|  |
| --- |
| **Une image contenant dessin, signe  Description générée automatiquement**  **Projet 6**  Dossier de conception technique  Version 2.0 |
| **Gaëtan GROND**  Gaëtan  *Developpeur* |

Table des matières

1 -Versions 3

2 -Introduction 4

2.1 -Objet du document 4

2.2 -Références 4

3 -Architecture Technique 5

3.1 -Composants généraux 5

3.1.1 -Package A 5

3.1.1.1 -Composant X 5

3.1.1.2 -Composant Y 5

3.1.2 -Package B 5

# Versions

|  |  |  |  |
| --- | --- | --- | --- |
| Auteur | Date | Description | Version |
| Gaetan | 20 Nov. 20 | Création du document | V2.0.0 |
| Gaetan | 22 Nov. 20 | Correction de fautes | V2.0.1 |
| Gaetan | 25 Nov. 20 | Relations de classe | V2.1.0 |
| Gaetan | 4 Dec. 20 | Finalisation dossier | V2.2.0 |

# Introduction

## Objet du document

Le présent document constitue le dossier de conception fonctionnelle de l'application OC PIZZA.

Il fait suite à un échange par téléphone avec les commanditaires du projet Franck et Lola gérants de la pizzeria OC Pizza.

Les commanditaires du projet Franck et Lola gérants de la pizzeria OC Pizza ont exprimés les besoins suivants :

* Moderniser leur système informatique
* Suivre le cycle d'une commande
  + 1. *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. :

* 5 points de vente
* 3 ouvertures supplémentaire à prévoir
* Une présence en ligne déjà établie

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

* + 1. *Enjeux et Objectifs*

L’objectif du projet est de permettre à OC Pizza de moderniser son système informatique et de proposer des nouvelles fonctionnalités pour améliorer l'expérience utilisateur, le suivi des commandes pour les livreurs et les clients ainsi que l'ajout de recettes pour l'équipe :

# 3– description du Domaine fonctionnel

## Diagramme de classe

**Relations entre les classes :**

Le but des descriptions ci-après est d'expliquer les liaisons entre les différentes classes du diagramme. Il est important de noter que pour une uniformisation avec le modèle physique de données, les noms des classes ainsi que leurs attributs sont en anglais.



Relations entre classes :

**User – Rôle :**

* « User » représente un utilisateur inscrit sur le site
* « Rôle » représente le rôle d’un utilisateur (admin, client, livreur…)

Un utilisateur est associé à un rôle et un rôle est occupé par un ou plusieurs utilisateurs

**User – Order :**

* « Order » représente une commande faite par un user

Un utilisateur peut avoir aucune ou plusieurs commandes et une commande peut avoir un utilisateur

**User – Address :**

* « Address » Adresse physique d’un restaurant ou un utilisateur

Un utilisateur peut avoir une seule adresse et une adresse un seul utilisateur

**Address – Restaurant :**

* « Restaurant » lieu physique de vente

Un « restaurant » peut avoir une seule « « adresse »" et une « adresse » un seul « restaurant »

**Order- OrderStatus :**

* « Order » Commande d’un client et sa description
* « OrderStatus » statut de la commande (en cours, terminée etc…)

Une « order » peut avoir un « OrderStatus » et un « OrderStatus » aucune ou plusieurs « order »

**Order – Product :**

* « Product » produit commandé

Une « order » peut avoir aucun (si annulé) ou plusieurs « products » et un « product » une où. Plusieurs « order »

**Order – Address :**

Une « order » peut avoir une « address » et une « address » aucune ou plusieurs « order »

**Order – Restaurant :**

Une « order » peut avoir un « restaurant » et un « restaurant » aucune ou plusieurs « order »

**Order – PaymentMethod:**

* “PaymentMethod”, le paiement choisi (chèque, CB, PayPal)

Une « order » peut avoir une seule « PaymentMethod » et une « PaymentMethod peut avoir aucun ou plusieurs « order »

**Order – DeliveryMode :**

* “DeliveryMode” le mode de livraison choisi

Une « order » peut avoir un seul « DelivreryMode » et un « DelivreryMode peut avoir aucune ou plusieurs « order »

**Restaurant – Reviews :**

* « Reviews » Un commentaire laissé par un utilisateur sur un restaurant

Un « restaurant » peut avoir aucune ou plusieurs « reviews » et une « reviews » peut avoir aucune ou un « restaurant »

**Restaurant – Product :**

Un « restaurant peut avoir aucun ou plusieurs « products » et un « product » peut avoir un ou plusieurs « restaurants »

**Product – Category :**

* Category : Catégorie du produit selon son type

Un « product » peut avoir une seule « category » et une « category » peut avoir aucun ou plusieurs « product »

## Composants du système



Le diagramme ci-dessus décrit les composant du système ainsi que les composants externes utilisés.

**Login**

Composant pour l'authentification des utilisateurs (tous rôles)

**Account**

Composant pour gérer le compte de l’utilisateur, modifier l’adresse, son email, son mot de passe

**Customer**

Composant qui est affiché uniquement pour l’interface d’un client de la pizzeria

**Staff**

Composant qui est affiché uniquement pour l’interface d’un employé interne à OC PIZZA

**Delivrer**

Composant qui est affiché uniquement pour l’interface d’un livreur de la pizzeria ou d’un service tiers (uber, delivroo)

**Shopping Cart**

Composant qui est utilisé pour la gestion du panier utilisateur et l’affichage des produits ainsi que le récapitulatif du tarif et promotions

**Interfaces requises :** Payment

**Payment**

Composant pour gérer les paiements en ligne qui est lié à une API externe (Stripe)

**Stock**

Composant qui affiche en temps réel le stock des produits d’une pizzeria spécifique

**Recette**

Composant qui affiche les recettes pour le staff de OC Pizza

## Diagramme de déploiement



Le solution OC Pizza est déployé sur un serveur physique contenant les composants et technologies suivantes :

* Les utilisateur « UserClient » et « StaffClient » interagiront avec un serveur HTTP utilisant la technologie **NGINX**
* L'application utilisera la Framework **Django** et le langage **Python** et sera déployé sur un serveur **WSGI**
* Le troisième serveur sera la base de données relationnelles utilisant **Postgres** **SQL** qui communiquera avec le serveur **WSGI** et le **serveur http**
* Les services « **Payment** **services** » et « **Point of Sale** » communiqueront avec le serveur WSGI

## Diagramme de déploiement - Descriptions

Le « http server » sera hébergé Unbuntu 20.04 en utilisant les services d’hébergement mutualisé DigitalOcean

Le « wsgi server » sera hébergé Unbuntu 20.04 en utilisant les services d’hébergement mutualisé DigitalOcean

La version de Python utilisé pour le serveur d’application est la 3.8, la version de Django est la 3.1.4.

La base de données utilise le moteur PostgreSQL version 13