



**INSTITUT
POLYTECHNIQUE**
PRIVÉ DE CASABLANCA

« Développement d'une application web de gestion des ressources humaines »

Réalisé par :

Wijdane Azaroua

Sous l'encadrement de :

Tarik Fassali

2023/2024

Remerciements

Je tiens à exprimer ma profonde gratitude à M. Tarik, mon professeur de Java, pour son soutien et ses conseils tout au long de ce projet. Son expertise, sa patience et sa disponibilité ont été inestimables pour la réussite de ce travail. Grâce à son enseignement et à ses encouragements, j'ai pu surmonter les défis techniques et développer une compréhension approfondie des concepts avancés de Java et des frameworks associés.

M. Tarik, merci pour votre dévouement et votre engagement envers l'enseignement. Votre passion pour le développement logiciel et votre volonté de partager vos connaissances ont grandement contribué à mon apprentissage et à la réalisation de ce projet. Vous avez été une source d'inspiration et de motivation, et je suis reconnaissant pour toutes les opportunités de croissance personnelle et professionnelle que vous m'avez offertes.

Je souhaite également remercier tous mes camarades de classe et les membres du corps professoral qui ont apporté leur soutien et leurs encouragements tout au long de cette aventure académique.

Merci encore, M. Tarik, pour votre précieuse aide et votre mentorat.

Table des Matières

Remerciements

Table des Matières

1. Introduction Générale

- Présentation du projet
- Objectifs du projet
- Contexte et justification
- Fonctionnalités et Rôles
- Gestion des Modules

2. Étude de l'État de l'Art

- Analyse des solutions existantes
- Technologies utilisées
- Java EE
- Spring Boot
- React
- MySQL
- Avantages et inconvénients des différentes approches

3. Analyse des Besoins

- Identification des besoins
- Description des fonctionnalités
- Diagrammes de cas d'utilisation (UML)

4. Conception du Système

- Méthodologie MERISE
- Modèle Physique de Données (MPD)
- Diagrammes UML
- Diagrammes de classes
- 5. Développement du Système
- Configuration du projet Spring Boot
- Structure du projet
- Dépendances Maven
- Implémentation des entités
- Implémentation des services
- Implémentation des contrôleurs
- Configuration de la sécurité (Spring Security)
- Gestion des rôles et des permissions
- Authentification et autorisation
- 6. Interface Utilisateur (Frontend)
- Choix technologiques (React)
- Structure du frontend
- Intégration avec le backend
- 7. Tests et Validation
- Stratégie de test
- Tests unitaires
- Tests d'intégration
- Tests de performance

8. Conclusion générale

1. Introduction Générale

Dans le cadre de l'évolution constante des technologies et de la digitalisation des processus organisationnels, le projet de gestion des ressources humaines que j'ai entrepris vise à répondre aux besoins modernes des entreprises en matière de gestion du personnel. Ce projet, développé avec Java EE, Spring Boot pour le backend et React pour le frontend, s'appuie sur une base de données MySQL pour la gestion des données.

a) Présentation du Projet

Le système de gestion des ressources humaines que j'ai conçu est destiné à simplifier et à optimiser la gestion des employés, des congés, des départements, des postes et des candidats. Il offre une plateforme intégrée permettant à différents utilisateurs de gérer efficacement les informations et les processus relatifs aux ressources humaines.

b) Objectifs du Projet

L'objectif principal de ce projet est de fournir une solution complète et intuitive pour la gestion des ressources humaines, adaptée aux besoins spécifiques des entreprises modernes. Le système permet de centraliser toutes les informations relatives aux employés et aux candidats, facilitant ainsi la prise de décision et la gestion quotidienne des ressources humaines.

c) Contexte et Justification

Dans un contexte où la gestion efficace des ressources humaines est cruciale pour la performance et la croissance des entreprises, il est essentiel de disposer d'outils performants pour gérer les diverses tâches associées à cette fonction. Notre projet répond à ce besoin en offrant une solution robuste et flexible qui prend en compte les différentes facettes de la gestion des ressources humaines.

d) Fonctionnalités et Rôles

Le système propose quatre rôles principaux, chacun ayant des fonctionnalités spécifiques :

1. Admin :

- Gestion globale du système
- Création et gestion des utilisateurs
- Supervision des opérations des autres rôles

2. Employé :

- Consultation de son profil et de ses informations personnelles
- Gestion de ses demandes de congés
- Accès aux informations sur les postes disponibles et les départements.

