



OC Pizza

Système de Gestion de Pizzeria

Dossier de conception technique

Version 1.3

Auteur

Lionel JOFFRAY

Analyste Programmeur



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 - Site web	5
3.2 - Application OC Pizza Insider	5
3.3 - Diagramme de Composants.....	6
3.3.1 - Descriptions des Composants	6
4 - Architecture de Déploiement.....	7
4.1 - Descriptions des nœuds	7
4.1.1 - Appareil fixe ou mobile.....	7
4.1.2 - Appareil mobile Android.....	8
4.1.3 - Apache server.....	8
4.1.4 - Serveur BDD.....	8
4.1.5 - Google API.....	8
4.1.6 - Paypal API.....	8
5 - Architecture logicielle	9
5.1 - Principes généraux.....	9
5.1.1 - Les couches.....	9
5.1.2 - Structure des sources Wordpress.....	9
5.1.3 - Structure des sources Application OC Pizza Insider	10
6 - Points particuliers.....	11
6.1 - Gestion des logs.....	11
6.2 - Environnement de développement	13
6.3 - Procédure de packaging / livraison.....	13
6.4 - Ressources	13
6.4.1 - API Google Maps	13
6.4.2 - API Paypal.....	13
6.4.3 - Images	13
7 - Glossaire	14



1 - VERSIONS

Auteur	Date	Description	Version
Lionel J	02/10/2019	Création du document	1.0
Lionel J	14/10/2019	Mise à jour de contenu	1.1
Lionel J	21/10/2019	Finalisation du document	1.2
Lionel J	30/10/2019	Mise à jour	1.3



2 - INTRODUCTION

2.1 - Objet du document

Le présent document constitue le dossier de conception technique du site web et de l'application **OC Pizza Insider**.

L'objectif du document est d'informer et aider les développeurs pour la conception, la maintenance et l'évolution de l'application.

Les éléments du présent dossier découlent :

- Du domaine fonctionnel
- Des spécifications techniques
- Du modèle physique de donnée

2.2 - Références

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

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



3 - ARCHITECTURE TECHNIQUE

Pour le projet, nous avons donc la partie **Website**, et la partie **Application**.

3.1 - Site web

- Le **website** sera réalisé grâce à un **CMS** puis héberger sur un serveur **OVH**. Ce procédé nous évite de devoir développer un site « *from scratch* », c'est-à-dire de zéro en codant tout un site web en *HTML*, *CSS* et *Javascript*.

Un **CMS** facilite grandement la gestion et l'organisation d'un site web.

- L'utilisation d'**API** externe sera faite concernant la localisation des utilisateurs, avec **Google Places**, et pour la vérification des paiement internet, avec l'api **Paypal**.

Une fois en ligne, la procédure de gestion du site est décrite sur ce lien :

