



# SPÉCIFICATIONS TECHNIQUES

11/07/2021 version 1.0

Document établi par : **Paul Ghibeaux**



Adresse: 10 rue de Paradis, PARIS 75010

Mail: [it-consulting@it.com](mailto:it-consulting@it.com)

Site web: [www.it-consulting.com](http://www.it-consulting.com)

S.A.R.L. au capital de 1 000,00 €,

Enregistrée au RCS de – SIREN 999 999 999 – Code APE : 6202A

# Sommaire

<b>I. Versions</b>	<b>3</b>
<b>II. Introduction</b>	<b>4</b>
A. Objet du document	4
B. Références	4
<b>III. Architecture</b>	<b>5</b>
A. Domaine fonctionnel	5
B. Déploiement	8
<b>IV. Modèle physique de données</b>	<b>11</b>
A. Définition	11
B. Diagramme d'activité	13
<b>V. Implémentation dans la base</b>	<b>14</b>
A. Présentation	14
B. Manipulation et migration de la base de données	15
<b>VI. Conclusion</b>	<b>17</b>

## I. VERSIONS

Version	Auteur	Date	Description
0.1	Paul Ghibeaux	11/07/2021	Document établi à partir du fichier de spécifications fonctionnelles.
0.2	Paul Ghibeaux	16/07/2021	Conseils mentors, présentation et mise en forme

## II. INTRODUCTION

### A. Objet du document

Ce document présente les spécifications techniques constituant le projet de conception d'un système informatique pour l'ensemble des pizzerias du groupe OC Pizza.

Par convention, ce projet de système informatique sera nommé 'système' dans le reste du document.

Ce document a pour objectif :

- d'identifier les différents éléments composant le domaine fonctionnel.
- de décrire le déploiement des différents composants (diagramme de déploiement).
- de représenter le modèle physique de données.
- de présenter les choix techniques pour le projet.

### • Références

Les diagrammes figurant dans ce document suivent la norme UML 2.5.1.

Ces diagrammes ont été créés avec draw.io, et miro.com.

Le présent document est mis en forme avec Pages.



### **Système d'Exploitation :**

macOS 11.4 Big Sur

### **Serveur Local :**

MAMP 6.4

Apache 2.4.46

PHP 7.4

MySQL 8.0.25

### **Modèle Physique de Données :**

MySQL Workbench 8.0

### **Editeur :**

Atom 1.57.0

Les fichiers sources sont disponibles dans le dossier "Source" joint à ce dossier.

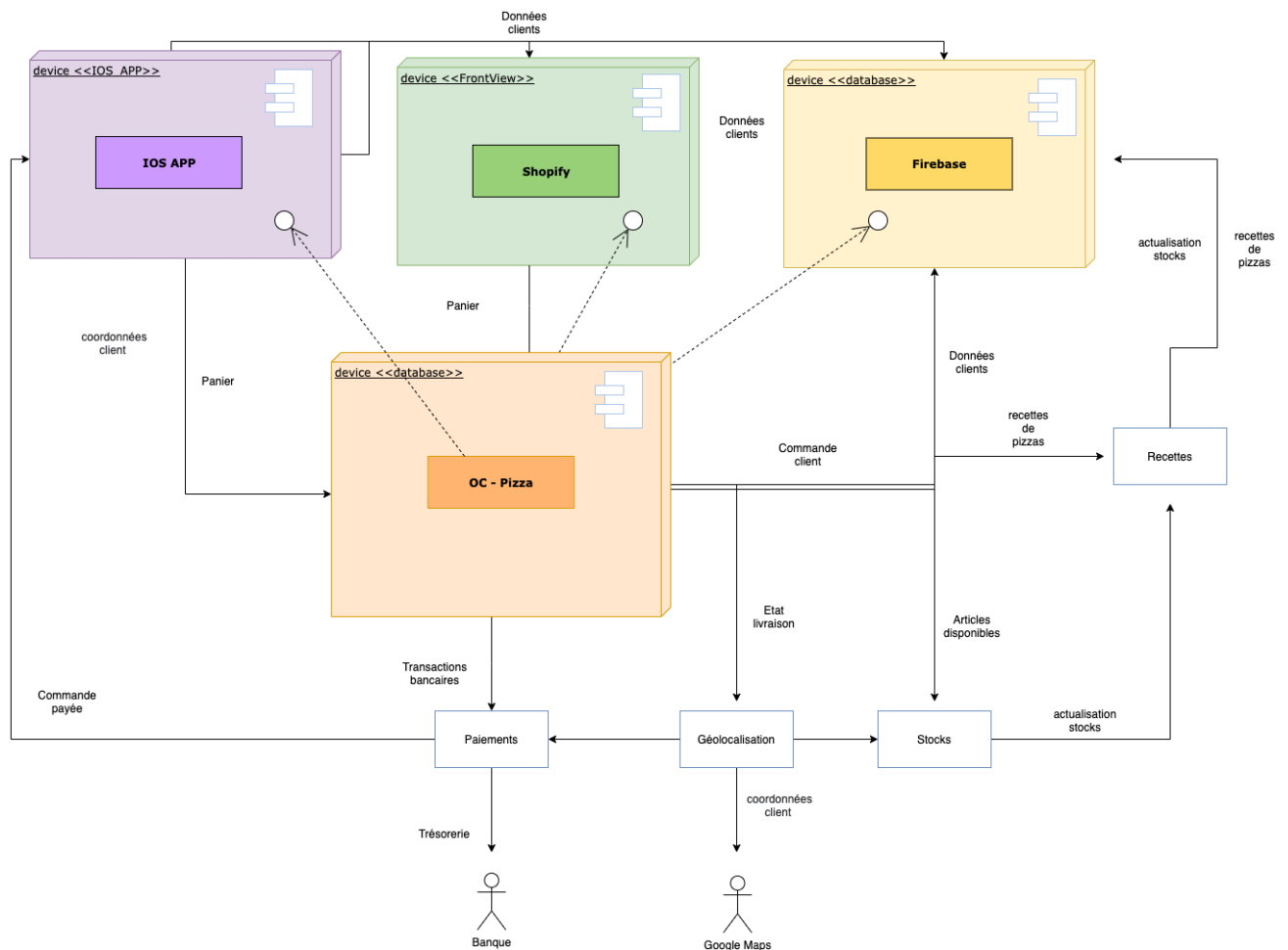
Nous pourrions nous référer aux documents suivants pour des compléments d'information :

