

Université de CY Cergy Paris Université



## **Projet Final JEE**

### **Groupe :**

DOS SANTOS Elyes

MESBAHY Rémi

ALI MLADJAO Ryan

NICOLAO Hyacinthe

THOYER Roman

### **Professeur référent :**

HADDACHE Mohamed

Année Universitaire : 2025/2026

## Table des matières

I.	Introduction.....	3
A.	Contexte et Motivation .....	3
B.	Description du Projet .....	3
C.	Technologies et outils utilisés .....	3
II.	Méthodologie de développement .....	4
A.	Organisation .....	4
B.	Analyse des besoins .....	4
C.	Choix des technologies .....	5
D.	Exemple de travail visuel : application de base.....	6
E.	Exemple de travail visuel : version Spring Boot .....	7
III.	Implémentation.....	9
A.	Architecture de l'application.....	9
B.	Spring Boot .....	9
C.	Modèle Logique de Données (MLD).....	10
D.	Modèle Conceptuel des données (MCD) .....	11
IV.	Difficultés rencontrées.....	11
A.	Configuration des Mappings avec Hibernate.....	11
B.	Difficultés avec les fichiers JSP .....	11
C.	Passage à la version Spring Boot .....	12
D.	Résumé des problèmes .....	12
V.	Conclusion et Améliorations .....	13
A.	Conclusion .....	13
B.	Améliorations.....	13
C.	Remerciements.....	14
VI.	Annexes .....	14
A.	Captures d'écran supplémentaires .....	14
VII.	Bibliographie .....	16
A.	Documentation officielle.....	16
B.	Tutoriels et ressources en ligne.....	16
C.	Outils utilisés .....	16

# **I. Introduction**

## **A. Contexte et Motivation**

Le projet de fin de semestre constitue une étape importante dans le cursus de chaque cycle ingénieur. Cette étape permet d'acquérir non seulement une nouvelle expérience précieuse en gestion de projet, mais nous a aussi permis de coder dans un nouveau langage de programmation, alliant le Java et divers langages de programmation de création de site internet. Il nous a permis d'exercer notre autonomie, de relever de nouveaux défis, de trouver des solutions à divers problèmes, d'envisager plusieurs options pour chaque cas d'usage, et de gérer la cohésion de groupe. Toutes ces compétences sont essentielles dans la formation d'un futur ingénieur.

## **B. Description du Projet**

Ainsi, dans le cadre du module de développement web avancé, notre équipe à entrepris la réalisation d'une application web permettant de gérer efficacement, pour une entreprise, ses ressources humaines, ses départements, ses projets internes ainsi que la génération des fiches de paie mensuelles. Ce projet s'inscrit dans un contexte entrepreneurial, visant à simplifier, pour le responsable, la gestion des employés, des départements, des projets d'entreprise, etc.

L'objectif principal de ce projet est de développer une plateforme intuitive et robuste, permettant d'automatiser des tâches essentielles telles que :

- Création de nouveaux employés, mise à jour de leurs informations, suppression, recherche avancée et affectation à un ou plusieurs projets.
- Ajout d'un département, visualisation de la liste des départements ainsi que les membres d'un département, affectation d'employés.
- Création, modification, suppression et suivi de l'état d'avancement d'un projet.
- Génération mensuelle automatique des fiches de paie, consultations des fiches existantes et possibilité de générer une version imprimable.
- Sécurisation de la plateforme, via un système de rôle prédéfinis (Administrateur, Chef de département, Chef de projet, Employé...)

## **C. Technologies et outils utilisés**

Pour répondre à ces exigences, nous avons utilisé des technologies actuelles et performantes, notamment les « Servlets » pour le backend, « Hibernate » pour la gestion des données, et une base de données relationnelle comme « MySQL ». L'architecture « MVC » (Modèle-Vue-Contrôleur) a été adoptée pour assurer une séparation claire des responsabilités et améliorer la maintenabilité de l'application. Enfin, la personnalisation

graphique de l'application, afin de la rendre moderne et ergonomique, a été réalisée grâce à l'utilisation de feuilles de style CSS.

Ce rapport détaille les différentes étapes de réalisation, de l'analyse des besoins à l'implémentation finale, en passant par la conception des modèles et la mise en œuvre des fonctionnalités principales. Nous présentons également les résultats obtenus et les perspectives d'amélioration, pour une possible évolution future de l'application.

## **II. Méthodologie de développement**

### **A. Organisation**

Durant la première partie du projet, l'organisation a été principalement effectuée de manière orale, basée sur une réunion quotidienne au cours de laquelle un point était fait sur le travail accompli et celui à venir. Par la suite, nous nous assignions chacun une tâche à réaliser, seul ou en groupe, en fonction de sa complexité et de sa durée. Chaque membre de notre groupe étant parfaitement à l'aise avec le télétravail, nous avons principalement travaillé à distance. Nous restions connectés sur un même groupe Discord, ce qui nous permettait d'échanger rapidement et facilement avec tous les membres de celui-ci en cas de besoin. De plus, la possibilité de partager nos écrans et d'utiliser un dépôt de code communautaire sur IntelliJ par exemple, facilitait la mise en commun de nos travaux et l'assistance mutuelle en cas de nécessité. À la suite d'une réunion, nous avons amélioré notre système d'organisation en utilisant Google Agenda ainsi que Jira, ce qui nous a permis de mieux visualiser toutes les tâches en cours, celles déjà réalisées, ainsi que la répartition du travail entre les membres du groupe.

