

OC PIZZA

Application Web

Dossier de conception technique

Version 1.0.0

Auteur
Patrice Preudhomme
Développeur

TABLE DES MATIERES

1 - Versions	3
2 - Introduction	4
2.1 - Objet du document	4
3 - Le domaine fonctionnel	5
3.1 - Référentiel	5
3.1.1 - Diagramme UML de classes.....	5
4 - Architecture Technique.....	7
4.1 - Application Web	7
4.1.1 - Diagramme UML de Composants	7
4.1.2 - Composant Frontend	8
4.1.3 - Composant Backend	8
4.1.4 - Composant Data Flux Manager	8
4.1.5 - Composant Bank System.....	8
5 - Architecture de Déploiement	9
5.1 - Diagramme UML de déploiement	9
5.2 - Serveur de Base de données	10
5.3 - Serveur Application	10
5.4 - Serveur WEB NGINX	10
6 - Modele Physique de Donnees	11
7 - Exemples de requêtes.....	12
8 - Glossaire.....	13

1 - VERSIONS

Auteur	Date	Description	Version
Preudhomme Patrice	29/10/2020	Création du document	1.0.0

2 - INTRODUCTION

2.1 - Objet du document

Le présent document constitue le dossier de conception technique de l'application OC Pizza et fait suite au dossier de conception fonctionnelle.

Pour rappel, « OC Pizza » est un jeune groupe de pizzeria 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 la fin de l'année.

L'objectif du document est la description du domaine fonctionnel et la conception de l'architecture technique de la solution répondant aux besoins du client. Les livrables sont :

- Une description du domaine fonctionnel (diagramme de classes)
- Le modèle physique de données (MPD) au format PDF
- Base de données MySQL avec un jeu de données de démonstration :
 - un fichier contenant l'ensemble des scripts SQL de création de la base de données
 - un fichier contenant le jeu de données de démonstration
 - un fichier contenant le « dump » de la base de données qui contient le tout
 - un fichier contenant des requêtes SQL afin d'illustrer le fonctionnement de la base de données
- Les différents composants du système et les composants externes utilisés par celui-ci et leur interaction (diagramme de composants)
- La description de l'organisation physique de ces composants (diagramme de déploiement)

3 - LE DOMAINE FONCTIONNEL

3.1 - Référentiel

3.1.1 - Diagramme UML de classes

Le diagramme de classes UML permet de décrire la structure d'un système en modélisant ses classes, ses attributs et les relations entre ses objets. Dans notre cas, les classes sont regroupées dans 3 domaines fonctionnels qui sont la gestion des commandes, des inventaires et la gestion administrative respectivement colorés en orange, vert et bleu dans le diagramme. Ce diagramme servira de support au modèle physique de données. Aussi, on distingue les classes d'association en blanc.

Nous avons choisi de créer deux tables « trading product » pour les produits de négoce et une table pour les pizzas afin d'éviter des confusions. Les attributs peuvent être très similaires et de plus une pizza est toujours liée aux ingrédients.

Un client peut passer plusieurs commandes. La classe « User » liée à la classes « Adress » et « Customer » contiennent les attributs décrivant les propriétés concernant le client. La classe « Order » de commande contient l'attribut « o_statut » pour suivre l'état de la commande.

L'attribut « start_date » de la classe d'association entre les classes « Product » et « Restaurant » permet d'historiser les prix.

La relation « Pizza » - Association » – « Ingredient » décrit quels ingrédients appartiennent aux pizzas. Une pizza peut contenir 1 ou plusieurs ingrédients. Il est impossible pour une pizza de ne contenir aucun ingrédient.

Nous avons une table « Dashboard » destinée à suivre l'activité des pizzerias.

La classe « Staff » décrit le rôle de l'employé et est liée à la classe « AccessRight » pour délimiter ce qu'un rôle peut et ne peut pas faire.

La classe « Address » est utilisée pour les utilisateurs (clients/employés) ainsi que les fournisseurs « Supplier ».



