

# DOSSIER DE CONCEPTION FONCTIONNELLE

## OC PIZZA



**Auteur**  
Elodie Meunier  
Développeur



1. Versions .....	3
2. Introduction.....	4
1. Objet du document .....	4
2. Références .....	4
3. Besoin du client .....	5
1. Contexte .....	5
2. Enjeux et objectifs .....	5
3. Description générale de la solution .....	6
1. Les acteurs .....	6
2. Les principes de fonctionnements.....	7
3. Les cas d'utilisation généraux .....	8
4. Le domaine fonctionnel .....	10
1. Explication .....	10
2. Diagramme de classe .....	11
5. Les workflows .....	12
1. Le workflow du cycle de vie d'une commande (format textuel) .....	12
2. Le workflow du cycle de vie d'une commande (format diagramme) .....	13
6. Les fiches descriptives .....	14
1. C'est quoi ? .....	14
2. Package interface « Frontend » .....	15
1. Cas d'utilisation « Consulte le menu » .....	16
2. Cas d'utilisation « Créé un compte » .....	18
3. Cas d'utilisation « Enregistre une commande » .....	20
4. Cas d'utilisation « Modifie une commande » .....	22
3. Package interface « Backend » .....	23
1. Cas d'utilisation « Réceptionne la commande » .....	24
2. Cas d'utilisation « Consulte le stock en temps réel » .....	26
3. Cas d'utilisation « Prépare la commande » .....	28
4. Cas d'utilisation « Livre » .....	30
5. Cas d'utilisation « Gère les points de vente » .....	32
4. Package « Authentification » .....	34
1. Cas d'utilisation « Se connecte » .....	35

# 1. VERSIONS

Auteur	Date	Description	Version
Elodie	09/02/2021	Création du document	1

## 2. INTRODUCTION

### 2.1 OBJET DU DOCUMENT

---

Le présent document constitue le dossier de conception fonctionnelle de l'application web **OC Pizza** développée par l'entreprise **EdenConception**.

### 2.2 RÉFÉRENCES

---

Pour de plus amples informations, se référer aux éléments suivants :

1. DCT : Dossier de conception technique de l'application
2. DE : Dossier d'exploitation de l'application

### 2.3 BESOIN DU CLIENT

#### 2.3.1 Contexte

« OC PIZZA » est un jeune groupe de pizzeria en plein essor, créé par Franck et Lola, le groupe est spécialisé dans les pizzas livrées ou à emporter. Il compte déjà 5 points de vente et prévoit d'en ouvrir au moins 3 de plus d'ici 6 mois. Le système informatique actuel ne correspond plus aux besoins du groupe car il ne permet pas une gestion centralisée de toutes les pizzerias. De plus, il est très difficile pour les responsables de suivre ce qui se passe dans les points de vente. Enfin, les livreurs ne peuvent pas indiquer « en live » que la livraison est effectuée.

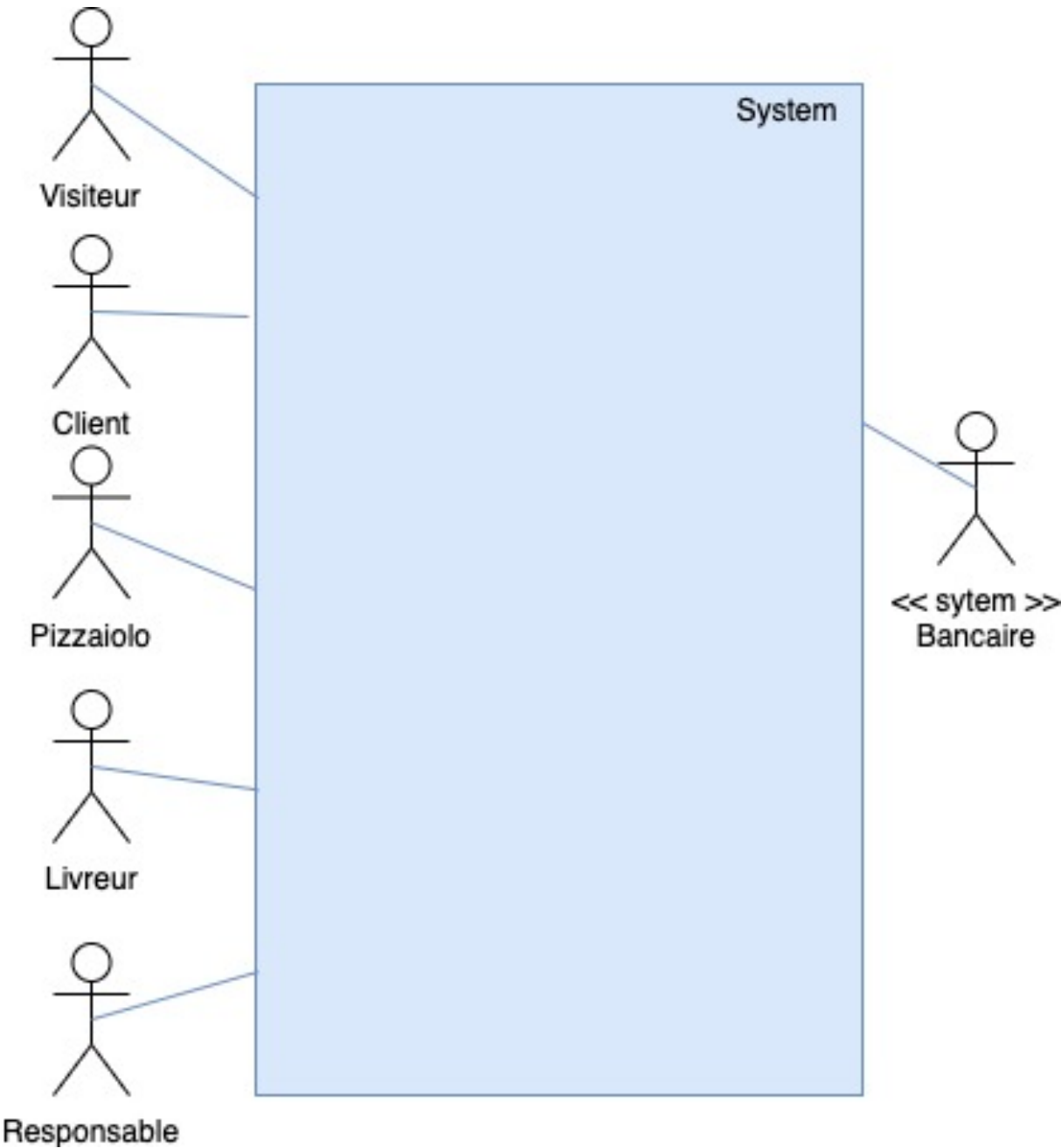


OC Pizza - Point de vente de Paris

#### 2.3.2 Enjeux et objectifs

- Être plus efficace dans la gestion des commandes, de leur réception à leur livraison en passant par leur préparation;
- Suivre en temps réel le stock d'ingrédients restants pour savoir quelles pizzas peuvent encore être réalisées;
- Proposer un site internet pour que les clients puissent passer leurs commandes, payer en ligne, modifier ou annuler leur commande;
- Permettre au pizzaiolo de visualiser la recette d'une pizza

