

**Dossier de conception**

>> Diginamic.fr

|  |  |  |  |
| --- | --- | --- | --- |
| **Révision** | **Rédacteurs** | **Date** | **Objet** |
| 1 | LV.CHARVET, M. PRIN, E. AFENYO | 22/11/2024 | Création du document |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Objet du document

Ce document a pour objectif de présenter l’essentiel des questions techniques liées à la mise en place de l’application "Mission et note de frais”.

Ce document présente :

* Le diagramme de classes
* Le modèle physique de données

# Sommaire

### Table des matières

[1 Introduction 2](#_Toc182817810)

[1.1 Objet du document 2](#_Toc182817811)

[2 Sommaire 3](#_Toc182817812)

[2.1.1 Table des matières 3](#_Toc182817813)

[3 Architecture logicielle 4](#_Toc182817814)

[3.1 Produits et versions 4](#_Toc182817815)

[3.1.1 Langages, frameworks et librairies spécifiques 4](#_Toc182817816)

[3.1.2 Serveur de base de données 4](#_Toc182817817)

[4 Focus Technique 5](#_Toc182817818)

[4.1 Diagramme de classes métier (ou MCD) 5](#_Toc182817819)

[4.2 Diagramme entités relations (ou MPD) 5](#_Toc182817820)

[4.3 Règles de développement coté back 5](#_Toc182817821)

[5 TESTS et intégration 6](#_Toc182817822)

[5.1 Stratégie de tests 6](#_Toc182817823)

[5.2 Indicateurs de qualité de code 6](#_Toc182817824)

# Architecture logicielle

## Produits et versions

### Langages, frameworks et librairies spécifiques

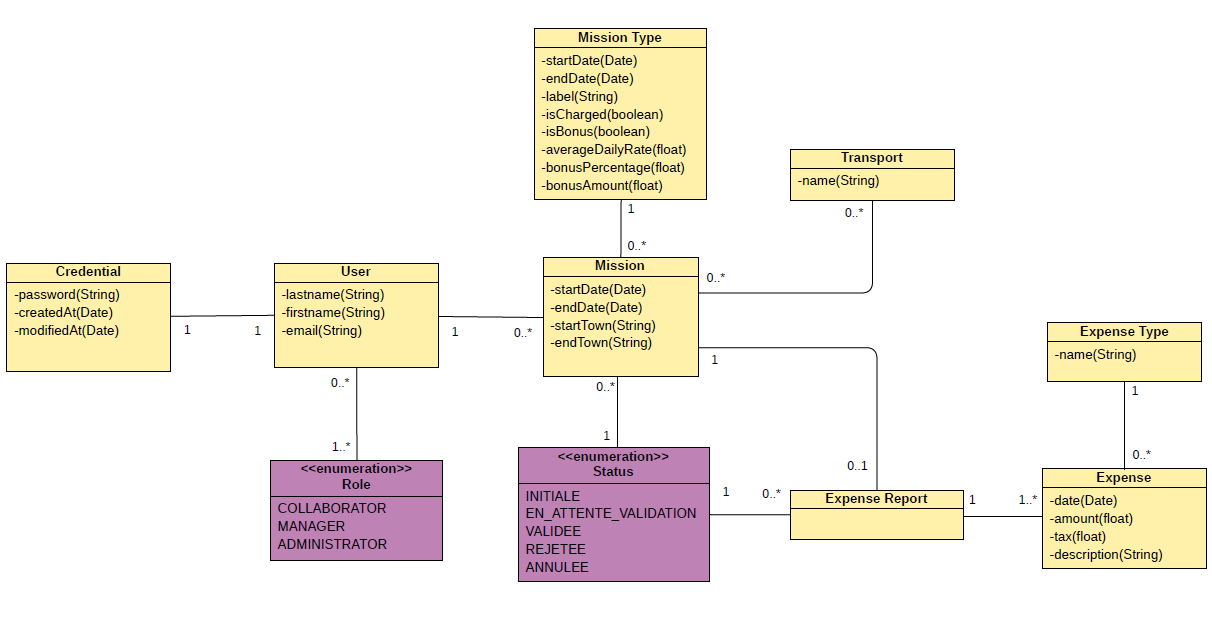
|  |  |
| --- | --- |
| **Nom** | **Version** |
| Langage Java | 17 |
| Spring Boot | 3.3.5 |
| JPA | 6.5.3 |
| Angular | 12 |
| iText (génération de PDF) | 5.5.13.4 |
| Apache POI (génération d’excel) | 5.3.0 |

### Serveur de base de données

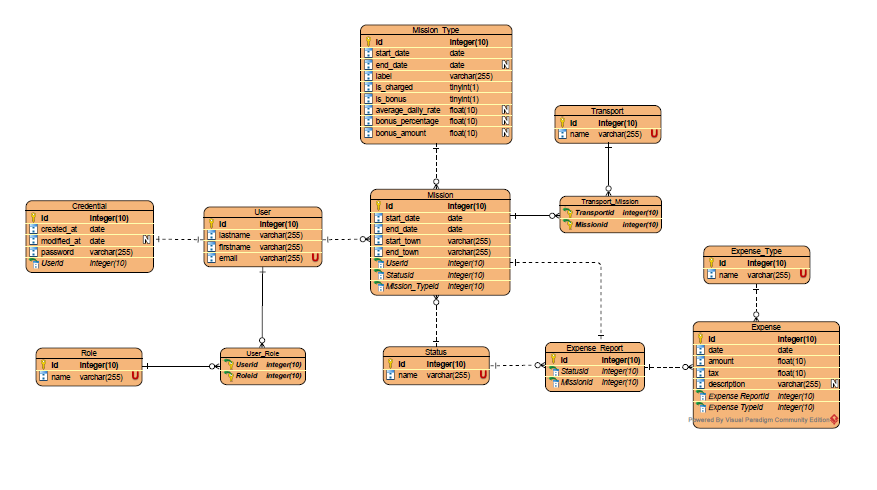
|  |  |
| --- | --- |
| **Nom** | **Version** |
| MySQL | Ver 9.1 Distrib 10.4.32-MariaDB |

# Focus Technique

## Diagramme de classes métier (ou MCD)



## Diagramme entités relations (ou MPD)



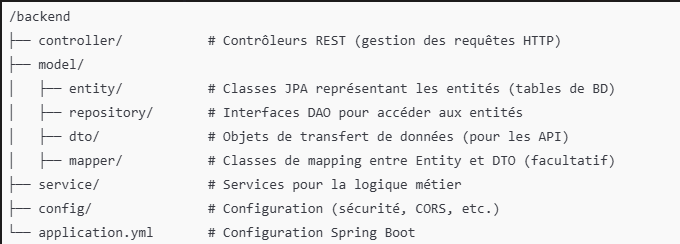
## Règles de développement coté back

**Règles de développement :**

* Javadoc
* Indentation du code

**Découpage en couches** :

* Couche contrôleur
* Couche de services qui va réaliser les contrôles métier.
* Couche repository
* Couche DTO : le contrôleur renvoie vers le front des instances de DTO.TESTS et intégration



## Stratégie de tests

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

Les tests seront réalisés avec mockito et JUnit pour la couche service et avec postman pour la couche controller.

## Indicateurs de qualité de code

Renseigner à 100% la javadoc.

Règles de nommage des classes, des packages : conventions standards à respecter