



OC Pizza

Création et mise en place d'un système d'information

Dossier de conception technique

Version 1.0

Auteur

Plaxine Philippe
Développeur

TABLE DES MATIÈRES

1 -Versions.....	4
2 -Introduction.....	5
2.1 -Objet du document.....	5
2.2 -Références.....	5
3 -Base de données relationnelle.....	6
3.1 -Modèle Physique de Données.....	6
4 -Architecture Technique.....	7
4.1 -Composants généraux.....	7
4.1.1 - <i>Package application front office</i>	7
4.1.1.1 -Composant webserver MVC.....	7
4.1.2 - <i>Package application mobile</i>	8
4.1.2.1 -Composant application web MVC côté client.....	8
4.1.3 - <i>Package applications back office</i>	8
4.1.3.1 -Composant applications web MVC côté client.....	8
4.1.4 - <i>Package web-service REST</i>	8
4.1.4.1 -Composant web-service REST	8
4.2 -Application Web.....	8
4.2.1 - <i>Composant GUI</i>	8
4.2.2 - <i>Composant Authentification</i>	9
4.3 -Application mobile.....	9
4.3.1 - <i>Composant GUI</i>	9
4.4 -Application back office.....	9
4.4.1 - <i>Composant GUI</i>	9
4.5 -Application Web service REST.....	9
4.5.1 - <i>Composant authentification</i>	9
4.5.2 - <i>Composant Compte Personnel</i>	10
4.5.3 - <i>Composants Compte Client</i>	10
4.5.4 - <i>Composants commandes</i>	10
4.5.5 - <i>Composants produits</i>	10
4.5.6 - <i>Composants règlements</i>	10
4.5.7 - <i>Composants stocks</i>	10
5 -Architecture de Déploiement.....	11
5.1 -Serveur de Base de données.....	12
5.2 -Serveur Applications.....	12
5.3 -Serveur Logs.....	12
6 -Architecture logicielle.....	13
6.1 -Principes généraux.....	13
6.1.1 - <i>Les couches</i>	13
6.1.2 - <i>Les modules</i>	13
6.1.2.1 -Modules du web-service.....	13
6.1.2.2 -Modules des applications clients.....	13
6.1.3 - <i>Structure des sources</i>	14
7 -Points particuliers.....	15
7.1 -Gestion des logs.....	15
7.2 -Fichiers de configuration.....	15
7.2.1 - <i>Application web-service</i>	15
7.2.1.1 -Fichier confDataBase.properties.....	15
7.3 -Ressources.....	15
7.3.1 - <i>Application web</i>	15

7.3.1.1 -Fichier message_en.properties.....	15
7.3.1.2 -Fichier message_fr.properties.....	15
Environnement de développement.....	15
7.4 -Procédure de packaging / livraison.....	15
8 -Glossaire.....	16

1 - VERSIONS

Auteur	Date	Description	Version
Philippe Plaxine	03/03/19	Création du document	1.0

2 - INTRODUCTION

2.1 - Objet du document

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

Le dossier de conception technique définit et justifie les hypothèses techniques structurant le projet.

L'objectif de ce document est de fournir la structure des composants du système, leurs interactions, ainsi que les caractéristiques du matériel physique sur lequel ceux-ci sont déployés.

