

Présentation

Dans le cadre de la digitalisation des processus internes, une entreprise souhaite mettre en place une application JEE permettant de gérer efficacement : ses ressources humaines, ses départements, ses projets internes ainsi que la génération des fiches de paie mensuelles.

Ce projet vise à développer une application JEE qui permette à l'administrateur ou au responsable RH de gérer : les employés (ajout, modification, suppression, affectation à un département ou à un projet), les départements (création, mise à jour, liste des membres), les projets d'entreprise (création, affectation des employés, suivi), les fiches de paie mensuelles (calcul automatique basé sur le salaire, les primes, les absences éventuelles, etc.)

1. Fonctionnalités attendues

1.1. 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. Gestion des departments

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

1.3. Gestion des projets

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

1.4. Gestion des fiches de paie

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

1.5. Authentification et Autorisation : gestion des rôles (Administrateur, chef du département, chef du projet, employé)

2. Architecture du Projet

L'architecture de l'application doit respecter le modèle MVC

- **Modèle** : Classes Java représentant les entités (Employé, Département, Projet, Fiche de paie).
- **Vue** : les interfaces qui interagissent directement avec les utilisateurs de l'application pour la récupération et l'affichage des données (JSP, HTML, CSS, JavaScript,)
- **Contrôleur** : Servlets pour gérer les requêtes HTTP et la logique métier (en Java)
- **Base de Données** : pour la gestion des données et un système de gestion de base de données (MySQL, PostgreSQL, ...).
- **Couche de persistance de données** : Hibernate et JDBC.

3. Etapes de Développement et contraintes à respecter

- Analyse des Besoins : Identifier les besoins spécifiques
- Installer et configurer, ApacheTomcat, Hibernate et un système de gestion de base de données (MYSQL, PostgreSQL, ...)
- Créer la base de données les différentes tables nécessaires.
- Développer les classes Java pour les différentes entités (employés, département,...)
- Créer des servlets pour gérer les opérations **CRUD** (Create, Read, Update, Delete), lecture, écriture et la mise à jour de la base de données.
- Développer les interfaces nécessaires permettant l'interaction avec les utilisateurs
- Mettre en place une gestion des utilisateurs avec authentification pour sécuriser l'accès aux différentes fonctionnalités.

- Toutes les données saisies des interfaces doivent être vérifiées par les servlets avant d'être sauvegardées sur la base de données
- Ajouter une fonctionnalité pour générer des rapports sur les statistiques des projets, le nombre d'employés par département, le nombre d'employés par projet et leurs grades.

4. Travail demandé

4.1. **La partie conception** : Dans cette partie, vous donnez le **MCD (Modèle Conceptuel des Données)** ou le diagramme de classes qui représente les différentes interactions entre les entités (classes) de l'application (Selon la méthode de conception utilisée).

4.2. **La partie implémentation** : vous créer une application Web dynamique qui doit :

- Fournir toutes les fonctionnalités décrites dans la section 1.
- Respecter tous les points les exigences données dans la section 3.
- Utiliser les technologies vues en cours et cités aussi en niveau de la section 2
- Assurer une cohérence de données pour toute manipulation de la base de données

4.3. Refaire l'application en utilisant **Spring Boot**

5. Exigences

- Le travail demandé sera réalisé par un groupe de **4 à 5 étudiants**.
- Chaque groupe doit être inscrit sur la liste qui sera mis sur teams et fournit un lien GitHub permettant l'accès à l'application (vous devez fournir un **lien privé au début** et vous le rendez **public à la date du dépôt final**) avant le 24/10/2025 23h59.

Une pénalité est appliquée en cas de retard. Voici le lien de la page :

https://docs.google.com/spreadsheets/d/15y4mWdAUwWHXYNM009-kSxcnMxjpmK_46aqVrTBNb9I/edit?usp=sharing

- Chaque groupe d'étudiants doit rendre un rapport qui détaille la partie conception, explique le schéma de la partie implémentation (partie 2+partie SpringBoot), un fichier qui explique les outils utilisés pour le développement et le déploiement de l'application et faire une présentation orale entre **(10 à 15 min)**. Le rapport ne doit pas dépasser **15 pages**.
- Le dépôt final comporte le rapport et l'application que vous avez implémentée
- La date limite de dépôt est le **29/11/2025, 23h59**. Une **pénalité est appliquée en cas du retard**