Introduction to Software Engineering Use-Case Model

Lecture7

What is a UML Use Case Diagram

- A UML (Unified Modeling Language) use case diagram is a visual representation of the interactions between actors (users or external systems) and a system under consideration. It depicts the functionality or behavior of a system from the user's perspective. Use case diagrams capture the functional requirements of a system and help to identify how different actors interact with the system to achieve specific goals or tasks.

- Use case diagrams provide a high-level overview of the system's functionality, showing the different features or capabilities it offers and how users or external systems interact with it. They serve as a communication tool between stakeholders, helping to clarify and validate requirements, identify system boundaries, and support the development and testing processes.

Importance of Use Case Diagrams

- **To identify functions and how roles interact with them** – The primary purpose of use case diagrams.

- **For a high-level view of the system** – Especially useful when presenting to managers or stakeholders. You can highlight the roles that interact with the system and the functionality provided by the system without going deep into inner workings of the system.

- **To identify internal and external factors** – This might sound simple but in large complex projects a system can be identified as an external role in another use case.

USE-CASE MODELS .. ELEMENTS & SYMBOLS

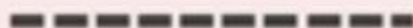|	
The actors, represented by **stick figures**;	
The use cases, represented by **ovals**, and are used to represent tasks.;	
The relationships between actors and use cases, represented by **lines**.	
Note symbol: is used to clarify an aspect or a task, it can be used with any UML diagram.	
A **dashed line** is used to attach the note to the model element to which it refers.	

USE-CASE DIAGRAMS .. EXAMPLE 1

A simple use case diagram for the main tasks in a hotel system: make reservation, check-in, and check-out, where the reservation can be done by the receptionist or a guest (by Phone/Web).

