MULTIMEDIA UNIVERSITY

PROJECT

TOP2121 – OBJECT-ORIENTED PROGRAMMING

TRIMESTER 1, 2023/2024 (TERM 2310)

Project Title: Point of Sales (POS) Management System

Lab Section: 1A

Submission Date: 13 February 2024

NO	STUDENT ID	STUDENT NAME	MAJOR
1.	1201101513	THAM THAO	AI
2.	1201102471	LIEW WEI MING	BIA
3.	1201102676	PHANG AO ZAN	BIA
4.	1201102736	KANG HE YUN	ST

Table of Contents

Task Distribution

No.	Student ID	Student Name	Tasks Completed
1.	1201101513	THAM THAO	• Discussion with group members and distribute tasks. • Complete distributed tasks. • Compile all the parts for programming, report, and video.
2.	1201102471	LIEW WEI MING	• discussion with group members • fulfilling tasks given by group leader • take part in documentations • take part in coding part for the project • system testing and error fixing with group members
3.	1201102676	PHANG AO ZAN	• discussion with group members • fulfilling tasks given by group leader • take part in documentations. • take part in coding part for the project. • system testing and error fixing with group members.

4.	1201102736	KANG HE YUN	• discussion with group members • fulfilling tasks given by group leader • take part in documentations. • take part in coding part for the project. • system testing and error fixing with group members.

Introduction

In today's dynamic business landscape, reliance on manual labor for information storage is increasingly considered antiquated and inefficient. Traditional methods, such as recording information on paper, not only incur significant costs but also lead to issues like data duplication and confusion. The inefficiency is compounded by the substantial storage space required for physical records, posing both logistical challenges and environmental concerns. In response to these challenges, our group is proud to introduce a cutting-edge Point of Sales Management System. This system is designed to empower users in efficiently managing business information and inventory, offering a more sustainable, user-friendly, and time-saving alternative. Key features include customer information retention, product price customization, and a comprehensive record of invoices. By automating tasks that were once prone to human error, our Point of Sales Management System not only enhances operational efficiency but also contributes to a more sustainable and eco-friendly business environment. Embracing this technology is not just a smart idea; it's a strategic imperative to stay competitive in the modern era. The intuitive user interface of our system further sets it apart from conventional software packages, ensuring swift navigation and seamless integration into daily business operations. It is our firm belief that by adopting this innovative solution, businesses can not only streamline their processes but also stay ahead in this fast-paced and ever-evolving business landscape.

Problem Statements

The main goal of the POS system is to solve problems which the users encountered by storing data with human power such as:

- Waste of resources: It is a waste of resources to record information using conventional techniques like a pen and pencil.

- Inconsistent data: The data on customers, inventory, invoices, and income may not be consistent due to human error when recording information.

- Lack of efficiency: An excessive number of tasks carried out by the user manually may result in inefficiency.

The information that is manually stored, such as entries in a book or paper, which requires labor, will waste resources if there is no Point-of-Sale management system. Occasionally, incorrect data might also result from using manpower to manually record the information. Additionally, looking for a certain piece of information manually across a series of documents takes time and is inefficient.

Objectives

A point-of-sale (POS) system is an essential tool used in various industries to facilitate transactions. When creating a POS system using Java, there are several objectives that need to be addressed. These objectives include:

User-Friendly Interface

The POS system will feature a user-friendly interface that allows for easy navigation and data entry. This will enable the user to effortlessly perform various sales transactions, such as adding items to the cart, processing payments, and generating receipts.

Cost-Effectiveness:

Java is an open-source programming language, which means it is free to download and use. Developing a POS system using Java can significantly reduce upfront licensing costs and long-term maintenance expenses.

Data Management:

A well-designed POS system needs to handle large amounts of data efficiently. The system allows businesses to analyze sales data, generate reports, and make data-driven decisions to optimize business performance.

Flexibility and Customization:

Each business has its unique requirements. A Java-based POS system allows for customization and tailoring to meet specific business needs. Java's modular architecture enables developers to add new features, modify existing ones, and create tailored solutions that address the specific requirements of a business.

Efficiency and Accuracy:

Java provides a robust and scalable platform that can easily handle high volumes of transactions. By using Java, the POS system can be designed to perform calculations, maintain data integrity, and ensure accurate record-keeping.

Program Scope

By using the Point of Sales (POS) system, the user can carry out simple tasks such as calculating total price in the cart, recording data of the products, recording of customer information, recording invoice details, and keeping track of income from the POS. These tasks will be easier to carry out and more convenient when users use this POS system. Users can insert, delete, and modify data in this POS system.

Features of POS system

- Efficient Transaction Processing
- Inventory Management
- Customer Information Management
- Invoice Management
- Income Tracking

Detailed Description

Login Screen: Users must register to login to our system.

Home Page: User is able to select different buttons that navigate to different pages that have different functions.

Point of Sales Page: User is able to carry out payment activities in this page.

Customer Page: User can add, delete, and update customers' information and view them in table form.

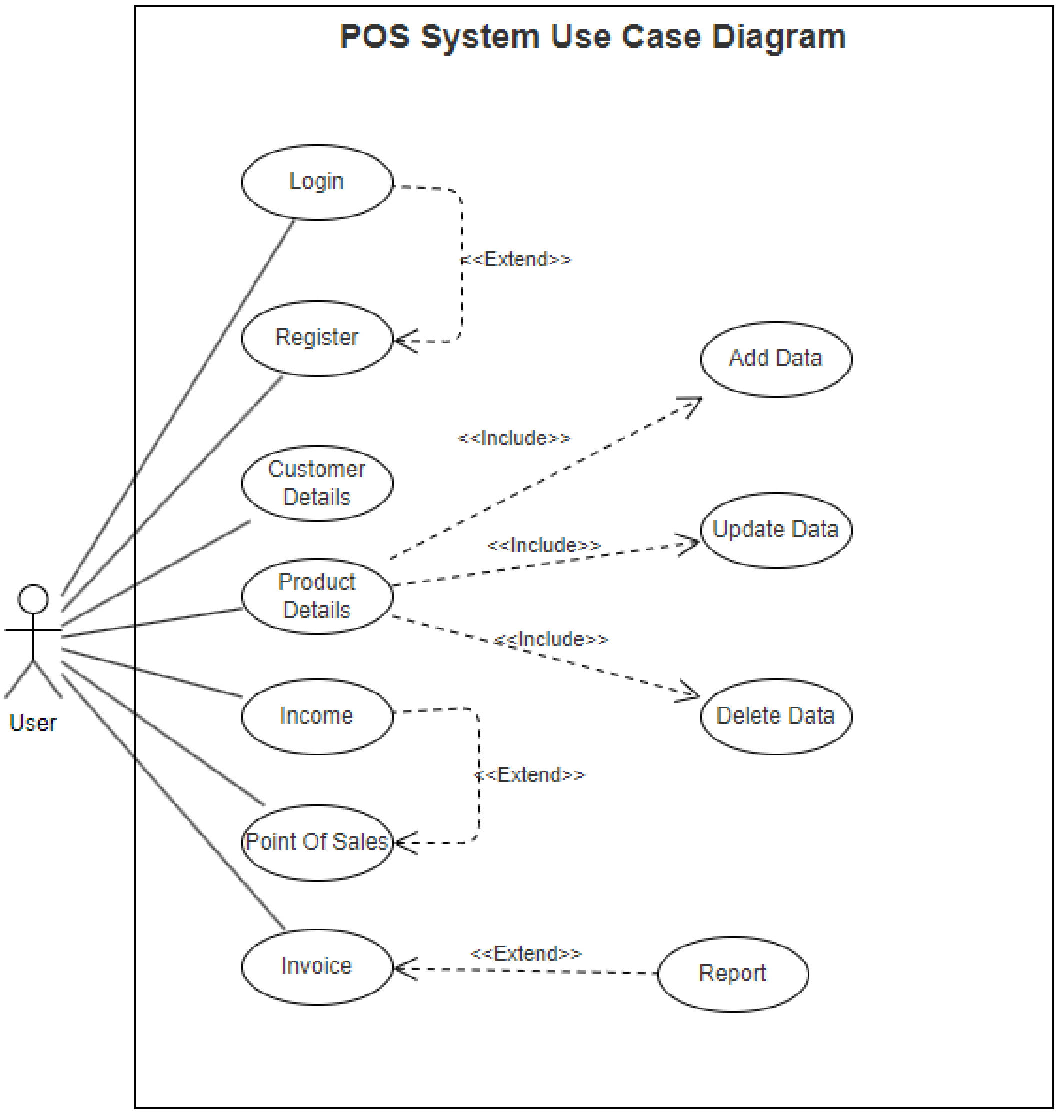
Product Page: User can add, delete, and update products' information and view them in table form.

Income Page: User can keep track of the income that been made by this POS which data is been recorded in a text file.

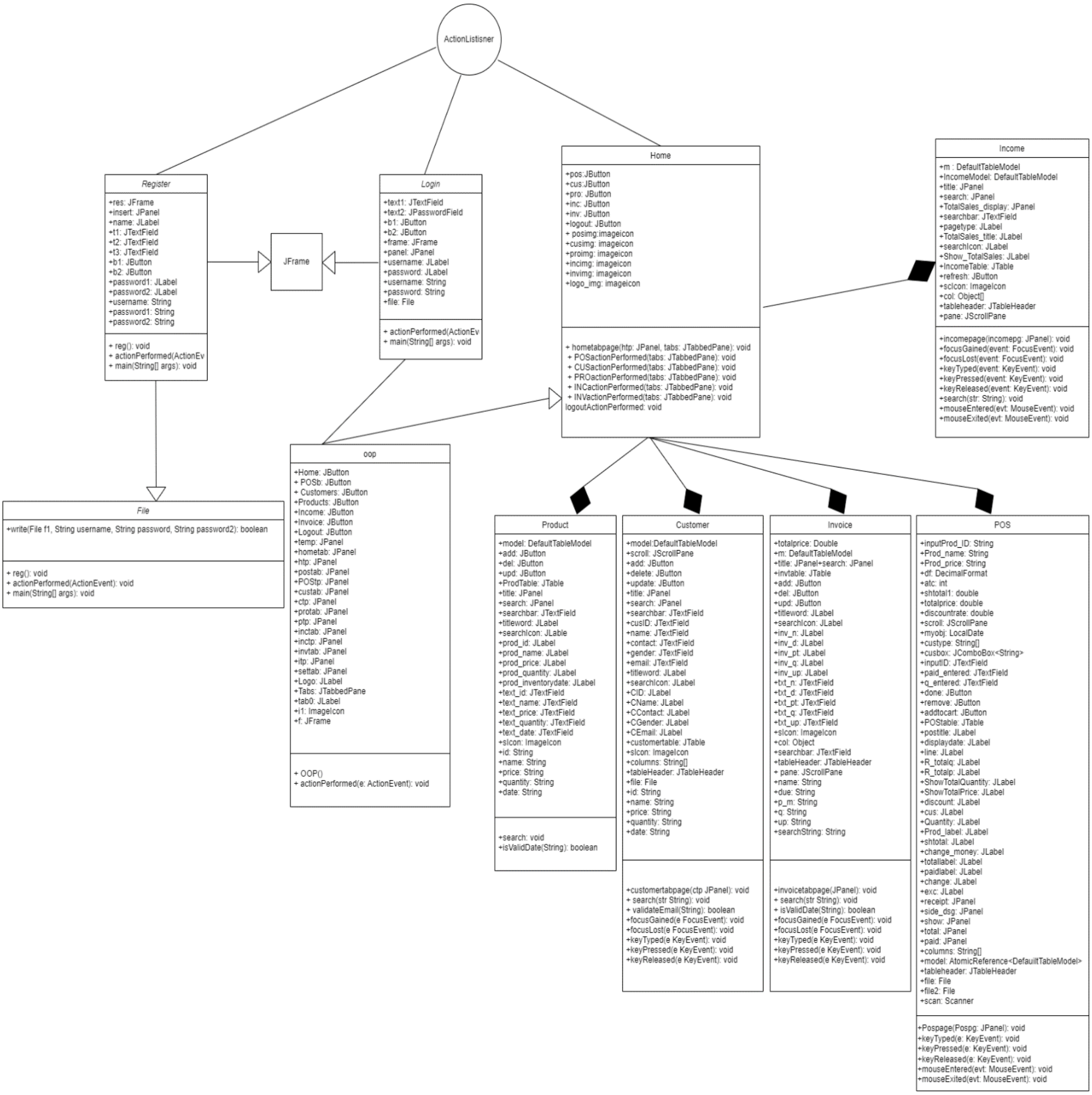
Invoice Page: User can add, delete, and update invoices' information and view them in table form.

