

Université Cadi Ayyad
École supérieure de technologie
Département informatique
Filière génie informatique

Rapport de TP3

Généricités, MVC, DAO et E/S

Réaliser par :

Imane El markhi

Encadré par :

Laila Elkhrof

Année universitaire : 2024/2025

Objectif :

Étendre l'application de gestion des congés en implémentant des mécanismes d'entrées/sorties pour permettre l'export et l'import des données tout en intégrant la notion des entrées/sorties suivant le modèle MVC, DAO.

- Étape 1: Gestion des données (DAO)

Creation d'un interface générique pour l'import/export DataImport Export et I implimentation de cette interface par la classe Employec DAOImpl.

```
package DAO;

import java.io.IOException;
import java.util.List;

import Model.Employe;

public interface DataImportExport<T> {
    void importData(String fileName) throws IOException;
    void exportData(String fileName, List<T> data) throws
IOException;
}
```

- Étape 2: Logique métier

Extension de classe Model (EmployeModeli pour gérer l'import/export)

```
@Override
    public void importData(String fileName) throws IOException {
        String query = "INSERT INTO employes (nom , prenom , email ,
telephone , salaire , role_id , poste_id) VALUES "
            + "(?,?,?,?,"
            + "(SELECT id FROM roles WHERE nom = ?),"
            + "(SELECT id FROM postes WHERE nom = ?))";
        try(BufferedReader reader = new BufferedReader(new
FileReader(fileName));
            PreparedStatement stmt =
Connexion.getConnection().prepareStatement(query)){

            String line = reader.readLine();
            while((line = reader.readLine()) != null) {
                String[] data = line.split(",");
                if(data.length == 7) {
                    stmt.setString(1, data[0].trim());
```

```

        stmt.setString(2, data[1].trim());
        stmt.setString(3, data[2].trim());
        stmt.setString(4, data[3].trim());
        stmt.setString(5, data[4].trim());
        stmt.setString(6, data[5].trim());
        stmt.setString(7, data[6].trim());
        stmt.addBatch();
    }
}

stmt.executeBatch();
System.out.println("Employees imported
successfully!");

} catch (IOException | SQLException e) {
    e.printStackTrace();
}

}

```

```

@Override
public void exportData(String fileName, List<Employee> data) throws
IOException {
    System.out.println(""+fileName);
    try(BufferedWriter writer = new BufferedWriter(new
FileWriter(fileName))){
        writer.write("nom, prenom, salaire, email, telephone, role,
poste");
        writer.newLine();
        for(Employee employee : data){
            String line = String.format("%s,%s,%.2f,%s,%s,%s,%s",
employee.getNom(),
employee.getPrenom(),
employee.getSalaire(),
employee.getEmail(),
employee.getTelephone(),
employee.getRole().name(),
employee.getPoste().name());
            writer.write(line);
            writer.newLine();
        }
    }
}

```

- Étape 3 : Interface graphique (couche View)

L'ajout des boutons d'import/export.

```
public JButton importerButton = new JButton("Importer");
public JButton ExporterButton = new JButton("Exporter");
```

- Étape 5 : Contrôleur (couche Controller).

Gestion des événements des boutons de l'exportation et l'importation.

```
this.view.importerButton.addActionListener(e-> handleImport());
this.view.ExporterButton.addActionListener(e-> handleExport());
```

```
private void handleImport() {
    JFileChooser fileChooser = new JFileChooser();
    fileChooser.setFileFilter(new
    FileNameExtensionFilter("Fichiers CSV", "txt") );

    if(fileChooser.showOpenDialog(view) ==
    JFileChooser.APPROVE_OPTION) {
        try {
            String filePath =
fileChooser.getSelectedFile().getAbsolutePath();
            model.importData(filePath);
            view.afficherMessageSucces("Importation
réussie");
        } catch (IOException e) {
            view.afficherMessageSucces("Erreur lors de
l'Importation:"+e.getMessage());
        }
    }
}

private void handleExport() {
    JFileChooser fileChooser = new JFileChooser();
    fileChooser.setFileFilter(new
    FileNameExtensionFilter("Fichiers CSV", "csv") );
    if(fileChooser.showOpenDialog(view) ==
    JFileChooser.APPROVE_OPTION) {
        try {
            String filePath =
fileChooser.getSelectedFile().getAbsolutePath();
            if(!filePath.toLowerCase().endsWith(".txt")) {
                filePath += ".txt";
            }
            List<Employe> employe = model.afficher();
            model.exportData(filePath, employe);
        } catch (IOException e) {
```

```
view.afficherMessageErreur("Erreur lors de  
l'Importation:"+e.getMessage());}}}
```

