

O.C. PIZZA

Projet Origanware

Dossier de conception technique

Version 0.1

Auteur
Cédric Joseph
Développeur

TABLE DES MATIÈRES

Versions	2
Introduction	3
Objet du document	4
Le domaine fonctionnel	4
Référentiel	4
Package enregistrements et gestion des achats:	6
Package préparation et logistique	7
Package gestion administrative	7
Package gestion commerciale	7
Architecture Technique	7
Architecture de Déploiement	11
Glossaire	13

1 - Versions

Auteur	Date	Description	Version
Cédric Joseph	03/10/2020	Création du document	0.1

2 - INTRODUCTION

2.1 -Objet du document

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

L'objectif du document est de préciser les éléments techniques nécessaires au développement de l'application

Les éléments du présent dossiers découlent :

- de l'analyse fonctionnelle effectuée au préalable
- de l'expérience acquise d'une première version élémentaire de l'application (POC)

3 - LE DOMAINE FONCTIONNEL

3.1 -Référentiel

Notre solution, Origanware, sera une application web à laquelle les différents acteurs pourront se connecter et qui leur proposera une interface adaptée à leur besoin.

Le diagramme de classes ci-dessous, met en évidence les relations entre les différentes données concernées par les différents acteurs du système.

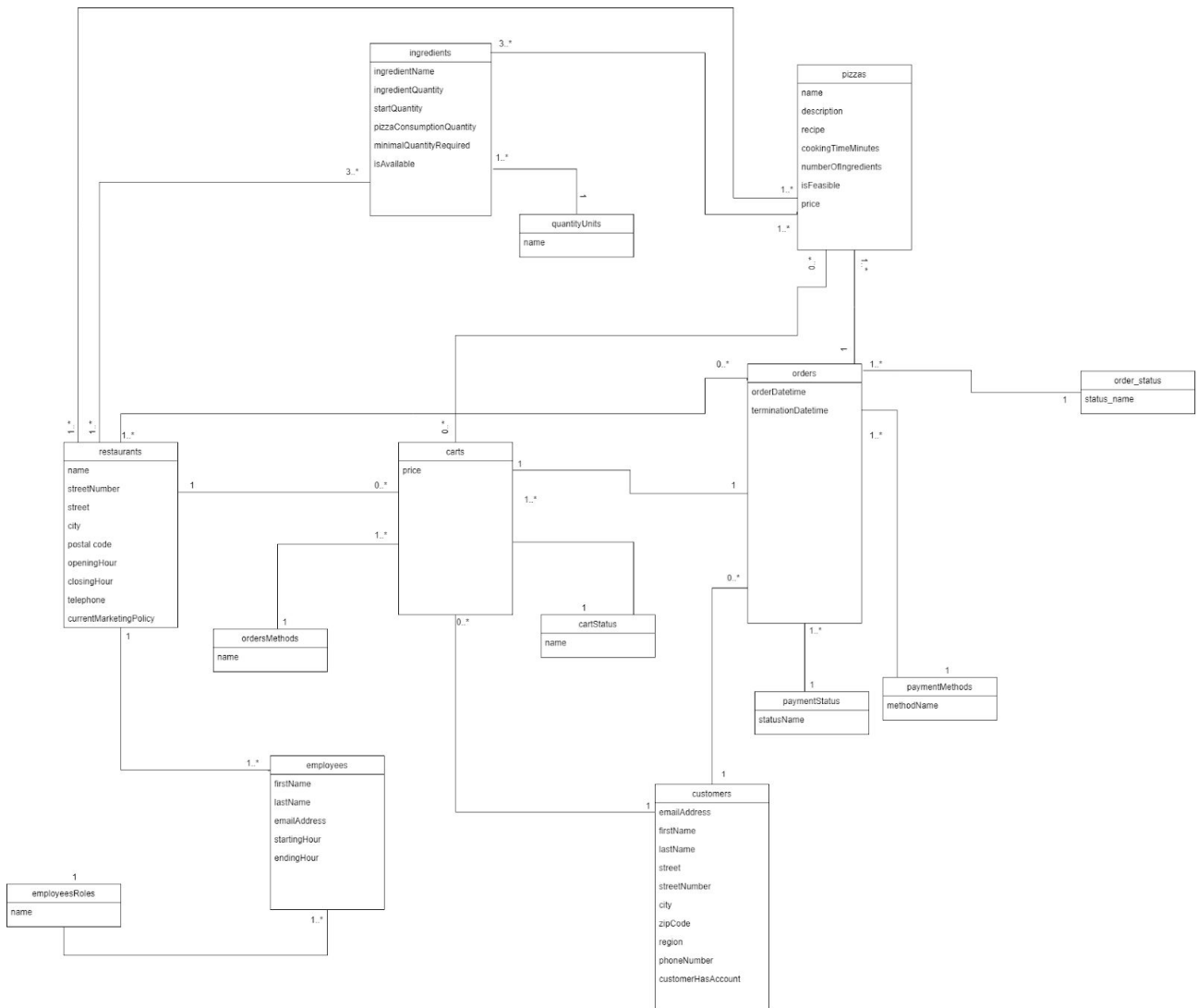


Diagramme UML de classes

Ce diagramme représente les relations de parenté et de dépendance de notre système. Il se compose de relations *one to one*, *one to many* / *many to one* et *many to many*.

3.2 -Package enregistrements et gestion des achats:

Lorsqu'un utilisateur remplit un panier :

- Un panier est lié à un unique client. Un client ne possède qu'un seul panier.
- Le panier est composé de plusieurs pizzas elles-mêmes composées de plusieurs ingrédients
- Un panier possède un statut. Plusieurs statuts peuvent être attribués à un panier (en cours, validé, archivé)
- Une commande est le résultat de la validation d'un unique panier
- Une commande est associée à une unique facture
- une commande comme un panier est associée à un unique client
