

Université Cadi Ayyad  
École Supérieure De Technologie-Safi  
Département : Informatique  
Filière : Génie Informatique

## Rapport du TP Gestion des Employés

---

# Développement d'une application de gestion des employés(MVC,DAO et Java Swing)

---

Date de réalisation : 6 Décembre

Réalisé par :

Mlle. BAILOUMY Hiba

Enseigné par :

M. EL ABDELLAOUI Said  
Mme. KACHBAL Ilham

ANNÉE UNIVERSITAIRE : 2024/2025

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>4</b>
<b>2</b>	<b>Contexte du TP</b>	<b>5</b>
1	Problématique . . . . .	5
2	Solution . . . . .	5
3	Processus de Développement . . . . .	5
3.1	Analyse des besoins . . . . .	5
3.2	Conception de l'architecture . . . . .	6
3.3	Développement . . . . .	6
3.4	Tests et validation . . . . .	6
<b>3</b>	<b>Conception</b>	<b>7</b>
1	Base de données . . . . .	7
2	Création de la base de données et de la table employés . . . . .	7
3	Environnement de travail . . . . .	8
3.1	Eclipse . . . . .	8
3.2	XAMPP . . . . .	8
3.3	MYSQL Workbench . . . . .	9
3.4	Architecture Utilisée . . . . .	9
<b>4</b>	<b>Réalisation</b>	<b>11</b>
1	Page d'accueil . . . . .	11
1.1	Formulaire de saisie des informations . . . . .	11
1.2	Boutons d'action . . . . .	11
2	Gestion des employes . . . . .	12
2.1	Ajouter un employe . . . . .	12
2.2	Afficher la liste des employes . . . . .	13
2.3	Modifier un employe . . . . .	14
2.4	Supprimer un employe . . . . .	15
<b>5</b>	<b>Conclusion Générale</b>	<b>17</b>

# Table des figures

3.1	création de la table employe . . . . .	8
3.2	eclipse . . . . .	8
3.3	xampp . . . . .	9
3.4	MYSQL Workbench . . . . .	9
3.5	MVC . . . . .	10
3.6	DAO . . . . .	10
4.1	page d'accueil . . . . .	12
4.2	Ajouter un employe . . . . .	13
4.3	Afficher les employes . . . . .	13
4.4	Message pour la modification . . . . .	14
4.5	Modifier un employe . . . . .	15
4.6	Message pour la suppression . . . . .	15
4.7	Supprimer un employe . . . . .	16

# Introduction

La gestion des employés constitue un pilier essentiel pour toute organisation cherchant à optimiser sa performance et à renforcer sa structure interne. Dans le cadre de ce travail, nous avons conçu une application de bureau axée sur la gestion des employés en adoptant les principes de l'architecture MVC (Modèle-Vue-Contrôleur) et du design pattern DAO (Data Access Object).

Développée en Java avec l'interface Swing et une base de données MySQL, cette application offre aux administrateurs une plateforme centralisée et intuitive pour gérer les données des employés. Elle vise à automatiser les processus de gestion courants tels que l'ajout, la modification, la suppression et la consultation des informations des employés, tout en assurant une expérience utilisateur fluide et efficace.

Reposant sur l'architecture MVC, cette solution se distingue par une séparation rigoureuse des responsabilités : le Modèle gère les données des employés, la Vue présente une interface graphique conviviale, et le Contrôleur orchestre les interactions entre l'utilisateur et le système. L'intégration du design pattern DAO permet d'isoler la logique d'accès aux données, simplifiant ainsi la maintenance et l'extension de l'application.

Parmi les fonctionnalités principales, l'application prend en charge la gestion complète des employés, incluant la création, la mise à jour et la suppression des profils, ainsi que la consultation des informations. Conçue pour être à la fois ergonomique et fiable, l'application garantit une manipulation sécurisée des données tout en répondant aux besoins de l'organisation.

L'objectif de ce projet est de proposer une solution efficace, modulaire et pérenne pour gérer les employés. Il s'inscrit dans une démarche d'apprentissage pratique et explore les améliorations potentielles pour répondre aux exigences évolutives des organisations.

# Contexte du TP

Dans ce chapitre, nous présentons le contexte du travail pratique, les objectifs visés, ainsi que les raisons qui ont motivé la réalisation de ce projet.

