***EXERCISE 1***🡪 <https://irm.com.au/use-use-cases/>

**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 give me the PlantUML code for the use case diagram corresponding to the following text:

***At the start of each semester a student can request a prospectus containing a course list. Information about a course is provided, such as the tutor, department and pre-requisites. The new system will allow students to create a schedule, then select four courses. Each student chooses two others in case their first choices become full or are cancelled. No course can have more than 10 students. No course can have less than 3 students or it will be cancelled. This will be the same functionality as available to other internal users of the system. When registration is complete, the registration system sends a message to the billing system to send out a bill to the student. Tutors use the system to find which classes they are teaching and who the students are. The registrar will administer the system. For a period at the beginning of the semester the student can change their schedule. Students must be allowed to access the system during this time to add or delete courses.***

ChatGPT

**ChatGPT**

Sure, let's analyze the text and identify the actors, use cases, and associations.

**Actors:**

1. Student
2. Tutors
3. Registrar
4. Billing System

**Use Cases:**

1. Request Prospectus
2. View Course Information
3. Create Schedule
4. Select Courses
5. Modify Schedule
6. Send Registration to Billing System
7. View Teaching Schedule (for Tutors)
8. Administer System (for Registrar)

**Associations:**

1. Student requests a prospectus.
2. Student views course information.
3. Student creates a schedule.
4. Student selects courses.
5. Student modifies schedule during the specified period.
6. Registration system sends registration to the billing system.
7. Billing system sends out a bill to the student.
8. Tutors view their teaching schedule and associated students.
9. Registrar administers the system.

Now, let's create the PlantUML code for the use case diagram:

plantumlCopy code

@startuml

left to right direction

actor Student

actor Tutors

actor Registrar

actor BillingSystem

rectangle "System" {

usecase "Request Prospectus" as UC1

usecase "View Course Information" as UC2

usecase "Create Schedule" as UC3

usecase "Select Courses" as UC4

usecase "Modify Schedule" as UC5

usecase "Send Registration to Billing System" as UC6

}

rectangle "Internal Users" {

usecase "View Teaching Schedule" as UC7

usecase "Administer System" as UC8

}

Student --> UC1

Student --> UC2

Student --> UC3

Student --> UC4

Student --> UC5

Student --> UC6

UC3 --> UC4

UC3 --> UC5

UC6 --> BillingSystem

Tutors --> UC7

Registrar --> UC8

@enduml

This PlantUML code represents the use case diagram with actors, use cases, and associations based on the provided text. Adjustments can be made according to your specific needs or preferences.

User

**You**

Shouldnt the registrar maintain course information, maintain tutors information, maintain curriculum and maintain student information for example? I mean, shouldnt these thing be use cases of the actor "registrar"?

ChatGPT

**ChatGPT**

Certainly! If the responsibilities of the Registrar involve maintaining various information, we can indeed represent those as separate use cases for the Registrar actor. Here's an updated PlantUML code reflecting these additional use cases:

plantumlCopy code

@startuml

left to right direction

actor Student

actor Tutors

actor Registrar

actor BillingSystem

rectangle "System" {

usecase "Request Prospectus" as UC1

usecase "View Course Information" as UC2

usecase "Create Schedule" as UC3

usecase "Select Courses" as UC4

usecase "Modify Schedule" as UC5

usecase "Send Registration to Billing System" as UC6

}

rectangle "Internal Users" {

usecase "View Teaching Schedule" as UC7

usecase "Administer System" as UC8

usecase "Maintain Course Information" as UC9

usecase "Maintain Tutors Information" as UC10

usecase "Maintain Curriculum" as UC11

usecase "Maintain Student Information" as UC12

}

Student --> UC1

Student --> UC2

Student --> UC3

Student --> UC4

Student --> UC5

Student --> UC6

UC3 --> UC4

UC3 --> UC5

UC6 --> BillingSystem

Tutors --> UC7

Registrar --> UC8

Registrar --> UC9

Registrar --> UC10

Registrar --> UC11

Registrar --> UC12

@enduml

Now, the Registrar actor has additional use cases related to maintaining course information, tutors' information, curriculum, and student information. This reflects the expanded responsibilities of the Registrar in the system.

