



OC PIZZA

Système informatique OC PIZZA

Dossier de conception technique

Version 1.0

Auteur

M.COZ

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 - Composants généraux.....	5
3.1.1 - Application.....	5
3.1.1.1 - Composant Authentification.....	5
3.1.1.2 - Composant Panier.....	5
3.1.1.3 - Composant Moteur de recherche.....	5
3.1.1.4 - Composant Produits.....	5
3.1.1.5 - Composant Stock	5
3.1.1.6 - Composant Commande.....	5
3.1.1.7 - Composant Employee.....	5
3.1.2 - Package Base de données.....	5
3.1.2.1 - Composant BDD	5
3.2 - Application Web	6
3.3 - Application Base de données	7
3.3.1 - Système de Gestion de Base de Données.....	7
3.3.1.1 - Digramme de classe du domaine :.....	7
3.3.1.2 - Modèle Physique de Données :.....	8
4 - Architecture de Déploiement	9
4.1 - Serveur de Base de données.....	9
4.2 - Serveur d'application	9
5 - Architecture logicielle	10
5.1 - Principes généraux.....	10
5.1.1 - Les couches	10
5.1.2 - Les modules	10
5.1.3 - Structure des sources.....	10
6 - Points particuliers	12
6.1 - Gestion des logs	12
6.2 - Fichiers de configuration.....	12
6.2.1 - Application web.....	12
6.3 - Environnement de développement.....	12
6.4 - Procédure de packaging / livraison.....	12

1 - VERSIONS

Auteur	Date	Description	Version
M.COZ	20/04/2020	Création du document	1.0

2 - INTRODUCTION

2.1 - Objet du document

Le présent document constitue le dossier de conception technique du projet de création d'un système informatique de gestion et de vente en ligne de l'entreprise OC 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. **Projet OC PIZZA – Dossier de conception fonctionnelle** : Dossier de conception fonctionnelle de l'application
2. **Projet OC PIZZA – Dossier d'exploitation** : Dossier d'exploitation de l'application
3. **Projet OC PIZZA – PV** : Procès Verbal

3 - ARCHITECTURE TECHNIQUE

3.1 - Composants généraux

3.1.1 - Application

3.1.1.1 - Composant Authentification

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

3.1.1.2 - Composant Panier

Ce sous-composant gère le panier, de sa constitution jusqu'à sa validation.

3.1.1.3 - Composant Moteur de recherche

Ce sous-composant gère toutes les recherches dans le catalogue.

3.1.1.4 - Composant Produits

Ce sous-composant gère tous les produits disponibles.

3.1.1.5 - Composant Stock

Ce sous-composant gère tous les stocks disponibles.

3.1.1.6 - Composant Commande

Ce sous-composant gère toute la partie de prise de commande, de sa validation à la livraison.

3.1.1.7 - Composant Employee

Ce sous-composant permet la gestion administrative des employés.

