Cahier des charges techniques

NSUKAMI Nathan
BTS SIO 2 – Projet Carte Grise

Sommaire

NSUKAMI Nathan
BTS SIO 2 – Projet Carte Grise

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 XX XXXX XX.

2. Besoins fonctionnels

L'application aura une partie publique pour soumettre des demandes de carte grise et une partie privée pour gérer ces demandes.

Les utilisateurs pourront entrer leurs informations et documents, tandis que les agents administratifs pourront valider et traiter les demandes via une interface dédiée.

Les données seront stockées dans une base de données relationnelle, permettant une gestion efficace et sécurisée des informations, accessible aux administrateurs.

3. Ressources nécessaires à la réalisation du projet

3.1. Ressources matérielles

Les ressources matérielles dont nous avons besoins sont :

- PC Fixe
- Connexion Internet (par câble ou wifi)
- Ecran
- Clavier
- Souris

3.2. Ressources logicielles

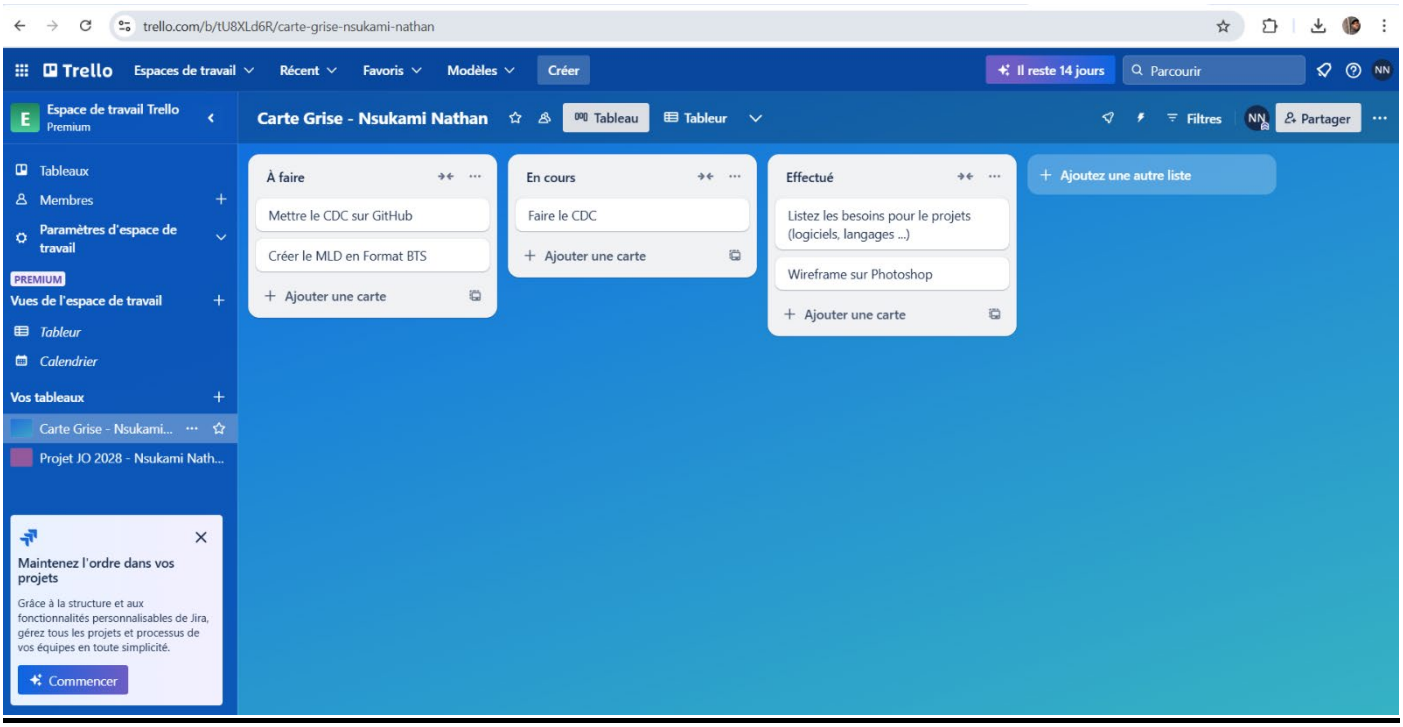
Ressources logicielles :

- IDE : Visual Studio Code
- Ensemble de solutions logicielles : MAMP : MySQL (gestionnaire de base de données relationnelle)
- Outil de modélisation de base de données : Mocodo
- Outil de modélisation UML : Visual Paradigm

