

<LogoClient>

<LogoEntreprise>

Open Classrooms Pizza

Solution de gestion de commande

Dossier de conception technique

Version 1.0

Auteur

Mathieu Janneau

TABLE DES MATIERES

1 -	3
2 - Introduction	4
2.1 - Objet du document	4
2.2 - Références	4
3 - Architecture Technique	5
3.1 - APPLICATION CLIENT	5
3.1.1 - Interface Client	5
3.1.1.1 - Composant Gestion de compte	5
3.1.1.2 - Composant Gestion des commandes	5
3.1.1.3 - Composant Présentation Menu	5
3.2 - APPLICATIONS EMPLOYES	6
3.2.1 - Composant Interface Employé	6
3.2.2 - Composant Gestion de compte	6
3.2.3 - Composant Gestion des commandes	6
3.2.3.1 - Sous - Composant Outil prise de commande	7
3.2.3.2 - Sous - Composant Outil cuisine / recette	7
3.2.3.3 - Sous - Composant Outil Livraison	7
3.2.4 - Composant File d'attente	7
3.3 - Administration	8
3.3.1 - Composant Authentification	8
3.3.2 - Composant Gestion compte	8
3.3.3 - Composant Gestion des stocks	9
3.3.4 - Composant Gestion des Menu	9
3.3.5 - Composant File d'attente	9
3.4 - Base de données	10
3.4.1 - Composant table Clients	10
3.4.2 - Composant table Employé	10
3.4.3 - Composant table Commandes	10
3.4.4 - Composant table Produit	10
3.4.5 - Composant table Stock	10
3.4.6 - Composant table Stock	11
4 - Architecture de Déploiement	13
4.1 - Serveur de Base de données	15
5 - Architecture logicielle	15
5.1 - Principes généraux	16
5.1.1 - Structure des sources	16
6 - Points particuliers	17
6.1 - Gestion des logs	17
6.2 - Fichiers de configuration	17
6.2.1.1 - Datasources	17
229.1 - Ressources	22
229.2 - Environnement de développement	26
230 - Glossaire	27

<LogoClient>

<LogoEntreprise>

1 - Versions

Auteur	Date	Description	Version
Mathieu Janneau	09/01/2018	Création du document	1.0

2 - Introduction

2.1 - Objet du document

Le présent document constitue le dossier de conception technique de la solution de gestion de commande de OC PIZZA

L'objectif de ce document est de mettre en avant la structure technique de la solution. Ce dernier décrira :

- les différents composants internes et externes constitutifs de la solution,
- la description de l'organisation physique de ces composants
- la description du domaine fonctionnel
- la description des fichiers de configuration de la base de donnée

Les éléments du présent dossier découlent :

- De dossier conception fonctionnelle 26/11/2017
- ...

2.2 - Références

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

1. **DCF – 1.0:** Dossier de conception fonctionnelle de l'application
2. Des retours de la présentation du DCF le 27/11/2017

3 - Architecture Technique

3.1 - APPLICATION CLIENT

3.1.1 - Interface Client

L'interface client gère l'ensemble des interactions du client avec la solution OC Pizza

3.1.1.1 - Composant Gestion de compte

Ce composant aura en charge la gestion de toutes les fonctionnalités en relation avec le compte utilisateur de l'inscription à l'authentification en passant par la modification des données. Pour cela, il s'appuiera sur deux composants externes :

- La base de données pour récupérer et valider les informations ou bien envoyer des informations lors de l'inscription ou de la modification du compte
- Le service de Géolocalisation pour situer ses coordonnées et valider sa couverture de livraison

3.1.1.2 - Composant Gestion des commandes

Ce composant prend en charge l'ensemble des fonctionnalités ayant trait à la commande. De l'ajout des articles dans le panier, la validation de la commande, le paiement de celle-ci à l'annulation si nécessaire.

Pour remplir ces fonctions, il requiert les composants suivants :

- Le composant gestion de compte afin de procéder à l'authentification de l'utilisateur.
- Le composant base de données afin d'enregistrer les informations dans la BDD et les transmettre aux magasins
- Le composant menu pour constituer le panier d'achat
- Le composant externe service de paiement pour procéder au paiement en ligne

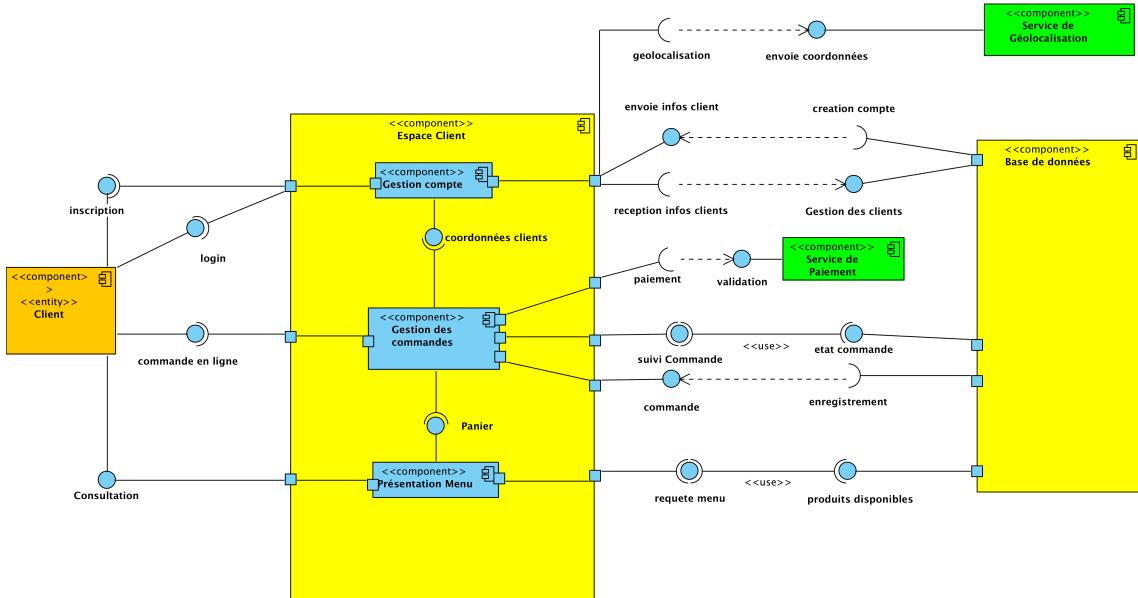
3.1.1.3 - Composant Présentation Menu

Ce composant a pour rôle la présentation des produits disponibles à la vente sous formes de fiches produits (nom, photo, prix). Il permet de rassembler les données nécessaires et de les présenter au client

- grâce à son interaction avec la table « produits » de la base de données.

<LogoClient>

<LogoEntreprise>



3.2 - APPLICATIONS EMPLOYES

3.2.1 - Composant Interface Employé

L'interface client gère l'ensemble des interaction des acteurs employés avec la solution OC Pizza quelques soit leur spécialisation :

- Commercial
- Cuisinier
- livreur

3.2.2 - Composant Gestion de compte

Ce composant aura en charge la gestion de toutes les fonctionnalités en relation avec le compte utilisateur de l'authentification à l'attribution des accès aux différents sous composants de l'interface employé. Pour cela, il s'appuiera :

- La base de donnée pour récupérer et valider les informations

De plus il est requis par tous les autres composants de l'interface pour des raisons de sécurité

3.2.3 - Composant Gestion des commandes

Ce composant prend en charge l'ensemble des fonctionnalités ayant attrait à la commande. De l'ajout des articles dans le panier, la validation de la commande, le paiement de celle-ci à l'annulation si nécessaire.

