17145546

Deniz Arikoglu

Contents

[Design 3](#_Toc26460233)

[Implementation 6](#_Toc26460234)

[AndroidManifest.xml 6](#_Toc26460235)

[activity\_add\_artist\_track.xml 6](#_Toc26460236)

[activity\_delete\_update\_artist.xml 8](#_Toc26460237)

[activity\_delete\_update\_track.xml 9](#_Toc26460238)

[activity\_listview\_artist.xml 10](#_Toc26460239)

[activity\_listview\_track.xml 11](#_Toc26460240)

[activity\_main.xml 11](#_Toc26460241)

[Artist.JAVA 12](#_Toc26460242)

[ArtistAdapter.JAVA 12](#_Toc26460243)

[dbHelper.JAVA 13](#_Toc26460244)

[DeleteUpdateArtistActivity.JAVA 16](#_Toc26460245)

[DeleteUpdateTrackActivity.JAVA 17](#_Toc26460246)

[MainActivity.JAVA 18](#_Toc26460247)

[NewArtistTrack.JAVA 18](#_Toc26460248)

[Track.JAVA 19](#_Toc26460249)

[TrackAdapter.JAVA 20](#_Toc26460250)

[Testing 21](#_Toc26460251)

[Test 1 21](#_Toc26460252)

[Test 2 21](#_Toc26460253)

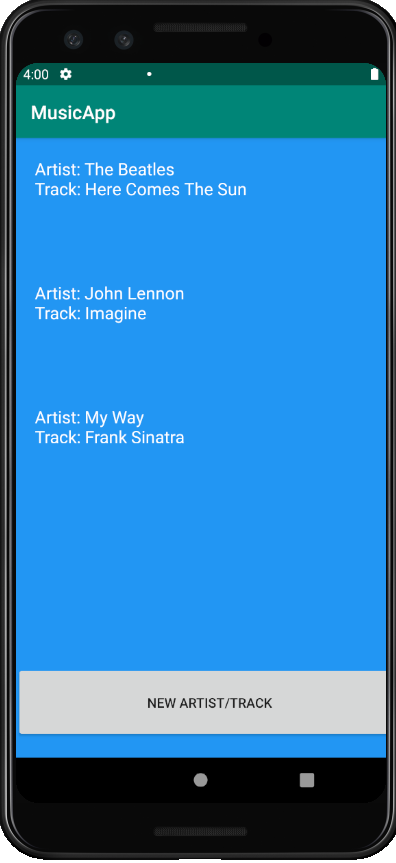
[Test 3 22](#_Toc26460254)

[Test 6 24](#_Toc26460255)

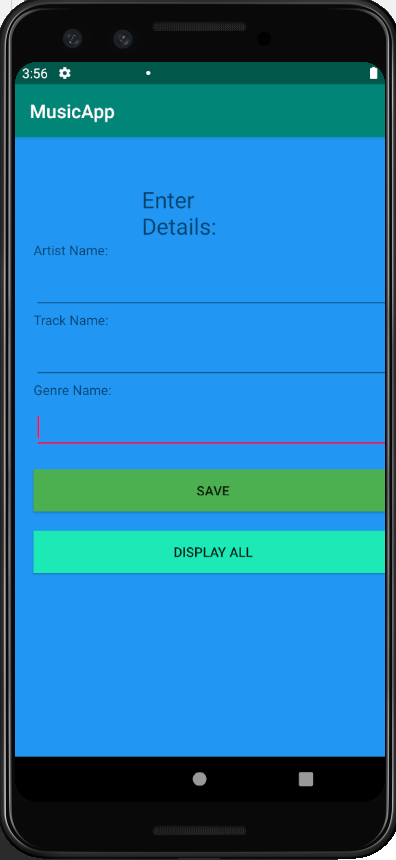
[Test 7 24](#_Toc26460256)

[Test 8 25](#_Toc26460257)

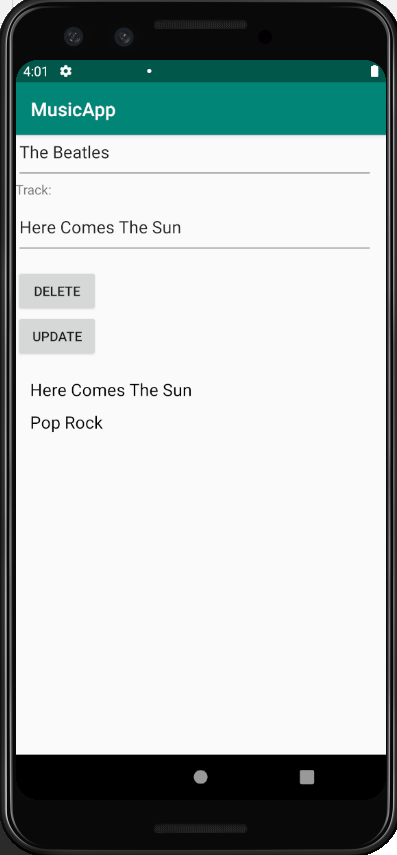
# Design

Main Activity:

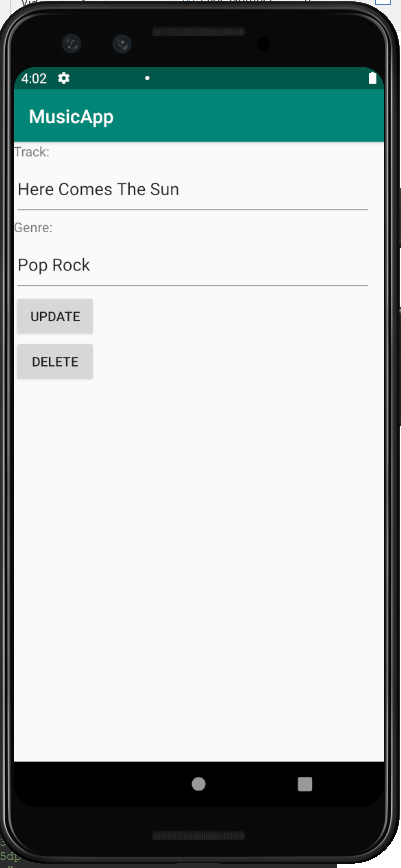
In the main activity I have a RecyclerView and a Button. The RecyclerView shows stored data in the Artist table showing us the Artist name and the track name. There is also an on click listener for the RecyclerView. The button is to add a new track/artist into this table.

