



APPLICATION JO2024 : CAHIER DES CHARGES TECHNIQUES

En JAVA



BTS SIO SLAM

FOUSSEYNOU SIDIBE

Sommaire

1. Contexte	2
1.1 Présentation	2
1.2 Période et Modalité	2
2. Objectifs du projet	2
3. Besoins fonctionnels	2
3.1 Fonctionnalités principales	2
3.2 Sécurité	3
4. Architecture technique	3
4.1 Ressources matérielles et logicielles	3
4.2 Langages utilisés	3
5. Conception et Modélisation	3
5.1 Modèle de données	3
5.2 Interfaces utilisateur	6
5.3 Diagrammes UML	9
6. Développement et Tests	10
6.1 Gestion du projet	10
7. Déploiement	11
8. Livrables	11
9. Conclusion	11

1. Contexte

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** permettant aux utilisateurs de consulter les **sports, le calendrier des épreuves et les résultats des Jeux Olympiques 2024**.

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

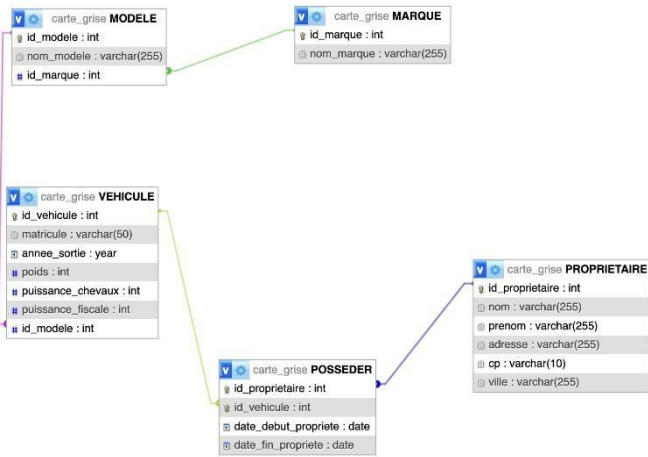
- **Back-end** : Java
- **Front-end** : HTML5, CSS3, JavaScript
- **Base de données** : SQL

5. Conception et Modélisation

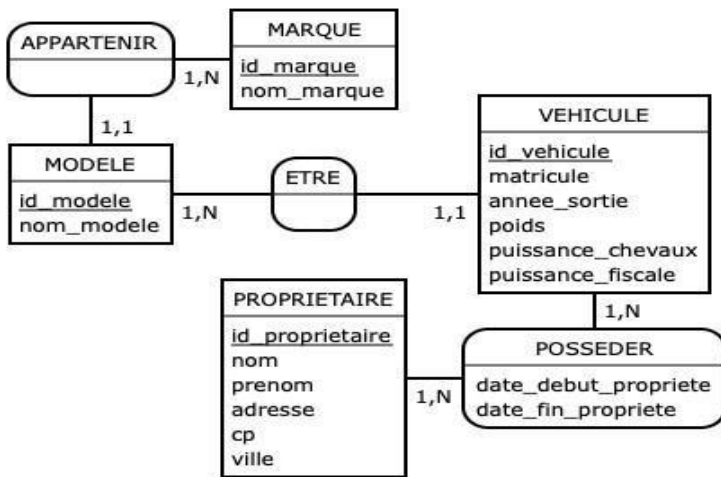
5.1 Modèle de données

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

Modèle Physique de données (MPD)



Modèle Conceptuel de données (MCD)



