

Cahier des charges techniques

Sommaire

1.	Contexte du projet	2
1.1.	Présentation du projet	2
1.2.	Date de rendu du projet.....	2
2.	Besoins fonctionnels.....	2
3.	Ressources nécessaires à la réalisation du projet	2
3.1.	Ressources matérielles	2
3.2.	Ressources logicielles.....	2
	Gestion du projet.....	2
4.	Conception du projet	3
5.1.	Le front-end.....	3
5.1.1.	Wireframes	3
5.1.2.	Maquettes.....	3
5.1.3.	Arborescences.....	5
5.2.	Le back-end	5
5.2.1.	Diagramme de cas d'utilisation	5
5.2.2.	Diagramme d'activités	6
5.2.3.	Modèles Conceptuel de Données (MCD)	6
5.2.4.	Modèle Logique de Données (MLD)	7
5.2.5.	Modèle Physique de Données (MPD)	7
5.	Technologies utilisées	8
6.1.	Langages de développement	8
6.2.	Base de données	8
6.	Sécurité	8
7.1.	Login et protection des pages administrateurs	Error! Bookmark not defined.
7.2.	Cryptage des mots de passe avec Bcrypt	8
7.3.	Protection contre les attaques XSS (Cross-Site Scripting)	9
7.4.	Protection contre les injections SQL.....	9

1.Contexte du projet

1.1. Présentation du projet

En tant que concepteur et développeur à la Direction des Systèmes d'Information de la préfecture, notre mission est de créer la documentation de conception de l'application actuelle du service des cartes grises, afin de faciliter sa future évolution. Cette démarche inclut un audit de l'application existante, la création de documents techniques (diagrammes, architecture, etc.), et la proposition d'améliorations. Le projet aboutira à la rédaction d'un cahier des charges détaillant les besoins fonctionnels et techniques, permettant ainsi de guider efficacement l'évolution de l'application.

1.2. Date de rendu du projet

Le projet doit être rendu au plus tard le 14 mars 2025.

2.Besoins fonctionnels

Les salarié doit avoir accès a la gestion des demandes de cartes grises.

3.Ressources nécessaires à la réalisation du projet

3.1. Ressources matérielles

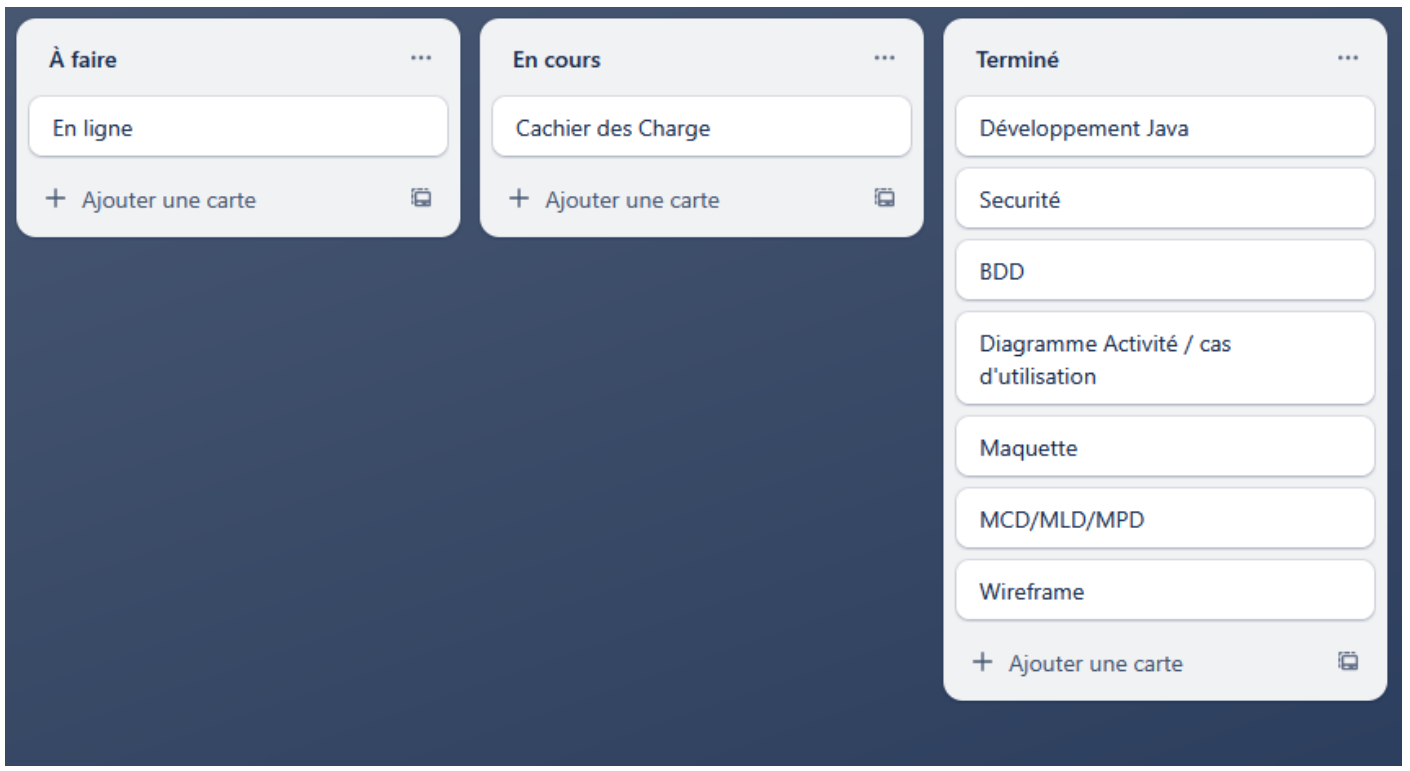
- **Ordinateurs**
- **Clavier**
- **Ecran**
- **Souris**
- **Connecter wifi/cablé**

