Name: Aravind Anand
Student ID: 030821269
Title: Assignment Unit 1

Consider the following system:
"Online Food Delivery System"

This system will help customers to order food online from different local restaurants using a mobile app client or website (A similar popular service is Doordash). Users can search the local restaurants and add items from the restaurant menus to their carts. After confirming the order, the app will charge the amount due from the users using its online payment system. The app will show you an estimated delivery time and will send you notifications when your order goes through different steps such as whenever it is picked up from the restaurants or it is dropped off in front of your door. The system also provides services for the business owners as well as the Drivers. For instance, business owners can apply to join this platform, edit their menu items and other information such as working hours. Drivers also interact with the system, for instance they can apply to join this service or mark a delivery as done in the app. There are also some services for the system admin such as defining the service fee and other standard costs or approving the new restaurants to join this platform or new Driver requests.

Question 1: Write the following use case scenarios in the format you learned in the class,

i. Users putting an order (fully dressed format)
ii. Restaurants fulfilling an order (fully dressed format)
iii. Drivers picking and delivering their orders (brief/casual format)
iv. System admin approving new restaurant requests (brief/casual format)

---

1. Users putting an order:

Use Case: UC01: Placing Order

Primary Actor: Customer

Stakeholders and Interests:

- Customer: Requires a varieties of restaurant options to choose from, different options available with an item, should not encounter any technical issues or payment errors, must get the order delivered with best turnaround time and regular updates throughout the delivery process.
- System Admin: Needs to maintain accurate records of the customers, business owners, delivery persons. Keep the item list up to date and should process the applicable service charges from business owners.
- Business Owner: Fulfilling an order for which payment has been successfully processed. Manage the updated menu and options available with the items in the application. Make the order available within the posted pickup time. Notify and discuss with the customer on any updates performed to the orders.

- Government Tax Agencies: Collect the appropriate taxes from the customers and business owners for each sale performed and may include multiple divisions such as department, state and national.
- Payment Authorization Service: Perform a secure payment authorization with appropriate protocol and requires accurate accounting of the transaction processing with essential logs including duration and methods of transactions.

Preconditions: Customer is successfully registered, authenticated, and logged in to the application

Success Guarantee (Postconditions): Customer placed an order. Sale price calculated including item prices, delivery charges, government taxes and payment gateway charges. Receipt of the order is generated. Payment details are recorded for the transaction performed. Delivery information provided to the customer. The order details are forwarded to the restaurant.

Main Success Scenario (or Basic Flow):

1. Customer sign in to the application.
2. Customer checks the restaurants available to make a food purchase.
3. Choose a restaurant and access the updated menu of the restaurant.
4. Customer chooses an item, review the calories/dietary instruction and options available for the item.
5. Customer mention the quantity of the item and add the item to the cart

Customer repeats steps 4-5 until they are ready to checkout.

6. Review the cart with the selected items.
7. Checkout to place an order.
8. System will generate the total with delivery fare, taxes and other charges for the selected items, and navigate to payment options.
9. Customer chooses the preferred payment method and completes the payment.
10. On successful payment, the order will be placed to the restaurant and a delivery person will be assigned by the system to deliver the food to the customer.

Extensions (or Alternative Flows):

*a System errors:

At any time, to support successful placement of order, the system must be able to capture the information from any state and events can be recovered from any scenario.

1. Customer can restart the application and access the previous recovered state of the order placement.
2. If the application cannot spawn from the previous recovered state:
   - System can notify the customer on error occurred and request to restart the ordering process.

- System should refund the amount deducted (if any) back to the customer's payment account.

1.a. Login Issues:

1. Customer can perform a password reset with proper authentication and login to the application.
2. Customer can recover their account if password or other account information is not accessible by answering to the security questions or by identifying the details of the previous orders.

7.a. Item unavailability:

Customer selected items might become unavailable during the order placement:

1. Customer must be notified with the updated menu of the restaurant.
2. An alternative suggestion from the available items to replace the unavailable items can be provided to the customer.

9.a. Payment Failure:

The system fails to catch a successful payment process due to any technical error with the application or the payment firm.

1. Customers must be notified about the unsuccessful payment, and they can perform the payment process again. The system should navigate to the previous state and request the customer to try again.
2. Customers can contact the application support if there was any amount deduction from the payment account and request for refund of the amount.

Special Requirements:

- The application requires internet connectivity across stakeholders to record live status and update successful ordering and delivery.
- The system must be able to recover from any state as it involves live interactions between customer, restaurant, and delivery person.
- Payment processing with minimal response time over 95% of the time.
- Application support across different platforms and online support services to handle issues faced by the actors of the applications.
- Multi-language support for essential and wide usage of the application services.

Technology and Data Variations List:

9a. Customer can choose different payment option and the system must be able to navigate among other payment applications and successfully capture the transaction.

9b. Various domains involved in the transaction must be notified on the amount deduction, duration, and mode of payment, as customer would have opted for security notification on transaction occurring with their accounts.

Frequency of Occurrence: Continuous and one of the primary operations of the application

Open Issues:

- Increasing the customer satisfaction: The time taken for delivering an order depends on the coordination of all the stakeholders and delay in order delivery might reflect customer dissatisfaction and affect the user count of the application.

- Tax regulation followed by different cities varies and processing orders based on location involves accurate capturing of customer location and processing the delivery charges.

- Some Remote locations might have less restaurants and the varieties available to the customer will be limited. Networks of restaurants in the remote location should increase with necessary delivery scale.

