**初始版本**

**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.

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

Actors:

Customer: Represents individuals who interact with the system to place orders, check order status, cancel orders, request catalogs, and return products (via phone).

Customer Representative: Represents employees of the company who interact with the system on behalf of customers over the phone.

Accounting System: Represents an existing software module responsible for handling invoices and forwarding them for accounting purposes.

Shipping Company: Represents the external entity responsible for filling and shipping orders.

System Administrator: Represents individuals who manage and maintain the system, including future enhancements.

Use Cases:

Place Order: Customers can place an order by selecting products, providing shipping and billing information, reviewing the order details, and submitting the order.

Check Order Status: Customers can check the status of their orders to get updates on processing, shipping, and delivery.

Cancel Order: Customers can request the cancellation of an existing order before it is shipped.

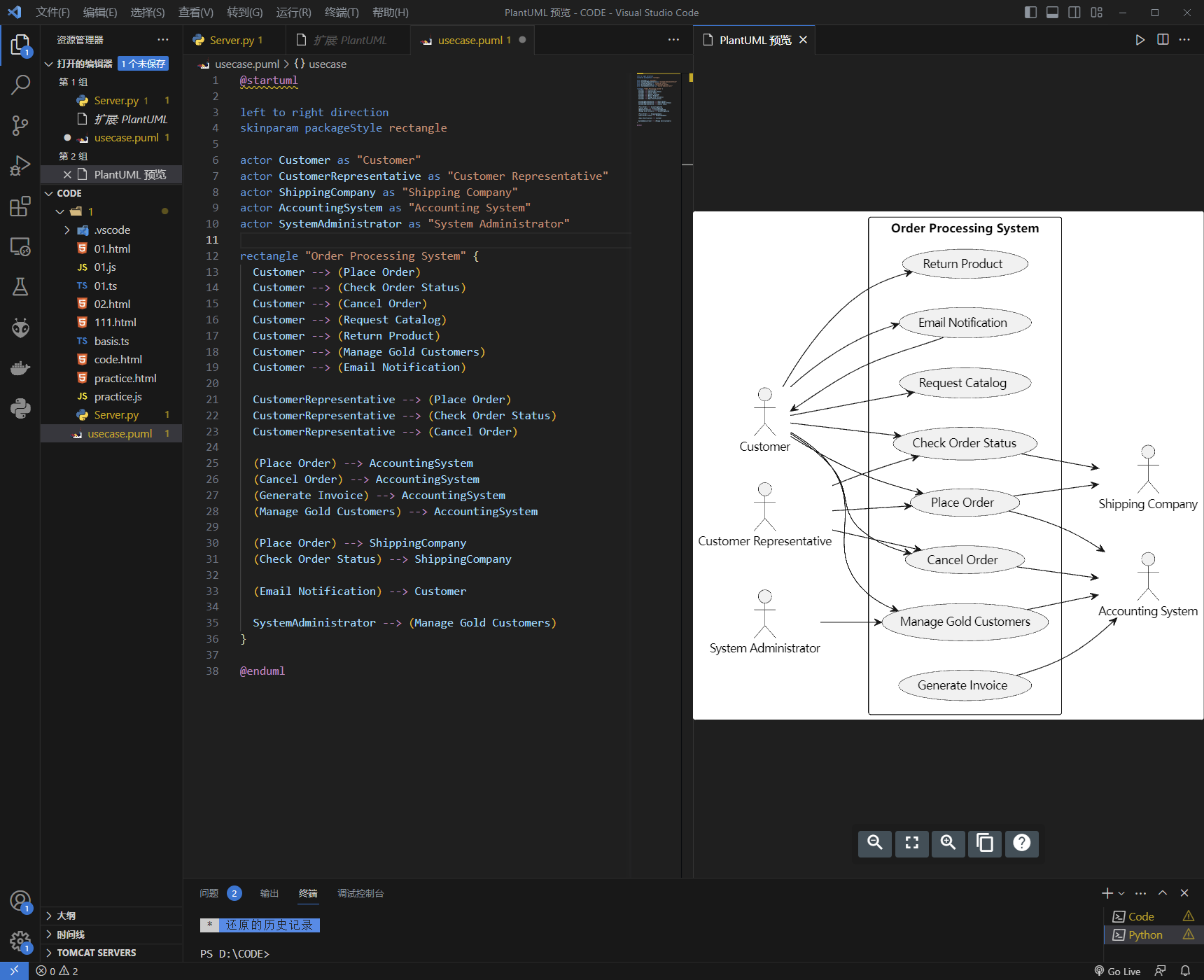
Request Catalog: Customers can request a catalog to be sent to them.

Return Product: Customers (via phone) can initiate the return process for a product.

Generate Invoice: The system generates an invoice with relevant information, including credit card details, billing address, and order cost, which is then forwarded to the accounting system.

Manage Gold Customers: The system identifies and manages gold customers based on their purchase history, allowing them extended return periods and bonus points.

Email Notification: Gold customers have the option to sign up for email notifications when a back-ordered product becomes available.



(2) 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.

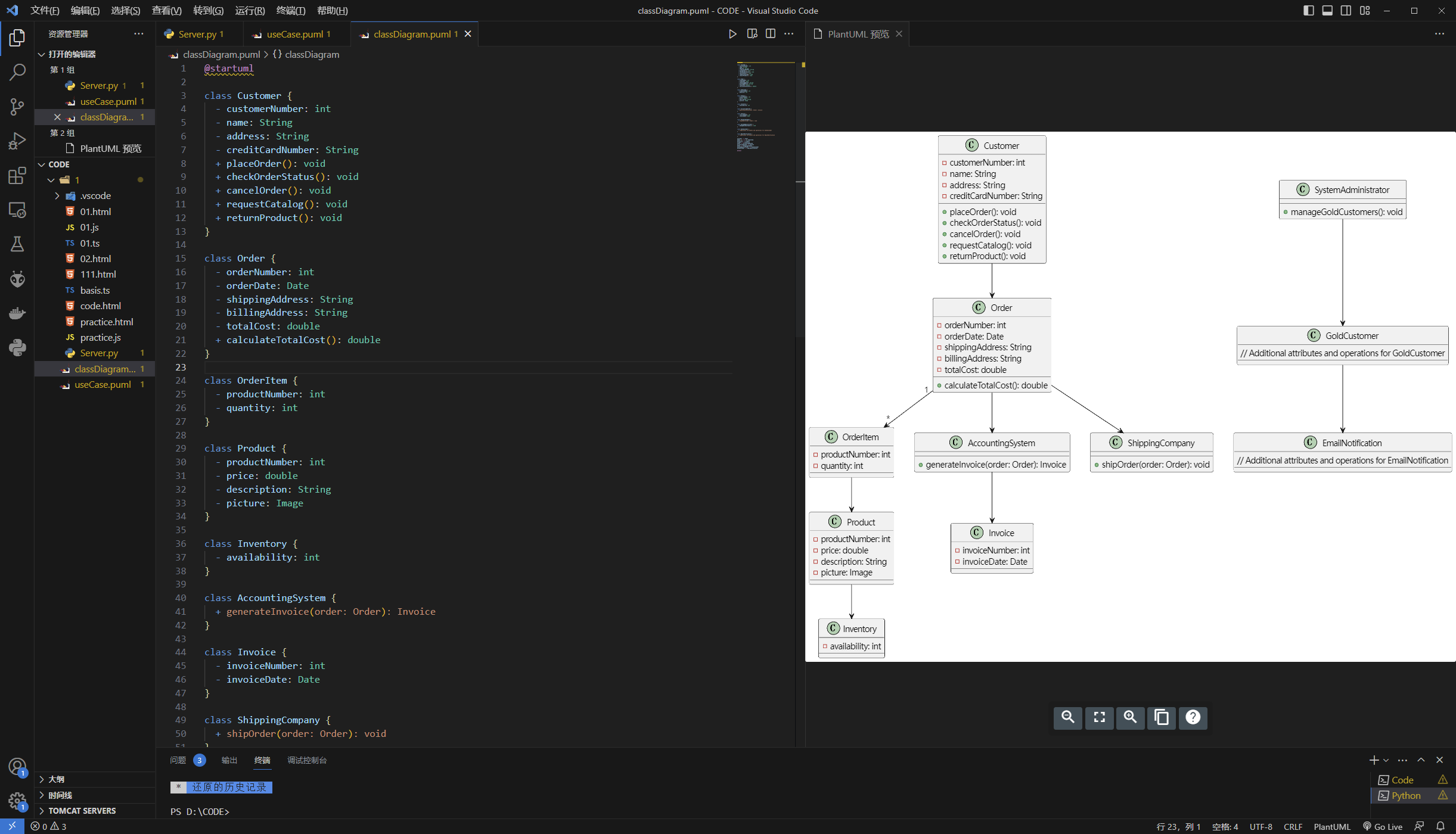
1. Customer:
   * Attributes: customerNumber, name, address, creditCardNumber
   * Associations: PlaceOrder, CheckOrderStatus, CancelOrder, RequestCatalog, ReturnProduct
2. Order:
   * Attributes: orderNumber, orderDate, shippingAddress, billingAddress, totalCost
   * Associations: Customer, OrderItem
3. OrderItem:
   * Attributes: productNumber, quantity
   * Associations: Order, Product
4. Product:
   * Attributes: productNumber, price, description, picture
   * Associations: OrderItem, Inventory
5. Inventory:
   * Attributes: availability
   * Associations: Product
6. AccountingSystem:
   * Associations: Order, Invoice
7. Invoice:
   * Attributes: invoiceNumber, invoiceDate
   * Associations: AccountingSystem, Order
8. ShippingCompany:
   * Associations: Order
9. SystemAdministrator:
   * Associations: ManageGoldCustomers
10. GoldCustomer:

* Associations: Customer

1. EmailNotification:

* Associations: GoldCustomer

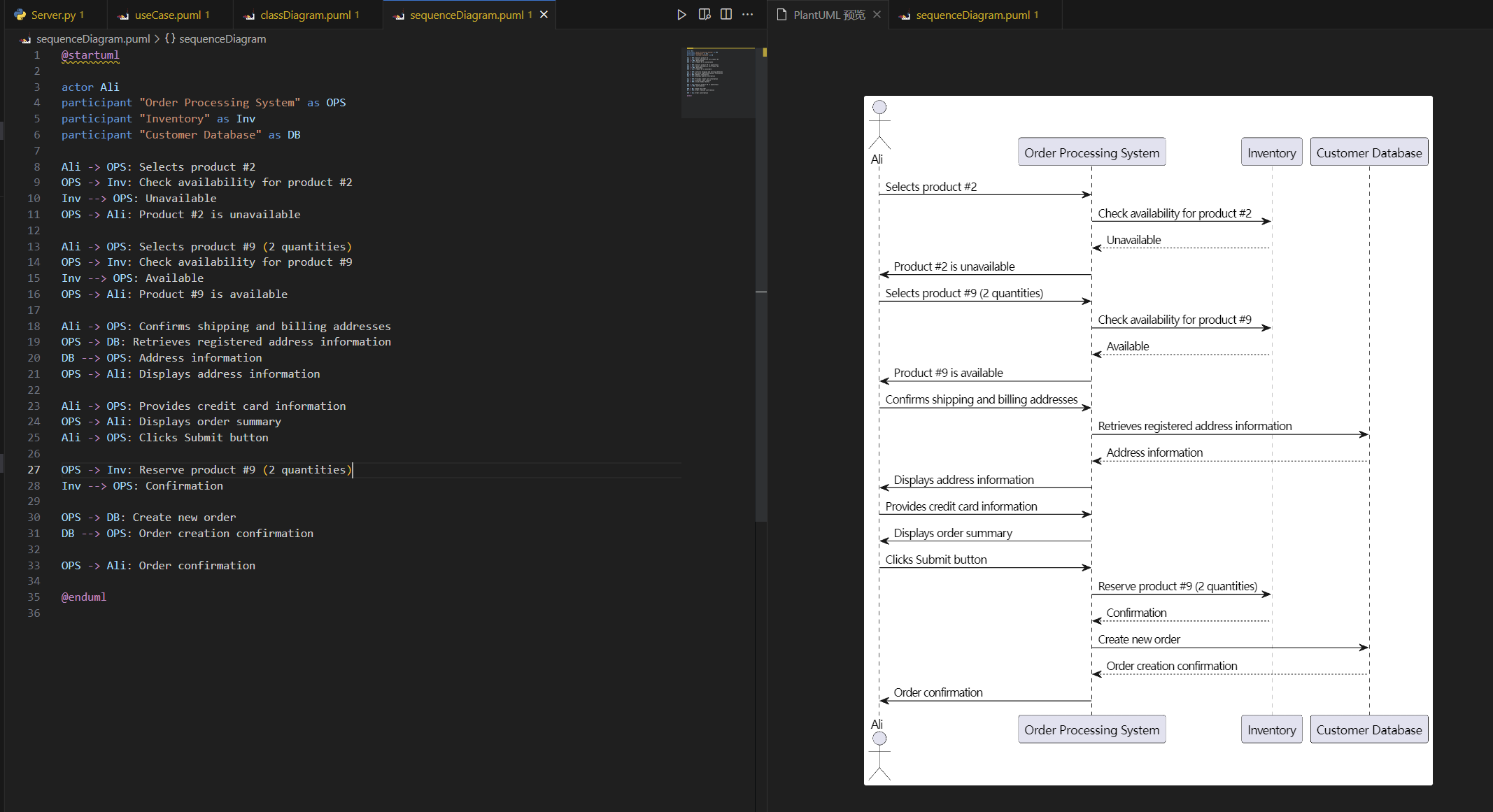
In this object model, the main entities such as Customer, Order, OrderItem, Product, Inventory, AccountingSystem, Invoice, ShippingCompany, SystemAdministrator, GoldCustomer, and EmailNotification are represented. The associations between entities indicate their relationships and interactions.



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.*

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



**优化版本**

**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.

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

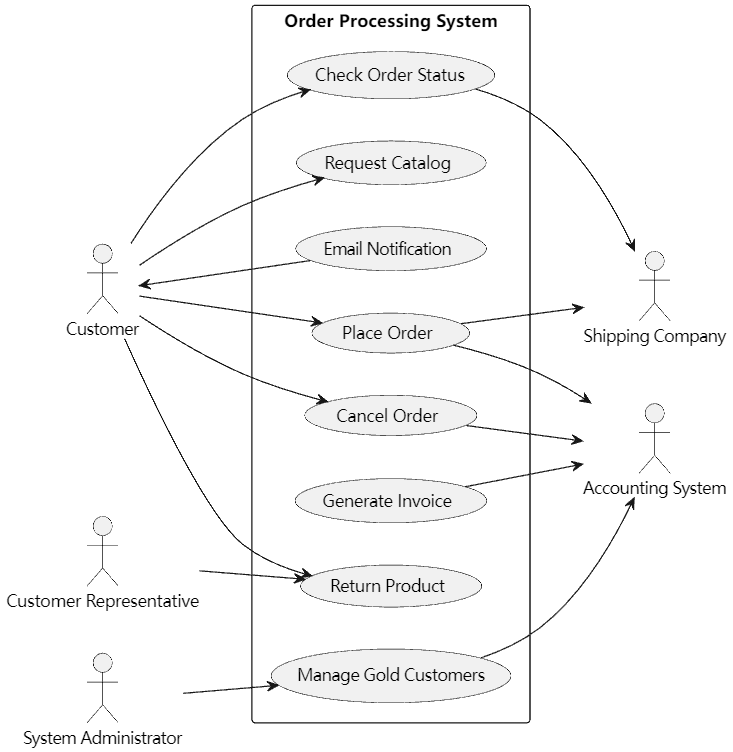
Actors:

1. Customer: Represents individuals who interact with the system to place orders, check order status, cancel orders, request catalogs, and return products (via phone).
2. Customer Representative: Represents employees of the company who interact with the system on behalf of customers over the phone.
3. Accounting System: Represents an existing software module responsible for handling invoices and forwarding them for accounting purposes.
4. Shipping Company: Represents the external entity responsible for filling and shipping orders.
5. System Administrator: Represents individuals who manage and maintain the system, including future enhancements.

Use Cases:

1. Place Order: Customers can place an order by selecting products, providing shipping and billing information, reviewing the order details, and submitting the order.
2. Check Order Status: Customers can check the status of their orders to get updates on processing, shipping, and delivery.
3. Cancel Order: Customers can request the cancellation of an existing order before it is shipped.
4. Request Catalog: Customers can request a catalog to be sent to them.
5. Return Product (Phone): Customers can initiate the return process for a product by contacting Customer Representative via phone.
6. Generate Invoice: The system generates an invoice with relevant information, including credit card details, billing address, and order cost, which is then forwarded to the accounting system.
7. Manage Gold Customers: The system identifies and manages gold customers based on their purchase history, allowing them extended return periods and bonus points.
8. Email Notification (Gold Customers): Gold customers have the option to sign up for email notifications when a back-ordered product becomes available.

By optimizing the actors and use cases, we have removed the unnecessary duplication of similar use cases and clarified the roles of actors in the system. This streamlined representation ensures better clarity and avoids redundancy.



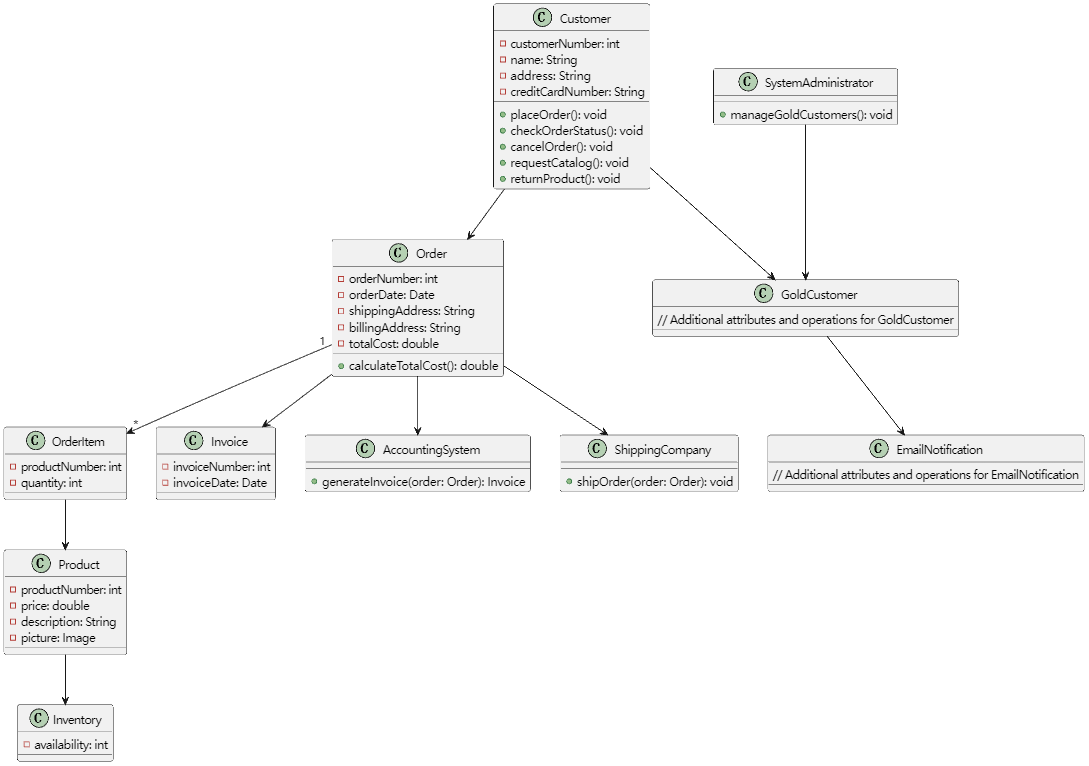
In this updated version of the PlantUML code, the Use Case Diagram reflects the optimized use cases and actors for the Order Processing System. The diagram highlights the relationships between the actors and the corresponding use cases. The system administrator is associated with the "Manage Gold Customers" use case, and the customer representative is associated with the "Return Product" use case.

(2) 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.

1. Customer:
   * Attributes: customerNumber, name, address, creditCardNumber
   * Associations: PlaceOrder, CheckOrderStatus, CancelOrder, RequestCatalog, ReturnProduct
2. Order:
   * Attributes: orderNumber, orderDate, shippingAddress, billingAddress, totalCost
   * Associations: Customer, OrderItem, Invoice
3. OrderItem:
   * Attributes: productNumber, quantity
   * Associations: Order, Product
4. Product:
   * Attributes: productNumber, price, description, picture
   * Associations: OrderItem, Inventory
5. Inventory:
   * Attributes: availability
   * Associations: Product
6. Invoice:
   * Attributes: invoiceNumber, invoiceDate
   * Associations: Order, AccountingSystem
7. AccountingSystem:
   * Associations: GenerateInvoice
8. ShippingCompany:
   * Associations: ShipOrder
9. SystemAdministrator:
   * Associations: ManageGoldCustomers
10. GoldCustomer:
    * Associations: Customer
11. EmailNotification:
    * Associations: GoldCustomer

In this object model, the main entities such as Customer, Order, OrderItem, Product, Inventory, Invoice, AccountingSystem, ShippingCompany, SystemAdministrator, GoldCustomer, and EmailNotification are represented. The associations between entities indicate their relationships and interactions.

The optimized object model removes the redundant associations and incorporates the updated use cases. It clarifies the relationships between the entities and their interactions within the system.

