OC Pizza

Concevez la solution technique d'un système de gestion de pizzeria

Définition du domaine fonctionnel et conception de l'architecture technique de la solution répondant aux besoins du client

Auteur	Date	Description	Version
C. Clarret	16/07/2020	Création du document	1.0

Table des matières

1. Eléments de contexte

« 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

Le client a déjà fait une petite prospection et les logiciels existants qu'il a pu trouver ne lui conviennent pas.

Le but de ce projet est de **définir le domaine fonctionnel** du futur système mais aussi de **concevoir l'architecture technique** de la solution.

Pour cela, nous avons :

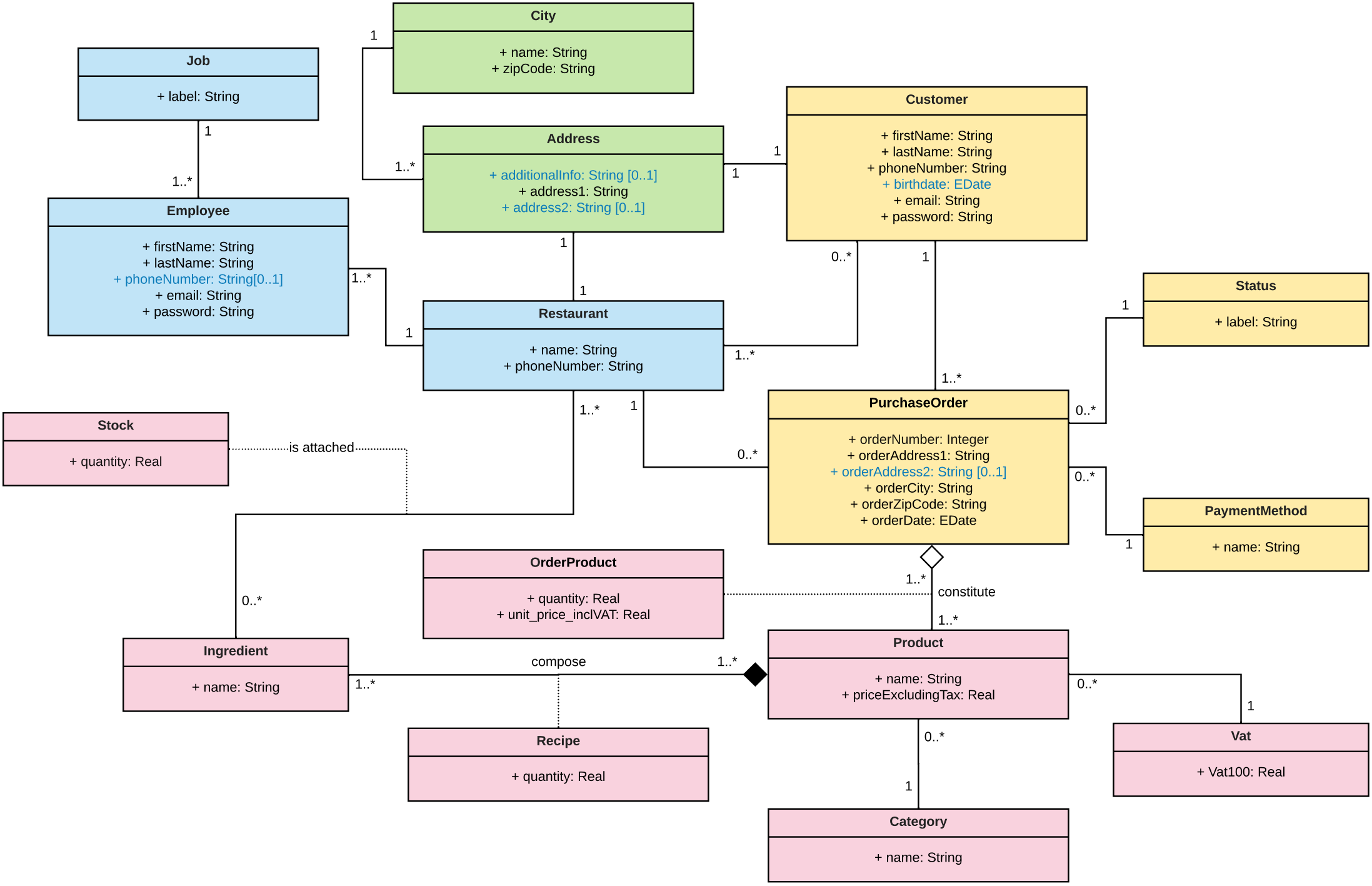
- décrit le domaine fonctionnel avec un diagramme de classe UML
- élaboré un diagramme de composants
- réalisé une diagramme de déploiement de ces composants
- puis élaboré un modèle physique de données

