7/4/2022

Customer Invoice Management System

(Student Name) - Daniel Giedraitis
(Student ID) – C00260331

Table of Contents

Description

This project based on an invoice management system that has been developed using java and MySQL. It tracks all the information on customers, products, and invoices in the system. I have developed all types of CRUD (Create, Read, Update and Delete) operations an administrator can do to the customer, product and invoice database tables using GUI’s.

An administrator will be able to add new customers to the database. See a list of all the customers details in the system. Admin will also be able to edit, update and delete records of the customers in the system.

An administrator will be able to create new invoices in the database. View a list of all invoice details stored in the system. In addition, administrators will be able to edit, update, and delete invoices in the system.

New products can be added to the database by an administrator. View a list of all the product details in the system as well as the product stock. Admin will also be able to edit, update, and delete customer records in the system.

This system has error handling, so if a user wants to sign up for the system but forgets to enter some required details, the system will prompt the user to fill in that field before they can sign up.

Requirements

For this project I created a system to manage customer purchases. The system is made up of the following components:

There is a backend database for this system with the following 4 tables, administrator table, customer table, product table and invoice table.

The customer table and product table are joint into invoice table because the customer ID and Product ID is needed in the invoice table.

I used MySQL Workbench for the database and MySQL Installer for the MySQL server.

This system has CRUD (create, retrieve, update, delete) operations on the database. Administrator can CRUD the customer, product, and invoice tables in the system.

This system provides error handling, so if the user wants to sign up into the system and forgets to enter some details that are required, it will prompt the user to fill in that field before they can sign up. Also, when the user is signing in into the system, the user must use the correct email and password to login or else it will prompt the user that the email or password is wrong.

This system demonstrates the use of several Swing components such as (JFrame, JPanel, JTextField, JLabel, JOptionPane, SwingConstants, ButtonGroup, JComboBox, JRadioButton, JScrollBar, UIManager and JPasswordField).

Functionality

Login GUI

- Use Email address and password to login into system.
- Click the sign-up button if you do not have an account to login in with.
- If the email and password entered does not match with the email and password in the database, a popup will appear saying “incorrect email or password”.

Sign up GUI

- User provides their details into the text fields and creates an account into the database so they can sign in.
- There is a button for the user to clear all the details entered in the text box if they user wants to re-write their details again.
- If the user created the account, the user could click the sign in button to log into the system.
- If the user clicks the sign-up button and text box fields are empty, it will prompt the user saying for example “first name is required” or “password is required” and so on until all the fields are completed…

Customer GUI

- Once the administrator logs in, they will be brought onto the Customer GUI.
- On this GUI the administrator will have CRUD operations to perform on the customer’s table.
- The administrator will be able to enter details into the text fields and press the create button to create a new customer into the database. They will also be able to select a row in the customer table in the GUI and it will show the customer details in the text box fields. The administrator will be able to update the fields by changing the details and clicking the update button to update the customer details in the database. The administrator can also select a customer from the table and click the delete customer button to remove a customer from the database.
- At the top right of the GUI the administrator can click the product or invoice button to go to the product or invoice GUI.

Product GUI

- The administrator will have CRUD operations to perform on the product table.
- The administrator can enter information into the text fields and press the create button to add a new product to the database. They will also be able to select a row in the product table in the GUI, and the product details will be displayed in the text box fields. The administrator will be able to update the fields in the database by changing the details and clicking the update button. The administrator can also remove a product from the database by selecting a product from the table and clicking the delete product button.
- At the top right of the GUI the administrator can click the invoice or customer button to go to the invoice or customer GUI.

Invoice GUI

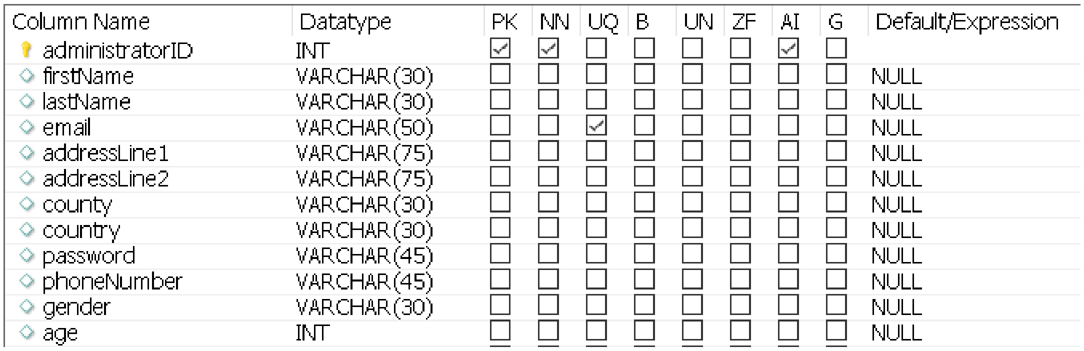
- The administrator will have CRUD operations to perform on the invoice table.
- The administrator will be able to enter details into the text fields and press the create button to create a new invoice into the database. They will also be able to select a row in the invoice table in the GUI and it will show the invoice details in the text box fields. The administrator will be able to update the fields by changing the details and clicking the update button to update the invoice details in the database. The administrator can also select an invoice from the table and click the delete invoice button to remove an invoice from the database.
- At the top right of the GUI the administrator can click the product or customer button to go to the product or customer GUI.

Database Tables

Administrator Table

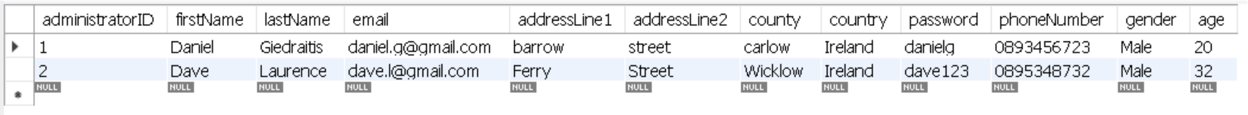
Structure:

administratorID has a primary key and auto increment on it. Email has a unique key.



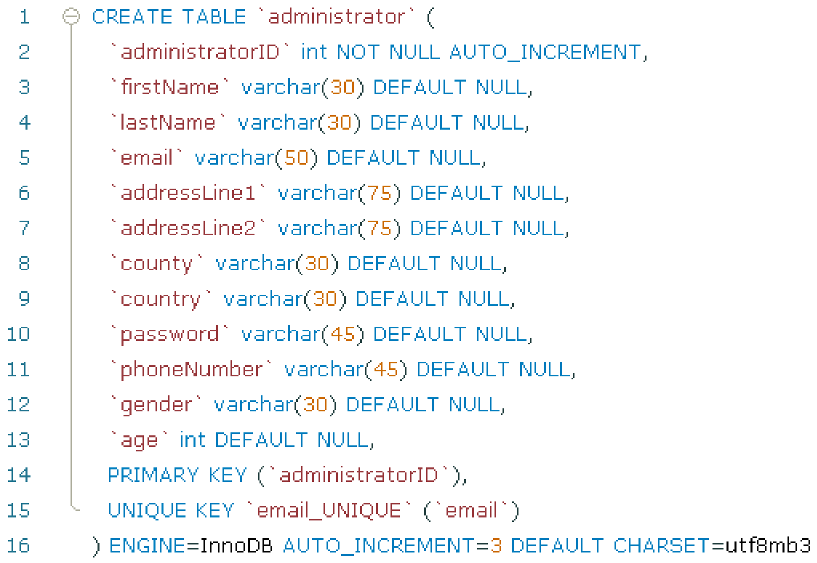
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
administratorID	INT	☑	☑	☐	☐	☐	☐	☑	☐	
firstName	VARCHAR(30)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
lastName	VARCHAR(30)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
email	VARCHAR(50)	☐	☐	☑	☐	☐	☐	☐	☐	NULL
addressLine1	VARCHAR(75)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
addressLine2	VARCHAR(75)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
county	VARCHAR(30)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
country	VARCHAR(30)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
password	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
phoneNumber	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
gender	VARCHAR(30)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
age	INT	☐	☐	☐	☐	☐	☐	☐	☐	NULL

Data:



administratorID	firstName	lastName	email	addressLine1	addressLine2	county	country	password	phoneNumber	gender	age
1	Daniel	Giedraitis	daniel.g@gmail.com	barrow	street	carlow	Ireland	danielg	0893456723	Male	20
2	Dave	Laurence	dave.l@gmail.com	Ferry	Street	Wicklow	Ireland	dave123	0895348732	Male	32
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

DDL for Administrator Table:

