

OC Pizza

Système de gestion informatique des pizzerias du groupe OC Pizza

Dossier de conception fonctionnelle

Version 1.0

Auteur

David Bouzerar

Développeur Backend

IT Consulting & Development



IT Consulting & Development

TABLE DES MATIERES

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 - Diagramme de contexte	5
2.3.3 - Enjeux et Objectifs	6
3 - Description générale de la solution	8
3.1 - Les principes de fonctionnement.....	8
3.2 - Les acteurs	8
3.3 - Les cas d'utilisation	9
3.3.1 - Cas d'utilisation général :.....	10
3.3.2 - Cas d'utilisation « Authentification » :	12
3.3.3 - Cas d'utilisation « Gestion de commande » (vue client).....	14
3.3.4 - Cas d'utilisation « Gestion de commande » (vue pizzaiolo).....	16
3.3.5 - Cas d'utilisation « Gestion de commande » (vue livreur)	17
3.3.6 - Cas d'utilisation « Gestion de commande » (scénario alternatif).....	19
4 - Le domaine fonctionnel	21
4.1 - Diagramme de classes.....	21
4.1.1 - Règles de gestion	26
4.2 - Packages	27
5 - Les workflows	29
5.1 - Le workflow « Gestion de commande » (Client)	29
5.2 - Le workflow « Gestion de commande » (Pizzaiolo).....	30
5.3 - Le workflow « Gestion de commande » (Livreur)	33
5.4 - Le workflow « Gestion de commande » (Cycle de vie).....	34
6 - Application « OC Pizza »	38
7 - Glossaire.....	41

1 - VERSIONS

Auteur	Date	Description	Version
David Bouzerar	16/03/2021	Création du document	0.1
David Bouzerar	16/03/2021	Ajout des propriétés	0.3
David Bouzerar	16/03/2021	Ajout de l'introduction	0.4
David Bouzerar	18/03/2021	Ajout description générale de la solution	0.7
David Bouzerar	04/04/2021	Finalisation du document	1.0

2 - INTRODUCTION

2.1 - Objet du document

Le présent document constitue le dossier de conception fonctionnelle de l'application OC Pizza. Cette solution est destinée à remplacer le système informatique utilisé actuellement en apportant, notamment, de nouvelles fonctionnalités inexistantes ou obsolètes sur le produit courant.

L'objectif de ce document est de présenter la solution fonctionnelle que nous proposons pour répondre aux besoins exprimés par le client.

2.2 - Références

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

1. **DCT-02** : Dossier de conception technique de l'application
2. **DEXP-03** : Dossier d'exploitation de l'application
3. **PV-04** : PV de livraison

2.3 - Besoin du client

2.3.1 - Contexte

OC Pizza est un jeune groupe de pizzeria en plein essor et spécialisé dans la vente en livraison 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.

A la demande des responsables du groupe, il nous est demandé de mettre en place un système informatique sur-mesure et qui sera déployé dans toutes les pizzerias du groupe. Celui-ci doit permettre notamment :

- D'augmenter l'efficacité dans la gestion de commandes, de la réception à la livraison et 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 afin de savoir quelles pizzas sont encore réalisables

- De proposer un site internet afin que les clients puissent :
 - Passer leurs commandes
 - Payer en ligne s'ils le souhaitent, le cas échéant ils paieront lors de la livraison
 - Modifier ou annuler une commande tant que celle-ci n'est pas prête
- 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 - Diagramme de contexte

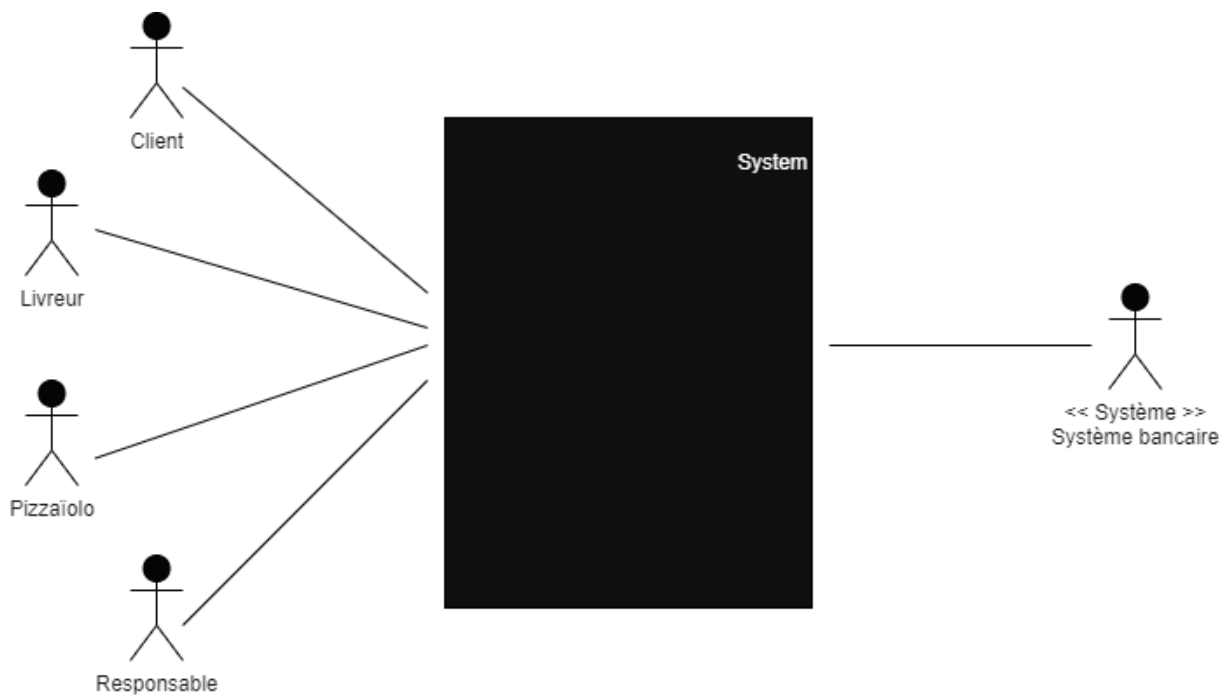


Diagramme de contexte

2.3.3 - Enjeux et Objectifs

L'application est, dans ce diagramme, représentée par le rectangle noir (**System**).

Les acteurs principaux (à gauche du rectangle) regroupent :

- **Le client :**

Préalablement « **visiteur** », il devient « **client** » lorsqu'il est authentifié sur l'application. Il peut :

- Consulter la carte des produits
- Effectuer une commande
- Choisir de régler sa commande en ligne ou lors de la livraison

- **Le pizzaïolo :**

- Peut consulter les commandes validées par les clients afin de les préparer
- Peut indiquer que les commandes sont prêtes à être livrées
- A également la possibilité de consulter l'**aide-mémoire** contenant les différentes recettes des produits proposés

Dans le cas où le client passe commande directement dans un point de vente :

- Il peut saisir une nouvelle commande et indiquer que celle-ci est réglée dans le point de vente

Dans le cas où le client passe commande par téléphone :

- Il peut saisir une nouvelle commande et indiquer que celle-ci sera réglée dans le point de vente ou lors de la livraison

- **Le livreur :**

- Peut consulter les commandes déjà préparées, qui sont donc éligibles à la livraison
- Dispose d'un GPS (**Google Maps**) pour se rendre à l'adresse de livraison
- Peut signaler qu'une commande a été livrée

- Le responsable :
 - Peut consulter l'activité des différents points de vente
 - Peut modifier le contenu de l'**aide-mémoire**
 - Peut modifier la carte des produits proposés
 - Peut consulter l'activité des employés

- Le système bancaire :
 - Le **système bancaire**, en tant qu'acteur secondaire, sera sollicité lors du règlement d'une commande en ligne par le client

3 - DESCRIPTION GENERALE DE LA SOLUTION

3.1 - Les principes de fonctionnement

La solution qui sera mise en place consiste à développer une **application WEB** selon les besoins exprimés précédemment. Cette application disposera d'un site internet depuis lequel, les clients authentifiés, pourront consulter la carte des produits disponibles, ainsi qu'éventuellement, passer une commande qui pourra être livrée, ou, retirée depuis un point de vente. A savoir que dans ce cas, les clients pourront suivre l'évolution de leur commande, ainsi qu'éventuellement l'annuler si c'est encore possible.

D'autre part, l'**application WEB** contiendra également une partie « **Administration** » destinée à être utilisée par les employés de la société. Les responsables pourront consulter l'état des différents points de vente, ou encore modifier la carte visible sur le site internet ou les recettes renseignées dans l'aide-mémoire.

Les Pizzaiolos pourront eux consulter les commandes disponibles dans leur point de vente et les prendre en charge. Ils auront également accès à l'aide-mémoire s'ils ont besoin de consulter une recette.

Pour terminer, les livreurs pourront voir les commandes une fois préparées par les Pizzaiolos et les prendre en charge. Ils seront également guidés vers l'adresse de livraison, par l'intermédiaire de l'**API GoogleMaps**.

