

OC Pizza

Création du nouveau système informatique

Dossier de conception technique

Document référencé DCT-P20210008

Version 1.1

Auteur

Benjamin Breton

Analyste-programmeur

TABLE DES MATIÈRES

1 - Les versions.....	5
2 - Introduction.....	6
2.1 - Objet du document.....	6
2.2 - Références.....	6
3 - L'architecture Technique.....	7
3.1 - Composants généraux.....	7
3.1.1 - Définition	7
3.1.2 - Diagramme de composants.....	8
3.1.3 - Légende.....	9
3.1.4 - Composants détaillés	9
3.1.4.1 - Composant Account.....	9
3.1.4.2 - Composant Customer account.....	9
3.1.4.3 - Composant Pizzeria account.....	10
3.1.4.4 - Composant Employee account.....	10
3.1.4.5 - Composant Pizzaiolo account.....	10
3.1.4.6 - Composant Delivery man account.....	10
3.1.4.7 - Composant Waiter account.....	10
3.1.4.8 - Composant Direction account.....	11
3.1.4.9 - Composant Order.....	11
3.1.4.10 - Composant Order line	11
3.1.4.11 - Composant Status.....	11
3.1.4.12 - Composant Shipping informations.....	11
3.1.4.13 - Composant Payment.....	11
3.1.4.14 - Composant Stock.....	11
3.1.4.15 - Composant Pizza	11
3.1.4.16 - Composant Recipe.....	12
3.1.4.17 - Composant Informations.....	12
3.2 - Application Web.....	12
4 - L'architecture de déploiement.....	13
4.1 - Diagramme de déploiement.....	13

4.1.1 - Définition	13
4.1.2 - Diagramme de déploiement.....	14
4.1.3 - Légende.....	15
4.2 - Serveur de Base de données	16
4.2.1 - Caractéristiques techniques.....	16
4.2.2 - Modèle physique de données	16
4.2.2.1 - Définition.....	16
4.2.2.2 - Diagramme.....	17
4.2.2.3 - Légende.....	18
5 - L'architecture logicielle.....	19
5.1 - Principes généraux	19
5.1.1 - Les couches.....	19
5.1.2 - Les modules.....	19
5.1.3 - Structure des sources	20
5.1.3.1 - assets	21
5.1.3.2 - bin.....	21
5.1.3.3 - config.....	21
5.1.3.4 - public.....	21
5.1.3.5 - src.....	21
5.1.3.6 - templates.....	21
5.1.3.7 - tests.....	22
5.1.3.8 - translations.....	22
5.1.3.9 - var	22
6 - Les points particuliers	23
6.1 - Gestion des logs	23
6.2 - Fichiers de configuration	23
6.3 - Environnement de développement.....	23
6.4 - Procédure de packaging / livraison	24
7 - Index thématique alphabétique	25

1 - LES VERSIONS

Auteur	Date	Description	Version
Benjamin Breton	04/03/21	Création du document	1.0
Benjamin Breton	19/03/21	Première version terminée	1.1

2 – INTRODUCTION

2.1 – Objet du document

Le présent document constitue le dossier de conception technique du système informatique de la société *OC Pizza*.

Ce document a pour objectif de présenter les technologies utilisées pour faire face aux contraintes de réalisation de ce futur système.

Les éléments du présent dossier découlent :

- de la demande initiale du client ;
- des échanges avec le client ;
- du dossier de conception fonctionnelle référencé DCF-P20210008.

2.2 – Références

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

- doc. réf. **DCF-P20210008** : Dossier de conception fonctionnelle ;
- doc. réf. **DEX-P20210008** : Dossier d'exploitation.