Activity Add Artist/Track:

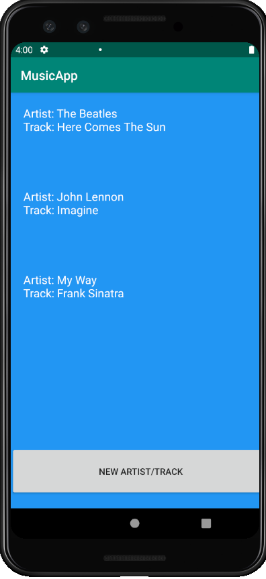
In this activity I have three Text Views to get the artist name, the track name and also the genre of the song. With two buttons, one to save the data in the text views to the database and one to go back to the main activity to show us the RecyclerView.

Activity Update/Delete Artist:

Once you click on an item in the recycler view in the main activity you reach this screen which allows you to update the Artist name or track name or even just delete the whole record. Under the buttons to delete and update a record is another RecyclerView that shows us the track name and the genre under it which also has an on click listener on It.

Activity Update/Delete Track:

Once the Recycler view in the previous activity is clicked on this activity is launched. This activity allows us to delete and update the track name or even the genre of a song.



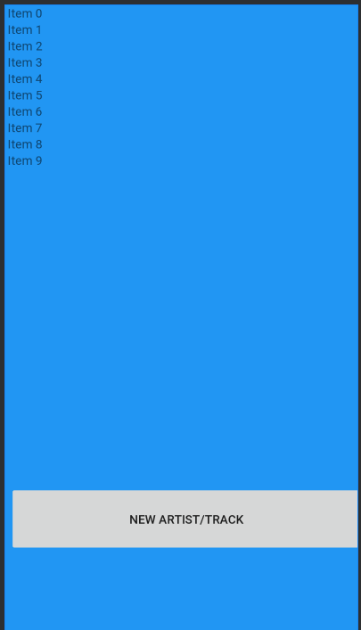
artist\_database.db

artistAdapter

Artist Table

Track Table

trackAdapter



RecyclerView ListView for both Track/Artist single item

My RecyclerView stores every item according to the formatting of the listviews I have for each table.

# Implementation

## AndroidManifest.xml

<?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.assignment1">  
  
 <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/AppTheme"  
 tools:ignore="AllowBackup,GoogleAppIndexingWarning">  
 <activity android:name=".MainActivity">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
 </activity>  
  
 <activity android:name=".NewArtistTrack"/>  
 <activity android:name=".DeleteUpdateArtistActivity"/>  
 <activity android:name=".DeleteUpdateTrackActivity"/>  
  
  
 </application>  
  
</manifest>

## activity\_add\_artist\_track.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:background="#2196F3"  
 android:orientation="vertical"  
 tools:context=".NewArtistTrack">  
  
 <TextView  
 android:id="@+id/textView2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="135dp"  
 android:layout\_marginLeft="135dp"  
 android:layout\_marginTop="50dp"  
 android:layout\_marginEnd="135dp"  
 android:layout\_marginRight="135dp"  
 android:text="Enter Details:"  
 android:textSize="24sp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <TextView  
 android:id="@+id/tvArtistName"  
 android:layout\_width="83dp"  
 android:layout\_height="26dp"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginLeft="20dp"  
 android:text="Artist Name:"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/textView2" />  
  
 <EditText  
 android:id="@+id/etArtistName"  
 android:layout\_width="380dp"  
 android:layout\_height="48dp"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginLeft="20dp"  
 android:layout\_marginEnd="20dp"  
 android:layout\_marginRight="20dp"  
 android:ems="10"  
 android:inputType="textPersonName"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/tvArtistName" />  
  
 <TextView  
 android:id="@+id/tvTrackName"  
 android:layout\_width="83dp"  
 android:layout\_height="26dp"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginLeft="20dp"  
 android:text="Track Name:"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/etArtistName" />  
  
 <EditText  
 android:id="@+id/etTrackName"  
 android:layout\_width="380dp"  
 android:layout\_height="48dp"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginLeft="20dp"  
 android:layout\_marginEnd="20dp"  
 android:layout\_marginRight="20dp"  
 android:ems="10"  
 android:inputType="textPersonName"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/tvTrackName" />  
 <TextView  
 android:id="@+id/tvGenreName"  
 android:layout\_width="83dp"  
 android:layout\_height="26dp"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginLeft="20dp"  
 android:text="Genre Name:"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/etGenreName" />  
  
 <EditText  
 android:id="@+id/etGenreName"  
 android:layout\_width="380dp"  
 android:layout\_height="48dp"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginLeft="20dp"  
 android:layout\_marginEnd="20dp"  
 android:layout\_marginRight="20dp"  
 android:ems="10"  
 android:inputType="textPersonName"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 android:text="" />  
  
 <Button  
 android:id="@+id/btnSave"  
 android:layout\_width="381dp"  
 android:layout\_height="45dp"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginLeft="20dp"  
 android:layout\_marginTop="20dp"  
 android:layout\_marginEnd="20dp"  
 android:layout\_marginRight="20dp"  
 android:background="#4CAF50"  
 android:onClick="btnSaveOnClick"  
 android:text="Save"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/etTrackName" />  
  
 <Button  
 android:id="@+id/btnDisplayAll"  
 android:layout\_width="381dp"  
 android:layout\_height="45dp"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginLeft="20dp"  
 android:layout\_marginTop="20dp"  
 android:layout\_marginEnd="20dp"  
 android:layout\_marginRight="20dp"  
 android:background="#1DE9B6"  
 android:onClick="btnDisplayAllOnClick"  
 android:text="Display All"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/btnSave" />  
  
  
  
</LinearLayout>

## activity\_delete\_update\_artist.xml

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout 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:id="@+id/linearLayout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:weightSum="100"  
 tools:context=".DeleteUpdateArtistActivity">  
  
 <EditText  
 android:id="@+id/etTrackName"  
 android:layout\_width="380dp"  
 android:layout\_height="61dp"  
 android:layout\_marginBottom="13dp"  
 android:ems="10"  
 android:inputType="textPersonName"  
 app:layout\_constraintBottom\_toTopOf="@+id/btnDelete"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintStart\_toStartOf="parent" />  
  
