

IT Consulting & Development

OC Pizza

Conception technique d'un système de gestion de groupe
de pizzerias

Dossier de conception technique

Version 0.2

Table des matières

Versions	1
Introduction	2
Objet du document	2
Rappel du contexte	2
Rappel des enjeux et objectifs	2
Architecture technique	3
Composants généraux	3
Application web	4
Architecture de déploiement	5
Le serveur web (Web Server):	5
Le serveur de base de donnée (Database Server):	5
Architecture logicielle	6
Diagramme de classe	6
Explications du diagramme de classe	7
Structure des sources	7
Modèle physique de données	8
Points particuliers	9
Configuration	9
Environnement de développement	9
Procédure de livraison	9

Versions

Auteur	Date	Description	Version
Sebastien Goliot	14/01/2021	Création du document	0.1
Sebastien Goliot	17/06/2021	Modification du serveur de base de donnée utilisé	0.2

Introduction

Objet du document

Le présent document constitue le dossier de conception technique du système de gestion des pizzerias du groupe OC Pizza.

Ce document a pour objectif de présenter les spécifications techniques du projet.

Rappel du contexte

“OC Pizza” est un jeune groupe de pizzerias spécialisé dans la vente de pizzas livrées ou à emporter, comprenant déjà 5 points de vente et prévoyant l’ouverture de 3 pizzerias supplémentaires d’ici 6 mois.

Leur système informatique ne répondant plus à leurs besoins, le groupe aimerait être plus efficace dans la gestion des commandes et bénéficier d’un suivi de chaque commande ainsi que des stocks.

La solution proposée devra être livrée pour l’ouverture des 3 nouvelles pizzerias.

Rappel des enjeux et objectifs

La solution développée devra répondre aux besoins cités par le client.

Ainsi, plusieurs points ont été évoqués :

1. Être plus efficace dans la gestion des commandes, de leur réception à leur livraison en passant par leur préparation
2. Bénéficier d’un suivi en temps réel des commandes passées, en préparation et en livraison
3. Suivre en temps réel l’état des stocks d’ingrédients afin de savoir quelles pizzas peuvent être réalisées
4. Proposer un site internet afin que les clients puissent :
 - 4.1. Passer leurs commandes en ligne, en plus de la prise de commande classique en pizzeria ou par téléphone
 - 4.2. Payer leur commande en ligne s’ils le souhaitent
 - 4.3. Modifier ou annuler leur commande tant que celle-ci n’a pas été préparée
5. Proposer un aide-mémoire aux pizzaiolos en indiquant la recette de chaque pizza à préparer

La solution proposée devra non seulement être en adéquation avec les besoins exprimés mais aussi prendre en compte le fait que le groupe est composé de plusieurs établissements, avec un nombre grandissant.

La situation géographique des clients sera aussi un caractère à prendre en compte, déterminant si le client peut être desservi et par quelle pizzeria la commande sera prise en charge.

Architecture technique

Composants généraux

Ce diagramme représente les principaux composants du système.

