

Cahier des charges techniques



Projet Carte Grise

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. Modèles Conceptuel de Données \(MCD\)](#)
 - [5.2.4. Modèle Logique de Données \(MLD\)](#)
 - [5.2.5. 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

Vous occupez actuellement le poste de concepteur et développeur de sein de la Direction des systèmes d'information de la préfecture de votre département.

La responsable du service des cartes grises souhaiterait faire évoluer leur application métiers.

Cependant, aucun document de conception n'est disponible. Votre travail consiste donc à travailler sur l'élaboration de documents de conception de l'application actuelle en vue de faciliter la réflexion autour de son évolution.

1.2. Date de rendu du projet

Le projet doit être rendu au plus tard le 06/03/2025.

2. Besoins fonctionnels

Il s'agit d'une application de gestion de cartes grises qui permettra de gérer les informations relatives aux cartes grises des véhicules.

Les données seront stockées dans une base de données relationnelle, et pourront être directement mises à jour et gérées via l'application.

3. Ressources nécessaires à la réalisation du projet

3.1. Ressources matérielles

- Ordinateur portable ou fixe : clavier, souris, écran, unité centrale.
- Connexion Internet : par câble ou Wi-Fi.

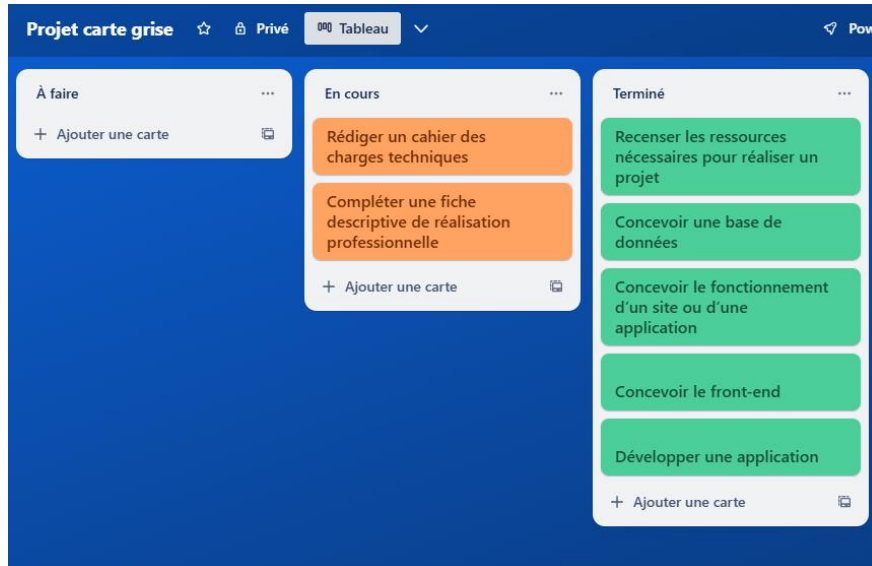
3.2. Ressources logicielles

- IDE : Visual Studio Code
- Plateforme de développement collaboratif : GitHub
- Serveur web : Apache (contenu dans MAMP)
- Système de Gestion de Base de Données relationnelle : MySQL (contenu dans MAMP)
- Outil de gestion de projet : Trello
- Conception UML et Arborescence : Visual Paradigm Online
- Maquettage : Figma

- Conception de la base de données : Mocodo

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

Sur ordinateur



Gestion des entités

AB-123-CD (Modèle : 208)

EF-456-GH (Modèle : Clio)

IJ-789-KL (Modèle : Yaris)

EH-573-BE (Modèle : Nexa)

Gestion des entités

AB-123-CD (Modèle : 208)

EF-456-GH (Modèle : Clio)

IJ-789-KL (Modèle : Yaris)

EH-573-BE (Modèle : Nexa)

Ajouter un Véhicule

?

