10 NOVEMBRE 2021

OC PIZZA

SPECIFICATION TECHNIQUE

1.INTRODUCTION

1.1 OBJET DU DOCUMENT

Le présent document constitue le dossier de conception technique de la future solution développée pour OC Pizza.

Il met en évidence le modèle fonctionnel du système ainsi que le modèle physique de données qui servira à la conception de la base de données OC Pizza.

Les composants internes et externes seront aussi analysés afin d'expliquer les liens du système avec les applications externes (Google Maps, Banque).

L'architecture de déploiement sera abordée pour identifier les éléments matériels et les éléments logiciels qui leurs sont rattachés.

Ces spécifications permettent de décrire comment la solution proposée sera implémentée d'un point de vue technique.

1.2 CONTEXTE

Les éléments du présent dossier se réfèrent au cahier des charges reçu précédemment.

« OC Pizza » est un jeune groupe de pizzeria en plein essor et spécialisé dans les pizzas livrées ou à emporter. Il compte déjà 5 points de vente et prévoit d'en ouvrir au moins 3 de plus d'ici la fin de l'année. Un des responsables du groupe a pris contact avec vous afin de mettre en place un système informatique, déployé dans toutes ses pizzerias et qui lui permettrait notamment :

- d'être plus efficace dans la gestion des commandes, de leur réception à leur livraison en passant par leur préparation ;
- de suivre en temps réel les commandes passées et en préparation ;
- de suivre en temps réel le stock d'ingrédients restants pour savoir quelles pizzas sont encore réalisables ;
- de proposer un site Internet pour que les clients puissent :
  - passer leurs commandes, en plus de la prise de commande par téléphone ou sur place
  - payer en ligne leur commande s'ils le souhaitent – sinon, ils paieront directement à la livraison
  - modifier ou annuler leur commande tant que celle-ci n'a pas été préparée
- de proposer un aide mémoire aux pizzaiolos indiquant la recette de chaque pizza
- d'informer ou notifier les clients sur l'état de leur commande

1.3 ENJEUX ET OBJECTIFS

Ce document est la deuxième étape d'une démarche qui se réalise, en collaboration avec OC Pizza, dans l'optique de répondre au plus près à ses besoins et à ceux de ses clients utilisateurs.

Les réunions à venir nous permettront de confirmer les spécifications techniques de la solution, en accord avec les spécifications fonctionnelles validées (cf. Dossier de spécifications fonctionnelles joint).

