

# Conception d'un système de gestion de pizzeria

Dossier de conception technique

Version 2.0

**Auteur**

Céline PELLETIER

## Table des matières

1. Versions.....	3
2. Introduction.....	4
2.1. Objet du document.....	4
2.2. Besoins du client.....	4
3. Le domaine fonctionnel.....	5
3.1. Référentiel.....	5
3.2. Règles de gestion.....	5
4. Architecture Technique.....	6
4.1. Diagramme de composant 1 : commande en ligne.....	6
4.2. Diagramme de composant 2 : commande sur place.....	7
4.3. Diagramme de composant 3 : livraisons.....	7
4.4. Diagramme de composant 4 : visuel cuisine.....	8
4.5. Diagramme de composant 5 : administrateur Accueil.....	8
4.6. Diagramme de composant 6 : base de données.....	9
4.7. Model Physique de Données.....	10
5. Architecture de Déploiement.....	11
5.1. Serveur de Base de données.....	12
5.2. Serveur Web NGINX.....	12
5.3. Serveur d'application GUNICORN.....	12
6. Glossaire.....	13

## 1. Versions

Auteur	Date	Description	Version
Céline PELLETIER	30/09/2020	Création du document	1.0
Céline PELLETIER	04/10/2020	Ajout des diagrammes	2.0

## **2. Introduction**

### **2.1. Objet du document**

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

L'objectif de ce document est de décrire la solution technique que nous allons utiliser pour mener à bien ce projet.

Les éléments du présent dossier découlent :

- de plusieurs entretiens avec le client.
- de plusieurs réunions avec l'équipe en charge du projet.
- du document de spécifications fonctionnelles générales.

### **2.2. Besoins du client**

Le client a évoqué le besoin d'avoir un système informatique lui permettant une meilleure gestion de ses commandes, de leurs réceptions à leurs livraisons.

Pour cela il a besoin d'un suivi en temps réel des commandes passées et en préparations. Il a aussi besoin d'un système de gestion des stocks d'ingrédients en temps réel, afin de savoir quelles pizzas sont encore réalisables.

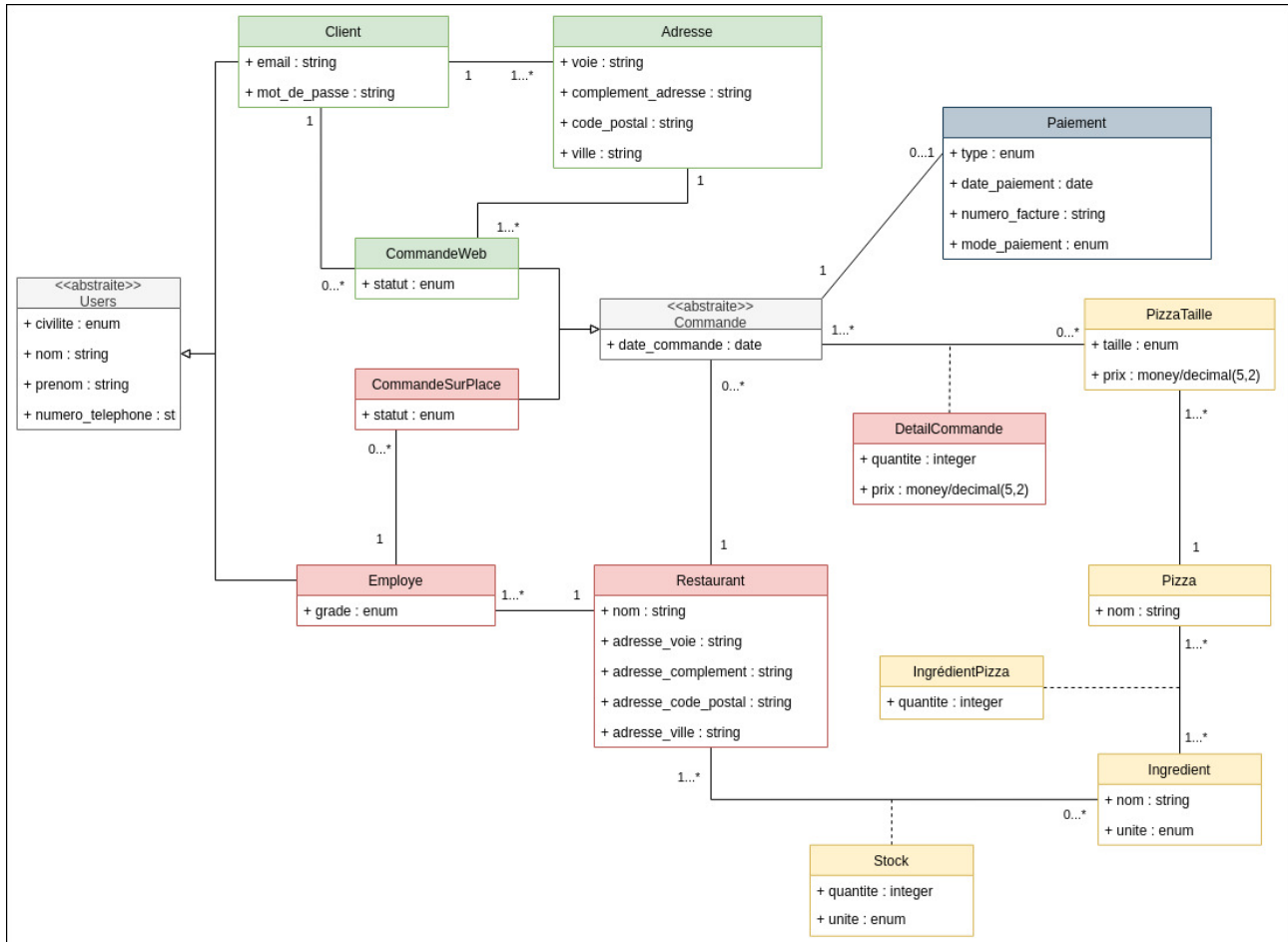
Il a aussi évoqué le besoin de proposer aux clients la possibilité de commander via internet, en plus des commandes sur place, à l'aide d'un site internet. Ce site permettra aux clients de pouvoir commander, suivre l'état de leurs commandes, choisir entre payer en ligne ou à la livraison et modifier ou annuler leurs commandes tant que celles-ci n'ont pas été préparées.

