

OC Pizza

Système de gestion de pizzerias

Dossier de conception technique

Version 1.0

Auteur
Charpentier Yves
Développeur iOS

TABLE DES MATIÈRES

1.Versions	4
2.Introduction	5
2.1.Objet du document	5
2.2.Références	5
3.Architecture Technique	6
3.1.Composants généraux	6
4.Architecture de Déploiement	7
4.1.Serveur de Base de données	8
4.2.Serveur web	8
4.3.Les terminaux et APIs	9
4.3.1.Les terminaux	9
4.3.2.Les APIs	9
5.Architecture logicielle	10
5.1.Principes généraux	10

5.1.1.Les couches	10
5.2.Structure de la base de données	10
5.2.1.Modèle physique de données	10
5.2.2.Description des tables	11
6.Glossaire	12

1. VERSIONS

Auteur	Date	Description	Version
Yves Charpentier	14/02/2022	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 : Mise en place d'un nouveau système informatique pour l'ensemble des pizzerias du groupe.

Les éléments du présents dossiers découlent :

- du recueil des besoins et du cahier des charges fourni par le client
- de la présentation de la solution fonctionnelle au client lors de l'entrevue au lancement du projet
- de la présentation de la solution technique au développeur expérimenté

2.2.Références

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

1. **DCF** - 1.0 : Dossier de conception fonctionnelle
2. **DE** - 1.0 : Dossier d'exploitation
3. **PV** - 1.0 : Procès verbal de livraison