2 - LE DOMAINE FONCTIONNEL

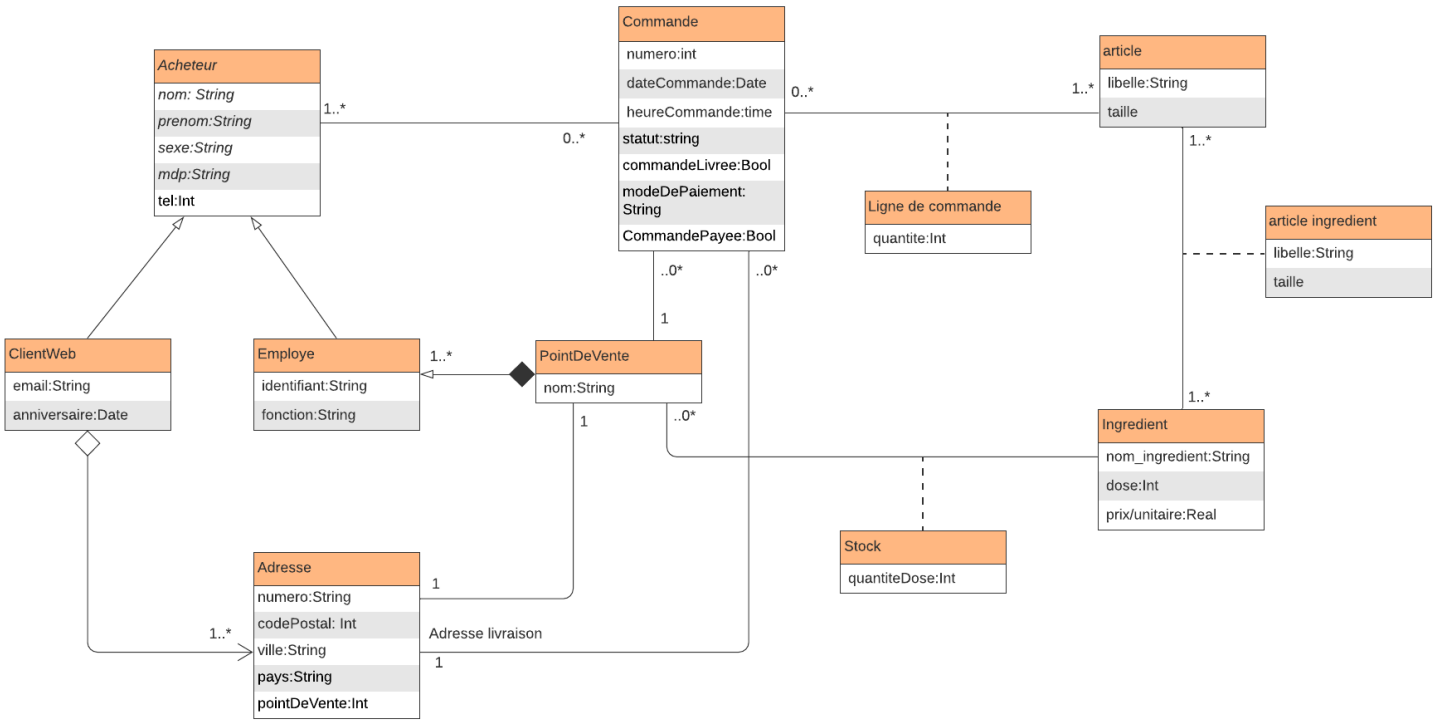
2.1 - DESCRIPTION

Le domaine fonctionnel du système OC Pizza établit l'ensemble des classes liées entre elles qui serviront de support à la programmation en PHP et à la création du Modèle Physique de Données.

Le domaine fonctionnel est représenté par un diagramme UML, ici le Diagramme de classes.

2.2 - DIAGRAMME DE CLASSES UML D'OC PIZZA

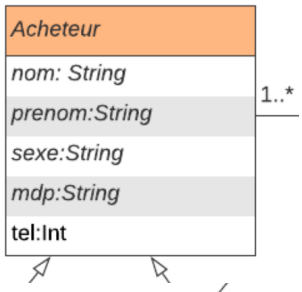
Le diagramme de classes UML (Class Diagram) représente ainsi l'organisation de l'information grâce aux différentes classes et aux liens entre-elles.



2.3 - DESCRIPTIF DES CLASSES PRINCIPAL

CLASS ACHETEUR

Cette classe regroupe les attributs communs à tous les « Acheteurs ».

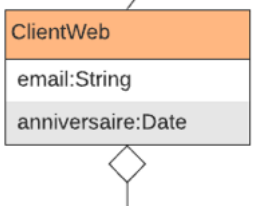


Cette **classe mère est associée aux classes filles ClientWeb et Employé.**

Elle est aussi associée à la classe Commande (many-to-many), car l'«Acheteur» saisit une commande. Chaque Acheteur peut faire zéro ou une infinité de commandes (cardinalité 0..*) et chaque commande peut être attribuée à un ou aucun Acheteur dans le cas d'une suppression de compte (cardinalité 1..*)

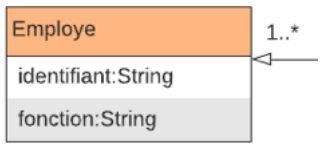
CLIENT WEB

Cette classes héritent de la classe Acheteur. C'est une **spécialisations de l'« Acheteur »**, association one-to-one.



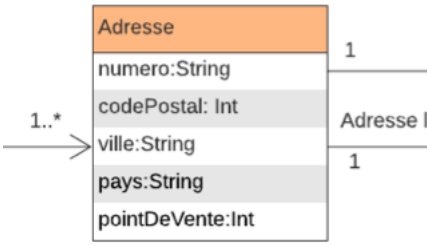
CLIENT EMPLOYE

La classe **Employé** est associée à la classe **PointDeVente** par une association de composition (one to many) : les utilisateurs salariés des différentes pizzerias du groupe OC Pizza sont affectés à un point de vente.



CLASS ADRESSE

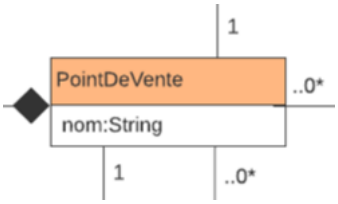
cette classe regroupe tous les attributs des adresses soit des commandes soit des points de vente.