### **B. Analyse des besoins**

L'analyse des besoins a permis de définir les fonctionnalités essentielles de l'application ainsi que les attentes des différents types d'utilisateurs. Ces utilisateurs, définis selon leurs responsabilités au sein de l'entreprise, sont :

- **Administrateur :**

Il possède un contrôle global sur l'application. Il peut gérer l'ensemble des employés, des départements, des projets ainsi que les fiches de paie. Il a également la possibilité d'administrer les comptes utilisateurs, d'attribuer des rôles, et d'accéder à l'ensemble des statistiques internes.

- **Chef de département :**

Il supervise son département et assure la gestion des employés qui y sont affectés. Il peut consulter la liste des membres du département, valider et suivre les affectations, et intervenir dans la gestion des projets associés à son équipe.

- **Chef de projet :**  
Il est responsable de la gestion de ces projets. Il peut créer ou modifier un projet, suivre son avancement (en cours, terminé, ou annulé), et gérer l'affectation des employés participant au projet.
- **Employé :**  
Il peut consulter ses informations personnelles, ses affectations à des projets, et accéder à ses fiches de paie. C'est l'utilisateur standard.

Les fonctionnalités clé identifiées sont :

- **Gestion des employés :**  
Ajout, mise à jour, suppression, recherche, affichage des informations et affectation à un département ou à un projet.
- **Gestion des départements**  
Ajout d'un département, affichage de la liste des départements, visualisation des membres et affectation des employés.
- **Gestion des projets :**  
Création, modification et suppression d'un projet. Affectation des employés et suivi de l'avancement.
- **Gestion des fiches de paie :**  
Génération automatique des fiches de paie, calcul du salaire net, affichages, recherche et génération d'une fiche imprimable.
- **Authentification et Autorisation :**  
Ajout d'un système de sécurité permettant de restreindre l'accès aux fonctionnalités selon le rôle de l'utilisateur.

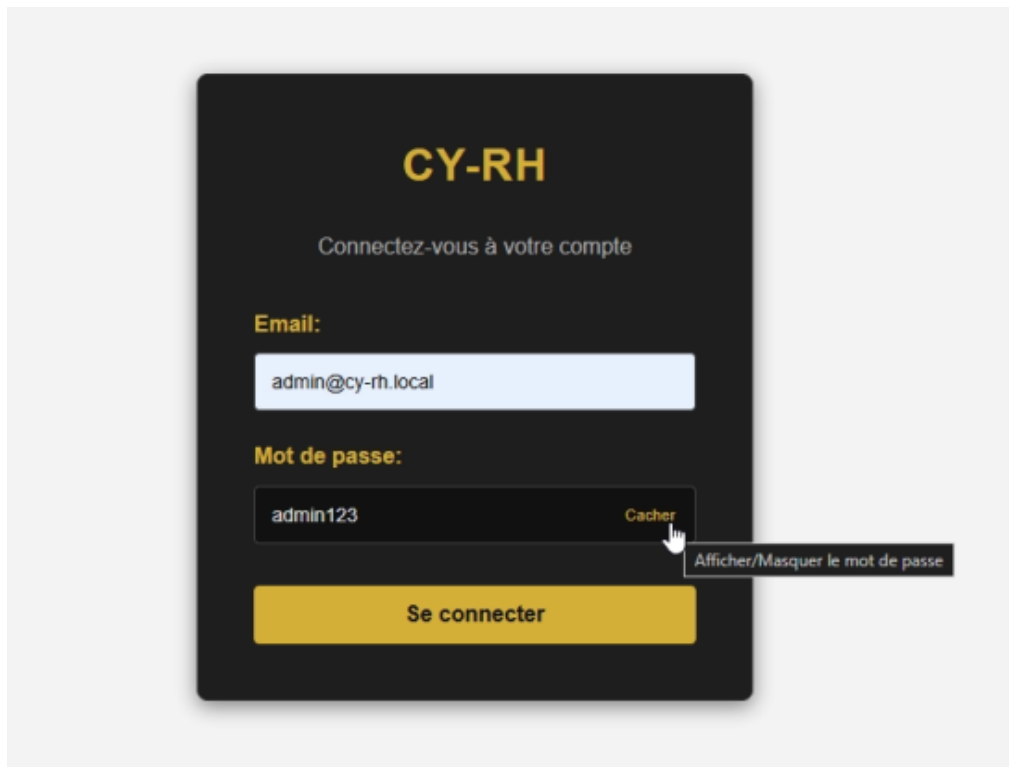
## **C. Choix des technologies**

Suite à ce travail préliminaire, nous avons choisi (ou avons été imposé) les technologies suivantes pour répondre aux besoins identifiés :

