

# CAHIER DES CHARGES



## Projet Carte Grises

CHARNI Louay

# SOMMAIRE

## I. Contexte du projet

1. Présentation du projet
2. Objectifs
3. Date de rendu du projet

## II. Besoins fonctionnels

## III. Ressources nécessaires à la réalisation du projet

4. Besoins matériels
5. Besoins logiciels

## IV. Gestion du projet

## V. Wireframes

## VI. Maquettes

## VII. Arborescences

## VIII. Le back-end

6. Diagramme de cas d'utilisation
7. Diagrammes d'activités
8. MCD (Modèle Conceptuel de Données)
9. MLD (Modèle Logique de Données)
10. MPD (Modèle Physique de Données)

## IX. Technologies utilisées

11. Langages de développement
12. Base de données

## X. Sécurité

13. Login et protection des pages administrateurs
14. Cryptage des mots de passe avec Bcrypt
15. Protection contre les attaques XSS
16. Protection contre les injections SQL

## I. Contexte du projet

Vous travaillez en tant que développeur au sein de la Direction des Systèmes d'Information de la préfecture de votre département.

Le service des cartes grises fait face à une gestion de plus en plus complexe des demandes d'immatriculation, entraînant des délais de traitement longs et une forte sollicitation des agents.

Face à ces défis, la préfecture a décidé de moderniser son système en développant un logiciel de gestion des cartes grises.

Ce nouvel outil devra permettre un suivi simplifié des demandes, une automatisation de certaines tâches administratives et une meilleure gestion des informations liées aux véhicules et aux propriétaires.

## Date de rendu du projet :

Le projet doit être rendu au plus tard le 05/12/2024.

## II. Besoins fonctionnels

L'application devra permettre :

- La gestion des marques et modèles de voitures.
- La gestion des propriétaires et des véhicules.
- L'attribution et mise à jour des propriétés.

## III. Ressources nécessaires à la réalisation du projet

### Besoins matériels :

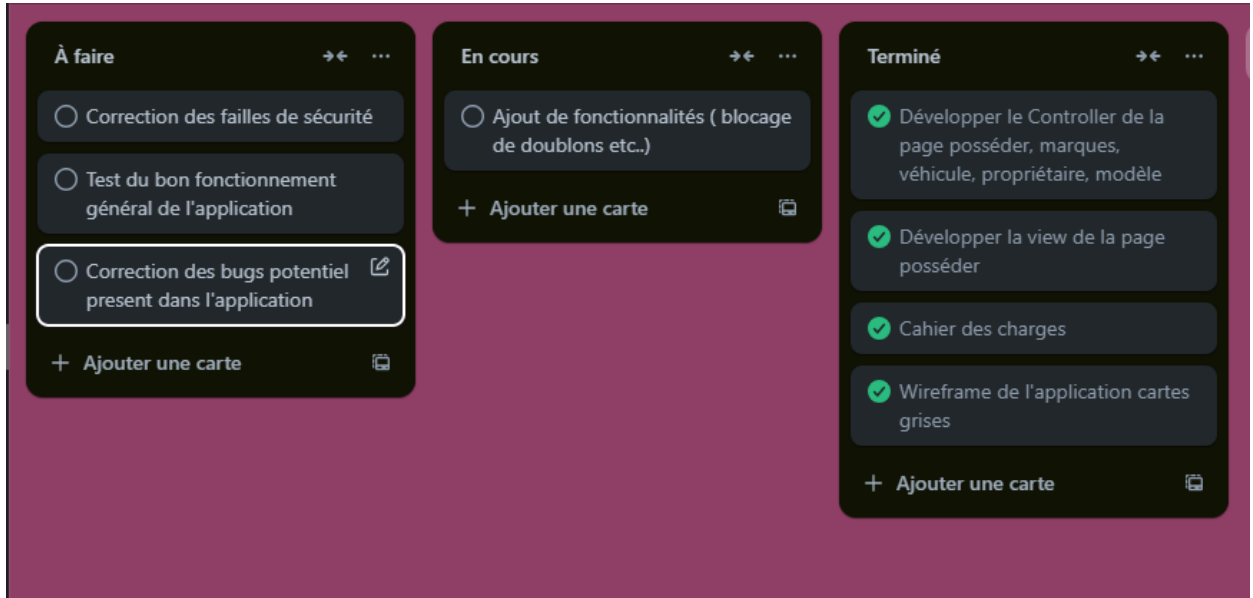
- Ordinateur portable connecté à Internet
- Ordinateur fixe connecté à Internet

