

OC Pizza

Gestion de pizzeria - application web

Dossier de conception technique

Version 1.1

Auteur

Jody Etienne

Analyste-programmeur

TABLE DES MATIÈRES

1 - Versions.....	4
2 - Introduction.....	5
2.1 - Objet du document.....	5
2.2 - Références.....	5
3 - Le domaine fonctionnel.....	6
3.1 - Référentiel - diagramme de classe.....	6
3.1.1 - Règles de gestion.....	7
3.2 - Référentiel – modèle physique de données.....	10
4 - Architecture Technique.....	11
4.1 - Application Web.....	11
4.1.1 - Composants sous-système Webstore – site web.....	12
4.1.1.1 - Composant SearchEngine.....	12
4.1.1.2 - Composant OrderDetail.....	12
4.1.1.3 - Composant Authentication.....	13
4.1.2 - Composants sous-système Menu – site web.....	13
4.1.2.1 - Composant Products.....	13
4.1.3 - Composants sous-système Accounting – site web.....	13
4.1.3.1 - Composant Orders.....	13
4.1.3.2 - Composant Customers.....	13
4.1.3.3 - Composant Employee.....	13
4.1.3.4 - Composant Accounts.....	14
4.1.3.5 - Composant Django ORM.....	14
4.1.3.6 - Composant OC Pizza DB.....	14
4.1.4 - Composants sous-système Back-office - restaurant.....	15
4.1.4.1 - Composant CreateOrder.....	15
4.1.4.2 - Composant Authentication.....	15
4.1.5 - Composants sous-système Menu – restaurant.....	16
4.1.5.1 - Composant Products.....	16
4.1.6 - Composants sous-système Accounting – restaurant.....	16
4.1.6.1 - Composant Orders.....	16
4.1.6.2 - Composant Users.....	16
4.1.6.3 - Composant Payment.....	16
4.1.6.4 - Composant Accounts.....	16
4.1.6.5 - Composant Django ORM.....	17
4.1.6.6 - Composant OC Pizza DB.....	17
4.1.7 - Composants sous-système Supplied - livraison.....	18
4.1.7.1 - Composant OrderDelivery.....	18
4.1.7.2 - Composant Authentication.....	18
4.1.8 - Composants sous-système Accounting – livraison.....	19
4.1.8.1 - Composant Orders.....	19
4.1.8.2 - Composant Users.....	19
4.1.8.3 - Composant Payment.....	19
4.1.8.4 - Composant Accounts.....	19
4.1.8.5 - Composant GeolocApi.....	19
4.1.8.6 - Composant Django ORM.....	19
4.1.8.7 - Composant OC Pizza DB.....	20
5 - Architecture de Déploiement.....	21
5.1 - Serveur de Base de données.....	22

5.2 - Serveur web NGINX.....	22
5.3 - Serveur d'application GUNICORN.....	23
5.4 - Serveur d'hébergement.....	23
6 - Points particuliers.....	25
6.1 - Gestion des logs.....	25
6.2 - Ressources.....	25
6.3 - Environnement de développement.....	25
6.4 - Procédure de packaging / livraison.....	25
7 - Glossaire.....	26

1 - VERSIONS

Auteur	Date	Description	Version
Jody Etienne	28/05/2020	Création du document	1.0
Jody Etienne	30/09/2020	Finalisation du document	1.1

2 - INTRODUCTION

2.1 - Objet du document

Le présent document constitue le dossier de conception technique de l'application OC Pizza. Nous nous intéressons aux besoins de nos clients tels que la gestion des utilisateurs, des commandes, des paiements à travers les spécifications techniques pour aboutir à un site de qualité qui répond aux besoins de OC Pizza.

Les éléments du présents dossiers découlent d'une prise de contact avec l'un des responsable du groupe.

Objectif :

Mise en place d'un nouveau système informatique pour l'ensemble des pizzerias du groupe OC Pizza.

2.2 - Références

Pour de plus amples informations, se référer également aux éléments suivants :

1. **DCF** : Dossier de conception fonctionnelle de l'application
2. **DE** : Dossier d'exploitation de l'application
3. **PVL** : Procès Verbal de livraison