Guest

make reservation

check in guest

Receptionist

check out guest
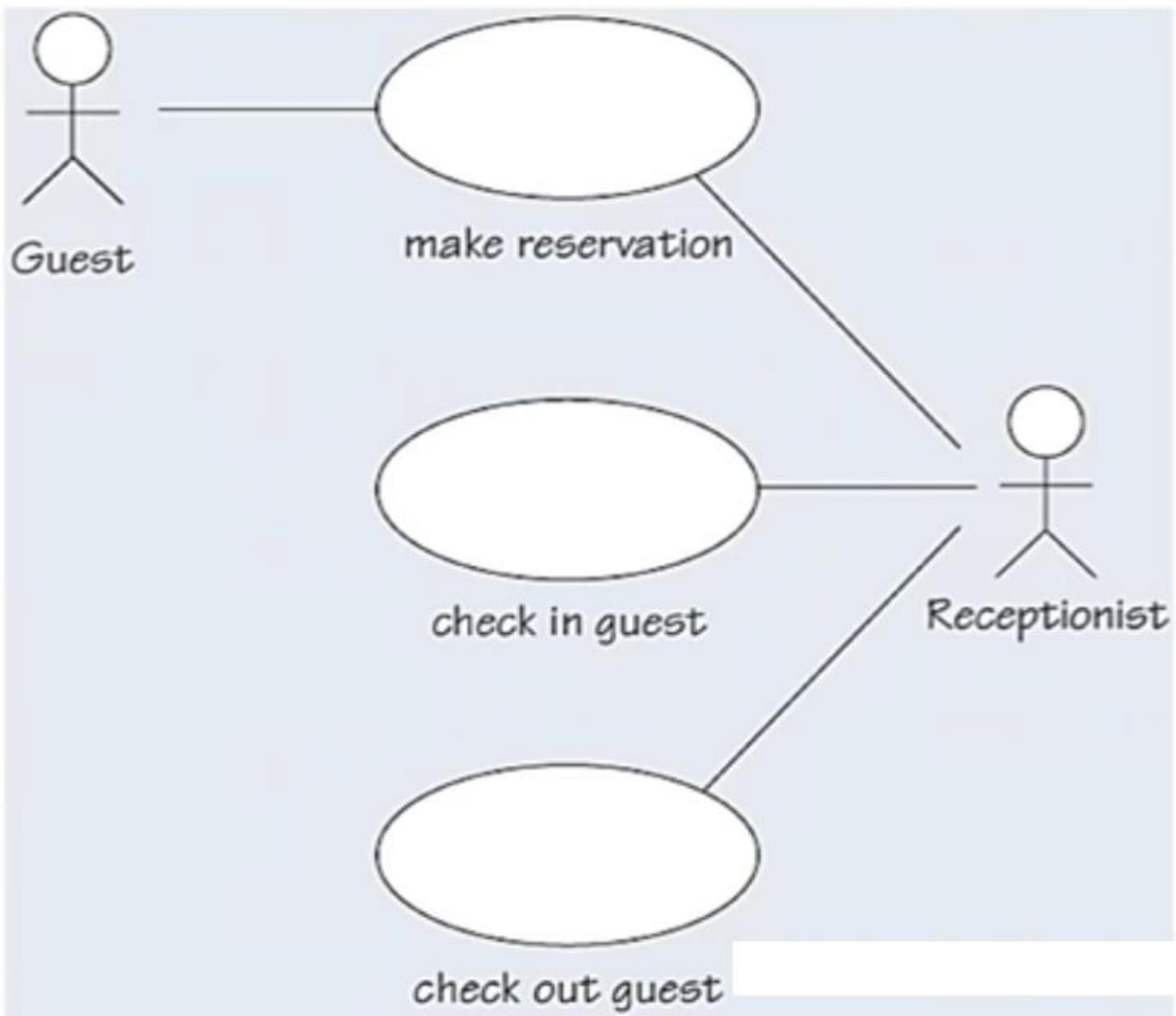
USE-CASE DIAGRAMS .. SYSTEM BOUNDARY
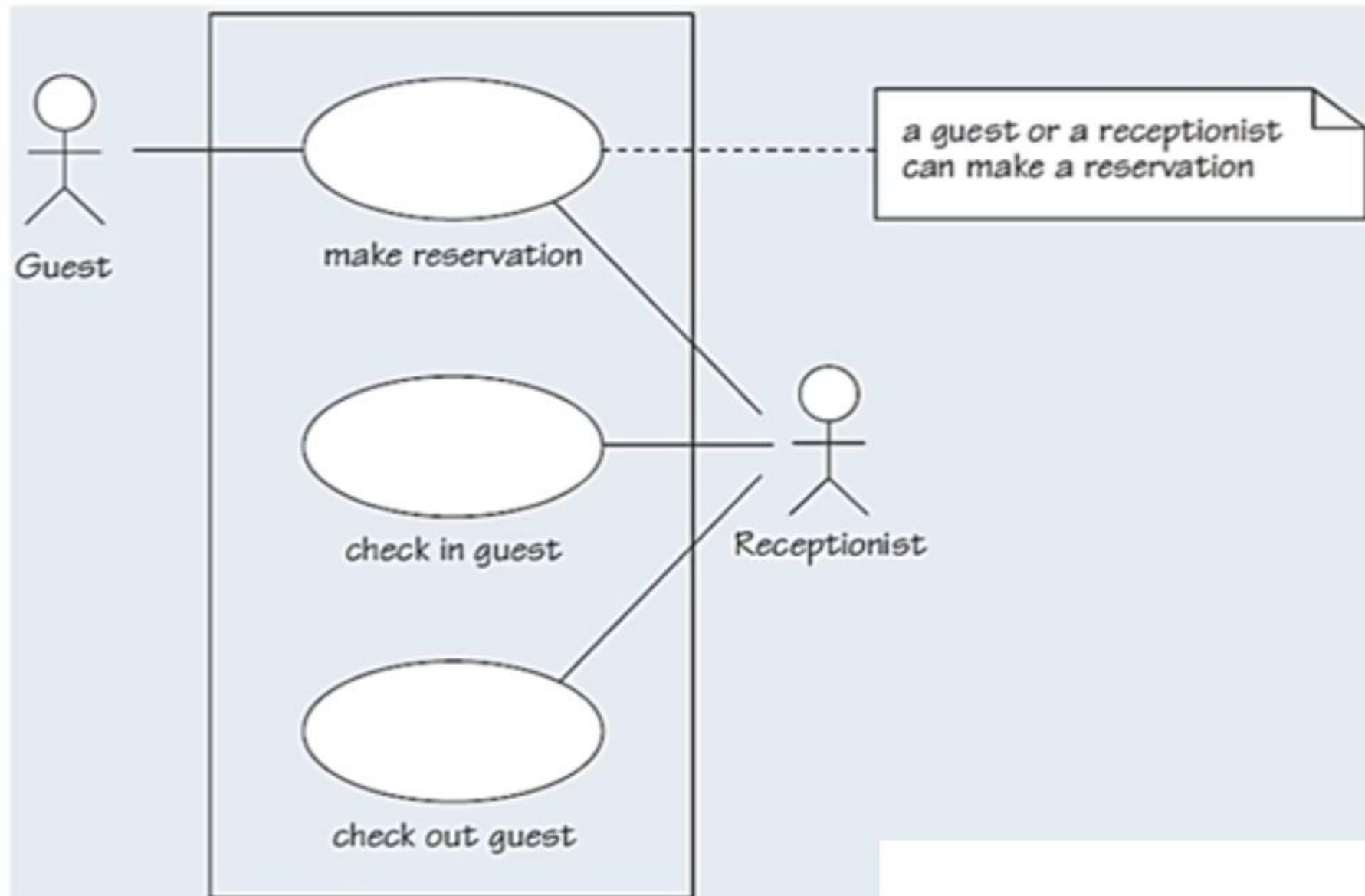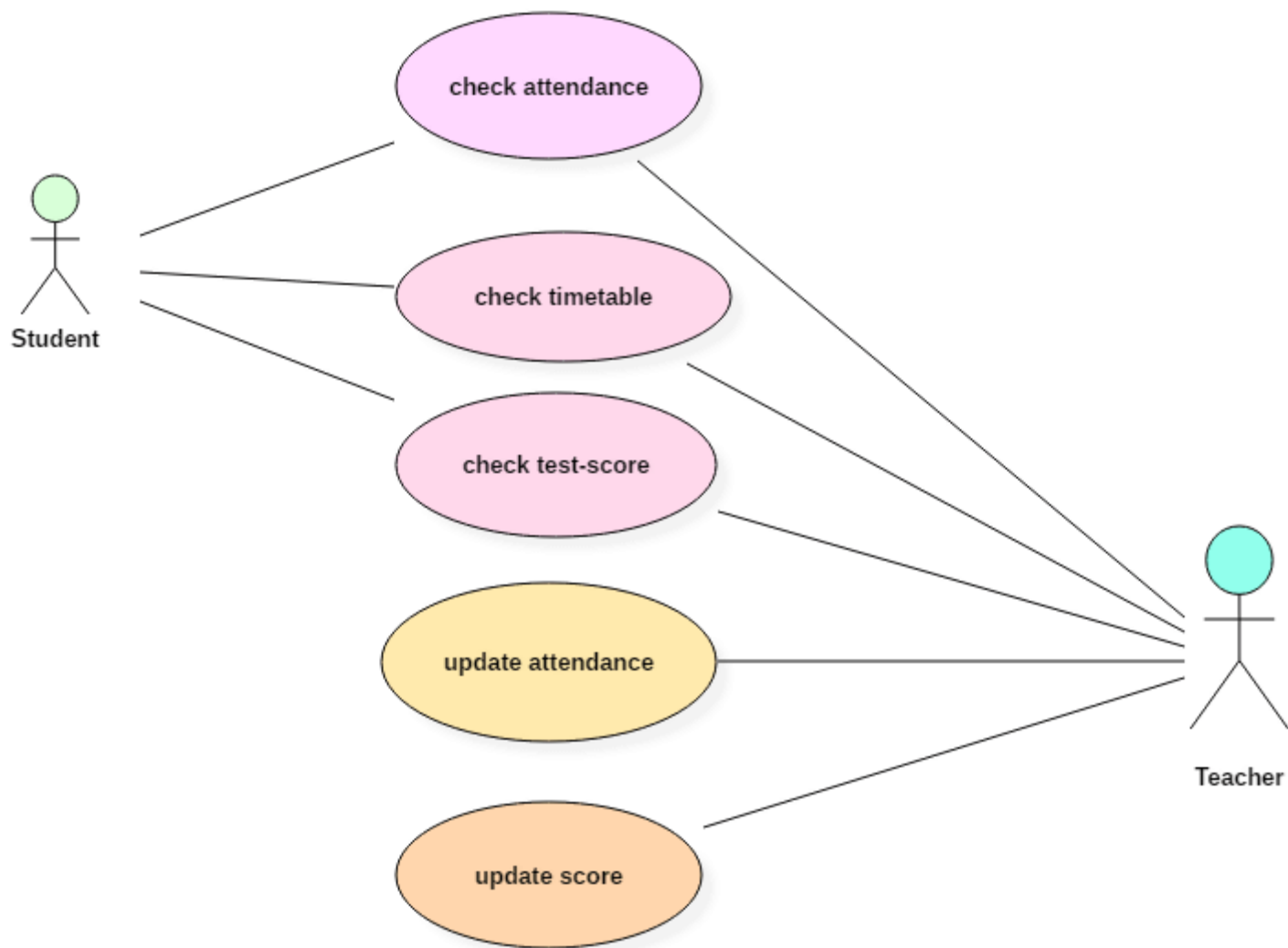
USE-CASE DESCRIPTIONS

