



# Plan

- ❑ **INTRODUCTION À LA MODÉLISATION DES BESOINS**
- ❑ **LES ÉLÉMENTS DU DIAGRAMME DE CAS D'UTILISATION**
- ❑ **LES RELATIONS DANS UN DIAGRAMME DE CAS D'UTILISATION**
- ❑ **DESCRIPTION TEXTUELLE D'UN CAS D'UTILISATION**
- ❑ **EXERCICE**

Avant de développer un système, il est essentiel de déterminer précisément **À QUOI** il devra servir et quels besoins il devra satisfaire.

## ❑ **Modéliser les besoins permet de :**

- Faire l'inventaire des fonctionnalités attendues.
- Organiser les besoins de manière à faire apparaître des relations (possibilités de réutilisation, etc.).

Avec UML, on modélise les besoins au moyen de **diagrammes de cas d'utilisation**.

## ➔ **Un diagramme de cas d'utilisation décrit**

- **le système**
- **les acteurs**
- **les cas d'utilisation**

## ➔ **Il contient: des descriptions textuelles**

## → Le système

- Le système est un ensemble de cas d'utilisation
- Le système ne comprend pas les acteurs.



Nom de système

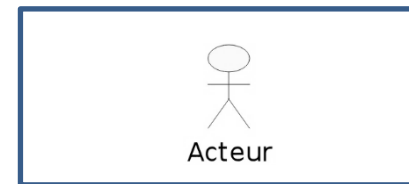
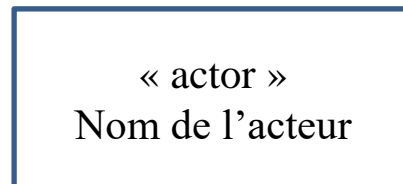
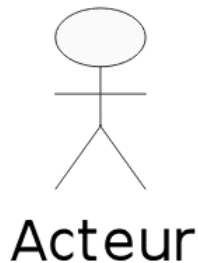
The diagram shows a large rectangular box with a black border, representing a system boundary. The text 'Nom de système' is centered at the top of the box.

## → Les acteurs

Un **acteur** est une entité extérieure au système modélisé, et qui interagit directement avec lui. *Exemple*: un client, un guichetier, un responsable maintenance, Une même personne peut jouer plusieurs rôles.

Un acteur exécute un ou plusieurs cas d'utilisation, et est représenté par:

- un petit bonhomme (stick man) avec son nom en dessous
- par un rectangle contenant le mot-clé << actor >> avec son nom (**Un acteur non-Humain**)
- Par un mélange de ces 2 représentations



### ▪ Pour les identifier:

Quelles sont les entités externes au système qui interagissent directement avec le système ?

## → Les acteurs

### ❑ Identification des acteurs

- Les principaux acteurs sont les utilisateurs du système.
- Une même personne physique peut être représentée par plusieurs acteurs si elle a plusieurs rôles.
- Si plusieurs personnes jouent le même rôle vis-à-vis du système, elles seront représentées par un seul acteur. En plus des utilisateurs, les acteurs peuvent être :
  - Des périphériques manipulés par le système (imprimantes...) ;
  - Des logiciels déjà disponibles à intégrer dans le projet ;
  - Des systèmes informatiques externes au système mais qui interagissent avec lui, etc.

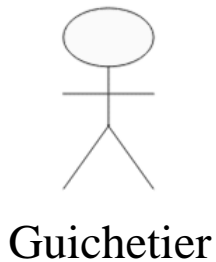
#### *Attention*

Un acteur correspond à un rôle, pas à une personne physique.

## → Les acteurs

- **Description des acteurs:** Pour chaque acteur :
  - Choisir un identificateur représentatif de son rôle
  - Donner une brève description textuelle

### *Exemple:*



- Un guichetier est un employé de la banque. Interface entre le système informatique et les clients.
- Il peut effectuer une série d'opérations : création d'un compte, dépôt et retrait d'argent, etc.

