

# COMPTE RENDU PROJET CPO *Partie Finale*

## **Rendez-vous médicaux**

Remis le 02/05/2018

*Vithusan Sutharsan  
Fabio Ranjit*





## Table des matières

• Introduction.....	3
• Listing du programme.....	4
• Jeu d'essai.....	50
• Conclusion.....	64

## Introduction

Le premier semestre a été l'occasion pour nous d'apprendre les bases du langage Java et notamment des notions nouvelles telles que les classes, l'héritage ou encore les collections d'objets que nous n'avions pas vues en première année. Ce projet s'est divisé en deux parties : premièrement, nous nous sommes penchés sur le code typé console, puis secondement sur l'implémentation d'une interface graphique afin de créer quelque chose de visuel se rapprochant d'un site internet.

Nous avons choisi comme sujet de projet celui des **rendez-vous médicaux** qui consiste à créer un site internet proposant aux utilisateurs :

- de rechercher des médecins, prendre et consulter des rendez-vous si ce sont des patients,
- de prendre, consulter ses rendez-vous si ce sont des médecins.
- Être en mesure de supprimer un rendez-vous préalablement réservé.

Pour cela, le site propose aux utilisateurs de se créer un compte pour qu'ils puissent s'y connecter, les personnes non inscrites ayant seulement la possibilité de consulter les médecins disponibles.

## Listing du programme

D:\Documents\EPF\2A\Projet Java\RDV\_MEDIC\_SUTH\_Ranjit\RDV\_MEDIC\_SUTH\_Ranjit\inscription\JAVA\_FINAL - Copie - Copie\JAVA\_FINAL\RDV\_MEDICAUX\_SUTHARSAN\_RANJIT\src\rdv\_medicaux\_sutharsan\_ranjit\Date.java

```

1 /*
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6 package rdv_medicaux_sutharsan_ranjit;
7
8 /**
9  *
10 * @author franjit
11 */
12 public class Date {
13
14     String nomDoc;
15     String nomPatient;
16     int numsocialDoc;
17     int numsocialPatient;
18     int date;
19
20     public Date(String nomDoc, int numsocialDoc, String nomPatient, int numsocialPatient, int date){
21
22         this.nomDoc = nomDoc;
23         this.numsocialDoc = numsocialDoc;
24         this.nomPatient = nomPatient;
25         this.numsocialPatient = numsocialPatient;
26         this.date = date;
27     }
28
29     public String getNomDoc() {
30         return nomDoc;
31     }
32
33     public String getNomPatient() {
34         return nomPatient;
35     }
36
37     public int getNumsocialDoc() {
38         return numsocialDoc;
39     }
40
41     public int getNumsocialPatient() {
42         return numsocialPatient;
43     }
44
45     public int getDateRDV() {
46         return date;
47     }
48
49
50 }
51

```

D:\Documents\EPF\2A\Projet Java\RDV\_MEDIC\_SUTH\_Ranjit\RDV\_MEDIC\_SUTH\_Ranjit\inscription\JAVA\_FINAL - Copie - Copie\JAVA\_FINAL\RDV\_MEDICAUX\_SUTHARSAN\_RANJIT\src\rdv\_medicaux\_sutharsan\_ranjit\ErrorIndexMenu.java

```

1 /*
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6 package rdv_medicaux_sutharsan_ranjit;
7

```

```

8 /**
9 *
10 * @author vsuthars
11 */
12 public class ErrorIndexMenu extends Exception{
13
14     public ErrorIndexMenu(){
15
16     }
17     public ErrorIndexMenu(String msg){
18         super(msg);
19     }
20 }
21

```

D:\Documents\EPF\2A\Projet Java\RDV\_MEDIC\_SUTH\_Ranjit\RDV\_MEDIC\_SUTH\_Ranjit\inscription\JAVA\_FINAL - Copie - Copie\JAVA\_FINAL\RDV\_MEDICAUX\_SUTHARSAN\_RANJIT\src\rdv\_medicaux\_sutharsan\_ranjit\Praticien.java

```

1 /*
2 * To change this license header, choose License Headers in Project Properties.
3 * To change this template file, choose Tools | Templates
4 * and open the template in the editor.
5 */
6 package rdv_medicaux_sutharsan_ranjit;
7
8 import java.util.ArrayList;
9
10 /**
11 *
12 * @author franjit
13 */
14 public class Praticien extends Utilisateurs{
15
16     private String ville;
17     private int specialite;
18     private ArrayList AgendaRDV = new ArrayList();
19
20     public Praticien(String prenom, String nom, int num, int datedenaissance, String pseudo, String mdp, int specialite, ArrayList
    AgendaRDV, String ville){
21
22         super(prenom, nom, num, datedenaissance, pseudo, mdp);
23         this.specialite = specialite;
24         this.AgendaRDV = AgendaRDV;
25         this.ville = ville;
26     }
27
28     public int getNumtelephone(){
29         return super.getNum();
30     }
31
32     public String getPseudo(){
33         return super.getIdentifiant();
34     }
35
36     public String getMdp(){
37         return super.getMotdepasse();
38     }
39
40     public String getName(){
41         return super.getNom();
42     }
43
44     public int getSpecialite(){
45         return specialite;
46     }
47
48     public ArrayList getAgendaRDV(){
49         return AgendaRDV;
50     }
51
52     public void changertype(ArrayList a){
53         this.AgendaRDV = a;
54     }
55

```

```

56 public String getVille() {
57     return ville;
58 }
59
60 }
61

```

D:\Documents\EPF\2A\Projet Java\RDV\_MEDIC\_SUTH\_Ranjit\RDV\_MEDIC\_SUTH\_Ranjit\inscription\JAVA\_FINAL - Copie - Copie\JAVA\_FINAL\RDV\_MEDICAUX\_SUTHARSAN\_RANJIT\src\rdv\_medicaux\_sutharsan\_ranjit\Patient.java

```

1 /*
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6 package rdv_medicaux_sutharsan_ranjit;
7
8 /**
9  *
10  * @author franjit
11  */
12 public class Patient extends Utilisateurs{
13
14     private int social;
15
16     public Patient(String prenom, String nom, int num, int datedenaissance, String pseudo, String mdp, int social){
17
18         super(prenom, nom, num, datedenaissance, pseudo, mdp);
19         this.social = social;
20     }
21
22     public String getPseudo(){
23         return super.getIdentifiant();
24     }
25
26     public String getMdp(){
27         return super.getMotdepasse();
28     }
29     public int getSocial(){
30         return social;
31     }
32
33     public String getNaming(){
34         return super.getNom();
35     }
36
37     public int getNumsocial(){
38         return super.getNum();
39     }
40 }
41
42

```

D:\Documents\EPF\2A\Projet Java\RDV\_MEDIC\_SUTH\_Ranjit\RDV\_MEDIC\_SUTH\_Ranjit\inscription\JAVA\_FINAL - Copie - Copie\JAVA\_FINAL\RDV\_MEDICAUX\_SUTHARSAN\_RANJIT\src\rdv\_medicaux\_sutharsan\_ranjit\Utilisateurs.java

```

1 /*
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6 package rdv_medicaux_sutharsan_ranjit;
7
8 import java.io.BufferedReader;
9 import java.io.FileInputStream;
10 import java.io.FileNotFoundException;
11 import java.io.FileOutputStream;
12 import java.io.FileReader;
13 import java.io.FileWriter;
14 import java.io.IOException;
15 import java.io.ObjectInputStream;
16 import java.io.ObjectOutputStream;
17 import java.util.ArrayList;

```

```

18 import java.util.Iterator;
19 import java.util.Scanner;
20 import javax.swing.DefaultListModel;
21 import javax.swing.JList;
22 import javax.swing.JOptionPane;
23
24 /**
25 *
26 * @author vsuthars
27 */
28 public class Utilisateurs{
29
30     private String prenom;
31     private String nom;
32     private int num;
33     private int datedenaissance;
34     private String identifiant;
35     private String motdepasse;
36     private ArrayList listpatient = new ArrayList();
37     private ArrayList listpraticien = new ArrayList();
38     private ArrayList listtamponpatient = new ArrayList();
39     private ArrayList listtamponpraticien = new ArrayList();
40     private ArrayList listRDV = new ArrayList();
41     //private String nomFichier;
42     private int a = 0;
43     public static int r = 0; //permet de gérer la méthode stock a revoir
44
45
46     public Utilisateurs(String prenom, String nom, int num, int datedenaissance, String identifiant, String motdepasse){
47
48         this.prenom = prenom;
49         this.nom = nom;
50         this.datedenaissance = datedenaissance;
51         this.num = num;
52         this.identifiant = identifiant;
53         this.motdepasse = motdepasse;
54     }
55
56
57     public void creationPraticien(){ //méthode qui va créer un spécialiste et un généraliste au début
58
59         int i = 20180312;
60         int j = 20180425;
61         int k = 20180212;
62         int l = 20181007;
63         int m = 20180606;
64         int n = 20180829;
65         int o = 20181125;
66         ArrayList AgendaRDV = new ArrayList();
67         ArrayList AgendaRDVbis = new ArrayList();
68         ArrayList AgendaRDVter = new ArrayList();
69         ArrayList AgendaRDVquad = new ArrayList();
70         AgendaRDV.add(j);
71         AgendaRDV.add(i);
72         AgendaRDVbis.add(j);
73         AgendaRDVbis.add(k);
74         AgendaRDVter.add(l);
75         AgendaRDVter.add(m);
76         AgendaRDVter.add(n);
77         AgendaRDVquad.add(m);
78         AgendaRDVquad.add(n);
79         AgendaRDVquad.add(o);
80         Praticien praticien1 = new Praticien("Jean","Dupont",0505050505,19621201,"Jean","Kevin",1,AgendaRDV, "Paris");
81         listpraticien.add(praticien1);
82         Praticien praticien3 = new Praticien("Paul","Eric",0606060606,19621201,"Paul","Kevin",1,AgendaRDVbis, "Paris");
83         listpraticien.add(praticien3);
84         Praticien praticien4 = new Praticien("Mika","Pierre",123,19621201,"Mika","Kevin",1,AgendaRDVter, "Lyon");
85         listpraticien.add(praticien4);
86         Praticien praticien2 = new Praticien("Michel","Francois",0707070707,19700623,"Michel","Bertrand",2,AgendaRDVquad,
"Lyon");
87         listpraticien.add(praticien2);
88         Patient patient1 = new Patient("Fabio","Ranjit",0303030303,19980118,"Fabio","Ranjit", 1234);
89         listpatient.add(patient1);
90         Date date = new Date("Francois",0606060606,"Fabio",1234,20180316);

