

CROS ARTHUR // Développement **JAVA**
P5 – OC Pizza – Spécifications Techniques

Table des matières

I. **CONTEXTE / Page 3**

DESCRIPTIF DU PROJET | Page 3

ACTEURS | Page 3

FONCTIONNALITÉS | Page 4

II. **DOMAINE FONCTIONNEL / Page 5**

DIAGRAMME DE COMPOSANTS | Page 5

DIAGRAMME DE DÉPLOIEMENT | Page 7

III. **MPD ET CLASSES / Page 8**

MODÈLE PHYSIQUE DE DONNÉES | Page 8

DIAGRAMME DE CLASSES | Page 12

I. CONTEXTE

DESRIPTIF DU PROJET //////////////////////////////////////

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 les 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 :
 - o Passer leurs commandes, en plus de la prise de commande par téléphone ou sur place ;
 - o Payer en ligne leur commande s'ils le souhaitent – sinon, ils paieront directement à la livraison ;
 - o 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 de notifier aux clients de l'état de leur commande.

ACTEURS //////////////////////////////////////

Les acteurs ont été définis précédemment dans le « [Projet 4 – Analysez les besoins de votre client pour son groupe de pizzerias](#) », en voici le rappel :

- Le visiteur, une personne n'ayant aucun compte de créé visitant simplement le site du restaurant ;
- Le client, une personne ayant créé un compte sur le site du restaurant et pouvant donc passer une commande ;
- Le gérant, pouvant accéder à toute fonctionnalité du système, principalement la création de comptes employés, la consultation du chiffre d'affaire, et l'attribution de livraisons ;
- Le livreur, pouvant se connecter au système pour consulter les livraisons lui étant affectées ;
- La caisse, personne en charge de l'encaissement des commandes, de remise des commandes sur place, et de l'enregistrement des commandes par téléphone ;
- Le préparateur, en charge de la préparation des commandes, pouvant accéder à un récapitulatif des différents produits proposés par le restaurant et pouvant mettre à jour le stock en temps réel ;
- L'approvisionnement, en charge de la maintenance du stock ;
- La banque, gestionnaire de l'encaissement par CB des commandes.

FONCTIONNALITÉS //////////////////////////////////////

De même, les fonctionnalités ont été préétablies dans le projet précédent :

- La consultation du menu, disponible pour le visiteur et le client. Cette fonction permet d'afficher la liste des produits disponibles ;
- La création d'un compte client, disponible au visiteur, lui permettant de pouvoir passer une commande sur le site ;
- La modification d'un compte client, disponible au client. Cette fonction permet au client de modifier ses informations de compte (adresse, mail, mot de passe, numéro de téléphone) ;
- La connexion à un compte, disponible à l'intégralité des acteurs sauf le visiteur, permettant à chaque acteur, employé ou client, de se connecter au système ;
- La création d'un compte employé, réservée au gérant, lui permettant de créer un compte pour chacun des employés ;
- La prise de commande par téléphone, possible pour la caisse, afin de pouvoir enregistrer les commandes en direct pour les clients ne passant pas par le site pour enregistrer leur commande ;
- La prise de commande via le site, proposée au client, afin de lui permettre de passer une commande directement depuis l'interface web ;
- L'affichage de l'historique des commandes, pensé pour le client, lui permettant de consulter ses commandes passées ;
- L'affichage d'un aide-mémoire pour le préparateur, lui affichant les étapes de confection d'un produit commandé ;
- L'affectation de commande à un livreur, réservée au gérant pour attribuer une commande préparée à un livreur ;
- La consultation et la modification des stocks, disponibles pour le gérant, le préparateur, ainsi que l'approvisionnement ;
- L'encaissement d'une commande, accessible à la caisse et au livreur ;
- La consultation du chiffre d'affaire, réservée au gérant.

II. DOMAINE FONCTIONNEL

DIAGRAMME DE COMPOSANTS

La décomposition du projet en composants est essentielle pour simplifier et clarifier les « modules » du programme à créer pour réaliser les tâches correspondantes aux fonctionnalités précédemment établies.

