# HNDIT2032 System Analysis and Design

Advanced Technological Institute
Galle

- presents techniques for documenting the functional requirements by creating a variety of models.
- These models are created as part of the analysis activity Define functional requirements, although remember that the analysis activities are actually done in parallel and in each iteration of the project.

# 1. Use Cases and User Goals

- One approach to identifying use cases, called the user goal technique, is to ask users to describe their goals for using the new or updated system.

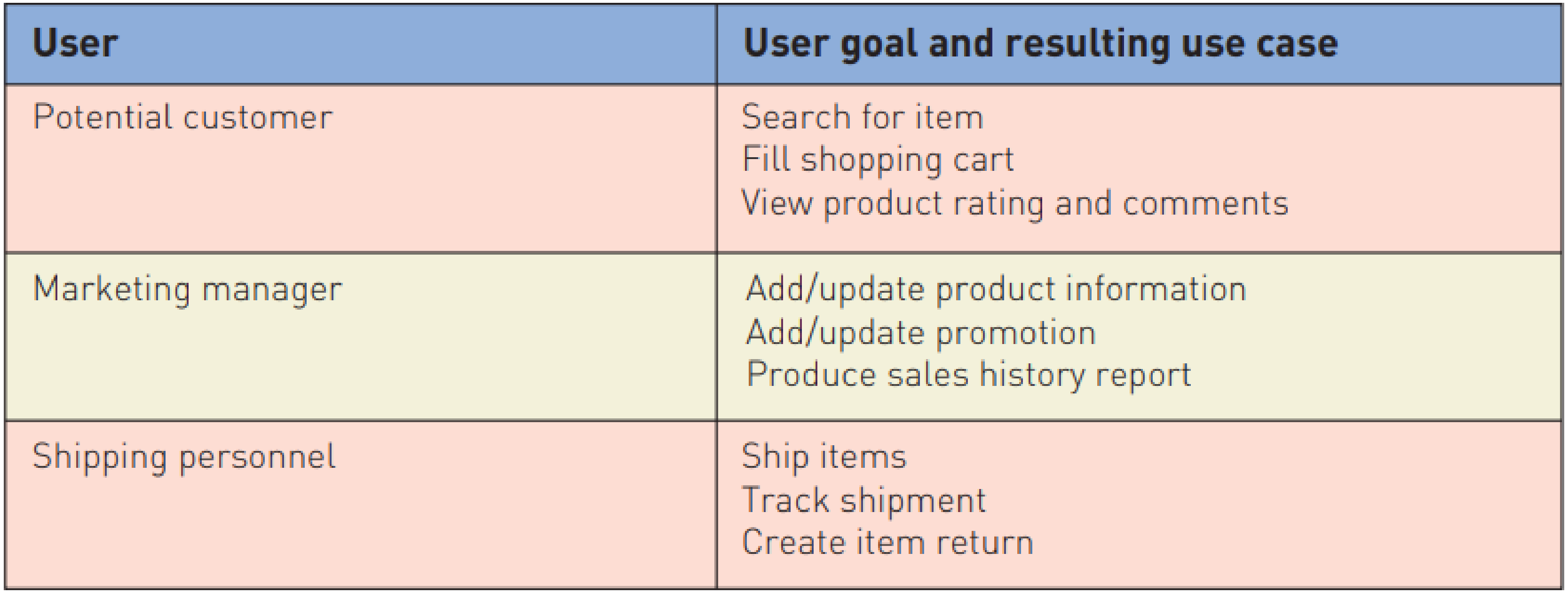
# user goal technique

- The user goal technique for identifying use cases includes these steps:

1. Identify all the potential users for the new system.
2. Classify the potential users in terms of their functional role (e.g., shipping, marketing, sales).
3. Further classify potential users by organizational level (e.g., operational, management, executive).

# user goal technique..

4. For each type of user, interview them to find a list of specific goals they will have when using the new system. Start with goals they currently have and then get them to imagine innovative functions they think would add value.
5. Create a list of preliminary use cases organized by type of user.
6. Look for duplicates with similar use case names and resolve inconsistencies.
7. Identify where different types of users need the same use cases.
8. Review the completed list with each type of user and then with interested stakeholders

