Lab 7: Database Connectivity with JDBC

Due date: Friday, November 30, **end of the day**

Assignment Preparation

This is a small teams lab. Please form three-person teams for this assignment. I will allow for two four-person teams OR one two-person team (we have 41 students in class). Team formation is your responsibility, if you do not find teammates due to missing class/lab period, it is your responsibility to complete the full assignment.

Lab Assignment

In a Nutshell

Each group is to build a Java JDBC application that implements a simple Bed & Breakfast Inn reservation system. You will be using the data from the INN dataset for this lab.

The Task: Overview

The INN dataset is designed to provide support for a simple hotel reservation system. For this lab, you will build a Java application that implements three components of such a system: the room reservation functionality for the prospective hotel guests (the GUEST subsystem), the browsing and analytical functionality for the hotel owner (the OWNER subsystem) and a number of convenience functions to make it easier to set up, reset and manage the system (the ADMIN subsystem).

Detailed requirements are outlined below.

General Requirements

GR-0. Filenames. Your system shall be implemented in Java. Name your main class `InnReservations` and your main program `InnReservations.java` respectively. The structure of your applications and inclusion of any other files is left up to you. You are required to submit a `README` file which explains how to compile and run your program from command line (note: even if you are developing using an IDE, please make sure you know how your code runs from command line. You can include a `make` file, if you want to).

GR-1. Dataset. Your system shall utilize the `INN` dataset. It shall store all data on the `ora10g` Oracle server, and shall use JDBC functionality to connect to the database and update/query it.

GR-2. DBMS Access. Your system shall access the Oracle server using JDBC via the following process. The information about the server URL, login Id of the user and the password shall NOT be hard-coded into your program. Instead, your program, upon startup shall read the contents of a file named `ServerSettings.txt` located in the directory from which your program is being run. `ServerSettings.txt` is organized as follows. It has three lines of text in it. The first line is the URL of the Oracle server to which the program will be connecting. The second line is the login Id (username) of the user, whose account the system will use to access the data. The third line is the password for that account.

If you are connecting to the `ora10g` server as user `scott` whose password is `"tiger"`, then the contents of the `ServerSettings.txt` file will be

```
jdbc:oracle:thin:@hercules.csc.calpoly.edu:1522:ora10g
scott
tiger
```

GR-3. UI. Your system shall have a graphical user interface capable of supporting user interactions with the software. Specific UI design is left to individual teams. Intuitively, there are three main components to the system, which assume three different categories of users for them. For simplicity, all three subsystems shall be unified under the same application (to make grading easier), but within the application, they shall be kept separate. You can implement this, for example, by using a main application window with three tabs in it, each containing the functionality of a different subsystem.

GR-4. Startup. Upon startup, your system shall perform the following activities (not necessarily in the order listed below):

- Read the contents of the `ServerSettings.txt` file.

- Establish connection with the Oracle server using credentials provided in the file.
- Check if the tables from the INN dataset already exist on the account. If they do not exist, create (but do NOT populate) them.
- Display the main UI of your program. The main UI must allow access to all three subsystems as well as provide a graceful exit option.
- Perform a check of whether the INN database is full or empty. A full database is a database that contains the tuples from the INN dataset and possibly some more tuples. An empty database is a database that has both Rooms and Reservations table, but the tables contain no records[1].
- Wait for user activity.

ADMIN subsystem

AR-0. Overview. The ADMIN subsystem is designed to make it easy to deploy the application and to reset the database state. The following functionality shall be provided for the ADMIN subsystem:

- Current status display.
- Viewing facility for the contents of each of the INN dataset tables.
- Ability to clear the database of all records.
- Ability to load and/or reload the contents of the database.
- Ability to destroy the database (i.e., drop all the tables).

Each function is described below.

AR-1. Current Status Display. The main UI for the ADMIN subsystem shall be the current status screen, displying the general information available about the dataset at present moment. All other functionality of the ADMIN subsystem shall be accessible from the current status display. The following information shall be made available:

- *Database Status.* Either *full* or *empty* or *no database* (note, that the latter status will never happen upon startup, but can happen under other circumstances).
- *Reservations.* Total number of tuples in the reservations table.
- *Rooms.* Total number of tuples in the rooms table.

