



OC PIZZA

Documentez votre système de gestion de pizzeria

Dossier de conception technique

Version 1.0

Auteur

F.CHA

Développeur d'Application Web



TABLE DES MATIERES

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 - <i>Package Authentification</i>	5
3.1.2 - <i>Package Gestion des commandes</i>	5
3.1.3 - <i>Package Gestion des stocks</i>	5
3.1.4 - <i>Package Gestion administrative / dashboard / paramétrage</i>	6
3.2 - Diagramme de Composants.....	7
4 - Architecture de Déploiement	8
4.1 - Serveur de Base de données.....	9
4.2 - Serveur Locaux.....	9
4.3 - Principes généraux	9
4.4 - Les Services.....	10
4.4.1 - <i>Les couches</i>	10
4.4.2 - <i>Les ressources</i>	10
5 - Outils Technique.....	11
6 - Points particuliers.....	13
6.1 - Ressources	14
6.2 - Environnement de développement.....	14
6.2.1 - <i>Intellij IDEA</i>	14
6.2.2 - <i>DBeaver</i>	14
6.2.3 - <i>PowerArchitect</i>	14
6.3 - Procédure de packaging / livraison	14



1 - VERSIONS

Auteur	Date	Description	Version
F.CHA	12/11/2022	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, système informatique de gestion et de vente en ligne.

Objectif du document : définir la démarche de conception de l'application OC PIZZA

2.2 - Références

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

1. **Projet OC PIZZA – Dossier de conception fonctionnelle** : Dossier de conception fonctionnelle de l'application
2. **Projet OC PIZZA – Dossier d_exploitation** : Dossier d'exploitation de l'application
3. **Projet OC PIZZA – PV** : Procès-Verbal



3 - ARCHITECTURE TECHNIQUE

3.1 - Composants généraux

Pour la mise en place de l'architecture technique, nous concevons d'abord le diagramme de package que l'on retrouve dans le dossier de conception fonctionnelle. Nous retrouvons à l'intérieur les éléments mettant en place l'architecture du projet et de l'application.

3.1.1 - Package Authentication

Il permet aux utilisateurs de s'authentifier pour pouvoir se connecter. Pour se faire, il compare les informations rentrées par l'utilisateur avec celles de la base de données. Si elles coordonnent, l'utilisateur est autorisé et donc connecté.

3.1.2 - Package Gestion des commandes

Il représente le système dans son ensemble de la gestion commandes pour les différents points de vente. Il est le cœur de l'application.

Cela va de la consultation de la carte et la prise de commande, jusqu'à la livraison finale des produits.

Il est en liaison avec tous les acteurs de l'entreprise et le client, mais également avec le système bancaire pour le règlement des commandes.

3.1.3 - Package Gestion des stocks

La gestion des stocks sur les points de vente est regroupée dans ce package.

Il gère l'ensemble des stocks sur les points de ventes, cela concerne les produits servant à élaborer les pizzas et autres, de la réception de la commande du fournisseur jusqu'à l'entrée physique et informatique du stock.

Mais également à toutes les sorties de stocks comprenant les boissons et les autres produits vendus.

Chaque point de vente gère son stock, et fait l'inventaire de celui-ci afin de pouvoir réconcilier le stock physique et informatique.



3.1.4 - Package Gestion administrative / dashboard / paramétrage

Il permet la gestion des employés uniquement pour les gérants. Toutes les informations nécessaires concernant les employés s'y retrouvent.

Il permet également le paramétrage du site en ligne, changement de produit mise en forme etc.

