

MARET-PRIGENT Mathis

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. 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 au 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 6 mars 2025.

2. Besoins fonctionnels

L'application devra être accessible en local permettant de gérer les données

Les données seront stockées dans une base de données relationnelle pour faciliter la gestion et la mise à jour des informations. Ces données peuvent être gérées directement via une application.

3. Ressources nécessaires à la réalisation du projet

3.1. Ressources matérielles

Les ressources matérielles sont:

- Ordinateur portable
- Ordinateur fixe (avec souris, clavier, écran et unité centrale)
- Connexion Ethernet ou Wifi

3.2. Ressources logicielles

Les ressources logicielles sont:

- Visual Studio Code (environnement de développement)
- Plateforme de développement collaborative: Github
- Système de gestion de base de données relationnels: MySQL (contenu dans MAMP)
- Outil de gestion de projet: Trello
- Conception d'UML et arborescence: Visual Paradigm Online
- Maquettage: Figma
- Conception de 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

- Java

5.1.1. Wireframes

Gérer les marques

Gérer les modèles

Gérer les propriétaires

Gérer les véhicules

Gérer les propriétés



5.1.2. Maquettes



Gérer les marques

Gérer les modèles

Gérer les propriétaires

Gérer les véhicules

Gérer les propriétés

 Liste des Marques

Peugeot

Modifier

Supprimer

Renault

Modifier

Supprimer


Toyota

Modifier

Supprimer

Ajouter une marque

Retour

 Gestion des Modèles

208 (Marque : Peugeot)

Modifier

Supprimer

Clio (Marque : Renault)

Modifier

Supprimer

Yaris (Marque : Toyota)

