

OC PIZZA

Vente de pizza en ligne

Dossier de conception technique

Version 1.0

Auteur

KIEFER Arnaud

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 - Application Web	5
3.1.1 - Package Customer Interface	6
3.1.2 - Package Store Interface	6
3.2 - Base de données	8
3.2.1 - SGBD utilisé	8
3.2.2 - Modèle Physique de données	8
4 - Architecture de Déploiement	10
4.1 - Serveur de Base de données	11
4.2 - Serveur	11
5 - Glossaire	12

1 - VERSIONS

Auteur	Date	Description	Version
Arnaud KIEFER	13/01/2021	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. Il est destiné à l'attention des développeurs, mainteneurs et de l'équipe technique d'OC PIZZA.

L'objectif de l'analyse technique étant de lister les contraintes spécifiques dont les développeurs vont devoir tenir compte pour coder l'application, le présent document présentera les langages et les conventions de développement, les conventions de développement, l'architecture logicielle et de déploiement.

Les éléments du présent dossier découlent :

- De l'analyse des besoins d'OC PIZZA,
- De la rédaction du dossier de conception fonctionnelle.

2.2 - Références

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

1. **DCF – OCPIZZA** : Dossier de conception fonctionnel de l'application
2. **DCE – OCPIZZA** : Dossier d'exploitation de l'application

