

# **OC Pizza**

## **Système de gestion de Pizzerias**

Dossier de conception technique

Version 1.0

**Auteur**  
BICHARI Mehdi  
*Développeur*

# TABLE DES MATIÈRES

<b>1 - Versions.....</b>	<b>3</b>
<b>2 - Introduction.....</b>	<b>4</b>
2.1 - Objet du document.....	4
<b>3 - Le domaine fonctionnel.....</b>	<b>5</b>
3.1 - Référentiel.....	5
3.1.1 - Règles de gestion.....	6
<b>4 - Architecture Technique.....</b>	<b>8</b>
4.1 - Application Web.....	8
4.1.1 - Composants Webpage.....	8
4.1.1.1 - Composant GUI.....	8
4.1.1.2 - Composant Authentication.....	9
4.1.1.3 - Composant Cart.....	9
4.1.1.4 - Composant Pay.....	9
4.1.2 - Composant Bank.....	9
4.1.3 - Composants Administration.....	9
4.1.3.1 - Composant Account.....	9
4.1.3.2 - Composant Contact.....	9
4.1.3.3 - Composant User.....	9
4.1.4 - Composants Logistic.....	9
4.1.4.1 - Composant Menu.....	10
4.1.4.2 - Composant Stock.....	10
4.1.4.3 - Composant Receipt.....	10
4.1.5 - Composant Database.....	10
4.1.6 - Composant Command.....	10
<b>5 - Architecture de Déploiement.....</b>	<b>11</b>
5.1 - Serveur de Base de données.....	11
5.2 - Serveur application.....	11
5.3 - Serveur web.....	11
<b>6 - Glossaire.....</b>	<b>12</b>

# 1 - VERSIONS

Auteur	Date	Description	Version
Mehdi	18/05/2020	Création du document	1.0

## 2 - INTRODUCTION

### 2.1 - Objet du document

Le présent document constitue le dossier de conception technique de l'application de gestion de pizzeria réalisée pour le client OC-Pizza

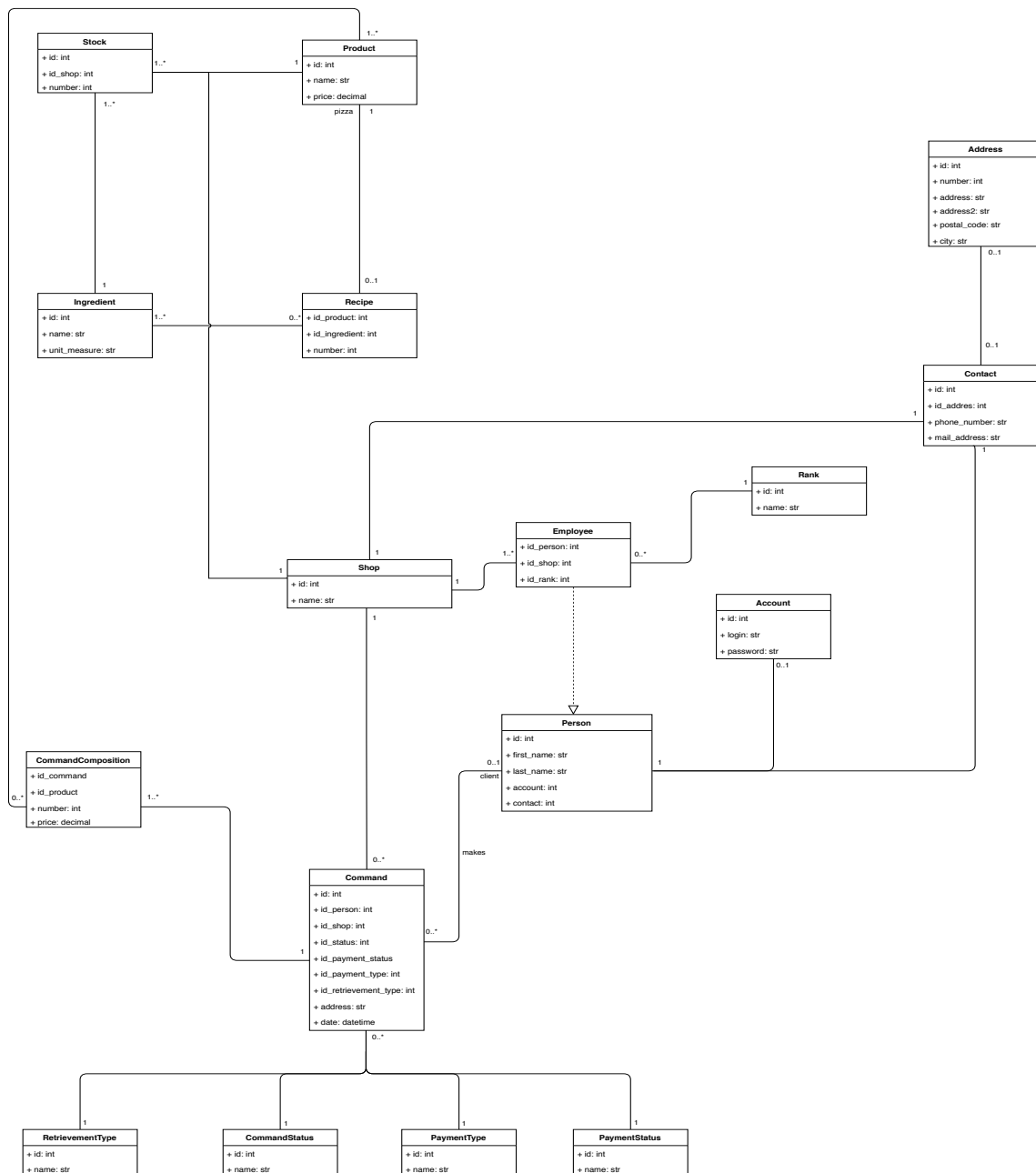
Nous développerons ici nos choix concernant le domaine fonctionnel de l'application, son architecture technique ainsi que son architecture de déploiement envisagée.

