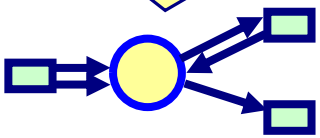
Example case: school

- Student
  - Full time
  - Part time
  - International
  - Scholarship
  - New student
  - Non-first time student
- Staff
- Parent
  - Married parents
  - Single parent
- Teacher
  - Full time
  - Part time

IDENTIFY GOALS:
USE CASE DIAGRAM

Agile requirements

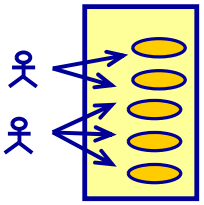
Understand the big picture



Context diagram

Personas

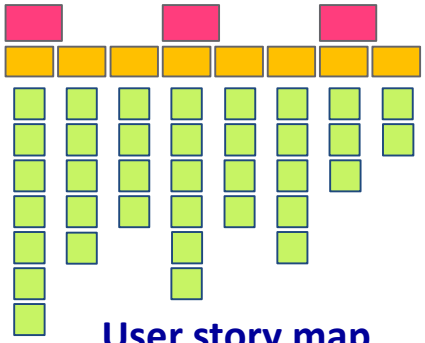
Identify the users



Use case diagram

Understand the goals of the users

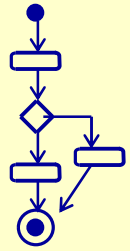
Find and understand the user stories



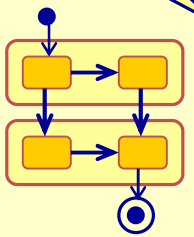
User story map

- Split user stories
- Write acceptance criteria

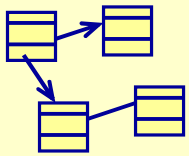
Communicate user stories



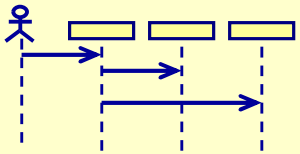
activity diagram



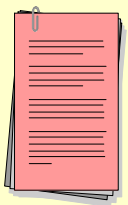
