1. Identify actors and use cases for the system described above and show them on a UML Use Case Diagram.

和ChatGPT的聊天内容如下：

I: I will give you a system description.Please identify actors and use cases for the system described and show them on a UML Use Case Diagram.

ChatGPT: Sure! Please provide me with the system description, and I'll be happy to identify the actors and use cases for the system and create a UML Use Case Diagram for you.

I: Case study: Order Processing System Consider the following problem description: A mail-order company wants to automate its order processing. The initial version of the order processing system should be accessible to customers via the web. Customers can also call the company by phone and interact with the system via a customer representative. It is highly likely that the company will enhance this system in upcoming years with new features. The system allows customers to place orders, check the status of their orders, cancel an existing order and request a catalog. Customers may also return a product but this is only possible through the phone, not available on the web. When placing an order, the customer identifies himself by means of customer number (only for existing registered customers) or by means of his name and address. He then selects a number of products by giving the product number or by selecting products from the online catalogue. For each product, information such as price, a description and a picture (only on demand as they are usually high-resolution images of large size) are presented to the customer. Also, the availability of the product is obtained from the inventory. The customer indicates whether he wants to buy the product and in what quantity. When all desired products have been selected, the customer provides a shipping address and a credit card number and a billing address (if different from the shipping address). Then an overview of the ordered products and the total cost are presented. If the customer approves, the order is submitted. Credit card number, billing address and a specification of the cost of the order are used on the invoice, which is forwarded to the accounting system (an existing software module). Orders are forwarded to the shipping company, where they are filled and shipped. Customers who spent over a certain amount within the past year are promoted to be gold customers. Gold customers have additional rights such as being able to return products in an extended time period as well as earning more bonus points with each purchase. In addition, in cases where a product is on back order, gold customers have the option to sign up for an email notification for when the particular product becomes available.

ChatGpt: Based on the system description provided, we can identify the following actors and use cases:

Actors:

1. Customer: The main actor who interacts with the system. The customer can access the system via the web or phone and perform various actions.

2. Customer Representative: A representative of the company who interacts with the system on behalf of the customer during phone interactions.

Use Cases:

1. Place Order: The customer can place an order by selecting products from the online catalog, providing shipping and billing information, and confirming the order.

2. Check Order Status: The customer can check the status of their orders to know whether they have been processed, shipped, or delivered.

3. Cancel Order: The customer can cancel an existing order if it has not been shipped yet.

4. Request Catalog: The customer can request a catalog to be sent to their address.

5. Return Product (Phone Only): The customer can return a product by contacting the customer representative via phone.

6. Promote to Gold Customer: The system automatically promotes customers to gold status based on their past year's spending.

7. Handle Extended Returns (Gold Customers): Gold customers have the ability to return products within an extended time period.

8. Earn Bonus Points (Gold Customers): Gold customers earn more bonus points with each purchase.

9. Sign Up for Product Availability Notification (Gold Customers): Gold customers can sign up to receive email notifications when a back-ordered product becomes available.

Here's a UML Use Case Diagram representing the actors and use cases:

```

+------------------+

| Customer |

+------------------+

| +------------------------+

+---------> | Place Order |

| +------------------------+

|

| +------------------------+

+---------> | Check Order Status |

| +------------------------+

|

| +------------------------+

+---------> | Cancel Order |

| +------------------------+

|

| +------------------------+

+---------> | Request Catalog |

| +------------------------+

|

| +------------------------+

+---------> | Return Product |

| +------------------------+

|

| +------------------------+

+---------> | Promote to Gold Customer |

| +------------------------+

|

| +------------------------+

+---------> | Handle Extended Returns (Gold Customers) |

| +------------------------+

|

| +------------------------+

+---------> | Earn Bonus Points (Gold Customers) |

| +------------------------+

|

| +------------------------+

+---------> | Sign Up for Product Availability Notification (Gold Customers) |

+------------------------+

```

Note: The diagram showcases the relationships between actors and use cases. Arrows pointing from actors to use cases represent the interactions or actions performed by the actors in the system.

