

# OC Pizza

## P6

### Dossier de conception technique

version 1

**Auteur**

Biedermann Claire  
*Développeur Web*

IT Consulting &  
Development

[www.it\\_consulting.com](http://www.it_consulting.com)

35 rue trust me i'm an engineer 75009 Paris – 01.45.09.39.45> – <it\_consulting@gmail.com>

S.A.R.L. au capital de 1 000,00 € enregistrée au RCS du Mont Fuji – SIREN 999 999 999 – Code APE :  
6202A

# TABLE DES MATIÈRES

|  |           |
|--|-----------|
| <b>1 - Versions.....</b>   | <b>3</b>  |
| <b>2 - Introduction.....</b>   | <b>4</b>  |
| 2.1 - Objet du document.....   | 4         |
| 2.2 - Besoin d'OC Pizza.....   | 4         |
| 2.2.1 - Le contexte.....   | 4         |
| 2.2.2 - Enjeux et objectifs.....   | 4         |
| <b>3 - Le domaine fonctionnel.....</b>   | <b>6</b>  |
| 3.1 - Référentiel.....   | 6         |
| 3.1.1 - Diagramme UML de classes.....  | 6         |
| 3.1.2 - Description textuelle entre les différentes relations entre les classes..... | 6         |
| 3.1.2.1 - Produit / Ingrédient.....  | 6         |
| 3.1.2.2 - Produit / Commande.....  | 7         |
| 3.1.2.3 - Produit / Prix.....  | 7         |
| 3.1.2.4 - Ingrédients / Pizzeria.....  | 7         |
| 3.1.2.5 - Pizzeria / Adresse.....  | 7         |
| 3.1.2.6 - Pizzeria / Commande.....   | 7         |
| 3.1.2.7 - Pizzeria / Utilisateur.....  | 7         |
| 3.1.2.8 - Adresse / Utilisateur.....   | 8         |
| 3.1.2.9 - Adresse / Commande.....  | 8         |
| 3.1.2.10 - Utilisateur / Commande.....   | 8         |
| 3.1.2.11 - Utilisateur / Rôle.....   | 8         |
| <b>4 - Architecture Technique.....</b>   | <b>9</b>  |
| 4.1 - Modèle Physique de donnée.....   | 9         |
| 4.2 - Application Web.....   | 9         |
| 4.2.1 - Le diagramme de composant.....   | 9         |
| 4.2.2 - Composants Authentification.....   | 10        |
| 4.2.2.1 - Inscription.....   | 10        |
| 4.2.2.2 - Authentification.....  | 10        |
| 4.2.2.3 - CompteUtilisateur.....   | 11        |
| 4.2.3 - Composants Gestion.....  | 11        |
| 4.2.3.1 - Manager.....   | 11        |
| 4.2.3.2 - Chiffre d'affaire.....   | 11        |
| 4.2.3.3 - Stock.....   | 12        |
| 4.2.4 - Composants Commande.....   | 12        |
| 4.2.4.1 - Commande.....  | 12        |
| 4.2.4.2 - Paiement.....  | 12        |
| 4.2.4.3 - Livraison.....   | 12        |
| 4.2.4.4 - Recette.....   | 12        |
| <b>5 - Architecture de Déploiement.....</b>  | <b>14</b> |
| 5.1 - Diagramme de déploiement.....  | 14        |
| 5.2 - Serveur Physique.....  | 14        |
| 5.3 - Serveur de base de donnée.....   | 14        |
| 5.4 - Framework applicatif.....  | 14        |
| 5.5 - Le serveur d'application.....  | 14        |
| 5.6 - Le serveur web.....  | 15        |
| 5.7 - Le système de paiement.....  | 15        |
| <b>6 - Glossaire.....</b>  | <b>16</b> |

# 1 - VERSIONS

| Auteur            | Date       | Description          | Version |
|-------------------|------------|----------------------|---------|
| Claire Biedermann | 22/11/2021 | Création du document | 1       |
|                   |            |                      |         |

## 2 - INTRODUCTION

### 2.1 - Objet du document

Le présent document constitue le dossier de conception fonctionnel du projet OC pizza.