L'ensemble des documents de la conception est disponible à à ce lien :

https://github.com/CamClrt/P6-OC_Pizza

2. Description du domaine fonctionnel

Diagramme de classe :


City
+ name: String
+ zipCode: String
Job
+ label: String
Address
+ additionalInfo: String [0..1]
+ address1: String
+ address2: String [0..1]
Customer
+ firstName: String
+ lastName: String
+ phoneNumber: String
+ birthdate: EDate
+ email: String
+ password: String
Employee
+ firstName: String
+ lastName: String
+ phoneNumber: String[0..1]
+ email: String
+ password: String
Restaurant
+ name: String
+ phoneNumber: String
Status
+ label: String
Stock
+ quantity: Real
is attached
PurchaseOrder
+ orderNumber: Integer
+ orderAddress1: String
+ orderAddress2: String [0..1]
+ orderCity: String
+ orderZipCode: String
+ orderDate: EDate
PaymentMethod
+ name: String
OrderProduct
+ quantity: Real
+ unit_price_inclVAT: Real
constitute
Ingredient
+ name: String
compose
Product
+ name: String
+ priceExcludingTax: Real
Recipe
+ quantity: Real
Vat
+ Vat100: Real
Category
+ name: String

Relations entre les classes :

Le but des descriptions ci-dessous est d'expliquer les liaisons entre les différentes classes du diagramme. Il est important de noter que pour une uniformisation avec le modèle physique de données, les noms des classes ainsi que leurs attributs sont en anglais.

Zone bleue : équipe

Employee - Job :

- 'Employee' représente un employé du restaurant
- 'Job' représente la fonction occupée par l'employé

Un emploi est associé à un ou plusieurs employés et un employé occupe une et un seule fonction dans un restaurant.

Employee - Restaurant :

- 'Employee' représente un employé du restaurant
- 'Restaurant' représente une pizzeria du groupe

Un employé travaille pour un seul et unique restaurant et une pizzeria emploie un où plusieurs salariés.

Zone verte : localité

Address - City :

- 'Address' représente l'adresse postale d'un restaurant où d'un client
- 'City' représente la ville où est localisée le restaurant ou le client

Une adresse est rattachée à une et une seule ville et une ville peut être rattachée à une ou plusieurs adresses.

Address - Restaurant :

Une pizzeria est rattachée à une seule et unique adresse et réciproquement.

Address - Customer :

Un client est rattaché à une seule et unique adresse et réciproquement.

Zone jaune : clientèle

Customer - Restaurant :

- ‘Customer’ représente un client

Un client peut être rattaché à un ou plusieurs restaurant et une pizzeria peut avoir zéro ou plusieurs clients.

Customer - PurchaseOrder :

- ‘PurchaseOrder’ représente la commande d’un client

Un client a une ou plusieurs commande à son actif et une commande n’est rattachée qu’à un seul et unique client.

PurchaseOrder - Status :

- ‘Status’ représente l’état d’une commande à un moment ‘T’

Une commande possède un seul et unique état au moment ‘T’ et un statut peut avoir zéro ou plusieurs commandes rattachées.

PurchaseOrder - PaymentMethod :

- ‘PaymentMethod’ représente le moyen de paiement d’une commande

Une commande possède un moyen de paiement et un moyen de paiement peut avoir zéro ou plusieurs commandes rattachées.

Zone rouge : pizzeria

OrderProduct - PurchaseOrder :

- ‘OrderProduct’ représente la composition d’une commande

‘OrderProduct’ est ici ce qu’on appelle une table d’association. Elle permet de faire le lien entre la commande d’un client et le catalogue du restaurant. On y retrouve le détail de la commande soit le nombre de produits souhaité par le client ainsi que les prix payés.

Product - OrderProduct :

- ‘Product’ représente une pizza au catalogue du restaurant

Une commande est constituée d’une ou plusieurs pizzas au menu du restaurant.

Product - Vat :

- ‘Vat’ représente le taux de TVA appliqué sur un produit

Un produit est soumis à un seul et unique taux de TVA et un taux de TVA peut être rattaché à zéro ou plusieurs produits.

Product - Category :

- ‘Category’ représente la catégorie à laquelle appartient un produit.