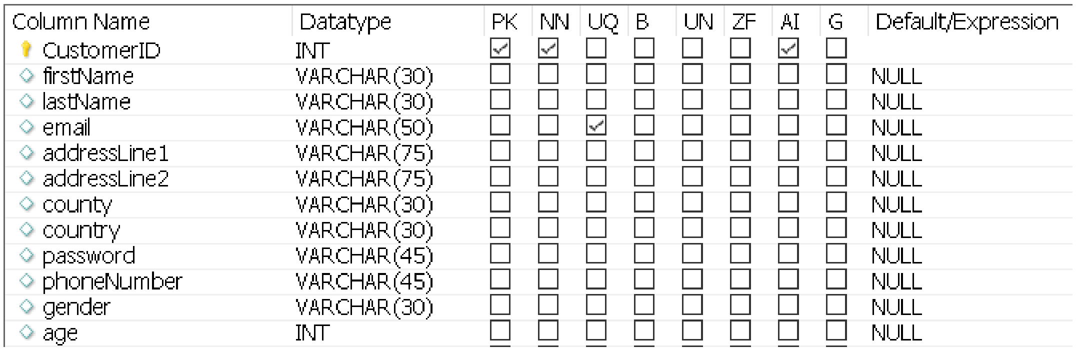


```
CREATE TABLE `administrator` (
  `administratorID` int NOT NULL AUTO_INCREMENT,
  `firstName` varchar(30) DEFAULT NULL,
  `lastName` varchar(30) DEFAULT NULL,
  `email` varchar(50) DEFAULT NULL,
  `addressLine1` varchar(75) DEFAULT NULL,
  `addressLine2` varchar(75) DEFAULT NULL,
  `county` varchar(30) DEFAULT NULL,
  `country` varchar(30) DEFAULT NULL,
  `password` varchar(45) DEFAULT NULL,
  `phoneNumber` varchar(45) DEFAULT NULL,
  `gender` varchar(30) DEFAULT NULL,
  `age` int DEFAULT NULL,
  PRIMARY KEY (`administratorID`),
  UNIQUE KEY `email_UNIQUE` (`email`)
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=utf8mb3
```

Customer Table

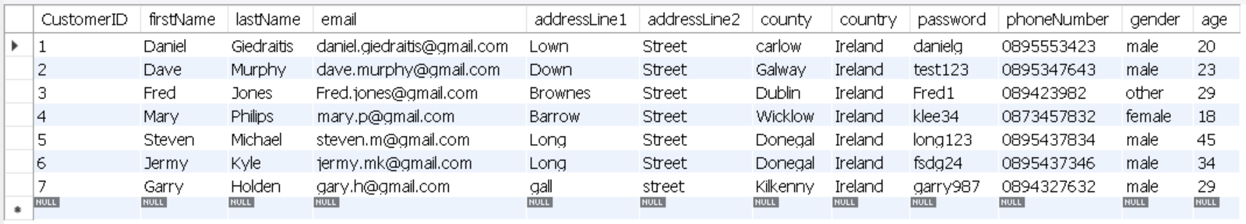
Structure:

customerID has a primary key and auto increment on it. Email has a unique key.



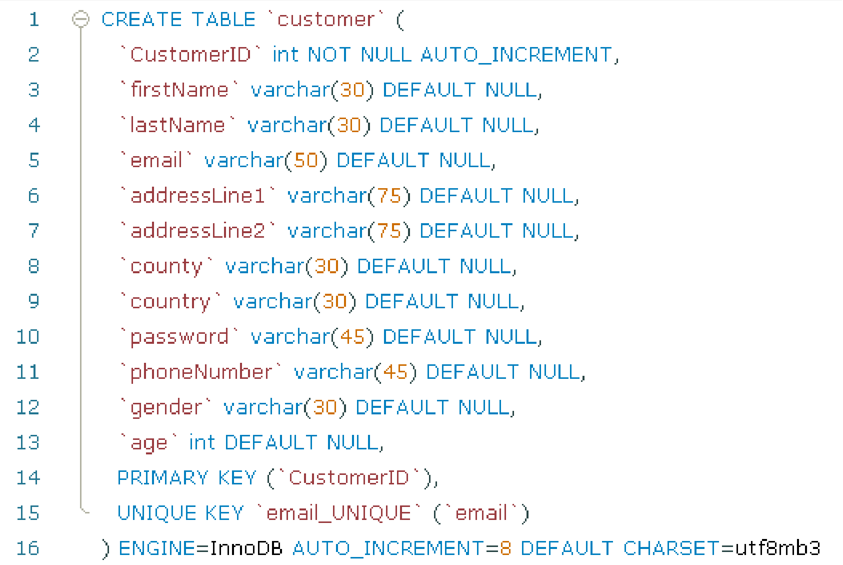
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
CustomerID	INT	☑	☑	□	□	□	□	☑	□	
firstName	VARCHAR(30)	□	□	□	□	□	□	□	□	NULL
lastName	VARCHAR(30)	□	□	□	□	□	□	□	□	NULL
email	VARCHAR(50)	□	□	☑	□	□	□	□	□	NULL
addressLine1	VARCHAR(75)	□	□	□	□	□	□	□	□	NULL
addressLine2	VARCHAR(75)	□	□	□	□	□	□	□	□	NULL
county	VARCHAR(30)	□	□	□	□	□	□	□	□	NULL
country	VARCHAR(30)	□	□	□	□	□	□	□	□	NULL
password	VARCHAR(45)	□	□	□	□	□	□	□	□	NULL
phoneNumber	VARCHAR(45)	□	□	□	□	□	□	□	□	NULL
gender	VARCHAR(30)	□	□	□	□	□	□	□	□	NULL
age	INT	□	□	□	□	□	□	□	□	NULL

Data:



CustomerID	firstName	lastName	email	addressLine1	addressLine2	county	country	password	phoneNumber	gender	age
1	Daniel	Giedraitis	daniel.giedraitis@gmail.com	Lown	Street	carlow	Ireland	danielg	0895553423	male	20
2	Dave	Murphy	dave.murphy@gmail.com	Down	Street	Galway	Ireland	test123	0895347643	male	23
3	Fred	Jones	Fred.jones@gmail.com	Brownes	Street	Dublin	Ireland	Fred1	089423982	other	29
4	Mary	Philips	mary.p@gmail.com	Barrow	Street	Wicklow	Ireland	klee34	0873457832	female	18
5	Steven	Michael	steven.m@gmail.com	Long	Street	Donegal	Ireland	long123	0895437834	male	45
6	Jermy	Kyle	jermy.mk@gmail.com	Long	Street	Donegal	Ireland	fsdg24	0895437346	male	34
7	Garry	Holden	gary.h@gmail.com	gall	street	Kilkenny	Ireland	garry987	0894327632	male	29
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

DDL for Customer Table:

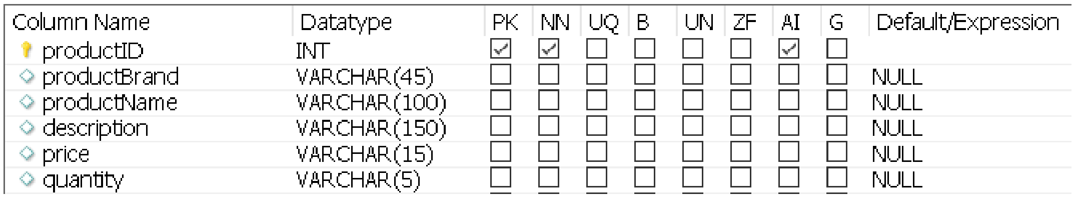


```sql
CREATE TABLE `customer` (
  `CustomerID` int NOT NULL AUTO_INCREMENT,
  `firstName` varchar(30) DEFAULT NULL,
  `lastName` varchar(30) DEFAULT NULL,
  `email` varchar(50) DEFAULT NULL,
  `addressLine1` varchar(75) DEFAULT NULL,
  `addressLine2` varchar(75) DEFAULT NULL,
  `county` varchar(30) DEFAULT NULL,
  `country` varchar(30) DEFAULT NULL,
  `password` varchar(45) DEFAULT NULL,
  `phoneNumber` varchar(45) DEFAULT NULL,
  `gender` varchar(30) DEFAULT NULL,
  `age` int DEFAULT NULL,
 PRIMARY KEY (`CustomerID`),
 UNIQUE KEY `email_UNIQUE` (`email`)
) ENGINE=InnoDB AUTO_INCREMENT=8 DEFAULT CHARSET=utf8mb3
```

Product Table

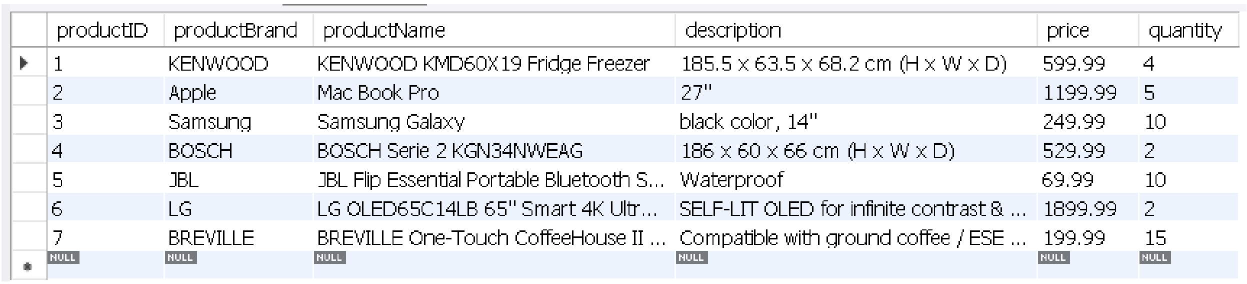
Structure:

productID has a primary key and auto increment on it.



