***EXERCISE 13***🡪 <https://www.uml-diagrams.org/airport-checkin-uml-use-case-diagram-example.html>

**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:

***Airport Check-in and security screening: Business actors are Passenger, Tour Guide, Minor (Child), Passenger with Special Needs (e.g. with disabilities), all playing external roles in relation to airport business. Business use cases are Individual Check-In, Group Check-In (for groups of tourists), Security Screening, etc. - representing business functions or processes taking place in airports and serving the needs of passengers. Business use cases Baggage Check-in and Baggage Handling extend Check-In use cases, because passengers might have no luggage, so baggage check-in and handling are optional.***

ChatGPT

**ChatGPT**

Certainly! Here's the PlantUML code for the use case diagram based on the provided text:

@startuml

left to right direction

actor "Passenger" as Passenger

actor "Tour Guide" as TourGuide

actor "Minor (Child)" as Minor

actor "Passenger with Special Needs" as SpecialNeedsPassenger

rectangle "Airport Business" {

usecase "Individual Check-In" as IndividualCheckIn

usecase "Group Check-In" as GroupCheckIn

usecase "Security Screening" as SecurityScreening

usecase "Check-In" as CheckIn {

usecase "Baggage Check-In" as BaggageCheckIn

usecase "Baggage Handling" as BaggageHandling

}

}

Passenger --> IndividualCheckIn

Passenger --> GroupCheckIn

Passenger --> SecurityScreening

GroupCheckIn --> CheckIn : <<extend>>

IndividualCheckIn --> CheckIn : <<extend>>

CheckIn --> BaggageCheckIn : <<extend>>

CheckIn --> BaggageHandling : <<extend>>

@enduml

**PROMPT:** Try providing a different code because this one has syntax errors and I am not able to generate the UCD

(AFTER THIS PROMPT, I NEEDED TO MAKE 3 MORE PROMPTS UNTILL A CODE WORKED)