Il sert à définir le domaine fonctionnel et à concevoir l'architecture technique de la solution répondant aux besoins du client et contiendra :

- une modélisation des objets du domaine fonctionnel
- une identification des différents éléments composant le système à mettre en place ainsi que leur interaction
- une description du déploiement des différents composants envisagés
- un schéma de la base de donnée

### 2.2 - Besoin d'OC Pizza

#### 2.2.1 - Le 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 cinq points de vente et prévoit d'en ouvrir au moins trois de plus d'ici six mois.

#### 2.2.2 - Enjeux et objectifs

##### **Enjeux:**

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.

##### **Objectifs:**

- 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 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.
- proposer un aide-mémoire aux pizzaïolos indiquant la recette de chaque pizza
  - permettre au responsable de suivre ce qui se passe dans les points de vente

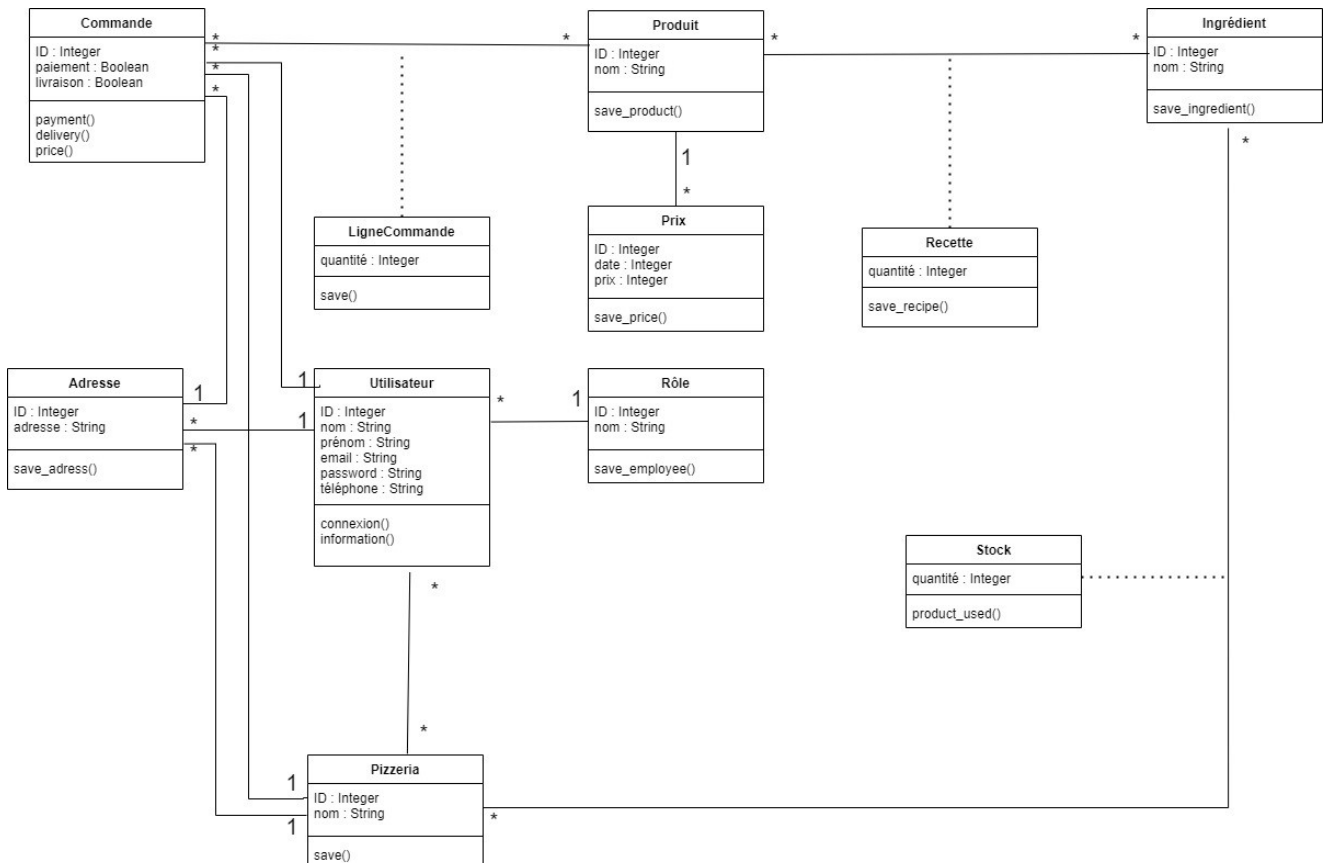
# 3 - LE DOMAINE FONCTIONNEL

## 3.1 - Référentiel

### 3.1.1 - Diagramme UML de classes

Ce diagramme permet de présenter les classes et les interfaces du système ainsi que leurs relations afin d'exprimer les besoins du système.

La section supérieure contient le nom de la classe, la section intermédiaire contient les attributs de la classe et la section inférieure contient les méthodes de la classe (opérations).



### 3.1.2 - Description textuelle entre les différentes relations entre les classes

#### 3.1.2.1 - Produit / Ingrédient

Un ou plusieurs produits peuvent utiliser un ou plusieurs ingrédients.

Un ou plusieurs ingrédients peuvent être utilisés dans un ou plusieurs produits.

La classe association Recette permet:

- d'associer un ingrédient à un produit, en indiquant la quantité

*Cette classe permet d'associer plusieurs ingrédients à un produit et un produit à plusieurs ingrédients.*

