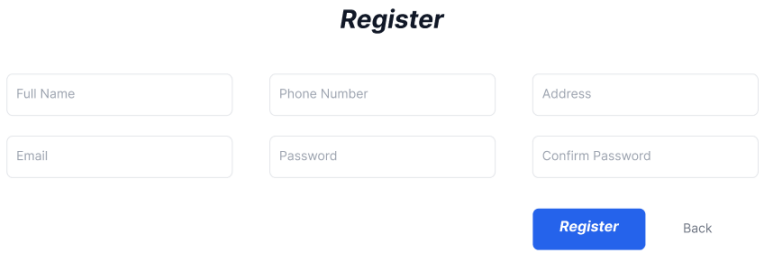
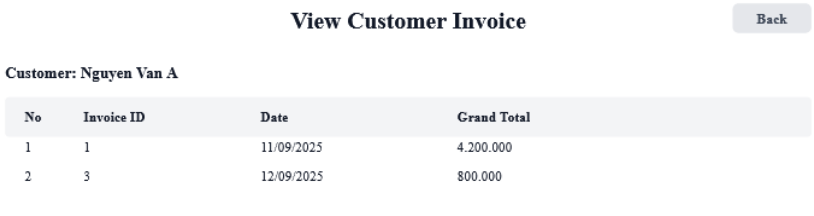
Scenario

Customer register as a member

Use case	Register as a member
Actor	Customer
Pre-condition	
Post-condition	Customer registered an account in the system
Main event	1. Customer open the supermarket website 2. The system display the main interface, with the register button 3. The customer click the register button 4. The system display a register interface, with text field to enter name, phone number, address, email and 2 password field for password  5. The customer enter their name, phone number, address, email and password twice and click register 6. The system display a success message
Exception	6. The system display an “Email already in use” error message 6.1 The customer either check their email or type a different email in the box and click Register again

Views customer statistics by revenue

Use case	Views customer statistics by revenue
Actor	Management staff
Pre-condition	The staff has a management account
Post-condition	The staff can view the customer statistics
Main event	1. The staff open the web 2. The system display the main interface 3. The management staff enter their username and password and click login 4. The system display the main employee interface 5. The staff click view statistics

	6. The system display the view statistic function, with an option to view customer statistics 7. The staff clicks view customer statistics 8. The system show an interface with 2 box to enter start and end date 9. The staff enter start date, end date and click enter 10 The system display the list of all customer and their generated revenue in the given time period  **View Customer Statistics** Back 11/09/2025 12/09/2025 Search No Name Customer ID Revenue 1 Nguyen Van A 2 5.000.000 2 Nguyen Van Anh 1 3.500.000 3 Tran Thi B 3 20.000 11. The staff click a customer 12. The system show all the transaction made by that customer in that time period  **View Customer Invoice** Back **Customer: Nguyen Van A** No Invoice ID Date Grand Total 1 1 11/09/2025 4.200.000 2 3 12/09/2025 800.000 13. The staff click on a specific transaction 14. The system show the detail of that transaction, including date, time, list of item in the transaction, number of each item, discount and total  **Invoice Detail** Back **Invoice ID: 1** Date: 11/09/2025 Time: 13:37 No Item Item ID Price Quantity Total 1 Pepsi 1 10.000 33 330.000 2 Kool-Aid 2 15.000 70 1.050.000 3 Watermelon 3 47.000 60 2.820.000 **Discount:** 0 **Grand Total:** 4.200.000
Exception	4. The system display “Username or password incorrect” message 4.1. The staff type their credential correctly and click Login button 10. The system display no customer in the period 10.1 The staff enter a different time period and click search again

Entity class diagram

Step 1: Describe the system

The system is a web-based platform that supports the management and operation of an online supermarket and it's user. Customers can register as members by selecting the registration menu, entering their personal information, and submitting it to be saved into the database. After registration, customers are able to search for items, place online orders, and also purchase products directly at the counter. Salers are able to sell items to customers directly at the counter. Warehouse staff can import items from suppliers, manage item and supplier information by adding, editing, or deleting records, browse online orders, and export them to delivery staff for delivery. Management staff can access statistical reports, with the option to view item, supplier or customer statistics by selecting a start and end date to review the list of customers along with their detailed transaction history or item with their related transactions or suppliers with their invoice in the given time period.