3. Employé RH :

- Gestion des employés (ajout, modification, suppression)
- Gestion des demandes de congés
- Gestion des départements et des postes
- Suivi et gestion des candidats

4. Candidat :

- Consultation des offres d'emploi
- Soumission de candidatures

e) Gestion des Modules

Le système est structuré autour de plusieurs modules clés :

1. Gestion des Congés :

- Demande, approbation et suivi des congés des employés

2. Gestion des Employés :

- Suivi des informations personnelles et professionnelles des employés

3. Gestion des Départements et des Postes :

- Organisation et structuration des départements et des postes au sein de l'entreprise

4. Gestion des Candidats :

- Suivi des candidatures et des processus de recrutement

5. Offres d'Emploi :

- Publication et gestion des offres d'emploi
- Permettre aux candidats de trouver et de postuler à des offres

Cette solution intégrée offre une interface conviviale et des fonctionnalités adaptées aux besoins spécifiques des utilisateurs, facilitant ainsi la gestion des ressources humaines et améliorant l'efficacité opérationnelle.

2. Étude de l'État de l'Art

a) Analyse des Solutions Existantes

Dans le domaine de la gestion des ressources humaines, plusieurs solutions logicielles sont actuellement disponibles sur le marché. Ces solutions varient en termes de fonctionnalités, de technologies utilisées et de niveaux de personnalisation offerts. Parmi les solutions populaires, on trouve :

1. SAP SuccessFactors : SAP SuccessFactors est une solution cloud complète qui offre une large gamme de fonctionnalités pour la gestion des talents, la gestion des performances, et l'analyse RH. Elle permet aux entreprises de centraliser et de standardiser leurs processus RH à travers une interface utilisateur intuitive. Cependant, cette solution peut être coûteuse et complexe à mettre en œuvre, surtout pour les petites et moyennes entreprises, en raison de son niveau de personnalisation et de sa vaste gamme de fonctionnalités.

2. Workday : Workday propose une suite intégrée de gestion des ressources humaines, de gestion des talents, de paie et de finances. C'est une solution appréciée pour sa convivialité et ses capacités analytiques avancées. Workday est souvent choisie par les grandes entreprises en raison de son coût et de sa complexité, mais elle offre une flexibilité considérable pour s'adapter aux besoins spécifiques des grandes organisations.

3. BambooHR : BambooHR est une solution plus adaptée aux petites et moyennes entreprises. Elle offre une gestion simplifiée des dossiers des employés, des congés et des performances. BambooHR est reconnue pour sa facilité d'utilisation et son interface conviviale. Bien qu'elle soit plus accessible financièrement, elle peut manquer de certaines fonctionnalités avancées nécessaires aux grandes entreprises ou aux organisations ayant des besoins spécifiques.

b) Technologies Utilisées

Pour le développement de notre projet de gestion des ressources humaines, nous avons choisi d'utiliser les technologies suivantes :

- Java EE
- Spring Boot
- React
- MySQL

c) Java EE

Java Enterprise Edition (Java EE) est une plateforme robuste et mature pour le développement d'applications d'entreprise. Elle offre un ensemble de spécifications pour le développement d'applications distribuées, évolutives et sécurisées. Les applications Java EE sont connues pour leur capacité à gérer des transactions complexes et à fournir des services fiables et évolutifs.

- Avantages :

- Grande communauté et support étendu
- Normes industrielles et interopérabilité
- Robustesse et évolutivité

- Inconvénients :

- Courbe d'apprentissage élevée
- Configuration et déploiement peuvent être complexes

d) Spring Boot

Spring Boot est un framework basé sur Spring qui simplifie le développement d'applications Java en fournissant des configurations par défaut et des fonctionnalités prêtes à l'emploi. Spring Boot permet

de créer des applications autonomes, prêtes pour la production, avec des dépendances minimales et une configuration simplifiée.

- Avantages :

- Configuration simplifiée et rapide
- Intégration facile avec Spring et autres technologies
- Prêt pour la production avec des outils de monitoring et de gestion

- Inconvénients :

- Peut être surdimensionné pour des applications très simples
- Certaines configurations avancées peuvent encore nécessiter des connaissances approfondies

e) React

React est une bibliothèque JavaScript développée par Facebook pour la construction d'interfaces utilisateur. Elle permet de créer des composants réutilisables et offre une performance élevée grâce à son modèle de rendu virtuel. React est largement adopté pour les applications web modernes en raison de sa flexibilité et de sa capacité à gérer des interfaces utilisateur dynamiques et réactives.

- Avantages :

- Performance et rapidité
- Réutilisabilité des composants
- Grande communauté et écosystème riche

- Inconvénients :

- Nécessite une courbe d'apprentissage pour les développeurs non familiers avec JavaScript moderne
- La gestion de l'état peut devenir complexe pour des applications très grandes

f) MySQL

MySQL est un système de gestion de base de données relationnelle open source très populaire. Il est connu pour sa performance, sa fiabilité et sa facilité d'utilisation. MySQL est largement utilisé dans les applications web pour sa capacité à gérer des volumes importants de données et sa compatibilité avec de nombreuses plateformes.

