

CAHIER DES CHARGES



Projet Carte Grises

Réaliser par JEAN Richardson Junior

Sommaire

I.	Contexte du projet.....	3
1.	Carte grises : présentation	3
2.	Objectifs.....	3
3.	Date de rendu du projet	3
II.	Besoins fonctionnels.....	3
III.	Ressources nécessaires à la réalisation du projet.....	4
4.	Besoins matériels	4
5.	Besoins logiciels.....	4
IV.	Gestion du projet	4
V.	Conceptions du projet	5
6.	Le front-end.....	5
	<ul style="list-style-type: none">• Wireframes• Maquettes• Arborescence	
7.	Le back-end.....	9
	<ul style="list-style-type: none">• Diagramme de cas d'utilisation• Modèles Conceptuel de Données (MCD)• Modèle logique de Données (MLD)• Modèle Physique de Données (MPD)• Diagramme de classe	
VI.	Technologies utilisées	13
8.	Langages de développement web.....	13
9.	Base de données.....	13
VII.	Sécurité	13
10.	Login et protection des pages administrateurs	13
11.	Cryptage des mots de passe avec Bcrypt	13
12.	Protection contre les attaques XSS (Cross-Site-Scripting)	13
13.	Protection contre les injections SQL.....	14

I. Contexte du projet

1. Carte grises : présentation

La carte grise, officiellement appelée certificat d'immatriculation, est un document obligatoire qui atteste de l'immatriculation d'un véhicule en France. Délivrée par l'État, elle contient des informations essentielles telles que l'identité ou propriétaire, les caractéristiques du véhicule (marque, modèle, puissance, etc...) et le numéro d'immatriculation. Indispensable pour circuler légalement, elle permet également de suivre l'historique du véhicule en cas de vente ou de contrôle routier.

2. Objectifs

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 métier. 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.

3. Date de rendu du projet

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

II. Besoins fonctionnels

L'application métier de gestion des cartes grises devra présenter :

- Des marques de voitures
- Des modèles de voitures
- Les propriétaires
- Les véhicules
- Les propriétés
- Devices compatible

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.

III. Ressources nécessaires à la réalisation du projet

1. Besoins matériels

- Ordinateur portable (Connecté à internet)
- Fixe (Connecté à internet)