<https://anthemcreation.com/tutoriels/administration-de-wordpress/>

3.2 - Application OC Pizza Insider

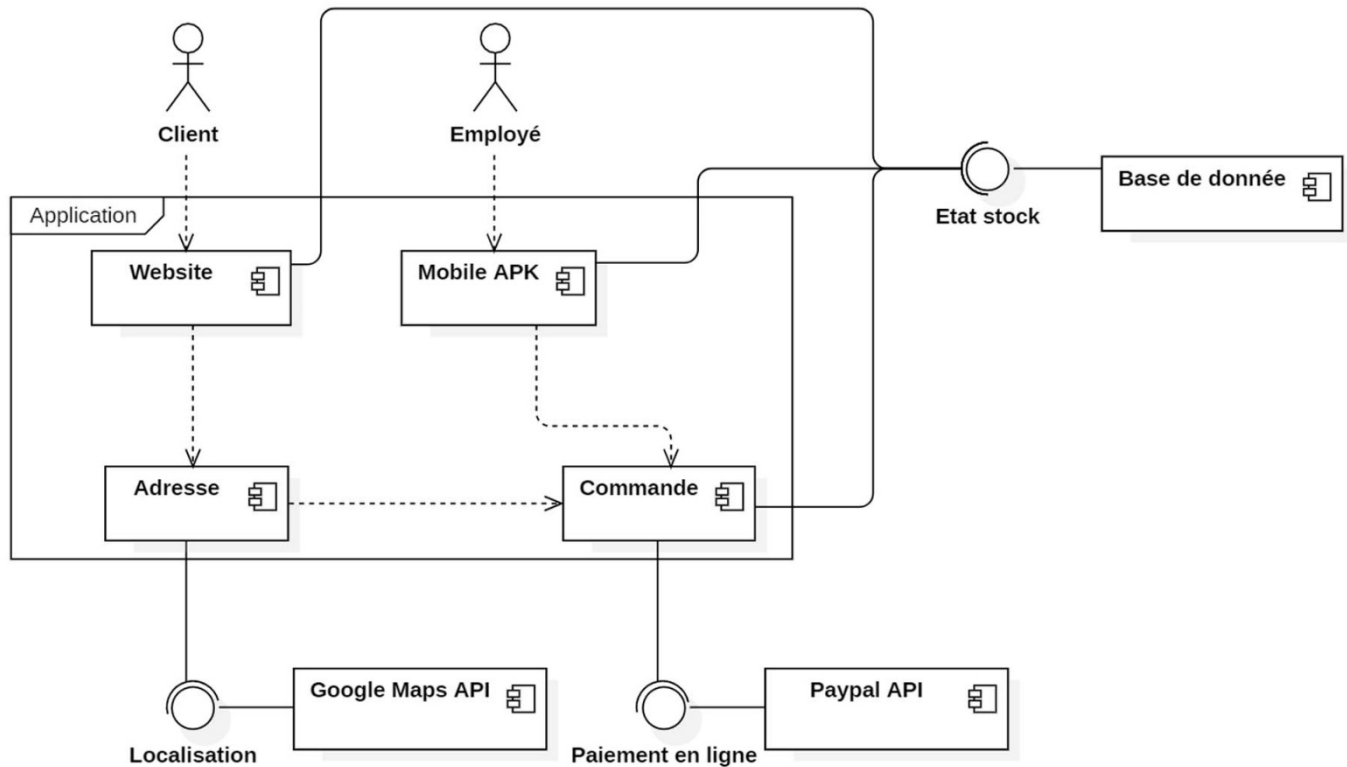
- L'**application** sera codée en **Java**, et réaliser sur un **IDE** comme Android Studio.

Le langage de programmation *Java* (possible portage en **Kotlin**), est tout désigné pour l'application Smartphone/Tablette. L'application communiquera avec la base de données **PostgreSQL**, qui elle sera hébergée en ligne et disponible pour le site web et l'application.

- Site internet **WordPress.org**
- Hébergeur **OVH**
- Serveur Web (**Apache 2.4.41**)
- Base de données (**PostgreSQL 11.5**)
- API **Paypal** (Checkout API)
- API **Google Maps** (Places API)



3.3 - Diagramme de Composants



3.3.1 - Descriptions des Composants

Le client va via le **site internet**, **cr er son compte**, qui sera ensuite stocker dans la **BDD** (PostgreSQL).

Il rentre pour pouvoir **commander**, une **adresse** valide qui sera ensuite localiser via l'**API externe Google Maps** et son interface offerte, pour soumettre au client la **pizzeria** la plus proche de chez lui et lui indiquer si la livraison est possible ou non, en comparant la **distance de livraison** **stocker dans la BDD**, sur les adresses des **pizzerias** qui sont d j  rentr es.

Une fois l'adresse valid e et la pizzeria choisie, le client **valide le panier** souhait . S'il choisit de **r gler** sa commande imm diatement via le site, l'interface de l'**API externe Paypal** entre en jeu et permet de **payer sa commande en ligne**.

