**1.Phân biệt các dạng ứng dụng di động:**

**-Native App**

* Là những ứng dụng được viết riêng cho một loại nền tảng như iOS, Android, Windows Phone…bằng các ngôn ngữ tương ứng của mỗi nền tảng đó.Ví dụ: Java, Kotlin trên Android; Object C, Swift trên iOS, C# trên Windows Phone.
* Mỗi Native App chỉ chạy được trên một nền tảng. Đa phần các game mobile hiện nay là Native App.

**-** **Mobile Web App**

* Mobile Web App được thiết kế chạy trên các trình duyệt của thiết bị di động. Người dùng có thể thực hiện trực tiếp một công việc nào đó bằng cách sử dụng Mobile Web App, không cần phải cài đặt từ kho ứng dụng.
* Ngôn ngữ lập trình: HTML, CSS, JavaScript…
* Ví dụ : [https://m.facebook.com](https://m.facebook.com/) là một Web App cho di động.

**-Hybrid**

* Là ứng dụng kết hợp của cả Mobile Web App và Native App.
* Hybrid được xây dựng dựa trên các công nghệ Web phổ biến như JavaScript, HTML hay CSS. Khác biệt của Hybrid là ứng dụng này được lưu trong ứng dụng native sử dụng cửa sổ trình duyệt Web View.
* Framework hybrid app sử dụng phổ biến là Flutter, React Native, Xamarin.
* Ví dụ: Báo mới, Dân trí…

**2.Giới thiệu về hệ điều hành Android**

-Hệ điều hành mã nguồn mở dựa trên nền tảng Linux.

-Trước đây, Android được phát triển bởi công ty liên hợp Android (Android Inc), sau đó được Google mua lại vào năm 2005. Phiên bản đầu tiên phát hành năm 2007.

-Phiên bản:



**- Android 11:** + API level: 30+ Version code: R (Red Velvet Cake)**- Android 12:** + API level: 31+ Version code: S (Snow Cone)**3.Công cụ và ngôn ngữ lập trình xây dựng Android App**

**-Công cụ:**

+Eclipse: Trước năm 2015, để lập trình ứng dụng Android cần cài đặt Eclipse, sau đó cài đặt thêm plugin vào Eclipse, sẽ cho phép lập trình các ứng dụng Android.

+Android Studio:

Năm 2015 Google cho ra mắt Android Studio, một công cụ lập trình dành riêng cho ứng dụng Android, và chính thức không hỗ trợ plugin cho Eclipse nữa.

Android Studio được xây dựng dựa trên IntelliJ IDEA nhưng hỗ trợ thêm nhiều chức năng để lập trình Android app nhanh và dễ dàng.

**-Ngôn ngữ lập trình:**

+Java

+Kotlin

**4.Sử dụng công cụ Android Studio**

-Cài đặt Android Studio

<https://developer.android.com/studio>

-Cài đặt bộ giả lập (Emulator)

-Tạo mới, mở, chạy…project

-Tạo package, thêm file, đổi tên file, xóa file…trong project

-Đồng bộ Gradle file với project

-…

**5.Chạy ứng dụng trên thiết bị thật**

[https://developer.android.com/codelabs/basic-android-kotlin-training-run-on-mobile-device?continue=https%3A%2F%2Fdeveloper.android.com%2Fcourses%2Fpathways%2Fandroid-development-with-kotlin-4%23codelab-https%3A%2F%2Fdeveloper.android.com%2Fcodelabs%2Fbasic-android-kotlin-training-run-on-mobile-device#0](https://developer.android.com/codelabs/basic-android-kotlin-training-run-on-mobile-device?continue=https%3A%2F%2Fdeveloper.android.com%2Fcourses%2Fpathways%2Fandroid-development-with-kotlin-4%23codelab-https%3A%2F%2Fdeveloper.android.com%2Fcodelabs%2Fbasic-android-kotlin-training-run-on-mobile-device" \l "0)

**6.Phân biệt các phiên bản Minimum Sdk, Target Sdk, Compile Sdk**

-Phiên bản tối thiểu (Minimum Sdk): phiên bản thấp nhất của API mà ứng dụng có thể chạy

-Phiên bản mục tiêu (Target Sdk): phiên bản mong muốn để chạy

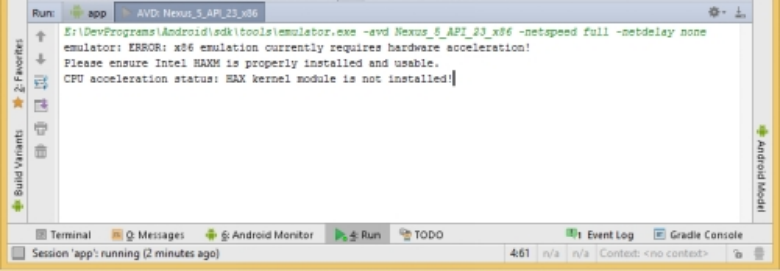
-Phiên bản biên dịch (Compile Sdk):

+Phiên bản của Android IDE của bạn (hoặc trình biên dịch nào đó) sử dụng để biên dịch và xây dựng ứng dụng để phát hành, chạy, hoặc gỡ lỗi.