La classes **Adresse** a une association one to one avec la classe **PointDeVente** (chaque pizzeria possédant une adresse).

CLASS POINT DE VENTE

Cette classe a pour seul attribut nom. Elle a des associations avec les classes **Employé, Adresse, Commande et Ingredient**.



Les deux premières associations ont été décrites ci-dessus. L'association avec **Commande**, de type one-to-many, relie la commande à un point de vente qui effectuera ladite commande.

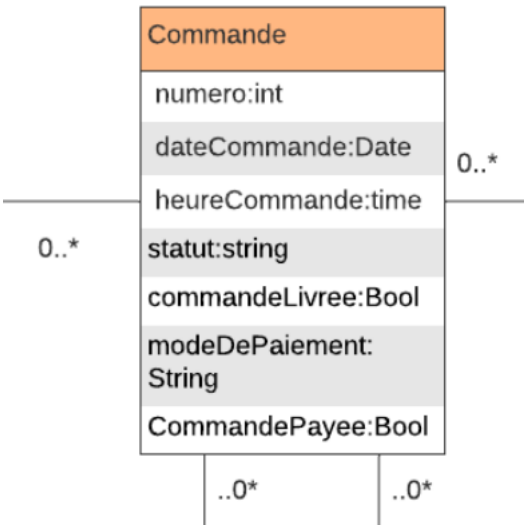
Il y a donc aucune ou une infinité de commandes passées à une pizzeria donnée du groupe (cardinalité 0..*) et chaque commande n'est affectée qu'a un seul point de vente (cardinalité 1).

CLASS COMMANDE

Cette classe regroupe les attributs nécessaires pour une commande de pizza(s). Elle a des associations avec les classes **Acheteur, PointDeVente, Adresse et Article**.

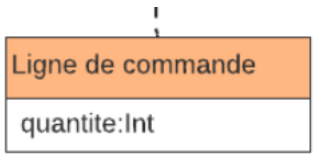
Les trois premières associations ont été décrites ci-dessus.

L'association avec Article (pizza) est de type many-to-many, avec la cardinalité 0..*, car pour un article il peut y avoir 0 ou une infinité de commandes ; et la cardinalité 1..*, car pour une commande il y a forcément au moins un article.



CLASS LIGNE COMMANDE

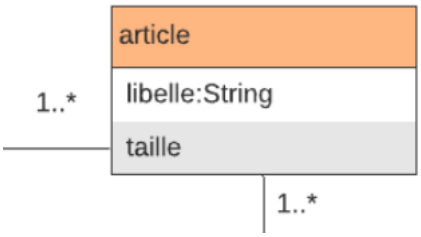
LigneCommande est une classe d'association, permettant d'ajouter l'attribut quantité à l'association entre les classes **Commande et Article** (type many-to-many).
Cette classe permet de comptabiliser pour une commande combien d'articles de tel type font partie du panier d'achat.



CLASS ARTICLE

Cette classe regroupe les attributs d'une pizza : le libellé et la taille.
Elle a des associations avec les classes **Commande et Ingredient.**
La première association a été décrite ci-dessus.

L'association avec **Ingredient** est de type many-to-many, avec la cardinalité 1..*, car un ingrédient se retrouve dans au moins une pizza ou article ; et la cardinalité 1..*, car une pizza ou article contient au moins 1 ingrédient.

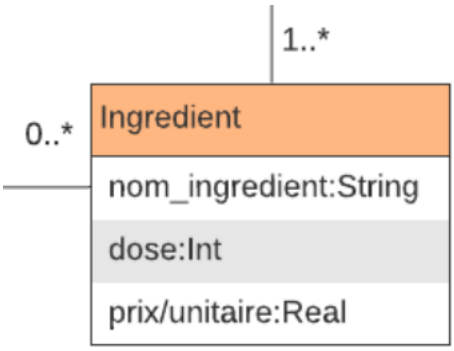


Pour la taille (size) des pizzas, la taille « Grand format » est le double de la taille « Format normal » et aura donc le double de dose de tous les ingrédients.

CLASS INGREDIENT

Cette classe regroupe les attributs d'un article. Elle a des associations avec les classes **Article** (décrite ci-dessus) et **PointDeVente.**
L'association avec **PointDeVente** est de type **many-to-many** avec la **cardinalité 0..* dans les deux sens**.