Et enfin il aimerait pouvoir proposer à ses cuisiniers un aide mémoire pour chacune des pizzas.

## 3. Le domaine fonctionnel

### 3.1. Référentiel

Ci- dessous vous pouvez voir le diagramme de classes effectué afin de répondre aux besoins du client.



### 3.2. Règles de gestion

Le diagramme de classe, ci-dessus, nous indique que les classes « Client » et « Employe » hériteront toutes les deux de la classe « Users ».

Un client aura la possibilité d'avoir une ou plusieurs adresses et d'avoir zéro ou plusieurs commandes en ligne.

Les classes « CommandeWeb » et « CommandeSurPlace » hériteront toutes deux de la classe **Commande**. Elles auront aussi toutes les deux un statut mais qui sera propre à chacune d'elles.

Chaque commande sera liée à un restaurant et à un paiement. Elles auront aussi un détail de la commande contenant les pizzas demandées par le client.

Pour les commandes en ligne, celles-ci seront liées à un client ainsi qu'à une adresse.

Et pour les commandes sur place, celles-ci seront liées à un employé du restaurant dans lequel elles ont été prises.

Chaque employé sera lié à un seul restaurant.

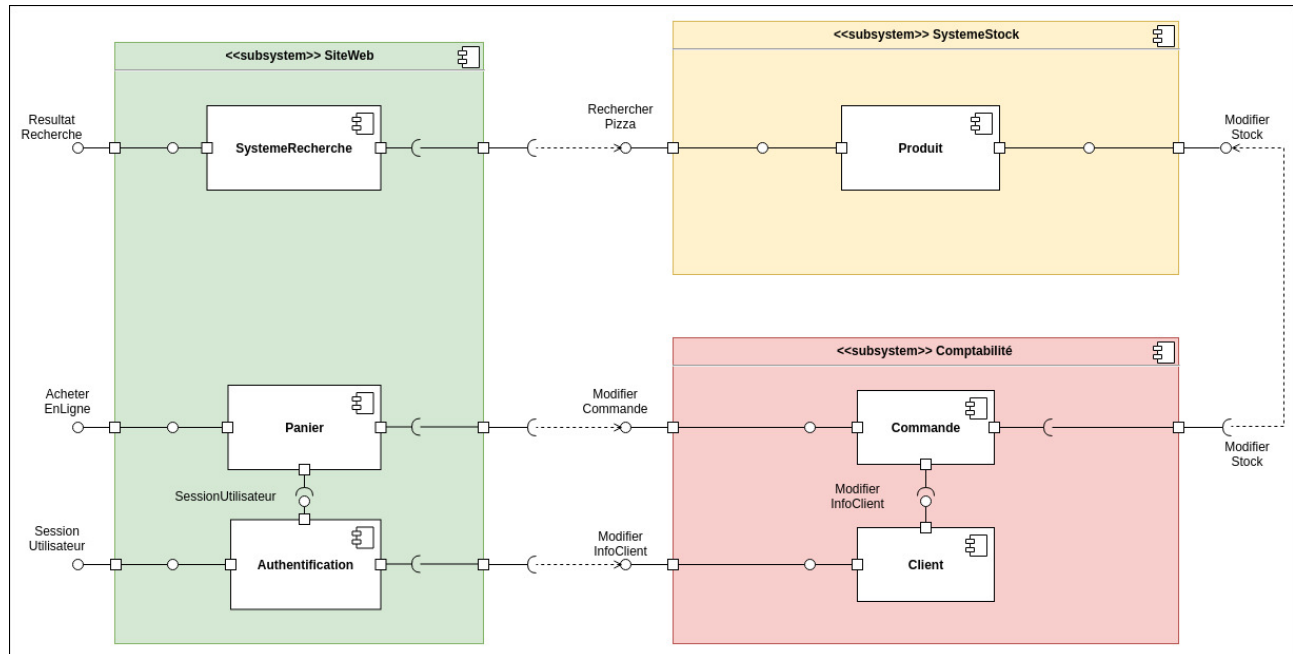
Et chaque restaurant possédera un stock contenant les ingrédients utiles pour la réalisation des pizzas.