3 - ARCHITECTURE TECHNIQUE

3.1 - Application Web

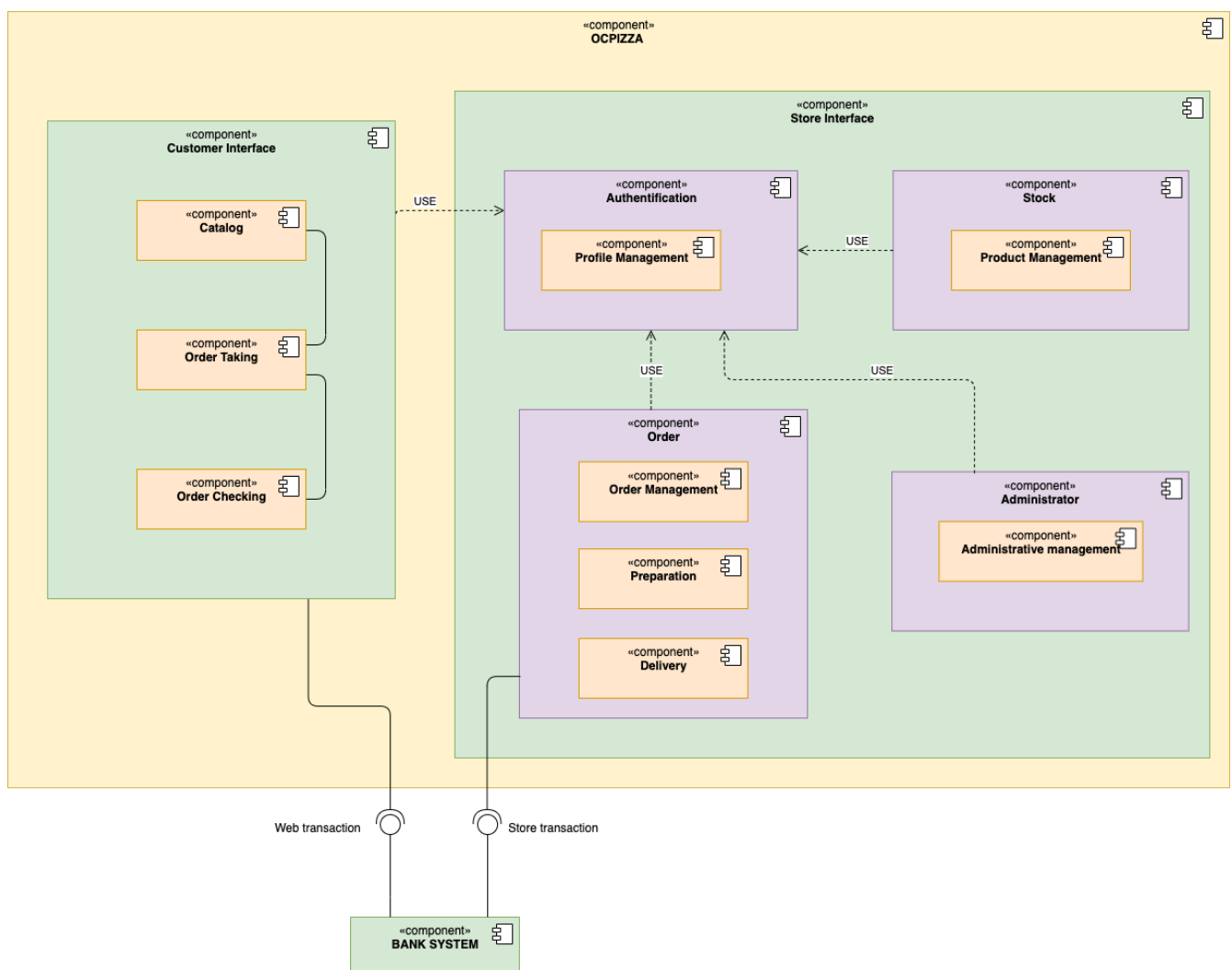


Diagramme UML de Composants

3.1.1 - Package Customer Interface

Il regroupe toute l'interface et les fonctions auxquelles le client aura accès. Il comprend les différents composants suivants.

3.1.1.1 - Composant Catalog

Le composant *Catalog* inclut la carte des plats de la boutique. Il permet au client connecté de voir ce qu'il peut commander.

3.1.1.2 - Composant Order Taking

Le composant *Order Taking* gère toute la création de commande pour le client connecté.

3.1.1.3 - Composant Order Checking

Le composant *Order Checking* gère toute la gestion de commande déjà créée. Le client pourra modifier, annuler ou visualiser sa commande.

3.1.2 - Package Store Interface

Il regroupe toute l'interface et les fonctions auxquelles les employés des boutiques et les responsables auront accès. il comprend les composant suivants.

3.1.2.1 - Composant Authentication

Le composant *Authentication* est lié aux autres composants (*Customer Interface*, *Stock*, *Order* et *Administrator*) pour autoriser l'accès ou non à ces composants suivant le profil.

3.1.2.1.1 Profile Management

Le composant *Profile Management* gère toute la gestion de compte (création, suppression, modification...).

3.1.2.2 - Composant Stock

Le composant *Stock* gère tout ce qui touche au stock des fournitures et ingrédients.

3.1.2.2.1 Product Management

Le composant *Product Management* permet de visualiser le stock de chaque ingrédient, de gérer les mouvements de stocks (entrée, sorties) et de réaliser l'inventaire des stocks.

3.1.2.3 - Composant Administrator

Le composant *Adminsitrator* est réservé aux responsables et à la direction. Il donne accès à la partie administrative et résultats.

3.1.2.3.1 Administrative management

Le composant *Administrative Management* permet d'avoir accès au chiffre d'affaires, toute la gestion de la carte (ajout, modification, suppression d'un plat...) et toute la partie gestion des profils employés (donner l'accès aux différents composants du système).

3.1.2.4 - Composant Order

Le composant *Order* permet de gérer la partie commande du point de vue des différents employés.

3.1.2.4.1 Order Management

Le composant *Order Management* permet créer, modifier, supprimer et visualiser les commandes.

3.1.2.4.2 Preparation

Le composant *Preparation* est destiné principalement au pizzaiolo qui pourra retrouver les recettes des plats à préparer.

3.1.2.4.3 Delivery

Le composant *Delivery* gère toute la partie livraison. Il est principalement destiné aux livreurs et aux employés de boutique pour visualiser l'état de la livraison (en cours, en livraison, livrée).

