Group Project Report

Coffee Delivery Web-app

Ma ZhiYuan – 201464

Wang YiDa – 201406

Li MengFan – 201505

Muhammad Izwan bin Suhaime – 193907

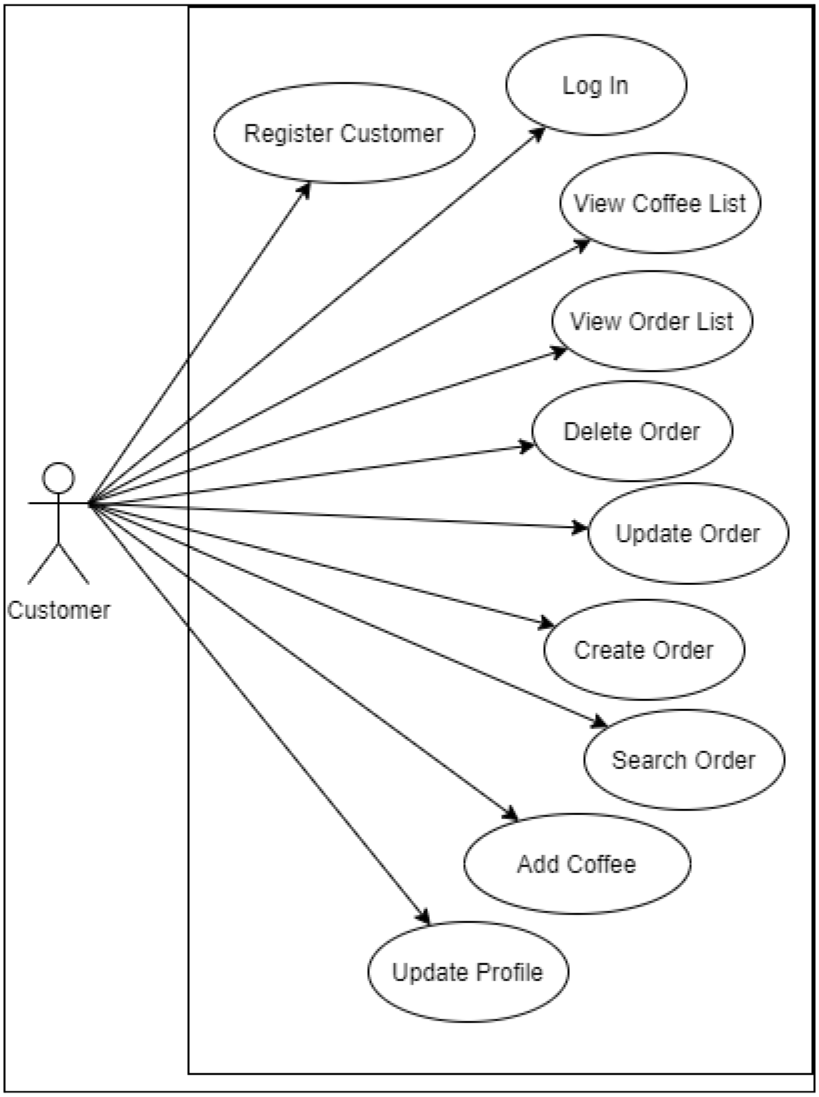
27th January, 2021

1.0 Introduction

This is a web application for placing coffee orders for customers to book coffee delivery.

2.0 Design Diagram

2.1 Use Case Diagram



2.2 Normalization

0NF

Customer [customerID, customerName, phone, (orderID, coffeeID, coffeeName, unitPrice, quantity, totalPrice, address)]

1NF

Customer [customerID, customerName,phone]
Order [customerID, orderID, coffeeID, coffeeName, unitPrice, quantity, totalPrice, address]

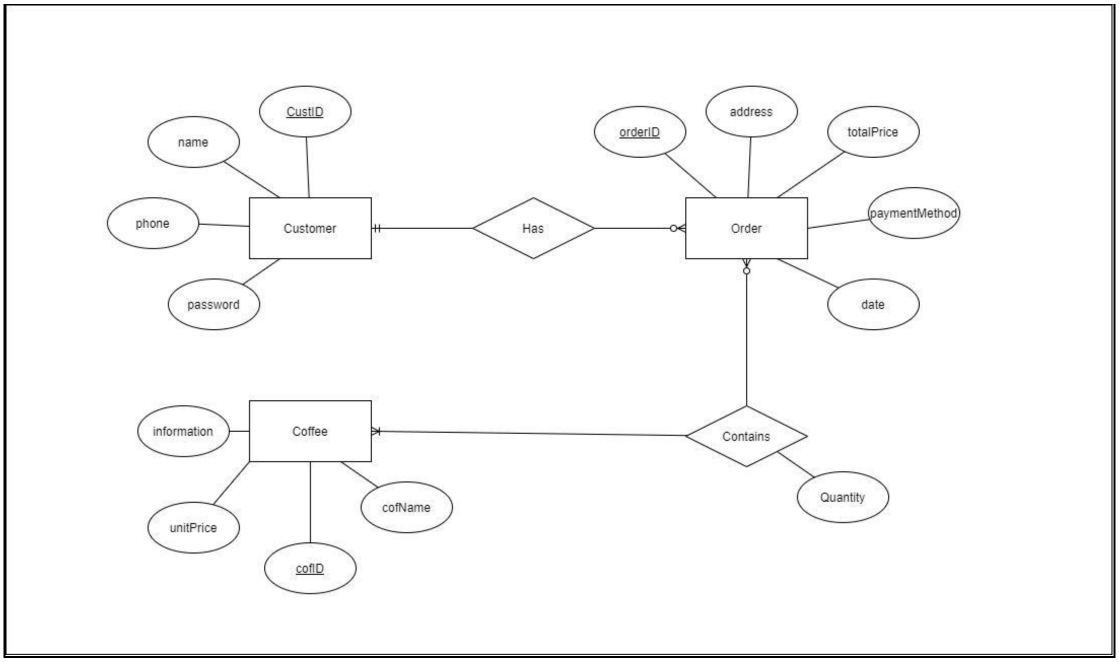
2NF

Customer [customerID,customerName,phone]
Order [customerID, orderID, coffeeID, quantity, totalPrice, address]
Coffee [coffeeID, coffeeName, unitPrice]

