

Compte Rendu du

TP N°2 : Gestion Congés

(Généricités, MVC et DAO)

Réalisé par : Aya EL ALAMA

Année universitaire : 2024-2025

Élément : Java Avancé

Groupe : 2

Introduction

Ce compte rendu présente les étapes et les résultats obtenus lors de la réalisation d'un TP visant à implémenter une application de gestion des congés. Le projet a été développé en suivant une architecture MVC et intègre la notion de généricité pour garantir une flexibilité et une réutilisabilité accrues dans la gestion des données. Les travaux ont couvert l'ensemble des couches de l'application, depuis la définition du modèle de données jusqu'à la conception de l'interface utilisateur. Les principaux défis abordés incluent :

- La gestion des congés via une base de données,
- L'implémentation de règles métier spécifiques,
- La création d'une interface graphique intuitive pour interagir avec l'utilisateur.

Organisation des fichiers Java

La structure du projet repose sur une organisation claire et modulaire des fichiers Java, conformément au modèle MVC. Chaque couche est représentée par des classes spécifiques regroupées dans des packages.

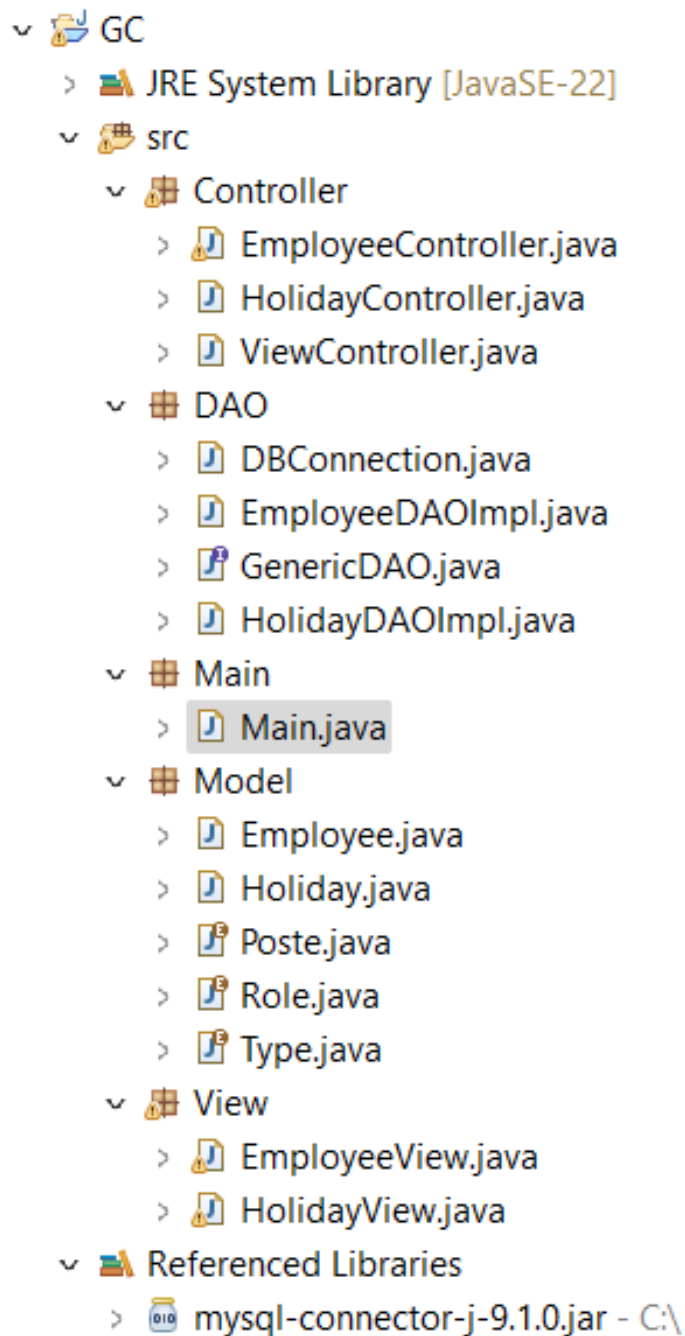


FIGURE 1 – Structure du projet

Dans cette section, nous présentons des captures d'écran illustrant les principales fonctionnalités de l'application, avant d'entrer dans les détails des codes qui les implémentent.