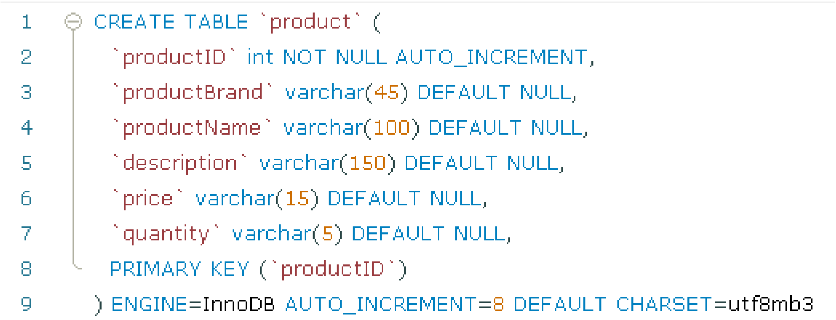
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
productID	INT	☑	☑	□	□	□	□	☑	□	
productBrand	VARCHAR(45)	□	□	□	□	□	□	□	□	NULL
productName	VARCHAR(100)	□	□	□	□	□	□	□	□	NULL
description	VARCHAR(150)	□	□	□	□	□	□	□	□	NULL
price	VARCHAR(15)	□	□	□	□	□	□	□	□	NULL
quantity	VARCHAR(5)	□	□	□	□	□	□	□	□	NULL

Data:



	productID	productBrand	productName	description	price	quantity
▶	1	KENWOOD	KENWOOD KMD60X19 Fridge Freezer	185.5 x 63.5 x 68.2 cm (H x W x D)	599.99	4
	2	Apple	Mac Book Pro	27"	1199.99	5
	3	Samsung	Samsung Galaxy	black color, 14"	249.99	10
	4	BOSCH	BOSCH Serie 2 KGN34NWEAG	186 x 60 x 66 cm (H x W x D)	529.99	2
	5	JBL	JBL Flip Essential Portable Bluetooth S...	Waterproof	69.99	10
	6	LG	LG OLED65C14LB 65" Smart 4K Ultr...	SELF-LIT OLED for infinite contrast & ...	1899.99	2
	7	BREVILLE	BREVILLE One-Touch CoffeeHouse II ...	Compatible with ground coffee / ESE ...	199.99	15
*	NULL	NULL	NULL	NULL	NULL	NULL

DDL for Product Table:

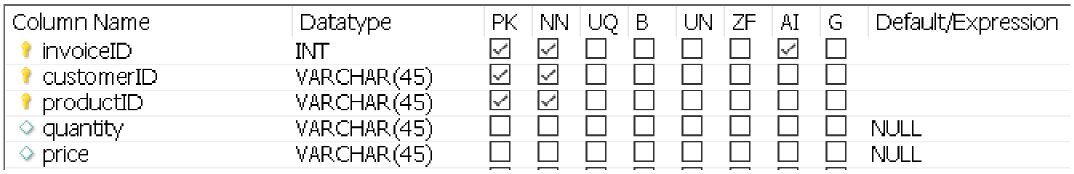


```
CREATE TABLE `product` (
  `productID` int NOT NULL AUTO_INCREMENT,
  `productBrand` varchar(45) DEFAULT NULL,
  `productName` varchar(100) DEFAULT NULL,
  `description` varchar(150) DEFAULT NULL,
  `price` varchar(15) DEFAULT NULL,
  `quantity` varchar(5) DEFAULT NULL,
  PRIMARY KEY (`productID`)
) ENGINE=InnoDB AUTO_INCREMENT=8 DEFAULT CHARSET=utf8mb3
```

Invoice Table

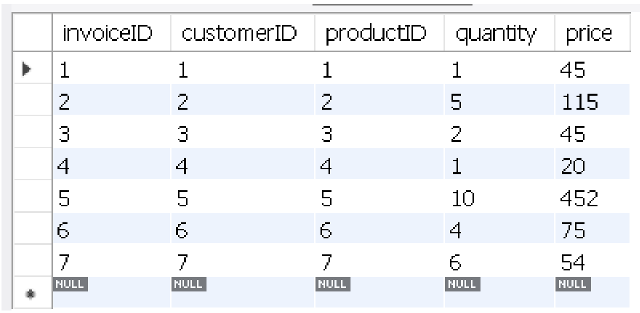
Structure:

invoiceID, customerID and productID has a primary key. invoiceID has auto increment on it.



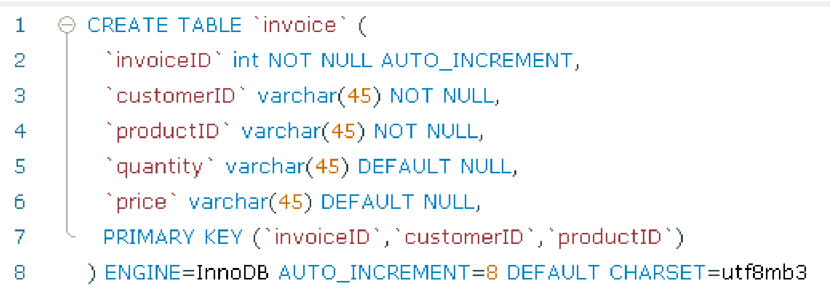
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
invoiceID	INT	☑	☑	□	□	□	□	☑	□	
customerID	VARCHAR(45)	☑	☑	□	□	□	□	□	□	
productID	VARCHAR(45)	☑	☑	□	□	□	□	□	□	
quantity	VARCHAR(45)	□	□	□	□	□	□	□	□	NULL
price	VARCHAR(45)	□	□	□	□	□	□	□	□	NULL

Data:



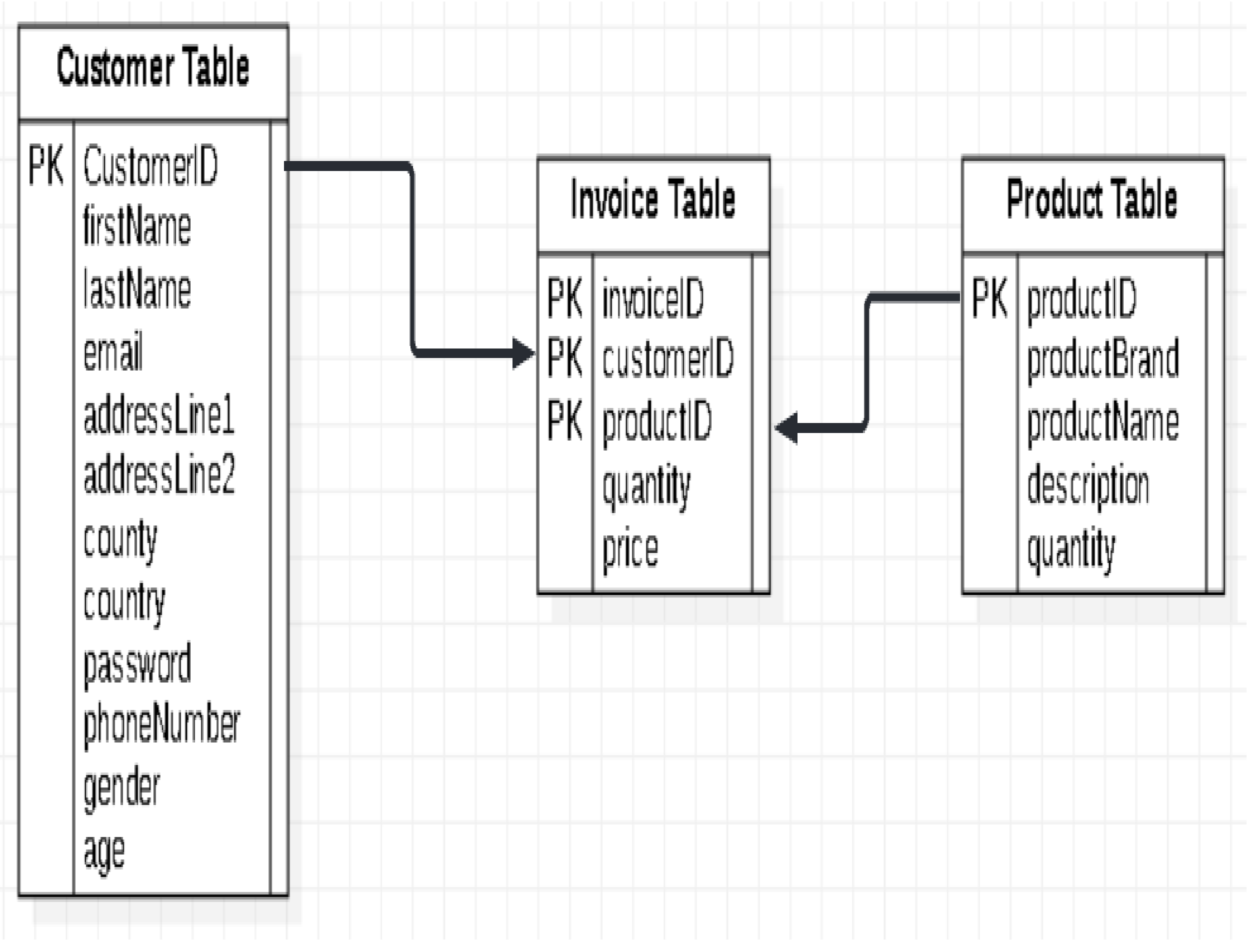
	invoiceID	customerID	productID	quantity	price
▶	1	1	1	1	45
	2	2	2	5	115
	3	3	3	2	45
	4	4	4	1	20
	5	5	5	10	452
	6	6	6	4	75
	7	7	7	6	54
✱	NULL	NULL	NULL	NULL	NULL