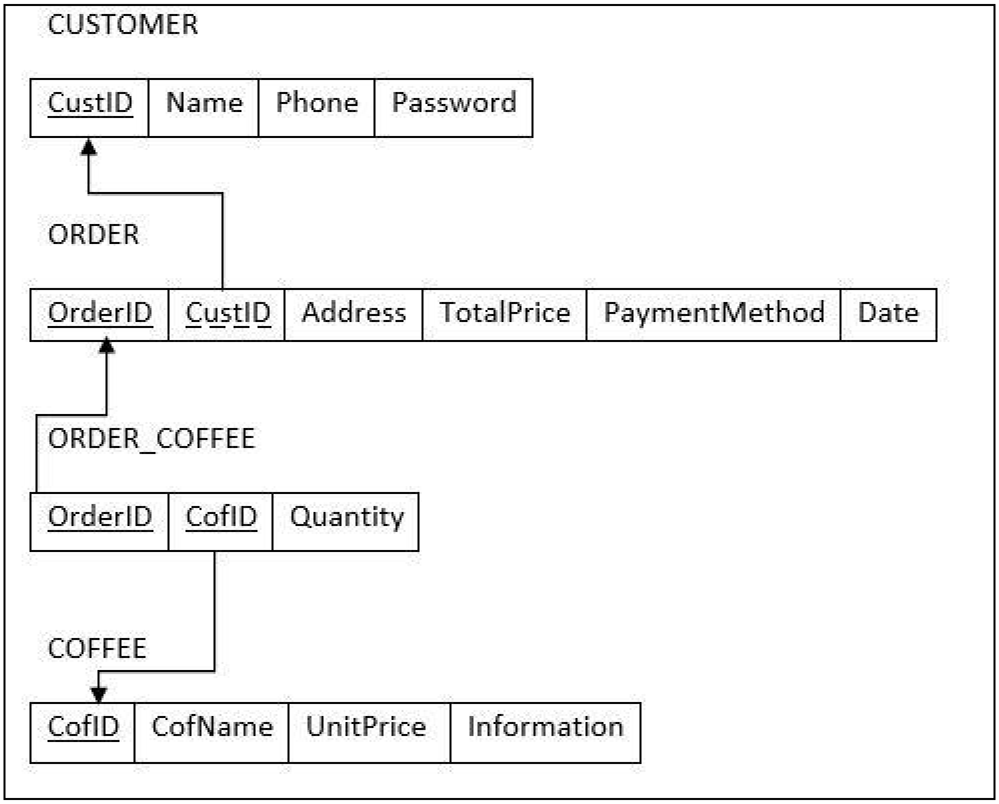
3NF

Customer [customerID,customerName,phone]
Order [customerID, orderID, totalPrice, address]
Cof_Order [orderID, coffeeID, quantity]
Coffee [coffeeID, coffeeName, unitPrice]

2.3 ERD Diagram



2.4 Mapping



CUSTOMER

CustID	Name	Phone	Password

ORDER

OrderID	CustID	Address	TotalPrice	PaymentMethod	Date

ORDER_COFFEE

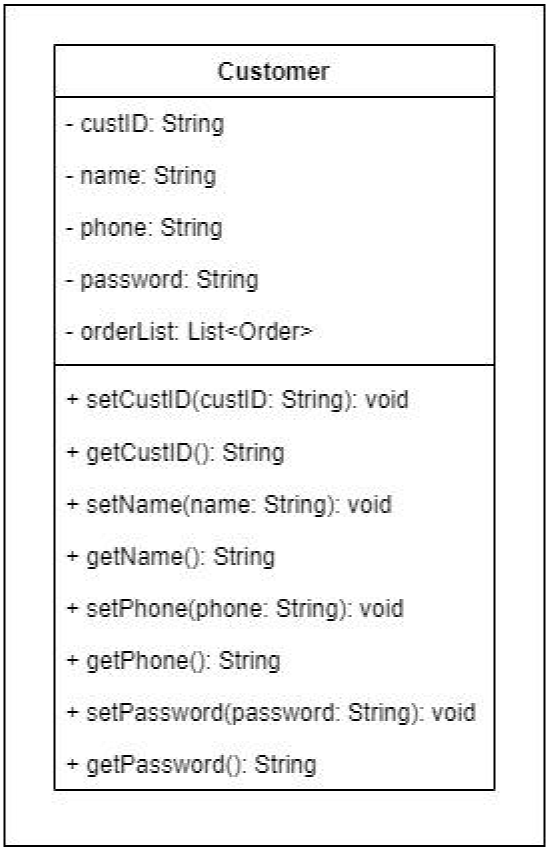
OrderID	CofID	Quantity

COFFEE

CofID	CofName	UnitPrice	Information

3.0 Model

3.1 Customer

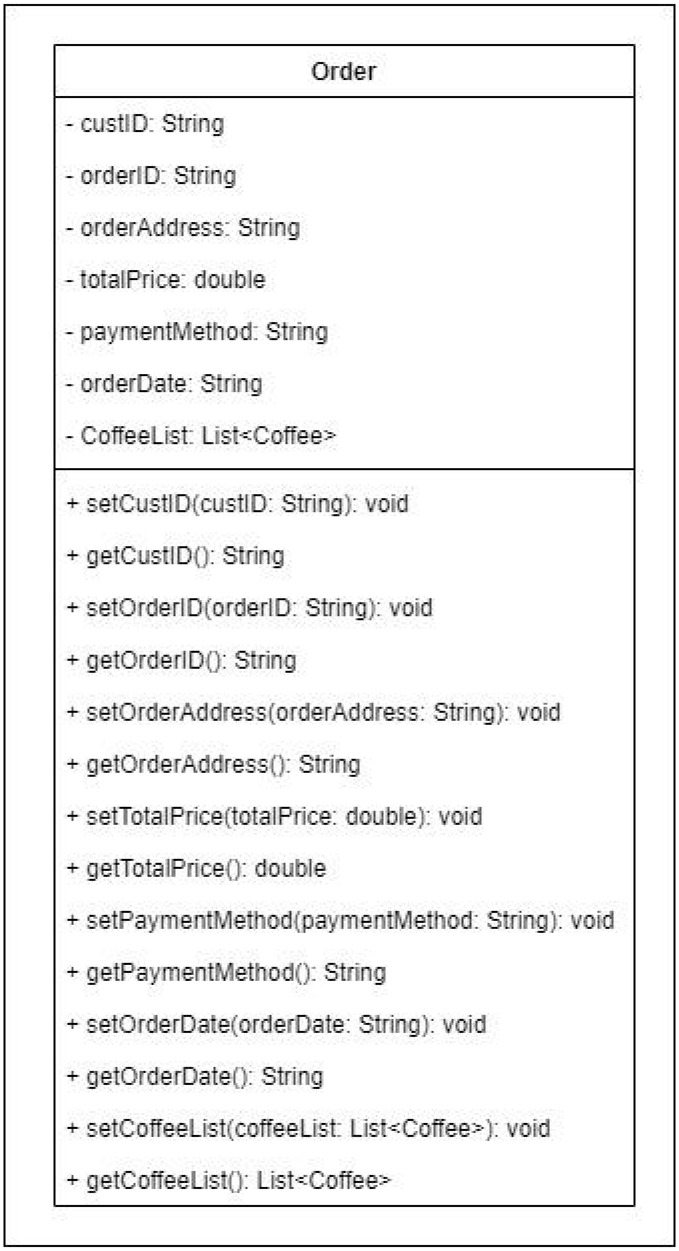


Customer

- custID: String
- name: String
- phone: String
- password: String
- orderList: List<Order>