3 - LE DOMAINE FONCTIONNEL

3.1 - Référentiel - diagramme de classe

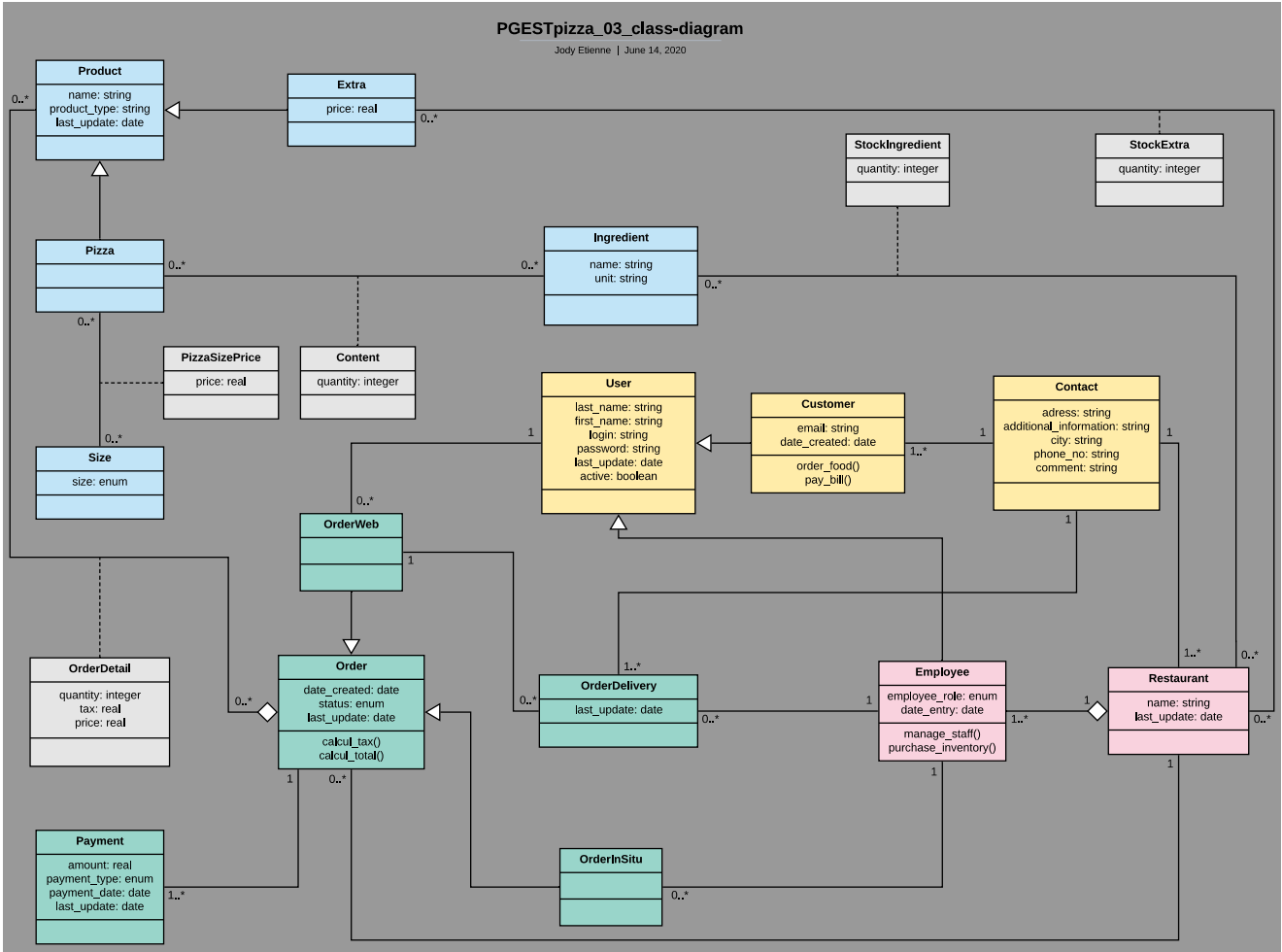


Figure 1: Diagramme de classe - OC Pizza

Les différents éléments de ce diagramme de classe représentent les classes qui seront effectivement programmées, les principaux objets ou les interactions entre les classes et les objets.

3.1.1 - Règles de gestion

Descriptions textuelles des différentes relations entre les classes.

Relations entre les classes :

> Product et Order :

Many to Many

- - Un Order est composé d'aucun, un ou plusieurs Product.
- - Un Product appartient à aucun, un ou plusieurs Order.

De cette relation va être implémenté une classe d'association nommée OrderDetail.

> OrderWeb et OrderInsitu avec Order :

- - Héritage de OrderWeb et OrderInsitu avec les tuples de Order.

> OrderWeb et OrderDelivery :

One to Many

- - Un OrderWeb est constitué de aucun, un ou plusieurs OrderDelivery.
- - Un OrderDelivery appartient à un OrderWeb.

> Order et Payment :

Many to One

- - Un Order est constitué d'un à plusieurs Payment (possibilité de régler une commande avec une partie en CB et une autre partie en espèces par exemple).
- - Un Payment appartient à un Order.

> OrderWeb et User :

One to Many

- - Un User possède aucun, un ou plusieurs OrderWeb.
- - Un OrderWeb appartient à un User.

> Order et Restaurant :

One to Many

- - Un Restaurant gère aucun, un ou plusieurs Order.
- - Un Order appartient à un Restaurant.

> Restaurant et Extra :

Many to Many

- - Un Restaurant gère aucun, un ou plusieurs Extra.
- - Un Extra appartient à aucun, un ou plusieurs Restaurant.

De cette relation va être implémenté une classe d'association nommée Stock_Extra.

> Restaurant et Ingredient :

Many to Many

- - Un Restaurant gère aucun, un ou plusieurs Ingredient.
- - Un Ingredient appartient à aucun, un ou plusieurs Restaurant.

De cette relation va être implémenté une classe d'association nommée Stock_Ingredient.

> Pizza et Ingredient :

Many to Many

- - Une Pizza est composé de aucun, un ou plusieurs Ingredient.
- - Un Ingredient compose aucun, un ou plusieurs Pizza.

De cette relation va être implémenté une classe d'association nommée Content.

> Pizza et Size :

Many to Many

- - Une Pizza est composé d'aucun, un ou plusieurs Size.
- - Une Size appartient à aucun, un ou plusieurs Pizza.

De cette relation va être implémenté une classe d'association nommée PizzaSizePrice.

> Pizza et Size avec Product :

- - Héritage de Pizza et Size avec les tuples de Product.

> Restaurant et Employee :

Many to One

- - Un Restaurant gère un à plusieurs Employee.

- - Un Employee travaille pour un Restaurant.

> Employee et Customer avec User :

- - Héritage de Employee et Customer avec les tuples de User.

> Employee et OrderDelivery :

Many to One

- - Un Employee gère aucun, un ou plusieurs OrderDelivery.

- - Un OrderDelivery est prise en charge par un Employee.

> Contact et Customer :

Many to One

- - Un Customer possède un Contact.

- - Un Contact appartient à un ou plusieurs Customer.

> Contact et Restaurant :

Many to One

- - Un Restaurant possède un Contact.

- - Un Contact appartient à un ou plusieurs Restaurant.

> Contact et OrderDelivery :

Many to One

- - Un OrderDelivery possède un Contact.

- - Un Contact appartient à un ou plusieurs OrderDelivery.