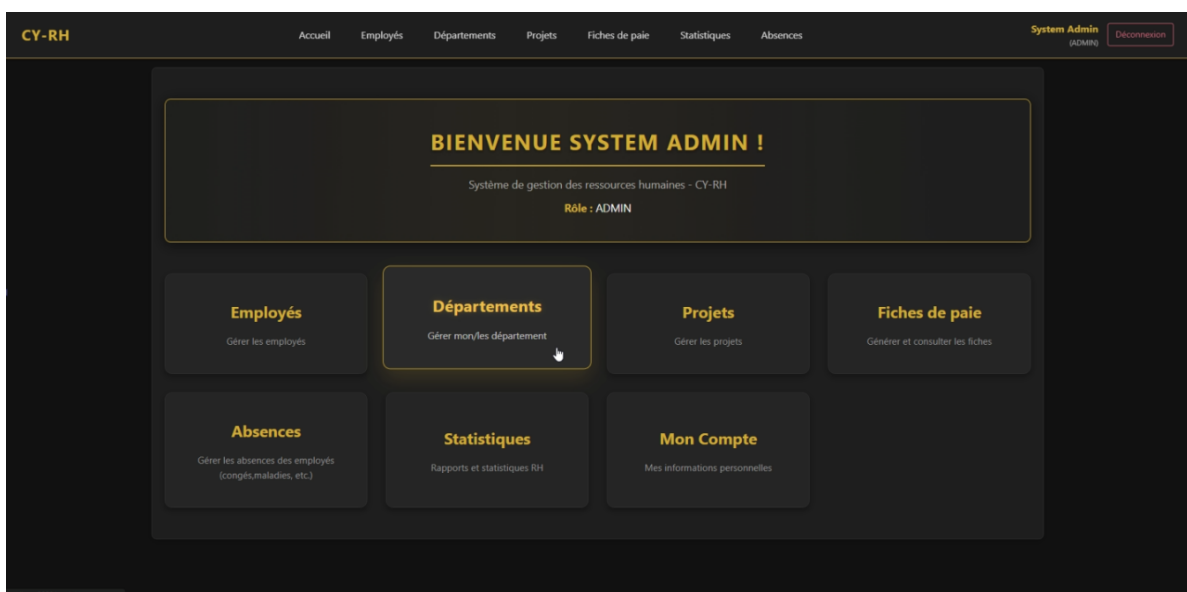
- Backend : les Servlets Java ont été utilisées pour gérer les requêtes http envoyées par le client.
- ORM (Object-Relational Mapping) : Hibernate pour la gestion des données relationnelles.
- Base de données : MySQL pour sa robustesse et sa comptabilité avec les frameworks choisis.
- Frontend : JSP, HTML, CSS et JavaScript pour une interface utilisateur moderne et dynamique.
- Serveur web : Apache Tomcat pour le déploiement.

## D. Exemple de travail visuel : application de base

Page de connexion : cette interface permet aux utilisateurs de se connecter à l'application grâce à leur email et leur mot de passe. La gestion des rôles détermine ensuite les fonctionnalités accessibles.



Page d'accueil : cette page constitue le point d'entrée principal de l'application après la connexion. Elle présente une vue d'ensemble des modules disponibles et permet à l'utilisateur d'accéder aux sections de gestion telles que les employés, les départements, les projets, les fiches de paie, les absences, les statistiques, ainsi que la visualisation de son propre compte.



Page de gestion des fiches de paie : cette page permet de consulter les fiches de paie générées, d'en créer une nouvelle, de modifier et d'ajouter une prime si l'utilisateur connecté est un administrateur (comme sur la capture d'écran ci-dessous).

**Gestion des Fiches de Paie**

Créer une Fiche de Paie | Mes Fiches

Rechercher des fiches

Tous les employés -- | Toutes les années -- | Tous les mois -- | Rechercher

	SALAIRE BASE	PRIMES	DÉDUCTIONS	NET À PAYER	DATE GÉNÉRATION	ACTIONS
25	32 000,00 €	+ 0,00 €	- 120,00 €	31 880,00 €	28/02/2025	<a href="#">Voir</a> <a href="#">Ajouter Prime</a> <a href="#">Modifier</a> <a href="#">Supprimer</a>
25	48 000,00 €	+ 1 000,00 €	- 0,00 €	49 000,00 €	28/02/2025	<a href="#">Voir</a> <a href="#">Ajouter Prime</a> <a href="#">Modifier</a> <a href="#">Supprimer</a>
25	46 000,00 €	+ 200,00 €	- 0,00 €	46 200,00 €	28/02/2025	<a href="#">Voir</a> <a href="#">Ajouter Prime</a> <a href="#">Modifier</a> <a href="#">Supprimer</a>
25	52 000,00 €	+ 400,00 €	- 0,00 €	52 400,00 €	28/02/2025	<a href="#">Voir</a> <a href="#">Ajouter Prime</a> <a href="#">Modifier</a> <a href="#">Supprimer</a>
25	35 000,00 €	+ 0,00 €	- 100,00 €	34 900,00 €	28/02/2025	<a href="#">Voir</a> <a href="#">Ajouter Prime</a> <a href="#">Modifier</a> <a href="#">Supprimer</a>
Février 2025	42 000,00 €	+ 0,00 €	- 0,00 €	42 000,00 €	28/02/2025	<a href="#">Voir</a> <a href="#">Ajouter Prime</a> <a href="#">Modifier</a> <a href="#">Supprimer</a>

Le site étant très bien fourni en termes de CSS, plus de capture d'écran sont disponibles dans la partie Annexes, à la fin de ce rapport.