#### **3.1.2.2 - Produit / Commande**

Un ou plusieurs produits peuvent se retrouver dans une ou plusieurs commandes.

Une ou plusieurs commandes peuvent contenir un ou plusieurs produits.

La classe association LigneCommande permet :

- d'associer un produit à une commande en indiquant la quantité

*Cette classe permet d'associer plusieurs produits à une commande et plusieurs commandes à un produit.*

#### **3.1.2.3 - Produit / Prix**

Un produit peut avoir un ou plusieurs prix, mais pas en même temps.

Un prix ne peut être associé qu'à un produit.

#### **3.1.2.4 - Ingrédients / Pizzeria**

Un ou plusieurs ingrédients peuvent se retrouver dans une ou plusieurs pizzeria.

Une ou plusieurs pizzeria peuvent avoir un ou plusieurs ingrédients.

La table association Stock permet :

- d'associer une pizzeria à un ingrédient en indiquant la quantité

*Cette classe permet d'associer plusieurs ingrédients à une pizzeria et plusieurs pizzeria à un ingrédient.*

#### **3.1.2.5 - Pizzeria / Adresse**

Une adresse ne peut être rattachée qu'à une pizzeria.

Une pizzeria peut avoir une ou plusieurs adresses.

#### **3.1.2.6 - Pizzeria / Commande**

Une commande ne peut être associée qu'à une pizzeria.

Une pizzeria peut avoir une ou plusieurs commandes.

#### **3.1.2.7 - Pizzeria / Utilisateur**

Une ou plusieurs pizzeria peuvent être associées à un ou plusieurs utilisateurs.

Un ou plusieurs utilisateurs peuvent être rattachés à une ou plusieurs pizzeria.

#### **3.1.2.8 - Adresse / Utilisateur**

Une adresse ne peut être associée qu'à un utilisateur.

Un utilisateur peut avoir une ou plusieurs adresses.

#### **3.1.2.9 - Adresse / Commande**

Une adresse peut avoir une ou plusieurs commandes.

Une commande ne peut avoir qu'une adresse.

#### **3.1.2.10 - Utilisateur / Commande**

Une commande ne peut avoir qu'un utilisateur.

Un utilisateur peut avoir une ou plusieurs commandes.

#### **3.1.2.11 - Utilisateur / Rôle**

Un utilisateur ne peut avoir qu'un rôle.