System Design

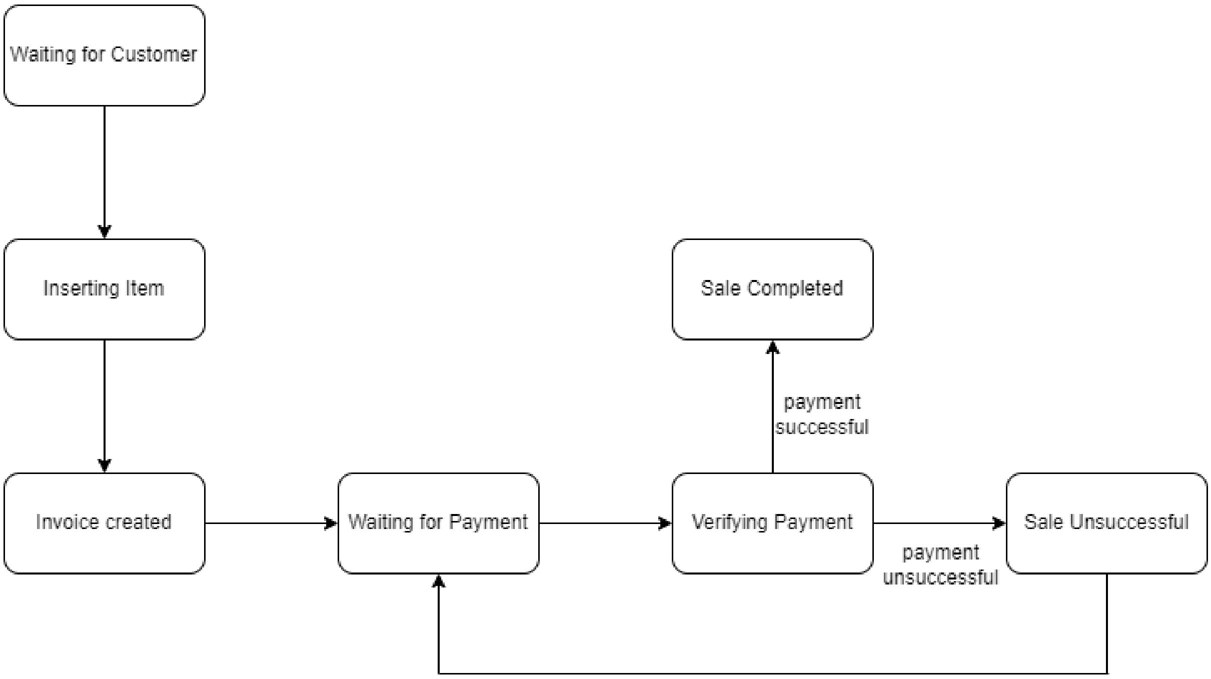
Use Case Diagram



Class Diagram

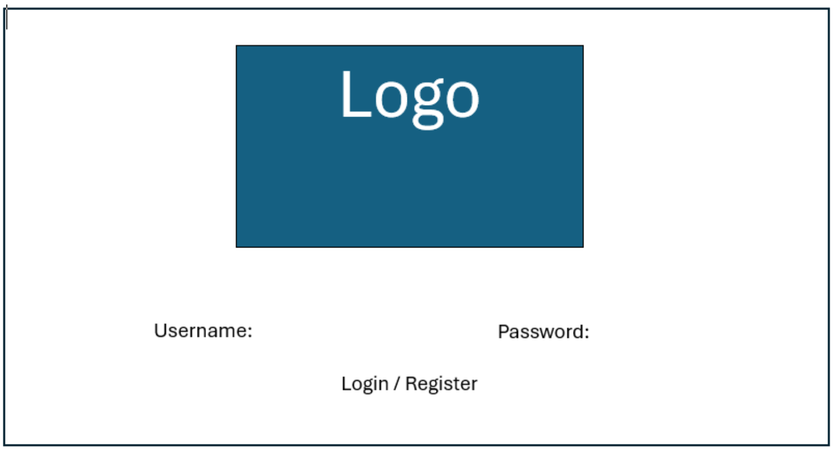


State Diagram

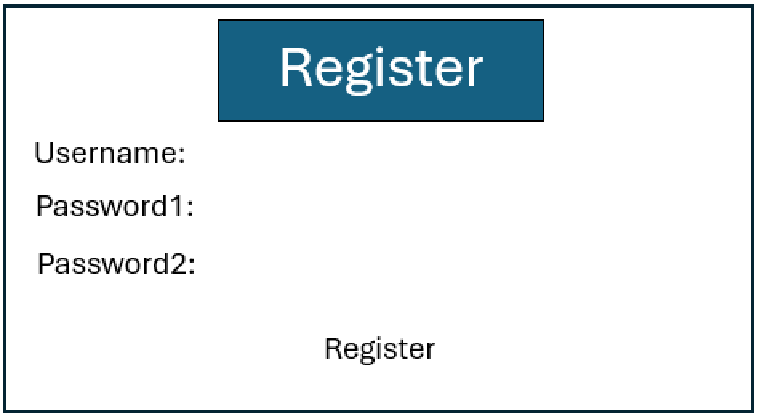


Draft Interface Design

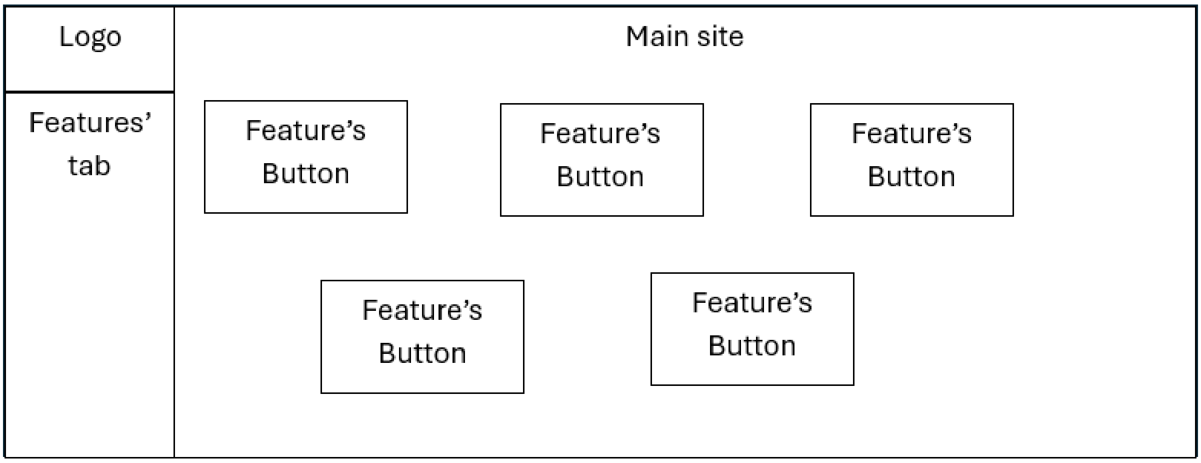
Login Page



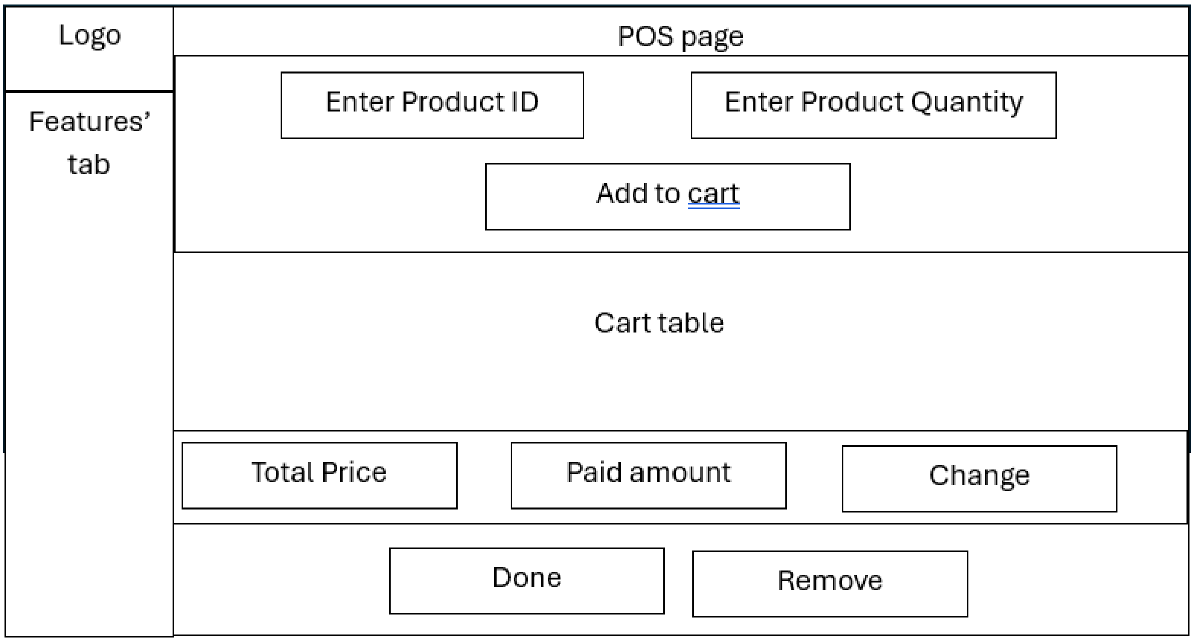
Register Page



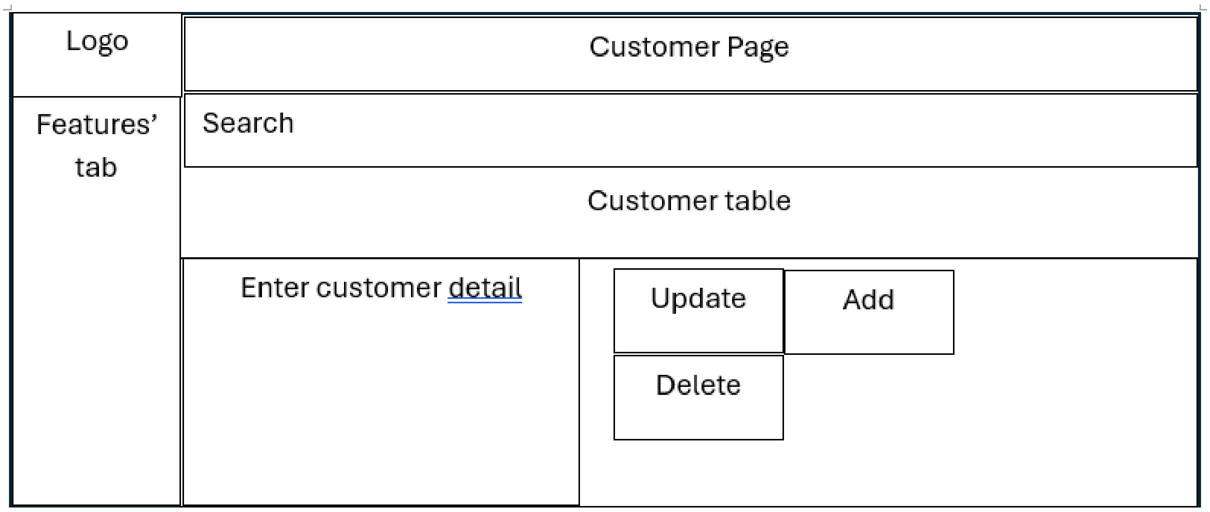
Home Page



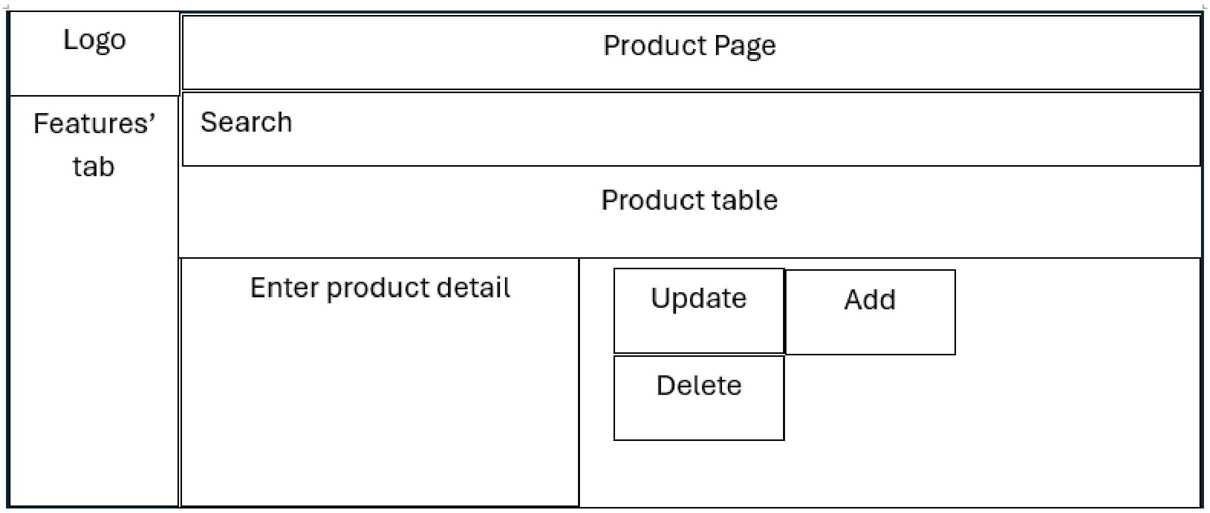
POS Page



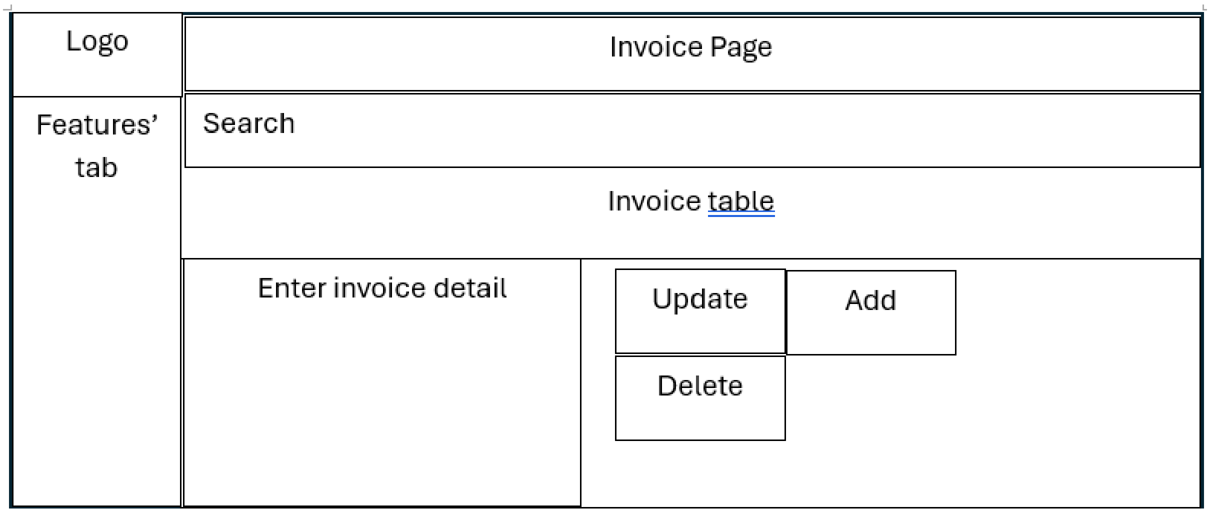
Customer Page



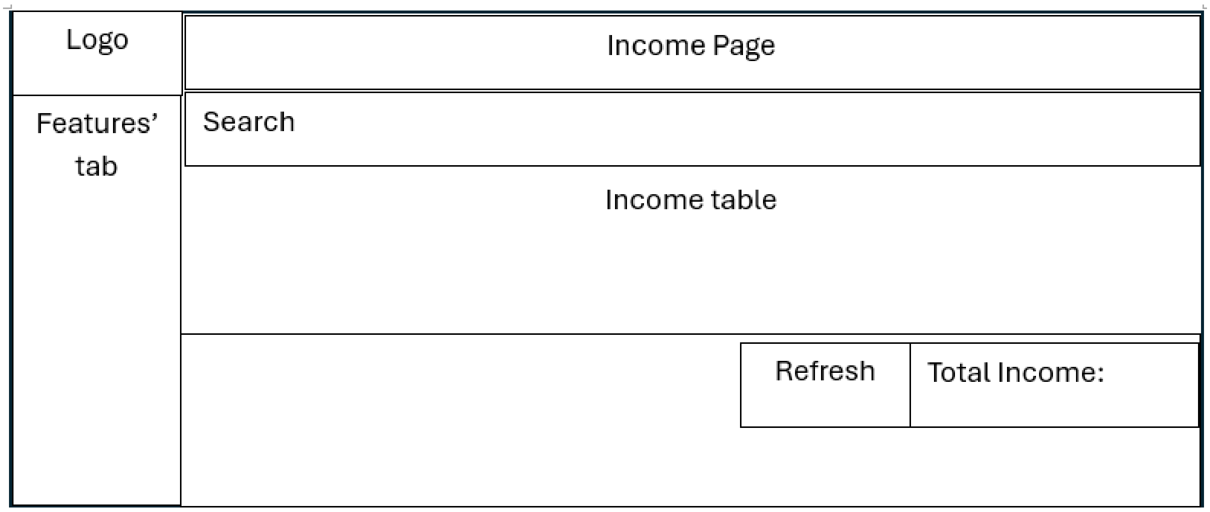
Product Page



Invoice Page



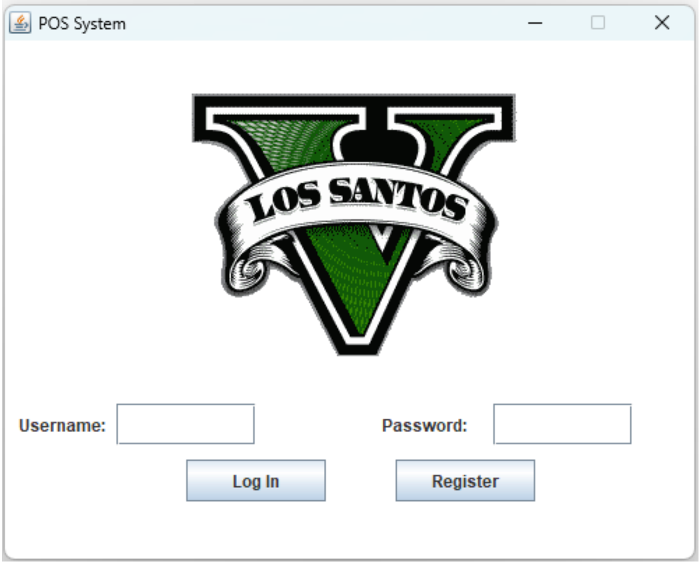
Income Page



Screenshot of the program

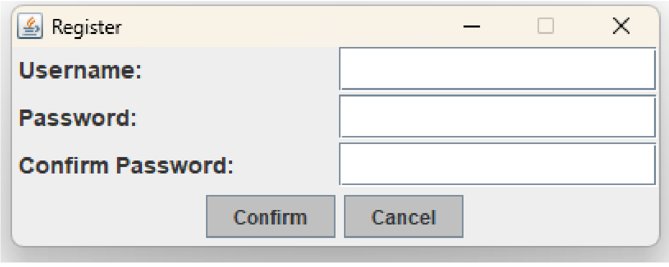
Login page

When the user keys in the command to start the system, the login page will come out.

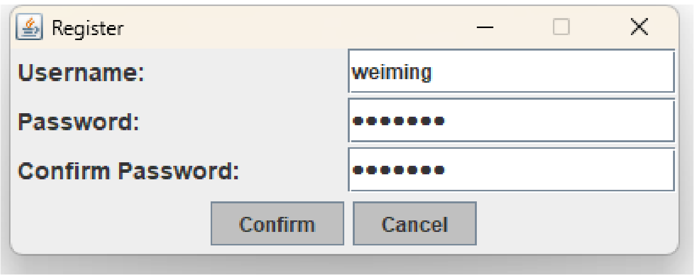


Register Page

If the user didn't have the account to login, the user can click the 'Register button', and the registration form will come out.

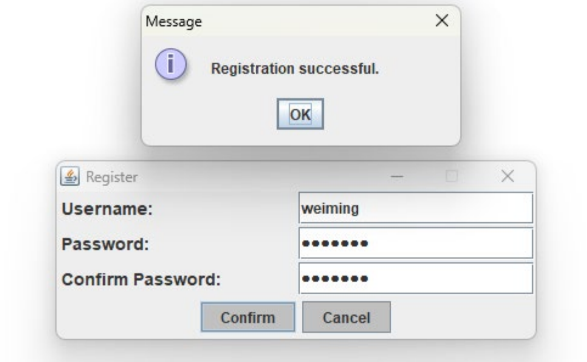


The user must fill in the details to register as a new user.



Username: weiming
Password: •••••••
Confirm Password: •••••••

If the user is successfully registered as a new user, a message of successful registration will pop out.



Message

Registration successful.

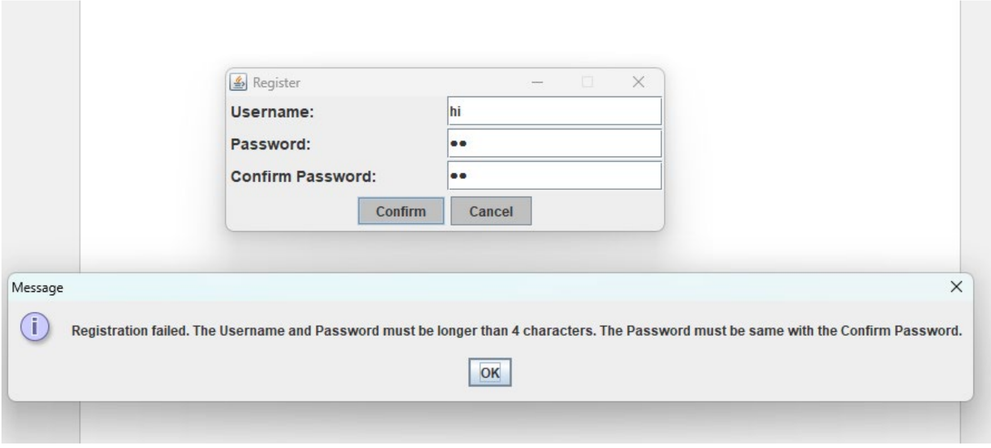
Register

Username: weiming

Password: •••••••

Confirm Password: •••••••

If the user did not fulfill the requirements to register as a new user, an error message will pop out, and the user will be needed to register again.



Register

Username: hi

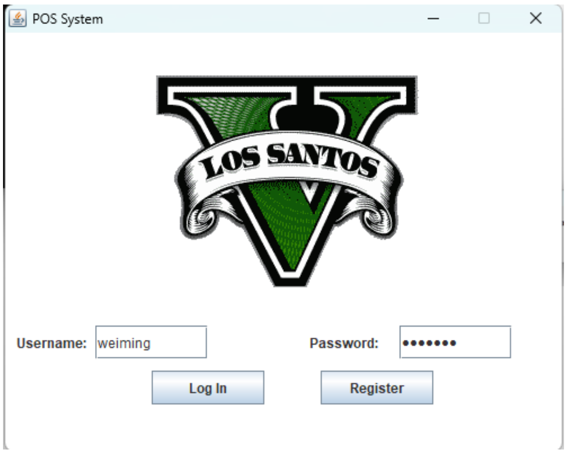
Password: ••

Confirm Password: ••

Message

Registration failed. The Username and Password must be longer than 4 characters. The Password must be same with the Confirm Password.

Then the user can use the new user details to login. Once the user is logged in successfully, the system will go to the home page.



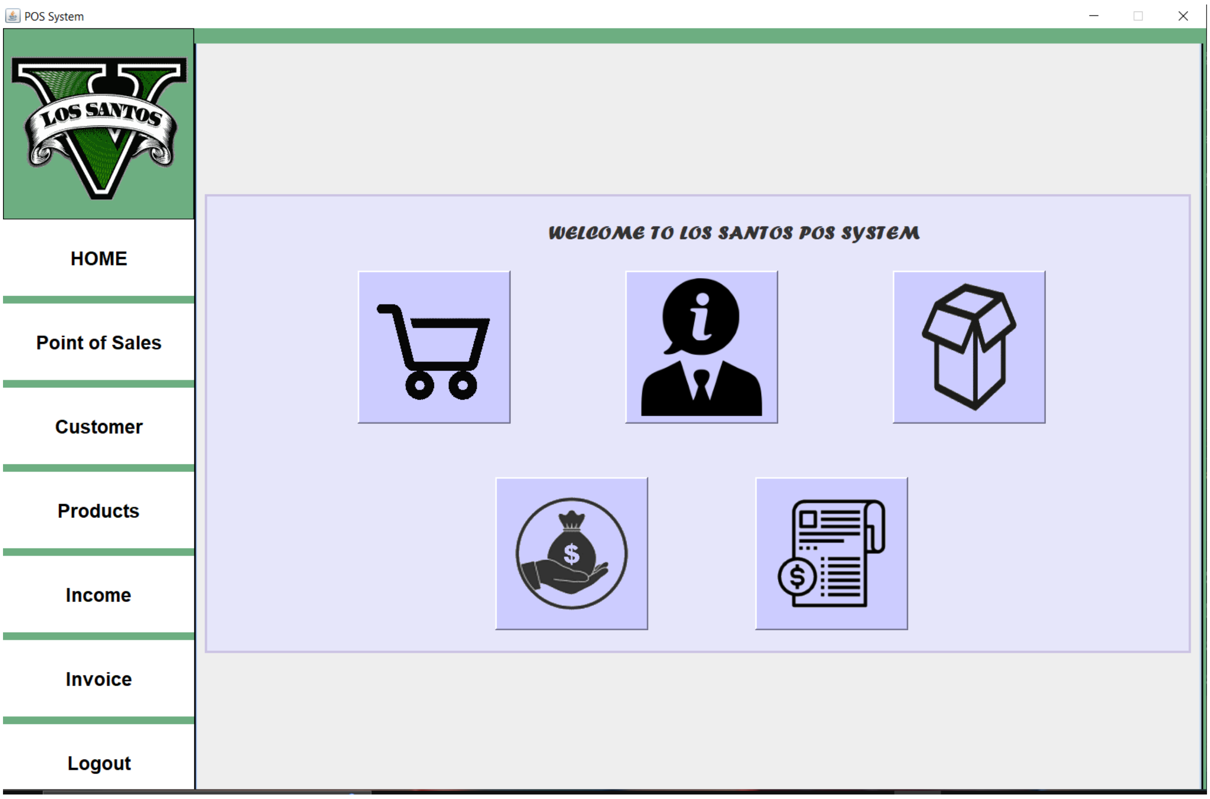
POS System

Username: weiming

Password: •••••••

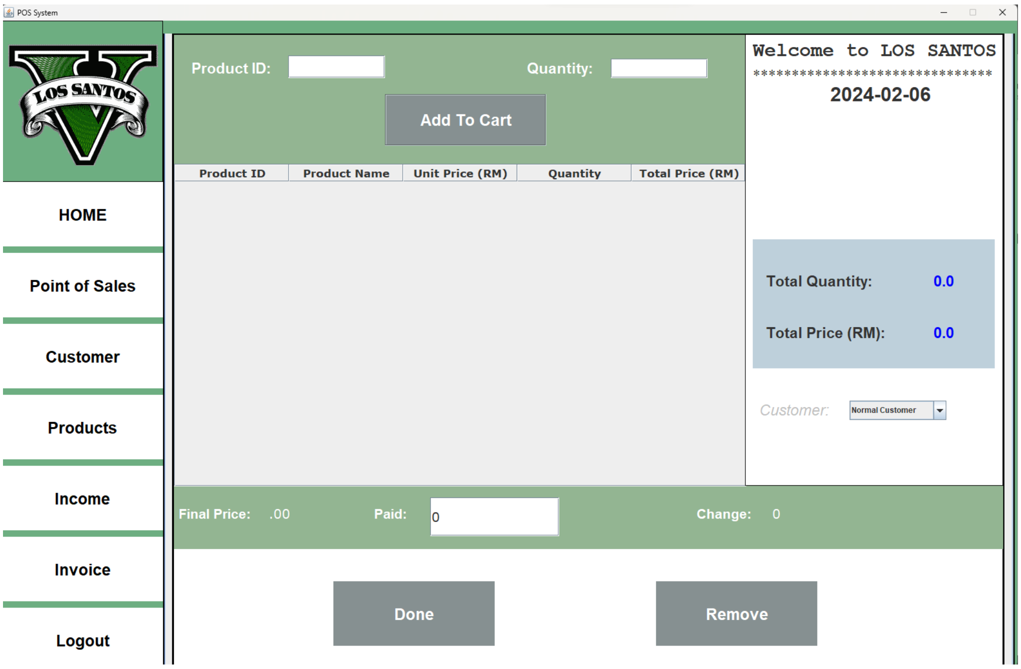
Home Page

This is the Home page of our POS management system. User can click on the button on the main site or click on the tab on the side bar to navigate to desired page.



