



ÉCOLE SUPÉRIEURE DE TECHNOLOGIE DE SAFI

DÉPARTEMENT: INFORMATIQUE

# Rapport de TP de la gestion des employés

Chaimaa Gomi

Année Universitaire: 2024/2025

# Contents

|   |  |   |
|---|--|---|
| 1 | Introduction                               | 2 |
| 2 | Ajout d'un employé                         | 3 |
| 3 | Affichage de la liste des employés         | 4 |
| 4 | Modification des informations d'un employé | 5 |
| 5 | Suppression d'un employé                   | 6 |
| 6 | Gestion des congés des employés            | 6 |
| 7 | Insertion des images du projet             | 6 |

# 1 Introduction

Dans ce rapport, nous allons détailler le développement d'un projet de gestion des employés. Ce projet permet de gérer une liste d'employés avec les fonctionnalités suivantes :

- Ajouter un nouvel employé avec des informations telles que le nom, le prénom, l'email, le téléphone, le salaire, le rôle, et le poste.
- Afficher la liste complète des employés.
- Modifier les informations d'un employé existant.
- Supprimer un employé de la liste.

Ce projet est un outil pratique pour toute entreprise souhaitant gérer efficacement son personnel.

## 2 Ajout d'un employé

L'ajout d'un employé dans le système se fait en remplissant un formulaire avec les informations suivantes :

- **Nom** : Le nom de l'employé.
- **Prénom** : Le prénom de l'employé.
- **Email** : L'email de l'employé.
- **Téléphone** : Le numéro de téléphone de l'employé.
- **Salaire** : Le salaire de l'employé.
- **Rôle** : Le rôle de l'employé (par exemple : Développeur, Chef de projet, etc.).
- **Poste** : Le poste de l'employé dans l'entreprise.

Voici un exemple de code en **\*\*Java\*\*** pour ajouter un employé :

```
public class Employe {
    private String nom;
    private String prenom;
    private String email;
    private String telephone;
    private double salaire;
    private String role;
    private String poste;

    // Constructeur
    public Employe(String nom, String prenom, String email, String telephone, double salaire, String role, String poste) {
        this.nom = nom;
        this.prenom = prenom;
        this.email = email;
        this.telephone = telephone;
        this.salaire = salaire;
        this.role = role;
        this.poste = poste;
    }

    // Getters et Setters pour chaque attribut
}
```

Exemple d'ajout d'un employé :

```
public class Main {
    public static void main(String[] args) {
        // Exemple d'ajout d'un employé
        Employe employe1 = new Employe("Dupont", "Jean", "jean.dupont@example.com", "01 23 45 67 89", 2500, "Développeur", "Ingénieur");
    }
}
```

### 3 Affichage de la liste des employés

Une fois que les employés ont été ajoutés, il est important de pouvoir afficher la liste complète des employés. Le code suivant montre comment afficher les informations de tous les employés :

```
import java.util.List;

public class Main {
    public static void afficherEmployes(List<Employe> listeEmployes) {
        for (Employe employe : listeEmployes) {
            System.out.println("Nom: " + employe.nom + ", Prénom: " + employe.prenom +
                                ", Email: " + employe.email + ", Téléphone: " + employe.tel +
                                ", Salaire: " + employe.salaire + ", Rôle: " + employe.role +
                                ", Poste: " + employe.poste);
        }
    }

    public static void main(String[] args) {
        // Exemple d'affichage
        List<Employe> employees = List.of(new Employe("Dupont", "Jean", "jean.dupont@e", 1000, "R", "D"));
        afficherEmployes(employees);
    }
}
```

## 4 Modification des informations d'un employé

Pour modifier les informations d'un employé, nous pouvons utiliser une méthode qui permet de changer les attributs de l'employé sélectionné. Exemple de code pour modifier un employé :

```
public class Main {
    public static void modifierEmploye(Employe employe, String nouveauNom, String nouveauPrenom, String nouvelEmail, String nouveauTelephone, int nouveauSalaire, String nouveauRole, String nouveauPoste) {
        if (nouveauNom != null) employe.nom = nouveauNom;
        if (nouveauPrenom != null) employe.prenom = nouveauPrenom;
        if (nouvelEmail != null) employe.email = nouvelEmail;
        if (nouveauTelephone != null) employe.telephone = nouveauTelephone;
        if (nouveauSalaire != 0) employe.salaire = nouveauSalaire;
        if (nouveauRole != null) employe.role = nouveauRole;
        if (nouveauPoste != null) employe.poste = nouveauPoste;
    }

    public static void main(String[] args) {
        Employe employe1 = new Employe("Dupont", "Jean", "jean.dupont@example.com", "01 23 45 67 89", 3500, "Lead Développeur", "Lead Développeur");
        modifierEmploye(employe1, null, null, null, null, 3500, "Lead Développeur", null);
    }
}
```

