Bahria University,

Karachi Campus

A logo with text on it

Description automatically generated

LAB EXPERIMENT NO.

**9**

LIST OF TASKS

|  |  |
| --- | --- |
| TASK NO | OBJECTIVE |
| **1** | **Create a** **Background Music Player Service:** |
| 2 | **Create a File Download Service:** |
|  |  |

Submitted On:

13 June2024

\_\_\_\_\_\_\_\_\_\_\_\_

(Date: DD/MM/YY)

**Task 1:**

**Create a** **Background Music Player Service**

**Main.XML code**

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".MainActivity">

<Button

android:id="@+id/playButton"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Play Music"

android:layout\_centerInParent="true" />

<Button

android:id="@+id/stopButton"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Stop Music"

android:layout\_below="@id/playButton"

android:layout\_marginTop="20dp"

android:layout\_centerHorizontal="true" />

</RelativeLayout>

**Main.Java code**

package com.example.lab;

import android.content.Intent;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

private Button playButton;

private Button stopButton;

@Override

Intent intent = new Intent(MainActivity.this, MusicService.class);

stopService(intent);

}

}):}}

**Music service.java**

package com.example.Lab;

import android.app.Notification;

import android.app.NotificationChannel;

import android.app.NotificationManager;

import android.app.Service;

public class MusicService extends Service {

private MediaPlayer mediaPlayer;

private static final String CHANNEL\_ID = "MusicServiceChannel";

@Override

public void onCreate() {

super.onCreate();

mediaPlayer = MediaPlayer.create(this, R.raw.sample\_music); // Replace with your music file

mediaPlayer.setLooping(true):}

@Override

public int onStartCommand(Intent intent, int flags, int startId) {

createNotificationChannel();

Notification notification = new NotificationCompat.Builder(this, CHANNEL\_ID)

.setContentTitle("Music Player")

.setContentText("Playing Music")

.setSmallIcon(R.drawable.ic\_music\_note) // Replace with your app icon

.build();

startForeground(1, notification);

"Music Service Channel",

NotificationManager.IMPORTANCE\_DEFAULT

);

NotificationManager manager = getSystemService(NotificationManager.class);

if (manager != null) {

manager.createNotificationChannel(serviceChannel);

}}}}

**Manifest code**

<?xml version="1.0" encoding="utf-8"?>

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

package="com.example.assignment2">

<application

android:allowBackup="true"

android:icon="@mipmap/ic\_launcher"

android:label="@string/app\_name"

android:roundIcon="@mipmap/ic\_launcher\_round"

android:supportsRtl="true"

android:theme="@style/Base.Theme.Assignment2">

>

<service android:name=".MusicService"

android:foregroundServiceType="mediaPlayback"

tools:ignore="ForegroundServicePermission" />

<activity android:name=".MainActivity"

android:exported="true">

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />

</intent-filter>

</activity>

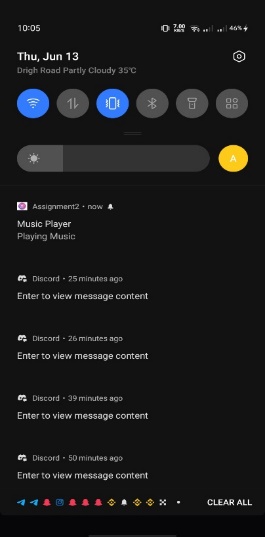
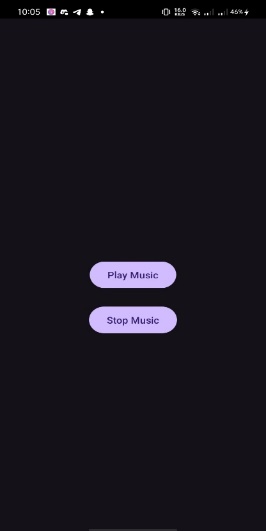
</application>

<uses-permission android:name="android.permission.READ\_EXTERNAL\_STORAGE"