3.1 LES ACTEURS

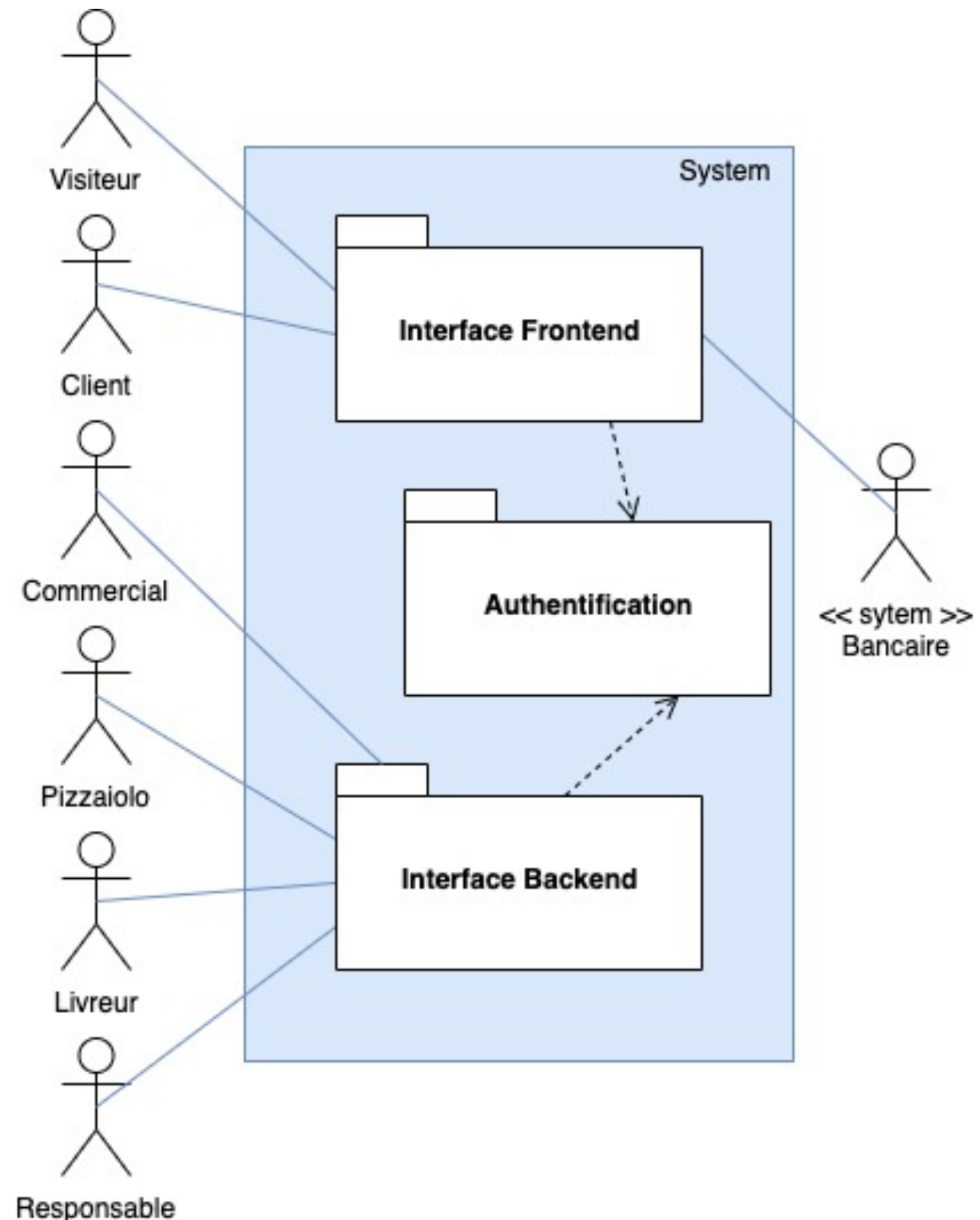


Acteur	Type	Actions
Visiteur	Principal	- Consulte le menu
Client	Principal	- Consulte le menu - Constitue son panier - Paye en ligne ou à la livraison - Modifie sa commande
Commercial	Principal	- Réceptionne la commande - Transmet la commande au pizzaiolo - Consulte les stocks - Change l'état de la commande (en cours)
Pizzaiolo	Principal	- Prépare la commande - Change l'état de la commande (en préparation)
Livreur	Principal	- Livre la commande - Change l'état de la commande (livré)
Responsable	Principal	- Gère les points de vente
« Système » Bancaire	Secondaire	- Traite le paiement en ligne

## 3.2 LES PRINCIPES DE FONCTIONNEMENT

Retrouvez un diagramme de package permettant de décomposer le système en parties distinctes.

- Nous retrouvons un package « **Interface Frontend** » qui permet d'identifier la partie visible par le client : Le site web. Les acteurs concernés seront donc le « client », le « visiteur » et le « système bancaire ».
- Nous avons également un package « **Interface Backend** » permettant quant à lui, de cibler la partie utilisée par l'équipe « OC PIZZA ». Les acteurs concernés seront donc le « pizzaiolo », le « livreur », le « commercial » et le ou les « responsable(s) ».
- Enfin, nous avons un package « **Authentification** » qui indique que chaque package aura besoin de celui-ci. Il s'agira de s'identifier afin que chaque acteurs puissent exécuter les actions voulus. Seul l'acteur « visiteur » ne nécessitera pas d'authentification.

