|  |
| --- |
| **OC Pizza**  **Système de gestion de Pizzerias**  Dossier de conception technique  Version 1.0 |
| **Auteur**  Rémy VALLET  *Développeur d’application* |

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 généraux 5

3.1.1 -Package A 5

3.1.1.1 -Composant X 5

3.1.1.2 -Composant Y 5

3.1.2 -Package B 5

3.1.2.1 -Composant Z 5

3.2 -Application Web 5

3.2.1 -Composants X 5

3.2.2 -Composants Y et Z 5

3.3 -Application XXX... 5

4 -Architecture de Déploiement 6

4.1 -Serveur de Base de données 6

4.2 -Serveur XXX 6

5 -Architecture logicielle 7

5.1 -Principes généraux 7

5.1.1 -Les couches 7

5.1.2 -Les modules 7

5.1.3 -Structure des sources 7

5.2 -Application Web 8

5.3 -Application Xxx 8

6 -Points particuliers 9

6.1 -Gestion des logs 9

6.2 -Fichiers de configuration 9

6.2.1 -Application web 9

6.2.1.1 -Datasources 9

6.2.1.2 -Fichier xxx.yyy 9

6.2.2 -Application Xxx 9

6.3 -Ressources 9

6.4 -Environnement de développement 9

6.5 -Procédure de packaging / livraison 9

6.6 -XXX 9

7 -Glossaire 10

# Versions

|  |  |  |  |
| --- | --- | --- | --- |
| Auteur | Date | Description | Version |
| Rémy VALLET | 30/10/2020 | Création du document | 1.0 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Objet du document

Le présent document constitue le dossier de conception technique de l'application ***OC Pizza*** à l’attention des développeurs, à la maintenance applicative et à l’équipe technique.

L’objectif de l’analyse technique est de préciser les règles de gestion applicatives pour le développement et la maintenance du système d’information.

Les éléments concernés sont :

* L’architecture technique des composants généraux des différents packages.
* Les différents composants de l’application.
* L’architecture de déploiement de la solution.
* L’architecture logicielle.
* Les points particuliers mis en œuvre pour cette solution tel que les fichiers de configurations, les logs, les ressources, l’environnement ou la procédure de livraison.

## Références

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

1. **Projet\_8-Dossier\_de\_conception\_fonctionnelle** : Dossier de conception fonctionelle de l'application.
2. **Projet\_8-Dossier\_d\_exploitation**: Dossier d’exploitation de l’application.
3. **Projet\_8-PV\_Livraison**: Procès-verbal de la livraison finale.

# Architecture Technique

## Composants généraux

La modélisation et l’identifier des éléments composant le système d’information à mettre en place et leur interaction est illustré par le digramme de composants en figure 1.