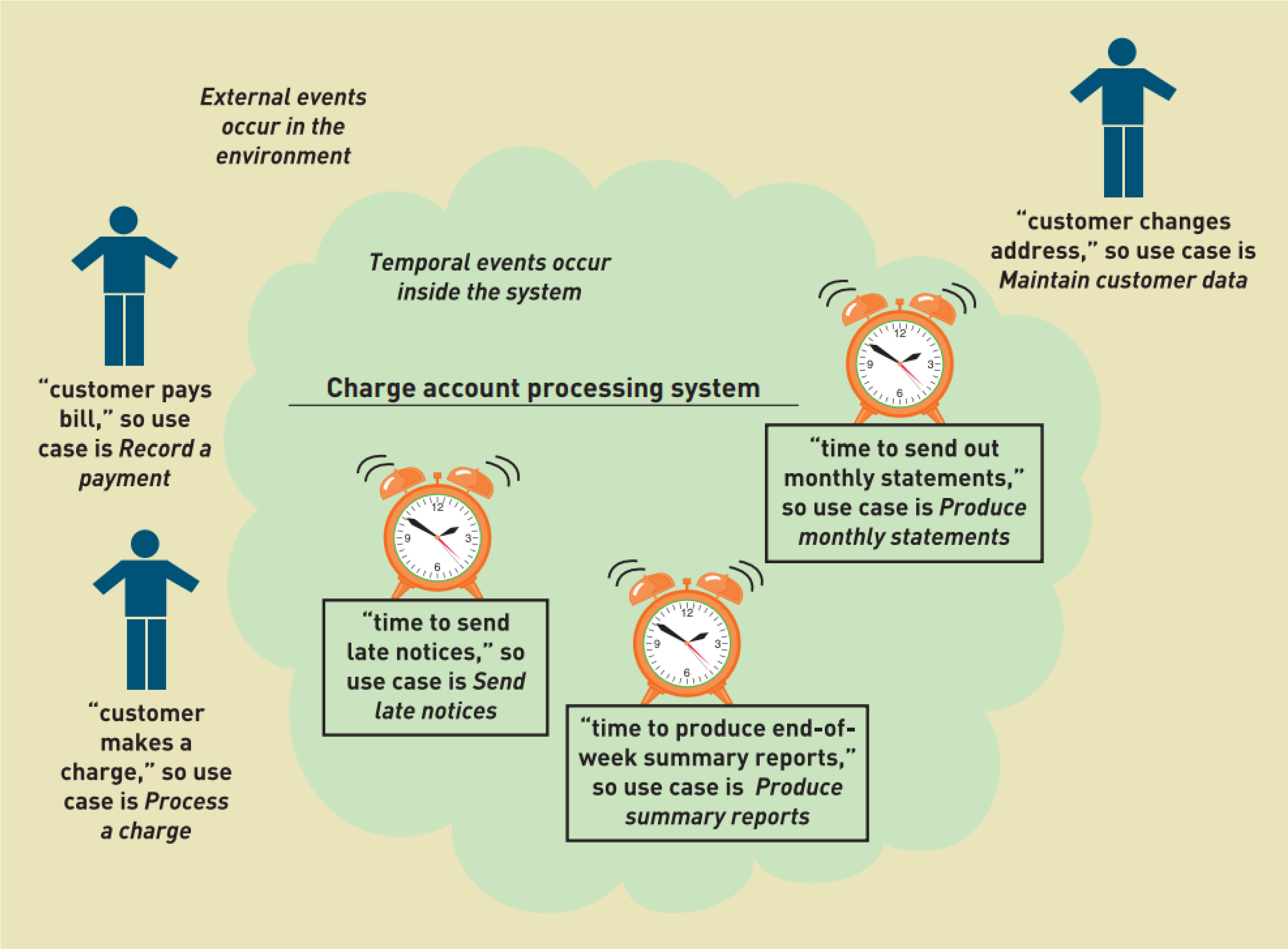


| User | User goal and resulting use case |
|---|---|
| Potential customer | Search for item<br>Fill shopping cart<br>View product rating and comments |
| Marketing manager | Add/update product information<br>Add/update promotion<br>Produce sales history report |
| Shipping personnel | Ship items<br>Track shipment<br>Create item return |

# 2. Use Cases and Event Decomposition

- The event decomposition technique begins by identifying all the business events that will cause the information system to respond, and each event leads to a use case.
- focuses on identifying the events to which a system must respond and then determining how a system must respond

3-2 *Events in a charge account processing system that lead to use cases*



*External events occur in the environment*

*Temporal events occur inside the system*

**Charge account processing system**

"customer pays bill," so use case is *Record a payment*

"customer makes a charge," so use case is *Process a charge*

"customer changes address," so use case is *Maintain customer data*

"time to send late notices," so use case is *Send late notices*

"time to produce end-of-week summary reports," so use case is *Produce summary reports*

"time to send out monthly statements," so use case is *Produce monthly statements*

# Types of Events

- There are three types of events to consider when using the event decomposition technique to identify use cases:
  - external events
  - temporal events
  - state events

# An external event

- An external event is an event that occurs outside the system—usually initiated by an external agent or actor.
- An external agent (or actor) is a person or organizational unit that supplies or receives data from the system.
- Eg: The customer may want to place an order for one or more products.(Ordering process)

# temporal event

- an event that occurs as a result of reaching a point in time.
- Many information systems produce outputs at defined intervals, such as payroll systems that produce a paycheck every two weeks (or each month).

- The analyst begins identifying temporal events by asking about the specific deadlines that the system must accommodate.
  - What outputs are produced at that deadline?
  - What other processing might be required at that deadline?
  - EG: “Time to produce biweekly payroll.”
  - “Time to produce monthly sales summary report.”

# state event (Internal Event)

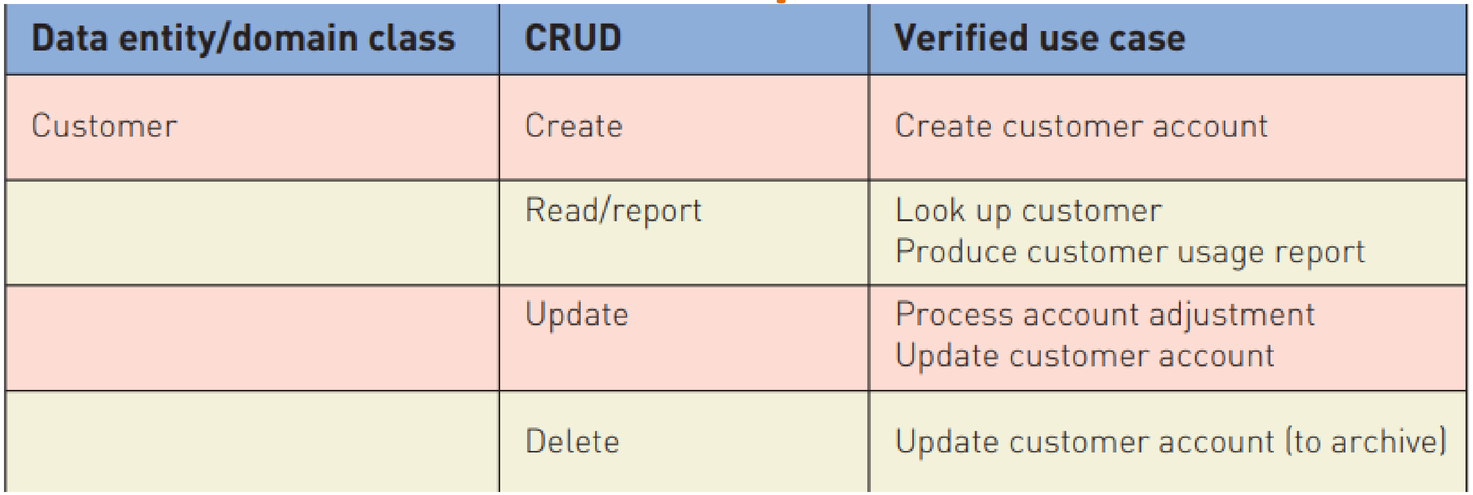
- an event that occurs when something happens inside the system that triggers the need for processing.
  - . For example, if the sale of a product results in an adjustment to an inventory record and the inventory in stock drops below a reorder point, it is necessary to reorder.

# Use Cases and CRUD

Important technique used to validate and refine use cases is the CRUD technique.