DDL for Invoice Table:



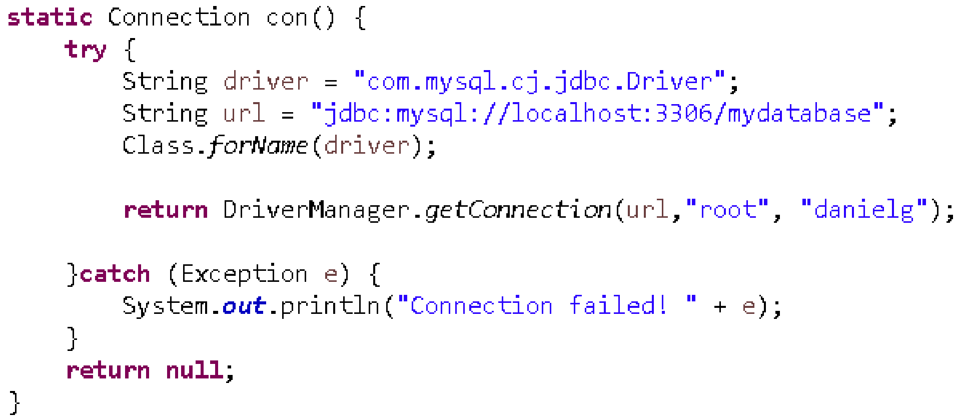
```
CREATE TABLE `invoice` (
  `invoiceID` int NOT NULL AUTO_INCREMENT,
  `customerID` varchar(45) NOT NULL,
  `productID` varchar(45) NOT NULL,
  `quantity` varchar(45) DEFAULT NULL,
  `price` varchar(45) DEFAULT NULL,
  PRIMARY KEY (`invoiceID`,`customerID`,`productID`)
) ENGINE=InnoDB AUTO_INCREMENT=8 DEFAULT CHARSET=utf8mb3
```

Entity Relationship Diagram



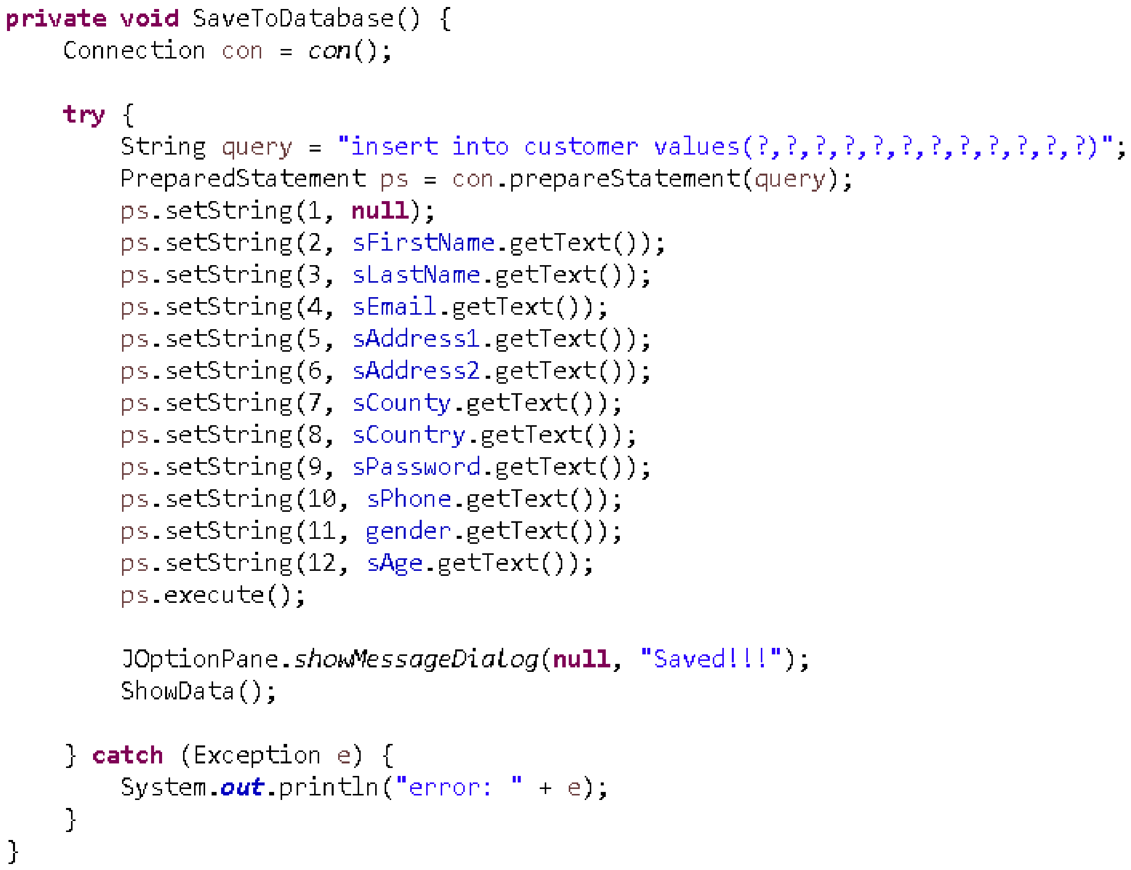
Interesting source code snippets

This method is used to connect to the database in workbench:



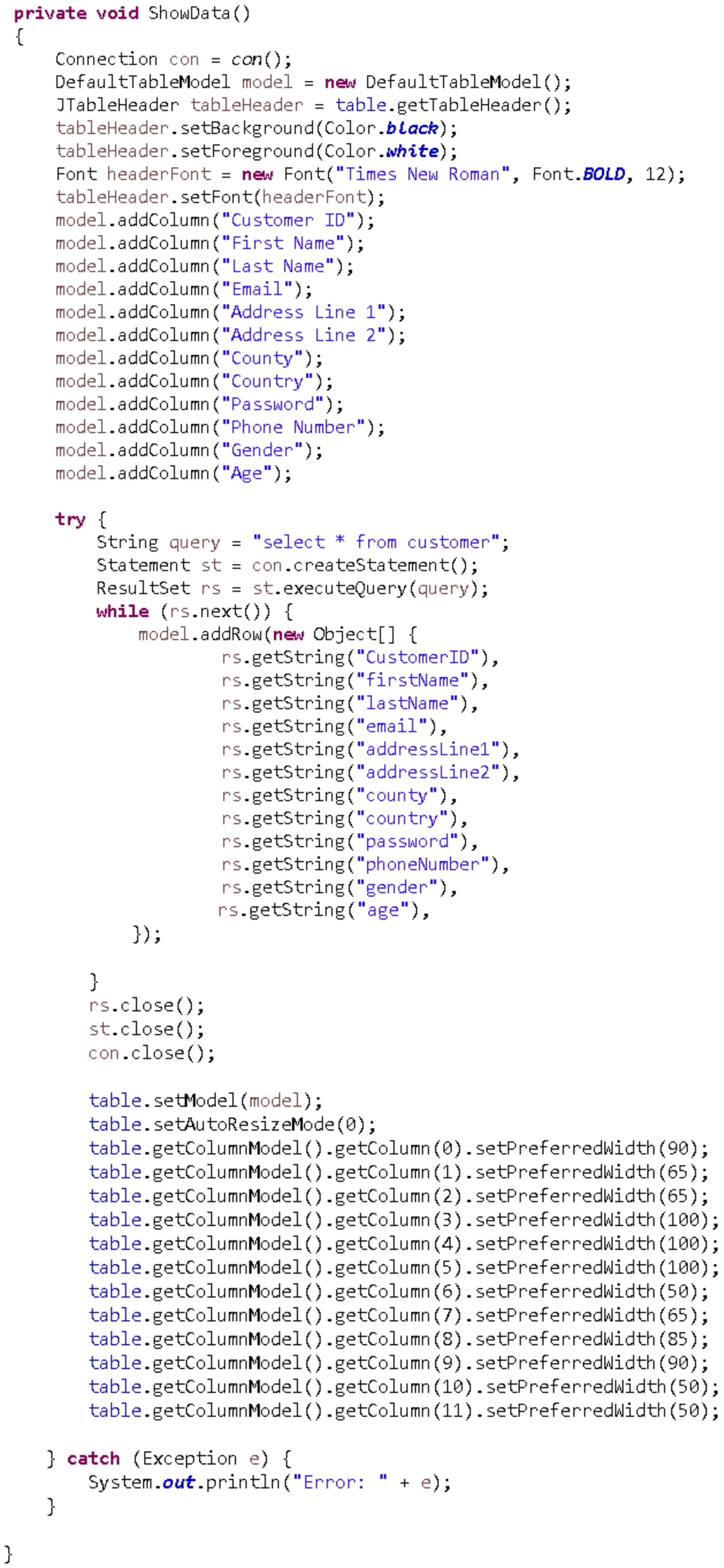
```
static Connection con() {
    try {
        String driver = "com.mysql.cj.jdbc.Driver";
        String url = "jdbc:mysql://localhost:3306/mydatabase";
        Class.forName(driver);

        return DriverManager.getConnection(url,"root", "danielg");

    }catch (Exception e) {
        System.out.println("Connection failed! " + e);
    }
    return null;
}
```

SaveToDatabase method is used to insert customer details such as first name, last name, email etc… into the database.