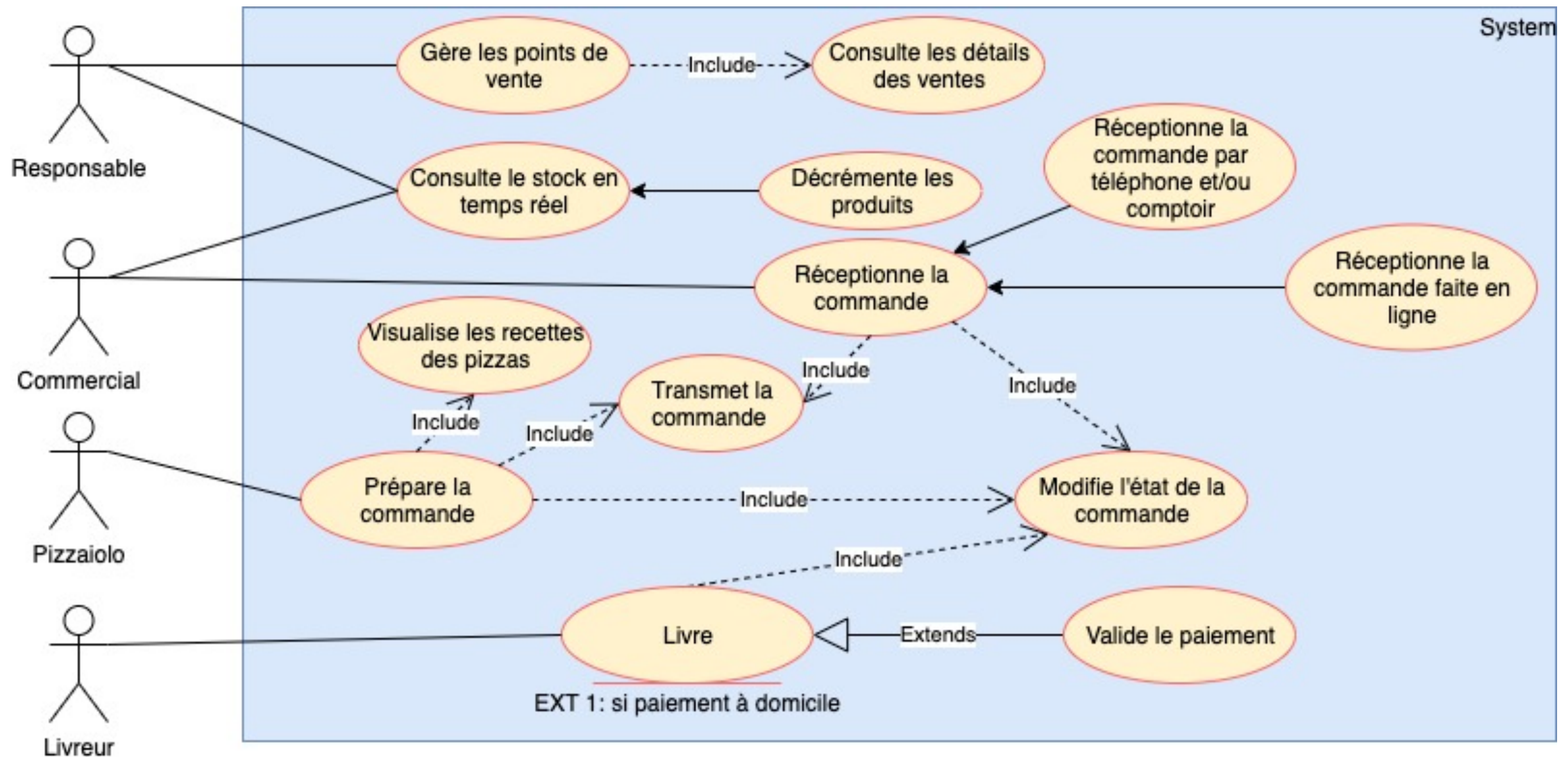




## 3.3 LES CAS D'UTILISATION GÉNÉRAUX

Diagramme de cas d'utilisation du package « Interface Backend »

Voici le diagramme de cas d'utilisation du package « Interface Backend » mettant en relation les acteurs « commercial », « pizzaiolo », « livreur » et « responsable » et décrivant les différentes fonctionnalités du back office.

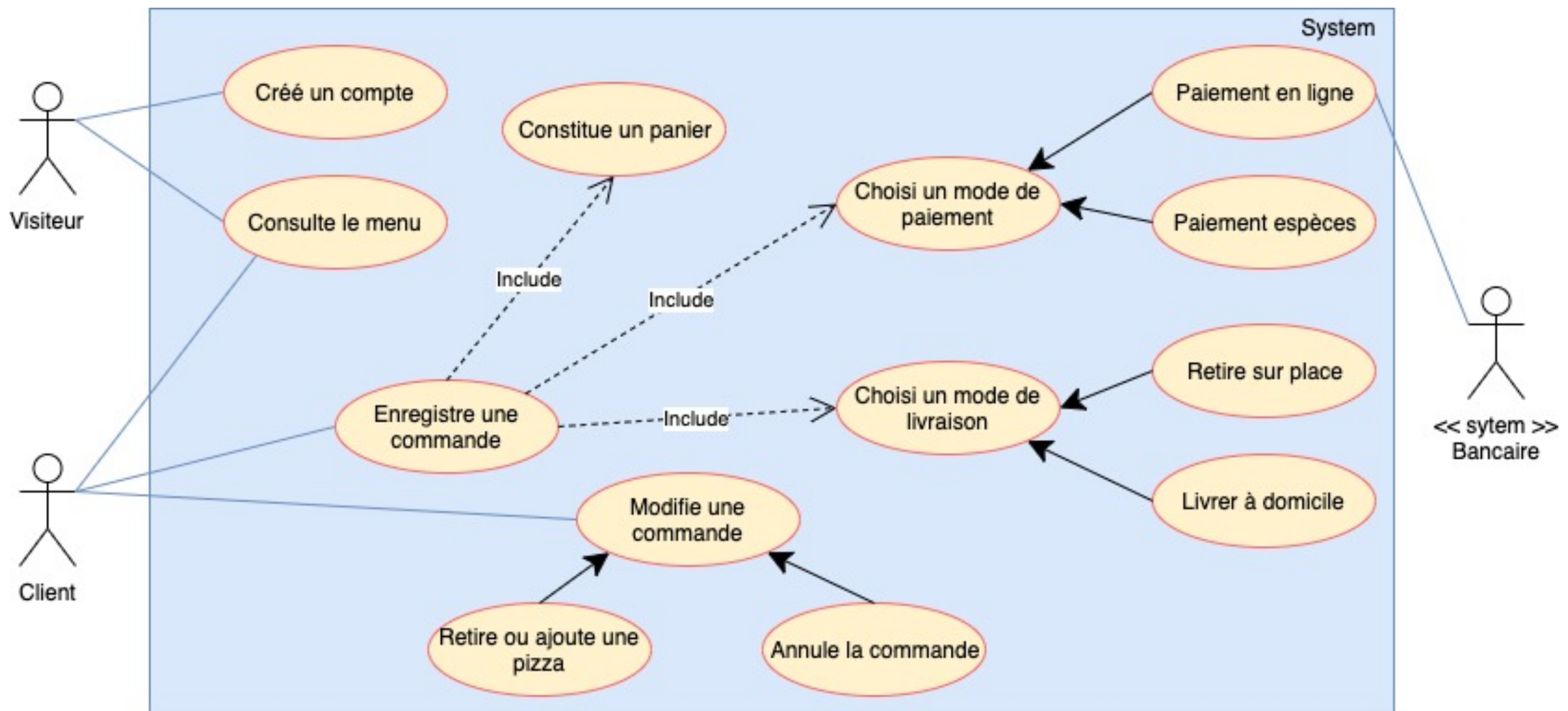




## 3.3 LES CAS D'UTILISATION GÉNÉRAUX

Diagramme de cas d'utilisation du package « Interface Frontend »

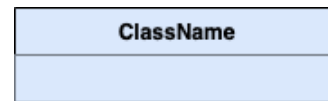
Voici le diagramme de cas d'utilisation du package « Interface Frontend » mettant en relation les acteurs « client », « visiteur » et « système Bancaire » et décrivant les différentes fonctionnalités du site web.