 <EditText  
 android:id="@+id/etArtistName"  
 android:layout\_width="380dp"  
 android:layout\_height="61dp"  
 android:ems="10"  
 android:inputType="textPersonName"  
 app:layout\_constraintBottom\_toTopOf="@+id/tvTrackName"  
 app:layout\_constraintStart\_toStartOf="parent" />  
  
 <TextView  
 android:id="@+id/tvArtistName"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Artist:"  
 app:layout\_constraintBottom\_toTopOf="@+id/etArtistName"  
 app:layout\_constraintStart\_toStartOf="parent" />  
  
 <TextView  
 android:id="@+id/tvTrackName"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Track:"  
 app:layout\_constraintBottom\_toTopOf="@+id/etTrackName"  
 app:layout\_constraintStart\_toStartOf="parent" />  
  
 <Button  
 android:id="@+id/btnDelete"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:onClick="btnDeleteOnClick"  
 android:text="DELETE"  
 app:layout\_constraintBottom\_toTopOf="@+id/btnUpdate"  
 app:layout\_constraintStart\_toStartOf="parent" />  
  
 <Button  
 android:id="@+id/btnUpdate"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:onClick="btnUpdateOnClick"  
 android:text="UPDATE"  
 app:layout\_constraintBottom\_toTopOf="@+id/trackRecycler"  
 app:layout\_constraintStart\_toStartOf="parent" />  
  
 <androidx.recyclerview.widget.RecyclerView  
 android:id="@+id/trackRecycler"  
 android:layout\_width="404dp"  
 android:layout\_height="420dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent" />  
  
  
</androidx.constraintlayout.widget.ConstraintLayout>

## activity\_delete\_update\_track.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="wrap\_content"  
 android:orientation="vertical">  
  
 <TextView  
 android:id="@+id/tvTrackName"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Track:"  
 app:layout\_constraintBottom\_toTopOf="@+id/etTrackName"  
 app:layout\_constraintStart\_toStartOf="parent" />  
  
 <EditText  
 android:id="@+id/etTrackName"  
 android:layout\_width="380dp"  
 android:layout\_height="61dp"  
 android:ems="10"  
 android:inputType="textPersonName"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 tools:layout\_editor\_absoluteY="138dp" />  
  
 <TextView  
 android:id="@+id/tvGenreName"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Genre:"  
 app:layout\_constraintBottom\_toTopOf="@+id/etGenreName"  
 app:layout\_constraintStart\_toStartOf="parent" />  
  
 <EditText  
 android:id="@+id/etGenreName"  
 android:layout\_width="380dp"  
 android:layout\_height="61dp"  
 android:ems="10"  
 android:inputType="textPersonName"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 tools:layout\_editor\_absoluteY="138dp" />  
  
 <Button  
 android:id="@+id/btnUpdate"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:onClick="btnUpdateOnClickt"  
 android:text="UPDATE"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent" />  
  
 <Button  
 android:id="@+id/btnDelete"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:onClick="btnDeleteOnClickt"  
 android:text="DELETE"  
 app:layout\_constraintBottom\_toTopOf="@+id/btnUpdate"  
 app:layout\_constraintStart\_toStartOf="parent" />  
  
</LinearLayout>

## activity\_listview\_artist.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:orientation="vertical" android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/parentLayout"  
 >  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:padding="10dp"  
 android:orientation="vertical">  
  
 <TextView  
 android:id="@+id/artistName"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textColor="#fff"  
 android:layout\_marginTop="10dp"  
 android:gravity="center\_vertical"  
 android:paddingLeft="10dp"  
 android:textAppearance="?android:attr/textAppearanceMedium"  
 />  
  
 <TextView  
 android:id="@+id/trackName"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textColor="#fff"  
 android:layout\_marginTop="10dp"  
 android:gravity="center\_vertical"  
 android:paddingLeft="10dp"  
 android:textAppearance="?android:attr/textAppearanceMedium" />  
  
 </LinearLayout>  
  
</LinearLayout>

## activity\_listview\_track.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:orientation="vertical" android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:id="@+id/parentLayout"  
 >  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:padding="10dp"  
 android:orientation="vertical">  
  
  
 <TextView  
 android:id="@+id/trackName"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textColor="#000"  
 android:layout\_marginTop="10dp"  
 android:gravity="center\_vertical"  
 android:paddingLeft="10dp"  
 android:textAppearance="?android:attr/textAppearanceMedium" />  
 <TextView  
 android:id="@+id/genreName"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textColor="#000"  
 android:layout\_marginTop="10dp"  
 android:gravity="center\_vertical"  
 android:paddingLeft="10dp"  
 android:textAppearance="?android:attr/textAppearanceMedium" />  
 </LinearLayout>  
  
</LinearLayout>

## activity\_main.xml

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout 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:background="#2196F3"  
tools:context=".MainActivity">  
  
<androidx.recyclerview.widget.RecyclerView  
 android:id="@+id/artistRecycler"  
 android:layout\_width="421dp"  
 android:layout\_height="560dp"  
 app:layout\_constraintTop\_toTopOf="parent"  
 tools:layout\_editor\_absoluteX="4dp" />  
  
<Button  
 android:id="@+id/btnNew"  
 android:layout\_width="412dp"  
 android:layout\_height="78dp"  
 android:onClick="btnNewArtistTrack"  
 android:text="New Artist/Track"  
 app:layout\_constraintTop\_toBottomOf="@+id/artistRecycler"  
 tools:layout\_editor\_absoluteX="6dp" />  
  
  
</androidx.constraintlayout.widget.ConstraintLayout>

## Artist.JAVA

package com.example.assignment1;  
  
import java.io.Serializable;  
  
public class Artist implements Serializable{  
 private String artistName;  
 private String trackName;  
 private int id;  
  
 public int getId(){  
 return id;  
 }  
  
 public void setId(int id){  
 this.id = id;  
 }  
  
 public String getArtistname(){  
 return artistName;  
 }  
  
 public void setArtistName(String artistName){  
 this.artistName = artistName;  
 }  
  
 public String getTrackName(){  
 return trackName;  
 }  
  
 public void setTrackName (String trackName){  
 this.trackName = trackName;  
 }  
  
 public String toString(){  
 String s = "Artist: " + artistName + "\n";  
 s += "Track: " + trackName + "\n";  
 return s;  
 }  
}

Artist.JAVA is a class that creates the database.