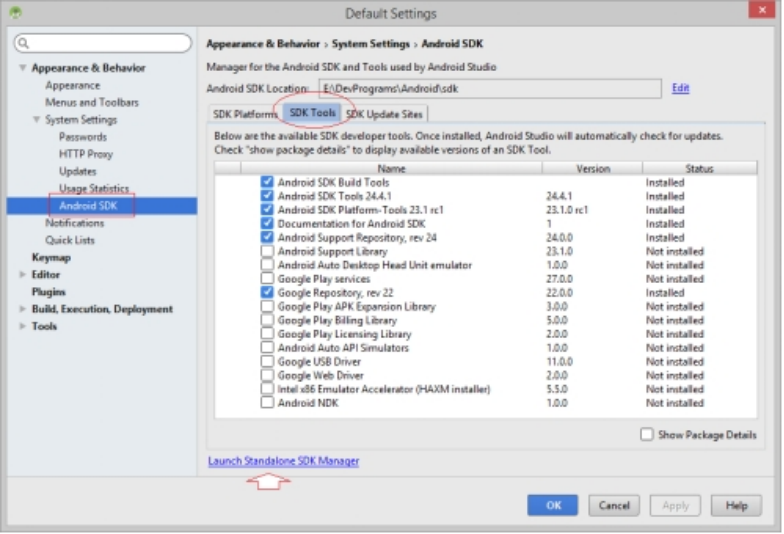
+Phiên bản này dùng để dịch ứng dụng thành tập tin .apk

**7.Một số lỗi khi chạy Android App**

-Cài đặt Intel Emulator Accelerator:



Trên Android Studio chọn: **Tools / SDK Manager** -> Chọn **SDK Tools** -> Check vào **Intel Emulator Accelerator** -> Kích **OK**.



-Tùy chỉnh Editor:

Chọn **File** -> **Settings** -> **Editor**

-Phóng to font chữ android studio:

Chọn **File** -> **Settings** -> **Editor** -> **Font** -> Đặt lại size.

-Khi chạy thấy hiển thị máy ảo nhưng không hiển thị chương trình:

Mở **AVD Manager** -> chọn mũi tên xổ xuống ở phần **Action** -> Chọn **wipe data** để xóa các dữ liệu trước đó.

-Sửa các phiên bản Minimum Sdk, Target Sdk, Compile Sdk: Sửa trong file Gradle

**8.Cấu trúc thư mục của Android app**

-Tập tin Manifests -> Tập tin chứa các thông tin về App

-Thư mục Java -> Mã nguồn (mã nguồn để chạy, test script)

-Thư mục Gradle -> Dịch

-Thư mục Res -> Tài nguyên không phải mã nguồn

+colors: màu sắc

*<?*xml version="1.0" encoding="utf-8"*?>*<resources>  
 <color name="colorPrimary">#008577</color>  
 <color name="colorPrimaryDark">#00574B</color>  
 <color name="colorAccent">#D81B60</color>  
  
 <color name="colorPrimary1">#FF8A80</color>  
 <color name="colorPrimaryDark1">#FF80AB</color>  
 <color name="myWindowBackground">#82B1FF</color>  
 <color name="colorAccent1">#00C853</color>  
 <color name="textColorPrimary1">#C51162</color>  
 <color name="navigationBarColor1">#0091EA</color>  
 <color name="textColorSecondary1">#B3E5FC</color>  
  
 <color name="purple\_200">#FFBB86FC</color>  
 <color name="purple\_500">#FF6200EE</color>  
 <color name="purple\_700">#FF3700B3</color>  
 <color name="teal\_200">#FF03DAC5</color>  
 <color name="teal\_700">#FF018786</color>  
 <color name="black">#FF000000</color>  
 <color name="white">#FFFFFFFF</color>  
</resources>

+strings: các chuỗi

+Có thể thêm các file khác như Theme, Style (.xml)

Theme:

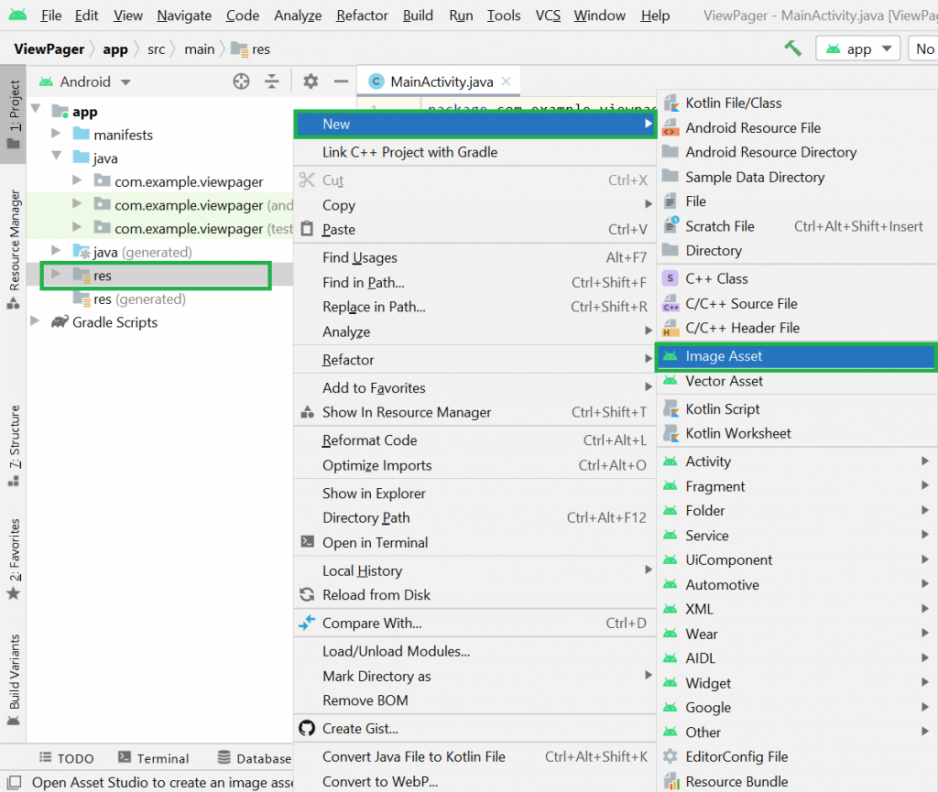
*<?*xml version="1.0" encoding="utf-8"*?>*<resources>  
 <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">  
 *<!--Màu chính (ví dụ nền AppBar)-->* <item name="colorPrimary">@color/colorPrimary</item>  
 *<!--Màu nền thanh trạng thái điện thoại-->* <item name="colorPrimaryDark">@color/colorPrimaryDark</item>  
 *<!--Màu nhấn mạnh (kiểm trong checkbox, gạch chân TextBox ...)-->* <item name="colorAccent">@color/colorAccent</item>  
 </style>  
  