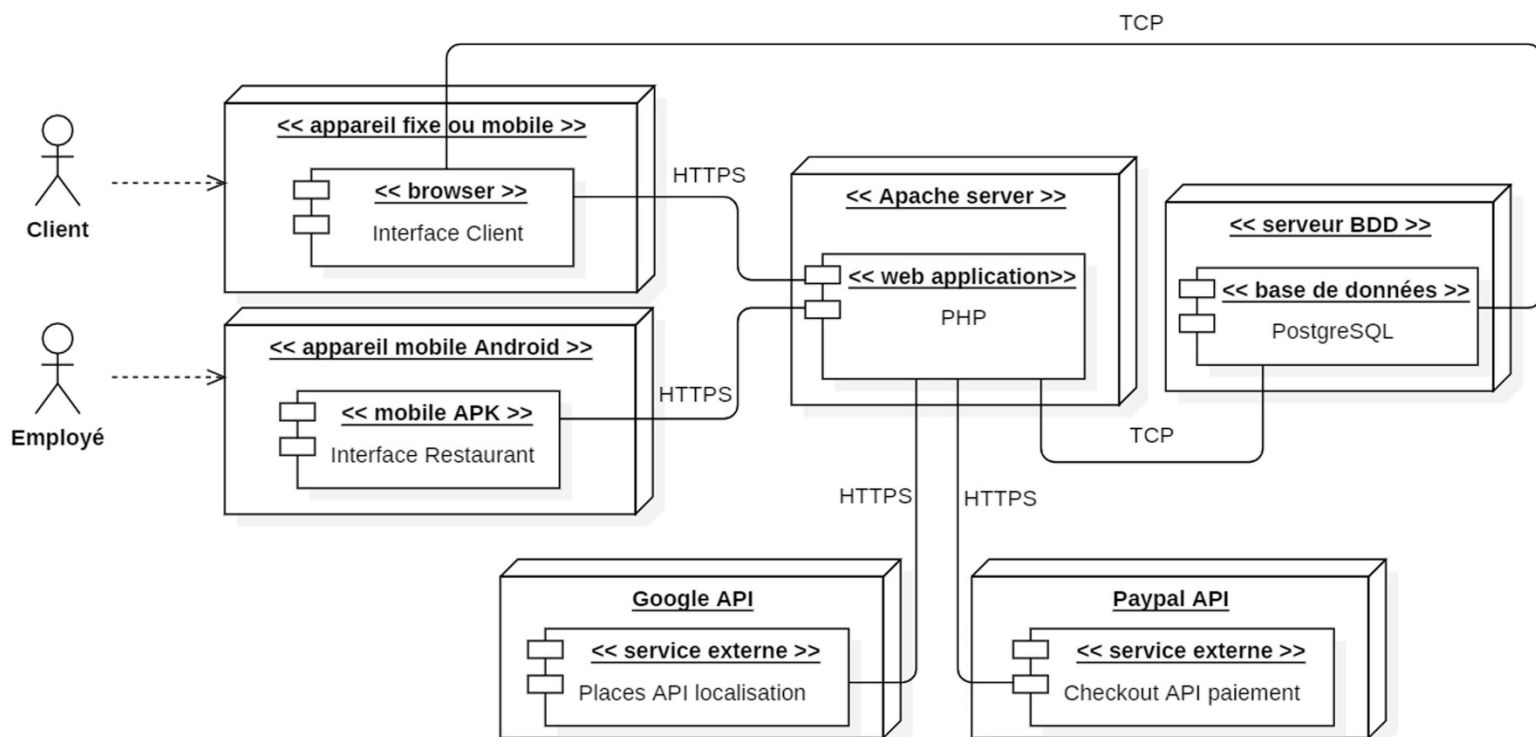
Une fois la commande lanc e, les **stocks** sont mis   jour via l'interface de la **BDD PostgreSQL** avec les **ingr dients retir s** qui sont n cessaires   la **pr paration** de la commande.

  tout moment, les **employ s** connecter   l'**APK Android (JAVA)** peuvent interroger la **BDD** pour le **stock** ou informations concernant une **commande** de leur pizzeria.