## ArtistAdapter.JAVA

package com.example.assignment1;  
  
import android.content.Context;  
import android.content.Intent;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.TextView;  
import androidx.annotation.NonNull;  
import androidx.recyclerview.widget.RecyclerView;  
import java.util.ArrayList;  
  
public class ArtistAdapter extends RecyclerView.Adapter<ArtistAdapter.ArtistViewHolder>{  
 private ArrayList<Artist> artistArrayList;  
 private LayoutInflater mInflater;  
 public ArtistAdapter(Context context, ArrayList<Artist>artistList){  
 mInflater = LayoutInflater.*from*(context);  
 this.artistArrayList = artistList;  
 }  
 @NonNull  
 @Override  
 public ArtistAdapter.ArtistViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {  
 View view = mInflater.inflate(R.layout.*activity\_listview\_artist*, parent, false);  
 return new ArtistViewHolder(view, this);  
 }  
  
 @Override  
 public void onBindViewHolder(@NonNull ArtistAdapter.ArtistViewHolder holder, int position) {  
 String mCurrent = String.*valueOf*(artistArrayList.get(position));  
 holder.artistItemView.setText(mCurrent);  
 }  
  
 @Override  
 public int getItemCount() {  
 return artistArrayList.size();  
 }  
  
 public class ArtistViewHolder extends RecyclerView.ViewHolder {  
  
 public final TextView artistItemView;  
 final ArtistAdapter mAdapter;  
 public ArtistViewHolder(View itemView, ArtistAdapter adapter){  
 super(itemView);  
 artistItemView = itemView.findViewById(R.id.*artistName*);  
 this.mAdapter = adapter;  
 artistItemView.setOnClickListener(new View.OnClickListener(){  
 @Override  
 public void onClick(View v){  
 int position = getLayoutPosition();  
 Context context = v.getContext();  
 Intent intent = new Intent(context, DeleteUpdateArtistActivity.class);  
 Artist artist = artistArrayList.get(position);  
 intent.putExtra("artist", artist);  
 context.startActivity(intent);  
 }  
 });  
 }  
 }  
}

This is the first adapter which holds many things that will be used in the following classes. It creates an artist view holder in which I call upon the activity\_listeview\_artist to set out how the data will look. Then I set the on click listener in the recycler view. I do this by firstly getting the position of which ever item in the recycler view the user has clicked on. Once all the data is collected I start a new intent which points at my DeleteUpdateArtistActivity.class.

## dbHelper.JAVA

package com.example.assignment1;  
import android.content.ContentValues;  
import android.content.Context;  
import android.database.Cursor;  
import android.database.sqlite.SQLiteDatabase;  
import android.database.sqlite.SQLiteOpenHelper;  
  
import java.util.ArrayList;  
  
  
public class dbHelper extends SQLiteOpenHelper{  
  
 public static String *DATABASE\_NAME* = "artist\_database.db";  
 private static final int *DATABASE\_VERSION* = 1;  
 private static final String *DATABASE\_TABLE\_ARTIST* = "artist";  
 private static final String *KEY\_ID* = "\_id";  
 private static final String *KEY\_ARTISTNAME\_COLUMN* = "ARTISTNAME\_COLUMN";  
 private static final String *KEY\_TRACKNAME\_COLUMN* = "TRACKNAME\_COLUMN";  
 private static final String *CREATE\_DATABASE\_TABLE\_ARTIST* = "CREATE TABLE IF NOT EXISTS "  
 + *DATABASE\_TABLE\_ARTIST* +"(" + *KEY\_ID* + " INTEGER PRIMARY KEY AUTOINCREMENT,"  
 + *KEY\_ARTISTNAME\_COLUMN* + " TEXT,"  
 + *KEY\_TRACKNAME\_COLUMN* + " TEXT );";  
  
  
 private static final String *DATABASE\_TABLE\_TRACK* = "track";  
 private static final String *KEY\_IDT* = "\_id";  
 private static final String *KEY\_TRACKNAME\_COLUMNT* = "TRACKNAMET\_COLUMN";  
 private static final String *KEY\_GENRE\_COLUMN* = "GENRE\_COLUMN";  
 private static final String *CREATE\_DATABASE\_TABLE\_TRACK* = "CREATE TABLE IF NOT EXISTS "  
 + *DATABASE\_TABLE\_TRACK* +"(" + *KEY\_ID* + " INTEGER PRIMARY KEY AUTOINCREMENT,"  
 + *KEY\_TRACKNAME\_COLUMNT* + " TEXT,"  
 + *KEY\_GENRE\_COLUMN* + " TEXT, \_fid TEXT, FOREIGN KEY (\"\_fid\") REFERENCES \"artist\"(\"\_id\") ON UPDATE CASCADE);";  
  
 public dbHelper(Context context){ super(context, *DATABASE\_NAME*, null, *DATABASE\_VERSION*);  
 }  
  
 @Override  
 public void onCreate(SQLiteDatabase db){  
 db.execSQL(*CREATE\_DATABASE\_TABLE\_ARTIST*);  
 db.execSQL(*CREATE\_DATABASE\_TABLE\_TRACK*);  
 }  
  
  
 @Override  
 public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion){  
 db.execSQL("DROP TABLE IF EXISTS '" + *DATABASE\_TABLE\_ARTIST* + "'");  
 db.execSQL("DROP TABLE IF EXISTS '" + *DATABASE\_TABLE\_TRACK* + "'");  
 onCreate(db);  
 }  
 public long addArtist(String artistName, String trackName){  
 SQLiteDatabase db = this.getWritableDatabase();  
 ContentValues newValues = new ContentValues();  
 newValues.put(*KEY\_ARTISTNAME\_COLUMN*, artistName);  
 newValues.put(*KEY\_TRACKNAME\_COLUMN*, trackName);  
 return db.insert(*DATABASE\_TABLE\_ARTIST*, null, newValues);  
 }  
  
