

Projet Registre de matériels

Introduction

L'objectif principal du Projet Registre de matériels est de construire une interface graphique permettant à l'utilisateur de remplir un formulaire d'entrée/sortie de stock de matériels consommables. A la validation du formulaire, ces données doivent être saisies automatiquement dans un fichier Excel dédié.

Caractéristiques du projet :

- ☐ Interface graphique avec un formulaire d'entrée/sortie de matériels de la boutique
- ☐ Panneau de saisi d'une nouvelle personne
- ☐ Panneau de saisi d'un nouveau matériel avec sa description
- ☐ Enregistrement des entrées/sorties de matériels dans un fichier Excel
- ☐ Transfert des données du tableau Excel dans un panneau dédié de l'application
- ☐ Panneau inventaire de matériels (gestion de stock)

Aspects spécifiques :

Langages : Java version 8, SQL

Logiciel de développement : Netbeans IDE 8.2

Outil de communication des tâches à faire, en cours et terminées : Trello

Ressources utilisées : projets effectués dans le cadre de mes études, recherche de solutions techniques sur Internet

Résolution technique :

J'ai créé le projet de registre des matériels avec l'outil logiciel Apache Maven permettant d'optimiser certaines tâches de gestion et la construction (build) des projets Java entre autres. Le projet est configuré par un POM (Project Object Model) dans un fichier XML. Ce fichier contient des informations sur le projet, notamment de configuration, et les bibliothèques (JAR) permettant à Maven de s'occuper de la gestion des dépendances en construisant le projet.

```

<build>
  <finalName>Registro</finalName>
</build>
<dependencies>
  <dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>8.0.28</version>
  </dependency>
  <dependency>
    <groupId>org.apache.poi</groupId>
    <artifactId>poi</artifactId>
    <version>5.2.2</version>
  </dependency>
  <dependency>
    <groupId>org.apache.poi</groupId>
    <artifactId>poi-ooxml</artifactId>
    <version>5.2.2</version>
  </dependency>
  <dependency>
    <groupId>commons-io</groupId>
    <artifactId>commons-io</artifactId>
    <version>2.11.0</version>
  </dependency>
  <dependency>
    <groupId>org.apache.poi</groupId>
    <artifactId>poi-ooxml-schemas</artifactId>
    <version>4.1.2</version>
  </dependency>
</dependencies>

```

Image 1 : Extrait du fichier *pom.xml* avec la liste des dépendances utiles au projet

Afin de stocker les noms des personnes et les noms des matériels dans les listes déroulantes respectives du formulaire d'entrée/sortie de l'interface graphique, j'ai décidé de créer une base de données relationnelle avec trois tables.

Table	Action
<input type="checkbox"/> material	★ Parcourir Structure Rechercher Insérer Vider Supprimer
<input type="checkbox"/> materialunits	★ Parcourir Structure Rechercher Insérer Vider Supprimer
<input type="checkbox"/> person	★ Parcourir Structure Rechercher Insérer Vider Supprimer
3 tables	Somme
↑ <input type="checkbox"/> Tout cocher	Avec la sélection : ▼

Image 2 : Les trois tables de la base de données *registroDB*

```

public ArrayList<Person> getPerson(){
    ArrayList<Person> list1 = new ArrayList<Person>();

    Person p;
    try {

        Class.forName("com.mysql.cj.jdbc.Driver");
        // create a connection object
        Connection conn = DriverManager.getConnection(
            "jdbc:mysql://localhost:3306/registroDB?zeroDateTimeBehavior=convertToNull", "root", "");

        // create a declaration object
        stmt = conn.createStatement();

        String sql= "select name from person";
        resultSet = stmt.executeQuery(sql);

        while (resultSet.next()) {
            p = new Person(
                resultSet.getString("name")
            );
            list1.add(p);
        }
        conn.close();

    } catch (SQLException e) {
        System.out.println("Unable to make connection with listasDB");
        e.printStackTrace();
    } catch (ClassNotFoundException ex) {
        Logger.getLogger(MyRegistro.class.getName()).log(Level.SEVERE, null, ex);
    }
    return list1;
}

```

Image 3 : Collecte des noms de la table *person* issus de la base de données *registroDB* pour les insérer dans l'ArrayList (liste dynamique) de l'objet *Person*

Les noms des personnes effectuant les transactions ajoutés à l'ArrayList sont ensuite insérés dans la liste déroulante dédiée du formulaire d'entrée/sortie de matériels.

Pour la communication avec un fichier Excel, j'ai dû inclure les dépendances Apache POI dans le fichier *pom.xml* pour que Maven puisse générer des fichiers de type Excel.

