

OC Pizza

oc_pizza

Dossier de conception fonctionnelle

Version 1.2

Auteur

Nathan

Développeur

TABLE DES MATIÈRES

1 -Versions.....	3
2 -Introduction.....	4
2.1 -Objet du document.....	4
2.2 -Références.....	4
2.3 -Besoin du client.....	4
2.3.1 -Contexte.....	4
2.3.2 -Enjeux et Objectifs.....	5
3 -Description générale de la solution.....	6
3.1 -Les principe de fonctionnement.....	6
3.1.1 -L'application OC Pizza :.....	6
3.1.2 -Création de l'application :.....	7
3.1.3 -Base de données :.....	7
3.1.4 -HTTP Serveur :.....	8
3.1.5 -Système d'exploitation du Serveur :.....	8
3.1.6 -Choix du serveur :.....	9
3.2 -Les acteurs.....	10
3.2.1 -Client.....	10
3.2.2 -Livreur.....	11
3.2.3 -Pizzaiolo.....	12
3.3 -Les cas d'utilisation généraux.....	13
3.3.1 -Diagrammes de Packages.....	13
4 -Les workflows.....	14
4.1 -Cycle de vie d'une commande.....	14
4.1.1 -Lors du paiement en ligne.....	14
4.1.2 -Paiement au retrait.....	16
5 -Application Web.....	18
5.1 -Les acteurs.....	18
5.2 -Les cas d'utilisation.....	18
5.2.1 -Cas d'utilisation générale.....	18
5.2.2 -Cas d'utilisation du client.....	20
5.2.3 -Cas d'utilisation des commandes.....	21
5.2.4 -Cas d'utilisation de l'administration.....	22
5.3 -Les règles de gestion générales.....	22
5.4 -Workflow.....	23
5.4.1 -Scénario d'une prise de commande en ligne:.....	23
5.4.1.1 -Légendes.....	23
5.4.1.2 -Scénario.....	23

1 - VERSIONS

Auteur	Date	Description	Version
Nathan	11/08/19	Création du document	v1.0
Nathan	12/07/20	Réadapation du format	v1.1
Nathan	13/07/20	Ajout de dernières sections	v1.2

2 - INTRODUCTION

2.1 - Objet du document

Le présent document constitue le dossier de conception fonctionnelle de l'application OC Pizza.

Objectif du document est de décrire en détails les différents éléments de la conception fonctionnelle.

Les éléments du présent dossier découlent :

- de la description générale de la solution.
- des workflows.
- de l'application Web.

2.2 - Références

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

1. **DCT – OC Pizza** : Dossier de conception technique de l'application
2. **DCT – OC Pizza** : Dossier d'exploitation de l'application

2.3 - Besoin du client

2.3.1 - Contexte

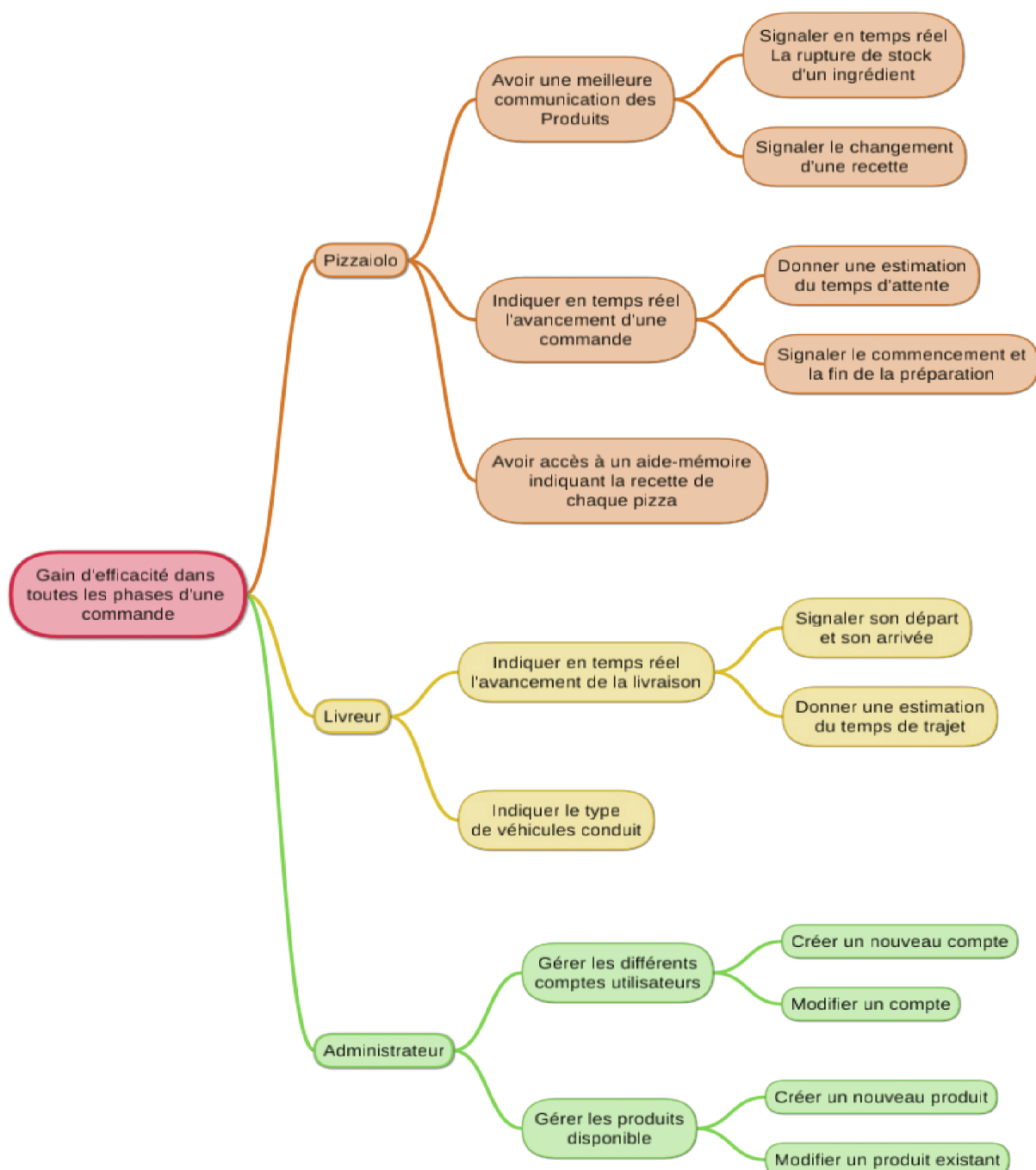
« OC Pizza » est un jeune groupe de pizzeria en plein essor et spécialisé dans les pizzas livrées ou à emporter. Il compte déjà 5 points de vente et prévoit d'en ouvrir au moins 3 de plus d'ici la fin de l'année. Un des responsables du groupe a pris contact avec vous afin de mettre en place un système informatique, déployé dans toutes ses pizzerias et qui lui permettrait notamment : d'être plus efficace dans la gestion des commandes, de leur réception à leur livraison en passant par leur préparation ;

- de suivre en temps réel les commandes passées et en préparation ;
- de suivre en temps réel le stock d'ingrédients restants pour savoir quelles pizzas sont encore réalisables ;
- de proposer un site Internet pour que les clients puissent :
 - passer leurs commandes, en plus de la prise de commande par téléphone ou sur place
 - payer en ligne leur commande s'ils le souhaitent – sinon, ils paieront directement à la livraison
- modifier ou annuler leur commande tant que celle-ci n'a pas été préparée
- de proposer un aide mémoire aux pizzaiolos indiquant la recette de chaque pizza
- d'informer ou notifier les clients sur l'état de leur commande

2.3.2 - Enjeux et Objectifs

Les objectifs de notre solution sont de simplifier et automatiser le processus de prise de commande (et accessoirement de paiements) ainsi que l'organisation des commandes au moment de la préparation et de la livraison tout en permettant un suivi de l'état par le client.

Enfin, notre solution a aussi pour but de simplifier la vie des employés pour la gestion des stocks qui seraient faits en temps réel en fonction des commandes traitées et donnera un aperçu en vue des futurs ravitaillements pour que tous les produits restent accessibles en permanence.



3 - DESCRIPTION GÉNÉRALE DE LA SOLUTION

L'application se veut simple pour tout les utilisateurs.