```



```

91     listRDV.add(date);
92 }
93
94 public int inscription(int indication, String nom, String prenom, int datenaissance, int numerotel, String id, String mdp, int
numsocial, int spec, String ville){ //méthode d'inscription
95
96     int i = 0;
97     //boolean b = true;
98     String prenom1;
99     String nom1;
100    String ident;
101    String motdp;
102    int date;
103    int numero;
104    int social;
105    int reussite = 2;
106    int specialite;
107    ArrayList AgendaRDV = new ArrayList();
108    i = indication;
109    prenom1 = prenom;
110    nom1 = nom;
111    date = datenaissance;
112    numero = numerotel;
113    ident = id;
114    motdp = mdp;
115    social = numsocial;
116    specialite = spec;
117
118    if(i==1){
119        reussite = 1;
120        Iterator<Patient> it = listpatient.iterator(); //création d'un itérateur
121        while(it.hasNext()){
122            Patient a = it.next();
123            if(a.getPseudo().equals(id)==true) //si un compte a le meme pseudo
124                JOptionPane.showMessageDialog(null, "Pseudo déjà utilisé", "Message d'avertissement",
JOptionPane.WARNING_MESSAGE); //on indique de recommencer
125            reussite = 0;
126            Patient pat = new Patient(prenom1,nom1,numero,date,ident, motdp,social); //on crée le patient avec les données
demandées
127            listpatient.add(pat); //on l'ajoute à la liste
128            listtamponpatient.add(pat); //on l'ajoute à une liste tampon qui nous permettra de savoir les détails de l'usager connecté
129            JOptionPane.showMessageDialog(null, "Inscription réussie", "Message d'information",
JOptionPane.INFORMATION_MESSAGE);
130        }
131    }
132    else{
133        reussite = 1;
134        Iterator<Praticien> its = listpraticien.iterator(); //création d'un itérateur
135        while(its.hasNext()){
136            Praticien a = its.next();
137            if(a.getPseudo().equals(id)==true) //si un compte a le meme pseudo
138                JOptionPane.showMessageDialog(null, "Pseudo déjà pris", "Message d'avertissement",
JOptionPane.WARNING_MESSAGE); //on indique de recommencer
139            reussite = 0;
140        }
141        Praticien prat = new Praticien(prenom1,nom1,numero,date,ident,motdp,specialite, AgendaRDV, ville); // on crée le
praticien avec les données demandées
142        listpraticien.add(prat); //on l'ajoute à la liste
143        listtamponpraticien.add(prat); //on l'ajoute à une liste tampon qui nous permettra de savoir les détails de l'usager
connecté
144        JOptionPane.showMessageDialog(null, "Inscription Reussie", "Message d'information",
JOptionPane.INFORMATION_MESSAGE);
145    }
146    return reussite;
147 }
148
149 public int connexion(int indicPraticien, String id,String mdp){ //méthode de connexion
150
151     int i;
152     int reussite = 0;
153     boolean connexionok = false;
154     i = indicPraticien;
155     if(i==1){ //on va dans l'arraylist du patient
156         Iterator<Patient> it = listpatient.iterator(); //création d'un itérateur

```

```

157     while(it.hasNext()){
158         Patient a = it.next();
159         a.getPseudo();
160         if(a.getPseudo().equals(id)==true){
161             if(a.getMdp().equals(mdp)==true){
162                 connexionok = true; //connexionok devient true si et seulement si l'utilisateur entre le bon pseudo et mdp
163                 listtamponpatient.add(a); //on l'ajoute à une liste tampon qui nous permettra de savoir les détails de l'utilisateur
connecté
164             }
165         }
166     }
167     if(connexionok == false){
168         JOptionPane.showMessageDialog(null,"Erreur de connexion!", "Erreur", JOptionPane.ERROR_MESSAGE);
169     }
170 }
171 else{ // on va dans l'arraylist du praticien
172     Iterator<Praticien> its = listpraticien.iterator(); //création d'un itérateur
173     while(its.hasNext()){
174         Praticien a = its.next();
175         a.getPseudo();
176         if(a.getPseudo().equals(id)==true){
177             if(a.getMdp().equals(mdp)==true){
178                 connexionok = true; //connexionok devient true si et seulement si l'utilisateur entre le bon pseudo et mdp
179                 listtamponpraticien.add(a); //on l'ajoute à une liste tampon qui nous permettra de savoir les détails de l'utilisateur
connecté
180             }
181         }
182     }
183     if(connexionok == false){
184         JOptionPane.showMessageDialog(null,"Erreur de connexion !", "Erreur", JOptionPane.ERROR_MESSAGE);
185     }
186 }
187
188 if(connexionok == true){
189     //System.out.println("Connexion réussie");
190     JOptionPane.showMessageDialog(null,"Connexion réussie !", "Message d'information",
JOptionPane.INFORMATION_MESSAGE);
191     reussite = 1; //permet de savoir si on peut passer à la fenêtre suivante ou non
192 }
193 return reussite;
194 }
195
196 public void consulterPraticien(int valeur, JList liste){ //méthode pour consulter les praticiens
197
198     DefaultListModel model = new DefaultListModel();
199     Iterator<Praticien> it = listpraticien.iterator(); //création d'un itérateur
200     while(it.hasNext()){
201         Praticien pn = it.next();
202         if(valeur == pn.getSpecialite()) //selon la spécialité on va afficher ou non le praticien
203             model.addElement("Dr."+pn.getName());
204     }
205     liste.setModel(model);
206 }
207
208 public void consulterPraticienVille(int valeur, JList liste, String ville){ //méthode pour consulter les praticiens
209
210     int check = 1;
211     DefaultListModel model = new DefaultListModel();
212     Iterator<Praticien> it = listpraticien.iterator(); //création d'un itérateur
213     while(it.hasNext()){
214         Praticien pn = it.next();
215         if(valeur == pn.getSpecialite()) //selon la spécialité on va afficher ou non le praticien
216             if(ville.equals(pn.getVille())){
217                 model.addElement("Dr."+pn.getName());
218                 check = 0;
219             }
220     }
221     if(check == 1){
222         JOptionPane.showMessageDialog(null,"Il n'y pas de praticien dans cette ville", "Message d'information",
JOptionPane.INFORMATION_MESSAGE);
223     }
224     liste.setModel(model);
225 }
226

```

```

227 public void lookrdv(int choix, String nom, JList liste){
228     int u=1;
229     ArrayList listb = new ArrayList();
230     int f=1;
231     int lock = 0;
232     DefaultListModel model = new DefaultListModel();
233     Iterator<Praticien> ite= listpraticien.iterator(); //on boucle sur la liste des praticien pour afficher les praticien dans un
premier temps
234     while(ite.hasNext()){
235         Praticien z = ite.next();
236         if(choix==z.getSpecialite()){
237             model.addElement(u+": Dr."+z.getName());
238         }
239         u++;
240     }
241     liste.setModel(model);
242     Iterator<Praticien> iter= listpraticien.iterator();
243     while(iter.hasNext()){
244         Praticien ze = iter.next();
245         if(nom.equals(ze.getName())==true) //si ce nom correspond bien à un nom qui appartient à celui de la liste on continue
246             lock = 1;
247         if(nom.equals("")==true)
248             lock = 1;
249     }
250     if(lock == 0){
251         JOptionPane.showMessageDialog(null, "Ce praticien n'existe pas", "Message d'avertissement",
JOptionPane.WARNING_MESSAGE);
252     }
253     DefaultListModel model1 = new DefaultListModel();
254     Iterator<Praticien> ites = listpraticien.iterator();
255     while(ites.hasNext()){
256         Praticien be = ites.next();
257         if(nom.equals(be.getName())==true){ //on accede au praticien en question
258             listb = be.getAgendaRDV();
259             if(listb.isEmpty()==true){ //si l'arraylist des rdv du praticien est vide on l'indique
260                 JOptionPane.showMessageDialog(null, "Désolé, ce praticien n'a pas ou n'a plus de RDV", "Message d'information",
JOptionPane.INFORMATION_MESSAGE);
261             }
262             else{
263                 Iterator<Integer> loel = listb.iterator();
264                 while(loel.hasNext()){
265                     model1.addElement(f+loel.next());
266                     f++;
267                 }
268                 liste.setModel(model1);
269             }
270         }
271     }
272 }
273
274 public void prendrerdv(int choix, String nom, JList liste, int o){
275     ArrayList listb = new ArrayList();
276     int t=1;
277     int u=1;
278     int y;
279     int plock = 0;
280     if(nom.length()==0){
281         JOptionPane.showMessageDialog(null, "Veuillez saisir un nom pour votre praticien", "Message d'avertissement",
JOptionPane.WARNING_MESSAGE);
282     }
283     else{
284         DefaultListModel model = new DefaultListModel();
285         Iterator<Praticien> it= listpraticien.iterator();
286         while(it.hasNext()){
287             Praticien a = it.next();
288             if(choix==a.getSpecialite()){
289                 model.addElement(u+": Dr."+a.getName());
290                 u++;
291             }
292         }
293         liste.setModel(model);
294         Iterator<Praticien> iteri= listpraticien.iterator();
295         while(iteri.hasNext()){

