[redacted] 210120110 60 _ Prem Paradava.

Ganpat University | U.V. Patel College of Engineering
U.V. Patel College of Engineering
GANPAT UNIVERSITY, KHERVA-384012, 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.

As per my knowledge, one notable trend in the mobile app industry that was influcing the android platform was the rise of progressive web apps (PWAS) are web applications that offers app like experience directly through web browsers.

Impact on android developers:

1. Cross platform connectivity: PWAS are designed to work seamlessly across various platform and devices, including android.

2. Enhance user experience: PWAs are designed to provide a smoother and more engaging user experience, which set higher expectation for android app developers.

3. Progressive Enhancement: Developers needed to adopt progressive enhancement strategies to ensure that android app remained competitive by offering progressive user.

- Impact on Business is the mobile App Industry:

1. Cost Saving: Business could potentially save on development cost by investing is a single PWA that works across multiple platform, including Android, rather than bulding separate native apps.

2. Increased Reach: PWAs enabled business to reach a wider range, including user with android devices, without relying on app store distribution.

3. Improved Engagement: To focus on delivering app-like experience through PWAs encourage business to prioritize user engagement and retention.

4. Competation and Innovation: The rise of PWAs introduce competition deriving business to innovate their android apps to keep up with evoluing user expectations.

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

In android development, an Inflator refers to the Layout Inflator which plays an crucial role is creating a user interface. from XML layout.

1. XML Layout files: In android, UI components are often defined using XML layout files. These files describe the structure and appearance of the UI, specifying things like widgets, their properties, within UI.

2. Layout Inflation: When your android app runs, it needs to turn these XML layout files into actual view object that can be displayed on the screen.

3. Layout Inflator: It is responsible for reading the XML layout files and instanting the corresponding view object is memory. It takes the XML file as input.

4. Dynamic UI creation: Layout inflation is particularly valuable when you need to create UI elements dynamically.

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

6. Displaying UI: after inflation and customization, the view objects can be added to the app's layout hierarchy and display on screen.

Q.3 Explain the concept of a custom dialog Box in android application. provide example to illustrate its use.

In android applications, a custom dialog box is a pop-up window that overlays the current activity and it often used to interact with the user, gather input or display information.

Purpose: custom dialogs are used when you want to present information, recieve user input or perform action within a self contained UI element.

Components: A custom dialog typically consist of various elements like button, text views, images, or input fields.

Customization: developers can design the dialog appearance, layout, behaviours according to their app branding or requirements.

ex:
```
fun ShowCustomDialog() {
    Val CustomDialog = Dialog(this)
    CustomDialog.SetContentView.layout.custom-dialog
    Val messageTextViews = Custom dialog.findViewById
                                <TextView>
    Val okButton = CustomDialog.find Viewsbyid<Button>
                                (R.id.okButton)
    messageTextview.text =" This is custom dialog!"
```

```
okButton.setOnLine {
    customDialog.dismiss() }
```

Q.4 How do activities, services and the android manifest file work together to make a android app. Can you describe their main roles and provide a basic example of how they cooperate to design a mobile app.

1. Activities:

Role: Activities represent the individual screen or UI components in an android app. They manage the user interface and user interaction.

2. Services:

Role: Services are background components that perform long running operations or handel task that don't require a user interface.

3. Android Manifest file:

Role: The android manifest file is like the app's blueprint. It declare the app's components and defines how they interact with the android system and other components.

ex: In android manifest.XML you specify which activities are part of your app, their launch modes, permissions and services declaration.

```
class mainactivity : AppCompactActivity() {
    override function onCreate(savedInstanceState) {
    setContentView(R.layout.activity_main)
    startService button.setOnClickListener {
    val serviceIntent = Intent(this, NotificationService::class.
            service(serviceInt)
        }
    }
}
```

```
class NotificationService : IntentService("Notification Services") {
    override fun onHandelIntent(intent: Intent) {
        if (intent != null) {
            create Notification
        }

Private fun createNotification() {
    val channelID = "my_channel"
if (Build.VERSION.SDK_INT > Build.VERSION_CODES.O) {
    val name = "my_channel"
val notificationManager = getSystemServices(Notification)
}
val builder = NotificationCompact.Builder(this channel)
.setSmallIcon(R.drawable.ic_launcher_foreground)
.setContentText("This is notification from service")
    }
}
```

Q05 How does the Android manifest file impact the development of Android application.? provide an example to demonstrate its significance.

The Android manifest file is a crucial component in the development of an Android application. It serves several important purpose, and its content significantly impacts how the Android system interacts.

- App Configuration
- Component declaration
- Permission
- Intent filters
- App Lifecycle.

Example

```
<manifest xmlns:android="https://schemas.android.com/apk"
 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:theme="@style/AppTheme"
    android:supportsRtl="true">

    <intent-filters>
    <action android:name="android.intent.action.MAIN"/>
    <intent filter>
    </activity>
```

```
<activity android name = "second activity">
    .... declare additional activities here.
</activity>

<user permission android: name = " android . permission.
                                    Intent"
    .... declare require permission
</application>
</manifest>
```

Q06 What is the role of resources is android development discuss the various types of resources and their significance is creating well structured applications. provide example to clarify your points.

Resources play a fundamental role is android development by providing a structural way to manage assets, values, layouts, and other elements, used is your app. They hlp you create flexible, maintainable, and device independent application.

1 Layout resources:
type: XML files is the 'res/layout'.
Significance: define the structure and apperance of app user interface.

```
< Button
android.id = "@ + id /my button"
android: layout_width = "wrap-content"
android : text = " click me".
```

2. Draw resources:
type: Images are drawable assets in the res/drawable directory.
Significance: Store graphics, icons, and images used in your apps.

3. String Resources:
type: String defined in XML files under 'res/values'
significance: Store text images, making it easier to provide translations.

4. String resource:
type: String defined in XML files under 'res/values'
significance: Store text string, making it easier to provide translation and maintain consistency.
ex. res/values/string.xml
<string name="app_name">my app</string>
<String name="Welcome msg">welcome</string>

5. Style resource:
type: Style defined in XML files under 'res/values'.
significance: define reusable styles for UI components.

ex: 'res/values/styles.xml' define styles.
<styles name="mybuttonstyle>
<item name="android.background">@drawable/my_button
<item name="android:textcolor">@color/primary_color</item>
</styles>

Q.7 How does android service contribute to the functionality of a mobile application? describe the process of developing an android service.

Contribution of Android Services:

1. Background Processing: services allow app to perform task in the background without blocking user interface.
2. Long running operations: services are ideal for handeling operations that require more time to complete.
3. Inter-component communication: services enable components like activities that broadcast receiver, and other.
4. Foreground services: Android services can run in the foreground background even when an app isn't is foreground.

Process of developing an android services:

1. define the service class: create a new java class or kotlin class that extends the 'service class' override method like oncreate(), ondestroy()

2. Configure service in manifest: declare your service in the android manifest.xml file to inform the android system about the existance.

3. Start or bind the services: decide wheather you want to start service or bind it to other components.

4. Implement service logic: In service class implement the specific logic your service need to perform.

5. Handel lifecycle: Release resources when you no longer needed.

6. Interact with other components: Use appropriate mechanism like intent, broadcast or callback to facilatate communication.

7. Foreground Services: for this 'start foreground'

8. Testing: Throughout test your services to ensure it functions as expected, including handling various scenarios.

9. Optimization: optimize your service for performance and resources to manage battery usage.

10. Error handling and logging: Implement proper error handling and logging mechanism to diagnose and address issues.

20/10/23