2. Restaurants fulfilling an order:

Use Case: UC02: Fulfilling Order

Primary Actor: Business Owner (Restaurant)

Stakeholders and Interests:

- Business Owner: Fulfilling an order for which payment has been successfully processed. Manage the updated menu and options available with the items in the application. Make the order available within the posted pickup time. Notify and discuss with the customer any updates performed to the orders.
- Customer: Requires a variety of restaurant options to choose from, different options available with an item, should not encounter any technical issues or payment errors, must get the order delivered with best turnaround time and regular updates throughout the delivery process.
- System Admin: Needs to maintain accurate records of the customers, business owners, and delivery persons. Keep the item list up to date and should process the applicable service charges from business owners.
- Government Tax Agencies: Collect the appropriate taxes from the customers and business owners for each sale performed and may include multiple divisions such as department, state and national.
- Payment Authorization Service: Perform a secure payment authorization with appropriate protocol and requires an accurate accounting of the transaction processing with essential logs including duration and methods of transactions.

- Delivery Person: Requires checking the status of the order to be delivered. Reach the restaurant location prior to the pickup time. Pick up the order with proper packaging and deliver the order to the customer's location.

Preconditions: Business Owner is successfully registered, authenticated, and logged in to the application to receive orders to process.

Success Guarantee (Postconditions): Business Owner processed the order and kept it ready for pickup. Order completion status notified to the customer and delivery person. Payment for any additional modification on the items calculated and collected from the customer. Delivery person arrives and pick up the order.

Main Success Scenario (or Basic Flow):

1. Business owner logins to the application.
2. Update the restaurant status and menu to receive and serve the orders.
3. New Orders with successful payment processing will be updated for fulfillment.
4. Check the items and options selected by the customer in the order.
5. Check the availability of the items and forward the order the kitchen desk to process the items.
6. When an item requires modification, check with customer on replacement for the items and process payment for any additional options.
7. Move the order from new orders section to in-progress order section with the expected pickup time.
8. Update the customer on the processing states of the order.
9. Share the order details and delivery instruction to the delivery person network.
10. Once a driver accepts the delivery task, share the live status and pickup time of the order to the driver.
11. Process the order and keep it ready within the pickup time.
12. Check for the driver arrival to the restaurant.
13. Handover the processed order to the driver for delivery at the customer location.

Extensions (or Alternative Flows):

*a System errors:

At any time, to support successful fulfillment of order, the system must be able to capture the information from any state and events can be recovered from any scenario.

1. Business Owner can restart the application and access the previous recovered state of the order placement.
2. If the application cannot spawn from the previous recovered state:
   - System can notify the business owner on error occurred and request to restart the order acceptance process and handle the consequences of loss.
   - System should refund the amount deducted (if any) back to the customer’s payment account.

1.a. Login Issues:

1. Customer can perform a password reset with proper authentication and login to the application.
2. Customer can recover their account if password or other account information is not accessible by answering to the security questions or by identifying the details of the previous orders.

5.a. Unavailability of Item:

1. Business owner should contact the customer to choose an item as replacement which is equal or less than the unavailable item value.
2. Restaurant can charge on the additional options opted by the customer for the new item or for fulfilling the options selected by the customer for any item in the order.

5.b. Payment Failure:

The system fails to catch a successful payment process due to any technical error with the application or the payment firm.

1. Customers must be notified about the unsuccessful payment, and they can perform the payment process again. The system should navigate to the previous state and request the customer to try again.
2. Customers can contact the application support if there was any amount deduction from the payment account and request for refund of the amount.

9.a. Unavailability of Delivery Person:

1. Business Partner should keep trying to find a driver to deliver the order to the customer and keep the customer posted on the delay in delivery due to unavailability of driver.
2. The restaurant should cancel the order and process a refund of the payment processed by the customer and the system admin should work with the Business Owner to manage the loss and avoid similar situation in future.

Special Requirements:

- The application requires internet connectivity across stakeholders to record live status and update successful ordering and delivery.
- The application should maintain the information of the business owner and update the important details such of closing time of the restaurant, average delivery time for the restaurant and ratings of the restaurant for better user experience.
- The system must be able to recover from any state as it involves live interactions between customer, restaurant, and delivery person.
- Payment processing with minimal response time over 95% of the time.
- Application support across different platforms and online support services to handle issues faced by the actors of the applications.
- Multi-language support for essential and wide usage of the application services

Technology and Data Variations List:

5a. Business owner must be provided with the option to share the order details to the kitchen/food packing department for essential fulfillment of the order and quacking processing of the items in the list.

6a. Customer can choose different payment option and the system must be able to navigate among other payment applications and successfully capture the transaction.

6b. Various domains involved in the transaction must be notified on the amount deduction, duration, and mode of payment, as customer would have opted for security notification on transaction occurring with their accounts.

Frequency of Occurrence: Continuous and one of the primary operations of the application.

Open Issues:

- Highly volatile food market throws the challenge of estimating the market prices and finding a right strategy for pricing.

- What packaging techniques followed by the restaurants in proper delivery of the food?

- Adhering food quality standards: delivering foods to customer location far from the restaurants while maintaining the quality of the food is highly challenging

3. Drivers picking and delivering their orders:

ID	**UC03**
Name	Drivers Delivering Orders
Primary Actor	Delivery Driver
Description	Delivery person (Driver) will pick up the order from the restaurant and delivers it to the customer
Pre-conditions	Driver must be authenticated by the system and the Driver should check the required items such as gas, phone charge and hot bags to pick up the food. Driver must be connected with the system throughout the delivery process. The system provides the delivery instructions to the Driver and the Driver decides to take the delivery task.

