Screenshots & Descriptions

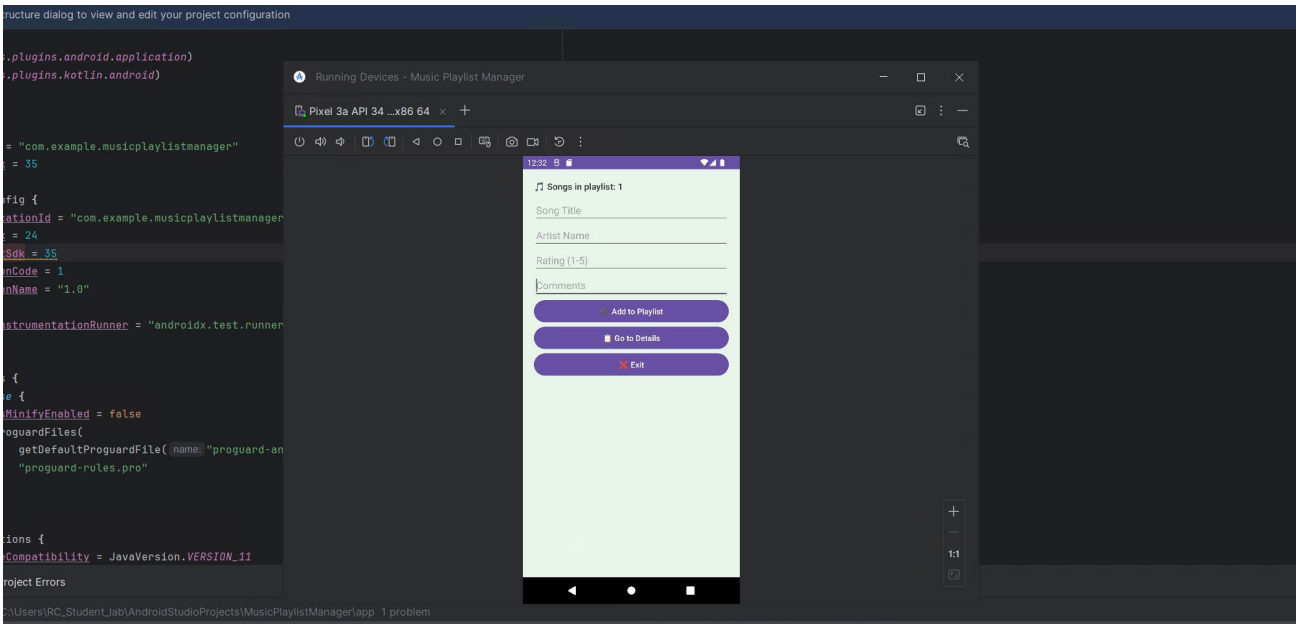
1. Source Code (MainActivity.kt)

```
package com.example.playlistapp

import android.content.DialogInterface

import android.content.Intent

import android.os.Bundle

import android.widget.*

import androidx.appcompat.app.AlertDialog

import androidx.appcompat.app.AppCompatActivity

class MainActivity : AppCompatActivity() {

    // Declare parallel arrays to hold song data

    private val songs = ArrayList<String>()

    private val artists = ArrayList<String>()

    private val ratings = ArrayList<Int>()

    private val comments = ArrayList<String>()

    override fun onCreate(savedInstanceState: Bundle?) {

        super.onCreate(savedInstanceState)

        setContentView(R.layout.activity_main)

        // Connect UI elements with their views

        val txtSong = findViewById<EditText>(R.id.txtSong)

        val txtArtist = findViewById<EditText>(R.id.txtArtist)

        val txtRating = findViewById<EditText>(R.id.txtRating)

        val txtComment = findViewById<EditText>(R.id.txtComment)
```

```
        val txtCount = findViewById<TextView>(R.id.txtCount)
        val btnAdd = findViewById<Button>(R.id.btnAdd)
        val btnDetails = findViewById<Button>(R.id.btnDetails)
        val btnExit = findViewById<Button>(R.id.btnExit)
        // Handle Add Button click
        btnAdd.setOnClickListener {
            try {
                val song = txtSong.text.toString().trim()
                val artist = txtArtist.text.toString().trim()
                val ratingText = txtRating.text.toString().trim()
                val comment = txtComment.text.toString().trim()
                // Validate input
                if (song.isEmpty() || artist.isEmpty() || ratingText.isEmpty() ||
comment.isEmpty()) {
                    Toast.makeText(this, "Please fill in all fields",
Toast.LENGTH_SHORT).show()
                    return@setOnClickListener
                }
                val rating = ratingText.toInt()
                if (rating !in 1..5) {
                    Toast.makeText(this, "Rating must be between 1 and 5",
Toast.LENGTH_SHORT).show()
                    return@setOnClickListener
```

```
            }
            // Store the details in parallel arrays
            songs.add(song)
            artists.add(artist)
            ratings.add(rating)
            comments.add(comment)
            Toast.makeText(this, "✅ Song added successfully!",
Toast.LENGTH_SHORT).show()
            // Clear input fields after adding
            txtSong.text.clear()
            txtArtist.text.clear()
            txtRating.text.clear()
            txtComment.text.clear()

            // Update count display
            txtCount.text = "🎵 Songs in playlist: ${songs.size}"

        } catch (e: Exception) {
            Toast.makeText(this, "Error: ${e.message}", Toast.LENGTH_LONG).show()
        }
    }
    // Navigate to Detailed View
    btnDetails.setOnClickListener {
```

```
            val intent = Intent(this, DetailedViewActivity::class.java)
            intent.putStringArrayListExtra("songs", songs)
            intent.putStringArrayListExtra("artists", artists)
            intent.putIntegerArrayListExtra("ratings", ratings)
            intent.putStringArrayListExtra("comments", comments)
            startActivity(intent)
        }
        // Confirm exit with dialog
        btnExit.setOnClickListener {
            AlertDialog.Builder(this)
                .setTitle("Exit App")
                .setMessage("Are you sure you want to exit the app?")
                .setPositiveButton("Yes") { _: DialogInterface, _: Int -> finishAffinity() }
                .setNegativeButton("No", null)
                .show()
        }
    }
}
```



