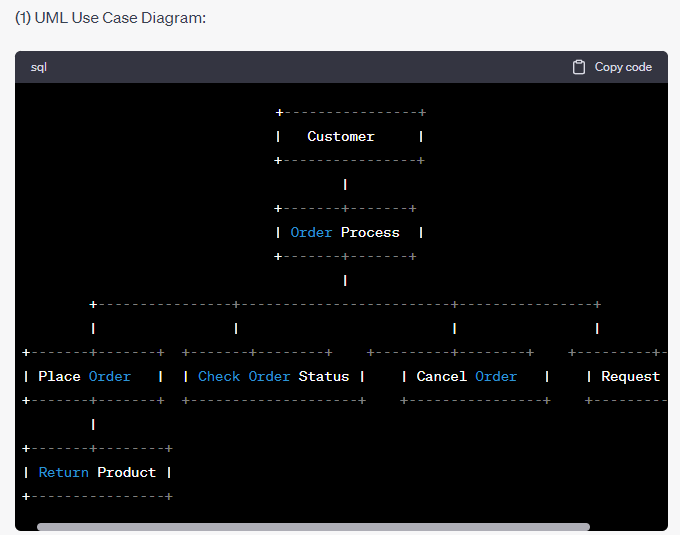
**最终方案：**

1.



**Actors:**

Customer

**Use Cases:**

Place Order

Check Order Status

Cancel Order

Request Catalog

Return Product

2.



Critical Operations:

Customer:

Get/Set customer number, name, and address.

Order Process:

Get/Set customer.

CatalogItem:

Get/Set product number, price, description, picture, and availability.

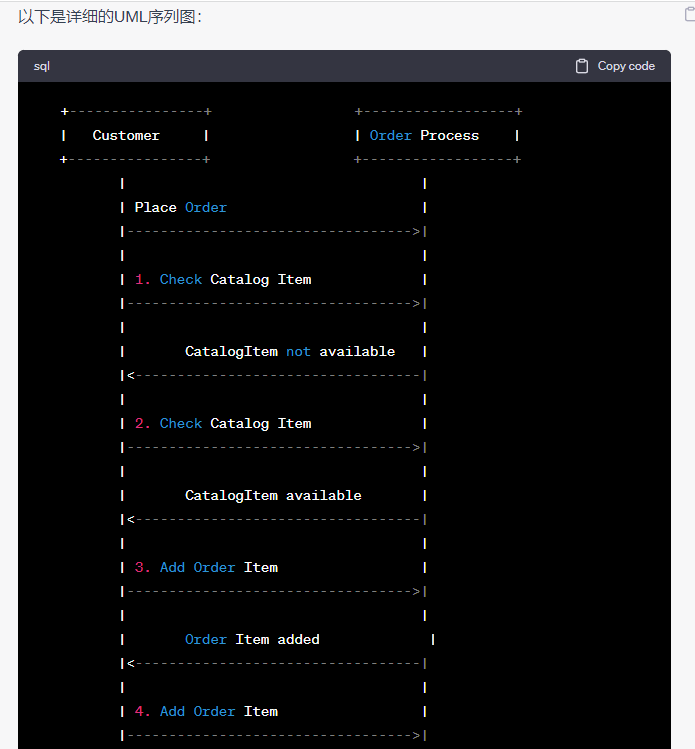
OrderItem:

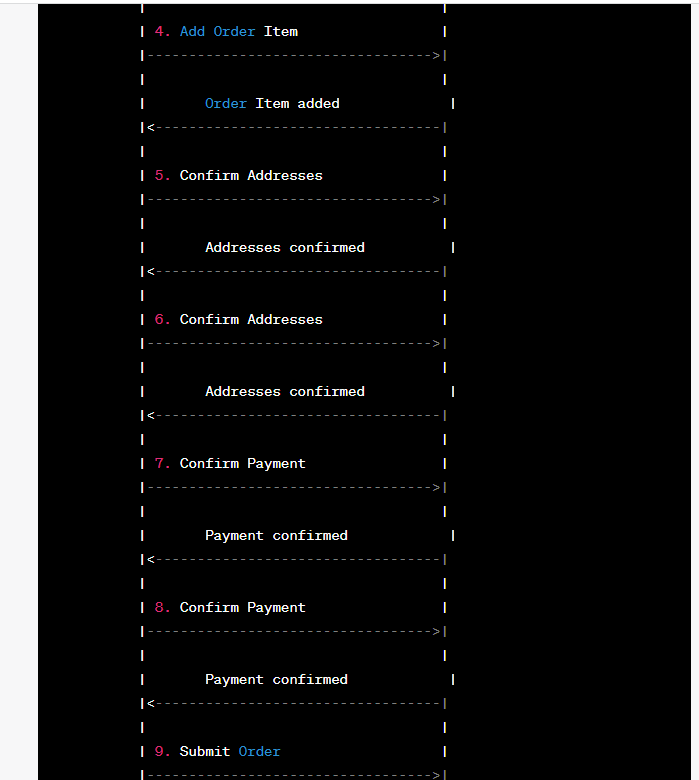
Get/Set product and quantity.