## 5 Suppression d'un employé

La suppression d'un employé se fait en le retirant de la liste des employés. Exemple de code pour supprimer un employé :

```
import java.util.List;

public class Main {
    public static void supprimerEmploye(List<Employe> listeEmployes, Employe employe) {
        listeEmployes.remove(employe);
    }

    public static void main(String[] args) {
        List<Employe> employees = List.of(new Employe("Dupont", "Jean", "jean.dupont@ex", "01 23 45 67 89"));
        supprimerEmploye(employees, employees.get(0));
    }
}
```

## 6 Gestion des congés des employés

La gestion des congés des employés repose sur une base de données structurée permettant de suivre efficacement les absences. Cette fonctionnalité s'appuie sur deux tables principales :

- **Table employe** : Contient les informations personnelles des employés, telles que leur nom, prénom, email, téléphone, salaire, rôle et poste.
- **Table holiday** : Enregistre les congés des employés, avec des champs tels que le type de congé (vacances, maladie, etc.), la date de début et la date de fin, ainsi qu'une relation avec l'identifiant de l'employé.

Ce système permet de :

- Ajouter et enregistrer les congés des employés.
- Consulter les dates de début et de fin des congés.
- Identifier facilement les périodes d'indisponibilité des employés.

## 7 Insertion des images du projet

Voici les images du projet que nous avons utilisées pour illustrer certaines fonctionnalités du système de gestion des employés :