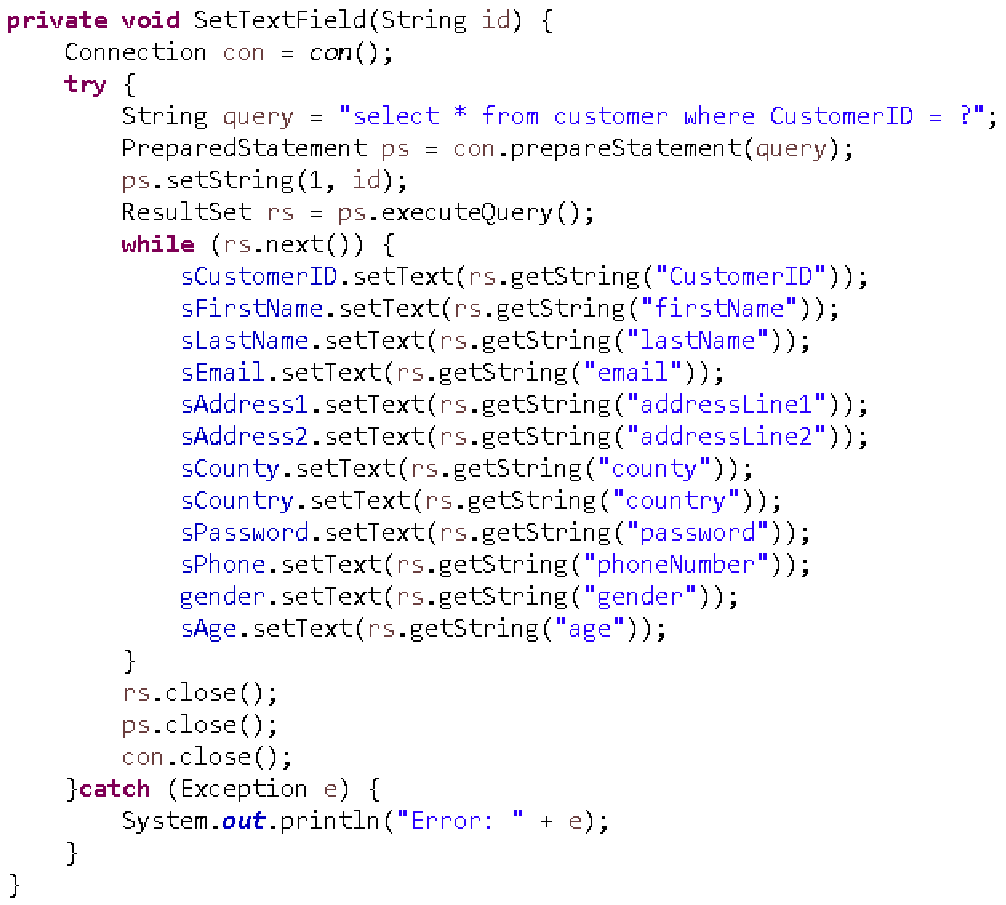
```
private void SaveToDatabase() {
    Connection con = con();

    try {
        String query = "insert into customer values(?,?,?,?,?,?,?,?,?,?,?,?)";
        PreparedStatement ps = con.prepareStatement(query);
        ps.setString(1, null);
        ps.setString(2, sFirstName.getText());
        ps.setString(3, sLastName.getText());
        ps.setString(4, sEmail.getText());
        ps.setString(5, sAddress1.getText());
        ps.setString(6, sAddress2.getText());
        ps.setString(7, sCounty.getText());
        ps.setString(8, sCountry.getText());
        ps.setString(9, sPassword.getText());
        ps.setString(10, sPhone.getText());
        ps.setString(11, gender.getText());
        ps.setString(12, sAge.getText());
        ps.execute();

        JOptionPane.showMessageDialog(null, "Saved!!!");
        ShowData();

    } catch (Exception e) {
        System.out.println("error: " + e);
    }
}
```

ShowData method is used to show all the customer details in the database into the table of the GUI so that the administrator can use CRUD operations on them.



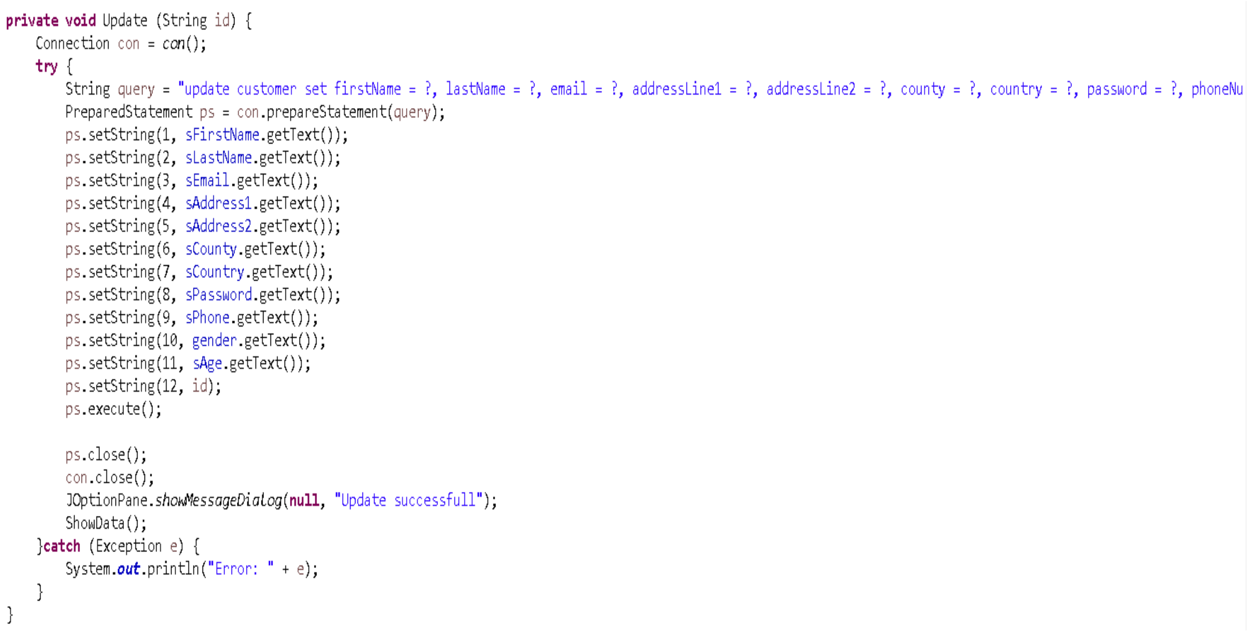
```
 private void ShowData()
 {
     Connection con = con();
     DefaultTableModel model = new DefaultTableModel();
     JTableHeader tableHeader = table.getTableHeader();
     tableHeader.setBackground(Color.black);
     tableHeader.setForeground(Color.white);
     Font headerFont = new Font("Times New Roman", Font.BOLD, 12);
     tableHeader.setFont(headerFont);
     model.addColumn("Customer ID");
     model.addColumn("First Name");
     model.addColumn("Last Name");
     model.addColumn("Email");
     model.addColumn("Address Line 1");
     model.addColumn("Address Line 2");
     model.addColumn("County");
     model.addColumn("Country");
     model.addColumn("Password");
     model.addColumn("Phone Number");
     model.addColumn("Gender");
     model.addColumn("Age");

     try {
         String query = "select * from customer";
         Statement st = con.createStatement();
         ResultSet rs = st.executeQuery(query);
         while (rs.next()) {
             model.addRow(new Object[] {
                     rs.getString("CustomerID"),
                     rs.getString("firstName"),
                     rs.getString("lastName"),
                     rs.getString("email"),
                     rs.getString("addressLine1"),
                     rs.getString("addressLine2"),
                     rs.getString("county"),
                     rs.getString("country"),
                     rs.getString("password"),
                     rs.getString("phoneNumber"),
                     rs.getString("gender"),
                    rs.getString("age"),
            });

        }
        rs.close();
        st.close();
        con.close();

        table.setModel(model);
        table.setAutoResizeMode(0);
        table.getColumnModel().getColumn(0).setPreferredWidth(90);
        table.getColumnModel().getColumn(1).setPreferredWidth(65);
        table.getColumnModel().getColumn(2).setPreferredWidth(65);
        table.getColumnModel().getColumn(3).setPreferredWidth(100);
        table.getColumnModel().getColumn(4).setPreferredWidth(100);
        table.getColumnModel().getColumn(5).setPreferredWidth(100);
        table.getColumnModel().getColumn(6).setPreferredWidth(50);
        table.getColumnModel().getColumn(7).setPreferredWidth(65);
        table.getColumnModel().getColumn(8).setPreferredWidth(85);
        table.getColumnModel().getColumn(9).setPreferredWidth(90);
        table.getColumnModel().getColumn(10).setPreferredWidth(50);
        table.getColumnModel().getColumn(11).setPreferredWidth(50);

    } catch (Exception e) {
        System.out.println("Error: " + e);
    }

}
```

SetTextField method is used to populate the text fields with customer details when ever the user selects a row in the table.