2. 🧠 Source Code (DetailedViewActivity.kt)

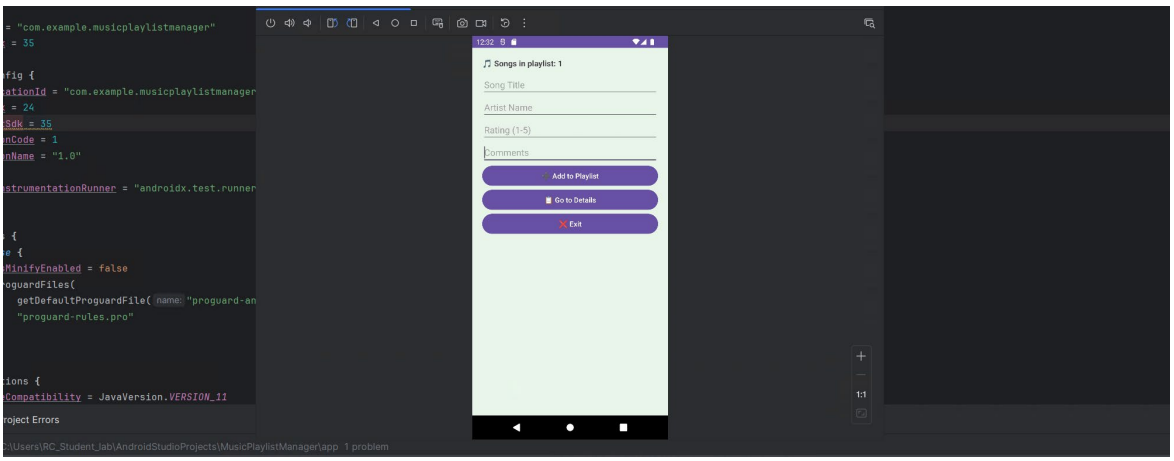
```
package com.example.musicplaylistmanager


import android.os.Bundle

import android.widget.*

import androidx.appcompat.app.AppCompatActivity


class DetailedViewActivity : AppCompatActivity() {


    private lateinit var songs: ArrayList<String>

    private lateinit var artists: ArrayList<String>

    private lateinit var ratings: ArrayList<Int>

    private lateinit var comments: ArrayList<String>
```