```

```

297     Praticien zer = iteri.next();
298     if(nom.equals(zer.getName())==true)
299         plock = 1;
300 }
301 if(plock == 0)
302     JOptionPane.showMessageDialog(null, "Ce praticien n'existe pas", "Message d'avertissement",
JOptionPane.WARNING_MESSAGE);
303     DefaultListModel model1 = new DefaultListModel();
304     Iterator<Praticien> its = listpraticien.iterator();
305     while(its.hasNext()){
306         Praticien b = its.next();
307         if(nom.equals(b.getName())==true){
308             listb = b.getAgendaRDV(); //peut etre passe au suivant a check
309             if(listb.isEmpty()==true){
310                 JOptionPane.showMessageDialog(null, "Désolé, ce praticien n'a pas ou n'a plus de RDV", "Message d'avertissement",
JOptionPane.WARNING_MESSAGE);
311             }
312             else{
313                 Iterator<Integer> lol = listb.iterator();
314                 while(lol.hasNext()){
315                     model1.addElement(t+":"+lol.next());
316                     t++;
317                 }
318                 liste.setModel(model1);
319                 if(o>(t-1)|| o<0){
320                     JOptionPane.showMessageDialog(null, "Saisissez un indice de la liste", "Message d'avertissement",
JOptionPane.WARNING_MESSAGE);
321                 }
322                 y = (int) listb.get((o-1)); //on donne à y la valeur de la date saisie
323                 listb.remove(o-1); //suppression de la date de la liste du praticien
324                 b.changertype(listb); //on met a jour la nouvelle liste sur le praticien concerné
325                 Date rdv = new Date(b.getName(),b.getNum(),nom, num, y); //on crée une date avec les données
326                 listRDV.add(rdv); //on l'ajoute à la liste des rdv
327                 JOptionPane.showMessageDialog(null, "Le rendez-vous a bien été pris pour le "+y, "Message d'information",
JOptionPane.INFORMATION_MESSAGE);
328             }
329         }
330     }
331 }
332 }
333
334 public void annulerRDV(int choix, String nom, JList liste, int s){
335     ArrayList listb = new ArrayList();
336     int v=1;
337     int block = 0;
338     int e=0;
339     DefaultListModel model = new DefaultListModel();
340     Iterator<Praticien> i= listpraticien.iterator();
341     while(i.hasNext()){
342         Praticien a = i.next();
343         if(choix==a.getSpecialite()){
344             model.addElement(v+": Dr."+a.getName());
345             v++;
346         }
347     }
348     liste.setModel(model);
349     Iterator<Date> iteratorbis = listRDV.iterator();
350     while(iteratorbis.hasNext()){
351         Date date = iteratorbis.next();
352         if(nom.equals(date.getNomDoc())==true){
353             block=1;
354         }
355     }
356     if(block == 0)
357         JOptionPane.showMessageDialog(null, "Ce praticien n'existe pas", "Message d'avertissement",
JOptionPane.WARNING_MESSAGE);
358     DefaultListModel model1 = new DefaultListModel();
359     Iterator<Date> itesi = listRDV.iterator();
360     while(itesi.hasNext()){
361         Date dat = itesi.next();
362         if(nom.equals(dat.getNomDoc())==true){
363             model1.addElement(dat.getDateRDV());
364         }
365     }

```

```

366     liste.setModel(model1);
367     Iterator<Date> iteratori = listRDV.iterator();
368     while(iteratori.hasNext()){
369         Date da = iteratori.next();
370         if(s==da.getDateRDV()){
371             e=1;
372         }
373     }
374     if(e==0)
375         JOptionPane.showMessageDialog(null, "Cette date n'existe pas", "Message d'avertissement",
JOptionPane.WARNING_MESSAGE);
376     if(e!=0){
377         Iterator<Date> iteratoris = listRDV.iterator();
378         while(iteratoris.hasNext()){
379             Date dat = iteratoris.next();
380             if(s==dat.getDateRDV()){
381                 listRDV.remove(dat);
382             }
383         }
384         Iterator<Praticien> iterat = listpraticien.iterator();
385         while(iterat.hasNext()){
386             Praticien prat = iterat.next();
387             if(nom.equals(prat.getName())==true){
388                 listb = prat.getAgendaRDV();
389             }
390             listb.add(s); //ajout de la date de la liste du praticien car elle est désormais disponible
391             prat.changertype(listb); //on met a jour la nouvelle liste sur le praticien concerné
392         }
393         JOptionPane.showMessageDialog(null, "La date du RDV a bien été supprimée", "Message d'information",
JOptionPane.WARNING_MESSAGE);
394     }
395 }
396
397 public void consulterSesRDV(JList liste){
398
399     ArrayList listb = new ArrayList();
400     DefaultListModel model = new DefaultListModel();
401     Iterator<Praticien> it = listtamponpraticien.iterator();
402     while(it.hasNext()){
403         Praticien a = it.next();
404         listb = a.getAgendaRDV();
405         if(listb.isEmpty()){
406             JOptionPane.showMessageDialog(null, "Veuillez ajouter une date, votre liste de rendez-vous est actuellement vide",
"Message d'avertissement", JOptionPane.WARNING_MESSAGE);
407         }
408         else{
409             Iterator<Integer> its = listb.iterator();
410             while(its.hasNext()){
411                 model.addElement(its.next());
412             }
413             liste.setModel(model);
414         }
415     }
416 }
417
418 public void ajouterRDV(JList liste, int d){
419
420     ArrayList listb = new ArrayList();
421     Iterator<Praticien> it = listtamponpraticien.iterator();
422     while(it.hasNext()){
423         Praticien a = it.next();
424         listb = a.getAgendaRDV();
425         if(listb.contains(d)==true){
426             JOptionPane.showMessageDialog(null, "Cette date existe deja", "Message d'averissement",
JOptionPane.WARNING_MESSAGE);
427         }
428         listb.add(d); //on ajoute la date
429         a.changertype(listb); //on met à jour la date
430         JOptionPane.showMessageDialog(null, "la date "+d+" a été ajoutée", "Message d'information",
JOptionPane.INFORMATION_MESSAGE);
431     }
432 }
433
434 public void modifierRDV(JList liste, int w, int j){

```

```

435
436 ArrayList listb = new ArrayList();
437 Iterator<Praticien> it = listtamponpraticien.iterator();
438 while(it.hasNext()){
439     Praticien a = it.next();
440     listb = a.getAgendaRDV();
441     if(listb.isEmpty()){
442         JOptionPane.showMessageDialog(null, "Veuillez ajouter une date, votre liste de rendez-vous est actuellement vide",
"Message d'averissement", JOptionPane.WARNING_MESSAGE);
443     }
444     else{
445         int u=1;
446         DefaultListModel model = new DefaultListModel();
447         Iterator<Integer> its = listb.iterator();
448         while(its.hasNext()){
449             model.addElement(u+" : "+its.next());
450             u++;
451         }
452         liste.setModel(model);
453
454         if(w>listb.size()){
455             JOptionPane.showMessageDialog(null, "Veuillez saisir un indice de la liste", "Message d'averissement",
JOptionPane.WARNING_MESSAGE);
456         }
457         else{
458             listb.set((w-1), j); //on change la date
459             a.changertype(listb); //on met à jour la liste
460             JOptionPane.showMessageDialog(null, "la nouvelle date a bien été pris en compte", "Message d'information",
JOptionPane.INFORMATION_MESSAGE);
461         }
462     }
463 }
464 }
465
466 public void annulerRDV(JList liste, int w){
467
468     ArrayList listb = new ArrayList();
469     Iterator<Praticien> it = listtamponpraticien.iterator();
470     while(it.hasNext()){
471         Praticien a = it.next();
472         listb = a.getAgendaRDV();
473         if(listb.isEmpty()){
474             JOptionPane.showMessageDialog(null, "Veuillez ajouter une date, votre liste de rendez-vous est actuellement vide",
"Message d'averissement", JOptionPane.WARNING_MESSAGE);
475         }
476         else{
477             int u=1;
478             DefaultListModel model = new DefaultListModel();
479             Iterator<Integer> its = listb.iterator();
480             while(its.hasNext()){
481                 model.addElement(u+" : "+its.next());
482                 u++;
483             }
484             liste.setModel(model);
485
486             if(w>listb.size()){
487                 JOptionPane.showMessageDialog(null, "Veuillez saisir un indice de la liste", "Message d'averissement",
JOptionPane.WARNING_MESSAGE);
488             }
489             else{
490                 listb.remove((w-1)); //on supprime la date
491                 JOptionPane.showMessageDialog(null, "La suppression a bien été effectué !", "Message d'information",
JOptionPane.INFORMATION_MESSAGE);
492                 a.changertype(listb); //on met à jour la liste
493             }
494         }
495     }
496 }
497 }
498
499 public int recupererMDPpat(String pseudo, int social){
500
501     int reussite = 0;
502     Iterator<Patient> it = listpatient.iterator(); //création d'un itérateur

```

```

503 while(it.hasNext()){
504     Patient a = it.next();
505     if(a.getPseudo().equals(pseudo)==true){
506         if(a.getSocial()==social){
507             reussite = 1;
508             listtamponpraticien.add(a); //on l'ajoute à une liste tampon qui nous permettra de savoir les détails de l'usager
connecté
509         }
510     }
511 }
512 }
513 return reussite;
514 }
515
516 public int recupererMDPprat(String pseudo, int telephone){
517     int reussite = 0;
518     Iterator<Praticien> its = listpraticien.iterator(); //création d'un itérateur
519     while(it.hasNext()){
520         Praticien a = its.next();
521         if(a.getPseudo().equals(pseudo)==true){
522             if(a.getNumtelephone()==telephone){
523                 reussite = 1;
524                 listtamponpraticien.add(a); //on l'ajoute à une liste tampon qui nous permettra de savoir les détails de l'usager
connecté
525             }
526         }
527     }
528 }
529 }
530 return reussite;
531 }
532
533 public void versFichierPrat() throws IOException{
534     FileWriter fich = new FileWriter("praticien.txt");
535
536     for(int i=0; i<listpraticien.size();i++){
537         Iterator<Praticien> iterator= listpraticien.iterator();
538         while(iterator.hasNext()){
539             Praticien a = iterator.next();
540
541         }
542         fich.write(a.getNom()+System.lineSeparator()+a.getPrenom()+System.lineSeparator()+a.getNum()+System.lineSeparator()+a.getDat
edenaissance()+System.lineSeparator()+a.getIdentifiant()+System.lineSeparator()+a.getMotdepasse()+System.lineSeparator()+a.get
Specialite()+System.lineSeparator()+a.getAgendaRDV()+System.lineSeparator()+a.getVille()+System.lineSeparator());
543     }
544     fich.close();
545 }
546
547 public void versFichierPat() throws IOException{
548     FileWriter fich = new FileWriter("patient.txt");
549
550     for(int i=0; i<listpatient.size();i++){
551         fich.write(listpatient.get(i)+System.lineSeparator());
552     }
553     fich.close();
554 }
555
556 // public void depuisFichierPrat() throws FileNotFoundException{
557 //     FileReader fich = new FileReader("praticien.txt");
558 //     BufferedReader br = new BufferedReader(fich);
559 //     String line = br.readLine();
560 //     while(line != null){
561 //         }
562 //     }
563 // }
564 // }
565
566 public int stock(){ //fonction qui permettra de savoir s'il s'agit d'un patient ou d'un praticien
567     return r;
568 }
569
570 public void versFichier() throws IOException{
571     FileWriter fich = new FileWriter("patient.txt"); //création fichier patient

```

```

572
573     for(int i=0;i<listpatient.size();i++){ //on ecrit sur le fichier
574         fich.write(listpatient.get(i)+System.lineSeparator());
575     }
576     fich.close();
577 }
578
579 public String getNom(){
580     return nom;
581 }
582
583 public String getPrenom(){
584     return prenom;
585 }
586
587 public int getNum(){
588     return num;
589 }
590
591 public int getDatedenaissance(){
592     return datedenaissance;
593 }
594
595 public String getIdentifiant(){
596     return identifiant;
597 }
598
599 public String getMotdepasse(){
600     return motdepasse;
601 }
602
603 }
604
605