POS Page

Users will be directed to POS page when they clicked the POS button on home page or Point of Sales button on the side tab.



Product ID:

Quantity:

Product ID	Product Name	Unit Price (RM)	Quantity	Total Price (RM)

```
Welcome to LOS SANTOS
******************************
2024-02-06
```

Total Quantity: 0.0

Total Price (RM): 0.0

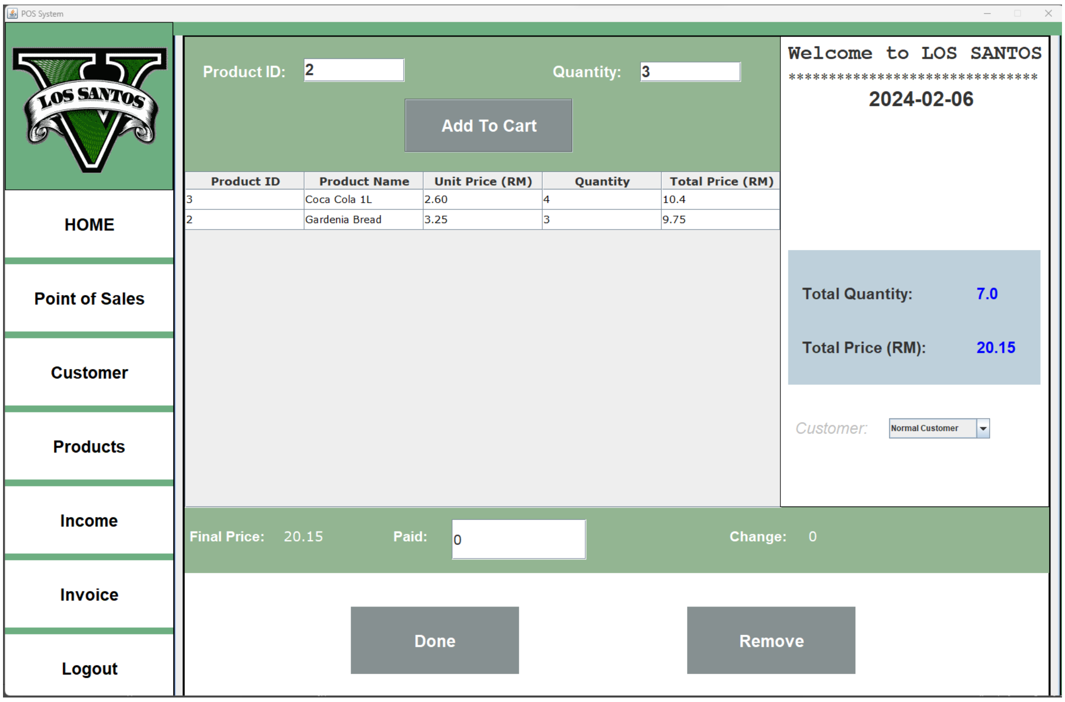
Customer: Normal Customer

Final Price: .00

Paid: 0

Change: 0

User is required to key in the product ID and quantity before adding the item into the cart by clicking Add To Cart button.



Product ID: 2 Quantity: 3

Product ID	Product Name	Unit Price (RM)	Quantity	Total Price (RM)
3	Coca Cola 1L	2.60	4	10.4
2	Gardenia Bread	3.25	3	9.75

Welcome to LOS SANTOS
****
2024-02-06

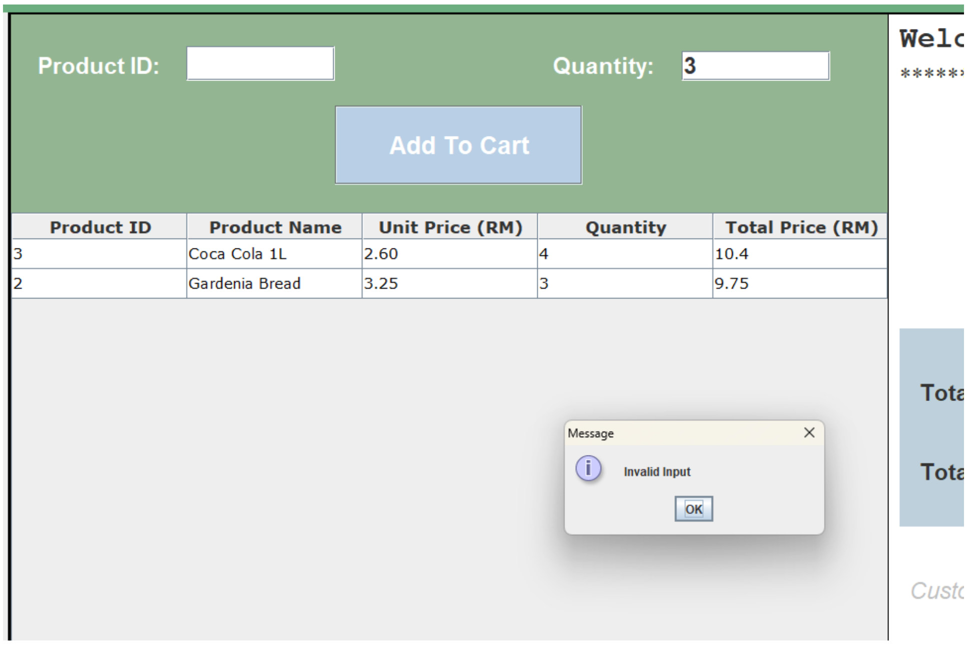
Total Quantity: 7.0

Total Price (RM): 20.15

Customer: Normal Customer

Final Price: 20.15 Paid: 0 Change: 0

The system will show an error message if the Product ID or Quantity is not entered.

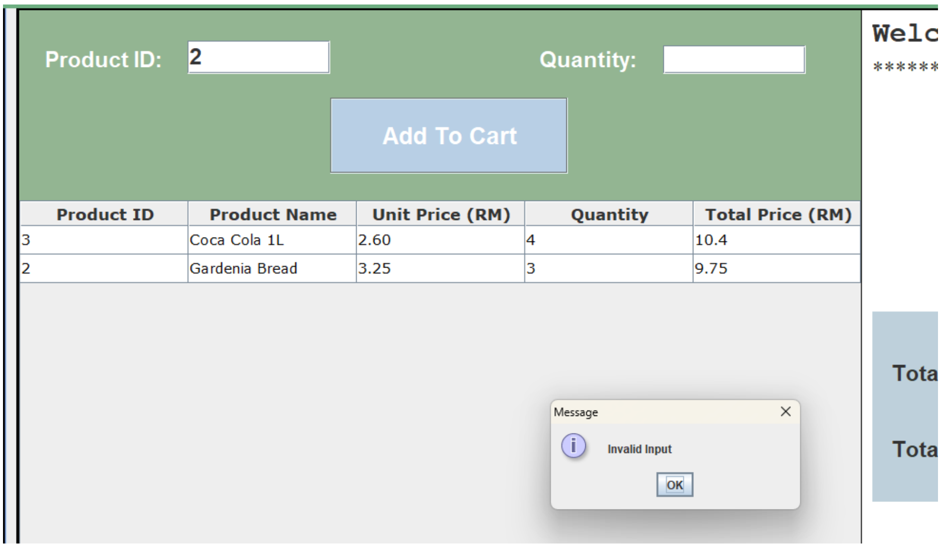


Product ID: Quantity: 3

Product ID	Product Name	Unit Price (RM)	Quantity	Total Price (RM)
3	Coca Cola 1L	2.60	4	10.4
2	Gardenia Bread	3.25	3	9.75

Message

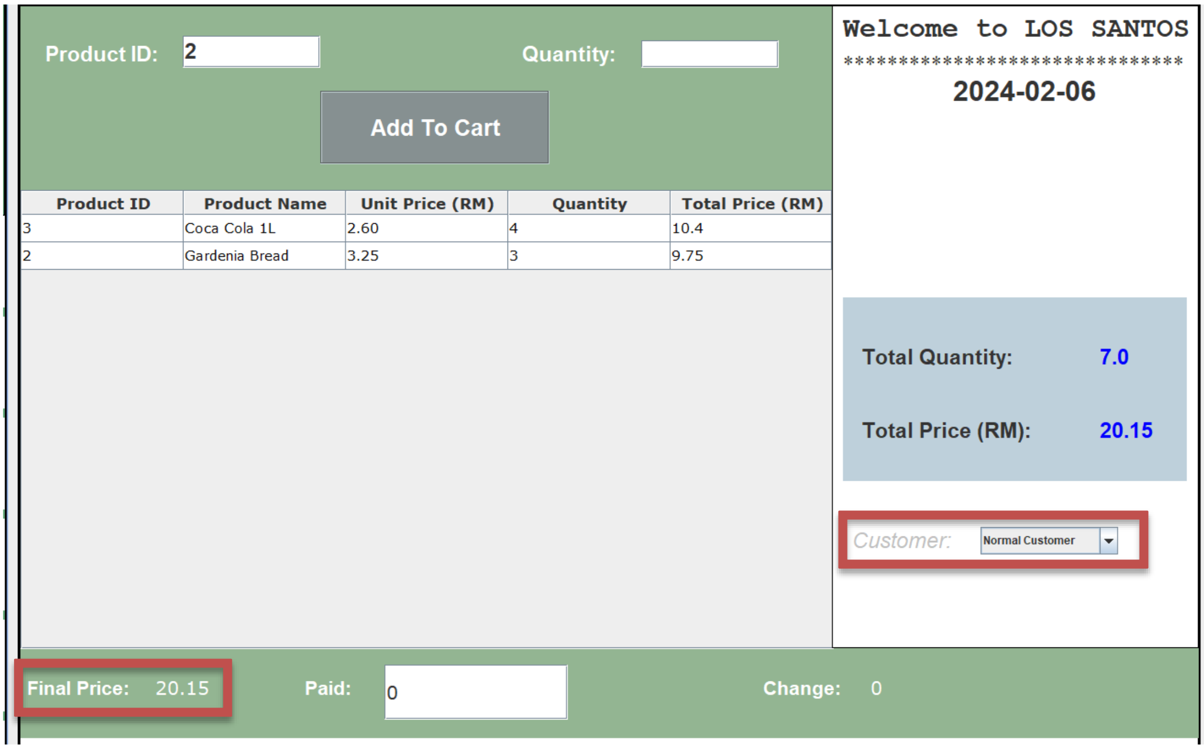
Invalid Input



Product ID	Product Name	Unit Price (RM)	Quantity	Total Price (RM)
3	Coca Cola 1L	2.60	4	10.4
2	Gardenia Bread	3.25	3	9.75

Invalid Input

The right bar will show the quantity of items in the cart and show the total price. In this tab, users can choose whether the customer is a member or not. A member will receive a 7% discount.



Product ID: 2

Quantity:

Product ID	Product Name	Unit Price (RM)	Quantity	Total Price (RM)
3	Coca Cola 1L	2.60	4	10.4
2	Gardenia Bread	3.25	3	9.75

Welcome to LOS SANTOS

****

2024-02-06

Total Quantity: 7.0

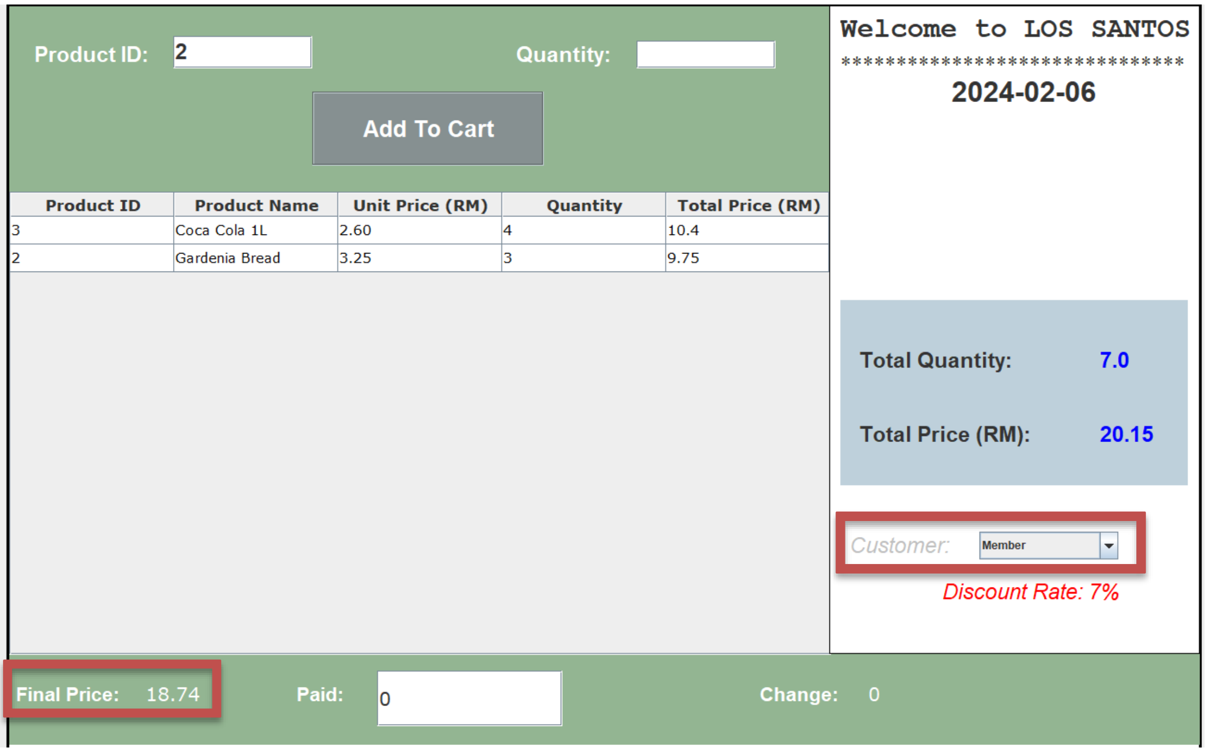
Total Price (RM): 20.15

Customer: Normal Customer

Final Price: 20.15

Paid: 0

Change: 0



Product ID: 2

Quantity:

Product ID	Product Name	Unit Price (RM)	Quantity	Total Price (RM)
3	Coca Cola 1L	2.60	4	10.4
2	Gardenia Bread	3.25	3	9.75

Welcome to LOS SANTOS

****

2024-02-06

Total Quantity: 7.0

Total Price (RM): 20.15

Customer: Member

Discount Rate: 7%

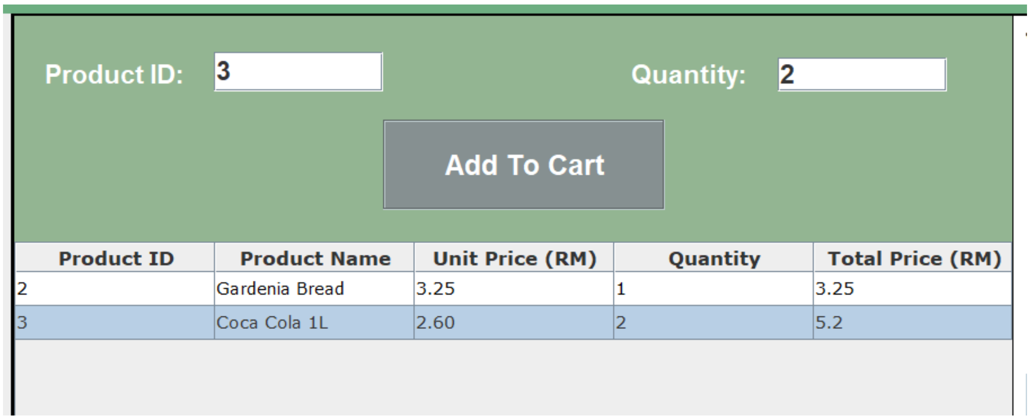
Final Price: 18.74

Paid: 0

Change: 0

*The final price is the price after the discount.

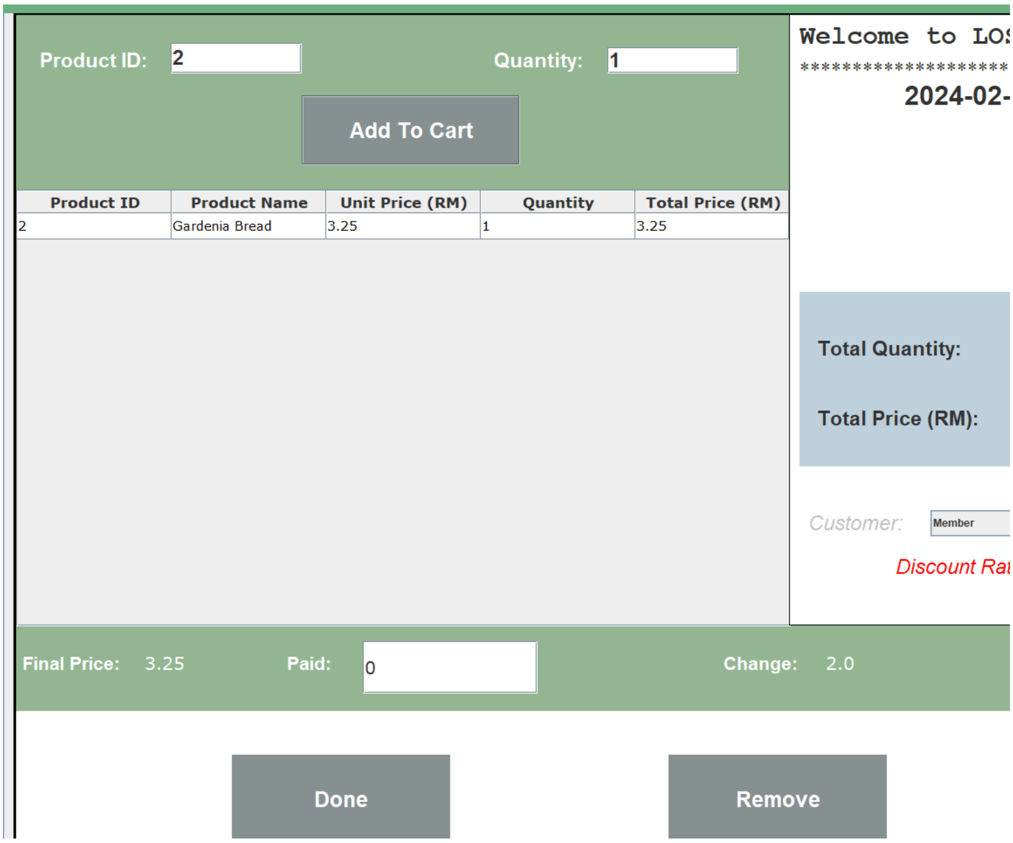
User can remove the item in the cart by clicking the item in the table and click Remove button.



Product ID: 3

Quantity: 2

Product ID	Product Name	Unit Price (RM)	Quantity	Total Price (RM)
2	Gardenia Bread	3.25	1	3.25
3	Coca Cola 1L	2.60	2	5.2



Product ID: 2

Quantity: 1

Product ID	Product Name	Unit Price (RM)	Quantity	Total Price (RM)
2	Gardenia Bread	3.25	1	3.25

```
Welcome to LOS
********************
        2024-02-
```

Total Quantity:

Total Price (RM):

Customer: Member

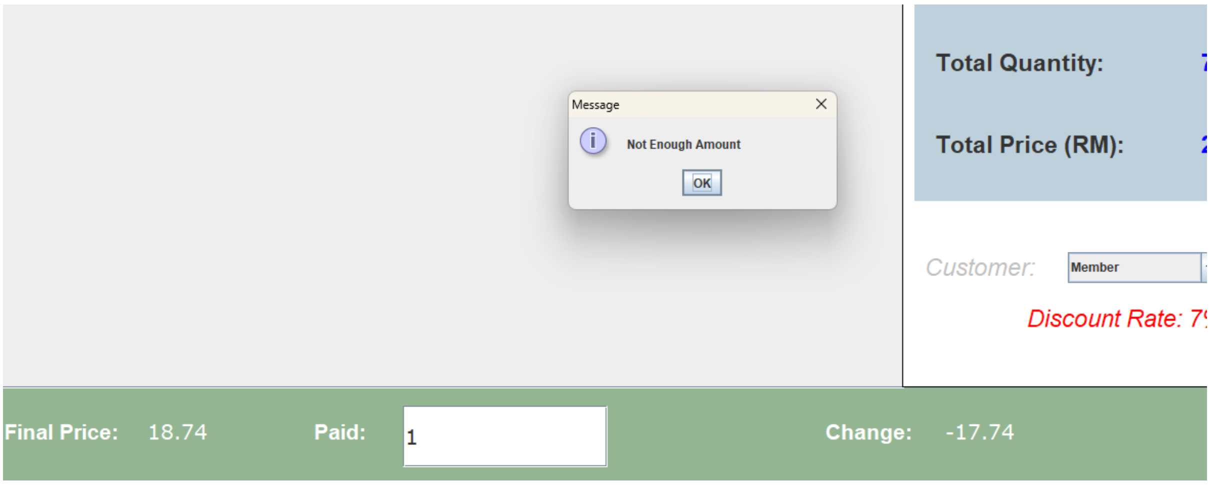
Discount Rat

Final Price: 3.25

Paid: 0

Change: 2.0

Users had to enter the amount of money that is greater or equal to the final price to complete the payment process. Users had to click the Done button to complete the payment process.



