4/6/2023

(Student) - Qadeer Hussain C00270632

Customer Invoice Management System

OOSD Project CA3

**Table of Contents**

[**Description** 2](#_Toc131692679)

[**Requirements** 3](#_Toc131692680)

[**Screenshot database tables (Structure & Data)** 4](#_Toc131692681)

[**Customer Table** 4](#_Toc131692682)

[**Product Table** 5](#_Toc131692683)

[**Invoice Table** 6](#_Toc131692684)

[**Cart Table** 7](#_Toc131692685)

[**ER Diagram** 8](#_Toc131692686)

[**Interesting source code snippets** 9](#_Toc131692687)

[**Tests** 12](#_Toc131692688)

# **Description**

I have created a Customer Invoice management System which connects to a MySQL database to store different types of data such as customer, products and customer invoices. This system allows user to add, view, amend, and delete details.

I used a system called XAMPP which launched the MySQL database, I found this system very simple to use and is very manageable. I used different SQL statement such as Select, Update, Delete and Inner joins.

The main Java Swing components of my system I used were JTextfield, JButton, JTable, JScrollPane, JTextArea. The layout of the GUI’s I used were GridBagLayout as it was very precise however it didn’t deem a challenge to use and I also used Absolute Layout.

# **Requirements**

The requirements for this project was to have at least the following

* Backend database
* Front end GUI

Backend database: This has to have Minimum 3 tables such as Customer, Invoice, Product tables, must provide inner join over multiple tables and any database of my choice but it must have a java connector.

Front end GUI: must provide CRUD (create, retrieve, update, delete) operations on the database and you must demonstrate the use of different Java Swings components such as were JTextfield, JButton, JTable etc.

# **Screenshot database tables (Structure & Data)**

## **Customer Table**

**Graphical user interface, text

Description automatically generated**

**Graphical user interface, application

Description automatically generated**

## **Product Table**

**Graphical user interface, text, application

Description automatically generated**

**Graphical user interface, text, application

Description automatically generated**

## **Invoice Table**

**Graphical user interface, text, application

Description automatically generated**

**Graphical user interface, application, table

Description automatically generated**

## **Cart Table**

**Graphical user interface, application

Description automatically generated**

**Graphical user interface, application

Description automatically generated**

# **ER Diagram**

**Diagram

Description automatically generated**

# **Interesting source code snippets**

This piece of code get data from database and inputs it into the separate rows and columns. I found this piece of code interesting as I did not know how to get data from the database into the table.

JScrollPane scrollPane = new JScrollPane();

scrollPane.setBounds(10, 100, 619, 201);

AmendViewCustomer.getContentPane().add(scrollPane);

table = new JTable();

scrollPane.setViewportView(table);

// load the customer data into the table

final String DATABASE\_URL = "jdbc:mysql://localhost/project";

Connection connection = null;

PreparedStatement pstat = null;

ResultSet rs = null;

try {

// establish connection to database

connection = DriverManager.*getConnection*(DATABASE\_URL, "root", "");

// create Prepared Statement for selecting data from the table

pstat = connection.prepareStatement("SELECT \* FROM customer");

rs = pstat.executeQuery();

// create a new TableModel to hold the data

DefaultTableModel customer = new DefaultTableModel();

// add columns to the TableModel

customer.addColumn("CustomerID");

customer.addColumn("Forename");

customer.addColumn("Surname");

customer.addColumn("Street");

customer.addColumn("Town");

customer.addColumn("County");

customer.addColumn("DateOfBirth");

customer.addColumn("Email");

// loop through the ResultSet and add rows to the TableModel

while (rs.next())

{

customer.addRow(new Object[] {

rs.getInt("CustomerID"),

rs.getString("Forename"),

rs.getString("Surname"),

rs.getString("Street"),

rs.getString("Town"),

rs.getString("County"),

rs.getString("DateOfBirth"),

rs.getString("Email")

});

}

// set the TableModel on the JTable

table.setModel(customer);

}

This code gains access into database to insert data into customer table.

pstat = connection.prepareStatement("INSERT INTO customer (Forename, Surname, Street, Town, County, DateOfBirth, Email)VALUES (?,?,?,?,?,?,?)");

pstat.setString (1, Forename);

pstat.setString (2, Surname);

pstat.setString (3, Street);

pstat.setString (4, Town);

pstat.setString (5, countyString);

pstat.setDate (6, DateOfBirth);

pstat.setString (7, Email);

This code allows you travel between pages as for this snippet of code when the back button it clicked it goes to customer sub menu and closes the add customer screen.

