



Projet 6

Dossier de conception technique

Version 2.0

Gaëtan GROND
Gaëtan
Développeur



TABLE DES MATIERES

| | |
|---------------------------------------|----------|
| 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 |



1 - 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 |

2 - INTRODUCTION

2.1 - 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

2.1.1 - 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

2.1.2 - Enjeux et Objectifs

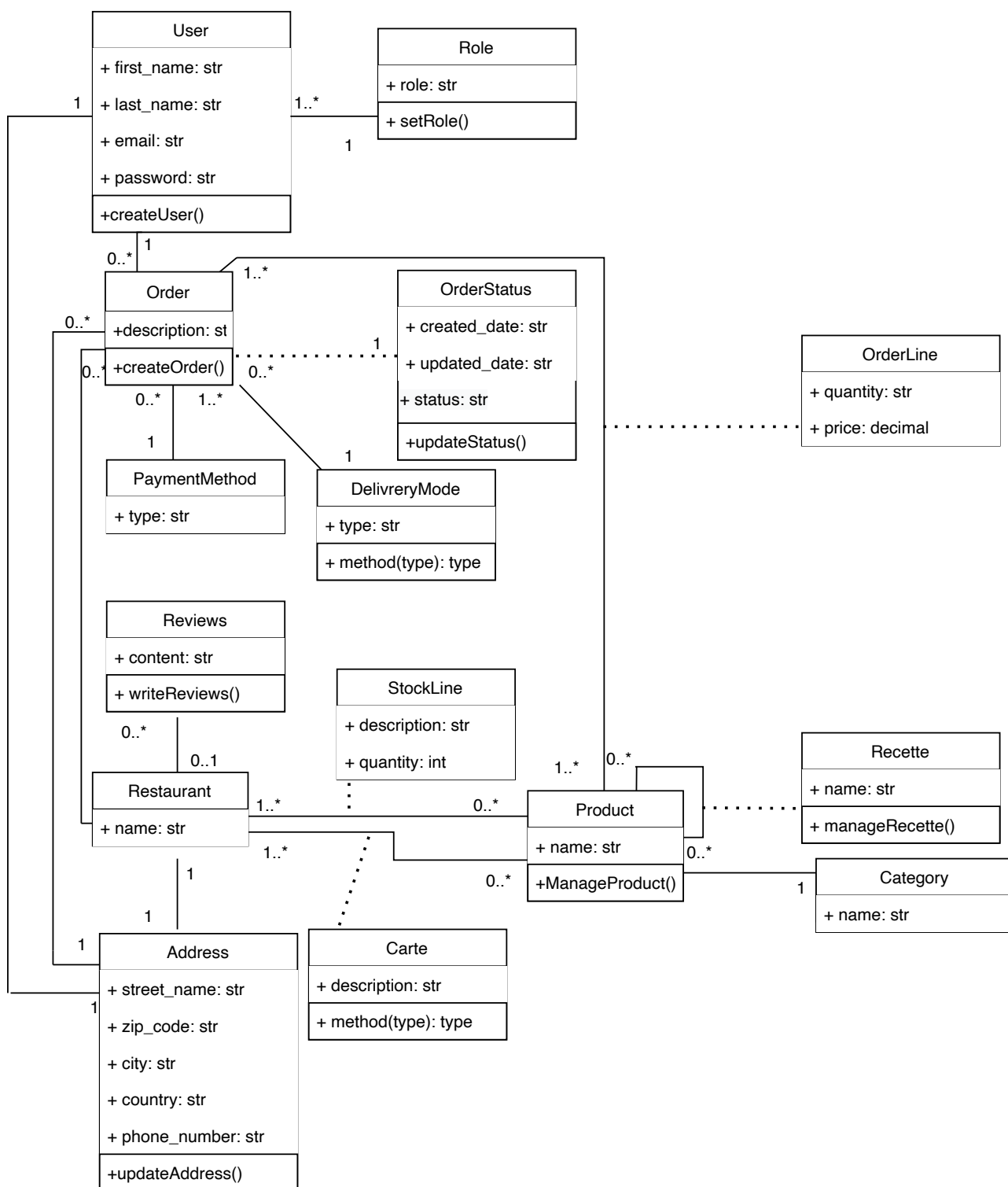
L'objectif du projet est de permettre à OC Pizza de moderniser son système informatique et de proposer des nouvelles fonctionnalités pour améliorer l'expérience utilisateur, le suivi des commandes pour les livreurs et les clients ainsi que l'ajout de recettes pour l'équipe :

