École Marocaine des Sciences de l'Ingénieur

Rapport de Projet de Fin de Module

Système de Gestion des Ressources Humaines

Application de bureau JavaFX pour la gestion monôme

Réalisé par :
LANOUARI MOUNA

Encadré par :
Abderrahim Larhlimi

Java Avancé & Programmation Orientée Objet

Année Universitaire : 2025-2026

Remerciements

Je tiens à remercier tout d'abord mon professeur, M. [Nom du Professeur], pour son encadrement et ses conseils précieux durant ce module de Java Avancé. Mes remerciements s'adressent également à l'administration de l'EMSI pour l'environnement de travail mis à notre disposition.

Table des matières

Introduction Générale

0.1 Contexte

Dans le cadre du module Java Avancé, ce projet vise à concevoir une solution logicielle répondant aux besoins de gestion administrative d'une entreprise moderne.

0.2 Problématique

La gestion manuelle des ressources humaines (bulletins de paie, congés) engendre souvent des erreurs de calcul et des pertes d'informations. Une centralisation numérique est devenue indispensable pour garantir l'intégrité des données.

0.3 Objectifs

— Automatisation du calcul des salaires nets.
— Gestion des demandes de congés et de leur statut.
— Authentification sécurisée des utilisateurs (RH et Employés).
— Génération de reçus de bulletins de paie formatés.

Chapitre 1

Analyse et Conception

1.1 Spécification des besoins

1.1.1 Besoins Fonctionnels

Le système permet au Responsable RH de gérer les employés et de valider les congés. L'employé peut consulter ses bulletins et effectuer des demandes.

1.1.2 Besoins Non-Fonctionnels

Performance (chargement rapide des listes via Hibernate), Ergonomie (Interface moderne JavaFX) et Sécurité.

1.2 Conception UML

1.2.1 Diagramme de Classes

Le diagramme de classes suivant illustre la structure des données et les relations entre les entités `Employe`, `Conge` et `BulletinPaie`.

FIGURE 1.1 – Diagramme Classes du Système RH

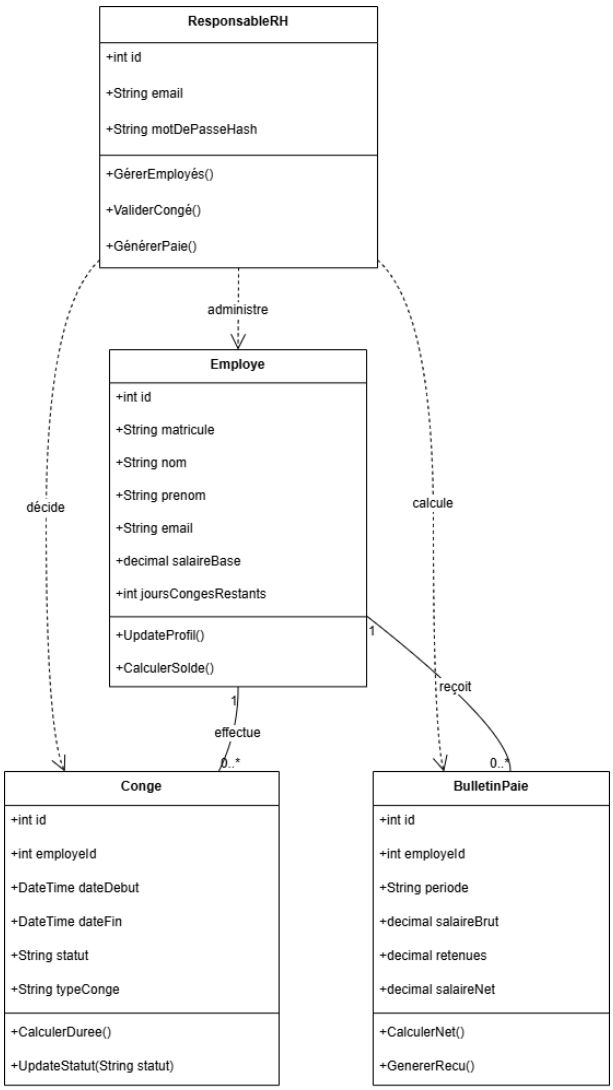

FIGURE 1.2 – Enter Caption

1.3 Conception de la Base de Données

Le dictionnaire de données suivant détaille la structure précise des tables MySQL. Chaque table correspond à une entité de notre modèle de données et inclut les clés étrangères nécessaires au fonctionnement de l'ORM Hibernate.