## 4. Architecture Technique

Afin de Réaliser ce projet nous allons utiliser différents outils :

- Le langage Python 3 pour la réalisation de l'application.
- Le framework Django pour la réalisation du site web.
- Le langage MySQL pour la réalisation de la base de données.

### 4.1. Diagramme de composant 1 : commande en ligne



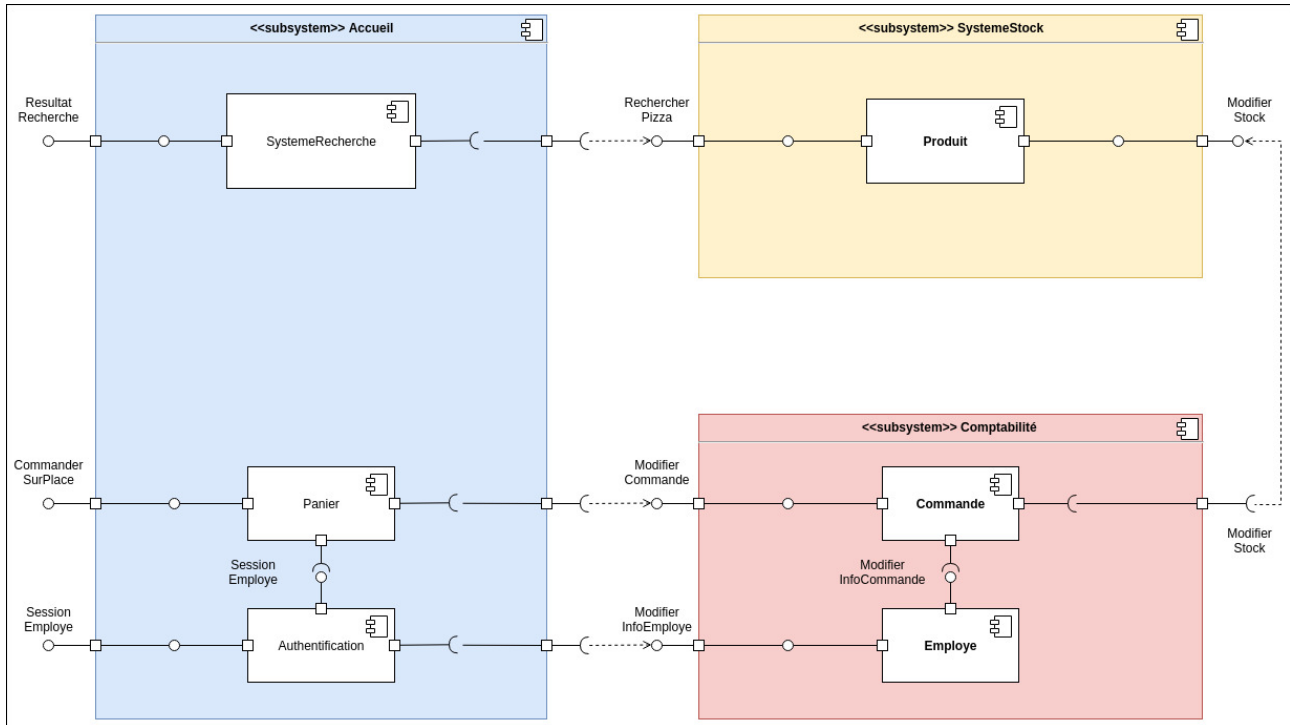
Le diagramme de composant, ci-dessus, nous montre les interactions entre les composants du site Web, du système de Stock et du système de Comptabilité.

Nous avons tout d'abord le site Web, qui permet au client d'acheter en ligne, d'accéder à leurs sessions et de faire une recherche de produit. Celui-ci est composé d'un composant « Authentification », permettant au client de s'authentifier et de gérer ses informations. Il possède aussi un composant « Panier », permettant au client de gérer les produits qu'il souhaite acheter.

Ensuite nous avons le système de Comptabilité qui contient les clients et leur commandes. Le composant « Commande » propose, au composant du site Web « Panier », une interface pouvant modifier le panier. Le composant « Client », lui, propose au composant du site Web « Authentification », une interface pouvant modifier les informations du compte du client.

Et pour finir nous avons le système Stock contenant le composant Produit qui propose deux interfaces, une pour gérer les stocks de la pizzeria et l'autre permettant de rechercher des produits que la pizzeria propose.