Les éléments du présents dossiers découlent :

- de l'étude de la demande du client
- d'une recherche approfondie des différentes technologies utilisées
- des bonnes pratiques en matière de développement d'application web

# 3 - LE DOMAINE FONCTIONNEL

## 3.1 - Référentiel



Ce diagramme représente les différents objets mis en relations dans l'application OC Pizza.

### **3.1.1 - Règles de gestion**

#### **Product - Stock**

Un Product pourra faire partie de plusieurs Stock. Un Stock concernera un seul Product.

#### **Stock - Ingrédient**

Un Ingredient pourra faire partie de plusieurs Stock. Un Stock concernera un seul Ingredient.

#### **Ingredient – Recipe**

Une Recipe pourra contenir plusieurs Ingredient. Un Ingredient pourra faire partie d'aucune ou de plusieurs Recipe.

#### **Recipe – Product**

Une Recipe pourra expliquer la composition d'un Product. Un Product pourra avoir ou non une Recipe.

#### **Product – CommandComposition**

Un Product pourra ou non faire partie d'une CommandComposition. Une CommandComposition renverra vers un Product.

#### **CommandComposition – Command**

Une Command sera composée d'une ou plusieurs CommandComposition. Une CommandComposition renverra vers une Command.

#### **Shop – Stock**

Un Shop contiendra un ou plusieurs stocks. Un Stock renverra lui vers un Shop.

#### **Shop – Command**

Un Shop pourra avoir aucune ou plusieurs Command. Une commande renverra forcément vers un Shop.

#### **Command – RetrievementType**

Une Command contiendra un RetrievementType. Un RetrievementType pourra être associé ou non à une Command.

#### **Command – CommandStatus**

Une Command contiendra un CommandStatus. Un CommandStatus pourra être associé ou non à une Command.

#### **Command – PaymentType**

Une Command contiendra un PaymentType. Un PaymentType pourra être associé ou non à une Command.

#### **Command – PaymentStatus**

Une Command contiendra un PaymentStatus. Un PaymentStatus pourra être associé ou non à une Command.

#### **Command – Person**

Une Command pourra être associée à aucune ou une personne. Une personne pourra avoir zéro ou plusieurs commandes.

#### **Shop – Employee**

Un Shop pourra avoir un ou plusieurs Employee. Un Employee pourra travailler dans un Shop.

#### **Rank – Employee**

Un Employee aura un Rank. Un Rank pourra être associé à zéro ou plusieurs Employee.

#### **Person – Account**

Une Person pourra avoir ou non un Account. Un Account renverra vers une Person.

#### **Person – Contact**

Une Person aura un Contact. Un Contact renverra vers une Person.

#### **Shop – Contact**

Un Shop aura un Contact. Un Contact renverra vers un Shop.