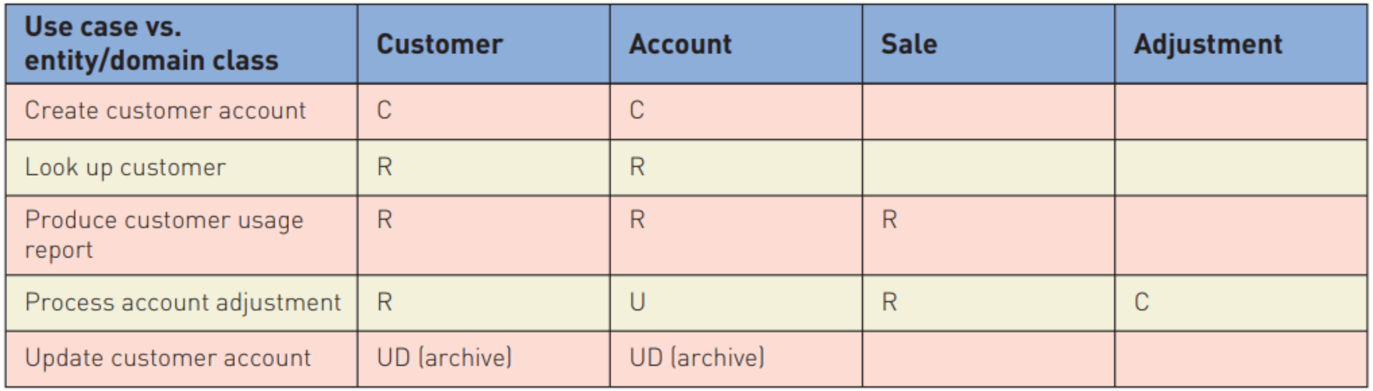
- "CRUD" is an acronym for Create, Read or Report, Update, and Delete.
- The CRUD technique is most useful when used as a cross-check along with the user goal technique.
- Another use of the CRUD technique is to summarize all use cases and all data entities/domain classes to show the connection between use cases and data.

# Verifying use cases with the CRUD technique



| Data entity/domain class | CRUD | Verified use case |
|---|---|---|
| Customer | Create | Create customer account |
| | Read/report | Look up customer<br>Produce customer usage report |
| | Update | Process account adjustment<br>Update customer account |
| | Delete | Update customer account (to archive) |

*CRUD table showing use cases and corresponding data entities/domain classes*



| Use case vs. entity/domain class | Customer | Account | Sale | Adjustment |
|---|---|---|---|---|
| Create customer account | C | C | | |
| Look up customer | R | R | | |
| Produce customer usage report | R | R | R | |
| Process account adjustment | R | U | R | C |
| Update customer account | UD (archive) | UD (archive) | | |

# Use Case diagram

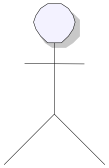
A use case diagram is a graphical depiction of the user’s interaction with a system and interactions among the elements of a system.

The use case diagram is the UML model used to graphically show the use cases and their relationship to users.

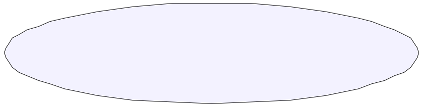
# Use Case diagram

- Use case diagram illustrates
  - A set of use cases for a system
  - The actors of these use cases
  - Relations between the actors and use cases
  - Relations among the use cases.
  - System boundary

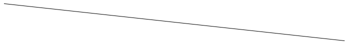
# Use Case Diagram Components



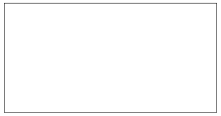
Actor



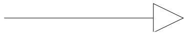
Use case



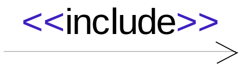
Actor to Use case Association



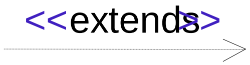
System Boundary



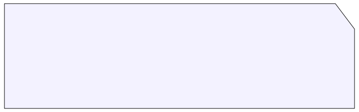
Inheritance



Inclusion Of a Use case by Another



Extension of a Use case by Another



Notes

# Actor

Type of *users of the system* or external systems that the system interacts with.

- Actor may anything that needs to interact with the system to exchange information
  - User
  - Separate computer system
  - External device

# Actor Variations

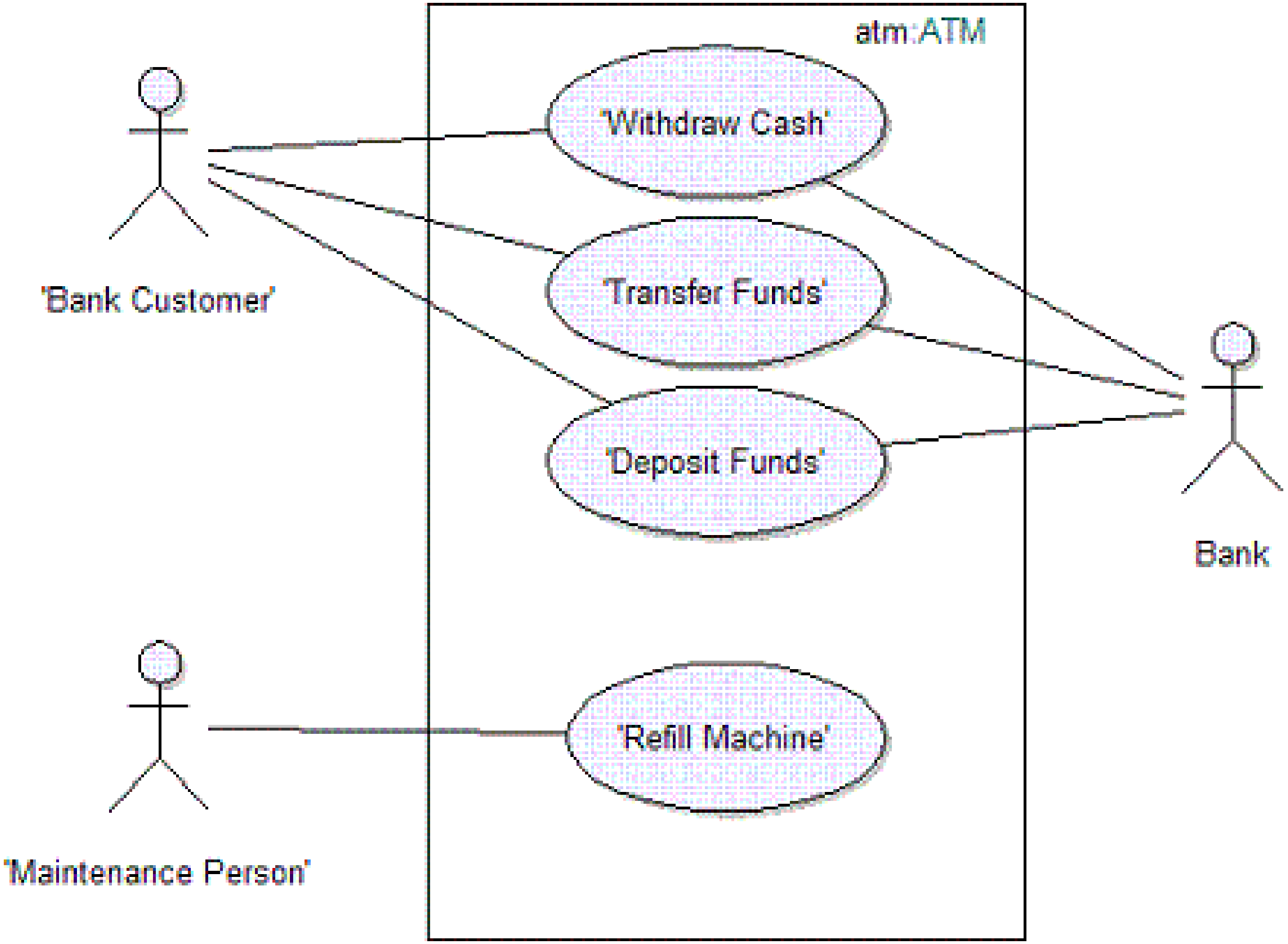
- Primary actor Vs. secondary actor
  - Primary
    - The first or main Actor who uses the system
    - The main Actor who benefits from the system
    - e.g. Customer, Patient, Doctor
  - Secondary Actor
    - The Actor who derives indirect benefits from or uses the system
    - e.g. Branch manager

