



Dossier de conception

>> Diginamic.fr

Révisio n	Rédacteurs	Date	Objet
1	R. BONNAMY	17/02/202 1	Création 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 XXXXXX.

Ce document présente :

- Le diagramme de classes
- Le modèle physique de données

2 SOMMAIRE

2.1.1 Table des matières

1	<u>INTRODUCTION</u>	2	
1.1	<u>Objet du document</u>		2
2	<u>SOMMAIRE</u>	3	
2.1.1	<u>Table des matières</u>		3
3	<u>ARCHITECTURE LOGICIELLE</u>	4	
3.1	<u>Produits et versions</u>		4
3.1.1	<u>Langages, frameworks et librairies spécifiques</u>		4
3.1.2	<u>Serveur de base de données</u>		4
4	<u>FOCUS TECHNIQUE</u>	5	
4.1	<u>Diagramme de classes métier (ou MCD)</u>		5
4.2	<u>Diagramme entités relations (ou MPD)</u>		5
4.3	<u>Règles de développement coté back</u>		5
5	<u>TESTS ET INTÉGRATION</u>	6	
5.1	<u>Stratégie de tests</u>		6
5.2	<u>Indicateurs de qualité de code</u>		6

3 ARCHITECTURE LOGICIELLE

3.1 Produits et versions

3.1.1 Langages, frameworks et librairies spécifiques

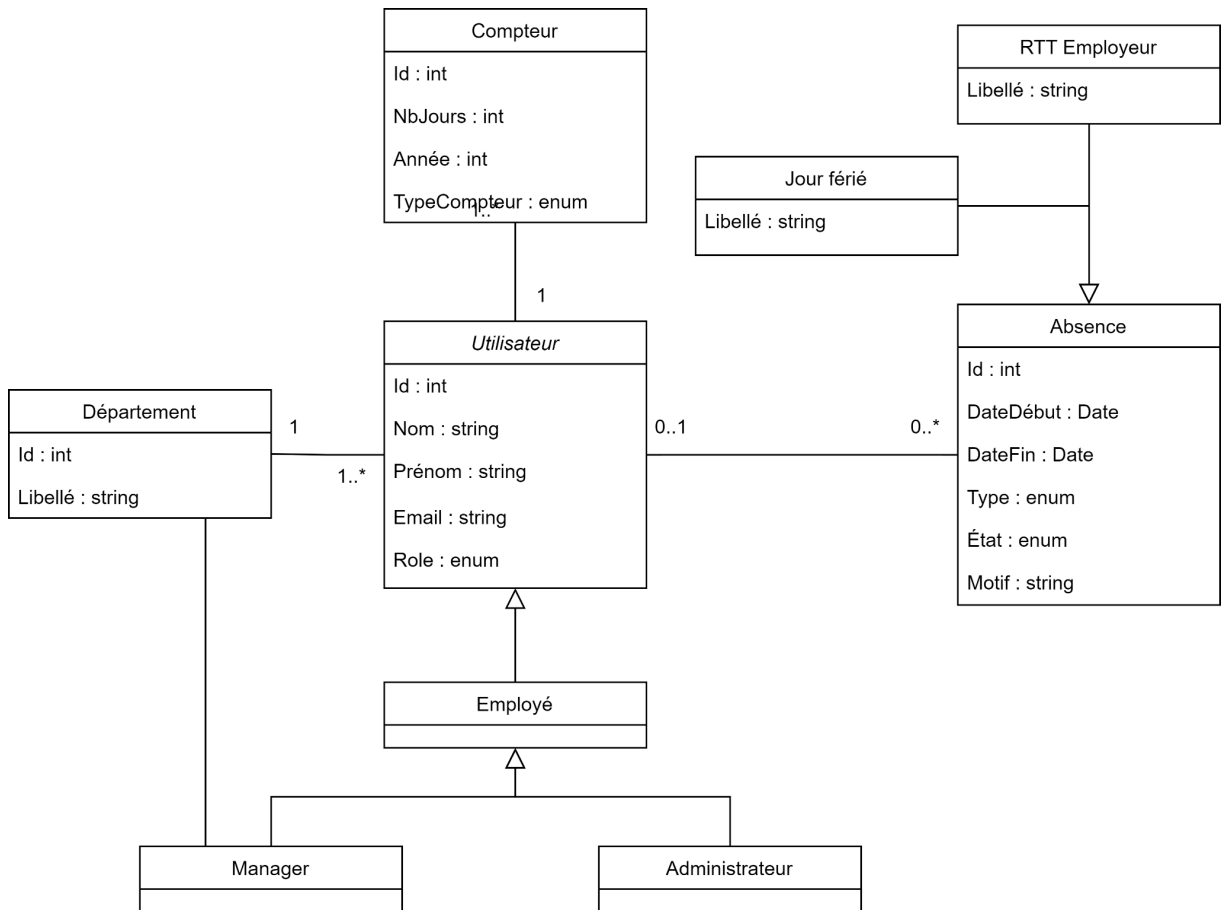
<u>Nom</u>	<u>Version</u>
Langage Java	17
Spring Boot	3.3.5
JPA	Dernière version
Angular	17
iText (génération de PDF)	7.1.3
Apache POI (génération d'excel)	5.2.5

