***EXERCISE 15***🡪 <https://www.uml-diagrams.org/examples/hospital-management-use-case-diagram-example.html?context=uc-examples>

**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:

***Hospital Management System is a large system including several subsystems or modules providing a variety of functions. Purpose: Describe major services (functionality) provided by a hospital's reception. The Hospital Reception subsystem or module supports some of the many job duties of hospital receptionists. Receptionist schedules patient's appointments and admission to the hospital, collects information from patient upon patient's arrival and/or by phone. For the patient that will stay in the hospital ("inpatient") she or he should have a bed allotted in a ward. Receptionists might also receive patient's payments, record them in a database and provide receipts, file insurance claims and medical reports.***

ChatGPT

**ChatGPT**

Here is the PlantUML code for the Hospital Management System use case diagram:

@startuml

actor Patient

actor Receptionist

actor InsuranceCompany

actor Database

rectangle "Hospital Reception" {

usecase "Schedule Appointment" as UC1

usecase "Admit Patient" as UC2

usecase "Collect Patient Information" as UC3

usecase "Allot Bed" as UC4

usecase "Receive Payments" as UC5

usecase "Record Payments" as UC6

usecase "Provide Receipts" as UC7

usecase "File Insurance Claims" as UC8

usecase "Record Medical Reports" as UC9

Patient --> UC1 : Requests

Patient --> UC3 : Provides

Receptionist --> UC1

Receptionist --> UC2

Receptionist --> UC3

Receptionist --> UC4

Receptionist --> UC5

Receptionist --> UC6

Receptionist --> UC7

Receptionist --> UC8

Receptionist --> UC9

UC2 --> UC4 : <<include>>

UC5 --> UC6 : <<include>>

UC5 --> UC7 : <<include>>

UC5 --> UC8 : <<include>>

}

Database --> UC3 : Record Patient Information

Database --> UC4 : Update Bed Allotment

Database --> UC6 : Record Payment Details

Database --> UC8 : Record Insurance Claims

Database --> UC9 : Save Medical Reports

InsuranceCompany --> UC8 : Submit Claims

@enduml

**PROMPT:** The UCD is too messy and difficult to read and understand, for simplicity, consider just the actor "Receptionist" which is the only primary actor who interacts with the system.

ChatGPT

**ChatGPT**

Certainly, for simplicity, we can focus on the interactions involving the "Receptionist" as the primary actor. Here is a simplified PlantUML code:

@startuml

actor Receptionist

rectangle "Hospital Reception" {

usecase "Schedule Appointment" as UC1

usecase "Admit Patient" as UC2

usecase "Collect Patient Information" as UC3

usecase "Allot Bed" as UC4

usecase "Receive Payments" as UC5

usecase "Record Payments" as UC6

usecase "Provide Receipts" as UC7

usecase "File Insurance Claims" as UC8

usecase "Record Medical Reports" as UC9

Receptionist --> UC1

Receptionist --> UC2

Receptionist --> UC3

Receptionist --> UC4

Receptionist --> UC5

Receptionist --> UC6

Receptionist --> UC7

Receptionist --> UC8

Receptionist --> UC9

UC2 --> UC4 : <<include>>

UC5 --> UC6 : <<include>>

UC5 --> UC7 : <<include>>

UC5 --> UC8 : <<include>>

}

@enduml

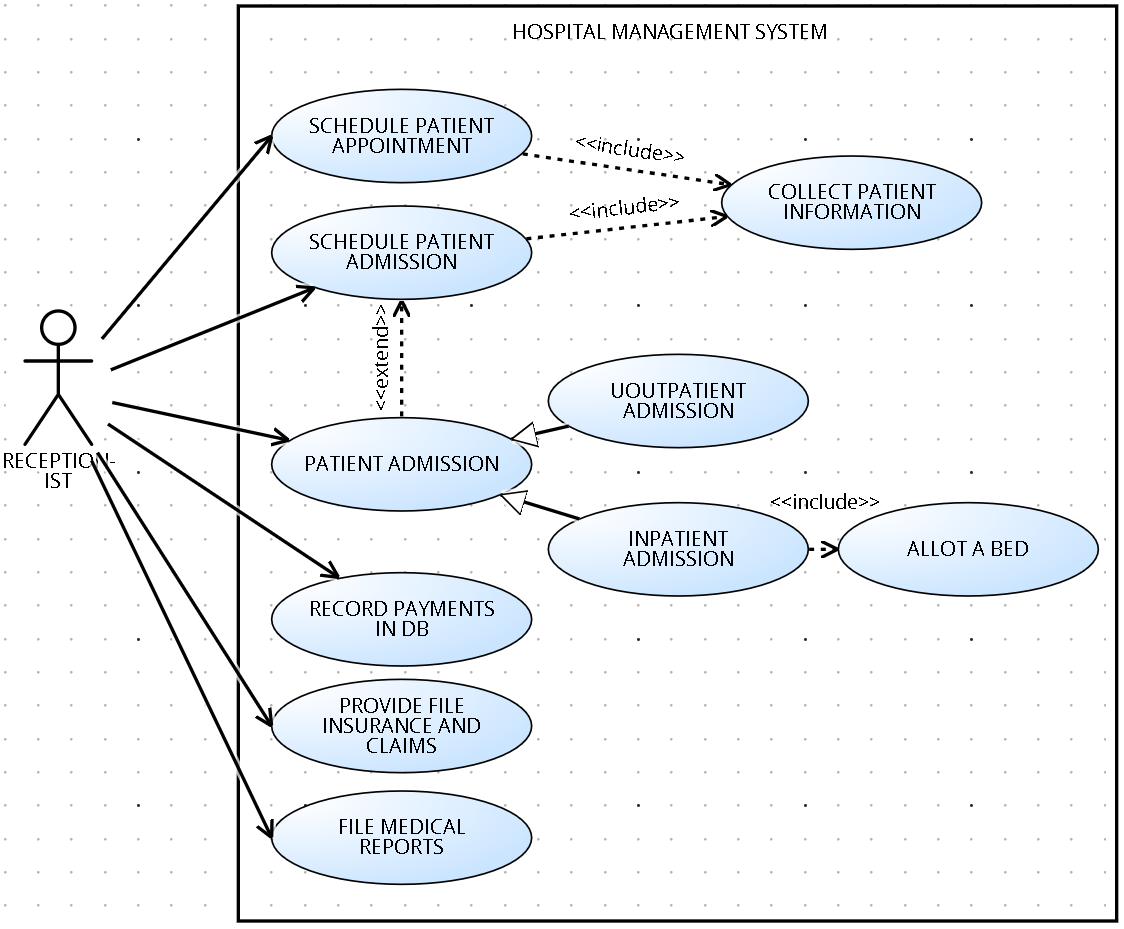
**Analysis:**

In this exercise, unlike others we've analyzed, the relatively short text description lacked some explicit information, requiring readers to infer details for inclusion in the Use Case Diagram (UCD) solution. The first code generated by ChatGPT produced an unreadable diagram with excessive associations and unnecessary actors, introducing confusion. The intended focus was on analyzing the hospital reception system with the sole primary actor being the receptionist. Failure to grasp this concept could potentially lead to an inaccurate diagram.

