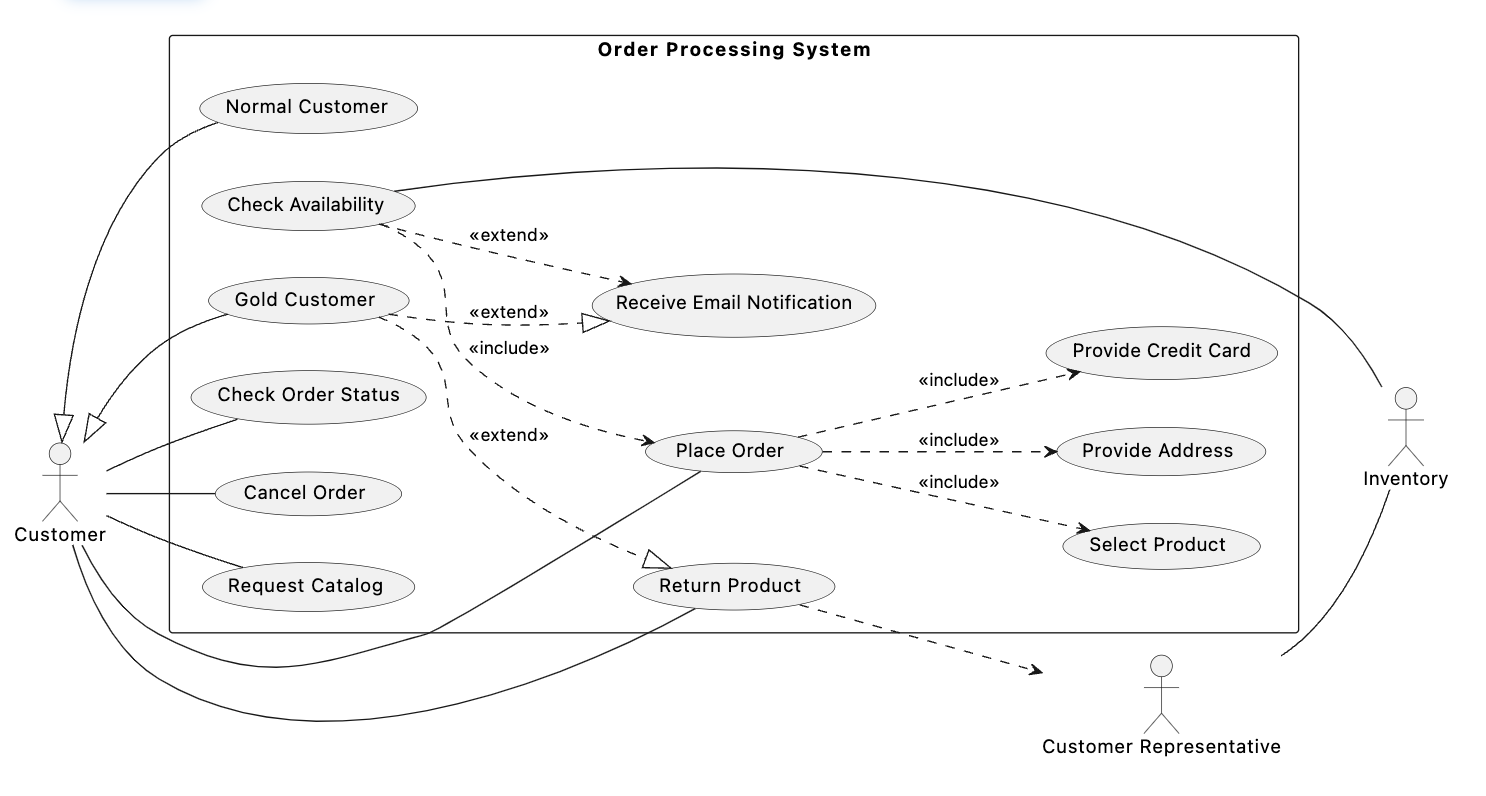
**Case study: Order Processing System**

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.