Une catégorie peut regrouper aucun où plusieurs produits et un produit est classifié sous une seule et unique catégorie.

Product - Recipe :

- ‘Recipe’ représente la recette d’un produit

‘Recipe’ est une table d’association. Elle permet de faire le lien entre le produit fini et les ingrédients qui le composent. On y retrouve le détail des quantités d’ingrédients requis pour élaborer ce produit.

Ingredient - Recipe :

- ‘Ingredient’ représente un ingrédient stocké par un restaurant

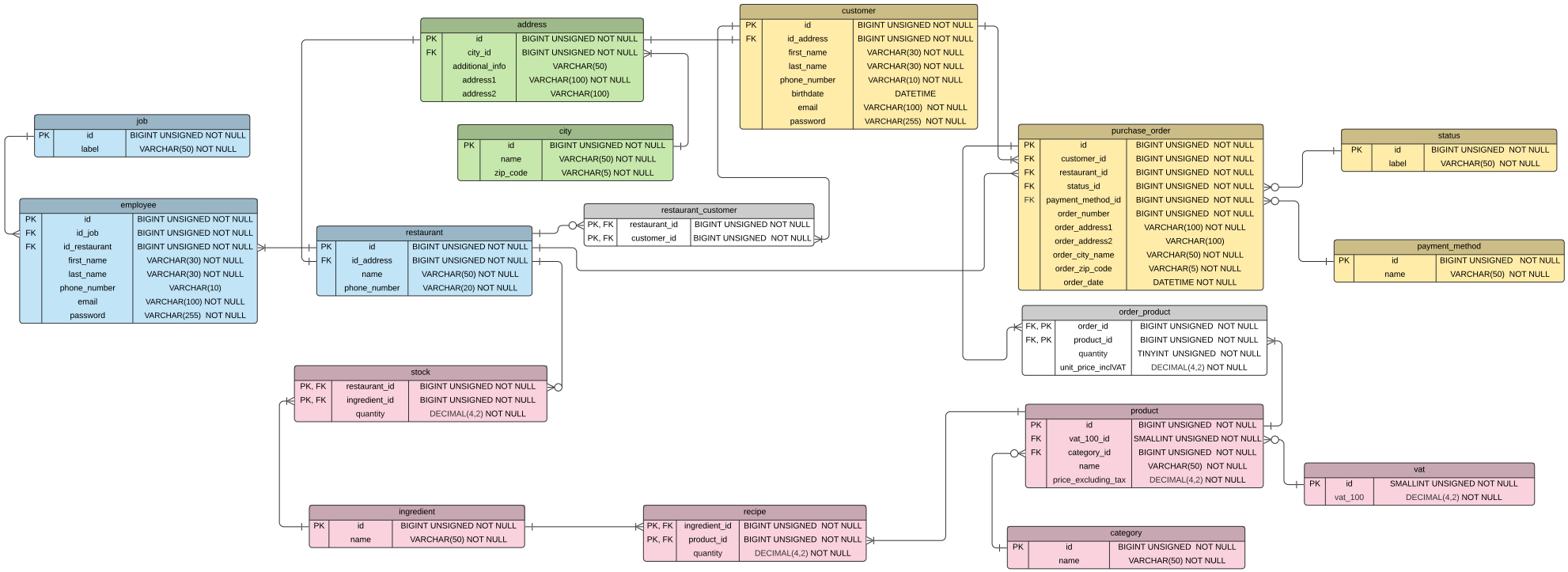
Un ou plusieurs ingrédients composent un produit au menu du restaurant. Un produit est lui-même composé d’un où plusieurs ingrédients.

Ingrédient - Stock :

