

# CAHIER DES CHARGES TECHNIQUES

## SOMMAIRE

### [1. Contexte du projet](#)

#### [1.1. Présentation du projet](#)

#### [1.2. Date de rendu du projet](#)

### [2. Besoins fonctionnels](#)

### [3. Ressources nécessaires à la réalisation du projet](#)

#### [3.1. Ressources matérielles](#)

#### [3.2. Ressources logicielles](#)

### [4. Gestion du projet](#)

### [5. Conception du projet](#)

#### [5.1. Le front-end](#)

##### [5.1.1. Wireframes](#)

##### [5.1.2. Maquettes](#)

##### [5.1.3. Arborescences](#)

#### [5.2. Le back-end](#)

##### [5.2.1. Diagramme de cas d'utilisation](#)

##### [5.2.2. Diagramme d'activités](#)

##### [5.2.3 Dictionnaire de données](#)

##### [5.2.4 Modèles Conceptuel de Données \(MCD\)](#)

##### [5.2.5. Modèle Logique de Données \(MLD\)](#)

##### [5.2.6. Modèle Physique de Données \(MPD\)](#)

### [6. Technologies utilisées](#)

#### [6.1. Langages de développement Web](#)

#### [6.2. Base de données](#)

### [7. Sécurité](#)

#### [7.1. Login et protection des pages administrateurs](#)

#### [7.2. Cryptage des mots de passe avec Bcrypt](#)

#### [7.3. Protection contre les attaques XSS \(Cross-Site Scripting\)](#)

#### [7.4. Protection contre les injections SQL](#)

# 1. Contexte du projet

## 1.1. Présentation du projet

Travail en classe et à la maison à partir d'un support étudiant reprenant un contexte professionnel.

Vous occupez actuellement le poste de concepteur et développeur au sein de la Direction des Systèmes d'Information de la préfecture de votre département. La responsable du service des cartes grises souhaite faire évoluer l'application métier utilisée par son service. Toutefois, aucun document de conception n'est disponible. Votre mission consiste donc à élaborer la documentation technique de l'application existante afin de faciliter la réflexion sur son évolution.

L'objectif est de concevoir et développer un logiciel de gestion des cartes grises (ajout, modifications et suppressions d'informations). Le logiciel est développé en Java.

## 1.2. Date de rendu du projet

Le projet doit être rendu au plus tard le 6 mars 2025.

# 2. Besoins fonctionnels

Les données seront stockées dans une base de données relationnelle pour faciliter la gestion et la mise à jour des informations.

# 3. Ressources nécessaires à la réalisation du projet

## 3.1. Ressources matérielles

Pc Portable ou fixe (unité centrale, souris, clavier). Ils doivent être reliés à internet.

## 3.2. Ressources logicielles

Environnement de développement : Visual code

Plateforme développement collaborative : GitHub

Serveur web Apache, Système de Base de Donnée Relationnelle MySQL (contenu dans MAMP)