## E. Exemple de travail visuel : version Spring Boot

Nous avons décidé de réaliser un style visuel différent de la version de base de l'application, de manière à bien différencier chacune de celles-ci.

**Connexion**

Connectez-vous pour accéder au système RH

Email professionnel

admin@cy-rh.local

Mot de passe

.....

[Se connecter](#)

**Information :** Seuls les employés enregistrés peuvent se connecter. Contactez votre administrateur RH si vous rencontrez des difficultés.

Page des départements : cette page affiche la liste des départements existants ainsi que les principales informations associées. Elle permet également d'accéder aux détails d'un département, notamment la liste de ses membres, ainsi qu'aux actions de gestion lorsque le rôle de l'utilisateur le permet.

**Bienvenue dans le système de gestion RH**

Sélectionnez une option dans le menu de gauche pour commencer

**Liste des départements**

Rechercher un département (intitulé, chef...)

Intitulé: Tous Chef de département (ID): Tous

[Réinitialiser](#)

ID	INTITULÉ	CHEF DE DÉPARTEMENT (ID)
1	DIRECTION GENERALE	1
2	RESSOURCES HUMAINES	4
3	INFORMATIQUE	7
4	FINANCE	13
5	MARKETING	17

[Gérer les départements \(Ajouter/Modifier/Supprimer\)](#)

Page des projets : cette page présente l'ensemble des projets enregistrés dans l'application. Pour chaque projet, l'utilisateur peut consulter son état d'avancement, ses informations principales, ainsi que les employés qui y sont affectés. Des actions de gestion sont accessibles selon le rôle de l'utilisateur.

**Bienvenue dans le système de gestion RH**

Sélectionnez une option dans le menu de gauche pour commencer

**Liste des projets**

Rechercher un projet (nom, chef, département...)

Nom du projet: Tous État: Tous Date début: Tous Date fin prévue: Tous Chef de projet: Tous Département: Tous

[Réinitialiser](#)

ID	NOM DU PROJET	ÉTAT	DATE DÉBUT	DATE FIN PRÉVUE	CHEF DE PROJET	DÉPARTEMENT
1	Audit Financier 2025					
2	CRM Commercial					
3	Campagne Publicitaire Q2	EN_COURS	2025-01-15	2025-06-30	Karim Ali	INFORMATIQUE
4	Formation Cybersécurité	EN_COURS	2025-01-20	2025-04-30	Karim Ali	INFORMATIQUE
5	Intranet RH					
6	Migration Comptable ERP					
7	Optimisation Supply Chain	PAS_COMMENCE	2025-05-01	2025-12-31	Thierry Nguyen	INFORMATIQUE
8	Plateforme E-commerce					
9	Refonte Infrastructure	TERMINE	2024-10-15	2024-12-20	Thierry Nguyen	INFORMATIQUE
10	Site Vitrine Corporate					
11	Transformation Digitale	EN_COURS	2025-02-01	2025-08-31	Nadia Petit	FINANCE

[Gérer les projets \(Ajouter/Modifier/Supprimer\)](#)

L'application permet également, sur la page des fiches de paie, de générer automatiquement une fiche de paie au format PDF. Le document présente l'ensemble des informations essentielles : identité de l'employé, période de paie, salaire de base, primes, déductions ainsi que le montant net à payer. Ce PDF est mis en forme de manière lisible,



et peut être téléchargé puis imprimé, offrant ainsi un support officiel et prêt à l'usage pour les responsables comme pour les employés.

**FICHE DE PAIE**  
Période : JANVIER 2025

**Employeur : CY-RH Project**  
123 Rue de l'Exemple, 75000 Paris

**Employé : Karim ALI**  
Poste : Chef de Projet IT  
Matricule : EMP008

Détails des Éléments de Paie :

Intitulé	Montant	Type
Salaire de Base	42000,00 €	Gain
Primes (Exceptionnel ou Objectifs)	300,00 €	Gain
Cotisations / Taxes (Simplifié)	-50,00 €	Retenue

**NET À PAYER :**  
**42250,00 €**

Fiche générée automatiquement le : 2025-11-30

### III. Implémentation

#### A. Architecture de l'application

L'application a été conçue en suivant le modèle MVC (Modèle-Vue-Contrôleur), qui sépare les différentes responsabilités de manière claire pour améliorer la maintenabilité et l'évolutivité du projet. Voici une description des trois couches principales :

- **Modèle** : gère la logique métier et les interactions avec la base de données. Par exemple, les entités telles que Département, Employé, Projet et FicheDePaie sont implémentées sous forme de classe Java. Chaque classe correspond à une table SQL et Hibernate assure la persistance entre les objets Java et la base MySQL.
- **Vue** : représente l'interface utilisateur. Les pages JSP sont utilisées pour afficher les données dynamiques sous forme de HTML/CSS et JavaScript, permettant aux utilisateurs de naviguer et d'interagir avec l'application internet.
- **Contrôleur** : traite les requêtes http, gère la logique métier et orchestre les interactions entre le modèle et la vue. Les Servlets remplissent ce rôle. Les Servlets, telles que EmployeServlet ou ProjetServlet, récupèrent les données du modèle, effectuent les traitements nécessaires, puis les transmettent à la vue.

