PUNE INSTITUTE OF COMPUTER TECHNOLOGY, PUNE – 411043

Department of Electronics & Telecommunication Engineering

BATCH AND ROLL NO: P8 42405
EXPERIMENT NO.8
TITLE: Design a mobile application for media player.
DATE OF PERFORMANCE
DATE OF CHECKING

Title: Design a mobile application for media player.

Requirements:
1 Android studio

Theory:
An Android media player app is an application that enables users to play and manage audio or video files on their Android devices. These apps typically provide a user interface for users to browse their media library, select a file to play, and control playback using on-screen controls or physical buttons. android media player apps can support a variety of media formats, including popular formats such as MP3, MP4, and WAV, as well as less common formats such as FLAC or Ogg Vorbis. Android media player apps are typically developed using the Android SDK and can be created using languages such as Java.

Overall, Android media player apps provide users with a convenient and customizable way to enjoy their media content on their Android devices.

Here are the general steps to design an Android mobile application for a media player:

1. Define the requirements for your media player app, including features such as audio playback, video playback, playlist management, and customization options.
2. Create a wireframe or mockup of your app's UI, including the main screen, playback controls, playlist view, and any settings or customization screens.
3. Choose a development environment, such as Android Studio, and create a new project.
4. Set up the project dependencies and add any necessary libraries or frameworks, such as ExoPlayer or Media Player.
5. Implement the UI layout for your app, using XML layout files and the appropriate UI components such as TextViews, ImageViews, and Buttons.
6. Implement the app's functionality, such as audio or video playback, playlist management, and customization options, using Java code.
7. Test the app on a physical device or emulator, and make any necessary adjustments to the UI or functionality.

8. Optimize the app's performance and stability, using tools such as Android Profiler or Crashlytics.
9. Add any additional features or functionality as needed, based on user feedback or evolving requirements.
10. Publish the app to the Google Play Store or other app marketplace, following the appropriate guidelines and requirements.

When designing a media player app, it's important to consider factors such as the types of media supported, the playback controls and options, and the user interface design. It's also important to ensure that the app is optimized for performance and stability, and that it meets any relevant platform or marketplace guidelines.

Code:

Activity_main.xml

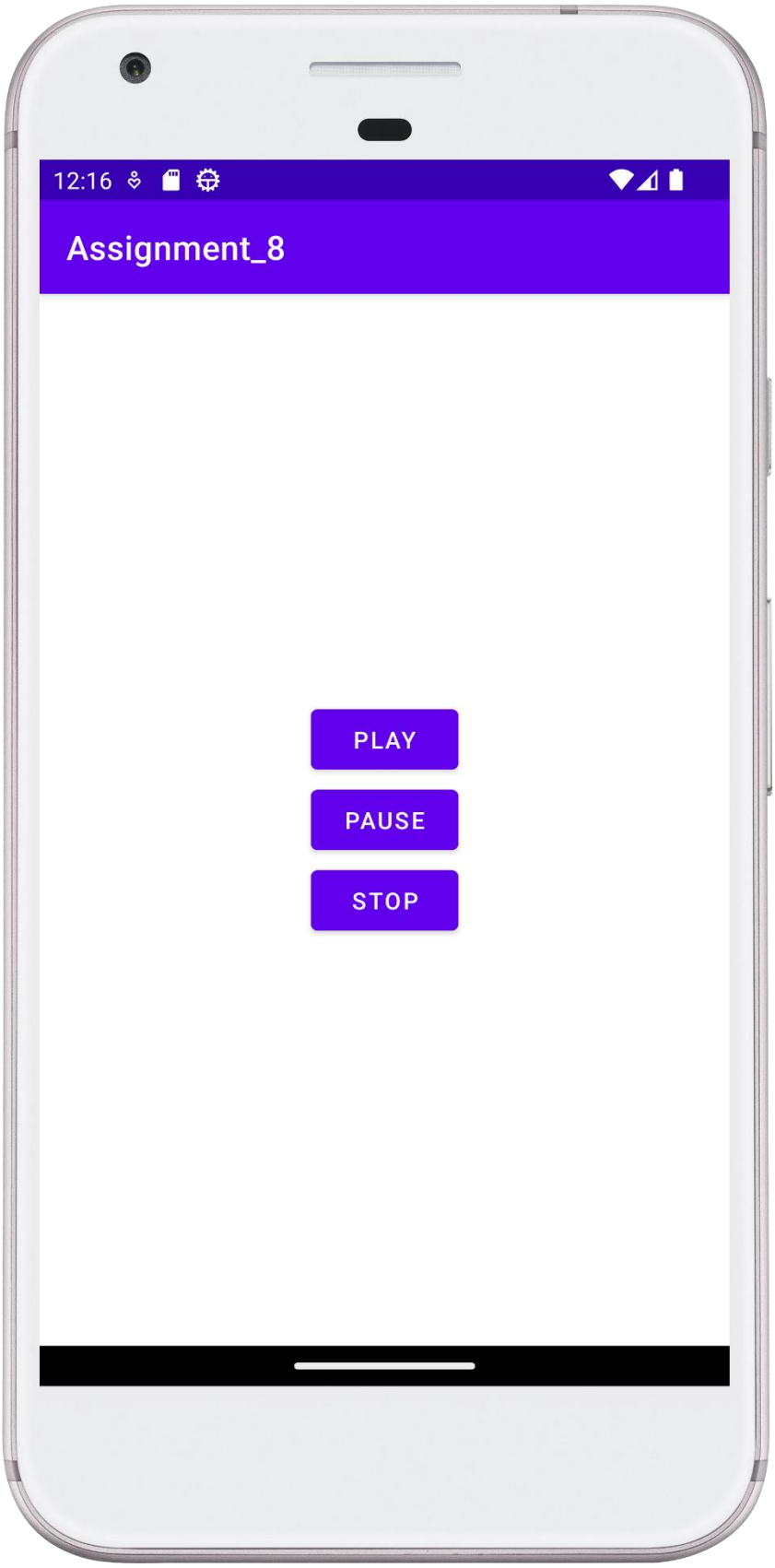
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:orientation="vertical"
    tools:context="com.example.assignment_8.MainActivity">

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="play"
        android:text="Play" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="pause"
        android:text="Pause" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="stop"
        android:text="Stop" />

</LinearLayout>
```

MainActivity.java

```
package com.example.assignment_8;
import android.media.MediaPlayer;
import android.os.Bundle;
import android.view.View;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;


public class MainActivity extends AppCompatActivity {
    MediaPlayer player;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    public void play(View v) {
```

```
        if (player == null) {
            player = MediaPlayer.create(this, R.raw.song);
            player.setOnCompletionListener(new MediaPlayer.OnCompletionListener() {
                @Override
                public void onCompletion(MediaPlayer mp) {
                    stopPlayer();
                }
            });
        }

        player.start();
    }

    public void pause(View v) {
        if (player != null) {
            player.pause();
        }
    }

    public void stop(View v) {
        stopPlayer();
    }

    private void stopPlayer() {
        if (player != null) {
            player.release();
            player = null;
            Toast.makeText(this, "MediaPlayer released", Toast.LENGTH_SHORT).show();
        }
    }

    @Override
    protected void onStop() {
        super.onStop();
        stopPlayer();
    }
}
```

Output:



CONCLUSION:

……

……

……