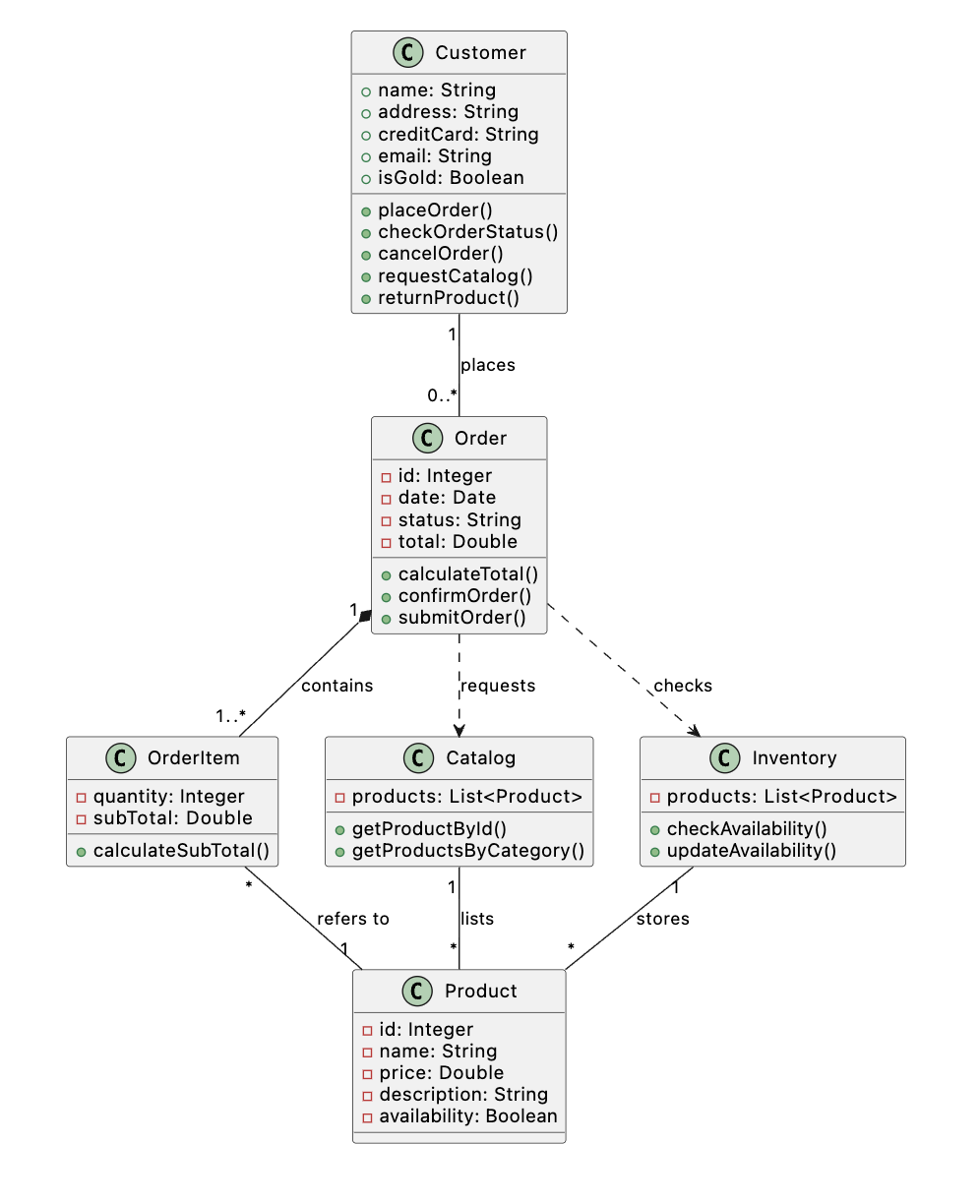
(1). Identify actors and use cases for the system described above and show them on a UML Use Case Diagram.



