

# CAHIER DES CHARGES TECHNIQUES : Application JAVA cartes grises



# Sommaire

## 1. Contexte du projet

### 1.1 Présentation

### 1.2 Période et Modalité

## 2. Objectifs du projet

## 3. Besoins fonctionnels

### 3.1 Fonctionnalités principales

### 3.2 Sécurité

## 4. Architecture technique

### 4.1 Ressources matérielles et logicielles 4.2 Langages utilisés

## 5. Conception et Modélisation

### 5.1 Modèle de données

### 5.2 Interfaces utilisateur

### 5.3 Diagrammes UML

## 6. Développement et Tests

### 6.1 Gestion du projet

### 6.2 Tests et validation

## 7. Déploiement et Maintenance

## 8. Livrables

## 9. Conclusion

# **1. Contexte du projet**

## **1.1 Présentation**

Ce projet a été réalisé dans le cadre du **BTS SIO – SLAM** lors des séances d'AP au **Lycée La Tournelle de La Garenne-Colombes**. L'objectif est de développer une **application Java** .

## **1.2 Période et Modalité**

- **Durée** : Du **06 novembre 2023** au **08 mars 2024**
- **Mode de réalisation** : **Travail individuel** avec un suivi pédagogique.

# **2. Objectifs du projet**

L'application doit permettre de :

- **Afficher la liste des sports et disciplines olympiques**
- **Gérer le calendrier des épreuves** avec les dates et lieux des compétitions
- **Afficher les résultats des épreuves en temps réel**
- **Créer un espace administrateur** pour gérer les informations des compétitions
- **Stocker et organiser les données via une base de données SQL**

# **3. Besoins fonctionnels**

## **3.1 Fonctionnalités principales**

- **Consultation des disciplines sportives**
- **Affichage dynamique du calendrier des épreuves**
- **Gestion et mise à jour des résultats des compétitions**
- **Espace administrateur sécurisé** (ajout, modification et suppression des épreuves)
- **Gestion des utilisateurs et droits d'accès**

## **3.2 Sécurité**

- **Authentification sécurisée** pour l'espace administrateur

- **Protection contre les injections SQL** via des requêtes préparées
- **Validation des entrées utilisateurs** pour éviter les failles XSS

## 4. Architecture technique

### 4.1 Ressources matérielles et logicielles

- **Matériel** : Ordinateur portable, connexion internet
- **IDE** : Visual Studio Code
- **SGBD** : MySQL via MAMP
- **Outils de conception** : Mocodo, Visual Paradigm

### 4.2 Langages utilisés

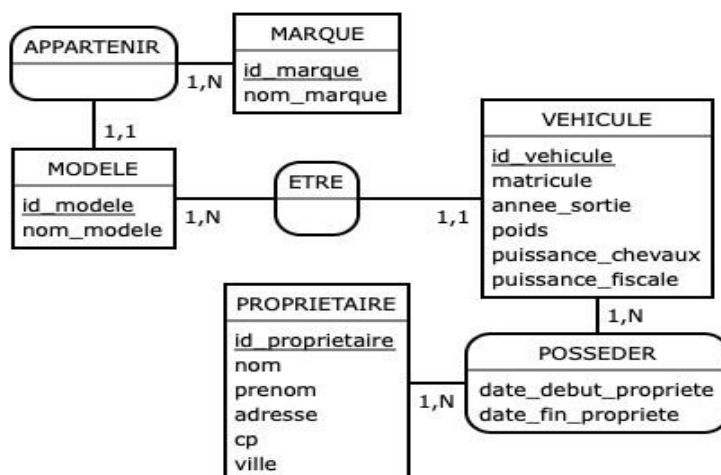
- **Back-end** : Java (JavaFX/Spring Boot)
- **Front-end** : HTML5, CSS3, JavaScript
- **Base de données** : SQL

## 5. Conception et Modélisation

### 5.1 Modèle de données

- **MCD , MLD & MPD** pour structurer les relations entre les tables

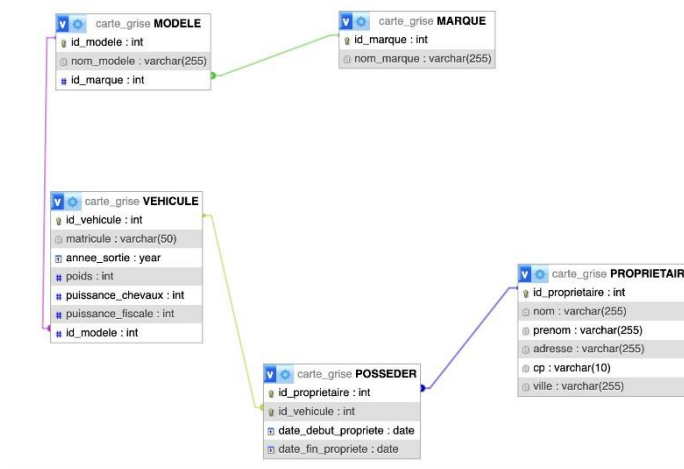
#### Modèle Conceptuel de données ( MCD )



