COMP9120 Database Management Systems

Assignment 2: Database Application Development

Group assignment (12%)

Introduction

The objectives of this assignment are to gain practical experience in interacting with a relational database management system using an Application Programming Interface (API) (JDBC). This assignment additionally provides an opportunity to use more advanced features of a database such as functions.

This is a group assignment for teams of about 3 members, and it is assumed that you will continue in your Assignment 1 group. You should inform the unit coordinator as soon as possible if you wish to change groups.

Please also keep an eye on your email and any announcements that may be made on Ed.

Submission Details

The final submission of your database application is due at 11:59pm on Friday 28th May. You should submit the *items for submission* (detailed below) via Canvas.

Items for submission

Please submit your solution to Assignment 2 in the 'Assignment' section of the unit's Canvas site by the deadline, including EXACTLY TWO files:

- A SQL file (`SETSchema.sql`) containing the SQL statements necessary to generate the database schema and sample data. This should contain the original schema and insert SQL statements, and any changes or additions you may have made.
- A Java file (`PostgresRepositoryProvider.java`) containing the Java code you have written to access the database.

Section 1: Introducing the Sydney Entertainment & Theatres (SET) System

In this assignment, you will be working with the SET System which is currently under development. The system still requires work in numerous areas, including the interaction with the database. Your main task in this assignment is to handle requests for reads and writes to the database coming from the user interface (UI). We first describe the main features that the SET System should include from a UI perspective, and then discuss where the majority of your database code needs to be implemented.

Main Features

Logging In

The first form a user is presented with when starting the SET System is Login, as shown in Figure 1. This feature is still under development and currently requires that a user (sales agent) enters their Username and Password to be validated prior to successfully log into the system. Security features such as password encryption/hashing will be implemented at a later stage (and not as part of this assignment). Once logged in, a user is taken to the Booking List screen to see their associated customer bookings for the various theatre

performances.

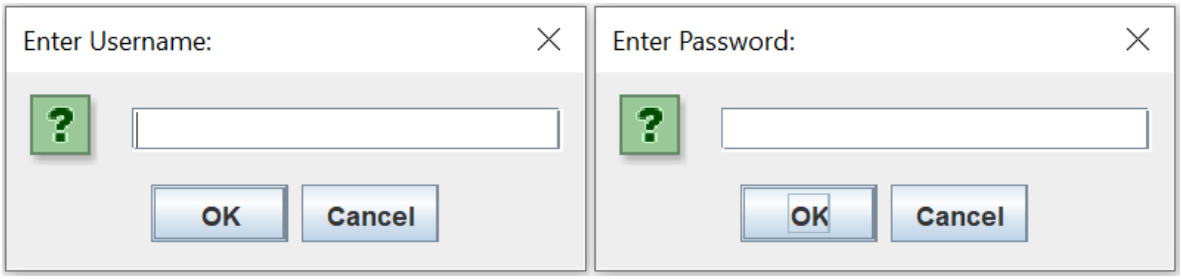

Figure 1 – Login form

Viewing Booking List

Once a user has logged in, they are shown a list of all their associated bookings, as illustrated below in Figure 2. This booking list must be ordered by customer's full name. Each booking may or may not have an assisting sales agent, and can also have a customer name, a performance name, performance date, and a description of booking instruction. A booking is associated with a sales agent if the sales agent is recorded as the user assisting the customer with his/her theatre performance booking.

