

---

# PIZZADELIVERY

---

## Projet : « Gestion d'une entreprise de pizzas à domicile »

**Pierre Lefebvre**

---

PizzaDelivery est un gestionnaire de commande pour des livraisons de pizzas à domicile. Pour la réalisation de ce projet nous avons choisi l'architecture trois tier avec une base de données (MySQL), un contrôleur en API REST (Java - SpringBoot) ainsi qu'une vue web (HTML/CSS - JavaScript). Cette architecture répond à des besoins modernes, puisqu'elle ne nécessite pas d'installation de logiciel sur son poste fixe et peut ainsi, grâce à un hébergement serveur, être disponible sur tous les terminaux de la pizzeria. De plus la technologie API REST associée à son contrôleur communiquant par formulaire web JSON le rend universel et facilement adaptable et utilisable par n'importe quel logiciel ou plug-in.

Tout a été entièrement codé par nos soins et je vous invite fortement à jeter un œil du côté du code. Le code lui-même a été organisé selon les couches 'traditionnelles' d'une application MVC trois tier.

La liste des routes de l'API REST est disponible en lançant l'application et en se rendant sur <http://localhost:8080/swagger-ui.html>

Le code est disponible sur github : <https://github.com/veikoon/PizzaDelivery>

**Merci de Lire attentivement le README**

Il se décompose en trois parties :

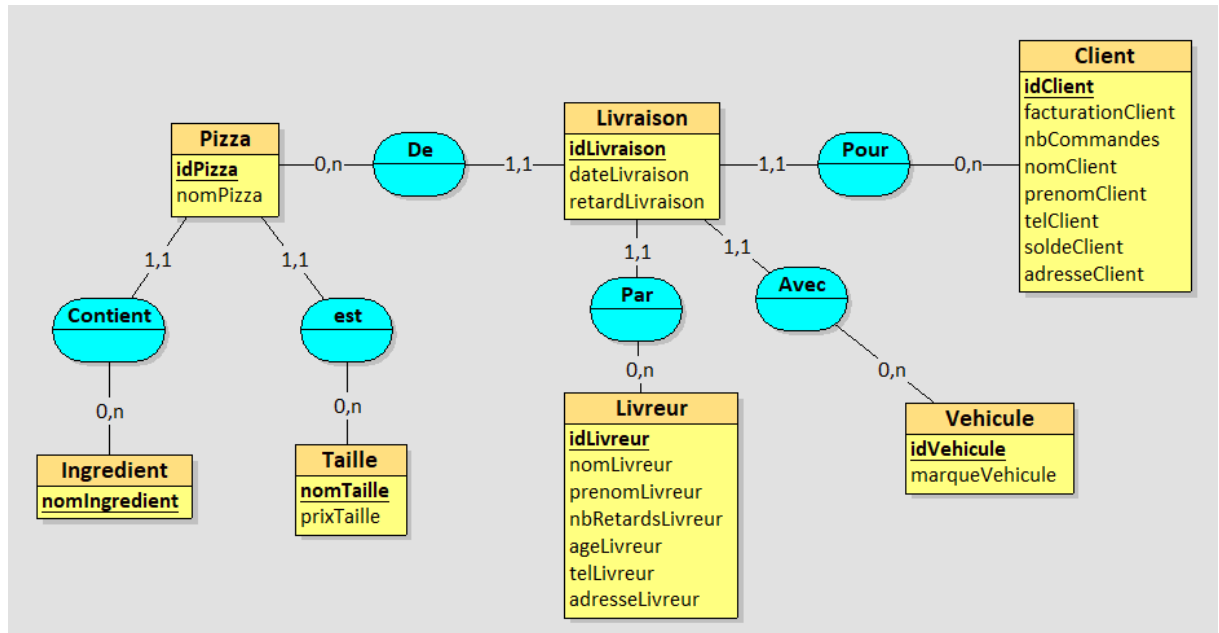
- La partie assets qui contient les scripts sql