A **scenario** describes a sequence of interactions between the system and some actors.

o In Each use case there is a set of possible scenarios. Where the main scenario is the successful scenario where nothing goes wrong and the use case is achieved.

For example: there are two scenarios for making reservation in a hotel:

Main Success Scenario:

o The guest wants to reserve a double room at the Hotel for 14 July. A double room is available for that date, and the reservation is done.

Unsuccessful Scenario:

o The guest wants to reserve a single room at the Hotel for the first week of August. There is no single room that is free for seven days in August, but there is one room available for four days and another one for the following three days. The system presents that option to the guest, who rejects it.

USE-CASE DESCRIPTIONS

What pre and post condition(s) you can obtain from the below description of a hotel check-in process?

Upon arrival, each guest provides the reservation number for his or her reservation to the hotel's receptionist, who enters it into the software system. The software system reveals the details of that reservation so that each guest can confirm them. The software system allocates an appropriate room to that guest and opens a bill for the duration of the stay. The receptionist issues a key for the room.

Precondition(s): There must be a reservation for the guest, and there must be at least one room available (of the desired type), and the guest must be able to pay for the room.

Post-condition(s): The guest will have been allocated to a room for the period identified in the reservation, the room will have been identified as being in use for a specific period, a bill will have been opened for the duration of the stay, and a key will have been issued.

USE-CASE DESCRIPTIONS

Table 1 A textual description of a use case in the hotel domain

Identifier and name	UC1 *make reservation*
Initiator	*Guest* or *Receptionist*
Goal	A room in a hotel is reserved for a guest.
Precondition	None (i.e. there are no conditions to be satisfied prior to carrying out this use case).
Postcondition	A room of the desired type will have been reserved for the guest for the requested period, and the room will no longer be free for that period.
Assumptions	The expected initiator is a guest (using a web browser to perform the use case) or a receptionist. The guest is not already known to the hotel's software system (see step 5).

