OC Pizza

Système de gestion de pizzerias

Dossier de conception technique

Version 1

Auteur
Sylvain Rieutor
Développeur

Table des matières

1 - Versions

Auteur	Date	Description	Version
SRR	01/05/20	Création du document	1

2 - Introduction

2.1 - Objet du document

Les spécifications techniques détaillées ont étés créés en appui avec l'analyse fonctionnelle effectuée en amont

Ce document à pour principal objectif de documenter les méthodes, procédés et technologies sélectionnées pour faire face aux contraintes de réalisation du projet.

2.2 - Références

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

1. **OC-DF-1 :** Dossier de conception fonctionnelle
2. **OC-DE-1:** Dossier d'exploitation

3 - Architecture Technique

3.1 - Solutions technologiques

Langage de programmation :

Notre choix s'est porté sur le **PYTHON (DJANGO)** d'une part car s'est un langage très utilisé parmi la communauté de programmeurs et ce depuis les années 90.

Il possède également une très vaste bibliothèque permettant à l'application de l'entreprise de rester réactive et compétitive avec une très grande marge de manœuvre.

Enfin, python nous permet de travailler avec de nombreux langages et technologies tellement il s'avère ouvert vers l'extérieur.

Utiliser python c'est assurer au client, une application fiable, robuste et ouverte sur l'avenir.

Système de gestion de base de données :

Notre choix s'est porté sur **PostgreSQL**, outre sa réputation qui n'est plus à prouvée postgreSQL et la base de donnée de référence avec le framework **DJANGO**

D'autre part, postgreSQL est fiable et l'intégrité des données y est performante.

De plus MySQL est prise en charge par défaut dans la majorité des hébergeurs.

Enfin, sa très grande communauté fournie un support et une documentation considérable

3.2 - Accessibilité

Compatibilité navigateur :

L'application web devra être compatible avec les navigateurs suivants :

- Google chrome,

- Microsoft Edge,

- Safari,

- Mozzila firefox,

- Opéra.

Types d'appareils :

L'application web sera conçu de manière « responsive » pour qu'il assure une navigation optimale sur l'ensemble des appareils disponible sur le marché.

Liste des appareils devant être prise en compte :