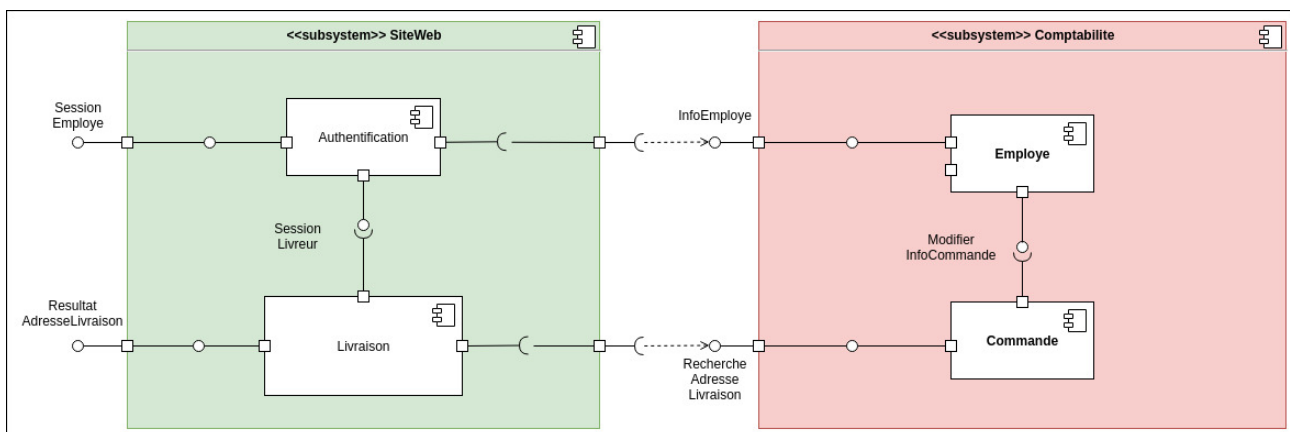
## 4.2. Diagramme de composant 2 : commande sur place



Le diagramme de composant, ci-dessus, nous montre les interactions entre les composants du système de l'Accueil, du système de Stock et du système de Comptabilité.

Le système de l'Accueil propose les mêmes interactions que le site Web mais cette fois-ci pour les employés de la pizzeria. Il permet donc à un employé d'accéder à sa session, d'entrer de nouvelles commandes sur place et de faire une recherche des produits proposés par la pizzeria.

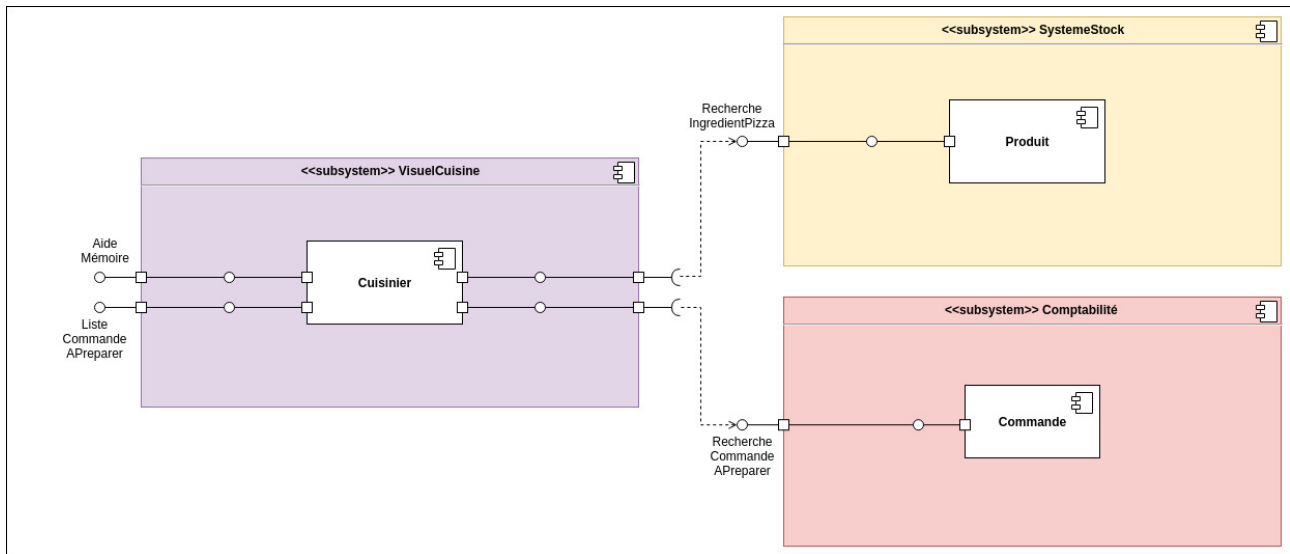
## 4.3. Diagramme de composant 3 : livraisons



Le diagramme de composant, ci-dessus, nous montre les interactions entre les composants du site Web et ceux du système de Comptabilité mais cette fois ci pour les livraisons.

