***EXERCISE 17***🡪 <https://online.visual-paradigm.com/diagrams/templates/use-case-diagram/restaurant-ordering-use-case-diagram/>

**PROMPT:** The following should be textually analyzed and a use case diagram created containing several use cases. Identify the actors, use cases and associations. Please, use the user goal level approach. Also, please consider any possible generalization relationship between use cases or between actors, and any possible "include" or "extend" relationship between use cases. Please give me the PlantUML code for the use case diagram corresponding to the following text:

***The Restaurant Ordering use case diagram provides a comprehensive visualization of the key functionalities and interactions within a restaurant's ordering process. The diagram presents a clear representation of the actions performed by the Chef, Waiter, and the overall flow of the order fulfillment process, ensuring smooth and efficient operations. The use case diagram encompasses a range of essential actions that occur during the restaurant ordering process. It illustrates the Waiter taking orders from customers, capturing their preferences, and ensuring accurate communication of the order details. The Waiter then passes the order to the Chef, who is responsible for preparing the requested dishes with precision and attention to detail. The Chef follows the recipes and cooking techniques to ensure the quality and taste of the food. Once the order is prepared, the Chef communicates with the Waiter to indicate that the order is ready for serving. The Waiter then takes the prepared order and serves it to the respective customer, ensuring prompt and attentive service. This step involves delivering the correct dishes to the right table and providing any necessary accompaniments or condiments requested by the customer.***

ChatGPT

**ChatGPT**

The actors in the Restaurant Ordering use case are Chef and Waiter, with the main use cases being Take Order, Communicate Order, Prepare Order, and Serve Order. Below is the PlantUML code for the use case diagram:

@startuml

actor Chef

actor Waiter

rectangle "Restaurant Ordering System" {

usecase "Take Order" as TakeOrder

usecase "Communicate Order" as CommunicateOrder

usecase "Prepare Order" as PrepareOrder

usecase "Serve Order" as ServeOrder

}

Chef --> CommunicateOrder

CommunicateOrder --> PrepareOrder

PrepareOrder --> ServeOrder

ServeOrder --> Chef

Waiter --> TakeOrder

TakeOrder --> CommunicateOrder

@enduml

**PROMPT:** In the provided code it is not clearly shown that the chef is the actor in charge of preparing the order. Also, you missed the secondary actor customer, who should be associated with the "serve order" use case. Also, you should add a use case for placing an order, whose actor should be the customer also.

ChatGPT

**ChatGPT**

Certainly, I've revised the use case diagram to clarify the association between the Chef and the "Prepare Order" use case, added the secondary actor "Customer," and included a new use case "Place Order" for the customer. Here's the updated PlantUML code:

@startuml

actor Chef

actor Waiter

actor Customer as "Customer"

