

OC Pizza

Développement D'un Système de Gestion Des Opérations des Pizzerias OC Pizza

Dossier de conception technique

Version 0.1

Auteur
Fabrice Jaouën
Stagiaire OC Pizza

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 - Règles de gestion	5
3.2 - Package X	Erreur ! Signet non défini.
4 - Architecture Technique	11
4.1 - Application Web.....	11
4.1.1 - Composants X.....	11
4.1.2 - Composants Y et Z	12
5 - Architecture de Déploiement	14
5.1 - Serveur de Base de données	14
5.2 - Serveur Unicorn	14
5.3 - Serveur Unicorn	15
6 - Glossaire	16

1 - VERSIONS

Auteur	Date	Description	Version
Fabrice Jaouën	13/06/2020	Création du document	0.1

2 - INTRODUCTION

2.1 - Objet du document

Le présent document constitue le dossier de conception technique du système d'information relatif à la gestion des opérations du groupe OC Pizza

L'objectif de ce document est de faciliter le travail de développement et de déploiement de ce système de d'information entre les différentes parties prenantes, du côté du prestataire, IT-C&D, mais aussi du client, OC Pizza.

Les éléments du présent dossier découlent :

- De l'étude du besoin exprimé par le client et ses collaborateurs ;
- De l'analyse des systèmes d'information utilisés par d'autres groupes de vente de pizzas à emporter ;
- Du travail commun entre les spécialistes d'IT C&D et ses prestataires habituels, notamment pour le déploiement des différents modules.

Customer – DeliveryAddress :

Un client peut avoir plusieurs adresses de livraisons (son domicile, ou le bureau par exemple), mais une adresse ne peut correspondre qu'à un seul client, ce qui permet d'éviter une erreur de livraison.

DeliveryAddress – PurchaseOrder :

Une commande peut être récupérée en point de vente, ou en boutique. En revanche, une commande doit spécifier une et une seule adresse de livraison ou de récupération (si c'est en point de vente).

Customer – PurchaseOrder :

Une commande ne peut avoir qu'un et un seul ordonnateur. Mais un client peut passer plusieurs commandes au fil du temps.

PurchaseOrder – ItemsInOrder :

A une commande correspondent un ou plusieurs items du menu. Cette relation permet de calculer le montant total d'une commande.

PurchaseOrder – OrderStatus :

Une seule et même commande ne peut avoir qu'un seul et même statut et réciproquement. Cette distinction entre PurchaseOrder et OrderStatus permet de gérer l'interface avec le package Point of Sales Operations.