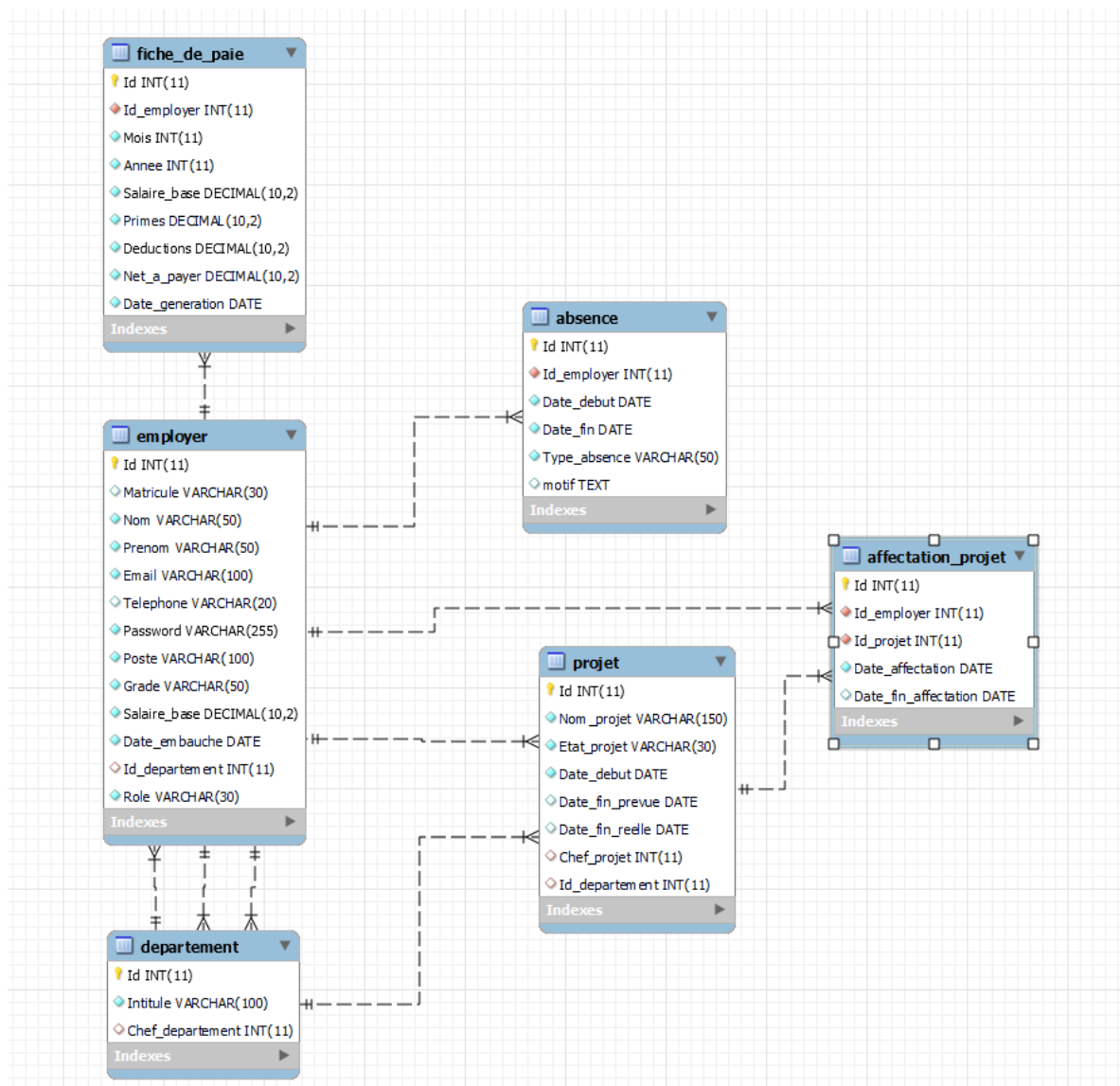
#### B. Spring Boot

Dans un second temps, nous avons réalisé, comme demandé dans le cahier des charges, une version de l'application en utilisant Spring Boot. Cette approche plus moderne du développement Java, simplifie la configuration en automatisant de nombreuses tâches techniques. Contrairement à l'architecture JEE classique (reposant sur les Servlets, les JSP), Spring Boot permet de structurer l'application autour de controllers, de services et de repositories.

Cette version de l'application reprend l'ensemble des fonctionnalités principales développées dans la version JEE initiale, tout en offrant une architecture plus modulaire, plus claire et plus facile à maintenir. Elle illustre ainsi la transition entre un développement Java traditionnel et les pratiques plus modernes, utilisées aujourd'hui dans l'industrie.