I: There is more than one actors, your diagram needs improvement

ChatGpt: Apologies for the oversight. Let's update the UML Use Case Diagram to include multiple actors and their interactions with the system. Based on the system description, we have the following additional actors:

Actors:

1. Web Customer: A customer who interacts with the system through the web interface.

2. Phone Customer: A customer who interacts with the system through phone interactions.

3. Shipping Company: A company responsible for fulfilling and shipping orders.

Here's an updated UML Use Case Diagram incorporating these actors and their respective use cases:

```

+------------------+

| Customer |

+------------------+

| +------------------------+

+---------> | Place Order |

| +------------------------+

|

| +------------------------+

+---------> | Check Order Status |

| +------------------------+

|

| +------------------------+

+---------> | Cancel Order |

| +------------------------+

|

| +------------------------+

+---------> | Request Catalog |

| +------------------------+

|

| +------------------------+

+---------> | Return Product (Phone Only) |

| +------------------------+

|

| +------------------------+

+---------> | Promote to Gold Customer |

| +------------------------+

|

| +------------------------+

+---------> | Handle Extended Returns (Gold Customers) |

| +------------------------+

|

| +------------------------+

+---------> | Earn Bonus Points (Gold Customers) |

| +------------------------+

|

| +------------------------+

+---------> | Sign Up for Product Availability Notification (Gold Customers) |

+------------------------+

/ \

/ \

/ \

/ \

| |

+--------------------|----------------------+ |

| | | |

+--------------------+ +------------------+ +--------------------+

| Web Customer | | Phone Customer | | Shipping Company |

+--------------------+ +------------------+ +--------------------+

| | |

| | |

+-------------------------+------------------------+

| |

| |

+------------------------------------------------+

|

|

+--------------------+

| Check Order |

| Availability |

+--------------------+

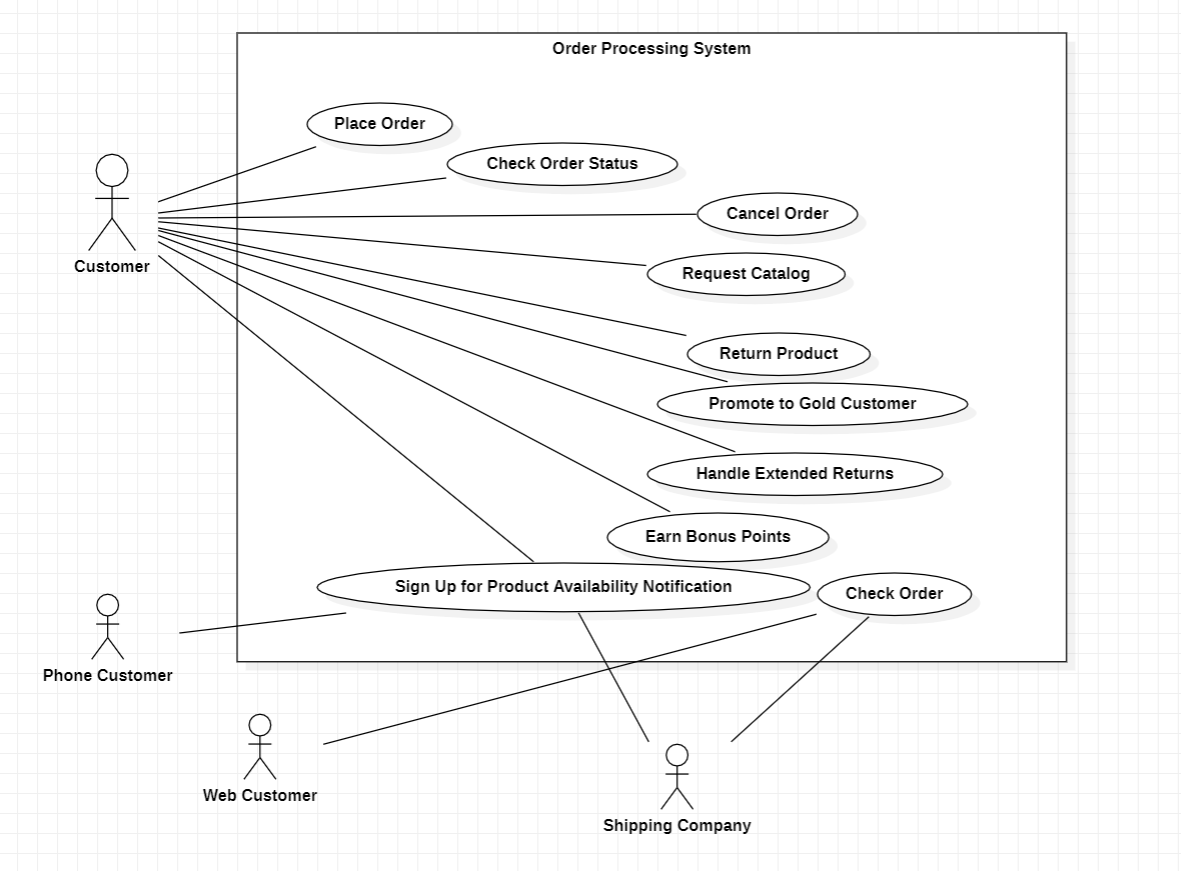
```