Nous avons donc dans le site Web, un composant « Livraison » qui requiert l'interface permettant de récupérer l'adresse de livraison de la commande, du composant « Commande » du système de comptabilité. Ce composant est lié au composant « Authentification », afin que seul les livreurs puissent accéder aux adresses de livraisons.

## 4.4. Diagramme de composant 4 : visuel cuisine

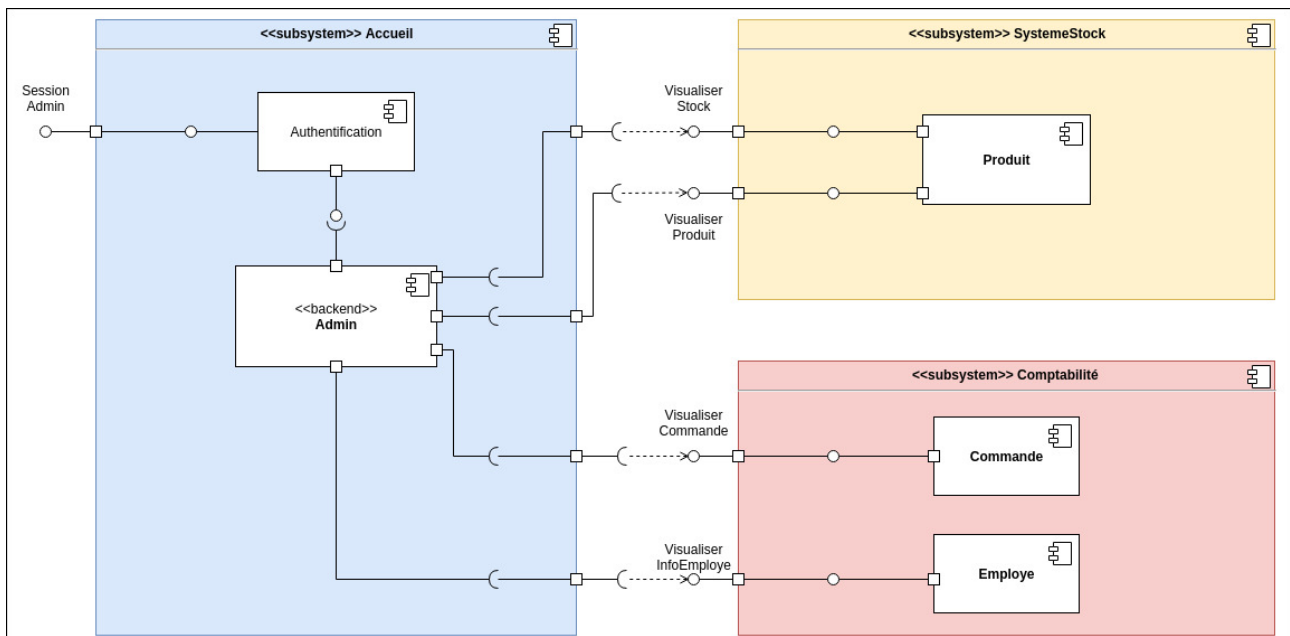


Le diagramme de composant, ci-dessus, nous montre l'interaction entre les composants du système Visuel de la Cuisine et des systèmes de Stock et de Comptabilité.

Le composant « Cuisinier » requiert l'interface permettant de rechercher les ingrédients nécessaires à la préparation des pizzas, du composant « Produit », afin de pouvoir retourner un aide mémoire.

Il a aussi besoin de l'interface permettant de rechercher les commandes à préparer, du composant « Commande », afin de pouvoir retourner une liste des commandes en attentes de préparation.