3.2 - Les acteurs

- **Le visiteur / client :**

Peut consulter la carte des pizzas et a également la possibilité de passer une commande. Il doit pour cela :

- I. **En tant que visiteur :**

- A. *S'authentifier (**Nécessite une inscription**)*

- B. *Il devient alors client*

II. En tant que client :

- A. *Valider son panier*
- B. *Saisir les informations de livraison*
- C. *Saisir type de règlement (en ligne, lors de la livraison ou dans un point de vente)*
- D. *Valider le paiement (si en ligne)*

La participation de l'acteur secondaire « **système bancaire** » est nécessaire pour ce dernier point (II. D).

▪ Le pizzaiolo :

Le pizzaiolo a lui accès à la fonctionnalité « **Préparation de commande** ». Elle lui offre la possibilité de prendre en charge des commandes validées par les clients. Il peut également consulter les recettes des produits disponibles depuis l'**aide-mémoire** ainsi que l'inventaire des ingrédients disponibles.

▪ Le livreur :

Le livreur peut consulter les commandes éligibles à la livraison et les prendre en charge. Il dispose également d'un GPS (**API GoogleMaps**)

▪ Le responsable :

Le responsable peut suivre l'activité des différents points de vente ainsi que l'inventaire « ingrédients » de ces derniers. Il est également celui qui modifie la carte des produits et l'**aide-mémoire**.

3.3 - Les cas d'utilisation

3.3.1 - Cas d'utilisation général :

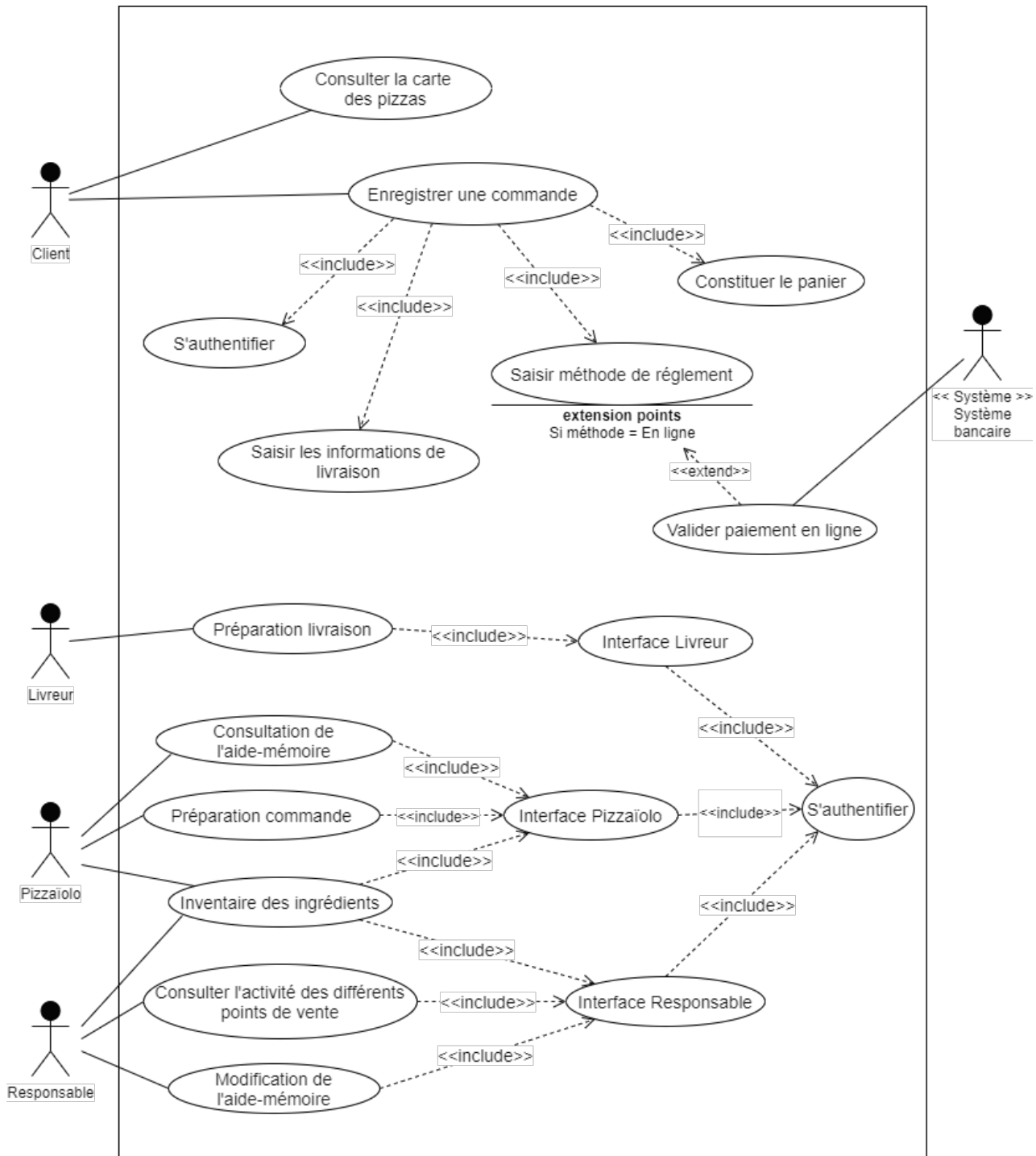


Diagramme de usecase général

Nous pouvons voir dans ce diagramme les différentes fonctionnalités utilisables par les différents acteurs :

- **Le visiteur / client :**

Peut consulter la carte des pizzas et a également la possibilité de passer une commande. Il doit pour cela :

- I. **En tant que visiteur :**

- A. *S'authentifier (Nécessite une inscription)*

- B. *Il devient alors client*

- II. **En tant que client :**

- A. *Valider son panier*

- B. *Saisir les informations de livraison*

- C. *Saisir type de règlement (en ligne, lors de la livraison ou dans un point de vente)*

- D. *Valider le paiement (si en ligne)*

La participation de l'acteur secondaire « **système bancaire** » est nécessaire pour ce dernier point (II. D).

- **Le pizzaïolo :**

Le pizzaïolo a lui accès à la fonctionnalité « **Préparation de commande** ». Elle lui offre la possibilité de prendre en charge des commandes validées par les clients. Il peut également consulter les recettes des produits disponibles depuis l'**aide-mémoire** ainsi que l'inventaire des ingrédients disponibles.

- Le livreur :

Le livreur peut consulter les commandes éligibles à la livraison et les prendre en charge. Il dispose également d'un GPS (**API Google Maps**)

- Le responsable :

Le responsable peut suivre l'activité des différents points de vente ainsi que l'inventaire « ingrédients » de ces derniers. Il est également celui qui dispose des droits pour modifier la carte des produits ainsi que l'**aide-mémoire**.

3.3.2 - Cas d'utilisation « Authentification » :

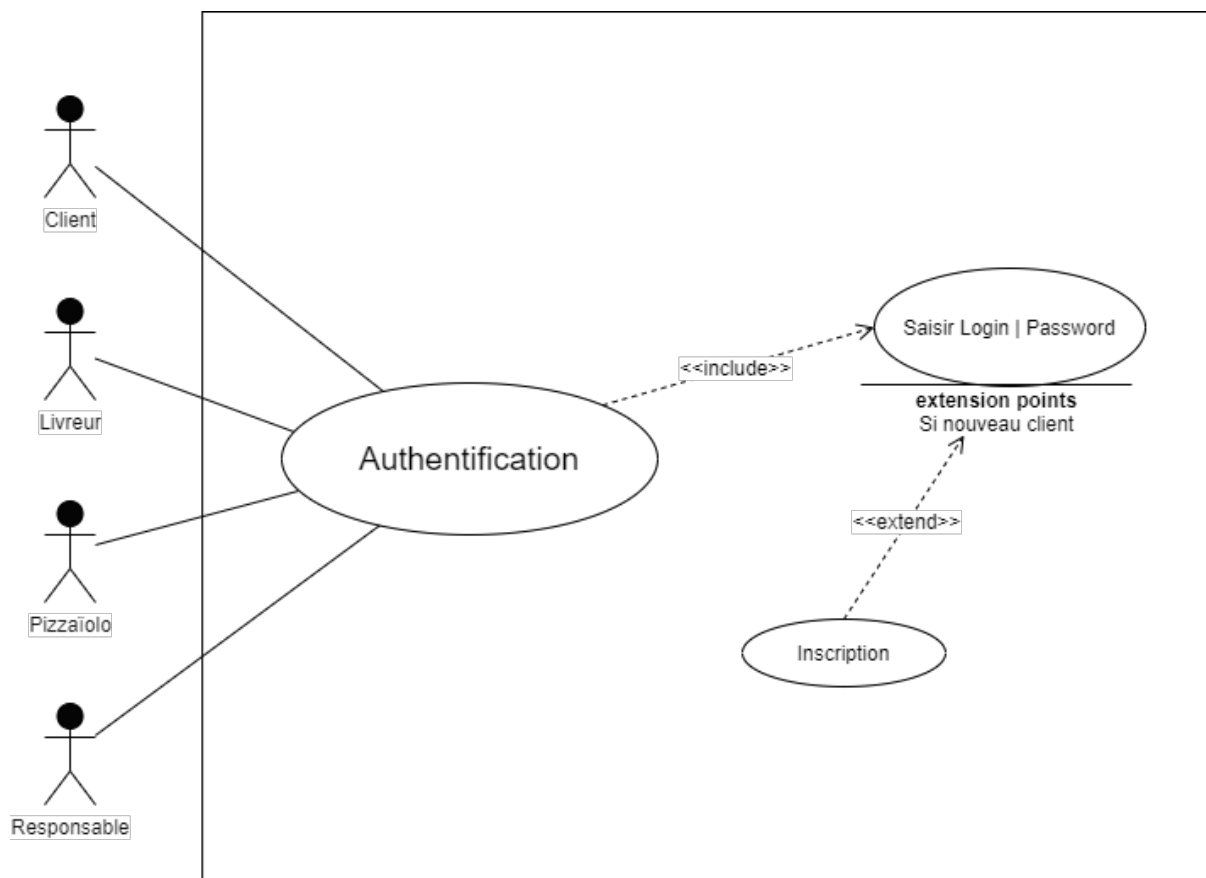


Diagramme de usecase « Authentification »

Ce package est utilisé par tous les acteurs principaux.