3.ARCHITECTURE TECHNIQUE

3.1.Composants généraux

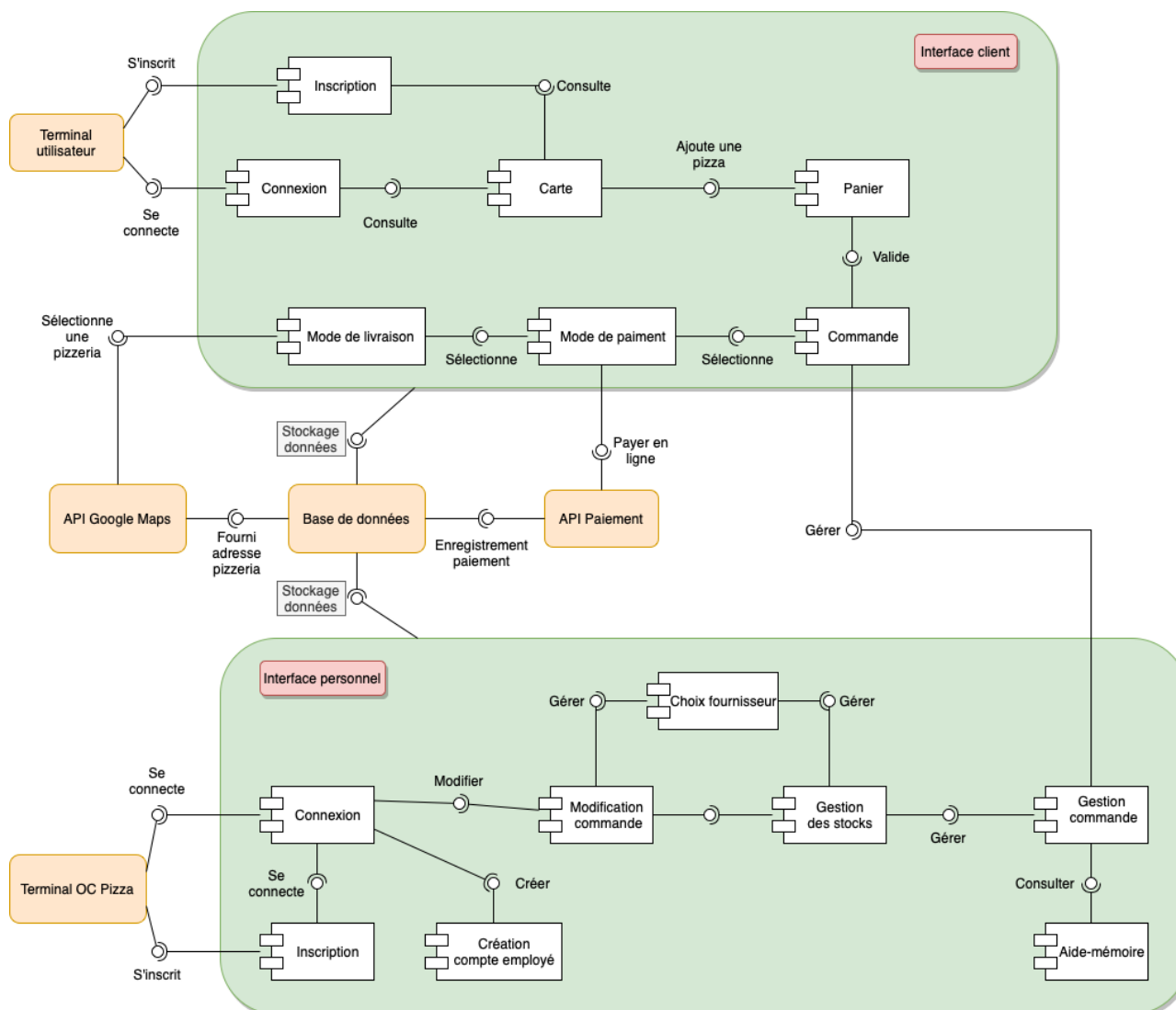


Diagramme UML de Composants

4.ARCHITECTURE DE DÉPLOIEMENT

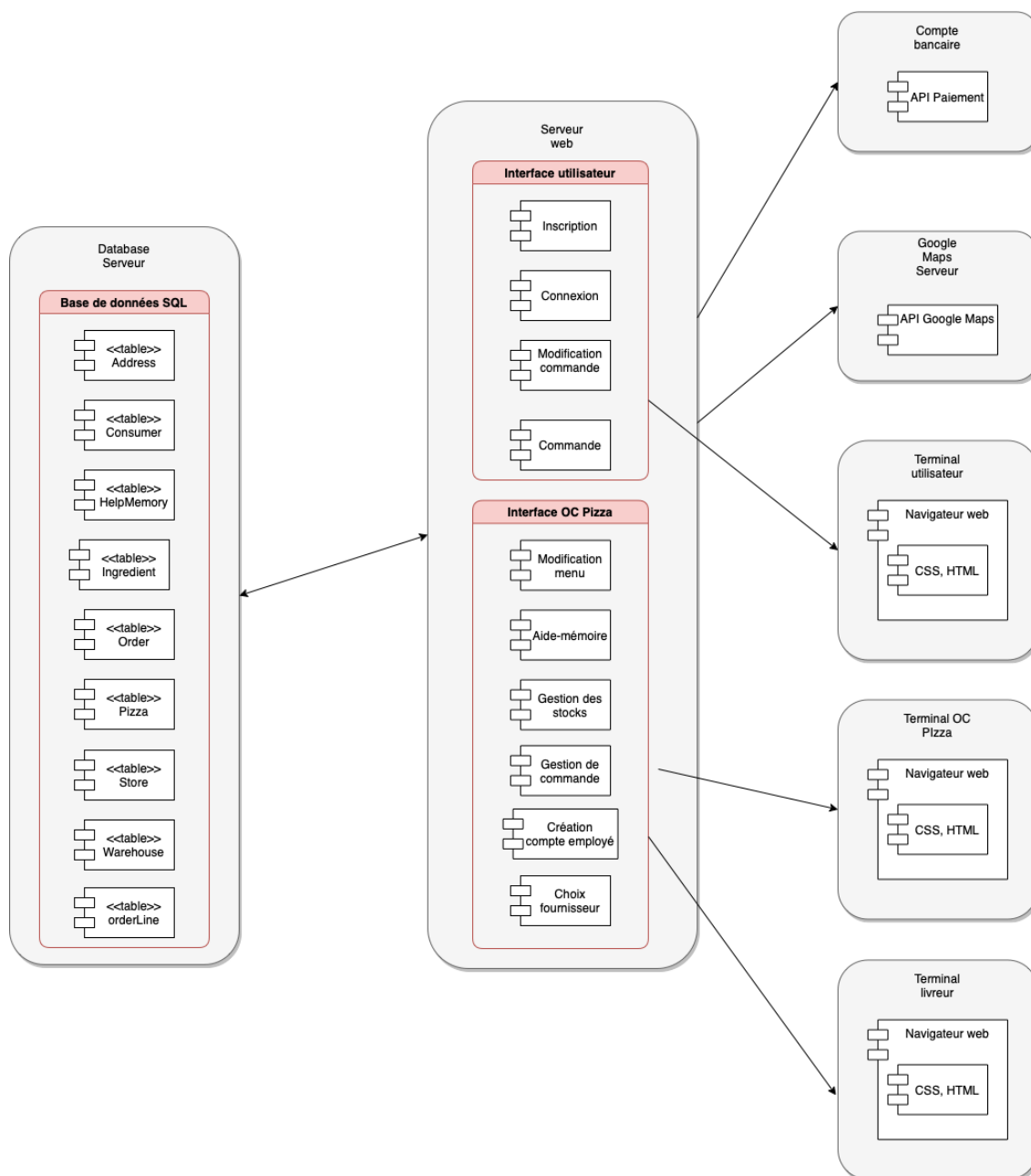


Diagramme UML de déploiement

DATABASE SERVEUR

Ici nous avons la Database Serveur qui va être sollicitée par le Serveur web dès lors qu'un client, un employé ou même un gérant va utiliser le navigateur web. Cette communication se fait évidemment dans les deux sens car elle va renvoyer des informations.