state machine diagram



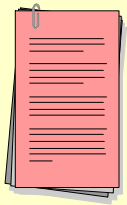
class diagram



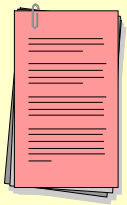
sequence diagram



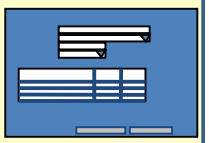
business rules



Non-Functional requirements



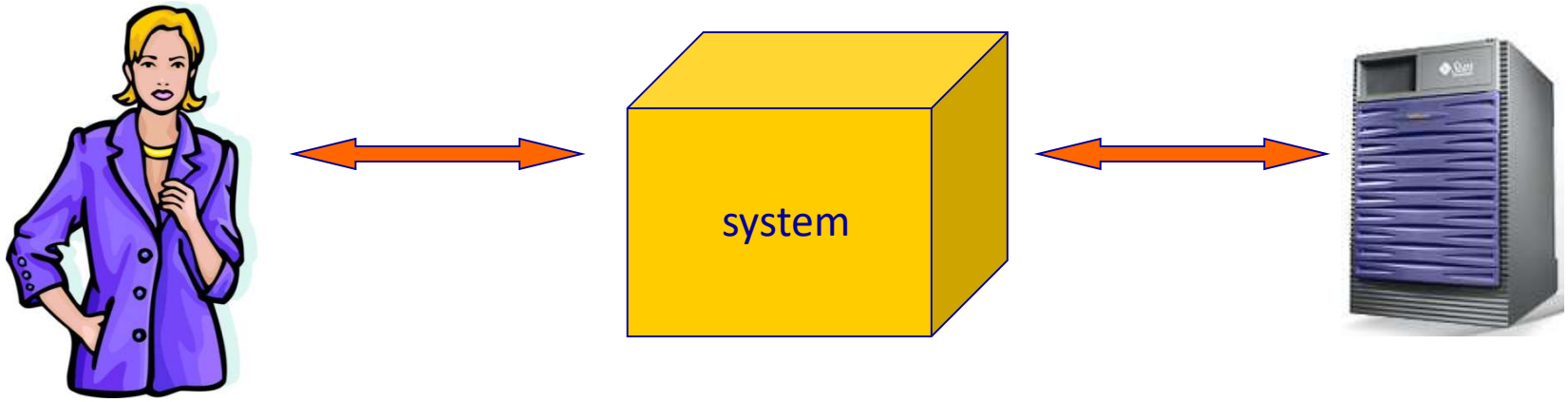
user scenario's



UI screens

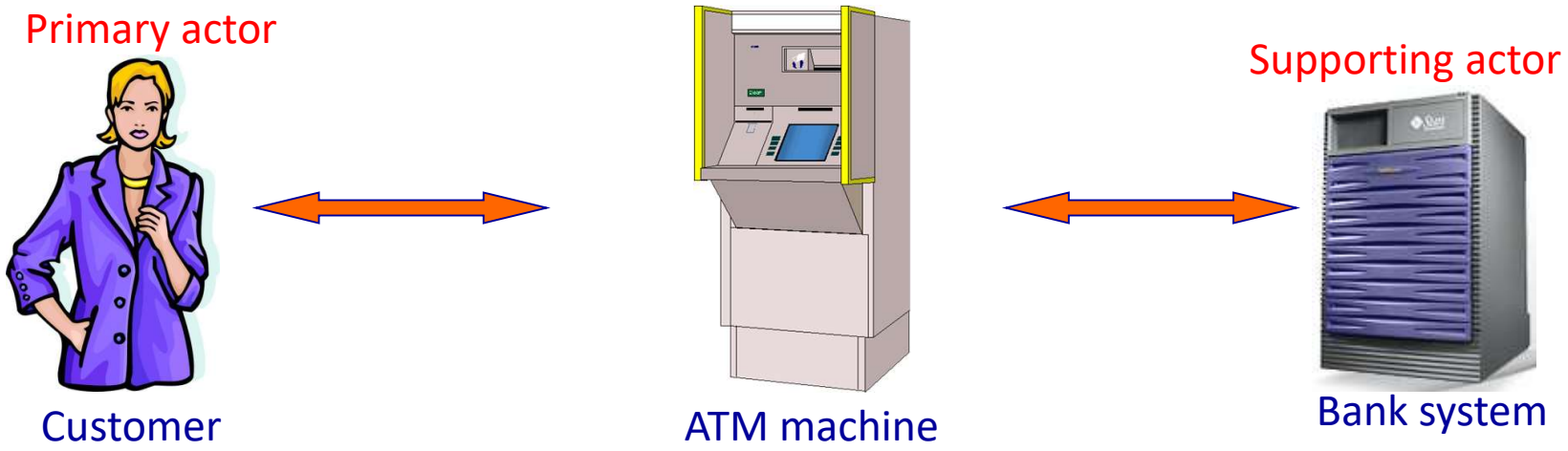
What is a Use Case?

- Use cases describe the sequential interaction between the user and the system to reach a certain goal.
- Use cases describe “what” the system does, not “how”.
- The system is seen as a black box, use cases describe the behavior of the black box



Actors

- External entity (human or other system) that uses the system, or is used by the system.
- Is not part of the system.
- Gives information to the system and/or receives information from the system.
  - Primary actor: starts the use case
  - Supporting actor: provides a service to the use case