Event flow (Main Success Scenario)	1. Driver login to the system and update the availability for the delivery task. 2. Driver will be provided with the delivery instructions including the pickup location, order availability time for pickup, drop location, overall distance to be covered and delivery charges to be received. 3. Driver accepts the delivery task. 4. The expected time of delivery will be updated and directions to the Restaurant will be shared via the selected navigation method. 5. Once Driver reaches the restaurant, they should confirm their arrival in the application which will be updated to the customer and the restaurant. 6. Driver should pick up the order if it is ready. While picking up, Driver should confirm the order details and items to be picked up. 7. Driver will be navigated to the customer's location. 8. Driver arrives at the customer location and updates that in the system. 9. Driver delivers the order to the customer. 10. Driver completes the delivery task and gets the delivery amount added to the wallet.
Extensions (or Alternative Flows):	3.a. When Driver declines the task, the reason for declining should be furnished and Driver will be notified of the effects on their profile. 4.a. Driver could also accept any additional delivery through the same route and when accepted, the order details will be added to the navigation route. 5.a. Driver may need to wait for the order to get ready for pickup and in that case, the Driver must update the waiting reason in the application. 9.a. Driver may be instructed by the customer to place the order at the front door or any desired location. Driver should place the order in that location and upload a picture of the order in that location to complete the delivery task.
Extension Points	This use case extends from the UC02 – **Fulfilling Order**
Frequency of Occurrence:	Could be nearly continuous
Post-condition	Driver successfully delivers the food items to Customer.

4. System admin approving new restaurant requests

ID	**UC04**
Name	Approving new restaurant
Primary Actor	System Admin
Description	The system admin should review the details of the newly registered restaurant and approve to conduct business via the application
Pre-conditions	System admin must be logged in to the application to review the request and must be provided with the access to add the restaurant.

Event flow (Main Success Scenario)	1. System admin Login to the application. 2. The requests to be approved by the admin will be available. 3. Choose a new restaurant request. 4. Review the details of the restaurant. 5. Validate the information provided by the restaurant based on the guidelines of the application. 6. Approve the restaurant request and generate a Merchant Id for the restaurant 7. Notify the restaurant on the successful addition and next steps to conduct the business in the app. 8. Repeat the procedure for other new restaurant requests. 9. Logout from the application
Extensions (or Alternative Flows):	5.a. Admin might require any additional information to validate the new restaurant. 5.b. The restaurant may not qualify to conduct a business in the application.
Extension Points	This use case extends to the UC02 – **Fulfilling Order**. To fulfill an order, the restaurant must be successfully added to the application by the system admin.
Frequency of Occurrence:	Could occur in regular intervals - when a new restaurant applies to conduct business
Post-condition	System admin successfully approves a restaurant’s request to carry-on online food delivery through the application

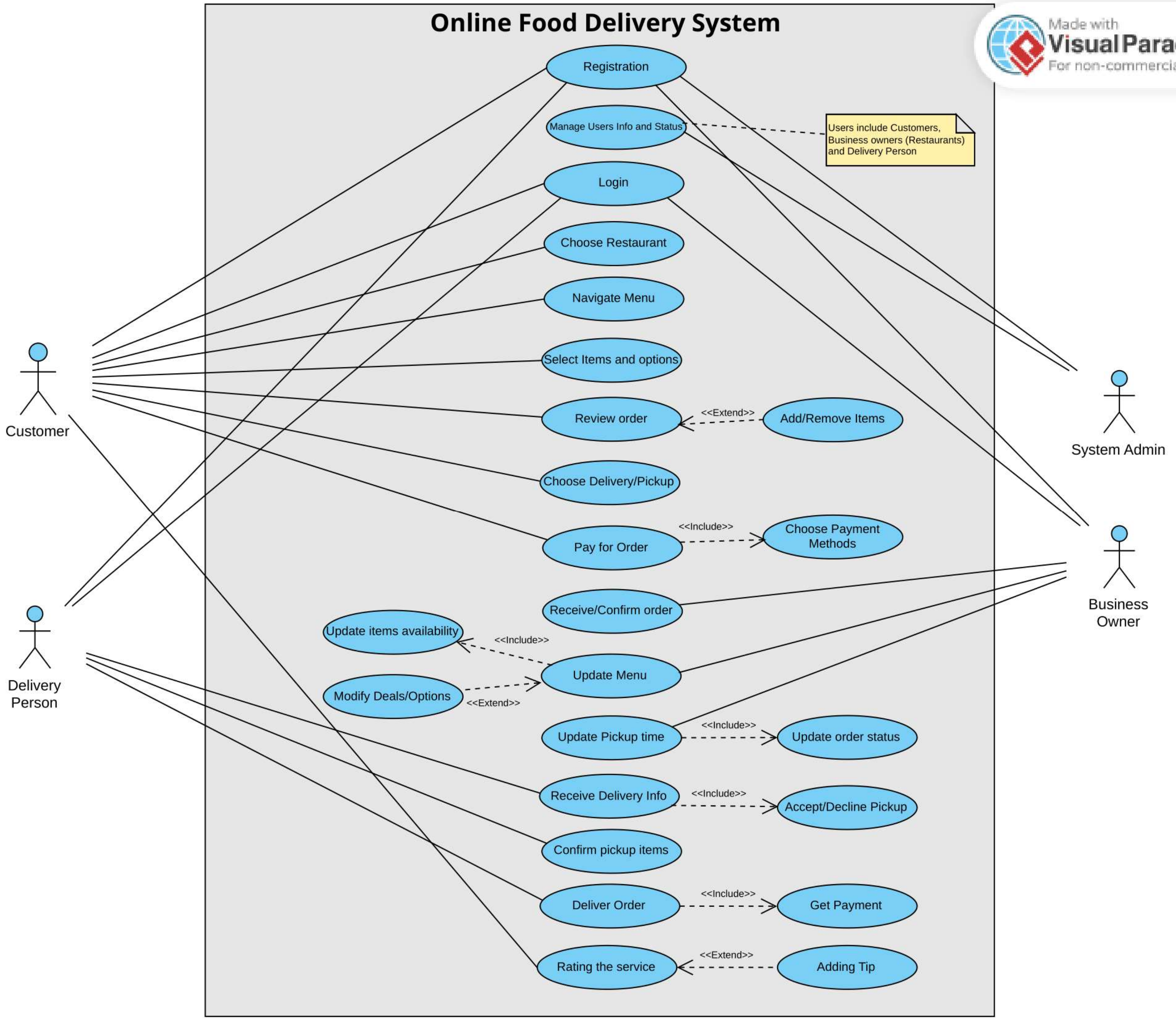
Question 2: Mention two non-functional requirements (FURPS+ model) for this system.

