

Université Cadi Ayyad

École Supérieure De Technologie - Safi

Département : Informatique

Filière : Génie Informatique (GI)

Compte Rendu TP2

Gestion des Congés en Java (DAO/MVC)

Nom et Prénom : Salaheddine ELMOHADEB

1. Introduction

Le système de gestion des congés a été intégré au projet existant de gestion des employés pour permettre une administration efficace des absences des employés. Cette fonctionnalité inclut la planification, la modification et le suivi des congés en fonction des types définis (payé, non payé, maladie). Ce rapport détaille la structure et le fonctionnement de ce module, y compris les composants DAO, Modèle et Vue.

2. Architecture et Organisation du Projet

2.1 Architecture du Projet

L'architecture du projet repose sur le modèle MVC, séparant la logique en trois couches principales :

1. **Modèle (Model)** : Cette couche contient les règles métier et les classes représentant les entités, telles que *Holiday* (pour les congés) et *Type* (pour les catégories de congés).
2. **Vue (View)** : Elle est responsable de l'interface utilisateur et de la présentation des données, utilisant des composants Swing comme *JTable* et *JComboBox*.
3. **Contrôleur (Controller)** : Cette partie connecte la vue et le modèle, tout en gérant les interactions utilisateur. Par exemple, la classe *HolidayController* traite les événements des boutons (ajouter, modifier, supprimer) et assure la mise à jour des données entre la vue et le modèle.

2.2 Organisation des Packages

Le projet est divisé en quatre packages principaux :

- **DAO** : Contient la gestion de la connexion à la base de données et les classes pour les opérations CRUD.
- **Controller** : Inclut les classes responsables de la logique et des interactions entre la vue et le modèle, comme *HolidayController*.
- **Model** : Comprend les entités et les structures de données, notamment *Holiday*, *HolidayModel*, et *Type*.
- **View** : Contient les interfaces utilisateur, y compris *MainView*, qui combine la gestion des employés et la gestion des congés.

2.3 MainView

MainView offre une interface utilisateur unifiée pour gérer à la fois les employés et les congés. Elle intègre des composants interactifs comme des tableaux et des formulaires pour faciliter la navigation et la gestion.

3. Étapes Suivies

Pour réaliser ce projet, nous avons suivi les étapes ci-dessous :

1. **Conception de la base de données** :
 - Création d'une table *Holiday*, structurée pour contenir les données liées aux congés, telles que l'identifiant, le nom de l'employé, les dates de début et de fin, ainsi que le type de congé.

2. Développement du module DAO :

- Élaboration d'une interface générique *GenericDAO* pour définir les opérations CRUD standard.
- Implémentation de cette interface dans *HolidayDAOImpl*, en utilisant des requêtes SQL paramétrées pour interagir avec la base de données.

3. Création des classes du modèle :

- Mise en place de la classe *Holiday*, qui représente un congé avec des propriétés comme *id*, *employeeName*, *startDate*, *endDate*, et *type*.
- Définition de l'énumération *Type* pour représenter les différents types de congés, tels que *PAYE*, *NON PAYE*, et *MALADIE*.

4. Développement de la vue :

- Conception d'une interface conviviale avec Swing, permettant de gérer les congés via des formulaires et des tableaux interactifs.

5. Développement du contrôleur :

- Intégration de la logique pour gérer les actions des utilisateurs (ajout, suppression, modification) et assurer la synchronisation des données entre la base et l'interface utilisateur.

4. Explications des Codes

4.1 Modèle (*Holiday*, *HolidayModel*)

- **Objectif** : Le modèle gère les données manipulées par l'application.
- **Détails** :
 - *Holiday* : Cette classe encapsule les informations relatives aux congés, avec des propriétés telles que *id*, *employeeName*, *startDate*, *endDate*, et *type*. Elle inclut également des getters et setters pour accéder à ces données.
 - *HolidayModel* : Responsable de la logique métier liée à la gestion des congés, comme la validation des dates ou le calcul des durées.
 - *Type* : Une énumération définissant les différentes catégories de congés, comme *PAYE*, *NON PAYE*, *MALADIE*, afin d'assurer la validité des données.