Step 2: Noun extraction

- Actor related noun:
  - Customer
  - Saler
  - Warehouse staff
  - Supplier
  - Delivery staff
  - Management staff
  - User
  - Staff
- Object related noun
  - Item
  - Order
  - Invoice
  - Supermarket
  - System
  - Menu
  - Report

Step 3: Noun evaluation

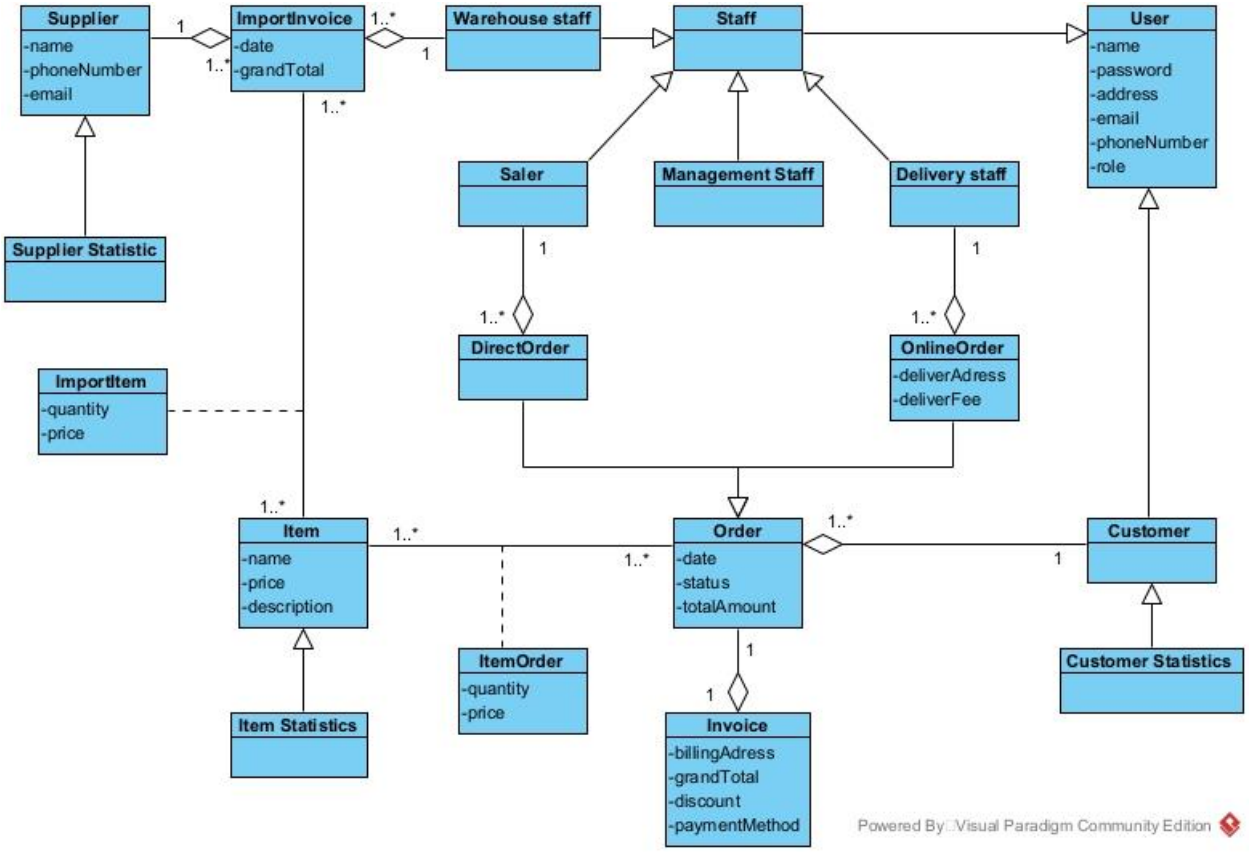
- System: too abstract
- Menu: too abstract
- Report: too abstract
- User -> Class User: name, password, address, email, phoneNumber, role

- Customer -> Class Customer(inherit from User)
- Staff -> Class Staff(inherit from User)
- Saler -> Class Saler(inherit from Staff)
- Warehouse staff -> Class WarehosueStaff(inherit from Staff)
- Management staff -> Class ManagerStaff(inherit from Staff)
- Delivery staff -> Class DeliveryStaff(inherit from Staff)
- Supplier -> Class Supplier: name, phoneNumber, email
- Item -> Class Item: name, price, description
- Order -> Class Order: date, status, totalAmount
- Online order -> Class OnlineOrder(inherit from Order): deliverAddress, deliverFee
- Direct order -> Class DirectOrder(inherit from Order)
- Invoice -> Class Invoice: billingAddress, grandTotal, discount, paymentMethod

