

OC Pizza

Documentez votre système de gestion de pizzeria

Dossier de conception technique

Version 1.0

Auteur

BRICHET Benoît

Développeur JAVA chez OpenClassrooms

TABLE DES MATIERES

1 - Versions	4
2 - Introduction	5
2.1 - Objet du document	5
2.2 - Références	5
3 - Architecture Technique	6
3.1 - Composants généraux	6
3.1.1 - <i>Package Site web</i>	6
3.1.1.1 - Composant 1 : Moteur de recherche	6
3.1.1.2 - Composant 2 : Panier	6
3.1.2 - <i>Package Stocks</i>	6
3.1.2.1 - Composant 3 : Ingrédient	6
3.1.2.2 - Composant 4 : Pizza	6
3.1.3 - <i>Package Commandes</i>	6
3.1.3.1 - Composant 5 : Commandes	6
3.1.4 - <i>Package Comptes</i>	7
3.1.4.1 - Composant 6 : Employé	7
3.1.4.2 - Composant 2 : Client	7
3.1.4.3 - Composant 3 : Authentification employé	7
3.1.4.4 - Composant 4 : Authentification client	7
3.2 - Application Web	8
3.2.1 - <i>Composants a : Application web</i>	8
3.2.2 - <i>Composants b : Base de données</i>	8
3.2.3 - <i>Composants c : Gestion de paiement en ligne</i>	9
4 - Architecture de Déploiement	10
4.1 - Serveur de Base de données	10
5 - Architecture logicielle	11
5.1 - Principes généraux	11
5.1.1 - <i>Les couches</i>	11
5.1.2 - <i>Les modules</i>	11
5.1.3 - <i>Structure des sources</i>	11
6 - Points particuliers	13
6.1 - Gestion des logs	13
6.2 - Fichiers de configuration	13
6.2.1 - <i>Application web</i>	13
6.2.1.1 - Fichier pom.xml	13
6.2.1.2 - Fichier de logs	14
6.2.1.3 - Fichier application.properties	14
6.3 - Environnement de développement	15
6.4 - Procédure de packaging / livraison	15



1 - VERSIONS

Auteur	Date	Description	Version
BRICHET Benoît	22/10/2020	Création du document	1.0

2 - INTRODUCTION

2.1 - Objet du document

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

Définit et explique les interfaces de programmation, les structures de données ainsi que les algorithmes.

Les éléments du présent dossier découlent :

- Du recueil des besoins clients.
- Du dossier fonctionnel

2.2 - Références

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

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

3 - ARCHITECTURE TECHNIQUE

3.1 - Composants généraux

3.1.1 - Package Site web

3.1.1.1 - Composant 1 : Moteur de recherche

Rôle : Moteur de recherche des pizzas

Description : Permet la recherche des pizzas, par catégorie, par nom, ...

3.1.1.2 - Composant 2 : Panier

Rôle : Regroupe les produits en attente de commande du client

Description : Permet de regrouper les produits du client en attente de commande et ainsi voir toutes les pizzas sélectionner avec leurs tailles et le prix des pizzas ainsi que de la commande

3.1.2 - Package Stocks

3.1.2.1 - Composant 3 : Ingrédient

Rôle : Répertoire les ingrédients par nom et sa quantité

Description : Permet de connaître le nom des ingrédients et la quantité et ainsi savoir quelles pizzas peut être fabriqué

3.1.2.2 - Composant 4 : Pizza

Rôle : Répertoire les ingrédients et la quantité par pizza

Description : Permet de connaître les ingrédients de la pizza et sa quantité demandée et ainsi voire comment la préparer et si elle peut être fabriqué selon la quantité en stocks restant

3.1.3 - Package Commandes

3.1.3.1 - Composant 5 : Commandes

Rôle : Répertoire les commandes passées, en attente de préparation jusqu'à la livraison

Description : Permet de regrouper toutes les commandes, du statut en attente de préparation à la livraison

3.1.4 - Package Comptes

3.1.4.1 - Composant 6 : Employé

Rôle : Répertoire les comptes des employés

Description : Permet de connaître ou de créer un compte employé et ainsi changer leurs paramètres, comme leurs statuts d'employé (responsable, livreur, pizzaiolo), ainsi que l'insérer en base

