Assignment - 9 (MAD)

Q-1 Based on your understaing Identify a recent business that has influenced the android platfrom Explain how his trend impacts android APP developers and business in the mobile APP industry

→ one significant trend in the Android APP industry was the increasing emphasis on user Privacy and data security

→ Impact on Android APP Developers:

1. Enhanced Permissions and Consent

→ Developers had to be more transparent about the data their APP called and request explicit user Content. This meant redessigning Permission dialogs and ensuring that user understood why certain data was being collected.

2. Limitations on Advertising

→ For apps relying on advertising revenue, changed in tracking and targeting due to Privacy Concern affected their monetization Strategies. Developers needed to adapt to these changes Possibly exploring alternative monetization models.

3. Data Handling and Storage

→ Developers had to review how they handled and stored user data, implementing data Protection measures. This could lead to increased

development time & costs.

1. Compilance Costs:

→ Businesses operating in the android app industry n to resources for compliance with stricter data privac regulation this could include legal and ~~th~~ technic measures to data protection.

2. Monetization Challenges

→ Businesses relying heavily on user data for advertis personalized content faced challanges in maintain their streams. They needed to find new ways to eng users a generalize uncomL.

3. Repulation management

→ Privacy breaches or mishanding of user data could result revere repulational damage. Building and maintaining trust users become even more criti

Q-2] what is purpose of a ~~In~~flator of layout in An and how does it fit into the architeture of And layout.

→ In android app Development, think of the "inflater like a tool. it helps turn your design plans into actual button boxes, and ~~other~~ things your see on your phones.

→ Purpose of Layout Inflator.

1. Dynamic UI Inflation: Layout inflater is used Create in of Android view objects from xmL Layout resource files runtime.

2. Reusability: it promotes the reusability of UI Comports defining their structure and appe in xmL Layout making it easier to instantiat and Populate them in difference parts of an AP

3. Separation of Concerns: layout inflater helps maintain separation b/w the UI ~~elements~~ design and the code that manipute and interact with these UI elements.

1) Architecture of Android Layouts.

1) XML Layout files :- Developers design the layout & structure elements in XML layout resource file.

2. Activity / fragment : In the java or kotlin code of an android activity or fragment, developers use the layout Inflater to "inflate" or parse the xml layout files creating a hireachy of view objects. This is typically done withing the 'onCreate' method.

3 View Hiearchy :- The result of infleting the layout xml is a ~~hire~~ hiearchy of view objects, with the root view being the top-level layout.

4. Data Binding & Event Handling :- Developers

otthen bind data to these views using data binding libraries or handle user interactions by attaching event listeners.

5. Rendering on the Screen :- The android system is responsible for rendering this hiearchy of views on the device screen

according to the layout specifications defined in the XML file.

Q-3 Explain the concept of custom DialogBox in A application. Provide examples to illustrate its us

→ A custom Dialog Box in Android application is a window. that developer can design and customiz to show specific information, receive input fro users or perform actions without navigating a new screen or activity. Custom Dialog Boxes are helpful for displaying messages, alert forms or any custom content in control and visually appealing manner.

1. Design flexibility: custom DialogBoxes allows developer to create unique and tailored user interface.

2. Contextual use - They are typically used when want capture user input or show information without taking the different screen.

3. User interaction: Custom Dialog Boxes can contain button fields check boxes, any other UI element allowing use interact with the content inside the dialog

→ Example of custom Dialog Box use

1) Confirmation Dialog: A Common use case is asking for confirmation before performing a critical action

2) Login or Registration Dialog: Instead of navigating to Screen for login or registration a custom box up, prompting the user to enter their credentials.

3) Error messages: when there's an error such a issues or invalid input, a custom dialog can display can display error message with details

Code:-

```
import android.app.AlertDialog
import android.app.content.DialogInterface
import android.os.Bundle
import androidx.appcompat.app.AppCompatActivity

class YourActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState:
                          Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        val builder = AlertDialog.Builder(this)
        builder.setTitle("Custom Dialog example")
        builder.setMessage("This is a custom dialog box")
        builder.setPositiveButton("OK") { dialog, which ->
            val dialog = builder.create()
            dialog.show()
        }
    }
}
```

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 how they cooperate to design a mobile app?

→ Activities, services & the Android manifest are essential components in the Android app, each with distinct role that contribute to functionality and behaviour of an app.

1. <u>Activities</u>

<u>Role</u> :- Activities represent the user interface screen of an Android app they handle interaction, display UI elements and the UI flow.

Example:- Imagine a simple note-taking app. Screen of the app such as the note list, note editing and settings can be

2. <u>Services</u>

Role :- Services run in the background and long-running or background task without a user Interface

<u>Example</u> :- In our note-taking app, you may have a service that periodically backs up to a cloud server without showing a user Interface.

3. <u>Android manifest file</u>:

- It provides essential information about the app to the Android i.e. declare the app's components, permissions and other

- <u>Example</u>:- In the manifest file, you define activities part of your app, specify permissions declare your app uses.