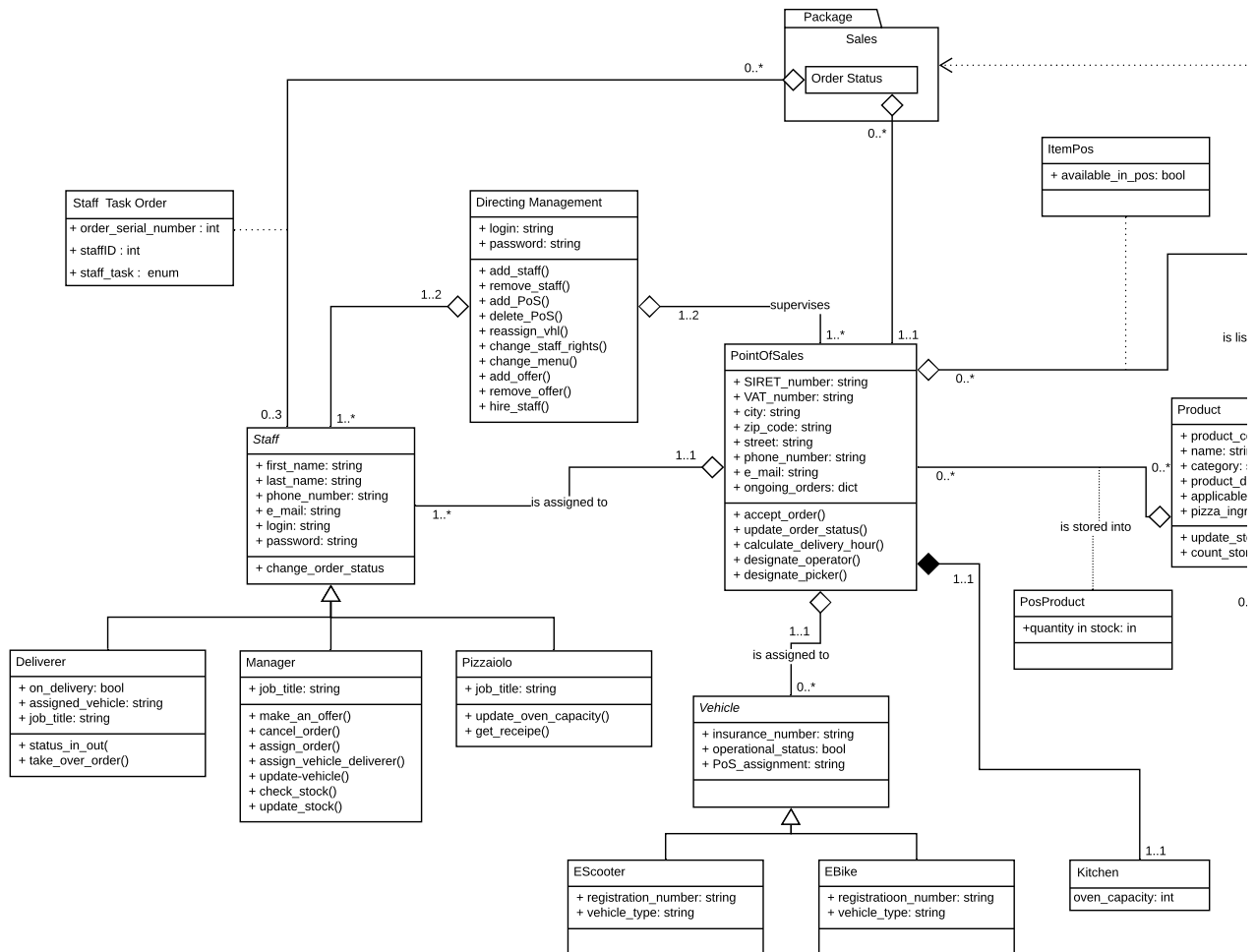
PurchaseOrder – Payment :

Un paiement doit être précédé d'une et une seule commande. Il peut arriver qu'une commande ne soit pas réglée par un client indélicat. La relation entre les deux classes doit pouvoir prendre en compte un tel incident.

OrderStatus – Payment :

L'unicité de la relation réciproque de la relation entre les deux classes permet de gérer plus clairement l'interface avec le système de paiement par une séparation des rôles, la classe Payment étant chargée des relations avec les organismes de paiement, la caisse et la comptabilité.

3.1.2 - Règles de gestion des classes liées au Staff (Package PoS Operations)



L'ensemble du diagramme est accessible sur LucidChart avec le lien suivant : [LIEN](#).

Il s'agit de décrire ici la gestion des ressources du point de vente en la séparant autant que possible de la partie relative aux produits alimentaires et de la partie Ventes, qui, elle, fait partie du package Sales.

DirectingManagement – PointOfSales :

Le Directing Management supervise plusieurs points de vente, mais chaque point de vente ne rapporte qu'à 1, voire 2 instances de Directing Management. Il s'agit ici essentiellement de reporting de l'activité. Il n'y a donc pas d'autre dépendance que la capacité à accéder aux données.

DirectingManagement – Staff :

La direction du groupe OC Pizza n'a pas vocation à interagir avec les points de vente autrement qu'en supervisant son activité ou en donnant des directives. Cette classe n'intervient donc pas sur les opérations et les instances créées dans ce contexte. C'est pourquoi elle n'a pas de relations avec une quelconque cardinalité.

Staff – Deliverer, Manager, Pizzaiolo :

La relation est celle d'une classe parent avec ses classes enfants. La classe Staff regroupe toutes les caractéristiques communes à chacun des trois types de poste. Les classes enfants que sont Deliverer, Manager et Pizzaiolo peuvent ainsi évoluer séparément avec le temps si les circonstances l'exigent.