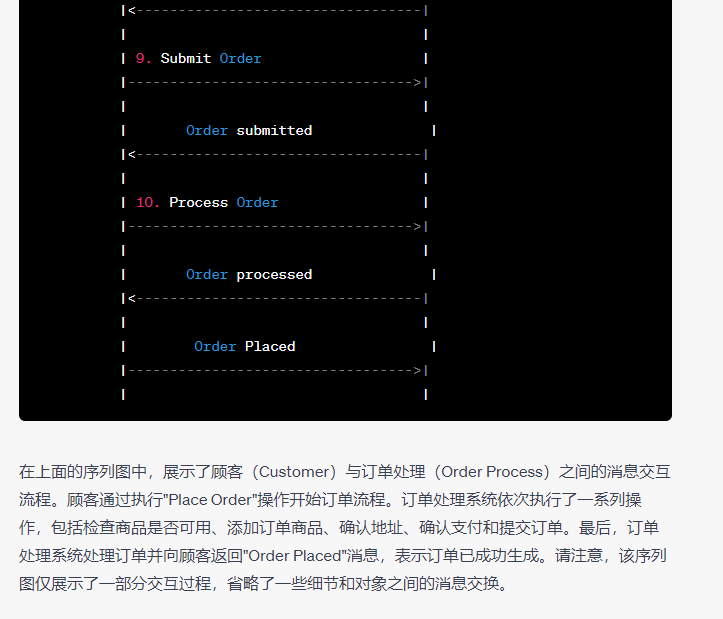
ShippingAddress:

Get/Set address.

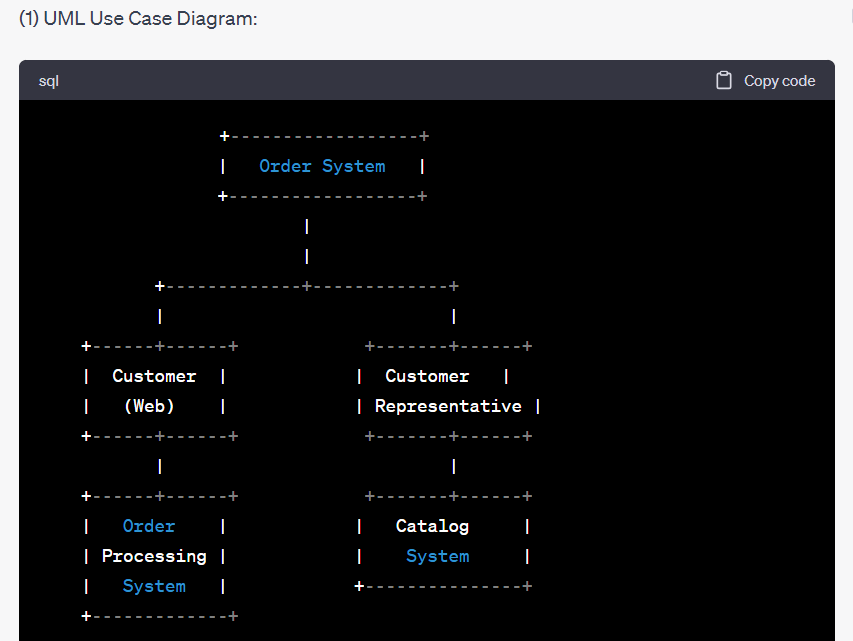
3.