Les composants identifiés sont les suivants (voir fichier UML_Components.png pour le diagramme complet) :

- Le Frontend, correspondant à l'interface utilisée par le client, le visiteur, ou bien l'employé, sur son poste personnel ;
- La Database (Base de données), centralisant l'ensemble des informations et données stockées nécessaires à la gestion de l'infrastructure (Factures, Commandes, Clients, Employés, Stocks, etc...) ;
- Le backend, représentant le gros du projet. Son rôle est de traiter l'information provenant de la Database, et de la restituer sous une forme utilisable par le Frontend (l'utilisateur final). Il est lui-même organisé en plusieurs composants :
 - o Un composant Customers (Clients), gérant les comptes créés par les clients sur le site ;
 - o Un composant Orders (Commandes), gérant les commandes passées auprès du restaurant ;
 - o Un composant Management (Gestion), gérant les employés du restaurant ainsi que l'attribution des livraisons si nécessaire ;
 - o Un composant Products (Produits), gérant les stocks ainsi que les divers produits proposés par le restaurant ;
 - o Un composant Payment (Paiement), gérant la partie financière du restaurant, l'encaissement, et la facturation.
- Un dernier composant externe doit être cité, il s'agit du composant Bank Payment API, qui sert d'interface entre le système de paiement du restaurant et la banque elle-même, gérant l'autorisation ou non des paiements des commandes.