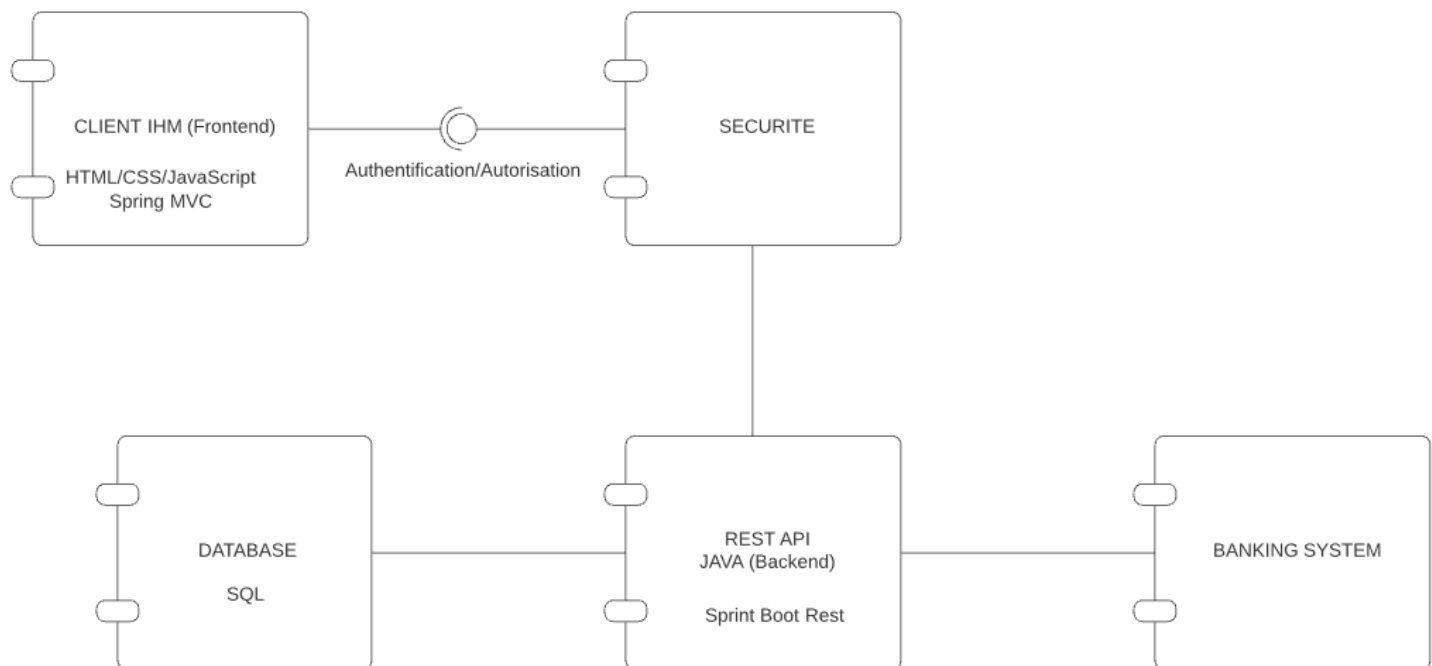
Le dashboard permet aux responsables points de ventes de voir leur stock, le chiffre d'affaire etc. Les gérants peuvent voir la même chose sur tous les points de ventes.



3.2 - Diagramme de Composants

Le diagramme de composants permet d'obtenir une vue d'ensemble du système logiciel.

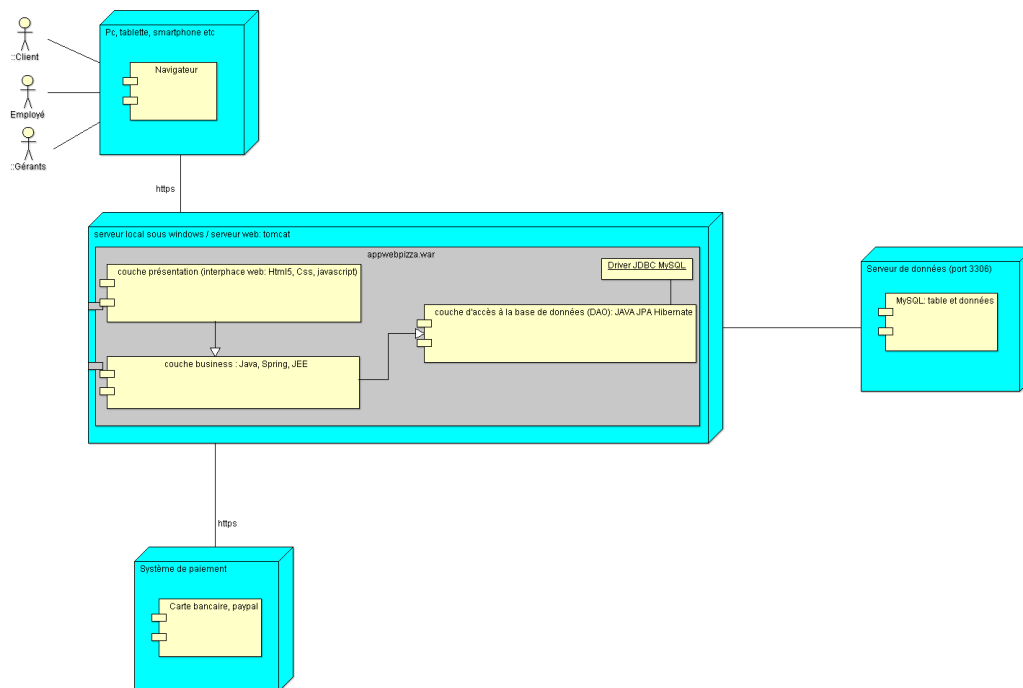
Diagramme UML de Composants





4 - ARCHITECTURE DE DÉPLOIEMENT

Diagramme UML de déploiement



L'IHM échange avec les serveurs Locaux (trois) par l'intermédiaire de requête disposant du protocole Http.

Il y a trois serveurs distincts (La base de donnée, l'API et le BATCH (partie BACKEND), la partie WEB (partie « FRONTEND »)).