- Dossier de Conception Fonctionnel, (DCF)
- Dossier de Conception Technique, (DCT)
- Dossier d'Exploitation, (DE)

### III. ARCHITECTURE

#### A. Domaine fonctionnel

Le système comprend trois éléments majeurs:



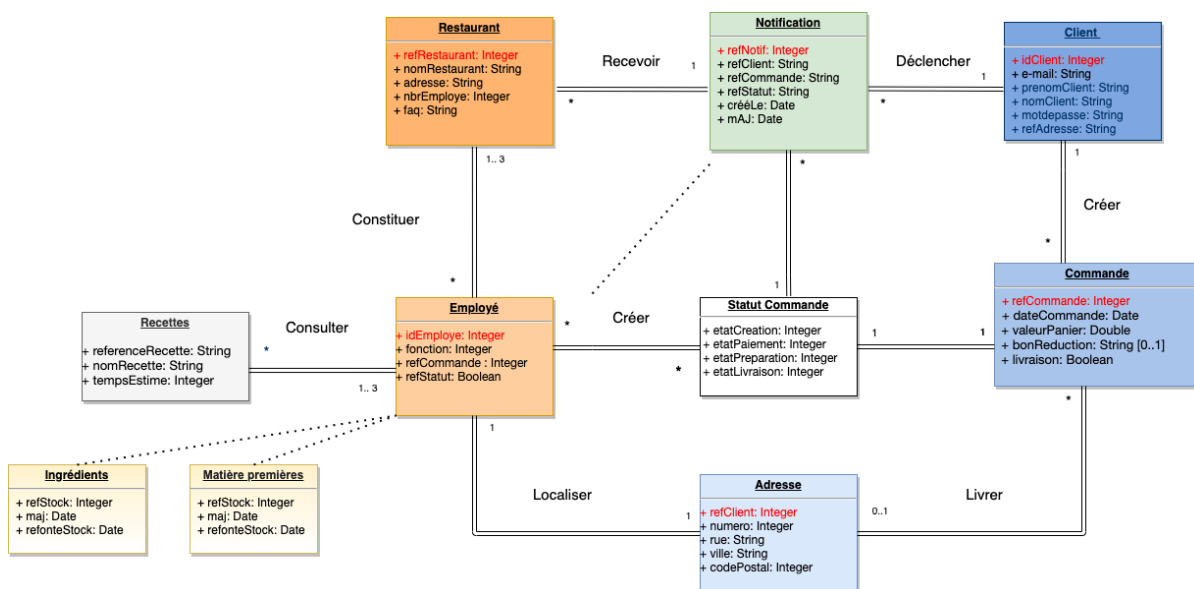
- Un site web (CMS marchand) pour commander.
- Une application iOS pour préparer et livrer la commande.
- Une base de données qui permet la transition des données collectées et manipulées.

Le site web et l'application iOS permettent la mise à jour des données en temps réel.

Le diagramme suivant, nous permet de mettre en place la structure du système, de concevoir les variables nécessaires au fonctionnement de celui-ci, et d'initialiser le modèle physique de données.

Par convention et pour aider à la lisibilité du diagramme, les classes sont arbitrairement colorées :

- les classes liées aux acteurs du système sont en couleur foncée
- les classes liées aux commande sont en bleu.
- les classes liées à OC-Pizza sont en orange, ou jaune.



