



OC PIZZA



Openclassrooms

OC PIZZA

Application PIZZA

Dossier de conception technique

Version 1.0.0

Auteur

Lehchibi Gaël
Analyste Programmeur

Openclassrooms

www.openclassrooms.com

7 cité Paradis – 01 80 88 80 30 – help@openclassrooms.com

S.A.S. au capital de 169 914 € enregistrée au RCS de Paris – SIRET 493 861 363 00064 –
Code APE : 8559A



OC PIZZA



Openclassrooms

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 - Application Web	5
3.2 - Organisation de l'application.....	5
3.2.1 - Base de données.....	6
3.2.2 - Modèle physique de donnée	7
4 - Architecture de Déploiement	8
4.1 - Diagramme de déploiement.....	8
4.2 - Serveur de déploiement	9
5 - Architecture logicielle	10
5.1 - Principes généraux.....	10
5.1.1 - Les modules	10
5.1.2 - Structure des sources.....	10
6 - Points particuliers	12
6.1 - Gestion des logs	12
6.2 - Ressources.....	12
6.3 - Environnement de développement.....	12
6.4 - Procédure de packaging / livraison.....	12



OC PIZZA



Openclassrooms

1 - VERSIONS

Auteur	Date	Description	Version
Lehchibi Gaël	09/03/2020	Création du document	1.0.0

Openclassrooms

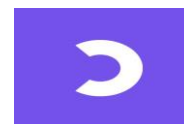
www.openclassrooms.com

7 cité Paradis – 01 80 88 80 30 – help@openclassrooms.com

S.A.S. au capital de 169 914 € enregistrée au RCS de Paris – SIRET 493 861 363 00064 –
Code APE : 8559A



OC PIZZA



Openclassrooms

2 - INTRODUCTION

2.1 - Objet du document

Le présent document constitue le dossier de conception fonctionnelle de l'application Pizza

L'objectif du document est de présenter les acteurs, intervenants, relations inter-entités et des règles de gestion desdits acteurs.

Les éléments du présent dossier découlent :

- De l'entretien avec le dirigeant de la société OC PIZZA lors de l'analyse des besoins du groupe.
- De l'entretien de la conception technique d'une solution d'un système de gestion de ses pizzerias pour le compte du client OC PIZZA.
- De l'entretien lors de la signature du contrat liant OC PIZZA et notre société, Openclassrooms.

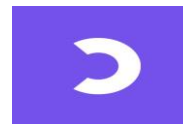
2.2 - Références

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

1. **PDOCPizza_01_lehchibi_gael_fonctionnelle** : Dossier de conception fonctionnelle de l'application
2. **PDOCPizza_01_lehchibi_gael_technique** : Dossier de conception technique de l'application
3. **PDOCPizza_01_lehchibi_gael_exploitation** : Dossier d'exploitation de l'application
4. **PDOCPizza_01_lehchibi_gael_finale** : PV de livraison finale de l'application



OC PIZZA



Openclassrooms

3 - ARCHITECTURE TECHNIQUE

3.1 - Application Web

Le développement de l'application sera réalisé en langage java pour la partie Backend de l'application (java 11 conformément à la version LTS en cours).

Le backend s'appuiera sur les technologies Maven (version 3.6.0) et Spring (version 2.2.4) pour l'organisation du code.

Le front sera réalisé à l'aide de l'intégration du Framework angular (version 9) pour gérer l'appel au service backend.

3.2 - Organisation de l'application

La pile logicielle est la suivante :

- Application **J2EE** (JDK version 11) pour les services :
 - Utilisateurs
 - Paiement
 - Livraison
 - Gestion des menus
- Serveur d'application **Tomcat** fourni avec SpringBoot.
- Application Angular pour la partie frontend appelant les services avec Socket I/O

