



OC Pizza

Pizzapp

Dossier de conception technique

Version 1.1

Auteur

LE HAY Ronan

Analyste Programmeur



TABLE DES MATIERES

1 - Versions	3
2 - Introduction	4
2.1 - Objet du document.....	4
2.2 - Références.....	4
3 - Architecture Technique	5
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
4 - Architecture de Déploiement	7
4.1 - Déploiement de l'applicaiton.....	7
4.2 - Serveur de déploiement.....	7
5 - Architecture logicielle.....	8
5.1 - Principes généraux	8
5.1.1 - Applications Django.....	8
5.1.2 - Les couches.....	8
5.1.3 - Structure des sources	9
6 - Points particuliers	10
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
7 - Glossaire	11



1 - VERSIONS

Auteur	Date	Description	Version
RLH	14/01/2018	Création du document	1.0
RLH	13/06/2018	Mise à jour du document avec nouvelle trame	1.1

2 - INTRODUCTION

2.1 - Objet du document

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

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

Les éléments du présent dossier découlent :

- De l'entretien réalisé avec le dirigeant de la société OC PIZZA du 01/12/2017
- 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. **P9 - DCF - 1.1** : Dossier de conception fonctionnelle de l'application.
2. **P9 - 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 1.11).

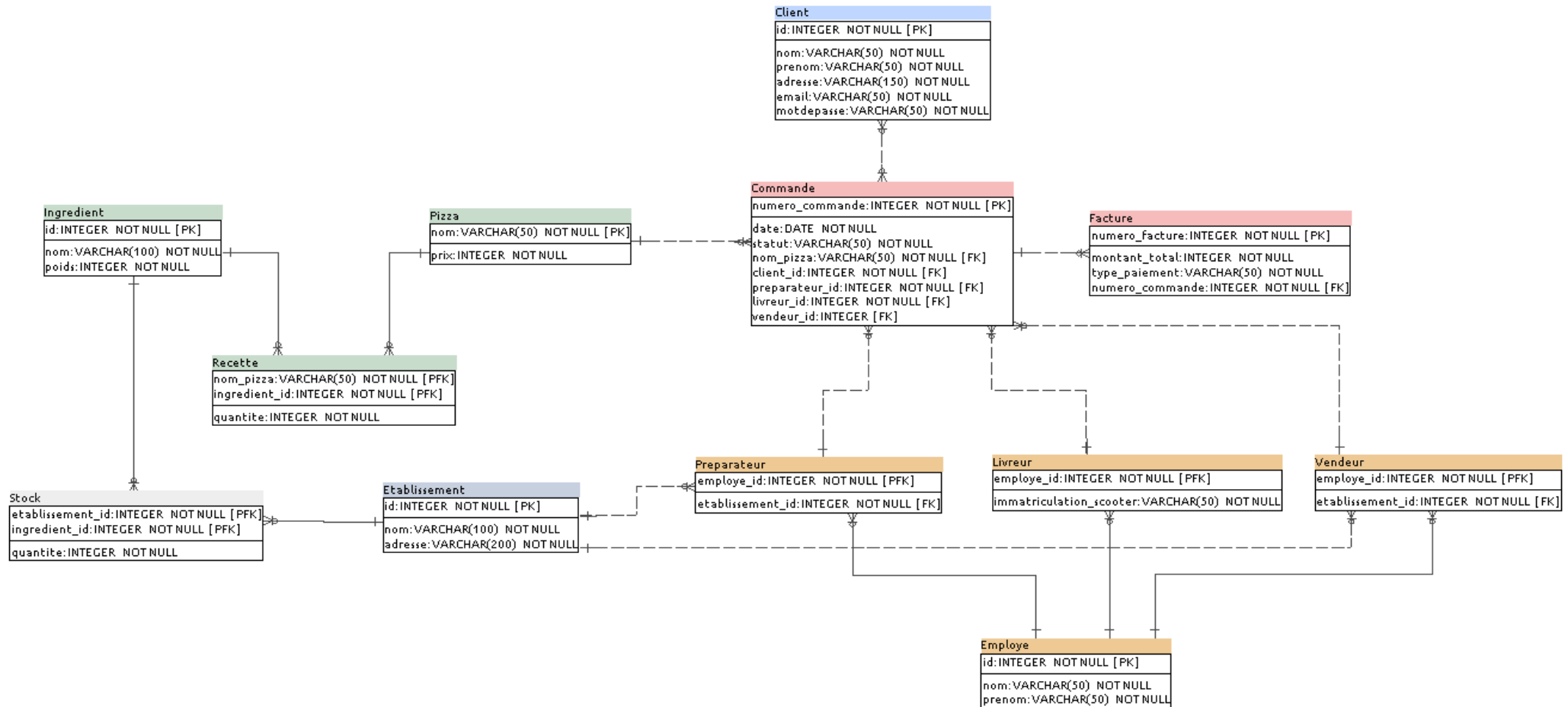
3.2 - Base de données

3.2.1 - SGBD Utilisé

Le Système de gestion de base de données utilisé sera PostgreSQL. Ce choix se justifie en raison de la compatibilité de ce SGBD avec la plateforme de déploiement.

