

	OC PIZZA	
	APPLICATION PIZZAPP	
	DOSSIER DE CONCEPTION TECHNIQUE	version 1.0

TABLE DES MATIERES

I. VERSIONS

Tableau	3
---------	---

II. INTRODUCTION

2.1 OBJET DU DOCUMENT	3
2.2 REFERENCES	3

III. ARCHITECTURE TECHNIQUE

3.1 COMPOSANTS GENERAUX	4
Diagramme de composants	4
3.2 BASE DE DONNEES	5
3.3 MODELE	5
Modèle physique de données	5

IV. ARCHITECTURE DE DEPLOIEMENT

Diagramme de déploiement	6
4.1 SERVEUR DE DEPLOIEMENT	6

V. ARCHITECTURE LOGICIELLE

5.1 PRINCIPE GENERAUX	7
5.1.1 Les couches	7
5.1.1 Structure des sources	8

VI. POINTS PARTICULIERS

6.1 RESSOURCES	9
6.2 PROCEDURE DE PACKAGING / LIVRAISON	9

VII. GLOSSAIRE

Tableau	10
---------	----

I. VERSIONS

AUTEUR	DATE	DESCRIPTION	VERSION
Villatte Lorelei	14/02/2019	Création du dossier de conception fonctionnelle	1.0

II. INTRODUCTION

2.1 OBJET DU DOCUMENT

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

Objectif du document : présenter les outils et technologies de l'application

Les éléments du présent dossier découlent :

- De l'entretien réalisé avec le dirigeant d'**OCPIZZA** en date du 01/09/2018
- De l'analyse des besoins effectués par l'équipe d'**OPENCLASSROOMS**.

2.2 REFERENCES

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

- P9 - DCF - 1.0 : Dossier de conception fonctionnelle de l'application
- P9 - DE - 1.0 : Dossier d'exploitation de l'application

III. ARCHITECTURE TECHNIQUE

3.1 COMPOSANTS GENERAUX

Développement de l'application avec le langage de programmation Python version 3.6 et le framework Django version 2.1.

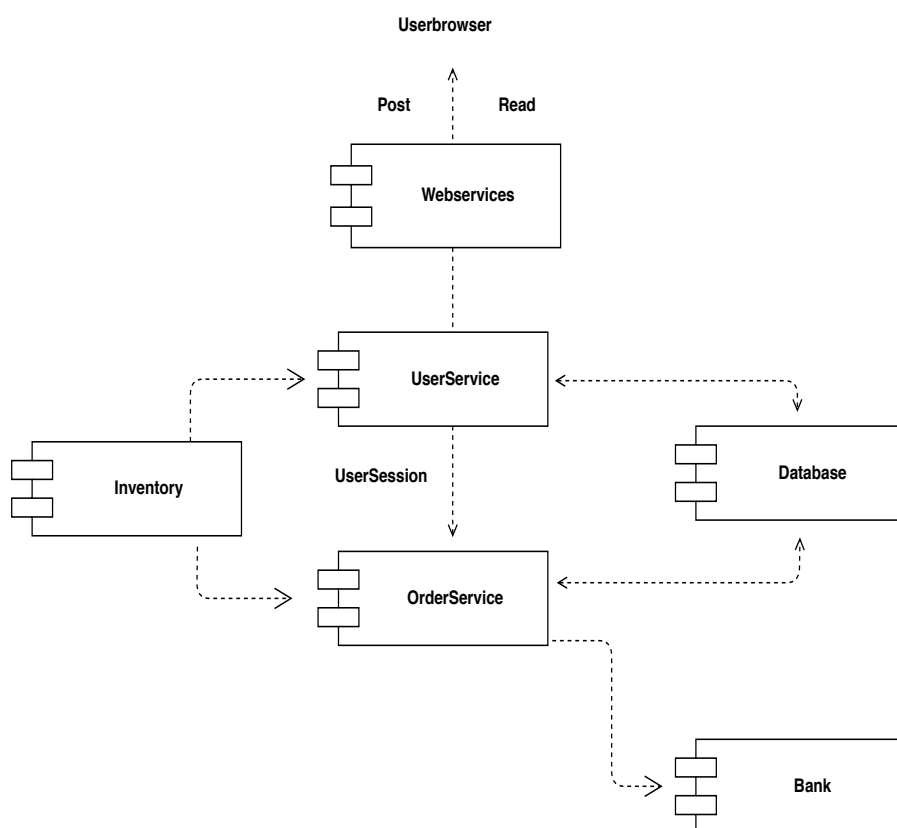
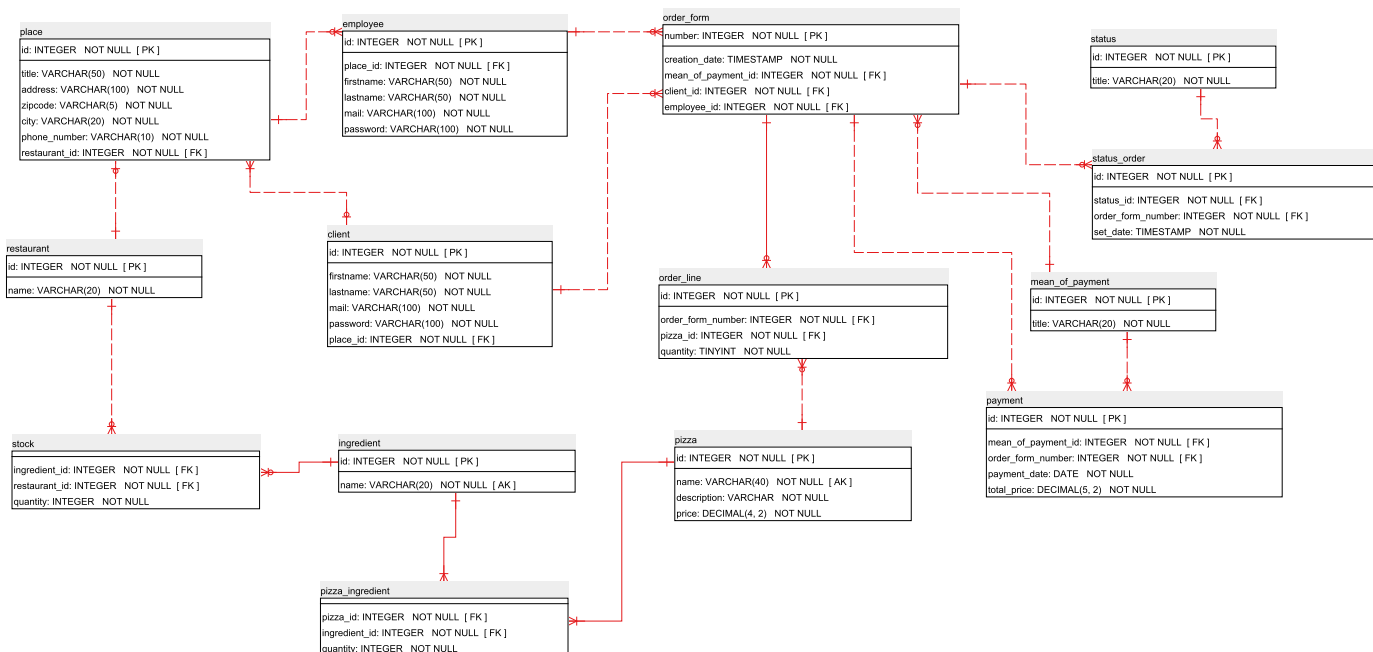


