Rapport de Projet Java

TP1 : Gestion des Employés

Nom de l'auteur : CHOKRANI Rachid
Date : 8 décembre 2024

Table des matières

Chapitre 1

Introduction

Ce projet consiste à développer un système de gestion des employés en Java. Il inclut des fonctionnalités telles que l'ajout, la mise à jour, la suppression et la recherche d'employés. Ce projet vise à appliquer les concepts de la programmation orientée objet (POO) tout en simulant un système de gestion pratique pour une entreprise.

1.1 Objectifs du Projet

— Implémenter un système de gestion des employés.
— Appliquer les concepts de la POO (encapsulation, héritage, polymorphisme).
— Utiliser une base de données pour sauvegarder et charger les données.
—

1.2 Diagramme UML

Voici un diagramme UML décrivant les principales classes du système (voire Figure 1.1 – Diagramme de Classes UML page 3) :

1.3 Architecture

L'architecture du projet suit une approche MVC (Modèle-Vue-Contrôleur) avec une couche DAO (Data Access Object) pour l'interaction avec la base de données. Voici un aperçu des entités principales :

— **Model** :
  — **Classe Employee** : Contient les informations d'un employé (ID, nom, prénom, email, téléphone, salaire, rôle, poste).
  — **Classe EmployeeModel** : Assure la liaison entre les classes DAO et la classe Employee.
— **DAO** :
  — **Classe DBconnexion** : Assure la connexion à la base de données.
  — **Interface EmployeeDAOI** : Contient les méthodes abstraites pour interagir avec la base de données.
  — **Classe EmployeeDAOImpl** : Implémente les méthodes de l'interface EmployeeDAOI.
— **View** :

  - **Classe EmployeeView** : Interface graphique du programme.
- **Controller** :
  - **Classe EmployeeController** : Gère les opérations (ajouter, modifier, supprimer, afficher) et assure la liaison entre les classes Model et View.
- **Classe Main** : Point d'entrée de l'application.

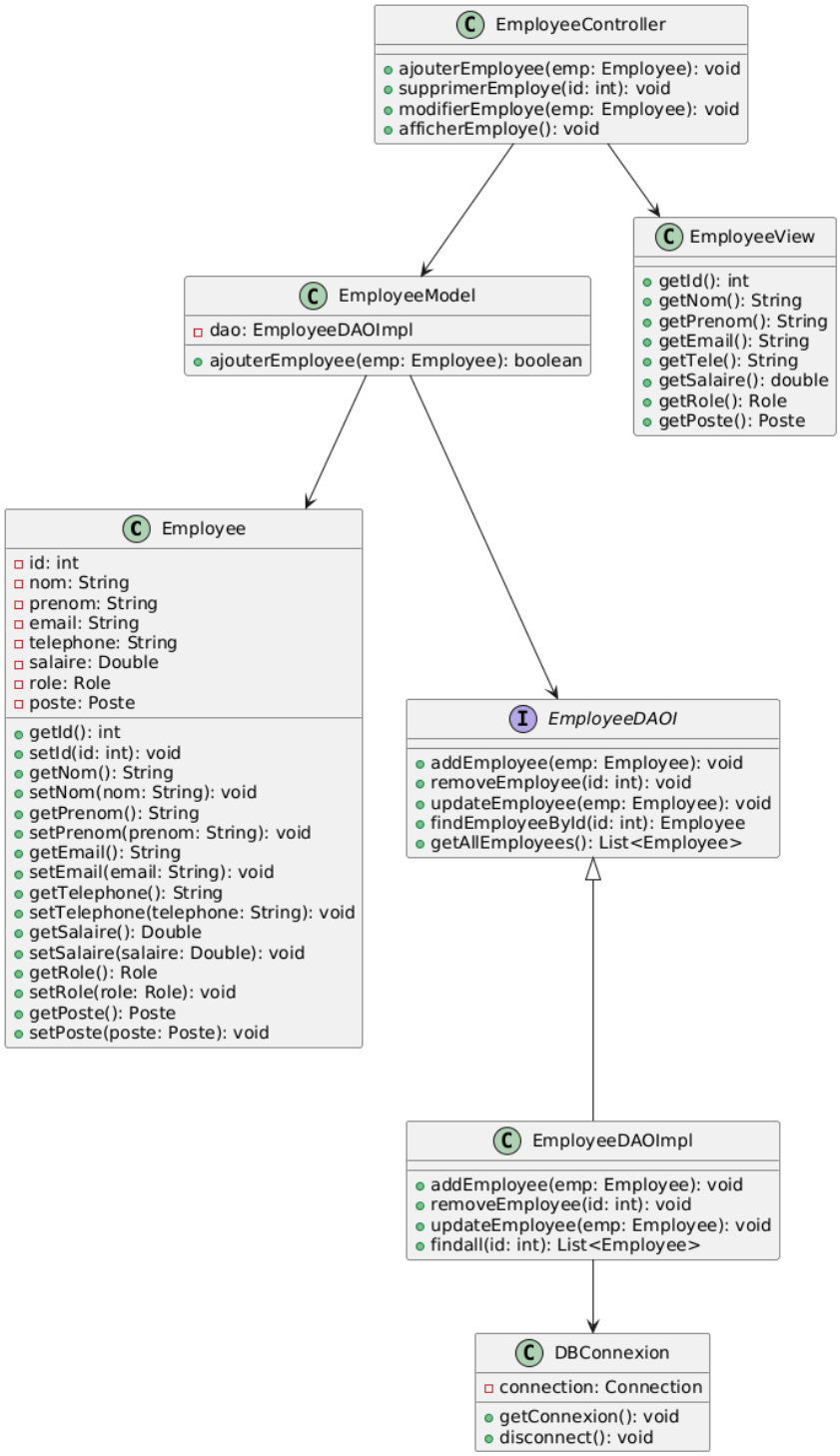

FIGURE 1.1 – Diagramme de Classes UML

—

Chapitre 2

Développement

2.1 Outils Utilisés

Les outils utilisés pour le développement du projet incluent :
— Java SE 17
— IntelliJ IDEA
— JDBC

