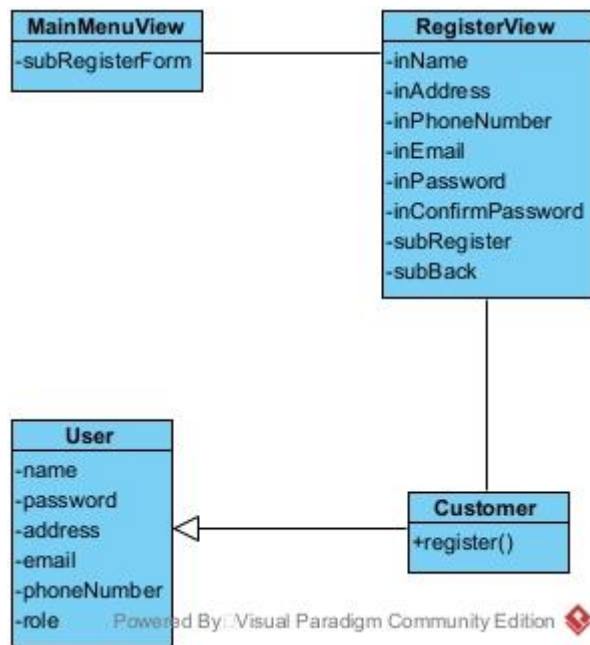


Analysis 2

Extract and draw the class diagram for the module.

Module register as a member

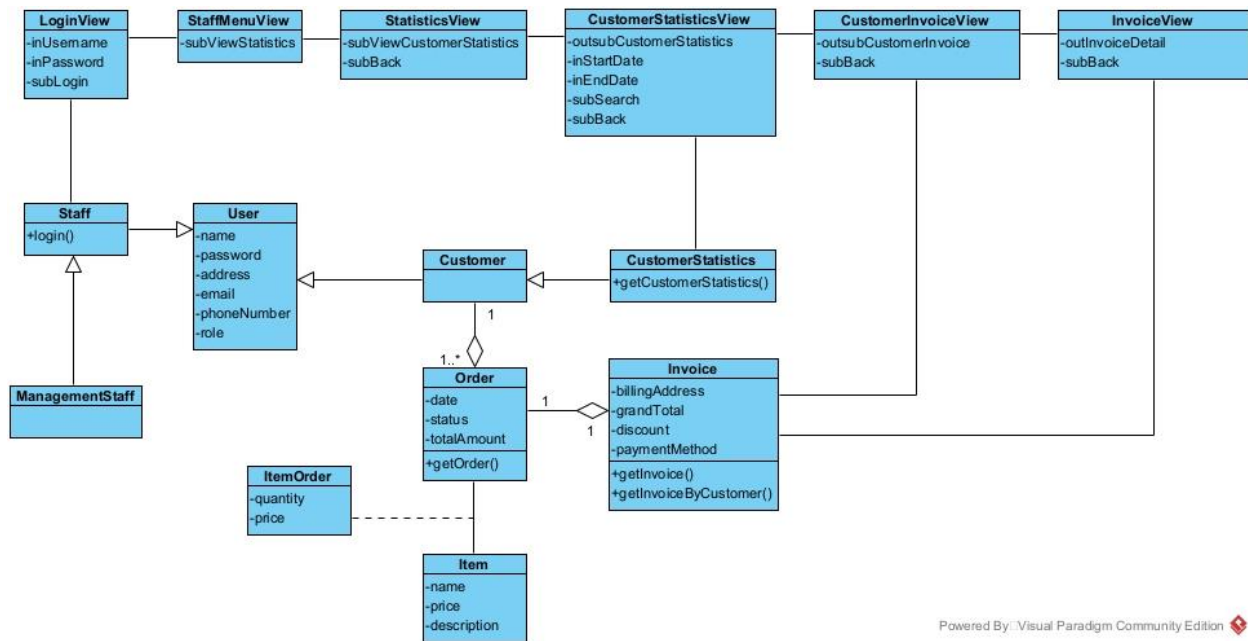
- Customer access the app: Need a class MainMenuView:
 - + Register button: subRegisterForm
- The system show a register form : Need a class RegisterView
 - + Input for name: inName
 - + Input for address: inAddress
 - + Input for phone number: inPhoneNumber
 - + Input for email: inEmail
 - + Input for 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