Captures d'écran de l'application

Voici les captures d'écran illustrant les principales fonctionnalités de l'application :

Gestion des Employés

Nom:

Prénom:

Email:

Téléphone:

Salaire:

Rôle: **Admin**

Poste: **INGENIEURE_ETUDE_ET_DEVELOPPEMENT**

ID	Nom	Prénom	Email	Téléphone	Salaire	Rôle	Poste
2	AMRANI	Nawal	nwlam@email.c...	0623456789	7000.0	EMPLOYE	INGENIEURE E...
3	El Yassir	Rachid	rachid.elyassir@...	0612345678	5000.0	EMPLOYE	TEAM_LEADER

Ajouter **Afficher** **Supprimer** **Modifier** **Gérer les Congés**

FIGURE 2 – Interface gestion des employés

Gestion des Congés

Employé Nom Complet: **AMRANI Nawal**

Date Début: **AMRANI Nawal**

Date Fin: **El Yassir Rachid**

Type: **CONGE_PAYE**

ID	Employé	Date Début	Date Fin	Type
3	AMRANI Nawal	2024-12-01	2024-12-10	CONGE_PAYE
5	El Yassir Rachid	2024-12-15	2024-12-20	CONGE_MALADIE

Ajouter **Afficher** **Supprimer** **Modifier** **Gerer les Employés**

FIGURE 3 – Interface gestion des congés

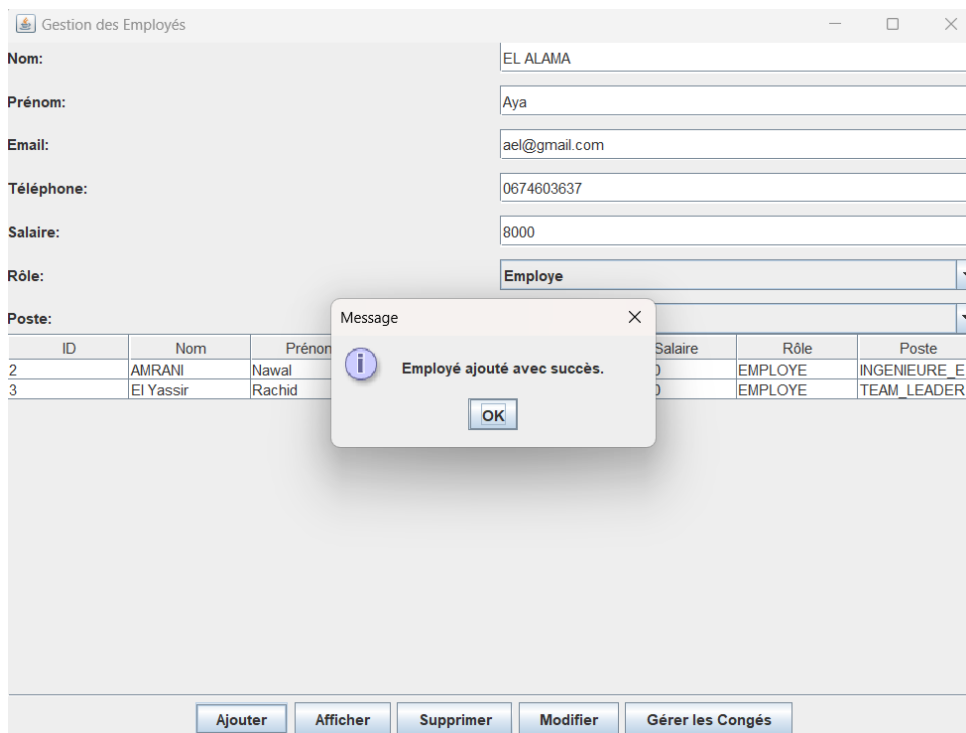


FIGURE 4 – Ajout réussi

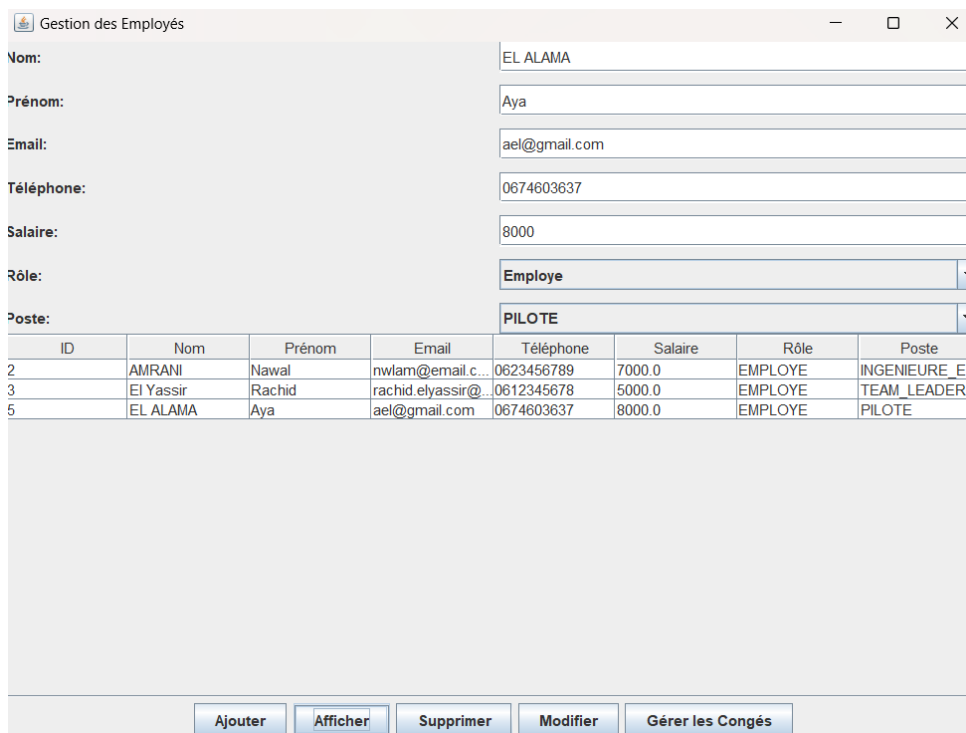


FIGURE 5 – Affichage réussi

	id	nom	prenom	email	phone	salaire	role	poste
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	2	AMRANI	Nawal	nwlam@email.com	0623456789	7000.00	EMPLOYE	INGENIEURE_ETUDE_ET_DEVELOPPEMENT
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	3	El Yassir	Rachid	rachid.elyassir@email.com	0612345678	5000.00	EMPLOYE	TEAM_LEADER
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	5	EL ALAMA	Aya	ael@gmail.com	0674603637	8000.00	EMPLOYE	PILOTE

FIGURE 6 – Base de données

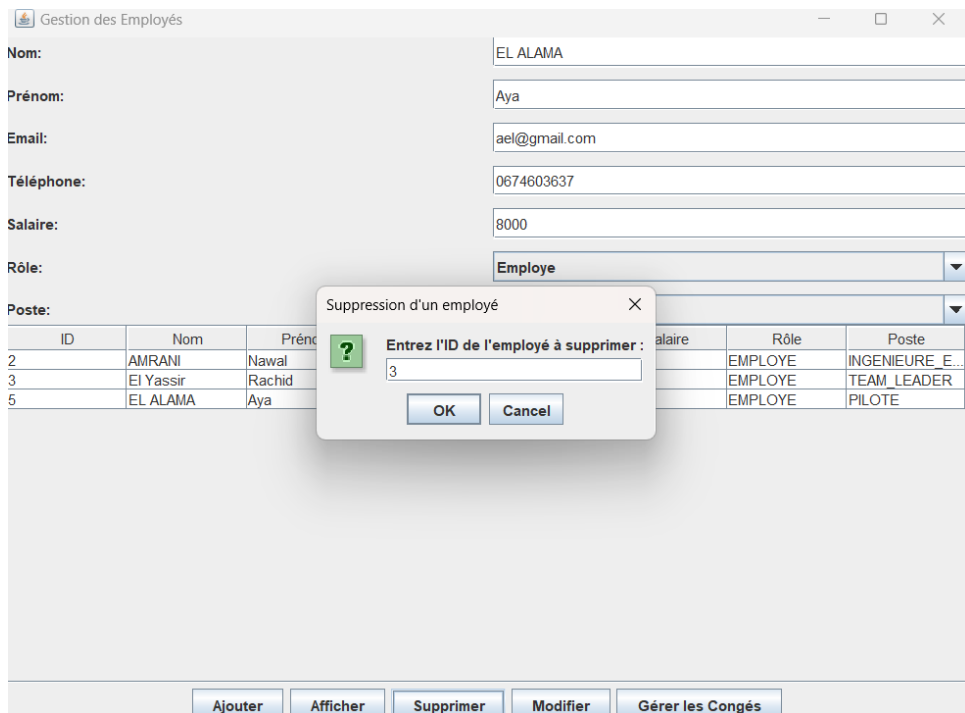


FIGURE 7 – Suppression avec succès (1)

