

MAD / Rotsavya Hrushul D.

21012011040 CEIT-B



SB-I
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.)

Assignment :- 1

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.

Ans

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

→ Impact on Android App Developers

1. Enhanced permissions and consent:

→ Developers have to be more transparent about the data their app collects and request explicit user consent. This means redesigning permission dialogs and ensuring that users understand why certain data was being collected.

2. Limitations on Advertising

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

- ③ Data Handling and storage
- Developers had to review how they handled and stored user data, implementing stricter data protection measures. This could lead to increased development time & costs.
 - Impacts on Businesses

- ① Compliance costs:
- Businesses operating in the Android app industry needed to allocate resources for compliance with stricter data privacy regulations.
- 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 Regulation 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 LayoutInflator:

- 1) Dynamic UI inflation: LayoutInflator is used to create instances of Android view objects from XML layout resource file 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 reuse them in different parts of an app.
- 3) Separation of concerns: LayoutInflator helps maintain clear separation b/w the UI design and the code that manipulates and interacts with these UI elements.

→ Architecture of Android Layouts:-

- 1) XML Layout files: Developers design the layout & structure of UI elements in XML layout resource file.
- 2) Activity / Fragment: In the Java or Kotlin code of an android activity or fragment developers use the LayoutInflator to "inflate" or reuse 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 being the top-level layout.

ii) Data Binding & Event Handling: Developers often bind data views using data binding libraries or handle user interactions by attaching event listeners.

④ Rendering on the Screen: The Android system is responsible for rendering this hierarchy of views on the device screen according to the layout specification 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 application is a pop-up window that developers can design and customize to show specific information, receive input from users or perform actions without navigating to a new screen or activity. 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's input show information without forcing the user to a different screen.
3. Examples of custom Dialog Box use:
 - 1) Confirmation Dialog: A common use case is asking the user for confirmation before performing a critical action.
 - 2) Confirmation Login & Registration Dialog: instead of navigating to a separate screen for login or registration a custom dialog box can pop up programming the user to enter their credentials.
 - 3) Error Message: 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.

Q-4 How do activities, services and the Android Manifest file work together to move an Android App? Can you describe their main roles and provide a basic example how they co-operate to design a mobile app?

① Activities

→ Role :- Activities represent the user interface and screen of an Android app. They handle user interactions, display UI 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 settings, 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 to a cloud server without showing a user interface.

③ Android Manifest file:

→ Role : The Android manifest file is a configuration file.

2) ~~Contextual Use :- They are typically used when you want to capture user input & show information without taking the~~

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 ~~long~~ gestures.

2) Services :-

which the user is using the app, a service runs in the background to periodically give 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:

```

<manifest xmlns:android="http://schemas-
        android.com/apk/res/android"

    package="com.example.myapp" >
        <application>
            <activity android:name=".MainActivity" >
                <intent-filter>
                    <action android:name="android.intent.
                        action.MAIN" />
                    <category android:name="android.intent.
                        category.LAUNCH" />
    
```

Q:-

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

→ The android Manifest file impacts app development by:

1) Component Declaration: Declaring app component to define the app's structure

~~ex~~ <activity android:name=".MainActivity" />

2) App permissions: Specifying permission 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:mimeType="application/pdf"/>
        </intent-filter>
    </activity>
```

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

D) Layout Resources:

.xml Layout: These define the structure and appearance of your app's user interface. They help keep the UI separate from core logic making it easier to maintain a code.

Ex : A layout xml file specifies how element like buttons and text fields are arranged on the screen.

2) Dynamical Resources

~~images & icons~~: Dynamical resources store images, icons and other graphical used in your app. Different versions can be provide for diff. screen densities.

~~etc~~: you might have 'ic_launcher.png' for the app icon of separate versions from low, medium & high - density screens.

3) String Resources:-

Text and strings: string text in resource files allows for easy localization and update without modifying code.

~~etc~~ :- A string resource ('app-name') contains the app's name, which can be changed for different language.

4) Color Resources

~~colors~~: By defining colors in resources, you can maintaining consistent color scheme across your app & easily switch themes.

~~etc~~ :- A color resource (Primary, color) refine the Primary color used in the app's UI elements.

⑤ Style Resources

- Themes & styles : styles define the appearance of UI element, making it simple to apply consistent styling across app.
- Ex :- you can create a custom style to define fonts, colors and other visual attributes.

⑥ Dimension Resources

• sizes and Dimensions : storing sizes and margins in dimension resources file makes it easy to adjust layouts for different screen sizes & orientation.

Ex :- A dimension resource refines a consistent margin size for defines all elements.

⑦ Raw Resources

- Raw : you can store non-compiled resources like audio, video or text files in the 'raw' directory.

Ex :- storing a JSON file in the 'raw' folder for configuration sake.

Ques:- How does an Android service contribute to the functionality of a mobile application?
 Describe the process of developing an Android Service writing in simple language and include main points.

- 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 essential functions like music playback, location tracking or battery monitoring can continue without disrupting user interface.
- ② long - Running operations: Services are ideal for operations that take a long time to complete such as downloading large files or performing complex calculations without causing the app to freeze.
- ③ Persistent services: Some services run in the background displaying a persistent notification to keep the user

aware of enjoying tasks like navigation on chat applications.

- ④ Inter-component communication: Services can communicate with other app components through interfaces allowing data exchange and coordination.

- Developing an Android Service

- ① Create a service class
- Extend the 'Service' class or one of its subclasses like 'IntentService' or 'JobService'
 -

② Declare in the manifest:-

- Register your service in the Android Manifest XML file to make it accessible to the system and other components.

③ Service Lifecycle:-

- Understand the service's lifecycle methods and override them as needed.
- Service can run in three methods:

④ Start and stop the service.

- start a service using 'startService()' and stop it using 'stopService()' or 'stopSelf()'.

⑤

foreground - Services

- To execute a foreground service, provide a notification that informs the user about ongoing tasks.
- use 'startForeground()

⑥

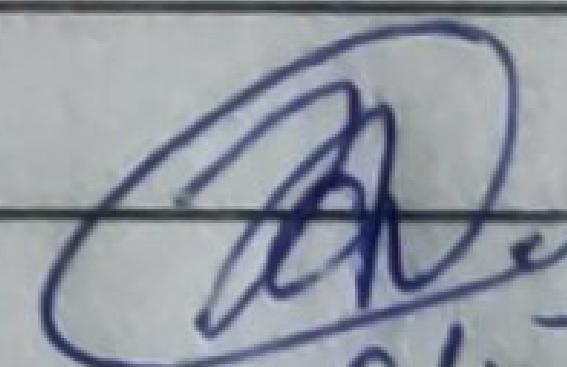
Thread management

- When performing time-consuming operations, consider using ~~concurrent threads~~ or ~~Asynchronous~~ to prevent blocking the main UI thread.

⑦

Testing

- Thoroughly test each service to ensure it works as expected, including scenarios like app ~~background~~ interruption and restarts.


OSIP/23