Autant pour le client, mais aussi en interne par les différents acteurs comme le livreur et le pizzaiolo. C'est pour cela que l'application sera disponible directement via un navigateur internet. Que ce soit sur un pc, un téléphone, une tablette ou autres appareils connectés à internet, tout utilisateur pourra y accéder avec n'importe quel navigateur web comme Google Chrome, Mozilla Firefox, Safari, et même Microsoft Edge.

Il disposera d'une interface responsive qui optimisera l'utilisation sur les petits écrans comme le téléphone portable du livreur.

3.1 - Les principe de fonctionnement

3.1.1 - L'application OC Pizza :

L'application sera divisé par plusieurs pages, qui auront chacune leurs objectifs pour un utilisateur spécifique, qui eux-mêmes seront séparés par plusieurs comptes. Chaque compte aura des droits d'accès qui leur permettra d'accéder ou non à certaines pages du site. Il y aura donc le compte de type « Administrateur », « Livreur », « Pizzaiolo », « Caissier », et enfin « Client ». Un utilisateur et le caissier pourront éventuellement créer un compte « Invité » qui aura les avantages d'un compte « Client » pour une commande. Ce compte sera supprimer à la fin de celle-ci.

Le catalogue de produits sera disponible par tous les visiteurs et utilisateurs. Ça sera donc cette page qui sera consulter par toutes personnes qui voudront effectuer une commande. Elle affichera toute les pizzas disponibles, leurs descriptions, et ainsi que leurs prix. Il y aura également d'autres produits comme des boissons ou des dessert, si ceux-ci sont proposés par la pizzeria, et si ces produits sont en stocks. Le caissier qui prendra la commande par téléphone ou en boutique, devra également accéder à cette page afin de pouvoir l'effectuer, et autant pour que le système puisse le gérer correctement.

Le client et le caissier devront ensuite passer par le panier qui donnera accès à une nouvelle page pour pouvoir saisir tout les informations nécessaires pour le paiement, la livraison, et confirmer la commande. Le panier sera modifiable tant que la commande n'a pas été confirmé.

Une fois la commande passée, **le client** aura accès à une page permettant de suivre en temps réel son avancement. Aussi il aura accès à une fonctionnalité qui permettra de modifier ou de supprimer sa commande, tant que celle-ci est en attente de préparation. Cette page nécessitera d'avoir un compte « client », et d'avoir une commande en cours pour y être accessible.

Le pizzaiolo aura accès à une page qui lui montrera toute les commandes en cours et en attente de préparation et sera trié chronologiquement et par statut (les commandes en cours sera en premier). Chaque pizza d'une commande sera accompagné d'un aide-mémoire si le pizzaiolo en a besoin. Il pourra également indiqué lorsqu'il commence une commande et quand il en termine une. Cette page sera accessible uniquement aux comptes ayant les droits d'accès.

Le livreur aura accès à une page lui montrant les commandes en cours et en attente de livraison, qui seront triées chronologiquement et par statuts (les commandes en cours seront en premier).

Chaque commande sera accompagné avec les coordonnées du client, ainsi qu'une estimation du temps de trajet. Il pourra également indiquer lorsque le livreur commence et termine une livraison. Cette page sera accessible uniquement aux comptes ayant les droits d'accès.

L'administrateur aura une page dédié à la gestion des comptes. Il pourra créer un compte et modifier les droits d'accès de celui-ci. Également il aura accès au chiffre d'affaire réaliser dans une certaine période (soir/journée/semaine/mois/année). L'administrateur pourra modifier les pages publiques, en ajoutant ou modifiant un produit comme une pizza, ainsi que leurs descriptions, leurs prix, et leurs images.

3.1.2 - Création de l'application :

Pour le visuelle, où autrement dit : « **l'intégration** », les langages utilisés seront HTML5, CSS3, et enfin Javascript, avec la bibliothèque jQuery.

HTML et **CSS** sont des langages essentiels pour le développement web, et Javascript est le seul langage de programmation que les navigateurs web peuvent lire.

jQuery est la bibliothèque Javascript la plus utilisée, et nous permettra de créer des pages dynamiques, et permet une plus grande compatibilité avec les différents navigateurs.

Pour le côté serveur nous utiliserons le langage de programmation **Python**, qui sera accompagné de son framework web **Django**.

Basé sur le principe de Modèle/View/Contrôleur, **Django** est très apprécié et considéré comme un outil moderne et efficace. Il permet de créer une application web rapidement, tout en ayant une très bonne qualité.

Le choix de ces langages et de ces frameworks sont dû à leurs efficacités dans leurs domaines respectifs, mais notamment dû à l'expérience que nous avons acquis sur ces différentes technologies. Nous les utilisons déjà, et nous avons certains acquis.

3.1.3 - Base de données :

En ce qui concerne la base de données, il y a souvent un choix assez important.

MySQL est très populaire et très performante, elle est très large et permet de travailler sur une très grande base de données. Mais celle-ci n'est pas la plus adaptée pour une application Django.

PostgreSQL est un peu moins populaire mais très performante également. Elle est plus utilisée pour les projets orientés web comme le nôtre, mais plus important, **PostgreSQL** est adaptée pour le développement d'une application **Django**.

Exemple de structure utilisée de données utilisées :

Id	Prénom	Nom	Adresse	Mail	Téléphone
Client_00001	Jean	Dupont	12 Rue des Rochers	JeanDupont@exemple.com	012345678

Id	Nom de la pizza	Ingrédient	Prix
Pizza_00001	Pizza Fromage	Sauce tomate, Emmental	8,00 €

Id	Ingrédient	Disponible (KG)	Seuil de mise en rupture (KG)
Ingrédient_0001	Sauce tomate	20	2
Ingrédient_0002	Emmental	10	1

3.1.4 - HTTP Serveur :

Le **HTTP Serveur** est un logiciel prenant en charge les requêtes client-serveur du protocoles HTTP pour le web. Les plus populaires sont **Apache** et **Nginx**. Dans notre cas nous utiliserons **Nginx**. Celui-ci est plus performant pour la gestion des ressources avec l'utilisation du Multi-Threading, permettant de procéder à plusieurs requêtes en même temps. De plus, celui-ci est de plus en plus utilisé avec les frameworks web récents comme **Django**. Nous utiliserons le protocole HTTPS pour plus de sécurité

3.1.5 - Système d'exploitation du Serveur :

Le choix d'un serveur vient aussi avec le choix d'un système d'exploitation. Chaque système apporte des outils différents, et doit être à jour pour régler les différents bugs et faille de sécurité. Les trois plus utilisé sont CentOS, Debian et Ubuntu Serveur.

Sachant qu'**Ubuntu Serveur** est développé à partir de **Debian**, ce qui le rend dépendant de celui-ci pour les grosse mise à jour, ce qui le rend plus vulnérable.

CentOS par contre est connu pour être un système très stable, le développement de chaque version est très longue, et apporte une meilleure correction de bug et des failles. Mais celle-ci a donc du retards sur les technologies et logiciels, les plus récents apportant donc une bibliothèque de logiciels moins importante que les deux autres.

Debian, certes un peu moins stable que CentOS, détient lui aussi des versions très stables, dont certaines, maintenues pendant 5 ans. Il détient d'une grande communauté très active qui permettra de trouver la plupart des solutions à nos problèmes. Debian fut conçu pour faire fonctionner des serveurs, et s'adapter aux applications professionnelles. En outre il est beaucoup plus facile à prendre en main que CentOS.

C'est pour ces raisons que nous utiliserons la dernière version stable de **Debian** pour notre serveur.

3.1.6 - Choix du serveur :

Nous avons besoin d'un serveur qui remplit les choix que nous avons fait précédemment.

Il faut que le serveur supporte une application **Django**, une base de données **PostgreSQL**, et qu'il puisse notamment tourner sur **Debian 10**.

De même il faut que le serveur soit localiser en **Europe**, et que son prix soit adapté pour une entreprise à taille humaine.