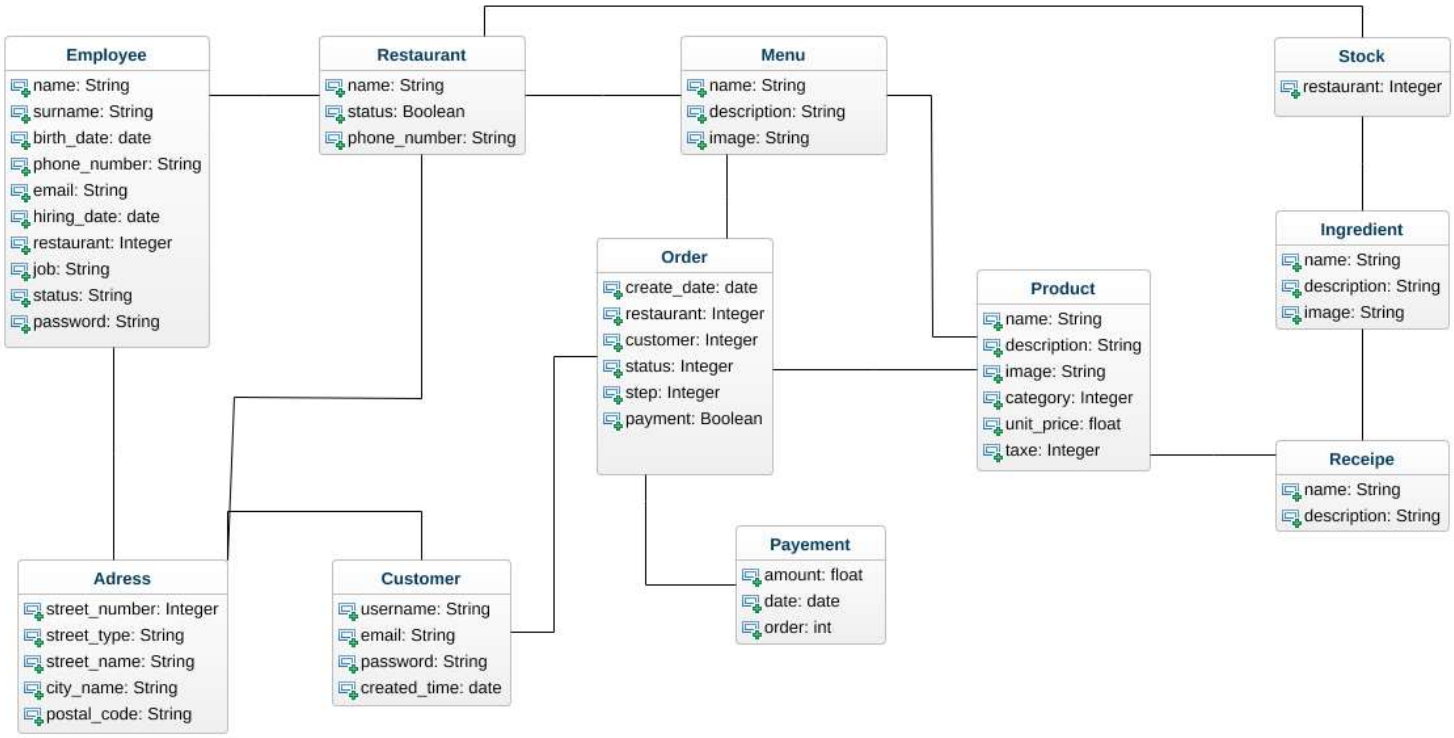
- Smartphones,

- Tablettes,

- Ordinateur portables,

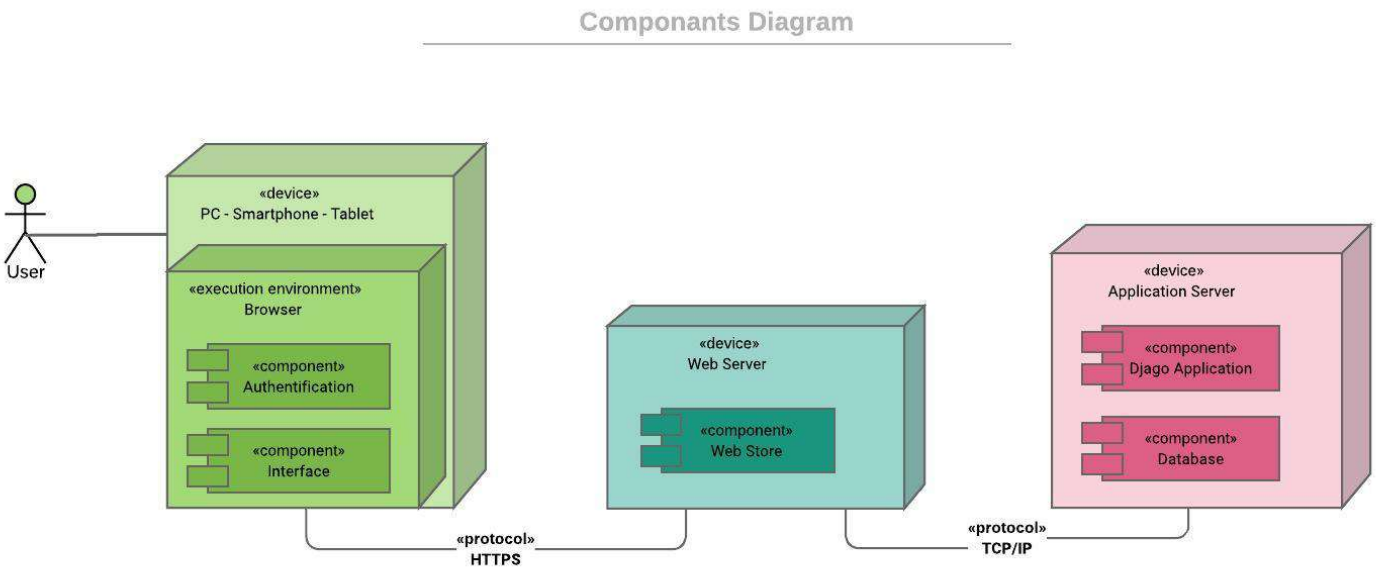
- Ordinateur de bureau.

3.3 - Diagramme de classe



Le diagramme de classes décrit la structure du système en modélisant ses classes/objets, leurs attributs et leurs relations.