 <style name="MyAppTheme" parent="Theme.AppCompat.Light.DarkActionBar">  
 <item name="colorPrimary">@color/colorPrimary1</item>  
 *<!--Màu chính (ví dụ nền AppBar)-->* <item name="colorPrimaryDark">@color/colorPrimaryDark1</item>  
 *<!--Màu nền thanh trạng thái điện thoại-->* <item name="android:navigationBarColor">@color/navigationBarColor1</item>  
 *<!--Màu nền điều hướng điện thoại-->* <item name="colorAccent">@color/colorAccent1</item>  
 *<!--Màu nhấn mạnh (kiểm trong checkbox, gạch chân TextBox ...)-->* <item name="android:windowBackground">@color/myWindowBackground</item>  
 *<!--Màu nền Activity-->* <item name="android:textColorPrimary">@color/textColorPrimary1</item>  
 *<!--Màu chữ trong các control như Button ... -->* <item name="android:textColorSecondary">@color/textColorSecondary1</item>  
 *<!--Màu chữ ở trạng thái ... -->* </style>  
</resources>

Style:

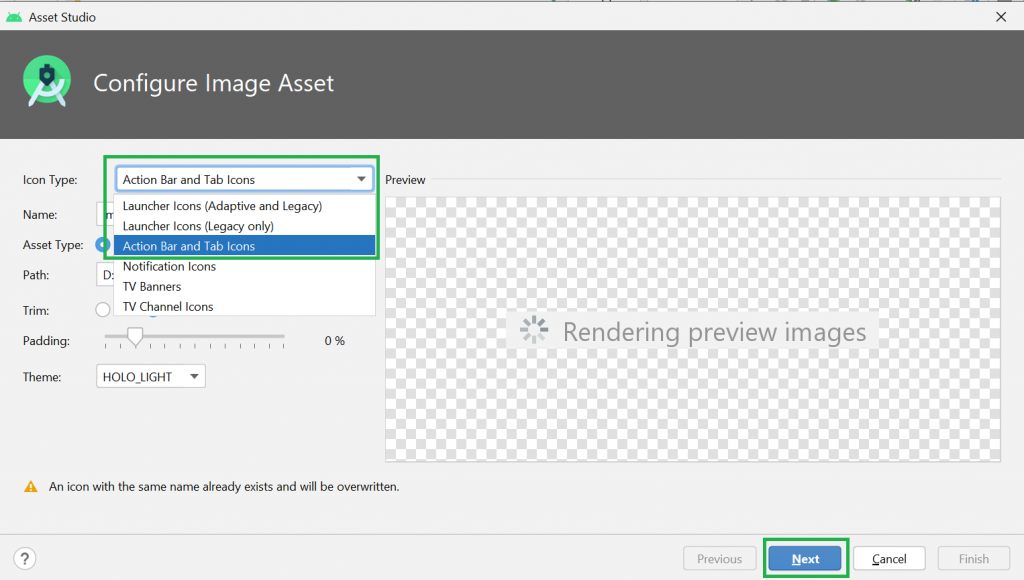
*<?*xml version="1.0" encoding="utf-8"*?>*<resources>  
 <style name="InformationTextView">  
 <item name="android:layout\_width">match\_parent</item>  
 <item name="android:layout\_height">wrap\_content</item>  
 <item name="android:layout\_margin">8dp</item>  
 <item name="android:gravity">center</item>  
 <item name="android:textSize">15sp</item>  
 <item name="android:textColor">#00FF00</item>  
 </style>  
</resources>

+Thêm ảnh

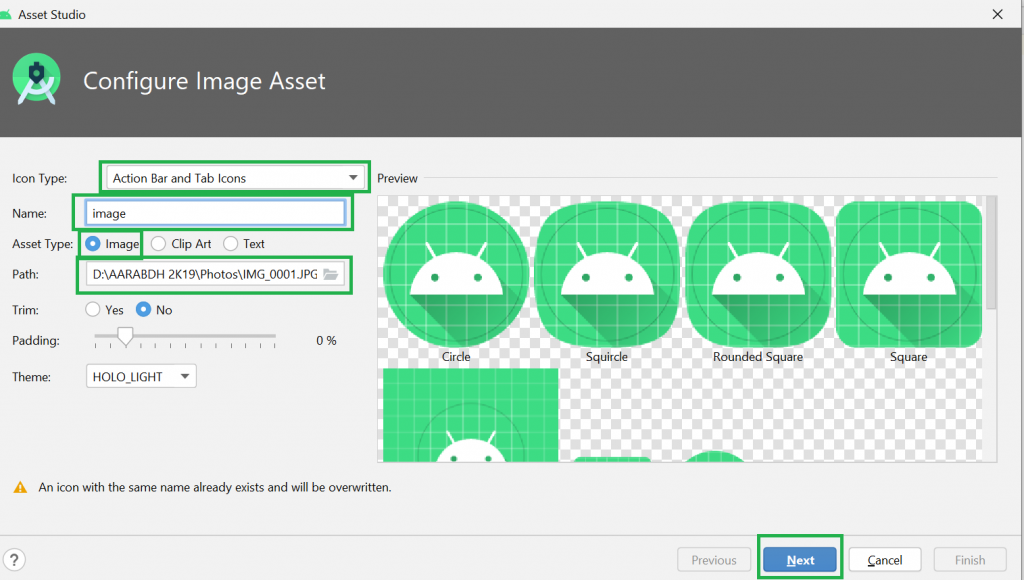
Bước 1: Mở Android Studio và vào app> res> right click> New> Image Asset

