



Micro Projet : JDBC

Préparer par :
Encadrer par :

Elhamidi Maroua
Karami Fahd

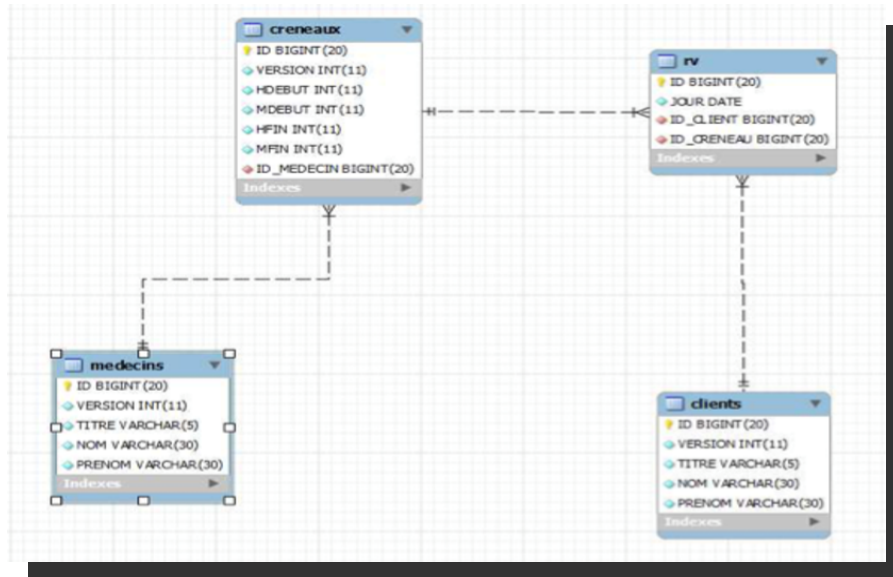
Année universitaire 2019/2020

Introduction :

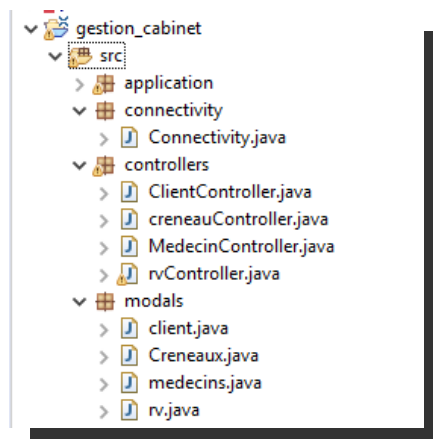
Java permet donc aux programmeurs d'écrire du code qui met en œuvre les requêtes SQL pour retrouver des renseignements dans des bases de données relationnelles, mais ce n'est pas tout. Le langage Java est un langage universel qui peut fonctionner sur différentes plates-formes.

Dans ce sens le rapport est une description d'un micro projet sert à gérer un cabinet en utilisant JAVA FX et le SJBD MYSQL et le patron de conception DAO.

Gestion d'un cabinet:



Patron de conception DAO (implémentation):



Mon projet se compose par une structure de model **MVC** :

Repo connectivity : sert à lier notre application avec la base de données en utilisant un driver MySQLconnector .

M (modals) : j'ai implémenté les tables de la bases de données dans des classe JAVA .

En-dessous je cite quelque exemple de nos modals :

Classe Creneaux :

```

2
3 public class Creneaux {
4     private int id;
5     private int version;
6     private int HDebut;
7     private int MDebut;
8     private int HFin;
9     private int Mfin;
10    private int ID_medecine;
11
12    public Creneaux(int id, int version, int hDebut, int mDebut, int hFin, int mfin, int id_medecine) {
13        super();
14        this.id = id;
15        this.version = version;
16        HDebut = hDebut;
17        MDebut = mDebut;
18        HFin = hFin;
19        Mfin = mfin;
20        ID_medecine = id_medecine;
21    }
22    public int getId() {
23        return id;
24    }
25    public void setId(int id) {
26        this.id = id;
27    }
28    public int getVersion() {
29        return version;
30    }
31    public void setVersion(int version) {
32        this.version = version;
33    }
34    public int getHDebut() {
35        return HDebut;
36    }
37    public void setHDebut(int hDebut) {
38        HDebut = hDebut;
39    }
40

