Mansuri Sanabanu Y. Class: CE IT: B Branch: CE
Enrollment: 21012011048 Batch: 5B4

Ganpat University | U.V. Patel College of Engineering

U.V. Patel College of Engineering

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

Mobile Application Development

Assignment: 1

1. Based on your understanding, identify a recent business trend that has influenced the Android app developers and business in the mobile app industry.

→ As per my knowledge, one notable trend on the mobile App industry that has influencing the Android platform, was the rise of progrise web Apps. PWAs are web applications that offer app-like experiences directly through web browsers.

- Impact on Android App Developers:

1. Cross-platform Compatibility: PWAs are designed to work seamlessly across various platforms and devices including Android.
2. 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 improving the quality and performance of their apps to complete effectively.
3. progressive Enhancement: Developers needed to adopt progressive enhancement strategies to ensure that Android apps remained competitive by offering progressive user experiences.

- Impact on Businesses in the Mobile App industry.

1. Cost Saving: Businesses could potentially save on development costs by investing in a single PWA that works across multiple platforms including Android.

2101201104 8

2. increased Reach: PWAs enable business to reach a wider audience, including users with Android devices, without delaying an app store distribution.
3. Improved Engagement: The focus on delivering app-like experiences through PWAs encouraged business to prioritize user engagement and ultimately benefiting their mobile strategy.
4. Competition and Innovation: The rise of PWAs introduced competition driving business to innovate their Android apps to keep up with evolving user expectation and technology trends.

Q What is the purpose of an inflator of layouts 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 a crucial role in creating a user interface (UI) from XML layout files. Its primary purpose is to take an XML layout resource and convert it into a corresponding view object in memory. Here's how the layout Inflator fits into the Architecture of Android layouts:

1. XML layout files: In Android, UI components are often defined using XML layout files.
2. layout Inflator: It is responsible for reading the XML layout files and instantiating the corresponding view objects in memory. it takes the XML files as inputs, parses it and creates the view objects such as TextViews, Buttons as specified in the XML.

Ganpat University | U.V. Patel College of Engineering

U.V. Patel College of Engineering

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

3. Dynamic UI creation: Layout inflation is particularly valuable when you need to create UI elements dynamically for example, in response to user interaction or when working with RecyclerViews to display lists of items.
4. Binding Data: Once the view objects are created, they can be further customized and data binding techniques or programmatically setting properties and values.
5. Displaying UI: After inflation & customization, the view objects can be added to the app's layout hierarchy and displayed on the screen.

Q3 Explain the concept of a CustomDialogBox in Android App. provide examples to illustrate its use.

→ In Android Application, a CustomDialogBox is a pop-up window that overlays the current activity and is often used to interact with the user, gather input or display information.

- purpose: Custom-dialogs are used when you want to present information, receive user input & perform actions within a self-contained, isolated UI element that temporarily interrupts.
- Components: A custom dialog typically consists of various UI elements like buttons, textviews, images to the specific interaction you want to facilitate.

Example:

```
fun showCustomDialog() {
    val customDialog = Dialog(this)
```

2101201104 8

```
customDialog.setContentView(R.layout.custom_dialog)
val messageTextView = customDialog.findViewById<TextView>
                      (R.id.messageTextView)

val OKButton = customDialog.findViewById<Button>(R.id.OKBtn)

messageTextView.text = "CustomDialog !!"
OKButton.setOnClickListener{
    customDialog.dismiss()
}
customDialog.show()
}
```

→ use cases of custom dialogbox: . login . setting . informational pop-up . media playback controls .

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 of how they cooperate to design a mobile app?

→ Activities: Activities represent individual screens or UI components in an Android App. They manage the UI and user interactions.

Services: Service are background components that perform long running operations or handle tasks that don't require a user interface. They can run even if the apps UI not visible.

Android Manifest file: This file is like the app's blueprint. It provide the declare the app's components and defines how they interact with the Android system & other components.

210120110048

Example

In AndroidManifest file, you specify which activity are part of your apps their launch models, permissions and service declarations.

```
class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        startServiceButton.setOnClickListener {
            val serviceIntent = Intent(this, NotificationService::class.java)
            startService(serviceIntent)
        }
    }
}
```

→

```
class NotificationService : IntentService("NotificationService") {
    override fun onHandleIntent(intent: Intent?) {
        if (intent != null) {
            createNotification() } }
    private fun createNotification() {
        val channelID = "my_channel"
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O)
        val name = "my_channel"
        val notificationManager = getSystemService(NotificationManager::class.java)
        notificationManager.createNotificationChannel(channel)
        val builder = Notification.Compat.Builder(this, channelID)
        .setSmallIcon(R.drawable.ic_launcher_foreground)
        .setContentText("This is notification from service")
    }
}
```

2101201104 8

5 How does the Android manifest the impact the development of an Android applications? provide an example to demestrate its significance.