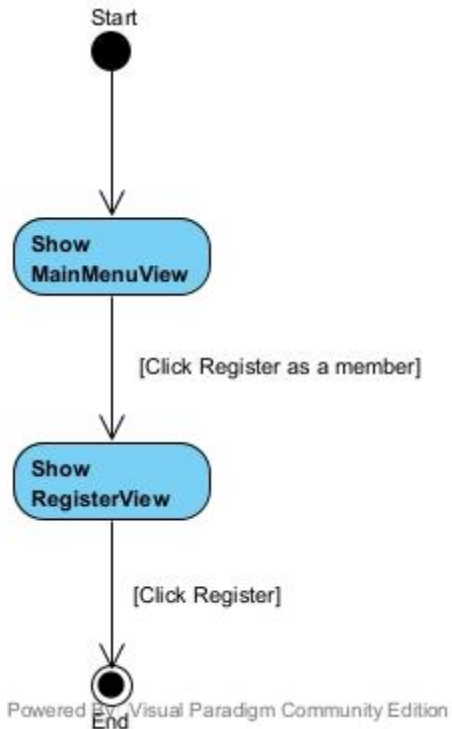


Powered By: Visual Paradigm Community Edition

Draw the state diagram for the module.

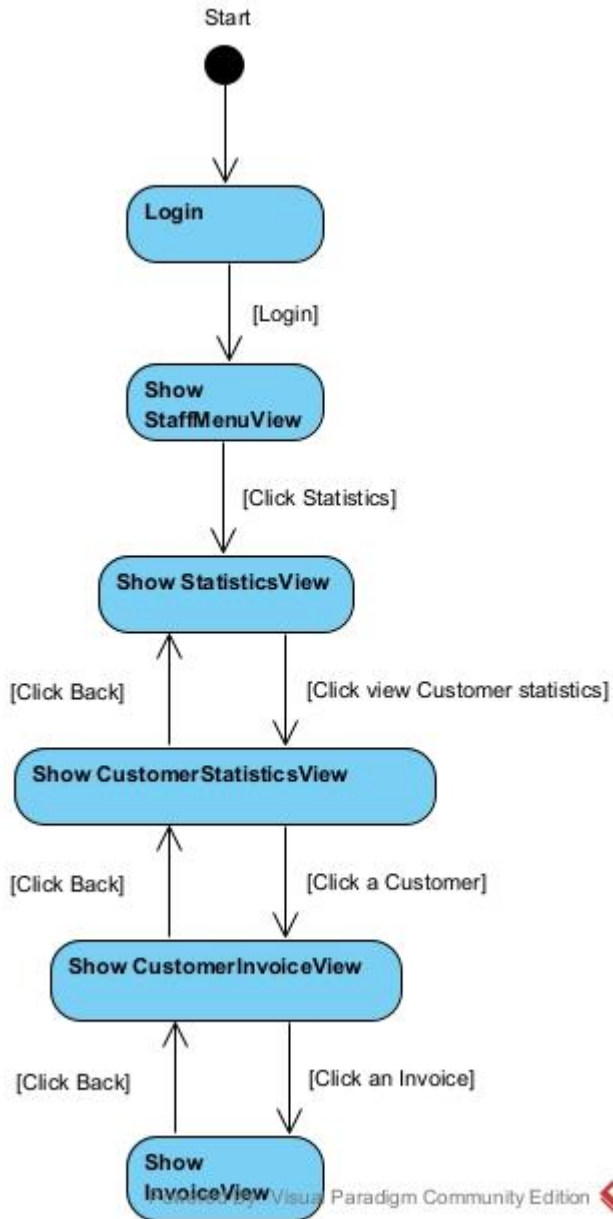
Module register as a member

- 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

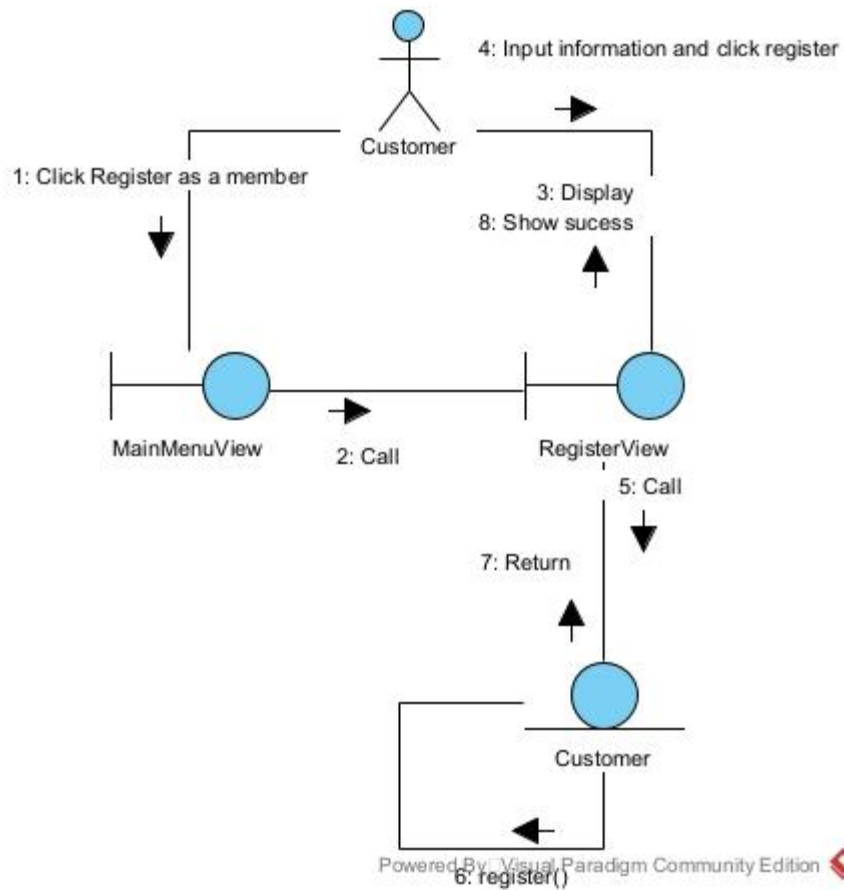
8. Class RegisterView display success message

Module view customer statistics by revenue