Use cases

Primary actor

Customer

ATM machine

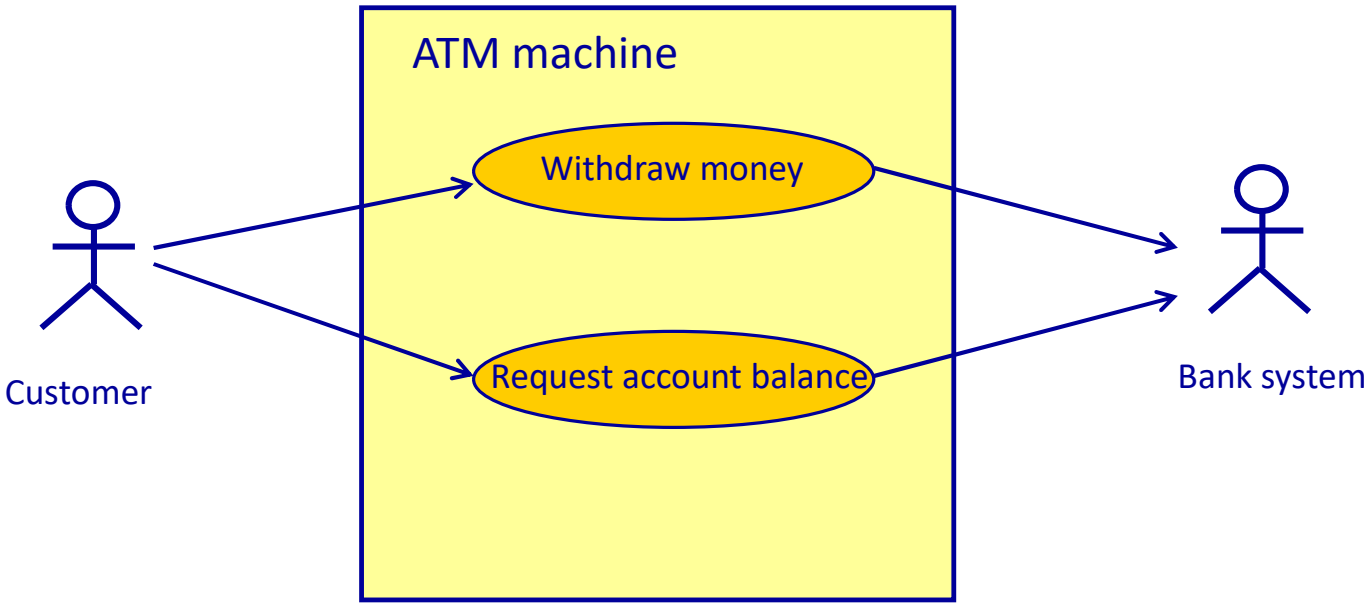
Supporting actor

Bank system

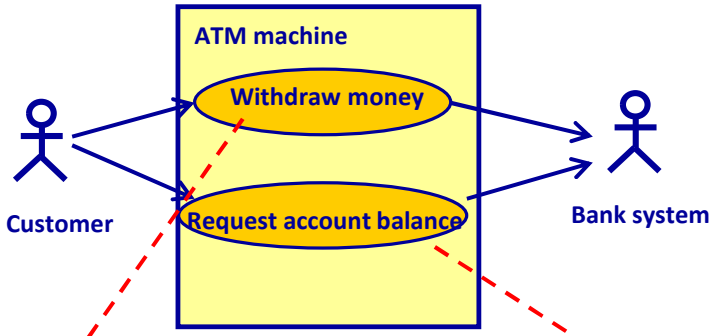
What are the goals of the customer to use the ATM machine?

Use case 1: Withdraw money
Use case 2: Request account balance

UML use case diagram



Use case descriptions



Withdraw money

- **Intend …**
- **Pre condition …**
- **Post condition …**
- **Main scenario(s)**
  1. **The customer enters the bank card**
  2. **The ATM asks for the PIN number**
  3. **The customers enters the PIN number**
  4. **…**
- **Alternate scenario's**
- **Exceptions**
- **Business rules**
- **Remarks**
- **…**

Request account balance

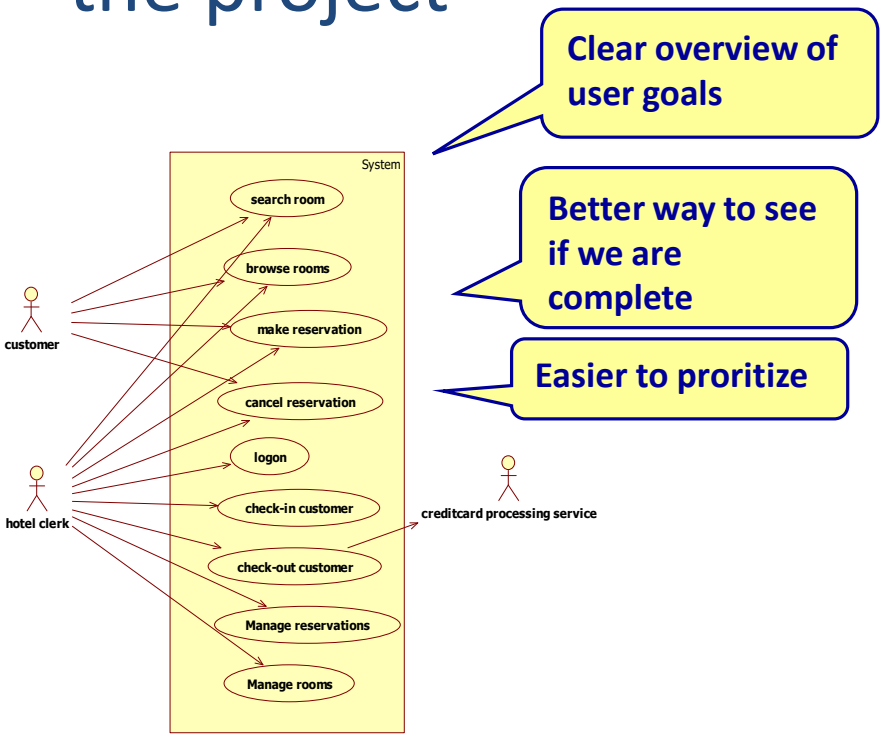
- **Intend …**
- **Pre condition …**
- **Post condition …**
- **Main scenario(s)**
  1. **The customer enters the bank card**
  2. **The ATM asks for the PIN number**
  3. **The customers enters the PIN number**
  4. **…**
- **Alternate scenario's**
- **Exceptions**
- **Business rules**
- **Remarks**
- **…**

Finding the right use cases

- A use case delivers a goal for the actor.
- A use case is atomic.
  - It has a beginning and an end.
  - When the use case is finished, you can go and drink a cup of coffee.

