

SPECIFICATIONS

TECHNIQUES



IPIZZA

IT Consulting & Development

Table des matières

1. Versions	3
2. Cadre du projet	3
2.1. Object du document	3
2.2 Contexte de l'entreprise	3
2.3 Besoins	4
2.4. Planning Prévisionnel	4
3. Domaine fonctionnel	4
3.1. Description	4
3.2. Diagramme des classes	4
3.3. Description des classes	6
3.4. Diagramme des composants	12
3.5. Description des composants	13
4. Model physique des données	15
4.1. Diagramme MPD	15
4.2. Description des tables	16
5. L'architecture de déploiement	17
5.1. Diagramme de déploiement	17
5.2. Description du diagramme	18
6. Conclusion	19

1. Versions

Version	Auteur	Date	Description
0.1	M. JOUDAR	13/05/2022	Création du document
0.2	M. JOUDAR	16/06/2022	Réajustement du domaine fonctionnel
0.3	M. JOUDAR	20/06/2022	Réajustement de l'architecture de déploiement
1.0	M. JOUDAR	23/06/2022	Mise en forme finale

2. Cadre du projet

2.1. Object du document

Ce document présente les spécifications techniques concernant le projet de conception d'un système informatique pour l'ensemble des pizzerias du groupe OC Pizza.

La rédaction de ce document est basé sur les différents échanges avec les commanditaires du projet, Franck et Lola, gérants de la pizzeria OC Pizza.

2.2 Contexte de l'entreprise

« OC Pizza » est un jeune groupe de pizzeria en plein essor. Créé par Franck et Lola, le groupe est 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 6 mois. Le système informatique actuel ne correspond plus aux besoins du groupe car il ne permet pas une gestion centralisée de toutes les pizzerias. De plus, il est très difficile pour les responsables

de suivre ce qui se passe dans les points de ventes. Enfin, les livreurs ne peuvent pas indiquer « en live » que la livraison est effectuée.

2.3 Besoins

Le système devra permettre :

- D'être plus efficace dans la gestion des commandes, de leur réception à leur livraison en passant par leur préparation ;
- De suivre en temps réel les commandes passées, en préparation et en livraison ;
- De suivre en temps réel le stock d'ingrédients restants pour savoir quelles pizzas peuvent encore être réalisées ;
- De proposer un site Internet pour que les clients puissent :
 - Passer leurs commandes, en plus de la prise de commande par téléphone ou sur place;
 - Payer en ligne leur commande s'ils le souhaitent – sinon, ils paieront directement à la livraison ;
 - Modifier ou annuler leur commande tant que celle-ci n'a pas été préparée.
- De proposer un aide-mémoire aux pizzaiolos indiquant la recette de chaque pizza.

2.4. Planning Prévisionnel

L'application doit être opérationnelle pour l'ouverture des 3 nouvelles pizzerias, dans 6 mois, soit pour le 2 Janvier 2023.