Matrice 00-AAA-00:

Année de sortie:

Poids:

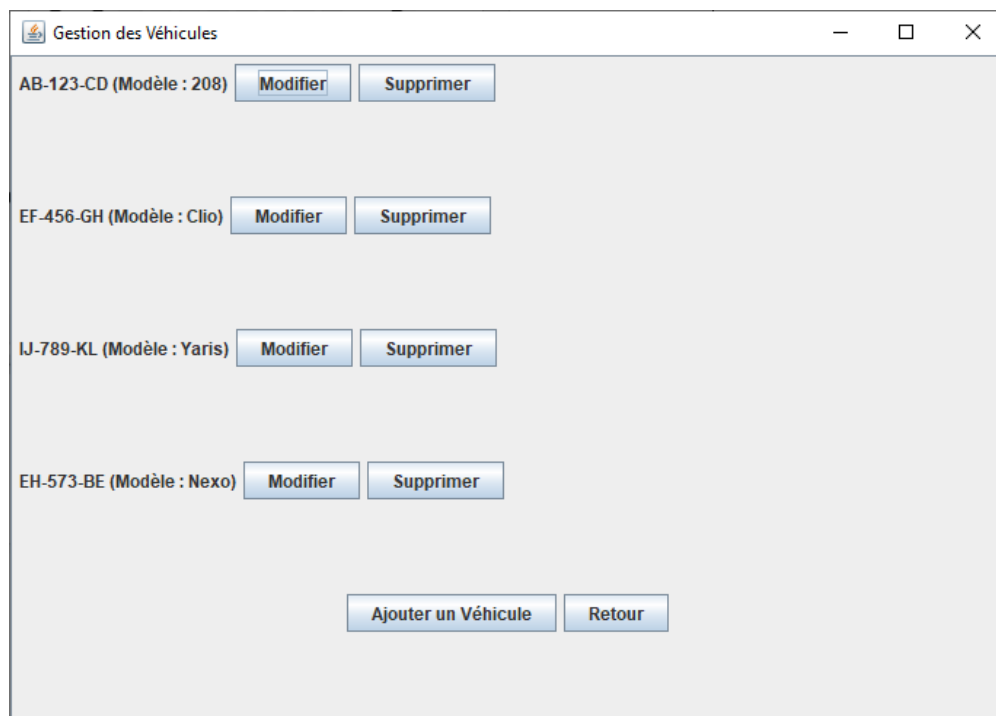
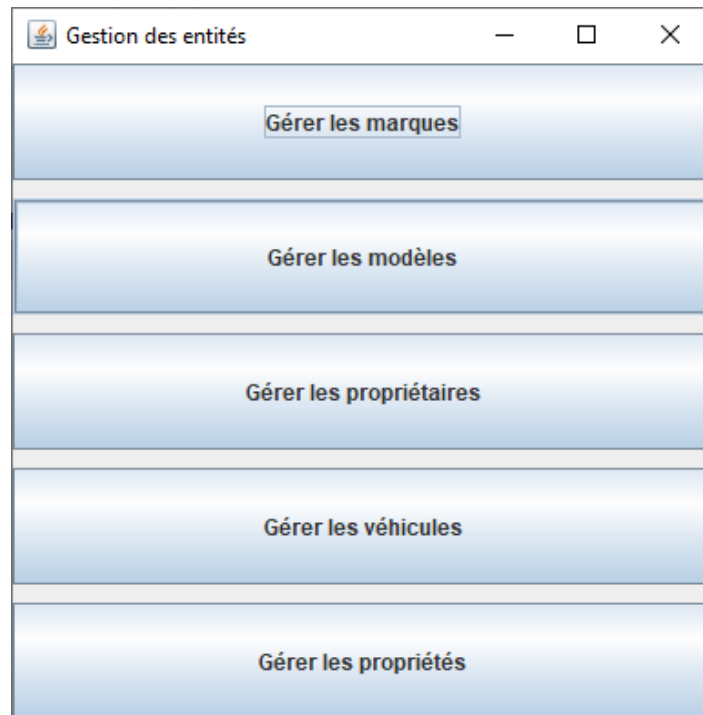
Puissance (chevaux):

