

UML : diagramme de cas d'utilisation



Unified
Modeling
Language

RAFII ZAKANI FATIMA
frafiizakani@gmail.com

L'Analyse des besoins avec UML

- L'analyse des besoins joue un rôle crucial dans le développement de systèmes informatiques, en ciblant avec précision les exigences et les interactions entre les utilisateurs et le système.
- L'UML avec sa vaste gamme de modèles, se révèle indispensable pour cerner ces besoins, améliorant ainsi leur compréhension pour toutes les parties prenantes.
- L'un des modèles les plus importants de cette phase on trouve le diagramme de contexte qui offre une vue d'ensemble des échanges d'informations entre le système et son environnement.
- En parallèle on trouve aussi le diagramme de cas d'utilisation qui vient compléter cette analyse en détaillant les interactions entre les utilisateurs et le système.

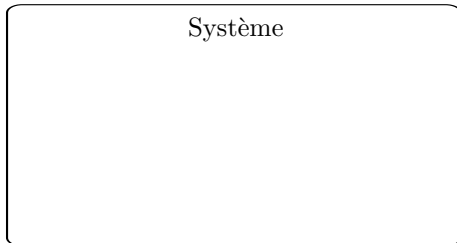
Diagramme de contexte

- Dans le processus d'analyse, il se situe au début constituant une étape intermédiaire entre le cahier des charges et la construction des premiers cas d'utilisation.
- Il n'est pas indispensable. D'ailleurs, il ne correspond pas aux 14 diagrammes de la norme UML 2.5
- Permettant de définir :
 - le système à modéliser
 - les différents acteurs qui vont interagir avec le système

Système ?

Au début de la phase d'analyse le système sera considéré comme une boîte noire et nous nous concentrons que sur les relations entre les acteurs et le système

Comment représenter un système ?



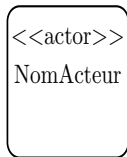
Acteur ?

- Un acteur est une entité externe qui interagit avec le système
- Un acteur est un rôle joué par l'utilisateur dans le système.
- Un acteur peut être un humain ou un système automatisé.
- Les acteurs se déterminent en observant les utilisateurs directs du système, ceux qui sont responsables de son exploitation, ainsi que les autres systèmes qui interagissent avec lui
- Il est important de distinguer les différents acteurs impliqués dans un système pour déterminer les limites du système.
- Il existe deux types d'acteurs dans un diagramme de contexte statique / cas d'utilisation : les acteurs primaires et les acteurs secondaires.

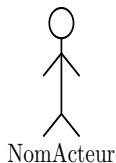
Acteur ?

- Les acteurs en UML se représentent sous la forme d'un petit personnage (stickman) ou sous la forme d'une case rectangulaire (appelé classeur) avec le mot clé « actor ».
- Chaque acteur porte un nom.
- Les acteurs se mettent obligatoirement à l'extérieur du système.

Deux représentations possibles d'un acteur en UML



En utilisant un classeur



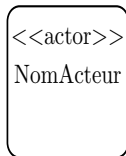
En utilisant un bonhomme

UML

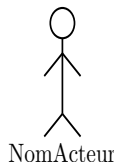
Acteur ?

- Les acteurs en UML se représentent sous la forme d'un petit personnage (stickman) ou sous la forme d'une case rectangulaire (appelé classeur) avec le mot clé « actor ».
- Chaque acteur porte un nom.
- Les acteurs se mettent obligatoirement à l'extérieur du système.

Deux représentations possibles d'un acteur en UML



En utilisant un classeur



En utilisant un bonhomme

Pas besoin d'un stéréotype `<<actor>>` *dans la 2eme représentation*

Attention !

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

- Une même personne physique peut être représenté 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.
- Un acteur peut être un humain ou un système automatisé.

Deux types d'acteurs en UML : Acteur principal

- L'acteur principal est l'entité qui initie une interaction avec le système afin de réaliser un objectif et c'est celui qui demande au système de réaliser un service
- Il présente un intérêt direct avec le système et est habituellement le déclencheur des cas d'utilisation.
- Il est placé à l'extérieur mais à proximité du système dans les diagrammes de contexte et de cas d'utilisation, illustrant son interaction directe avec le système.
- L'objectif principal de l'identification des acteurs principaux est de s'assurer que le système est conçu pour répondre à leurs besoins et exigences. Cela aide à définir clairement les frontières du système et à identifier les fonctionnalités nécessaires pour satisfaire les utilisateurs ou les entités qui interagissent avec le système.