- Étape 6: Main

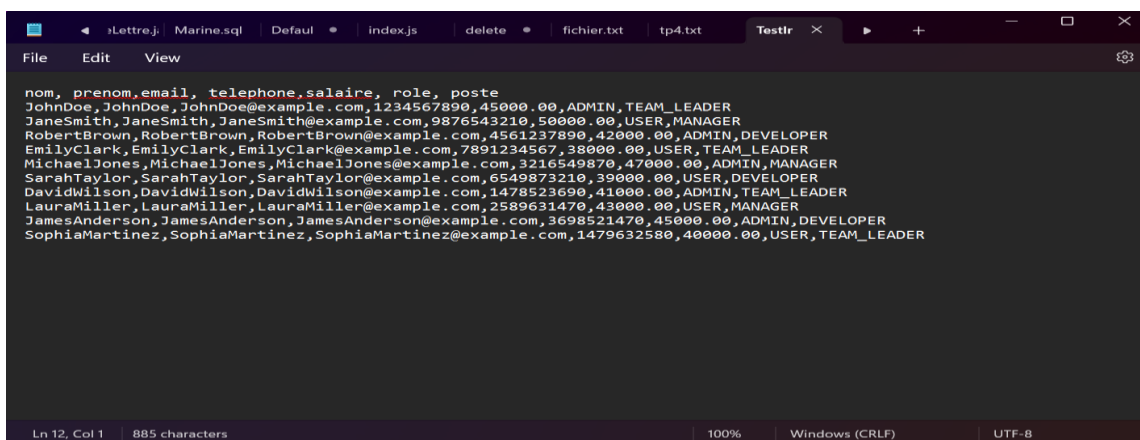
```
public class Main {  
    public static void main(String[] args) {  
        EmployeeView view = new EmployeeView();  
        HolidayView view2 = new HolidayView();  
        EmployeeDAOImpl dao = new EmployeeDAOImpl();  
        HolidayDAOImpl dao2 = new HolidayDAOImpl();  
        EmployeeModel model = new EmployeeModel(dao);  
        HolidayModel model2 = new HolidayModel(dao2);  
        new EmployeeController(view,model);  
        new HolidayController(view2,model2);  
        ManagementInterfaces combinedView = new  
ManagementInterfaces(view,view2);  
        combinedView.setVisible(true);  
    }  
}
```

L'implémentation :

- L'importation :

On va essayer de créer un fichier text qui va contenir les informations des employes, et le importer pour que ces employes vont être ajoutés dans la base de données.

