



OC Pizza

OCPizzaApp

Dossier de conception technique

Version 1.0

Auteur

Martin Gaucher

Analyste développeur

TABLE DES MATIÈRES

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 - Système de Gestion de Base de Donnée.....	5
4 - Architecture de Déploiement.....	6
4.1 - Système de Gestion de Base de Données.....	6
4.2 - Déploiement sur Heroku.....	6
5 - Architecture logicielle.....	8
5.1 - Principes généraux.....	8
5.1.1 - Les couches: architecture trois tiers.....	8
5.1.2 - Les modules.....	8
5.1.3 - Structure des sources.....	8
6 - Points particuliers.....	10
6.1 - Ressources.....	10
6.1.1 - Données.....	10
6.1.2 - Éléments graphiques.....	10
6.2 - Environnement de développement.....	10
6.3 - Procédure de packaging / livraison.....	10

1 - VERSIONS

Auteur	Date	Description	Version
Martin Gaucher	18/02/2020	Création du document	1.0

2 - INTRODUCTION

2.1 - Objet du document

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

L'objectif du document est de présenter la solution technique en plusieurs points.

Les éléments du présent dossier découlent :

- de la demande du client
- du développement de l'application

2.2 - Références

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

1. **MG - FO – 1.1** : Dossier de conception fonctionnelle de l'application
2. **MG – DE – 1.0** : Dossier d'exploitation de l'application

3 - ARCHITECTURE TECHNIQUE

3.1 - Application Web

L'application sera développée en utilisant les langages suivants:

- Python (version 3.7)
- Javascript (avec la librairie JQuery) pour tout ce qui est lié aux animations et les requêtes asynchrones avec AJAX.

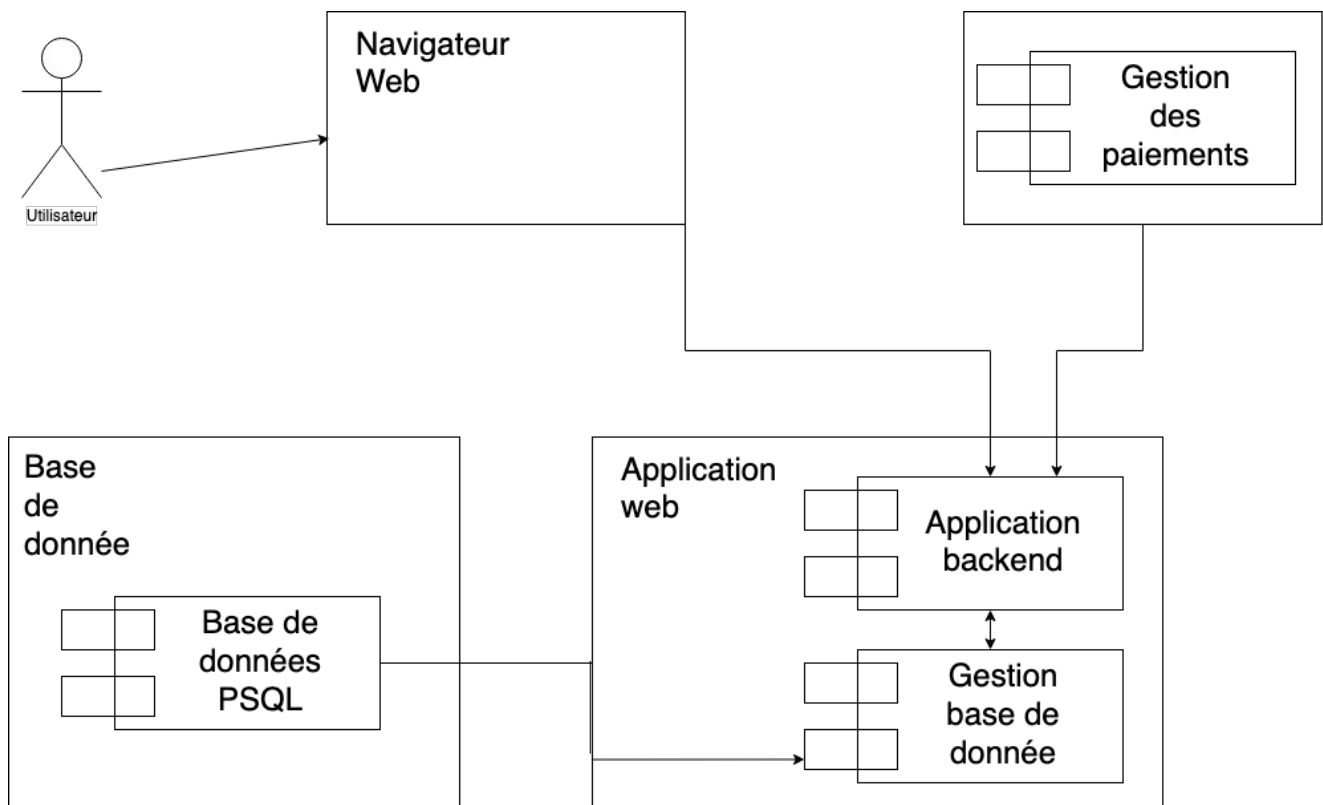
Nous allons utiliser le framework Django (version la plus récente et stable).