1.4 Conception de la Base de Données

Le dictionnaire de données suivant détaille la structure des tables MySQL, ajusté pour s'intégrer parfaitement dans les marges du document.

Table	Champ	Type	Contraintes	Description
ResponsableRH	id	INT	PK, AI	Identifiant unique de l'administrateur.
	email	VARCHAR(100)	Unique, Not Null	Email utilisé pour l'authentification.
	motDePasseHash	VARCHAR(255)	Not Null	Mot de passe sécurisé (BCrypt).
Employe	id	INT	PK, AI	Identifiant unique de l'employé.
	matricule	VARCHAR(20)	Unique, Not Null	Code d'identification interne.
	nom / prenom	VARCHAR(50)	Not Null	Informations civiles de l'employé.
	salaireBase	DECIMAL(10,2)	Not Null	Salaire brut contractuel.
	joursConges	INT	Default 30	Solde de congés restants.
Conge	id	INT	PK, AI	Identifiant unique de la demande.
	employeId	INT	FK (Employe.id)	Lien vers l'employé demandeur.
	dateDebut/Fin	DATE	Not Null	Période du congé demandé.
	statut	VARCHAR(20)	Not Null	EN_ATTENTE, APPROUVE, REFUSE.
BulletinPaie	id	INT	PK, AI	Identifiant unique du bulletin.
	employeId	INT	FK (Employe.id)	Lien vers l'employé bénéficiaire.
	periode	VARCHAR(7)	Not Null	Format MM/YYYY.
	salaireNet	DECIMAL(10,2)	Not Null	Résultat du calcul après retenues.

Chapitre 2

Environnement Technique

Ce chapitre décrit avec précision les outils, langages et bibliothèques utilisés pour le développement de l'application de gestion RH.

2.1 Outils de Développement

- **Langage de programmation :** Java (Version 17/21). Le choix s'est porté sur cette version pour sa stabilité et le support des fonctionnalités modernes de la Programmation Orientée Objet.
- **Environnement de développement (IDE) :** VS Code. Cet outil offre une gestion fluide des projets JavaFX et une intégration native avec Maven.
- **Système de Gestion de Base de Données (SGBD) :** MySQL 8.0. Utilisé pour sa robustesse et sa gestion efficace de la persistance des données via l'ORM Hibernate.
- **Outils de modélisation : Draw.io** . Cet outil a été utilisé pour la conception graphique des diagrammes UML afin d'assurer une documentation claire et normalisée du système.

2.2 Frameworks et Bibliothèques Tierces

- **Framework UI (JavaFX) :** Utilisé pour concevoir une interface utilisateur riche, moderne et réactive.
- **Hibernate (ORM) :** Framework incontournable pour assurer la persistance des objets Java vers les tables de la base de données MySQL sans écrire de requêtes SQL complexes.
- **Lombok :** Utilisé pour réduire le code "boilerplate" en générant automatiquement les Getters, Setters et Constructeurs dans les entités.

