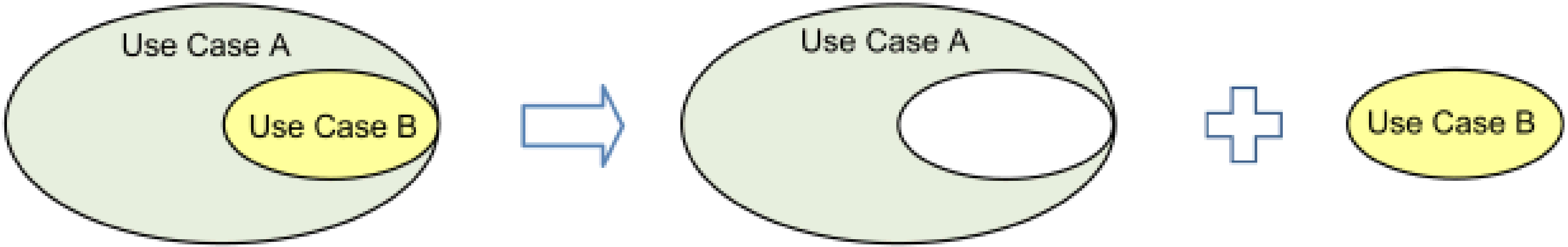
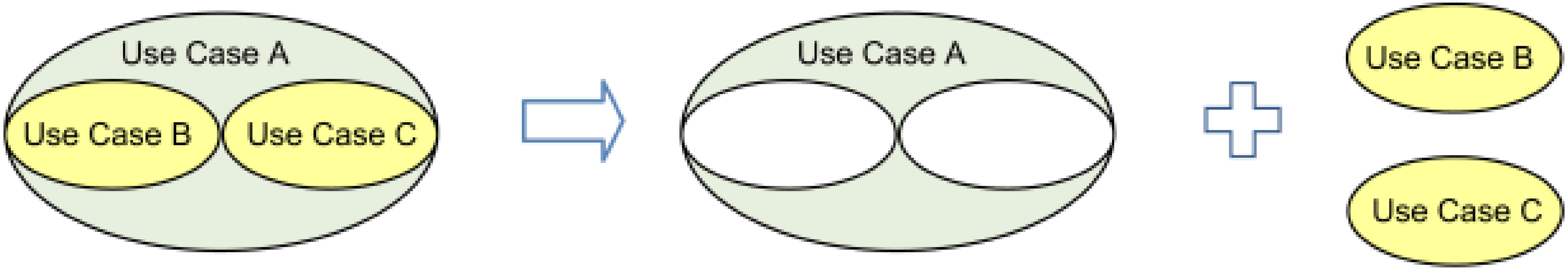
# <<Include>> in Use case

- The include relationship could be used:
  - To simplify large use case by splitting it into several use cases
  - To extract common parts of the behaviors of two or more use cases.
  - The base use case is incomplete without the included use case.
  - The included use case is mandatory and not optional.

