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

Q1:

Please read the following case description first:

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. .......

A1:

Based on the provided case description, the mail-order company wants to automate its order processing system. Here are some key points and features mentioned:

Accessible via the web: The initial version of the system should be accessible to customers through a web interface. This allows customers to interact with the system online.

Phone interaction: Customers can also call the company by phone and interact with the system through a customer representative. This means there is a phone interface available for order processing.

more advice...

Q2:

Identify actors and use cases for the system described above and show them on a UML Use Case Diagram.Please give the PlantUML code.

A2:

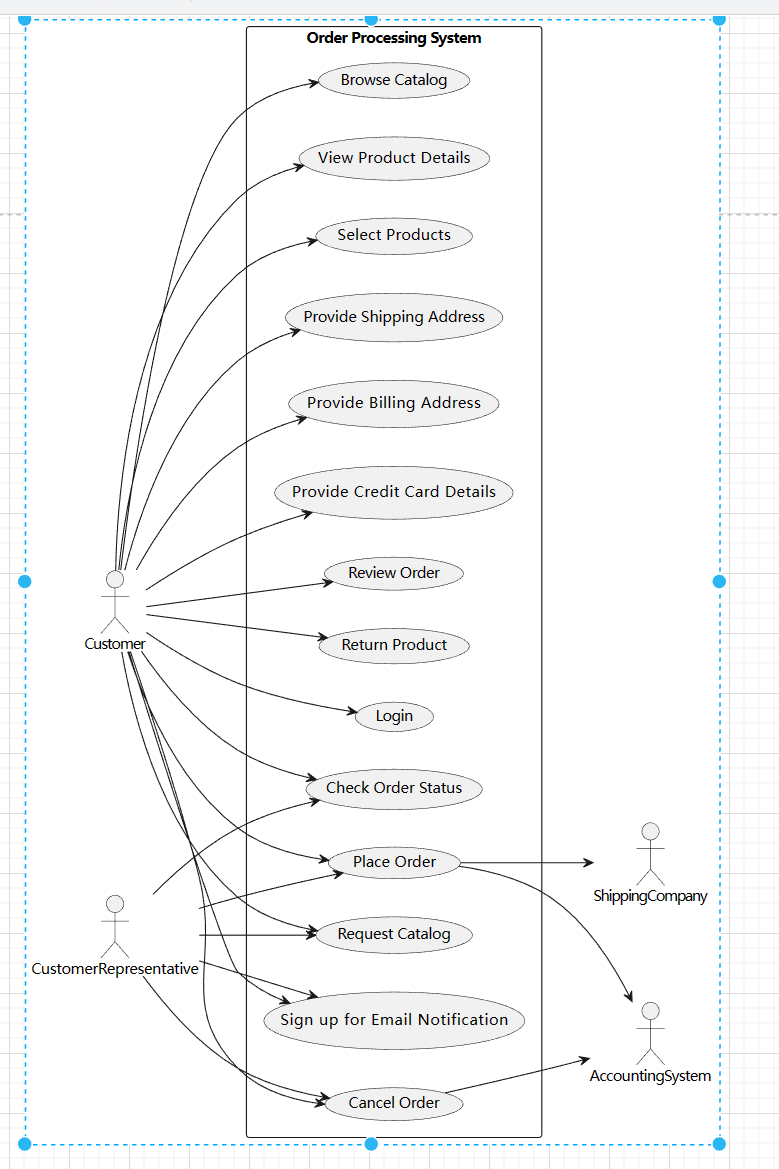
The diagram represents the actors and use cases for the system. The actors are represented by stick figures, and the use cases are shown as ovals. The relationships between the actors and use cases are indicated by arrows.

The Customer and CustomerRepresentative actors interact with various use cases such as Place Order, Check Order Status, Cancel Order, Request Catalog, Return Product, Sign up for Email Notification, Login, Browse Catalog, View Product Details, Select Products, Provide Shipping Address, Provide Billing Address, Provide Credit Card Details, and Review Order.

The AccountingSystem and ShippingCompany are also included as actors, as they interact with the Place Order and Cancel Order use cases.

Note that the diagram doesn't show all possible relationships or details, but it provides a high-level overview of the actors and their interactions with the system.

