



mobile Application Development.

Q:- I

Based on your understanding identify recent business trend that has influenced the Android Platform. Explain how this trend impacts Android APP development and business in the mobile APP industry.

- one recent business trend that has influenced the Android Platform is the rise of mobile-commerce (m-commerce)
- The growth of m-commerce has led to an increase in demand mobile APP.

Ex Android APP development.

- Increased demand for m-commerce (UPPER TUMHAR)
- Android APP developers now have the opportunity to develop app for a wide range of m-commerce activity
- This has led to an increase demand for Android APP development skills and expertise

- new revenue opportunities
- The growth of m-commerce has created new revenue opportunities for android APP developers.

- for example developers can earn revenue through purchase, advertising and subscription.

- need for new skill and expertise
- Android app developers who want to develop m-commerce app need to have a good understanding e-commerce app and payment processing
- They also need to be able to develop secure and reliable app that can handle large volume of transactions.

⇒ Business in the mobile APP industry

- Increased competition
 - The growth of m-commerce has led to an increase in competition in the mobile app industry.
 - Businesses need to develop high quality and innovative m-commerce app in order to attract potential customers.
- need to invest in mobile APP development
 - Businesses need to invest in mobile app development in order to offer their users seamless experience.
- new opportunities for growth
- The growth of m-commerce has created

Assignment - I

MEHSANA DISTRICT EDUCATION FOUNDATION SANCHALIT



U.V. Patel
College of
Engineering

MEHSANA DISTRICT EDUCATION FOUNDATION SANCHALIT
U. V. Patel College of Engineering
GANPAT UNIVERSITY, KHERVA - 384 012 DIST. MEHSANA. (N.G.)

for business in the mobil. app
industry.

Q 1-3

What is the Purpose of XML
or LayoutInflater in Android development
and how does it fit into the
architecture of Android apps?

Purpose of Layout Inflater

1) Dynamic UI Creation

- Inflater enables dynamic UI creation
by allowing developer to create view
programmatically at runtime based
on predefined XML layout

2) Reuse of Layout Components

- Inflaters facilitate the reuse of
layout components instead of duplicating
the same layout definition multiple
times within code. You can define
it once and use it in XML inflated whenever

3) Separation of concerns

→ In Android UI components are
typically defined between UI
and business logic

- 1) How it fits into Android layout architecture
- 2) XML layout definition
 - Developers define the structure and appearance of UI components using XML files in the "res/layout" directory.
- 3) Activity / Fragment Initialization
 - In the "onCreate" method of an Activity or fragment the layout ID used to inflate the layout.
- 4) Inflating layout
 - The "LayoutInflater" class is employed to inflate XML layout containing the views.
 - ex:- ViewInflater = LayoutInflater.from(context)
- 5) Accessing views
 - Views within the inflated layout can be accessed programmatically.
 - ex:- View myTextView = findViewById(R.id.TextViewID)
- 6) Setting content view
 - The inflated view hierarchy is set as the content view of the Activity.
 - ⇒ ex:- setContentView(R.layout.main)

3)

Explain the concept of a ^{tooBox} custom dialog in android applications example to illustrate its use.

Ans: A "custom dialog" in Android is a pop-up window that developers can design and customize to suit the specific need and branding of their application.

- It's a way to present information to the user or confirm action in a visually customized manner.

→ The Android SDK provides the "Dialog" class and developers often extend it or use the "AlertDialog" class to create custom dialogs.

Ex:- MainActivity.kt.

```
import android.os.Bundle
import androidx.appcompat.app.AppCompatActivity
import android.widget.Button
import com.example.mymain.CustomDialog
import com.example.myapp.R
```

class MainActivity : AppCompatActivity

override fun onCreate(savedInstanceState: Bundle?) {
 super.onCreate(savedInstanceState)
 setContentView(R.layout.activity_main)

Q1-4) How do Activities Service and the Android manifest file work together to make an Android APP? Can you design simpler manifest and provide a brief example of how can we design a ~~Android~~ mobile App.

Ans:- Activities

Activities are the user interface components of an Android. They represent individual screen with which user can interact. Each activity is a self-contained unit with its own UI layout.

2) Services: Platform background few without user interface, like such as picdm music delivery server long runs operations.



3)

AndroidManifest file

It is a configuration file that provides essential info about the app to a system. It declares the app's compiler properties and permission requirements.

