**TRƯỜNG ĐẠI HỌC PHENIKAA KHOA KHOA HỌC CƠ BẢN**

**⸎⸎⸎⸎⸎**

****

**BÀI TẬP LỚN KẾT THÚC HỌC PHẦN MÔN LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG**

**Đề tài :” LẬP TRÌNH ỨNG DỤNG MOBILE**

**XÂY DỰNG APP QUẢN LÝ CHI TIÊU ”**

**Giảng viên : TS. TRẦN ĐĂNG HOAN**

**Sinh viên : VŨ THỊ MIÊN – 20010912**

**: ĐINH VĂN THỨC – 20010927**

**: PHẠM TIẾN THÀNH CÔNG – 2001088**

**LỚP : LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG \_1.2(14IT).2\_LT**

**LINK VIDEO : https://www.youtube.com/watch?v=TcjHdAycWmI**

**HÀ NỘI, THÁNG 8/2021**

Mục lục

[LỜI NÓI ĐẦU 3](#_Toc106013720)

[I.KHÁI NIỆM ANDROID. 4](#_Toc106013721)

[a. khái niệm. 4](#_Toc106013722)

[b. Ưu và nhược điểm của hệ điều hành android. 4](#_Toc106013723)

[II.CÓ THỂ LẬP TRÌNH ANDROID BẰNG NHỮNG NGÔ NGỮ: 4](#_Toc106013724)

[1. Flutter - Ngôn ngữ lập trình Darf 4](#_Toc106013725)

[2. Lập trình Android bằng Java 4](#_Toc106013726)

[3. Lập trình Android bằng C++ 4](#_Toc106013727)

[4. Lập trình Android Kotlin 4](#_Toc106013728)

[5. Bộ ba ngôn ngữ HTML5 + CSS + JavaScript 4](#_Toc106013729)

[6. Lập trình Android bằng Python 4](#_Toc106013730)

[7. Lập trình Android bằng C# 4](#_Toc106013731)

[8. Viết ứng dụng Android bằng ngôn ngữ AngularJS 4](#_Toc106013732)

[Bằng Java tạo ra ứng dụng mobie như thế nào ? 4](#_Toc106013733)

[III.Cấu Trúc của một ứng dụng mobie bằng androi studio (java). 5](#_Toc106013734)

[1. Cấu trúc chuẩn trên Android Studio 5](#_Toc106013735)

[2. Giải thích ý nghĩa các thưc mục 6](#_Toc106013736)

[Thư mục Java 6](#_Toc106013737)

[Thư mục res (Resources) 7](#_Toc106013738)

[Thư mục drawable (res/drawable) 7](#_Toc106013739)

[Thư mục layout (res/layout) 7](#_Toc106013740)

[Thư mục Mipmap (res / mipmap) 8](#_Toc106013741)

[Thư mục values (res/values) 8](#_Toc106013742)

[Thư mục Manifests 9](#_Toc106013743)

[Gradle Scripts 9](#_Toc106013744)

[File Android Layout (Activity\_main.xml) 10](#_Toc106013745)

[File Android Main Activity (MainActivity.java) 11](#_Toc106013746)

[File Android Manifest (AndroidManifest.xml) 12](#_Toc106013747)

[IV.CÁCH TẠO MỘT APP ANDROID BẰNG NGÔN NGỮ JAVA. 13](#_Toc106013748)

[B1. Bắt đầu một dự án. 13](#_Toc106013749)

[B2.Công cụ. 15](#_Toc106013750)

[B3. Làm quen Gradle 16](#_Toc106013751)

[B5. Quản lý AVD 19](#_Toc106013758)

[B6.Trình quản lý SDK 20](#_Toc106013761)

[B7. Đăng ký APK 21](#_Toc106013762)

[V.Phân tích ứng dụng quản lí chi tiêu: 22](#_Toc106013765)

[1.Mục đích: 22](#_Toc106013766)

[2.Giao tiếp với hệ thống: 22](#_Toc106013767)

[3.Công nghệ sử dụng : 22](#_Toc106013768)

[4.Tổng quan về App Quản lí chi tiêu: 22](#_Toc106013769)

[5.Phân tích chức năng: 23](#_Toc106013770)

[6. thêm đối tượng 23](#_Toc106013771)

[7. Sửa thông tin 24](#_Toc106013772)

[8. Xoá thông tin 24](#_Toc106013773)

[VI.Phân tích thiết kế giao diện 25](#_Toc106013774)

[1.Phân tích giao diện tổng quan: 25](#_Toc106013775)

[2.Phân tích giao diện khoản thu và khoản chi 25](#_Toc106013776)

[3.Phân tích giao diện thêm hoặc sửa 25](#_Toc106013777)