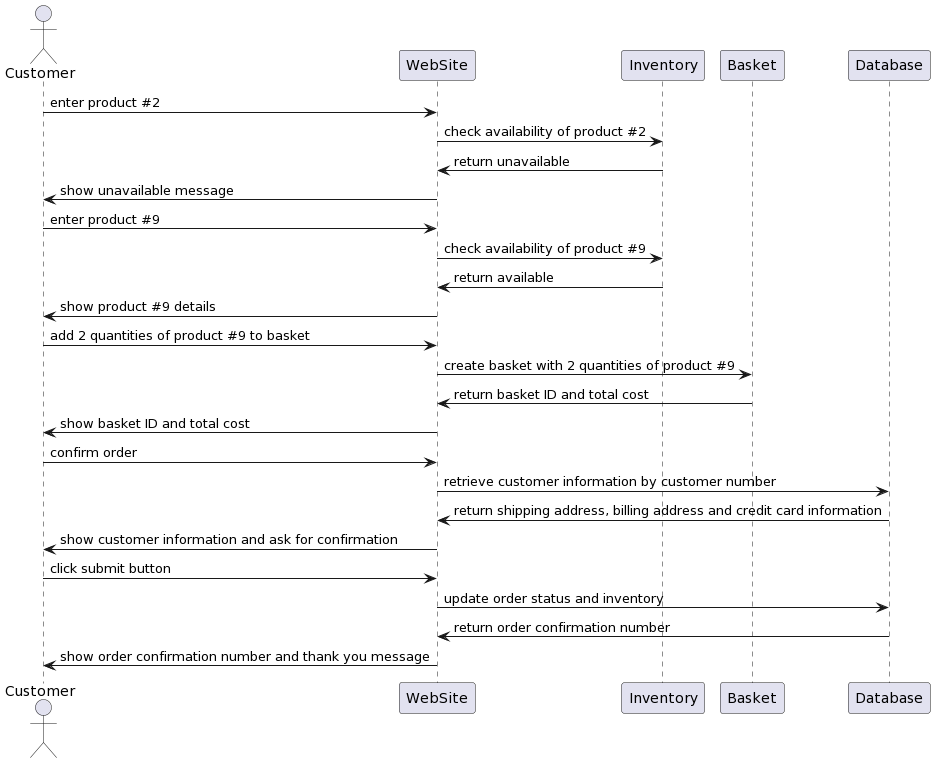
(2) Perform a quick application domain analysis to come up with an object model for the above system. Express your findings with a UML Class Diagram, making sure to identify any critical operations of classes.

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.



附：与newbing的交流记录

请你阅读以下材料。在你阅读完成后，我会给你关于这段材料的任务。

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.

好的，我已经阅读了这段材料。请问你想让我做什么任务呢？😊

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

[1]: https://support.microsoft.com/en-us/office/create-a-uml-use-case-diagram-92cc948d-fc74-466c-9457-e82d62ee1298 "Create a UML use case diagram - Microsoft Support"

[2]: https://www.ibm.com/docs/en/rsar/9.5?topic=diagrams-use-case "Use-case diagrams in UML modeling - IBM"

[3]: https://www.lucidchart.com/pages/uml-use-case-diagram "UML Use Case Diagram Tutorial | Lucidchart"

[4]: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/ "What is Use Case Diagram? - Visual Paradigm"

根据我从网络上搜索到的信息[^1^][2] [^2^][3] [^3^][4]，UML用例图是一种用来描述系统的行为和需求的图形化工具。用例图中有两种主要的元素：\*\*参与者\*\*和\*\*用例\*\*。参与者是指与系统交互的人或其他系统，用例是指系统提供的功能或服务。参与者和用例之间用\*\*关联\*\*线连接，表示参与者可以使用或影响用例。用例图还可以使用其他元素来表示更复杂的关系，例如\*\*泛化\*\*，\*\*包含\*\*，\*\*扩展\*\*和\*\*子系统\*\*。

对于你给出的问题描述，我根据以下步骤来识别参与者和用例，并绘制了一个简单的UML用例图：

1. 从问题描述中找出所有可能的参与者。我发现有三个参与者：客户，客服代表和库存系统。

2. 从问题描述中找出所有可能的用例。我发现有六个用例：下订单，查询订单状态，取消订单，请求目录，退货和接收邮件通知。

3. 为每个参与者和用例画一个图形符号，并在符号下方写上名称。参与者用一个人形符号表示，用例用一个椭圆形符号表示。

4. 为每个参与者和用例之间的关联画一条直线，并在线上标注方向箭头。箭头指向参与者表示参与者发起了这个用例，指向用例表示参与者接收了这个用例的结果。

5. 为一些特殊的关系画出相应的符号，并在符号上写上名称。例如，客户是一个泛化的参与者，可以分为普通客户和金牌客户，所以我用一条带空心三角形的直线连接它们，并在三角形上写上“泛化”。金牌客户有一些额外的权利，例如退货期限更长，所以我用一条带实心三角形的虚线连接它们，并在三角形上写上“扩展”。下订单这个用例包含了选择产品，提供地址和信用卡信息等子步骤，所以我用一条带空心菱形的虚线连接它们，并在菱形上写上“包含”。接收邮件通知这个用例只有当产品缺货时才会发生，所以我用一条带实心菱形的虚线连接它们，并在菱形上写上“扩展”。