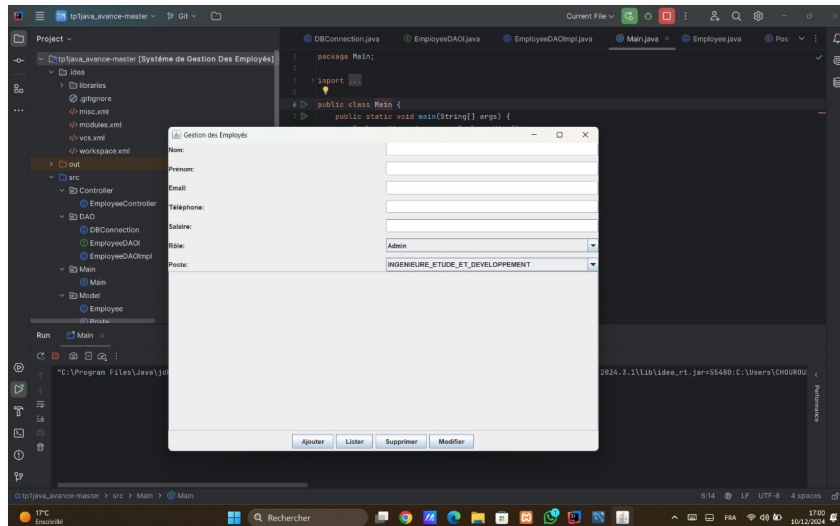


Figure 1: Vue de l'interface d'ajout d'un employé

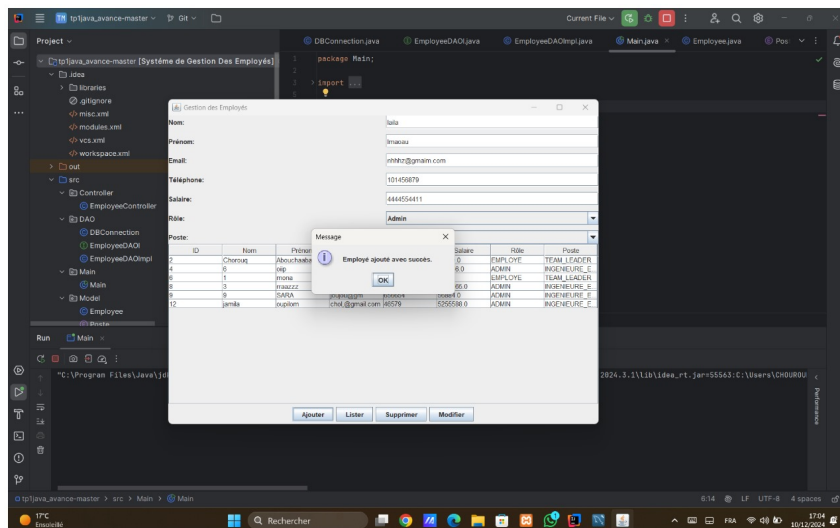


Figure 2: Vue de la liste des employés

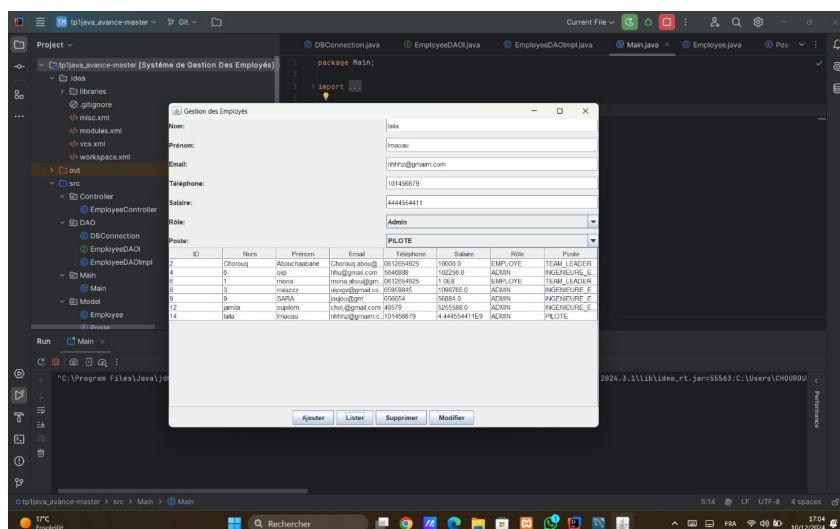


Figure 3: Modification des informations d'un employé



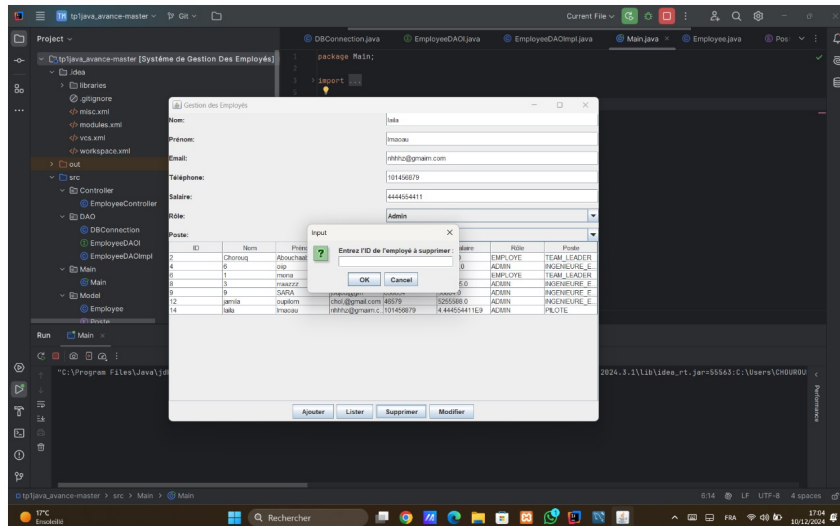


Figure 4: Suppression d'un employé de la liste