[VII.Phân tích hệ thống 26](#_Toc106013778)

[1.OverViewFragment (tổng quan) 26](#_Toc106013779)

[2.AddFragment (Khoản thu) 26](#_Toc106013780)

[3.Ô thêm, sửa 26](#_Toc106013781)

[VIII.Giao diện demo 27](#_Toc106013782)

## LỜI NÓI ĐẦU

Mạng điện thoại di động xuất hiện tại Việt Nam từ đầu năm 1990 và theo thời gian số lượng thuê bao cũng như các nhà cung cấp dịch vụ vận động tại Việt Nam ngày càng tăng. Tin tức ngày càng tăng và nhu cầu sử dụng sản phẩm công nghệ cao nhiều tính năng, cấu hình cao, chất lượng tốt, kiểu mẫu mã đẹp, phong phú nên nhà cung cấp phải luôn luôn cải thiện, nâng cao các sản phẩm mình.

Cùng với sự phát triển của trường điện thoại dị biệt là sự phát triển mạnh mẽ của xu hướng lập phần mềm ứng dụng cho các thiết bị di động. Phần mềm, ứng dụng cho điện thoại di động hiện nay rất đa dạng và phong phú trên hệ điều hành di động cũng phát triển mạnh mẽ và được thay đổi từng ngày. Các hệ điều hành J2ME, Android, IOS, Hybrid, Web dased Mobile Application đã rất phát triển trên thị trường truyền thông di động.

Trong vài năm trở lại đây, hệ điều hành Android ra đời với sự kế thừa những ưu việt hệ điều hành ra đời trước và sự kết hợp của nhiều công nghệ tiên tiến nhất hiện nay, đã được nhà phát triển công nghệ rất nổi tiếng hiện nay là Google. Android nhanh chóng là đối thủ cạnh tranh mạnh mẽ với các hệ thống hành động trước đó và đang là hệ điều hành của tương lai và được nhiều người ưa chuộng nhất.

## I.KHÁI NIỆM ANDROID.

## a. khái niệm.

**- Android** là hệ điều hành có mã nguồn mở dựa trên nền tảng Linux do chính Google phát hành được thiết kế dành cho các nhà phát triển thiết bị, các nhà mạng, lập trình viên có thể tiếp cận nhanh chóng, điều chỉnh tự do trong mã nguồn mở đó.

## b. Ưu và nhược điểm của hệ điều hành android.

**- Ưu điểm:**Hệ điều hành mở, vì hầu hết cách thiết bị di động điều sử dụng nên Android sở hữu kho ứng dụng khổng lồ, khả năng tùy biến cao, dễ dàng đặt lại thiết bị nếu như quên mật khẩu.

**- Nhược điểm:**Hiện tại độ bảo mật của Android là khá cao nhưng sẽ không bằng nếu so sánh với iOS.

## ****II.CÓ THỂ LẬP TRÌNH ANDROID BẰNG NHỮNG NGÔ NGỮ:****

### ****Flutter - Ngôn ngữ lập trình Darf****

### ****Lập trình Android bằng Java****

### Lập trình Android bằng C++

### Lập trình Android Kotlin

### Bộ ba ngôn ngữ HTML5 + CSS + JavaScript

### Lập trình Android bằng Python

### Lập trình Android bằng C#

### Viết ứng dụng Android bằng ngôn ngữ AngularJS

## Bằng Java tạo ra ứng dụng mobie như thế nào ?

***Bước 1: Tạo Project và Add thư viện .Jar***

***Bước 2: dựa vào chuỗi để tạo các class trong Packge Data***

***Bước 3: Code class***

***Bước 4: Tạo Gui(giao diện đồ họa)***

***Bước 5: Xử lý sự kiện***

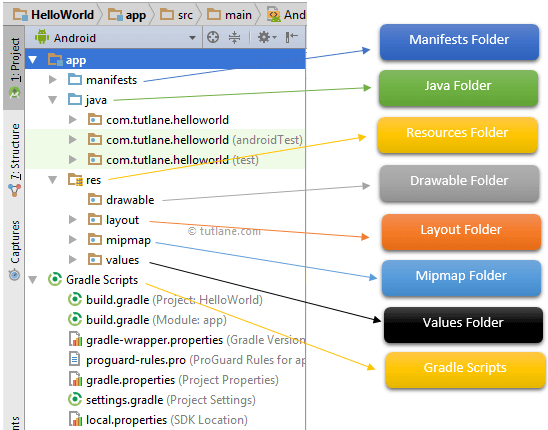
***Bước 6:******Viết Class Start và chạy***

## III.Cấu Trúc của một ứng dụng mobie bằng androi studio (java).

**1. Cấu trúc chuẩn trên Android Studio**

Để phát triển các ứng dụng Android thì bạn cần một IDE, và Android Studio là IDE chính thức được Google cung cấp miễn phí. Nó đã được tích hợp môi trường phát triển nên bạn chỉ việc sử dụng mà không cần quan tâm gì thêm.