Message

Not Enough Amount

OK

Total Quantity:

Total Price (RM):

Customer: Member

Discount Rate: 7

Final Price: 18.74

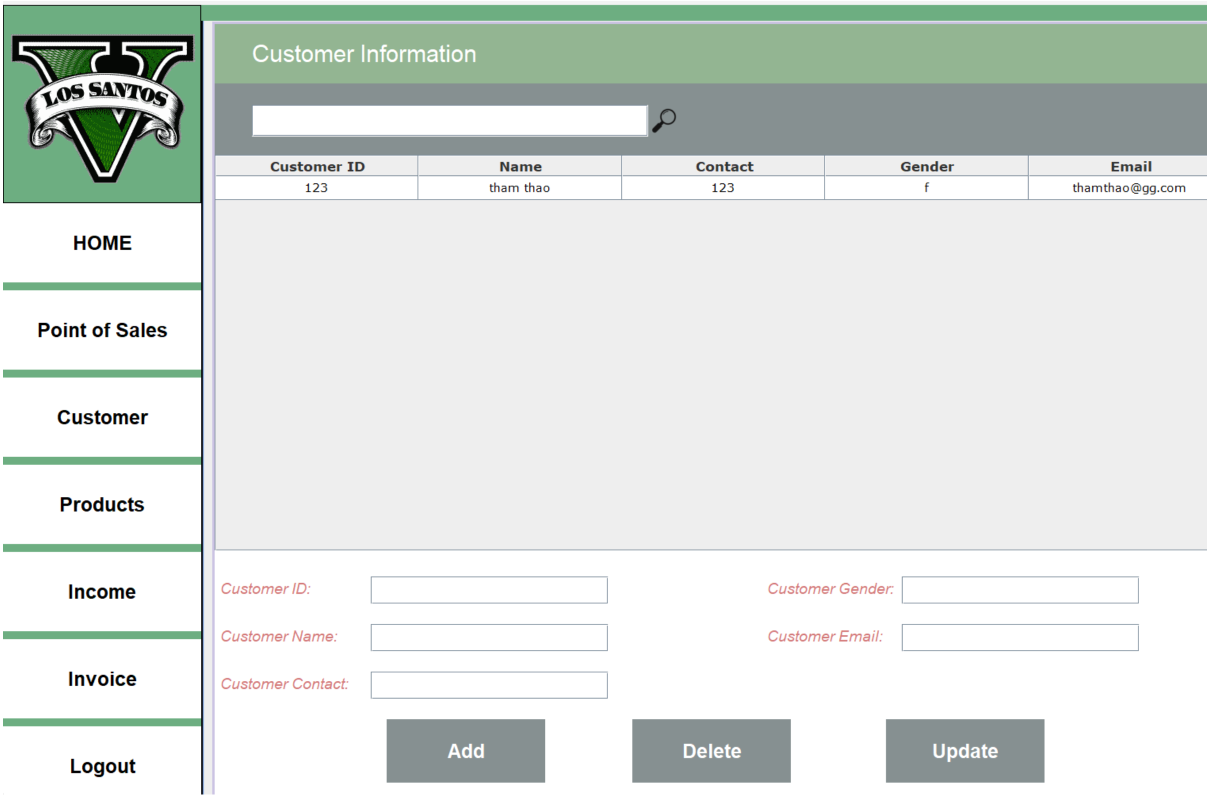
Paid: 1

Change: -17.74

*An error message will pop out if the entered amount is less than final price.

Customer Page

Users can go to the Customer Page to view or edit the customer information.



Customer Information

Customer ID	Name	Contact	Gender	Email
123	tham thao	123	f	thamthao@gg.com

Customer ID:

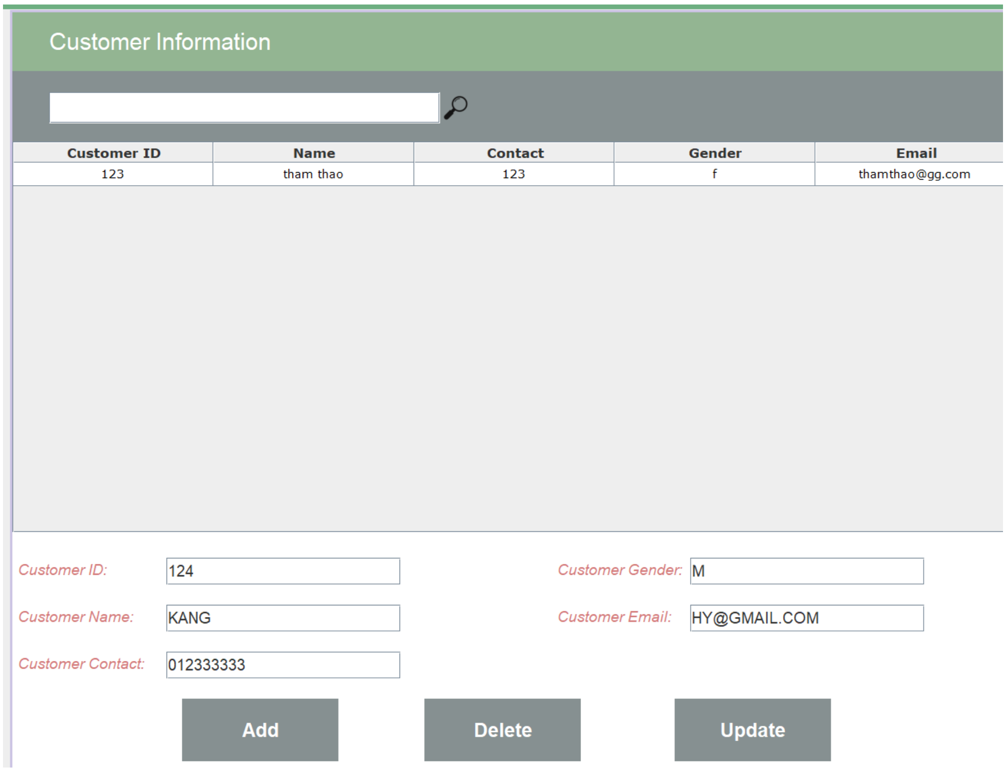
Customer Name:

Customer Contact:

Customer Gender:

Customer Email:

User can input the customer information of prepared.



Customer ID	Name	Contact	Gender	Email
123	tham thao	123	f	thamthao@gg.com

Customer ID: 124

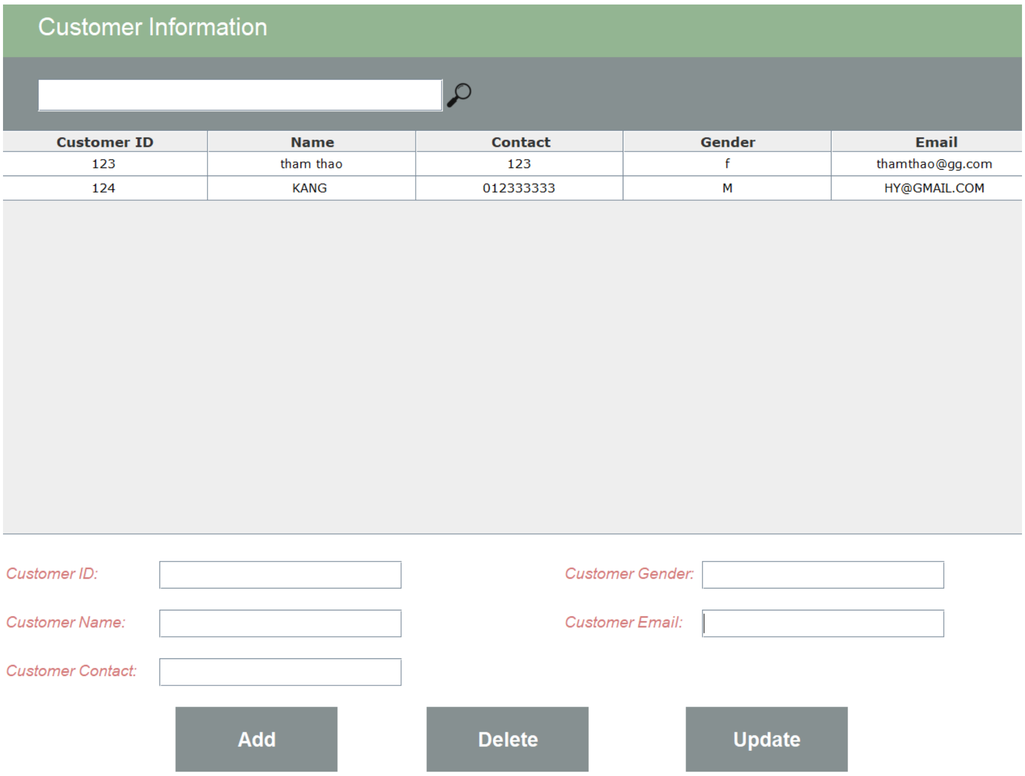
Customer Name: KANG

Customer Contact: 012333333

Customer Gender: M

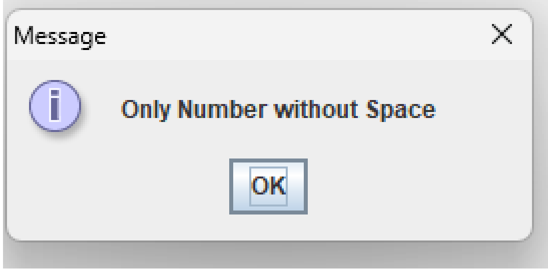
Customer Email: HY@GMAIL.COM

After the user fill in all blanks, press the add button to add customer information.



Customer ID	Name	Contact	Gender	Email
123	tham thao	123	f	thamthao@gg.com
124	KANG	012333333	M	HY@GMAIL.COM

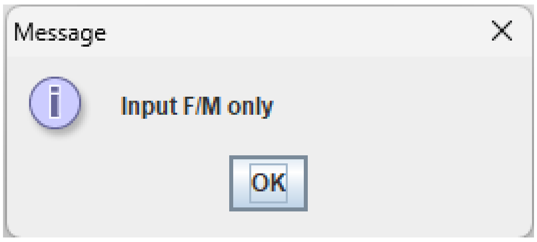
For the customer id and contact only can input number



Message

Only Number without Space

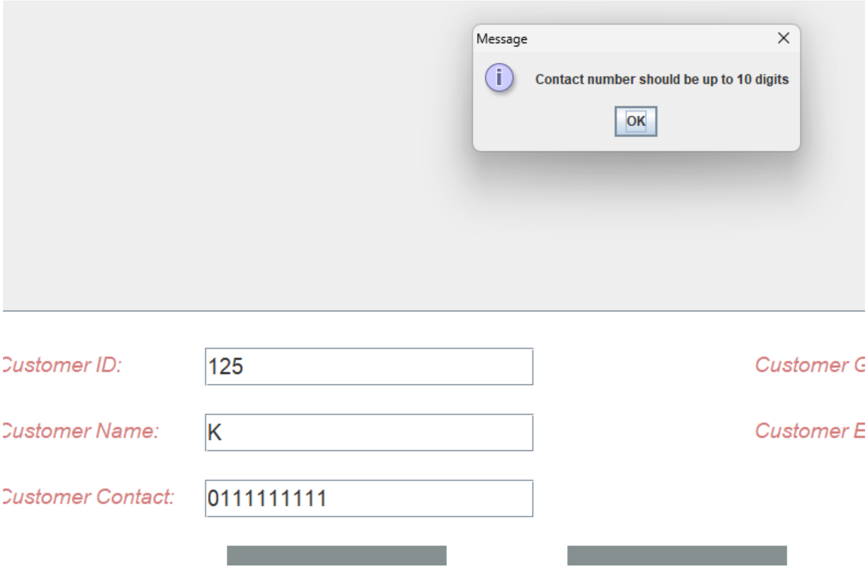
For the customer gender only can input f/m



Message

Input F/M only

For the customer contact should be up to 10 digits

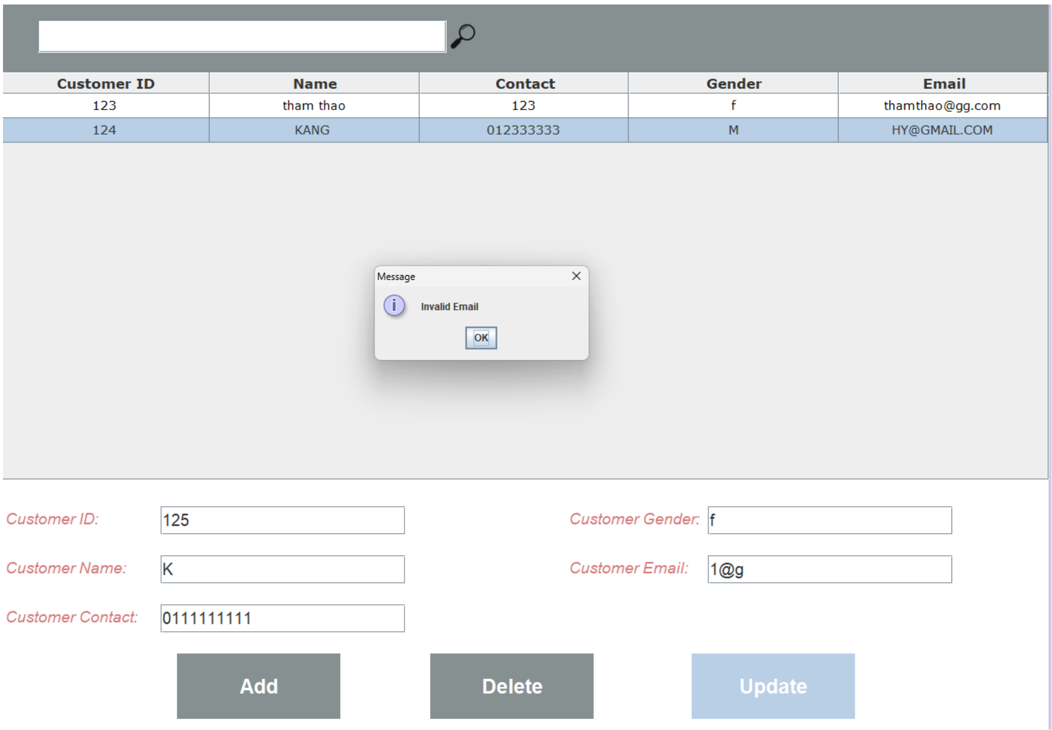


Message

Contact number should be up to 10 digits

Customer ID:	125
Customer Name:	K
Customer Contact:	0111111111

For the customer email should be "xxx@xxx.com", must include "@" and ".com"



Customer ID	Name	Contact	Gender	Email
123	tham thao	123	f	thamthao@gg.com
124	KANG	012333333	M	HY@GMAIL.COM

Message

Invalid Email

OK

Customer ID: 125

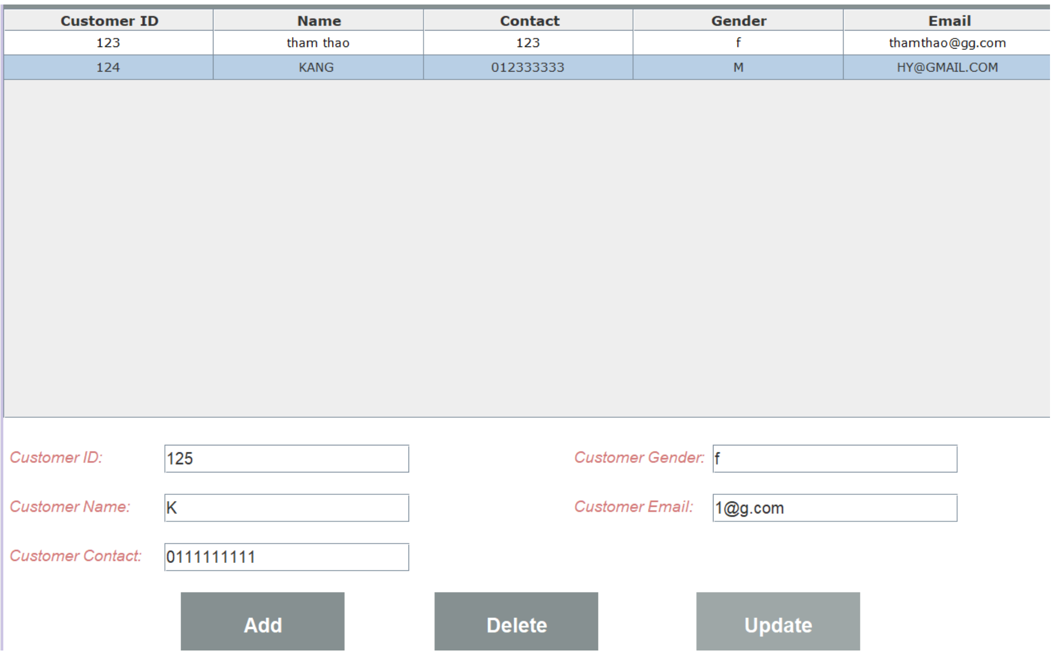
Customer Name: K

Customer Contact: 0111111111

Customer Gender: f

Customer Email: 1@g

Users can use the update button to update selected customer information.



Customer ID	Name	Contact	Gender	Email
123	tham thao	123	f	thamthao@gg.com
124	KANG	012333333	M	HY@GMAIL.COM

Customer ID: 125

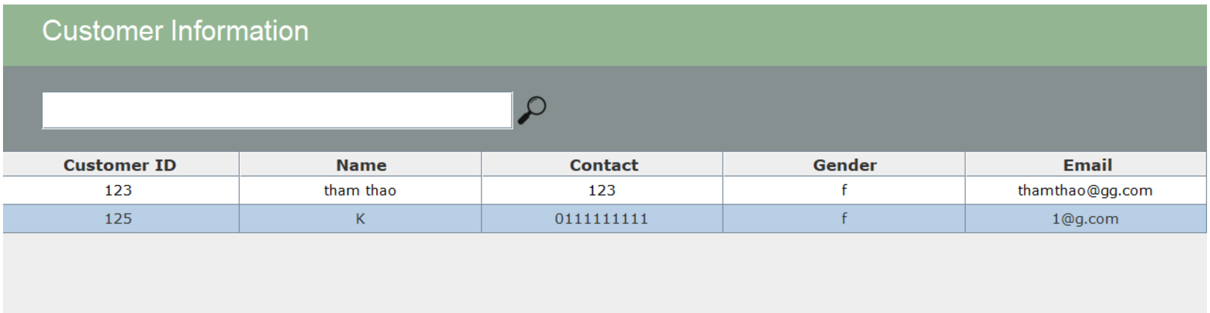
Customer Name: K

Customer Contact: 0111111111

Customer Gender: f

Customer Email: 1@g.com

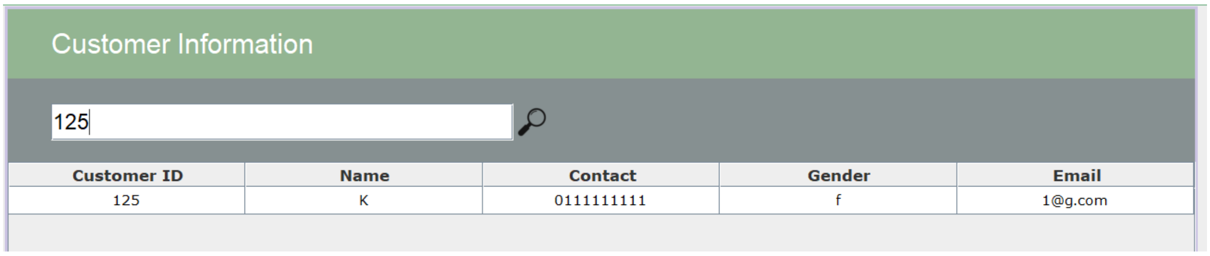
After user press the update button



Customer Information

Customer ID	Name	Contact	Gender	Email
123	tham thao	123	f	thamthao@gg.com
125	K	0111111111	f	1@g.com

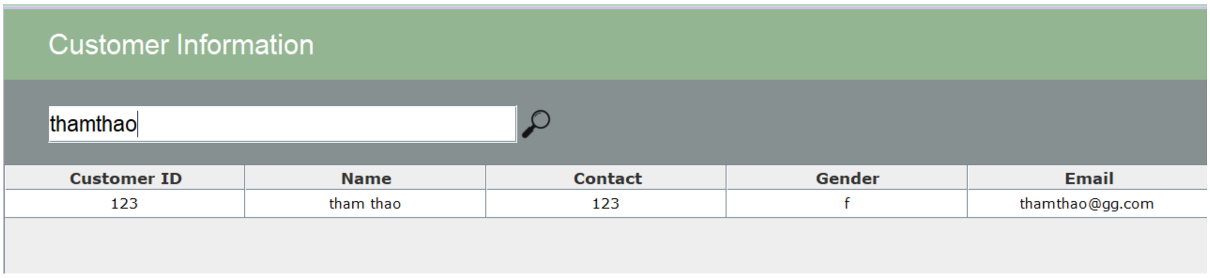
Users can use the search bar to search the customer information they want to find by input the customer information such as id, name, contact and so on.