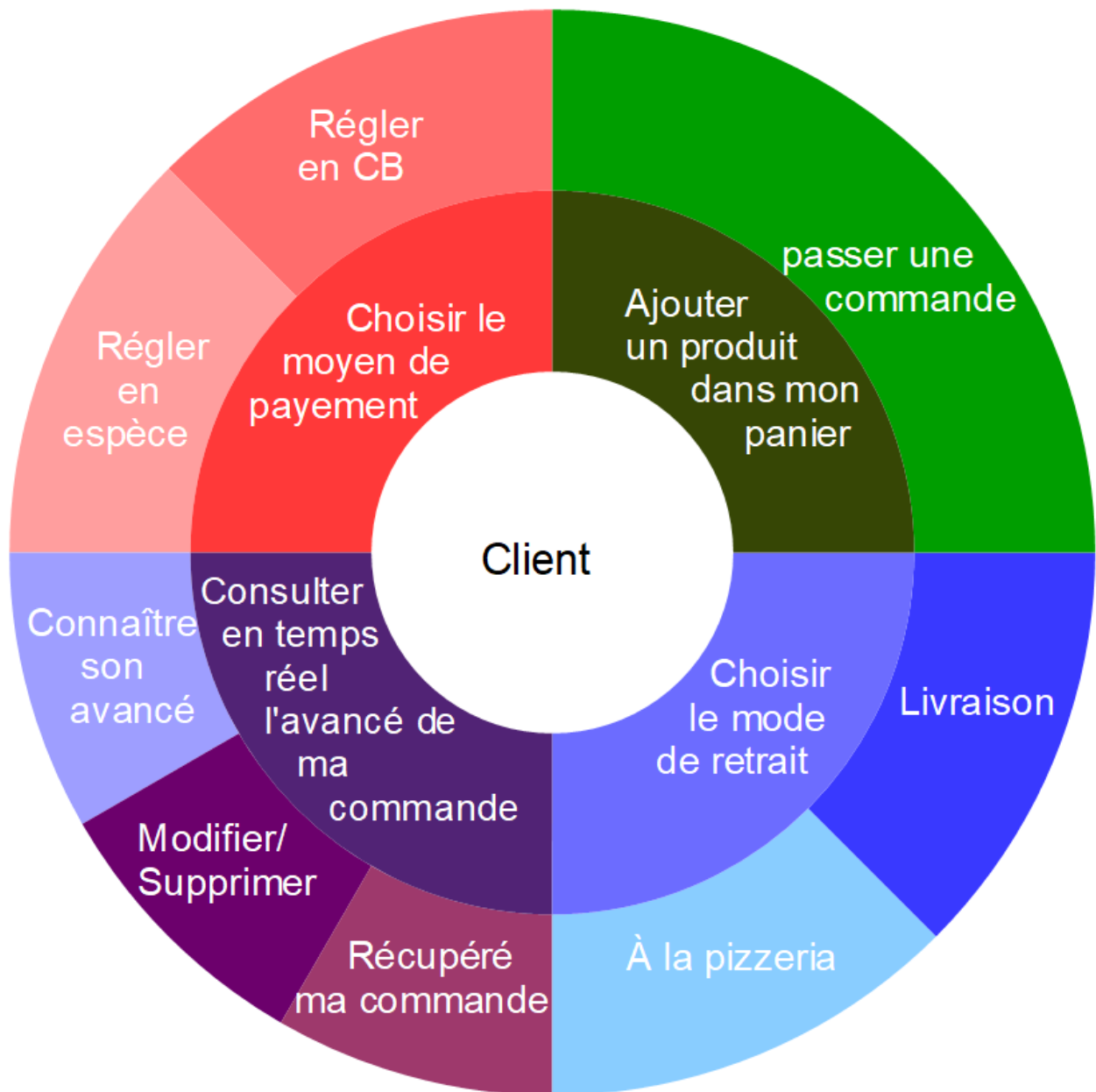
Donc pour ces raisons que nous avons choisis « **Digital Ocean** ».

Il est configurable à notre besoin sur le plan logiciel, comme le support de Debian 10, mais aussi matériel avec la possibilité d'ajouter de la puissance si besoin.

Le prix est donc adapté au besoin du client.

3.2 - Les acteurs

3.2.1 - Client



En tant que Client, je souhaite ajouter un produit dans mon panier.

En tant que Client, je souhaite choisir le mode de retrait *afin de* me faire livrer.

En tant que Client, je souhaite choisir le mode de retrait *afin de* récupérer ma commande à la pizzeria.

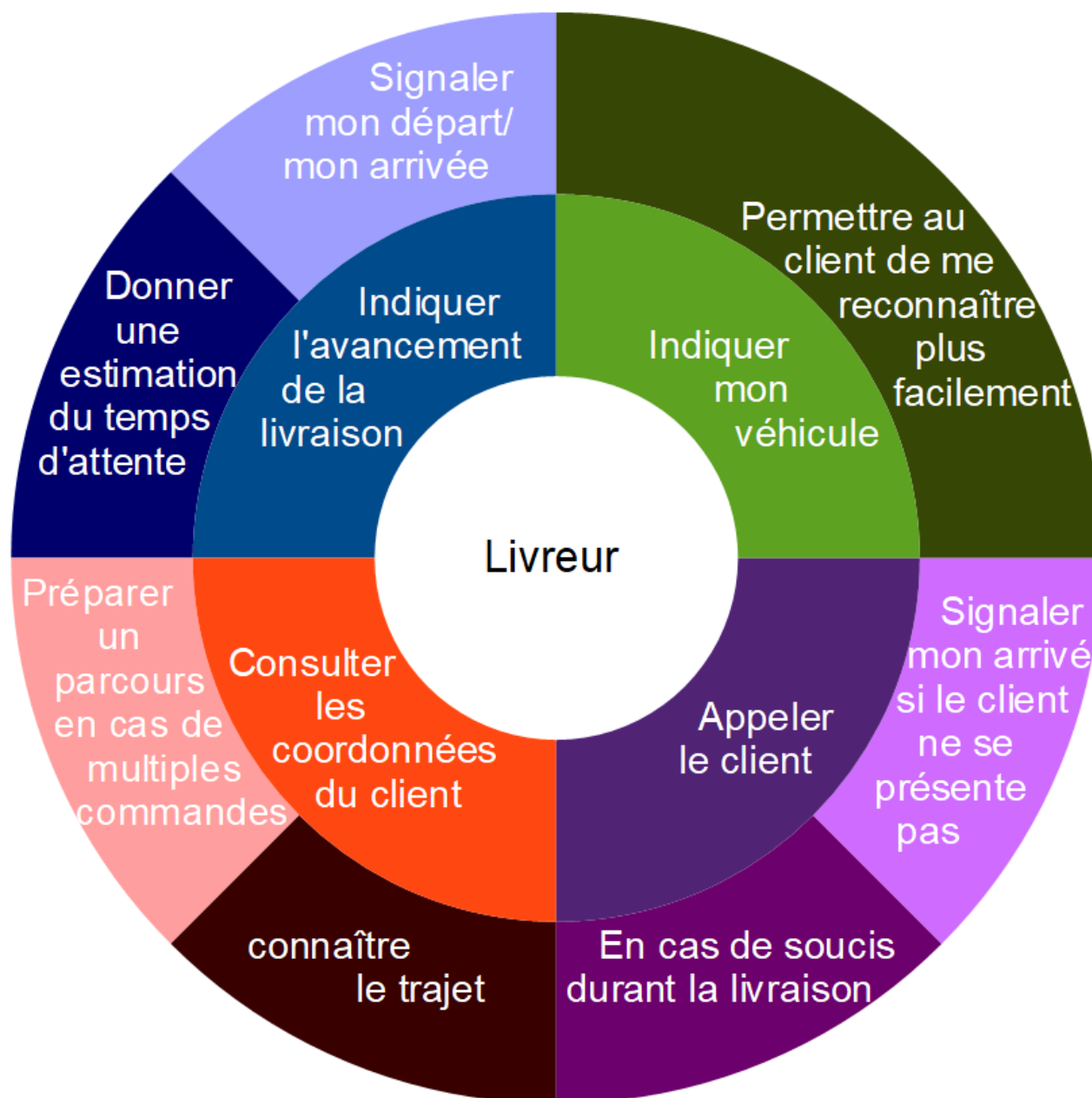
En tant que Client, je souhaite consulter en temps réel l'avancer de ma commande *afin de* connaître son avancé.

En tant que Client, je souhaite consulter en temps réel l'avancer de ma commande *afin de* la modifier ou la supprimer.

En tant que Client, je souhaite consulter en temps réel l'avancer de ma commande *afin de* la récupérer.

En tant que Client, je souhaite choisir le moyen de paiement afin de payer en carte bancaire.
En tant que Client, je souhaite choisir le moyen de paiement afin de payer en espèce.

