

OC Pizza

Système informatique OC Pizza

Dossier de conception technique

Version 1.0

Auteur

Arnaud Laval
Analyst Programmer

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 - Composants.....	5
3.1.1 - <i>Application Web</i>	5
3.1.1.1 - Authentification.....	5
3.1.1.2 - Passer commande.....	5
3.1.1.3 - Suivi et Gestion de commande.....	5
3.1.1.4 - Produits et Ingrédients.....	5
3.1.1.5 - Gestion restaurant.....	5
3.1.1.6 - Gestion des comptes Utilisateurs.....	5
3.1.2 - <i>Base de données PostgreSql</i>	5
3.1.3 - <i>Api de paiement externe Stripe (Banque)</i>	5
3.2 - Application Web.....	6
3.2.1 - <i>Serveur Web</i>	6
3.3 - Base de Données.....	7
4 - Architecture de Déploiement.....	8
5 - Architecture logicielle.....	9
5.1 - Principes généraux.....	9
5.1.1 - <i>Les couches</i>	9
5.1.2 - <i>Les modules</i>	9
5.1.3 - <i>Structure des sources</i>	9
6 - Points particuliers.....	10
6.1 - Gestion des logs.....	10
6.2 - Fichiers de configuration.....	10
6.3 - Environnement de développement.....	10
6.4 - Procédure de packaging / livraison.....	10

1 - VERSIONS

Auteur	Date	Description	Version
Arnaud Laval	04/03/2020	Création du document	1.0

2 - INTRODUCTION

2.1 - Objet du document

Le présent document constitue le dossier de conception technique du système informatique de gestion et de vente en ligne de pizza.

Objectif du document : définir la démarche de conception de l'application OC Pizza.

2.2 - Références

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

1. Système informatique OC Pizza - Dossier de conception fonctionnelle
2. Système informatique OC Pizza - Dossier d_exploitation

3 - ARCHITECTURE TECHNIQUE

3.1 - Composants

3.1.1 - Application Web

3.1.1.1 - Authentification

Ce sous-composant gère de login et de logout. On y vérifie les identifiants de connexion et les mots de passe des utilisateurs en interrogeant la base de données.

3.1.1.2 - Passer commande

Ce sous-composant gère toute la partie de prise de commande, de la conception du panier à sa validation et au paiement.

3.1.1.3 - Suivi et Gestion de commande

Ce sous-composant gère la partie suivant la prise de commande. On y retrouve la fonctionnalité de modification de commande (qui renverra dans la partie « Passer commande ») ainsi que l'affichage de ces dernières, que ce soit pour les employés (toutes les commandes en cours) ou les clients (leurs propres commandes).

3.1.1.4 - Produits et Ingrédients

Ce sous-composant gère le stock d'ingrédients, la liste des produits finis (pizzas).

3.1.1.5 - Gestion restaurant

Ce sous-composant gère les fonctionnalités propre aux gérants comme l'ajout de nouveaux produits à la carte, les promotions et le suivi des résultat.

3.1.1.6 - Gestion des comptes Utilisateurs

Ce sous-composant gère les créations de comptes ainsi que les modifications et les suppressions. Son propre compte utilisateur pour un client et les comptes employés pour un gérant.

3.1.2 - Base de données PostgreSql

Ce composant représente la base de données PostgreSQL de l'application, « ocpizza_db ».