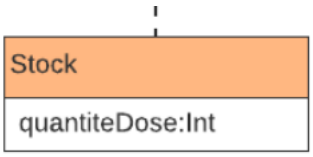
Chaque pizzeria (5 aujourd'hui, potentiellement plus dans le futur) stocke (Stocks) tous les types d'ingrédients qui sont nécessaires à la préparation des commandes.
Un ingrédient peut n'être dans aucune pizzeria en cas de rupture de stock générale, de même, dans une pizzeria il peut n'y avoir aucun ingrédient.



CLASS STOCK

Stock est une classe d'association, permettant d'ajouter l'attribut quantiteDose à l'association entre les classes **Ingredient** et **PointDeVente** (type many-to- many).

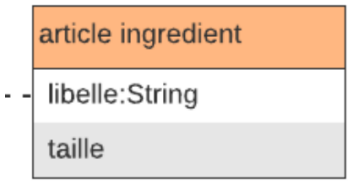
Cette classe permet de comptabiliser la quantité de doses de chaque ingrédient dans chaque pizzeria. Cette quantité peut être nulle en cas de rupture de stock.



CLASS ARTICLE INGREDIENT

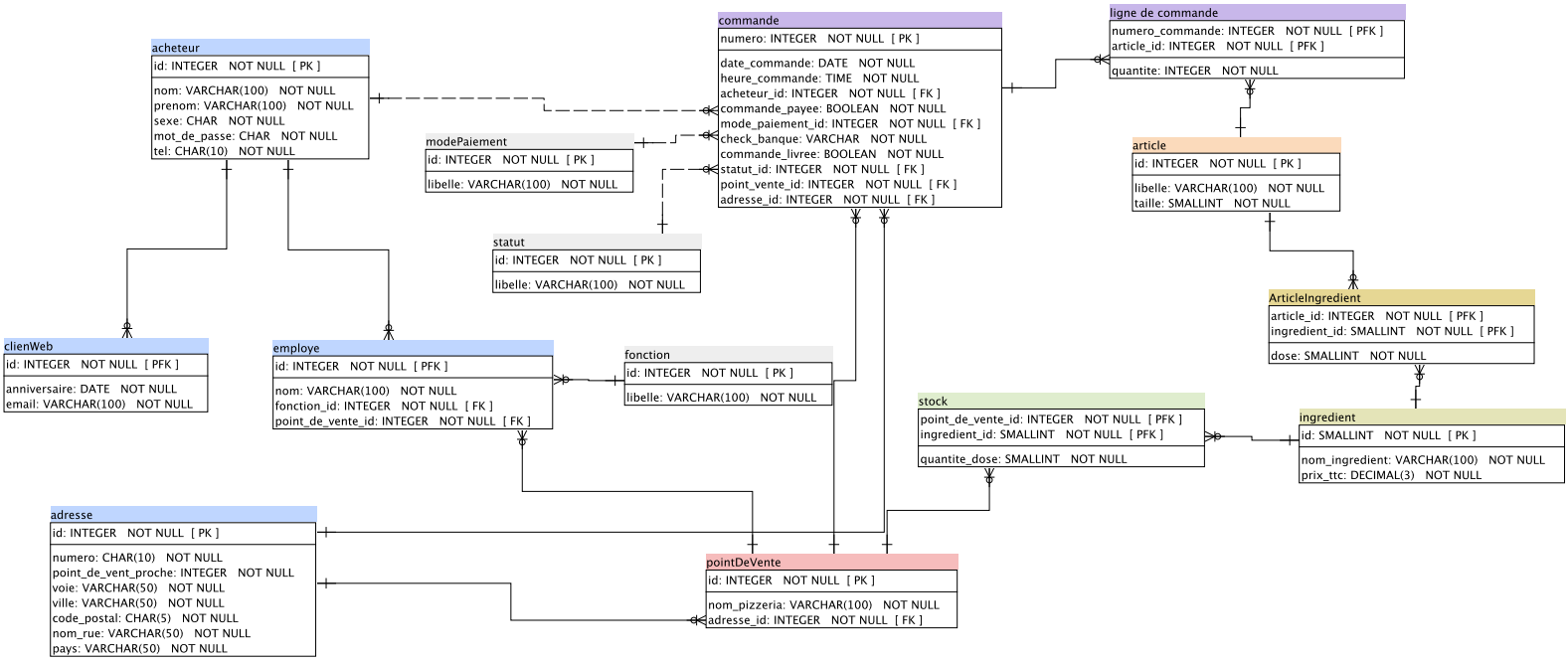
Article ingrédient est une classe d'association, permettant d'ajouter l'attribut quantiteDose à l'association entre les classes **Ingredient**

Cette classe permet de comptabiliser la quantité de doses de chaque ingrédient dans chaque pizzeria. Cette quantité peut être nulle en cas de rupture de stock.