In the updated diagram, the actors Web Customer, Phone Customer, and Shipping Company are represented. The interactions between the actors and the use cases are shown with arrows. The Check Order Availability use case is added, indicating that both Web Customers and Phone Customers can perform this action.

Please let me know if there's anything else I can assist you with!

在该例中当我第一次把问题抛给ChatGPT时，它给出答案的两个参与者是客户和客户代表，并且给出的用例图中只画出客户一个参与者，显然这个图并不全面。

第二次提问时我让GPT注意参与者的问题，第二次给的答案中参与者有了客户，打电话的客户（实际上还是客户代表），和运输公司，终于跟我想的差不多了。它给出的用例图画出来大致是这样：



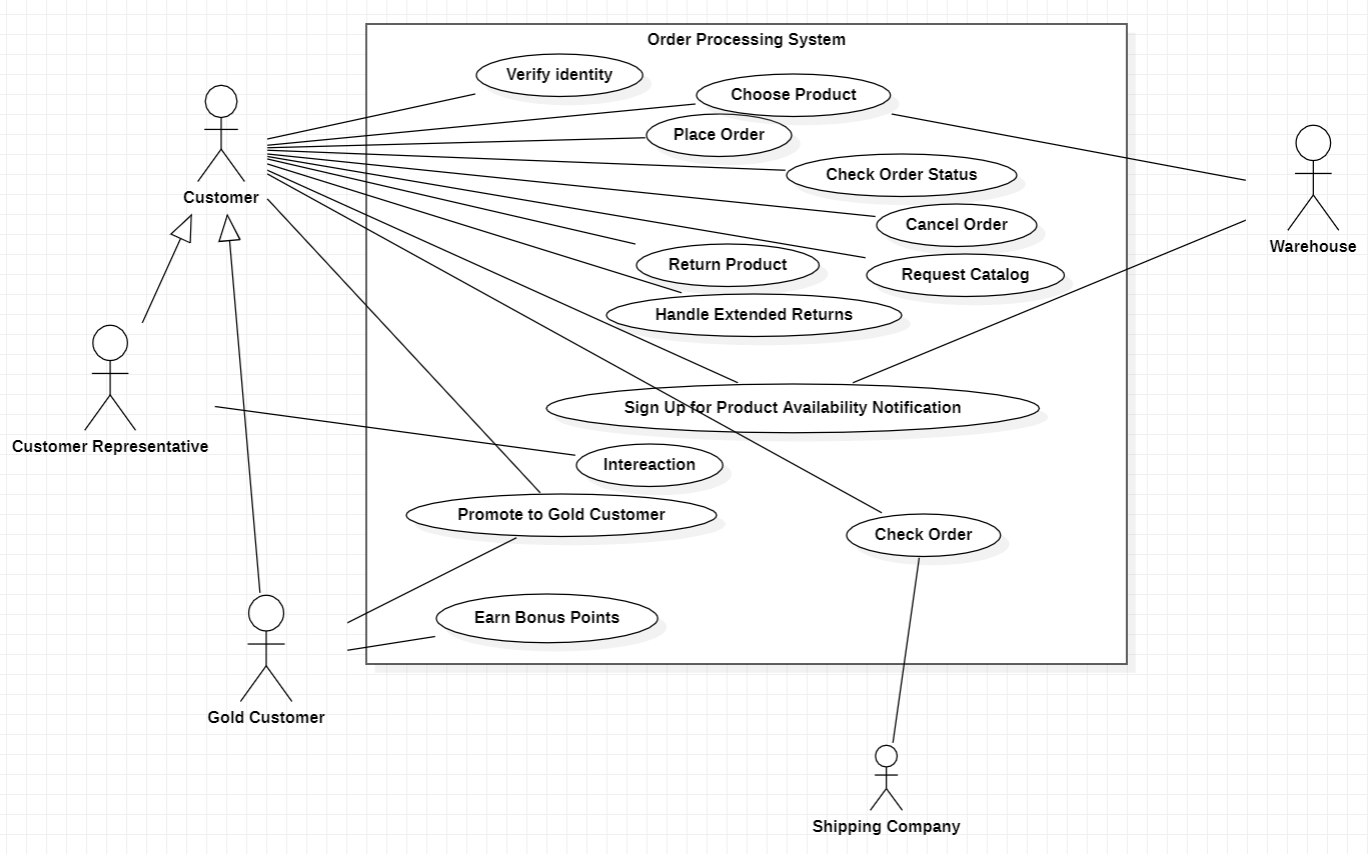
问题很多，我们进行修正：

·将Customer Representative和Gold Customer泛化为一般Customer

·客户的操作需要从仓库里获取商品的信息，我们增加一个参与者Warehouse

·增加验证身份和选择产品的用例

改正后的用例图：



1. Perform a quick application domain analysis to come up with an object model for the above system. Express your findings with a UML Class Diagram, making sure to identify any critical operations of classes.

I: Perform a quick application domain analysis to come up with an object model for the above system. Express your findings with a UML Class Diagram, making sure to identify any critical operations of classes.

ChatGPT: To perform a quick application domain analysis and create an object model for the Order Processing System, we'll identify the key entities and their relationships based on the system description. Then, we'll represent the entities as classes in a UML Class Diagram, highlighting the critical operations of each class. Here's the UML Class Diagram for the system:

```

+---------------------------------+

| Order |

+---------------------------------+

| - orderNumber: string |

| - status: string |

