

**Contact : Jean-Baptiste Baillet**

[jbaillet@itconsultingdevelopment.com](mailto:jbaillet@itconsultingdevelopment.com)

# **Documentation Technique**

## **du Projet**

## **OC Pizzas**



Date de création

Date de modification

Échéance livraison

## Table des matières

1- Domaine Fonctionnel.....	3
11- Cas d'utilisation (spécifications fonctionnelles).....	3
112- Package Authentification.....	3
113- Package Fonctionnalités Client.....	3
114- Package Préparation Commandes.....	3
115- Package Business Commandes.....	3
12- Diagramme de classes.....	4
2- Composants du système.....	5
21- Composants internes.....	5
211- Authentification.....	5
212- Utilisateur.....	6
213- Adresse.....	6
214- Commande.....	6
215- Produit.....	6
216- Aide-Mémoire.....	6
217- Ingrédient.....	6
218- Stock-Ingrédient.....	6
219- Pizzeria.....	6
220- Paiement.....	6
22- Composants externes.....	7
221- Banque.....	7
3- Déploiement de l'application.....	8
31- Diagramme de déploiement.....	8
4- Modèle Physique de données.....	9
5- Sigles.....	9
6- Glossaire.....	9

# 1- Domaine Fonctionnel

## 11- Cas d'utilisation (spécifications fonctionnelles)

### **111- Package Authentication**

- Créer Compte Client
- S'authentifier
- Réinitialiser mot de passe

### **112- Package Fonctionnalités Client**

- Consulter le catalogue
- Commander en ligne
- Payer la commande en ligne
- Annuler la commande
- Modifier la commande
- Suivre l'avancée de la commande client

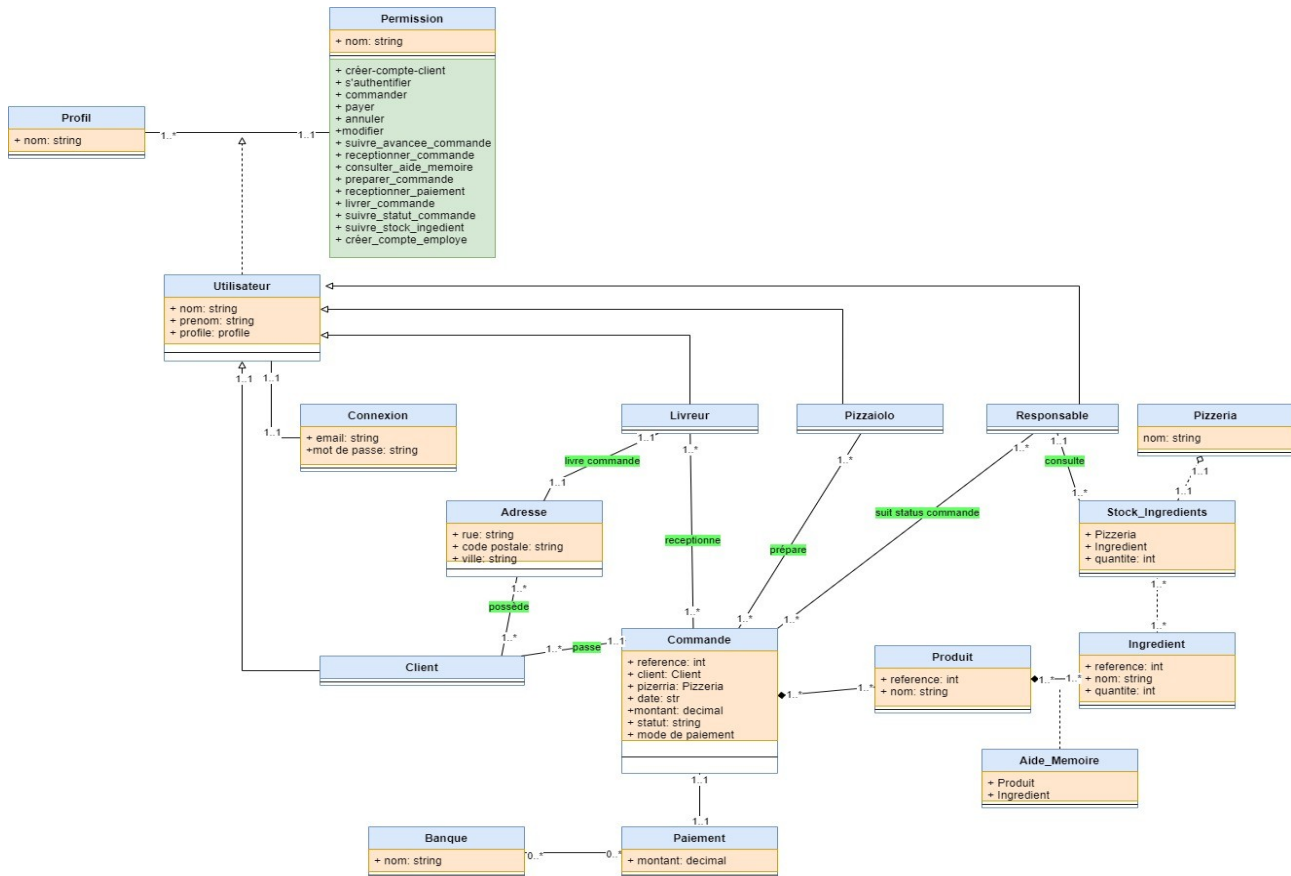
### **113- Package Préparation Commandes**