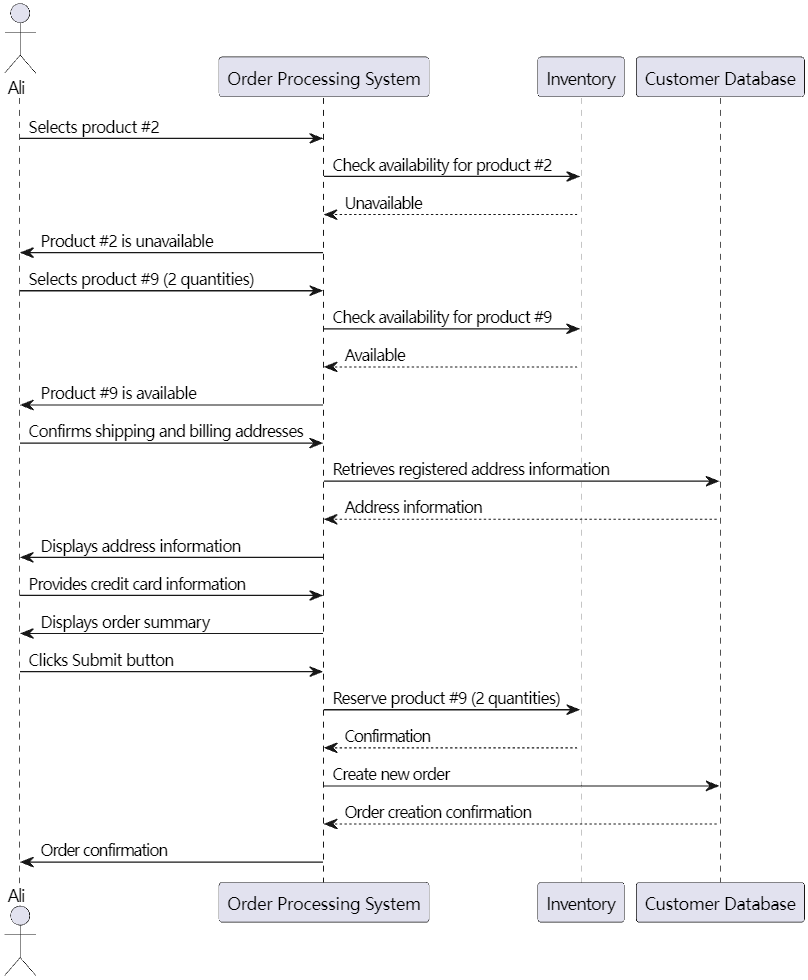


This PlantUML code represents the UML class diagram for the optimized Order Processing System. The classes are defined with their attributes and critical operations. Associations between classes are depicted using arrows, indicating the relationships and multiplicities. The "+" symbol before the operation names represents public visibility.

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.*

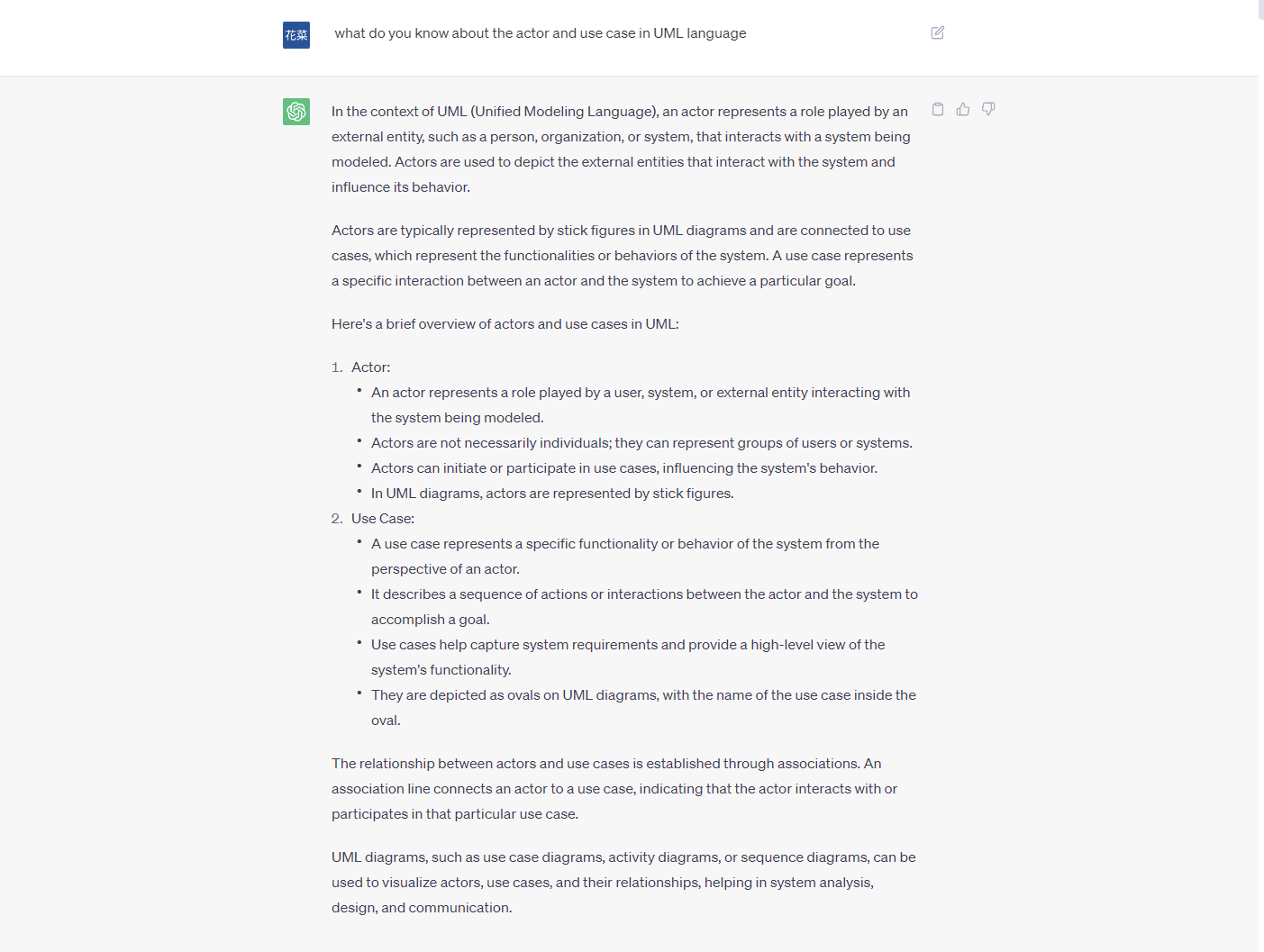
(3) draw a UML Sequence Diagram for this particular scenario. You may use any software/solution domain objects if needed as well.