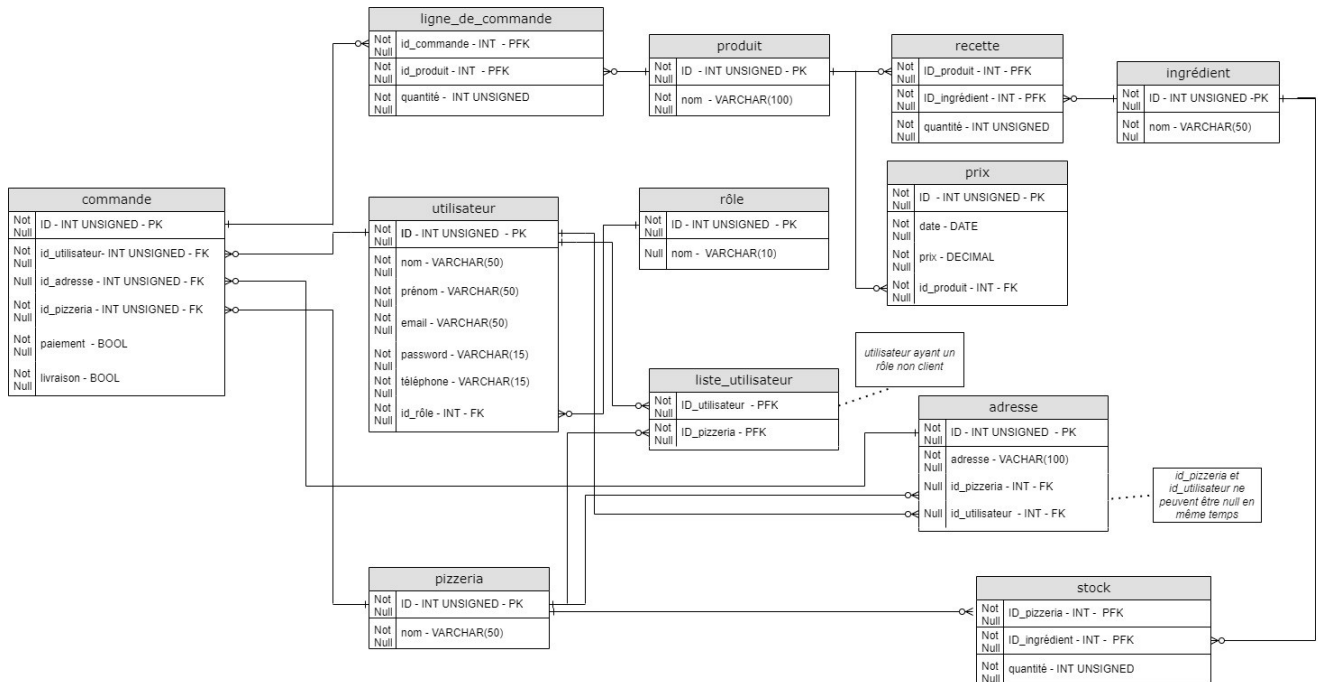
Un rôle peut avoir un ou plusieurs utilisateurs.