@startuml

left to right direction

actor "Passenger" as Passenger

actor "Tour Guide" as TourGuide

actor "Minor (Child)" as Minor

actor "Passenger with Special Needs" as SpecialNeedsPassenger

rectangle "Airport Business" {

usecase "Individual Check-In" as IndividualCheckIn

usecase "Group Check-In" as GroupCheckIn

usecase "Security Screening" as SecurityScreening

usecase "Baggage Check-In" as BaggageCheckIn

usecase "Baggage Handling" as BaggageHandling

}

Passenger --> IndividualCheckIn

Passenger --> GroupCheckIn

Passenger --> SecurityScreening

GroupCheckIn --> BaggageCheckIn

BaggageCheckIn --> BaggageHandling

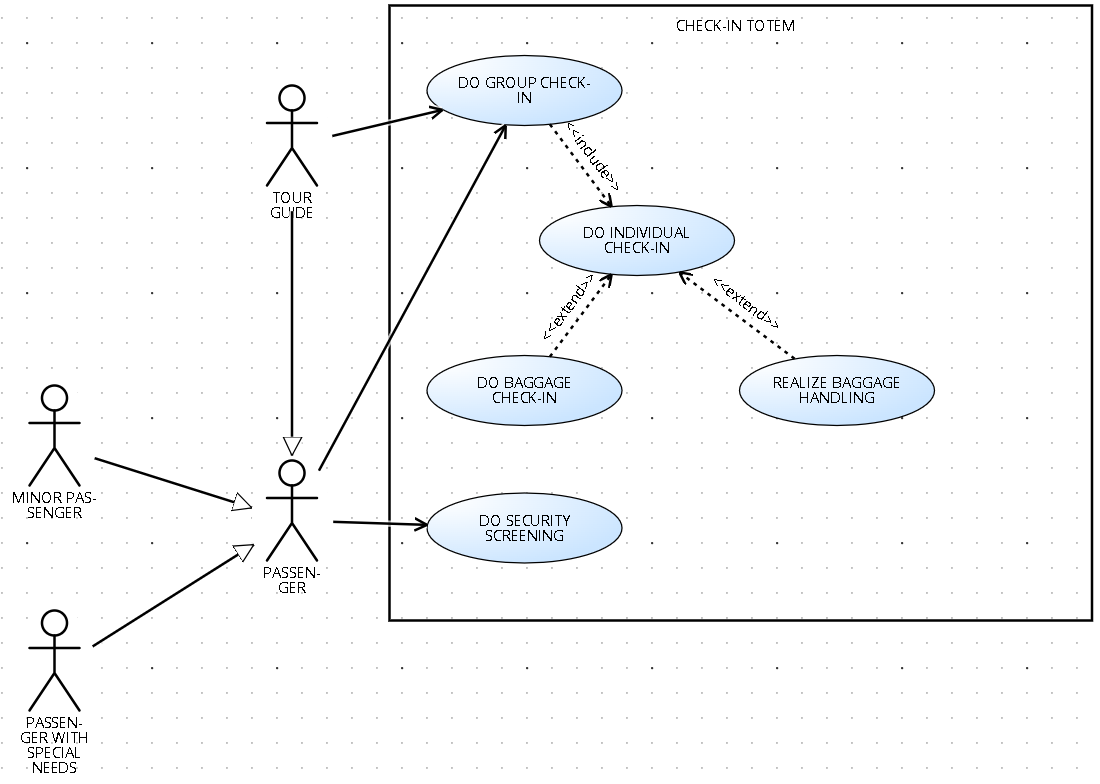
@enduml

**Analysis:**

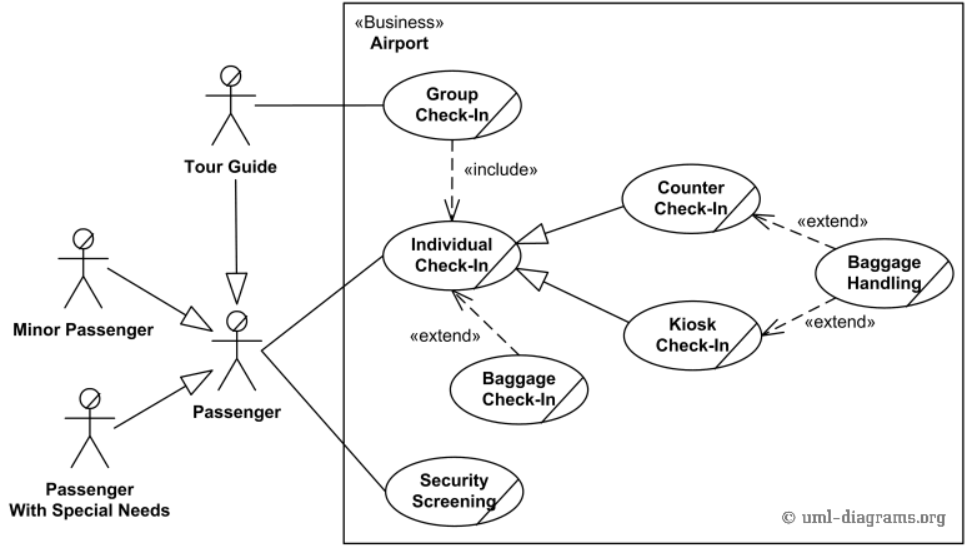
In this problem, information about the actors and use cases in an Airport Check-in and security screening context was provided without detailing the sequence of actions and interactions for each actor, as in other exercises. ChatGPT was expected to comprehend and generate a reasonable solution, but the first code it produced had syntax errors, preventing the generation of the UCD. It required three prompts before a correct code suitable for the PlantUML website was generated. However, the UCD lacked quality in terms of critical indicators such as correctness, completeness, and readability.