BackScreenButton = new JButton("Back");

BackScreenButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

customer\_menu custSubmenu = new customer\_menu();

custSubmenu.*main*(null);

AddCustomer.dispose();

}

});

This code puts the Customer ID, Forename into a JComboBox to create a dropdown to select from and when you select a customer it fills in the dedicated fields such as Customer ID text field and Forename text field and makes them so they are not editable.

final String DATABASE\_URL = "jdbc:mysql://localhost/project";

Connection connection = null ;

PreparedStatement pstat = null;

try {

connection = DriverManager.*getConnection*(DATABASE\_URL, "root", "");

pstat = connection.prepareStatement("SELECT CustomerID, Forename FROM customer");

ResultSet resultSet = pstat.executeQuery();

while (resultSet.next())

{

String CustomerID = resultSet.getString("CustomerID");

String Forename = resultSet.getString("Forename");

// Add the CustomerID, Forename to the JComboBox

CustomerDropdown.addItem(CustomerID + " - " + Forename);

}

pstat . close () ;

connection. close () ;

}

catch (SQLException ex) {

ex.printStackTrace();

}

CustomerDropdown.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

// Get the selected item from the JComboBox

String selectedCustomer = (String) CustomerDropdown.getSelectedItem();

// Extract the CustomerID and Forename from the selected item

String[] parts = selectedCustomer.split(" - ");

String CustomerID = parts[0];

String Forename = parts[1];

// Set the CustomerIDtextfield and Nametextfield to the extracted values

CustomerIDtextfield.setText(CustomerID);

Nametextfield.setText(Forename);

// Set the fields to be uneditable

CustomerIDtextfield.setEditable(false);

Nametextfield.setEditable(false);

}

});