3.2.2 - Modèle physique de données

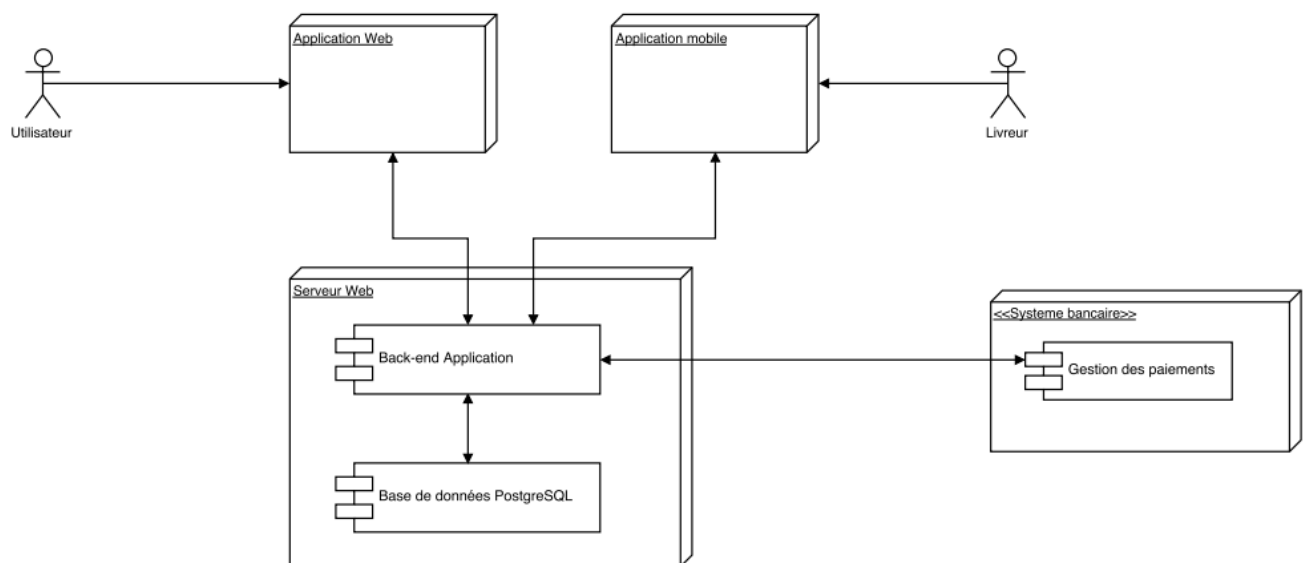
Le modèle physique de donnée présenté ci-dessous se base sur le diagramme de classe détaillée dans le *Dossier de conception fonctionnelle*.



1. Modèle physique de données

4 - ARCHITECTURE DE DEPLOIEMENT

4.1 - Déploiement de l'applicaïton



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 prévues. Que ce soit la commande par le client, la consultation de la recette par le préparateur de la commande ou encore la consultation des factures par le dirigeant. Il est cependant envisagé de développer en complément une application mobile, destinée principalement aux livreurs pour le permettre d'avoir accès à la liste des commandes à livrer ou à la validation d'une livraison. En effet, les livreurs étant itinérants, ce type d'interface leur facilitera la tâche.

4.2 - Serveur de déploiement

L'application PIZZAPP sera déployée sur les serveurs de 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 PIZZAPP se présente sous la forme d'un « Projet » Django, lui-même divisé en deux « Applications » : Ventes et Production.

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

5.1.2 - Les couches

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

- une couche **Présentation** : responsable de l'interface client/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 fonctionnalités et algorithmes nécessaires au fonctionnement de l'application telles que : <ul style="list-style-type: none">- L'affichage du catalogue en fonction des pizzas disponibles- Choix de l'établissement en fonction de la distance du client
Modèle	Regroupe les classes utilisées par l'application « Ventes » pour manipuler les données. Ex : <ul style="list-style-type: none">- Pizza

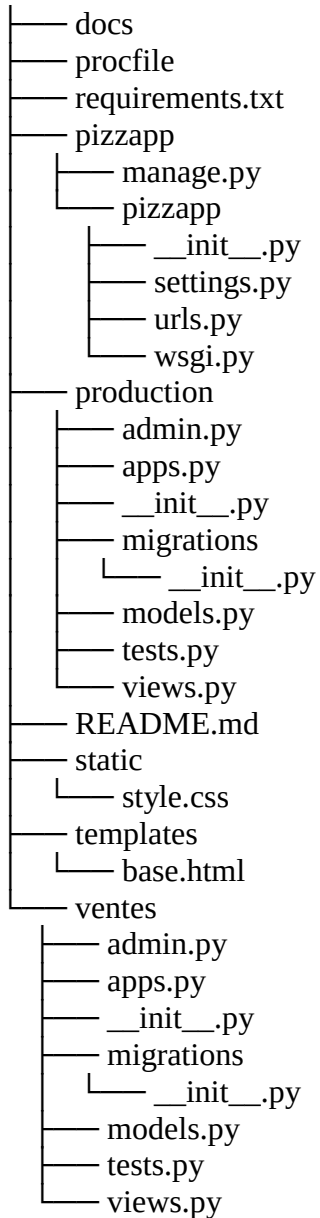


	<ul style="list-style-type: none">- Etablissement- Commande- Client
--	---

5.1.3 - Structure des sources

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

Racine





6 - POINTS PARTICULIERS

6.1 - Ressources

6.1.1 - Ressources graphiques

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

6.1.2 - Données

Les données de base (catalogue de pizzas, ingrédients, magasin) 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

SGBD	Système de gestion de base de données
DJANGO	Framework permettant la réalisation d'application web en Python.
IDE	Environnement de développement intégré.