Outil de Gestion de Projet : Trello

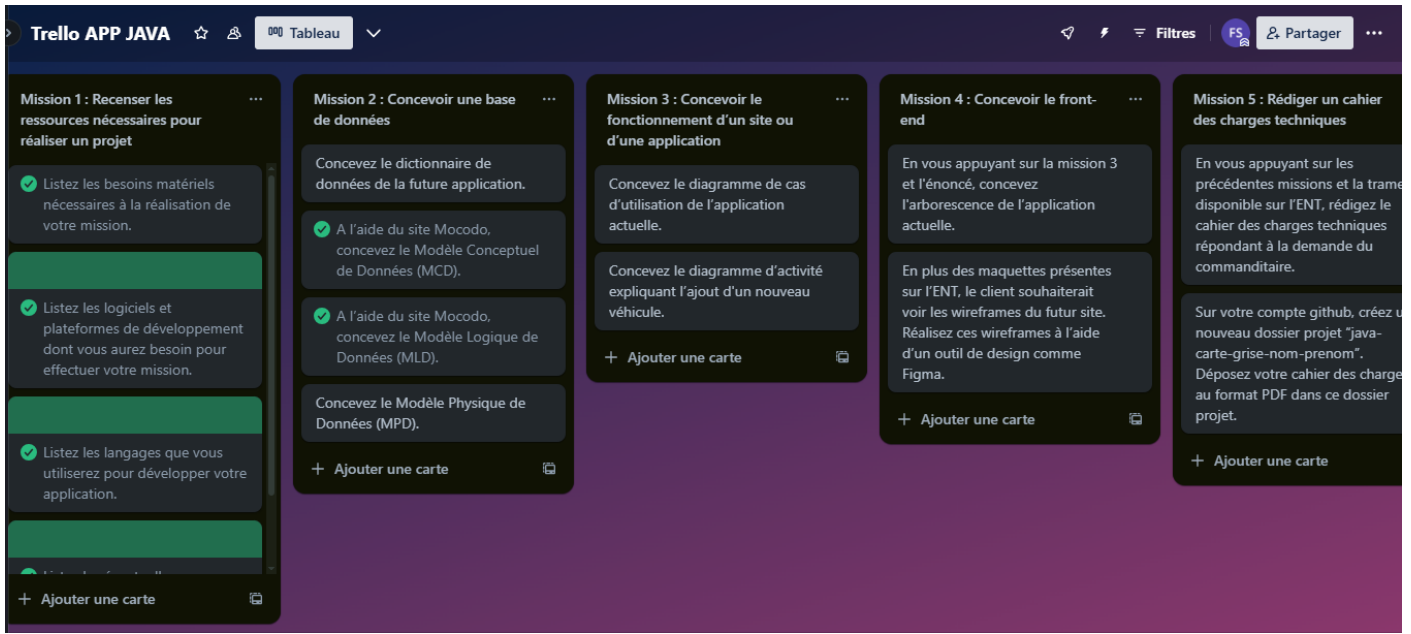
Conception UML Arborescence : Visual Paradigme

Maquette : Figma

Conception Base de données : MoCoDo online.

## 4. Gestion du projet

Pour réaliser le projet, nous utiliserons la méthode Agile Kanban. Nous utiliserons également l'outil de gestion de projet en ligne Trello.



Nous travaillons également sur GitHub, plateforme de développement collaboratif.