4 - ARCHITECTURE DE DEPLOIEMENT

Diagramme de Déploiement



4.1 - Descriptions des nœuds

4.1.1 - Appareil fixe ou mobile

Le nœud appareils fixe fait référence à une connexion via un **navigateur web** (Browser). Les clients qui se connectent au **site internet** utilisent des **ordinateurs fixe**, **ordinateurs portables**, **tablettes** et **smartphones**, la communication se fait en **HTTPS** entre le client et l'application et en **TCP** vers la BDD.



4.1.2 - Appareil mobile Android

Le nœud appareils mobile Android fait références à une connexion via un l'**application Android**. Les employés qui se connectent à la **mobile APK** (application, codé en **JAVA**) utilisent des **tablettes** et **smartphones**, la communication se fait en **HTTPS** entre le client et l'application et en **TCP** vers la BDD.

4.1.3 - Apache server

Le nœud Apache server est le **serveur web** du projet.

Apache est un serveur http, c'est un serveur qui gère exclusivement des requêtes HTTP. Il a pour rôle d'intercepter les requêtes Http, sur un port qui est par défaut 80, pour les traiter et générer ensuite des réponses Http.

Son travail de base est d'accepter les requêtes des clients (p. ex. le navigateur Web d'un visiteur) et d'envoyer la réponse à cette requête (p. ex. les composantes de la page qu'un visiteur souhaite voir).

La version utilisée sera la dernière en date, **Apache 2.4.41**.

4.1.4 - Serveur BDD

Le nœud serveur BDD contient la **Base de données relationnel**.

Le chemin de communication **TCP** permet à l'application et au site internet de communiquer avec la base de données (**Serveur BDD**) qui est situé sur un autre matériel, externe.

Ce serveur héberge la Base de données, sur OVH, et tourne sous **PostgreSQL 11.5**, version actuelle.

4.1.5 - Google API

Le nœud Google API est le service fournis par Google.

Les adresses sont envoyées à l'**API Places** pour permettre de localiser une adresse par rapport à un restaurant et inversement.

4.1.6 - Paypal API

Le nœud Paypal API est le service fournis par Paypal.

Ici Paypal fourni une **Rest API**, qui permet le règlement d'une commande par un client, et ce en toute sécurité sur internet.

La version utilisée sera l'**API Checkout**.



5 - ARCHITECTURE LOGICIELLE

5.1 - Principes généraux

Les sources et versions du projet sont gérées par **Git**.

5.1.1 - Les couches

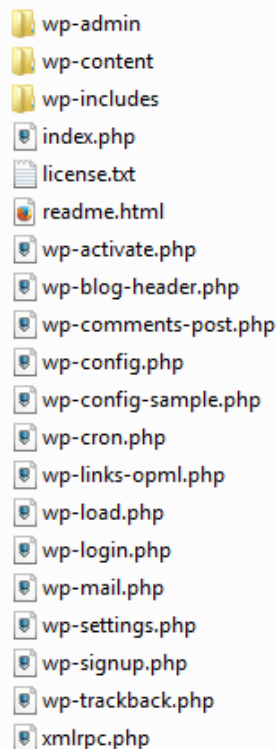
L'architecture **WordPress** de base reste inchangée.

L'architecture de l'application **Java** est dite **MVC** (Modèle Vue Contrôleur en Français) :

- Une couche **Model** : ensemble des données utilisées par l'application
- Une couche **View** : responsable de la logique vue du composant
- Une couche **Controler** : fonction permettant l'échange de données entre la vue et les données