The diagram proved unhelpful due to outstanding actors with no associations to any use case, missing generalizations between actors and use cases, and the omission of obvious extend/include relationships that one would expect the AI to understand. Upon requesting ChatGPT to address these issues and generate a new code, a subsequent code was provided, once again with syntax errors. It's evident that when generalizations/specializations and extend/include relationships are involved in the UCD, ChatGPT lacks the capacity to fully understand and provide correct code, resulting in a frustrating experience for the user.

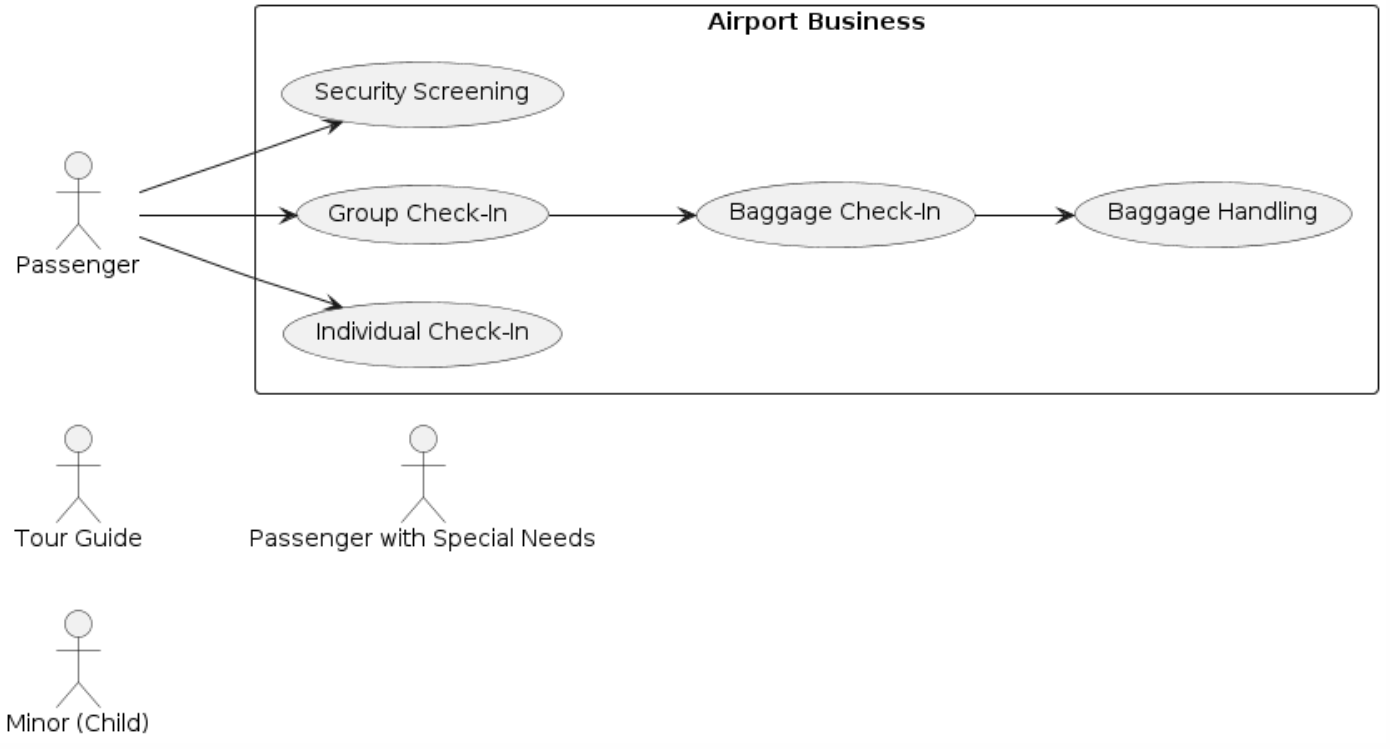
**MY SOLUTION:**

****

**REFERENCE SOLUTION:**

****

**LLM SOLUTION:**

****