

OC PIZZA



Spécifications techniques

Préparé pour : **OC Pizza**

Préparé par : Yves Charpentier, Développeur iOS, IT Consulting & Development

28 janvier 2022

Table des matières

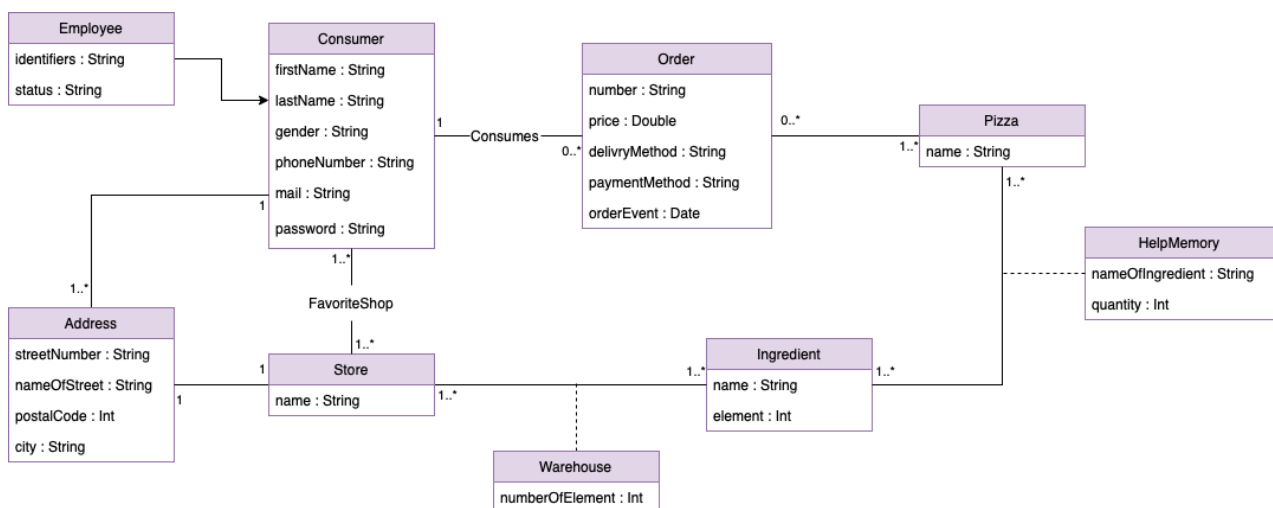
LE MODÈLE FONCTIONNEL	3
Description	3
Diagramme de classe (UML)	3
Description des classes	3
LE MODÈLE PHYSIQUE DE DONNÉES	4
Le diagramme	4
Description des tables	5
LES COMPOSANTS	7
Le diagramme de composant	7
ARCHITECTURE DE DÉPLOIEMENT	8
Le diagramme de déploiement	8
Description du diagramme	9

LE MODÈLE FONCTIONNEL

Description

Il comprend toutes les classes qui serviront pour créer le site internet du groupe OC Pizza.

Diagramme de classe (UML)



Description des classes

Consumer : Elle regroupe tous les consommateurs (sur place, en ligne, par téléphone ou en borne) avec leurs descriptions et comptes clients, s'ils en possèdent un. Elle est associée à la classe *Order*.

Employee : Elle comprend les différents employés du groupe (pizzaiolos, livreurs) et hérite de la classe mère *Consumer*.

Order : Elle comprend tous les détails d'une commande (date, heure, numéro de commande, etc...). Elle est associée aux classes *Consumer* et *Pizza*.

Pizza : Elle permet de récupérer le nom d'un article. Elle est associée aux classes *Order* et *Ingredient*.

HelpMemory : C'est une classe d'association qui sert d'aide mémoire pour le pizzaiolo et qui détaille le nombre d'ingrédients utilisés avec leur nom. Elle est associée aux classes *Pizza* et *Ingredient*.

