



# OC Pizza

## OCPizzap

Dossier de conception technique

Version 1.1

**Auteur**

FERRANDEAU MATHIEU

*Analyste Programmeur*



# TABLE DES MATIÈRES

<b>1 - Versions.....</b>	<b>3</b>
<b>2 - Introduction.....</b>	<b>4</b>
2.1 - Objet du document.....	4
2.2 - Références.....	4
<b>3 - Architecture Technique.....</b>	<b>5</b>
3.1 - Application web.....	5
3.2 - Base de données.....	5
3.2.1 - SGBD utilisé.....	5
3.2.2 - Modèle physique de données.....	5
<b>4 - Architecture de Déploiement.....</b>	<b>7</b>
4.1 - Déploiement de l'application.....	7
4.2 - Serveur de déploiement.....	7
<b>5 - Architecture logicielle.....</b>	<b>8</b>
5.1 - Principes généraux.....	8
5.1.1 - Applications Django.....	8
5.1.2 - Les couches.....	8
5.1.3 - Structure des sources.....	9
<b>6 - Points particuliers.....</b>	<b>10</b>
6.1 - Ressources.....	10
6.1.1 - Ressources graphiques.....	10
6.1.2 - Données.....	10
6.2 - Environnement de développement.....	10
6.3 - Procédure de packaging / livraison.....	10
<b>7 - Glossaire.....</b>	<b>11</b>



# 1 - VERSIONS

Auteur	Date	Description	Version
MF	12/11/2019	Création du document	1.0
MF	12/11/2019	Mise a jour du document	1.1



## 2 - INTRODUCTION

### 2.1 - Objet du document

Le présent document constitue le dossier de conception technique de l'application OCPizzap.

L'objectif du document est de présenter les outils, les technologies et les méthodes mises en oeuvre pour réaliser l'application.

Les éléments du présents dossiers découlent :

- de l'entretien réalisé avec le dirigeant de la société OC PIZZA le 01/10/2019
- de l'analyse des besoins et de la rédaction du dossier de conception fonctionnelle.

### 2.2 - Références

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

1. **DCF – 1.1** : Dossier de conception fonctionnelle de l'application
2. **DE – 1.1** : Dossier d'exploitation de l'application.



## 3 - ARCHITECTURE TECHNIQUE

### 3.1 - Application web

Le développement de l'application web sera fait en langage Python (version 3.5) avec l'utilisation du framework Django (version 2.1).

