
1 INTRODUCTION

1.1 *Objet du document*

Ce document a pour objectif de présenter l'essentiel des questions techniques liées à la mise en place de l'application GP3 Covoiturage Series.

Ce document présente :

- L'architecture technique retenue
- Le modèle physique de données

2 SOMMAIRE

2.1.1 Table des matières

1	INTRODUCTION	1
1.1	Objet du document	1
2	SOMMAIRE	2
2.1.1	Table des matières	2
3	ARCHITECTURE TECHNIQUE	3
3.1	Architecture technique	3
3.1.1	Schéma de l'architecture technique	3
3.1.2	Détails de l'architecture technique	4
3.1.3	Contraintes techniques	4
4	ARCHITECTURE LOGICIELLE	5
4.1	Produits et versions	5
4.1.1	Langages, frameworks et librairies spécifiques	5
4.1.2	Serveur de base de données	5
4.2	Mise en oeuvre de la base de données	5
4.2.1	Utilisation des procédures stockées	5
4.2.2	Mise en place d'index	6
5	FOCUS TECHNIQUES	7
5.1	Diagramme de classes métier	7
5.2	Modèle logique de données	Erreur ! Signet non défini.
5.3	Modèle physique de données	8
5.4	Diagramme de classes avec les classes techniques	Erreur ! Signet non défini.
5.5	Règles de développement	8
6	ACQUISITION DE DONNEES	9
6.1	Diagramme de séquences pour l'acquisition n°1	9
6.2	Diagramme de séquences pour l'acquisition n°2	Erreur ! Signet non défini.
7	TESTS ET INTEGRATION	10
7.1	Stratégie de tests	10
7.2	Indicateurs de qualité de code	10

3 ARCHITECTURE TECHNIQUE

3.1 Architecture technique

3.1.1 Schéma de l'architecture technique

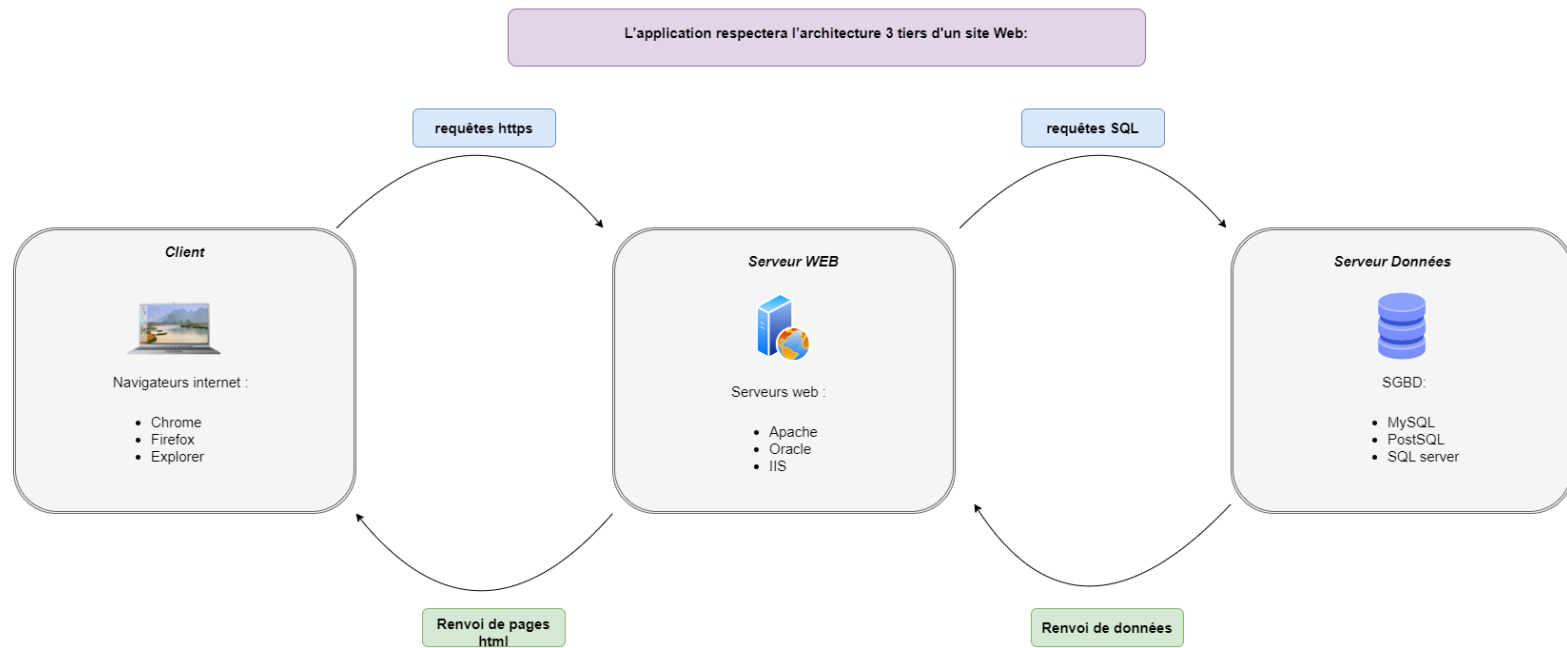


Figure 1 - schéma de l'architecture technique de l'application

3.1.2 Détails de l'architecture technique

Hébergement de l'application (exemple : sur clever-cloud).