Staff est une classe abstraite, puisque chaque collaborateur a vocation à occuper un poste dès la signature de son contrat.

PointOfSales – Vehicle :

Un véhicule est assigné à un seul point de vente, tandis qu'un point de vente peut posséder plusieurs véhicules. La mutualisation de la flotte entre plusieurs points de vente n'est pas envisagée, puisqu'il ne s'agit pas là d'un mode de fonctionnement courant. En cas de fermeture provisoire d'un point de vente, le véhicule est affecté à un autre.

Vehicle – Escoter, EBike :

Ces trois classes sont dans une relation de type parent – enfant pour permettre une gestion différenciée de la flotte de moyens de transport.

Vehicle est une classe abstraite.

PointOfSales – Kitchen :

Chaque point de vente dispose d'une et une seule cuisine. Comme elle relève d'une réglementation particulière et figure dans les amortissements, il est pertinent de la gérer à part.

PointOfSales – Staff :

Nous sommes ici dans une relation de 1 à plusieurs. Un membre du staff est salarié d'un et un seul point de vente, tandis qu'un point de vente emploie plusieurs salariés.

Le modèle choisi ici est donc celui d'une large autonomie des points de vente dans le recrutement et la gestion des employés, c'est-à-dire un modèle de type franchise. Si le client venait à demander à gérer des points de vente en propre pour introduire plus de flexibilité dans la gestion des salariés, une table de jointure serait à créer pour permettre à un salarié de travailler dans plusieurs points de vente.

Staff – OrderStatus et StaffTaskOrder :

Un membre du staff peut avoir à gérer plusieurs commandes simultanément, dès lors qu'une tâche lui est attribuée. De même, une commande est prise en compte par plusieurs personnes : comme le pizzaiolo est à l'œuvre sur la partie arrière, un deuxième acteur est indispensable.

C'est pourquoi une table de jointure StaffTaskOrder est créée : elle permet la gestion de cette relation de plusieurs à plusieurs.

PointOfSales – Product et PosProduct.

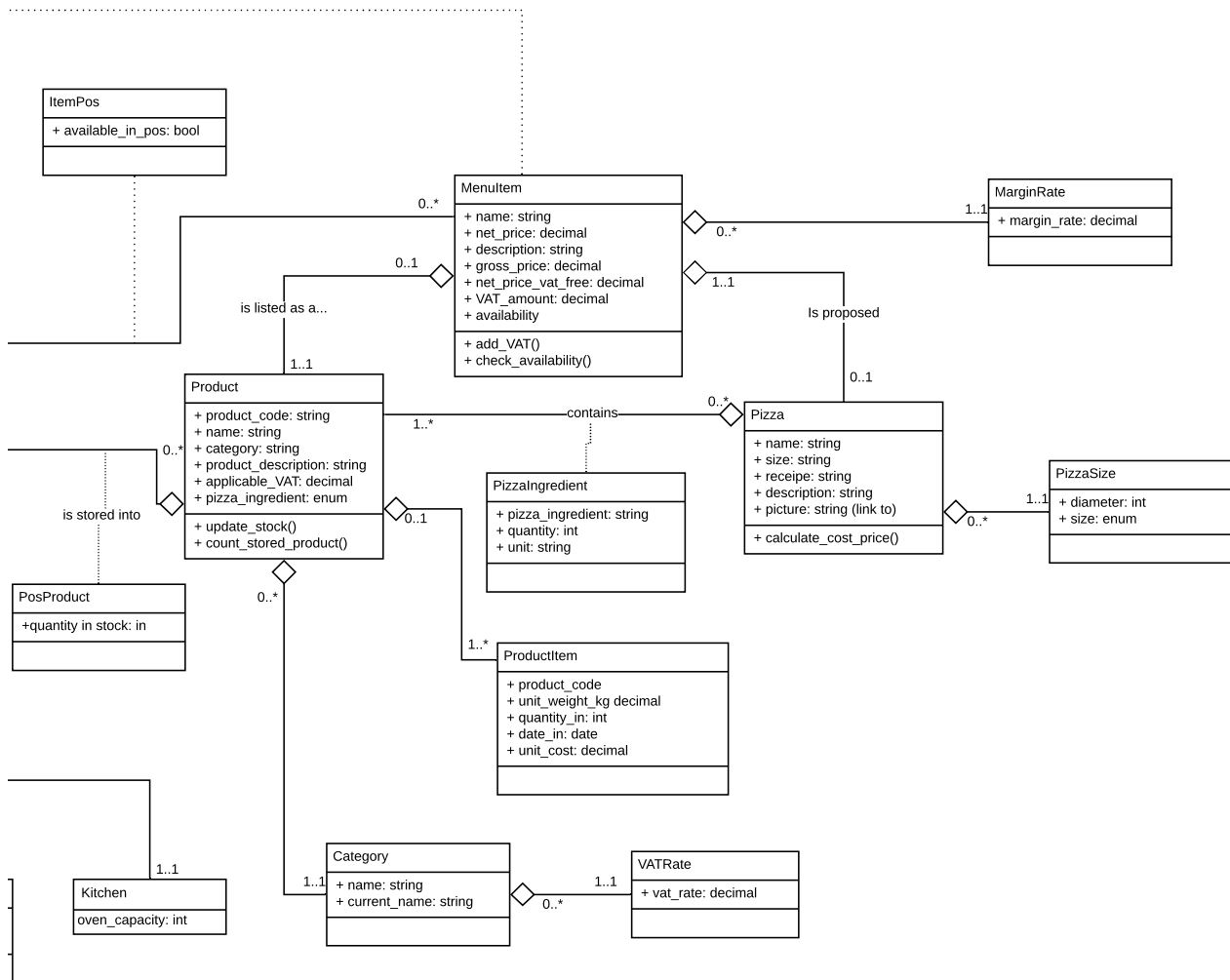
La chaîne OC Pizza a défini une liste de produits entrant dans la composition de sa carte. Il peut s'agir de denrées nécessaires à la préparation des pizzas, comme de produits destinés à la consommation. Par conséquent : un même produit est nécessairement utilisé par plusieurs points de vente, tandis qu'un point de vente s'appuie sur une gamme de produits pour fonctionner.