3. Domaine fonctionnel

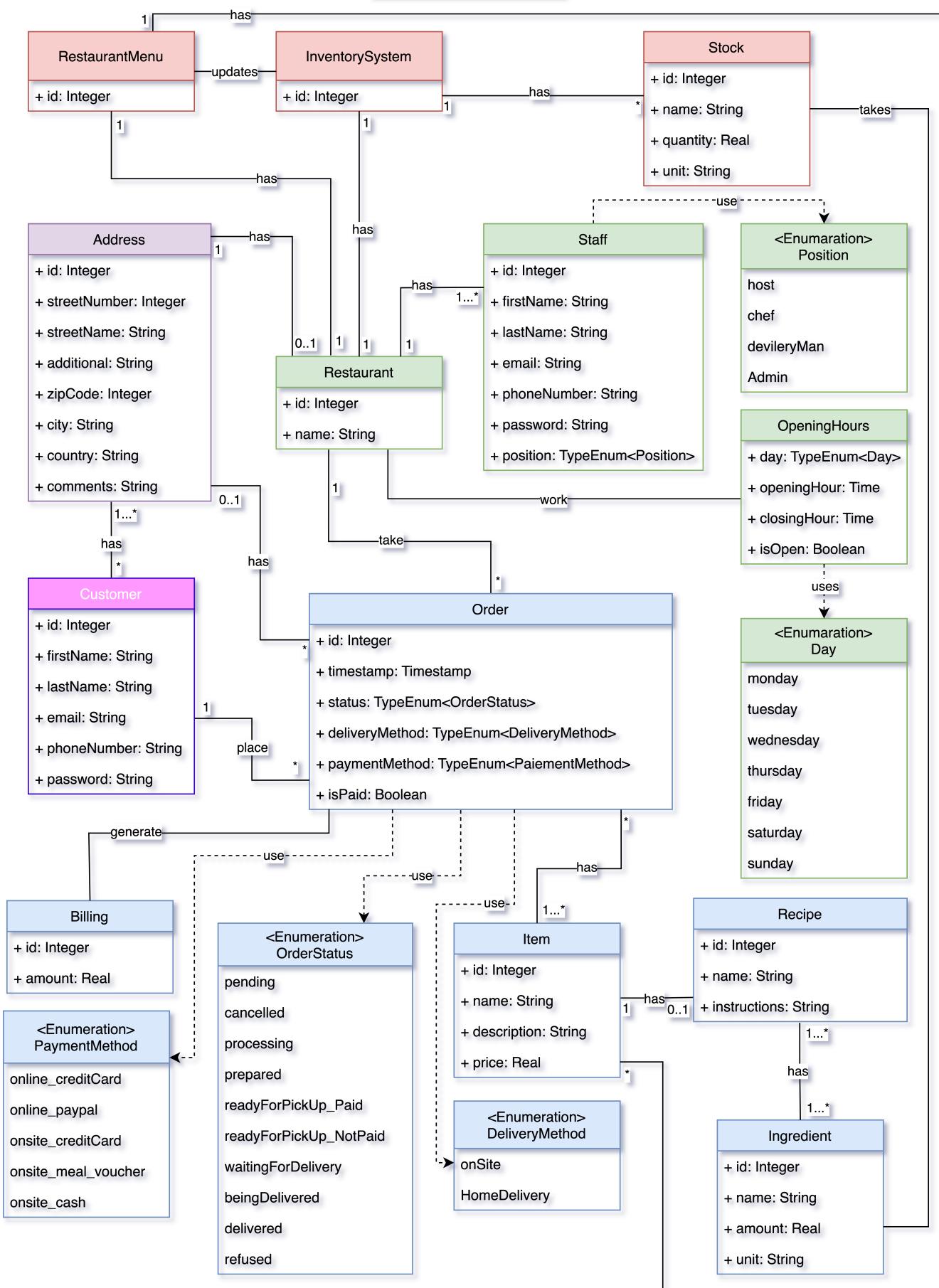
3.1. Description

Pour la représentation du domaine fonctionnel, une approche orientée objet a été utilisée en respectant les normes UML.

Le diagramme des classes ci-dessous, servira de base à la modélisation du domaine fonctionnel de ce projet. Il permet de mettre en place la structure du système, de concevoir les variables nécessaires au fonctionnement de celui-ci, et d'initialiser le modèle physique de données.

3.2. Diagramme des classes

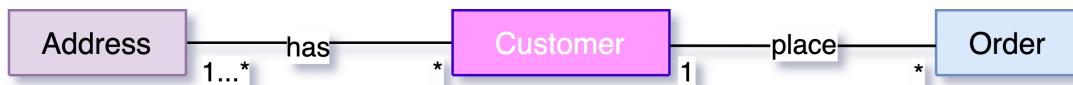
Diagramme des classes



3.3. Description des classes

Customer

Cette classe regroupe les attributs associés à un client du restaurant.

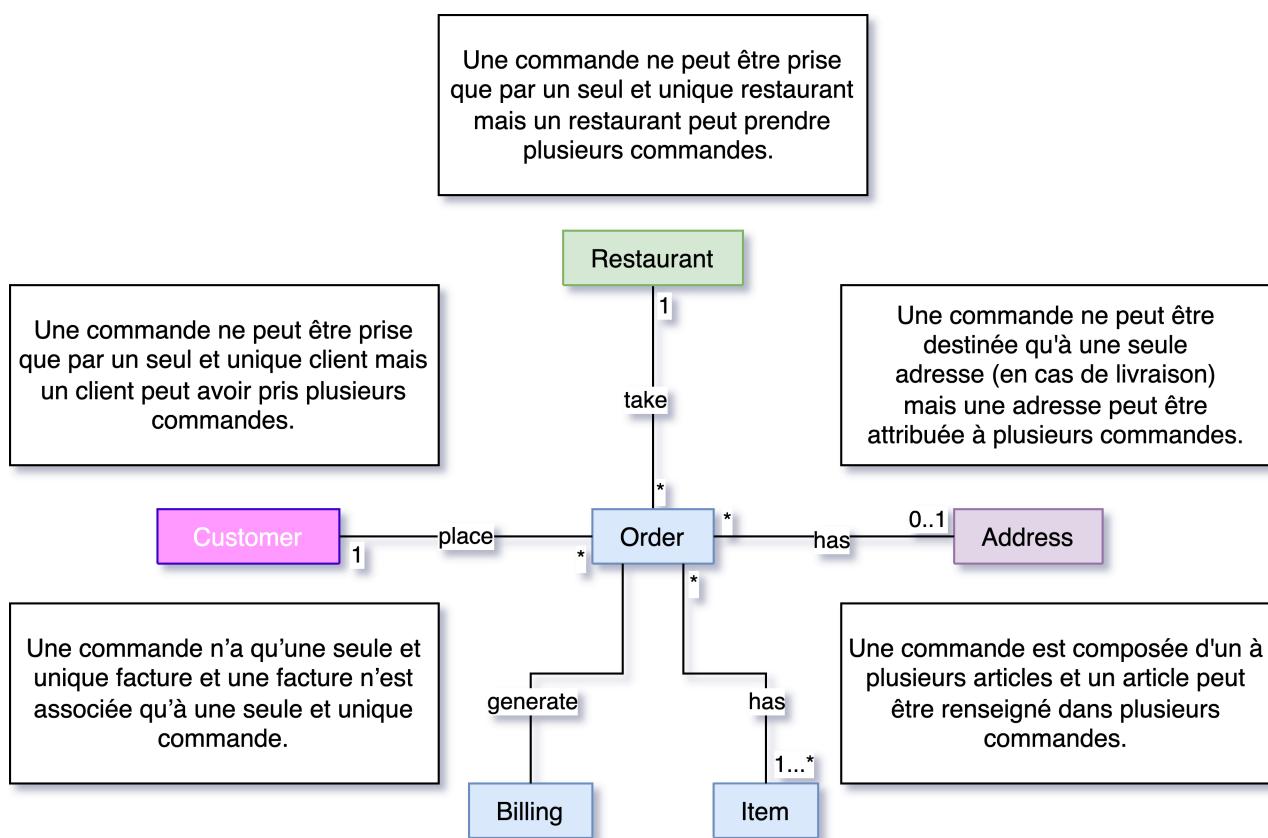


Une adresse peut appartenir à plusieurs clients (eg. membres d'une famille, colocataires) et un client peut avoir une à plusieurs adresses (eg. domicile, travail).

Une commande ne peut être prise que par un seul et unique client mais un client peut avoir pris plusieurs commandes.

Order

Cette classe regroupe tous les attributs associés à une commande effectuée.