# <<Include>> in Use case

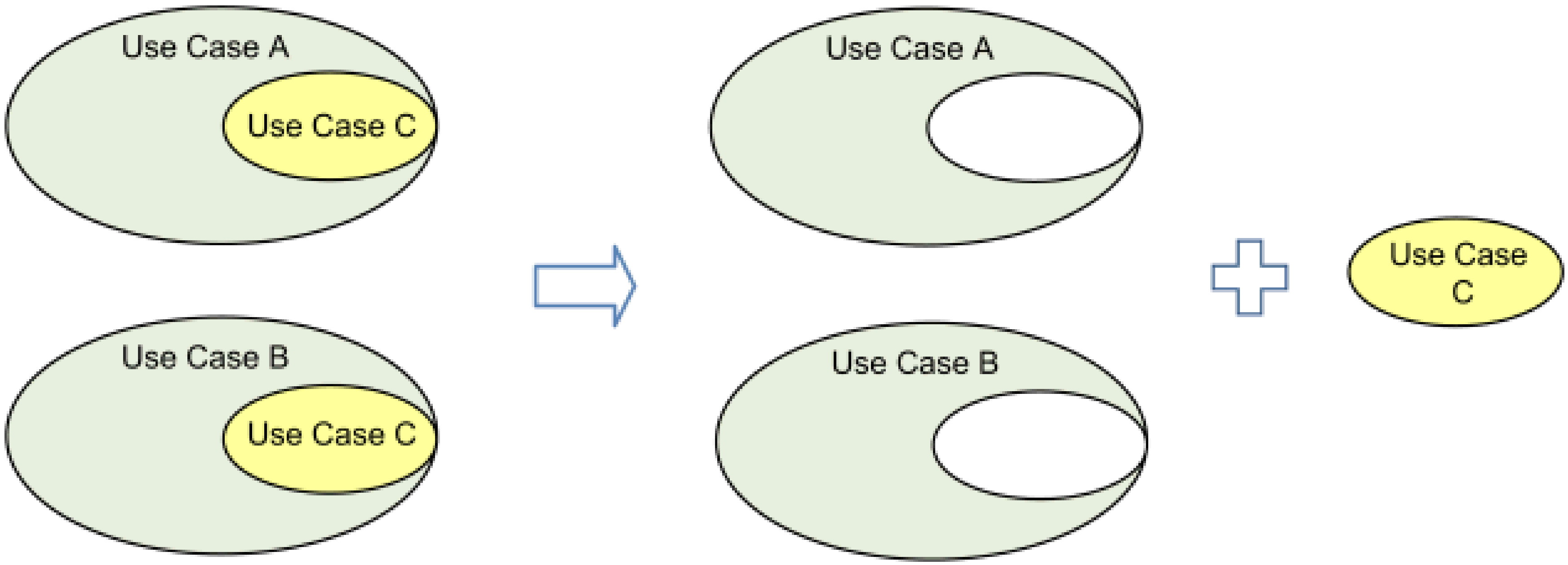


*Use case B is extracted from larger use case A into a separate use case.*



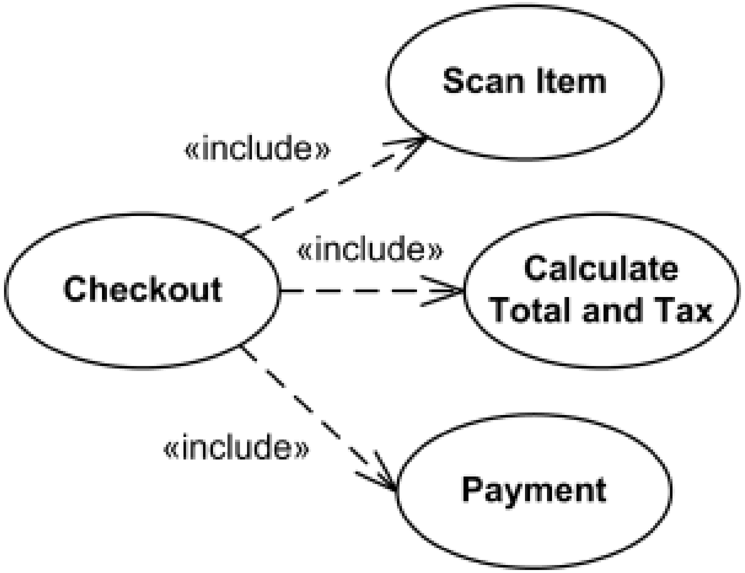
*Use cases B and C are extracted from larger use case A into separate use cases.*

# <<Include>> in Use case

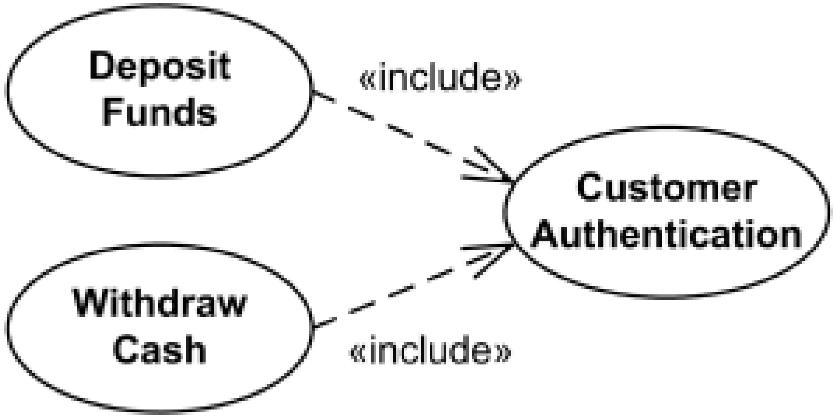


*Use case C is extracted from use cases A and B to be reused by both use cases using UML include relationship.*

# <<Include>> in Use case



*Checkout use case includes several use cases - Scan Item, Calculate Total and Tax, and Payment*