1) How they cooperate :-

1) Activities
- The apps starts with an activity showing a list of notes
- when the user taps on a note, another activity open display and edit the note's content.
- Users can navigate between activities using open display and edit the gestures.

2) Services
- while the user is using the app, a service runs background to periodically save the user's notes storage.
- This service doesn't have a user interface but independently to ensure duty is continuosly

3. Android manifest file
- In the manifest file, you declare the activies used in your app
- you specify permission like "INTERNET" to allow the accell the internet for cloud backup.
- The manifest file also define which activi to app launches.

<manifest xmlns: android = "http://schemas.android.com/apk/res/android"
package = "com.example.mynotes app">
<application>

<activity android:name=".MainActivity"
<intent-filter>
<action android:name="android.intent.
<category android:name="android.intent.

Q-5 How does the Android Manifest file
the development of an android app
Provider an example to demonstrate
Significance

→ The android Manifest file impacts
development by

1. Compone Declaration: Declaring app com
to define the app's structure
ex <activity android:name=".MainAc

2. App permissions: Specifying permissions fo
device resources
ex <uses-permission android:name="androi
CA

3. Intent filters: Defining how the ap
responds to external actions or request
ex. Registering to open PDF files
trapped.
<activity android:name=".PdfViewerAct
<intent-filter>
<action android:name="android.int
<category android:name="android.intent.
<data android:mimeType="application/pdf"
</intent-filter>
</activity>

Q 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 in Android development are essential component helps you create well-structured and flexible application server several purposes such as separating code from to different devices and simplifying localization are the main types of resource and their significance.

1) Layout Resource)

- XML Layout :- These define the structure and appearance apps user interface. They help keep the UI from code logic, making it easier to maintain

Example :- A layout xml file specifies how element and text fields are arranged on the screen

2) Drawable Resources.

- Images and icons :- Drawable resource store image other graphics used in your app different can be provided for diff. screen

Example: You might have separate versions for low medium & hig

3) String Resources:

· Text and String: Storing text in resource allow easy localization and updates modifying code

Example: A string resource ('app_name') app's name, which can be changed for languages.

4) Color Resours

: Colors :- By defining colors in resours, you consistent color scheme across your easily switch themes.

Example: A color resource (primary_color) the primary color used in the app's elements.

5) style Resources

Theme and styles: Styles define the appea of UI elements, making it simple to appl consistent styling across app

Example: You can create a custom style - define fonts, colors and other visual attribu

6) Dimension Resources

Size and Dimension: Storing sizes and margins in res file makes it easy to adj layouts for different screen size and

example : A dimension resource define a consistent margin size for element.

(7) Raw Resources :

~~Raw Data~~ : ~~You can store non-Compiled resources define a consistant margin size for elements~~

Raw Data : You can store non-Complied resources like audio, video, or text files in the 'res/raw' directory.

Example - Storing a JSON file in the 'raw' folder for configuration data.

8. Animation and Drawble Animation Resources.

Animation : You can define animations in XML resource file, making it simple to reuse and apply animations to UI elements.

Example : A ~~resource~~ file (fade_in.xml) can define in Animation for an Image view.

7 How does an Android service Contribute to Functionality of a mobile application? Describe Process of developing an android service.

→ An android service plays a crucial role in launch of a mobile application by allowing to run background, even when the app is not actively in use.

· Contribution of Android service

1) Background processing: services run tasks ensuring the essential function music location, tracking, or data syncing continues without disrupting the user interface.

2) long-running operation: services are ideal for that take a long time such downloading large files or performing calculation, without causing the app.

3) Foreground services: some services can run in foreground to keep the user aware of ongoing task like navigation or chat application

→ Developing an Android service:

1) Create a service class
→ Extend the 'service' class or one of its sub like 'IntentService' or 'JobService'
→ Implement the services functionality the 'onCreate' & 'onStartCommand' m

2) Declare in the manifest
→ Register your service in the AndroidManif XML file to make it accessible to the sy and other components.

3) Service Lifecycle

understand the service's lifecycle method and override them as needed.

→ service can run in three method; foreground, background or bound chuose the appropriate mode based on your app's requirement

1) Start and Stop the service

Start a service using 'start service' or bind to it using 'bind service' (Intent, S.C, int) stop a service when it's no longer needed using 'stopservice()' or 'stopself()'

Foreground Services

To create a foreground service provide a notification that informs the user about ongoing tasks.

Use 'Start foreground()' to start a service in the foreground mode.

Thread management

When performing time-consuming operations consider thread or AsyncTask to prevent blocking the main UI

Communications

use Intent extras, broadcast receivers, or interface enable communication b/w service and other app.

8] Cleanup and Resource Management.

→ Ensure that you release and stop the s
it's no longer needed to prevent unn
battery.

9] Testing

→ Thoroughly test your service to ensure it
expected, including scenarios like c
background interruptions and restart

6-10-23