- **Utilité des acteurs :** La définition d'acteurs permet;
  - D'identifier les cas d'utilisation

### *Exemple:* Que peut faire un guichetier ? un client ? le directeur ?

- De voir le système de différents points de vues
- De déterminer des droits d'accès par type d'acteur
- De fixer des ordres de priorité entre acteurs

## → Les cas d'utilisation

Un **cas d'utilisation** est un service rendu à l'utilisateur, il implique des séries d'actions plus élémentaires.

- Un **cas d'utilisation** est représenté par un ovale.
- Le nom du cas est inclus dans l'ellipse
- Il est relié par des associations « participe à » à ses acteurs



Cas d'utilisation

- **Un cas d'utilisation (use case) décrit:**

- Une fonctionnalité du système vue par un acteur (et utile à ce dernier)
- Une suite d'interactions entre un utilisateur et le système qui produit un résultat observable et intéressant pour un acteur particulier
- Un comportement attendu du système du point de vue d'un ou de plusieurs acteurs
- Un service rendu par le système

- **Pour les identifier:**

par acteur, quelles sont les différentes manières d'utiliser le système ?

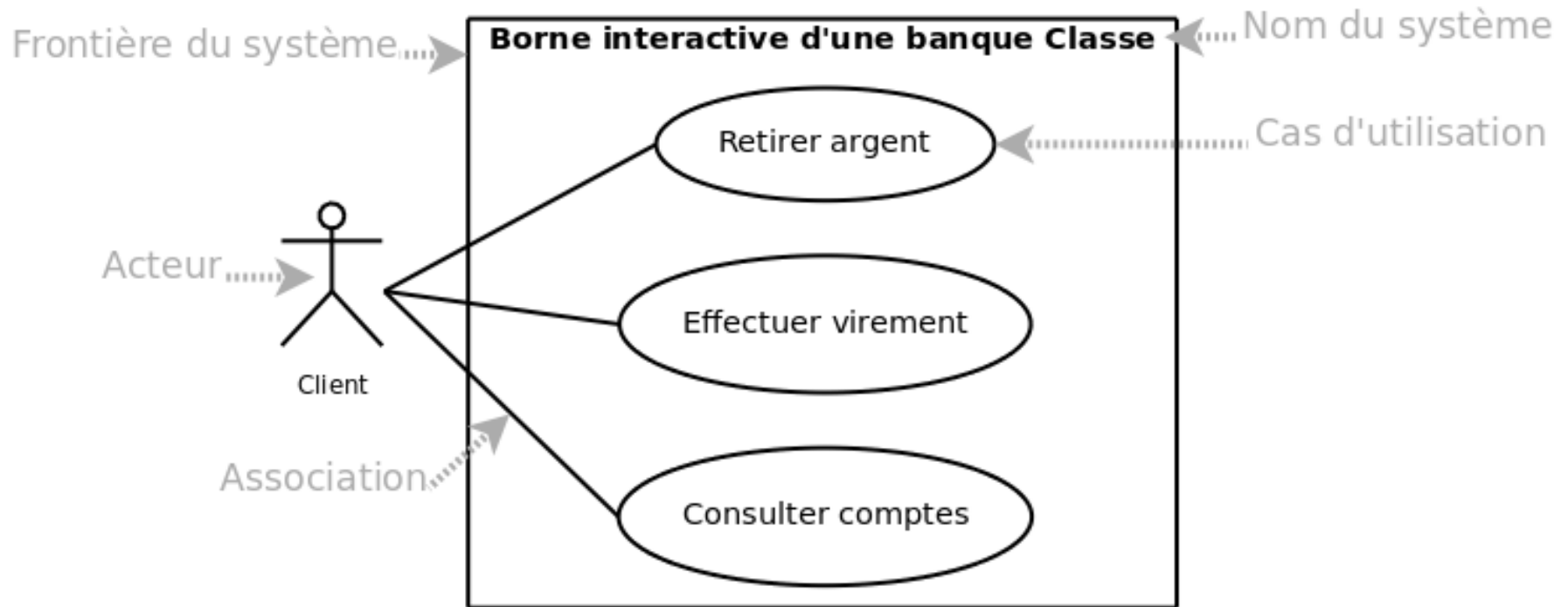


## → Les cas d'utilisation

### ❑ Recenser les cas d'utilisation

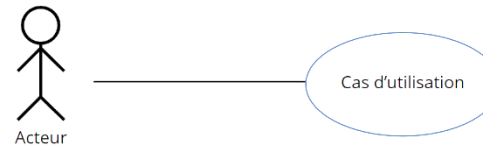
- Il n'y a pas une manière mécanique et totalement objective de repérer les cas d'utilisation.
- Il faut se placer du point de vue de chaque acteur et déterminer comment il se sert du système, dans quels cas il l'utilise, et à quelles fonctionnalités il doit avoir accès.
- Il faut éviter les redondances et limiter le nombre de cas en se situant au bon niveau d'abstraction (par exemple, ne pas réduire un cas à une seule action).