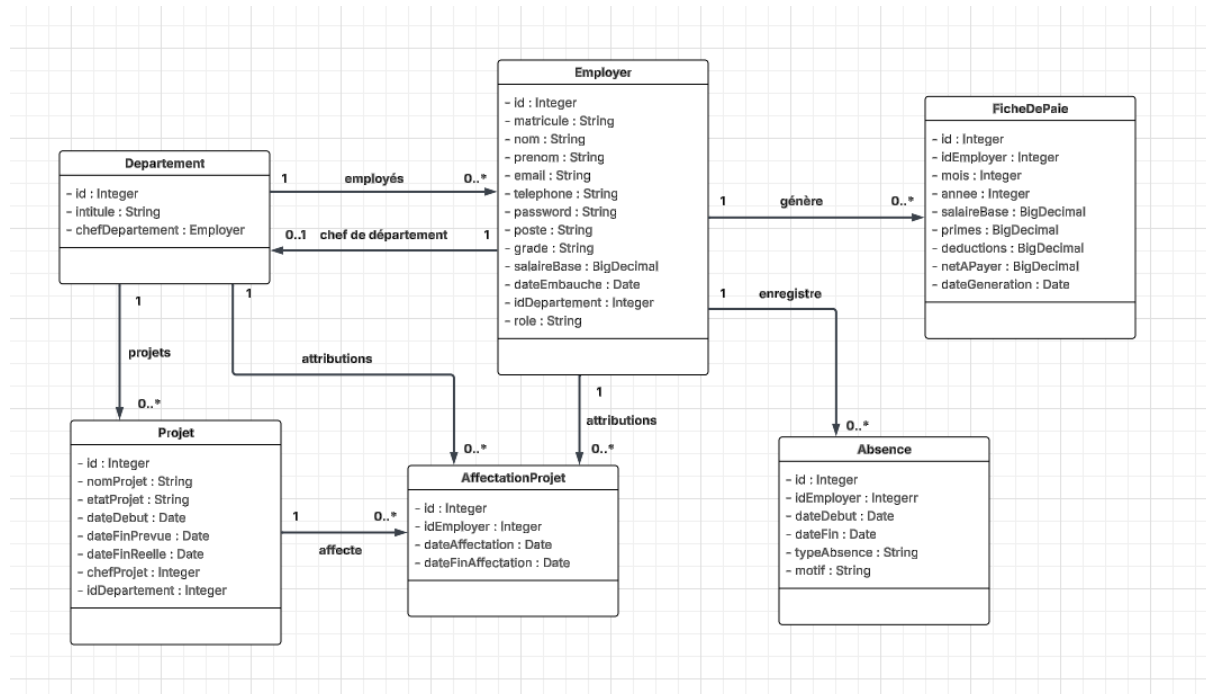
### C. Modèle Logique de Données (MLD)

Le Modèle Logique de Données (MLD) présenté ci-dessous représente la version finale de la base de données de l'application, donnant un aperçu de l'ensemble des tables ainsi que de leurs relations.



## D. Modèle Conceptuel des données (MCD)

Le diagramme de classes ci-dessous présente les principales entités du système ainsi que leurs relations.



## IV. Difficultés rencontrées

### A. Configuration des Mappings avec Hibernate

La configuration des mappings objet-relationnel avec Hibernate s'est avérée complexe, en particulier pour définir les relations entre les entités. Des erreurs comme « LazyInitializationException », ou des incohérences dans les données ont également été rencontrées. Ainsi, nous avons porté une attention particulière à la configuration des annotations pour les relations. Nous avons activé le chargement « eager » pour certaines relations critiques afin de limiter les problèmes liés à l'accès différé des données. Enfin, les transactions ont été gérées avec soin pour éviter des incohérences dans la base. Cela impliquait de s'assurer que chaque opération impliquait un « commit() » approprié.

### B. Difficultés avec les fichiers JSP

Les fichiers JSP, utilisés pour générer les pages web, ont parfois rendu difficile la séparation entre la logique de programmation et l'affichage. Cela a conduit à des problèmes tels que des erreurs inattendues dans les pages ou des difficultés à afficher correctement les données. Nous avons donc amélioré notre manière de travailler avec les JSP en réduisant la complexité dans ces fichiers. Nous avons mieux structuré notre code

pour que les données soient préparées avant d'être envoyées à l'affichage. Pour faciliter la gestion des données dynamiques, nous avons utilisé des outils standards qui simplifient l'insertion des informations dans les pages sans encombrer le code.

### **C. Passage à la version Spring Boot**

La transition vers la version Spring Boot de l'application a représenté de nombreux défis au cours du projet, en raison de la différence profonde avec l'architecture JEE classique utilisée dans la première partie. En effet, Spring Boot introduit une organisation plus modulaire, et comprendre comment chaque élément de cette architecture communique a nécessité un temps d'adaptation non négligeable.

Une des difficultés résidait dans le fonctionnement des balises et des expressions *Thymeleaf*, qui diffèrent beaucoup des JSP classiques. Manipuler correctement les expressions, les attributs dynamiques et la syntaxe spécifique de *Thymeleaf* a souvent mené à des erreurs difficiles à diagnostiquer, notamment lorsque les données ne s'affichaient pas comme prévu.

Un autre point complexe était d'apprendre à exploiter les *templates Thymeleaf* de manière efficace (incluant la réutilisation de fragments, la mise en page globale, la gestion des formulaires et la liaison de données). La plupart de ces notions étaient nouvelles pour nous, ce qui a nécessité des recherches, des essais et plusieurs corrections avant d'obtenir un affichage fonctionnel et cohérent.

Malgré ces difficultés, cette étape nous a permis de comprendre en profondeur le fonctionnement d'un framework moderne, et nous a permis d'en ressortir ces avantages, comme la facilité d'installation, ainsi qu'un code simple et plus condensé.