- Avantages :

- Gratuit et open source
- Performance et fiabilité
- Large adoption et support communautaire

- Inconvénients :

- Certaines fonctionnalités avancées disponibles dans d'autres SGBD commerciaux peuvent manquer
- Performance peut être limitée pour des applications très volumineuses sans configuration adéquate

g) Avantages et Inconvénients des Différentes Approches

Les différentes technologies et solutions analysées présentent des avantages et des inconvénients spécifiques. Pour notre projet, le choix de Java EE, Spring Boot, React et MySQL a été motivé par les besoins de robustesse, de flexibilité, et de performance de l'application.

- Java EE et Spring Boot : Ces technologies offrent une base solide pour le développement de backends sécurisés et évolutifs. Spring Boot en particulier simplifie grandement le processus de développement et de déploiement. Java EE fournit une plateforme standardisée et éprouvée pour le développement d'applications d'entreprise, tandis que Spring Boot offre une approche plus légère et rapide pour démarrer les projets.

- React : Pour le frontend, React permet de créer une interface utilisateur moderne, réactive et performante. Sa popularité et son large écosystème de bibliothèques complémentaires en font un choix judicieux pour une application web dynamique. La réutilisabilité des composants React permet également de maintenir une base de code propre et modulaire.

- MySQL : En tant que base de données, MySQL assure une gestion efficace des données tout en étant facilement intégrable avec le reste de la stack technologique choisie. MySQL est capable de gérer des transactions complexes et offre une performance fiable pour les applications de taille moyenne à grande.

3. Analyse des Besoins

a) Identification des Besoins

L'analyse des besoins est une étape cruciale dans le développement de notre projet de gestion des ressources humaines. Cette phase permet de comprendre et de documenter les attentes des utilisateurs finaux et les exigences fonctionnelles du système. Pour ce projet, les besoins ont été identifiés à travers des entretiens avec les parties prenantes, l'analyse des processus existants, et des études de marché.

- **Besoins généraux**
 - Gestion centralisée des informations : Centraliser toutes les informations relatives aux employés, aux départements, aux postes, et aux candidats dans un système unique.
 - Sécurité : Assurer la confidentialité et la sécurité des données sensibles relatives aux employés et aux candidats.

- **Besoins spécifiques par rôle**

1. Admin

- Gestion des utilisateurs et des rôles
- Supervision des activités et des processus
- Génération de rapports et d'analyses

2. Employé

- Accès à ses informations personnelles et professionnelles
- Soumission et suivi des demandes de congés
- Consultation des offres d'emploi internes

3. Employé RH

- Gestion des dossiers des employés (ajout, modification, suppression)
- Gestion des demandes de congés

- Gestion des départements et des postes
- Suivi et gestion des candidatures

4. Candidat

- Consultation des offres d'emploi
- Soumission de candidatures
- Suivi de l'état de ses candidatures

b) Description des Fonctionnalités

Pour répondre aux besoins identifiés, le système de gestion des ressources humaines offrira les fonctionnalités suivantes :

1. Gestion des Utilisateurs

- Création et gestion des profils utilisateurs
- Attribution de rôles et de permissions

2. Gestion des Employés

- Ajout, modification, et suppression des informations des employés
- Suivi des performances et des évaluations

3. Gestion des Congés

- Soumission, approbation, et suivi des demandes de congés
- Gestion des quotas de congés
- Historique des congés pris

4. Gestion des Départements et des Postes

- Création et modification des départements
- Définition et gestion des postes
- Association des employés aux départements et aux postes

5. Gestion des Candidatures

- Publication des offres d'emploi
- Réception et suivi des candidatures
- Communication avec les candidats

6. Interface Utilisateur

- Tableau de bord personnalisé pour chaque rôle
- Interface réactive et intuitive pour la gestion des tâches quotidiennes

7. Sécurité et Confidentialité

- Authentification et autorisation sécurisées
- Chiffrement des données sensibles
- Suivi et audit des activités

c) Diagrammes de Cas d'Utilisation (UML)

Les diagrammes de cas d'utilisation permettent de représenter les interactions entre les utilisateurs (acteurs) et le système, ainsi que les différentes fonctionnalités offertes. Voici un exemple de diagramme de cas d'utilisation pour notre projet.