5.1.2 - Structure des sources Wordpress

Structure Wordpress



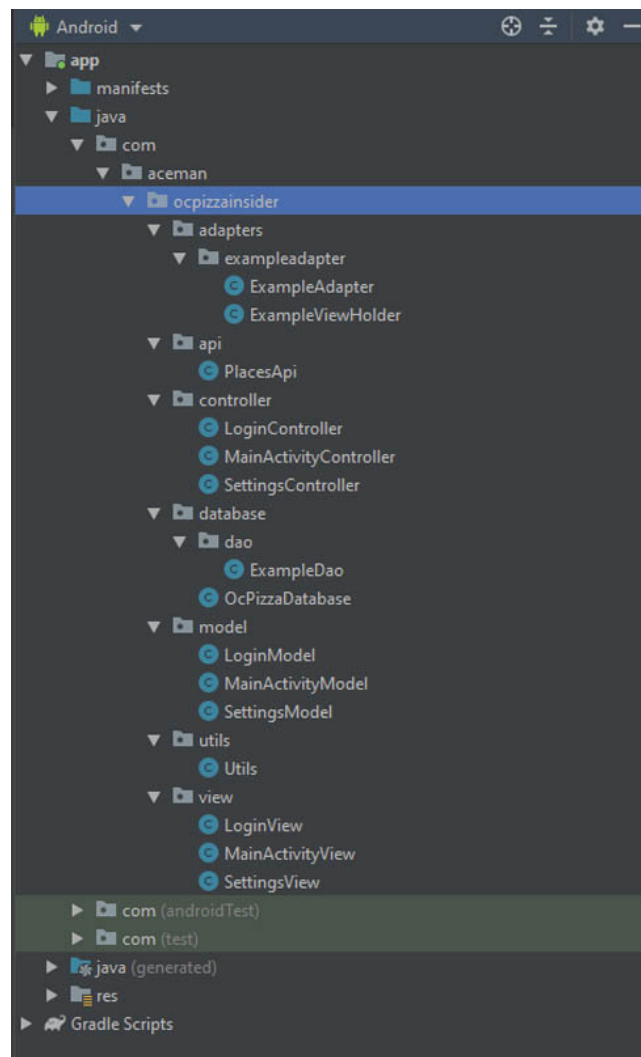
A screenshot of a file explorer window showing the directory structure of a WordPress installation. The files are listed in a vertical column, each with a small icon representing its type (folder or file). The files include:

- wp-admin (folder)
- wp-content (folder)
- wp-includes (folder)
- index.php (file)
- license.txt (file)
- readme.html (file)
- wp-activate.php (file)
- wp-blog-header.php (file)
- wp-comments-post.php (file)
- wp-config.php (file)
- wp-config-sample.php (file)
- wp-cron.php (file)
- wp-links-opml.php (file)
- wp-load.php (file)
- wp-login.php (file)
- wp-mail.php (file)
- wp-settings.php (file)
- wp-signup.php (file)
- wp-trackback.php (file)
- xmlrpc.php (file)



5.1.3 - Structure des sources Application OC Pizza Insider

Exemple de structure





6 - POINTS PARTICULIERS

6.1 - Gestion des logs

Pour accéder aux statistiques du site, aller sur :


<https://logs.ovh.net/ocpizza>

La connexion se fait via l'identifiant OVH et le mot de passe associé.

Cliquez sur le lien généré automatiquement dans le **Manager** pour accéder aux statistiques et aux logs.

Il faudra s'identifier avec la référence client (Nic-handle) et le mot de passe. Une alerte par mail en cas d'erreur du système sera envoyée directement sur l'adresse de l'équipe technique paramétrée sur le site.

Aperçu « votre espace statistiques »




NUMERO 1 DE L'HEBERGEMENT EN FRANCE

MANAGER WEBMAIL
SUPPORTS GUIDES


Domaines SSL Emails Hébergement RPS Serveurs dédiés Housing VS Téléphonie

Bienvenue sur votre espace statistiques




Consultez les statistiques détaillées de fréquentation de votre site web grâce à Awstats (en temps réel). Contient uniquement les logs journalisés

[Cliquez ici pour voir les statistiques calculées avec Awstats](#)



A compter du 15 avril 2010, les statistiques urchin v3 ne sont plus calculées. Les [anciennes statistiques](#) demeurent cependant disponibles. Veuillez utiliser [urchin v6](#).




Consultez les statistiques détaillées de fréquentation de votre site web grâce à URCHIN v6.

[Anciennes statistiques](#) (avant août 2011)
[Cliquez ici pour voir les statistiques calculées avec Urchin v6 \(EULA\)](#)
Choisissez la langue d'urchin v6

OVH Speed Log

OSL vous permet de consulter les logs bruts en temps réel (un décalage d'environ 5 minutes).