/>

<uses-permission android:name="android.permission.FOREGROUND\_SERVICE" />

</manifest>

**OUTPUT:**

**Task 2:**

**Create a File Download Service:**

**Main.XML code**

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".MainActivity">

<ListView

android:id="@+id/fileListView"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_alignParentTop="true"

android:layout\_marginTop="20dp" />

<TextView

android:id="@+id/downloadStatusTextView"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Download Status"

android:layout\_below="@id/fileListView"

android:layout\_centerHorizontal="true"

android:layout\_marginTop="20dp"

android:textSize="18sp" />

</RelativeLayout>

**MainJava code**

package com.example.Lab;

import android.Manifest;

import android.content.BroadcastReceiver;

import android.content.Context;

import android.content.Intent;

import android.content.IntentFilter;

public class MainActivity extends AppCompatActivity {

private static final int REQUEST\_PERMISSION\_WRITE = 1001;

private ListView fileListView;

private TextView downloadStatusTextView;

private String[] files = {"file1.pdf", "file2.mp4", "file3.jpg"};

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

fileListView = findViewById(R.id.fileListView);

downloadStatusTextView = findViewById(R.id.downloadStatusTextView);

ArrayAdapter<String> adapter = new ArrayAdapter<>(this, android.R.layout.simple\_list\_item\_1, files);

fileListView.setAdapter(adapter);

fileListView.setOnItemClickListener(new AdapterView.OnItemClickListener() {

@Override

public void onItemClick(AdapterView<?> parent, View view, int position, long id) {

if (ContextCompat.checkSelfPermission(MainActivity.this,

Manifest.permission.WRITE\_EXTERNAL\_STORAGE) != PackageManager.PERMISSION\_GRANTED) {

ActivityCompat.requestPermissions(MainActivity.this,

new String[]{Manifest.permission.WRITE\_EXTERNAL\_STORAGE}, REQUEST\_PERMISSION\_WRITE);

} else {

startDownloadService(files[position]);

}

}

});

// Register receiver to get updates from the service

registerReceiver(downloadReceiver, new IntentFilter(DownloadService.ACTION\_UPDATE));

}

downloadStatusTextView.setText("Permission denied to write to storage");

}

}

}

}

**DownloadService.java code**

package com.example.lab;

import android.app.Notification;

import android.app.NotificationChannel;

import android.app.NotificationManager;

import android.app.Service;

import android.content.Intent;

import android.os.Build;

import android.os.Environment;

import android.os.IBinder;

import androidx.annotation.Nullable;

import androidx.core.app.NotificationCompat;

import java.io.BufferedInputStream;

import java.io.FileOutputStream;

import java.io.InputStream;

import java.net.HttpURLConnection;

import java.net.URL;

switch (fileName) {

case "file1.pdf":

fileURL = "https://www.w3.org/WAI/ER/tests/xhtml/testfiles/resources/pdf/dummy.pdf";

break;

case "file2.mp4":

fileURL = "http://techslides.com/demos/sample-videos/small.mp4";

break;

case "file3.jpg":

fileURL = "https://via.placeholder.com/150";

break;

}

URL url = new URL(fileURL);

HttpURLConnection connection = (HttpURLConnection) url.openConnection();

}

**Manifest code**

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

package="com.example.Lab ">

<uses-permission android:name="android.permission.INTERNET" />

<uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE" />

<uses-permission android:name="android.permission.FOREGROUND\_SERVICE" />

<application

android:allowBackup="true"

android:icon="@mipmap/ic\_launcher"

android:label="@string/app\_name"

android:roundIcon="@mipmap/ic\_launcher\_round"

android:supportsRtl="true"

android:theme="@style/Base.Theme.Assignment2">

<service

android:name=".DownloadService"

android:foregroundServiceType="dataSync"

tools:ignore="ForegroundServicePermission" />

<activity android:name=".MainActivity"

android:exported="true">

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />

</intent-filter>

</activity>

</application>

</manifest>

**OUTPUT:**