3.1.3 Contraintes techniques

L'application doit être compatible pour l'ensemble des navigateurs récents.

4 ARCHITECTURE LOGICIELLE

4.1 Produits et versions

4.1.1 Langages, frameworks et librairies spécifiques

<u>Nom</u>	<u>Version</u>
Java	17
JUnit	5
Maven	3.8.7
Spring Boot	3.0
Spring Security	6.0.1
Hibernate / JPA	6.0
Angular	15
Angular Calendar	
Jasmine	4
Bootstrap	5
npm	9.20

4.1.2 Serveur de base de données

<u>Nom</u>	<u>Version</u>
MySQL	8

4.2 Mise en oeuvre de la base de données

4.2.1 Utilisation des procédures stockées

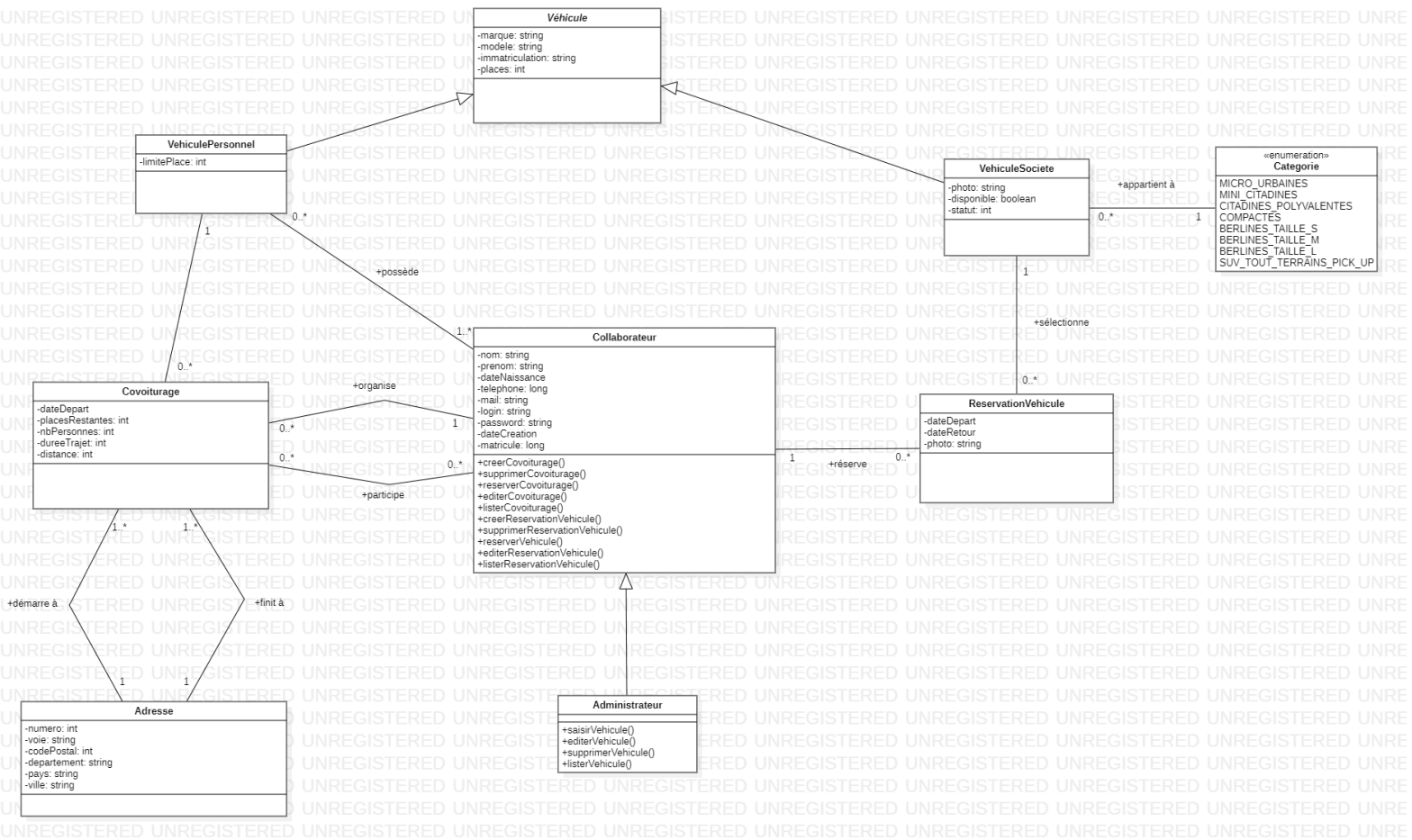
Sans objet

4.2.2 Mise en place d'index

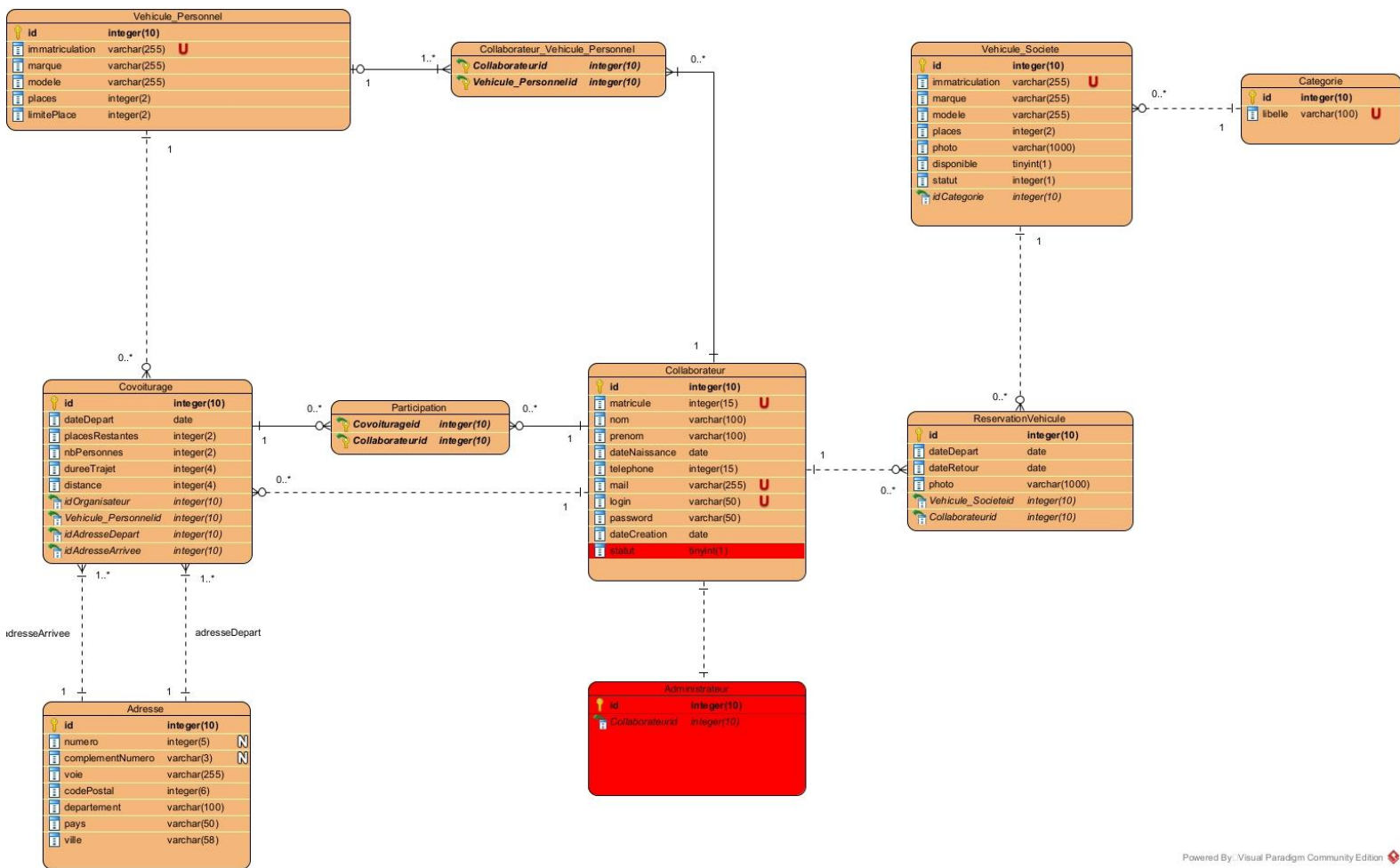
A définir.

5 FOCUS TECHNIQUES

5.1 Diagramme de classes métier



5.2 Modèle physique de données



Powered By : Visual Paradigm Community Edition

5.3 Règles de développement

L'application sera développée en respectant le paradigme de programmation orientée objet.