+ setCustID(custID: String): void
+ getCustID(): String
+ setName(name: String): void
+ getName(): String
+ setPhone(phone: String): void
+ getPhone(): String
+ setPassword(password: String): void
+ getPassword(): String

3.2 Order



Order

- custID: String

- orderID: String

- orderAddress: String

- totalPrice: double

- paymentMethod: String

- orderDate: String

- CoffeeList: List<Coffee>

+ setCustID(custID: String): void

+ getCustID(): String

+ setOrderID(orderID: String): void

+ getOrderID(): String

+ setOrderAddress(orderAddress: String): void

+ getOrderAddress(): String

+ setTotalPrice(totalPrice: double): void

+ getTotalPrice(): double

+ setPaymentMethod(paymentMethod: String): void

+ getPaymentMethod(): String

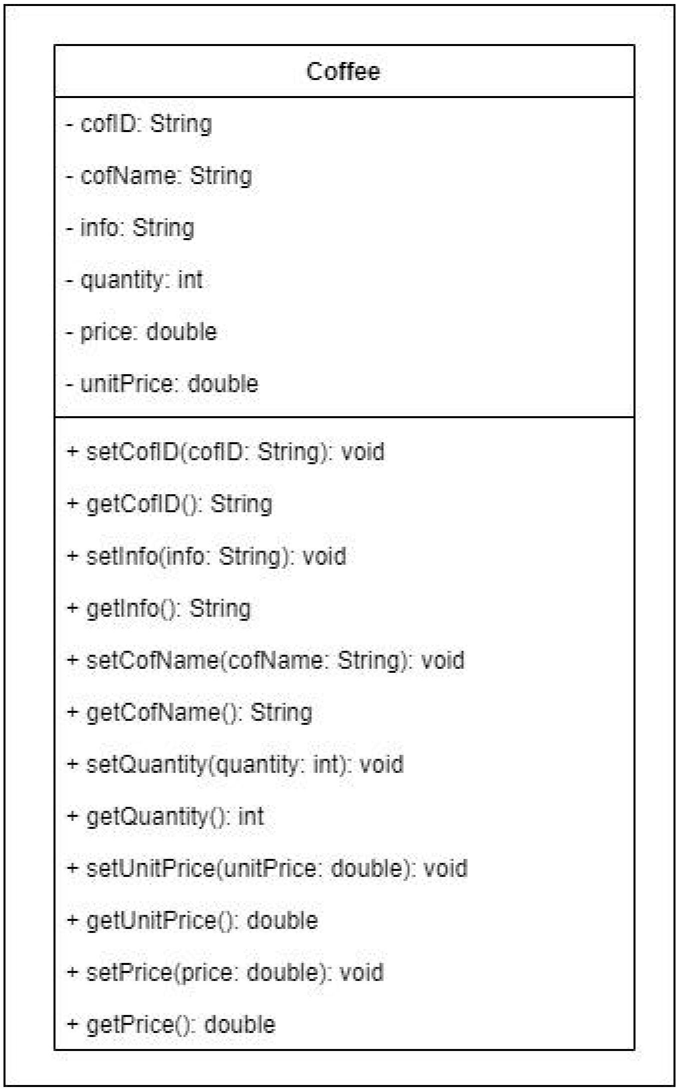
+ setOrderDate(orderDate: String): void

+ getOrderDate(): String

+ setCoffeeList(coffeeList: List<Coffee>): void

+ getCoffeeList(): List<Coffee>

3.3 Coffee



Coffee

- cofID: String

- cofName: String

- info: String

- quantity: int

- price: double

- unitPrice: double

+ setCofID(cofID: String): void

+ getCofID(): String

+ setInfo(info: String): void

+ getInfo(): String

+ setCofName(cofName: String): void

+ getCofName(): String

+ setQuantity(quantity: int): void

+ getQuantity(): int

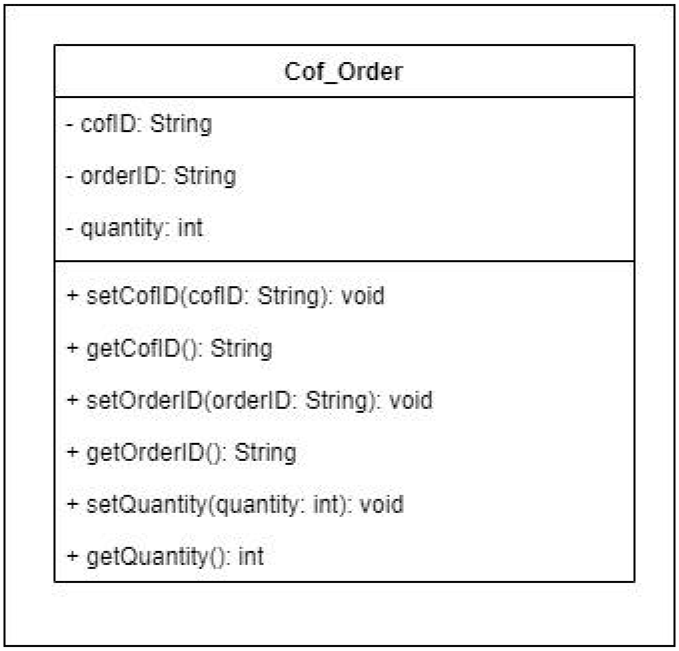
+ setUnitPrice(unitPrice: double): void

+ getUnitPrice(): double

