Test Project

IT Software Solution for Business

Session 3

Submitted by: Independent Test Project Design Team

Introduction

Seoul Stay is the first and only platform that allows international travelers from around the world to rent the best homes, estates, or condominiums in Seoul.

The project designers are asking for a management interface for the call center and the people operating the phones. They will use this system to assist both home-owners / managers and travelers and renters.

In this session the project designers are asking for a desktop application and will provide you with the schematics and the initial required data.

Contents

This Test Project proposal consists of the following documentation/files:

1. WSC2022_TP09_S3_EN.pdf   (Session 3 instructions)
2. Session3-MySQL.sql   (SQL Script to create tables with data for MySQL)
3. Session3-MsSQL.sql   (SQL Script to create tables with data for Microsoft SQL)

Description of Project and Tasks

While developing the test project, please make sure the deliverables conform to the basic guidelines drawn out by the project designers:

- There should be consistency in using the provided style guide throughout development.
- All required software modules must have applicable and useful validation and error messages as expected by the industry.
- Offer a scrollbar if the number of records on a list or a table that do not fit in the form area comfortably. Hide scrollbars if all content can all be displayed in the designated area.
- The de-facto standard, ISO compliant date format is DD/MM/YYYY which will be used in this task where applicable.
- Where applicable, use comments in code to have the code more programmer-readable.
- The use of valid and proper naming conventions is expected in all material submitted.
- Any form or report once created should be displayed in the center of the screen.
- When a form or a dialogue is in focus, operations on other forms need to be suspended.
- The caption of Delete and Cancel buttons need to be in Seoul red to help with accidental mishaps and be in line with the brand style.
- When using colors to differentiate between rows or records, there needs to be visible clarification on the screen as to what they stand for.
- The wireframe diagrams provided as part of this document are only suggestions and the solution produced does not have to in any way, mirror what has been presented
- Time management is critical to the success of any project and so it is expected of all deliverables to be complete and operational upon delivery.

Instructions to the Competitor

3.1 Creating the database

Create a database by the name of "Session3" in your desired RDBMS Platform (MySQL or Microsoft SQL Server). This will be the main and only database you will use in this session.

3.2 Importing the structure

Depending on your preferred RDBMS platform, a SQL scripts is made available. The said scripts consist of the database structure and data required to complete the required tasks. The data needs to be imported to the database created for this session named "Session3".

As instructed by the designers, the database structure provided for the purpose of this section cannot be altered. This applies to removal of tables, adding or deleting any fields on the tables or of change in their data types.

Note: The database table and field details will only be provided in the formal test project.

To help further perceive the thinking behind the structure of the database, the database designers provide an Entity-Relationship Diagram (ERD). The included diagram explains the conceptual and representational model of data used in the database.

Since this is the first time you are working with the dataset, the designers have chosen to provide a short description of some of the entities used in the database:

TYPE	TITLE	DESCRIPTION
Table	Scores	Where base information on scoring of properties is stored.
Table	ItemScores	Where the scores awarded by the clients are stored.

3.3 Simple search form

This is the initial form when opening the software. Call center employees do not need to login to look and operate this form. The following is a short description of the characteristics of this form:

- Place a search box (which can be left unfilled) on the form that can be used to search the database for any of the following fields:

  - Area (suburb) name.
  - Name of attraction (landmarks).
  - Name or title of the property (listing).
  - Type of property (listing) e.g. Apartment etc.
  - Name of amenities e.g. Parking.
  - Place input boxes (must be filled) that would allow the client to filter based on the following:
  - The date of the start of the reservation with an option to choose it from a date picker. The value of this fields cannot be before the current system date.
  - The duration of the stay as in the number of nights the reservation will be for. The value of this field is a number between and not including 0 and 15.
  - The number of people (capacity) the properties or listings can support (capacity). The value for this field cannot be less than 1.

- Place a button with a magnifying glass icon that when pressed directs the client to the results of the search as documented on section 3.4.

- Place a button captioned "Advanced Search" that directs the client to the form documented on section 3.5 of this document.