Depends on Perspective

# Use Case

- System functionality to complete a process
- Describe **what** a system does

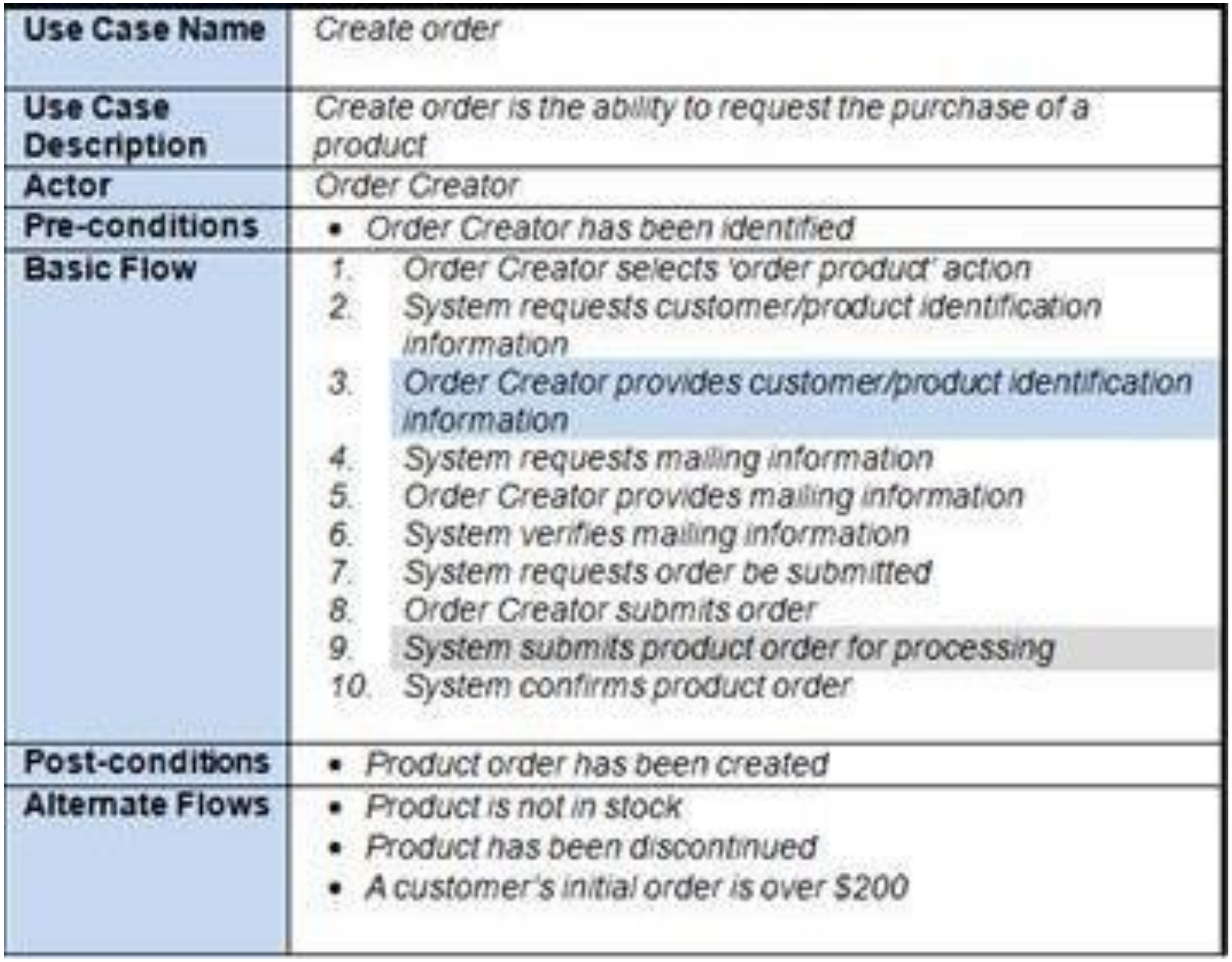
# Use Case Diagram Example



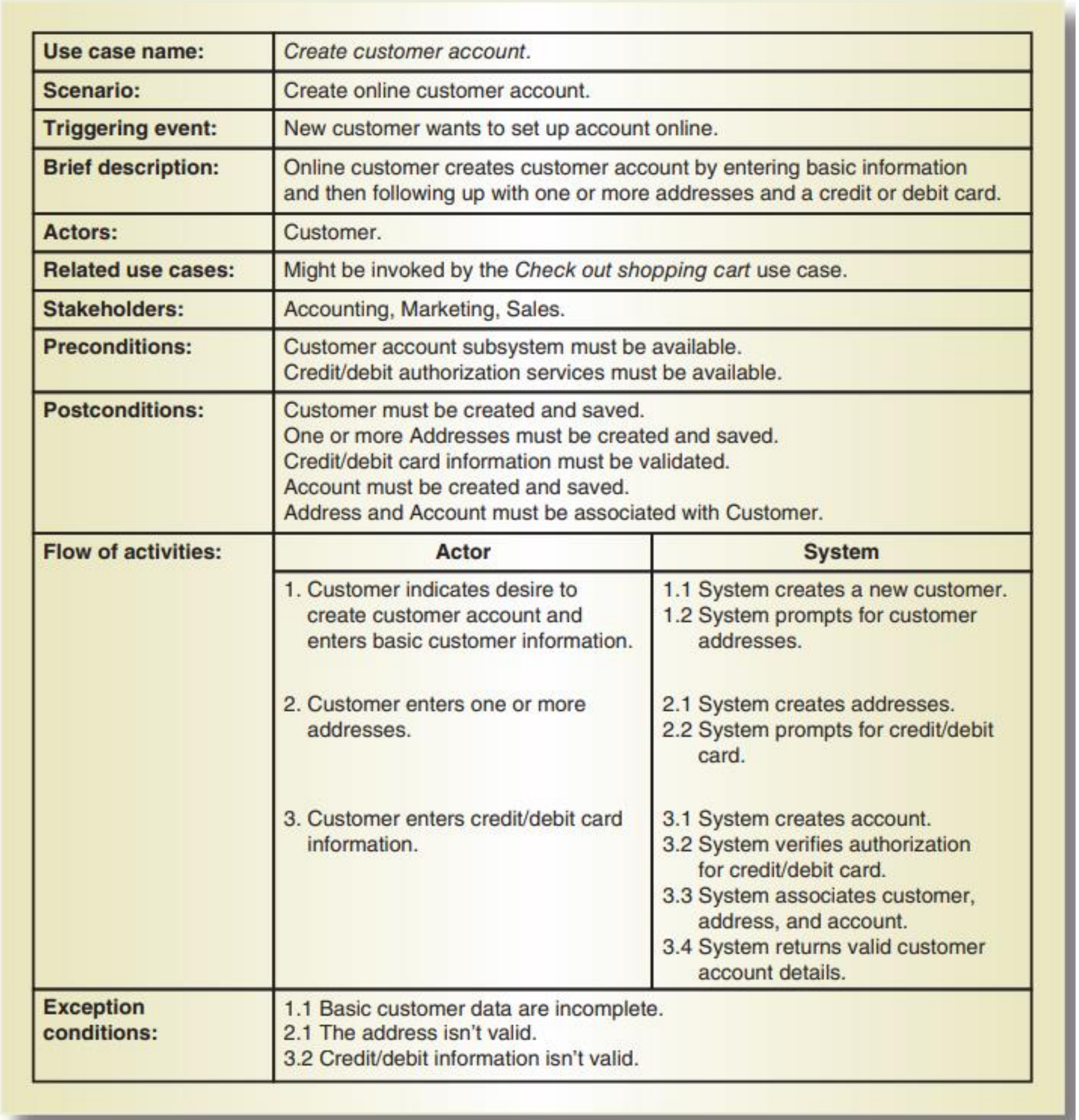
# Narratives

Narration is a textual representation of the course of events encountered when an actor is interacting with the system

# Narratives



| | |
|---|---|
| **Use Case Name** | *Create order* |
| **Use Case Description** | *Create order is the ability to request the purchase of a product* |
| **Actor** | *Order Creator* |
| **Pre-conditions** | • *Order Creator has been identified* |
| **Basic Flow** | 1. *Order Creator selects 'order product' action*<br>2. *System requests customer/product identification information*<br>3. *Order Creator provides customer/product identification information*<br>4. *System requests mailing information*<br>5. *Order Creator provides mailing information*<br>6. *System verifies mailing information*<br>7. *System requests order be submitted*<br>8. *Order Creator submits order*<br>9. *System submits product order for processing*<br>10. *System confirms product order* |
| **Post-conditions** | • *Product order has been created* |
| **Alternate Flows** | • *Product is not in stock*<br>• *Product has been discontinued*<br>• *A customer's initial order is over $200* |



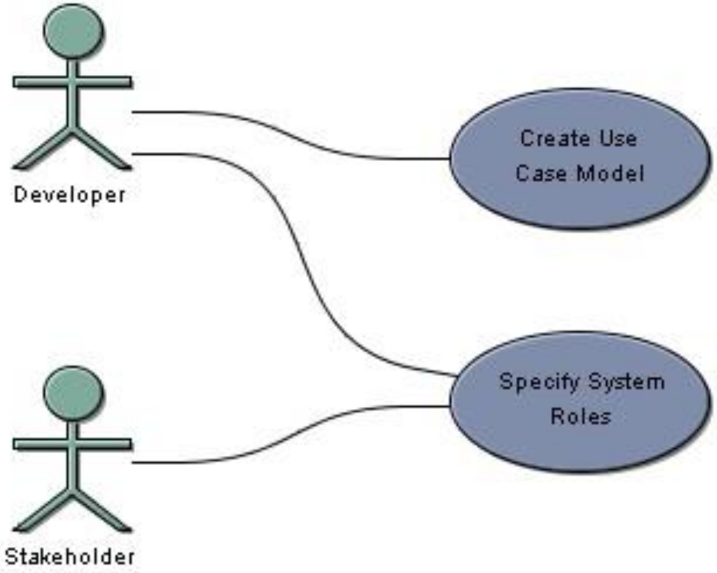
| | | |
|---|---|---|
| **Use case name:** | *Create customer account*. | |
| **Scenario:** | Create online customer account. | |
