Assignment : 1

Basend on your understanding, indentify a recent business trend that has influenced the Android Platform. Explain trend impacts Android app developers & bussiness in the mobile App Industry.

As per my knowlege, one notable trend in the mobile app industry that was influence the Android plat form was use of Progress web Apps PWAs are web application that offer app-like experiences directly through web brow.

> Impact on Android App Developers

1. Cross-Platform compatibility: PWAs are designed to work seamlessly across various platform and devices including Android Devlopers had to consider creating PWAs alongside traditional Android apps to ensure broad accessibility
2. Enhanced user Experiens: PWAs aimed to provide a smoother & more engaging user experience which set higher expectations for Am. app de. These encouraged them to focus on improving the quality and performance of their apps to complete efficiently
3. Progressive Enhanced: Developers needed to adapt progressive enhancement strategies to ensure that Android apps remained competitive by offering progressive and response user experiences similar to PWAS

SHYAM

1 Impact on Bussiness in the Mobile app Industry:
Cost saving: Business could potentially save development costs by investing in a single PWA that works across multiple platforms including Android rather than building separate
2 Increased Reach: PWAs enabled business to reach a wider audience including users with Android devies without relying soley on app store Increased user acquisition
3 Improved Engagement: The focus on delivering app-like experiences through PWAS encouraged business to Periorities user engagement & retention ultimately benefiting their mobile strategy
- Competition & Innovation: The rise of PWAS introduced competition driving business to innovate their And. apps to keep up with evolving user expectation & technology trends

2 What is the purpose of an Inflater of layout in And dev. & how does it fet into the architecture of Android layouts?
In And de. on Inflater refers to the layout Inflater which plays a crucial role in creating a user interface from xml layout files. Its primary purpose is to take an XML layout resource & convert it into a corresponding view object in memory. Here's how the layout

Inflator fits into the architecture of Android Layout
XML Layout files: In Android UI components are often defined using XML layout files. These files describe the structure & appearance of the UI, their Placement within the UI.
Layout Inflation: when your An. app runs often defined these XML layout files into actual view objects that can be displayed on the screen This proces is know as Layout Inflator.
LayoutInflator: when your responsible for reading the XML layout files and instantating the corresponding view object in memory It takes such as TextViews Buttons etc
Dynamic UI Creation: Layoutinfation is particular valuable when you need to create UI element dynamically
Binding Data: Once the vied objects are created they can be further customized and data can be bound to them
Displaying UI: After inflation and customized the view objects can be added to the app's layout hierarchy & displayed on the screen

3 Explain the concept of a custom Dialog in An. app. Provide examples to customize its use

In Android app a customee Dialog Box is a pop-up window that ovelays the current activity & is often used to interact with the usee gethee Input o&
Pu&pose: custom-dialog cae used when you want to peesent info &ecieves usee Input o& pa&fo&m actions within a self-contained isolated UI element that tempo&a&y inte&upts
Components: A custom dialog typically consists of vae. UI element like buttons textview image o& input fields tapped you want to facilitate
customization: Develope&s can design the dialog appeaeence Layout & behaviou&s according to thee app's b&andge o& specific &equi&ement
This customization in design & functionality.

```
fun CustomDialog() {
    val customDialog = Dialog(this)
    customDialog.setContentView(
        R.layout.custom_Dialog)
    val messageTextview = customDialog
    .findViewById<TextView>(R.id.messege
                            textview)
    val okButton = customDilog.findViewById
                        <Button>(
                        R.id.okButton)
    messageTextview.text = "This is a Customdialog"
    okButton.setOnClickListenee {
    customDialog.dismiss()
    }
    customDialog.show()
}
```

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.

Activities: Role: Activities represent individual screen or UI components in an And. app They manage the user interact and user interactions.

Services: Role: Services are background that perform long running operations They can run even if the app UI not visible.