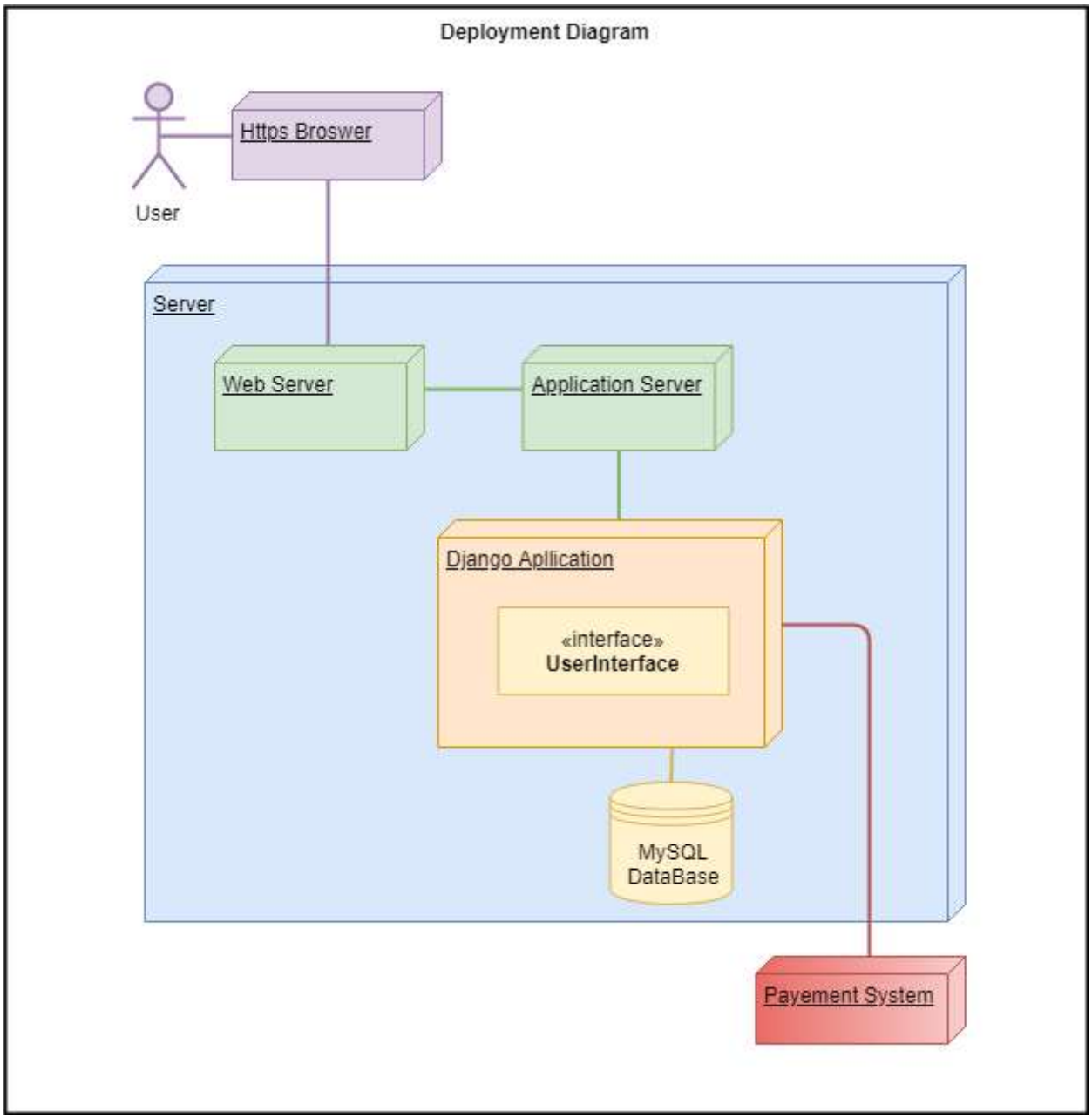
3.4 - Diagramme de composants



Le diagramme de composants représente la structure du système à travers ses composants, leurs interfaces et leurs dépendances.