### **D. Résumé des problèmes**

Bien que ces problèmes aient ralenti l'avancé du développement, ils nous ont également permis de mieux comprendre les bases des Servlets, JSP, SpringBoot et Hibernate. Chaque défi technique rencontré a renforcé notre compréhension des outils utilisés, tout en nous poussant à optimiser nos configurations et notre gestion des données. Résoudre ces problèmes a été une opportunité d'apprendre et d'améliorer notre capacité à coder, à réfléchir ainsi qu'à pousser la coopération d'équipe, toujours plus loin.

## **V. Conclusion et Améliorations**

### **A. Conclusion**

La réalisation de cette application, permettant de digitaliser les processus internes d'une entreprise, s'est révélé être un défi stimulant et formateur. Malgré les contraintes techniques et les imprévus, nous avons su nous adapter aux nouvelles technologies qui nous étaient proposées, comme les Servlets, JSP, Hibernate, SpringBoot, etc. Celles-ci nous ont permis de développer une application fonctionnelle, respectant l'ensemble des fonctionnalités du cahier des charges, et répondant aux besoins des utilisateurs finaux :

- La conception et la modélisation des bases de données.
- La structuration d'une application web en respectant l'architecture MVC.
- L'utilisateur des ORM pour simplifier les interactions entre les objets Java et la base de données.
- La gestion de la sécurité et des autorisations dans un contexte multi-utilisateur.

### **B. Améliorations**

Tout d'abord, et comme mentionné précédemment, l'ensemble du cahier des charges est respecté et fonctionnel par l'application. Néanmoins, nous avons pu imaginer plusieurs améliorations « bonus », pour enrichir l'expérience de l'utilisateur.

Une première amélioration consisterait à intégrer un système de notifications ou de messagerie interne, ce qui permettrait aux administrateurs, aux chefs de départements ou aux chefs de projet, d'envoyer des alertes et des messages ciblés aux différents employés. Ce système pourrait également notifier automatiquement un employé lors de la création d'une nouvelle fiche de paie, d'une affectation à un projet, ou même de l'approche d'une échéance importante.

Une seconde amélioration concerne le module « départements », qui pourrait être enrichi en y ajoutant plus d'informations, comme la charge de travail actuelle ou un historique des employés ayant appartenu au département. Cela offrirait une vision plus détaillée de la structure interne de l'entreprise.

La gestion des absences, actuellement simplifiée dans notre version, pourrait également être améliorée. Aujourd'hui, pour chaque absence non justifiée, une déduction fixe sur salaire est appliquée. Il serait pertinent de rendre cela plus réaliste, en associant la pénalité à la durée de l'absence ainsi qu'au salaire de l'employé, afin d'obtenir un traitement plus précis et conforme aux pratiques réelles.

Enfin, une dernière amélioration que nous pensons pertinente, serait l'ajout d'une fonctionnalité de génération automatique de rapports PDF. Ceux-ci pourraient contenir : la liste des employés, les récapitulatifs des fiches de paie, les statistiques des projets, etc.

Ce type de fonctionnalité serait très utile pour les responsables RH et offrirait un moyen rapide d'expert les informations essentielles.

Une dernière amélioration possible serait l'ajout d'un mode clair/sombre, permettant à l'utilisateur de choisir le thème de l'interface selon sa préférence. Nous avons d'ailleurs souhaité représenter cette idée dans le projet : la version JEE adopte un thème sombre, tandis que la version Spring Boot utilise un thème clair, illustrant ainsi les deux variations possibles.

Ces améliorations, bien que facultatives, pourraient rendre l'application plus complète et proche d'un outil professionnel utilisé en entreprise.

## **C. Remerciements**

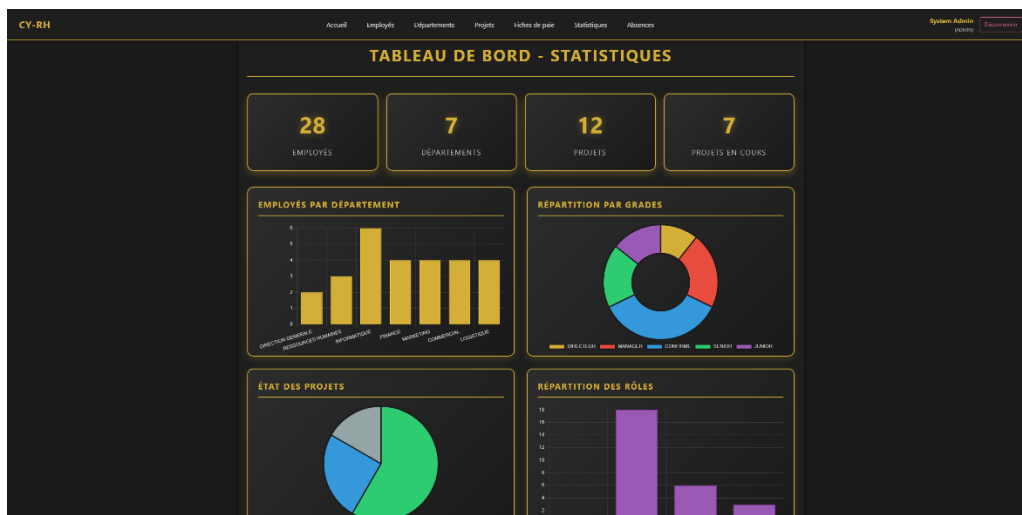
La réalisation de ce projet a présenté non seulement un intérêt pédagogique, mais il a aussi permis de développer des compétences techniques variées, allant de la programmation à la conception d'interfaces utilisateur, en passant par la gestion de bases de données et l'intégration de divers langages de programmation. Ce projet a ainsi été une excellente opportunité de mettre en pratique nos connaissances et de les enrichir dans un contexte concret et stimulant. Les compétences acquises et les leçons tirées de cette expérience constituent un atout précieux pour nos futurs projets, que ce soit dans un cadre académique ou, très bientôt, professionnel. Nous sommes confiants qu'avec de telles bases solides, nous pourrions relever des défis encore plus complexes à l'avenir. C'est pourquoi nous tenons à remercier monsieur HADDACHE Mohamed, pour sa pédagogie et son aide tout au long de la réalisation de ce projet, ainsi que le lecteur de ce rapport, pour le temps qu'il nous accorde.