## Modèle Logiques de données ( MLD ) format 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 → MARQUE (id\_marque)
- VEHICULE (id\_vehicule, matricule, annee\_sortie, poids, puissance\_chevaux, puissance\_fiscale, id\_modele)
  - Clé primaire : id\_vehicule
  - Clé étrangère : id\_modele → MODELE (id\_modele)
- PROPRIETAIRE (id\_proprietaire, nom, prenom, adresse, cp, ville)
  - Clé primaire : id\_proprietaire
- POSSEDER (id\_proprietaire, id\_vehicule, date\_debut\_propriete, date\_fin\_propriete)
  - Clé primaire : id\_proprietaire, id\_vehicule, date\_debut\_propriete
  - Clé étrangère : id\_proprietaire → PROPRIETAIRE (id\_proprietaire)
  - Clé étrangère : id\_vehicule → VEHICULE (id\_vehicule)

## Modèle Physique de données ( MPD )



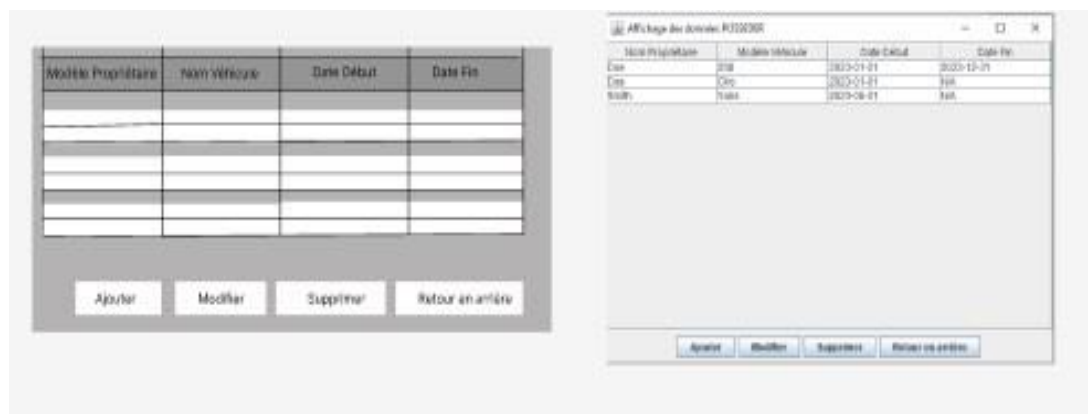
## 5.2 Interfaces utilisateur

- Wireframes et maquettes créés avec le logiciel Figma.