- **Le visteur/client :**

Il peut s'authentifier uniquement après avoir validé le formulaire d'inscription. S'il possède déjà un compte, il peut s'authentifier en saisissant ses identifiants, sinon, il doit donc s'inscrire. Il a ensuite la possibilité de constituer un panier puis d'enregistrer une commande.

- **Les employés (pizzaïolo, livreur et responsable) :**

Concernant les employés, ils doivent s'authentifier grâce aux identifiants fournis par la société, de façon à pouvoir accéder aux différentes fonctionnalités, selon les droits octroyés.

3.3.3 - Cas d'utilisation « Gestion de commande » (vue client)

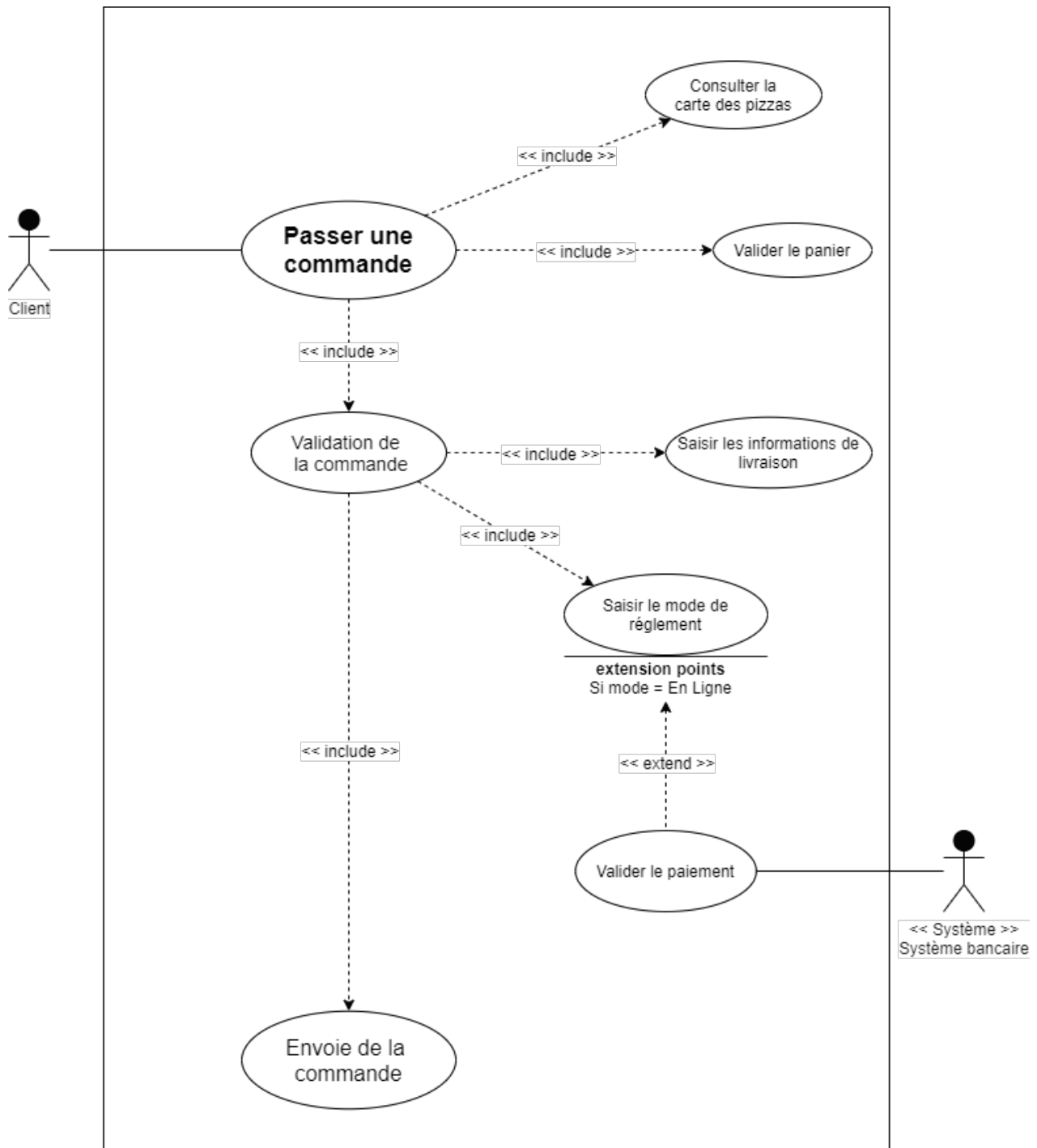


Diagramme de usecase « Gestion de commande » côté client

Le diagramme ci-dessus, permet de visualiser le cas d'utilisation de la gestion des commandes côté client.

Rappel : Avant de devenir « *client* », l'acteur est un visiteur. L'appellation « *client* » indique que le « *visiteur* » est authentifié sur la plateforme.

Voici les différentes étapes nécessaires au passage d'une commande :

- Le client consulte la carte des pizzas
- Il constitue ensuite son panier, puis le valide
- Il est alors invité à renseigner ses informations de livraison
- Il renseigne le mode de règlement (**en ligne** ou à la livraison) :
 - Si le paiement en ligne est sélectionné, il est alors orienté vers le système bancaire pour renseigner ses informations de paiement
 - Si le paiement se fait lors de la livraison, le processus se poursuit
- La commande est traitée et orientée vers le point de vente le plus proche

3.3.4 - Cas d'utilisation « Gestion de commande » (vue pizzaiolo)

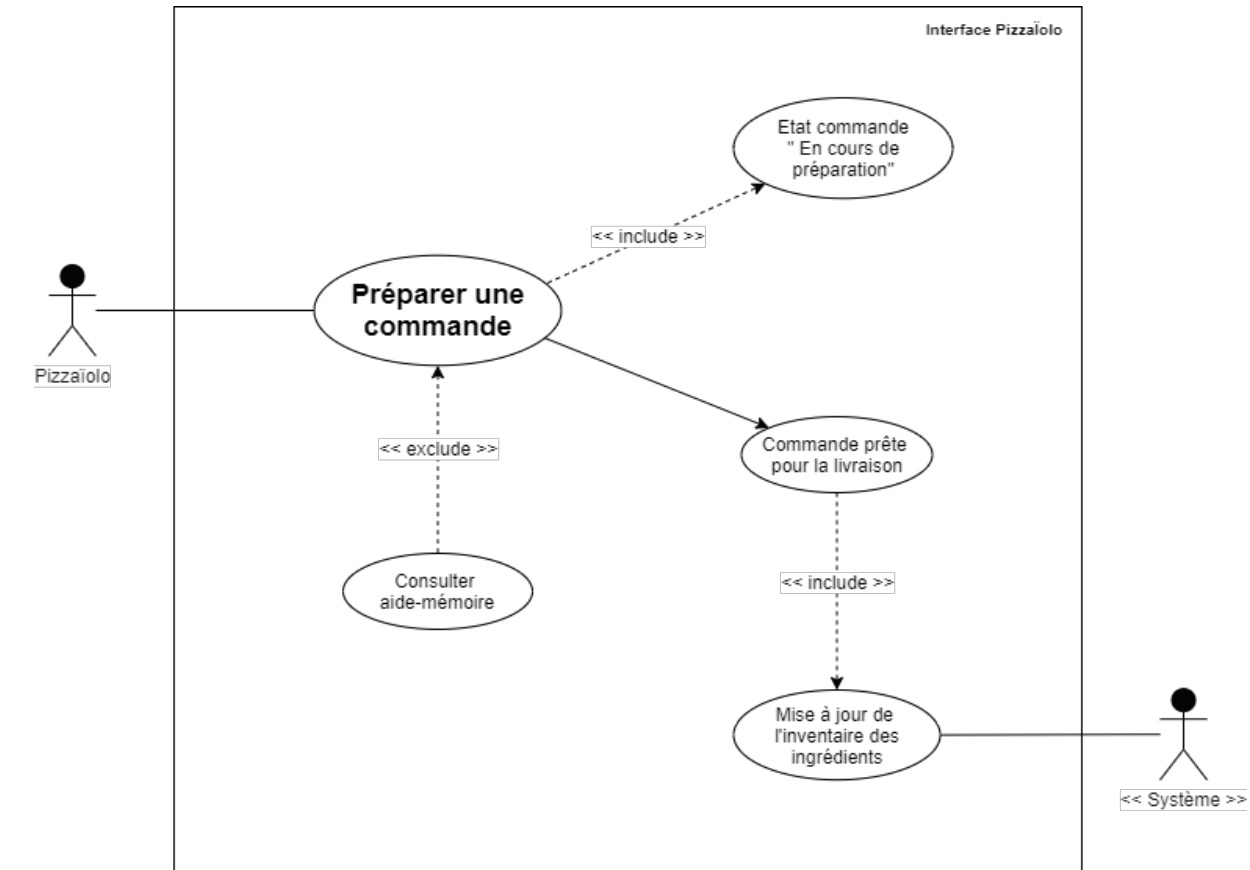


Diagramme de usecase « Gestion de commande » vue pizzaiolo

Le diagramme ci-dessus, nous permet d'observer le cas d'utilisation de la gestion des commandes côté pizzaiolo.

