I. Requirement

1. Glossary list

#	Name	Meaning
Related to human		
1	Staff	A person that works in the restaurant and have an account to use the system
2	Management Staff	A person with the highest authority including view statistics, manage dish information, make combo menus
3	Warehouse Staff	A person responsible for import ingredients from suppliers, manage supplier information
4	Sale Staff	A person who receives customers, takes orders, receives payments at the table, makes membership cards for customers, confirms table reservation information and order online
5	Customer	A person who benefits from the service, can search, book a table and order food online
6	Supplier	Partner providing ingredients
Related to activity		
7	View statistics	An action of manager staff that viewing the information of dish (name, description, price), ingredient (name, price, stock), customer (name, phone, email), supplier (name, phone, address)
8	Manage dish information	An action of manager staff that including adding new dish, changing or deleting existed dish information (name, description, price) in database
9	Make combo menu	An action of manager staff that combining two or more dish into a combo menu (name, description, list of dishes, price) and insert its information into database
10	Import ingredients from suppliers	An action of warehouse staff that confirming all the ingredients brought from suppliers and insert new ingredient or update stock of the existed ingredients in database
11	Manage supplier information	An action of warehouse staff that adding new supplier information (name, phone, email) and changing or deleting existed supplier information in database
12	Receive customer	An action of sale staff that meeting customer at reception hall, checking customer's information (name, phone, email) and guild them
13	Take orders	An action of sale staff that taking note of customer's orders (list of dishes, beverages) face-to-face and inform to chefs
14	Receive payment at the table	An action of sale staff that receiving customer's payment after they finish their meal and print bill (id, name, list of dishes and combos, price for each and total)

15	Make membership card for customer	An action of sale staff that creating and issue a card for customer to earn loyal points for their next reservation
16	Confirm table reservation information	An action of sale staff that must do when customer come to restaurant. The customer will tell the sale staff about their name, phone number, the table they pre-booked and show the staff their mail that the restaurant sent to confirm their successful reservation. The sale staff will check their information and lead them to the table pre-booked.
17	Order online	An action of sale staff. The customer first calls the restaurant to book a table. The sale staff picks up and then checks the available table according to the schedule the customer gives them then informs the customer. If the customer agrees to the staff suggestion, the staff will ask the customer's information to make a table reservation and save it to database. After that, the sale staff will send a mail to confirm their successful reservation
18	Add new dish	An action of management staff that is included in manage dish information action. The management staff will select adding new dish option in the app interface then fill the dish information in (name, description, price). The dish information will be saved into database
19	Search table	An action of customer to search for available table in a specific time online
Related to object		
20	Dish	A specific type of food that has been prepared and is ready to be served or eaten. Its information will be saved into database (id, name, description, price)
21	Ingredient	A specific food item is used to make a dish. For example, in a salad, the ingredients are lettuce, tomatoes, cucumbers, and dressing. In a cake, the ingredients are flour, sugar, eggs, and milk. Its information will be saved into database (id, name, price)
22	Combo	Short for "combination," is a package deal that bundles several individual items together and sells them as a single offer, usually at a discounted price. In the context of a restaurant, a combo typically includes a main dish, a side dish, and a drink. Its information will be saves into database (id, name, list of dishes and beverages, price)
23	Menu	A list of the food and beverages that are available for purchase. The customers can see the menu online or menu book on the table. It contains name of the dish, price

24	Order	A request made by a customer to a restaurant for food and drinks. Customer can make order right at the table or via website (order online)
25	Payment	It is the process of a customer settling the cost of their order. The sale staff will receive money from customer and update payment information in database
26	Table	A physical location in the restaurant where customers are seated and is linked to a customer order
27	Membership card	A physical or digital card issued by the restaurant to its customers. It is linked to a customer's account in the system and grants them special privileges (such as discounts, loyalty points)

2. Project information

a. Object

A website system for restaurant management including viewing statistics (dishes, ingredients, customers, suppliers), managing dish information, making combo menus for manager; importing ingredients from suppliers, managing suppliers information for warehouse staff; receiving customers (taking orders, receiving payments at the table, make membership cards, confirm table reservation information) for sale staff. Booking table and order foods online for customer

b. Scope

- Staff:
  + Login/ logout
- Manager:
  + View statistics
  + Manage dish information
  + Making combo menus
- Warehouse staff:
  + Import ingredients from suppliers
  + Manage supplier's information
- Sale staff:
  + Confirm table reservation
  + Taking order
  + Receiving payment
  + Make membership card for customer
- Customer:
  + Booking table
  + Order foods

c. How the module works

Management staff adds new dish: Staff logins into the system → selects the menu to manage dish information → selects the function of adding dish information →the system displays adding dish information interface with text fields (name, description, price) → enters dish information and clicks add → The system saves to the database.

Management staff views customer statistics: Selects the menu to view statistical reports → selects customer statistics by revenue → the system displays an interface with a combo box to select start and end date→ selects the time to start and end statistics → view with a list of customers (name, phone number, revenue)→ selects a customer to view expenses details at chosen start and end date → the system displays an interface with the order list that the customer has called (date, price) → selects view 1 order → views detail the selected order (list of dishes with number and price).

d. Information about objects

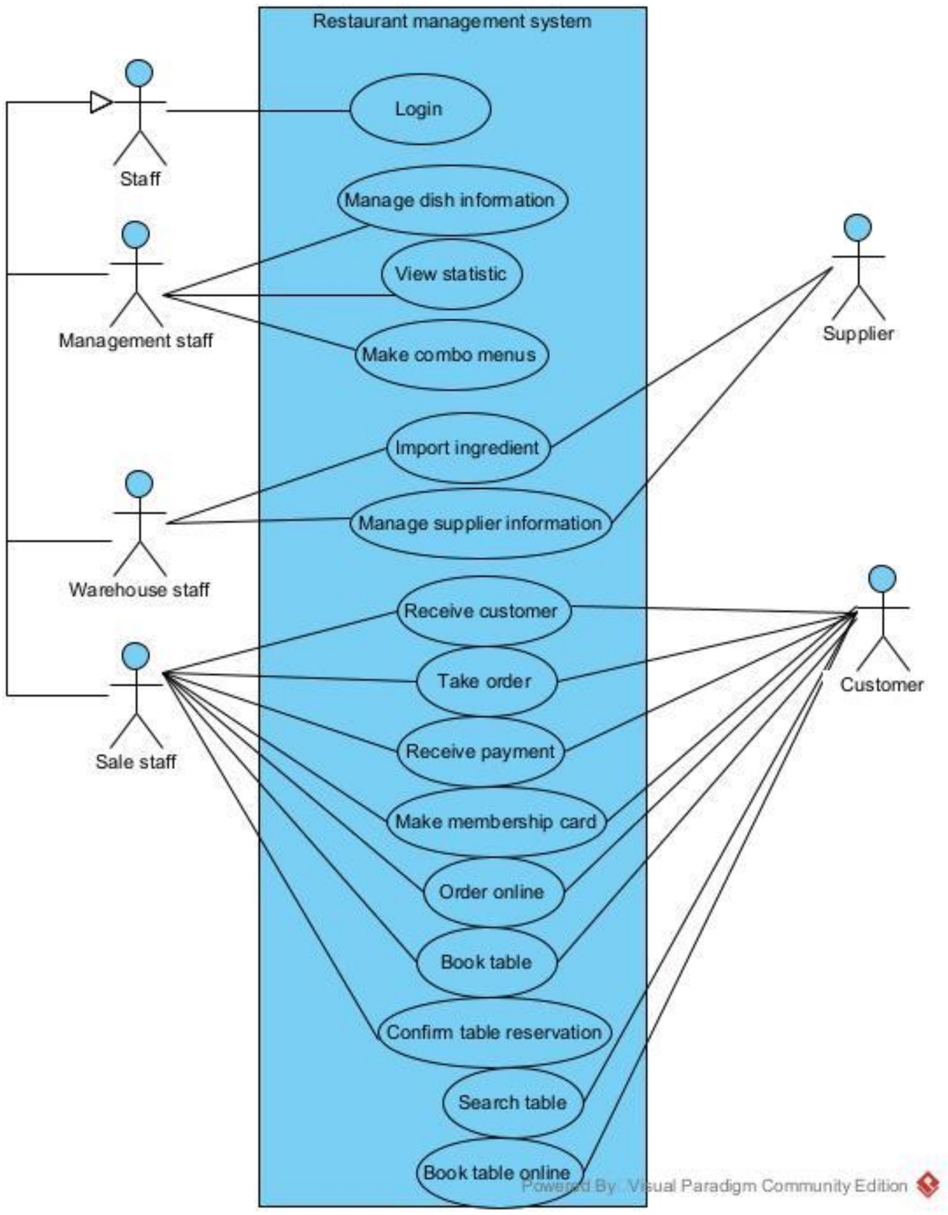
- Staff: username, password, name, phone number, email, role
- Customer: name, phone number, email
- Reservation:
- Invoice:
- Dish: name, description, price
- Order: date, price
- Table: type, available
- Supplier: name, phone, email
- Ingredient: name, stock, price
- Combo: name, price
- Membership: type, point, start date, due date
- Import invoice: import date, price

e. Relationships among objects

- Customer – Membership: 1 – 1
- Customer – Reservation: 1 – n
- Reservation – Order: 1 – n
- Reservation – Table: n – 1
- Reservation – Invoice: 1 – n
- Order – Invoice: n – 1
- Order – Dish: n – n
- Order – Combo: n – n
- Dish – Combo: n – n
- Sale staff – Invoice: 1 – n
- Ingredient – ImportInvoice: n – n
- Supplier – Import invoice: 1 – n
- Warehouse staff – Import invoice: 1 – n