Exemples d'acteurs Principaux

- Dans un système de gestion de bibliothèque, un acteur primaire pourrait être un lecteur (personne) qui cherche à emprunter un livre.
- Dans un système de commerce électronique, cela pourrait être un client (personne) qui souhaite acheter un produit.
- Pour un système de gestion des ressources humaines, cela pourrait être un employé (personne) soumettant une demande de congé.

Deux types d'acteurs en UML : Acteur secondaire

- L'acteur secondaire est l'entité qui interagit avec le système, mais ne déclenche pas directement les cas d'utilisation.
- Il a une relation indirecte avec le système et joue un rôle de soutien dans les cas d'utilisation, souvent en fournissant des services ou des informations nécessaires à l'accomplissement des cas d'utilisations.
- Dans les diagrammes de contexte et de cas d'utilisation, il est également positionné à l'extérieur du système, soulignant son interaction avec le système.
- L'identification des acteurs secondaires aide à assurer que le système est conçu pour interagir efficacement avec toutes les parties prenantes, garantissant une prise en main complète des exigences pour le fonctionnement et la réussite du système. Elle permet également de clarifier les responsabilités et les interactions entre le système et son environnement externe.

Représentation d'acteurs secondaires en UML

On utilise le stéréotype `<<secondary>>` sur les fonctionnalités liées aux acteurs secondaires

Exemple : considérons un système d'achat et vente en ligne

Considérons un système d'achat et de vente en ligne :

- Le client peut gérer son panier, il peut passer une commande, il peut payer sa commande en ligne et il peut commenter des articles qui a déjà acheté
- L'administrateur peut gérer les commandes des clients
- Le vendeur il peut gérer les articles qu'il met en vente

Exemple : considérons un système d'achat et vente en ligne

Les différents acteurs possibles sont

- Client
- Vendeur
- Administrateur
- Banque

Exemple : considérons un système d'achat et vente en ligne

Les différents acteurs possibles sont

- Client
- Vendeur
- Administrateur
- Banque



Client



Vendeur



Administrateur



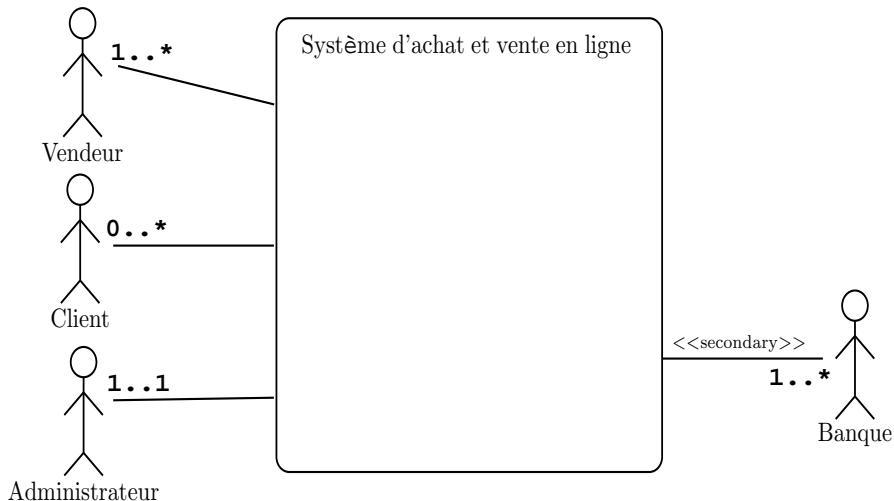
Banque

Acteurs principaux et secondaires du système de vente en ligne

- Tous les acteurs sauf la banque sont des acteurs principaux
- Tous les acteurs ont un besoin / une relation direct avec notre système
- La banque est un acteur secondaire (sollicitée par notre système pour valider le paiement)
- L'acteur banque n'a pas un besoin direct avec notre application, c'est notre système qui fait appelle a l'acteur banque

UML

Le diagramme de contexte statique d'un système d'achat et vente en ligne



Le diagramme cas d'utilisation

- Un diagramme dynamique d'UML
- Parmi les diagrammes les plus importants
- Le diagramme de "Use Case" ou "Cas d'utilisation" est utilisé dans l'activité d'analyse des besoins

Le diagramme cas d'utilisation

Il est utilisé pour :

- Recueillir, analyser et organiser les besoins
- Recenser les fonctionnalités d'un système
 - Il permet de décrire ce que le futur système devra faire en terme de fonctionnalité, sans spécifier comment il le fera.
 - Il permet de représenter les fonctions du système du point de vue des utilisateurs.
 - Il permet de clarifier les relations entre les acteurs et les cas d'utilisation

Le diagramme Use Case doit permettre de répondre à la question
Qui fait quoi?