Rappel : Nous partons du principe qu'une commande a été passée par un client (illustré dans le diagramme par « *En cours de préparation* »).

- Le pizzaiolo prend en charge la commande
- Il consulte l'**aide-mémoire** (si nécessaire) pour s'assurer de la bonne application de la recette
- La commande est prête pour la livraison
- L'inventaire est mis à jour, selon les produits préparés pour la commande

3.3.5 - Cas d'utilisation « Gestion de commande » (vue livreur)

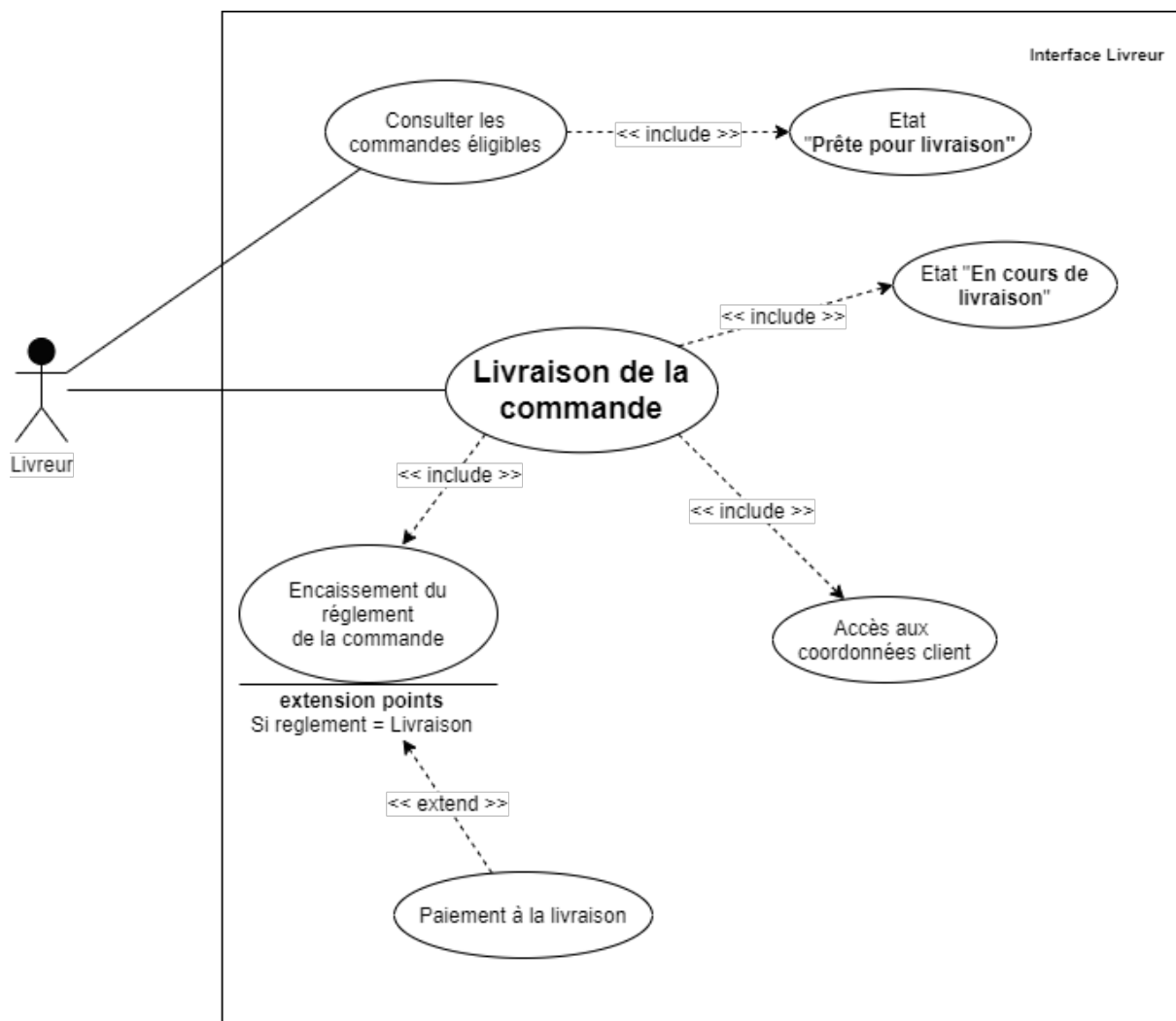


Diagramme de séquence « Gestion de commande » vue livreur

Le diagramme ci-dessus nous permet d'observer le cas d'utilisation de la gestion des commandes côté Livreur.

Dans le cas présent, nous partons du principe qu'une commande a été préparée par le pizzaiolo (« En cours de livraison »).

Voici les différentes étapes :

- Le livreur consulte les commandes éligibles à la livraison
- Le livreur prend en charge la commande
- Il a ensuite accès aux informations (coordonnées, état du règlement...)
- La livraison se termine lorsque le livreur remet la commande au client (et éventuellement encaissé la commande)

3.3.6 - Cas d'utilisation « Gestion de commande » (scénario alternatif)

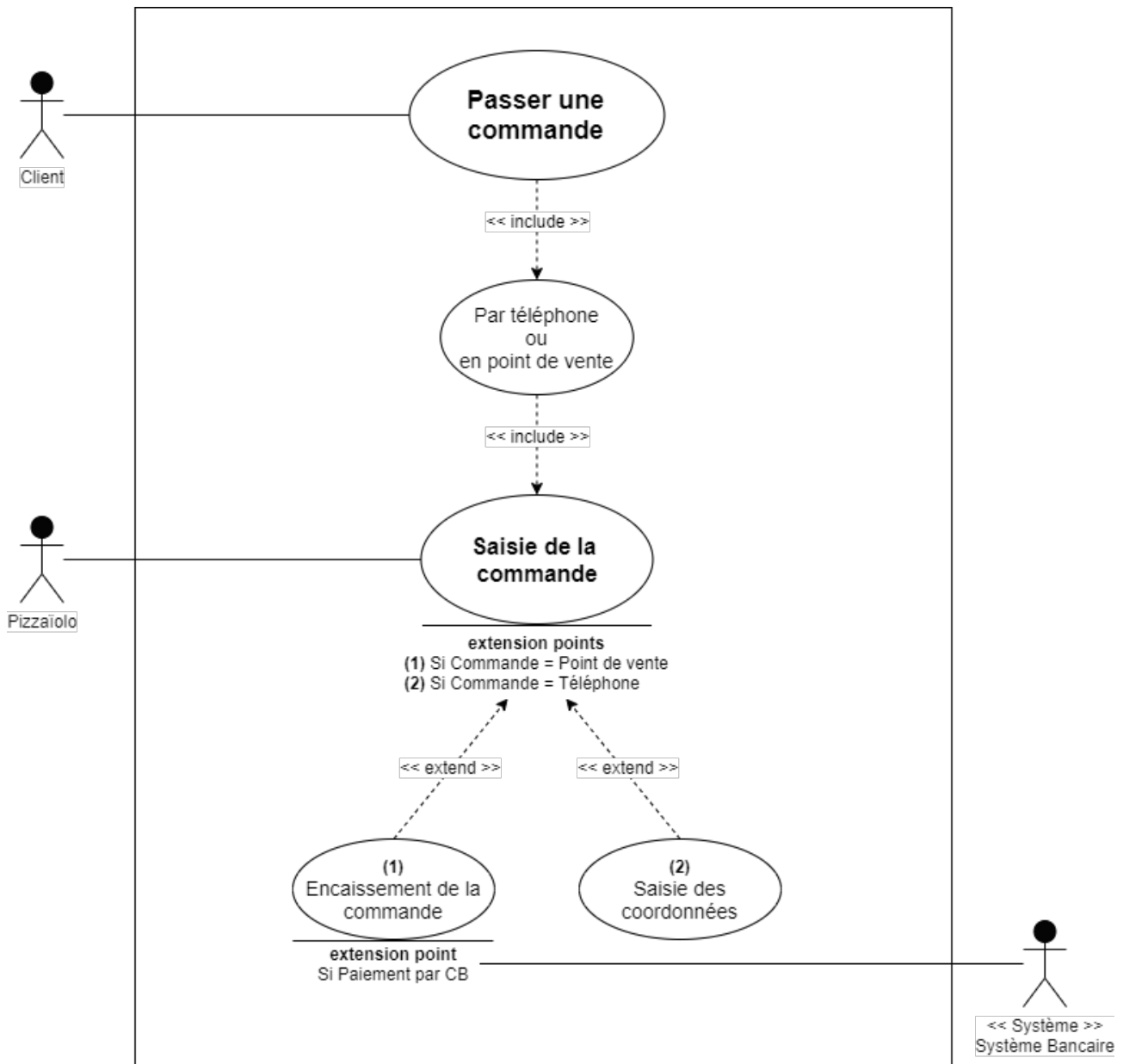


Diagramme de usecase « Gestion de commande » alternatif

Le diagramme ci-dessus, permet donc de visualiser le scénario alternatif lors du passage d'une nouvelle commande. Comme expliqué plus tôt, cela concerne principalement le client et le pizaiolo, et se positionne au niveau de la création de la commande dans le cycle de vie.

