

Assignment - 1

- ① 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.

Ans - ① The remote work trend has impacted Android apps:-

- ① Demand for collaboration tools → Increased need for video conferencing and communication apps.
- ② Security Emphasis → Stronger focus on app security due to sensitive data sharing.
- ③ Cloud Integration → Integration with cloud services for efficient data access.
- ④ Remote Work Management → Apps for monitoring and managing remote employees.
- ⑤ Flexible Work Solutions → Apps for scheduling and time tracking.
- ⑥ Subscription Models → Adoption of subscription-based models for sustained income.

② What is the purpose of an Inflater of Layout in Android development, and how does it fit into the architecture of Android Layouts?

Ans - ② The inflater helps create and populate the user interface by dynamically inflating XML layouts, allowing developers to define the app's UI structure and appearance in a separate, easily modifiable format.

① Layout XML files → It transforms XML Layout files into View objects, which represent UI components like buttons, text fields and images.

② Dynamic UI Creation → It enables the dynamic creation and population of the user interface, facilitating flexibility and modularity.

③ Separation of Concerns → It separates UI design (defined in XML) from program logic, promoting clean and maintainable code.

④ Efficient Resource Management → It efficiently manages UI resources loading, improving memory usage and performance.

⑤ Consistent UI → It ensures a consistent UI design by allowing developers to reuse layout components across multiple screens or activities.



MEHSANA DISTRICT EDUCATION FOUNDATION SANCHALIT
U. V. Patel College of Engineering
GANPAT UNIVERSITY, KHERVA - 382 711 DIST. MEHSANA. (N.G.)

③ Explain the concept of a Custom Dialog Box in Android application. Provide examples to illustrate its use.

Ans - ③ Custom Dialog Boxes are implemented by extending the 'DialogFragment' class in Android.

- They allow developers to design the dialog's layout using XML files and customize its appearance and behaviour programmatically.
- Custom Dialog Boxes can include various UI elements such as TextViews, EditTexts, Buttons, or any other widgets needed for the specific purpose of the dialog.
- Developers can control when and how the dialog is displayed, making it suitable for a wide range of use cases.

Example

Create a Dialog Fragment:

~~public class CustomLoginDialogFragment extends DialogFragment {~~

~~public Dialog onCreateDialog(Bundle savedInstanceState) {~~

~~AlertDialog.Builder builder = new AlertDialog.Builder(getActivity());~~

~~LayoutInflater inflater = getActivity().getLayoutInflater();~~

~~View dialogView = inflater.inflate(R.layout.custom_login_dialog,~~

~~builder.setView(dialogView);~~

~~return builder.create();~~

Display the Custom Dialog:

```
import android.app.AlertDialog
import android.app.Dialog
import android.os.Bundle
import android.fragment.app.DialogFragment

class CustomLoginDialogFragment : DialogFragment() {

    override fun onCreateDialog(savedInstanceState: Bundle?): Dialog {
        val builder = AlertDialog.Builder(activity)
        val inflater = requireActivity().layoutInflater
        val dialogView = inflater.inflate(R.layout.custom_login_dialog, null)
        builder.setView(dialogView)

        return builder.create()
    }
}
```

Display the custom Dialog:

```
val customDialog = CustomLoginDialogFragment()
customDialog.show(supportFragmentManager, "custom-login-dialog")
```

Handle user Input:

```
val loginButton = dialogView.findViewById<Button>(R.id.login_button)
loginButton.setOnClickListener {
    dismiss()
}
```

3



MEHSANA DISTRICT EDUCATION FOUNDATION SANCHALIT
U. V. Patel College of Engineering
GANPAT UNIVERSITY, KHERVA - 382 711 DIST. MEHSANA. (N.G.)

④ 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 in Kotlin.

Ans - ④ ① Activities:

Role → Activities represent the user interface and user interactions. Each screen or UI component is typically implemented as an activity.

Ex → In a Kotlin Android app, you might have activities for the main screen, settings, and a login page.

② Services:

Role → Services handle background tasks and long-running operations, often without a user interface. They ensure that the app can perform tasks asynchronously.

Ex → A Kotlin service might handle data synchronization in the background or manage media playback.

③ Android Manifest file:

Role → The manifest file defines the app's components, permissions and configurations.

Ex → In the manifest, you declare activities, services, and their intent filters, specifying how they interact.

Cooperation Example:-

MainActivity.kt :

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

SettingsActivity.kt :

```
class SettingsActivity : AppCompatActivity() {  
}
```

BackgroundService.kt :

```
class BackgroundService : Service() {  
    override fun onCreate() {  
    }  
}
```

```
override fun onStartCommand(intent: Intent?,  
    flags: Int, startId: Int): Int {
```

return START_STICKY

```
} }
```



MEHSANA DISTRICT EDUCATION FOUNDATION SANCHALIT
G. V. Patel College of Engineering
GANPAT UNIVERSITY, KHERVA - 382 711 DIST. MEHSANA. (N.G.)