+ setPrice(price: double): void

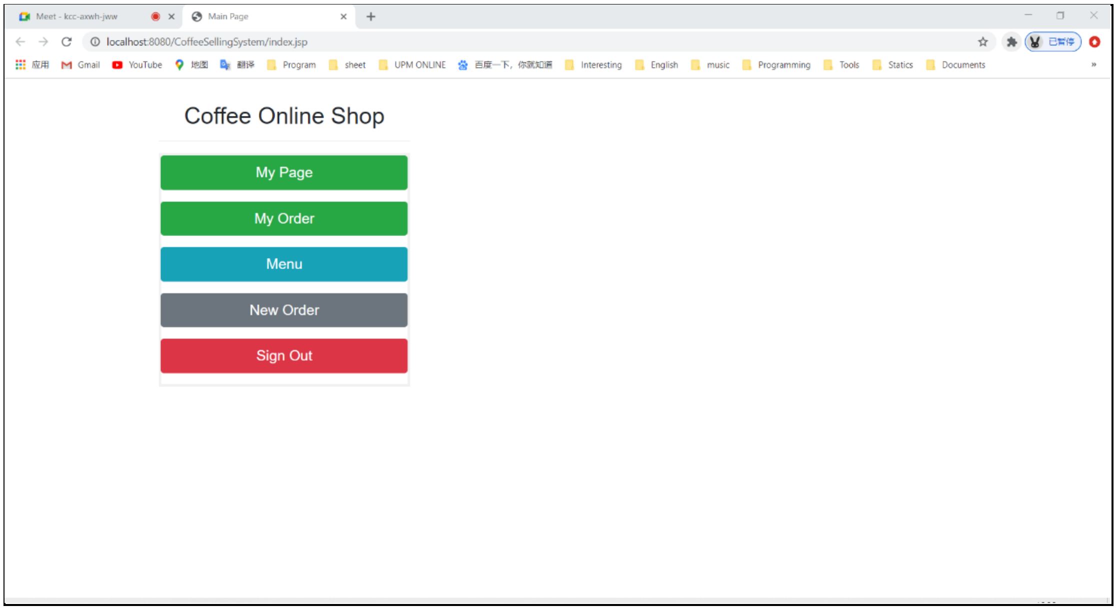
+ getPrice(): double

3.4 Cof_Order


Cof_Order
- cofID: String
- orderID: String
- quantity: int
+ setCofID(cofID: String): void
+ getCofID(): String
+ setOrderID(orderID: String): void
+ getOrderID(): String
+ setQuantity(quantity: int): void
+ getQuantity(): int

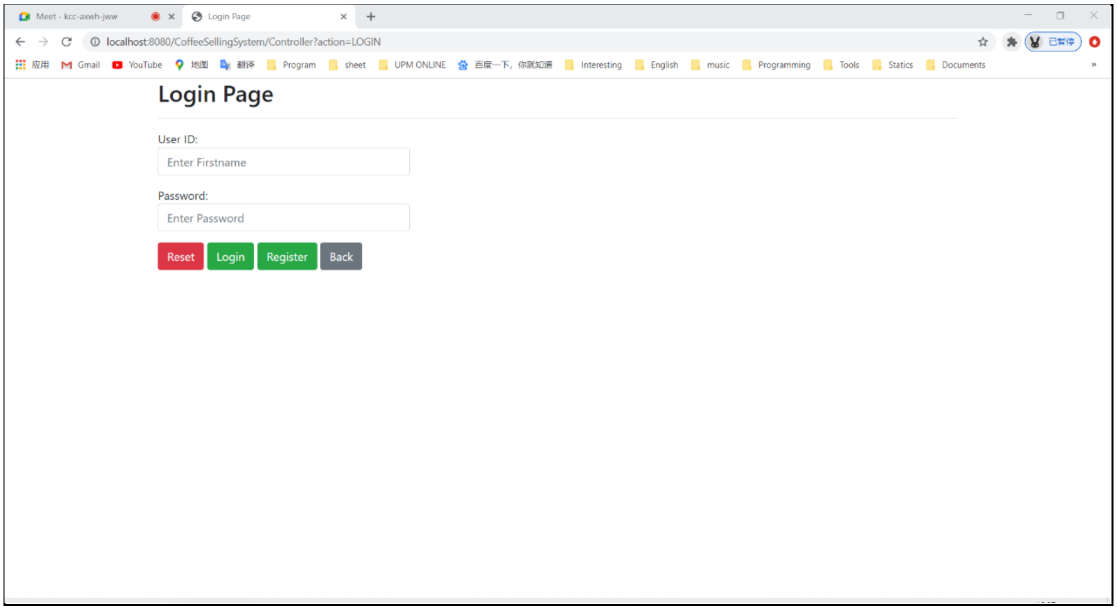
4.0 Interfaces

4.1 Main menu (Index.jsp)



This is the index of Coffee Online Shop. Clicking on any of the buttons in this interface other than the "sign out" and "menu" buttons will transfer the user to the "login" interface. If the user is already logged in, clicking on "My Page" will transfer the user to the private home page, clicking on "My order" will display the user's order, clicking on "menu" will display the available coffee types and prices, clicking on "New order" will allow the user to create a new order, and clicking on "Sign Out" will log you out.

4.2 Login (FormLogin.jsp)



Login Page

User ID:

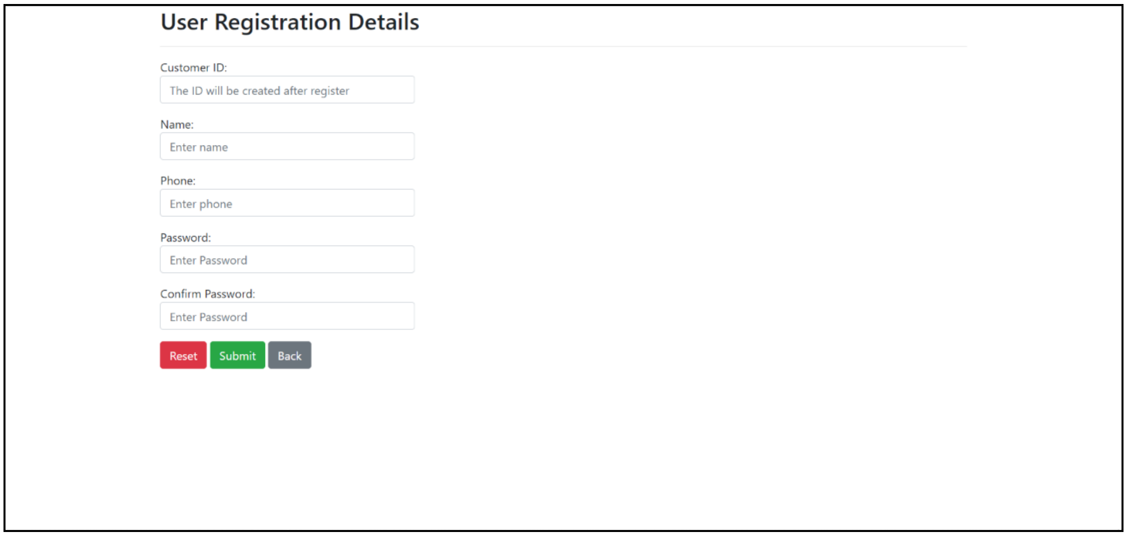
Enter Firstname

Password:

Enter Password

This is the Login page of Coffee Online shop, Users can login by entering ID and password, and return to the index page. If you do not have an account, you can select "Register" to enter the registration interface to register, and click "Reset" to clear the ID and password you have entered.