- Diagramme de Cas d'Utilisation Global

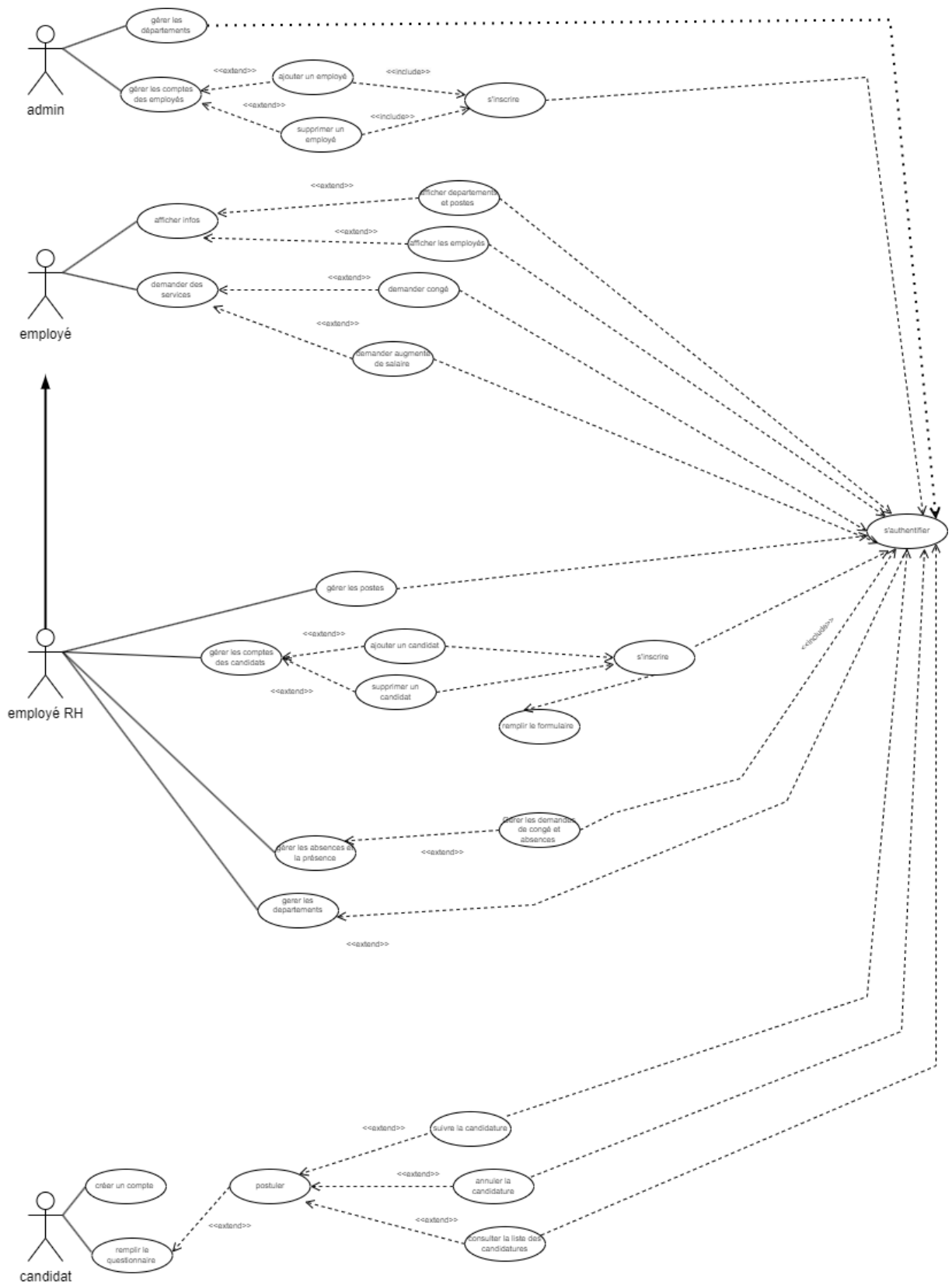


Diagramme de cas d'utilisation

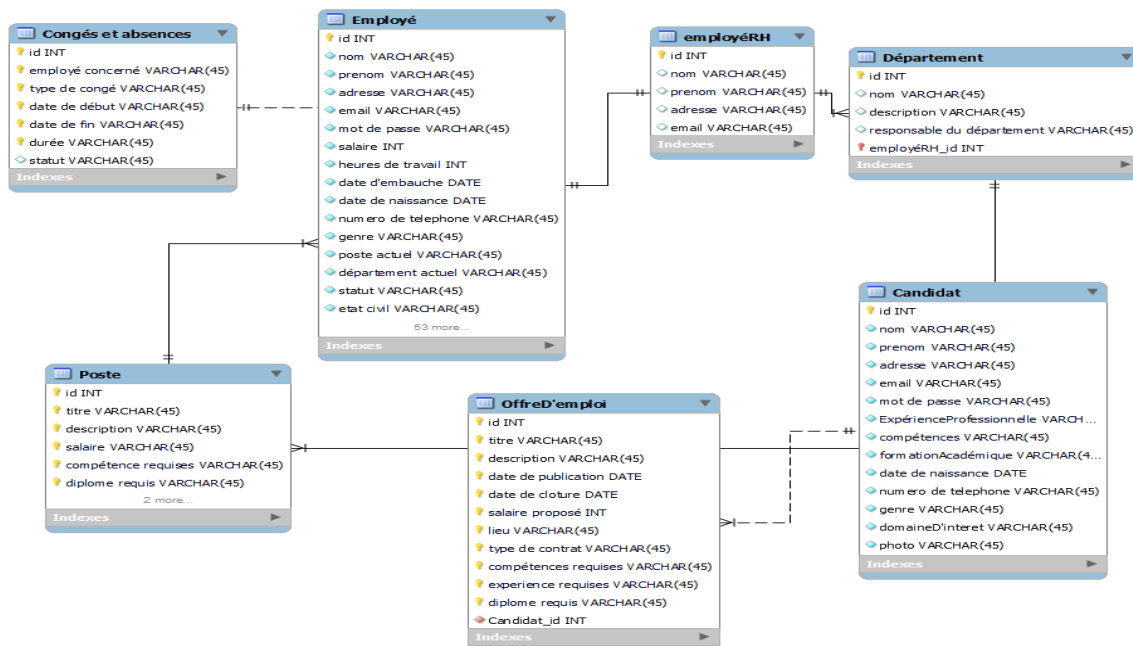
4. Conception du Système

a) Méthodologie MERISE

La méthode MERISE est une méthodologie de conception et de gestion de projet informatique qui se concentre sur la modélisation des données et des traitements. Le Modèle Physique de Données (MPD) est une étape clé de cette méthodologie.

b) Modèle Physique de Données (MPD)

Le MPD est une représentation détaillée de la structure des données dans une base de données. Il prend en compte les contraintes techniques et les performances du système de gestion de bases de données (SGBD) utilisé.



c) Diagrammes UML

Les diagrammes UML (Unified Modeling Language) sont des outils de modélisation utilisés pour la conception de systèmes orientés objet. Ils permettent de représenter visuellement les différents aspects du système.

- Diagrammes de classes

Le diagramme de classes est un type de diagramme UML qui montre la structure statique d'un système en représentant ses classes, leurs attributs, méthodes, et les relations entre les classes.