| **Triggering event:** | New customer wants to set up account online. | |
| **Brief description:** | Online customer creates customer account by entering basic information and then following up with one or more addresses and a credit or debit card. | |
| **Actors:** | Customer. | |
| **Related use cases:** | Might be invoked by the *Check out shopping cart* use case. | |
| **Stakeholders:** | Accounting, Marketing, Sales. | |
| **Preconditions:** | Customer account subsystem must be available.<br>Credit/debit authorization services must be available. | |
| **Postconditions:** | Customer must be created and saved.<br>One or more Addresses must be created and saved.<br>Credit/debit card information must be validated.<br>Account must be created and saved.<br>Address and Account must be associated with Customer. | |
| **Flow of activities:** | **Actor** | **System** |
| | 1. Customer indicates desire to create customer account and enters basic customer information. | 1.1 System creates a new customer.<br>1.2 System prompts for customer addresses. |
| | 2. Customer enters one or more addresses. | 2.1 System creates addresses.<br>2.2 System prompts for credit/debit card. |
| | 3. Customer enters credit/debit card information. | 3.1 System creates account.<br>3.2 System verifies authorization for credit/debit card.<br>3.3 System associates customer, address, and account.<br>3.4 System returns valid customer account details. |
| **Exception conditions:** | 1.1 Basic customer data are incomplete.<br>2.1 The address isn't valid.<br>3.2 Credit/debit information isn't valid. | |