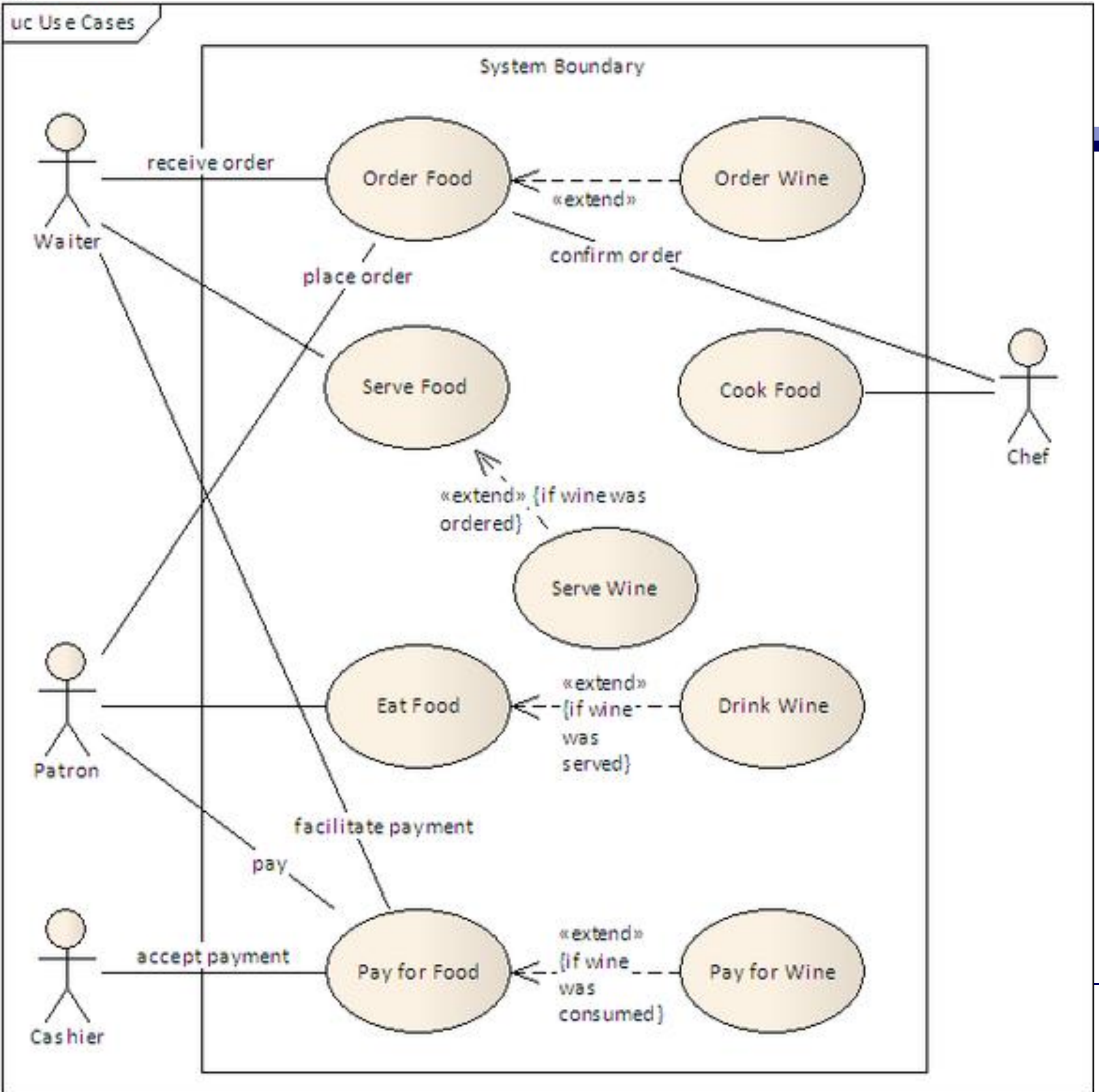


***Deposit Funds** and **Withdraw Cash** use cases include **Customer Authentication** use case.*

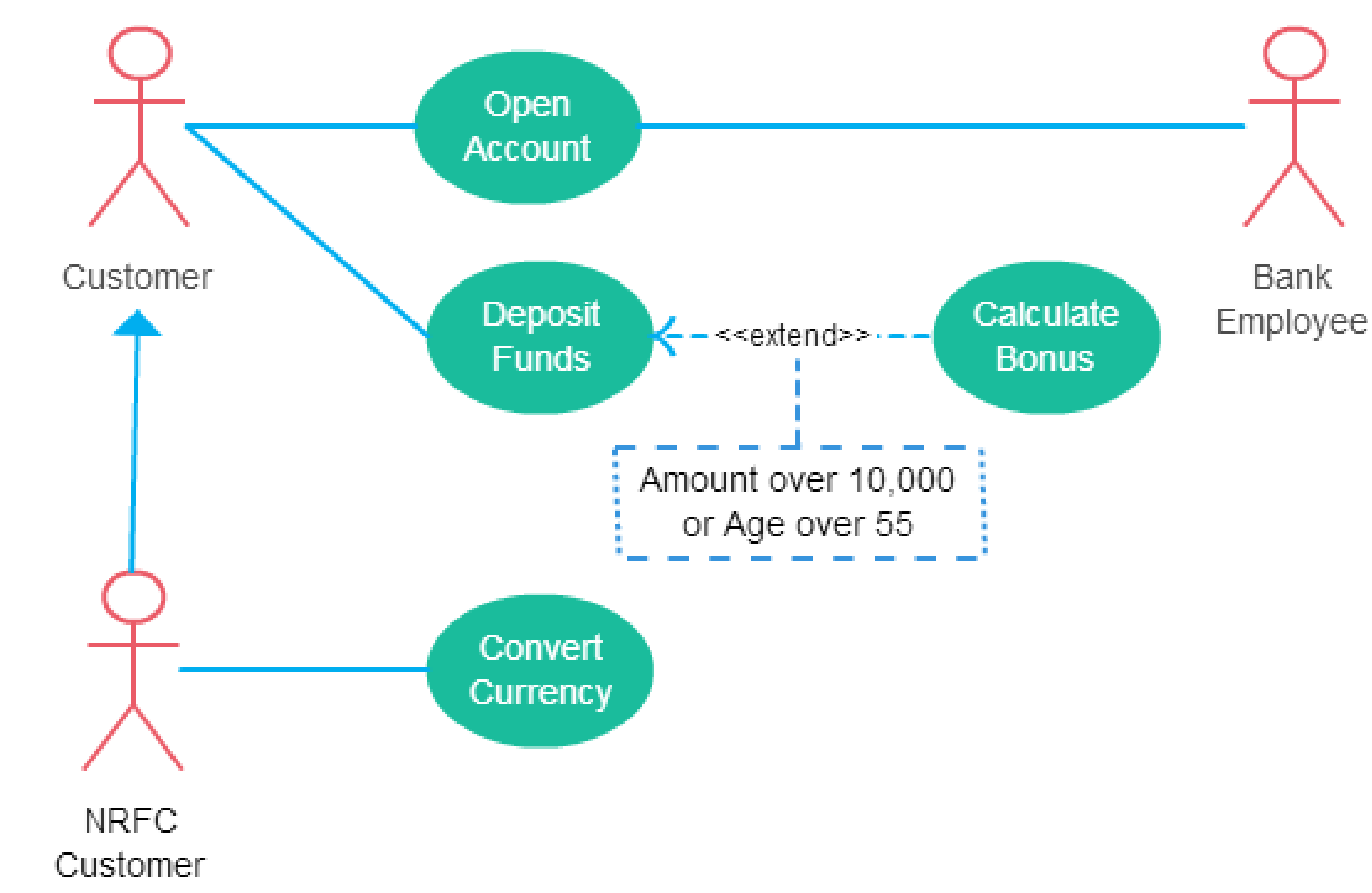
# <<extend>> in Use case

- Extends relationship is **"optional"**!
- It adds further functionality to the base use case which may be restricted by constraints
- The extending use case is dependent on the extended (base) use case.
- The extended (base) use case must be meaningful on its own.

