

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

Rapport du TP

Développement d'une application de gestion des employés(MVC-DAO et java SWING)

Réalisé par :

M.AZAOUAGHE MOHAMED AMINE

Encadré par :

M. EL ABDELLAOUI SAID
Mme. KACHBAL ILHAM

ANNÉE UNIVERSITAIRE : 2023/2024

Table des matières

1	Introduction	4
2	Présentation du projet	5
1	Problématique	5
2	Solution proposée	5
3	Processus de développement	6
4	Conclusion	6
3	Conception	7
1	Base de données	7
2	La création de la base de données et de la table des employés	7
3	Technologies et outils utilisés	7
4	Réalisation	10
1	Présentation des Fonctionnalités de l'Interface	10
2	L'interface graphique	10
3	Ajouter un Employé	11
4	Lister les Employés	11
5	Modifier un employé	12
6	Supprimer un employé	12
5	Conclusion	13

Table des figures

3.1	Eclipse	8
3.2	PostgreSQL	8
3.3	JavaSwing	8
3.4	DAO	9
4.1	l'interface graphique	10
4.2	Ajouter un employé	11
4.3	Afficher la liste des employés	11
4.4	Modifier un employé	12
4.5	Supprimer un employé	12

Introduction

La gestion des employés constitue un enjeu fondamental pour toute organisation désireuse d'améliorer son efficacité et d'optimiser sa gestion interne. Dans cette optique, nous avons conçu une application de bureau dédiée à la gestion des employés, en intégrant 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 PostgreSQL, cette application permet aux administrateurs de centraliser et d'automatiser les tâches liées à la gestion des employés. Parmi ses fonctionnalités principales, on retrouve l'ajout, la modification, la suppression et la consultation des informations des employés.

L'architecture MVC, utilisée comme fondement de cette application, garantit une séparation rigoureuse des responsabilités. Le Modèle gère les données des employés et leurs règles métiers, la Vue fournit une interface utilisateur intuitive, et le Contrôleur coordonne les interactions entre les deux. En parallèle, l'adoption du design pattern DAO permet de structurer l'accès aux données et de faciliter la maintenance et l'extensibilité du système. L'interface utilisateur a été pensée pour être simple et ergonomique, afin de permettre une gestion fluide et sécurisée des informations.

L'objectif principal de ce projet est de proposer une solution fiable, modulaire et facile à maintenir, qui répond aux besoins actuels tout en restant évolutive pour s'adapter aux futurs défis de l'organisation. En explorant les possibilités d'amélioration continue, cette application s'inscrit comme un outil clé dans la transformation numérique des processus de gestion des employés, garantissant ainsi une meilleure performance et une organisation optimale.

Présentation du projet

Ce travail pratique a pour objectif de développer une application de gestion des employés en utilisant les concepts du modèle MVC (Modèle-Vue-Contrôleur), du DAO (Data Access Object) et de Java Swing pour l'interface graphique. L'application vise à permettre aux utilisateurs de gérer les informations liées aux employés d'une entreprise, telles que l'ajout, la modification, et la suppression des données, tout en offrant une interface utilisateur simple et intuitive.

1 Problématique

Dans le cadre de la gestion des employés, il est crucial de disposer d'un système performant pour organiser, stocker et accéder aux informations liées aux employés. Une gestion mal structurée des données peut rapidement engendrer des inefficacités, des erreurs, et une perte considérable de temps pour l'organisation. Afin de répondre à ce besoin, l'objectif principal de ce projet est de développer une application qui centralise la gestion des employés en s'appuyant sur une architecture MVC (Modèle-Vue-Contrôleur) et en intégrant le design pattern DAO (Data Access Object). L'architecture MVC permet une séparation claire des responsabilités, en structurant le développement autour du Modèle (données et logique métier), de la Vue (interface utilisateur) et du Contrôleur (interactions). Par ailleurs, l'utilisation du DAO facilite la maintenance en isolant la logique d'accès à la base de données. Ce projet vise ainsi à fournir une solution efficace et modulaire pour une gestion optimisée et évolutive des employés.

2 Solution proposée

L'application repose sur l'architecture MVC, offrant une séparation claire des responsabilités. Le Modèle (Model) gère les données des employés via des classes Java dédiées, tandis que la Vue (View), réalisée avec Java Swing, fournit une interface utilisateur intuitive pour visualiser, modifier et supprimer les informations. Le Contrôleur (Controller) coordonne les interactions utilisateur, interroge la base de données via le DAO et met à jour la vue selon les actions réalisées. Le DAO (Data Access Object) abstrait l'accès à la base de données, simplifiant les opérations CRUD (Créer, Lire, Mettre à jour, Supprimer). Cette organisation garantit une application robuste et facile à maintenir.

