* Name :- Panchal Akshay N.
* Class : CE-IT-B
* Batch : 5B-3
* Enrollment NO.:- 210120210055
* Subject :- Mobile Application Development.

Assignment :- 1

Q.1) Based on your understanding, Identify a recent business trend that has influenced by the Android Platform. Explain how this trend impacts Android APP developers and business in the mobile APP industry.

→ In Recent business trend, Android is the most Popular mobile operating system in the world. We have approx 6.378 billion mobile users in world and mostly they are using android as operating system.

→ As, Per statatics In 2022 Play Store have, 90,000 new mobile apps, It simply means that android having very big impact in recent business trend.

→ Now, this trend make big impacts on APP developers, because as Per statastics mobile and users are incresing day by day and their requirement also getting increying so it's very ~~bright~~ future for android APP developers in this mobile APP industry.

→ Some future APP developments Topic for developers.

- Augmented Reality (AR).
- IoT
- chatbots
- cloud computing
- AI

Etc....

Q.2 What is the purpose of an Inflator of in Android development? and how does it fit the architecture of android layouts?

→ In android development, an Inflater, specifically a layout Inflator, is a crucial component that used to instantiate and create instances of objects from XML layout resource files. its primary purpose to take an XML layout file turn it into a corresponding hierachy of V objects, which can then be displayed on screen.

- Here, how inflator fits into architecture Android :-

1) XML layouts :- In Android, UI layouts are defined using XML files. These layout files describe the arr and attributes of UI (Text buttons, Imageview) within an activity or fragment. It is a and convenient way to de UI.

2) Activity or fragment :- Activities and fr are the building blocks of an app. They represent individual screens or parts of the These components need to inflate the XML layout files display their UI to user

3) Layout Inflator :- This is where the la inflator comes into play. It's Essential part of the Android framework that allows you to dynamically create instances o view objects from XML layo

4] View Hierachy : The view hierarchy created by the layout inflator forms the structure of the UI for an activity or fragment. It includes UI elements defined in the xml layout file, orginal in a tree-like structure.

→ Here's basic Example how we use inflator in Android:-

```
Val Inflator = layoutInflator
val rootview = Inflater inflate (R.layout.
                activity.main, null)
Set Contentview (root view)
```

Q.3) Explain concept of custom dialouge box in Android Applications. Provide Examples to illustrate its use.

→ A custom dialogbox in androids in user interface element that allows developers to create and display a customized PopUp dialog on the screen. unlike the standard dialogs give developers complete control over the layout, apperence, and behaviour of input. It used to present information, gather user input, or perform actions without navigating to a different activity or fragment

1.) Design and Flexibility:-
→ Custom dialog boxes allows developers create unique and tailored user inter

2) Contextual use:-
→ They are typically used when you want capture input or show information wit taking the user to a different screen

3) User Interactions:
→ Custom Dialog boxes can contain buttons TextFields, check boxes or any other UI Element inside the dialog box.

- Example of custom dialog box uses:

1.) Confirmation dialog:
→ A common use case is asking the for information before performing a action.

2) Error message:-
→ When there is an error occurs as network issues or invalid, a customer dialog can display an error message with details, helping the understand and correct the prob

3) Date and time picker:-
→ you can create a custom dialog for reflecting date or time, Providi a more user-friendly way to in this information.

* Code:-

```
import android.app.AlertDialog
import android.app.contentDialogInterface
import android.os.Bundle
import android.appcompat.app.
        AppCompatActivity.

class YourActivity : AppCompatActivity() {

override fun onCreate(savedInstanceState: Bundles)
{
Super.onCreate(savedInstanceState)
Set contentView(R.layout.activity_main)

Val builder = AlertDialogBuilder(this)
builder.setTimes("Custom Dialog box
        Example")

builder.setMessages("This is a
        custom dialog")
builder.setPositionButton("OK") {dialog,
        which ->}
Val dialog = builder.create()
dialog.show()
}
```

(4) How do activities, services and the android file work together to make an android app? describe their main role and provide a h Example how they cooperate to design a mo

→ Activities, services and android manifest are essential components in the android architecture each with distinct role that bute to the functionality and behaviour an app.

1) Activities:-

→ Role:- Activities represent the use interface and screen of an a app they handle user intera display all elements and mana how

→ Example:- Imageine a simple not taking app. Each screen of the such as the note list, note note editing and settings can implemented as per activities.

2) Services:-

→ Role:- Service run in the back gro and perform long-running o background task without u

→ Example:- In our note-taking app might have service that per backs up notes to a cl server without showing

3). Android manifest File:-

→ Role:- The Android manifest file is a configuration file Provides essential information about the app. to the Android system it declares the app's components, Permission and other settings.

→ Example:- In the manifest file, you define which activities are Part of your app. Speify Permissions and declare servile your app uses.

* How they cooperate?

1). Activities:- The app starts with an activity showing a list of notes
→ when the user taps on a note, another activity opens to display and edit the note's content.
→ users can navigate betn activities using buttons or gestures.

2) Services:- while the user is using the app, a service runs in background to Periodically save the user's notes to cloud Storage.
→ This service doesn't have a UI but operats independently to Ensure data is continously back up

3) Android manifest file:- In the manifest you declare the activities and used in your app.