3.1 - DESCRIPTION

Le **Modèle Physique de Données** (Physical Data Model) va permettre de modéliser dans le détail la base de données relationnelle d'OC Pizza.
Nous décrirons les tables et les liens entre-elles, les types de données des différentes colonnes de chaque table et les clés primaires et étrangères.



3.3 - DESCRIPTIF DES TABLES PRINCIPALES

3.3.1 - TABLE « ACHETEUR »

Cette table regroupe les informations de tous les « Acheteurs », en général. **Toutes les colonnes** de cette table sont renseignées **(NOT NULL)**.
La clé primaire est id, avec un **AUTO_INCREMENT**, et de type **INTEGER**. Pour les autres colonnes, le type est :

• **VARCHAR**, avec **50** caractères pour le **nom** et le **prenom** ;
• **CHAR**, avec **10** caractères pour **numero_telephone**, **1** pour **genre** (« H » ou « F »), et **16** caractères pour le **mot_de_passe**.

3.3.2 - TABLE « CLIENTWEB »

Cette table regroupe les informations spécifiques d'une **spécialisation** des « Acheteurs » que sont les « Clients Internet ».
La colonne **email (VARCHAR, 100**) est obligatoirement renseignée **(NOT NULL).**
La colonne **anniversaire** est de type **DATE.**

La colonne **acheteur_id est une clé étrangère** se référant à l'**id** correspondant dans la table **Acheteur**.

3.3.3 - TABLE « EMPLOYE »

Cette table regroupe les informations spécifiques d'une **spécialisation** des « Acheteurs » que sont les employés d'OC Pizza.
Ici, **toutes les colonnes sont à renseigner (NOT NULL).**
Le nom est un VARCHAR (50) et la fonction un INTEGER.

La colonne **acheteur_id** est **une clé étrangère** se référant à l'**id** correspondant dans la table **Acheteur.**

3.3.4 - TABLE « ADRESSE »

Les colonnes numero, voie, nom_rue, ville, code_postal, pays et point_de_vente_proche doivent être renseignées (NOT NULL).
La colonne **commentaire** n'est pas à renseigner obligatoirement. Elle permet d'ajouter des précisions par rapport à l'adresse (étage, code...).

La **clé primaire est id**, avec un **AUTO_INCREMENT**, et de type **INTEGER**. Pour les colonnes, le type de données est :

• **VARCHAR**, avec **150** caractères pour **nom_rue** ;
• **VARCHAR**, avec **50** caractères pour **voie, ville, pays** ;
• **CHAR**, avec **10** caractères pour **numero** et **5** pour **code_postal** ; • **VARCHAR**, avec **300** caractères pour **commentaire** ;

3.3.5 - TABLE « POINTDEVENTE »

La colonne nom, de type **VARCHAR**, avec **100** caractères, doit être renseignée **(NOT NULL)**.
La **clé primaire** est **id**, avec un **AUTO_INCREMENT**, de type **INTEGER.**
La colonne **adresse_id est une clé étrangère** se référant à l'**id** correspondant dans la table **Adresse**, afin de récupérer l'adresse de chaque pizzeria.

3.3.6 - TABLE « COMMANDE »

Toutes les colonnes, doivent être renseignées **(NOT NULL).**

La clé primaire est numero, avec un **AUTO_INCREMENT**, et de type **INTEGER**. Pour les colonnes, le type de données est :

• **DATE**, pour **date_commande** ;
• **TIME**, pour **heure_commande** ;
• **INTEGER**, pour **statut** ;
• **BOOLEAN**, pour **commande_livree** (true ou false si la commande est à livrer à domicile ou non) ; • **INTEGER**, pour **mode_paiement**;
• **BOOLEAN**, pour **commande_payee** (true ou false si le paiement a été effectué ou non) ;

Il y a **3 clés étrangères** :
• la colonne **acheteur_id** se référant à l'**id** correspondant dans la table **Acheteur**, afin de relier chaque commande à un « acheteur ».
• la colonne **point_de_vente_id** se référant à l'**id** correspondant dans la table **PointDeVente**, afin de relier chaque commande à une pizzeria du groupe. Le choix de la pizzeria dépend du résultat du calcul de géolocalisation entre le lieu de livraison et les différentes pizzerias : la plus proche sera choisie.
• la colonne **adresse_id** se référant à l'**id** correspondant dans la table **Adresse**, afin de relier chaque commande à l'adresse du client.

