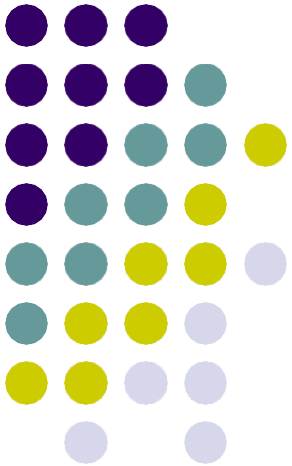
# **Course Code:** IS332
# Analysis and Design of Information Systems

By

Dr. Sherif Zahran

Information Systems Department
Faculty of Computers and Artificial Intelligence
Cairo University, Egypt



# Lecture Rules








# Course Code

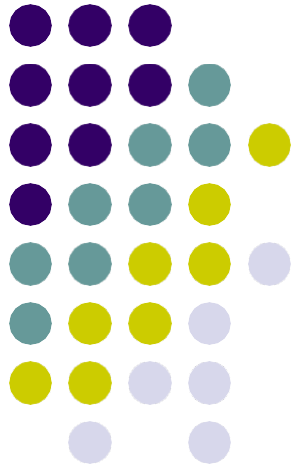
- **222301 FCI IS332 Analysis and Design of Information Systems**
- Code: **222301 FCI IS332**
- Published

# Use Cases

# Chapter 3

Systems Analysis and Design
in a Changing World 7th Ed

Satzinger, Jackson & Burd



# Chapter 3 Outline



- User Stories and Use Cases
- Use Cases and the **User Goal** Technique
- Use Cases and **Event Decomposition**
- Use Cases in the Ridgeline Mountain
- Outfitters Case



# Learning Objectives



- Explain why identifying user stories and use cases is the key to defining functional requirements
- Write user stories with acceptance criteria
- Describe the two techniques for identifying use cases
- Apply the user goal technique to identify use cases
- Apply the event decomposition technique to identify use cases
- Describe the notation and purpose for the use case diagram
- Draw use case diagrams by actor and by subsystem

# Overview



Chapter 2 provided an **overview of systems analysis activities, functional** and **non-functional requirements**, modelling, and **information gathering techniques**



This chapter focuses on **identifying** and **modelling** the key aspect of functional requirements– use cases



In the RMO Tradeshow System from Chapter 1, some use cases are *Look up supplier*, *Enter/update product information*, *Enter/Update contact information*



In this chapter's opening case Waiters on Call, examples of use cases are *Record an order*, *Record delivery*, *Update an order*, *Sign in driver*, *Reconcile driver receipts, Produce end of day deposit slip,* and *Produce weekly sales reports*

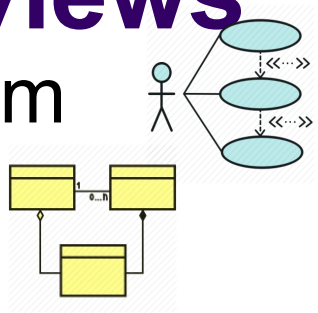
# System Views

| *Major Area* | *View* | *Diagrams* |
|---|---|---|
| Structural | Static | **Class** |
| | Use Case | **Use Case** |
| | Implementation/Deployment | Component |
| | | Deployment |
| Dynamic | State Machine | **Statechart** |
| | Activity | **Activity** |
| | Interaction | **Sequence** |
| | | Collaboration |

[Rumbaugh,Jacobson,Booch1999]

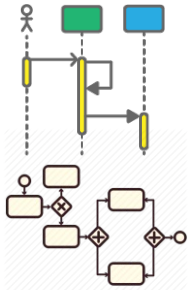
# UML: Structural vs. Behavioural views

- Use Case diagram
- Class diagram

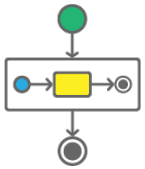


Structural: static systems aspects

- Sequence diagram
- Activity diagram
- State machine diagram



Activity and sequence diagrams extend **use cases**



Dynamic: behavior of different aspects of the system

State machine captures the behavior of the data objects (**class** diagram)

# User Stories

- A **user story** is usually one short sentence in the everyday language of the end user that states what a user does as part of his or her work.
  - They document the **functional requirements** quickly and less formally than traditional requirements modeling by focusing on *who*, *what*, and *why* for each function.
- The standard template for a user story looks like this:
  - **"As a <*role played*>, I want to <*goal or desire*> so that <*reason or benefit*>."**
- **Acceptance criteria**. These indicate the features that must be present for the user to be satisfied with the resulting implementation.
  - They focus on functionality, not on features or user-interface design.

# User Stories with acceptance criterea



**User Story**

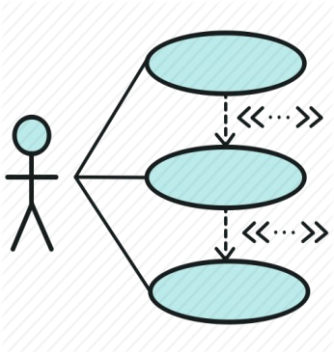
*As a shipping clerk, I want to ship an order as accurately as possible as soon as the order details are available.*

