***EXERCISE 20*** 🡪 <https://online.visual-paradigm.com/diagrams/templates/use-case-diagram/movie-ticket-reservation-system-/>

**PROMPT:** The following should be textually analyzed and a use case diagram created containing the necessary 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 Movie Ticket Reservation System provides a seamless experience to users who wish to browse movies and reserve tickets online. The system has a user-friendly interface that allows users to easily browse movies and select the movie of their choice. Users can browse through the list of movies, view their details, and select the movie they want to watch. This feature is particularly useful for users who are not sure which movie they want to watch and want to explore their options before making a decision. Once the user has selected a movie, they can choose their preferred showtime from the available options. The system provides a list of showtimes for each movie, and users can select the showtime that works best for them. This feature is especially useful for users who have busy schedules and need to plan their movie outings accordingly. Once the showtime is selected, users can proceed to the next step. The next step in the Movie Ticket Reservation System is selecting seats. Users can choose the seats they want to reserve, and the system will show them which seats are available and which ones are already taken. This feature allows users to pick their preferred seats and ensures that they have a comfortable and enjoyable movie experience. The final step in the Movie Ticket Reservation System is confirming payment and printing the ticket. Users can make their payment using a secure payment gateway, and once the payment is confirmed, they can print their ticket. The system generates a unique ticket code for each reservation, which ensures that each ticket is valid and can only be used once. This feature allows users to have a hassle-free movie experience and eliminates the need to stand in long queues to buy tickets. In conclusion, the Movie Ticket Reservation System provides a convenient and easy-to-use platform for users to browse movies, select showtimes, choose seats, make payments, and print tickets. The system is designed to provide a seamless experience to users and eliminates the need for manual ticketing processes. With the Movie Ticket Reservation System, users can enjoy their favorite movies without any hassle and have a comfortable and enjoyable movie experience.***

ChatGPT

**ChatGPT**

Based on the provided text, I've identified the actors, use cases, and their relationships. Below is the PlantUML code for the corresponding use case diagram:

@startuml

actor User

rectangle "Movie Ticket Reservation System" {

usecase "Browse Movies" as BrowseMovies

usecase "Select Movie" as SelectMovie

usecase "Select Showtime" as SelectShowtime

usecase "Select Seats" as SelectSeats

usecase "Confirm Payment" as ConfirmPayment

usecase "Print Ticket" as PrintTicket

}

User --> BrowseMovies

User --> SelectMovie

User --> SelectShowtime

User --> SelectSeats

User --> ConfirmPayment

User --> PrintTicket

SelectMovie --> SelectShowtime

SelectShowtime --> SelectSeats

SelectSeats --> ConfirmPayment

ConfirmPayment --> PrintTicket

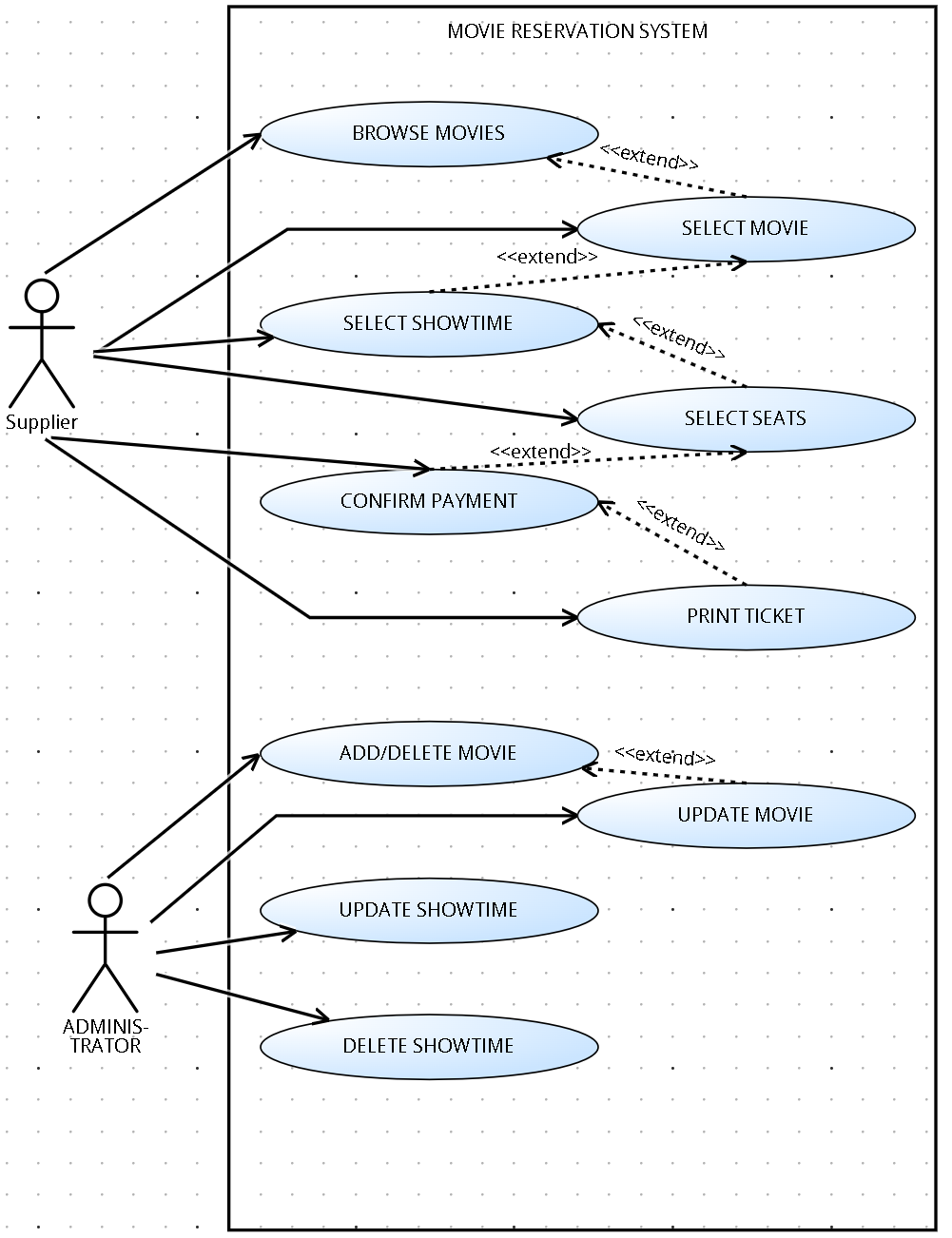
@enduml

**Analysis:**

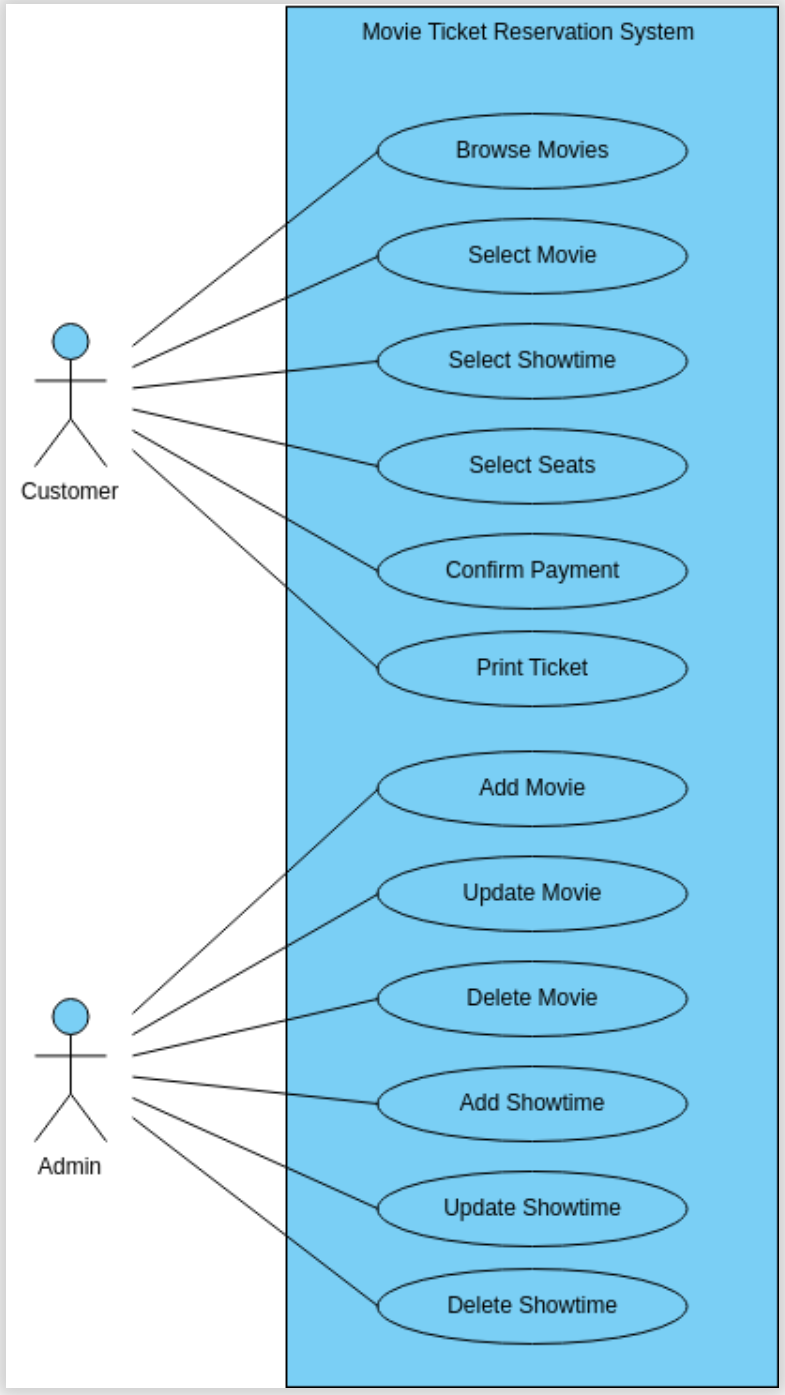
In this exercise, the description focused entirely on the customer's perspective and their interactions with the movie ticket reservation system. However, the original Use Case Diagram (UCD) solution incorporated the administrator actor and coherent use cases, even though the administrator was not mentioned in the text. It's crucial to note this discrepancy because the code generated by ChatGPT did not consider the administrator, even though it accurately identified the customer actor and all of its use cases. In fact, ChatGPT demonstrated a higher proficiency than the original solution by incorporating some "include" relationships, which, upon rigorous analysis, were perfect.

The only notable defect in ChatGPT's output was that the "include" relations were not clearly identified as such, lacking the corresponding dashed arrows and the "<<include>>" notation either on top or beside the arrows. Nevertheless, this remains a minor defect. Overall, the quality of the code and UCD generated by ChatGPT was very high, showcasing its capability to produce accurate and detailed diagrams even in scenarios with slight deviations from the provided textual descriptions.

**MY SOLUTION:**

****

**REFERENCE SOLUTION:**

****

**LLM SOLUTION:**