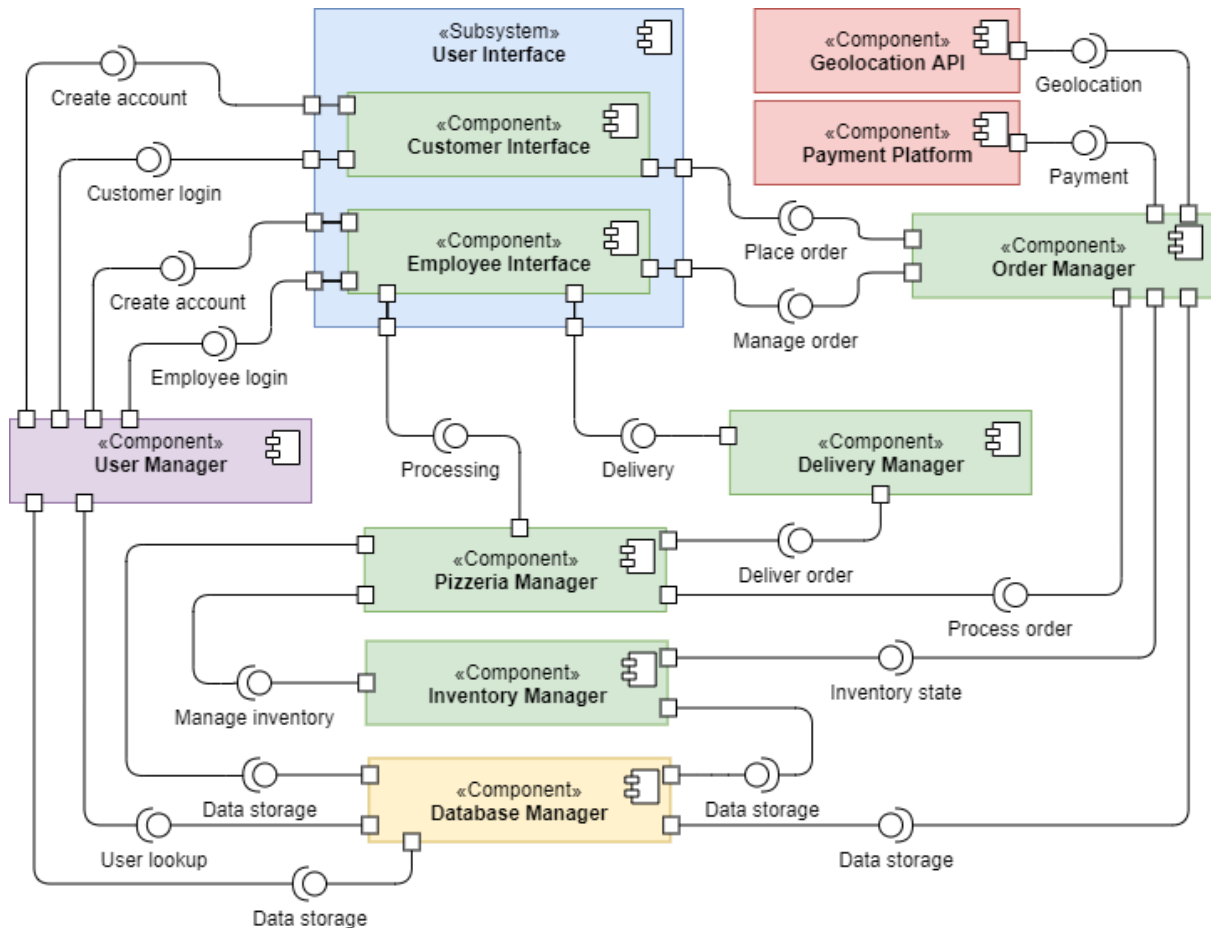


fig. 01

On retrouve ici les composants suivants :

- Customer / employee interface
 - Ces composants constituent l'interface utilisateur pour les clients et les employés
- Order manager
 - Ce composant gère les éléments suivants :
 - La prise de commande
 - L'attribution des commandes aux pizzerias
 - La disponibilité des pizzas commandées
 - La mise à jour de l'état des commandes
 - Le paiement des commandes
 - La gestion des commandes

- Delivery manager
 - Ce composant permet la prise en charge des commandes à livrer, il fournit toutes les informations nécessaires au bon déroulement des livraisons.
- Pizzeria manager
 - Ce composant permet aux pizzerias de traiter les commandes reçues:
 - L'attribution des commandes aux employés.
 - La mise à jour automatique de l'inventaire.
 - La gestion des pizzerias:
 - La gestion comptable des pizzerias.
 - La gestion des employés.
 - La gestion des stocks (inventaire).
- Inventory manager
 - Ce composant expose l'interface de gestion de l'inventaire des pizzerias.
 - Les employés ayant les permissions requises peuvent interagir avec l'inventaire via le composant "Pizzeria manager".
- User Manager
 - Ce composant permet la gestion des utilisateurs:
 - Il permet aux clients de créer un compte et de se connecter.
 - Il permet aux employés de se connecter.
 - Il permet aux employés ayant les permissions requises de créer d'autres comptes d'employés.
- Database manager
 - Ce composant expose une interface abstraite de gestion de base de données
 - Il permet l'enregistrement et l'édition en base de donnée des éléments suivants:
 - Les comptes utilisateur
 - Les stocks
 - Les commandes
 - Les données de gestion de pizzeria
- Composants externes :
 - Geolocation API
 - L'API de géolocalisation utilisée pour récupérer les positions des clients et des pizzerias
 - Payment platform
 - La plateforme utilisée par l'application pour récupérer le paiements des commandes

Application web

Le système sera déployé sous la forme d'une application web accessible via internet depuis n'importe quel navigateur web.