4 - ARCHITECTURE TECHNIQUE

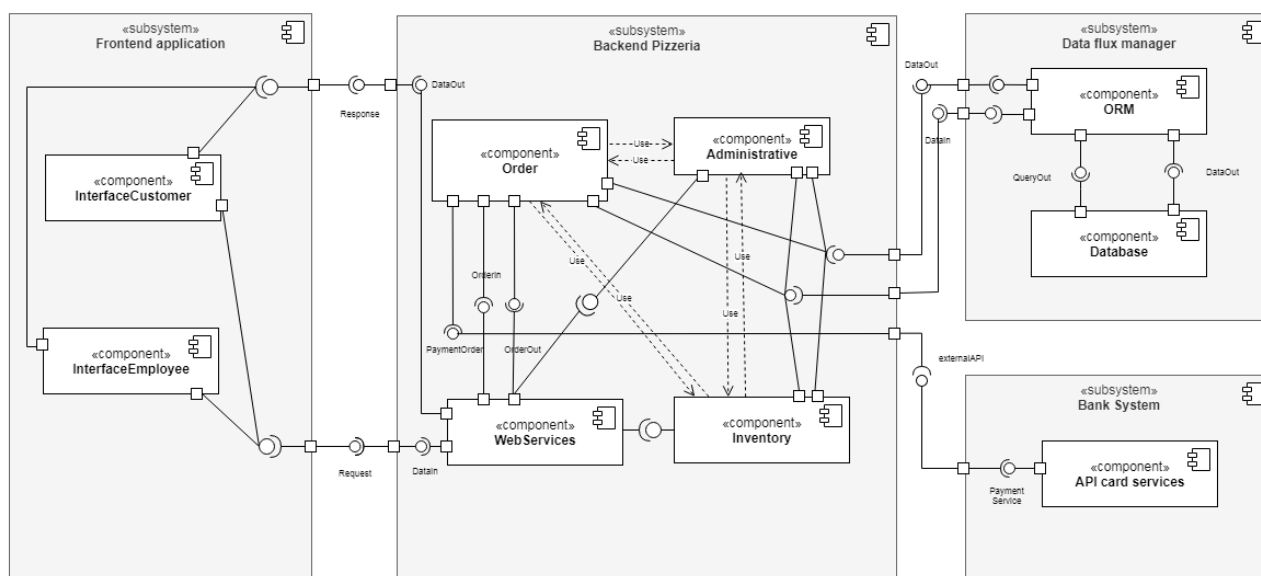
4.1 - Application Web

La pile logicielle est la suivante :

- Application Python 3.7
- Serveur d'application Unix
- Serveur WEB NGINX
- MySQL version 8.0.18

4.1.1 - Diagramme UML de Composants

Ce diagramme de composants a pour objectif d'illustrer la relation entre les différents composants de système composés de 4 composants ou sous-systèmes.



4.1.2 - Composant Frontend

Côté client - Interface client : site web responsive développé de façon à pouvoir s'adapter à toutes les résolutions d'écran (ordinateur, smartphone ou tablette). 3 langages : HTML, CSS et Javascript. C'est donc un seul et même système qui sera développé. Cette solution a de nombreux atouts: un coût maîtrisé puisque la programmation n'est faite qu'une fois, une compatibilité avec tous les ordinateurs, smartphones et tablettes équipés d'un navigateur et d'une connexion internet (iPhones, Androids, Blackberry, Windows..) ainsi qu'une administration unique et simplifiée.

Il existe deux interfaces graphiques distinctes. L'une est pour les clients pour créer les commandes et une pour les employés pour le traitement des commandes et à des fins administratives.

La communication se fait en utilisant le protocole HTTP sur internet.

4.1.3 - Composant Backend

Il est composé des quatre composants *Order*, *Administrative*, *Inventory* et *Webservices*.