Below are the two Non-Functional (NF) Requirements for the system:

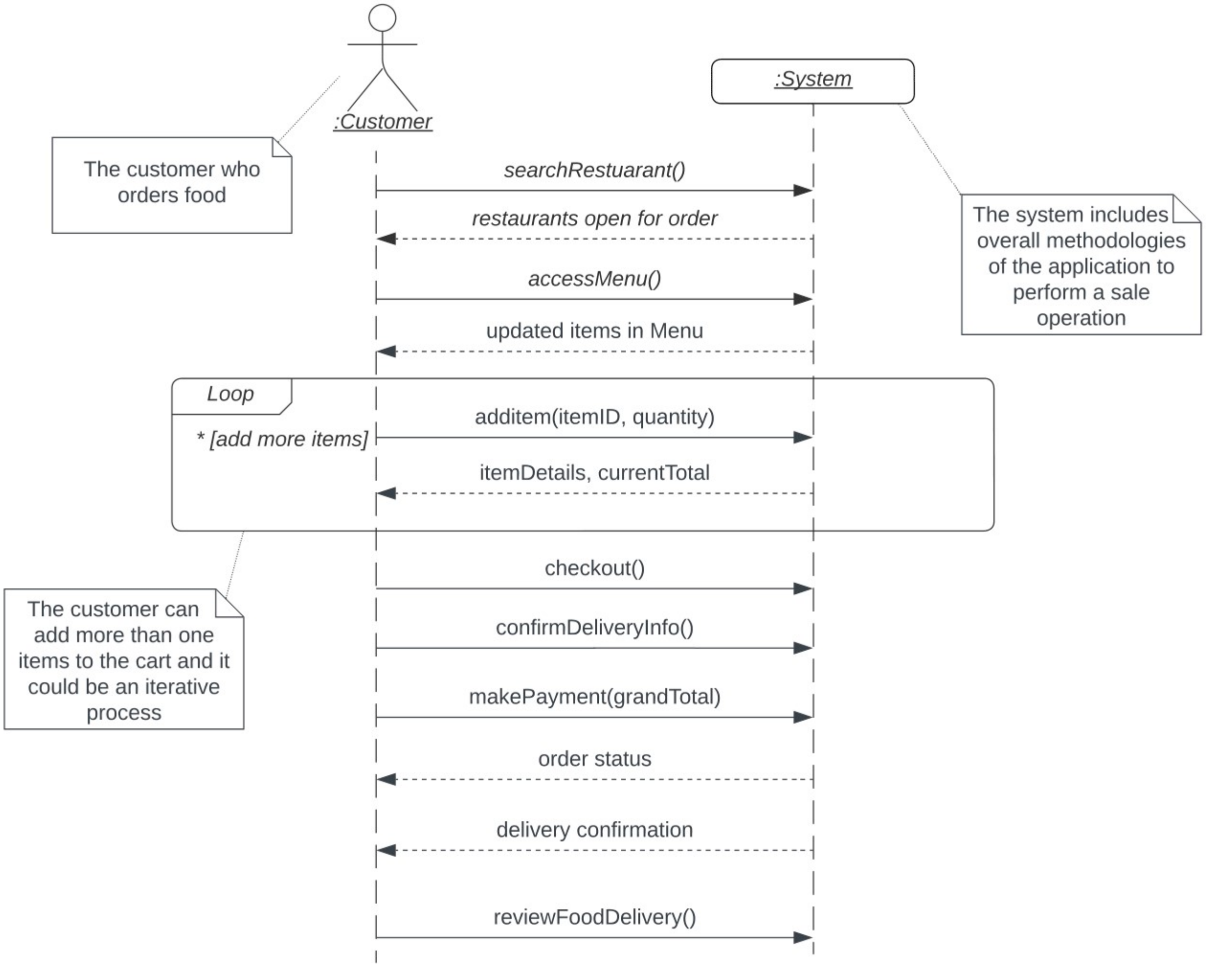
ID	NF001
Name	Keeping the User Account Protected
Type of Requirement	Security
Criteria	The system should protect the user account information as it contains confidential information such as payment details, address, and other personal information.
Test/Functionality	When the system detects a user logging into to their account, an email notification will be sent and if the user doesn't recognize the device or find any suspicious activity, they can modify the password and logout from all logged in devices. Also, the user can opt for two step authentication which includes an OTP verification while logging into their account in a device.

ID	NF002
Name	Software Installation and Behavior
Type of Requirement	Usability
Criteria	The users should be able to access the software intuitively in devices with different host platforms such as Mobile: Android, IOS or Webpage accessed from different browsers.
Test/Functionality	The software must be tested in different platforms, operating systems, different display aspect ratios and all the functionalities should behave as per the design. Proper documentation on the behavior and operating instructions of the system in different platforms must be shared to the users with up-to-date versions of operating systems/browsers supported by the system for best user experience.

