



المدرسة العليا  
للتكنولوجيا-أسفي  
ECOLE SUPÉRIEURE  
DE TECHNOLOGIE -SAFI



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

Rapport Du TP :Gestion des employés  
GI 2ème année

---

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

---

*Réalisé par :*

EL COURT Zakaria

*Enseigné par :*

M.EL ABDELLAOUI Said

DATE DE LA SÉANCE : 6 DÉCEMBRE

ANNÉE UNIVERSITAIRE : 2024/2025

## **Introduction**

Dans le cadre de ce projet, nous avons développé une application de bureau pour la gestion des employés, en appliquant les principes de l'architecture MVC (Modèle-Vue-Contrôleur) et en utilisant le design pattern DAO (Data Access Object).

Cette application, conçue avec Java Swing et MySQL, offre une interface intuitive permettant aux administrateurs de gérer efficacement les informations des employés. Elle met en œuvre des concepts clés du génie logiciel, tels que la séparation des responsabilités et l'abstraction de l'accès aux données, pour garantir une solution robuste, maintenable et adaptée aux besoins des utilisateurs.

Ce rapport détaille les étapes de conception et d'implémentation de l'application, en soulignant les défis rencontrés et les solutions apportées.

# Table des matières

<b>Introduction</b>	<b>6</b>
<b>1 Contexte du TP</b>	<b>7</b>
1.1 Présentation du TP . . . . .	7
1.1.1 Problématique . . . . .	7
1.1.2 Solution proposée . . . . .	7
1.2 Processus de développement . . . . .	8
<b>2 Conception</b>	<b>10</b>
2.1 Base de données . . . . .	10
2.1.1 Création de la base de données et de la table <code>employees</code> . . . . .	10
2.2 Environnement de travail . . . . .	11
2.2.1 Outils utilisés . . . . .	11
2.2.2 Approches et concepts appliqués . . . . .	12
<b>3 Réalisation</b>	<b>14</b>
3.1 Présentation des Fonctionnalités de l'Interface . . . . .	14
3.2 Page d'Accueil . . . . .	14
3.3 Gestion des Employés . . . . .	15
3.3.1 Ajouter un Employé . . . . .	15
3.3.2 Lister les Employés . . . . .	16
3.3.3 Modifier un Employé . . . . .	17
3.3.4 Supprimer un Employé . . . . .	17
3.4 Conclusion . . . . .	18

<b>Conclusion</b>	<b>18</b>
<b>Webographie</b>	<b>20</b>

# Liste des figures

2.1	Schéma de la base de données . . . . .	11
2.2	Visual Studio Code . . . . .	11
2.3	XAMPP . . . . .	11
2.4	phpMyAdmin . . . . .	12
2.5	Data Access Object (DAO) . . . . .	12
2.6	Swing - Framework Java pour l'interface graphique . . . . .	12
2.7	Architecture MVC (Model-View-Controller) . . . . .	13
3.1	fenetre de l'application . . . . .	15
3.2	l'ajout d'un employé . . . . .	16
3.3	lister les employés . . . . .	16
3.4	modifier un employé . . . . .	17
3.5	supprimer un employé . . . . .	18

# Introduction

La gestion des employés est un élément clé pour toute organisation souhaitant améliorer son efficacité et sa gestion interne. Dans le cadre de ce projet, nous avons développé une application de bureau dédiée à la gestion des employés, en appliquant les principes de l'architecture MVC (Modèle-Vue-Contrôleur) et le design pattern DAO (Data Access Object).

Cette application, réalisée en Java avec l'interface Swing et une base de données MySQL, permet aux administrateurs de gérer de manière centralisée les informations des employés, tout en assurant une interface intuitive et facile à utiliser. L'objectif principal de l'application est d'automatiser les tâches courantes de gestion des employés telles que l'ajout, la modification, la suppression et la consultation des données des employés.

L'application repose sur l'architecture MVC pour garantir une séparation claire des responsabilités, avec un **Modèle** qui gère les données des employés, une **Vue** qui affiche l'interface utilisateur, et un **Contrôleur** qui gère les interactions. Le DAO permet de séparer la logique d'accès aux données, facilitant ainsi la maintenance et l'extensibilité de l'application.