# Relationships

- Association
- Extend
- Include
- Depends on
- Inheritance
- Generalization

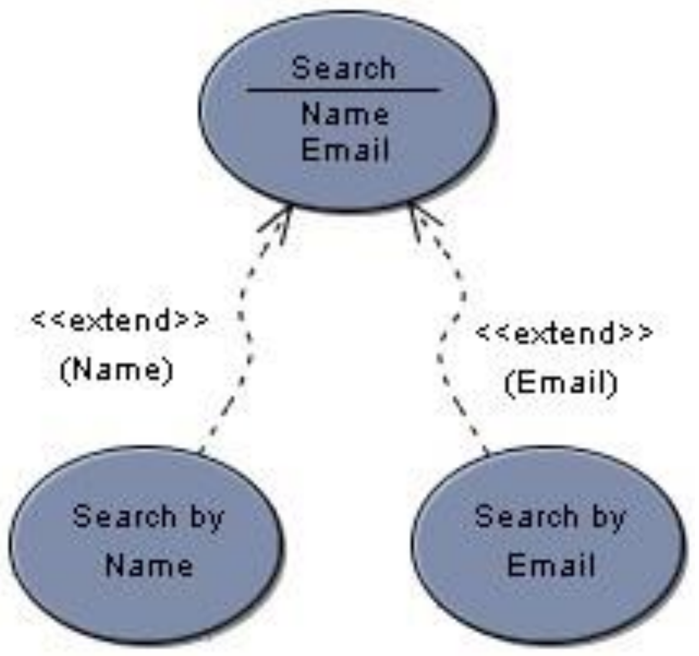
# Relationships

- Associations
  - Drawn between actors and use cases to show that an actor carries out a use case
  - Solid line connecting the actor and the use case

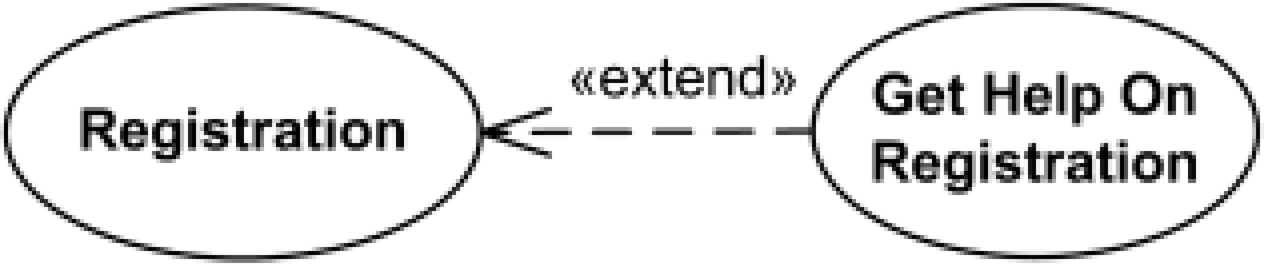


# Relationships

- Extend
  - Extends the functionality of the original use case
  - Arrow headed (either solid/dashed) line
  - Shows optional behavior

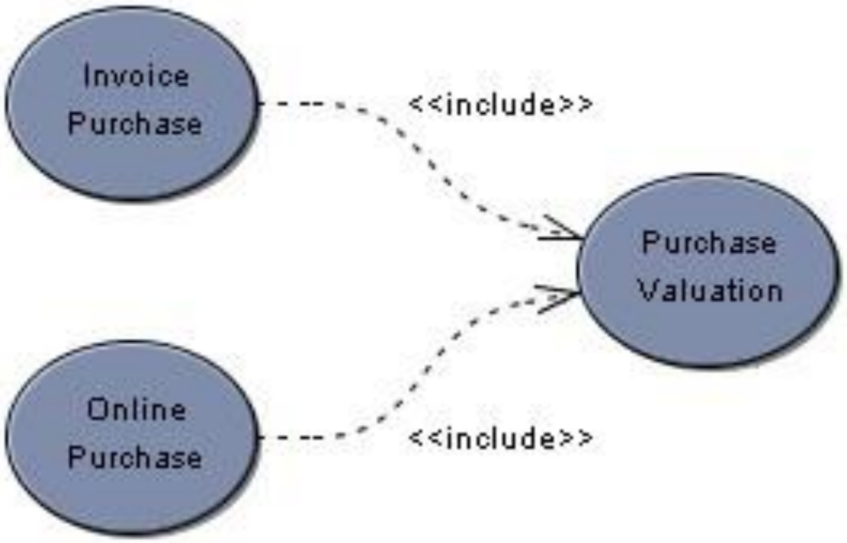


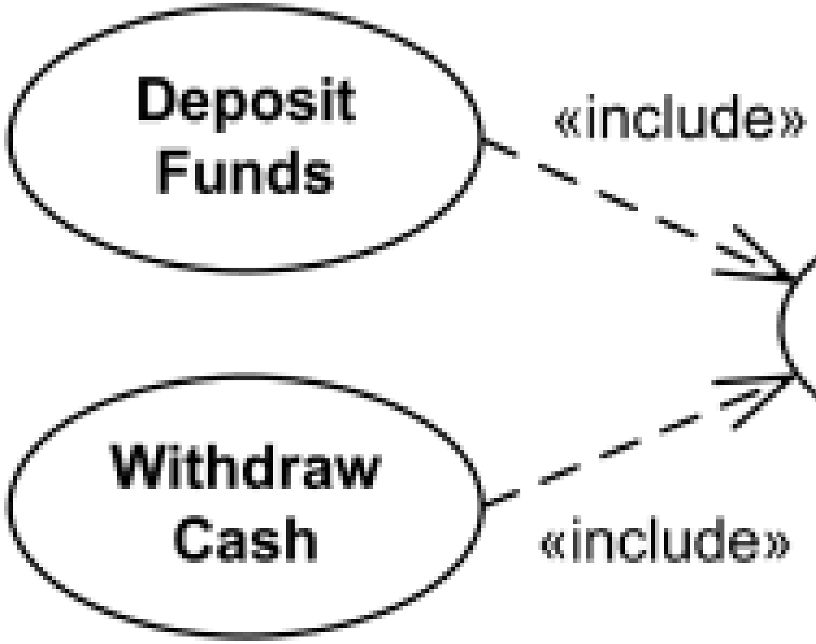
# Example



# Relationships

- Include
  - Depict the Relationship between an abstract use case and it included use case.
  - Arrow headed (either solid/dashed) line





