



**Société : OC Pizza**

-

**Système de gestion informatique**

**Spécifications techniques**



## Table des matières

I. Cadrage du projet.....	3
I.1 Contexte de l'entreprise.....	3
I.2 Objectifs.....	3
I.3 Enjeux.....	3
I.4 Objet.....	3
II. Architecture technique.....	4
II.1 Généralités.....	4
II.2 Pile logicielle.....	4
II.2.1 Serveur.....	4
II.2.2 Client.....	4
II.2.3 Mobile.....	4
II.2.4 Base de données.....	4
II.3 Diagramme de composants.....	5
III. Architecture de l'application.....	7
III.1 Description du domaine fonctionnel.....	7
III.1.1 Diagramme de classes.....	7
III.1.2 Interactions.....	7
V. Architecture de la base de données.....	8
VI. Architecture de déploiement.....	9
VII. Points particuliers.....	10
VII.1 Environnement de développement.....	10
VII.2 Monitoring.....	10
VII.3 Gestion des logs.....	10



# I. Cadrage du projet

## I.1 Contexte de l'entreprise

La société OC Pizza est une jeune chaîne de pizzeria spécialisée dans la livraison ou la vente à emporter.

Les deux créateurs du groupe, Franck et Lola, souhaitent ouvrir 3 points de vente supplémentaires d'ici 6 mois, portant le nombre total de points de vente à 8.

Pour mener à bien ce développement, il est nécessaire au groupe de centraliser les données de ses points de vente, par l'intermédiaire d'un système informatique robuste. Cela permettra aux deux patrons d'avoir une vision globale du fonctionnement de chaque pizzeria, afin d'améliorer la gestion de la société.

De plus, ce système permettra d'améliorer l'efficacité des salariés en ayant un suivi en temps réel des livraisons.

## I.2 Objectifs

Fournir à OC Pizza une solution clé en main, permettant aux clients d'effectuer leurs commandes en ligne, tout en centralisant les données du groupe dans un système de gestion d'entreprise.

## I.3 Enjeux

- Rester compétitif face à la concurrence dense du milieu dans lequel évolue OC Pizza
- Augmenter la zone de chalandise de la société en proposant un site de vente en ligne
- Augmenter l'efficacité interne en améliorant la réactivité et la gestion des commandes
- Consolider les données du groupe pour avoir un meilleur pilotage d'entreprise

## I.4 Objet

Le présent document est le dossier de conception technique de l'application OC Pizza. Il a pour objectif de modéliser les objets du domaine fonctionnel, d'identifier les éléments qui composent le système ainsi que leurs interactions, puis leur déploiement, et enfin de présenter le schéma de la base de données.



## II. Architecture technique

### II.1 Généralités

L'application OC Pizza sera composée de 3 interfaces différentes :

- Une interface front, qui permettra aux clients de consulter et commander leurs pizzas
- Une interface back, qui permettra aux employés d'interagir avec le système
- Une interface mobile, qui permettra aux livreurs de gérer leurs livraisons à effectuer

La base de données sera commune pour toute l'application OC Pizza.

L'application sera hébergée sur un serveur DigitalOcean, pour plus de détails, se référer à la partie VI.