→ your specify permission like: to allow the app to access internet for cloud backups.

→ The manifest file also defines activity to start when the launcher.

```
<manifest xmlns:android = "https://
        android.com/apk/res"
    Package = "com.Example.mynote
  <application>

    <activity android:name = ".main
      <intent-filter>

      <action android:name = "android
                action.main

      <category android:name:"android.
                category.LAUN
```

[5] How does the Android manifest file in the development of an android application provider an example to demonstrate its significance.

→ The android manifest file impacts app development by

1) Component Declaration: Declaring app components to define the app's structure.

Ex:- <activity android:name=".mainActivity"
/>

2) App Permissions : Specifying permission for accessing device resources.

Ex: <user-Permission android:name="android.Permission.AMERA"/>

3.) Intent.filters : Defining how the app responds to external actions or requests

Ex:- Registering to open PDF files when trapped

<activity android:name=".PdfviewersActivity">
<intent-filter>
<action android:name="android.intent.action.VIEW"/>
<category android:name="android.intent.category.DEFAULT"/>
<data android:mimetype="application/pdf"/>
</intent-filter>
</activity>

Q.6) What is the role of resources in android development Discuss the various type of resources and their significance in creating well-structured application. Provide Example to clarify your Points.

→ Resources in Android development are essential components that helps you create well-structured and flexible app. They servers servel purposes, as seperating code from context adopting to different devices and simplifying localizations. Here are the main types of resources and their significance.

7.) Layout Resources:-

- XML Layouts :- There define the structure and apperence of your app UI. They help keep the UI seperate from code logic. making it easier to maintain adopt

Ex:- A Layout XML File specifies. the Element like buttons and text fields are arranged on the screen.

2.) Drawable Resources :-

- Image and Icons: Drawable resources store images icons and other graphics used in your app difference versions can be provided for different screen densities.

- Example :- You might have "ic-launcher.png" for the app. icon of seperate versions for low, medium and high density screens.

3.) String Resources:

- Text and strings :- Storing text in resource files allows for easy localization and updates without modifying code.

- Example :- A string resource ("app_name") contains the app's name, which can be changed for different language.

4) Color resources:

- Colors :- By defining colors in resources, you can maintain a consistent color schemas across your app and easily switch themes.

• Example :- A color resou... the primary color used in ... UI Elements.

5) Style Resources :-

* Themes and Styles :- Style defines the appearance of UI elements making it simple to app... consistent styling across ...

• Example :- You can create a custom (APP themes) to define font colors and other visual att...

6) Dimension Resources :-

* Sizes and dimension :- Storing size margins in res file mak... it easy to adjust layout different screen sizes an... orientations.

• Example : A dimension resources (mar... small) defines a consistent margin size for elements.

7) Raw Resources :-

* Raw data :- You can store non-compiled resources like audio, video or text in the (res/raw) directory.

• Examples :- Storing a json file in the 'raw' folder for configuration data.

8). Animations and Drawble Animation Resource :-

Animation :-

* If you can define animation in xml resource file, making it simple to reuse and apply animations in xml to UI.

• Example :- A resource file (fade-in.xml) can define a fade in animations for an image view.

7] How does an android service contribute to the functionality of a mobile application & Describe the process of developing an android services write in simple language can include main points.

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

• Contribution of Android Se

1) Background Processing :- Services run in background ensuring the ex fun's like music playback, tracking, or data sending continue without disrupting user interface.

2) long - Runtime operations.

→ Services are ideal for operation that take a long time to complete such such as downloading larg file or performing complex calcu without causing the app to freez

3) Foreground services:-

→ Some services can run in forground displaying a persistent notification to the user aware of angoing tasks like navigation or chat applications.

4) Inter - component Communication :-

→ service can communicate with other app components activity fragment through interfaces, allowing data ex and coordination.

* Developing an android servic :-

*→ 1) Create a service class:

→ Extend the 'service' class or one its subclasses like (Intent service) or 'Job Service'.

→ Implement the service's functionality within the "on create" and "on Start" Command methods

*→ 2) Declare in the manifest:

→ Register your service in the android manifest XML. file to make it accessible to the System and other Components.

*→ 3 Service Lifecycle:-

→ understand the services life cycle methods [on create, on start, on Bind, on Destroy] and override them as needed.

- Service can run in there three methods: foreground, background or Bound. Choose the appropriate mode based on your app's requirement.

* → 4) Start and stop the servile:-

→ Start a service using 'start service' or bind to it using (bind service (Intent, service connection, int)

→ stop a service when its no long needed using 'stop scrvice (Intent)' 'stopself()'.

* → (5) Foreground - services:-

→ To create a foreground service, provide a notification that 'Inform the user about ongoing tasks.

→ use 'start foreground()' to start a service in the foreground mode.

* → 6) Thread management:-

→ When Performing time consuming operations consider using workers thre or Asynctask to prevent blocking the main UI thread.

* → 7) Communications:-

→ use Intent Extras, broadcast rece or interface, to enable communicat betn service and other app components.

* → 8) clean-up and Resource manag

→ Ensure that you release resource and stop the services when it's no longer needed to prevent t unnecessary battery drain.

9) Testing:-

- Thoroughly test your service to ensure it works as expected, including scenarios, like app background ing, task interruptions and restarts.

6/10/23