3.1.2.5 - Composant Bank System

Le composant *Bank System* permet de gérer les transactions lors d'une commande. il intervient directement avec les clients lors de leur commande en ligne mais aussi en boutique lors du paiement.

3.2 - Base de données

3.2.1 - SGBD utilisé

Le système de gestion de base de données (SGBD) un logiciel qui permet le stockage d'informations dans une base de données. Il permettra de créer, lire, mettre à jour, et supprimer les données qui sont contenues dans la base de données.

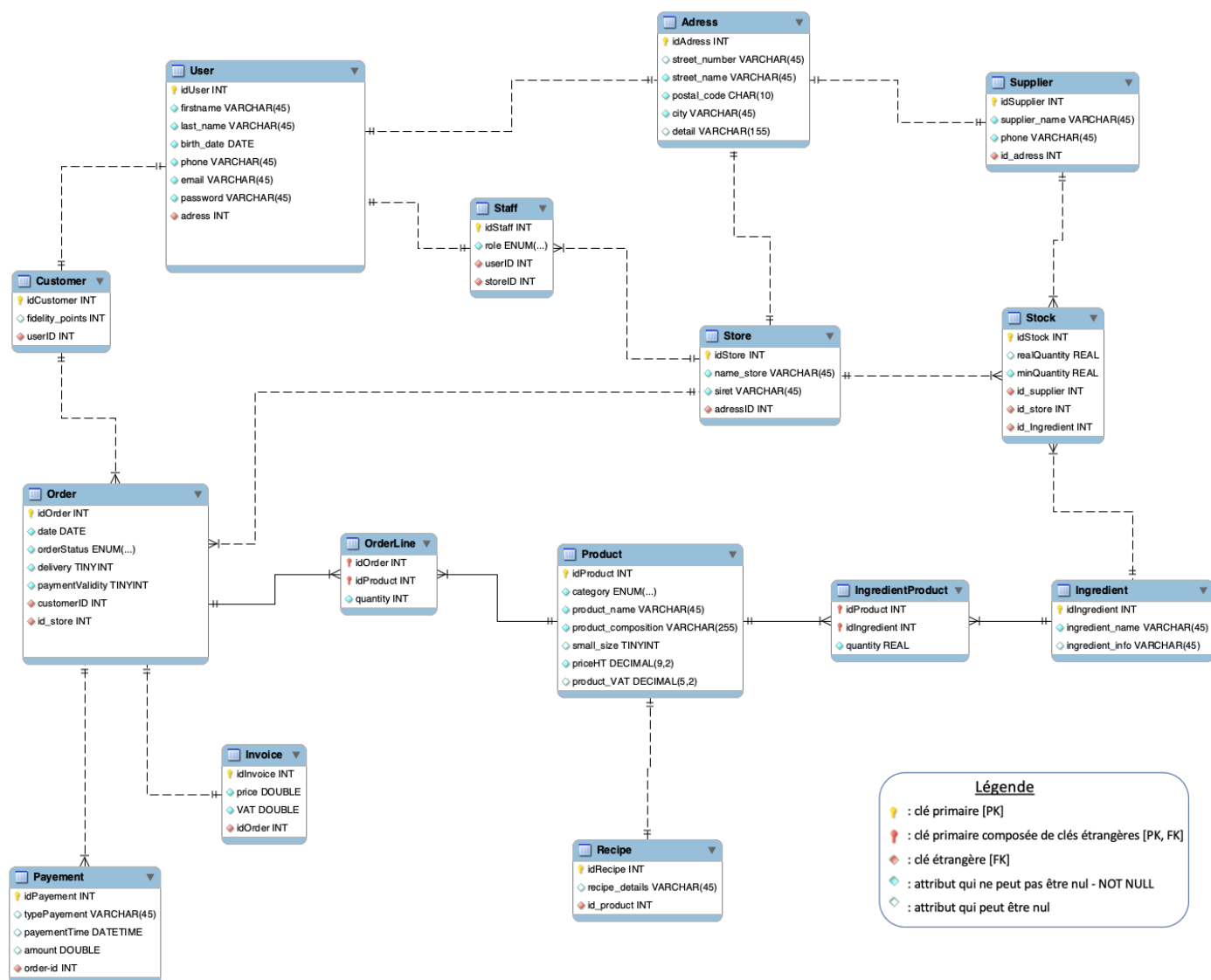
Nous proposons d'utiliser le SGBDR MySQL.