2.2 - Références

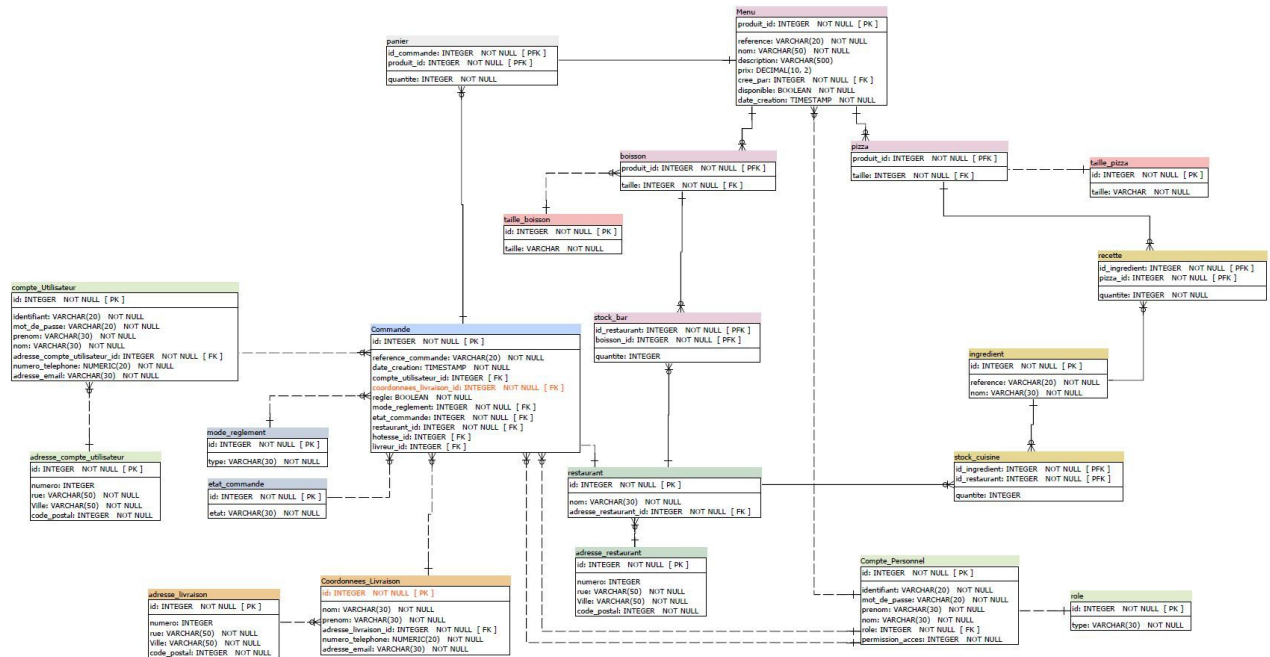
Pour de plus amples informations, se référer également aux éléments suivants:

1. **01 Projet OCPizza - Dossier de conception fonctionnelle.pdf** : Dossier de conception fonctionnelle de l'application.
2. **09 – DcomposantsOCP.jpg** : Diagramme de composant du système d'information.
3. **10 – Ddéploiement.jpg** : Diagramme de déploiement du système d'information.
4. **12 – Modèle physique de données.jpg** : Diagramme de déploiement du système d'information.

3 - BASE DE DONNÉES RELATIONNELLE

3.1 - Modèle Physique de Données

La modélisation du système de gestion de la base de données OC Pizza est représentée via le modèle physique de données (disponible en fichier annexe) ci-dessous:



Modèle Physique de Données : BDD OC Pizza

4 - ARCHITECTURE TECHNIQUE

4.1 - Composants généraux

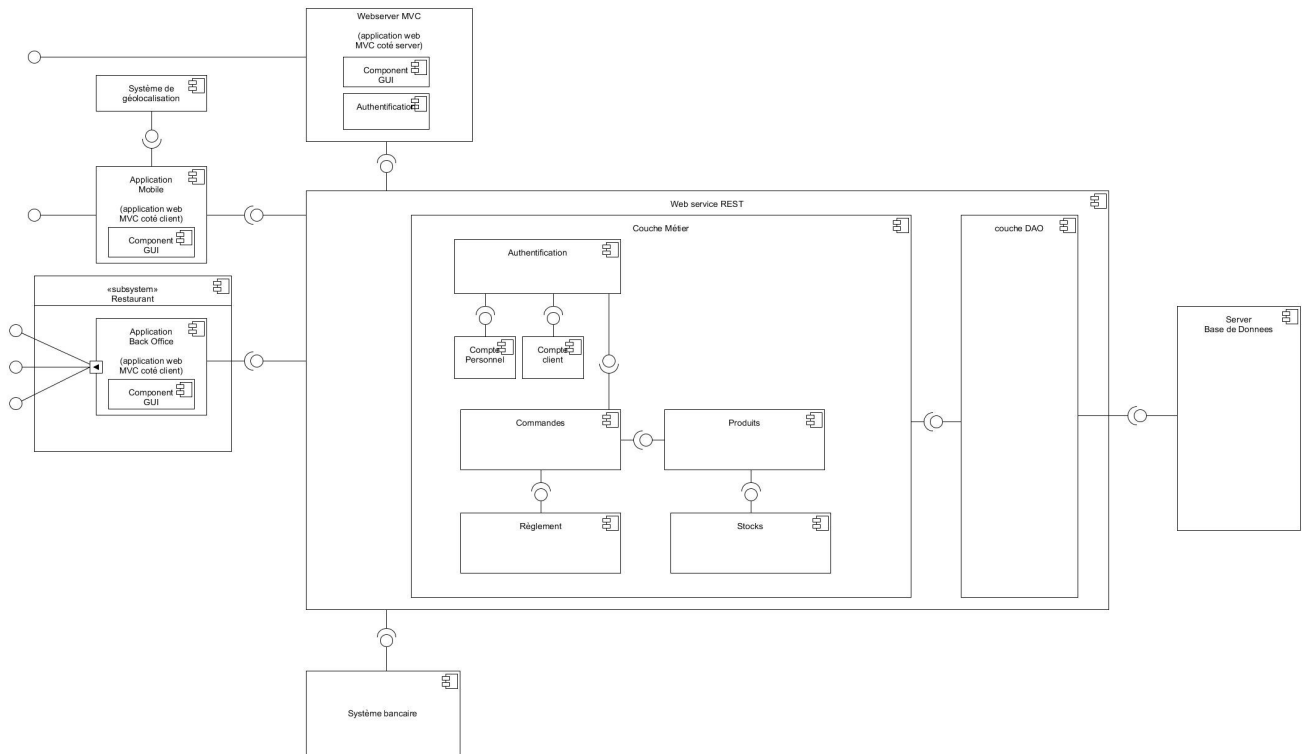


Diagramme de Composants : Système d'information OC Pizza (disponible en version.pdf)

4.1.1 - Package application front office

4.1.1.1 - Composant webserver MVC

Le web server (situé côté serveur) renvoie la réponse au format HTML, à la suite d'une requête envoyée par l'utilisateur de l'application via un moteur de recherche.

Le web server respecte une structure Model, View, Controller qui peut se résumer ainsi : La requête est récupérée par le Controller (Servlet), celui-ci fait appel au Model afin de réaliser la requête. Puis, il sollicite la View (Jsp) afin que la réponse soit envoyée au format adéquat (ex: html).

A noter, dans l'architecture application proposée, la partie Model de l'applications peut stocker des éléments propres à l'application dont elle est le modèle. Néanmoins, dans la majorité des cas, elle fera appel aux services proposés par le Web service REST.

4.1.2 - Package application mobile

4.1.2.1 - Composant application web MVC côté client

L'application mobile est une application web installé côté client. Celle-ci respecte, tout autant, une structure MVC. L'application fait appel au Web service REST afin de répondre aux requêtes utilisateur.

Celle-ci interagit, également, avec le système de géolocalisation (externe), afin de transmettre au système la position du terminal (mobile) et obtenir le tracé de l'itinéraire grâce à la récupération de l'adresse de livraison.

4.1.3 - Package applications back office

4.1.3.1 - Composant applications web MVC côté client

Similaire dans sa structure à l'application mobile, les applications du back office sont déployées sur multiples types de terminaux au sein de chaque restaurant. Sur les bornes de prise de commande des hôtes, le terminal cuisine pour les pizzaiolos, les ordinateurs de la direction pour les modifications du back office.

4.1.4 - Package web-service REST

4.1.4.1 - Composant web-service REST

Contenant la couche métier de la solution, celui-ci, traite, à travers ses services, les requêtes des différentes applications web.

Il contient une couche DAO permettant la gestion de l'accès aux données du server de BDD.

Egalement, le webservice interagit avec le système bancaire (service externe), afin de gérer les requêtes concernant les règlements.