- un client peut avoir passé plusieurs commandes
- une commande est associée à plusieurs pizzas
- une commande possède une date et heure de création
- une commande possède un statut de paiement
- plusieurs statuts de paiement peuvent être associés à une même commande
- une commande possède un statut lié à sa livraison (comprend les étapes de préparation)
- plusieurs statuts de livraisons s'appliquent (à tour de rôle), à une même commande
- Une commande est associée à un unique restaurant
- Un restaurant enregistre plusieurs commandes

3.3 -Package préparation et logistique

- Un restaurant possède sa propre liste d'ingrédients. En fonction de leur quantité des pizzas sont proposées sur le site
- Un ingrédient est présent en une quantité de début de service
- Chaque pizza consommant l'ingrédient réduit automatiquement cette quantité lors d'une commande, ce qui donne la quantité courante
- Un ingrédient est présent en une quantité enregistrée en fin de service
- Un ingrédient possède une unité de mesure
- Une unité de mesure peut être associée à plusieurs ingrédients

3.4 -Package gestion administrative

- Chaque restaurant possède son catalogue de pizzas
- Une pizza peut être proposée par plusieurs restaurants
- Une pizza possède une description, une recette, un prix, des ingrédients et en fonction de leur quantité, un statut faisable ou non
- Un utilisateur possède un seul compte par adresse email
- Un employé possède un unique rôle
- Un même rôle peut être attribué à plusieurs employés

3.5 -Package gestion commerciale

- Un restaurant est soumis à une seule politique commerciale
- Une politique commerciale peut s'appliquer à plusieurs restaurants

4 - ARCHITECTURE TECHNIQUE

4.1 -Application Web

La pile logicielle est la suivante :

- Application **Python 3.6.9**
- Affichage dans le navigateur **Bootstrap 4**
- Serveur d'application **Gunicorn 20.0.1**
- Server web **NGINX 1.19.2**
- Base de données **PostgreSQL 12**
- Framework de développement **Django 3.1**
- OS du serveur **Ubuntu 20**
- Outil de monitoring : **Sentry**
- Outil de test fonctionnel : **Selenium Webdriver**