3 – L'ARCHITECTURE TECHNIQUE

3.1 – Composants généraux

3.1.1 – Définition

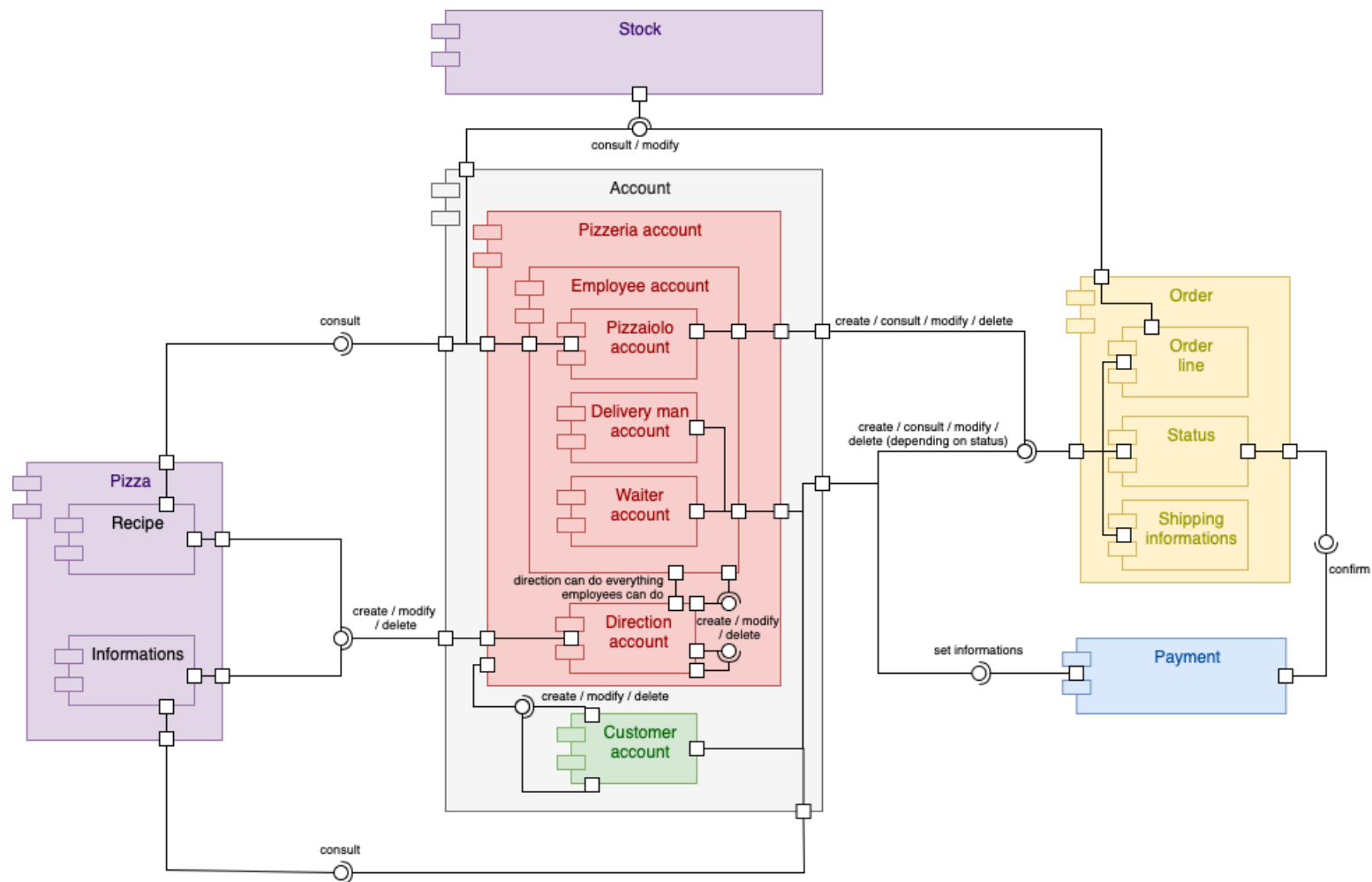
« Les diagrammes de composants décrivent le système modélisé sous forme de composants réutilisables et mettent en évidence leurs relations de dépendance. »¹

La page qui suit présente le diagramme de composants. Ensuite est indiquée la légende qui lui correspond.

¹ Définition tirée de la page :

<https://laurent-audibert.developpez.com/>

3.1.2 - Diagramme de composants



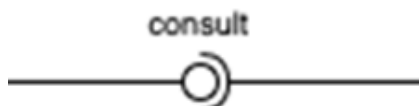
3.1.3 - Légende

> Composant



Un composant est représenté par un rectangle comprenant deux autres rectangles sur le côté gauche.

> Interface

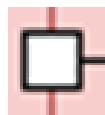


L'interface se caractérise par un cercle connecté à un demi-cercle.

Le composant relié au demi-cercle est celui qui agit (dans l'exemple ci-dessus celui qui consulte).

Le composant relié au cercle est celui qui est requis pour l'action (dans l'exemple ci-dessus, celui qui est consulté).

