

Assignment 1 [MADJ]

Q-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 business in the mobile App industry.

→ As per my knowledge, one notable trend in the mobile APP industry that was influencing the Android platform was the rise of progressive web apps [PWAs]. PWAs are web apps that offer app-like experiences directly through web.

- Impact on android app developers:

1) Platform capability:

PWAs are designed to work seamlessly across various platforms and devices, including Android. Developers had to consider creating PWAs alongside traditional Android apps to ensure broad accessibility.

2) Progressive Enhancement:

Developers needed to adopt progressive enhancement strategies to ensure a responsive user experience, similar to native apps.

3) Enhanced user experience:

PWAs aimed to provide a smoother and more engaging user experience, which set higher expectations for Android app developers. This encouraged them to focus on the quality and performance of their apps to compete efficiently.

- Impact on Business in the mobile APP Industry!

1) cost savings: Business can potentially save one development costs by investing in a single PWAs that works across platforms, including Android, rather than building separate native apps.

2) Increased reach:

The focus on delivering app-like experiences through PWAs encouraged businesses to prioritize user engagement and retention, ultimately benefiting their mobile strategy.

3) Competition and Innovation:

The rise of PWAs introduced competition, driving business to innovate their Android apps to keep up with user expectations and technology trends.

Q-2) What is the purpose of an Inflater of layout in Android development and how does it fit into the architecture of Android layout?

→ Dynamic UI inflation: Layout inflater is used to create instances of Android view objects from XML layout resource files at runtime.

→ Reusability: It promotes reusability of UI components by defining their structure and appearance in XML layout files, allowing it easier to instantiate and populate them in different parts of an APP.

→ Separation of concerns: Layout inflater helps maintain clear separation between the UI design and the code that manipulates and interacts with these UI elements.

Q) Architecture of Android layouts.

→ XML layout files: Developers design the layout structure of UI elements in XML resource file.

- Activity / fragment: In the Java or Kotlin code of an Android activity or fragment, developers use the layout inflater to "inflate" or parse the XML layout files, creating a hierarchy of view objects. This is typically done within the "onCreate" method.
- View hierarchy: The result of inflating the layout XML is a hierarchy of view objects, with the root view being the top-level layout.
- Data Binding & Event handling: Developers of the bond declare to these views using data binding libraries or handle user interactions by attaching event listeners.

Q-3) Explain the concept of custom Dialog Box. In Android application, provide examples to illustrate its use.

→ A custom provide examples to illustrate its use window that developers can design and customize to show specific information, receive input from user or perform action without navigating to a new screen or activity; custom ~~Dialogue~~ Boxes are helpful for displaying message, alerts, forms, or any custom content in continued and visually appealing manner.

1. Design flexibility: custom Dialog Boxes allow developers to create unique and tailored user interface.
2. Contextual use: They are typically used when you want to capture user input, show confirmation without taking the user to a different screen.

3. User Interaction: custom Dialog Boxes can contain buttons, text fields, checkboxes, any other UI element, allowing users to interact with the content inside the dialog.

→ Examples of custom DialogBox uses:

1) Information Dialog: A common use case is asking the user for confirmation before performing a critical action.

2) Login or Registration Dialog: Instead of navigating to a separate screen for login or registration, a custom dialog box can pop up, prompting the user to enter their credentials.

Code:

```
import android.app.AlertDialog  
import android.app.content.DialogInterface  
import android.os.Bundle  
import android.app.Activity.  
  
class MainActivity : AppCompatActivity() {  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
  
        val builder = AlertDialog.Builder(this)  
        builder.setTitle("Custom Dialog Example")  
        builder.setMessage("This is a custom dialog box")  
        builder.setPositiveButton("OK") { dialog, which ->  
  
            val dialog = builder.create()  
            dialog.show()  
        }  
    }  
}
```



Q-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 how they cooperate to design a mobile app?

→ Activities, services & the Android file are essential components in the Android app architecture each with distinct role that contribute to the functionality and behaviour of an app.

1. Activities

Role: Activities represent the user interface and screen of an android app they handle user interaction, display all elements and manage the UI flow.

Example: Imagine a simple note-taking app. Each screen of the app, such as the note list, note editing and setting, can be implemented as separate activities.



2. Services

Role: Services run in the background and perform long-running or background tasks without a user interface.

Example: In our note-taking app, you might have a service that periodically backs up notes as a cloud sync without showing a user interface.