- Éléments d'un diagramme de classes :

1. Classes : Représentées par des rectangles divisés en trois parties :

- Le nom de la classe
- Les attributs de la classe
- Les méthodes de la classe

2. Relations :

- Associations : Lien entre deux classes qui interagissent entre elles.
- Héritage (généralisation/spécialisation) : Représente une relation hiérarchique où une classe dérivée hérite des attributs et méthodes d'une classe de base.
- Agrégation : Relation de type "partie de", où une classe est une partie d'une autre.
- Composition : Type particulier d'agrégation où la partie ne peut exister sans le tout.
- Dépendances : Représentent une relation de dépendance où une classe utilise une autre.

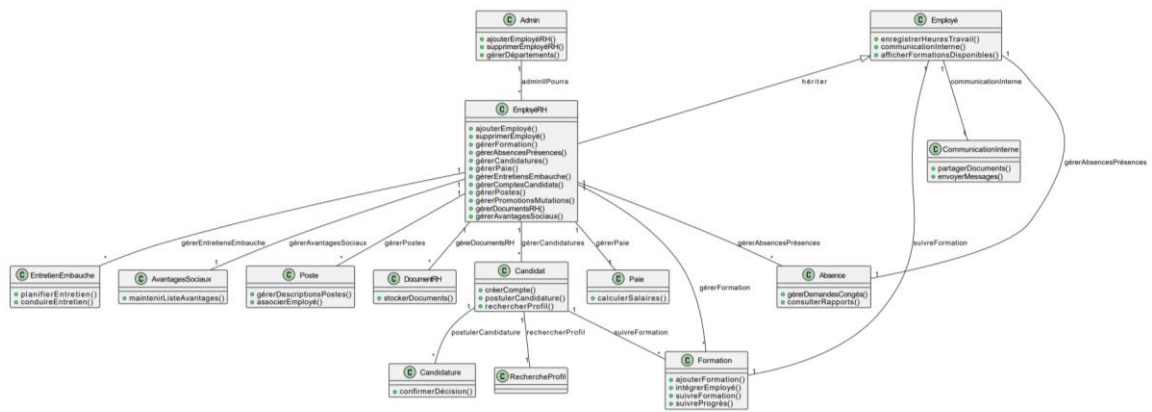


Diagramme de classe

5. Développement du Système

a) Configuration du Projet Spring Boot

- Structure du Projet

La structure d'un projet Spring Boot suit une convention bien définie pour organiser les fichiers et les répertoires. Voici une description de la structure typique d'un projet Spring Boot :

