

Rapport sur les TP Java : Application de Gestion des Employés et des Vacances (MVC)

AMINE ZAGHLOUL

December 27, 2024

1 Introduction

Ce rapport résume les apprentissages acquis lors de deux séances de travaux pratiques consacrées au développement d'une application de gestion des employés et des vacances en Java, suivant le modèle architectural MVC (Modèle-Vue-Contrôleur). L'interface utilisateur a été réalisée avec Swing, et l'accès aux données est géré par un objet DAO (Data Access Object).

2 Objectifs des TP

- Comprendre et appliquer le modèle MVC.
- Utiliser Swing pour la création d'interfaces graphiques.
- Implémenter un DAO pour la gestion de la persistance des données.
- Mettre en œuvre la communication entre les différentes couches de l'application.

3 Déroulement des TP et Apprentissages

3.1 Mise en place de la structure MVC et de la Vue

- **Création du projet et des packages :** J'ai appris à organiser le projet en packages distincts pour chaque composante du MVC (model, view, controller, dao). Cela permet une meilleure organisation et maintenabilité du code.
- **Conception des classes du Modèle :** J'ai défini les classes métiers représentant les employés (avec des attributs comme nom, prénom, date d'embauche, etc.) et les demandes de vacances (avec des attributs comme date de début, date de fin, motif, statut, etc.).

- **Développement de l'interface utilisateur (Vue) avec Swing :** J'ai appris à utiliser les composants Swing tels que `JFrame`, `JPanel`, `JLabel`, `TextField`, `Button`, `JTable` pour créer les interfaces pour :
 - L'affichage de la liste des employés.
 - L'ajout, la modification et la suppression d'employés.
 - La gestion des demandes de vacances.
- **Liens entre les composants Swing et le Controller :** J'ai commencé à mettre en place les écouteurs d'événements (`ActionListener`, etc.) sur les composants Swing pour interagir avec le Controller.

3.2 Implémentation du Controller et du DAO

- **Développement des classes du Controller :** J'ai implémenté les classes du Controller qui reçoivent les événements de la Vue, exécutent la logique métier en utilisant le Modèle, et mettent à jour la Vue en conséquence. J'ai notamment appris à :
 - Récupérer les données saisies par l'utilisateur dans les composants Swing.
 - Valider les données.
 - Appeler les méthodes du Modèle et du DAO.
 - Mettre à jour l'interface graphique.
- **Création du DAO et interaction avec la base de données (ou stockage de données) :** J'ai implémenté le DAO pour gérer la persistance des données. J'ai appris à :
 - Établir une connexion à la base de données (ou utiliser un fichier pour stocker les données dans le cas d'un TP d'initiation).
 - Créer les méthodes pour les opérations CRUD (Create, Read, Update, Delete) sur les employés et les demandes de vacances.
 - Gérer les exceptions liées à l'accès aux données.
- **Communication entre le Controller et le DAO :** J'ai mis en place la communication entre le Controller et le DAO pour persister les données saisies par l'utilisateur.

4 Difficultés rencontrées et solutions

- *(Décrivez les difficultés que vous avez rencontrées, par exemple :)*
 - Difficulté à comprendre le flux d'exécution entre les différentes couches du MVC.
 - Problèmes de mise en page avec Swing.
 - Erreurs de connexion à la base de données.
- *(Décrivez les solutions que vous avez trouvées, par exemple :)*
 - Relecture du cours et consultation de ressources en ligne sur le MVC.
 - Utilisation de Layout Managers (comme `BorderLayout`, `GridLayout`) pour une meilleure gestion de la mise en page.
 - Vérification des paramètres de connexion à la base de données et gestion des exceptions.

5 Conclusion

Ces deux séances de TP m'ont permis de consolider mes connaissances sur le modèle MVC, l'utilisation de Swing et l'implémentation d'un DAO. J'ai acquis une expérience pratique dans la conception et le développement d'une application Java avec une architecture bien définie, ce qui est essentiel pour la création d'applications maintenables et évolutives.