[1] You do not need to look at the contents of every single record to determine if the DB is full. Checking the number of records in each table is sufficient.

AR-2. Table display. The status screen shall provide the facility (a button, a menu item, and so on) to display the contents of the rooms and the reservations tables. For example, two buttons, "View Rooms table" and "View Reservations table" can be used. When the user selects to view one of the database tables, a SQL query retrieving the full contents of the table is executed, and the results are displayed in a specially-designed for this purpose UI (as designed and implemented by each team).

AR-3. Clear database. A "Clear DB" button shall be provided on the current status display. When this button is pressed, SQL commands deleting the contents from both the rooms and the reservations tables shall be executed. As the result of the "Clear DB" command, the database becomes empty and this information needs to be immediately propagated to the current status display.

AR-4. Load/Reload DB. A "Load DB" and/or "ReLoad DB" button or buttons (or other UI constructs facilitating the same functionalioty) shall be provided bon the current status display. When this functionality is engaged, the system shall behave as follows:

- The system checks if the current database exists and is full. If the database is full, your system shall return back to the current status display. The status display shall contain a diagnostic message indicating that the table(s) already exist and contain information in them.
- If the database exists but is empty, the system shall insert into the database the full contents of the `INN` dataset.
- If the database does not exist, your system shall execute SQL commands creating the database and then populating it with the records.

Upon completion, the system returns back to the status display screen, which should reflect the current state of the database.

AR-5. Database Removal. When this functionality is engaged, your system shall drop the rooms and reservations tables and return to the status display screen, which should show whether the deletion was successful.

AR-6. Switch subsystem. The ADMIN subsystem UI shall provide the ability to switch between the subsystems at any moment of time. That is, a user of the system shall be able at any moment to switch from the ADMIN subsystem to either the GUEST or the OWNER subsystem.

OWNER subsystem.

OR-0. Overview. This subsystem is designed to allow the Bed & Breakfast management to monitor the overall state of the reservations. The main subsystem UI shall list the functionality available within the subsystem and allow the user to select desired functionality. The following functionality shall be available to the user in the `OWNER` subsystem.

OR-1. Occupancy overview. When selected, your system shall do the following:

- Bring up a "fresh" UI screen containing UI elements into which the user can enter one date or two dates.
- When a user enters a date (Month, Day - use 2010 year by default), your system shall display a list of rooms. For each room the system shall specify, whether it is occupied or empty on the given date. If a user selects an occupied room and presses "Enter" (or clicks a mouse button), the system shall display full information about the reservation that covers the given day (see below).
- When a user enters a pair of dates (start date and end date), your system shall display a list of rooms. For each room the system shall specify, whether the room is fully occupied during the time period (i.e., someone is staying in the room every night of the time period), partially occupied (someone is staying in the room part of the time, but not every night of the time period) or empty (no one has reserved the room for any night of the time period).

  If a fully occupied or partially occupied room is selected by the user and "Enter" or mouse button is pressed, the system shall display a brief list of reservations the cover the time interval. Selecting a specific reservation from the list shall yield the display of full information about the reservation.

Note: try getting the output for each of the two cases described above using a single SQL query per task.

OR-2. Revenue. When this option is selected, your system shall provide a month-by-month overview of the reservations volume and the cash flow. For the purpose of this assignment, all revenue from the reservation is assigned to the month when the reservation commenced. For example a seven-day hotel stay that started on October 30 will be treated as October revenue.

Your system shall display a list of rooms, and, for each room, 13 columns of numbers. The numbers displayed shall be one of the following:

1. reservation counts, or
2. total nubmer of days occupied, or
3. the dollar revenue for each month.

The final column will totaling the numbers for each room. There shall also be a "totals" row in the table, which adds up the numbers for each month. Your UI shall provide the option of selecting which set of numbers, reservations, days of occupancy, or revenue is to be displayed, and shall switch the numbers, when a different selection is made.

OR-3. Reservations. When this option is selected, the user shall be able to browse the list of reservations. The system shall provide the means for the user to enter the start and the end day for the search, and/or select a specific room, and shall report all reservations that commence within the specified time period and are for a selected room. When the user selects a reservation, detailed information about it shall be displayed.

OR-4. Rooms. When this option is selected, the system shall display the list of rooms. Two options shall be made available: viewing full information about a room and viewing the list of reservations for a given room.

