



# OC PIZZA

**Spécifications techniques**

**Projet 6 : Concevez la solution technique d'un système de gestion de pizzeria**

## Sommaire

1. Contexte.....	3
2. Objectifs.....	3
3. Contraintes.....	3
4. Détail des fonctionnalités.....	4
5. Réalisation de la solution.....	5
6. Déploiement de la solution.....	6
7. Description du domaine fonctionnel.....	7
8. Modèle physique de données.....	9
9. Exemples de requêtes.....	10

## **1. Contexte**

OC Pizza est une jeune entreprise de vente de pizza en livraison ou à emporter. L'entreprise dispose déjà de 5 points de ventes dans la ville et a des perspectives d'évolution devant l'amener à en ouvrir au moins 3 de plus d'ici la fin de l'année. Cependant, la gestion d'une telle activité avec des points de ventes espacés géographiquement et des modes de consommation différents (à emporter ou en livraison) demande beaucoup d'organisation. Aujourd'hui, la société OC Pizza ne dispose pas d'un système d'information répondant à tous ses besoins. Faute de trouver l'outil dont elle rêve sur le marché, elle cherche à faire développer une solution sur mesure qui lui permettra d'avoir une gestion plus sereine de son activité.

## **2. Objectifs**

La solution que nous sommes chargés de développer pour le groupe OC Pizza doit répondre impérativement à certaines attentes qui ont été clairement énoncées lors du premier rendez-vous avec le gérant d'OC Pizza.

L'objectif principal est d'obtenir un outil permettant une gestion efficace des commandes de pizzas à tout niveau. A savoir de la réception de la commande à la livraison de celle-ci, en passant par sa préparation.

- L'outil doit permettre d'améliorer la gestion du stock et son suivi en temps réel. En effet, il doit permettre de savoir à tout moment quels sont les ingrédients restant en stock et quelles sont les pizzas réalisables à partir de ces ingrédients. Et ce pour chacun des points de ventes.

- L'outil doit permettre à tout moment de consulter la liste des commandes passées, en préparation et à venir.

- La solution doit également permettre aux préparateurs des commandes de consulter la recette de la pizza qu'ils sont en train de préparer.