Le diagramme cas d'utilisation

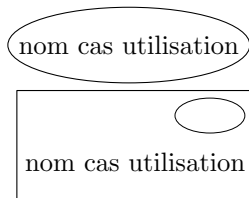
- Le cas d'utilisation représente une fonctionnalité du système.
- Un cas d'utilisation se représente par une ellipse contenant le nom du cas d'utilisation (phrase commençant par un verbe à l'infinitif) .
- Les cas d'utilisation peuvent éventuellement être représentés par un rectangle doté d'un pictogramme d'ellipse

UML

Le diagramme cas d'utilisation

- Le cas d'utilisation représente une fonctionnalité du système.
- Un cas d'utilisation se représente par une ellipse contenant le nom du cas d'utilisation (phrase commençant par un verbe à l'infinitif) .
- Les cas d'utilisation peuvent éventuellement être représentés par un rectangle doté d'un pictogramme d'ellipse

Comment visualiser un cas d'utilisation en UML ?



Le diagramme cas d'utilisation

- Les besoins des utilisateurs sont modélisés dans le diagramme de cas d'utilisation
- Les diagrammes de cas d'utilisation permettent d'identifier les futurs fonctionnalités du système et ainsi ses limites
- Représentant les interactions entre le système et ses utilisateurs.
- Un diagramme de cas d'utilisation général (Jacobson conseille de ne pas dépasser 25 cas d'utilisation par diagramme)
- Un diagramme de cas d'utilisation détaillé pour(chaque) acteur

UML

Le diagramme de cas d'utilisation se compose en principe de deux éléments :



Acteurs



Cas d'utilisations

Mots-clés associés

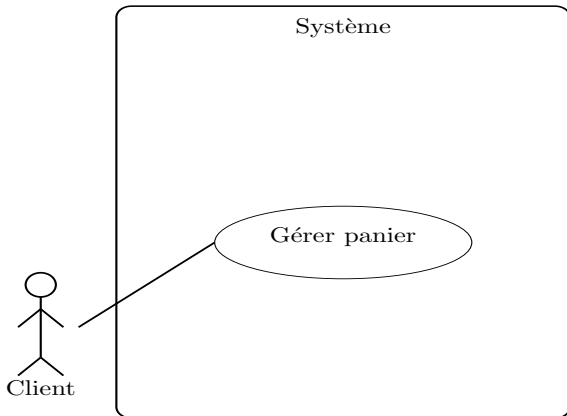
- acteur
- cas d'utilisation
- dépendance
- héritage

Représentation des acteurs dans le diagramme de cas d'utilisation

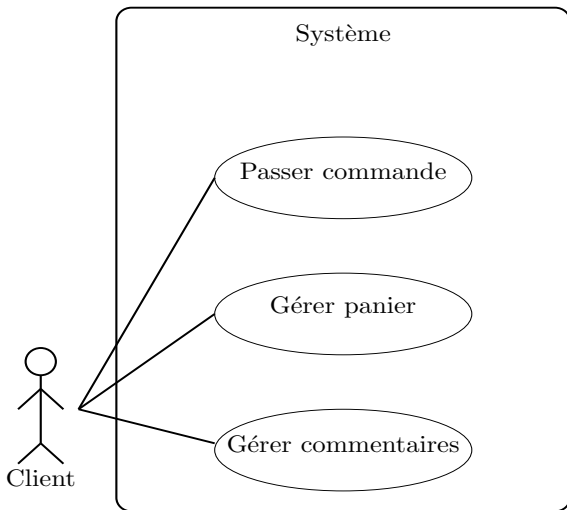
- Chaque acteur est associé un ou plusieurs cas d'utilisations.
- Les acteurs principaux sont placés à gauche et les acteurs secondaires à droite du système.