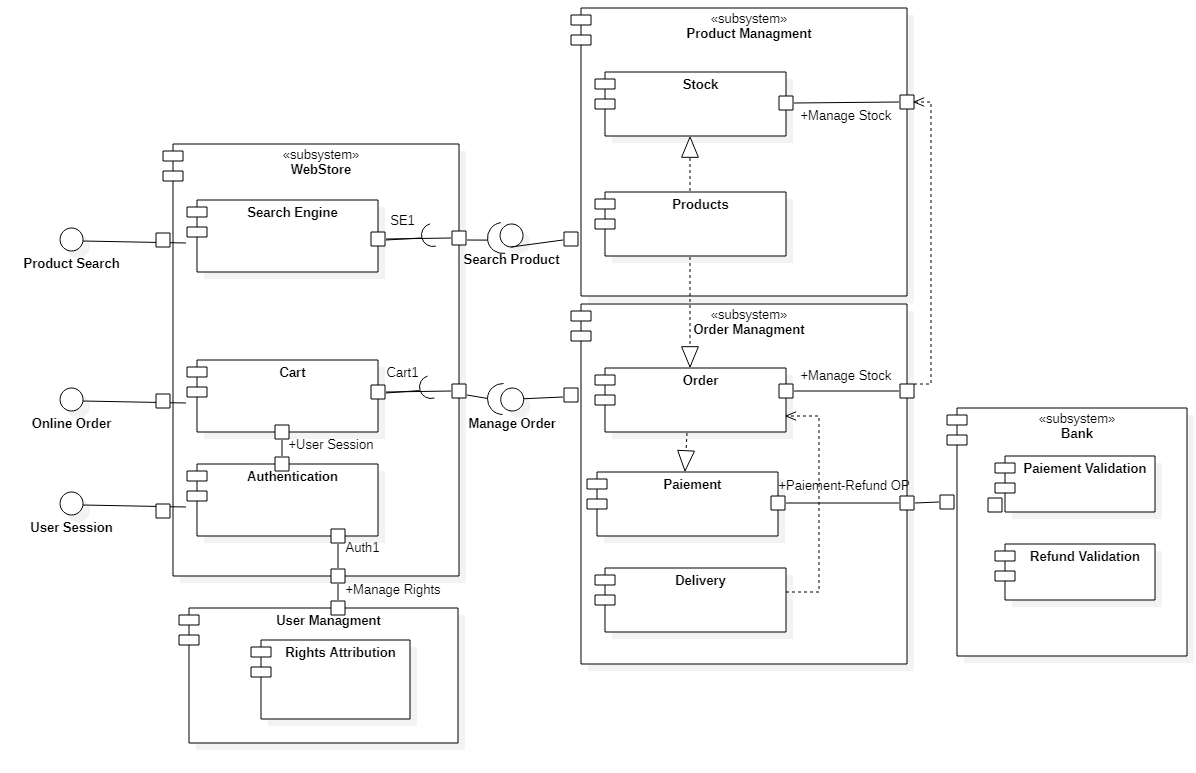


Figure 1: Diagramme de composants

### Package WebStore

Ce package regroupe les fonctionnalités nécessaires à la navigation sur l’application et à la prise de commande.

#### Composant Search Engine

Ce composant permet l’échange et le tri de données entre l’application et la base de données. Une API Rest qui attend en entrée des paramètres de requêtes définis et fournis une réponse paginé et trié selon les paramètres reçus. Les requêtes peuvent provenir d’un formulaire exposé à l’utilisateur de l’application (client, employé) ou d’un applicatif tiers (affichage du stock d’un produit, d’une liste de commande, d’une recette…)

#### Composant Cart

Ce composant permet la création d’une commande par un utilisateur. Selon le profil de la session utilisateurs il expose des options plus ou moins complètes, notamment la possibilité de passer une commande.

#### Composant Authentification

Ce composant permet l’authentification de l’utilisateur et la création de la session. Il interagit avec le package User Management pour l’attribution des droits de la session utilisateur.

### Package User Management

#### Composant Rights Attribution

Ce composant permet de remonter les droits utilisateurs sur la session lors de l’authentification.

### Package Product Management

#### Composant Stocks

Ce composant permet de mettre à jour les mouvements de stock et de répondre aux requêtes d’information de stock, notamment pour l’affichage des produits, l’affichage des recettes et le calcul de disponibilité des produits au panier lors de la validation de commande.

#### Composant Products

Ce composant permet l’affichage des produits. Il contient les informations détaillé des produits, notamment la constitution des produits composés. Il interagit avec le composant stock pour remonter le stock disponible.

### Package Order Management

#### Composant Order

Ce composants permet l’affichage des commandes et contient les informations relatives aux commandes, notamment les lignes de commandes, la validation et l’annulation de commande. Il interagit avec le composant stock pour envoyer les mouvements de stocks suite à une validation ou à une annulation de commande. Il est en charge de l’envoie d’une demande de paiement ou de remboursement.

#### Composant Paiement

Ce composant permet les traitements relatifs aux transactions bancaires de validation de paiement et de remboursement.

#### Composant Delivery

Ce composant permet les traitements relatifs à la livraison. Il remonte les informations à la partie commande.

