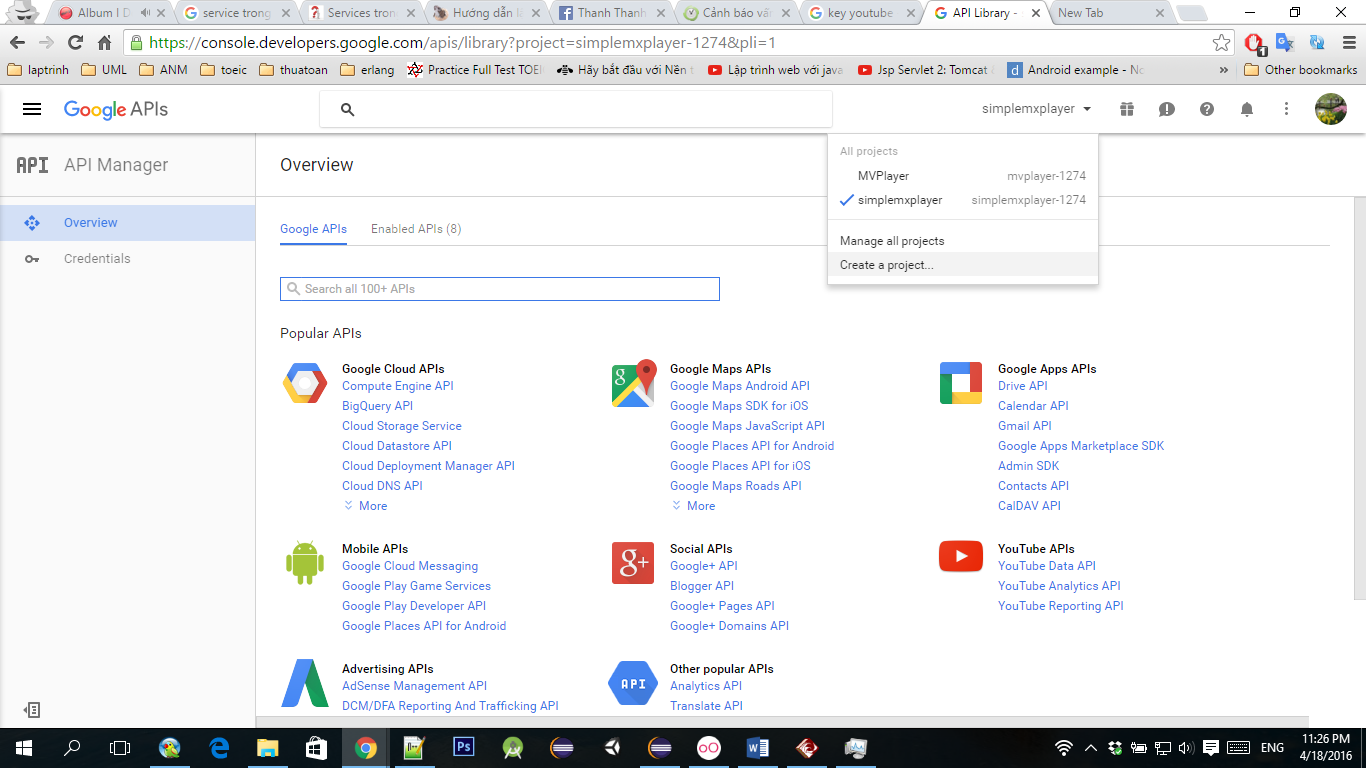
khai bái key mà chúng ta lấy được khi đăng kí với youtube.