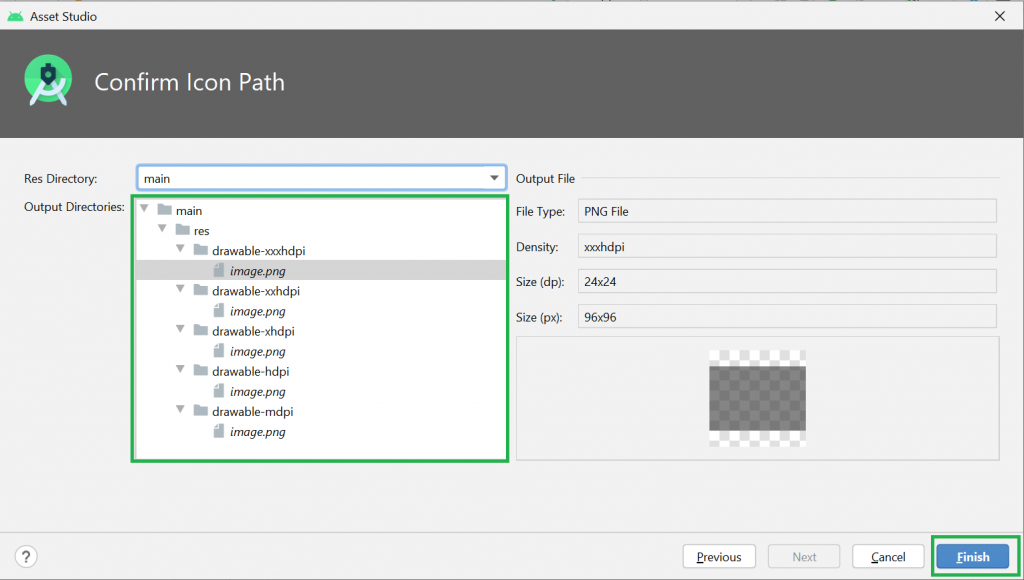


Bước 2: Một màn hình bật lên sẽ xuất hiện như bên dưới. Ở đây chọn Thanh tác vụ và Biểu tượng tab trong Loại biểu tượng.



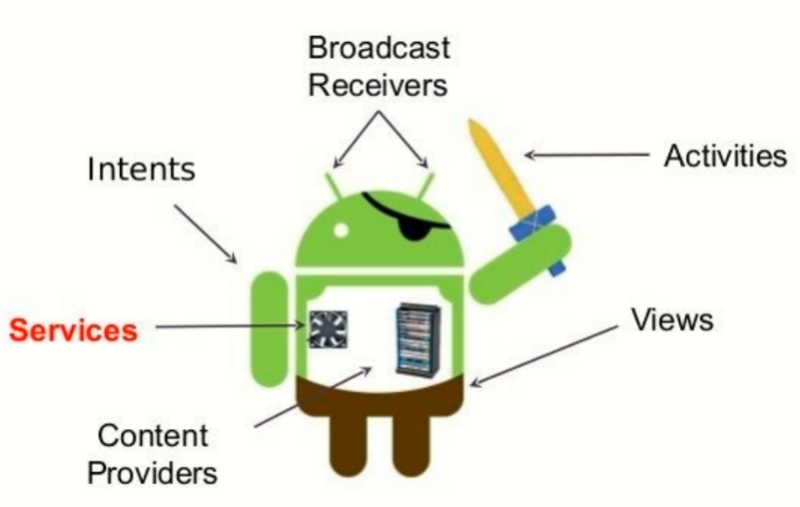
Bước 3: Sau đó chọn Loại nội dung là Hình ảnh và nhập Đường dẫn của hình ảnh của bạn





Kích Finish để hoàn thành.

**9.Các component chính của Android App**



**10.Activity**

-Thêm/sửa/xóa 1 Activity

-Cài đặt để 1 Activity chạy đầu tiên:

<activity  
 android:name=".MainActivity"  
 android:exported="true">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
</activity>