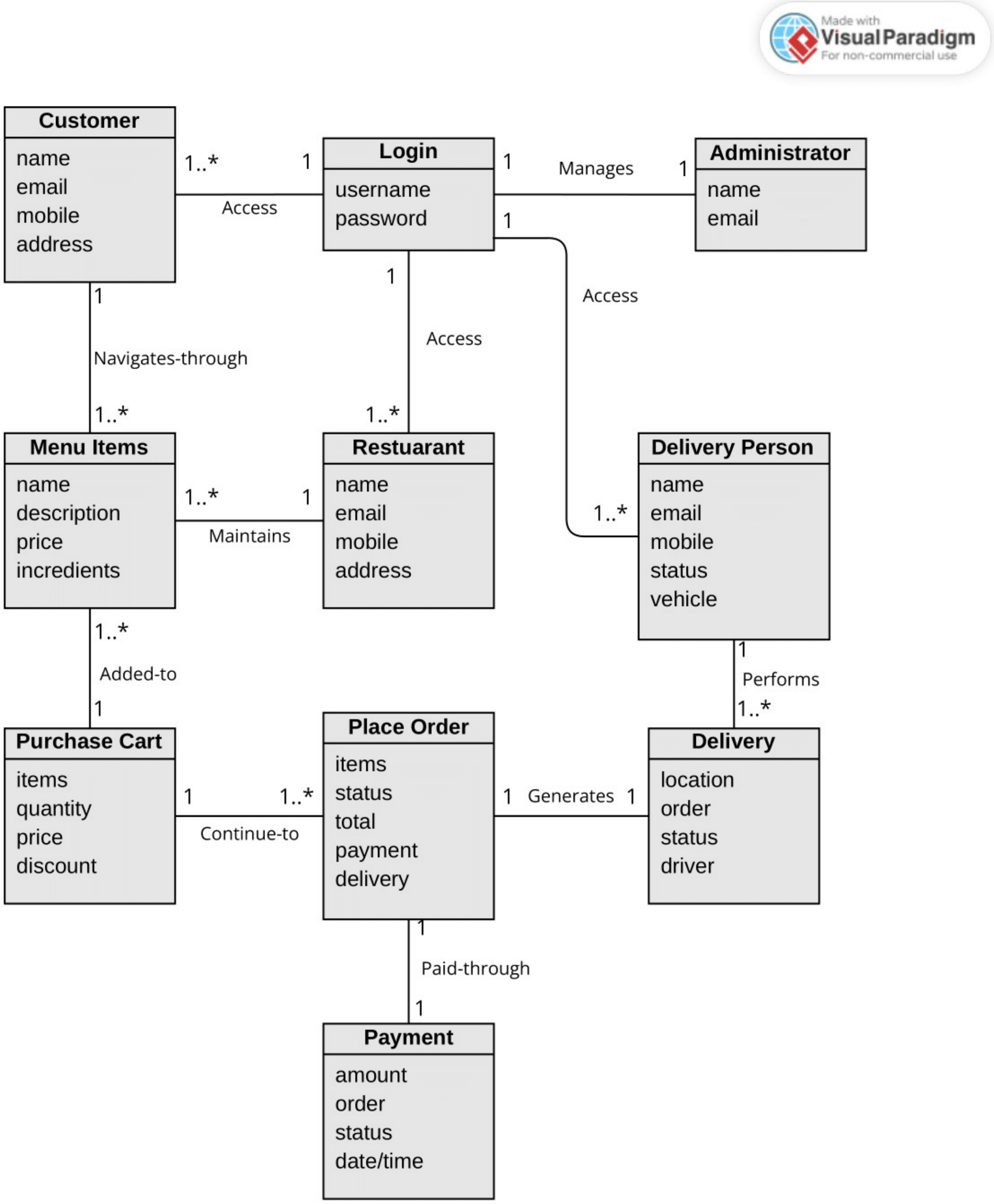
Question 3: Draw a UML use case diagram depicting the use cases from step 1 and their relationship.



Question 4: Draw a UML system sequence diagram (SSD) the “Users putting an order” use case.


:Customer
:System
The customer who orders food
The system includes overall methodologies of the application to perform a sale operation
searchRestuarant()
restaurants open for order
accessMenu()
updated items in Menu
Loop
* [add more items]
additem(itemID, quantity)
itemDetails, currentTotal
The customer can add more than one items to the cart and it could be an iterative process
checkout()
confirmDeliveryInfo()
makePayment(grandTotal)
order status
delivery confirmation
reviewFoodDelivery()