**Acceptance Criteria:**

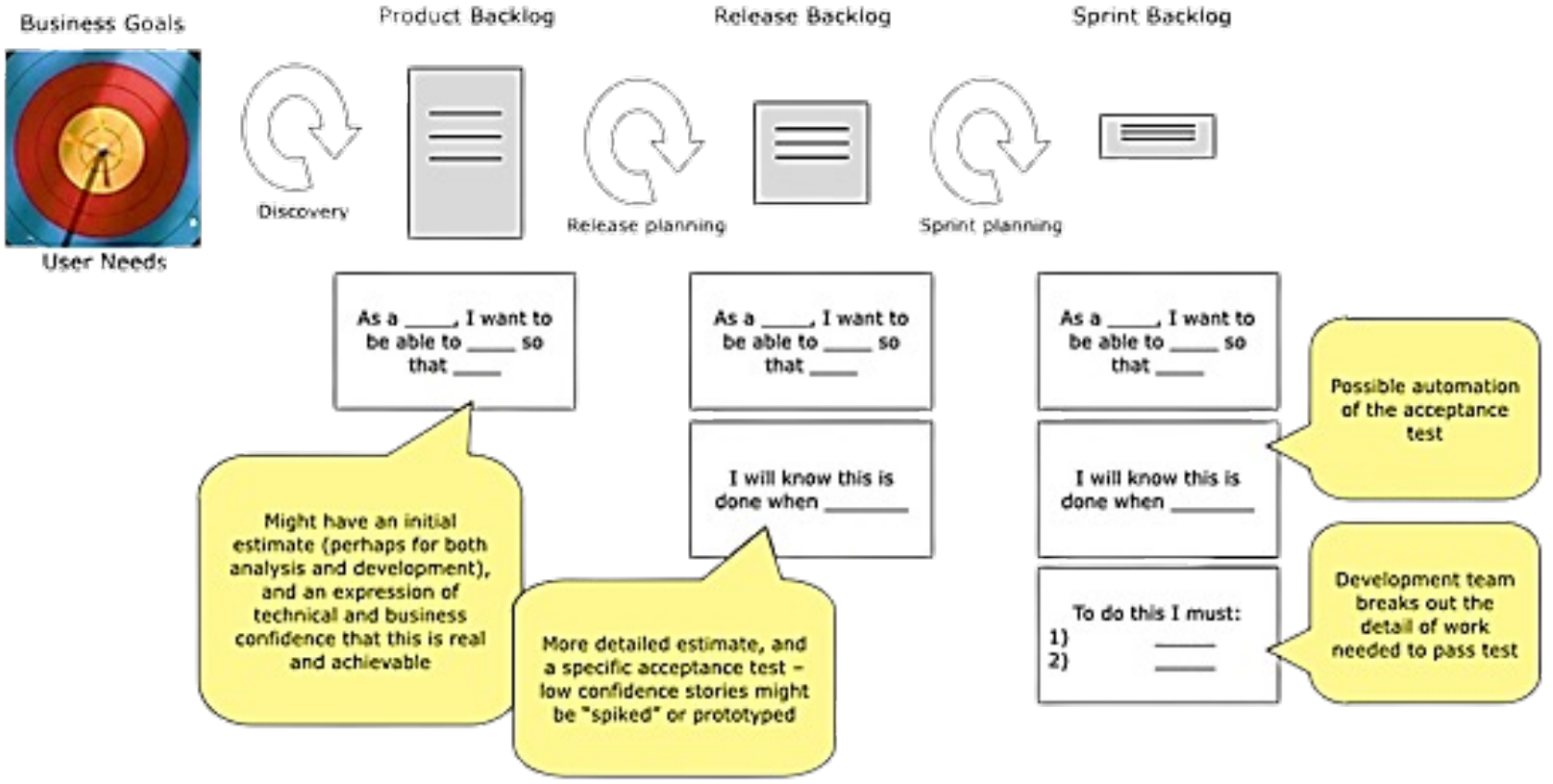
1. *Available order details must pop up on the screen when available.*
2. *Portable display and scan device would cut time in half.*
3. *Sort the items by bin location.*
4. *Indicate number of items in stock for each item and mark backorder for those not available.*
5. *Recommend shipper based on weight, size, and location.*
6. *Print out shipping label for selected shipper.*

# Use Cases

- Use case— an **activity** that the system **performs**, usually in **response** to a **request** by a **user**
- Use cases define **functional requirements**
- Analysts **decompose** the system into a set of **use cases** (functional decomposition)
- Two techniques for Identifying use cases
  1. User goal technique
  2. Event decomposition technique
- Name each use case using ***Verb-Noun***



# User Story Lifecycle



# User Story Lifecycle



# How to write an agile user story



**1** Define your **end user**
Who will be using your product?

As a parent,

**2** Specify what **they want**
What solution are you offering?

I want to check on my sleeping baby without entering his room,

**3** Describe **the benefit**
What will your user gain from using your product?

so that I know he is safe without disturbing him.

**4** Add **acceptance criteria**
What determines this story as 'done'?

e.g. Alert to be sent to the registered smartphone if problem is detected.

## Top tips to create a good user story

- Develop a persona profile to visualise your end user
- Always write from your end user's perspective
- Avoid adding technical details too early
- Try not to add too many acceptance criteria
- Keep stories brief, breaking them up if you need to
- Make sure they meet your "definition of done"



**I** ndependent

**N** egotiable

**V** aluable

**E** stimable

**S** mall (Sized appropriately)

**T** estable

# User Story Sample

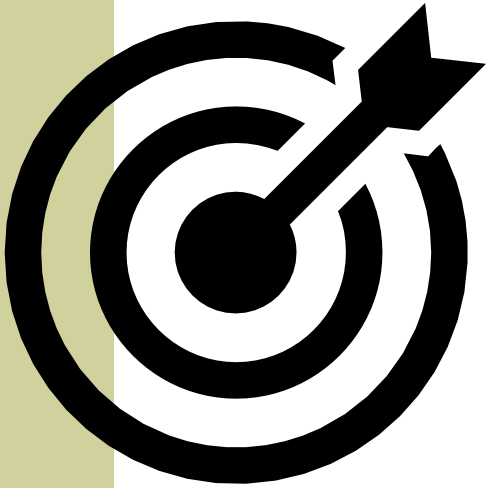


**User story:** *As a user, I want to be able to register online, so that I can start shopping online.*

**Acceptance criteria:**

- [ ] User can only submit a form by filling in all required fields
- [ ] The email user provided must not be a free email
- [ ] Submission from same IP can only be made three times within 30 minutes
- [ ] User can only submit a form by filling in all required fields
- [ ] User will receive a notification email after successfully registration

# User Goal Technique



# User Goal Technique

- This technique is the most common in the industry
- Simple and effective
- **Identify** all the potential categories of **users** of the system
- **Interview** and ask them to **describe the tasks** the computer can help them with
- Probe further to **refine the tasks** into specific **user goals**:
  - "I need to *Ship items"*
  - "*Track a shipment*"
  - "*Create a return*"

