Assignment - 1

1. Based on your understanding, identify a recent bussiness trends that has influenced the Android platform. Explain how this trends impacts android app developers and businesses in the mobile app industry.

→ Internet of Things App Integration.

· The Internet of Things is not new. It is all about connecting everyday devices to the internet and making them smart. This has become a need of our daily life as we depend on multiple devices and even must be connected to the internet. Further the increasing use of mobile devices in various sectors and areas has opened up countless posibilities for the IOT.

Apps for Foldable Devices

· Have you seen those cool foldable smartphones and tablet? They're gaining popularity because they give you a larger screen when needed and can be easily folded for convenience. The user demand is pushing the trend of mobile app development for foldable device.

Artificial Intelligence (AI) and Machine learning (ML)

· Artificial Intelligence and Machine learning were integrated into mobile app development to make it smarter. ~~what come to your mind~~ The Advance of Artificial Intelligence and Machine learning has encouraged the mobile app development industry to integrate different use cases

ANSHYAM

like chatbots, personalized ads, etc. Features like image recognition, face detection text and image classification, speech recognization, and other can be implemented in mobile apps.

Wearable App Integration

Wearable devices are becoming popular day by day and greatly impact the world with the presence of smartwatches trackers and fitness bands becoming increasingly common

cloud-Based Mobile Applications

Apps incorporating advanced technologies like Artificial Intelligence, Machine learning and the Internet of Things often require significant storage in a mobile device's internal memory. However, user prefere not to install large sized app on their devices.

Cross-platform Mobile development

Android and ios are the dominant mobile app operating systems. people in the mobile application widely use the OS, and due to this, it is most cost effective for the bussiness to build cross-platform app which can run on both Android and ios devices.

2 What is the purpose of an Inflator or layout in Android development, and how does it fit into the architecture of Android layouts.

→ In Android development the purpose of an Inflater is to convert an XML layout resource into corresponding view objects in your app's user interface. It's an essential part of the Android layout architecture.

→ How it fits into the architecture of Android layout.

· layout XML files : In android you define your app's UI using XML layout files. These file describe the structure and appearance of your app's user interface elements such as buttons, text field, and more.

· Resource Files : These XML layout files are stored in the res/layout directory of your Android project as resource files. They are not directly usable as view object in your code.

· LayoutInflater : This is where the LayoutInflater comes in. It's a system service that takes a layout resource ID and Inflates it, creating a corresponding hierarchy of view objects in memory.

· Activity / Fragment : Use a LayoutInflater within an Activity or Fragment to inflate layouts. Once inflated, you can access and work with the views contained in the layout within the context of your Activity or Fragment.

GHANSHYAM

UI pendering : After inflating and configuring the view objects, Android take care of rendering them on the screen based on their properties and layout rules.

3 Explain the concept of CustomDialogBox in Android applications. Provide examples to illustrate its use.

→ A CustomDialogBox in Android is a user interface element that displays a customized popup dialog to interact with the user. It allow developers to create custom layout and content within a dialog, making it suitable for various purpose like displaying additional information gathering user input or showing alerts.

```
val builder: AlertDialog.Builder = AlertDialog.Builder(context)
builder.setTitle(dialogTitle)
val customLayout : view =
LayoutInflater.from(context).inflate(R.layout.note_edit, null)
customLayout.findViewById(R.id.note_title).setText(note.title)
customLayout.findViewById(R.id.note_subTitle).setText(note.subTitle)
customLayout.findViewById(R.id.note_desc).setText(note.description)

val reminderSwitch = customLayout.findViewById(R.id.remSwitch)
val timePicker = customLayout.findViewById(R.id.remTime)
remSwitch.isChecked = note.isReminder
timePicker.hour = note.getHour()
timePicker.minute = note.getMinute()
builder.setView(customLayout)
```

~~builder.setPosition~~

```
builder.setPositiveButton(
DialogInterface.onClickListener { dialog, which -> })
{
val dialog: AlertDialog = builder.create()
dialog.show()
}
```

- Customize the dialog by adding your own UI element or layout inside the onCreateDialog method. For more complex custom dialog create layout XML file and inflate in the dialog.

- To show the custom dialog you can create an instance of your customDialogFragment and call show() on it from your activity or fragment.

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 are the user interface components of android app. Each screen or view in your app is typically represented by an activity
- Activities handle the presentation of UI elements, user interaction and navigation within app.
- Activities can start other activities to transition between screens and pass data between them.

