

OC Pizza

Nouveau Système Informatique

Dossier de conception technique

Version 1.0

Auteur

Johann Ulma

Analyste Programmeur



TABLE DES MATIERES

1 - Versions	5
2 - Introduction	6
2.1 - Objet du document	6
2.2 - Références	6
3 - Architecture Technique	7
3.1 - Composants généraux	7
3.2 - Les composants internes	8
3.2.1 - Composant Boutique Web	8
3.2.1.1 - Sous-composant Authentification	8
3.2.1.2 - Sous-composant Panier	9
3.2.1.3 - Sous-composant Point de vente	9
3.2.2 - Composant Commandes	9
3.2.2.1 - Sous-composant Commande en cours	9
3.2.2.2 - Sous-composant Statut de la commande	9
3.2.2.3 - Sous-composant Historique des commandes	9
3.2.2.4 - Sous-composant Livraison	9
3.2.3 - Composant Pizzeria	10
3.2.3.1 - Sous-composant Produit	10
3.2.3.2 - Sous-composant Stock	10
3.2.3.3 - Sous-composant Recette	10
3.2.4 - Composant Administration	10
3.2.4.1 - Sous-composant Gestion du Groupe	10
3.2.4.2 - Sous-composant Gestion des comptes	10
3.2.4.3 - Sous-composant Gestion des droits	10
3.3 - Les composants externes	11
3.3.1.1 - La base de données	11
3.3.1.2 - Le système de paiement	11
3.4 - Application Web et Mobile	11
3.4.1 - Back end	11
3.4.2 - Front end	11
3.4.3 - Mobile	12
3.5 - La base de données	12
3.5.1 - SGBD	12
3.5.2 - Modèle Physique de données	12
3.5.3 - Détail des tables	13
3.5.4 - Détail des énumérations	15
4 - Architecture de Déploiement	18
4.1 - Serveur de Base de données	19
4.2 - Serveur web	19
5 - Architecture logicielle	20
5.1 - Principes généraux	20
5.1.1 - Les couches	20

5.1.2 - Structure des sources	20
6 - Points particuliers	21
6.1 - Gestion des logs.....	21
6.2 - Ressources.....	21
6.2.1 - Charte graphique.....	21
6.2.2 - Données.....	21
6.3 - Environnement de développement.....	21
6.3.1 - Documentation	21
6.3.2 - Environnement	22
6.4 - Procédure de packaging / livraison.....	22
7 - Glossaire	23

1 - VERSIONS

Auteur	Date	Description	Version
J. Ulma	18/05/2021	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.

Objectif du document :

Décrire l'architecture technique afin de fournir les éléments nécessaires aux développeurs à mettre en place la solution technique.

Les éléments du présent dossier découlent :

- Du recueil des besoins et du cahier des charges donné par le client
- De la présentation de la solution technique avec le Chef de projet
- Du dossier fonctionnel

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. **CE – 1.0** : Dossier d'exploitation de l'application

3 - ARCHITECTURE TECHNIQUE

3.1 - Composants généraux

Le diagramme de composant décrit l'organisation du système du point de vue des éléments logiciels. Nous retrouvons donc les composants **Boutique Web**, **Commandes** et **Pizzeria**, à cela s'ajoute le composant **Administration** qui est spécifique à l'**Administrateur** ou au **Gérant**.

On a deux types d'accès celui du **client** (Front Office) et celui du **personnel de la Pizzeria** (Back Office).

Avant de pouvoir commander le client doit se connecter. Son **Panier** va être lié automatiquement au composant **Commandes** qui lui-même va nécessiter une interface de **Pizzeria** afin de savoir ce qui est disponible.

Dès qu'un **Panier** est validé il devient une **Commande en cours** et le processus de **Suivi** s'active.

On peut aussi constater des dépendances comme la **Livraison** qui va dépendre du Suivi ou la **Recette** qui dépend du **Produit**.

