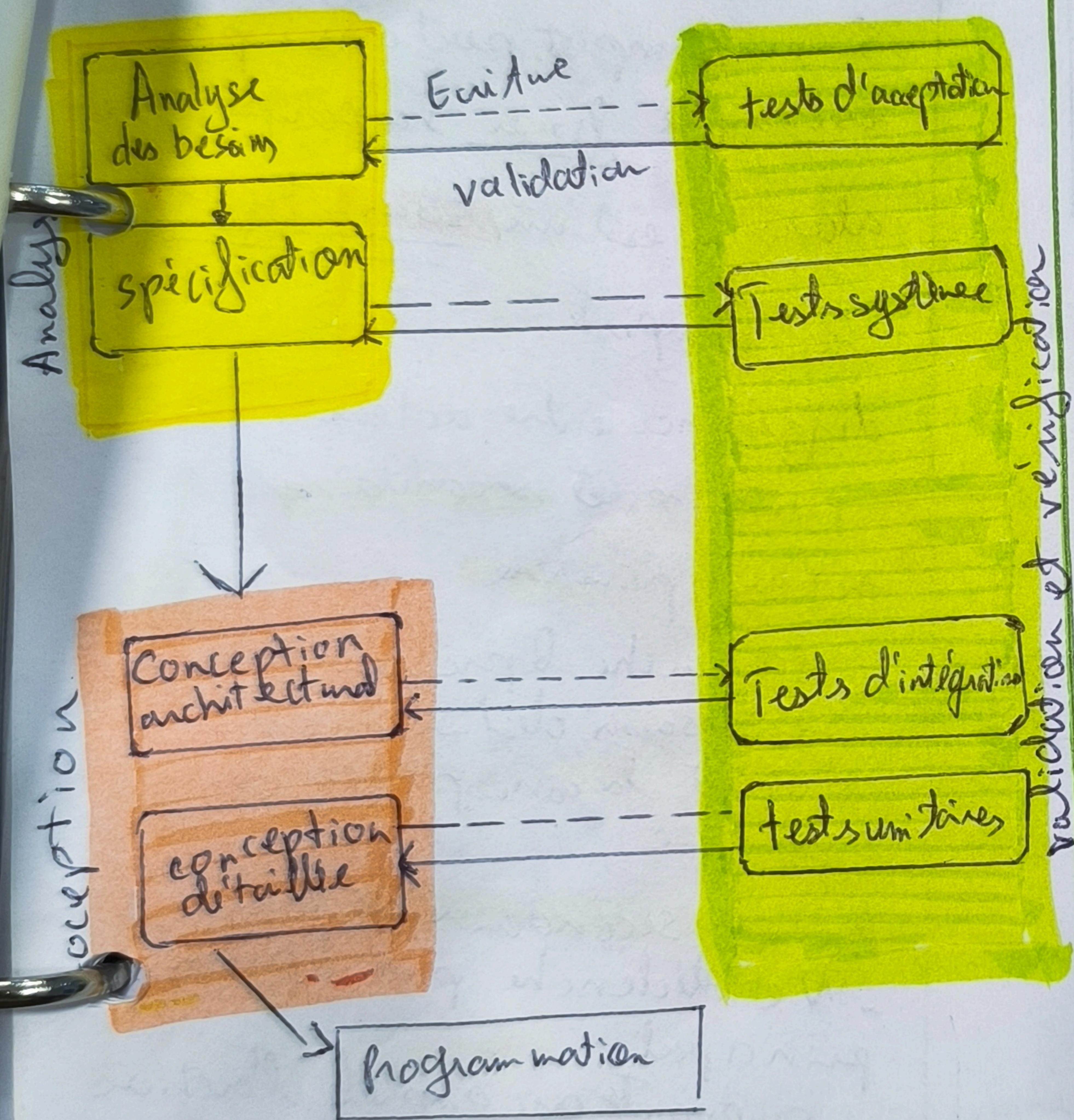


# Conception des systèmes d'information

## Diagramme de cas d'utilisation :



## phase d'analyse :

Comprendre les besoins du client pour rédiger le cahier des charges fonctionnel

Trois questions :

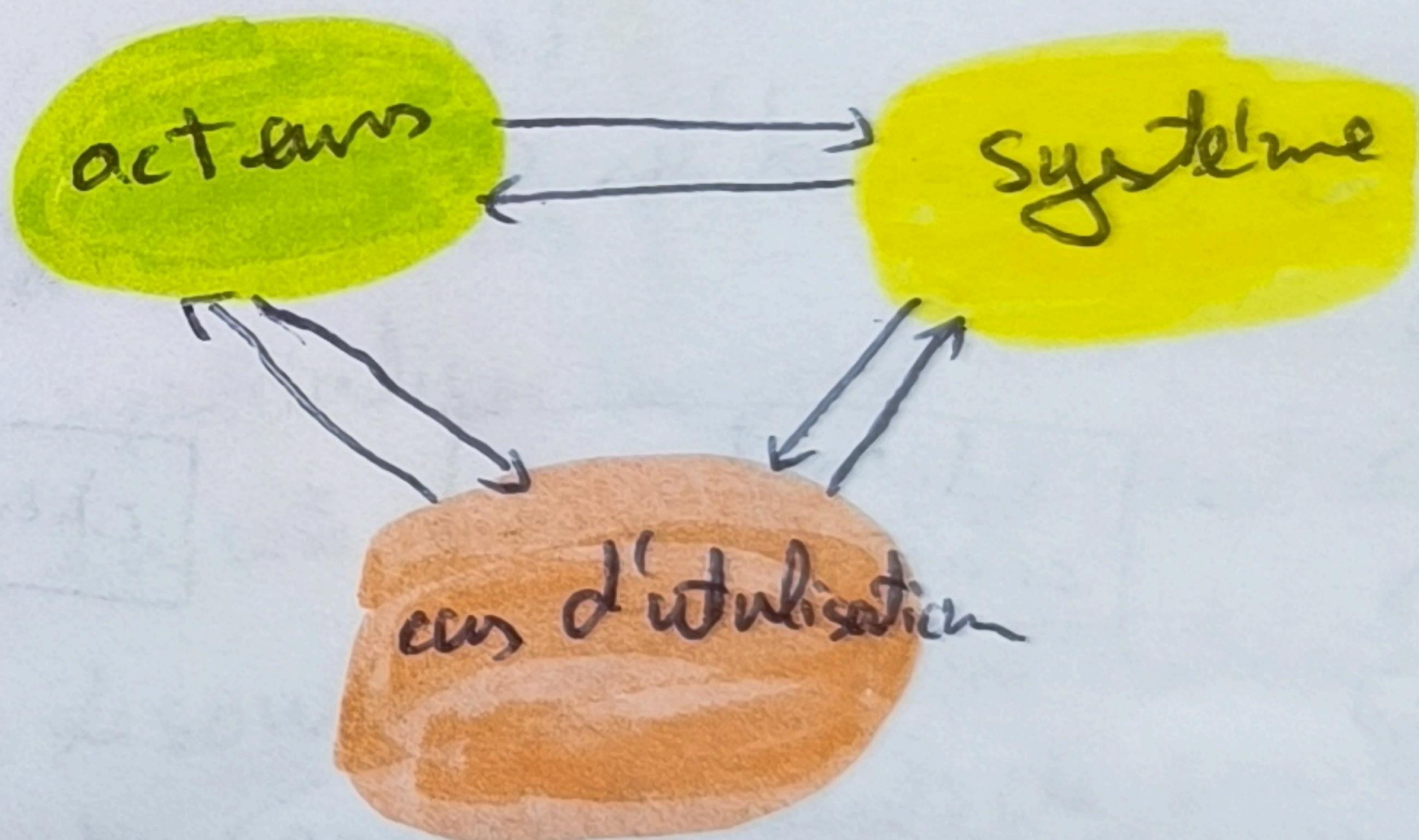
- 1 - Définir les utilisations principales du système : à quoi sert-il ?

## UML

2 - Définir l'environnement du système : qui va l'utiliser ou interagir avec lui ?