2. Besoins logiciels

Environnement de développement (IDE) : Visual Studio Code

Plateforme de développement collaboratif : Github

Outils de gestion de projet : Trello

Conception UML et arborescence : Visual Paradigme online

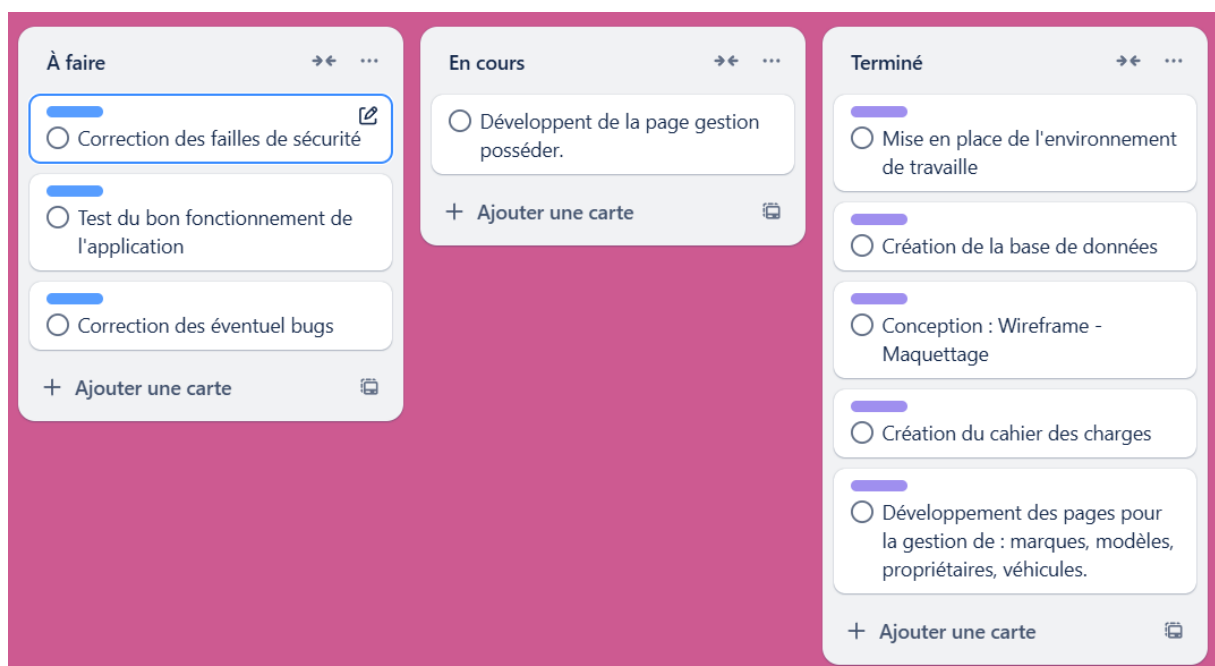
Maquettage : Figma

Conception de base de données : Mocodo

Sources documentaire possible : StackOverFlow

IV. Gestion du projet

Pour la gestion du projet, nous avons créé un tableau avec la plateforme Trello qui est visible ci-dessous :



Nous travaillons également sur Github, plateforme de développement collaboratif.

V. Conceptions du projet

I. Le front-end

- **Wireframes**

Les Wireframes réalisés pour ce projet avec Figma, sont les suivants :

A gray rectangular wireframe with a horizontal gradient, containing the text 'Gérer les marques' centered in black.

Gérer les marques

A gray rectangular wireframe with a horizontal gradient, containing the text 'Gérer les modèles' centered in black.

Gérer les modèles

A gray rectangular wireframe with a horizontal gradient, containing the text 'Gérer les propriétaires' centered in black.

Gérer les propriétaires

A gray rectangular wireframe with a horizontal gradient, containing the text 'Gérer les véhicules' centered in black.

Gérer les véhicules

A gray rectangular wireframe with a horizontal gradient, containing the text 'Gérer les propriétés' centered in black.

Gérer les propriétés

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

208 (Marque : Peugeot)

Modifier

Supprimer

Clio (Marque : Renault)

Modifier

Supprimer

Yaris (Marque : Toyota)

Modifier

Supprimer

Ajouter un Modèle

Retour

AB-123-CD (Modèle : 208)

Modifier

Supprimer

EF-456-GH (Modèle : Clio)

Modifier

Supprimer

IJ-789-KL (Modèle : Yaris)

Modifier

Supprimer

Ajouter un Véhicule

Retour

Peugeot

Modifier

Supprimer

Peugeot

Modifier

Supprimer

Peugeot

Modifier

Supprimer

Ajouter une marque

Retour

- **Maquettes**

Gérer les marques

Gérer les modèles

Gérer les propriétaires

Gérer les véhicules

Gérer les propriétés

AB-123-CD (Modèle : 208)

ModifierSupprimer

EF-456-GH (Modèle : Clio)

ModifierSupprimer

IJ-789-KL (Modèle : Yaris)

ModifierSupprimer

Ajouter un Véhicule

Retour

Doe John (123 Rue de paris, 75000 Paris)

ModifierSupprimer

Doe Jane (456 Avenue des champs, 75008 Paris)

ModifierSupprimer

Smith Alice (789 Boulevard Haussmann, 75009 Paris)

ModifierSupprimer

Ajouter un Propriétaire

Retour

Cahier des charges – Carte grises

8

Peugeot [Modifier](#) [Supprimer](#)

Peugeot [Modifier](#) [Supprimer](#)

Peugeot [Modifier](#) [Supprimer](#)

[Ajouter une marque](#)

[Retour](#)

208 (Marque : Peugeot) [Modifier](#) [Supprimer](#)