3– DESCRIPTION DU DOMAINE FONCTIONNEL

3.1 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 ou. 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 « DeliveryMode » et un « DeliveryMode » 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 :

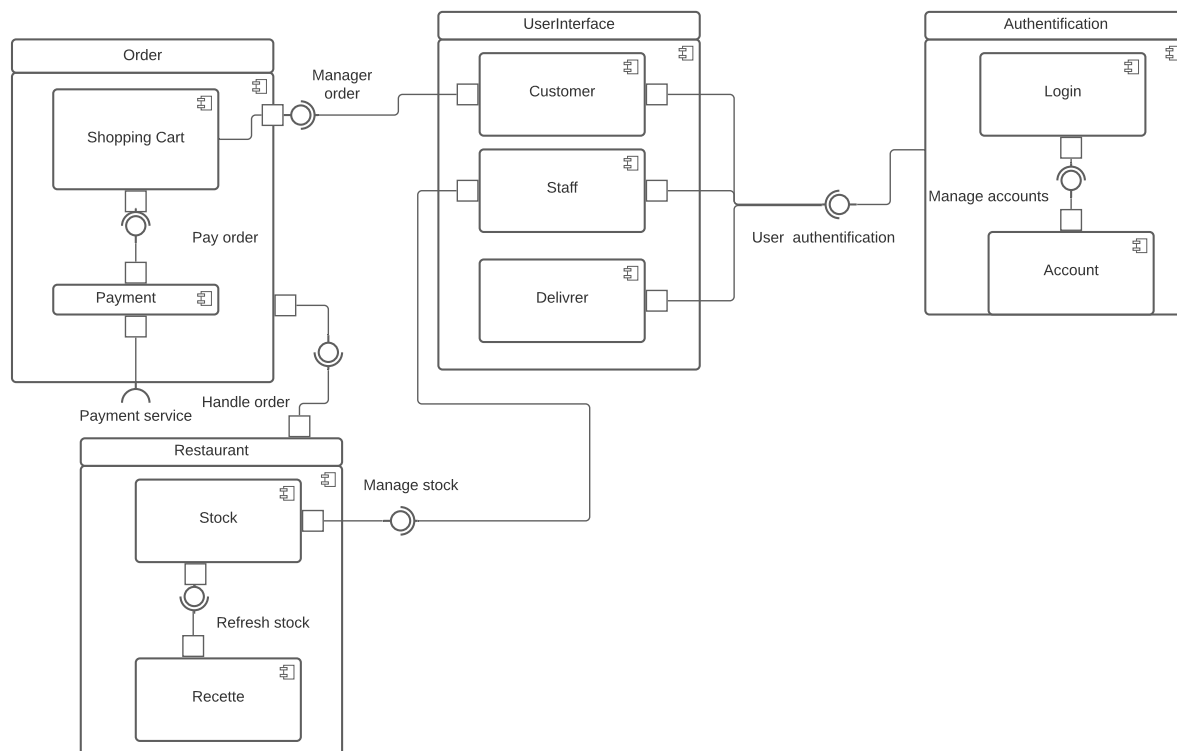
Un « restaurant » peut avoir aucun ou plusieurs « products » et un « product » peut avoir un ou plusieurs « restaurants »

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 »

3.2 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)