- ‘Stock’ représente la quantité d’ingrédients possédée par un restaurant

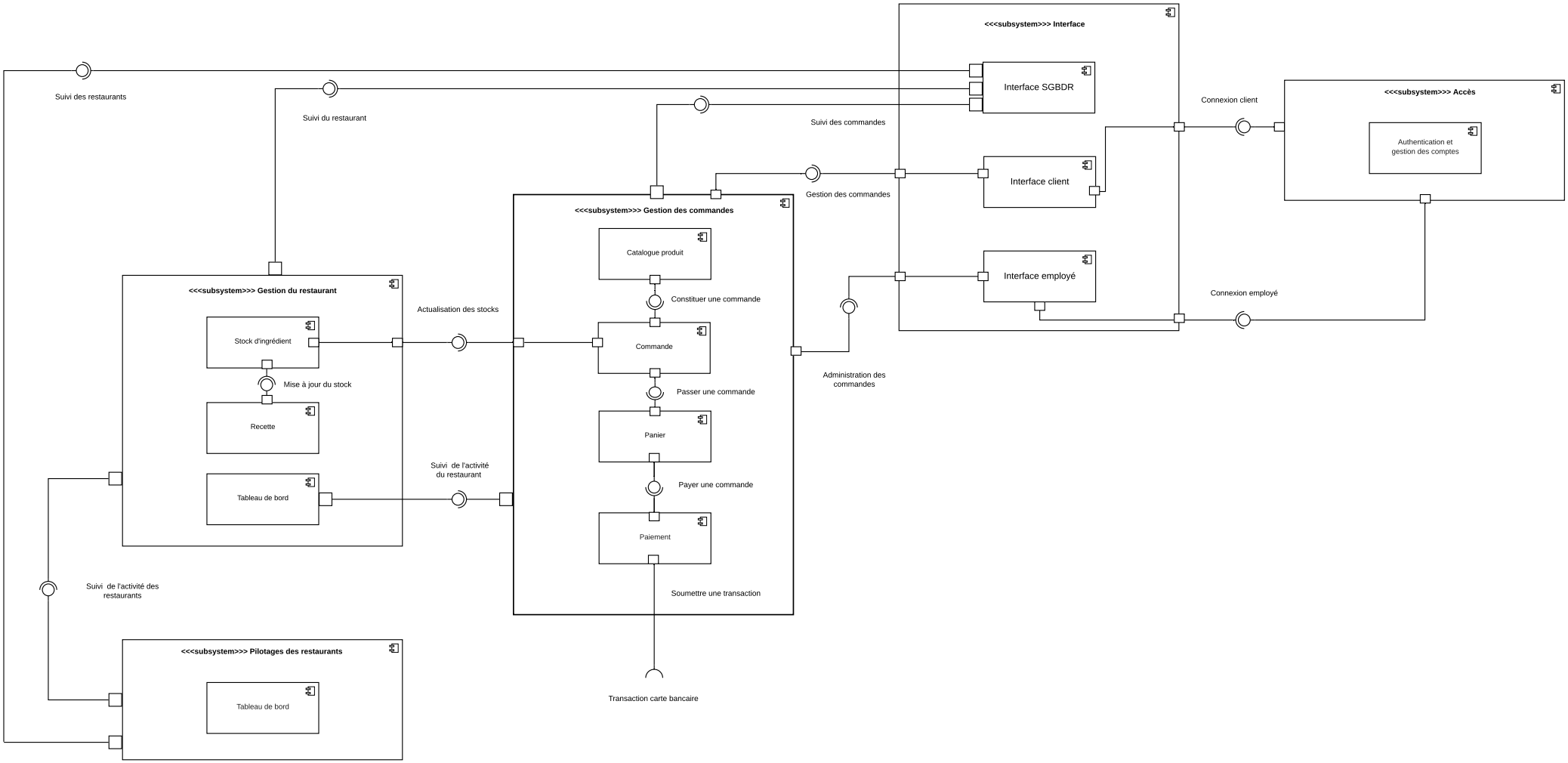
Stock est une table d’association. Elle permet d’établir le lien entre les ingrédients et les restaurants. On y retrouve donc le détail des quantités d’ingrédients possédées pour chaque restaurant.