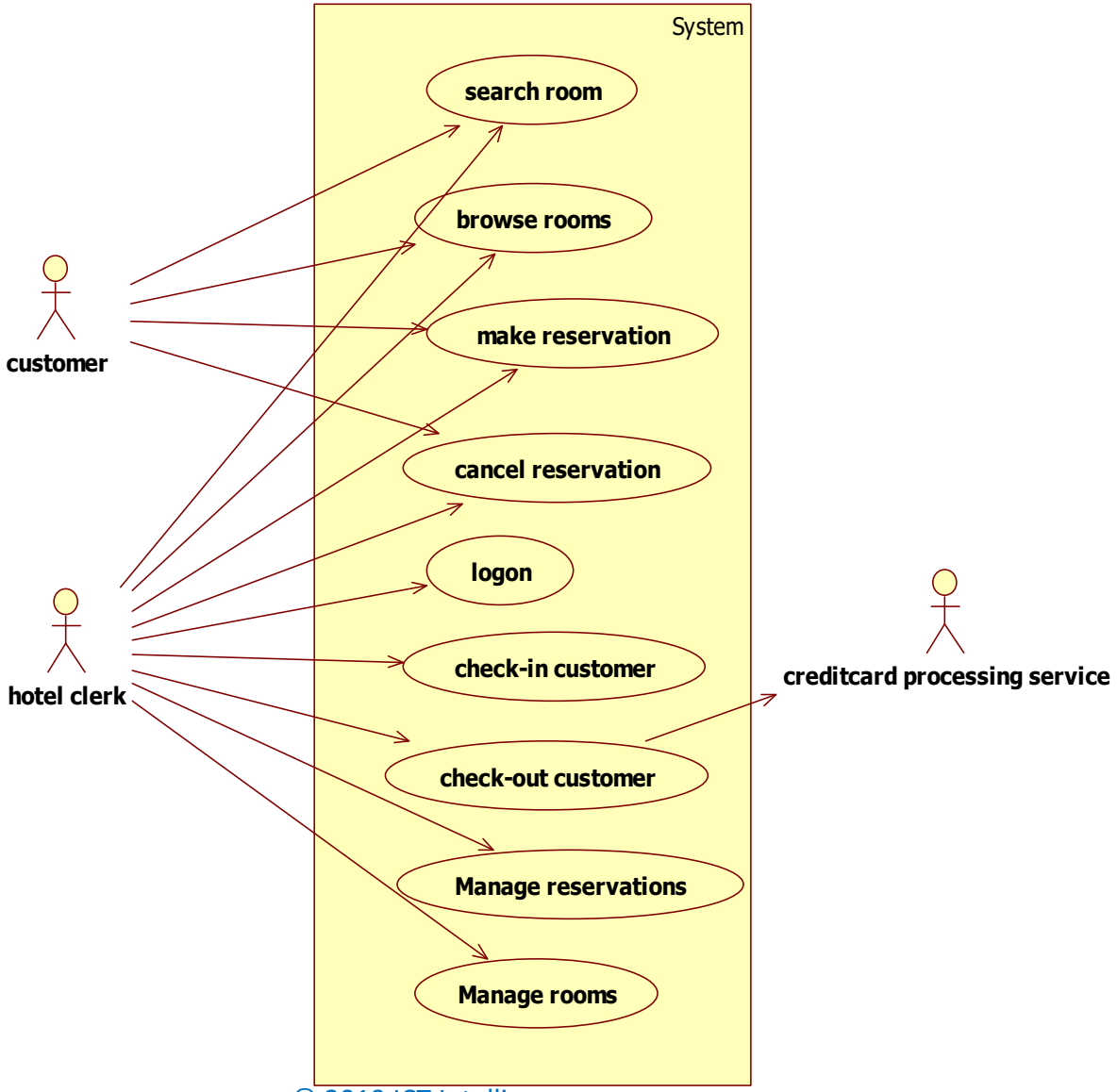
Why use cases in an agile project?

- Use cases show clearly the high level view of the project



The user can logon
The user view all its reservations
The user can view reservation details
The user can add a reservation
The user can cancel a reservation
The user can browse rooms by price
The user can browse rooms by roomtype
The user can browse rooms by availability
The user can view room details
The user can search rooms
The clerck can check-in customers with reservation
The clerck can check-in customers without reservation
The clerck can check-out customers
The clerck can process a creditcard payment
The clerck can process a cash payment
The clerck can process a check payment
…
…
…
…

Hotel reservation system use case diagram



Give actors a good name

- The name of the actor should be the role of that person
- The name should be as specific as possible