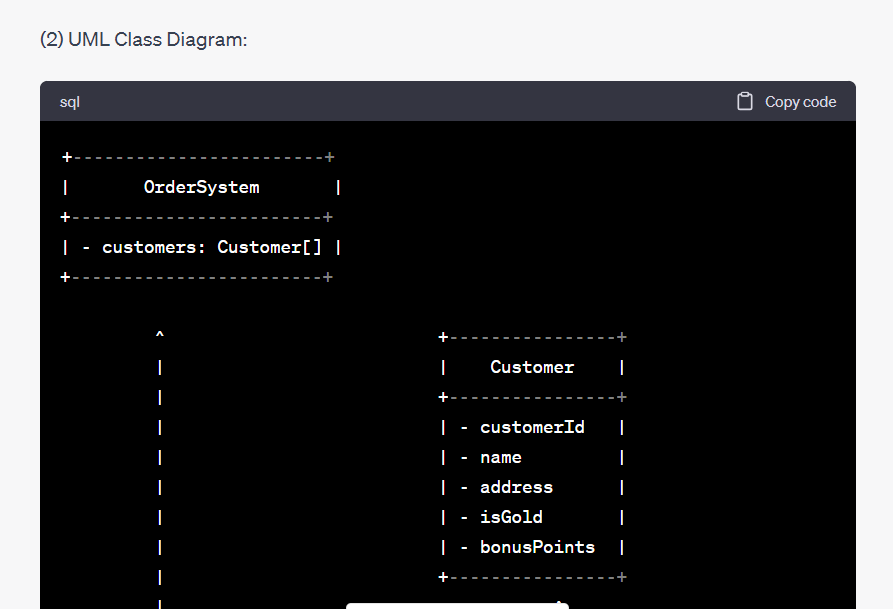


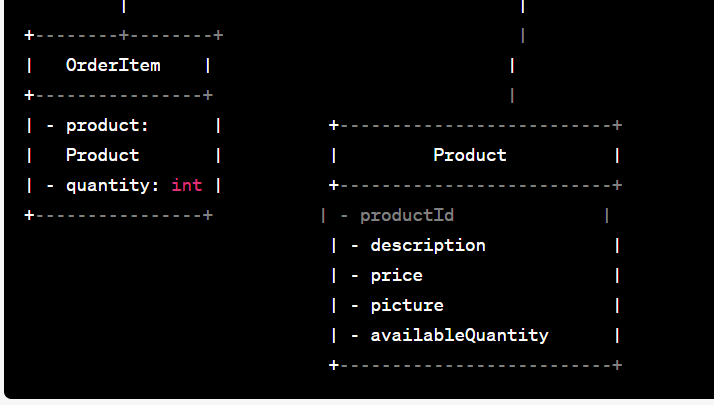


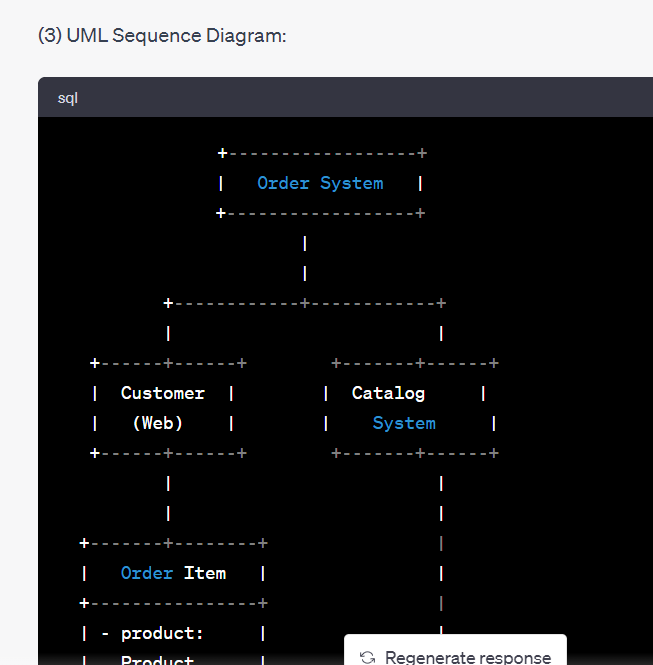


**初始方案：**











**区别：**在初始方案中，用例图没有给出actor和对应的用例。最终方案中给出了actor和custom，且用例图更完整。在初始方案中，类图不完整且没有给出具体函数，最终方案更完整。在初始方案中，时序图不符合格式，且没有给出消息传递和具体函数，最终版本中更加完整。

**提问记录：**

共提问20次，初始的几次效果过差，不能给出有效信息，没有记录。花费时间1小时左右。

**问：**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. (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. 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.

