

# SPECIFICATION FONCTIONNELLE



Répo Github : [https://github.com/SergueiNK/oc\\_projet\\_p6](https://github.com/SergueiNK/oc_projet_p6)

## TABLE DES MATIERES

<b>1 - Versions .....</b>	<b>3</b>
<b>2 - Votre SOCIETE : contexte et besoin .....</b>	<b>4</b>
2.1 - Rappel de contexte et besoin .....	4
<b>3- description du Domaine fonctionnel .....</b>	<b>5</b>
3.1 Diagramme de classe .....	5
3.2 Composants du système .....	9
3.3 Diagramme de déploiement.....	11

# 1 - VERSIONS

Auteur	Date	Description	Version
Serguei	11 MAI 2021	Dossier final	V2.0.0

## 2 - VOTRE SOCIETE : CONTEXTE ET BESOIN

### 2.1 - Rappel de contexte et besoin

Vous êtes un jeune groupe de pizzeria en plein essor. 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.

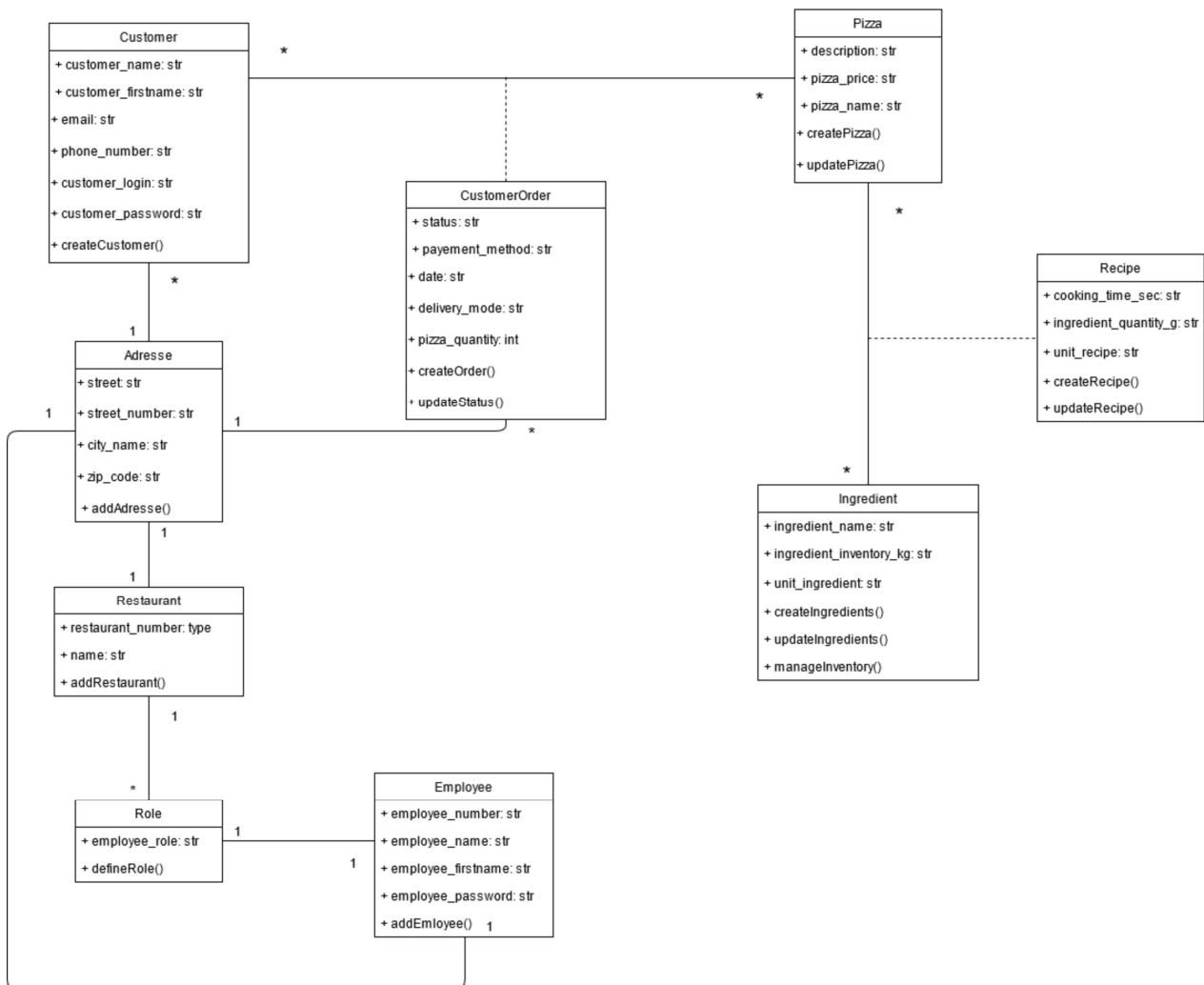
Le système informatique actuel ne correspond plus aux besoins du groupe, car il ne permet pas une gestion centralisée de toutes les pizzerias.