- Il ne faut pas faire apparaître les détails des cas d'utilisation, mais il faut rester au niveau des grandes fonctions du système.
- Trouver le bon niveau de détail pour les cas d'utilisation est un problème difficile qui nécessite de l'expérience.

## Exemple:



## → Relations entre cas d'utilisation et acteurs

- Les **acteurs** impliqués dans un **cas d'utilisation** lui sont liés par une **association**.
- Un **acteur** peut utiliser plusieurs fois le même **cas d'utilisation**.



## → Relations entre les cas d'utilisation

- ❑ **Inclusion** : le cas A **inclut** le cas B (B est une partie obligatoire de A)



- ❑ **Extension** : le cas B **étend** le cas A (A est une partie optionnelle de B)



- ❑ **Généralisation** : le cas A est une généralisation du cas B (B est une sorte de A).



## → Relations entre les cas d'utilisation

### ❑ Dépendances d'inclusion et d'extension

- Les **inclusions** et les **extensions** sont représentées par des dépendances.
  - Lorsqu'un cas B inclut un cas A, B dépend de A.
  - Lorsqu'un cas B étend un cas A, B dépend aussi de A.
  - On note toujours la dépendance par une flèche pointillée B - -> A qui se lit: **B dépend de A**.
- ❑ Lorsqu'un élément **A** dépend d'un élément **B**, toute modification de B sera susceptible d'avoir un impact sur A.
- ❑ Les ***include*** et les ***extend*** sont des stéréotypes (entre guillemets) des relations de dépendance.

### ***Attention***

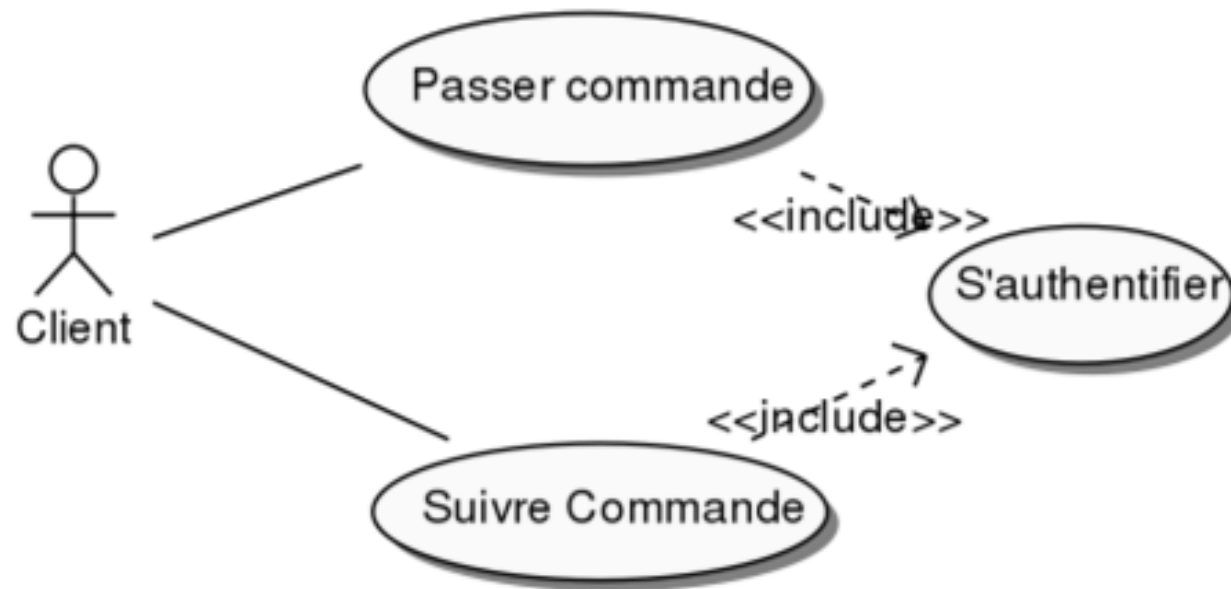
Le sens des flèches indique le dépendance, pas le sens de la relation d'inclusion.