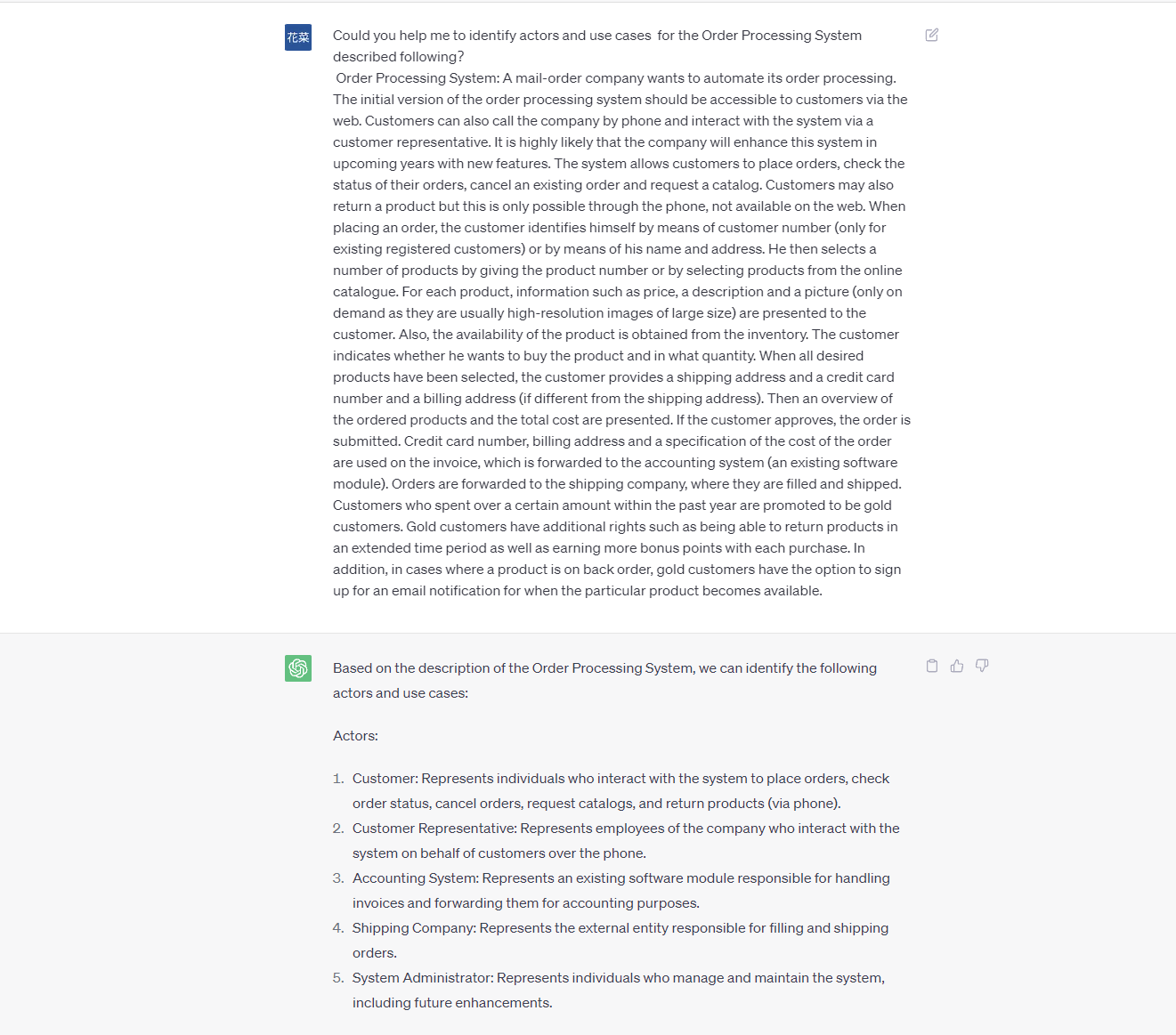


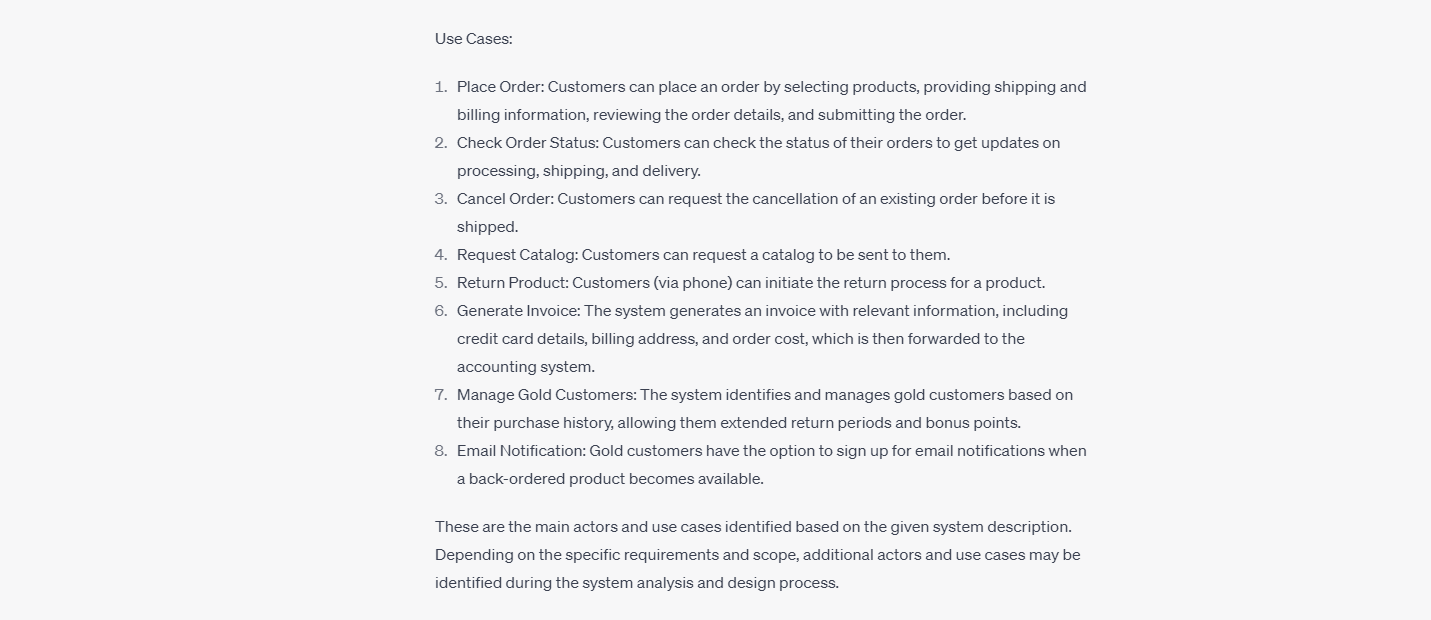
This PlantUML code represents the UML sequence diagram for the 'place order' scenario in the optimized Order Processing System. It illustrates the interactions between the actor (Ali) and the system components (Order Processing System, Inventory, and Customer Database). Arrows represent the flow of messages between the participants.

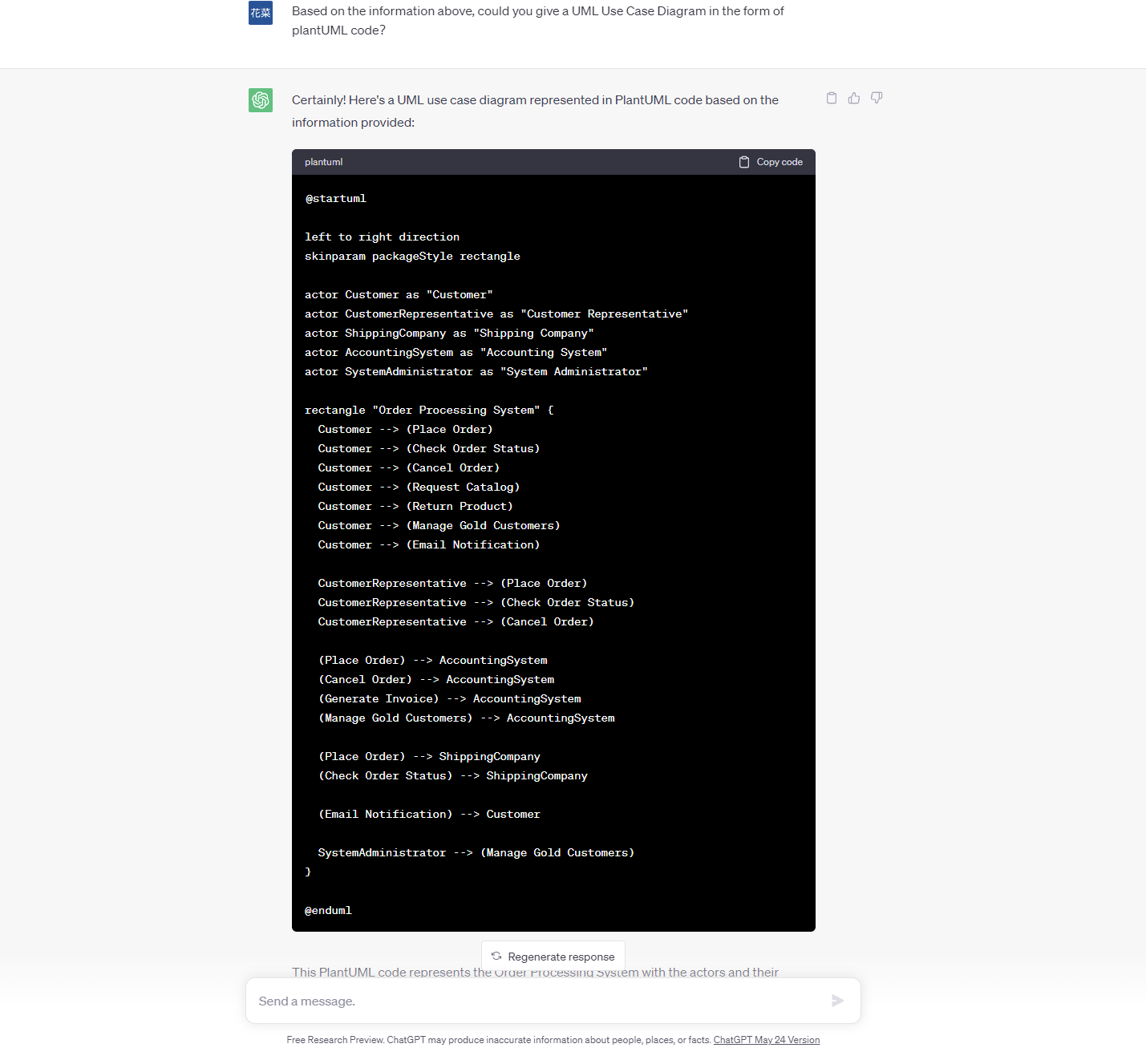
**初始方案构建ChatGPT辅助过程**

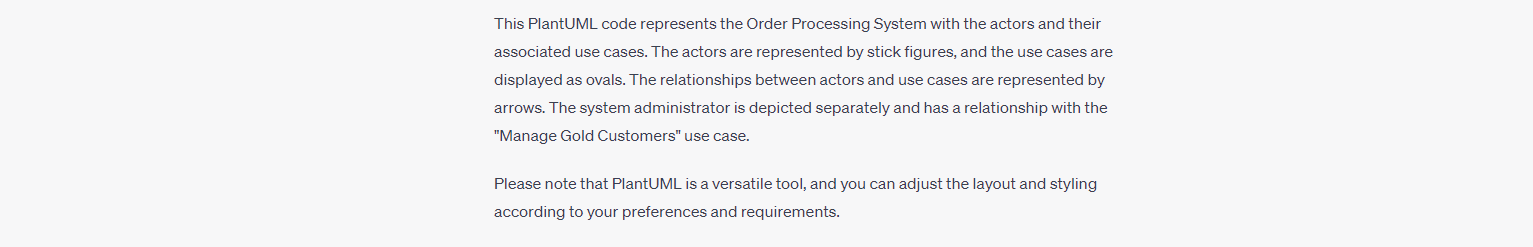
**Actor / Use case / Use case diagram：**