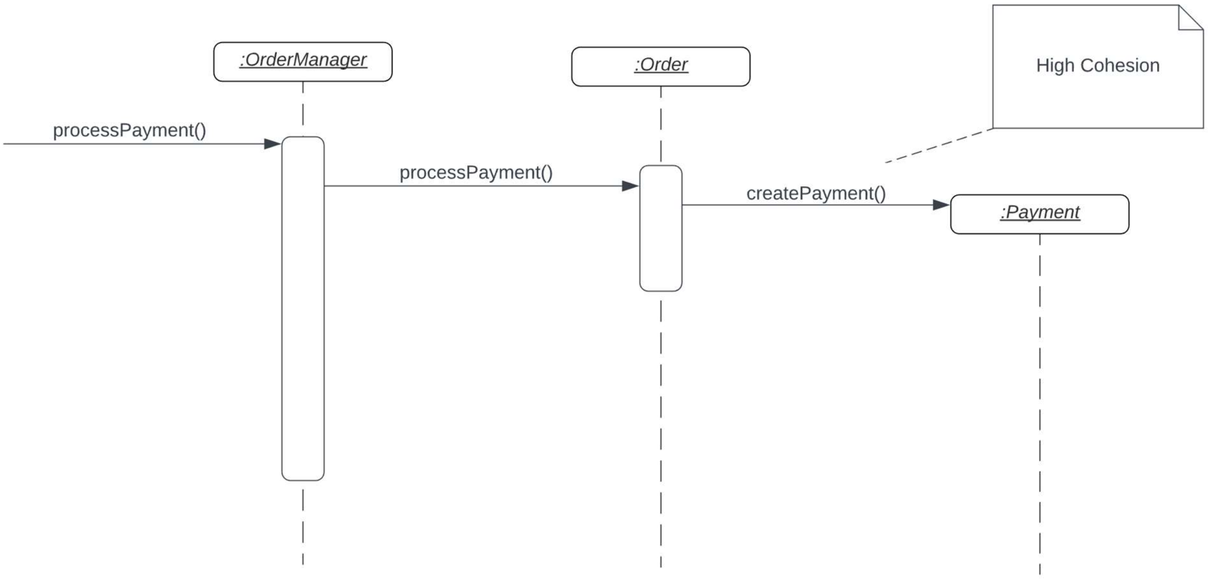
Question 5: Identify the domain objects in your system related to the use cases you have defined in step1 and draw a UML domain model.



Question 6: Show four examples of how you can assign responsibilities to your classes by applying four different GRASP patterns. Create the corresponding collaboration diagram for two of them and sequence diagrams for the other two (your choice which ones are collaboration diagrams and which one's sequence). Annotate which GRASP pattern you applied in each scenario by UML comments (note symbol) in the diagrams.

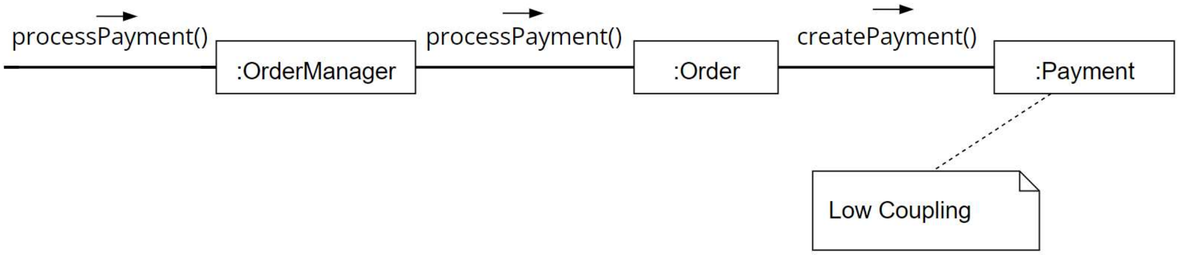
Grasp Pattern 1: High Cohesion: Assigning responsibilities so that Cohesion is highly maintained.

The responsibility to manage the payment functionality of the system is delegated to the Payment Instance, although processing the payment originates with the OrderManager object. This way of delegation will result in maintaining the high cohesion of the system as the responsibility are narrowed to the specific instance of the system rather than creating the payment process in OrderManager object even though it has the privilege to process the payment.



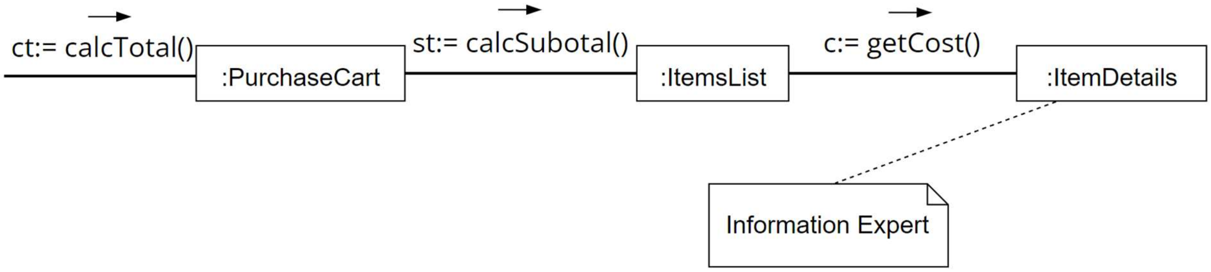
Grasp Pattern 2: Low Coupling: Assigning the responsibility to maintain low coupling.

The class OrderManager has to process payment and that responsibility was assigned to the Payment class as it consists of the payment modes and payment methods, and it will maintain a low coupling in the system.

