"Actually, we had something completely different in mind." Not a sentence that you want to hear from a customer. But it happens. Because reality always shows us how different people communicate differently, or misunderstand things – Paul Watzlawick, the Austrian-American communication scientist, sends his greetings. When requirements in your software are misunderstood, imprecisely communicated or just not mentioned by the customer, that doesn't just lead to uncomfortable conflict, it also costs you time and money. That is why use cases and use case diagrams have proven themselves in project management: a graphic visualization of the behavior of a system from the view of the user, described through defined visual means of UML (Unified Modeling Language). They offer great support when finding and defining requirements.

Use case diagram: a short definition

A use case diagram is a behavior diagram and visualizes the observable interactions between actors and the system under development. The diagram consists of the system, the related use cases and actors and relates these to each other:

System: What is being described?

Actor: Who is using the system?

Use Case: What are the actors doing?

What isn't a use case diagram?

A use case diagram doesn't describe the order in which the use cases are carried out. For example, let's define the use case *Withdraw money*. This process is carried out in many steps, like *Insert card, Enter PIN, Select amount, take out amount* and *take card out*. Of course, these are activities are all from the customer's point of view – but this sequence should not be covered by the use case diagram. Other diagrams are recommended for that, like the activity diagram. A use case diagram should describe the desired functionality of the system and relates it to use cases and actors. That way it can represent existing viewpoints of the system and how they are interpreted differently – only through this can requirements be completely understood.

UML - Use Case Diagrams

Previous Page Next Page

To model a system, the most important aspect is to capture the dynamic behavior. Dynamic behavior means the behavior of the system when it is running/operating.

Only static behavior is not sufficient to model a system rather dynamic behavior is more important than static behavior. In UML, there are five diagrams available to model the dynamic nature and use case diagram is one of them. Now as we have to discuss that the use case diagram is dynamic in nature, there should be some internal or external factors for making the interaction.

These internal and external agents are known as actors. Use case diagrams consists of actors, use cases and their relationships. The diagram is used to model the system/subsystem of an application. A single use case diagram captures a particular functionality of a system.

Hence to model the entire system, a number of use case diagrams are used.

Purpose of Use Case Diagrams

The purpose of use case diagram is to capture the dynamic aspect of a system. However, this definition is too generic to describe the purpose, as other four diagrams (activity, sequence, collaboration, and Statechart) also have the same purpose. We will look into some specific purpose, which will distinguish it from other four diagrams.

- HR Interview Questions
- Computer Glossary
- Who is Who

- Give a suitable name for actors.
- Show relationships and dependencies clearly in the diagram.
- Do not try to include all types of relationships, as the main purpose of the diagram is to identify the requirements.
- Use notes whenever required to clarify some important points.

Following is a sample use case diagram representing the order management system. Hence, if we look into the diagram then we will find three use cases **(Order, SpecialOrder, and NormalOrder)** and one actor which is the customer.

The SpecialOrder and NormalOrder use cases are extended from *Order* use case. Hence, they have extended relationship. Another important point is to identify the system boundary, which is shown in the picture. The actor Customer lies outside the system as it is an external user of the system.

