

Projet JEE 2025-2026 - ING2 GSI1 - Gestion d'entreprise

Spécifications fonctionnelles et techniques

Ulrich RANAIVOJAONA - Baptiste LAMBERT - Soufiane FIKRI - Waally BOUHADOU -
Valérie TEXIER

Table des matières

I - Analyse du besoin.....	2
I.1 Les acteurs.....	2
I.2 Les fonctionnalités.....	3
1.2.1 Gestion des employés.....	3
1.2.2 Gestion des départements.....	3
1.2.3 Gestion des projets.....	3
1.2.4 Gestion des fiches de paie.....	3
I.3 Cas d'utilisation.....	4
I.4 Gestion des autorisations et authentifications.....	4
II - Architecture du projet.....	7
II.1 Le modèle de données.....	7
II.2 L'implémentation de la base relationnelle.....	8
II.2.1 La base de données pour le projet avec Hibernate.....	8
II.2.2 La base de données pour le projet Spring boot.....	9
II.3 L'architecture applicative.....	10
II.3.1 Version Hibernate.....	10
II.3.2 Version Spring Boot.....	11
II.3.3 Structure des classes.....	12
III - Présentation des écrans.....	14
III.1 Le projet Hibernate.....	14
III.2 Le projet Spring Boot.....	17
IV - Organisation du travail.....	19

Dans le cadre de la numérisation de ses process, l'entreprise *FoodNCo* souhaite développer une solution de gestion informatique de ses ressources humaines et de son administration générale.

L'objectif de ce document est de présenter l'analyse de besoins et la solution **Gestion Entreprise** conçue pour les besoins de *FoodNCo*.

I - Analyse du besoin

Nous présentons ici les acteurs de *FoodNCo* et les fonctionnalités souhaitées. Les fonctionnalités pourront être accédées par plusieurs acteurs, selon les rôles. Nous listons donc les acteurs et analysons les besoins en fonctionnalités atomiques, qui seront ensuite regroupées en modules. Selon les cas d'utilisations identifiés, nous définirons ensuite une matrice croisant les acteurs et les fonctions.

I.1 Les acteurs

L'analyse fait ressortir les acteurs suivants :

- Le responsable RH
- Le chef de département
- Le chef de projet
- L'employé
- L'administration de la totalité de l'application est assurée par un administrateur général.

Chaque employé est caractérisé par :

- un matricule,
- un grade qui correspond à son expérience
- un rôle parmi ceux identifiés ci-dessus
- un rattachement à un ou plusieurs projets
- un rattachement à un département.

Le chef de projet, le chef de département, l'administrateur RH et l'administrateur sont des employés dotés d'un rôle particulier, disposant de droits supplémentaires sur le système selon ce rôle. Ce sont des acteurs à privilèges. Le chef de département est assigné à un département unique, tandis que le chef de projet peut être assigné à un ou plusieurs projets.

I.2 Les fonctionnalités

L'application doit permettre d'assurer les fonctions atomiques suivantes, dont les familles d'actions constituent les modules.

1.2.1 Gestion des employés

- Ajouter un nouvel employé
- Modifier les informations d'un employé
- Supprimer un employé
- Lister tous les employés
- Lister les employés par grade, poste
- Rechercher un employé par nom, prénom, matricule ou département
- Affecter un employé à un ou plusieurs projets

1.2.2 Gestion des départements

- Ajouter un département
- Lister les départements
- Affecter un employé à un département
- Visualiser les membres d'un département

1.2.3 Gestion des projets

- Créer, modifier, supprimer un projet
- Affecter des employés à un projet
- Suivre l'état d'avancement (état : en cours, terminé, annulé)

1.2.4 Gestion des fiches de paie

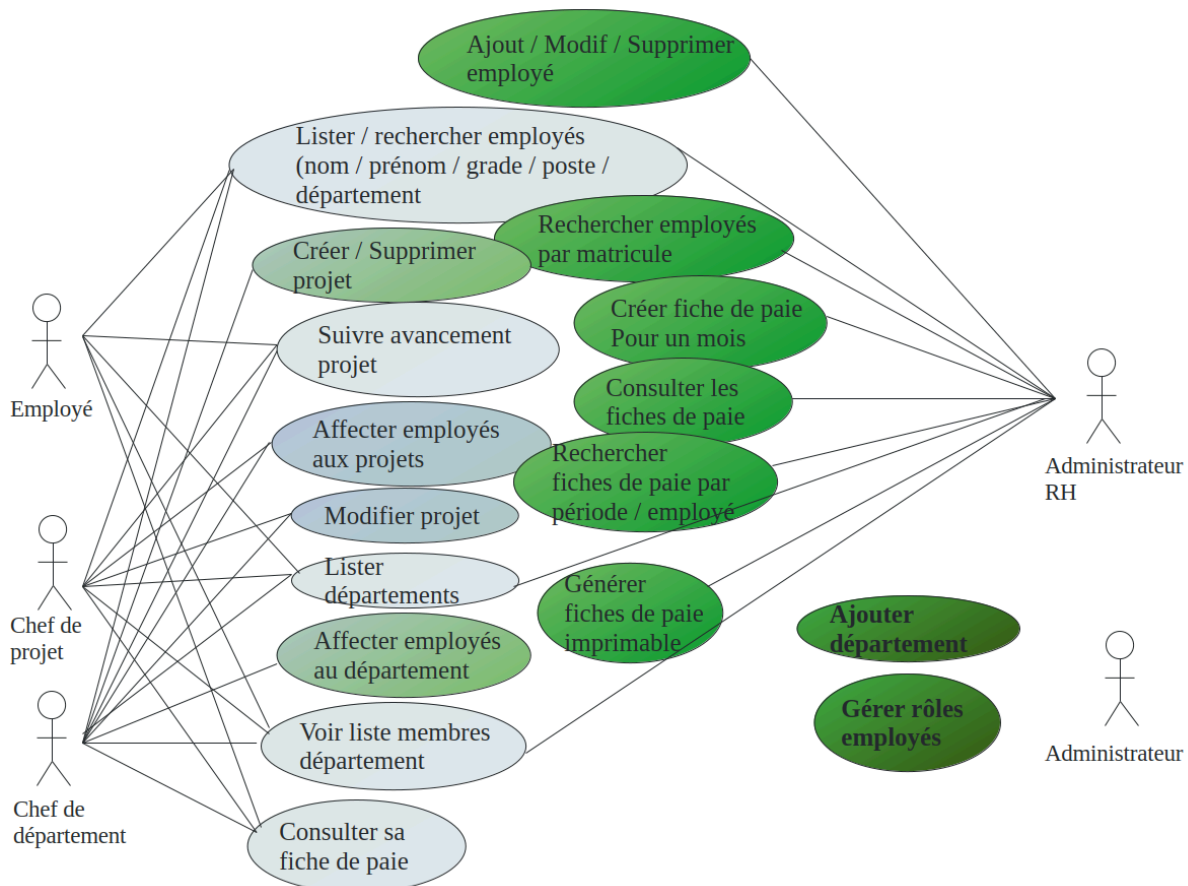
- Créer une fiche de paie pour un employé pour un mois donné
- Calcul automatique du net à payer : $\text{net} = \text{salaire de base} + \text{primes} - \text{déductions}$
- Consulter les fiches de paie d'un employé
- Générer une fiche de paie imprimable
- Rechercher les fiches par période ou par employé