Rappel : La suite du processus rejoint le scénario principal, notamment lors de la prise en charge de la commande (en vue de la préparation) par le pizaiolo.

Voici donc le déroulement :

- Le client passe commande :
 - (1) En point de vente
 - (2) Par téléphone
- Le pizaiolo saisie la commande directement sur l'application
- Dans le cas (1), il s'occupe de l'encaissement de la commande :
 - Si le client souhaite régler par carte bancaire, le **Système bancaire** est sollicité
- Dans le cas (2), si le client souhaite être livré, le pizaiolo saisit les coordonnées du client. Le règlement de la commande se fera alors par le livreur.
- La commande est créée, son état passe « **En attente** »

Le scénario alternatif rejoint le scénario principal lors de la prise en charge par le pizaiolo « **Vue Pizaiolo** »

4 - LE DOMAINE FONCTIONNEL

4.1 - Diagramme de classes

```

classDiagram
    class Contact {
        +phone: String
        +number: String
        +street: String
        +addressComplement: String
        +postalCode: String
        +city: String
    }
    class User {
        +title: String
        +lastName: String
        +firstName: String
        +email: String
        +login: String
        +password: String
    }
    class Employee {
        +role: Role
    }
    class Customer {
        +clientNumber: Integer
    }
    class Item {
        +quantity: Integer
        +unitPrice: Real
        +rateVAT100: Real
        +pizza: Pizza
    }
    class Pizza {
        +name: String
        +description: String
        +priceHT: Real
        +rateVAT100: Real
    }
    class ShoppingCart {
        +priceVAT: Real
        +delivery: Boolean
    }
    class Order {
        +number: String
        +state: State
        +paidOnLine: Boolean
        +totalHT: Real
        +totalVAT: Real
        +delivery: Boolean
    }
    class Ingredient {
        +name: String
    }
    class Restaurant {
        +name: String
        +email: String
    }
    class Stock {
        +quantity: Integer
    }
    class Reminder {
        +recipe: String
        +description: String
    }
    class Bill {
        +number: Integer
        +totalHT: Boolean
        +totalVAT: Real
    }
    class Payment {
        +type: Solution
        +totalVAT: Real
    }
    class Role {
        <<enumeration>>
        Manager
        Pizza chef
        Delivery man
    }
    class State {
        <<enumeration>>
        awaiting for preparation
        preparation in progress
        awaiting for delivery
        delivery in progress
        order delivered
    }
    class Solution {
        <<enumeration>>
        Credit card
        Cash
        Check
    }

    Contact "1" -- "1" Customer
    Customer "1" -- "1" User
    Employee "1" -- "1" User
    Employee "1" -- "1" Role
    Customer "1" -- "1" Item
    Item "0..1" -- "1" ShoppingCart
    ShoppingCart "1" -- "1" Order
    Order "1" -- "1" User
    Order "1" -- "1" Pizza
    Order "1..*" -- "1" Item
    Order "1" -- "1" Bill
    Order "1" -- "1" Payment
    Pizza "*" -- "1" Ingredient
    Pizza "1" -- "1" Reminder
    Restaurant "*" -- "1" Ingredient
    Restaurant "*" -- "1" Stock
    Restaurant "1" -- "1" Bill
    Restaurant "1" -- "1" Payment
    Ingredient "1" -- "1" Stock
    Stock "1" -- "1" Restaurant
    Reminder "*" -- "1" Pizza
    Bill "1" -- "1" Restaurant
    Payment "1" -- "1" Restaurant
    
```

Le diagramme ci-dessus représente les classes qui seront utilisées dans le cadre de notre solution technique.

Il peut être divisé en 3 parties distinctes, une première qui concernera la « **Gestion des utilisateurs** », une seconde pour la « **Gestion des commandes** » et une dernière pour la « **Gestion des produits** ».

Voici les classes que ces différentes parties contiennent :

- **Gestion des utilisateurs :**

- **User** : Classe mère utilisée pour identifier tous les utilisateurs, que ce soit les clients (classe héritée « **Customer** ») ou les employés (classe héritée « **Employee** »). La classe « **Employee** » possède un attribut « **role** » permettant d'identifier le rôle d'un employé au sein de la société. Cet attribut est une instance de l'énumération « **Role** ».
- **Contact** : Classe qui est utilisée pour stocker les coordonnées des clients, mais également des différents points de vente (cinq actuellement, huit avant la fin de l'année).

- **Gestion des commandes :**

- **Item** : Dans cette partie, une première classe « **Item** » (liée également à la classe « **Pizza** ») est utilisée lorsque le client ajoute un produit dans son panier. On retrouve dans les attributs de cette classe, notamment, la quantité du produit ajouté, le prix à l'unité, le taux TVA ainsi que le produit lui-même (instance de la classe « **Pizza** »).
- **ShoppingCart** : Cette classe représente le panier du client et contient les produits ajoutés par le client (instance(s) « **Item** »). Dans cette classe, un attribut booléen « **delivery** » aura une valeur « **true** » ou « **false** » selon que le client choisisse de se faire livrer la commande ou pas.

- **Order** : Cette classe est instanciée lorsque la commande est validée par le client. Elle contient un numéro unique et les différentes informations de la commande passée. En plus de l'attribut « **delivery** » qui concerne la livraison de la commande, un autre booléen apparaît ici, « **paidOnLine** ». Cet attribut varie selon que la commande soit déjà payée (en ligne) ou pas, notamment dans le cadre d'une livraison ou de la récupération d'une commande dans un point de vente. Un autre attribut nommé « **state** » varie selon l'état d'avancement de la commande. Les différents états sont contenus dans l'énumération « **State** ».
- **Bill / Payment** : Ces deux classes concernent la facturation de la commande au client. La classe « **Payment** » est utilisée lors du paiement de la commande et contient comme attribut « **type** » qui est la solution de paiement choisie par le client (*carte de crédit, espèce, chèque*). Les différentes solutions sont contenues dans une énumération nommée « **Solution** ».

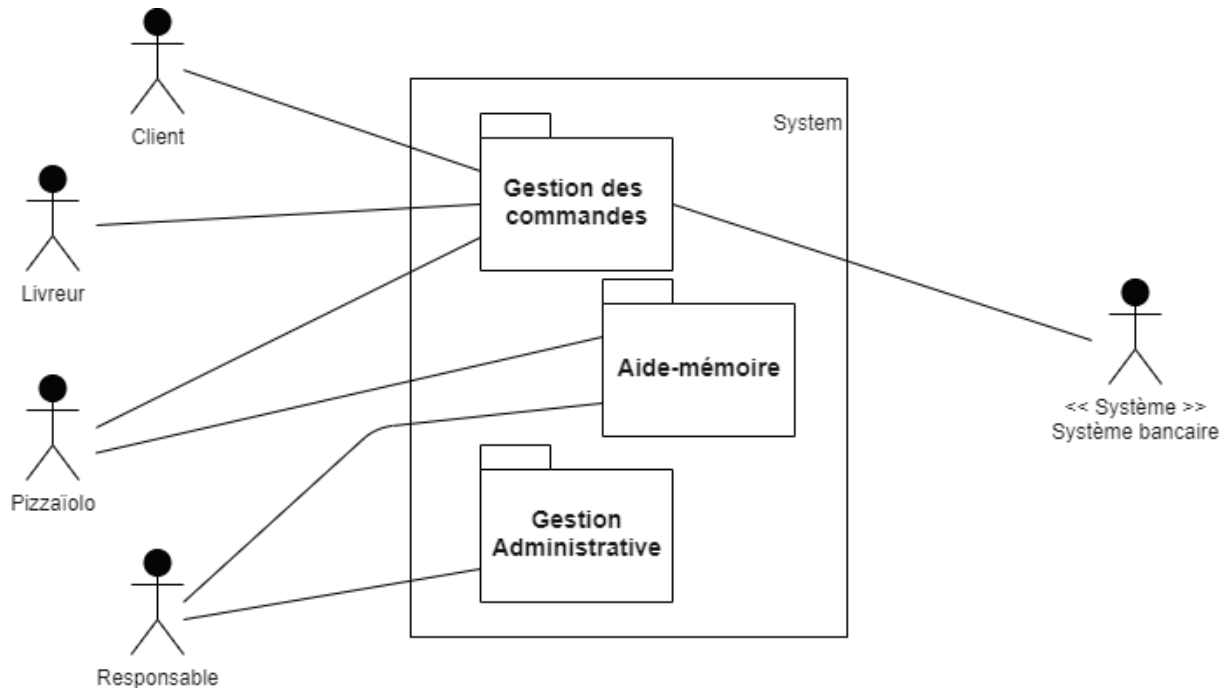
Une autre classe nommée « **Bill** » concerne la facture de la commande. Elle permet de garder un historique des commandes traitée, elle est également liée au point de vente qui a traité la commande.
- **Gestion des produits :**
 - **Pizza** : Cette classe est commune aux différentes parties énoncées. Elle est instanciée notamment lors de l'ajout de produit dans le panier et contient les informations des produits.
 - **Ingredient / Stock** : Ces deux classes concernent l'inventaire des ingrédients nécessaires à la confection des pizzas. La classe « **Stock** » est, quant à elle, liée à un point de vente en particulier.
 - **Reminder** : La classe « **Reminder** » est utilisée comme aide-mémoire et contient les recettes des différentes pizzas proposées par le groupe. Elle est commune aux différents points de vente.