Services

- Services are background component that perform long-running tasks or background processing without a UI.
- They are used for task that need to continue running even when the app is not in the foreground such as playing music, handling network request, or updating data in the background.
- Services can run independently of activities and can be started stopped and bond to activities as needed.

- Android Manifest File

- The android Manifest file is a configuration file that provides essential information about the app to the Android system.

· It contains declarations of all the app's components including activities, services, broadcast receivers, and permissions.
· The Manifest file define the structure of your app, its entry points, and how it interacts with other components and system features.

· Suppose you are building a simple music player app.

· You would have multiple activities each representing different screens of the app, etc.
· When the user starts the app. the Main Activity is launched. The Android Manifest file declares this activity as the app's entry point.
· To play music in the background you create a MusicService component. This service is declared in the android Manifest files and handle tasks like streaming music managing playback control and updating the notification bar with song information.
· When the user selects a song in the "Main Activity" the activity can start the MusicService to play the selected song. The activity communicate with the services to send play/pause stop commands and the service runs independently of the activity, ensuring that music continues to play even if the user switches to another activity or exists the app.
· The android Manifest file also specifies necessary permissions such as accessing storage for music files and controlling audio playback.

5 How does the Android Manifest file impact the development or an android application? provide example to demonstrate its significance.

→ App configuration : The manifest file specifies various configuration for the app, such as its package name, version, permissions and components. This helps the android system understand how to interact with and manage the app.

Permission : It define the permission the app require to access certain device features or data.

• Activities and Intents : The manifest lists the activities within the app and their relationships. It also specifies how these activities can be started using intent.

• App lifecycle : The Manifest define the app's lifecycle behaviors, like which activity to launch when the app is started and how the app should respond to certain system events, such as incoming phone calls or SMS messages.

• Suppose you are developing a photo-sharing app, In your Manifest file you would.

• Declare permision like camera and Internet to access the device's camera and upload photo online

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 enable the execution of long running operation in the background, such as downloading files monitoring sensors, or handling network request. This ensure that critical functionality can continue even if the user switches to another app or locks their devices.

Communication : Services can fecilitate communication between different parts of an app or between ~~of way for~~ component different part of an app or between multiple apps. They provide a way for components like activities and broadcast receiver to send and receive data or instruction even when they are not actively visible to the user.

Multitasking : Services help in multitasking by allowing apps to perform task concurently. For example music apps use services to play music while the user interacts with other parts of the app or their devices.

Notification : Services can create notification to keep users information about ongoing background activities, ensuring a seamless and informative user experience.

GHANSHYAM

· Developing an android service involves the following step.

Create a service class : start by creating a working class that extends the service class or its subclass.

· Define life cycle Method : override key lifecycle method such as onCreate(), onStartCommand() and onDestroy(). These method control when the service starts what it does and when it stops.

· If a component starts the service by calling startService() the service continues to run until it stops itself with stopSelf() or another component stop it by calling stopService()

· Declaring a service in the Manifest.

· you must declare all service in your application's manifest file, just as you do for activities and other components.

```
<manifest ... >
    <application ... >
        <service android:name = ". ExampleService" />
    </application>
</manifest>
```

· Define activities for various screens : the main feed, photo upload screen and user profile.
· Set launcher activity, specifying which screen should be displayed when the user taps ~~it~~ app icon.
· Specify intent filters so that when another app want to share a photo with your app it know how to do it based on the manifest information.

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

→ · Resources in Android development play a vital role in separating the presentation and content of an app from its code. They enhance maintainability, support various device configuration and improve overall development efficiency

1. Layout Resources
· XML layouts : These define the structure and appearance of your app UI elements such as buttons, text view and images etc.
example : creating a activity_main.xml layout file to design main screen of an app.

2. Drawable Resources
Drawable resources store image and icon used in your app. Different version provided for different screen

example : storing app icons, images for buttons or background graphics in drawable folders.

3. String Resources

· Text and Localization : String resources store text used in your app's UI making it easy to support multiple languages and maintain consistent branding.
Example : using string.xml to store app labels like welcome or submit or facilitate localization.

4. Color Resources

· Color resources define the colors used throughout your app, promoting consistency and enabling dynamic theming.
Example : defining primary and accent colors in color.xml for a consistent color scheme.

5. Mipmap resources

· Mipmap resources store app icons. They are use to generate launcher icons at different resolutions.
· Example : placing app icons in mipmap folders for different screen densities.

6. Raw Resources

· Raw resources store arbitrary files that your app can use. These files can be accessed using resources identifiers.
· Example : storing audio clips that your app plays.

ANSHYAM

6/10/23