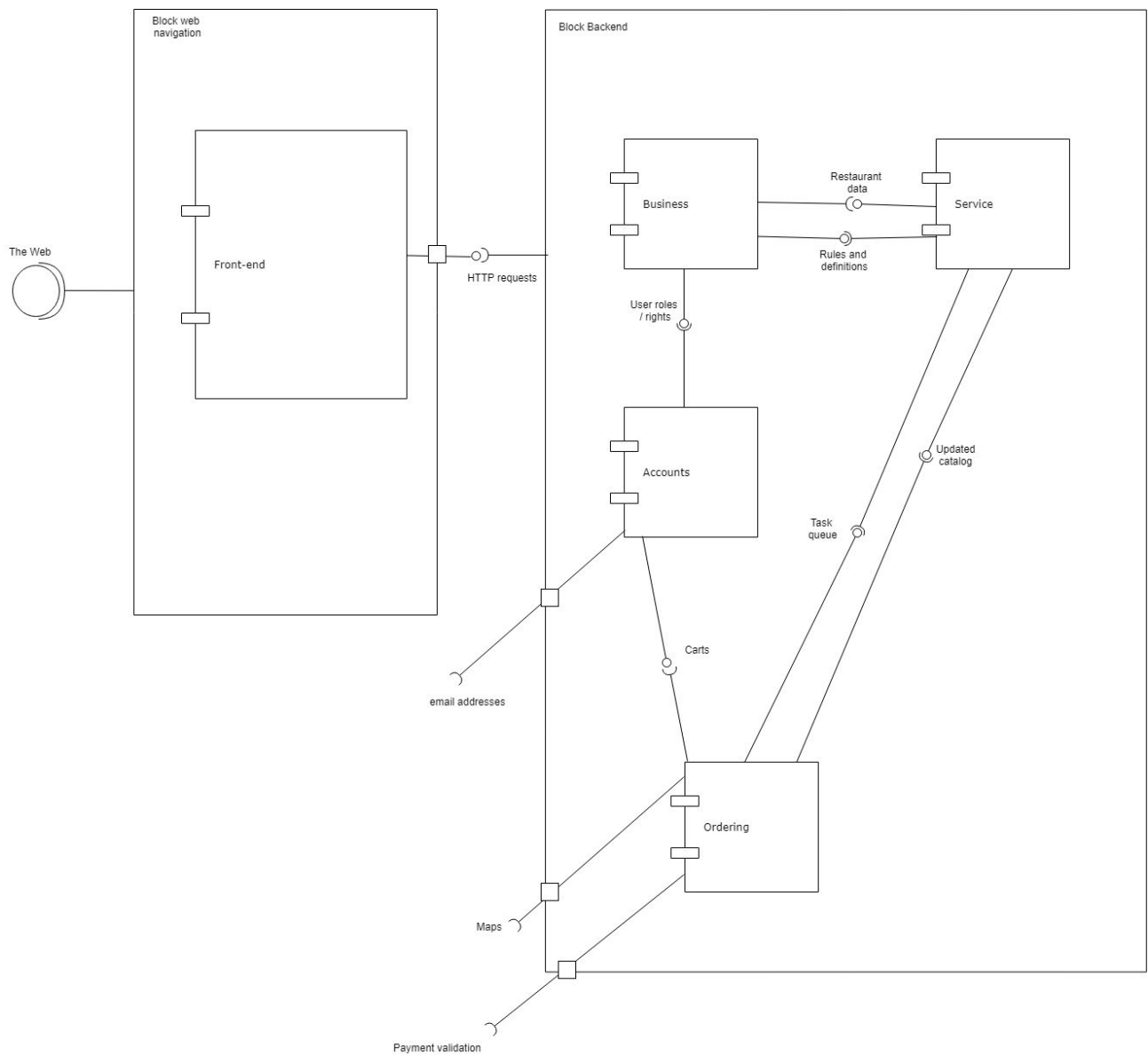


Diagramme UML de Composants

4.1.1 - Composant base de données

La base de données enregistre et conserve les données utilisées par les restaurant et leurs clients. Elle réunit l'ensemble des tables correspondant au modèle logique de données complété par le typage des différents champs attributs d'entités.