### Besoins logiciels :

- IDE : Visual Studio Code
- Plateforme collaborative : Github
- Outils de gestion de projet : Trello
- Conception UML : Visual Paradigm Online
- Maquettage : Figma
- Conception BDD : Mocodo
- Documentation : StackOverflow/ IA

## IV. Gestion du projet

Le projet est géré à l'aide de Trello, avec des tableaux organisés par tâches et priorités.



## V. Wireframes



---

208 (Marque : Peugeot)

Modifier

Supprimer

Clio (Marque : Renault)

Modifier

Supprimer

Yaris (Marque : Toyota)

Modifier

Supprimer

Ajouter un Modèle

Retour

---

Peugeot

Modifier

Supprimer

Renault

Modifier

Supprimer

Toyota

Modifier

Supprimer

Ajouter une marque

Retour

Doe John (123 Rue de paris, 75000 Paris)

Modifier

Supprimer

Doe Jane (456 Avenue des champs, 75008 Paris)

Modifier

Supprimer

Smith Alice (789 Boulevard Haussmann, 75009 Paris)

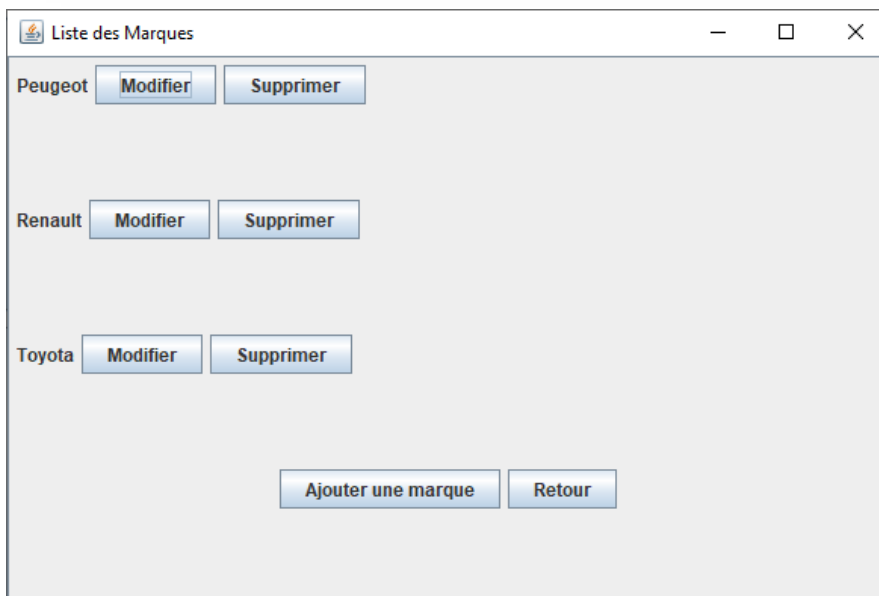
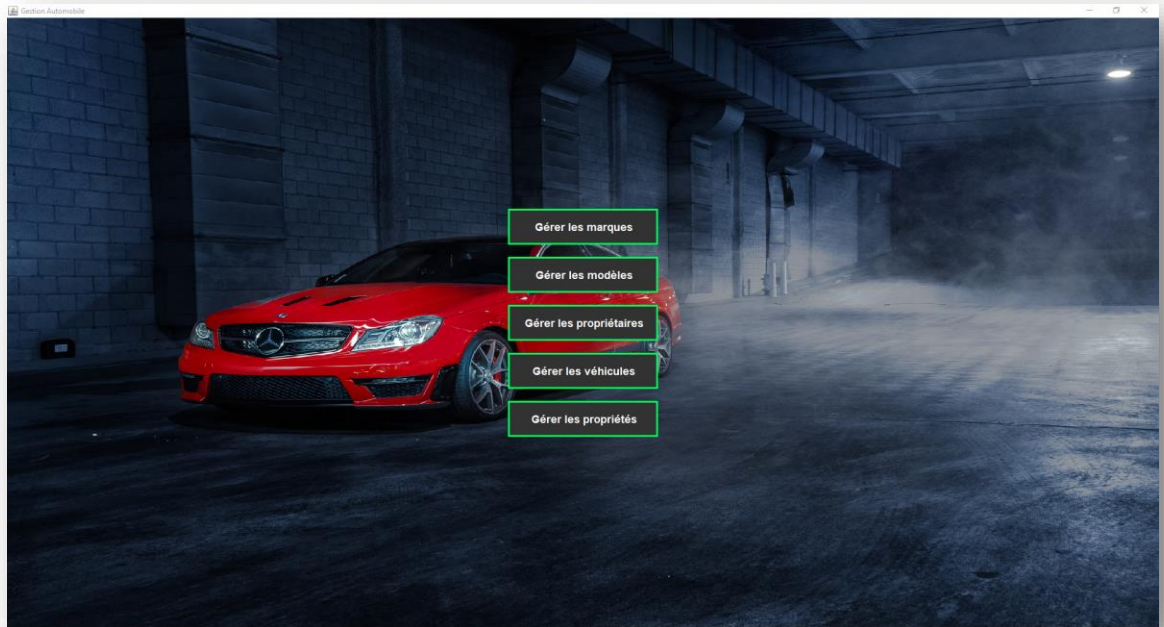
Modifier