- **Restaurant** : Cette dernière est utilisée pour identifier les différents points de vente. Elle est également liée à la classe « **Contact** » pour obtenir ici les coordonnées des différents points de vente

4.1.1 - Règles de gestion

- Pour passer une commande, un client doit préalablement s'authentifier sur l'application. Il doit ensuite ajouter les produits souhaités dans son panier puis saisir les informations de livraison s'il souhaite se faire livrer.
- Pour préparer une commande, un pizzaiolo doit préalablement s'authentifier sur l'application. Il aura alors la possibilité de voir les commandes en attente de préparation depuis son interface. Une fois la commande prise en charge, il peut, s'il le souhaite, consulter les recettes des produits à préparer depuis l'aide-mémoire.
- Pour livrer une commande, un livreur doit préalablement s'authentifier sur l'application. Il peut voir alors les commandes qui sont en attente de livraison. Une fois la commande prise en charge, il est guidé vers l'adresse de destination par le biais de l'API **GoogleMaps** et peut consulter les informations de la commande, pour notamment connaître l'état de règlement de la commande.
- Pour consulter l'état des points de vente, un responsable doit préalablement s'authentifier sur l'application. Il peut alors consulter les informations des différents points de vente comme par exemple l'état de l'inventaire ou encore l'état des commandes. Il peut également modifier l'aide-mémoire des pizzaiolos ou encore la carte des produits que les clients peuvent consulter pour commander.

4.2 - Packages



Nous avons fait le choix de découper le « **système** » en trois **packages** différents :

- **Gestion des commandes :**

Le client, le livreur ainsi que le pizzaiolo seront amenés à interagir avec les différentes fonctionnalités de ce **package**. Le client lors du passage d'une commande depuis l'interface web. Le pizzaiolo lorsqu'il devra saisir certaines commandes et/ou les préparer. Le livreur, lors de la prise en charge des commandes préparées.

- **Gestion Administrative :**

L'interaction de ce **package** se fera uniquement avec le responsable de la société. Il pourra ici, notamment, observer l'activité des différents points de vente et saisir la carte des produits.

- Aide-mémoire :

Ce **package** concernera principalement le pizzaïolo et le responsable. La base sera également commune à tous les points de vente. Le pizzaïolo aura notamment la possibilité de consulter, les différentes recettes des produits proposés par l'enseigne et préalablement saisies par le responsable.

5 - LES WORKFLOWS

5.1 - Le workflow « Gestion de commande » (Client)

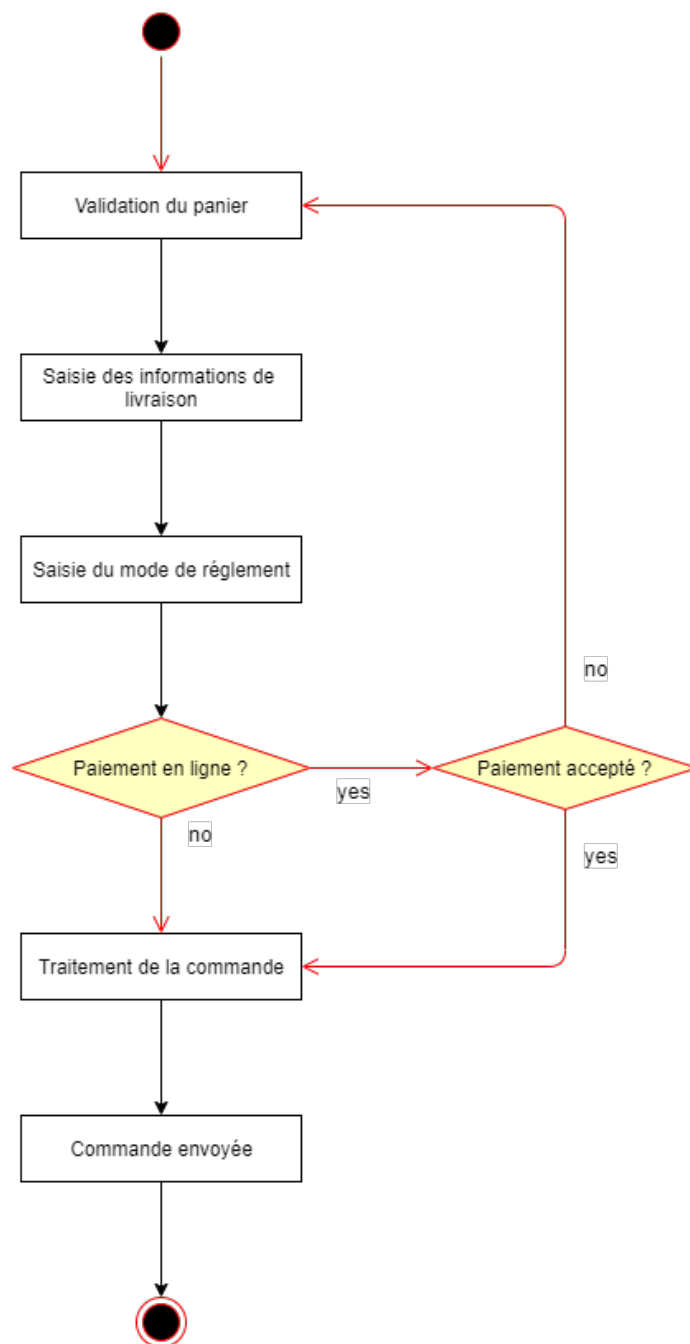


Diagramme d'activité « Gestion de commande » côté client

Voici le détail du diagramme d'activité proposé ci-dessus. Le client commence par constituer son panier :

- Il valide ensuite son panier
- Saisie les informations de livraison
- Choix du mode de règlement :

➤ **Paiement en ligne :**

- Le paiement est accepté, le processus de commande se poursuit.
- Le paiement est refusé, l'erreur est signalée au client, il est invité à payer à la livraison s'il le souhaite.

➤ **Paiement à la livraison :**

- Le processus de commande se poursuit

- La commande est ensuite traitée :
 - La commande est validée, elle est envoyée à la pizzeria
 - La commande est refusée, le client est informé de l'erreur et invité à recommencer le processus de commande
- Fin du processus de commande

5.2 - Le workflow « Gestion de commande » (Pizzaiolo)