> Port



Un port est représenté par un carré blanc sur le bord d'un composant, lui permettant d'être relié à un autre composant.

> Délégation



Une délégation entre deux composants est marquée par deux ports reliés par une ligne.

3.1.4 - Composants détaillés

3.1.4.1 - Composant Account

Ce composant comporte tous les comptes permettant de se connecter au système.

3.1.4.2 - Composant Customer account

Ce composant correspond au compte client. Le compte client doit lui permettre de pouvoir passer des commandes, les suivre et gérer les données personnelles. Ainsi, avec son compte, le client peut :

- consulter les informations des pizzas ;
- entrer des informations de paiement ;
- créer, consulter, modifier, et effacer une commande (la modification et la suppression d'une commande dépend de son statut : si la commande est en cours de préparation, ces actions sont bloquées) ;
- modifier et effacer les données du compte.

3.1.4.3 - Composant Pizzeria account

Ce composant comporte tous les comptes des membres des pizzerias. Les membres de la pizzeria peuvent créer, modifier et effacer les comptes client.

3.1.4.4 - Composant Employee account

Ce composant comporte tous les comptes des membres du personnel des pizzerias.

3.1.4.5 - Composant Pizzaiolo account

Ce composant correspond au compte pizzaiolo. Le compte pizzaiolo fait partie des comptes du personnel des pizzerias. Il permet au pizzaiolo de :

- consulter les commandes et en modifier le statut ;
- consulter et modifier le stock ;
- consulter les recettes.

3.1.4.6 - Composant Delivery man account

Ce composant correspond au compte livreur. Le compte livreur fait partie des comptes du personnel des pizzerias. Il permet au livreur de :

- consulter les commandes et en modifier le statut ;
- entrer les informations de paiement si le client n'a pas payé en ligne.

3.1.4.7 - Composant Waiter account

Ce composant correspond au compte serveur. Le compte serveur fait partie des comptes du personnel des pizzerias. Il permet au serveur de :

- créer, consulter, modifier, et effacer les commandes ;
- consulter les informations des pizzas ;
- entrer les informations de paiement.

3.1.4.8 - Composant Direction account

Ce composant correspond au compte des membres de la direction. Le compte direction est le compte des gérants et des responsables des pizzerias. Il permet :

- d'effectuer tout ce que les membres du personnel peuvent faire ;
- de gérer tous les comptes présents sur le système ;
- de créer, modifier, et effacer les informations et les recettes des pizzas.

3.1.4.9 - Composant Order

Ce composant comporte toutes les informations sur la commande.

3.1.4.10 - Composant Order line

Ce composant correspond aux lignes de commande. Une ligne de commande existe pour chaque pizza commandée et permet d'en obtenir la quantité commandée et les éventuelles indications du client. Le stock est modifié selon les quantités commandées par le client.

3.1.4.11 - Composant Status

Ce composant correspond au statut actif de la commande. Lorsque la commande est remise au client, la modification de son statut permet de confirmer le paiement.

3.1.4.12 - Composant Shipping informations

Ce composant comporte toutes les indications nécessaires au livreur pour livrer la commande.

3.1.4.13 - Composant Payment

Ce composant correspond aux informations de paiement.

3.1.4.14 - Composant Stock

Ce composant contient les ingrédients et leurs quantités contenus dans le stock.

3.1.4.15 - Composant Pizza

Ce composant contient toutes les informations relatives aux pizzas : recettes et informations à faire apparaître sur la page publique du site.

3.1.4.16 - Composant Recipe

Ce composant correspond aux recettes des pizzas. Elles sont consultées par le pizzaiolo et modifiées par les dirigeants.

3.1.4.17 - Composant Informations

Ce composant comporte les informations à indiquer sur le site pour chaque pizza. Elles sont consultées par le serveur et le client, et sont modifiées par les dirigeants.

3.2 - Application Web

La pile logicielle est la suivante :

- Linux (Ubuntu v. 20.04) en tant que système d'exploitation ;
- Apache (v. 2.4.46) en guise de serveur Web ;
- MySQL (v. 8.0.23) en base de données ;
- PHP (v. 8.0.2) en interpréteur de script ; couplé au framework symfony (v. 5.2.4)

Ce sont des logiciels open source, ont une licence dite libre et sont gratuits. Cette pile logicielle est tout à fait adaptée au développement de sites Web dynamiques, et bénéficie d'un support très actif de la communauté, qui est très large, et de tutoriaux très concis.