- Create an autosuggest live feature after 3 characters are typed into the search box that allows the client see if the phrase they are looking for is found in different places and allows them to choose one of the listed items as their search criteria.

```
Seoul Stay - Search Results
┌─ Simple search ──────────────────────────────────────────────┐
│  Seoul Tower                                    🔍 Search properties...
│
│  From: [ / / ]  📅   for  [3 ▲▼] nights  and  [2 ▲▼] people     ⩔ Advanced Search
└──────────────────────────────────────────────────────────────┘
┌─ Search results ─────────────────────────────────────────────┐
```
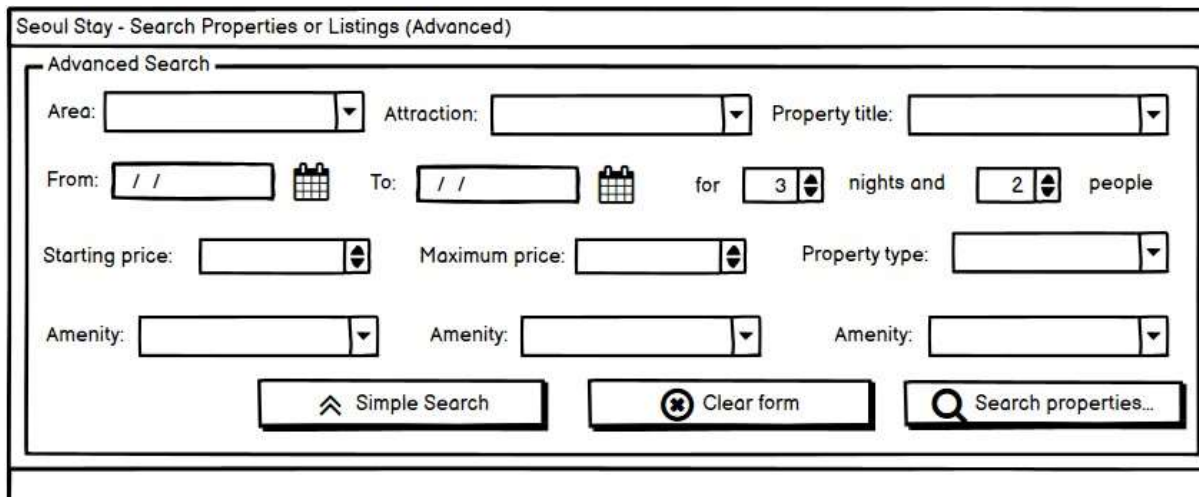
Property ⬍	Area ⬍	Average Score ⬍	Total completed reservations ⬍	Amount payable ⬍
White Swan	Seoul Tower	4.8	40	30$
Red hut	Seoul Tower	3.9	22	42$
Blue Sky	Seoul Tower	3.8	1	35$
Phoenix 38	Seoul Tower		0	40$

```
Displaying 4 options
```

3.4 Search results form

This form can be an extension of the simple search form or can be a completely new form that will display the results of the search criteria submitted. Here is a short description of the layout and the elements that are shown:

- The list of results in a grid or similar that includes the following data:

  - Title of property or listing.
  - Name of the area or suburb each property resides in.
  - The average score given to each property or listing based on the user feedback stored on the table "Scores".
  - Total number of completed reservations or bookings for the property or listing until the current date of the system. This number should not include cancelled bookings.
  - The amount payable is the total amount that needs to be paid to reserve the property for the selected date and period.

- The list should initially be sorted ascending using the property title but can be sorted by the client based on each of the above data fields both ascending and descending.

- The form should have all the buttons and inputs that were documented on section 3.3 of this document as the client can change their search criteria from this form as well.

- The number of results should be placed on the bottom of the form on the status bar.

3.5 Advanced search form

Using this advanced search form, the client is able to define more specific criteria. The following is a short description of the characteristics of this form:

- Place input boxes so that the client can search based on each of these data fields.:

  ○ Area (suburb) name as a drop-down. If an area is selected, the input fields on "Attraction" and "Property title" will turn to drop down boxes populated with attractions and properties that are located within the area. This field can be left unfilled.

  ○ Attraction which is a normal text box input by default and turns to a drop-down when an area is selected by the client. This field can be left unfilled.

  ○ Property title which is a normal text box input by default and turns to a drop-down when an area is selected by the client. This can be left unfilled.