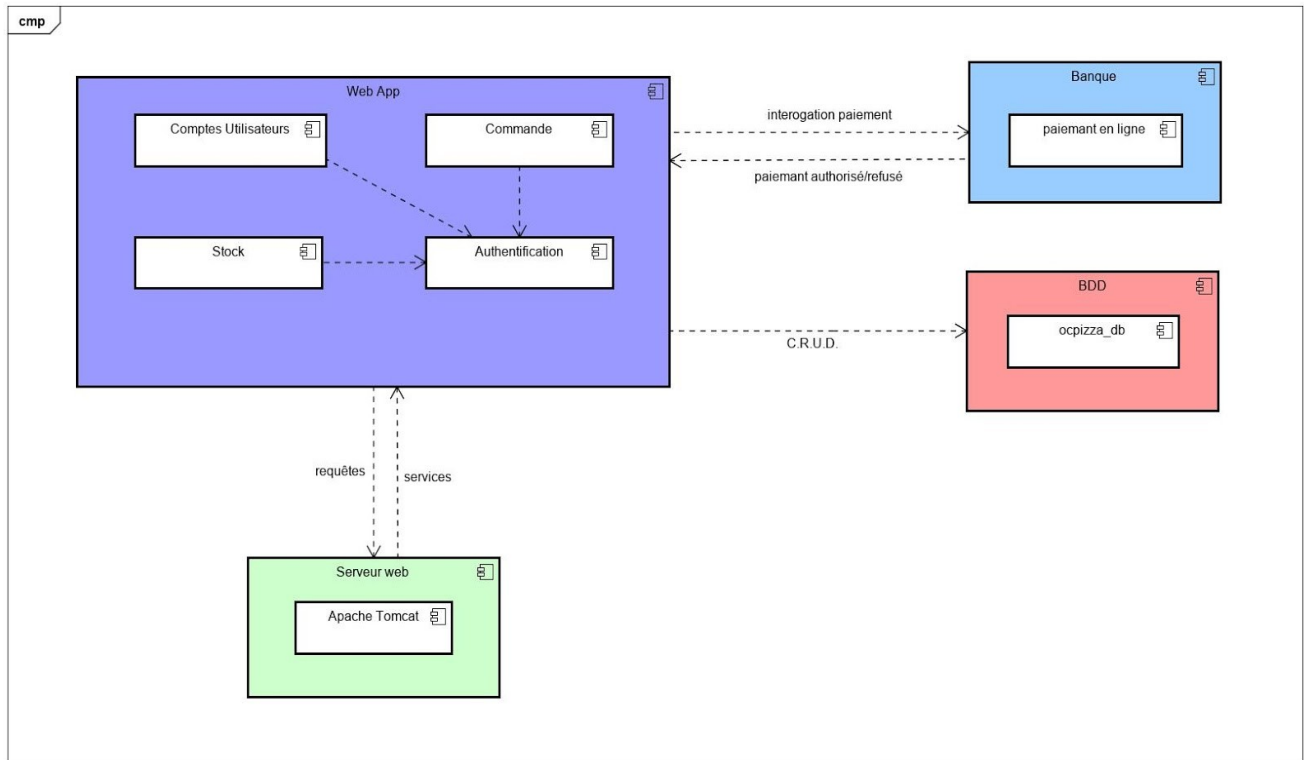
3.1.3 - Api de paiement externe Stripe (Banque)

Ce composant représente l'API externe Stripe, qui gère les paiements. Il s'agit d'un composant externe.

3.2 - Application Web

La pile logicielle est la suivante :

- Application **J2EE** (JDK version 1.8) / Html 5 / Css 3 / Javascript.
- Bdd PostgreSQL.
- Serveur d'application Tomcat.



3.2.1 - Serveur Web

Il contient l'application web et rend accessibles ses pages web depuis un client (comme un navigateur web) grâce au protocole Http.