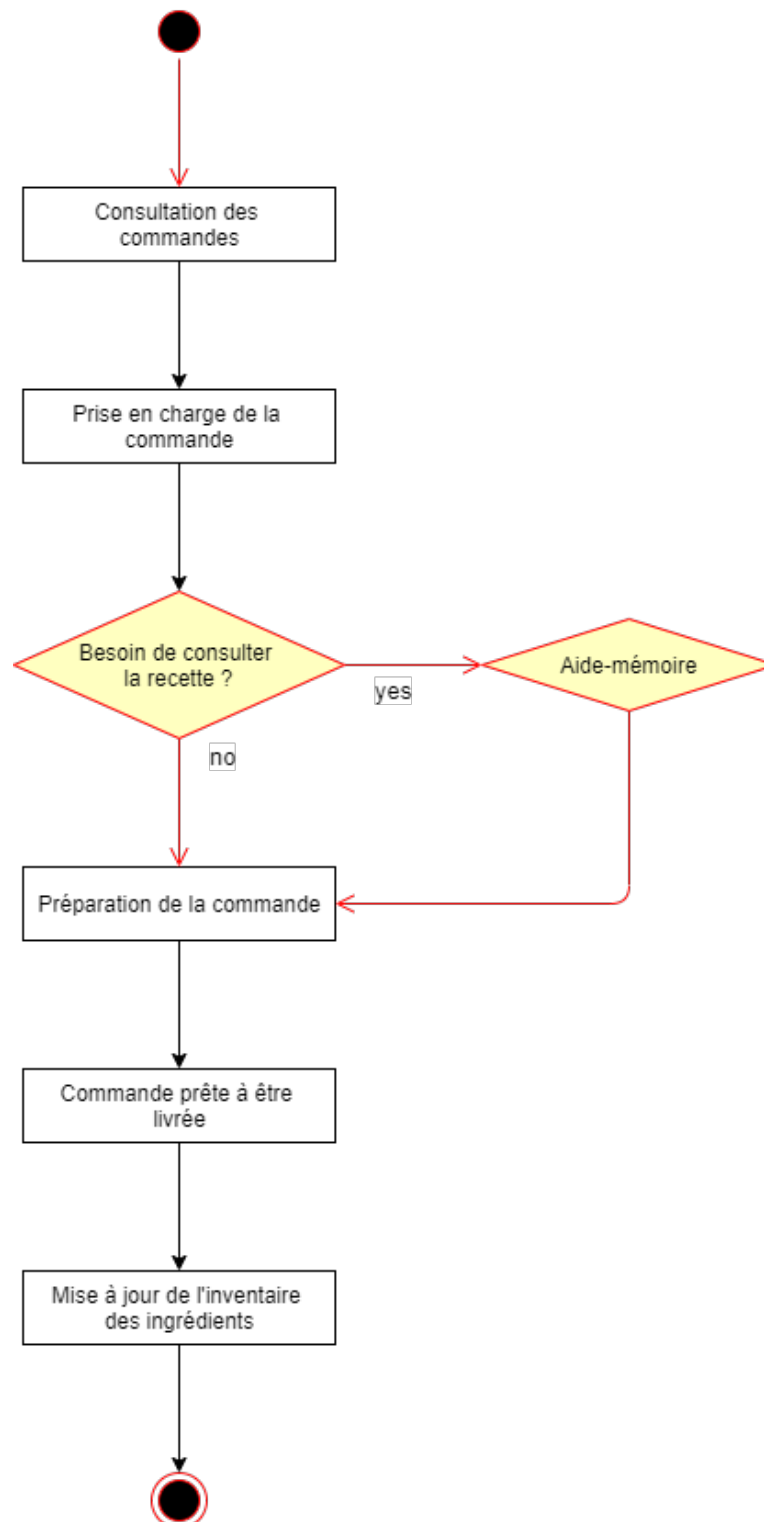


Diagramme d'activité « Gestion de commande » vue pizzaiolo

Voici, ci-dessous, le détail du diagramme d'activité proposé :

Rappel : Une fois validée, l'état de la commande passe à « *En cours de préparation* ».

- Le pizzaïolo prend la commande en charge
- Si nécessaire, il peut consulter la recette :
 - Il est dans ce cas redirigé vers l'aide-mémoire
- Le pizzaïolo prépare sa commande
- La commande est prête pour la livraison
- L'inventaire des ingrédients est mis à jour

5.3 - Le workflow « Gestion de commande » (Livreur)

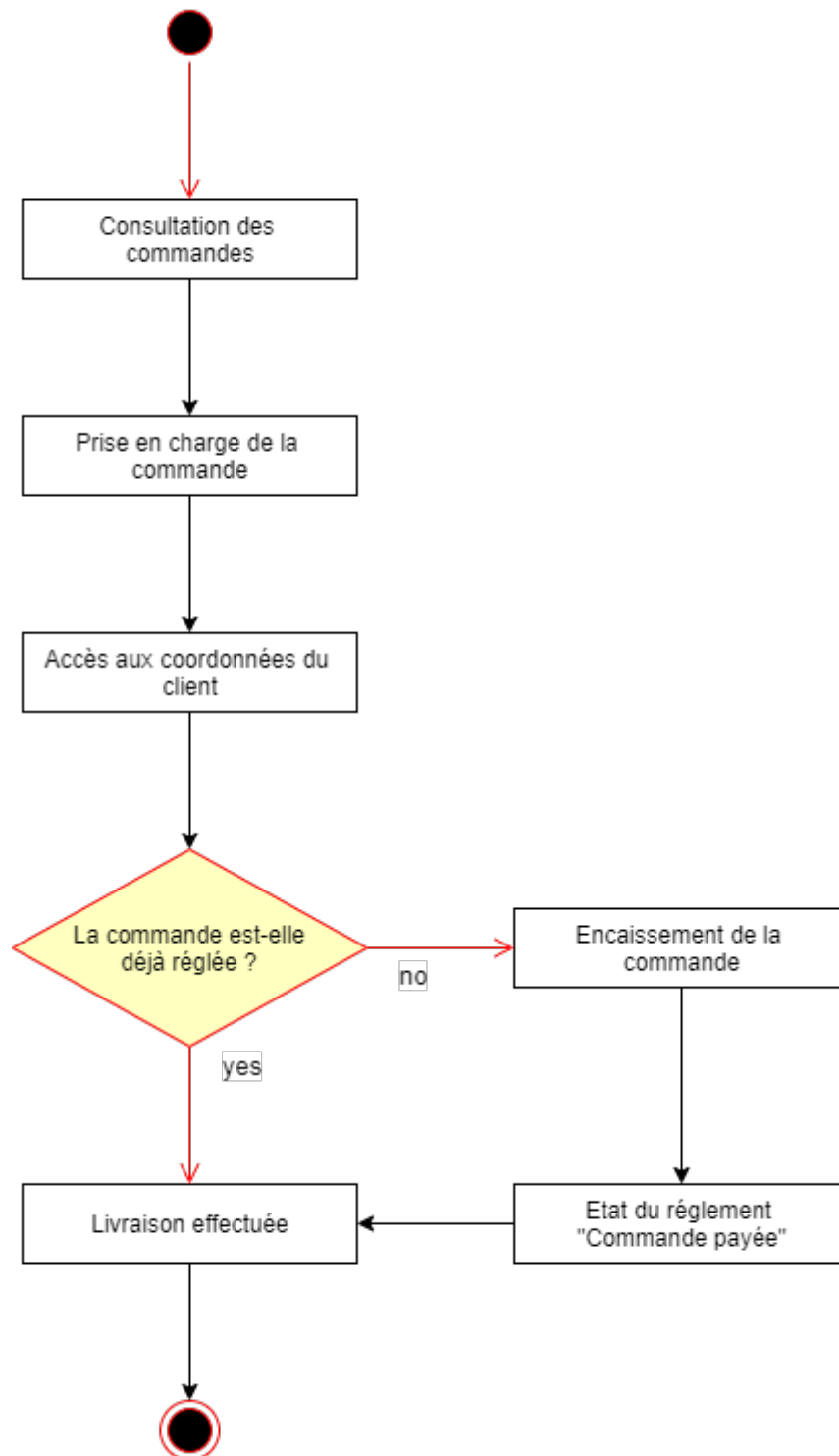


Diagramme d'activité « Gestion de commande » vue livreur

Voici le détail du diagramme d'activité proposé ci-dessous.

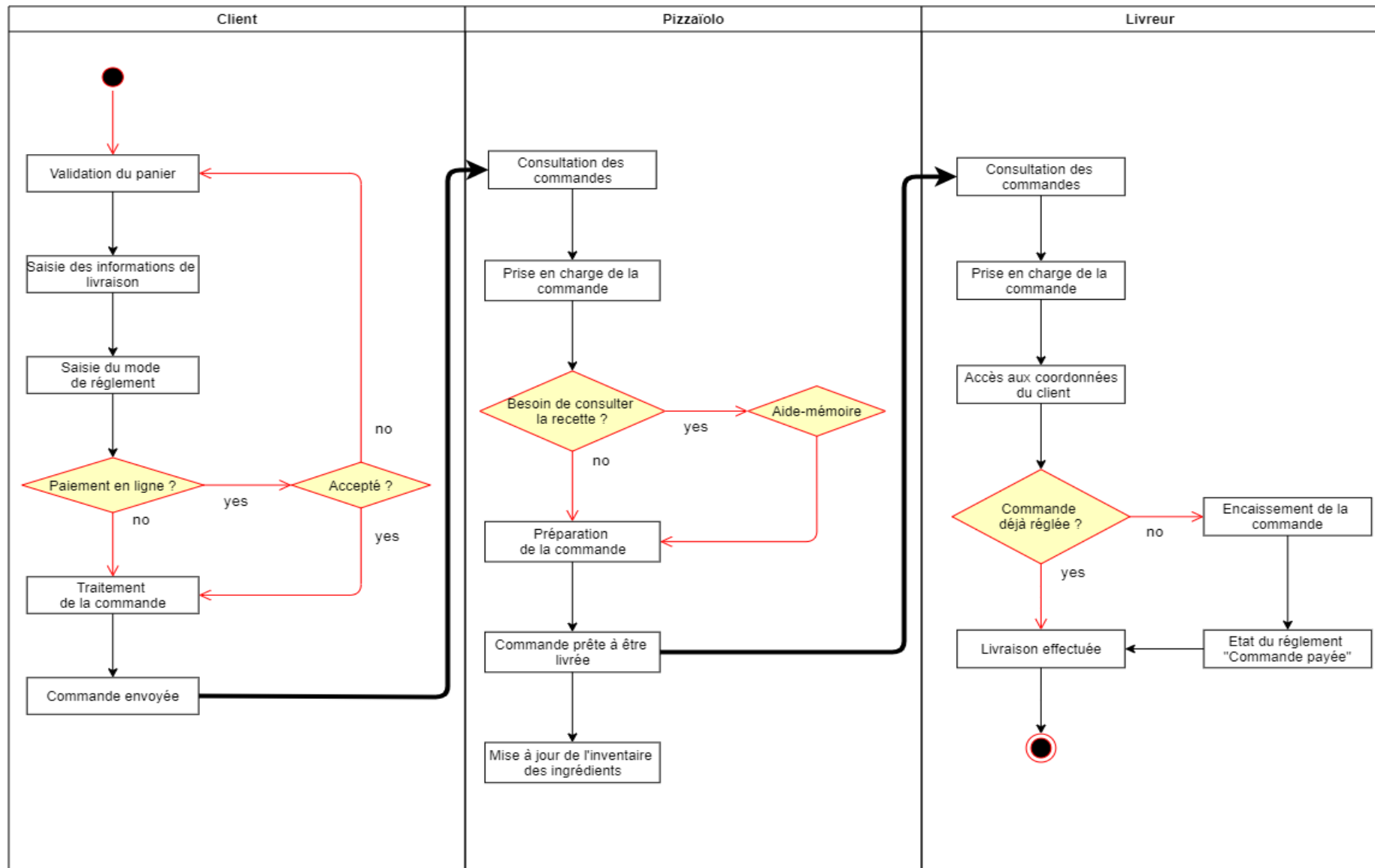
Rappel : Ce diagramme débute lorsque le diagramme de la vue « Pizzaiolo » prend fin. A ce moment, la commande passe à l'état « **En cours de livraison** ».