Khi thiết lập môi trường phát triển Android và tạo một ứng dụng bằng Android Studio thì bạn sẽ nhận được cấu trúc thư mục project sẽ giống như hình dưới đây.



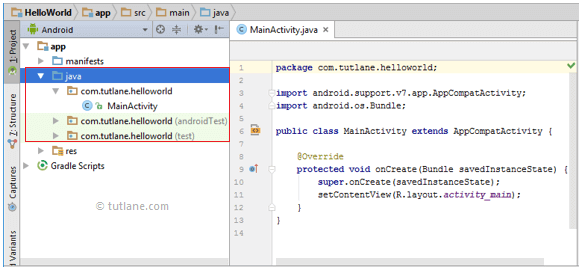
Cấu trúc project của bạn có thể khác với minh họa trên. Để xem cấu trúc file thực tế của project, chọn Project từ menu dropdown nằm ở góc trên bên trái (mặc định chọn Android).

**2. Giải thích ý nghĩa các thưc mục**

Project của ứng dụng Android sẽ chứa nhiều loại app modules, source code files và resource files khác nhau. Ta sẽ khám phá tất cả các thư mục và file ở dưới đây.

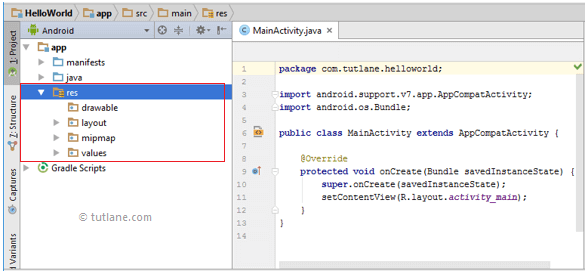
**Thư mục Java**

Thư mục này sẽ chứa tất cả các file mã nguồn java ( .java ) mà chúng ta sẽ tạo trong quá trình phát triển ứng dụng, bao gồm cả mã JUnit test code. Bất cứ khi nào tạo bất kỳ project / ứng dụng mới nào, file lớp MainActivity.java sẽ tự động tạo trong package là com.tutlane.helloworld, giống như dưới đây.



**Thư mục res (Resources)**

Đây là một thư mục quan trọng sẽ chứa tất cả các resource không phải code, chẳng hạn như ảnh bitmap, UI strings, XML layouts như hiển thị bên dưới.



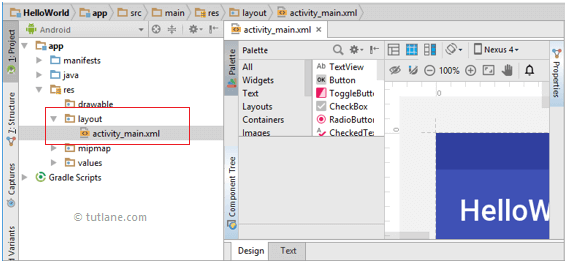
Thư mục res (Resources) sẽ chứa một số loại thư mục khác :

**Thư mục drawable (res/drawable)**

Nó sẽ chứa các dạng ảnh khác nhau. Tốt nhất là nên thêm tất cả các hình ảnh vào thư mục drawable ngoại trừ các biểu tượng ứng dụng / laucher.

**Thư mục layout (res/layout)**

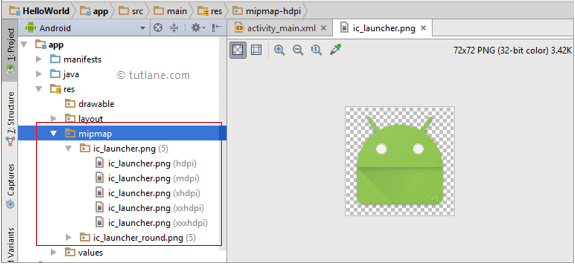
Thư mục này sẽ chứa tất cả các file XML layout đã sử dụng để xác định Giao diện người dùng của ứng dụng. Sau đây là cấu trúc của thư mục layout.



**Thư mục Mipmap (res / mipmap)**

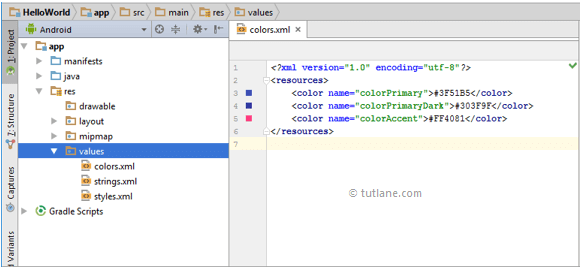
Thư mục này sẽ chứa các biểu tượng ứng dụng / laucher được sử dụng để hiển thị trên màn hình chính. Các loại biểu tượng sẽ có tỷ trọng khác nhau như hdpi, mdpi, xhdpi, xxhdpi, xxxhdpi, để sử dụng dựa trên kích thước của thiết bị.