3.3.7 - TABLE « LIGNECOMMANDE »

La table **LigneCommande** est une **table d'association** (association many-to-many).
La colonne **quantite**, de type **INTEGER**, doit être renseignée (**NOT NULL**). C'est la quantité de chaque type d'article commandé.
Il y a **2 clés étrangères** :

• la colonne **numero_commande** de la table **Commande**, afin de relier chaque ligne de commande à une commande.
• la colonne **article_id** se référant à l'**id** de la table **Article**, afin de relier chaque ligne de commande à un article.

3.3.8 - TABLE « ARTICLE »

La colonne **libelle**, de type **VARCHAR**, avec **100** caractères, doit être renseignée **(NOT NULL).**

La colonne **taille**, de type **SMALLINT**, doit être renseignée **(NOT NULL)**. Elle prend la valeur **1** pour les articles de taille « Format standard» et **2** pour les articles de taille « Grand Format».
La clé primaire est id, avec un **AUTO_INCREMENT**, et de type **INTEGER**.

3.3.9 - TABLE « INGREDIENT »

La colonne **nom_ingredient**, de type **VARCHAR**, avec **100** caractères, doit être renseignée **(NOT NULL).**
La colonne **prix_ttc**, de type **DECIMAL**, de **3 chiffres dont 2 après la virgule**, doit être renseignée **(NOT NULL)**. **La clé primaire est id**, avec un **AUTO_INCREMENT**, et de type **INTEGER**.

3.3.10 - TABLE « ARTICLEINGREDIENT »

La table **ArticleIngredient**, n'apparaît pas dans le Diagramme de classes. C'est une **table d'association** qui doit être créée du fait de l'association many-to-many entre les tables **Article** et **Ingredient**.
La colonne dose, de type **SMALLINT**, doit être renseignée **(NOT NULL)** et prend la valeur **1** pour les articles « Format normal», et **2** pour les articles « Grand Format ». C'est le nombre de dose de chaque ingrédient que nécessite chaque article.

Il y a 2 clés étrangères :
• la colonne **article_id** se référant à l'id correspondant dans la table **Article**, afin de relier chaque ingrédient à un article donné.
• la colonne **ingredient_id** se référant à l'id correspondant dans la table **Ingredient**, afin de relier chaque article à un ingrédient donné.