Supprimer


Ajouter un Propriétaire

Retour

## VI. Maquettes





 Gestion des Modèles

206 (Marque : Peugeot)

Modifier

Supprimer

Clio (Marque : Renault)

Modifier

Supprimer


Yaris (Marque : Toyota)

Modifier

Supprimer

Ajouter un Modèle

Retour

 Liste des Propriétaires

Doe John (123 Rue de Paris, 75000 Paris)

Modifier

Supprimer

Doe Jane (456 Avenue des Champs, 75008 Paris)

Modifier

Supprimer

Smith Alice (789 Boulevard Haussmann, 75009 Paris)

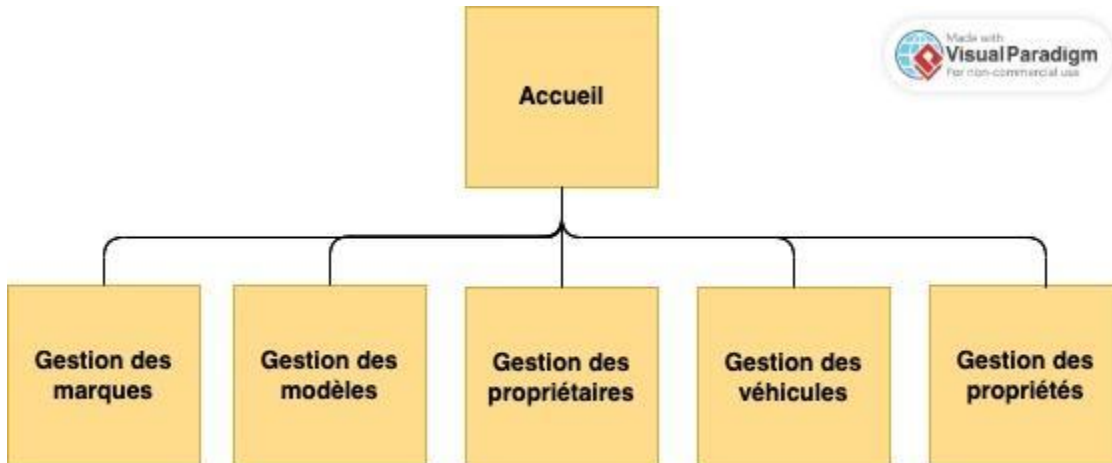
Modifier

Supprimer