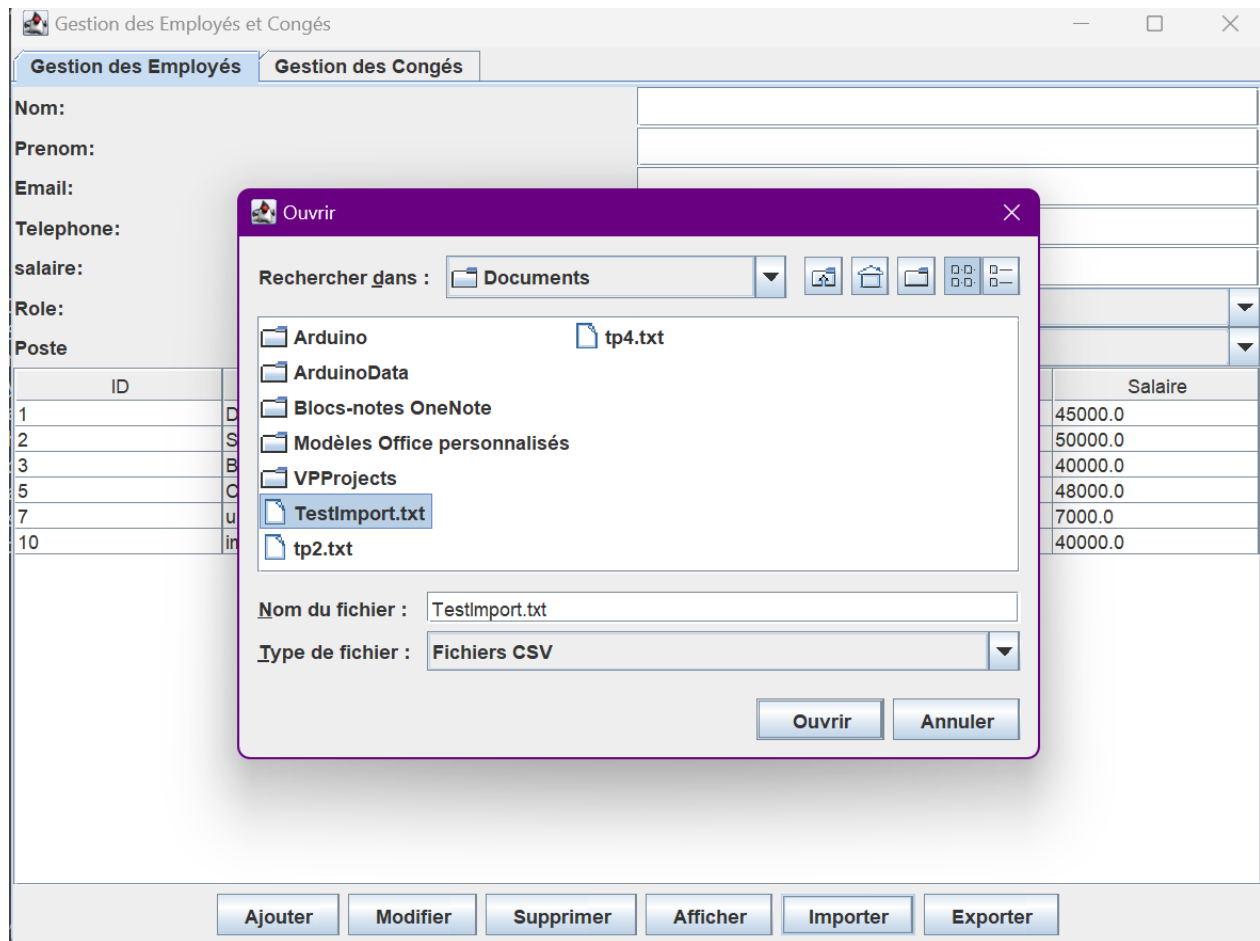
Le fichier crée ;