# 4 - ARCHITECTURE TECHNIQUE

## 4.1 - Modèle Physique de donnée

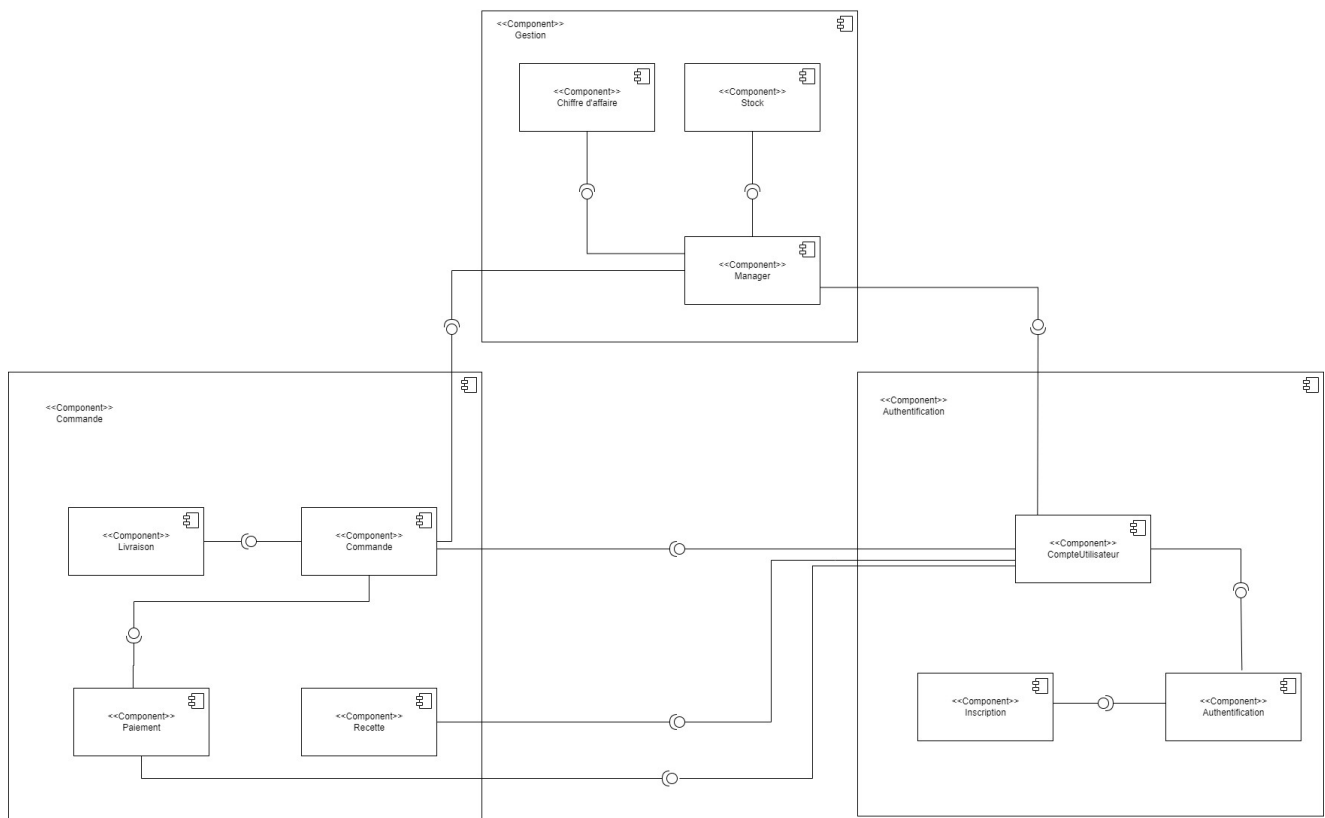
Ce diagramme décrit les différentes tables de la base de donnée ainsi que leur relation.



## 4.2 - Application Web

### 4.2.1 - Le diagramme de composant

Le diagramme UML de composant, permet de représenter les différentes fonctionnalités du système et leur interaction.



## 4.2.2 - Composant Authentification

### 4.2.2.1 - Inscription

Descriptif : Permet à l'utilisateur de s'inscrire sur l'application

Interface fournie :

- Le statut d'inscrit

### 4.2.2.2 - Authentification

Descriptif : Permet de se connecter et de se déconnecter de l'application

Interface fournie :

- Le statut d'authenticé

Interface requise :

- Le statut d'inscrit

### 4.2.2.3 - CompteUtilisateur

Descriptif : Permet à l'utilisateur d'accéder à son compte, modifier ces renseignements et utiliser les fonctionnalités de l'application en fonction des droits d'accès régit par le rôle de l'utilisateur

Interface fournie :

- L'identité de l'utilisateur pour le composant « commande »
- L'identité de l'utilisateur pour le composant « paiement »

