**ANDROID APPLICATION**

**ASSIGNMENT WEEK 3**

1.Discuss About different types of  Android Architecture.

[Android architecture is a multi-layered system composed of several components, including the operating system, middleware, and key applications](https://trainings.internshala.com/blog/android-architecture/). The main components of Android architecture are as follows:

1. [**Linux Kernel**: It is the foundation of the Android platform and provides the main functionality of the operating system](https://developer.android.com/guide/platform/).
2. [**Hardware Abstraction Layer (HAL)**: It provides standard interfaces that expose device hardware capabilities to the higher-level Java API framework](https://developer.android.com/guide/platform/).
3. **Android Runtime**: It consists of core libraries and the Dalvik Virtual Machine (DVM) or Android Runtime Environment (ART). [The runtime environment powers Android applications with the help of core libraries](https://trainings.internshala.com/blog/android-architecture/).
4. **Application Framework**: It provides important classes and services for creating an Android application. [It includes various types of services such as activity manager, notification manager, view system, package manager, etc.](https://trainings.internshala.com/blog/android-architecture/).
5. [**Applications**: This is the top layer of the Android architecture and includes pre-installed applications like home, contacts, camera, gallery, as well as third-party applications downloaded from the Play Store](https://trainings.internshala.com/blog/android-architecture/).

In addition to these main components, there are different types of architecture patterns followed in Android development. Some popular architecture patterns are:

* [**MVC (Model-View-Controller)**: This pattern separates an application into three interconnected components: model, view, and controller](https://trainings.internshala.com/blog/android-architecture/).
* [**MVP (Model-View-Presenter)**: This pattern separates an application into three components: model, view, and presenter](https://trainings.internshala.com/blog/android-architecture/).
* [**MVVM (Model-View-ViewModel)**: This pattern separates an application into three components: model, view, and view model](https://trainings.internshala.com/blog/android-architecture/).

[These architecture patterns help structure the project’s code and give it a modular design by separating concerns](https://trainings.internshala.com/blog/android-architecture/).

2.  Differentiate Implicit and Explicit Intent.

In Android, **intents** are messaging objects that specify the type of action to be performed. [They are used for launching activities and passing data between components1](https://www.geeksforgeeks.org/difference-between-implicit-intent-and-explicit-intent-in-android/). [There are two types of intents: **implicit** and **explicit**1](https://www.geeksforgeeks.org/difference-between-implicit-intent-and-explicit-intent-in-android/).

* **Implicit Intent**: An implicit intent does not specify the component to be invoked. [Instead, it provides information about the available components provided by the system that can handle the intent](https://www.geeksforgeeks.org/difference-between-implicit-intent-and-explicit-intent-in-android/)[1](https://www.geeksforgeeks.org/difference-between-implicit-intent-and-explicit-intent-in-android/). For example, you can use an implicit intent to view a webpage by writing code like this:

Intent intent = new Intent(Intent.ACTION\_VIEW);

intent.setData(Uri.parse("https://www.example.com"));

startActivity(intent);

In this case, the intent does not specify a particular component but instead declares an action to be performed (viewing a webpage). [The system then determines which component can handle the intent and opens the webpage](https://www.geeksforgeeks.org/difference-between-implicit-intent-and-explicit-intent-in-android/)[1](https://www.geeksforgeeks.org/difference-between-implicit-intent-and-explicit-intent-in-android/).

* **Explicit Intent**: In contrast, an explicit intent specifies the component to be invoked. [It provides the external class that should be called](https://www.geeksforgeeks.org/difference-between-implicit-intent-and-explicit-intent-in-android/)[1](https://www.geeksforgeeks.org/difference-between-implicit-intent-and-explicit-intent-in-android/). For example, you can use an explicit intent to navigate from one activity to another like this:

Intent intent = new Intent(getApplicationContext(), SecondActivity.class);

startActivity(intent);

[In this case, the intent explicitly names the target activity (SecondActivity), and the system launches that specific activity](https://www.geeksforgeeks.org/difference-between-implicit-intent-and-explicit-intent-in-android/)[1](https://www.geeksforgeeks.org/difference-between-implicit-intent-and-explicit-intent-in-android/).

To summarize:

* **Implicit Intent**: Does not specify the component and allows the system to determine which component can handle the intent.
* **Explicit Intent**: Specifies the component to be invoked.

3. Create an android application which allows you to navigate to two different websites from its home page. Implement navigation to first website through implicit intent and second using explicit intent.

Here’s an example of an Android application that allows you to navigate to two different websites from its home page using implicit and explicit intents.

First, you need to create a new Android project in Android Studio. Then, follow these steps:

1. Open the activity\_main.xml layout file and add two buttons to the layout. You can use the following code as a starting point:

<LinearLayout 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"

android:gravity="center"

tools:context=".MainActivity">

<Button

android:id="@+id/buttonWebsite1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Website 1" />

<Button

android:id="@+id/buttonWebsite2"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Website 2" />

</LinearLayout>

1. Open the MainActivity.java file and add the following code inside the onCreate method:

Button buttonWebsite1 = findViewById(R.id.buttonWebsite1);

Button buttonWebsite2 = findViewById(R.id.buttonWebsite2);

buttonWebsite1.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent intent = new Intent(Intent.ACTION\_VIEW);

intent.setData(Uri.parse("https://www.example.com"));

startActivity(intent);

}

});

buttonWebsite2.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent intent = new Intent(getApplicationContext(), SecondActivity.class);

startActivity(intent);

}

});

1. Create a new Java class called SecondActivity.java and add the following code:

public class SecondActivity extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_second);

// Add your code for navigating to the second website here

}

}

1. Open the AndroidManifest.xml file and add the following code inside the <application> tag:

<activity android:name=".SecondActivity" />

That’s it! You have created an Android application with two buttons on the home page that allow you to navigate to two different websites. The first button uses an implicit intent to open a web browser and load the first website, while the second button uses an explicit intent to navigate to the SecondActivity class.