DIAGRAMME DE COMPOSANTS

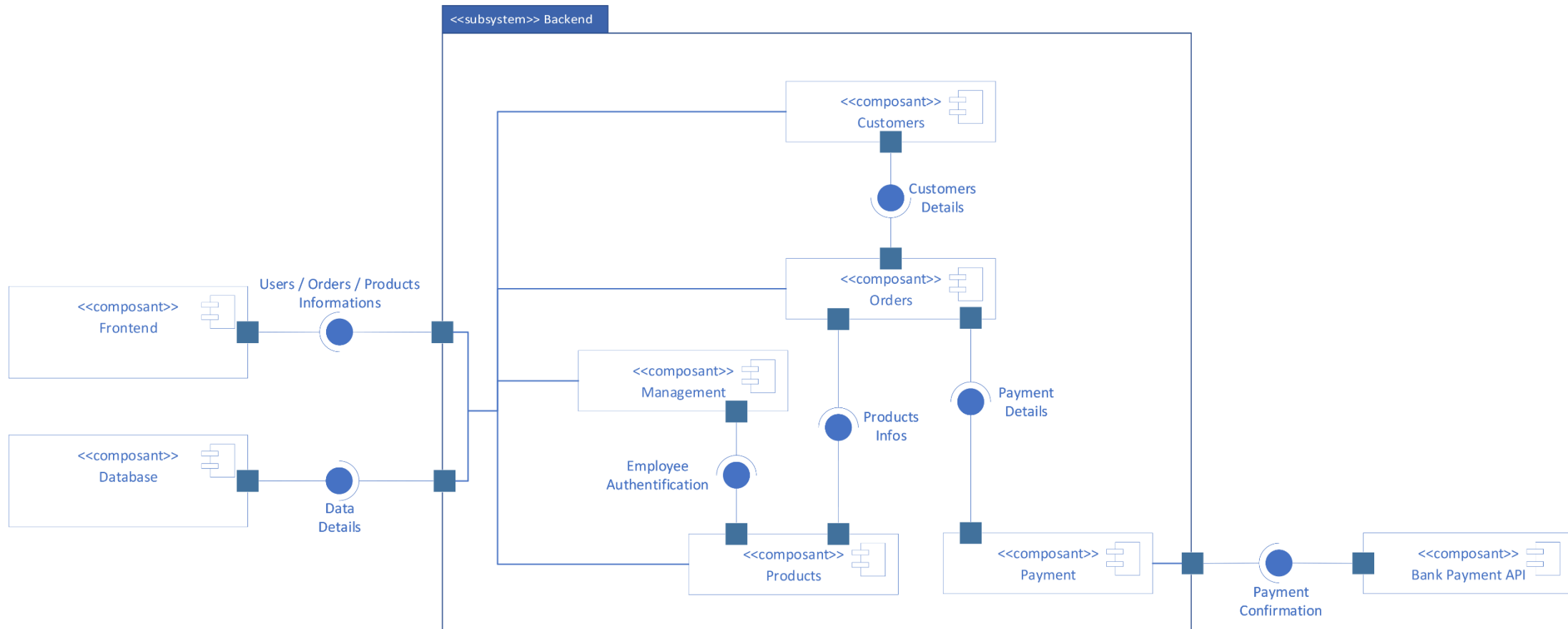


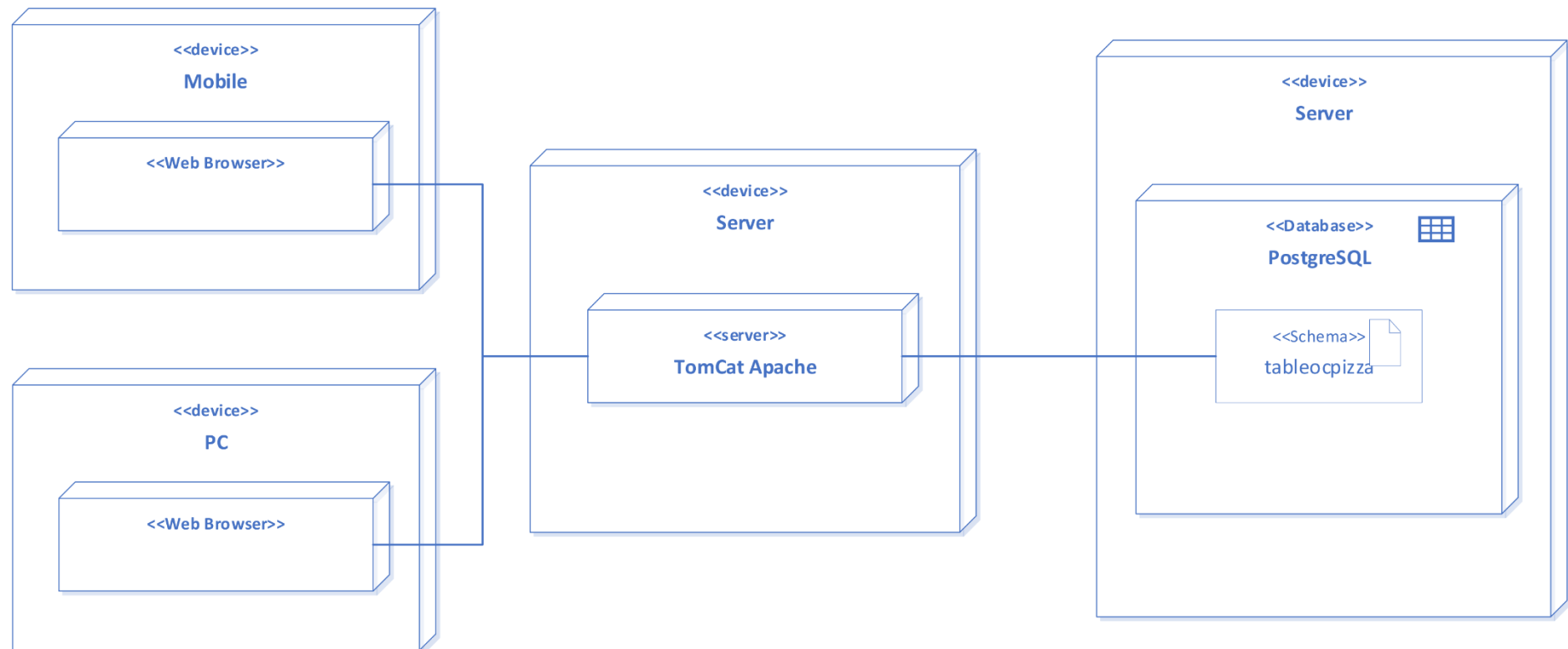
DIAGRAMME DE DÉPLOIEMENT //////////////////////////////////////

Une fois les composants déterminés, il est important de prévoir les machines physiques sur lesquelles ces composants seront déployés.