3.1.2 - Package Base de données

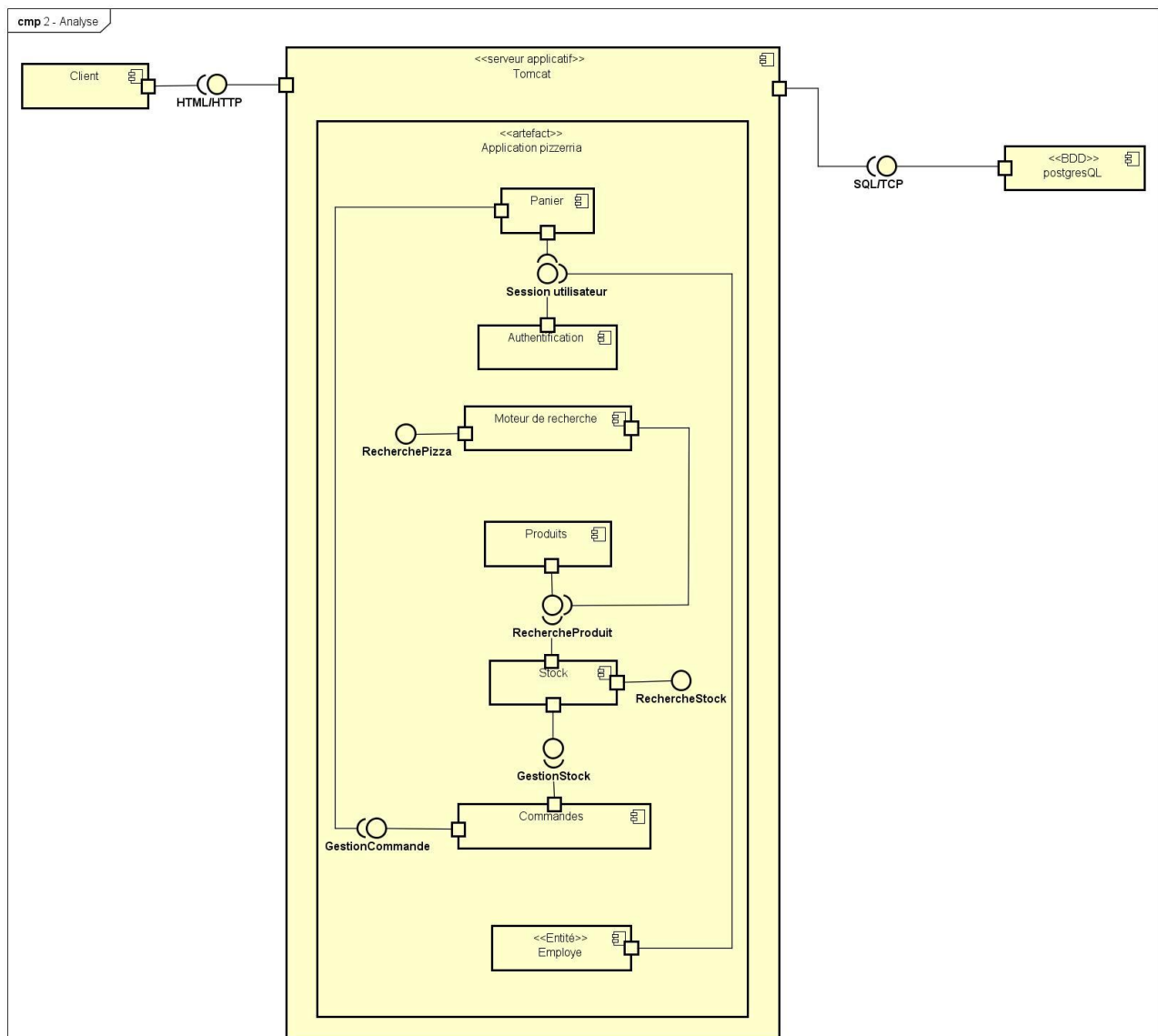
3.1.2.1 - Composant BDD

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

3.2 - Application Web

La pile logicielle est la suivante :

- Application **J2EE** (JDK version 1.8) / HTML5 / CSS3 / Javascript
- Serveur d'application **TOMCAT v9**
- Base de données **PostgreSQL v11.5.2**



3.3 - Application Base de données

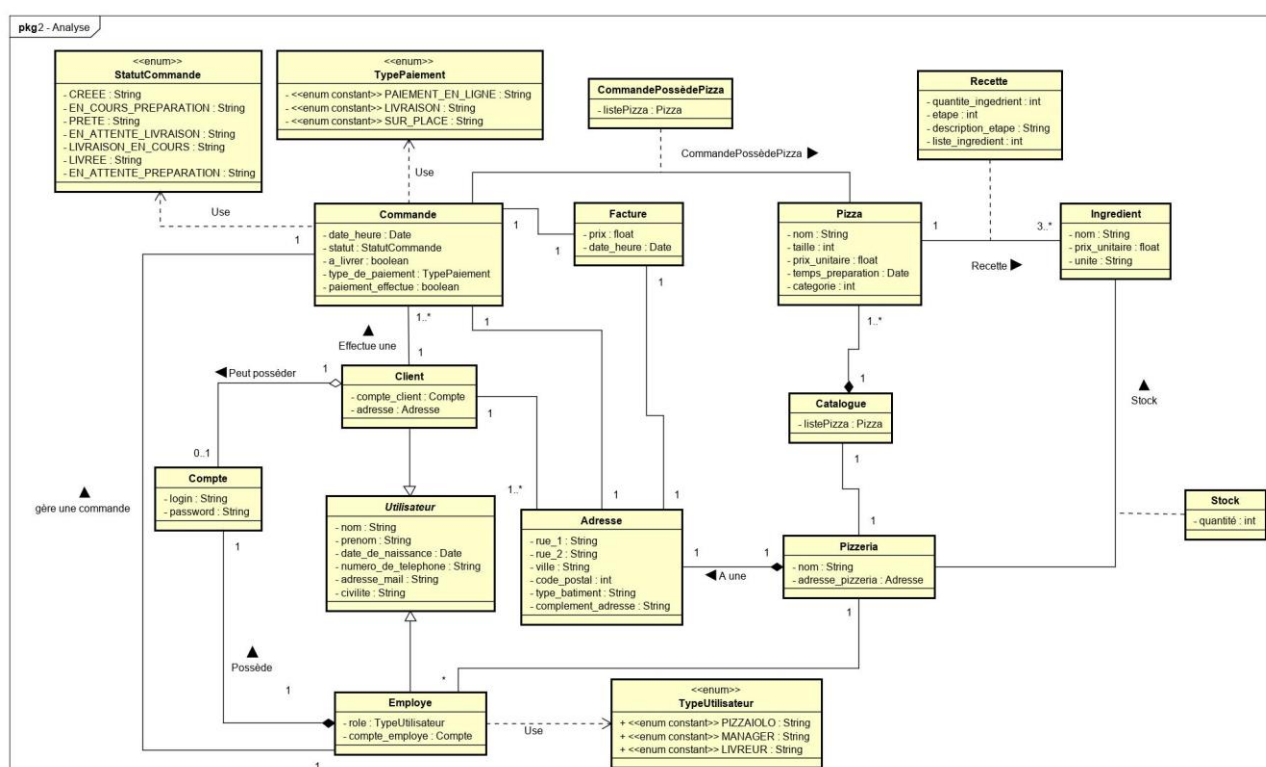
3.3.1 - Système de Gestion de Base de Données