Diagramme de cas d'utilisation

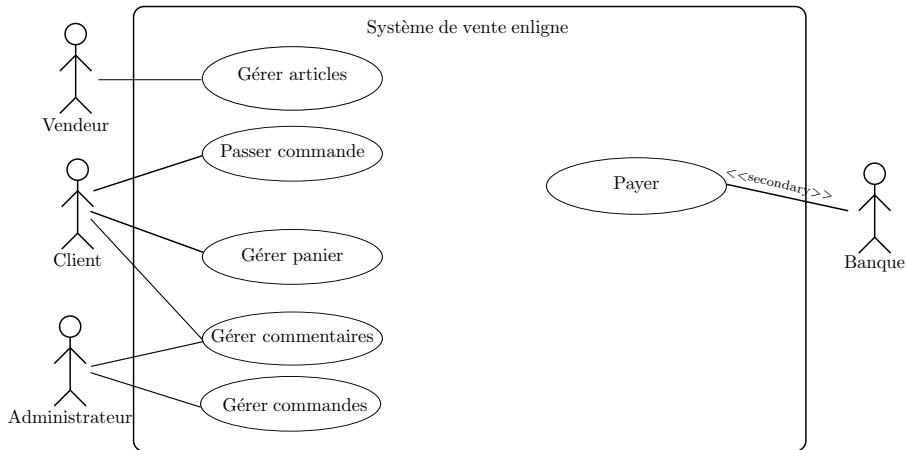
= contexte + acteurs + cas d'utilisation + associations



Un acteur peut être en relation avec plusieurs cas d'utilisation



Ajoutons les cas d'utilisation de tous les acteurs de notre exemple précédent



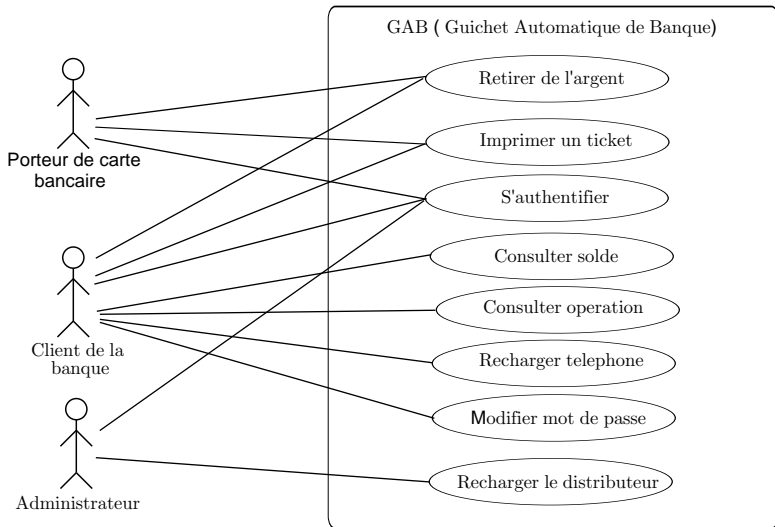
Exemple

L'étude de cas concerne un système simplifié de GAB (Guichet Automatique de Banque)

Le GAB offre les services suivants :

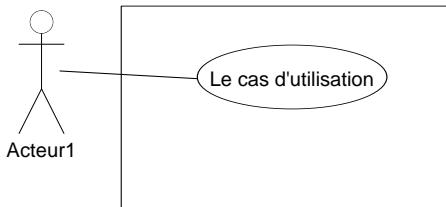
1. Distribution d'argent à tout porteur de Carte Bancaire Via un lecteur de carte. Après chaque transaction de retrait d'argent, le système offre la possibilité d'imprimer un ticket récapitulant les détails de la transaction pour le porteur de la carte.
2. Consultation de solde de compte, consultation des opérations, modification du mot de passe, recharge de tel pour les clients de la banque porteurs d'une carte de crédit de la banque.
3. Les transactions sont sécurisées via une authentification
4. Il est parfois nécessaire de recharger le distributeur par les administrateur.

Solution proposée



Relation acteur-cas d'utilisation

- Dans le diagramme de cas d'utilisation le système est modélisé sous forme de rectangle comprenant les cas d'utilisation et un acteur est souvent associé à plusieurs cas d'utilisation
- Une ligne entre un acteur et un cas d'utilisation signifie qu'une communication est établie. Elle est modélisée sous forme d'association en UML.



Relation cas d'utilisation-cas d'utilisation

Dans un diagramme de cas d'utilisation il arrive que l'on trouve que certains cas d'utilisation nécessite l'appel à d'autre cas d'utilisation pendant leurs exécution.