3. Android manifest file

Role: The Android manifest file is a configuration file.

Page No.

Provides essential information about the app to the Android system. It declares the app's component and other settings.

Example: In the manifest file, you define which activities are part of your app, specify permissions & declare services your app uses.

→ How they cooperate!

1) Activities:

- The app starts with an activity showing a list of notes
- When the user taps on a note, another activity opens to display and edit the note's content.
- Users can navigate between activities using buttons or gestures.

2) Services:

- While the user is using the app, a service runs in the background to periodically save the user's notes to cloud storage.
- This service doesn't have a user interface but operates independently to ensure data is continuously backed up.

3) Android manifest file:

- In the manifest file, you declare the activities and services used in your app
- You specify permission like "INTERNET" to allow the app to access the Internet for cloud backup.
- The manifest files also defines which activity to start when the app launches.



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

Q-5) How does the Android manifest file impact the development of an android application? provide an example to demonstrate its significance.

→ The android manifest file impacts app development by:

1. common declaration :- Declaring app content components to define the app's structure
Ex:- <activity android:name=".mainActivity" />

2. App permissions:- specify permission for accessing device resources.
Ex:- <uses-permission android:name="android.permission.CAMERA" />

3. Intent filters:- defining how the app responds to external options or requests
Ex:- Registering to open PDF files when tapped.

```
<activity android:name=".PdfViewActivity">
```

```
    <intent-filter>
```

```
        <action android:name="android.intent.action.MAIN" />
```

Page No.

<category android:name="android.intent.category.DEFAULT">
- <action android:name="android.intent.action.MAIN" />
<intent-filter>

<activity>

Q-6] what is the role of resources in Android development
Discuss the various types of resources and their
significance for creating well-structured applications. provide
examples to clarify your points.

→ Resources in Android development are essential components
that helps you create well-structured and flexible applications.
they serve several purposes, such as separating code
from UI.

1) Layout Resources.

XML layouts: These define the structure and appearance of your
app's user interface. They help keep the UI separate
from code logic, making it easier to maintain and
adapt.

Example: A layout XML file specifies how elements like
buttons and text fields are arranged on the screen.

2) Drawable Resources.

Images & Icons: Drawable resources store images, icons &
other graphics used in your app. Diff.
resources can be provided for diff. screen
devices.

3) Color Resources.

Colors: By defining colors in resources you can maintain
consistent color scheme across your app & easily
switch them.

Ex: A color resource [primary_color] defines the primary
color used in the app's UI elements.

4) Style Resources:

- Theme & style: style define the appearance of UI elements, making it simple to apply consistent styling across app.

Ex: you can create a custom style to define fonts, colors & other visual attribute

5) Dimension Resources:

- size & dimension: string size & margin in res file make it easy to adjust layouts for different screen size and orientations.

Ex: dimension resource defines a consistent margin size for elements.

6) Row Resources:

- Row data: you can store non-compiled resources like table, list files in the 'res/layout' dir.

Ex: String action in the 'row' file for config data.

7) Animations & Drawable Animation Resources

- Animations: you can define animations in xml resource file, making it simple to get and apply animation to UI elements

Ex: A resource file can define a code in Animation for an image view.

Q-7) How does an Android service contribute to the functionality of a mobile application & describe the process of developing an android service. write an simple language & include min points.

→ An android service plays a critical role in the functionality of a mobile application by allowing tasks to run in the background.

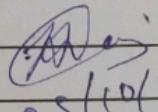
⇒ Contribution of Android

1. Background processing :- Service allow apps to perform task in the background without blocking the user interface.
2. long running operations : Service is ideal for lengthy operations that require more time to complete.
3. Inter-components communication : services enable components like activities, broadcast receivers & other service

→ Process of developing an Android service:

1. Define the service class :- Create a kotlin class that the service class override methods like onCreate(), onDestory(), onStartCommand() to define the behavior of service.
2. Configure service in manifest : Declare service in the Android manifest XML file to inform the Android System about its existence & configuration.
3. Start or bind the service : Decide whether you want to start a service or bind it to other components.
4. Implement Service logic : In service class implement the specific logic your service needs to perform its task

5. Handle lifecycles! Release Resources when they're no longer needed.
6. Interact with other components! Use appropriate mechanisms to facilitate the communication.
7. foreground services : if your service needs to run in the foreground.
8. testing : Thoroughly test your service to ensure it functions as expected.


05/10/13