3.2 - Système de Gestion de Base de Donnée

Le système de base de donnée utilisé sera PSQL (PostGreSql). Car compatible avec Django et pour ses performances.

On peut retrouver le modèle physique de données dans le dossier de conception fonctionnelle.

4 - ARCHITECTURE DE DÉPLOIEMENT



Le diagramme de déploiement ci dessus montre le cheminement employé par l'application et tout ce qui tourne autour. L'utilisateur devra passer par l'interface web et la base de donnée sera sollicitée par cette dernière pour afficher les informations voulues. Il y a également le système de gestion bancaire qui sera appelé lors des transactions entre le client et l'application web.

4.1 - Système de Gestion de Base de Données

Le Système de Gestion de Base de Données (SGBD) sera PSQL (PostgreSQL). PostgreSQL est recommandé pour travailler avec les applications web. Pour quelles raisons ? Voici les avantages (par rapport à MySQL considérant que ces derniers sont les meilleurs SGBD pour les applications web) qu'offrent PSQL :

- Plus fiable et l'intégrité des données y est plus performantes.
- Tout ce qui est lié aux requêtes est meilleur avec PSQL. PSQL dispose d'un planificateur de requêtes sophistiqué et d'un optimiseur de requête.
- La documentation de PSQL est plus dense et plus complète et de façon générale le support est meilleur.
- Argument non technique : PSQL ne dépend pas d'un contributeur principal privé comme c'est le cas pour MySQL avec Oracle.
- Peut gérer de nombreux utilisateurs et données. Il suffit de bien configurer PSQL.

4.2 - Déploiement sur Heroku

Nous allons déployer l'application sur Heroku, une PaaS (Plateforme en tant que service) qui nous permettra d'économiser des ressources en terme de personnel qualifié et d'infrastructures. Car il n'y a qu'à lancer l'application sur Heroku avec git, et du côté serveur Heroku s'occupe de tout.

5 - ARCHITECTURE LOGICIELLE

5.1 - Principes généraux

L'architecture logicielle sera découpée en deux parties:

- Production
- Ventes

La partie Production sera utilisée du côté des pizzaiolos, des livreurs et du patron.

La partie Ventes sera essentiellement utilisée par les clients mais aussi par les vendeurs.

5.1.1 - Les couches: architecture trois tiers

Première couche – La couche de présentation:

Cette couche correspond à l'Interface Homme Machine (IHM). C'est la partie visible de l'application par l'utilisateur.

Deuxième couche – La couche de traitement:

C'est la partie fonctionnelle de l'application qui traite les informations reçues par la couche de présentation.

Troisième couche – La couche d'accès aux données

C'est la gestion de l'accès aux données persistantes.

5.1.2 - Les modules

Deux « modules » (appelé applications dans Django) seront implémentés dans l'application : Ventes et Production.

5.1.3 - Structure des sources

OCPizzApp

```
|-----docs
|-----profile
|-----requirements.txt
|-----ocpizzapp
|----README.md
|-----manage.py
|-----ocpizzapp
|-----__init__.py
|-----settings.py
|-----urls.py
|-----wsgi.py
```

```
|-----ventes
|-----admin.py
|-----apps.py
|-----__init__.py
|-----migrations
|-----__init__.py
|-----models.py
|-----tests.py
|-----views.py
|-----production
|-----admin.py
|-----apps.py
|-----__init__.py
|-----migrations
|-----__init__.py
|-----models.py
|-----tests.py
|-----views.py
|-----static
|-----styles.css
|-----template
|-----index.html
```


6 - POINTS PARTICULIERS

6.1 - Ressources

6.1.1 - Données

Les données (pizzas, ingrédients, base de clients...) seront fournies par OCPizza. Il faudra intégrer ces données dans la base de données.

6.1.2 - Éléments graphiques

Tous les éléments graphiques seront également fournis par la société OCPizza. Il faudra également les intégrer en respectant les maquettes clients.

6.2 - Environnement de développement

L'environnement de développement sera sur une machine pourvue d'un OS permettant le développement de l'application (Windows, Linux...). Il faudra y installer un environnement virtuel afin de rendre maintenable l'application. Un outil de versionning est également fortement souhaité si une équipe travaille dessus. Git sera l'outil utilisé en tant qu'outil collaboratif.

6.3 - Procédure de packaging / livraison

La livraison finale se fera au moment du déploiement de l'application sur Heroku. Avec ceci, on pourra y ajouter la documentation nécessaire pour se servir de l'application. Il faudrait également prévoir une formation pour les pizzaiolos, les vendeurs, les livreurs et le patron.