

Rapport de Projet

Gestion des Employés



Réalisé par :
AKALAI NOURALLAH

Encadré par :
ILHAM KACHBAL

Date :
9 décembre 2024

Table des matières

Introduction	2
1 Modèle MVC	3
2 Structure du Projet	4
3 Objectifs du Projet	6
4 Interfaces Utilisateur	7
4.1 Interface d'Inscription	7
4.2 Interface de Connexion	8
4.3 Interface de Gestion des Employés	8
4.3.1 Ajouter un Employé	8
4.3.2 Modifier un Employé	8
5 Conclusion	11

Introduction

Dans le cadre des travaux pratiques, ce projet a été réalisé en suivant le modèle architectural MVC (Model-View-Controller). Il s'agit d'une application de gestion des employés qui inclut des fonctionnalités d'inscription, de connexion, et de gestion des employés (ajout, modification, suppression, affichage).

Ce rapport présente les objectifs du projet, la structure adoptée et les interfaces développées.

Chapitre 1

Modèle MVC

Le modèle MVC (Model-View-Controller) est une architecture logicielle qui sépare l'application en trois couches principales :

- **Modèle (Model)** : Représente la logique métier et les données de l'application.
- **Vue (View)** : Gère l'affichage des données et l'interface utilisateur.
- **Contrôleur (Controller)** : Interagit avec le modèle et la vue pour gérer les requêtes des utilisateurs.

Cette séparation permet une meilleure organisation du code, facilitant ainsi la maintenance et l'évolution de l'application.

Chapitre 2

Structure du Projet

La structure de ce projet est organisée selon le modèle architectural MVC (Model-View-Controller) afin de séparer les différentes responsabilités de l'application. Chaque composant a un rôle bien défini, facilitant la gestion et l'évolution du code.

1. ****Le package 'Model'**** : Ce package contient les classes qui représentent la logique métier de l'application, telles que les objets représentant les employés (par exemple, 'Employee.java'), ainsi que les modèles associés qui gèrent les données (par exemple, 'EmployeeModel.java'). Le package 'DAO' (Data Access Object) fait également partie du modèle, en incluant des classes permettant d'interagir avec la base de données (par exemple, 'EmployeeDAOImpl.java').

2. ****Le package 'Controller'**** : Ce package est responsable de la gestion des actions des utilisateurs et de la logique de contrôle. Les classes dans ce package, comme 'EmployeeController.java', communiquent avec le modèle pour récupérer ou modifier les données, puis mettent à jour la vue en conséquence. Elles traitent les demandes des utilisateurs et décident des actions à entreprendre en fonction des événements reçus.

3. ****Le package 'View'**** : Le package 'View' contient les classes responsables de l'interface utilisateur. Ces classes affichent les données provenant du modèle et gèrent l'interaction avec l'utilisateur (par exemple, 'LoginView.java', 'SignUpView.java'). La vue reçoit les données traitées par le contrôleur et les affiche de manière lisible et interactive pour l'utilisateur.

Cette séparation permet de rendre le code plus modulaire, facilitant sa maintenance et son extension. Voici la structure des répertoires de votre projet, montrant la répartition entre les packages 'Model', 'Controller', et 'View' :