3.1.4.2 - Composant 2 : Client

Rôle : Répertoire les comptes des clients

Description : Permet de créer ou de connaître un compte client et l'insérer en base

3.1.4.3 - Composant 3 : Authentification employé

Rôle : Authentification des employés

Description : Permet d'authentifier un employé pour qu'il puisse avoir accès aux fonctionnalités selon son statut (responsable, pizzaiolo ou livreur)

3.1.4.4 - Composant 4 : Authentification client

Rôle : Authentification des clients

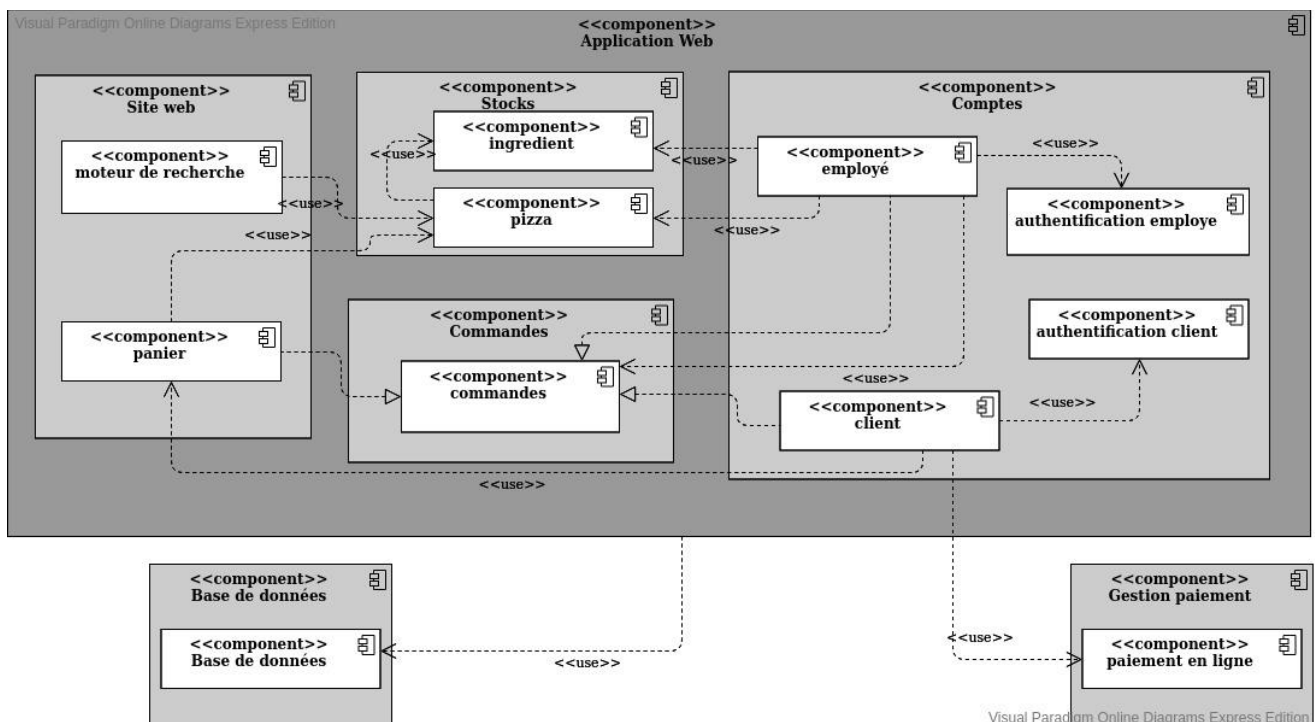
Description : Permet d'authentifier un client pour qu'il puisse passer commande, modifier, annuler ou suivre sa commande, ...

3.2 - Application Web

La pile logicielle est la suivante :

- Application **J2EE** (JDK version 11)
- Serveur d'application **Tomcat**
- Base de données **postgresql**

Diagramme UML de Composants



3.2.1 - Composants a : Application web

Rôle : Regroupe l'application web

Description : Permet l'utilisation du site web ainsi que du tableau de bord pour les employés. Logique métier de l'application.