4 - Architecture de Déploiement

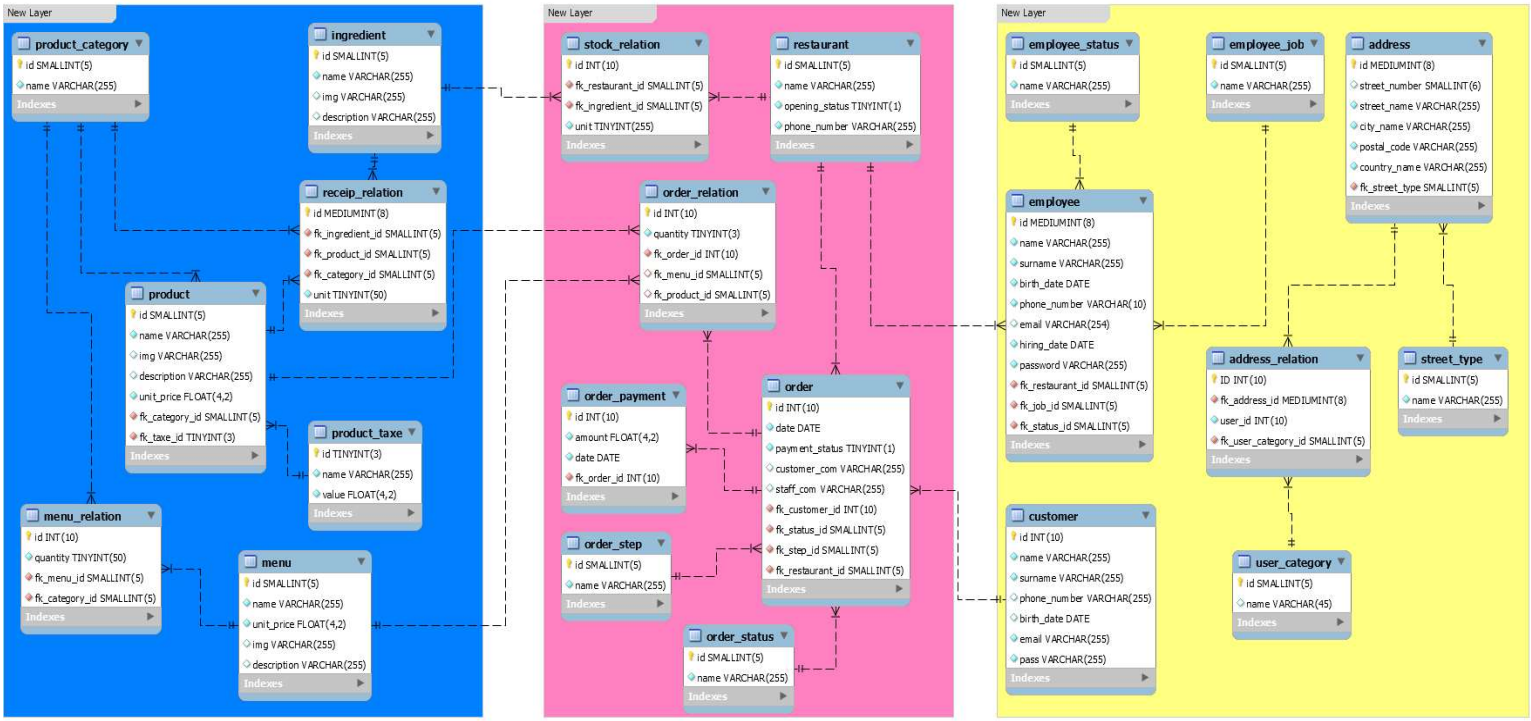
Diagramme de déploiement



Le diagramme de déploiement représente l'agencement des composants physiques du système.