| - products: list<Product> |

| - customer: Customer |

+---------------------------------+

| + placeOrder(): void |

| + checkStatus(): string |

| + cancelOrder(): void |

+---------------------------------+

+---------------------------------+

| Customer |

+---------------------------------+

| - customerNumber: string |

| - name: string |

| - address: string |

| - isGoldCustomer: boolean |

+---------------------------------+

| + getCustomerNumber(): string |

| + getName(): string |

| + getAddress(): string |

| + isGoldCustomer(): boolean |

| + returnProduct(): void |

+---------------------------------+

+---------------------------------+

| Product |

+---------------------------------+

| - productNumber: string |

| - description: string |

| - price: decimal |

| - availability: int |

+---------------------------------+

| + getProductNumber(): string |

| + getDescription(): string |

| + getPrice(): decimal |

| + getAvailability(): int |

+---------------------------------+

+---------------------------------+

| ShippingCompany |

+---------------------------------+

| - name: string |

| - address: string |

+---------------------------------+

| + getName(): string |

| + getAddress(): string |

| + fulfillOrder(order: Order): void |

+---------------------------------+

```

In the above UML Class Diagram:

- The Order class represents an order placed by a customer. It has attributes such as orderNumber, status, products (a list of Product objects), and a customer object representing the customer who placed the order. Critical operations of the Order class include placeOrder(), checkStatus(), and cancelOrder().