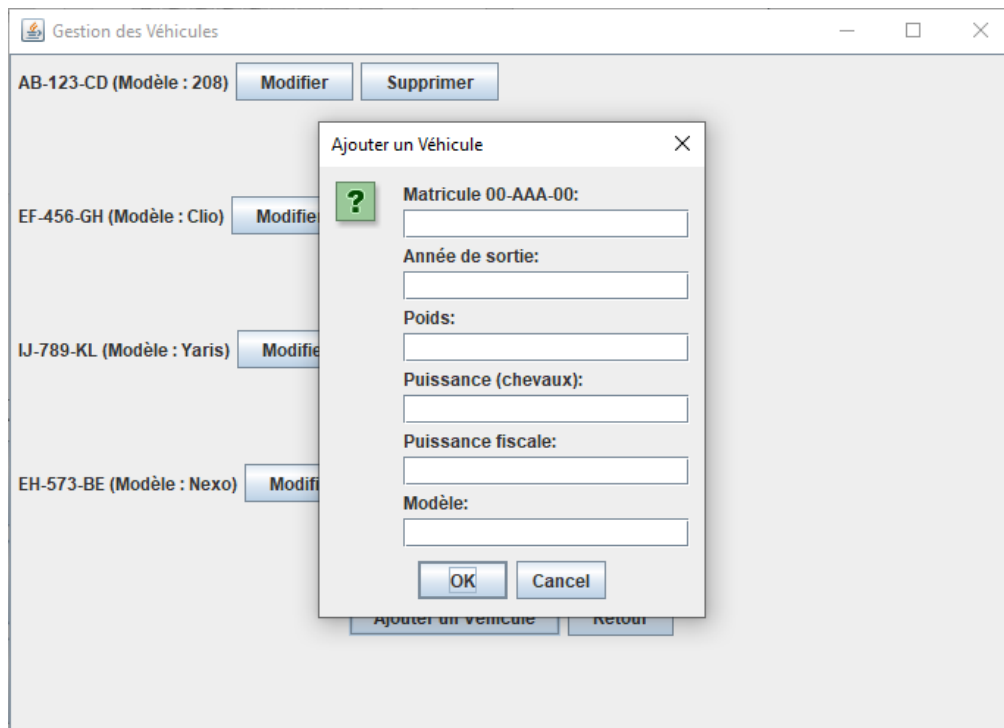
Puissance fiscale:

Modèle:

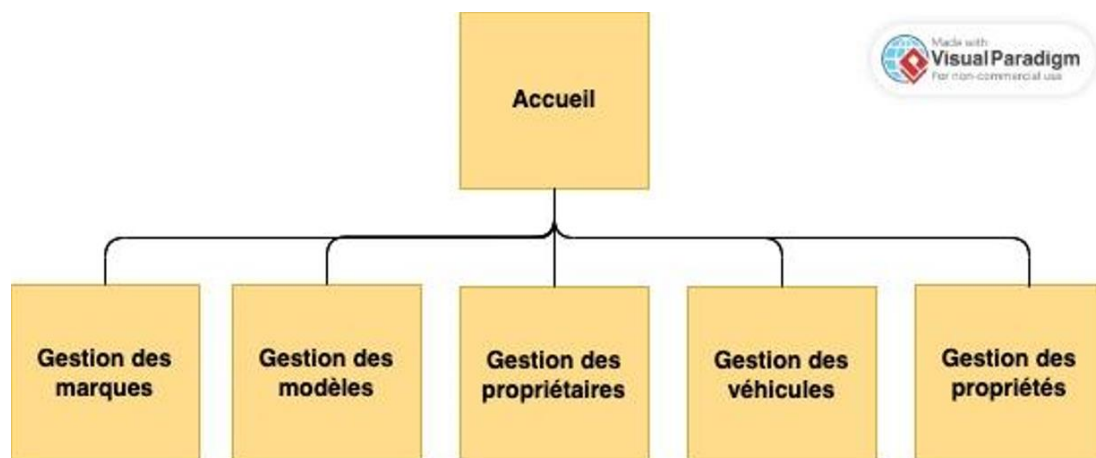
5.1.2. Maquettes

Sur ordinateur



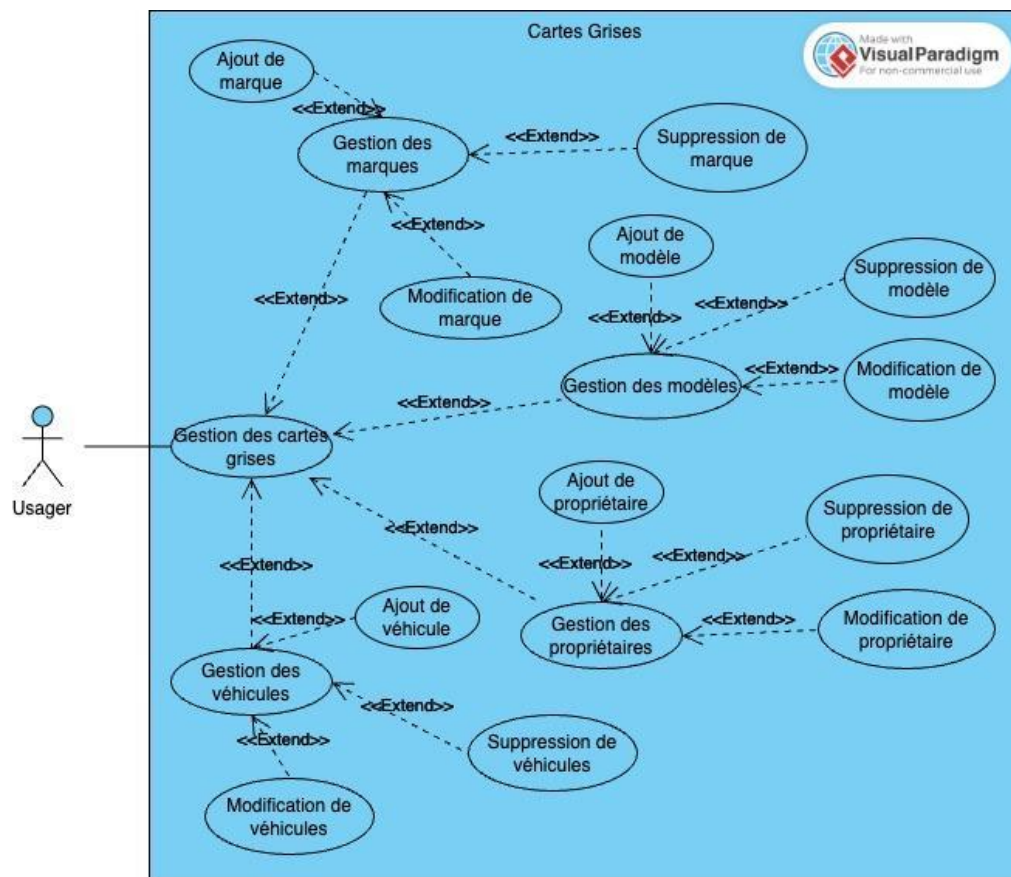


5.1.3. Arborescences

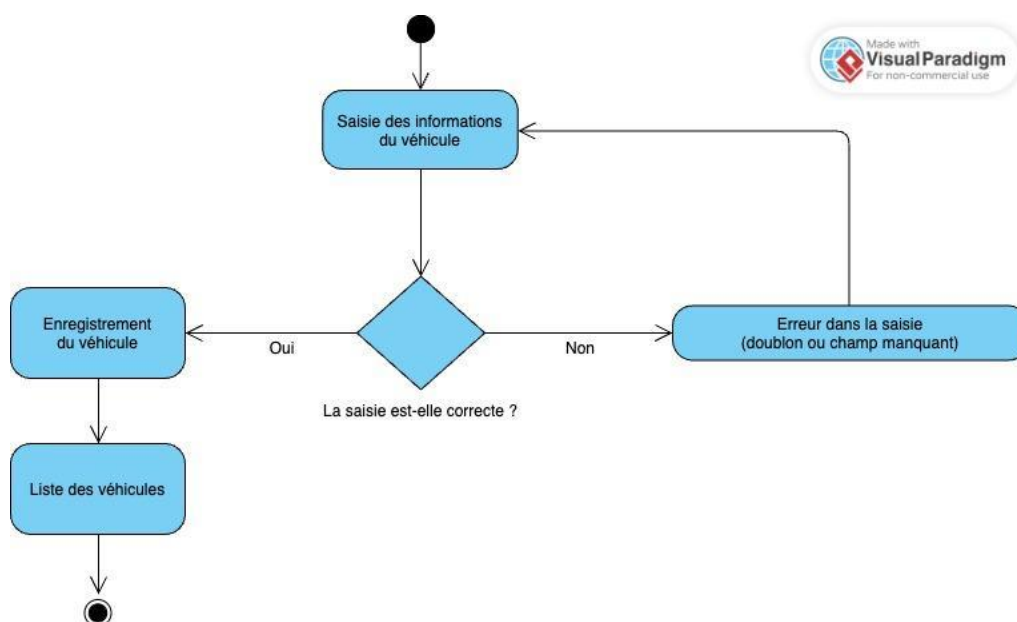


5.2. Le back-end

5.2.1. Diagramme de cas d'utilisation



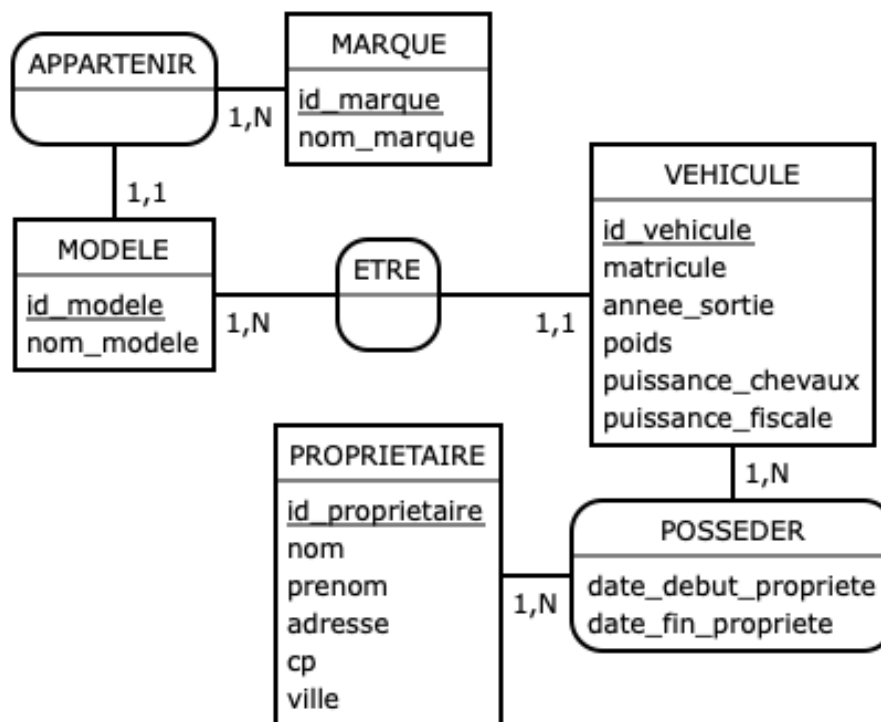
5.2.2. Diagramme d'activités



5.2.3. Dictionnaire des données

Table	Nom de l'attribut	Type	Description
VEHICULE	id_vehicule	INTEGER(4)	Identifiant unique du véhicule
VEHICULE	matricule	VARCHAR(255)	Matricule du véhicule
VEHICULE	annee_sortie	DATE	Année de sortie du véhicule
VEHICULE	poids	VARCHAR(255)	Poids du véhicule
VEHICULE	puissance_chevaux	VARCHAR(255)	Puissance chevaux du véhicule
VEHICULE	puissance_fiscale	VARCHAR(255)	Puissance fiscale du véhicule
VEHICULE	id_modele	INTEGER(4)	Identifiant du modèle (clé étrangère)
MODELE	id_modele	INTEGER(4)	Identifiant unique du modèle
MODELE	Nom_modele	VARCHAR(255)	Nom du modèle
MODELE	id_marque	INTEGER(4)	Identifiant du modèle (clé étrangère)
MARQUE	id_marque	INTEGER(4)	Identifiant unique de la marque
MARQUE	nom_marque	VARCHAR(255)	Nom de la marque
POSSEDER	id_vehicule	INTEGER(4)	Identifiant du véhicule (clé étrangère)
POSSEDER	id_proprietaire	INTEGER(4)	Identifiant du propriétaire (clé étrangère)
POSSEDER	date_debut_propriete	DATE	Date du début de la propriété
POSSEDER	date_fin_propriete	DATE	Date de fin de la propriété
PROPRIETAIRE	id_proprietaire	INTEGER(4)	Identifiant unique du propriétaire
PROPRIETAIRE	nom	VARCHAR(255)	Nom du propriétaire
PROPRIETAIRE	prenom	VARCHAR(255)	Prénom du propriétaire
PROPRIETAIRE	adresse	VARCHAR(255)	Adresse du propriétaire
PROPRIETAIRE	cp	VARCHAR(5)	Code Postal du propriétaire
PROPRIETAIRE	ville	VARCHAR(255)	Ville du propriétaire