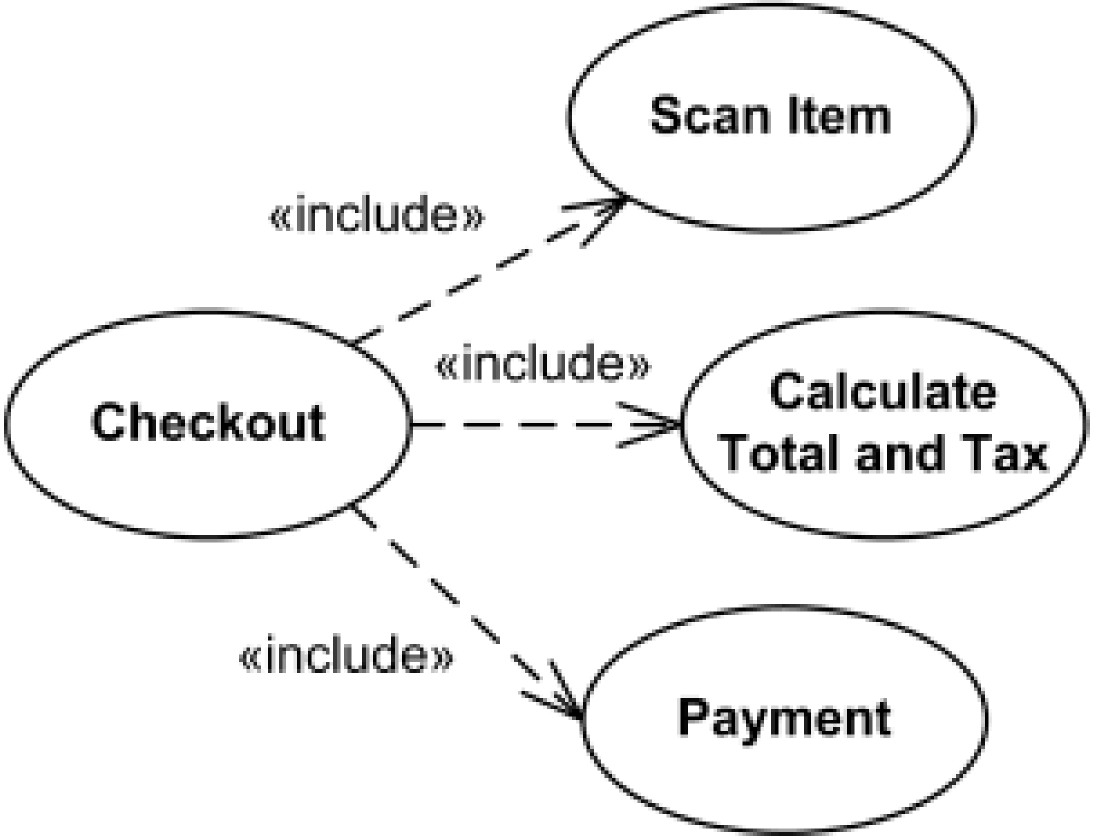
Deposit Funds

«include»

Customer Authentication

Withdraw Cash

«include»



Scan Item

«include»

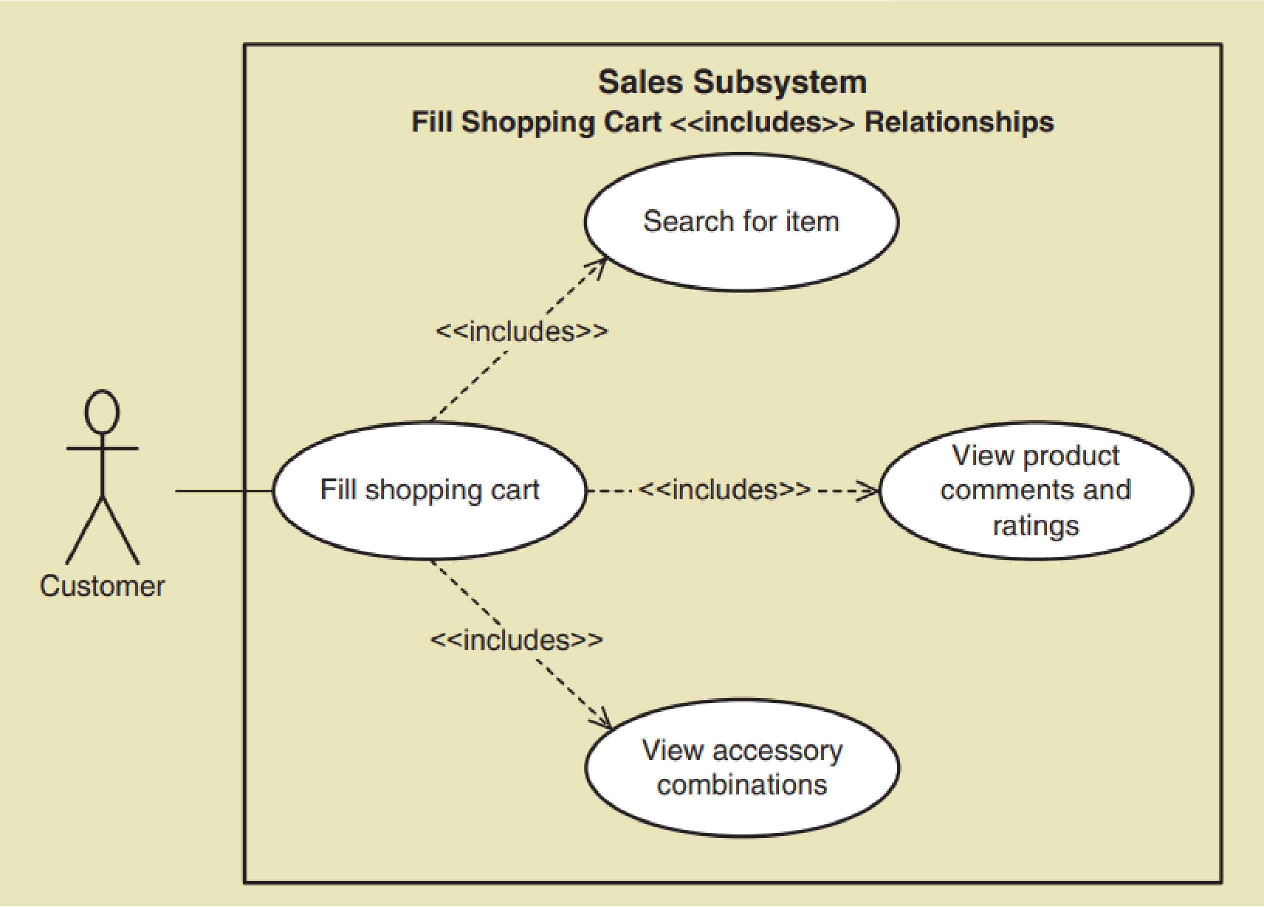
Checkout

«include»

