Java : Rapport de Session Pratique

Ayoub Elamrani

December 22, 2024

Table des matières

1 Objectif :

L'objectif de cette session pratique était de concevoir et d'implémenter un système de gestion des employés basé sur Java avec une architecture multi-couche comprenant des modèles, des vues, des contrôleurs et des interactions avec la base de données.

2 Aperçu de l'Exercice :

L'exercice consistait à créer les composants suivants :

- **Modèle** : Une représentation des données des employés incluant des attributs tels que nom, email, téléphone, salaire, poste et rôle.
- **DAO (Data Access Object)** : Une couche pour interagir avec la base de données PostgreSQL via des requêtes SQL.
- **Contrôleur** : Gère la logique pour ajouter, supprimer, mettre à jour et afficher les employés.
- **Vue** : Une interface graphique (GUI) pour les interactions avec l'utilisateur.

3 Mise en Oeuvre du Code :

3.1 Classe Principale :

```
// Import des classes necessaires
import Controllers.EmployeeController;
import DAO.EmployeeDAOImpl;
import Views.EmployeeView;

public class Main {
    public static void main(String[] args) {
        // Creation objet DAO pour acces aux donnees employees
        EmployeeDAOImpl dao = new EmployeeDAOImpl();

        // Creation controleur pour gerer interactions vue et DAO
        EmployeeController ec = new EmployeeController();

        // Creation vue pour afficher infos employees


    }
}
}
```

3.2 DAO :

3.2.1 Interface DAO

```java
// Interface definissant les operations de base pour la gestion des employees en base de donnees
package DAO;

import Models.Employee;

interface EmployeeDAOI {
    // Constantes de connexion a la base de donnees MySQL
    public String url = "jdbc:mysql://localhost:3306/tp1_db";  // URL de connexion
    public String dbuser = "root";                              // Nom utilisateur DB
    public String dbpw = "root";                                // Mot de passe DB



    // Ajoute un nouvel employee dans la base
    public boolean addEmployee(Employee em);

    // Supprime un employee par son ID
    public boolean deleteEmployee(int id);

    // Met a jour les infos d'un employee existant
    public boolean updateEmployee(int id, Employee em);
}
```

3.2.2 Implémentation DAO :

```java
package DAO;

public class EmployeeDAOImpl implements EmployeeDAOI {
  // Constructeur
  public EmployeeDAOImpl();

  // M thodes    surcharger ici
}
```

3.3 Modèle :

```
package Models;

public class Employee {
  // Constructeur
  public Employee(ResultSet rs);

  // setters
  public int getId();
  public String getNom();
  public String getPrenom();
  public String getEmail();
  public double getSalaire();
  public String getTelephone();
  public String getPoste();
  public String getRole();

  // getters
  public void setId(int id);
  public void setNom(String nom);
  public void setPrenom(String prenom);
  public void setEmail(String email);
  public void setSalaire(double salaire);
  public void setTelephone(String telephone);
  public void setPoste(String poste);
  public void setRole(String role);

  // Methodes pour les interactions avec le contr leur
  public boolean addEmployee();
  public static boolean deleteEmployee(int id);
  public boolean updateEmployee(int id);

  public String toString();
}
```

3.4 Contrôleur :

```
package Controllers;

public class EmployeeController {
  // Constructeur
  public EmployeeController();

  // Methodes d'initialisation des    vnements
  private void initAddEvent();
  private void initDeleteEvent();
  private void initUpdateEvent();
  private void initShowEvent();

  // Methodes utiles pour g rer la vue
  public static void populateTable();
  public static void emptyFields();
}
```

3.5 Vue :

```java
package Views;

public class EmployeeView extends JFrame {
  // Constructeur
  public EmployeeView();
}
```

4 Résultats :

.

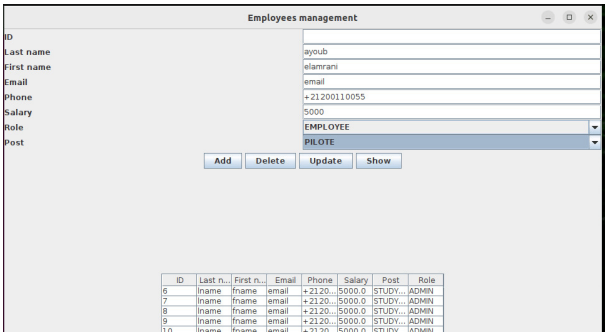

Figure 1: Aperçu de l'application

La mise en œuvre a démontré avec succès les fonctionnalités suivantes :

- Établir une connexion avec une base de données MySQL.
- Ajouter, supprimer et mettre à jour des enregistrements d'employés via des interactions GUI.
- Afficher les données des employés sous forme tabulaire dans la GUI.

5 Conclusion :

L'objectif de cette session était de développer une application Java complète intégrant une interface utilisateur graphique (GUI) et une base de données, tout en renforçant la compréhension de l'architecture MVC et des interactions avec une base de données SQL .

6 Références :

- Documentation Java : https://docs.oracle.com/en/java/
- Documentation MySQL :https://dev.mysql.com/doc/
- Overleaf :https://www.overleaf.com//
- github :https://github.com/AyoubElam/tp_java