## 1 Problématique

Dans ce TP, nous nous concentrons sur le développement d'un système performant et structuré pour la gestion des employés. L'absence d'une architecture claire peut conduire à une désorganisation des données, engendrant des erreurs et une perte de temps. Pour remédier à cela, l'objectif est de concevoir une application centralisée qui optimise la gestion des informations des employés. En adoptant l'architecture MVC et le design pattern DAO, nous visons à séparer la logique métier de l'accès aux données, tout en assurant une organisation efficace, une meilleure maintenance et une extensibilité accrue du système.

## 2 Solution

L'application développée s'appuie sur les principes de l'architecture MVC pour garantir une séparation claire des responsabilités et une meilleure organisation du code. Voici comment elle est structurée :

Le Modèle (Model) : Représente les données des employés sous forme d'objets Java, assurant la gestion des informations comme le nom, le poste, et d'autres détails liés aux employés. La Vue (View) : Offre une interface utilisateur graphique intuitive grâce à Java Swing, permettant de visualiser, modifier ou supprimer les données des employés de manière interactive. Le Contrôleur (Controller) : Agit comme un intermédiaire entre la Vue et le Modèle. Il traite les actions de l'utilisateur, interagit avec la base de données via le DAO et met à jour l'interface en conséquence. Le DAO (Data Access Object) : Simplifie et centralise l'accès à la base de données MySQL, facilitant les opérations CRUD (Créer, Lire, Mettre à jour, Supprimer) sur les données des employés tout en rendant le code plus modulaire et maintenable.

## 3 Processus de Développement

### 3.1 Analyse des besoins

Identification des principales fonctionnalités de l'application, telles que la gestion des informations des employés (ajouter, modifier, supprimer, afficher). Définition des interactions utilisateur avec la base de données, assurant la fluidité des opérations CRUD (Créer, Lire, Mettre à jour, Supprimer).

## 3.2 Conception de l'architecture

Choix de l'architecture MVC pour séparer clairement la logique métier, l'accès aux données et l'interface graphique. Conception de la structure de la base de données pour stocker les informations des employés et développement des classes Java correspondant aux entités.

## 3.3 Développement

Création des classes Java pour la gestion des informations des employés, en implémentant les propriétés et méthodes nécessaires. Développement de l'interface graphique avec Java Swing, offrant une expérience utilisateur claire et fonctionnelle. Implémentation du design pattern DAO pour gérer l'accès aux données de manière modulaire et sécurisée.

## 3.4 Tests et validation

Tests des différentes fonctionnalités CRUD pour s'assurer de leur bon fonctionnement. Vérification de l'intégration entre l'interface graphique et la base de données, garantissant une synchronisation correcte des données.

Ce travail pratique a permis de concevoir et de développer une application fonctionnelle de gestion des employés en appliquant des concepts avancés de Java, comme l'architecture MVC et le design pattern DAO. L'application, dotée d'une interface utilisateur fluide et simple, répond aux besoins de gestion des informations des employés de manière efficace. Ce projet a renforcé ma maîtrise des bonnes pratiques de développement logiciel et m'a permis de mieux comprendre les principes de conception d'architectures logicielles robustes.

# Conception

## 1 Base de données

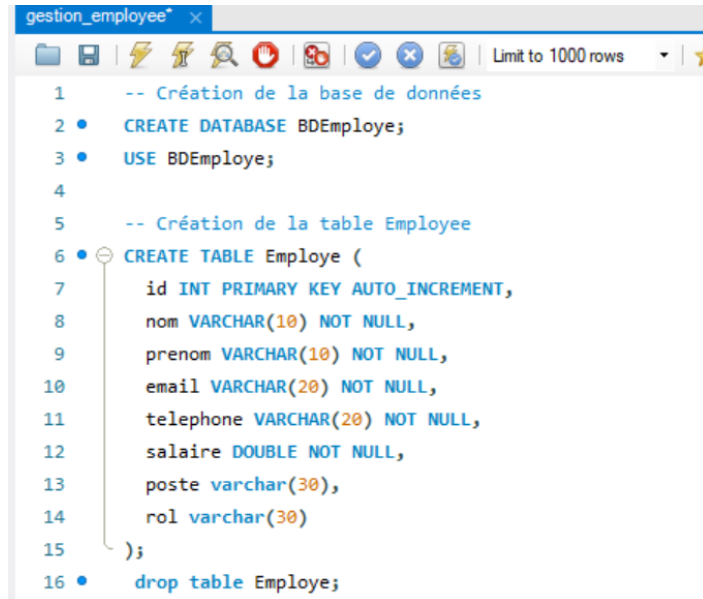
Le développement de l'application a commencé par la création de la base de données, une étape essentielle pour structurer et organiser les informations relatives aux employés. Cette base de données est au cœur du système, permettant un stockage efficace des données et facilitant les interactions avec les différentes fonctionnalités de l'application.