Un Système de Gestion de Base de Données (SGBD) est un logiciel qui permet le stockage d'informations dans une base de données. Il permet de créer, lire, mettre à jour, et supprimer les données qui sont contenues dans la base de données. Ces données seront stockées dans le SGBD Relationnelles **PostgreSQL**, qui est une **base de données open source, populaire et robuste**.

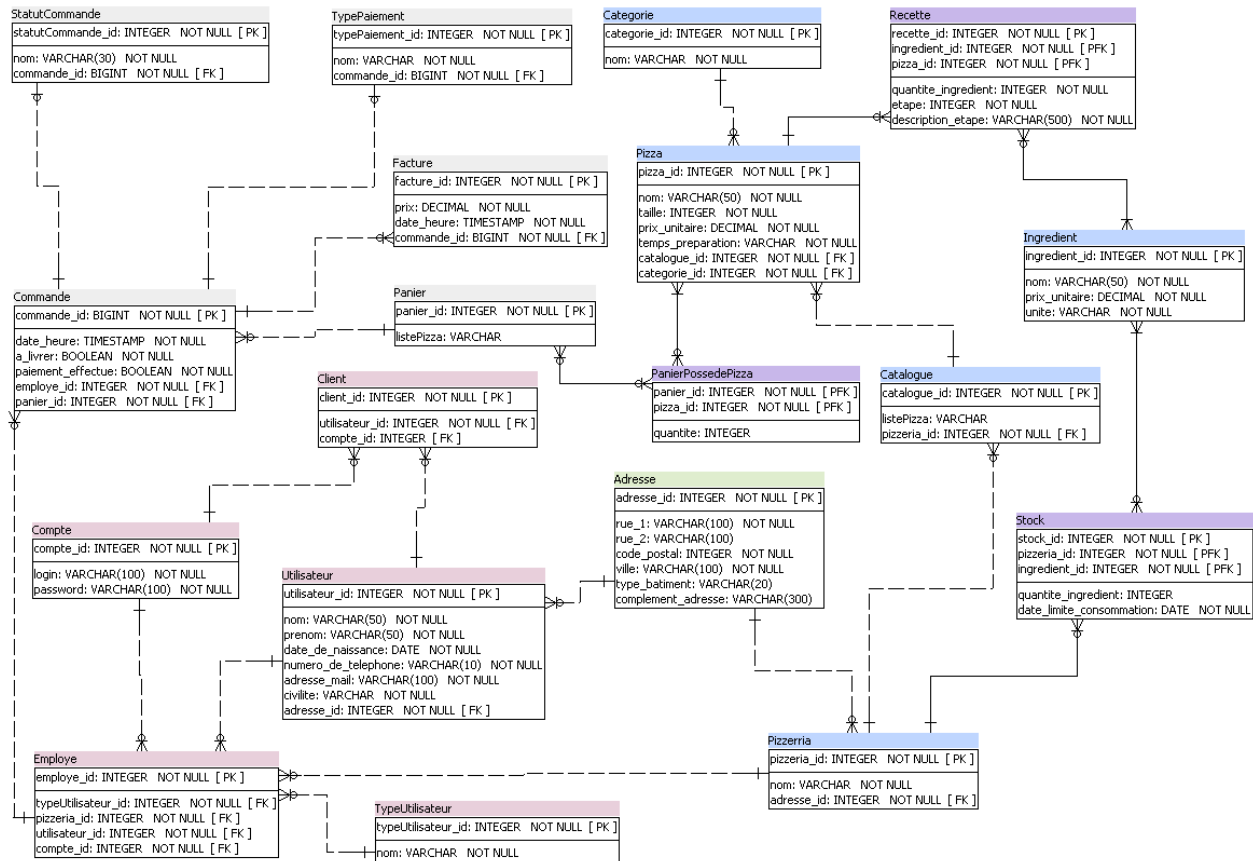
Nous utiliserons l'interface de programmation JDBC (Java Database Connectivity).