4.2 - Application Web

La pile logicielle est la suivante:

- Application **J2EE** (JDK version 1.8)
- Serveur d'application **Tomcat 9.0.12**

4.2.1 - Composant GUI

Le composant GUI (Graphical User Interface) permet à l'utilisateur d'interagir avec l'application à travers une interface graphic (fenêtres, boutons à cliquer, champs à remplir).

4.2.2 - Composant Authentification

Le composant authentification permet à l'utilisateur de l'application de pouvoir s'authentifier et bénéficier d'une expérience de navigation personnalisée à travers l'application web.

4.3 - Application mobile

4.3.1 - Composant GUI

Le composant GUI (Graphical User Interface) permet à l'utilisateur d'interagir avec l'application à travers une interface graphique (fenêtres, boutons à cliquer, champs à remplir).

L'application fait appel au Web service REST afin de répondre aux requêtes utilisateur.

4.4 - Application back office

4.4.1 - Composant GUI

Le composant GUI (Graphical User Interface) permet à l'utilisateur d'interagir avec l'application à travers une interface graphique (fenêtres, boutons à cliquer, champs à remplir).

Les application du back office font appel au Web service REST afin de répondre aux requêtes utilisateur.

4.5 - Application Web service REST

La pile logicielle est la suivante:

- Application **J2EE** (JDK version 1.8)
- Serveur d'application **Tomcat 9.0.12**

4.5.1 - Composant authentification

Le composant authentification permet de vérifier la validité de l'identité de l'initiateur de la requête passée au web-service. Ce composant agit ainsi tel un filtre, donnant seulement suite aux requêtes dont l'identité du requêteur est valide.

Ce composant utilise, pour sa réalisation, les composants **compte personnel** et **compte client**.

4.5.2 - Composant Compte Personnel

Le composant Compte Personnel regroupe les éléments relatifs aux comptes du personnel gérés par le système d'information.

Ce composant est utilisé par le composant **authentification**.

4.5.3 - Composants Compte Client

Le composant Compte Client regroupe les éléments relatifs aux comptes Client gérés par le système d'information.

Ce composant est utilisé par le composant **authentification**.

4.5.4 - Composants commandes

Le composant commandes regroupe les éléments relatifs à la gestion des commandes.

Etant un des composants central de l'application celui-ci utilise, pour sa réalisation, les composants **authentification**, **produits** et **règlement**.

4.5.5 - Composants produits

Le composant produits regroupe les éléments relatifs aux produits vendus dans les restaurants.

Ce composant utilise, pour sa réalisation, le composant **stocks**, est utilisé par le composant **commandes**.

4.5.6 - Composants règlements

Le composant règlements regroupe les éléments relatifs aux règlements effectuées par les clients à la suite d'une commande.

Ce composant utilise, pour sa réalisation, un composant externe **Système bancaire** (gère les transactions bancaire), et est utilisé par le composant **commandes**.

4.5.7 - Composants stocks

Le composant stocks regroupe les éléments relatifs aux stock d'ingrédients disponibles dans chaque restaurant.

Ce composant est utilisé par le composant **produit**.