Les principales fonctionnalités comprennent la gestion complète des employés, avec des options telles que la création, la mise à jour et la suppression d'employés, ainsi que la consultation des informations. L'interface est simple et fluide, permettant aux administrateurs de gérer efficacement les informations des employés tout en garantissant la sécurité et la fiabilité des données.

À travers ce projet, l'objectif est de fournir une solution fonctionnelle, modulaire et facile à maintenir, permettant à l'organisation de gérer ses employés de manière optimale. Nous explorerons également les possibilités d'amélioration et d'adaptation de cette application pour répondre aux futurs besoins de l'organisation.

# Chapitre 1

## Contexte du TP

### Introduction

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.

### 1.1 Présentation du TP

#### 1.1.1 Problématique

Dans le cadre de la gestion des employés, il est essentiel de disposer d'un système efficace pour organiser, stocker et accéder aux données des employés. Sans une architecture bien structurée, la gestion des données devient désorganisée, ce qui peut entraîner des erreurs et des pertes de temps. L'objectif principal de ce TP est de créer une application qui permette de centraliser la gestion des employés, en appliquant une architecture MVC, et en utilisant le DAO pour séparer la logique métier de la gestion de la base de données.

#### 1.1.2 Solution proposée

L'application développée repose sur les principes de l'architecture MVC, où :

- **Le Modèle (Model)** : Représente les données des employés, avec une classe Java pour gérer les informations des employés.

- **La Vue (View)** : Fournit l'interface utilisateur avec Java Swing, permettant à l'utilisateur de visualiser, modifier et supprimer les informations des employés.
- **Le Contrôleur (Controller)** : Gère les interactions de l'utilisateur, interroge la base de données via le DAO, et met à jour la vue en fonction des actions de l'utilisateur.
- **DAO (Data Access Object)** : Permet d'abstraire l'accès aux données de la base de données, facilitant les opérations CRUD (Créer, Lire, Mettre à jour, Supprimer) sur les employés.

## 1.2 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.



## Conclusion

Ce travail pratique a permis de développer une application de gestion des employés en utilisant les concepts avancés de Java, notamment le modèle MVC et le pattern DAO. L'application est entièrement fonctionnelle, avec une interface graphique simple et intuitive, permettant de gérer efficacement les informations des employés. Ce projet a contribué à renforcer ma compréhension des architectures logicielles et des bonnes pratiques en développement Java.

# Chapitre 2

## Conception

### 2.1 Base de données

Le développement de l'application a débuté par la création de la base de données et de la table `employes`, qui joue un rôle central en permettant de gérer efficacement les informations relatives aux employés du système. Cette base de données assure un stockage structuré et facilite les interactions avec les différentes fonctionnalités de l'application.

#### 2.1.1 Création de la base de données et de la table `employes`

La table `employes`, implémentée dans une base de données MySQL, regroupe des informations clés sur les employés : leur identifiant unique, leur nom, prénom, email, numéro de téléphone, poste, rôle et salaire. La structure de cette table a été pensée pour répondre aux besoins fonctionnels du système.

- **Table `employes`** : Cette table contient les champs suivants :
  - `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 EMPLOYE).
  - `poste` : Poste de l'employé (INGENIEURE\_ETUDE\_ET\_DEVELOPPEMENT, TEAM\_LEADER, ou PILOTE).

Table	Action	Lignes	Type	Interclassement	Taille	Perte
employe	Parcourir Structure Rechercher Insérer Vider Supprimer	7	InnoDB	utf8mb4_general_ci	32,0 kio	-
1 table	Somme	7	InnoDB	utf8mb4_general_ci	32,0 kio	0 o

FIGURE 2.1 – Schéma de la base de données

## 2.2 Environnement de travail

### 2.2.1 Outils utilisés

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

- **Visual Studio Code** : Utilisé comme éditeur principal pour écrire et gérer le code Java, grâce à ses fonctionnalités avancées et ses extensions.