- Le livreur consulte les commandes
- Le livreur prend en charge la commande
- Il accède aux coordonnées du client
- Il consulte l'état du règlement de la commande :
 - La commande est déjà réglée (Etat du règlement : « **Commande payée** »), le paiement a été effectué en ligne
 - La commande n'est pas encore réglée (Etat du règlement : « **Commande non payée** »), le livreur doit encaisser le règlement de la commande, l'état du règlement passe ensuite à « **Commande payée** »
- La commande est remise au client
- La livraison est effectuée

5.4 - Le workflow « Gestion de commande » (Cycle de vie)



IT Consulting & Development



Le diagramme ci-dessous, nous permet d'observer le cycle de vie d'une commande. Celle-ci débute lorsque le client valide son panier, et jusqu'à la livraison de la commande au client, en passant bien sûr par la préparation de celle-ci.

Voici les étapes importantes de ce cycle :

DEBUT DU CYCLE DE VIE

Le client valide son panier

Le client saisie les informations de livraison et le mode de règlement

Le client confirme la commande, celle-ci est envoyée au système

TRANSITION - CLIENT → PIZZAIOLO

[Le système met à jour l'inventaire des ingrédients]

Le pizaiolo consulte les commandes à préparer

Le pizaiolo prend en charge et prépare la commande

Le pizaiolo indique la commande est prête à être livrée

TRANSITION - PIZZAIOLO → LIVREUR

Le livreur consulte les commandes prêtes pour livraison

Le livreur prend en charge et se dirige vers l'adresse du client

Le livreur livre la commande

FIN DU CYCLE DE VIE

6 - APPLICATION « OC PIZZA »

Le développement de la solution se fera « **From scratch** », le délai imposé (6 mois) étant suffisant pour développer les différentes fonctionnalités et effectuer tous les tests avant la livraison sur les différents points de vente.

Voici les principaux éléments de la solution **OC Pizza** :

Interface Web	HTML5 – JavaScript – CSS / Bootstrap
Authentification	Python « Django » – PostgreSQL
Gestion des commandes	Python « Django » – PostgreSQL
Aide-mémoire	Python « Django » – PostgreSQL
Géolocalisation	API Google Maps

Le développement se fera par l'intermédiaire du Framework **Django** basé sur le langage de programmation **Python**. Ce Framework a été utilisé pour concevoir des sites très célèbres, nous pourrions citer notamment :

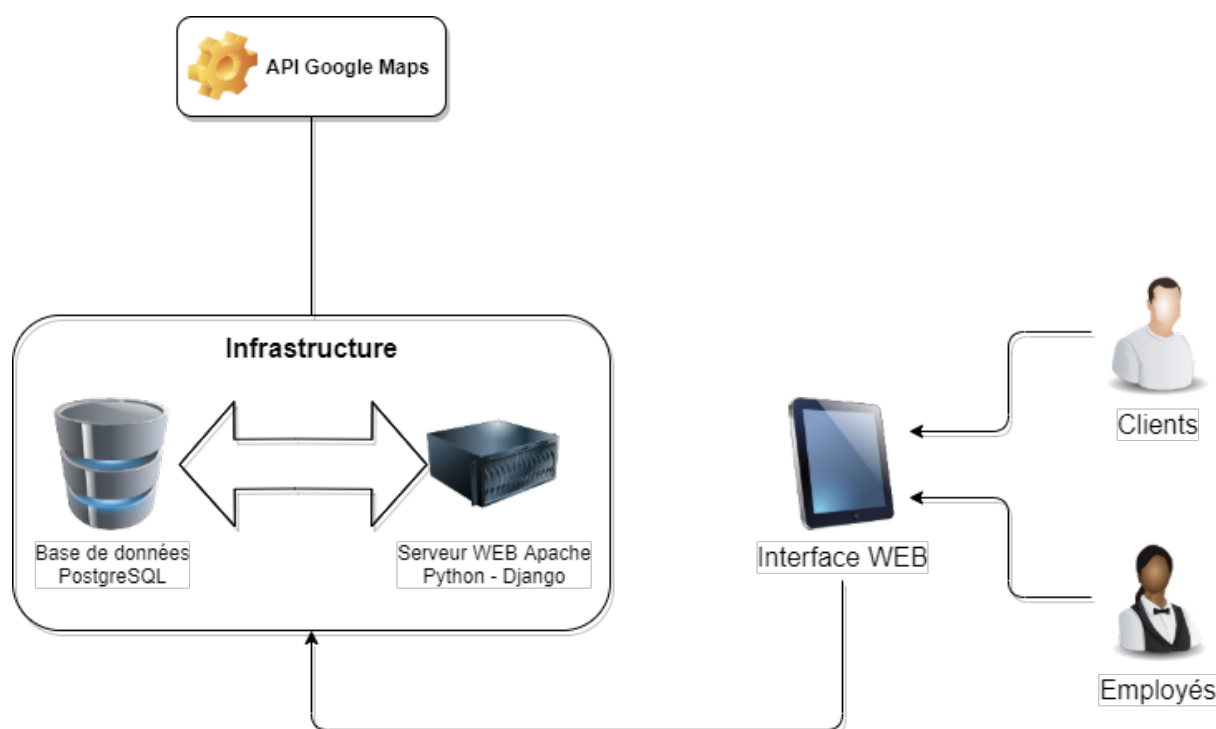
- [Pinterest](#)
- [Instagram](#)
- [The Washington Post](#)
- [National Geographic](#)
- Et bien d'autres...

HTML5 et **CSS** seront également utilisés, notamment pour la partie « **application web** ». Un Template type **Bootstrap** sera utilisé pour la partie graphique du site, le choix de celui-ci restant à définir.

Concernant le **SGBDR** (Système de Gestion de Base de Données Relationnelle), nous utiliserons **PostgreSQL**. Celui-ci présente l'avantage d'être libre mais également d'être reconnu pour sa stabilité.

Le système de Géolocalisation, nécessaire au livreur pour assurer ses livraisons, se fera par l'intermédiaire de l'API Google Maps. Elle permettra d'intégrer facilement cette fonctionnalité sur la solution OC Pizza.

Le diagramme ci-dessous permet d'illustrer la partie matérielle de la solution :



Vue « Serveurs » de la solution OC Pizza

L'interface WEB étant accessible depuis n'importe quel périphérique (PC/Mac, tablette iOS/Android, Smartphone iOS/Android), il est nécessaire que celle-ci soit « **Responsive** ». C'est-à-dire qu'elle pourra s'adapter à n'importe quel périphérique et quel que soit la résolution.

Cela permettra, dans un premier temps, de s'affranchir du développement d'application pour iOS et Android. Il sera toujours possible de le faire, si nécessaire, par la suite.

Il sera tout de même judicieux de proposer une version « **mobile** » du site web, plus adaptée aux petites diagonales sur certains smartphones notamment.

Concernant les employés, le pizaiolo disposera d'une tablette (**iOS** ou **Android**) pour interagir avec l'**application Web**. Le livreur, quant à lui, disposera d'un smartphone (**iOS** ou **Android**) avec abonnement 4G, nécessaire à l'utilisation de l'**application Web** et notamment la Géolocalisation.

L'application sera hébergée sur un serveur (**Linux**) Web **NGINX** où seront installés **Python** et **Django**. Un autre serveur (**Linux**) où **PostgreSQL** sera installé hébergera la base de données. Le serveur Web pourra communiquer avec la base de données lorsque cela sera demandé par l'application.

7 - GLOSSAIRE

A

API

Application Programming Interface. Interface permettant de communiquer avec un autre système..... 6, 7, 10, 24, 35, 36

application WEB

Application manipulable en ligne grâce à un navigateur internet6

B

Bootstrap

Framework CSS utilisé pour l'aspect graphique35

C

CSS

Cascading Style Sheets.....35

D

Django

Framework web basé sur le langage Python35

F

Framework

Ensemble d'outils et de composants logiciels à la base d'un logiciel ou d'une application35

From scratch

Développement de l'application en partant de "zéro"35

G

Google Maps

Service de cartographie créé par Google 10, 35, 36

GPS

Système de localisation par satellite7, 10

H

HTML5

HyperText Markup Language 535

J

JavaScript	
Language de programmation JavaScript	35

P

packages	
Groupement d'éléments en un ensemble cohérent	25
PostgreSQL	
Système de gestion de base de données relationnelle	35, 37
Python	
Language de programmation Python	35, 37

R

Responsive	
Dit d'une application pouvant s'adapter à différentes résolutions d'écran	36