```

### Partie graphique

D:\Documents\EPF\2A\Projet Java\RDV\_MEDIC\_SUTH\_Ranjit\RDV\_MEDIC\_SUTH\_Ranjit\inscription\JAVA\_FINAL - Copie - Copie\JAVA\_FINAL\RDV\_MEDICAUX\_SUTHARSAN\_RANJIT\src\pFiches\Faccueil.java

```

1  /*
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6  package pFiches;
7
8  import javax.swing.JOptionPane;
9  import rdv_medicaux_sutharsan_ranjit.Utilisateurs;
10
11  /**
12   *
13   * @author franjit
14   */
15  public class Faccueil extends javax.swing.JFrame {
16
17      /**
18       * Creates new form Faccueil
19       */
20      private Utilisateurs User= new Utilisateurs("unknown", "unknown", 0, 0,"unknown","unknown");
21      private Fconnexion fichconnexion;
22      private Finscription fichinscription;
23      private Fselection fichselection;
24
25      public Faccueil() {
26          initComponents();
27          fichconnexion = new Fconnexion(this, false, User);
28          fichinscription = new Finscription(this, false, User);
29          fichselection = new Fselection(this, false, User);
30          User.creationPraticien();
31      }
32  }

```



```

33
34 /**
35  * This method is called from within the constructor to initialize the form.
36  * WARNING: Do NOT modify this code. The content of this method is always
37  * regenerated by the Form Editor.
38  */
39 @SuppressWarnings("unchecked")
40 // <editor-fold defaultstate="collapsed" desc="Generated Code">
41 private void initComponents() {
42
43     buttonGroup1 = new javax.swing.ButtonGroup();
44     jPanel1 = new javax.swing.JPanel();
45     jPanel2 = new javax.swing.JPanel();
46     jLabel2 = new javax.swing.JLabel();
47     jLabel1 = new javax.swing.JLabel();
48     rbConnexion = new javax.swing.JRadioButton();
49     rbInscription = new javax.swing.JRadioButton();
50     rbVisiteur = new javax.swing.JRadioButton();
51     bContinuer = new javax.swing.JButton();
52     jMenuBar1 = new javax.swing.JMenuBar();
53     jMenu1 = new javax.swing.JMenu();
54     jAide = new javax.swing.JMenuItem();
55     jQuitter = new javax.swing.JMenuItem();
56
57     setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
58     setTitle("Docto rapido");
59
60     jPanel1.setBackground(new java.awt.Color(0, 0, 0));
61
62     javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
63     jPanel1.setLayout(jPanel1Layout);
64     jPanel1Layout.setHorizontalGroup(
65         jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
66             .addGap(0, 306, Short.MAX_VALUE)
67     );
68     jPanel1Layout.setVerticalGroup(
69         jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
70             .addGap(0, 0, Short.MAX_VALUE)
71     );
72
73     jPanel2.setBackground(new java.awt.Color(0, 102, 102));
74
75     jLabel2.setText("Le premier site internet vous facilitant les prises de rendez-vous chez votre medecin traitant");
76
77     jLabel1.setFont(new java.awt.Font("Tahoma", 0, 24)); // NOI18N
78     jLabel1.setText("Mon Praticien");
79
80     rbConnexion.setBackground(new java.awt.Color(0, 102, 102));
81     buttonGroup1.add(rbConnexion);
82     rbConnexion.setSelected(true);
83     rbConnexion.setText("Connexion");
84     rbConnexion.addActionListener(new java.awt.event.ActionListener() {
85         public void actionPerformed(java.awt.event.ActionEvent evt) {
86             rbConnexionActionPerformed(evt);
87         }
88     });
89
90     rbInscription.setBackground(new java.awt.Color(0, 102, 102));
91     buttonGroup1.add(rbInscription);
92     rbInscription.setText("Inscription");
93     rbInscription.addActionListener(new java.awt.event.ActionListener() {
94         public void actionPerformed(java.awt.event.ActionEvent evt) {
95             rbInscriptionActionPerformed(evt);
96         }
97     });
98
99     rbVisiteur.setBackground(new java.awt.Color(0, 102, 102));
100    buttonGroup1.add(rbVisiteur);
101    rbVisiteur.setText("Visiteur");
102
103    bContinuer.setBackground(new java.awt.Color(0, 102, 102));
104    bContinuer.setText("Continuer");
105    bContinuer.addActionListener(new java.awt.event.ActionListener() {
106        public void actionPerformed(java.awt.event.ActionEvent evt) {

```

```

107     bContinuerActionPerformed(evt);
108 }
109 });
110
111 javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
112 jPanel2.setLayout(jPanel2Layout);
113 jPanel2Layout.setHorizontalGroup(
114     jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
115         .addGroup(jPanel2Layout.createSequentialGroup()
116             .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
117                 .addGroup(jPanel2Layout.createSequentialGroup()
118                     .add(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
119                         .add(jPanel2Layout.createSequentialGroup()
120                             .addGap(47, 47, 47)
121                             .add(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
122                                 .addComponent(jLabel2)
123                                 .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 172,
javax.swing.GroupLayout.PREFERRED_SIZE)))
124                             .addGroup(jPanel2Layout.createSequentialGroup()
125                                 .addGap(56, 56, 56)
126                                 .add(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
127                                     .addComponent(rbInscription)
128                                     .addComponent(rbVisiteur)
129                                     .addComponent(rbConnexion))))
130                             .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 135, Short.MAX_VALUE))
131                             .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel2Layout.createSequentialGroup()
132                                 .addGap(0, 0, Short.MAX_VALUE)
133                                 .addComponent(bContinuer))))
134                             .addContainerGap())
135             );
136     jPanel2Layout.setVerticalGroup(
137     jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
138         .addGroup(jPanel2Layout.createSequentialGroup()
139             .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 55, javax.swing.GroupLayout.PREFERRED_SIZE)
140             .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
141             .addComponent(jLabel2)
142             .addGap(69, 69, 69)
143             .addComponent(rbConnexion)
144             .addGap(37, 37, 37)
145             .addComponent(rbInscription)
146             .addGap(45, 45, 45)
147             .addComponent(rbVisiteur)
148             .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 189, Short.MAX_VALUE)
149             .addComponent(bContinuer)
150             .addContainerGap())
151     );
152
153 jMenuBar1.setBackground(new java.awt.Color(51, 51, 51));
154
155 jMenu1.setBackground(new java.awt.Color(51, 51, 51));
156 jMenu1.setText("Help");
157
158 jAide.setText("Aide");
159 jAide.addActionListener(new java.awt.event.ActionListener() {
160     public void actionPerformed(java.awt.event.ActionEvent evt) {
161         jAideActionPerformed(evt);
162     }
163 });
164 jMenu1.add(jAide);
165
166 jQuitter.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_F4,
java.awt.event.InputEvent.ALT_MASK));
167 jQuitter.setText("Quitter");
168 jQuitter.addActionListener(new java.awt.event.ActionListener() {
169     public void actionPerformed(java.awt.event.ActionEvent evt) {
170         jQuitterActionPerformed(evt);
171     }
172 });
173 jMenu1.add(jQuitter);
174
175 jMenuBar1.add(jMenu1);
176
177 setJMenuBar(jMenuBar1);
178

```

```

179 javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
180 getContentPane().setLayout(layout);
181 layout.setHorizontalGroup(
182     layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
183     .addGroup(layout.createSequentialGroup()
184         .addContainerGap()
185         .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
186         .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
187         .addComponent(jPanel2, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
188     );
189 layout.setVerticalGroup(
190     layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
191     .addGroup(layout.createSequentialGroup()
192         .add(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
193         .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING), layout.createSequentialGroup()
194             .addGap(0, 0, Short.MAX_VALUE)
195             .addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
196         .addContainerGap())
197     );
198 } // </editor-fold>
199
200 private void jQuitterActionPerformed(java.awt.event.ActionEvent evt) {
201     this.dispose();
202 }
203
204 private void bContinuerActionPerformed(java.awt.event.ActionEvent evt) {
205     this.setVisible(false);
206     if (rbConnexion.isSelected()){
207         fichconnexion.setVisible(true);
208     }
209     if (rbInscription.isSelected()){
210         fichinscription.setVisible(true);
211     }
212     if (rbVisiteur.isSelected()){
213         fichselection.setVisible(true);
214     }
215 }
216 }
217
218 private void jAideActionPerformed(java.awt.event.ActionEvent evt) {
219     JOptionPane.showMessageDialog(null, "Veuillez cliquer sur inscrire / connexion / visiteur afin de passer à l'étape suivante
sinon veuillez quitter le site en allant dans Aide puis Quitter", "Message d'information", JOptionPane.INFORMATION_MESSAGE);
220 }
221
222 private void rbInscriptionActionPerformed(java.awt.event.ActionEvent evt) {
223     // TODO add your handling code here:
224 }
225
226 private void rbConnexionActionPerformed(java.awt.event.ActionEvent evt) {
227     // TODO add your handling code here:
228 }
229
230 /**
231  * @param args the command line arguments
232  */
233 public static void main(String args[]) {
234     /* Set the Nimbus look and feel */
235     //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
236     /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
237      * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
238      */
239     try {
240         for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
241             if ("Nimbus".equals(info.getName())) {
242                 javax.swing.UIManager.setLookAndFeel(info.getClassName());
243                 break;
244             }
245         }
246     } catch (ClassNotFoundException ex) {
247         java.util.logging.Logger.getLogger(Faccueil.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

```

```

248 } catch (InstantiationException ex) {
249     java.util.logging.Logger.getLogger(Faccueil.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
250 } catch (IllegalAccessException ex) {
251     java.util.logging.Logger.getLogger(Faccueil.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
252 } catch (javax.swing.UnsupportedLookAndFeelException ex) {
253     java.util.logging.Logger.getLogger(Faccueil.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
254 }
255 //</editor-fold>
256
257 /* Create and display the form */
258 java.awt.EventQueue.invokeLater(new Runnable() {
259     public void run() {
260         new Faccueil().setVisible(true);
261     }
262 });
263 }
264
265 // Variables declaration - do not modify
266 private javax.swing.JButton bContinuer;
267 private javax.swing.ButtonGroup buttonGroup1;
268 private javax.swing.JMenuItem jAide;
269 private javax.swing.JLabel jLabel1;
270 private javax.swing.JLabel jLabel2;
271 private javax.swing.JMenu jMenu1;
272 private javax.swing.JMenuBar jMenuBar1;
273 private javax.swing.JPanel jPanel1;
274 private javax.swing.JPanel jPanel2;
275 private javax.swing.JMenuItem jQuitter;
276 private javax.swing.JRadioButton rbConnexion;
277 private javax.swing.JRadioButton rbInscription;
278 private javax.swing.JRadioButton rbVisiteur;
279 // End of variables declaration
280 }
281