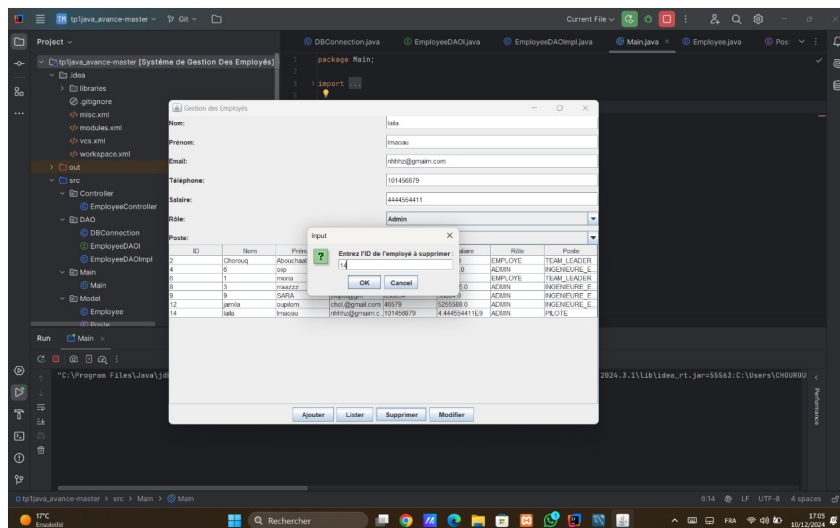


Figure 7: Vue de l'interface de gestion des employés

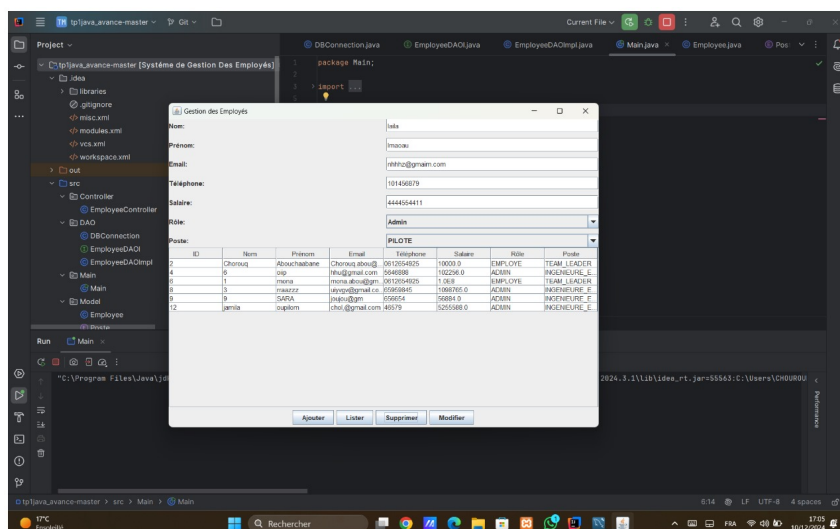


Figure 8: Ajout d'un employé avec des informations spécifiques

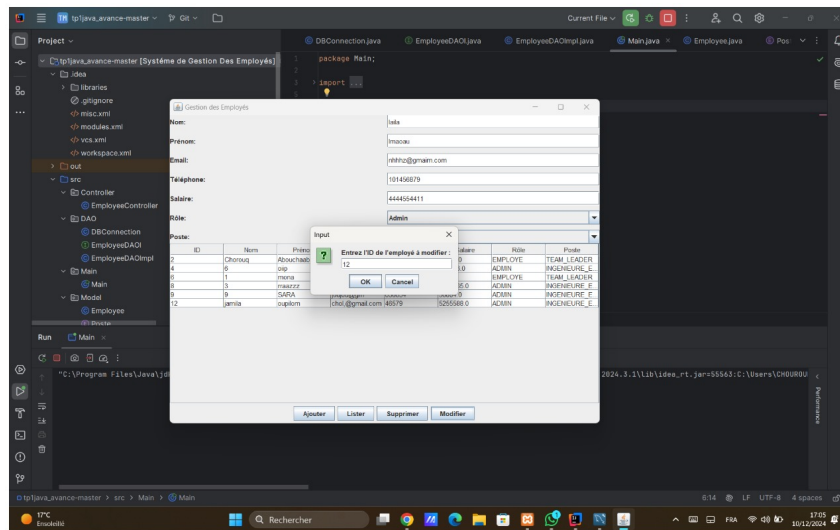


Figure 9: Affichage des informations de tous les employés

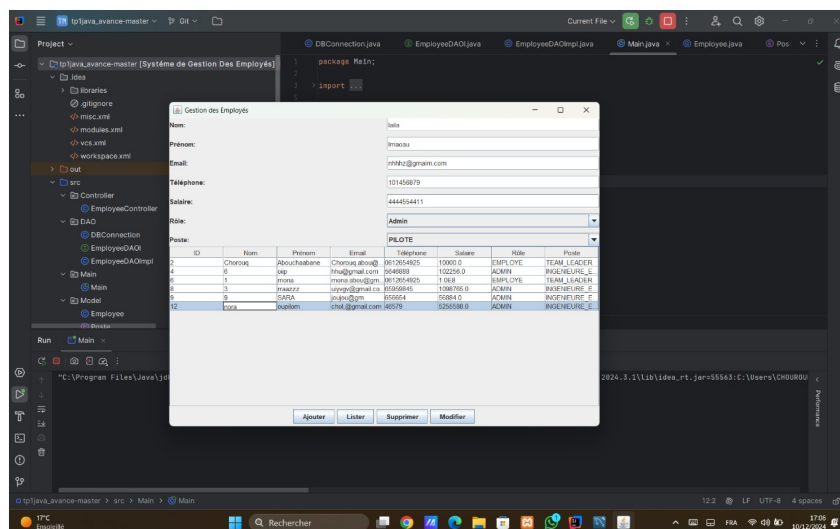


Figure 10: Suppression d'un employé confirmée