Cliquez ici pour voir les logs bruts en temps réel [web](#) - ftp - error - cgi - out - ssh - cron



Vous avez la possibilité de visualiser les logs bruts générés par nos serveurs en cliquant sur les mois ci dessous :

OC Développement
ocDevTeam74.fr

51 Route de Frize, 74140 Messery – 04 50 00 74 00 – OCDev74@gmail.com
S.A.R.L. au capital de 1 000,00 € enregistrée au RCS de 123456789 – SIREN 999 999 999 – Code APE : 6202A



Urchin v6

Ces statistiques donnent des renseignements sur le trafic du site

- Le nombre de visiteurs,
- Le nombre de pages visualisées,
- Le "poids" des pages visualisées,
- Le nombre de requêtes http.
- Les durées moyennes de connexion à l'ensemble de du site ou une page particulière
- Comment les visiteurs du site l'ont-ils connu ?
- Par quels moteurs de recherche ont-ils trouvé l'URL de du site ?
- Quels mots-clés ont-ils utilisé lors de leur recherche ?

Logs bruts

Il est possible de visualiser les logs du site pratiquement en direct en moins de 15 minutes, ce qui permet de vérifier le bon fonctionnement du site presque en temps réel.

Différents types de logs sont à disposition :

- Logs Web : trouvez ici les différents logs de consultation du site, ainsi que les différentes actions réalisées à partir du site. Cela permet par exemple de repérer des tentatives de hacks.
- Logs FTP : les différentes connexions FTP seront enregistrées et conservées dans ces logs.
- Logs erreur : les différentes erreurs générées par le site.
- Logs CGI : les différents appels aux scripts cgi.bin qui ont été réalisés.
- Logs out : les statistiques de l'hébergement sur les différents appels externes réalisés.
- Logs SSH : ces logs indiquent les différentes connexions réalisées avec le protocole SSH.
- Logs cron : le résultat de l'exécution des tâches planifiées



6.2 - Environnement de développement

- Le site **Wordpress.org** pour obtenir Wordpress 5.2.3.
- L'IDE **Android Studio** pour l'application.

6.3 - Procédure de packaging / livraison

La solution fera l'objet d'un déploiement sur le serveur OVH au moment de la livraison finale. Les identifiants et mots de passe seront donnés au client après la mise en ligne et le paiement de ce dernier.

6.4 - Ressources

6.4.1 - API Google Maps

Pour intégrer l'API Google Maps, se référer à la documentation officielle au lien suivant :
<https://developers.google.com/maps/documentation/directions/start?hl=fr>

6.4.2 - API Paypal

Pour intégrer l'API Paypal, se référer à la documentation officielle au lien suivant :
<https://developer.paypal.com/docs/api/overview/>

6.4.3 - Images

Le client a fourni les images qu'il souhaite pour personnaliser le site web et/ou l'application mobile.



7 - GLOSSAIRE

API	interface de programmation d'application (souvent désignée par le terme API pour <i>application programming interface</i>) est un ensemble normalisé de classes, de méthodes, de fonctions et de constantes qui sert de façade par laquelle un logiciel offre des services à d'autres logiciels.
CMS	<i>Content Management System</i> , cela désigne une famille d'applications qui ont pour but de créer et mettre à jours facilement un site web dynamique.
Serveur web	C'est un serveur qui gère exclusivement des requêtes HTTP . Il a pour rôle d'intercepter les requêtes Http, sur un port qui est par défaut 80, pour les traiter et générer ensuite des réponses Http.
Git	Git est un logiciel de gestion de versions décentralisé.
IDE	Un environnement de développement est un ensemble d'outils qui permet d'augmenter la productivité des programmeurs qui développent des logiciels.
MVC	Modèle-vue-contrôleur ou MVC est un motif d'architecture logicielle destiné aux interfaces graphiques lancé en 1978 et très populaire pour les applications web. Le motif est composé de trois types de modules ayant trois responsabilités différentes : les modèles, les vues et les contrôleurs.