### Package Bank

#### Composant Paiement Validation

Ce composant intègre l’API bancaire pour les demandes et réponse relatives aux transactions de paiement.

#### Composant Refund Validation

Ce composant intègre l’API bancaire pour les demandes et réponse relatives aux transactions de remboursement.

# Architecture de Déploiement

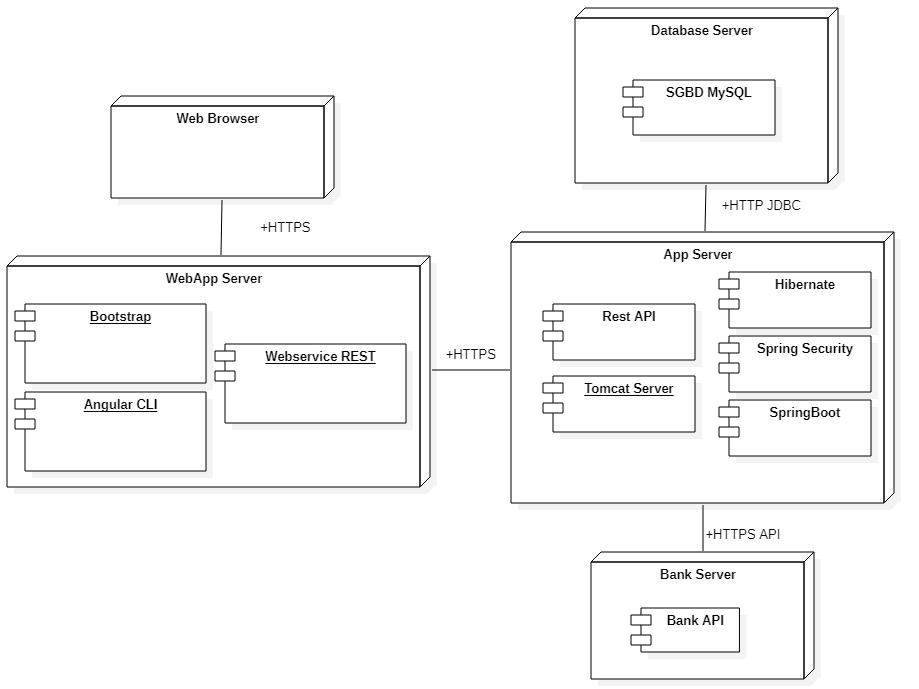


Figure 2: Diagramme de déploiement

TODO : Explication / commentaires si nécessaires...

## Serveur de Base de données

Les données sont persistées sur une base de données relationnelle SQL sur un serveur MySQL installé sur un système d’exploitation Linux centOS.