3. Use case diagram

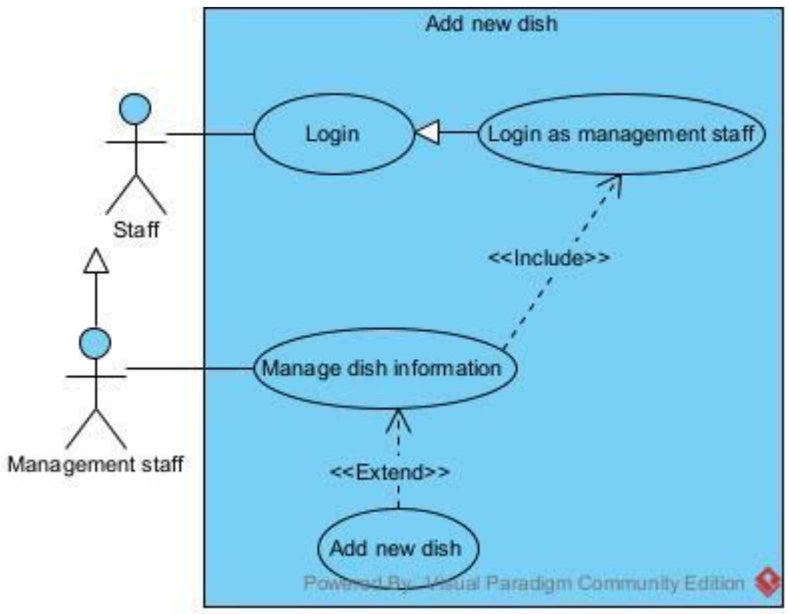
f. General use case



- Description:
  + Login: This use case enables the staff to login into the restaurant system
  + Manage dish information: This use case enables the management staff to adding new dish or change/delete exited dish
  + View statistic: This use case enables the management staff to view the information of dish, ingredient, customer, supplier
  + Make combo menus: This use case enables the management staff to make combo menus
  + Import ingredient: This use case enables the warehouse staff to import ingredient
  + Manage supplier information: This use case enables the warehouse staff to adding new supplier information or change/delete exited one

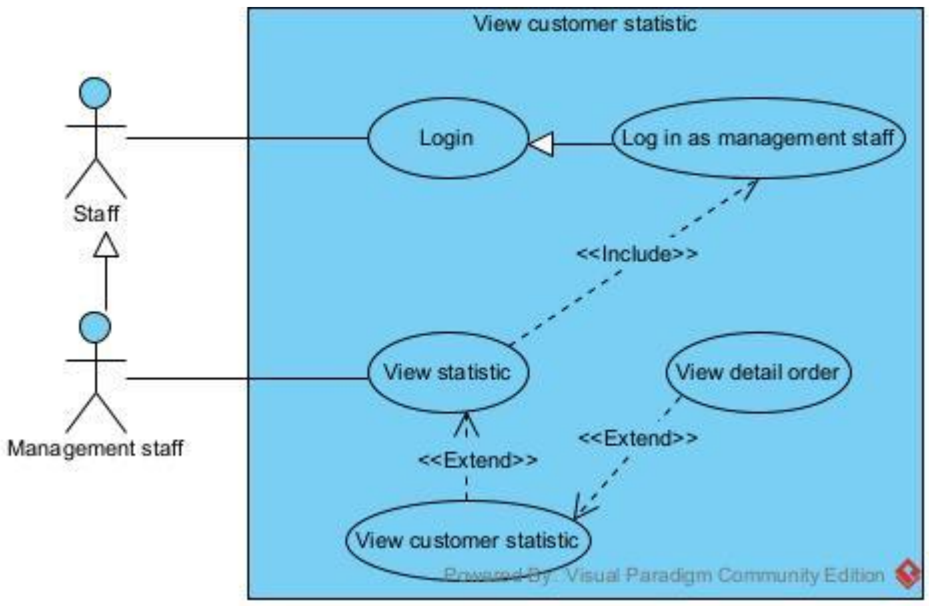
+ Receive payment: This use case enables the sale staff to receiving customer's payment after they finish their meal and print bill
+ Take order: This use case enables the sale staff to receive order offline
+ Confirm table reservation: This use case enables the sale staff to confirm the customer table reservation
+ Make membership card: This use case enables the sale staff to create and issue a card for customer to earn loyal points for their next reservation
+ Order online: This use case enables the customer to order online
+ Book table: This use case enables the sale staff to book table offline according to customer's request
+ Confirm reservation: This use case enables the sale staff confirm the reservation of the customer
+ Search table: This use case enables the customer to search for available table
+ Book table online: This use case enables the customer to book table online

g. Management staff adds new dish use case



- Login as management staff: This use case enables the management staff to login into their account
- Add new dish: This use case enables the management staff to add new dish into database

h. Management staff views customer statistic



- View customer statistic: This use case enables the management staff to view customer statistic
- View detail order: This use case enables the management staff to view a detail order in the list of orders of the selected customer

II. Analysis

1. Scenario

i. Module 1

Use case	Add new dish
Actor	Management staff
Pre-condition	The management staff has an account
Post-condition	The management staff has added new dish into database
Main events	1. The management staff logins with username = "a", password = "12345" into the system to add new dish 2. The main interface which contains a Manage dish information button appears 3. The management staff selects the Manage dish information button 4. The system displays manage dish interface with an Add new dish button and a list of all dishes: | # | Name | Description | Price | | 1 | b | A delicious meal | 100.000 | 5. The management staff click the Add new dish button 6. The system shows the Add new dish interface with input of dish name, description, price and an Add button 7. The management staff enters dish name = "Noodle", description = "a staple food made from unleavened dough that is rolled flat and cut, or extruded, into long strips or

various other shapes”, price = “10.000” and clicks Add button

8. The system displays the success message box
9. The staff clicks OK button
10. The system returns to the manage dish interface

#	Name	Description	Price
1	b	A delicious meal	100.000
2	Noodle	a staple food made from unleavened dough that is rolled flat and cut, or extruded, into long strips or various other shapes	10.000

Exception	2. The system shows error message box 2.1. The management staff clicks OK button 2.2. The system returns to the login interface

j. Module 2

Use case	View customer statistics
Actor	Management staff
Pre-condition	The management staff has logged in the account
Post-condition	
Main events	

1. After login, the management staff selects View statistics button from the main interface
2. The system displays an interface that has 3 buttons which is View dish statistics, View customer statistics, View supplier statistic
3. The management clicks on View customer statistics
4. The system shows an interface with the input of start and end time of the statistics and a Search button
5. The management staff enters start time = “9/9/2025”, end time = “16/9/2025” and click Search
6. The system displays a list of customers

#	Name	Phone	Email	Revenue
1	a	098720	A@gmai.com	1.000.000
2	ab	213230	AB@gmail.com	500.000

7. The management staff click on the first row of the list
8. The system displays the order list interface of the customer

#	Order id	Order date	Total amount
1	1	10/9/2025	600.000
2	2	13/9/2025	400.000

9. The management staff select the first row
10. The system displays an interface contains order date = “10/9/2025”, Total amount= “600.000” and a list of dishes and combos that was ordered

List of dishes			
#	Name	Price	Amount
1	Noodle	10.000	10
2	Korean dish	100.000	5

	List of coombos			
	#	Name	Price	Amount
	1	Family combo	50.000	2
Exception	6. No customer displays on the list			

2. Entity class

k. Step 1:

The system supports the management of information about dishes, ingredients, customers, suppliers, orders, reservations, and combo menus in a restaurant. The system enables management staff to manage dish information, set up combo menus, manage suppliers, and view statistical reports such as dish statistics, customer statistics, and supplier statistics. The system enables warehouse staff to import ingredients from suppliers and manage supplier information. The system enables sales staff to receive customers, take orders, confirm reservations, process payments, and issue membership cards for customers. The system enables customers to search for dish information, book a table online, and order food online. Once a payment or import is processed, the system generates invoices with details about the ordered dishes, imported ingredients, suppliers, and customers involved.

l. Step 2 + 3:

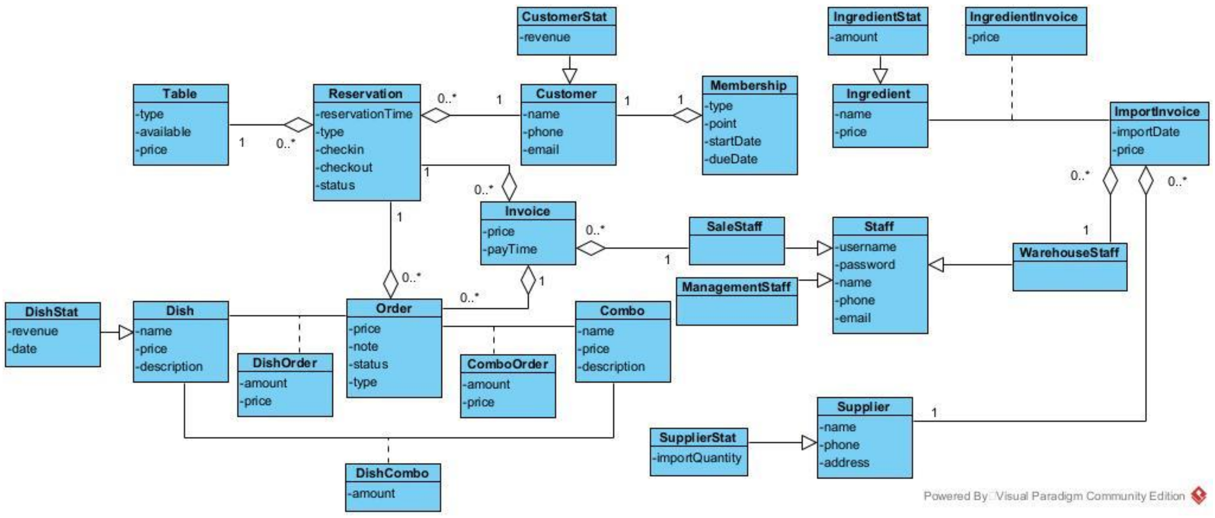
- Restaurant: Out of scope => eliminate
- System: Too abstract => eliminate
- Management staff, sale staff, warehouse staff => Class Staff: username, password, name, role
- Statistics: Too abstract => eliminate
- Dishes => Class Dish: name, price, description
- Ingredients => Class Ingredient: name, price, amount
- Customers => Class Customer: name, phone, email
- Suppliers => Class Supplier: name, phone, email
- Information: Too general
- Combo => Class Combo: name, price, description
- Menu: Too general
- Orders => Class Order: price, note, type, status
- Payment => Class Invoice: price, payTime
- Membership card => Class Membership: type, point, startDate, dueDate
- Table => Class Table: type, available
- Reservation => Class Reservation: reservationTime, type
- Food: Too general
- Invoice for imported ingredients => ImportInvoice: importDate, price

m. Step 4

- Customer – Membership: 1 – 1

- Customer – Reservation: 1 – n
- Reservation– Invoice: 1 – n
- Reservation – Order: 1 – n
- Reservation – Table: n – 1
- Order – Invoice: n – 1
- Order – Dish: n – n => create a new class DishOrder: amount, price
- Order – Combo: n – n => create a new class ComboOrder: amount, price
- Dish – Combo: n – n => create a new class DishCombo: amount
- Ingredient – ImportInvoice: n – n => create a new class IngredientInvoice
- Supplier – ImportInvoice: 1 – n
- WarehouseStaff – ImportInvoice: 1 – n
- SaleStaff – Invoice: 1 – n

n. Step 5



3. Module 1:

o. Class diagram

- Management staff logs into the system: need a class: LoginView:

  + Input for username: inUsername

  + Input for password: inPassword

  + Login button: subLogin

- Enter the username/ password to login: need a method: checkLogin()

  + Input: username, password (of the class Staff)

  + Output: boolean

  + Assign to class Staff

- Main interface contains Manage dish button appear => need a class: ManagementHomeView:

+ Manage dish button => subManageDish

+ View statistics => subViewStatistic

- System displays an interface with 2 buttons which is Edit dish and Add new dish => need a class: ManageDishView

  + Add dish button => subAddDish

  + A list of all dishes=> outListDish

- Display a list of all dishes => need a method: getAllDish():

  + Input: None

  + Output: List of Dish

  + Assign to class Dish

- System shows the Add new dish interface => need a class: AddDishView

  + Input for dish name: inName

  + Input for dish description: inDescription

  + Input for price description: inPrice

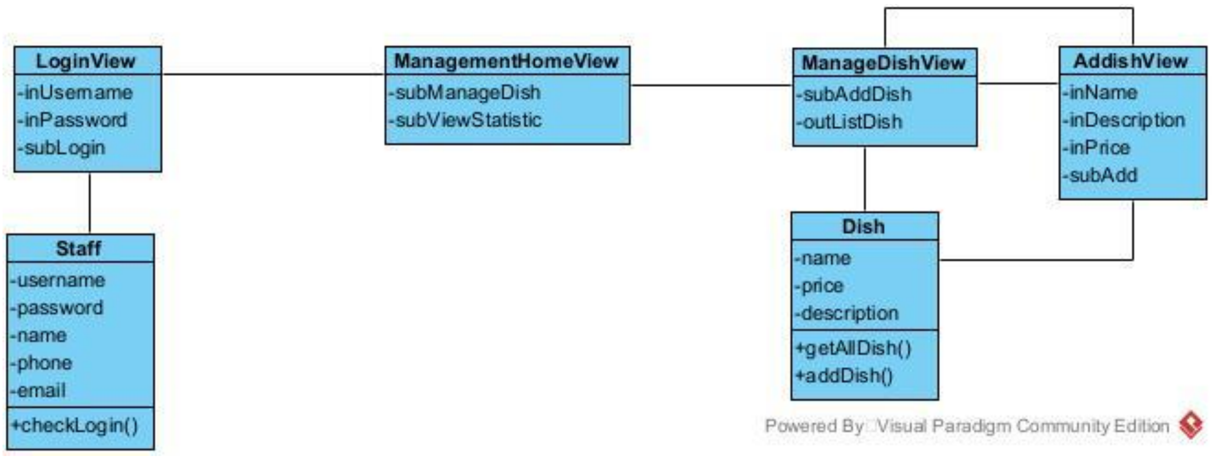
  + Add button: subAdd

- The dish is added into the database => need a method: addDish()

  + Input: name, description, price (belongs to class Dish)

  + Output: boolean

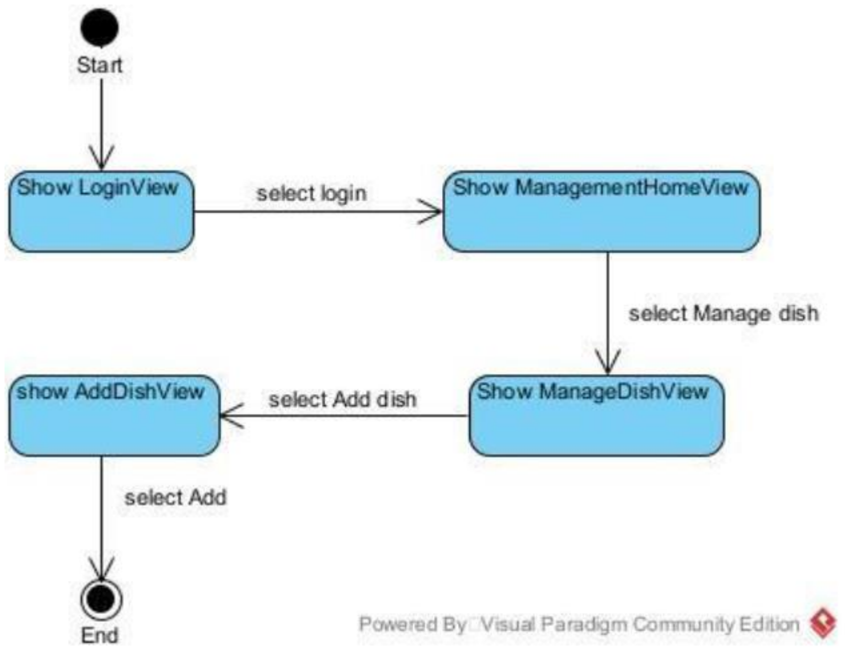
  + Assign to class Dish



p. State diagram:

- From LoginView, after entering username/ password and select login, the system will move to ManagementHomeView

- From ManagementHomeView, select Manage Dish option will have the system move to ManageDishView
- From ManageDishView, when select option add dish, the system will show the AddDishView
- From AddDishView, after entering username/ description/ price and click Add, the system will save it into database and finish

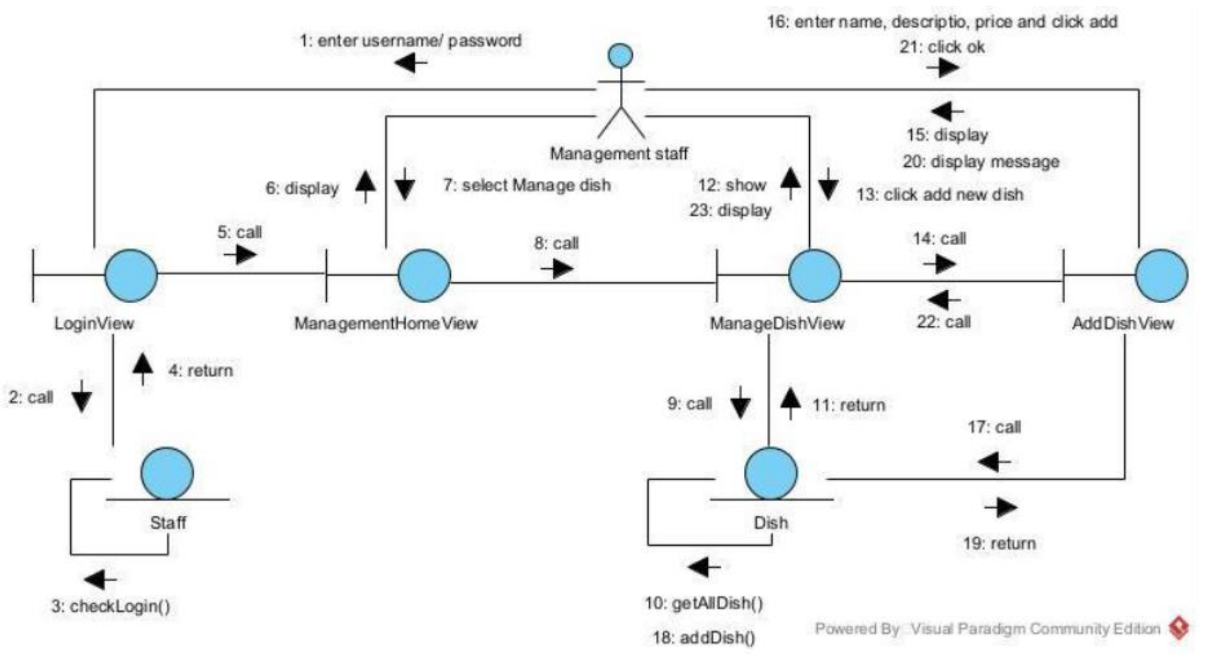


q. Scenario ver 2:

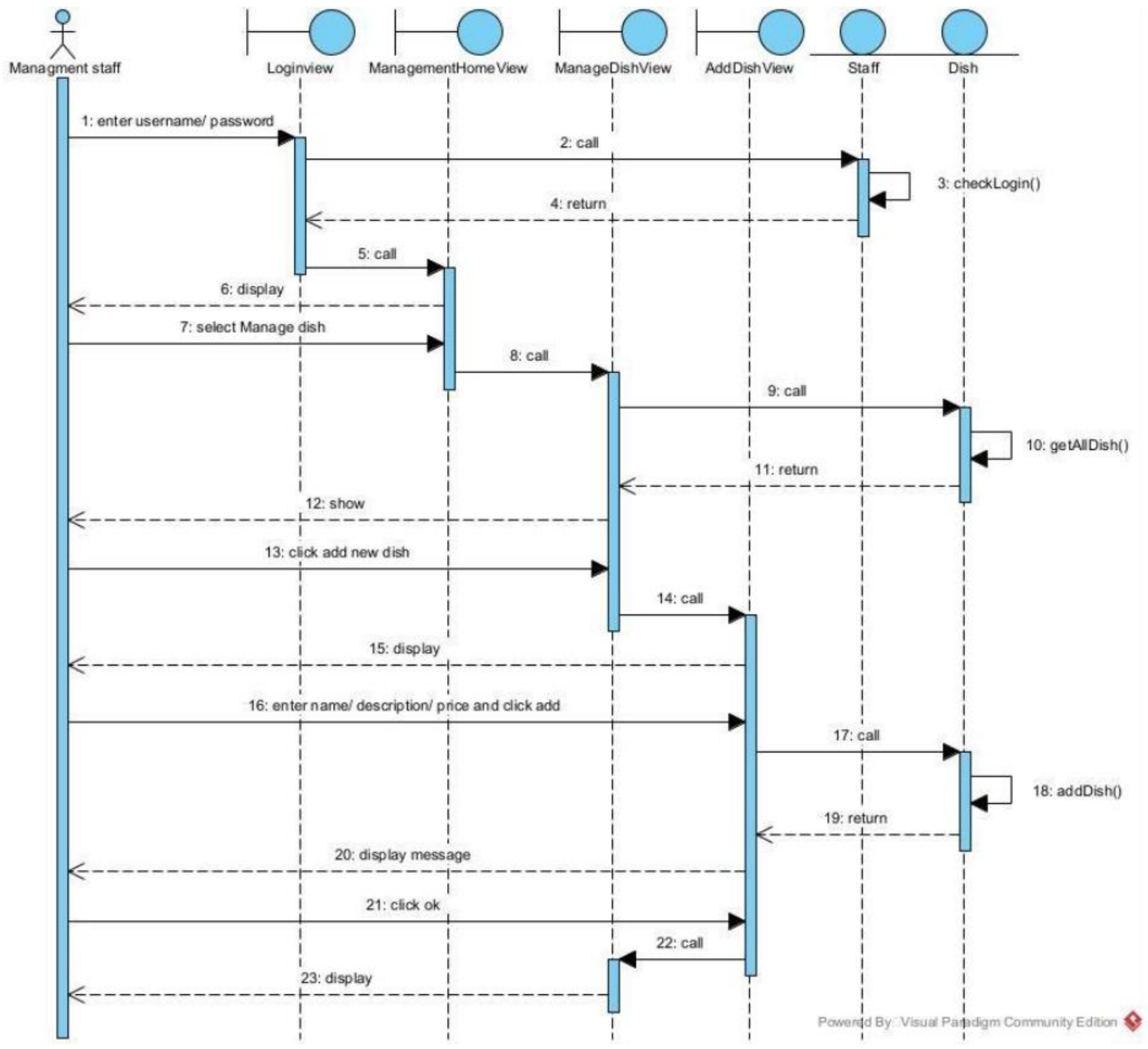
1. The management staff enters username/ password and clicks Login button
2. The class LoginView calls the class Staff to process
3. The class Staff calls the method checkLogin()
4. The class Staff returns the results to the class LoginView
5. The class LoginView calls the class ManagementHomeView
6. The class ManagementHomeView displays itself to the management staff
7. The management staff clicks on the Manage dish button
8. The class ManagementHomeView calls the class ManageDishView
9. The class ManageDishView calls the class Dish to process
10. The class Dish calls the method getAllDish()
11. The class Dish returns the result to the class ManageDishView
12. The class ManageDishView shows itself to the management staff
13. The management staff clicks on Add new disk button
14. The class ManageDishiew calls the class AddDishView
15. The class AddDishView displays itself to the management staff
16. The management staff enters dish name, description, price and click Add button
17. The class AddDishView calls the class Dish to process
18. The class Dish calls the method addDish()
19. The class Dish returns to the class AddDishView
20. The class AddDishView displays a successful message to the management staff

21. The management staff clicks OK button of the message
22. The class AddDishView calls the class ManageDishView
23. The class ManageDishView displays itself to the management staff

r. Communication diagram



s. Sequence diagram:



1. Module 2:

t. Class diagram

- Management staff selects View statistics from the main interface => need a class: ManagementHomeView:

  + View statistic button: subViewStatistic

- System displays an interface that has 3 buttons which is View dish statistics, View customer statistics, View supplier statistic => need a class: SelectStatisticView:

  + View customer statistics button: subCustomerStatistic

  + View supplier statistics button: subSupplierStatistic

  + View dish statistics button: subDishStatistic

- System shows an interface with the input of start and end time of the statistics and a Search button: need a class: CustomerStatView:

+ input of start date: inStartDate

+ input of end date: inEndDate

+ Search button: subSearch

+ Display list of orders: outsubListCustomer

- Enter start and end time of the statistics and search => need a method: getCustomerStat():

  + Input: start date, end date

  + Output: list of CustomerStat

  + Assign to class: CustomerStat

- System displays the order list interface of the customer => need a class: ListOrderView:

  + List of orders of the customer: outsubListOrder

- An order list of the customer is displayed => need a method: getListOder():

  + Input: Customer id, start date, end date

  + Output: list of orders

  + Assign to class: Order

- System displays an interface contains a list of dishes and combos that was ordered: need a class: OrderStatView:

  + A list of dished and combos: outListDishCombo

- When select an order, the system displays a list of dishes => need a method: getListDish():

  + Input: Order id

  + Output: List of Dish

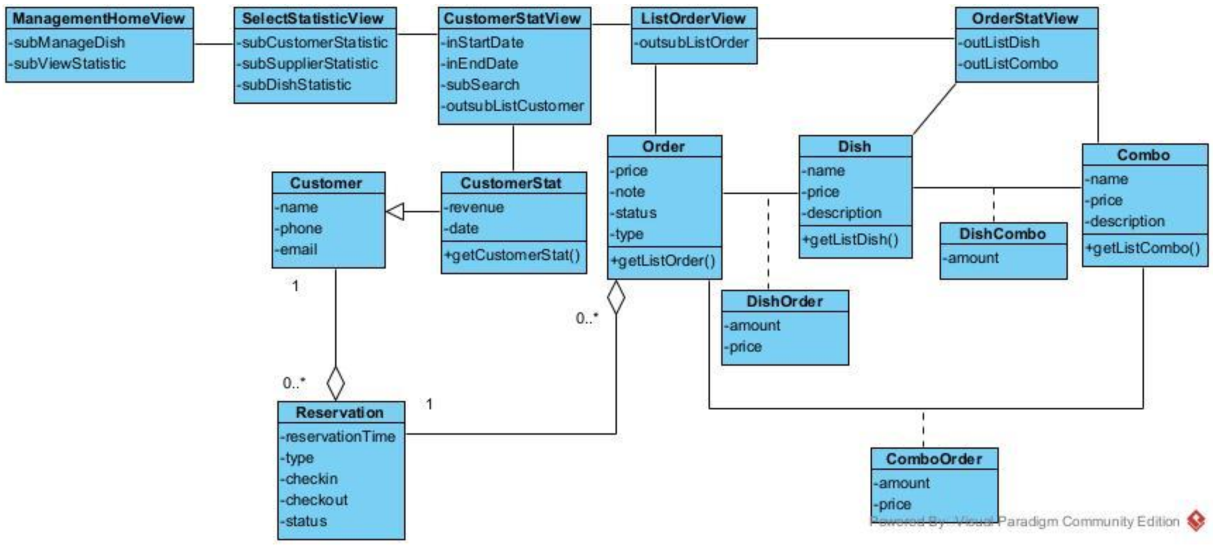
  + Assign to class: Dish

- When select an order, the system displays a list of combos=> need a method: getListCombo():

  + Input: Order id

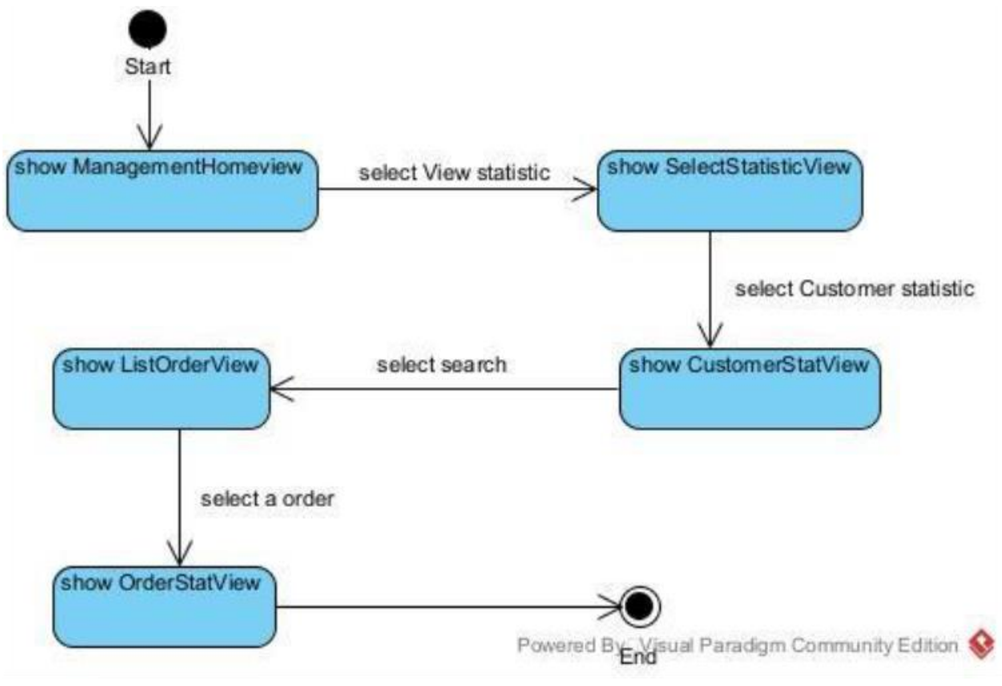
  + Output: List of Combo

  + Assign to class: Combo



u. State diagram:

- From ManagementHomeView, if choose view statistic option, the system will display SelectStatisticView
- From SelectStatisticView, if select Customer statistic button, the system will show the CustomerStatView
- From CustomerStatView, after entering start date/ end date and click search, the system will show ListOrderView
- From ListOrderView, after select an order to view, the system will show OrderStateView and finish

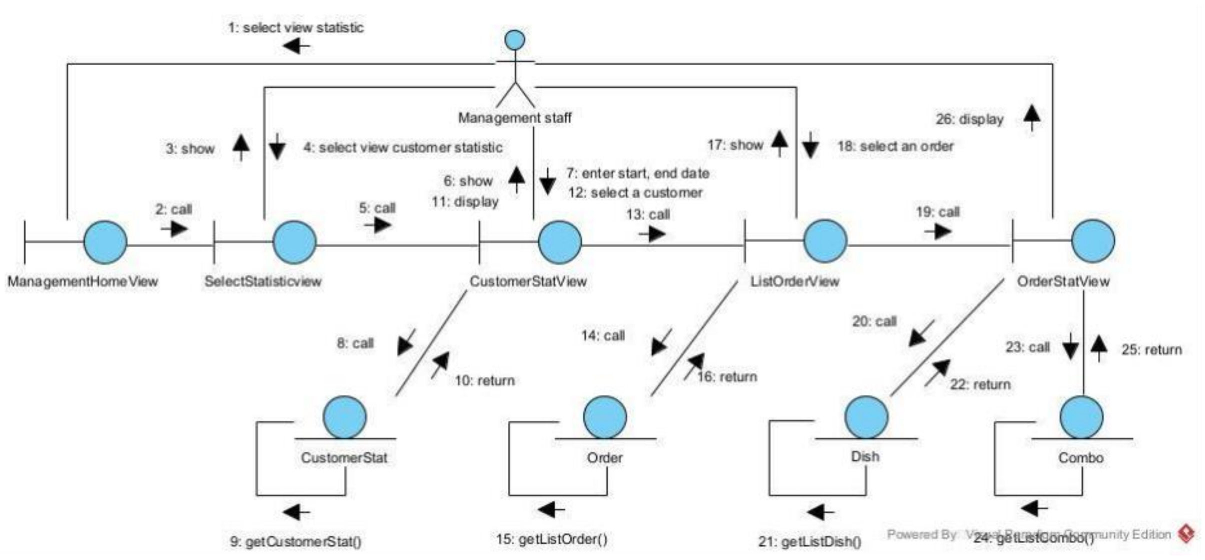


v. Scenario ver 2:

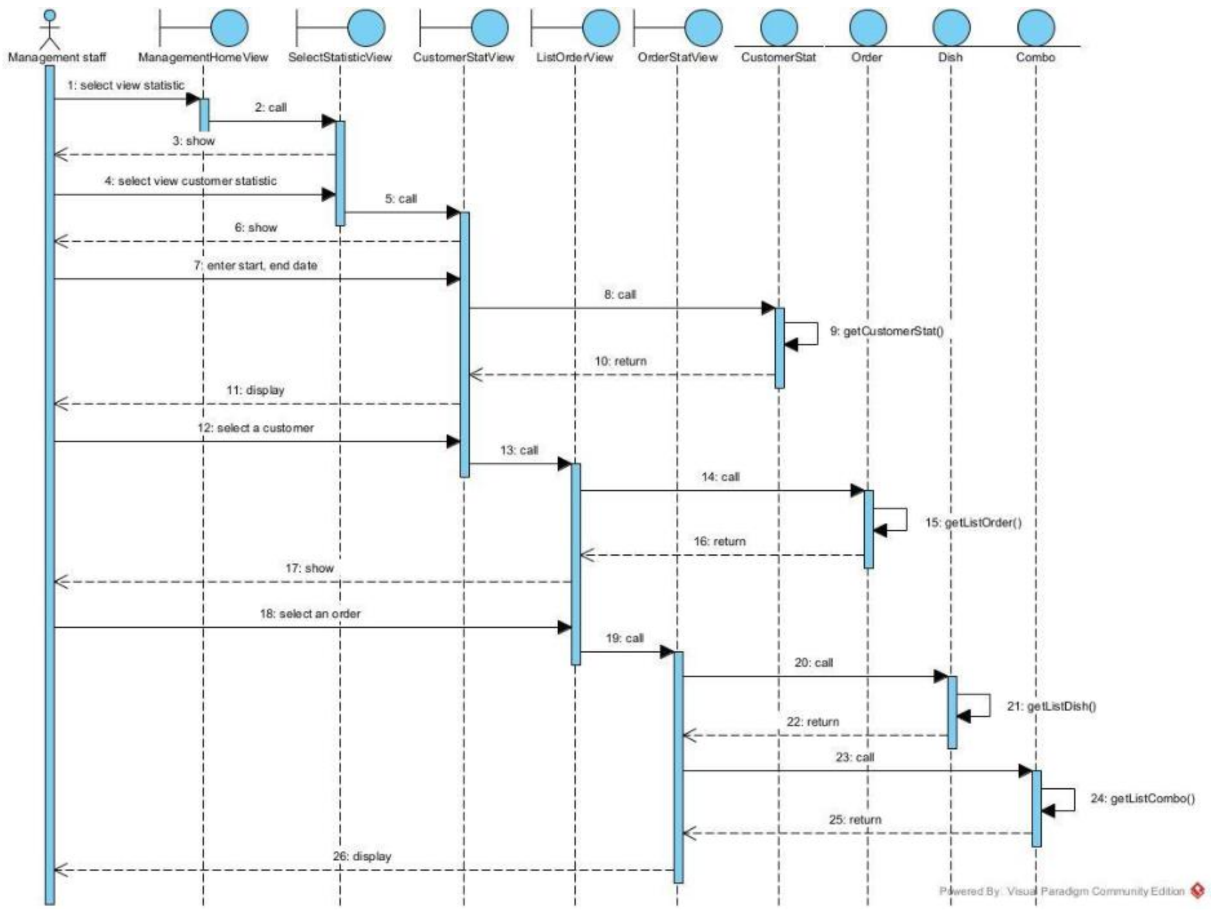
1. The management staff selects the View statistics button
2. The class ManagementHomeView calls the class SelectStatisticView
3. The class SelectStatisticView shows itself to the management staff

4. The management staff clicks on View customer statistics button
5. The class SelectStatisticView calls the class CustomerStatView
6. The class CustomerStatView shows itself to the management staff
7. The management staff enters start date, end date and click search
8. The class CustomerStatView calls the class CustomerStat to process
9. The class CustomerStat calls the method getCustomerStat()
10. The class CustomerStat returns the result to the class CustomerStatView
11. The class CustomerStatView displays the results to the management staff
12. The management staff select a customer
13. The class CustomerStatView calls the class ListOrderView
14. The class ListOrderView calls the class Order to get data
15. The class Order calls the method getListOrder()
16. The class Order returns the results to the class ListOrderView
17. The class ListOrderView shows the results to the management staff
18. The management staff selects an order
19. The class ListOrderView calls the class OrderStatView
20. The class OrderStatView calls the class Dish to get data
21. The class Dish calls the method getListDish()
22. The class Dish returns the results to the class OrderStatView
23. The class OrderStatView calls the class Combo to get data
24. The class Combo calls the method getListCombo()
25. The class Combo returns the results to the class OrderStatView
26. The class OrderStatView displays the results to the management staff

w. Communication diagram

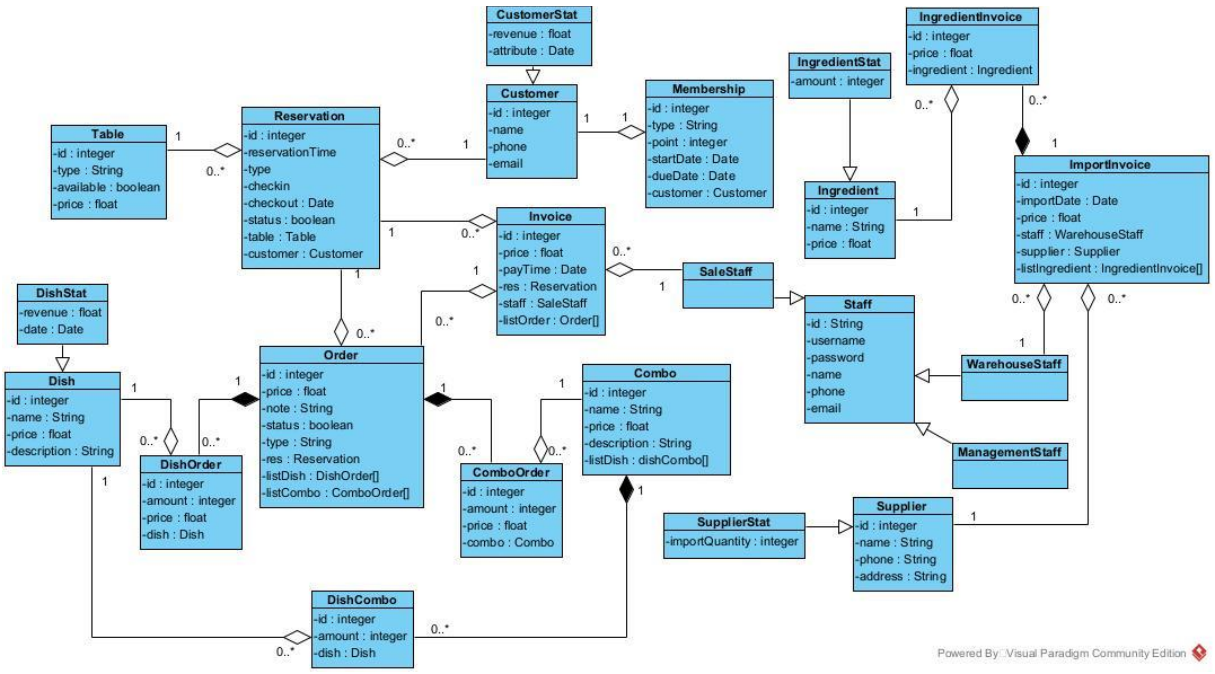


x. Sequence diagram:

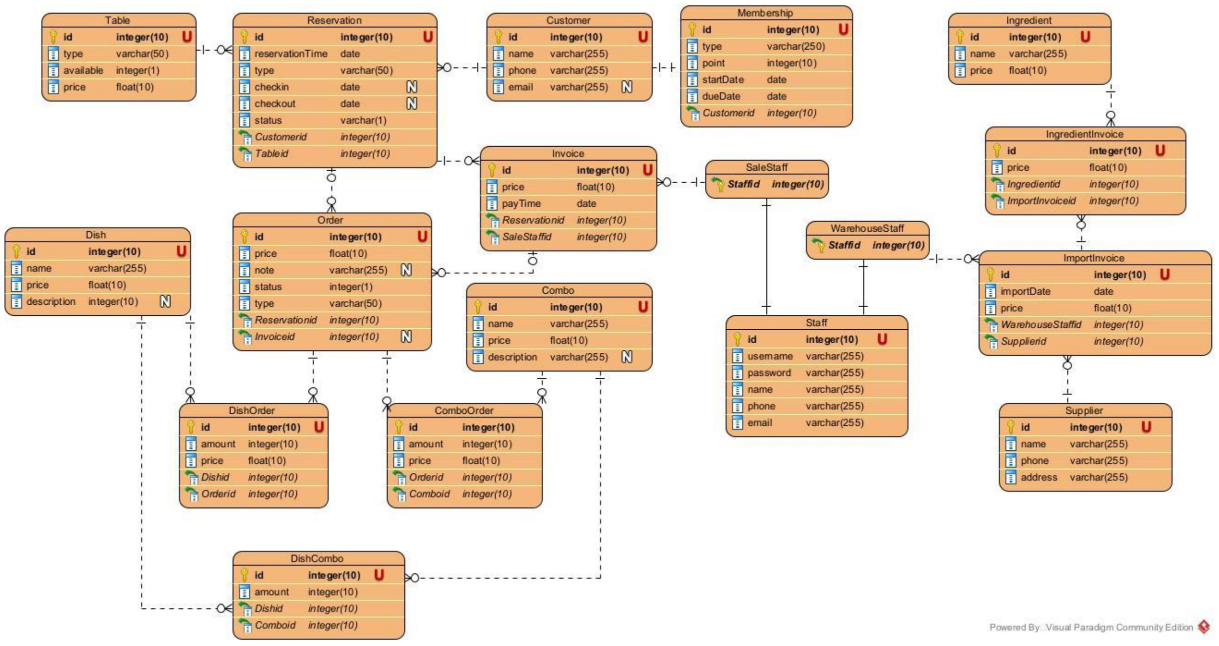


III. Design

1. Entity class diagram

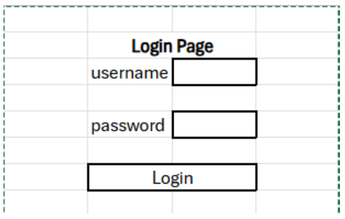


2. Database design



3. Module 1:

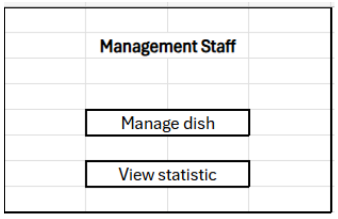
y. Class diagram


Login Page

username

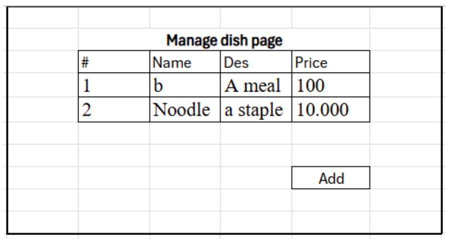
password

Login


Management Staff

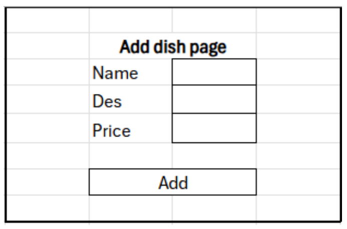
Manage dish

View statistic


Manage dish page

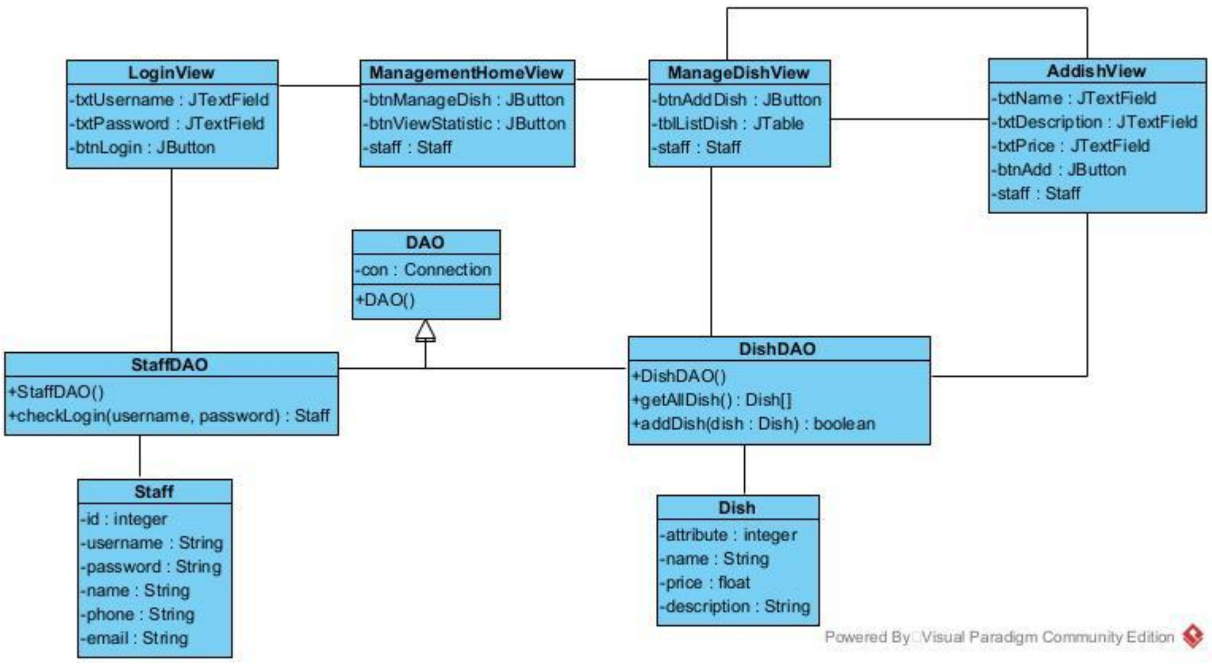
#	Name	Des	Price
1	b	A meal	100
2	Noodle	a staple	10.000