3 - Définir les limites du système

Un diagramme de cas d'utilisation UML constitue la première étape de l'analyse de l'UML. Ce diagramme comprend 3 choses :



### Acteur :

Entité qui interagit avec le système

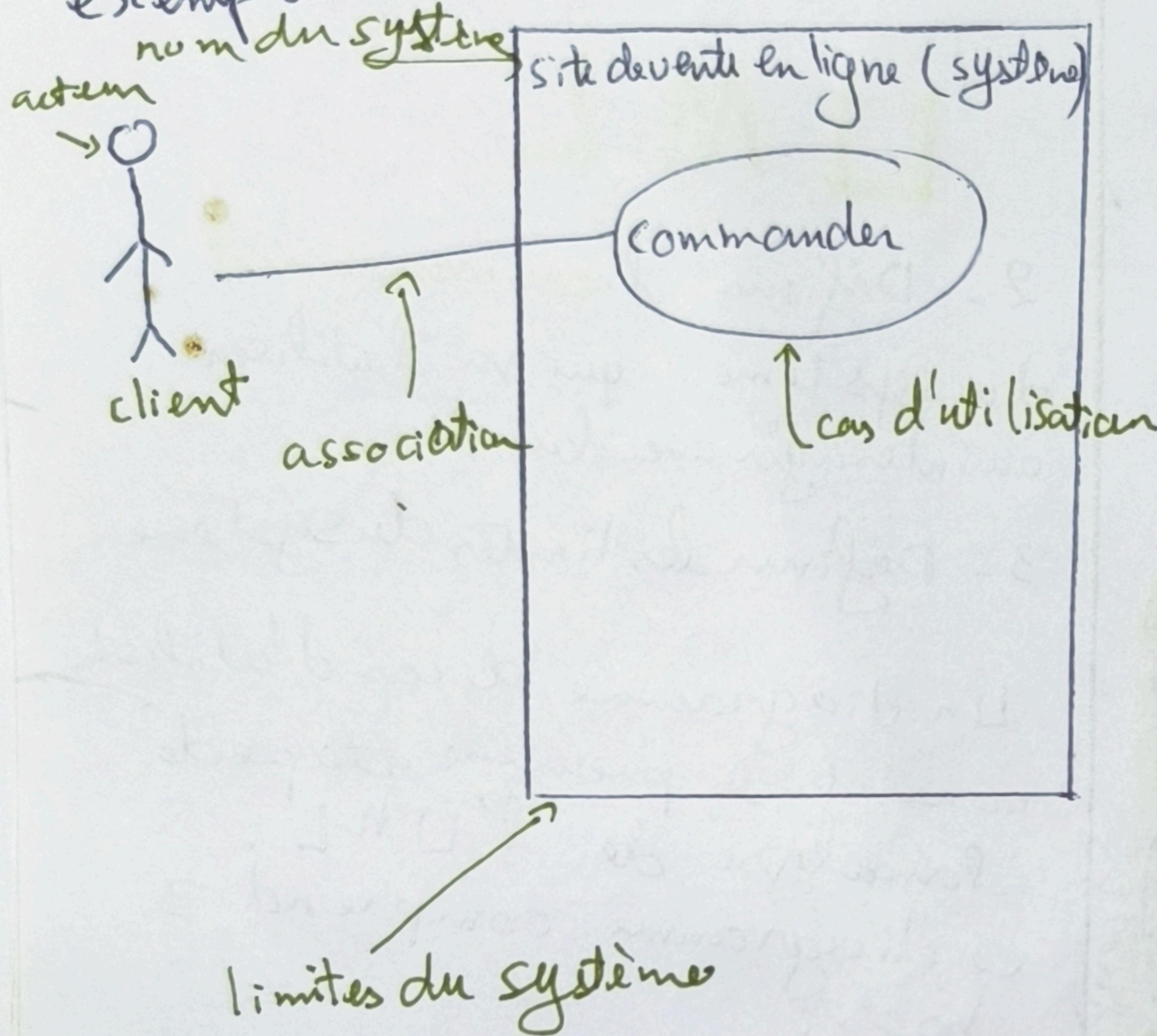
• peut être :

- une personne
- matériel (chose, logiciel)
- un sys extérieur
- représente un rôle et identifié par un cas d'utilisation

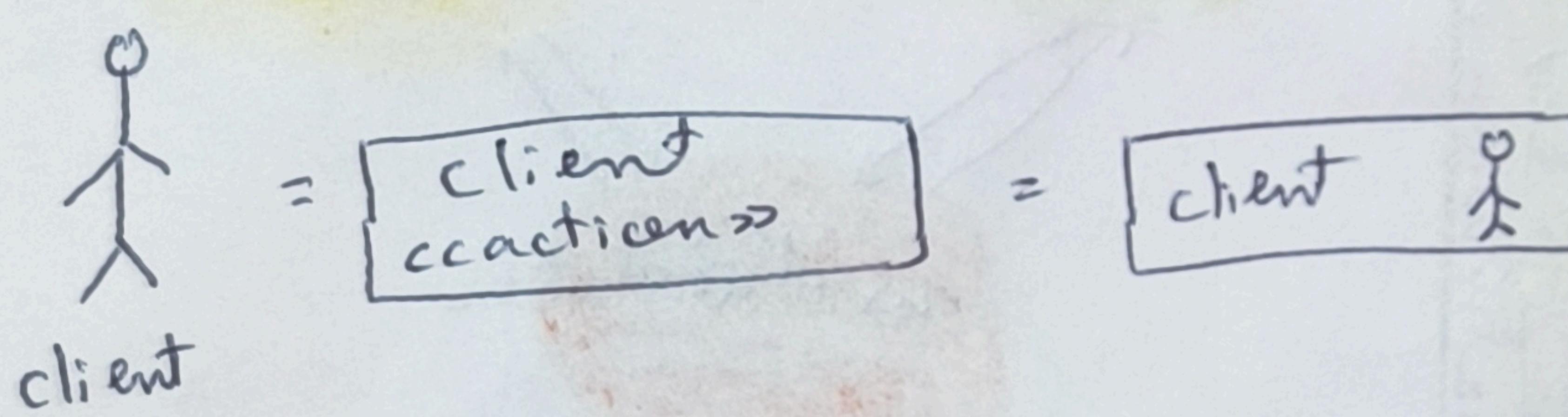
Fonctionnalité visible de l'ext intérieur du système.

- Action déclenché par un acteur
- identifié par une action (verbe à l'infinitif)

exemple:



des schéma d'un acteur



Types d'acteurs:

1- Acteur interne :

Accès généralement complet aux données internes, connaissances des procédures internes.

2- Acteur externe : il interagit avec le SI pour obtenir/fournir des informations mais ne participe pas directement à la gestion interne de l'entreprise

Exercice :

quand ahmed le pompiste utilise la station pour remplir sa boîte il est un acteur

### Primaire

Si ahmed le pompiste arrive avec sa camion-citerne pour remplir les réservoirs des pompes il est considéré comme un nouvel acteur secondaire.