Generated case diagram:



Q3:

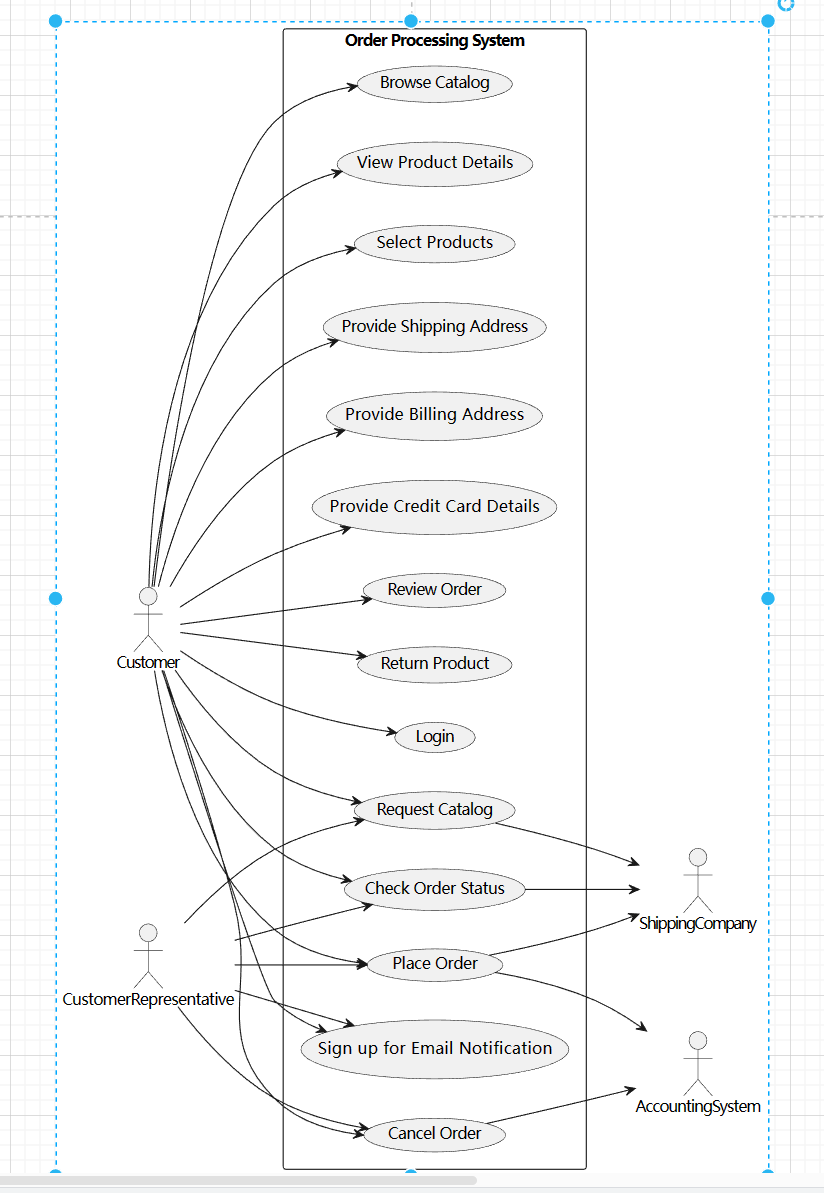
Are you sure that the Use case diagram you give can fully meet all requirements? If not, please optimize。GIve the PlantUMLcode

A3:

This updated diagram includes the missing relationships between actors and use cases. The Customer and CustomerRepresentative actors now have the appropriate relationships with the Check Order Status, Request Catalog, and Sign up for Email Notification use cases. Additionally, the ShippingCompany actor is associated with the Check Order Status and Request Catalog use cases.

Please note that the diagram still provides a high-level representation and may not capture all the intricate details of the system. However, it now includes the necessary relationships based on the given requirements.