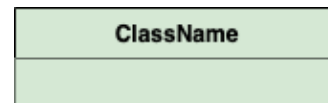
## 4.1 EXPLICATION

Voici une légende permettant la lecture du diagramme

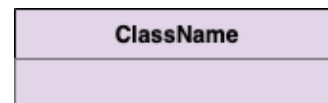
Classe (bleu) : Toute les classes liés au module « **Contact** ».



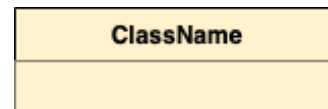
Classe (vert) : Toute les classes liés au module « **Ventes** ».



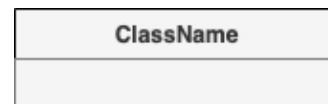
Classe (violet) : Toute les classes liés au module « **Achats** ».



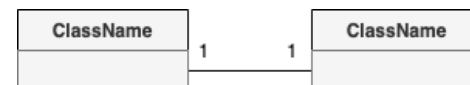
Classe (jaune) : Toute les classes liés au module « **Produits** ».



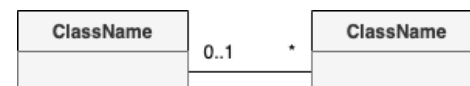
Classe (gris) : Toute les classes liés au module « **Point de ventes** ».



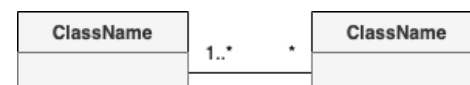
Association (**One to One**) Un à un



Association (**One to Many**) Un à plusieurs (ou plusieurs à un)



Association (**Many to Many**)  
Plusieurs à plusieurs

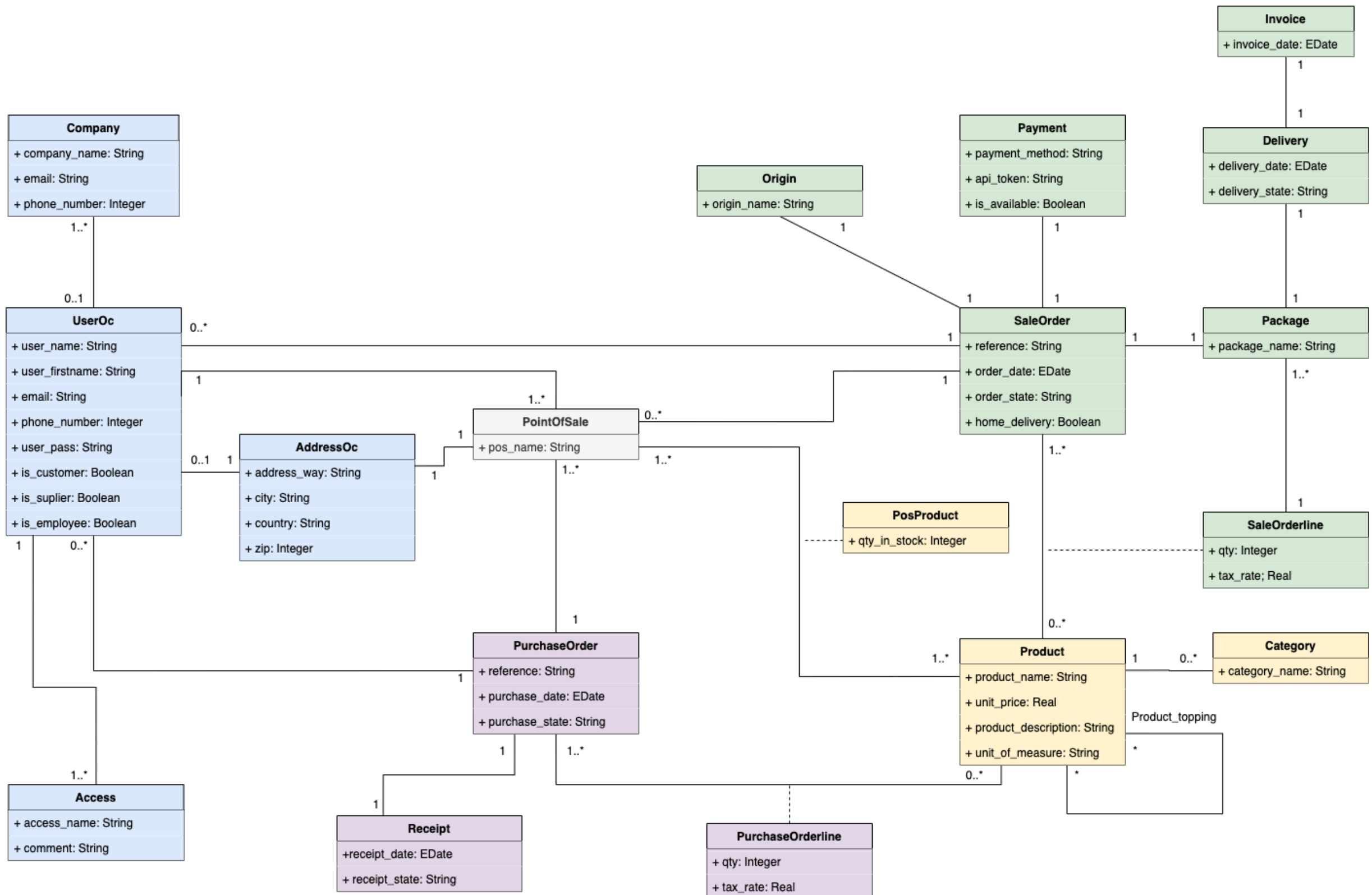


Voici le [diagramme de classes](#) de votre futur système. Ce diagramme représente les [éléments](#) qui composeront votre système. Chaque application qui va mettre en oeuvre votre système sera [une instance](#) des différentes classes qui le compose.

Ce diagramme de classe sert de base à la [génération du code](#) de l'application. Une classe est définie par son [nom](#), ses [attributs](#) et ses [opérations](#). Sur le diagramme de classe suivant, nous retrouverons uniquement le nom et ses attributs.

