OC Pizza

Application web OC Pizza

Dossier de conception technique

Version 1.1

Auteur
Léonard Colin
Développeur d'application

TABLE DES MATIÈRES

1 - Versions

Auteur	Date	Description	Version
Léonard COLIN	13/11/2020	Création du document	1.0
Leonard COLIN	20/03/2021	Mise à jour 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.

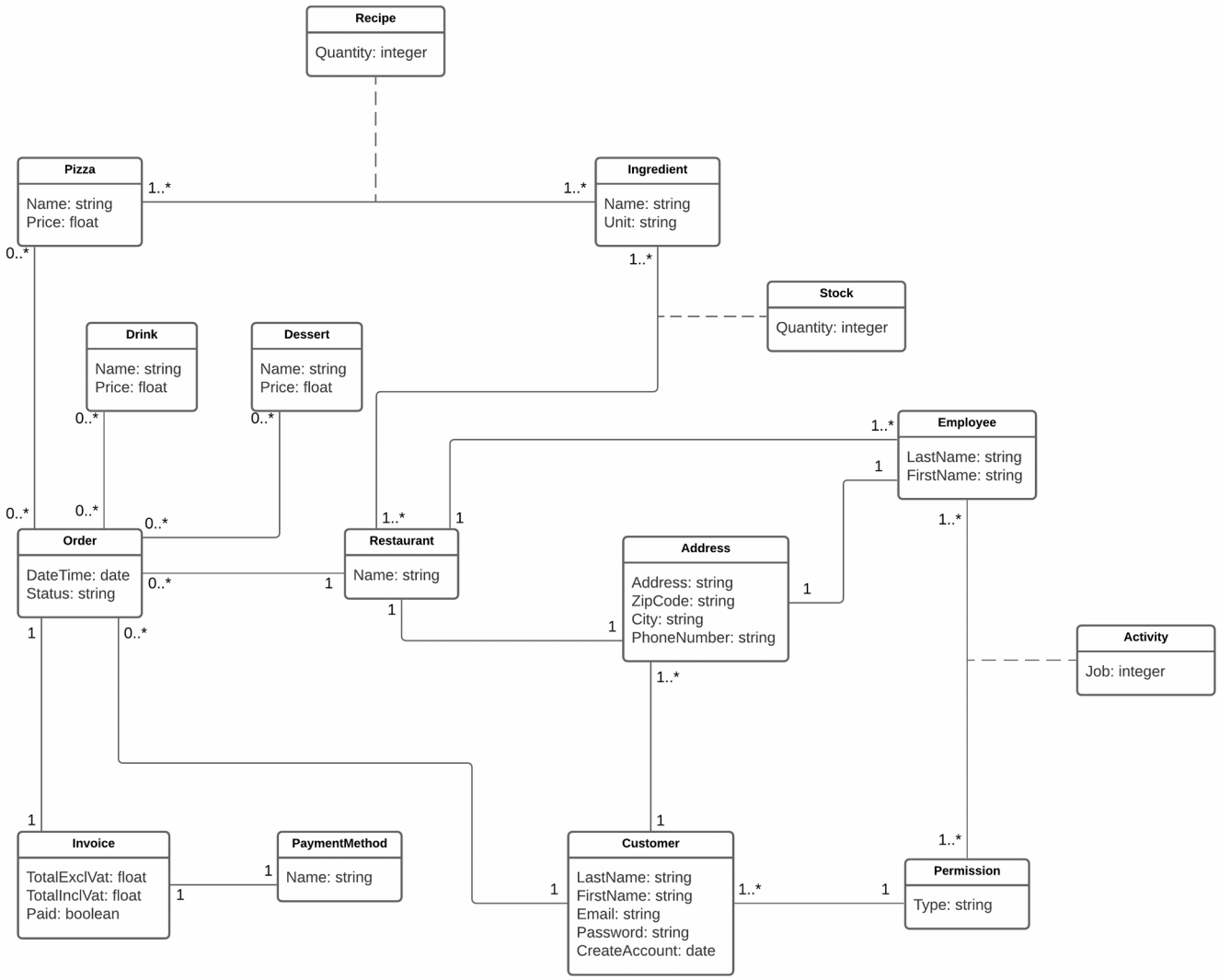
L’objectif du document est de décrire la conception et la solution technique choisie de l’application web pour OC Pizza

Les éléments du présents dossiers découlent :

- de l’analyse des besoins clients
- de l’analyse du domaine fonctionnel
- de l’analyse des différents composants du système

3 - Domaine fonctionnel

3.1 - Diagramme de classes



Ce diagramme représente les classes qui seront présentes dans notre application et leurs relations entre chacune d'elles ainsi que leurs cardinalités.

Nous pouvons observer les différents liens qui relient chaque classe.