Le client ainsi que les employés accéderont à l'application via une interface web, accessible depuis n'importe quel navigateur répondant aux standards W3C (Edge, FireFox, IE, Chrome, Safari, Opera, etc...), installé sur un environnement MacOS, Windows, ou encore Linux.

Le Backend sera hébergé sur un serveur Apache Tomcat dédié utilisera la technologie J2EE accompagnée du framework Spring MVC pour faciliter son intégration.

La Database sera quant à elle gérée sous PostgreSQL.



III. MPD ET CLASSES

MODÈLE PHYSIQUE DE DONNÉES //////////////////////////////////////

L'opération suivante est d'établir la table de la base de données qui sera le support de notre logiciel. Plusieurs éléments ainsi que leurs attributs ont été établis comme nécessaires. Ces éléments sont les suivants, regroupés par les composants du Backend desquels ils dépendent :

- **MANAGEMENT**

- JobType – Liste des différents postes occupés par les employés
 - jobId – ID du poste de l'employé
 - jobName – Intitulé du poste de l'employé
- Employee – Liste des utilisateurs employés
 - employeId – ID de l'employé
 - restaurantId – ID du restaurant auquel l'employé est affecté
 - jobId – ID du poste de l'employé
 - title – Titre de l'employé
 - firstName – Prénom de l'employé
 - lastName – Nom de l'employé
 - email – Adresse mail de l'employé
- Restaurant – Liste des restaurants du groupe
 - restaurantId – ID du restaurant
 - zipCode – Code postal du restaurant
 - city – Ville du restaurant
 - street – Adresse du restaurant

- **CUSTOMERS**

- Customer – Liste des utilisateurs clients
 - customerId – ID du client
 - addressId – ID de l'adresse du client
 - title – Titre du client
 - firstName – Prénom du client
 - lastName – Nom du client
 - telNum – Numéro de téléphone du client
 - email – Adresse mail du client
- Address – Liste des adresses des clients
 - addressId – ID de l'adresse du client
 - zipCode – Code postal de l'adresse du client
 - city – Ville de l'adresse du client
 - street – Adresse du client
 - moreInfo – Information supplémentaire importante pour une livraison

- **ORDERS**

- Orders – Liste des commandes
 - orderId – ID de la commande
 - restaurantId – ID du restaurant gestionnaire de la commande
 - orderNumber – Numéro de la commande
 - orderDate – Date de la commande
- Bill – Liste des factures
 - billId – ID de la facture
 - payTypeId – ID du moyen de paiement de la facture
 - orderId – ID de la commande rattachée à la facture
 - billNumber – Numéro de la facture
 - totalAmount – Montant TTC de la facture
 - vatAmount – Montant TVA de la facture
 - billStatus – Statut de la facture (Payée ou non)

- **PAYMENT**

- PaymentType – Liste des différents moyens de paiement
 - payTypeId – ID du moyen de paiement
 - payTypeName – Intitulé du moyen de paiement

- **PRODUCTS**

- Product – Liste des différents produits proposés par les restaurants
 - productId – ID du produit
 - restaurantId – ID du restaurant proposant le produit
 - productName – Nom du produit
 - productPrice – Prix TTC du produit
- Ingredient – Liste des ingrédients utilisés par le restaurant
 - ingredientId – ID de l'ingrédient
 - supplyId – ID du stock de l'ingrédient
 - ingredientName – Nom de l'ingrédient
 - spoilDate – Date de péremption de l'ingrédient
 - measureUnit – Unité de mesure de l'ingrédient
- Supply – Liste des stocks des ingrédients
 - supplyId – ID du stock
 - restaurantId – ID du restaurant propriétaire du stock
 - quantity – Quantité de produit en stock
 - resupplyDate – Date de stock de l'ingrédient