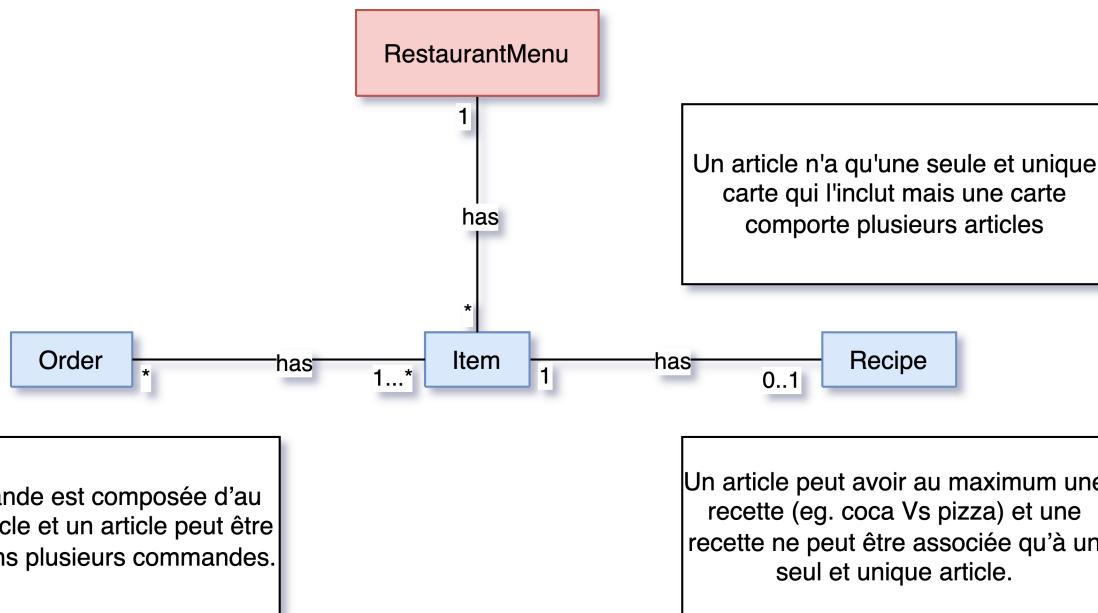
Billing

Cette classe regroupe tous les attributs associés à une facturation.



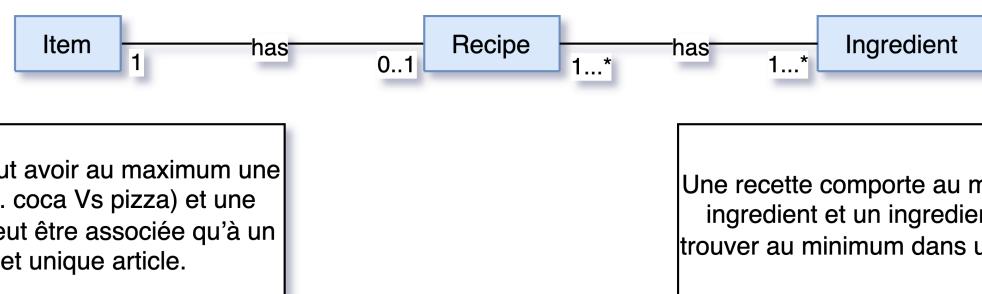
Item

Cette classe regroupe les attributs spécifiques d'un article proposé (eg. pizza, salade, boisson...).



Recipe

Cette classe regroupe tous les attributs associés à une recette.



Ingredient

Cette classe regroupe les attributs spécifiques d'un ingrédient.



Une recette comporte au minimum un ingrédient et un ingrédient doit se trouver au minimum dans une recette.

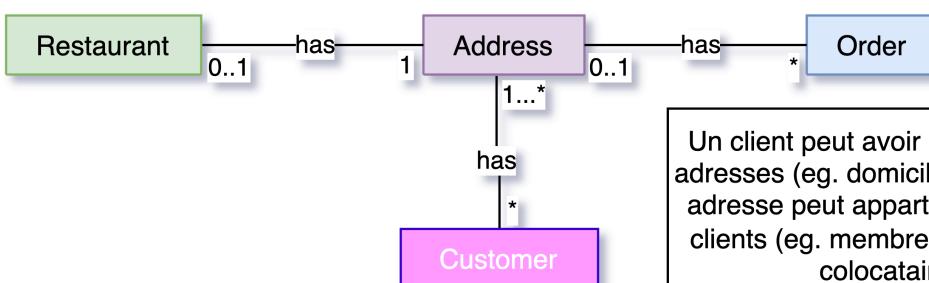
L'ingredient fait partie du stock, c'est une micro quantité du stock.

Address

Cette classe regroupe les attributs des adresses des clients, de livraison et des restaurants.

Un restaurant ne peut être localisé qu'à une seule et unique adresse et une adresse ne peut être attribuée qu'à un seul restaurant.