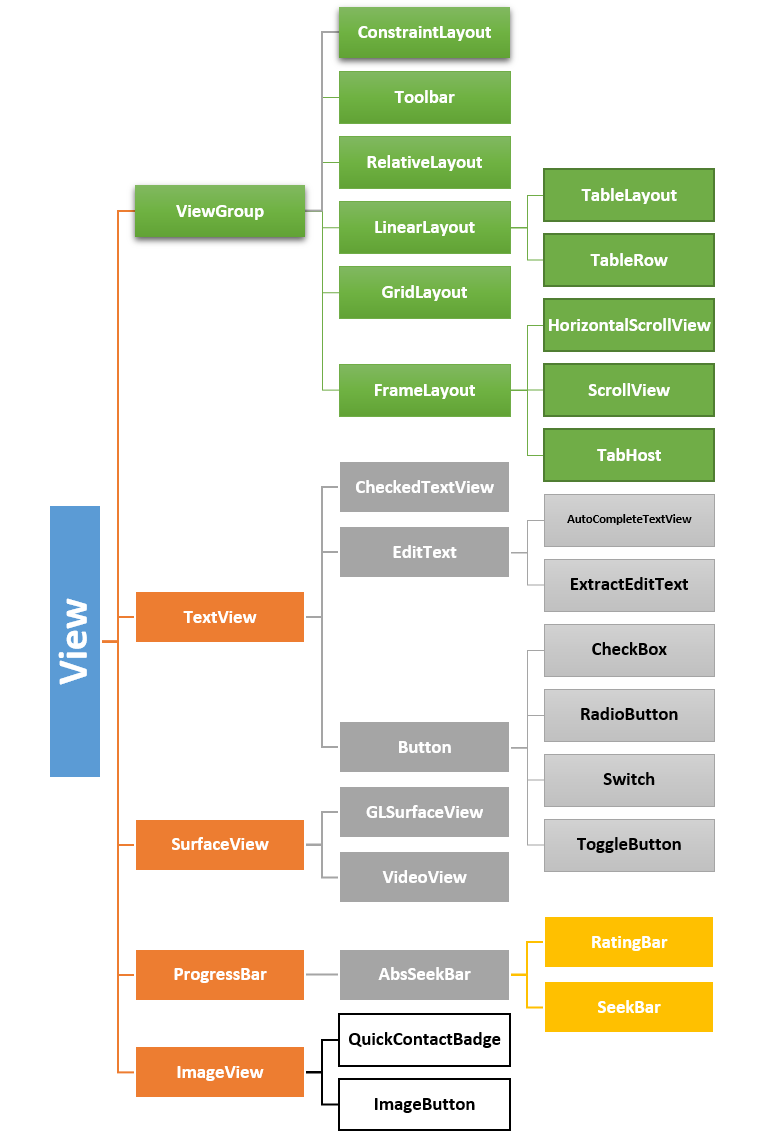
-Chạy 1 Activity nào đó của ứng dụng:

+Chỉnh thông tin về Activity trong file Manifest: exported="true"

<activity  
 android:name=".MainActivity2"  
 android:exported="true" />

+Kích chuột phải vào Activity cần chạy và chọn Run

-Sử dụng View để thiết kế giao diện Activity



-Sử dụng Layout để thiết kế giao diện Activity

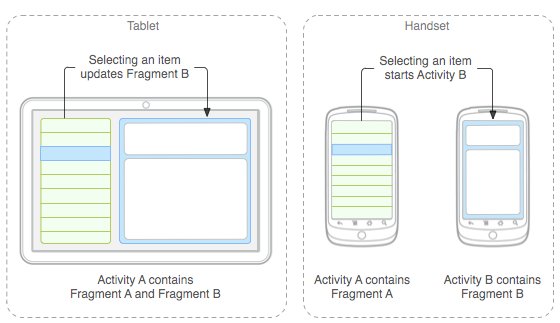
* FrameLayout: đơn giản chỉ cung cấp một vùng màn hình, thường dùng nó để hiện thị một View con duy nhất. Nếu đặt vào nó nhiều view con, thì mặc định các view con này sẽ xếp chồng lên nhau.
* ConstraintLayout (giới thiệu trong Android 7): sử dụng layout này được khuyến khích cho hầu hết trường hợp. ConstraintLayout cho phép điều khiển vị trí và ứng sử của các view con trong layout bằng cách gán dàng buộc đơn giản vảo mỗi view con. Từ đó mà một bố cục phức tạp có thể dễ dàng được tạo ra mà sử dụng ít nhất sự lồng nhau trong layout (layout này nằm trong layout khác) giúp cho cải thiện tốc độ. ConstraintLayout cũng tích hợp sẵn vào Android Studio Layout Editor nên có thể điều chỉnh một cách trực quan các View con trong layout này.
* LinearLayout: các view con được xếp nối tiếp nhau (linear) thành một hàng hay một cột (tùy vào lúc thiết kế thiết lập hướng xếp). Có một giá trị weight có thể gán vào mỗi View con để cho biết View con đó chiếm bao nhiêu không gian trong một tỷ lệ tương quan với các View con khác.
* RelativeLayout: cho phép các view con định vị căn vào liên hệ với các view con khác đồng thời liên hệ với view cha thông qua các tham số align và margin.
* GridLayout: chia ra thành lưới gồm một số hàng và một số cột để chứa các view con.
* TableLayout: cung cấp khả năng bố trí các view con thành một lưới dạng bảng (gồm có hàng và cột). Một dòng của bảng biểu diễn bàng đối tượng view con TableRow, trong nó chứa có phần tử View con hiểu như các ô bảng.

