



# OC Pizza

## Application Web

Dossier de conception technique

Version 1.1

**Auteur**

Eri Schön

Développeur Python



## TABLE DES MATIÈRES

<b>Versions</b>	<b>4</b>
<b>Introduction</b>	<b>5</b>
<b>Objet du document</b>	<b>6</b>
<b>Le domaine fonctionnel</b>	<b>7</b>
<b>Référentiel</b>	<b>7</b>
<b>Règles de gestion</b>	<b>7</b>
<b>Architecture Technique</b>	<b>8</b>
<b>Application Web</b>	<b>9</b>
<b>Authentification</b>	<b>9</b>
<b>Interfaces</b>	<b>9</b>
<b>Gestion des comptes clients</b>	<b>9</b>
<b>Création de commandes</b>	<b>9</b>
<b>Paieement</b>	<b>9</b>
<b>Back Office</b>	<b>9</b>
<b>Suivi de commandes</b>	<b>9</b>
<b>Livraison</b>	<b>9</b>
<b>Gestion administrative</b>	<b>9</b>
<b>Architecture de Déploiement</b>	<b>9</b>
<b>Serveur de Base de données</b>	<b>11</b>
<b>Serveur d'application</b>	<b>11</b>
<b>Serveur Web</b>	<b>11</b>
<b>Glossaire</b>	<b>12</b>



**iö**





# 1 - VERSIONS

Auteur	Date	Description	Version
Eri Schön	23/02/21	Création du document	1.0
Eri Schön	23/02/21	Ajout du contenu	1.1

# 2 - INTRODUCTION

## 2.1 - Objet du document

Le présent document constitue le dossier de conception Technique en vue de la mise en place d'un nouveau système informatique pour l'ensemble des pizzerias du groupe OC Pizza.

Les éléments du présents dossiers découlent :

de notes prises lors d'une discussion téléphonique entre Lola (co-fondatrice de OC Pizza) et Alexandra (responsable chez IT Consulting & Development).

d'une recherche terrain de la part d'Eri Schön (développeur junior chez IT Consulting & Development) auprès de différentes Pizzerias.

## 2.2 - Contexte et besoins du client

### Contexte

« OC Pizza » est un jeune groupe de pizzeria en plein essor. Créé par Franck et Lola, le groupe est 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 6 mois.



Le système informatique actuel ne correspond plus aux besoins du groupe car il ne permet pas une gestion centralisée de toutes les pizzerias.

De plus, il est très difficile pour les responsables de suivre ce qui se passe dans les points de ventes.

Enfin, les livreurs ne peuvent pas indiquer « en live » que la livraison est effectuée.

## **Besoins**

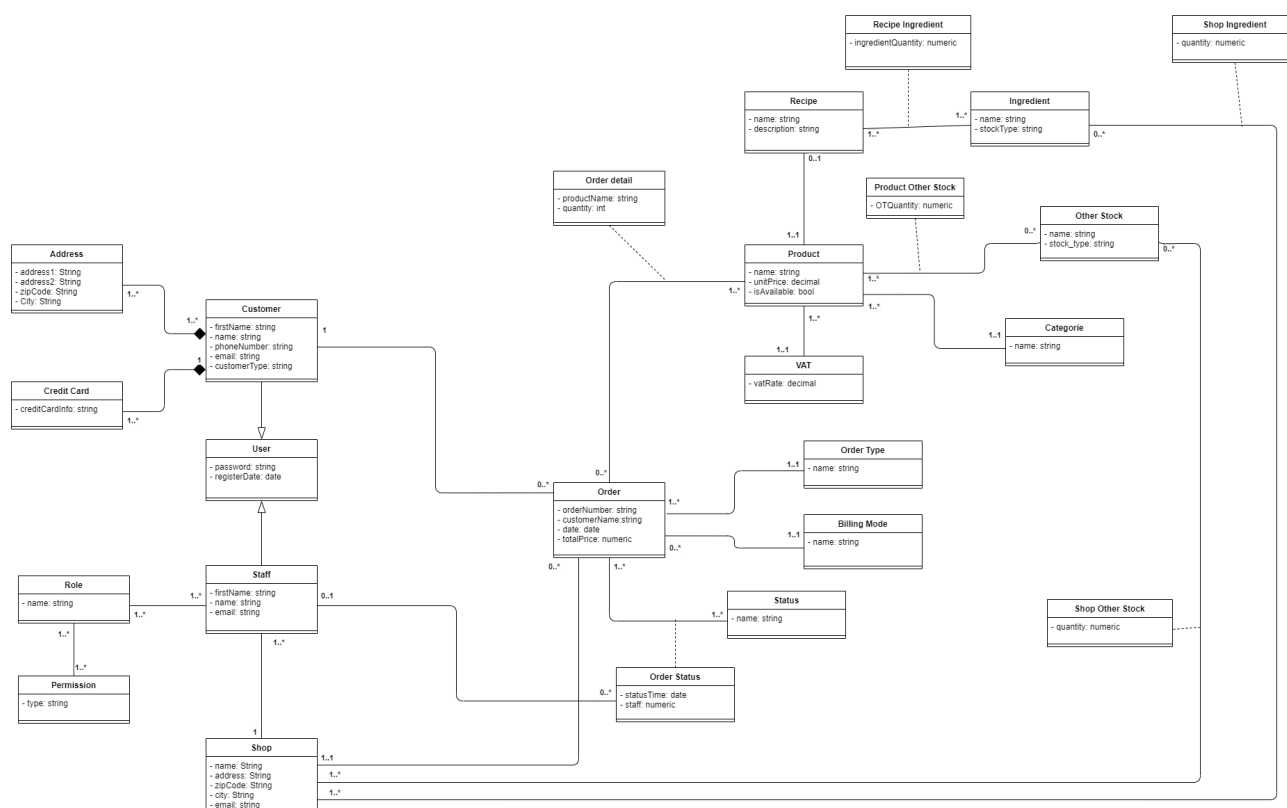
Besoins exprimés par le client :