3.2.2 - Livreur

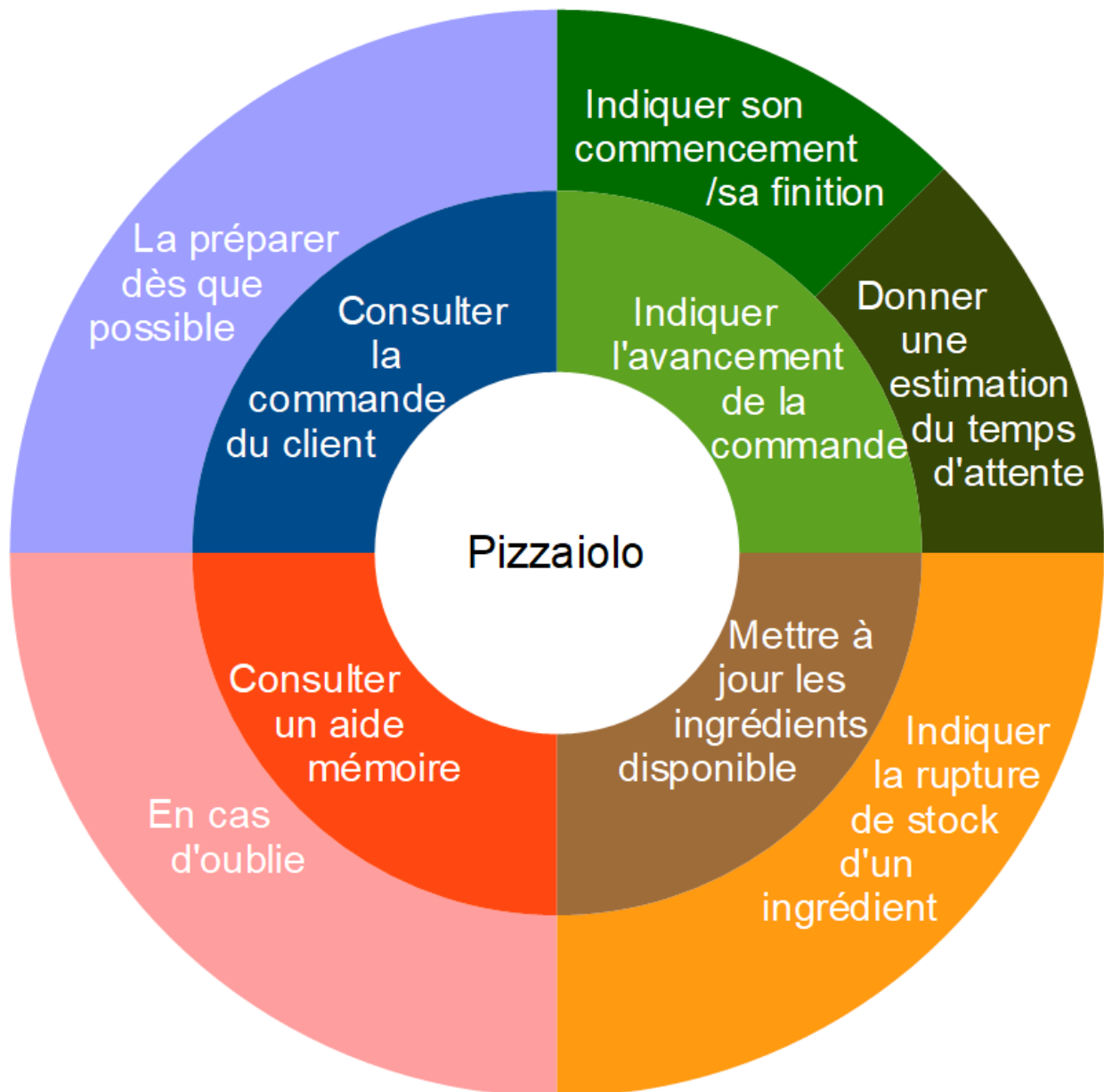


En tant que Livreur, je souhaite indiquer mon véhicule afin de permettre au client de me reconnaître plus facilement.
En tant que Livreur, je souhaite appeler le client afin de signaler mon arrivé si le client ne se présente pas.
En tant que Livreur, je souhaite appeler le client en cas de soucis durant la livraison.
En tant que Livreur, je souhaite consulter les coordonnées du client afin de connaître le trajet.
En tant que Livreur, je souhaite consulter les coordonnées du client afin de préparer un parcours en cas de multiples commandes.

En tant que Livreur, je souhaite indiquer l'avancement de la livraison *afin de* signaler mon départ et mon arrivé.

En tant que Livreur, je souhaite indiquer l'avancement de la livraison *afin de* donner une estimation du temps d'attente.

3.2.3 - Pizzaiolo



En tant que Pizzaiolo, je souhaite consulter la commande d'un client *afin de* la préparer dès que possible.

En tant que Pizzaiolo, je souhaite indiquer l'avancement de la commande *afin d'*indiquer son commencement et sa finition.

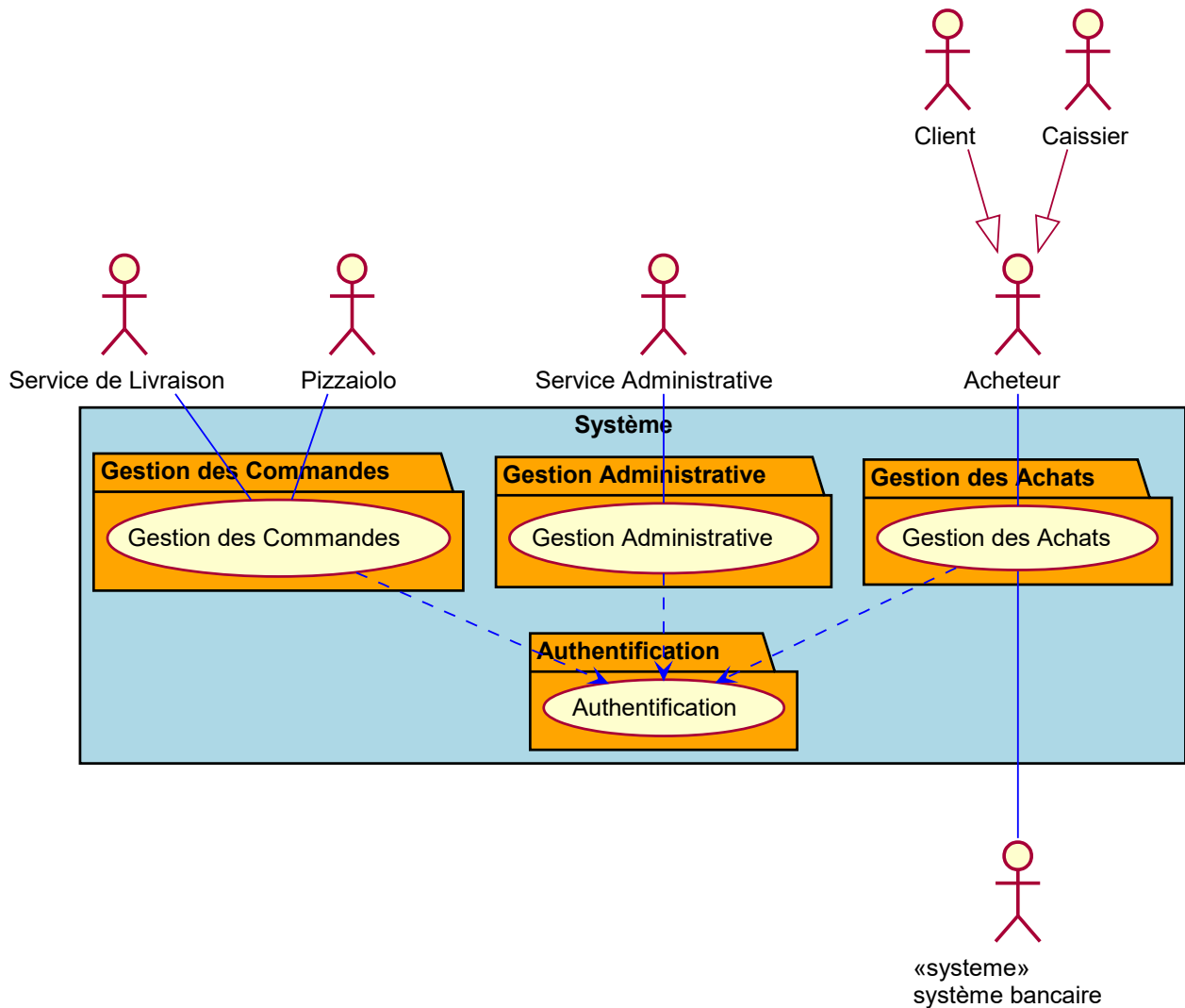
En tant que Pizzaiolo, je souhaite indiquer l'avancement de la commande *afin de* donner une estimation du temps d'attente.

En tant que Pizzaiolo, je souhaite mettre à jour les ingrédients disponible *afin d'*indiquer la rupture de stock d'un ingrédient.