# <<extend>> in Use case

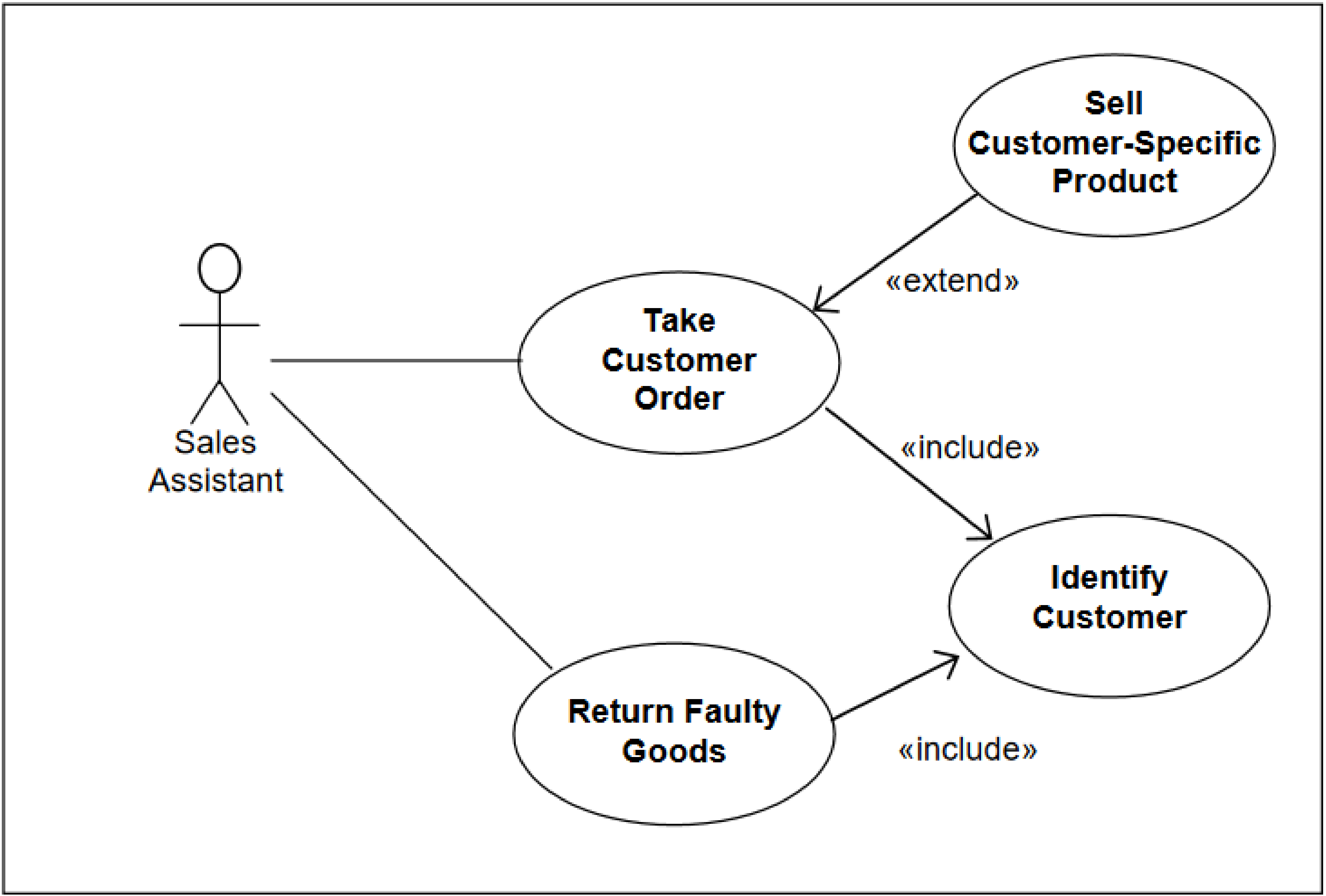

uc Use Cases
System Boundary
Waiter
receive order
Order Food
«extend»
Order Wine
confirm order
place order
Serve Food
Cook Food
Chef
«extend» {if wine was ordered}
Serve Wine
Patron
Eat Food
«extend» {if wine was served}
Drink Wine
facilitate payment
pay
Cashier
accept payment
Pay for Food
«extend» {if wine was consumed}
Pay for Wine

# <<extend>> in Use case



*Extend relationship in use case diagrams*

# <<extend>> in Use case


Sell Customer-Specific Product
«extend»
Take Customer Order
Sales Assistant
«include»
Identify Customer
Return Faulty Goods
«include»

# How to Create a Use Case Diagram

1. **Identifying Actors**
2. **Identifying Use Cases**
3. **Look for Common Functionality to use Include**
4. **Is it Possible to Generalize Actors and Use Cases**
5. **Optional Functions or Additional Functions**