3 Processus de développement

Analyse des besoins

- Identification des fonctionnalités principales de l'application.
- Définition des interactions utilisateur avec la base de données (ajouter, modifier, supprimer, afficher).

Conception de l'architecture

- Choix de l'architecture MVC pour séparer la logique métier de l'interface graphique.
- Conception de la base de données et des classes Java correspondant aux entités d'employés.

Développement

- Création des classes Java pour gérer les données des employés.
- Développement de l'interface graphique avec Java Swing.
- Implémentation du DAO pour accéder aux données dans la base de données.

Tests et validation

- Test des fonctionnalités CRUD de l'application.
- Vérification de l'intégration de la base de données avec l'interface graphique.

4 Conclusion

Ce projet a permis de créer une application de gestion des employés en appliquant des concepts avancés de programmation Java, tels que l'architecture MVC et le pattern DAO. L'application, entièrement fonctionnelle, offre une interface graphique claire et facile à utiliser, facilitant la gestion des informations des employés. Ce travail m'a permis d'approfondir mes connaissances des architectures logicielles et d'acquérir des compétences pratiques sur les bonnes pratiques de développement en Java.

Conception

1 Base de données

Le développement de l'application a commencé par la création de la base de données et de la table `employees`, qui constitue un élément clé pour la gestion efficace des informations des employés dans le système. Cette base de données garantit un stockage organisé des données et simplifie les interactions avec les diverses fonctionnalités de l'application.

2 La création de la base de données et de la table des employés

La table `employees`, implémentée dans une base de données PostgreSQL, centralise les informations essentielles concernant les employés, telles que leur identifiant unique, nom, prénom, email, numéro de téléphone, salaire, rôle et poste. Cette structure a été conçue pour répondre aux besoins spécifiques du système tout en assurant une gestion efficace des données.

- `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.
- `phone` : Numéro de téléphone de l'employé.
- `salaire` : Salaire de l'employé.
- `role` : Rôle de l'employé (ADMIN ou EMPLOYEE).
- `poste` : Poste de l'employé (INGENIEURE_ET_DEVELOPPEMENT, TEAM_LEADER, ou PILOTE).

3 Technologies et outils utilisés

Pour le développement et la gestion du projet, plusieurs outils ont été mobilisés afin d'assurer une efficacité optimale. Voici les principaux :

Eclipse est également un environnement de développement intégré (IDE, pour Integrated Development Environment) très populaire pour la programmation. Il est principalement utilisé pour le développement en Java, mais prend en charge de nombreux autres langages de programmation via des plugins. Eclipse offre des outils puissants pour la rédaction, le débogage et la gestion de projets de code. Il est open-source et peut être utilisé pour le développement d'applications de bureau, web, et mobiles. PostgreSQL est un système de gestion de base



FIGURE 3.1 – Eclipse

de données relationnelle (SGBDR) open-source, puissant et très populaire. Il est utilisé pour stocker et gérer des données de manière structurée en suivant le modèle relationnel, c'est-à-dire en organisant les données sous forme de tables reliées entre elles par des clés primaires et étrangères. Java Swing est une bibliothèque graphique



FIGURE 3.2 – PostgreSQL

pour la création d'interfaces utilisateur (UI) en Java. Elle fait partie de la bibliothèque Java Foundation Classes (JFC) et fournit un ensemble de composants graphiques qui permettent de construire des applications de bureau avec des interfaces graphiques complexes.



FIGURE 3.3 – JavaSwing

un DAO est une classe ou un objet qui fournit une interface abstraite pour accéder aux données dans une base de données ou un autre système de stockage. Il sert d'intermédiaire entre l'application et la source de données, permettant ainsi de centraliser et de simplifier les opérations d'accès aux données.



FIGURE 3.4 – DAO

Réalisation

1 Présentation des Fonctionnalités de l'Interface

Dans ce chapitre, nous allons présenter les fonctionnalités principales de l'application, qui se focalise principalement sur la gestion des employés via une interface conviviale et facile à utiliser. Les actions principales proposées sont l'ajout, la modification, la suppression et la consultation des informations des employés.

2 L'interface graphique

La page d'accueil de l'application propose une interface claire et pratique, avec des boutons d'action facilitant la gestion des employés pour les administrateurs. Les principales fonctionnalités disponibles sont les suivantes :

- **Ajouter** : Permet d'ajouter de nouveaux employés à la base de données en renseignant leurs informations personnelles.
- **Lister** : Affiche la liste complète des employés enregistrés dans le système.
- **Modifier** : Permet de mettre à jour les informations d'un employé existant.
- **Supprimer** : Permet de retirer un employé du système.