3.2 - Référentiel – modèle physique de données

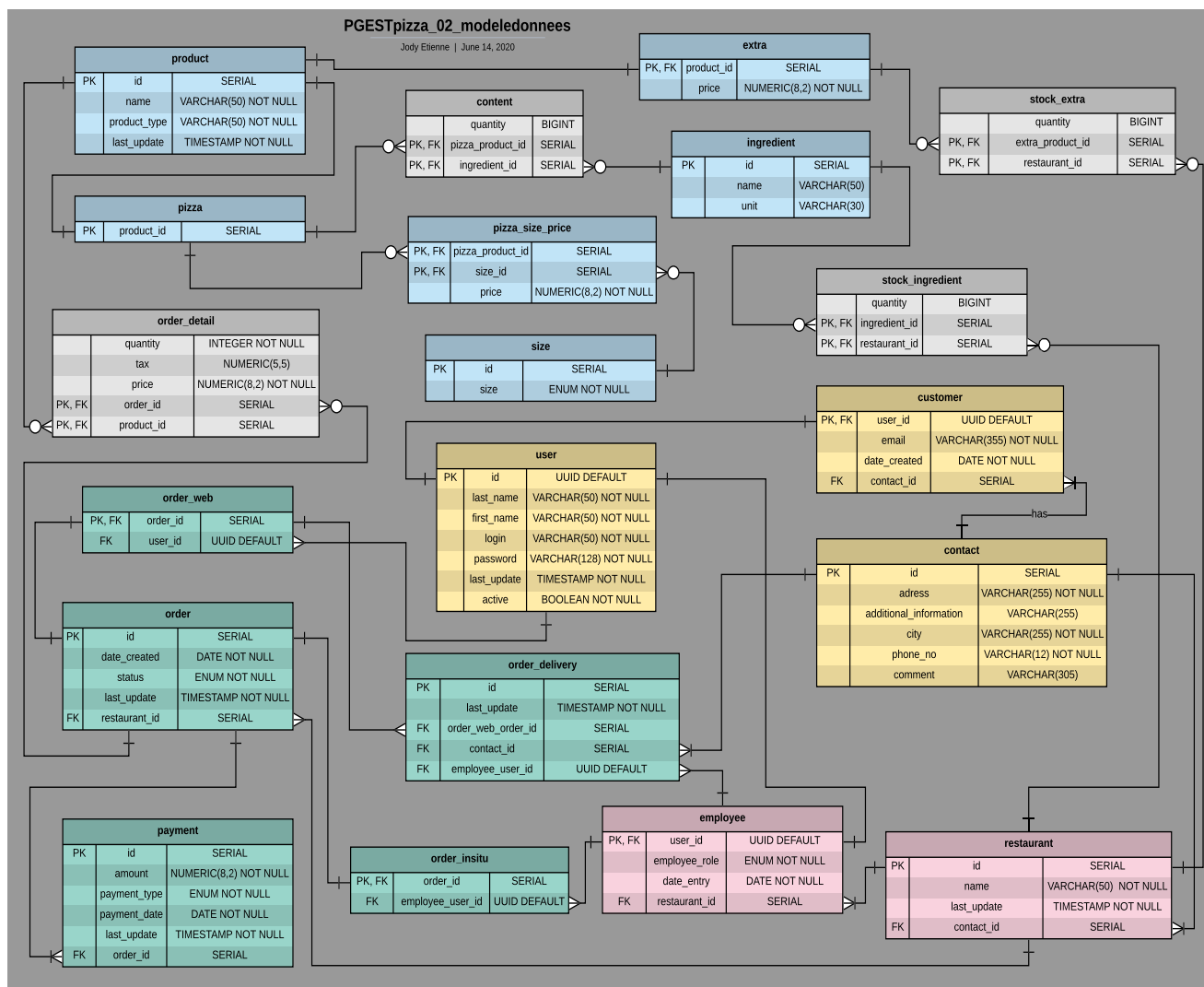


Figure 2: Modèle physique de données - OC Pizza

Les différents éléments de ce modèle physique de données représentent les tables qui seront effectivement implantés dans le serveur de base de donnée PostgreSQL de l'application de OC Pizza.

4 - ARCHITECTURE TECHNIQUE

4.1 - Application Web

La pile logicielle est la suivante :

- Application : **Django** version 3.1 + **Python** version 3.8.6
- Serveur d'application : **Gunicorn** version 20.0.4
- Serveur web : **NGINX** version 1.19.2
- Serveur de Gestion de Base de Donnée Relationnelle : **PostgreSQL** version 13

Avec **Django**, nous pouvons faire passer les applications Web du concept au lancement en quelques heures. Django s'occupe de la plupart des complexités du développement Web, nous pouvez donc vous concentrer sur la rédaction de l'application sans avoir à réinventer la roue. C'est gratuit et open source.

Django a été conçu pour aider à faire passer les applications du concept à la réalisation le plus rapidement possible.

Django comprend des dizaines d'extensions que nous pouvons utiliser pour gérer les tâches courantes de développement Web. Django s'occupe de l'authentification des utilisateurs, de l'administration du contenu, des plans du site, des flux RSS et de bien d'autres tâches - dès le départ.

Django prend la sécurité au sérieux et aide à éviter de nombreuses erreurs de sécurité courantes, telles que l'injection SQL, le cross-site scripting, le cross-site request forgery et le clickjacking. Son système d'authentification des utilisateurs offre un moyen sûr de gérer les comptes et les mots de passe des utilisateurs.

Certains des sites les plus fréquentés de la planète utilisent la capacité de Django à s'adapter rapidement et avec souplesse pour répondre aux demandes de trafic les plus importantes.

Les entreprises, les organisations et les gouvernements ont utilisé Django pour construire toutes sortes de choses, des systèmes de gestion de contenu aux réseaux sociaux en passant par les plateformes de calcul scientifique.