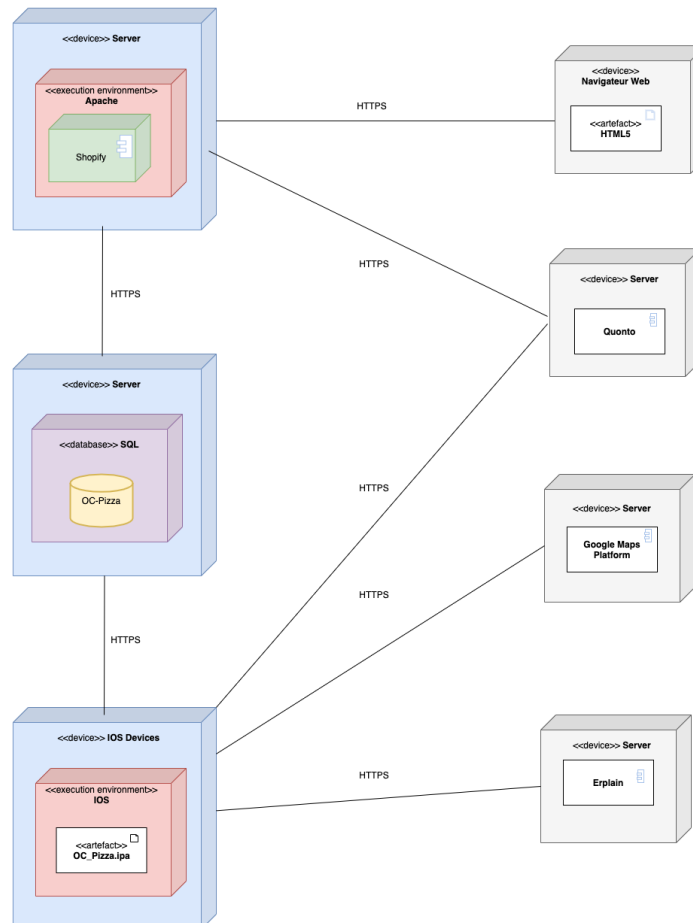
## Diagramme de classe

- **Commande** et **client** sont liés 1 à \* par la relation composite créer.
- **Adresse** est lié de 0 à n car il faut laisser la possibilité au **client** de récupérer sa commande sur le lieu de préparation de la commande.

- Lorsque le **client** crée une commande, une **notification** est déclenchée pour le **restaurant** le plus proche.
- Le **client** pourra recevoir plusieurs notifications par la suite avec le **statut de sa commande**, c'est pour cela que client et notification sont liés de 1 à \*.
- L'attribut booléen livraison permet au **client** d'indiquer au système si la **commande** sera livrée ou retirée.
- Le **restaurant** localisé reçoit la **notification de commande**, le préparateur prend les ingrédients et les matières premières et a accès aux recettes.
- La classe **employé** est liée 1 à \* car il peut y avoir plusieurs types d'employés pour chaque OC-Pizza en fonction de l'activité du point de vente.
- L'attribut booléen 'refStatut' permet de savoir si la **commande** est prise en charge par un **restaurant**. Si 'false' le **restaurant** est invité à accepter la **commande** avant d'être renvoyé sur un autre.

## B. Déploiement

Le déploiement du système est présenté sous forme de diagramme de déploiement.



### Diagramme de déploiement

Chaque entrée de cette liste représente un nœud du déploiement du système.  
La communication entre ces nœuds est conforme au protocole sécurisé HTTPS.



Il est proposé que :

- Le site web soit Shopify, déployé sur un serveur Apache.



+




- L'application iOS soit déployée sur :  
iPad WiFi 32Go pour les pizzaïolos et le management opérationnel.



iPhone 8 64Go pour les livreurs.



- Un serveur de base de données soit configuré pour un système de gestion de base de données relationnelles MySQL
  - La banque soit Qonto, qui permettrait une automatisation des factures, et un meilleur suivi de compte.
- 
- Les données de géolocalisation soient fournies par Google Maps Platform.
  - L'actualisation des stocks soit faite par le progiciel de gestion intégrée (PGI, en anglais Enterprise Resource Planning, ERP) Dolibarr ERP, ou Erplain qui propose une période d'essai de 14 jours et un abonnement mensuel entre 25 à 89 € HT mensuels.



Il a été proposé d'utiliser les appareils mobiles Apple pour la gestion des commandes, de la préparation à la livraison car ils sont fiables pour un environnement professionnel. Par ailleurs le système de distribution de l'application iOS proposé par Apple est simple à mettre en œuvre et facile à maintenir.