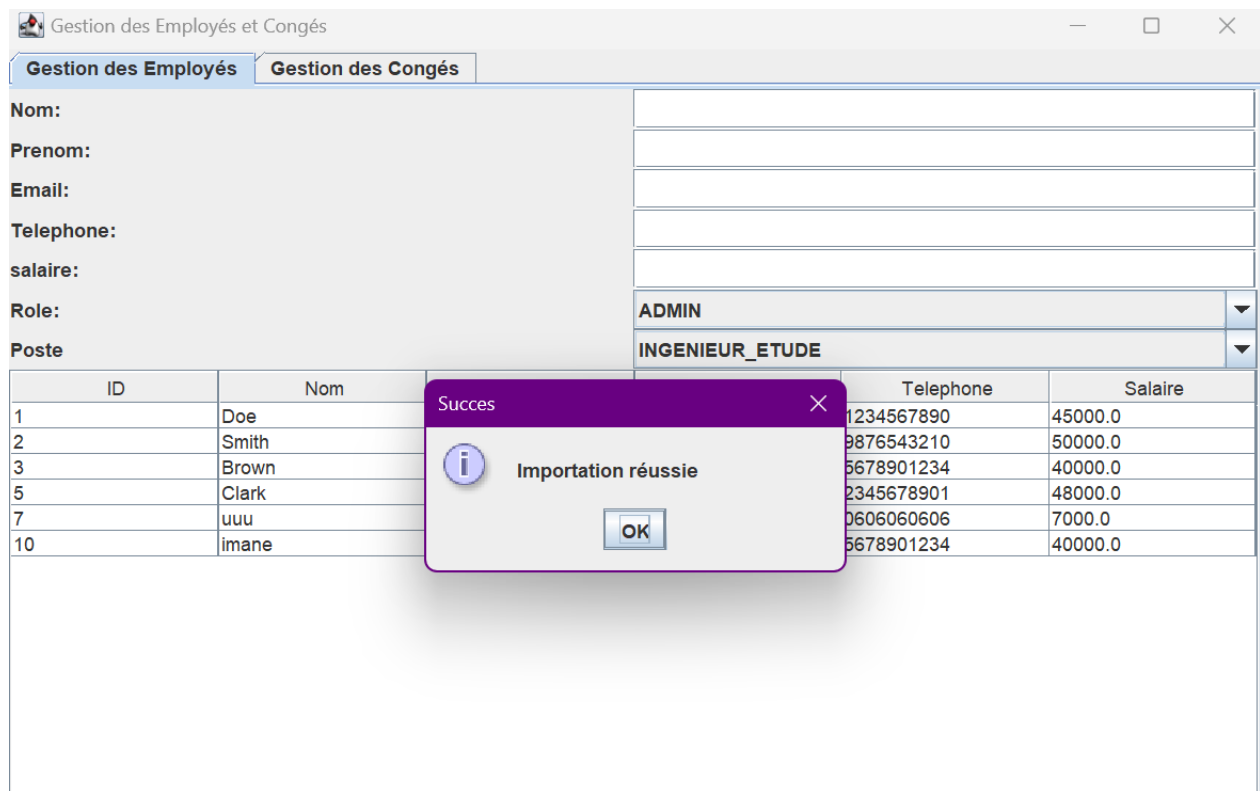
The screenshot shows a text editor window with a dark theme. The title bar includes tabs for 'iLettre.j', 'Marine.sql', 'Default', 'index.js', 'delete', 'fichier.txt', and 'tp4.txt'. The 'fichier.txt' tab is active. The editor content is a CSV file with 10 columns: nom, prenom, email, telephone, salaire, role, poste. It contains 10 rows of employee data. The status bar at the bottom indicates 'Ln 12, Col 1', '885 characters', '100%', 'Windows (CRLF)', and 'UTF-8'.

```
nom, prenom, email, telephone, salaire, role, poste  
JohnDoe, JohnDoe, JohnDoe@example.com, 1234567890, 45000.00, ADMIN, TEAM_LEADER  
JaneSmith, JaneSmith, JaneSmith@example.com, 9876543210, 50000.00, USER, MANAGER  
RobertBrown, RobertBrown, RobertBrown@example.com, 4561237890, 42000.00, ADMIN, DEVELOPER  
EmilyClark, EmilyClark, EmilyClark@example.com, 7891234567, 38000.00, USER, TEAM_LEADER  
MichaelJones, MichaelJones, MichaelJones@example.com, 3216549870, 47000.00, ADMIN, MANAGER  
SarahTaylor, SarahTaylor, SarahTaylor@example.com, 6549873210, 39000.00, USER, DEVELOPER  
DavidWilson, DavidWilson, DavidWilson@example.com, 1478523690, 41000.00, ADMIN, TEAM_LEADER  
LauraMiller, LauraMiller, LauraMiller@example.com, 2589631470, 43000.00, USER, MANAGER  
JamesAnderson, JamesAnderson, JamesAnderson@example.com, 3698521470, 45000.00, ADMIN, DEVELOPER  
SophiaMartinez, SophiaMartinez, SophiaMartinez@example.com, 1479632580, 40000.00, USER, TEAM_LEADER
```