Avec **Python**, nous pouvons bénéficier de la vaste bibliothèque d'outils d'analyse de données open source, de frameworks web et d'instruments de test de Python qui fait de son écosystème l'un des plus vastes de toutes les communautés de programmation. Il fonctionne sur tous les systèmes d'exploitation (Windows, Mac et Linux).

Python est un langage accessible pour tout les niveaux de programmeurs car la communauté fournit de nombreuses ressources d'introduction, de formation et de cas pratique d'utilisation.

PGESTpizza_04_component-diagram_webstore

Jody Etienne | June 14, 2020

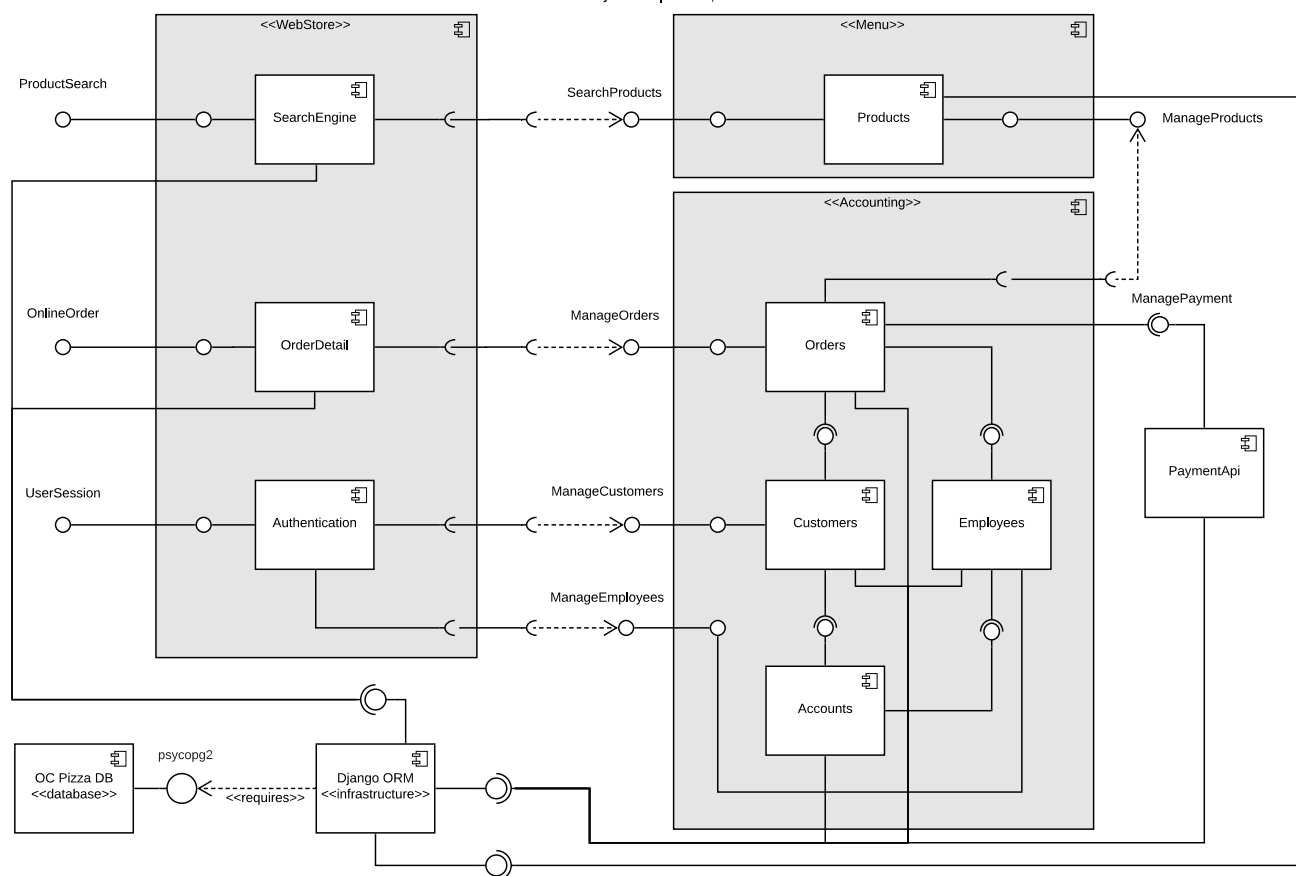


Figure 3: Diagramme de composant - site web - OC Pizza

Ce diagramme de composants a pour objectif d'illustrer la relation entre les différents composants du système applicatif lié au site web.

4.1.1 - Composants sous-système Webstore – site web

4.1.1.1 - Composant SearchEngine

Ce composant illustre la recherche d'éléments sur le site web.

Interface fournie : ProductSearch

Interfaces requises : SearchProducts par le biais du connecteur de délégation et le lien de dépendance et Persistance

4.1.1.2 - Composant OrderDetail

Ce composant illustre les détails de la commande sur le site web.

Interface fournie : OnlineOrder

Interfaces requises : ManageOrders par le biais du connecteur de délégation et le lien de dépendance et Persistance

4.1.1.3 - Composant Authentication

Ce composant illustre les moyens de s'authentifier sur le site web.

Interfaces fournie : UserSession

Interfaces requises : ManageEmployees et ManageCustomers par le biais du connecteur de délégation et le lien de dépendance

4.1.2 - Composants sous-système Menu – site web

4.1.2.1 - Composant Products

Ce composant illustre les produits sur le site web.

Interfaces fournies : ProductSearch, ManageProducts

Interface requise: Persistance

4.1.3 - Composants sous-système Accounting – site web

4.1.3.1 - Composant Orders

Ce composant illustre les commandes sur le site web.

Interface fournie : ManageOrders

Interfaces requises : ManageCustomers, ManageEmployees, ManagePayment et Persistance. ManageProducts par le biais du connecteur de délégation et le lien de dépendance.

4.1.3.2 - Composant Customers

Ce composant illustre les clients sur le site web.