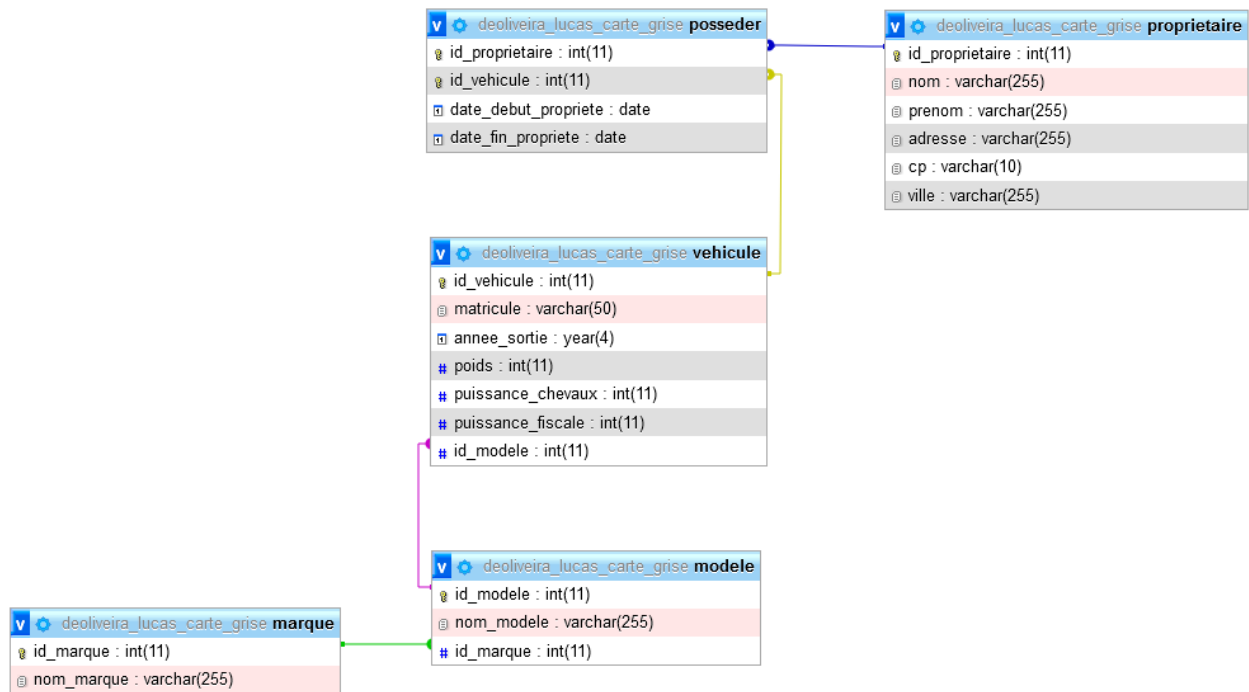
5.2.4. Modèles Conceptuel de Données (MCD)