Generated use case diagram:

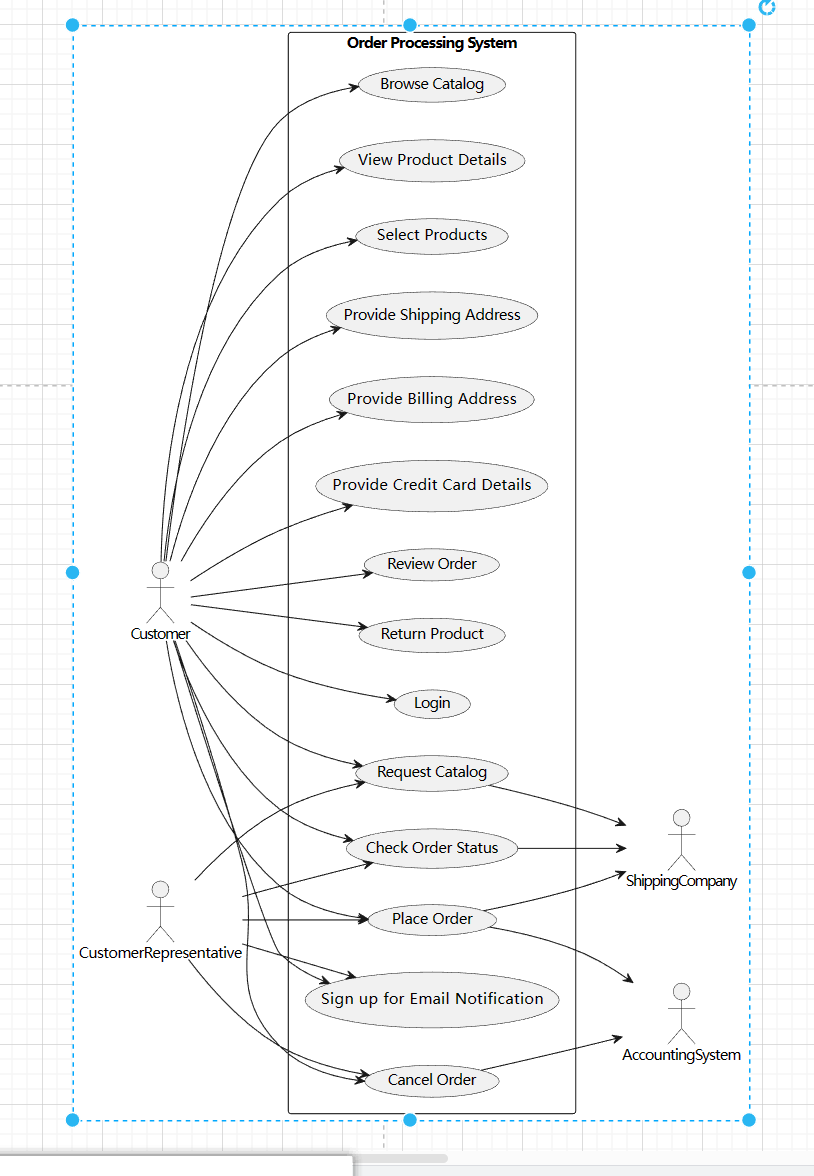


Q4:

Considering other improvement methods for Use case diagram, improve the original Use case diagram,give the plantuml code

A4:

Ultimate diagram:



Differences:

1.Actor naming: In the initial plan, actors were not explicitly named. In the final plan, actors were named as "Customer" and "CustomerRepresentative," and their names were further shortened to "Cust" and "Rep" in the final plan for improved clarity and conciseness.

2.Missing relationships: In the initial plan, there were some missing relationships between actors and use cases. In the final plan, the missing relationships were added. For example, the relationships between CustomerRepresentative and Check Order Status, Request Catalog, and Sign up for Email Notification were included in the final plan.

3.Actor alignment: In the initial plan, the actors and use cases were not aligned in a specific direction. In the final plan, the diagram was adjusted to have a left-to-right direction for better readability and visual appeal.

4.Use case ordering: In the initial plan, the order of the use cases was not explicitly defined. In the final plan, the use cases were reordered for better logical flow. For example, the Place Order and Cancel Order use cases were placed together, and the Check Order Status and Request Catalog use cases were associated with the ShippingCompany actor.

