

Mid-term project :internship management System Using Java Swing

**School of Digital Engineering and Artificial
Intelligence: 3rd Year Big Data**

School year : 2024-2025

DONE BY :

- **IMANE MALIKI**

I. Introduction :

As part of our mid-term project, we designed and developed an application titled **Internship Management System**. This project aims to meet the needs of academic institutions and companies for the efficient management of student internships. The main objective of this system is to provide a centralized solution to organize and track information related to internships, students, supervisors, and partner companies. Through this application, users can register new internship opportunities, monitor student progress, manage assignments, and generate detailed reports. The development of this project was carried out in Java, using the Swing graphical library, which allowed us to create a user-friendly and interactive interface. This project also provided us with the opportunity to explore key concepts of object-oriented programming. This report outlines the process of designing, developing, and testing the system, while highlighting the challenges encountered and the solutions adopted to achieve our objectives.

II. Classes principales et leurs rôles :

1. Class Personne :

- Represents a general person with basic attributes: name, surname, and level of study.
- Serves as a parent class for other entities like Stagiaire (Trainee) or Parent.

➤ Code :

```
➤ package PK;
public abstract class Personne
{
    private String nom;
    private String prenom;
    public Personne(String nom, String prenom)
    {
        this.nom = nom;
        this.prenom = prenom;
    }
    public String getNom()
    {
        return nom;
    }
    public String getPrenom()
    {
        return prenom;
    }
    abstract void afficher();
}
```

2. Class Stagiaire (Trainee)

- Inherits from `Personne` and adds specific information like `currentInternship`.
- Implements a method to display the complete information of a trainee

➤ Code :

```
package PK;
import java.util.ArrayList;
import java.util.Scanner;
public class Stagiaire extends Personne
{
    private String niveauEtude;

    public Stagiaire(String nom, String prenom, String niveauEtude)
    {
        super(nom, prenom);
        this.niveauEtude = niveauEtude;
    }

    public String getNiveauEtude() {
        return niveauEtude;
    }

    public void afficher()
    {
        System.out.println(" le nom est : " + getNom()+" , Son prénom est : "+getPrenom()+" Son Niveau d'étude est : "+niveauEtude);
    }
}
```

3. Class Stage (Internship Management)

- Contains a list of trainees (`ArrayList`) and manages their addition, removal, and display.

➤ Code :

```
➤ package PK;

import java.util.*;

public class Stage
{
    private String titrestage;
```

```

private String Duree ;
private ArrayList<Stagiaire> liststage;

public Stage(String titrestage, String Duree)
{
    this.titrestage = titrestage;
    this.Duree = Duree;
    this.liststage= new ArrayList<>();
}
public String getTitrestage ()
{
    return this.titrestage;
}
public String getDuree() {
    return Duree;
}
public ArrayList<Stagiaire> getAssociestagiares ()
{
    return this.liststage;
}

public void afficher ()
{
    System.out.println("Titre de stage est : "+titrestage+"
Sa durée est : "+Duree );
    for (Stagiaire stag : liststage) {
        stag.afficher();
    }
}
public void AjouterStagiaire(Stagiaire stagiaire) {
    liststage.add(stagiaire);
}

}

```

4. Class Gestion

- It handles operations related to managing trainees (Stagiaire) and internships (Stage) and includes functionality for adding, removing, searching, and associating trainees with internships , using the two arraylist , liststage : A list to store all the internships (Stage) , liststagiaire : A list to store all the trainees (Stagiaire) , and using the following functionalities , ajouterStagiaire, Supprimerparnom, afficherStagiaire, ajouterstage, rechercherstage, associerStagiaireStage, Association, rechercherStagiaire .