**答：**

(1) UML Use Case Diagram:

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

| Order System |

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

|

|

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

| |

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

| Customer | | Customer |

| (Web) | | Representative |

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

| |

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

| Order | | Catalog |

| Processing | | System |

| System | +---------------+

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

Actors:

* Customer (Web)
* Customer Representative

Use Cases:

* Place Order
* Check Order Status
* Cancel Order
* Request Catalog
* Return Product (Phone only)

(2) UML Class Diagram:

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

| OrderSystem |

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

| - customers: Customer[] |

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

^ +----------------+

| | Customer |

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

| | - customerId |

| | - name |

| | - address |

| | - isGold |

| | - bonusPoints |

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

| ^

| |

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

| OrderItem | |

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

| - product: | +--------------------------+

| Product | | Product |

| - quantity: int | +--------------------------+

+----------------+ | - productId |

| - description |

| - price |

| - picture |

| - availableQuantity |

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

Critical Operations:

* Customer:
  + **placeOrder()**
  + **checkOrderStatus()**
  + **cancelOrder()**
  + **requestCatalog()**
  + **returnProduct()**
  + **getCustomerId()**
  + **getName()**
  + **getAddress()**
  + **isGold()**
  + **getBonusPoints()**
* OrderItem:
  + **getProduct()**
  + **getQuantity()**
* Product:
  + **getProductId()**
  + **getDescription()**
  + **getPrice()**
  + **getPicture()**
  + **getAvailableQuantity()**

(3) UML Sequence Diagram:

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

| Order System |

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

|

|

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

| |

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

| Customer | | Catalog |

| (Web) | | System |

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

| |

| |

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

| Order Item | |

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

| - product: | |

| Product | |

| - quantity: 2 | |

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

| |

| |

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

| Customer | |

| Database | |

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

| |

| |

| |

| |

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

|

|

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

| Order System |

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

| - customers[] |

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

|

|

|

|

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

| Order System |

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

| - customers[] |

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

**问：**重新完成第三小问，要求显示时序图中对象的交互

**答：**请注意，由于文字格式的限制，无法以图形方式显示序列图。以下是描述该特定场景的序列图说明：

Customer (Web)：Ali选择要购买的产品并点击提交按钮。

调用OrderSystem的placeOrder()操作。

传递OrderItem对象（包含所选产品和数量）作为参数。

返回订单确认信息。

OrderSystem：

创建新订单。

检查产品#2的库存。

检查产品#9的库存。

更新OrderItem的产品信息和数量。

调用CustomerDatabase的操作以获取Ali的注册信息。

检查并更新客户的金牌状态。

生成订单号并返回给Customer (Web)。

Customer (Web)：

确认订单。

点击提交按钮。

OrderSystem：

接收订单确认。

创建Invoice对象。

将订单信息传递给AccountingSystem。

AccountingSystem：

接收订单信息并生成发票。

OrderSystem：

将发票信息传递给ShippingCompany。

ShippingCompany：

接收发票信息并准备订单。

总结：这个序列图显示了Customer (Web)与OrderSystem之间的交互，以及OrderSystem与CustomerDatabase、AccountingSystem和ShippingCompany之间的交互。通过这个序列图，可以更好地理解系统在特定场景下的行为和信息流动。

**问：**重新完成第三小问

**答：**

请注意，由于文字格式的限制，无法以图形方式显示序列图。以下是描述该特定场景的序列图说明：

Customer (Web)：Ali选择要购买的产品并点击提交按钮。

发送placeOrder()消息给OrderSystem。

传递OrderItem对象（包含所选产品和数量）作为参数。

OrderSystem：接收placeOrder()消息。

创建新订单。

检查产品#2的库存。

检查产品#9的库存。

更新OrderItem的产品信息和数量。

调用CustomerDatabase的操作以获取Ali的注册信息。

检查并更新客户的金牌状态。

生成订单号。

OrderSystem：向Customer (Web)发送订单确认信息。

Customer (Web)：接收订单确认信息。

确认订单。

发送confirmOrder()消息给OrderSystem。

OrderSystem：接收confirmOrder()消息。

创建Invoice对象。

将订单信息传递给AccountingSystem。

AccountingSystem：接收订单信息。

生成发票。

OrderSystem：将发票信息传递给ShippingCompany。

ShippingCompany：接收发票信息。

准备订单。

总结：这个序列图描述了在特定场景下，Customer (Web)与OrderSystem之间的交互，以及OrderSystem与CustomerDatabase、AccountingSystem和ShippingCompany之间的消息传递。通过这个序列图，可以更好地理解系统中各个对象之间的交互过程和消息流动。