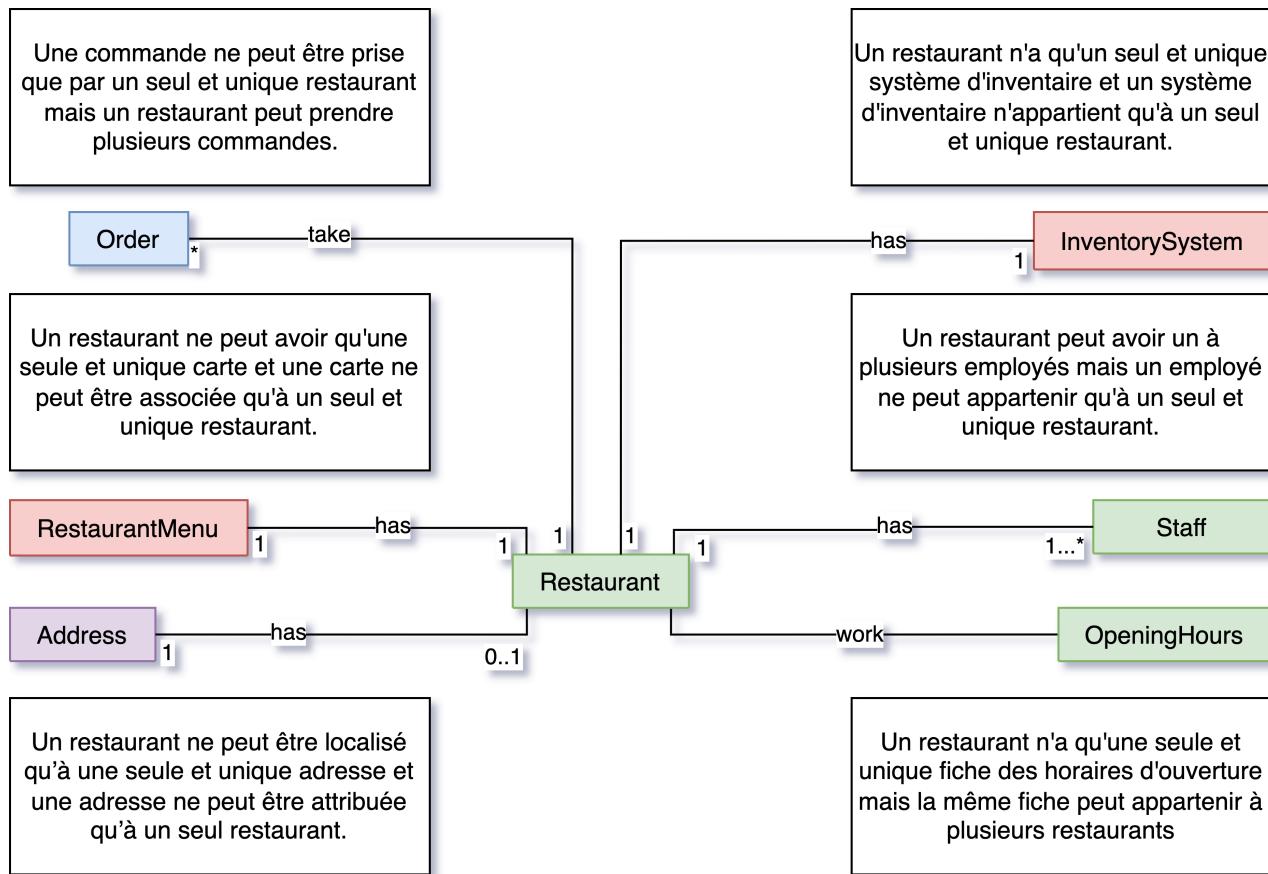
Une commande ne peut être destinée qu'à une seule adresse (en cas de livraison) mais une adresse peut être attribuée à plusieurs commandes.



Un client peut avoir une à plusieurs adresses (eg. domicile, travail) et une adresse peut appartenir à plusieurs clients (eg. membres d'une famille, colocataires).

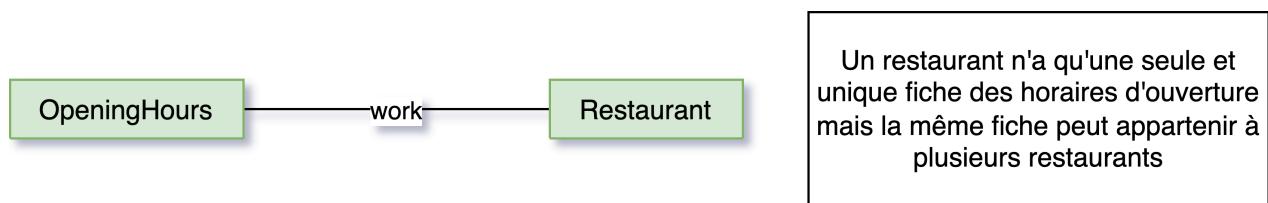
Restaurant

Cette classe regroupe les attributs des différentes restaurants du groupe.



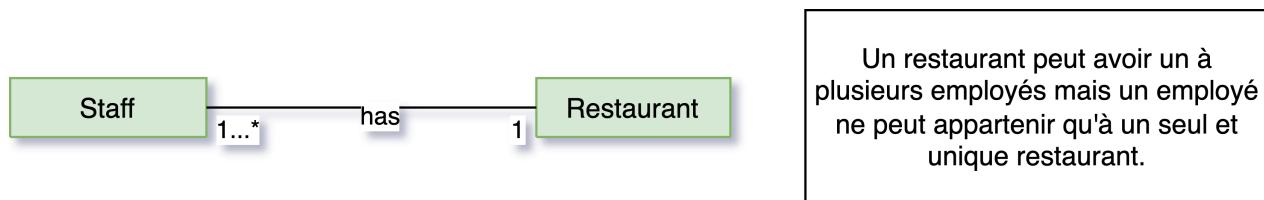
OpeningHours

Cette classe regroupe les attributs des horaires d'ouverture des restaurants.