## → Relations entre les cas d'utilisation

### ❑ Dépendances d'inclusion et d'extension

- Réutilisabilité avec les inclusions et les extensions

*Exemple:* Les relations entre cas permettent la réutilisabilité du cas « s'authentifier » : il sera inutile de développer plusieurs fois un module **d'authentification**

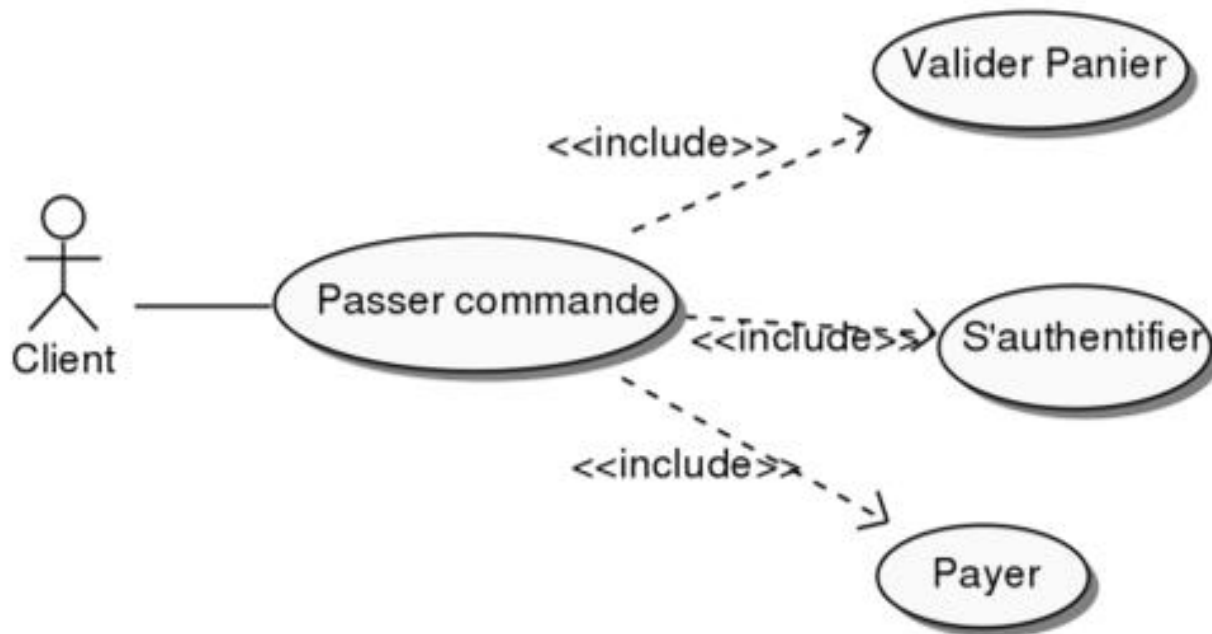


## → Relations entre les cas d'utilisation

### □ Dépendances d'inclusion et d'extension

- Décomposition grâce aux inclusions et aux extensions

Quand un cas est trop complexe (faisant intervenir un trop grand nombre d'actions), on peut procéder à **sa décomposition en cas plus simples**.