Minutes:5

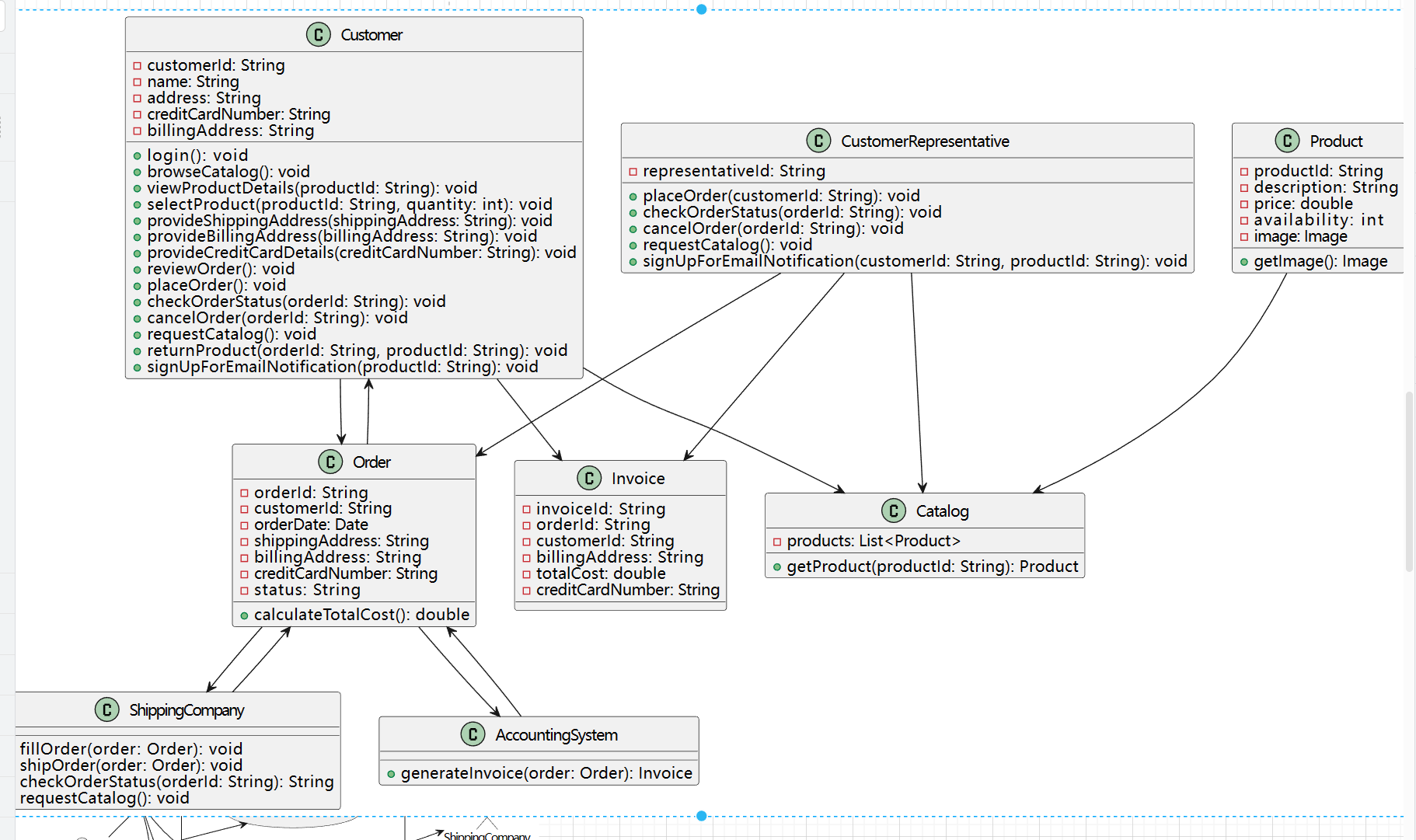
1. Perform a quick application domain analysis to come up with an object model for the above system. Express your findings with a UML Class Diagram, making sure to identify any critical operations of classes.

A5:

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.Give the plantuml code

Q5:

Generated diagram:

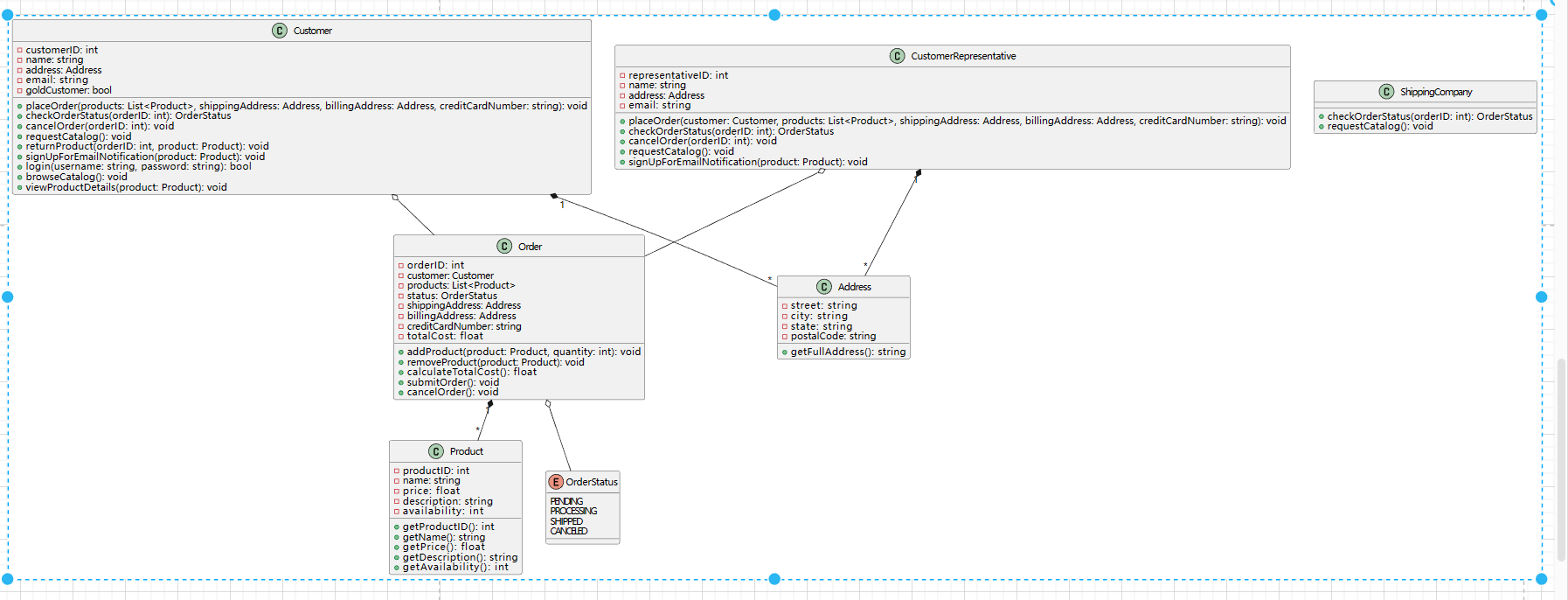


Q6:

Optimize class diagrams based on GRASP principles (including information experts, high cohesion, low coupling, creators, controllers, etc.),give the plantuml code

A6:

Generated diagram:



Differences:

Overall, the optimized class diagram differs from the initial version in the following aspects:

Information Expert Principles:

In the optimized version, the Order class introduces a customer object that represents the customer who placed the order. This is in line with the principle of information experts, as the Order class can directly access the required customer information.

High Cohesion and Low Coupling:

In the optimized version, the placeOrder method of the Customer class has been simplified to only accept the parameters required for placing an order. This improves cohesion, concentrates related behaviors within the same class, and reduces coupling between classes.

Creator principle:

In the optimized version, the placeOrder method of the Customer Representative class takes a Customer object as a parameter, indicating that the Customer Representative represents the customer placing the order. This is in line with the creator principle, where Customer Representative creates orders under the customer's name.

Remain unchanged for other classes:

The ShippingCompany class, Product class, OrderStatus enumeration, and Address class have not undergone significant changes in the optimized version, maintaining the same structure and functionality as the initial version.

Minutes:5

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

Q7:

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....

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.give the plantuml code

A7:

In this sequence diagram:

Ali, represented as an actor, interacts with the Web Application.