### II.2 Pile logicielle

#### II.2.1 Serveur

- Langage : Python 3.6.x
- Framework et ORM : Django 2.2.x

#### II.2.2 Client

- Langages : HTML5, CSS3, JavaScript
- Frameworks : Bootstrap, JQuery

#### II.2.3 Mobile

- Langages : Swift, Java

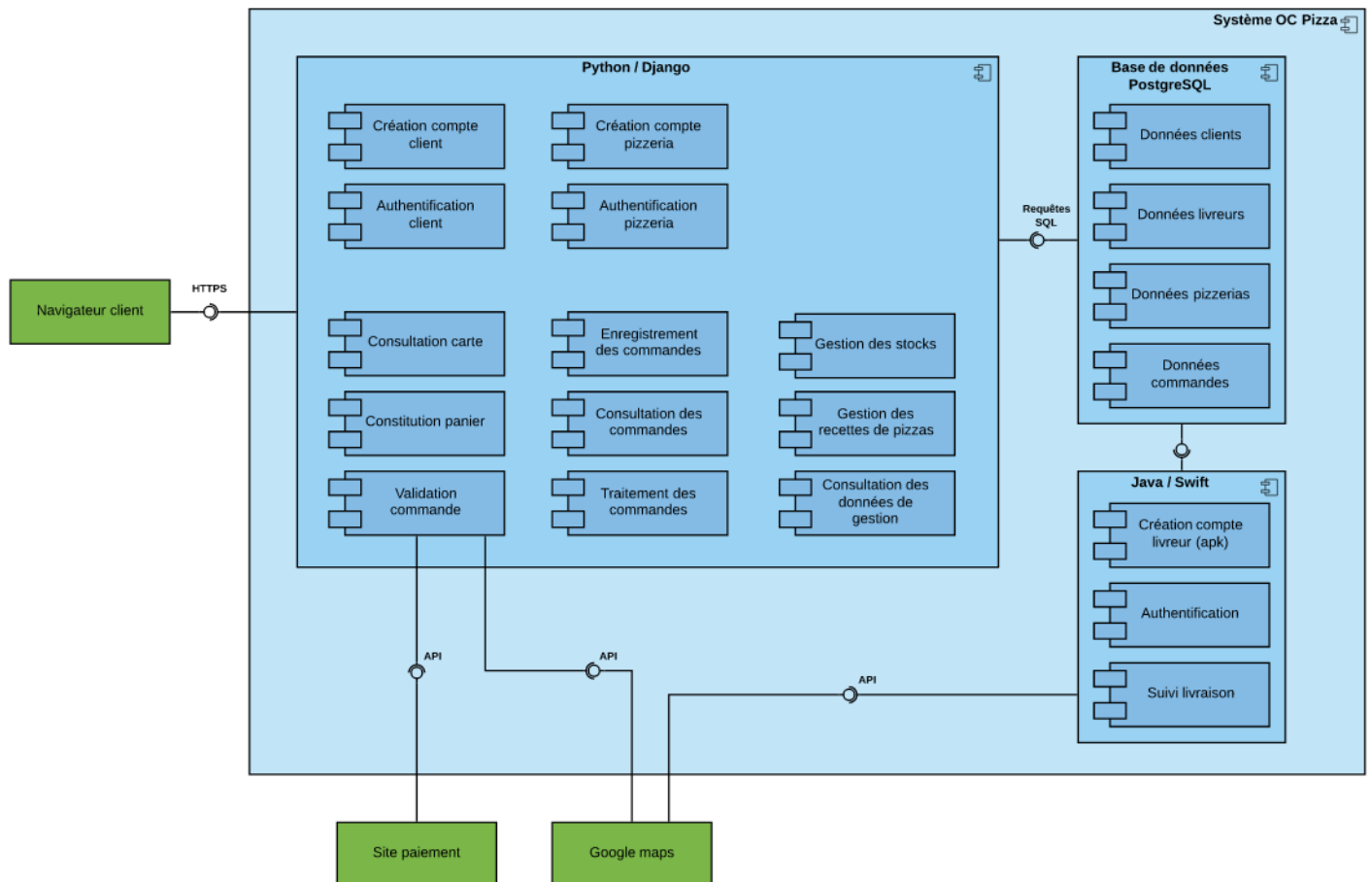
#### II.2.4 Base de données

Pour la base de données nous utilisons le SGBD PostgreSQL, avec un encodage en UTF-8.



## II.3 Diagramme de composants

Pour la réalisation de ce projet, IT Consulting & Development utilise un certains nombre de composants logiciels, dont le schéma ci-dessous représente les interactions.



La partie logicielle, développée en Python sous le framework Django, intègre plusieurs composants :

- création de compte, et authentification
- affichage de la liste des produits
- constitution du panier de commande
- validation de la commande, et paiement en ligne. La communication avec le site de paiement se fait via une API
- enregistrement et consultation de la liste des commandes
- traitement de la commande (changement de statuts)
- gestion des stocks d'ingrédients
- ajout/modification/suppression de recettes de pizzas
- affichage d'indicateurs de pilotage (données de gestion)



Pour fonctionner, ces composants communiqueront via des requêtes SQL avec la Base de donnée PostgreSQL. Celle-ci contient 4 grandes catégories de données :

- les données qui concernent les clients (adresse, nom, prénom identifiant, mot de passe, etc)
- les données qui concernent les pizzerias (siret, adresse, tel, etc)
- les données qui concernent les livreurs (nom, prénom, disponibilité, etc)
- les données concernant les commandes de pizzas (panier, quantité, pizzas, heure, etc)

De plus, les livreurs auront à disposition une application mobile. Celle-ci permettra une plus grande réactivité dans la validation du paiement à la livraison, ainsi qu'un fonctionnement simplifié de l'organisation des livraisons (disponibilité, itinéraire, etc). OC Pizza souhaite que les livreurs puissent utiliser leur téléphone personnel, c'est pourquoi l'application sera développée en Java et Swift.