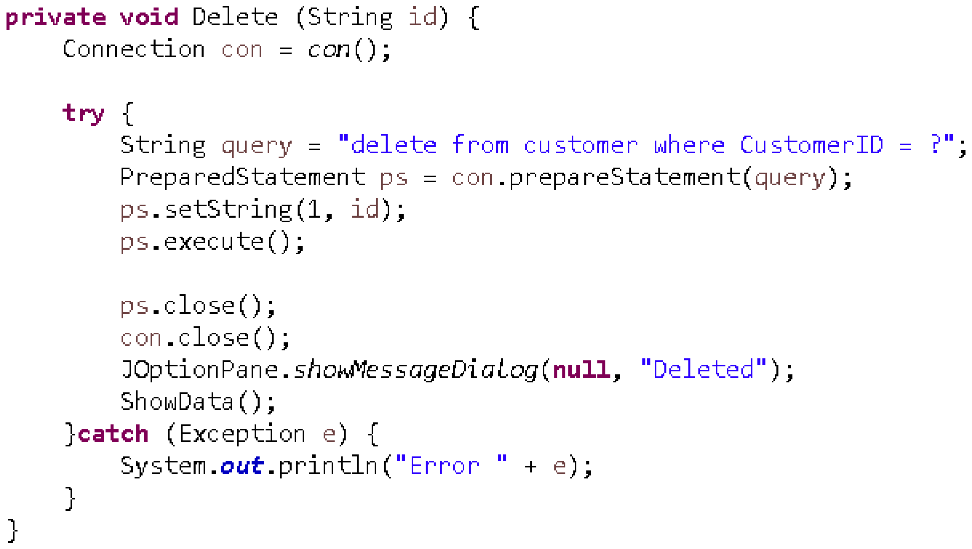
```
private void SetTextField(String id) {
    Connection con = con();
    try {
        String query = "select * from customer where CustomerID = ?";
        PreparedStatement ps = con.prepareStatement(query);
        ps.setString(1, id);
        ResultSet rs = ps.executeQuery();
        while (rs.next()) {
            sCustomerID.setText(rs.getString("CustomerID"));
            sFirstName.setText(rs.getString("firstName"));
            sLastName.setText(rs.getString("lastName"));
            sEmail.setText(rs.getString("email"));
            sAddress1.setText(rs.getString("addressLine1"));
            sAddress2.setText(rs.getString("addressLine2"));
            sCounty.setText(rs.getString("county"));
            sCountry.setText(rs.getString("country"));
            sPassword.setText(rs.getString("password"));
            sPhone.setText(rs.getString("phoneNumber"));
            gender.setText(rs.getString("gender"));
            sAge.setText(rs.getString("age"));
        }
        rs.close();
        ps.close();
        con.close();
    }catch (Exception e) {
        System.out.println("Error: " + e);
    }
}
```

The Update method is used when the administer decides to edit the customer details and update the details into the database.



```
private void Update (String id) {
    Connection con = con();
    try {
        String query = "update customer set firstName = ?, lastName = ?, email = ?, addressLine1 = ?, addressLine2 = ?, county = ?, country = ?, password = ?, phoneNu
        PreparedStatement ps = con.prepareStatement(query);
        ps.setString(1, sFirstName.getText());
        ps.setString(2, sLastName.getText());
        ps.setString(3, sEmail.getText());
        ps.setString(4, sAddress1.getText());
        ps.setString(5, sAddress2.getText());
        ps.setString(6, sCounty.getText());
        ps.setString(7, sCountry.getText());
        ps.setString(8, sPassword.getText());
        ps.setString(9, sPhone.getText());
        ps.setString(10, gender.getText());
        ps.setString(11, sAge.getText());
        ps.setString(12, id);
        ps.execute();

        ps.close();
        con.close();
        JOptionPane.showMessageDialog(null, "Update successfull");
        ShowData();
    }catch (Exception e) {
        System.out.println("Error: " + e);
    }
}
```

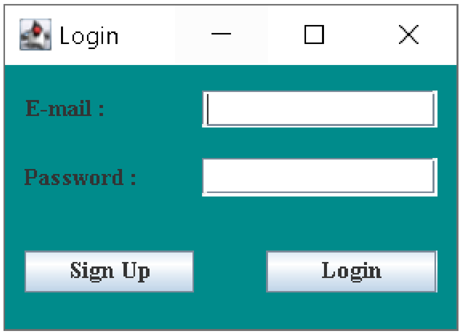
The Delete method is used whenever the administer click on a customer in the table and clicks the delete button to remove the customer from the table and database.



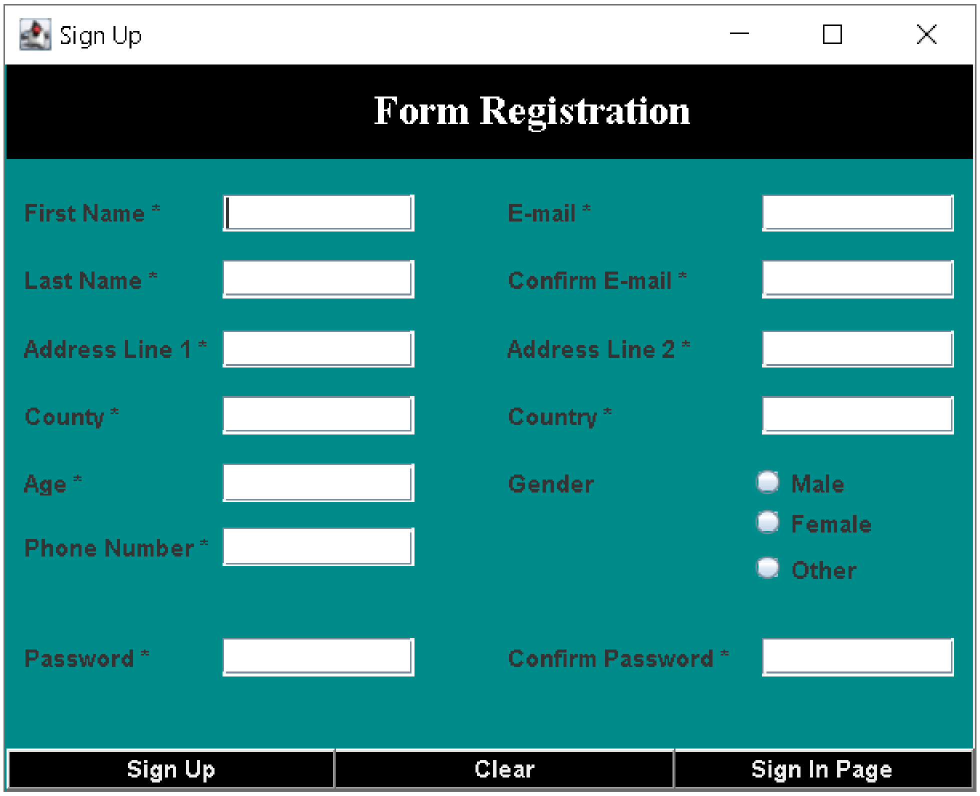
```
private void Delete (String id) {
    Connection con = con();

    try {
        String query = "delete from customer where CustomerID = ?";
        PreparedStatement ps = con.prepareStatement(query);
        ps.setString(1, id);
        ps.execute();

        ps.close();
        con.close();
        JOptionPane.showMessageDialog(null, "Deleted");
        ShowData();
    }catch (Exception e) {
        System.out.println("Error " + e);
    }
}
```