- The Customer class represents a customer of the mail-order company. It has attributes such as customerNumber, name, address, and isGoldCustomer to determine if the customer is a gold customer. Critical operations of the Customer class include getCustomerNumber(), getName(), getAddress(), isGoldCustomer(), and returnProduct().

- The Product class represents a product available in the system. It has attributes such as productNumber, description, price, and availability. Critical operations of the Product class include getProductNumber(), getDescription(), getPrice(), and getAvailability().

- The ShippingCompany class represents the shipping company responsible for fulfilling orders. It has attributes such as name and address. The critical operation fulfillOrder(order: Order) is used to handle order fulfillment.

The UML Class Diagram provides an overview of the classes, their attributes, and critical operations in the system. Note that the diagram captures only a subset of the classes and their relationships based on the given system description. Additional classes, relationships, and operations may exist depending on the complete system requirements.

Let me know if you need further clarification or if there's anything else I can assist you with!

I: Can you draw the relationship between these classes on the diagram?

ChatGPT: Certainly! Here's an updated UML Class Diagram that includes the relationships between the classes:

```

+---------------------------------+

| Order |

+---------------------------------+

| - orderNumber: string |

| - status: string |

| - products: list<Product> |

| - customer: Customer |

+---------------------------------+

| + placeOrder(): void |

| + checkStatus(): string |

| + cancelOrder(): void |

+---------------------------------+

|

|

|

|

|

+-----|-----------------+

| v |

+---------------------------------+ +---------------------------------+

| Customer | | Product |

+---------------------------------+ +---------------------------------+

| - customerNumber: string | | - productNumber: string |

| - name: string | | - description: string |

| - address: string | | - price: decimal |

| - isGoldCustomer: boolean | | - availability: int |

+---------------------------------+ +---------------------------------+

| + getCustomerNumber(): string | | + getProductNumber(): string |

| + getName(): string | | + getDescription(): string |

| + getAddress(): string | | + getPrice(): decimal |

| + isGoldCustomer(): boolean | | + getAvailability(): int |

| + returnProduct(): void | +---------------------------------+

+---------------------------------+

|

|

|

|

|

+---------|---------+

| v |

+---------------------------------+

| ShippingCompany |

+---------------------------------+

| - name: string |

| - address: string |

+---------------------------------+

| + getName(): string |

| + getAddress(): string |

| + fulfillOrder(order: Order): void |

+---------------------------------+

```

In the updated UML Class Diagram, the relationships between classes are represented as lines connecting the related classes. The relationships are as follows:

- The Order class has a composition relationship with the Product class, represented by the association line between Order and Product. This indicates that an order consists of multiple products.

- The Order class has an aggregation relationship with the Customer class, represented by the association line between Order and Customer. This indicates that an order is associated with a customer, but the customer can exist independently of the order.

- The Customer class and Product class are connected to the Order class through the association lines, indicating the relationship between the classes involved in an order.

- The ShippingCompany class is connected to the Order class through an association line, indicating that the shipping company fulfills orders.

Please let me know if there's anything else I can assist you with!