COMPTE BANCAIRE

De la même manière, lorsqu'un client souhaite payer en ligne, le composant API Paiement sera appelé par le Serveur web depuis l'Interface utilisateur.

GOOGLE MAPS SERVEUR

Quant au composant API Google Maps, il est sollicité par l'Interface utilisateur lorsque celui-ci effectue une commande et ainsi pour obtenir l'adresse postale d'un point de vente parmi ceux disponibles.

SERVEUR WEB

Également, l'Interface utilisateur fera appel au composant Navigateur web lors de la navigation sous forme de CSS et HTML.

Il en va de même pour l'Interface OC Pizza qui exploitera pareillement via le Terminal OC Pizza et le Terminal livreur, le composant Navigateur web.

4.1. Serveur de Base de données

Notre choix pour stocker la base de données s'est porté sur PostgreSQL.

4.2. Serveur web

Pour héberger le site internet, nous avons choisi Lightsail (Amazon). Les échanges avec le serveur de base de donnée se fera en TCP/IP.

4.3.Les terminaux et APIs

4.3.1.Les terminaux

TERMINAL UTILISATEUR

Le client se connecte au site via le navigateur web depuis son smartphone, tablette ou ordinateur. Le site web est développé en HTML et CSS. La communication avec le serveur web se réalisera grâce au protocole HTTPS.

TERMINAL OC PIZZA

Administrateurs comme employés pourront se connecter au site via le navigateur web depuis leur smartphone, tablette ou ordinateur de l'établissement. Le site web est développé en HTML et CSS. La communication avec le serveur web se réalisera grâce au protocole HTTPS.

TERMINAL LIVREUR

Le livreur se connecte au site via le navigateur web grâce à son smartphone. Le site web est développé en HTML et CSS. La communication avec le serveur web se réalisera grâce au protocole HTTPS.

4.3.2.Les APIs

API GOOGLE MAPS

Elle permet l'intégration de la carte sur le site web. La communication avec le serveur web se fait grâce au protocole HTTPS.

API STRIPE

C'est l'outil qui permettra d'effectuer des paiements en ligne directement sur le site web. La communication avec le serveur web se fait grâce au protocole HTTPS.

5. ARCHITECTURE LOGICIELLE

5.1. Principes généraux

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

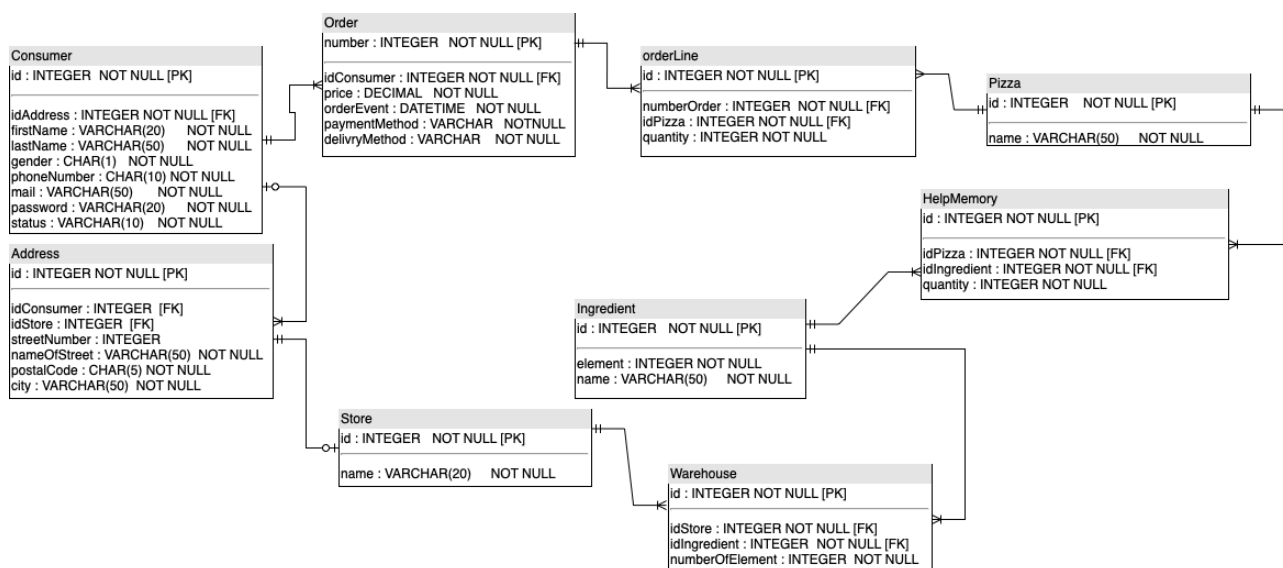
5.1.1. Les couches

L'architecture applicative est la suivante :