**Analysis:**

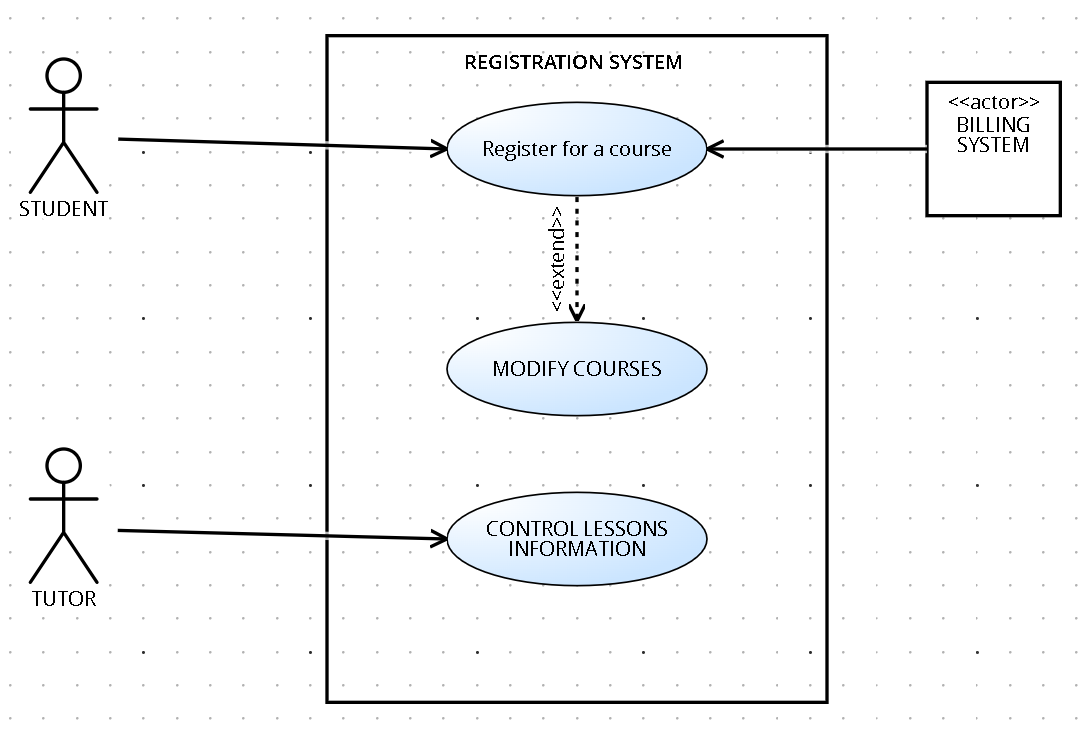
From this exercise, we observe that the PlantUML code provided by ChatGPT generated a use case diagram that differed from the one in the original answer, but in terms of correctness, it is a good diagram. In completeness, both answers are substantially complete, but the original answer contained only one use case for the actor "student," seen as a summary use case of all the actions the text describes that the student can do. In contrast, ChatGPT's output gives a more detailed use case diagram for the actions associated with the student.

Analyzing the text, we realize that the use cases associated with the student, generated by the PlantUML code, align perfectly with the explicit description given. The original answer included some use cases related to the actor "registrar," not explicitly stated in the text but inferred from a commonsense perspective, essentially thought by human intelligence. For instance, the use case "maintain tutor’s information" is not explicitly stated, but it is reasonable to think that the registrar must do this for the student to access information and register for courses.