La pile logicielle est la suivante :

- Django >= 3.1
- Python >= 3.8

Architecture de déploiement

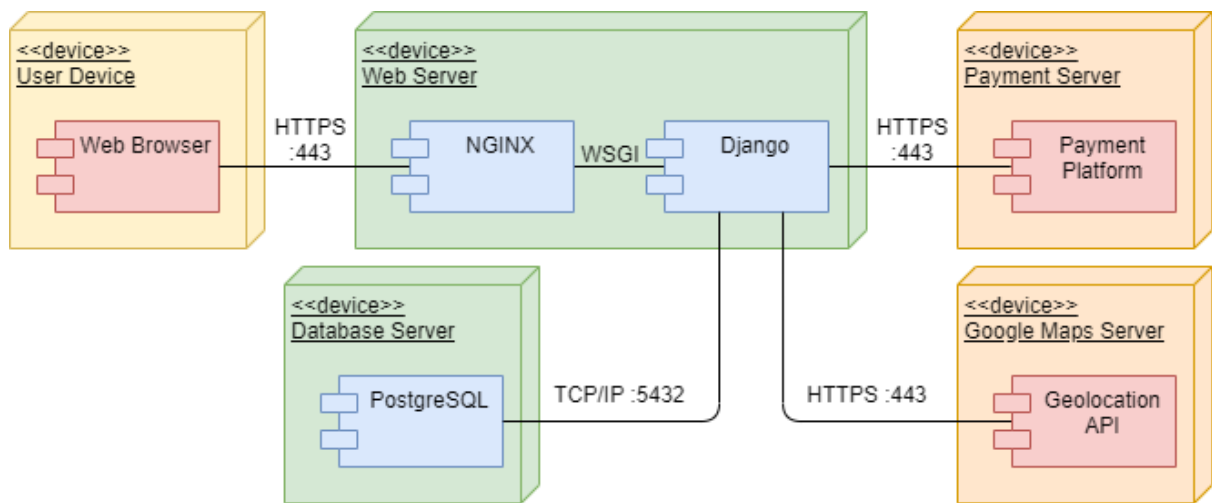


fig. 02

Le serveur web (Web Server):

Ce serveur est responsable du déploiement en production de l'application web
Sa pile logicielle est la suivante:

- Ubuntu 20.04 (LTS) x64
- NGINX >=1.18
 - Configuré en reverse-proxy vers Django via WSGI.
- Django >=3.1
- Python >= 3.8

Le serveur de base de donnée (Database Server):

Ce serveur est responsable de la gestion de la base de données
Ce dernier est managé par DigitalOcean.

- PostgreSQL 13

Le diagramme de classes représente la structure du système en modélisant ses classes, leurs attributs et leurs relations.



Explications du diagramme de classe

On retrouve ici 4 groupes, rassemblés par couleurs:

- **Utilisateur**
 - Ces classes représentent les utilisateurs, les clients et les employés avec leurs rôles.
- **Adresse**
 - Ces classes représentent une adresse, qu'elle soit attribuée à un client ou un restaurant.
- **Production et gestion**
 - Ces classes représentent les pizzerias, les stocks et les pizzas.
- **Commande**
 - Ces classes représentent les commandes, leur contenu et leur état.

Structure des sources

Le projet suivra l'architecture MVT propre à Django : Model View Template

Les répertoires sources du projet suivent la structure classique d'un projet Django:

```
.
├── manage.py
├── oc_pizza_app
│   ├── admin.py
│   ├── forms
│   ├── models
│   ├── static
│   │   ├── css
│   │   ├── img
│   │   ├── js
│   │   └── svg
│   ├── templates
│   │   └── oc_pizza_app
│   │       └── index.html
│   ├── urls.py
│   └── views
└── oc_pizza
    ├── settings.py
    ├── urls.py
    └── wsgi.py
```


Modèle physique de données

Le modèle physique de données représente la structure de la base de données. Il a été créé à partir du diagramme de classes.