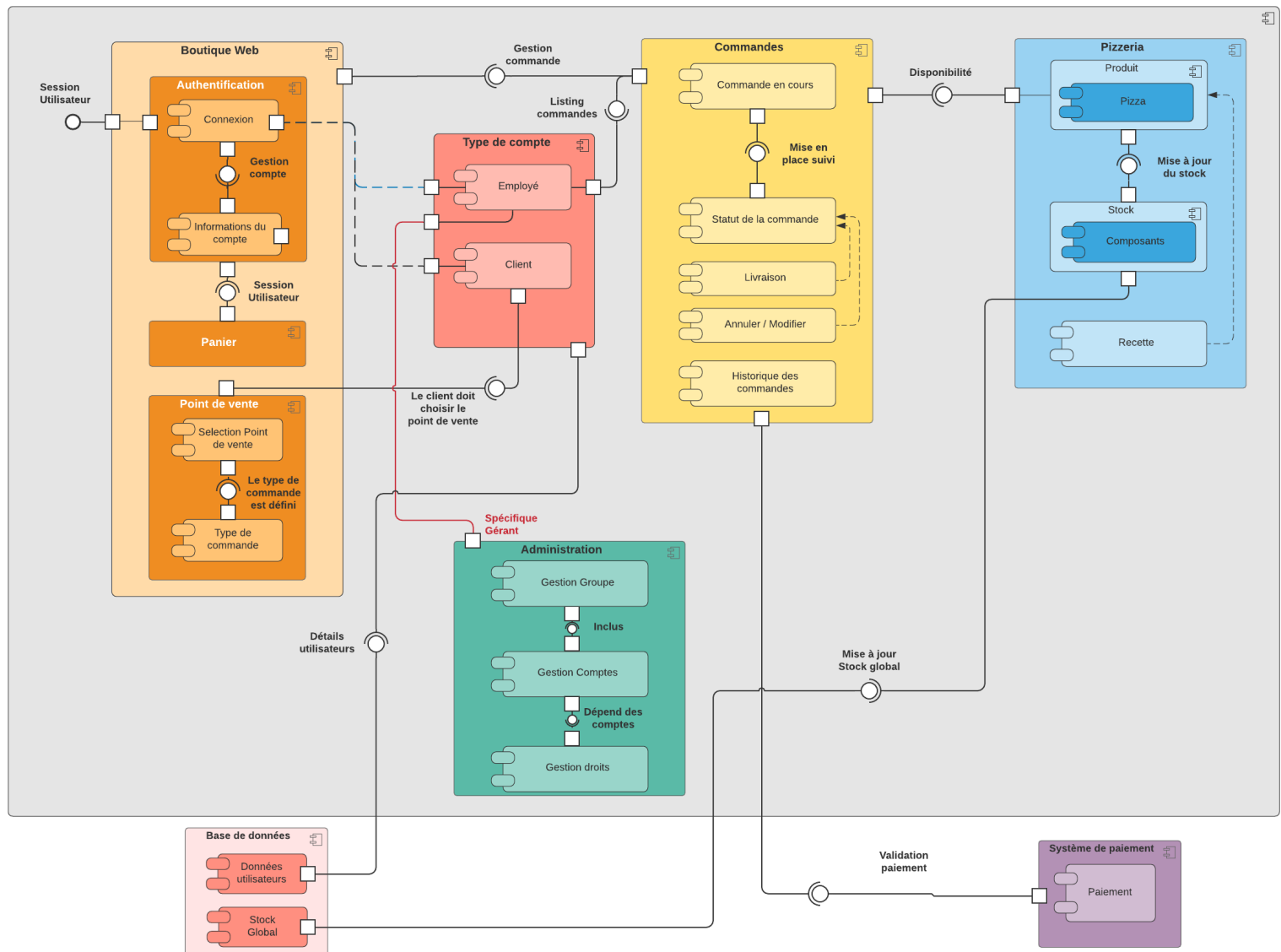


Diagramme UML de Composants

3.2 - Les composants internes

3.2.1 - Composant Boutique Web

On va regrouper les fonctionnalités nécessaires pour la connexion au compte et à la prise d'une commande

3.2.1.1 - Sous-composant Authentification

Il permet l'authentification de l'utilisateur. Il va permet la connexion ou l'inscription d'un

utilisateur et va contenir les informations du comptes (Adresse, Téléphone, Mail, ...)

Par défaut lors de la création d'un compte, le rôle est défini sur client.

3.2.1.2 - Sous-composant Panier

Il permet la création d'une commande par l'utilisateur.

Les options seront plus ou moins poussées selon le rôle de l'utilisateur.

3.2.1.3 - Sous-composant Point de vente

Lors de son arrivée sur le site le client sélectionne le point de vente et le type de commande (à emporter, en livraison)

3.2.2 - Composant Commandes

Regroupe les fonctionnalités nécessaires au suivi des commandes pour un client ou une pizzeria.

3.2.2.1 - Sous-composant Commande en cours

Il permet d'afficher une commande et son détail avec les lignes de commandes et le montant de la commande. Il va donc permettre le suivi des commandes actuelles.

3.2.2.2 - Sous-composant Statut de la commande

Il permet de donner les informations liées à une commande et de l'avancement de son état, il dépend de la commande en cours.

L'avancement du statut va permettre d'annuler / modifier une commande quand cela est encore possible.

3.2.2.3 - Sous-composant Historique des commandes

Va contenir l'ensemble des commandes selon le cas il peut s'agir de la Pizzeria, du client voir du livreur.

3.2.2.4 - Sous-composant Livraison

Il permet la gestion des informations liées à la livraison et remonte les informations à suivi de la commande.

3.2.3 - Composant Pizzeria

Va contenir les informations sur une Pizzeria pour ses produits disponibles et apporter des informations supplémentaires dans la partie cuisine.

3.2.3.1 - Sous-composant Produit

Il permet l'affichage des produits proposés par une pizzeria notamment les pizzas. Il contient les informations détaillées des différents produit et leur prix de vente.

Il est relié au stock afin d'afficher la disponibilité des produits.

3.2.3.2 - Sous-composant Stock

Il permet la mise à jour en temps réel des composants qui sont encore disponibles afin d'informer sur les produits que peut proposer la pizzeria.

3.2.3.3 - Sous-composant Recette

Il permet l'affichage d'une recette et des composants pour le Pizzaiolo, il est relié aux produits afin d'afficher la recette correspondante.

3.2.4 - Composant Administration

Ce package est spécifique au gérant car il permet des manipulations avancées sur les comptes et un visu global du groupe.

3.2.4.1 - Sous-composant Gestion du Groupe

Permet au gérant d'avoir une vue en temps réel sur l'activité du groupe (commandes, stocks, livraisons, ...)