- être plus efficace dans la gestion des commandes, de leur réception à leur livraison en passant par leur préparation.
- suivre en temps réel les commandes passées, en préparation et en livraison.
- suivre en temps réel le stock d'ingrédients restants pour savoir quelles pizzas peuvent encore être réalisées.
- proposer un aide-mémoire aux pizzaiolos indiquant la recette de chaque pizza.
- 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.



## 3 - LE DOMAINE FONCTIONNEL

### 3.1 - Référentiel



#### 3.1.1 - Règles de gestion

- **Customer - Address**
  - Un Customer doit avoir une Address et peut en avoir plusieurs.
  - Une Address doit avoir au moins un Customer et peut avoir plusieurs Customer.
- **Customer - Credit Card**
  - Un Customer doit avoir une Credit Card et peut en avoir plusieurs.
  - Une Credit Card ne peut avoir qu'un seul Customer



- **Customer - User**
  - Un Customer ne peut avoir qu'un User.
  - Un User ne peut avoir qu'un Customer.
- **Customer - Order**
  - Un Customer peut ne pas avoir d'Order ou peut en avoir plusieurs.
  - Un Order ne peut avoir qu'un Customer.
- **Role - Permission**
  - Un Role doit avoir une Permission et peut en avoir plusieurs
  - Une Permission doit avoir un Role et peut en avoir plusieurs.
- **Staff - Role**
  - Un Staff doit avoir un Role et peut en avoir plusieurs.
  - Un Role doit avoir un Staff et peut en avoir plusieurs.
- **Staff - User**
  - Un Staff ne peut avoir qu'un User.
  - Un User ne peut avoir qu'un Staff
- **Staff - Order Status**
  - Un Staff peut ne pas avoir d'Order Status et peut en avoir plusieurs.
  - Un Order Status peut ne pas avoir de Staff et n'en n'a qu'un seul.
- **Staff - Shop**
  - Un Staff ne peut avoir qu'un Shop.
  - Un Shop doit avoir un Staff et peut en avoir plusieurs.
- **Shop - Order**
  - Un Shop peut ne pas avoir d'Order et peut en avoir plusieurs.
  - Un Order doit avoir un Shop et un seul.
- **Shop - Ingredient**
  - Un Shop peut ne pas avoir d'Ingredient et en avoir plusieurs.
  - Un Ingredient doit avoir un Shop et peut en avoir plusieurs.
- **Shop - Other Stock**
  - Un Shop peut ne pas avoir d'Other Stock et en avoir plusieurs.
  - Un Other Stock doit avoir un Shop et peut en avoir plusieurs.
- **Shop - Shop Ingredient**
  - Un Shop peut ne pas avoir de Shop Ingredient et en avoir plusieurs.
  - Un Shop Ingredient ne peut avoir qu'un seul Shop.
- **Shop - Shop Other Stock**
  - Un Shop peut ne pas avoir de Shop Other Stock et en avoir plusieurs.
  - Un Shop Other Stock ne peut avoir qu'un seul Shop.
- **Order - Product**
  - Un Order doit avoir au moins 1 Product et peu en avoir plusieurs.
  - Un Product peut ne pas avoir d'Order et en avoir plusieurs.
- **Order - Order Detail**