# User Goal Technique

## Some RMO CSMS Users and Goals



| 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 |



# User Goal Technique: Specific Steps

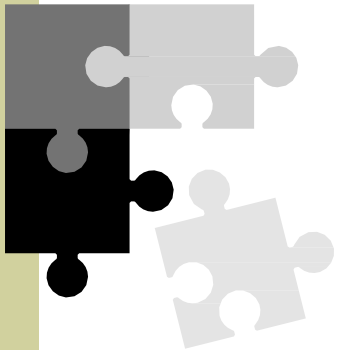
1. Identify all the **potential users** for the new system
2. **Classify** the potential users in terms of their **functional role** (e.g., shipping, marketing, sales)
3. Further **classify** potential users by **organizational level** (e.g., operational, management, executive)
4. For each type of user, **interview** them to **find** a list of **specific goals** they will have when using the new system (**current goals and innovative functions** to add value)



# User Goal Technique: Specific Steps

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

# Event Decomposition Technique



# Event Decomposition Technique



More **Comprehensive** and Complete Technique

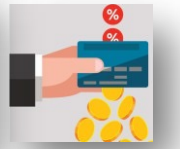
Identify the **events** that occur to which the system must **respond**

For **each event**, name a **use case** (verb-noun) that describes **what the system does** when the event occurs

**Event**– something that occurs at a **specific time** and **place**, can be described, and should be remembered by the system

# Events and Use Cases





*External events occur in the environment*



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



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

*Temporal events occur inside the system*

Charge account processing system



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



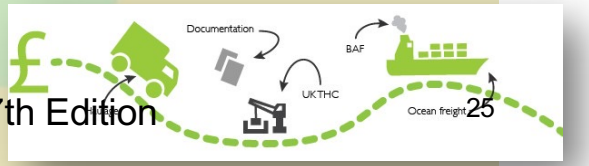
"time to send late notices," so use case is *Send late notices*



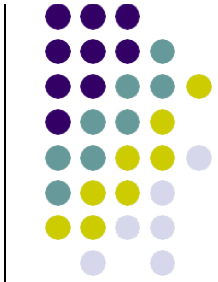
"time to produce end-of-week summary reports," so use case is *Produce summary reports*



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



# Types of Events





- External Event
  - an **event** that occurs **outside** the system, usually **initiated** by an **external** agent or actor
- Temporal Event
  - an **event** that occurs as a **result** of **reaching** a point in **time**



- State Event (also *internal event*)
  - an event that occurs when something happens inside the system that triggers some process
    - Usually as a consequence of an external event
    - E.g., reorder point is reached for inventory item

# External Event Checklist



External **agent** or **actor** wants something resulting in a **transaction**

- Customer buys a product

External **agent** or **actor** wants some **information**

- Customer wants to know product details

**External data changed** and needs to be updated

- Customer has new address and phone

**Management** wants some **information**

- Sales manager wants update on production plans

# Temporal Event Checklist



## **Internal outputs** needed at **points in time**

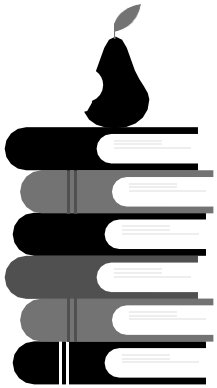
- **Management reports** (summary or exception)
- **Operational reports** (detailed transactions)
- **Internal statements** and documents (including payroll)

## **External outputs** needed at points of time

- Statements
- Status reports
- Bills
- Reminders

# nich one to use: User goal OR Event decomposition? Why?

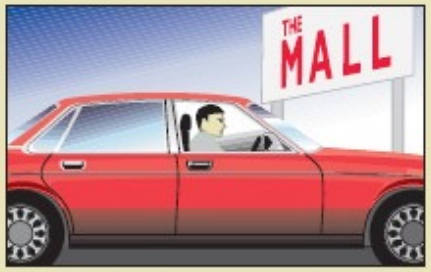
# Guidelines to identify events/use cases