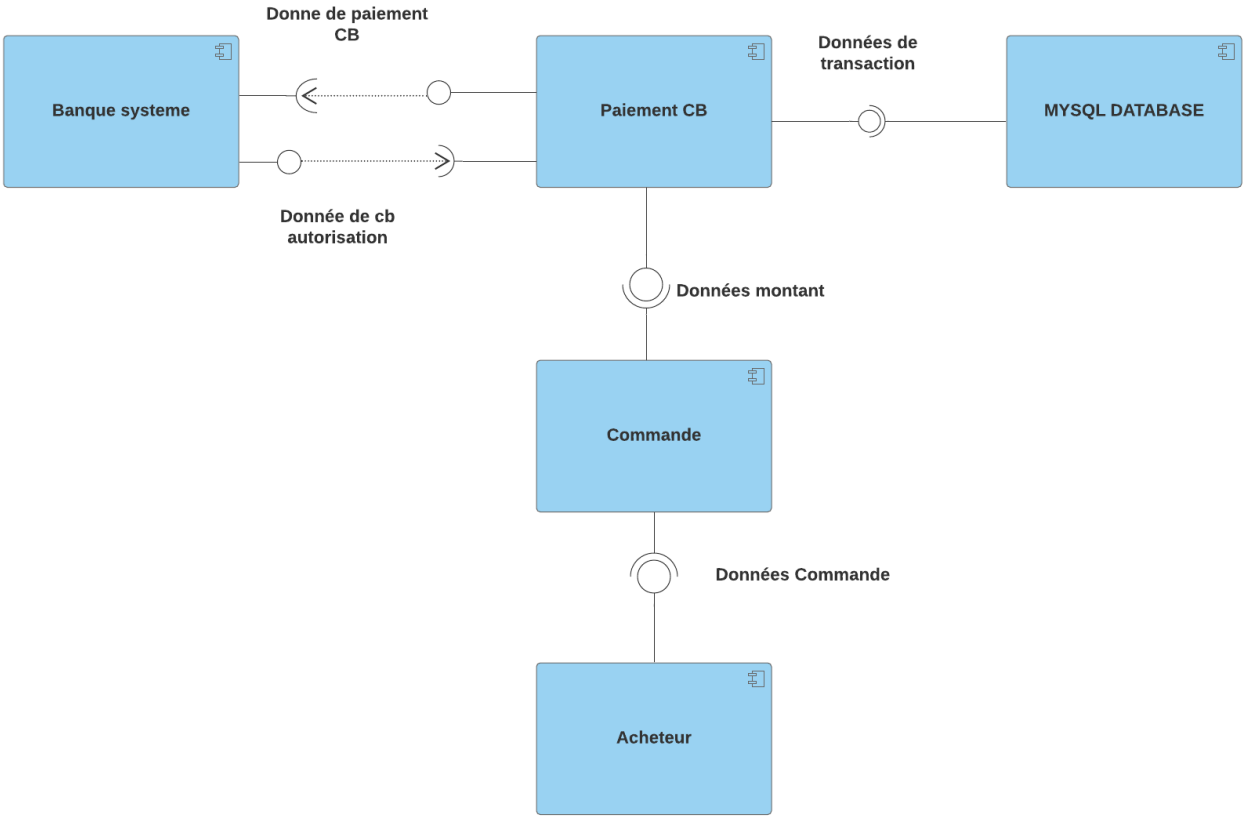
3.3.11 - TABLE « STOCK »

La table **Stock** est une table d'association (association many-to-many).
La colonne **quantite_dose**, de type **SMALLINT**, doit être renseignée mais peut être de valeur nulle (fin de stock d'un ingrédient). C'est la quantité de doses que chaque pizzeria possède de chaque ingrédient.
Il y a 2 clés étrangères :

• la colonne **point_de_vente_id** se référant à l'**id** correspondant dans la table **PointDeVente**, afin de relier chaque pizzeria donnée à un stock d'ingrédient.
• la colonne **ingredient_id** se référant à l'**id** correspondant dans la table **Ingredient**, afin de relier un stock d'ingrédient donné à chaque pizzeria.