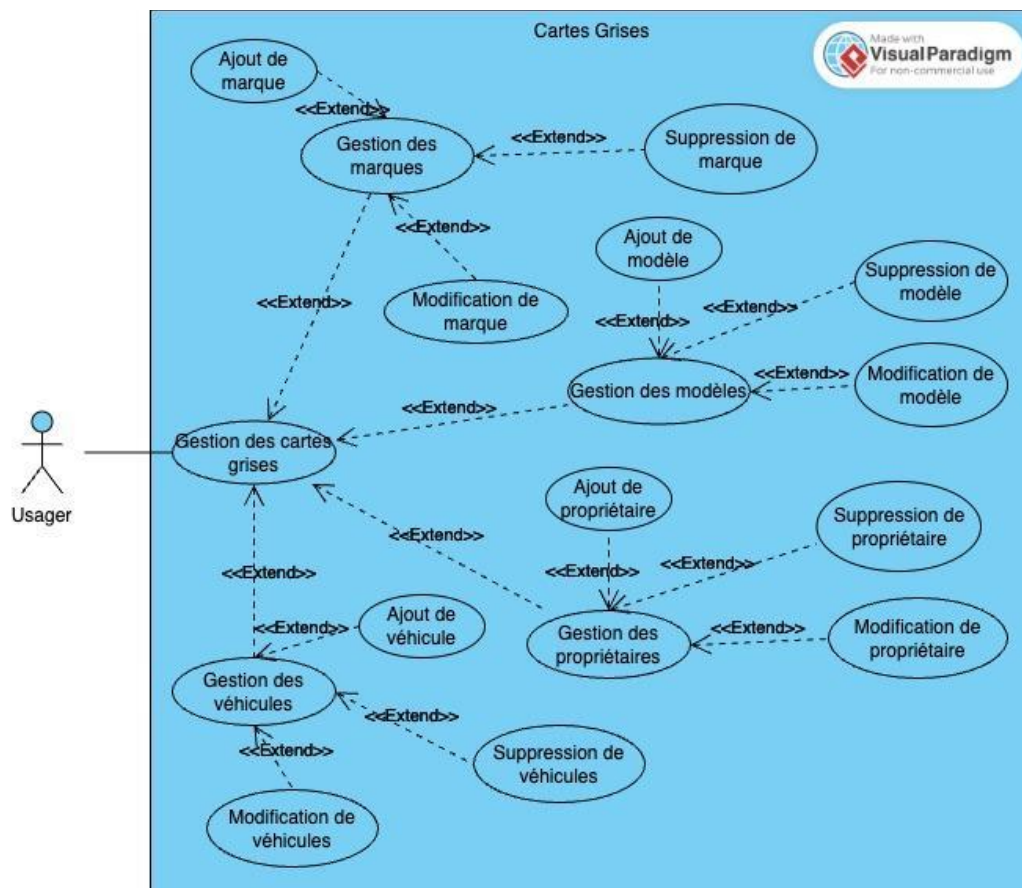
Ajouter un propriétaire

Retour

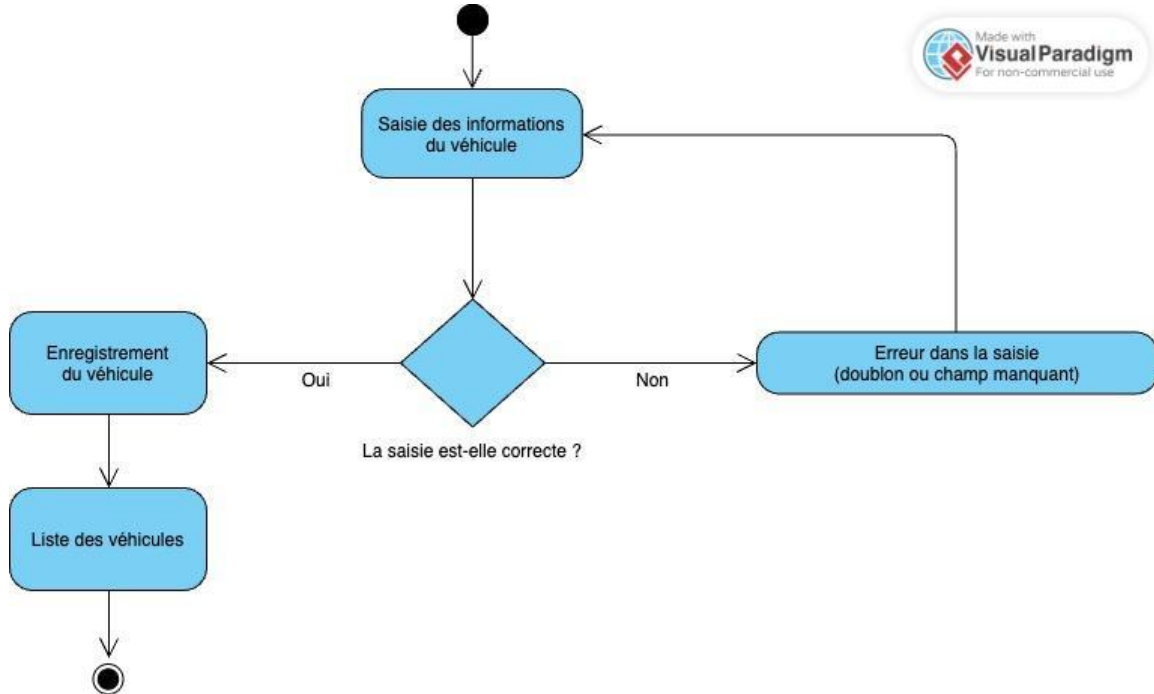
## VII. Arborescences



## Diagramme de cas d'utilisation



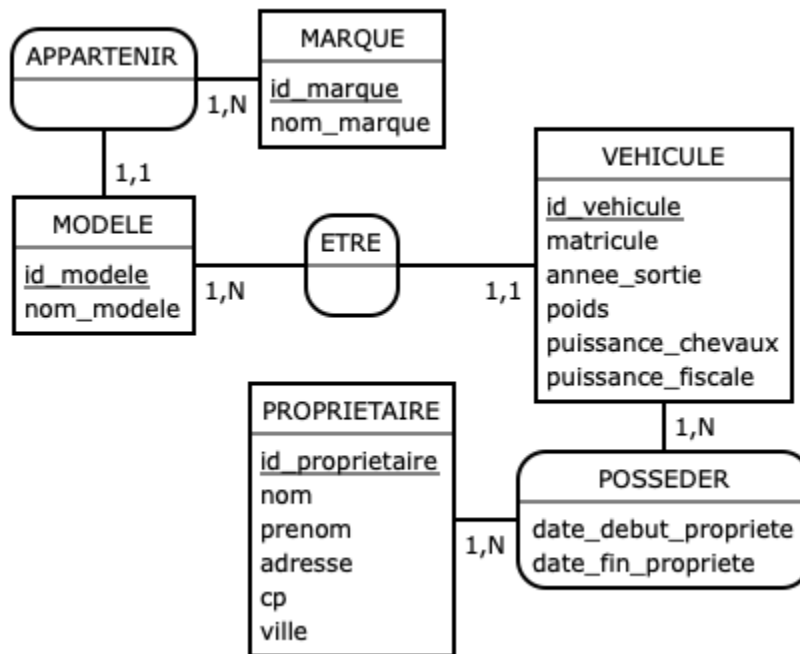
## Diagrammes d'activités (ajout des vehicules)