## 4.5. Diagramme de composant 5 : administrateur Accueil

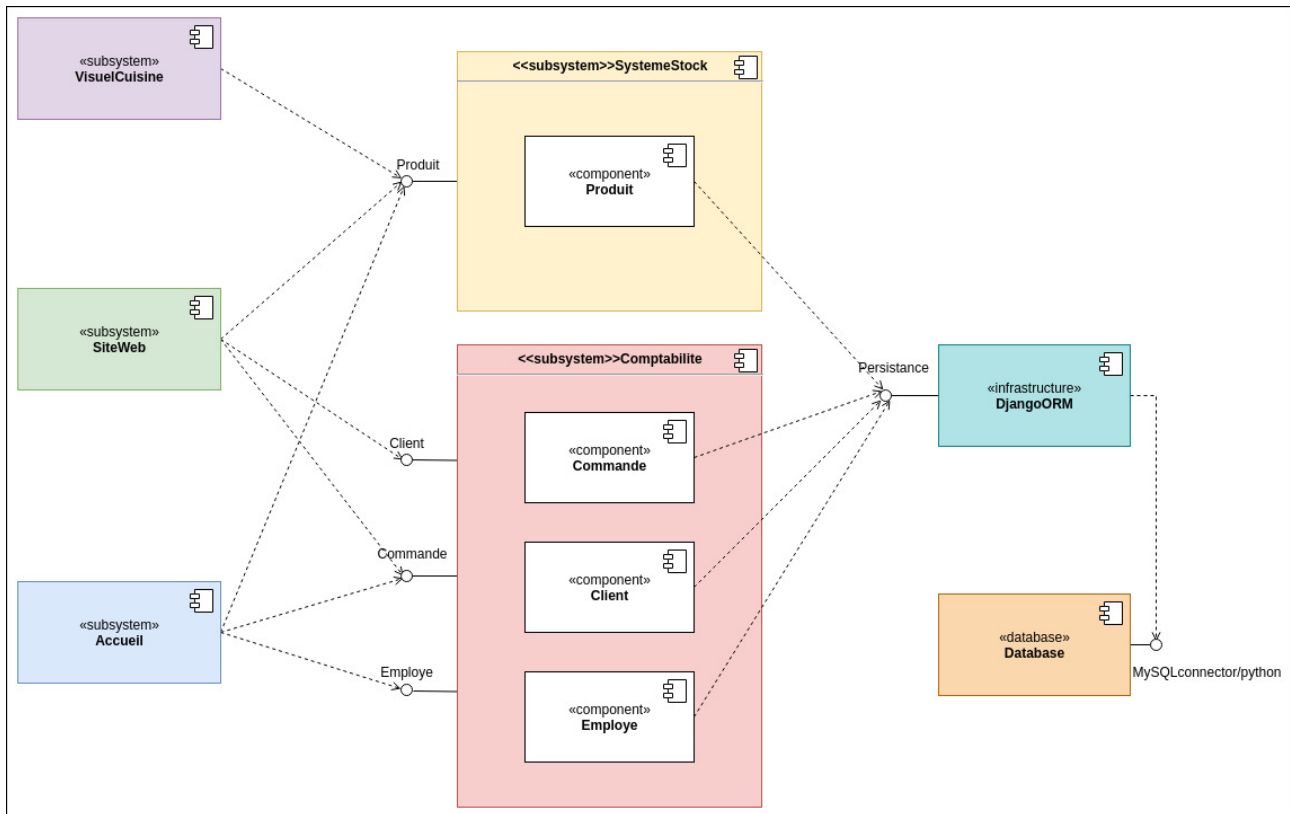


Le diagramme de composant, ci-dessus, nous montre les différentes interactions entre le composant « admin » du système de l'Accueil et les composants des systèmes de Comptabilité et de Stock.

Le composant Admin requiert les interfaces permettant de visualiser les différents composants de ces deux systèmes, afin de pouvoir visualiser les informations des employés, les commandes, les produits et les stocks des différentes pizzerias.



## 4.6. Diagramme de composant 6 : base de données



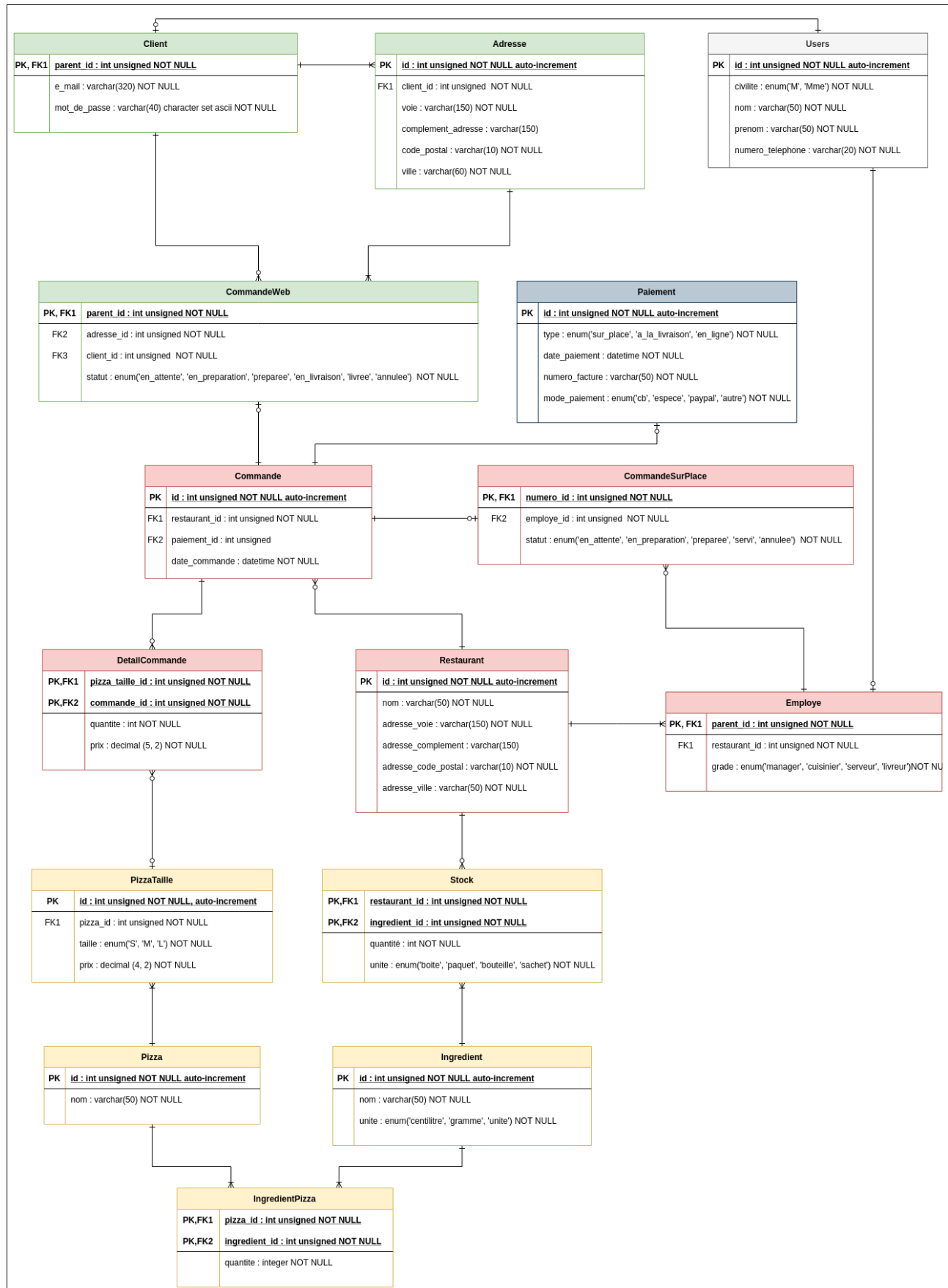
Le diagramme de composant, ci-dessus, nous montres quels composants sont en relation avec la base de données.