```

D:\Documents\EPF\2A\Projet Java\RDV\_MEDIC\_SUTH\_Ranjit\RDV\_MEDIC\_SUTH\_Ranjit\inscription\JAVA\_FINAL - Copie - Copie\JAVA\_FINAL\RDV\_MEDIC\_AUX\_SUTHARSAN\_RANJIT\src\pFiches\Fconnexion.java

```

1 /*
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6 package pFiches;
7
8 import java.io.IOException;
9 import java.util.logging.Level;
10 import java.util.logging.Logger;
11 import javax.swing.JOptionPane;
12 import rdv_medicaux_sutharsan_ranjit.Utilisateurs;
13
14 /**
15  *
16  * @author franjit
17  */
18 public class Fconnexion extends javax.swing.JDialog {
19
20     /**
21      * Creates new form Fconnexion
22      */
23     private Utilisateurs User ;
24     private Fselection fichselection;
25     public Fconnexion(java.awt.Frame parent, boolean modal) {
26         super(parent, modal);
27         initComponents();
28     }
29     public Fconnexion(java.awt.Frame parent, boolean modal, Utilisateurs u) {
30         super(parent, modal);
31         initComponents();
32         User= u;
33         jMDPpat.setVisible(false);
34         tfMDPpat.setVisible(false);
35         jMDPprat.setVisible(false);
36         tfMDPprat.setVisible(false);

```

```

37  lbMDPpat.setVisible(false);
38  lbMDPprat.setVisible(false);
39  }
40  /**
41  * This method is called from within the constructor to initialize the form.
42  * WARNING: Do NOT modify this code. The content of this method is always
43  * regenerated by the Form Editor.
44  */
45  @SuppressWarnings("unchecked")
46  // <editor-fold defaultstate="collapsed" desc="Generated Code">
47  private void initComponents() {
48
49      buttonGroup1 = new javax.swing.ButtonGroup();
50      jPanel1 = new javax.swing.JPanel();
51      jPanel2 = new javax.swing.JPanel();
52      jConnexion = new javax.swing.JLabel();
53      rbPatient = new javax.swing.JRadioButton();
54      rbPraticien = new javax.swing.JRadioButton();
55      jIdentifiant = new javax.swing.JLabel();
56      tfId = new javax.swing.JTextField();
57      jMotdepasse = new javax.swing.JLabel();
58      pfMotdepasse = new javax.swing.PasswordField();
59      jRetour = new javax.swing.JButton();
60      jContinuer = new javax.swing.JButton();
61      jSeparator1 = new javax.swing.JSeparator();
62      jSeparator2 = new javax.swing.JSeparator();
63      jMDPpat = new javax.swing.JButton();
64      tfMDPpat = new javax.swing.JTextField();
65      jMDPprat = new javax.swing.JButton();
66      tfMDPprat = new javax.swing.JTextField();
67      lbMDPprat = new javax.swing.JLabel();
68      lbMDPpat = new javax.swing.JLabel();
69      jMenuBar1 = new javax.swing.JMenuBar();
70      jMenu1 = new javax.swing.JMenu();
71      jMenuItem2 = new javax.swing.JMenuItem();
72      jMenuItem1 = new javax.swing.JMenuItem();
73
74      setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
75      setBackground(new java.awt.Color(102, 102, 102));
76
77      jPanel1.setBackground(new java.awt.Color(0, 0, 0));
78
79      javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
80      jPanel1.setLayout(jPanel1Layout);
81      jPanel1Layout.setHorizontalGroup(
82          jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
83              .addGroup(jPanel1Layout.createSequentialGroup()
84                  .addGap(0, 404, Short.MAX_VALUE)
85                  .addContainerGap())
86              .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
87                  .addGap(0, 0, Short.MAX_VALUE))
88              .addGap(0, 0, Short.MAX_VALUE));
89
90      jPanel2.setBackground(new java.awt.Color(0, 102, 102));
91
92      jConnexion.setFont(new java.awt.Font("Tahoma", 0, 24)); // NOI18N
93      jConnexion.setText("Connexion");
94
95      rbPatient.setBackground(new java.awt.Color(0, 102, 102));
96      buttonGroup1.add(rbPatient);
97      rbPatient.setText("Patient");
98      rbPatient.addActionListener(new java.awt.event.ActionListener() {
99          public void actionPerformed(java.awt.event.ActionEvent evt) {
100              rbPatientActionPerformed(evt);
101          }
102      });
103
104      rbPraticien.setBackground(new java.awt.Color(0, 102, 102));
105      buttonGroup1.add(rbPraticien);
106      rbPraticien.setText("Praticien");
107      rbPraticien.addActionListener(new java.awt.event.ActionListener() {
108          public void actionPerformed(java.awt.event.ActionEvent evt) {
109              rbPraticienActionPerformed(evt);
110          }
111      });

```

```

111     });
112
113     jIdentifiant.setText("Identifiant");
114
115     tfId.setBackground(new java.awt.Color(0, 102, 102));
116     tfId.setBorder(null);
117     tfId.addActionListener(new java.awt.event.ActionListener() {
118         public void actionPerformed(java.awt.event.ActionEvent evt) {
119             tfIdActionPerformed(evt);
120         }
121     });
122
123     jMotdepasse.setText("Mot de passe");
124
125     pfMotdepasse.setBackground(new java.awt.Color(0, 102, 102));
126     pfMotdepasse.setBorder(null);
127     pfMotdepasse.addActionListener(new java.awt.event.ActionListener() {
128         public void actionPerformed(java.awt.event.ActionEvent evt) {
129             pfMotdepasseActionPerformed(evt);
130         }
131     });
132
133     jRetour.setBackground(new java.awt.Color(0, 102, 102));
134     jRetour.setText("Retour");
135     jRetour.addMouseListener(new java.awt.event.MouseMotionAdapter() {
136         public void mouseDragged(java.awt.event.MouseEvent evt) {
137             jRetourMouseDragged(evt);
138         }
139     });
140     jRetour.addActionListener(new java.awt.event.ActionListener() {
141         public void actionPerformed(java.awt.event.ActionEvent evt) {
142             jRetourActionPerformed(evt);
143         }
144     });
145
146     jContinuer.setBackground(new java.awt.Color(0, 102, 102));
147     jContinuer.setText("Continuer");
148     jContinuer.addActionListener(new java.awt.event.ActionListener() {
149         public void actionPerformed(java.awt.event.ActionEvent evt) {
150             jContinuerActionPerformed(evt);
151         }
152     });
153
154     jMDPpat.setBackground(new java.awt.Color(0, 102, 102));
155     jMDPpat.setText("Mot de passe oublié ?");
156     jMDPpat.addActionListener(new java.awt.event.ActionListener() {
157         public void actionPerformed(java.awt.event.ActionEvent evt) {
158             jMDPpatActionPerformed(evt);
159         }
160     });
161
162     jMDPprat.setBackground(new java.awt.Color(0, 102, 102));
163     jMDPprat.setText("Mot de passe oublié ?");
164     jMDPprat.addActionListener(new java.awt.event.ActionListener() {
165         public void actionPerformed(java.awt.event.ActionEvent evt) {
166             jMDPpratActionPerformed(evt);
167         }
168     });
169
170     lbMDPprat.setText("Saisissez le numéro de téléphone associé à votre compte et l'identifiant de votre compte");
171
172     lbMDPpat.setText("Saisissez le numéro social associé à votre compte et l'identifiant de votre compte");
173
174     javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
175     jPanel2.setLayout(jPanel2Layout);
176     jPanel2Layout.setHorizontalGroup(
177         jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
178             .addGroup(jPanel2Layout.createSequentialGroup()
179                 .add(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
180                     .add(jRetour)
181                     .add(jContinuer)
182                 )
183                 .addContainerGap())

```



```

184 .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel2Layout.createSequentialGroup())
185 .addContainerGap(36, Short.MAX_VALUE)
186 .addComponent(lbMDPprat)
187 .addGap(21, 21, 21))
188 .addGroup(jPanel2Layout.createSequentialGroup())
189 .addGap(50, 50, 50)
190 .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
191 .addComponent(lbMDPpat)
192 .addComponent(jConnexion)
193 .addGroup(jPanel2Layout.createSequentialGroup())
194 .addComponent(jMDPpat)
195 .addGap(18, 18, 18)
196 .addComponent(tfMDPpat, javax.swing.GroupLayout.PREFERRED_SIZE, 116,
javax.swing.GroupLayout.PREFERRED_SIZE))
197 .addGroup(jPanel2Layout.createSequentialGroup())
198 .addComponent(jMDPprat)
199 .addGap(18, 18, 18)
200 .addComponent(tfMDPprat, javax.swing.GroupLayout.PREFERRED_SIZE, 116,
javax.swing.GroupLayout.PREFERRED_SIZE))
201 .addGroup(jPanel2Layout.createSequentialGroup())
202 .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
203 .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel2Layout.createSequentialGroup())
204 .addComponent(rbPatient)
205 .addGap(18, 18, 18))
206 .addGroup(jPanel2Layout.createSequentialGroup())
207 .addComponent(jIdentifiant)
208 .addGap(28, 28, 28)))
209 .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)
210 .addComponent(tfId, javax.swing.GroupLayout.DEFAULT_SIZE, 129, Short.MAX_VALUE)
211 .addComponent(rbPraticien)
212 .addComponent(jSeparator1)))
213 .addGroup(jPanel2Layout.createSequentialGroup())
214 .addComponent(jMotdepasse)
215 .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
216 .addGroup(jPanel2Layout.createSequentialGroup())
217 .addGap(15, 15, 15)
218 .addComponent(jSeparator2, javax.swing.GroupLayout.PREFERRED_SIZE, 123,
javax.swing.GroupLayout.PREFERRED_SIZE))
219 .addGroup(jPanel2Layout.createSequentialGroup())
220 .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
221 .addComponent(pfMotdepasse, javax.swing.GroupLayout.PREFERRED_SIZE, 122,
javax.swing.GroupLayout.PREFERRED_SIZE))))))
222 .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
223 );
224 jPanel2Layout.setVerticalGroup(
225 jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
226 .addGroup(jPanel2Layout.createSequentialGroup())
227 .addGap(55, 55, 55)
228 .addComponent(jConnexion)
229 .addGap(42, 42, 42)
230 .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
231 .addComponent(rbPatient)
232 .addComponent(rbPraticien))
233 .addGap(62, 62, 62)
234 .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
235 .addComponent(jIdentifiant)
236 .addComponent(tfId, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
237 .addGap(4, 4, 4)
238 .addComponent(jSeparator1, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
239 .addGap(32, 32, 32)
240 .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
241 .addComponent(jMotdepasse)
242 .addComponent(pfMotdepasse, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
243 .addComponent(jSeparator2, javax.swing.GroupLayout.PREFERRED_SIZE, 10,
javax.swing.GroupLayout.PREFERRED_SIZE)
244 .addGap(49, 49, 49)
245 .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
246 .addComponent(jMDPpat)
247 .addComponent(tfMDPpat, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
248 .addGap(18, 18, 18)