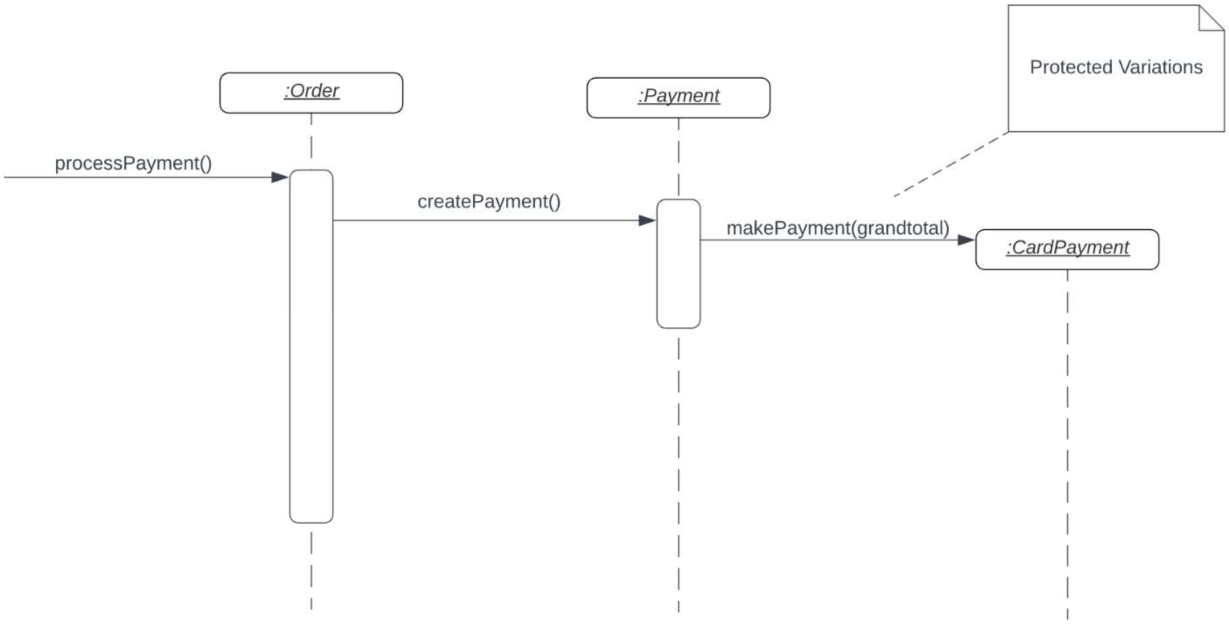


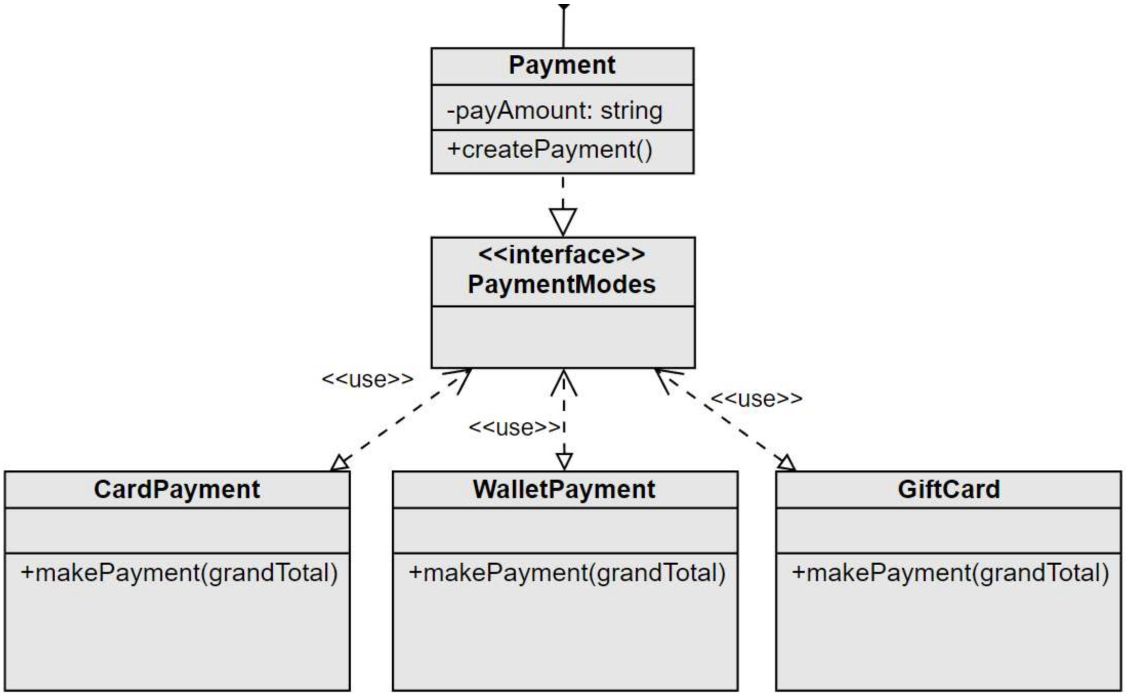
Grasp Pattern 3: Information Expert: Assigning the responsibility to the information Expert.

The class PurchaseCart has to calculate the overall total of all the items in the cart and that responsibility was assigned to the ItemDetails class as it contains the details of the price of each item, and it is the information expert to calculate the total amount of items in the cart.