3.1.2 Serveur de base de données

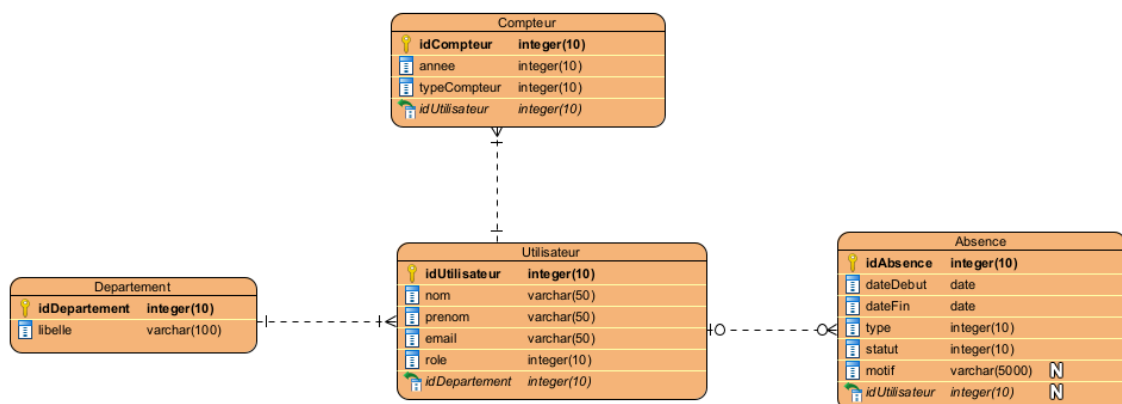
<u>Nom</u>	<u>Version</u>
MySQL	8.3.0

4 FOCUS TECHNIQUE

4.1 Diagramme de classes métier (ou MCD)



4.2 Diagramme entités relations (ou MPD)



4.3 Règles de développement coté back

Règles de développement :

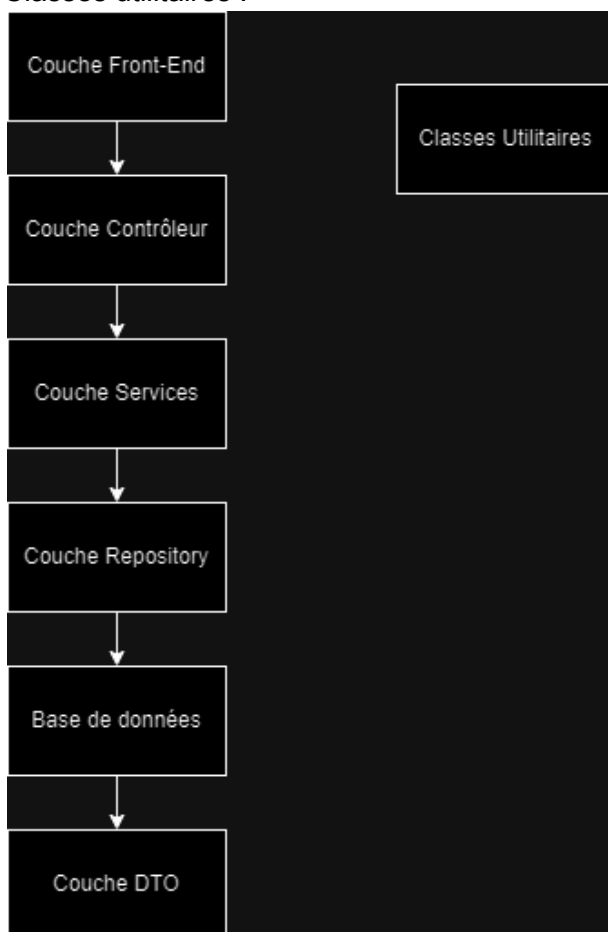
- Javadoc
- Indentation du code
- ...

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

Découpage en couches :

décrire les différentes 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.
- Classes utilitaires :



Couche contrôleur :

Cette couche est responsable de la gestion des requêtes http entrantes et des réponses associées. Elle agit comme une interface entre le client et le reste de l'application.

Couche services:

Cette couche contient la logique métier de l'application. Elle effectue les vérifications et traitements nécessaires avant de faire appel à la couche repository. C'est ici que se trouveront les vérifications sur la cohérence des dates par exemple.

Couche repository :

Cette couche est responsable de l'accès aux données et de leur persistance. Elle communique avec la base de données et retourne des images des objets stockés en base.

Couche DTO :

Les objets DTO (Data Transfer Objects) sont des structures de données qui servent à transmettre des informations entre la couche contrôleur et le front-end. Ils permettent d'éviter d'exposer directement les entités de la base de données, d'éviter les boucles infinies entre les objets qui se référencent entre eux et permet de ne manipuler seulement les informations nécessaires à un certain traitement.

Classes Utilitaires:

Les classes utilitaires regroupent des fonctions réutilisables et indépendantes des couches. Dans notre application, nous y trouverons les différentes énumérations ainsi que certains traitements sur les dates. En effet, ces actions n'étant pas dépendantes d'une couche métier en particulier, il est de mise de les mettre à disposition d'un point de vue global dans notre application afin d'éviter la duplication de code et de garder une même logique de traitement d'un développeur à l'autre.

5 TESTS 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.

5.2 *Indicateurs de qualité de code*

Renseigner à 100% la javadoc.

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