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

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

## 3– DESCRIPTION DU DOMAINE FONCTIONNEL

### 3.1 Diagramme de classe



Le descriptif de diagramme de classe ci-dessous permet de mieux comprendre les interactions entre de différentes classes.

### **Customer**

Représente la classe client avec ses informations personnelles nécessaires pour la création de compte

Relation Customer – Pizza :

Un client peut avoir une ou plusieurs pizzas.

Relation Customer – Adresse :

Un client peut avoir une ou plusieurs adresses.

### **Customer Order**

Représente la classe commande avec toutes les informations nécessaires à la création au suivie d'une commande.

### **Address**

Représente la classe adresse avec toutes les informations nécessaires pour la sauvegarde d'informations sur une adresse.

Relation Adresse – Customer :

Une adresse peut avoir un ou plusieurs clients.

Relation Adresse – Customer Ordre

Une adresse peut avoir une ou plusieurs commandes.

Relation Adresse – Restaurant

Une adresse peut avoir un seul restaurant.

Relation Adresse – Employé

Une adresse peut avoir un seul employé.

## **Restaurant**

Représente la classe restaurant avec le nom et les numéros des restaurants de la franchise.

### Relation Restaurant – Role

Un restaurant peut avoir un ou plusieurs roles.

## **Role**

Représente la classe rôle avec l'attribution pour chaque employé du rôle qu'il occupe dans la pizzeria exemple : livreur, staff ou le directeur.

### Relation Role – Restaurant

Un rôle peut avoir un seul restaurant.

### Relation Role – Employé

Un rôle peut avoir un seul employé.

## **Employee**

Représente la classe employée avec les informations nécessaires pour un compte employé.

### Relation Employee – Role :

Un employé peut avoir un et un seul rôle.

### Relation Employee – Adresse :

Un employé avoir un et une seule adresse.

### **Pizza**

Représente la classe pizza avec les informations nécessaires pour la vente.

#### Relation Pizza – Customer :

Une pizza peut avoir un ou plusieurs clients.

#### Relation Pizza – Ingredient :

Une pizza peut avoir plusieurs Ingrédients.

### **Recipe**

Représente la classe recette avec les différentes recettes pour une pizza.

### **Ingredients**

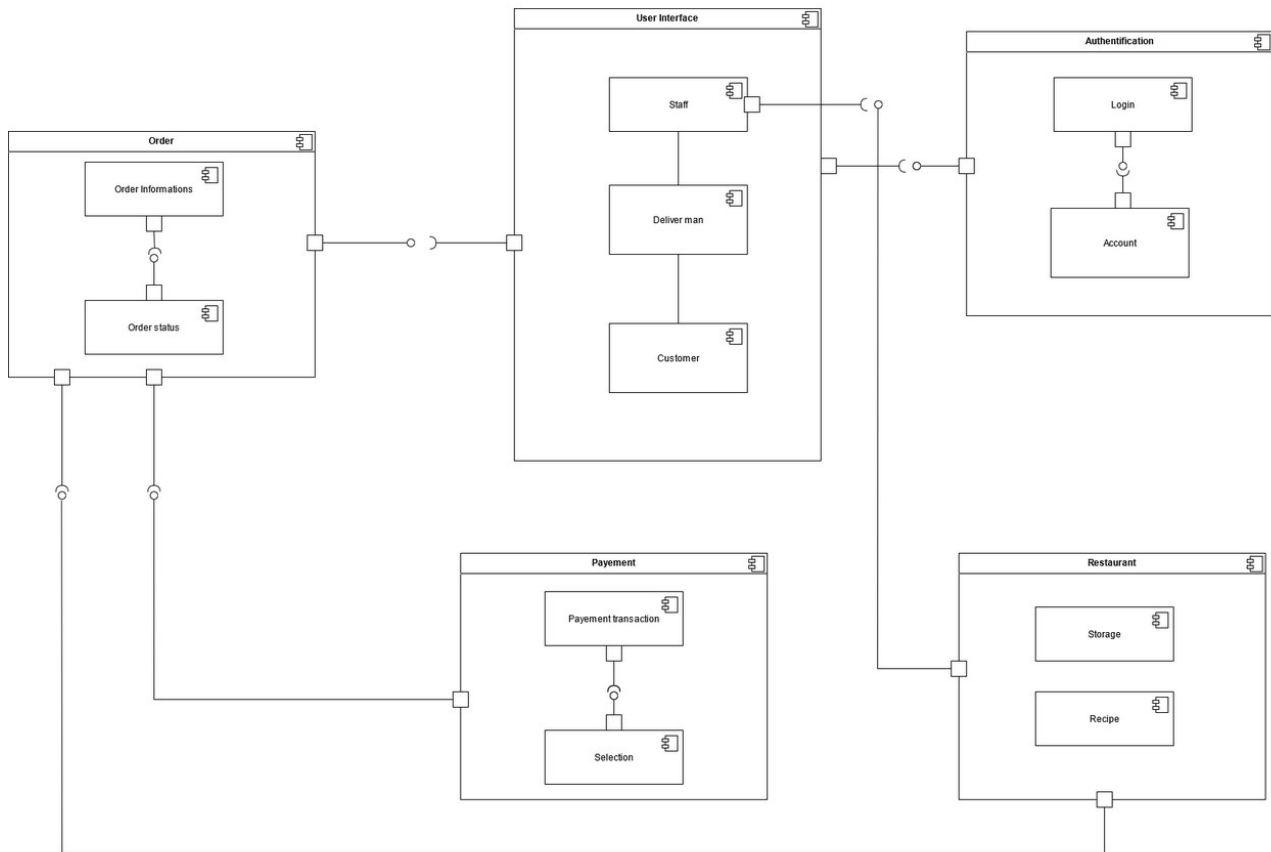
Représente la classe ingrédients avec leur description et le stock.