En tant que Pizzaiolo, je souhaite consulter un aide mémoire en cas d'oublie

3.3 - Les cas d'utilisation généraux

3.3.1 - Diagrammes de Packages



L'**acheteur** doit pouvoir accéder au package de la *Gestion des Achats*.

Le **Service de livraison** et le **Pizzaiolo** doit pouvoir accéder au package de la *Gestion des Commandes*.

Le **service administrative** doit pouvoir accéder au package de la *Gestion Administrative*.

Tout les packages ont besoin d'accéder au packages d'*Authentification*.

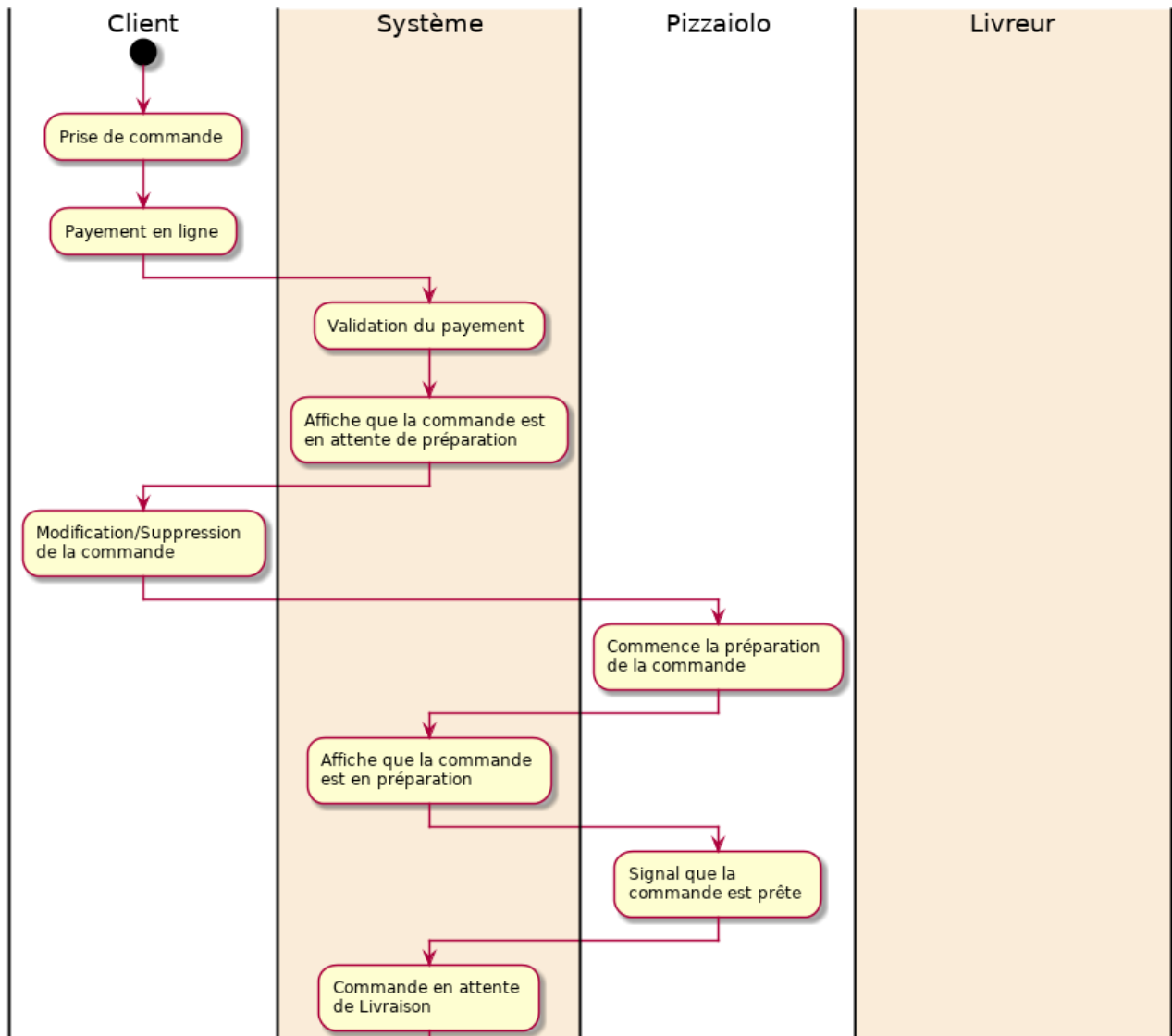
Le package **Gestion des Achats** aura acces à l'acteur système "Système bancaire".