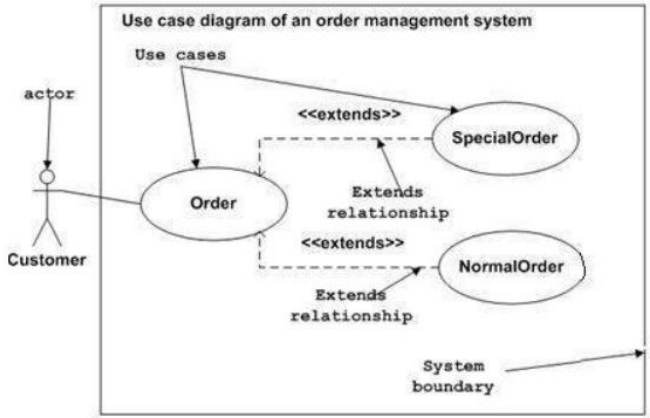
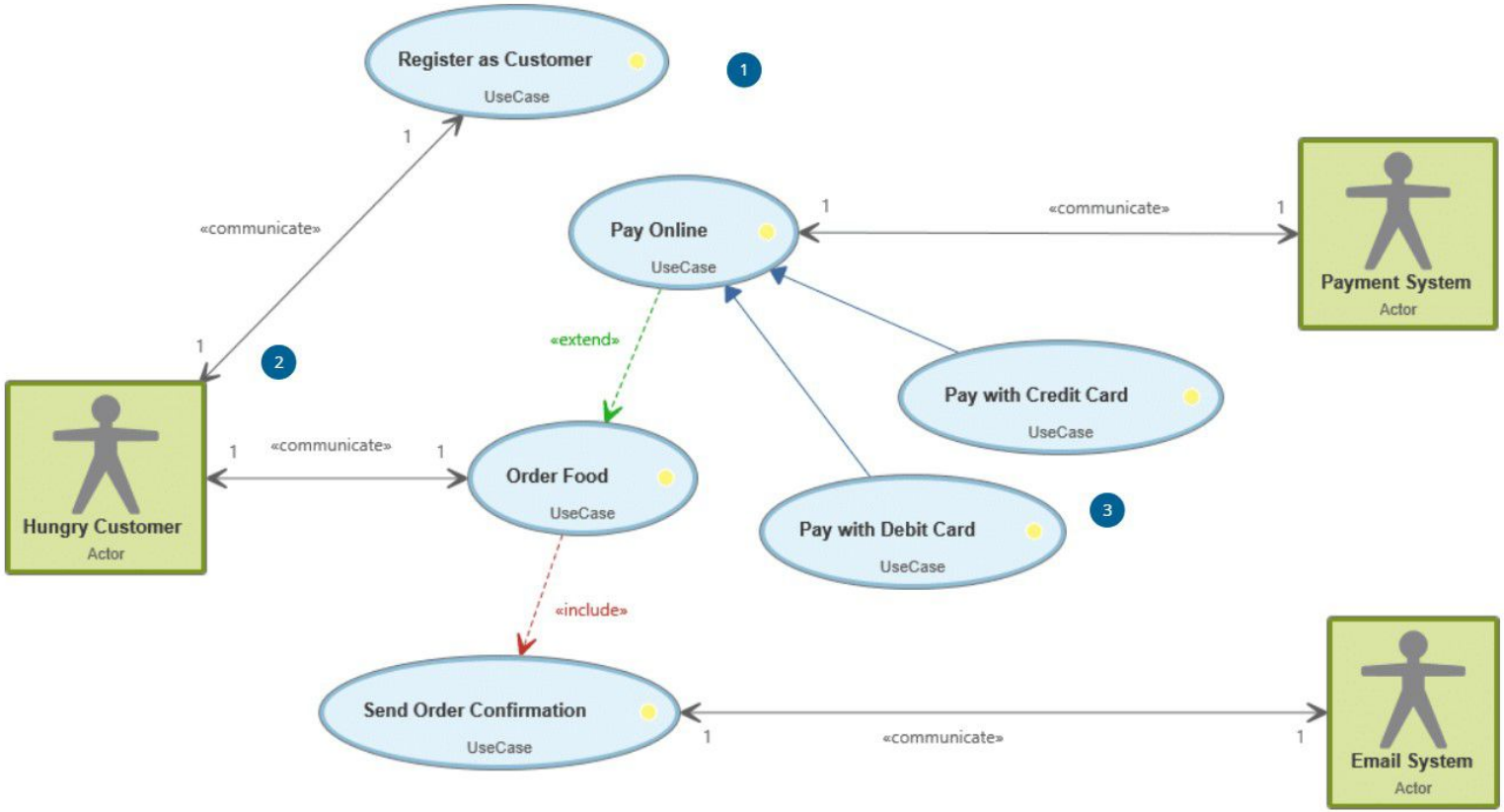


Figure: Sample Use Case diagram


Register as Customer
UseCase
1
Pay Online
UseCase
«communicate»
Payment System
Actor
«extend»
Pay with Credit Card
UseCase
Hungry Customer
Actor
2
Order Food
UseCase
Pay with Debit Card
UseCase
3
«include»
Send Order Confirmation
UseCase
Email System
Actor