



# Dossier de conception

Vroum-Vroum

***Élèves :***

Johan GUILLEN  
Loïc MAYRAN  
Maxence OGIER  
Pauline BOUYSSOU

***Enseignants :***

Richard BONNAMY

23 janvier 2025



# Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Objet du document . . . . .	2
<b>2</b>	<b>Architecture logicielle</b>	<b>3</b>
2.1	Produits et versions . . . . .	3
<b>3</b>	<b>Focus technique</b>	<b>4</b>
3.1	Diagramme de classes métier (ou MCD) . . . . .	4
3.2	Diagramme entités relations (ou MPD) . . . . .	5
3.3	Diagramme en couche . . . . .	5
3.4	Règles de développement coté back . . . . .	6
<b>4</b>	<b>Tests et intégration</b>	<b>6</b>
4.1	Tests unitaires . . . . .	6
4.2	Tests fonctionnels . . . . .	6
4.3	Indicateurs de qualité de code . . . . .	6



# 1 Introduction

## 1.1 Objet du document

Ce document a pour objectif de présenter l'essentiel des questions techniques liées à la mise en place de l'application Vroum-Vroum. Ce document présente :

- Le diagramme de classe
- Le modèle physique de données
- le diagramme de couche

## 2 Architecture logicielle

### 2.1 Produits et versions

#### 2.1.1 Langages, frameworks et librairies spécifiques

Nom	Version
Langage Java	21
Spring Boot	3
Hibernate	7.2
Angular	18