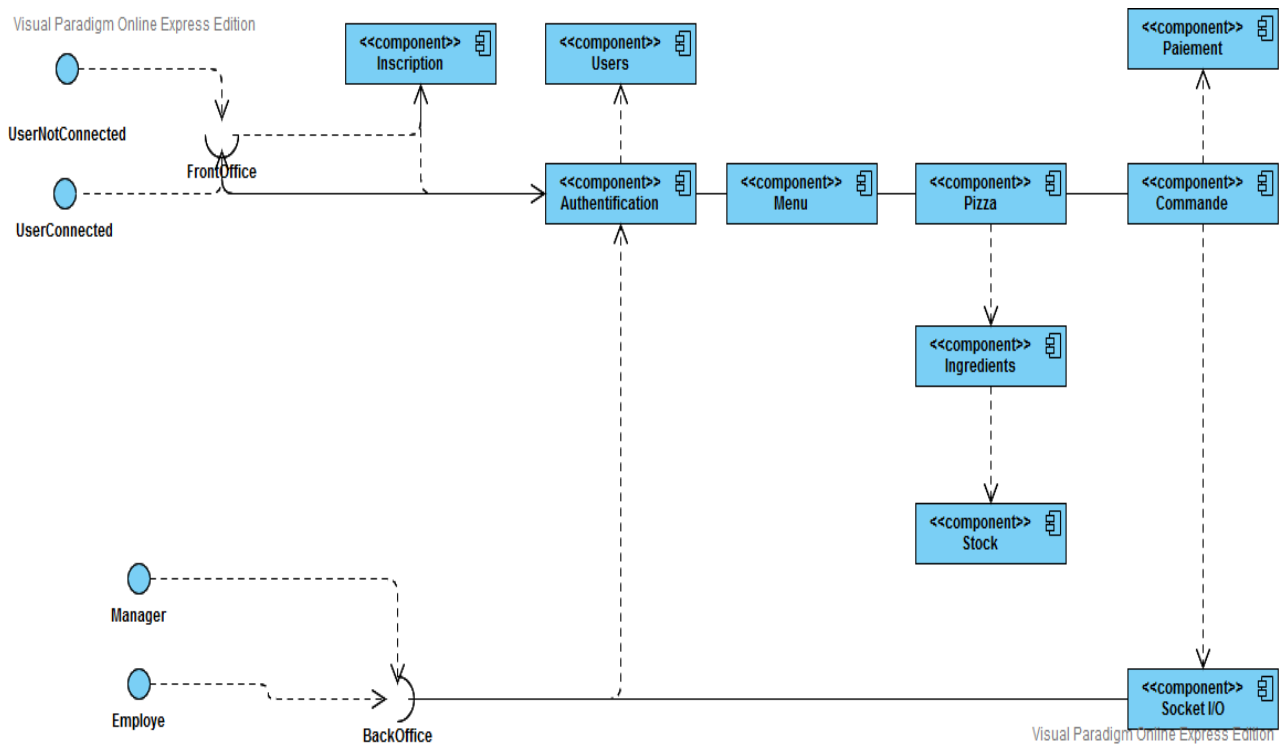
Nous rappelons l'architecture composant de l'application pour permettre le découpage des services ci-dessous.



OC PIZZA



Openclassrooms

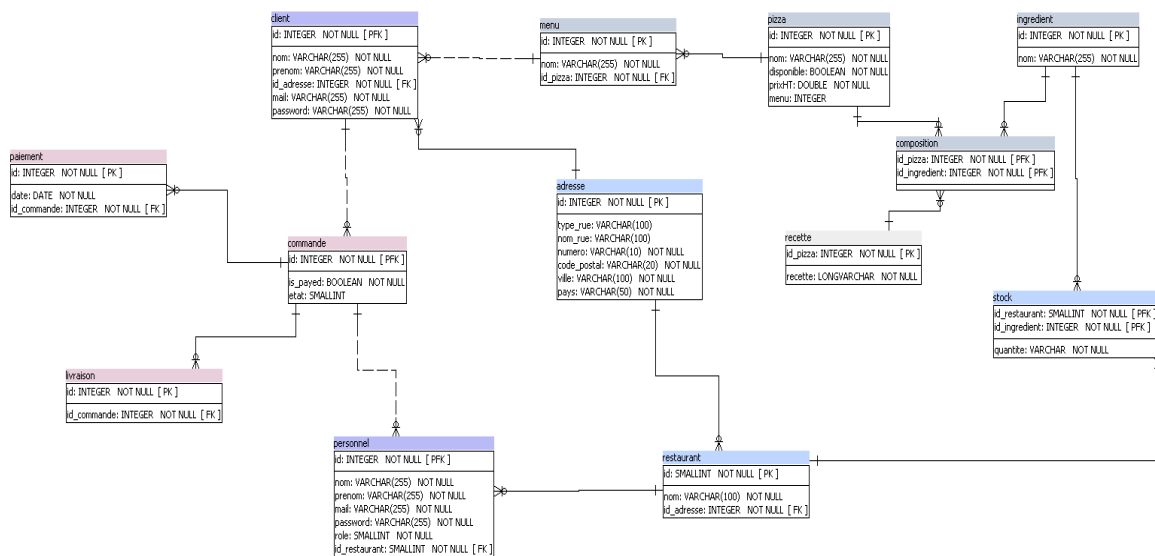


3.2.1 - Base de données

Les différents services utiliseront des bases de données spécifiques gérée avec le SGBD MySQL Entreprise Edition, ce choix se justifie par la fiabilité de ce système soutenu par Oracle.