When the user selects to view a specific room, a screen containing information about the room shall be displayed.

When the user selects to view the reservations for the room, a screen displaying a list of reservations for the selected room sorted in chronological order shall be displayed. When the user selects a reservation, the detailed reservation screen shall be displayed.

OR-5. Detailed reservation information. When a reservation is selected from a list of reservations (see OR-1, OR-3, OR-4), a display shall appear that shows the full details of the reservation. The display shall show the contents of every attribute from the reservations table (as well as the correct name of the room, and any extra information about the room you want to add).

GUEST subsystem

R-0. The `GUEST` subsystem shall provide the functionality for the putative hotel guest to reserve a room. Following the traditions of hotel web sites, the following options shall be offered to the guest:

- Rooms and Rates.
- Reservations.

R-1. Rooms and Rates. When this option is selected, the system shall output the list of rooms to the user. When a room is selected from the list, the system shall show a detailed information screen, similar to the screen shown to the owner (see **OR-4**.) The information screen shall contain specific details about the room, and a "Check Availability" button.

R-2. Checking Room Availability. When the user presses "Check Availability" button, the system shall offer the following dialog to the user. The system shall provide the means for the user to enter a pair of dates (check-in and check-out). Once the dates are entered, the system shall output information about whether or not the room is available on each of the nights of the putative stay[2]. If the room is available for a given night, the system shall display the room rate (see **R-3** for room rate computation), If the room is unavailable, the system shall display "Occupied" for that night. If the room is available for the duration of the selected stay (i.e., available on each night), a "Place a Reservation" button shall be provided for the user to complete the reservation.

R-3. Pricing. The price of one night of stay is determined as follows. On a weeknight, the price of the stay is the base rate of the room. On a weekend, the price of the stay is 110% of the base for the room. For any reservation or putative stay that covers the following nights: January 1, July 4, September 6, October 30, the nightly rate for each night shall be 125% of the base for the room.

These rates can get a 10% AAA or a 15% AARP discount adjustment during the reservation time.

R-4. Reservations. When the user chooses the "Reservations" option from the main `GUEST` subsystem menu, the system shall offer the user to enter the putative dates of the stay. When the dates are entered, the system shall output the list of rooms, that have availability **for each night of the stay**. For each such room, the room name, and the applicable nightly rate (see **R-3**) shall be displayed.

When the user chooses a room, a screen containing detailed information about the room (similar to that in **R-2**) shall be displayed, with the "Make a Reservation" button available.

R-5. Completing a reservation. When the user chooses to complete a reservation, either through the "Rooms and Rates" or "Reservations" subtasks, the system shall display the reservation completion form. In this form, the system shall ask the user to specify the following details (we will not worry about the payment options in this system, so only the details for the Reservations table in the `INN` database shall be collected):

- Name (last, first) of the hotel guest.
- Number of adults staying in the room.
- Number of kids staying in the room.

[2]Note: you may have to execute a separate query for each night of the stay to get this information.

- Applicable rate discounts.

The total number of guests in the room shall not exceed the room capacity. If it does your system shall provide appropriate feedback and wait for the guest to change the numbers.

There are two applicable rate discounts that can be applied to the nightly rate computed via **R-3**. AAA gives a 10% discount. AARP gives a 15% discount. The user can select one of the two (but not both) discounts, or no discount.

The system shall provide "Place reservation" button, which shall become visible/clickable once the user enters all necessary and correct information.

R-6. Updating the database. Once the user hits the "Place reservation" button, the system shall do the following:

- Generate a six-digit reservation code. The code must be unique - i.e., not found in the current reservations table.
- Insert a record about the new reservation into the reservations table.
- Display a "Your reservation is complete!" message which includes the details of the reservation and the reservation code.

The system then shall provide the facility for the user to return back to the main screen of the GUEST subsystem.

Submission

Submit one set of deliverables per team. The deliverables include:

- the entire code base for your application;
- a `README` file which includes (a) the list of all members of the team, (b) any compilation/running instructions and (c) information about any known bugs and/or deficiencies.

Submit the lab either as a `lab7.zip` or `lab7.tar.gz` archive using `handin` as follows:

```
$ handin dekhtyar lab7 lab7.ext
```

Good luck!