Vous retrouverez également des [relations](#) entre ces classes. Il s'agit d'[associations](#). Cela signifie simplement qu'une classe peut en utiliser une autre. Vous retrouverez également des chiffres permettant d'indiquer quel [type de relation](#) est utilisé entre elles. On parle alors de multiplicités.

## 4.2 DIAGRAMME DE CLASSE





## 4.1 LE WORKFLOW D'UN CYCLE DE VIE D'UNE COMMANDE (FORMAT TEXTUEL)

---

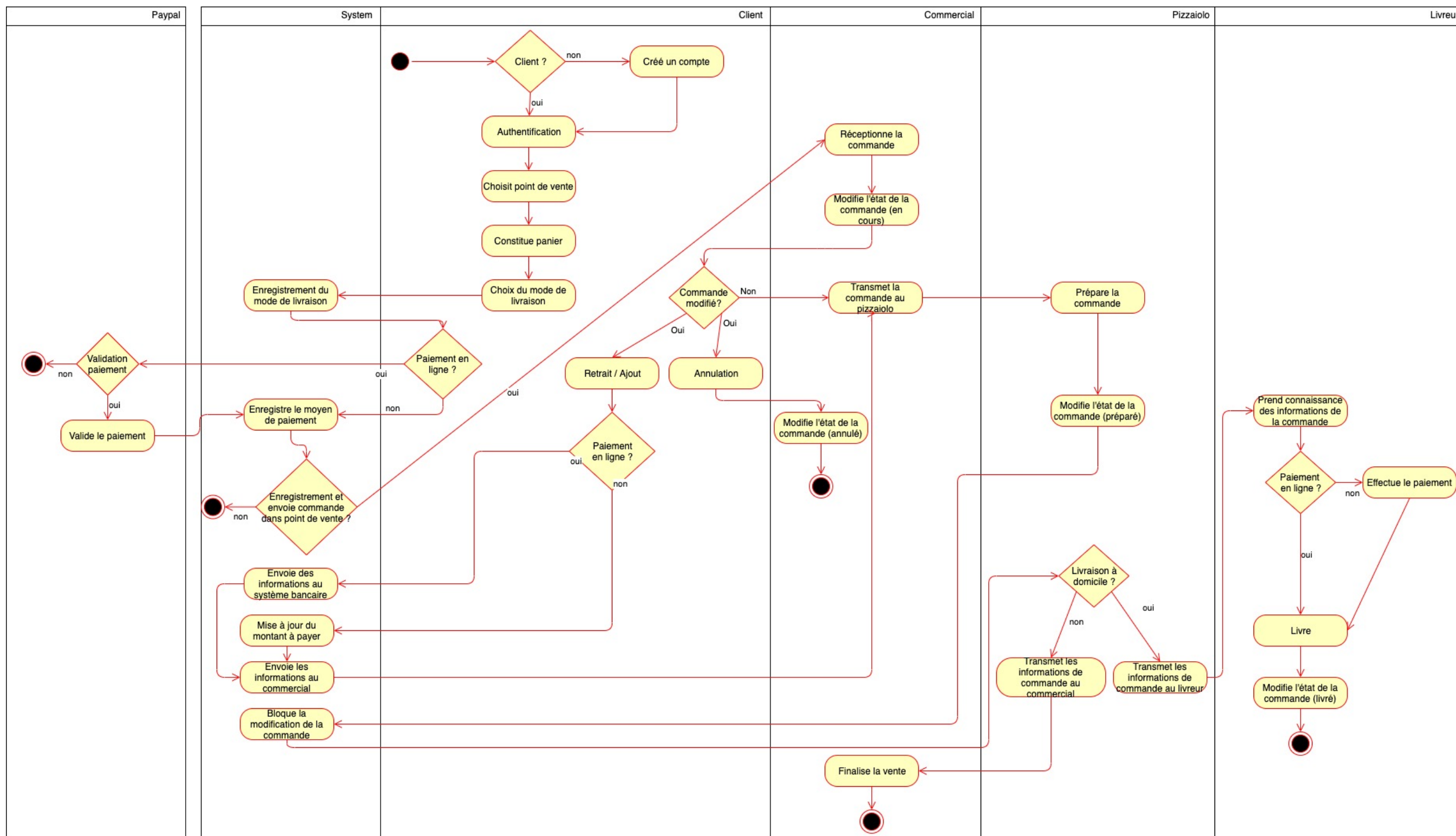
### Explication textuelle

Une commande démarre lorsque le client est [authenticé](#). Il pourra alors [choisir son point de vente](#), [constituer son panier](#) pour ensuite [choisir son mode de livraison](#). Quelque soit le choix de notre client, le système [enregistrera le mode de livraison](#) qui nous sera utile juste avant la fin du cycle de vie de la commande. Le client devra ensuite [choisir le moyen de paiement](#). Si l'utilisateur choisit le mode [paiement en ligne](#), il se verra redirigé vers un système bancaire afin d'effectuer le paiement. Si en revanche, il souhaite payer [en Espèces](#), le système [enregistre le moyen de paiement](#) dans les informations concernant la commande.

Le système sait à présent qu'il doit [distribuer ces informations](#) au commercial du point de vente. Celui-ci [réceptionne la commande](#) et [modifie l'état de la commande](#) afin d'indiquer qu'elle est en cours. A tout moment, le client peut, s'il le souhaite [modifier ou annuler sa commande](#). S'il souhaite retirer ou ajouter une pizza à sa commande, le système va vérifier le moyen de paiement utilisé. Si c'est un paiement en ligne, il se charge d'[envoyer le changement du montant](#) au système bancaire, tout en prévenant le responsable et le commercial. Sinon, il [met à jour le montant](#) que le client devra payer. Ces modifications sont ensuite [transférées par notification au commercial](#). En revanche, s'il souhaite [annuler sa commande](#), le commercial devra alors [changer l'état de la commande](#) afin d'indiquer à ses collègues qu'elle est annulée dans le cas où le pizzaiolo aurait démarré la commande. Cela permettra également de prévenir éventuellement le service bancaire afin de rembourser le client.

Ensuite la commande est [transmise au pizzaiolo](#) qui se chargera de [modifier l'état](#) de celle-ci lorsqu'il aura terminé. Le système empêchera donc toute tentative de modification de commande. Le pizzaiolo [fait le point sur le mode livraison](#) indiqué et [transmet la commande au commercial](#) s'il s'agit d'une livraison sur place, ou [au livreur](#) afin qu'il puisse [effectuer la livraison](#). Lorsque cette livraison est effectuée, le livreur [met à jour l'état de la commande](#) afin d'indiquer au système et à ses collègues que celle-ci est livrée et terminée.