Quand on aura vu les GRASPS et les design patterns : à renseigner plus tard.

Découpage en couches : décrire les différentes couches avec contrôleurs, DAO, classes techniques, DTO, etc.

Découpage en packages : organisation du code

Règles de nommage :

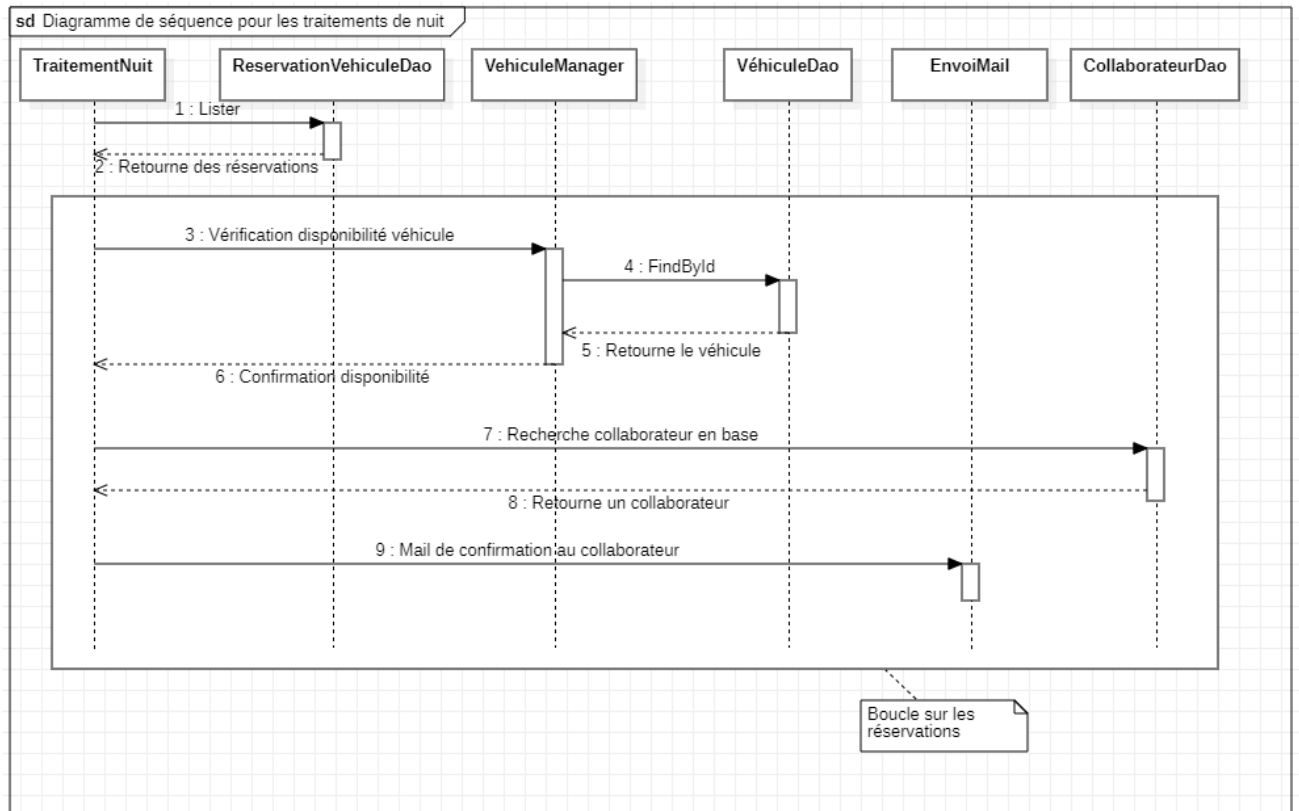
- Les DAOS ?
- Les DTOs ?
- Les classes de services ?
- Les classes utilitaires ?

Bonnes pratiques à respecter ?

- Respect des GRASP

6 ACQUISITION DE DONNEES

6.1 Diagramme de séquences pour le traitement de nuit



7 TESTS ET INTEGRATION

7.1 *Stratégie de tests*

Chaque fonctionnalité fera l'objet d'un test unitaire.

7.2 *Indicateurs de qualité de code*

7.2.1.1 *Qualité de code*

La documentation JAVA (Javadoc) doit être renseignée de manière exhaustive pour les parties de code concernées.

La qualité de code de l'application sera assurée grâce à l'outil de qualimétrie SonarQube. Les rapports SonarQube permettront d'identifier les duplications de code, de mesurer le niveau de documentation et d'estimer la couverture de test.

7.2.1.2 *Conventions de nommage*

- ❖ Classes : Pascal case
- ❖ Méthodes et attributs : Camel case
- ❖ Constantes : Screaming Snake case