→ Example of Activity

class MainActivity : AppCompatActivity

{
 onCreate(savedInstanceState) {
 super.onCreate(savedInstanceState);
 setContentView(R.layout.activity_main);
 }
}

val playButton : Button = findViewById(R.id.playbutton)
 playButton.setOnClickListener {
 service.startService(Intent(this, MusicService::class.java))
 }
}

val intent = Intent(this, MusicService::class.java)
 startService(intent)
}

→ Example of service

class MusicService : Service

private Intent & IntentFilter
Media Player Extends
GARIBA UNIVERSITY KHEERA - 790050 OUT PANCHALA

override fun onBind(Intent intent)

{
 extra null
}

override fun onStartCommand(intent)

Intent? flags: Int; startId:
: In

{ mediaPlayer.onCreate (this: View: String)

mediaPlayer.onStart()

return START_STICKY;

override fun onDestroy()

{ mediaPlayer.onDestroy()

super.onDestroy()

} }

#> AndroidManifest.xml

<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="com.ksumar.musicplayer"/>

<application

android:allowBackup="true"
android:icon="@mipmap/ic_launcher"/>

```

    android:label = "@string/up_name"
    android:icon = "@mipmap/ic_launcher"
    android:supportRt = "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>

<service android:name="myService" />
</application>
</manifest>

```

Ques) How does the Android manifest file impact the development of an application? Provide an example & demonstrate its significance.

Ans:- The android manifest file impacts the development

1) Declaration of APP Components

- The manifest file declares all the components of an Android APP, including activity, services, broadcast receivers, and content providers.

2) Setting main activity

- The manifest file specifies the main activity, which is the entry point of the app. This activity that is launched when the app is opened.

3) Permission and security

- The manifest file is used to declare the permission that the app requires.
- The manifest file is used to declare the permissions the app require to access certain device features / data.

4) Intent filters

- Intent filters in the manifest file define how the app responds to implicit intent actions and type app commands.

Ex:

AndroidManifest XML: android = "schemas.android.com/apk/res/android"

Package = "com.example.clima"

uses-permission android: name = "android.permission.CAMERA"

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

android:name = "CumulativeActivity"

activity

Application

</manifest>

Q) What is the role resources in Android development? Discuss the various types of resource and their significance in creating well-structured applications. Give example to clarify your points.

Ans.:- In Android development, resources play a crucial role in creating well-structured and user-friendly applications.

- Resources in Android are external elements such as Images, String, Layout, Columns, other assets that are separate from application code.
- They are used to provide flexibility, maintainability and support for various device configurations.

1) String Resource

Role :- String resources are used to store text strings that are displayed in the user interface.

- String resources make it easier to manage themes, colors and adapt to different screen sizes.

Ex:- res/values/strings.xml

```
<resources>
    <string name="app_name">My App</string>
    <string name="welcome_message">Welcome my App</string>
</resources>
```

2) Drawable Resource

- Drawable resource include images and graphics and used in the UI
- Different version images can be provided for different screen densities

Ex:- res/drawable/icon.png

```
android:layout_width="wrap_content"
android:layout_height="wrap_content"
src="@drawable/icon"/>
```

3) Color Resources

- It stores color values that can be easily reused across the app thus for consistent theming and makes it simple to update the color scheme.

<resources>

```

<color name="red"># FF 0000</color>
<color name="green"># 00FF00</color>
</resources>
  
```

7) How does an Android Service contribute to the functionality of a mobile app? Describe the process of developing an android service.

Ans: Contribution to Functionality

1) Background Processing

- Services are ideal for tasks that should continue runtime even when the app is not actively interacting with the user.
- Examples include playing music, fetching data from the internet.

2) Inter-Component Communication

- Services can communicate with other components such as activities or other services using Android Inter Process communication mechanism like Intent and Binder.

3) Long-Running Operations

- Services are suitable for executing long running operations.

- Services are suitable for executing long-running operation such file synchronization with service or continuous sensor monitoring.

4) Foreground Service

- They are Service type service that provides a persistent notification, ensuring ensuring that user is aware of ongoing task.

#) Process of Developing an Android Service

1) Create a Service Class

- Create a new class that extends the Service class. This class will contain the logic for your service.

```
class myservice : Service {
    override fun onBind(intent: Intent): IBinder? {
        return null
    }
}
```

```
override fun onStart(intent: Intent,
                     flags: Int, startId: Int) {
    return START_STICKY
}
```

onCreate() and onDestroy()

MEHSANA DISTRICT EDUCATION FOUNDATION SANCHARI
GANDHAT UNIVERSITY, KHERA - 384 015 DIST. MEHSANA. (N.G.)

U.A. Patel
Engineering
College of
Engineering



2. Declare the service in the Manifest
- declare your service in the AndroidManifest.xml file to let the Android system know about it.

```
<service android: name = "myService" />
```

3) Start your Service

You can start the service by creating an Intent and using 'startService()'.

Val intent = Intent(context, MyService::class.java)
context.startService(intent)

4) Service Life Cycle

- The Service life cycle method (onCreate(), onStartCommand(), onBind(), and onDestroy())

allow you to manage the behavior of your service at different points in the life.