## 4.1 LE WORKFLOW D'UN CYCLE DE VIE D'UNE COMMANDE (FORMAT DIAGRAMME)





## 6.1 C'EST QUOI ?



À présent, voici les fiches descriptives permettant de détailler les différentes fonctionnalités du programme. Ces fiches descriptives vous permettront de visualiser la chronologie des actions qui devront être réalisées, et ainsi, clarifier le déroulement de la fonctionnalité en question. C'est le rôle du scénario.

Vous visualiserez 2 ou 3 sortes de scénarios. Le **scénario nominal** permet de décrire le déroulement idéal des actions qui seront réalisées par l'utilisateur et le système. Le **scénario alternatif** permet de décrire les éventuelles étapes liées aux choix de l'utilisateur. Enfin, le **scénario d'exception** permet de définir une étape du déroulement pouvant être amenée à être perturbée à cause d'un événement anormal.

Ensuite, nous récapitulons toutes les situations d'**arrêt du cas d'utilisation**. Nous incluons également des post-conditions permettant de vérifier et témoigner le bon fonctionnement du cas d'utilisation après son arrêt.

Enfin, si cela nous a semblé nécessaire, nous indiquons quelques **compléments** en matière d'ergonomie, des performances attendue ou encore indiquer des problèmes non résolus.





## 6.2 PACKAGE INTERFACE « FRONTEND »

---

Retrouvez les cas d'utilisation suivants détaillés sous forme de scénarios.

1. Consulte le menu
2. Créé un compte
3. Enregistre une commande
4. Modifie une commande





### Cas n°1

**Nom :** Consulte le menu (Package « **Interface Frontend** »)

**Acteur(s) :** Visiteur, client

**Description :** La consultation de la carte doit être possible pour un client et un visiteur

**Date :** 19/12/2019

**Pré-conditions :** L'utilisateur doit être sur le site « OC - PIZZA »

**Démarrage :** L'utilisateur a demandé la page contenant la liste des pizzas

### Description

#### Le scénario nominal :

1. **Le système** affiche la page contenant la liste des pizzas
2. L'utilisateur peut sélectionner une pizza
3. **Le système** affiche une photo, la liste des ingrédients et le prix de la pizza sélectionné
4. L'utilisateur peut ensuite quitter cette sélection
5. **Le système** retourne à l'affichage de la liste des pizzas

#### Les scénarios alternatifs :

- 2.b L'utilisateur peut quitter la consultation de la pizza sélectionnée
- 4.b L'utilisateur peut la consultation de la liste des pizzas

**Fin** : Scénario nominal : aux étapes 2 et 4, sur décision de l'utilisateur

**Post-conditions** : Aucun

### Compléments

#### Ergonomie

L'affichage des produits devra se faire par groupe de 15 si vous possédez plus de 20 choix de pizzas.

#### Performance attendue

La recherche des produits doit se faire rapidement. Elle doit donc se faire en moins de 10 secondes.

#### Problèmes non résolus

Doit-on prévoir des critères de recherche de pizzas comme par catégorie, ingrédients ?

Doit-on prévoir un filtre afin de trier les pizzas (par exemple : par tranche de prix, taille, disponibilité) ?



### Cas n°2

**Nom** : Créé un compte (Package « **Interface Frontend** »)

**Acteur(s)** : Visiteur

**Description** : Permettre à un visiteur de créer un compte client

**Date** : 19/12/2019

**Pré-conditions** : L'utilisateur doit être sur le site « OC - PIZZA »

**Démarrage** : L'utilisateur a demandé la page de création de compte

### Description

#### Le scénario nominal :

1. **Le système** affiche la page de création de compte
2. L'utilisateur remplit le formulaire de création de compte et valide
3. **Le système** vérifie les données saisies
4. **Le système** enregistre les données saisies en base de données et envoie un mail de confirmation
5. **Le système** affiche la carte des pizzas et indique à l'utilisateur qu'il est connecté

#### Les scénarios alternatifs :

- 2.a L'utilisateur peut quitter la page de création de compte

### Les scénarios d'exception

4.a **Le système** n'a pas réussi à enregistrer les données

4.b **Le système** affiche un message à l'utilisateur prévenant que l'enregistrement a échoué et l'invite à reprendre la création de compte. Les informations sont envoyés au(x) responsable(s). (Retour à l'étape 2)

### Fin :

- Scénario nominal : à l'étape 5 lorsque l'enregistrement est effectué
- Scénario d'exception : après le point 4, si l'enregistrement à échoué

### Post-conditions :

- Scénario nominal : L'utilisateur est enregistré en base de données
- Scénario d'exception : L'enregistrement a échoué et les informations ont été envoyés au(x) responsable(s).

### Compléments

#### Ergonomie

La création d'un compte doit se faire sur un maximum de 2 pages. Les éventuels messages d'erreurs devront apparaître sous forme de pop-up.

### Cas n°3

**Nom** : Enregistre une commande (Package « **Interface Frontend** »)

**Acteur(s)** : Client

**Description** : Permet à un client de passer une commande

**Date** : 19/12/2019

**Pré-conditions** : Le client doit être identifié et se trouver sur la page de la consultation du menu

**Démarrage** : L'utilisateur est sur la page de la liste des pizzas (menu)

### Description

#### Le scénario nominal :

1. L'utilisateur parcourt le menu - **Appel du cas d'utilisation** « [Consulte le menu](#) »
2. L'utilisateur choisit son point de vente pour avoir accès aux stocks de celui-ci
3. **Le système** fait **appel au cas d'utilisation** « [Constitue un panier](#) »
4. **Le système** fait **appel au cas d'utilisation** « [Choisit le mode livraison](#) »
5. L'utilisateur sélectionne la livraison à domicile
6. **Le système** demande l'adresse de livraison
7. L'utilisateur indique son adresse de livraison
8. **Le système** ajoute l'adresse de livraison au récapitulatif de la commande
9. **Le système** fait **appel au cas d'utilisation** « [Choisit le mode de paiement](#) »
10. L'utilisateur choisit le mode de paiement en ligne par carte bancaire
11. **Le système** bancaire prend la main
12. **Le système** ajoute le moyen de paiement au récapitulatif de la commande
13. **Le système** valide la commande et envoie la commande au point de vente sélectionné



### Les scénarios alternatifs

5.a L'utilisateur sélectionne la livraison « sur place »

**Le système** passe à l'étape 9

10.a L'utilisateur choisit de payer en espèces

**Le système** passe à l'étape 12

### Les scénarios d'exceptions

13.a **Le système** a échoué dans l'enregistrement de la commande

13.b **Le système** envoie les informations au(x) responsable(s) - **Arrêt du cas d'utilisation**

### Fin :

- Scénario nominal : à l'étape 13, la commande est validée et est envoyé au point de vente sélectionné par l'utilisateur
- Scénario d'exception : après le point 13, si l'enregistrement à échoué

### Post-conditions :

Scénario nominal : La commande est enregistré en base de données

Scénario d'exception : Les informations pouvant être récupérées sont transmises au(x) responsable(s).