MySQL est la base de données open source la plus populaire au monde et elle est totalement compatible avec le serveur de données que nous avons sélectionné (RDS de AWS). Il permettra de gérer la base de données OC Pizza.

3.2.2 - Modèle Physique de données

Partant du diagramme de classe présenté dans le dossier de conception fonctionnelle, nous pouvons décrire le modèle physique de données de notre système.

Il permet de visualiser les tables de données nécessaires, leurs liens et les clés qui permettront de trouver la donnée qu'il nous faut.



4 - ARCHITECTURE DE DEPLOIEMENT

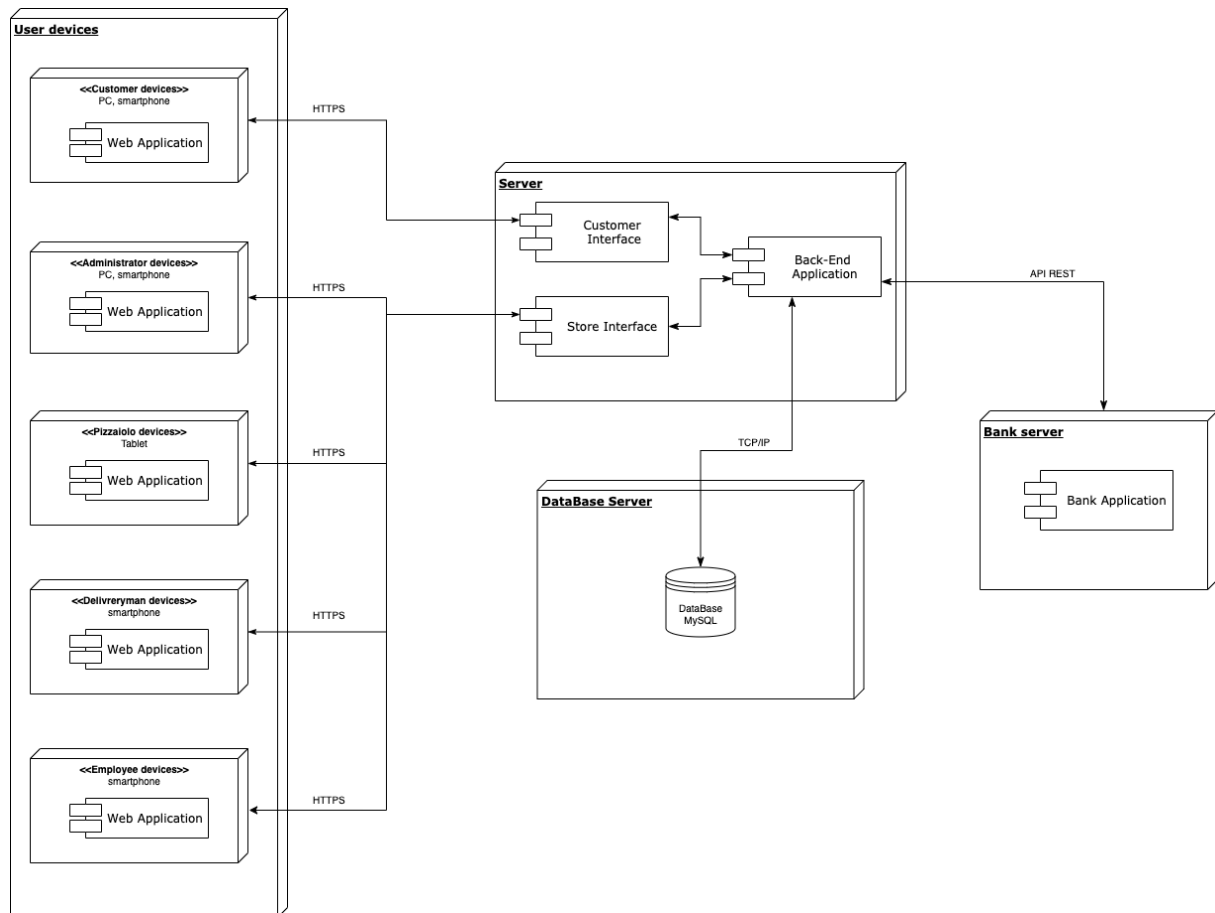


Diagramme UML de déploiement

Le diagramme de déploiement se concentre sur les éléments physiques qui composeront le système. Nous y retrouvons les différents appareils des utilisateurs (PC, smartphone, tablette...) sur lesquels sera installée l'application web. Ces différents appareils communiqueront avec le serveur qui contiendra les deux interfaces et le cœur de l'application. C'est elle qui communiquera avec les serveurs de la banque pour les paiements. Elle communiquera également avec la base de données qui stockera les informations dans le serveur de données. Pour ce déploiement nous avons choisi d'utiliser les services d'AWS (Amazon Web Services), leader sur ce type de service, pour les parties serveur et serveur de base de données. AWS nous fournit une solution complète avec tous les outils dont nous aurons besoin pour utiliser, configurer et maintenir à jour notre solution.

4.1 - Serveur de Base de données

Hébergé par AWS, le serveur de base de données sera une instance **d'Amazon Lightsail**. Il nous permettra d'installer, de gérer et de mettre à l'échelle facilement notre base de données relationnelle.

Pour la partie technique, nous partirons sur une instance **Lightsail**, prévue pour un usage général, sous **Ubuntu**.