4 - COMPOSANTS EXTERNES DU SYSTEME

4.1 PAIEMENT EN LIGNE



4.2 DESCRIPTION PAIEMENT EN LIGNE

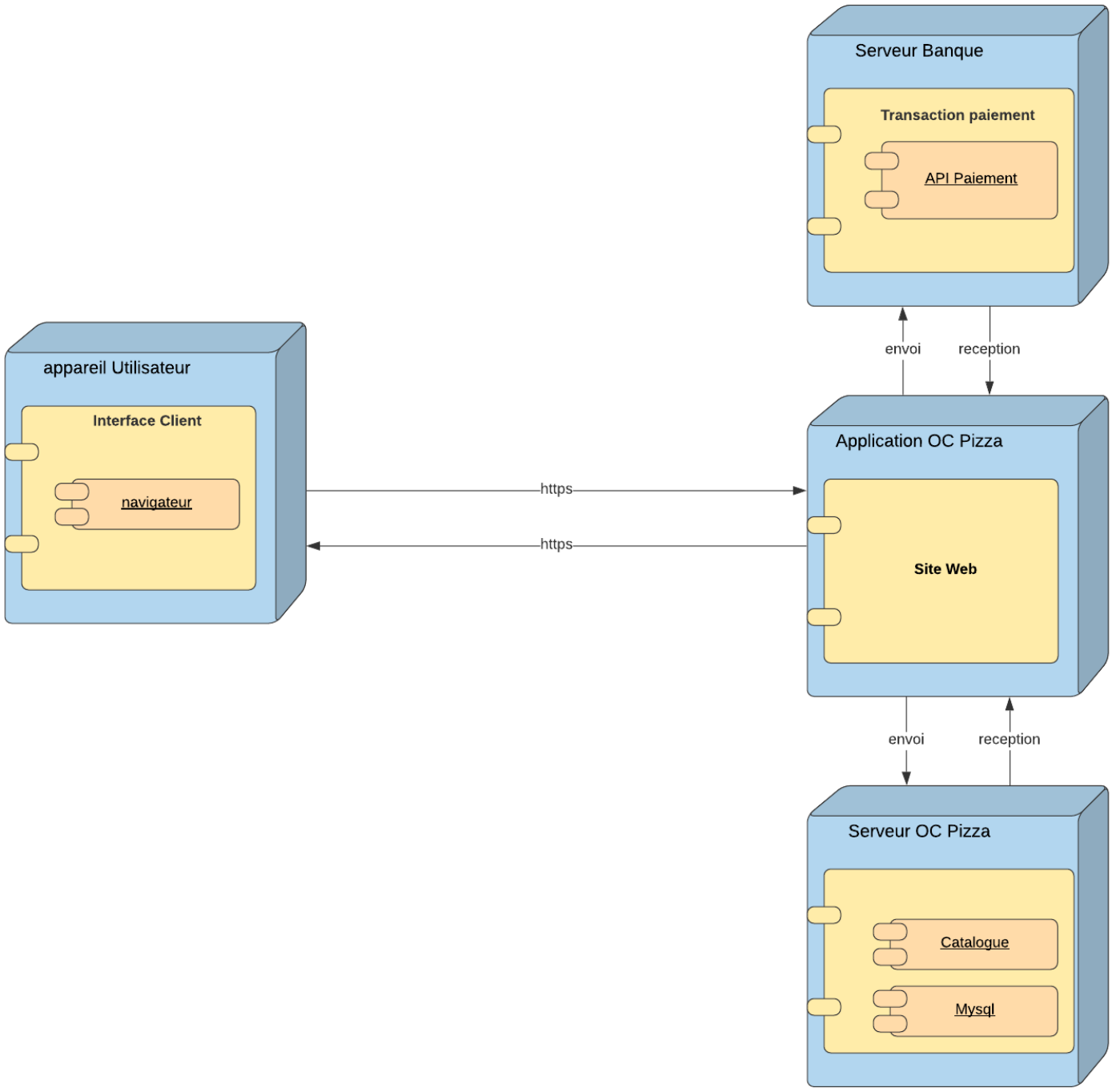
Le **composant Acheteur envoie au composant Commande les informations sur la commande** et celui-ci calcule le montant total. Ce montant est requis par le **composant Paiement CB** qui récupère aussi le nom du client.

Il échange avec le système bancaire des informations sur le client et le montant à régler. Le **client** aura directement accès au service de carte bancaire de la banque pour rentrer ses **coordonnées personnelles (numéro de carte, date d'expiration...).**
En retour, le **composant Banque système** renvoie au composant interne les l'information sur la réalisation du paiement.

Cette information sera **gardée en base de données, via le composant Mysql** Database, le système d'OC Pizza.

5. ARCHITECTURE DE DÉPLOIEMENT

5.1 DIAGRAMME DE DÉPLOIEMENT



5.2 DESCRIPTION DU DÉPLOIEMENT

-> ***Appareil Utilisateur*** indique l'ordinateur avec lequel l'utilisateur se connecte.
L'**Interface Client** représente l'interface utilisateur, à travers laquelle il peut accéder au site web d'OC Pizza (via le **Navigateur Web**).

-> ***Serveur Banque*** indique le serveur qui gère les requêtes pour les paiements en ligne, effectués grâce aux APIs des banques (**API Paiement**).
Le transfert des données entre les deux nœuds se fait grâce au protocole https.

-> ***Application OC Pizza*** indique l'ensemble de pages web qui composent le site web Oc Pizza, comme décrit par le composant **Site Web**.
Le transfert de données entre ce nœud et le nœud **Appareil Utilisateur** se fait grâce au protocole HTTPS.

-> ***Serveur OC Pizza*** indique le serveur chargé de gérer les requêtes des pages du site web.
Il repère les informations grâce à la base de données dédiée **Base de données OC Pizza** et les affiche à l'utilisateur sous forme de **Catalogue**. Les interactions avec la base de données sont assurées par **Mysql**. L'échange des données entre les nœuds **Application OC Pizza** et **Serveur OC Pizza** se fait toujours avec HTTPS.