

## Assignment 01

Q:01 Based on your understanding, identify a recent business trend that has influenced the Android platform. Explain how this trend impacts Android App developers and businesses in the mobile app industry?

→ Impact on Android app development and businesses:

i) Steady Revenue: Subscription-based models provide a ~~surviving~~ revenue stream for businesses, ensuring a more predictable income.

ii) Enhanced User Engagement: To justify ongoing subscriptions, businesses and developers are motivated to continually improve their apps, resulting in better features, updates, and user experiences on the Android platform.

iii) Challenges for User Acquisition: While subscriptions can provide stable revenue, the competitive landscape makes user acquisition more challenging.

Q:02 What is the purpose of an Inflater of Layout in Android development, and how does it fit into the architecture of Android layouts?

→ In the Android development, the purpose of a layout inflater is to convert XML layout files into the actual View object during runtime. It fits into the architecture by dynamically creating user interface elements, allowing for efficient resource utilization and enabling the creation of dynamic UI. The XML layout files describe the arrangement and properties of various views like

button, text fields and many more.

Q.03 Explain the concept of a Custom Dialog Box in Android application. provide example to illustrate its use.

→ A Custom Dialog Box in Android application is a pop-up window that developers can create and design to display custom content or functionality, allowing user to interact with it without leaving the current screen.

⇒ Code:

- XML files

< Linear Layout

```
    xmlns: android = "http://schemas.android.com/apk/res/android"  
    android: layout-width = "match-parent"  
    android: layout-height = "match-parent"  
    android: orientation = "vertical" >
```

< TextView

```
    android:id = "@+id/dialog-text"  
    android: layout-width = "match-parent"  
    android: layout-height = "wrap-content"  
    android: text = "This is a custom dialog!" />
```

< Button

```
    android:id = "@+id/dialog-button"  
    android: layout-width = "match-parent"  
    android: layout-height = "wrap-content"  
    android: text = "OK" />
```

</ Linear Layout >



## Main Activity . Kt

```
import android.app.Dialog
```

```
import android.os.Bundle
```

```
import android.view.View
```

```
import android.widget.Button
```

```
import android.widget.TextView
```

```
class MainActivity : AppCompatActivity() {
```

```
    override fun onCreate(savedInstanceState: Bundle?) {
```

```
        super.onCreate(savedInstanceState)
```

```
        setContentView(R.layout.activity_main)
```

```
        val customDialog = Dialog(this)
```

```
        customDialog.setContentView(R.layout.custom_dialog)
```

```
        val dialogText = customDialog.findViewById<TextView>(
```

```
(R.id.dialog_text))
```

```
        val dialogButton = customDialog.findViewById<Button>(
```

```
(R.id.dialog_button))
```

```
        dialogText.text = "This is a custom dialog!"
```

```
        dialogButton.setOnClickListener { }
```

```
        customDialog.dismiss()
```

```
        customDialog.show()
```

3

Q:04 How do activities, services, and the Android manifest file work together to make an Android app? Can you describe their main roles and provide a basic example of how they cooperate to design a mobile app?

→ Activities, service and the Android manifest file are essential components of an Android app, each with its distinct role in the app's functionality and lifecycle.

### i) Activities:

• Role: Activities represent the user interface and serve as the primary building blocks of an Android app. They handle the interaction between the user and the app's functionality.

### ii) Service:

Role: Service run in the background independent of any user interface and perform long-running tasks or background operation. They help maintain app functionality even when the app's UI is not in the foreground.

### iii) Android manifest File:

Role: The android manifest file is a crucial configuration file that provides essential information about the app to the Android system. It defines various components and their attributes, permission, and app-level setting.

→ Android Manifest.xml:

```

<manifest xmlns:android="https://schemas.android.com/apk/res/android"
    package="com.example.notetakingapp">
    <application>
        <activity android:name=".MainActivity" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".NoteListActivity" />
        <activity android:name=".NoteEditActivity" />
        <service android:name=".NoteSyncService" />
    </application>
    <uses-permission android:name="android.permission.INTERNET" />
</manifest>

```

→ Activities:

- **MainActivity :** This is the main entry point when the app is launched.
- **NoteList Activity :** Display a list of notes.
- **NoteEdit Activity :** Allow the user to create or edit note.

# COLLEGE OF ENGINEERING

## → Service

- NoteSyncService : A background service responsible for syncing notes with a remote server.

Q:05 How does the Android Manifest file impact the development of an Android application? Provide an example to demonstrate its significance?

→ These are critical components in Android app development. It serves several key purposes, impacting various aspects of the app's functionality and behavior.

i] Component Declaration: The Manifest file declares all the app's components, including activities, service, broadcast receivers, and content providers.

ii] Permission: It specifies permission that the app needs to access system resources or perform specific actions, like accessing the internet, using the camera, or reading contacts.

iii] Intent Filters: Intent filters define how components respond to intent allowing the Android system to route requests to the appropriate component.

iv] App Configuration: The Manifest file can include metadata and configuration setting for the app, affecting its behavior.

v] App Entry Point: It designates the app's main activity, the starting point when the app is launched.