Cette instance se base sur un processeur AMD, 512Mo de RAM et 20Go SSD de stockage.

Ce choix est évolutif en fonction des performances et du développement futur du système.

Nous pourrons y installer **MySQL** afin de gérer notre base.

4.2 - Serveur

Hébergé par AWS, le serveur sera une instance **d'Amazon Lightsail**.

Pour la partie technique, nous partirons sur une instance, prévue pour un usage général, sous **Linux**, suivant une configuration **LAMP (PHP 7)**.

Cette instance se base sur un processeur AMD, 512Mo de RAM et 20Go SSD de stockage.

Nous pourrons y installer un serveur web **Apache avec PHP** qui contiendra notre application.

La connexion avec entre ce serveur et le serveur de données sera privé et donc seul ce serveur pourra se connecter à la base de données (risque de piratage de la base de données est réduit au maximum).

5 - GLOSSAIRE

AWS (Amazon Web Services) : Division du groupe Amazon, spécialisée dans les services de cloud computing. Elle donne accès à des services informatiques (serveurs, stockage, mise en réseau, logiciels,...) via internet.

Amazon Lightsail : Service d’AWS qui permet aux développeurs de créer des sites web ou des applications web. Il inclut tout pour lancer un projet : serveurs privés virtuels, services de conteneur, bases de données gérées, équilibreurs de charge, stockage, adresses IP statiques et sauvegardes.

SGBDR (Système de Gestion de Base de Données Relationnel) : logiciel système servant à stocker, à manipuler ou gérer, et à partager des données dans une base de données, où les données sont placées dans des tables avec des clés primaires.

UML : Langage de Modélisation Unifié, de l’anglais Unified Modeling Language (UML), est un langage de modélisation graphique à base de pictogrammes conçu comme une méthode normalisée de visualisation de projet de développement.