Step 4: Determine quantity relationship between entity

- Supplier – ImportInvoice: 1 – n
- WarehouseStaff – ImportInvoice: 1 – n
- ImportInvoice – Item: n – n -> Class ImportItem: quantity, price
- Order – Item: n – n -> Class ItemOrder: quantity, price
- Order – Invoice: 1 – 1
- Customer – Order: 1 – n
- Saler – DirectOrder: 1 – n
- DeliveryStaff – OnlineOrder: 1 – n

Step 5: Determine object relationship between entity



Extract and draw the class diagram for the module.

Module register as a member

- Customer access the app: Need a class MainMenuView:

+ Register button: subRegisterForm

- The system show a register form : Need a class RegisterView

+ Input for name: inName

+ Input for address: inAddress

+ Input for phone number: inPhoneNumber

+ Input for email: inEmail

+ Input for password: inPassword

+ Input for confirm password: inConfirmPassword

+ Register button: subRegister

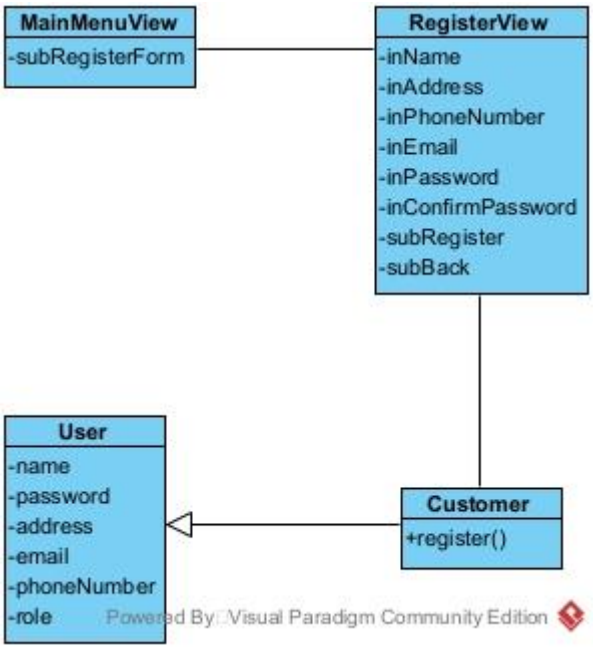
+ Back button: subBack

- The customer register an account: Need a method register():

+ Input: name, address, phone number, email, password

+ Output: boolean

+ Assign to class Customer



Module view customer statistics by revenue

- Management staff open the open and login : need a class LoginView

+ Input for username: inUsername

+ Input for password: inPassword

+ Login button: subLogin

- System displays an interface that has at least 1 button view statistic: Need a class StaffMenuView:

+ View statistics button: subStatistic

- System shows an interface that has at least 1 button view customer statistic: Need a class StatisticsView

+ View customer statistics button: subViewCustomerStatistics

- The system displays an interface to enter start and end date: Need a class CustomerStatisticsView

+ Input for start date: inStartDate

+ Input for end date: inEndDate

+ Display list of customer: outsubCustomerStatistics

+ Search button: subSearch

- Enter start and end date for customer statistics: Need a method getCustomerStatistics:

+ Input: start date, end date

+ Output: list of customer

+ Assign to class CustomerStatistics

- The system display an interface with list of a customer's invoice: Need a class CustomerInvoiceView

+ List of invoice: outsubCustomerInvoice

+ Back button: subBack

- Get all invoice from a customer in a period: Need method getInvoiceByCustomer():

+ Input: customer id, start date, end date

+ Output: list of Invoice

+ Assign to class Invoice

- The system display an Invoice: Need a class InvoiceView

+ Invoice detail: outInvoiceDetail

+ Back button: subBack

- Get an invoice by id: Need a method getInvoice()