### Wireframes et maquettes pour Gestion des Véhicules



### Wireframes et maquettes pour Affichage des données POSSEDER

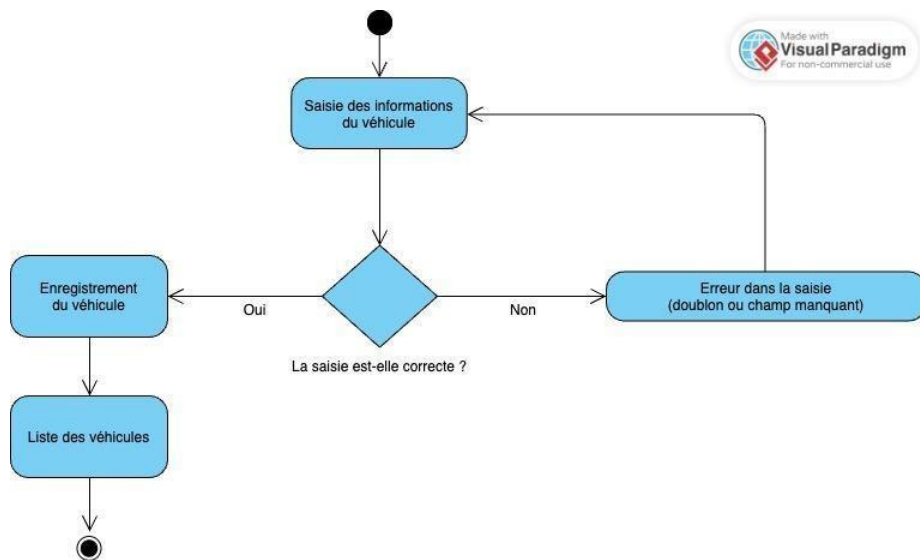


### Wireframes et maquettes pour Gestion des entrées (menu)

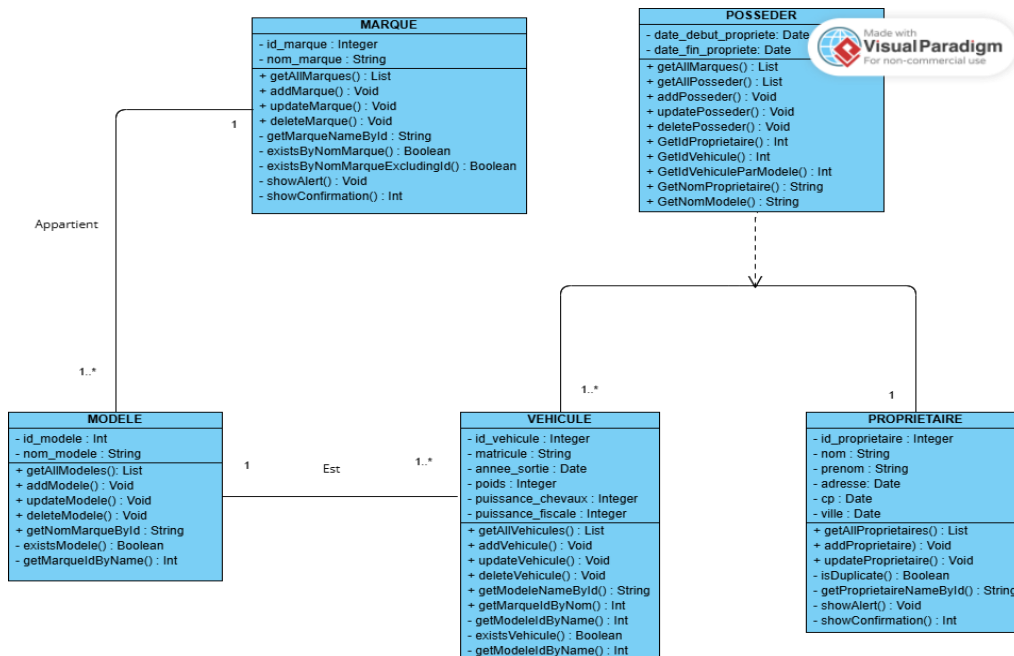


## 5.3 Diagrammes UML

- **Diagramme activité** pour définir les interactions



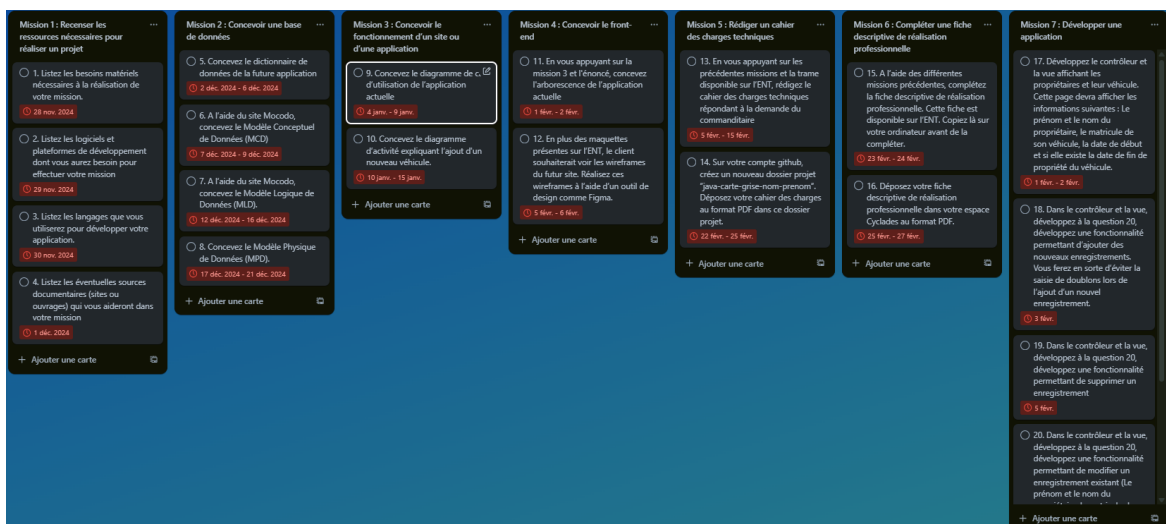
- **Diagramme de classe** pour modéliser le fonctionnement



## 6. Développement et Tests

### 6.1 Gestion du projet

- Méthodologie Classique en cycle V
- Utilisation de GitHub pour la gestion du code source
- Utilisation de Trello pour l'organisation du projet



### 6.2 Tests et validation

- Tests unitaires en JUnit pour valider chaque fonctionnalité
- Tests d'intégration pour assurer le bon fonctionnement global



## **7. Déploiement et Maintenance**

- Hébergement sur un serveur local via **MAMP**
- **Documentation technique** et **guide utilisateur** fournis
- **Maintenance évolutive** pour mises à jour futures

## **8. Livrables**

- Code source du projet sur **GitHub**
- Base de données SQL exportable
- Documentation technique complète
- Rapport de test détaillé

## **9. Conclusion**

Ce projet vise à offrir une **application Java performante et intuitive** pour gérer les informations des **cartes grises de plusieurs modèles de véhicules**. Il met en application les compétences acquises en **conception, développement et gestion des données** dans un cadre professionnel.