6. 为整个系统或其主要组成部分画一个矩形框，并在框内放置所有相关的参与者和用例。我把整个系统命名为“订单处理系统”，并把所有的参与者和用例放在框内。

以下是我绘制的UML用例图：

```uml

@startuml

left to right direction

skinparam packageStyle rectangle

actor Customer

actor "Customer Representative" as CR

actor Inventory

rectangle "Order Processing System" {

Customer <|-- (Normal Customer)

Customer <|-- (Gold Customer)

(Gold Customer) ..|> (Return Product) : <<extend>>

(Gold Customer) ..|> (Receive Email Notification) : <<extend>>

Customer -- (Place Order)

(Place Order) ..> (Select Product) : <<include>>

(Place Order) ..> (Provide Address) : <<include>>

(Place Order)

好的，我继续绘制UML用例图：

```uml

(Place Order) ..> (Provide Credit Card) : <<include>>

Customer -- (Check Order Status)

Customer -- (Cancel Order)

Customer -- (Request Catalog)

Customer -- (Return Product)

(Return Product) ..> CR

CR -- Inventory

Inventory -- (Check Availability)

(Check Availability) ..> (Place Order) : <<include>>

(Check Availability) ..> (Receive Email Notification) : <<extend>>

}

@enduml

```

这是我根据问题描述和搜索结果绘制的UML用例图。请问你对我的回答有什么意见或建议吗？😊

非常好！接下来请完成有关之前材料的第二个任务：

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.

[1]: https://en.wikipedia.org/wiki/Class\_diagram "Class diagram - Wikipedia"

[2]: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-class-diagram-tutorial/ "UML Class Diagram Tutorial - Visual Paradigm"

[3]: https://www.lucidchart.com/pages/uml-class-diagram "UML Class Diagram Tutorial | Lucidchart"

根据我从网络上搜索到的信息[^1^][1] [^2^][2] [^3^][3]，UML类图是一种用来描述系统的静态结构的图形化工具。类图中有三种主要的元素：\*\*类\*\*，\*\*属性\*\*和\*\*操作\*\*。类是指系统中的一个实体或概念，属性是指类的特征或状态，操作是指类的行为或功能。类图还可以使用其他元素来表示类之间的关系，例如\*\*关联\*\*，\*\*聚合\*\*，\*\*组合\*\*，\*\*继承\*\*和\*\*依赖\*\*。

对于你给出的问题描述，我根据以下步骤来进行应用领域分析，并绘制了一个简单的UML类图：

1. 从问题描述和用例场景中找出所有可能的类。我发现有六个类：客户，产品，订单，订单项，目录和库存。

2. 为每个类画一个矩形框，并在框内写上类的名称。框内分为三个部分：第一部分写上类的名称，第二部分写上类的属性，第三部分写上类的操作。属性和操作之前加上可见性符号（+表示公开，-表示私有）。

3. 为每个类之间的关系画出相应的符号，并在线上标注方向箭头和多重性。箭头指向被依赖或被继承的类，多重性表示一个类实例可以与另一个类实例有多少个关联。关系有以下几种类型：

- 关联：表示两个类之间有一种结构化的连接，用一条直线连接两个类，并在线上写上关联的名称或角色。

- 聚合：表示一个类是另一个类的部分，但两者可以独立存在，用一条带空心菱形的直线连接两个类，并把菱形放在整体的一端。

- 组合：表示一个类是另一个类的部分，但两者不能独立存在，用一条带实心菱形的直线连接两个类，并把菱形放在整体的一端。

- 继承：表示一个类是另一个类的子类，继承了其属性和操作，用一条带空心三角形的直线连接两个类，并把三角形放在父类的一端。

- 依赖：表示一个类需要另一个类提供某些服务或信息，用一条带箭头的虚线连接两个类，并在箭头上写上依赖的类型或原因。

4. 根据问题描述和用例场景中给出的信息，为每个类确定其属性和操作，并为每个关系确定其名称或角色和多重性。

以下是我绘制的UML类图：