## 2 Création de la base de données et de la table employés

La table employes a été conçue dans une base de données MySQL afin de regrouper toutes les informations pertinentes concernant chaque employé. Cette table permet de stocker les données de manière organisée, en facilitant les opérations de gestion comme l'ajout, la modification et la suppression des informations des employés.

Table employes : Elle comprend les champs suivants :

Champ	Description
id	Identifiant unique de l'employé (clé primaire).
nom	Nom de l'employé.
prenom	Prénom de l'employé.
email	Adresse e-mail de l'employé.
telephone	Numéro de téléphone de l'employé.
salaire	Salaire de l'employé.
role	Rôle de l'employé (par exemple, ADMIN ou EMPLOYE).
poste	Poste de l'employé (par exemple, INGENIEUR, PROFESSEUR, ou DOCTEUR).



```
1  -- Création de la base de données
2  • CREATE DATABASE BDEmploye;
3  • USE BDEmploye;
4
5  -- Création de la table Employee
6  • CREATE TABLE Employee (
7      id INT PRIMARY KEY AUTO_INCREMENT,
8      nom VARCHAR(10) NOT NULL,
9      prenom VARCHAR(10) NOT NULL,
10     email VARCHAR(20) NOT NULL,
11     telephone VARCHAR(20) NOT NULL,
12     salaire DOUBLE NOT NULL,
13     poste varchar(30),
14     rol varchar(30)
15 );
16 • drop table Employee;
```

FIGURE 3.1 – création de la table employe

## 3 Environnement de travail

Pour le développement et la gestion de ce projet, j’ai utilisé un environnement de travail composé des outils et technologies suivants :

### 3.1 Eclipse

Eclipse est un environnement de développement intégré (IDE) open-source principalement utilisé pour le développement de logiciels en Java, mais qui prend également en charge d’autres langages de programmation grâce à des plugins. Il offre des outils puissants pour l’écriture, le débogage et le test de code, ainsi que des fonctionnalités avancées telles que la gestion de projets, la navigation dans le code et l’intégration avec des systèmes de contrôle de version.



FIGURE 3.2 – eclipse

### 3.2 XAMPP

XAMPP est une solution de serveur local multiplateforme qui regroupe plusieurs outils nécessaires au développement web, notamment Apache, MySQL (ou MariaDB), PHP et Perl. Cet environnement permet de créer et de tester des applications web localement sans nécessiter un serveur en ligne. Grâce à sa simplicité d’installation et d’utilisation, XAMPP est souvent utilisé par les développeurs pour des projets de développement web et de bases de données.





FIGURE 3.3 – xampp

### 3.3 MySQL Workbench

MySQL Workbench est un outil graphique officiel de MySQL, conçu pour la conception, la gestion et l'administration des bases de données. Il permet de créer des modèles de données, d'exécuter des requêtes SQL, de gérer les connexions et de visualiser les relations entre les tables. Grâce à son interface intuitive, il facilite le travail des développeurs et administrateurs de bases de données.



FIGURE 3.4 – MYSQL Workbench

### 3.4 Architecture Utilisée

L'application a été développée en suivant l'architecture **MVC (Modèle-Vue-Contrôleur)** combinée avec le patron de conception **DAO (Data Access Object)**.

- **MVC** : Cette architecture permet une séparation claire entre :
  - Le **Modèle**, qui gère les données et les règles métier.
  - La **Vue**, qui est responsable de l'interface utilisateur.
  - Le **Contrôleur**, qui sert d'intermédiaire entre le Modèle et la Vue, en gérant les interactions utilisateur et les mises à jour des données.

Cette structure améliore la maintenabilité et la modularité de l'application.