+ Input: invoice id

+ Output: invoice detail

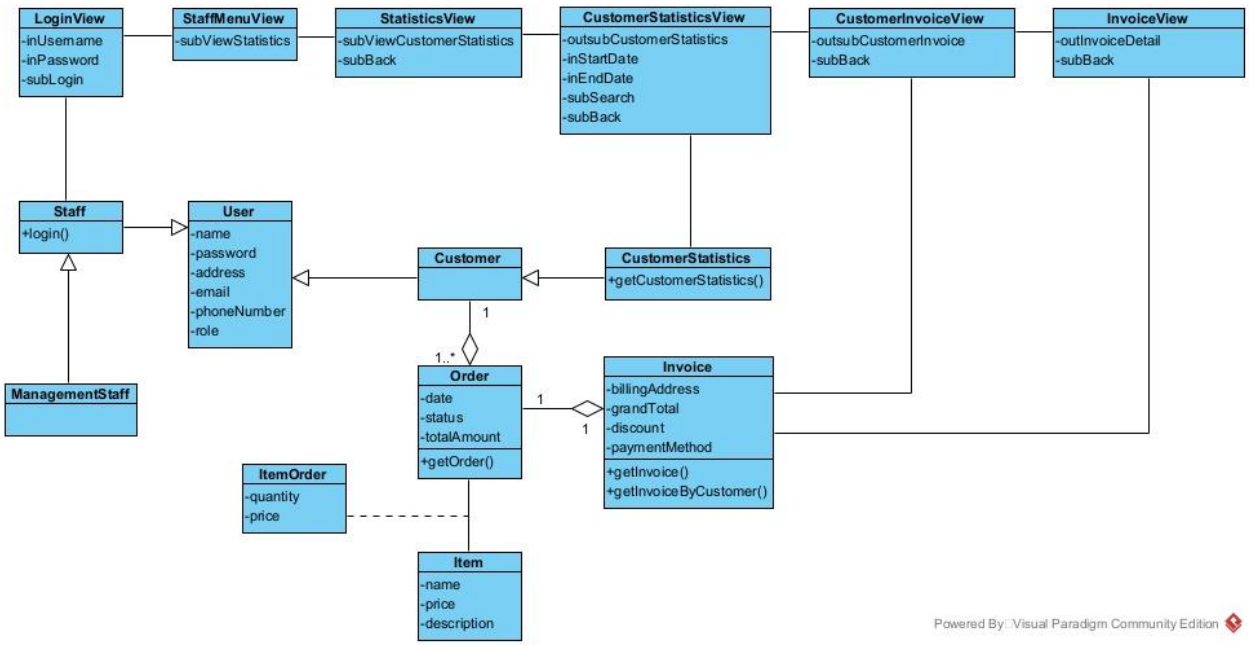
+ Assign to class Invoice

- Get order detail from invoice: Need a class getOrder()

+ Input: invoice id

+ Output: List of item in the order

+ Assign to class Order

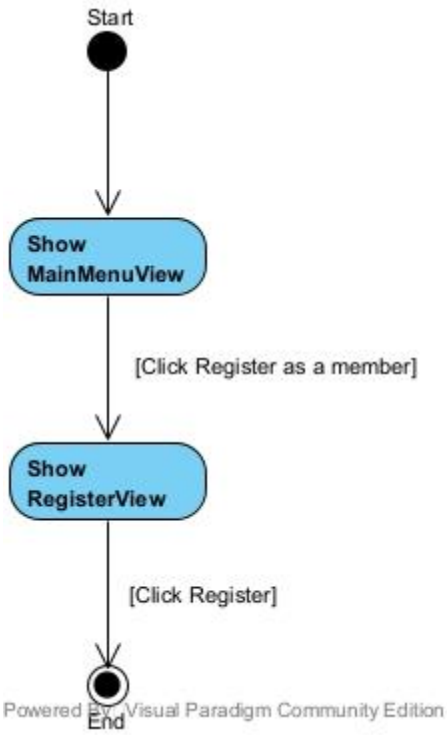


Draw the state diagram for the module.