3.3.1.1 - Diagramme de classe du domaine :

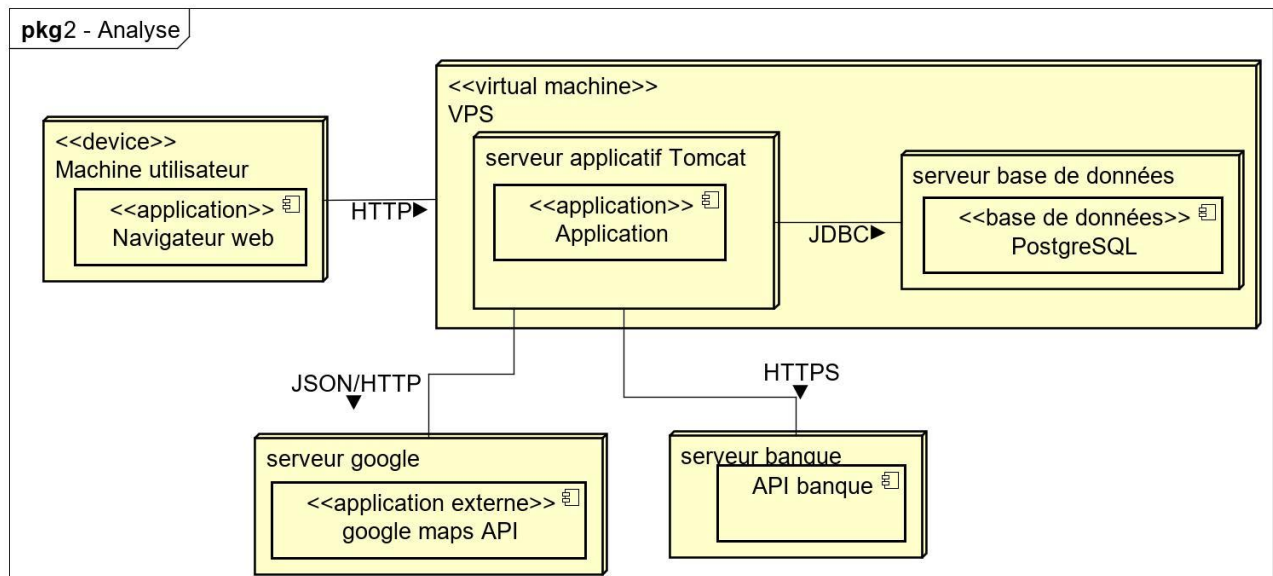
Le modèle physique de données (MPD) a été conçu sur l'analyse du diagramme de classe UML suivant :



3.3.1.2 - Modèle Physique de Données :



4 - ARCHITECTURE DE DEPLOIEMENT



4.1 - Serveur de Base de données

La partie Base De Données (BDD) sera déployée sur une machine virtuelle. Nous avons sélectionné la base de données PostgreSQL pour répondre au besoin de l'application.

4.2 - Serveur d'application

Le serveur d'application choisi est Apache Tomcat. Il sera hébergé sur la machine virtuelle.

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 pattern MVC.

5.1.2 - Les modules

Il y a un module par couche applicative :

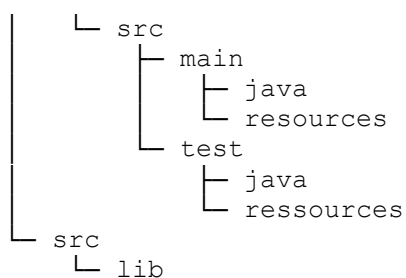
- Un module **business**.
- Un module **model**.
- Un module **consumer**.
- Un module **web**.

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 »)

```
ocpizza
├── pom.xml
├── ocpizza_model
│   ├── pom.xml
│   └── src
│       ├── main
│       │   ├── java
│       │   └── resources
│       └── test
│           ├── java
│           └── resources
└── ocpizza_business
    └── pom.xml
```



- ...

6 - POINTS PARTICULIERS

6.1 - Gestion des logs

Les logs seront gérés en utilisant la librairie Log4j2.

6.2 - Fichiers de configuration

Le développement sera réalisé avec l'utilisation du Framework Spring.

6.2.1 - Application web

Un fichier de configuration « application.properties » sera 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

L'environnement de développement utilisé par nos équipes est IntelliJ IDEA.

6.4 - Procédure de packaging / livraison

L'application sera packagée dans un fichier .WAR. L'ensemble des scripts de création de la base de données seront livrés dans un fichier zip.