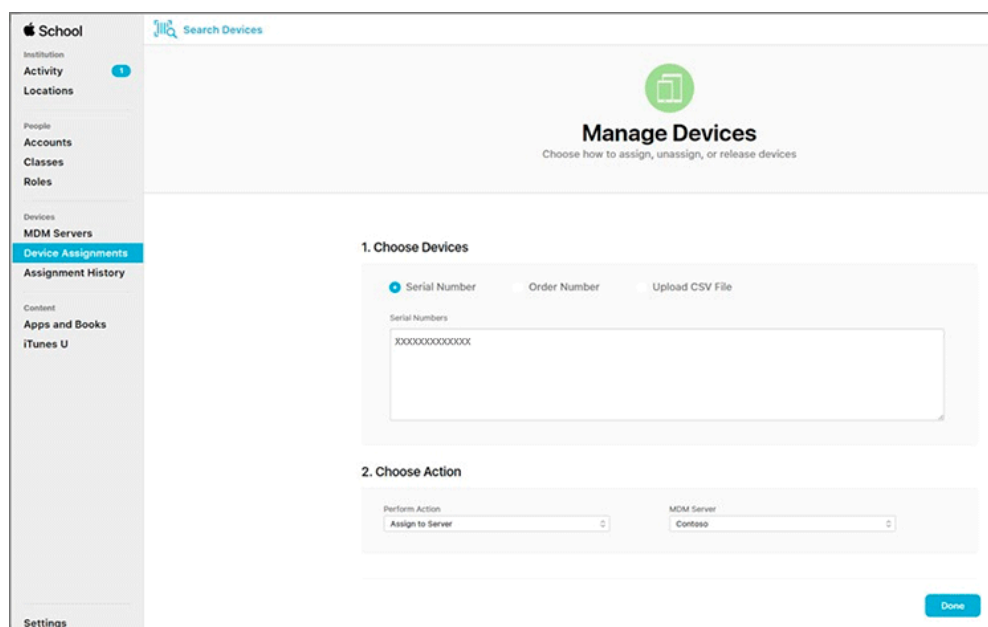
Ce processus efficace facilite l'utilisation, et l'évolution de l'application.

Il est conseillé d'utiliser le système de flux de développement git flow parce qu'il rationalise le processus de travail collaboratif de nouvelles fonctionnalités et sécurise la version de l'application en production.

Les outils collaboratifs proposés pour la mise en place du système forment un ensemble cohérent et s'intègrent parfaitement les uns avec les autres. Cela a pour but de fluidifier les processus de travail et de communication, d'être réactif et efficace, et de gagner en productivité.

## Déploiement des appareils mobiles

Une vue d'ensemble des appareils mobiles en entreprise est présentée dans le document [Managing Devices & Corporate Data on iOS](#) proposé par apple.