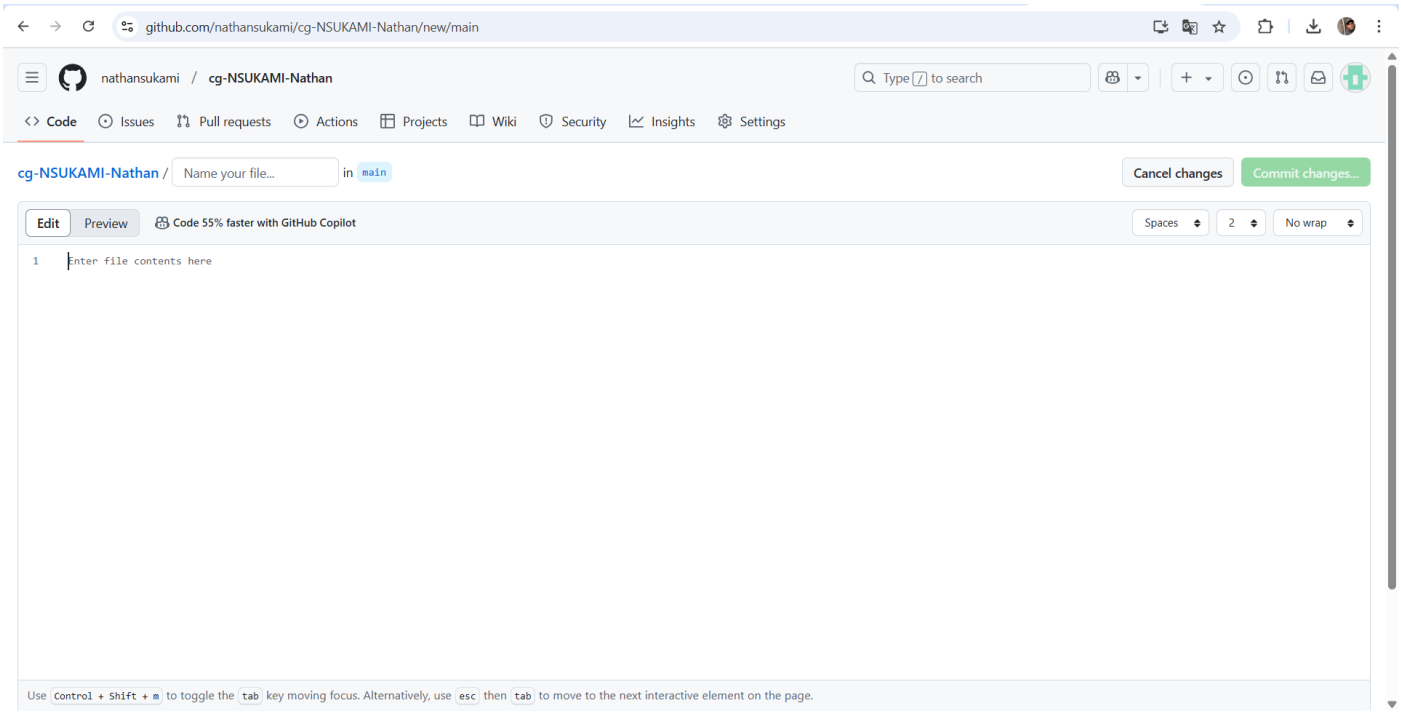
- Gestion de projet : Trello
- Outil de conception de design : Figma
- Plateforme de développement collaboratif : GitHub

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.