4. Modèle physique de données


address
PK id BIGINT UNSIGNED NOT NULL
FK city_id BIGINT UNSIGNED NOT NULL
additional_info VARCHAR(50)
address1 VARCHAR(100) NOT NULL
address2 VARCHAR(100)
customer
PK id BIGINT UNSIGNED NOT NULL
FK id_address BIGINT UNSIGNED NOT NULL
first_name VARCHAR(30) NOT NULL
last_name VARCHAR(30) NOT NULL
phone_number VARCHAR(10) NOT NULL
birthdate DATETIME
email VARCHAR(100) NOT NULL
password VARCHAR(255) NOT NULL
job
PK id BIGINT UNSIGNED NOT NULL
label VARCHAR(50) NOT NULL
city
PK id BIGINT UNSIGNED NOT NULL
name VARCHAR(50) NOT NULL
zip_code VARCHAR(5) NOT NULL
purchase_order
PK id BIGINT UNSIGNED NOT NULL
FK customer_id BIGINT UNSIGNED NOT NULL
FK restaurant_id BIGINT UNSIGNED NOT NULL
FK status_id BIGINT UNSIGNED NOT NULL
FK payment_method_id BIGINT UNSIGNED NOT NULL
order_number BIGINT UNSIGNED NOT NULL
order_address1 VARCHAR(100) NOT NULL
order_address2 VARCHAR(100)
order_city_name VARCHAR(50) NOT NULL
order_zip_code VARCHAR(5) NOT NULL
order_date DATETIME NOT NULL
status
PK id BIGINT UNSIGNED NOT NULL
label VARCHAR(50) NOT NULL
payment_method
PK id BIGINT UNSIGNED NOT NULL
name VARCHAR(50) NOT NULL
employee
PK id BIGINT UNSIGNED NOT NULL
FK id_job BIGINT UNSIGNED NOT NULL
FK id_restaurant BIGINT UNSIGNED NOT NULL
first_name VARCHAR(30) NOT NULL
last_name VARCHAR(30) NOT NULL
phone_number VARCHAR(10)
email VARCHAR(100) NOT NULL
password VARCHAR(255) NOT NULL
restaurant
PK id BIGINT UNSIGNED NOT NULL
FK id_address BIGINT UNSIGNED NOT NULL
name VARCHAR(50) NOT NULL
phone_number VARCHAR(20) NOT NULL
restaurant_customer
PK, FK restaurant_id BIGINT UNSIGNED NOT NULL
PK, FK customer_id BIGINT UNSIGNED NOT NULL
order_product
FK, PK order_id BIGINT UNSIGNED NOT NULL
FK, PK product_id BIGINT UNSIGNED NOT NULL
quantity TINYINT UNSIGNED NOT NULL
unit_price_inclVAT DECIMAL(4,2) NOT NULL
stock
PK, FK restaurant_id BIGINT UNSIGNED NOT NULL
PK, FK ingredient_id BIGINT UNSIGNED NOT NULL
quantity DECIMAL(4,2) NOT NULL
product
PK id BIGINT UNSIGNED NOT NULL
FK vat_100_id SMALLINT UNSIGNED NOT NULL
FK category_id BIGINT UNSIGNED NOT NULL
name VARCHAR(50) NOT NULL
price_excluding_tax DECIMAL(4,2) NOT NULL
vat
PK id SMALLINT UNSIGNED NOT NULL
vat_100 DECIMAL(4,2) NOT NULL
ingredient
PK id BIGINT UNSIGNED NOT NULL
name VARCHAR(50) NOT NULL
recipe
PK, FK ingredient_id BIGINT UNSIGNED NOT NULL
PK, FK product_id BIGINT UNSIGNED NOT NULL
quantity DECIMAL(4,2) NOT NULL
category
PK id BIGINT UNSIGNED NOT NULL
name VARCHAR(50) NOT NULL

5. Composants du système

Diagramme de composants :


Suivi des restaurants
Suivi du restaurant
Suivi des commandes
<<<subsystem>>> Interface
Interface SGBDR
Interface client
Interface employé
Connexion client
<<<subsystem>>> Accès
Authentication et gestion des comptes
Gestion des commandes
<<<subsystem>>> Gestion des commandes
Catalogue produit
Constituer une commande
Commande
Passer une commande
Panier
Payer une commande
Paiement
Soumettre une transaction
Administration des commandes
Connexion employé
Actualisation des stocks
<<<subsystem>>> Gestion du restaurant
Stock d'ingrédient
Mise à jour du stock
Recette
Tableau de bord
Suivi de l'activité du restaurant
Suivi de l'activité des restaurants
<<<subsystem>>> Pilotages des restaurants
Tableau de bord
Transaction carte bancaire

Accès : l'utilisateur qu'il soit client ou employé accède à l'application par ce composant. La connexion sera elle différenciée selon le type d'utilisateur.

Interface : l'employé accède à l'interface qui lui est dédiée tout comme le client.