L'importation : On va choisir le fichier qu'on a crée pour l'importer



Importation réussie :



Gestion des Employés et Congés

Gestion des Employés | **Gestion des Congés**

Nom:

Prenom:

Email:

Telephone:

salaire:

Role: **ADMIN**

Poste: **INGENIEUR_ETUDE**

ID	Nom	Prenom	Email	Telephone	Salaire
7	uuu	uuuuuu	Faith22@gmail.com	0606060606	7000.0
5	Clark	MichaelIII	michael.clark@exam...	2345678901	48000.0
1	Doe	John	john.doe@example.c...	1234567890	45000.0
2	Smith	Jane	jane.smith@example....	9876543210	50000.0
16	JamesAnderson	JamesAnderson	JamesAnderson@ex...	3698521470	45000.0
14	MichaelJones	MichaelJones	MichaelJones@exam...	3216549870	47000.0
13	RobertBrown	RobertBrown	RobertBrown@exam...	4561237890	42000.0
12	DavidWilson	DavidWilson	DavidWilson@examp...	1478523690	41000.0
11	JohnDoe	JohnDoe	JohnDoe@example.c...	1234567890	45000.0
10	imane	imane	imane.brown@exam...	5678901234	40000.0
3	Brown	Charlie	charlie.brown@exam...	5678901234	40000.0

Ajouter Modifier Supprimer Afficher Importer Exporter

- Exportation :

On va nommer le fichier qui va contient nos employe :

Gestion des Employés et Congés

Gestion des Employés | **Gestion des Congés**

Nom:

Prenom:

Email:

Telephone:

salaire:

Role:

Poste:

ID	Nom	Prenom	Email	Telephone	Salaire
7	uuu	uuuuuu	Faith22@gmail.com	0606060606	7000.0
5	Clark	MichaelIII	michael.clark@exam...	2345678901	48000.0
1	Doe	John	john.doe@example.c...	1234567890	45000.0
2	Smith	Jane	jane.smith@example....	9876543210	50000.0
16	JamesAnderson	JamesAnderson	JamesAnderson@ex...	3698521470	45000.0
14	MichaelJones	MichaelJones	MichaelJones@exam...	3216549870	47000.0
13	RobertBrown	RobertBrown	RobertBrown@exam...	4561237890	42000.0
12	DavidWilson	DavidWilson	DavidWilson@examp...	1478523690	41000.0
11	JohnDoe	JohnDoe	JohnDoe@example.c...	1234567890	45000.0
10	imane	imane	imane.brown@exam...	5678901234	40000.0
3	Brown	Charlie	charlie.brown@exam...	5678901234	40000.0

Ajouter Modifier Supprimer Afficher Importer Exporter

Ouvrir

Rechercher dans : **Documents**

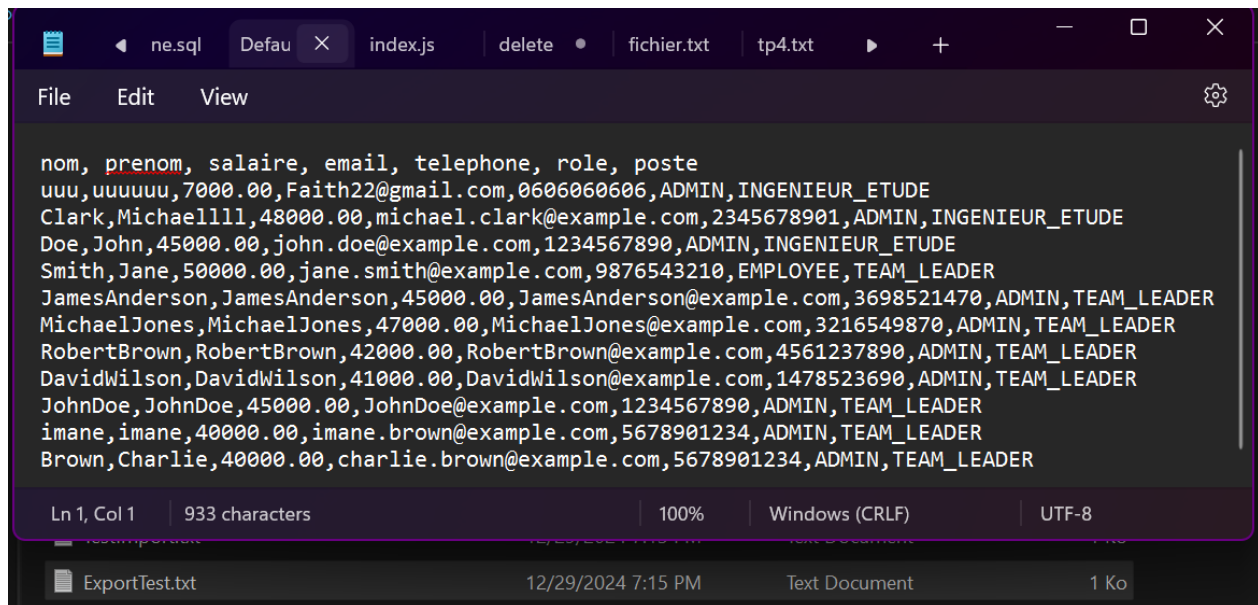
- Arduino
- ArduinoData
- Blocs-notes OneNote
- Modèles Office personnalisés
- VPPProjects