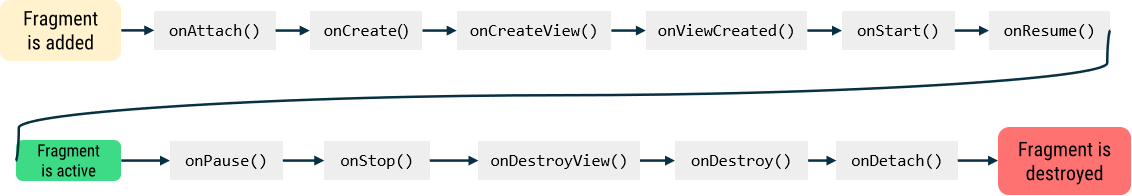
-Sử dụng OnClickListener để viết sự kiện cho View

-Quản lý vòng đời của Activity



**11.Fragment**





|  |  |  |
| --- | --- | --- |
| **Tiêu chí** | **Activity** | **Fragment** |
| **Vị trí đối với ứng dụng** | Activity là hoạt động, cửa số chính, tồn tại độc lập | Fragment là một phần của Activity. Đóng góp UI và hoạt động của nó vào thành phần chính |
| **Vị trí tương đối với nhau** | Activity có thể chứa nhiều Fragment | Fragment là một phần của Activity |
| **Tái hoạt động** | Không thể tái hoạt động | Một fragment có thể được tái sử dụng trong một activity,do đó nó hoạt động như một thành phần tái sử dụng trong các hoạt động. |
| **Vòng đời** | C:\Users\ADMIN\AppData\Local\Temp\ksohtml9276\wps1.png | C:\Users\ADMIN\AppData\Local\Temp\ksohtml9276\wps2.jpg |
| **Tư tưởng hình thành** | Các hoạt động là một trong những khối xây dựng cơ bản của ứng dụng trên nền tảng Android. Chúng đóng vai trò là điểm vào cho sự tương tác của người dùng với một ứng dụng và cũng là trung tâm cho cách người dùng điều hướng trong ứng dụng hoặc giữa các ứng dụng | Fragment đại diện cho một hành vi hoặc một phần của giao diện người dùng trong một hoạt động. Bạn có thể kết hợp nhiều fragment trong một hoạt động để xây dựng giao diện đa cửa sổ và sử dụng lại một đoạn trong nhiều hoạt động. Bạn có thể nghĩ ra một đoạn như một phần mô đun của một hoạt động, có vòng đời riêng của nó, nhận các sự kiện đầu vào của chính nó, và bạn có thể thêm hoặc xoá trong khi hoạt động đang chạy. |
| Tổ chức vòng đời | Các phương pháp vòng đời được tổ chức bởi hệ điều hành (OS). | Phương pháp vòng đời được tổ chức bởi tổ chức bởi hoạt động lưu trữ (hosting activity) |

**12.Quản lý vòng đời (lifecycle) của Activity hoặc Fragment**

-Override các phương thức cho vòng đời. Sử dụng Log hoặc Timber để theo dõi vòng đời trong cửa sổ Logcat:

Log.v(TAG, "some verbose logs here");

Timber.v("some verbose logs here");

Log.d(TAG, "some debug logs here");

Timber.d("some debug logs here");

Log.i(TAG, "some info logs here");

Timber.i("some info logs here");

Log.w(TAG, "some warning logs here");

Timber.w("some warning logs here");

Log.e(TAG, "some error logs here");

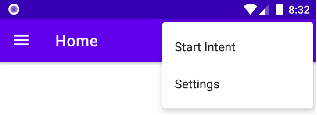
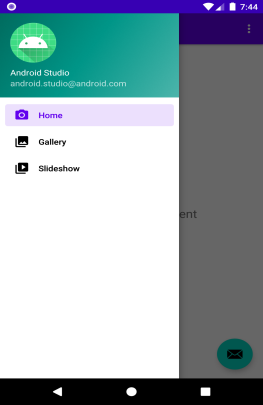
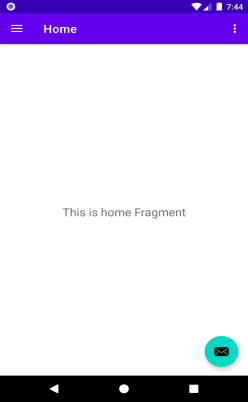
Timber.e("some error logs here");

Log.wtf(TAG, "some error logs here");

Timber.wtf("some error logs here");

-Sử dụng [thư viện lifecycle](https://developer.android.com/reference/androidx/lifecycle/package-summary) (thư viện thuộc gói Jetpack) để quản lý vòng đời

**13. Các cách điều hướng trong ứng dụng**

-Tạo App bar, navigation drawer, and menus

-Sử dụng intent để từ 1 Activity gọi 1 Activity khác, truyền các tham số khi gọi

Intent intent = new Intent();

//Gán context và tên lớp Activity cần chạy

intent.setClass(view.getContext(), UpdateActivity.class);

//Nếu có truyền tham số cho Activity

intent.putExtra("dulieu1", 345);

intent.putExtra("dulieu2", "Một chuỗi ...");

//Gửi Intent cho hệ thống Android để kích hoạt Activity

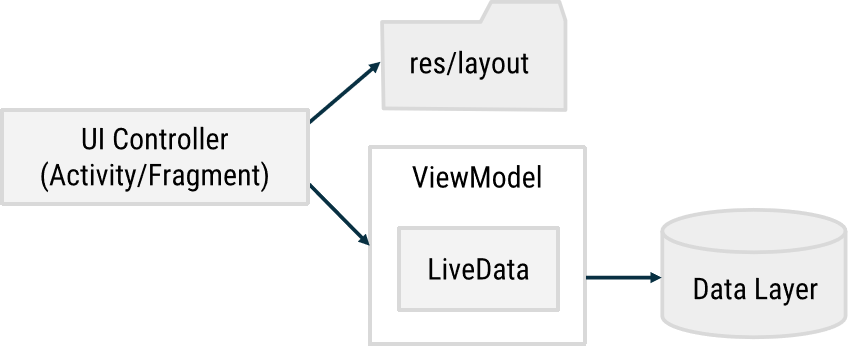
startActivity(intent);