- Un Order doit avoir au moins un Order Detail et peut en avoir plusieurs.
  - Un Order Detail ne peut avoir qu'un Order.
- **Order - Order Type**
  - Un Order ne peut avoir qu'un Order Type.
  - Un Order Type doit avoir un Order et peut en avoir plusieurs.
- **Order - Status**
  - Un Order doit avoir un Status et peut en avoir plusieurs.
  - Un Status doit avoir un Order et peut en avoir plusieurs.
- **Order - Order Status**
  - Un Order ne peut avoir qu'un Order Status.
  - Un Order Status ne peut avoir qu'un Status.
- **Order - Billing Mode**
  - Un Order ne peut avoir qu'un Billing Mode.
  - Un Billing Mode peut ne pas avoir d'Order et peut en avoir plusieurs.
- **Product - Recipe**
  - Un Product peut ne pas avoir de Recipe.
  - Une Recipe ne peut avoir qu'un Product.
- **Product - Other Stock**
  - Un Product peut ne pas avoir d'Other Stock.
  - Un Other Stock doit avoir un Product et peut en avoir plusieurs.
- **Product - Product Other Stock**
  - Un Product peut ne pas avoir de Product Other Stock et ne peut en avoir qu'un.
  - Product Other Stock ne peut avoir qu'un Product
- **Product - Categorie**
  - Product ne peut avoir qu'une Categorie.
  - Categorie doit avoir un Product et peut en avoir plusieurs.
- **Product - VAT**
  - Product ne peut avoir qu'une VAT.
  - VAT doit avoir au moins un Product et peut en avoir plusieurs.
- **Recipe - Ingredient**
  - Recipe doit avoir au moins un Ingredient et peut en avoir plusieurs.
  - Ingredient doit avoir au moins un Recipe et peut en avoir plusieurs.
- **Recipe - Recipe Ingredient**
  - Recipe doit avoir au moins un Recipe Ingredient et peut en avoir plusieurs.
  - Recipe Ingredient ne peut avoir qu'un Recipe.