C'est pourquoi une table de jointure PosProduct est en place pour permettre d'isoler le stock de chaque magasin.

PointOfSales – OrderStatus :

Un point de vente manage plusieurs commandes à la fois, tandis qu'une commande ne peut être prise en charge que par un seul point de vente. La liaison avec le package Sales est essentiellement destiné à la prise en charge d'une commande, le suivi et l'actualisation de son statut ainsi que son paiement.

3.1.3 - Règles de gestion des classes liées au Product (Package PoS Operations)



Menuitem – Product :

Un Product est soit un produit fini, prêt à la vente, soit un ingrédient de pizza. Cette classification correspond à la liste des références de la chaîne OC Pizza.

Menuitem correspond à la carte des produits proposés à la vente dans la chaîne OC Pizza. Chaque produit n'apparaît qu'une seule fois dans la carte et chaque item de la carte est soit un Product, soit une Pizza. C'est ici que s'effectuent tous les calculs de prix des produits et de pizzas par la prise en compte de la marge et de la TVA.

MenuItem – PointOfSales et ItemPos :

Un item de la carte est proposé dans plusieurs pizzérias et chaque pizzeria offre une carte de produits disponibles à la vente. Cette relation de plusieurs à plusieurs est résolue par la ItemPos qui permet de gérer les ruptures d'approvisionnement d'un produit dans une pizzeria spécifique.

MenuItem – MarginRate :

Nous sommes dans une relation de type un à plusieurs, puisqu'un item de la carte ne peut avoir qu'un taux de marge, tandis que le même taux de marge s'applique à plusieurs produits.

MenuItem – Pizza :

Chaque type de pizza est proposé dans la carte. Nous sommes donc dans une relation de un à plusieurs, puisque chaque pizza n'est présente qu'une fois dans la carte et mais la carte comporte évidemment plusieurs pizzas.

A noter, la taille de la pizza n'étant qu'un paramètre, elle est choisie par le menu et son prix est recalculé en proportion de sa taille.