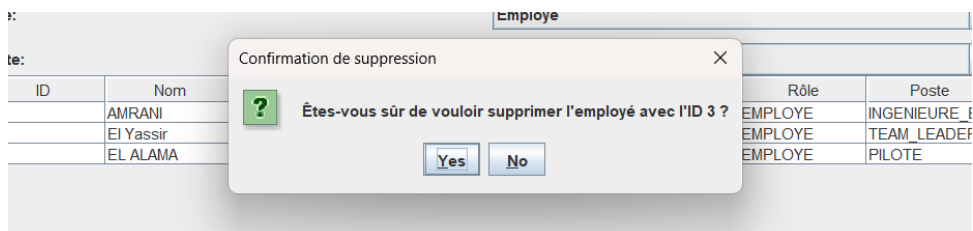


FIGURE 8 – Suppression avec succès (2)

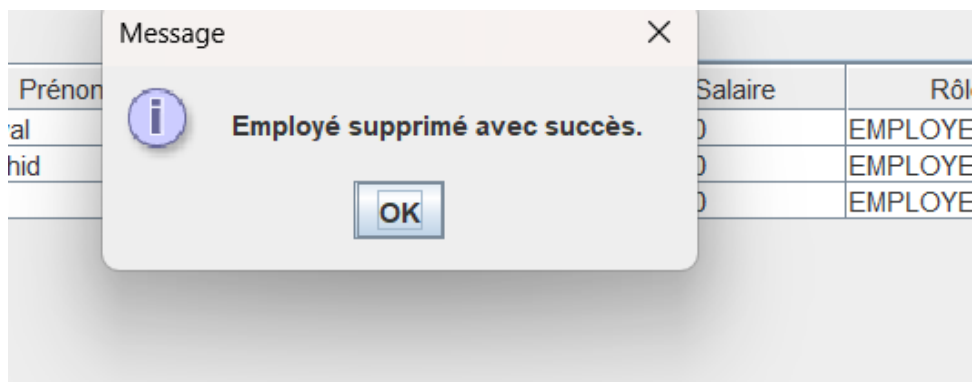


FIGURE 9 – Suppression avec succès (3)

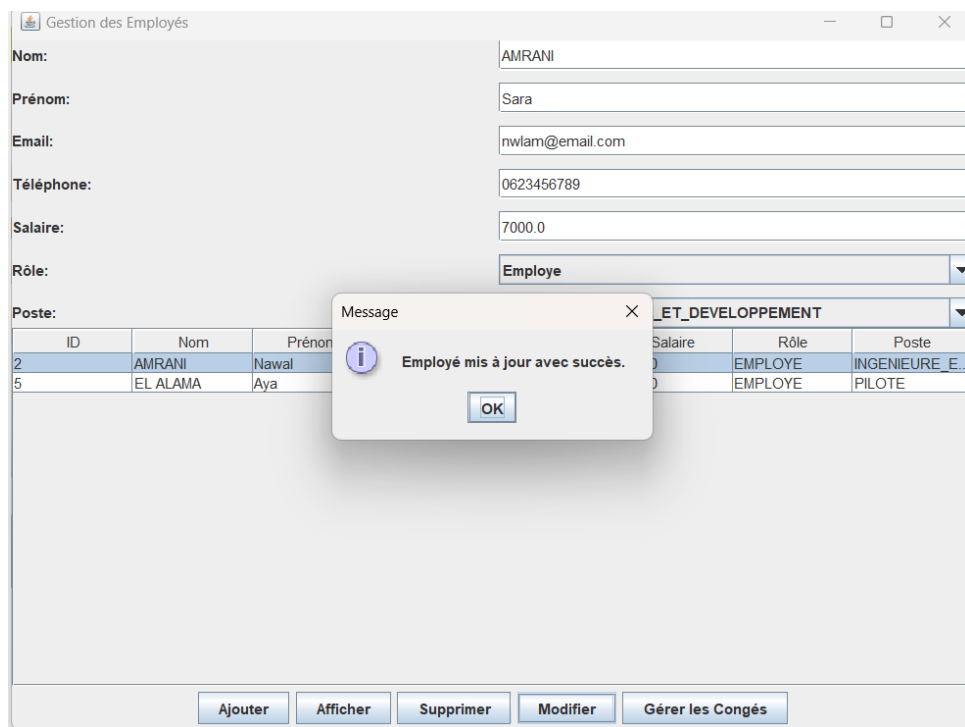


FIGURE 10 – Modification réussie (1)