Par exemple, un restaurant n'aura qu'une seule adresse. Cependant un client, lui, pourra avoir une ou plusieurs adresses.

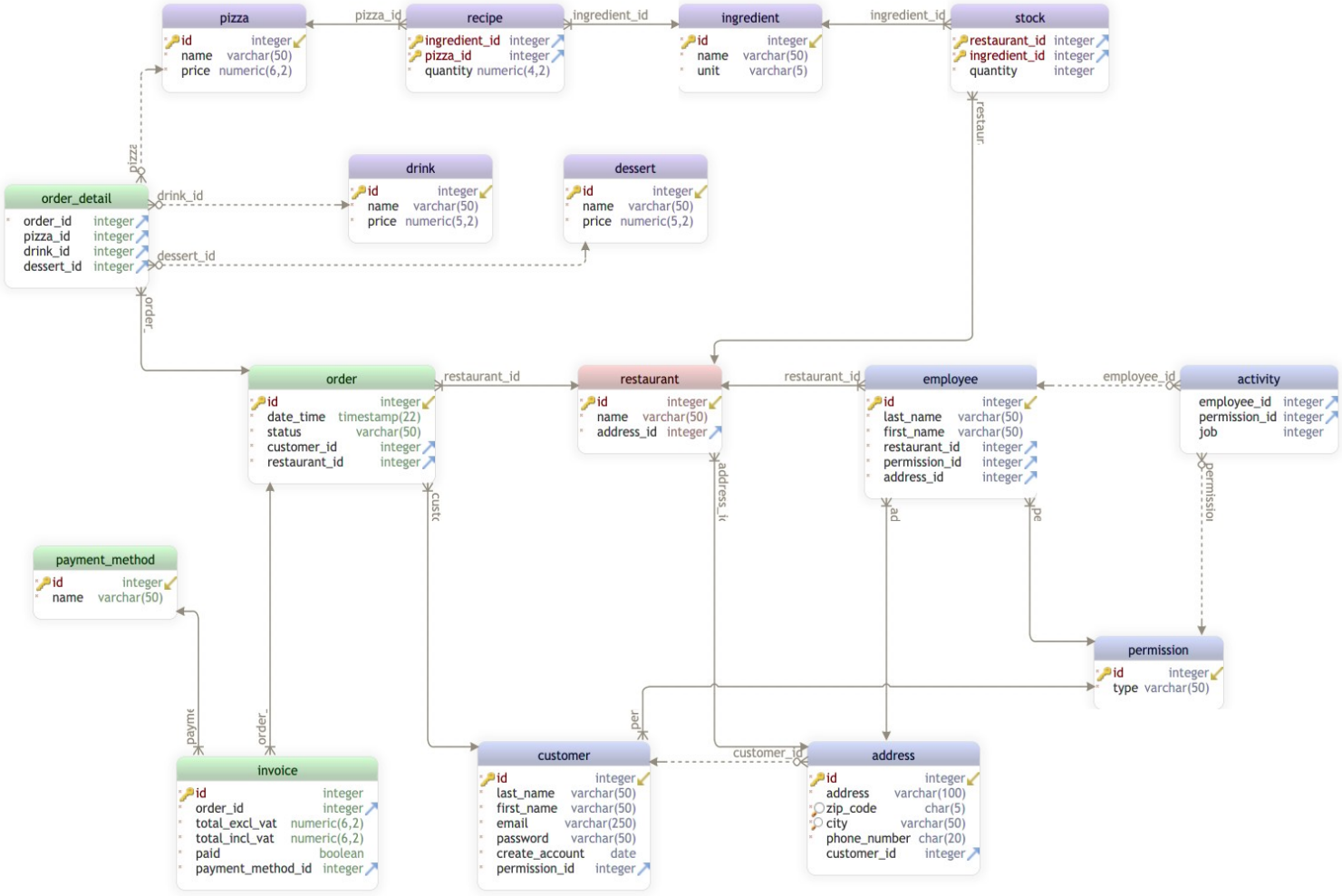
Autrement, la classe Recipe qui correspond aux recettes de chaque pizza, fait le lien entre les classes Pizza et Ingredient qui ont une relation « many to many ».
En effet, un ingrédient peu composer une ou plusieurs pizza et une pizza est composée de un ou plusieurs ingrédient.

La classe « Permission » représentera le droit d’accès de l’utilisateur.

Nous pouvons penser d’une manière généralisée à 3 droits :

- « Admin »pour un manager, qui aura un accès total à toutes les fonctionnalités utilisateur.
- « Employee » pour un employé d’un restaurant qui aura accès à la partie qui lui servira spécifiquement selon le poste qu’il occupe.
- « Customer » pour un client qui aura un accès restreint mais avec tous les éléments lui permettant de passer commande et de suivre sa commande, entre autres.