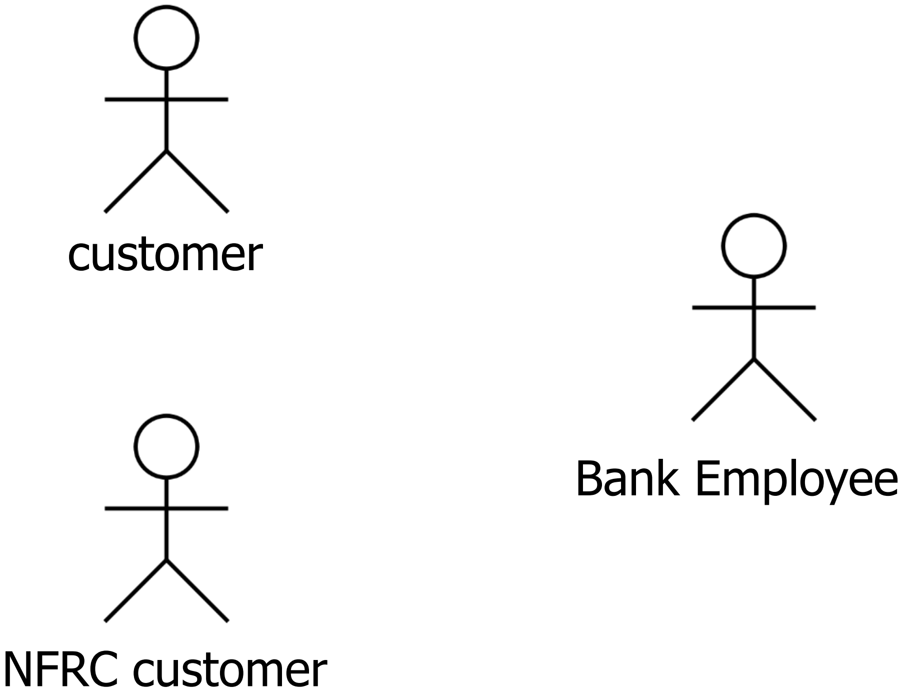
# How to Create a Use Case Diagram

## Identifying Actors

Customer

Bank employee

NFRC Customer



customer

Bank Employee

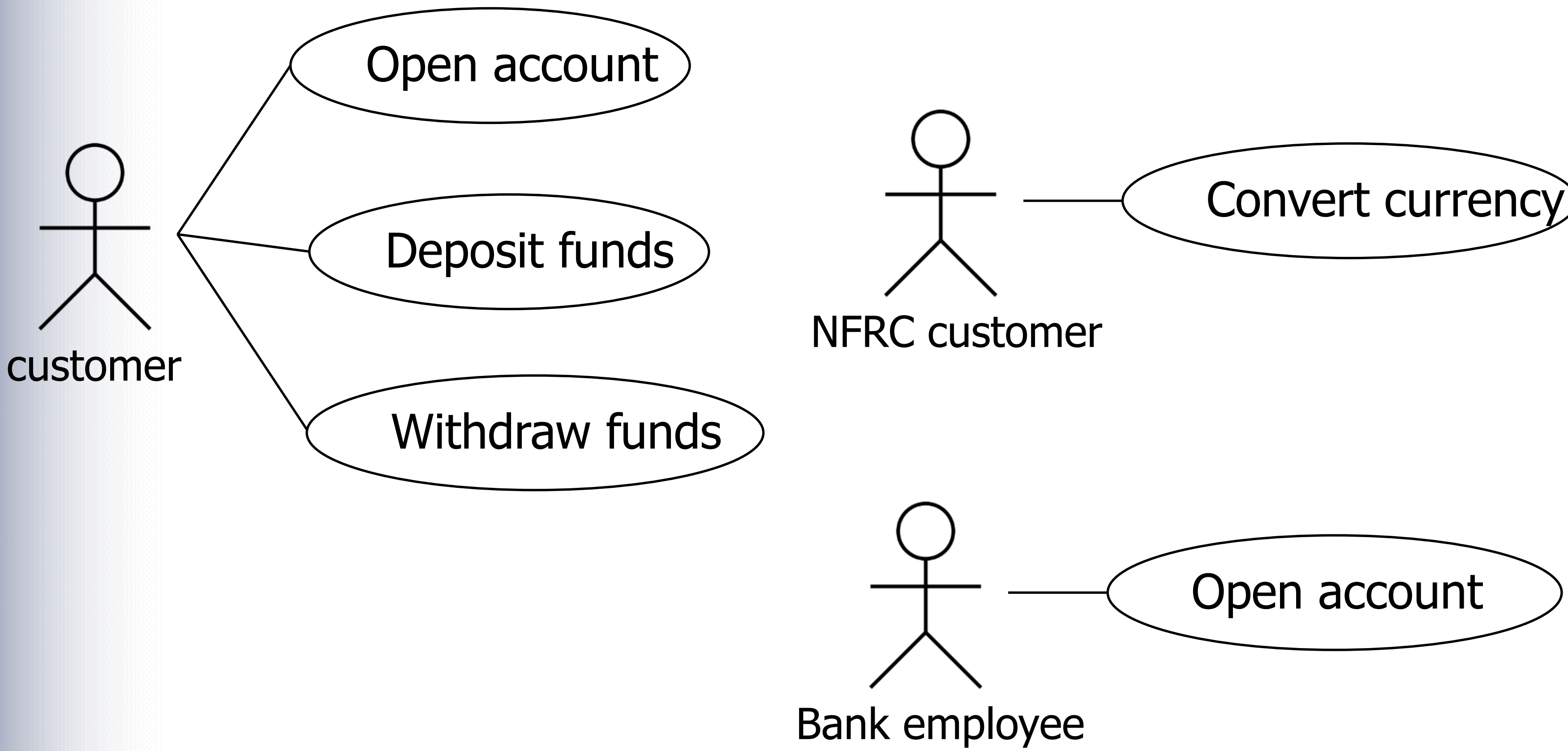
NFRC customer

# How to Create a Use Case Diagram

## Identifying Use Cases

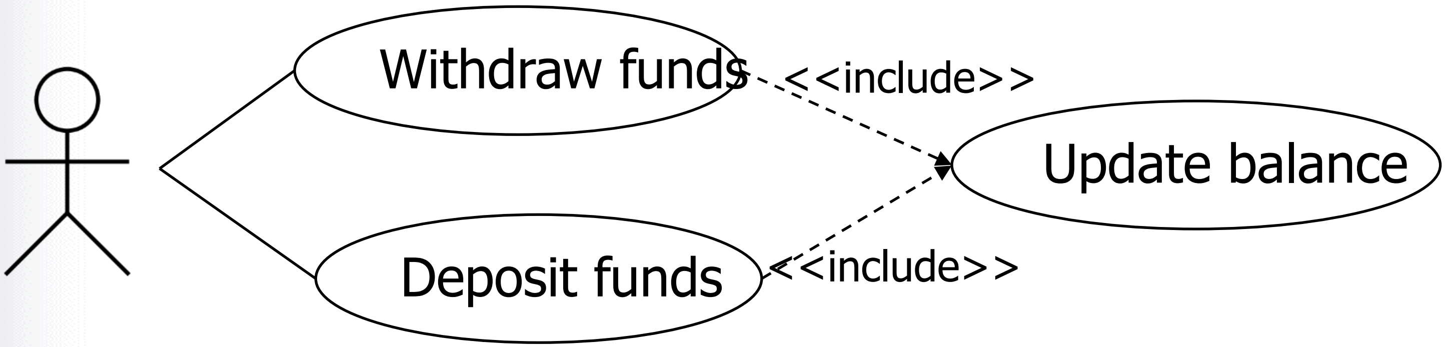
- A good way to do this is to identify what the actors need from the system
- A customer will need to
  1. open accounts
  2. Deposit funds
  3. withdraw funds
  4. request check books