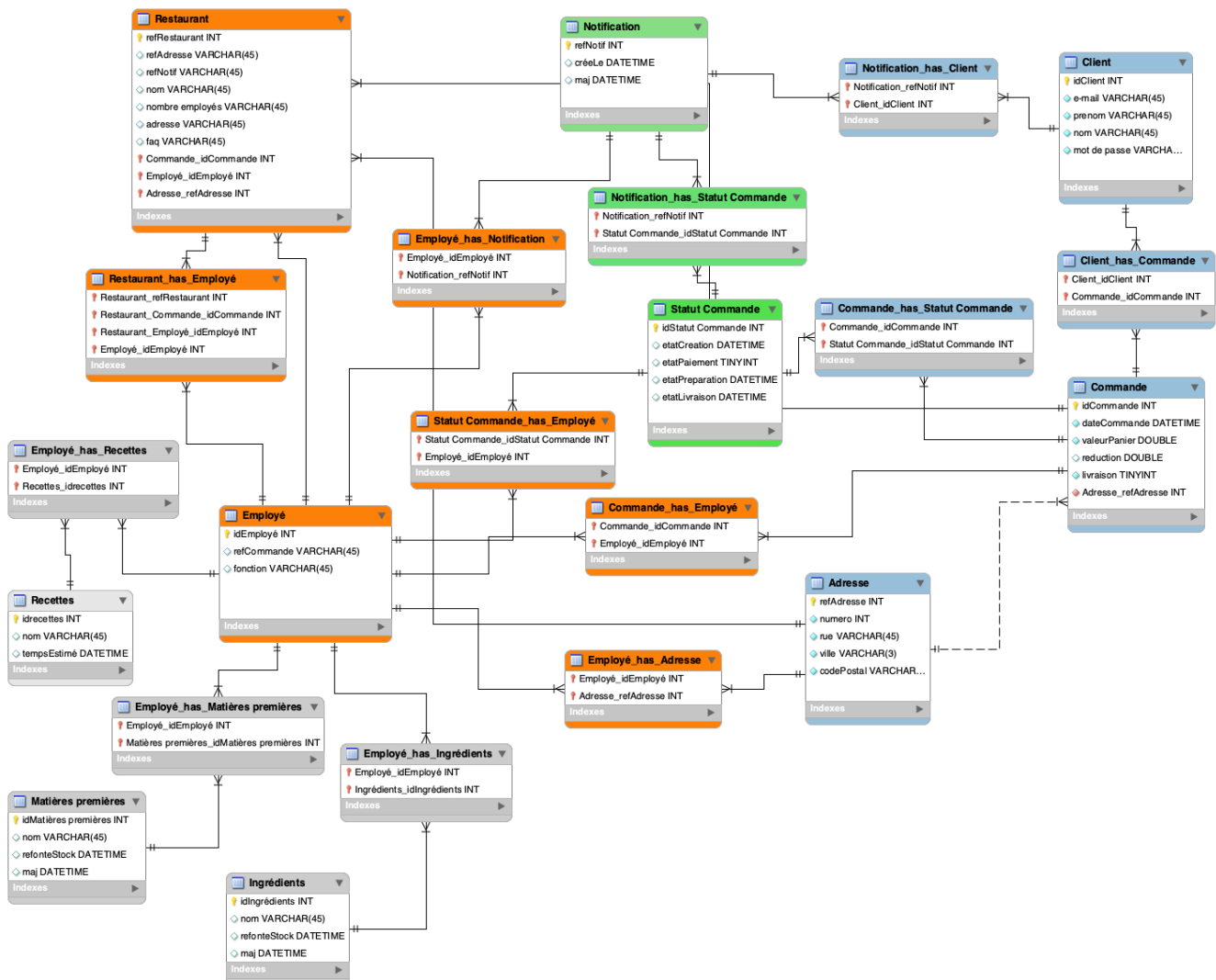
### **MDM Exemple**

On peut se référer au document [iOS Deployment Reference](#) relatif au déploiement en entreprise des appareils mobiles Apple. Ce document nous informe sur les nécessités d'infrastructures WiFi et réseau.

Il nous renseigne aussi sur l'intégration de gestionnaire de parc d'appareils mobiles (Mobile Device Management, MDM).



Après avoir fait ce premier modèle, nous avons réalisé un autre diagramme plus détaillé à l'aide du logiciel **MYSQL Workbench** qui permet d'avoir une meilleure visibilité sur les clés primaires, les différents attributs et les connexions entre tables.



**Diagramme de classe détaillé**

En plus d'être ergonomique et simple d'utilisation, le logiciel prévoit une implémentation facile des modèles vers les bases de données destinataires.

Ce qui permettrait un gain de temps potentiel sur le déploiement de la base de données pour OC-Pizza.

## B. Diagramme d'activité

Le diagramme d'activité donne une vision d'ensemble des principes de fonctionnement du système lors de la réception d'une notification traitée par OC-Pizza. Il illustre de manière dynamique les fonctionnalités du système mises en jeu lors d'une commande.

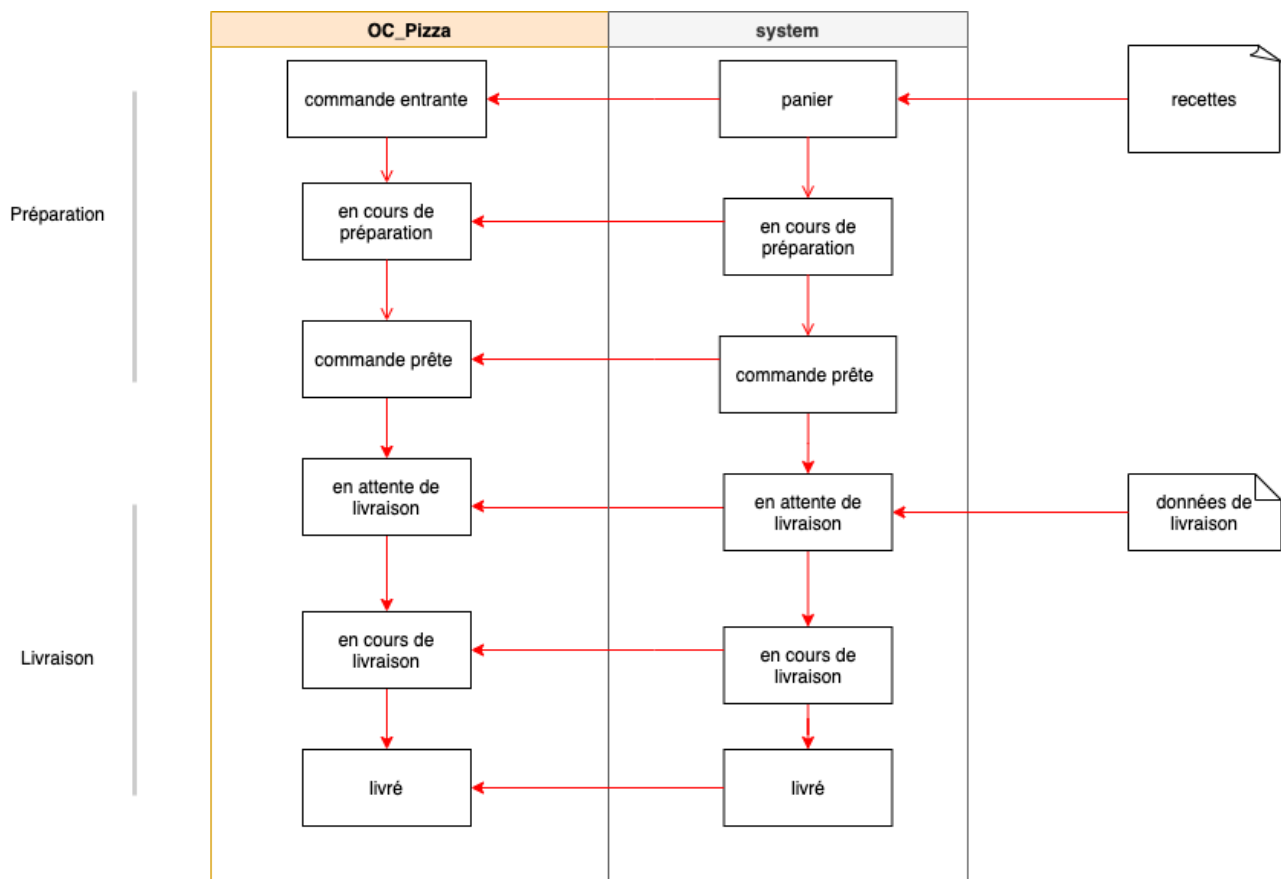


Schéma des notifications

En se référant au diagramme, OC-pizza représente l'employé, le système l'application IOS et le serveur de base de données.