4.1.1 Classe Holiday

```
1 package Model;
2 import java.text.SimpleDateFormat;
3 import java.util.Date;
4 import java.util.concurrent.TimeUnit;
5
6 public class Holiday {
7     private String date_debut, date_fin;
8     private Types type;
9     private int id, id_empl;
10
11     public Holiday (int id, String date_debut, String date_fin,
12                     Types type){
13         this.id = id;
14         this.date_debut = date_debut;
15         this.date_fin = date_fin;
16         this.type = type;
```

```

16     }
17
18     public Holiday ( String date_debut, String date_fin, Types type
19     , int id_empl){
20         this.date_debut = date_debut;
21         this.date_fin = date_fin;
22         this.type = type;
23         this.id_empl = id_empl;
24     }
25
26     public String getDate_debut() {
27         return date_debut;
28     }
29
30     public void setDate_debut(String date_debut){this.date_debut =
31     date_debut;}
32
33     public String getDate_fin() {
34         return date_fin;
35     }
36
37     public void setDate_fin(String date_fin){this.date_fin =
38     date_fin;}
39
40
41     public int getId() {
42         return id;
43     }
44
45     public void setId(int id) {
46         this.id = id;
47     }
48
49     public int getId_empl() {
50         return id_empl;
51     }
52
53     public void setId_empl(int id_empl) {
54         this.id_empl = id_empl;
55     }
56
57     public Holiday.Types getTypes() {
58         return type;
59     }
60
61     public void setTypes(Holiday.Types type) {
62         this.type = type;
63     }
64
65     public enum Types{
66         Conge_Paye ,
67         Conge_non_Paye ,
68         Conge_Maladie
69     }
70 }

```

Listing 1 – Classe Holiday

4.1.2 Classe HolidayModel

La classe HolidayModel est responsable de la gestion des congés des employés. Elle interagit avec le DAO HolidayDAOImpl pour ajouter, afficher, supprimer et mettre à jour des congés dans la base de données

```
1 package Model;
2
3 import DAO.CongedAOImpl;
4 import DAO.EmployeeDAOImpl;
5
6 import java.text.ParseException;
7 import java.text.SimpleDateFormat;
8 import java.util.ArrayList;
9 import java.util.Date;
10 import java.util.concurrent.TimeUnit;
11
12 public class HolidayModel {
13     private CongedAOImpl conge;
14     private EmployeeDAOImpl empl;
15     public HolidayModel(CongedAOImpl conge, EmployeeDAOImpl empl ){
16         this.conge = conge;
17         this.empl = empl;
18     }
19
20     public boolean ajouterConge (String date_debut, String date_fin
21 , Holiday.Types type, int id_empl){
22
23         int jours = duree(date_debut, date_fin);
24         if(empl.is_solde_enough(id_empl, jours )){
25             Holiday nouveauConge = new Holiday(date_debut, date_fin
26 , type, id_empl);
27             conge.add(nouveauConge);
28             empl.updateSolde(id_empl, jours);
29             return true;
30         }
31         System.out.println("hna model makhdamch");
32         return false;
33     }
34
35     public int duree(String startDate, String endDate) {
36         try {
37             SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd
38 ");
39
40             Date date1 = sdf.parse(startDate);
41             Date date2 = sdf.parse(endDate);
42
43             long diffInMillies = Math.abs(date2.getTime() - date1.
44 getTime());
45
46             return (int) TimeUnit.DAYS.convert(diffInMillies,
47 TimeUnit.MILLISECONDS);
48         } catch (ParseException e) {
49             e.printStackTrace();
50         }
51     }
52 }
```

```

44         } catch (ParseException e) {
45             e.printStackTrace();
46             return 0;
47         }
48     }
49
50     public boolean updateHoliday(String date_debut, String date_fin
51     , Holiday.Types type, int id_empl) {
52
53         Holiday new_conge = new Holiday(date_debut, date_fin, type,
54         id_empl);
55         conge.update(new_conge, id_empl);
56
57         return true;
58     }