AndroidManifest.xml

```
<manifest xmlns: android = "http://schemas.android.com/apk/res/android"
    package = "com.example.myapp" >

    <application
        android:allowBackup = "true"
        android:icon = "@mipmap/ic_launcher"
        android:label = "@string/app-name"
        android:roundIcon = "@mipmap/ic_launcher-round"
        android:supportRtl = "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>

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

        <service android:name = ".BackgroundService" />

    </application>

</manifest>
```

Q5 How does the Android Manifest file impact the development of an Android application? Provide an example to demonstrate its significance.

Ans-5 Impact of the Android Manifest file:

① Component Declaration → The manifest file declares all components of your Android app, such as activities, services, broadcast receivers, and content providers. Without these declarations, the Android System wouldn't recognize or launch your app's components.

② Permissions → It specifies the permissions required by your app to access system features and data, ensuring that your app functions correctly and securely.

③ App Configuration → The manifest defines various configuration settings, including the minimum Android API level required, screen orientation, theme and more. These settings influence the app's behavior and appearance.

④ Intent Filter → You can define intent filters in the manifest to specify how your app interacts with other apps. This allows your app to handle actions like opening specific file types or URLs.



Example

MainActivity :

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

Android Mani fest. xml

```
<manifest xmlns: android = "http://schemas.android.com/apk/res/android"  
    package = "com.example.myapp">  
  
<application  
        android: allowBackup = "true"  
        android: icon = "@mipmap/ic_launcher"  
        android: label = "@ string / app-name"  
        android: soundIcon = "@ 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 >
```

⑥ What is the role of resources in Android development? Discuss the various types of resources and their significance in creating well-structured application. Provide examples to clarify your points.

Ans - ⑥ Roles of Resources :-

1) Separation of Concerns → Resources separate content, layout, and other assets from the code, following the Model-View-Controller (MVC) design pattern. This separation makes it easier to update and maintain the app.

2) Adaptability → Resources enable adaptability by allowing different layouts, strings and images to be provided for different screen sizes, orientation and resolution, ensuring a consistent user experience across various devices.

3) Localization → String resources can be localized for different languages and regions, allowing the app to support multiple languages without code changes.

4) Efficient Asset Management

5) Consistency



MEHSANA DISTRICT EDUCATION FOUNDATION SANCHALIT
U. V. Patel College of Engineering
GANPAT UNIVERSITY, KHERVA - 382 711 DIST. MEHSANA. (N.G.)

Types of Resources :

- ① Layout Resources → These define the UI structure using XML layouts, separating the UI design from the code logic.

Ex → val inflater = LayoutInflater.from(context)

val view = inflater.inflate(R.layout.activity_main, null)

- ② String Resources → String resources stores text string, making them easily translatable.

Ex → val appName = getString(R.string.app_name)

- ③ Drawable Resources → Drawable resources contain images and graphics in various resolutions.

Ex → val logoDrawable = ContextCompat.getDrawable(context, R.drawable.app_logo)

- ④ Color Resources → Color resources define colors used in the app, promoting a consistent color scheme.

Ex → val primaryColor = ContextCompat.getColor(context, R.color.primary_color)

- ⑤ Style Resources → Style resources define the visual appearance of UI elements.

Ex → textView.setTextAppearance(context, R.style.AppTextStyle)

⑦ How does an Android service contribute to the functionality of a mobile application? Describe the process of developing an Android service.

Ans ⑦ An Android service contributes to the functionality of a mobile application by enabling background tasks and long-running operations, allowing the app to perform actions independently of the user interface.

Developing an Android Service:-

① Create a Kotlin Class:

```
class MyService : Service() {  
    ...
```

② Override Lifecycle Methods :-

```
override fun onCreate() {  
    ...
```

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

```
    override fun onDestroy() {  
        ...
```



MEHSANA DISTRICT EDUCATION FOUNDATION SANCHALIT
D. V. Patel College of Engineering
GANPAT UNIVERSITY, KHERVA - 382 711 DIST. MEHSANA. (N.G.)

(3) Start and Stop the Service:

val serviceIntent = Intent(context, MyService::class.java)
context.startService(serviceIntent)

(4) Declare in Manifest:

<service android:name=".MyService" />

(5) Service Behavior: Define how your Service behaves
in the 'onStartCommand()' method. Return value
like 'START_STICKY' to specify how the
Service should respond to certain condition.

*Adis
6.10.23*