I: Please add Gold Customer

ChatGPT: Apologies for the oversight. Here's an updated UML Class Diagram that includes the GoldCustomer class and its relationships:

```

+---------------------------------+

| Order |

+---------------------------------+

| - orderNumber: string |

| - status: string |

| - products: list<Product> |

| - customer: Customer |

+---------------------------------+

| + placeOrder(): void |

| + checkStatus(): string |

| + cancelOrder(): void |

+---------------------------------+

|

|

|

|

|

+-----|-----------------+

| v |

+---------------------------------+ +---------------------------------+

| Customer | | Product |

+---------------------------------+ +---------------------------------+

| - customerNumber: string | | - productNumber: string |

| - name: string | | - description: string |

| - address: string | | - price: decimal |

| - isGoldCustomer: boolean | | - availability: int |

+---------------------------------+ +---------------------------------+

| + getCustomerNumber(): string | | + getProductNumber(): string |

| + getName(): string | | + getDescription(): string |

| + getAddress(): string | | + getPrice(): decimal |

| + isGoldCustomer(): boolean | | + getAvailability(): int |

| + returnProduct(): void | +---------------------------------+

+---------------------------------+

|

|

|

|

|

+---------|---------+

| v |

+---------------------------------+

| ShippingCompany |

+---------------------------------+

| - name: string |

| - address: string |

+---------------------------------+

| + getName(): string |

| + getAddress(): string |

| + fulfillOrder(order: Order): void |

+---------------------------------+

|

|

|

|

|

+-----|-----------------+

| v |

+---------------------------------+

| GoldCustomer |

+---------------------------------+

| |

+---------------------------------+

| + extendReturnPeriod(): void |

| + earnBonusPoints(): void |

| + signUpForNotifications(): void|

+---------------------------------+

```

In the updated UML Class Diagram, the GoldCustomer class is added, and its relationships are represented as follows:

- The GoldCustomer class is a subclass of the Customer class, indicated by the inheritance arrow.

- The GoldCustomer class has additional operations specific to gold customers: extendReturnPeriod(), earnBonusPoints(), and signUpForNotifications().

Please let me know if there's anything else I can assist you with!

I: I think gold customer should have association with customer.