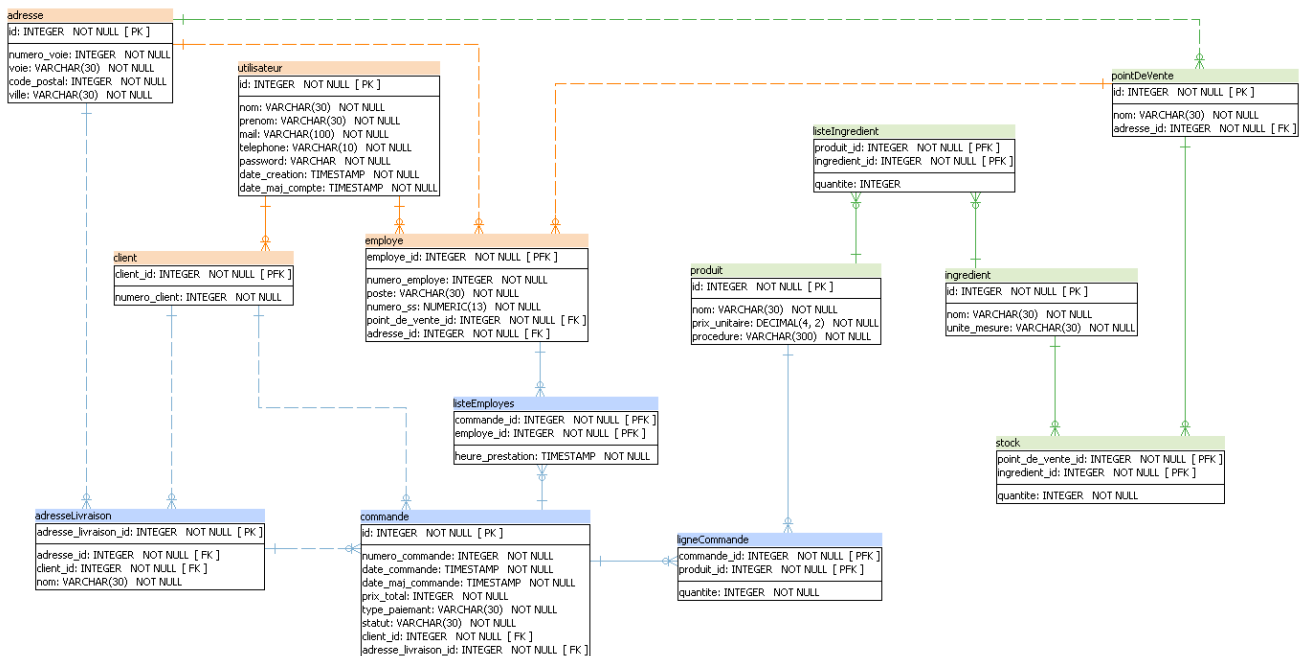
3.3 - Base de Données

Nous utiliserons l'interface de programmation JDBC (Java Database Connectivity).