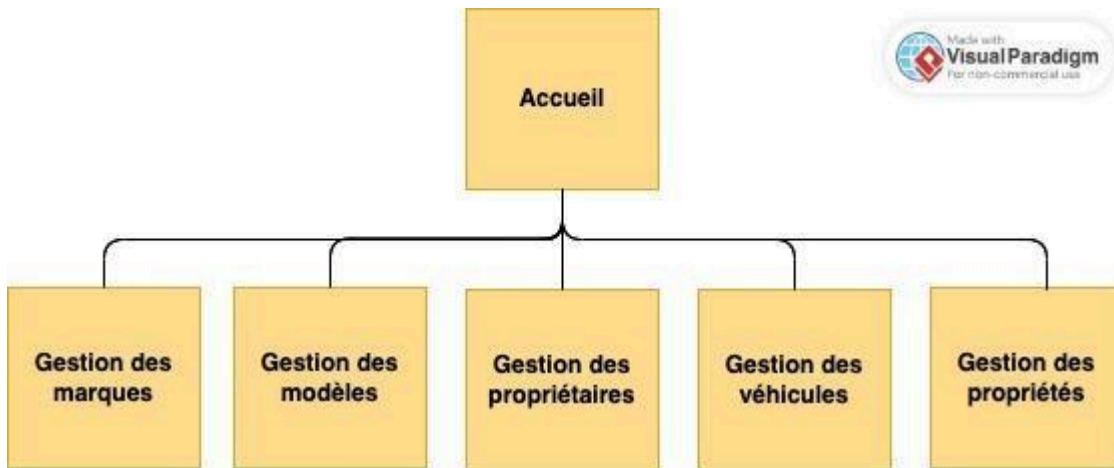
Modifier

Supprimer

Ajouter un Modèle

Retour

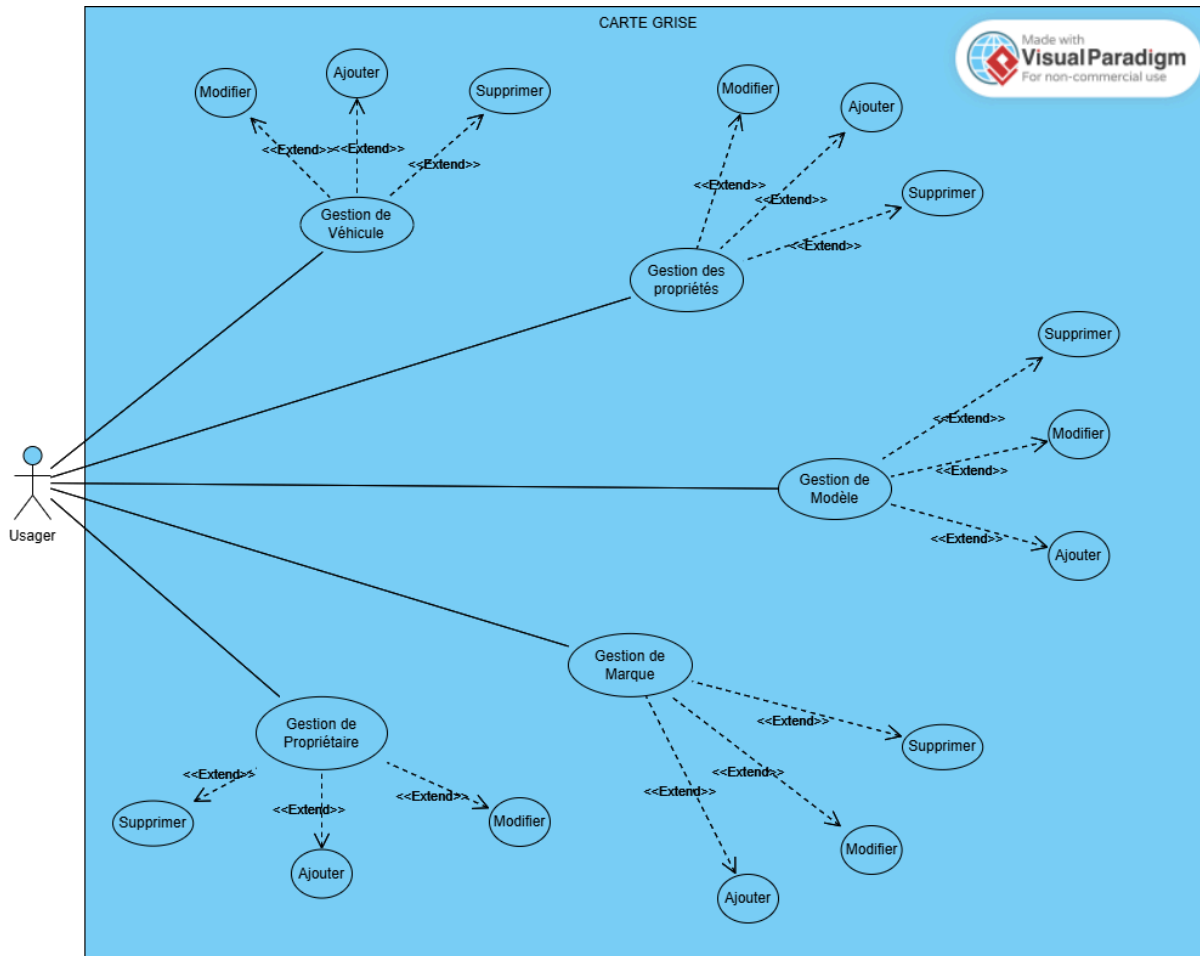
5.1.3. Arborescences



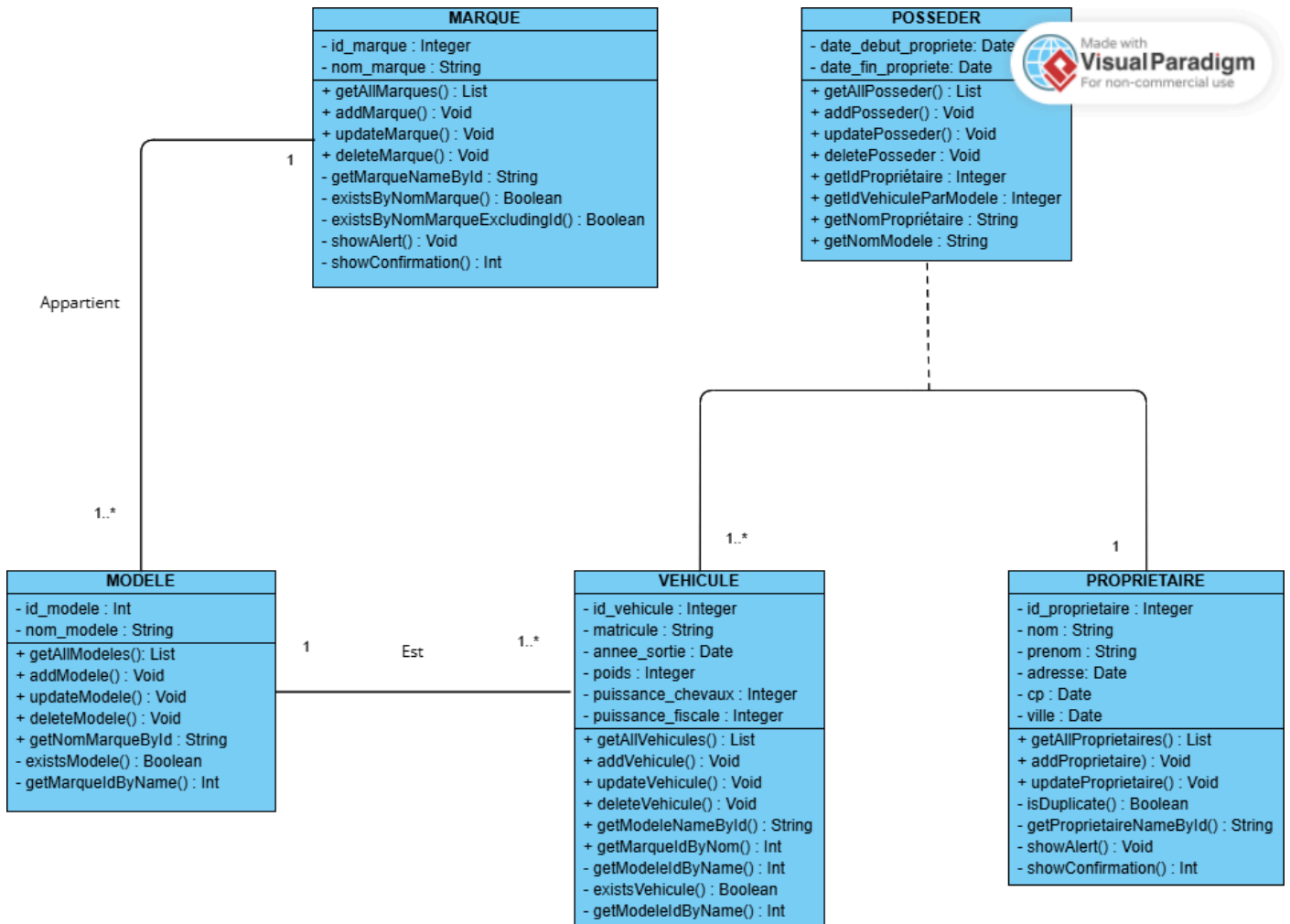
5.2. Le back-end

- Langage Interrogation base de données: SQL

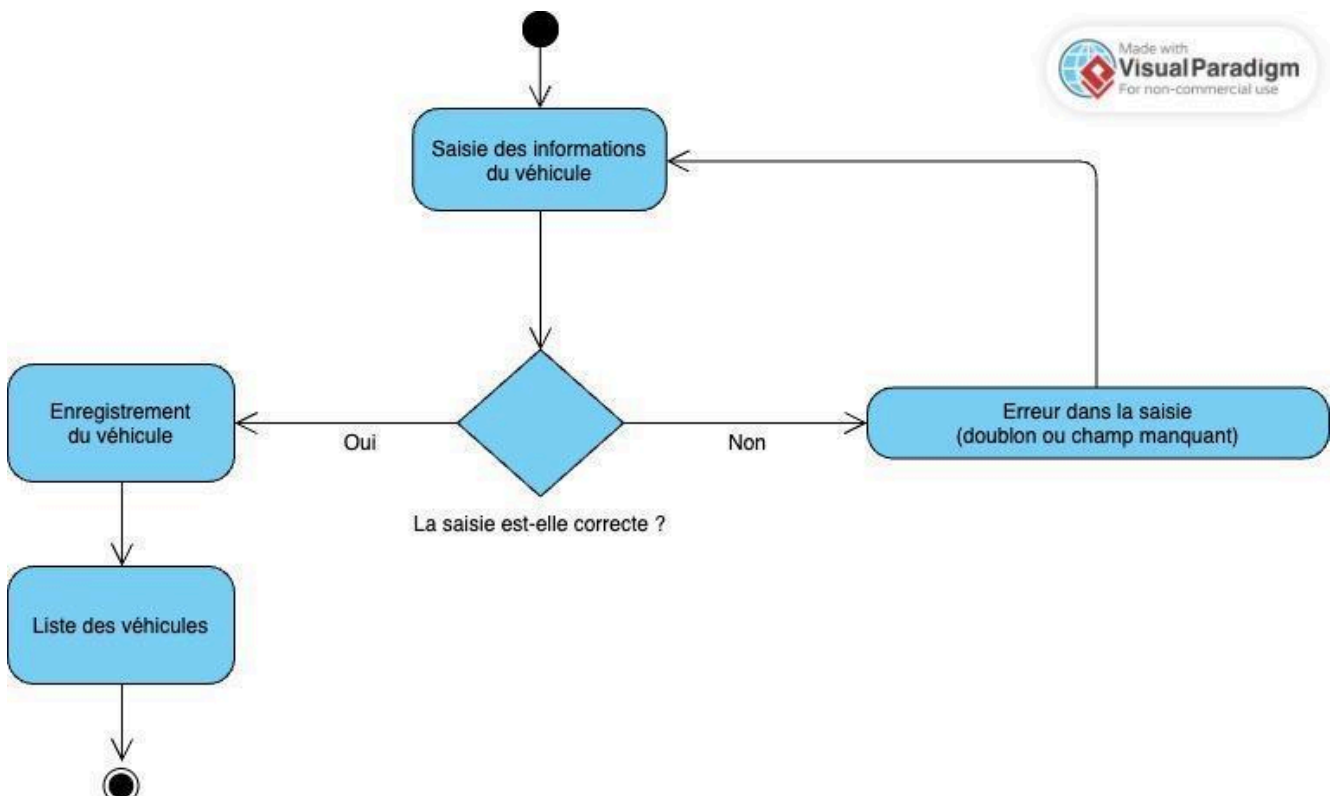
5.2.1. Diagramme de cas d'utilisation



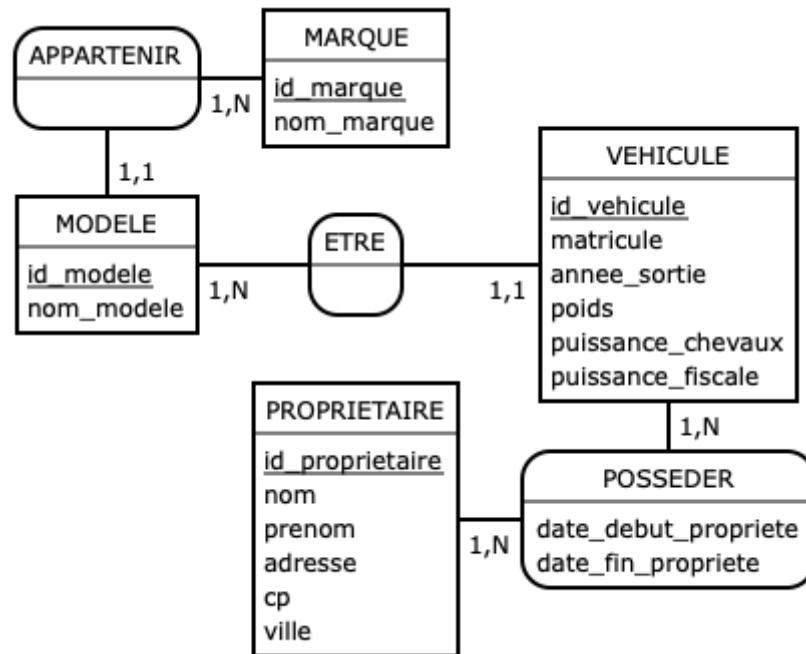
5.2.2. Diagramme de classe



5.2.3. Diagramme d'activités



5.2.4. Modèles Conceptuel de Données (MCD)

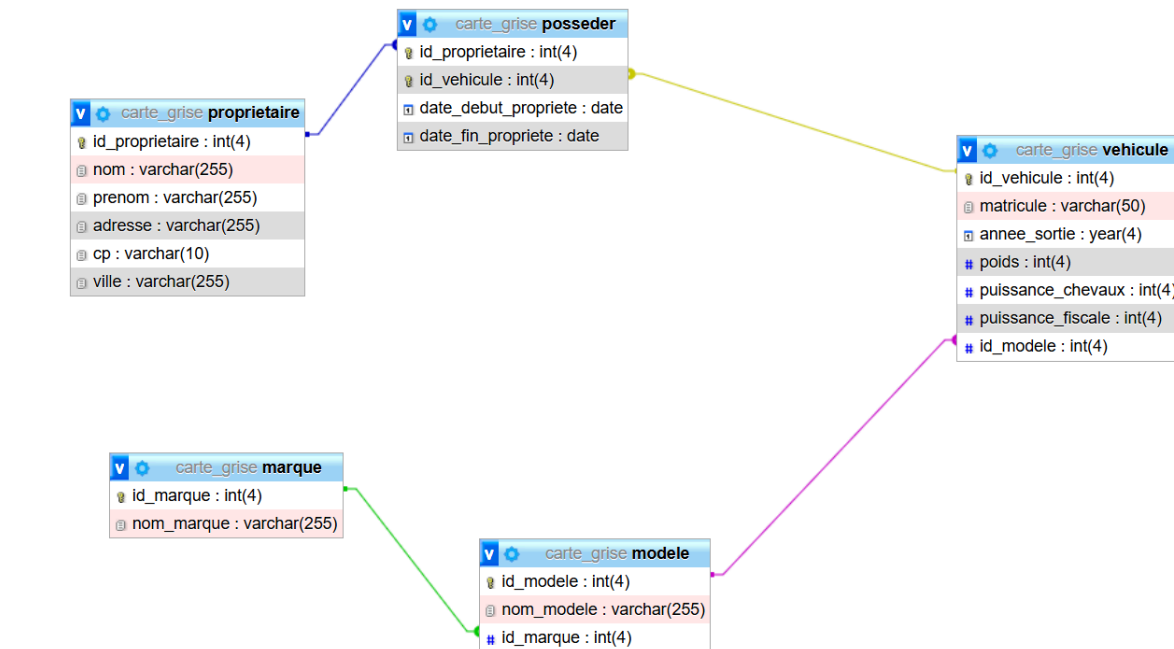


5.2.5. Modèle Logique de Données (MLD)

Version BTS:

- 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 référence 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 référence 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_vehicule, id_proprietaire
