|  |
| --- |
| **{{Client}}**  **{{Projet}}**  Dossier de conception technique  Version {{Version}} |
| **Auteur**  {{Auteur}}  *{{Auteur\_Role}}* |

A Faire :

* Remplacer manuellement les valeurs entre <>
* Les valeurs entre {{}} sont à renseigner dans les propriétés du document

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 |
| Xxx | JJ/MM/AAAA | Création du document | XXX |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Objet du document

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

**Objectif du document**

L'objectif principal de ce document est de fournir une vue détaillée de la conception technique de l'application XXX. Il vise à décrire les choix architecturaux, les technologies utilisées, ainsi que les composants et les modules du système. Ce document a pour but de guider l'équipe de développement dans la mise en œuvre efficace et cohérente de l'application.

**Origine des Éléments**

Les éléments du présent dossier découlent des spécifications fonctionnelles et des exigences identifiées lors de l'analyse du système. Ces éléments ont été définis en collaboration avec les parties prenantes clés, y compris les représentants d'OC Pizza, les utilisateurs finaux et les architectes logiciels.

## Références

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

1. **DCF - Xxx** : Dossier de conception fonctionnelle de l'application
2. …

# Architecture Technique

## Composants généraux

**Package Recettes**

Le Package A englobe des fonctionnalités liées à un aspect spécifique de l'application. Il contient plusieurs composants qui coopèrent pour fournir ces fonctionnalités.

**Composant GR**

**Description et Rôle/Objectif** Le Composant GR est responsable de la gestion des recettes de pizzas. Il stocke les données des recettes, permet aux pizzaiolos de les consulter, de les mettre à jour et d'assurer la cohérence des informations de recette. Ce composant communique avec le Package A pour accéder aux informations de commande et de recette, garantissant que les pizzas sont préparées conformément aux recettes définies.

**Composant CC**

Ce composant, appartenant au Package Recettes, gère la communication avec les clients. Il traite les notifications, les mises à jour d'état des commandes et facilite les interactions entre les clients et le système. Le Composant Y assure une communication fluide et en temps réel entre les parties prenantes.

**Package Fonctionnalitées**

Le Package Fonctionnalitées englobe une autre série de fonctionnalités essentielles pour l'application.

**Composant P**

**Description et Rôle/Objectif** Le Composant P est chargé de gérer les paiements des clients. Il interagit avec des services de paiement externes pour traiter les transactions en ligne et à la livraison. Ce composant s'assure que les paiements sont effectués de manière sécurisée et que les commandes sont correctement marquées comme payées dans le système.

Cette architecture technique modulaire facilite la maintenance, l'évolutivité et le développement parallèle des différentes parties de l'application. Chaque composant a des responsabilités spécifiques et interagit avec d'autres composants pour offrir une expérience complète et cohérente aux utilisateurs d'OC Pizza.

**Package Préparation des Pizzas**

Ce package englobe les fonctionnalités liées à la préparation des pizzas. Il est divisé en plusieurs composants qui travaillent ensemble pour assurer une préparation efficace des pizzas.

**Composant Préparation**

**Description et Rôle/Objectif** Le Composant Préparation gère l'ensemble du processus de préparation des pizzas. Il récupère les commandes à préparer, attribue les recettes correspondantes et surveille l'état de préparation. Ce composant assure la coordination entre les pizzaiolos et garantit que les pizzas sont préparées dans l'ordre approprié.

**Composant Gestion des Stocks**

**Description et Rôle/Objectif** Le Composant Gestion des Stocks surveille les niveaux d'ingrédients disponibles. Il maintient une base de données des ingrédients et informe les pizzaiolos lorsque les stocks sont bas. Ce composant aide à éviter les interruptions de préparation dues à des ingrédients manquants.

## Application Web

L'application web d'OC Pizza est développée en utilisant une combinaison de technologies pour offrir une expérience utilisateur optimale. Voici la pile logicielle utilisée dans le développement :

* Application J2EE (JDK version 1.8)
* Serveur d'application JOnAS 5.2.4

Cette pile logicielle a été choisie pour sa compatibilité, sa stabilité et sa capacité à gérer les exigences de l'application OC Pizza.

**Diagramme UML de Composants**

Le diagramme UML de composants offre une vue d'ensemble de la structure de l'application en termes de composants et de leurs relations. Il permet de visualiser comment les différentes parties de l'application interagissent entre elles.