- **Généralisation/Spécialisation** : principe d'héritage entre les cas d'utilisation
- **L'inclusion («include»)** : La relation "include" est une relation de dépendance entre deux cas d'utilisation. Elle permet de décrire une situation où un cas d'utilisation est inclus dans un autre cas d'utilisation en tant que sous-étape ou sous-fonctionnalité.
- **L'extension («extend»)** : le comportement d'un cas d'utilisation peut être étendu par un autre cas d'utilisation (avec condition éventuelle)

La relation "include"

- L'inclusion est une relation entre deux cas d'utilisation qui indique que le comportement d'un cas d'utilisation (appelé cas d'utilisation inclus) est inclut dans le comportement d'un autre cas d'utilisation (appelé cas d'utilisation de base).
- Le cas d'utilisation inclus est exécuté à chaque fois que le cas d'utilisation de base est exécuté.
- Le cas d'utilisation inclus peut être utilisé dans plusieurs cas d'utilisation de base, ce qui permet de réduire la duplication de comportement et de faciliter la maintenance du système.
- La relation d'inclusion peut être exploiter pour réutiliser le comportement commun entre plusieurs cas d'utilisation.

La relation "include"

- **Cas d'utilisation de base** : C'est le cas d'utilisation qui inclut un autre cas d'utilisation dans son flux d'exécution. Il dépend du cas d'utilisation inclus pour compléter une partie de son comportement.
- **Cas d'utilisation inclus** : C'est le cas d'utilisation qui est inclus par un autre. Il représente un ensemble d'actions qui sont extraites du cas d'utilisation de base et qui peuvent être partagées par plusieurs cas d'utilisation.

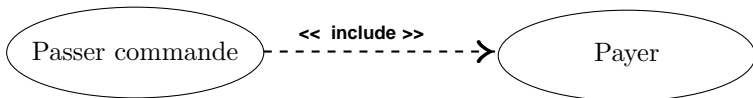
UML

La relation "include"

Si le cas **B** inclut le cas **A**, alors **A** est une partie obligatoire de **B**.

La relation "include"

- Dans un diagramme de cas d'utilisation, la relation d'inclusion est représentée par une flèche pointillée reliant les 2 cas d'utilisation et munie du stéréotype "include"



Quelques exemples :

Application bancaire :

- Le cas d'utilisation de base est "Effectuer un virement". Ce cas d'utilisation pourrait inclure le cas d'utilisation "Authentifier l'adhérent", qui donne le droit à l'adhérent d'effectuer le virement une fois authentifié.

Système de réservation dans un restaurant :

- Le cas d'utilisation de base est "Réserver une table". Ce cas d'utilisation pourrait inclure le cas d'utilisation "Vérifier les disponibilités", qui vérifie que la table demandée est disponible avant de permettre à l'utilisateur de la réserver.

Système de gestion de bibliothèque :

- Soit les deux cas d'utilisation : "Emprunter un livre" et "Renouveler un emprunt". Les deux pourraient inclure un cas d'utilisation commun "Vérifier la disponibilité", qui vérifie si un livre est disponible pour l'emprunt ou le renouvellement.

La relation "extend"

- La relation d'extension (extend) est utilisée pour décrire une relation optionnelle entre deux cas d'utilisation.
- Elle permet de décrire un cas d'utilisation optionnel qui peut étendre le comportement d'un cas d'utilisation de base.
- L'extension peut être activée ou non en fonction de certaines conditions.
- Les conditions indiquent les circonstances dans lesquelles le cas d'utilisation étendu est ajouté au cas d'utilisation de base.
- L'utilisation de la relation "extend" est généralement recommandée lorsque les fonctionnalités supplémentaires ne sont nécessaires que dans certaines conditions, car elle permet de maintenir la simplicité et la clarté du cas d'utilisation de base.

La relation "extend"

- **Cas d'utilisation de base** : C'est le cas d'utilisation qui peut être étendu par un autre cas d'utilisation. Son flux d'exécution peut être complété par l'ajout de comportements supplémentaires fournis par le cas d'utilisation étendu.
- **Cas d'utilisation étendu** : C'est le cas d'utilisation qui étend le comportement d'un autre cas d'utilisation. Il représente un ensemble d'actions qui sont conditionnellement exécutées pour enrichir ou ajuster le comportement du cas d'utilisation de base.

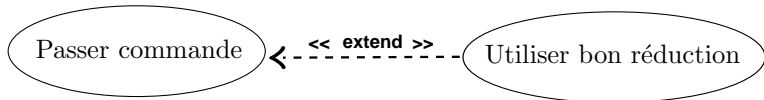
UML

La relation "extend"

Si le cas **B** étend le cas **A**, alors **B** est une partie optionnelle de **A**.