- Le droit d'accès pour le composant « manager » aux utilisateurs de type « manager »
- Le droit d'accès pour le composant « recette » aux utilisateurs de type « pizzaïolo »

Interface requise :

- Le statut d'Authentifié

## **4.2.3 - Composant Gestion**

### **4.2.3.1 - Manager**

Descriptif : Ce composant est réservé au rôle utilisateur de type « manager » et permet :

- de consulter, modifier et gérer les ingrédients dans le composant « stock »
- de modifier les accès et compte utilisateur du composant « CompteUtilisateur »
- suivi des commandes en cours et leur statut du composant « commande »

Interface fournie :

- les modifications, ajouts ou suppressions dans le composant « stock »
- les modifications ou suppressions dans le composant « compte utilisateur »
- information du composant « commande »
- information du composant « chiffre d'affaire »

Interface requise :

- Accès par le composant « CompteUtilisateur »

### **4.2.3.2 - Chiffre d'affaire**

Descriptif : Permet la gestion financière de la pizzeria à travers le récapitulatif des commandes passées et leur montant

Interface fournie :

- information relative au chiffre d'affaire de chaque pizzeria

### **4.2.3.3 - Stock**

Descriptif : Permet de modifier et/ou vérifier les produits disponibles

Interface fournie :

- information relative à l'état des stocks de chaque pizzeria
- modification uniquement par le composant « manager »

## 4.2.4 - Composant Commande

### 4.2.4.1 - Commande

Descriptif : Lien entre l'utilisateur, les produits disponibles, le paiement des produits en ligne et la livraison

Interface fournie :

- les produits disponibles choisis par le composant « utilisateur »
- le statut du paiement si paiement en ligne
- le statut de la livraison si livraison commandée

Interface requis :

- L'identité utilisateur par le composant « CompteUtilisateur »

Particularité :

Est dépendant du composant « stock »

### 4.2.4.2 - Paiement

Descriptif : Lien entre le système bancaire et l'utilisateur afin d'effectuer le paiement en ligne des produits choisis et valider le statut de la transaction.

Interface fournie :

- Statut de la transaction si paiement en ligne

Interface requise :

- L'identité utilisateur par le composant « CompteUtilisateur » qui effectue le paiement

### 4.2.4.3 - Livraison

Descriptif : Permet de traiter les commandes en livraison, statut et informations.

Interface fournie :

- Statut de l'avancée de la livraison

### 4.2.4.4 - Recette

Descriptif : Ce composant n'est accessible que par les utilisateurs de type « pizzaïolo » et permet l'accès aux différentes recettes des pizzas avec les ingrédients correspondants

Interface fournie :