Android Manifest file: Role: The Android manifest. It declares the app's components with the And. system & other components In Android/manifest file.xml you which activities are part of your app these launch modes permission and services declaration This file acts as a blueprint for Android system to understand your APP's structure and behaviour

class mainActivity: APP compact Activity() {
override fun onCreate(Saved Instance state: Bundle?) {
super.onCreate(Saved Instanstate)
set contentview(R.layout.activity_main)
startServiceButton.setonclick Listeners {
val serviceIntent=Intent(this,
Notification service::class.java)
startService(serviceIntent)}}

```
class NotificationSeevice: IntentSeevice
("Notification Seevice") {
    Oveeside fun onHandle Intent
(intent: Intent ?) {
    if (intent != Null) {
        ceeateNotification() }
Pvivate fun ceeate Notification() {
    val channelID = "My_Channel"
    if (Build.Veesion.SDK_Int >= Build.Veesion
                    _Codes.O) {
val name = "my Channel"
val notification Manageer = getSystemSeevice
        (Notification Manageer::class.Java)
notification Manageer.ceeate Notification
    channel (Chennel) }
val Buildeer = Notification.compact
        Buildeer (this, channelID)
    .setSmallIcon (R.deavable.ic
        launcheer_foreground)
    Set content Text ("This is notification
    feom Seevice") }}
```

5 How does the ~~Android Manifest~~ file impact the development of ~~an~~ Android app. Peovide an ex to demoStevate its significance.

The Android manifest file is a cevicial component in the development of an And. app. It seeves several imp. puepose & its content significantly impact hove

the android system interacts with & manage your app

Significance of the And. Manifest file:

- App configuration, compact Declaration
- Permission & Intent filters. App Lifecycle.

```
<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"
android:supportRTL="true"
android:theme="@style/AppTheme" >
<activity android:name=".mainActivity">
<intent-filter>
    <activity>
<activity android:name=".secondActivity" >
    ... Declare additional activities here...
</activity>
<uses-permission android:name
        = "android.permission
            Internet" />
... Declare required Permission here...
</application>
</manifest>
```

What is the role of resource in Android dev. ? Discuss the various types of resources & there significance in creating well-structured app provide points.

Resources Play a fundamental role in And. dev. by providing a structured way to manages assets values, layouts & other elements used in your app They create flexible maintainable & device independent application The various types of resources & their significance with

1) Layout Resources:
- type: XML files in the 'res/layout' directory
- significance: Define the structure & appearance of the apps user interface 'activity_main.xml' defines the layout of your main activity specifying UI components like buttons text views & their arguments.

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

Drawable Resources:
type: Images & drawable assets in the 'res/drawable' directory

- significence: store graphics, icons, & image used in your app
E.x: 'ic_launcher.png' is the app's launcher icon

String Resources:
- type: Strings defined in XML files under 'res/values'
- significance: store text strings, making it easier to provide translations & maintain consistency
E.x.: 'res/values/strings.XML' contains string resources

<string name = "app_name" > myAPP </string >
<string name: "welcom_message" >
welcome to my APP </string >

Color Resources:
- type: colours defined in XML files under 'res/values'
- significance: store dimension values ensuring a consistent layout
'res/values/dimens.xml'
defines dimension resources
<dimen name = "margin_large" > 16dp
</dimen >

Raw Resources:
type: files stored in the 'res/raw' directory
- significance: store non_XML files such as JSON data, quick
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 And. ser.

Background Processing: Services allow app to perform tasks in the bac. without blocking the user Interfaces

Long-running operation: Services are ideal for handling operations that require more time to complete such as playing music

Inter-component communication: Services enable components like activites, broadcast receivers & other services to communicate with each other efficiently

foreground services: Android services can run in the fore ground, even when the app isn't in the foreground This is useful for features that require ongoing user interaction like music play back process

Process of Developing an And. service

Define the service_class: Create a new Java or Kotlin class that extends the 'service' class. Override methods like onCreate(), onDestroy() or start command() to define the behavior of your service

Configure service in manifest: Declare your service in the And. Manifest.xml 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

bind Services()
Implement Service logic: In service class implement the specific logic for the service needs to perform its task
Handle Lifecycle: Release resource when they are no longer needed & consider using 'stopself()' or 'stopService()
Interact with other components: use appropriate mechanism like init's broadcast or callbacks to facilities communications.
foreground service (Optional): If your service needs to run in the foreground, start foreground()
Testing: Thoroughly test your service ensure it fun as expected including handling various scenarios like network failure
optimization: optimize your service for performance & resource efficiency to minimize battery usage
Error Handling & logging: Implement proper error handling & logging mechanism to diagnose & address issues running

6-10-23