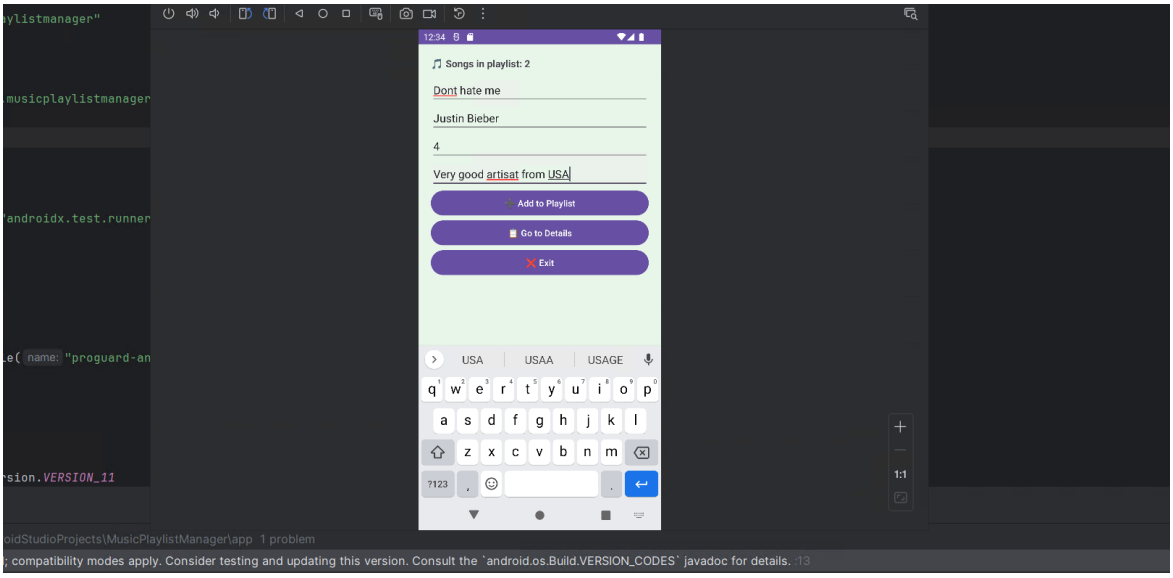
```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_detailed_view)

    val txtOutput = findViewById<TextView>(R.id.txtOutput)
    val btnDisplay = findViewById<Button>(R.id.btnDisplay)
    val btnAverage = findViewById<Button>(R.id.btnAverage)
    val btnBack = findViewById<Button>(R.id.btnBack)

    // Receive arrays from MainActivity
    songs = intent.getStringArrayListExtra("songs") ?: arrayListOf()
    artists = intent.getStringArrayListExtra("artists") ?: arrayListOf()
    ratings = intent.getIntegerArrayListExtra("ratings") ?: arrayListOf()
    comments = intent.getStringArrayListExtra("comments") ?: arrayListOf()

    // Display all song entries
    btnDisplay.setOnClickListener {
        val builder = StringBuilder()
        if (songs.isEmpty()) {
            builder.append("No songs available.\n")
        } else {
            for (i in songs.indices) {
```

```
            builder.append("🎵 Song: ${songs[i]}\n")
            builder.append("🎤 Artist: ${artists[i]}\n")
            builder.append("⭐ Rating: ${ratings[i]}\n")
            builder.append("💬 Comment: ${comments[i]}\n\n")
        }
    }
    txtOutput.text = builder.toString()
}

// Calculate average rating using loop
btnAverage.setOnClickListener {
    if (ratings.isEmpty()) {
        txtOutput.text = "No ratings to calculate average."
    } else {
        var total = 0
        for (rating in ratings) {
            total += rating
        }
        val average = total.toDouble() / ratings.size
        txtOutput.text = "⭐ Average Rating: %.2f".format(average)
    }
}
```

```
        // Go back to Main screen
        btnBack.setOnClickListener {
            finish()
        }
    }
}
```