Ingredient : Elle contient les informations sur les ingrédients qui seront nécessaire pour la réalisation d'une pizza. Elle est associée aux classes *Pizza* et *Store*.

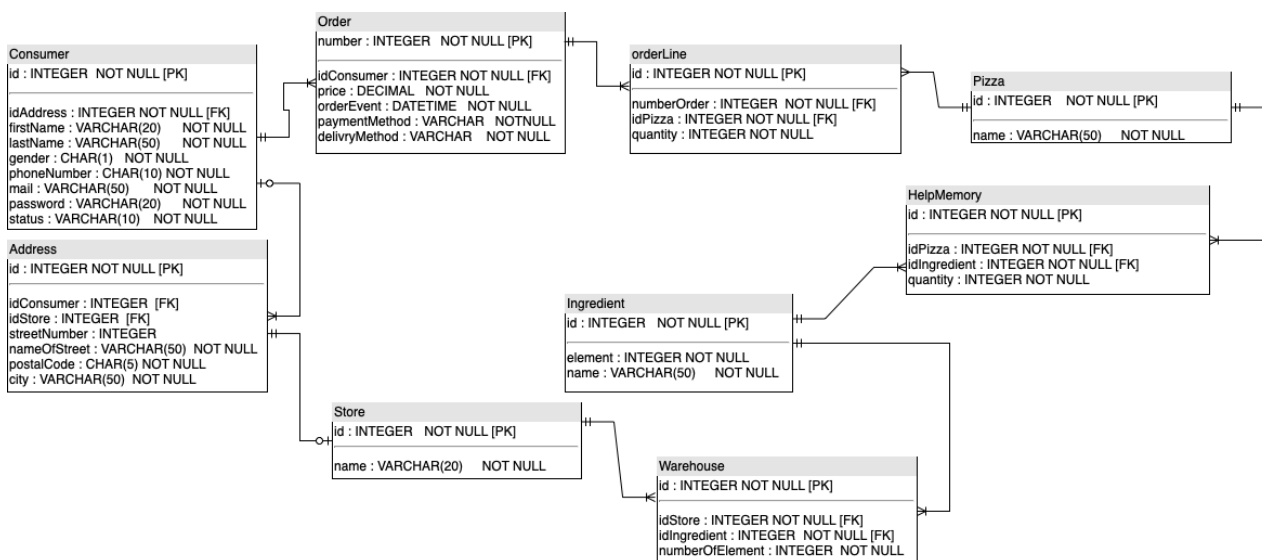
Warehouse : Cette classe d'association comptabilise le nombre d'ingrédients par établissement. Elle est associée aux classes *Store* et *Ingredient*.

Store : Elle attribut un nom à un établissement. Le groupe en possède actuellement 5. Le lien avec *Consumer* permet également d'avoir une information supplémentaire : l'établissement préféré d'un consommateur. Elle est associée aux classes *Ingredient*, *Warehouse*, *Address* et *Consumer*.

Address : Elle définit toutes les informations nécessaires que comporte une adresse postale. Elle est associée aux classes *Store* et *Consumer*.

LE MODÈLE PHYSIQUE DE DONNÉES

Le diagramme



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.