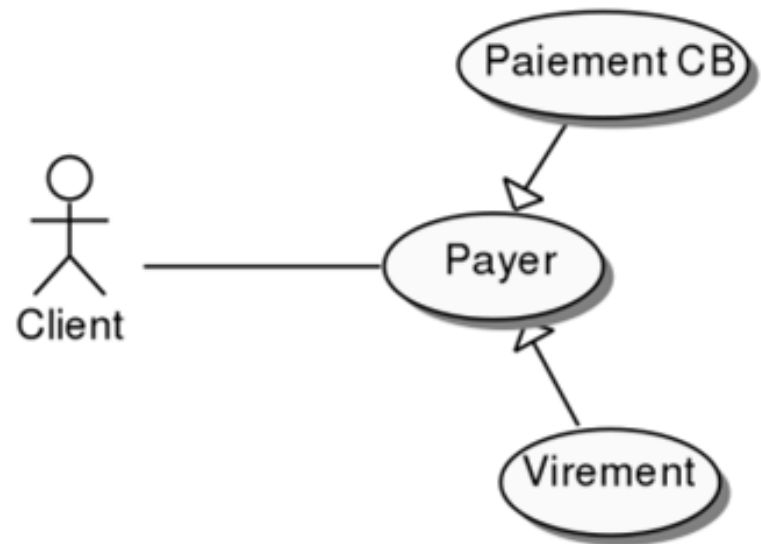
## → Relations entre les cas d'utilisation

### □ Généralisation

La relation de généralisation/spécialisation est présente dans la plupart des diagrammes UML et se traduit par le concept **d'héritage** dans les langages orientés objet