rectangle "Restaurant Ordering System" {

usecase "Take Order" as TakeOrder

usecase "Communicate Order" as CommunicateOrder

usecase "Prepare Order" as PrepareOrder

usecase "Serve Order" as ServeOrder

usecase "Place Order" as PlaceOrder

}

Chef --> PrepareOrder

Waiter --> TakeOrder

Waiter --> CommunicateOrder

Chef --> CommunicateOrder

CommunicateOrder --> PrepareOrder

PrepareOrder --> ServeOrder

ServeOrder --> Customer : <<include>>

Customer --> PlaceOrder : <<include>>

PlaceOrder --> TakeOrder : <<extend>>

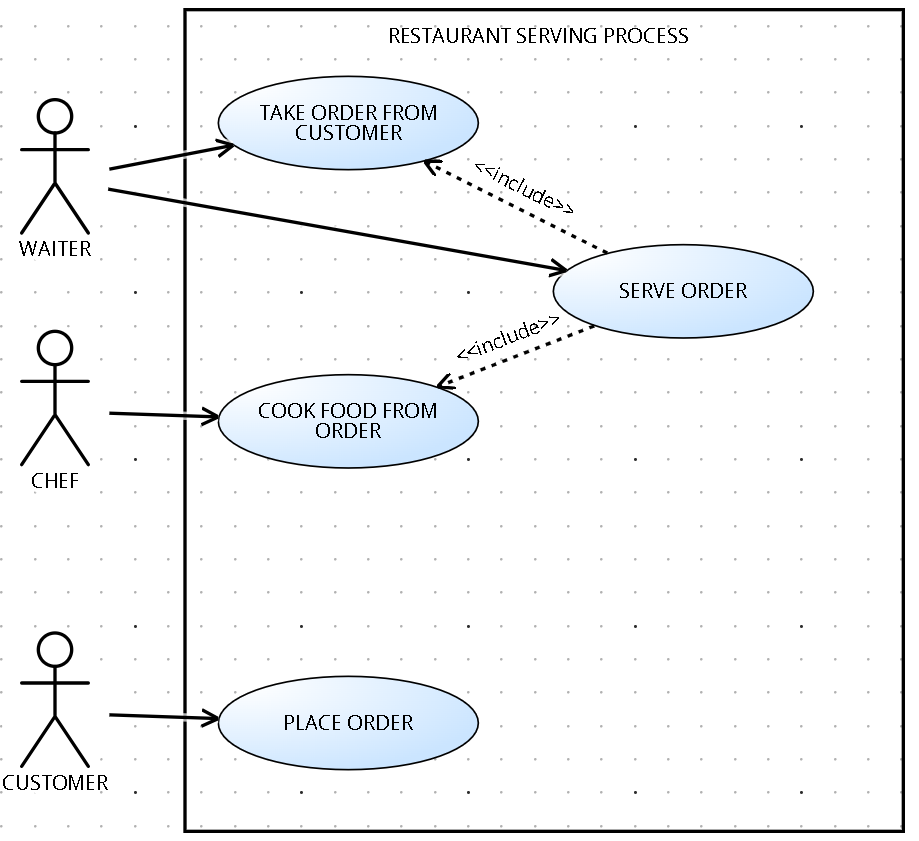
@enduml

**Analysis:**

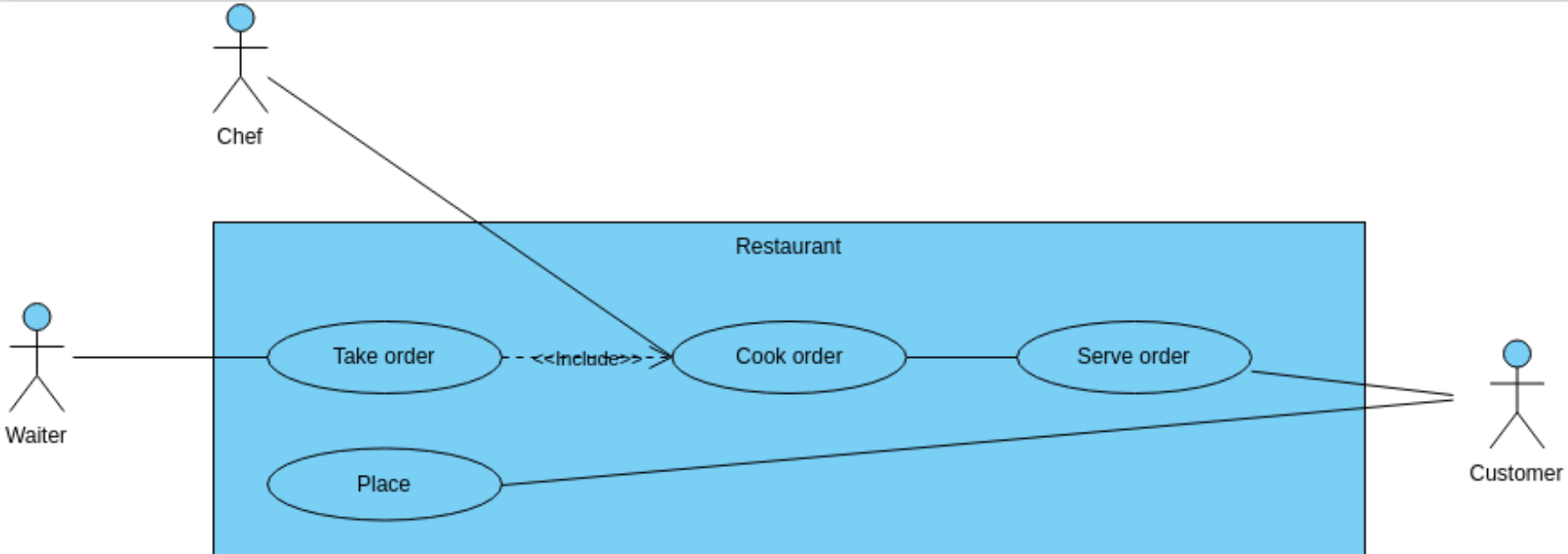
In this problem, the first code generated by ChatGPT was largely accurate and similar to the provided solution, with only a few minor errors. These issues were highlighted and requested to be addressed in the second prompt. Surprisingly, despite the second prompt specifically asking for precise issues to be solved, ChatGPT generated a new outcome that included more elements than were requested. The new Use Case Diagram (UCD) partially solved the problem, but it introduced new relationships that were fundamentally incorrect.