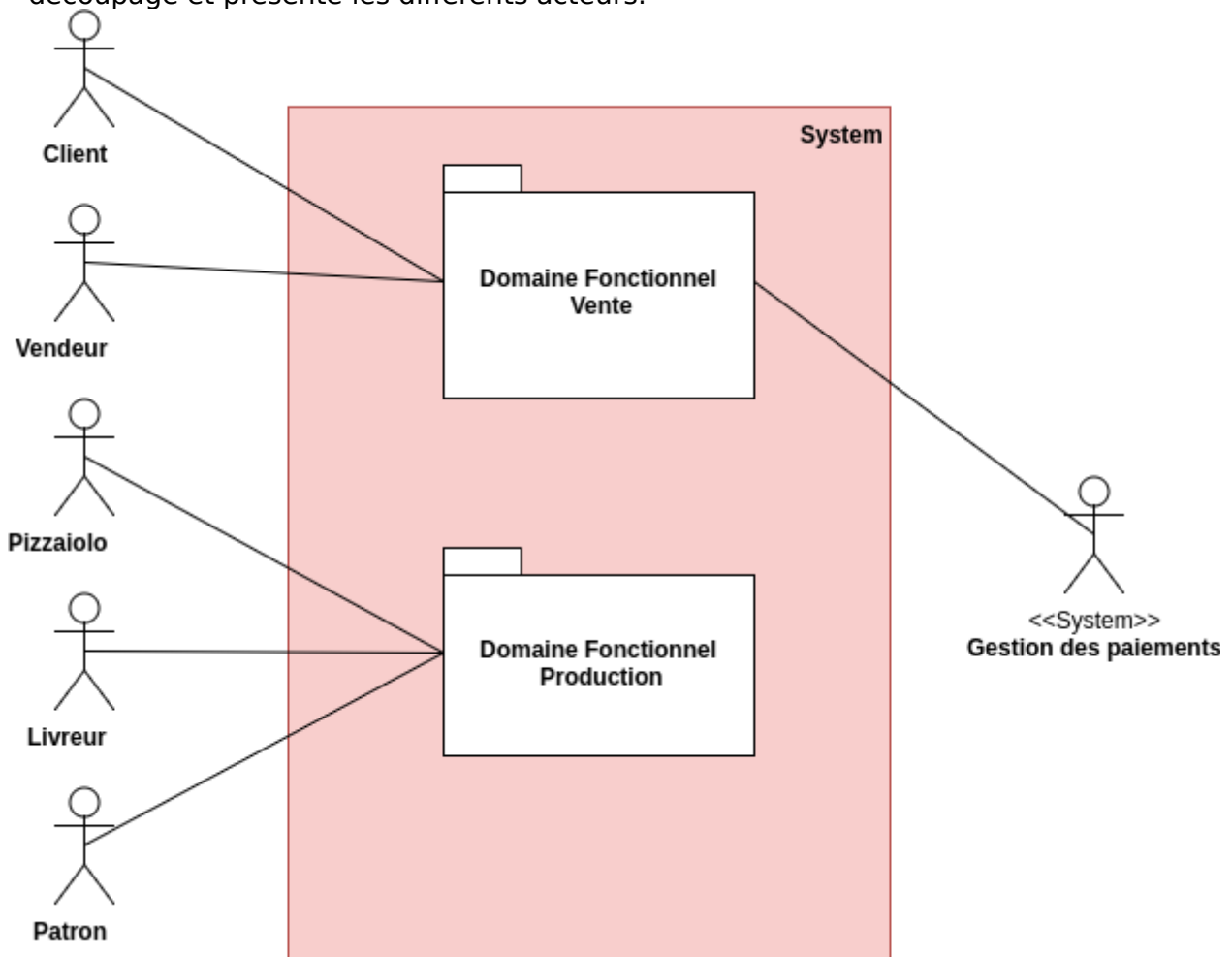
- Enfin, la solution proposée devra permettre aux clients d'« OC Pizza » de passer leur commande via un site internet et de régler directement cette commande en ligne s'ils le souhaitent. Les clients doivent également pouvoir annuler leur commande tant que celle-ci n'est pas en phase de préparation.

## **3. Contraintes**

Le groupe OC Pizza est aujourd'hui composé de cinq établissements et compte s'agrandir. La solution proposée doit donc tenir compte de cette contrainte de multi-établissement que ce soit pour la gestion des commandes et des livraisons que pour le suivi administratif. De plus, cet outil doit tenir compte de la réalité du marché, à savoir les clients. En effet, la conception d'un système de commande à distance et de livraison implique de définir des règles d'attribution des commandes à un lieu de préparation et à un livreur, afin d'optimiser le temps d'attente du client et donc d'améliorer sa satisfaction, tout en ne surchargeant pas un pizzaiolo ou un livreur.

## 4. Détail des fonctionnalités

D'après l'analyse des besoins énoncés précédemment, il en est ressorti de notre réflexion un découpage de la solution en deux packages avec lesquels les différents acteurs utilisant le programme pourront interagir. Le schéma suivant représente ce découpage et présente les différents acteurs.



1. Diagramme de package

On distingue donc 5 acteurs :

4 acteurs internes à la structure OC Pizza : Vendeur, Pizzaiolo, Livreur et Patron

1 acteur externe : Le client.

Le vendeur aura une utilisation du logiciel très proche de celle du client étant donné qu'il fera le lien entre le système et le client dans les cas où ce dernier n'utilise pas le site web pour passer sa commande.