Si un pompiste peut aussi par exemple fixer la pompe alors il est un sous-type

de pompiste

différence entre acteur primaire et secondaire

### Acteur primaire

- il déclenche le processus principal
- Ses besoins déterminent souvent la conception du système

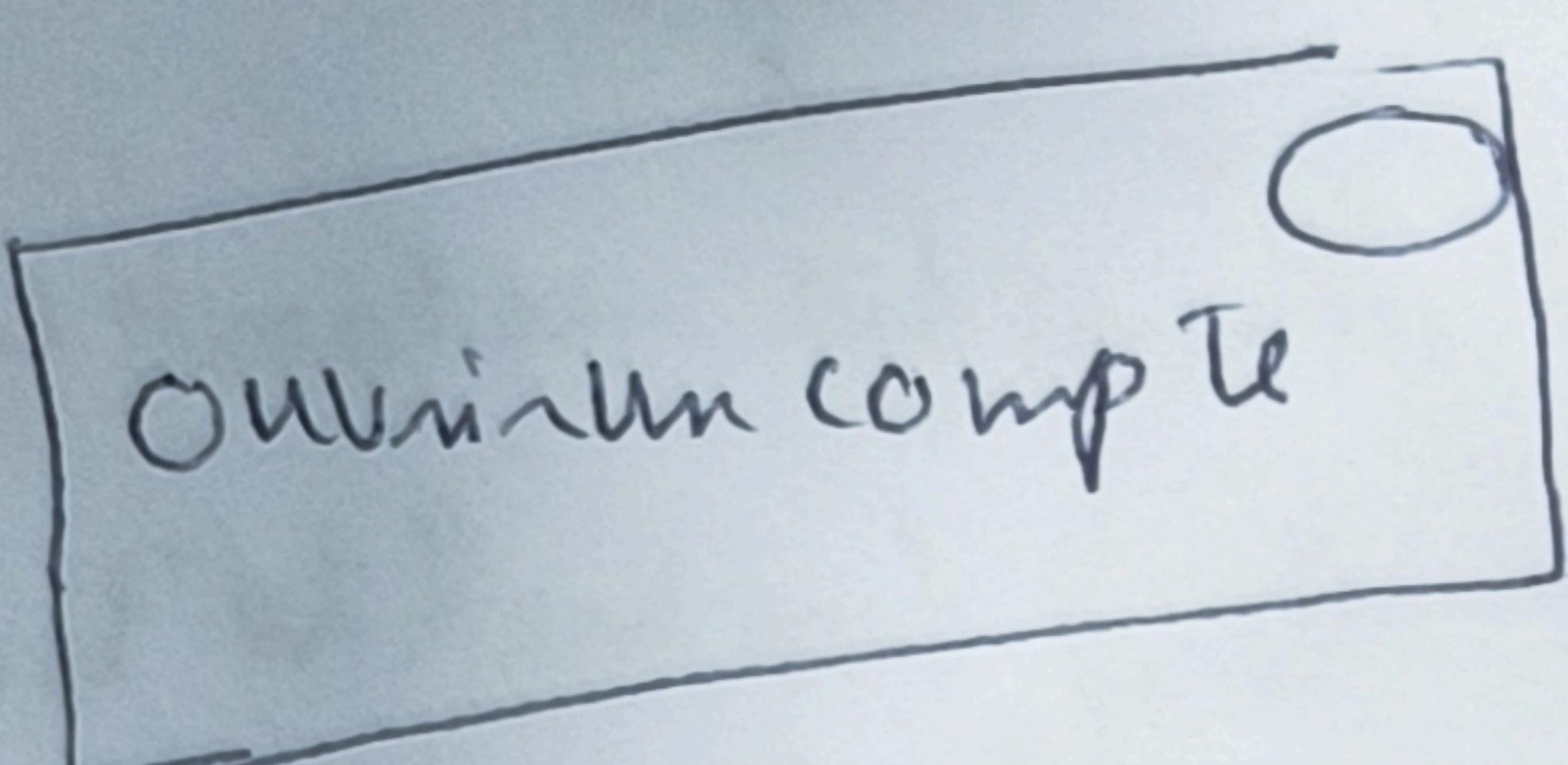
### Acteur secondaire :

- Ne déclenche pas le processus principal
- supporte ou complète l'action de l'acteur primaire

cas d'utilisation

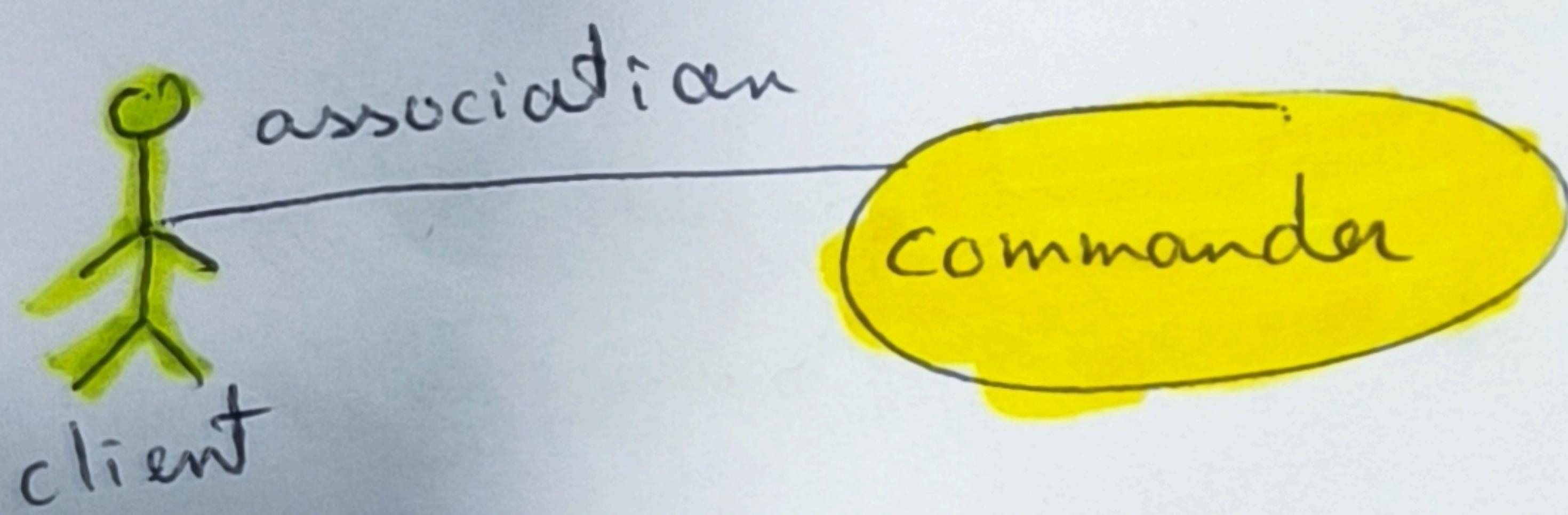
own un compte

ou



## Association

- relation entre acteurs et cas d'utilisation
- représente la possibilité pour l'acteur de déclencher le cas



cas d'utilisation interne: un cas n'est pas directement lié à un acteur (peut être un acteur externe)

## Inclusion:

- X implique Y si Y est nécessaire pour X

impliquer = obligation

cas d'utilisation destinataire **includes** cas d'utilisation source

**extension :**

- extension = optionnelle:
  - extends Y => X peut être provoqué par Y
  - X est optionnelle pour Y
- point d'extension:
  - l'extension peut intervenir à un point précis du cas
  - peut avoir plusieurs extensions