  
 public void addTrack(String trackName, String genreName, int id){  
 SQLiteDatabase db = this.getWritableDatabase();  
 ContentValues newValues = new ContentValues();  
 newValues.put(*KEY\_GENRE\_COLUMN*, genreName);  
 newValues.put(*KEY\_TRACKNAME\_COLUMNT*, trackName);  
 newValues.put("\_fid", id);  
 db.insert(*DATABASE\_TABLE\_TRACK*, null, newValues);  
 }  
 public ArrayList<Artist>getAllArtists(){  
 SQLiteDatabase db = this.getReadableDatabase();  
 ArrayList<Artist>artistsModelArrayList = new ArrayList<Artist>();  
 String selectQuery = "SELECT \* FROM " + *DATABASE\_TABLE\_ARTIST*;  
 Cursor cursor = db.rawQuery(selectQuery, null);  
 while(cursor.moveToNext()){  
 Artist artist = new Artist();  
 artist.setId(cursor.getInt(cursor.getColumnIndex(*KEY\_ID*)));  
 artist.setArtistName((cursor.getString(cursor.getColumnIndex(*KEY\_ARTISTNAME\_COLUMN*))));  
 artist.setTrackName(cursor.getString(cursor.getColumnIndex(*KEY\_TRACKNAME\_COLUMN*)));  
 artistsModelArrayList.add(artist);  
 }  
 return artistsModelArrayList;  
 }  
 public ArrayList<Track>getAllTracks(int id) {  
 SQLiteDatabase db = this.getReadableDatabase();  
 ArrayList<Track> trackModelArrayList = new ArrayList<Track>();  
 String selectQuery = "SELECT \* FROM " + *DATABASE\_TABLE\_TRACK* + " WHERE \_fid=" + id + ";";  
 Cursor cursor = db.rawQuery(selectQuery, null);  
  
 if (cursor != null && cursor.moveToFirst()) {  
 do {  
 Track track = new Track();  
 track.setId(cursor.getInt(cursor.getColumnIndex(*KEY\_IDT*)));  
 track.setGenreName((cursor.getString(cursor.getColumnIndex(*KEY\_GENRE\_COLUMN*))));  
 track.setTrackName(cursor.getString(cursor.getColumnIndex(*KEY\_TRACKNAME\_COLUMNT*)));  
 trackModelArrayList.add(track);  
 } while (cursor.moveToNext());  
 }  
  
 return trackModelArrayList;  
 }  
 public void updateArtists(int id, String artistName, String trackName){  
 SQLiteDatabase db = this.getWritableDatabase();  
 ContentValues updatedValues = new ContentValues();  
 updatedValues.put(*KEY\_ARTISTNAME\_COLUMN*, artistName);  
 updatedValues.put(*KEY\_TRACKNAME\_COLUMN*, trackName);  
 String where = *KEY\_ID* + "= ?";  
 String whereArgs[] = new String[]{String.*valueOf*(id)};  
 db.update(*DATABASE\_TABLE\_ARTIST*, updatedValues, where, whereArgs);  
 }  
 public void updateTracks(int id, String genreName, String trackName)  
 {  
 SQLiteDatabase db = this.getWritableDatabase();  
 ContentValues updatedValues = new ContentValues();  
 updatedValues.put(*KEY\_TRACKNAME\_COLUMNT*, trackName);  
 updatedValues.put(*KEY\_GENRE\_COLUMN*, genreName);  
 String where = *KEY\_ID* + "= ?";  
 String whereArgs[] = new String[]{String.*valueOf*(id)};  
 db.update(*DATABASE\_TABLE\_TRACK*, updatedValues, where, whereArgs);  
 }  
 public void deleteArtist(int id){  
 SQLiteDatabase db = this.getWritableDatabase();  
 db.delete(*DATABASE\_TABLE\_ARTIST*, *KEY\_ID*+ " =?", new String[]{String.*valueOf*(id)});  
 }  
 public void deleteTracks(int id){  
 SQLiteDatabase db = this.getWritableDatabase();  
 db.delete(*DATABASE\_TABLE\_TRACK*, *KEY\_ID*+ " =?", new String[]{String.*valueOf*(id)});  
 }  
}

This is my database helper which is control of making most of the changes used in the rest of the programme. Firstly it has the task of creating all the tables and fields that will be used. After it hold many functions like deleting from both tables and updating items in both tables. It also holds the code for which allows me to get all the items in the tables to be stored in my recycler views for the user to see.

## DeleteUpdateArtistActivity.JAVA

package com.example.assignment1;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.EditText;  
import android.widget.Toast;  
import androidx.annotation.Nullable;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.recyclerview.widget.LinearLayoutManager;  
import androidx.recyclerview.widget.RecyclerView;  
  
import java.util.ArrayList;  
import java.util.LinkedList;  
  
public class DeleteUpdateArtistActivity extends AppCompatActivity {  
 private Artist artist;  
 private RecyclerView mRecyclerView;  
 private ArrayList<Track> trackArrayList = new ArrayList<>();  
 private TrackAdapter trackAdapter;  
 private EditText etArtistName, etTrackName;  
 private dbHelper databaseHelper;  
  
 @Override  
 protected void onCreate(@Nullable Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_delete\_update\_artist*);  
  
 Intent intent = getIntent();  
 artist = (Artist) intent.getSerializableExtra("artist");  
  
 databaseHelper = new dbHelper(this);  
 ArrayList<Track> tracks = databaseHelper.getAllTracks(artist.getId());  
 trackArrayList.addAll(tracks);  
 trackAdapter = new TrackAdapter(this, trackArrayList);  
 mRecyclerView = findViewById(R.id.*trackRecycler*);  
 mRecyclerView.setAdapter(trackAdapter);  
 mRecyclerView.setLayoutManager(new LinearLayoutManager(this));  
  
 etArtistName = (EditText) findViewById(R.id.*etArtistName*);  
 etTrackName = (EditText) findViewById(R.id.*etTrackName*);  
 etArtistName.setText(artist.getArtistname());  
 etTrackName.setText(artist.getTrackName());  
 }  
 public void btnUpdateOnClick(View view){  
 databaseHelper.updateArtists(artist.getId(),etArtistName.getText().toString(),etTrackName.getText().toString());  
 Toast.*makeText*(DeleteUpdateArtistActivity.this, "Updated Successfully", Toast.*LENGTH\_SHORT*).show();  
 Intent intent = new Intent(DeleteUpdateArtistActivity.this, MainActivity.class);  
 intent.addFlags(Intent.*FLAG\_ACTIVITY\_CLEAR\_TASK* | Intent.*FLAG\_ACTIVITY\_NEW\_TASK*);  
 startActivity(intent);  
 }  
 public void btnDeleteOnClick(View view){  
 databaseHelper.deleteArtist(artist.getId());  
 Toast.*makeText*(DeleteUpdateArtistActivity.this , "Deleted Successfully", Toast.*LENGTH\_SHORT*).show();  
 Intent intent = new Intent(DeleteUpdateArtistActivity.this, MainActivity.class);  
 intent.addFlags(Intent.*FLAG\_ACTIVITY\_CLEAR\_TASK* | Intent.*FLAG\_ACTIVITY\_NEW\_TASK*);  
 startActivity(intent);  
 }  
  
}

In this class I collect all the data from the tables and set the data into text views that are appropriate for the user to have an easy time updating them. I call the update and delete function from the database helper.