Customer Information

125

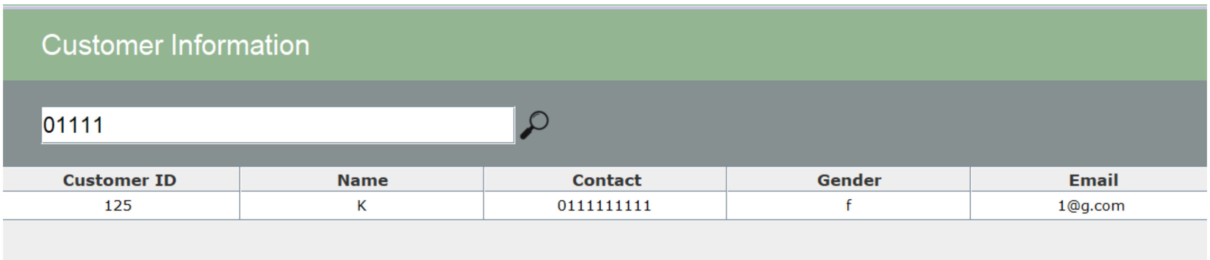
Customer ID	Name	Contact	Gender	Email
125	K	0111111111	f	1@g.com



Customer Information

thamthao

Customer ID	Name	Contact	Gender	Email
123	tham thao	123	f	thamthao@gg.com

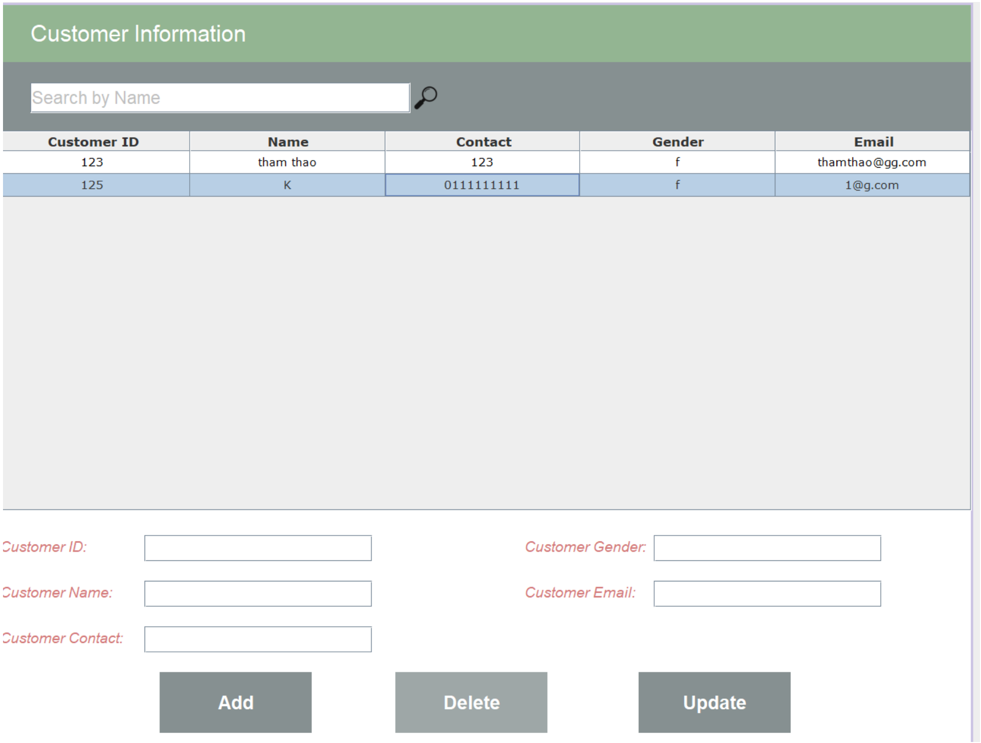


Customer Information

01111

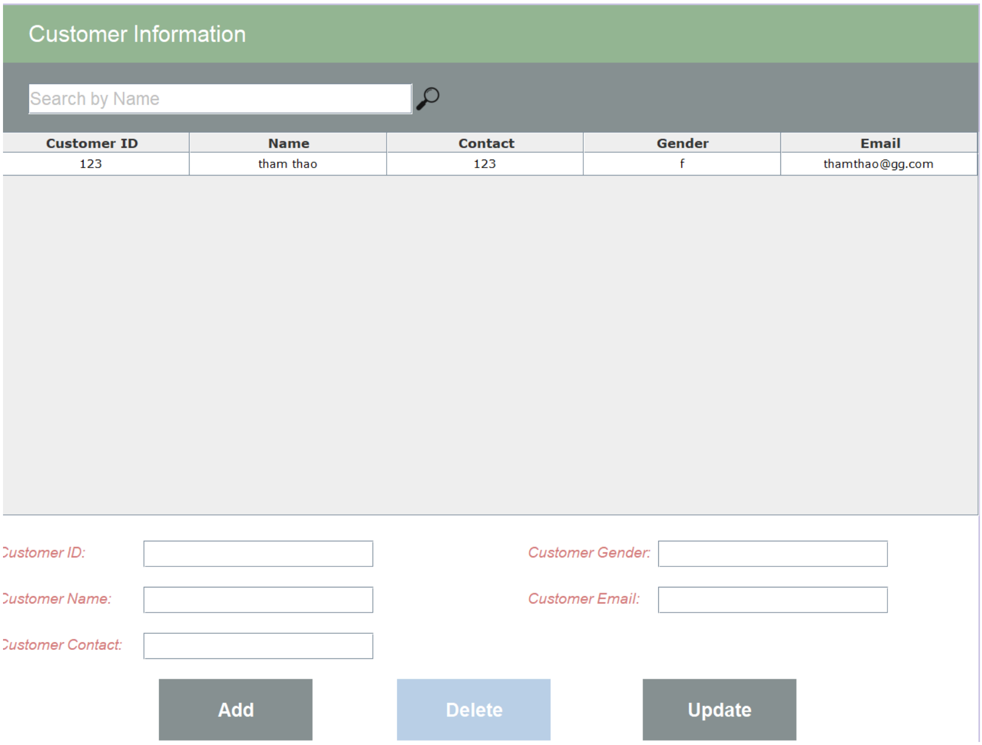
Customer ID	Name	Contact	Gender	Email
125	K	0111111111	f	1@g.com

Users can use the delete button to delete the customer information they choose.



Customer ID	Name	Contact	Gender	Email
123	tham thao	123	f	thamthao@gg.com
125	K	0111111111	f	1@g.com

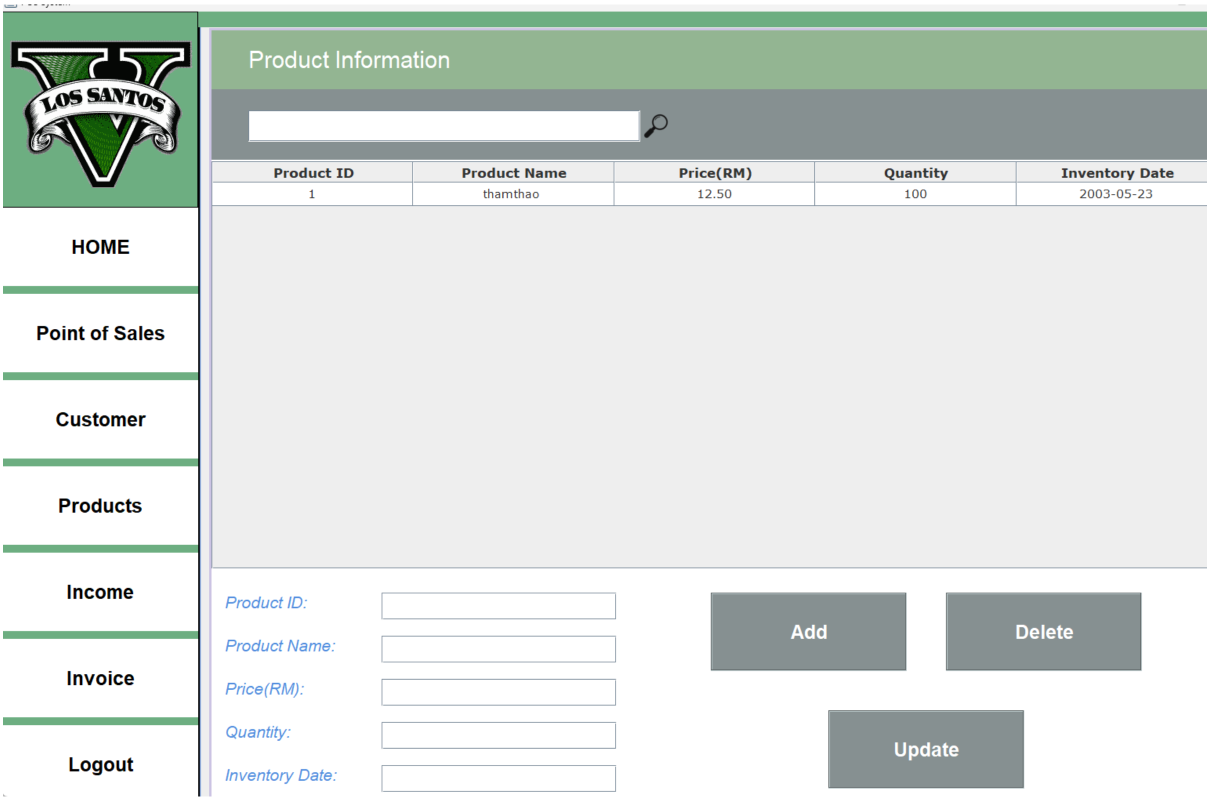
After user press the delete button



Customer ID	Name	Contact	Gender	Email
123	tham thao	123	f	thamthao@gg.com

Product Page

Users can go to the Product Page to view or edit the product information.



Product Information

Product ID	Product Name	Price(RM)	Quantity	Inventory Date
1	thamthao	12.50	100	2003-05-23

Product ID:

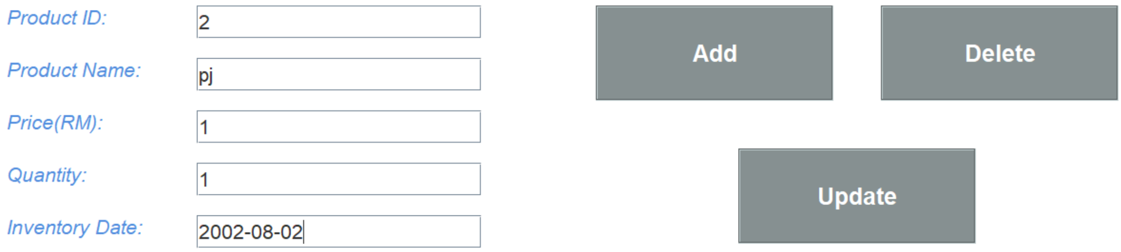
Product Name:

Price(RM):

Quantity:

Inventory Date:

Users can input the product information to the blanks.



Product ID: 2

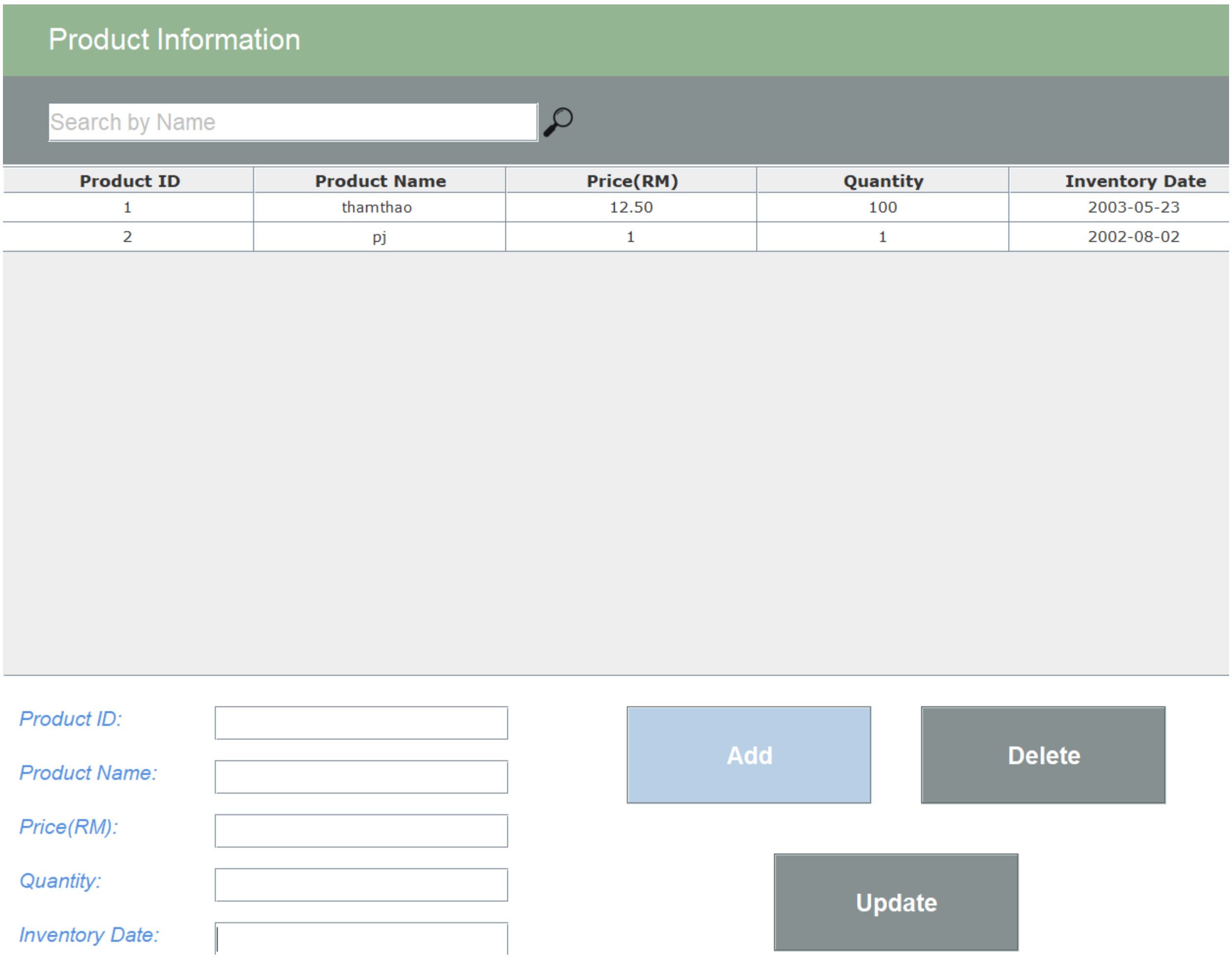
Product Name: pj

Price(RM): 1

Quantity: 1

Inventory Date: 2002-08-02

After the user fills in all blanks, press the add button to add customer information.



Product ID	Product Name	Price(RM)	Quantity	Inventory Date
1	thamthao	12.50	100	2003-05-23
2	pj	1	1	2002-08-02

Product ID:

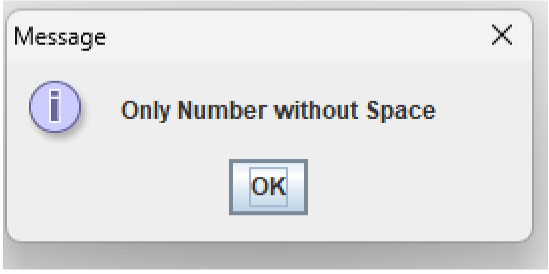
Product Name:

Price(RM):

Quantity:

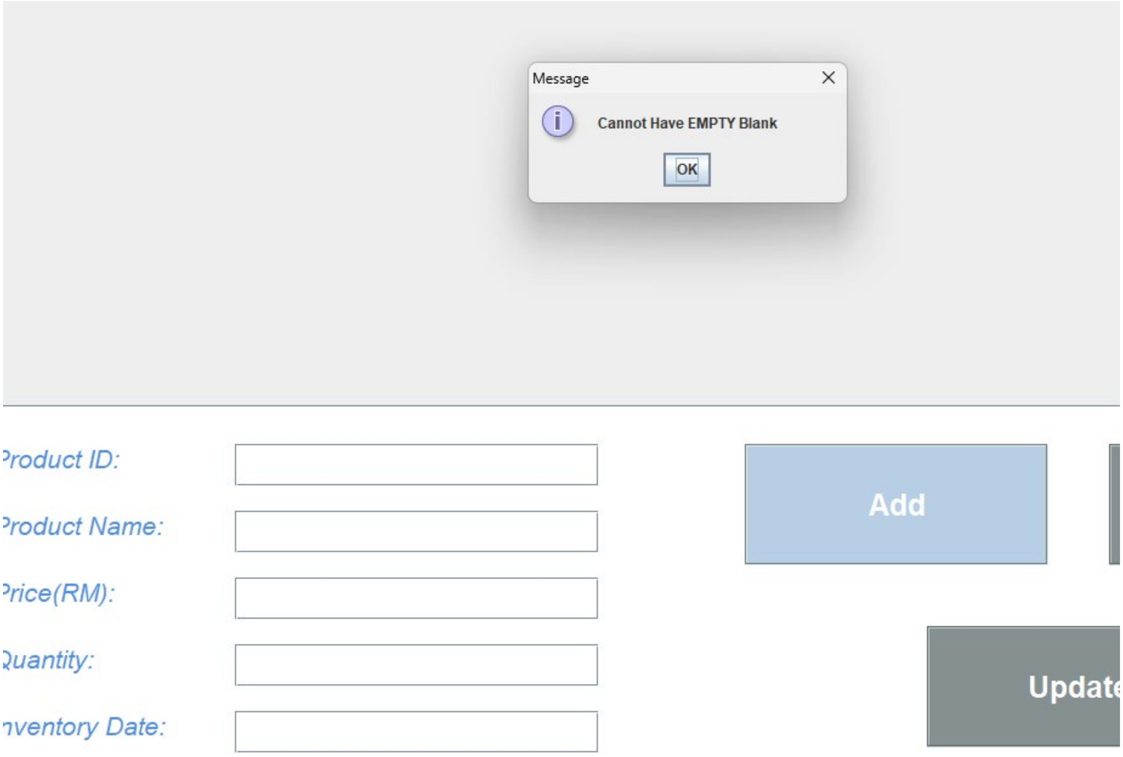
Inventory Date:

For the product id, price and quantity only can input number.



Only Number without Space

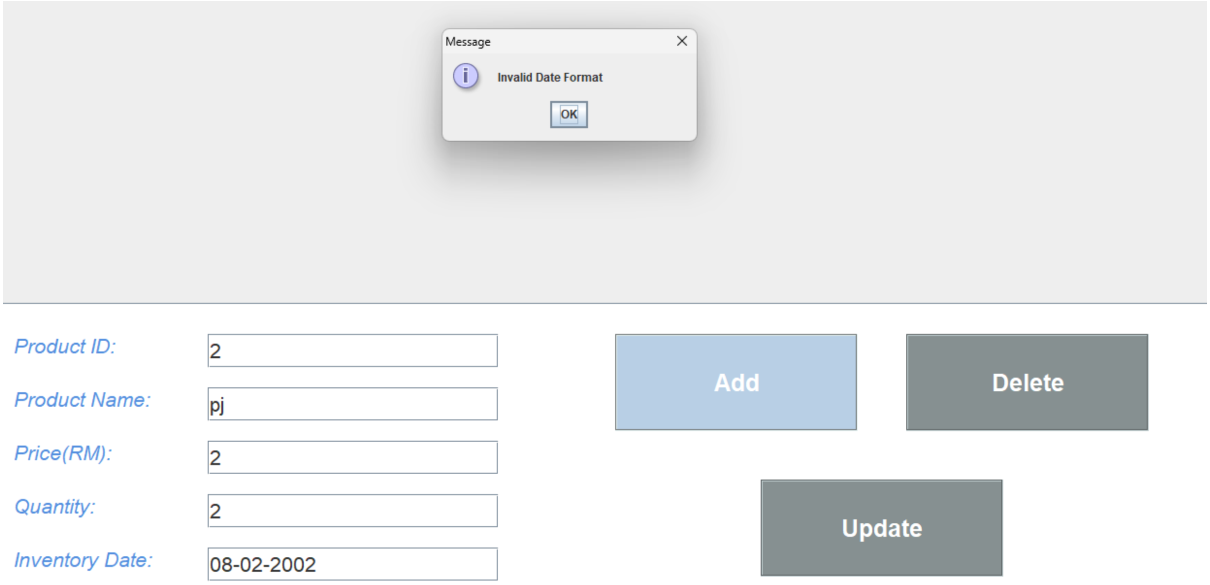
User must fill the all blanks.



