Java Full Stack

Module 1 Assessment
[Core Java, SQL, DB Concepts & JDBC]

JSpiders Unit of

TESTYANTRA Software Solutions

Table of Contents

App Development - Assessment Overview

As part of this assessment you need to develop a console based application i.e. application which will not have any GUI but has a main method which takes input from console, which simulates "Music Player Application". As part of this assessment you need to use

- MySQL DB to store the Data
- Core Java to develop Console Based application
- JDBC & SQL to perform CURD operation on Data

Please Note: You need not to play actual music file as part of this assessment & you just need to print message like "Playing the Song - SONG TITLE" with the sleep for 20 seconds is enough

App Development - Assessment Description

1. Create a Database by name "MusicPlayer"

2. Create a Table by name "MusicFiles" with the following structure

Song_ID	int(10) AutoIncrement PrimaryKey
Song_Title	varchar(50)
Artist_Name	varchar(50)
Album_Name	varchar(50)
Song_Location	varchar(50)
Description	varchar(250)

3. Create a Java Program (let's call this as an application henceforth) which has main method which prints the following options
   i. Press 1 to "Play a Song"
   ii. Press 2 to "Search a Song"
   iii. Press 3 to "Show all Songs"
   iv. Press 4 to "Operate on Songs Database"

If user press 1, then application should print following options

- Press A to "Play all songs"
- Press B to "Play Songs Randomly"
- Press C to "Play a Particular Song"

If user choses
Option A, then Application should play all the songs sequentially order by song title
Option B, then Application should play all the songs randomly
Option C, then User has to provide "song title" & Application should play that particular song

If user press 2, then application should collect the song name from console, search the DB for that song name (search on song title) & print all the songs which matches to the name. The display should be in Table View. The table should display a column called "File Name" followed by column called Attributes & Play.

- Under Attributes, the application should display the Title, Artist and Album associated with the song file
- Under Play, the application should print unique number to play that particular song. When user input that number, then that particular song should be played

If user press 3, then application should display all of the songs (in Table View, as explained above) order by song title

If user press 4, then application should print following options

- Press A to "Add Song" to Songs Repository
- Press B to "Edit an existing Song" info
- Press C to "Delete an existing Song" info

If user choses
Option A, then Application should collect all the info & insert that data into a song to repository
Option B, then Application should collect all the info & edit a particular song
Option B, then Application should collect "song info" & delete that song from DB

Grading

Complete development of Music Player will fetch you **100 points**. For this assignment, you need to earn at least 50 points in order to get any points at all on your course grade from this assignment. **You will be responsible for running PMD on ALL of your code.**

- Ability to understand the main requirements along with hidden requirements (ex, what if user enter the song title to play which doesn't exist in DB) and implementing them along with unit test cases (not JUnit) [20 Points]
- Code structuring, following OOPS principles (ex, code reuse) [25 Points]
- Displaying the songs in Table View [10 Points]
- Playing song randomly [10 Points]
- Playing song with a delay [5 Points]
- Creating a DB and Table along with some sample data [5 Points]
- Inserting a Song into DB [5 Points]
- Editing a Song in a DB [5 Points]
- Deleting Song from DB [5 Points]
- Displaying a Songs order by song name [5 Points]
- Searching a Song based on song name [5 Points]

Tips and Considerations

- ✓ **Submit code that compiles!!!** Non-compiling code will receive a 0 points
- ✓ Prioritize your work to get as many points as you can. Even if you cannot display songs in table view, there would still be 90 potential points you could get.
- ✓ For each of your assignment we will run PMD and deduct one point for every PMD error. For this assignment the **PMD cap is 100**, meaning you can lose up to 100 points on this assignment due to PMD errors.
- ✓ Do not import full package (ex, import java.util.*) & import only specific class

Submission

- Your assignment should be submitted to QSpiders QuickView Application using anyone of the following approach
  - ✓ Either as an attachment (assignment programs along with other files such as information or SQL Query files)
    OR
  - ✓ Keep these assignment codes in GitHub & provide the URL in QuickView Application

- Verify the Success of Your Submission to QSpiders QuickView Application

- Practice safe submission! Verify that your assignment files are truly submitted correctly, the upload was successful, and that your program runs with no syntax or runtime errors. It is solely your responsibility to turn in your assignment and practice this safe submission safeguard.

- After submitting the files to QuickView, return to the Assignment option and it should show the submitted files.

- Download copies of your submitted files from the QuickView Assignment page placing them in a new folder. Re-run and test the files you downloaded from QuickView to make sure it's what you expect. This procedure helps guard against a few things.
  - ✓ It helps ensure that you turn in the correct files
  - ✓ It helps you realize if you omit a file or files

- **NOTE: -**
  - ✓ Missing files will not be given any credit. If you do discover that you omitted a file, submit all of your files again, not just the missing one
  - ✓ Non-compiling/non-running assessment programs will receive zero points
  - ✓ Late assignment will not be accepted regardless of excuse.
  - ✓ Treat the due date with respect. Do not wait until the last minute!