3.2.2 - Composants b : Base de données

Rôle : Répertorie toutes les données de l'application

Description : Permet de regrouper toutes les données de l'application et qui peut ainsi être



utilisés selon la fonctionnalité

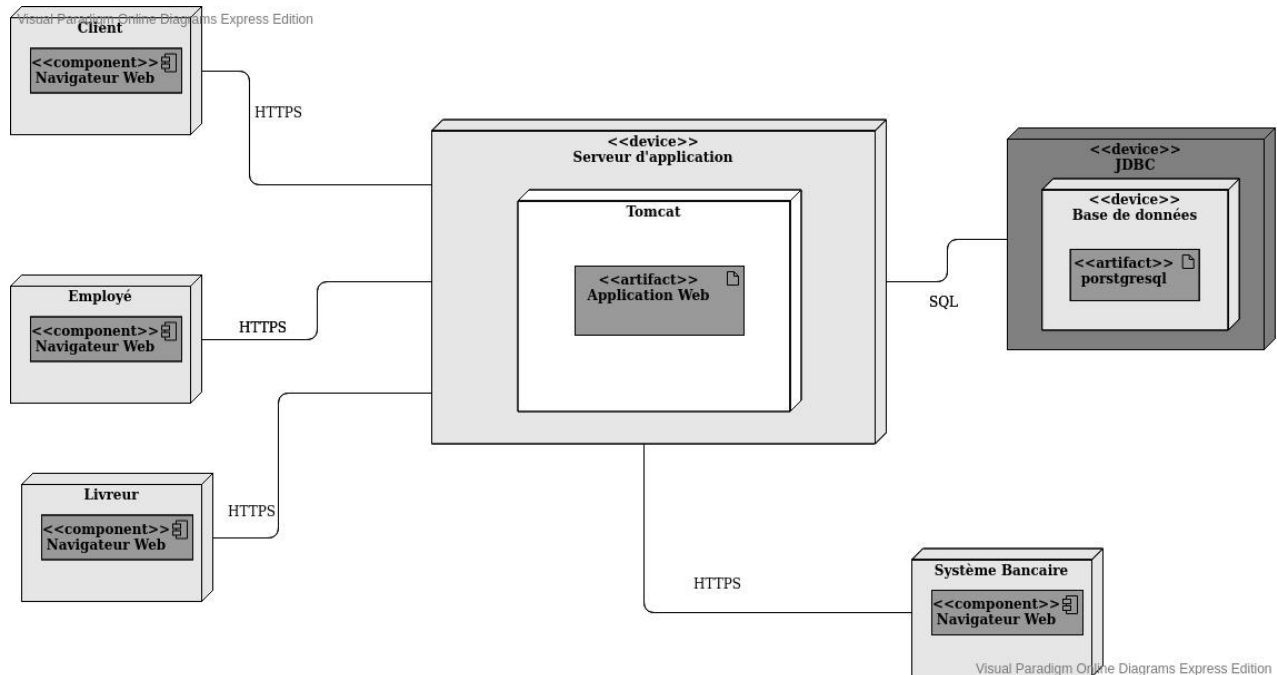
3.2.3 - Composants c : Gestion de paiement en ligne

Rôle : Protocole de paiement en ligne

Description : Protocole qui permet le paiement en ligne si le client souhaite payer via le site web

4 - ARCHITECTURE DE DEPLOIEMENT

Diagramme UML de déploiement



Explication / commentaires si nécessaires...

4.1 - Serveur de Base de données

Serveur de base de données : MySQL

MySQL est un serveur de base de données relationnel Open Source.

Caractéristiques techniques (MySQL 8.0.20)

Informations importantes / points particuliers

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 **repository** : Interface qui consiste à rendre la création de la couche d'accès aux données. (CRUD)
- une couche **controller** : Gère les requêtes client. Point d'entrée de l'application