3.2. Ressources logicielles

- **Outil de versioning** : Git et plateforme de dépôt comme GitHub.
- **Environnements de développement** : Visual Studio Code.
- **Outils de design** : Figma pour les wireframes et maquettes.
- **SGBD** : MySQL pour la base de données relationnelle.

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.

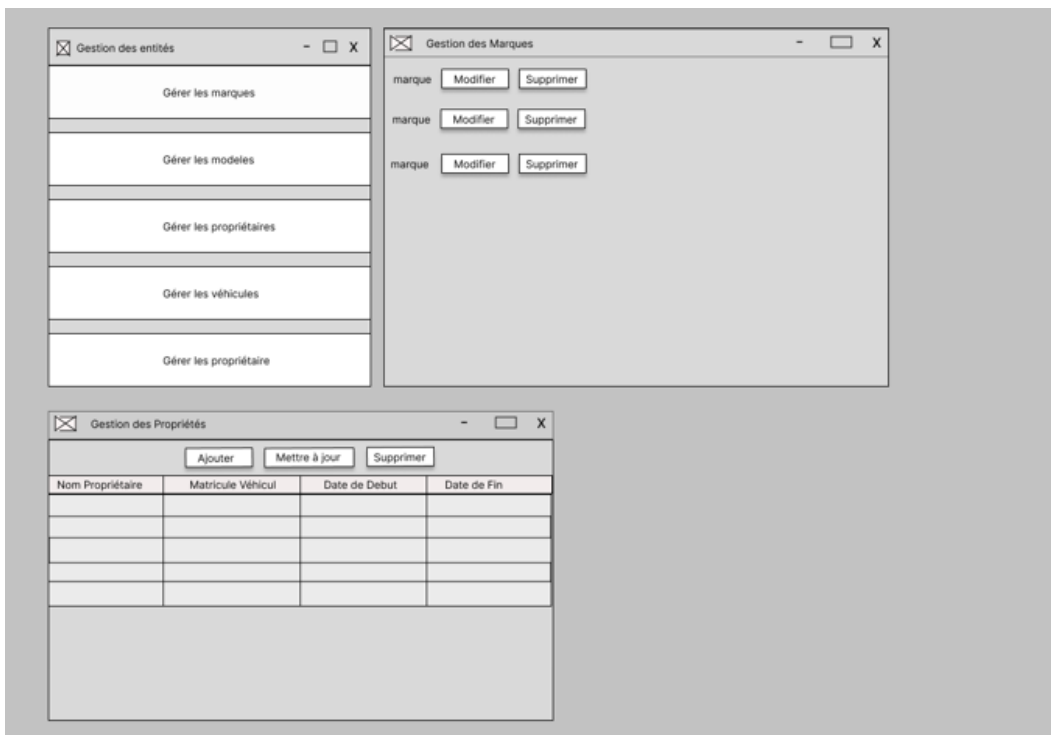


Nous travaillons également sur GitHub, plateforme de développement collaboratif.