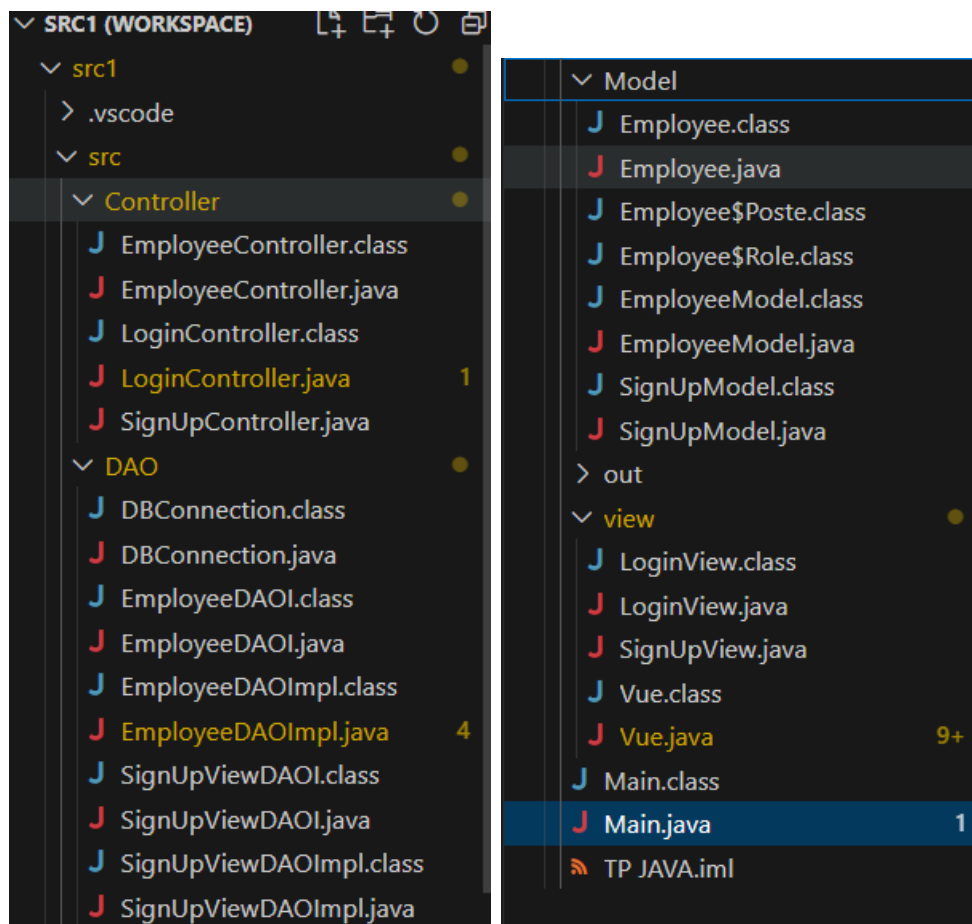


FIGURE 2.1 – Structure des répertoires du projet.

Chapitre 3

Objectifs du Projet

L'objectif principal de ce projet est de développer une application pour la gestion des employés, avec les fonctionnalités suivantes :

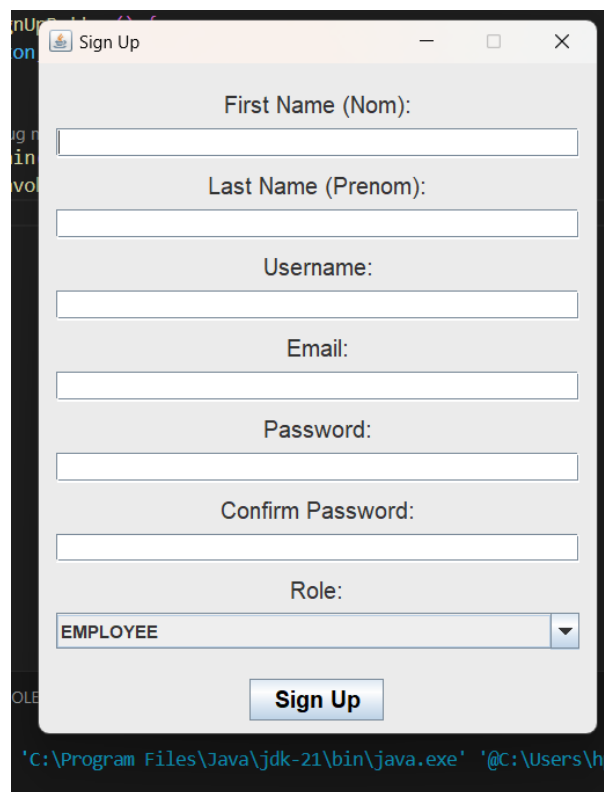
- Inscription des utilisateurs (nom, prénom, nom d'utilisateur, email, mot de passe, rôle).
- Connexion des utilisateurs (nom d'utilisateur et mot de passe).
- Gestion des employés : ajout, modification, suppression, et affichage.

Chapitre 4

Interfaces Utilisateur

4.1 Interface d'Inscription

Voici une capture d'écran de l'interface d'inscription : Pour créer un compte, l'utilisa-



The screenshot shows a Java Swing window titled "Sign Up". The window contains a registration form with the following fields and controls:

- First Name (Nom):** A text input field.
- Last Name (Prenom):** A text input field.
- Username:** A text input field.
- Email:** A text input field.
- Password:** A text input field.
- Confirm Password:** A text input field.
- Role:** A dropdown menu with "EMPLOYEE" selected.
- Sign Up:** A button at the bottom of the form.

The window's title bar includes standard OS controls (minimize, maximize, close). The background of the window is light gray.

FIGURE 4.1 – Interface d'inscription : Cette page permet aux utilisateurs de créer un compte en remplissant les informations nécessaires telles que le nom, prénom, email, mot de passe, etc. Le mot de passe doit contenir au moins 8 caractères.

teur doit remplir tous les champs obligatoires tels que le nom, prénom, nom d'utilisateur, email, mot de passe, confirmer le mot de passe et le rôle (employee ou admin). Il est important de noter que le mot de passe doit contenir au moins 8 caractères pour assurer une sécurité minimale. Une fois tous les champs remplis, l'utilisateur peut cliquer sur "Créer un compte" pour finaliser l'inscription.