Main success scenario

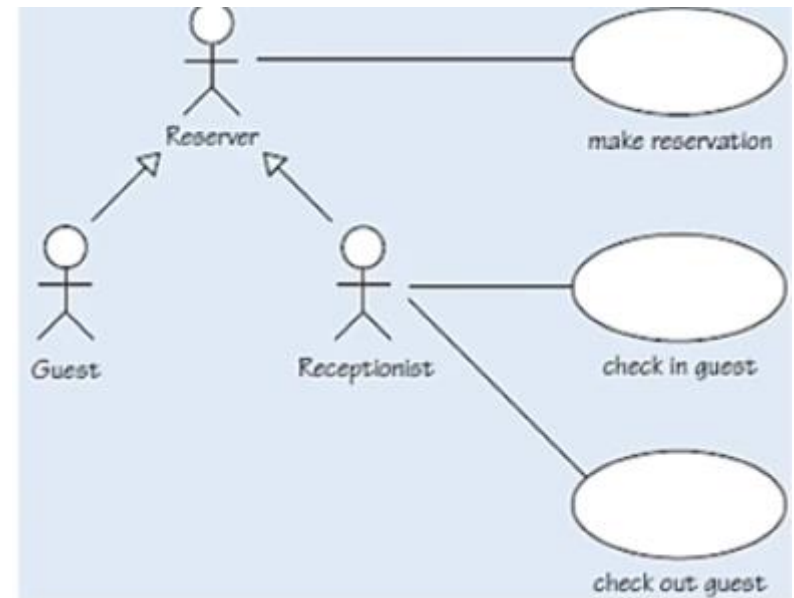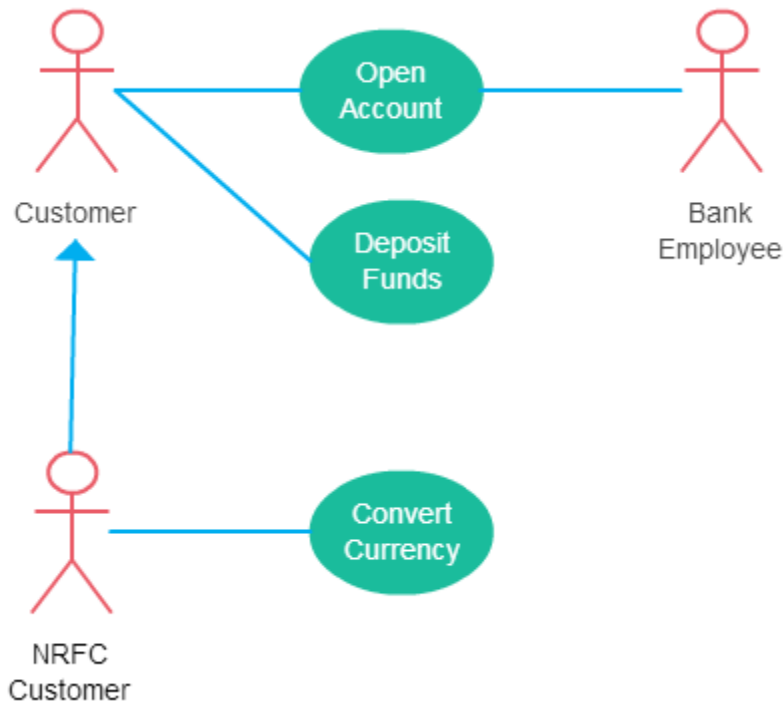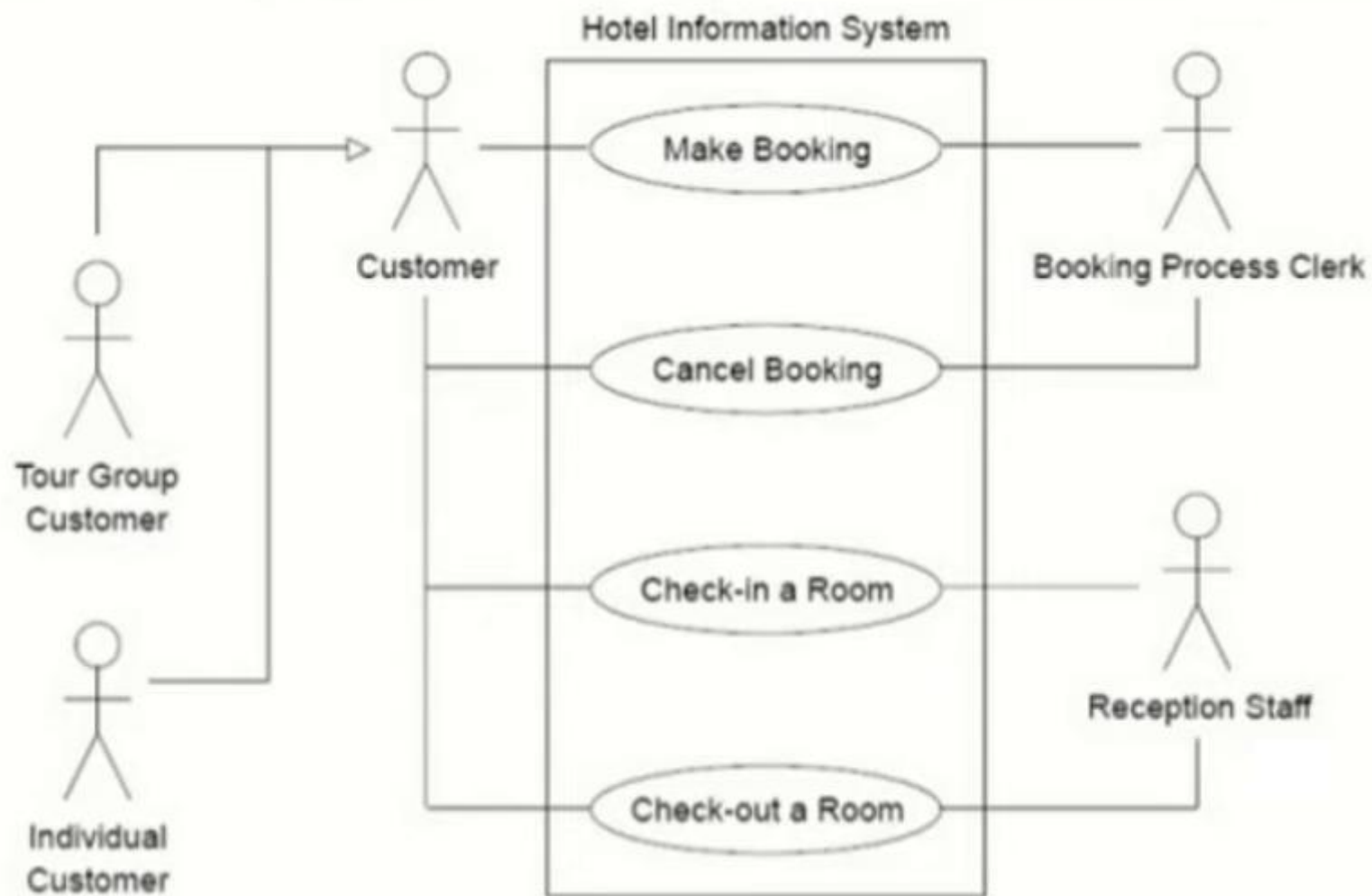|	
1	The guest/receptionist chooses to make a reservation.
2	The guest/receptionist selects the desired hotel, dates and type of room.
3	The hotel system provides the availability and price for the request. (An offer is made.)
4	The guest/receptionist agrees to proceed with the offer.
5	The guest/receptionist provides identification and contact details for the hotel system's records.
6	The guest/receptionist provides payment details.
7	The hotel system creates a reservation and gives it an identifier.
8	The hotel system reveals the identifier to the guest/receptionist.
9	The hotel system creates a confirmation of the reservation and sends it to the guest.