Origanware gère :

- les données propres à chaque restaurant
- les différents produits et leur disponibilité
- les ingrédients et leur disponibilité
- les données du personnel
- les données de paniers
- les commandes et leurs statuts de la validation à la livraison
- les paiements et leurs statuts de la validation à la livraison

4.1.2 - Composant Django App Business

Origanware contient une application qui rassemble les caractéristiques du restaurant (nom, adresse, politique commerciale, attribution de pizzas et d'ingrédients à un restaurant, affectation d'employés aux différents restaurants, etc). Elle permet l'ajout, la mise à jour ou la suppression de restaurants. Elle intéressera le directeur commercial qui pourra aussi récupérer les commandes qui référencent un même restaurant.

4.1.3 - Composant Django App Service_management

Origanware contient une application qui permet de fournir au personnel du restaurant l'état des stocks avec les seuils d'indisponibilité, la liste des recettes de leur magasins et le statut des commandes. Elle permet aussi de retrouver et de modifier les heures de début et de fin du personnel.

4.1.4 - Composant Django App Accounts

Origanware contient une application qui permet au personnel et aux clients de créer un compte et de s'identifier grâce à leur email.

4.1.5 - Composant Django App Ordering

Origanware contient une application permettant au client d'avoir un panier dans lequel des pizzas seront ajoutées ou supprimées avant annulation ou confirmation de validation d'un panier. Cette étape change annonce la commande et crée une commande dont le statut évoluera jusqu'à sa clôture ou son archivage. Cette application rend disponible pour l'utilisateur et pour le panier, la liste des pizzas qu'il est encore possible de réaliser et donc de commander.

L'application *Ordering* donnera au client et au cadres d'OC Pizza une grande lisibilité par rapport au service et à l'efficacité du personnel aux différentes étapes du parcours client.

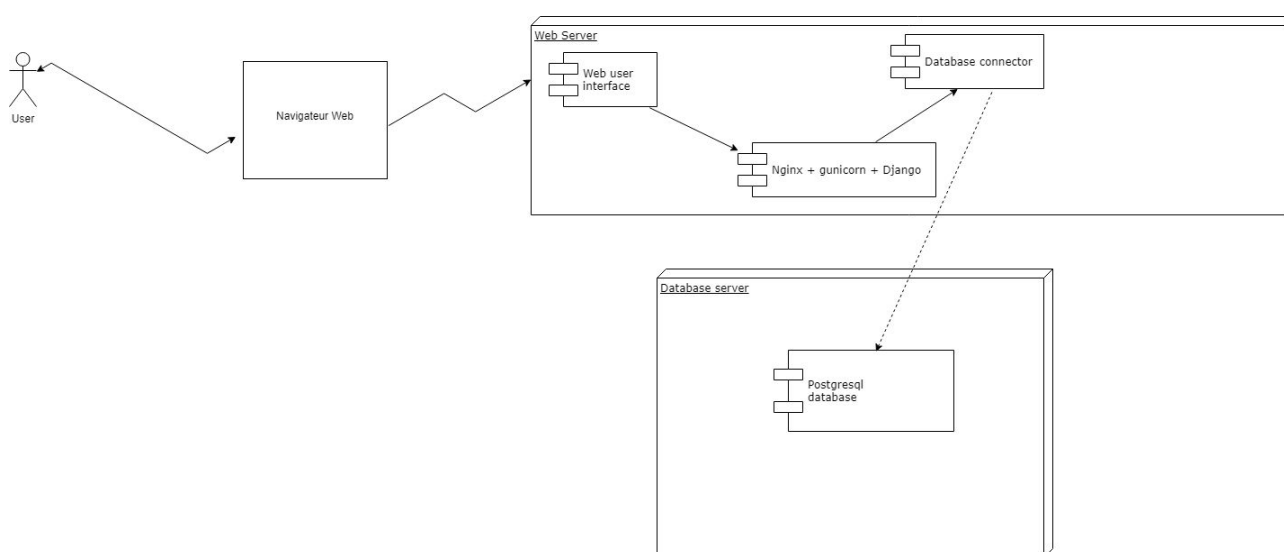
4.1.6 - Composant Front-End (Interface utilisateur)

Origanware affiche un contenu dynamique, souvent interactif dans le navigateur.