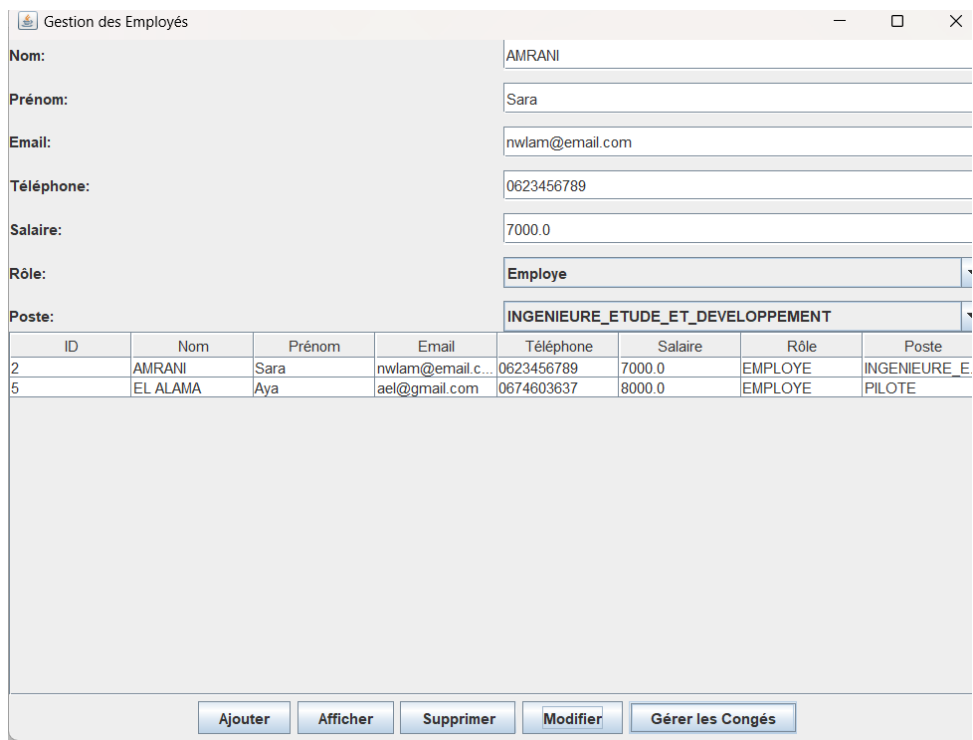


FIGURE 11 – Modification réussie (2)