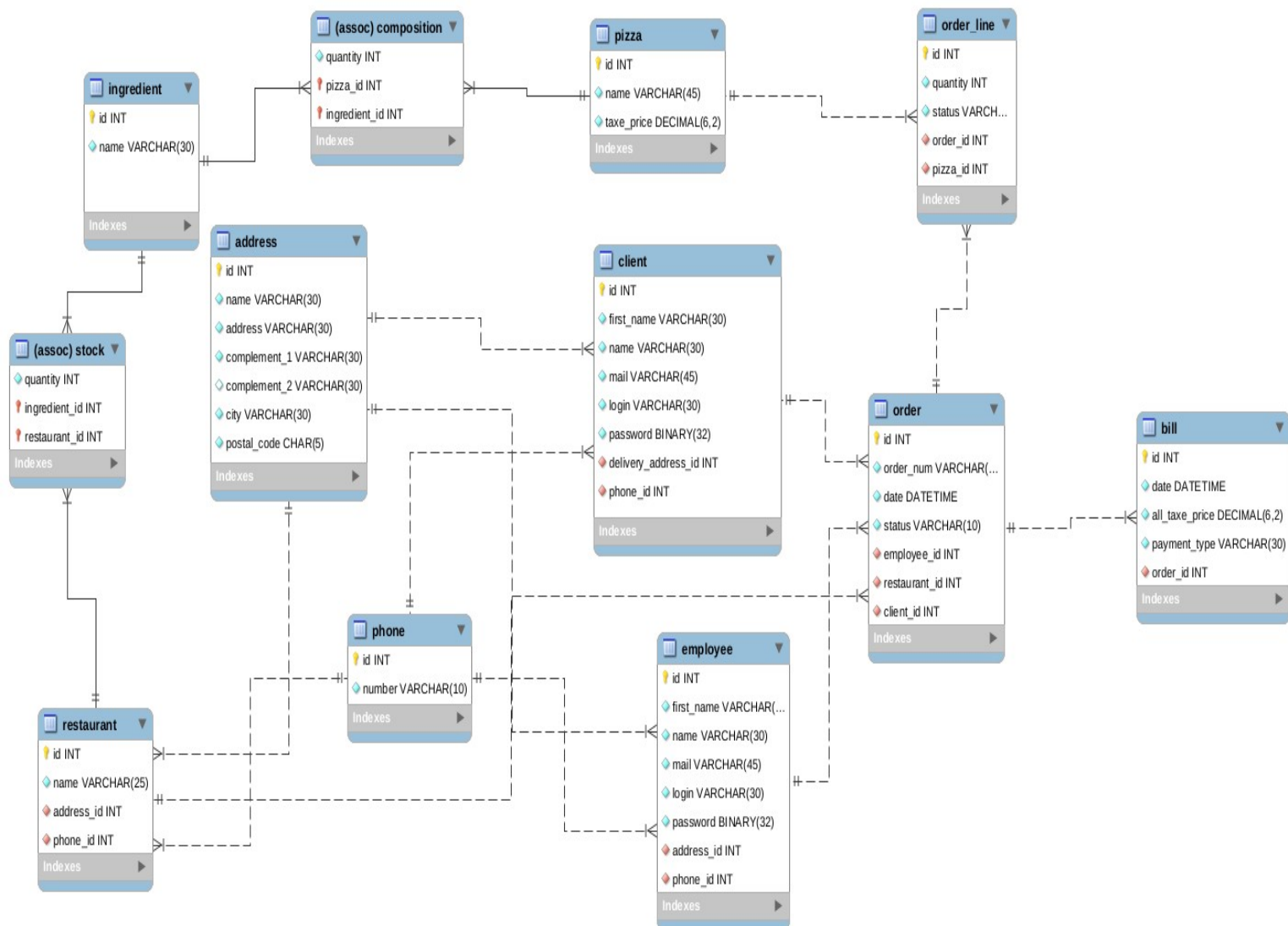
### 3.2 - Base de données

#### 3.2.1 - SGBD utilisé

Le système de gestion de base de données utilisé sera MySQL.

#### 3.2.2 - Modèle physique de données

Le modèle physique de donnée présenté i-dessous se base sur le diagramme de classe détaillé dans le dossier de conception fonctionnelle.