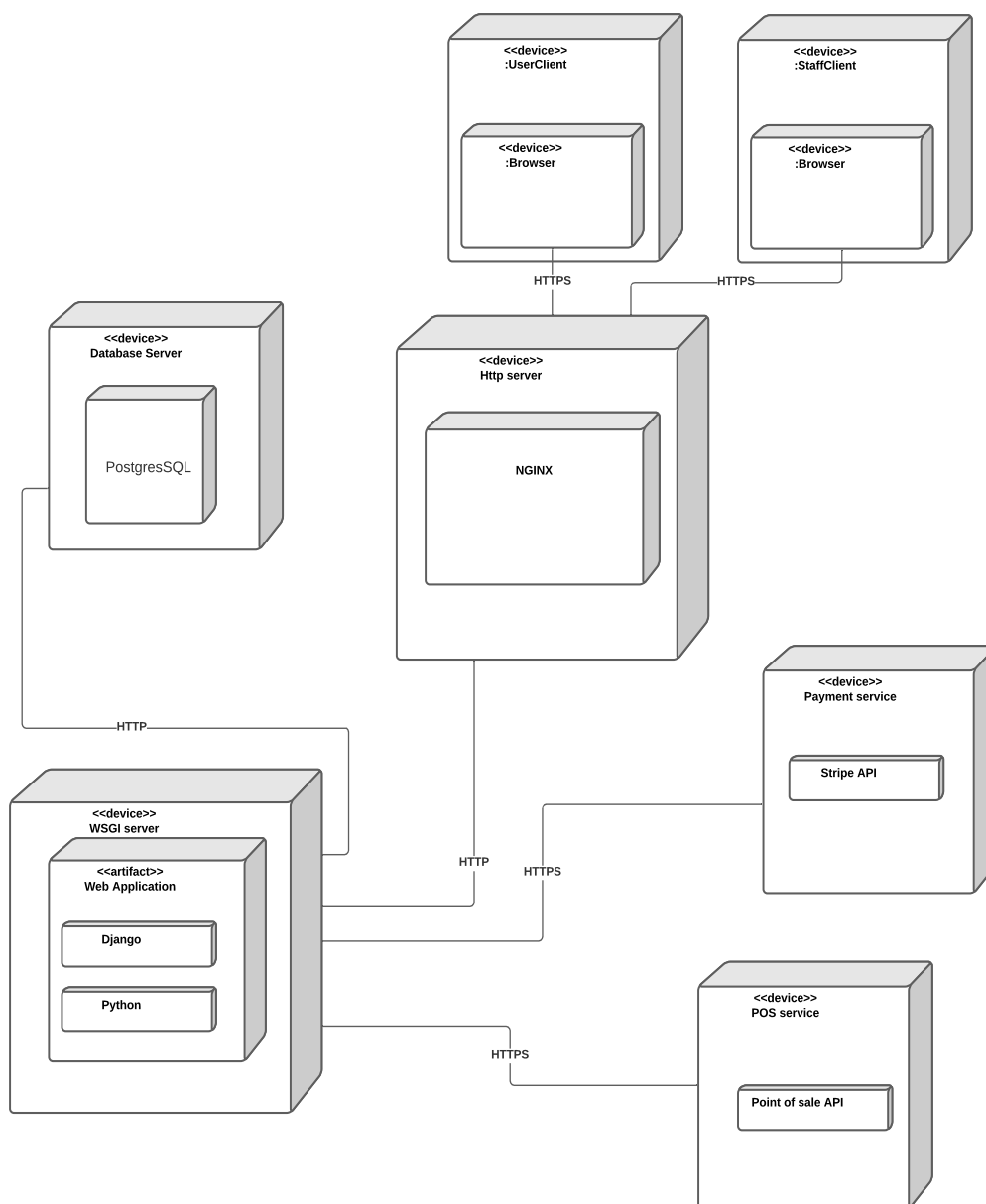
Stock

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

3.3 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

3.4 Diagramme de déploiement - Descriptions

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

Le « wsgi server » sera hébergé Ubuntu 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

3 - ARCHITECTURE LOGICIELLE

3.1 - Principes généraux

Les sources et versions du projet sont gérées par **Git** et **Github** pour l'hébergement, les dépendances sont gérées par l'outil PIPENV.

