**A**

**Project Report on**

**MUSIC PLAYER**

Submitted by**:**

**Abhishek Kumar Das**

**Roll No : 2004000629126016**

*for the partial fulfillment of the requirements for the degree of*

**Bachelors of Computer Applications (BCA)**

****

**Lakshya Institute Of Technology**

**M4/46,Acharya Vihar, Near Water Tank**

**Bhubaneswar, Odisha-751013**

**INTRODUCTION**

Android is open source code mobile phone operating system that comes out by Google. Music player in this, project is an application based software based on Google Android. Music is one of the best ways to relive pressure in stressful modern society life. The purpose of this project is to develop a player which can play the mainstream file format. To browse and query the storage space as well as operation of playing can be realised. Meanwhile, this software can play, pause and select songs with latest button and next button according to sets requirement as well as set up songs .

Xml files are used to create the front end design like AndroidMainfest.xml , activity\_main.xml and many more for designing the complete front end of my project.

PROBLEM STATEMENT

There are many music player applications are present in the market. Here the question arises what does our application contains that other application don’t contain? So, here’s the answer that our application is ad free not like other paid applications which requires to be paid to block the ads or else we can say that via paying the application or buying premium membership we can be ad free, but in my application it is free of cost as the user do not need to pay any money for blocking ads or for premium membership .

As my application is a trial once so it is completely offline. A user do not need internet or network connection to access my application. We do got a demerit that user cannot download the song within the application. He / She can only play the songs present in the his / her device.

GOAL OF THE PROPOSED SYSTEM

So talking about my main goal to develop this application is to provide the client / customer a ad free platform to play music without any interference of the sudden ads during the playtime of songs. My goal is to provide best quality music like spotify and wynk with low space consumption and more features compared to other applications present in the current market.

HARDWARE SPECIFICATIONS

My System Specifications:

🡪CORE i5 8th Gen Processor

🡪1GB NVIDIA GEFORCE Graphics Card

🡪8GB RAM

🡪128GB SSD

Minimum System Specifications:

🡪CORE i3 5th Gen Processor

🡪500MB Integrated Graphics Card

🡪4GB RAM

🡪No SSD required

SOFTWARE SPECIFICATIONS

The required software to develop this application are –

🡪Operating System: Windows 10/11, Linux

🡪Software: Android Studio & SDK(Software Development Kit), ADT(Android Development Tool )

🡪JDK: Java Runtime Environment Virtual Machine, Java Development Kit

(JDK)

🡪Emulator: Bluestacks 5(if using any)

Installation Steps of the developing environment:

🡪Step-1: install the Java virtual machine JDK version-7

🡪Step-2: install the Android Studio & SDK: first download the Android SDK

🡪Download Address: <https://developer.android.com/studio>

🡪Input SDK tools path in the SDK location: D:\android\software\android SDK- Windows and click OK.

🡪The Android environment is setup successfully .

DATA FLOW DIAGRAM(DFD)

SONG MANAGEMENT

USER MANAGEMENT

SONG TYPE SELECTION

UPLOAD SONG

CATEGORY MANAGEMENT

LEVEL-0 DFD For My Music Player Software

User Input For Taking Instruction

Entering Into From User To Go to

System The Next Step

USER

Show Result Carrying Instruction

From Menu

LEVEL-I DFD For My Music Player Software

CODING

package com.example.project1;  
  
import android.Manifest;  
import android.content.Intent;  
import android.os.Bundle;  
import android.os.Environment;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.AdapterView;  
import android.widget.BaseAdapter;  
import android.widget.ListView;  
import android.widget.TextView;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import com.karumi.dexter.Dexter;  
import com.karumi.dexter.PermissionToken;  
import com.karumi.dexter.listener.PermissionDeniedResponse;  
import com.karumi.dexter.listener.PermissionGrantedResponse;  
import com.karumi.dexter.listener.PermissionRequest;  
import com.karumi.dexter.listener.single.PermissionListener;  
  
import java.io.File;  
import java.util.ArrayList;  
  
public class MainActivity extends AppCompatActivity {  
 ListView listView;  
 String[] items;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 listView = findViewById(R.id.*listView*);  
 runtimePermission();  
 }  
  
  
 private void runtimePermission() {  
 Dexter.*withContext*(this).withPermission(Manifest.permission.*READ\_EXTERNAL\_STORAGE*)  
 .withListener(new PermissionListener() {  
 @Override  
 public void onPermissionGranted(PermissionGrantedResponse permissionGrantedResponse) {  
 displaySongs();  
 }  
  
 @Override  
 public void onPermissionDenied(PermissionDeniedResponse permissionDeniedResponse) {  
  
 }  
  
 @Override  
 public void

onPermissionRationaleShouldBeShown(PermissionRequest permissionRequest, PermissionToken permissionToken) {  
 permissionToken.continuePermissionRequest();  
 }  
 }).check();  
 }  
  
 public ArrayList<File> findSongs(File file) {  
 ArrayList<File> arrayList = new ArrayList<>();  
 File[] files = file.listFiles();  
 for (File singlefile : files) {  
 if (singlefile.isDirectory() && !singlefile.isHidden()) {  
 arrayList.addAll(findSongs(singlefile));  
 } else {  
 if (singlefile.getName().endsWith(".mp3") || singlefile.getName().endsWith(".wav")) {  
 arrayList.add(singlefile);  
 }  
 }  
 }  
 return arrayList;  
 }  
  
 public void displaySongs() {  
 final ArrayList<File> mySongs = findSongs(Environment.*getExternalStorageDirectory*());  
  
 items = new String[mySongs.size()];  
 for (int i = 0; i < mySongs.size(); i++) {  
 items[i] = mySongs.get(i).getName().replace(".mp3", "").replace(".wav", "");  
  
 }  
  
 /\*ArrayAdapter<String> myAdapter = new ArrayAdapter<String>(this, android.R.layout.simple\_list\_item\_1, items);  
 listView.setAdapter(myAdapter);\*/  
  
 CustomAdapter customAdapter= new CustomAdapter();  
 listView.setAdapter(customAdapter);  
  
 listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
 @Override  
 public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {  
 String songName = (String) listView.getItemAtPosition(i);  
 startActivity(new Intent(getApplicationContext(), PlayerActivity.class)  
 .putExtra("songs",mySongs)  
 .putExtra("songsname",songName)  
 .putExtra("pos",i)  
 );  
 }  
 });  
  
 }  
 class CustomAdapter extends BaseAdapter{  
  
 @Override  
 public int getCount() {  
 return items.length;  
 }  
  
 @Override  
 public Object getItem(int i) {  
 return null;  
 }  
  
 @Override  
 public long getItemId(int i) {  
 return 0;  
 }  
  
 @Override  
 public View getView(int i, View view, ViewGroup viewGroup) {  
  
 View myView = getLayoutInflater().inflate(R.layout.*list\_item*,null);  
 TextView textView = myView.findViewById(R.id.*txtsongname*);  
 textView.setSelected(true);  
 textView.setText(items[i]);  
 return myView;  
 }  
 }  
}

TESTING

Testing for this application was done on an emulator device called as bluestacks version 5 and also it was tested on my mobile device. During the testing many errors occurred like methods not declared , class not defined , method not found, image not present and many more. For these problems I had to rectify all those problems and declare all the classes perfectly , check whether all the methods are within the class or outside the class. And with the images first I had to change the image format from .jpeg to .png .

CONCLUSION