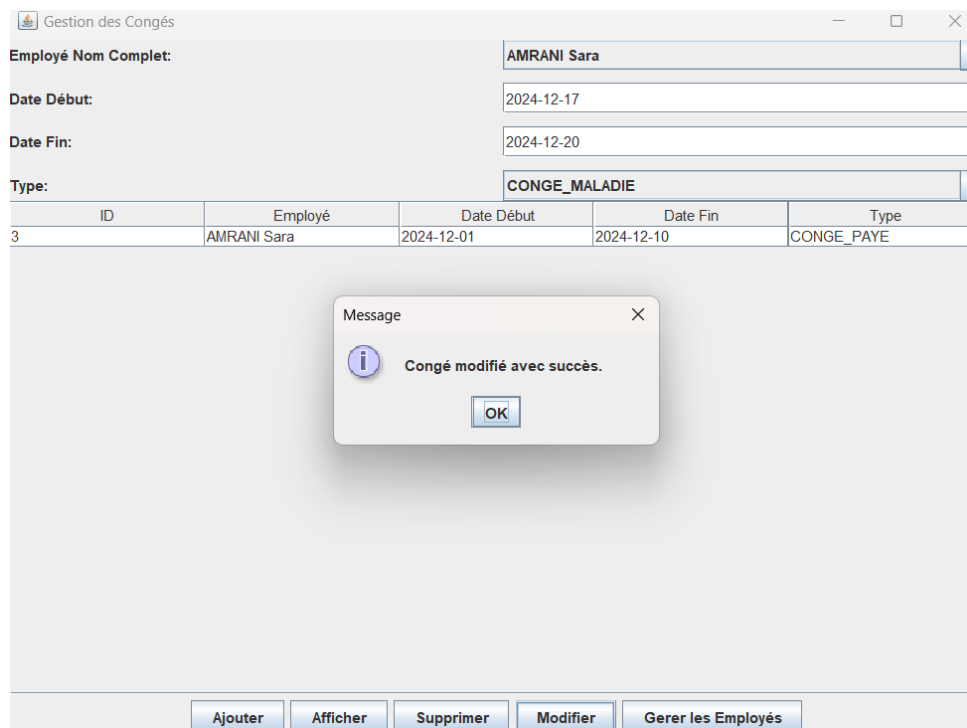


FIGURE 12 – Modification de la date début réussie

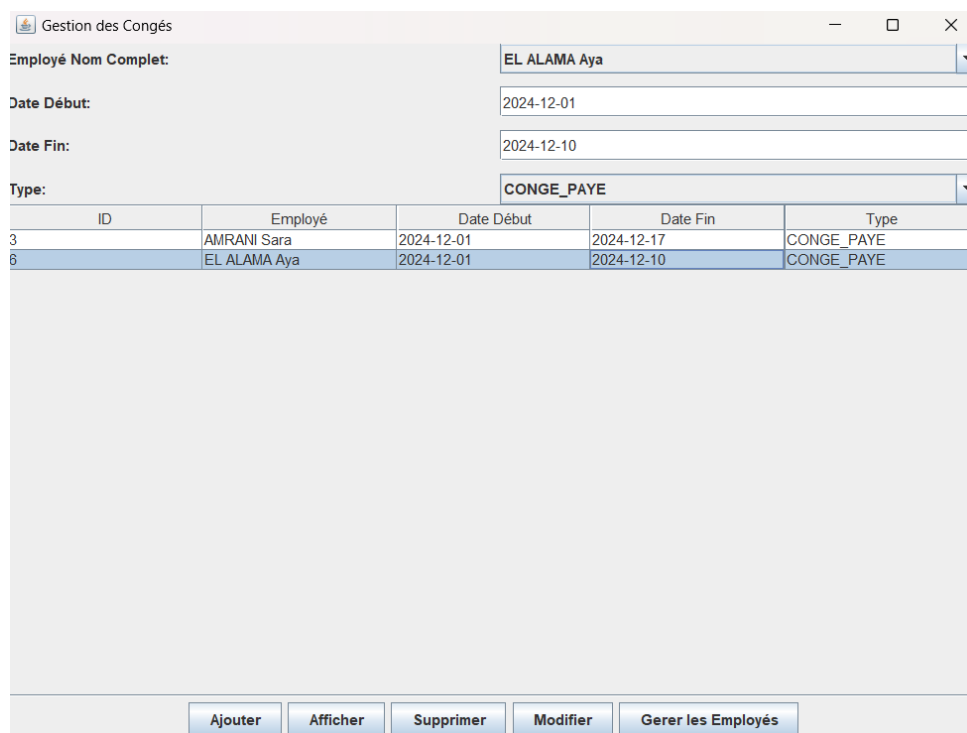


FIGURE 13 – Ajout d'un nouveau congé réussi

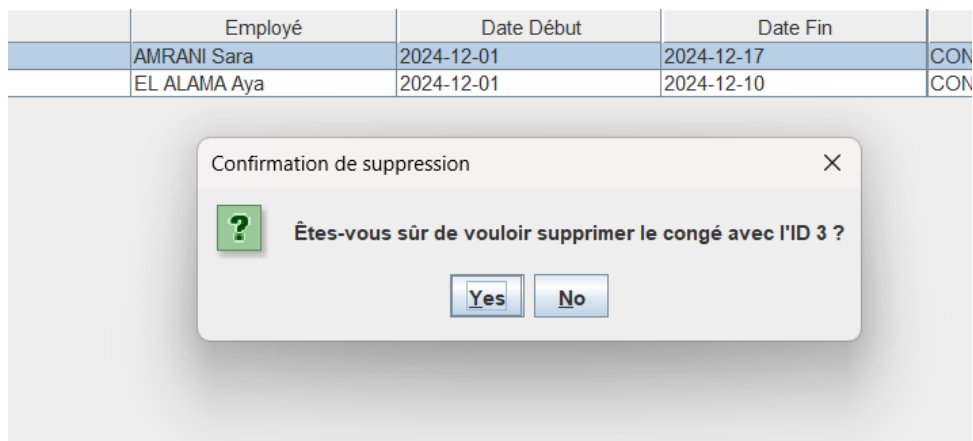


FIGURE 14 – Suppression réussie (1)