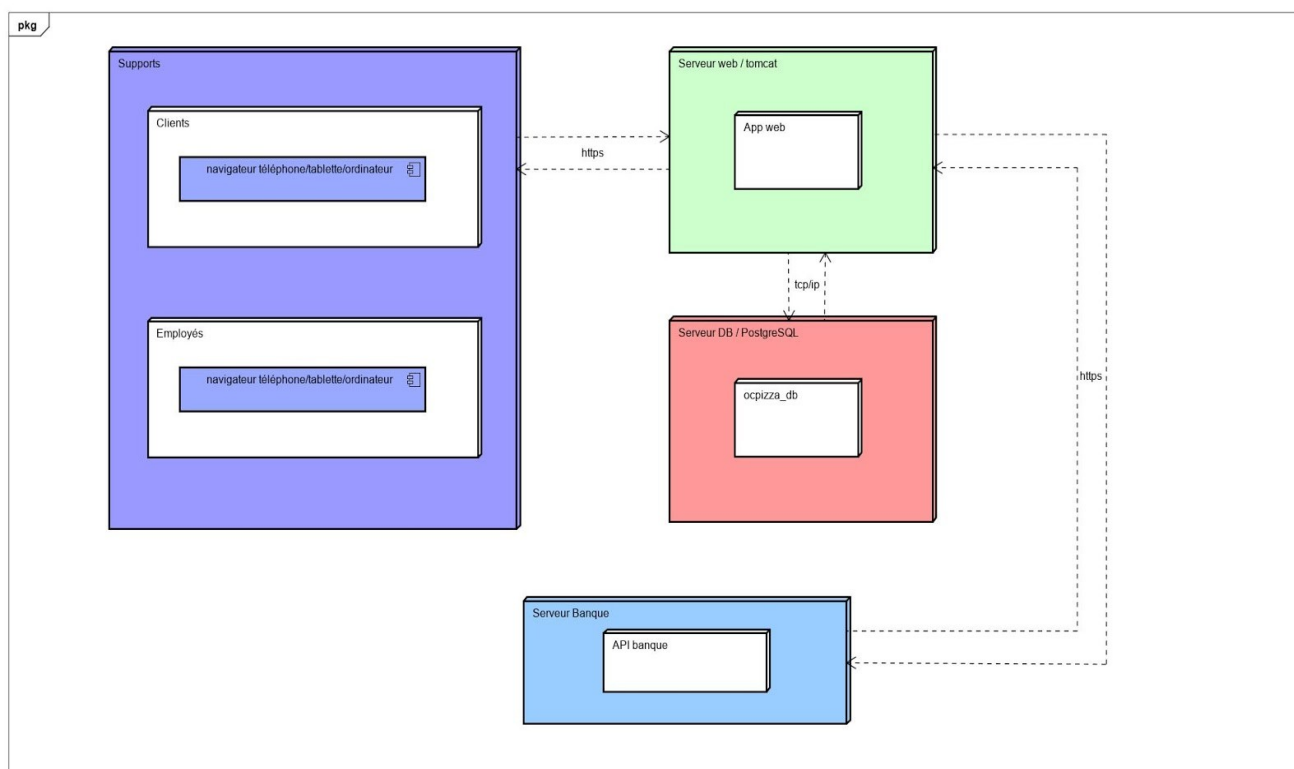
Notre SGBDR (Système de Gestion de Base de Données Relationnelle) est PostgreSQL 10.3.

L'administrateur de cette base de données se définit par :

- Identifiant : admin
- Mot de passe : admin



4 - ARCHITECTURE DE DÉPLOIEMENT



La base de données et le serveur web qui contiendra l'application seront déployés sur un serveur vps (virtual private server).

5 - ARCHITECTURE LOGICIELLE

5.1 - Principes généraux

Les sources et versions du projet sont gérées par **Git**, les dépendances et le packaging par **Apache Maven**.

5.1.1 - Les couches

L'architecture applicative est la suivante :

- une couche **business** : responsable de la logique métier du composant
- une couche **model** : implémentation du modèle des objets métiers
- une couche **consumer** : responsable de la persistance des données
- une couche **web** : regroupe les parties controller et vue du MVC pattern. (...)

5.1.2 - Les modules

Un module par couche applicative : module business/model/consumer/web.

5.1.3 - Structure des sources

La structuration des répertoires du projet suit la logique suivante :

- les répertoires sources sont créés de façon à respecter la philosophie Maven (à savoir : « convention plutôt que configuration »)

```
OC Pizza
├── pom.xml
├── OC Pizza - model
│   ├── pom.xml
│   └── src
│       ├── main
│       │   ├── java
│       │   └── resources
│       └── test
│           ├── java
│           └── resources
├── OC Pizza - business
│   ├── pom.xml
│   └── src
│       ├── main
│       │   ├── java
│       │   └── resources
│       └── test
│           ├── java
│           └── ressources
├── ...
├── src
└── lib
```

6 - POINTS PARTICULIERS

6.1 - Gestion des logs

Les logs seront gérés en utilisant le librairie Log4j2.

6.2 - Fichiers de configuration

L'application sera développée avec Spring Boot. Nous utiliseront un fichier configuration.properties placé à la racine du projet dans lequel nous aurons les propriétés du projet comme la configuration pour l'accès à la base de données.

6.3 - Environnement de développement

Le développement de l'application se fera avec IntelliJ Ultimate 2019.3.3 .

6.4 - Procédure de packaging / livraison

L'application sera packagée dans un fichier .WAR .