→ The Android manifest file is a crucial component in the development of an Android application. It serves several important purposes, and its content significantly impact how the android system interacts with and manages your App.

App Configuration
Intent filters
App Lifecycle
Compact Declaration
permissions.

Example: <manifest xmlns:android = "http://android.com/res/android"
package = "com.exp.myapp" >
<application
android:allowBackup = true
android:icon = "@ipmap/ic_launcher"
android:label = "@string/app_name"
android:supportRTL = "true"
android:theme = "@style/AppTheme" >
<activity android:name = ".MainActivity" >
<intent-filter>
<action android:name="android.intent.action.MAIN" />
<category android:name = "android.intent.category.
launcher" />
</intent-filter>
</activity>
<activity android:name = ".SecondActivity" />

2101201104 8


```
</activity>
<uses-permission android:name="android.permission.INTERNET" />
</application>
</manifest>
```

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

→ Resource play a fundamental role in Android development by providing a structured way to manage assets, values, layouts and other elements used in your app. They help create flexible, maintainable and device independent applications. The various types of resources and their significance with example:

1. layouts Resource:
XML files in the 'res/layout' directory.
Significance: Define the structure and appearance of the app's user interface.
Example: 'activity_main.xml' defines the layout of your main activity, specifying UI components like buttons, text views and their arrangement.

```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/btn"
    android:text="Click Me" />
```

2101201104g

2. Drawable Resources:
Images and drawable assets in the 'res/drawable' directory
significance: Store graphics, icons and images used in your app.
Example: 'ic_launcher.png' is the app's launcher icon.

3. String Resources:
Strings defined in XML files under 'res/values'.
Store ~~text~~ strings, making it easier to provide translations and maintain consistency.
Example:
<string name = "app_name"> My App </string>
<string name = "welcome_message"> welcome to my app
</string>

4. Color Resources:
colors defined in XML files under 'res/values'.
Significance: Store color values, ensuring consistency in the app's design.
Example:
<color name = "primary_color"> #007AEC </color>
<color name = "accent_color"> #FFA500 </color>

5. Style Resources:
Style defined in XML files under 'res/values'.
Significance: Define reusable style for UI components.
Example:
<style name = "button_style">
<item name = "android:background"> @drawable/button
</item>
<item name = "android:textcolor"> @color/primary_
color </item>
</style>

Ganpat University | U.V. Patel College of Engineering

U.V. Patel College of Engineering

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

6. Dimension Resources:

Dimension defined in xml files under 'res/values'.

Significance: Store dimension values, ensuring a consistent layout.

Example: <dimen name="margin_large">16dp</dimen>
<dimen name="padding_medium">8dp</dimen>

7. Raw Resources:

Files stored in the 'res/raw' directory.

Significance: Store non-XML files, such as JSON data, audio:

7 How does an Android service contribute to the functionality of a mobile application? Describe the process of developing an Android Service:

- Background processing: Services allow apps to perform tasks in the background without blocking the user interface.
- Long-running operations: Services are ideal for handling operations that require more time to complete, such as playing music.
- Inter-Component Communication: Service enable components like activities, broadcast receivers and other services to communicate with each other efficiently.
- Foreground Services: Android services can run in the foreground, even when the app isn't in the foreground. This is useful for features that require ongoing user interaction, like music playback. process of Developing an Android Services:

2101201104K

- Define the Service class: Create a new Java or Kotlin class that extends the 'Service' class. Override methods like onCreate(), onDestroy(), onStartCommand() to define the behaviour of your service.
- Configure Service in manifest: Declare your service in the Android manifest.XML file to inform the Android system about its existence and configuration.
  <service android:name=".myservice"/>
- Start or Bind the Service: Decide whether you want to start your service or bind it to other components. Use startService() or bindService().
- Implement Service logic: In service class, implement the specific logic your service needs to perform its task.
- Handle lifecycle: Release resources when they are no longer needed and consider using 'stopself()' or 'stopService()'.
- Foreground Service: If your service needs to run in the foreground, 'startforeground()'.
- Testing: Thoroughly test your service to ensure it functions as expected, including handling various scenarios like network failures.
- Optimization: optimize your service for performance and resource efficiency to minimize battery usage.
- Error Handling and Logging: Implement proper error handling and logging mechanism to diagnose and address issue that may occur while service is running.

6/10/23