Nous aurons également besoin de gérer l'authentification de manière centralisée, et un système d'autorisation qui associe des profils utilisateur selon le rôle et les fonctions nécessaires.

I.3 Cas d'utilisation

Selon les profils, les cas d'utilisation sont identifiés. Nous les illustrons ci-dessous. Afin de rester lisible, le schéma ne contiendra pas les traits entre l'administrateur et les cas. Cependant, l'administrateur étant l'acteur de plus haut niveau, ce dernier aura bien l'usage de l'ensemble des fonctionnalités représentées.

Les fonctions sont représentées avec l'un des 5 niveaux de couleurs. Plus la couleur est foncée, plus la fonction est "sensible" et accédée par un nombre restreint d'acteurs.



I.4 Gestion des autorisations et authentifications

En fonction des accès, certaines fonctionnalités ont été scindées en deux, de manière à gérer l'accès de l'employé "simple" sur ses propres données, et l'accès plus large de ces mêmes données sur l'ensemble des employés pour les acteurs à privilèges. C'est le cas par exemple de la consultation des fiches de paie. L'association des fonctionnalités nécessaires à chaque type d'acteur définit une matrice de droits. Cette matrice sera utilisée pour l'accès aux modules et fonctions. L'authentification dans l'application permettra de personnaliser les accès à la session ouverte par l'utilisateur d'un certain type.

L'analyse du croisement rôle / fonctionnalité nous procure la matrice de droits suivante.

	Employé	Ched de projet	Chef de département	Administrateur RH	Administrateur
Ajout nouvel employé				X	X
Modifier les informations d'un employé				X	X
Supprimer employé				X	X
Lister tous les employés	X	X	X	X	X
Filtrer par grade / poste	X	X	X	X	X
Filtrer par nom, prénom, département	X	X	X	X	X
Filtrer par matricule				X	X
Affecter employé aux projets		X	X		X
Ajouter un département					X
Lister départements	X	X	X	X	X
Affecter employé au département			X		X
Visualiser membres département	X	X	X	X	X
Créer / supprimer projet			X		X
Modifier projet		X	X		X
Affecter employés au		X	X		X

projet					
Suivre avancement projet	X	X	X		X
Créer fiche de paie employé pour un mois				X	X
Consulter sa propre fiche de paie	X	X	X	X	X
Consulter les fiches de paie des employés				X	X
Générer fiche de paie imprimable				X	X
Rechercher fiche de paie par période / par employé				X	X
Gérer les rôles des employés					X

