Name: Patel Jimil H.

En. no: 210120110084

Sub: MAD (2CEIT5PE5)

Class: 2CEIT-B

Batch: 5B4

Assignment : 1

Q.1 Basend on your understending, identify a recent bussiness trend that has influenced the Android Platform. Explain how trend impacts Android app developers and bussiness in the mobile App Industry

→ As per my knowledge, one noteable trend in the mobile app industry that was influence the Android plat from was rese of Progress Web Apps PWAs are web application that offer app-like experiences directly through web brocosers

- Impact on Android App Developers

1. Cross-Platform Compatibility: PWAs are designed to work seemlessly across various Platform and devices including Android. Developers had to Consider Creating PWAs alongside traditional Android apps to ensure broad accesibility
2. Enhanced User Experience: PWAs aimed to provide a smoother and more engaging engaging engaging user experience which set higher expectations for Android app developers These encouraged them to focus on improving the quality and performance of their apps to Complete efficiently

3. Progressive Enhanced : Developers needed to adopt progressive enhancement strategies to ensure that Android apps remained competitive by offering progressive and responsive user experiences similar to PWAs

- Impact on Bussiness in the mobile app Industry :

1. Cost savings: Bussiness could potentially save development costs by investing in a single PWA that works across multiple platforms including Android rather than building seprate
2. Increased Reach : PWAs enabled bussiness to reach a wider audience including users with Android devices without relying solely on app store Increased user acquisition
3. Improved Engagement : The focus on delivering app-like experiences through PWAs encouraged bussiness to prioritize user engagement and retention ultimately benefiting ~~their~~ 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.2 what is the purpose of an Inflater of layout in Android development and how does it fet into the architecture of Android layouts?

→ In Android development an Inflater refers to the layout Inflater 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 LayoutInflater fits into the architecture of Android 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, their placement within the UI
2. layout Inflation: when your Android app runs often defined

these XML layout files into actual view objects that can be displayed on the screen. This process is known as layout Inflator

3. Layout Inflator: Is responsible for reading the xml layout files and instantiating the corresponding view objects in memory. It takes such as TextViews Buttons etc.

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

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

6. Displaying UI: After inflation and customized the view objects can be added to the app's layout hierarchy and displayed on the screen.

Q.3 Explain the Concept of a custom dialog in Android application Provide examples to customer its use.

→ In Android application a custom dialog Box is a pop-up window

that overlays the current activity and is often used to interact with the user gather Input or

- Purpose: Custom - dialog are used when you want to present information receiver user Input or Perform actions within a Self - Contained isolated UI Element that temporary interupt
- Components: A Custom dialog typically Consists of various UI element like buttons textviews images or input fields tailored you want to facilitate
- Customization: Developers can design the dialog's appearance layout and behaviour according to these app's bandge or specific requirements This Customization in design and ~~Functionality~~ Functionality

simple ~~example~~ of creating and using a Custom dialog in Android

```
fun CustomDialog() {
    Val customDialog = Dialog(this)
    CustomDialog.setContentView
        (R.layout.Custom_dialog)
    Val messageTextview = CustomDialog
```

. findview BY ID, Textview (R.id, messegetexview)

Val OkButton = customdilog .find view BY ID <Button> (R.id. ok Button)
messageTextview.text = "This is a custom dialog"
OkButton OkButton. Set on Click listener { CustomDialog. dismess() }
CustomDialog : Show() }

→ use Cases of Custom dialog box:
login conformation dialog. settings
Inremational POP-up · media
Playbook Controls

Q.4 How do activities, services and the Android manifest file work together to make an Android app Can you describe their main to design a mobile app?

→ 1. Activities:

Role: Activities represent individual screen or UI Components in an Android app They manage the user interface and user interactions

2. Services:

Role: Services are background

that perform long running operations They can run even if the app UI not visible

3. Android manifest file:

Role: The Android manifest file declares the. It It declares the app's components with the Android system and other components

Example: In Android manifest file.xml you which activities are part of your app their launch modes permission and and services declaration This file acts as a blueprint for Android system to understand your app's structure and behaviour

→ Class mainActivity: AppCompactActivity() {
override fun onCreate (Saved Instance State: Bundle?) {
super.onCreate (Saved InstanceState)
setContentView (R.layout.activity_main)
StartServiceButton.setOnClickListener {