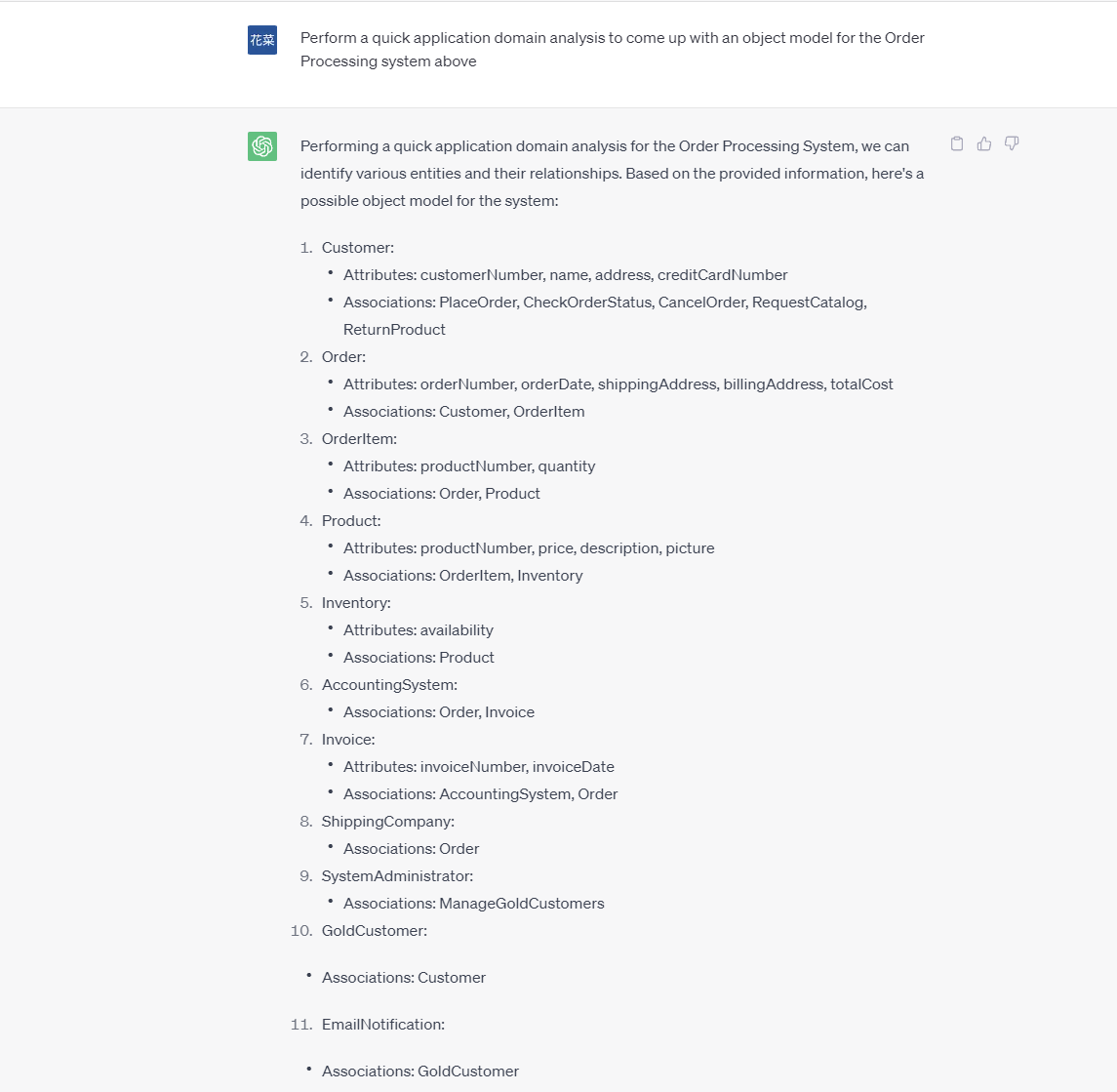


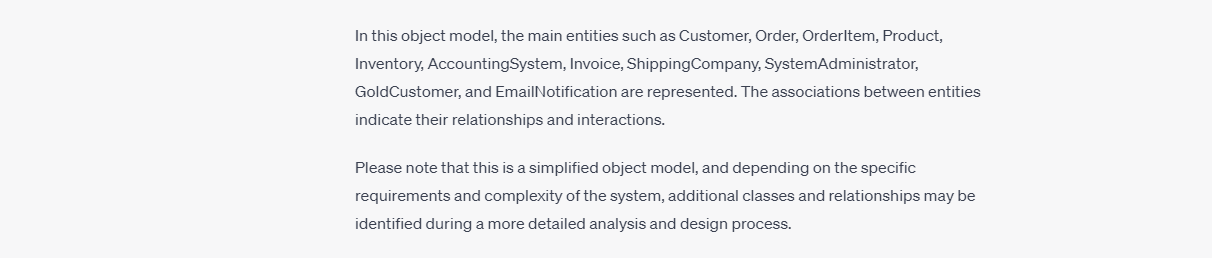


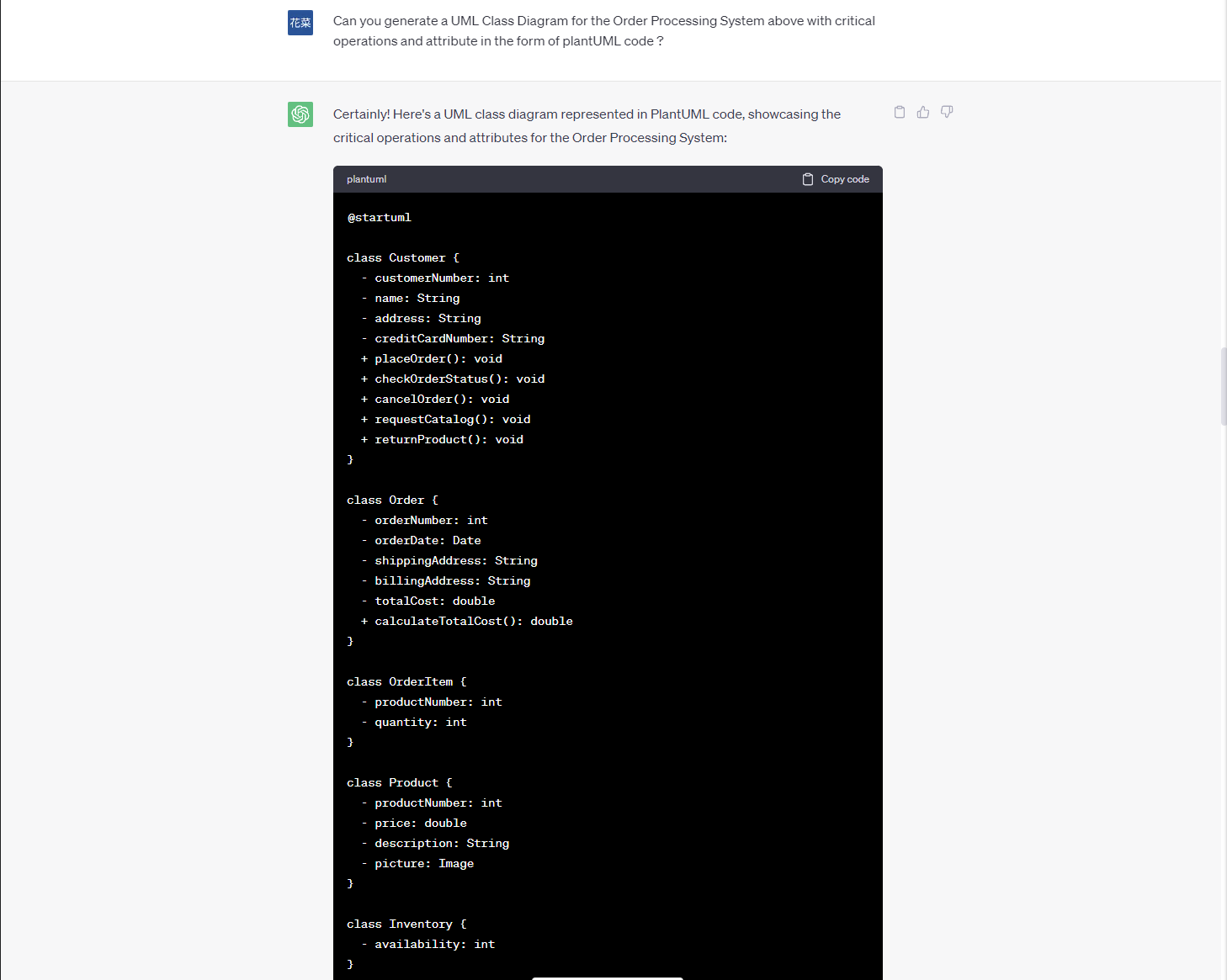


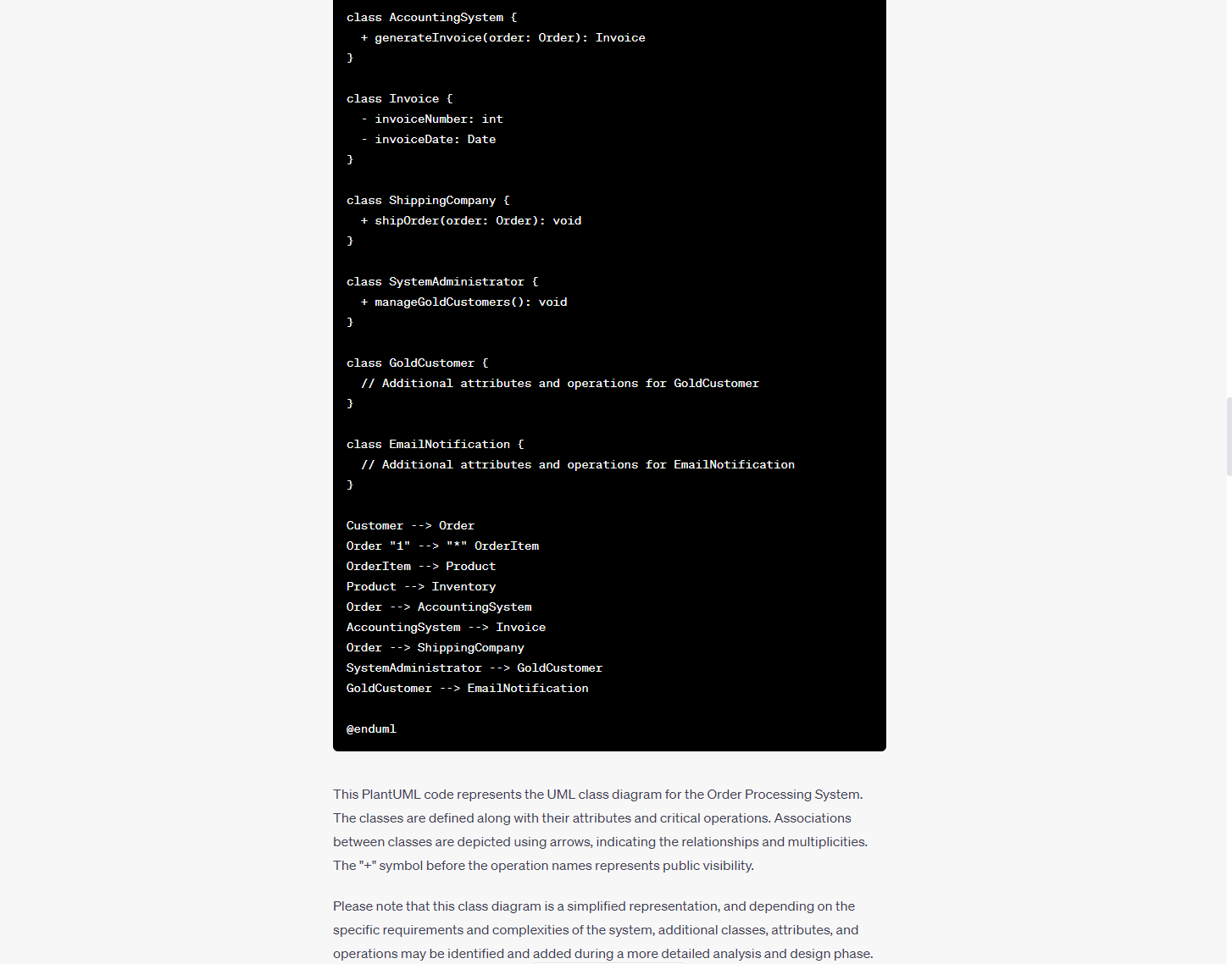
**共计三次提问。**

**Object model / Class diagram:**



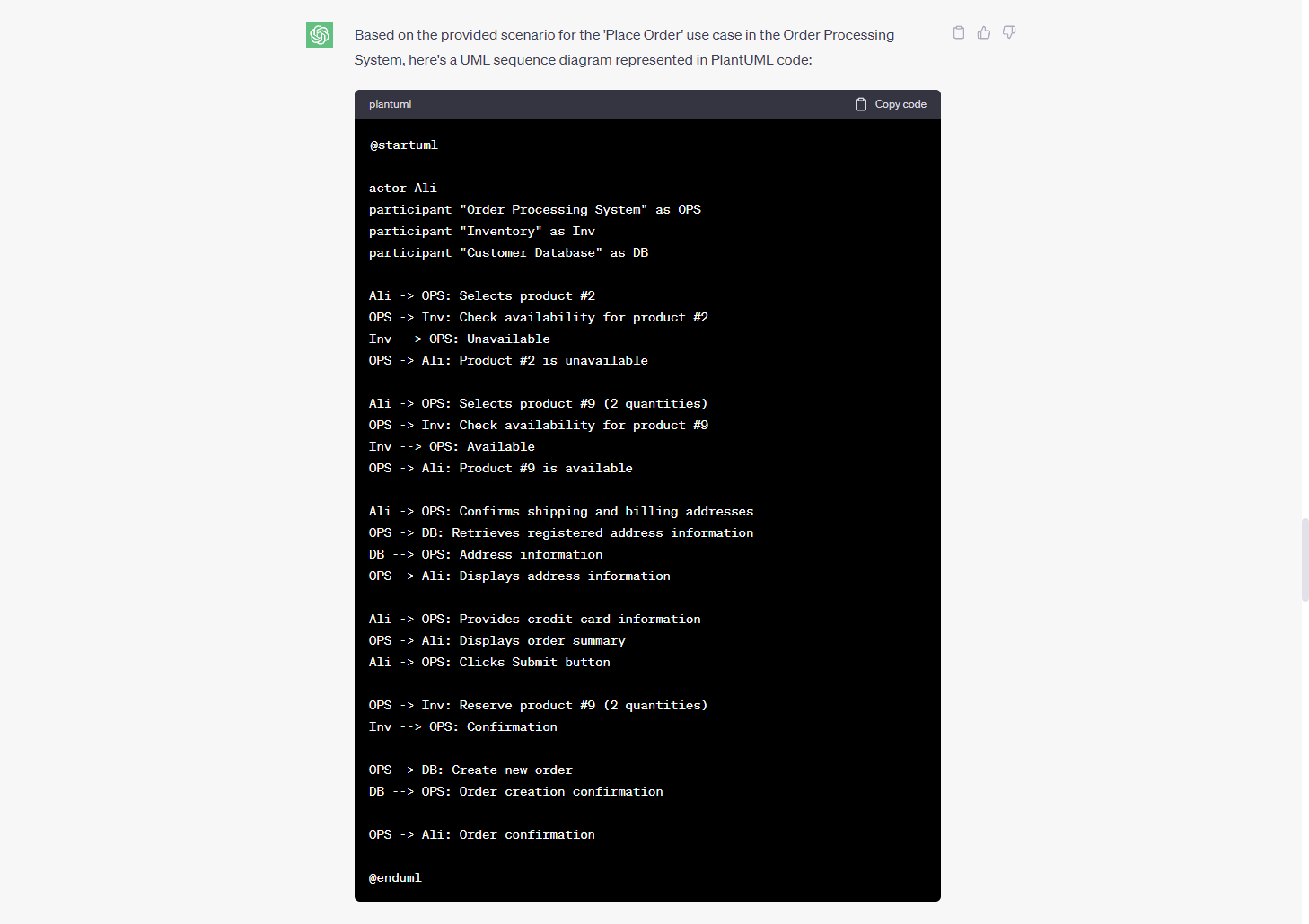