## DeleteUpdateTrackActivity.JAVA

package com.example.assignment1;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.EditText;  
import android.widget.Toast;  
  
import androidx.annotation.Nullable;  
import androidx.appcompat.app.AppCompatActivity;  
  
  
public class DeleteUpdateTrackActivity extends AppCompatActivity {  
  
 private Track track;  
 private EditText etTrackName, etGenreName;  
 private dbHelper databaseHelper;  
 @Override  
 protected void onCreate(@Nullable Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_delete\_update\_track*);  
  
 Intent intent = getIntent();  
 track = (Track) intent.getSerializableExtra("track");  
 databaseHelper = new dbHelper(this);  
 etTrackName = (EditText) findViewById(R.id.*etTrackName*);  
 etGenreName = (EditText) findViewById(R.id.*etGenreName*);  
 etGenreName.setText(track.getGenreName());  
 etTrackName.setText(track.getTrackName());  
 }  
 public void btnUpdateOnClickt(View view){  
  
 databaseHelper.updateTracks(track.getId(), etGenreName.getText().toString(),etTrackName.getText().toString());  
 Toast.*makeText*(DeleteUpdateTrackActivity.this, "Updated Successfully", Toast.*LENGTH\_SHORT*).show();  
 Intent intent = new Intent(DeleteUpdateTrackActivity.this, MainActivity.class);  
 intent.addFlags(Intent.*FLAG\_ACTIVITY\_CLEAR\_TASK* | Intent.*FLAG\_ACTIVITY\_NEW\_TASK*);  
 startActivity(intent);  
 }  
 public void btnDeleteOnClickt(View view){  
  
 databaseHelper.deleteTracks(track.getId());  
 Toast.*makeText*(DeleteUpdateTrackActivity.this , "Deleted Successfully", Toast.*LENGTH\_SHORT*).show();  
 Intent intent = new Intent(DeleteUpdateTrackActivity.this, MainActivity.class);  
 intent.addFlags(Intent.*FLAG\_ACTIVITY\_CLEAR\_TASK* | Intent.*FLAG\_ACTIVITY\_NEW\_TASK*);  
 startActivity(intent);  
 }  
}

In this class I collect all the data from the tables and set the data into text views that are appropriate for the user to have an easy time updating them. I call the update and delete function from the database helper.

## MainActivity.JAVA

package com.example.assignment1;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.Toast;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.recyclerview.widget.LinearLayoutManager;  
import androidx.recyclerview.widget.RecyclerView;  
  
import java.util.ArrayList;  
  
public class MainActivity extends AppCompatActivity {  
 private Button btnSave;  
 private Button btnDisplayAll;  
 private EditText etArtistName;  
 private EditText etTrackName;  
 private EditText etGenreName;  
 private dbHelper databaseHelper;  
 private RecyclerView mRecyclerView;  
 private ArrayList<Artist> artistModelArrayList;  
 private ArtistAdapter artistsCustomAdapter;  
  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState){  
  
 super.onCreate(savedInstanceState);  
  
 setContentView(R.layout.*activity\_main*);  
  
 databaseHelper = new dbHelper(this);  
 mRecyclerView = findViewById(R.id.*artistRecycler*);  
  
 artistModelArrayList = databaseHelper.getAllArtists();  
 artistsCustomAdapter = new ArtistAdapter(this, artistModelArrayList);  
 mRecyclerView.setAdapter(artistsCustomAdapter);  
 mRecyclerView.setLayoutManager(new LinearLayoutManager(this));  
  
 }  
  
  
 public void btnNewArtistTrack(View view) {  
 Intent intent = new Intent(this, NewArtistTrack.class);  
 startActivity(intent);  
 }  
  
  
}

In the main activity there isn’t really much going on apart from setting the adapter and recyclervew and also an intent to point towards the class of adding a new artist/track.

## NewArtistTrack.JAVA

package com.example.assignment1;  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.Toast;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.recyclerview.widget.RecyclerView;  
  
import java.util.ArrayList;  
  
public class NewArtistTrack extends AppCompatActivity {  
 private Button btnSave;  
 private Button btnDisplayAll;  
 private dbHelper databaseHelper;  
 private RecyclerView mRecyclerView;  
 private ArrayList<Artist> artistModelArrayList;  
 private ArtistAdapter artistsCustomAdapter;  
 private EditText etArtistName;  
 private EditText etTrackName;  
 private EditText etGenreName;  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_add\_artist\_track*);  
  
 databaseHelper = new dbHelper(this);  
  
 etArtistName = (EditText) findViewById(R.id.*etArtistName*);  
 etTrackName = (EditText) findViewById(R.id.*etTrackName*);  
 etGenreName = (EditText) findViewById(R.id.*etGenreName*);  
 }  
 public void btnSaveOnClick(View view){  
 long id = databaseHelper.addArtist(etArtistName.getText().toString(), etTrackName.getText().toString());  
  
 databaseHelper.addTrack(etTrackName.getText().toString(), etGenreName.getText().toString(), Math.*toIntExact*(id));  
 etTrackName.setText("");  
 etArtistName.setText("");  
 etGenreName.setText("");  
 Toast.*makeText*(NewArtistTrack.this, "Stored Successfully!", Toast.*LENGTH\_SHORT*).show();  
 }  
 public void btnDisplayAllOnClick(View view){  
 Intent intent = new Intent(this, MainActivity.class);  
 startActivity(intent);  
 }  
  
}

In this class I am reading the data the user has inputted into text views and if they click on save I am storing all this new information into the tables.