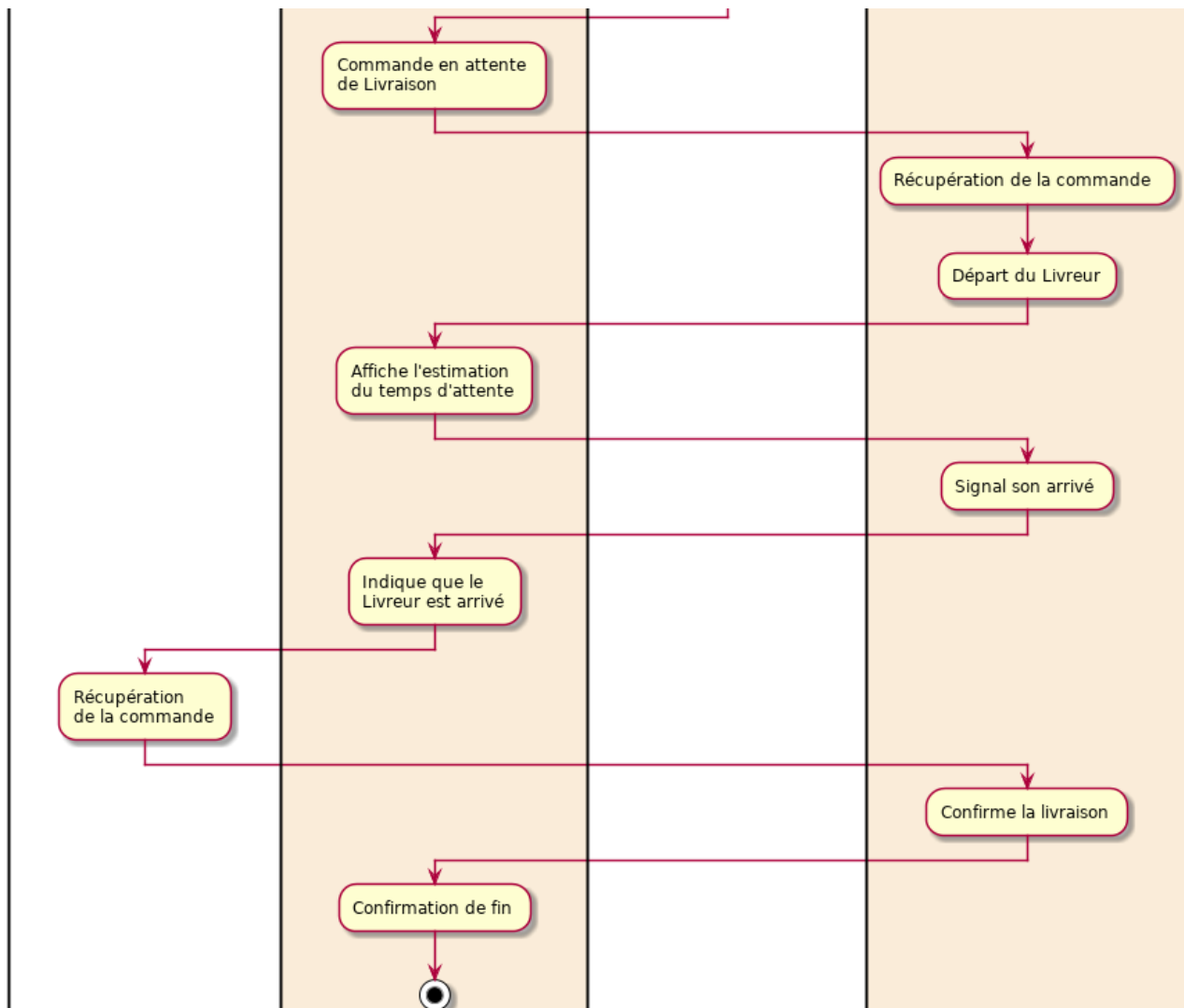
4 - LES WORKFLOWS

4.1 - Cycle de vie d'une commande

4.1.1 - Lors du paiement en ligne

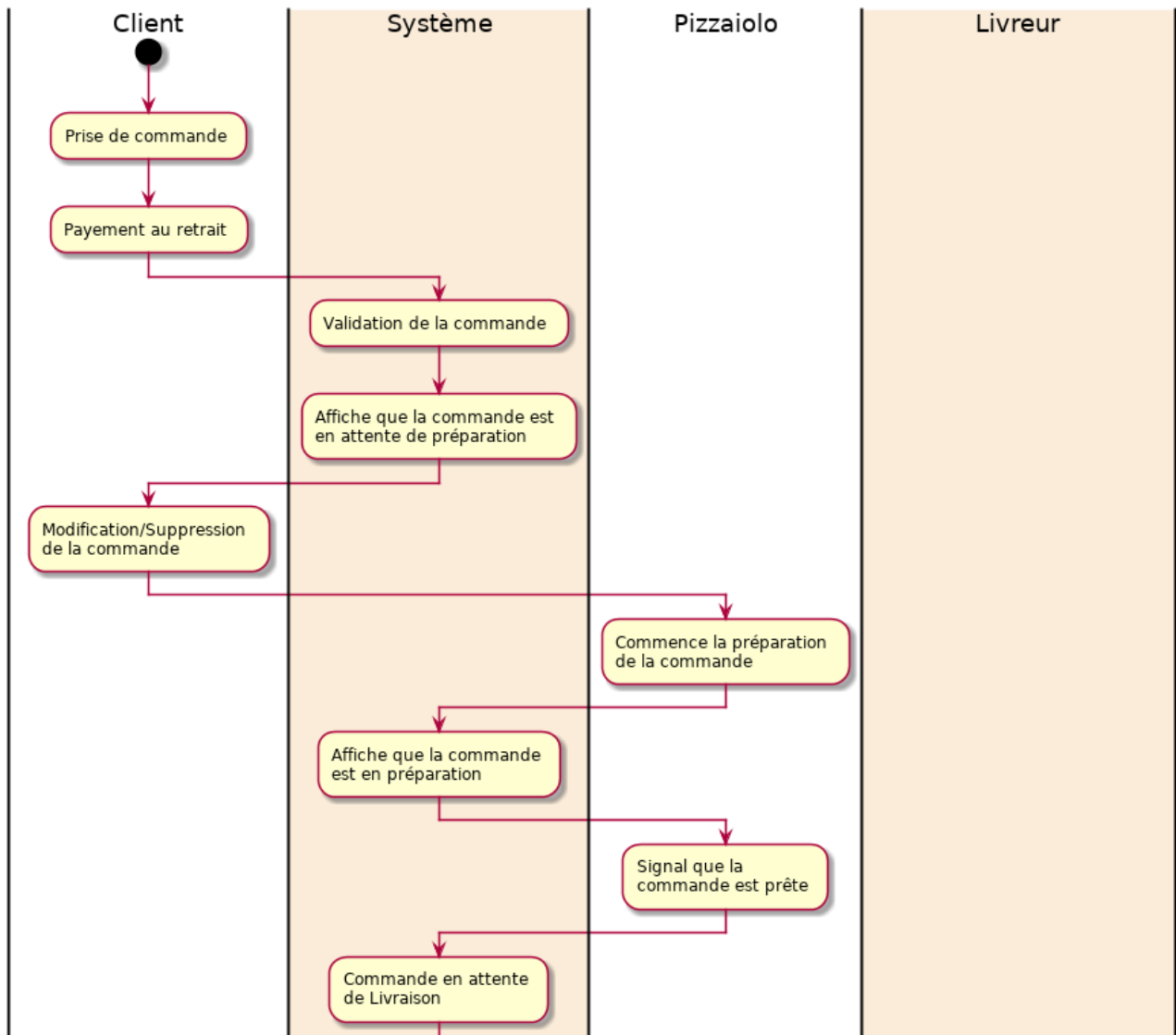
Ceci est le cycle de vie d'un produit lors du paiement en ligne. On peut voir toutes les étapes principaux par lequel tout les acteurs concerné vont devoir effectué.

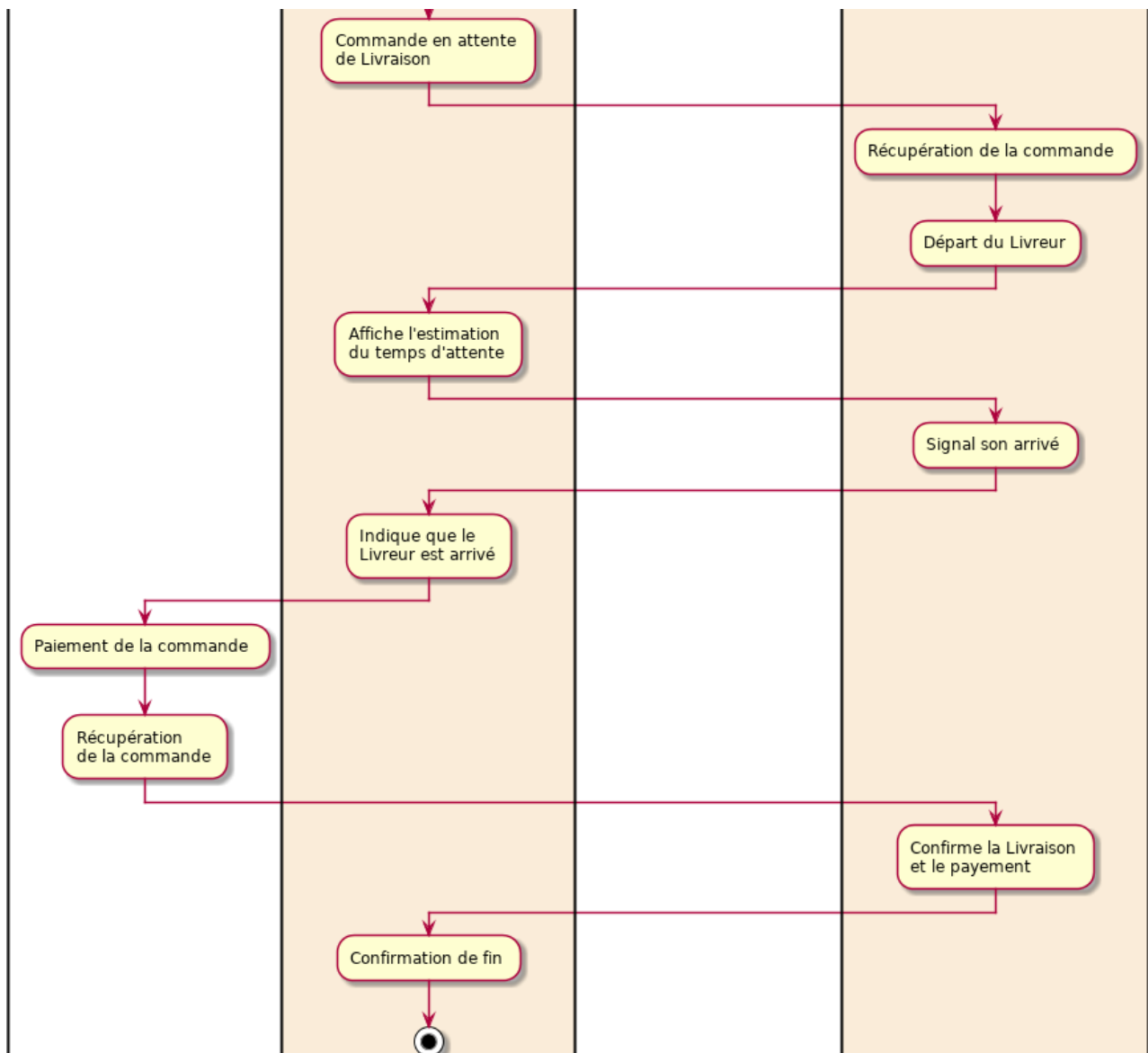




4.1.2 - Paiement au retrait

Ceci est le cycle de vie d'un produit lors du paiement au retrait. On peut voir toutes les étapes principaux par lequel tout les acteurs concerné vont devoir effectué.