Par exemple sur un user story création de panier /commande, celui-ci est créé dans la base de données et rattaché au client directement.

Le système notifie alors le cuisinier avec une recette en attente de préparation.

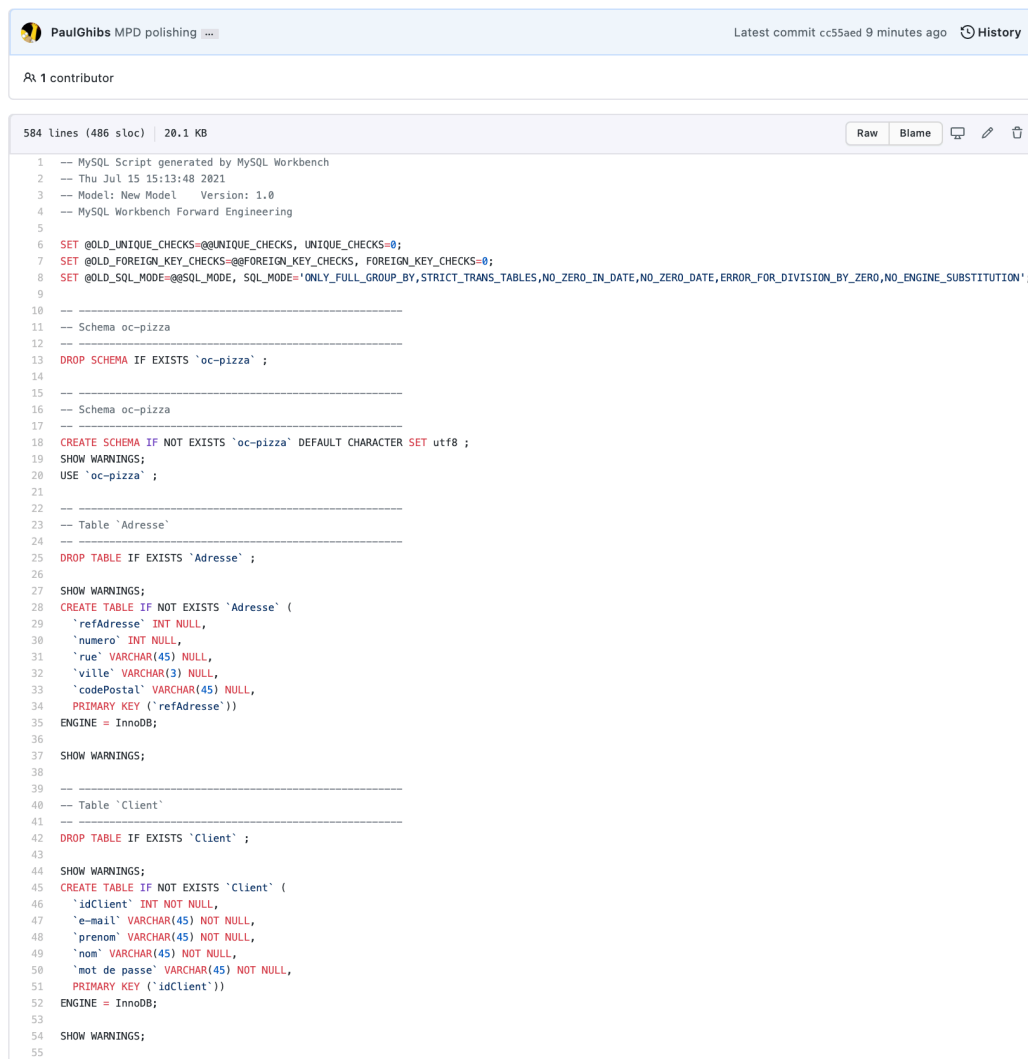
Si le cuisinier accepte de préparer la commande, le système enregistre son id-employé dans l'audit pour savoir qui est le dernier à avoir mis à jour le statut. Lorsque la commande est prête, le système passe son statut « en attente de livraison ».

Ensuite celle-ci est attribuée à un livreur et ainsi de suite.

