OC Pizza

Concevez la solution technique d’un système de gestion de pizzeria

Dossier de conception technique

Version 0.1

Auteur
Mourgues Benjamin
Analyste-programmeur

RestoPro
Restopro.fr

115 Rue Godard, 33200 Bordeaux – +33762887200 – contact@restopro.fr
S.A.R.L. au capital de 1 000,00 € enregistrée au RCS de Bordeaux – SIREN 999 999 999 – Code APE : 6202A

Table des matières

1 - Versions

Auteur	Date	Description	Version
Benjamin M	05/11/2021	Création du document	0.1
Benjamin M	12/11/2021	Modification diagramme déploiement	1.0
Benjamin M	15/01/2022	Mise à jour des diagrammes	1.1

2 - Introduction

2.1 - Objet du document

Le présent document constitue le dossier de conception technique de l'application OC Pizza.

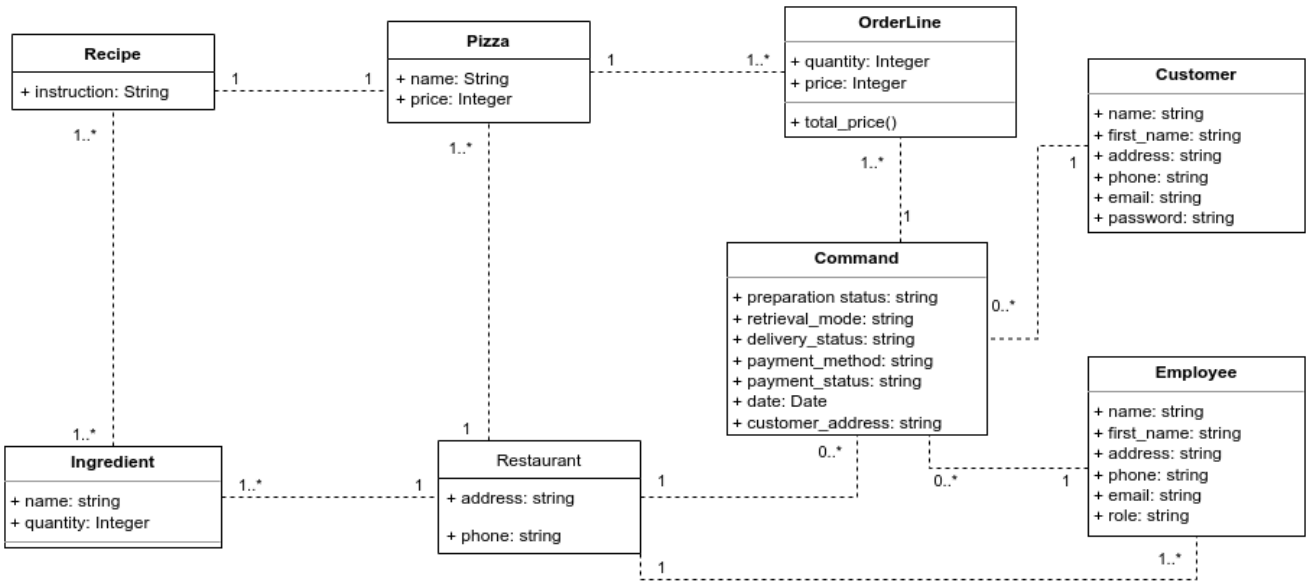
L’objectif du document est de regrouper, résumer et expliquer les différents éléments du dossier de conception technique.

Les éléments du présents dossiers découlent :

- des demandes formulées par le client, OC Pizza
- des considérations techniques jugées pertinentes par RestoPro.

3 - Le domaine fonctionnel

3.1 - Diagramme de classe UML



3.1.1 - Règles de gestion

Chacune des relations étant à deux sens (la relation Restaurant Employee est la même que Employee Restaurant), les relations seront toutes décrites en se positionnant par rapport aux classes suivantes : Command, Restaurant, Pizza et ingrédient.

Ainsi l'ensemble des relations de Customer est décrite dans son unique relation, Command.

Relations de "Command" :

- Avec OrderLine
  - Chaque instance de Orderline fait partie d'une Commande
  - Chaque Commande a au moins une OrderLine.
- Avec Customer
  - Chaque Command est rattachée à un Customer
  - Chaque Customer peut avoir 0 ou plusieurs Commands
- Avec Employee
  - Chaque Command peut avoir 0 ou 1 Employee
  - Chaque Employee peut être associé à 0 ou plus Commands
- Avec Restaurant
  - Chaque Command est reliée à un unique Restaurant
  - Un Restaurant peut avoir 0 ou plusieurs Commands

Relations de “Restaurant” :

- Avec Pizza :
  - Chaque Restaurant doit avoir une Pizza ou plus
  - Chaque Pizza est associée à un restaurant.
- Avec Ingrédient :
  - Chaque Restaurant dispose d’au moins 1 Ingredient
  - Chaque Ingrédient est relié à 1 restaurant.
- Avec Employee :
  - Chaque Employee travaille dans 1 Restaurant
  - Chaque Restaurant peut avoir 1 ou plus Employee

Relations de “Pizza”

- Avec Recipe :
  - Chaque Pizza a une seule Recipe
  - Chaque Recipe ne correspond qu’à une seule Pizza
- Avec OrderLine
  - Chaque Pizza peut faire partie d’une ou plusieurs Orderlines
  - Chaque OrderLine ne contient qu’un seul type de Pizza (la quantitée est située dans Orderline).