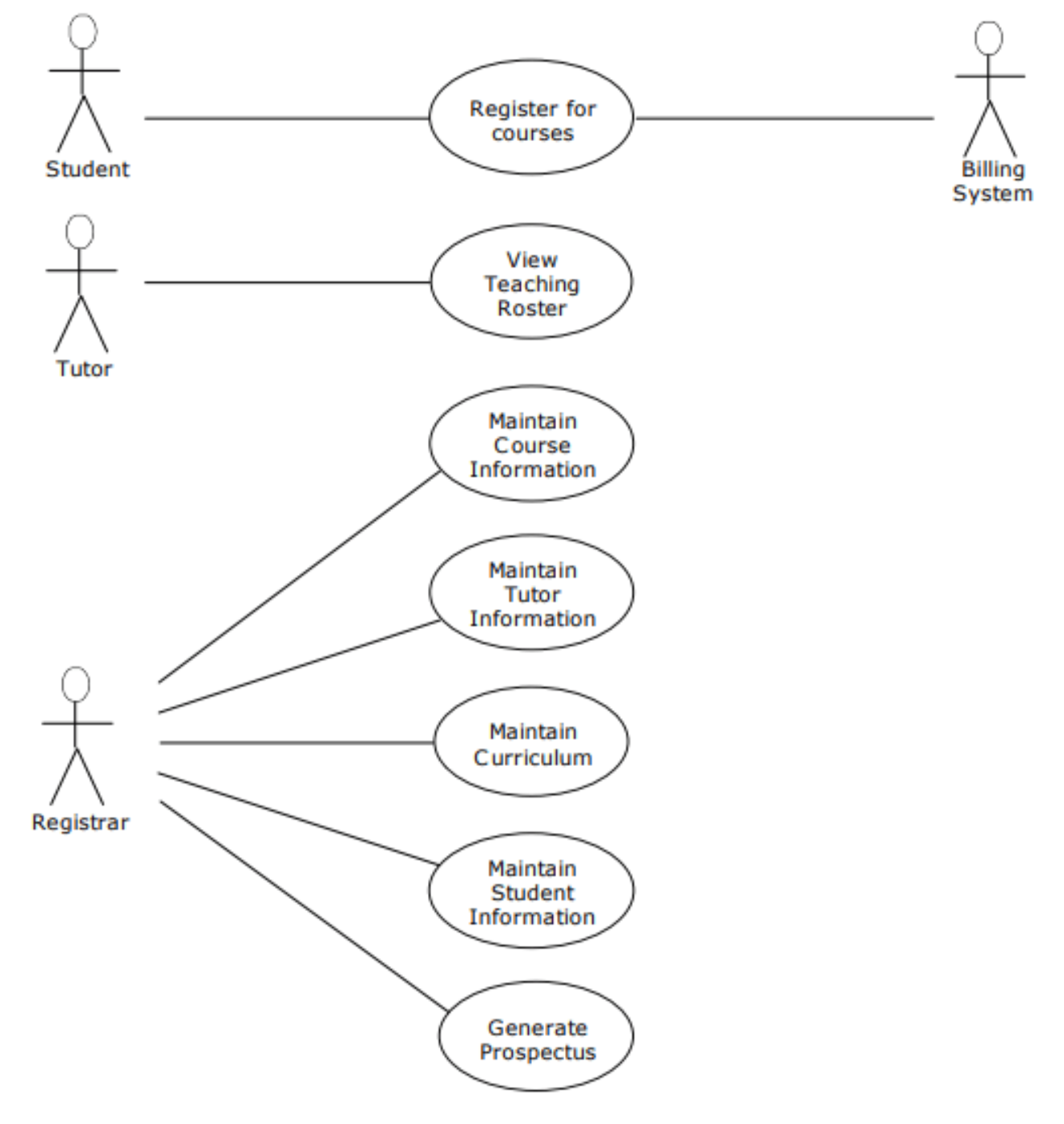
Moreover, the code generated by ChatGPT provides two different systems: one called "internal users," where tutors and registrars can interact with the system, and the other called "system," where students can interact, along with the secondary actor, the billing system. Although this separation of use case diagrams is not necessary, it adds a higher level of detail, which is a positive aspect of ChatGPT's output.