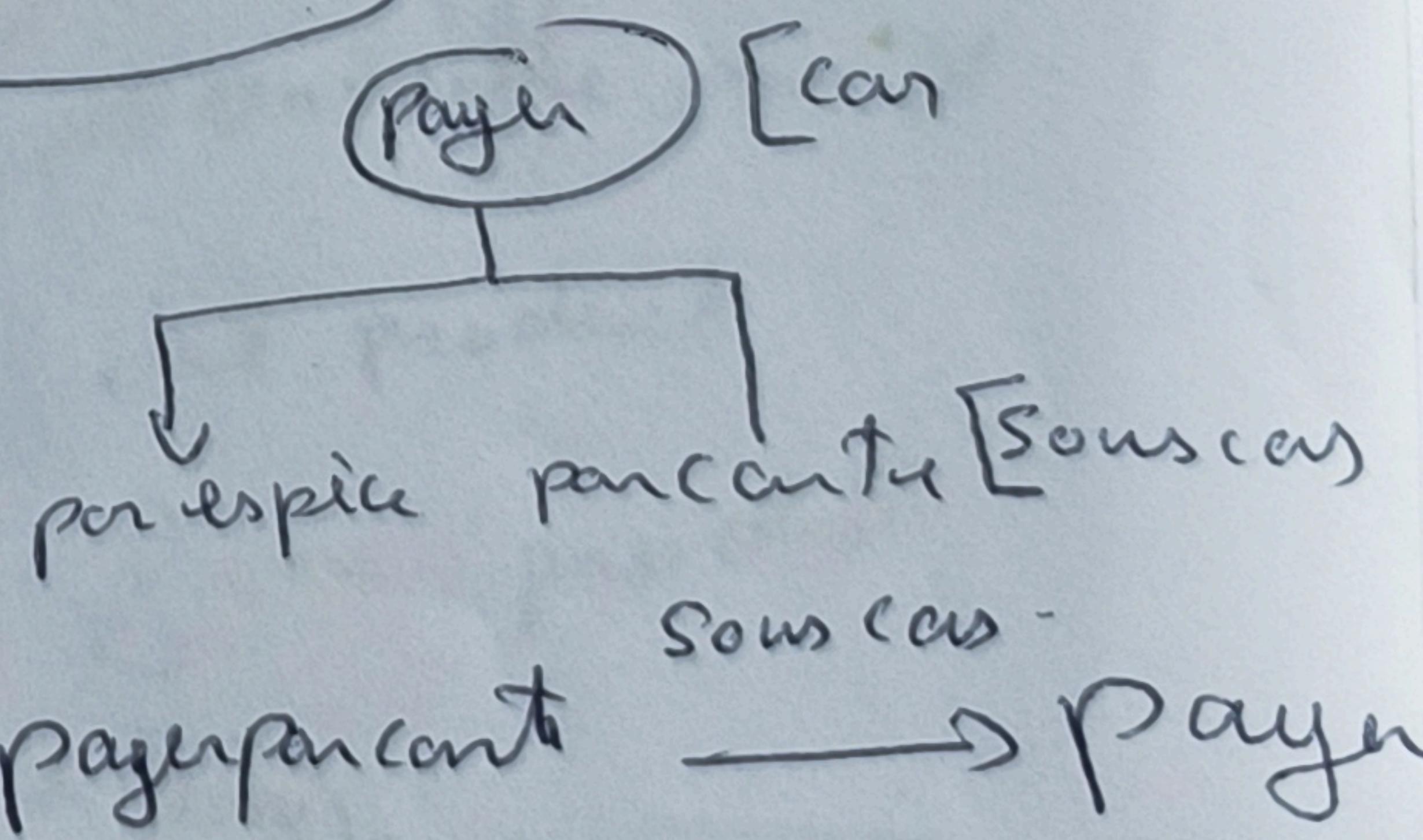
## généralisation ou spécialisation

(héritage)

X est un cas particulier de Y  
spécialiser un cas d'utilisation en un autre cas d'utilisation

- des sous cas d'utilisation héritent du comportement du sup-cas d'utilisation (fille-mère)

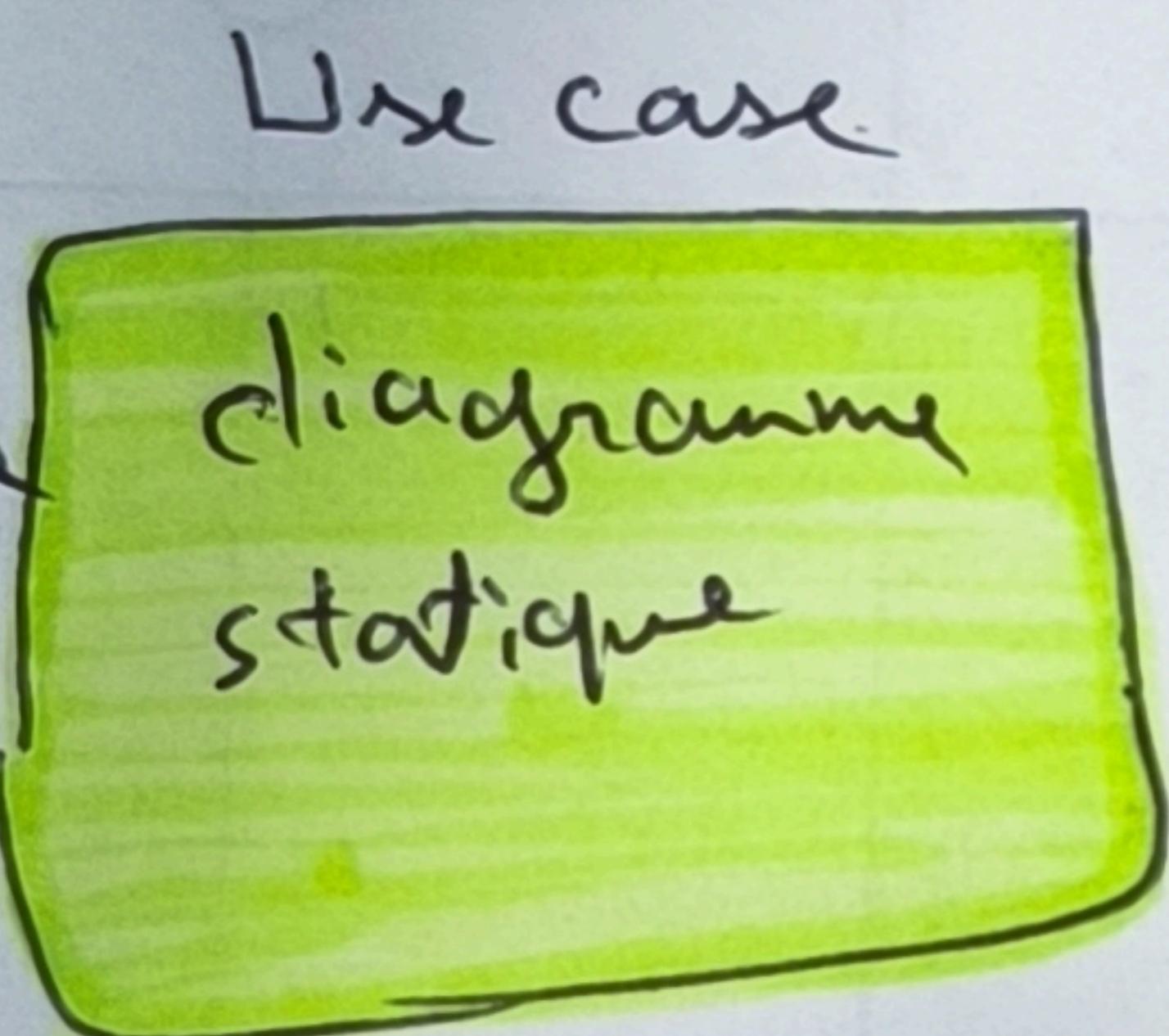
- des sous cas d'utilisation héritent aussi de toutes les associations du sup-cas