C'est la clé de voûte de l'application web et en accord avec le cahier des charges. La partie back-end va construire, développer et mettre en interaction les 3 piliers essentiels au fonctionnement de l'application que sont le serveur d'hébergement, l'application web et la base de données qui dans notre cas est représentée par le sous-système *Data flux manager*. Le langage PYTHON est associé au Framework DJANGO

4.1.4 - Composant Data Flux Manager

Le sous-système *Data flux manager* contient le composant *ORM* qui se place en interface entre le programme applicatif et le composant *Database*, base de données relationnelle pour simuler une base de données orientée objet. Il propose l'interface *DataOut* et requiert l'interface *DataIn*.

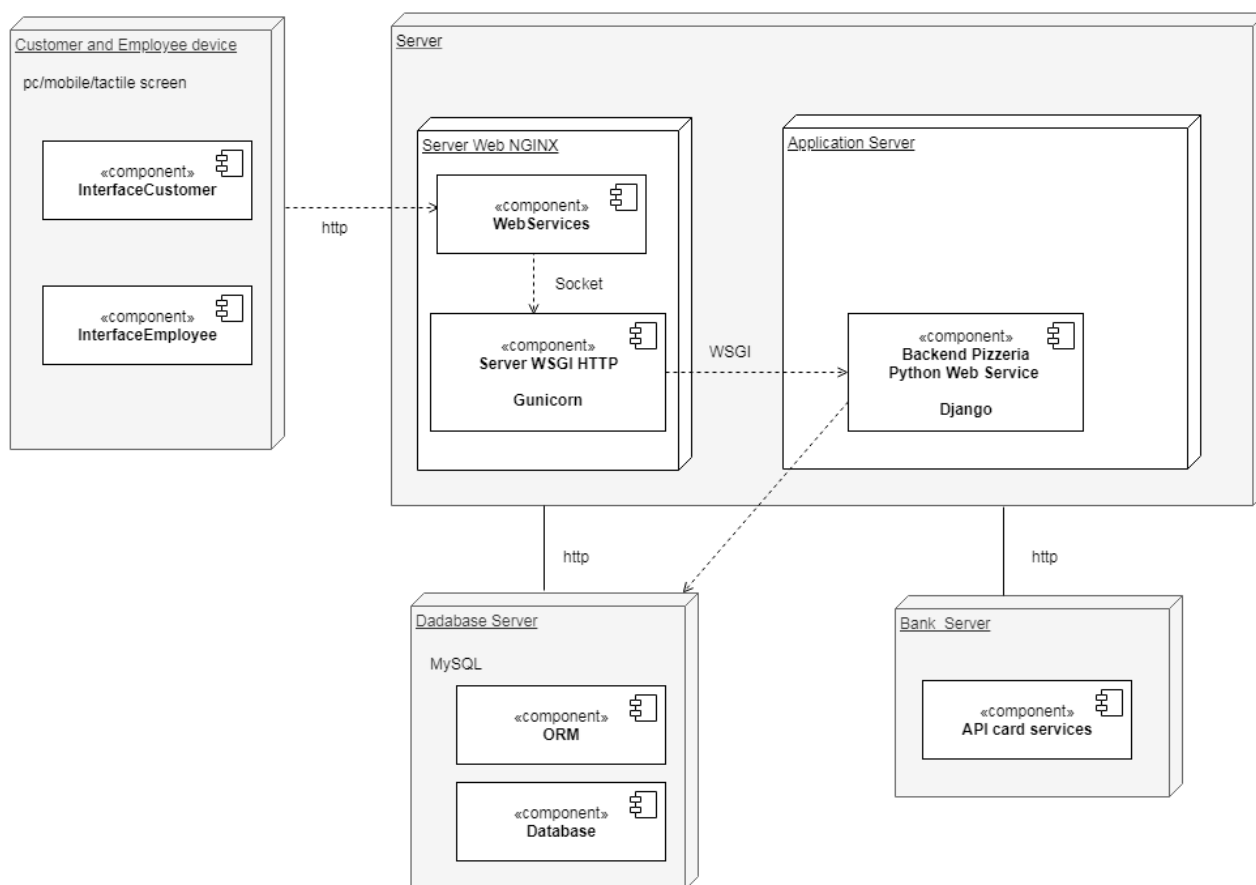
4.1.5 - Composant Bank System

Ce composant fournit l'interface *PaymentService* pour effectuer les paiements via notre application. L'interface *PaymentOrder* exposée par le composant *Order* peut joindre l'interface *External API* et utiliser l'interface *PaymentService* de la banque.