4. Conception du projet

4.1. Le front-end

4.1.1. Wireframes



4.1.2. Maquettes

 Gestion des entités

Gérer les marques

Gérer les modèles

Gérer les propriétaires

Gérer les véhicules

Gérer les propriétés

 Gestion des Modèles

208 (Marque : Peugeot)

Modifier

Supprimer

Clio (Marque : Renault)

Modifier

Supprimer


Yaris (Marque : Toyota)

Modifier

Supprimer

Ajouter un Modèle

Retour

 Gestion des Propriétés

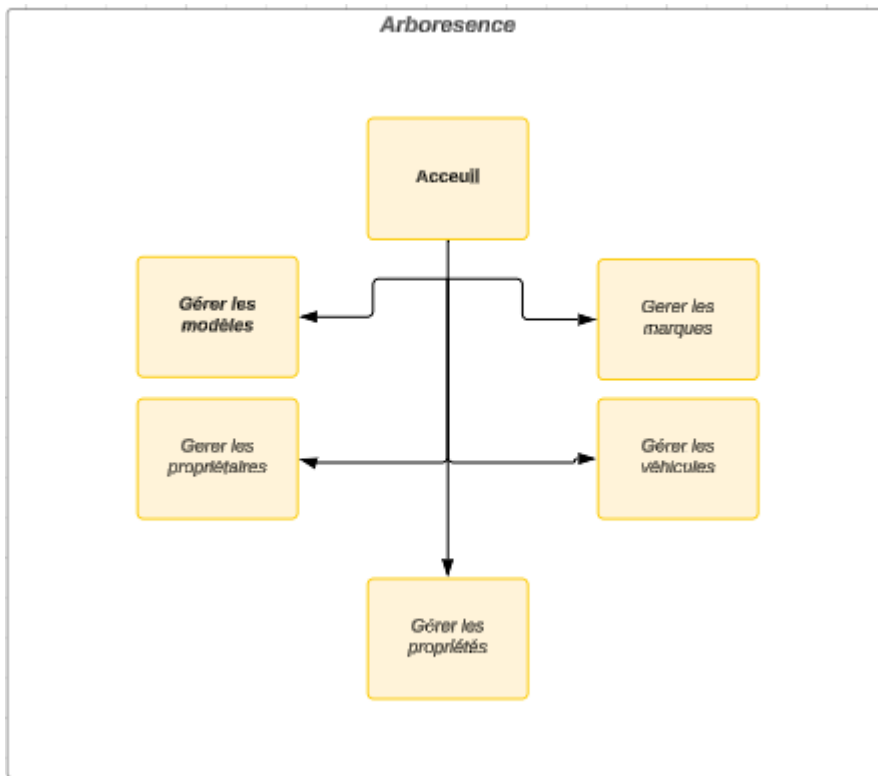
Ajouter

Mettre à jour

Supprimer

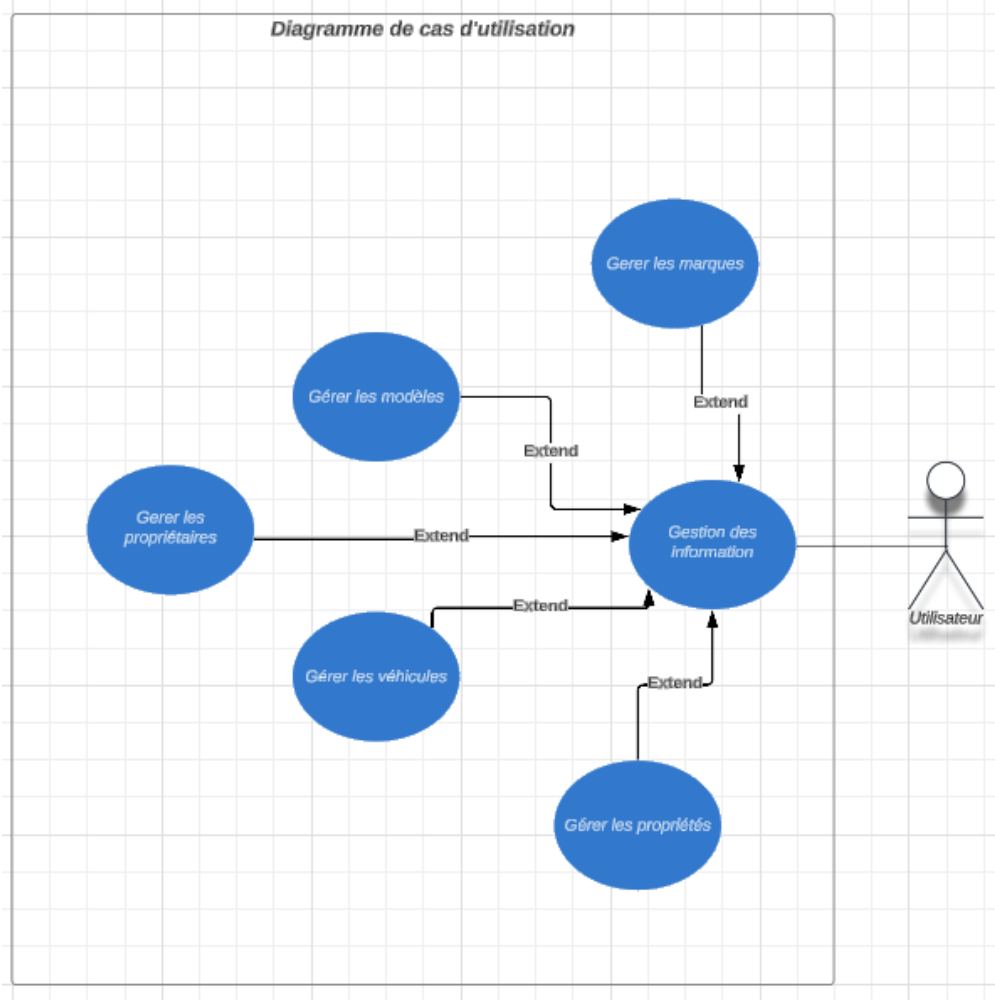
Nom Propriétaire	Matricule Véhicule	Date Début	Date Fin
John Doe	AB-123-CD	2023-01-01	2023-12-31
John Doe	00-aaa-00	2000-10-10	2150-10-10
Jane Doe	EF-456-GH	2023-01-01	2750-10-10
Alice Smith	IJ-789-KL	2023-06-01	