4 – L'ARCHITECTURE DE DÉPLOIEMENT

4.1 - Diagramme de déploiement

4.1.1 - Définition

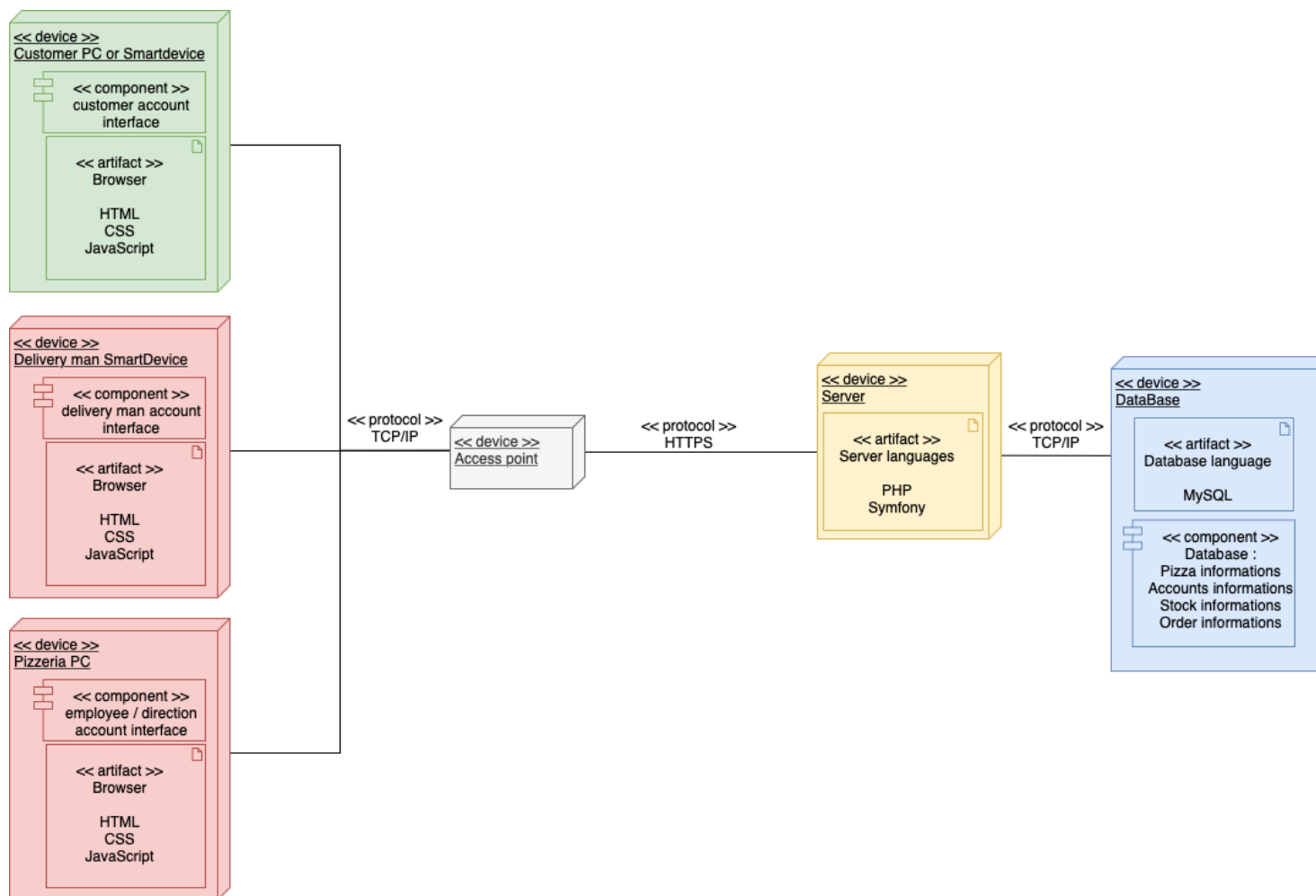
« Un diagramme de déploiement décrit la disposition physique des ressources matérielles qui composent le système et montre la répartition des composants sur ces matériels. »²

La page qui suit présente le diagramme de déploiement. Ensuite est indiquée la légende qui lui correspond.