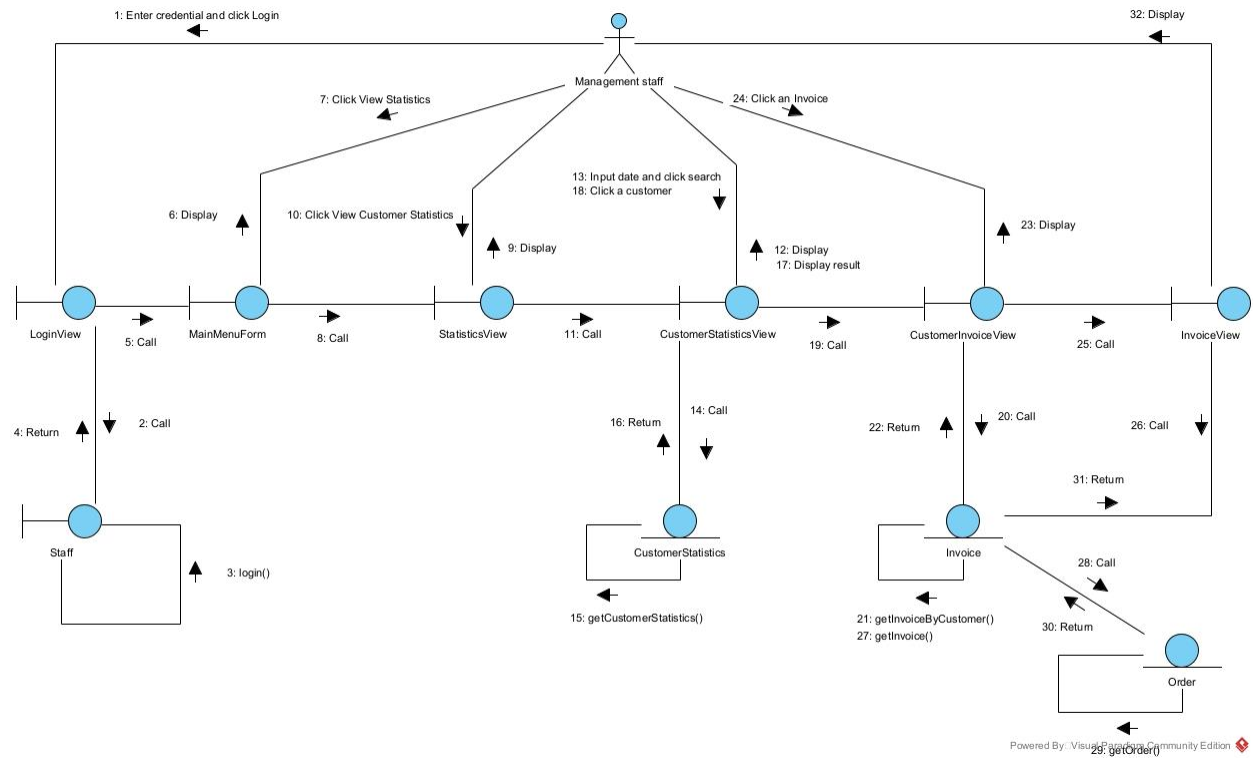
1. In the login interface, management staff enter username, password and click Login
2. The class LoginView call the class Staff
3. The class Staff execute login()
4. The class staff return the result to the class LoginView
5. The class LoginView call the class StaffMenuView
6. The class StaffMenuView 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