Le modèle physique de donnée présenté ci-dessous est basé sur le diagramme de classe fourni dans le dossier de conception fonctionnelle.





OC PIZZA



Openclassrooms

4 - ARCHITECTURE DE DEPLOIEMENT

4.1 - Diagramme de déploiement

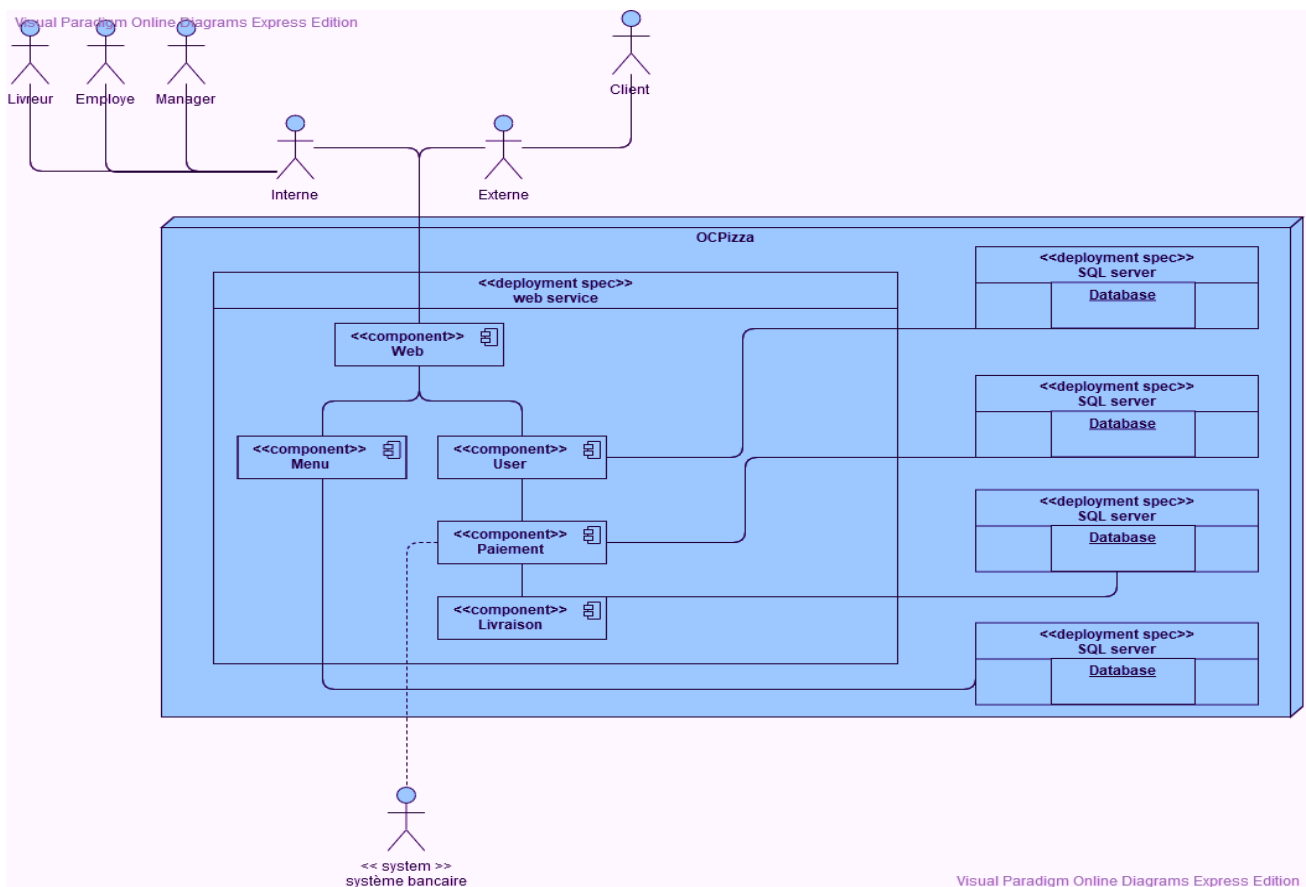


Diagramme UML de déploiement

Le diagramme ci-dessus représente le modèle de déploiement de l'application avec une application Java EE compose d'un back et front office se connectant à une base de données et un système bancaire pour la gestion des paiements.