Use Case Diagram Relationships Explained

- **Association Between Actor and Use Case**
- An actor must be associated with at least one use case.
- An actor can be associated with multiple use cases.
- Multiple actors can be associated with a single use case.
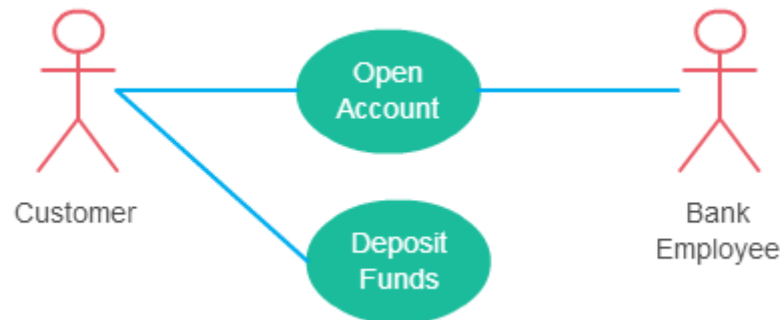
Generalization of an Actor

- Generalization of an actor means that one actor can inherit the role of the other actor. The descendant inherits all the use cases of the ancestor. The descendant has one or more use cases that are specific to that role.

Hotel Information System

Make Booking

Cancel Booking

Check-in a Room

Check-out a Room

Customer

Tour Group Customer

Individual Customer

Booking Process Clerk

Reception Staff

Include Relationship Between Two Use Cases

- Include relationship show that the behavior of the included use case is part of the including (base) use case. The main reason for this is to reuse common actions across multiple use cases. In some situations, this is done to simplify complex behaviors. Few things to consider when using the <<include>> relationship.

- The base use case is incomplete without the included use case.