Il permet aux employés de types commerciaux d'enregistrer des commandes en boutique ou par téléphone

Pour remplir ces fonctions, il requiert les composants suivants :

- Le composant gestion de compte afin de procéder à l'authentification de l'utilisateur.
- Le composant base de données afin d'enregistrer les informations dans la BDD et les transmettre aux magasins
- Le composant externes service de paiement pour procéder au paiement en ligne

3.2.3.1 - Sous - Composant Outil prise de commande

Ce composant est dédié aux employés de spécialisation « commerciale ». Il permet de procéder à :

- La prise de commande
- L'annulation de commande
- Le paiement de commande

Ce sous-composant dépend du composant externe Service de paiement

3.2.3.2 - Sous - Composant Outil cuisine / recette

Ce composant est dédié aux employés de spécialisation « cuisinier ». Il permet de procéder à :

- La prise en charge d'une préparation
- La consultation des recettes

3.2.3.3 - Sous - Composant Outil Livraison

Ce composant est dédié aux employés de spécialisation « livreur ». Il permet de procéder à :

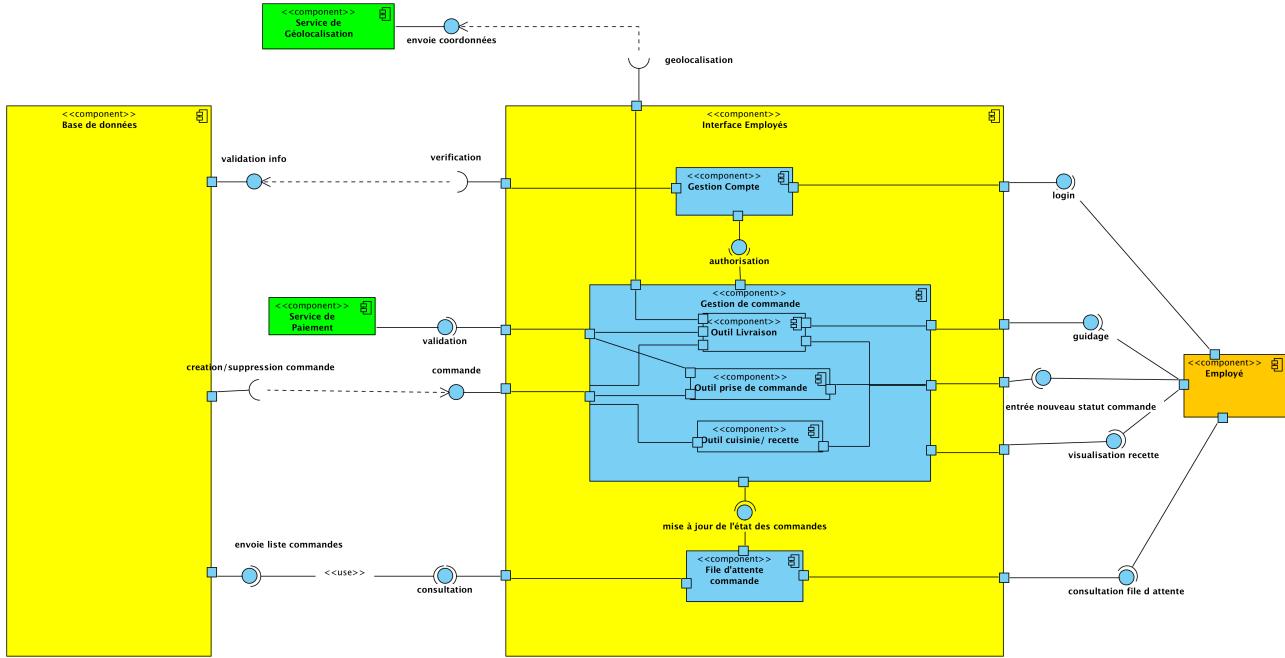
- La prise en charge de produit à livrer
- Le guidage à l'adresse de livraison
- La confirmation de livraison
- Le paiement

Ce sous-composant dépend des composants externe Service de Géolocalisation et Service de paiement

3.2.4 - Composant File d'attente

Ce composant a pour rôle la présentation de l'état de toutes les commandes à un instant T. il donne accès aux différents composants spécialisés selon les droits d'accès. Il permet donc la prise en charge de commandes. Ce composant requiert :

- les informations de la table Commande de la BDD
- l'accès à un des composants spécialisé



3.3 - Administration

Le composant administration prend en charge l'ensemble des fonctionnalités dédiées à l'administrateur

- Ajout/suppression produit au menu
- Ajout/suppression boutiques et employés
- Ajout/suppression de produit des stocks
- Consultation en temps réel des stocks et des commandes

3.3.1 - Composant Authentification

Ce composant aura en charge la gestion de l'accès au Back Office aux différents sous composants de l'interface administrateur. Pour cela, il s'appuiera :

- La base de donnée pour récupérer et valider les informations

De plus il est requis par tous les autres composants de l'interface pour des raisons de sécurité

3.3.2 - Composant Gestion compte

Ce composant permet à l'administrateur d'intervenir sur les bases de données clients, employé, boutique. Il peut ainsi gérer et attribuer les droits d'accès, ajouter ou supprimer des utilisateurs et/ou boutiques. Pour cela, il s'appuie sur :

- La base de donnée pour récupérer et valider les informations

3.3.3 - Composant Gestion des stocks

Ce composant permet à l'administrateur d'intervenir sur les bases de données stocks. Il peut ainsi ajouter, supprimer et modifier la table ingrédients. Ce composant a aussi pour fonction d'alerter l'administrateur lorsqu'une quantité seuil minimale est atteinte par un ingrédient il peut alors contacter le fournisseur correspondant pour passer une commande. Pour cela, il s'appuie sur :

- Les tables Stock et Fournisseur de la base de donnée pour récupérer et valider les informations

3.3.4 - Composant Gestion des Menu

Ce composant donne la possibilité à l'administrateur d'ajouter, modifier supprimer des produits de sa carte. Il peut relier chaque produit à des ingrédients et fournir les recettes à ses employés. Ce composant permet de composer une carte et de la rendre visible ou non au public.

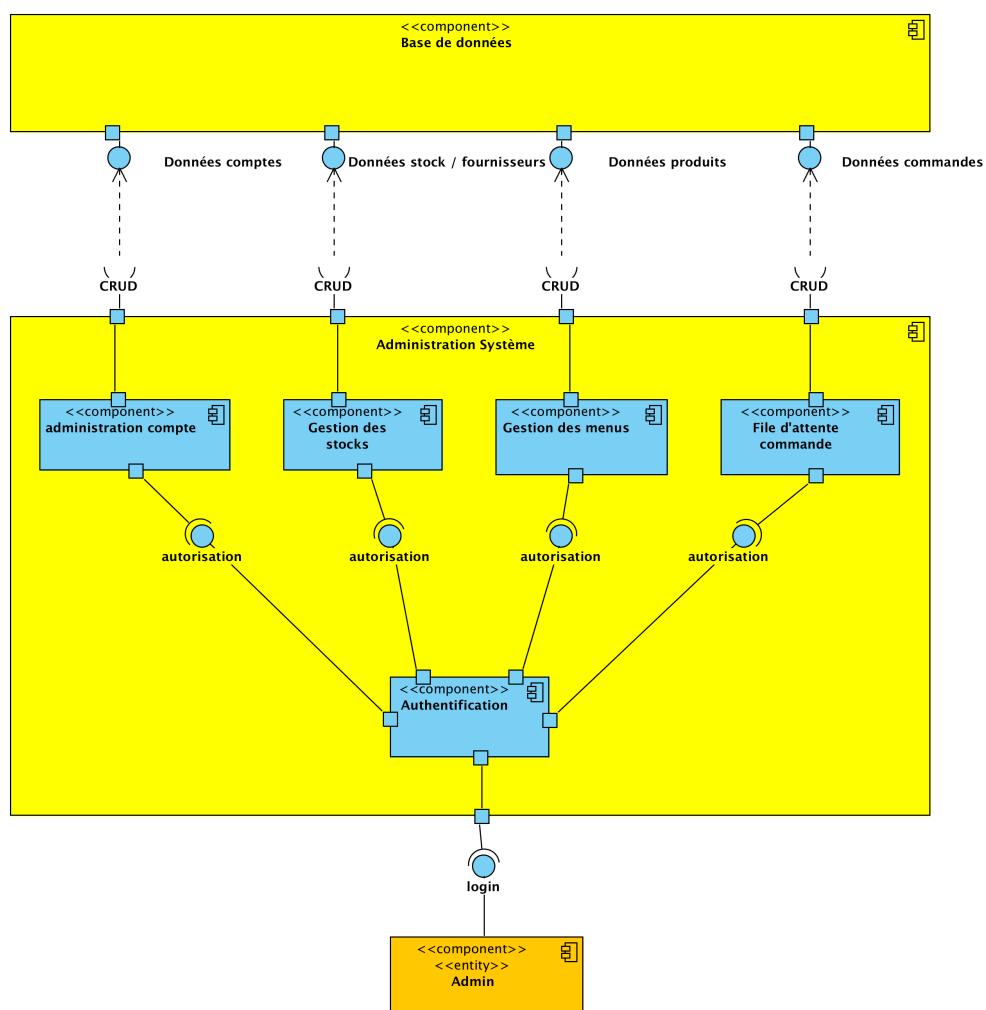
. Pour cela, il s'appuie sur :