Notably, the second solution included "include" relationships between a use case and an actor, which is conceptually illogical. Interestingly, the first solution generated was simpler and more similar to the original one, in contrast to the second solution that led to confusion. This discrepancy highlights the challenges and limitations in guiding ChatGPT to provide accurate and focused responses during the iterative refinement process.

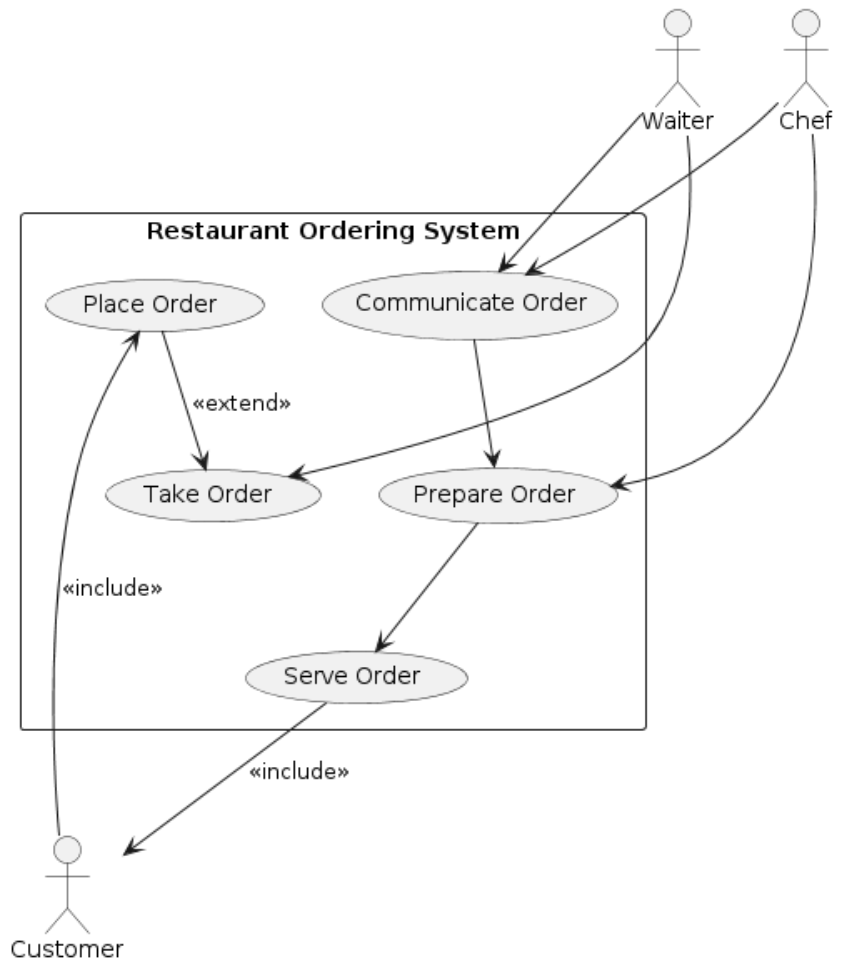
**MY SOLUTION:**

****

**REFERENCE SOLUTION:**

****

**LLM SOLUTION:**

****