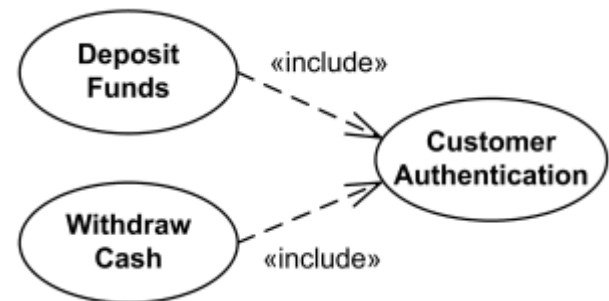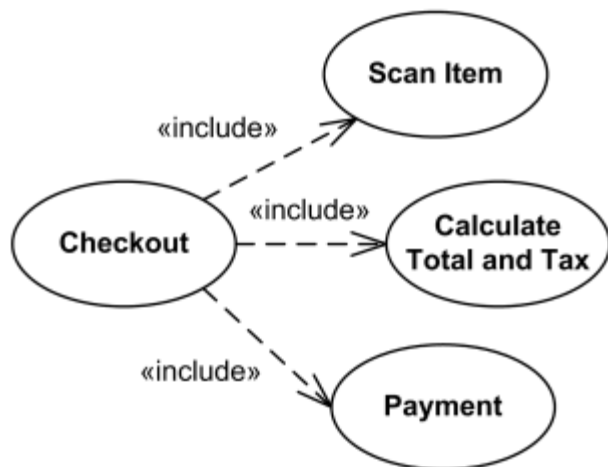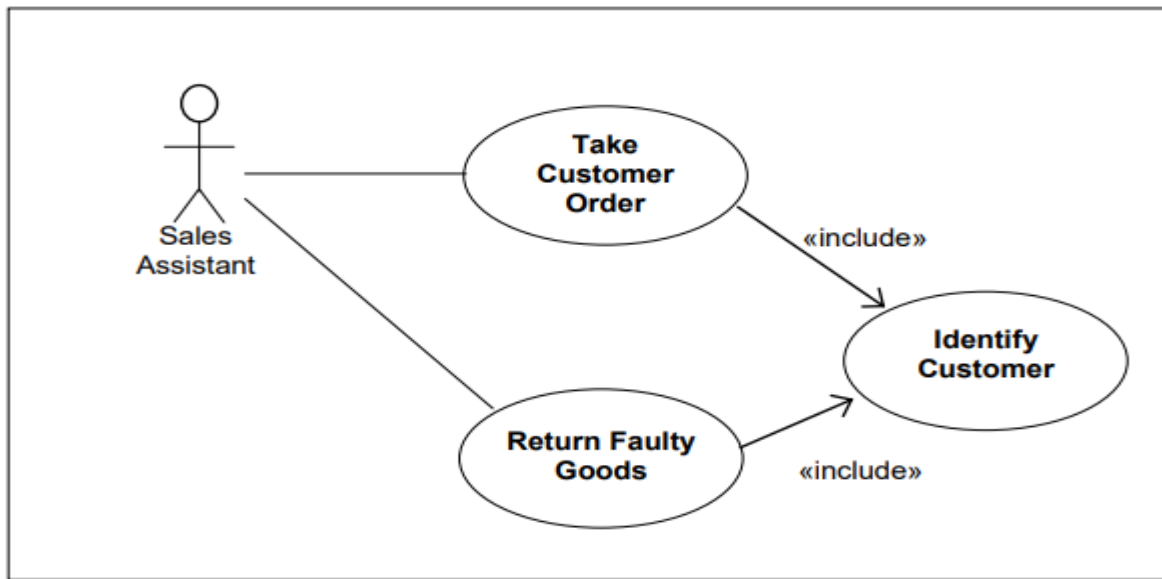- The included use case is mandatory and not optional.
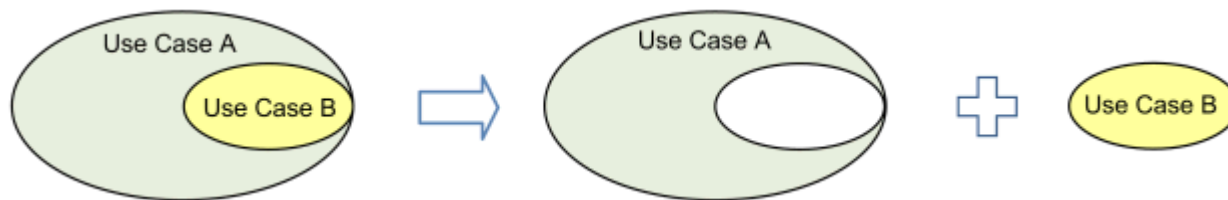
If Use Case A includes Use Case B, then A cannot be completed without also doing B.
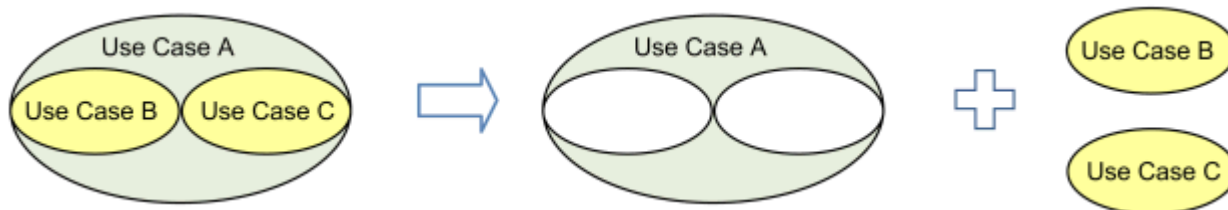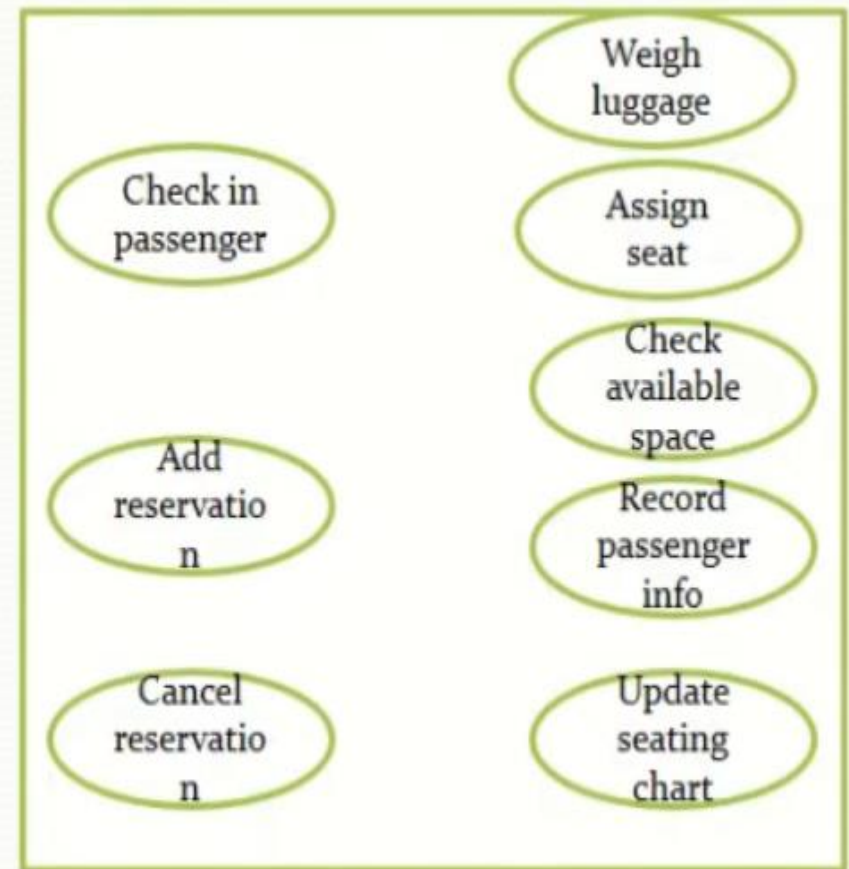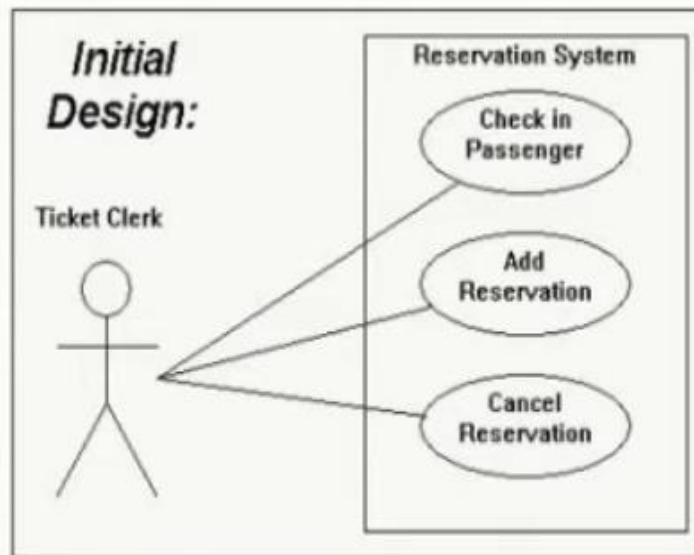
Similarly, if Use Case A includes both Use Case B and Use Case C, then A is made up of both B and C.