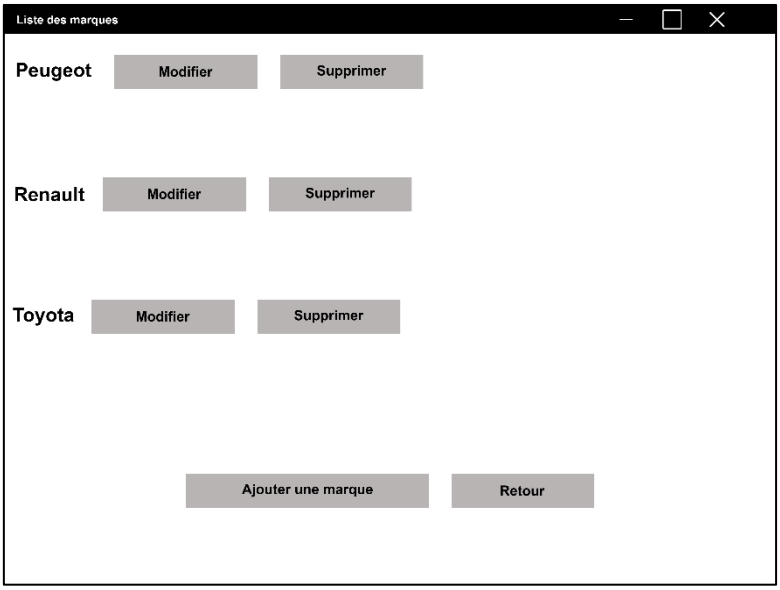


5. Conception du projet

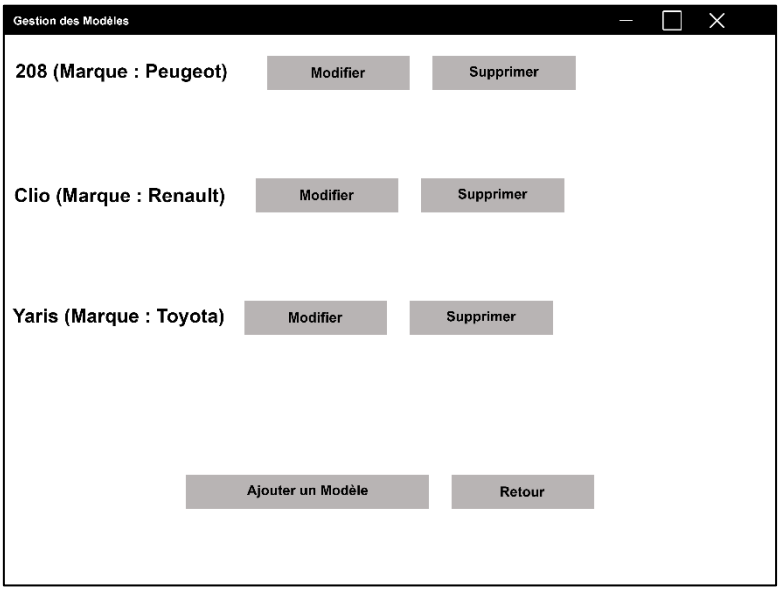
5.1. Le front-end

5.1.1. Wireframes

Wireframe de l'ordinateur pour la page Accueil



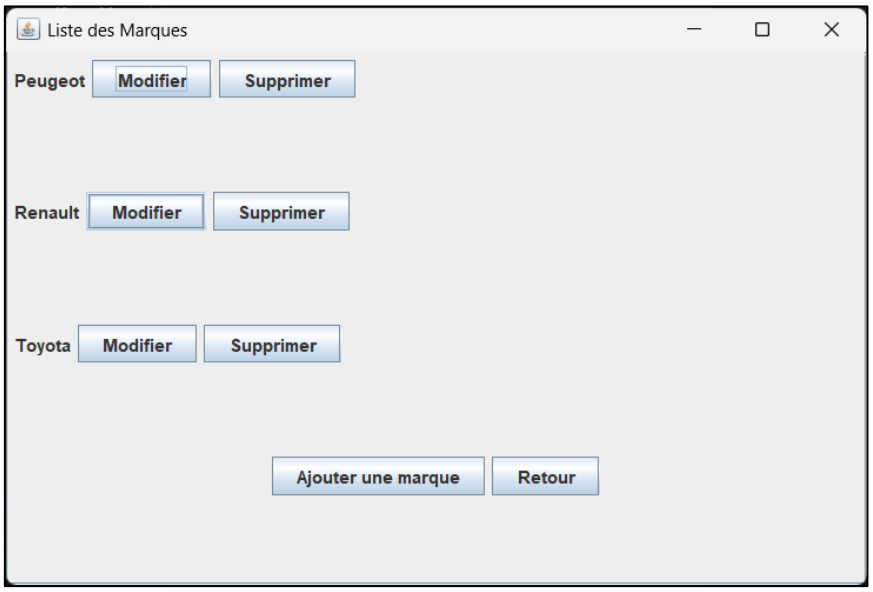
Wireframe de l'ordinateur pour la page Liste des Marques