- La table Produits de la base de donnée pour récupérer et valider les informations

3.3.5 - Composant File d'attente

Ce composant a pour rôle la présentation de l'état de toutes les commandes à un instant T. Ce composant requiert :

- les informations de la table Commande de la BDD



3.4 - Base de données

Le composant BDD prend en charge le stockage et la persistance des données du système

3.4.1 - Composant table Clients

Gère le stockage des informations clients.

Il interagit avec les composants suivants :

- Commande de la BDD
- Gestion de compte des packages (client/employé/administrateur)

3.4.2 - Composant table Employé

Gère le stockage des informations « Employé ».

Il interagit avec les composants suivants :

- Table Commande de la BDD
- Gestion de compte des packages (employé/administrateur)

3.4.3 - Composant table Commandes

Gère le stockage des informations relatives aux commandes.

Il requiert les composants suivants :

- Table Client
- Table Employé
- Table Produits

Il interagit avec les composants suivants :

- File d'attente des packages (employé/administrateur)

3.4.4 - Composant table Produit

Gère le stockage des informations Produits.

Il interagit avec les composants suivants :

- La table Commande de la BDD
- Gestion des menus Package Administration

Il requiert le composant table Stock de la BDD

3.4.5 - Composant table Stock

Gère le stockage des informations Produits.

Il interagit avec les composants suivants :

- Table Produit de la BDD
- Gestion des Stocks Package Administration

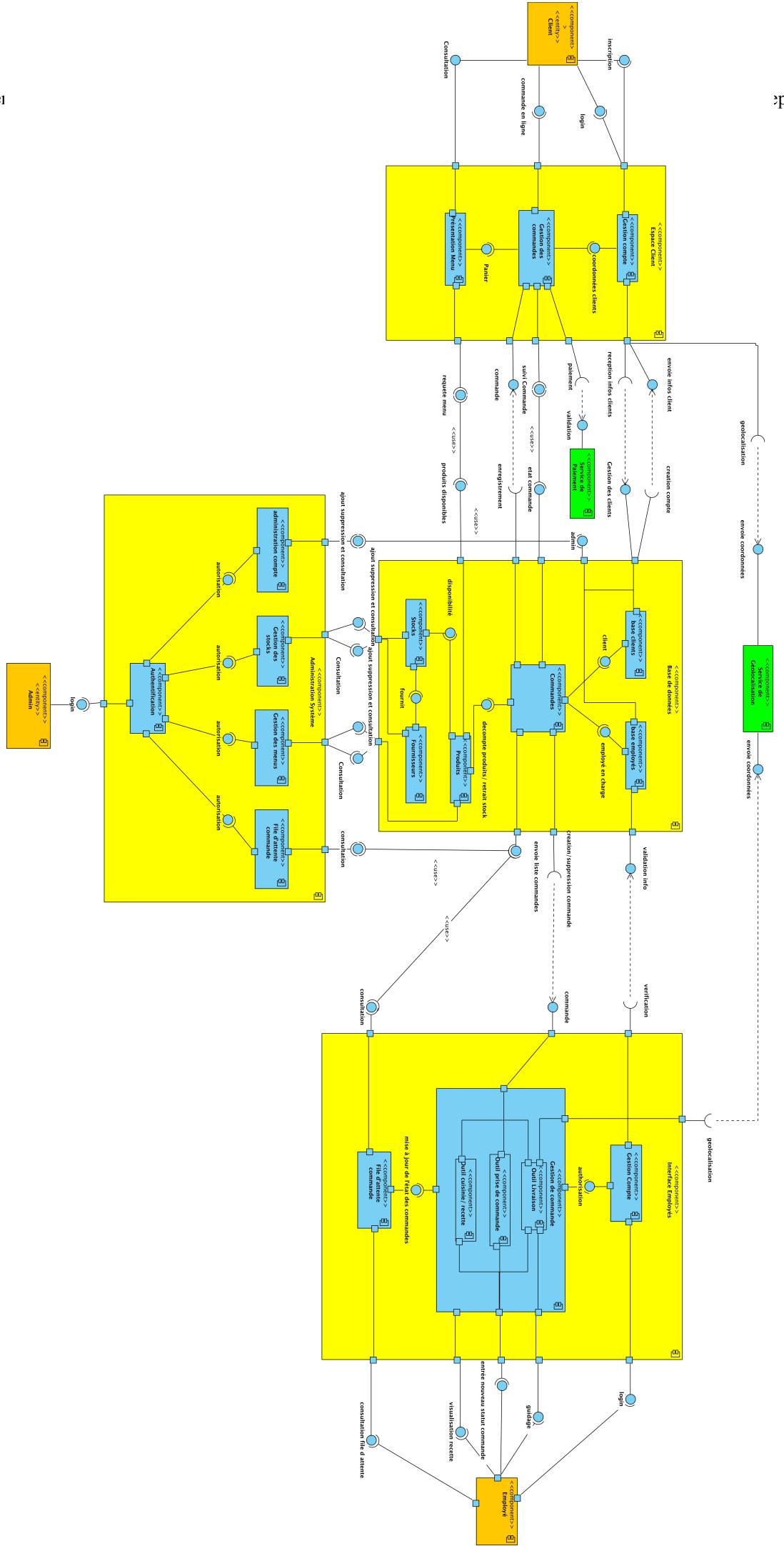
Il requiert le composant table Fournisseurs de la BDD

3.4.6 - Composant table Stock

Gère le stockage des informations Produits.

Il interagit avec les composants suivants :

- Table Stock de la BDD
- Gestion des Stocks Package Administration



4 - Architecture de Déploiement

Les différents packages de la solution seront déployés sur différentes plateformes afin d'optimiser leur utilisation.