Sau đây là cấu trúc của thư mục mipmap.



**Thư mục values (res/values)**

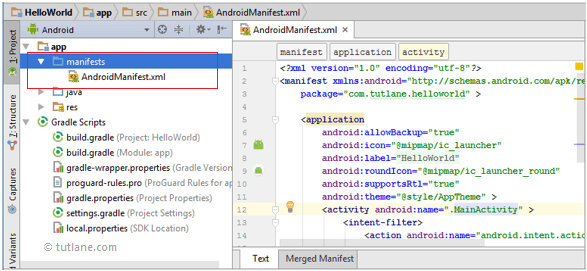
Thư mục này sẽ chứa một số file XML khác nhau, chẳng hạn như strings, colors, styles . Sau đây là cấu trúc của thư mục values.



**Thư mục Manifests**

Thư mục này sẽ chứa một file Manifest ( AndroidManifest.xml ) cho ứng dụng Android . File manifest này sẽ chứa thông tin về ứng dụng , chẳng hạn như android version, access permissions, metadata, v.v. và các component của ứng dụng. File manifest sẽ hoạt động như một trung gian giữa hệ điều hành Android và ứng dụng .

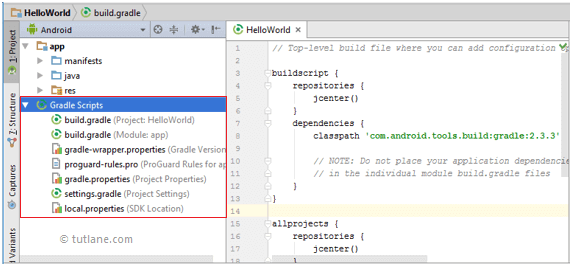
Sau đây là cấu trúc của thư mục mainfests.



**Gradle Scripts**

Trong Android, Gradle là công cụ build hệ thống và Gradle được tích hợp sẵn vào Android Studio, và được điều khiển một cách tự động thông qua Android Studio. Trong gradle có **build.gradle (Project)**và **build.gradle (Module)** được sử dụng để build các cấu hình áp dụng cho tất cả các module ứng dụng hoặc dành riêng cho một mô-đun ứng dụng.

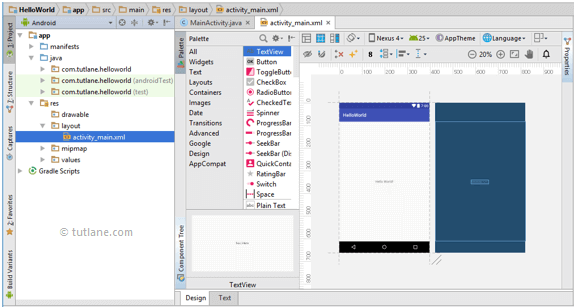
Sau đây là cấu trúc của Gradle Script .



sau đây là các file quan trọng cần để triển khai một ứng dụng trong android studSio.

**File Android Layout (Activity\_main.xml)**

Giao diện người dùng của ứng dụng sẽ được thiết kế trong file này và nó sẽ có hai chế độ **Design**và **Text**. Nó sẽ tồn tại trong thư mục layout. Cấu trúc của file **activity\_main.xml** trong chế độ **Design**như dưới đây.



Ta có thể sửa đổi file activity\_main.xml bằng cách sử dụng các chế độ **Design**và **Text**. Nếu chúng ta chuyển sang chế độ **Text**, file activity\_main.xml sẽ chứa code như dưới đây.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18 | <?xml version="1.0" encoding="utf-8"?>  <android.support.constraint.ConstraintLayout xmlns:android="<http://schemas.android.com/apk/res/android>"      xmlns:app="<http://schemas.android.com/apk/res-auto>"      xmlns:tools="<http://schemas.android.com/tools>"      android:layout\_width="match\_parent"      android:layout\_height="match\_parent"      tools:context="com.tutlane.helloworld.MainActivity">      <TextView          android:layout\_width="wrap\_content"          android:layout\_height="wrap\_content"          android:text="Hello World!"          app:layout\_constraintBottom\_toBottomOf="parent"          app:layout\_constraintLeft\_toLeftOf="parent"          app:layout\_constraintRight\_toRightOf="parent"          app:layout\_constraintTop\_toTopOf="parent" />      </android.support.constraint.ConstraintLayout> |

**File Android Main Activity (MainActivity.java)**

File main activity trong ứng dụng Android là **MainActivity.java** và nó nằm trong thư mục **java**. File **MainActivity.java** sẽ chứa mã java để xử lý tất cả các activity liên quan đến ứng dụng .