## **VI. Annexes**

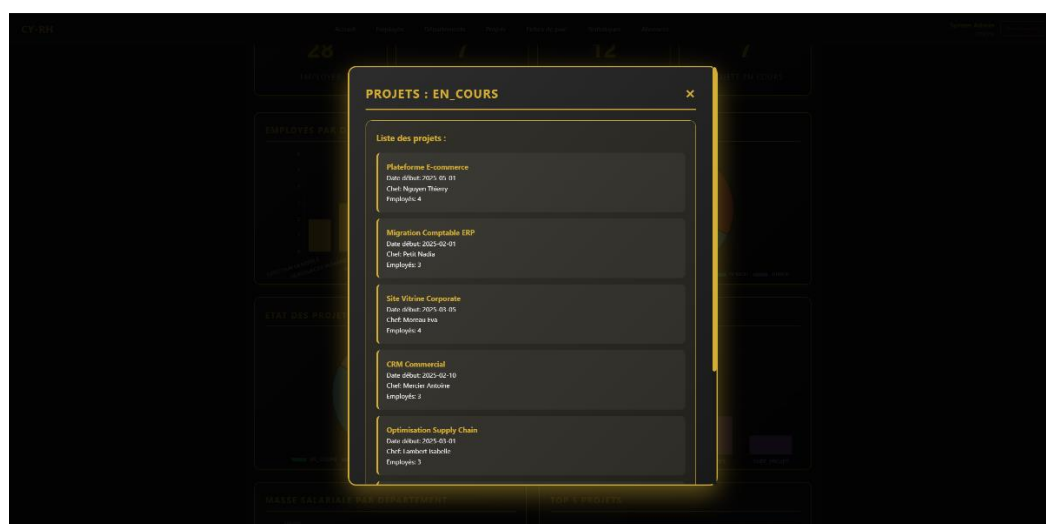
### **A. Captures d'écran supplémentaires**

Même si l'ensemble des fonctionnalités du cahier des charges sont respectées, nous voulions être complètement fier du travail rendu. C'est pourquoi nous avons aussi mis l'accent sur le design de l'application, pour qu'elle soit moderne, propre et agréable pour l'utilisateur. Nous fournissons donc les captures d'écrans supplémentaires ci-dessous.

Page statistiques : cette page regroupe plusieurs indicateurs clés de l'application. On y retrouve des données concernant les employés, les projets (en cours ou non) et les départements. Elles sont représentées sous forme de diagrammes circulaires, horizontaux et verticaux. Ceux-ci permettent de visualiser rapidement la répartition des projets, la distribution des employés, etc.



Sur chacun des différents type de diagramme, l'utilisateur peut cliquer sur une catégorie (par exemple, dans le diagramme « Etat des projets », la partie verte représentant les projets en cours), ce qui lui affichera une page listant les projets en cours. Cet ajout assure une recherche facile et efficace des différentes données de l'application, comme le montre la capture d'écran ci-dessous.



## VII. Bibliographie

### A. Documentation officielle

- Hibernate : <https://hibernate.org/>
- Servlets et JSP : <https://docs.oracle.com/javaee/>
- Apache Tomcat : <https://tomcat.apache.org/>
- Spring Boot : <https://spring.io/projects/spring-boot>
- Thymeleaf : <https://www.thymeleaf.org/documentation.html>

### B. Tutoriels et ressources en ligne

- Baeldung, *Introduction to Hibernate* : <https://www.baeldung.com/hibernate>
- Baeldung, Spring Boot Tutorials : <https://www.baeldung.com/spring-boot>
- GeeksforGeeks, *ServletBasics* : <https://geeksforgeeks.org/servlets/>
- Marco Codes Playlist :  
<https://www.youtube.com/playlist?list=PLIRBoI92yMam1HaUYrMAaPdbZKV1BFW0F>
- Oracles : <https://www.oracle.com/java/technologies/jee-tutorials.html>
- W3Schools – HTML/CSS Basics : <https://www.w3schools.com/>
- Spring Initializr (outil de génération Spring Boot) : <https://start.spring.io/>

### C. Outils utilisés

- IntelliJ IDEA : <https://www.jetbrains.com/idea/>
- MySQL Workbench : <https://dev.mysql.com/workbench/>
- Eclipse : <https://eclipseide.org/>
- Apache Maven : <https://maven.apache.org/>
- GitHub : <https://github.com/>