² Définition tirée de la page :

<https://laurent-audibert.developpez.com/>

4.1.2 - Diagramme de déploiement



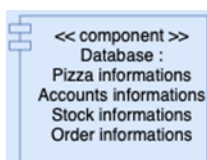
4.1.3 - Légende

> Nœud



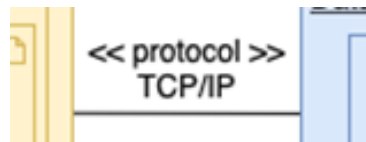
Les nœuds, représentés par des cubes, sont des composants mécaniques.

> Composants



Un composant est représenté par un rectangle comprenant deux autres rectangles sur le côté gauche.

> Connexions



Les connexions sont des liens de communication.

> Artefacts



Un artefact est un rectangle avec une icône de fichier dans le coin supérieur droit. C'est une manière de définir un fichier, un programme, des langages utilisés.

4.2 - Serveur de Base de données

4.2.1 - Caractéristiques techniques

- **Processeur :** Intel Xeon E3-1245v5 - 4 c / 8 t - 3.5 GHz / 3.9 GHz
- **Mémoire :** 32 Go DDR4 ECC 2 133 MHz
possibilités :
 - 64 Go DDR4 ECC 2 133 MHz
- **Stockage :** 2 x 2 To HDD SATA Soft RAID
possibilités :
 - 2 x 480 Go SSD SATA Soft RAID
- **Bande passante publique :** 250 Mbit/s 10 To trafic
possibilités :
 - 100 Mbit/s illimité
 - 250 Mbit/s illimité

4.2.2 - Modèle physique de données

4.2.2.1 - Définition

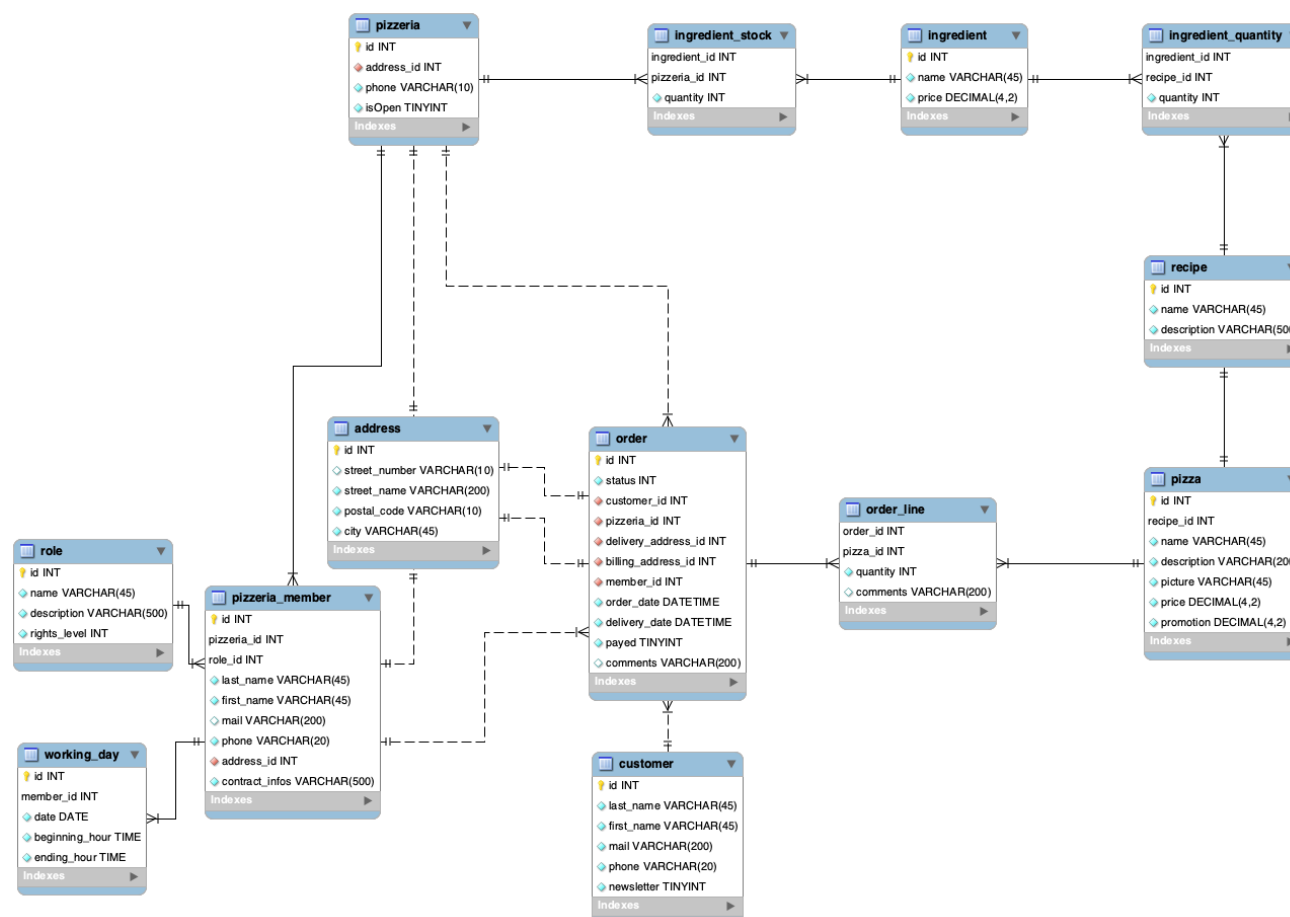
Il s'agit de la « structure finale de la base de données avec les différents liens entre les éléments qui la composent »³ [entre crochets ci-dessous la correspondance dans le diagramme de classes présenté dans le dossier de conception fonctionnelle référencé DCF-P20210008] :