### Cas n°4

**Nom** : Modifie une commande (Package « **Interface Frontend** »)

**Acteur(s)** : Client

**Description** : Permet à un client d'apporter des modifications à la commande, soit pour retirer ou ajouter une pizza, soit pour l'annuler.

**Date** : 19/12/2019

**Pré-conditions** : Le client doit être identifié et doit posséder une commande en cours, non préparée.

**Démarrage** : L'utilisateur choisit de modifier une commande

### Description

#### Le scénario nominal :

1. **Le système** récapitule la commande choisie et indique l'état de la commande : « *En cours* »
2. L'utilisateur décide d'annuler la commande et clique sur le bouton annuler
3. **Le système** transmet les informations d'annulation de la commande au commercial du point de vente, au système bancaire et au(x) responsable(s) et met à jour le stock

#### Le scénario alternatif :

- 1.a **Le système** récapitule la commande et indique l'état de celle-ci : « *Préparé* »

**Le système** empêche le clique sur le bouton d'ajout et/ou retrait d'une pizza ou d'annulation de la commande

- 2.a L'utilisateur décide d'ajouter ou de retirer une pizza

**Le système** transmet les information d'ajout ou de retrait d'une pizza au commercial du point de vente, au système bancaire et au(x) responsable(s) et met à jour le stock



## 6.3 PACKAGE INTERFACE « BACKEND »

---

Retrouvez les cas d'utilisation suivants détaillés sous forme de scénarios.

1. Réceptionne la commande
2. Consulte le stock en temps réel
3. Prépare la commande
4. Livre
5. Gère les points de vente





### Cas n°1

**Nom** : Réceptionne la commande (Package « **Interface Backend** »)

**Acteur(s)** : Commercial

**Description** : Le commercial réceptionne une commande par téléphone, au comptoir ou en ligne

**Date** : 20/12/2019

**Pré-conditions** : Le commercial doit être authentifié avec un compte administrateur

**Démarrage** : L'utilisateur se trouve sur la page de réception des commandes

### Description

#### Le scénario nominal :

1. **Le système** affiche en temps réel une nouvelle commande en ligne déjà réglé
2. L'utilisateur clique sur la commande et fait appel au cas d'utilisation « **Modifie l'état de la commande** » en précisant « *en cours* »
3. **Le système** fait appel au cas d'utilisation « **Transmet la commande au pizzaiolo** »

#### Le scénarios alternatifs :

- 1.a L'utilisateur clique sur « **nouvelle commande** » et enregistre la commande **prise au comptoir**
- 1.b L'utilisateur encaisse et enregistre le paiement de la commande
- 1.c **Le système** met à jour le stock
- 1.d Reprise étape 2 afin de finaliser la réception de la commande

### Les scénarios alternatifs (suite) :

- 1.a L'utilisateur clique sur « **nouvelle commande** » et enregistre la commande **prise par téléphone**
- 1.b L'utilisateur enregistre le mode de paiement (sur place, à domicile) et le mode de livraison
- 1.c Le système met à jour le stock
- 1.d Reprise à l'étape 2 afin de finaliser la réception de la commande

### Fin :

- Scénario nominal : à l'étape 3, la commande est transmise au pizzaiolo

### Post-conditions :

Scénario nominal : La commande est enregistré en base de données

## Compléments

### Ergonomie

La dernière commande réalisée devra se situer en premier de la liste pour éviter un temps de recherche inutile.

### Performance attendue

L'arrivée d'une nouvelle commande en ligne devra être accompagnée d'une notification aussi bien sonore que textuelle.

### Problèmes non résolus

La réception d'une commande sera-t-elle consacrée qu'à l'acteur commercial (en supposant qu'il existe) ? Le pizzaiolo ou le livreur pourront-ils réceptionner eux aussi une commande ?

### Cas n°2

**Nom :** Consulte le stock en temps réel (Package « **Interface Backend** »)

**Acteur(s) :** Commercial, Responsable

**Description :** L'utilisateur suit en temps réel le stock décrétementé par le système à chaque nouvelle commande confirmé

**Date :** 20/12/2019

**Pré-conditions :** L'utilisateur doit être authentifié avec un compte administrateur

**Démarrage :** L'utilisateur se trouve sur la page des stocks

### Description

#### Le scénario nominal :

1. **Le système** lui affiche le niveau de stock en cours pour tous les produits du **point de vente** en question
2. L'utilisateur vérifie que l'état des stocks est bon.
3. L'utilisateur quitte la session.

#### Les scénarios alternatifs :

- 3.a L'utilisateur souhaite changer la quantité sur des ingrédients perdus et non pris en compte par le système de gestion des stocks - Appel du cas d'utilisation « **Décrémente les produits** »
- 3.b **Le système** prend en compte les modifications manuelles et met à jour le stock.
- 3.c L'utilisateur quitte la session



### Les scénarios d'exceptions :

1.a **Le système** échoue lors de l'affichage du niveau de stock en temps réel.

**Le système** enregistre les informations - **Arrêt du cas d'utilisation**

### Fin :

- Scénario nominal : à l'étape 3, lorsque l'utilisateur quitte la session
- Scénarios d'exceptions : 1ère étape si l'affichage du stock réel échoue

**Post-conditions** : Le stock est mis à jour à chaque nouvelle commande enregistrée

### Compléments

#### Ergonomie

La mise à jour du stock doit pouvoir se faire de manière automatique sans rechargement de page.

#### Performance attendue

Lorsque la quantité d'un produit atteint un certain seuil (par exemple 10 selon les statistiques de vente) envoyer une notification pour prévenir le commercial

### Cas n°3

**Nom** : Prépare la commande (Package « **Interface Backend** »)

**Acteur(s)** : Pizzaiolo

**Description** : Le pizzaiolo prend connaissance d'une commande que lui a transmis le commercial et l'a réalise

**Date** : 20/12/2019

**Pré-conditions** : Le pizzaiolo doit être authentifié avec un compte administrateur

**Démarrage** : L'utilisateur a cliqué sur le bouton « Commande à préparer »

### Description

#### Le scénario nominal :

1. **Le système** lui affiche une liste de commande à préparer trié par date et heure
2. L'utilisateur clique sur la commande à préparer
3. **Le système** affiche le détail de la commande reprenant chaque pizzas à réaliser
4. L'utilisateur clique sur une pizza de la commande
5. **Le système** fait appel au cas d'utilisation « [Visualise les recettes de pizzas](#) »
6. L'utilisateur termine sa commande et fait appel au cas d'utilisation « [Modifie l'état de la commande](#) » indiquant « préparé »
7. **Le système** bloque le cas d'utilisation « [Modifie une commande](#) » du package Frontend
8. **Le système** vérifie quel mode de livraison il s'agit pour cette commande.
9. **Le système** fait appel au cas d'utilisation « [Transmet les informations de commande](#) »