3.2 - Modèle physique de données



Le modèle physique de données est construit sur le modèle du diagramme de classe. Il représente l'architecture de notre base de données.

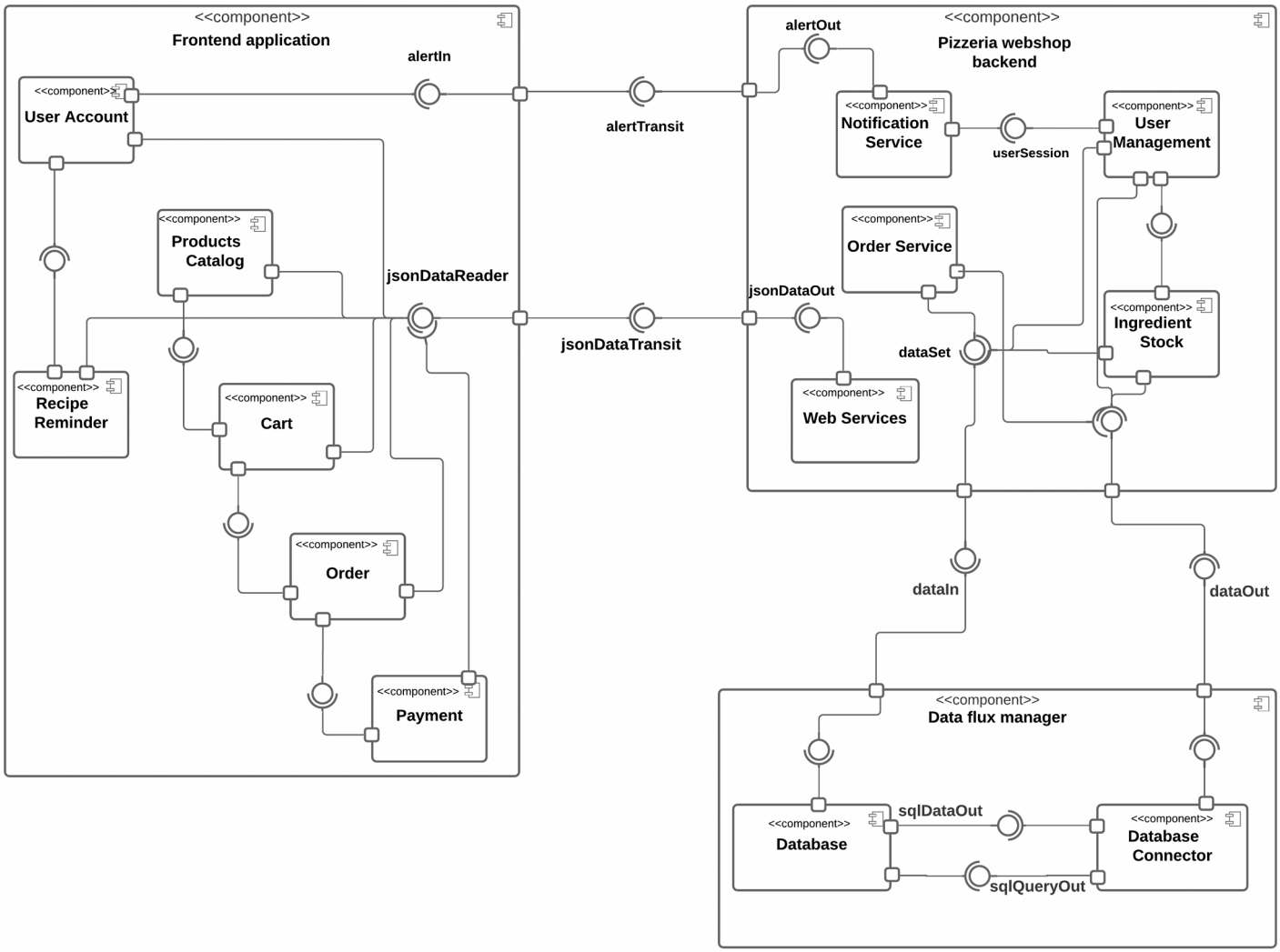
Nous observons les différentes tables qui représentent les classes vues précédemment, ainsi que leurs champs qui composeront la base de données de notre application.

L'utilisation de clés étrangères et de tables de jointures nous permettra de faire tous les liens entre les différentes données du projet.