## 5. Conception du projet

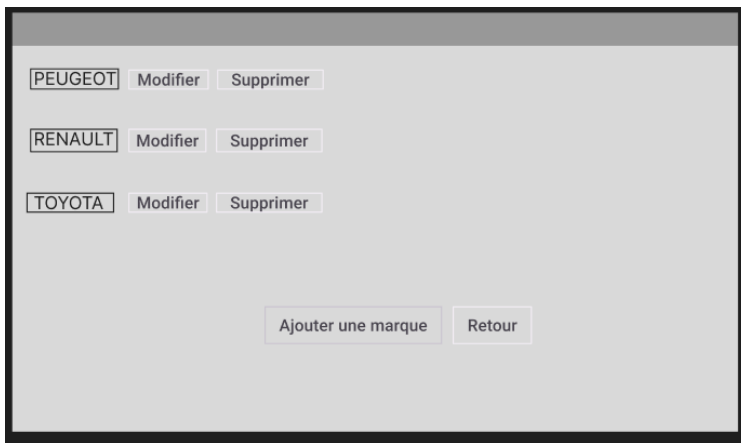
### 5.1. Le front-end

#### 5.1.1. Wireframes

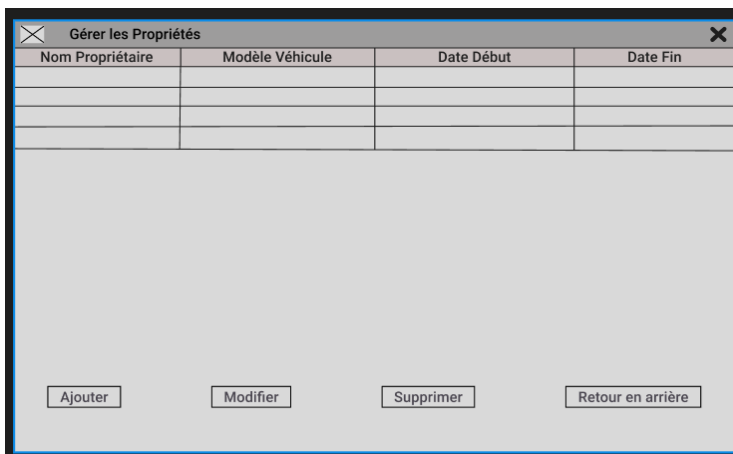
Wireframe page d'accueil



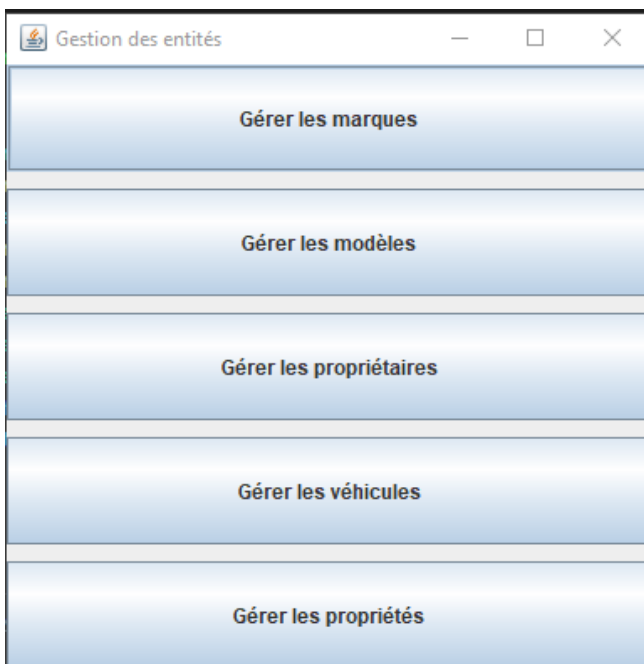
Wireframe liste des marques



Wireframe gérer les propriétés