//Muốn Activity thứ nhất kết thúc thì thêm **finish();**

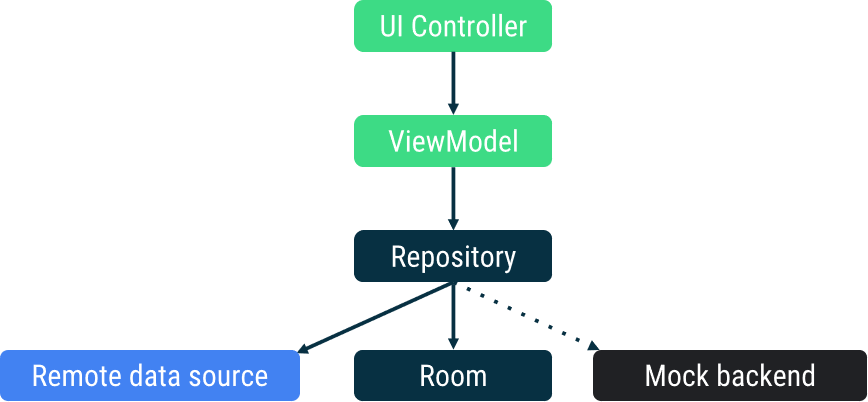
-Tạo Fragment và sử dụng navigation component điều hướng giữa các Fragment, sử dụng thư viện Safe Args để truyền tham số giữa các Fragment

**14. Kiến trúc của ứng dụng**

-Không sử dụng Repository: MVVM (Model View ViewModel)



-Sử dụng Repository:



Retrofit, Repository: <https://www.youtube.com/watch?v=D29vhvGv9Cc>

-Các khái niệm:

+ViewModel

+Data binding

+LiveData

+Repository

**15.Làm việc với cơ sở dữ liệu cục bộ sqlite**

-Kết nối và thao tác với với cơ sở dữ liệu cục bộ sqlite

<https://www.geeksforgeeks.org/android-sqlite-database-in-kotlin/>

<https://www.javatpoint.com/kotlin-android-sqlite-tutorial>

-Kết nối và thao tác với với cơ sở dữ liệu cục bộ sqlite sử dụng Room database

<https://rrtutors.com/tutorials/Kotlin-Room-Database-example-with-ViewModel-and-Live-Data>

**16.Sử dụng RecyclerView**

<https://developer.android.com/courses/pathways/android-development-with-kotlin-10>

-Các thành phần của RecyclerView:

+LayoutManager: Sắp xếp các item trong RecyclerView: ngang, dọc, grid (số dòng, số cột), so le

LinenarLayoutManager: Hỗ trợ scroll các item theo chiều ngang hay chiều dọc.

GridLayoutManager: Layout các item trong RecyclerView dưới dạng Grid giống như khi chúng ta sử dụng GridView.

StaggerdGridLayoutManager: Layout các item trong ListView dưới dạng Grid so le.

+Adapter: Lấy dữ liệu từ dữ liệu nguồn (database, array, list…) và trình bày cho người dùng trên RecyclerView theo 1 LayoutManager nào đó

+ViewHolder: mô tả Item View cùng với vị trí của Item đó trong RecyclerView.

+ItemAnimator: hoạt hình, chuyển tiếp khi add hay remove 1 item ra khỏi RecyclerView

-Set sự kiện khi kích 1 row

**-**Tạo RecyclerView có nhiều Item khác nhau

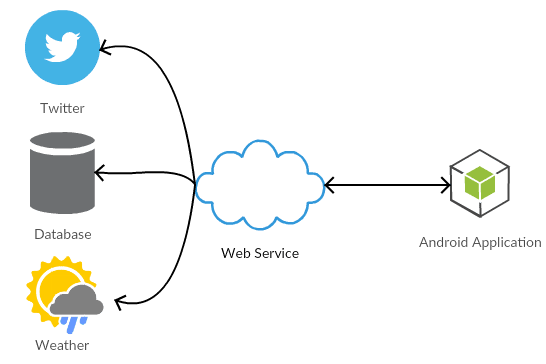
**17.Lập trình bất đồng bộ**

-Các hoạt động chạy lâu thì thay vì chạy trên main thread thì có thể được lập trình để chạy trên các background thread

-Sử dụng thư viện coroutine để lập trình bất đồng bộ

<https://tech.miichisoft.net/kotlin-coroutine/>

**18.Làm việc với internet**



-Lấy dữ liệu:

+Tạo Web service API hoặc sử dụng các Web service API có sẵn

+Sử dụng Retrofit để làm việc với các Web service API

-Sử dụng Glide để load ảnh từ 1 Url

**19.WorkManager**

<https://www.youtube.com/watch?v=sPPQ8lTsdhM>

-Quản lý các background work: dùng AsyncTask / Service / WorkManager

-Các bước sử dụng WorkManager

+Bước 1: Khai báo thư viện:

implementation "androidx.work:work-runtime-ktx:$work\_version"+Bước 2: Tạo class Worker, override phương thức doWork để cho biết công việc cần thực hiện ở background