*Le détail de ce découpage a déjà fait l'objet d'un premier document présentant les spécifications fonctionnelles de la solution qui a été présenté au groupe OC PIZZA.*

## **5. Réalisation de la solution**

Lors d'un entretien précédent réalisé avec l'équipe en charge du projet chez OC Pizza, il a été proposé de réaliser la solution à l'aide des technologies suivantes :

L'essentiel de l'application sera programmée via le langage Python3, un langage performant et multiplateforme qui semble adapté à la structure du projet.

L'interface web du logiciel sera réalisée également en Python3 via le Framework Django.

Enfin, toutes les données de l'application seront stockées dans un système de gestion de base de données (SGBD) relationnel dont la structure sera définie par la suite en collaboration avec OC Pizza.

Le choix de PostgreSQL semble opportun pour la réalisation de cette base de données en raison du volume potentiel de celle-ci et de la structure des données qui seront traitées.

Tous ces choix ont été validés au cours de cette réunion et les spécificités techniques présentées dans la suite de ce document prennent donc en compte ces choix.

## **6. Déploiement de la solution**

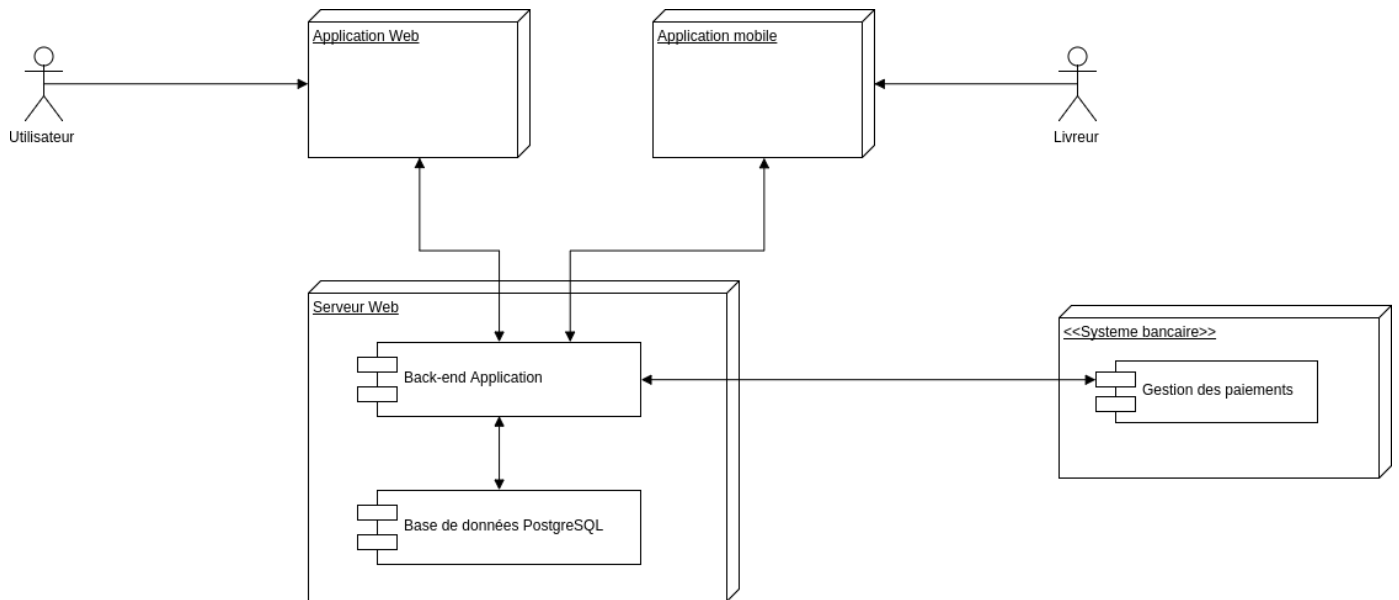
La solution envisagée prend la forme d'une application web à partir de laquelle les utilisateurs pourront effectuer les différentes actions prévues. Que ce soit la commande par le client, la consultation de la recette par le préparateur de la commande ou encore la consultation des factures par le dirigeant. Il est cependant envisagé de développer en complément une application mobile, destinée principalement aux livreurs pour le permettre d'avoir accès à la liste des commandes à livrer ou à la validation d'une livraison. En effet, les livreurs étant itinérants, ce type d'interface leur facilitera la tâche.