2.3 Gestion de projet et Build (Maven)

L'application utilise **Maven** comme gestionnaire de build pour automatiser la gestion des dépendances et faciliter l'intégration des drivers. Voici un extrait des dépendances clés intégrées dans le fichier `pom.xml` :

```xml
<dependencies>
    <dependency>
        <groupId>mysql</groupId>
        <artifactId>mysql-connector-java</artifactId>
        <version>8.0.33</version>
    </dependency>

    <dependency>
        <groupId>org.hibernate</groupId>
        <artifactId>hibernate-core</artifactId>
        <version>6.2.7.Final</version>
    </dependency>

    <dependency>
        <groupId>org.openjfx</groupId>
        <artifactId>javafx-controls</artifactId>
        <version>21</version>
    </dependency>
</dependencies>
```

Listing 2.1 – Extrait du fichier pom.xml

Chapitre 3

Architecture et Implémentation

Ce chapitre détaille l'organisation technique de mon application de Gestion RH et les solutions choisies pour répondre aux problématiques de persistance et de calcul métier.

3.1 Architecture Logicielle

L'application est structurée selon une architecture en couches pour isoler la base de données MySQL de l'interface utilisateur JavaFX :

- **Couche Modèle (Entities) :** Regroupe les classes `Employe`, `Conge` et `BulletinPaie`. Ce sont des entités Hibernate qui portent les annotations JPA pour mapper les colonnes de la base de données.
- **Couche Service :** C'est ici que réside l'intelligence de l'application. Elle contient les algorithmes de calcul des salaires nets (retenues sur le brut) et la validation des droits aux congés.
- **Couche DAO (Data Access Object) :** Elle utilise la `SessionFactory` pour exécuter les opérations CRUD. Elle permet, par exemple, de récupérer la liste des congés "En attente" pour que le RH puisse les traiter.
- **Couche Contrôleur :** Elle fait le lien entre les fichiers FXML (Vues) et les Services. Chaque bouton de Dashboard (ex : "Approuver") appelle une méthode spécifique ici.

3.2 Extraits de code clés et Logique Métier

3.2.1 Calcul du Salaire Net (Logique Métier)

Voici l'implémentation de la méthode de calcul dans la classe `BulletinPaie`, qui déduit les retenues du salaire brut :

```
1 public void calculerNet() {
2     // Le salaire net est calcule en retranchant les retenues (CNSS
          , AMO...)
3     this.salaireNet = this.salaireBrut.subtract(this.retenues);
4 }
```

Listing 3.1 – Logique de calcul du salaire net

3.2.2 Validation des Congés (Accès aux données)

Cet extrait montre comment le Responsable RH modifie l'état d'une demande dans la base de données via Hibernate :

```
public void updateStatut(int congeId, String nouveauStatut) {
    Transaction tx = session.beginTransaction();
    Conge c = session.get(Conge.class, congeId);
    c.setStatut(nouveauStatut); // "APPROUVE" ou "REFUSE"
    session.update(c);
    tx.commit();
}
```

Listing 3.2 – Mise a jour du statut d'un conge

3.2.3 Génération de l'aperçu du Bulletin

Pour l'affichage dans l'interface, j'utilise un `StringBuilder` afin de construire proprement la chaîne de caractères qui sera affichée dans la zone de texte du Dashboard :

```
public String genererRecu() {
    return String.format("Bulletin de : %s\nPeriode : %s\nNet a
       payer : %.2f DH",
           this.employe.getNom(), this.periode, this.salaireNet);
}
```

Listing 3.3 – Formatage du bulletin pour l'affichage

Chapitre 4

Conclusion

La réalisation de ce projet de gestion des Ressources Humaines a permis de concevoir une application robuste en exploitant la puissance de l'écosystème Java moderne.

4.1 Bilan Technique

L'apport majeur réside dans l'utilisation de l'ORM Hibernate, qui a permis de s'affranchir des limites du JDBC classique. En remplaçant les requêtes SQL manuelles par un mapping objet-relationnel, nous avons sécurisé la couche de persistance et simplifié la manipulation des entités (`Employe`, `Conge`, `Bulletin`). Cette approche garantit une meilleure maintenabilité et une séparation stricte entre les données et la logique métier.

L'interface JavaFX, structurée selon le pattern MVC, offre une expérience utilisateur fluide grâce à la liaison de données dynamique (*Data Binding*). L'architecture globale assure ainsi une gestion cohérente et performante des processus RH.