## Track.JAVA

package com.example.assignment1;  
import java.io.Serializable;  
public class Track implements Serializable{  
 private String trackName;  
 private String genreName;  
 private int id;  
 public int getId(){  
 return id;  
 }  
  
 public void setId(int id){  
 this.id = id;  
 }  
  
 public String getGenreName(){  
 return genreName;  
 }  
  
 public void setGenreName(String genreName){  
 this.genreName = genreName;  
 }  
  
 public String getTrackName(){  
 return trackName;  
 }  
  
 public void setTrackName (String trackName){  
 this.trackName = trackName;  
 }  
  
 public String toString(){  
 String s = "Track: " + trackName + "\n";  
 s += "Genre: " + genreName + "\n";  
 return s;  
 }  
  
  
}

This is creating the database for the track which holds the track name and genre.

## TrackAdapter.JAVA

package com.example.assignment1;  
import android.content.Context;  
import android.content.Intent;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.TextView;  
import androidx.annotation.NonNull;  
import androidx.recyclerview.widget.RecyclerView;  
import java.util.ArrayList;  
  
public class TrackAdapter extends RecyclerView.Adapter<TrackAdapter.TrackViewHolder> {  
 private LayoutInflater mInflater;  
 private ArrayList<Track> trackArrayList;  
 public TrackAdapter(Context context, ArrayList<Track> trackList){  
 mInflater = LayoutInflater.*from*(context);  
 this.trackArrayList = trackList;  
 }  
  
 @NonNull  
 @Override  
 public TrackAdapter.TrackViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {  
 View view = mInflater.inflate(R.layout.*activity\_listview\_track*, parent, false);  
 return new TrackViewHolder(view, this);  
 }  
  
 @Override  
 public void onBindViewHolder(@NonNull TrackAdapter.TrackViewHolder holder, int position) {  
 Track mCurrent = trackArrayList.get(position);  
 holder.trackItemView.setText(mCurrent.getTrackName());  
 holder.genreItemView.setText(mCurrent.getGenreName());  
 }  
 public class TrackViewHolder extends RecyclerView.ViewHolder{  
 public final TextView trackItemView;  
 public final TextView genreItemView;  
 final TrackAdapter mAdapter;  
 public TrackViewHolder(@NonNull View itemView, TrackAdapter adapter) {  
 super(itemView);  
 trackItemView = itemView.findViewById(R.id.*trackName*);  
 genreItemView = itemView.findViewById(R.id.*genreName*);  
 this.mAdapter = adapter;  
 trackItemView.setOnClickListener(new View.OnClickListener(){  
 @Override  
 public void onClick(View v) {  
 int position = getLayoutPosition();  
 Context context = v.getContext();  
 Intent intent = new Intent(context, DeleteUpdateTrackActivity.class);  
 Track track = trackArrayList.get(position);  
 intent.putExtra("track", track);  
 context.startActivity(intent);  
 }  
 });  
 }  
 }  
 @Override  
 public int getItemCount() {  
 return trackArrayList.size();  
 }  
}

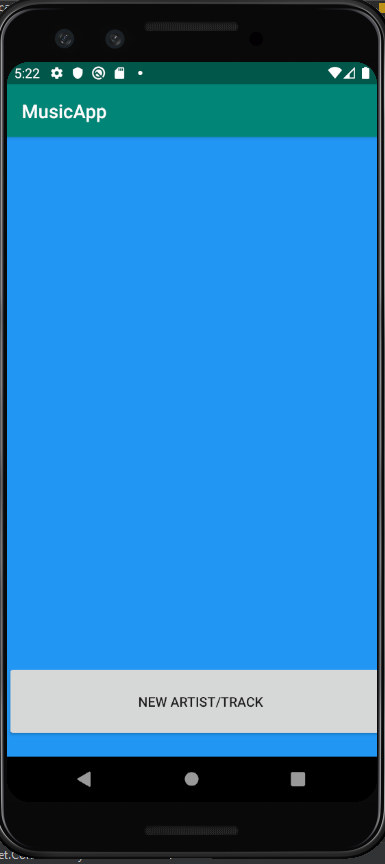
The track adapter is basically the same as the artist adapter. It is where I set the on click listener for the recycler view in activity\_delete\_update\_artist.xml.

# Testing

## Test 1

Starting the application.

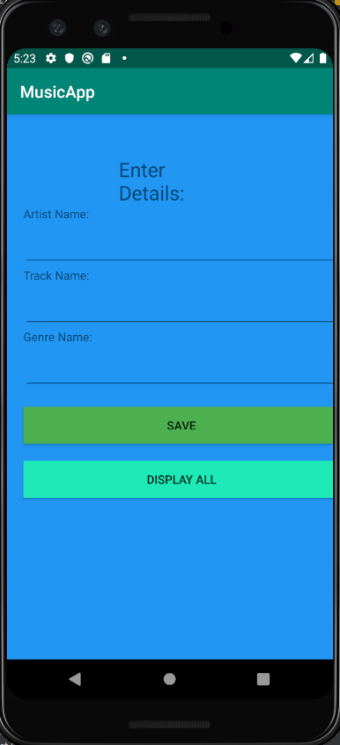
It should start loading in the main activity layout which contains an empty recycler view and a button to add in data.

 This is indeed what happens.

## Test 2

Pressing the NEW ARTIST/TRACK button

It should launch the activity\_add\_artist\_track activity.

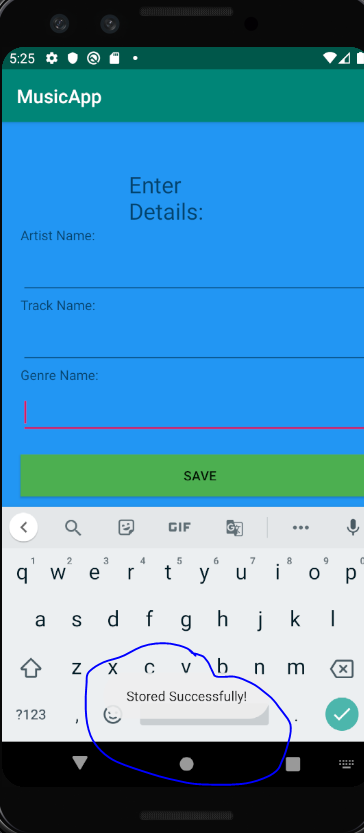


This is indeed what happens

## Test 3

Inputting data and pressing save

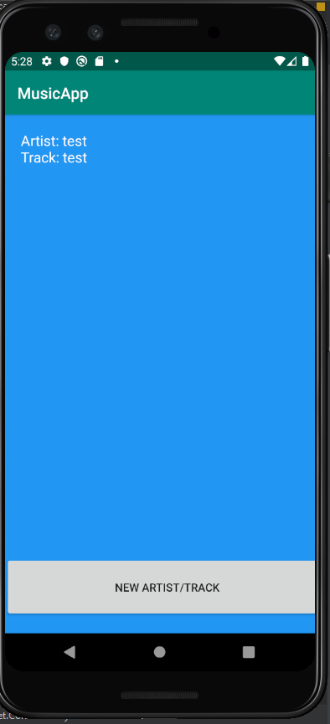
It should pop up with a small toast message informing the user that it was saved.