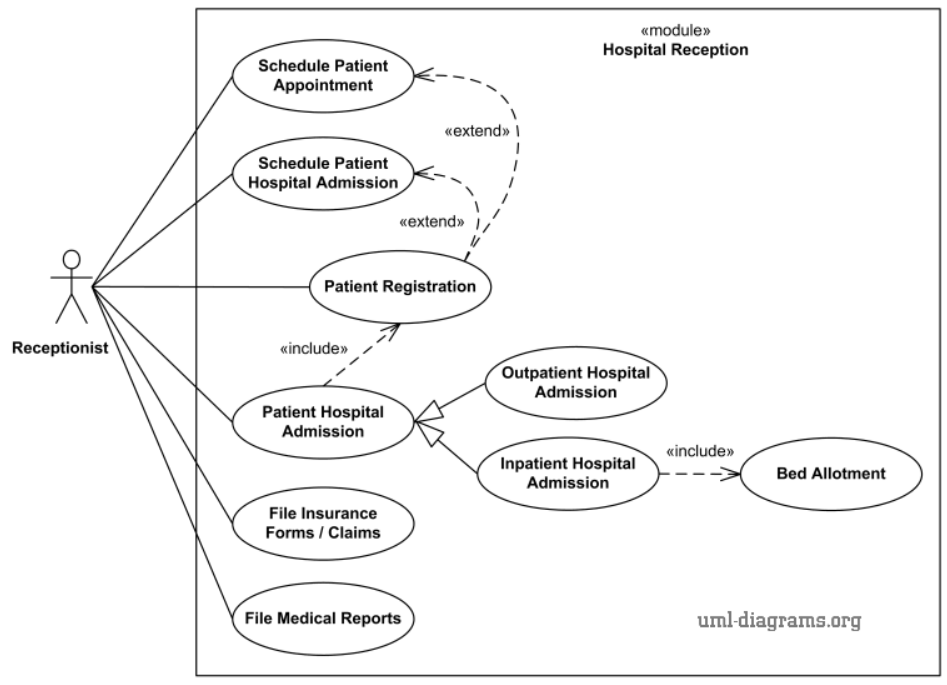
Upon highlighting this issue in the second prompt, a new code was generated that resulted in a more refined UCD, comparable to the original solution, albeit with some pros and cons. On the positive side, it correctly incorporated the use cases "record payment," "provide receipt," and "file insurance claims" with "include" relationships regarding the "receive payments" use case. However, it did not consider "schedule appointment" and "admit patient" as extended use cases of "collecting patient information." Additionally, the patients were not differentiated into outpatients and inpatients. Nevertheless, the "bed allotment" use case was correctly incorporated with the corresponding "include" relation.

In summary, the solution provided by ChatGPT was satisfactory for a common user and could be further improved by a more UCD-specialized user.

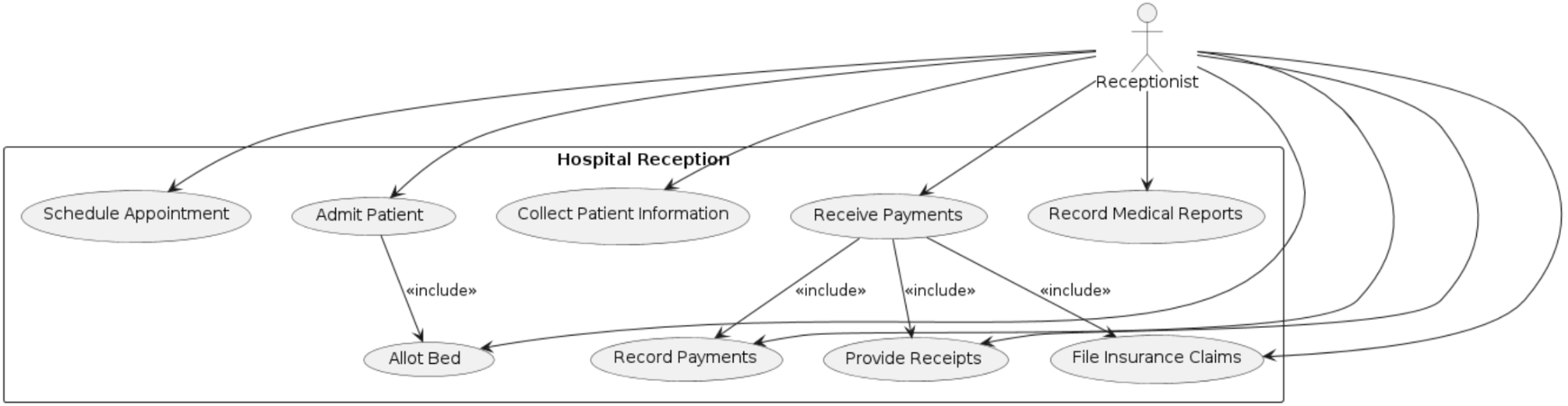
**MY SOLUTION:**

****

**REFERENCE SOLUTION:**

****

**LLM SOLUTION:**

****