MAD Assignment-1 21012011064

Ganpat University | U.V. Patel College of Engineering

U.V. Patel College of Engineering

GANPAT UNIVERSITY, KHERVA-384012, DIST - MEHSANA. (N.G.)

1) ~~Recent business trend wh~~

Q.1 As Per my knowledge, one notisable trend in the mobile app industrys that was influcing the Android platform ans the rise of progresive web App (PWAS). PWAS are web applications that often app-like experiences directly through web browsers.

- Impact on Android Add Developess:

1. Cross-platform Compatibility :- PWAS are designed to work seamlessly more engaging user experience, which set highes expetition for Android app developers.

2. Progresive Enhancement :- Developess needed to adopt progresive enhancement strategies to ensure that Android apps remained competivie by offering progressive and responce user experience, similas to PWAS.

-

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

→ In Android development, On "Inflater refers to the layout Inflater, which plays a crucial role in creating a user interface (UI) from XML layout files.

21012011064

1. XML Layout files:-
In Android, UI components are often defined using XML layout files.

2. Layout Inflation:-
When your Android app runs, It need to turn these XML layout files into actual view objects that can be displayed on the screen.

3. Layout Inflater:-
It is responsible for reading the XML layout files and Instantiating the corresponding view objects in memory.

4. Dynamic UI creation:-
Layout inflation. IS particularly valuable when you need to create UI elements dynamically, for example, in responce to user interaction.

5. Binding Data:-
Once the view objects are created, they can be further customized and data can be bound to them.

Ganpat University | U.V. Patel College of Engineering

U.V. Patel College of Engineering

GANPAT UNIVERSITY, KHERVA-384012, DIST - MEHSANA. (N.G.)

Q.3 A CustomDialogBox in Android is a popup window that developers can design and customize to display information, options, or actions to the user. It's a versatile UI element for showing alerts, input forms, confirmation dialogs, or any other customs interaction that doesn't fit the standard layout of an Activity.

Ex: When you need to provide a tailored user experience or gather specific input from the user without navigating to a new screen like alerting warnings asking for input at runtime like grades, Name, etc into, for noting etc.

Q.4 Activities, Services, and the Android Manifest file are fundamental components of an Android app, each with it's own specific Role in the overall application architecture.

1.) Activities

Role: Activities represent individual screens or user interfaces in an Android App.

Example: Imagine a simple email app with two activities: one for composing emails & another for viewing the inbox.

2. Services:

Role: Services are background components that perform long-running operations without a user Interface.

21012011064

3. Android Manifest file:

Role: The AndroidManifest.xml file is a configuration file that defines essential information about the app. its components, and required permissions.

Example: In the manifest file, you specify which activities & services the app contains, their properties & any permission required, among other things.

Q.5 The Android manifest its significance. file is a critical. Component in Android app development it serves several significant purposes that impact the development and functionality of an Android application.

* Examples :-

1. Component Declaration :-

The Android manifest file is where you delare all the Components of your android application, including activities, services, broadcast recivers and Contect providers.

```
-> <application...>
     <activity android:name=".main Activity">
     <intent-filter>
       <action android:name="android:intent.action.MAIN"/>
       <category android:name="android.intent.category.LAUNCHER"/>
     </intent-filter>
     </activity>
     <service android:name=".MyService"/>
     <receiver android:authorities="com.exple.myapp.provider"/>
               :name=".MyReceiver"/>
```

```
<Provider
    android:name = ".MyContentProvider"
    android:authorities = "com.example.myapp.Provider" />
</application>
```

2. Application Configuration :-

```
<application
    android:icon = "@drawable/app_icon"
    android:label = "@String/app_name"
    android:theme = "@style/app_theme"
    android:versionCode = "1"
    android:versionName = "1.0" >
```

Q.6 Resources are the additional files static content that your code uses. such as bitmaps, layout definitions user interface string, animation instruction & more.

⇒ Resources types overview.

1.) Animation resources :-

Define pre-determined animations Same animations are saved in res/drawable/ and accessed from the R.drawable class.

2.) Color State List resource :-

Define a color resource that changes based on the view state. Saved in res/color/ and accessed from the R.color class.

2101201106 4

3. Drawble resources :-

Define various graphics with bitmap or XML.

4. Layout resources :-

Define the layout for your application UI.

5. Menu resources :-

Define the contats of your application Menus.

6. Style resources :-

Define the look and format for UI element.

7. Font resources :-

Define the families so include custom font in XML.

Q.7

1.) Background processing :-

Services allow apps to perform tasks in the background without blocking the user interface.

2.) long running operation :-

Services are ideal for healing operations that require more time to complete such as playing music.

Ganpat University | U.V. Patel College of Engineering

U.V. Patel College of Engineering

GANPAT UNIVERSITY, KHERVA-384012, DIST - MEHSANA. (N.G.)

3.) Inter Component Communication :-

Services enable Components like activities broadcast receivers and other services to communicat with each other efficiently.

4.) Foreground Service :-

Android services can Run in the foreground. even when the app isn't in the foreground.

⇒ Process of Developing an android service :-

1.) Defines the Service class :-

Create a new Java or kotlin class that extend the 'service' class.

2.) Configure Service in manifest :-

Declare your service in the Android manifest. XML file to inform the android system about its exsit. and configuration.

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

3.) Start or Bind the Seriece :-

Declare whether you want to start your service or bind it to other Components.

2101201106

4.) Implement service logic :-

In service class, implement the specific logic your service need to perform its task.

5.) Handle lifecycle :-

Release resource when they no longer needed and consider using 'stopself()' or 'stopservice()'.

6.) Interact with other components :-

use appropriate mechanisume like intents broadcast or call back to facilitate communication.

7.) Foreground Services (optional) :-

If your service need to run in the foreground 'startforeground()'.

19/10/23