# conception des systèmes d'information

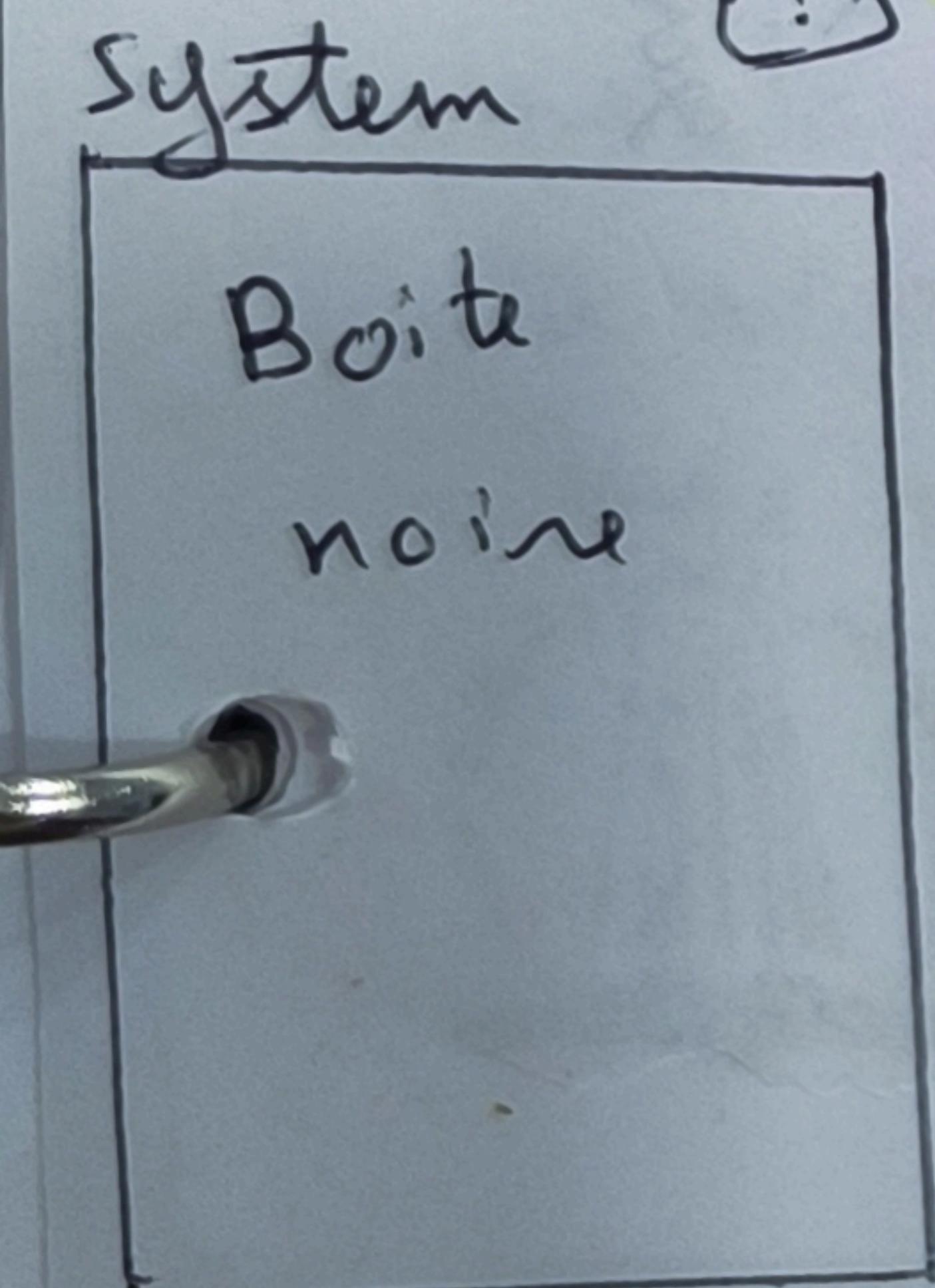
spécification des besoins fonctionnels

- \* acteur : Un acteur est un rôle
- \* use case (cas d'utilisation) : fonctionnalité
- \* Diagramme de cas d'utilisation détaillé