La relation "extend"

Cette relation est représentée par une flèche en pointillée reliant les 2 cas d'utilisations et munie du stéréotype "**extend**"



Quelques exemples :

Système de réservation d'hôtel :

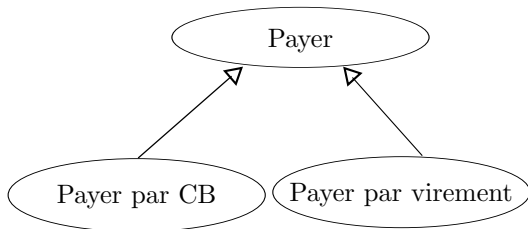
- Le cas d'utilisation de base "Réserver une chambre" peut être étendu par le cas d'utilisation "Ajouter options supplémentaires" pour que l'utilisateur puisse choisir des options supplémentaires telles que le petit-déjeuner ou le service en chambre.

Système de vente en ligne :

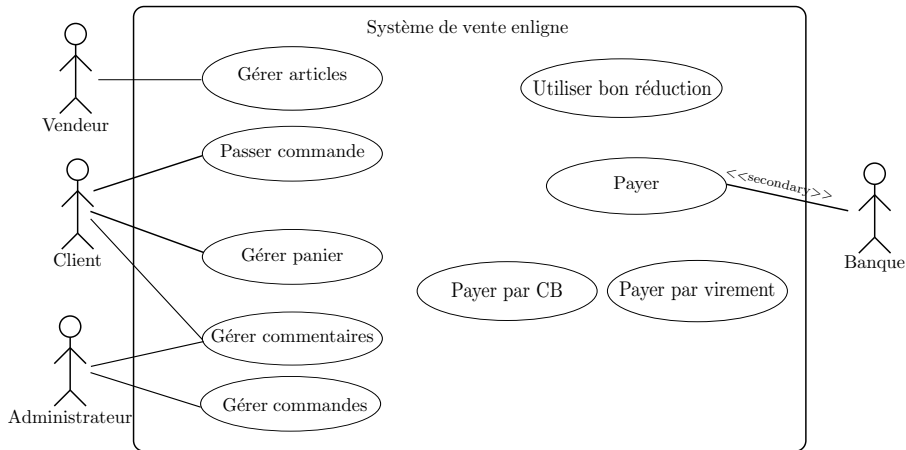
- Le cas d'utilisation de base "Passer une commande" peut être étendu par le cas d'utilisation "Appliquer un bon de réduction" pour permettre à l'utilisateur d'appliquer un bon de réduction avant de finaliser sa commande

La relation d'héritage

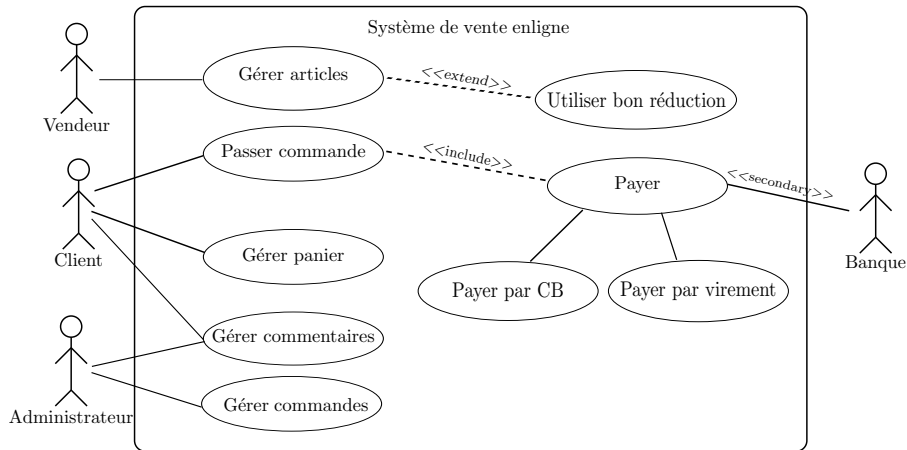
- La relation d'héritage dans un diagramme cas d'utilisation indique la généralisation / spécialisation.
- Dans un diagramme de cas d'utilisation on peut spécialiser un cas d'utilisation en un autre cas d'utilisation. Nous obtenons alors un sous-cas d'utilisation.
- La relation de généralisation est représentée par une flèche avec une extrémité triangulaire