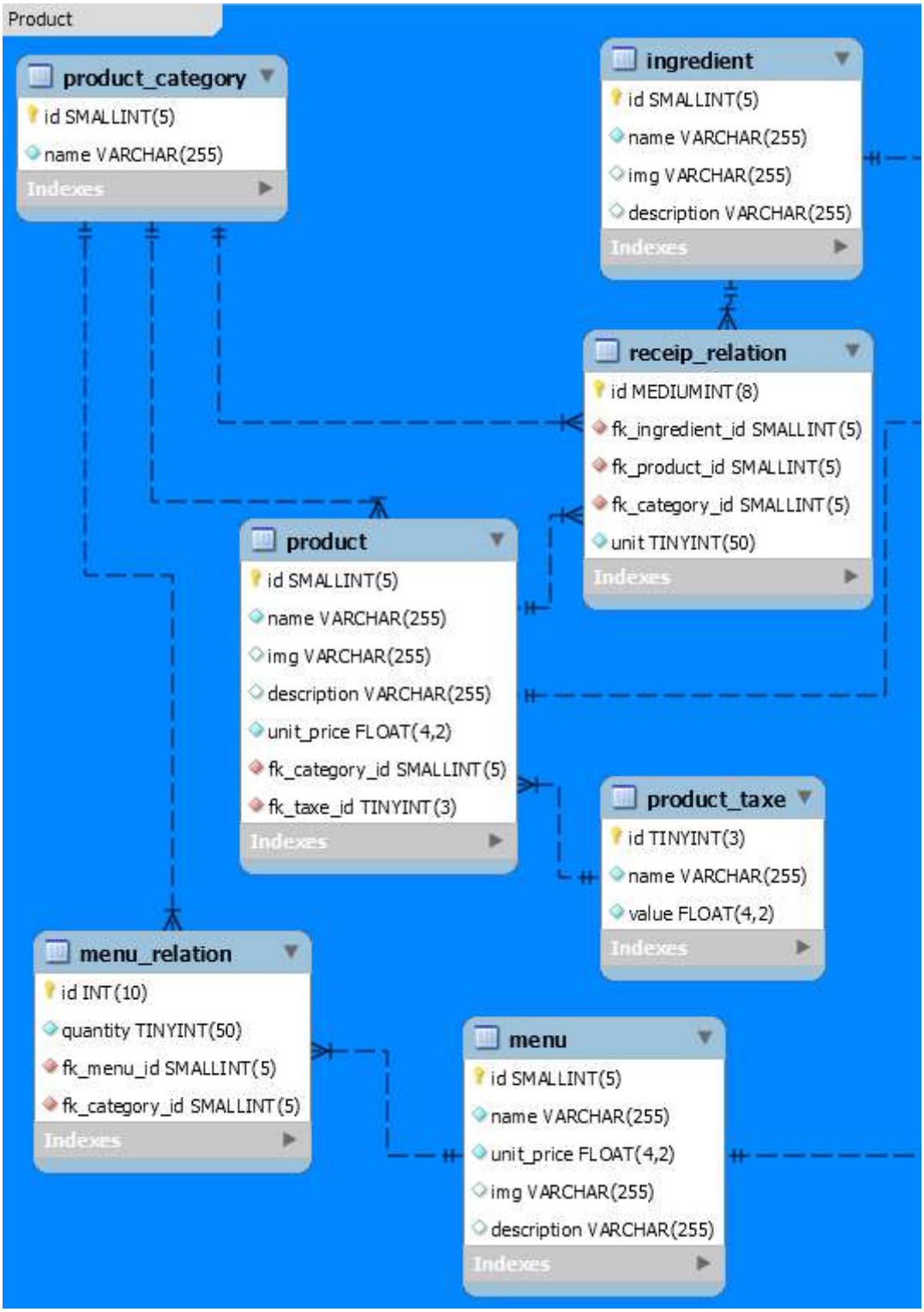
4.1 - Diagramme de la Base de données

Diagramme général MPD

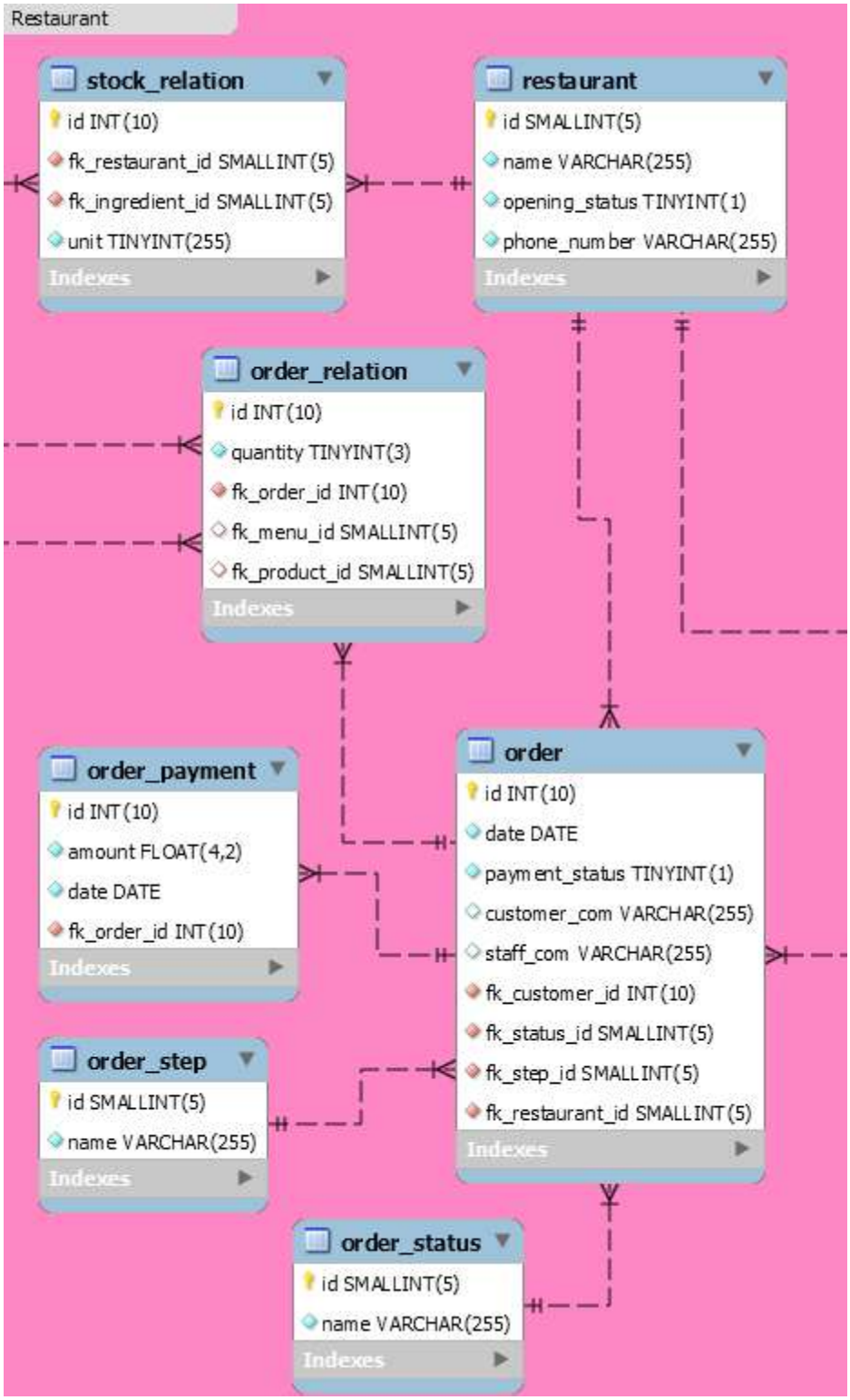


Le Modèle Physique de Données (MPD) représente la structure de la base de données et permet de visualiser les relations et dépendances entre les tables qui la composent.