L'interface Client sera déployée sur les devices suivants :

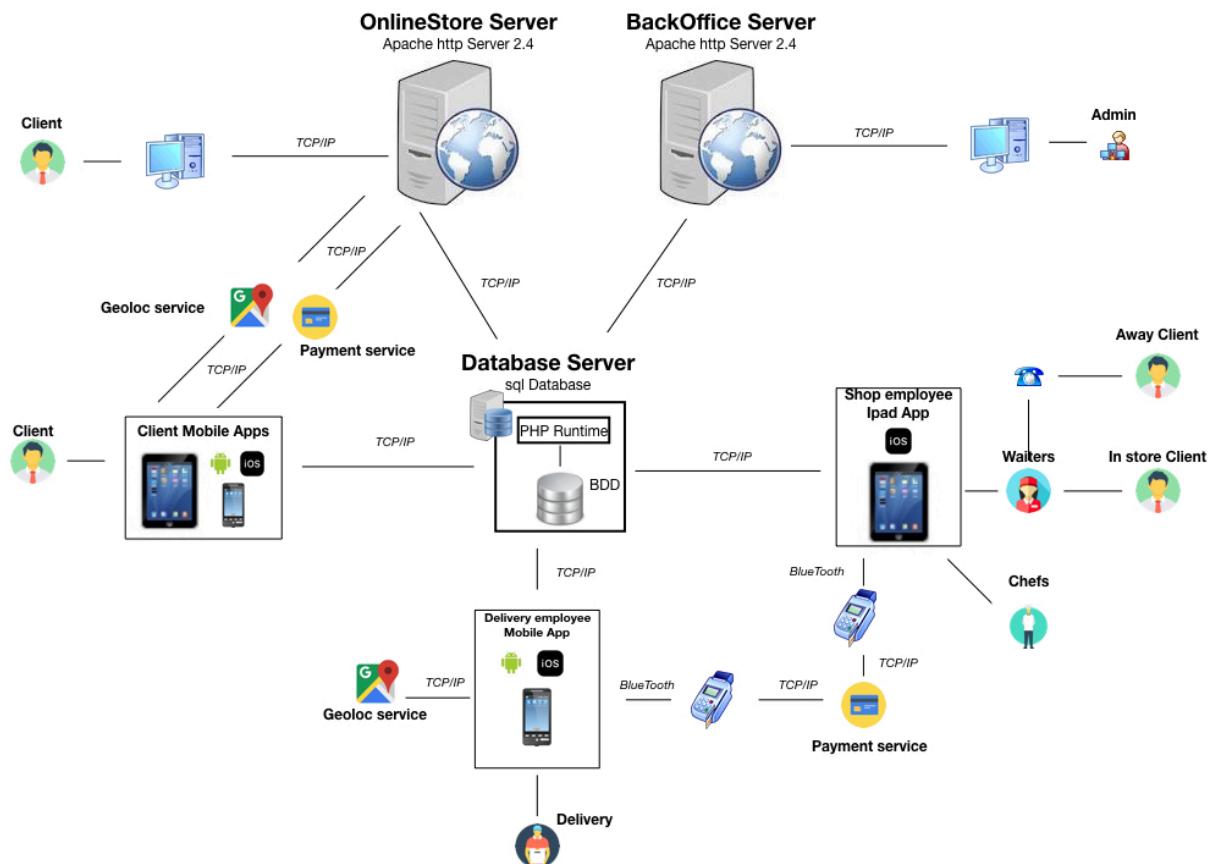
- Plateforme Web
- Application ios tablette mobile
- Application Android tablette mobile

Ce choix a été fait pour optimiser l'accèsibilité de la solution côté client

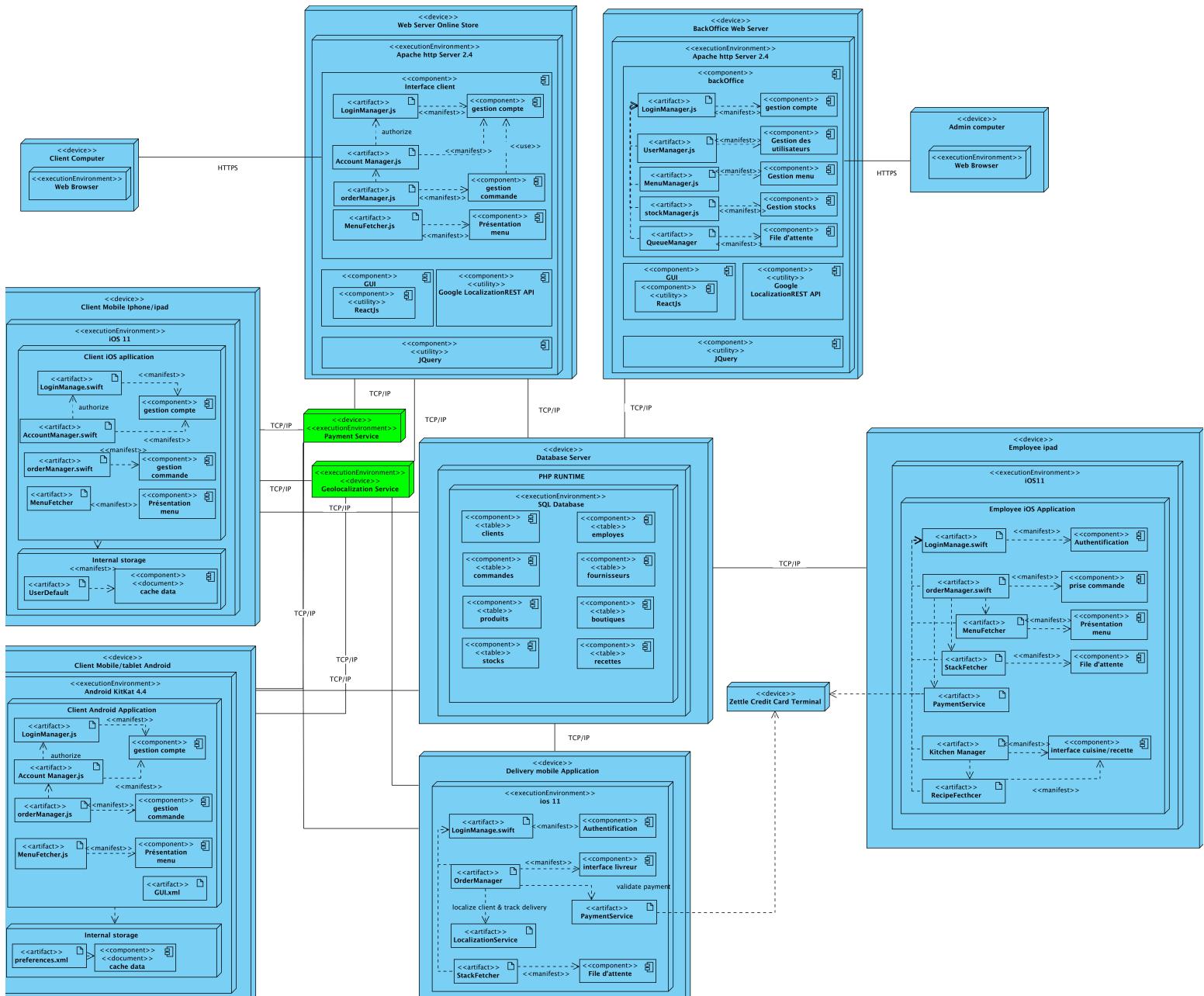
L'interface Employé sera déployée sur tablette ipad dans les boutiques de l'enseigne pour des raisons d'ergonomie et sur les mobiles iOS et Android pour les livreurs afin de ne pas se couper d'une population d'employés et ou avoir à fournir des iphones