Code

```

<manifest xmlns:android = "http://schemas.android.com/apk/res/android"
    package = "com.example.weatherapp">
    <application>
        <activity android:name = ".WeatherActivity" >
            <intent-filter>
                <action android:name = "android.intent.action.MAIN" />
                <category android:name = "android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

        <service android:name = ".WeatherFetchService" />
        <uses-permission android:name = "android.permission.INTERNET" />
        <uses-permission android:name = "android.permission.ACCESS_COARSE_LOCATION" />

        <activity-alias
            android:name = ".WeatherAliasActivity" >
            android:enabled = "true"
            android:targetActivity = ".WeatherActivity" >
            <intent-filter>
                <action android:name = "android.intent.action.VIEW" />
                <category android:name = "android.intent.category.DEFAULT" />
                <data android:scheme = "http" />
                <data android:scheme = "https" />
            </intent-filter>
        </activity-alias>
    </application>
    </manifest>

```

Q: 06

What is the role of resource in Android development?

Discuss the various types of resources and their significance in creating well-structured application. Provide examples to clarify your points.

→ Resources in Android development refer to external assets such as images, layouts, strings, and more that are crucial for building well-structured and adaptable applications.

i) Layout Resources: Layout resources define the UI structure of Android activities and fragments, facilitating a clean separation between UI design and code logic.

### Code

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

ii) String Resources: String resources store text and string constants, making it easier to support multiple languages and maintain consistent text throughout the app.

### Code

```
val welcomeMessage = getString(R.string.welcome_message)
```

iii)

Drawable Resources: Drawable resources store images, icons, and graphics, catering to various screen densities and resolutions for different devices.

Code

```
val imageView = findViewById<ImageView>(R.id.image_view)
imageView.setImageResource(R.drawable.ic_logo)
```

iv)

Color Resources: Color resources define colors used in the app's UI, facilitating a consistent color scheme and making it easy to update the app's color theme.

Code

```
val primaryColor = ContextCompat.getColor(this, R.color.primary_color)
```

v)

Style Resources: Style resources define reusable styles for UI elements, promoting a consistent look and feel throughout the app.

Code:

```
val textView = findViewById<TextView>(R.id.text_view)
textView.setTextAppearance(R.style.TextAppearance_App_Title)
```

Q:07

How does an Android Service contribute to the functionality of a mobile Application? Describe the process of developing an Android Service!

# Ch 10: Android Services

→ An Android service is a component that runs in background, independently of the user interface and contributes to the functionality of a mobile application by performing long running tasks, handling background processes, and providing functionalities.

⇒ Contribution to App Functionality.

i) Background Processing : Service are ideal for tasks that should continue running even if the user switches to another app. For example, a music player app uses a service to play music while the user interacts with other parts of the app.

ii) Long Running :

iii) Foreground : These type of service that perform operations in the background is noticeable for the users. This kind of service must display a notification and it should continue running even user is not dealing with the help.

iv) Bound : In the bound client-server interface that allows components to interact with the service, send requests, receive results and even do so across processes with interprocess communication (IPC). no own lifecycle. use the lifecycle of the activity or fragment they were bounded.

⇒ Process of Developing an Android Service.

i) Create a Service class : start by creating a class that extends 'Service' or one of its subclasses.

ii) Define Service Lifecycle : Implement methods like 'onCreate()',

'OnStartCommand', and 'OnDestroy' to define how your service should behave during its lifecycle.

- 3] Register in the manifest : declare your service in the android manifest.xml file. This tells the Android system about the existence of your service.

### Code

```
<service
    android:name = ".myService"
    android:exported = "false" />
```

- 4] Start and Bind to the Service :- You can start a service using 'startService(intent)' or bind to it using 'bindService(intent, serviceConnection, flags)'. Starting the service typically triggers the 'onStartCommand()' method, while binding allows communication between the service

- 5] Interact with the Service : communicate with your service through intents or other mechanisms, depending on whether you started or bound to it

- 6] Handle Background Tasks : Implement the core functionality of your service to perform the required background tasks. Make sure to manage threads or handle UI if your tasks are long-running.

- 7] Clean Up : In the 'OnDestroy()' method, release any resources, unregister listeners, and stop and going fast to ensure proper cleanup.

## Introduction to Google Cloud Functions

(GCP - Google Cloud Platform)

8] Testing and Debugging: Test your service thoroughly to ensure it behaves as expected. Debugging tools like Logcat can help in diagnosing issues.

9] Permissions and Security: Consider the permission your service needs and ensure that they are declared in the manifest. Be mindful of security practices, especially if your service interacts with sensitive data.

“service” - main function  
“info” - metadata

~~not  
optional~~