Relation de “Ingredient” :

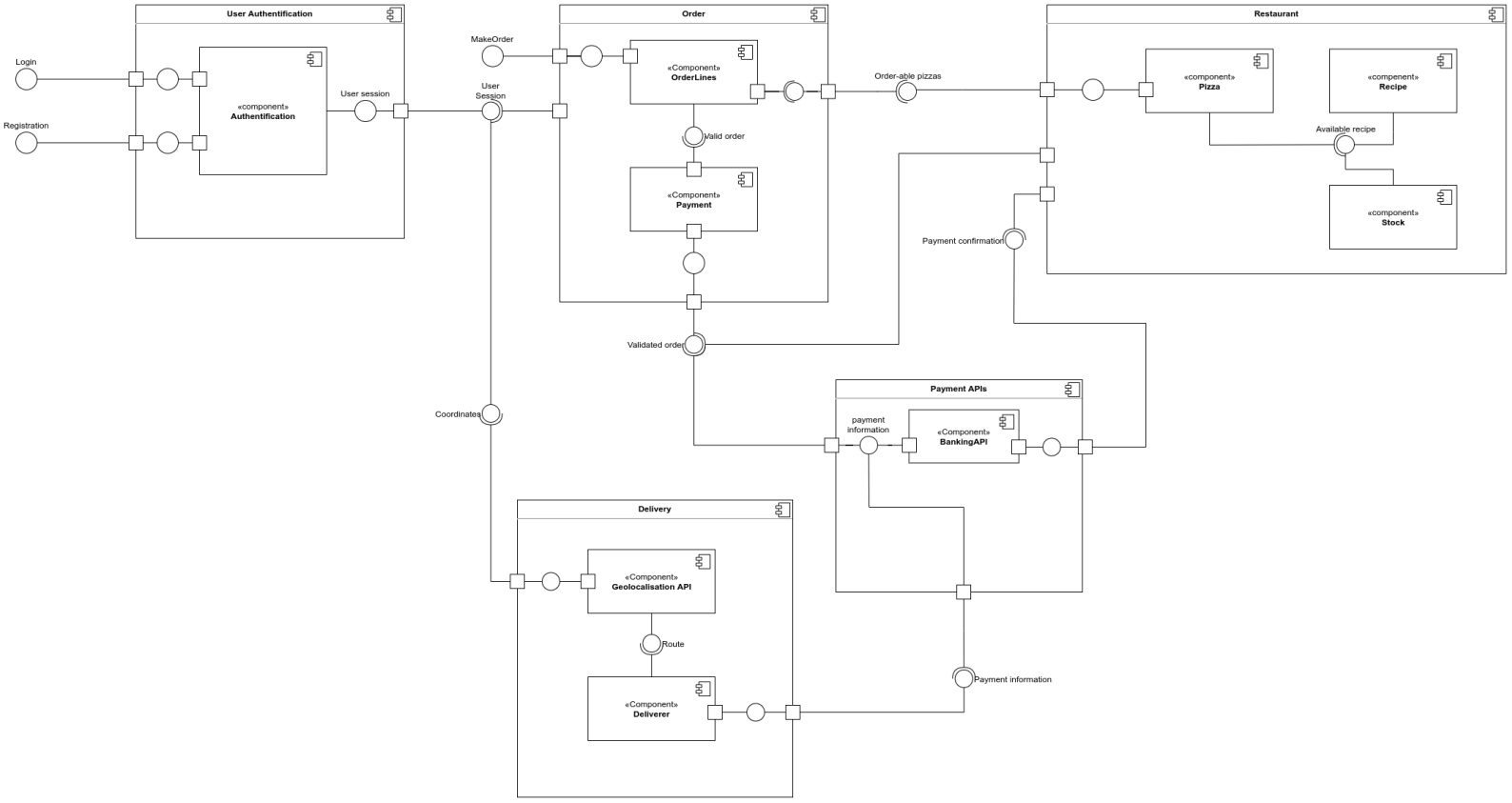
- Avec Recipe:
  - Chaque Ingrédient peut être dans 1 ou plusieurs Recipe
  - Chaque Recipe peut avoir 1 Ingredient ou plus.

4 - Architecture Technique

4.1 - Application Web

La pile logicielle est la suivante :

- Application **Python version 3.8**
- Serveur d'application **Gunicorn 20.1.0**
- Serveur Web **NGINX 1.21.3**
- Base de données **PostgreSQL 14.2**



4.1.1 - User authentification

Ce composant gère la création d'une session utilisateur, que celle ci soit avec un login ou non à l'aide d'une session "invité".

En résulte une session utilisateur qui sera associée aux prochaines actions sur le site, notamment le processus de commande.