Sau đây là mã mặc định file MainActivity.java của ứng dụng HelloWorld.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | package com.tutlane.helloworld;  import android.support.v7.app.AppCompatActivity;  import android.os.Bundle;  public class MainActivity extends AppCompatActivity {      @Override      protected void onCreate(Bundle savedInstanceState) {          super.onCreate(savedInstanceState);          setContentView(R.layout.activity\_main);      }  } |

**File Android Manifest (AndroidManifest.xml)**

Nói chung, ứng dụng sẽ chứa nhiều activity và cần xác định tất cả các activity trong file AndroidManifest.xml . Trong file manifest, ta cần đề cập đến main activity cho ứng dụng bằng cách sử dụng các thuộc tính MAIN action và LAUNCHER category trong các **intent filters (<intent-filter>)**. Trong trường hợp nếu ta không đề cập đến MAIN action và LAUNCHER category cho main activity, biểu tượng ứng dụng sẽ không xuất hiện trong danh sách ứng dụng của màn hình chính.

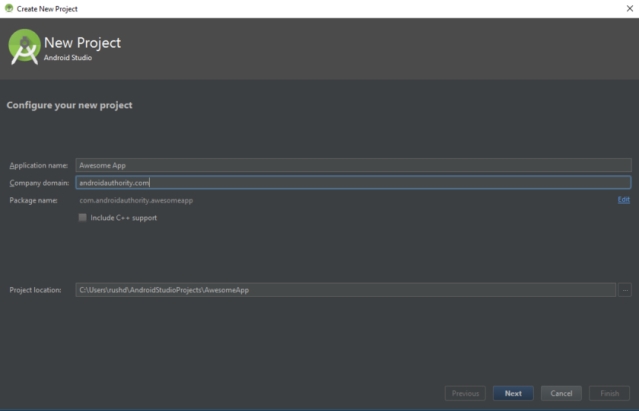
Sau đây là mã mặc định file AndroidManifest.xml của ứng dụng HelloWorld.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | <?xml version="1.0" encoding="utf-8"?>  <manifest xmlns:android="<http://schemas.android.com/apk/res/android>"      package="com.tutlane.helloworld" >        <application          android:allowBackup="true"          android:icon="@mipmap/ic\_launcher"          android:label="@string/app\_name"          android:roundIcon="@mipmap/ic\_launcher\_round"          android:supportsRtl="true"          android:theme="@style/AppTheme" >          <activity android:name=".MainActivity" >              <intent-filter>                  <action android:name="android.intent.action.MAIN" />                  <category android:name="android.intent.category.LAUNCHER" />              </intent-filter>          </activity>      </application>  </manifest> |

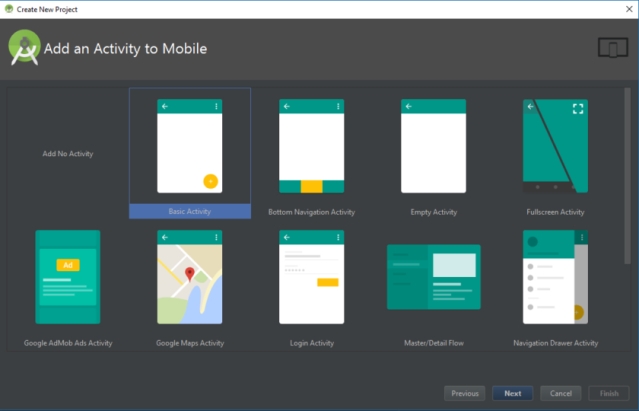
# **IV.CÁCH TẠO MỘT APP ANDROID BẰNG NGÔN NGỮ JAVA.**

## B1. Bắt đầu một dự án.

Mở **Android Studio** và chọn **New Project** hoặc truy cập **File > New > New Project** từ IDE.

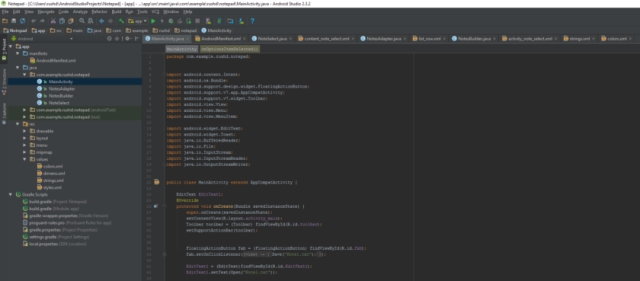


Sau đó, chọn loại hoạt động (Activity) bạn muốn. Activity là “màn hình” thực sự của một ứng dụng.