- La partie pizzawebsite qui contient l'interface web

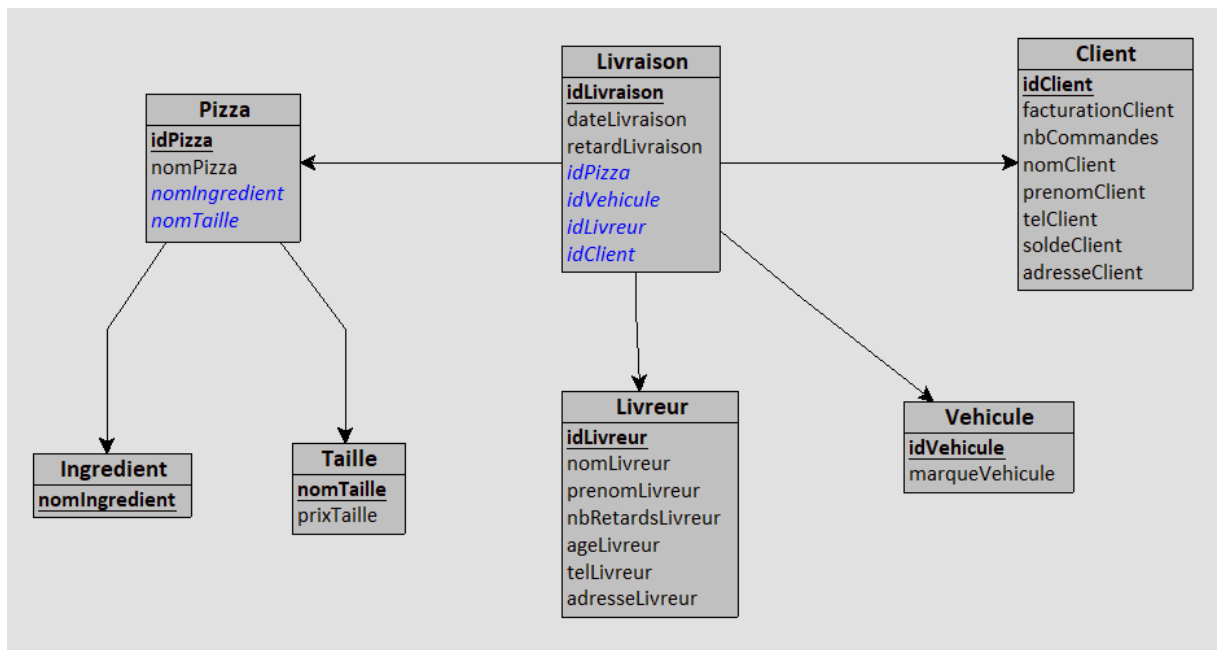
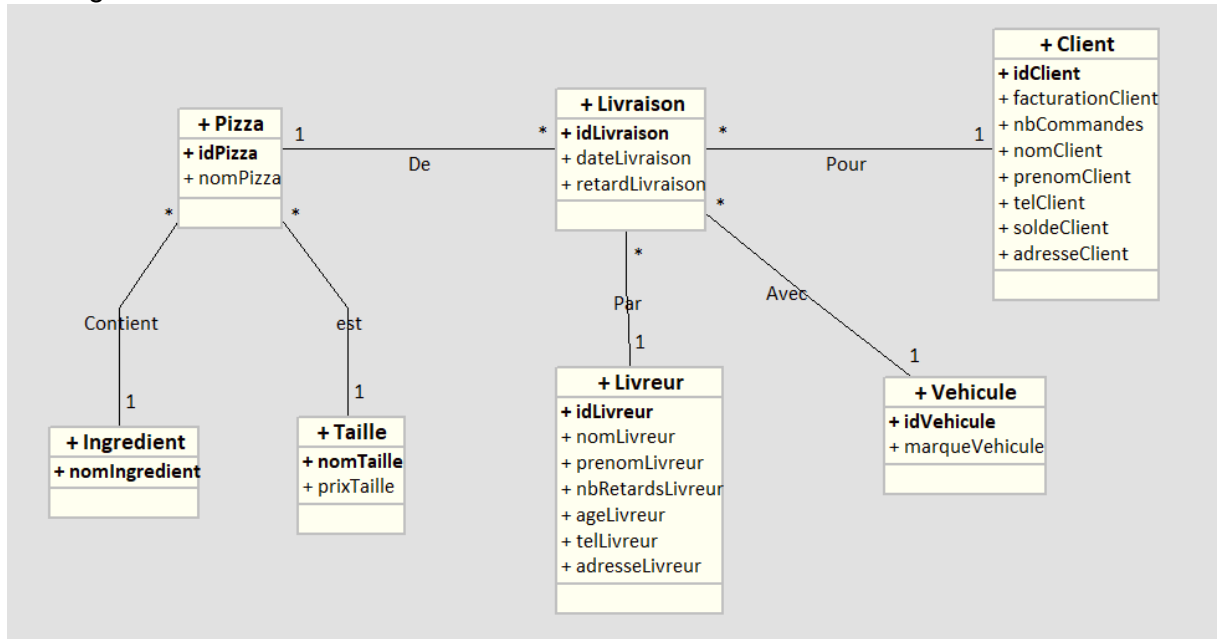
- Et enfin à la racine la partie contrôleur java qui est une API REST