4.1.1 - Diagramme de déploiement simplifié

Diagramme de déploiement simplifié

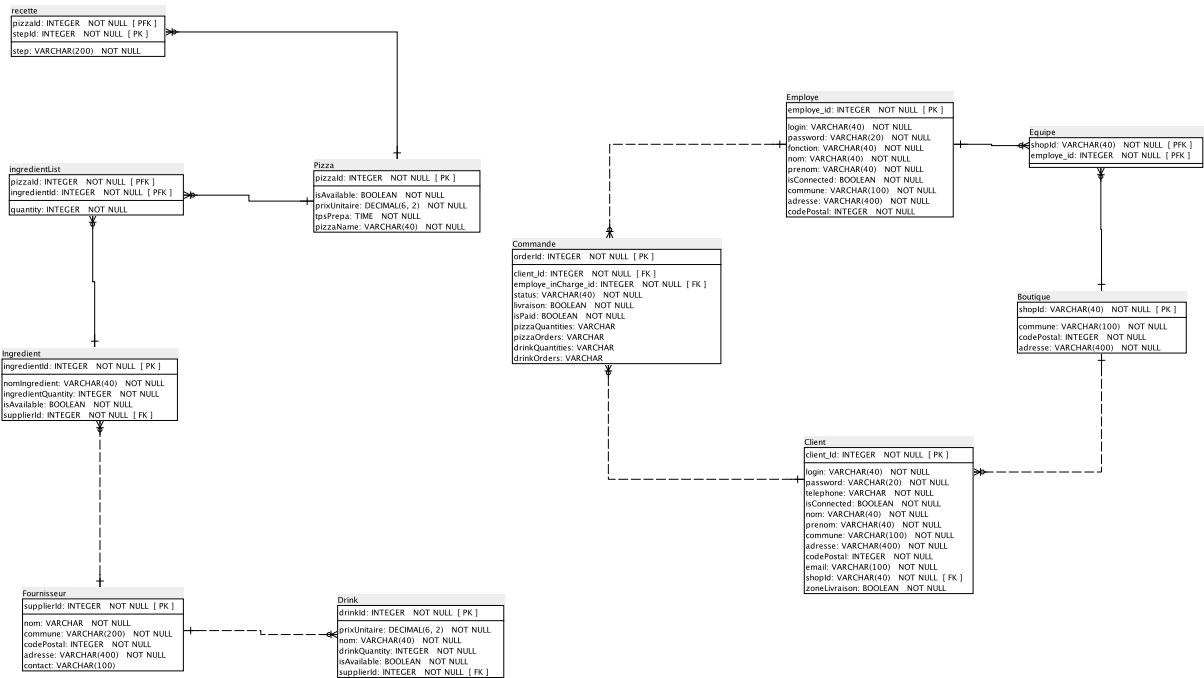


4.1.2 - Diagramme de déploiement complet

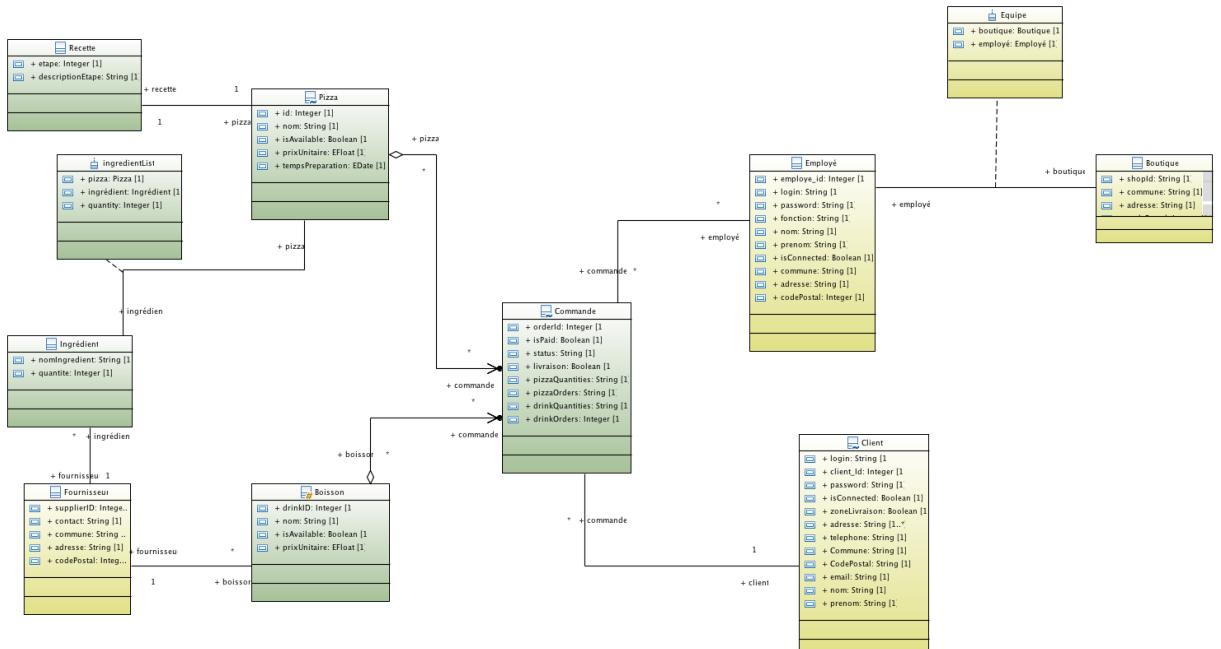


4.2 - Serveur de Base de données

4.2.1 - Modélisation du domaine fonctionnel



4.2.2 - Modélisation du Model physique de données



5 - Architecture logicielle

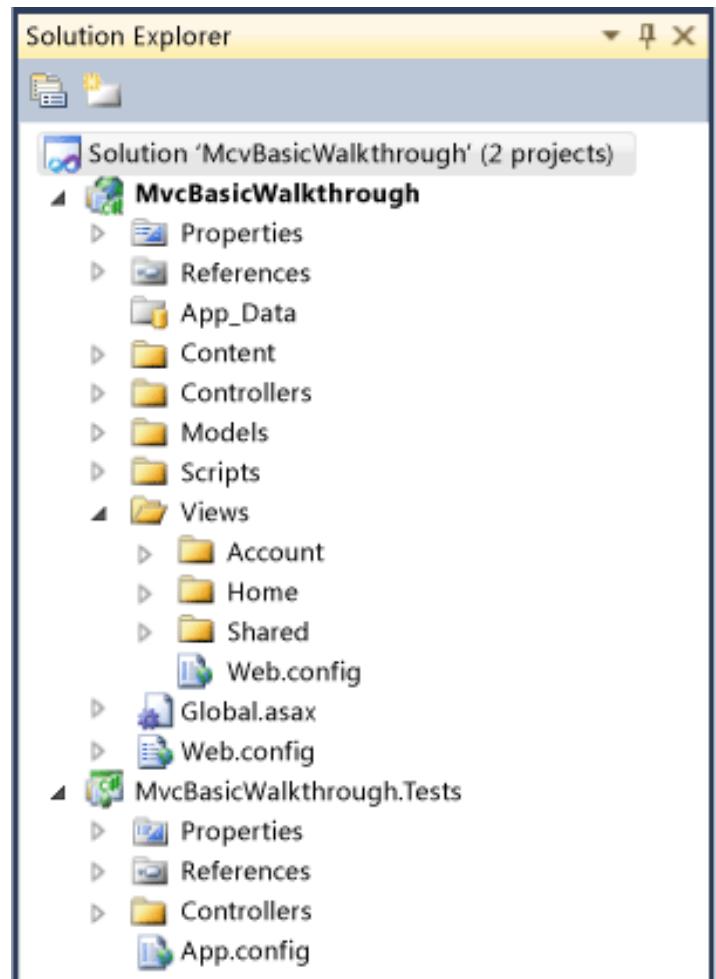
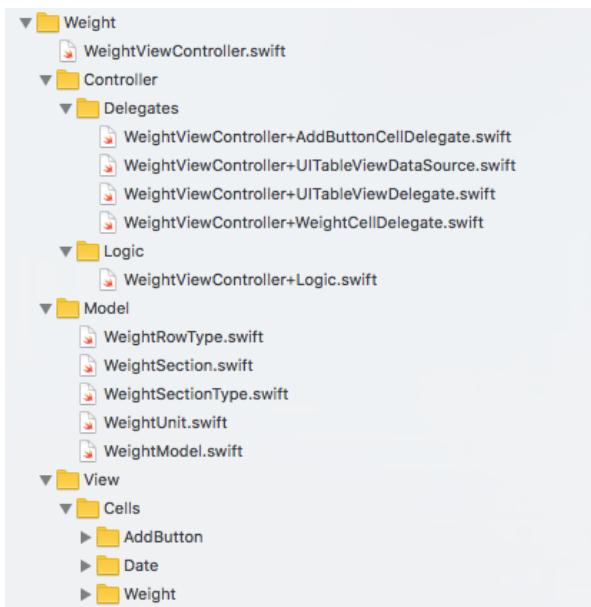
5.1 - Principes généraux