- Réceptionner la commande du client
- Consulter l'aide-mémoire recettes
- Préparer la commande du client
- Réceptionner la commande prête à être livrée
- Réceptionner le paiement
- Livrer la commande au client

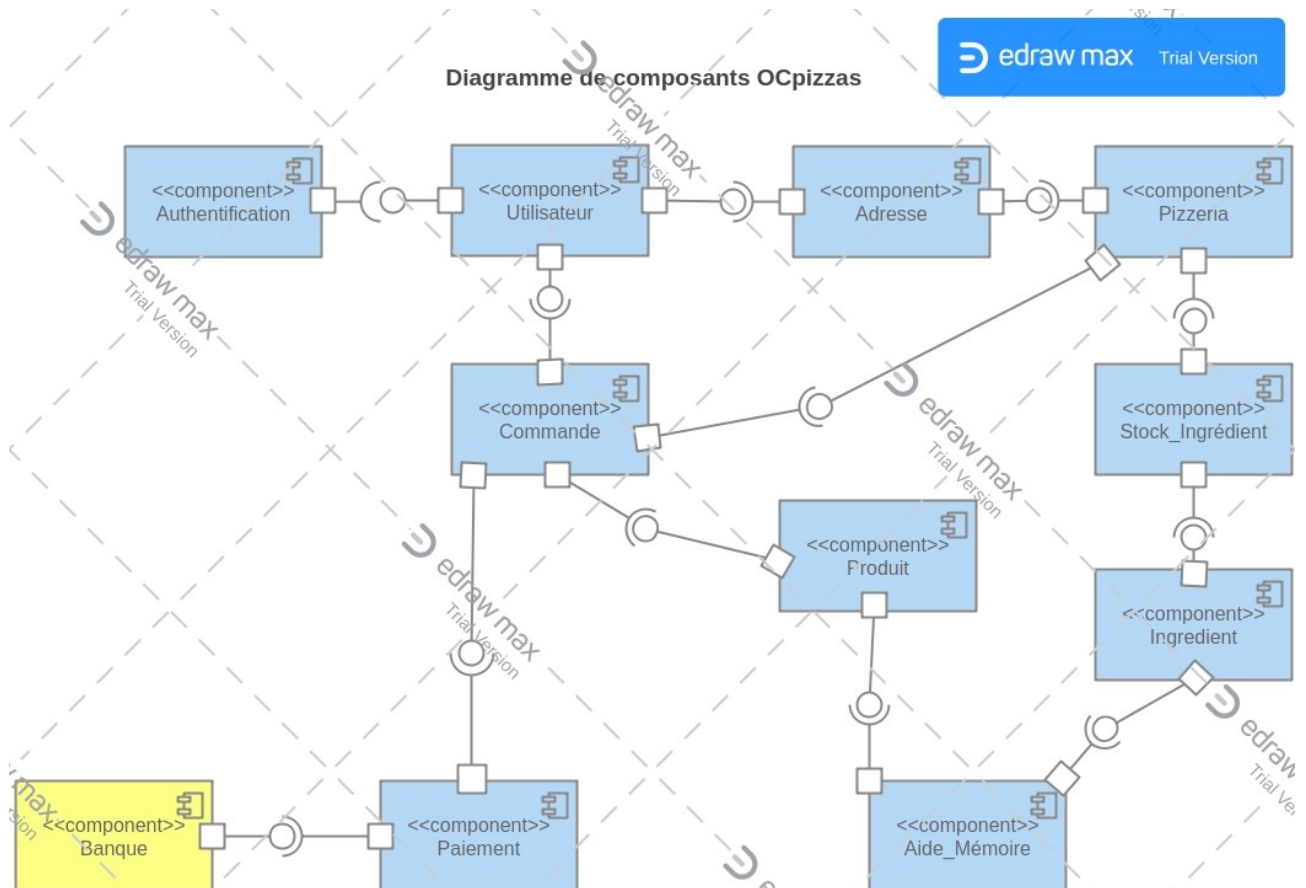
### **114-Package Business Commandes**

- Suivre en temps réel le statut des commandes
- Suivre le stock d'ingrédients
- Créer les comptes Pizzaiolo et Livreur

## 12- Diagramme de classes



## 2- Composants du système



## 21- Composants internes

### 211- Authentification

- **Authentification** fournit l'interface **Création\_Compte**, afin de permettre à l'**Utilisateur** de créer un compte.
- **Authentification** fournit l'interface **Authentification**, afin de permettre à l'**Utilisateur** de s'authentifier et d'accéder à sa session.
- **Authentification** fournit l'interface **Réinitialisation\_mot\_de\_passe**, afin de permettre à l'**Utilisateur** de réinitialiser son mot de passe.
- **Authentification** requiert l'Interface **Utilisateur**, afin de récupérer les informations **Utilisateur**.

## **212- Utilisateur**

- **Utilisateur** fournit l'interface **Commande**, afin de permettre à l'**Utilisateur** de commander, annuler, ou modifier la commande.