➤ Code :

```

➤ package PK;
import java.util.*;
public class Gestion
{
    private ArrayList<Stage> liststage;
    private ArrayList<Stagiaire> liststagiaire;
    public Gestion ()
    {
        this.liststage = new ArrayList<>();
        this.liststagiaire = new ArrayList<>();
    }
    // ajouter info de stagiaire
    public void ajouterStagiaire()
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Nom : ");
        String nomStagiare = sc.nextLine();
        System.out.print("Prénom: ");
        String nomfamillstagiare = sc.nextLine();
        System.out.print("Niveau d'étude: ");
        String niveaуетude = sc.nextLine();

        for (Stagiaire stagiaire1 : liststagiaire) {
            if (stagiaire1.getNom().equals(nomStagiare) &&
                stagiaire1.getPrenom().equals(nomfamillstagiare) &&
                stagiaire1.getNiveauEtude().equals(niveaуетude)) {
                System.out.println("stagiaire exist ");
            }
        }
        Stagiaire stagiaire=new
        Stagiaire(nomStagiare,nomfamillstagiare,niveaуетude);
        liststagiaire.add(stagiaire);
    }
    //Supprimer les stagiers par nom
    public void Supprimerparnom()
    {
        Scanner sc=new Scanner(System.in);
        System.out.print("Entrer le nom : ");
        String nomStagiare = sc.nextLine();
        List<Stagiaire> aSupprimer = new ArrayList<>();

        for(Stagiaire stagiaire1 : liststagiaire) {
            if (stagiaire1.getNom().equals(nomStagiare)){
                aSupprimer.add(stagiaire1);
            }
        }
        // ila mdrтч haka o msht chi element fost la list lkh rin mn9drch
nwsllhom
        liststagiaire.removeAll(aSupprimer);
        // pour gestion d'errue
        if(aSupprimer.isEmpty())
        {
            System.out.println("aucun nom n'est trouvée ");
        }
        else
        {
            System.out.println("Stagaires supprimée !");
        }
    }
}

```

```

    }
    //afficher tout les stagerie
    public void afficherStagaire()
    {
        for (Stagiaire stag1 : liststagaire) {
            stag1.afficher();
        }
    }
    // 4 ajouter un stage
    public void ajouterstage()
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Titre du stage : ");
        String titrestage = sc.nextLine();
        System.out.print("Duree: ");
        String Duree = sc.nextLine();

        for (Stage stag1 : liststage) {
            if (stag1.getTitrestage().equals(titrestage) &&
                stag1.getDuree().equals(Duree)) {
                System.out.println("Stage deja dans la lise");
            }
        }
        Stage stag = new Stage(titrestage, Duree);
        liststage.add(stag); // Ajout à la liste des stages
        System.out.println("Stage ajouté avec succès !");
        System.out.println("Le nombre totale de stage Enregistrée est : 
        "+liststage.size());

    }
    // recstage
    public Stage rechercherstage()
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Nom du stage : ");
        String recherchnomstage = sc.nextLine();

        for (Stage stagel : liststage) {
            if (stagel.getTitrestage().equals(recherchnomstage) )
            {
                return stagel;
            }
        }
        return null;
    }

    // 5 asscosier
    public void associerStagaireStage() {
        Stagiaire stagiaire = rechercherStagiaire();
        Stage stage =rechercherstage();
        if(stagiaire != null && stage!= null)
        {
            stage.AjouterStagiaire(stagiaire);
            System.out.println("Association réussie :");
            System.out.println("- Stagiaire : " + stagiaire.getNom() + " 
            " + stagiaire.getPrenom());
            System.out.println("- Stage : " + stage.getTitrestage());
        }
    }
    //gestion d'erreur

```

```

        else
        {
            System.out.println("Stagiaire ou stage n'existe pas");
        }
    }
    // 6 affichage association
    public void Association ()
    {
        if (liststage.isEmpty())
        {
            System.out.println("Aucun stage enregistré.");
            return;
        }
        else
        {
            for (Stage stage : liststage) {
                System.out.println("Stage : " + stage.getTitrestage() + " |
Durée : " + stage.getDuree());
                System.out.println("Stagiaires associés :");
                if (stage.getAssociestagiares().isEmpty()) {
                    System.out.println("  Aucun stagiaire associé à ce
stage.");
                } else {
                    for (Stagiaire stagiaire : stage.getAssociestagiares()) {
                        System.out.println("    - " + stagiaire.getNom() + " "
+ stagiaire.getPrenom());
                    }
                }
                System.out.println("-----");
            }
        }

    }

    //7 Rechercher stagiaireparinfo
    public Stagiaire rechercherStagiaire () {
        Scanner sc = new Scanner(System.in);
        System.out.print("Rechercher un stagiaire: \n ");
        System.out.print("Nom : ");
        String recherchnom = sc.nextLine();
        System.out.print("Niveau d'étude : ");
        String recherchniveauEtude = sc.nextLine();
        for (Stagiaire stagiaire1 : liststagiaire) {
            if (stagiaire1.getNom().equals(recherchnom) &&
stagiaire1.getNiveauEtude().equals(recherchniveauEtude)) {
                return stagiaire1;
            }
        }
        return null;
    }
}

```

5. Class interfacegraphique :

- The InterfaceGraphique class is designed to provide a graphical user interface (GUI) for managing trainees (Stagiaire) and internships (Stage). It allows the user to interact with the system in a more intuitive way than using the command line. This class is built using the Swing library in Java, which is commonly used for creating GUI applications.