Il est nécessaire que l'application communique avec la base de données, afin de récupérer les informations concernant les livreurs et les détails de la livraison (adresse, nom, heure, etc).

Nous utiliserons également l'API Google map pour deux usages :

- proposer un itinéraire de livraison aux livreurs
- le client effectue sa commande automatiquement auprès du magasin de sa ville

Le site OC Pizza sera consultable, via protocole https, depuis n'importe quel navigateur client (Firefox, IE, Chrome, etc).

Le site étant développé de manière responsive, une application mobile pour les clients n'est pas prévue. C'est pourquoi, l'application livreur sera au format apk, pour qu'elle ne puisse pas être installée de manière publique.



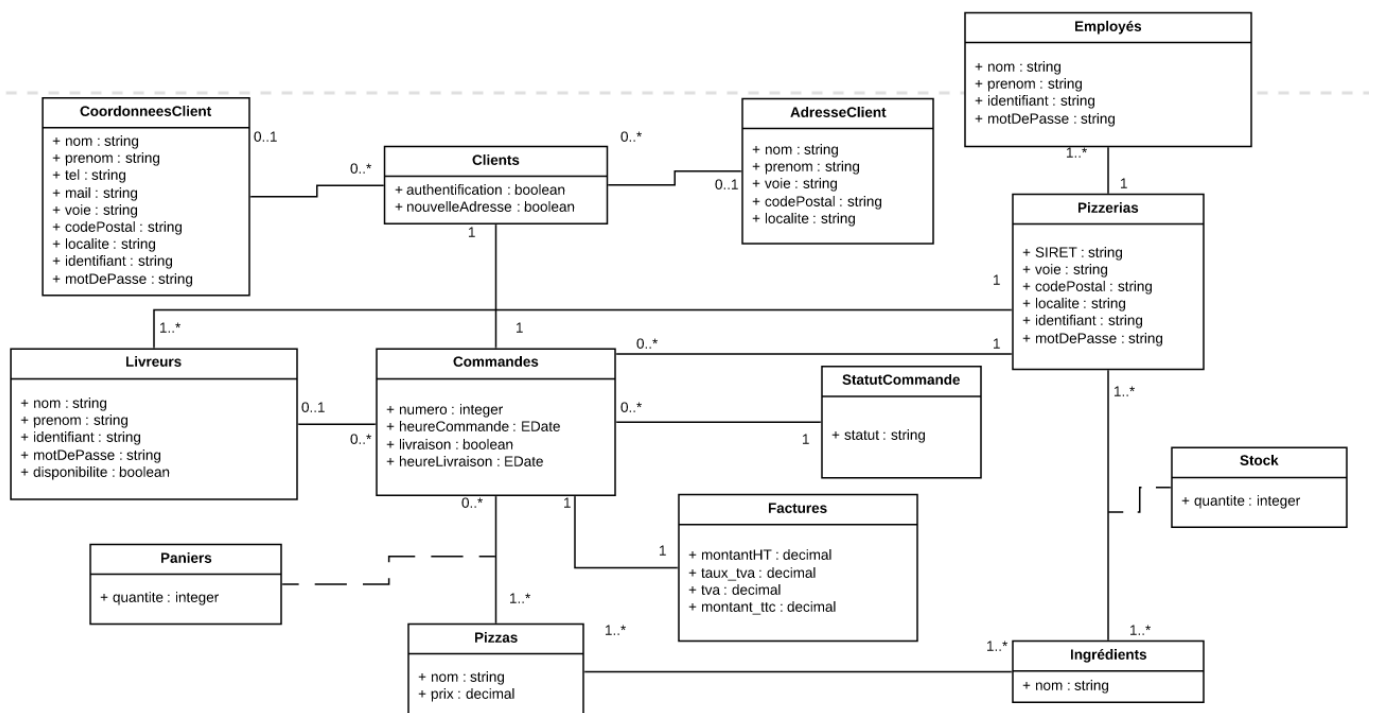
## III. Architecture de l'application

### III.1 Description du domaine fonctionnel

#### III.1.1 Diagramme de classes

Ce schéma représente les différentes classes utilisées pour le système prévu pour l'entreprise OC Pizza.

Les interactions entre les différentes classes sont mises en évidence.



#### III.1.2 Interactions

- CoordonneesClient – Clients : Association **One-to-many**, car un client peut avoir un compte utilisateur (ou pas). Et un compte client peut apparaître pour plusieurs clients (un nouveau client à chaque nouvelle commande).
- AdresseClient – Clients : Association **One-to-many**, idem association précédente. L'adresse client concerne les clients qui commandes sans avoir de compte client, ou un compte client mais avec une adresse différente (le client n'est pas chez lui par exemple).
- Clients – Commandes : Association **One-to-one**, une commande est affiliée à un client unique.
- Livresurs – pizzerias : Association **One-to-many**, un livreur est rattaché à une pizzeria et une pizzeria peut contenir plusieurs livreurs.