OK

HR manager

Order system

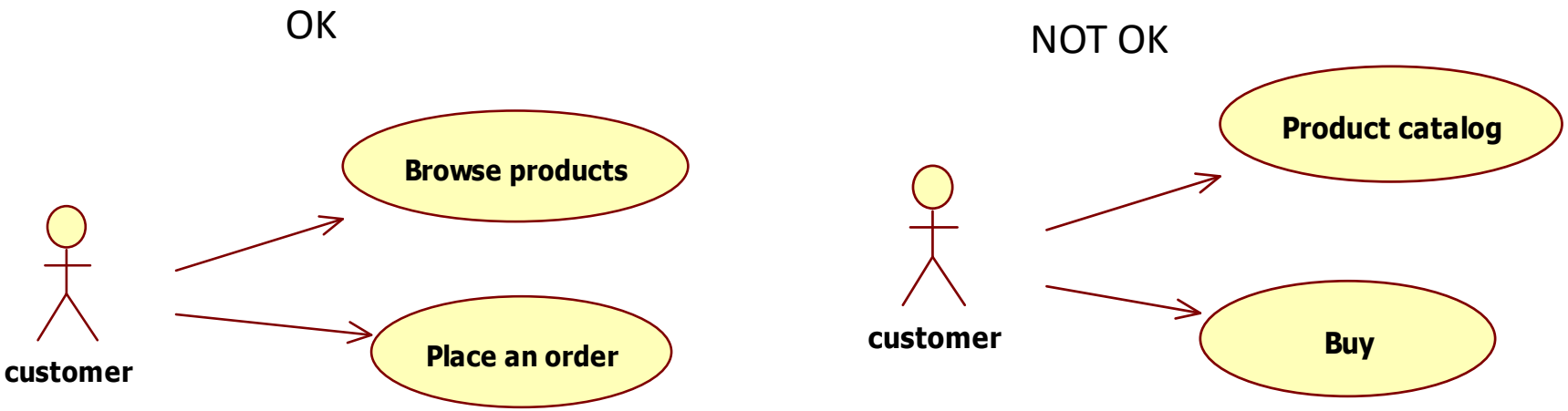
NOT OK

person

mainframe

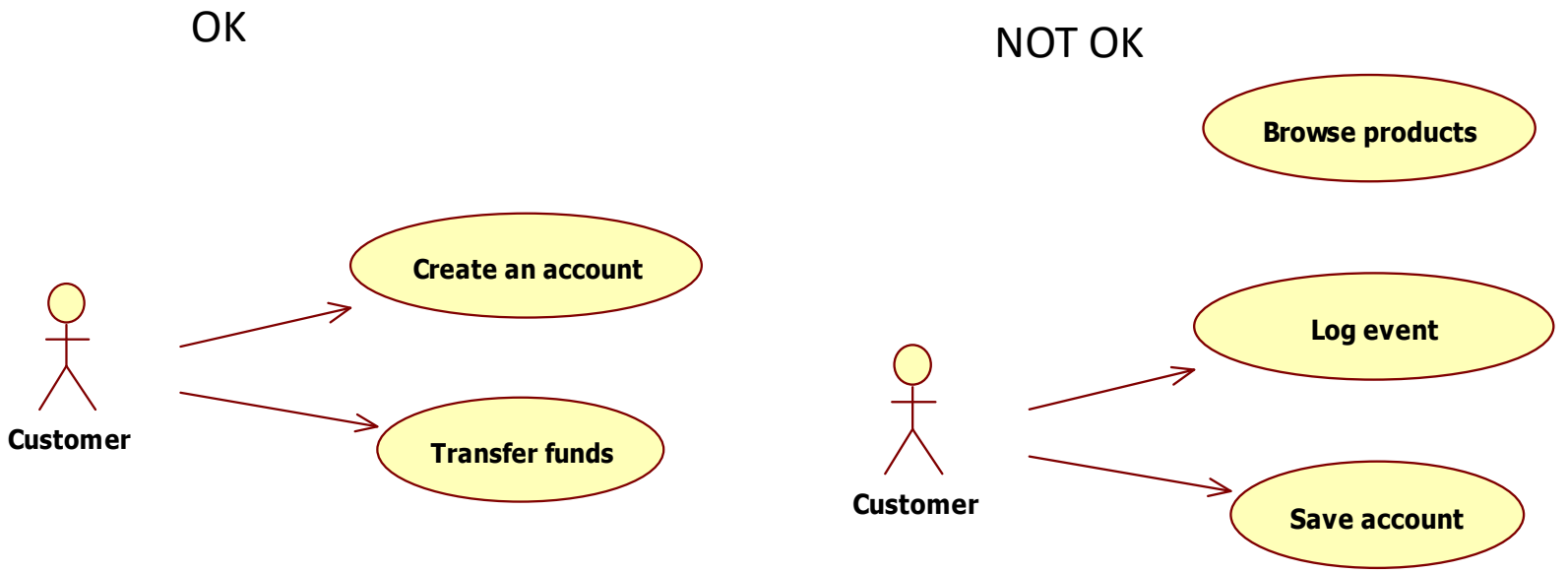
Give use cases a good name

- Should specify the goal the actor wants to accomplish
- Should be simple
- Should be short (max 5 words)
- Contains a verb