Add

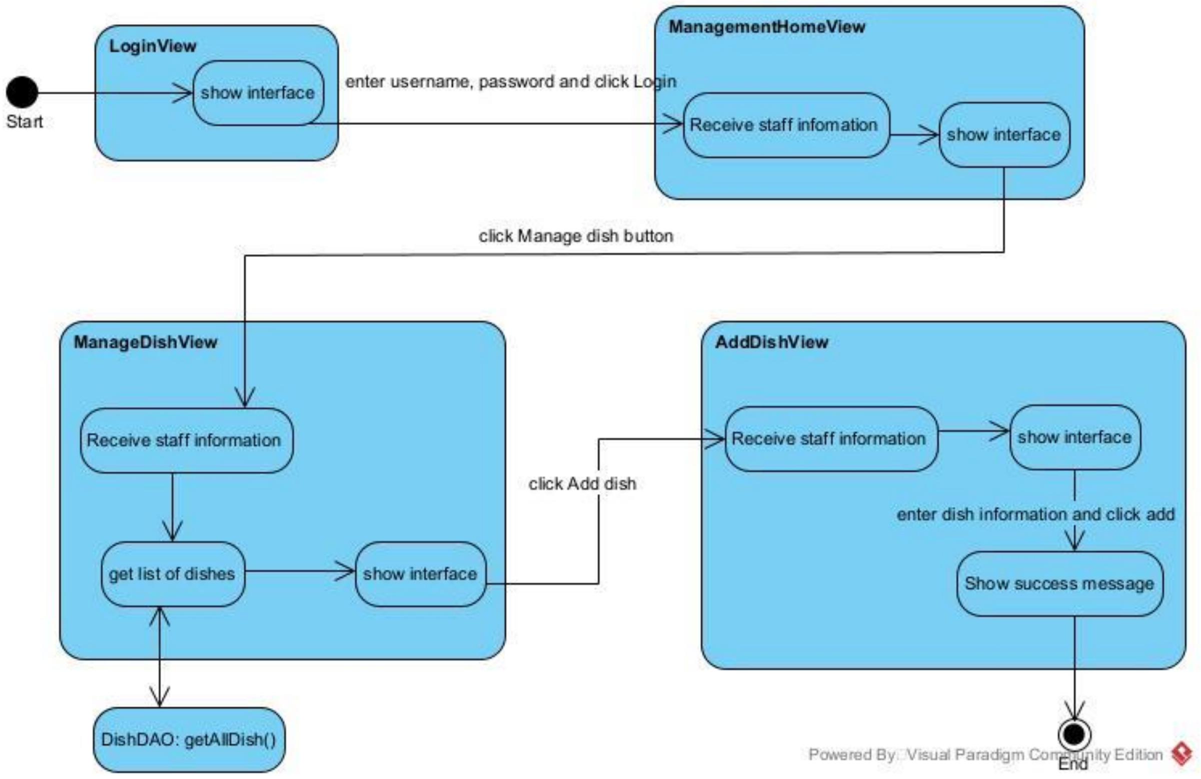

Add dish page

Name	
Des	
Price	

Add



z. Activity diagram

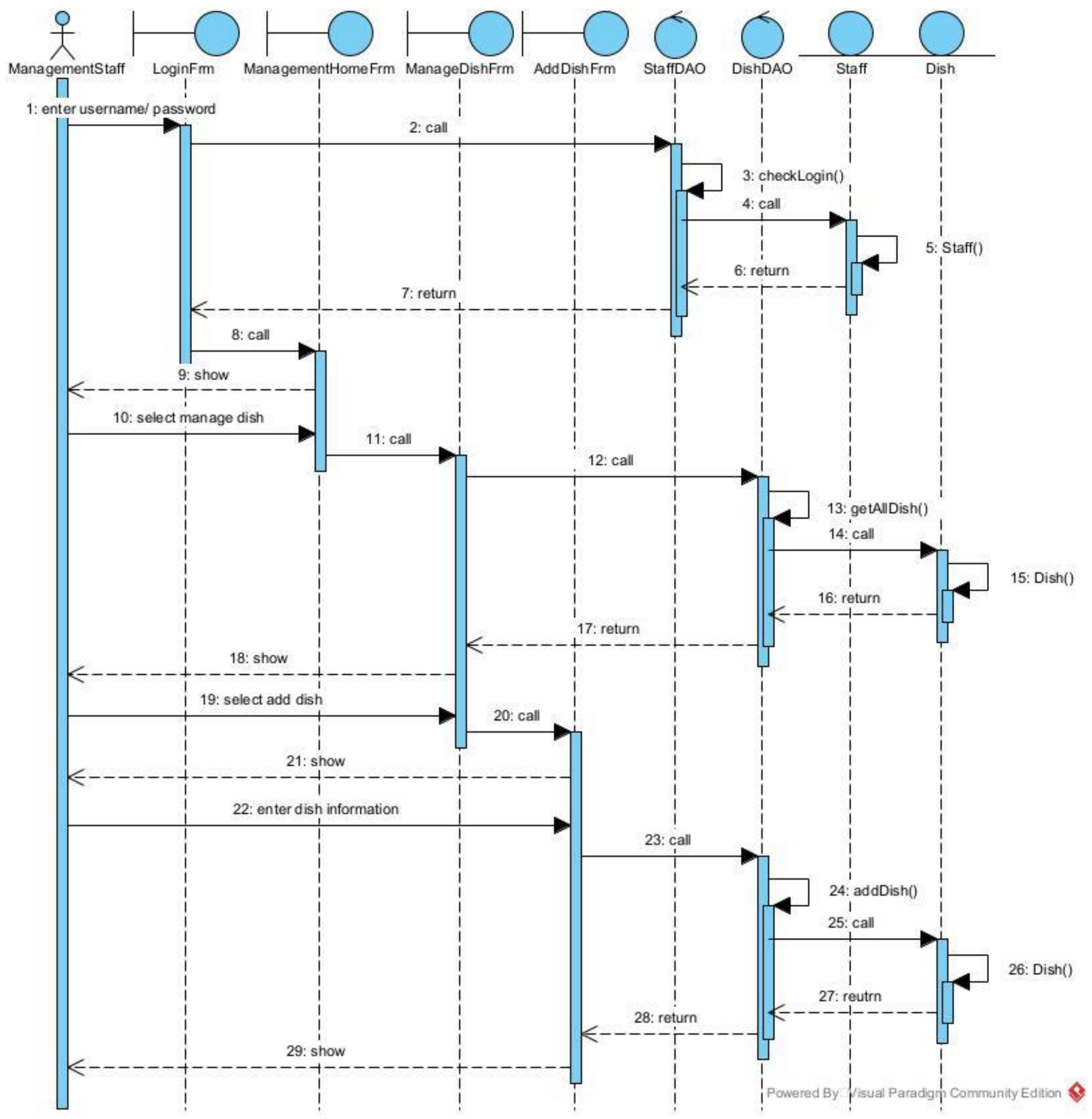


aa. Scenario v3

1. A management staff enters username, password and clicks on the login button on LoginFrm
2. The interface LoginFrm calls StaffDAO
3. The class StaffDAO calls the method checkLogin() to check login information
4. The method checkLogin() calls the class Staff
5. The class Staff packs the information to Staff object
6. The class Staff returns the object to the method checkLogin()

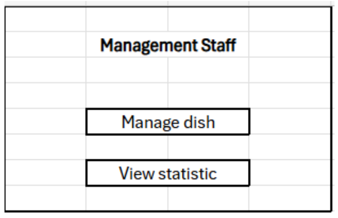
7. The method checkLogin() returns the result to the interface LoginFrm
8. The interface LoginFrm calls the interface ManagementHomeFrm
9. The interface ManagementHomeFrm shows itself to the management staff
10. The management staff selects the the manage dish button
11. The interface ManagementHomeFrm calls the interface ManageDishFrm
12. The ManageDishFrm calls the class DishDAO to get the list of all dishes
13. The class DishDAO calls the method getAllDish()
14. The method getAllDish() calls the class Dish
15. The class Dish packs each Dish object
16. The class Dish returns the list of objects to the method getAllDish()
17. The method getAllDish() returns the result to the interface ManageDishFrm
18. The interface ManageDishFrm shows itself to the management staff
19. The management staff select add dish button
20. The interface ManageDishFrm calls the interface AddDishFrm
21. The interface AddDishFrm shows itself to the management staff
22. The management staff enters the dish information they want to add and click Add button
23. The interface AddDishFrm calls the class DishDAO
24. The class DishDAO calls the method addDish()
25. The method addDish() calls the class Dish
26. The class Dish create a Dish object
27. The class Dish returns the object to the method addDish()
28. The method addDish() return the result to the interface AddDishFrm
29. The interface AddDishFrm show the successful result to the management staff

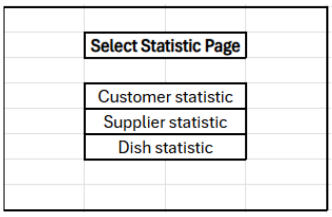
bb.Sequence diagram:

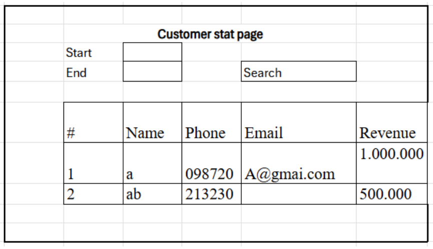


4. Module 2

cc. Class diagram





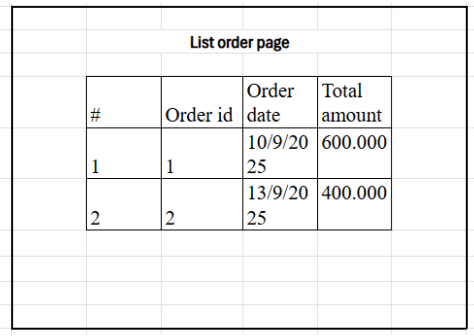

Customer stat page

Start

End

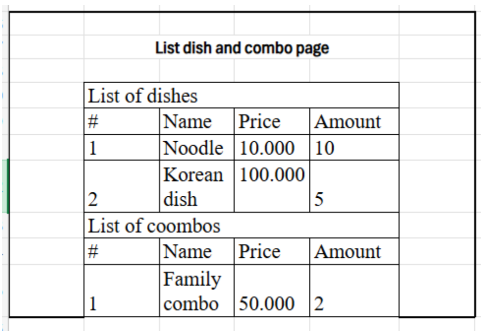
Search

#	Name	Phone	Email	Revenue
1	a	098720	A@gmai.com	1.000.000
2	ab	213230		500.000



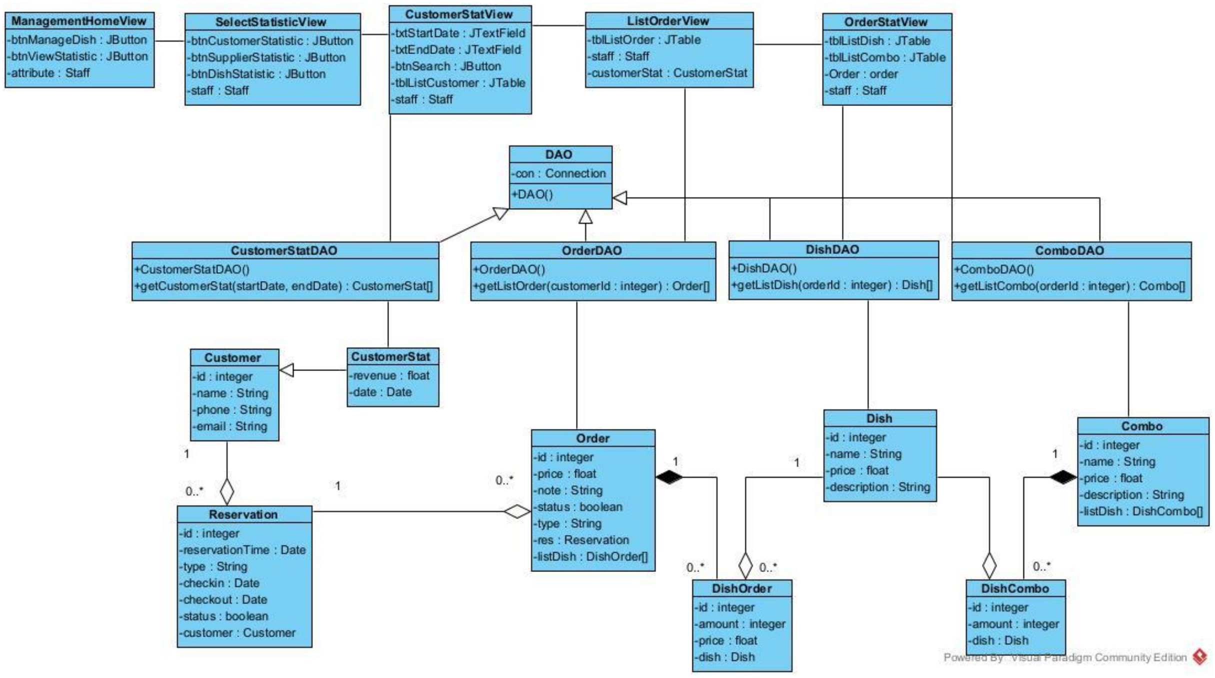
List order page

#	Order id	Order date	Total amount
1	1	10/9/2025	600.000
2	2	13/9/2025	400.000

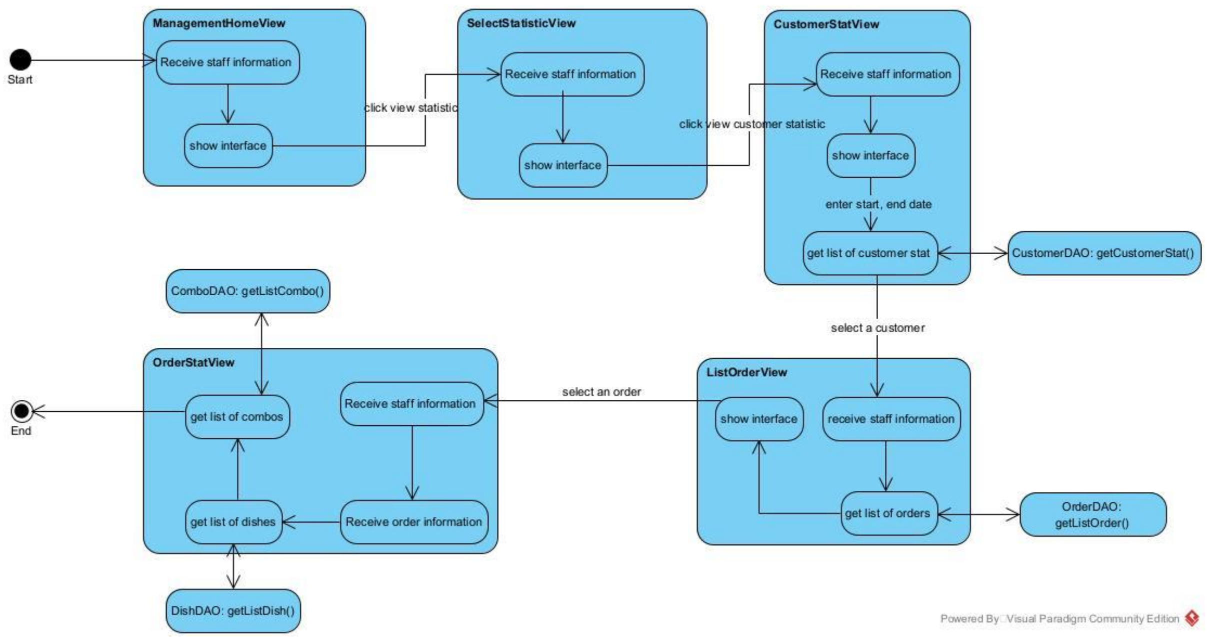


List dish and combo page

List of dishes			
#	Name	Price	Amount
1	Noodle	10.000	10
2	Korean dish	100.000	5
List of coombos			
#	Name	Price	Amount
1	Family combo	50.000	2



dd.Activity diagram

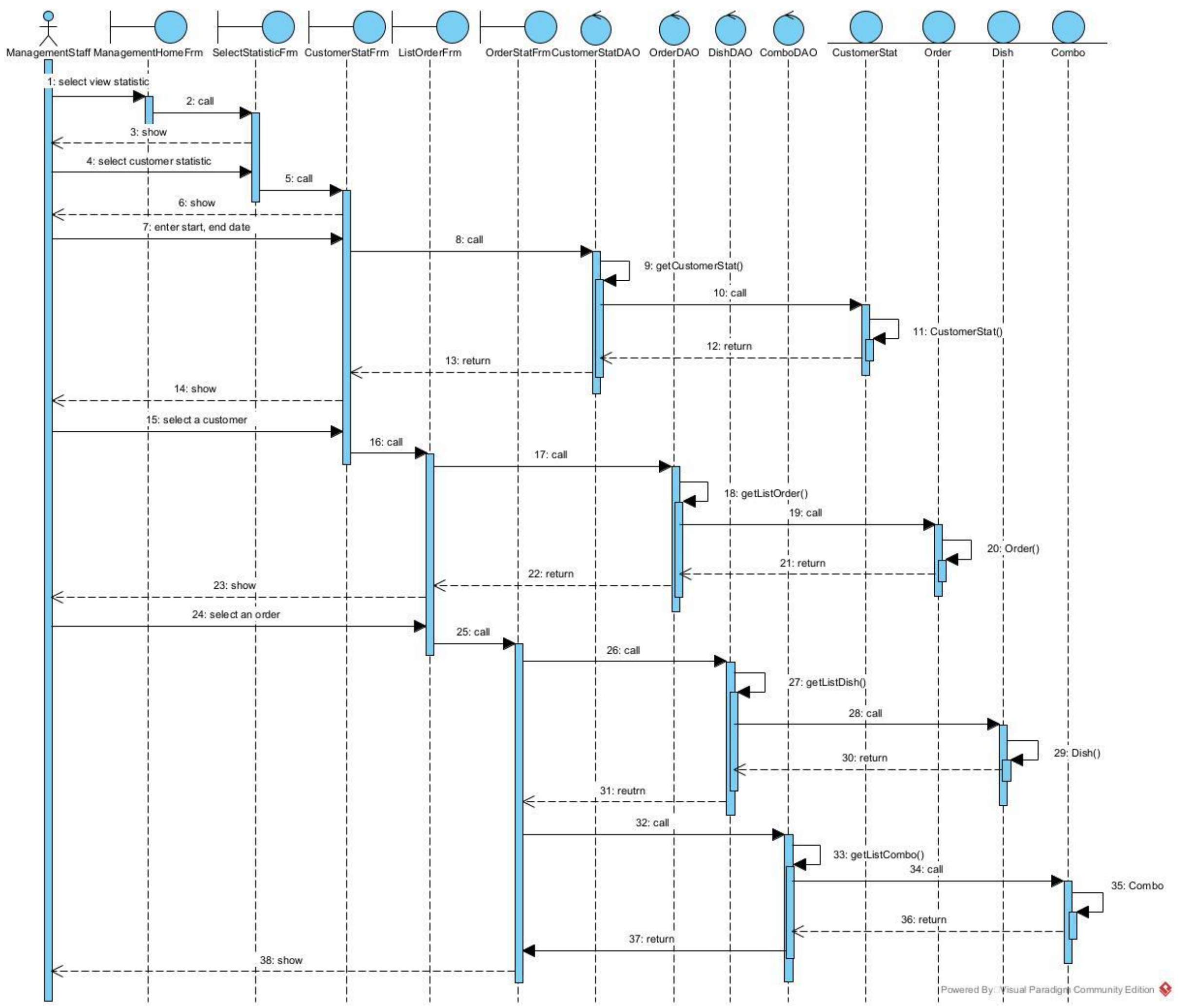


ee. Scenario v3

1. The management staff selects the view statistic button in the interface ManagementHomeFrm

2. The interface ManagementHomeFrm calls the interface SelectStatisticFrm
3. The interface SelectStatisticFrm shows itself to the management staff
4. The management staff selects the customer statistic button
5. The interface SelectStatisticFrm calls the interface CustomerStatFrm
6. The interface CustomerStatFrm shows itself to the management staff
7. The management staff enters start, end date and click search
8. The interface CustomerStatFrm calls the class CustomerStatDAO
9. The class CustomerStatDAO calls the method getCustomerStat()
10. The method getCustomerStat() calls the class CustomerStat
11. The class CustomerStat packs the information into an object
12. The class CustomerStat returns the object to the method getCustomerStat()
13. The method getCustomerStat() returns the result to the interface CustomerStatFrm
14. The interface CustomerStatFrm shows the result to the management staff
15. The management staff selects a customer in the list
16. The interface CustomerStatFrm calls the interface ListOrderFrm
17. The interface ListOrderFrm calls the class OrderDAO
18. The class OrderDAO calls the method getListOrder()
19. The method getListOrder() calls the class Order
20. The class Order packs each result into an object
21. The class Order returns the objects to the method getListOrder()
22. The method getListOrder() returns the result to the interface ListOrderFrm
23. The interface ListOrderFrm shows the result to the management staff
24. The management staff selects an order
25. The interface ListOrderFrm calls the interface OrderStatFrm
26. The interface OrderStatFrm calls the class DishDAO
27. The class DishDAO calls the method getListDish()
28. The method getListDish() calls the class Dish
29. The class Dish packs each result into an object
30. The class Dish returns the objects to the method getListDish()
31. The method getListDish() returns the result to the interface OrderStatFrm
32. The interface OrderStatFrmcalls the class ComboDAO
33. The class ComboDAO calls the method getListCombo()
34. The method getListCombo() calls the class Combo
35. The class Combo packs each result into an object
36. The class Combo returns the objects to the method getListCombo()
37. The method getListCombo() returns the result to the interface OrderStatFrm
38. The interface OrderStatFrm shows the result to the management staff

ff. Sequence diagram


ManagementStaff
ManagementHomeFrm
SelectStatisticFrm
CustomerStatFrm
ListOrderFrm
OrderStatFrm
CustomerStatDAO
OrderDAO
DishDAO
ComboDAO
CustomerStat
Order
Dish
Combo
1: select view statistic
2: call
3: show
4: select customer statistic
5: call
6: show
7: enter start, end date
8: call
9: getCustomerStat()
10: call
11: CustomerStat()
12: return
13: return
14: show
15: select a customer
16: call
17: call
18: getListOrder()
19: call
20: Order()
21: return
22: return
23: show
24: select an order
25: call
26: call
27: getListDish()
28: call
29: Dish()
30: return
31: reutrn
32: call
33: getListCombo()
34: call
35: Combo
36: return
37: return
38: show
Powered By: Visual Paradigm Community Edition