TABLE 1 – tableau des langages, frameworks et librairies spécifiques

#### 2.1.2 serveur de base de données

Nom	Version
MySQL	8

TABLE 2 – tableau des serveurs et BDD utilisé

### 3 Focus technique

#### 3.1 Diagramme de classes métier (ou MCD)

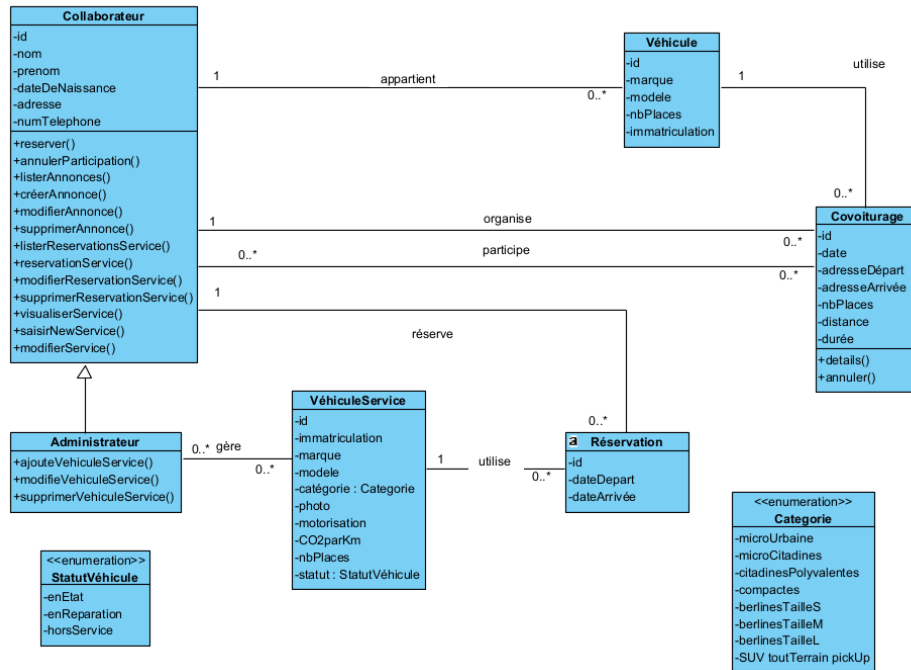


FIGURE 1 – Diagramme de cas d'utilisation

### 3.2 Diagramme entités relations (ou MPD)

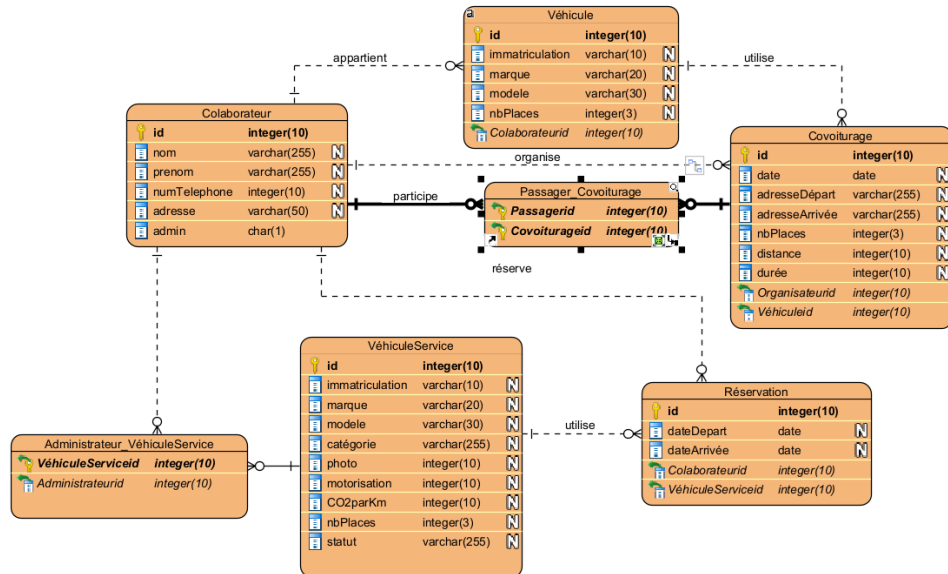


FIGURE 2 – Diagramme de cas d'utilisation

### 3.3 Diagramme en couche

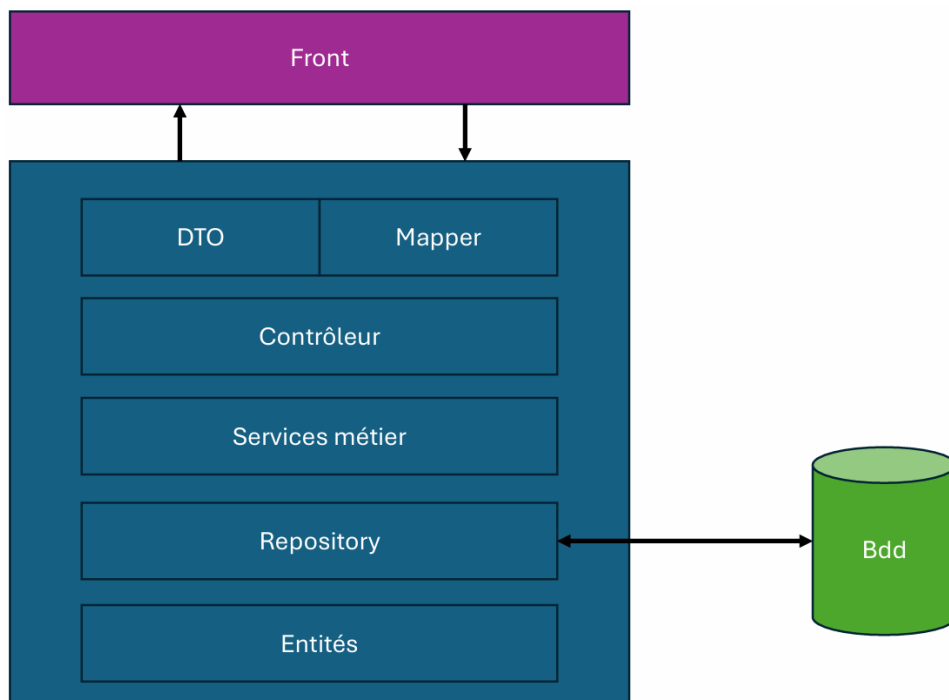


FIGURE 3 – Diagramme de cas d'utilisation



### 3.4 Règles de développement coté back

Règles de développement :

- 100 % de la Javadoc doit être renseignée
- Les règles de nommage respectent les conventions de nommage en usage en 2025.

Découpage en couches :

- Couche contrôleur (qui contient les routes d'accès au back)
- Couche de services qui va réaliser les contrôles métier.
- Couche repository (spring data)
- Couche DTO : le contrôleur renvoie vers le front des instances de DTO.
- Couche entités avec les annotations JPA.
- Classes utilitaires.

## 4 Tests et intégration

### 4.1 Tests unitaires

### 4.2 Tests fonctionnels

### 4.3 Indicateurs de qualité de code