Message

Cannot Have EMPTY Blank

For the inventory date must input “YYYY-MM-DD”

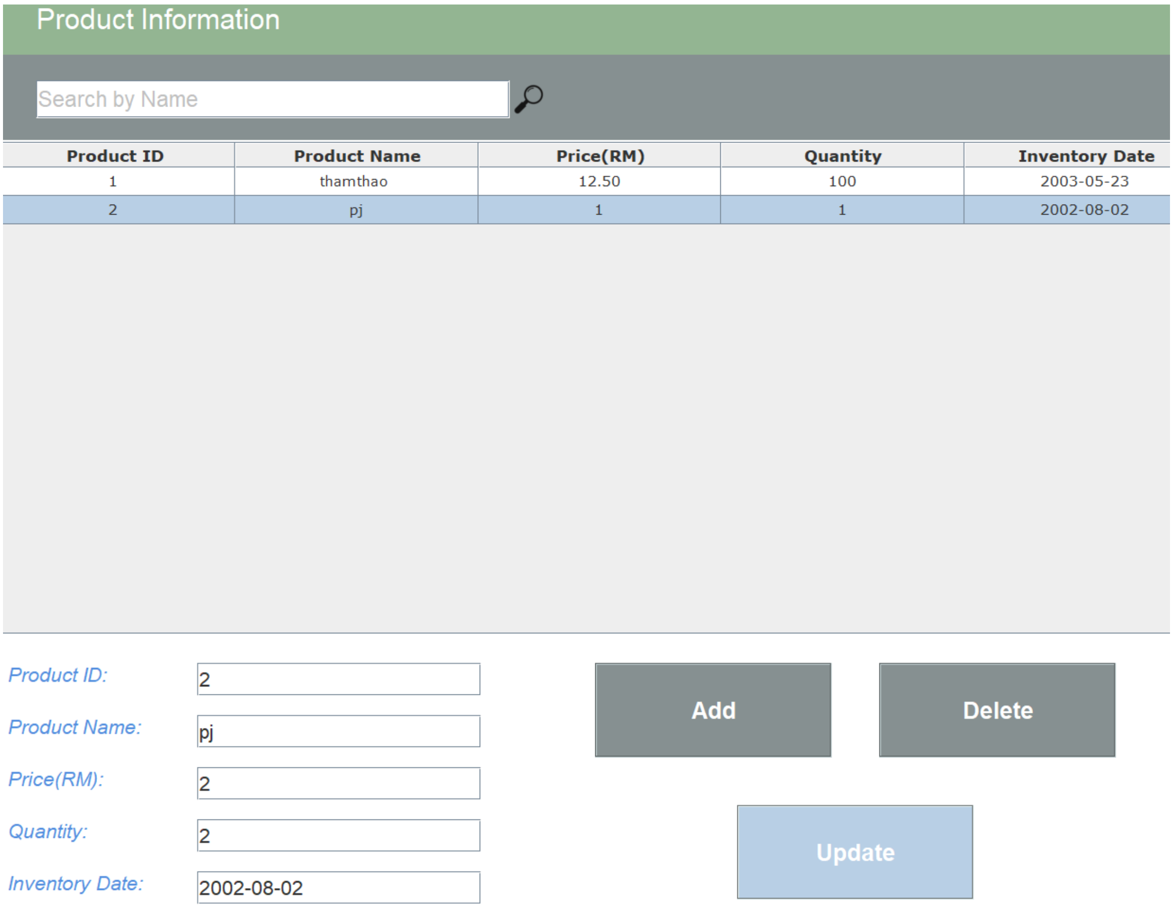


Message

Invalid Date Format

Product ID:	2
Product Name:	pj
Price(RM):	2
Quantity:	2
Inventory Date:	08-02-2002

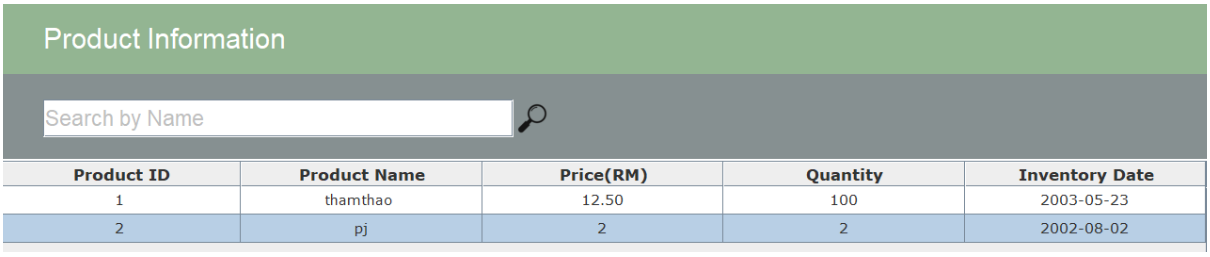
Users can use the update button to update selected product information.



Product ID	Product Name	Price(RM)	Quantity	Inventory Date
1	thamthao	12.50	100	2003-05-23
2	pj	1	1	2002-08-02

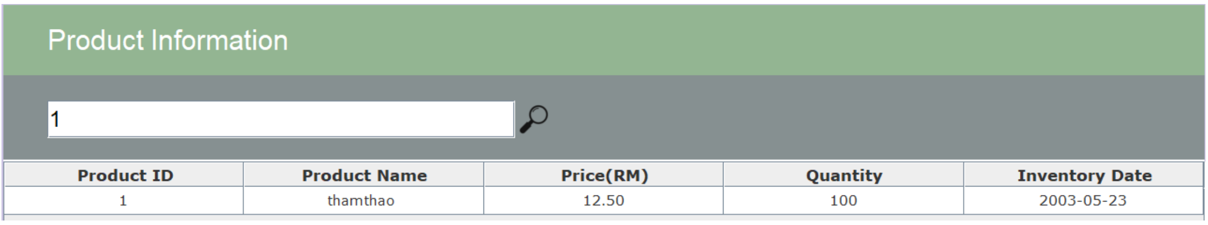
Product ID:	2
Product Name:	pj
Price(RM):	2
Quantity:	2
Inventory Date:	2002-08-02

After user press the update button



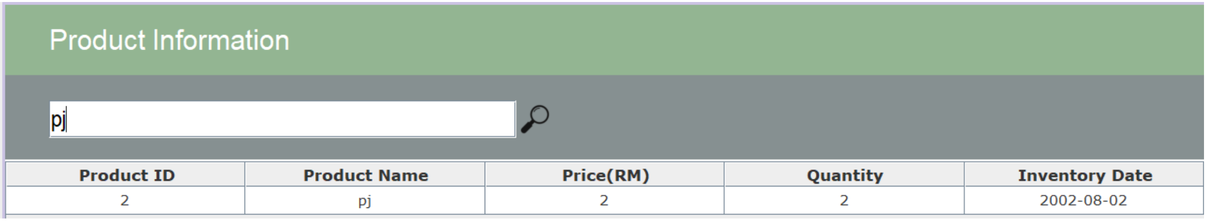
Product ID	Product Name	Price(RM)	Quantity	Inventory Date
1	thamthao	12.50	100	2003-05-23
2	pj	2	2	2002-08-02

Users can use the search bar to search for the product information they want to find by input the customer information such as id, name, date and so on.



1

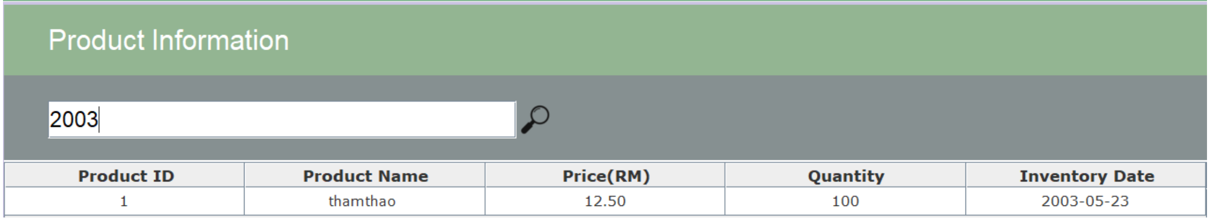
Product ID	Product Name	Price(RM)	Quantity	Inventory Date
1	thamthao	12.50	100	2003-05-23



Product Information

pj

Product ID	Product Name	Price(RM)	Quantity	Inventory Date
2	pj	2	2	2002-08-02

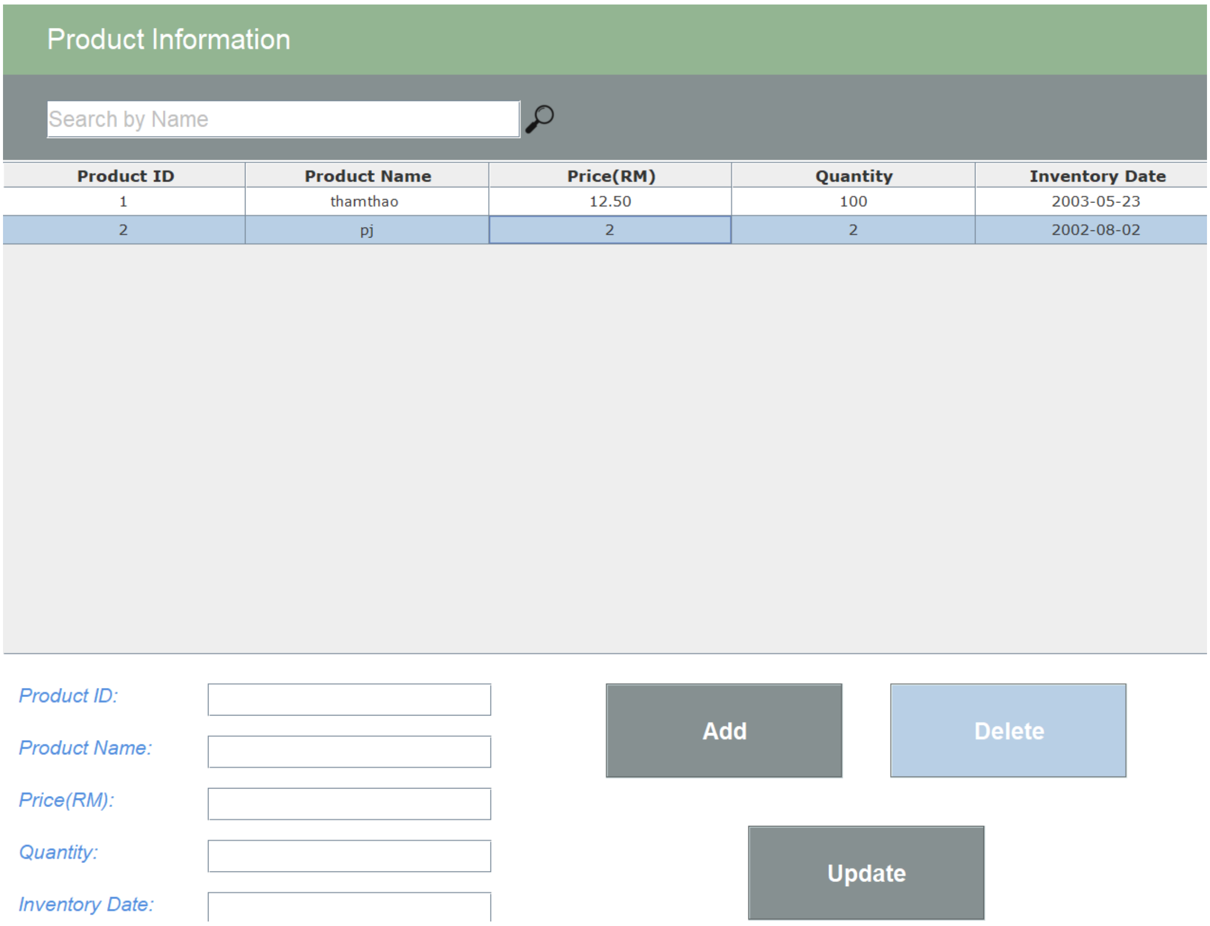


Product Information

2003

Product ID	Product Name	Price(RM)	Quantity	Inventory Date
1	thamthao	12.50	100	2003-05-23

Users can use the delete button to delete the product information they choose.



Product Information

Search by Name

Product ID	Product Name	Price(RM)	Quantity	Inventory Date
1	thamthao	12.50	100	2003-05-23
2	pj	2	2	2002-08-02

Product ID:

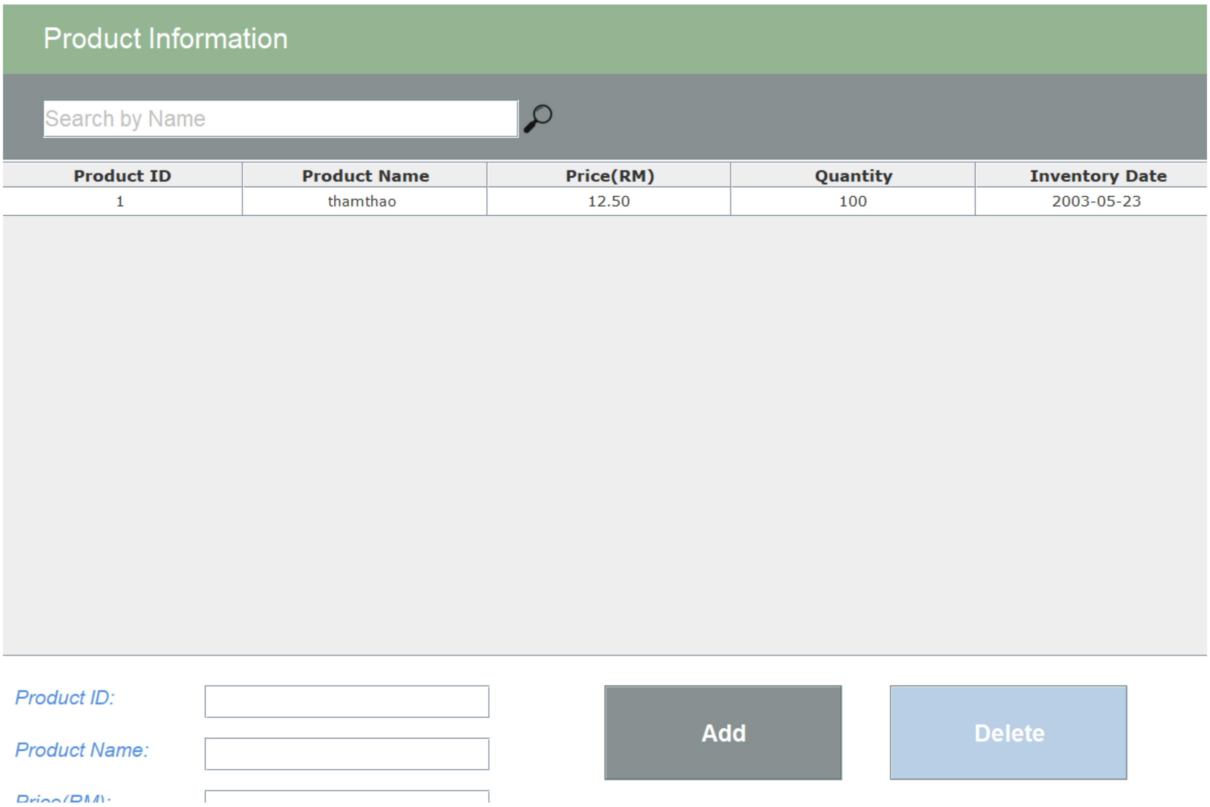
Product Name:

Price(RM):

Quantity:

Inventory Date:

After user press the delete button



Product Information

Search by Name

Product ID	Product Name	Price(RM)	Quantity	Inventory Date
1	thamthao	12.50	100	2003-05-23

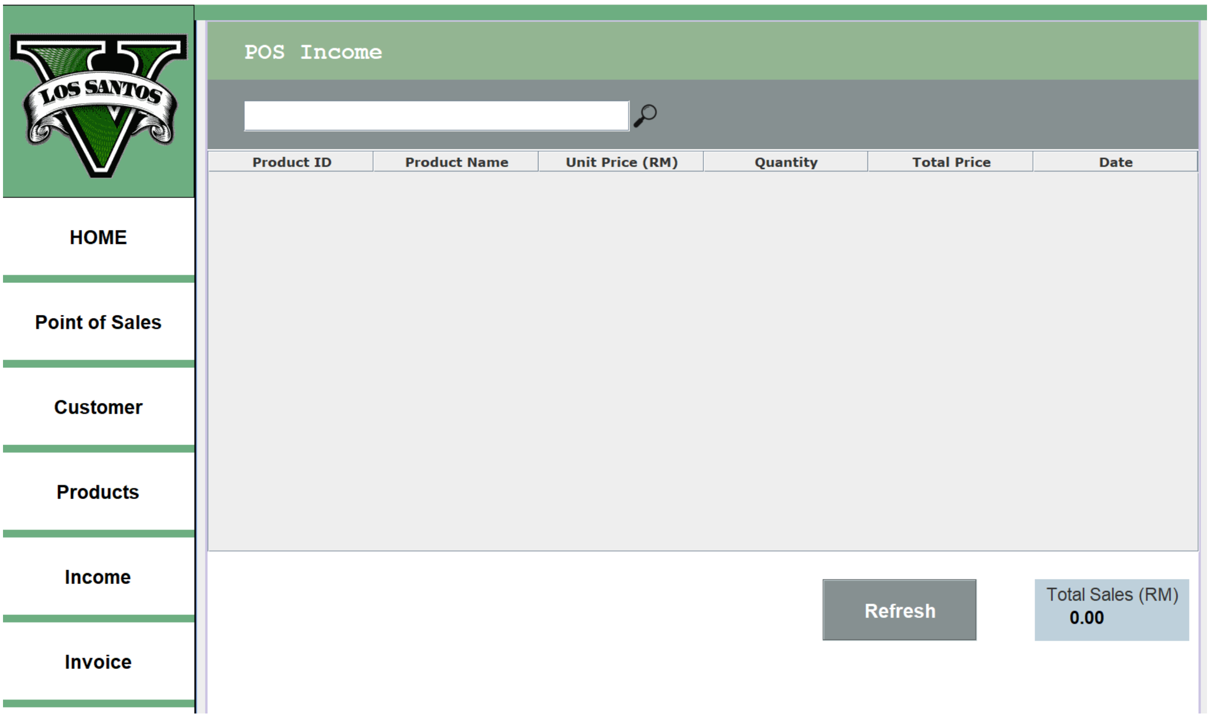
Product ID:

Product Name:

Price(RM):

Income Page

Users can direct to Income Page to view the income of the POS by clicking Income button in home page or Income button on the side bar.