After evaluating the output, I decided to prompt something new to ChatGPT, as shown in the prompt sequence. In this second case, it generated a very high-level diagram, encompassing complete use cases related to both the student and the registrar. It can be said that this final use case diagram is the best one provided in terms of quality indicators.

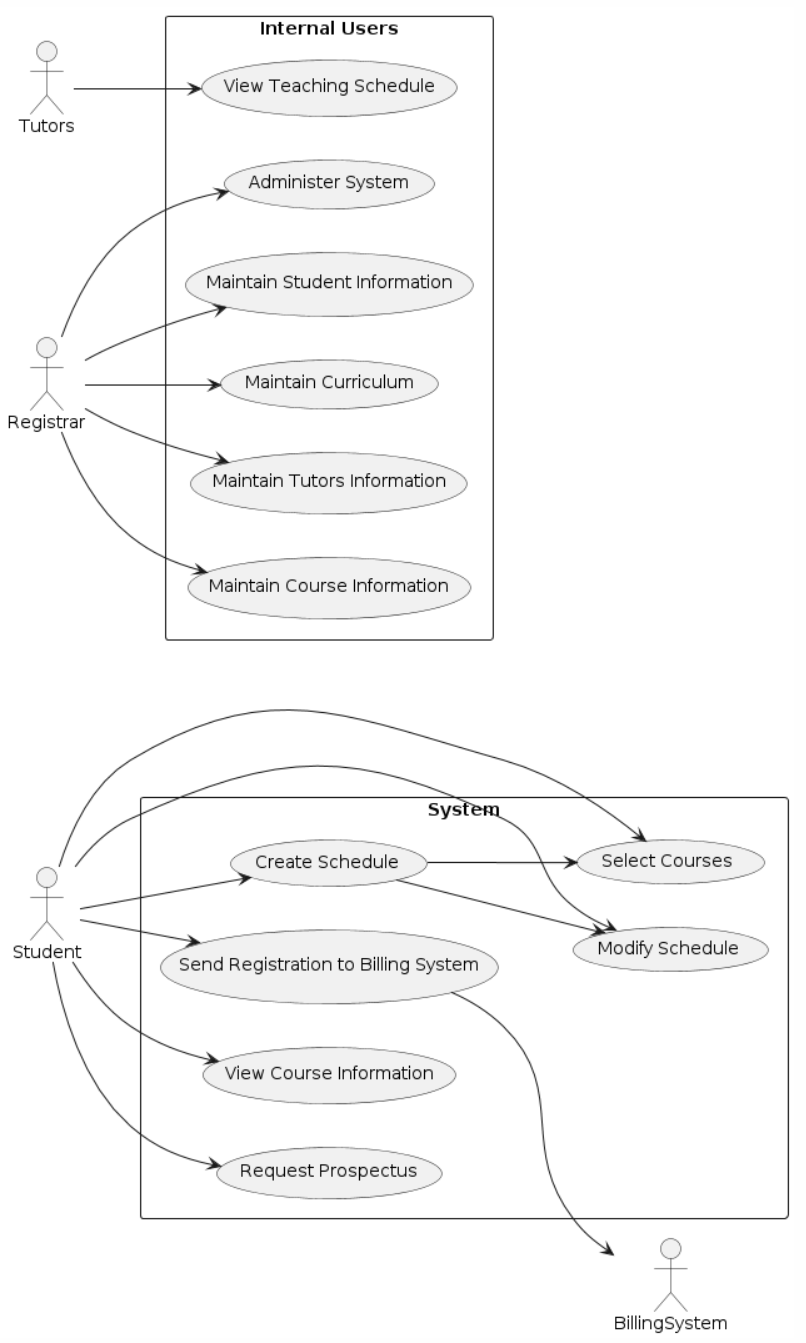
**MY SOLUTION:**

****

**REFERENCE SOLUTION:**

****

**LLM SOLUTION:**

****