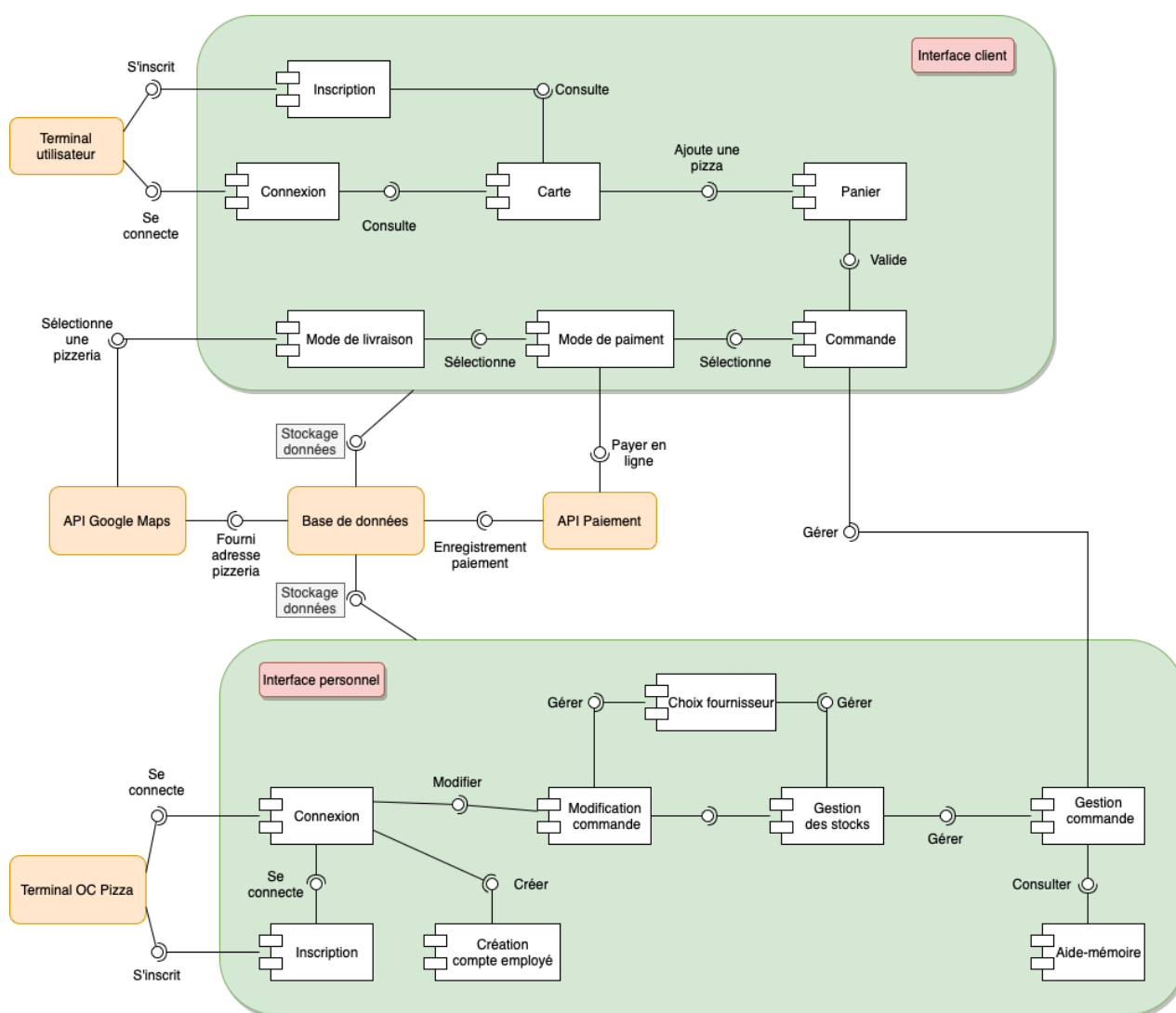
Warehouse : C'est une table d'association entre les tables **Ingredient** et **Store**. Sa clé primaire « id » permet d'identifier pour un établissement le numéro du stockage. Sa clé étrangère « idIngredient » permet de faire le lien (many-to-one) avec la table **Ingredient**. Quant à la ligne « numberOfElement » permet de comptabiliser le nombre d'éléments pour chaque ingrédient.

Store : Sa clé primaire « id » permet d'identifier chaque établissement de la marque. La 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.