4.3 Registration (FormRegistration.jsp)



User Registration Details

Customer ID:

The ID will be created after register

Name:

Enter name

Phone:

Enter phone

Password:

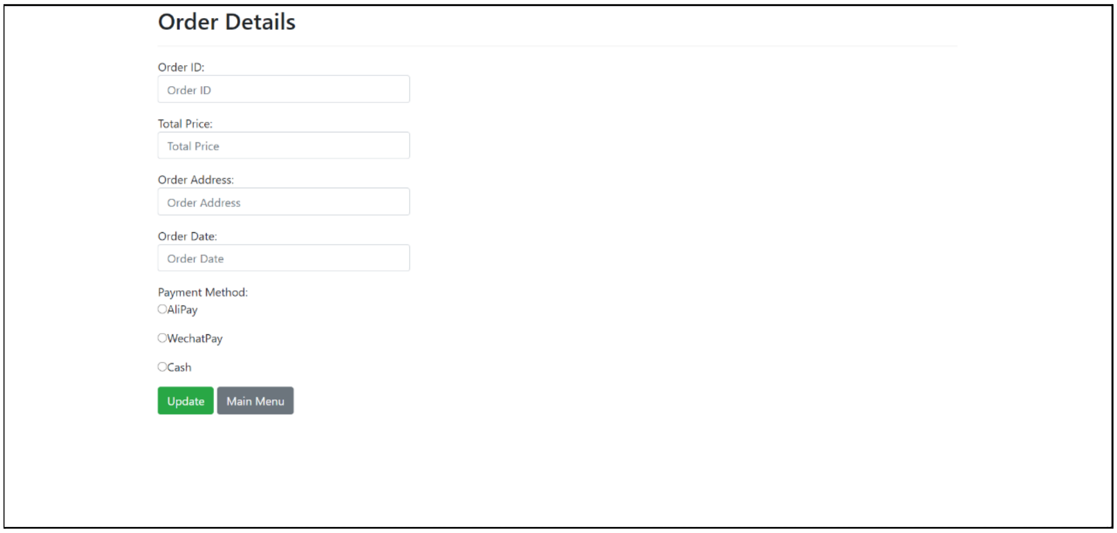
Enter Password

Confirm Password:

Enter Password

This is the registration screen, where users can enter their private information to register.

4.4 Order Detail Updating Page (FormUpdateOrderEdit.jsp)



Order Details

Order ID:

Order ID

Total Price:

Total Price

Order Address:

Order Address

Order Date:

Order Date

Payment Method:

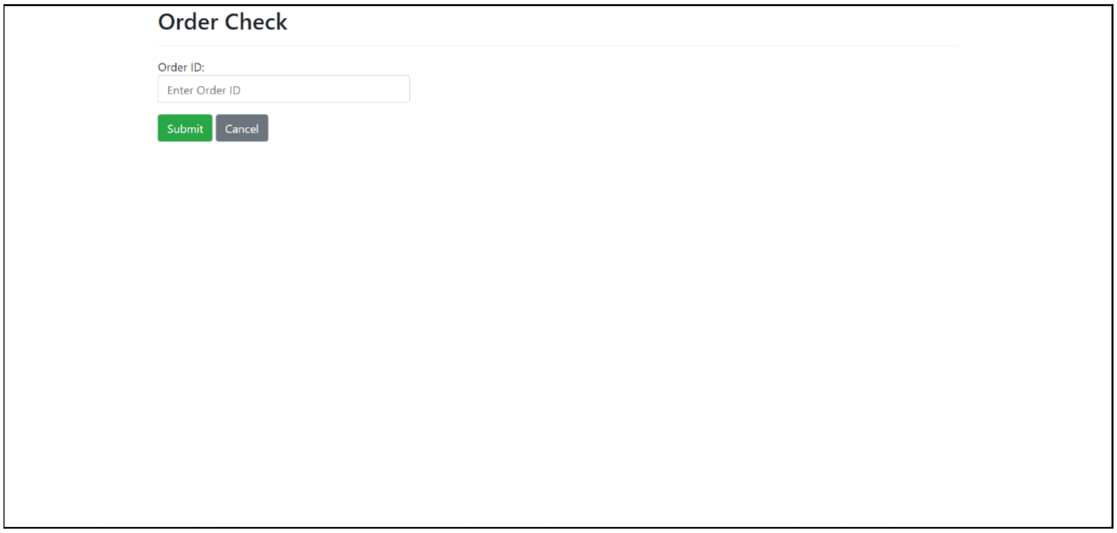
○AliPay

○WechatPay

○Cash

Users produce orders in this interface.

4.5 Order Checking Input Page (FormUpdateOrder.jsp)



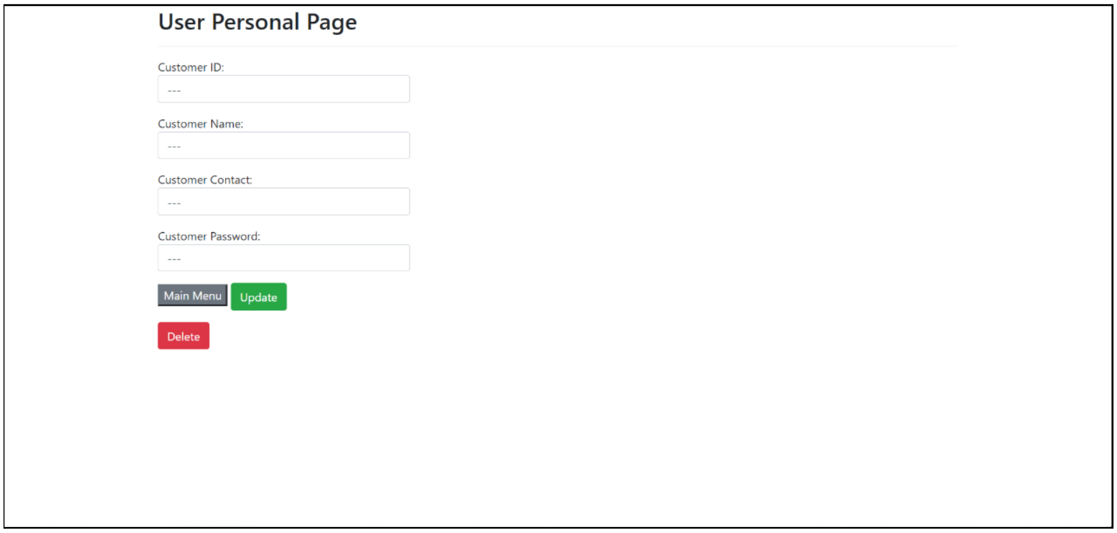
Order Check

Order ID:

Enter Order ID

The user enters the OrderID in this screen in order to view the details of the order.