Modèle Logiques de données (MLD) format BTS

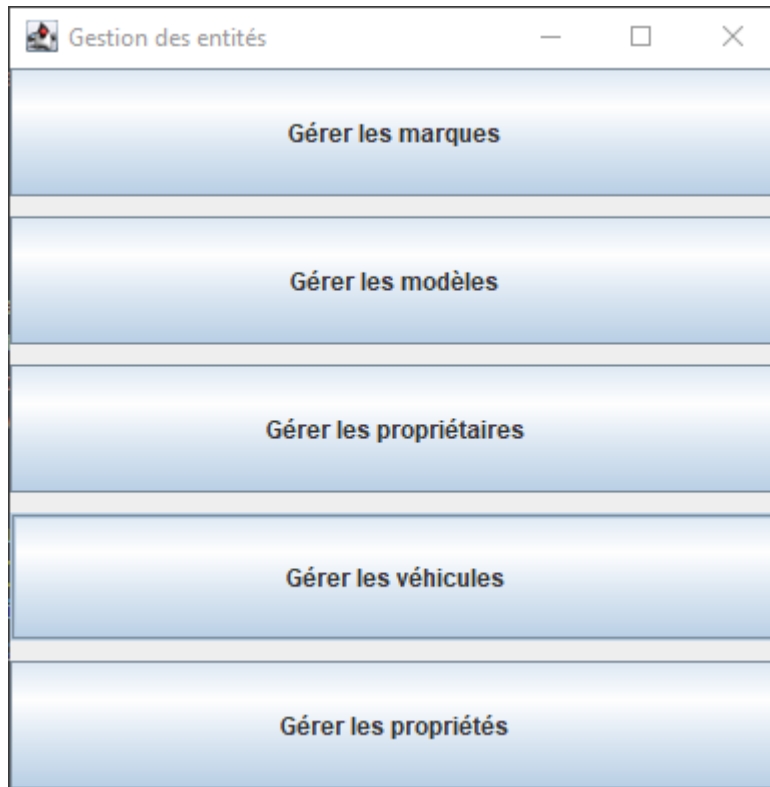
- MARQUE (id_marque, nom_marque)
Clé primaire : id_marque
- MODELE (id_modele, nom_modele, id_marque)
Clé primaire : id_modele
Clé étrangère : id_marque en référence à MARQUE (id_marque)
- VEHICULE (id_vehicule, matricule, annee_sortie, poids, puissance_chevaux, puissance_fiscale, id_modele)
Clé primaire : id_vehicule
Clé étrangère : id_modele en référence à 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 en référence à PROPRIETAIRE (id_proprietaire)
Clé étrangère : id_vehicule en référence à VEHICULE (id_vehicule)

5.2 Interfaces utilisateur

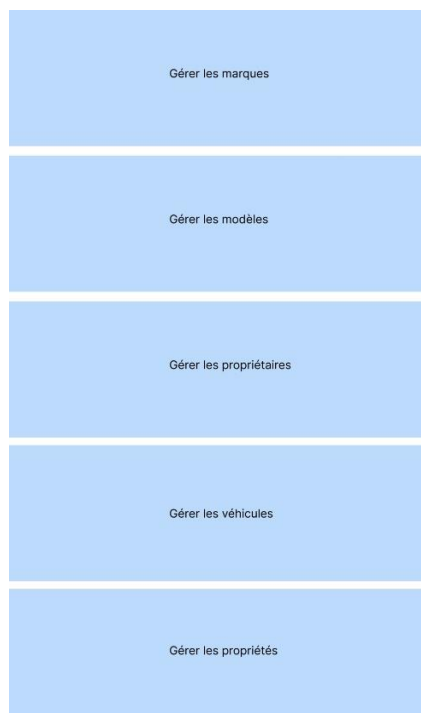
• **Wireframes** créés avec le logiciel **Figma**.

MENU :

Maquette :



Wireframe :



POSSEDER View :

Maquette :

Gestion des Propriétés

Propriétaire: Doe John, Véhicule: AB-123-CD, Début: 2023-01-01, Fin: 2023-12-31 (Épuisé)

ModifierSupprimer

Propriétaire: Doe Jane, Véhicule: EF-456-GH, Début: 2023-01-01, Fin: En cours (Actuel)

ModifierSupprimer

Propriétaire: Smith Alice, Véhicule: IJ-789-KL, Début: 2023-06-01, Fin: En cours (Actuel)

ModifierSupprimer

AjouterRetour

Wireframe :

Gestion des Propriétés

Propriétaire: Lorem ipsum Véhicule : XX-XXX-XX Debut:-----, Fin:----- (-----)

ModifierSupprimer

Propriétaire: Lorem ipsum Véhicule : XX-XXX-XX Debut:-----, Fin:----- (-----)

ModifierSupprimer

Propriétaire: Lorem ipsum Véhicule : XX-XXX-XX Debut:-----, Fin:----- (-----)

ModifierSupprimer

AjouterSupprimer

Ajout POSSEDER :

Maquette :

Ajouter une relation

?