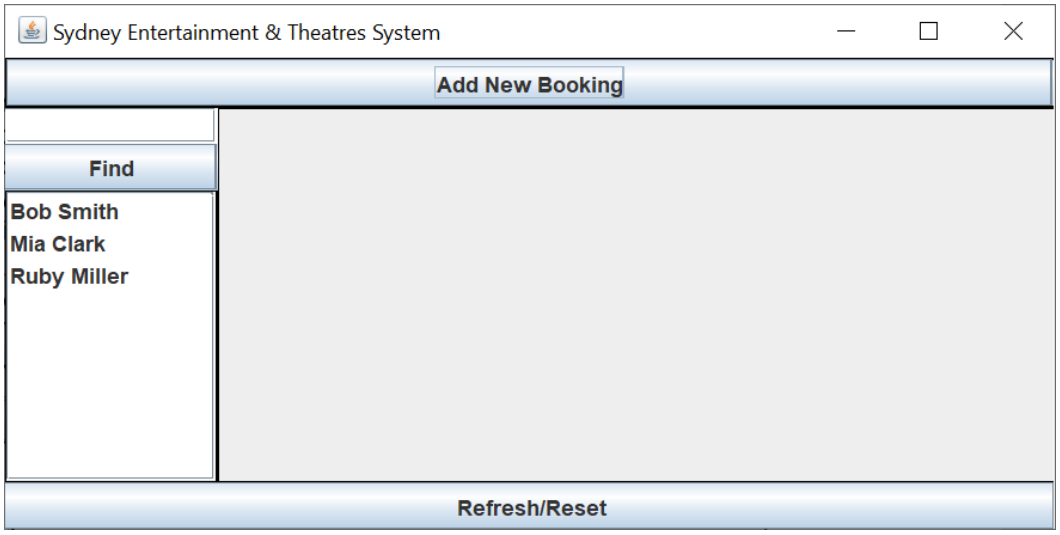

Figure 2 – Viewing Booking List

Finding Bookings

A user can search through all bookings by entering a word or phrase (a 'keyword') in the field above the Find button, as shown in Figure 3, and then clicking on Find. When such a keyword is specified, then only bookings including this word or phrase in their customer full names, booking agent full names, or performances will be retrieved and shown in the list. The search must also ignore case sensitivity. For example, given the search keyword 'lion', Find will return all bookings that include the word 'lion' in their customer full names, booking agent full names, or performances. Searching with a blank/empty keyword field will show all of the logged in user's associated bookings. Any search results returned must be ordered by customer's full name.

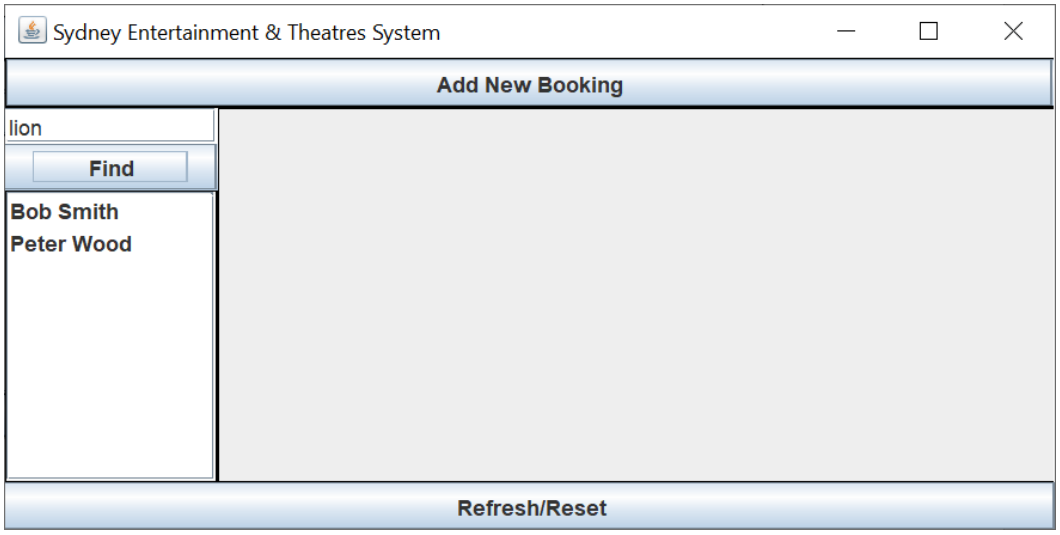

Figure 3 – Finding Bookings

Viewing a Booking's details

Selecting the full name of a customer from the booking list will present details for that booking, as shown in Figure 4. The description of booking's instruction appears in the large text area in the bottom-right part of the form.