Include Relationship Between Two Use Cases

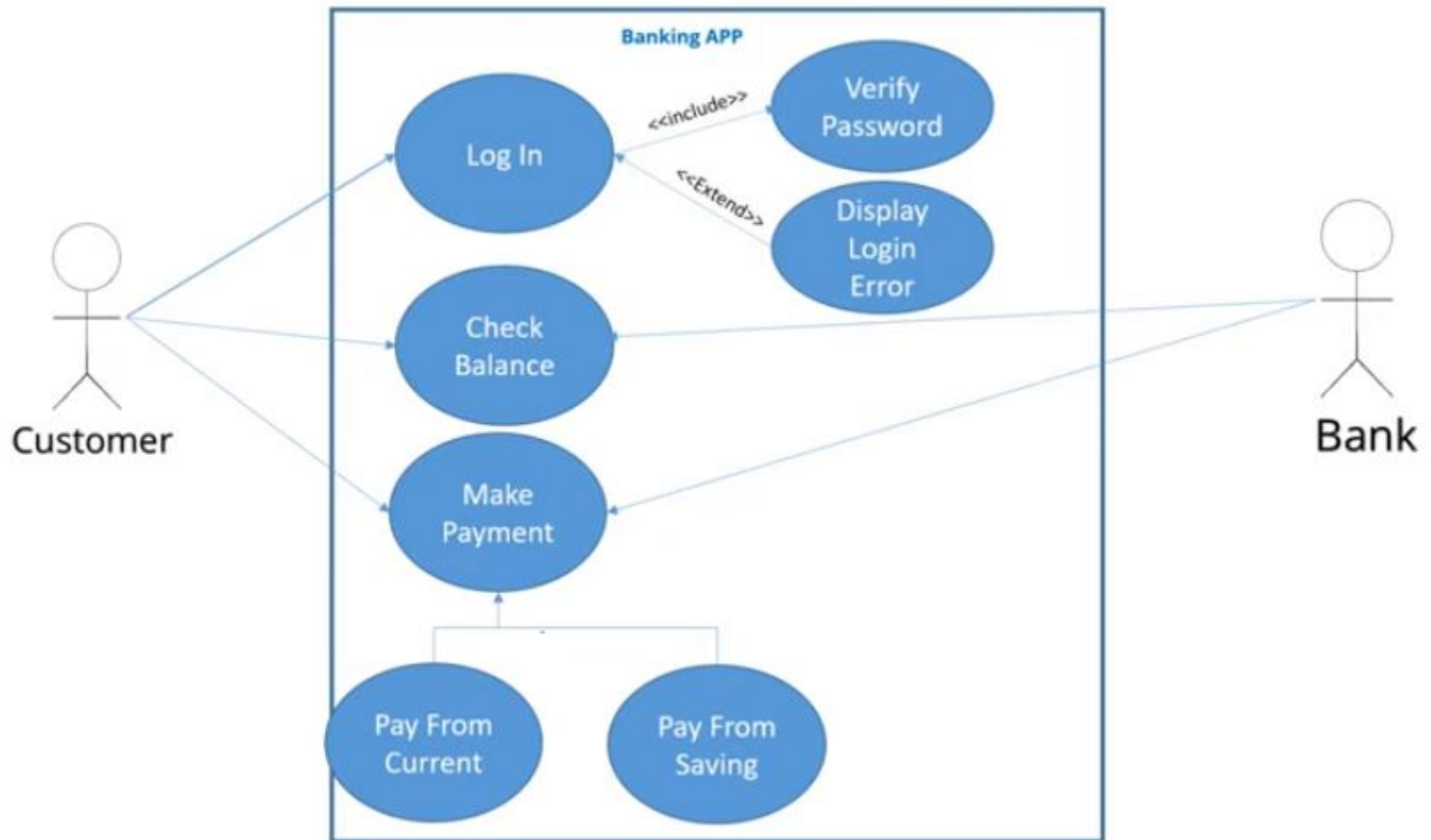# Include Relationship Between Two Use Cases
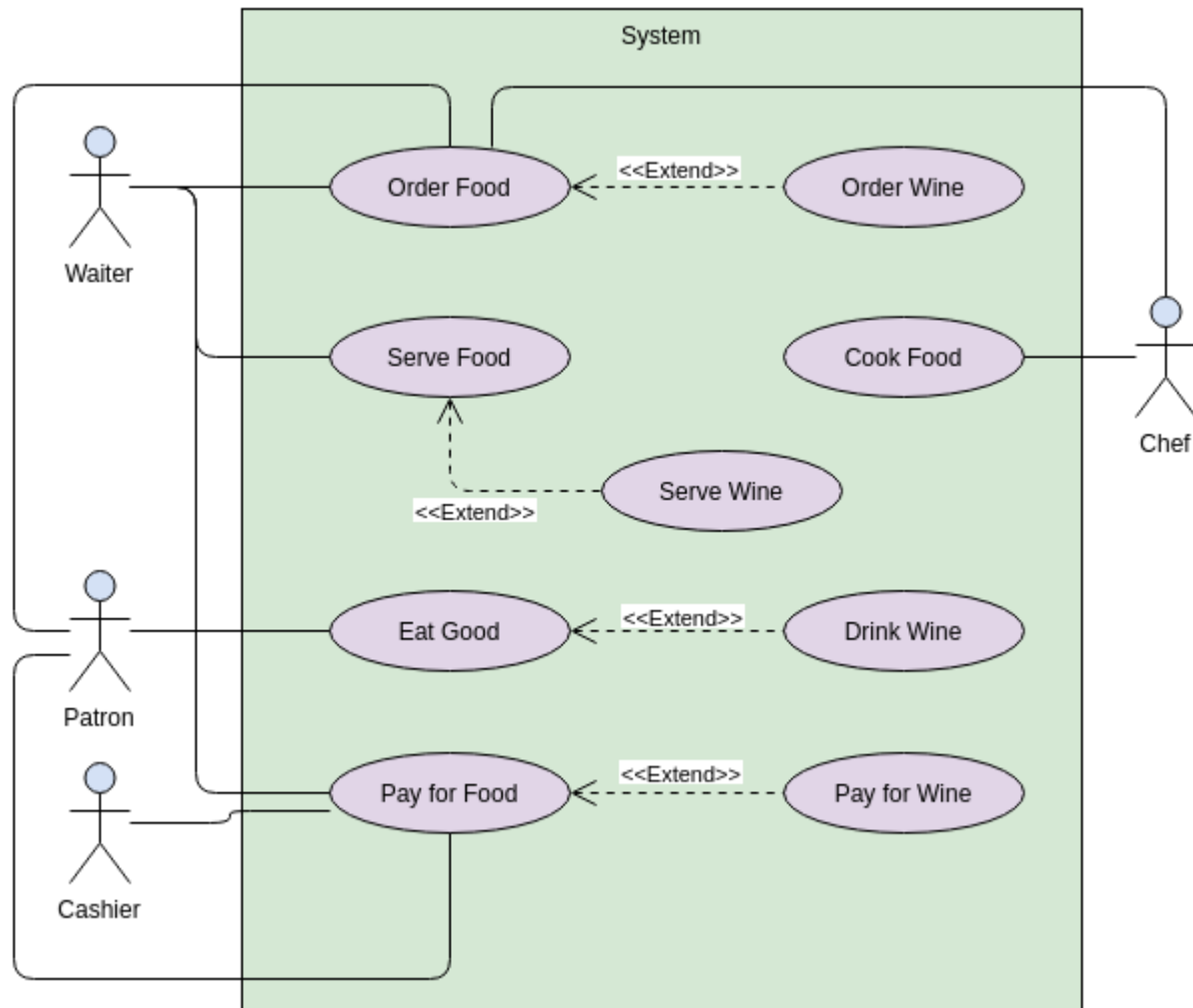
Extend Relationship Between Two Use Cases