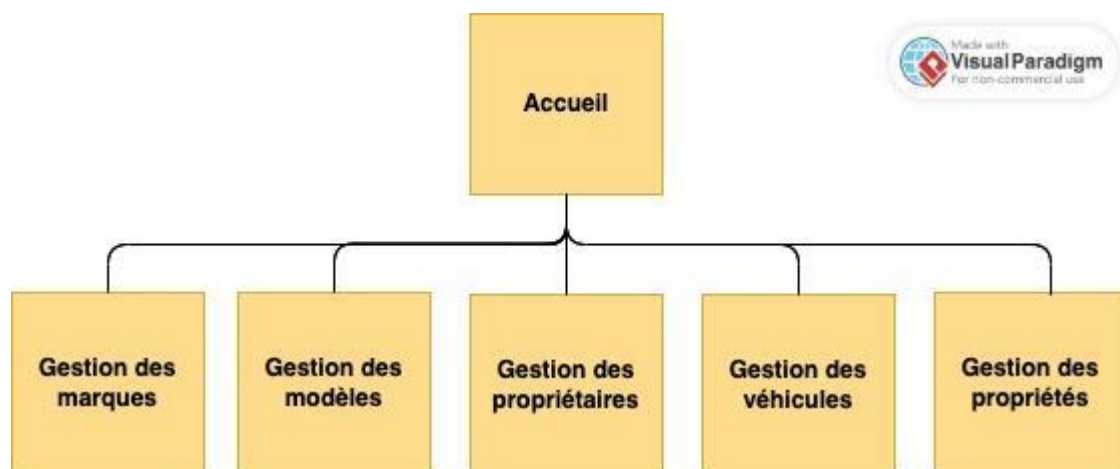
Clio (Marque : Renault) [Modifier](#) [Supprimer](#)

Yaris (Marque : Toyota) [Modifier](#) [Supprimer](#)

[Ajouter un Modèle](#)

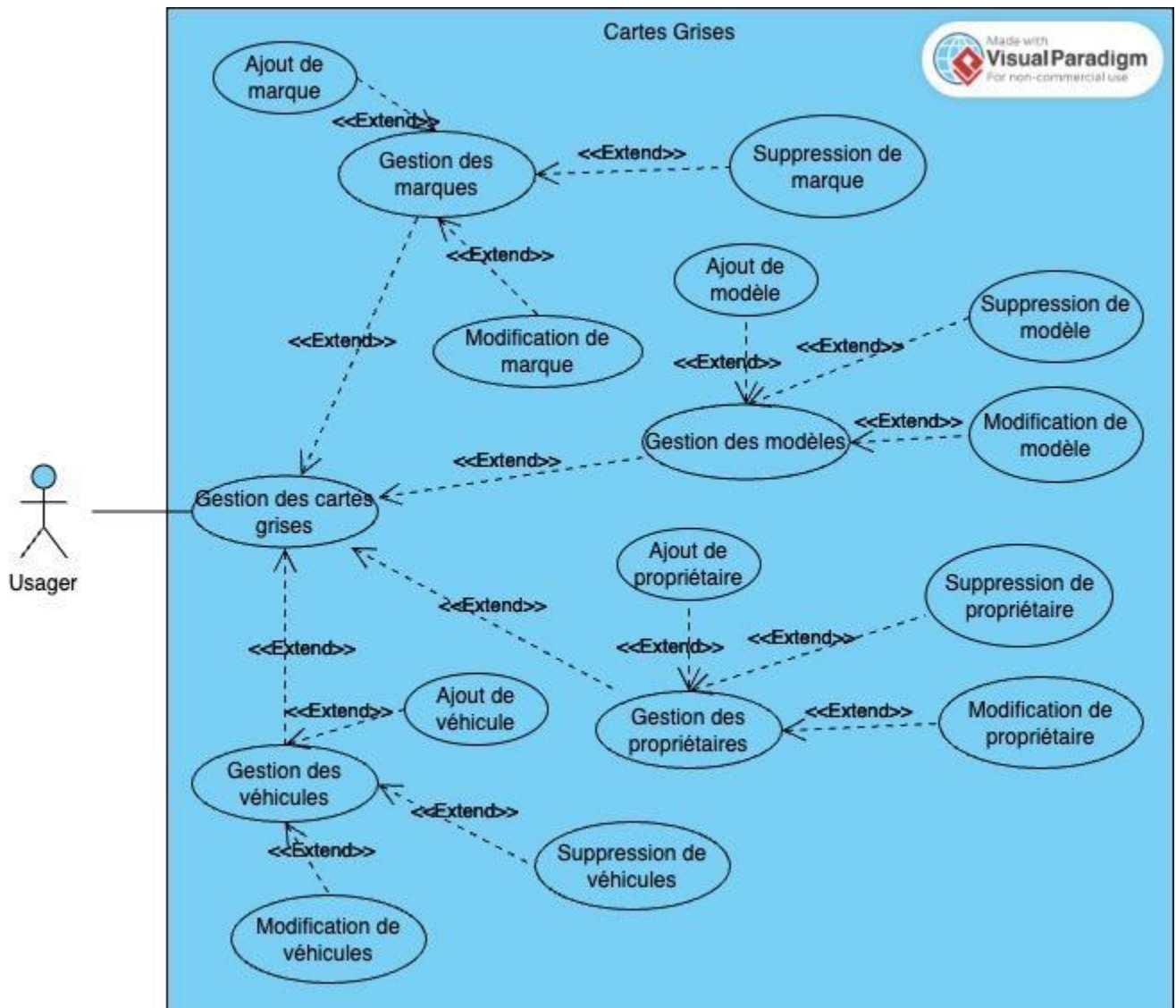
[Retour](#)