Module register as a member

- From MainMenuView, clicking the register button, the system will move to RegisterView

- From RegisterView, after enter name, address, phone number, email, password, confirm password and click Register, the system will save the account and finish



Module view customer statistics by revenue

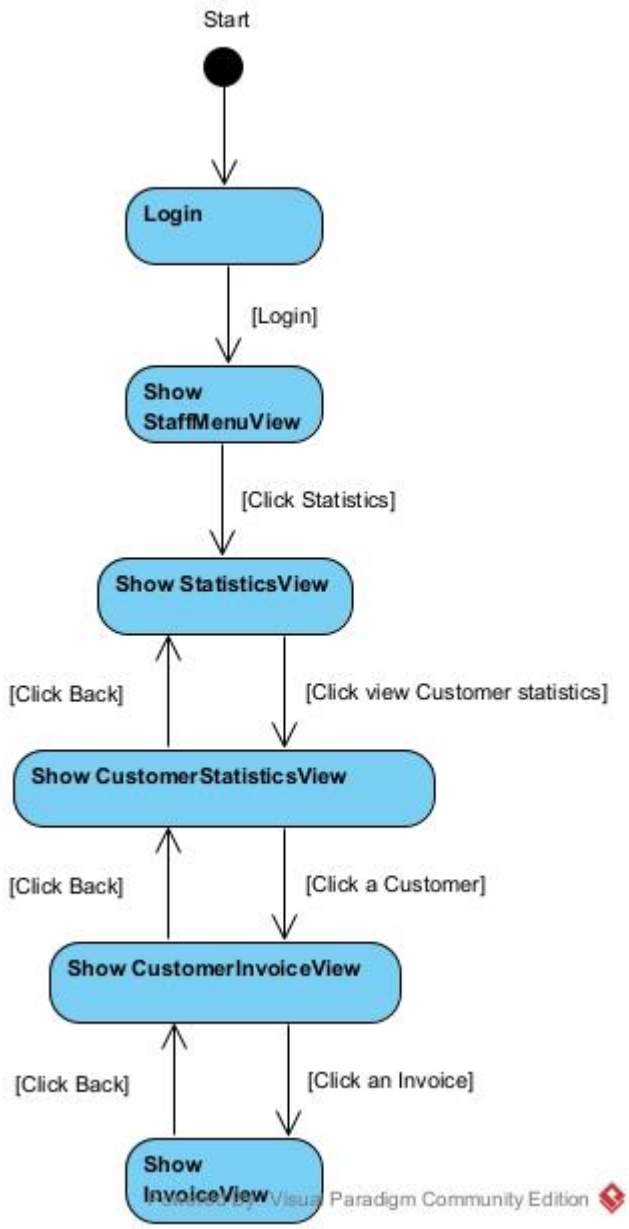
- From LoginView, the management staff enter username, password and click Login, the system will display the StaffMenuView

- From StaffMenuView, the staff click Statistics, the system move to StatisticsView

- From StatisticsView, after clicking view Customer statistics, the CustomerStatisticsView will display

- From CustomerStatisticsView, after enter start date, end date and click enter, a list of customer will be display. After clicking a customer, the system will move to CustomerInvoiceView.

- From CustomerInvoiceView, after click on an Invoice, the system will display the InvoiceView



Write detail scenario(ver 2.0)

Module register as a member

1. In main menu interface, customer click register
2. Class MainMenuForm call the class RegisterForm
3. The class RegisterForm display itself to the customer
4. The customer input information and click register
5. Class RegisterForm call the class Customer
6. Class Customer execute register()
7. Class Customer return the result to RegisterView

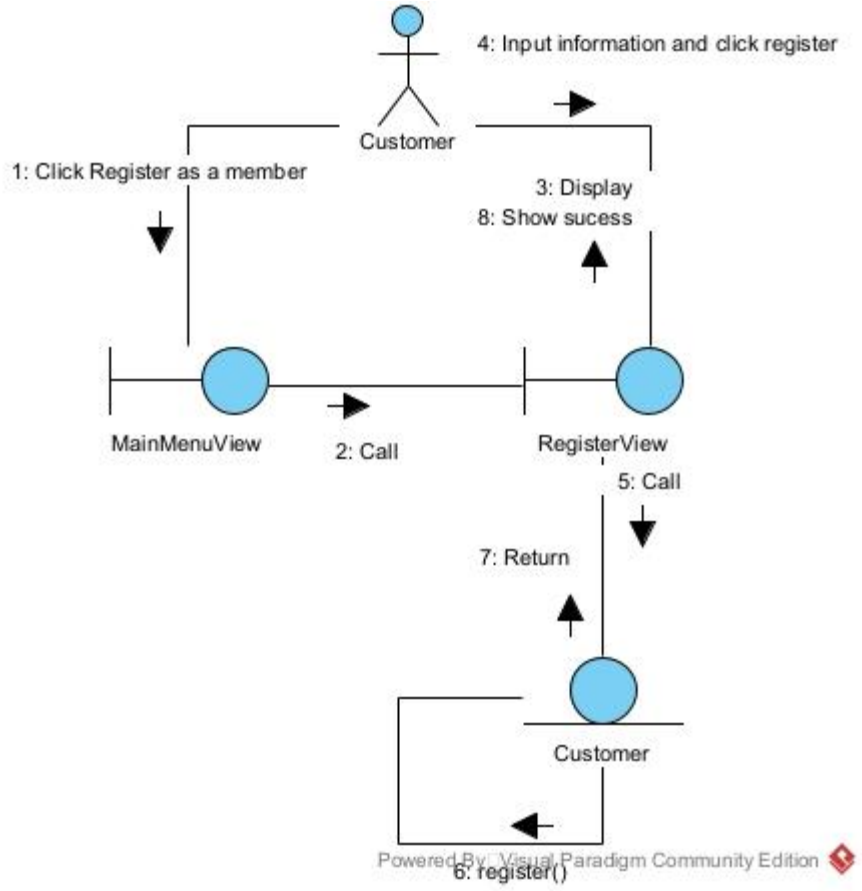
8. Class RegisterView display success message