3. 🏠 Main Screen – App UI



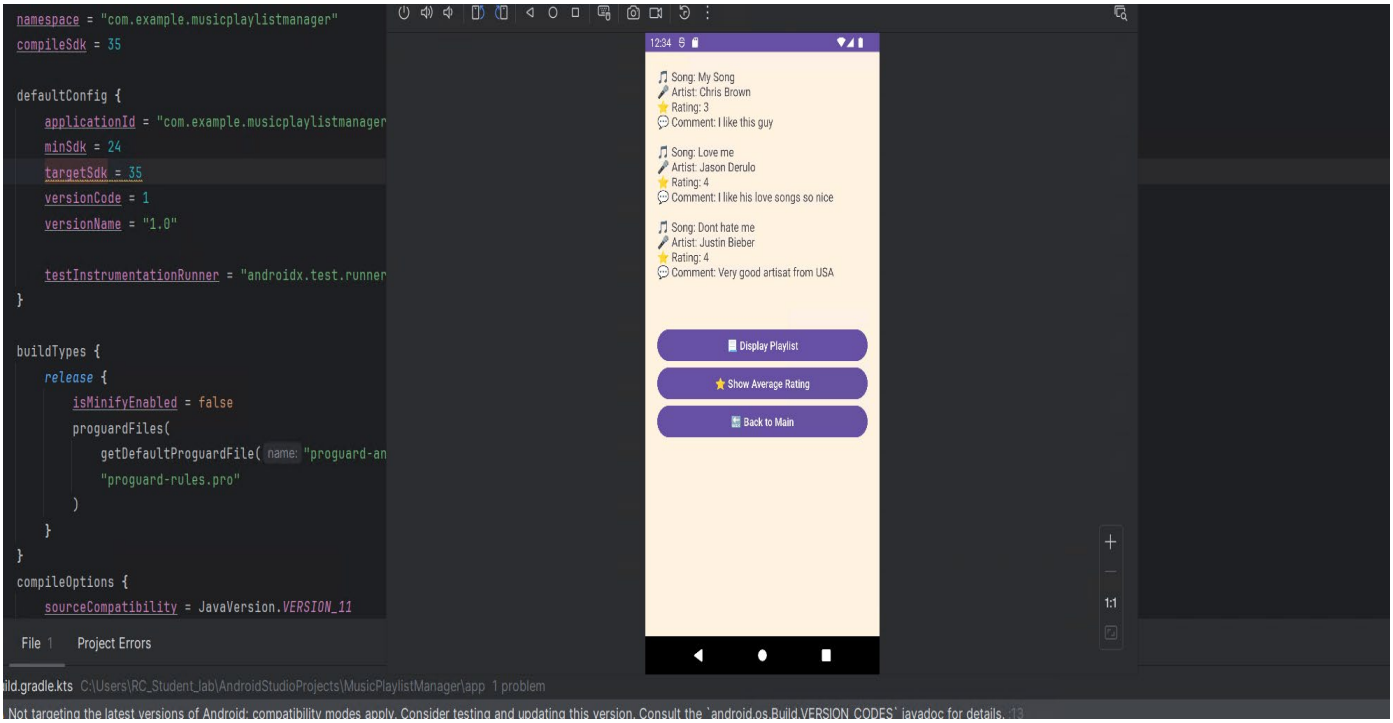


4. 📋 Detailed View Screen

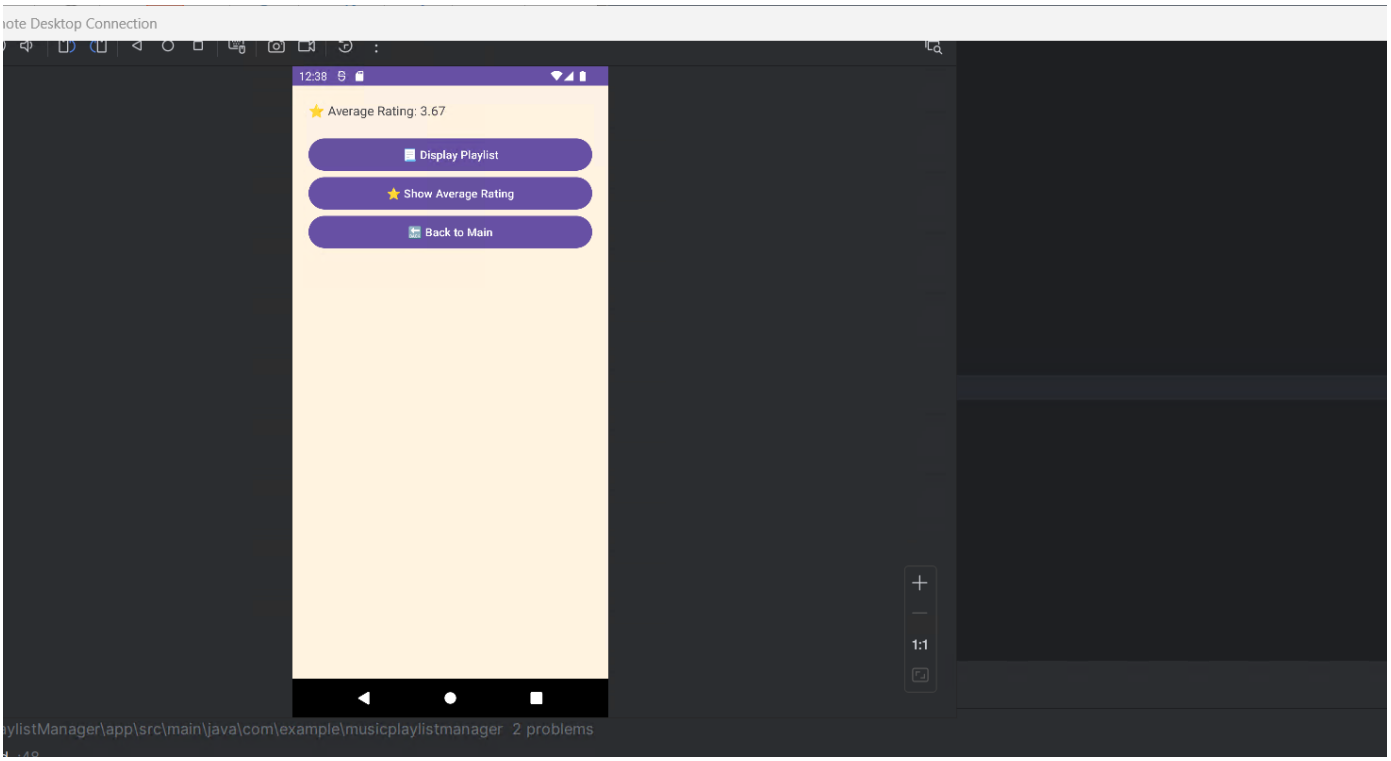
```
package com.example.musicplaylistmanager
import android.os.Bundle
import android.widget.*
import androidx.appcompat.app.AppCompatActivity
class DetailedViewActivity : AppCompatActivity() {
    private lateinit var songs: ArrayList<String>
    private lateinit var artists: ArrayList<String>
    private lateinit var ratings: ArrayList<Int>
    private lateinit var comments: ArrayList<String>
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_detailed_view)
        val txtOutput = findViewById<TextView>(R.id.txtOutput)
        val btnDisplay = findViewById<Button>(R.id.btnDisplay)
        val btnAverage = findViewById<Button>(R.id.btnAverage)
        val btnBack = findViewById<Button>(R.id.btnBack)
        // Receive arrays from MainActivity
        songs = intent.getStringArrayListExtra("songs") ?: arrayListOf()
        artists = intent.getStringArrayListExtra("artists") ?: arrayListOf()
```