FIGURE 2.2 – Visual Studio Code

- **XAMPP** : Employé pour la configuration d’un environnement serveur local.



FIGURE 2.3 – XAMPP

- **phpMyAdmin** : Utilisé pour gérer la base de données MySQL via une interface graphique web.



FIGURE 2.4 – phpMyAdmin

### 2.2.2 Approches et concepts appliqués

Le développement a été structuré en utilisant l'architecture **MVC** (Model-View-Controller) pour séparer les responsabilités entre la gestion des données, l'interface utilisateur, et la logique métier. Les concepts et outils suivants ont été intégrés :

- **Model** : Gestion des données et de la logique métier, utilisant le design pattern **DAO** (Data Access Object) pour une interaction optimisée avec la base de données.



FIGURE 2.5 – Data Access Object (DAO)

- **View** : Création de l'interface utilisateur via **Swing**, qui offre une gamme de composants graphiques pour concevoir une interface interactive et intuitive.

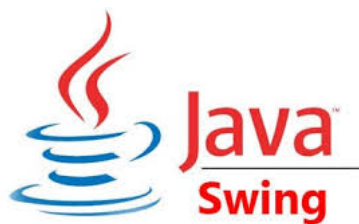


FIGURE 2.6 – Swing - Framework Java pour l'interface graphique

- **Controller** : Gestion des interactions entre l'utilisateur et les données, assurant une communication fluide entre la vue et le modèle.

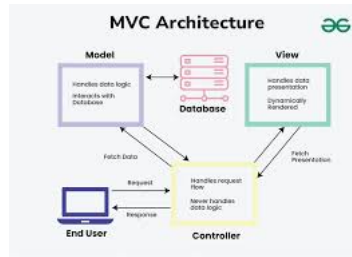


FIGURE 2.7 – Architecture MVC (Model-View-Controller)

L'intégration de **DAO** a permis une séparation claire entre la logique métier et l'accès aux données, renforçant la maintenabilité du code. De même, **Swing** a facilité la création d'une interface utilisateur ergonomique tout en respectant les principes de modularité de l'architecture **MVC**.

## Conclusion

Cette première étape a permis d'établir une base solide pour le projet, tant sur le plan de la structure de la base de données que sur celui des outils et des approches utilisés. L'architecture claire et les outils performants ont contribué à un développement fluide et organisé.

# Chapitre 3

## Réalisation

### 3.1 Présentation des Fonctionnalités de l'Interface

Dans ce chapitre, nous allons présenter les principales fonctionnalités de l'application, qui se concentre exclusivement sur la gestion des employés à travers une interface simple et intuitive. Les actions essentielles incluent l'ajout, la modification, la suppression et la consultation des employés.

### 3.2 Page d'Accueil

La page d'accueil de l'application offre une interface épurée et fonctionnelle, avec des boutons d'action permettant aux administrateurs de gérer les employés de manière efficace. Les fonctionnalités disponibles sont les suivantes :

- **Ajouter** : Permet d'ajouter de nouveaux employés à la base de données en entrant leurs informations de base.
- **Lister** : Affiche la liste complète des employés enregistrés dans le système.
- **Modifier** : Permet de modifier les informations d'un employé déjà existant.
- **Supprimer** : Permet de supprimer un employé du système.

The screenshot shows a window titled "Gestion des Employés" with a standard Windows-style title bar (minimize, maximize, close buttons). The window contains a form with the following fields and controls:

- Nom:** A text input field.
- Prénom:** A text input field.
- Email:** A text input field.
- Téléphone:** A text input field.
- Salaire:** A text input field.
- Rôle:** A dropdown menu with "Admin" selected.
- Poste:** A dropdown menu with "INGENIEURE\_ETUDE\_ET\_DEVELOPPEMENT" selected.

Below the form fields is a large, empty rectangular area, likely intended for displaying a list of employees or additional details.

FIGURE 3.1 – fenetre de l'application

Ces boutons sont clairement identifiables et offrent une navigation facile pour l'administrateur.