- src/main/java : Contient le code source de l'application.
- com.example.project : Package racine pour l'application.
- Application.java : Classe principale qui contient la méthode `main()` pour démarrer l'application.
- controller : Contient les classes contrôleurs Spring MVC.
- Service : Contient les classes de services qui implémentent la logique métier.
- repository : Contient les interfaces pour l'accès aux données.
- entité : Contient les classes entités JPA.
- application.properties: Fichier de configuration principal pour l'application.
- pom.xml : Fichier de configuration Maven qui spécifie les dépendances et les plugins utilisés par le projet.

b) Dépendances Maven

Les dépendances Maven sont des bibliothèques externes que le projet utilise. Le fichier `pom.xml` permet de gérer ces dépendances. Voici quelques dépendances typiques pour un projet Spring Boot :

- spring-boot-starter-web : Fournit les fonctionnalités nécessaires pour développer des applications web, y compris Spring MVC.
- spring-boot-starter-data-jpa : Fournit les fonctionnalités pour utiliser JPA avec Spring Data.
- spring-boot-starter-security : Fournit les fonctionnalités pour sécuriser l'application avec Spring Security.

c) Implémentation des Entités

Les entités représentent les objets persistants dans la base de données. Chaque entité est généralement une classe annotée avec `@Entity`, et elle mappe une table de la base de données.

- Définition des attributs : Les attributs de la classe représentent les colonnes de la table.
- Annotations JPA : Utilisation d'annotations comme `@Id`, `@GeneratedValue`, `@Column`, `@ManyToOne`, etc., pour définir les relations et les caractéristiques des colonnes.

d) Implémentation des Services

Les services contiennent la logique métier de l'application. Ils sont généralement annotés avec `@Service`.

- Interfaces et Implémentations : Définition d'interfaces pour les services et implémentation de ces interfaces.
- Utilisation des Répositories : Les services interagissent avec les repositories pour effectuer des opérations CRUD sur les entités.
- **Gestion des Transactions** : Utilisation de l'annotation `@Transactional` pour gérer les transactions.

e) Implémentation des Contrôleurs

Les contrôleurs gèrent les requêtes HTTP et sont annotés avec `@RestController` ou `@Controller`.

- Mapping des URLs : Utilisation des annotations `@RequestMapping`, `@GetMapping`, `@PostMapping`, etc., pour définir les endpoints.
- Injection des Services : Les contrôleurs utilisent les services pour accéder à la logique métier.
- Gestion des Réponses : Utilisation des classes comme `ResponseEntity` pour gérer les réponses HTTP.

f) Configuration de la Sécurité (Spring Security)

- Gestion des Rôles et des Permissions

Spring Security permet de définir des rôles et des permissions pour sécuriser l'application.

- Définition des Rôles : Création de rôles comme ADMIN, USER, etc.
- Configuration des Permissions : Assignment des permissions aux rôles pour restreindre l'accès aux ressources.

g) Authentification et Autorisation

Spring Security gère l'authentification et l'autorisation des utilisateurs.

- Configuration de l'Authentification : Utilisation d'un fournisseur d'authentification pour vérifier les identifiants des utilisateurs.
- Configuration de l'Autorisation : Définition des règles d'accès aux ressources en fonction des rôles et des permissions.
- Filtres de Sécurité : Implémentation de filtres pour intercepter et traiter les requêtes HTTP.

6. Interface Utilisateur (Frontend)

a) Choix Technologiques

Pour la partie frontend de notre application, nous avons choisi d'utiliser React, une bibliothèque JavaScript développée par Facebook, qui est largement adoptée pour la création d'interfaces utilisateur dynamiques et réactives. Les avantages de React incluent :

- Composants Réutilisables : La possibilité de créer des composants réutilisables facilite la maintenance et l'évolutivité de l'application.
- Virtual DOM : Permet des mises à jour rapides et efficaces de l'interface utilisateur.
- Communauté Active et Écosystème : Une vaste communauté de développeurs et un écosystème riche en bibliothèques et outils complémentaires.
- Interopérabilité : Facilité d'intégration avec d'autres bibliothèques et frameworks JavaScript.

b) Structure du Frontend

La structure d'un projet React suit généralement une convention pour organiser les fichiers et les répertoires. Voici une description de la structure typique :

- src : Contient tout le code source de l'application.
- components : Contient les composants réutilisables.
- pages : Contient les composants de pages qui représentent les différentes vues de l'application.
- services : Contient les services pour interagir avec le backend, par exemple, pour les appels API.
- assets : Contient les fichiers statiques comme les images, les fichiers CSS, etc.