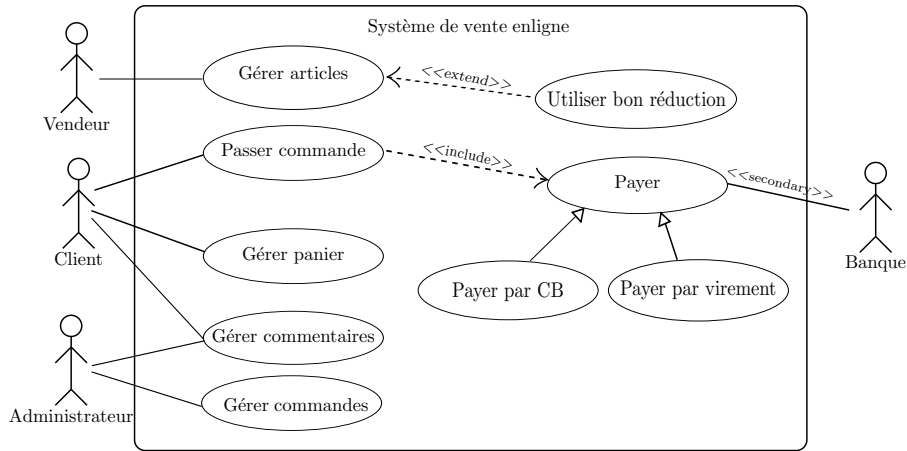
Identifiant les relations qui existe dans notre diagramme de cas d'utilisation



Le diagramme de cas d'utilisation de notre exemple



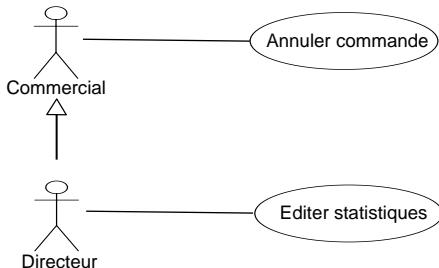
Le diagramme de cas d'utilisation de notre exemple



Pas d'ordre chronologique dans un diagramme de cas d'utilisation.

Relation acteur-acteur

- Une seule relation possible : la généralisation/spécialisation
- Un acteur **A** hérite d'un acteur **B** si l'acteur **A** peut faire tout ce que l'acteur **B** a le droit de faire
- Tous les cas d'utilisation accessibles à B le sont aussi à A (la réciproque est fausse)



Comment construire un diagramme de cas d'utilisation ?

- Identifier les acteurs et les classifier
- Trouver les cas d'utilisation de chaque acteur
- Établir les relations entre les cas d'utilisation et les acteurs
- Identifier les relations de dépendance entre les cas d'utilisation
- Vérifier s'il est possible de simplifier le diagramme avec la relation d'héritage
- Identifier les cas d'utilisation les plus complexes pour les détailler avec les diagrammes de séquence et d'activité

Revenant a cet exemple, comment peut on l'optimiser ?

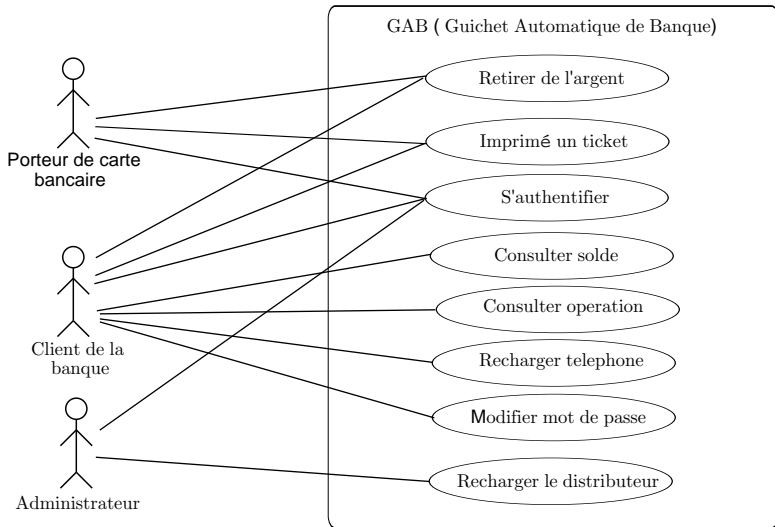
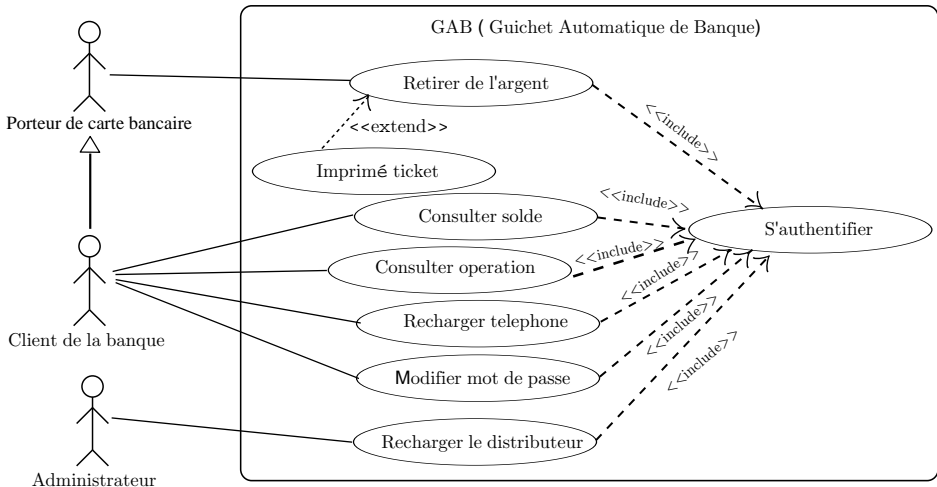
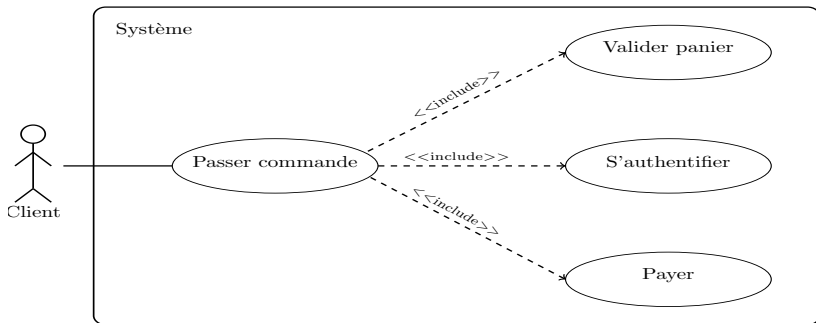