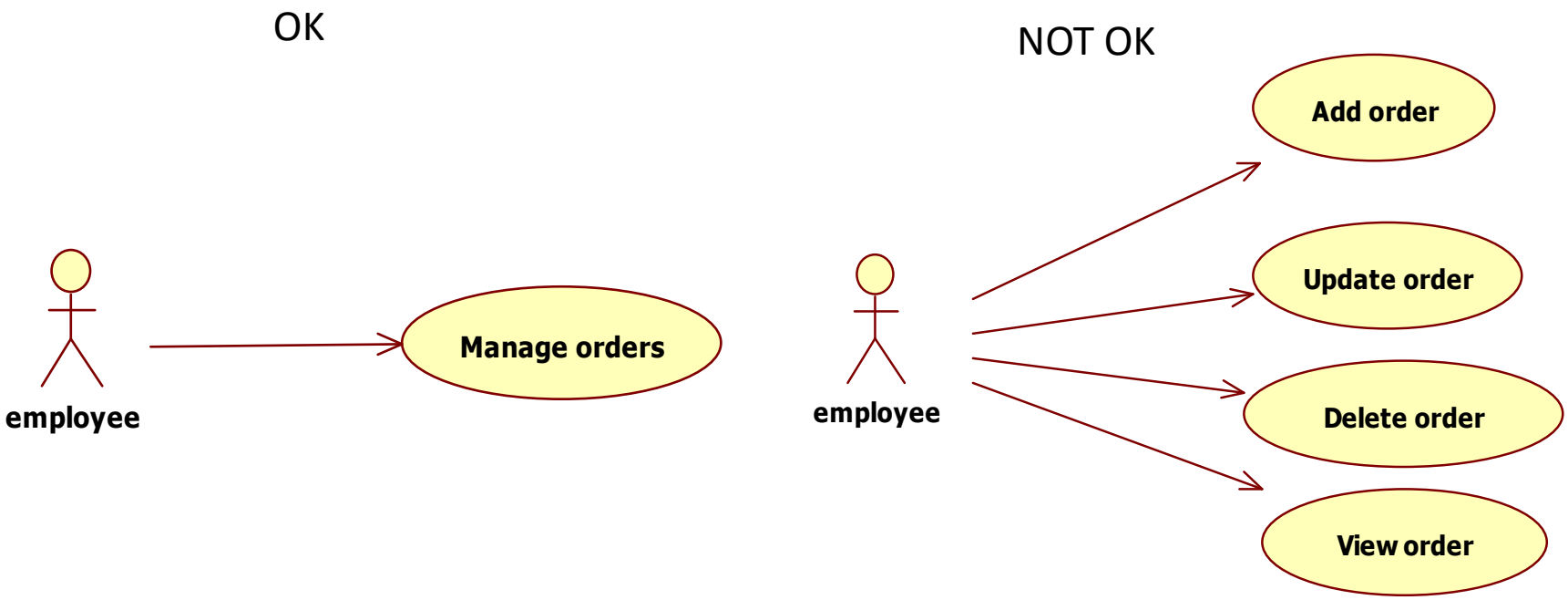


The result of a use cases is visible

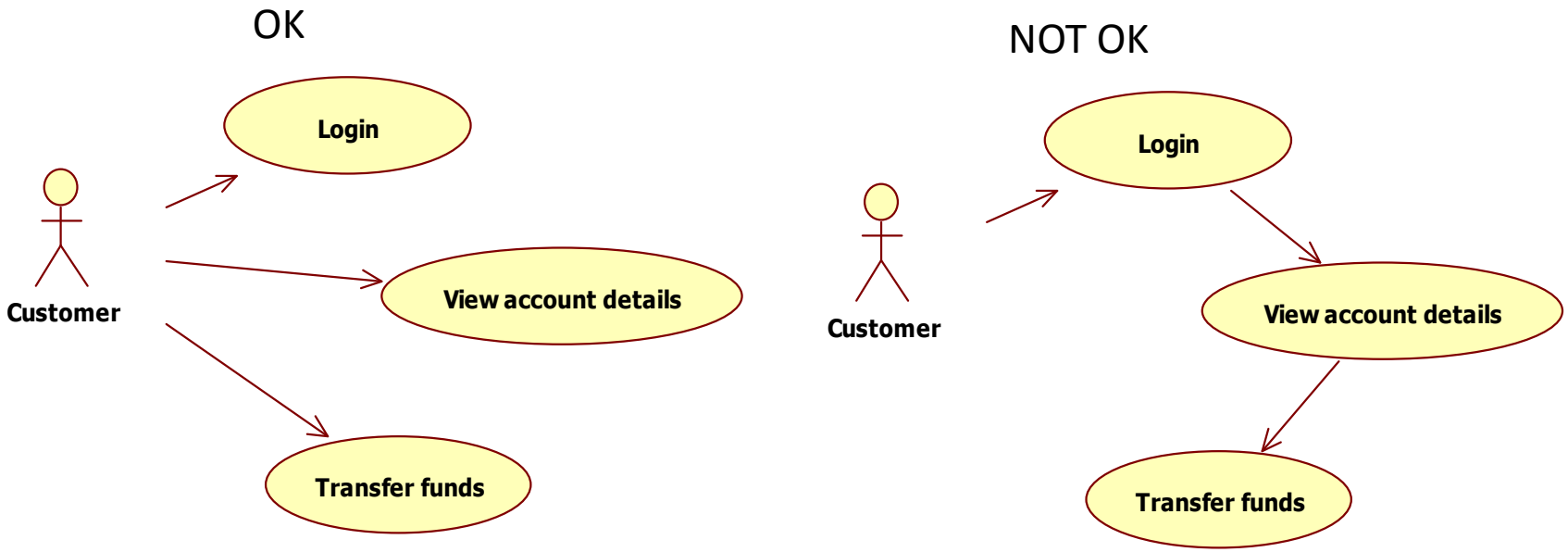
- Every use case has an actor (with the exception of <<include>> and <<extend>> use cases)
- Use cases describe the interaction between the actor and the systeem
- Use cases don’t describe internal actions
- Use cases describe requirements, not design or implementation