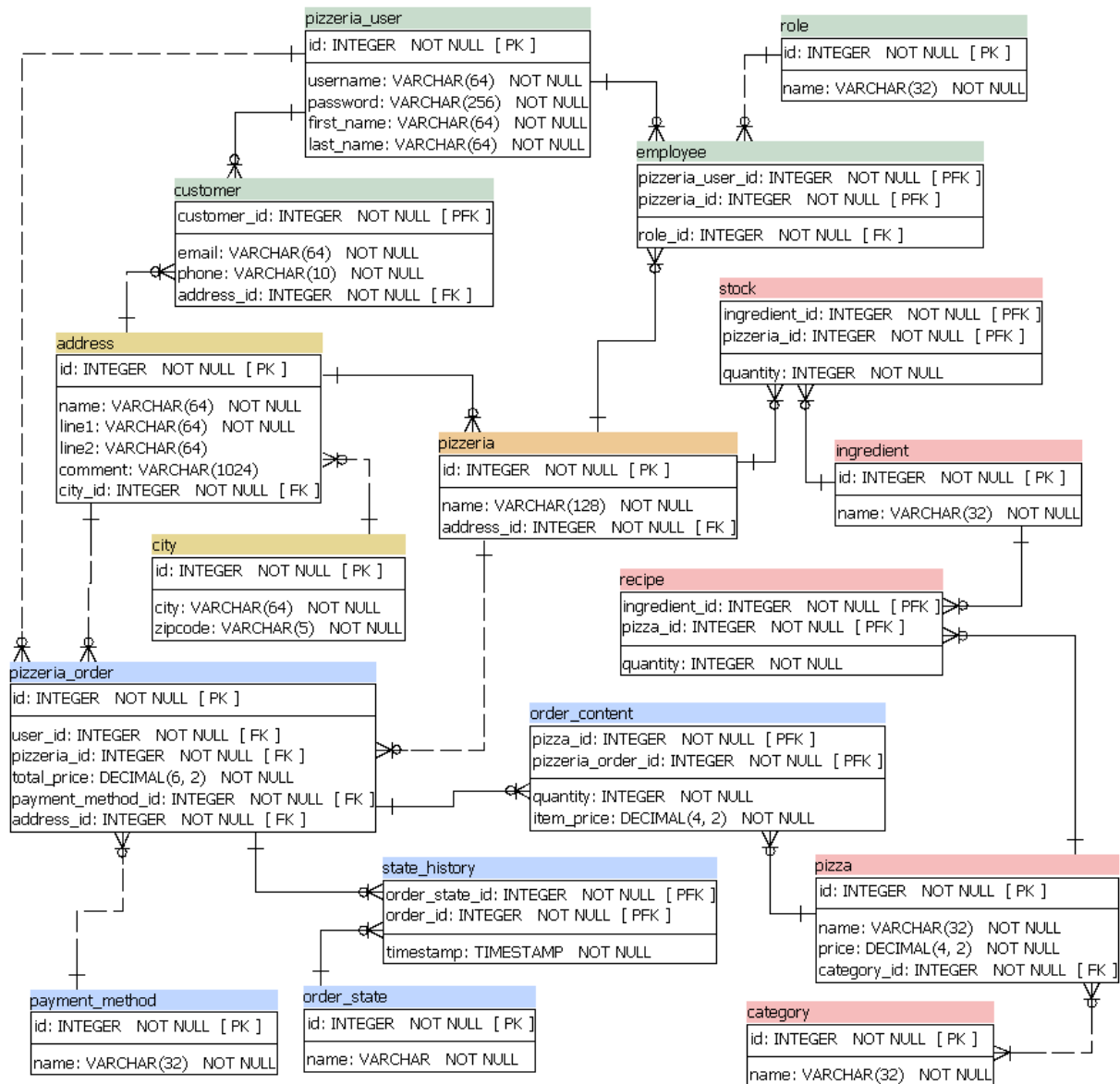


fig. 04

Points particuliers

Configuration

Les points suivants devront être vérifiés avant de pouvoir déployer l'application:

Dans `settings.py`:

- La valeur `SECRET_KEY` devra être générée de manière à ce qu'elle soit suffisamment sécurisée, et devra être placée en variable d'environnement.
Cette valeur doit absolument être gardée secrète, la sécurité de l'application en dépend.
- La valeur `DEBUG` devra être définie à `False`.

Environnement de développement

L'application sera développée dans un environnement virtuel.

La version de Python utilisée sera supérieure ou égale à 3.8.

Procédure de livraison

L'application web sera déployée sur un serveur hébergé chez DigitalOcean.

La livraison se fera sous la forme d'un dépôt Git privé auquel le client aura accès.