3.2.4.2 - Sous-composant Gestion des comptes

Va permet de créer des comptes utilisateurs pour la pizzeria ou le cas échéant de faire des modifications.

3.2.4.3 - Sous-composant Gestion des droits

Par défaut un utilisateur dispose du droit client par mesure de sécurité.

Le gérant peut affecter un niveau de droit différent pour un utilisateur de manière à lui donner un accès plus spécifique.

3.3 - Les composants externes

Notre Application a besoin de deux composants externes pour fonctionner.

3.3.1.1 - La base de données

Elle va contenir les informations nécessaires à la partie Authentification, Commande et Pizzeria.

Chaque composant a recours à la base de données pour fournir les informations nécessaires que ce soit pour vérifier les informations d'un compte utilisateur, d'afficher le détail d'une commande ou d'informer sur les produits disponibles et leur disponibilité.

3.3.1.2 - Le système de paiement

Le paiement d'une commande passera par le système de paiement qui va enregistrer le règlement d'une commande.

3.4 - Application Web et Mobile

La pile logicielle est la suivante :

• Debian « <i>Buster</i> » version 10.9 ou ultérieure	OS Serveur
• Apache version 2.4.46 ou ultérieure	Serveur Web
• Python version 3.9.5 ou ultérieure	Back End
• Django version 3.2.3 ou ultérieure	Back End
• HTML version 5.3 ou ultérieure	Front End
• CSS version 4.15 ou ultérieure	Front End
• JavaScript version <i>ECMAScript2020</i> ou ultérieure	Front End
• AngularJS version 1.8.2 ou ultérieure	Front End
• MySQL version 8.0.25 ou ultérieure	Base de données
• Swift version 5.4 ou ultérieure	Application iOS

3.4.1 - Back end

Le Back-end est la partie invisible du site en charge du côté logique du site.

Celui-ci sera codé en Python afin un Framework Django.

Le back-end sera en lien direct avec notre base de données.

3.4.2 - Front end

Le front-end est la partie visible du site-web (notre interface client et équipe pizzeria) pour ce

faire nous allons utiliser les standards que sont HTML, CSS et JavaScript.
Afin de modéliser notre site nous utiliserons un Framework Angular JS

3.4.3 - Mobile

L'application mobile étant sous iOS nous utiliserons le langage Swift pour développer notre application.