Nous retrouvons pour chaque service un environnement permettant de faire fonctionner l'application aussi bien du côté Backend que Frontend.

Nous retrouvons également notre SGBDR Mysql contenant notre base de données 'projet8_OCPizza' se trouvant dans un autre serveur Local.

Celle-ci est en relation avec les services (API et BATCH) par l'intermédiaire de l'API JDBC créant une connexion à partir d'un URL. Elle est en relation avec l'ORM Hibernate qui lui génère les requêtes Mysql permettant de pouvoir interroger la base de données.



4.1 - Serveur de Base de données

La base de données est hébergée sur un serveur local sous Windows 10.

La mise en place de celle-ci est effectuée comme décrite dans le document d'exploitation.

Une fois installée, nous configurons le fichier application.properties, en modifiant l'URL.

CPU: Dual Core 3.7 GHz

RAM: 8Go en DDR4

DD: chaque serveur possède 250Go d'espace.

4.2 - Serveur Locaux

Les services (l'application web, l'api, le batch) sont déployés par le biais des serveurs Locaux, il y en a 3 en plus de celui de la base de données. Ils ont tous les 3 la même configuration comme suit :

CPU: Dual Core 3.7 GHz

RAM: chaque serveur possède 4Go en DDR4

DD: chaque serveur possède 25Go d'espace.

4.3 - Principes généraux

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

Vous retrouverez comment avoir accès à Git et comment restaurer le projet sur le document d'exploitation.

Apache Maven sert à packager le projet et notamment chaque service en générant un fichier JAR.

Nous retrouvons à l'intérieure de chaque fichier pom le type du packaging souhaité pour qu'ensuite Maven puisse packager le projet.

Maven sert également à gérer les dépendances du projet. Il récupère les dépendances mise en place dans le fichier pom, ainsi que les dépendances requises à celles dont on a besoin grâce à la gestion de la transitivité des dépendances.



4.4 - Les Services

4.4.1 - Les couches

L'architecture applicative est la suivante :

- Une couche **controller** : contient les méthodes permettant de récupérer, d'envoyer, de sauvegarder les données et d'interagir avec les objets pour y avoir accès. Elle fait le lien entre la couche dao et la couche model.
- Une couche **model** : implémentation du modèle des objets métiers.
- Une couche **dao** : interface permettant d'interroger la base de données.
- Une couche **service** : Elle permet d'ajouter et de faciliter la manipulation des objets suivant le résultat attendu, lorsque cela est compliqué avec la couche controller.

4.4.2 - Les ressources

Nous retrouvons nos fichiers .properties détaillés dans le dossier d'exploitation.

Pour l'application Web, nous retrouvons également deux sous-dossiers :

- Static : contenant les fichiers CSS, Javascript et également les photos du site.
- Template : contenant toutes les pages HTML de l'interface du projet.

Pour le service Batch, nous retrouvons les mêmes sous-dossiers :

- Static : contenant les photos affichées dans les emails
- Template : où l'on trouve les différentes pages HTML contenant les emails.



5 - OUTILS TECHNIQUE

Les outils techniques utilisés dans le projet sont tous répertoriés dans ce tableau :

Technologie	Utilisation Projet
Java	L'utilisation du langage java permet d'implémenter les uses cases.
Maven	Maven est un outil de gestion d'automatisation et de production logiciels Java. Il est en général utilisé pour la gestion des dépendances et la compilation du projet.
Spring	Spring est Framework Java pour construire et définir l'infrastructure d'une application Java, dont il facilite le développement et les tests.
Spring Boot	Spring Boot permet de faciliter le développement de l'application, en gérant automatiquement les dépendances se trouvant dans les fichiers pom, afin de gérer uniquement l'ensemble des dépendances utiles, par l'ajout des starters.
Spring Boot Starter Web	Spring Boot Starter Web permet d'ajouter les dépendances permettant le fonctionnement du microservice (Tomcat, Hibernate...)
Feign	Client http se trouvant dans le starter Spring Cloud. Il permet de scanner le microservice puis de créer les requêtes Http, afin de communiquer avec le microservice souhaité.
Lombok	Librairie Java qui permet par l'intermédiaire d'annotation, de générer les getters, setters, constructeur, toString, equals, Hashcode des objets Java.
Thymeleaf	Moteur de template qui simplifie la syntaxe, par l'intermédiaire de tags HTML via des attributs, qui permettent d'accéder aux variables et objets à l'intérieur des controllers, pour ensuite les afficher dans les pages web.