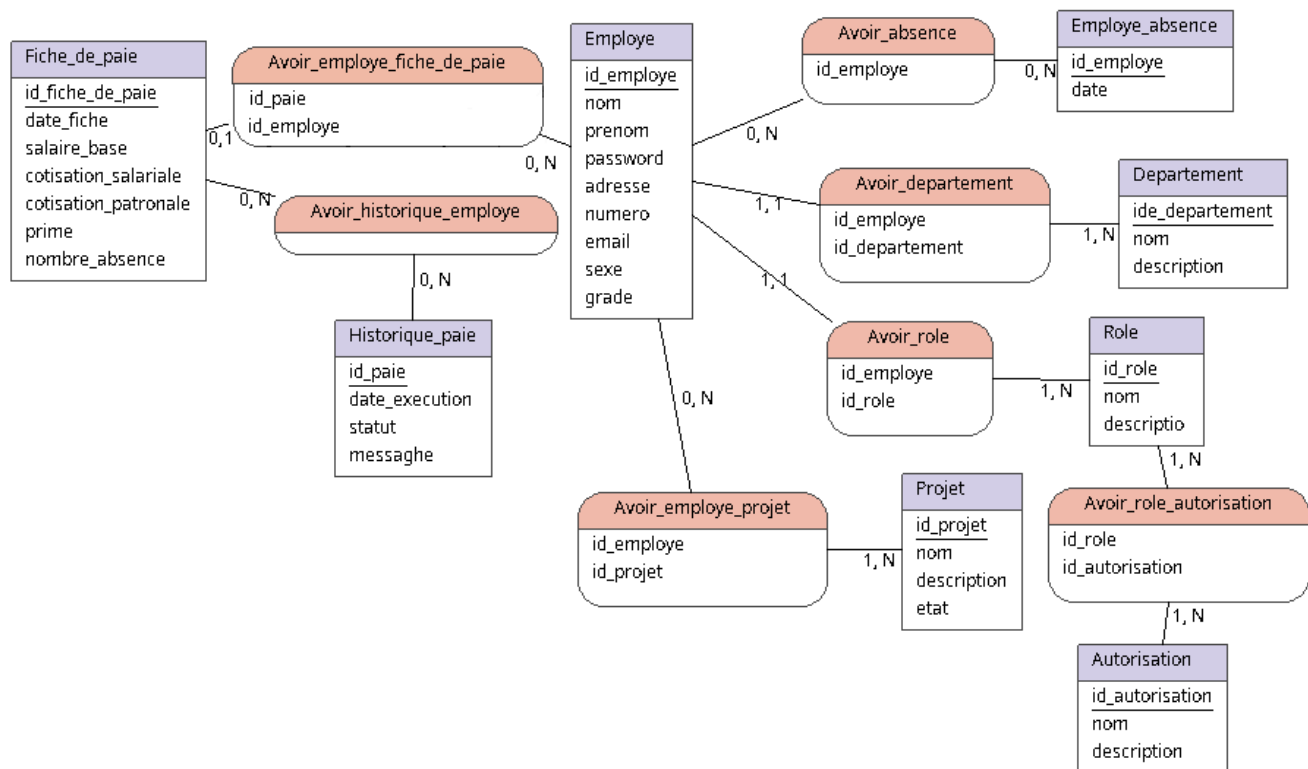
II - Architecture du projet

Notre application **Gestion Entreprise** est réalisée en JEE. Deux technologies alternatives seront proposées, avec Jakarta d'une part et Spring Boot d'autre part. Si l'architecture reste globalement la même, il existe quelques différences inhérentes aux technologies que nous présenterons également.

II.1 Le modèle de données

Afin de créer notre base de données, nous réalisons un schéma conceptuel (MCD).

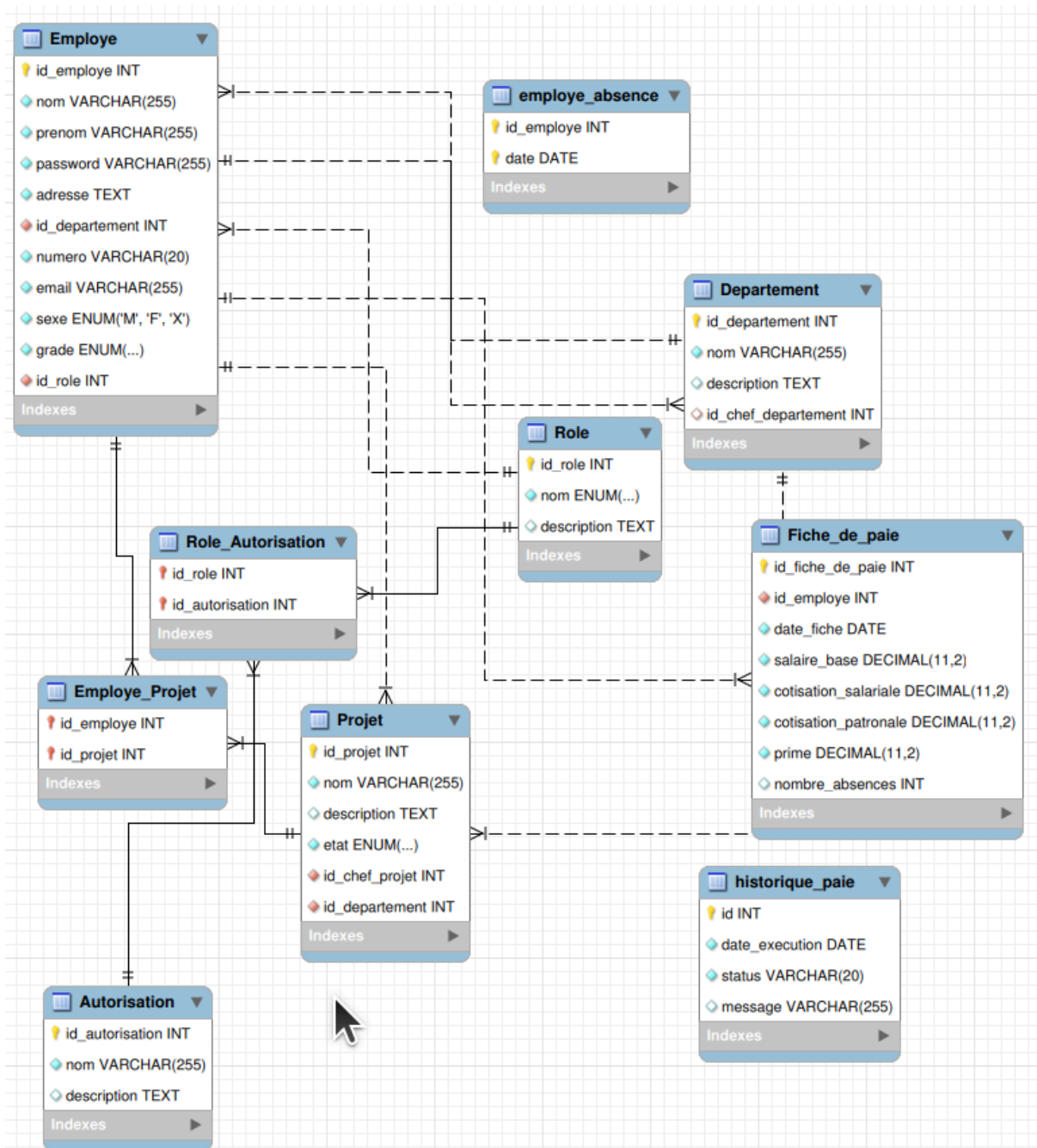
La traduction dans le modèle relationnel suivra le formalisme standard afin d'obtenir le schéma relationnel des tables.