- Many people confuse the extend relationship in use cases. As the name implies it extends the base use case and adds more functionality to the system. Here are a few things to consider when using the <<**extend**>> relationship.

- **The extending use case is dependent on the extended (base) use case**. In the below diagram the "Calculate Bonus" use case doesn't make much sense without the "Deposit Funds" use case.

- **The extending use case is usually optional** and can be triggered conditionally. In the diagram, you can see that the extending use case is triggered only for deposits over 10,000 or when the age is over 55.

- **The extended (base) use case must be meaningful on its own**. This means it should be independent and must not rely on the behavior of the extending use case.

An extend relationship lets you add extra, sometimes optional steps to a use case, but the main (base) use case can still work perfectly on its own. The extension only happens if certain conditions are met.

Extend Relationship Between Two Use Cases

Extend Relationship Between Two Use Cases

Simple ATM Machine System

Administrator (Third-party) — System Maintainence

System Maintainence <<include>> System Reporting

System Maintainence <<include>> System Shutdown

Bank Customer — Login

Login <<include>> Transaction

Bad PIN <<extend>> Transaction

Transaction — Bank

Deposit → Transaction

Withdraw

Check Balance

Print Receipt

Travel agency — System

- Book a tour
- Book airline ticket(s) -- <<extend>> --> Book a tour
- Reserve a seat
- Deliver airline ticket(s) -- <<extend>> --> Arrange tour
- Pay travel agent -- <<include>> --> Pay for airline ticket(s)
- Pay travel agent -- <<include>> --> Travel agent commission
- Pay for airline ticket(s) -- <<extend>> --> Pay for tour

Type something

fddddfdff

Testing

Useful Links

- [10 Use Case Diagram Examples (and How to Create Them) – Venngage](#)

- [Use Case Diagrams - Use Case Diagrams Online, Examples, and Tools (smartdraw.com)](#)

- [Karona (karonaconsulting.com)](#)

- [Use Case Diagram Example: Generalization Use Case | Use Case Diagram Template (visual-paradigm.com)](#)