5 - ARCHITECTURE DE DÉPLOIEMENT

5.1 - Diagramme UML de déploiement

Ce diagramme de déploiement décrit la disposition physique des ressources matérielles qui composent le système et montre la répartition des composants sur ces matériels.



5.2 - Serveur de Base de données

Comme pour le serveur HTTP, le terme « serveur base de données » caractérise à la fois l'ordinateur physique sur lequel se trouve hébergé la base de données et le programme appelé Système de gestion de base de données (SGBD). La base de données collecte des informations et les organise pour que ces dernières soient facilement consultables, gérables et mises à jour. Grâce à une base de données, il est possible de créer, de mettre à jour ou de supprimer des données.

En l'espèce, le serveur MySQL 8.0.18 avec moteur de stockage InnoDB (transactionnel et gère les clés étrangères) est installé sous Unix.

5.3 - Serveur Application

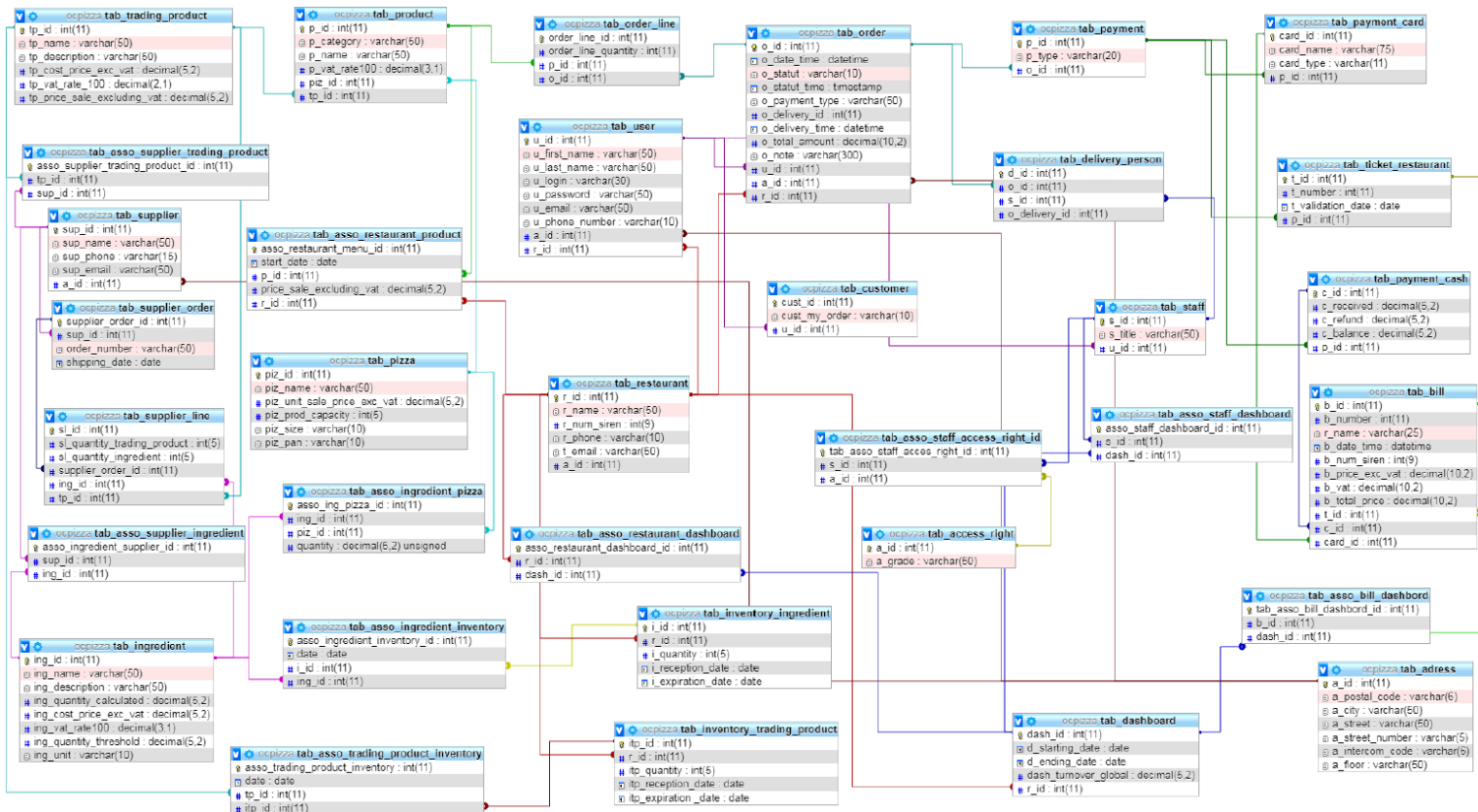
Côté serveur - Interface d'OC Pizza : le langage PYTHON est associé au Framework DJANGO. Le serveur hébergeant l'application communique avec le serveur hébergeant la base de données et le serveur en charge d'effectuer les transactions bancaires.