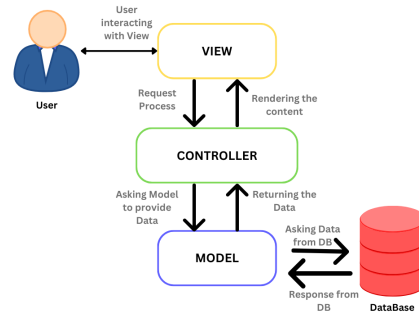


FIGURE 3.5 – MVC

- **DAO (Data Access Object)** : Utilisé pour gérer les interactions entre l'application et la base de données. Chaque table (par exemple, `employees`, `role`, `poste`) dispose d'un DAO dédié, qui centralise les opérations de lecture, écriture, mise à jour et suppression des données. Cela garantit une meilleure abstraction et réduit le couplage entre la logique métier et l'accès aux données.



FIGURE 3.6 – DAO

# Réalisation

Dans ce chapitre, nous allons présenter les principales fonctionnalités de l'application, conçue pour simplifier et optimiser la gestion des employés. Cette application offre une interface simple et intuitive permettant d'effectuer les opérations essentielles telles que l'ajout, la modification, la suppression et la consultation des informations des employés.

L'objectif principal est de fournir une solution pratique et efficace, en exploitant les technologies modernes et une architecture logicielle robuste. Nous aborderons également les outils et méthodologies employés pour le développement, ainsi que les défis rencontrés lors de la mise en œuvre des différentes fonctionnalités.

## 1 Page d'accueil

La page d'accueil de l'application constitue le point central permettant à l'utilisateur de gérer efficacement les employés. Cette interface a été conçue pour être à la fois simple, intuitive et fonctionnelle, offrant un accès rapide aux fonctionnalités principales.

### 1.1 Formulaire de saisie des informations

La page d'accueil contient un formulaire où l'utilisateur peut entrer les informations nécessaires sur les employés, notamment :

- Identifiant unique (`id`).
- Nom et prénom.
- Adresse e-mail.
- Numéro de téléphone.
- Salaire.
- Rôle (ADMIN ou EMPLOYÉ).
- Poste (par exemple, INGENIEUR, PROFESSEUR, ou DOCTEUR).

Ce formulaire garantit une saisie rapide et organisée des données.

### 1.2 Boutons d'action

-Plusieurs boutons sont disponibles pour interagir avec la base de données et gérer les informations des employés :

- Ajouter : Permet d'insérer un nouvel employé dans la base de données.
- Modifier : Offre la possibilité de mettre à jour les informations d'un employé existant.
- Supprimer : Permet de supprimer un employé de la base de données.
- Afficher : Affiche la liste complète des employés enregistrés, facilitant ainsi leur consultation.

L’objectif de cette page est de fournir une interface conviviale et ergonomique, facilitant la gestion des employés pour l’utilisateur. En intégrant toutes les fonctionnalités essentielles sur une seule page, cette conception améliore l’expérience utilisateur et simplifie les opérations quotidiennes.

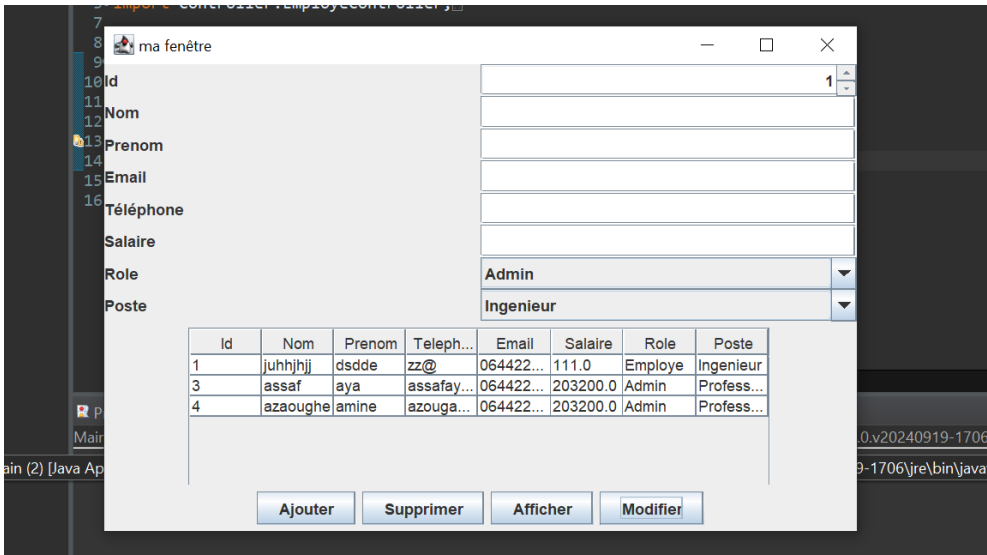


FIGURE 4.1 – page d’accueil

## 2 Gestion des employes

### 2.1 Ajouter un employe

La fonctionnalité Ajouter permet d’insérer un nouvel employé dans la base de données en saisissant ses informations via le formulaire de la page d’accueil. Les informations requises incluent :