Note: cách đăng kí KEY của google như sau:

Đầu tiên cần đăng nhập vào link sau <https://console.developers.google.com>. Google yêu cầu đăng nhập. Bạn cần có một tài khoản với google. Khi đăng nhập xong có giao diện sau:



Bạn cũng cần có một project đăng kí với youtube API v3



Sau khi xong thì ta click vào credentials trên giao diện chọn tiếp create credentials.ta chọn API key, tùy theo ta sẽ chọn. ở đây em chọn android key cho phát triển android. Tiến hành theo các bước như hướng dẫn. để tìm đượcmã SHA1 vào eclipse của ứng dụng copy sang để tạo key như sau vào eclipse-> window-> preferences->android-> build. Ta sẽ nhìn thấy 2 mã MD5 và SHA1 . copy và paste vào mục yêu cầu, sẽ trả về key ta mong muốn.

Ta được key sau

public static final String KEY = "AIzaSyBEi8abO73H6elO3WnlOabtyCcJ9-UUN8M";

thực hiện kết nối với youtube.

public YoutubeConnector(Context context) {

youtube = new YouTube.Builder(new NetHttpTransport(),

new JacksonFactory(), new HttpRequestInitializer() {

@Override

public void initialize(HttpRequest hr) throws IOException {}

}).setApplicationName(context.getString(R.string.app\_name))

.build();

try {

query = youtube.search().list("id,snippet");

query.setKey(KEY);

query.setType("video");

query.setMaxResults((long) 50);

query.setFields("items(id/videoId,snippet/title,snippet/description,snippet/thumbnails/default/url)");

} catch (IOException e) {

Log.d("YC", "Could not initialize: " + e);

}

}

tiến hành search video qua từ khóa mà ta chỉ định

public List<VideoItem> search(String keywords) {

query.setQ(keywords);

try {

SearchListResponse response = query.execute();

List<SearchResult> results = response.getItems();

List<VideoItem> items = new ArrayList<VideoItem>();

for (SearchResult result : results) {

VideoItem item = new VideoItem();

item.setTitle(result.getSnippet().getTitle());

item.setDescription(result.getSnippet().getDescription());

item.setThumbnailURL(result.getSnippet().getThumbnails()

.getDefault().getUrl());

item.setId(result.getId().getVideoId());

items.add(item);

}

return items;

} catch (IOException e) {

Log.d("YC", "Could not search: " + e);

return null;

}

}

#### **Load data chứa hình ảnh có nội dung lớn**

Do dữ liệu trả về khá lớn, khả năng load chậm xảy ra nếu thumbnail của chúng lớn. Khi đó ta sẽ dùng một thư viện Picasso để load nhanh hơn. Code như sau

Picasso.with(getApplicationContext()).load(url).into(thumbnail);

Vì sao nên dùng thư viện picasso?

Thư viện picasso hỗ trợ việc code nhanh gọn đơn giản chỉ với có thể một dòng code.

Một downloader ảnh đơn được tự động tạo khi load ảnh

Vùng nhớ RAM và bộ nhớ đẹm được giải phóng khi kết thúc.

Có thể hủy yêu cầu.

Có thể tải nhiều cùng một lúc.

#### **Một số lưu ý khi làm việc với MediaPlayer và ViewView**

Một khi đã gán data source cho một đối tượng MediaPlayer thì ta không dễ dàng thay đổi nó với một data source khác. Thay vào đó tốt nhất bạn nên tạo một MediaPlayer mới hoặc gọi phương thức MediaPlayer.reset() để khởi tạo lại đối tượng MediaPlayer. Sau khi gọi phương thức MediaPlayer.prepare() bạn có thể gọi phương thức getCurrentPosition(), getDuration() và isPlaying() để kiểm tra trạng thái của MediaPlayer. Ngoài ra bạn còn có thể gọi phương thức setLooping() và setVolume() để yêu cầu MediaPlayer chơi lặp đi lặp lại tệp tin media của bạn cũng như thay đổi âm lượng. Sau khi gọi phương thức Media.start() bạn có thể gọi các phương thức MediaPlayer.pause(), MediaPlayer.stop() và MediaPlayer.seekTo(). Mỗi MediaPlayer sẽ tự tạo một luồng mới để chơi tệp tin media bạn chỉ định, chính vì vậy cần phải gọi phương thức MediaPlayer.release() khi không còn sử dụng MediaPlayer. Nếu bạn sử dụng VideoView để trình diễn một tệp tin video thì VideoView sẽ làm việc này cho bạn. Có thể sử dụng MediaPlayer để chơi tập tin video, tất nhiên là ta chỉ có thể nghe được âm thanh của tệp tin video này mà không xem được hình ảnh.