4.6 Personal Page (FormPersonalPage.jsp)



User Personal Page

Customer ID:

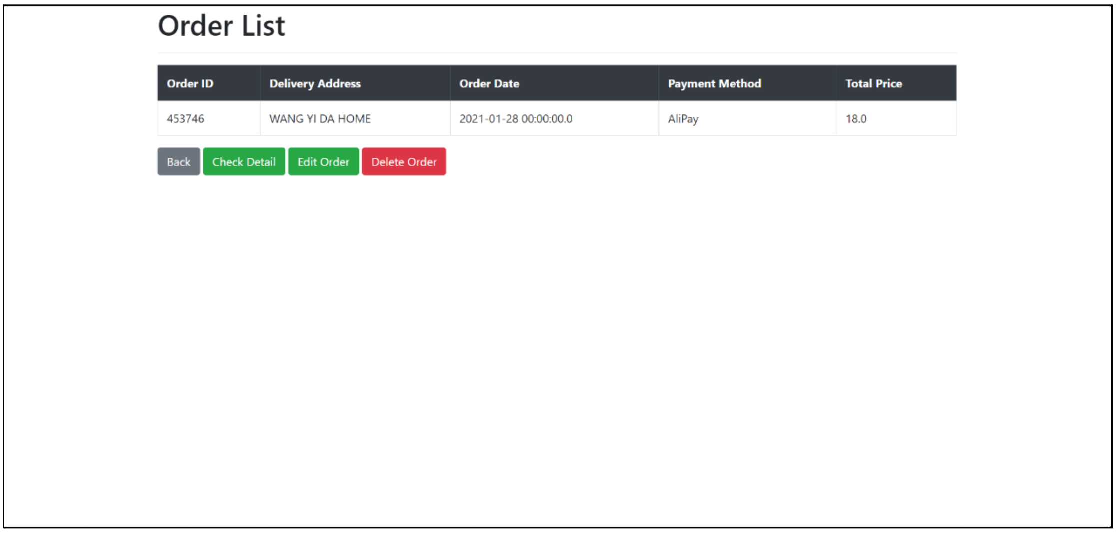
Customer Name:

Customer Contact:

Customer Password:

This screen is the user's personal home page, where the user can update his or her password.

4.7 Order List (FormOrderList.jsp)

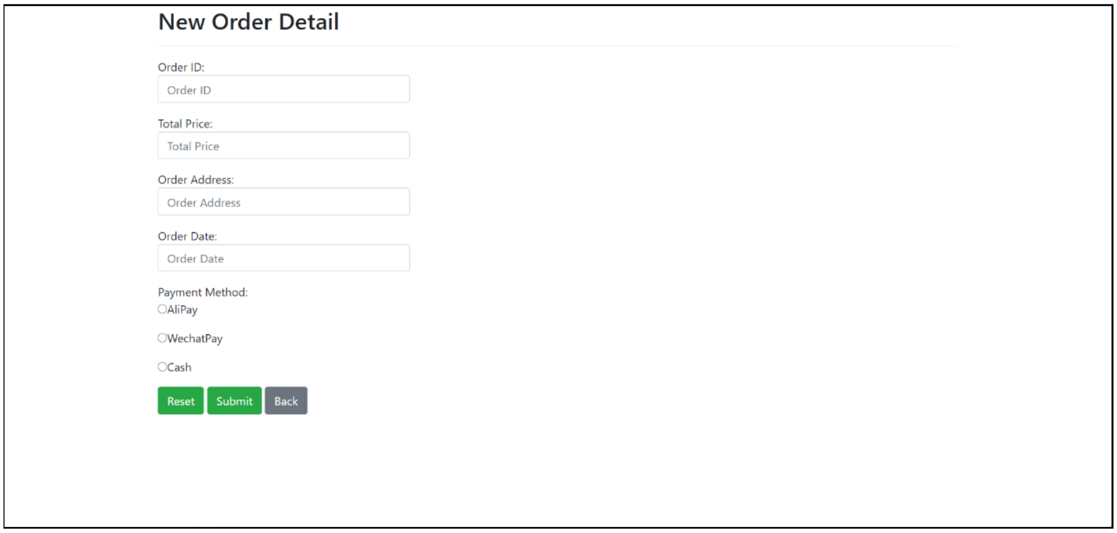


Order List

Order ID	Delivery Address	Order Date	Payment Method	Total Price
453746	WANG YI DA HOME	2021-01-28 00:00:00.0	AliPay	18.0

This screen displays the user's full order and full details.

4.8 New Order (FormNewOrder.jsp)



New Order Detail

Order ID:

Order ID

Total Price:

Total Price

Order Address:

Order Address

Order Date:

Order Date

Payment Method:

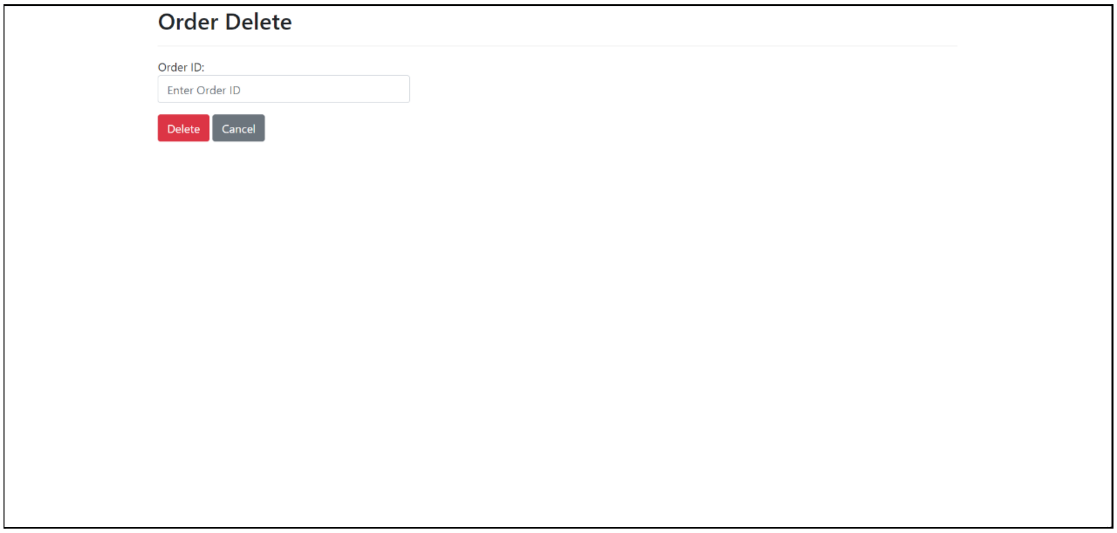
○AliPay

○WechatPay

○Cash

Users can create a new order on this screen.