## V. IMPLÉMENTATION DANS LA BASE

### A. Présentation

Dans la mesure où le modèle physique de données conçu à l'aide de **SQL POWER ARCHITECT**, ne fournirait pas de script de création de base de données, nous avons opté pour un outil plus récent **MYSQL Workbench**, qui génère le script vu plus bas.



```

1  -- MySQL Script generated by MySQL Workbench
2  -- Thu Jul 15 15:13:48 2021
3  -- Model: New Model    Version: 1.0
4  -- MySQL Workbench Forward Engineering
5
6  SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
7  SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
8  SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';
9
10 --
11 -- Schema oc-pizza
12 --
13 DROP SCHEMA IF EXISTS `oc-pizza` ;
14
15 --
16 -- Schema oc-pizza
17 --
18 CREATE SCHEMA IF NOT EXISTS `oc-pizza` DEFAULT CHARACTER SET utf8 ;
19 SHOW WARNINGS;
20 USE `oc-pizza` ;
21
22 --
23 -- Table `Adresse`
24 --
25 DROP TABLE IF EXISTS `Adresse` ;
26
27 SHOW WARNINGS;
28 CREATE TABLE IF NOT EXISTS `Adresse` (
29   `refAdresse` INT NULL,
30   `numero` INT NULL,
31   `rue` VARCHAR(45) NULL,
32   `ville` VARCHAR(3) NULL,
33   `codePostal` VARCHAR(45) NULL,
34   PRIMARY KEY (`refAdresse`))
35 ENGINE = InnoDB;
36
37 SHOW WARNINGS;
38
39 --
40 -- Table `Client`
41 --
42 DROP TABLE IF EXISTS `Client` ;
43
44 SHOW WARNINGS;
45 CREATE TABLE IF NOT EXISTS `Client` (
46   `idClient` INT NOT NULL,
47   `e-mail` VARCHAR(45) NOT NULL,
48   `prenom` VARCHAR(45) NOT NULL,
49   `nom` VARCHAR(45) NOT NULL,
50   `mot de passe` VARCHAR(45) NOT NULL,
51   PRIMARY KEY (`idClient`))
52 ENGINE = InnoDB;
53
54 SHOW WARNINGS;
55

```

Ce script est également disponible dans le dossier source sous le nom « **OC-pizza\_creationSQL** ».

Il permet la création des 22 tables présentes dans le modèle physique de données, par ordre alphabétique sur MYSQL.

## B. Manipulation et migration de la base de données

Un script d'insertion avec un jeu de données de démo est également disponible dans le dossier source sous le nom « **OC-pizza\_InsertionSQL** ».

```

1  START TRANSACTION;
2
3  --- Table `ocpizza`.`client`
4  ---
5
6  INSERT INTO `oc-pizza`.`Client`
7  (`idClient`, `e-mail`, `prenom`, `nom`, `mot de passe`)
8  VALUES
9  (1, 'alan.turing@gmail.com', 'Alan', 'Turing', 'crypto'),
10 (2, 'grace.hopper@outlook.com', 'Grace', 'Hopper', 'compilation'),
11 (3, 'john.vonneumann@yahoo.com', 'John', 'Von Neumann', 'memoire')
12 ;
13
14 --- Table `ocpizza`.`adresse`
15 ---
16
17 INSERT INTO `oc-pizza`.`Adresse`
18 (`refAdresse`, `numero`, `rue`, `ville`, `codePostal`)
19 VALUES
20 (1, '7', 'rue Poissonnière', 'Paris', '75002'),
21 (2, '7', 'boulevard Haussman', 'Paris', '75009'),
22 (3, '3', 'rue Bleue', 'Paris', '75009')
23 ;
24
25 --- Table `ocpizza`.`restaurant`
26 ---
27
28 INSERT INTO `oc-pizza`.`Restaurant`
29 (`refRestaurant`, `refAdresse`, `refNotif`, `nom`, `nombre employés`, `adresse`, `faq`, `Commande_IdCommande`, `Employé_IdEmployé`, `Adresse_refAdresse`)
30 VALUES
31 (1, '0173456989', '#58979', 'OCpizza_parcmentier', '2', 'Parmentier', 'Lorem Ipsum questionate', '70990', '277', '1'),
32 (2, '0323456189', '#28098', 'OCpizza_SaintDenis', '3', 'SaintDenis', 'Lorem Ipsum questionate', '68003', '276', '2'),
33 (3, '0623656739', '#78639', 'OCpizza_VictorHugo', '6', 'VictorHugo', 'Lorem Ipsum questionate', '56009', '275', '3')
34 ;
35
36 --- Table `ocpizza`.`commande`
37 ---
38
39 INSERT INTO `oc-pizza`.`Commande`
40 (`idCommande`, `dateCommande`, `valeurPanier`, `reduction`, `livraison`, `Adresse_refAdresse`)
41 VALUES
42 ('1790', '2018-07-14 12:46:28', '25', NULL, '1', '1'),
43 ('7090', '2018-07-13 13:02:12', '11', NULL, '0', '2')
44 ;
45
46 --- Table `Statut Commande`
47 ---
48
49 INSERT INTO `oc-pizza`.`Statut Commande`
50 (`idStatut Commande`, `etatCreation`, `etatPaiement`, `etatPreparation`, `etatLivraison`)
51 VALUES