Interface fournie : ManageCustomers

Interfaces requises : ManageAccounts et Persistance

4.1.3.3 - Composant Employee

Ce composant illustre les employés sur le site web.

Interfaces fournies : ManageEmployees

Interface requise : ManageAccounts et Persistance

4.1.3.4 - Composant Accounts

Ce composant illustre la comptabilité sur le site web.

Interfaces fournies : ManageAccounts

Interface requise : Persistance

4.1.3.5 - Composant Django ORM

Ce composant illustre la couche d'abstraction entre le monde objet et monde relationnel avec le site web.

Interface fournie : Persistance

Interface requise : psycopg2

4.1.3.6 - Composant OC Pizza DB

Ce composant illustre la base de données en lien avec le site web.

Interfaces fournie : psycopg2

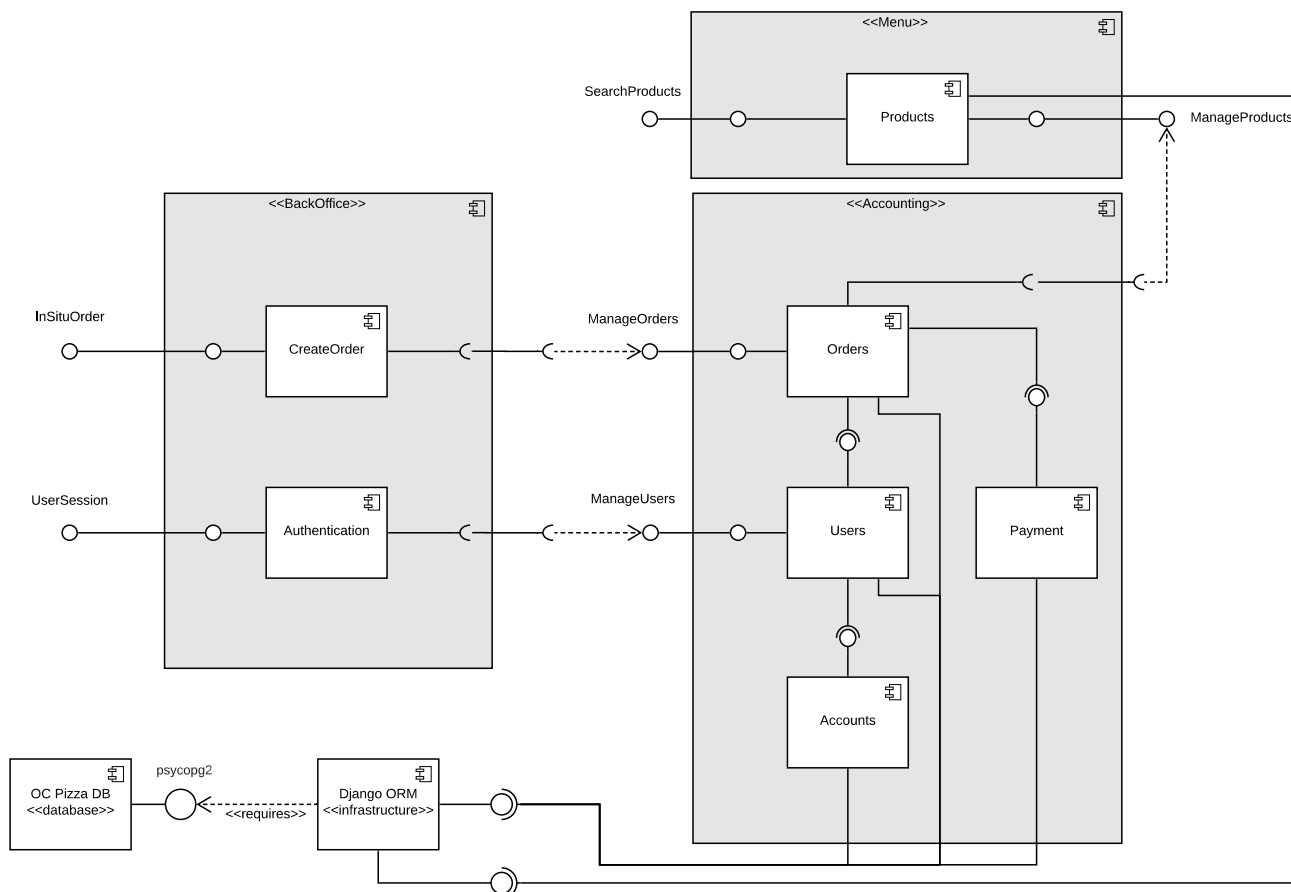


Figure 4: Diagramme de composant - restaurant - OC Pizza

Ce diagramme de composants a pour objectif d'illustrer la relation entre les différents composants du système applicatif liée au restaurant (point de vente physique).

4.1.4 - Composants sous-système Back-office - restaurant

4.1.4.1 - Composant CreateOrder

Ce composant illustre la création de commande sur le point de vente physique des restaurants.

Interface fournie : InSituOrder

Interface requise : ManageOrders par le biais du connecteur de délégation et le lien de dépendance

4.1.4.2 - Composant Authentication

Ce composant illustre les moyens de s'authentifier sur le point de vente physique des

restaurants.

Interface fournie : UserSession

Interface requise : ManageUsers par le biais du connecteur de délégation et le lien de dépendance

4.1.5 - Composants sous-système Menu – restaurant

4.1.5.1 - Composant Products

Ce composant illustre les produits sur le point de vente physique des restaurants.

Interfaces fournies : SearchProducts, ManageProducts

Interface requise : Persistance

4.1.6 - Composants sous-système Accounting – restaurant

4.1.6.1 - Composant Orders

Ce composant illustre les commandes sur le point de vente physique des restaurants.