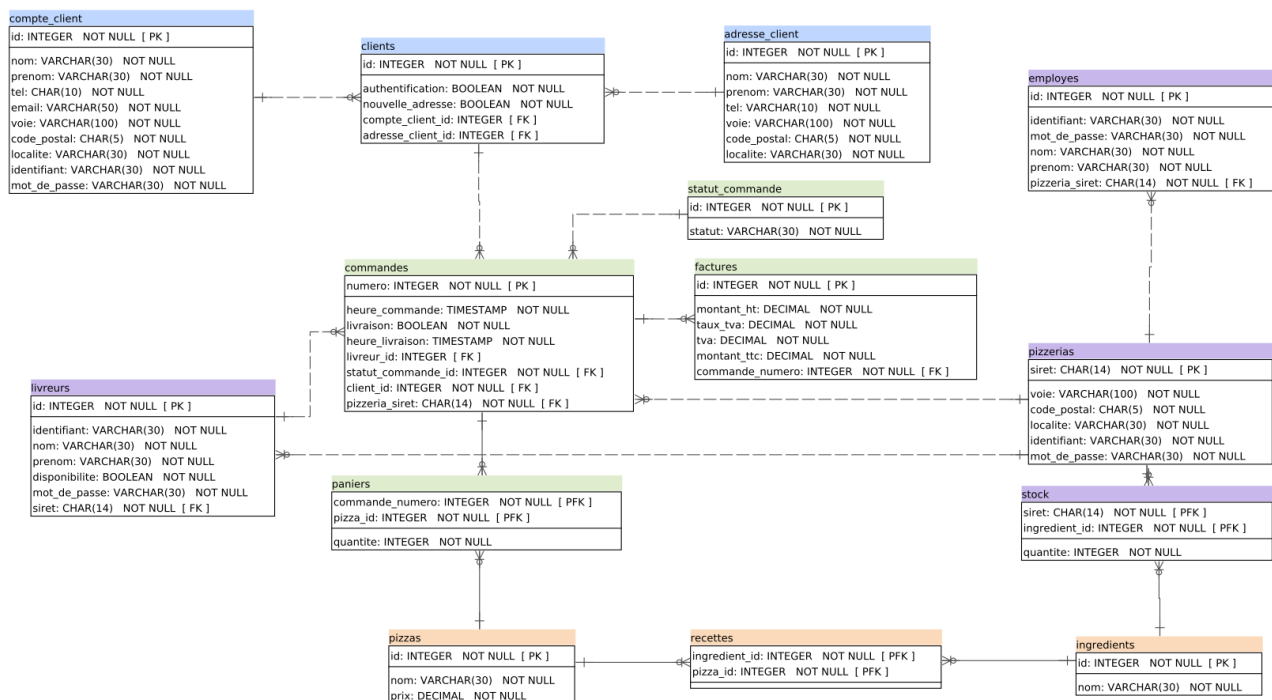


- Livreurs – commandes : Association **One-to-many**, un livreur peut être affecté à plusieurs commandes, mais une commande n'est liée qu'à un seul livreur (ou 0).
- Commandes – StatutCommande : Association **One-to-many**, une commande ne possède qu'un seul statut, mais un statut peut concerner plusieurs commandes.
- Commandes – pizzerias : Association **One-to-many**, une commande est rattachée à une seule pizzeria, mais une pizzeria peut avoir plusieurs commandes.
- Pizzerias – ingrédients : Association **Many-to-many**, un ingrédient peut se trouver dans différentes pizzerias, et une pizzeria peut contenir plusieurs ingrédients. La **classe d'association** Stock permet de connaître la quantité d'ingrédients pour chaque pizzeria.
- Pizzas – ingrédients : Association **Many-to-many**, une pizza peut contenir plusieurs ingrédients, et un ingrédient peut se trouver dans plusieurs pizzas.
- Pizzas – commandes : Association **Many-to-many**, une commande peut contenir plusieurs pizzas, et une pizza peut se trouver dans plusieurs commandes. La **classe d'association** Paniers permet d'indiquer la quantité de chaque pizza dans chaque commandes.

## V. Architecture de la base de données

Le schéma suivant représente les différentes tables qui composent la base de données.

Pour plus de lisibilité, ce schéma est disponible en annexe.