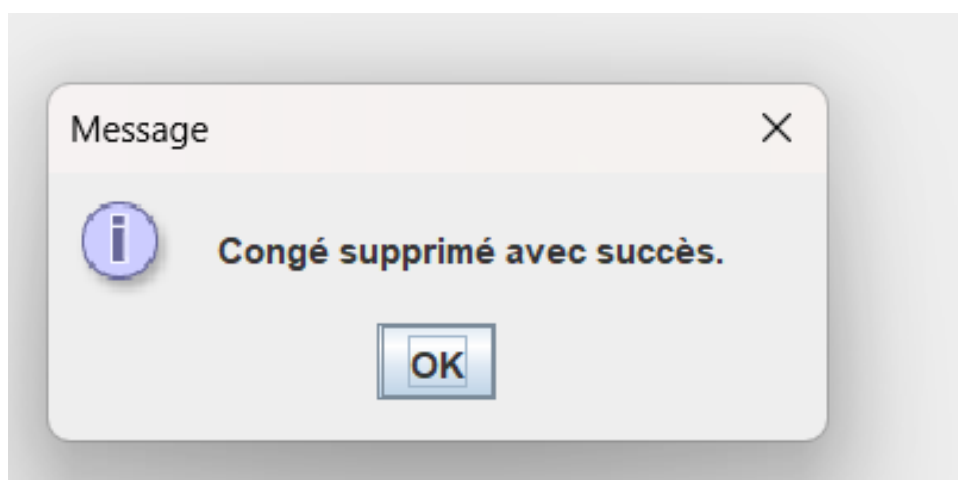


FIGURE 15 – Suppression réussie (2)

Explication détaillée des classes et de leurs rôles

Après avoir présenté les captures d'écran illustrant les principales fonctionnalités de l'application, nous allons maintenant passer à l'explication détaillée du code, en commençant par les différentes classes et leurs rôles.

Holiday.java

1. Attributs (Variables d'instance) :

- **id** : Identifiant unique du congé.
- **employeeId** : ID de l'employé associé.
- **employeeName** : Nom de l'employé.
- **startDate** et **endDate** : Dates de début et de fin du congé.
- **type** : Type de congé (défini par une énumération **Type**).

2. Constructeurs :

- **Avec tous les paramètres** : Inclut **id**, **employeeName**, **startDate**, **endDate**, et **type**.
- **Sans employeeId** : Utilisé quand l'ID de l'employé n'est pas nécessaire (par exemple pour l'affichage).
- **Sans employeeName** : Pour ajouter ou mettre à jour un congé (lorsqu'on n'a pas le nom complet).

3. Méthodes :

- **Getters** : Permettent d'obtenir les valeurs des attributs.
- **Setter pour employeeName** : Permet de modifier le nom de l'employé après création.

Code

Vous pouvez voir tous les codes via le site : <https://github.com/ayaelalama/GestionCongesTP>

GenericDAO.java

La classe `GenericDAO` est une interface générique utilisée pour définir des opérations de base sur des entités dans une application, telles que l'ajout, la suppression, la mise à jour et la recherche. Voici une explication de ses méthodes :

- **add(T entity)** : Ajoute un objet de type générique T (par exemple, une entité `Holiday` ou une autre) à la base de données.
- **delete(int id)** : Supprime un objet de type T en fonction de son identifiant unique `id`.
- **listAll()** : Retourne une liste de tous les objets de type T présents dans la base de données.
- **findById(int id)** : Recherche et retourne un objet de type T à partir de son identifiant unique `id`.
- **update(T entity, int id)** : Met à jour un objet existant dans la base de données en fonction de son `id`.

Cette interface permet de centraliser les opérations courantes de gestion des données pour différentes entités, tout en offrant la flexibilité d'utiliser des types différents grâce à la généricité (`<T>`). Cela permet une réutilisation facile et une gestion uniforme des entités dans le projet.

HolidayDAOImpl.java

La classe `HolidayDAOImpl` gère l'interaction avec la base de données pour manipuler les informations relatives aux congés des employés. Elle permet d'ajouter, de supprimer, de lister, de mettre à jour et de rechercher des congés, tout en effectuant des vérifications sur les employés et les types de congé.

L'utilisation de SQL dans les méthodes garantit une gestion dynamique et efficace des données.