Val ServiceIntent = Intent (this, NotificationService::Class.java)
startService (ServiceIntent) }}}

→ Class NotificationService : IntentService ("NotificationService") {
override fun ~~onCreate (Saved~~
OnHandleIntent

```
Cintent : 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
                    (Notification manager::
                    class.java)
notification Manager :Create Notification
    Channel (channel) }
Val Builder = Notification.Compact.
                    Builder (this, ChannelID)
    .SetSmallIcon(R.drawable.ic_
                launcher_foreground)
    .Set.setContentText("This is
                notification from service.")
}}
```

How does the Android Manifest file impact the development of an 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 service several important purposes and its content significantly impact how how

the android system interacts with and manages your APP

Significence of the Android Manifest file :

- APP Configuration
- Compact declaration
- Permission S
- Intent filters
- APP Lifecycle

Example:

```
<manifest xmlns: android = "http://schemas
.android.com/apk/res/android"
    Package = "com.example.myapp">
<application
        Android: allowBackup = "true"
    android: icon = "@ic_launcher"
    android: label = "@string/app_name"
    android: roundIcon = "@mipmap/ic_
        launcher_round" launcher="launcher"
    android: supportSRti = "true"
    android: theme = "@style/APP Theme" >
<activity android: name = ".mainActivity" >
<intent - filter>
    </activity>
<activity android: name = ".secondActivity"
                    >
    ... declare additional activities here ...
</activity>
<uses - permission android: name
        = "android.Permission.
                Intent" />
... declare required Permissions
                here ...
```

</application>
</manifest>

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

→ Resources 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 create flexible, maintainable and device independent application. The various types of resources and their significance with examples

1. Layout Resources:
   - type: xml files in the 'res/layout' directory
   - significance: define the structure and appearance of the app's user interface

Examples: 'activity_main.xml' defines the layout of your main activity specifying UI components like buttons text views and their arguments

```
<Button
    android:id="@id/myButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="click me" />
```

2. Drawable Resources:
- type: 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:
- type: ~~Images~~ strings defined in xml files under 'res/values'
- significance: store text strings, making it ~~easier~~ to provide translations and maintain consistency

Example: 'res/values/strings.xml' Contains string resources

```
<string name="app_name">myApp</string>
<string name="welcom_message">
        Welcome to myApp</string>
```

4. Color Resources:
- type: Colors defined in xml files ~~under~~ under 'res/values'
- significance: store dimension values ensuring a consistent layout

Example: `res/values/dimens.xml`
defines dimension resources
<dimen name="margin_large">16dp</dimen>

7. Raw Resources:
- type: files stored in the `res/raw` directory
- significance: Store non-xml files such as JSON data, quick
- Example: Store a JSON file for app configuration

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

→ Contributions of Android services:
1. Background Processing: services allow app to perform tasks in the background without blocking the user interfaces
2. Long-running operations: services are ideal for handling operations that require more time to complete, such as playing music
3. Inter-Component Communication:

Services enable components like activities, broadcast receivers and other services to communicate with each other

efficiently

4. 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

1. 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

2. Configure service in manifest: Declare your service in the Android manifest.xml Configuration <service android:name=".myservice" />

3. Start or Bind the service: Decide whether you want to Start your service or Bind it to other Components. Use StartService() or bind Service()

4. Implement Service logic: In service class implement the specific logic your service

needs to perform its task

5. Handle Lifecycle: Release resource when they are no longer needed and consider using 'stopself()' or 'stopservice()'

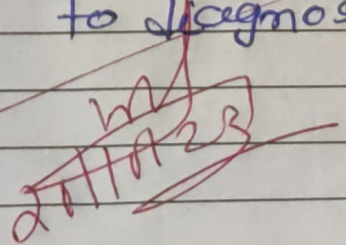
6. Interact with other Components: use use appropriate mechanism like inits broadcast or callbacks to feecilities Communication

7. foreground services (optional): If your Service needs to run in the foreground, 'start foreground()'

8. Testing: Throughly test your service Ensure it functions as expected including handling various scenarios like network failure

9. optimization: optimize your Service for performance and resource efficiency to minimize battery usage

10. Error Handling and Logging: Implement proper error handling and logging mechanism to diagnose and address is running



NSHYAM