ChatGPT: Apologies for the oversight. You're right, the GoldCustomer class should have an association with the Customer class. Here's the revised UML Class Diagram:

```

+---------------------------------+

| Order |

+---------------------------------+

| - orderNumber: string |

| - status: string |

| - products: list<Product> |

| - customer: Customer |

+---------------------------------+

| + placeOrder(): void |

| + checkStatus(): string |

| + cancelOrder(): void |

+---------------------------------+

|

|

|

|

|

+-----|-----------------+

| v |

+---------------------------------+ +---------------------------------+

| Customer | | Product |

+---------------------------------+ +---------------------------------+

| - customerNumber: string | | - productNumber: string |

| - name: string | | - description: string |

| - address: string | | - price: decimal |

| - isGoldCustomer: boolean | | - availability: int |

+---------------------------------+ +---------------------------------+

| + getCustomerNumber(): string | | + getProductNumber(): string |

| + getName(): string | | + getDescription(): string |

| + getAddress(): string | | + getPrice(): decimal |

| + isGoldCustomer(): boolean | | + getAvailability(): int |

| + returnProduct(): void | +---------------------------------+

+---------------------------------+

|

|

|

|

|

+-----|-----------------+

| v |

+---------------------------------+

| GoldCustomer |

+---------------------------------+

| |

+---------------------------------+

| + extendReturnPeriod(): void |

| + earnBonusPoints(): void |

| + signUpForNotifications(): void|

+---------------------------------+

|

|

|

|

|

+---------------------------------+

| Customer |

+---------------------------------+

| - customerNumber: string |

| - name: string |

| - address: string |

| - isGoldCustomer: boolean |

+---------------------------------+

| + getCustomerNumber(): string |

| + getName(): string |

| + getAddress(): string |

| + isGoldCustomer(): boolean |

| + returnProduct(): void |

+---------------------------------+

|

|

|

|

|

+---------|---------+

| v |

+---------------------------------+

| ShippingCompany |

+---------------------------------+

| - name: string |

| - address: string |

+---------------------------------+

| + getName(): string |

| + getAddress(): string |

| + fulfillOrder(order: Order): void |

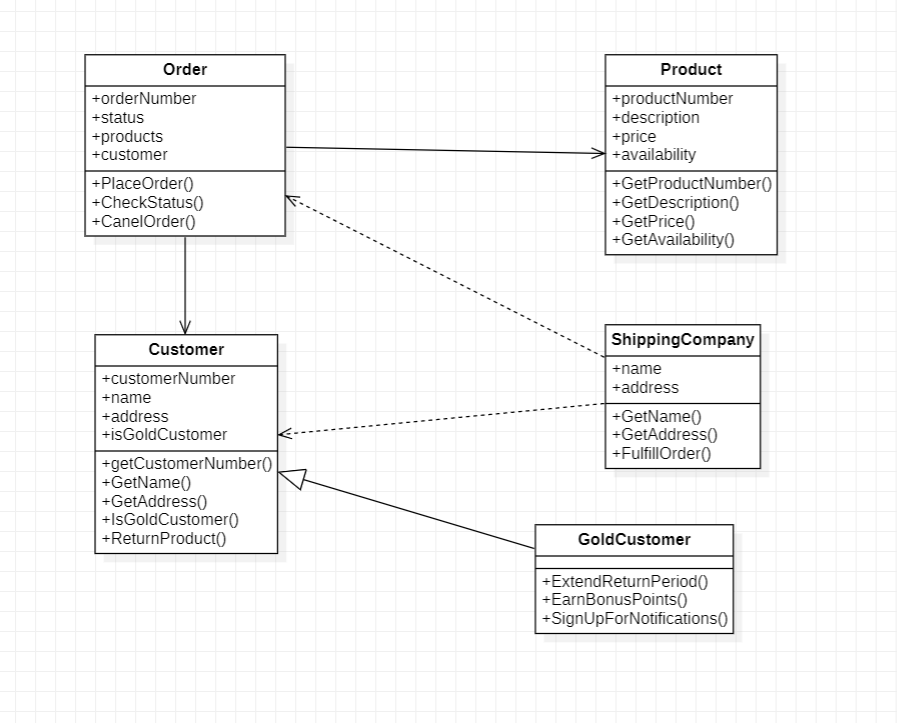
+---------------------------------+