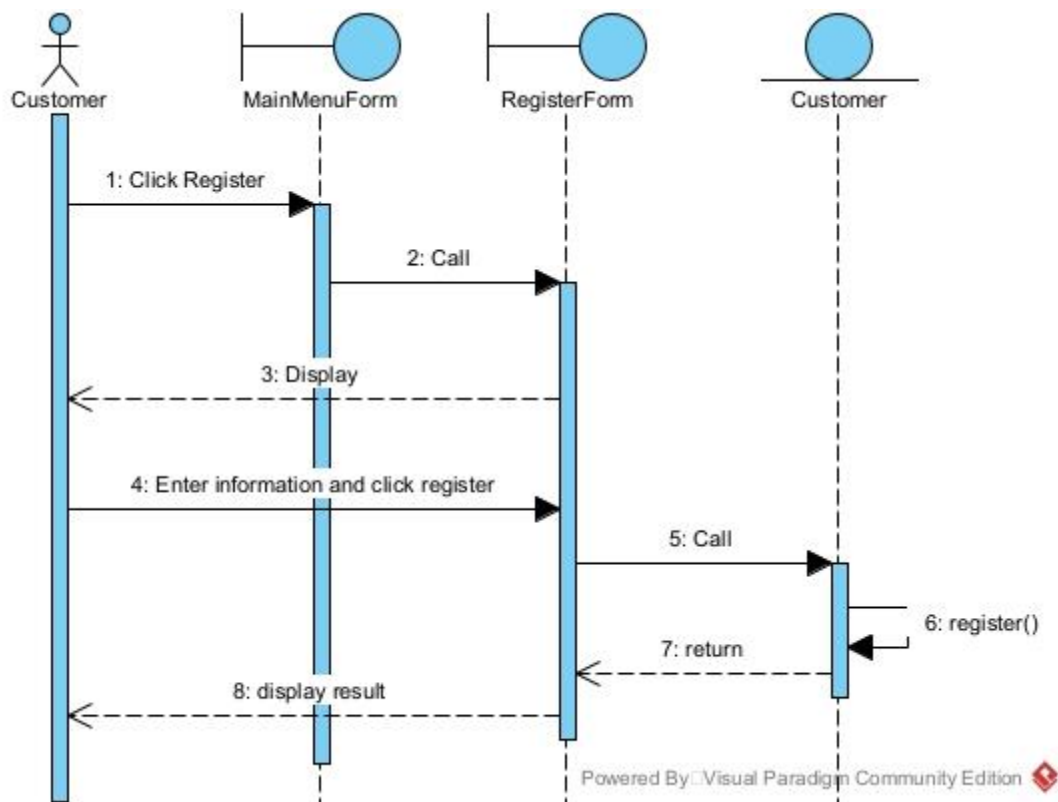


Module view customer statistics by revenue



Draw the sequence diagram for the module.

Module register as a member



Module view customer statistics by revenue