3.5 - La base de données

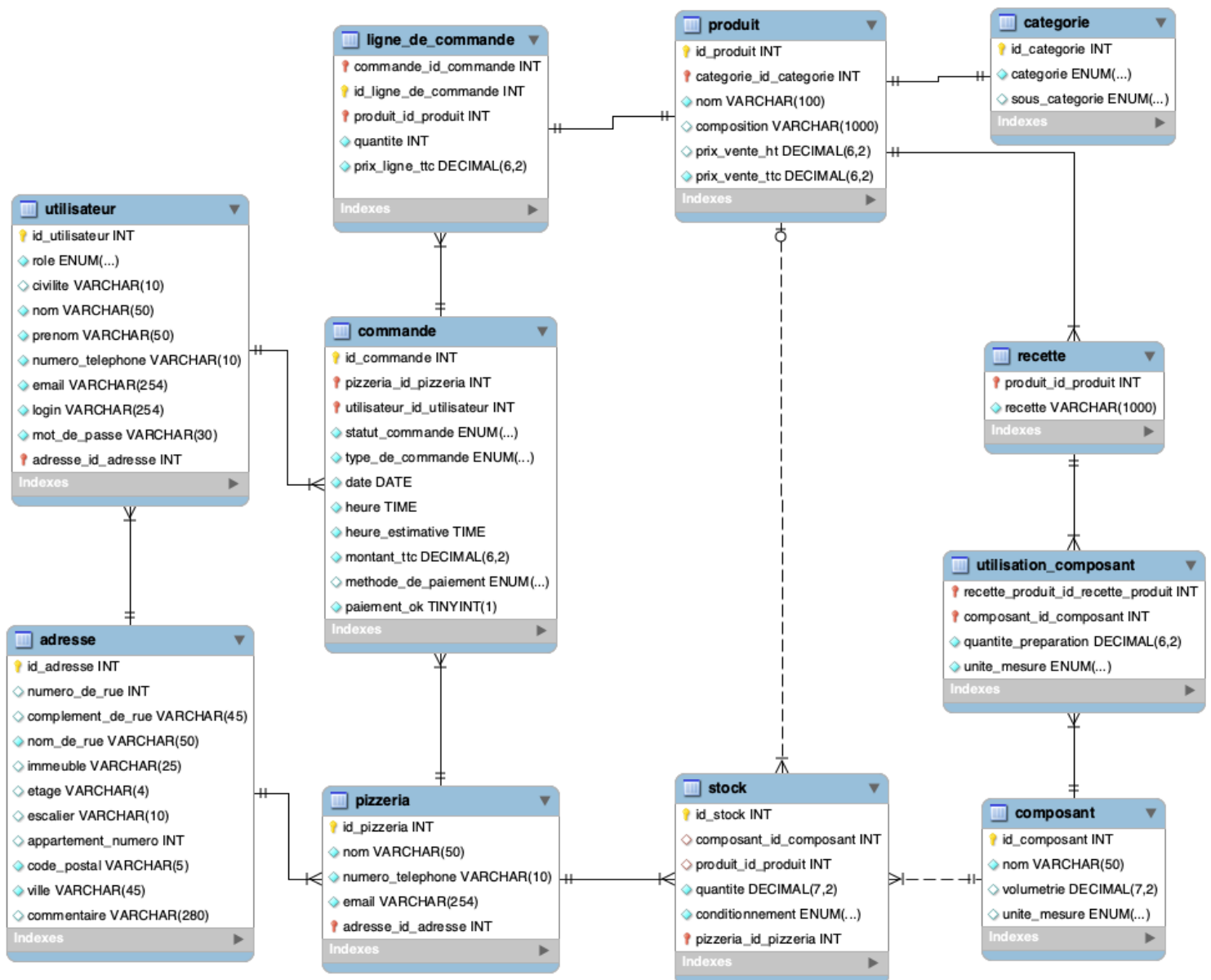
3.5.1 - SGBD

Notre choix pour le système de gestion de base de données c'est porté sur MySQL.

3.5.2 - Modèle Physique de données

Le modèle physique de données représente notre base de données telle qu'elle a été conçue en se basant sur notre diagramme de classe visible dans la documentation fonctionnelle.

Le modèle va représenter les relations entre les différentes tables.



3.5.3 - Détail des tables

Les caractéristiques des différentes tables sont rappelées ci-dessous, nous retrouvons la table cible avec la précision des attributs.

L'encart de droite vient expliquer le rôle de chaque attribut préciser si la valeur peut être NULL et donner un exemple d'utilisation.

utilisateur	
id_utilisateur	INT
role	ENUM(...)
civilite	VARCHAR(10)
nom	VARCHAR(50)
prenom	VARCHAR(50)
numero_telephone	VARCHAR(10)
email	VARCHAR(254)
login	VARCHAR(254)
mot_de_passe	VARCHAR(30)
adresse_id_adresse	INT
Indexes	

Cette table va contenir les informations, coordonnées des utilisateurs (clients et employés). Elle sera reliée aux tables Adresse, Pizzeria et Commande.

id_utilisateur	Identifiant unique attribué lors de la création d'un utilisateur	99	NOT NULL
role	Définit le rôle de l'utilisateur (client ou employé)	Pizzaiolo	NOT NULL
civilite	Titre de civilité pour identifier la personne (Mr, Mme, ...)	Mr	NULL
nom	Nom de famille de l'utilisateur	Bachelard	NOT NULL
prenom	Prénom de l'utilisateur	Morgan	NOT NULL
numero_telephone	Définit le numéro de téléphone de la personne	0199000277	NOT NULL
email	Définit le mail de la personne	morgan.bachelard@ocpizza.fr	NOT NULL
login	Identifiant de connexion	morgan.bachelard@ocpizza.fr	NOT NULL
mot_de_passe	Mot de passe de connexion	9kDcYh2Q	NOT NULL

pizzeria	
id_pizzeria	INT
nom	VARCHAR(50)
numero_telephone	VARCHAR(10)
email	VARCHAR(254)
adresse_id_adresse	INT
Indexes	

Cette table va contenir la liste des Pizzerias du groupe. Elle sera reliée aux tables Adresse, Utilisateur, Commande et Stock.

id_pizzeria	Identifiant unique attribué lors de la création d'une pizzeria	9	NOT NULL
nom	Nom de la Pizzeria	OC Pizza Mantes-la-jolie	NOT NULL
numero_telephone	Numéro de téléphone de la Pizzeria	0199000278	NOT NULL
email	Adresse mail de contact de la Pizzeria	mantes@ocpizza.fr	NOT NULL

adresse	
id_adresse	INT
numero_de_rue	INT
complement_de_rue	VARCHAR(45)
nom_de_rue	VARCHAR(50)
immeuble	VARCHAR(25)
etage	VARCHAR(4)
escalier	VARCHAR(10)
appartement_numero	INT
code_postal	VARCHAR(5)
ville	VARCHAR(45)
commentaire	VARCHAR(280)
Indexes	

Cette table va contenir les adresses des utilisateurs et des pizzerias. Elle sera reliée aux tables Utilisateur et Pizzeria.

id_adresse	Identifiant unique attribué lors de la création d'une adresse	56	NOT NULL
numero_de_rue	Numéro lié à la rue de l'utilisateur ou de la pizzeria	124	NULL
complement_de_rue	Type spécifique de rue (bis, ter, lieu dit, ...)	Bis	NULL
nom_de_rue	Nom de la rue de l'utilisateur ou de la pizzeria	Rue Francis Bacon	NOT NULL
immeuble	Nom de l'immeuble	Immeuble Lumière	NULL
etage	Numéro de l'étage	3	NULL
escalier	Numéro d'escalier	3	NULL
appartement_numero	Numéro de l'appartement	45	NULL
code_postal	Code Postal lié à l'adresse	78200	NOT NULL
ville	Nom de la ville	Mantes-la-Jolie	NOT NULL
commentaire	Permet de renseigner des éléments spécifiques (digicode, ...)	Digicode 1212, Sonner en bas	NULL

commande	
id_commande	INT
pizzeria_id_pizzeria	INT
utilisateur_id_utilisateur	INT
statut_commande	ENUM(...)
type_de_commande	ENUM(...)
date	DATE
heure	TIME
heure_estimative	TIME
montant_ttc	DECIMAL(6,2)
methode_de_paiement	ENUM(...)
paiement_ok	TINYINT(1)
Indexes	