4.1.2 - Order

Le composant Order matérialise les intéractions nécessaires pour mener à bien une commande.

Pour faire une commande, il faut dans un premier temps choisir au moins une pizza, la liste des pizzas disponibles sera récupérée depuis le composant “Restaurant”.

Une fois le contenu de la commande validé par le client, une autre étape cruciale est celle du paiement, selon le mode de paiement choisi par le client, la commande sera transmise au composant “Payment APIs” ou directement au composant “Restaurant”.

4.1.3 - Restaurant

Le composant restaurant va fournir le composant “Order” la liste des pizzas commandables ainsi que leurs informations sur la base du stock d’ingrédients disponibles.

Ce composant recevra d’autre part les informations concernant le paiement des commandes si le client décide de payer en ligne.

4.1.4 - Delivery

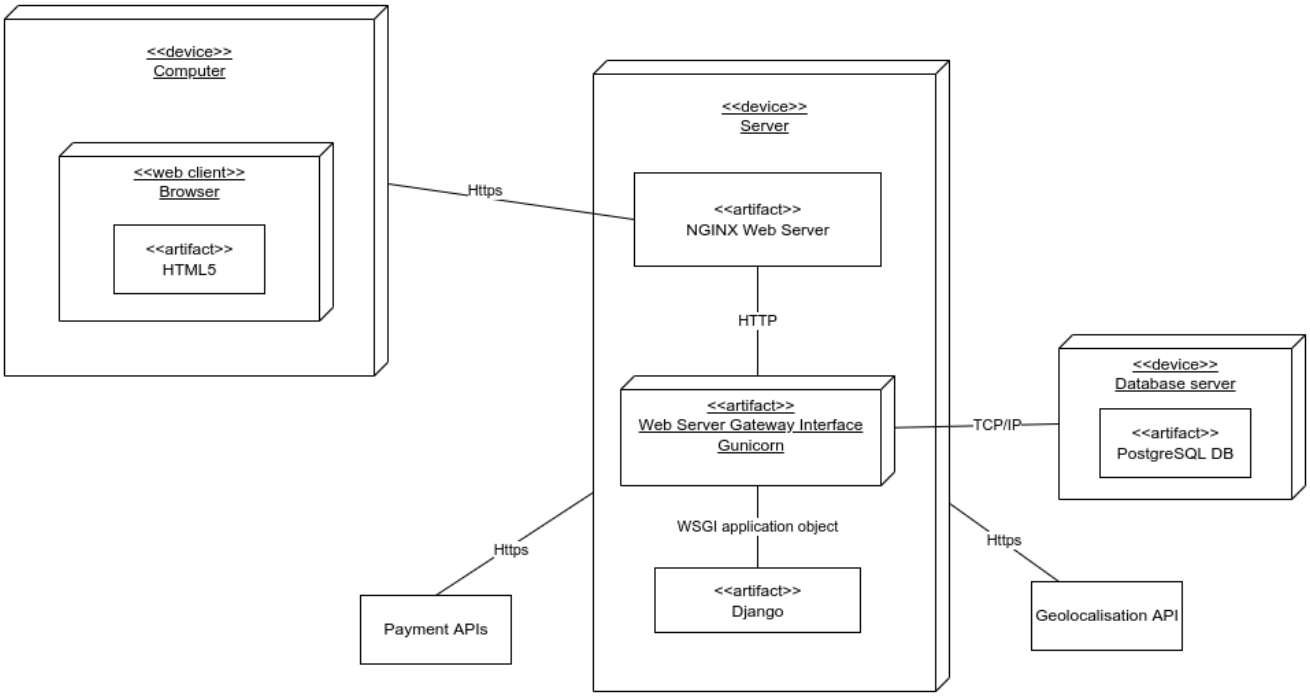
Ce composant utilisera les informations récupérées dans la session utilisateur les coordonnées du client à livrer afin d’alimenter l’API de géolocalisation.

Le livreur, si le client décide de payer à réception est chargé de transmettre les informations de paiement par le biais d’un terminal de paiement portatif ou simplement en renseignant le montant en espèce dans l’application au composant “Payment APIs”.

4.1.5 - Payment APIs

Ce composant prendra en entrée les informations de paiement des clients et enverra une confirmation (ou non) du paiement du client au composant “Restaurant”.

5 - Architecture de Déploiement



5.1 - Serveur de Base de données

Un serveur Ubuntu Server 20.04 Focal Fossa avec une base de donnée PostgreSQL 14.2, c'est cette base de donnée qu'interrogera Django via son ORM.

5.2 - Serveur application

C'est un serveur WSGI ("*Web Server Gateway Interface* ") Gunicorn 20.1.0, il fait l'interface entre Django et le serveur Web.

5.3 - Serveur web

C'est un serveur Web NGINX version 1.21.3, il reçoit les requêtes des navigateurs avant de les transmettre au serveur d'application.

5.4 - Navigateur

Le navigateur pourra être n'importe quel logiciel grand public permettant le rendu de réponses HTML (Chrome, Firefox, Safari, Edge ...)

6 - Glossaire

API	Application Programming Interface
WSGI	Web Server Gateway Interface