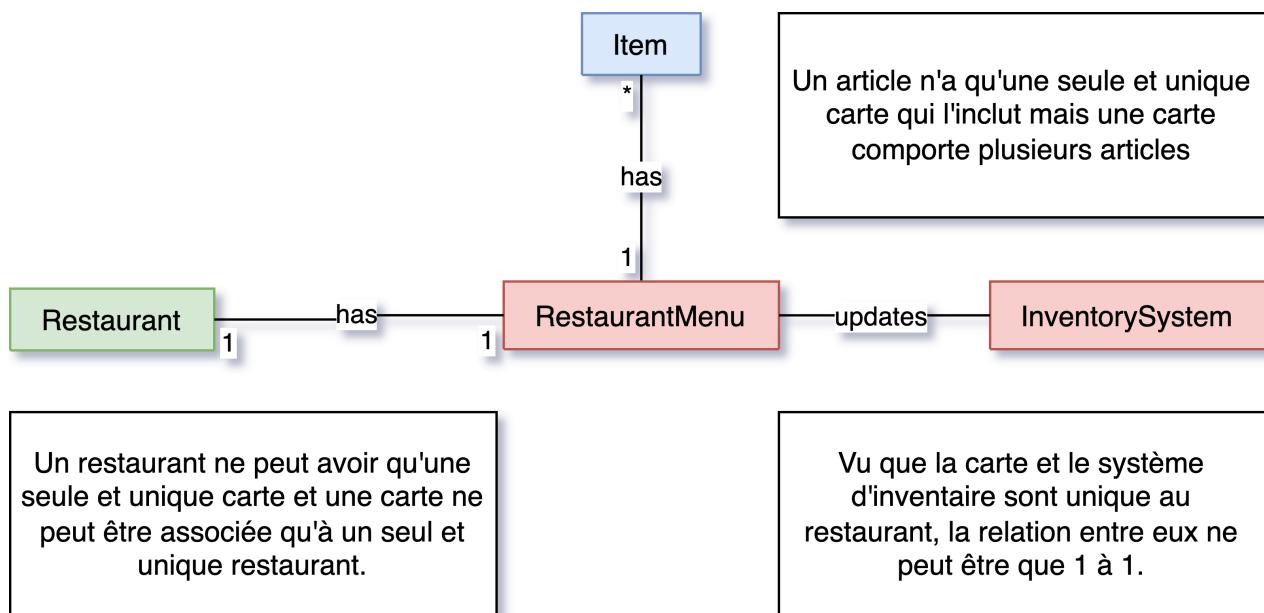
Staff

Cette classe regroupe tous les attributs associés à un membre du personnel.



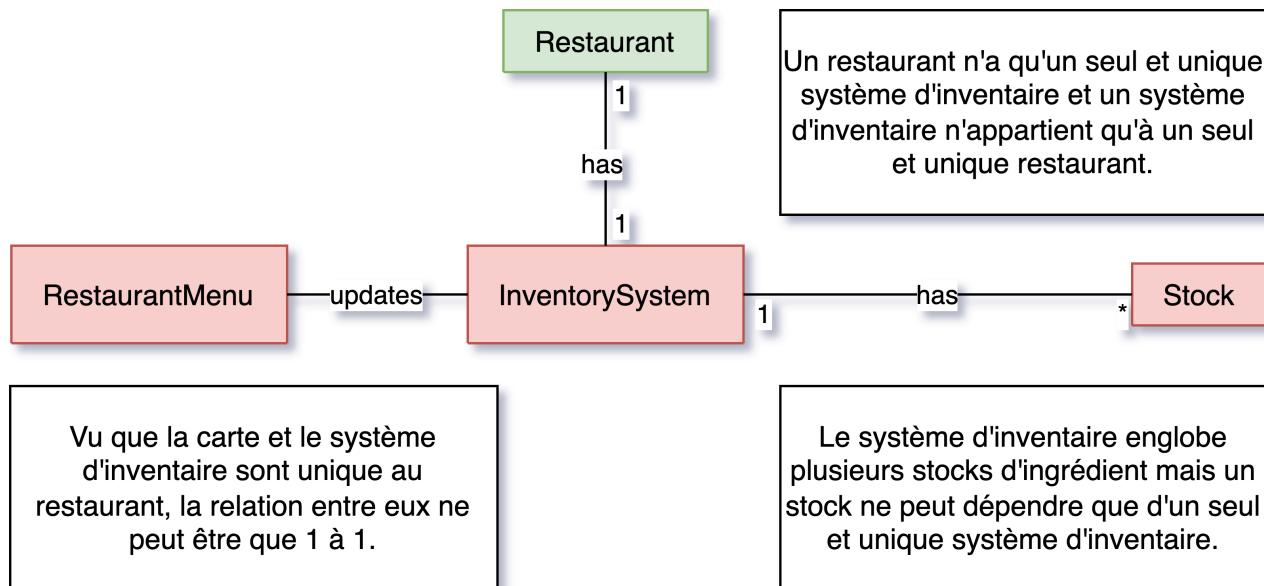
RestaurantMenu

Cette classe regroupe tous les attributs associés à la carte du restaurant.



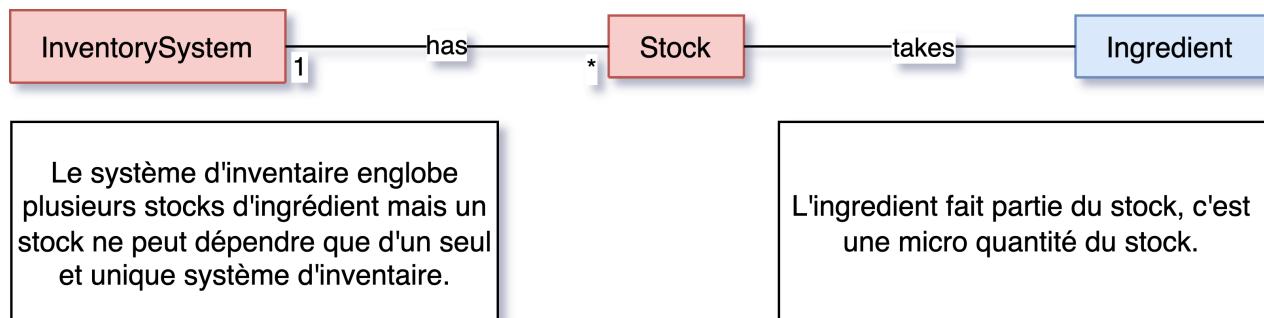
InventorySystem

Cette classe regroupe tous les attributs associés au système d'inventaire du restaurant.