Nom et prénom. Adresse e-mail. Numéro de téléphone. Salaire. Rôle (ADMIN ou EMPLOYÉ). Poste (par exemple, INGENIEUR, PROFESSEUR, ou DOCTEUR). Une fois les informations correctement saisies et validées, le système effectue les opérations suivantes :

Ajoute les informations dans la table correspondante de la base de données. Affiche un message de succès confirmant que l’ajout a été effectué avec succès, comme : "employe added successfully"

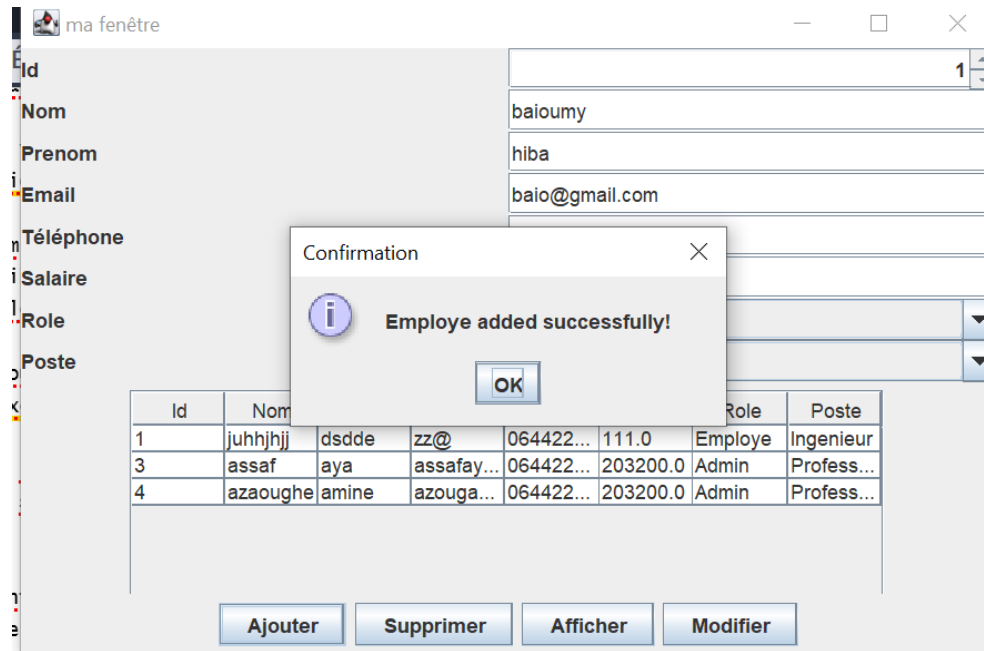


FIGURE 4.2 – Ajouter un employe

## 2.2 Afficher la liste des employes

La fonctionnalité Afficher permet de visualiser la liste complète des employés enregistrés dans la base de données. Cette opération est essentielle pour consulter rapidement les informations des employés sans avoir à effectuer de recherches complexes.