```

In the revised diagram, the GoldCustomer class has an association with the Customer class, indicating that a GoldCustomer is a type of Customer. The Customer class retains its original attributes and operations.

Once again, I apologize for the confusion, and thank you for pointing out the correction. Let me know if there's anything else I can assist you with!

多次尝试后ChatGPT始终没法给出正确的类与类之间的关系，我们直接在此基础上修改（这里就没有写类的属性类型和方法的返回值）：



1. Consider the following use case scenario (for use case “place order”):

Ali is an existing customer of the order processing company described earlier, registered with their web site. Also assume that having browsed the printed catalogue he has, he already identified the two items (including their prices) he likes to buy from the company’s website using their product numbers (i.e. #2 and #9). First, he tries to buy one of product #2, but it is listed as unavailable in the inventory. Then, he adds two quantities of product #9, which turns out to be available, to his basket. He is then asked to confirm his registered shipping and billing addresses and credit card information from the customer database. He completes the order by clicking the Submit button. You may ignore processing of customer authentication.

draw a UML Sequence Diagram for this particular scenario. You may use any software/solution domain objects if needed as well.

当我把问题抛给ChatGPT时，它给出的答案是这样的：

Here's a UML Sequence Diagram depicting the use case scenario "place order" for the given scenario:

```

+-------------------+

| Customer |

+-------------------+

|

| placeOrder()

|

+-------------------+

| Order |

+-------------------+

|

| addProductToOrder(productNumber: #2)

|

+-------------------+

| Product |

+-------------------+

|

| checkAvailability()

|

+-------------------+

| Order |

+-------------------+

|

| addProductToOrder(productNumber: #9, quantity: 2)

|

+-------------------+

| Product |

+-------------------+

|

| checkAvailability()

|

+-------------------+

| Order |

+-------------------+

|

| confirmShippingAddress()

|

+-------------------+

| Customer |

+-------------------+

|

| getShippingAddress()

|

+-------------------+

| Order |

+-------------------+

|

| confirmBillingAddress()

|

+-------------------+

| Customer |

+-------------------+

|

| getBillingAddress()

|

+-------------------+

| Order |

+-------------------+

|

| confirmCreditCard()

|

+-------------------+

| Customer |

+-------------------+

|

| getCreditCardInformation()

|

+-------------------+

| Order |

+-------------------+

|

| submitOrder()

|

+-------------------+

| OrderSystem |

+-------------------+

|

| processOrder(order)

|

+-------------------+

| ShippingCompany |

+-------------------+

|

| fulfillOrder(order)

|

```

In the above UML Sequence Diagram, the interactions between different objects are shown in a chronological order based on the use case scenario "place order." Here's a step-by-step breakdown of the interactions:

1. The Customer initiates the placeOrder() operation.

2. The Customer adds a product with productNumber #2 to the Order.

3. The Product checks its availability.

4. Since the product #2 is unavailable, the OrderSystem notifies the Customer.

5. The Customer adds two quantities of product #9 to the Order.

6. The Product checks the availability of product #9.

7. The OrderSystem confirms the shipping address with the Customer.

8. The Customer retrieves their shipping address.

9. The OrderSystem confirms the billing address with the Customer.

10. The Customer retrieves their billing address.

11. The OrderSystem confirms the credit card information with the Customer.

12. The Customer retrieves their credit card information.

13. The Order is submitted by the Customer.

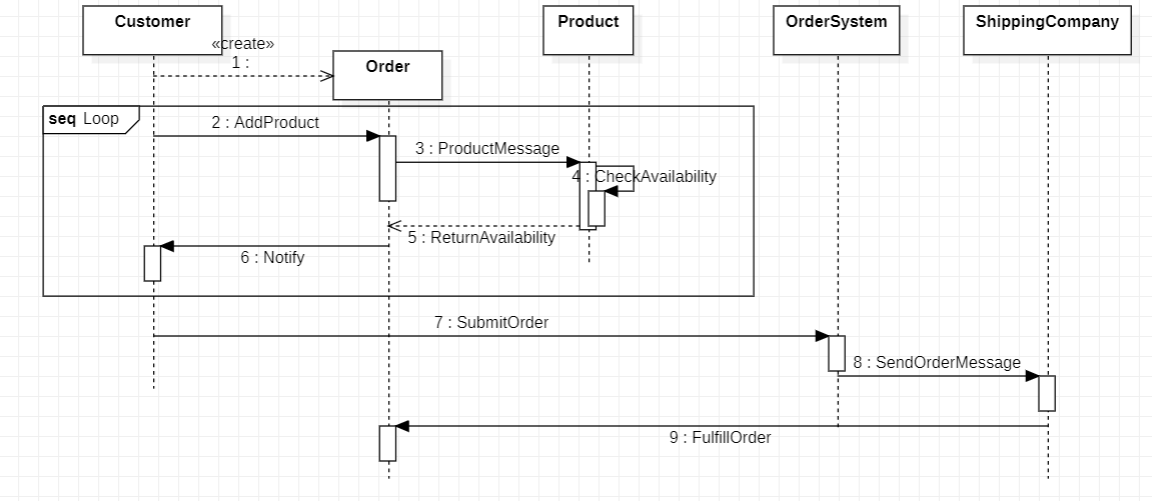
14. The OrderSystem processes the Order.

15. The Order is fulfilled by the ShippingCompany.

Please note that the sequence diagram illustrates the interactions between objects and the flow of messages based on the given scenario. It represents a high-level overview and may not include every detail or exception handling.

Let me know if you need further clarification or if there's anything else I can assist you with!

它的顺序图格式显然有问题，对格式进行纠正：



以上就是最终的顺序图。（不太会用软件所以没写传递的参数）