**Composants GR**

**Description et Rôle/Objectif** Le Composant GR représente la partie de l'application responsable de la gestion des recettes de pizzas. Il s'agit d'un module clé qui permet aux pizzaiolos de consulter, mettre à jour et suivre les recettes pour assurer une préparation cohérente des pizzas. Le Composant X interagit avec la base de données pour stocker et récupérer les recettes, ainsi qu'avec d'autres composants pour coordonner le processus de préparation.

**Description et Rôle/Objectif** Le Composant Préparation est au cœur du processus de préparation des pizzas au sein de l'application OC Pizza. Il a pour objectif de garantir une préparation fluide et efficace des commandes de pizzas. Ce composant récupère les commandes en attente de préparation et les associe aux recettes appropriées. Il suit également l'état de préparation de chaque commande et coordonne le flux de travail entre les pizzaiolos pour s'assurer que les pizzas sont préparées dans l'ordre adéquat.

Le Composant Préparation interagit de manière harmonieuse avec d'autres composants pour accéder aux informations des recettes et des commandes. Il joue un rôle crucial dans la réalisation d'une expérience client cohérente en veillant à ce que les pizzas soient préparées conformément aux normes définies.

**Composant Gestion des Stocks**

**Description et Rôle/Objectif** Le Composant Gestion des Stocks est dédié à la surveillance et à la gestion efficace des niveaux d'ingrédients nécessaires à la préparation des pizzas. Son rôle principal est d'assurer que les pizzaiolos disposent toujours des ingrédients nécessaires pour satisfaire les commandes des clients. Ce composant maintient une base de données des ingrédients et surveille en temps réel les niveaux d'approvisionnement.

Lorsque les stocks d'ingrédients sont bas, le Composant Gestion des Stocks alerte les pizzaiolos pour qu'ils puissent prendre des mesures pour réapprovisionner les ingrédients. Ainsi, il évite les interruptions dans le processus de préparation et garantit que les pizzas peuvent être préparées sans délai.

En interagissant avec d'autres composants tels que le Composant Préparation, le Composant Gestion des Stocks assure une cohésion entre les processus de gestion des stocks et de préparation des pizzas, contribuant ainsi à une expérience de service de haute qualité pour les clients d'OC Pizza.

# Architecture de Déploiement

## Serveur de Base de données

Description

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

Informations importantes / points particuliers

# Architecture logicielle

## Principes généraux

* Les principes généraux définissent les normes et les outils utilisés pour gérer et structurer le projet d'application OC Pizza.
* **Gestion de Projet et Outils**
* Les sources et versions du projet sont gérées par Git, facilitant le suivi des modifications et la collaboration entre les développeurs.
* Les dépendances et le packaging sont gérés par Apache Maven / Grunt / (autre outil de build), simplifiant la gestion des bibliothèques et des packages.
* **Couches de l'Architecture**
* L'architecture applicative est organisée en couches pour une meilleure séparation des responsabilités et une facilitation du développement.
* Une couche business : Responsable de la logique métier du composant, où les traitements métier spécifiques sont mis en œuvre.
* Une couche model : Implémentation du modèle des objets métiers, où les structures de données et les objets sont définis.
* **Modules**
* Dans le cas d'une application multi-module, les modules Maven sont utilisés pour diviser l'application en unités fonctionnelles distinctes.
* **Structure des Sources**
* La structuration des répertoires suit la philosophie Maven, où la convention prime sur la configuration.

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

* La racine du projet contient le fichier de configuration principal (pom.xml pour Maven) ainsi que les différents modules du projet.
* Chaque module a son propre fichier de configuration (pom.xml) et est structuré en suivant la convention Maven : répertoires "main" pour le code source et les ressources, et "test" pour les tests unitaires.
* Le répertoire "lib" à la racine contient les bibliothèques ou les dépendances qui ne sont pas disponibles via Maven.

# Points particuliers

## Gestion des log

La gestion des logs est essentielle pour le suivi des activités de l'application et le dépannage. Les logs fournissent une vue d'ensemble des opérations et des erreurs, ce qui facilite la maintenance et l'amélioration continue.

## Fichiers de configuration

### Application web

Les configurations liées à l'application web sont gérées ici, définissant les paramètres tels que les URL, les sessions, la sécurité, etc.