# Finding the actual event that affects the system



Customer thinks about getting a new shirt



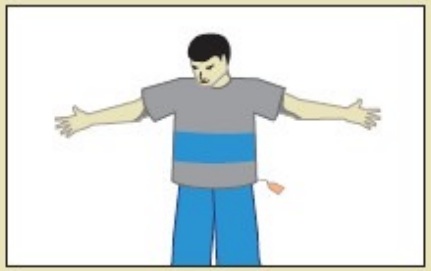

Customer drives to the mall




Customer tries on a shirt at Sears




Customer goes to Walmart



Customer tries on a shirt at Walmart



Customer buys a shirt

*(the event that directly affects the system!)*

# Tracing a sequence of transactions resulting in many events




Customer requests a catalog



Customer wants to check item availability



Customer places an order



Customer changes or cancels an order



Customer wants to check order status




Customer updates account information

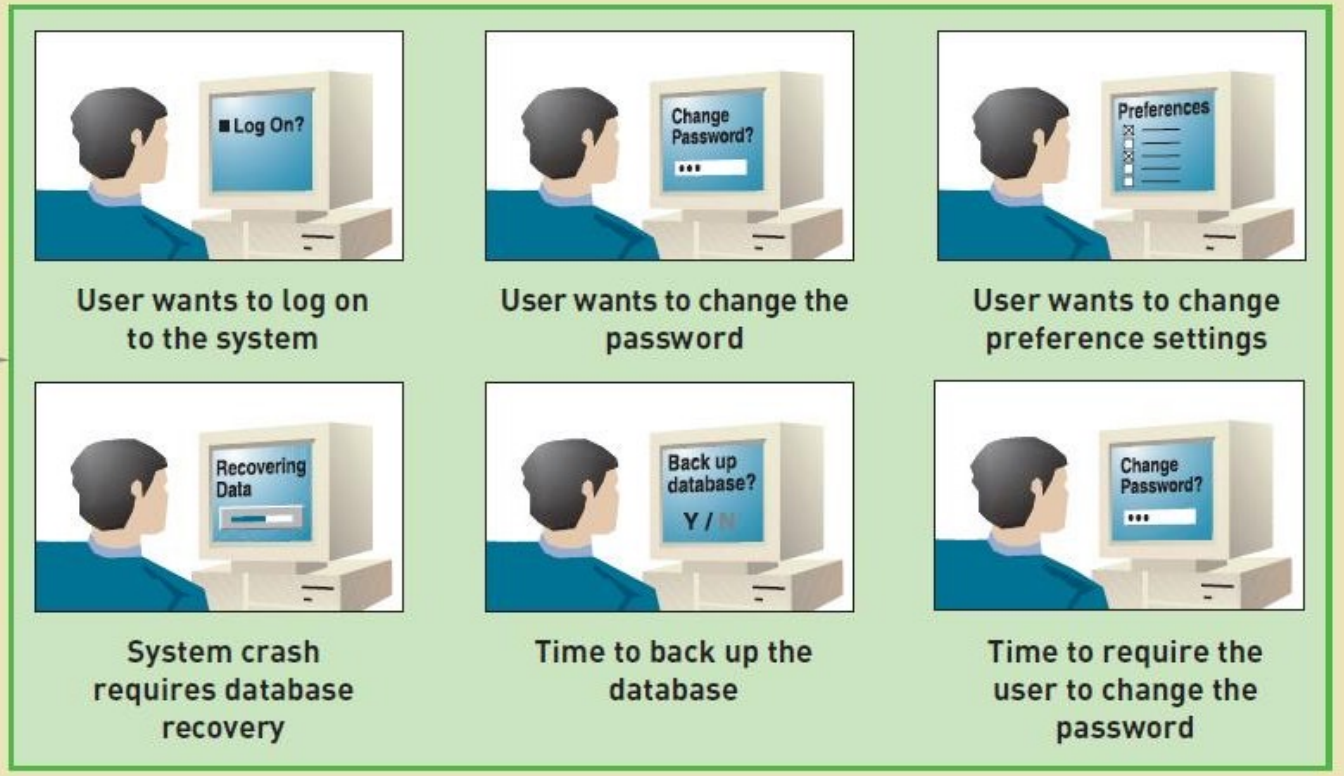



Customer returns the item

# Perfect Technology Assumption

- Don't worry about functions built into system because of limits in technology and people. Wait until design.

*Don't worry much about these until you are considering design issues*



# Event Decomposition Technique: Specific Steps



1. Consider the **external events** in the system environment that require a response from the system by using the checklist shown below

External events to look for include:
√ External agent wants something resulting in a transaction
√ External agent wants some information
√ Data changed and needs to be updated
√ Management wants some information

2. For each **external event**, **identify** and **name** the **use case** that the system requires

# Event Decomposition Technique: Specific Steps



3. Consider the **temporal events** that require a **response** from the system by using the checklist shown below

Temporal events to look for include:
√ Internal outputs needed
  √ Management reports (summary or exception)
  √ Operational reports (detailed transactions)
  √ Internal statements and documents (including payroll)
√ External outputs needed
  √ Statements, status reports, bills, reminders

# Event Decomposition Technique: Specific Steps



4. For each **temporal event**
   - **identify** and **name** the **use case** that the system requires
   - establish the point of **time** that will **trigger** the use case
5. Consider the **state events** that the system might respond to, particularly if it is a real-time system in which devices or internal state changes trigger use cases.

# Event Decomposition Technique: Specific Steps



6. For each state event, **identify** and **name** the use case that the system requires and then **define the state change**
7. When events and use cases are defined, check to see if they are required by using the **perfect technology assumption**.
   - Do not include events that involve such **system controls** as login, logout, change password, and backup or restore the database
   - These are put in later.

# Event Decomposition Technique: Benefits



- **Events are broader** than user goal:
  - Capture temporal and state events
- Help decompose at the **right level of analysis**: an elementary business process (EBP)
  - **EBP** is a *fundamental* business process *performed* by one person, in one place, in *response* to a business event
- Uses **perfect technology assumption** to make sure functions that support the users work are identified and *not additional functions* for security and system controls

# Use Cases and Brief Use Case Descriptions



- Brief use case description is often a one sentence description showing the main steps in a use case

| Use case | Brief use case description |
|---|---|
| *Create customer account* | User/actor enters new customer account data, and the system assigns account number, creates a customer record, and creates an account record. |
| *Look up customer* | User/actor enters customer account number, and the system retrieves and displays customer and account data. |
| *Process account adjustment* | User/actor enters order number, and the system retrieves customer and order data; actor enters adjustment amount, and the system creates a transaction record for the adjustment. |

# RMO CSMS Project Use Cases

| CSMS sales subsystem | |
|---|---|
| **Use cases** | **Users/actors** |
| Search for item | Customer, customer service representative, store sales representative |
| View product comments and ratings | Customer, customer service representative, store sales representative |
| View accessory combinations | Customer, customer service representative, store sales representative |
| Fill shopping cart | Customer |
| Empty shopping cart | Customer |
| Check out shopping cart | Customer |
| Fill reserve cart | Customer |
| Empty reserve cart | Customer |
| Convert reserve cart | Customer |
| Create phone sale | Customer service representative |
| Create store sale | Store sales representative |

| CSMS order fulfillment subsystem | |
|---|---|
| **Use cases** | **Users/actors** |
| Ship items | Shipping |
| Manage shippers | Shipping |
| Create backorder | Shipping |
| Create item return | Shipping, customer |
| Look up order status | Shipping, customer, management |
| Track shipment | Shipping, customer, marketing |
| Rate and comment on product | Customer |
| Provide suggestion | Customer |
| Review suggestions | Management |
| Ship items | Shipping |
| Manage shippers | Shipping |

# RMO CSMS Project Use Cases

| CSMS Customer account subsystem | |
|---|---|
| **Use cases** | **Users/actors** |
| Create/update customer account | Customer, customer servic sales representative |
| Process account adjustment | Management |
| Send message | Customer |
| Browse messages | Customer |
| Request friend linkup | Customer |
| Reply to linkup request | Customer |
| Send/receive points | Customer |
| View "mountain bucks" | Customer |
| Transfer "mountain bucks" | Customer |