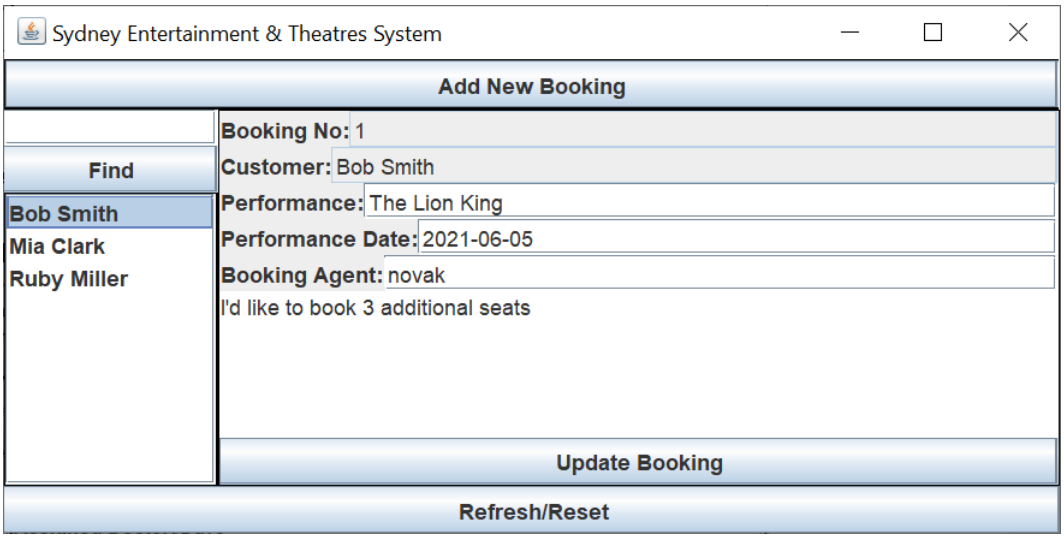

Figure 4 – Viewing a Booking's Details

Adding a Booking

Users may also add a new booking by clicking on the Add New Booking button, entering booking details in the popup dialog that appears, and then clicking on Add Booking, as shown in Figure 5. User must enter a dash ('-') as the Booking Agent if there is no agent associated with that booking.

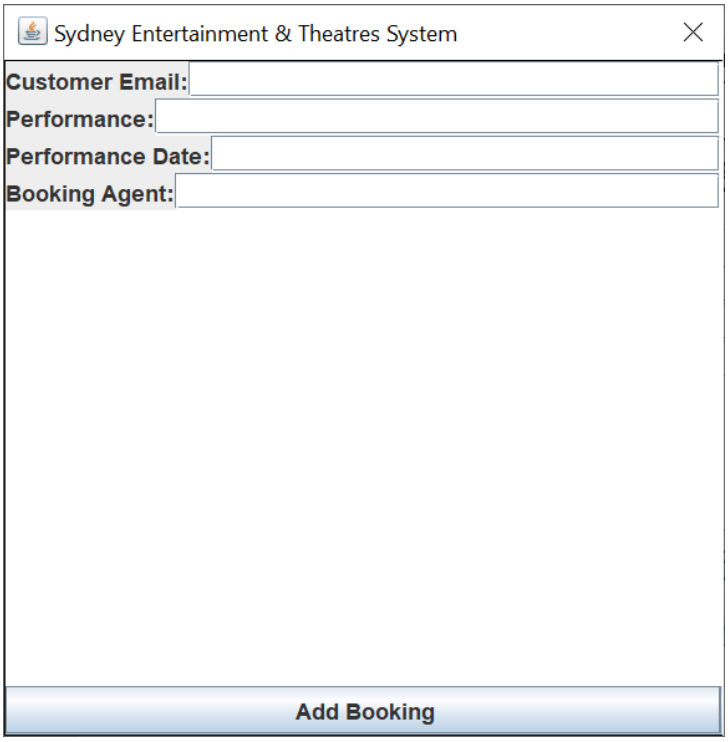

Figure 5 – Adding a Booking

Updating a Booking

Users can also update a booking by modifying data in the booking details screen, as shown in Figure 6, and clicking on the Update Booking button.

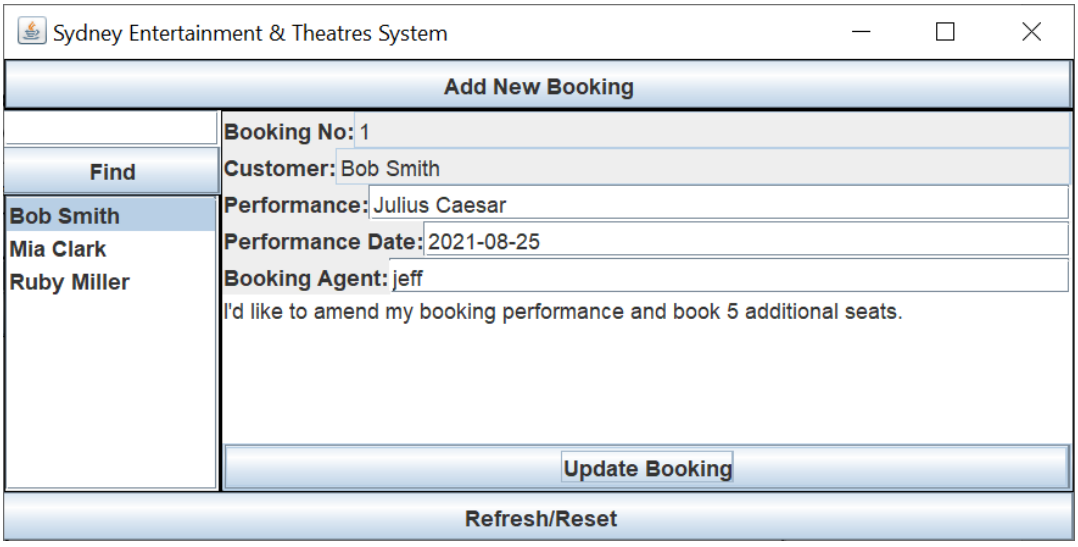

Figure 6 – Updating a Booking

The Refresh/Reset button can be used to refresh the booking details from the database, allowing the user to check that their changes have been made, or to see if a booking has been updated.