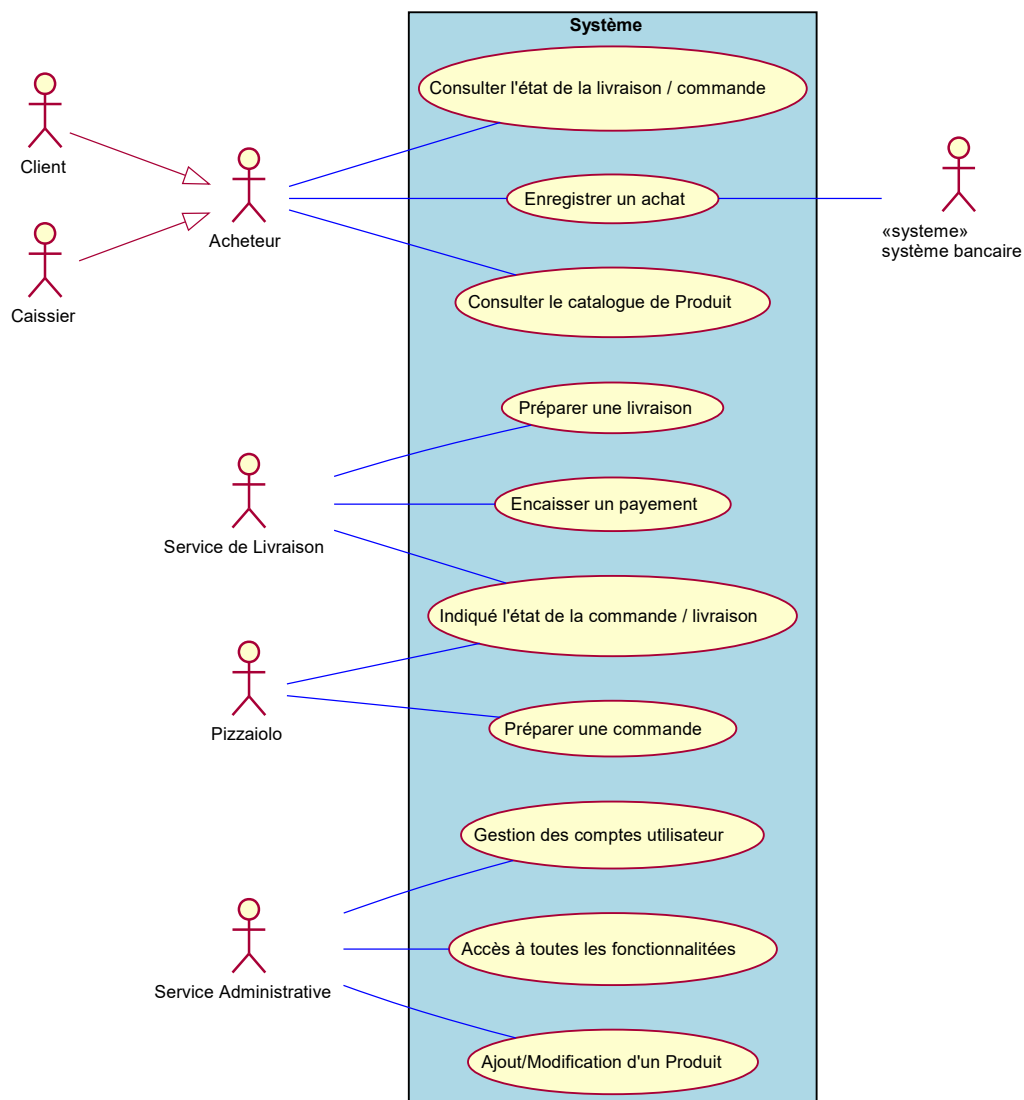
5 - APPLICATION WEB

5.1 - Les acteurs

Les acteurs représenté dans ce chapitre sont les même que ceux décrit dans la section 3.2 dédié à la description de tout les acteurs de l'application.

5.2 - Les cas d'utilisation

5.2.1 - Cas d'utilisation générale



L'acheteur souhaite consulter :

- l'état de la commande et de la livraison
- le catalogue de produit

L'acheteur doit pouvoir enregistrer un achat

Le Service de livraison doit pouvoir encaisser un paiement.

Le Service de livraison souhaite

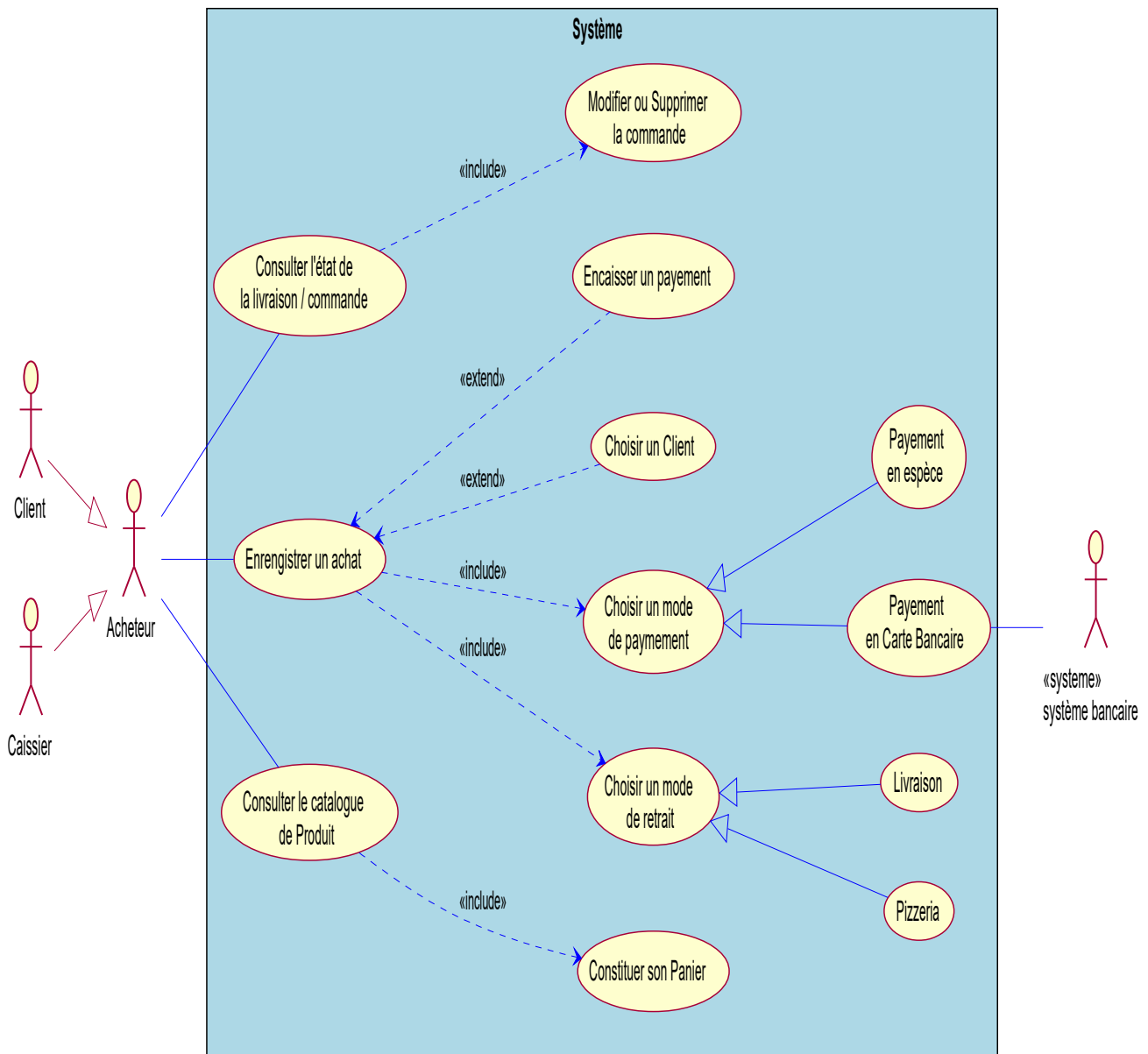
- préparer une livraison.
- indiquer l'état de la commande et de la livraison

Le Pizzaiolo souhaite préparer une commande et indiquer l'état de la commande et de la livraison.

Le service administrative doit pouvoir

- gérer les comptes utilisateur.
- ajouter et modifier un produit.
- accéder à toutes les fonctionnalité.