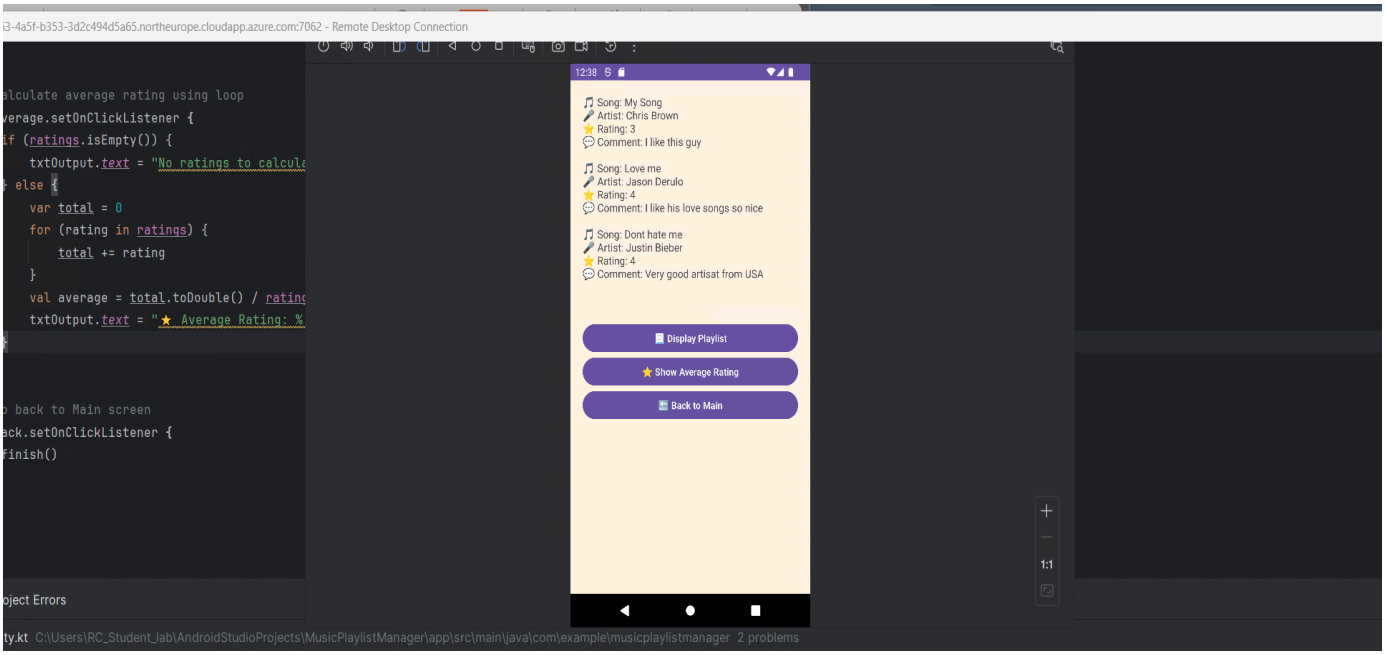
```
ratings = intent.getIntegerArrayListExtra("ratings") ?: arrayListOf()
comments = intent.getStringArrayListExtra("comments") ?: arrayListOf()

// Display all song entries
btnDisplay.setOnClickListener {
    val builder = StringBuilder()
    if (songs.isEmpty()) {
        builder.append("No songs available.\n")
    } else {
        for (i in songs.indices) {
            builder.append("🎵 Song: ${songs[i]}\n")
            builder.append("🎤 Artist: ${artists[i]}\n")
            builder.append("⭐ Rating: ${ratings[i]}\n")
            builder.append("💬 Comment: ${comments[i]}\n\n")
        }
    }
    txtOutput.text = builder.toString()
}

// Calculate average rating using loop
btnAverage.setOnClickListener {
    if (ratings.isEmpty()) {
        txtOutput.text = "No ratings to calculate average."
    } else {
        var total = 0
        for (rating in ratings) {
            total += rating
        }
        val average = total.toDouble() / ratings.size
        txtOutput.text = "⭐ Average Rating: %.2f".format(average)
```

```
        }
    }
    // Go back to Main screen
    btnBack.setOnClickListener {
        finish()
    }
  }
}
```



5. Emulator Running App


ote Desktop Connection
12:38
Average Rating: 3.67
Display Playlist
Show Average Rating
Back to Main
1:1
ylistManager\app\src\main\java\com\example\musicplaylistmanager 2 problems


3-4a5f-b353-3d2c494d5a65.northeurope.cloudapp.azure.com:7062 - Remote Desktop Connection
lculate average rating using loop
verage.setOnClickListener {
if (ratings.isEmpty()) {
txtOutput.text = "No ratings to calcula
} else {
var total = 0
for (rating in ratings) {
total += rating
}
val average = total.toDouble() / rating
txtOutput.text = "★ Average Rating: %
back to Main screen
ack.setOnClickListener {
finish()
oject Errors
ty.kt C:\Users\RC_Student_lab\AndroidStudioProjects\MusicPlaylistManager\app\src\main\java\com\example\musicplaylistmanager 2 problems
12:38
Song: My Song
Artist: Chris Brown
Rating: 3
Comment: I like this guy
Song: Love me
Artist: Jason Derulo
Rating: 4
Comment: I like his love songs so nice
Song: Dont hate me
Artist: Justin Bieber
Rating: 4
Comment: Very good artisat from USA
Display Playlist
Show Average Rating
Back to Main
1:1

Summary

This Kotlin-based app uses:

- Arrays and loops for data handling
- Screen navigation between two activities
- Error handling and user feedback
- Clean, readable code with comments
- A user-friendly interface with color, icons, and responsiveness