- les tables [classes, et classes d'association] ;
- les attributs [propriétés] ;
- les clés [id pour les clés primaires] ;
- les relations sont marquées par des clés étrangères au sein des attributs [les relations existent dans le diagramme de classe mais sans clés].

³ Définition tirée de la page :

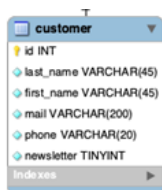
<https://www.base-de-donnees.com/mpd/>

4.2.2.2 - Diagramme



4.2.2.3 - Légende

> les tables



Le nom de la table est indiqué en gras dans l'encadré du haut. Les attributs sont dans l'encadré blanc.

> les attributs obligatoires



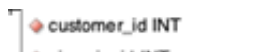
> les attributs optionnels



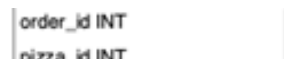
> les clés primaires



> les clés étrangères

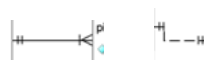


> les clés étrangères également primaires



> les relations

• lignes continues et en pointillés



Une ligne continue indique qu'une clé primaire d'une table sert également de clé primaire dans l'autre.

Sinon elle est en pointillés.

• multiplicités



Une ligne marquée par un chevron (> ou <) indique une relation un à plusieurs.



Sinon il n'y a pas de chevrons.

Il n'y a pas de multiplicité plusieurs à plusieurs car ces multiplicités possèdent une table correspondant antérieurement à une classe d'association.

5 - L'ARCHITECTURE LOGICIELLE

5.1 - Principes généraux

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

5.1.1 - Les couches

L'architecture applicative respectera le MVC et sera donc la suivante :

- une couche *View* : responsable de l'interface client/application ;
- une couche *Model* : responsable de la logique ;
- une couche *Controller* : responsable de la réception et de l'interprétation des interactions avec le client et de la réponse à lui donner via la vue au regard de la logique comprise dans le modèle.

5.1.2 - Les modules

Les modules utilisées sont ceux du framework *symfony*, à savoir le module *Doctrine* qui est le module par défaut permettant de communiquer avec la base de données.

5.1.3 - Structure des sources

La structuration des répertoires du projet suit la logique par défaut du framework *symfony*, qui est la suivante pour sa version 5 :

```
Racine
├── assets
├── bin
│   ├── console
│   └── phpunit
├── config
│   ├── packages
│   │   ├── dev
│   │   ├── prod
│   │   ├── test
│   │   ├── doctrine.yaml
│   │   ├── doctrine_migrations.yaml
│   │   ├── framework.yaml
│   │   ├── routing.yaml
│   │   ├── security.yaml
│   │   └── twig.yaml
│   ├── routes
│   │   ├── dev
│   │   └── annotations.yaml
│   ├── bundles.php
│   ├── routes.yaml
│   └── services.yaml
├── public
│   └── index.php
├── src
│   ├── Controller
│   ├── Entity
│   ├── Migrations
│   ├── Repository
│   └── Kernel.php
├── templates
├── tests
├── translations
├── var
│   ├── cache
│   └── log
├── composer.json
├── composer.lock
├── package.json
└── symfony.lock
```

5.1.3.1 - assets

Ce dossier contient tous les fichiers .css et .js qui seront par la suite compilés pour le front du système informatique.