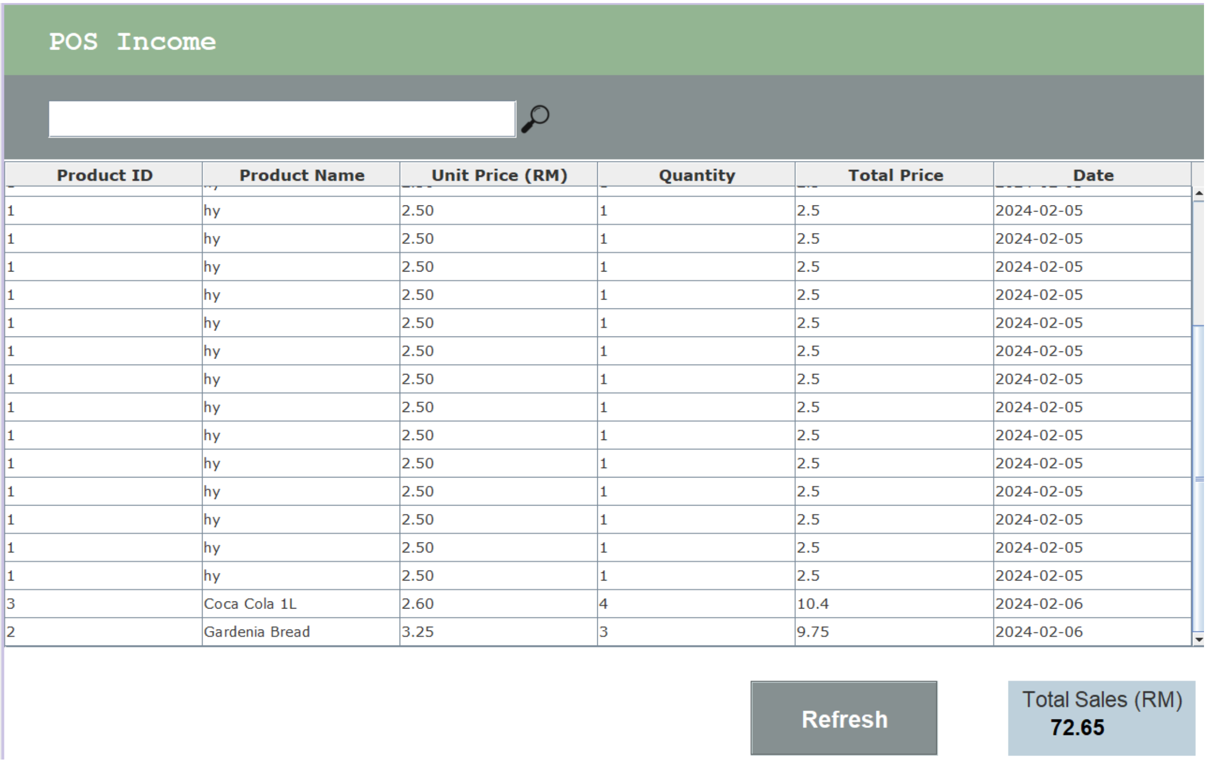


POS Income

Product ID	Product Name	Unit Price (RM)	Quantity	Total Price	Date

Total Sales (RM)
0.00

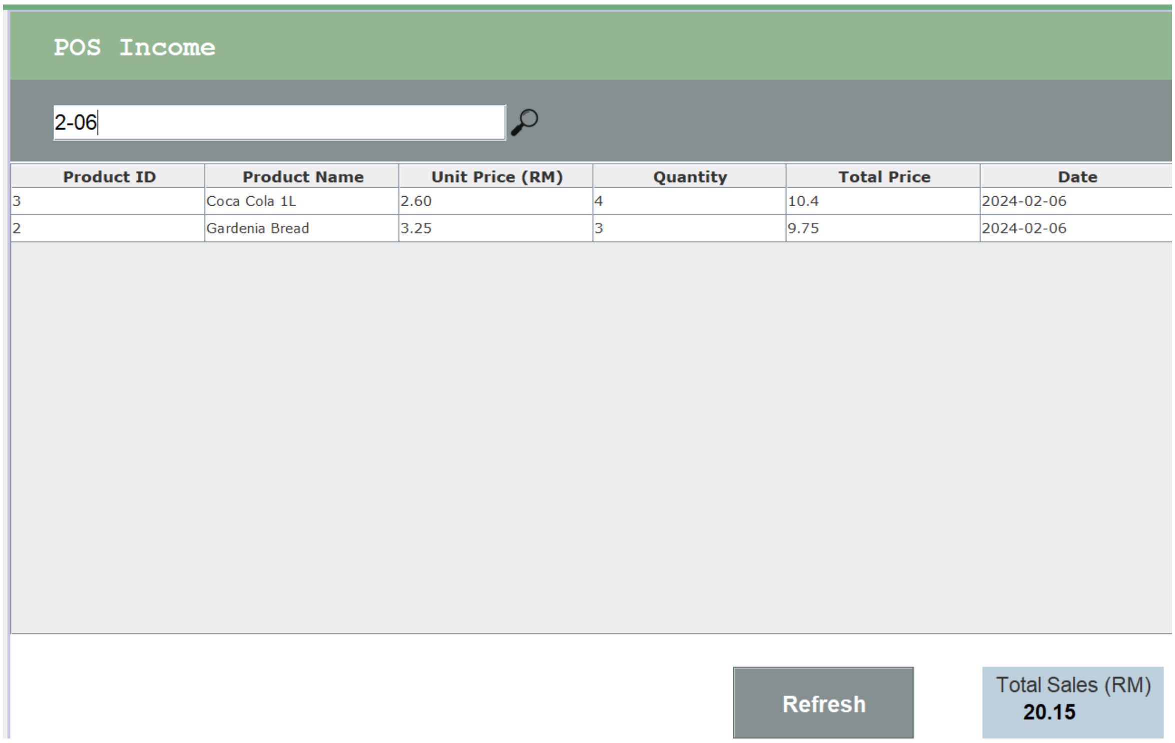
Users can click on the Refresh button to show the details of income.



Product ID	Product Name	Unit Price (RM)	Quantity	Total Price	Date
1	hy	2.50	1	2.5	2024-02-05
1	hy	2.50	1	2.5	2024-02-05
1	hy	2.50	1	2.5	2024-02-05
1	hy	2.50	1	2.5	2024-02-05
1	hy	2.50	1	2.5	2024-02-05
1	hy	2.50	1	2.5	2024-02-05
1	hy	2.50	1	2.5	2024-02-05
1	hy	2.50	1	2.5	2024-02-05
1	hy	2.50	1	2.5	2024-02-05
1	hy	2.50	1	2.5	2024-02-05
1	hy	2.50	1	2.5	2024-02-05
1	hy	2.50	1	2.5	2024-02-05
1	hy	2.50	1	2.5	2024-02-05
1	hy	2.50	1	2.5	2024-02-05
3	Coca Cola 1L	2.60	4	10.4	2024-02-06
2	Gardenia Bread	3.25	3	9.75	2024-02-06

Total Sales (RM)
72.65

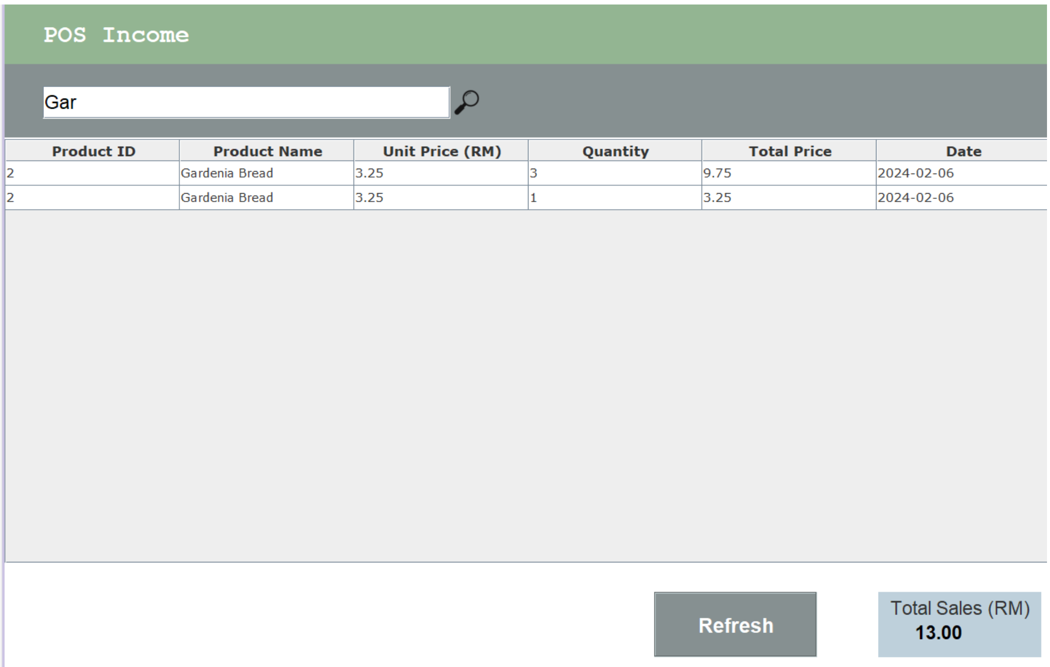
Users can select the date to view the income of the selected date on the search bar. Users also can view the income contributed by certain items.



2-06

Product ID	Product Name	Unit Price (RM)	Quantity	Total Price	Date
3	Coca Cola 1L	2.60	4	10.4	2024-02-06
2	Gardenia Bread	3.25	3	9.75	2024-02-06

Total Sales (RM)
20.15



POS Income

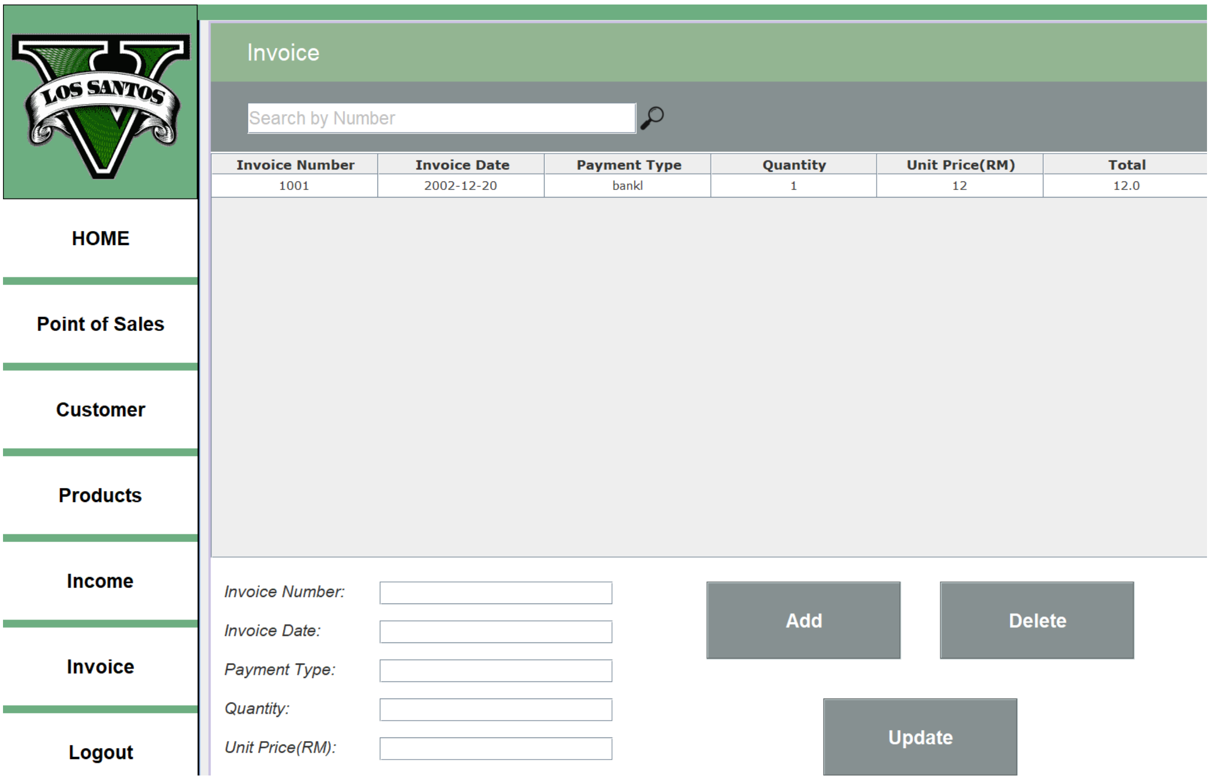
Gar

Product ID	Product Name	Unit Price (RM)	Quantity	Total Price	Date
2	Gardenia Bread	3.25	3	9.75	2024-02-06
2	Gardenia Bread	3.25	1	3.25	2024-02-06

Total Sales (RM)
13.00

Invoice Page

Users can go to the Invoice Page to view or edit the invoice.



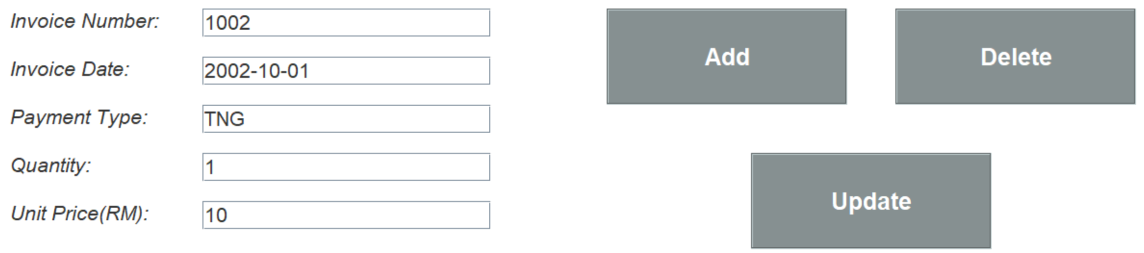
Invoice

Search by Number

Invoice Number	Invoice Date	Payment Type	Quantity	Unit Price(RM)	Total
1001	2002-12-20	bankl	1	12	12.0

Invoice Number:
Invoice Date:
Payment Type:
Quantity:
Unit Price(RM):

Users can input the invoice to the blanks.



Invoice Number: 1002

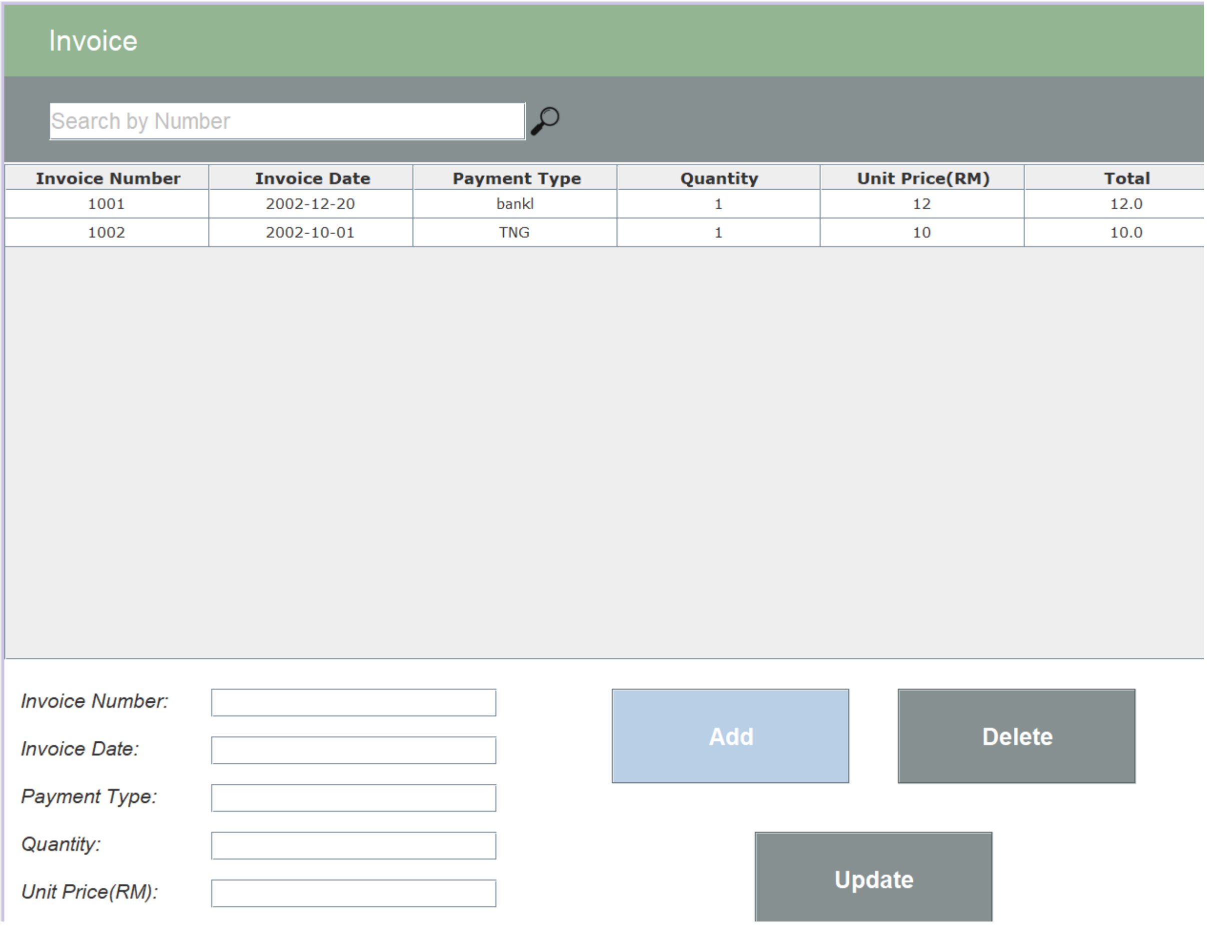
Invoice Date: 2002-10-01

Payment Type: TNG

Quantity: 1

Unit Price(RM): 10

After the user fills in all blanks, press the add button to add invoice.



Invoice

Search by Number

Invoice Number	Invoice Date	Payment Type	Quantity	Unit Price(RM)	Total
1001	2002-12-20	bankl	1	12	12.0
1002	2002-10-01	TNG	1	10	10.0

Invoice Number:

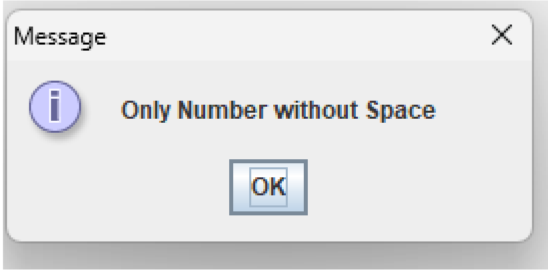
Invoice Date:

Payment Type:

Quantity:

Unit Price(RM):

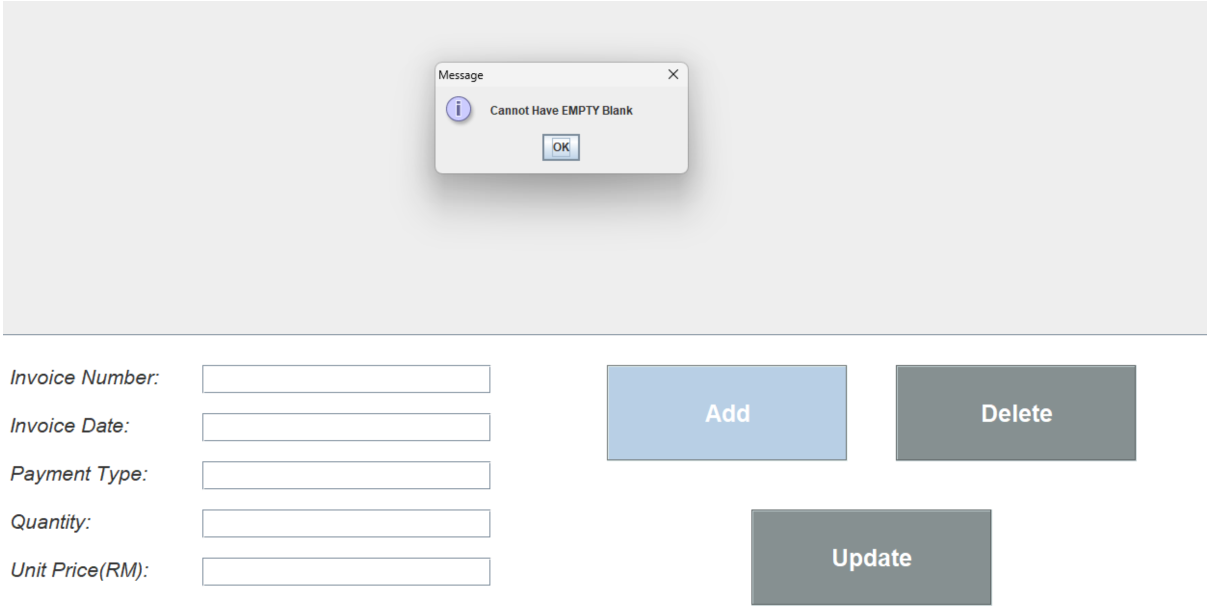
For the Invoice number, price and quantity only can input number.



Message

Only Number without Space

Users must fill all blanks.