# RMO CSMS Project Use Cases

# RMO CSMS Project Use Cases

| CSMS marketing subsystem | |
|---|---|
| **Use cases** | **Users/actors** |
| Add/update product information | Merchandising, marketing |
| Add/update promotion | Marketing |
| Add/update accessory package | Merchandising |
| Add/update business partner link | Marketing |

| CSMS reporting subsystem | |
|---|---|
| **Use cases** | **Users/actors** |
| Produce daily transaction summary report | Management |
| Produce sales history report | Management, marketing |
| Produce sales trends report | Marketing |
| Produce customer usage report | Marketing |
| Produce shipment history report | Management, shipping |
| Produce promotion impact report | Marketing |
| Produce business partner activity report | Management, marketing |

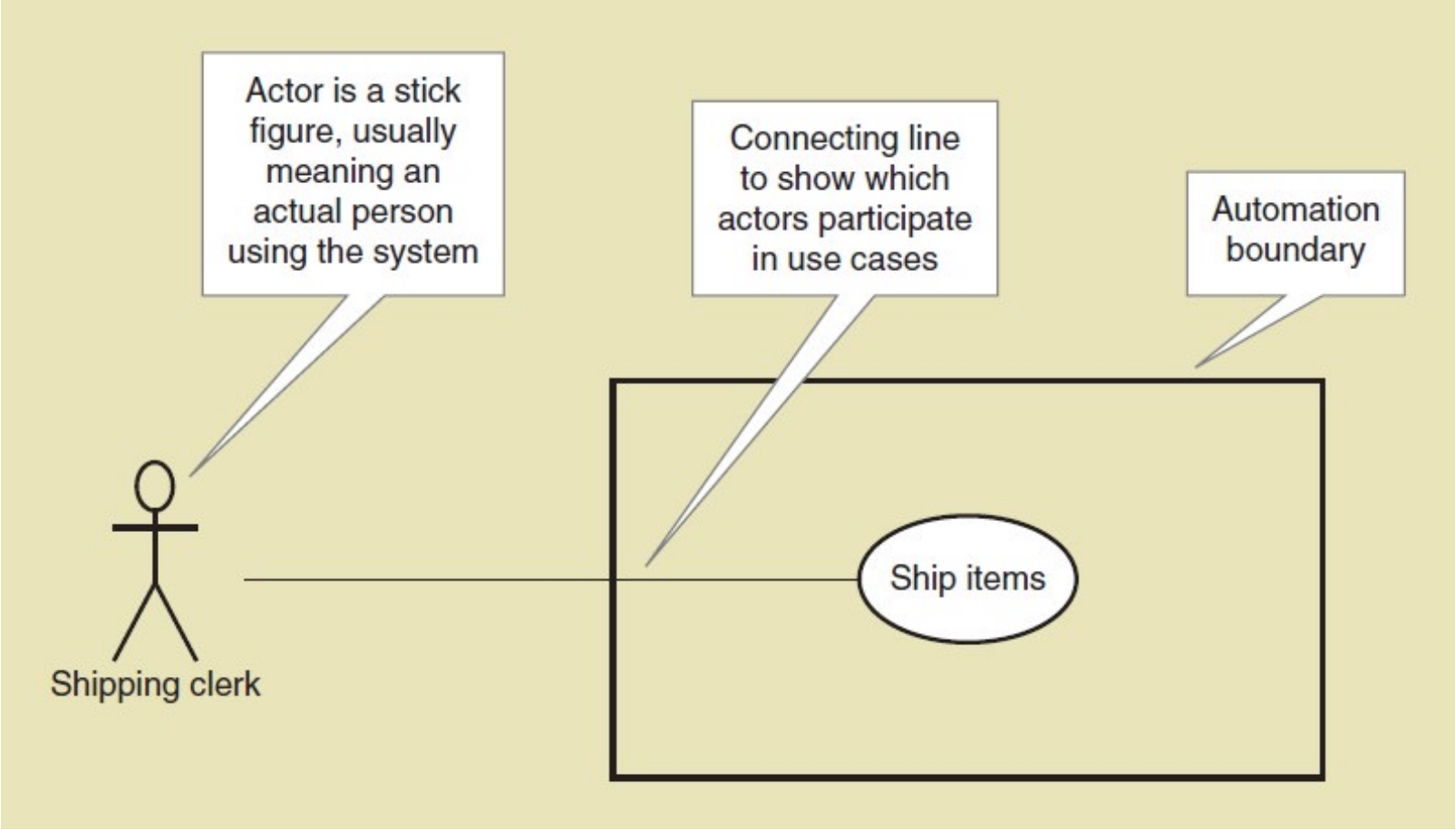
# UML Use Case Diagram

# Use Case Diagrams



- **Use case diagram**— a UML model used to **graphically** show uses cases and their relationships to actors
  - Recall UML is Unified Modeling Language, the standard for diagrams and terminology for developing information systems
- **Actor** is the UML name for a specific **ROLE**, **NOT** a person.
  - Actor could be a **human** or another **system**/**sub-system**
- **Automation boundary**— the boundary between the computerized portion of the application and the actors who operate the application