# How to Create a Use Case Diagram



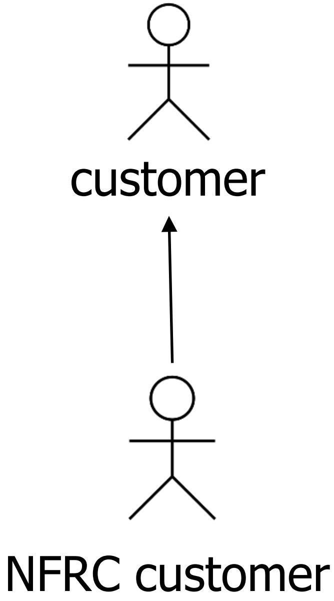
# How to Create a Use Case Diagram

- **Look for Common Functionality to use Include**
  - find two or more use cases that share common functionality



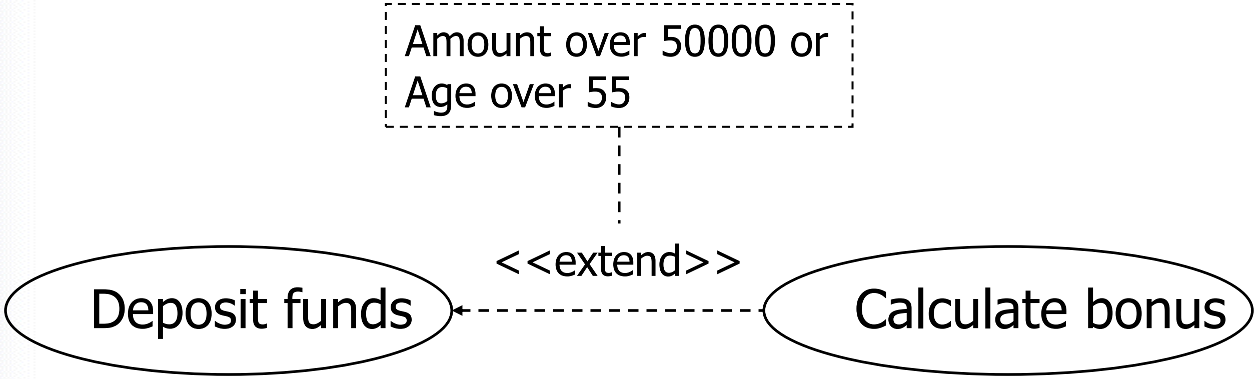
# How to Create a Use Case Diagram

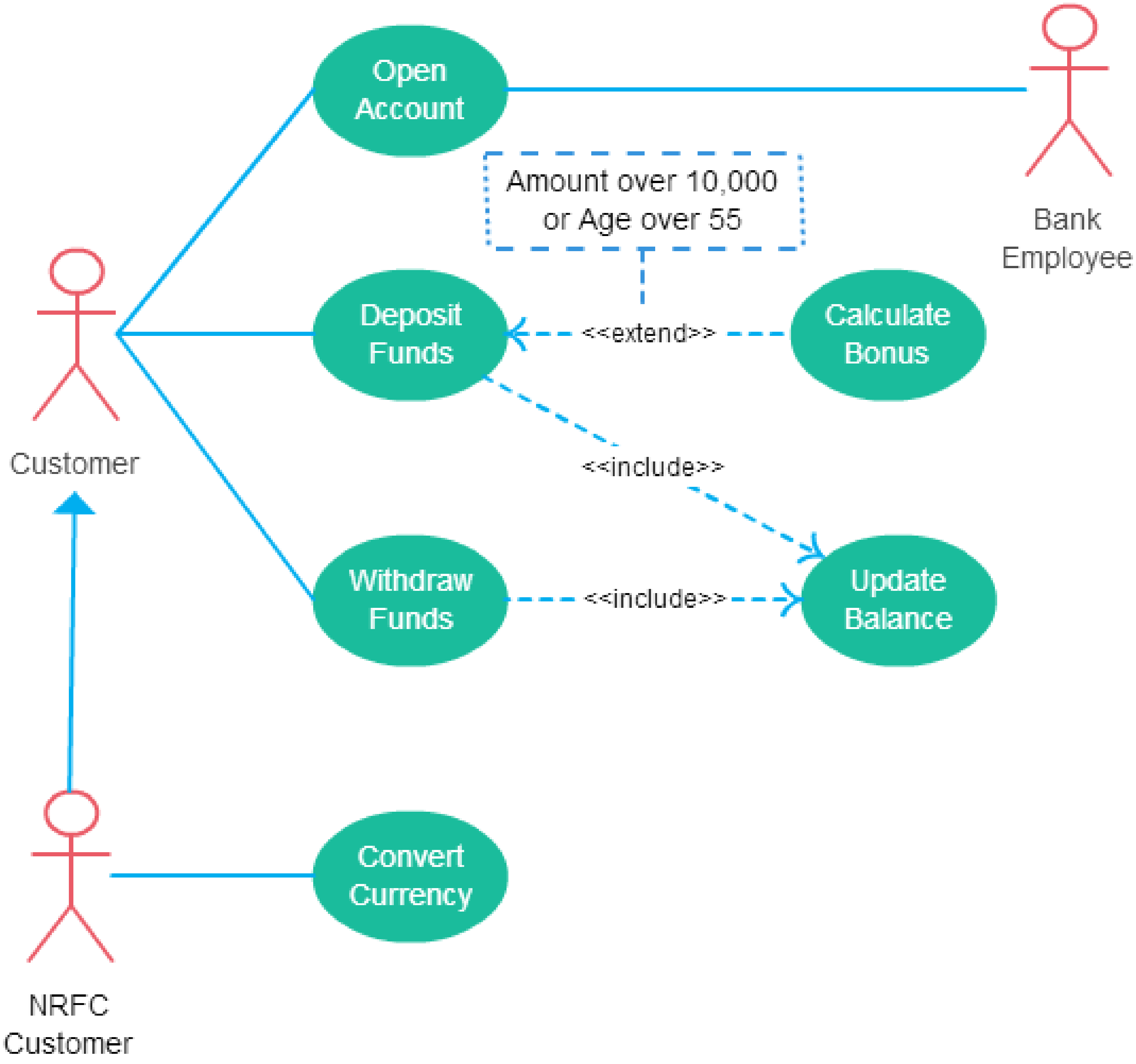
- **Is it Possible to Generalize Actors ?**



