|  |
| --- |
| **OC Pizza**  **Projet 08**  Dossier de conception technique  Version 1.0 |
| **Auteur**  Laurent Cordier  *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 |
| Laurent Cordier | 19/12/2019 | Création du document | 1.0 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Objet du document

Le présent document constitue le dossier de conception technique de l'application OC Pizza portant sur la mise en place d’un nouveau système informatique pour l’ensemble des pizzerias du groupe.

## Références

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

1. **DCF – Projet08** : Dossier de conception fonctionnelle de l'application.
2. **DEX** **– Projet08** : Dossier d’exploitation.

# Architecture Technique

## Composant commun

### Package Sécurité

#### Composant Contrôle-accès

Description et rôle/objectif...

## Application Web e-Commerce

### La pile logicielle

La pile logicielle est la suivante :

* Angular : 8.2.14, Angular/Material : 8.2.3, Node.js : 12.13.0, Typescript : 3.5.3

### Diagramme UML de Composants

Une image contenant capture d’écran

Description générée automatiquement

### Composant panierAchat

Description et rôle/objectif...

### Composants gestionCompte

Description et rôle/objectif...

## Composants e-Management

### La pile logicielle

La pile logicielle est la suivante :

* Angular : 8.2.14, Angular/Material : 8.2.3, Node.js : 12.13.0, Typescript : 3.5.3

### Diagramme UML de Composants

Une image contenant capture d’écran

Description générée automatiquement

## Application Serveur

### La pile logicielle

La pile logicielle est la suivante :

* Application **J2EE** (JDK version 1.8) / **PHP** (version) / **Python**…
* Serveur d'application **JOnAS 5.2.4 / ...**

#### Diagramme UML de Composants

### Service Frontal-Web

Description et rôle/objectif...

### Service Persistence-Crud

Description et rôle/objectif...

### Service Paiement-Api

Description et rôle/objectif...

# Architecture de Déploiement

Diagramme UML de déploiement

Explication / commentaires si nécessaires...

## Serveur de Base de données

Description

Caractéristiques techniques (ex: Serveur Linux Debian Jessie + PostgreSQL 9.6…)

Informations importantes / points particuliers

## Serveur XXX

...

# Architecture logicielle

## Principes généraux

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

...

### Les couches

L'architecture applicative est la suivante :

* unecouche **business** : responsable de la logique métier du composant
* unecouche **model** : implémentation du modèle des objets métiers
* …
  + …
* ...

### Les modules

Ex: modules Maven dans le cas d’application multi-module...

### 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

* ...

## Application Web

Uml component dans l’optique archi logicielle (mes services vu sous forme uml avec interdependances)

Si besoin, diagramme UML de composants pour monter les différents modules et leur inter-dépendances

## Application Xxx

# 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

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