Il se case A includes B

⚠ n'est pas temporel



limiter le système

acteur

« primary » déclenche le système

« secondary » sollicité par le système

Y est optionnelle pour X

consultation produit

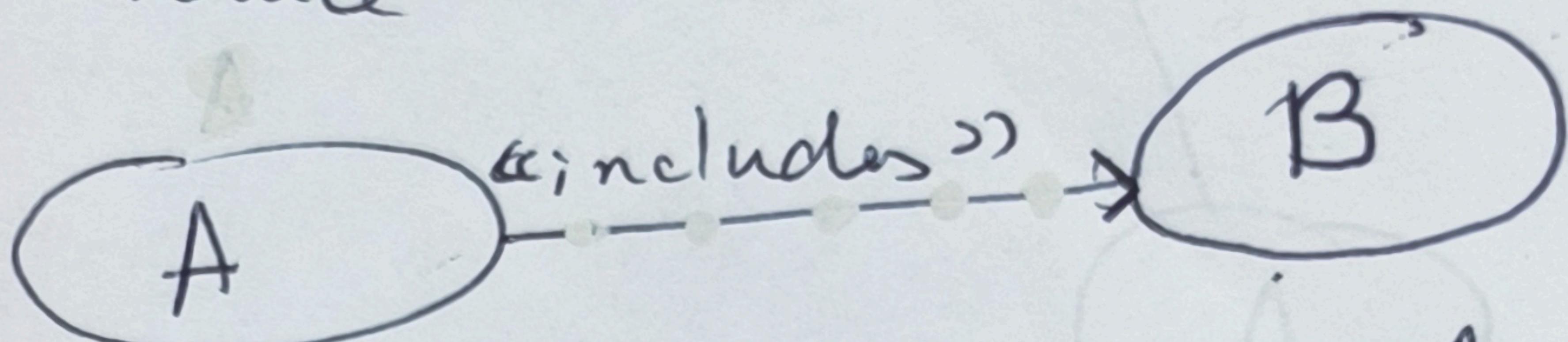
imprimer fiche produit

consultation fiche produit

extends

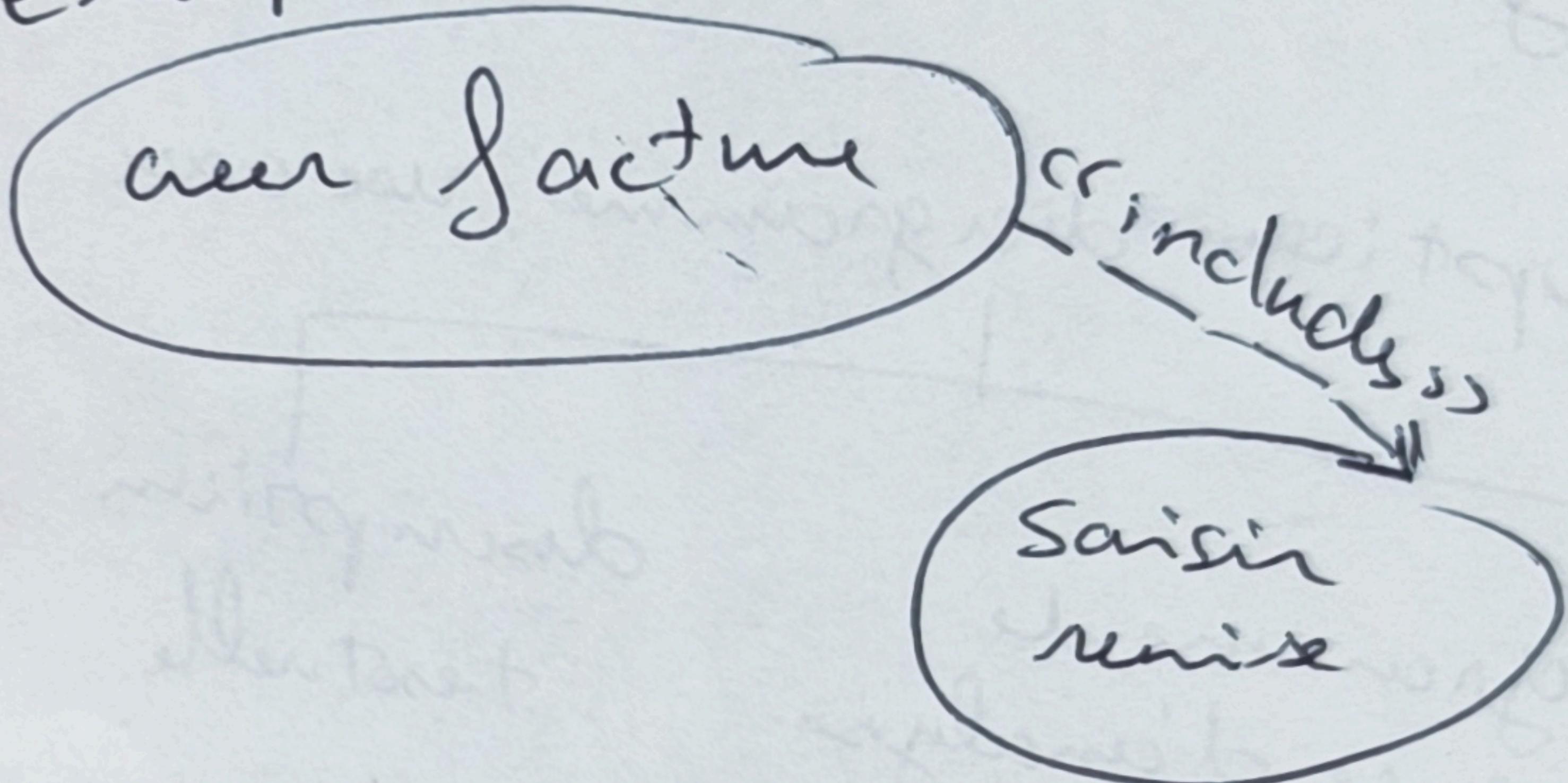
imprimer fiche produit

A  $\xrightarrow{\text{includes}}$  B  
Source destination



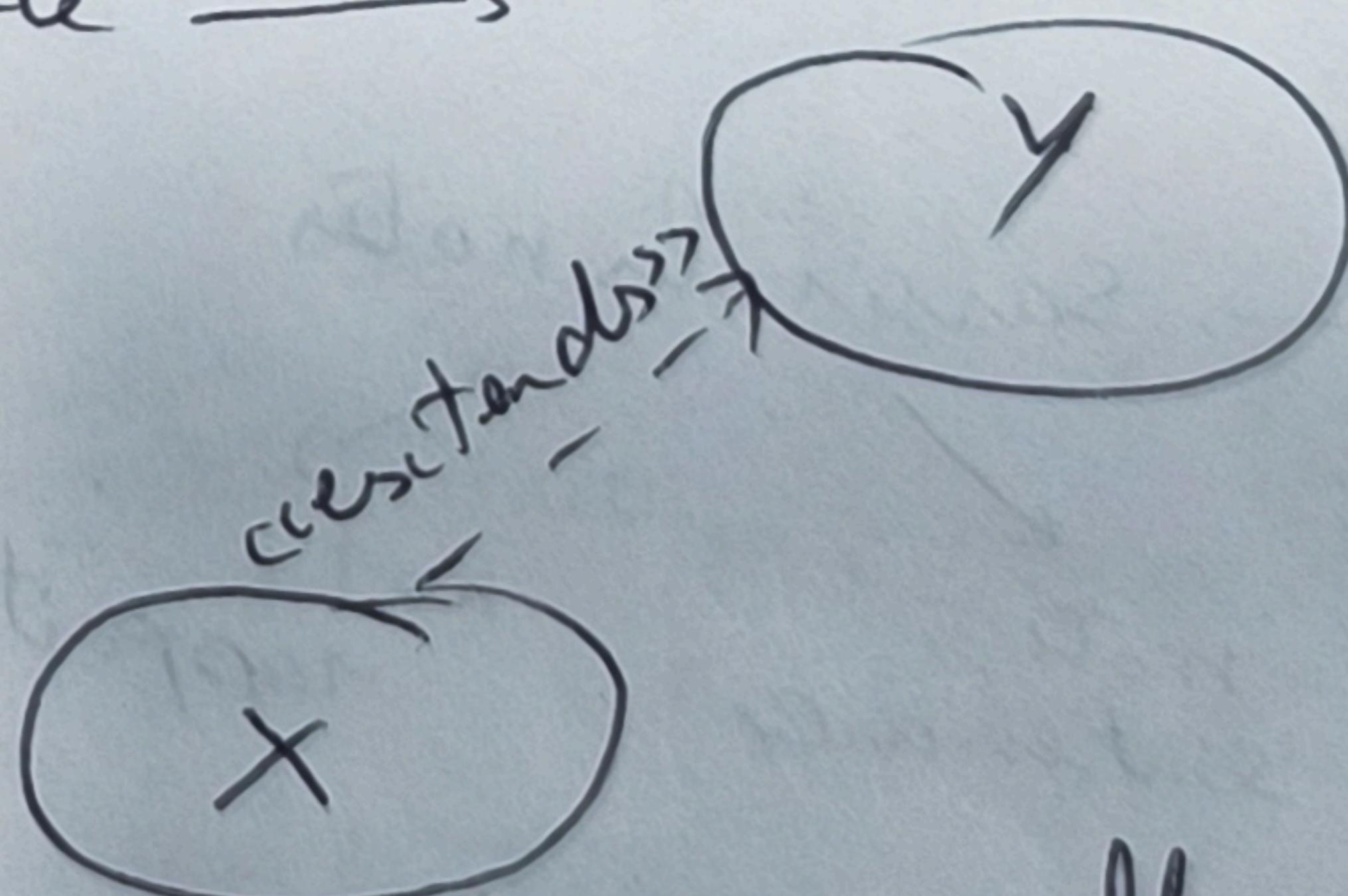
B est obligatoire pour A

exemple



⇒ Saisir remise est obligatoire pour création facture

X extends Y  
source  $\longrightarrow$  destination



Y est optionnelle pour X

consultation produit

imprimer fiche produit

consultation fiche produit

extends

imprimer fiche produit

Specialisation  
A specialise y

A est un sons car de Y

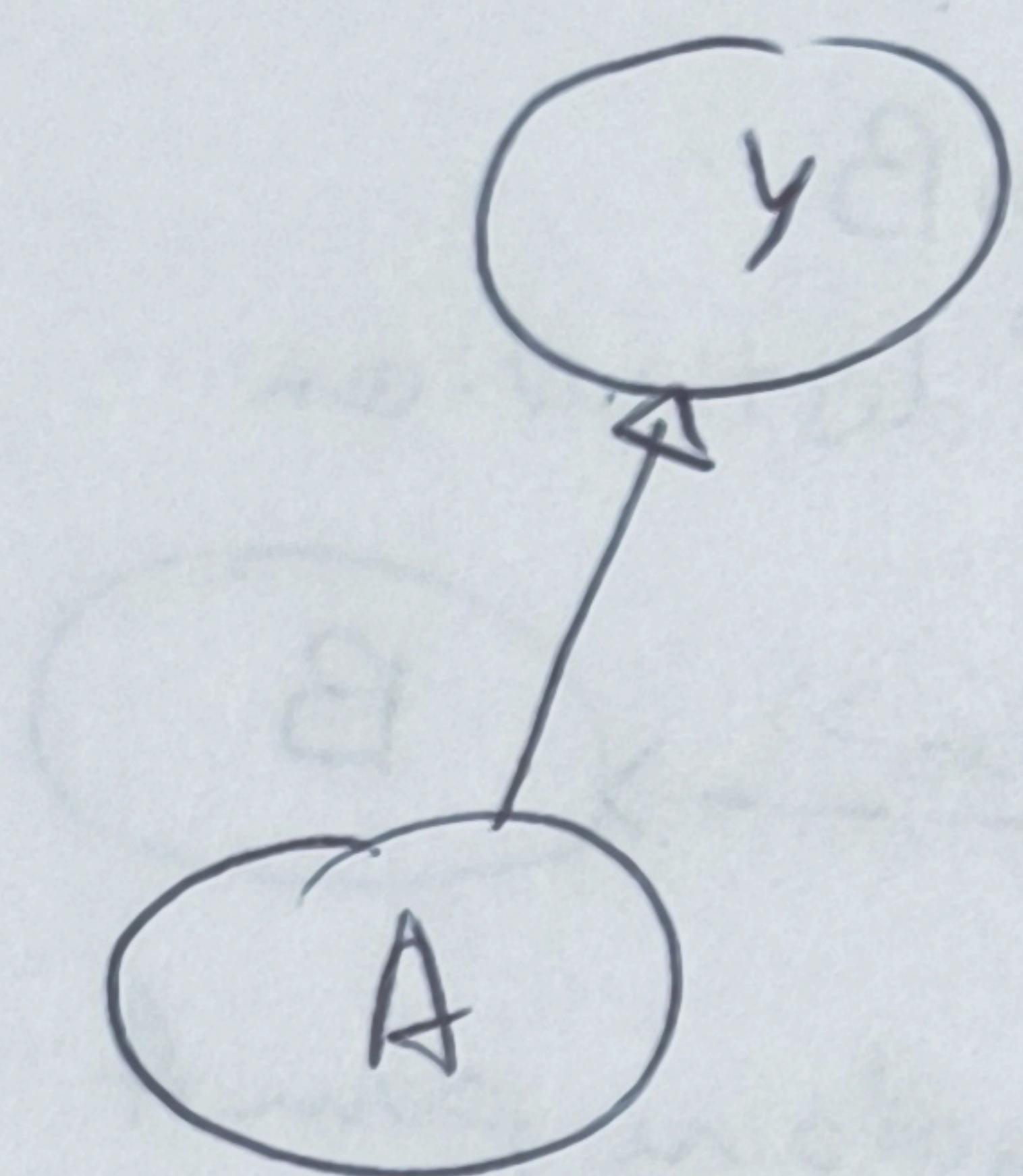
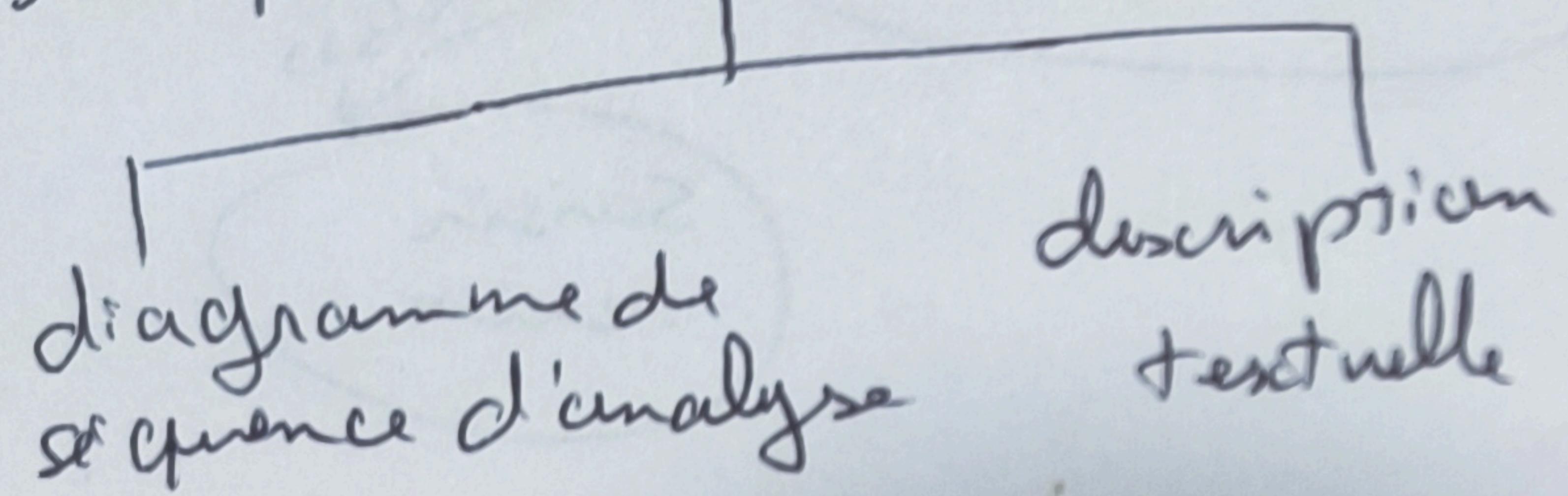