- App.js : Le composant racine de l'application.
- index.js : Le point d'entrée principal qui rend le composant `App` .
- public : Contient les fichiers publics accessibles directement par le navigateur.
- index.html : Le fichier HTML principal où l'application React est montée.

c) Intégration avec le Backend

L'intégration du frontend avec le backend se fait principalement via des appels API. Voici les étapes générales pour cette intégration :

1. Configuration des Services API : Création de services dans le dossier `services/` pour gérer les appels HTTP.
2. Utilisation des Services dans les Composants : Les composants utilisent ces services pour récupérer et manipuler les données.
3. Gestion de l'Authentification et de l'Autorisation : Intégration avec le backend pour gérer les utilisateurs et les rôles. Utilisation de bibliothèques comme `axios` pour ajouter les en-têtes d'autorisation aux requêtes.

d) Interface



Page d'accueil

"Le cœur de notre entreprise : l'excellence !"

Ceci est un site web pour les employés et les candidats aussi



Offres d'emploi



Chef de Production
Prêt à optimiser les processus de production et à garantir la qualité ?
[En savoir plus](#)



Directeur des Ventes
Êtes-vous prêt à booster les ventes et à atteindre des objectifs ambitieux ?
[En savoir plus](#)



Technicien de Laboratoire
Vous avez des compétences en laboratoire et cherchez un nouveau défi ?
[En savoir plus](#)



Directeur Financier
Passionné par la gestion financière et stratégique ? Découvrez nos postes en finance.
[En savoir plus](#)

Page d'accueil (suite)

Wijdane Company

Login

Login

Email:

Password:

Login

login

Wijdane Company Mon profil Employés Départements Postes Candidats Congés



Informations
Personnelles

Services

Messages internes

Profile Information

Nom: Azaroua

Prénom: Wijdane

Email: wijdane@gmail.com

Numéro de téléphone: 0661548156

Infos personnelles

Liste des Employés

Add Employee

Id	Nom	Prénom	Email	Adresse	Numéro de téléphone	Salaire	Département	Date d'embauche	Poste	Status	Actions
9	Elmouahhid	Khadija	khadija@gmail.com	benslimane	0612457812	61000	Production	2024-06-23	Chef de Production	ACTIVE	<div>UpdateDelete</div>
14	Akhdari	Siham	siham@gmail.com	derb seltan	0912345678	80000	Comptabilité et Finance	2024-06-04	Directeur Financier	ACTIVE	<div>UpdateDelete</div>
12	Labihi	Nohayla	nohayla@gmail.com	sidi maarouf	0612457812	50000	Ressources humaines	2024-06-13	Spécialiste en Recrutement	ACTIVE	<div>UpdateDelete</div>
13	El aoud	Abdghani	abdghani@gmail.com	sbata	0645784578	50000	Ressources humaines	2024-06-02	Spécialiste en Recrutement	ACTIVE	<div>UpdateDelete</div>

Liste des employés

Departments List

Add Department

Department ID	Department Name	Description	Actions
5	Production	Département responsable de la fabrication et de la production	<div>UpdateDelete</div>
6	Ventes et Marketing	Département responsable des ventes et des activités de marketing	<div>UpdateDelete</div>
7	Recherche et Développement	Département en charge de la recherche et du développement de nouveaux produits	<div>UpdateDelete</div>
8	Comptabilité et Finance	Département responsable de la gestion financière et de la comptabilité	<div>UpdateDelete</div>
9	Ressources humaines	Département chargé de la gestion des ressources humaines	<div>UpdateDelete</div>

Liste des départements

Postes List

[Add Poste](#)

Poste ID	Poste Titre	Description	Salaire	Département	Actions
2	Chef de Production	Responsable de la gestion de l'ensemble du processus de production	61000	Production	Update Delete
3	Directeur des Ventes	Supervise et gère l'équipe de vente	70000	Ventes et Marketing	Update Delete
4	Technicien de Laboratoire	Assiste les ingénieurs dans les expériences et les tests	40000	Recherche et Développement	Update Delete
5	Directeur Financier	Supervise toutes les activités financières de l'entreprise	80000	Comptabilité et Finance	Update Delete
6	Spécialiste en Recrutement	Gère le processus de recrutement de l'entreprise	50000	Ressources humaines	Update Delete
7	Directeur	Supervise toutes les opérations informatiques	90000	IT	Update