#### Relation Ingredients – Pizza :

Un ingrédient peut avoir une ou plusieurs pizzas.



## 3.2 Composants du système



### Login

Composant pour l'authentification d'un utilisateur, que ce soit pour le personnel ou les clients.

### Account

Composant pour la gestion des informations d'un utilisateur tels que nom, prénom, adresse, mot de passe.

### Staff

Composant pour l'affichage de la surface utilisateur du personnel d'OCpizza.

### Deliver man

Composant pour l'affichage de la surface utilisateur des livreurs OCPizza.

### Customer

Composant pour l'affichage de la surface utilisateur des clients.

### **Order Informations**

Composant permettant de rassembler et d'afficher toutes les informations concernant une commande tels que la date, montants, produits.

### **Order status**

Composant permettant de modifier et d'afficher l'état d'une commande.

### **Payement transaction**

Composant permettant de réaliser le payement d'une commande que ce soit en ligne, en guichet ou en livraison.

### **Sélection**

Composant permettant de sélectionner les produits et réaliser un panier pour la commande.

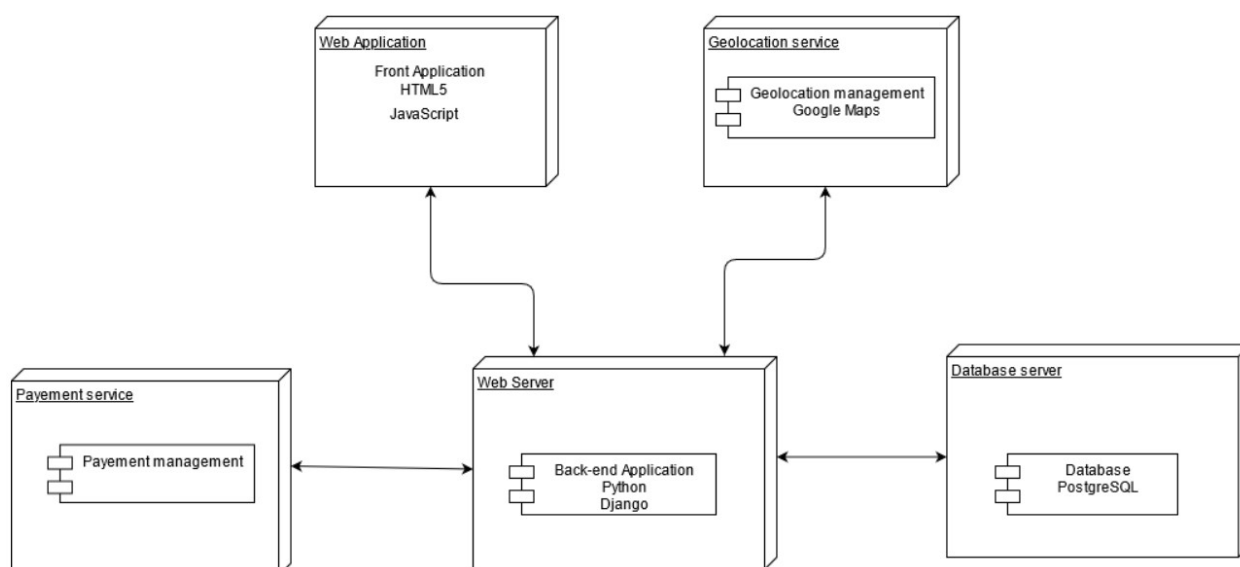
### **Storage**

Composant permettant de réaliser la gestion de stock des ingrédients.

### **Recipe**

Composant permettant de réaliser et d'afficher les recettes.

### 3.3 Diagramme de déploiement



Le déploiement de la solution OCPizza se fera sur les serveurs contenant les particularités suivantes :

- Un serveur dédié pour la gestion de la base de données en utilisant le système de gestion PostgreSQL
- Un deuxième serveur dédié s'en occupera de la gestion de l'application web destiné aux clients d'OCPizza. Cette application web sera créée et alimentée grâce aux technologies HTML5 et JavaScript. Ce même serveur hébergera également l'application destinée aux employés d'OCPizza. Cette application sera développée grâce au langage Python et Framework Django.
- Le service de paiement communiquera avec le serveur de l'application interne d'OCPizza, grâce à WSGI.