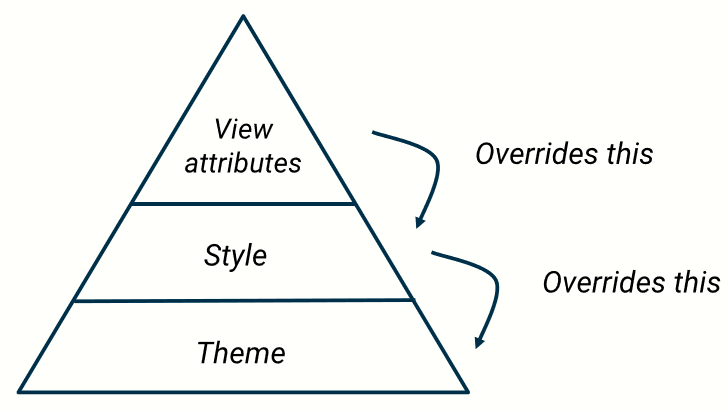
class UploadWorker(appContext: Context, workerParams: WorkerParameters) : Worker(appContext, workerParams) { override fun doWork(): Result { // Do the work here. In this case, upload the images. uploadImages() // Indicate whether work finished successfully with the Result return Result.success() }}+Bước 3: Khai báo WorkRequest để điều khiển việc chạy Worker:

* Chạy 1 lần
* Chạy lặp lại
* Có truyền tham số và nhận kết quả
* Có các ràng buộc

+Bước 4: Thêm WorkRequest vào hàng đợi của WorkManager

+Có thể sử dụng lệnh Log và cửa sổ Logcat để theo dõi việc thực hiện các background work

**20.Theme, Style, View attribute**

****

-Tạo các Theme và sử dụng cho các Activity

+Điều chỉnh file color:

*<?*xml version="1.0" encoding="utf-8"*?>*<resources>  
 <color name="colorPrimary">#008577</color>  
 <color name="colorPrimaryDark">#00574B</color>  
 <color name="colorAccent">#D81B60</color>  
  
 <color name="colorPrimary1">#FF8A80</color>  
 <color name="colorPrimaryDark1">#FF80AB</color>  
 <color name="myWindowBackground">#82B1FF</color>  
 <color name="colorAccent1">#00C853</color>  
 <color name="textColorPrimary1">#C51162</color>  
 <color name="navigationBarColor1">#0091EA</color>  
 <color name="textColorSecondary1">#B3E5FC</color>  
  
 <color name="purple\_200">#FFBB86FC</color>  
 <color name="purple\_500">#FF6200EE</color>  
 <color name="purple\_700">#FF3700B3</color>  
 <color name="teal\_200">#FF03DAC5</color>  
 <color name="teal\_700">#FF018786</color>  
 <color name="black">#FF000000</color>  
 <color name="white">#FFFFFFFF</color>  
</resources>

+Tạo các Theme: file .xml trong res/value

<resources xmlns:tools="http://schemas.android.com/tools">  
 <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">  
 *<!--Màu chính (ví dụ nền AppBar)-->* <item name="colorPrimary">@color/colorPrimary</item>  
 *<!--Màu nền thanh trạng thái điện thoại-->* <item name="colorPrimaryDark">@color/colorPrimaryDark</item>  
 *<!--Màu nhấn mạnh (kiểm trong checkbox, gạch chân TextBox ...)-->* <item name="colorAccent">@color/colorAccent</item>  
 </style>  
  
 <style name="MyAppTheme" parent="Theme.AppCompat.Light.DarkActionBar">  
 <item name="colorPrimary">@color/colorPrimary1</item>  
 *<!--Màu chính (ví dụ nền AppBar)-->* <item name="colorPrimaryDark">@color/colorPrimaryDark1</item>  
 *<!--Màu nền thanh trạng thái điện thoại-->* <item name="android:navigationBarColor">@color/navigationBarColor1</item>  
 *<!--Màu nền điều hướng điện thoại-->* <item name="colorAccent">@color/colorAccent1</item>  
 *<!--Màu nhấn mạnh (kiểm trong checkbox, gạch chân TextBox ...)-->* <item name="android:windowBackground">@color/myWindowBackground</item>   
 *<!--Màu nền Activity-->* <item name="android:textColorPrimary">@color/textColorPrimary1</item>   
 *<!--Màu chữ trong các control như Button ... -->* <item name="android:textColorSecondary">@color/textColorSecondary1</item>  
 *<!--Màu chữ ở trạng thái ... -->* </style>  
</resources>

+Áp dụng Theme: gán thuộc tính theme của Activity trong manifests bằng tên của Theme này

*<?*xml version="1.0" encoding="utf-8"*?>*<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 package="com.example.themestyleapp">  
  
 <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/MyAppTheme">  
 <activity  
 android:name=".MainActivity"  
 android:exported="true">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
 </activity>  
 </application>  
  
</manifest>

-Tạo các Style và sử dụng cho các View

+Tạo Style: file .xml trong res/value

*<?*xml version="1.0" encoding="utf-8"*?>*<resources>  
 <style name="InformationTextView">  
 <item name="android:layout\_width">match\_parent</item>  
 <item name="android:layout\_height">wrap\_content</item>  
 <item name="android:layout\_margin">8dp</item>  
 <item name="android:gravity">center</item>  
 <item name="android:textSize">15sp</item>  
 <item name="android:textColor">#00FF00</item>  
 </style>  
</resources>

+Áp dụng Style cho View nào đó:

<TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Hello World!"  
 style="@style/InformationTextView"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />

+Tạo Style kế thừa từ các Style khác:

<style name="BaseTextViewStyle">

<item name="android:layout\_width">wrap\_content</item>

<item name="android:layout\_height">wrap\_content</item>

<item name="android:textColor">@color/cl\_default</item>

<item name="android:textSize">@dimen/text\_size\_normal</item>

</style>

<style name="LargeTextViewStyle" parent=" BaseTextViewStyle ">

<item name="android:textSize">@dimen/text\_size\_large</item>

</style>

<style name="SmallTextViewStyle" parent=" BaseTextViewStyle ">

<item name="android:textSize">@dimen/text\_size\_small</item>

</style>

-Khai báo các View attribute