### 3.3 Gestion des Employés

L'interface de gestion des employés permet à l'administrateur d'effectuer diverses actions liées à la gestion des informations des employés.

#### 3.3.1 Ajouter un Employé

Le bouton "Ajouter" permet à l'administrateur d'ajouter un nouvel employé. Un formulaire est affiché, permettant de saisir les informations essentielles, telles que le nom, le prénom, l'email, le téléphone, le salaire, ainsi que le rôle et le poste de l'employé.

**Gestion des Employés**

Nom: el abdellaoui

Prénom: said

Email: s.elabdellaoui@uca.ac.ma

Téléphone: 0606060606

Salaire: 120000

Rôle: Admin

Poste: TEAM\_LEADER

Message: Employé ajouté avec succès.

Ajouter Afficher Supprimer Modifier

FIGURE 3.2 – l'ajout d'un employé

### 3.3.2 Lister les Employés

Le bouton "Lister" 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.

**Gestion des Employés**

Nom:

Prénom:

Email:

Téléphone:

Salaire:

Rôle: Admin

Poste: INGENIEURE\_ETUDE\_ET\_DEVELOPPEMENT

ID	Nom	Prénom	Email	Téléphone	Salaire	Rôle	Poste
1	zakaria	el	elcortz@gmail.com	0606197011	12000.0	EMPLOYE	TEAM_LEADER
2	aicha	danone	ok@gmail.com	0808080808	1.0	ADMIN	INGENIEURE_ETUDE_E...
4	IMRAN	up	kokk@gmail.com	0808080807	123.0	ADMIN	PILOTE
6	hiba	baillade	elcort@gmail.com	09090909	12345.0	ADMIN	PILOTE
8	test	test	test@gmail.com	1234567890	18000.0	EMPLOYE	INGENIEURE_ETUDE_E...
10	zakzak	spam	tzak@gmail.com	5647890321	180000.0	EMPLOYE	PILOTE
13	ikram	bairouk	ikram@gmail.com	1212121212	8798798.0	EMPLOYE	PILOTE
14	el abdellaoui	said	s.elabdellaoui@uca.ac.ma	0606060606	120000.0	ADMIN	TEAM_LEADER

Ajouter Afficher Supprimer Modifier

FIGURE 3.3 – lister les employés



### 3.3.3 Modifier un Employé

Lorsqu'un employé est sélectionné par un mouse event, l'option "Modifier" permet à l'administrateur de mettre à jour les informations de l'employé. Cela peut inclure la modification du nom, de l'email, du téléphone, du salaire, ou du rôle. Une fois les modifications apportées, l'administrateur soumet le formulaire et les données sont enregistrées dans la base de données.

**Gestion des Employés**

Nom: zakaria

Prénom: el court

Email: elcourtzd1@gmail.com

Téléphone: 2323232321

Salaire: 98008

Rôle: Employé

Poste: PILOTE

ID	Nom	Prénom	Email	Téléphone	Salaire	Rôle	Poste
1	zakaria	el	elcourtzd@gmail.com	0606197011	12000.0	EMPLOYEE	TEAM_LEADER
2	aicha	danone	ok@gmail.com	0808080808	1.0	ADMIN	INGENIEURE_ETUDE_E...
4	IMRAN	up	kokk@gmail.com	0808080807	123.0	ADMIN	PILOTE
6	hiba	baillade	elcourt@gmail.com	09090909	12345.0	ADMIN	PILOTE
8	test	test	test@gmail.com	1234567890	18000.0	EMPLOYEE	INGENIEURE_ETUDE_E...
10	zakzak	spam	tzak@gmail.com	5647890321	180000.0	EMPLOYEE	PILOTE
13	ikram	bairouk	ikram@gmail.com	1212121212	8798798.0	EMPLOYEE	PILOTE
14	el abdellaoui	said	s.elabdellaoui@uca.ac.ma	0606060606	120000.0	ADMIN	TEAM_LEADER

Ajouter Afficher Supprimer Modifier

FIGURE 3.4 – modifier un employé

### 3.3.4 Supprimer un Employé

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

**Gestion des Employés**

Nom:

Prénom:

Email:

Téléphone:

Salaire:

Rôle:

Poste:

**Input**

Entrez l'ID de l'employé à supprimer :

10

OK Cancel

ID	Nom	Prénom	Email	Téléphone	Salaire	Rôle	Poste
1	zakaria	el	elcourtzi@gmail.com	0606197011	12000.0	EMPLOYE	TEAM_LEADER
2	aicha	danone	ok@gmail.com	0808080808	1.0	ADMIN	INGENIEUR_ETUDE_E...
4	IMRAN	up	kokk@gmail.com	0808080807	123.0	ADMIN	PILOTE
6	hiba	baillade	elcourt@gmail.com	09090909	12345.0	ADMIN	PILOTE
8	test	test	test@gmail.com	1234567890	18000.0	EMPLOYE	INGENIEUR_ETUDE_E...
10	zakzak	spam	tzak@gmail.com	5647890321	180000.0	EMPLOYE	PILOTE
13	ikram	bairouk	ikram@gmail.com	1212121212	8798798.0	EMPLOYE	PILOTE
14	el abdellaoui	said	s.elabdellaoui@uca.ac.ma	0606060606	120000.0	ADMIN	TEAM_LEADER

Ajouter Afficher Supprimer Modifier

FIGURE 3.5 – supprimer un employé

### 3.4 Conclusion

En résumé, l'application offre une interface claire et fonctionnelle dédiée à la gestion des employés. Les administrateurs peuvent facilement ajouter de nouveaux employés, consulter la liste des employés existants, modifier leurs informations et supprimer des employés si nécessaire. Cette gestion simple et efficace permet de maintenir une base de données à jour et de garantir une administration fluide et rapide des employés.

# Conclusion Générale

La réalisation de ce travail pratique a été une expérience significative qui m'a permis de renforcer mes compétences en développement d'applications et de comprendre les différentes étapes nécessaires pour concevoir un système de gestion performant. Travailler sur ce projet a été l'occasion d'explorer les concepts théoriques étudiés en cours tout en les appliquant à un cas concret et structuré.

Le TP avait pour objectif de développer une application permettant de gérer les informations des employés de manière efficace à travers les fonctionnalités d'ajout, de modification, de suppression et de consultation. Grâce à ce projet, j'ai pu approfondir mes connaissances en Java, notamment en mettant en œuvre des principes essentiels tels que la gestion des bases de données, la manipulation des interfaces graphiques et l'intégration d'éléments interactifs.

En concevant l'application, j'ai utilisé une architecture claire et structurée qui facilite l'organisation des données et leur interaction avec l'interface utilisateur. La gestion des employés, comme l'ajout ou la modification, a été implémentée de manière intuitive, ce qui reflète une attention particulière portée à l'ergonomie et à la simplicité d'utilisation. Ce projet m'a également donné l'opportunité de maîtriser l'utilisation d'outils tels que MySQL pour la gestion de la base de données et Visual Studio Code pour le développement de l'application.

Les défis rencontrés tout au long de ce TP, notamment la synchronisation des données entre l'interface et la base de données, ou encore la validation des entrées utilisateur, m'ont permis de développer des stratégies efficaces pour résoudre des problèmes complexes. Ces expériences m'ont appris l'importance de la rigueur dans le développement logiciel, ainsi

que l'importance de tester et de valider chaque étape du processus.

Ce TP a donc été un véritable tremplin pour renforcer mes compétences techniques et mes capacités d'analyse. Il a également amélioré ma capacité à structurer un projet, à collaborer avec des outils modernes et à respecter les bonnes pratiques de développement.

En guise de conclusion, cette expérience m'a permis de mettre en œuvre des compétences précieuses en programmation et en gestion de projet informatique. Je suis convaincu que les acquis de ce TP seront des atouts majeurs pour mes projets futurs dans le domaine du développement logiciel. Ce travail pratique m'a donné les outils nécessaires pour aborder avec confiance et méthode des problématiques similaires dans des contextes professionnels.

# Webographie

[1] <https://github.com/ishowzakay/tp7-java>