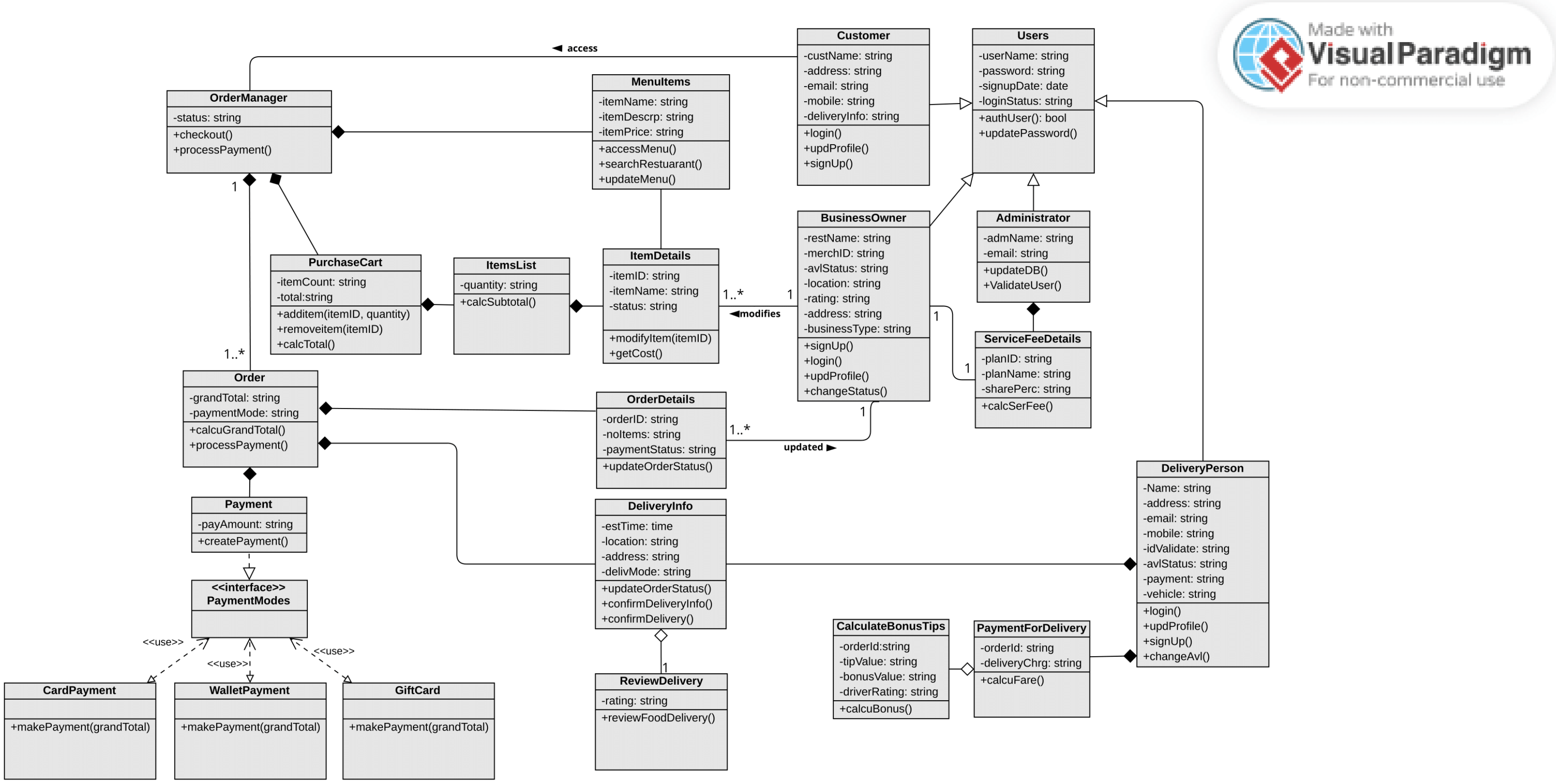
Grasp Pattern 4: Protected Variation: The customer can choose any payment method and the procedure for payment has predicted variations. So, creating a stable interface for payment methods will help in maintaining the stability of system's payment process. In the given sequence diagram the system is implementing the makePayment(grandtotal) methods of CardPyament Class. There is a stable interface for the payment methods and they classes are available as of now such as CardPayment, WalletPayment, and GiftCard (check system Class Diagram image below). If there is a new payment type included in the system, a new class can be added to process the new payment method. This way of implementation also achieves the other Grasp patterns such as polymorphism and indirection as well.





Class Diagram

Question 7: Create a Design class diagram for your system (please cover all the classes required for the use cases you defined from step 1)


Made with Visual Paradigm For non-commercial use
◄ access
OrderManager
-status: string
+checkout()
+processPayment()
MenuItems
-itemName: string
-itemDescrp: string
-itemPrice: string
+accessMenu()
+searchRestuarant()
+updateMenu()
Customer
-custName: string
-address: string
-email: string
-mobile: string
-deliveryInfo: string
+login()
+updProfile()
+signUp()
Users
-userName: string
-password: string
-signupDate: date
-loginStatus: string
+authUser(): bool
+updatePassword()
1
PurchaseCart
-itemCount: string
-total:string
+additem(itemID, quantity)
+removeitem(itemID)
+calcTotal()
ItemsList
-quantity: string
+calcSubtotal()
ItemDetails
-itemID: string
-itemName: string
-status: string
+modifyItem(itemID)
+getCost()
1..*
1
◄modifies
BusinessOwner
-restName: string
-merchID: string
-avlStatus: string
-location: string
-rating: string
-address: string
-businessType: string
+signUp()
+login()
+updProfile()
+changeStatus()
Administrator
-admName: string
-email: string
+updateDB()
+ValidateUser()
1
1
ServiceFeeDetails
-planID: string
-planName: string
-sharePerc: string
+calcSerFee()
1..*
Order
-grandTotal: string
-paymentMode: string
+calcuGrandTotal()
+processPayment()
OrderDetails
-orderID: string
-noItems: string
-paymentStatus: string
+updateOrderStatus()
1
1..*
updated ►
DeliveryPerson
-Name: string
-address: string
-email: string
-mobile: string
-idValidate: string
-avlStatus: string
-payment: string
-vehicle: string
+login()
+updProfile()
+signUp()
+changeAvl()
Payment
-payAmount: string
+createPayment()
DeliveryInfo
-estTime: time
-location: string
-address: string
-delivMode: string
+updateOrderStatus()
+confirmDeliveryInfo()
+confirmDelivery()
<<interface>>
PaymentModes
<<use>>
<<use>>
<<use>>
CalculateBonusTips
-orderId:string
-tipValue: string
-bonusValue: string
-driverRating: string
+calcuBonus()
PaymentForDelivery
-orderId: string
-deliveryChrg: string
+calcuFare()
1
ReviewDelivery
-rating: string
+reviewFoodDelivery()
CardPayment
+makePayment(grandTotal)
WalletPayment
+makePayment(grandTotal)
GiftCard
+makePayment(grandTotal)