Nous nous proposons de déployer l'application sur un Cloud en mode PAAS (Platform as a Service). La structure du projet ne présente pas de nombreuses spécificités justifiant d'un paramétrage personnalisé d'un serveur et le recours à ce type de Cloud permettra d'éviter les coûts qui seraient liés à la gestion d'un serveur de A à Z. Le choix de la plateforme sur laquelle déployer l'application relèvera de la direction d'OC PIZZA après discussion entre elle et nous.

La base de données utilisée pour le fonctionnement de l'application utilisera le SGBD PostgreSQL qui a été choisi lors d'une précédente réunion.

Enfin, l'application aura recours à un composant externe, un prestataire de services de paiement (PSP), permettant de gérer les paiements de commande réalisés en ligne. Là encore il appartiendra à la direction d'OC PIZZA de choisir ce prestataire, il pourra s'agir de la solution proposée par la banque d'OC PIZZA ou d'un système proposé par un autre acteur indépendant.

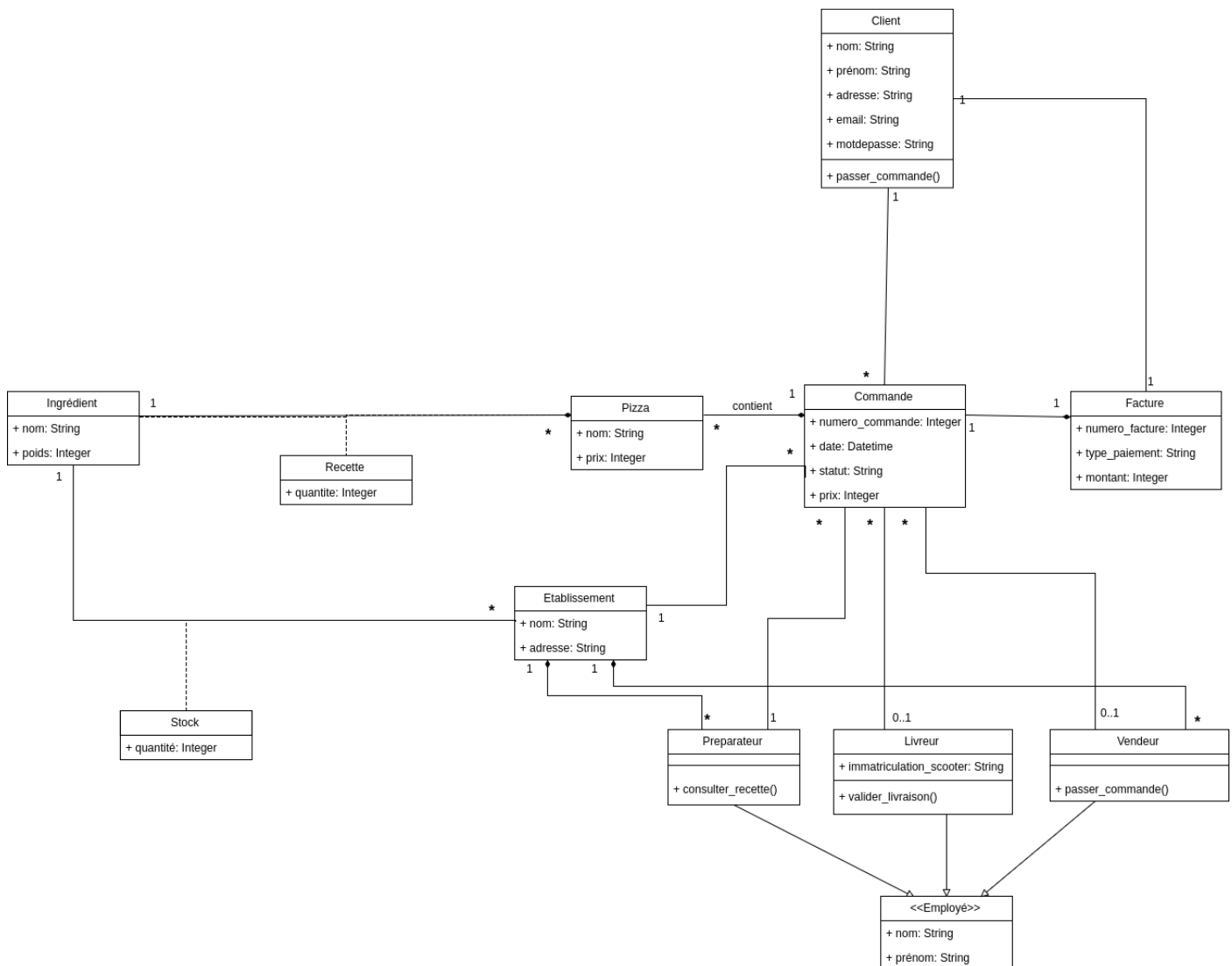
Le diagramme suivant récapitule les informations ci-dessus énoncée, et présente l'agencement des différents composants entre eux :