### Les scénarios alternatifs :

5.a L'utilisateur est en pleine préparation des pizzas

5.b **Le système** envoie une notification indiquant que le client a annulé la commande

5.c L'utilisateur stoppe son travail et enregistre la quantité d'ingrédients utilisé afin de le transmettre au commercial ou au responsable pour les passer en « *perte* » - **Arrêt du cas d'utilisation**

### Fin :

- Scénario nominal : à l'étape 8, toutes les informations concernant la commande sont répertoriées et transmises au livreur
- Scénario alternatif : à l'étape 5 lorsque le client annule sa commande

**Post-conditions** : La commande est prête et transmise au livreur afin d'effectuer la livraison

### Compléments

#### Ergonomie

L'affichage de la liste des commandes ne doit pas excéder 10 commandes pour garder une certaine lisibilité.  
Pour afficher les ingrédients d'une pizza, une fenêtre pop-up s'ouvre indiquant les ingrédients présents dans la pizza sélectionnée

#### Problèmes non résolus

La transmission des informations de la commande prête doit-elle cibler l'acteur selon le mode de livraison choisi ou doit-on prévenir chaque employé ?



### Cas n°4

**Nom** : Livre (Package « **Interface Backend** »)

**Acteur(s)** : Livreur

**Description** : Le livreur prend connaissance d'une commande que lui a transmis le pizzaiolo et effectue la livraison

**Date** : 20/12/2019

**Pré-conditions** : Le pizzaiolo doit être authentifié avec un compte administrateur

**Démarrage** : L'utilisateur a cliqué sur la notification reçu pour une commande prête et se trouve sur le détail de celle-ci

### Description

#### Le scénario nominal :

1. L'utilisateur prend connaissance de la commande avec l'adresse de livraison et le mode de paiement (Paiement en ligne)
2. L'utilisateur livre le client
3. L'utilisateur fait appel au cas d'utilisation « **Modifie l'état de la commande** » en tant que « *livré* »
4. **Le système** ajoute la commande dans le tableau des commandes terminées
5. **Le système** envoie une notification pour indiquer en temps réel que la commande est livrée

#### Les scénarios alternatifs :

- 2.a L'utilisateur donne la somme à régler au client
- 2.b L'utilisateur fait appel au cas d'utilisation « **Valide le paiement** »
- 2.c Reprise à l'étape à 3

### **Fin :**

- Scénario nominal : à l'étape 5, lorsque la livraison est effectuée et validée

### **Post-conditions :**

La commande est effectuée et livrée.

Le système a indiqué en temps réel que la livraison a été effectuée.

## **Compléments**

### **Ergonomie**

L'utilisateur doit pouvoir rester sur les informations de la commande en cours et avoir un bouton « Valider » permettant de cibler la commande et d'indiquer que celle-ci est livrée (en temps réel).

### **Performances attendues**

La notification permettant de spécifier que la commande est livrée doit se faire de manière instantanée (en temps réel).

### **Problèmes non résolus**

La validation de la livraison de la commande doit-elle être envoyée uniquement au commercial, au responsable, au pizzaiolo ou à tout le monde ?

### Cas n°5

**Nom** : Gère les points de vente (Package « **Interface Backend** »)

**Acteur(s)** : Responsable(s)

**Description** : Le responsable peut gérer chaque point de ventes du groupe et consulter les détails des ventes

**Date** : 20/12/2019

**Pré-conditions** : L'utilisateur doit s'authentifier avec un compte Super Administrateur

**Démarrage** : L'utilisateur se trouve sur le tableau de bord

### Description

#### Le scénario nominal :

1. **Le système** affiche le tableau de bord avec un aperçu des ventes
2. L'utilisateur choisit un point de vente
3. **Le système** affiche le détail du point de vente (stocks, achats, ventes)
4. L'utilisateur choisit d'afficher l'historique des ventes
5. **Le système** affiche l'historique des ventes du point de vente sélectionné
6. L'utilisateur visualise les informations et quitte sa session

#### Les scénarios alternatifs :

- 4.a L'utilisateur choisit de consulter l'historique des achats
- 4.b **Le système** affiche l'historique des achats du point de vente sélectionné
- 4.c L'utilisateur prend connaissance des informations et se déconnecte



### Fin :

- Scénario nominal : Au point 6 lorsque l'utilisateur se déconnecte de sa session

**Post-conditions** : Aucun

### Compléments

#### Ergonomie

L'utilisateur doit pouvoir (sur n'importe quelle page) accéder aux détails des stocks, des ventes ou des achats.

#### Performances attendues

Le système devra vérifier si la session est toujours occupée par l'utilisateur auquel cas, l'utilisateur sera automatiquement déconnecté pour des raisons de sécurité et de confidentialités de données.



## 6.4 PACKAGE « AUTHENTIFICATION »

---

Retrouvez le cas d'utilisation suivant détaillés sous forme de scénarios.

### 1. Se connecte





### Cas n°1

**Nom** : Se connecte (Package « **Authentification** »)

**Acteur(s)** : Équipe OC PIZZA, Client

**Description** : Les acteurs pourront se connecter afin d'utiliser les fonctionnalités du programme

**Date** : 27/12/2019

**Pré-conditions** : L'utilisateur doit avoir créé un compte

**Démarrage** : L'utilisateur se trouve sur la page de connexion

### Description

#### Le scénario nominal :

1. **Le système** invite l'utilisateur à saisir ses identifiants
2. L'utilisateur saisit ses identifiants (Appel du cas d'utilisation « **Saisit les identifiants** »)
3. **Le système** fait appel au cas d'utilisation « **Vérifie les identifiants** »
4. **Le système** fait appel au cas d'utilisation « **Redirection** »

#### Les scénarios alternatifs :

- 2.a L'utilisateur n'a pas saisi les bons identifiants
- 2.b **Le système** vérifie les identifiants mais aucun ne correspondent à ceux présent en base de données
- 2.c **Le système** affiche un message invitant l'utilisateur à saisir ses bons identifiants.



### Fin :

- Scénario nominal : à l'étape 5, lorsque l'utilisateur est redirigé
- Scénario alternatif : à l'étape 2 lorsque l'utilisateur ne saisit pas les bons identifiants

### Post-conditions :

L'utilisateur est redirigé et peut utiliser le programme

### Compléments

#### Ergonomie

La connexion doit se faire sur une seule page.

#### Performances attendues

La vérification des identifiants ne doit pas excéder 10 sec.

#### Problèmes non résolus

L'utilisateur aura-t-il un espace personnel ?



FIN