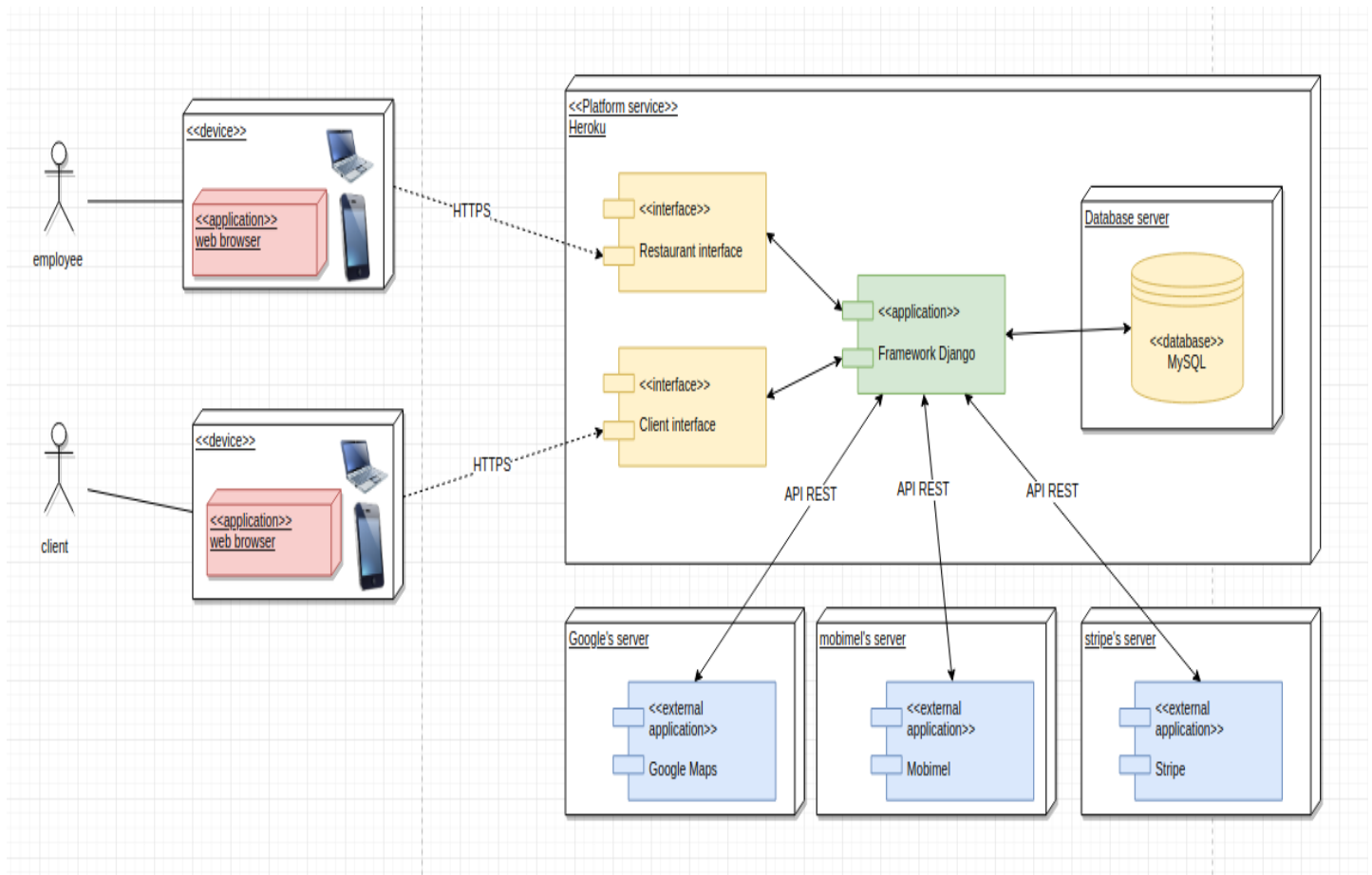


## 1. Modèle physique de données



## 4 - ARCHITECTURE DE DÉPLOIEMENT

### 4.1 - Déploiement de l'application



#### 2. Diagramme de déploiement

La solution envisagée prend la forme d'une application web à partir de laquelle les utilisateurs pourront effectuer les différentes actions prévus. De la commande par un client, la consultation de recette par un pizzaiolo ou encore la consultation des adresses livraison par les livreurs.

### 4.2 - Serveur de déploiement

L'application OCPizzap sera déployée sur les serveurs Heroku, un service de cloud-computing de type Plateforme as a Service (PaaS).

Cette solution semble adaptée au projet et permettra de faciliter le déploiement et la gestion de l'application.



## 5 - ARCHITECTURE LOGICIELLE

### 5.1 - Principes généraux

#### 5.1.1 - Applications Django

Le développement de l'application OCPizzap se présente sous la forme d'un projet Django, lui même divisé en trois « applications » : Authentification, Ventes et Production.

Chaque application respecte le pattern Modeul Vue Template (MVT).

#### 5.1.2 - Les couches

L'architecture de chacune des trois applications est la suivante :

- une couche **présentation** : responsable de l'interface utilisateur/application.
- une couche **métier** : responsable de la logique métier et de la manipulation des données.
- une couche **modèle** : responsable de la représentation des données.