Product – Pizza et PizzaIngredient :

La relation est de type plusieurs à plusieurs puisque chaque pizza est composée plusieurs ingrédients de la classe Product, tandis que chaque instance de Product peut être présent dans plusieurs recettes de Pizza. Par conséquent, PizzaIngredient est la table de jointure.

Pizza – PizzaSize :

Cette relation de un à plusieurs (plusieurs pizzas ont la même taille et une pizza n'a qu'un seul diamètre) permet choisir la taille d'une pizza et d'en actualiser la quantité d'ingrédients pour le pizzaïolo. A noter que la taille standard est la taille « normale » à partir de laquelle les proportions et le prix sont entrés par défaut.

Product – ProductItem :

Il s'agit d'une relation de 1 à plusieurs : un élément du stock a une seule référence de produit. En revanche il est complété par une date d'entrée qui permet ensuite au point de vente de décider s'il applique la méthode LIFO ou FIFO.

Product – Category :

Relation de un à plusieurs pour permettre le calcul de la TVA, chaque produit référencé se voyant attribuer une catégorie.

Category – VATRate :

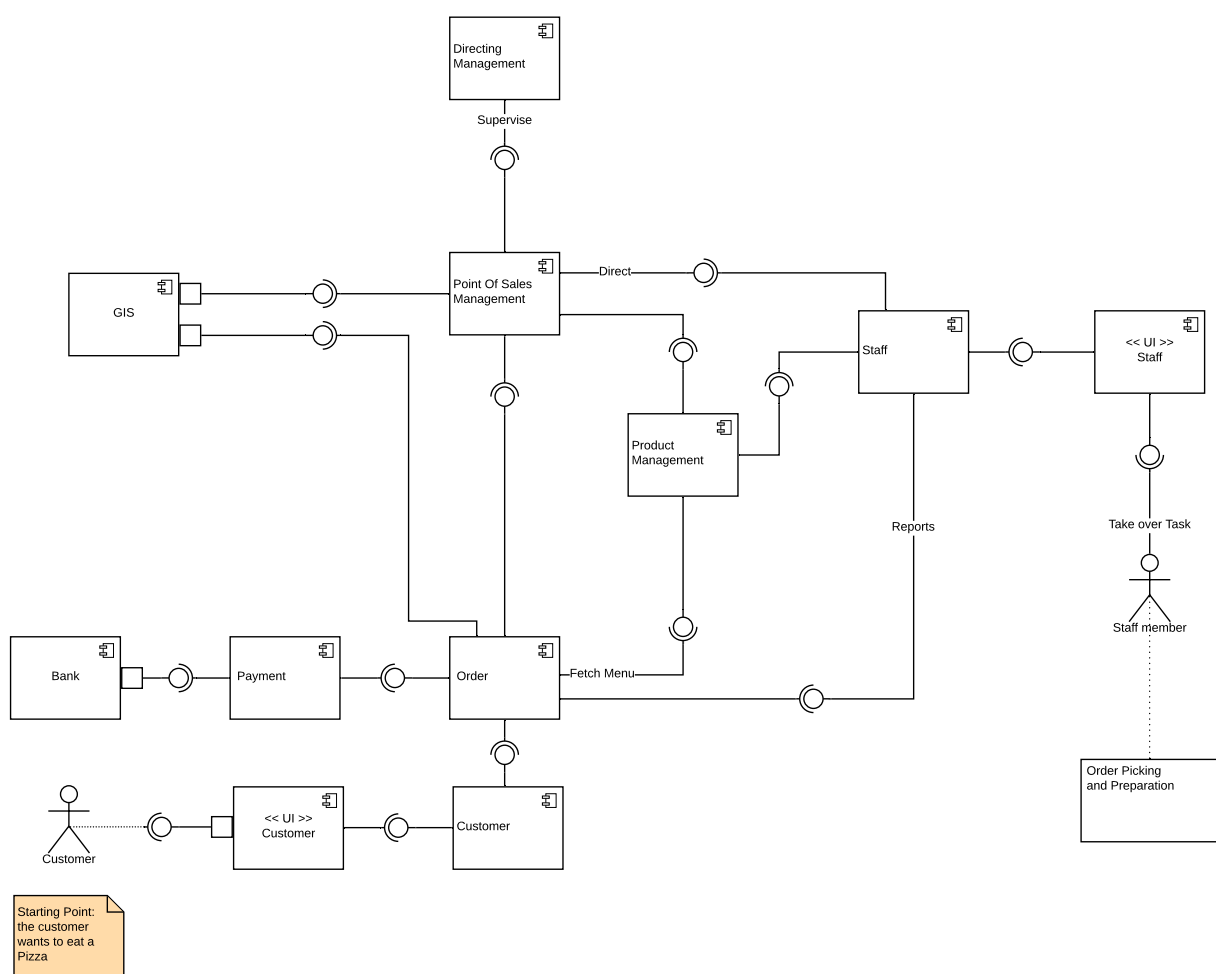
Nous sommes dans une relation de 1 à plusieurs. Les taux de TVA s'appliquent légalement à telle ou telle catégorie, le même taux pouvant être utilisé pour différentes catégories d'aliments.