## VI. Architecture de déploiement

L'application d'OC Pizza sera hébergée sur un serveur virtuel DigitalOcean pour les raisons suivantes :

- Hébergement dans le Cloud, forte puissance de calcul et bande passante élevée à moindre coût
- Offres professionnelles avec des prix attractifs
- Scalabilité de la mémoire, du processeur, du disque dur et de la bande passante en quelques clics

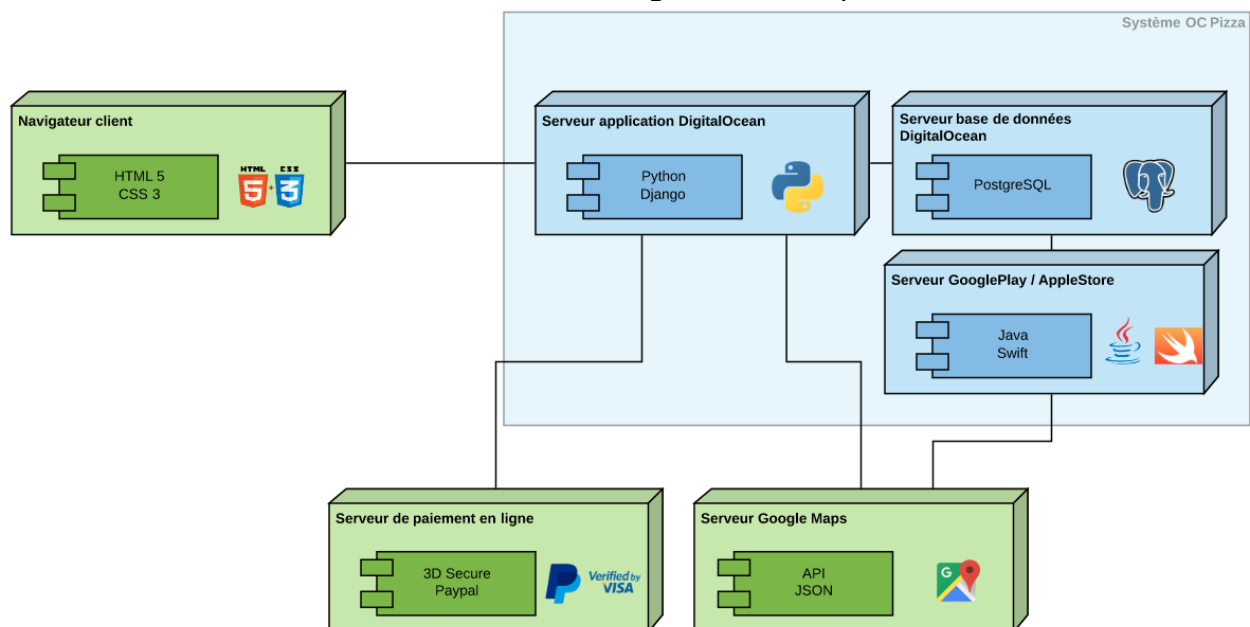
Pour les mêmes raisons, le serveur de la base de donnée sera également hébergée chez DigitalOcean.

L'application mobile sera, quand-à-elle hébergée sur les serveurs GooglePlay (pour Android), et l'AppStore (pour Apple).

Les serveurs OVH et mobiles communiquerons avec les serveurs GoogleMap, pour permettre l'utilisation de l'API GoogleMap, via fichiers JSON.

Les technologies de paiement permises seront 3D Secure et Paypal, donc il est nécessaire également de pouvoir communiquer avec leurs serveurs respectifs.

Toutes ces informations sont résumées dans le diagramme de déploiement ci-dessous :





## VII. Points particuliers

### VII.1 Environnement de développement

Pour le développement de l'application, nous utilisons un environnement virtuel afin de gérer les dépendances relatives au projet. De plus, cela permet d'éviter les problèmes liés aux conflits de version qui peuvent intervenir suite à des mises à jour des logiciels.

Nous utilisons Virtualenv pour la création et la gestion de notre environnement virtuel.

### VII.2 Monitoring

Pour monitorer les performances du serveur, nous utilisons NewRelic. Cela permettra de contrôler la charge du serveur, l'utilisation des différents composants (RAM, CPU, etc).

OVH propose de nombreuses offres professionnelles d'hébergement, ce monitoring permettra d'évaluer s'il est nécessaire d'évoluer vers un serveur plus performant, en fonction de l'évolution du groupe OC Pizza, et de l'affluence de ses clients sur l'application.

### VII.3 Gestion des logs

Les logs de l'application seront surveillés grâce à Sentry. Cela permettra de contrôler les alertes/erreurs éventuelles qui peuvent intervenir.