```

```

41    public int getHFin() {
42        return HFin;
43    }
44    public void setHFin(int hFin) {
45        HFin = hFin;
46    }
47    public int getMfin() {
48        return Mfin;
49    }
50    public void setMfin(int mfin) {
51        Mfin = mfin;
52    }
53    public int getID_medecine() {
54        return ID_medecine;
55    }
56    public void setID_medecine(int id_medecine) {
57        ID_medecine = id_medecine;
58    }
59

```

Classe Rv :

```

public class rv {
    private int id;
    private int jour;
    private int id_client;
    private int id_creneau;
    public rv(int id, int jour, int id_client, int id_creneau) {
        super();
        this.id = id;
        this.jour = jour;
        this.id_client = id_client;
        this.id_creneau = id_creneau;
    }
    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
    public int getJour() {
        return jour;
    }
    public void setJour(int jour) {
        this.jour = jour;
    }
    public int getId_client() {
        return id_client;
    }
    public void setId_client(int id_client) {
        this.id_client = id_client;
    }
    public int getId_creneau() {
        return id_creneau;
    }
    public void setId_creneau(int id_creneau) {
        this.id_creneau = id_creneau;
    }
}

```

V (views): est notre application qui va nous rediriger vers nos interfaces .

C (Controllers): est le dossier Controllers qui sert à contrôler nos tables contenant les méthodes que nous allons utiliser comme (add,edit,list...)

Exemple rvController / la méthode liste :

La méthode list() sert à lister les rendez-vous qui sont existe dans la base de données .

```

public void list() {
    ObservableList<rv> rvList = FXCollections.observableArrayList();
    PreparedStatement pst;
    ResultSet rs;

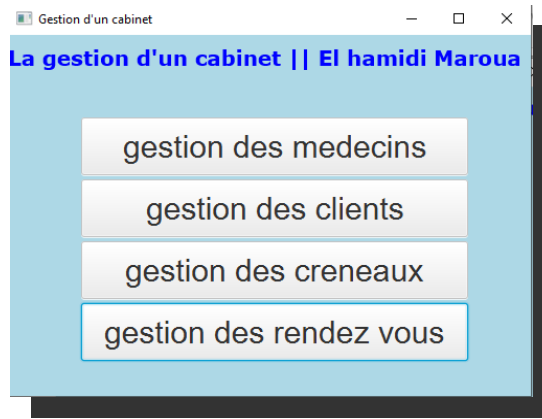
    String req = "select*from rv";

    try {
        pst=con.prepareStatement(req);
        rs=pst.executeQuery();
        while(rs.next()) {
            rv r = new rv(rs.getInt(1),rs.getInt(2),rs.getInt(3),rs.getInt(4));
            rvList.add(r);
            ListView<rv> listViewOfRv = new ListView<>(rvList);
        }
    } catch (SQLException e1) {
        e1.printStackTrace();
    }
}

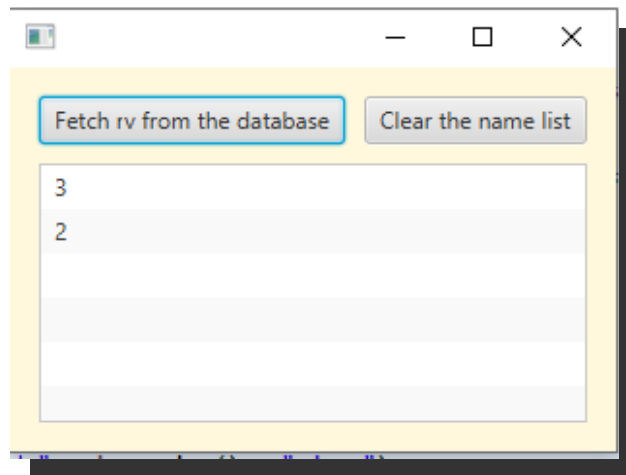
```

Démonstration :

Interface d'accueil :



La liste des jour de rendez vous :



Conclusion :

Le micro projet sert à nous donner une chance pour qu'on puisse appliquer la notion JDBC en utilisant un patron de conception DAO qui sert à mettre notre codes plus compréhensible , et claire .