GUI Screens

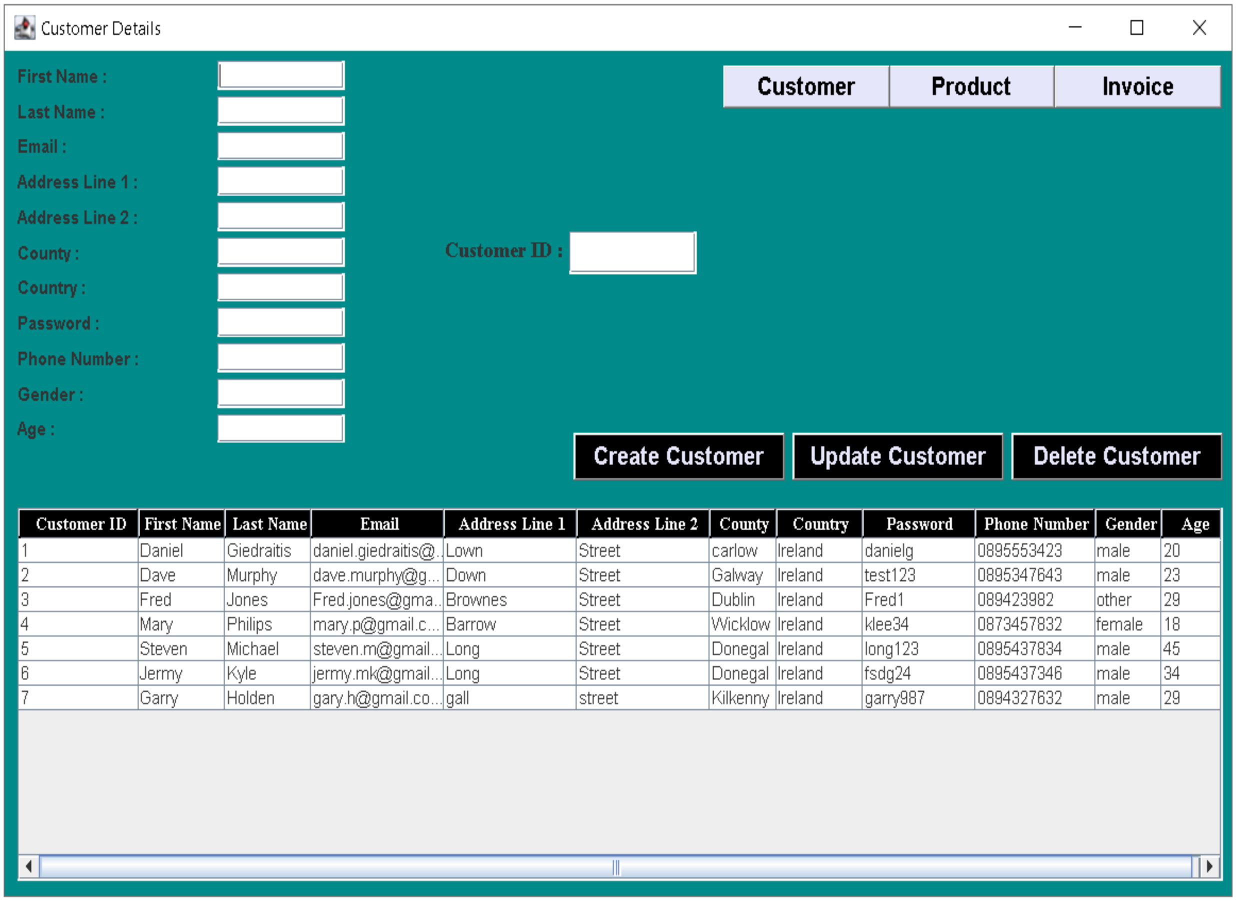
Login GUI:



Sign Up GUI:

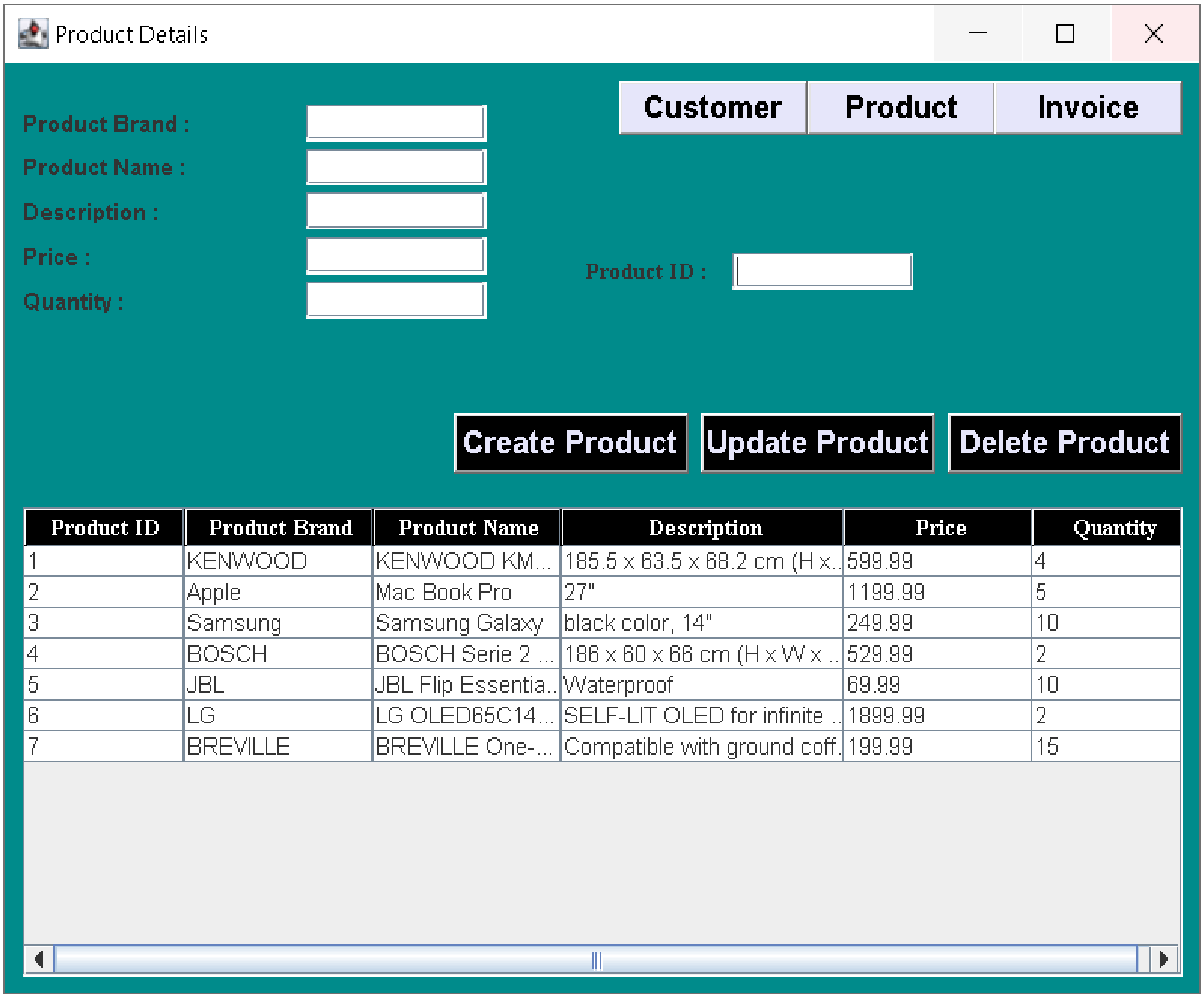


GUI for Administrator to use CRUD operations on Customer details in database:



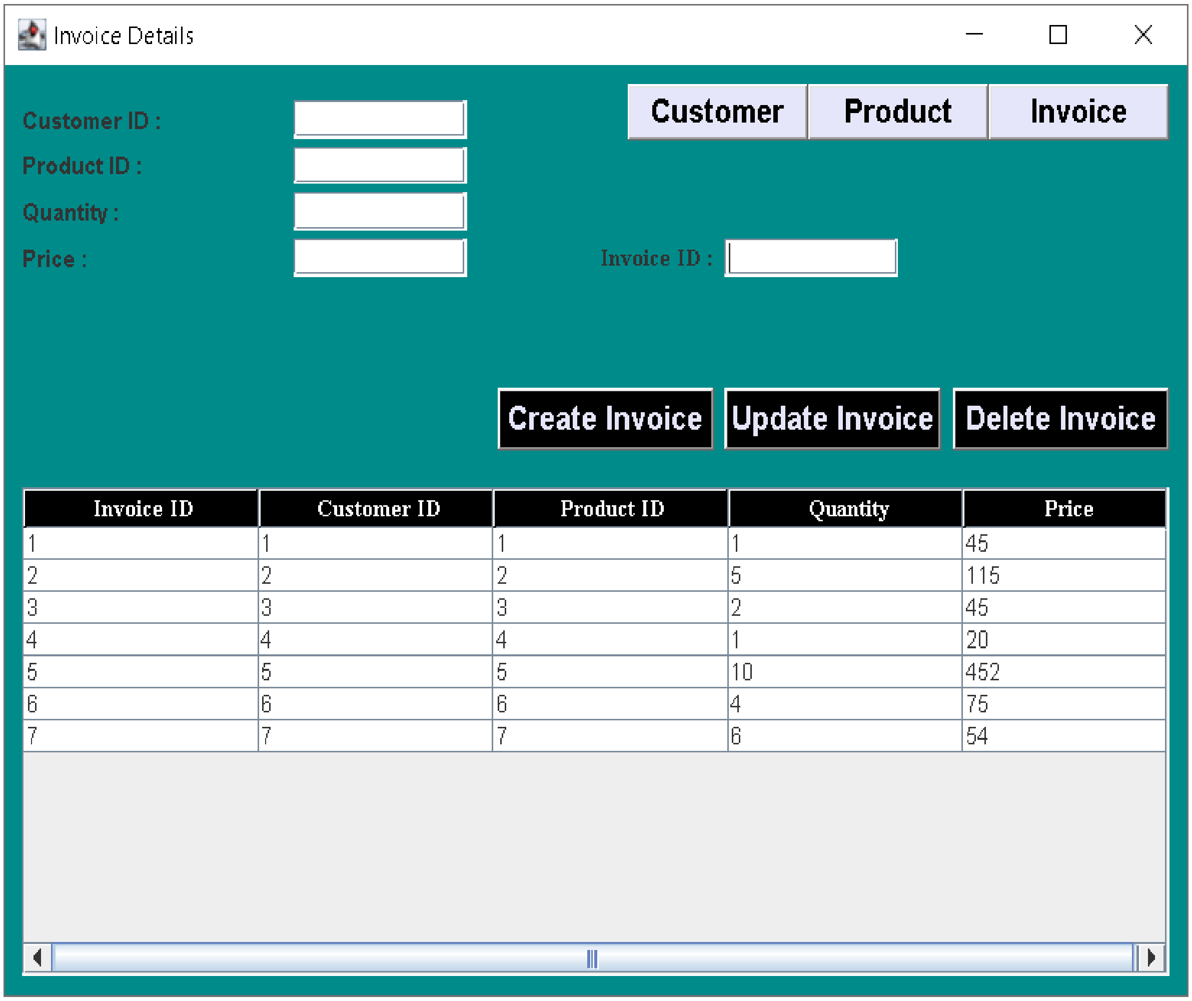
Customer ID	First Name	Last Name	Email	Address Line 1	Address Line 2	County	Country	Password	Phone Number	Gender	Age
1	Daniel	Giedraitis	daniel.giedraitis@...	Lown	Street	carlow	Ireland	danielg	0895553423	male	20
2	Dave	Murphy	dave.murphy@g...	Down	Street	Galway	Ireland	test123	0895347643	male	23
3	Fred	Jones	Fred.jones@gma...	Brownes	Street	Dublin	Ireland	Fred1	089423982	other	29
4	Mary	Philips	mary.p@gmail.c...	Barrow	Street	Wicklow	Ireland	klee34	0873457832	female	18
5	Steven	Michael	steven.m@gmail...	Long	Street	Donegal	Ireland	long123	0895437834	male	45
6	Jermy	Kyle	jermy.mk@gmail...	Long	Street	Donegal	Ireland	fsdg24	0895437346	male	34
7	Garry	Holden	gary.h@gmail.co...	gall	street	Kilkenny	Ireland	garry987	0894327632	male	29

GUI for Administrator to use CRUD operations on Product details in database:



Product ID	Product Brand	Product Name	Description	Price	Quantity
1	KENWOOD	KENWOOD KM...	185.5 x 63.5 x 68.2 cm (H x...	599.99	4
2	Apple	Mac Book Pro	27"	1199.99	5
3	Samsung	Samsung Galaxy	black color, 14"	249.99	10
4	BOSCH	BOSCH Serie 2 ...	186 x 60 x 66 cm (H x W x ...	529.99	2
5	JBL	JBL Flip Essentia...	Waterproof	69.99	10
6	LG	LG OLED65C14...	SELF-LIT OLED for infinite ...	1899.99	2
7	BREVILLE	BREVILLE One-...	Compatible with ground coff...	199.99	15

GUI for Administrator to use CRUD operations on Invoice's in database:



Invoice ID	Customer ID	Product ID	Quantity	Price
1	1	1	1	45
2	2	2	5	115
3	3	3	2	45
4	4	4	1	20
5	5	5	10	452
6	6	6	4	75
7	7	7	6	54

Test Data

Login Screen

Test Data Number	Step	Expected Result	Result	Status
L1	System Started	Connection to database complete and login screen displayed	Connection to database complete and login screen displayed	Pass
L2	User clicks Sign Up button	Sign up GUI is shown	Sign up GUI is shown	Pass
L3	User clicks Sign in button with incorrect details in Email and Password field	User is prompted with message saying “incorrect email or password”	User is prompted with message saying “incorrect email or password”	Pass
L4	User clicks Sign in with correct details in Email and Password field	The main customer GUI is then displayed for the user	The main customer GUI is then displayed for the user	Pass
L5	User clicks the x button at the top right of the GUI	GUI closes	GUI closes	Pass

Sign Up Screen

Test Data Number	Step	Expected Result	Result	Status
L1	User clicks the “Sign in page” button	Sign in GUI is shown	Sign in GUI is shown	Pass
L2	User enters details into the text box fields and clicks the “clear button”	All the text fields and radio buttons get cleared and screen is back to default	All the text fields and radio buttons get cleared and screen is back to default	Pass
L3	User clicks the “Sign up button” with no details entered	GUI will prompt user saying information is missing	GUI will prompt user saying information is missing	Pass
L4	User clicks “Sign Up button” with password field missing	“Password is required” message is prompted	“Password is required” message is prompted	Pass
L5	User clicks the “Sign up button” with correct details in the form	The Administrator table in the database is updated with the new details entered	The Administrator table in the database is updated with the new details entered	Pass
L6	User clicks the x button at the top right of the GUI	GUI closes	GUI closes	Pass

Customer Details Screen

Test Data Number	Step	Expected Result	Result	Status
L1	User enters details into the text box field and click on the “Create Customer” button	All the customer details that got entered in the text box fields are sent to the customer table of the database, also table in GUI is updated	All the customer details that got entered in the text box fields are sent to the customer table of the database, also table in GUI is updated	Pass
L2	User selects a customer from the table in the GUI	The text box fields are then populated with the customer details	The text box fields are populated with the customer details	Pass
L3	User changes customer details and clicks the “Update Customer” button	The customer details are updated in the customer table of the database, also table in GUI is updated	The customer details are updated in the customer table of the database, also table in GUI is updated	Pass
L4	User selects a customer and clicks the “Delete Customer” button	Customer is deleted from the customer table of the database, also removes customer from the table in the GUI	Customer is deleted from the customer table of the database, also removes customer from the table in the GUI	Pass
L5	User clicks either the “Customer”, “Product” or “Invoice” button	The button the user selects, the screen pops up, for example user selects “Product” button, Product Screen is then displayed	The button the user selects, the screen pops up, for example user selects “Product” button, Product Screen is then displayed	Pass
L6	User clicks the x button at the top right of the GUI	GUI closes	GUI closes	Pass

Product Details Screen

Test Data Number	Step	Expected Result	Result	Status
L1	User enters details into the text box field and click on the "Create Product" button	All the product details that got entered in the text box fields are sent to the product table of the database, also table in GUI is updated	All the product details that got entered in the text box fields are sent to the product table of the database, also table in GUI is updated	Pass
L2	User selects a product from the table in the GUI	The text box fields are then populated with the product details	The text box fields are then populated with the product details	Pass
L3	User changes product details and clicks the "Update Product" button	The product details are updated in the product table of the database, also table in GUI is updated	The product details are updated in the product table of the database, also table in GUI is updated	Pass
L4	User selects a product and clicks the "Delete Product" button	Product is deleted from the product table of the database, also removes product from the table in the GUI	Product is deleted from the product table of the database, also removes product from the table in the GUI	Pass
L5	User clicks either the "Customer", "Product" or "Invoice" button	The button the user selects, the screen pops up, for example user selects "Customer" button, Customer Screen is then displayed	The button the user selects, the screen pops up, for example user selects "Customer" button, Customer Screen is then displayed	Pass
L6	User clicks the x button at the top right of the GUI	GUI closes	GUI closes	Pass

Invoice Details Screen

Test Data Number	Step	Expected Result	Result	Status
L1	User enters details into the text box field and click on the "Create Invoice" button	All the invoice details that got entered in the text box fields are sent to the invoice table of the database, also table in GUI is updated	All the invoice details that got entered in the text box fields are sent to the invoice table of the database, also table in GUI is updated	Pass
L2	User selects an invoice from the table in the GUI	The text box fields are then populated with the invoice details	The text box fields are then populated with the invoice details	Pass
L3	User changes invoice details and clicks the "Update Invoice" button	The invoice details are updated in the invoice table of the database, also table in GUI is updated	The invoice details are updated in the invoice table of the database, also table in GUI is updated	Pass
L4	User selects an invoice and clicks the "Delete Invoice" button	Invoice is deleted from the invoice table of the database, also removes invoice from the table in the GUI	Invoice is deleted from the invoice table of the database, also removes invoice from the table in the GUI	Pass
L5	User clicks either the "Customer", "Product" or "Invoice" button	The button the user selects, the screen pops up, for example user selects "Customer" button, Customer Screen is then displayed	The button the user selects, the screen pops up, for example user selects "Customer" button, Customer Screen is then displayed	Pass
L6	User clicks the x button at the top right of the GUI	GUI closes	GUI closes	Pass