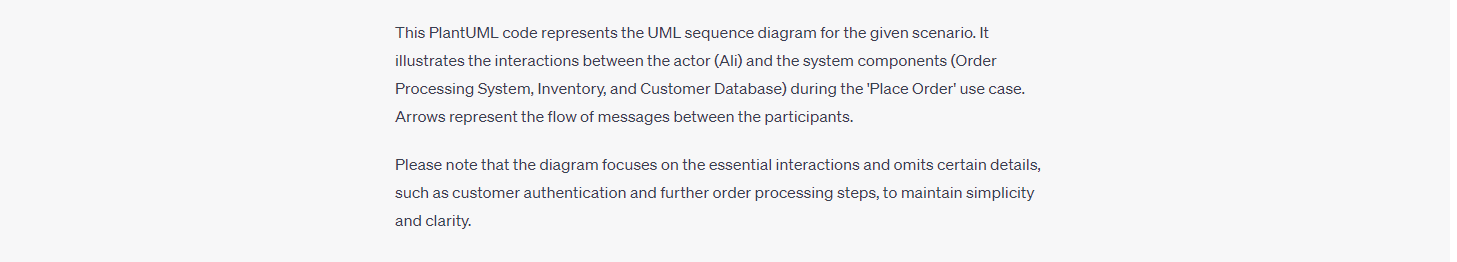




**共计两次提问。**

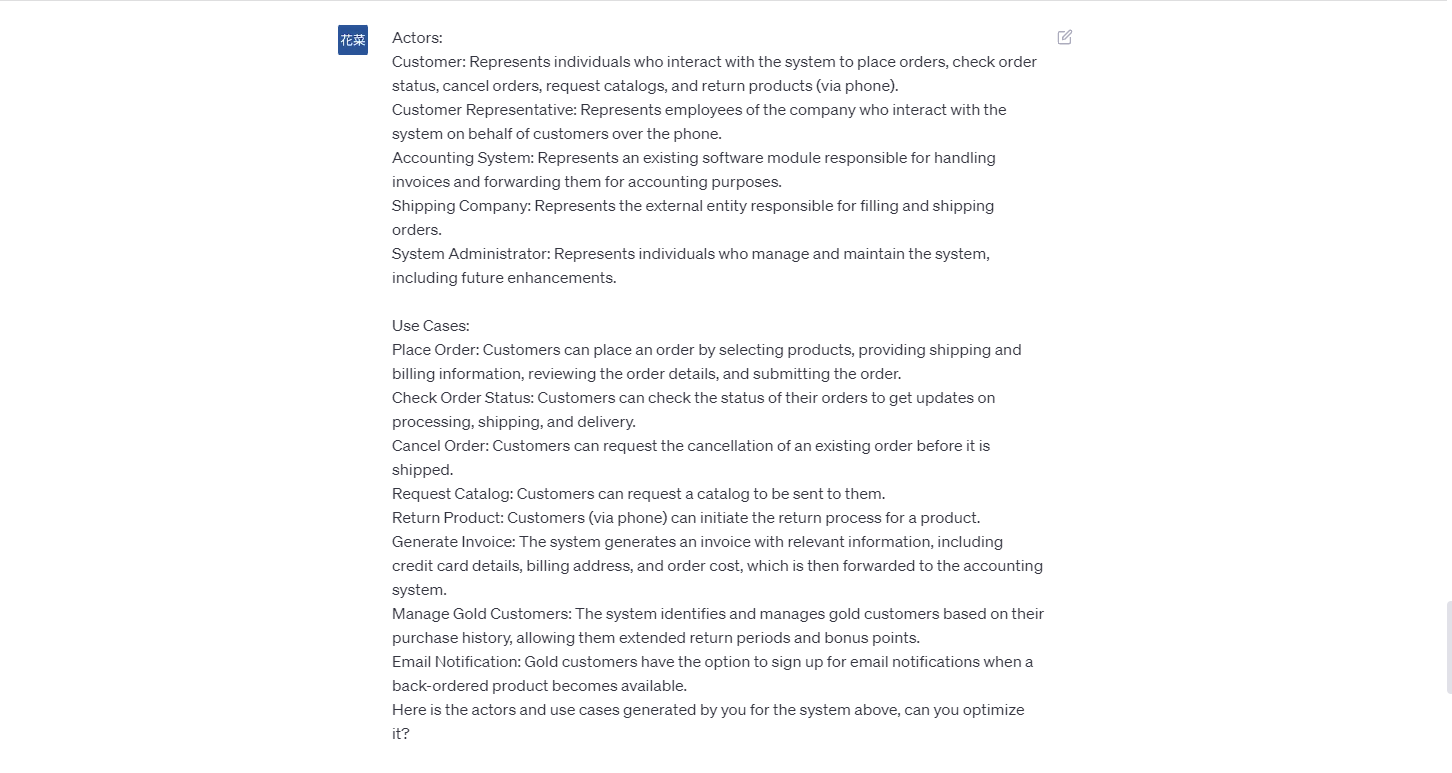
**Sequence Diagram:**

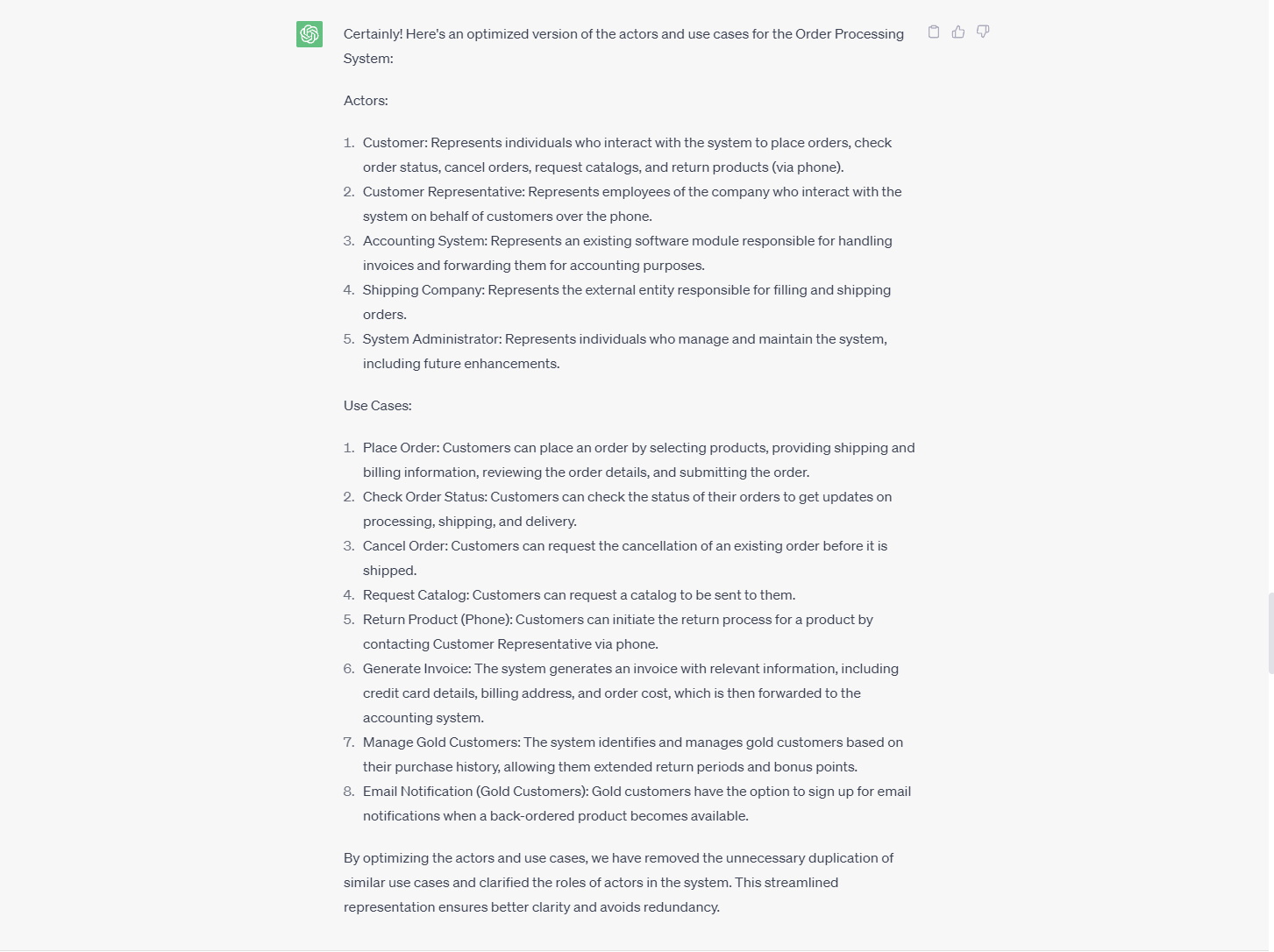


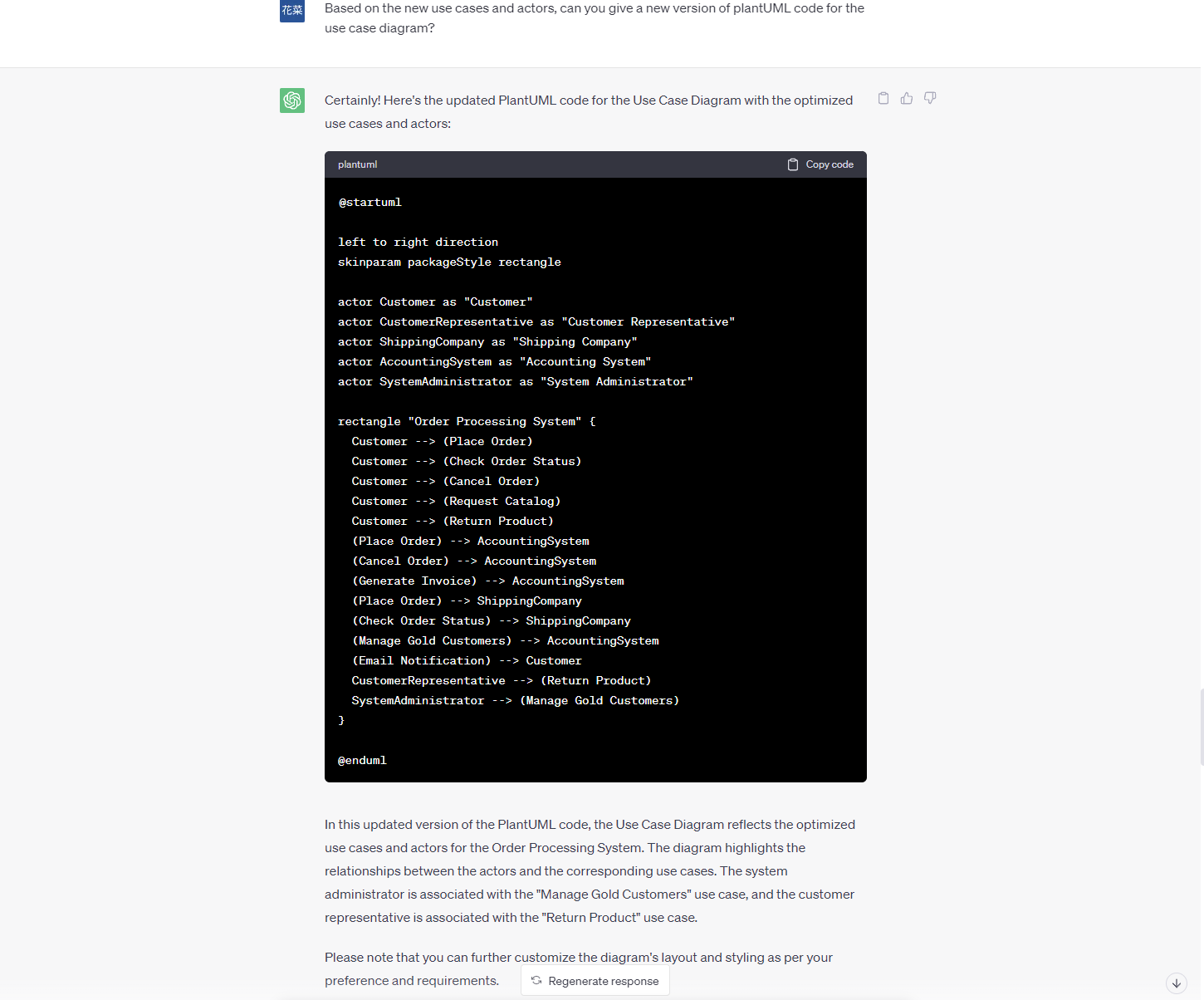


**一次提问。  
  
  
  
  
  
  
  
  
  