No CRUD use cases

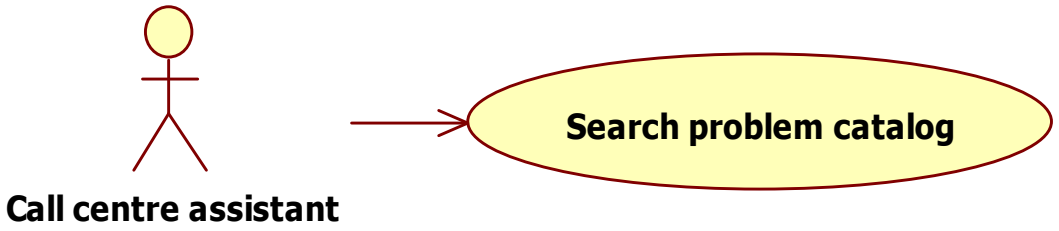


Do not model use case order

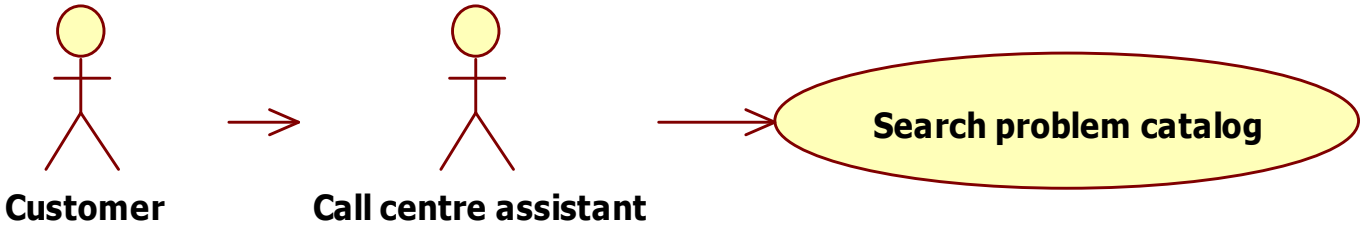


Do not model actor-actor relations

OK



NOT OK



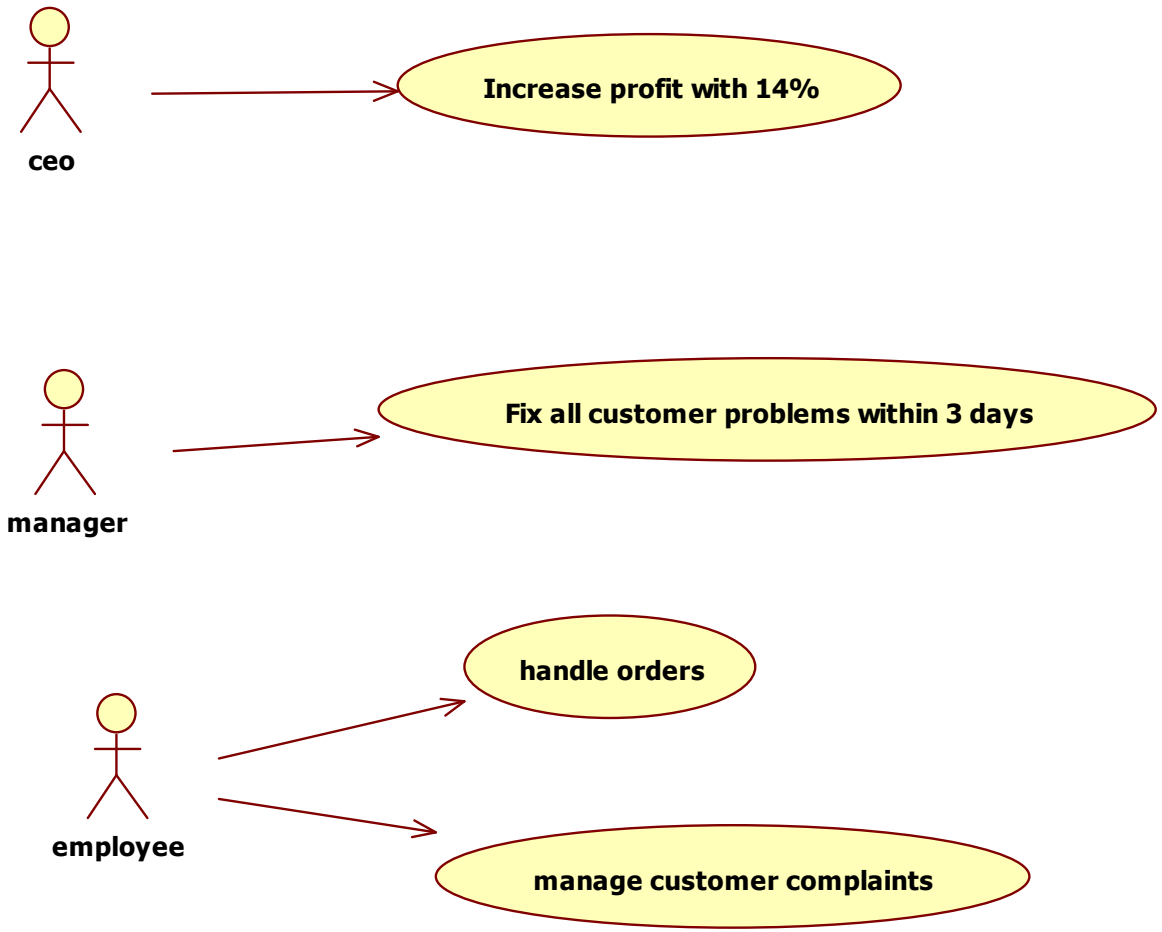