```uml

@startuml

class Customer {

+name: String

+address: String

+creditCard: String

+email: String

+isGold: Boolean

+placeOrder()

+checkOrderStatus()

+cancelOrder()

+requestCatalog()

+returnProduct()

}

class Product {

-id: Integer

-name: String

-price: Double

-description: String

-availability: Boolean

}

class Order {

好的，我继续绘制UML类图：

```uml

-id: Integer

-date: Date

-status: String

-total: Double

+calculateTotal()

+confirmOrder()

+submitOrder()

}

class OrderItem {

-quantity: Integer

-subTotal: Double

+calculateSubTotal()

}

class Catalog {

-products: List<Product>

+getProductById()

+getProductsByCategory()

}

class Inventory {

-products: List<Product>

+checkAvailability()

+updateAvailability()

}

Customer "1" -- "0..\*" Order : places

Order "1" \*-- "1..\*" OrderItem : contains

OrderItem "\*" -- "1" Product : refers to

Catalog "1" -- "\*" Product : lists

Inventory "1" -- "\*" Product : stores

Order ..> Catalog : requests

Order ..> Inventory : checks

@enduml

```

这是我根据问题描述和搜索结果绘制的UML类图。请问你对我的回答有什么意见或建议吗？😊

非常好！接下来请完成第三个任务：

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

[1]: https://www.lucidchart.com/pages/uml-sequence-diagram "UML Sequence Diagram Tutorial | Lucidchart"

[2]: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/ "What is Sequence Diagram? - Visual Paradigm"

[3]: https://support.microsoft.com/en-us/office/create-a-uml-sequence-diagram-c61c371b-b150-4958-b128-902000133b26 "Create a UML sequence diagram - Microsoft Support"

根据搜索结果[^1^][1] [^2^][2]，UML序列图是一种交互图，用于描述操作是如何执行的。它们捕捉了对象在协作上下文中的交互。序列图是时间焦点，它们通过使用图的垂直轴来表示时间，显示了消息是什么以及何时发送的。

为了绘制这个特定的场景的UML序列图，我使用了一个UML序列模板[^3^][3]，并添加了一些软件/解决方案域对象，如Customer, WebSite, Inventory, Basket, Database等。我假设你已经定义了这些对象的属性和方法。

下面是我生成的UML序列图：

```uml

@startuml

actor Customer as Ali

participant WebSite

participant Inventory

participant Basket

participant Database

Ali -> WebSite : enter product #2

WebSite -> Inventory : check availability of product #2

Inventory -> WebSite : return unavailable

WebSite -> Ali : show unavailable message

Ali -> WebSite : enter product #9

WebSite -> Inventory : check availability of product #9

Inventory -> WebSite : return available

WebSite -> Ali : show product #9 details

Ali -> WebSite : add 2 quantities of product #9 to basket

WebSite -> Basket : create basket with 2 quantities of product #9

Basket -> WebSite : return basket ID and total cost

WebSite -> Ali : show basket ID and total cost

Ali -> WebSite : confirm order

WebSite -> Database : retrieve customer information by customer number

Database -> WebSite : return shipping address, billing address and credit card information

WebSite -> Ali : show customer information and ask for confirmation

Ali -> WebSite : click submit button

WebSite -> Database : update order status and inventory

Database -> WebSite : return order confirmation number

WebSite -> Ali : show order confirmation number and thank you message

@enduml

```

请问你对这个UML序列图有什么问题或建议吗？😊

# 总结

本次实验在AI的辅助下极大地缩短了完成所需时间，全程只用了7分钟。另外，在PlantUML语言的支持下，newbing展现出了惊人的效率和准确性，所生成的UML图几乎不需任何迭代和优化，最初版本即最终版本。相信经过一段时间的发展，AI必然会在软件建模方面对人类产生极大的帮助。