Database Interaction Code

The files that are needed for the Java version of assignment are as follows:

1. **`SETSchema.sql`:** a file which contains SQL statements you need to run to create and initialise the SET system database, before starting the application https://canvas.sydney.edu.au/files/15805311/download?download_frd=1
2. **`Assignment2_JavaSkeleton.zip`**: a zip file encapsulating the Java project for the SET system https://canvas.sydney.edu.au/files/16327665/download?download_frd=1

To look through the SET system code, you'll need to import the Eclipse project archive file into Eclipse. Please begin by trying to import the project into Eclipse using the steps below:

1) Right Click the zip file containing the assignment 2 archive > 7zip > Extract Here.

2) Start Eclipse and choose a new location for your workspace in a new folder.

3) Click on File > Import > General > Existing Projects into Workspace > Next.

4) Browse to the folder you extracted zip file to, and then click Select Folder.

5) Ensure the Assignment 2 Project is ticked in the Projects box, and click Finish.

If you experience any difficulties importing the eclipse project, ask your tutor or lecturer for assistance.

Once you import the project in Eclipse, you should notice that there are 3 main packages in the solution: Presentation, Business, and Data. The UI (in the Presentation package) is currently coded to invoke logic in the business layer (eg: see BookingProvider class in the Business package), which in turn delegates responsibility for interacting with the database to an appropriate repository in the data layer (eg: see PostgresRepositoryProvider in the Data package). Notice that separating concerns in this way makes it easier to write a different RepositoryProvider in the data layer (eg: we could write a MySQLRepositoryProvider) if we wanted to change our database management system. In this assignment, you will mainly be working on writing the appropriate queries and SQL commands to fulfil the SET system functionality described in Section 1; where these SQL commands and queries should be written in the PostgresRepositoryProvider. You should use the correct username/password details as specified in tutorial 7. The application's main method can be found in the BookingTrackerFrame class. You can run this main method to run the entire application by right clicking the BookingTrackerFrame file on the Project Explorer > Run As > Java Application.

Section 2: Your Task

Core Functionality

In this assignment, you are provided with a Java skeleton project that must serve as the starting point for your assignment. Your task is to provide a complete implementation for the file **`PostgresRepositoryProvider.java`**, as well as make any modifications necessary to the database schema (i.e., **`SETSchema.sql`**). Specifically, you need to modify and complete these functions:

1. **`checkUserCredentials`** (for login)
2. **`findBookingsBySalesAgent`** (for viewing booking list)
3. **`findBookingsByCustomerAgentPerformance`** (for finding bookings)
4. **`addBooking`** (for adding booking)
5. **`updateBooking`** (for updating booking)

Note that, for each function, the corresponding action should be implemented by issuing SQL queries to the

database management system. If you directly output the result without issuing SQL queries, you are considered as cheating, and you will get zero point for the assignment.

Marking

This assignment is worth 12% of your final grade for the unit of study. Your group's submission will be marked according to the attached rubric.

Group member participation

If members of your group do not contribute sufficiently you should alert the unit coordinator as soon as possible. The course instructor has the discretion to scale the group's mark for each member as follows:

Level of contribution	Proportion of final grade received
No participation.	0%
Full understanding of the submitted work.	50%
Minor contributor to the group's submission.	75%
Major contributor to the group's submission.	100%

Marking Rubric

Your submissions will be marked according to the following rubric, with a maximum possible score of 12 points.

	Part marks (0 – 1)	Full marks (1.5 – 2)
Login	Can correctly login the user 'novak' and validate his username and password.	All valid users can be logged in successfully, and unsuccessful user logins should be rejected.
View Booking List	Correctly list all bookings associated with user 'novak' in the correct order (see Figure 2).	Correctly list all bookings associated with a user, in the correct order, for all possible username input from Figure 1.
Find Booking	Correctly list bookings for keyword 'lion' (see Figure 3).	Correctly list bookings for all possible keywords.
Add Booking	Can correctly add a booking ('abrown87@outlook.com', 'The Lion King', '04/07/2021', 'novak', 'Please book 5 house seats') to the database.	Can correctly add all valid bookings to the database. Bookings associated with invalid customer or sales agent should be rejected.
Update Booking	Can correctly update the instruction of a booking as shown in Figure 6.	Can correctly update details of all bookings, ensuring the associated sales agents are valid.
Stored Procedure	A stored procedure is correctly created in the submitted SQL file.	A stored procedure is correctly created in the submitted SQL file, and is correctly called in one of the specified functions.