HolidayController.java

La classe `HolidayController` fait le lien entre la vue, le modèle et la base de données, en gérant la logique de validation et en mettant à jour la vue en conséquence. Elle utilise des actions utilisateur (ajout, modification, suppression) pour interagir avec la base de données et garantir la cohérence des données.

Explication du Code : HolidayView.java

Ce code définit une classe graphique appelée `HolidayView`, héritant de la classe `JFrame`, qui constitue l'interface utilisateur pour la gestion des congés dans une application. La structure de cette classe se décompose en plusieurs parties :

- **Champ d'entrée utilisateur** : Un panneau d'entrée en haut (`inputPanel`) permet de saisir les informations sur les congés. Il inclut :
 - Une liste déroulante (`employeeNameComboBox`) pour sélectionner le nom de l'employé.
 - Des champs texte (`startDateField`, `endDateField`) pour les dates de début et de fin du congé.
 - Une liste déroulante (`typeCombo`) pour choisir le type de congé (`CONGE_PAYE`, `CONGE_MALADIE`, `CONGE_NON_PAYE`).

- **Tableau des congés** : Une table centrale (`holidayTable`) affiche les données existantes, entourée d'un panneau de défilement (`JScrollPane`) pour une meilleure navigation.
- **Boutons d'action** : Plusieurs boutons sont regroupés dans un panneau en bas (`buttonPanel`). Ces boutons permettent de :
 - Ajouter un congé (`addButton`),
 - Afficher les congés (`listButton`),
 - Supprimer un congé (`deleteButton`),
 - Modifier un congé (`modifyButton`),
 - Basculer vers une vue de gestion des employés (`switchViewButton`).
- **Configuration générale de la fenêtre** : La fenêtre est configurée avec un titre ("Gestion des Congés"), une taille fixe (800x600 pixels), et une disposition en bordures (`BorderLayout`) pour organiser les composants. Elle ferme l'application lorsque l'utilisateur clique sur le bouton de fermeture (`setDefaultCloseOperation`).

En résumé, la classe `HolidayView` offre une interface utilisateur intuitive et bien structurée pour gérer les congés, permettant à l'utilisateur de saisir des données, d'interagir avec les données existantes, et de basculer entre différentes vues, en ligne avec l'architecture MVC.

Main.java

La classe `Main` configure et gère les vues et contrôleurs pour l'application de gestion des employés et des congés. Elle permet de basculer entre la vue des employés et celle des congés en utilisant des boutons de commutation. Ce modèle facilite l'organisation de l'application en termes de séparation des préoccupations : chaque vue est responsable de l'affichage, tandis que chaque contrôleur gère la logique d'affaires associée.

Conclusion

Le projet présente une application de gestion des employés et des congés utilisant l'architecture Model-View-Controller (MVC), où chaque composant joue un rôle spécifique :

- **Vues** : Les classes `HolidayView` et `EmployeeView` servent à afficher les informations et à interagir avec l'utilisateur. Elles permettent de gérer les congés et les employés respectivement.
- **Contrôleurs** : Les classes `HolidayController` et `EmployeeController` sont responsables de la logique d'affaires et de la gestion des actions des utilisateurs, telles que l'ajout, la modification, et la suppression des données.
- **Modèles** : Les objets comme `Holiday` et `Employee` (gérés via des classes DAO comme `HolidayDAOImpl` et `EmployeeDAOImpl`) représentent les données, permettant leur gestion et leur persistance.

Les actions utilisateurs sont gérées par les contrôleurs, et l'interface utilisateur est définie par les vues. L'application offre une fonctionnalité de navigation entre la gestion des employés et celle des congés via des boutons de commutation. Cela permet à l'utilisateur de basculer facilement entre les deux vues.

En résumé, ce projet illustre une implémentation claire et fonctionnelle de l'architecture MVC pour gérer des données relatives aux employés et aux congés. Le code est conçu pour être extensible et facile à maintenir.

Remarque

L'explication détaillée et l'analyse des autres parties du code relatives à la gestion des employés (modèles, contrôleurs et autres fonctionnalités associées) sont disponibles dans l'autre document réalisé avec LaTeX. Vous y trouverez des informations complémentaires sur la gestion des employés et d'autres fonctionnalités de l'application.

Ce dernier compte rendu présente aussi les modifications apportées à la gestion des employés, qui ont été implémentées pour améliorer l'efficacité et l'ergonomie de l'application.