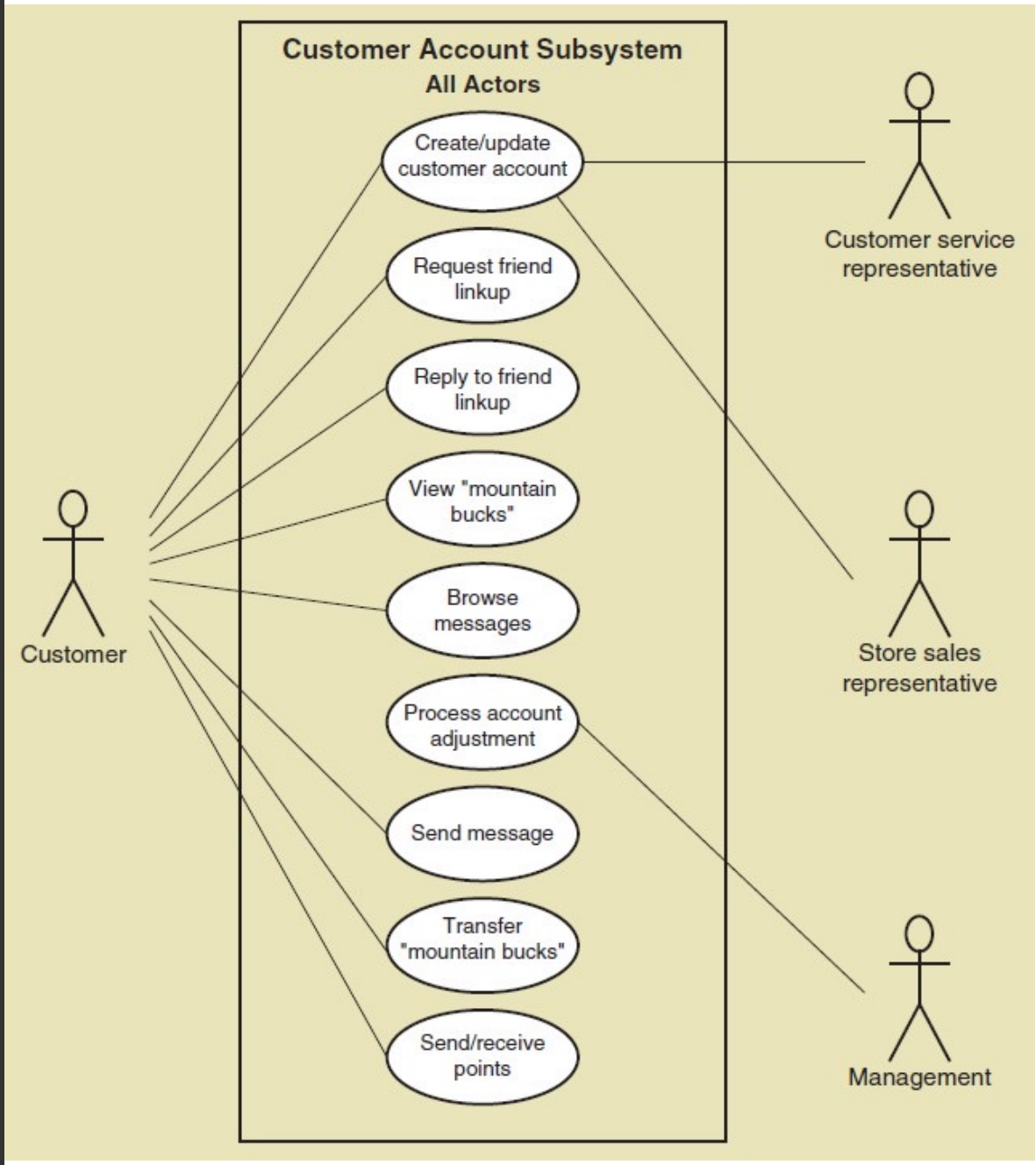
# Use Case Diagrams Symbols



# Use Case Diagrams

Draw for each **subsystem**

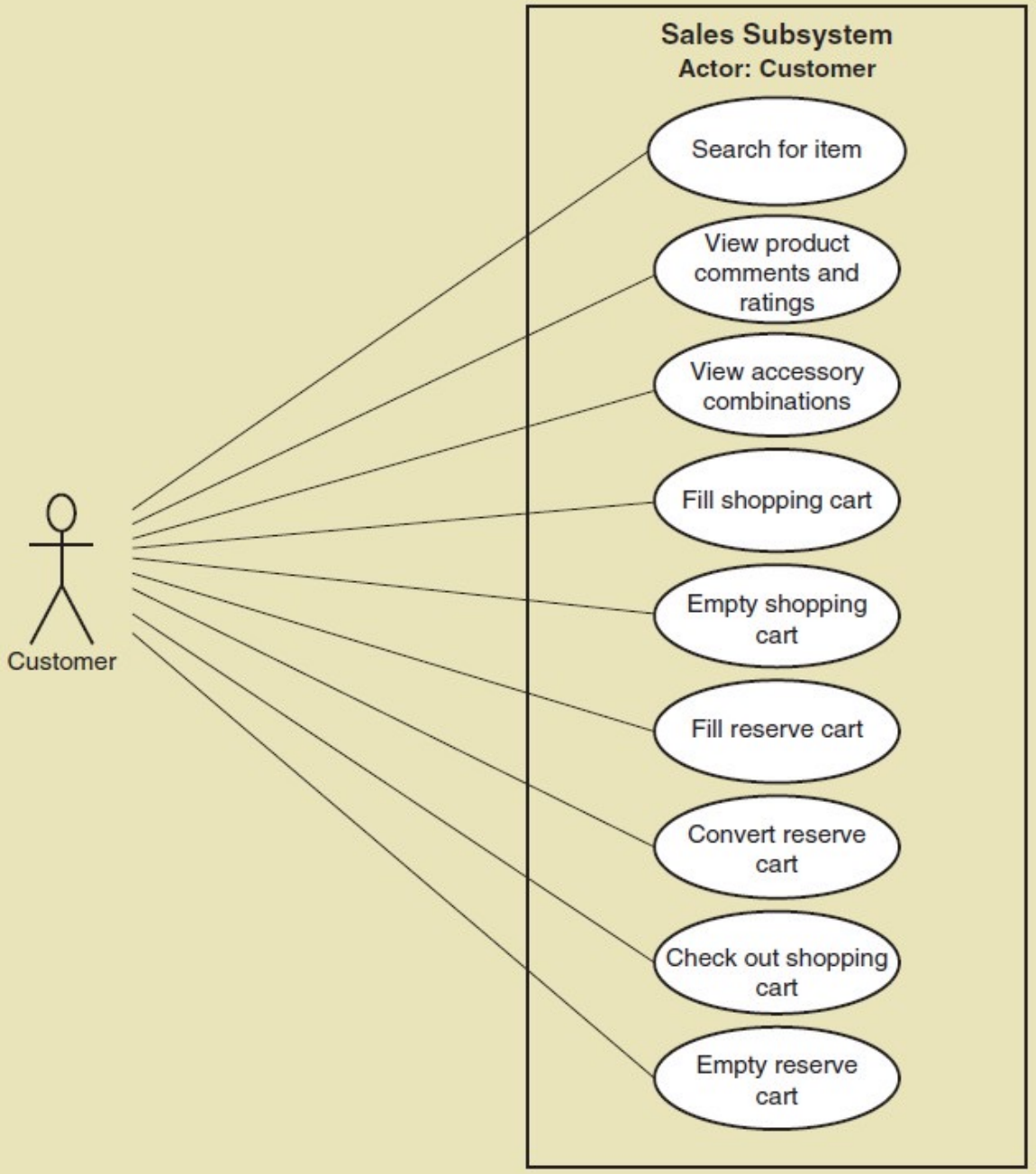
*Customer Account subsystem for RMO, showing all actors*