Cette table va contenir l'ensemble des commandes. Elle sera reliée aux tables Utilisateur, Pizzeria et LigneDeCommande.

id_commande	Identifiant unique attribué lors de la création d'une commande	173	NOT NULL
statut_commande	Contient le statut de la commande en cours	En preparation	NOT NULL
type_de_commande	Identifie quelle type de commande nous avons	Livraison	NOT NULL
date	Date de la commande	14-04-2021	NOT NULL
heure	Heure de la commande	12:36:23	NOT NULL
heure_estimative	Heure estimative pour la disponibilité de la commande	13:03:00	NOT NULL
montant_ttc	Montant de la commande	35.6	NOT NULL
methode_de_paiement	Méthode de paiement choisie	Carte bancaire	NULL
paiement_ok	Paiement effectué ou non	true	NOT NULL

PaiementOK est un Booléen permettant rapidement de changer le statut de celui-ci	
true	Le paiement est validé
false	Le paiement n'est pas encore validé

ligne_de_commande	
commande_id_commande	INT
id_ligne_de_commande	INT
produit_id_produit	INT
quantite	INT
prix_ligne_ttc	DECIMAL(6,2)
Indexes	

Cette table va contenir le détail d'une ligne de commande. Elle sera reliée aux tables Commande et Produit

id_ligne_commande	Identifiant unique pour chaque ligne	1	NOT NULL
quantite	Définit le nombre d'éléments	2	NOT NULL
prix_ligne_ttc	Prix de la Ligne TTC	30.0	NOT NULL

produit	
id_produit	INT
categorie_id_categorie	INT
nom	VARCHAR(100)
composition	VARCHAR(1000)
prix_achat_ht	DECIMAL(6,2)
prix_vente_ht	DECIMAL(6,2)
prix_vente_ttc	DECIMAL(6,2)
Indexes	

Cette table va contenir les informations des produits vendus dans l'enseigne.
Elle sera reliée aux tables LigneDeCommande, Categorie, Recette et Stock.

id_produit	Identifiant unique attribué lors de la création d'un produit	53	NOT NULL
nom	Nom du produit	Onion Rings	NOT NULL
composition	Détail des éléments qui font le produit	6 anneaux d'oignons panés	NULL
prix_achat_ht	Prix d'achat du produit HT	NULL	NULL
prix_vente_ht	Prix de vente du produit HT	3.33	NULL
prix_vente_ttc	Prix de vente du produit TTC	3.99	NOT NULL

categorie	
id_categorie	INT
categorie	ENUM(...)
sous_categorie	ENUM(...)
Indexes	

Cette classe va contenir les catégories et sous catégories de produits.
Elle sera reliée à la classe Produit.

id_categorie	Identifiant unique attribué à une catégorie	3	NOT NULL
categorie	Type de produit	pizza	NOT NULL
sous_categorie	Sous Catégorie pour un type de produit	Base crème	NULL

recette	
produit_id_produit	INT
recette	VARCHAR(1000)
Indexes	

Cette classe va contenir le détail des étapes d'une recette.
Elle sera reliée aux classes Produit et UtilisationComposant.

recette	Affiche le détail de la recette de la pizza	1 - Préchauffer ...	NOT NULL
---------	---	---------------------	----------

composant	
id_composant	INT
nom	VARCHAR(50)
volumetrie	DECIMAL(7,2)
unite_mesure	ENUM(...)
Indexes	

Cette classe va contenir les composants lors de la préparation d'un produit.
Elle sera reliée aux classes Recette, Stock et UtilisationComposant.

id_composant	Identifiant unique attribué lors de la création d'un composant	28	NOT NULL
nom	Nom de l'ingrédient	Roquette	NOT NULL
volumetrie	Définit la quantité numéraire	200.00	NOT NULL
unite_mesure	Définit l'unité de mesure (gramme, pièce, ...)	grammes(s)	NOT NULL

utilisation_composant	
recette_produit_id_recette_produit	INT
composant_id_composant	INT
quantite_preparation	DECIMAL(6,2)
unite_mesure	ENUM(...)
Indexes	

Cette classe va afficher des ingrédients de la pizza pour le client.
Elle sera reliée aux classes Composant et Recette.

quantite_preparation	Définit la quantité numéraire d'un composant	7	NOT NULL
unite_mesure	Définit l'unité de mesure (gramme, pièce, ...)	tranche(s)	NOT NULL

stock	
id_stock	INT
composant_id_composant	INT
produit_id_produit	INT
quantite	DECIMAL(7,2)
conditionnement	ENUM(...)
pizzeria_id_pizzeria	INT
Indexes	

Cette classe va contenir le détail du Stock disponible d'une pizzeria.
Elle sera reliée aux classes Pizzeria, Composant et Produit.

id_stock	Nom de l'élément du stock	29	NOT NULL
quantite	Définit la quantité numéraire	6.00	NOT NULL
conditionnement	Définit l'unité de stockage de l'élément (boîte, bouteille, ...)	sachet(s)	NOT NULL

3.5.4 - Détail des énumérations

Tout comme les tables nous détaillons les énumérations de la même manière.