4 - Composants du système

4.1 - Diagramme de composants

Diagramme UML de composants



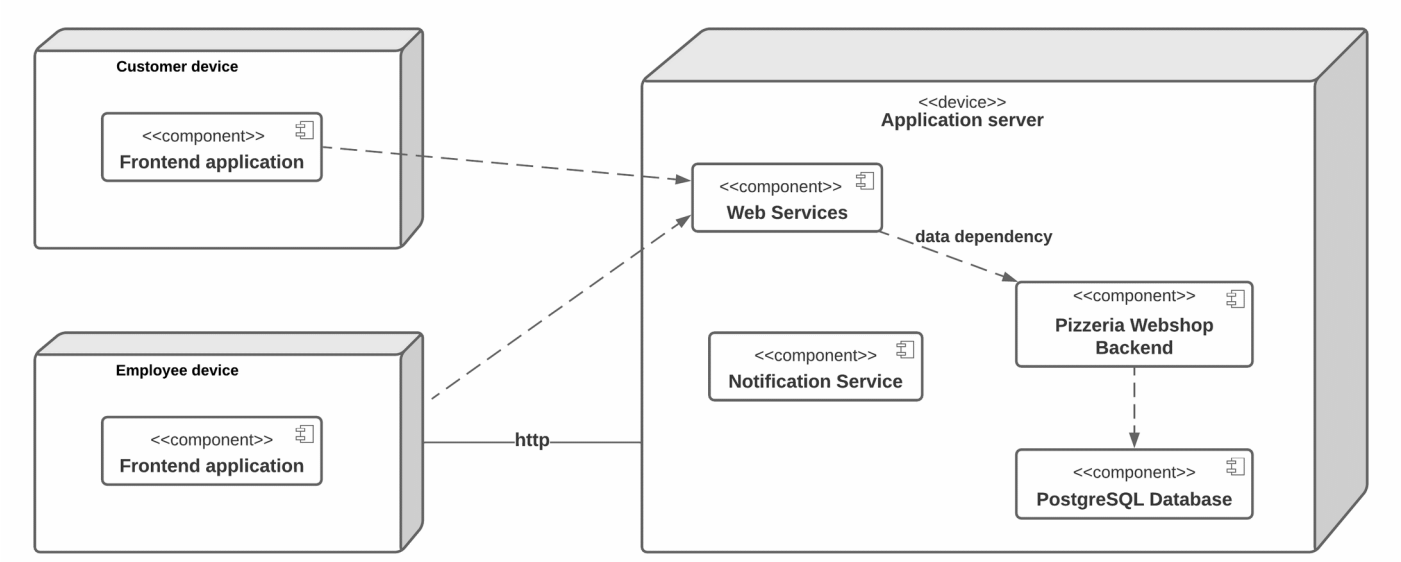
Voici la vue d'ensemble des composants importants de l'application et leur façon de communiquer entre eux.

- Le User Account va se connecter via le User Management afin de pouvoir exécuter ses fonctionnalités. Par exemple, s'il s'agit de l'utilisateur « Pizzaiolo », celui-ci pourra consulter le mémo des recettes si besoin.
- La base de données recevra des données via le service de commande afin de les transmettre au Database Connector qui pourra notamment mettre à jour le stock d'ingrédients à la suite d'une commande passée.

5 - Architecture de Déploiement

5.1 - Diagramme de déploiement

Diagramme UML de déploiement



Le diagramme de déploiement ci-dessus représente l'organisation de l'infrastructure physique des composants du système et la manière dont ceux-ci sont répartis et communiquent entre eux.

6 - ARCHITECTURE LOGICIELLE

6.1 - Logiciels utilisés

6.1.1 - Gestions des versions

Le projet sera versionné avec Git. C'est un outil utilisé dans la grande majorité des projets de développement d'application. Il permet une collaboration efficace entre les développeurs d'un même projet.

6.1.2 - Front-end

Le Front-end sera développé avec Bootstrap.

Il s'agit d'un outil très complet et répandu qui permet un démarrage rapide et efficace d'une application web.

Sa solution responsive sera parfaitement adaptée aux besoins utilisateurs pour ce projet.

6.1.3 - Back-end

Pour le Back-end, Django sera de rigueur.

Ce framework est développé en Python, qui sera le langage de programmation utilisé.

Il est facile à mettre en place et à déployer sur Heroku et sa documentation est très fournie.

6.1.4 - Base de données

La base de données sera gérée par PostgreSQL. Ce choix se justifie par sa compatibilité avec la plateforme de déploiement Heroku.

C'est un système de gestion de base de données performant et très répandu, aux fonctionnalités riches et avancées. Il saura gérer avec fiabilité de gros volumes de données.

De plus, il sera très adapté pour fonctionner de pair avec Django.

7 - GLOSSAIRE

Front-end	En développement web, le front-end fait référence à l'ensemble des éléments visibles et accessibles directement par l'utilisateur, en opposition au back-end
Back-end	Le back-end concerne toute la partie invisible de la conception d'une application web, tel que le développement de bases de données, par exemple.
Framework	Un framework est, comme son nom l'indique en anglais, un "cadre de travail". L'objectif d'un framework est généralement de simplifier le travail des développeurs informatiques en leur offrant une architecture "prête à l'emploi" et qui leur permette de ne pas repartir de zéro à chaque nouveau projet.