

## Assignment :- I

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 & business in the mobile app industry.

→ One significant trend that is shaping the Android platform & the mobile app industry is the growth of Progressive Web Apps. & their impact on native Android app development. Here's an explanation of how this trend could have influenced Android app developers & business.

### \* Progressive Web Apps:

→ PWAs are web applications that offer a native app-like experience within a web browser. They leverage modern web technology to provide features such as offline access, push notifications, & a responsive design. What gets PWA's unique is their ability to function seamlessly on both desktop & mobile devices.

### \* Impact On Android App Developers:

1) Reduced Deployment Effort: Developers could choose to build PWAs instead of traditional native Android apps, reducing the need for separate codebase & resources for different platforms.

## **U. V. Patel College of Engineering**

2. **Cross-Platform Compatibility:** PWAs are designed to work on various platforms, including Android Developers can reach a broader audience with a single PWA, potentially simplifying their development and maintenance processes.
3. **Enhanced User Experience:** PWAs can provide smooth, app-like experience to users, including offline access & push notifications. Developers could focus on delivering these experiences without having to worry about platform-specific constraints.
4. **Cost Savings:** Building a PWA can be more cost-effective than developing separate native apps for multiple platforms. This can benefit businesses by reducing development & maintenance expenses.
5. **Global Reach:** PWAs can be accessed through a web browser, making them accessible to a broader audience without the need for app stores. This can help businesses reach users more easily.
6. **Improved Engagement:** The app-like features of PWAs such as push notifications, can lead to increased user engagement.
7. **Faster Updates:** Updating a PWA is often quicker & more efficient than updating native apps through

21012021053

### QPP Stores.

5. Reduced Friction: Since PWAS don't require users to install an app, they can reduce the friction associated with app download.

→ This factor want to note that while PWAS offer many advantages, they may not be suitable for all types of apps or businesses. Developers & businesses should evaluate their specific needs & target audience to determine the most suitable approach.

Q-2 What is the purpose of inflater or layout In Android development how does it fit into the architecture of android layout?

→ Layout Inflater is used to create a new view object from one of your XML layouts individually just gives you a reference to a view then has already been created.

1) Dynamic UI Creation: If android app's UI requirements & create & display UI components dynamically based on user interactions, it's a dynamic conditions.

2) Separation of concerns: Separating the UI layout definition from code that manages it helps maintain and run separation concerns.

21/01/2021 053

9. Code Reusability: Developers can reuse it.

4) Localization & Theming:

5) Efficiency:

6) Custom Views:

\* How it fits into The Architecture:

- Activity & Fragment: In Android : Activities & Fragments are responsible for managing UI Component.
- Binding Views: After inflating a layout, developer can bind views to variables in the code using method like 'findViewById'.
- Events Handling & Interactivity: Once the layout is inflated & views are bound.
- Adaptability: The Layout Inflater allows for adaptability in UI design.

Q-3: Explain the concept of a Custom Dialog Box in Android applications. Provide example to illustrate its use.

→ In Android applications, a 'CustomDialogBox' often is a UI component that allows developers to create a customized pop-up dialog window that can display information, collect user input, or perform various actions.

\* How it works:

- 1). **Layout Definition:** Developers define the layout for the custom dialog using an XML layout file.
- 2). **Dialog Creation:** In code a custom dialog is created & associated with the XML layout. Developers can set properties for the dialog, such as its size, style, & specific behaviors.
- 3). **Displaying the Dialog:** when needed, the custom dialog can be displayed to the user.
- 4). **User Interaction:** Users can interact with the elements within the custom dialog, entering text, clicking buttons.
- 5). **Handling Actions:** Developers can define actions or event handlers for the UI element within the custom dialog to respond to user input or perform specific tasks.

\* Examples:-

1). Login Dialog:

- **Purpose:** A custom dialog for user login or authentication.
- **Layout:** The dialog may contain Edit Texts for Username, and Password, a "Login" button, & forgot password.

21/01/2021 053

- **useCase:** Displays when the user needs to log in or provide authentication credentials.
  - **Confirmation Dialog:-**
    - Purpose: A dialog that tasks the user for confirmation before proceeding with a critical action.
    - Layout: Contains a message describing the actions "Yes" & "No" buttons.
    - Use Case: Used to prevent accidental actions like deleting data or making irreversible changes.
  - 3. **Custom Picker Dialog:-**
    - Purpose: A dialog with a custom picker. Such as a date picker, time picker or color picker.
    - Layout: The dialog includes the picker (dropdown) & "Cancel" buttons.
  - 4. **Information Dialog:-**
    - Purpose: Displaying informative messages or additional details.
    - Layout: Contains text or multimedia elements to convey information.
    - Use Case: Used for displaying pop-up help message, terms of service.
- Q-4
- How do activities services & the Android manifest file work together to make communication? Can you describe their main role to provide a basic example of how they cooperate for design a mobile app?

201201053

- In Android app development, Activities, Services & the Android manifest file work together to create the structure & functionality of an app.

## I. 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.
- Example: In our emails app example, we could be used to periodically ask for new emails in the background & update the inbox.

## 3). Android Manifest File:

- Role: The AndroidManifest.xml file is a configuration file that defines essential information about the app. It's component contains required permissions.
- Example: In the manifest file, you specify which activities & services the app contains, their preparation & grant permission required, among other things.

21012021053

## Q. 5 M. V. Patel College of Engineering

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 critical component in Android app development. It specifies several significant purposes that impact the development and functionality of an Android application.

### \* Examples:

1). Component Declaration: The Android Manifest File is where you declare all the components of your Android application, including activities, services, broadcast receivers & content providers.

```
<!-- Application -->
<application android:name=".MainActivity">
    <!-- Intent Filter -->
    <action android:name="android.intent.action.MAIN"
        android:category="LAUNCHER"/>
    <!-- Intent Filter -->
    <!-- Activity -->
    <service android:name=".MyService"/>
    <!-- Receiver android:name=".MyReceiver"/>
    <!-- Provider -->
    <provider android:name=".MyContentProvider"/>
    <!-- Content provider -->
    <!-- Content provider -->
```

2. Application Configuration: This includes setting the app theme, icon, label, version information & determine the minimum android API level required.

### → Application

```
Android: icon = "@drawable/appIcon"
Android: label = "GatingApp-Main"
Android: theme = "GatingAppTheme"
<application>
    <activity>
        <code> android:versionCode="1"
        <code> android:versionName="1.0">
```

3. Permission: You declare the permissions your app requires to access various system resources data, such as the camera, location intent.

```
<uses-permission android:name="android.permission.CAMERA"/>
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
```

### 4. Intent Filter:

### 5. App permission:

~~Ques~~ about 5) The role of resources in android development  
 Discuss the various types of resources & their significance in creating our android application.  
 → Resources are the additional files such as content files, your code files such as bitmaps, layout definition, wear interface strings, animation, post drawable & more.

2101201053

## U. V. Patel College of Engineering

- ② Declare The Service In The manifest.xml In The Androidmanifest.xml File, declare The Service with The name & Intent filter.  
→ Service android:name="u\_myService" />
  - ③ Implement The Service's Functionality, write The logic for The Service inside The onStart() method.
  - ④ Start The Service : To Start The Service, Create An Intent that Specifies The Services Class & call StartService() Method in The Intent.
  - ⑤ Optionally Bind To the Service:  
→ Stop Or Unbind From The Service: you can Stop A Service Using StopService(), Or Unbind From It Using unbindService()!
- ~~ANSWER~~  
~~QUESTION~~