###### River’s Edge Hotel

Now that you finished developing the database backend, it is time to design and develop a front end for the application. You will create forms and a menu to allow users to access your database. Listed below are the requirements for the forms and menus.

**Please be aware that the appendix is an integral part of the lab requirements.**

**Please do not put your name or any other identifying information on your forms.**

**NOTE:** Prior to starting Lab 2, ensure your Lab 1 tables are correct. You will also need to **drop** all procedures, functions, packages, and database triggers associated with your Lab 1 tables.

P => Form Property, T => Form Trigger

1. **Source Code / Executables / Submission 5 marks**

* Ensure you back up your source code files regularly on a medium separate from the Virtual Machine
* Each form will be in its own physical file
* Each data block will be based a single table, do **NOT** use table joins or views when creating a data block!
* Note: master-detail relationships are not considered table joins
* All multi-line forms MUST have all the data from one record on a single line
* Create all your forms and menu in the OracleDocs\Marking directory
* When you are ready to start fully testing your lab: leave the executable files in OracleDocs\Marking, **move** all your source code files to the OracleDocs\Source directory, open an Internet Explorer browser, and type the following URL:

http://localhost:7001/forms/frmservlet?form=C:\OracleDocs\Marking\Lab2Q6.fmx

* When you are done your testing, submit your lab via Moodle as follows:
* Copy your executable files to “Lab 2 Marking Files Submission”
* Copy your source code files to “Lab 2 Source Files Submission”
* Ensure you only submit your final Lab 2 files (i.e. NO classroom, test files or versions of Lab 2) and do **not** compress any files
* Email your instructor a discussion about the lab including, but not necessarily limited to, the following:
* What you liked and/or disliked about the lab
* How long the lab took to complete (divided into “in class” and “out of class” time)
* Any significant problems you encountered
* Whether there was enough material provided in the lectures to complete the lab
* What, if any, changes, or improvements could be made to the lab
* Anything additional as specified by your instructor

1. **Customer Maintenance 15 Marks**

* This form will be used to insert new customers as well as query and update, but not delete, existing customer information
* Name this form **Lab2Q2** (P)
* The form requirements are:
* Display all fields from the Customer table
* Display one record in a form style layout
* The province will be displayed using a list item displaying province names, not abbreviations (P)
* Valid province values will be the thirteen Canadian provinces and territories – do not allow Nulls (P)
* The default province will be Alberta (P)
* Ensure the postal code field is entered in upper case (P)
* When records are queried, they should be displayed sorted by last name (P)
* The customer source code will be available in a List of Values (LOV) (P)
* Display both the customer source code and customer source description
* Sort the customer source description in descending order
* Do not allow the user to enter a customer source code that does not exist in the LOV
* The form must open with all records retrieved (T)

1. **Room Maintenance 15 Marks**

* This form will be used to insert new rooms as well as query, update and delete existing room information
* Name this form **Lab2Q3** (P)
* The form requirements are:
* Display the room number, smoking status, pet status, room rate, room type code and room type description in one block
* Display 4 records at a time in a tabular style layout with a scrollbar on the left side of the data
* Users will not be able to navigate to the room number field unless they are in query mode (P)
* A check box (with an appropriate label) will be used to display the smoking status where checked implies non-smoking and unchecked implies smoking (P)
* Attach a List of Values (LOV) to the room type code field (P)
* In the LOV display the room type description only, do not **display** the room type code
* Do not allow the user to enter a room type that does not exist in the LOV
* When a record is accepted, ensure the room type code and room type description fields are **both** populated
* The room type description field will be read only and must be populated when an existing record is retrieved (T)
* Suppress room 0 (zero) when querying records (P) Where Clause on Datablock
* When a new record is **saved,** the room number will be generated from the sequence “Seq\_Room” (T)

1. **Invoice Entry / Maintenance 35 Marks**

* This form will be used to insert new invoices and/or invoice items as well as query, update and delete existing invoice and/or invoice item information
* Name this form **Lab2Q4** (P)
* The form requirements are:
* Invoice Information
* Display the invoice number, registration number, customer first name, customer last name, invoice date, paid status, invoice total and invoice GST fields in one block
* Display one record at a time in a form style layout
* Only invoices from **current** registrations (i.e. not cancelled or no shows) will be allowed to be retrieved into the form (P)
* Ensure the customer first name and customer last name are populated when an existing record is retrieved (T)
* In data entry mode, the user may not navigate to the invoice number, invoice date, customer names, invoice total or invoice GST fields (P)
* The invoice number and invoice date will be determined programmatically when a new record is **saved**: (T)
* The invoice number will be generated using the sequence “Seq\_Invoice”
* The invoice date will be set to the current system date
* Ensure the invoice date field displays both the date **and time** (P)
* Display the paid status using two radio buttons (Paid and Unpaid) (P)
* There will be a LOV on the registration number so the user can select from a list of **current** registrations (P)
* Do not allow the user to enter a registration that does not exist in the LOV
* Have the LOV sorted by arrival date
* After a registration is accepted ensure that the registration number, customer first and last name fields are all populated
* Invoice Item Information
* Display the item number, item description, quantity sold and selling price in one block
* Display three records in a tabular layout with a scroll bar
* Do not **display** the selling cost
* Add an extended price field (quantity sold \* selling price) and an extended price total field (sum of extended price) to the Invoice Item block (P) or (T)
* There will be a LOV on the item number (P)
* The LOV will display the item description sorted alphabetically
* Do not allow the user to enter an item that is not in the LOV
* When an item number is selected ensure the item number, description, selling cost (Current\_Cost) and selling price (Current\_Price minus Discount) are populated
* In data entry mode the user will NOT be able to navigate to the item description, selling price, extended price or extended price total fields (P)
* Have the item description filled correctly when:
* A new item number is entered, or (P) and/or (T)
* An existing record is retrieved, or (T)
* An existing item number is changed (P) and/or (T)
* Have the selling price filled correctly (current price minus discount) when: (P) and/or (T)
* A new item number is entered, or
* An existing item number is changed
* Have the extended price filled correctly when: (P) and/or (T)
* A new record is entered, or
* An existing record is retrieved, or
* An existing record is changed
* Have the extended price total filled correctly when: (P) and/or (T)
* A new invoice item record is entered, or
* An existing invoice item record is retrieved, or
* An existing invoice item record is changed
* **Buttons**
* There will be three buttons in addition to the “standard” buttons as described in the Appendix:
* Pay An Invoice – marks the current invoice as paid; if the invoice is already paid, do no processing and inform the user with an appropriate message (T)
* Reverse An Invoice Payment – marks the current invoice as unpaid; if the invoice is not currently paid, do no processing and inform the user with an appropriate message (T)
* Update Invoice Totals – updates the current Invoice\_GST (using 5%) and Invoice\_Total (including GST) fields based on the invoice item information on the form (T)
* All three buttons will perform their action and save changes (if applicable)
* **Navigation**
* In order to make invoice entry easier, set properties so that the following occur:
* When the user tries to navigate to the next field from the last field in the invoice block, the focus will move to the invoice item block (P)
* When the user tries to navigate to the next field from the last field in the current record in the invoice item block, the focus will move to the next record in the invoice item block (P)