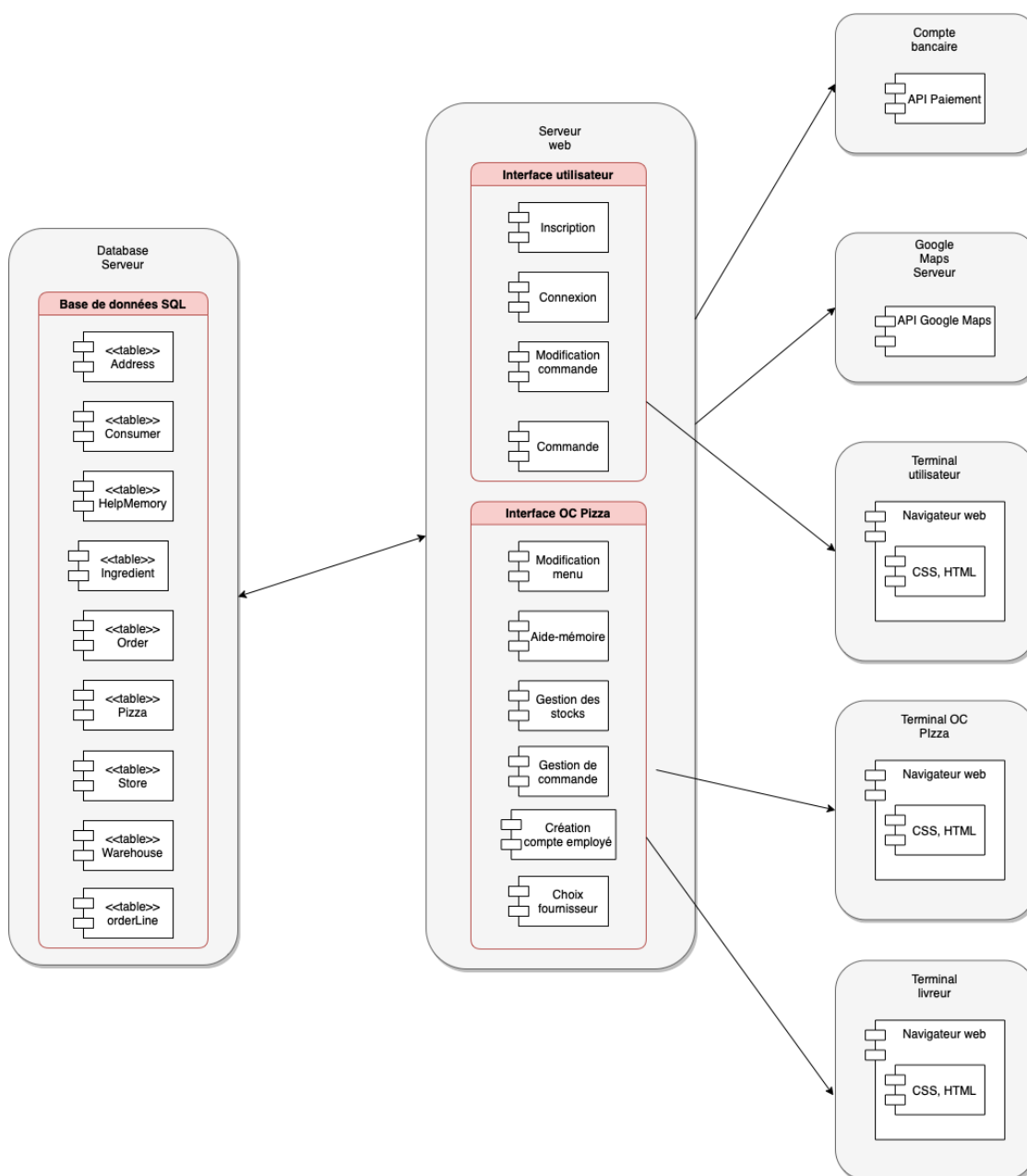
LES COMPOSANTS

Le diagramme de composant



ARCHITECTURE DE DÉPLOIEMENT

Le diagramme de déploiement



Description du diagramme

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*.