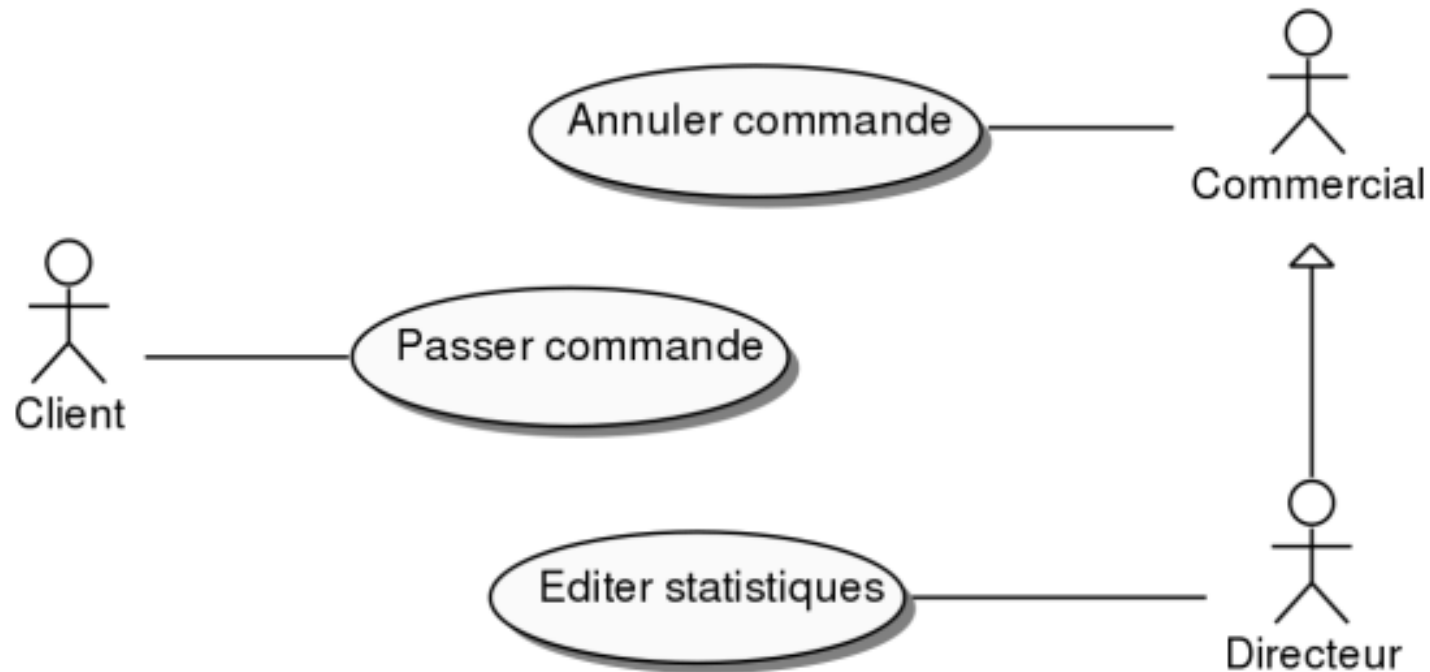
#### Exemple

- Un virement est un cas particulier de paiement.
- Un virement est une sorte de paiement.
- La flèche pointe vers l'élément général.



## → Relations entre les acteurs

- ❑ Une seule relation possible : **la généralisation**.





### → Acteurs principaux et secondaire

- ❑ L'acteur est considéré comme **principal** pour un cas d'utilisation lorsque c'est lui qui initie les échanges nécessaires pour réaliser ce cas d'utilisation.



- ❑ **Les acteurs secondaires** sont sollicités par le système, tandis que le plus souvent, ce sont les acteurs principaux qui prennent l'initiative des interactions.  
En général, les acteurs secondaires représentent d'autres systèmes informatiques avec lesquels le système développé est interconnecté.

## Description textuelle d'un cas d'utilisation

- *Le diagramme de cas d'utilisation* décrit les grandes fonctions d'un système du point de vue des acteurs, mais n'expose pas de façon détaillée le dialogue entre les acteurs et les cas d'utilisation.
- Un simple nom est tout à fait insuffisant pour décrire un cas d'utilisation.
- Chaque cas d'utilisation doit être documenté pour qu'il n'y ait aucune ambiguïté concernant son déroulement et ce qu'il recouvre précisément.

### ❑ Description textuelle

**Une description textuelle** d'un cas d'utilisation se compose de trois parties : **identification**, **description des scénarios** et **exigence non fonctionnelle**

# Description textuelle d'un cas d'utilisation

<b>Cas d'utilisation</b>	Nom de cas d'utilisation.
<b>Acteur principale</b>	.Les acteurs principales.
<b>Acteur secondaire</b>	Les acteurs secondaires.
<b>Objectif</b>	L'objectif du cas d'utilisation.
<b>Pré-condition</b>	Définissent ce qui doit être vrai en amont du cas d'utilisation pour que celui-ci puisse démarrer.
<b>Post-conduction</b>	Les post-conditions définissent ce qui doit être vrai lorsque le cas d'utilisation se termine.
<b>Scénario nominal</b>	Celui qui satisfait les objectifs des acteurs par le chemin le plus direct de succès.
<b>Scénario alternative</b>	Comprennent tous les autres scénarios, de succès (fin normale) ou d'échec (erreur).
<b>Exigences supplémentaires</b>	Il s'agira de <b>performance</b> , de <b>sécurité</b> ou d' <b>ergonomie</b> . On complètera par exemple la description des scénarios par des copies d'écran de la maquette