Liste des postes


Liste des Candidats

Id	Nom	Prénom	Adresse	Email	Numéro de téléphone	Diplôme	Expérience	Domaine d'Expérience	Date de Création	Actions	CV
1	Azaroua	Walid	maarif	walid@gmail.com	0645781220	bac +5 en ingenierie en info	3	developement	2024-06-23	Delete	View CV
2	Azaroua	Hamza	sidi maarouf	hamza@gmail.com	0645781220	bac +3 en marketing	4	vente et marketing	2024-06-23	Delete	View CV

Liste des candidats

Wijdane Company

Mon profilEmployésDépartementsPostesCandidatsCongés



Informations Personnelles

Services

Messages internes

Services

Début de congé:

jj/mm/aaaa

Fin de congé:

jj/mm/aaaa

Raison:

Demander Congé

Demande de congés

7. Tests et Validation

a) Stratégie de Test

La stratégie de test de notre projet comprend plusieurs niveaux de tests pour assurer la qualité et la robustesse du système. Nous avons utilisé une combinaison de tests unitaires, de tests d'intégration et de tests de performance pour valider les différentes composantes du système.

b) Tests Unitaires

Les tests unitaires sont utilisés pour vérifier que chaque unité de code fonctionne correctement. Nous avons utilisé le framework Mockito pour effectuer ces tests. Mockito permet de créer des objets factices (mock objects) pour isoler les composants à tester, en simulant le comportement des dépendances.

- Objectifs des tests unitaires :
 - Vérifier que les méthodes des services fonctionnent comme prévu.
 - Assurer que les contrôleurs renvoient les réponses attendues.
 - Tester les fonctions utilitaires indépendamment.

c) Tests d'Intégration

Les tests d'intégration vérifient que les différents composants du système fonctionnent ensemble correctement. Nous avons utilisé Postman pour effectuer ces tests. Postman permet de tester les endpoints de notre API en envoyant des requêtes HTTP et en vérifiant les réponses.

- Objectifs des tests d'intégration :
 - Vérifier la communication entre le frontend et le backend.
 - Assurer que les appels API renvoient les réponses attendues.
 - Tester les scénarios d'utilisation complets impliquant plusieurs composants.

d) Tests de Performance

Les tests de performance sont utilisés pour évaluer la réactivité et la stabilité du système sous des charges variées. Ces tests assurent que le système peut gérer un grand nombre de requêtes simultanées sans dégradation significative des performances.

- Objectifs des tests de performance :
 - Mesurer le temps de réponse des endpoints critiques.
 - Évaluer la capacité du système à gérer des charges élevées.
 - Identifier les goulots d'étranglement et les points de défaillance potentiels.

8. Conclusion générale

Le projet de gestion des ressources humaines que nous avons entrepris a été couronné de succès, aboutissant à la création d'un système intégré et performant. En combinant les technologies Spring Boot pour le backend et React pour le frontend, nous avons développé une solution moderne répondant aux exigences actuelles en matière de gestion du personnel.

Dès le début, nous avons mis en place une architecture backend solide. La configuration du projet Spring Boot, l'utilisation des entités JPA pour la modélisation des données, et l'implémentation de services encapsulant la logique métier ont permis de créer un système backend robuste et sécurisé. Les contrôleurs Spring Boot ont été développés pour gérer efficacement les requêtes HTTP, assurant ainsi une communication fluide entre le frontend et le backend.

Du côté frontend, l'utilisation de React a permis de concevoir une interface utilisateur dynamique et réactive. La structure du projet React avec des composants réutilisables a facilité le développement et la maintenance de l'application. L'intégration avec le backend a été réalisée via des services API, garantissant une interaction harmonieuse entre les différentes parties du système.

Pour assurer la qualité et la fiabilité de notre application, nous avons adopté une stratégie de tests complète. Les tests unitaires effectués avec Mockito ont permis de vérifier le bon fonctionnement de chaque composant individuellement. Les tests d'intégration réalisés avec Postman ont validé la communication et l'interaction entre les composants. Enfin, les tests de performance ont assuré que le système reste réactif et stable même sous des charges élevées.

En conclusion, ce projet de gestion des ressources humaines a abouti à une solution efficace et prête à l'emploi, capable de répondre aux besoins des entreprises modernes. Le système offre une gestion complète des employés et des congés, tout en assurant la sécurité des données et la performance. Ce projet pose les bases pour de futures améliorations et extensions, garantissant ainsi sa pertinence et son utilité à long terme. Nous sommes convaincus que cette solution contribuera de manière significative à l'efficacité et à la gestion optimisée des ressources humaines dans toute organisation qui l'adoptera.