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

5.1.1 - Structure des sources

La structuration des répertoires du projet suit la logique suivante :

- les répertoires sources sont créés de façon à respecter le model MVC



6 - Points particuliers

6.1 - Gestion des logs

La gestion des logs se fait depuis PhpMyAdmin 4.0

Un dump de la base de données sera programmé chaque jour en dehors des horaires d'ouverture

6.2 - Fichiers de configuration

6.2.1.1 - Datasources

Création de la Base données

```

7 -
8 - CREATE SEQUENCE public.fournisseur_supplierid_seq_2_1_2;
9 -
10 - CREATE TABLE public.Fournisseur (
11 -         supplierId INTEGER NOT NULL DEFAULT
12 -             nextval('public.fournisseur_supplierid_seq_2_1_2'),
13 -         nom VARCHAR NOT NULL,
14 -         commune VARCHAR(200) NOT NULL,
15 -         codePostal INTEGER NOT NULL,
16 -         adresse VARCHAR(400) NOT NULL,
17 -         contact VARCHAR(100),
18 -         CONSTRAINT fournisseur_pk PRIMARY KEY (supplierId)
19 - );
20 -
21 - ALTER SEQUENCE public.fournisseur_supplierid_seq_2_1_2 OWNED BY
22 -     public.Fournisseur.supplierId;
23 -
24 - CREATE INDEX fournisseur_idx
25 - ON public.Fournisseur
26 - ( supplierId );
27 -
28 - CREATE SEQUENCE public.ingredient_ingredientid_seq;
29 -
30 - CREATE TABLE public.Ingredient (
31 -         ingredientId INTEGER NOT NULL DEFAULT
32 -             nextval('public.ingredient_ingredientid_seq'),
33 -         nomIngredient VARCHAR(40) NOT NULL,
34 -         ingredientQuantity INTEGER NOT NULL,
35 -         isAvailable BOOLEAN NOT NULL,
36 -         supplierId INTEGER NOT NULL,
37 -         CONSTRAINT ingredient_pk PRIMARY KEY (ingredientId)
38 - );
39 -
40 - ALTER SEQUENCE public.ingredient_ingredientid_seq OWNED BY
41 -     public.Ingredient.ingredientId;

```

```

38 -
39 - CREATE SEQUENCE public.drink_drinkid_seq_1;
40 -
41 - CREATE TABLE public.Drink (
42 -         drinkId INTEGER NOT NULL DEFAULT
43 -             nextval('public.drink_drinkid_seq_1'),
44 -         prixUnitaire NUMERIC(6,2) NOT NULL,
45 -         nom VARCHAR(40) NOT NULL,
46 -         drinkQuantity INTEGER NOT NULL,
47 -         isAvailable BOOLEAN NOT NULL,
48 -         supplierId INTEGER NOT NULL,
49 -         CONSTRAINT drink_pk PRIMARY KEY (drinkId)
50 - );
51 -
52 -
53 - ALTER SEQUENCE public.drink_drinkid_seq_1 OWNED BY public.Drink.drinkId;
54 -
55 - CREATE TABLE public.Pizza (
56 -         pizzaId INTEGER NOT NULL,
57 -         isAvailable BOOLEAN NOT NULL,
58 -         prixUnitaire NUMERIC(6,2) NOT NULL,
59 -         tpsPrepa TIME NOT NULL,
60 -         pizzaName VARCHAR(40) NOT NULL,
61 -         CONSTRAINT pizza_pk PRIMARY KEY (pizzaId)
62 - );
63 -
64 - CREATE TABLE public.ingredientList (
65 -         pizzaId INTEGER NOT NULL,
66 -         ingredientId INTEGER NOT NULL,
67 -         quantity INTEGER NOT NULL,
68 -         CONSTRAINT ingredientlist_pk PRIMARY KEY (pizzaId, ingredientId)
69 - );
70 -
71 - CREATE TABLE public.recette (
72 -         pizzaId INTEGER NOT NULL,
73 -         stepId INTEGER NOT NULL,
74 -         step VARCHAR(200) NOT NULL,
75 -         CONSTRAINT recette_pk PRIMARY KEY (pizzaId, stepId)
76 - );
77 -
78 - CREATE TABLE public.Boutique (
79 -         shopId VARCHAR(40) NOT NULL,
80 -         commune VARCHAR(100) NOT NULL,
81 -         codePostal INTEGER NOT NULL,
82 -         adresse VARCHAR(400) NOT NULL,
83 -         CONSTRAINT boutique_pk PRIMARY KEY (shopId)
84 - );

```

```

83 -
84 - CREATE SEQUENCE public.employe_employe_id_seq;
85 -
86 - CREATE TABLE public.Employe (
87 -     employe_id INTEGER NOT NULL DEFAULT
88 -         nextval('public.employe_employe_id_seq'),
89 -         login VARCHAR(40) NOT NULL,
90 -         password VARCHAR(20) NOT NULL,
91 -         fonction VARCHAR(40) NOT NULL,
92 -         nom VARCHAR(40) NOT NULL,
93 -         prenom VARCHAR(40) NOT NULL,
94 -         isConnected BOOLEAN NOT NULL,
95 -         commune VARCHAR(100) NOT NULL,
96 -         adresse VARCHAR(400) NOT NULL,
97 -         codePostal INTEGER NOT NULL,
98 -         CONSTRAINT employe_pk PRIMARY KEY (employe_id)
99 - );
100 -
101 -
102 - CREATE TABLE public.Equipe (
103 -     shopId VARCHAR(40) NOT NULL,
104 -     employe_id INTEGER NOT NULL,
105 -     CONSTRAINT equipe_pk PRIMARY KEY (shopId, employe_id)
106 - );
107 -
108 - CREATE SEQUENCE public.client_client_id_seq;
109 -
110 - CREATE TABLE public.Client (
111 -     client_Id INTEGER NOT NULL DEFAULT
112 -         nextval('public.client_client_id_seq'),
113 -         login VARCHAR(40) NOT NULL,
114 -         password VARCHAR(20) NOT NULL,
115 -         telephone VARCHAR NOT NULL,
116 -         isConnected BOOLEAN NOT NULL,
117 -         nom VARCHAR(40) NOT NULL,
118 -         prenom VARCHAR(40) NOT NULL,
119 -         commune VARCHAR(100) NOT NULL,
120 -         adresse VARCHAR(400) NOT NULL,
121 -         codePostal INTEGER NOT NULL,
122 -         email VARCHAR(100) NOT NULL,
123 -         shopId VARCHAR(40) NOT NULL,
124 -         zoneLivraison BOOLEAN NOT NULL,
125 -         CONSTRAINT client_pk PRIMARY KEY (client_Id)
126 - );