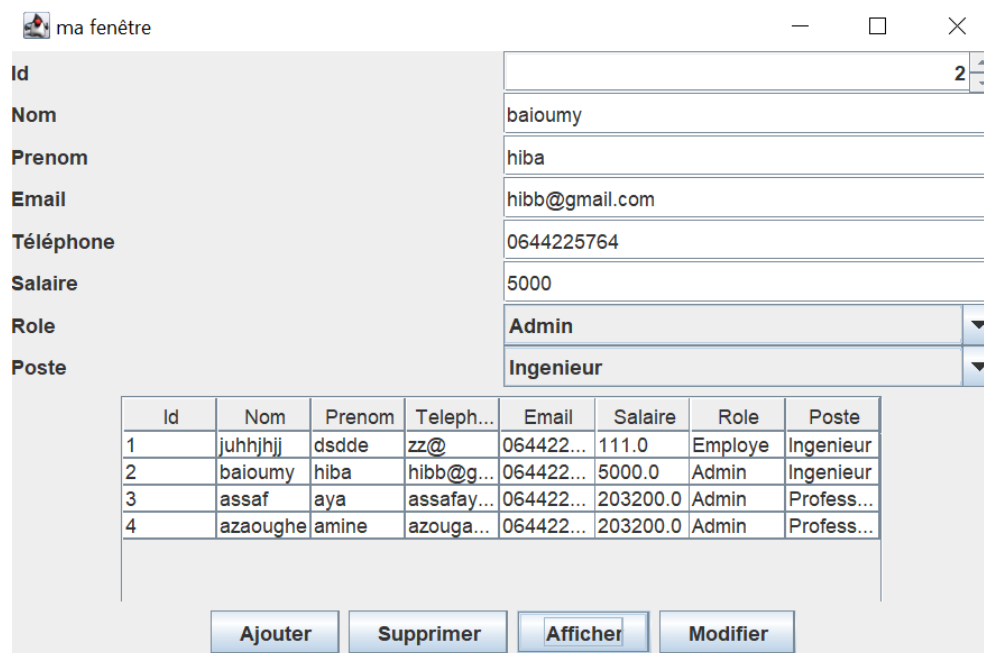


FIGURE 4.3 – Afficher les employes

## 2.3 Modifier un employe

La fonctionnalité Modifier permet de mettre à jour les informations d'un employé existant dans la base de données. Cette option est particulièrement utile pour corriger ou actualiser les données des employés.

L'utilisateur commence par sélectionner la ligne correspondant à l'employé dont il souhaite modifier les informations. Cela peut être fait en cliquant sur une ligne spécifique dans la liste affichée des employés.

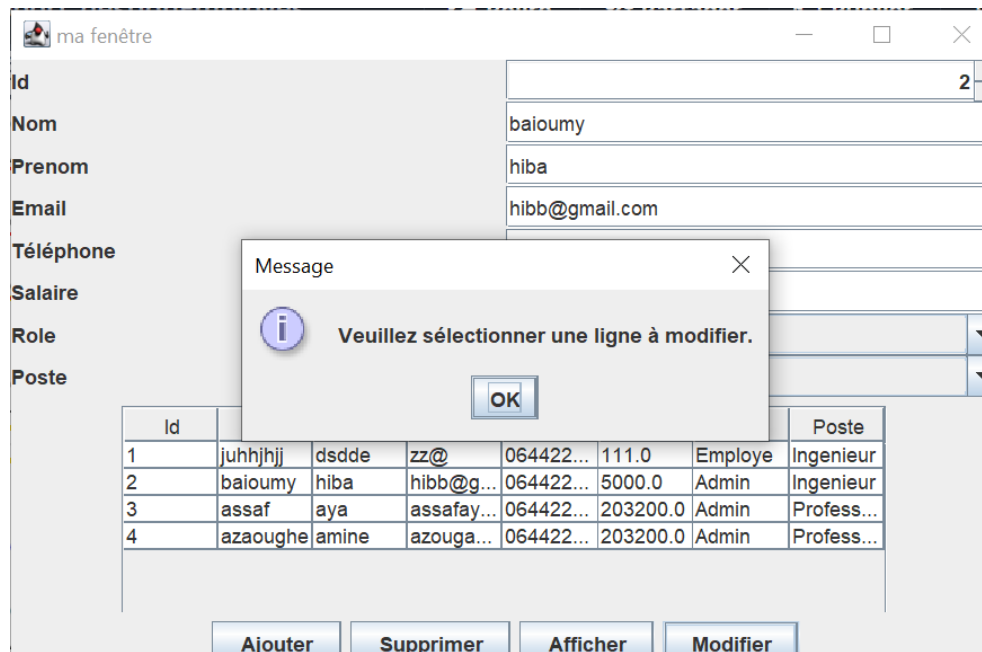


FIGURE 4.4 – Message pour la modification

Après avoir modifié les informations souhaitées, l'utilisateur clique sur un bouton modifier pour enregistrer les changements dans la base de données.

Un message de confirmation, tel que "employe modifier avec succes", est affiché pour informer l'utilisateur que les modifications ont été prises en compte.