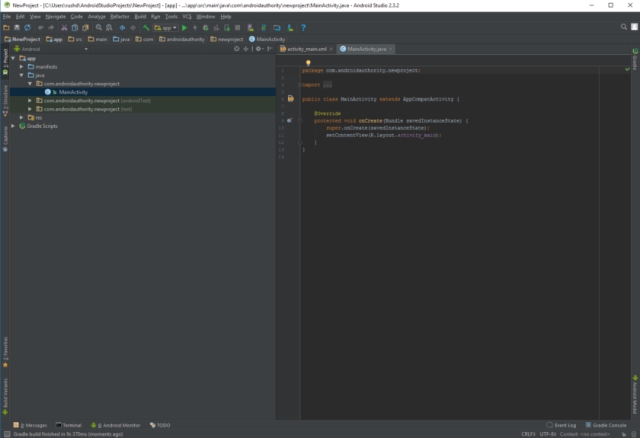
Và chọn “**Basic Activity**” - Nó sở hữu giao diện Android mặc định và trông như một ứng dụng mới.

Chọn “**Basic Activity**” thay vì “**Empty Activity**”, bạn sẽ có một **activity\_main.xml** thiết lập vị trí FAB, các nhân tố UI khác và **content\_main.xml** thêm nội dung vào giữa màn hình. Cuối cùng, bạn có thể thêm “**view**” (các nhân tố như nút bấm, text box và danh sách).

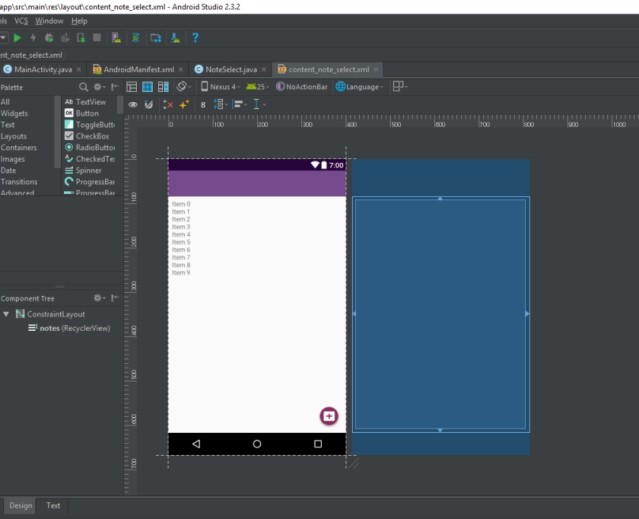


# **B2.Công cụ.**

Chúng ta đều biết ưng dụng Android được hợp thành bởi nhiều file khác nhau. Nhiệm vụ của Android Studio là lưu giữ tất cả chúng ở cùng một vị trí. Cửa sổ chính bên phải màn hình hiển thị từng scrip và file, các tab dọc theo phía trên cho phép chuyển đổi cửa sổ đang mở.



Nếu muốn mở hoạt động mới, di chuyển tới khu vực phân cấp file bên trái. Tại đây, bạn sẽ thấy tất cả thư mục và các thư mục bên trong chúng. File Java nằm trong java, tiếp đến là tên gói ứng dụng. Nhấp đúp vào **MainActivity.Java**, nó sẽ hiện ở phía trên cửa sổ bên phải.



Khi chỉnh sửa file XML, bạn có thể chú ý tới 2 tab bên dưới. Chúng cho phép bạn chuyển đổi giữa trình xem “**Tex**t” và “**Design**”. Ở Text, bạn có thể thay đổi code XML trực tiếp bằng cách thêm và chỉnh sửa các dòng. Ở Design, bạn có thể thêm, di chuyển, kéo các nhân tố riêng lẻ xung quanh màn hình và xem chúng trông như thế nào. Trình xem Text có cửa sổ **Preview** hiển thị hình ảnh bạn đang tạo.

## B3. Làm quen Gradle

## Gradle là “công cụ tự động xây dựng” giúp Android Studio biến tất cả những file khác nhau thành một APK duy nhất.

## Nếu mọi thứ dừng hoạt động, bạn có thể chọn ****Build > Clean Project****. Nó sẽ xác nhận lại vị trí toàn bộ tập tin và vai trò của chúng. Bình thường sẽ có hai file trong số các tệp ****Gradle****: một cho dự án, một cho module ứng dụng.

## 

## B4. Gỡ lỗi, thiết bị ảo và quản lý SDK

## Khi đã sẵn sàng chạy thử ứng dụng, bạn có hai lựa chọn: chạy trên thiết bị vật lý hoặc tạo thiết bị ảo để kiểm tra ứng dụng.