diagramme de cas d'utilisation

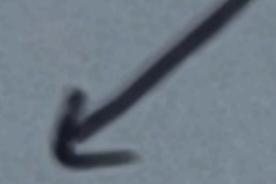
Description diagramme use case



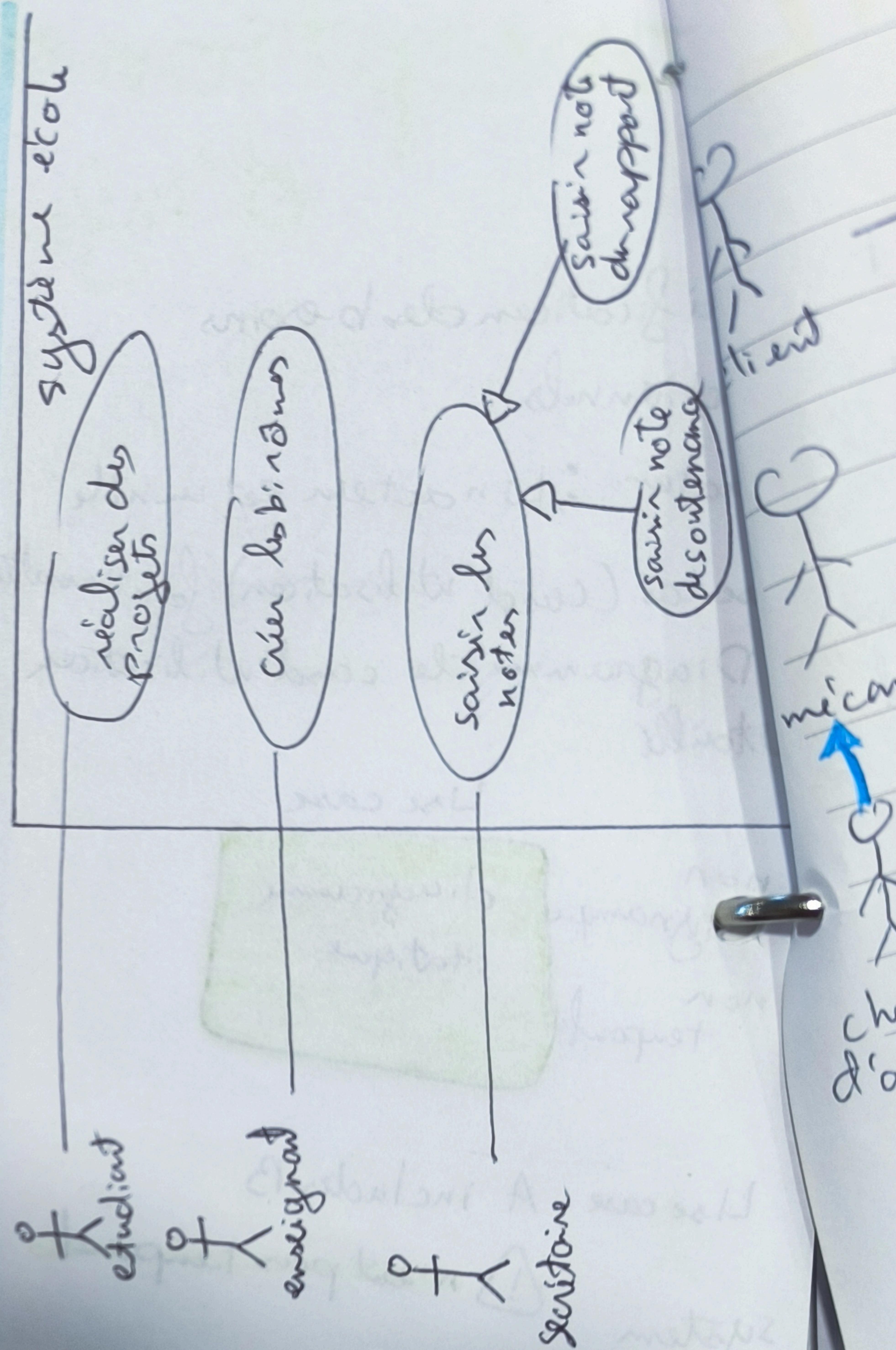
Etudiant : réaliser des projets

enseignant → enier lə̃bi nɔ̃m


scrivain → saisir les notes

 note  
sentence

 note rapport



Le ..... / ..... / .....



↑ : on représente chef d'atelier comme une classe qui hérite à la fois de mechanicien et de gérant → c'est une héritage multiple.

# Enoncé LM

## Exercice n°1:

1)

L'acteur du système est le client.

Le pistolet et la gâchette sont des ressources utilisées par le système

2)

Si le pompiste fait uniquement les tâches d'un client, il est inutile de créer un nouvel acteur. Il sera lui-même client

tâche pompiste = tâche client

⇒ même acteur dans le système

3)

Oui, le gérant est un nouvel acteur alias interne qui gère le système.