#### **Address – Contact**

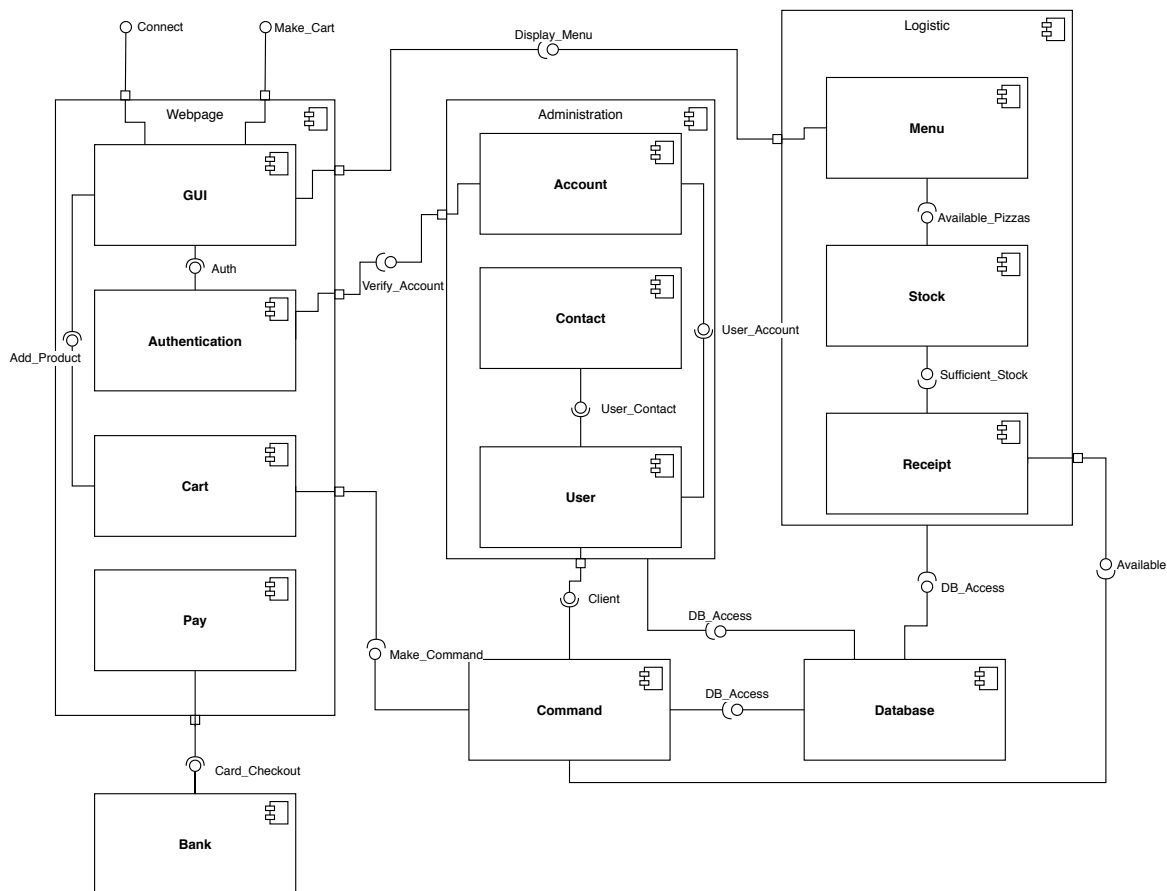
Une Address pourra être associée à un Contact. Un Contact pourra être associé à une Address.

## 4 - ARCHITECTURE TECHNIQUE

## 4.1 - Application Web

La pile logicielle est la suivante :

- Application: **Python 3.7**
- Serveur d'application: **Gunicorn 20.0.4**
- Server web: **NGINX 1.17.0**
- Server Base de données **MySQL 8.0.19**



#### 4.1.1 - Composants Webpage

Le composant Webpage regroupe des composants relatifs à l'interaction avec l'utilisateur

#### 4.1.1.1 - Composant GUI

Le composant GUI est responsable de l'affichage à l'utilisateur. Il permet par ailleurs :

- De se connecter via l'interface Connect.
- De réaliser un panier via l'interface Make\_Cart.



- D'afficher le menu via l'interface Display\_Menu

Il a besoin de Cart afin de pouvoir ajouter un produit au panier (interface Add\_Product) et d'Authentication afin de pouvoir se connecter (interface Auth).

#### **4.1.1.2 - Composant Authentication**

Le composant Authentication permet de se connecter à l'application. Il offre cette interface à GUI et a besoin de Account (interface Verify\_Account) afin de vérifier les credentials.

#### **4.1.1.3 - Composant Cart**

Le composant carte permet la gestion du panier il permet à GUI d'ajouter un produit. Il a besoin de l'interface MakeCommand afin d'être transféré en base de données.

#### **4.1.1.4 - Composant Pay**

Le composant Pay se sert de l'interface Card\_Checkout de Bank afin de réaliser un paiement en ligne.

### **4.1.2 - Composant Bank**

Le composant Bank permet la vérification d'un paiement par carte bancaire via son interface Card\_Checkout.

### **4.1.3 - Composants Administration**

Ce composant regroupe des composant relatifs à l'administration des informations. Il a besoin d'avoir accès à la base de données via l'interface DB\_Access.

#### **4.1.3.1 - Composant Account**

Le composant Account concerne la gestion des comptes utilisateurs. Il permet de vérifier si les credentials correspondent bien à un compte via l'interface Verify\_Account et permet de retrouver le compte d'un utilisateur via l'interface User\_Account.