II.2 L'implémentation de la base relationnelle

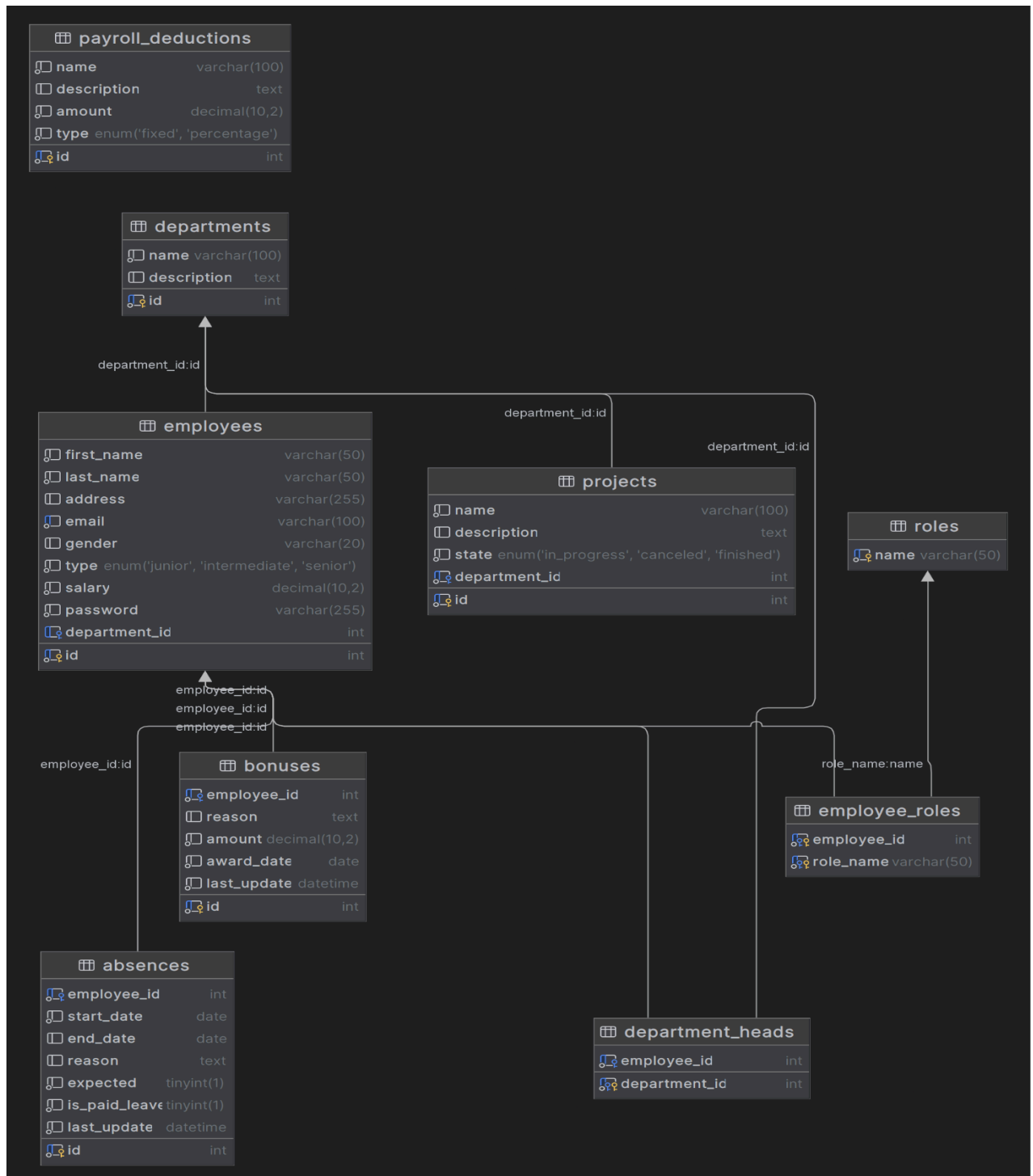
II.2.1 La base de données pour le projet avec Hibernate

Le modèle physique correspondant est le suivant :



II.2.2 La base de données pour le projet Spring boot

Le schéma est resté quasiment le même, à quelques détails près, comme la table des autorisations qui n'est pas utile dans Spring Boot.



II.3 L'architecture applicative

La structure générale de l'application respecte le principe de la séparation des responsabilités selon le modèle MVC :

- La couche Vue gère l'interface utilisateur grâce aux JSP
- Les Contrôleurs traitent les requêtes utilisateur et transmettent à la couche métier (Service), qui invoquent les DAO.
- La couche Modèle contient les entités, les opérations CRUD étant implémentées par les DAO via Hibernate (Session, Transaction). La couche DAO (Data Access Object) encapsule l'accès aux données via Hibernate qui s'occupe du mapping objet-relationnel et des opérations SQL. Hibernate gère la couche persistance en traduisant objets en requêtes SQL vers la base.