5.4 - Serveur WEB NGINX

A noter, que la proposition technique du dossier de conception fonctionnelle initialement orientée vers une solution Apache a évolué vers une solution NGINX qui est également libre. Ce programme qui tourne constamment en arrière-plan a pour mission d'intercepter les requêtes HTTP et d'y donner suite. NGINX est utilisé comme un serveur Web classique pour servir des fichiers statiques et comme un proxy pour les requêtes dynamiques typiquement acheminées en utilisant une interface avec le serveur applicatif Gunicorn

6 - MODELE PHYSIQUE DE DONNEES

Pour une meilleure résolution d'affichage du diagramme, un fichier PDF est fourni « PGESTpizza_02_PDM diagram.pdf ». Pour concevoir la base de données MySQL, phpMyAdmin a été utilisé car l'outil est gratuit et son interface permet un gain de temps significatif.



7 - EXEMPLES DE REQUÊTES

Afin d'illustrer le fonctionnement de la base de données, sont présentées ci-dessous les requêtes réalisées sur la base de données de démonstration. Ces requêtes sont dans le fichier « ocpizza_requete.sql »

- Ma commande 6 peut-elle contenir plusieurs pizzas?
- Puis-je retrouver le contenu de la commande 2?
- Puis-je afficher les commandes en statut préparation dans le point de vente OC Pizza 4 ?
- Puis-je afficher les commandes en statut préparation des clients dans les pizzerias?
- Puis-je retrouver le prix payé pour une pizza dans une commande terminée même si le prix a changé depuis?
- Puis-je lister les pizzas pour lesquelles tous les ingrédients sont en stock?

8 - GLOSSAIRE

API	Une API est un ensemble de définitions et de protocoles qui facilite la création et l'intégration de logiciels d'applications. API est un acronyme anglais qui signifie « Application Programming Interface », que l'on traduit par interface de programmation d'application.
Django-Python	Django est un framework Python destiné au web. Python est un langage utilisé dans la robotique, le big data et le web.
Gunicorn	Diminutif de Green Unicorn. Il s'agit d'un serveur HTTP Python pour Unix qui utilise les spécifications WSGI (Web Server Gateway Interface).
HTTP	HTTP est la langue dans laquelle le navigateur Web parle au serveur Web afin de lui communiquer ce qui est demandé.
MySQL	Système de Gestion de Bases de Données Relationnelles
ORM	Un mapping objet-relationnel (en anglais object-relational mapping ou ORM) est un type de programme informatique qui se place en interface entre un programme applicatif et une base de données relationnelle pour simuler une base de données orientée objet. Ce programme définit des correspondances entre les schémas de la base de données et les classes du programme applicatif.
UML	Unified Modeling Language (UML), est un langage de modélisation graphique à base de pictogrammes conçu pour fournir une méthode normalisée pour visualiser la conception d'un système.
WSGI	Norme qui sert à définir comment un serveur Python et son application peuvent discuter