2.2 Exemple de Code

Voici un extrait de la classe ‘Employee‘ :

```java
public class Employee {
    private String Nom, Prenom, Email, Tele;
    private Double Salaire;
    private Role role;
    private Poste poste;
    private int id;

    public Employee(int id, String Nom, String Prenom, String Email
        , String Tele, Double Salaire, Role role, Poste poste) {
        this.id = id;
        this.Nom = Nom;
        this.Prenom = Prenom;
        this.Email = Email;
        this.Tele = Tele;
        this.Salaire = Salaire;
        this.role = role;
        this.poste = poste;
    }
    // Getters et Setters
    public int getId() { return id; }
    public void setId(int id) { this.id = id; }
    // Autres getters et setters...
}
```

Listing 2.1 – Classe Employee

—

Chapitre 3

Résultats

3.1 Fonctionnalités Implémentées

Les fonctionnalités suivantes ont été implémentées dans le système :
— Ajouter un employé.
— Modifier les informations d'un employé.
— Supprimer un employé.
— Rechercher un employé.

3.2 Captures d'Écran

Voici une capture d'écran de l'application en fonctionnement :

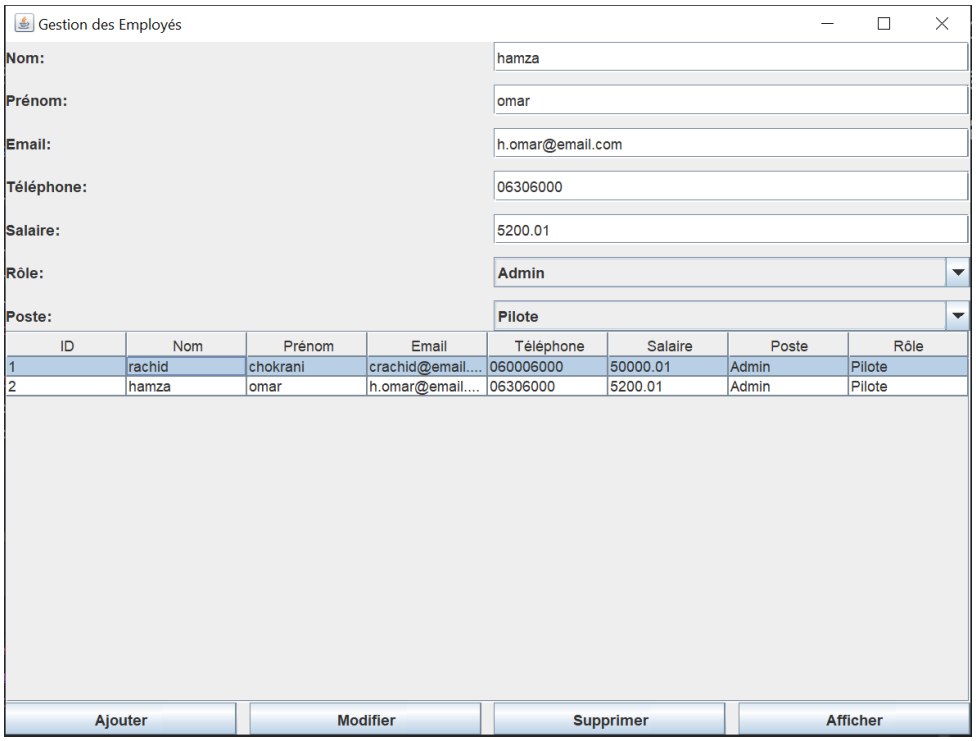

FIGURE 3.1 – Exemple d'écran de l'application

—

3.3 Problèmes Rencontrés et Solutions

Voici quelques problèmes rencontrés lors du développement et leurs solutions :

— **Problème** : Gestion des doublons lors de l'ajout d'employés.
Solution : Ajout d'une vérification basée sur l'ID de l'employé.
— **Problème** : Erreurs de connexion à la base de données.
Solution : Implémentation de blocs 'try-catch' pour gérer les exceptions.

3.4 Conclusion et Perspectives

Ce projet a permis de mettre en œuvre un système de gestion des employés en Java en appliquant les principes de la POO. Les futures améliorations pourraient inclure :

— Ajout d'une interface graphique (Swing ou JavaFX).
— Intégration avec une base de données comme MySQL.
— Gestion des rôles et des autorisations des utilisateurs.
—

Annexe A

Annexes

Voici un extrait du Main :

```
public class Main {
    public static void main(String[] args) {

        EmployeeView view = new EmployeeView();
        EmployeeDAOImpl dao = new EmployeeDAOImpl();
        EmployeeModel model = new EmployeeModel(dao);

        new EmployeeController(model,view);
    }
}
```

Listing A.1 – Extrait du Main

—