Clé étrangère: id_vehicule en référence à id_vehicule de vehicule
id_proprietaire en référence à id_proprietaire de proprietaire

5.2.6. Modèle Physique de Données (MPD)



6. Technologies utilisées

6.1. Langages de développement Web

Pour ce projet, le langage de développement web sera Visual Studio Code

6.2. Base de données

Notre base de données sera développée sur phpMyAdmin

7. Sécurité

7.1. Login et protection des pages administrateurs

Pour le login, on fera un formulaire en PHP avec la méthode POST pour pas que les informations saisies soient révélées dans la barre de recherche. Pour protéger cela on peut utiliser htmlspecialchars().

7.2. Cryptage des mots de passe avec Bcrypt

Expliquez à quoi sert Bcrypt comment vous faites en PHP.

Bcrypt est un algorithme de hachage sécurisé utilisé pour le cryptage des mots de passe. Il est conçu pour être résistant aux attaques par force brute grâce à deux mécanismes clés :

- **Salage** : Ajoute un élément aléatoire (sel) au mot de passe pour le rendre unique.
- **Facteur de coût** : Permet d'ajuster la difficulté du calcul, ralentissant les tentatives de piratage.

Pour le hachage, on utilise la commande `password_hash()` et pour la vérification d'un mot de passe on utilise la commande `password_verify()`.

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

Donnez la définition et expliquez comment vous faites en PHP.

Le XSS (Cross-Site Scripting) est une attaque qui permet d'injecter du code JavaScript malveillant dans une application web, pouvant voler des données ou manipuler l'affichage. Il existe trois types de XSS : stocké, réfléchi et basé sur le DOM. Pour s'en protéger, il faut échapper les entrées utilisateur (`StringEscapeUtils.escapeHtml4()` en Java), désactiver l'exécution de scripts non autorisés, utiliser un Content Security Policy (CSP), et valider les entrées avant de les afficher. Ces bonnes pratiques empêchent l'exécution de scripts malveillants et sécurisent l'application.

7.4. Protection contre les injections SQL

Les injections SQL sont des attaques où un utilisateur malveillant manipule des requêtes SQL pour accéder ou modifier des données sensibles dans une base de données. Comme solution on peut utiliser les requêtes préparées avec PDO, cela va empêcher toute modification notamment les malveillantes