**问：**重新完成第三小问，给出UML Sequence Diagram:

**答：**由于文字格式的限制，无法以图形方式显示序列图。以下是描述该特定场景的序列图说明：

Customer (Web)：Ali选择要购买的产品并点击提交按钮。

调用OrderSystem的placeOrder(orderItem)操作。

传递OrderItem对象（包含所选产品和数量）作为参数。

OrderSystem：接收placeOrder(orderItem)消息。

创建新订单。

检查产品#2的库存。

检查产品#9的库存。

更新OrderItem的产品信息和数量。

调用CustomerDatabase的操作以获取Ali的注册信息。

检查并更新客户的金牌状态。

生成订单号。

OrderSystem：向Customer (Web)发送订单确认信息。

Customer (Web)：接收订单确认信息。

确认订单。

发送confirmOrder()消息给OrderSystem。

OrderSystem：接收confirmOrder()消息。

创建Invoice对象。

将订单信息传递给AccountingSystem。

AccountingSystem：接收订单信息。

生成发票。

OrderSystem：将发票信息传递给ShippingCompany。

ShippingCompany：接收发票信息。

准备订单。

总结：这个序列图描述了在特定场景下，Customer (Web)与OrderSystem之间的交互，以及OrderSystem与CustomerDatabase、AccountingSystem和ShippingCompany之间的消息传递。通过这个序列图，可以更好地理解系统中各个对象之间的交互过程和消息流动。

**问：**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. (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. 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.

**答：**(1) UML Use Case Diagram:

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

| Customer |

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

|

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

| Order Process |

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

|

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

| | | |

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

| Place Order | | Check Order Status | | Cancel Order | | Request Catalog |

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

|

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

| Return Product |

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

Actors:

* Customer

Use Cases:

* Place Order
* Check Order Status
* Cancel Order
* Request Catalog
* Return Product

(2) UML Class Diagram:

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

| Customer |

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

| - customerNumber |

| - name |

| - address |

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

|

|

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

| Order Process |

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

| - customer |

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

|

|

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

| | | | CatalogItem |

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

| OrderItem | | ShippingAddress| | | - productNumber|

+------------------+ +----------------+ | | - price |

| - product | +------>| - description |

| - quantity | | - picture |

+------------------+ | - availability|

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

Critical Operations:

* Customer:
  + Get/Set customer number, name, and address.
* Order Process:
  + Get/Set customer.
* CatalogItem:
  + Get/Set product number, price, description, picture, and availability.
* OrderItem:
  + Get/Set product and quantity.
* ShippingAddress:
  + Get/Set address.

(3) UML Sequence Diagram:

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

| Customer |

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

|

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

| Order Process |

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

|

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

| |

+-------v-------+ +-------v-------+

| CatalogItem | | OrderItem |

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

| |

+-------v-------+ |

| ShippingAddress| |

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

| |

+-------v-------+ |

| OrderItem | |

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

| |

+-------v-------+ |

| ShippingAddress| |

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

| |

+-------v-------+ |

| OrderItem | |

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

| |

+-------v-------+ |

| ShippingAddress| |

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

| |

+-------v-------+ |

| OrderItem | |

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

| |

+-------v-------+ |

| ShippingAddress| |

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

| |

+-------v-------+ |

| Order Process | |

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

| |

+-------v-------+ |

| CatalogItem | |

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

| |

+-------v-------+ |

| Order Process | |

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

| |

+-------v-------+ |

| CatalogItem | |

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

| |

+-------v-------+ |

| Order Process | |

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

| |

+-------v-------+ |

| CatalogItem | |

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

| |

+-------v-------+ |

| Order Process | |

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

| |

+-------v-------+ |

| CatalogItem | |

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

| |

+-------v-------+ |

| Order Process | |

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

| |

+-------v-------+ |

| CatalogItem | |

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

| |

+-------v-------+ |

| Order Process | |

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

| |

+-------v-------+ |

| CatalogItem | |

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

| |

+-------v-------+ |

| Order Process | |

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

| |

+-------v-------+ |

| CatalogItem | |

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

| |

+-------v-------+ |

| Order Process | |

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

| |

+-------v-------+ |

| CatalogItem | |

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

| |

+-------v-------+ |

| Order Process | |

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

| |

+-------v-------+ |

| CatalogItem | |

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

| |

+-------v-------+ |

| Order Process | |

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

| |

+-------v-------+ |

| CatalogItem | |

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

| |

+-------v-------+ |

| Order Process | |

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

| |

+-------v-------+ |

| CatalogItem | |

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

| |

+-------v-------+ |

| Order Process | |

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

| |

+-------v-------+ |

| CatalogItem | |

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

| |

+-------v-------+ |

| Order Process | |

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

| |

+-------v-------+ |

**问：**重新完成第三小问

**答：**

抱歉，下面是重新完成的第三小问的UML序列图：

```

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

| Customer |

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

|

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

| Order Process |

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

|

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

| CatalogItem |

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

|

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

| Order Process |

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

|

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

| CatalogItem |

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

|

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

| Order Process |

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

|

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

| CatalogItem |

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

|

+

**问：**重新完成第三小问，给出UML时序图

**答：下面是重新完成的第三小问的UML序列图：**

**```**

**+----------------+ +------------------+**

**| Customer | | Order Process |**

**+----------------+ +------------------+**

**| |**

**| Place Order |**

**|------------------------------->|**

**| |**

**| 1. Check Catalog Item |**

**|------------------------------->|**

**| |**

**| 2. Check Catalog Item |**

**|<-------------------------------|**

**| |**

**| 3. Add Order Item |**

**|------------------------------->|**

**| |**

**| 4. Add Order Item |**

**|<-------------------------------|**

**| |**

**| 5. Confirm Addresses |**

**|------------------------------->|**

**| |**

**| 6. Confirm Addresses |**

**|<-------------------------------|**

**| |**

**| 7. Confirm Payment |**

**|------------------------------->|**

**| |**

**| 8. Confirm Payment |**

**|<-------------------------------|**

**| |**

**| 9. Submit Order |**

**|------------------------------->|**

**| |**

**| 10. Process Order |**

**|------------------------------->|**

**| |**

**| Order Placed |**

**|<-------------------------------|**

**| |**

**```**

**注意：这是一个简化的序列图，省略了一些细节和对象之间的消息交换。**

**问：**给出详细的序列图

**答：以下是详细的UML序列图：**

**```**

**+----------------+ +------------------+**

**| Customer | | Order Process |**

**+----------------+ +------------------+**

**| |**

**| Place Order |**

**|---------------------------------->|**

**| |**

**| 1. Check Catalog Item |**

**|---------------------------------->|**

**| |**

**| CatalogItem not available |**

**|<----------------------------------|**

**| |**

**| 2. Check Catalog Item |**

**|---------------------------------->|**

**| |**

**| CatalogItem available |**

**|<----------------------------------|**

**| |**

**| 3. Add Order Item |**

**|---------------------------------->|**

**| |**

**| Order Item added |**

**|<----------------------------------|**

**| |**

**| 4. Add Order Item |**

**|---------------------------------->|**

**| |**

**| Order Item added |**

**|<----------------------------------|**

**| |**

**| 5. Confirm Addresses |**

**|---------------------------------->|**

**| |**

**| Addresses confirmed |**

**|<----------------------------------|**

**| |**

**| 6. Confirm Addresses |**

**|---------------------------------->|**

**| |**

**| Addresses confirmed |**

**|<----------------------------------|**

**| |**

**| 7. Confirm Payment |**

**|---------------------------------->|**

**| |**

**| Payment confirmed |**

**|<----------------------------------|**

**| |**

**| 8. Confirm Payment |**

**|---------------------------------->|**

**| |**

**| Payment confirmed |**

**|<----------------------------------|**

**| |**

**| 9. Submit Order |**

**|---------------------------------->|**

**| |**

**| Order submitted |**

**|<----------------------------------|**

**| |**

**| 10. Process Order |**

**|---------------------------------->|**

**| |**

**| Order processed |**

**|<----------------------------------|**

**| |**

**| Order Placed |**

**|---------------------------------->|**

**| |**

**```**

**在上面的序列图中，展示了顾客（Customer）与订单处理（Order Process）之间的消息交互流程。顾客通过执行"Place Order"操作开始订单流程。订单处理系统依次执行了一系列操作，包括检查商品是否可用、添加订单商品、确认地址、确认支付和提交订单。最后，订单处理系统处理订单并向顾客返回"Order Placed"消息，表示订单已成功生成。请注意，该序列图仅展示了一部分交互过程，省略了一些细节和对象之间的消息交换。**