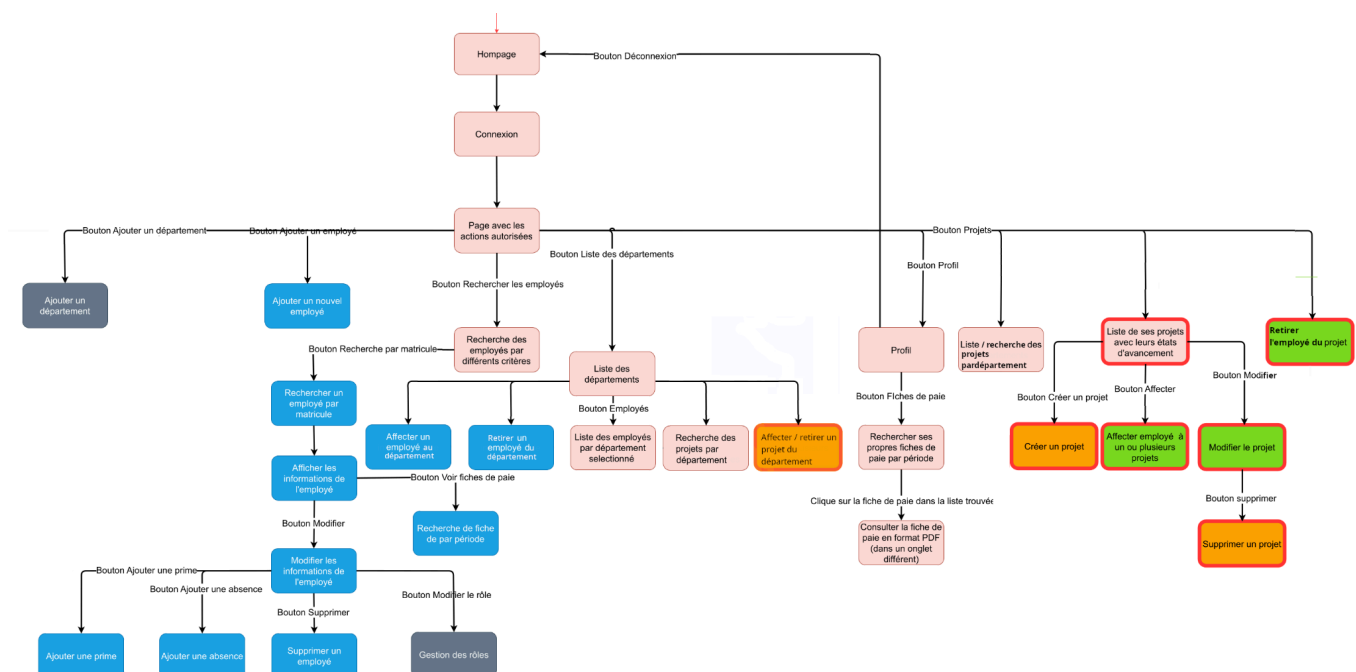
Ainsi nous retrouvons l'organisation **Vue** → **Contrôleur** → **Modèle**.

II.3.1 Version Hibernate

Les contrôleurs et les JSP sont organisés en sous-dossiers.

Sur le plan de la persistance, une session unique est partagée par toutes les pages. Les mots de passe sont chiffrés avec BCrypt, un employé pouvant modifier son mot de passe à tout moment.

Du point de vue ergonomique, la navigation a été pensée de façon à organiser logiquement les regroupements des actions. En voici le graphe.



Légende :

- Rose saumon : employé simple
- Vert : chef de projet
- Orange : chef de département
- Bleu : administrateur RH
- Gris : administrateur général

II.3.2 Version Spring Boot

Pour la version Spring Boot, quelques particularités sont ajoutées à l'architecture de base, notamment entre le contrôleur et le modèle sont intercalés le *service* et le *repository*.

L'architecture devient : **Vue** → **Contrôleur** → **Service** → **Repository** → **Modèle**.

Après passage par le contrôleur, le repository est une interface qui prend en paramètre l'entité et le type de sa clé primaire. Chaque entité a son repository (exemple : EmployeeRepository). Le repository permet de générer des méthodes d'accès à la BDD par réflexion. Par exemple, il suffit de définir une méthode "getEmployeesByld" pour que Spring, au moment de l'exécution, définit automatiquement cette méthode tant que son nom suit une convention spécifique. Le repository ne s'occupant que de l'accès direct à la BDD, le service intervient pour implémenter la logique métier.

Typiquement :

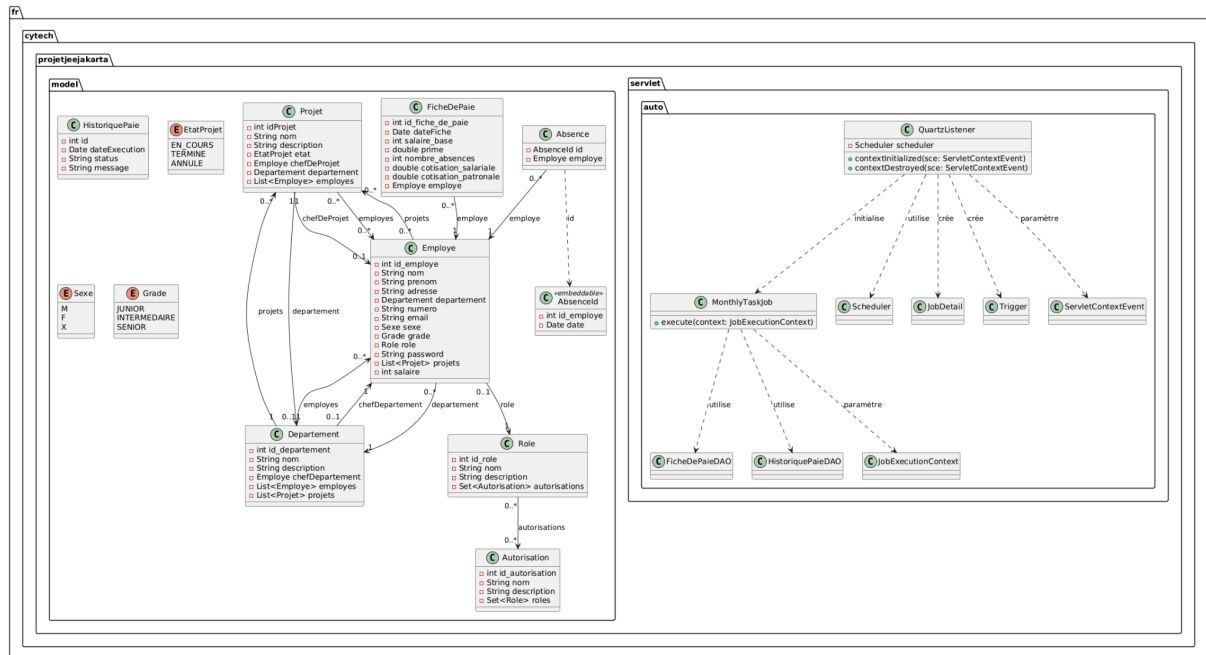
- Le controller traite la requête en communiquant avec le service et en transformant les requêtes HTTP en objets Java.
- Le service s'occupe de la logique métier abstraite, comme ajouter un utilisateur ou effectuer des recherches de BDD complexes
- Le repository accède à la BDD en générant des méthodes communiquant avec la BDD par la JPA.