  ○ Two date picker type input boxes that allows the client to select and enter a date range. This date is then used as starting and ending dates for the available properties. It goes without saying that the starting date must not be before the current date on the system and the ending date cannot be before the starting date. These dates cannot be left unfilled.

  ○ The duration of the stay as in the number of nights the reservation will be for. The value of this field is a number between and not including 0 and 15.

  ○ The number of people (capacity) the properties or listings can support. The value for this field cannot be less than 1.

  ○ Two input boxes to define a price range with a starting or minimum price and a maximum price in dollars with two decimal points. These can be left unfilled.

  ○ Type of property (listing) as a drop-down which can be left unfilled.

  ○ Three input drop-down boxes for amenities that are populated with all the records from the database where they may choose different and not the same amenities. This means if one amenity is chosen on one of the boxes, the other ones should not have them available for selection. The client may choose not to select any amenities.

- Place a button captioned "Clear form" which resets all the filters and clears the values entered.

- Place a button with a magnifying glass icon that when pressed directs the client to the results of the search as documented on section 3.6.

- Place a button captioned "Simple Search" that directs the client to the form documented on section 3.3 of this document.

Seoul Stay - Search Properties or Listings (Advanced)

Advanced Search

| Area: [_____ ▾] | Attraction: [_____ ▾] | Property title: [_____ ▾] |

| From: [/ /] 📅 | To: [/ /] 📅 | for [3 ⇕] nights and [2 ⇕] people |

| Starting price: [____ ⇕] | Maximum price: [____ ⇕] | Property type: [_____ ▾] |

| Amenity: [_____ ▾] | Amenity: [_____ ▾] | Amenity: [_____ ▾] |

[⌃ Simple Search] [⊗ Clear form] [🔍 Search properties...]

Search results

Property ⇕	Area ⇕	Average Score ⇕	Total completed reservations ⇕	Amount payable ⇕	Check-in at ⇕
White Swan	Seocho-gu	4.8	40	30$	20/11/2022
White Swan	Seocho-gu	4.8	40	30$	21/11/2022
White Swan	Seocho-gu	4.8	40	35$	27/11/2022
White Swan	Seocho-gu	4.8	40	37$	28/11/2022
Red hut	Seocho-gu	3.9	22	42$	20/11/2022
Blue Sky	Seocho-gu	3.8	1	35$	20/11/2022
Phoenix 38	Seocho-gu			40$	20/11/2022
Phoenix 38	Seocho-gu			45$	25/11/2022

Displaying 8 options from 4 properties

3.6 Advanced Search results form

This form can be an extension of the advanced search form or can be a completely new form that will display the results of the search criteria submitted. Here is a short description of the layout and the elements that are shown:

- The list of results in a grid or similar that includes the following data:

  - Title of property or listing.
  - Name of the area or suburb each property resides in.
  - The average score given to each property or listing based on the user feedback from the database.
  - Total number of completed reservations or bookings for the property or listing until the current system date. This number should not include cancelled bookings.
  - The amount payable is the total amount that needs to be paid to reserve the property from the date indicated as check-in date for the duration indicated in the search form.
  - Check-in date is the starting date for the reservation. So, for example if the caller to the call center asks for a stay for a certain number of nights in a date range, they can get different pricing options on a single listing based on their check-in dates.

- The list should initially be sorted ascending using the property title but can be sorted by the client based on each of the above data fields both ascending and descending.

- The form should have all the buttons and inputs that were documented on section 3.5 of this document as the client can change their search criteria from this form and get results on this same form.

- The number of options available and the total number of properties should be placed on the bottom of the form on the status bar.

3.7 Write Unit Test

Unit Testing is a crucial step in software design and implementation. It not only improves the efficiency and effectiveness of the code, but it also makes the code more robust and reduces the regressions in future development and maintenance.

Unit Testing is a methodology of testing source code for its fitment of use in production. We start out writing unit tests by creating various test cases to verify the behaviors of an individual unit of source code.

- Implementation of unit test for:

  ○ No property or listing should have availabilities before 19/02/2017.

- Implementation of unit test for:

  ○ No score associated with any of the properties or listings should have higher score than 5 and lower score than 0.

- Implementation of unit test for:

  ○ The system can never have duplicate users registered.

Develop the unit test using the built-in testing feature in your IDE.