Exemple de découpage en couches de l'application « Ventes » :

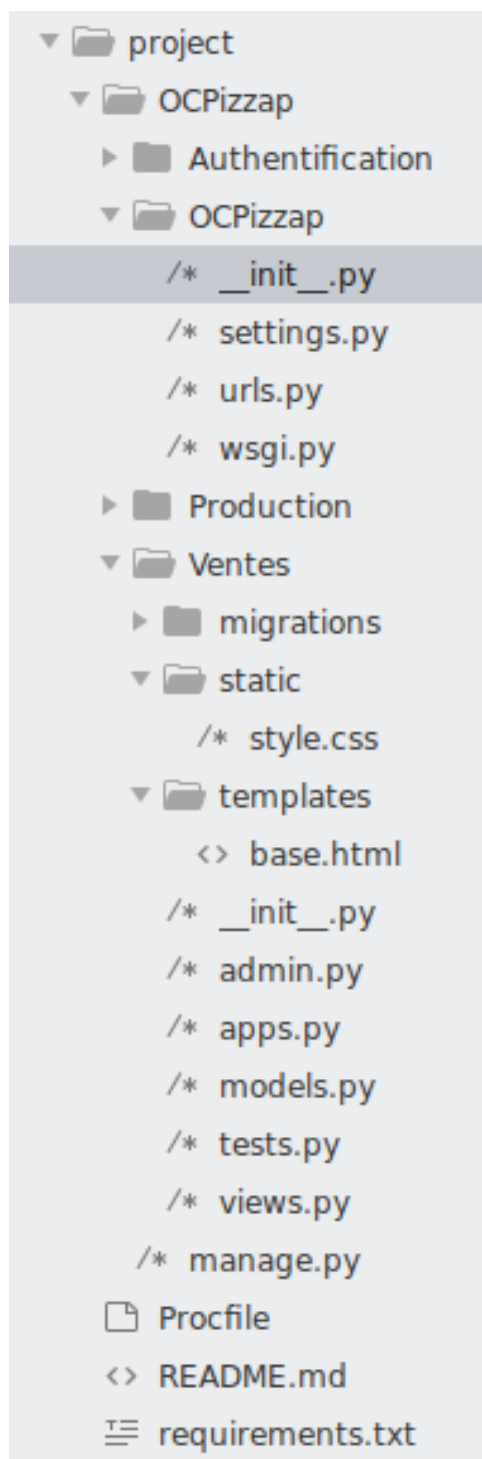
Couche	Rôle
Présentation	Regroupe les notions de Views et Templates de l'application. Elle représente la partie servant d'interface entre l'utilisateur et l'application.
Métier	Regroupe les fonctions et algorithmes nécessaires au fonctionnement de l'application telles que : <ul style="list-style-type: none"><li>- L'affichage du catalogue en fonction des pizzas disponibles.</li><li>- Choix de l'établissement en fonction de la distance du client.</li></ul>
Modèle	Regroupe les classes utilisées par l'application « Ventes » pour manipuler les données (ex : Pizza, Restaurant, Order, Client)





### 5.1.3 - Structure des sources

La structuration des répertoires du projet est faite de façon à respecter les bonnes pratiques d'un projet Django et elle suit donc la logique suivante :





## 6 - POINTS PARTICULIERS

### 6.1 - Ressources

#### 6.1.1 - Ressources graphiques

Les ressources graphiques permettant le design de l'application web sont fournies par OC PIZZA.

#### 6.1.2 - Données

Les données de base (catalogue pizza, ingrédients, magasins) implémentées dans la base de données de l'application sont également de la responsabilité de la société OC PIZZA.

### 6.2 - Environnement de développement

Le développement de l'application ne requiert pas l'utilisation d'un IDE. Elle sera développée en utilisant un éditeur de texte au choix des développeurs.

### 6.3 - Procédure de packaging / livraison

L'application fera l'objet d'un déploiement sur la plateforme Heroku au moment de la livraison finale.

Il sera également remis à cette occasion un dossier d'exploitation permettant la continuité de l'utilisation de l'application.



## 7 - GLOSSAIRE

<b>DJANGO</b>	Framework permettant la réalisation web en Python
<b>IDE</b>	Environnement de développement intégrée
<b>SGBD</b>	Système de gestion de base de données