



### Assignment : I

Q. 1

Based on your understanding identify a recent business trend that has influenced the android Platform Explain how this trend impacts developer and business in the mobile app industry

→

One significant trend in the Android App industry was the increasing emphasis on user privacy and data security

→

Impact on android App Developers

1. Impacted permissions and consent!

→ Developers had to be more transparent about the data their apps collect and request explicit user consent. This meant redesigning permission dialogs and ensuring that user understood why certain data was being collected

2.

Limitations on Advertising

→ for apps relying on advertising revenue, changes in ad tracking and targeting due to privacy concerns affected their monetization strategies. Developers needed to adapt to these changes possibly exploring new monetization models

3.

Data Handling and Storage

→ Developers had to review how they implemented and stored user data, implementing certain duty implementing this could lead to increased



-> impacts on Businesses

1. Compliance costs :-

-> Businesses operating in the android app industry need to allocate resources for compliance with stricter data privacy regulations. This could include legal and technical measures to ensure data protection.

2. monetization challenges

-> Businesses relying heavily on user data for advertising and personalized content faced challenges in maintaining their revenue streams. They needed to find new ways to engage users and generate income.

3. Reputation management

-> Privacy breaches or mishandling of user data could result in severe reputational damage. Building and maintaining trust with users became even more critical.

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

-> In android app development, think of the "inflater" like a magic tool. It helps turn your design plans into actual buttons, text boxes, and other things you see on your phone's screen.

-> Purpose of LayoutInflater

1) Dynamic UI inflation : LayoutInflater is used to create instances of android view objects from XML layout resource files at runtime.

2) Reusability: It promotes the reusability of UI components by defining their structure and appearance in XML layout files making it easier to instantiate and populate them in different parts of an app.

- Architecture of Android layouts
  - 1) XML Layout files : Developers design the layout structure of UI elements in XML layouts resource file.
  - 2) Activity / Fragment in the Java or Kotlin code of an Android activity or fragment, developers use the LayoutInflater to "inflate" or parse the XML layout files creating a hierarchy of view objects. This is typically done within the 'onCreate' method.
  - 3. View Hierarchy : The result of inflating the layout XML is a hierarchy of view objects with the root view object, with the root view being the top level layout.
  - 4. Data Binding or Event Handling : Developers often bind data to these views using data binding libraries or handle user interaction by attaching event listeners.
  - 5. Rendering on the screen : The Android system is responsible for rendering this hierarchy of views defined screen according to the layout specifications defined in the XML file.
- Q.3 Explain the concept of custom Dialog Box in Android application. Provide examples to illustrate its use.
- A custom Dialog Box in Android applications

is a pop-up window that developers can design to customize to show specific information, receive input from users or perform actions without navigating to other activities. Custom Dialog Boxes are helpful for displaying messages, alerts, forms, or any custom content in a controlled and visually appealing manner.

1. Design Flexibility: Custom Dialog Boxes allows developers to create unique and tailored user interface.
  2. Contextual use: They are typically used when you want to capture user input or show information without taking the user to a different screen.
  3. User Interaction: Custom Dialog Boxes can contain buttons, text fields, checkboxes, and other UI elements allowing users to interact with the content inside the dialog.
- Examples of custom Dialog Boxes
- 1) Confirmation Dialog: A common use case is using 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.
  - 3) Error messages: When there's an error, such as network issues or invalid input, a custom dialog can display an error message with details helping the user understand and correct the problem.
  - 4) Date & Time Picker: You can create a custom dialog box for selecting dates or times.

### Code

```
import android.app.AlertDialog
import android.content.DialogInterface
import android.os.Bundle
import androidx.appcompat.AppCompatActivity
```

Class YourActivity : AppCompatActivity () {

    Override fun onCreate(savedInstanceState: Bundle?) {

        super.onCreate(savedInstanceState)

        setContentView(R.layout.activity\_main)

    val builder = AlertDialog.Builder(this)

        builder.setMessage("This is a custom dialog box")

        builder.setPositiveButton("oh") { 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 role and provide a basic example?

->

Activities Services the Android manifest file are essential components in the android app architecture each with distinct role that contribute to the functionality and behaviour of an app.

1.

GHANSHYAM

Activities :- Role Activities represents the user interface and logic of an android app they

is  
Cu.  
f8

→ Example: imagine a simple note editing application such as the note app. note editing and can be implemented as responsive activities.

## Q8 2. Services

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

1. Example: In our note editing app you might have a service that periodically backs up notes to a cloud server without showing a user interface.

2. C

3. Android manifest file is a configuration file provides essential information about the app to the Android system. It declares the app's components, permissions and other settings.

2 → 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 sync the user's note to cloud database.

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 access the internet, cloud backup.

Q.5 How does the Android manifest impact the development of an android application? Provide an example to demonstrate its significance.  
→ The android manifest file impacts app development

1. Component Declaration : Declaring app components to define the app's structure  
ex <activity android:name="mainActivity" />
2. App permission : specifying permissions for accessing device resources  
ex <uses-permission android:name="android.permission.CAMERA" />
3. Intent filters : Defining how the app responds to external actions or requests.  
ex: Registering to open PDF files when tapped

```
<activity android:name="PdfViewerActivity">
    <intent-filter>
        <action android:name="android.intent.action.VIEW" />
        <category android:name="android.intent.category.DEFAULT" />
        <data android:mimeType="application/pdf" />
    </intent-filter>
</activity>
```

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

→ Resources in Android development are essential components that helps you create well-structured and flexible apps. They serve several purposes, such as separating code from content adapting to different devices, and simplifying localization. Here are the main types of resources and their significance.

## 1) Layout Resources:

- XML Layout: 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 or text fields are arranged on the screen.

## 2) Drawable Resources:

- Images and Icons: Drawable resources store images, icons, and other graphics used in your app. Different versions can be provided for different screen densities.
- Example: You might have 'ic\_launcher.png' for the app icon or separate versions for low, medium, or high density screens.

## 3) String Resources:

- Text and Strings: Storing text in resource files allows for easy localization and update without modifying code.

Example: A string resource containing the app's name, which can be changed for different languages.

## Color Resources

Color: By defining colors in resources, you can maintain



Consistent color scheme across your app and easily switch themes

- Example :- A color resource (Primary\_color) define the primary color used in the app's UI elements

### 5) Style Resources

- Themes and styles :- Styles define the appearance of UI elements making it simple to apply consistent styling across app
- Examples :- You can create a custom style (AppTheme) to define fonts colors and other visual attributes

### 6) Dimension Resources

- Sizes and Dimensions : Storing sizes and margins in sizes file makes it easy to adjust layouts different screen sizes and orientations
- Example : A dimension resource (margin-small) defines a consistent margin size for elements

### 7) Raw Resources

- Raw Data : You can store non-compiled resource like audio, video, or text files

in the 'res/layout' directory

- > 8) Animations and Drawable Animation Resources:
- Animation :- You can define animations in xml resource file, making it simple to resources and apply animations to UI elements.

7)

Example: A resource file ('faderin.xml') code define a code in Animation for an image view

Ques 7

How does an Android service contribute to the functionality of a mobile application? Describe the process of developing an android Service write in simple language and include main point

- > An android service plays a crucial role in the functionality of a mobile application by allowing tasks to run in the background even when the app is not actively in use.

• Contribution of Android service:

- 1) Background processing : services run tasks in the background ensuring the functions like music playback, location tracking or data syncing can continue without disrupting the user interface.

2) long - Running operating

Such as decoding large files or performing complex calculations without causing the app to freeze.

Ans  
6/10/20

- color: By defining colors in resources, you