```

```

249     .addComponent(lbMDPpat)
250     .addGap(18, 18, 18)
251     .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
252         .addComponent(jMDPprat)
253         .addComponent(tfMDPprat, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
254     .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 17, Short.MAX_VALUE)
255     .addComponent(lbMDPprat)
256     .addGap(18, 18, 18)
257     .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
258         .addComponent(jRetour)
259         .addComponent(jContinuer))
260     .addContainerGap()
261 );
262
263 jMenuBar1.setBackground(new java.awt.Color(51, 51, 51));
264 jMenuBar1.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 0, 0)));
265
266 jMenu1.setBackground(new java.awt.Color(51, 51, 51));
267 jMenu1.setText("Help");
268
269 jMenuItem2.setText("Aide");
270 jMenuItem2.addActionListener(new java.awt.event.ActionListener() {
271     public void actionPerformed(java.awt.event.ActionEvent evt) {
272         jMenuItem2ActionPerformed(evt);
273     }
274 });
275 jMenu1.add(jMenuItem2);
276
277 jMenuItem1.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_F4,
java.awt.event.InputEvent.ALT_MASK));
278 jMenuItem1.setText("Quitter");
279 jMenuItem1.addActionListener(new java.awt.event.ActionListener() {
280     public void actionPerformed(java.awt.event.ActionEvent evt) {
281         jMenuItem1ActionPerformed(evt);
282     }
283 });
284 jMenu1.add(jMenuItem1);
285
286 jMenuBar1.add(jMenu1);
287
288 setJMenuBar(jMenuBar1);
289
290 javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
291 getContentPane().setLayout(layout);
292 layout.setHorizontalGroup(
293     layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
294     .addGroup(layout.createSequentialGroup()
295         .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
296         .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
297         .addComponent(jPanel2, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
298 );
299 layout.setVerticalGroup(
300     layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
301     .addGroup(layout.createSequentialGroup()
302         .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
303         .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
304         .addComponent(jPanel2, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
305 );
306 pack();
307 // </editor-fold>
308 private void jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt) {
309     this.dispose();
310 }
311
312 private void jMenuItem2ActionPerformed(java.awt.event.ActionEvent evt) {
313     JOptionPane.showMessageDialog(null, "Veuillez sélectionner patient ou praticien puis entrer votre identifiant et mot de
passe ou bien retourner à la page précédente", "Message d'information", JOptionPane.INFORMATION_MESSAGE);
314 }
315

```



```

316 private void jContinuerActionPerformed(java.awt.event.ActionEvent evt) {
317     int indicPraticien = 2;
318     int reussite = 2;
319
320
321     if (rbPraticien.isSelected()){
322         indicPraticien=0;
323         reussite = User.connexion(indicPraticien, tfld.getText(), pfMotdepasse.getText());
324         if(reussite == 1){
325             fichselection = new Fselection(this, false, indicPraticien,User);
326             this.setVisible(false);
327             fichselection.setVisible(true);
328         }
329     }
330     else if (rbPatient.isSelected()){
331         indicPraticien=1;
332         reussite = User.connexion(indicPraticien, tfld.getText(), pfMotdepasse.getText());
333         if(reussite == 1){
334             fichselection = new Fselection(this, false, indicPraticien,User);
335             this.setVisible(false);
336             fichselection.setVisible(true);
337         }
338     }
339     else
340     JOptionPane.showMessageDialog(null, "Veuillez selectionner patient ou praticien !");
341     // if(User.connexion(indicPraticien,tfld.getText(),pfMotdepasse.getText())==1){
342     //     this.setVisible(false);
343     //     fichselection.setVisible(true);
344
345     // }
346
347 }
348
349 private void jRetourActionPerformed(java.awt.event.ActionEvent evt) {
350     this.setVisible(false);
351     this.getParent().setVisible(true);
352 }
353
354 private void jRetourMouseDragged(java.awt.event.MouseEvent evt) {
355     // TODO add your handling code here:
356 }
357
358 private void pfMotdepasseActionPerformed(java.awt.event.ActionEvent evt) {
359     // TODO add your handling code here:
360 }
361
362 private void tfldActionPerformed(java.awt.event.ActionEvent evt) {
363     // TODO add your handling code here:
364 }
365
366 private void rbPraticienActionPerformed(java.awt.event.ActionEvent evt) {
367     jMDPpat.setVisible(false);
368     tfMDPpat.setVisible(false);
369     lbMDPpat.setVisible(false);
370     jMDPprat.setVisible(true);
371     tfMDPprat.setVisible(true);
372     lbMDPprat.setVisible(true);
373 }
374
375 private void rbPatientActionPerformed(java.awt.event.ActionEvent evt) {
376
377     jMDPpat.setVisible(true);
378     tfMDPpat.setVisible(true);
379     lbMDPpat.setVisible(true);
380     jMDPprat.setVisible(false);
381     tfMDPprat.setVisible(false);
382     lbMDPprat.setVisible(false);
383
384 }
385
386 private void jMDPpatActionPerformed(java.awt.event.ActionEvent evt) {
387
388     String social = tfMDPpat.getText();

```

```

389 //JOptionPane.showMessageDialog(null, "Saisissez le numéro social associé à votre compte et l'identifiant de votre compte",
"Message d'information", JOptionPane.INFORMATION_MESSAGE);
390
391 if(tfld.getText().length()==0 || social.length()==0){
392     JOptionPane.showMessageDialog(null, "Saisissez l'identifiant et le numero de telephone", "Message d'avertissement",
JOptionPane.ERROR_MESSAGE);
393 }
394 if(tfld.getText().length()!=0 && social.length()!=0){
395     if(User.recupererMDPpat(tfld.getText(), Integer.parseInt(social))==1){
396
397         JOptionPane.showMessageDialog(null, "Vous avez reussi vous connecter", "Message d'information",
JOptionPane.INFORMATION_MESSAGE);
398         fichselection = new Fselection(this, false, 1, User);
399         this.setVisible(false);
400         fichselection.setVisible(true);
401     }
402     else{
403         JOptionPane.showMessageDialog(null, "Vous n'avez pas reussi vous connecter", "Message d'erreur",
JOptionPane.ERROR_MESSAGE);
404     }
405 }
406 }
407
408 private void jMDPpratActionPerformed(java.awt.event.ActionEvent evt) {
409
410     String telephone = tfMDPprat.getText();
411
412     if(tfld.getText().length()==0 || telephone.length()==0){
413         JOptionPane.showMessageDialog(null, "Saisissez l'identifiant et le numero de telephone", "Message d'avertissement",
JOptionPane.ERROR_MESSAGE);
414     }
415     if(tfld.getText().length()!=0 && telephone.length()!=0){
416         if(User.recupererMDPprat(tfld.getText(), Integer.parseInt(telephone))==1){
417
418             JOptionPane.showMessageDialog(null, "Vous avez reussi vous connecter", "Message d'information",
JOptionPane.INFORMATION_MESSAGE);
419             fichselection = new Fselection(this, false, 0, User);
420             this.setVisible(false);
421             fichselection.setVisible(true);
422         }
423     }
424     else{
425         JOptionPane.showMessageDialog(null, "Vous n'avez pas reussi vous connecter", "Message d'erreur",
JOptionPane.ERROR_MESSAGE);
426     }
427 }
428
429 /**
430  * @param args the command line arguments
431  */
432 public static void main(String args[]) {
433     /* Set the Nimbus look and feel */
434     //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
435     /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
436     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
437     */
438     try {
439         for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
440             if ("Nimbus".equals(info.getName())) {
441                 javax.swing.UIManager.setLookAndFeel(info.getClassName());
442                 break;
443             }
444         }
445     } catch (ClassNotFoundException ex) {
446         java.util.logging.Logger.getLogger(Fconnexion.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
447     } catch (InstantiationException ex) {
448         java.util.logging.Logger.getLogger(Fconnexion.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
449     } catch (IllegalAccessException ex) {
450         java.util.logging.Logger.getLogger(Fconnexion.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
451     } catch (javax.swing.UnsupportedLookAndFeelException ex) {
452         java.util.logging.Logger.getLogger(Fconnexion.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
453     }
454     //</editor-fold>
455 }

```

```

456  /* Create and display the dialog */
457  java.awt.EventQueue.invokeLater(new Runnable() {
458      public void run() {
459          Fconnexion dialog = new Fconnexion(new javax.swing.JFrame(), true);
460          dialog.addWindowListener(new java.awt.event.WindowAdapter() {
461              @Override
462              public void windowClosing(java.awt.event.WindowEvent e) {
463                  System.exit(0);
464              }
465          });
466          dialog.setVisible(true);
467      }
468  });
469  }
470
471  // Variables declaration - do not modify
472  private javax.swing.ButtonGroup buttonGroup1;
473  private javax.swing.JLabel jConnexion;
474  private javax.swing.JButton jContinuer;
475  private javax.swing.JLabel jIdentifiant;
476  private javax.swing.JButton jMDPpat;
477  private javax.swing.JButton jMDPprat;
478  private javax.swing.JMenu jMenu1;
479  private javax.swing.JMenuBar jMenuBar1;
480  private javax.swing.JMenuItem jMenuItem1;
481  private javax.swing.JMenuItem jMenuItem2;
482  private javax.swing.JLabel jMotdepasse;
483  private javax.swing.JPanel jPanel1;
484  private javax.swing.JPanel jPanel2;
485  private javax.swing.JButton jRetour;
486  private javax.swing.JSeparator jSeparator1;
487  private javax.swing.JSeparator jSeparator2;
488  private javax.swing.JLabel lbMDPpat;
489  private javax.swing.JLabel lbMDPprat;
490  private javax.swing.JPasswordField pfMotdepasse;
491  private javax.swing.JRadioButton rbPatient;
492  private javax.swing.JRadioButton rbPraticien;
493  private javax.swing.JTextField tfId;
494  private javax.swing.JTextField tfMDPpat;
495  private javax.swing.JTextField tfMDPprat;
496  // End of variables declaration
497  }
498