4.2 Interface de Connexion

Voici une capture d'écran de l'interface de connexion : Dans l'interface de connexion,

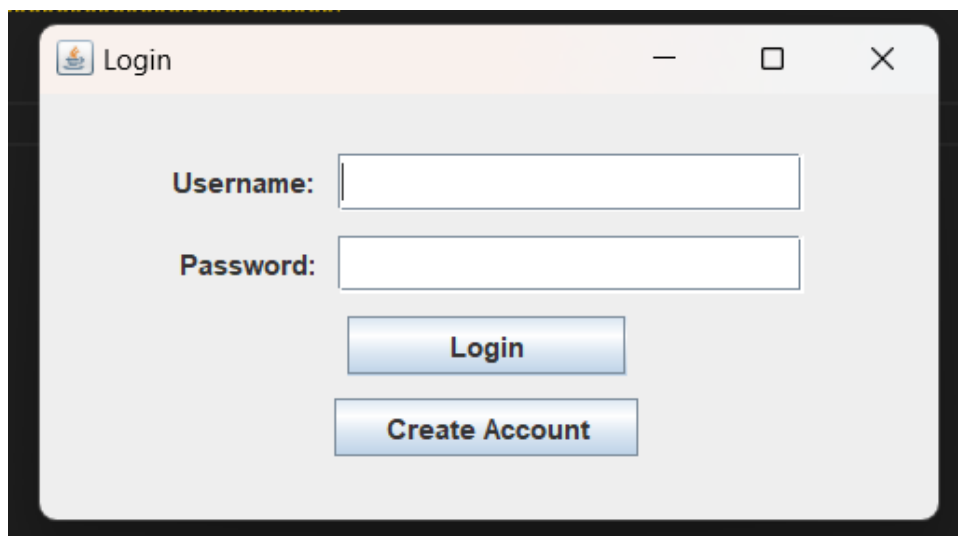


FIGURE 4.2 – Interface de connexion : L'utilisateur doit entrer un nom d'utilisateur et un mot de passe valides pour se connecter à l'application.

l'utilisateur doit entrer un nom d'utilisateur et un mot de passe corrects. Si les informations sont valides, l'utilisateur est redirigé vers l'interface de gestion des employés. Si l'utilisateur n'a pas de compte, il peut créer un compte en cliquant sur le bouton "Créer un compte".

4.3 Interface de Gestion des Employés

4.3.1 Ajouter un Employé

Pour ajouter un employé, l'utilisateur doit remplir un formulaire avec les informations nécessaires telles que le nom, prénom, email, téléphone, salaire et rôle. Une fois le formulaire complété, l'utilisateur clique sur le bouton "Ajouter" pour enregistrer l'employé dans le système.

4.3.2 Modifier un Employé

Pour modifier un employé, l'utilisateur doit sélectionner l'employé dans la liste des employés, puis modifier les informations dans le formulaire. Après avoir effectué les modifications, l'utilisateur clique sur le bouton "Modifier" pour enregistrer les changements.

Gestion des employes

Nom: YASSINE

Prenom: BENHADOU

Email: YASSINE@GMAIL.COM

Telephone: 738928940

Salaire: 10000

Role: ADMIN

Poste: TEAM_LEADER

ID	Nom	Prenom	Tel	Email	Salaire	Poste	Role
10	nour	akalai	7386349	akalai@gmail.com	11100.0	INGENIEUR_ETUDE_ET_DEVELOPPEMENT	EMPLOYEE
12	amine	berrada	969854	AMINE@gmail.com	10000.0	INGENIEUR_ETUDE_ET_DEVELOPPEMENT	EMPLOYEE
15	ADHAM	AKALAI	37293938	ADHAM@gmail.com	5010.0	PILOTE	EMPLOYEE

Success

Employee added successfully!

OK

Ajouter Modifier Supprimer Afficher

Gestion des employes

Nom:

Prenom:

Email:

Telephone:

Salaire:

Role: EMPLOYEE

Poste: INGENIEUR_ETUDE_ET_DEVELOPPEMENT

ID	Nom	Prenom	Tel	Email	Salaire	Poste	Role
10	nour	akalai	7386349	akalai@gmail.com	11100.0	INGENIEUR_ETUDE_ET_DEVELOPPEMENT	EMPLOYEE
12	amine	berrada	969854	AMINE@gmail.com	10000.0	INGENIEUR_ETUDE_ET_DEVELOPPEMENT	EMPLOYEE
15	ADHAM	AKALAI	37293938	ADHAM@gmail.com	5010.0	PILOTE	EMPLOYEE
17	YASSINE	BENHADOU	738928940	YASSINE@GMAIL.COM	10000.0	TEAM_LEADER	ADMIN

Ajouter Modifier Supprimer Afficher

FIGURE 4.3 – Ajout d'un employé : L'utilisateur peut ajouter un employé en remplissant un formulaire avec les informations nécessaires telles que le nom, prénom, email, téléphone, salaire et rôle.