#### Datasources

Les configurations des sources de données, comme les bases de données, sont gérées dans cette section. Cela inclut les détails de connexion et les paramètres de base de données.

#### Fichier configure

Ce fichier contient des configurations spécifiques à une fonctionnalité particulière de l'application. Il peut regrouper des paramètres personnalisés ou avancés.

## Ressources

...

## Environnement de développement

* Système d'exploitation : Windows, macOS, Linux (versions prises en charge).
* Outils de développement : Eclipse IDE (version XYZ) ou IntelliJ IDEA (version XYZ).
* Langage : Java (JDK version 1.8).
* Prérequis matériels : Au moins 4 Go de RAM, processeur dual-core.
* Gestionnaire de version : Git (version XYZ).

## Procédure de packaging / livraison

* Clonez le référentiel Git du projet sur votre environnement de développement local : **git clone** [**https://github.com/oc-pizza/projet.git**](https://github.com/oc-pizza/projet.git)
* À l'intérieur de chaque module, exécutez la commande Maven suivante pour construire l'application : **mvn clean install**
* Une fois la construction terminée, les artefacts sont générés dans le dossier "target" de chaque module.
* Rassemblez les artefacts et les dépendances nécessaires dans un package distribuable.
* Pour déployer l'application sur un serveur de production, suivez les instructions spécifiques au serveur utilisé (par exemple, déploiement sur un serveur Tomcat).

# Glossaire

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

* **Dossier de conception technique** : Le présent document constitue le dossier de conception technique de l'application OC Pizza, détaillant les choix architecturaux, les composants, les modules et les fonctionnalités du système.
* **Version {{Version}}** : Correspond à la version spécifique du document, à remplir conformément à la numérotation du document.
* **Table des matières** : Cette section présente un aperçu structuré des différentes parties du document, permettant une navigation rapide vers les informations spécifiques.
* **Versions** : Répertorie les différentes versions du document avec leurs descriptions et les dates associées.
* **Architecture Technique** : Cette section décrit l'architecture globale de l'application OC Pizza, en mettant en évidence les composants et les packages essentiels.
* **Composants généraux** : Indique les packages et les composants qui sont fondamentaux pour le bon fonctionnement de l'application.
* **Application Web** : Décrit les technologies et les composants utilisés pour développer l'interface web de l'application OC Pizza.
* **Diagramme UML de Composants** : Fournit une représentation visuelle de l'architecture des composants de l'application.
* **Composants GR** : Désigne le composant responsable de la gestion des recettes de pizzas au sein de l'application.
* **Composants Préparation** : Décrit le composant central responsable de la préparation des pizzas dans l'application.
* **Composant Gestion des Stocks** : Indique le composant chargé de surveiller et de gérer les niveaux d'ingrédients disponibles pour la préparation des pizzas.
* **Architecture de Déploiement** : Cette section expose les éléments liés à l'infrastructure et au déploiement de l'application, y compris les serveurs et les bases de données.
* **Architecture logicielle** : Décrit les principes généraux, les outils et la structure utilisés pour organiser le développement du projet OC Pizza.
* **Principes généraux** : Décrit les normes de gestion de projet, les outils de versionnage, et la structure des couches et des modules.
* **Les couches** : Explique les différentes couches architecturales utilisées pour organiser le code et la logique de l'application.
* **Modules** : Fait référence aux unités fonctionnelles distinctes en cas d'application multi-module.
* **Structure des sources** : Décrit l'organisation des répertoires du projet en suivant les conventions Maven.
* **Gestion des logs** : Détaille l'importance de la gestion des logs pour la maintenance et le dépannage de l'application.
* **Fichiers de configuration** : Présente les fichiers de configuration utilisés pour paramétrer l'application, notamment pour les applications web et les bases de données.
* **Datasources** : Explique comment sont gérées les configurations des sources de données, en particulier les bases de données.
* **Fichier configure** : Précise la fonction du fichier de configuration spécifique à une fonctionnalité particulière de l'application.
* **Ressources** : Aborde les ressources utilisées dans l'application, sans entrer dans les détails spécifiques.
* **Environnement de développement** : Liste les éléments requis pour le développement de l'application, y compris le système d'exploitation, les outils, les langages et les prérequis matériels.
* **Procédure de packaging / livraison** : Décrit les étapes pour empaqueter et déployer l'application sur un serveur de production.