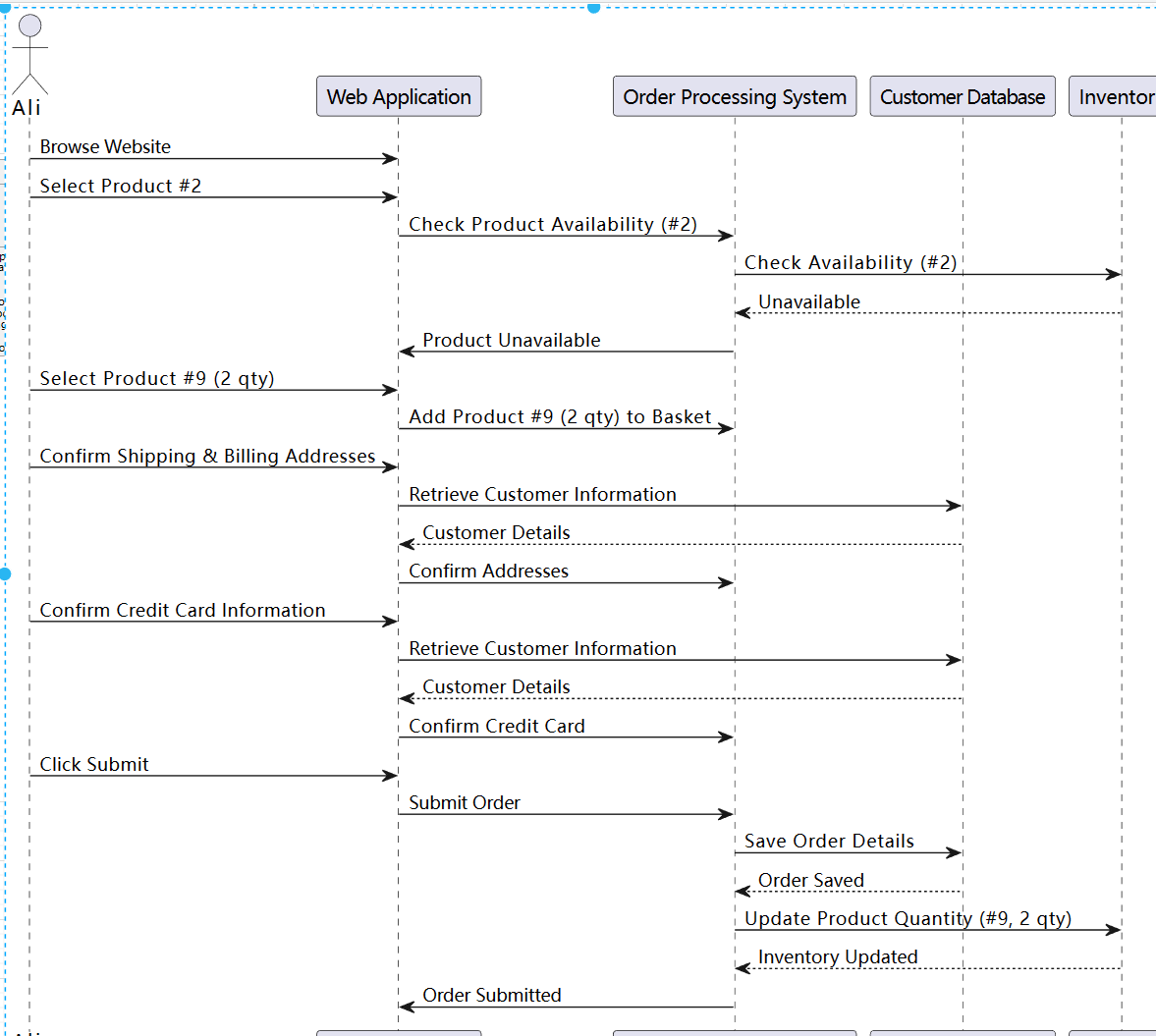
The Web Application communicates with the Order Processing System (OPS), Customer Database, and Inventory.

Ali browses the website, selects product #2 (unavailable), selects product #9 (2 qty), confirms shipping and billing addresses, confirms credit card information, and clicks the submit button to place the order.

The OPS checks product availability, retrieves customer information from the database, confirms addresses and credit card information, and submits the order.

The OPS saves the order details in the database, updates the product quantity in the inventory, and notifies the Web Application that the order has been submitted.

Generated diagram:

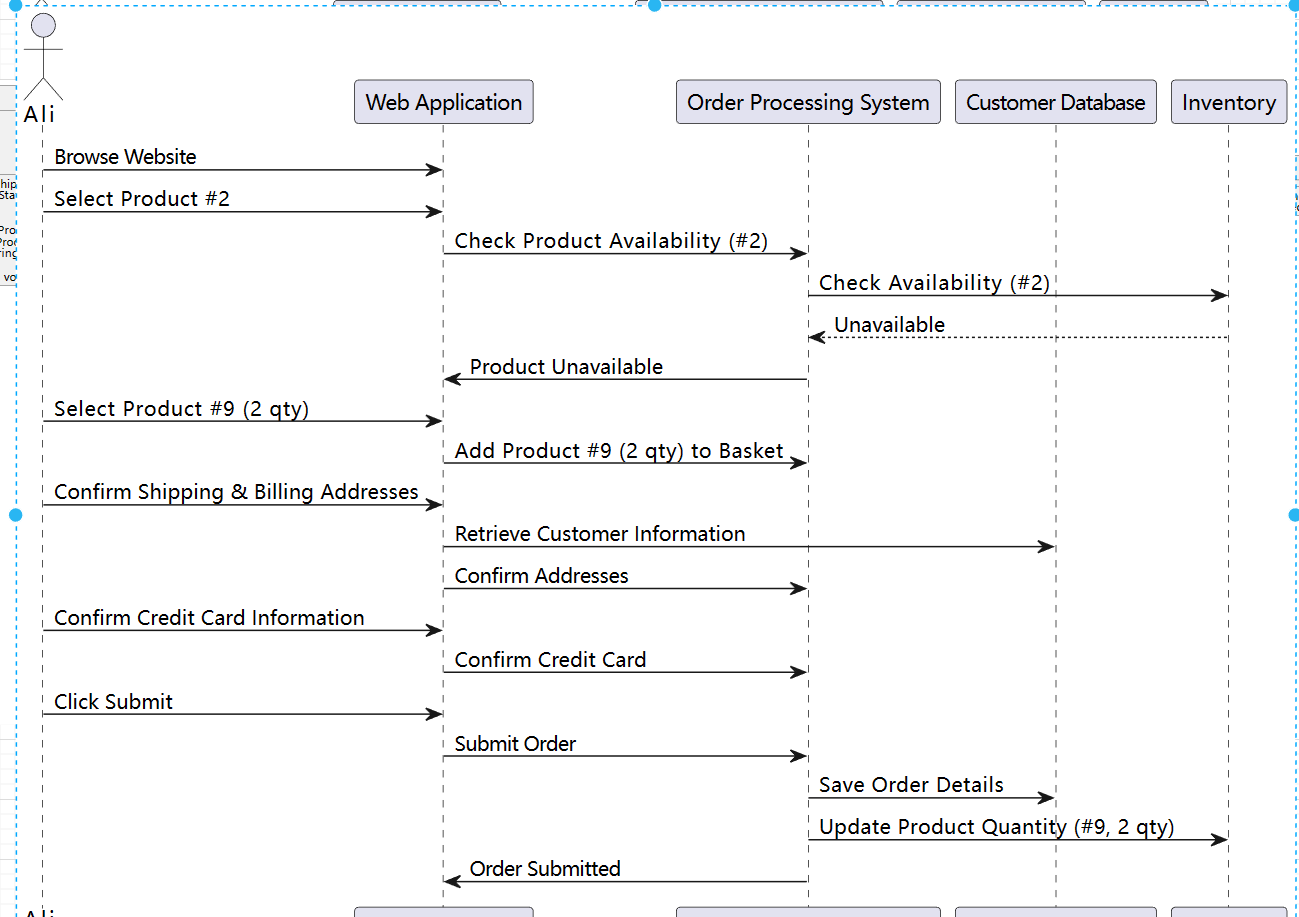


Q8:

Optimize the diagram with the goal of making it clearer, simpler, and easier to understand, highlighting key points and key interactions, and accurately representing the system's timing and interaction process.GIve the PlantUML code

A8:

Generated diagram:



Differences:

1.Simplification of interactions:

The optimized version removes unnecessary interactions and focuses on the key steps of the "place order" scenario, making it easier to understand.

2.Removal of customer authentication:

The optimized version omits the customer authentication process, as it is not directly relevant to the "place order" scenario.

Clearer representation of unavailable product:

3.In the optimized version, the interaction between OPS and Inventory explicitly shows that the product #2 is unavailable, providing clearer information to the user.

4.Streamlined confirmation steps:

The optimized version combines the confirmation of shipping and billing addresses into a single step, reducing the complexity and making the process more straightforward.

5.Simplified credit card confirmation:

The optimized version removes the redundant retrieval of customer information for credit card confirmation, streamlining the interaction between WebApp and OPS.

6.Focus on key interactions:

The optimized version highlights the essential interactions between the actors and the system, omitting less significant details and interactions that do not directly contribute to the understanding of the "place order" scenario.

Minutes:5