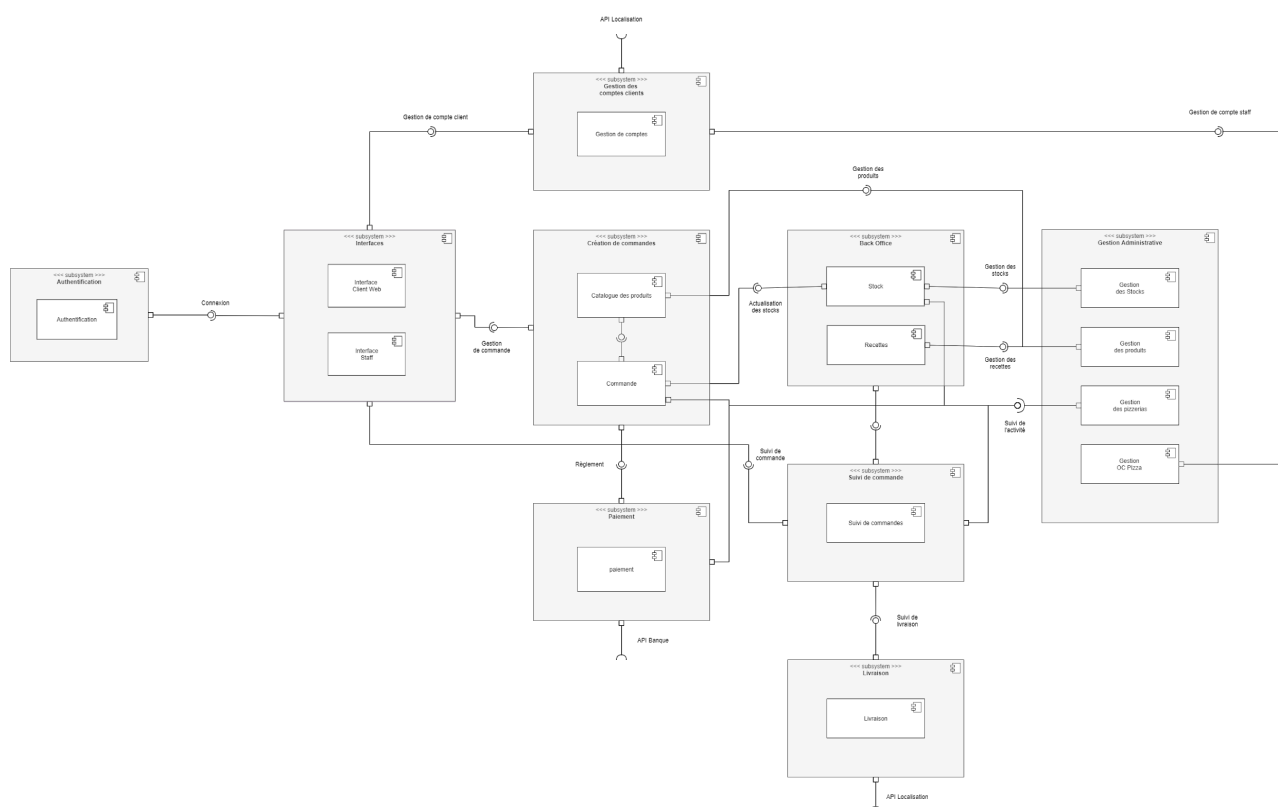


## 4 - ARCHITECTURE TECHNIQUE

### 4.1 - Application Web

La pile logicielle est la suivante :

- Application : **Python 3.8**
- Serveur d'application : **Gunicorn 20.0.4**
- Serveur web : **NGINX 1.18**
- Base de données : **PostgreSQL 13.2**



#### 4.1.1 - Composant Authentification

- *Authentification* à l'application
- *Authentification* propose :
  - l'interface de connexion



#### 4.1.2 - Composant Interfaces

- *Interfaces* s'occupe de l'interface client et de l'interface pour le Staff.
- *Interfaces* requiert :
  - l'interface de connexion
- *Interfaces* propose :
  - une interface de gestion de compte client
  - une interface de gestion de commande
  - une interface de suivi de commande

#### 4.1.3 - Composant Gestion des comptes clients

- *Gestion des comptes clients* gère les comptes clients (CRUD).
- *Gestion des comptes clients* requiert :
  - l'interface gestion de compte client
  - l'API de localisation
- *Gestion des comptes clients* propose :
  - une interface de gestion de compte staff

#### 4.1.4 - Composant Création de commandes

- *Création de commandes* gère les commandes.
- *Création de commandes* requiert :
  - une interface d'actualisation des stocks
- *Création de commandes* propose :
  - une interface de gestion de commande
  - une interface de gestion des produits
  - une interface de suivi de l'activité
  - une interface de règlement

#### 4.1.5 - Composant Paiement

- *Paiement* s'occupe du paiement des commandes.
- *Paiement* requiert :
  - une API banque
  - une interface règlement
- *Paiement* propose :
  - une interface de suivi de l'activité



#### **4.1.6 - Composant Back Office**

- Back Office s'occupe des stocks et des recettes.
- Back Office propose :
  - une interface de suivi
  - une interface d'actualisation des stocks
  - une interface de gestion des stocks
  - une interface de suivi de l'activité

#### **4.1.7 - Composant Suivi de commandes**

- *Suivi de commandes* s'occupe du suivi des commandes de la création à la clôture.
- *Suivi de commandes* requiert :
  - une interface de suivi de livraison
  - une interface de suivi de commande
  - une interface back office
  - une interface suivi de commande
- *Suivi de commandes* propose :
  - un interface de suivi de l'activité