# **Tests**

|  |  |  |  |
| --- | --- | --- | --- |
| Test No. | Scenario | Result | Pass/Fail |
| T1 | Application started | Connection to database made. Display home screen | Pass |
| T2 | User Clicks “Customer” | Display Customer Sub menu | Pass |
| T3 | User Clicks “Product” | Display Product Submenu | Pass |
| T4 | User Clicks “Invoice” | Display Invoice Submenu | Pass |
| T5 | User clicks “Add Customer” on Customer Submenu | Display Add Customer Screen | Pass |
| T5.1 | User adds customer details to be inserted into database | Pop up alert user entry was created after confirmation | Pass |
| T5.2 | User leaves text fields empty | Pop up alerting the text fields are empty and nothing has been added | Pass |
| T5.3 | User inputs invalid date of birth | Pop up alerting user input is invalid and throw exception in console | Pass |
| T5.4 | User clicks Clear | Clears all text fields | Pass |
| T6 | User clicks “Amend/View Customer” on Customer Submenu | Display Amend/View Customer Screen | Pass |
| T6.1 | After opening Amend/View Customer screen | Display all customers in a JTable from customer table in database | Pass |
| T6.2 | User clicks update | Pop up alerting user that no row selected to be updated from table | Pass |
| T6.3 | Edit the row | Pop up after clicking update confirmation do you want to update customer | Pass |
| T7 | User Clicks “Delete Customer” on Customer Submenu | Display Delete  Customer Screen | Pass |
| T7.1 | User clicks delete  button | Pop alerting user to select a row from the JTable to be deleted | Pass |
| T7.2 | User selects row to be deleted | Pop up confirmation asking if they wish to continue with the deletion | Pass |
| T7.3 | User clicks refresh button | Refresh JTable after deleting | Pass |
| T8 | User clicks “Add Product” on Product Submenu | Display Add Product Screen | Pass |
| T8.1 | User adds product details to be inserted into database | Pop up alert user entry was created after confirmation | Pass |
| T8.2 | User leaves text fields empty | Pop up alerting the text fields are empty and nothing has been added | Pass |
| T8.3 | User inputs invalid Price and Stock | Pop up alerting user input is invalid and throw exception in console | Pass |
| T8.4 | User clicks Clear | Clears all text fields | Pass |
| T9 | User clicks “Amend/View Product” on Product Submenu | Display Amend/View Product Screen | Pass |
| T9.1 | After opening Amend/View Product screen | Display all products in a JTable from product table in database | Pass |
| T9.2 | User clicks update | Pop up alerting user that no row selected to be updated from table | Pass |
| T9.3 | Edit the row | Pop up after clicking update button confirmation do you want to update product | Pass |
| T10 | User Clicks “Delete Product” on Product Submenu | Display Delete  Product Screen | Pass |
| T10.1 | User clicks delete  button | Pop alerting user to select a row from the JTable to be deleted | Pass |
| T10.2 | User selects row to be deleted | Pop up confirmation asking if they wish to continue with the deletion | Pass |
| T10.3 | User clicks refresh button | Refresh JTable after deleting | Pass |
| T11 | User clicks “Add Invoice” on Invoice Submenu | Display Add Invoice Screen | Pass |
| T11.1 | User select customer from Customer JComboBox and then adds rest of invoice details to be inserted into database | Pop up alert user entry was created after confirmation | Pass |
| T11.2 | User leaves text fields empty | Pop up alerting the text fields are empty and nothing has been added | Pass |
| T11.3 | User inputs invalid Total owed and Date | Pop up alerting user input is invalid and throw exception in console | Pass |
| T11.4 | User clicks Clear | Clears all text fields | Pass |
| T12 | User clicks “Amend/View Invoice” on Invoice Submenu | Display Amend/View Invoice Screen | Pass |
| T12,1 | After opening Amend/View Invoice screen | Display all invoices in a JTable from invoice table in database | Pass |
| T12.2 | User clicks update | Pop up alerting no invoice has been selected to update | Pass |
| T12.3 | Pick the invoice through the invoice JComboBox you wish to edit | Pop up after editing the invoice and clicking update button alerting user of they wish to continue with the update | Pass |
| T12.4 | User clicks Clear | Clears al text fields | Pass |
| T12.5 | User clicks refresh button | Refresh JTable after updating | Pass |
| T13 | User clicks “Delete Invoice” on Invoice Submenu | Display Delete  Invoice Screen | Pass |
| T13.1 | User clicks delete  button | Pop alerting user to select a row from the JTable to be deleted | Pass |
| T13.2 | User selects row to be deleted | Pop up confirmation asking if they wish to continue with the deletion | Pass |
| T13.3 | User clicks refresh button | Refresh JTable after deleting | Pass |
| T14 | User clicks “Add Product to Invoice” on Invoice Submenu | Display Add Product to Invoice Screen | Pass |
| T14.1 | User select Invoice and Product from JComboBox and then adds rest of details to be inserted into database | Pop up alert user entry was created after confirmation | Pass |
| T14.2 | User leaves text fields empty | Pop up alerting the text fields are empty and nothing has been added | Pass |
| T14.3 | User inputs invalid Quantity | Pop up alerting user input is invalid and throw exception in console | Pass |
| T14.4 | User clicks Clear | Clears all text fields | Pass |
| T15 | User clicks “View Products on Invoice” on Invoice Submenu | Display View Products on Invoice Screen | Pass |
| T15.1 | After opening View Products on Invoice Screen | Display all products on invoices in a JTable from cart table in database | Pass |
| T16 | User clicks “Delete Products on Invoice” Submenu | Display Delete Products on Invoice | Pass |
| T16.1 | User clicks delete  button | Pop alerting user to select a row from the JTable to be deleted | Pass |
| T16.2 | User selects row to be deleted | Pop up confirmation asking if they wish to continue with the deletion | Pass |
| T16.3 | User clicks refresh button | Refresh JTable after deleting | `Pass |