L'architecture applicative est la suivante :

- Une couche **model** : implémentation du modèle des objets métiers
- Une couche **vue** : Responsable de recevoir une requête HTTP et d'y répondre de manière intelligible par le navigateur
- Une couche **template** : Responsable de l'affichage des variables

3.1.1 - Les modules

Les différents modules représentés dans l'application OC pizza sont les suivantes :

- Back_office
- Front_office
- Users
- Order

Ces modules représentent des applications Django

3.1.2 - Structure des sources

La structuration des répertoires du projet suit la logique suivante :

- └─ README.md
- └─ config
- | └─ settings
- | | └─ base.py
- | └─ urls.py
- └─ manage.py
- └─ ocpizza
- | └─ back_office
- | | └─ admin.py



- | | | — apps.py
- | | | — models.py
- | | | — urls.py
- | | | — views.py
- | | — front_office
- | | | — admin.py (...)
- | | — order
- | | | — admin.py (...)
- | | — templates
- | | — users
- | | — admin.py (...)
- | — requirements.txt...

3.2 - Environnement de développement

L'environnement de développement utilise les outils suivants :

- VSCode Version: 1.52.1
- MacOS 11.1 Big Sur

4 - POINTS PARTICULIERS

4.1 - Gestion des logs

La gestion des logs et du monitoring sont gérés par deux services distincts :



- Digital Ocean Agent : Gère le monitoring de l'application
- Sentry : Gère les logs

4.2 - Fichiers de configuration

4.2.1 - *Application web*

Vous pouvez trouver ci-dessous les différents fichiers de configuration nécessaire au bon fonctionnement et la configuration de OC Pizza :

```
1 language: python
2 python:
3   - "3.7"
4   - "3.8" # Current default Python version for the project
5   - "3.9"
6
7 addons:
8   # Adding Chrome for functional testing
9   chrome: stable
10
11 env: DJANGO_SETTINGS_MODULE="config.settings.local"
12
13 # Update PIP
14 before_install:
15   - python -m pip install --upgrade pip
16
17 before_script:
18   - install-webdrivers --path webdrivers
19
20 # Install dependencies from local requirements
21 install:
22   - pip install -r requirements/local.txt
23
24 services:
25   - postgresql
26
27 script:
28   - pytest
29   - python manage.py test
30   - black .
31   - isort .
32
```


__init__.py
base.py
local.py
production.py
test.py
base.txt
local.txt
production.txt

4.3 - Ressources

Vous pouvez retrouver les documentations officielles des services et technologies utilisées ci-dessous :

- Django : <https://www.djangoproject.com>
- Gunicorn : <https://gunicorn.org>

IT CONSULTING

www.itconsulting.fr

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



- Nginx : <https://www.nginx.com>
- Stripe : <https://stripe.com/fr>
- Python : <https://www.python.org>

4.4 - Serveur web NGINX

Permet que chaque requête soit traitée avec un processus dédié, pour une meilleure rapidité de traitement.

Conçu spécialement pour un trafic élevé, ce qui optimise le site et ouvre des possibilités.

NGINX est le serveur le plus utilisé au monde.

4.5 - Serveur gunicorn (WSGI)

Gunicorn (WSGI) sert à assurer une compatibilité avec les applications web et une bonne communication entre celui-ci et le serveur web afin de générer les pages à afficher à l'utilisateur.

4.6 - Serveur postgres SQL

Ce serveur nous permet de stocker l'ensemble des données nécessaire au bon fonctionnement de l'application

On peut l'illustrer grâce à ce modèle de donnée.

4.7 - Procédure de packaging / livraison

Github gère les différents paquets de l'application OC PIZZA

7 - GLOSSAIRE

| | |
|------------------------|---|
| Framework | En programmation informatique, un Framework désigne un ensemble cohérent de composants logiciels structurels, qui sert à créer les fondations ainsi que les grandes lignes de tout ou d'une partie d'un logiciel |
| Python | Python est un langage de programmation interprété. Il favorise la programmation structurée |
| Django | Django est un cadre de développement web open source en Python. Il a pour but de rendre le développement web 2.0 simple et rapide |
| Application web | En informatique, une application web est une application manipulable directement en ligne grâce à un navigateur web et qui ne nécessite donc pas d'installation sur les machines clientes, contrairement aux applications mobiles. |