```

Listing 2 – Classe HolidayModel

4.2 DAO (*DBConnection*, *CongesDAOimpl*, *GenericDAOi*)

4.2.1 DBConnection

Objectif : Assurer l'établissement et la gestion de la connexion avec la base de données MySQL dédiée à la gestion des congés.

```

1 package DAO;
2
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5 import java.sql.SQLException;
6
7 public class DBConnection {
8     private static final String URL = "jdbc:mysql://localhost:3406/
9     gestion_employee";
10    private static final String UTILISATEUR = "root";
11    private static final String MOT_DE_PASSE = "root";
12    static Connection conn = null;
13
14    public static Connection getConnexion(){
15        if(conn == null){
16            try {
17                conn = DriverManager.getConnection(URL, UTILISATEUR,
18                MOT_DE_PASSE);
19            } catch (SQLException e){
20                e.printStackTrace();
21                throw new RuntimeException("Error lors de la
22                connexion avec DB");
23            }
24        }
25        return conn;
26    }
27
28    //jointure entre conge and employee
29 }

```

4.2.2 CongeDAOimpl

```

1 package DAO;
2
3 import Model.Holiday;
4
5 import java.sql.*;
6 import java.util.ArrayList;
7 import java.util.List;
8
9 public class CongeDAOImpl implements GenericDAOI<Holiday> {
10     private Connection conn;
11
12     public CongeDAOImpl() {
13         this.conn = DBConnection.getConnection();
14     }
15
16
17     @Override
18     public List<Holiday> findAll() {
19         List<Holiday> conge = new ArrayList<>();
20         String query = "SELECT * FROM conge";
21         try (Statement stmt = conn.createStatement();
22             ResultSet rs = stmt.executeQuery(query)) {
23             while (rs.next()) {
24                 conge.add(new Holiday(
25                     rs.getInt("id"),
26                     rs.getString("date_debut"),
27                     rs.getString("date_fin"),
28                     Holiday.Types.valueOf(rs.getString("type"))
29                 ));
30             }
31         } catch (SQLException e) {
32             System.out.println("Erreur lors de la r cup ration de
33             tous les cong s !!!!!");
34         }
35         return conge;
36     }
37 }

```

Listing 4 – CongeDAOimpl

4.2.3 GenericDAOi

L'interface GenericDAOI définit les méthodes génériques pour effectuer des opérations CRUD (ajouter, supprimer, mettre à jour et afficher des entités), tandis que la classe HolidayDAOImpl implémente cette interface pour gérer spécifiquement les opéra tions sur les entités Holiday.

```

1 package DAO;

```

```

2
3 import Model.Employee;
4 import Model.Holiday;
5
6 import java.util.*;
7
8 public interface GenericDAOI<T>{
9
10
11     List<T> findAll();
12     void add(T E);
13     void update(T E,int id);
14     void delete(int id);
15
16 }

```

Listing 5 – GenericDAOi

4.3 VIEW

```

1 package View;
2
3
4 import Model.Employee;
5 import Model.Holiday;
6 import javax.swing.*;
7 import javax.swing.table.DefaultTableModel;
8 import java.awt.*;
9
10
11 public class HolidayView extends JPanel{
12     private JPanel mainPanel, topPanel, centerPanel, bottomPanel,
13     employeePanel;
14     private JTabbedPane switchPanels;
15
16     private JLabel lblNom, lblType, lblDate_fin, lblDate_debut;
17     private JTextField txtDate_fin, txtDate_debut, txtid_empl;
18
19     public JTable congeTable;
20     private JComboBox<Holiday.Types> cbTypes;
21
22     public JButton ajouterButton = new JButton("Ajouter");
23     public JButton modifierButton = new JButton("Modifier");
24     public JButton supprimerButton = new JButton("Supprimer");
25     public JButton afficherButton = new JButton("Afficher");
26
27     public HolidayView() {
28         setSize(800, 600);
29         setLayout(new BorderLayout());
30
31
32         mainPanel = new JPanel(new BorderLayout());