Propriétaire:
1 - Doe John

Modèle de véhicule:
1 - 208

Date de début (YYYY-MM-DD):

Date de fin (YYYY-MM-DD, facultatif):

OKAnnuler

Wireframe :

Ajouter une relation

X

X

Propriétaire:

Modèle de véhicule:

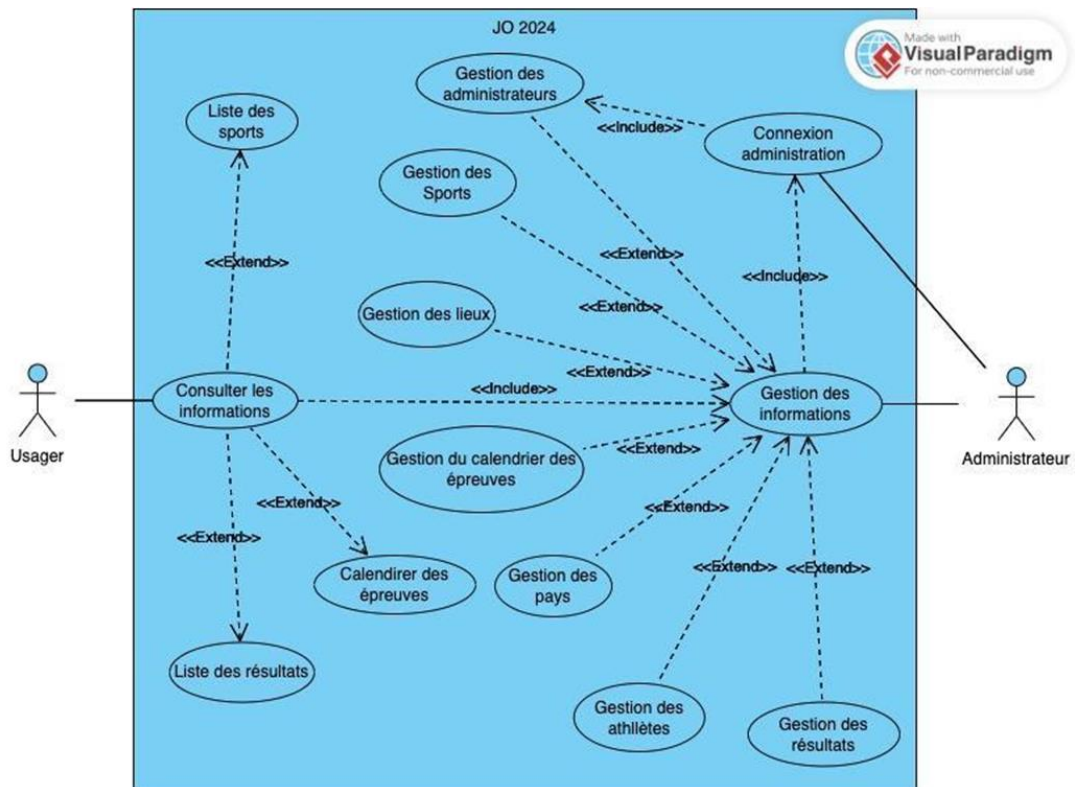
Date de début (YYYY-MM-DD):

Date de fin (YYYY-MM-DD, facultatif) :