## 2. Diagramme de déploiement

## 7. Description du domaine fonctionnel

La mise en place de la solution telle que présenté précédemment ainsi que dans le document détaillant les spécifications fonctionnelles, nécessite de déterminer les « Objets » (donc les données) qui seront manipulés par le programmes. Après avoir étudié les différents aspects du projet, il en est ressorti le diagramme de classe suivant :



3. Diagramme de classes

### Explication des choix utilisés :

- **Pizza** : Chaque pizza sera matérialisée par un nom et un prix.

- **Ingrédient** : La classe ingrédient comprendra comme informations le nom de l'ingrédient ainsi que le poids d'une « portion ». Cette notion de portion sera par la suite utilisée dans la classe « Recette ». Pour chaque pizza, la recette sera un nombre de portion de chaque ingrédient. Le stock sera géré également en nombre de portions.

- *Recette* : Cette classe aura pour but de contenir la liste des ingrédients nécessaires pour chaque type de pizza et également d'indiquer le nombre de portions à utiliser pour chaque ingrédient.

- *Stock* : Le Stock sera géré par établissement. La classe stock permettra donc d'obtenir pour chaque ingrédient, sa quantité pour un établissement donné. Cette gestion permettra à la fois de gérer les pizzas disponibles pour une commande dans un établissement donné, mais également par la suite d'organiser éventuellement des prêts d'ingrédients entre établissements.

- *Client* : La classe client comprendra toutes les informations nécessaires à l'identification du client ainsi qu'à la livraison de sa commande. Elle comprendra également les méthodes nécessaires au passage d'une commande par le client sur l'application.

- *Commande* : La classe commande est la classe autour de laquelle toute l'application est tournée. Elle contient les informations nécessaires pour la gestion des commandes et de ce fait, elle fait le lien avec la quasi-totalité des autres classes.