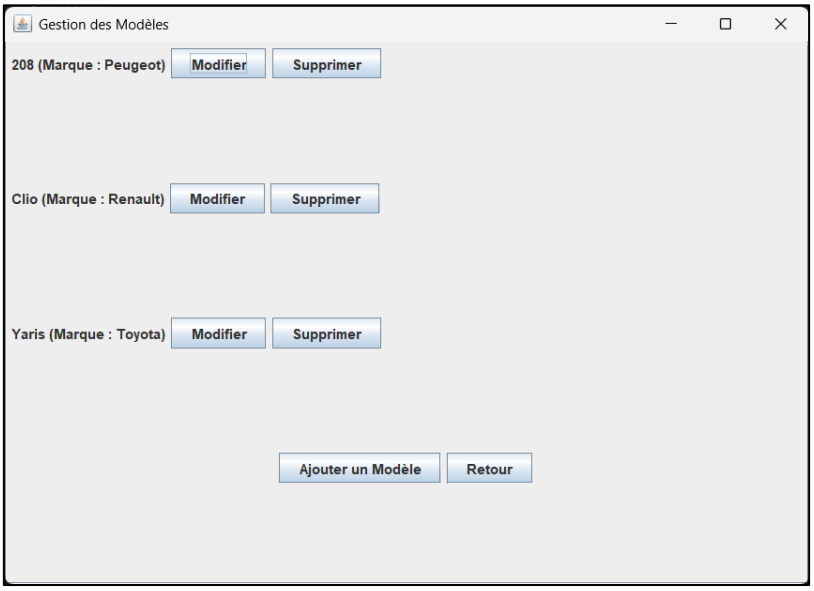
Wireframe de l'ordinateur pour la page Gestion des Modèles

5.1.2. Maquettes

Maquette de l'ordinateur pour la page Accueil

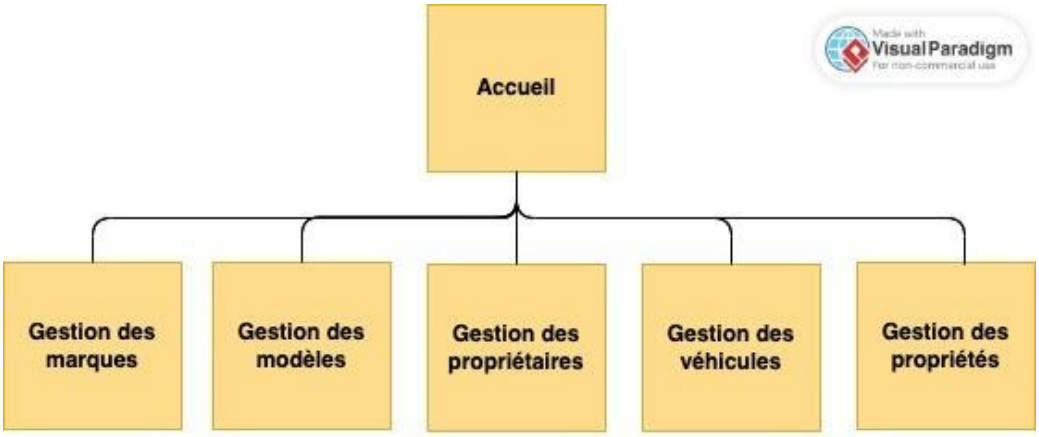


Maquette de l'ordinateur pour la page Liste des Marques



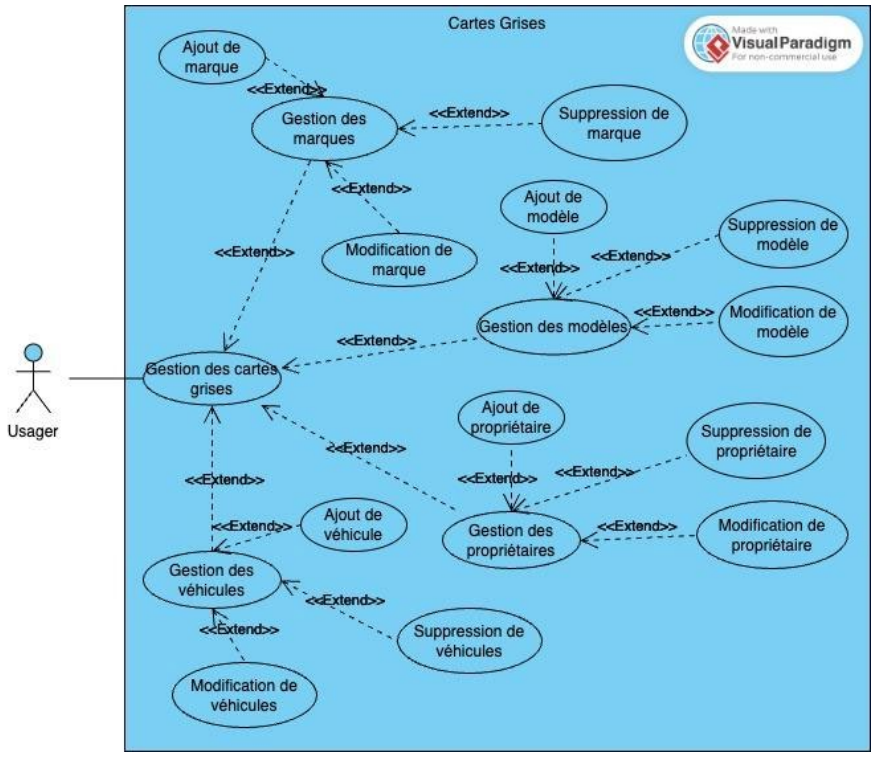
Maquette de l'ordinateur pour la page Gestion des Modèles