4.9 Order Deleting Page (FormDeleteOrder.jsp)



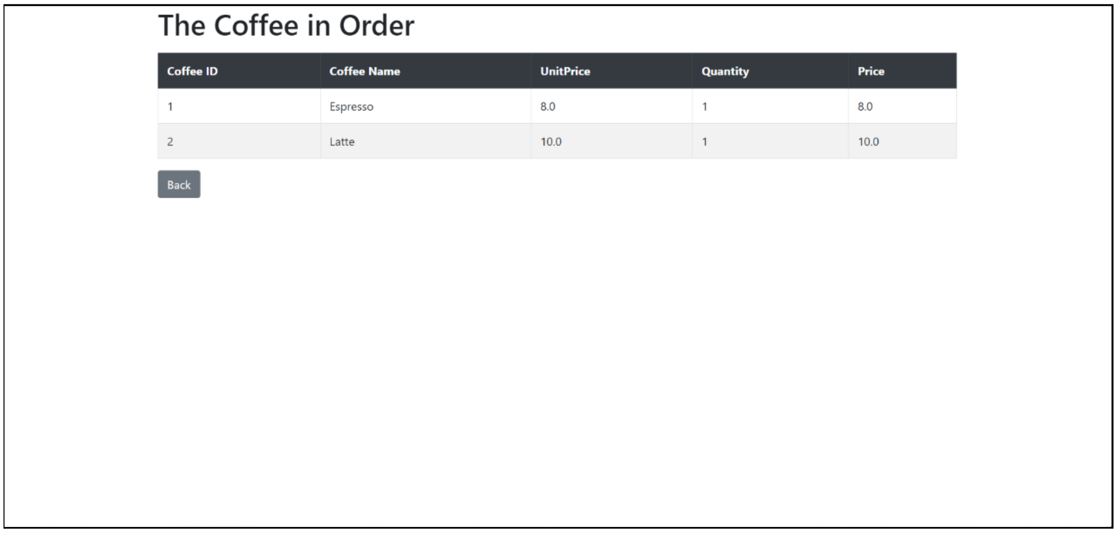
Order Delete

Order ID:

Enter Order ID

Users can delete an order by entering the order ID in this screen.

4.10 Coffee List In Order (FormCof_Order.jsp)

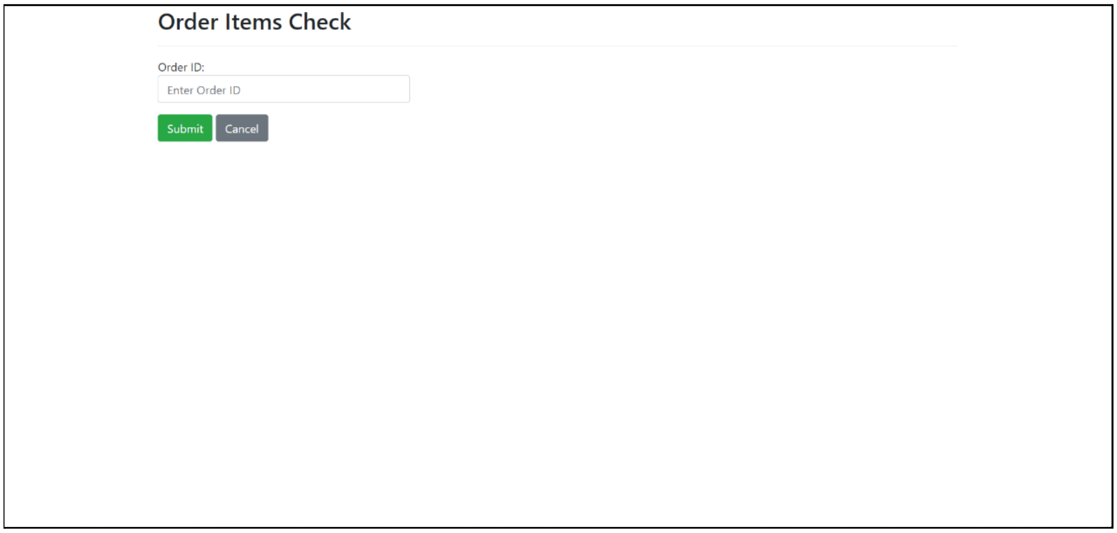


The Coffee in Order

Coffee ID	Coffee Name	UnitPrice	Quantity	Price
1	Espresso	8.0	1	8.0
2	Latte	10.0	1	10.0

This screen will display the unit price and quantity and total price of all the coffees in the order.

4.12 Check Coffee List (FormCheckCof.jsp)

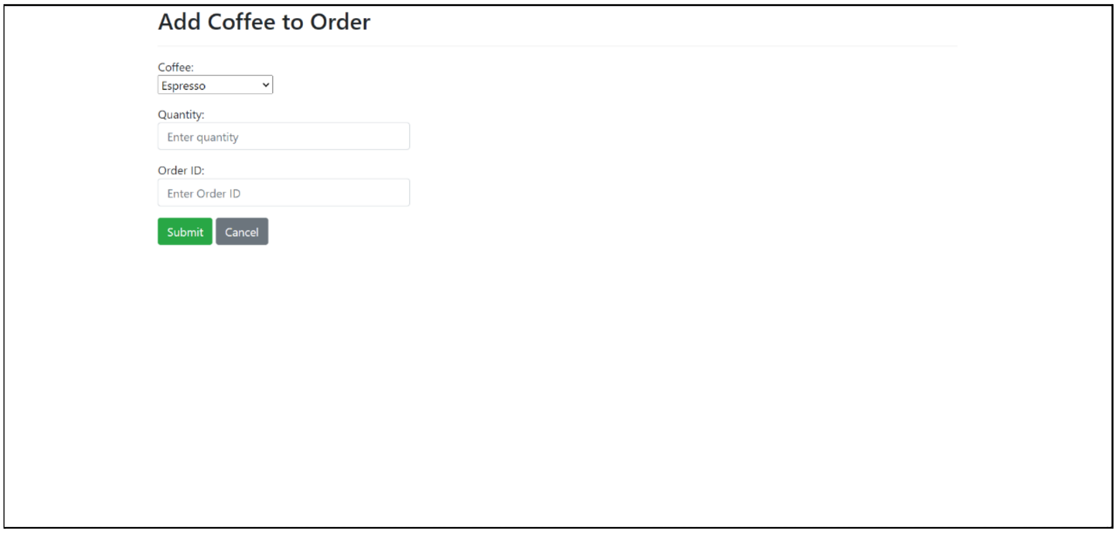


Order Items Check

Order ID: Enter Order ID

The user queries whether the order exists by entering the order ID on this page, and if it exists the details will be displayed.