Message

Cannot Have EMPTY Blank

Invoice Number:

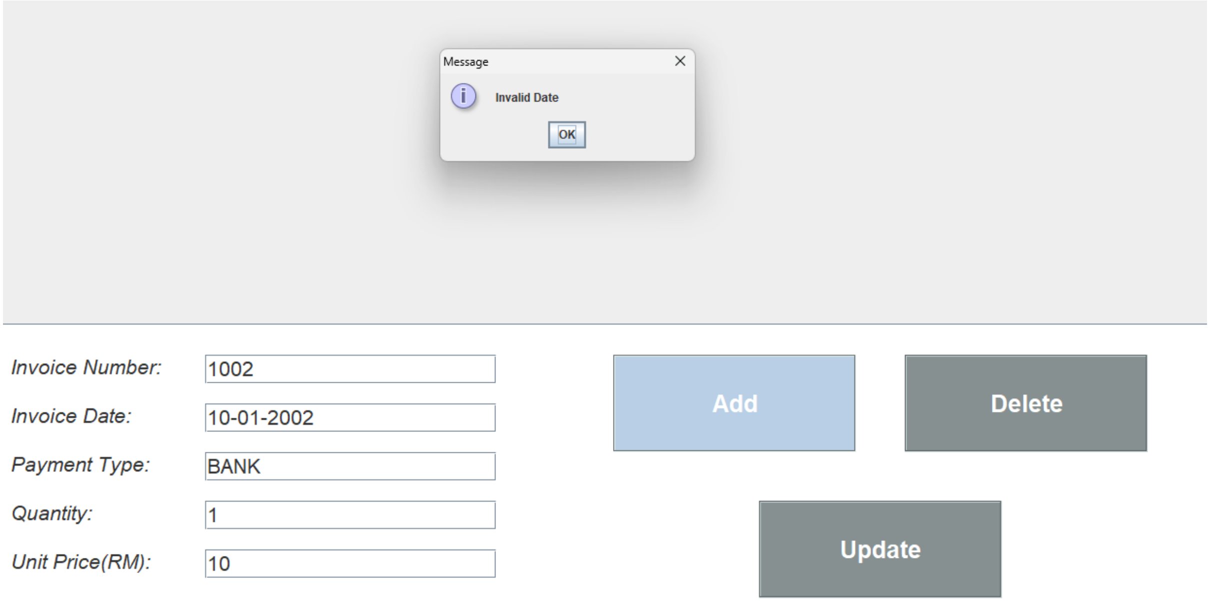
Invoice Date:

Payment Type:

Quantity:

Unit Price(RM):

For the invoice date must input “YYYY-MM-DD”



Message

Invalid Date

Invoice Number: 1002

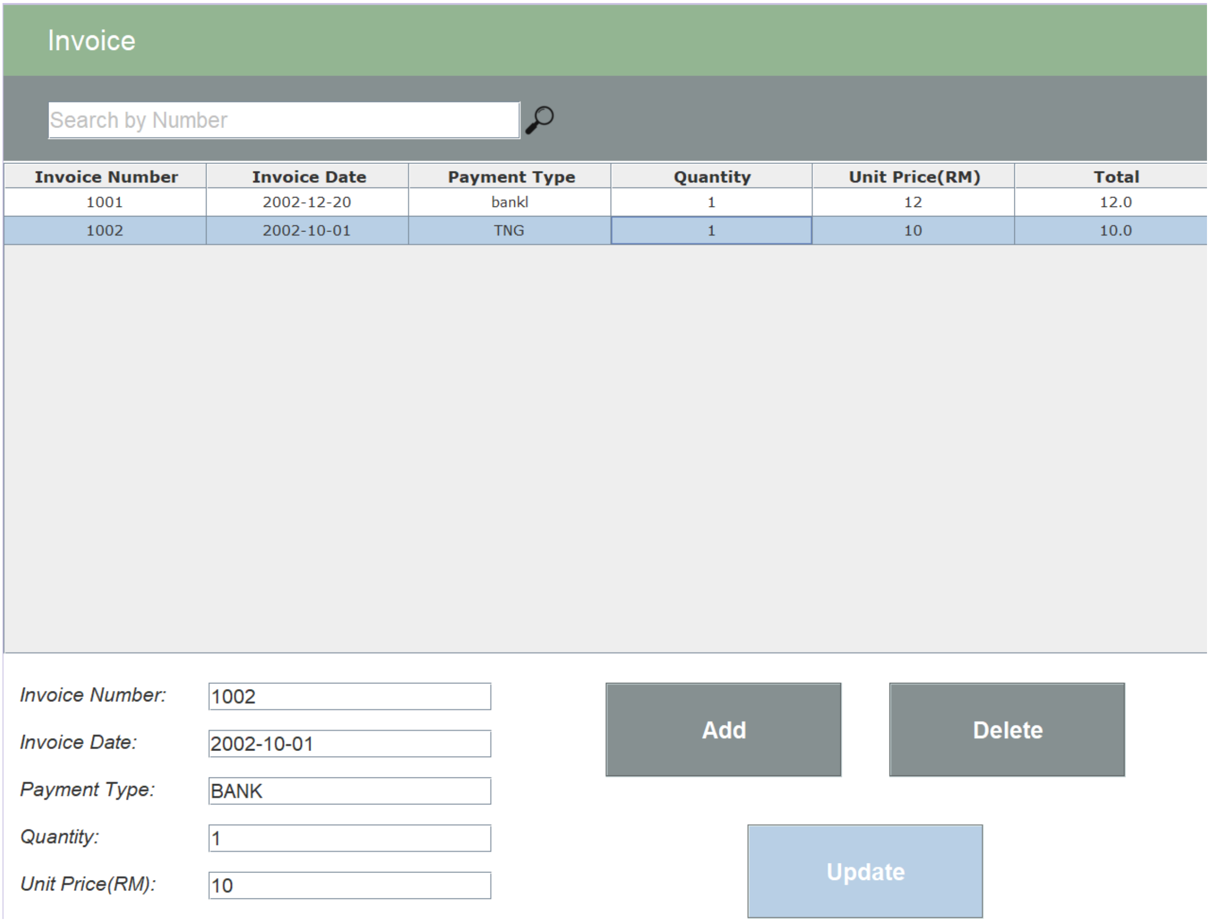
Invoice Date: 10-01-2002

Payment Type: BANK

Quantity: 1

Unit Price(RM): 10

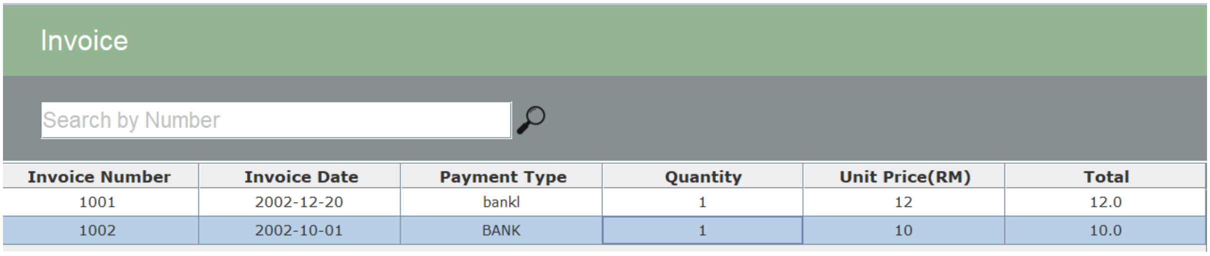
Users can use the update button to update selected invoice.



Invoice Number	Invoice Date	Payment Type	Quantity	Unit Price(RM)	Total
1001	2002-12-20	bankl	1	12	12.0
1002	2002-10-01	TNG	1	10	10.0

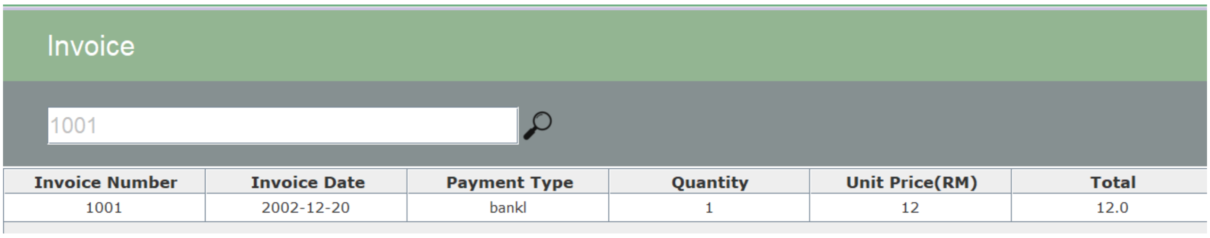
Invoice Number:	1002
Invoice Date:	2002-10-01
Payment Type:	BANK
Quantity:	1
Unit Price(RM):	10

After user press the update button



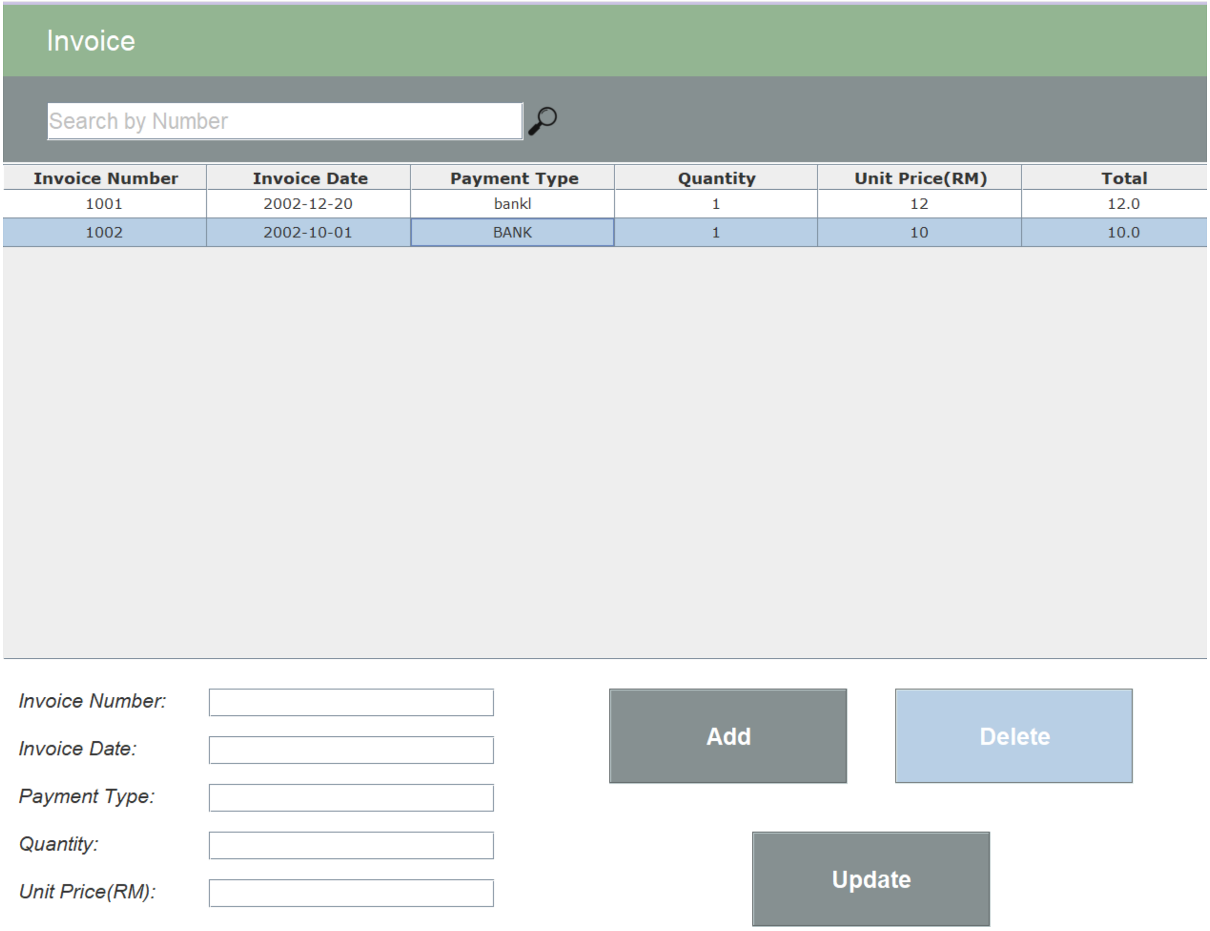
Invoice Number	Invoice Date	Payment Type	Quantity	Unit Price(RM)	Total
1001	2002-12-20	bankl	1	12	12.0
1002	2002-10-01	BANK	1	10	10.0

Users can use the search bar to search for the product information they want to find by input the invoice number.



Invoice Number	Invoice Date	Payment Type	Quantity	Unit Price(RM)	Total
1001	2002-12-20	bankl	1	12	12.0

Users can use the delete button to delete the product information they choose.



Invoice Number	Invoice Date	Payment Type	Quantity	Unit Price(RM)	Total
1001	2002-12-20	bankl	1	12	12.0
1002	2002-10-01	BANK	1	10	10.0

Invoice Number:

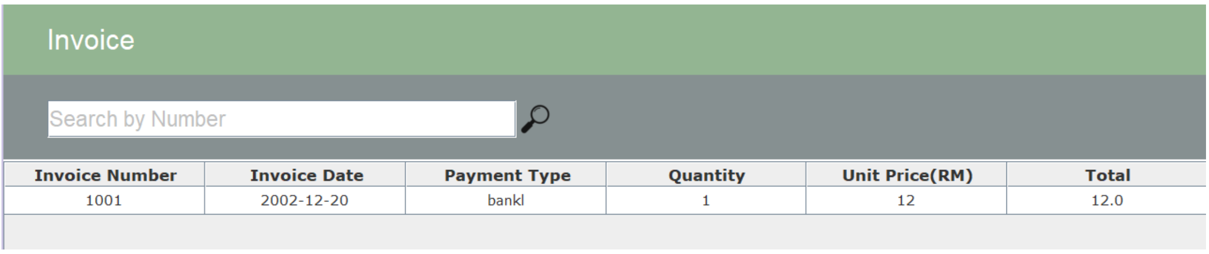
Invoice Date:

Payment Type:

Quantity:

Unit Price(RM):

After user press the delete button



Invoice Number	Invoice Date	Payment Type	Quantity	Unit Price(RM)	Total
1001	2002-12-20	bankl	1	12	12.0

Reference

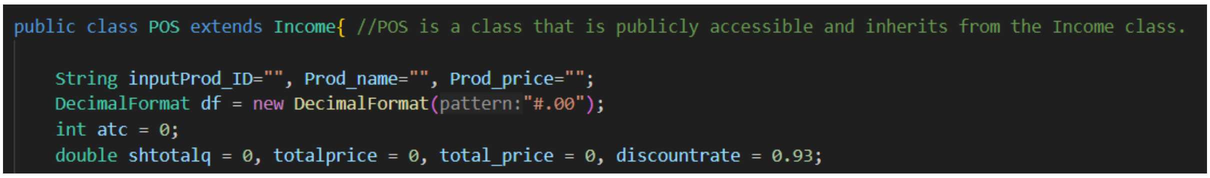
https://www.decipherzone.com/blog-detail/java-pos-guide

https://www.w3schools.com/java/

https://www.geeksforgeeks.org/file-class-in-java/?ref=lbp

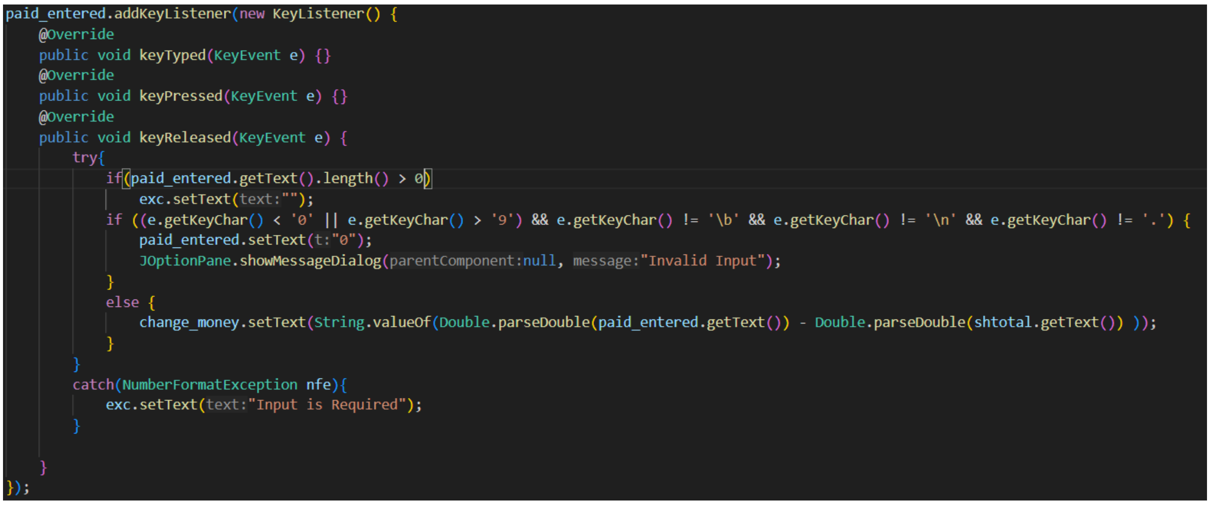
Appendix

Part of the code



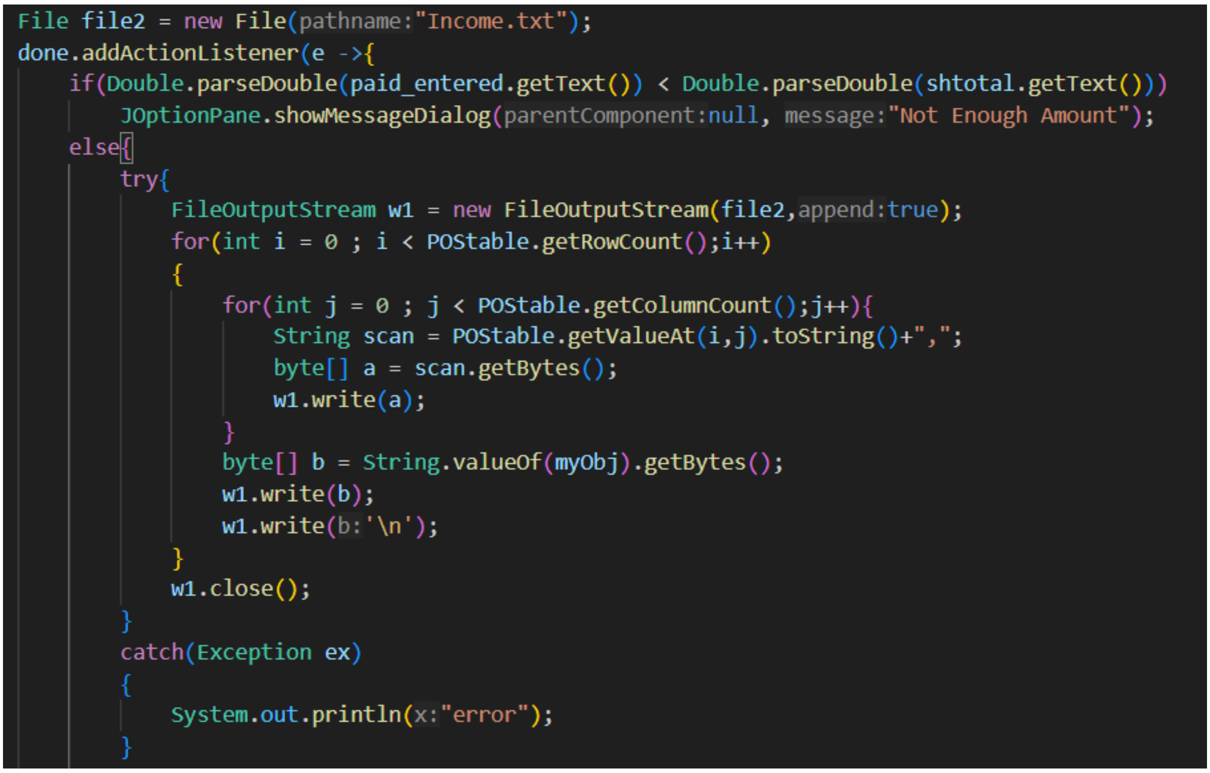
```
public class POS extends Income{ //POS is a class that is publicly accessible and inherits from the Income class.

    String inputProd_ID="", Prod_name="", Prod_price="";
    DecimalFormat df = new DecimalFormat(pattern:"#.00");
    int atc = 0;
    double shtotalq = 0, totalprice = 0, total_price = 0, discountrate = 0.93;
```



```
paid_entered.addKeyListener(new KeyListener() {
    @Override
    public void keyTyped(KeyEvent e) {}
    @Override
    public void keyPressed(KeyEvent e) {}
    @Override
    public void keyReleased(KeyEvent e) {
        try{
            if(paid_entered.getText().length() > 0)
                exc.setText(text:"");
            if ((e.getKeyChar() < '0' || e.getKeyChar() > '9') && e.getKeyChar() != '\b' && e.getKeyChar() != '\n' && e.getKeyChar() != '.') {
                paid_entered.setText(t:"0");
                JOptionPane.showMessageDialog(parentComponent:null, message:"Invalid Input");
            }
            else {
                change_money.setText(String.valueOf(Double.parseDouble(paid_entered.getText()) - Double.parseDouble(shtotal.getText()) ));
            }
        }
        catch(NumberFormatException nfe){
            exc.setText(text:"Input is Required");
        }

    }
});
```



```
File file2 = new File(pathname:"Income.txt");
done.addActionListener(e ->{
    if(Double.parseDouble(paid_entered.getText()) < Double.parseDouble(shtotal.getText()))
        JOptionPane.showMessageDialog(parentComponent:null, message:"Not Enough Amount");
    else{
        try{
            FileOutputStream w1 = new FileOutputStream(file2,append:true);
            for(int i = 0 ; i < POStable.getRowCount();i++)
            {
                for(int j = 0 ; j < POStable.getColumnCount();j++){
                    String scan = POStable.getValueAt(i,j).toString()+",";
                    byte[] a = scan.getBytes();
                    w1.write(a);
                }
                byte[] b = String.valueOf(myObj).getBytes();
                w1.write(b);
                w1.write(b:'\n');
            }
            w1.close();
        }
        catch(Exception ex)
        {
            System.out.println(x:"error");
        }
```