#### **4.1.3.2 - Composant Contact**

Le composant Contact concernant les informations de contacts des personnes. Elle permet via son interface User\_Contact de récupérer le contact d'un User .

#### **4.1.3.3 - Composant User**

Le composant User concerne la gestion des utilisateurs. Il a besoin de l'interface User\_Account pour retrouver le compte d'un utilisateur et de l'interface User\_Contact afin de connaître les informations de contact d'un utilisateur. Il fournit une interface Client pour permettre d'associer un client à une commande.

### **4.1.4 - Composants Logistic**

Le composant Logistic regroupe des composants relatifs à la gestion de la logistique de la pizzeria. Il a besoin d'une accès à la base de données via l'interface DB\_Access.

#### **4.1.4.1 - Composant Menu**

Le composant Menu est responsable de l'affichage du menu via son interface Display\_Menu qu'elle fournit à GUI. Elle a besoin de l'interface Available\_Pizzas de Stock afin de connaître les pizzas disponible au menu.

#### **4.1.4.2 - Composant Stock**

Le composant Stock est responsable de la gestion du Stock, il fourni les interfaces Available\_Pizzas à Menu et Sufficent\_Stock à Receipt afin de s'assurer du stock lors de la prise/préparation de commande.

#### **4.1.4.3 - Composant Receipt**

Le composant Receipt est relatif aux recettes. Il a besoin de Sufficent\_Stock pour s'assurer que la recette est faisable en fonction du stock et il fournit une interface Available à Command pour s'assurer que chacun des éléments des recettes constitutives d'une commandes peuvent être réalisées.

#### **4.1.5 - Composant Database**

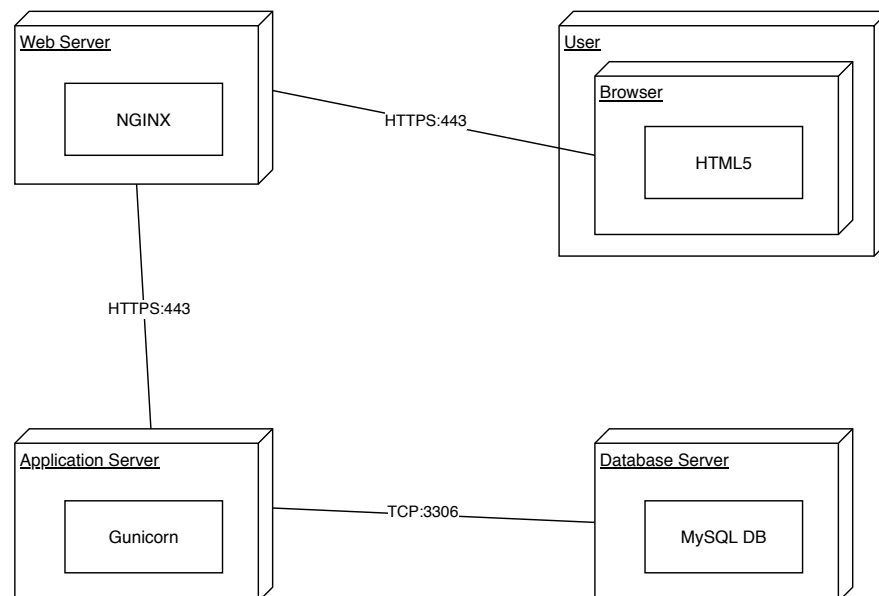
Le composant Database est responsable du stockage des données. Il offre une interface DB\_Access pour permettre l'accès à ses données.

#### **4.1.6 - Composant Command**

Le composant Command est responsable de la gestion des commandes. Il a besoin de l'interface Client afin de pouvoir récupérer les informations concernant un client, de l'interface Available afin de s'assurer qu'une commande est faisable et de l'interface DB\_Access pour l'accès aux données.

Il fourni quant à lui l'interface Make\_Command à Cart pour qu'une fois le panier constitué, il soit transformé en commande et inséré en base de données.

# 5 - ARCHITECTURE DE DÉPLOIEMENT



## 5.1 - Serveur de Base de données

Serveur hébergeant la base de données MySQL

Serveur Linux CentOS 7 + MySQL 8.0.19

Communication sur port par défaut MySQL : TCP 3306

## 5.2 - Serveur application

Serveur hébergeant l'application Python

Serveur Linux CentOS 7

Gunicorn 20.0.4

Python 3.7 + Django 3.0

## 5.3 - Serveur web

Serveur traitant et générant les requêtes HTTP pouvant les transformer en objet Python.

Serveur Linux CentOS 7

NGINX 1.17.0

## 6 - GLOSSAIRE

<b>Serveur d'application</b>	Serveur WSGI
<b>Serveur web</b>	Serveur HTTP