- Une couche **Contrôleur** : elle prend en charge les actions que l'utilisateur soumet
- Une couche **Modèle** : elle est appelée par le **Contrôleur** pour réaliser l'action fonctionnelle et éventuellement agir avec une base de données
- Une couche **Vue** : elle est mise à jour par le **Contrôleur** lorsque les actions du **Modèle** sont terminées.

5.2. Structure de la base de données

5.2.1. Modèle physique de données



5.2.2. Description des tables

Consumer : Sa clé primaire (PK) est « id », qui correspond à un numéro en auto incrémentation et ensuite attribué pour un consommateur de façon unique. Sa clé étrangère (FK) est « idAddress » et correspond à une adresse qu'un consommateur possède. La ligne « status » permet notamment de stocker les comptes employés mais également les autres consommateurs comme un client ou un visiteur.

Order : Sa clé primaire est « number » qui correspond à un numéro de commande en auto incrémentation. Sa clé étrangère « idConsumer » est le numéro d'identification d'un consommateur. Toutes les lignes de cette table doivent contenir une valeur.

orderLine : C'est une table d'association entre les tables **Order** et **Pizza**. Sa clé primaire « id » est le numéro d'une ligne de commande. Sa clé étrangère « numberOrder » est le lien avec la ligne « number » de la table **Order**. Sa deuxième clé étrangère « idPizza » est le numéro d'une pizza qui apparaîtra dans la ligne de commande grâce à la ligne « id » de la table **Pizza**.

Pizza : Sa clé primaire « id » est l'identifiant unique d'un article et est utilisée pour faire référence à la clé étrangère « idPizza » de la table **orderLine**. Également, une pizza peut être présent dans plusieurs lignes de commandes, mais une ligne de commande ne contient qu'une seule pizza (one-to-many).

HelpMemory : C'est une table d'association entre les tables **Pizza** et **Ingredient**. Sa clé primaire « id » permet de faire le lien avec la ligne « id » de la table **Pizza** en many-to-one, car un aide-mémoire n'est prévu que pour une seule pizza, mais l'id d'une pizza peut se retrouver dans plusieurs aide-mémoires.

Ingredient : Sa clé primaire « id » est associé en one-to-one à la ligne « idIngredient » de la table **HelpMemory**. La ligne « element » permet de quantifier le nombre d'ingrédients utilisés dans la composition d'une pizza, et sera utilisé pour le stockage.

Warehouse : C'est une table d'association entre les tables **Ingredient** et **Store**. Sa clé primaire « id » permet d'identifier pour un établissement le numéro du stockage. Quant à la ligne « numberOfElement » permet de comptabiliser le nombre d'éléments pour chaque ingrédient.

Store : Sa clé primaire « id » permet d'identifier chaque établissement de la marque. La clé étrangère « idWarehouse » servira pour savoir quel stock appartient à quel établissement. La ligne « name » attribue un nom à l'établissement et peut contenir jusqu'à 20 caractères.

Address : Sa clé primaire « id » sert d'identification pour toutes adresses (consommateurs, établissements). Sa clé étrangère « idConsumer » servira à identifier un consommateur dans la liste des adresses ; elle est en relation avec la table **Consumer** (many-to-one optional) car un consommateur peut posséder plusieurs adresses mais une adresse est utilisé pour un consommateur de façon optionnelle. Sa deuxième clé étrangère « idStore » permet quant à elle une relation one-to-one optional avec la table **Store** qui servira à identifier l'adresse de chaque établissement. La ligne « streetNumber » peut être NULL car le numéro n'est pas obligatoire, cependant, les lignes « nameOfStreet », « postalCode » et « city » contiennent forcément une valeur et ont un nombre de caractères limités.

6. GLOSSAIRE

Terme	Définition
API	Acronyme d'Application Programming Interface, que l'on traduit en français par interface de programmation applicative ou interface de programmation d'application.
HTTPS	Hypertext Transfer Protocol Secure : version sécurisée du langage informatique http, un protocole de communication qui permet la liaison entre un client et un <u>serveur</u> . En français, le terme se traduit par protocole de transfert hypertexte sécurisé.
TCP/IP	Transmission Control Protocol/ <u>Internet Protocol</u> réunit les deux protocoles TCP et IP. Il s'agit donc d'une suite de protocoles associée au domaine d'Internet pour lequel elle facilite le transfert de données.