```



```

33     topPanel = new JPanel(new GridLayout(7, 2, 10, 10));
34     centerPanel = new JPanel(new BorderLayout());
35     bottomPanel = new JPanel(new GridLayout(1, 4, 10, 10));
36
37     lblNom = new JLabel("id de l'employe:");
38     txtid_empl = new JTextField();
39
40     lblType = new JLabel("Type:");
41     cbTypes = new JComboBox<>(Holiday.Types.values());
42
43     lblDate_debut = new JLabel("Date de debut:");
44     txtDate_debut = new JTextField();
45
46     lblDate_fin = new JLabel("Date de fin:");
47     txtDate_fin = new JTextField();
48
49     topPanel.add(lblNom);
50     topPanel.add(txtid_empl);
51     topPanel.add(lblType);
52     topPanel.add(cbTypes);
53     topPanel.add(lblDate_debut);
54     topPanel.add(txtDate_debut);
55     topPanel.add(lblDate_fin);
56     topPanel.add(txtDate_fin);
57
58
59     congeTable = new JTable(new DefaultTableModel(new Object[][]{
60         "id", "id de Employe", "Date de debut", "Date de fin", "Type"}, 0)
61     );
62     JScrollPane scrollPane = new JScrollPane(congeTable);
63
64     centerPanel.add(scrollPane, BorderLayout.CENTER);
65
66     bottomPanel.add(ajouterButton);
67     bottomPanel.add(modifierButton);
68     bottomPanel.add(supprimerButton);
69     bottomPanel.add(afficherButton);
70
71     mainPanel.add(topPanel, BorderLayout.NORTH);
72     mainPanel.add(centerPanel, BorderLayout.CENTER); mainPanel.
73     add(bottomPanel, BorderLayout.SOUTH);
74
75     add(mainPanel);
76     //setVisible(true);
77 }
78
79 public String getDate_debut() {
80     return txtDate_debut.getText();
81 }
82 public String getDate_fin() {
83     return txtDate_fin.getText();
84 }

```

```

83     public int getId_empl() {
84         return Integer.parseInt(txtid_empl.getText());
85     }
86     public Holiday.Types getTypes() {
87         return (Holiday.Types) cbTypes.getSelectedItemAt();
88     }
89
90     public int getId(JTable table) {
91         int selectedRow = table.getSelectedRow();
92
93         if (selectedRow == -1) {
94             JOptionPane.showMessageDialog(null, "Veuillez
95 s lectionner une ligne !");
96             return -1;
97         }
98         return (int) table.getValueAt(selectedRow, 0);
99     }
100 }

```

Listing 6 – HolidayView

4.4 Main

Objectif

La classe Main est le point d'entrée de l'application. Elle initialise les composants de la vue (MainView), les objets de persistance (DAO), les modèles de données (EmployeeModel, HolidayModel), et les contrôleurs (EmployeeController, HolidayController)

```

1
2 import Controller.EmployeeController;
3 import Controller.HolidayController;
4 import DAO.CongeDAOImpl;
5 import DAO.EmployeeDAOImpl;
6 import Model.EmployeeModel;
7 import Model.HolidayModel;
8 import View.EmployeeView;
9 import View.GlobalView;
10 import View.HolidayView;
11
12 public class Main {
13     public static void main(String[] args) {
14
15         EmployeeView Emplview = new EmployeeView();
16         EmployeeDAOImpl Empldao = new EmployeeDAOImpl();
17         EmployeeModel Emplmodel = new EmployeeModel(Empldao);
18
19         HolidayView HolidayView = new HolidayView();
20         CongeDAOImpl CongeDao = new CongeDAOImpl();
21         HolidayModel HolidayModel = new HolidayModel(CongeDao,
22 Empldao);
23
24         GlobalView view = new GlobalView(Emplview, HolidayView);

```

```

24     HolidayController controllerHoliday = new HolidayController
      (HolidayView,HolidayModel);
25
26     new EmployeeController(Emplview, Emplmodel);
27 }
28 }
29

```

Listing 7 – Main

4.5 Affichage

Id	Id de Employee	Date de début	Date de fin	Type
2	1	2024-10-02	2024-10-04	Congé_Paye
3	2	2024-10-02	2024-10-08	Congé_Paye
4	3	2024-10-01	2024-10-17	Congé_non_Paye

ID	Nom	Prénom	Email	Téléphone	Salaire	Poste	Rôle	solde
26	rachid	test	crachid@em...	060006000	50000.01	Admin	Pilote	25
27	hamza	omar	h.omar@ema...	06306000	5000.01	Admin	Pilote	25
28	omar	omari	omaromari@...	060606060	100.0	Admin	Ingenieur_Et...	25