Diagramme de composants

3.2 BASE DE DONNEES

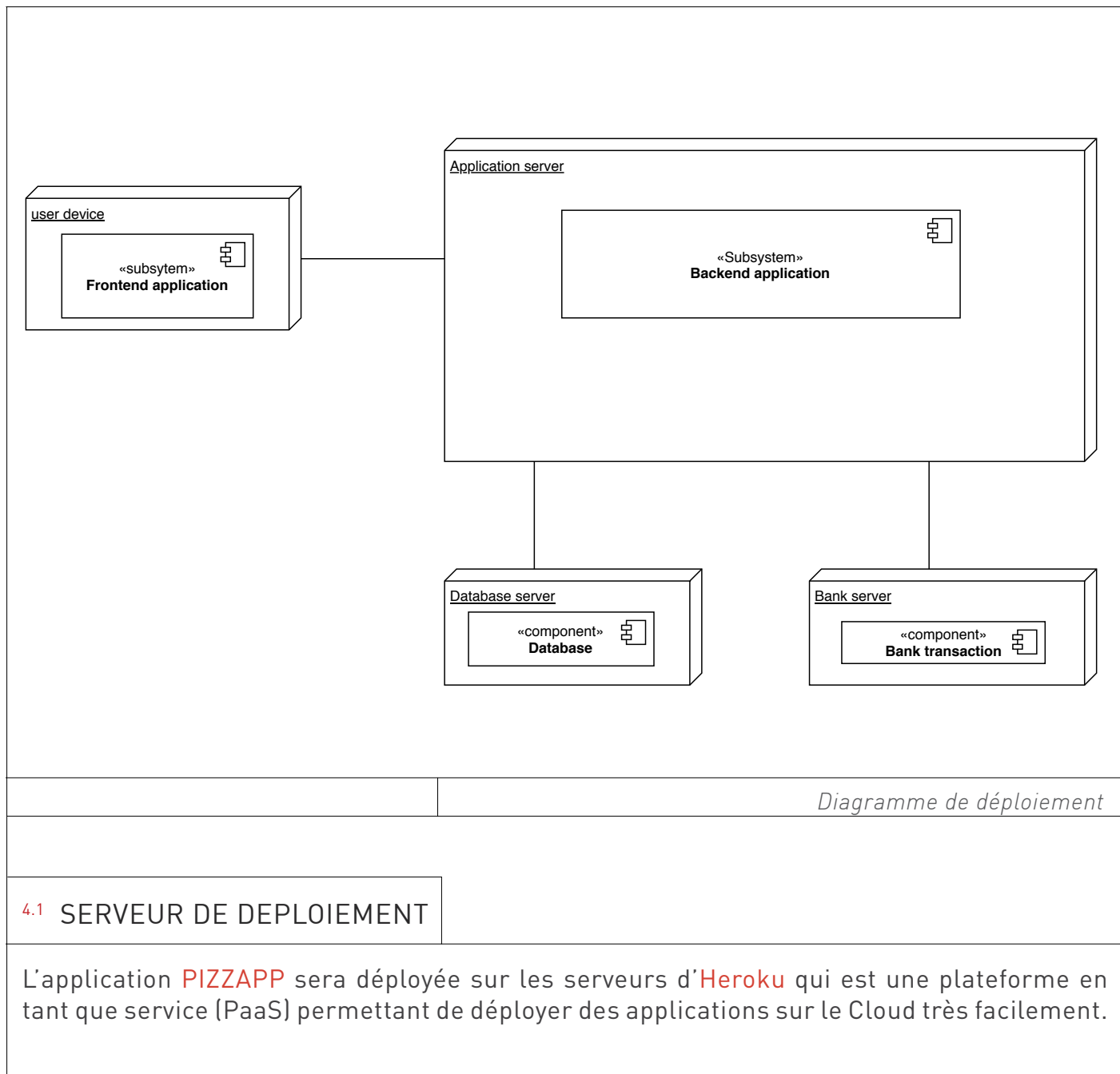
Utilisation du système de gestion de base de données relationnelle PostgreSQL, en raison de sa compatibilité avec la plateforme de déploiement.