## Description textuelle d'un cas d'utilisation

### *Exemple 1* : Description textuelle de cas d'utilisation : 'Ajouter une commande'

<b>Acteur</b>	Client.
<b>Objectif</b>	Ajouter une commande.
<b>Pré condition</b>	le cas d'utilisation commence lorsque l'utilisateur est s'authentié.
<b>Post condition</b>	Le client passe une commande
<b>Scénario nominal</b>	<ol style="list-style-type: none"><li>1. Le client demander de remplir le formulaire de commande</li><li>2. Le système affiche le formulaire.</li><li>3. le client saisie ces informations nécessaire (nom, prénom, adresse, ...).</li><li>4. le système enregistre les informations , il renvoie un message de réussite.</li></ol>
<b>Scénario d'erreur (alternative)</b>	<p>4-a- Le client n'a pas remplit un champ obligatoire ou a saisie des informations non valides.</p> <p>Le système indique au client « le champ non accepté avec une couleur rouge et le scénario reprend à partir de 2 ».</p>
<b>Exigences supplémentaires</b>	/

## Description textuelle d'un cas d'utilisation

### Exemple 2 : Description textuelle de cas d'utilisation : 'S'authentifier'

N° 1	S'authentifier
Résumé	L'authentification permet d'accéder à des fonctionnalités réservées à un type d'utilisateur donné.
Acteurs	Tous les acteurs
Pré-conditions	Application accessible
Scénario nominal	<ol style="list-style-type: none"><li>1. Le système affiche le formulaire d'authentification</li><li>2. L'utilisateur saisit son login et son mot de passe.</li><li>3. Le système vérifie la conformité des informations</li><li>4. Le système donne l'accès à l'interface correspondante.</li></ol>
Scénario alternatif	Les informations fournies sont incorrectes ; Le système réaffiche le formulaire d'authentification et attend que l'utilisateur ressaisisse ses informations.
Scénario d'exception	Compte inexistant.

# Description textuelle d'un cas d'utilisation

## Exemple 3 : Description textuelle de C.U : Payer CB

### ○ Identification :

- *Nom du cas* : Payer CB
- *Objectif* : Détailler les étapes permettant à client de payer par carte bancaire
- *Acteurs* : Client, Banque (secondaire)
- *Date* : 18/09
- *Responsables* : Toto
- *Version* : 1.0

### ○ Séquencements :

- Le cas d'utilisation commence lorsqu'un client demande le paiement par carte bancaire
- **Pré-conditions**
  - .. Le client a validé sa commande
- **Enchaînement nominal**
  1. Le client saisit les informations de sa CB
  2. Le système vérifie que le numéro de CB est correct
  3. Le système vérifie la carte auprès du Sys.Ban
  4. Le système demande au système bancaire de débiter le client
  5. Le système notifie le client du bon déroulement de la transaction

### - Enchaînements alternatifs

1. En (2) : si le numéro est incorrect, le client est averti de l'erreur, et invité à recommencer
2. En (3) : si les informations sont erronées, elles sont re-demandées au client