4.1.3. Arborescences

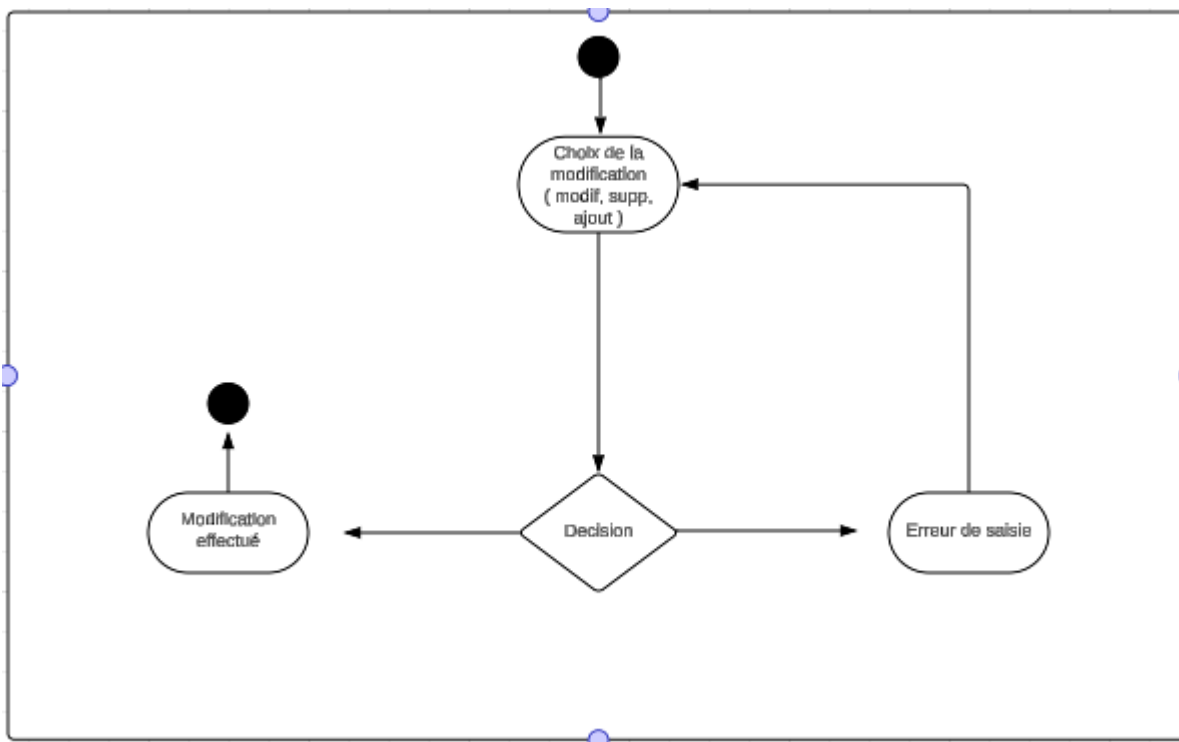


4.2. Le back-end

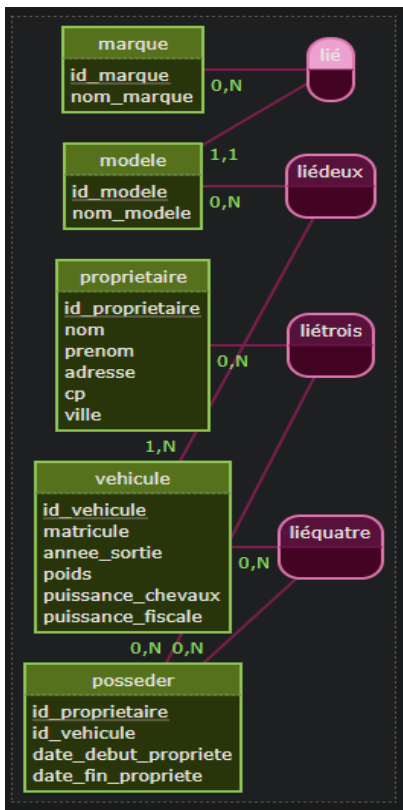
4.2.1. Diagramme de cas d'utilisation



4.2.2. Diagramme d'activités



4.2.3. Modèles Conceptuel de Données (MCD)



4.2.4. Modèle Logique de Données (MLD)

lièdeux (#id_modele, #id_vehicule)

liétrois (#id_proprietaire1, #id_proprietaire 2)

liéquatre (#id_vehicule, #id_proprietaire)

marque (id_marque, nom_marque)