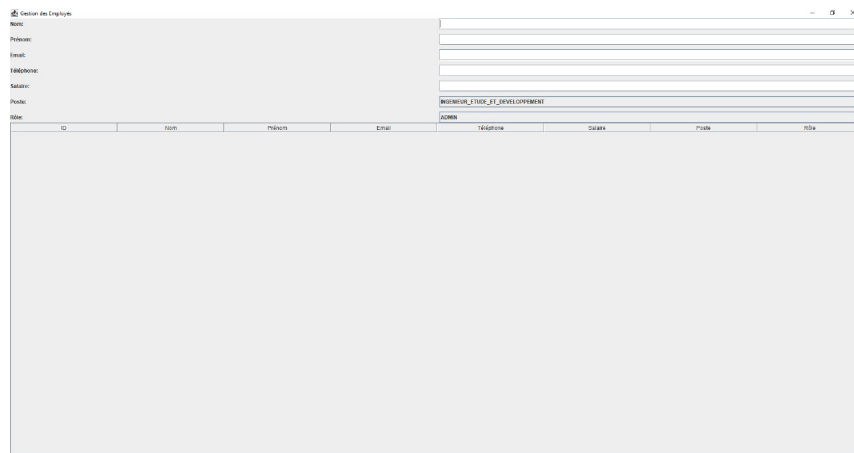


FIGURE 4.1 – l'interface graphique

3 Ajouter un Employé

Le bouton **Ajouter** permet à l’administrateur d’enregistrer un nouvel employé. Un formulaire s’affiche, offrant la possibilité de saisir les informations nécessaires, telles que le nom, le prénom, l’adresse, etc.

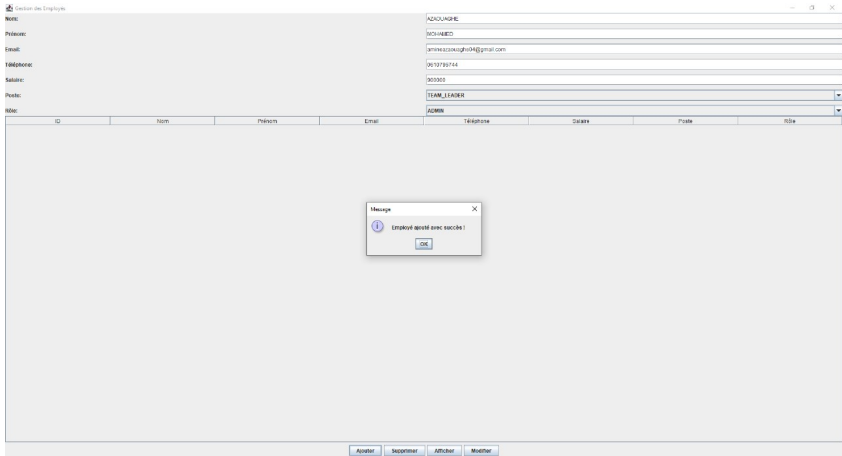


FIGURE 4.2 – Ajoter un employé

4 Lister les Employés

Le bouton **Afficher** permet d’afficher la liste des employés enregistrés dans le système. Cette liste présente les informations de chaque employé sous forme de tableau, avec la possibilité d’accéder aux options **Modifier** et **Supprimer** pour chaque entrée.