FreeMarker	Librairie Java permettant à partir d'un fichier HTML de pouvoir générer un mail.
Mysql	Mysql est un système de gestion de base de données relationnelle et objet (SGBDRO) permettant d'interagir avec la base de données.
Dbeaver	Interface permettant de rédiger et appliquer les requêtes Mysql
Tomcat	Tomcat est un serveur d'applications Java permettant de gérer les requêtes entre le client et le serveur, de façon dynamique.
HTML / CSS	Le langage HTML et CSS permettent de créer et de représenter le contenu d'une page web et son aspect.
Bootstrap	Librairie CSS et javascript, permettant de faciliter la conception de l'interface du site et sa responsivité.
Javascript	Le langage javascript permet de créer du contenu dynamique sur les pages web.

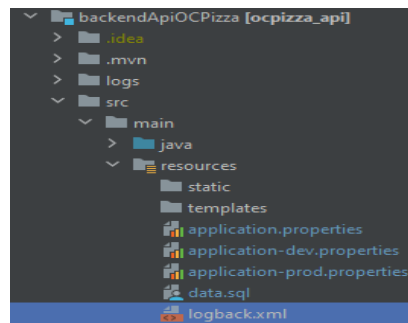
La première partie du tableau sont les outils utilisés pour le coté Backend de l'application.
La deuxième partie regroupe les outils utilisés pour le coté Frontend de l'application.



6 - POINTS PARTICULIERS

La gestion des logs est gérée par log4j2.

La gestion des logs et le paramétrage se trouvent dans le fichier « logback.xml » dans le dossier « resources » de l'application :



Le détail du fichier :

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>

  <property name="HOME_LOG" value="logs/batch.log"/>

  <appender name="STDOUT" class="ch.qos.logback.core.ConsoleAppender">
    <encoder>
      <pattern>%-5relative %-5level %logger{35} - %msg%n</pattern>
    </encoder>
  </appender>

  <appender name="FILE-ROLLING" class="ch.qos.logback.core.rolling.RollingFileAppender">
    <file>${HOME_LOG}</file>

    <rollingPolicy class="ch.qos.logback.core.rolling.SizeAndTimeBasedRollingPolicy">
      <fileNamePattern>logs/archived/batch.%d{yyyy-MM-dd}.%i.log</fileNamePattern>
      <maxFileSize>10MB</maxFileSize>
      <totalSizeCap>20GB</totalSizeCap>
      <maxHistory>60</maxHistory>
    </rollingPolicy>

    <encoder>
      <pattern>%d %p %c{1.} [%t] %m%n</pattern>
    </encoder>
  </appender>

  <logger name="com.ocpizza" level="debug" additivity="false">
    <appender-ref ref="FILE-ROLLING"/>
    <appender-ref ref="STDOUT" />
  </logger>

  <root level="error">
    <appender-ref ref="FILE-ROLLING"/>
    <appender-ref ref="STDOUT" />
  </root>

</configuration>
```

Dans le fichier, la sortie des logs se fait dans un fichier appelé « api.log ».

Cela permet d'avoir une trace des logs, pour vérifier le bon fonctionnement de l'application et de gérer les bugs en cas de problème.



6.1 - Ressources

Les ressources sont détaillées dans le dossier d'exploitation.

6.2 - Environnement de développement

6.2.1 - IntelliJ IDEA

Le projet est réalisé sur l'environnement de développement « IntelliJ IDEA 2022.2.3 »

Les informations de l'IDE se trouvent [ici](#).

Le téléchargement de l'IDE se trouve dans ce [lien](#).

6.2.2 - DBeaver

DBeaver est une interface graphique permettant d'administrer une base de données et exécuter des requêtes Mysql.

C'est un client web qui utilise le navigateur pour afficher son interface.

La procédure d'installation est dans le dossier d'exploitation.

Toutes les manipulations d'installation, d'exécution des requêtes, de sauvegarde, de restauration sont réalisées sur DBeaver, et pour certaines fonctions la possibilité également, de le faire sur l'IDE.

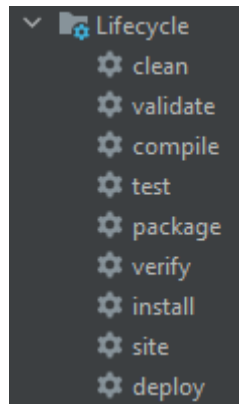
6.2.3 - PowerArchitect

Logiciel qui permet de modéliser une base de données puis de générer automatiquement le schéma dans le système de gestion de base de données de notre choix.

La procédure d'installation du logiciel se trouve dans le dossier d'exploitation.

6.3 - Procédure de packaging / livraison

Maven package le projet par l'intermédiaire de l'interface de l'IDE :



La livraison du projet comprend :

- Le code de l'application.
- Les fichiers JAR de tous les services (l'application web, l'api, le batch).
- Les sauvegardes data et structure de la base de données.
- Le dossier de conception fonctionnelle.
- Le dossier d'exploitation.
- Le dossier de conception technique.
- Le PV de livraison.