OC PIZZA



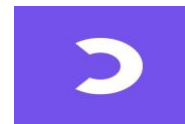
Openclassrooms

4.2 - Serveur de déploiement

L'application sera déployée sur un serveur lonos, reconnu pour le sérieux de leurs solutions d'hébergement. La connectivité en SSH permise par lonos permettra le déploiement rapide en cas de mise à jour de l'application.



OC PIZZA



Openclassrooms

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**. Le code source de l'application sera libéré sous forme de release et prendra en charge le versioning de l'application. Les développeurs devront donc utiliser les fonctionnalités des branches de git pour organiser le code

L'architecture applicative est la suivante :

- Un module **user** : responsable de la logique utilisateur de l'application
- Un module **livraison** : responsable des fonctionnalités de livraison.
- Un module **restauration** : responsable des fonctionnalités liés à la consultation des menus
- Un module **commande** : responsable des fonctionnalités liés à la commande, il sera en communication avec le service paiement fourni par la banque du client
- Un module **web** qui contiendra le déploiement de l'application angular pour l'affichage

5.1.1 - Les modules

Les modules seront divisés en services pour le fonctionnement de l'application. Les services communiqueront à l'aide d'un proxy fourni par Spring Gateway.

5.1.2 - 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 » voir ci-dessous)



OC PIZZA



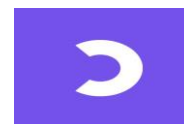
Openclassrooms

```
PizzaAPP
├── pom.xml
├── pizza-user
│   ├── pom.xml
│   └── src
│       ├── main
│       │   ├── java
│       │   └── resources
│       └── test
│           ├── java
│           └── resources
├── pizza-restauration
│   ├── pom.xml
│   └── src
│       ├── main
│       │   ├── java
│       │   └── resources
│       └── test
│           ├── java
│           └── ressources
└── src
    └── lib

- ...
```



OC PIZZA



Openclassrooms

6 - POINTS PARTICULIERS

6.1 - Gestion des logs

Les logs seront gérés par un fichier de configuration présent dans chaque module pour identifier plus rapidement les problèmes. Le fichier de configuration paramètrera la configuration de la dépendance log4j, un logeur reconnu pour ses qualités.

Les fichiers seront déposés dans un répertoire protégé pour l'accès futurs aux développeurs.

6.2 - Ressources

Les ressources graphiques pour le visuel de l'application seront fournies par le client OC PIZZA. Notre entreprise Openclassrooms s'impose de respecter la charte graphique de l'application fournie par ses clients. Un cahier des charges sera fourni aux développeurs de l'application ainsi qu'aux managers pour rappeler toutes les contraintes de l'application.

Nous rappelons également que sauf demande exceptionnelle, le jeu de donnée fourni lors de la livraison de l'application sera factice et ne permettra que de constater le fonctionnement normal de l'application à charge pour le client de remplir lui-même ses champs ou de nous transmettre ces données afin qu'elles soient implémentées, prix en dehors du développement de l'application.

6.3 - Environnement de développement

L'environnement de développement sera pris en charge par l'équipe de développement. L'utilisation d'un IDE (environnement de développement intégré) est préférable mais n'est pas obligatoire. Les développeurs seront en revanche responsable de l'application en cas de retard dû à une complexité du choix des équipes.

6.4 - Procédure de packaging / livraison

La livraison se fera par lots, en respectant les principes de la méthode agile SCRUM.

Une équipe métier sera dédié à ce projet pour accompagner les développeurs à la gestion du projet, un Scrum master sera également élu au sein de l'équipe de développement pour faciliter les contacts entre métiers et développeurs.

Les lots auront une périodicité à respecter, et seront eux même organisés en sprint (court laps de temps de développement avec des objectifs pour développer et prioriser des fonctionnalités d'une application).