```

D:\Documents\EPF\2A\Projet Java\RDV\_MEDIC\_SUTH\_Ranjit\RDV\_MEDIC\_SUTH\_Ranjit\inscription\JAVA\_FINAL - Copie - Copie\JAVA\_FINAL\RDV\_MEDICAUX\_SUTHARSAN\_RANJIT\src\pFiches\FconsulterRdv.java

```

1  /*
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6  package pFiches;
7
8  import rdv_medicaux_sutharsan_ranjit.Utilisateurs;
9
10 /**
11  *
12  * @author Vithusan
13  */
14 public class FconsulterRdv extends javax.swing.JDialog {
15
16     /**
17     * Creates new form FconsulterRdv
18     */
19     Utilisateurs User;
20
21     public FconsulterRdv(java.awt.Frame parent, boolean modal, Utilisateurs u) {
22         super(parent, modal);
23         initComponents();
24         User = u;
25     }
26
27     public FconsulterRdv(java.awt.Frame parent, boolean modal) {

```

```

28     super(parent, modal);
29     initComponents();
30 }
31
32 /**
33  * This method is called from within the constructor to initialize the form.
34  * WARNING: Do NOT modify this code. The content of this method is always
35  * regenerated by the Form Editor.
36  */
37 @SuppressWarnings("unchecked")
38 // <editor-fold defaultstate="collapsed" desc="Generated Code">
39 private void initComponents() {
40
41     jPanel1 = new javax.swing.JPanel();
42     jContinuer = new javax.swing.JButton();
43     jRetour = new javax.swing.JButton();
44     rbGénéraliste = new javax.swing.JRadioButton();
45     rbSpécialité = new javax.swing.JRadioButton();
46     jMenuBar1 = new javax.swing.JMenuBar();
47     jMenu1 = new javax.swing.JMenu();
48     jMenuItem1 = new javax.swing.JMenuItem();
49     jMenuItem2 = new javax.swing.JMenuItem();
50
51     setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
52
53     jPanel1.setBackground(new java.awt.Color(0, 102, 102));
54
55     jContinuer.setBackground(new java.awt.Color(0, 102, 102));
56     jContinuer.setText("Continuer");
57     jContinuer.addActionListener(new java.awt.event.ActionListener() {
58         public void actionPerformed(java.awt.event.ActionEvent evt) {
59             jContinuerActionPerformed(evt);
60         }
61     });
62
63     jRetour.setBackground(new java.awt.Color(0, 102, 102));
64     jRetour.setText("Retour");
65     jRetour.addActionListener(new java.awt.event.ActionListener() {
66         public void actionPerformed(java.awt.event.ActionEvent evt) {
67             jRetourActionPerformed(evt);
68         }
69     });
70
71     rbGénéraliste.setBackground(new java.awt.Color(0, 102, 102));
72     rbGénéraliste.setText("Généraliste");
73
74     rbSpécialité.setBackground(new java.awt.Color(0, 102, 102));
75     rbSpécialité.setText("Spécialiste");
76
77     javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
78     jPanel1.setLayout(jPanel1Layout);
79     jPanel1Layout.setHorizontalGroup(
80         jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
81             .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup()
82                 .addContainerGap()
83                 .addComponent(jRetour)
84                 .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
85                 .addComponent(jContinuer)
86                 .addContainerGap())
87             .addGroup(jPanel1Layout.createSequentialGroup()
88                 .addGap(77, 77, 77)
89                 .addComponent(rbGénéraliste)
90                 .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
91                 .addComponent(rbSpécialité)
92                 .addGap(76, 76, 76))
93     );
94     jPanel1Layout.setVerticalGroup(
95         jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
96             .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup()
97                 .addGap(98, 98, 98)
98                 .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
99                     .addComponent(rbGénéraliste)
100                     .addComponent(rbSpécialité))

```

```

101     .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 124, Short.MAX_VALUE)
102     .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
103         .addComponent(jContinuer)
104         .addComponent(jRetour))
105     .addContainerGap()
106 };
107
108 jMenuBar1.setBackground(new java.awt.Color(51, 51, 51));
109
110 jMenu1.setBackground(new java.awt.Color(51, 51, 51));
111 jMenu1.setText("Help");
112
113 jMenuItem1.setText("Aide");
114 jMenuItem1.addActionListener(new java.awt.event.ActionListener() {
115     public void actionPerformed(java.awt.event.ActionEvent evt) {
116         jMenuItem1ActionPerformed(evt);
117     }
118 });
119 jMenu1.add(jMenuItem1);
120
121 jMenuItem2.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_F4,
java.awt.event.InputEvent.ALT_MASK));
122 jMenuItem2.setText("Quitter");
123 jMenuItem2.addActionListener(new java.awt.event.ActionListener() {
124     public void actionPerformed(java.awt.event.ActionEvent evt) {
125         jMenuItem2ActionPerformed(evt);
126     }
127 });
128 jMenu1.add(jMenuItem2);
129
130 jMenuBar1.add(jMenu1);
131
132 setJMenuBar(jMenuBar1);
133
134 javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
135 getContentPane().setLayout(layout);
136 layout.setHorizontalGroup(
137     layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
138     .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
139 );
140 layout.setVerticalGroup(
141     layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
142     .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
143 );
144
145 pack();
146 } // </editor-fold>
147
148 private void jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt) {
149     // TODO add your handling code here:
150 }
151
152 private void jMenuItem2ActionPerformed(java.awt.event.ActionEvent evt) {
153     this.dispose();
154 }
155
156 private void jContinuerActionPerformed(java.awt.event.ActionEvent evt) {
157 }
158
159
160 private void jRetourActionPerformed(java.awt.event.ActionEvent evt) {
161 }
162 }
163
164 /**
165  * @param args the command line arguments
166  */
167 public static void main(String args[]) {
168     /* Set the Nimbus look and feel */
169     //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
170     /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
171     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

```

```

172  */
173  try {
174      for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
175          if ("Nimbus".equals(info.getName())) {
176              javax.swing.UIManager.setLookAndFeel(info.getClassName());
177              break;
178          }
179      }
180  } catch (ClassNotFoundException ex) {
181      java.util.logging.Logger.getLogger(FconsulterRdv.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
182  } catch (InstantiationException ex) {
183      java.util.logging.Logger.getLogger(FconsulterRdv.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
184  } catch (IllegalAccessException ex) {
185      java.util.logging.Logger.getLogger(FconsulterRdv.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
186  } catch (javax.swing.UnsupportedLookAndFeelException ex) {
187      java.util.logging.Logger.getLogger(FconsulterRdv.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
188  }
189  //</editor-fold>
190
191  /* Create and display the dialog */
192  java.awt.EventQueue.invokeLater(new Runnable() {
193      public void run() {
194          FconsulterRdv dialog = new FconsulterRdv(new javax.swing.JFrame(), true);
195          dialog.addWindowListener(new java.awt.event.WindowAdapter() {
196              @Override
197              public void windowClosing(java.awt.event.WindowEvent e) {
198                  System.exit(0);
199              }
200          });
201          dialog.setVisible(true);
202      }
203  });
204  }
205
206  // Variables declaration - do not modify
207  private javax.swing.JButton jContinuer;
208  private javax.swing.JMenu jMenu1;
209  private javax.swing.JMenuBar jMenuBar1;
210  private javax.swing.JMenuItem jMenuItem1;
211  private javax.swing.JMenuItem jMenuItem2;
212  private javax.swing.JPanel jPanel1;
213  private javax.swing.JButton jRetour;
214  private javax.swing.JRadioButton rbGénéraliste;
215  private javax.swing.JRadioButton rbSpécialité;
216  // End of variables declaration
217  }
218

```

D:\Documents\EPF\2A\Projet Java\RDV\_MEDIC\_SUTH\_Ranjit\RDV\_MEDIC\_SUTH\_Ranjit\inscription\JAVA\_FINAL - Copie - Copie\JAVA\_FINAL\RDV\_MEDICAUX\_SUTHARSAN\_RANJIT\src\pFiches\Finscription.java

```

1  /*
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6  package pFiches;
7
8  import javax.swing.JOptionPane;
9  import rdv_medicaux_sutharsan_ranjit.Utilisateurs;
10
11  /**
12   *
13   * @author franjit
14   */
15  public class Finscription extends javax.swing.JDialog {
16
17      /**
18       * Creates new form Finscription
19       */
20      private Utilisateurs User;
21      private Fselection fichselection;
22      public Finscription(java.awt.Frame parent, boolean modal) {
23          super(parent, modal);