Interface fournie : ManageOrders

Interfaces requises : ManageUsers, ManagePayment et Persistance. ManageProducts par le biais du connecteur de délégation et le lien de dépendance

4.1.6.2 - Composant Users

Ce composant illustre les utilisateurs sur le point de vente physique des restaurants.

Interface fournie : ManageUsers

Interfaces requises : ManageAccounts et Persistance

4.1.6.3 - Composant Payment

Ce composant illustre le paiement sur le point de vente physique des restaurants.

Interface fournie : ManagePayment

Interface requise : Persistance

4.1.6.4 - Composant Accounts

Ce composant illustre la comptabilité sur le point de vente physique des restaurants.

Interfaces fournies : ManageAccounts

Interface requise : Persistance

4.1.6.5 - Composant Django ORM

Ce composant illustre la couche d'abstraction entre le monde objet et monde relationnel avec le site web.

Interface fournie : Persistance

Interface requise : psygcop2

4.1.6.6 - Composant OC Pizza DB

Ce composant illustre la base de données en lien avec le site web.

Interfaces fournie : psycopg2

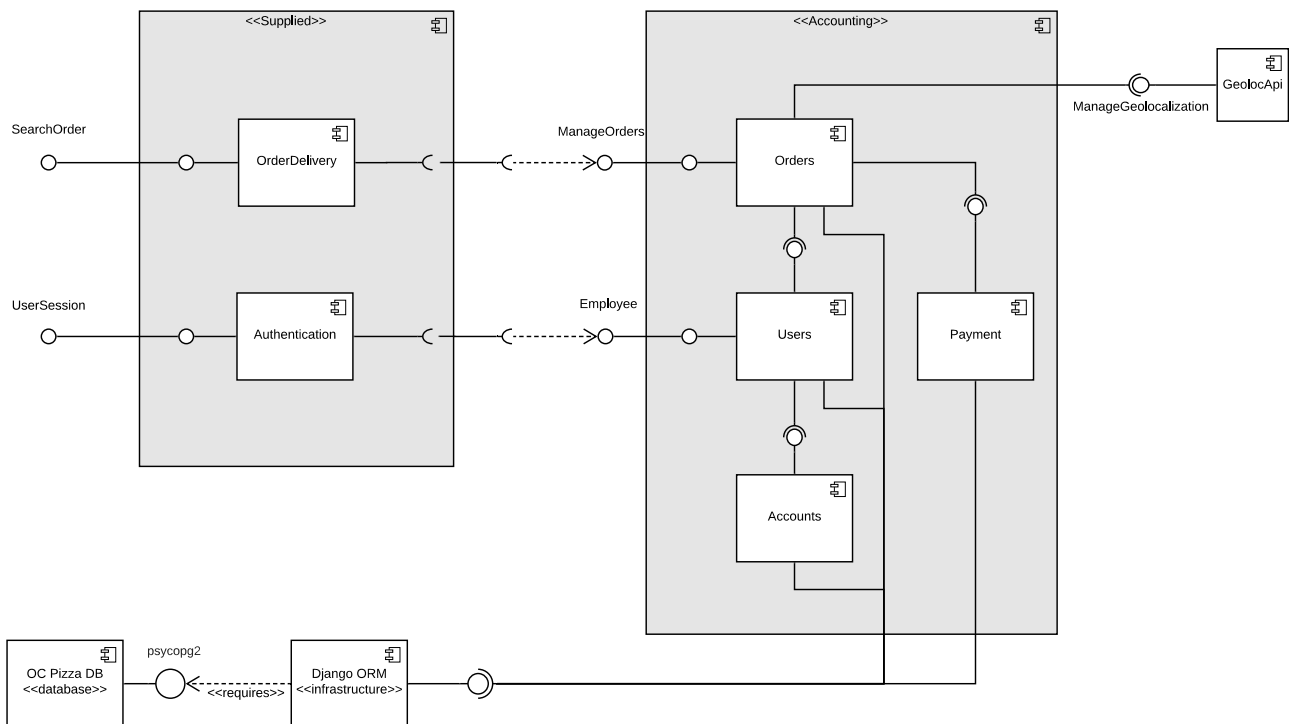


Figure 5: Diagramme de composant - livraison - OC Pizza

Ce diagramme de composants a pour objectif d'illustrer la relation entre les différents composants du système applicatif liée à la livraison.

4.1.7 - Composants sous-système Supplied - livraison

4.1.7.1 - Composant OrderDelivery

Ce composant illustre la livraison de commandes.

Interface fournie : SearchOrder

Interface requise : ManageOrders par le biais du connecteur de délégation et le lien de dépendance

4.1.7.2 - Composant Authentication

Ce composant illustre les moyens de s'authentifier en phase de la livraison de commandes.

Interface fournie : UserSession

Interfaces requise : Employee par le biais du connecteur de délégation et le lien de dépendance

4.1.8 - Composants sous-système Accounting – livraison

4.1.8.1 - Composant Orders

Ce composant illustre les commandes en phase de la livraison de commandes.

Interface fournie : ManageOrders

Interfaces requises : ManageUsers, ManagePayment, ManageGeolocalization et Persistance

4.1.8.2 - Composant Users

Ce composant illustre les utilisateurs en lien avec la livraison de commandes.

Interface fournie : Employee

Interfaces requises : ManageAccounts et Persistance

4.1.8.3 - Composant Payment

Ce composant illustre le paiement en phase de la livraison de commandes.

Interface fournie : ManagePayment

Interface requise : Persistance

4.1.8.4 - Composant Accounts

Ce composant illustre la comptabilité en phase de la livraison de commandes.

Interface fournie : ManageAccounts

Interface requise : Persistance

4.1.8.5 - Composant GeolocApi

Ce composant illustre l'API de géolocalisation en lien avec la livraison de commandes.

Interface fournie : ManageGeolocalization

4.1.8.6 - Composant Django ORM

Ce composant illustre la couche d'abstraction entre le monde objet et monde relationnel avec le site web.