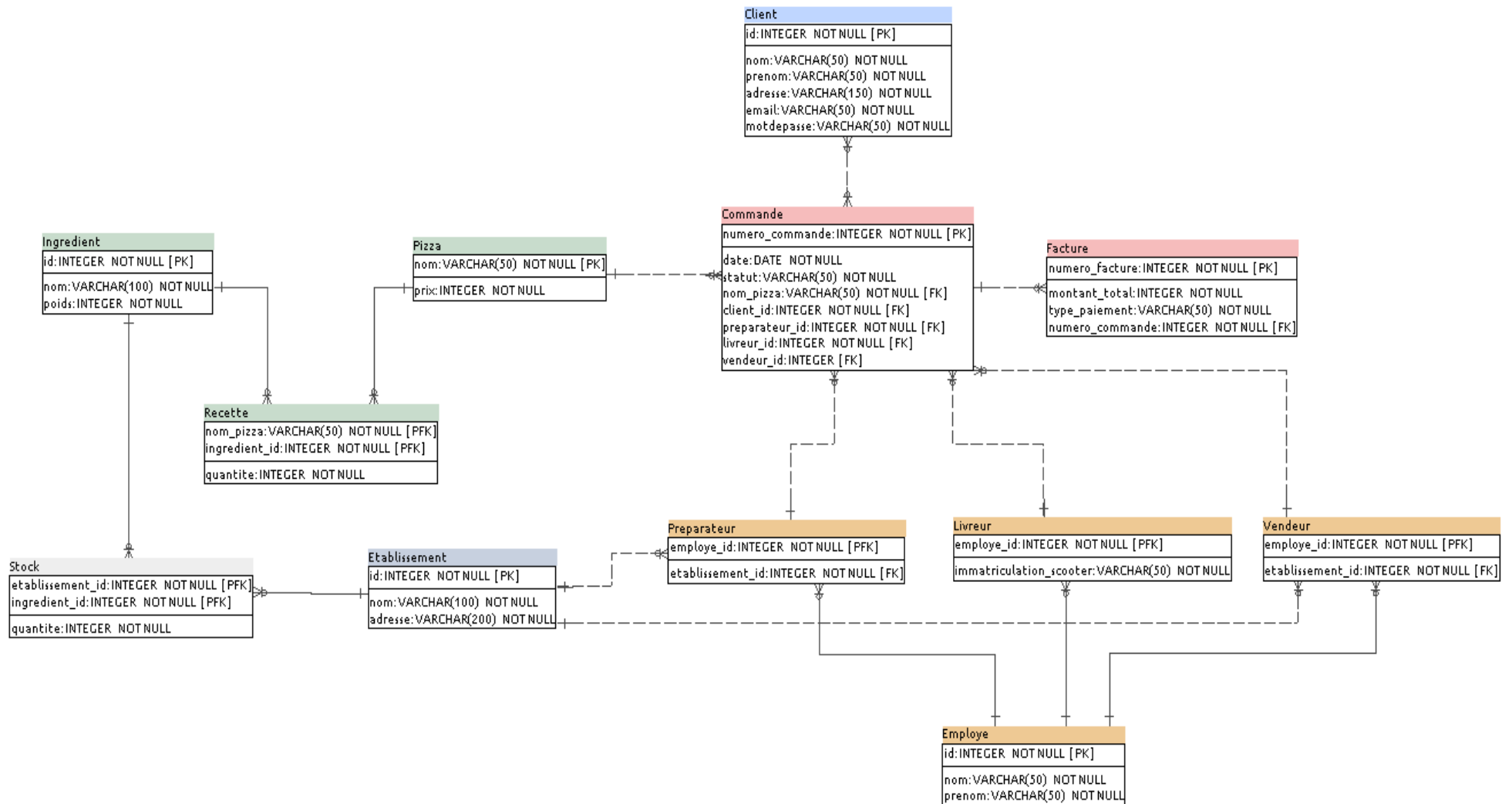
- *Facture* : La facture reprend les éléments essentiels de la commande et y ajoute des informations qui serviront la gestion administrative des ventes.

- *Employé* : Tous les types d'employés (préparateur, livreur, vendeur) découleront d'une même classe abstraite « Employé ». Une classe héritant de celle-ci sera ensuite créée pour chaque type d'employé permettant l'utilisation d'informations ou de méthode particulières.

## **8. Modèle physique de données**

Le schéma présenté ci-dessous représente le modèle physique de données, il permet la traduction du diagramme de classe présenté précédemment au sein du SGBD.