## Conception de la base de données

### 1) Elaboration du modèle entité-association



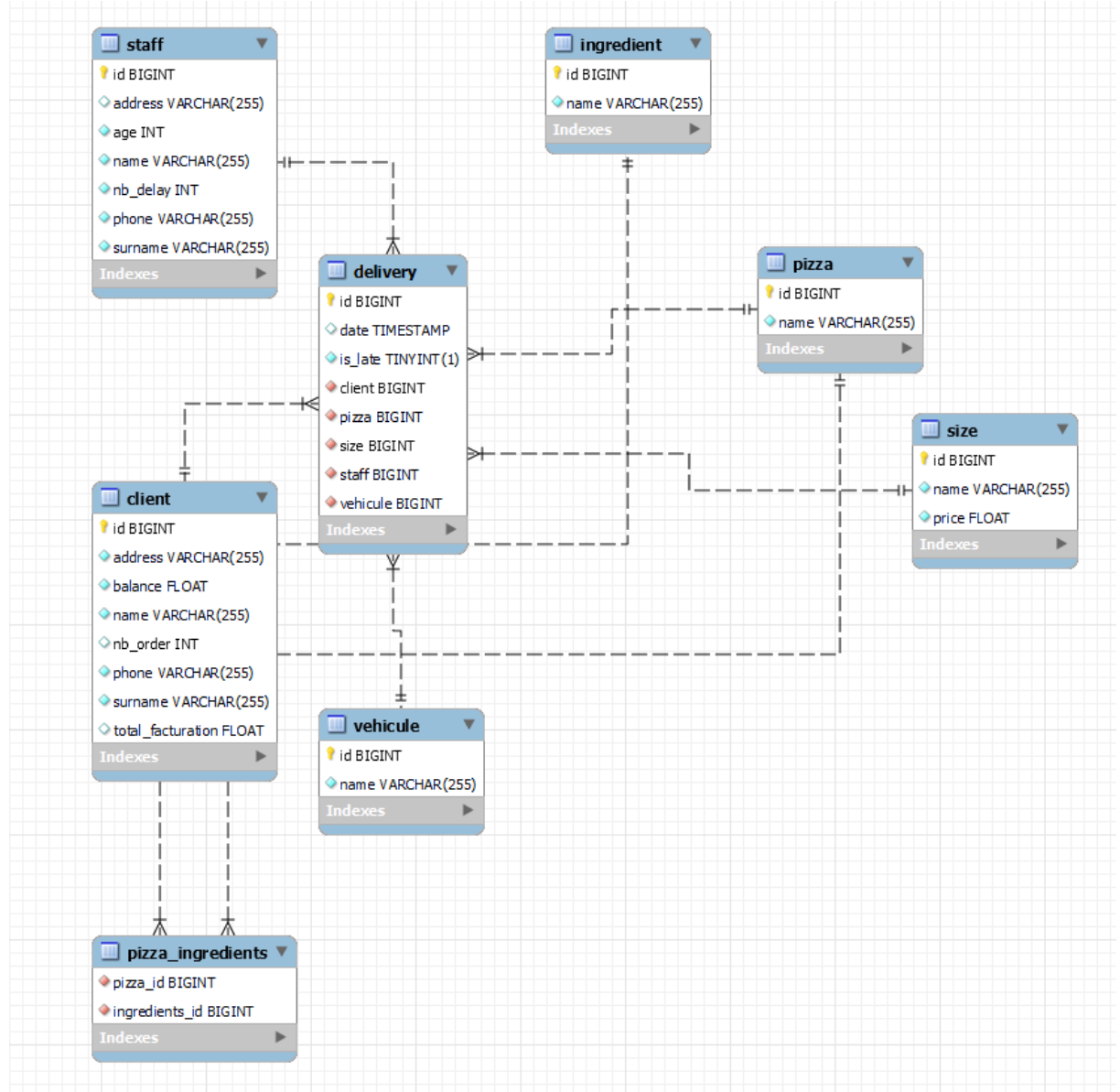
## 2) Passage au modèle relationnel



```

Taille = (nomTaille VARCHAR(50), prixTaille INT);
Ingredient = (nomIngredient VARCHAR(50));
Livreur = (idLivreur INT, nomLivreur VARCHAR(50), prenomLivreur VARCHAR(50), nbRetardsLivreur INT, ageLivreur INT, telLivreur VARCHAR(50), adresseLivreur VARCHAR(50));
Client = (idClient INT, facturationClient DECIMAL(15,2), nbCommandes INT, nomClient VARCHAR(50), prenomClient VARCHAR(50), telClient VARCHAR(50), soldeClient DECIMAL(15,2), adresseClient VARCHAR(50));
Vehicule = (idVehicule INT, marqueVehicule VARCHAR(50));
Pizza = (idPizza INT, nomPizza VARCHAR(50), #nomIngredient, #nomTaille);
Livraison = (idLivraison INT, dateLivraison DATETIME, retardLivraison LOGICAL, #idPizza, #idVehicule, #idLivreur, #idClient);
  
```

Schéma final de la BDD :



### 3) Script de la création des tables

<https://github.com/veikoon/PizzaDelivery/blob/master/assets/createDatabase.sql>

### 4) Script d'insertion des données dans la base de données

<https://github.com/veikoon/PizzaDelivery/blob/master/assets/insertDatabase.sql>

### **Interrogation de la base de données**

Le menu et les fiches de livraisons sont disponible sur le portail web.

#### 3) Questions diverses

- Quels sont les véhicules n'ayant jamais servi ?

```
select * from vehicule where id not in (select vehicule from delivery);
```

- Calcul du nombre de commandes par client ?

```
select name, count(*) as totalCommandes from client inner join delivery on client.id = delivery.client  
group by name;
```

- Calcul de la moyenne des commandes ?

```
select avg(price) from size inner join delivery on size.id = delivery.size;
```

- Extraction des clients ayant commandé plus que la moyenne ?

```
select * from client where nb_order > (select avg(nb_order) from client);
```

Toutes ces informations sont récupérables directement via l'interface web sans soucis d'utilisation de query sql, elles sont mises ici à simple titre indicative.