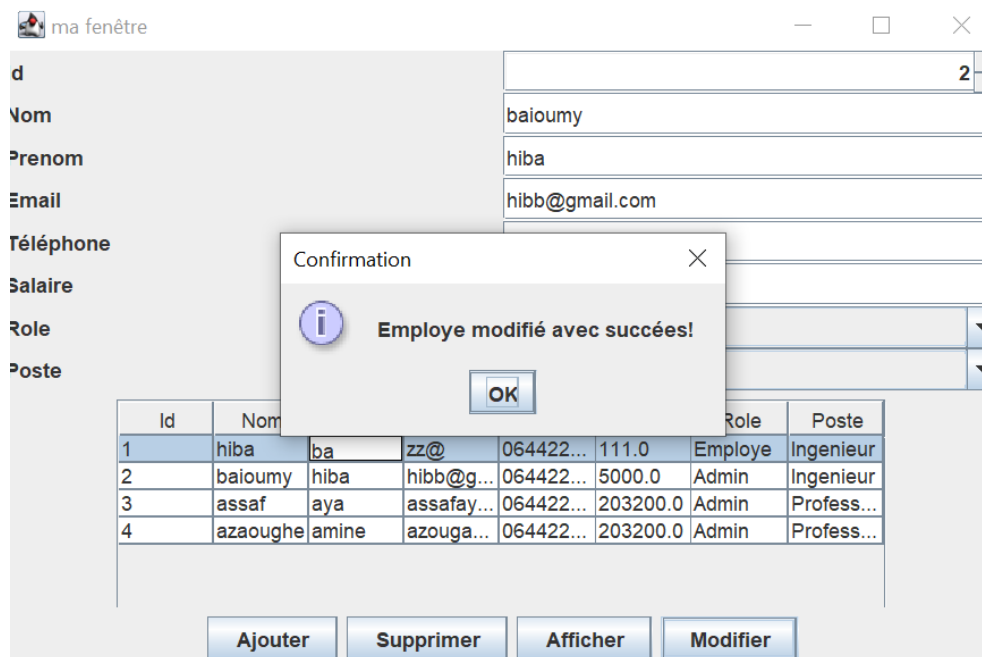


FIGURE 4.5 – Modifier un employe

## 2.4 Supprimer un employe

La fonctionnalité Supprimer permet de retirer un employé de la base de données. Cette opération est irréversible, il est donc important de s'assurer de la sélection correcte avant de procéder à la suppression. L'utilisateur commence par sélectionner la ligne correspondant à l'employé qu'il souhaite supprimer. Cette sélection peut être effectuée en cliquant sur une ligne spécifique dans la liste des employés affichée.