- **Arborescences**

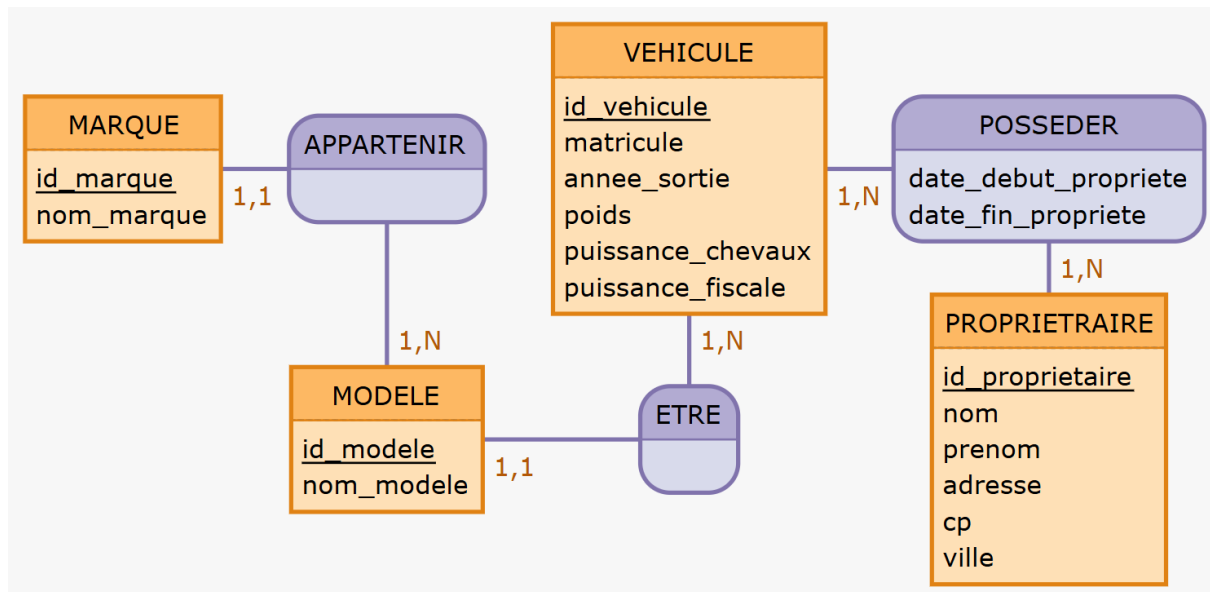


2. Le back-end

- Diagramme de cas d'utilisation



- **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 références à MARQUE (id_marque)