# How to Create a Use Case Diagram

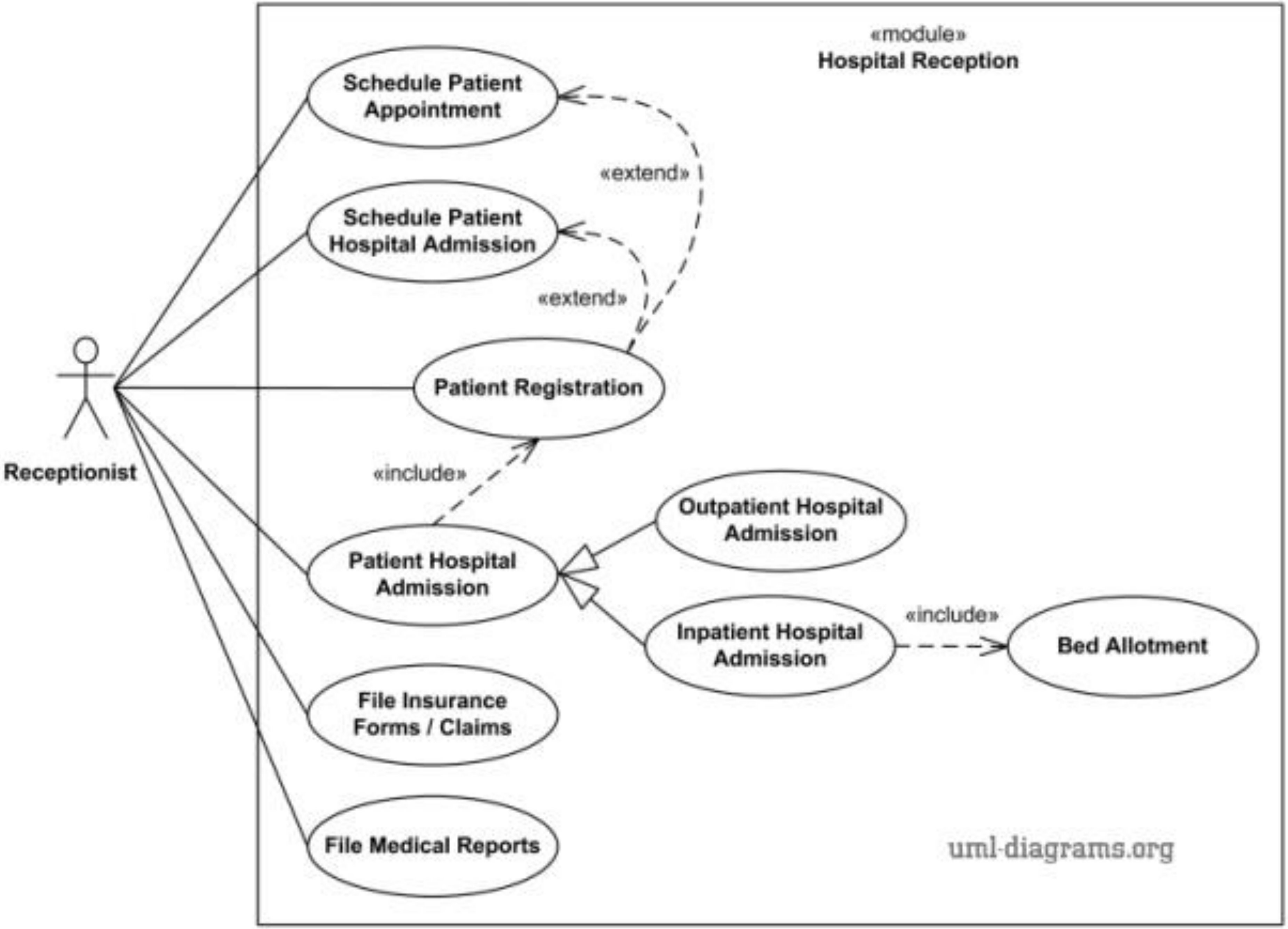
- **Optional Functions or Additional Functions**
  - There are some functions that are triggered optionally




Open Account
Bank Employee
Amount over 10,000 or Age over 55
Customer
Deposit Funds
<<extend>>
Calculate Bonus
<<include>>
Withdraw Funds
<<include>>
Update Balance
NRFC Customer
Convert Currency