- **Utilisateur** requiert l'Interface **Adresse**, afin de permettre à l'**Utilisateur** de fournir l'adresse de livraison.

## **213- Adresse**

- **Adresse** fournit l'Interface **Adresse**, afin qu'**Utilisateur** et **Pizzeria** reçoivent une adresse.

## **214- Commande**

- **Commande** fournit l'Interface **Paielement**, afin que **Paielement** reçoive le montant de la **Commande**.
- **Commande** requière l'Interface **Pizzeria**, afin de relier la **Commande** à la **Pizzeria**.
- **Commande** requiert l'Interface **Produit**, afin de lister les produits que l'**Utilisateur** gère dans la commande.

## **215- Produit**

- **Produit** fournit l'Interface **Produit**, afin que **Commande** et **Aide-Mémoire** reçoivent **Produit**

## **216- Aide-Mémoire**

- **Aide-Mémoire** requiert l'Interface **Produit** et l'Interface **Ingrédient**, afin de relier le produit aux ingrédients nécessaire à la recette.

## **217- Ingrédient**

- **Ingrédient** fournit l'Interface **Ingrédient** à **Aide-mémoire** et **Stock-Ingrédients**.

## **218- Stock-Ingrédient**

- **Stock-Ingrédients** fournit l'Interface **Stock-Ingrédient** à **Pizzeria**, afin de relier le stock à la pizzeria.
- **Stock-Ingrédient** requiert l'Interface **Ingrédient**.