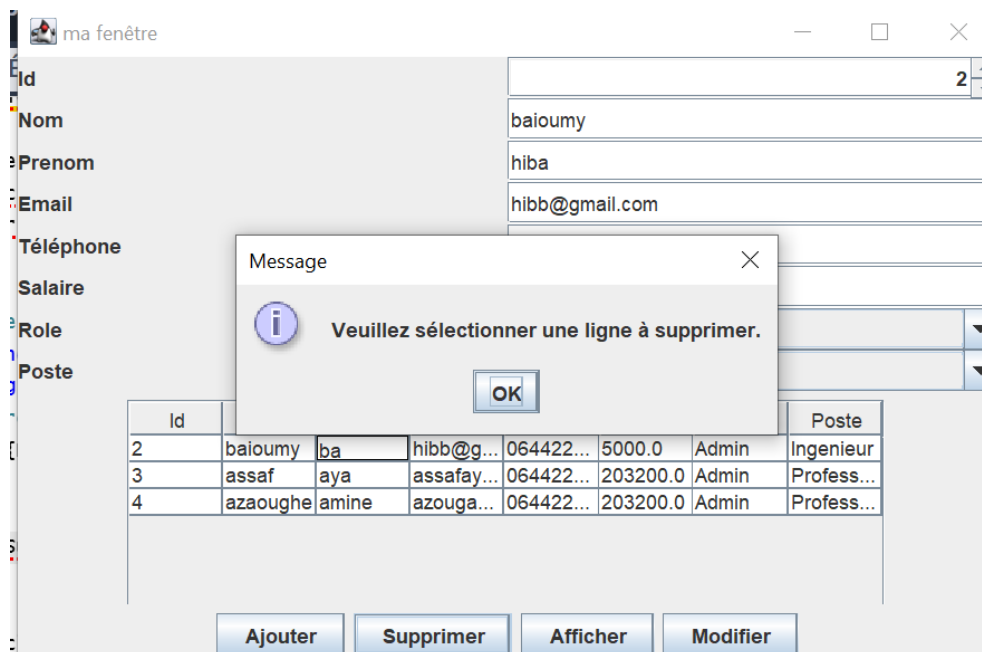


FIGURE 4.6 – Message pour la suppression

les données de l'employé sont supprimées de la base de données. Un message de succès, tel que "L'employé a été supprimé avec succès!", est affiché pour informer l'utilisateur que l'opération a été réalisée avec succès.

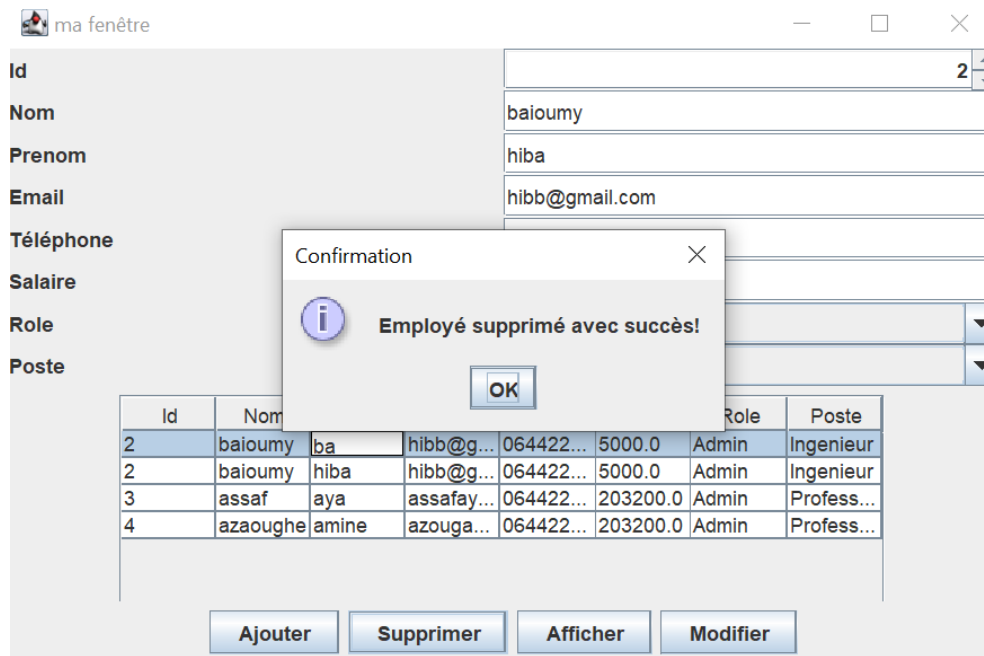


FIGURE 4.7 – Supprimer un employe



# Conclusion Générale

Ce projet a permis de développer une application de gestion des employés offrant des fonctionnalités essentielles telles que l'ajout, la modification, la suppression et la consultation des informations des employés. Grâce à une interface simple et intuitive, l'utilisateur peut facilement interagir avec la base de données, assurant ainsi une gestion efficace des employés.

L'application repose sur une architecture MVC, garantissant une séparation claire des responsabilités entre les différentes parties de l'application, ce qui facilite sa maintenance et son évolutivité. L'utilisation du modèle DAO (Data Access Object) permet une gestion optimale des interactions avec la base de données, offrant ainsi une abstraction des opérations de lecture, écriture, mise à jour et suppression des données.

L'implémentation de l'interface utilisateur a permis de garantir une expérience conviviale pour l'utilisateur, avec des fonctionnalités claires pour l'ajout, la modification, la suppression et l'affichage des employés. Chaque opération est accompagnée de messages de confirmation, assurant ainsi une bonne communication avec l'utilisateur et minimisant les risques d'erreurs.

En conclusion, ce projet a permis d'acquérir des compétences en développement d'applications de gestion, en gestion de bases de données, ainsi qu'en conception d'interfaces utilisateurs simples et efficaces. Il représente un pas important dans le développement d'applications pratiques et professionnelles.

Le code source de l'application est disponible sur GitHub pour consultation et utilisation : [GitHub - Gestion des Employés](#)