3.3 MODELE



Modèle physique de données

IV. ARCHITECTURE DE DEPLOIEMENT



V. ARCHITECTURE LOGICIELLE

5.1 PRINCIPE GENERAUX

Les sources et versions du projet sont gérées par Git, les dépendances et le packaging par un fichier requirements.txt et Procfile

Le développement de l'application est réalisé avec le framework Django divisé en deux applications **Production** et **Vente** en respectant l'architecture MVT Modèle Vue Template.

5.1.1 Les couches

L'architecture applicative est la suivante :

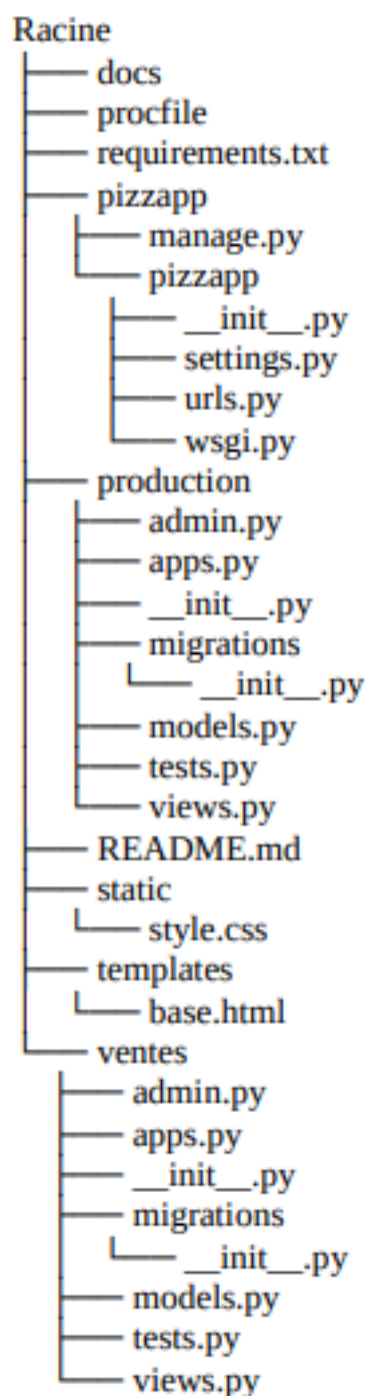
Une couche **[présentation]** en contact avec l'utilisateur de l'application.

Une couche **[métier]** implémente les règles de gestion de l'application., Cette couche utilise des données provenant de l'utilisateur via la couche **[présentation]** et du SGBD via la couche **[modèle]**.

Une couche **[modèle]** gère l'accès aux données du SGBD.

5.1.2 Structure des sources

les répertoires sources sont créés de façon à respecter l'architecture MVT d'un projet Django



VI. POINTS PARTICULIERS

6.1 RESSOURCES

Les ressources graphiques ainsi que les données nécessaires à la réalisation de la base de données sont fournies par la direction **OCPIZZA**.

6.2 PROCEDURE DE PACKAGING / LIVRAISON

Déploiement de l'application sur la plateforme Heroku.
Se référer au dossier d'exploitation pour la marche à suivre.

VIII. GLOSSAIRE

SGBD	Système de Gestion de Base de Données
PAAS	Platform As A Service