### - Post-conditions

- .. La commande est validée
- .. Le compte de l'entreprise est crédité

### ○ Rubriques optionnelles

#### - Contraintes non fonctionnelles :

- .. Fiabilité : les accès doivent être sécurisés
- .. Confidentialité

#### - Contraintes liées à l'interface homme-machine :

- .. Toujours demander la validation des opérations bancaires

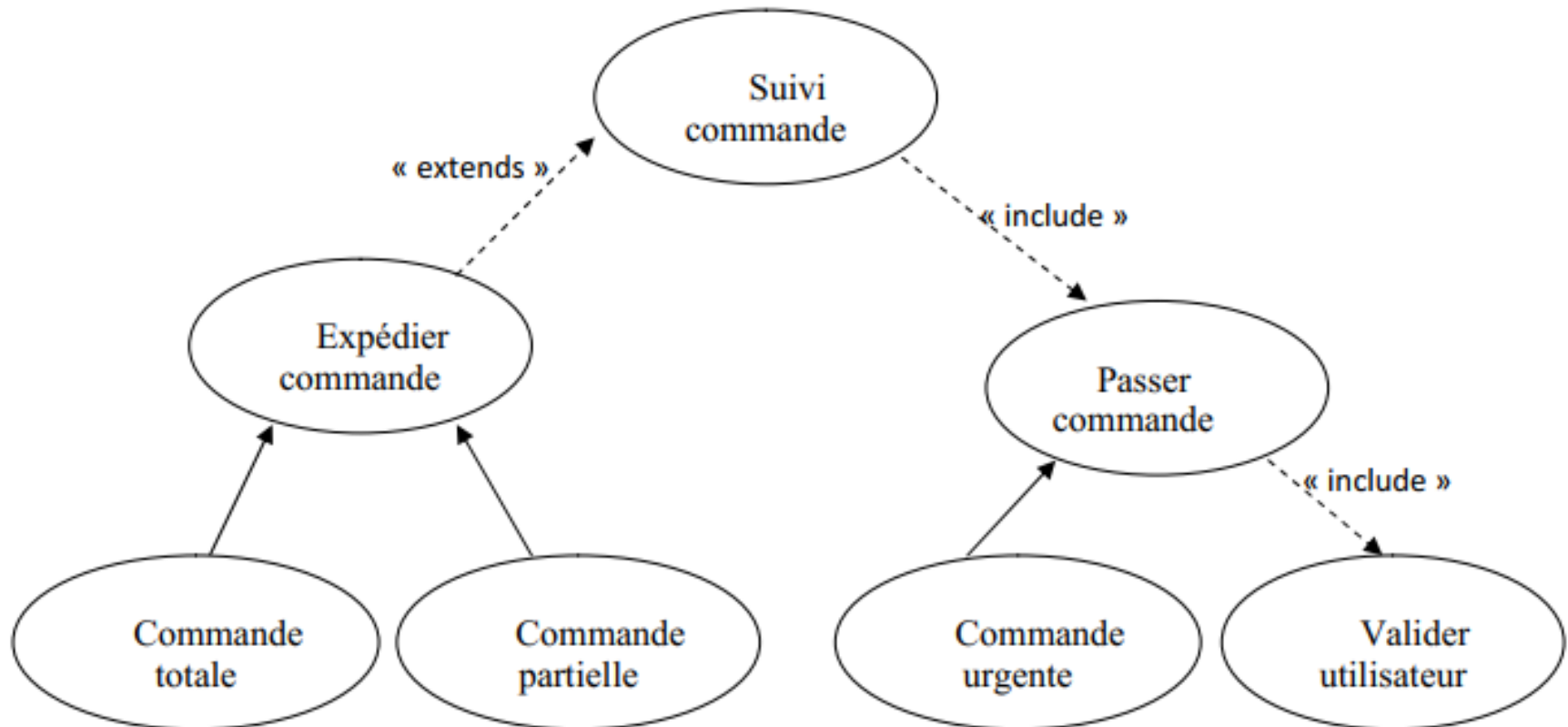
## Exercice 1

Nous disposons des cas d'utilisation suivants :

1. Passer une commande
  2. Passer une commande urgente
  3. Suivre une commande
  4. Valider l'utilisateur
  5. Expédier commande totale ou partielle
- Le suivi de la commande englobe l'ensemble du processus, depuis la commande initiale jusqu'à l'expédition.
  - Cependant, il peut arriver que certaines commandes passées ne soient pas expédiées.
  - Il convient de noter que passer une commande urgente est un cas particulier de passer une commande.
  - Pour passer une commande, il est impératif de valider l'utilisateur.

Présentez **les cas d'utilisation** et **identifiez leurs relations de dépendance**.

## Exercice 1 - Correction





## **Exercice 2**

Un comptable est chargé du traitement des factures d'une entreprise. Chaque fois qu'il traite une facture, il doit effectuer un calcul de remise, même si la remise est de 0%. De plus, les factures étrangères nécessitent un traitement spécial.

**Donner un diagramme de cas d'utilisation UML qui représente cette situation en identifiant les acteurs et les cas d'utilisation principaux, ainsi que leurs interactions**

## Exercice 2 - Correction