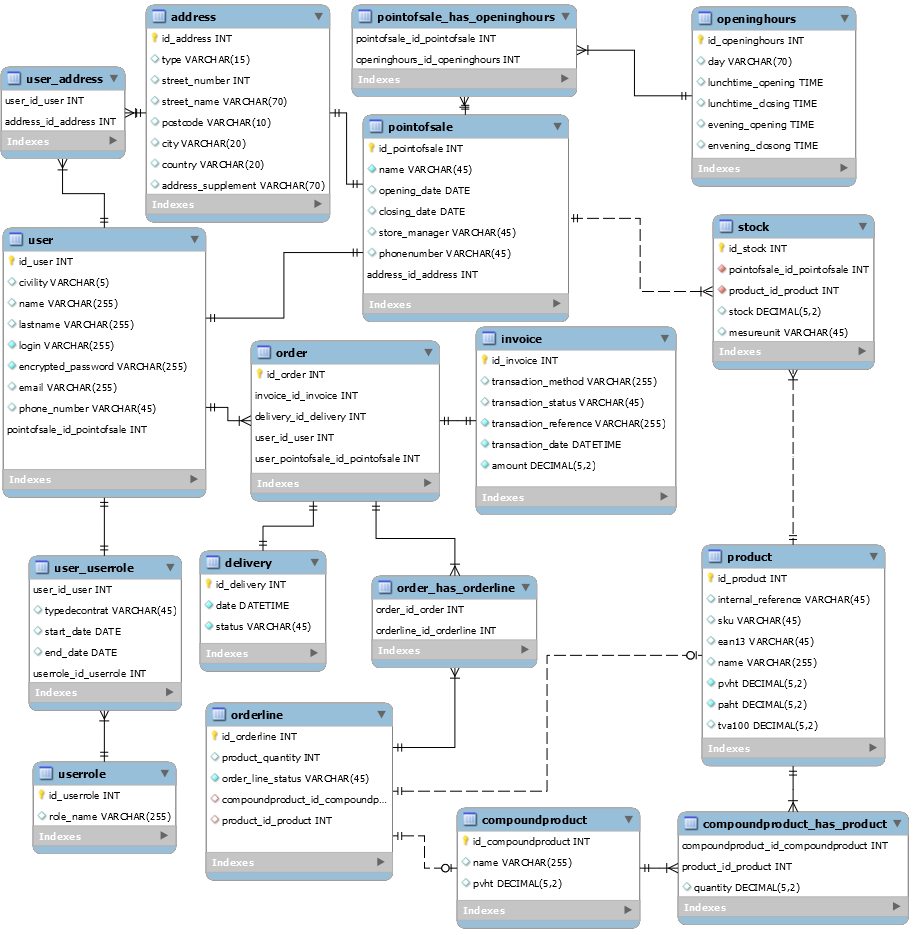


Figure 3Modèle Physique de Données (MPD)

## Serveur d’application Back-Office (BO) et Front-Middle-Office (FMO)

Le BO sera déployé sur un serveur apache Tomcat sous Linux centOS avec le FMO sous Angular CLI.

# Architecture logicielle

## Principes généraux

Les sources et versions du projet sont gérées par **GitLab.** Les dépendances et le packaging par **Apache Maven.** La persistance des données en BDD MySQL par **Hibernate**.

La solution s’articule autour de plusieurs modules de micro-services autonomes exposants des **API Rest** consommables par les autres modules.

Les micro services sont également documentés et exposés via **Swagger** sur une interface HTML accessibles aux développeurs.

### Les couches

L'architecture applicative de chaque micro-service est la suivante :

* unecouche **application** : responsable de la parie opérationnelle du micro-service. Permet d’exposer le métier et le model sans jamais entrer dans le business, de faire la transition entre les couches applicatives (interne/externe). Cette partie contient également le paramétrage et les fonctions nécessaires au lancement du service.
* unecouche **model** : implémentation du modèle des objets métiers. Gère le traitement et la logique métier des objets manipulés.
* une couche **infrastructure** : responsable de la partie des traitements en BDD. Elle répond aux demandes de la couche model et assure la persistance en retour des données traitées.

### Les modules

#### Module X

#### Module Y

#### Module Z

#### Module T

### 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 »)

racine  
 ├─ *pom.xml*  
 ├─ <moduleX>  
 │ ├─ *pom.xml*  
 │ └─ src  
 │ ├─ main  
 │ │ ├─ java  
 │ │ └─ resources  
 │ └─ test  
 │ ├─ java  
 │ └─ resources  
 ├─ <moduleY>  
 │ ├─ *pom.xml*  
 │ └─ src  
 │ ├─ main  
 │ │ ├─ java  
 │ │ └─ resources  
 │ └─ test  
 │ ├─ java  
 │ └─ ressources  
 └─ src  
 └─ lib

* ...

# Points particuliers

## Gestion des logs

…

## Fichiers de configuration

### Application web

...

#### Datasources

...

#### Fichier xxx.yyy

...

### Application Xxx

...

## Ressources

...

## Environnement de développement

## Procédure de packaging / livraison

## XXX

…

# Glossaire

|  |  |
| --- | --- |
|  |  |
|  |  |