## 5.1.2. Maquettes



Liste des Marques

Peugeot

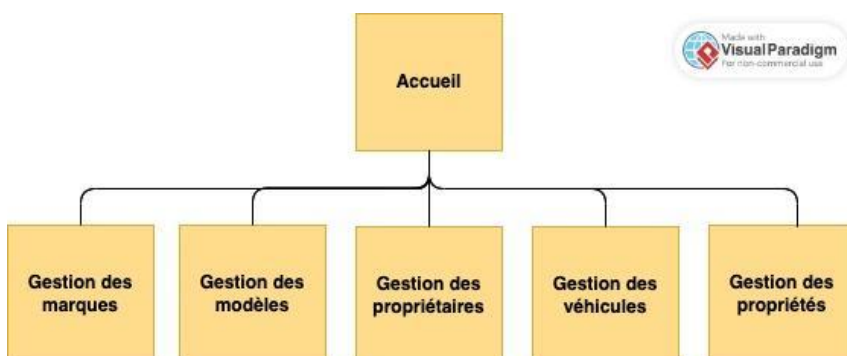
Renault

Toyota

Affichage des véhicules possédés

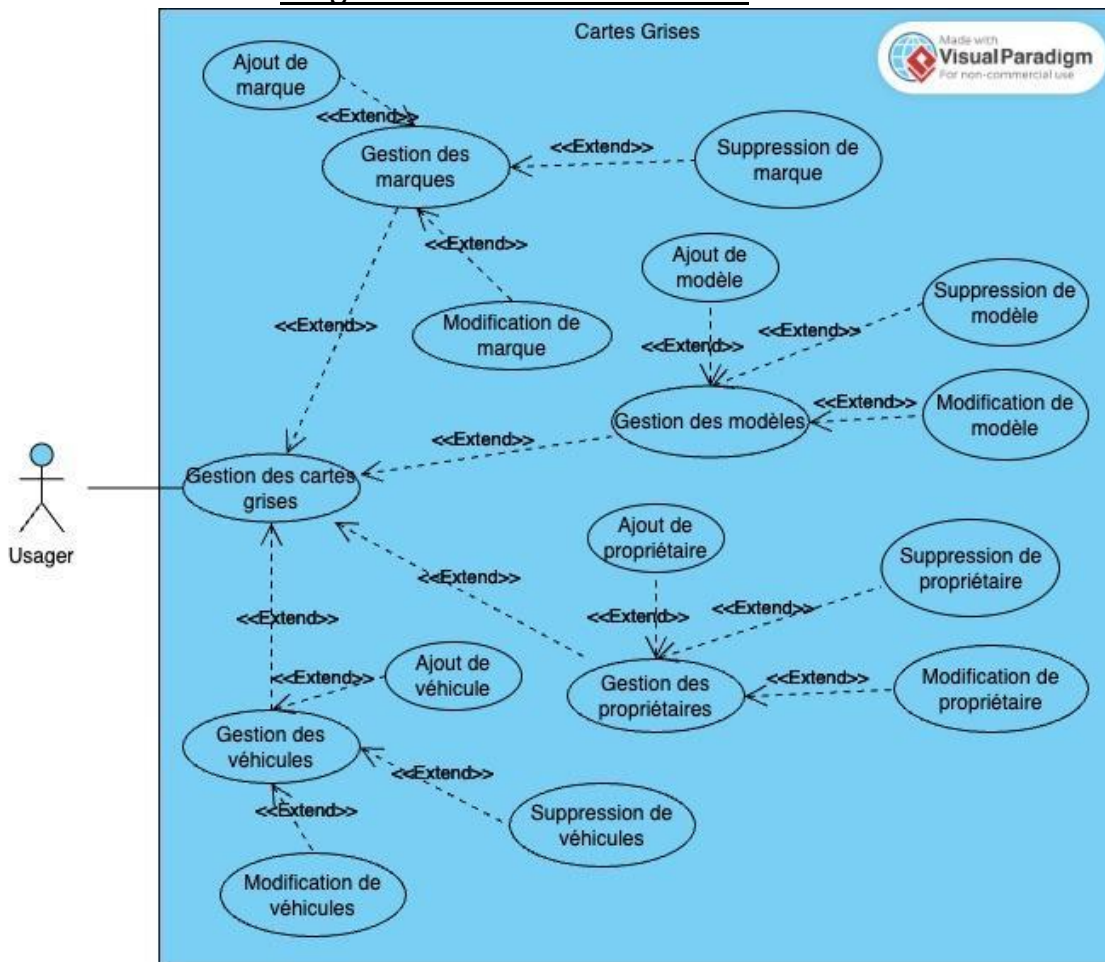
Nom Propriétaire	Modèle Véhicule	Date Début	Date Fin
Doe	208	2023-01-01	2023-12-31
Doe	Yaris	2024-06-01	N/A
Doe	Clio	2023-01-01	N/A