4)

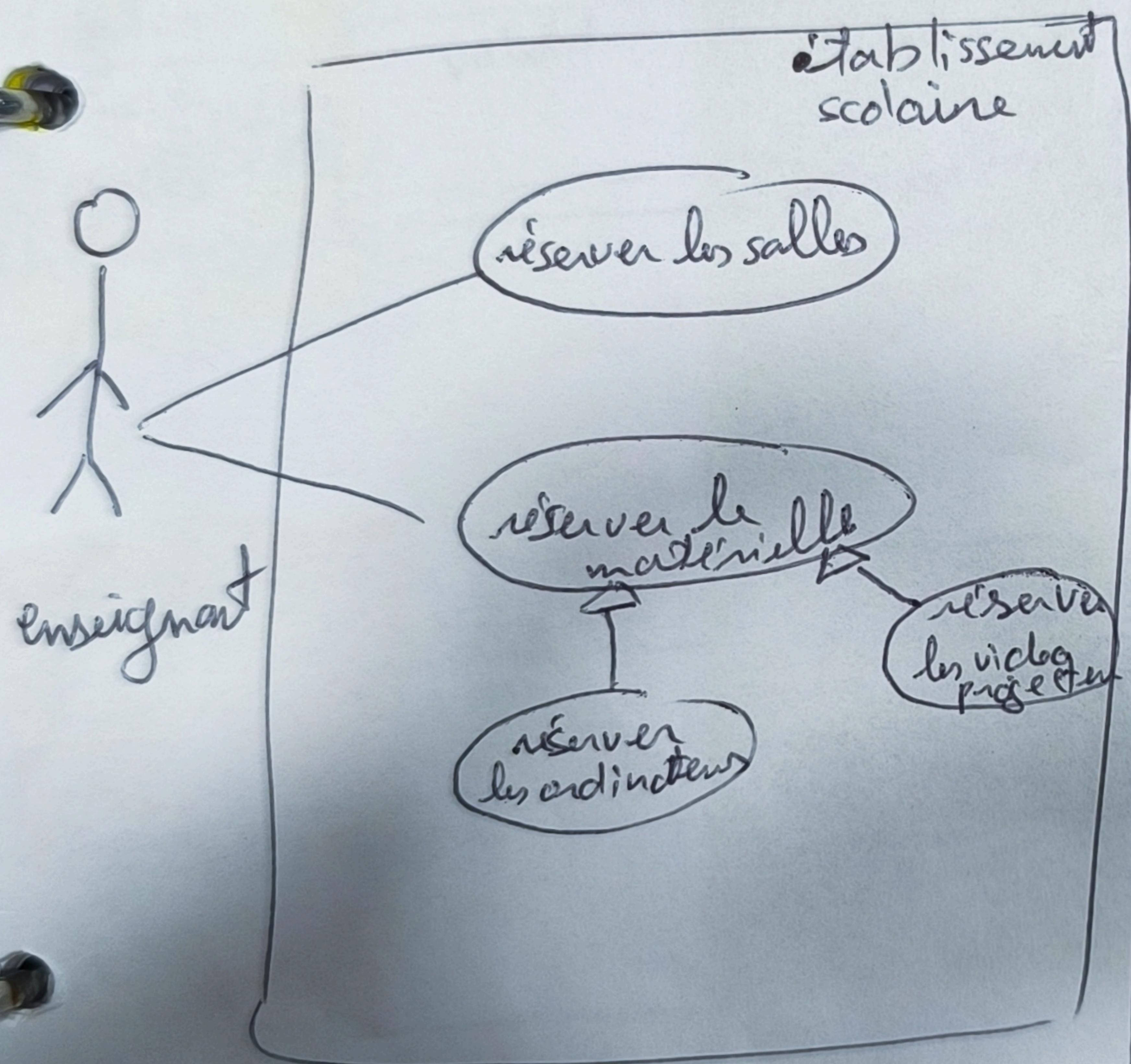
Le chef d'atelier est un sous cas de mécanisme car il possède certaines fonctions (généralisation)



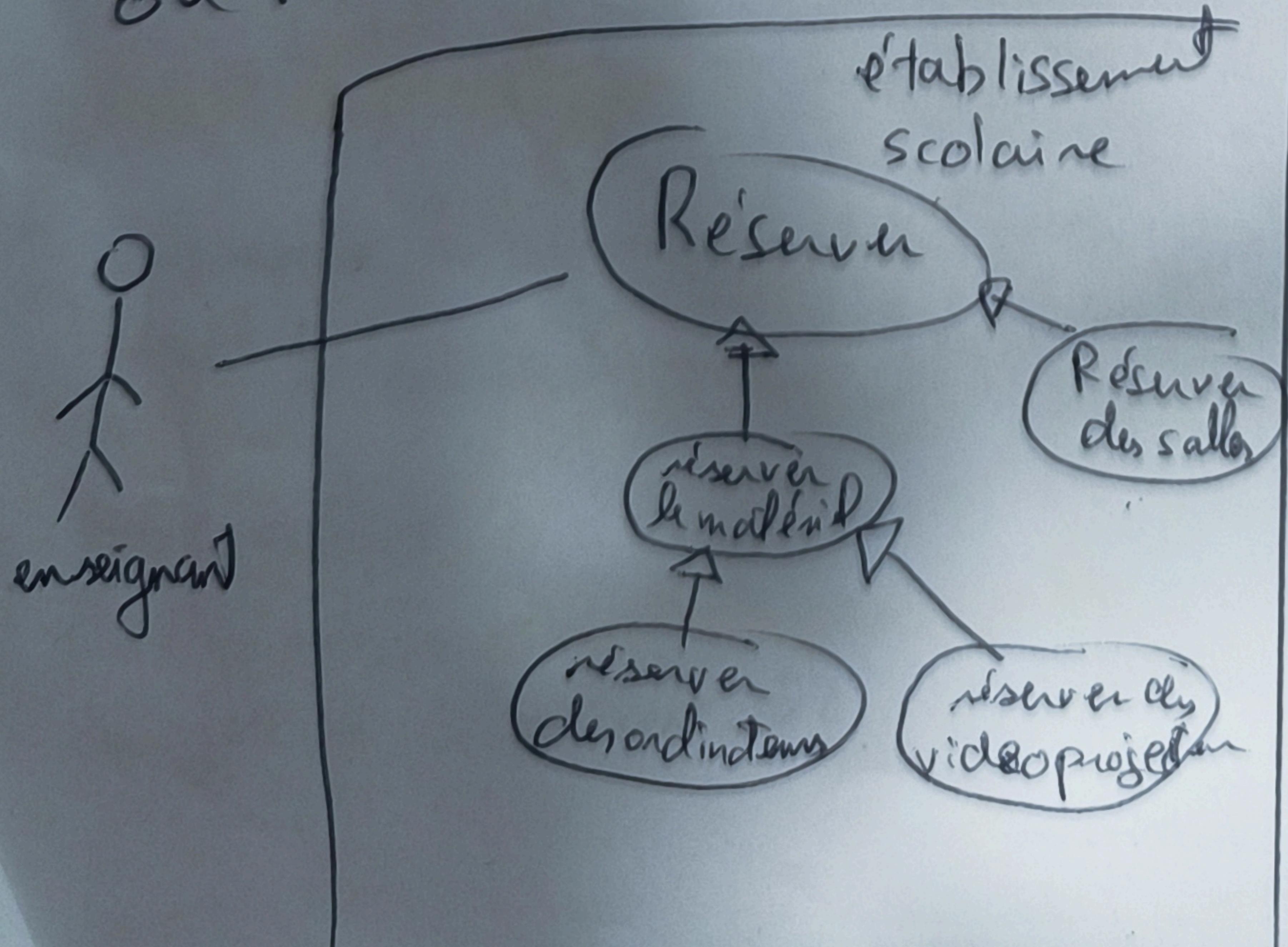
# Conception des Systèmes d'information

a) l'enseignant est en action

- b/ réservé les salles de cours
- réservé le matériel pédagogique



ou bien :



Q2if 3 + 4 + 5