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

Clé primaire = id_proprietaire

VEHICULE (id_vehicule, matricule, annee_sortie, poids, puissance_chevaux, puissance_fiscale, #id_modele)

Clé primaire = id_vehicule

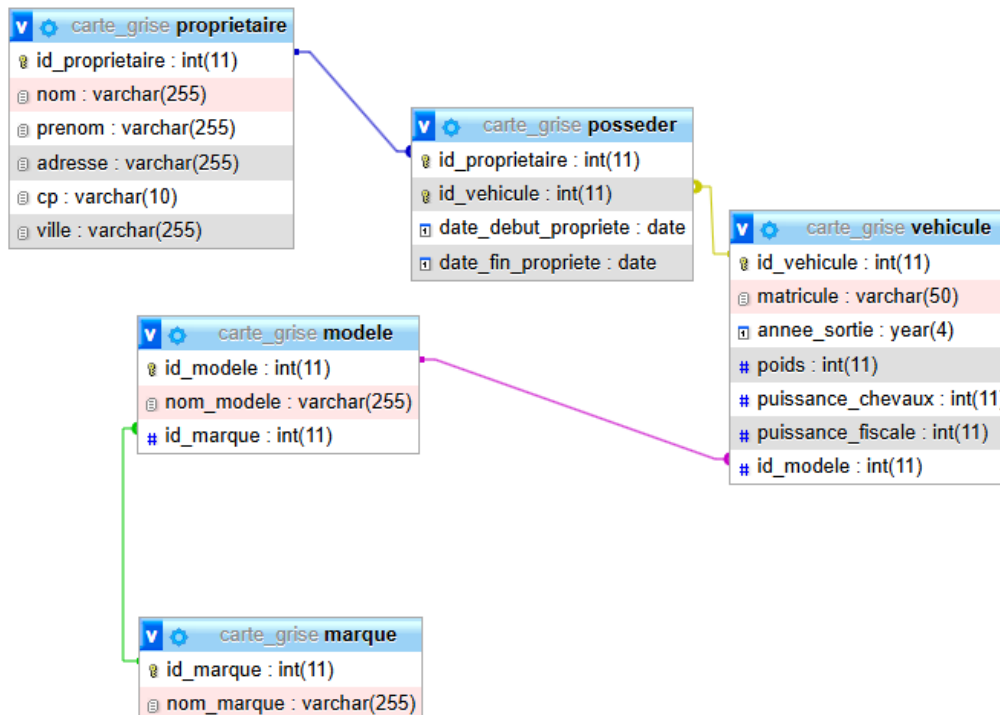
Clé étrangère = id_modele en références à MODELE (id_modele)

POSSEDER (#id_proprietaire, #id_vehicule, date_debut_propriete, date_fin_propriete)