## Dictionnaire de donnée

Entité ou association	Libellé de l'attribut	Type
MARQUE	id_marque	INT(4)
	nom_marque	VARCHAR(255)
MODELE	id_modele	INT(4)
	nom_modele	VARCHAR(255)
	id_marque	INT(4)
VEHICULE	id_vehicule	INT(4)
	matricule	VARCHAR(50)
	annee_sortie	VARCHAR(255)
	poids	VARCHAR(255)
	puissance_chevaux	VARCHAR(255)
	puissance_fiscale	VARCHAR(255)
	id_modele	INT(4)
POSSEDER	id_proprietaire	INT(4)
	id_vehicule	INT(4)
	date_debut_propriete	DATE
	date_fin_propriete	DATE
PROPRIETAIRE	id_proprietaire	INT(4)
	nom	VARCHAR(255)
	prenom	VARCHAR(255)
	adresse	VARCHAR(255)
	cp	VARCHAR(10)
	ville	VARCHAR(255)

## Modèles Conceptuel de Données (MCD)



## 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

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

POSSEDER(id\_proprietaire, id\_vehicule, date\_debut\_propriete, date\_fin\_propriete)

Clé primaire : id\_proprietaire, id\_vehicule

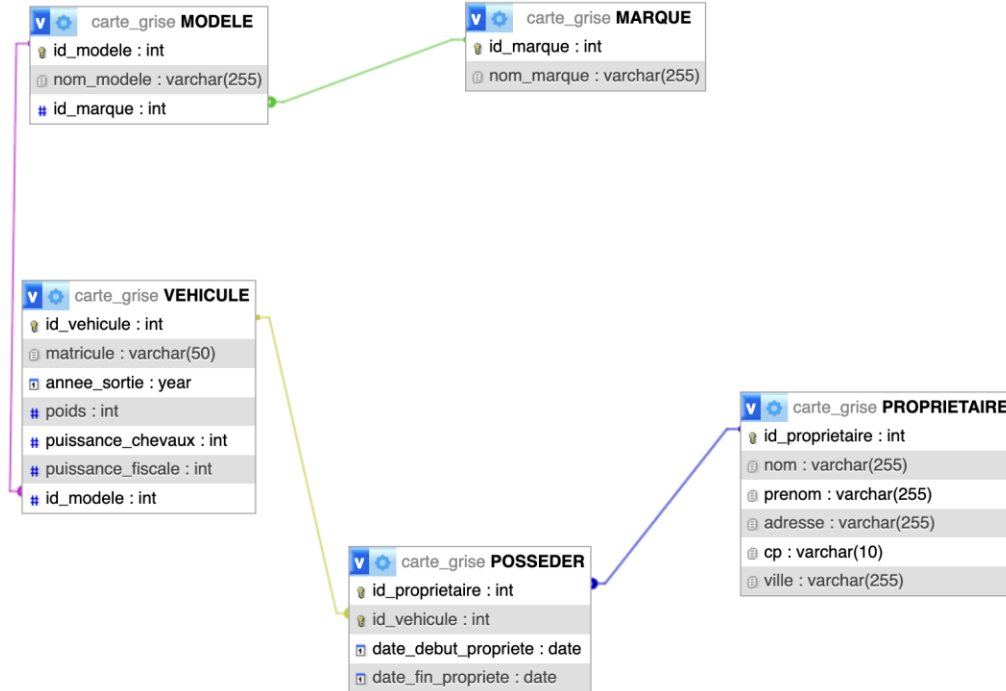
Clé étrangère : id\_proprietaire en référence à id\_proprietaire de PROPRIETAIRE

id\_vehicule en référence à id\_vehicule de VEHICULE

PROPRIETAIRE(id\_proprietaire, nom, prenom, adresse, cp, ville)

Clé primaire : id\_proprietaire

# Modèle Physique de Données (MPD)



## IX. Technologies utilisées

Langages de développement :

- Front-end : Java
- Back-end : Java

Base de données :

- Serveur web : Apache (MAMP)
- SGBD : MySQL (MAMP)

## X. Sécurité

Login et protection des pages administrateurs :

- HTTPS, 2FA, limitation des tentatives, méthodes POST.

Cryptage des mots de passe avec Bcrypt :

- Hashage, salage, facteur de coût élevé.

Protection contre les attaques XSS :

- Filtrage, échappement, Content Security Policy (CSP).

Protection contre les injections SQL :

- Requêtes préparées, échappement des entrées, droits limités.