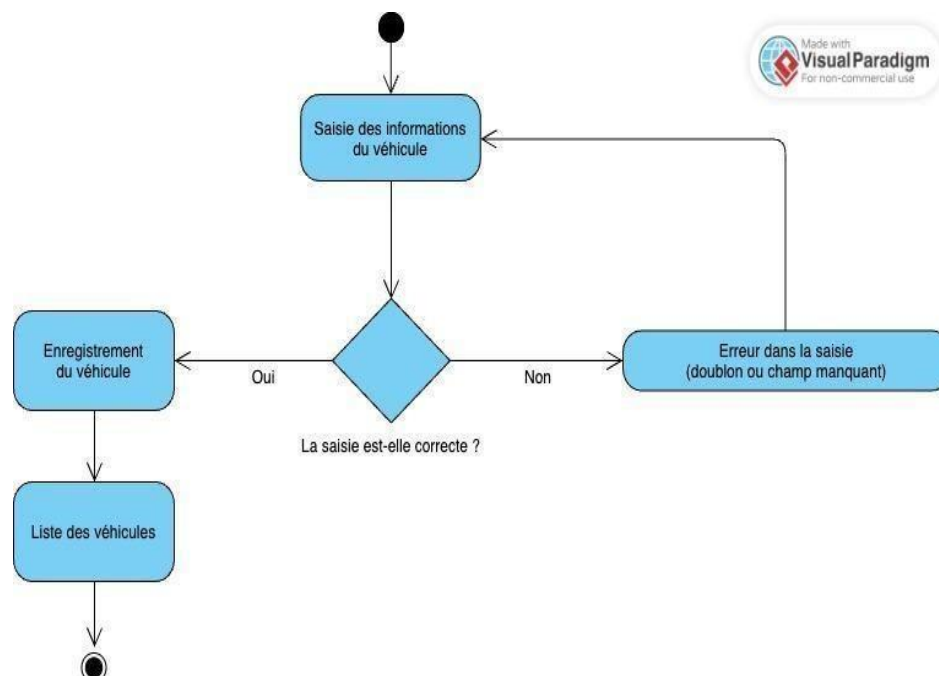
OKAnnuler

5.3 Diagrammes UML

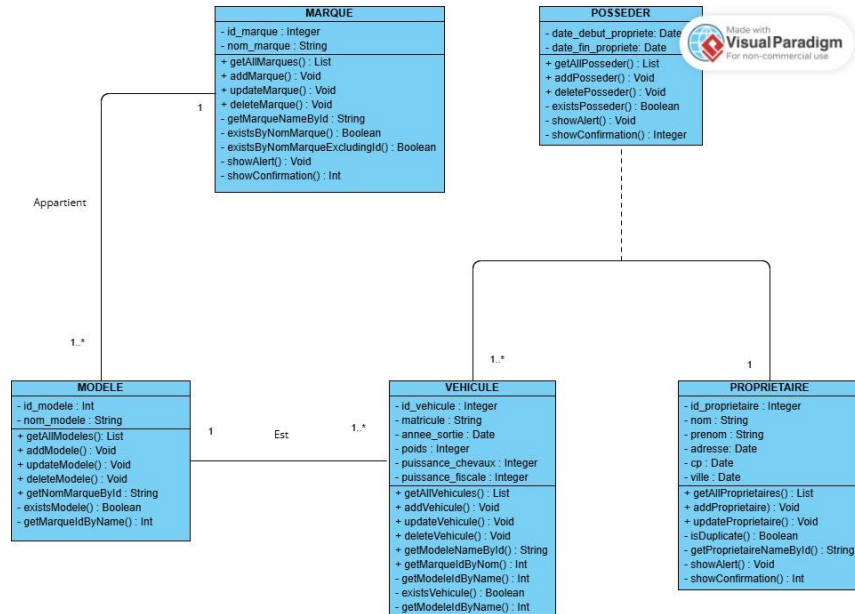
- Diagramme de cas d'utilisation :



- Diagramme d'activités :



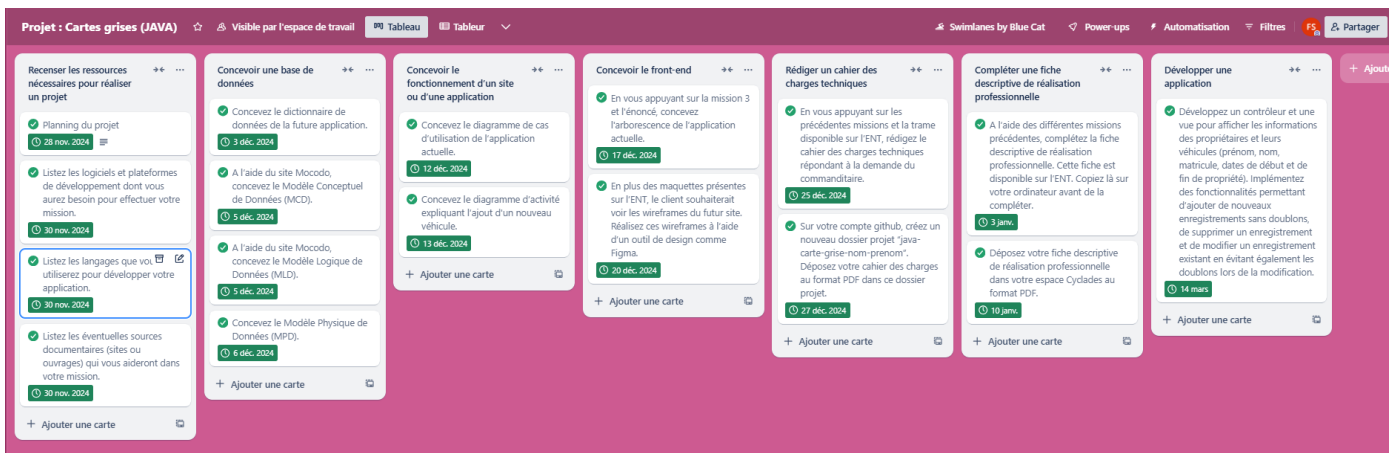
- Diagramme de classes :



6. Développement et Tests

6.1 Gestion du projet

- Méthode classique : Modèle V
- Utilisation de GitHub
- Trello :



7. Déploiement

- Hébergement sur un serveur local via MAMP

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.