Module view customer statistics by revenue

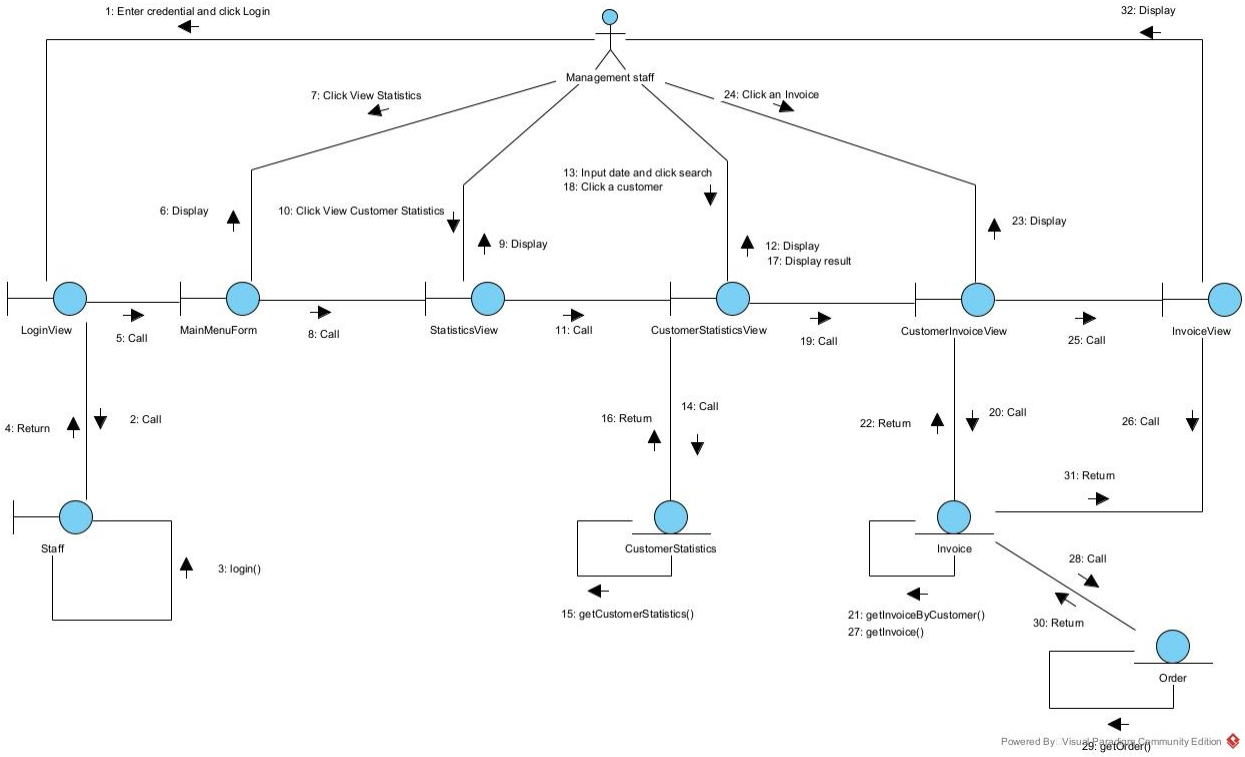
1. In the login interface, management staff enter username, password and click Login
2. The class LoginView call the class Staff
3. The class Staff execute login()
4. The class staff return the result to the class LoginView
5. The class LoginView call the class StaffMenuView
6. The class StaffMenuView display to the management staff
7. In main menu interface, management staff click view statistics
8. Class StaffMenuView call the class StatisticsView
9. The class StatisticsView display
10. The staff click view customer statistics by revenue
11. The class StatisticsView call the class CustomerStatisticsView
12. The class CustomerStatisticsView display
13. The staff enter date and click search
14. The class CustomerStatisticsView call the class CustomerStatistics
15. The class CustomerStatistics execute getCustomerStatistics
16. The class CustomerStatistics return the result
17. The class CustomerStatisticsView display the result
18. The staff click a Customer
19. The class CustomerStatisticsView call the class CustomerInvoiceView
20. The class CustomerInvoiceView call the class Invoice
21. The class Invoice execute getInvoiceByCustomer()
22. The class Invoice return the result
23. The class CustomerInvoiceView display the result
24. The staff click an invoice
25. The class CustomerInvoiceView call the class InvoiceView
26. The class InvoiceView call the class Invoice
27. The class Invoice execute getInvoice()
28. The class Invoice call the class Order
29. The class Order execute getOrder()
30. The class Order return the result to the class Invoice
31. The class Invoice return the result
32. The class InvoiceView display the result