Calculate Total and Tax

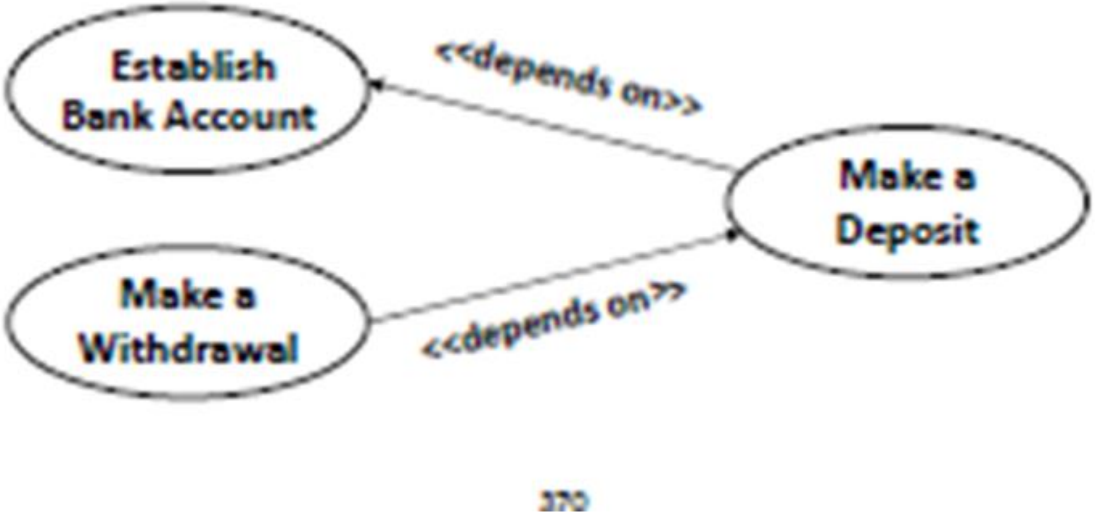
«include»

Payment


Sales Subsystem
Fill Shopping Cart <<includes>> Relationships
Search for item
<<includes>>
Customer
Fill shopping cart
<<includes>>
View product comments and ratings
<<includes>>
View accessory combinations

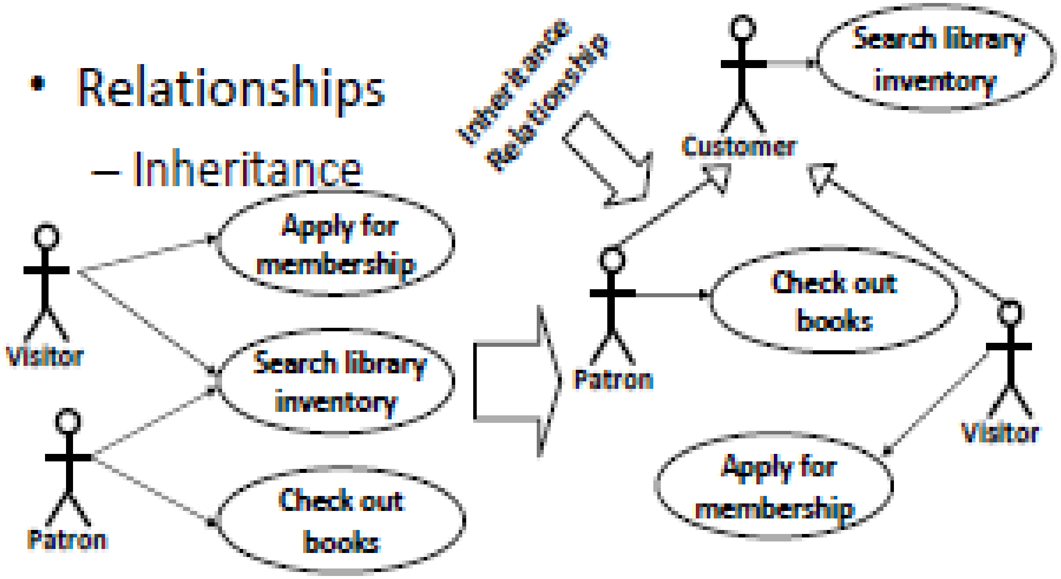
# Relationships

- Depends on
  - Indicating that one use case cannot be performed until another use case has been performed



# Relationships

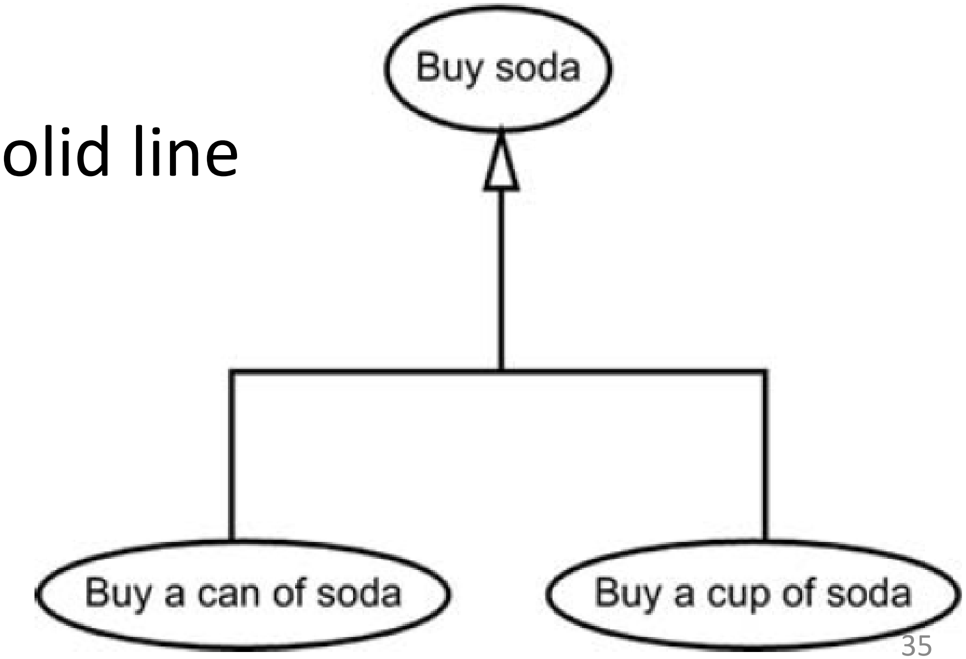
- Inheritance
  - Relationship between actors
  - Simplify the drawing when an abstract actor inherits the role of multiple actors



- Relationships
  - Inheritance

# Relationships

- Generalization
  - Relation between use cases
  - Relation between abstract use case and specialized use case
  - Use arrow headed solid line



# The Process of Modeling Use Cases

Step 1: Identify business external systems or users

- Actors

Step 2: Identify Business Requirements

- Use Cases

Step 3: Construct Use-Case Model

- Diagram

Step 4: Document Business Requirement

- Use-case Narratives

# END