D'autres particularités sont l'utilisation du système de rôles de Spring. Les utilisateurs sont implémentés comme des User Spring (modifiés) et les controllers sont gérés par des annotations, telles que `@PreAuthorize("hasAnyRole('HR', 'ADMIN')")`, par exemple. On utilise aussi un système de DTO (*Data Transfer Object*) pour pouvoir envoyer aux vues des projections de la BDD pour cacher et encapsuler la structure interne de la BDD.

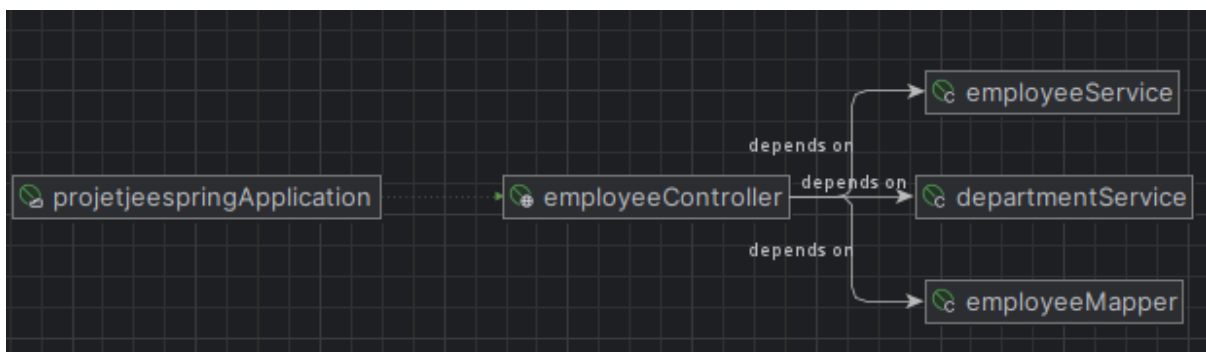
La BDD est également générée avec un système de migrations. Lors d'une première version, ces migrations ont permises de facilement mettre à jour la BDD. Plus tard, lors d'un gros refactoring, la BDD a été simplifiée à un seul fichier SQL.

II.3.3 Structure des classes

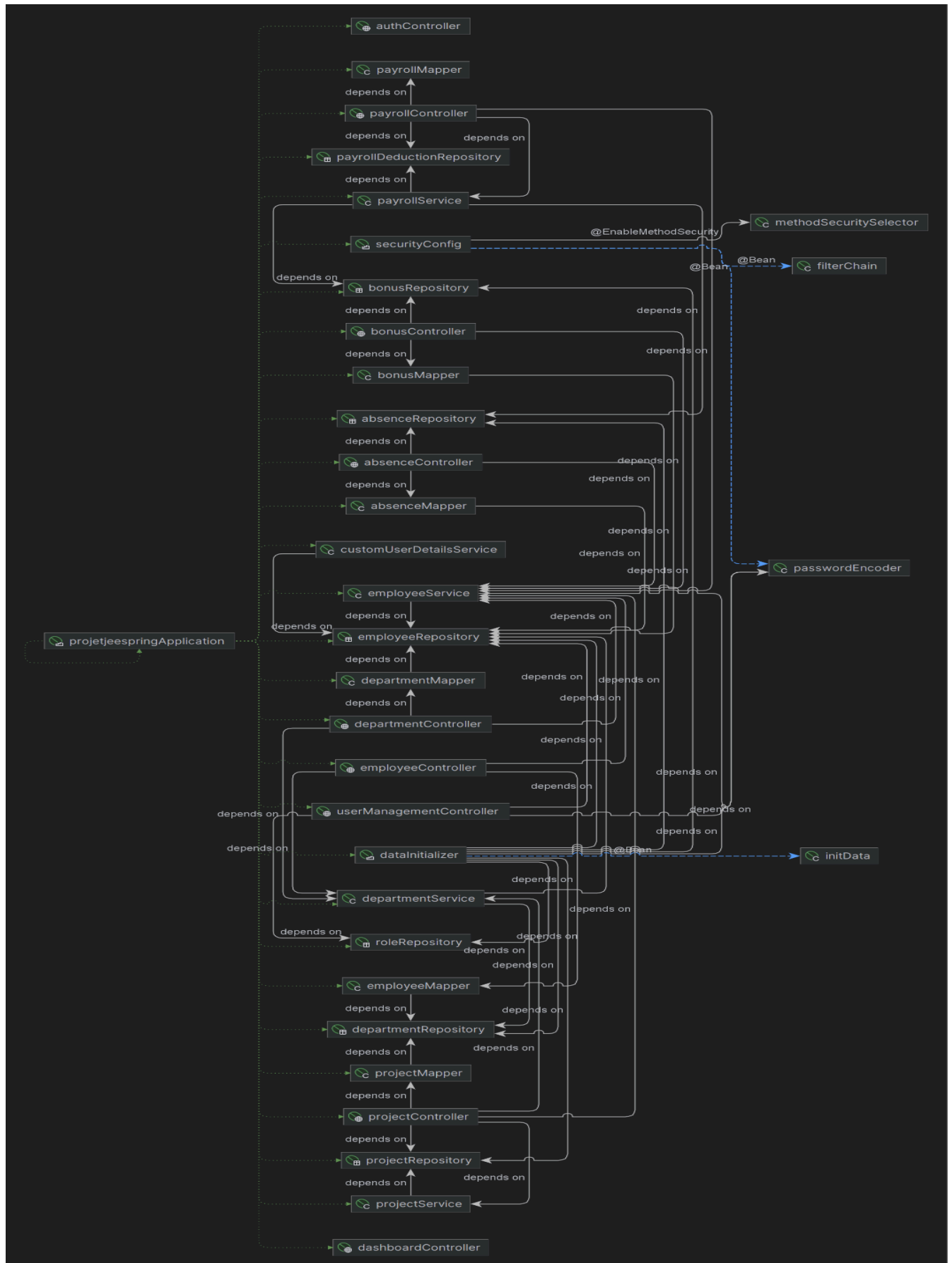
Les classes du projet Hibernate sont représentées dans ce diagramme