Enum Role
client
pizzaiolo
livreur
hotesse
gérant
administrateur

Détail des différents types de rôles possibles pour l'utilisateur.
Il s'agit d'une énumération liée à la table Utilisateur.

Client	La personne n'est pas un membre de l'équipe OC Pizza, les droits sont limités
Pizzaiolo	Définit les droits pour le Pizzaiolo
Livreur	Définit les droits pour le Livreur
Hôtesse	Définit les droits pour l'hôtesse
Gérant	Définit les droits pour le Gérant
Administrateur	Définit les droits pour l'Administrateur

Enum TypeDeCommande
retrait
livraison

Définit le type de commande effectuée par le client.
Il s'agit d'une énumération liée à la classe Commande

1	Retrait
2	Livraison

Enum StatutCommande
En attente
En preparation
Prete
A retirer
En Livraison
Livree
Close
Annulée

Détail des différents statuts possibles pour la commande.
Il s'agit d'une énumération liée à la table Commande.

En attente	La commande est passée mais pas encore lancée en cuisine
En preparation	La commande est en cuisine
Prête	La commande est finie côté cuisine
A retirer	La commande est disponible en retrait
En livraison	La commande est en cours de livraison
Livree	La commande est confirmée comme livrée
Close	La commande est clôturée
Annulée	La commande est annulée

Enum MethodeDePaiement
Espec
Carte Bancaire
Titre Restaurant
Paypal

Cette Enumération va contenir les méthodes de Paiement proposées.
Il s'agit d'une énumération liée à la table Commande.

Espèce	Le client paye en espèce
Carte Bancaire	Le client paye par carte ou sans contact
Titre Restaurant	Le client utilise un Titre Restaurant (Ticket Restaurant, Chèque Déjeuner, ...)
Paypal	Le client paye via Paypal

Enum Categorie
Entree
Pizza
Salade
Bruschetta
Dessert
Boisson
Sauce

Détail des différents types de catégories pour l'appartenance d'un produit.
Il s'agit d'une énumération liée à la classe Categorie.

Entree	Le produit est du type Entree (Produits panés)
Pizza	Le produit est du type Pizza
Salade	Le produit est du type Salade
Bruschetta	Le produit est du type Bruschetta
Dessert	Le produit est du type Dessert
Boisson	Le produit est du type Boisson
Sauce	Le produit est du type Sauce

Enum SousCategorie
Base crème
Base tomate
Vegan
Spéciale
Sucrée
Glacé
Pâtisserie
Gazeuse
Plate

Détail des différents types de catégories pour l'appartenance d'un produit.
Il s'agit d'une énumération liée à la classe `Categorie`.

Base crème	Sous-classe de pizza
Base tomate	Sous-classe de pizza
Vegan	Sous-classe de pizza
Spéciale	Sous-classe de pizza
Sucrée	Sous-classe de pizza
Glacé	Sous-classe de dessert
Pâtisserie	Sous-classe de dessert
Gazeuse	Sous-classe de boisson
Plate	Sous-classe de boisson

Enum Conditionnement
boite
bouteille
pot
sachet
canette
piece

Détail des différents conditionnements pour les composants.
Il s'agit d'une énumération liée à la classe `Stock`.

Boite	Le contenant du composant est une boite
Bouteille	Le contenant du composant est une bouteille
Pot	Le contenant du composant est un pot
Sachet	Le contenant du composant est un sachet
Canette	Le contenant du composant est une canette
Piece	Le composant est une pièce sans contenant spécifique

Enum UniteMesure
Gramme
Millilitre
Centilitre
Piece
Tranche

Détail des différentes unités de mesure pour les composants.
Il s'agit d'une énumération liée aux classes `Composant` et `UtilisationComposant`.

Gramme	L'unité de mesure est liée au poids : Gr
Millilitre	L'unité de mesure est liée au volume (litre) : Ml
Centilitre	L'unité de mesure est liée au volume (litre) : Cl
Piece	L'unité de mesure est liée à la quantité : Pièce
Tranche	L'unité de mesure est liée à la quantité : Tranche

4 - ARCHITECTURE DE DEPLOIEMENT

Notre architecture de déploiement est donc la suivante. Elle explique la disposition des composants du système sur les infrastructures physiques.