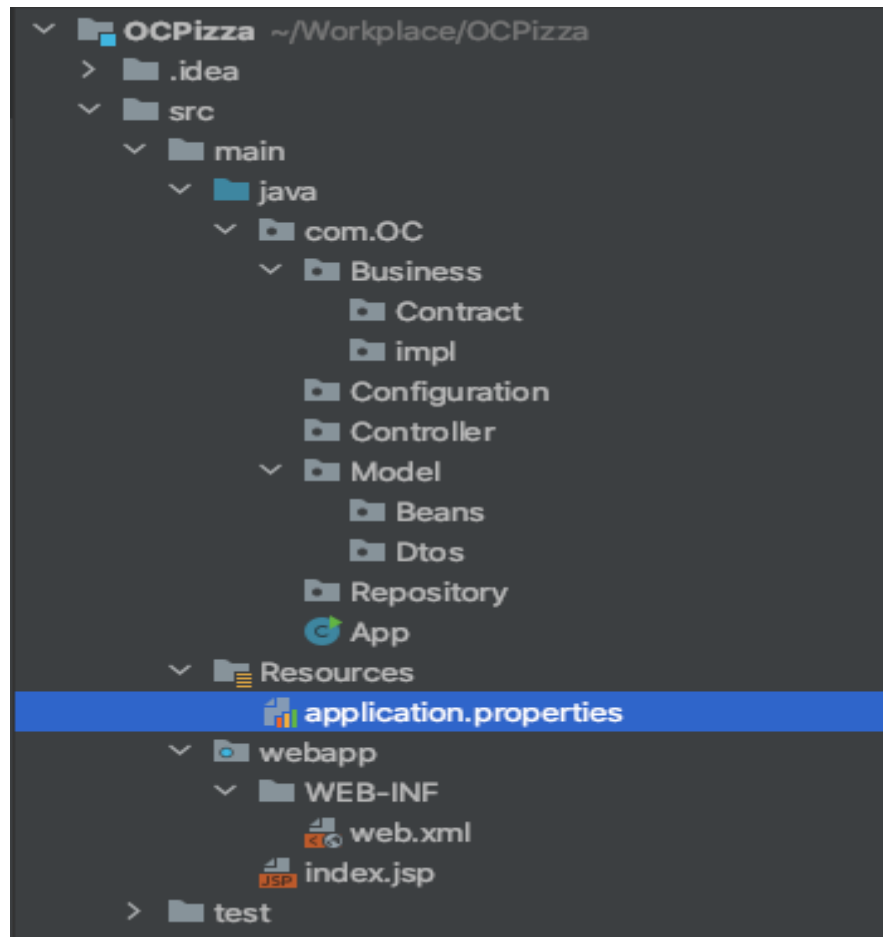
5.1.2 - Les modules

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

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 »)



6 - POINTS PARTICULIERS

6.1 - Gestion des logs

La gestion des logs se fera avec Log4j.

6.2 - Fichiers de configuration

6.2.1 - Application web

6.2.1.1 - Fichier pom.xml

```
<properties>
  <java.version>11</java.version>
  <spring-cloud.version>Hoxton.SR8</spring-cloud.version>
</properties>

<dependencies>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-test</artifactId>
    <scope>test</scope>
    <exclusions>
      <exclusion>
        <groupId>org.junit.vintage</groupId>
        <artifactId>junit-vintage-engine</artifactId>
      </exclusion>
    </exclusions>
  </dependency>
</dependencies>
```

```
<dependency>
    <groupId>jstl</groupId>
    <artifactId>jstl</artifactId>
    <version>1.2</version>
</dependency>
</dependencies>
```

6.2.1.2 - Fichier de logs

```
<?xml version="1.0" encoding="UTF-8"?>
<Configuration status="TRACE">
    <Appenders>
        <File name="log" fileName="all.log" immediateFlush="true"
append="false">
            <PatternLayout pattern="%d{yyy-MM-dd HH:mm:ss.SSS} %-5level
[%C] (%M) - %msg%n"/>
        </File>
    </Appenders>
    <Loggers>
        <Root level="TRACE">
            <AppenderRef ref="log"/>
        </Root>
    </Loggers>
</Configuration>
```

6.2.1.3 - Fichier application.properties

Propriétés Database

```
spring.datasource.driver.classe.name=com.mysql.cj.jdbc.Driver
spring.datasource.url=jdbc:mysql://localhost:3306/(nomDatabase)?useSSL=false&serverTim
ezone=UTC
spring.datasource.username=(username_DATABASE)
spring.datasource.password=(password_DATABASE)
spring.jpa.hibernate.ddl-auto=update
hibernate.dialect=org.hibernate.dialect.MySQL5Dialect
```

6.3 - Environnement de développement

L'environnement de développement utilisé sera IntelliJ.

6.4 - Procédure de packaging / livraison

L'application sera packagée dans un fichier .war. Pour l'ensemble des scripts utilisés pour la création de la base de données, ils seront livrés dans un .zip.