➤ Code :

```
➤ package PK;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.*;
public class Interfacegraphique {
    public static void afficherInterface() {
        Gestion gestion = new Gestion();
        JFrame frame = new JFrame("Gestion des Stagiaires et
Stages");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(500, 400);
        frame.setLayout(new BorderLayout());

        JPanel panel = new JPanel();
        panel.setLayout(new FlowLayout());
        frame.add(panel, BorderLayout.CENTER);
        frame.setVisible(true);

        // creee bouton
        JButton btnAjouterstagiaire = new JButton("1.Ajouter un
stagiaire : ");
        JButton btnSupprimerstagiaire = new
JButton("2.Supprimer un stagiaire : ");
        JButton btnaAfficherinfo = new JButton("3.afficher
info : ");
        JButton btnAjouterStage = new JButton("4.Ajouter Stage
: ");
        JButton btnAssocier = new JButton("5.Associer : ");
        JButton btnAfficher = new JButton("6.Afficher
Association : ");
        JButton btnRechercher = new JButton("7.Rechercher
Stager : ");
        JButton btnQuitter = new JButton("8.Quitter ");
        // ajouter bouton lpaneau
        panel.add(btnAjouterstagiaire);
        panel.add(btnSupprimerstagiaire);
        panel.add(btnaAfficherinfo);
        panel.add(btnAjouterStage);
        panel.add(btnAssocier);
        panel.add(btnAfficher);
        panel.add(btnRechercher);
    }
}
```



```

        panel.add(btnQuitter);
        btnAjouterstagaire.addActionListener(new
        ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                gestion.ajouterStagaire();
            }
        });
        btnSupprimerstagaire.addActionListener(new
        ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                gestion.Supprimerparnom();
            }
        });
        btnaAfficherinfo.addActionListener(new
        ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                gestion.afficherStagaire();
            }
        });
        btnAjouterStage.addActionListener(new ActionListener()
        {
            @Override
            public void actionPerformed(ActionEvent e) {
                gestion.ajouterstage();
            }
        });
        btnAssocier.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                gestion.associerStagaireStage();
            }
        });
        btnAfficher.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                gestion.Association();
            }
        });
        btnRechercher.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                gestion.rechercherStagiaire();
            }
        });
        btnQuitter.addActionListener(new ActionListener() {
            @Override

```

```

        public void actionPerformed(ActionEvent e) {
            System.exit(0);
        }
    });
}

```

6. Class Main :

- The `Main` class is the entry point of the application. It serves to launch the program by initializing the necessary components and starting the GUI or other core functionalities based on the user's needs.

➤ Code :

```

➤ package PK;
import java.util.*;;

public class Main
{
    public static void main(String[] args) {
        Gestion gestion=new Gestion();
        //Interfacegraphique.afficherInterface();

        Scanner scanner = new Scanner(System.in);
        int choice;
        do {
            System.out.println("1. Ajouter un stagiaire");
            System.out.println("2. Supprimer un stagiaire");
            System.out.println("3. Afficher tous les
stagiaires");
            System.out.println("4. Ajouter un stage ");
            System.out.println("5. Associer un stagiaire à un
stage ");
            System.out.println("6. Afficher tous les stages
avec les stagiaires associés");
            System.out.println("7. Rechercher un stagiaire");
            System.out.println("8. Quitter l'application");
            System.out.print("Choisissez une option: ");
            choice = scanner.nextInt();
            scanner.nextLine();
            switch (choice) {
                case 1:
                    gestion.ajouterStagaire();
                    break;

                case 2:
                    System.out.print("Supprimer Info : \n ");

                    gestion.Supprimerparnom();
                    break;

                case 3:
                    System.out.print("Affichage des infos : \n
");

```

```

        gestion.afficherStagiaire();
        break;
    case 4:
        System.out.print("---Ajouter Stage :---\n");

        gestion.ajouterstage();
        break;
    case 5 :
        System.out.println("-----Associer un stagiaire a un stage -----");
        gestion.associerStagiaireStage();

        break;
    case 6:
        System.out.println(" ----- Afficher tous les stages avec les stagiaires associés-----\n ");
        gestion.Association();
        break;
    case 7:
        System.out.print("---Rechercher un Stagaitre :--- \n");

        gestion.rechercherStagiaire();

        break;

    default:

        System.out.print("Option invalid , quitter programme : ");
        break;
    }
} while (choice != 8);

}

}

```