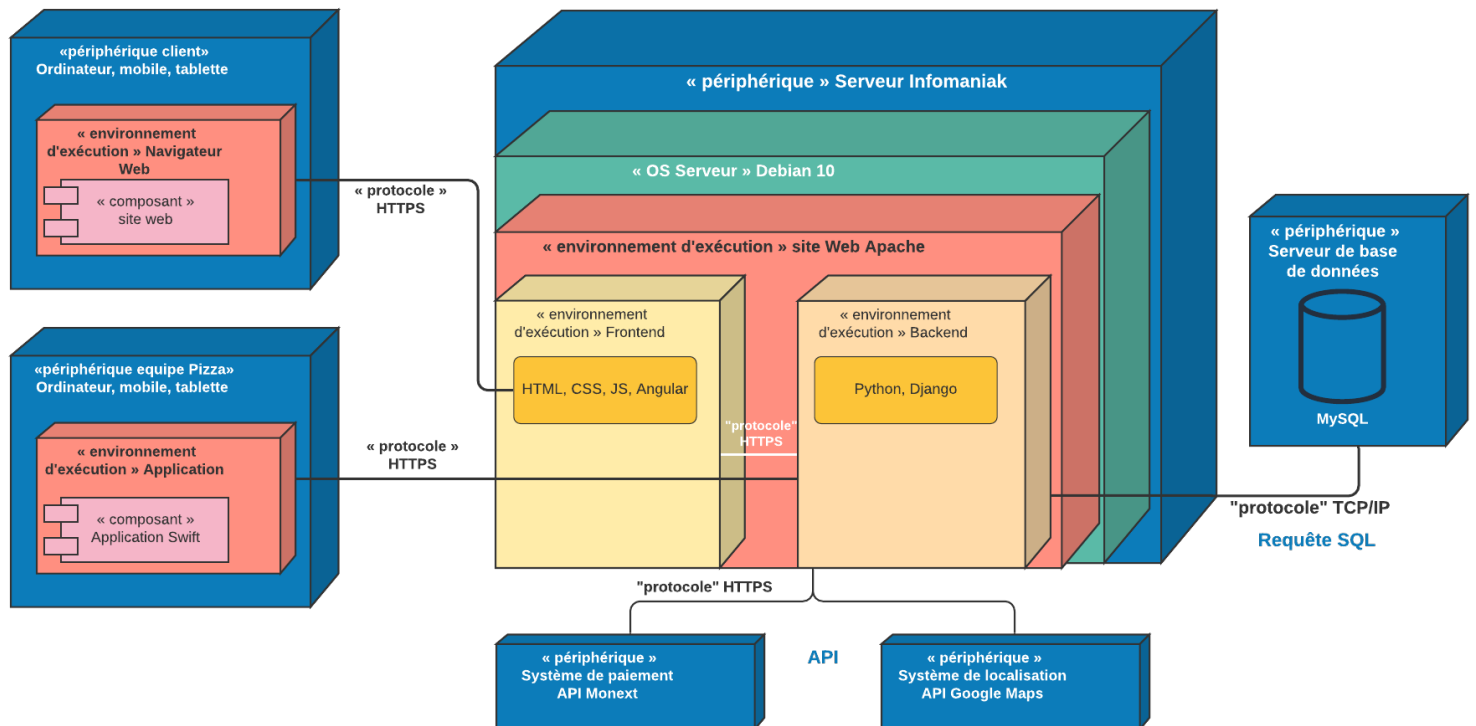


Diagramme UML de déploiement

Notre serveur est composé d'un Front-end et d'un Back-end qui vont communiquer ensemble sur notre serveur.

Nos utilisateurs qu'il s'agisse des clients ou du personnel de la pizzeria accèdent à la partie Front-end du site-web qui va permettre d'afficher les informations voulues.

Notre application iOS va elle aussi être en lien avec le Back-end.

Notre partie Back-end quant à elle, va communiquer avec la base de données via des requêtes SQL.

Enfin nous avons recours à des API pour la partie système de paiement et le système de localisation.

4.1 - Serveur de Base de données

Point central de notre système la base de données sera sous MySQL.

La base de données sera hébergée sur un serveur Infomaniak de marque Dell, gamme Poweredge doté de :

- 2 CPU
- 6 Go de RAM
- 100Go de SSD
- L'OS sera un Debian Buster.

Le schéma de la base de données a été vu dans la partie 3.5.2

4.2 - Serveur web

Pour le serveur Web nous avons choisi l'offre Hébergement Web d'Infomaniak qui inclus :

- 100 Go d'espace disque en SSD
- Traffic Mensuel illimité
- Disponibilité de 99.99%
- Protection Anti DDOS

A cela s'ajoute d'autres fonctions utiles :

- L'hébergeur fait appel aux technologies Open Source
- Elle garantit un engagement permanent sur les nouvelles méthodes de piratage

Les bases là aussi sont les mêmes avec un serveur de marque Dell, gamme Poweredge doté de :

- 2 CPU
- 6 Go de RAM
- 100Go de SSD
- L'OS sera un Debian Buster

L'offre choisie chez Infomaniak se nomme « Serveur Cloud Non managé », elle permet la création d'un serveur tel que nous l'avons conçu pour le client.

Le coût mensuel est de 29€ TTC par mois par serveur.

L'offre est sécurisée et facilement Scalable.

5 - ARCHITECTURE LOGICIELLE

5.1 - Principes généraux

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

L'application est construite selon le pattern **MVC**.

Les dépendances et le packaging par **Swift Package Manager**.