## 

Chạy nó trên thiết bị của bạn rất đơn giản. Chỉ cần cắm nó vào USB chứa ứng dụng (đảm bảo bạn đã phân quyền gỡ lỗi và cài đặt USB từ nguồn không xác định trong cấu hình điện thoại), sau đó, nhấn “**Run > Run App**”.

Bạn sẽ nhận được một tin nhắn thông báo Gradle build đang chạy (nghĩa là code của bạn đang được dựng thành ứng dụng hoàn chỉnh), sau đó, nó sẽ “sống” trên thiết bị của bạn.

## Trong khi ứng dụng chạy, bạn có thể xem báo cáo trực tiếp qua tab “****logcat****” trong Android Monitor.

## 

## Nếu xảy ra lỗi khiến ứng dụng của bạn gặp sự cố hoặc không phản hồi, mô tả lỗi bằng dòng chữ màu đỏ sẽ hiện ra. Lỗi có thể do thiếu quyền hay vấn đề nào đó dễ khắc phục.

## B5. Quản lý AVD

## Về cơ bản, nó là một trình giả lập bạn có thể sử dụng để bắt chước giao diện, hiệu suất của thiết bị Android bất kỳ cùng cài đặt như kích thước màn hình, năng suất, phiên bản

## 

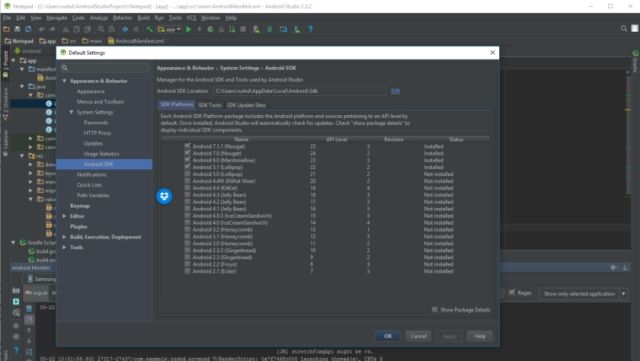
## Khi đã thiết lập một vài thiết bị sử dụng, sau đó, bạn có thể chọn một trong số chúng để chạy ứng dụng và gỡ lỗi y hệt như trên thiết bị vật lý.

## 

Ứng dụng tự tạo không thể chạy trên Surface Pro 3, nhưng trên MSI GT72VR 6RE, nó có thể chạy ở chế độ tăng tốc khá nhanh.Bạn hoàn toàn có khả năng truy cập Play Store để tải nó. Nếu có phần cứng, đây là một cách khả thi để chạy một số ứng dụng trên PC chạy Windows!

## B6.Trình quản lý SDK

Nếu muốn nhắm tới một phiên bản Android hay tạo máy ảo chạy ở phiên bản nhất đinh, bạn sẽ cần download nền tảng và công cụ SDK cần thiết qua trình quản lý SDK bằng cách chọn **Tools > SDK Manager**. Tại đây, bạn cũng sẽ có thể tìm thêm các tài nguyên như **Google Glass Development Kit hay Android Repository** cung cấp chức năng bổ sung để sử dụng trong ứng dụng của bạn.



## B7. Đăng ký APK

## Cuối cùng, khi hoàn thành thử nghiệm, Hãy chọn Build > Generate Signed APK. Thao tác này sẽ cung cấp file upload cần thiết lên Google Play.

## 

## Một thông báo nhắc bạn tạo hoặc nhập Key. Đây là một hình thức chứng nhận xác thực chứng minh APK upload là ứng dụng bạn đang nói tới.

# **V.Phân tích ứng dụng quản lí chi tiêu:**

# 1.Mục đích:

Hiện nay việc quản lí chi tiêu là vấn đề mà mỗi cá nhân và gia đình rất quan tâm đến. Thấu hiểu được vấn đề này nhóm chúng em đã tạo nên một ứng dụng mobile tiện ích và đơn giản để mỗi cá nhân đều có thể tiếp cận và sử dụng một cách dễ dàng.  
 Nhằm đáp ứng nhu cầu ở mọi độ tuổi đều có thể sử dụng được. Ứng dụng quản lí chi tiêu hướng đến sự tối giản nhất.

# 2.Giao tiếp với hệ thống:

Để sử dụng được ứng dụng người dùng có 3 thao tác chính :

+Thêm, sửa, xoá

# **3**.Công nghệ sử dụng :

\*Môi trường sử dụng:

-Android Studio

-Device manager

-Android 12

\*Tài liệu tham khảo:

Google,github,…….

# 4.Tổng quan về App Quản lí chi tiêu:

-Gồm 1 đối tượng sử dụng

-Chức năng sử dụng:

+Thống kê

+Thu

+Chi

+Lọc chi tiêu theo tháng năm

# 5.Phân tích chức năng:

* Thống kê: Tính tổng khoản thu, khoản chi,số dư tài khoản của người sử dụng.