Interface fournie : Persistance

Interface requise : psycpg2

4.1.8.7 - Composant OC Pizza DB

Ce composant illustre la base de données en lien avec le site web.

Interface fournie : psycopg2

5 - ARCHITECTURE DE DÉPLOIEMENT

PGESTpizza_05_deployment-diagram

Jody Etienne | June 13, 2020

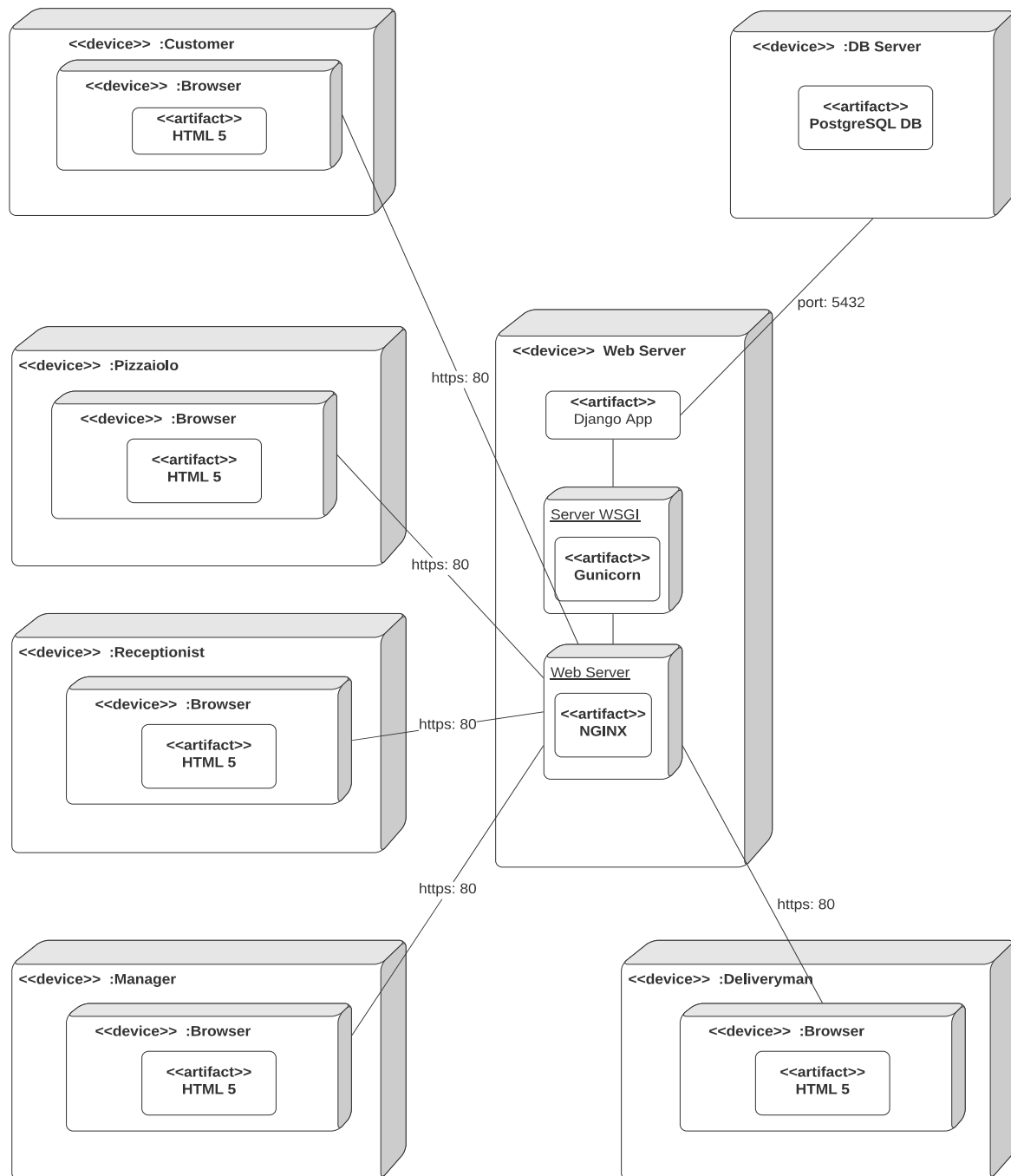


Figure 6: Diagramme de déploiement - OC Pizza

5.1 - Serveur de Base de données

PostgreSQL, ou Postgres, est un système de gestion de base de données relationnelle qui fournit une implémentation du langage de requête SQL. Il est très apprécié pour de nombreux projets, petits et grands, et présente l'avantage de respecter les normes et d'être doté de nombreuses fonctionnalités avancées telles que les transactions fiables et la simultanéité sans verrouillage en lecture.

La licence open source de PostgreSQL permet aux développeurs d'exploiter une ou plusieurs bases de données sans coût de licence dans leurs applications. Le modèle d'exploitation sous licence open source est beaucoup moins coûteux que celui d'Oracle ou d'autres bases de données propriétaires, d'autant plus que la réplication et le sharding deviennent nécessaires à grande échelle. En outre, comme un grand nombre de personnes, allant des développeurs indépendants aux organisations multinationales, utilisent PostgreSQL, il est souvent plus facile de trouver des développeurs ayant une expérience de PostgreSQL que d'autres bases de données relationnelles.

Caractéristiques techniques:

PostgreSQL : version 13

5.2 - Serveur web NGINX

NGINX est un serveur HTTP et un serveur proxy inverse, un serveur proxy de messagerie et un serveur proxy TCP/UDP générique, écrit à l'origine par Igor Sysoev. Depuis longtemps, il fonctionne sur de nombreux sites russes très chargés, dont Yandex, Mail.Ru, VK et Rambler. Selon Netcraft, nginx desservait ou servait de proxy à 25,62 % des sites les plus fréquentés en mai 2020. Voici quelques-unes de ses réussites : Dropbox, Netflix, Wordpress.com, FastMail.FM.