```

```
127 - ALTER SEQUENCE public.client_client_id_seq OWNED BY public.Client.client_Id;
128 -
129 - CREATE SEQUENCE public.commande_orderid_seq;
130 -
131 - CREATE TABLE public.Commande (
132 -     orderId INTEGER NOT NULL DEFAULT
133 -         nextval('public.commande_orderid_seq'),
134 -     client_Id INTEGER NOT NULL,
135 -     employe_inCharge_id INTEGER NOT NULL,
136 -     status VARCHAR(40) NOT NULL,
137 -     livraison BOOLEAN NOT NULL,
138 -     isPaid BOOLEAN NOT NULL,
139 -     pizzaQuantities VARCHAR,
140 -     pizzaOrders VARCHAR,
141 -     drinkQuantities VARCHAR,
142 -     drinkOrders VARCHAR,
143 -     CONSTRAINT commande_pk PRIMARY KEY (orderId)
144 - );
145 - ALTER SEQUENCE public.commande_orderid_seq OWNED BY public.Commande.orderId;
146 -
147 - ALTER TABLE public.Ingredient ADD CONSTRAINT fournisseur_ingredient_fk
148 - FOREIGN KEY (supplierId)
149 - REFERENCES public.Fournisseur (supplierId)
150 - ON DELETE NO ACTION
151 - ON UPDATE NO ACTION
152 - NOT DEFERRABLE;
153 -
154 - ALTER TABLE public.Drink ADD CONSTRAINT fournisseur_drink_fk
155 - FOREIGN KEY (supplierId)
156 - REFERENCES public.Fournisseur (supplierId)
157 - ON DELETE NO ACTION
158 - ON UPDATE NO ACTION
159 - NOT DEFERRABLE;
160 -
161 - ALTER TABLE public.ingredientList ADD CONSTRAINT ingredient_ingredientlist_fk
162 - FOREIGN KEY (ingredientId)
163 - REFERENCES public.Ingredient (ingredientId)
164 - ON DELETE NO ACTION
165 - ON UPDATE NO ACTION
166 - NOT DEFERRABLE;
167 -
168 - ALTER TABLE public.recette ADD CONSTRAINT pizza_recette_fk
169 - FOREIGN KEY (pizzaId)
170 - REFERENCES public.Pizza (pizzaId)
171 - ON DELETE NO ACTION
```

```
172 - ON UPDATE NO ACTION
173 - NOT DEFERRABLE;
174 -
175 - ALTER TABLE public.ingredientList ADD CONSTRAINT pizza_ingredientlist_fk
176 - FOREIGN KEY (pizzaId)
177 - REFERENCES public.Pizza (pizzaId)
178 - ON DELETE NO ACTION
179 - ON UPDATE NO ACTION
180 - NOT DEFERRABLE;
181 -
182 - ALTER TABLE public.Equipe ADD CONSTRAINT boutique_equipe_fk
183 - FOREIGN KEY (shopId)
184 - REFERENCES public.Boutique (shopId)
185 - ON DELETE NO ACTION
186 - ON UPDATE NO ACTION
187 - NOT DEFERRABLE;
188 -
189 - ALTER TABLE public.Client ADD CONSTRAINT boutique_client_fk
190 - FOREIGN KEY (shopId)
191 - REFERENCES public.Boutique (shopId)
192 - ON DELETE NO ACTION
193 - ON UPDATE NO ACTION
194 - NOT DEFERRABLE;
195 -
196 - ALTER TABLE public.Equipe ADD CONSTRAINT employe_equipe_fk
197 - FOREIGN KEY (employe_id)
198 - REFERENCES public.Employe (employe_id)
199 - ON DELETE NO ACTION
200 - ON UPDATE NO ACTION
201 - NOT DEFERRABLE;
202 -
203 - ALTER TABLE public.Commande ADD CONSTRAINT employe_commande_fk
204 - FOREIGN KEY (employe_inCharge_id)
205 - REFERENCES public.Employe (employe_id)
206 - ON DELETE NO ACTION
207 - ON UPDATE NO ACTION
208 - NOT DEFERRABLE;
209 -
210 - ALTER TABLE public.Commande ADD CONSTRAINT client_commande_fk
211 - FOREIGN KEY (client_Id)
212 - REFERENCES public.Client (client_Id)
213 - ON DELETE NO ACTION
214 - ON UPDATE NO ACTION
215 - NOT DEFERRABLE;
216 -
217 -
```

217.1 - Ressources

Exemple de requête SQL avec jeu de données test

```

INSERT INTO fournisseur
(supplierId, nom, commune, codepostal, adresse, contact)
VALUES
(1, 'SuperLegumes',      'Rungis',    77000,  '4 allée des platanes', 'Paul Martin'),
(2, 'France Boisson',    'Paris',     75018,  '45 boulevard Ney',     'Georges Dupont'),
(3, 'Viva Italia',       'Meudon',    92000,  '12 rue de la cote',   'Marc Durant'),
(4, 'Le boucher',        'Paris',     75004,  '24 rue Saint Antoine', 'Grégoire Labille'),
(5, 'Fromager',          'Paris',     75008,  '4 rue Castex',        'Olivier Fau'),
(6, 'Moulin du chat',    'Paris',     75014,  '2 rue des graviers',  'paul le boulanger');

INSERT INTO ingredient
(ingredientId, nomingredient, ingredientquantity, isavailable, supplierid)
VALUES
(1,  'tomate',           200,    true,   1),
(2,  'farine',            500,    true,   6),
(3,  'mozzarella',        200,    true,   3),
(4,  'gruyère',           200,    true,   5),
(5,  'gorgonzola',        200,    true,   3),
(6,  'jambon',             200,    true,   4),
(7,  'jambon sec',         100,    true,   4),
(8,  'viande hachée',      100,    true,   4),
(9,  'parmesan',           200,    true,   5),
(10, 'pepperoni',          100,    true,   3),
(11, 'roquette',            150,    true,   1),
(12, 'champignon',         200,    true,   1);

INSERT INTO pizza
(pizzaId, PizzaName,  isavailable, prixunitaire, tpsPrepa)
VALUES
(1, 'Reine',           true,   10.90,  '00:20:00'),
(2, 'Margarita',        true,   10.90,  '00:20:00'),
(3, 'Orientale',         true,   12.90,  '00:20:00'),
(4, 'Pepperoni',         true,   12.90,  '00:20:00'),
(5, 'Chèvre Miel',       true,   12.90,  '00:20:00'),
(6, 'Veggie',             true,   14.90,  '00:20:00'),
(7, '4 fromages',        true,   14.90,  '00:20:00'),

```

```

(8,'Saumon',           true, 14.90,  '00:20:00'),
(9,'Hawaienne',        true, 14.90,  '00:20:00');

INSERT INTO ingredientlist
(ingredientid, quantity, pizzaId)
VALUES
(1, 3, 2),
(2, 5, 2),
(3, 3, 2),
(1, 3, 1),
(2, 5, 1),
(3, 3, 1),
(6, 1, 1),
(12,2, 1),
(1, 3, 3),
(2, 5, 3),
(3, 3, 3),
(7, 2, 3),
(10,3, 3);

INSERT INTO recette (pizzaId,stepid, step)
VALUES
(1,1, 'Faire la pate'),
(1,2, 'Etaler le coulis de tomate'),
(1,3, 'Mettre 50 grammes de mozzarella'),
(1,4, 'Déposer des copeaux de jambon'),
(1,5, 'Déposer une poignée de champignons'),
(1,6, 'Mettre au four à 220°C / 10 minutes'),
(1,7, 'Saupoudrer d Origan'),
(2,1, 'Faire la pate'),
(2,2, 'Etaler le coulis de tomate'),
(2,3, 'Mettre 50 grammes de mozzarella'),
(2,4, 'Mettre au four à 220°C / 10 minutes'),
(2,5, 'Saupoudrer d Origan'),
(3,1, 'Faire la pate'),
(3,2, 'Etaler le coulis de tomate'),
(3,3, 'Mettre 50 grammes de mozzarella'),
(3,4, 'Déposer des copeaux de jambon'),
(3,5, 'Déposer des tranches de Chorizo'),
(3,6, 'Mettre au four à 220°C / 10 minutes'),
(3,7, 'Saupoudrer d Origan');

INSERT INTO drink
(drinkId,prixunitaire, nom, drinkquantity, isavailable, supplierid)
VALUES
(1,1.50,   'Coca Cola 33cl',      200,    true,    2),