4 - ARCHITECTURE TECHNIQUE

4.1 - Application Web

La pile logicielle est la suivante :

- Application **Python 3.8**
- Serveur d'application **Gunicorn 20.0.4**
- Serveur Web **Nginx 1.18.8**
- **MySQL 8.0.19**



Le diagramme de composants est accessible sur LucidChart avec le lien suivant : [LIEN](#)

4.1.1 - Composant Customer

Customer permet au client de s'enregistrer chez OC Pizza et ensuite d'ouvrir une session pour passer une commande.

Ce composant fait l'objet d'une attention particulière dans le développement de l'application en raison des contraintes légales portant sur la protection des données personnelles.

4.1.2 - Composant Payment et Bank

Payment permet de communiquer avec le logiciel de caisse et la banque pour suivre les transactions. Bien que n'étant pas inclus dans le système à proprement parler, *Bank* est considéré comme un composant en raison de son rôle central dans la gestion des paiements : son interface joue un rôle essentiel dans le suivi à la fois des encaissements et de la comptabilité d'OC Pizza.

4.1.3 - Composant Order

Order contient les classes permettant de gérer une commande, qu'il s'agisse de la constitution de la commande (Items In Order) ou du suivi de la commande (Order Status).

Il joue un rôle central dans le projet par ses interactions avec les composants *Payment*, *Product Management*, *Point of Sales Management* et *Staff*.

4.1.4 - Composant Point of Sales Management

Point of Sales Management prend en compte la gestion des ressources et le suivi de l'activité des points de vente.

Composant Product Management

Product Management prend en charge le suivi de la mercuriale, la constitution de la carte et le suivi des stocks, la constitution des recettes, entre autres. Ses utilisateurs sont donc les employés comme les clients, les premiers au début de la chaîne de production, les derniers en consommateurs.

4.1.5 - Composant Staff

Staff a vocation à structurer la gestion du personnel. Il est séparé du composant *Customer*, car tous deux relèvent de logiques différentes dans l'utilisation du système. Ce composant a vocation à s'interfacer ultérieurement avec un SIRH, au gré de l'expansion d'OC Pizza ou de l'évolution de certaines fonctions comme la paie.