5.1.3. Arborescences

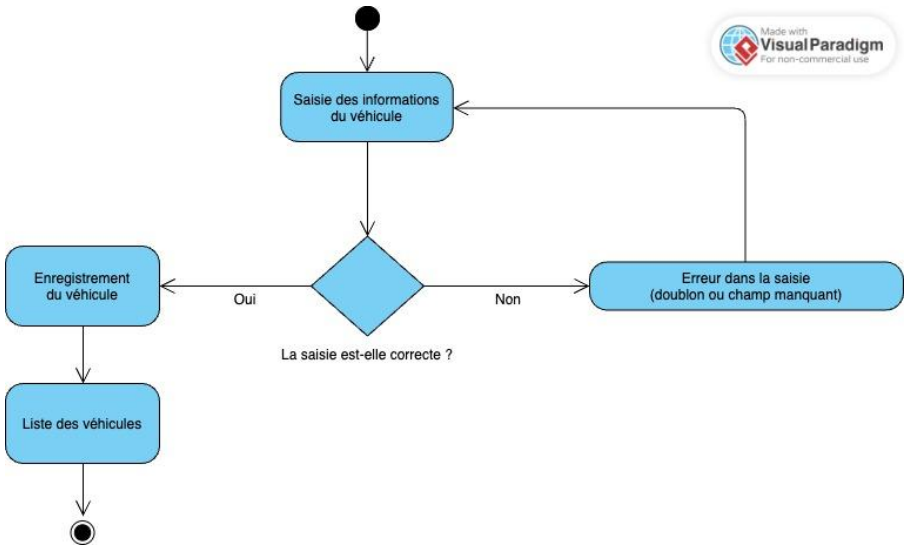


5.2. Le back-end

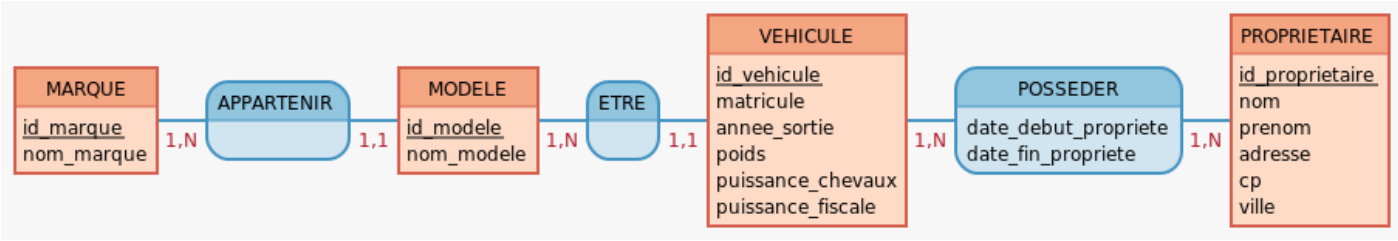
5.2.1. Diagramme de cas d'utilisation



5.2.2. Diagramme d'activités



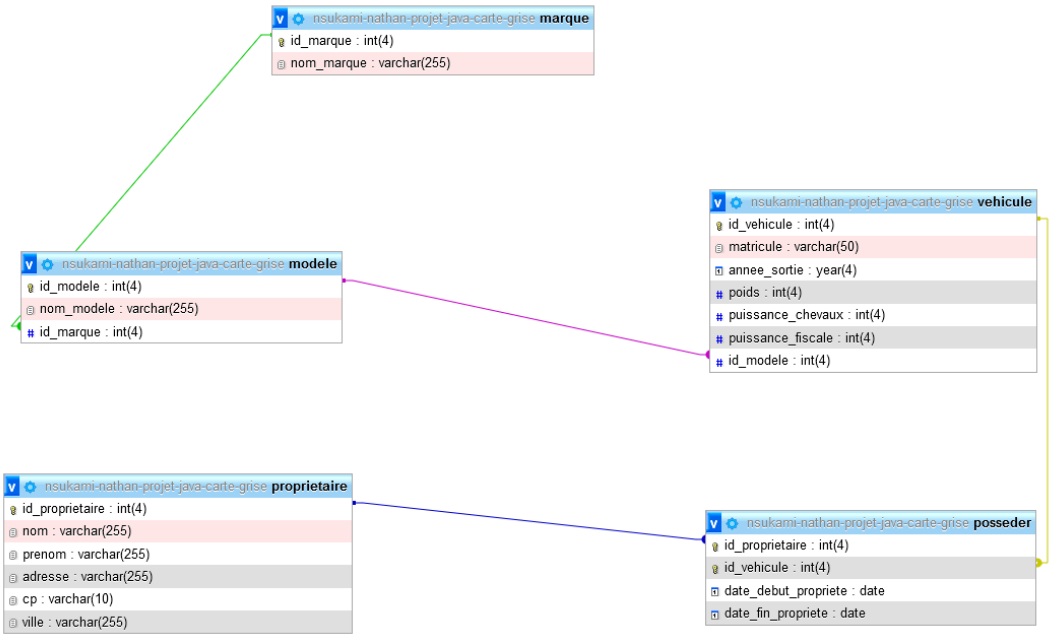
5.2.3. Modèles Conceptuel de Données (MCD)



5.2.4. Modèle Logique de Données (MLD)

- 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
- POSSEDER (id_vehicule, id_proprietaire, date_debut_propriete, date_fin_propriete)
    - Clé primaire : id_vehicule en référence à id_vehicule de VEHICULE, id_proprietaire en référence à id_proprietaire de PROPRIETAIRE
    - Clé étrangère : id_vehicule en référence à id_vehicule de VEHICULE, id_proprietaire en référence à id_proprietaire de PROPRIETAIRE
- PROPRIETAIRE (id_proprietaire, nom, prenom, adresse, cp, ville)
- 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

5.2.5. Modèle Physique de Données (MPD)



5.2.6. Diagramme de classe

6. Technologies utilisées

6.1. Langages de développement d’application

Les différents langages de développement Web que nous avons utilisés pour la réalisation du site web sont :

- HTML
- CSS
- JAVA SCRIPT

6.2. Base de données

Pour créer et gérer la base de données, nous avons utilisés le langage de programmation SQL par le biais de MySQL de MAMP

7. Sécurité

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

La protection contre les attaques XSS en Java consiste à empêcher l'injection de code JavaScript malveillant. Voici les principales mesures :