Draw the communication diagram for the module.

Module register as a member

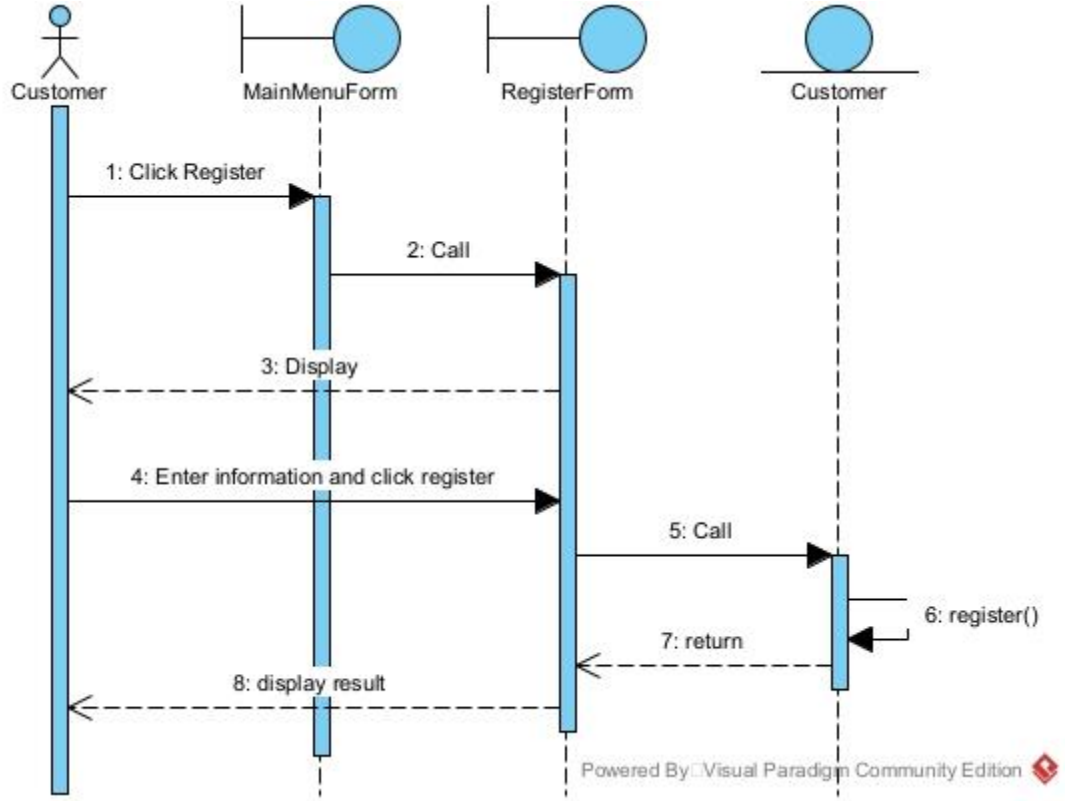


Module view customer statistics by revenue



Draw the sequence diagram for the module.

Module register as a member



Module view customer statistics by revenue