5.1.1 - Les couches

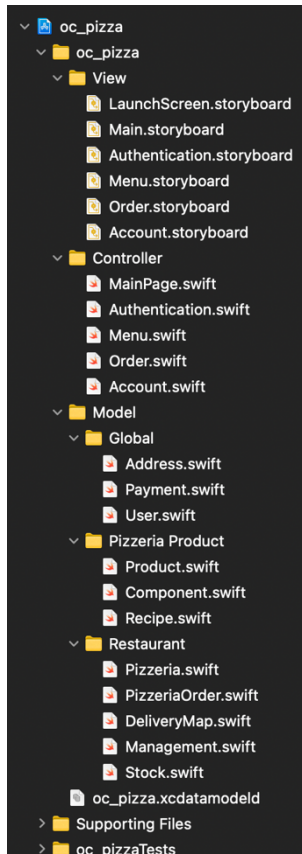
L'architecture applicative est la suivante :

- une couche **Modele** : responsable de la logique métier du composant
- une couche **Vue** : gestion de l'interface utilisateur
- une couche **Contrôleur** : qui permet la communication entre le **Modele** et la **Vue**

5.1.2 - Structure des sources

La structuration des répertoires du projet suit la logique suivante :

- les répertoires sources sont créés de façon à respecter l'architecture MVC



6 - POINTS PARTICULIERS

6.1 - Gestion des logs

Les logs serveurs seront dans le dossier /var/logs

Ce dossier regroupe les logs des différents services utilisés (Apache, MySQL, ...)

Afin de visualiser les logs on utilisera Grafana avec Loki et Promtail.

6.2 - Ressources

6.2.1 - Charte graphique

Les éléments graphiques (polices, logos, photos des produits, jeux de couleur, ...) sont fournis par **OC Pizza**.

6.2.2 - Données

Les données pour la création de la base de données sont-elles aussi fournies par **OC Pizza**, cela comprend notamment : les produits, les données utilisateurs, les adresses, ...

Mais aussi les références bancaires pour la partie encaissement.

6.3 - Environnement de développement

6.3.1 - Documentation

Les documentations techniques sont listées ci-dessous :

- Debian <https://www.debian.org/doc/index.fr.html>
- Apache <https://httpd.apache.org/docs/2.4/>
- Python <https://docs.python.org/3/>
- Django <https://docs.djangoproject.com/en/3.2/>
- HTML <https://html.spec.whatwg.org/multipage/>
- CSS <https://www.w3.org/Style/CSS/>
- JavaScript <https://developer.mozilla.org/fr/docs/Web/JavaScript>
- AngularJS <https://angularjs.org>
- MySQL <https://dev.mysql.com/doc/>
- Swift <https://swift.org/documentation/>
- Monext <https://docs.monext.fr/display/DT/API+Reference>
- GoogleMaps <https://developers.google.com/maps/documentation?hl=fr>

6.3.2 - Environnement

Le choix de l'environnement reste libre à chaque développeur néanmoins il existe des prérequis et des préconisations.

- L'IDE pour le développement de l'application iOS sera Xcode et nécessitera un environnement MacOS
- L'IDE recommandé pour le développement de la partie Site Web est Visual Studio Code
- L'outil de gestion pour MySQL reste libre mais phpMyAdmin est préférable
- Le versionning passera par un client Git qui sera interfacé avec GitHub

6.4 - Procédure de packaging / livraison

Le système fera l'objet d'un déploiement sur Infomaniak et sur l'App Store au moment de la livraison finale.

En parallèle le code source du site sera donnée dans un zip au client.

De même pour le code de l'application qui aura son propre zip.

A cela nous ajoutons un zip qui va contenir le dump de la base de données.

7 - GLOSSAIRE

API	Une API pour « Application Programming Interface », que l'on traduit par interface de programmation d'application. Elle permet d'utiliser les ressources, données ou fonctionnalités, d'une application web depuis une autre application.
Back end	Le back end (arrière-plan) en opposition au front end, fait référence à l'ensemble des éléments qui ne sont pas visibles directement sur le site web mais qui va permettre le bon fonctionnement d'un site internet. Le back end est exécuté côté serveur. Le back end utilise principalement Python, Java, PHP.
Back Office	Le back office regroupe un ensemble de tâches administratives ou logistiques liées à la vente (, mise à jour des stocks, livraison, moyens de paiement, etc.) C'est la partie du site visible par le personnel de la pizzeria.
Front end	Le front end ou (web frontal) est la partie d'une page internet ou d'une application qu'un utilisateur peut voir et avec lesquelles il peut interagir directement. Le front end est exécuté côté client (PC, Smartphone, Tablette) Le front end utilise principalement HTML, CSS et JavaScript.
Front Office	Le front office est l'ensemble des actions, fonctions ou tâches liées à la vente se faisant en contact avec le client. C'est la partie du site visible par le client.
Grafana	Grafana est une plate-forme open-source spécialisé dans l'analyse et la surveillance. Il fonctionne spécifiquement sur des données chronologiques provenant de sources telles que Prometheus et Loki.
Git	Logiciel de gestion de versions
IDE	Un environnement de développement informatique est un ensemble d'outils qui permet d'augmenter la productivité des programmeurs qui développent des logiciels.
Logs	Le terme log désigne un type de fichier, ou une entité équivalente, dont la mission principale consiste à stocker un historique des événements.
Loki	Loki est un système d'agrégation de journaux scalable horizontalement, hautement disponible et inspiré de Prometheus. Loki va s'occuper de la centralisation des logs.
MVC	Le MVC est un motif d'architecture logicielle destiné aux interfaces graphiques. Le motif est composé de trois types de modules ayant trois responsabilités différentes : les modèles, les vues et les contrôleurs.
Promtail	Promtail est un agent qui expédie le contenu des journaux locaux vers une instance Loki privée ou Grafana Cloud. Il est généralement déployé sur chaque machine sur laquelle des applications doivent être surveillées.
SGBD	Système de Gestion de Base de données.