# Practice

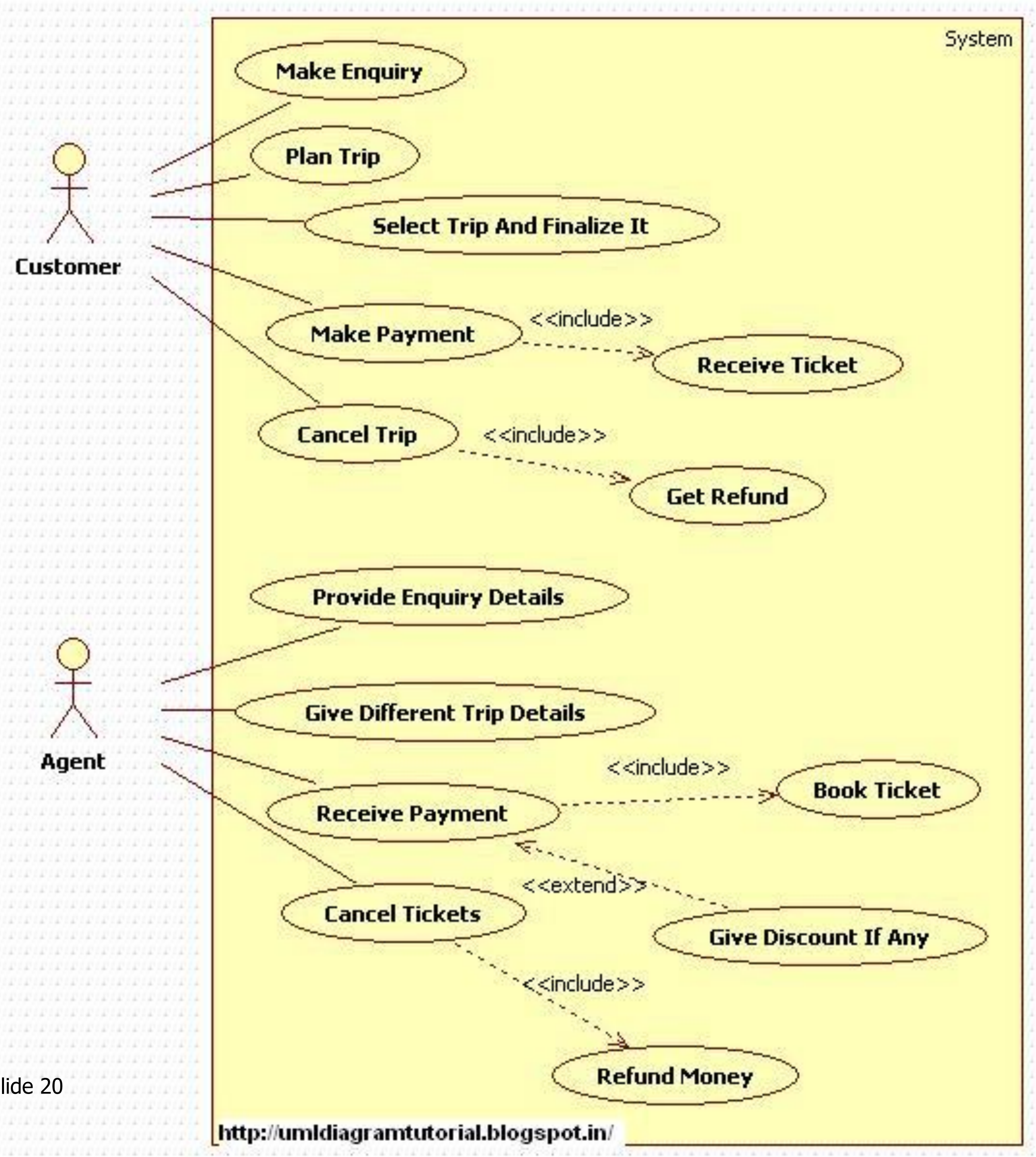
- Hospital Management System is a large system including several subsystems or modules providing variety of functions. UML use case diagram example below shows actor and use cases for a hospital's reception.
- Purpose: Describe major services (functionality) provided by a hospital's reception.
- Hospital Reception subsystem or module supports some of the many job duties of hospital receptionist. 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.


«module»
Hospital Reception
Receptionist
Schedule Patient Appointment
Schedule Patient Hospital Admission
«extend»
«extend»
Patient Registration
«include»
Patient Hospital Admission
Outpatient Hospital Admission
Inpatient Hospital Admission
«include»
Bed Allotment
File Insurance Forms / Claims
File Medical Reports
uml-diagrams.org

WILEY

- UML Use Case diagram for Travel Agency is shown below. The various participants of the same are detailed below:
- Actors: - Customer, Agent

- **The corresponding use cases for these actors are:-**
  - Customer: Make Enquiry, Plan Trip, Select a Trip and Finalize It, Make Payment, Cancel Trip, Receive Ticket, Get Refund
  - Agent: Provide Enquiry Details, Give Different Trip Details, Receive Payment, Cancel Tickets, Book Ticket, Give Discount If Any, Refund Money
- **Other relationships:**
  - While a customer makes a payment will receive the ticket and get a refund for the cancelation of the ticket.
  - The agent will book a ticket while receiving payment and give a discount if applied. It also refunds money while canceling tickets.
  - Customer and Agent can be divided into two types, new and existing.
  - **You have to show include, extern, and generalization relationship.



http://umldiagramtutorial.blogspot.in/

- Bank customer has to provide pin for login, verified by bank, there could be mistake while entering the pin, customer can do transactions like (fund transfer, withdraw, change pin, balance check, deposit) etc., ATM Machine will provide a print out if customer wants to get a receipt of their transaction considering some charges, charges will also apply for each transaction. There is system administrator who monitors users’ transaction and report if any suspicious activities is noticed. Also maintains ATM machine money loading and maintenance.