### 5.1.3. Arborescences



## 5.2. Le back-end

### 5.2.1. Diagramme de cas d'utilisation



### 5.2.2. Diagramme d'activités

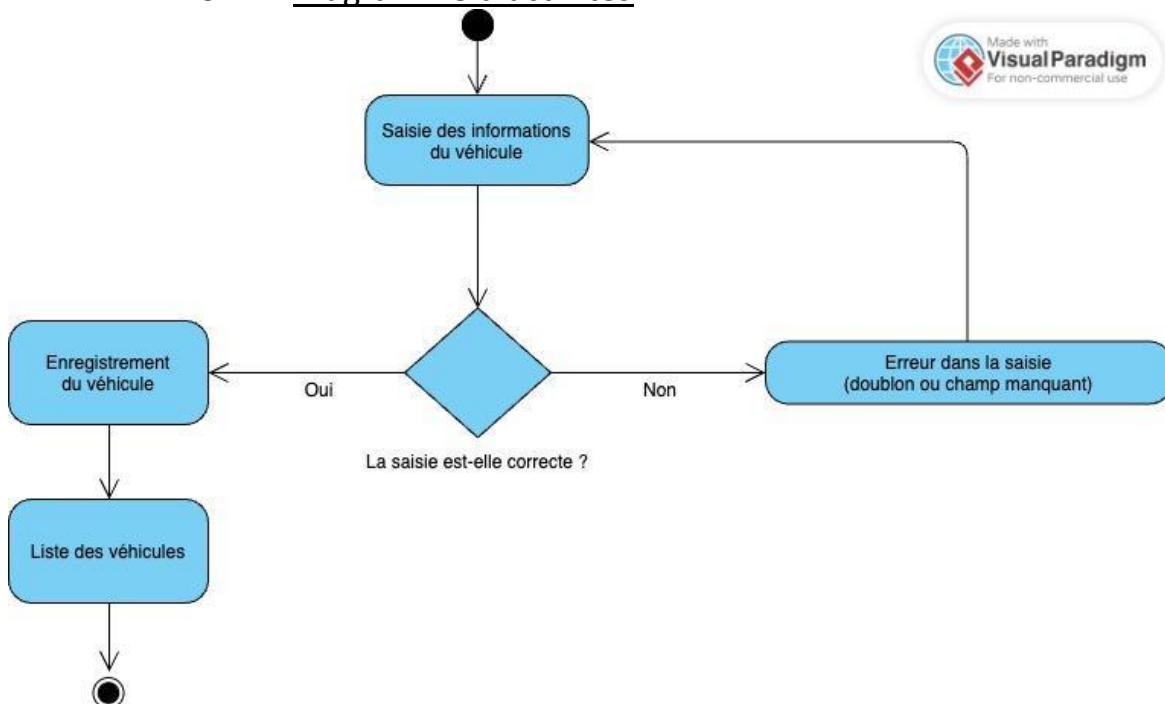
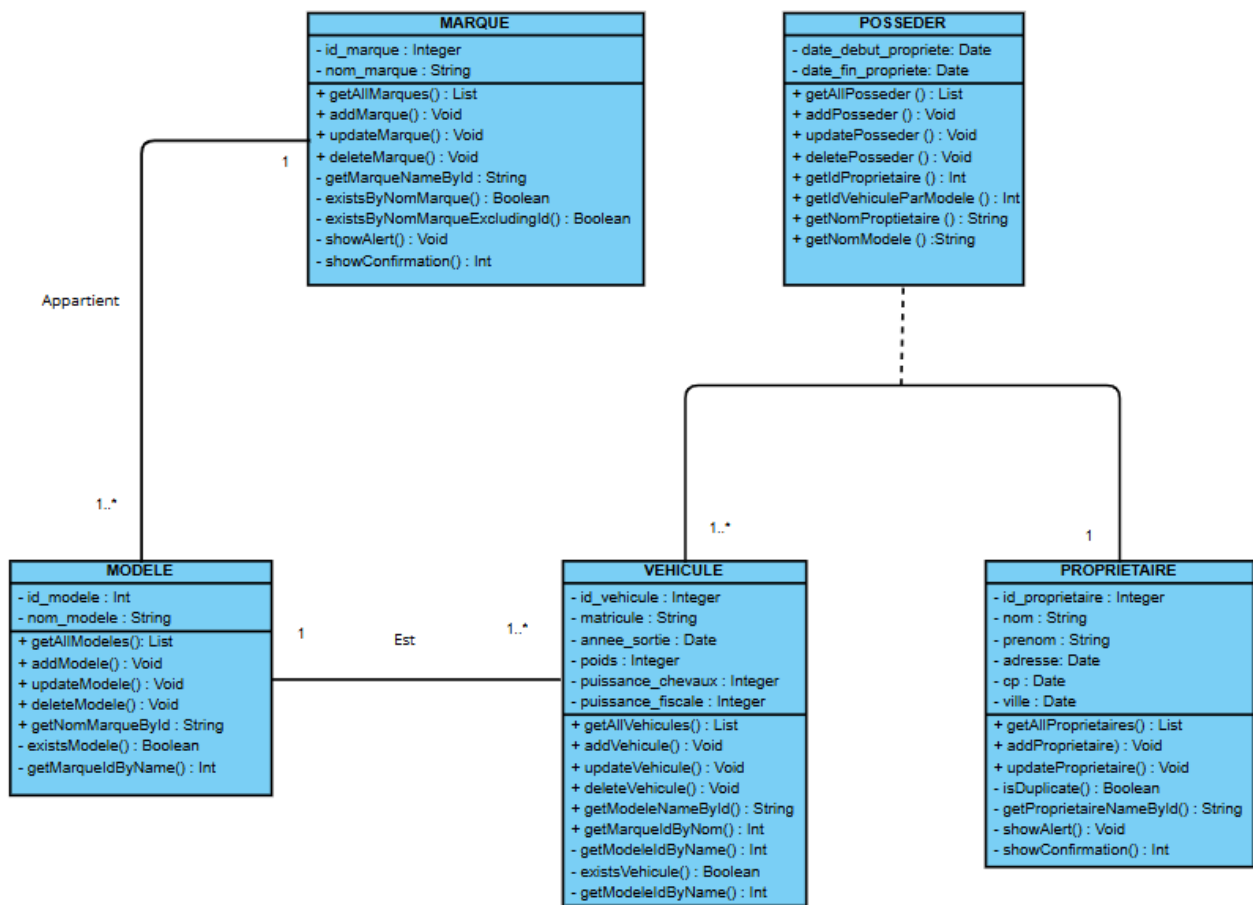