5 - ARCHITECTURE DE DÉPLOIEMENT

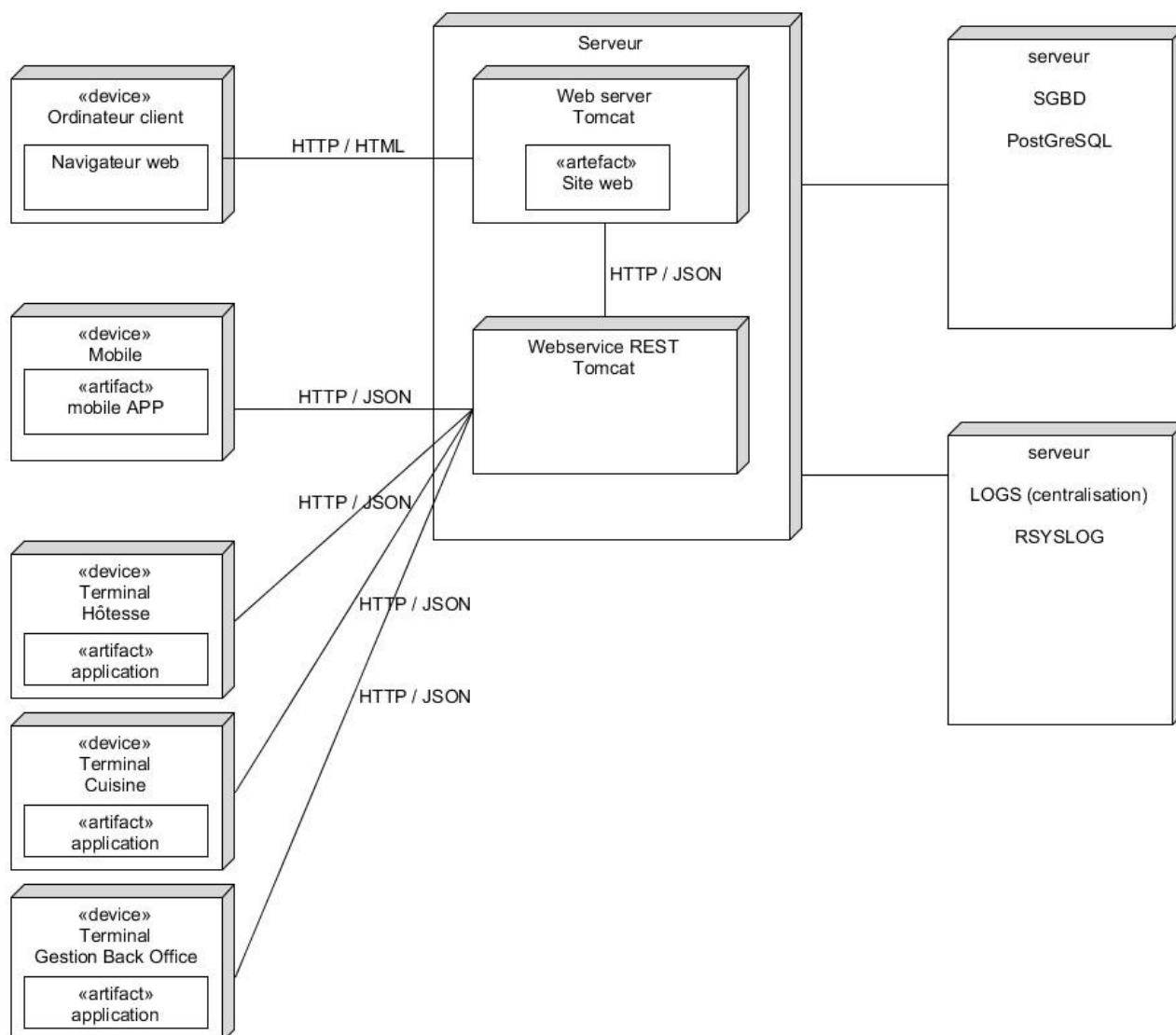


Diagramme de Composants : Système d'information OC Pizza

Les applications du Back office sont déployées sur les différents terminaux d'un restaurant (terminaux hôtesse, terminal cuisine, ordinateurs de la direction). Elle communique avec le webservice REST via un protocole HTTP, avec échange d'informations au format JSON. L'application Mobile déployée sur le mobile de chaque utilisateur, utilise le même procédé.

Les conteneurs Tomcat du webservice et de la web application sont déployés sur un même serveur et communiquent, également via un protocole HTTP / JSON. Aucun composant n'est déployé sur l'ordinateur client, le site internet étant déployé côté serveur. Un simple moteur de recherche leur est nécessaire afin de communiquer via un protocole HTTP en utilisant l'échange d'information au format html.

Le déploiement du SGBD est effectué sur un serveur qui lui est propre.

5.1 - Serveur de Base de données

Le serveur de base de données est le serveur destiné à accueillir les composants utiles à la gestion des données relatives au système d'information.

Le système d'exploitation du serveur de base de donnée est Linux Debian Jessie.

La gestion des données est réalisé grâce au SGBD PostgreSQL 11.2.