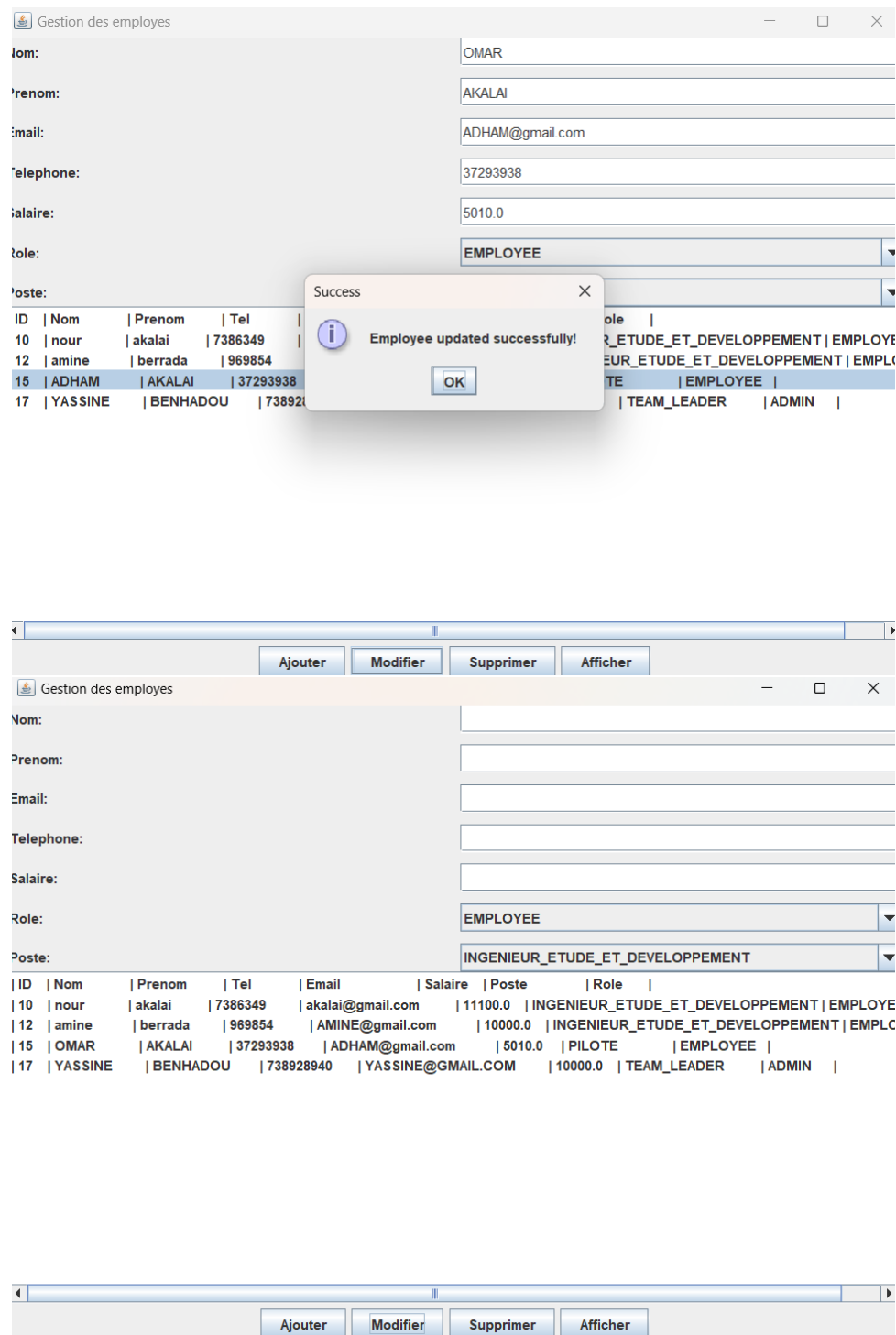


FIGURE 4.4 – Modification d'un employé : Pour modifier un employé, l'utilisateur doit sélectionner un employé et modifier ses informations, puis cliquer sur "Modifier" pour enregistrer les changements.

Chapitre 5

Conclusion

Ce projet permet de gérer les employés dans un environnement simple mais efficace. Grâce à l'utilisation du modèle MVC, l'application est bien structurée et facile à maintenir. Les interfaces utilisateur sont conviviales et répondent aux besoins de l'application. Ce projet peut être étendu pour inclure d'autres fonctionnalités telles que la gestion des départements ou des rôles.