Dans la fonction qui remplit le contenu de la feuille Registro, j'ai créé le nouveau tableau avec les titres des colonnes et leur contenu. J'ai saisi les titres des colonnes par ordre d'apparition dans le tableau de gauche à droite : les colonnes sont numérotées de cellule 0 à cellule n. Pour chaque colonne numérotée, j'ai appelé les getters d'attribut d'objet pour récupérer les valeurs d'attribut d'objet non nulles de chaque transaction correspondant aux titres des colonnes.

```

public void addElementsToExcel() {

    ArrayList<Elements> list3 = getElements();

    try {
        //name of excel file
        String excelFileName = "C:/Users/LEHUCHER/Downloads/Registro de material de oficina.xlsx";

        //Creating input stream
        FileInputStream inputStream = new FileInputStream(excelFileName);

        //Create workbook in .xlsx format
        XSSFWorkbook workbook = (XSSFWorkbook) WorkbookFactory.create(inputStream);

        //Reading first sheet of excel file
        XSSFSheet sheet = workbook.getSheetAt(0);

        //Create top row with column headings
        String[] columnHeadings = {"Fecha y hora", "Acción", "Material", "Número de unidades", "Persona"};
        //foreground color.
        XSSFFont headerFont = workbook.createFont();
        headerFont.setBold(true);
        headerFont.setFontHeightInPoints((short)12);
        headerFont.setColor(IndexedColors.BLACK.index);
        //Create a CellStyle with the font
        CellStyle headerStyle = workbook.createCellStyle();
        headerStyle.setFont(headerFont);
        headerStyle.setFillPattern(FillPatternType.SOLID_FOREGROUND);
        headerStyle.setFillForegroundColor(IndexedColors.GREY_25_PERCENT.index);
        //Create the header row
        XSSFRow headerRow = (XSSFRow) sheet.createRow(0);
        //Iterate over the column headings to create columns
        for(int i=0;i<columnHeadings.length;i++) {
            XSSFCell cell = headerRow.createCell(i);
            cell.setCellValue(columnHeadings[i]);
            cell.setCellStyle(headerStyle);
        }

        //Getting the count of existing records
        int rowCount = sheet.getLastRowNum();

        for(Elements e : list3) {

            XSSFRow row = sheet.createRow(++rowCount);

            row.createCell(0).setCellValue(e.getDatetime());
            row.createCell(1).setCellValue(e.getAction());
            row.createCell(2).setCellValue(e.getMaterial());
            row.createCell(3).setCellValue(e.getNumUnits());
            row.createCell(4).setCellValue(e.getPerson());

        }

        //Autosize columns
        for(int i=0;i<columnHeadings.length;i++) {
            sheet.autoSizeColumn(i);
        }

        //Close input stream
        inputStream.close();

        //Write the output to file
        FileOutputStream fileOut = new FileOutputStream(excelFileName);
        workbook.write(fileOut);
        JOptionPane.showMessageDialog(this, "Adición exitosa!");
        fileOut.close();
        workbook.close();

    }
    catch(Exception ex) {
        ex.printStackTrace();
    }
}

```

Image 4 : Insertion d'un tableau et de ses données dans la première feuille du fichier Excel

Résultats obtenus :

Image 5 : Visuel de l'interface graphique comprenant le formulaire de transaction d'entrée/sortie de matériels

	A	B	C	D	E
1	Fecha y hora	Acción	Material	Número de unidades	Persona
2	04/04/2022 12:18:29	Añadir material	Lápiz HD	3	Sebastian
3	04/04/2022 12:19:12	Añadir material	Bolígrafo azul	6	Anastasia
4	04/04/2022 12:20:01	Añadir material	Lápiz HD	2	Manuel
5	04/04/2022 12:21:12	Añadir material	Bolígrafo azul	1	José
6	04/04/2022 12:21:49	Retirar material	Bolígrafo azul	2	Manuel
7	04/04/2022 12:22:52	Retirar material	Bolígrafo azul	2	Sebastian
8	04/04/2022 18:01:05	Añadir material	Ratón APPLE	0	José
9					
10					
11					
12					
13					
14					

Image 6 : liste dans le fichier Excel des transactions effectuées d'entrée/sortie de matériels

L'application de registre de matériels avec le formulaire d'entrée/sortie de matériels envoie avec succès les données au fichier Excel. J'ai créé une base de données en local pour que l'application garde en mémoire les prénoms, les matériels saisis par l'utilisateur et nécessaires au remplissage du formulaire. Cependant, dans le tableau d'inventaire, je n'ai pas réussi la requête SQL calculant une différence de deux sommes d'unités de matériels en fonction de l'action : soit ajouter matériels, soit retirer matériels.