4.1.6 - Composant GIS

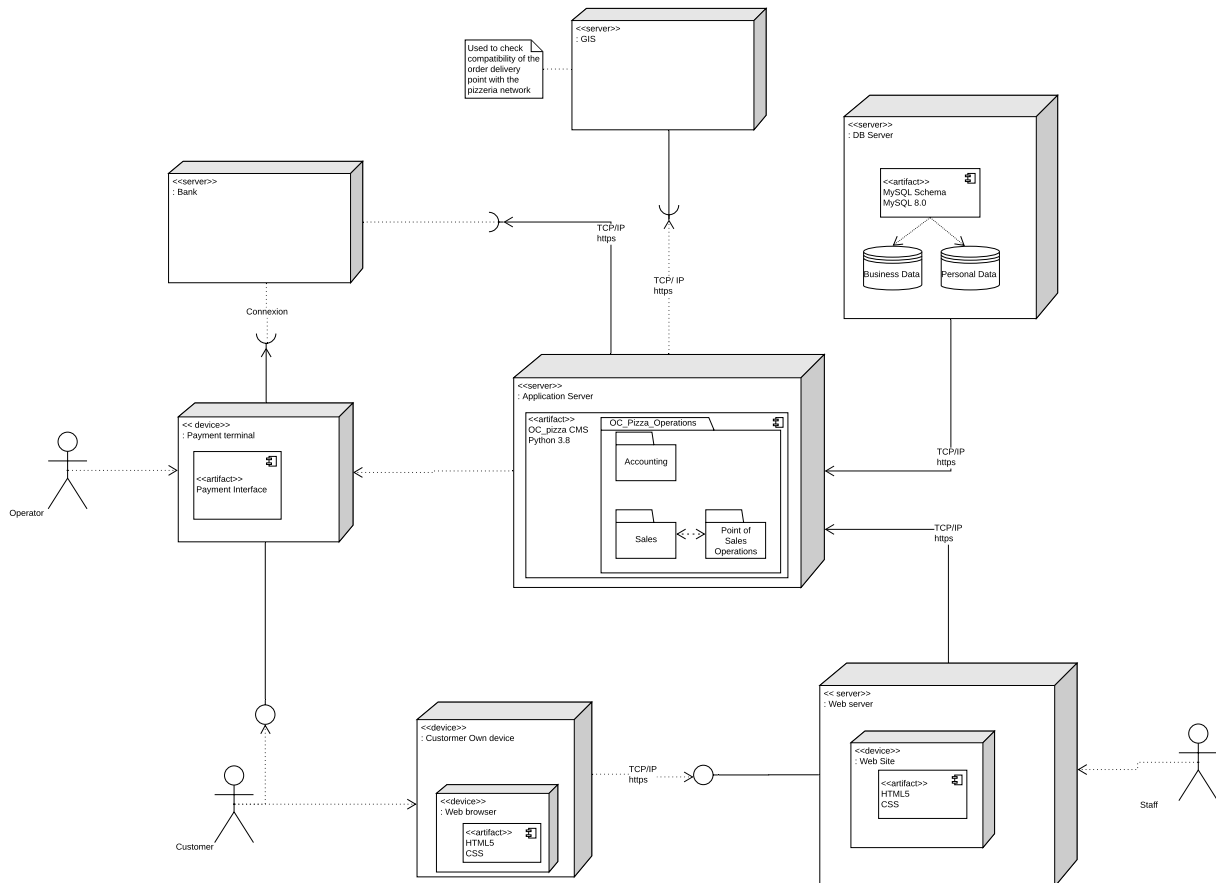
GIS est un composant de tierce partie. Il est destiné à vérifier la compatibilité des souhaits de livraison avec la couverture géographique des points de vente d'OC Pizza. En effet, pour un maillage plus précis, il s'agit avec ce composant d'estimer les temps de trajets entre le point de vente et le souhait du client pour déterminer l'éligibilité de la commande.

4.1.7 - Composant Directing Management

La particularité de *Directing Management*, rattaché à ce composant, est qu'il n'intervient pas sur les

opérations des points de vente, mais a une fonction de contrôle de l'activité de l'ensemble du groupe.

5 - ARCHITECTURE DE DÉPLOIEMENT



Explication / commentaires si nécessaires...

5.1 - Serveur MySQL

MySQL 8.0.19

Caractéristiques techniques : Hébergement privé pour s'assurer de l'utilisation des fonctionnalités de MySQL 8.0 (OVH est sur MySQL 5.7)

Faire une partition des données clients et des données business.

Intégrer la sécurité des authentifications.

5.2 - Serveur Gunicorn

Gunicorn 20.0.4

Déployer également Django comme WSGI.

Installation recommandée avec Nginx.

5.3 - Serveur Nginx

5.4 - Serveur GIS

Utiliser Google Maps Web Services et son API.

6 - GLOSSAIRE

Amortissements	Extinction graduelle d'une dette. Tous les biens d'une valeur supérieure à 500€ doivent être listés.
Mercuriale	Liste des produits référencés dans un restaurant.
WSGI	Web Server Gateway Interface.