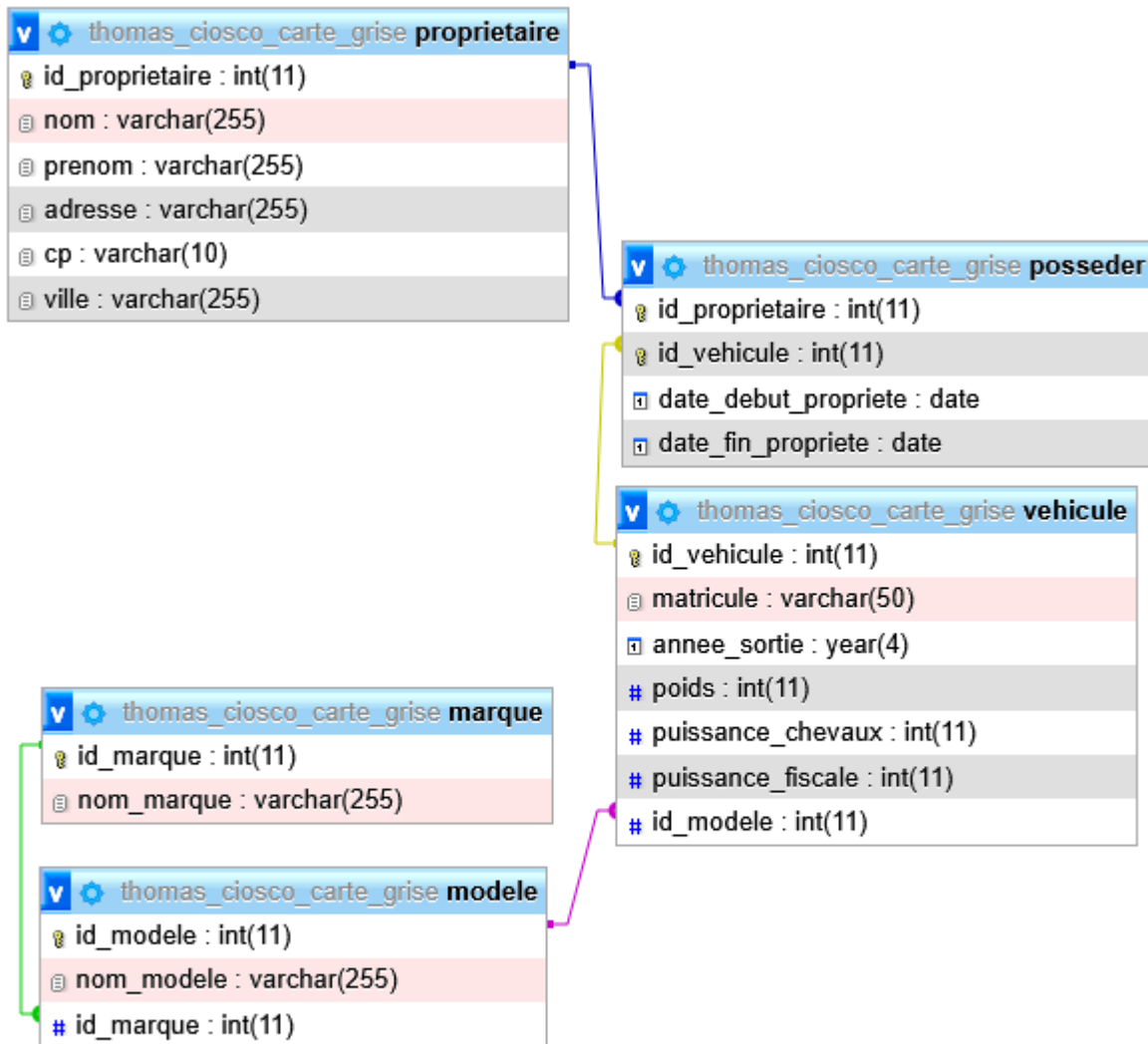
modele (id_modele, nom_modele, #id_marque)

posseder (id_proprietaire, id_vehicule, date_debut_propriete, date_fin_propriete)

proprietaire (id_proprietaire, nom, prenom, adresse, cp, ville)

vehicule (id_vehicule, matricule, annee_sortie, poids, puissance_chevaux,
puissance_fiscale)

4.2.5. Modèle Physique de Données (MPD)



5. Technologies utilisées

5.1. Langages de développement

- **Java** : Mise en page et design.

5.2. Base de données

- **MySQL** : Système de gestion de base de données relationnelles.

6. Sécurité

6.1. Cryptage des mots de passe avec Bcrypt

Pour la sécurité des MDP, nous utiliserons **Bcrypt**, un algorithme de hachage qui permet de rendre difficile leur récupération même en cas de fuite de données.

- `$password = 'mot_de_passe_utilisateur';`
`$hash = password_hash($password, PASSWORD_BCRYPT);`

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

Cela empêche l'exécution de scripts malveillants injectés dans le contenu.

```
$input = htmlspecialchars($_POST['input'], ENT_QUOTES, 'UTF-8');
```

6.3. Protection contre les injections SQL

Pour prévenir les injections SQL, toutes les requêtes à la base de données seront préparées et paramétrées, ce qui bloque toute tentative d'injection.

```
$stmt = $pdo->prepare('SELECT * FROM users WHERE email = :email');  
$stmt->execute(['email' => $email]);
```