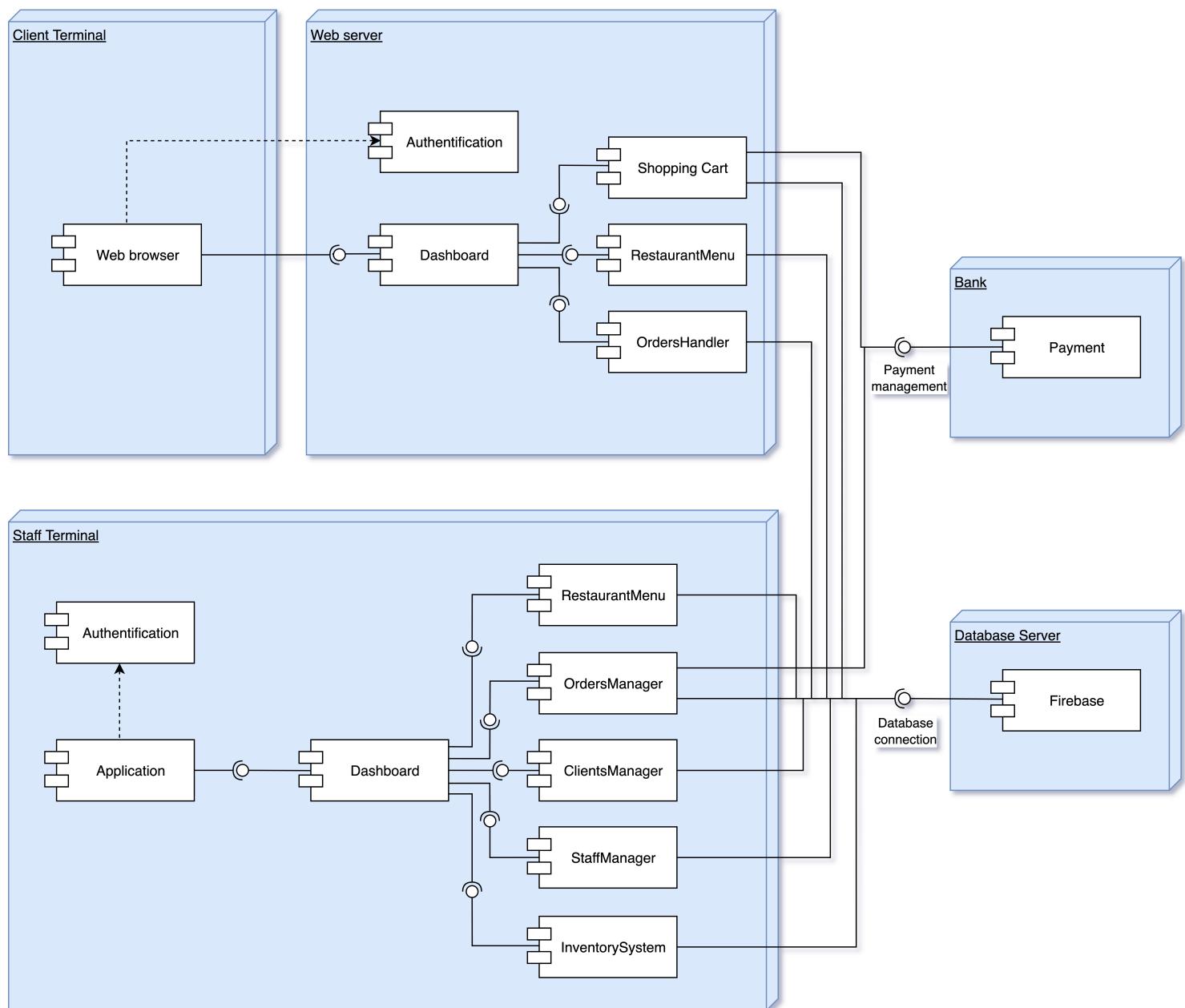
Stock

Cette classe regroupe tous les attributs associés à un stock d'un ingrédient donné (eg. farine, mozzarella...).