```

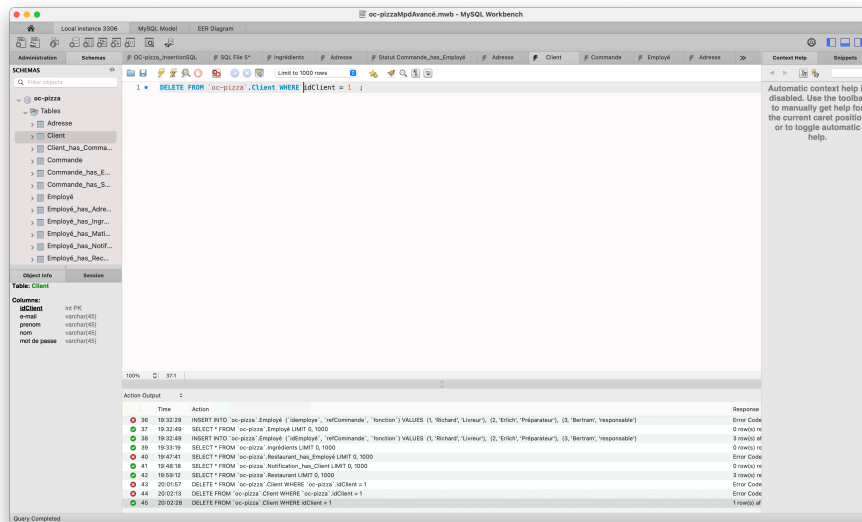
### OC-pizza\_InsertionSQL

Ce script est sous le langage SQL, il permet de construire le schéma de la base de données et vérifie également si celle-ci ne contient aucun doublon ou erreur dans les attributs des tables.

Ce schéma a mis en lumière l'oubli d'attributs dans les classes d'ingrédients et matières premières, ce qui pouvait entrainer un problème dans la gestion des stocks.

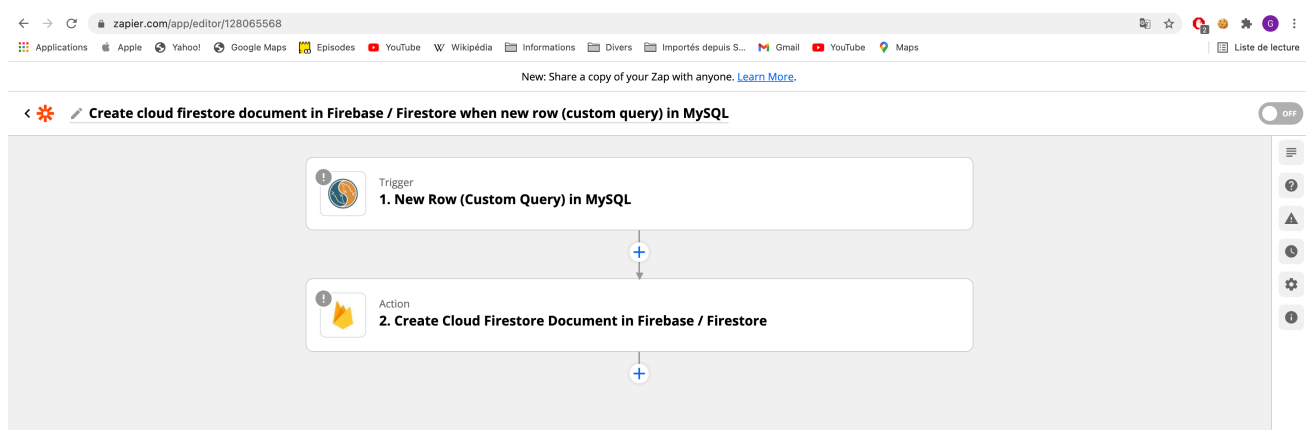
Les tables ont donc été modifiées pour contenir le nom en plus d'une référence de date.

Les autres opérations classiques du « **CRUD** », comme CREATE, READ, UPDATE, DELETE fonctionnent également sur la base créée.



### Migration vers firebase :

En se référant à l'article « [10.8 Using the MySQL Workbench Migration Wizard](#) » de la documentation officielle de MYSQL, il est possible d'exporter sa base de données créée à l'aide de MYSQL Workbench. Une fois que la migration est réalisée, et que la base de données est hébergée en ligne, nous pouvons faire l'ajout des clients automatiquement dans **Firestore**, en passant par un outil comme Zapier ou Integromat.





## VI. CONCLUSION

Le choix de la solution proposée est fondé sur une argumentation basée sur des sources fiables.

Afin de répondre aux besoins exprimés par OC-Pizza, nous avons pris la décision de mettre en place :

- une application **iOS** pour :
  - la gestion des commandes en temps réel,
  - leurs préparations,
  - leurs livraisons.
- un site marchand **Shopify** pour :
  - la gestion des commandes,
  - la gestion des clients.
- une base de données relationnelle **MySQL**, exportée sur **Firebase** pour :
  - structurer les données mises en jeu par le système,
  - la gestion des données en temps réel,
  - la gestion de leurs persistances.

Nous avons dans un premier temps présenté le domaine fonctionnel, en annonçant l'architecture de notre système, sous la forme de diagrammes de classes et de composants.

Nous avons clairement défini le modèle physique de données ainsi que les éléments constituant la base de données.

Pour finir nous avons établi la base de données, à l'aide de l'outil **MYSQL Workbench**, elle contient un jeu de données de démonstration, pour simuler une commande fictive.