1. **Registration / Invoice Display Form 15 Marks**

* This form will display registrations and the corresponding invoice(s) they are associated with
* Name this form **Lab2Q5** (P)
* The user must be able to query but must **NOT** be able to **CHANGE** any data (P)
* You must utilize multiple canvases on this form
* If you use a stacked canvas, you must have functional show and hide buttons
* If you use a tab canvas, you must have data on multiple tab pages (i.e. one data block per tab page)
  + - You may use a content canvas if you choose
* You will need to display the following information:
  + Registration Information (display one record at a time in a form style layout)
    - Registration number
    - Customer first and last names (T)
    - Room number
    - Registration status descriptor (T)
    - The number of invoices a registration is associated with (T)
  + Invoice Information (display three records in a tabular layout with a scroll bar)
    - Invoice number
    - Invoice date
    - Paid flag (using a check box with an appropriate label) (P)
    - Invoice GST
    - Invoice Total
* Display all registrations regardless of status
* Add an **informational** alert to notify the user whenever a registration record is retrieved that has a registration status of **Outstanding** (T)
* If you chose to have the default buttons appear on a canvas, ensure that the SAVE button will either be permanently disabled or not displayed (your choice)

1. **Opening Form 15 marks**

* All the forms and reports will be able to be run from this form
* Name this form **Lab2Q6** (P)
* This form includes command buttons **AND** a drop-down menu
* Buttons
* Customer Maintenance Form – open so the user can navigate between (and operate) the opening form and this form (T)
* Room Maintenance Form – open so the user can navigate between (and operate) the opening form and this form (T)
* Invoice Entry / Maintenance Form – open so the user may not operate any item on the opening form without closing this form (T)
* Registration / Invoice Display Form – open so the user may not operate any item on the opening form without closing this form (T)
* Customer Summary Report – provide a message to the user (**including** the report name) that this report is not currently available (T)
* Invoice Summary Report – provide a message to the user (**including** the report name) that this report is not currently available (T)
* Exit (P or T)
* Drop-down Menu:

|  |  |
| --- | --- |
| Forms | Reports |
| Customer Maintenance | Customer Summary |
| Room Maintenance | Invoice Summary |
| ----------------------------------------- | ---------------------------- |
| Invoice Entry / Maintenance | Exit |
| Registration / Invoice Display |  |
| ----------------------------------------- |  |
| Exit |  |

* Ensure that you include menu separator items (as indicated by the dashed lines)
* A form opened with the drop-down menu will operate in the same manner as if the form was opened by pressing the corresponding command button (P and/or T)

**Appendix**

* The customer wants the standard screen colours changed to something **completely** different than the standard Forms colour (P)
* All forms can be the same colour, or all different colours, your choice
* Ensure the values in all attributes are completely displayed (height and width) (P)
* Each form will have its own menu (unless otherwise specified, Oracle’s default menu is **required**)
* The entire form **must** be visible upon opening
* The close button on the form’s title bar will close the form **regardless** of what mode the form is in (P and T)
* The “Customer Maintenance” and “Invoice Entry / Maintenance” forms will have the following buttons:
* Next Record – moves focus to the next record (T)
* Previous Record – moves focus to the previous record (T)
* Query – if the user is in normal mode, pushing your query button will work the same as pressing the standard Enter-Query button; if the user is in query mode, pushing your query button will work the same as pressing the standard Execute-Query button (T)
* Clear – clears the form (T)
* Save – saves all pending changes (T)
* Exit – closes the form (T)
* You may add the standard buttons to the other forms if you choose
* Do not allow the user to minimize the window (P)
* All objects are to be named appropriately (i.e. “Window1” is not appropriate) (P)
* You will use the following item types appropriately at least once in this lab: text item, display item, list item, check box, radio group, list of values displaying at least two fields, list of values including at least one displayed field and one hidden field

Lab Expectations:

* All monetary amounts will be displayed with a dollar sign, commas, and at least one digit to the left, and two digits to the right, of the decimal place
* All dates will include an easily identifiable month, day and four-digit year
* All code must be executable from within the Oracle DMIT environment at NAIT
* Anything additional as specified by your instructor