Diagramme de cas d'utilisation du GAB



Résumé

- Un logiciel à développer est utilisé par des acteurs principaux, et par fois, il peut être lié à des acteurs secondaires.
- Les acteurs secondaires échangent des informations avec le système, mais ne déclenchent aucune fonctionnalités.
- Les fonctionnalités utiles aux acteurs sont appelées des cas d'utilisation.
- Un diagramme de cas d'utilisation permet d'illustrer QUI devrait pouvoir faire QUOI, grâce au système.
- Les liens entre cas d'utilisation peuvent être : d'inclusion (Impérative et systématique) ou d'extension (Optionnel).
- Une relation de généralisation peut être présente entre deux cas d'utilisation ou entre les acteurs.



Exercice 1

Le système permet-il au client de payer ses commandes ?

Le système permet-il au client de suivre ses commandes ?

Le client est-il obligé de valider son panier pour payer ?

Le client est-il obligé de valider son panier pour passer sa commande ?

Exercice d'application

Nous souhaitons mettre en place une gestion automatisée pour une école privée. Cette école se compose principalement d'enseignants, d'étudiants, ainsi que d'un administrateur. Les enseignants ont la possibilité de réserver des salles pour leurs cours ou travaux pratiques (TP), nécessitant au préalable une vérification de leur disponibilité.

L'école crée également des emplois du temps accessibles tant aux enseignants qu'aux étudiants. Chaque enseignant peut visualiser son volume horaire. Celui-ci peut être modifié par un enseignant responsable, mais son accès reste exclusif aux enseignants.

Chaque fonctionnalité du système exige une authentification préalable. Il est demandé de modéliser ces besoins à travers un diagramme de cas d'utilisation.

Définition des acteurs du système

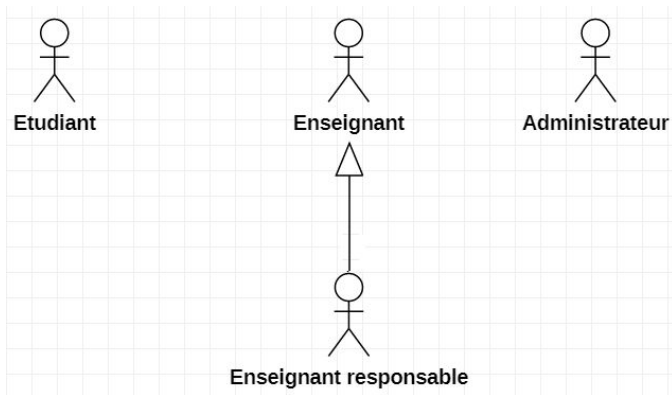


Diagramme de cas d'utilisation