# KẾT LUẬN

## 1. Những vấn đề đã giải quyết.

Sau một thời gian thực hiện đồ án thực tập tốt nghiệp dưới sự hướng dẫn của TS Phạm Văn Hà, đề tài thực tập tốt ngiệp của em đã thực hiện được các mục tiêu đề ra như sau:

+ Quản lí các video: hiển thị danh sách, xóa video, hiển thị chi tiết về video;

+ Cho phép đọc video có các tác vụ như: tạm dừng, chuyển tiếp, lùi lại;

+ Cho phép chia sẻ video qua bluetooth;

+ Cho phép tìm kiếm video;

+ Cho phép trình phát video dạng pop up .

+ Quản lí audio: xem danh sách audio, xem chi tiết về audio.

+Tìm kiếm audio thông qua từ khóa

+Chia sẻ audio qua bluetooth.

+Trình phát audio: tạm dừng, chuyển tiếp, lùi lại.

Ngoải ra:

+Cho phép cắt video theo mong muốn người dùng

+Xem video online thông qua việc tìm kiếm từ khóa

+Chia sẻ video qua các kênh mạng xã hội facebook.

* **Kết quả đạt được:**

Các vấn đề nêu ra đã giải quyết tương đối tốt

## Mở rộng

Để đáp ứng nhu cầu người dùng và đa dạng tác vụ trên ứng dụng em đã mở rộng thêm ra nhiều tác vụ trong ứng dụng như sau:

* Tích hợp đọc hầu hết các video trên thiết bị

Kết quả: đã thu được kết quả mong muốn, tuy nhiên do việc decode của phần cứng thiết bị mà video đọc chậm hơn so với audio tương ứng

* Tích hợp phần đăng nhập giống youtube để có thể sử dụng tính năng tài khoản của youtube
* Hỗ trợ đọc nhiều định dang audio
* Hỗ trợ download khi người dùng muốn lưu trữ lại trên máy để sử dụng lâu dài
* Tích hợp chạy các audio online để người dùng có thêm trải nghiệm, có thể xem thêm lời bài hát.

## Hướng phát triển

Nghiên cứu hoàn thành các tác vụ trong phần mở rộng

**TÀI LIỆU THAM KHẢO**

https://duythanhcse.wordpress.com/lap-trinh-di-dong/android/

[*http://stackoverflow.com/*](http://stackoverflow.com/)

[*http://www.tutorialspoint.com/*](http://www.tutorialspoint.com/)

[*http://developer.android.com/intl/vi/develop/index.html*](http://developer.android.com/intl/vi/develop/index.html)

[*http://createdineden.com/blog/2015/may/12/android-tutorial-how-to-integrate-youtube-videos-into-your-app/*](http://createdineden.com/blog/2015/may/12/android-tutorial-how-to-integrate-youtube-videos-into-your-app/)

[*http://timtailieu.vn/tai-lieu/tut-huong-dan-xay-dung-ung-dung-media-player-tren-android-1520/*](http://timtailieu.vn/tai-lieu/tut-huong-dan-xay-dung-ung-dung-media-player-tren-android-1520/)

<http://code.tutsplus.com/tutorials/create-a-youtube-client-on-android--cms-22858>

<https://source.android.com/devices/media/>

<http://laptrinhandroid.net.vn/media-player-trong-lap-trinh-android.html>

http://timtailieu.vn/tai-lieu/tut-huong-dan-xay-dung-ung-dung-media-player-tren-android-1520/

Phát triển ứng dụng Smartphone.pdf của Nhóm nghiên cứu và ứng dụng công nghệ A106-Đại học Hoa Sen.