Nous pouvons donc voir que les composants « Produit », « Commande », « Client » et « Employe » interagissent tous avec la base de données.

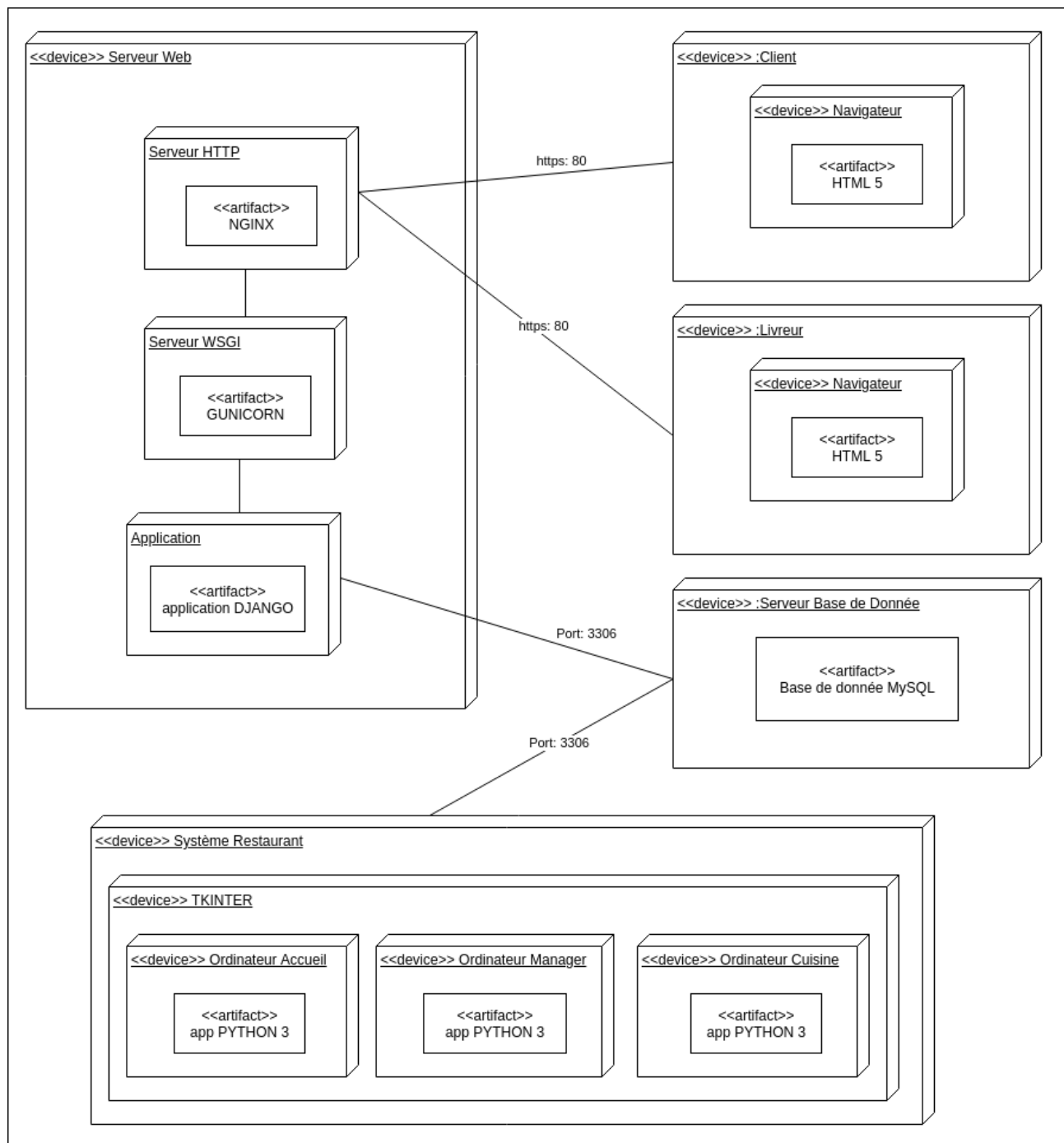
## 4.7. Model Physique de Données

Ci-dessous vous pouvez voir le Model Physique de Données .

Celui-ci nous montre les différentes tables de la base de données et les relations entre chacune d'elles.



## 5. Architecture de Déploiement



Ci-dessus, vous pouvez voir le diagramme de déploiement du système.

Il est composé d'un serveur Web, contenant un serveur HTTP, un serveur WSGI et de l'application du site web. Cette application est reliée au serveur de la base de données.

Le système Restaurant, lui, est une application qui est installée sur chacun des ordinateurs du restaurant et qui est reliée au serveur de la base de données.

Le client et le livreur eux accèdent au site via un navigateur.

## **5.1. Serveur de Base de données**

La base de données va nous permettre de stocker, de manipuler et de rechercher les données que nous auront stockées dans celle-ci.

Nous utilisons une base de données relationnelle, car ce genre de base de données va nous permettre de lier les données à d'autres informations au sein de la base de données. Celle-ci est composée de plusieurs tableaux en relations les uns avec les autres.

Pour cela nous allons utiliser le système de gestion de base de données relationnelle (SGBDR) MySQL. Ce système fait partie des logiciels de gestion de base de données les plus utilisés au monde et il est notamment très utilisé par les sites Web.

## **5.2. Serveur Web NGINX**

NGINX est un logiciel libre de serveur Web, qui a été développé au départ pour les besoins d'un site russe à très fort trafic. Il est l'un des serveurs Web les plus utilisés au monde. Il est conçu pour offrir une faible consommation de la mémoire.

Il sert à récupérer des requêtes HTTP qui seront gérées par l'application Django puis retournées. La réponse retournée par l'application Django est renvoyée à NGINX pour qu'il envoie la réponse au navigateur.

## **5.3. Serveur d'application GUNICORN**

GUNICORN (Green Unicorn) est un serveur HTTP Python qui utilise les spécifications WSGI. Il est compatible avec un grand nombre de framework dont Django et il est particulièrement puissant.