  
优化方案构建chatGPT辅助过程**

**Actor / Use case / Use case diagram：**

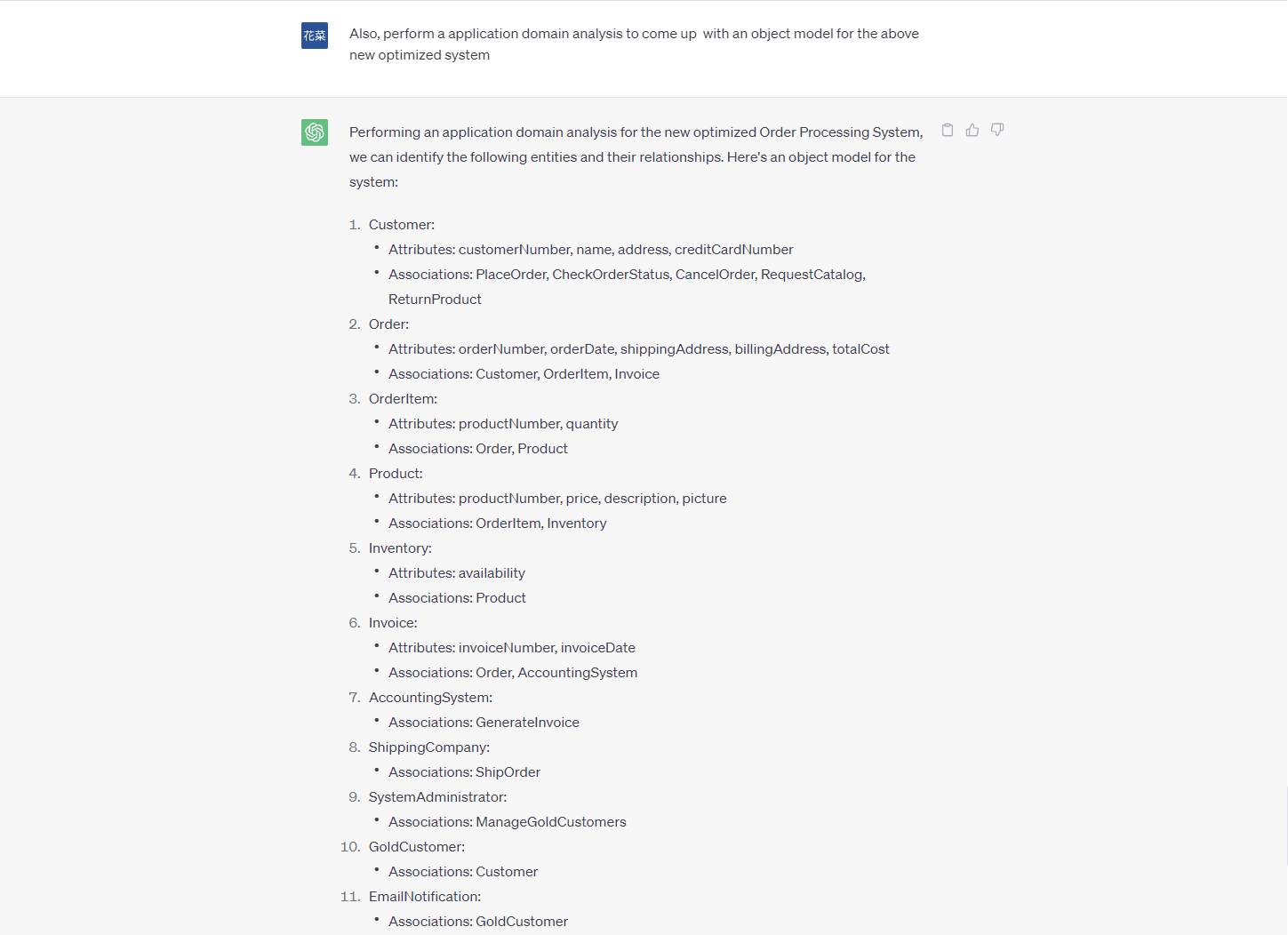


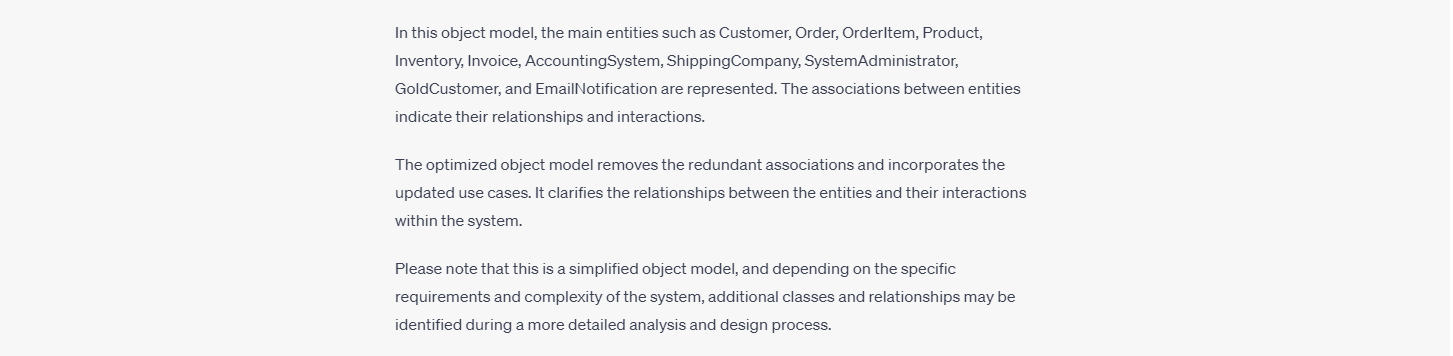


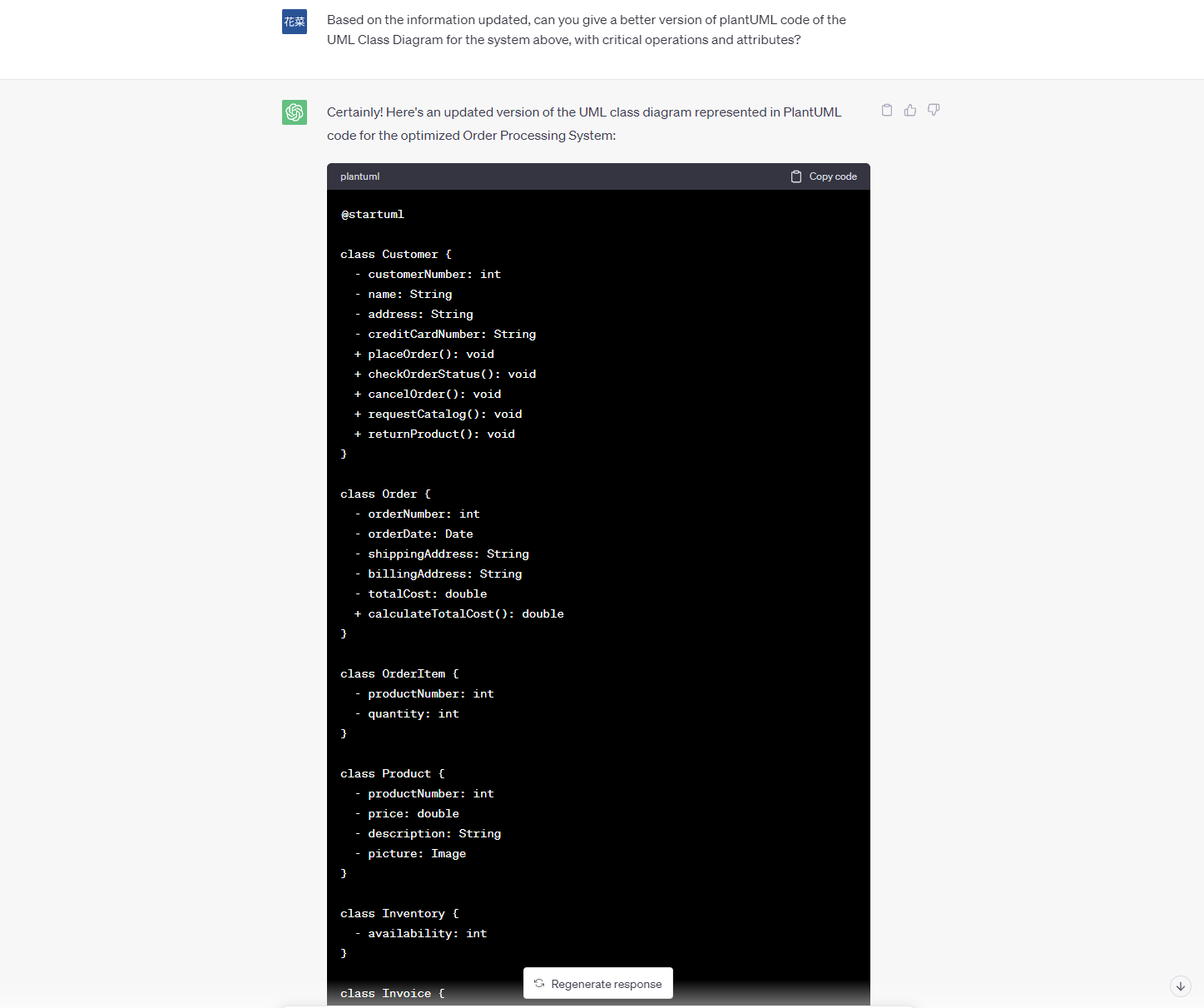