3.4. Diagramme des composants

Diagramme des composants



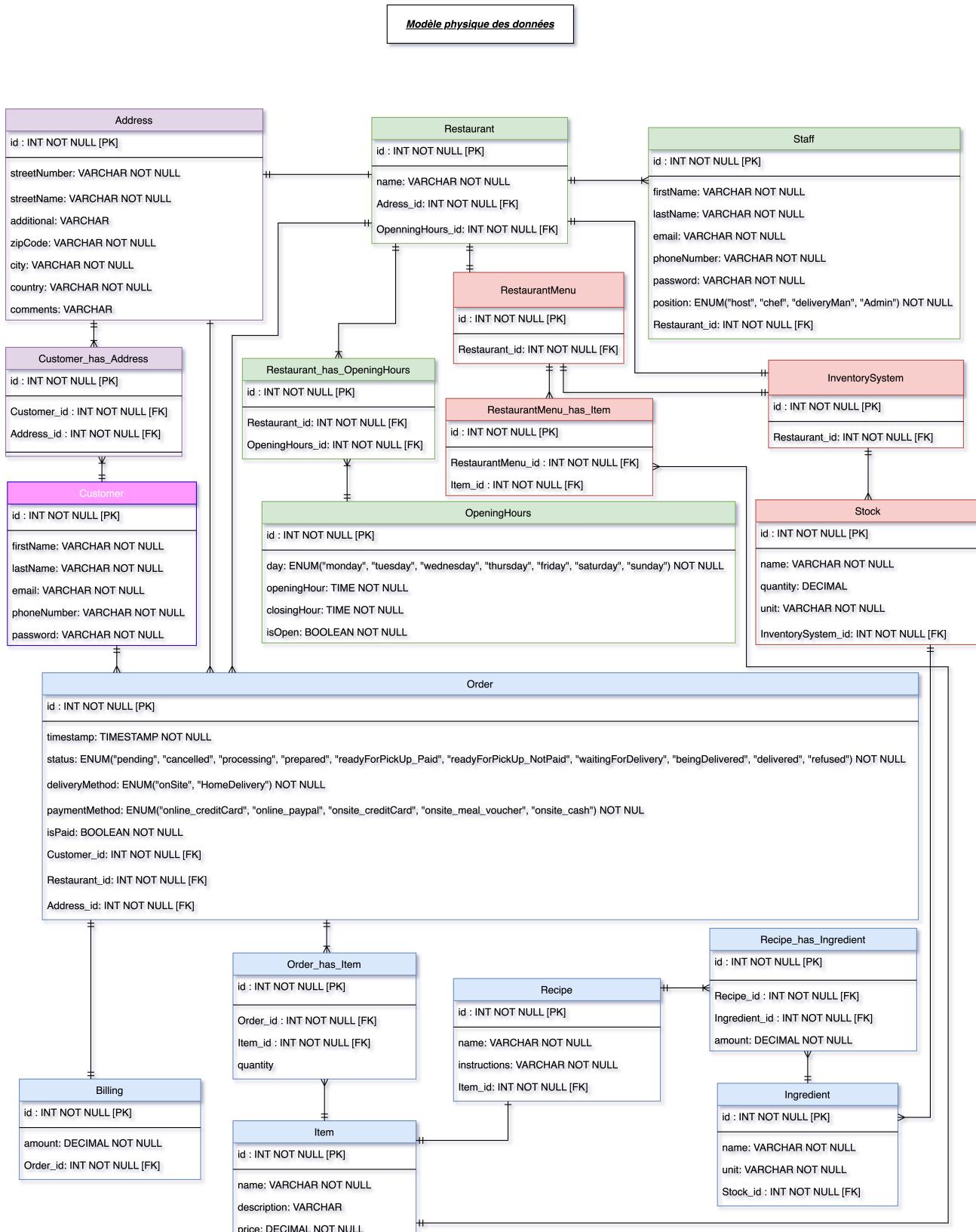
3.5. Description des composants

- Un **site web pour le Front office** (l'espace client pour commander, suivre ses commandes, gérer son compte...).
 - Le composant “Web browser” est un outil utilisé par les clients pour naviguer sur Internet et accéder au site web du restaurant.
 - Le composant “Authentication” du Front office (site web) permettra à un nouveau client de s'inscrire et aux clients déjà inscrits de se connecter. Cette étape est obligatoire pour passer des commandes.
 - Le composant “Dashboard” du Front office (site web) permettra aux clients de gérer leur espace personnel ainsi que d'accéder à toutes les fonctionnalités qui leurs sont proposées par le site du restaurant.
 - Le composant “ShoppingCart” permettra aux clients de constituer un panier, d'ajouter ou supprimer des articles et de passer commande.
 - Le composant “RestaurantMenu” du Front office (site web) permettra aux clients de consulter la liste des articles proposés par le restaurant. C'est à partir de ce composant que le panier pourra être rempli.
 - Le composant “OrderHandler” permettra aux clients de gérer et suivre leurs commandes, modifier ou supprimer les commandes en cours.

- Une **application Android pour le Back office** (l'interface du personnel pour recevoir, préparer, livrer les commandes, gérer le restaurant...).
 - Le composant « Authentication » du Back office (l'application staff) permettra aux différents employés du groupe de se connecter sur l'application. Cette étape est obligatoire pour accéder aux fonctionnalités de l'application.
 - Le composant “Dashboard” du Back office (l'application staff) permettra aux employés d'accéder à toutes les fonctionnalités qui leurs sont autorisées par l'application du Back office.
 - Le composant “RestaurantMenu” du Back office (l'application staff) permet aux employés de consulter la liste des articles proposés par le restaurant ainsi que leurs aide-mémoire-recette, mais aussi, pour les employés autorisés, de modifier cette liste.
 - Le composant “OrderManager” permet aux employés de gérer les commandes reçues (modifier ou supprimer des articles d'une commandes, modifier les coordonnées d'une commandes, supprimer une commande...) selon leurs autorisations.
 - Le composant “ClientManager” permet aux employés autorisé de gérer les comptes client (modifier ou supprimer un compte).
 - Le composant “StaffManager” permet aux employés autorisé de gérer les comptes staff (modifier ou supprimer un compte).
 - Le composant “InventorySystem” permet aux employés autorisé de gérer l'inventaire du restaurant.
- Une **base de données Firebase** qui permettra la transition des données collectées et manipulées.