```

```

(2,1.50,      'Coca Cola Zero 33cl',  200,      true,    2),
(3, 1.50,     'Coca Cola Light 33cl', 200,      true,    2),
(4,1.50,      'Sprite 33 cl',        200,      true,    2),
(5, 1.50,     'Nestea',            200,      true,    2),
(6, 2.70,     'Heineken 25cl',       200,      true,    2),
(7,3.50,      'Desperados 33cl',    200,      true,    2),
(8,2.70,      'Vittel 50cl',         200,      true,    2),
(9,4.00,      'San Pellegrino 25cl', 200,      true,    2);

INSERT INTO Boutique
(shopId,commune,adresse,codePostal)
VALUES
(1,'Paris','58 rue de Belleville', 75020),
(2,'Paris','113 rue Cambronne',    75015),
(3,'Paris','23 rue Claude Tillier', 75012),
(4,'Paris','47 rue Gay Lussac',    75005),
(5,'Paris','98 rue Saussure',     75017);

INSERT INTO Client
(client_Id,login, password, isconnected, nom, prenom, telephone, commune, adresse,
codepostal, email, shopid, zonelivraison)
VALUES
(1,'client1',  'eiffel',   true,    'Picasso', 'Pablo',  '01.45.45.45.45',   'Paris',
'1 rue des cubistes',      75001,'ppicasso@gmail.com', 1,  true),
(2,'client2',  'foot',      true,    'Tahar',    'Edmond', '01.45.45.45.45','Paris',
'30 boulevard Malsherbes', 75017,'etahar@gmail.com', 2,  true),
(3, 'client3', 'basket',   true,    'Bernard',  'Romain', '01.45.45.45.45','Paris',
'14 rue Valadon',          75007,'rbernard@gmail.com', 3,  true),
(4, 'client4', 'bowling',  true,    'Boissier', 'Clara',  '01.45.45.45.45','Paris',
'60 rue marcadet',          75018,'mboissier@gmail.com', 2,  false),
(5, 'client5', 'ecume',    false,   'Vian',     'Boris',  '01.45.45.45.45','Paris',
'45 rue de la plaine',     75013,'bvian@gmail.com', 3,  true),
(6, 'client6', 'sabre',    true,    'De la
Vega', 'Diego',  '01.45.45.45.45','Paris',  '50 boulevard du
cheval', 75019,'zorro@gmail.com', 3,  true),
(7, 'client7', 'petitdoux',true,    'Gates',    'Bill',    '01.45.45.45.45','Neuilly',
'78 rue des platanes',     92000,'bgates@gmail.com', 1,  true);

INSERT INTO employe
(employe_id, login, password, fonction, nom, prenom, isconnected, commune, adresse,
codepostal)
VALUES
(0, 'noOne',      'no',      'no',      'noOne',      'noOne',      true,    'noW
here', 'noWhere',      0),
(1, 'pdubois',    'test',    'livreur',  'Dubois',    'Paul',      true,    ,
'Paris', '37 rue de Sèvres', 75006),

```

<LogoClient>

<LogoEntreprise>

```
(2, 'onguyen',      'fleur',      'commercial',    'Nguyen',     'Océane',        true ,  
'Paris',   '350 rue Lecourbes' , 75015),  
(3, 'cbourgeois',  'scooter',    'cuisinier',     'Bourgeois','Christophe',  true , '  
Paris' ,   '40 rue des Thermopyles', 75014),  
(4, 'fbisson',     'jeuVideo',   'livreur',      'Bisson',     'François',       true , '  
Paris' ,   ' 9 rue Maison Dieu',    75019),  
(5, 'mmoercant',   'afteffe',   'commercial',   'Moercant',   'Mikael',        true , '  
Asnieres', '12 rue Salengro',     92000),  
(6, 'bbauwens',    'oie',       'cuisinier',    'Bauwens',    'Béatrice',       true , '  
'Béatrice',  true , 'Paris',     '35 rue d Avron',    75020),  
(7, 'hallart',     'abeille',    'livreur',      'Allart',     'Hugues',        true , '  
Ermont',   '12 avenue des marroniers', 95230),  
(8, 'jroig',       'moto',      'cuisinier',    'Roig',       'Jean',          true , '  
Versailles', '1 boulevard Royal',    78000),  
(9, 'ggaillard',   'geek',      'livreur',      'Gaillard',   'Gilles',        true , '  
Paris',     '141 rue de Charonne',    75003);
```

```
INSERT INTO equipe  
(shopid, employe_id)  
VALUES  
(1, 1),  
(1, 2),  
(1, 3),  
(2, 4),  
(2, 5),  
(2, 6),  
(3, 7),  
(3, 8),  
(3, 9);
```

```
INSERT INTO Commande  
(client_Id,livraison, isPaid,pizzaQuantities,pizzaOrders,drinkQuantities,  
drinkOrders,employe_inCharge_id,status)  
VALUES  
(1, true, true, '1', '1', 1,8,1,'Livraison'),  
(4, true, true, '3', '3', 2,1,3,'Préparation'),  
(6, true, false, '1', '5', 4,1,6,'Préparation'),  
(2, true, true, '1', '2', 4,1,2,'Confirmée');
```

```
INSERT INTO Commande  
(client_Id,livraison, isPaid,pizzaQuantities,pizzaOrders,drinkQuantities,  
drinkOrders,employe_inCharge_id,status)  
VALUES  
(1, true, false, '1', '1', 1,8,0,'Livrée'),
```

```
(1, true, true, '3', '3', 2,1,0,'Attente'),
(1, true, true, '1', '5', 4,1,0,'Attente'),
(1, true, false, '1', '2', 4,1,0,'Livrée');

INSERT INTO Commande
(client_Id,livraison, isPaid,pizzaQuantities,pizzaOrders,employe_inCharge_id,status)
VALUES
(7, true, true, '3', '1', 1,'Livraison');

INSERT INTO Commande
(client_Id,livraison, isPaid,pizzaQuantities,pizzaOrders,drinkQuantities,
drinkOrders,employe_inCharge_id,status)
VALUES
(1, true, true, '1,2', '1, 3', '1',8,1,'Livraison'),
(4, true, true, '3,1,2', '3, 2, 1', '2',1,3,'Préparation'),
(6, true, false, '1,5', '5,', '4',1,6,'Préparation');
```

217.2 - Environnement de développement

Environnement de développement iOS

- Logiciel IDE : Xcode 9
- Langage : swift 4.0
- Injection framework : CocoaPods/Carthage
- Framework : Alamofire, Swifty ,Json, GoogleMap Rest API

Environnement de développement Android

- Logiciel IDE: Android Studio
- Langage : js/xml
- Framework : JQuery, ReactNative, GoogleMap Rest API

Environnement de développement Android

- Logiciel IDE: Sublime Text/Atom
- Langage : js/xml/php/html
- Framework : JQuery, ReactNative, GoogleMap Rest API

Environnement de développement SQL

- Logiciel IDE: Sublime Text/Atom
- JDBC : PhpMyAdmin 4.0
- Langage : SQL

<LogoClient>

<LogoEntreprise>

218 - Glossaire