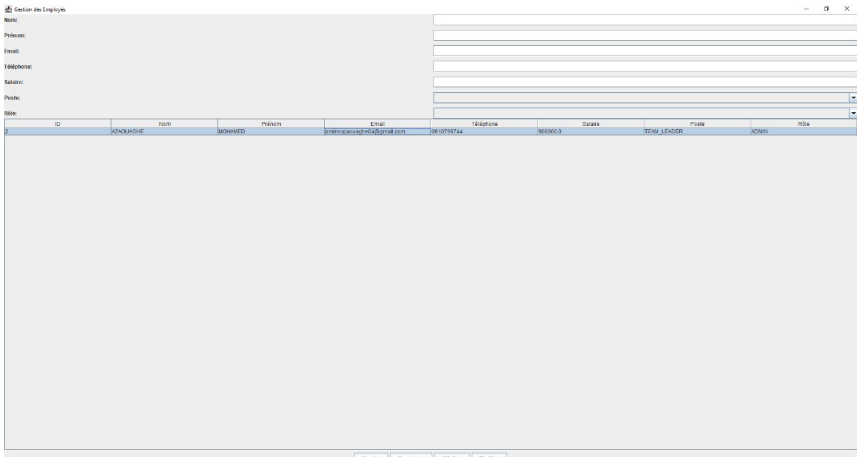


FIGURE 4.3 – Afficher la liste des employés

5 Modifier un employé

Lorsqu'un employé est sélectionné via un événement de souris, l'option **Modifier** permet à l'administrateur de mettre à jour les informations de cet employé. Cela peut concerner des éléments tels que le nom, l'email, le téléphone, le salaire, ou le rôle. Après avoir effectué les modifications, l'administrateur soumet le formulaire, et les données sont ensuite enregistrées dans la base de données.

FIGURE 4.4 – Modifier un employé

6 Supprimer un employé

Le bouton **Supprimer** permet de retirer un employé du système. L'administrateur sélectionne l'ID de l'employé dans la liste. Une confirmation est requise avant de procéder à la suppression définitive de l'employé.

FIGURE 4.5 – Supprimer un employé

Conclusion

La réalisation de ce travail pratique a été une expérience extrêmement enrichissante qui m'a permis de développer de nouvelles compétences et d'approfondir mes connaissances en développement d'applications. Ce projet m'a offert l'opportunité de mettre en pratique les concepts théoriques que j'ai étudiés durant mes cours, tout en me confrontant à des défis concrets liés à la création d'un système de gestion performant. Travailler sur cette application m'a permis de mieux comprendre les différentes étapes du processus de conception, de la planification initiale à l'implémentation, jusqu'à la mise en œuvre des fonctionnalités essentielles.

L'objectif principal du projet était de concevoir une application de gestion des employés, avec des fonctionnalités permettant l'ajout, la modification, la suppression et la consultation des données. En réalisant cette application, j'ai non seulement renforcé mes compétences en programmation Java, mais j'ai aussi appris à gérer des bases de données, à manipuler des interfaces graphiques et à intégrer des éléments interactifs qui améliorent l'expérience utilisateur. Ces compétences sont essentielles pour le développement d'applications modernes et robustes, et ce travail m'a permis de mieux comprendre comment organiser et structurer une application de manière optimale.

Pour concevoir l'application, j'ai utilisé une architecture claire et bien définie, qui permet une gestion efficace des données et une interaction fluide avec l'interface utilisateur. La mise en œuvre des différentes fonctionnalités, telles que l'ajout ou la modification des employés, a été pensée de manière intuitive pour garantir une expérience utilisateur simple et agréable. L'ergonomie de l'application a été une priorité tout au long de son développement, car une interface bien conçue est essentielle pour assurer l'adoption et la facilité d'utilisation d'un outil.

Ce travail m'a également permis de me familiariser avec l'utilisation d'outils et technologies importants, tels que MySQL pour la gestion de la base de données et Visual Studio Code pour le développement. Ces outils sont largement utilisés dans l'industrie et m'ont permis d'acquérir des compétences pratiques directement applicables dans des projets futurs. La gestion des données et leur synchronisation avec l'interface utilisateur ont représenté un défi majeur, mais ces obstacles ont également été l'occasion de développer des solutions efficaces pour résoudre des problèmes complexes. La validation des entrées utilisateur, par exemple, a été un aspect essentiel pour garantir l'intégrité des données dans le système.

Les défis techniques rencontrés m'ont permis de développer une rigueur accrue dans le processus de développement. J'ai appris à tester minutieusement chaque fonctionnalité et à valider les différentes étapes de la réalisation pour m'assurer de la qualité du produit final. Cette approche systématique m'a également appris l'importance de la collaboration avec des outils modernes et la nécessité de suivre les bonnes pratiques de développement. À travers ce projet, j'ai compris que chaque étape du développement logiciel doit être réalisée avec soin, de la planification initiale aux tests finaux.

En conclusion, ce travail pratique a été une excellente occasion de renforcer mes compétences techniques, notamment en programmation Java, gestion de base de données et développement d'interfaces utilisateur. Il m'a aussi permis d'améliorer ma capacité à organiser un projet, à utiliser des outils professionnels et à résoudre des problèmes techniques. Les connaissances et compétences acquises au cours de ce TP constitueront sans aucun doute des atouts précieux pour mes futurs projets dans le domaine du développement logiciel. Cette expérience m'a donné des bases solides pour aborder avec confiance et méthode des problématiques similaires dans un contexte professionnel, et je suis convaincu que les enseignements tirés de ce travail me seront utiles tout au long de ma carrière.