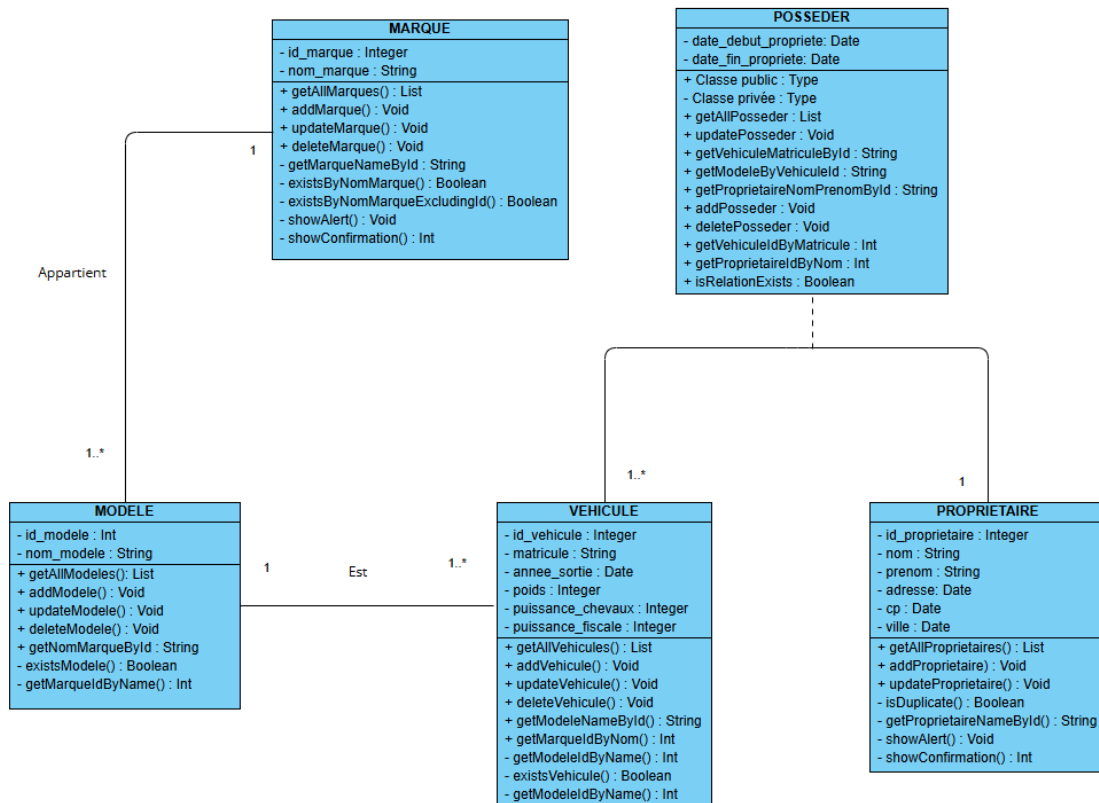
5.2.5. Modèle Logique de Données (MLD)

- VEHICULE (id_vehicule, matricule, annee_sortie, poids, puissance_chevaux, puissance_fiscale, #id_modele)
- MODELE (id_modele, nom_modele, #id_marque)
- MARQUE (id_marque, nom_marque)
- POSSEDER (#id_vehicule, #id_proprietaire, date_debut_propriete, date_fin_propriete)
- PROPRIETAIRE (id_proprietaire, nom, prenom, adresse, cp, ville)

5.2.6. Modèle Physique de Données (MPD)



5.2.6. Diagramme de classes



6. Technologies utilisées

6.1. Langages de développement

- Java : pour la mise en page et le design

6.2. Base de données

- Langage base de données : SQL

7. Sécurité

7.1. Cryptage des mots de passe avec Bcrypt

Bcrypt est un algorithme de hachage de mot de passe fort qui est résistant aux attaques par force brute. Il est plus lent que d'autres algorithmes, tels que SHA-256, ce qui rend plus difficile pour les attaquants de deviner les mots de passe.

- ```
bcrypt . genSalt (saltRounds , function (err, salt) {
 bcrypt . hash (myPlaintextPassword , salt , function (err , hash) {
 // Stockez le hachage dans votre base de données de mots de passe.
 }) ;
}) ;
```

### 7.2. Protection contre les attaques XSS (Cross-Site Scripting)

Le Cross-site Scripting ou attaque XSS est une vulnérabilité par injection de code. Pour détecter et exploiter ce type de menace, il est indispensable qu'un hacker injecte du code malveillant à partir des paramètres d'accès côté client.

- ```
<input type="hidden" name="csrf_token" value="<?php echo  
htmlspecialchars($_SESSION['csrf_token']); ?>">  
<label for="login">Login :</label>  
<input type="text" name="login" id="login" required><br><br>
```

7.3. Protection contre les injections SQL

Une injection SQL se produit lorsqu'un utilisateur malveillant communique une entrée qui modifie la requête SQL envoyée par l'application web à la base de données. Cela lui permet alors d'exécuter d'autres requêtes SQL non souhaitées directement sur la base de données.

Pour ce faire, l'attaquant doit injecter du code en dehors des limites de l'entrée utilisateur attendue, afin qu'il ne soit pas exécuté comme une entrée standard. Dans le cas le plus simple, il suffit d'injecter un guillemet simple ou double pour échapper aux limites de la saisie utilisateur et ainsi insérer des données directement dans la requête SQL.

- `SELECT * FROM users WHERE username= ' ' AND password= 'demo' ;`