#### 4. Modèle Physique de Données

## 9. Exemples de requêtes

Afin d'illustrer le fonctionnement de la base de données qui serait manipulée par l'application, il a été remis à OC PIZZA, en plus de ce document, des scripts permettant la création de la base de donnée telle que décrite par le modèle physique de données.

Sont présentées ci-dessous 4 requêtes pouvant être réalisées sur la base de données de démonstration ainsi que leur résultat.

### Requête 1 : Affichage d'un client

Requête :

```
-- D'affichage des informations du client 'Jimbo KERN'
SELECT *
FROM Client
WHERE nom='KERN';
```

Résultat :

```
+-----+-----+-----+-----+-----+-----+
| id | nom | prenom | adresse | email | motdepasse |
+-----+-----+-----+-----+-----+-----+
| 2 | KERN | Jimbo | 112 Main street, South Park, Colorado | jimbokern@southpark.com | comingrightforus |
+-----+-----+-----+-----+-----+-----+
1 row in set (0,00 sec)
```

### Requête 2 : Affichage du stock d'un ingrédient dans un établissement donné

Requête :

```
-- Selection du stock d'olives disponibles dans l'établissement 'GO COWS'
SELECT Ingredient.nom AS Ingredient, Stock.quantite AS Stock
FROM Stock
INNER JOIN Etablissement ON Etablissement.id=Stock.etalblissement_id
INNER JOIN Ingredient ON Ingredient.id=Stock.ingredient_id
WHERE Etablissement.nom='GO COWS';
```

Résultat :

```
+-----+-----+
| Ingredient | Stock |
+-----+-----+
| tomate | 97 |
| mozzarella | 98 |
| jambon | 98 |
| olives | 93 |
| oeuf | 100 |
+-----+-----+
5 rows in set (0,01 sec)
```

### Requête 3 : Affichage d'une recette

Requête :

```
--Selection du poids des ingrédients composant une pizza REINE
SELECT Ingredient.nom as Ingredient, Recette.quantite*Ingredient.poids AS Poids
FROM Recette
INNER JOIN Pizza ON Pizza.nom=Recette.nom_pizza
INNER JOIN Ingredient ON Ingredient.id=Recette.ingredient_id
WHERE nom_pizza = 'REINE';
```

Résultat :

```
+-----+-----+
| Ingredient | Poids |
+-----+-----+
| tomate    | 150   |
| mozzarella| 200   |
| jambon    | 100   |
| olives    | 35    |
| oeuf      | 30    |
+-----+-----+
5 rows in set (0,01 sec)
```

### Requête 4 : Affichage d'une commande

Requête :

```
-- Selection des informations d'une commande
SELECT date, statut, nom_pizza, CONCAT_WS(' ', Client.nom, Client.prenom) AS client,
        CONCAT_WS(' ',Preparateur.nom, Preparateur.prenom) AS preparateur,
        CONCAT_WS(' ', Livreur.nom, Livreur.prenom) AS livreur,
        CONCAT_WS(' ', Vendeur.nom, Vendeur.prenom) AS vendeur
FROM Commande
INNER JOIN Employe AS Preparateur ON Preparateur.id=Commande.preparateur_id
INNER JOIN Employe AS Livreur ON Livreur.id=Commande.livreur_id
INNER JOIN Employe AS Vendeur ON Vendeur.id=Commande.vendeur_id
INNER JOIN Client ON Client.id=Commande.client_id
WHERE Client.nom='MARSH';
```

Résultat :

```
+-----+-----+-----+-----+-----+-----+-----+
| date      | statut | nom_pizza | client      | preparateur | livreur    | vendeur    |
+-----+-----+-----+-----+-----+-----+-----+
| 2018-01-03 | LIVREE | REINE     | MARSH Randy | BROFLOVSKI Kyle | CARTMAN Eric | McCORMICK Kenny |
+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0,00 sec)
```