5.2.2 - Cas d'utilisation du client



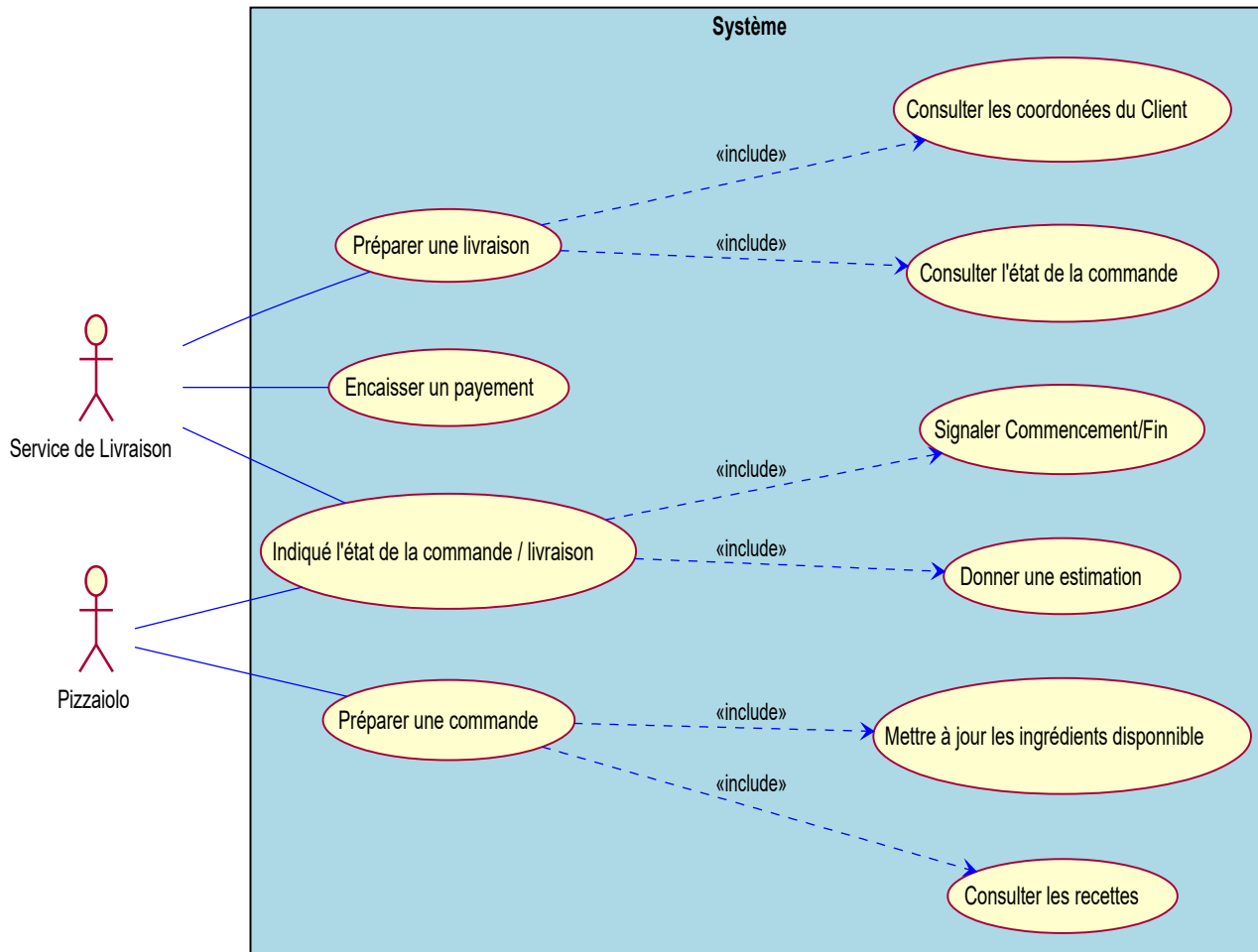
L'acheteur doit pouvoir :

- modifier ou supprimer sa commande
- choisir un mode de paiement
- choisir un mode de retrait
- constituer son panier

Le Caissier doit pouvoir :

- choisir un client
- Encaisser un paiement

5.2.3 - Cas d'utilisation des commandes



Le service de livraison doit pouvoir :

- consulter les coordonnées du client
- consulter l'état de la commande.

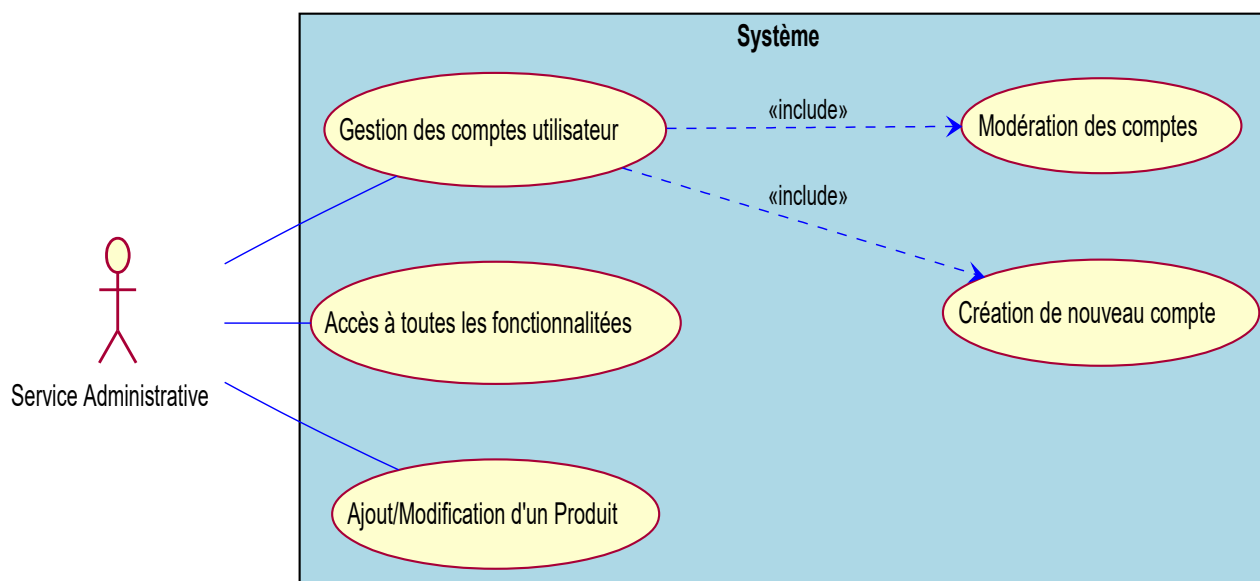
Le pizzaiolo doit pouvoir :

- mettre à jour les ingrédients disponible
- consulter les recettes

Le pizzaiolo et le service de livraison doit pouvoir :

- signaler le commencement et la fin d'une préparation
- donner une estimation

5.2.4 - Cas d'utilisation de l'administration



Le service Administrative doit pouvoir :

- créer un nouveau compte
- modifier le droit d'accès d'un compte

5.3 - Les règles de gestion générales

Les différentes règles générale de l'application sont les suivantes :

- Une commande peut contenir une à plusieurs pizza.
- L'utilisateur ne peut modifier ou annuler sa commande uniquement si sa commande n'est pas en préparation.

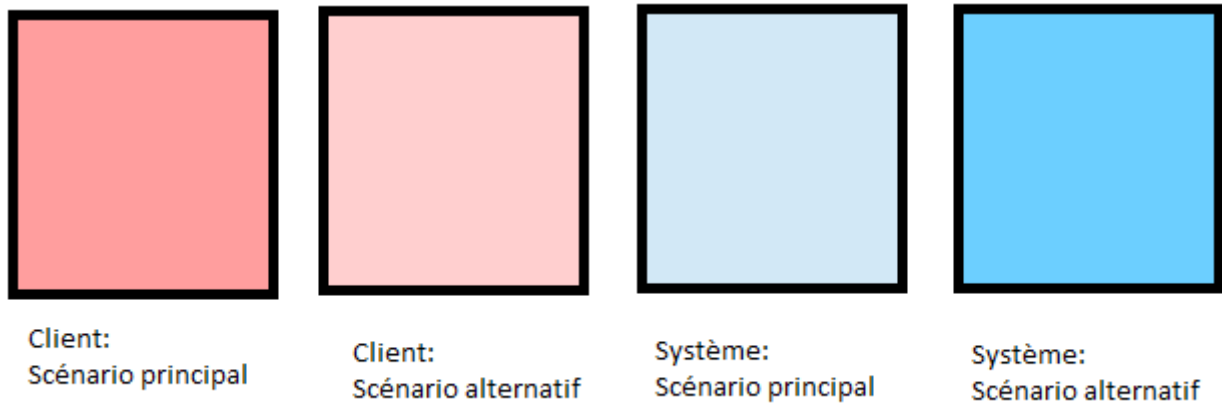
5.4 - Workflow

5.4.1 - Scénario d'une prise de commande en ligne:

Ce schéma représente le scénario de l'utilisateur lors de la prise de commande via notre application web.

Nous avons les chemins et les choix possible de l'utilisateur.

5.4.1.1 - Légendes



5.4.1.2 - Scénario