## **219- Pizzeria**

- **Pizzeria** fournit l'Interface **Commande**
- **Pizzeria** requière l'Interface **Stock-Ingrédient** et l'Interface **Adresse**

## **220- Paiement**

- **Paiement** fournit l'Interface **Paiement** à **Banque**.

- Paiement requière l'Interface Banque

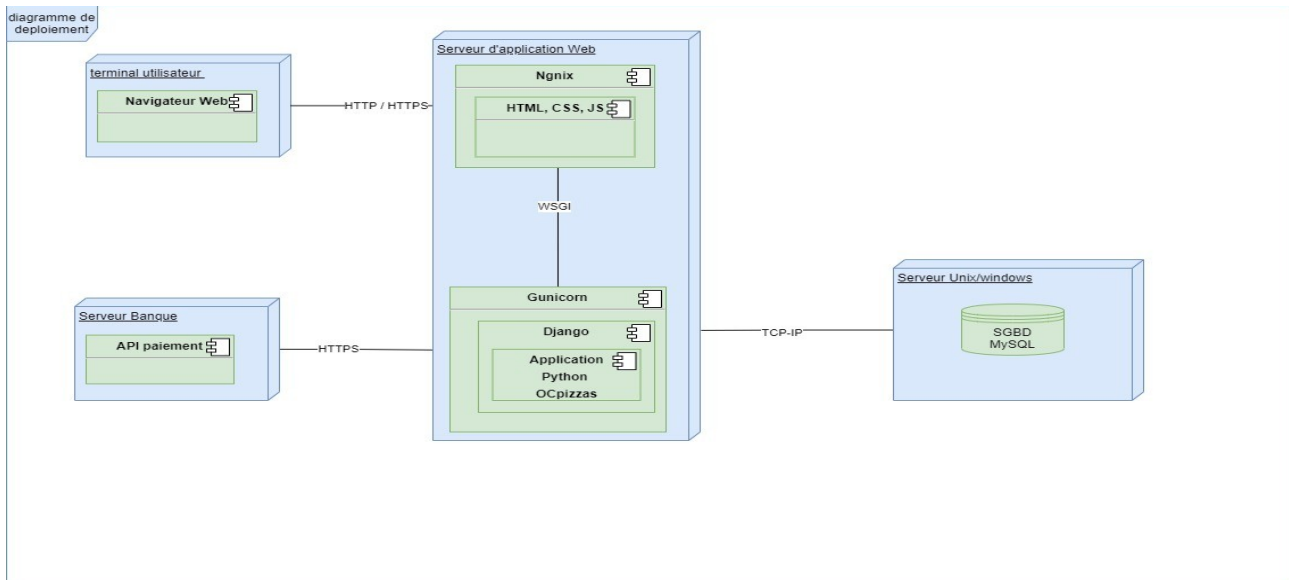
## **22- Composants externes**

### **221- Banque**

- Banque fournit l'Interface Banque à Paiement

## 3- Déploiement de l'application

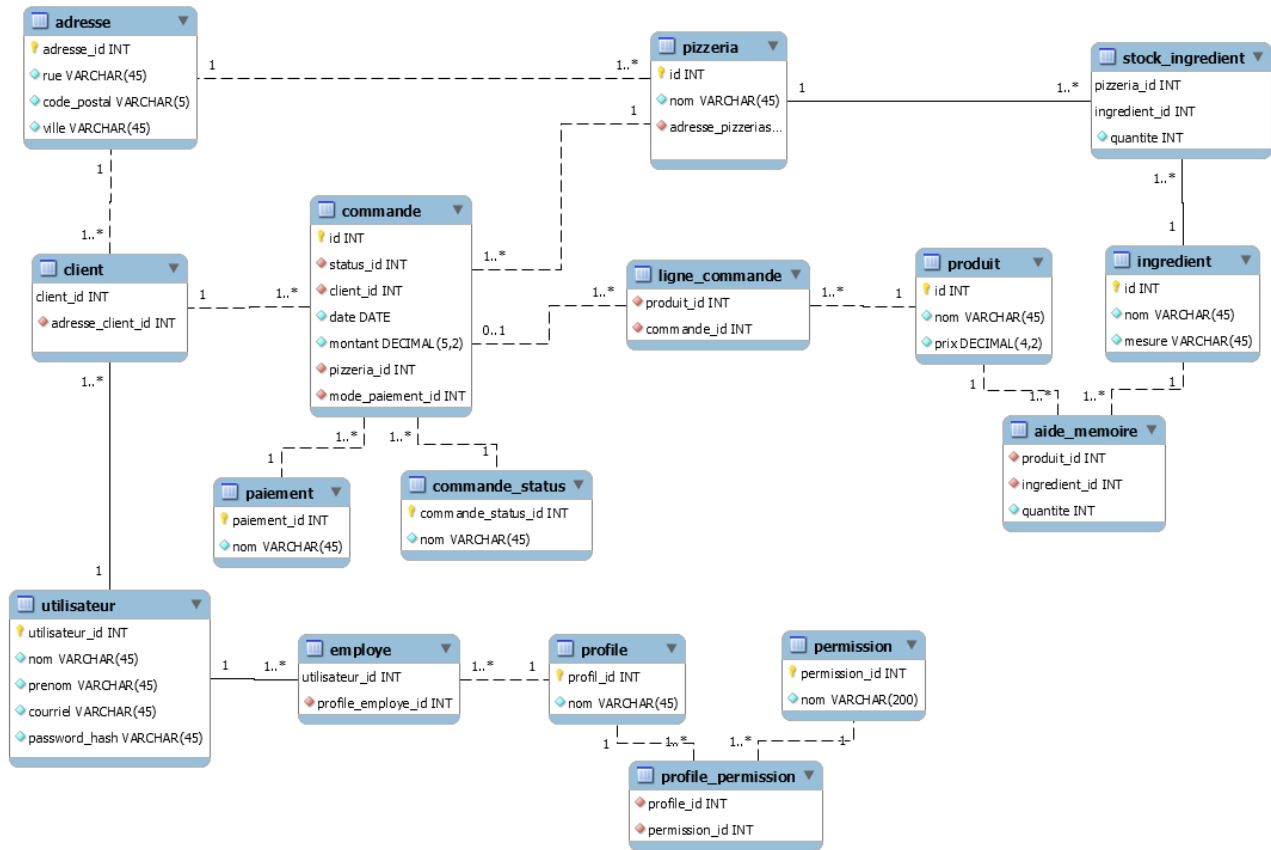
### 31- Diagramme de déploiement



- Serveur d'application : [Gunicorn 20.1.0](#)
- Serveur web : [Nginx 1.20.0](#)
- Framework : [Django 3.2.2](#)
- Application : [Python 3.9](#)
- Système de Gestion de Base de Donnée : [MySQL 8.0.24](#)



## 4- Modèle Physique de données



## 5- Sigles

HTTP	<b>L'Hypertext Transfer Protocol</b> est un protocole de communication client-serveur développé pour le Web
WSGI	<b>Web Server Gateway Interface</b> est une spécification qui définit une interface entre des serveurs et des applications web pour le langage Python.
TCP/IP	<b>Transmission Control Protocol/Internet Protocol</b> est l'ensemble des protocoles utilisés pour le transfert des données sur Internet.
SGBD	<b>Système de Gestion de Base de Données</b> est un logiciel système servant à stocker, à manipuler ou gérer, et à partager des données dans une base de données, en garantissant la qualité, la pérennité et la confidentialité des informations, tout en cachant la complexité des opérations.
HTML	<b>HyperText Markup Language</b> est le langage de balisage conçu pour représenter les pages web.
CSS	<b>Cascading Style Sheets</b> forment un langage informatique qui décrit la présentation des documents HTML.
JS	<b>Javascript</b> est un langage de programmation de scripts principalement employé dans les pages web interactives.
UML	<b>Unified Modeling Language</b> est un langage de modélisation graphique à base de pictogrammes conçu comme une méthode normalisée de visualisation dans les domaines du développement logiciel et en conception orientée objet.