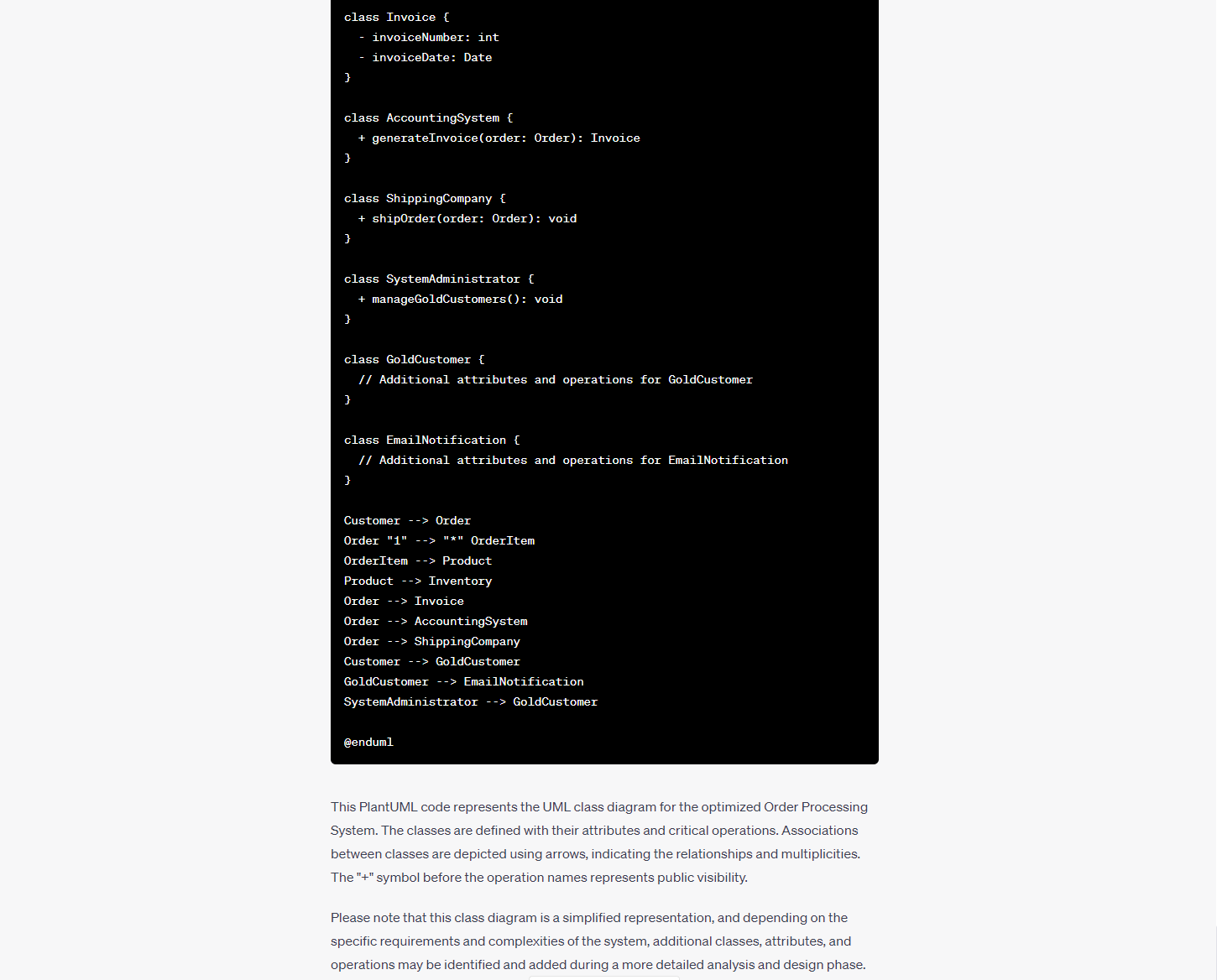
**共计两次提问。**

**Object model / Class diagram:**



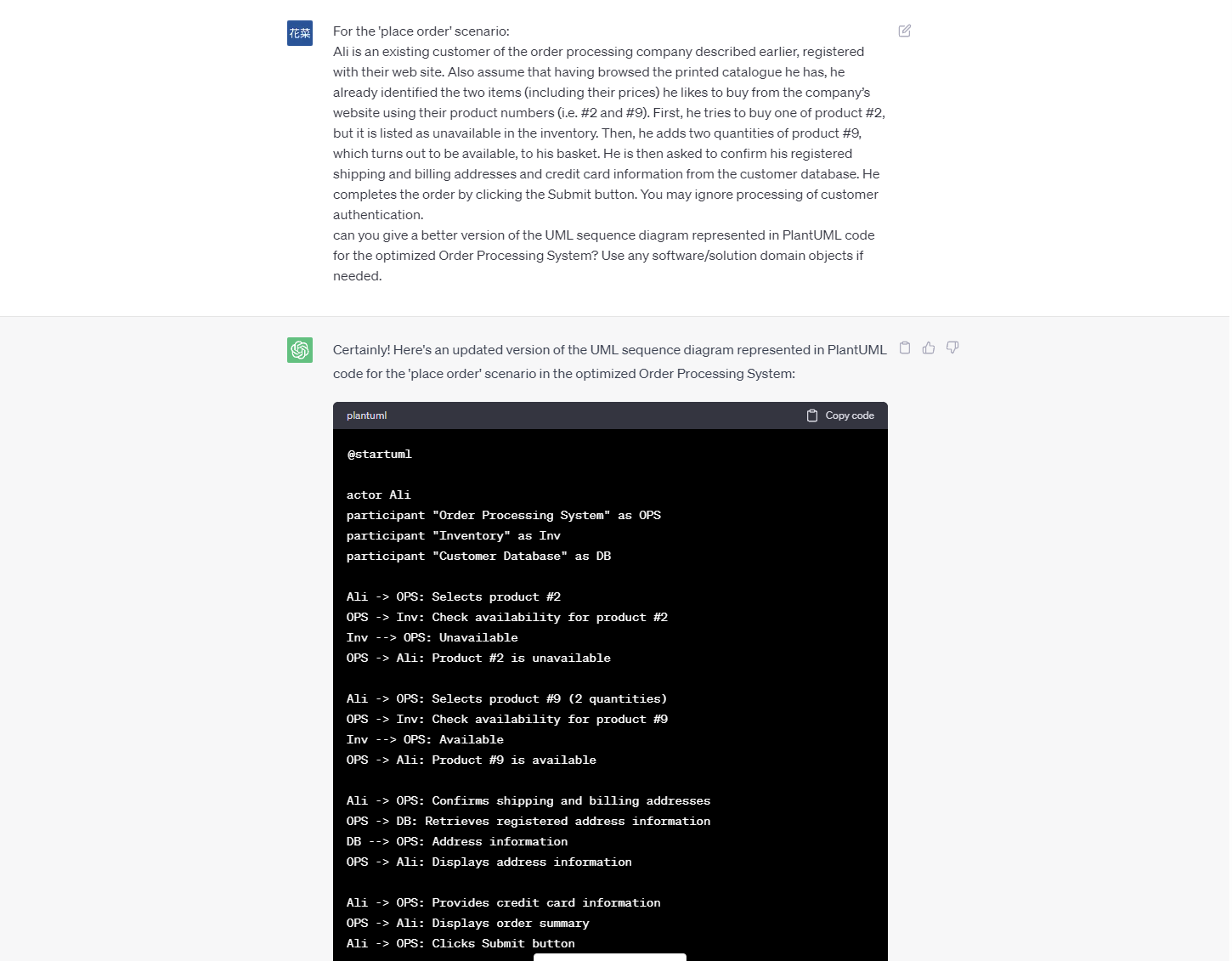


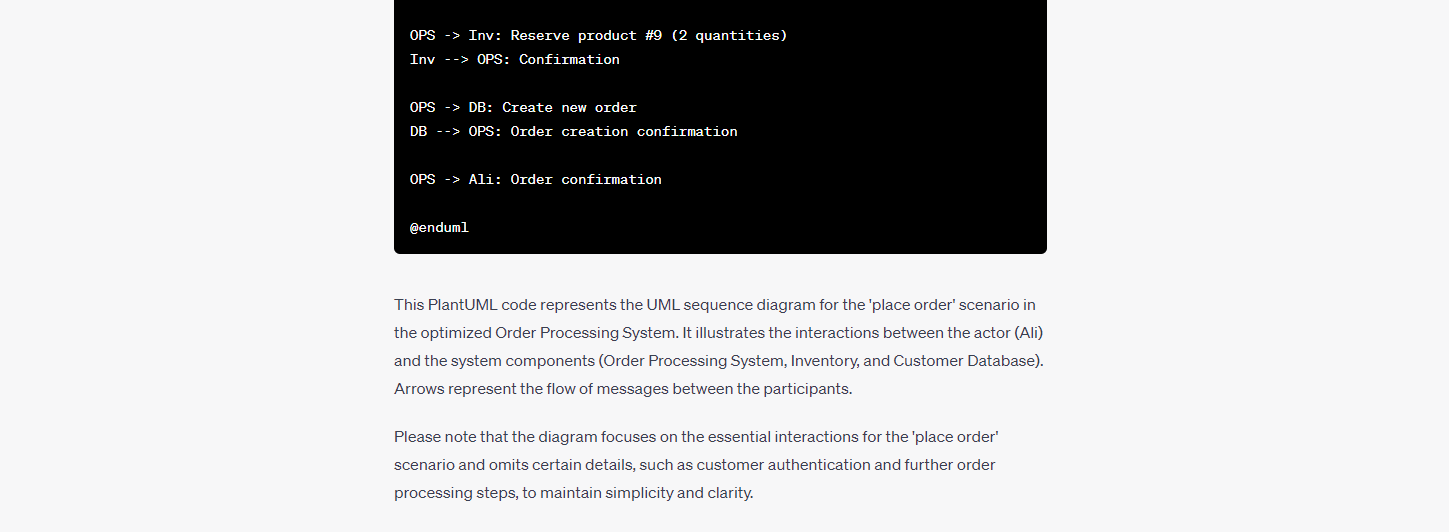




**共计两次提问。**

**Sequence Diagram:**





**一次提问。**