Caractéristiques de base du serveur HTTP :

- Servir des fichiers statiques et d'index, autoindexation ; cache de descripteurs de fichiers ouverts ;
- Reverse proxying accéléré avec mise en cache ; équilibrage de charge et tolérance aux pannes ;
- Prise en charge accélérée avec mise en cache des serveurs FastCGI, uwsgi, SCGI et memcached ; équilibrage de la charge et tolérance aux pannes ;
- Architecture modulaire. Les filtres comprennent le gzippage, les plages d'octets, les réponses en morceaux, le XSLT, le SSI et le filtre de transformation d'image. Plusieurs inclusions de SSI dans une même page peuvent être traitées en parallèle si elles sont gérées par des serveurs proxy ou FastCGI/uwsgi/SCGI ;

- Prise en charge des SNI SSL et TLS ;
- Prise en charge de HTTP/2 avec priorisation pondérée et basée sur les dépendances.

Caractéristiques techniques:

Nginx : version 1.19.2

5.3 - Serveur d'application GUNICORN

Gunicorn "Green Unicorn" est un serveur Python WSGI HTTP pour UNIX. Le serveur Gunicorn est largement compatible avec divers cadres de travail web, il est simplement mis en œuvre, léger en ressources serveur et assez rapide.

Le serveur WSGI (Web Server Gateway Interface) met en œuvre le côté serveur web de l'interface WSGI pour exécuter des applications web en Python.

Pourquoi utiliser le WSGI et ne pas simplement pointer un serveur web directement sur l'application ?

Le WSGI nous donne de la flexibilité. Les développeurs d'applications peuvent remplacer les composants de la pile web par d'autres. Par exemple, un développeur peut passer de Green Unicorn à uWSGI sans modifier l'application ou le cadre qui met en œuvre le WSGI.

Les serveurs WSGI favorisent la mise à l'échelle. Servir des milliers de demandes de contenu dynamique en même temps est le domaine des serveurs du WSGI, et non des frameworks. Les serveurs WSGI traitent les demandes provenant du serveur web et décident de la manière de communiquer ces demandes au processus d'un cadre d'application. La séparation des responsabilités est importante pour une mise à l'échelle efficace du trafic web.

Caractéristiques techniques:

Gunicorn : version 20.0.4

5.4 - Serveur d'hébergement

L'hébergement est de type serveur privé virtuel (VPS, pour Virtual Private Server). Il s'effectue avec **OVH** avec la formule **VPS Comfort**, une entreprise française spécialisée dans les services de cloud computing, avec l'avantage de pouvoir dédier plus ou moins de ressources selon les besoins du système (scalability).

Emplacement du datacenter: **Gravelines, France**

Offre : **Comfort**

Processeur : **4 vCore**

Mémoire : **8 Go**

Stockage : **160 Go SSD NVMe**

Bande passante publique : **1 Gbit/s**

Image : **Ubuntu 20.04**

6 - POINTS PARTICULIERS

6.1 - Gestion des logs

La gestion des logs sera gérée par **Sentry** en relevant différents évènements liés à la plateforme de OC Pizza.

6.2 - Ressources

Les ressources graphiques de l'interface ainsi que les données brutes telles que les tarifs, les recettes, les coordonnées intégrées à la base de données sont fournies par le client OC PIZZA.

6.3 - Environnement de développement

L'application web est développée avec **Django**. Cet environnement de développement est le plus complets dans sa catégorie.

La future modification des fichiers pourra également être effectuée à l'aide d'éditeurs de texte tels que **Visual Studio Code**, **SublimeText** ou **Atom**.

6.4 - Procédure de packaging / livraison

L'application web sera déployée sur un serveur privé virtuel **OVH** avec la formule **VPS Comfort**. Elle sera également remise au client OC PIZZA sous forme de dossier de fichiers sources pour les futures mises à jour et modifications.

L'application web sera fournie sous forme de dossier de fichiers sources directement installable sur tous les appareils compatibles.

Les liens **URL** de téléchargement des dossiers sont indiqués dans le PV de livraison.

7 - GLOSSAIRE

SQL	SQL (sigle de Structured Query Language, en français langage de requête structurée) est un langage informatique normalisé servant à exploiter des bases de données relationnelles. La partie langage de manipulation des données de SQL permet de rechercher, d'ajouter, de modifier ou de supprimer des données dans les bases de données relationnelles.
API	Interface de programmation d'application par laquelle un logiciel offre des services.
UML	Unified Modeling Language, un langage graphique de modélisation des données et des traitements.
Diagramme	En langage UML, les diagrammes sont dépendants hiérarchiquement et se complètent, de façon à permettre la modélisation d'un projet pendant tout son cycle de vie.
WSGI	La Web Server Gateway Interface (WSGI) est une spécification qui définit une interface entre des serveurs et des applications web pour le langage Python.
Proxy	Un proxy est un composant logiciel informatique qui joue le rôle d'intermédiaire en se plaçant entre deux hôtes pour faciliter ou surveiller leurs événements.
Http	HTTP est un protocole de la couche application. Il peut fonctionner sur n'importe quelle connexion fiable, dans les faits on utilise le protocole TCP comme couche de transport. Un serveur HTTP utilise alors par défaut le port 80 (443 pour HTTPS).
Frameworks	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 (architecture).
SNI	SNI (Server Name Indication) est une méthode qui permet de configurer plusieurs certificats SSL sur un seul serveur et pour la même adresse IP.
SSI	La sécurité des systèmes d'information (SSI) ou plus simplement sécurité informatique, est l'ensemble des moyens techniques, organisationnels, juridiques et humains nécessaires à la mise en place de moyens visant à empêcher l'utilisation non autorisée, le mauvais usage, la modification ou le détournement du système d'information.