DIAGRAMME DE CLASSE

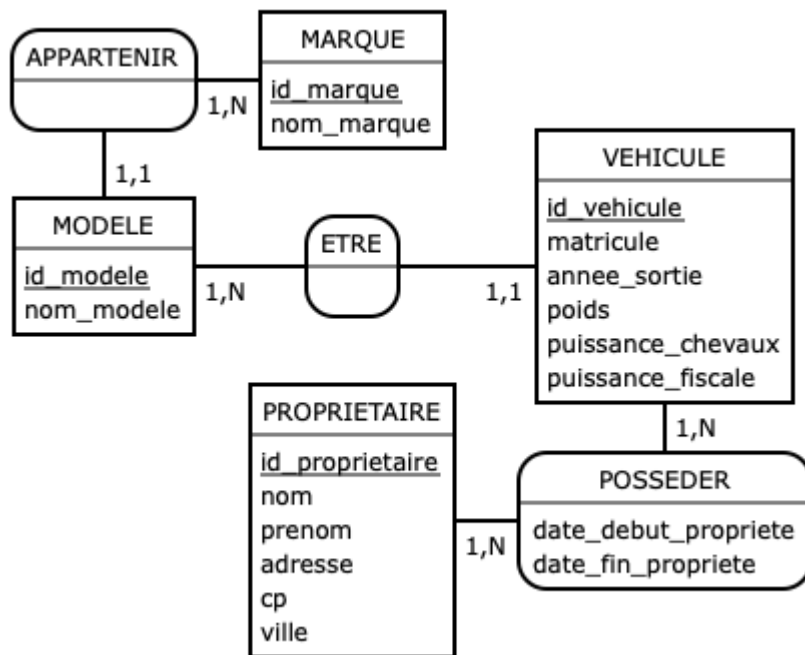


### 5.2.3. Dictionnaire de données

Nom de l'attribut	Type	Contraintes	Description
id_marque	INT	PRIMARY KEY, AUTO_INCREMENT	Identifiant unique de la marque
nom_marque	VARCHAR(100)	NOT NULL	Nom de la marque
id_modele	INT	PRIMARY KEY, AUTO_INCREMENT	Identifiant unique du modèle
nom_modele	VARCHAR(100)	NOT NULL	Nom du modèle
id_marque	INT	FOREIGN KEY → Marque(id_marque)	Référence à la marque
id_vehicule	INT	PRIMARY KEY, AUTO_INCREMENT	Identifiant unique du véhicule
matricule	VARCHAR(20)	NOT NULL, UNIQUE	Numéro d'immatriculation
annee_sortie	YEAR	NOT NULL	Année de mise en circulation
poids	DECIMAL(6,2)	NOT NULL	Poids du véhicule (en kg)
puissance_chevaux	INT	NOT NULL	Puissance en chevaux
puissance_fiscale	INT	NOT NULL	Puissance fiscale
id_modele	INT	FOREIGN KEY → Modèle(id_modele)	Référence au modèle du véhicule
id_proprietaire	INT	PRIMARY KEY, AUTO_INCREMENT	Identifiant unique du propriétaire
nom	VARCHAR(100)	NOT NULL	Nom du propriétaire
prenom	VARCHAR(100)	NOT NULL	Prénom du propriétaire
adresse	TEXT	NOT NULL	Adresse postale
cp	VARCHAR(10)	NOT NULL	Code postal
ville	VARCHAR(100)	NOT NULL	Ville
id_vehicule	INT	PRIMARY KEY (avec id_proprietaire), FOREIGN KEY → Véhicule(id_vehicule)	Référence au véhicule
id_proprietaire	INT	PRIMARY KEY (avec id_vehicule), FOREIGN KEY → Propriétaire(id_proprietaire)	Référence au propriétaire
date_debut_propriete	DATE	NOT NULL	Date de début de possession
date_fin_propriete	DATE	NULL (si toujours propriétaire)	Date de fin de possession