4.13 Coffee Adding Page (FormAddCoffee.jsp)



Add Coffee to Order

Coffee: Espresso

Quantity:

Order ID:

Users can select the number and type of coffees to be added to a known order in this interface.

5.0 Controller (Servlet)

doGet(HttpServletRequest request, HttpServletResponse response)

This function includes actions like Login, Order, EditOrder, Exit, Menu. Aims to show the data about the order or customer, when the actions are executed, this function will give the feedback in the page.

doPost(HttpServletRequest request, HttpServletResponse response)

This function allows users to do Like Login, Register, Update, EditOrder, DeleteCustomer. Aims to edit data about the customers or the orders. The difference between doGet(...) and doPost(...) is that doGet(...) is the function to show the data, but doPost(...) aims to make edits.

registerCustomer(HttpServletRequest request, HttpServletResponse response)

This function is used for customers to register, including setCustID (by System), setName, setPhone and setPassword. When the customer data is saved successfully, System will give the feedback to show customer details.

newOrder(HttpServletRequest request, HttpServletResponse response)

Aims to add a new order, includes setCustID, setOrderID, setOrderDate, setTotalPrice, setOrderAddress and setPaymentMethod. If the customer has not registered, the system will back the exception. When orders already update successfully, System will give the details to the customer.

addCoffee(HttpServletRequest request, HttpServletResponse response)

The function is used for adding coffee to the order, it includes setCofID, setOrderID, setQuantity. Customers can order coffee after they login the system, and get the details when the coffee has been added in the order successfully.

loginCustomer(HttpServletRequest request, HttpServletResponse response)

The function is used for customers to log in, it includes the text field that asks customers to enter the ID and password. DAO will match the customer details and the system will show user details and print out" User Login Successfully!" if the data is correct.

If the ID and password are not matched, the system will print "Invalid customer ID or Password!"

listCustomer(HttpServletRequest request, HttpServletResponse response)

It will ask users to enter customerID then the system will back the customer personal page.

updateCustomer(HttpServletRequest request, HttpServletResponse response)

This function will show the customer personal page and allow customers to edit personal page details, Then update in the database, if the data updated successfully, the system will give feedback "User Updated Successfully!".

listOrder(HttpServletRequest request, HttpServletResponse response)

List the orders with the orderID for customer ordered

listCof_Order(HttpServletRequest request, HttpServletResponse response)

Contain in the orders, to show the types of coffee that have been ordered and quantities.

listCoffee(HttpServletRequest request, HttpServletResponse response)

List, the menu, all types of coffee to the user.

updateOrder(HttpServletRequest request, HttpServletResponse response)

Edit the order by inputting the orderID, users can check, edit and delete the order and its details.

deleteOrder(HttpServletRequest request, HttpServletResponse response)

This function is used for customers to delete their order, and the data will be uploaded, then give the feedback" Order Deleted Successfully!".

getSingleOrderUpdate(HttpServletRequest request, HttpServletResponse response)

This function is used for checking if the userID and the orderID matched.

6.0 DBConnection

This class contains the data about logging in the database.

7.0 DAO

saveCustomer(Customer c)

This function is used to get customers data and update the data in the database. System will get CustID, CustName, CustPhone, CustPassword.

saveOrder(Order o)

This function is used to get orders data and update the data in the database.System will get CustID, OrderID, OrderAddress, TotalPrice

addCoffee(Cof_Order c)

This function is used to ask customers to add new coffee in the order.System will get CofID, OrderID, Quantity.

listCustomer()

This function is used to list the customer details which includes CustID, CustName, CustPhone.

listAllOrder()

This function is used to show all orders as a list. It includes OrderID, OrderAddress, RrderDate, PaymentMethod.

listOrder(String id)

This function is used to get the orderID, then show the data of specified order. It includes OrderID, OrderAddress, OrderDate, PaymentMethod.

RandomID()

This function is used to make IDs for orders or customers with 9 numbers.

getRandomID(String choose)

This function is used to get the ID form RandomID(), it provides choices(CUSTOMER or ORDER) and judges the IDs the same or not.by IDSame(String id, String choose).

IDSame(String id, String choose)

This function is used to check the repetition for IDs.

getDate()

This function is used to get the recent date as "DD/MM/YYYY".

listCof_Order(String orderID)

This function is used to get the correspondence between coffee and orders. It includes Quantity, CofID, OrderID.

transferCof_Order(Cof_Order co)

This function is used to order the coffee price with code which the order already had. It includes Quantity, Price.

listCoffee()

This function is used to get the coffee list and show the details about different types of coffee. It includes CofID, CofName, UnitPrice, Info.

calcTotalPrice(String orderID)

This function is used to calculate the total price for one order. It will get quantity and unit price to calculate the total price as “TotalPrice”. The function includes OrderID, TotalPrice.

loginCustomer(String id, String pw)

This function is used to do log in for customers. Customers need input id and password.

checkCustomer(String custID)

This function is used to check if the CustID is available.

checkOrder(String orderID)

This function is used to check if the OrderID is available.

getCustomer(String custID)

This function is used to get details of the customers, and to show as a table. It includes CustID, Name, Phone.

update(Customer customer)

This function is used to update the data of customers. System will get CustID, Name, Password, Phone.

editOrder(Order order)

This function is used to edit orders

delete(String custID)

This function is used to delete orders. The System will delete the data which is matched with CustID.

getOrder(String orderID)

This function is used to show details about one order. It includes OrderID, OrderAddress, TotalPrice, PaymentMethod, OrderDate.

deleteOrder(String orderID)

This function is used to delete the orders. System will delete the data which is matched with OrderID.

orderUnderCustomer(String custID, String orderID)

This function is used to check if the CustID gets matched with OrderID. It returns true or false.