MODÈLE PHYSIQUE DE DONNÉES (MPD)

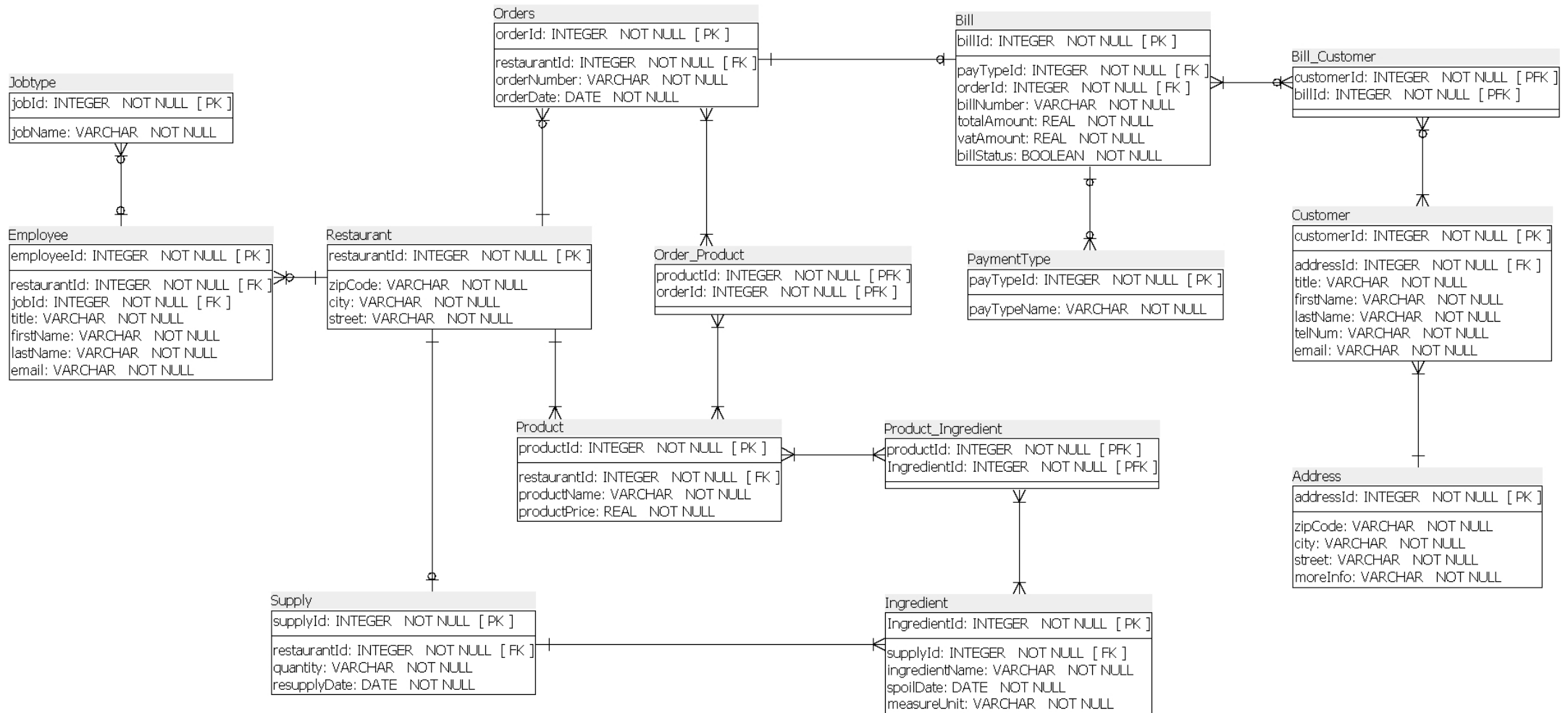


DIAGRAMME DE CLASSES //////////////////////////////////////

Pour terminer, il est nécessaire d'établir les classes du projet JAVA lui-même à partir du MPD. La construction du diagramme de classes reprend l'architecture du MPD, sans les éléments relatifs aux relations de tables dans la base de données.