Use cases should be on the same level

High level goals

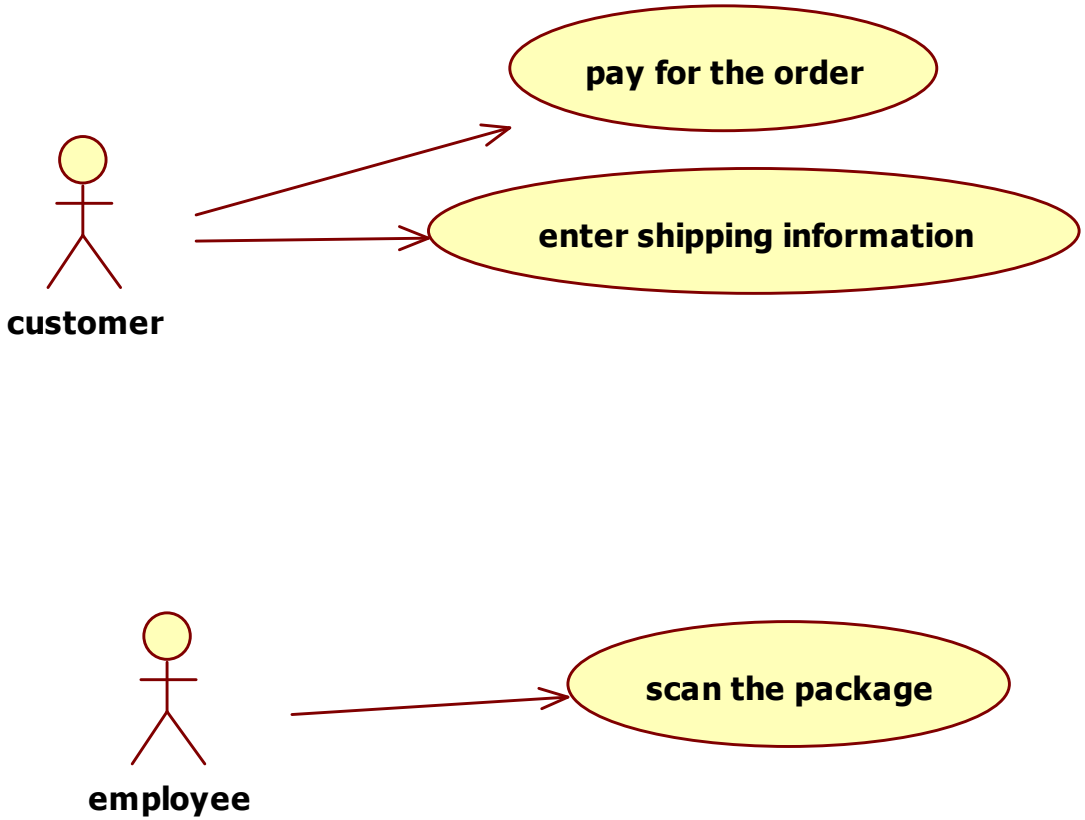
Goals you have with the system

Low level actions

Too high level use cases



Too low level use cases



Correct use cases

- Goals the actor has with the system