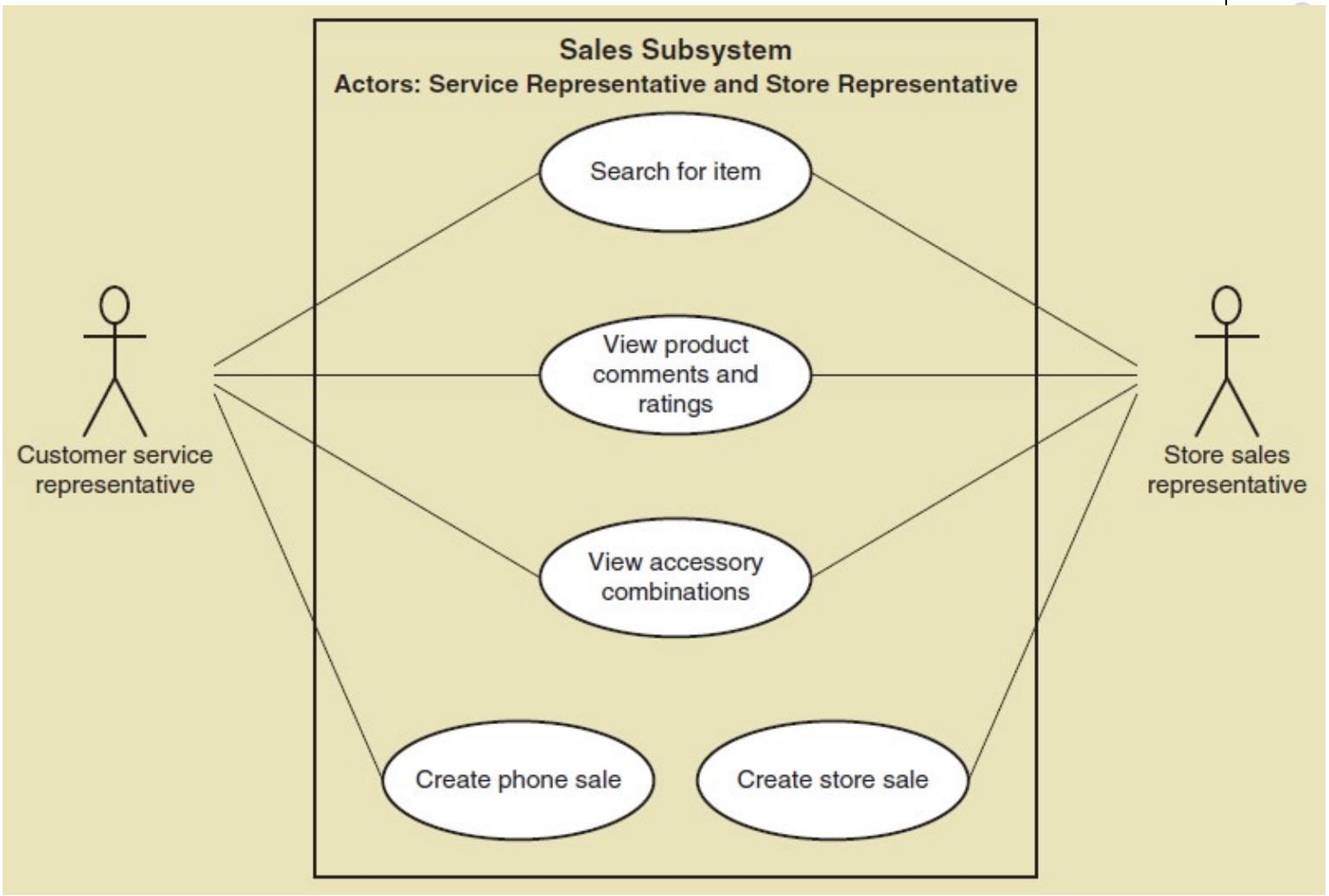
# Use Case Diagrams

Draw for actor, such as **customer**

*Sales subsystem*

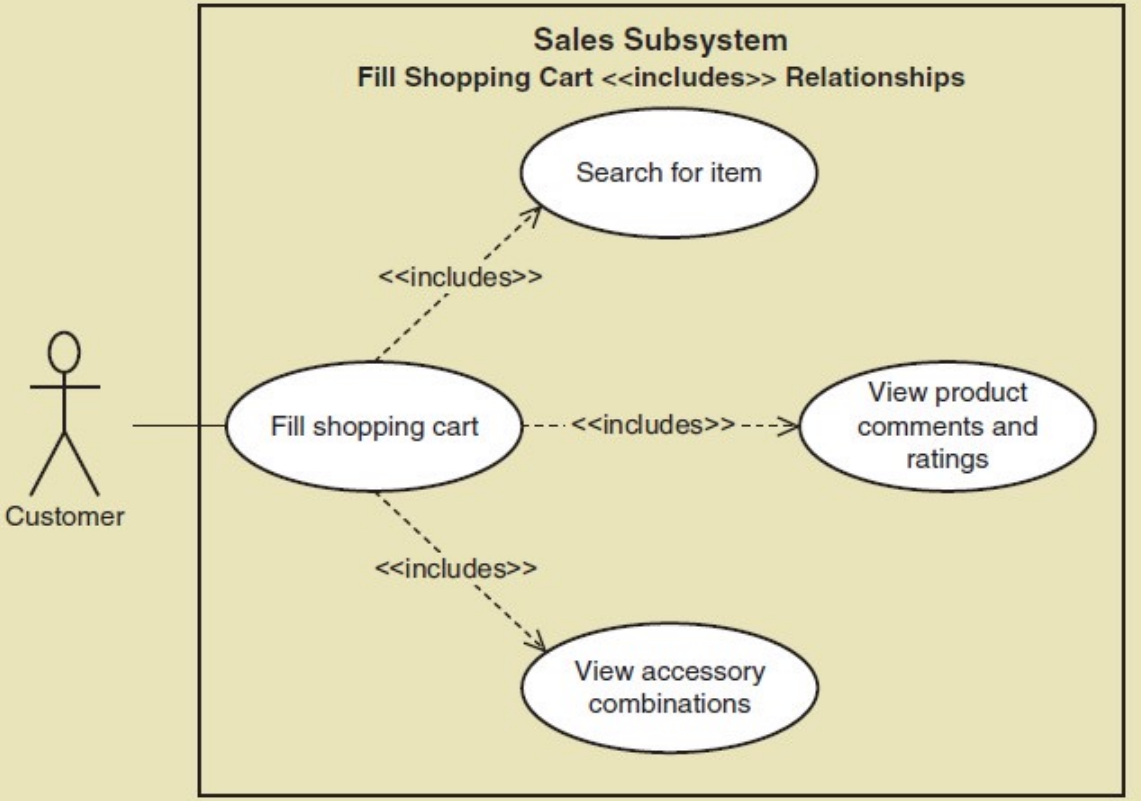


# Use Case Diagrams
## Draw for internal RMO actors



# Use Case Diagrams The <<Includes>> relationship

- A relationship between use cases where one use case is stereotypically **included** within the other use case— like a called subroutine. Arrow points to subroutine



# Use Case Diagrams: Steps



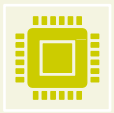
**Identify**

Identify all the **stakeholders** and **users** who would benefit by seeing a use case diagram



**Determine**

Determine what each stakeholder or user needs to review in a use case diagram: each subsystem, for each type of user, for use cases that are of interest



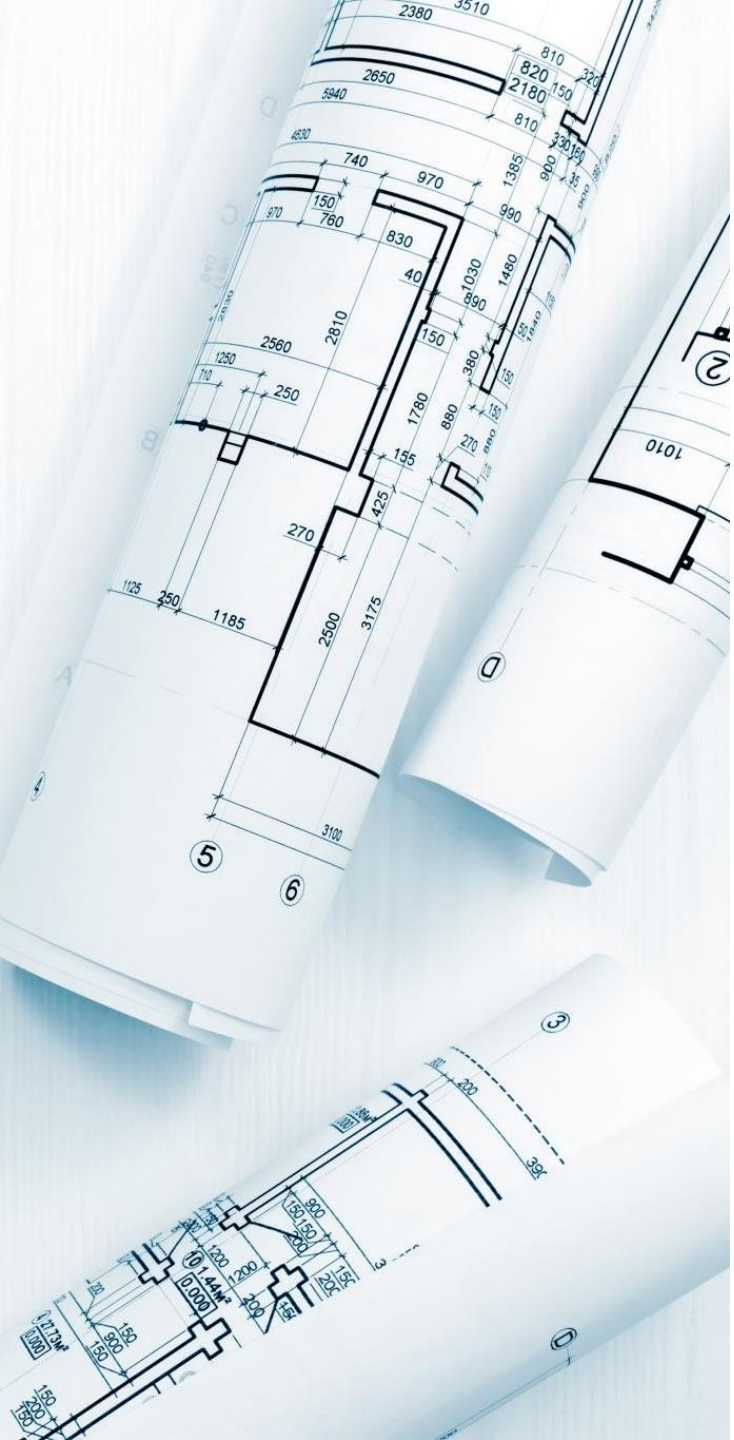
**Select**

For each potential communication need, select the use cases and actors to show and draw the use case diagram. There are many **software packages** that can be used to draw use case diagrams



**Name**

Carefully **name** each use case diagram and then **note how and when** the diagram should be used to review use cases with stakeholders and users



# Summary

- Modelling functional requirements:
  - User stories or use cases
- Use cases can be identified by:
  - User goal technique
  - Event decomposition technique
- User goal technique:
  - Types of system end users (actors)
  - Each user specify user goal
- Event decomposition technique:
  - Event types: external, temporal, state (internal)
  - Brief use case descriptions are written for use cases
- A variety of use case diagrams are draw depending on the presentation needs of the analysis



# **Course Code:** IS332
# **Course:** Analysis and Design of Information Systems



## Thank you



s.zahran@fci-cu.edu.eg