This is what happened the data inputted into the text views were removed so if the user would like to input more tracks and artists they can.  
Test 4

Pressing display all.

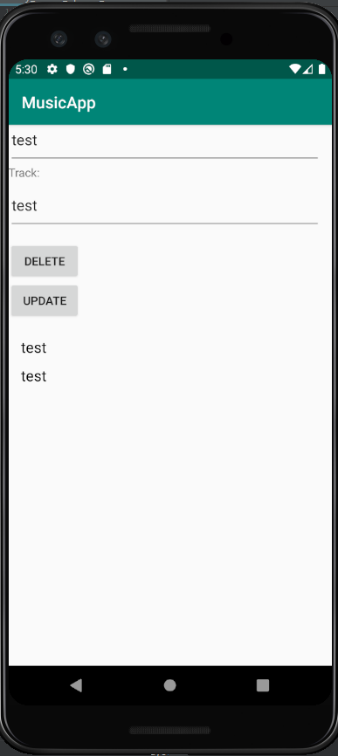
Pressing display all to go back to main activity to see all the saved data



This is what happened clearly shows the item that was added into the interface.  
Test 5

Clicking on the item in the recycler view.

This should take us to activity\_delete\_update\_artist.xml with the data already in the fields.

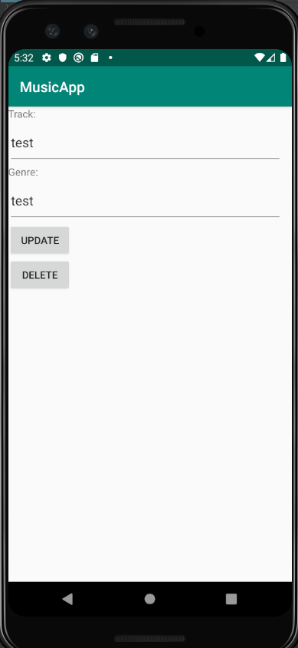


This is what has happened and has the genre and track name in the Recycler View at the bottom and also has the fields already filled in.

Test 6

Clicking on the item in the Recycler View

It should take us to activity\_delete\_update\_track.xml which will have options on deleting and updating the genre and track name.

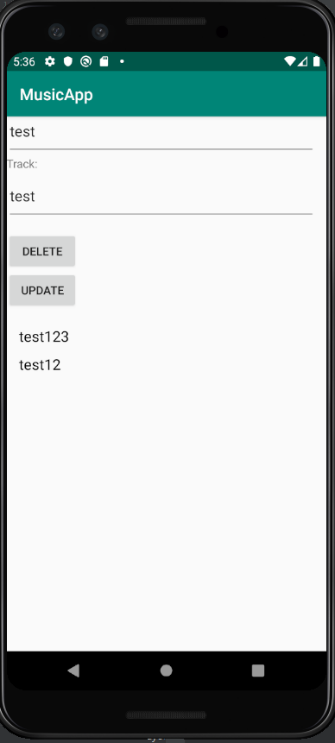


It has worked.

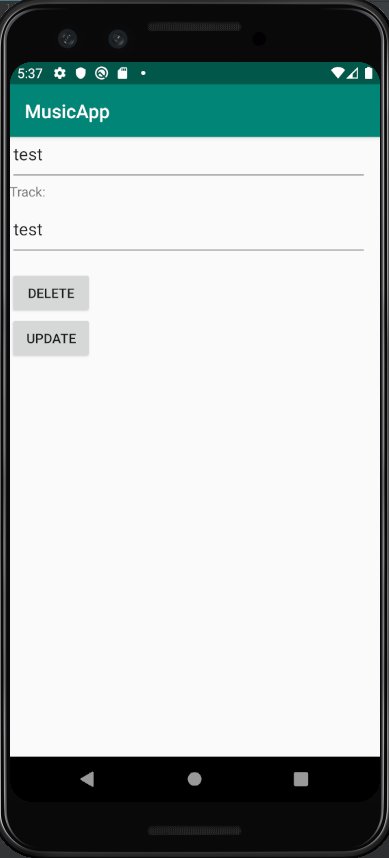
## Test 7

Updating the genre and track name then deleting them.

It should update the details first then once I press delete should delete.



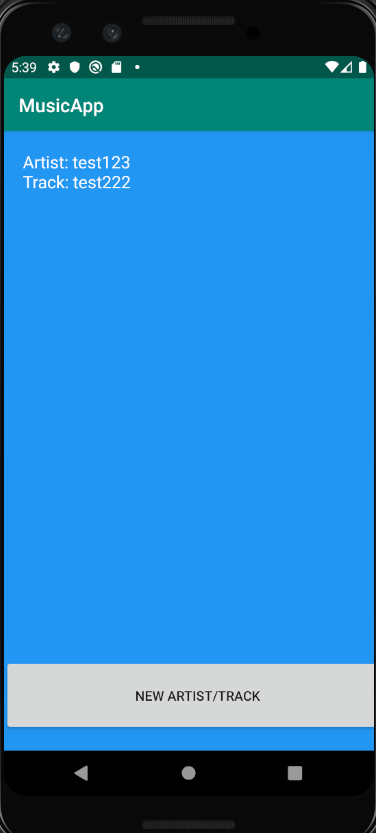
The data in the recycler view has updated

 And the items have deleted.

## Test 8

Updating and deleting the track name and artist name.

Should update the information held in the recycler view in main activity.

Updates.

Deleted.