#### 5.2.4. Modèles Conceptuel de Données (MCD)



#### 5.2.5. Modèle Logique de Données (MLD)

**Marque** (id\_marque, nom\_marque)

Clé primaire : id\_marque

**Modele** (id\_modele, nom\_modele, id\_marque)

Clé primaire : id\_modele

Clé étrangère : id\_marque en reference à id\_marque de MARQUE

**Vehicule** (id\_vehicule, matricule, annee\_sortie, poids, puissance\_chevaux, puissance\_fiscale, id\_modele)

Clé primaire : id\_vehicule

Clé étrangère : id\_modele en reference à id\_modele de MODELE

**Proprietaire** (id\_proprietaire, nom, prenom, adresse, cp, ville)

Clé primaire : id\_proprietaire

**Posseder** (id\_vehicule, id\_proprietaire, date\_debut\_propriete, date\_fin\_propriete)

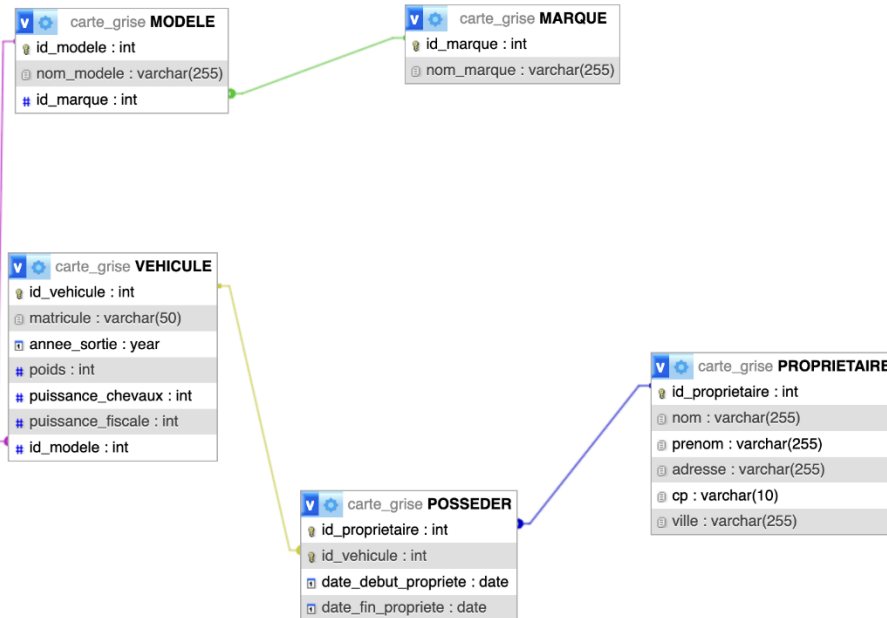
Clé primaire : id\_proprietaire en reference à id\_proprietaire de PROPRIETAIRE

id\_vehicule en reference à id\_vehicule de VEHICULE

Clé étrangère : id\_proprietaire en reference à id\_proprietaire de PROPRIETAIRE

id\_vehicule en reference à id\_vehicule de VEHICULE

#### 5.2.6. Modèle Physique de Données (MPD)



## 6. Technologies utilisées

### 6.1. Langages de développement Web

JAVA

## 6.2. Base de données

Serveur Web : Apache contenu dans MAMP

Système de base de données : SQL

Langage de base de données : SQL

## 7. Sécurité

### 7.1. Protections utilisées

- Utilisation des requêtes préparées (PreparedStatement) afin d'empêcher l'exécution de commandes SQL malveillantes.
- Vérification des doublons afin de garantir l'intégrité des données en évitant les doublons.

### 7.2. Protection contre les injections SQL

Une injection SQL, parfois abrégée en SQLi, est un type de vulnérabilité dans lequel un pirate utilise un morceau de code SQL (« Structured Query Language », langage de requête structuré) pour manipuler une base de données et accéder à des informations potentiellement importantes. C'est l'un des types d'attaques les plus répandus et menaçants, car il peut potentiellement être utilisé pour nuire à n'importe quelle application Web ou n'importe quel site Web qui utilise une base de données SQL (soit la plupart).

Pour se protéger des injections SQL, nous allons donc utiliser les requêtes préparées **PreparedStatement** , **try-catch**.