5.2 - Serveur Applications

Le serveur applications est le serveur destiné à accueillir l'application web et l'application web-service.

Le système d'exploitation du serveur application est Microsoft Windows Server.

5.3 - Serveur Logs

Le serveur logs est le serveur destiné à accueillir la centralisation des logs de toutes les applications du système d'information.

Le système d'exploitation du serveur logs est linux Debian Jessie.

6 - ARCHITECTURE LOGICIELLE

6.1 - Principes généraux

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

6.1.1 - Les couches

L'architecture applicative est la suivante:

- une couche **consumer** : responsable de la récupération des données dans la base de données.
- une couche **model** : implémentation du modèle des objets métiers.
- une couche **business** : responsable de la logique métier du composant.
- une couche **exposition** : rend l'application disponible aux utilisateurs.

6.1.2 - Les modules

6.1.2.1 - Modules du web-service

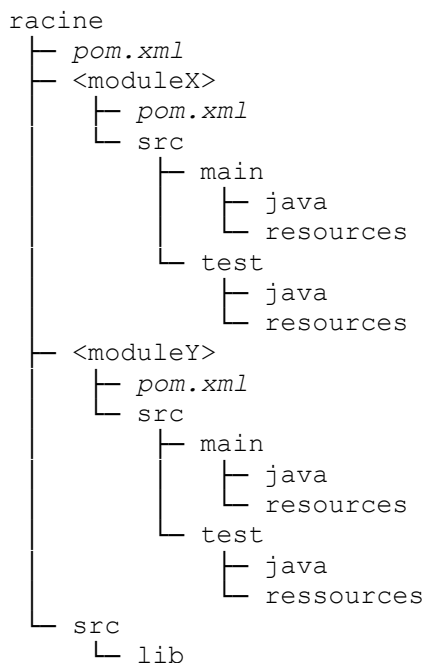
- ocpWS : module parent regroupant les autres modules de l'application.
- ocpWS-technical : dépendances utiles à tous les autres modules de l'application.
- ocpWS-model : modèles des objets métiers.
- ocpWS-consumer : interaction avec la base de données.
- ocpWS-business : responsable de la logique métier.
- ocpWS-exposure : rend les services accessibles aux applications clients.

6.1.2.2 - Modules des applications clients

- ocpApp : module parent regroupant les autres modules de l'application.
- ocpApp-technical : dépendances utiles à tous les autres modules de l'application.
- ocpApp-model : modèles des objets métiers.
- ocpApp-consumer : interaction avec les services du web-service.
- ocpApp-business : responsable de la logique métier.
- ocpApp-exposure : rend l'application accessibles aux utilisateurs.

6.1.3 - Structure des sources

La structuration des répertoires du projet suit la logique suivante :



Cette structure est implémentée dans les applications du système d'information, énumérées ci-dessous :

1. **Web-service**
2. **Application Web**
3. **Application mobile**
4. **Applications du back office**

7 - POINTS PARTICULIERS

7.1 - Gestion des logs

La gestion des logs dans les applications du système d'information est réalisée via Apache Log4j2. La centralisation des logs des applications est géré par Rsyslog sur un serveur dédié.

7.2 - Fichiers de configuration

7.2.1 - Application web-service

7.2.1.1 - Fichier *confDataBase.properties*

Fichier de configuration de la base de données.

7.3 - Ressources

7.3.1 - Application web

7.3.1.1 - Fichier *message_en.properties*

Fichier utilisé pour l'internalisation de l'application web (version anglaise).

7.3.1.2 - Fichier *message_fr.properties*

Fichier utilisé pour l'internalisation de l'application web (version française).

Environnement de développement

Les différentes applications du système d'information sont développées via Eclipse IDE for Java EE Developments Version 2018-9 (4.9).

7.4 - Procédure de packaging / livraison

Les applications du système sont packagées et livrées en fichier .war. Ceux-ci sont déployés sur les différents terminaux des utilisateurs pour les applications client side. Pour le webservice et l'application web (site internet), ceux-ci sont déployés sur le serveur dédié.

8 - GLOSSAIRE