Clé primaire = id_proprietaire, id_vehicule

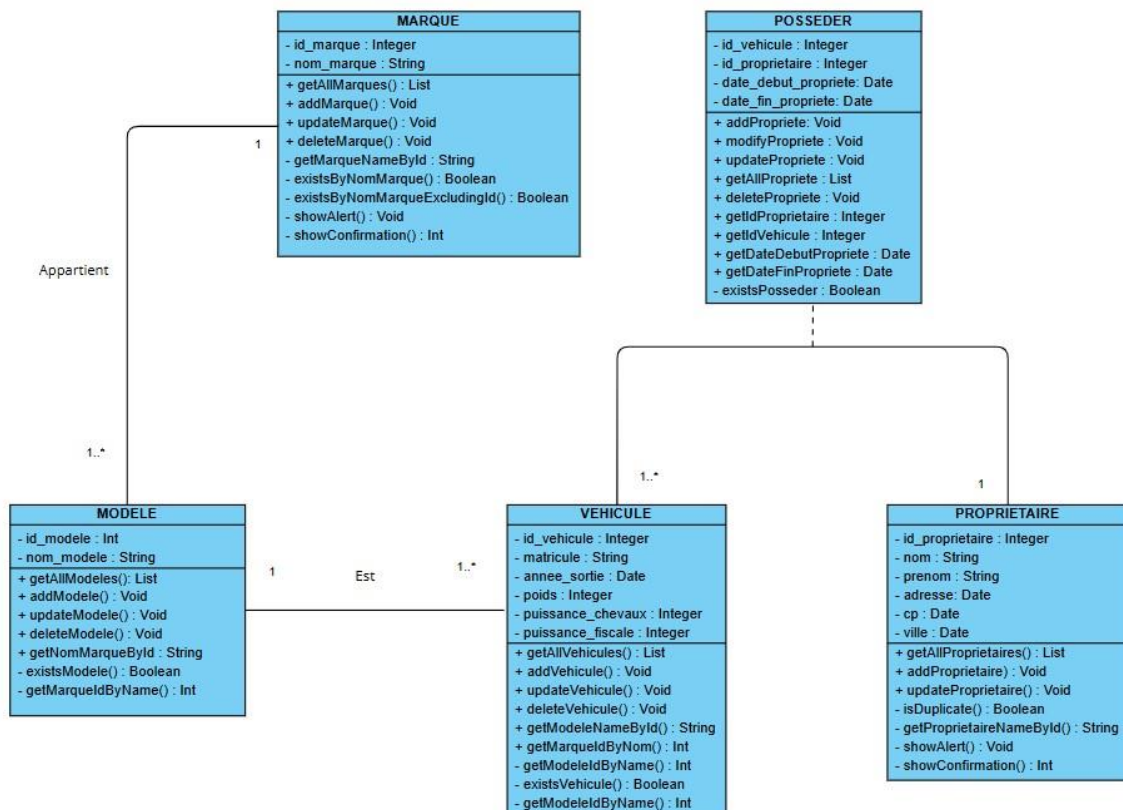
Clé étrangère = id_proprietaire en références à PROPRIETAIRE (id_proprietaire)
id_vehicule en références à VEHICULE (id_vehicule)

- **Modèle Physique de Données (MPD)**



• Diagramme de classe

2.



VI. Technologies utilisées

3. Langages de développement web

Front-end web :

- JAVA

Back-end web :

- JAVA

4. Base de données

- Serveur web :Apache (contenue dans MAMP)
- Système de base de données relationnel : MySQL (contenue dans MAMP)

VII. Sécurité

5. Login et protection des pages administrateurs

Définition :Authentification pour limiter l'accès aux zones sensibles (pages administrateurs)

Solutions : Utiliser HTTPS pour sécuriser la connexion, implémenter l'authentification à deux facteurs (2FA) et bloquer les comptes après trop de tentatives de connexion échouées. Ou la méthode POST pour transmettre secrètement les données.

6. Cryptage des mots de passe avec Bcrypt

Définition : Bcrypt hache les mots de passe, les rendant difficiles à déchiffrer grâce à un salage et un facteur de coût ajustable.

Solutions :

- Hacher les mots de passe avec Bcrypt avant stockage.
- Utiliser un facteur de coût élevé.
- Ne jamais stocker de mots de passe en clair.

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

Le Cross-site scripting (XSS) est une vulnérabilité permettant à un attaquant d'injecter du code malveillant dans un site web, contournant ainsi les contrôles d'accès et usurpant l'identité des utilisateurs ; les méthodes de prévention incluent le filtrage, l'échappement des caractères spéciaux, et l'utilisation de Content Security Policy (CSP) pour restreindre les sources de contenu exécutables.

8. Protection contre les injections SQL

Définition : Une injection SQL permet à un attaquant d'exécuter des commandes SQL malveillantes via des formules ou des URL.

Solutions :

- Utiliser des requêtes préparées.

```
Connection conn =  
DriverManager.getConnection("jdbc:mysql://localhost:3306/maB  
ase", "user", "password");  
String sql = "SELECT * FROM utilisateurs WHERE username = ?  
AND password = ?";  
PreparedStatement  
pstmt = conn.prepareStatement(sql);  
pstmt.setString(1, username);  
pstmt.setString(2, password);  
ResultSet rs = pstmt.executeQuery();
```
- Échapper les entrées utilisateurs.
- Limiter les droits d'accès à la base de données.