Nom du fichier : **ExportTest**

Type de fichier : **Fichiers CSV**

Ouvrir Annuler

Exportation réussie et le fichier est créée avec l'extention.txt :



The screenshot shows a code editor window with a dark theme. The top bar displays several tabs: 'ne.sql', 'Defau', 'index.js', 'delete', 'fichier.txt', and 'tp4.txt'. Below the tabs is a menu bar with 'File', 'Edit', and 'View'. The main text area contains a list of test data entries, each on a new line. The entries are formatted as CSV-like strings with fields separated by commas. The first line is a header: 'nom, prenom, salaire, email, telephone, role, poste'. The subsequent lines contain test data for various users, including 'Faith22@gmail.com', 'Michael Clark', 'John Doe', 'Jane Smith', 'James Anderson', 'Michael Jones', 'Robert Brown', 'David Wilson', 'John Doe', 'Imane', and 'Charlie Brown'. The status bar at the bottom indicates 'Ln 1, Col 1', '933 characters', '100%', 'Windows (CRLF)', and 'UTF-8'. A file explorer at the bottom shows a file named 'ExportTest.txt' with a size of '1 Ko'.

```
nom, prenom, salaire, email, telephone, role, poste
uuu,uuuuuu,7000.00,Faith22@gmail.com,0606060606,ADMIN,INGENIEUR_ETUDE
Clark,Michael111,48000.00,michael.clark@example.com,2345678901,ADMIN,INGENIEUR_ETUDE
Doe,John,45000.00,john.doe@example.com,1234567890,ADMIN,INGENIEUR_ETUDE
Smith,Jane,50000.00,jane.smith@example.com,9876543210,EMPLOYEE,TEAM_LEADER
JamesAnderson,JamesAnderson,45000.00,JamesAnderson@example.com,3698521470,ADMIN,TEAM_LEADER
MichaelJones,MichaelJones,47000.00,MichaelJones@example.com,3216549870,ADMIN,TEAM_LEADER
RobertBrown,RobertBrown,42000.00,RobertBrown@example.com,4561237890,ADMIN,TEAM_LEADER
DavidWilson,DavidWilson,41000.00,DavidWilson@example.com,1478523690,ADMIN,TEAM_LEADER
JohnDoe,JohnDoe,45000.00,JohnDoe@example.com,1234567890,ADMIN,TEAM_LEADER
imane,imane,40000.00,imane.brown@example.com,5678901234,ADMIN,TEAM_LEADER
Brown,Charlie,40000.00,charlie.brown@example.com,5678901234,ADMIN,TEAM_LEADER
```

Ln 1, Col 1 | 933 characters | 100% | Windows (CRLF) | UTF-8

ExportTest.txt | 12/29/2024 7:15 PM | Text Document | 1 Ko