- Information relative aux différentes recettes de pizza

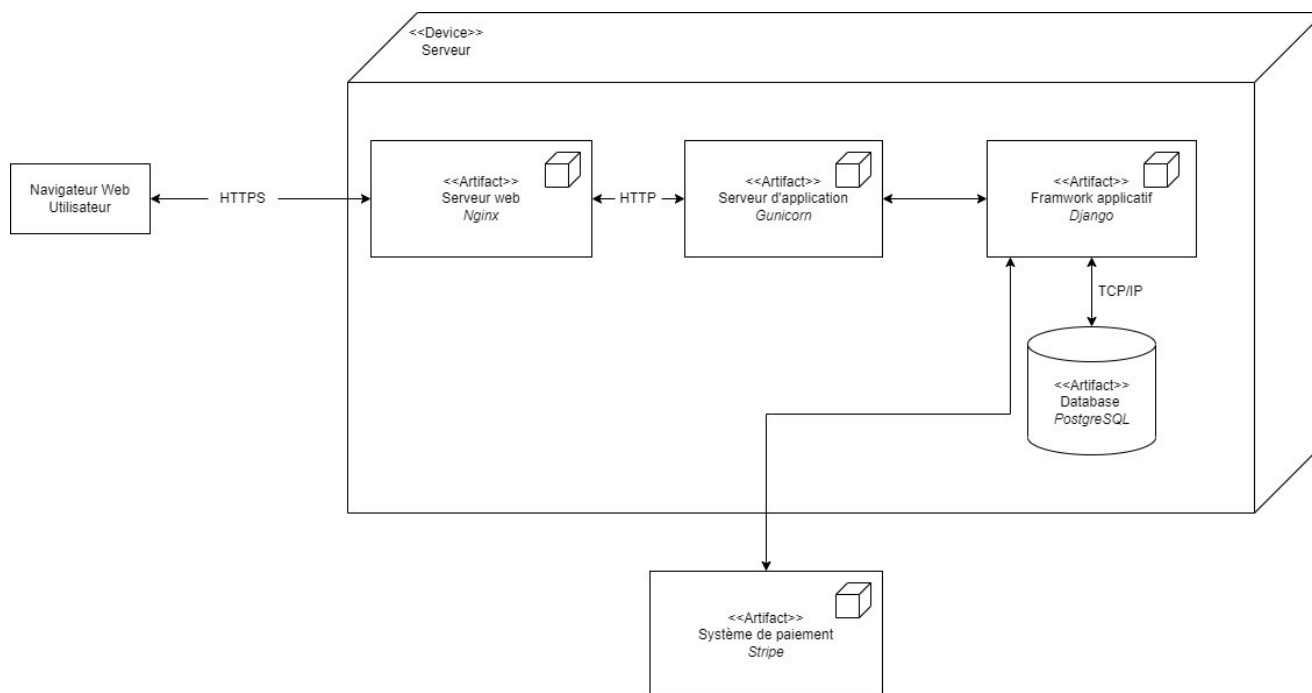
Interface requise :

- Le droit d'accès utilisateur par le composant « CompteUtilisateur »

# 5 - ARCHITECTURE DE DÉPLOIEMENT

## 5.1 - Diagramme de déploiement

Diagramme UML de déploiement permet de 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.



## 5.2 - Serveur Physique

L'application sera déployé sur un serveur physique unique.

## 5.3 - Serveur de base de données

Un serveur de base de données sert à stocker, à extraire et à gérer les données dans une base de données. Il permet également de gérer la mise à jour des données. Il donne un accès simultané à cette base à plusieurs serveurs Web et utilisateurs.

Le système de gestion de base de donnée relationnelle proposée est PostgreSQL (version 14.1.0).

## 5.4 - Framework applicatif

Le Framework donne un cadre permettant un développement web simplifié et rapide.

Le Framework proposé est Django (version 3.2.9) utilisant Python (version 3.10.0).

## **5.5 - Le serveur d'application**

Le serveur d'application permet le dialogue entre le serveur Web et le Framework. Il permet de traduire les informations entre les deux, car ils ne dialoguent pas dans le même langage.

Le serveur d'application proposé est Gunicorn (version 20.1.0)

## **5.6 - Le serveur web**

Un serveur web permet de recevoir et renvoyer des requêtes HTTP. C'est le pivot entre les demandes utilisateurs et le serveur applicatif.

Le serveur web proposé est Nginx (version 1.20.0)

## **5.7 - Le système de paiement**

Le système de paiement permet de transférer de l'argent du compte bancaire d'un client vers le compte de l'entreprise au moyen d'une transaction par carte de crédit.

Le système proposé est Stripe.

## 6 - GLOSSAIRE

|                        |   |
|------------------------|---|
| <b>Application</b>     | Programme permettant de réaliser une ou plusieurs tâches  |
| <b>Artifact</b>        | Serveur logiciel  |
| <b>Base de donnée</b>  | Ensemble de donnée structurée et organisée représentant un système d'information                                |
| <b>Classe (python)</b> | Ensemble d'attribut et de méthode permettant de créer des objets  |
| <b>Devise</b>          | Serveur physique  |
| <b>Diagramme</b>       | Représentation graphique d'un ensemble d'évènement  |
| <b>Programme</b>       | Succession d'instruction rédigée dans un langage de programmation compréhensible par le système de l'ordinateur |