Ci-dessous un exemple de Bean Spring, le employeeController dépend du service pour les employés mais aussi du service pour les départements (le controlleur inclus la fonctionnalité de vérifier si un employé est un chef de département par exemple), mais aussi du bean employeeMapper (permet de passer d'une entité de la base de données à un DTO sans mot de passe (entre outre)).



La totalité des classes du projet Spring boot :

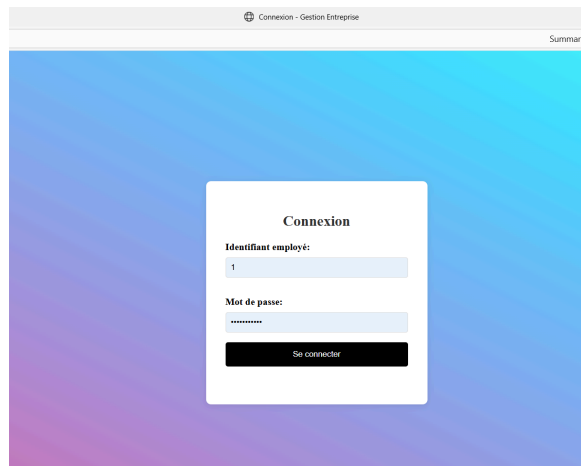


III - Présentation des écrans

III.1 Le projet Hibernate

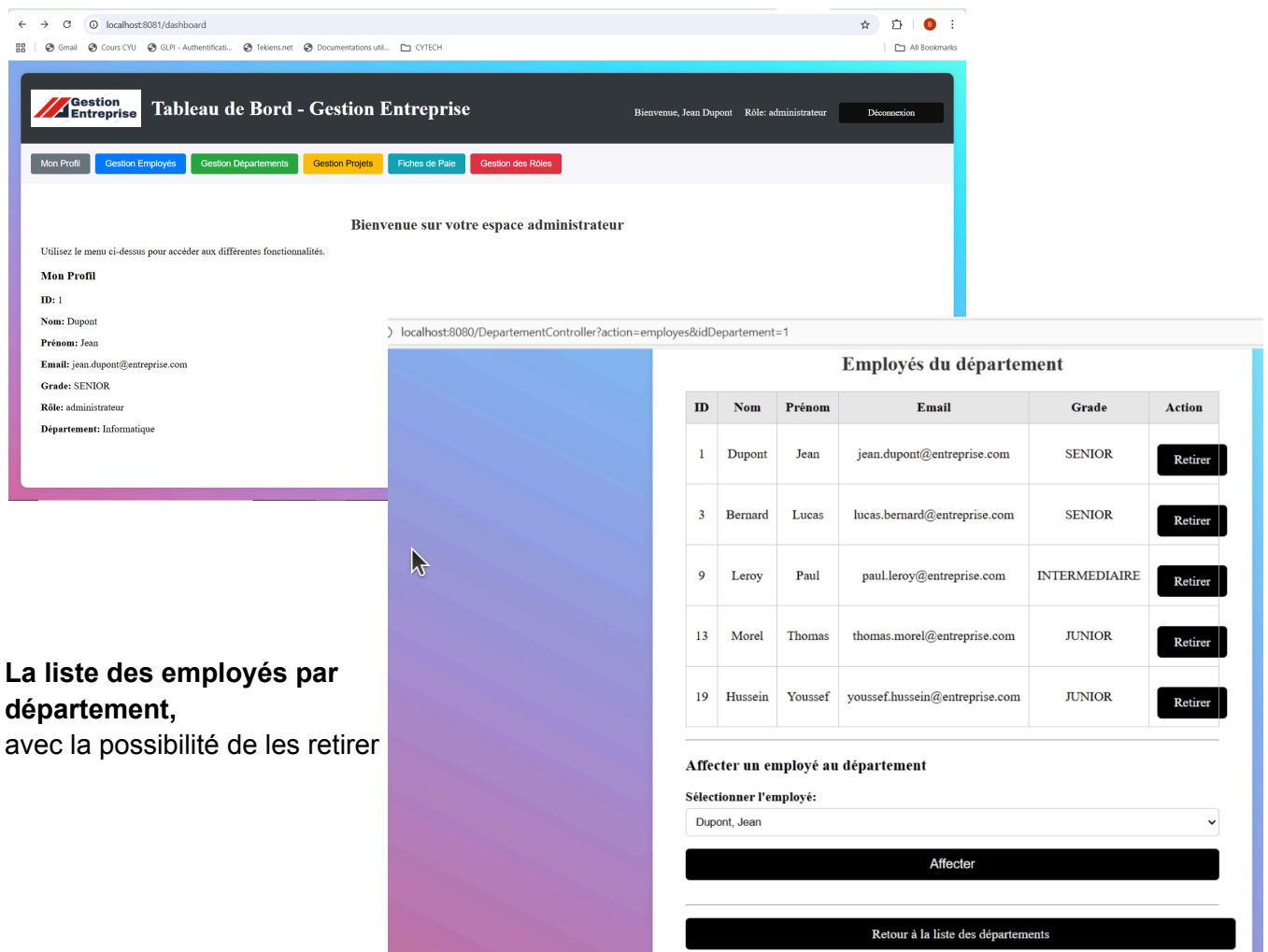
Voici quelques exemples d'écrans ci-dessous du projet Hibernate.