5.1.3.2 - bin

Ce dossier contient les fichiers de commandes.

5.1.3.3 - config

Ce dossier contient la configuration des packages, services et routes.

Les fichiers de configuration sont par défaut en *yaml*.

Il est cependant possible d'utiliser des fichiers *php* ou *xml*.

5.1.3.4 - public

Ce dossier contient le point d'entrée de l'application, et donc le fichier index.php.

Il peut être utilisé pour mettre à disposition des fichiers de ressources.

5.1.3.5 - src

Ce dossier contient tout le code *php*.

Les dossiers qui seront obligatoires à utiliser pour le fonctionnement de l'application sont :

- Contrôleur : définition des points d'entrée de votre application ;
- Entity : définition de la structure de la base de données. Chaque Entity représente généralement une table. La commande `php bin/console doctrine:migrations` permettra de mettre à jour la base à chaque modification de l'Entity ;
- Repository : un Repository est toujours rattaché à une Entity, il permet de créer des fonctions qui iront requêter la table (ainsi que les tables liées). *symfony* utilise l'ORM *Doctrine* qui permet de créer nos requêtes SQL à travers les queryBuilder.

5.1.3.6 - templates

Ce dossier contient les fichiers *Twig* qui viennent rajouter quelques fonctionnalités au HTML classique.

5.1.3.7 - tests

Ce dossier contient la définition des tests unitaires PHPUnit pour tester le système.

5.1.3.8 - translations

Ce dossier contient la traduction du système grâce au package translation.

5.1.3.9 - var

Ce dossier contient le cache et les fichiers de log.

6 – LES POINTS PARTICULIERS

6.1 – Gestion des logs

Les logs permettent d'assurer un suivi dans l'utilisation du système.

Pour ce suivi, il sera utilisé la bibliothèque *Monolog* permettant de journaliser différentes informations lors de l'exécution de scripts php.

Son intégration via *symfony* se fait de manière quasi automatique.

6.2 – Fichiers de configuration

Les fichiers de configuration sont dans le dossier config.

La configuration du comportement du système se fait dans le fichier services.yaml.

6.3 – Environnement de développement

L'environnement de développement est le suivant :

- système d'exploitation : *MacOS* (v. 11.2.2) ;
- serveur local :
 - *Apache* (v. 2.4.46);
 - *MySQL* (v. 8.0.23);
 - *PHP* (v. 8.0.2), *symfony* (v. 5.2.4) ;
- création du MPD : *MySQL Workbench* (v. 8.0.23) ;
- IDE : *Sublime Text* (v. 3.2.2).

6.4 - Procédure de packaging / livraison

Les fichiers du système seront livrés avec les dumps de la fausse base de données.

Ils se trouveront dans un dossier compressé et seront envoyés sur un serveur FTP. Le téléchargement du dossier pourra se faire au moyen d'un logiciel tel que *FileZilla* au moyen des codes d'accès qui seront transmis par mails séparés.

Ils seront également stockés sur *GitHub* via un lien présent sur le PV de livraison.

7 - INDEX THÉMATIQUE ALPHABÉTIQUE

Les nombres indiqués renvoient aux numéros de page.

A

Application web

pile logicielle 13

Architecture logicielle

couches..... 21
modules 21
principes 21
structure des sources..... 22

Architecture technique

application web..... 13
composants 7

B

Base de données

caractéristiques techniques.... 17
modèle physique de données 17

C

Composants (diagramme)

détails..... 10
diagramme (définition)..... 7

diagramme (représentation
graphique) 8
légende..... 10

Configuration

fichiers 25

D

Dossier de conception fonctionnelle

référence..... 6

Dossier de conception technique

architecture de déploiement... 14
architecture logicielle 21
architecture technique..... 7
base de données 17
objet 6
version 5

Déploiement (diagramme)

définition 14
légende..... 16
représentation graphique..... 15

Développement

environnement..... 25

Dossier d'exploitation

référence 6

Diagramme

composants 8
déploiement..... 14
modèle physique de données.17

L

Livraison

procédure 26

Logiciel

pile logicielle 13

Logs

gestion..... 25

M

Modèle physique de données

définition..... 17
légende 20
représentation graphique..... 19