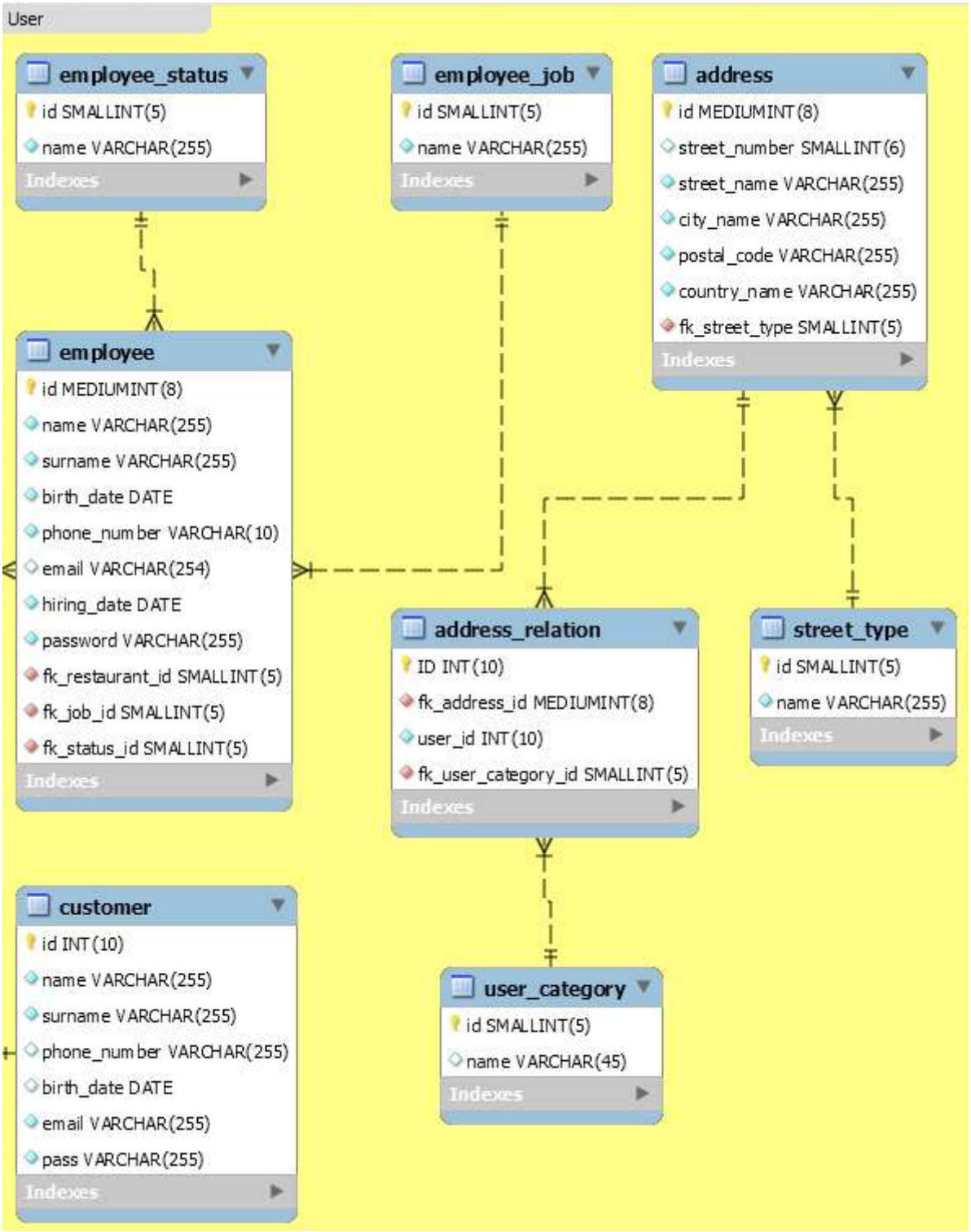
Base de donnée 'Product'



Base de donnée 'Restaurant'



Base de données 'Users'



4.2 - Serveur VPS

L'herbgement se fera avec **OVH**, entreprise française spécialisée dans les services de cloud computing (informatique en nuage)

L'application sera hébergée sur un serveur VPS (Serveur Privé Virtue)

5 - Architecture logicielle

5.1 - Principes généraux

Les sources et versions du projet sont gérées par **Git**.

Les applications respectent le pattern **MVC** (Model View Controller).

5.1.1 - Structure des sources

La structuration des répertoires du projet suit la logique suivante :

- **ocpizza :** Dossier contenant les fichiers de configuration (Django)
  - **settings:** Dossier contenant les paramètres
    - **__init__.py**: réglages communs
    - **production.py:** réglages en production
  - **static :** Dossier contenant les 'static' : images, .CSS, .JS
- **product :** Dossier contenant les fichiers du package *'product'.*
- **restaurant :** Dossier contenant les fichiers du package *'restaurant'.*
- **user:** Dossier contenant les fichiers du package *'user'.*
- **requirements.txt :** ensemble des librairies nécessaires.
- **manage.py:** Utilitaire en ligne de commande de Django
- **Docs :** Dossier contenant la documentation

6 - Points particuliers

6.1 - Gestion des logs

Les logs seront géré par **Sentry,** application de traçage

6.2 - Procédure de packaging / livraison

L'application web sera déployée sur un serveur **OVH**. Elle sera également remise au client OC Pizza sous forme de dossier et de fichiers sources pour les futures mises à jour et modifications.