4. Model physique des données

4.1. Diagramme MPD



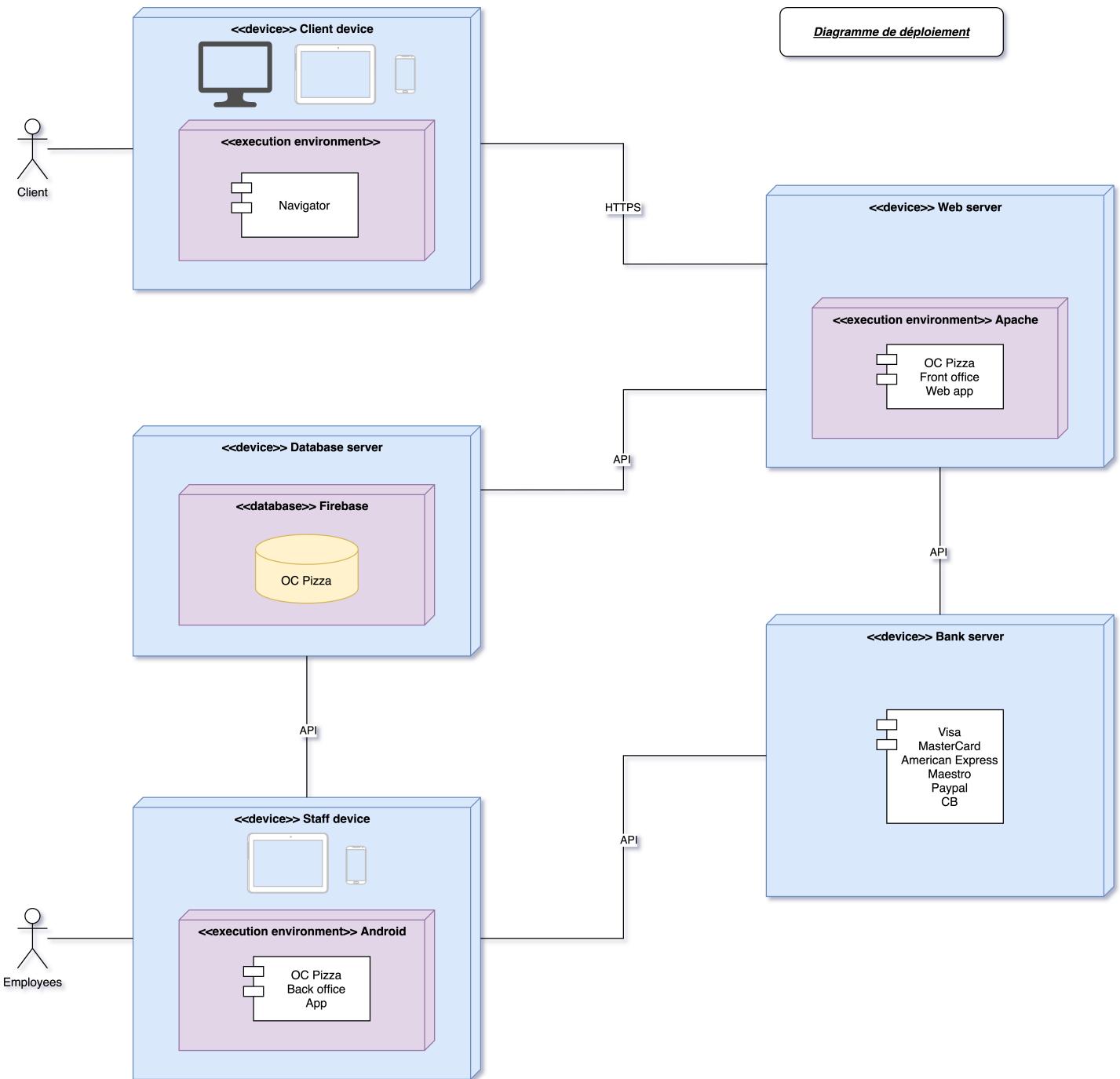
4.2. Description des tables

Le diagramme ci-dessus est une adaptation fidèle au modèle vu précédemment dans le diagramme des classes. Les tables, leurs noms, leurs composants et leurs significations restent les-mêmes que ceux du diagramme des classes. Seules quelques clés étrangères y ont été rajoutées.

De ce fait, une description détaillée du MPD est superflue.

5. L'architecture de déploiement

5.1. Diagramme de déploiement



5.2. Description du diagramme

Dans ce diagramme chaque entrée représente un nœud du déploiement du système. La communication entre ces noeuds se fait à travers des API sécurisées.

Client device

Le terminal à travers lequel le client, usant d'un navigateur, accède au site web (front office) du restaurant et à ses fonctionnalités.

Web server

Le serveur Apache qui héberge le site web (front office) du restaurant. Le front-end du site sera développé en HTML, CSS et JavaScript et le back-end en Python (Django Framework). Le serveur communique directement avec le Database server et le Bank server via des API sécurisées afin d'enregistrer les informations liées aux clients ou aux commandes, ainsi que les transactions bancaires.

Staff device

Le terminal utilisé par le personnel du restaurant, un appareil Android où l'application OC Pizza (back office) du restaurant y est installée. Il communique directement avec le Database server et le Bank server via des API sécurisées afin d'enregistrer les informations liées à la gestion des commandes (eg. suivi d'une commande, modification, remboursement...) et à la gestion du restaurant (stock, carte du restaurant...).

Bank server

Le serveur de la banque qui permettra et validera les paiements.

Database server

Le serveur qui héberge la base de données Firebase Realtime Database. Celle-ci prendra en charge toutes les données liées au système OC Pizza.

6. Conclusion

Afin de répondre aux besoins exprimés par OC-Pizza, la solution proposée est la suite; mettre en place un système OZ Pizza qui sera composé des éléments ci-dessous :

- Un site web du restaurant destiné aux clients pour:
 - Afficher la carte du restaurant;
 - Proposer un espace personnel client;
 - Effectuer des commandes.
- Une application Android destinée au personnel du restaurant pour:
 - Gérer les commandes en temps réel, de leur réception à leur livraison;
 - Gérer le restaurant, le stock et le personnel.
- Une base de données Firebase pour:
 - Structurer les données mises en jeu par le système;
 - Gérer ces données en temps réel.

Nous avons, dans un premier temps, présenté le domaine fonctionnel, en annonçant l'architecture de notre système, sous la forme de diagrammes de classes et de composants.

Dans un second temps, nous avons clairement défini le modèle physique de données et les éléments constituants la base de données.

Pour finir nous avons établi la base de données qui contient un jeu de données de démo, pour simuler une commande factice.