#### **4.1.8 - Composant Livraison**

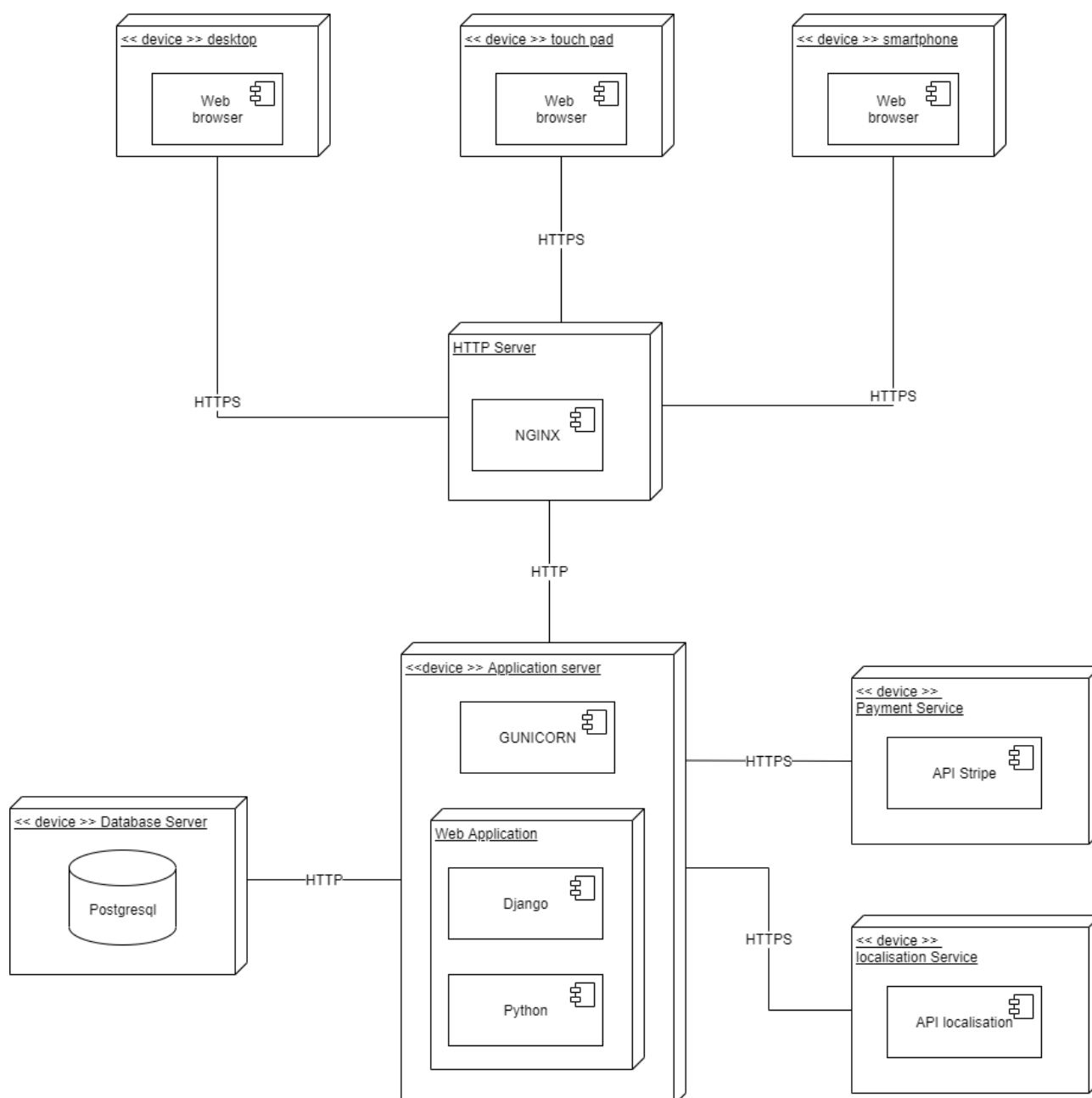
- *Livraison* gère la localisation et la navigation pour la livraison de commandes
- *Livraison* requiert :
  - l'API de localisation
- *Livraison* propose :
  - une interface de suivi de livraison

#### **4.1.9 - Composant Gestion administrative**

- *Gestion administrative* gère toute la partie administrative d'OC Pizza
- *Gestion administrative* requiert :
  - l'interface de gestion des comptes staff
  - l'interface de suivi de l'activité
  - l'interface de gestion des recettes
  - l'interface de gestion des produits
  - l'interface de gestion des stocks



## 5 - ARCHITECTURE DE DÉPLOIEMENT





## 5.1 - Serveur de Base de données

### Description

Le serveur de Base de données stocke et gère les données du Système de gestion des Pizzerias.

### Caractéristiques techniques

- PostgreSQL 13.2 :
  - Serveur de Base de Données Relationnelle SQL
  - Systèmes d'exploitation principaux supportés :
    - Linux
    - MacOS
    - Windows
  - Licence : Liberal Open Source Licence

## 5.2 - Serveur d'application

### Description

Gunicorn est un serveur dit « WSGI » (pour Web Server Gateway Interface) chargé de faire l'interface entre une application Web développée en Python et un serveur web.

### Caractéristiques techniques

- GUNICORN 20.0.4 :
  - Serveur WSGI
  - Systèmes d'exploitation principaux supportés :
  - Licence : MIT Licence

## 5.3 - Serveur Web

### Description

Le serveur Web (ou serveur HTTP) réceptionne les requêtes clients et renvoie les informations demandées sous forme de pages Web en HTML.

### Caractéristiques techniques

- NGINX 1.18 :
  - Serveur Web et Proxy inverse
  - Systèmes d'exploitation principaux supportés :
    - Linux



iö

- MacOS
- Windows
- Licence : BSD 2-clauses

## 6 - GLOSSAIRE
