Dossier de conception

| Révision | Rédacteurs | Date | Objet |
| --- | --- | --- | --- |
| 1 | Mohamed BOUNACEUR  Noémie LETENNEUR  Frédéric SMEE | 02/07/2024 | Création du document |
| 2 | Mohamed BOUNACEUR  Noémie LETENNEUR  Frédéric SMEE | 02/07/2024 | Avancement du document |

# 

# 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 **XXXXX**.

Ce document présente :

* Le diagramme de classes
* Le modèle physique de données

# 2. SOMMAIRE

## 2.1.1 Table des matières

[**1. INTRODUCTION 2**](#_gv0vae19z84q)

[1.1 Objet du document 2](#_6lkcuejr5syf)

[**2. SOMMAIRE 3**](#_77u6e775jvdz)

[2.1.1 Table des matières 3](#_d8hoxj9wetek)

[**3. ARCHITECTURE LOGICIELLE 4**](#_1cqbsdomgg2w)

[3.1 Produit et versions 4](#_qkosfxv0x39q)

[3.1.1 Langages, frameworks et librairies spécifiques 4](#_4vc1hzhs8n21)

[3.1.2 Serveur de base de données 4](#_yzoo1kq0khem)

[**4. FOCUS TECHNIQUE 5**](#_thu21nk0mo9w)

[4.1 Diagramme de classes métier 5](#_3tf377as6kyi)

[4.2 Modèle physique de données 5](#_wf8i8ocmmcc7)

[4.3 Règles de développement côté back 5](#_m2l8dxlhsqg)

[**5. TEST ET INTÉGRATION 6**](#_quphmqq0rc04)

[5.1 Stratégie de tests 6](#_8j9oq16kuaqd)

[5.2 Indicateurs de qualité de code 6](#_yobavhjfzgwy)

# 3. ARCHITECTURE LOGICIELLE

## 3.1 Produits et versions

### 3.1.1 Langages, frameworks et librairies spécifiques

| **Nom** | **Version** |
| --- | --- |
| Langage Java | 17+ |
| Spring Boot |  |
| JPA |  |
| Angular | 17+ |
| iText (générateur de PDF) |  |

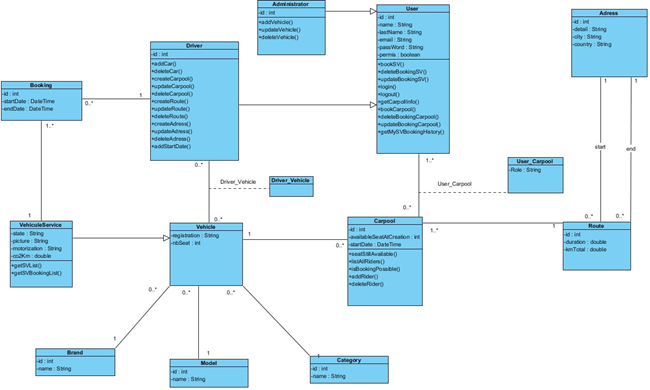
### 3.1.2 Serveur de base de données

| **Nom** | **Version** |
| --- | --- |
| MySQL | 8.4.1 LTS |

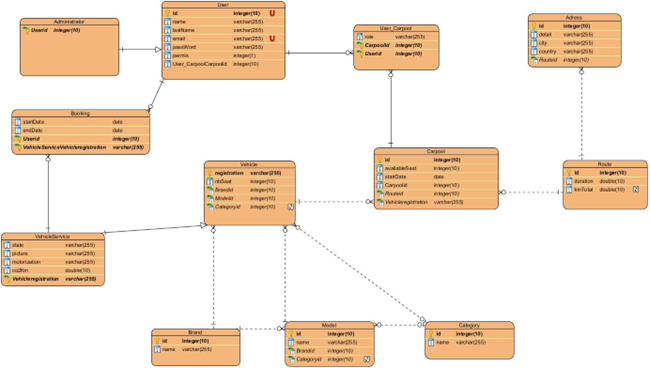
### 

# 4. FOCUS TECHNIQUE

## 4.1 Diagramme de classes métier

****

## 4.2 Diagramme entités-relations

****

## 4.3 Règles de développement côté back

Règles de développement :

* Javadoc
* Indentation du code
* **…**

Découpage en couches : **à repréciser quand vous aurez vu Spring Boot.**

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 ou Repositories ?
* Les DTOs ?
* Les classes de services ?
* Les classes utilitaires ? exemple : classe DateUtils pour le traitement des dates

# 5. TEST ET INTÉGRATION

## 5.1 Stratégie de tests

Il y aura des tests unitaires à développer pour tester le code de toutes les classes de services.

**Renseigner autre si vous le souhaitez…**

## 5.2 Indicateurs de qualité de code

Renseigner à 100% la javadoc.

Règles de nommage des classes, des packages :

* **…**

Nom du garant de la qualité : XXXXX (Scrum master de l’équipe)