La connexion



L'accueil

C'est à partir du tableau de bord que l'on accède à la gestion des employés, départements, projets, des rôles, et aux fiches de paies (selon les droits associés au profil).



ID	Nom	Prénom	Email	Grade	Action
1	Dupont	Jean	jean.dupont@entreprise.com	SENIOR	<button>Retirer</button>
3	Bernard	Lucas	lucas.bernard@entreprise.com	SENIOR	<button>Retirer</button>
9	Leroy	Paul	paul.leroy@entreprise.com	INTERMEDIAIRE	<button>Retirer</button>
13	Morel	Thomas	thomas.morel@entreprise.com	JUNIOR	<button>Retirer</button>
19	Hussein	Youssef	youssef.hussein@entreprise.com	JUNIOR	<button>Retirer</button>

Affecter un employé au département

Sélectionner l'employé:

Dupont, Jean

Affecter

Retour à la liste des départements

La liste des employés par département, avec la possibilité de les retirer

La liste des employés par projet, avec la possibilité de les retirer, ou d'en affecter un nouveau.

localhost:8080/ProjetController?action=employees&idProjet=2

Liste des employés affectés au projet Refonte Site Web

ID	Nom	Prénom	Email	Grade	Action
3	Bernard	Lucas	lucas.bernard@entreprise.com	SENIOR	Détacher
9	Leroy	Paul	paul.leroy@entreprise.com	INTERMEDIAIRE	Détacher
14	Fabre	Chloé	chloe.fabre@entreprise.com	JUNIOR	Détacher
20	Barbier	Lucie	lucie.barbier@entreprise.com	JUNIOR	Détacher

Affecter un employé du département au projet

Sélectionner l'employé :

Dupont, Jean

Affecter

Retour à la liste des projets

Les projets d'un département avec la possibilité d'associer un nouveau projet au département.

localhost:8080/DepartementController?action=projects&idDepartement=1

Projets du département

ID	Nom	Description	État	Chef de projet	Action
2	Refonte Site Web	Nouvelle version du site corporate	EN_COURS	Leroy	Retirer
19	Game Jam	Compétition de développement de jeu vidéo	EN_COURS	Leroy	Retirer

Affecter un projet au département

Sélectionner le projet :

Refonte Site Web

Affecter

Retour à la liste des départements

La fiche de paie (pdf généré)



Employeur : FoodNCo Adresse : 55 Rue du Faubourg Saint-Honoré, 75008 Paris	Période de paie : 2025-10-01 2025-10-31
---	---

Nom :	Dupont
Prénom :	Jeanne
Adresse :	1 Rue de la bretonnière, Paris
Grade :	SENIOR
Poste :	administrateur

Cotisation / Élément	Base (€)	Taux (%)	Montant (€)
Salaire Brut	6000.00	-	6000.00
Prime	0.00	-	0.00
Sécurité Sociale (URSSAF)	6000.00	0.10	-600.00
CSG / CRDS	6000.00	0.18	-1080.00
Assurance Chômage	6000.00	4.05	-243.00
Absence (0 jours)	-	-	-0.00
NET IMPOSABLE			

NET À PAYER : 4077.00 €

Signature de l'employeur

III.2 Le projet Spring Boot

Voici quelques écrans Spring Boot
Panneau principal

Welcome, Alice!

Logout

Logged in as: 10000 | Roles: ADMIN

My Space

- My Profile
- My Absence History
- My Bonuses
- My Team's Projects
- Generate My Payslip
- Change Password

Company Directory

- Employee Directory
- Departments List
- Projects Overview

Management

- My Department Details
- Manage Projects

Human Resources

- Register New Employee
- Manage Absences
- Manage Bonuses
- Run Payroll

Administration

- Manage Users & Roles
- Create Department
- Delete/Edit Employees

Saisie d'une absence pour un employé

[Dashboard](#) | [Back to Absences](#)

Report Absence

Employee:

Alice Admin

Start Date:

01 / 02 / 2000

End Date (Empty if ongoing):

01 / 03 / 2000

Reason:

Combustion spontanée

Details:

☒ Expected?
☐ Paid Leave?

Save Absence

Cancel

Génération d'une fiche de paie (avec l'absence saisie)

[Dashboard](#)

Generate My Payslip

Select Period

From:

01 / 01 / 2000

 To:

mm / dd / yyyy

Generate

Fiche de paie générée avec l'absence saisie

Print / PDF [Exit](#)

PAYSLIP	
Employee: Alice Admin	ID: 10000
Period: 2000-01-01 to 2000-01-20	Days Covered: 20
EARNINGS	
Base Salary	EUR58,064.52
Unpaid Absences (2 days)	- EUR5,806.45
GROSS SALARY (After Absences)	EUR52,258.07
Bonuses:	
2000-01-05: M'a offert un café (2.00 EUR)	
Total Bonuses	EUR2.00
DEDUCTIONS	
Impots dans 50 ans	- EUR47,034.06
Total Deductions	- EUR47,034.06
NET PAYABLE	EUR5,226.01

Liste des projets d'un chef de projet

[Dashboard](#) | [All Projects](#) | [My Department's Projects](#)

All Projects Directory

[+ Create New Project](#)

Project Name	State	Department	Description	Actions
Migration to Java 21	IN_PROGRESS	IT Department	Updating legacy systems	Edit Delete
Internal Wiki	FINISHED	IT Department	Documentation platform	Edit Delete
AI Overlord	CANCELED	IT Department	Skynet implementation	Edit Delete
Recruitment Drive 2025	IN_PROGRESS	Human Resources	Hiring new grads	Edit Delete
Q4 Sales Push	IN_PROGRESS	Sales	Call everyone	Edit Delete
Rebranding	FINISHED	Marketing	New Logo	Edit Delete

IV - Organisation du travail

Le travail a été réparti et les tâches ont été réalisées comme indiqué dans le diagramme ci-dessous :