5 - ARCHITECTURE DE DÉPLOIEMENT

Diagramme UML de déploiement



Le backend de l'application est déployé sur un unique serveur Linux qui sert au client web les pages demandées par l'utilisateur.

Certaines actions de l'utilisateur envoient des requêtes HTTP qui communiquent avec le code Python via un serveur WSGI Gunicorn. Certaines de ces requêtes HTTP sont à l'origine de communications avec la base de données du projet. L'ORM de Django utilise le connecteur Psycopg2 pour envoyer des requêtes vers la base de données.

5.1 -Serveur de Base de données

Description

Caractéristiques techniques Serveur Linux Ubuntu 20 + PostgreSQL 12

5.2 -Serveur d'application

Serveur WSGI Gunicorn (souvent associé à Nginx et Django).

5.3 -Serveur Web

Serveur HTTP NGinx pour pouvoir traiter un grand nombre de connexions simultanées.

5.4 - Outil d'intégration continue

Travis CI pour sa configuration simple.



6 - GLOSSAIRE

Bootstrap 5	Bootstrap est une collection d'outils utiles à la création du design de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option.
Ubuntu	Ubuntu est un système d'exploitation GNU/Linux basé sur la distribution Linux Debian. Il est développé, commercialisé et maintenu pour les ordinateurs individuels, les serveurs et les objets connectés par la société Canonical
Gunicorn	Gunicorn, pour « Green Unicorn », est un serveur web HTTP WSGI écrit en Python et disponible pour Unix.
WSGI	La Web Server Gateway Interface est une spécification qui définit une interface entre des serveurs et des applications web pour le langage Python. Elle permet au serveur HTTP de communiquer avec une application écrite en Python.
NGINX	NGINX Open Source ou NGINX est un logiciel libre de serveur Web ainsi qu'un proxy inverse écrit par Igor Sysoev, dont le développement a débuté en 2002 pour les besoins d'un site russe à très fort trafic. La documentation est disponible dans plusieurs langues.
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. Pour cette raison, le projet a pour slogan « Le framework pour les perfectionnistes avec des deadlines. »
PostgreSQL	PostgreSQL est un système de gestion de base de données relationnelle
Travis	Travis CI est un logiciel libre d'intégration continue. Il fournit un service en ligne utilisé pour compiler, tester et déployer le code source des logiciels développés, notamment en lien avec le service d'hébergement du code source GitHub. Le logiciel est publié sous licence MIT. Sa configuration s'effectue en YAML.
Sentry	Sentry est un tableau de bord qui permet de visualiser ce qui se passe dans l'application Django. Il affiche des logs et fournit des détails concernant le navigateur de l'utilisateur, l'URL.

Selenium Webdriver	<p>Selenium est projet englobant un éventail d'outils et de bibliothèques permettant l'automatisation des navigateurs internet.</p> <p>Au cœur de Selenium se trouve WebDriver, une interface permettant d'écrire des instructions pouvant être exécutées indifféremment par de nombreux navigateurs.</p>
Psycopg2	<p>psycopg2 est l'adaptateur de base de données PostgreSQL le plus populaire, à la fois léger et efficace. C'est l'implémentation actuelle de l'adaptateur PostgreSQL.</p>
ORM (Object-Relational Mapping)	<p>Technique de programmation informatique qui permet de simplifier l'accès à une base de données en proposant à l'informaticien des « objets » plutôt que d'accéder directement à des données relationnelles. Ce niveau d'abstraction supplémentaire fait correspondre le monde objet (programmation orientée objet) et le monde relationnel (les bases de données relationnelles).</p>