- Validation et échappement des entrées utilisateurs : Vérifiez et échappez les caractères spéciaux (<, >, &, etc.) pour éviter l'exécution de code malveillant.
- Utilisation de frameworks sécurisés : Utilisez des frameworks comme JSF, Spring Security, ou Thymeleaf qui gèrent l'échappement des données automatiquement.
- Content Security Policy (CSP) : Mettez en place une politique CSP pour restreindre les sources de scripts.
- Manipulation sécurisée du DOM : Évitez d'utiliser innerHTML ou document.write pour insérer du contenu non échappé.
- Cookies sécurisés : Utilisez les attributs HttpOnly et SameSite pour sécuriser les cookies.

Ces pratiques permettent de protéger les applications Java contre les attaques XSS.

7.2. Protection contre les injections SQL

La protection contre les injections SQL en Java repose sur ces mesures clés :

- Requêtes préparées (PreparedStatement) : Utilisez-les pour séparer le code SQL des données utilisateurs, empêchant l'injection.
- Éviter la concaténation : Ne concaténez pas directement les données utilisateur dans les requêtes SQL.
- Validation des entrées : Vérifiez les données saisies par les utilisateurs pour éviter des valeurs malveillantes.
- Procédures stockées : Utilisez-les pour isoler la logique SQL côté serveur.
- Limitation des privilèges : Restreignez les droits d'accès à la base de données pour minimiser les risques.

Ces pratiques sécurisent les applications contre les injections SQL.