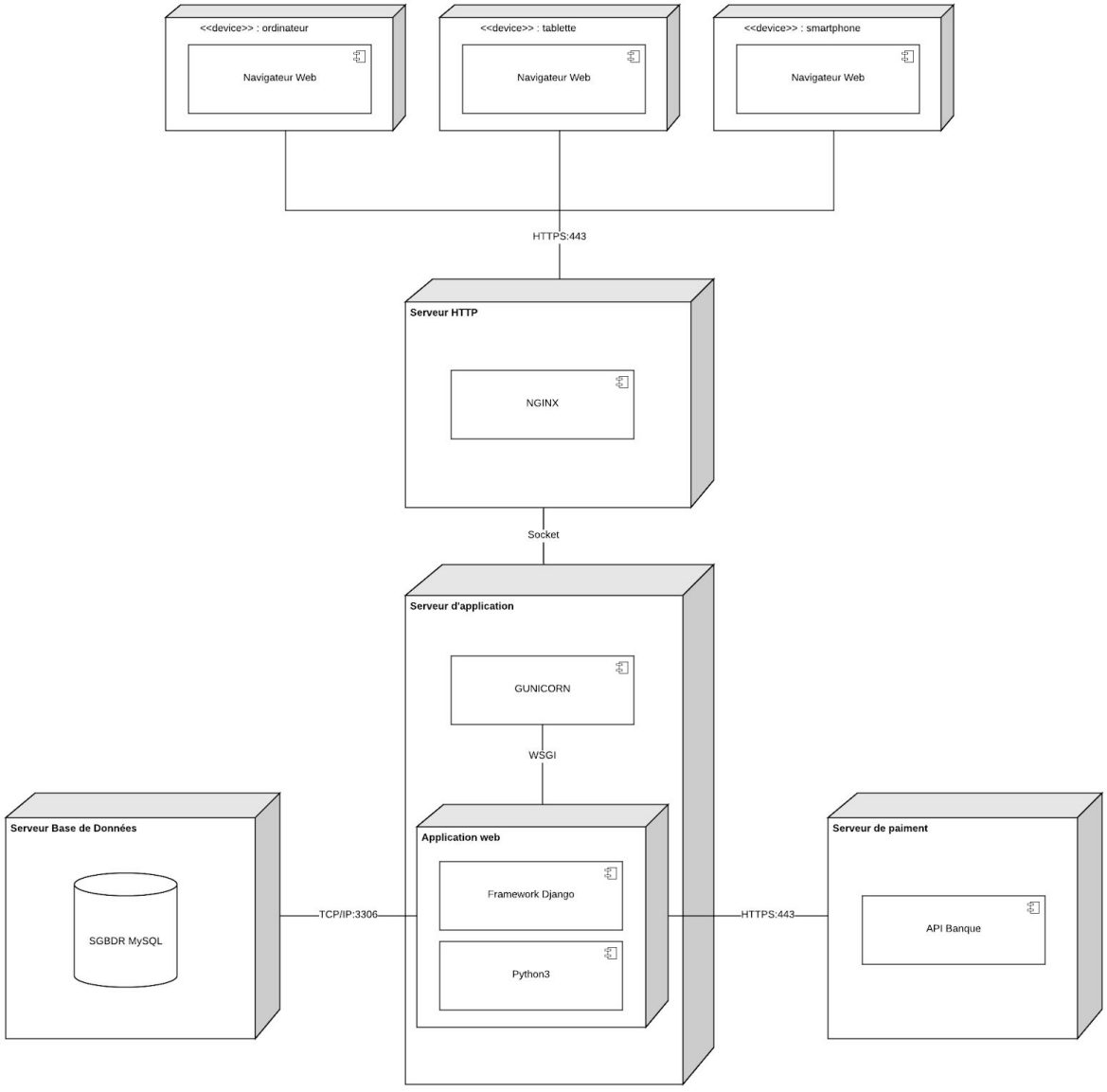
Gestion des commandes : l'employé à la possibilité d'accéder à l'administration des commandes (consultation et mise à jour du statut) quant au client il peut gérer ses propres commandes (constituer son panier, le valider, revenir dessus si la commande n'est pas partie en préparation, payer, s'informer du suivi de celle-ci).

Gestion du restaurant : Suivant les interactions avec le système de la part du client ou de l'employé, le suivi du restaurant, de son activité où de ses stocks est ici effectuée.

Pilotages des restaurants : ce dernier composants permet de doter les fondateurs d'une vision générale de l'activité de son groupe.

6. Déploiement des composants

Diagramme de déploiement :



La solution est déployée sur un serveur physique.

- Les utilisateurs interagissent avec un serveur HTTP utilisant Nginx.
- L'application sera déployée sur un serveur utilisant Gunicorn. Cette application sera réalisée à l'aide du framework Django.
- La base de données relationnelles sera un SGBDR MySQL qui communiquera uniquement avec le serveur d'application.
- Concernant les transactions bancaires, le serveur d'application communiquera directement avec l'API de la banque du client.