Cách tính số dư : số dư = tổng thu – tổng chi.

* Tổng Thu :Bằng tổng số tiền người dùng thu nhập được(Người dùng tự thêm vào) Mỗi khoản thu người dùng sẽ dùng phím (+) ở góc dưới bên phải màn hình để thêm vào.

* Tổng chi: Bằng tổng số tiền người dùng chi ra (Người dùng tự thêm vào ) Mỗi khoản chi người dùng sẽ dùng phím (+) bên dưới góc phải màn hình để thêm vào.
* Ngoài ra chức năng xoá cũng là một phần không thể thiếu:

Để người dùng có thể xoá những dữ liệu nhập sai người dùng có thể dùng icon (x) ở cuối mỗi phần người dùng nhập

Các yêu cầu chức năng:

# 6. thêm đối tượng

+ mục đích

Thêm các thông tin về đối tượng mới vào cơ sở dữ liệu.

+ Điều kiện trước:

Nhập thông tin của đối tượng

+Điều kiện sau:

Thêm thông tin về đối tượng thành công hoặc thất bại

# 7. Sửa thông tin

+Mục đích:

Sửa thông tin đối tượng đã được thêm.

+Điều kiện trước:

Đối tượng đã được thêm vào.

+điều kiện sau:

Sửa thành công hoặc thất bại.

# 8. Xoá thông tin

+ Mục đích :

Xoá đối tượng khỏi cơ sở dữ liệu.

+Điều kiện trước:

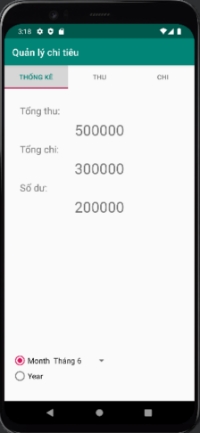
Đối tượng đã có trong cơ sở dữ liệu

+điều kiện sau:

Đối tượng được xoá thành công.

Giao diện cơ bản của ứng dụng:

Thống kê: Thu : Chi:



# **VI.Phân tích thiết kế giao diện**

Giao diện chính ta có 3 phần: tổng quan, khoản chi, khoản thu.Ở mỗi giao diện chính sẽ bao gồm từ 2-3 file ghép lại

# 1.Phân tích giao diện tổng quan:

* Bao gồm 2 phần chính: phần kết quả và phần tìm kiếm
* Phần kết quả bao gồm các TextView: để show kết quả
* Phần tìm kiếm: bao gồm select radio: để lựa chọn, và thẻ Spinner để lựa chọn tháng, năm

# 2.Phân tích giao diện khoản thu và khoản chi

* Ở đây giao diện thu và chi sẽ gần tương ứng vói nhau
* Bao gồm phần RecyclerView để list các khoản thu và chi
* Bao gồm 1 nút để thêm khoản thu hoặc chi mới

# 3.Phân tích giao diện thêm hoặc sửa

* Bao gồm 1 phần Spinner để lựa chọn thể loại. và các thẻ editext để thêm hoặc sửa dữ liệu

# **VII.Phân tích hệ thống**

Ứng dụng sẽ chạy từ AndroidManifest.xml, sau đó sẽ gọi MainActivity.java

Sau đó từ MainActivity.java ta gọi các hàm để show giao diện tổng quan, Khoản thu, khoản chi (OverViewFragment.java, AddFragment.java, SubFragment.java) từ hàm setupViewPager

# 1.OverViewFragment (tổng quan)

* Bước đầu ứng hàm sẽ lấy tất cả những biến id cần từ giao diện đồ họa
* Bước tiếp thêm các dữ liệu vào phần lọc: tháng và năm
* Sau đó gọi hàm để lấy dữ liệu, và xuất ra màn hình
* Lắng nghe các sự kiện thay đổi tháng năm để cập nhập dũ liệu

# 2.AddFragment (Khoản thu)

* Nhận dữ liệu và in ra màn hình
* Lắng nghe sự kiện ấn nút tạo => hiện lên ô để thêm
* Lắng nghe sự kiện ấn vào nút xóa => xóa khoản
* Lắng nghe sự kiện ấn vào các khoản => hiện lên ô để sửa

# 3.Ô thêm, sửa

* Hiện giao diện cho người nhập
* Lắng nghe sự kiện nút xóa khi ấn vào thì ẩn đi
* Lắng nghe sự kiện nút thêm khi ấn vào sẽ kiểm ra xem các dữ liệu có trống không, nếu trống thông báo cho người dùng, không trống sẽ thêm mới

# **VIII.Giao diện demo**