Il sert à faire la communication entre le serveur HTTP NGINX et l'application Django.

## 6. Glossaire

Therme	Définition
Diagramme de classes	Schéma utilisé pour présenter les classes intervenant dans le système. C'est une représentation statique des éléments qui composent un système et de leurs relations.
Classe	Concept abstrait permettant de représenter toutes les entités d'un système.
Héritage	Mécanisme permettant, lors de la déclaration d'une nouvelle classe, d'y inclure les caractéristiques d'une autre classe.
Framework	Ou « Cadre de travail », est un ensemble d'outils et de composants logiciels. Il établit les fondations d'un logiciel ou son squelette applicatif. Il s'utilise de la même manière qu'un patron en couture.
Base de données	Ensemble d'informations structurées accessible au moyen d'un logiciel.
Diagramme de composant	Décrit l'organisation du système du point de vue des éléments logiciels, comme les modules ou encore les éléments de configuration. Il permet de mettre en évidence les dépendances entre les composants.
Model physique de données	Schéma permettant de construire la structure finale de la base de données, avec les différents liens entre les éléments qui la composent.
Diagramme de déploiement	Vue statique qui sert à représenter l'utilisation de l'infrastructure physique par le système et la manière dont les composants du système sont répartis, ainsi que leurs relations entre eux.
WSGI (Web Server Gateway Interface)	Spécification qui définit une interface entre des serveurs et des applications web pour le langage Python.
Système de gestion de base de données relationnelles (SGBDR)	Système stockant et gérant l'accès et la recherche d'informations dans des bases de données.
Serveur Web	Est, soit un logiciel de service de ressources web (serveur HTTP), soit un serveur informatique (ordinateur) qui répond à des requêtes du World Wide Web sur un réseau public (Internet) ou privé (intranet), en utilisant principalement le protocole HTTP.
Requête HTTP	Demande effectuée par le navigateur web (Firefox, Google Chrome, ...) au serveur HTTP lorsqu'il souhaite télécharger une page Web.
Serveur d'applications	Logiciel permettant à des utilisateurs, connectés en réseau, d'accéder à tout ou une partie d'un logiciel applicatif(programme, interface graphique, ...), à partir d'un exemplaire unique situé sur une machine informatique.