```

```

24     initComponents();
25 }
26 public Finscription(java.awt.Frame parent, boolean modal, Utilisateurs u) {
27     super(parent, modal);
28     initComponents();
29     User = u;
30     lbErreur.setVisible(false);
31 }
32
33 /**
34  * This method is called from within the constructor to initialize the form.
35  * WARNING: Do NOT modify this code. The content of this method is always
36  * regenerated by the Form Editor.
37  */
38 @SuppressWarnings("unchecked")
39 // <editor-fold defaultstate="collapsed" desc="Generated Code">
40 private void initComponents() {
41
42     buttonGroup1 = new javax.swing.ButtonGroup();
43     scrollPane1 = new java.awt.ScrollPane();
44     jPanel2 = new javax.swing.JPanel();
45     jPanel1 = new javax.swing.JPanel();
46     bRetour = new javax.swing.JButton();
47     bContinuer = new javax.swing.JButton();
48     lbNom = new javax.swing.JLabel();
49     tfNom = new javax.swing.JTextField();
50     lbPrenom = new javax.swing.JLabel();
51     tfPrenom = new javax.swing.JTextField();
52     lbDatedenaissance = new javax.swing.JLabel();
53     tfNumtel = new javax.swing.JTextField();
54     lbNumtel = new javax.swing.JLabel();
55     tfDatedenaissance = new javax.swing.JTextField();
56     jLabel1 = new javax.swing.JLabel();
57     lbInscription = new javax.swing.JLabel();
58     rbPraticien = new javax.swing.JRadioButton();
59     rbPatient = new javax.swing.JRadioButton();
60     lbPseudo = new javax.swing.JLabel();
61     tfPseudo = new javax.swing.JTextField();
62     pfMotdepasse = new javax.swing.JPasswordField();
63     lbMotdepasse = new javax.swing.JLabel();
64     tfNumsocial = new javax.swing.JTextField();
65     tfSpecialite = new javax.swing.JTextField();
66     lbSpecialite = new javax.swing.JLabel();
67     lbNumsocial = new javax.swing.JLabel();
68     jSeparator1 = new javax.swing.JSeparator();
69     jSeparator2 = new javax.swing.JSeparator();
70     jSeparator3 = new javax.swing.JSeparator();
71     jSeparator4 = new javax.swing.JSeparator();
72     jSeparator5 = new javax.swing.JSeparator();
73     jSeparator6 = new javax.swing.JSeparator();
74     jSeparator7 = new javax.swing.JSeparator();
75     jSeparator8 = new javax.swing.JSeparator();
76     lbVille = new javax.swing.JLabel();
77     tfVille = new javax.swing.JTextField();
78     jSeparator9 = new javax.swing.JSeparator();
79     lbErreur = new javax.swing.JLabel();
80     jMenuBar1 = new javax.swing.JMenuBar();
81     jMenu1 = new javax.swing.JMenu();
82     jMenuItem1 = new javax.swing.JMenuItem();
83     jMenuItem2 = new javax.swing.JMenuItem();
84
85     setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
86
87     jPanel2.setBackground(new java.awt.Color(0, 0, 0));
88
89     javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
90     jPanel2.setLayout(jPanel2Layout);
91     jPanel2Layout.setHorizontalGroup(
92         jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
93             .addGroup(jPanel2Layout.createSequentialGroup()
94                 .addGap(0, 437, Short.MAX_VALUE)
95                 .addContainerGap())
96             .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
97                 .addGap(0, 486, Short.MAX_VALUE))

```



```

98 );
99
100 jPanel1.setBackground(new java.awt.Color(0, 102, 102));
101
102 bRetour.setBackground(new java.awt.Color(0, 102, 102));
103 bRetour.setText("Retour");
104 bRetour.addActionListener(new java.awt.event.ActionListener() {
105     public void actionPerformed(java.awt.event.ActionEvent evt) {
106         bRetourActionPerformed(evt);
107     }
108 });
109
110 bContinuer.setBackground(new java.awt.Color(0, 102, 102));
111 bContinuer.setText("Continuer");
112 bContinuer.addActionListener(new java.awt.event.ActionListener() {
113     public void actionPerformed(java.awt.event.ActionEvent evt) {
114         bContinuerActionPerformed(evt);
115     }
116 });
117
118 lbNom.setText("Nom");
119
120 tfNom.setBackground(new java.awt.Color(0, 102, 102));
121 tfNom.setBorder(null);
122 tfNom.addActionListener(new java.awt.event.ActionListener() {
123     public void actionPerformed(java.awt.event.ActionEvent evt) {
124         tfNomActionPerformed(evt);
125     }
126 });
127
128 lbPrenom.setText("Prénom");
129
130 tfPrenom.setBackground(new java.awt.Color(0, 102, 102));
131 tfPrenom.setBorder(null);
132 tfPrenom.addActionListener(new java.awt.event.ActionListener() {
133     public void actionPerformed(java.awt.event.ActionEvent evt) {
134         tfPrenomActionPerformed(evt);
135     }
136 });
137
138 lbDatedenaissance.setText("Date de naissance");
139
140 tfNumtel.setBackground(new java.awt.Color(0, 102, 102));
141 tfNumtel.setBorder(null);
142
143 lbNumtel.setText("Num tel ");
144
145 tfDatedenaissance.setBackground(new java.awt.Color(0, 102, 102));
146 tfDatedenaissance.setBorder(null);
147
148 jLabel1.setFont(new java.awt.Font("Tahoma", 0, 10)); // NOI18N
149 jLabel1.setText("Ecrire sous la forme JJ/MM/AAAA");
150
151 lbInscription.setFont(new java.awt.Font("Tahoma", 0, 24)); // NOI18N
152 lbInscription.setText("Inscription");
153
154 rbPraticien.setBackground(new java.awt.Color(0, 102, 102));
155 buttonGroup1.add(rbPraticien);
156 rbPraticien.setSelected(true);
157 rbPraticien.setText("Praticien");
158 rbPraticien.addActionListener(new java.awt.event.ActionListener() {
159     public void actionPerformed(java.awt.event.ActionEvent evt) {
160         rbPraticienActionPerformed(evt);
161     }
162 });
163
164 rbPatient.setBackground(new java.awt.Color(0, 102, 102));
165 buttonGroup1.add(rbPatient);
166 rbPatient.setText("Patient");
167 rbPatient.addActionListener(new java.awt.event.ActionListener() {
168     public void actionPerformed(java.awt.event.ActionEvent evt) {
169         rbPatientActionPerformed(evt);
170     }
171 });

```



```

172
173 lbPseudo.setText("Pseudo");
174
175 tfPseudo.setBackground(new java.awt.Color(0, 102, 102));
176 tfPseudo.setBorder(null);
177 tfPseudo.addActionListener(new java.awt.event.ActionListener() {
178     public void actionPerformed(java.awt.event.ActionEvent evt) {
179         tfPseudoActionPerformed(evt);
180     }
181 });
182
183 pfMotdepasse.setBackground(new java.awt.Color(0, 102, 102));
184 pfMotdepasse.setBorder(null);
185
186 lbMotdepasse.setText("Mot de passe");
187
188 tfNumsocial.setBackground(new java.awt.Color(0, 102, 102));
189 tfNumsocial.setBorder(null);
190 tfNumsocial.addActionListener(new java.awt.event.ActionListener() {
191     public void actionPerformed(java.awt.event.ActionEvent evt) {
192         tfNumsocialActionPerformed(evt);
193     }
194 });
195
196 tfSpecialite.setBackground(new java.awt.Color(0, 102, 102));
197 tfSpecialite.setBorder(null);
198 tfSpecialite.addActionListener(new java.awt.event.ActionListener() {
199     public void actionPerformed(java.awt.event.ActionEvent evt) {
200         tfSpecialiteActionPerformed(evt);
201     }
202 });
203
204 lbSpecialite.setText("Spécialite");
205
206 lbNumsocial.setText("Num social");
207
208 lbVille.setText("Ville");
209
210 tfVille.setBackground(new java.awt.Color(0, 102, 102));
211
212 lbErreur.setBackground(new java.awt.Color(255, 0, 0));
213 lbErreur.setText("REMPLISSEZ TOUS LES CHAMPS !");
214
215 javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
216 jPanel1.setLayout(jPanel1Layout);
217 jPanel1Layout.setHorizontalGroup(
218     jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
219         .addGroup(jPanel1Layout.createSequentialGroup()
220             .addContainerGap()
221             .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
222                 .addGroup(jPanel1Layout.createSequentialGroup()
223                     .addComponent(rbPatient)
224                     .addGroup(jPanel1Layout.createSequentialGroup()
225                         .addComponent(bRetour)
226                         .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 256, Short.MAX_VALUE)))
227                     .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
228                         .addGroup(jPanel1Layout.createSequentialGroup()
229                             .addGap(41, 41, 41)
230                             .addComponent(bContinuer))
231                         .addGroup(jPanel1Layout.createSequentialGroup()
232                             .addGap(18, 18, 18)
233                             .addComponent(rbPraticien)))
234                     .addContainerGap())
235                 .addGroup(jPanel1Layout.createSequentialGroup()
236                     .addComponent(jSeparator8, javax.swing.GroupLayout.PREFERRED_SIZE, 30,
237                         javax.swing.GroupLayout.PREFERRED_SIZE)
238                     .addGroup(jPanel1Layout.createSequentialGroup()
239                         .addComponent(lbSpecialite)
240                         .addGap(18, 18, 18)
241                         .addComponent(tfSpecialite, javax.swing.GroupLayout.PREFERRED_SIZE, 30,
242                             javax.swing.GroupLayout.PREFERRED_SIZE)))
243                     .addGap(0, 0, Short.MAX_VALUE)))

```

```

244 .addGroup(jPanel1Layout.createSequentialGroup())
245 .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
246 .addGroup(jPanel1Layout.createSequentialGroup()
247 .addGap(169, 169, 169)
248 .addComponent(lblInscription))
249 .addGroup(jPanel1Layout.createSequentialGroup()
250 .addContainerGap()
251 .addComponent(lblNom)
252 .addGap(18, 18, 18)
253 .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
254 .addGroup(jPanel1Layout.createSequentialGroup()
255 .addComponent(jSeparator2, javax.swing.GroupLayout.PREFERRED_SIZE, 99,
javax.swing.GroupLayout.PREFERRED_SIZE)
256 .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
257 .addComponent(jSeparator1, javax.swing.GroupLayout.PREFERRED_SIZE, 100,
javax.swing.GroupLayout.PREFERRED_SIZE))
258 .addGroup(jPanel1Layout.createSequentialGroup()
259 .addComponent(tfNom, javax.swing.GroupLayout.PREFERRED_SIZE, 100,
javax.swing.GroupLayout.PREFERRED_SIZE)
260 .addGap(47, 47, 47)
261 .addComponent(lblPrenom)
262 .addGap(18, 18, 18)
263 .addComponent(tfPrenom, javax.swing.GroupLayout.PREFERRED_SIZE, 99,
javax.swing.GroupLayout.PREFERRED_SIZE))))
264 .addGroup(jPanel1Layout.createSequentialGroup()
265 .addContainerGap()
266 .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
267 .addGroup(jPanel1Layout.createSequentialGroup()
268 .addComponent(lblDatedenaissance)
269 .addGap(18, 18, 18)
270 .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
271 .addComponent(jSeparator3)
272 .addComponent(tfNumtel, javax.swing.GroupLayout.DEFAULT_SIZE, 100, Short.MAX_VALUE)))
273 .addGroup(jPanel1Layout.createSequentialGroup()
274 .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
275 .addComponent(lblNumtel)
276 .addComponent(lblPseudo))
277 .addGap(18, 18, 18)
278 .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
279 .addComponent(jSeparator5)
280 .addComponent(tfPseudo, javax.swing.GroupLayout.DEFAULT_SIZE, 100, Short.MAX_VALUE)
281 .addComponent(tfDatedenaissance)
282 .addComponent(jSeparator4)))
283 .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
284 .addComponent(jSeparator7, javax.swing.GroupLayout.PREFERRED_SIZE, 100,
javax.swing.GroupLayout.PREFERRED_SIZE)
285 .addGroup(javax.swing.GroupLayout.Alignment.LEADING,
jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)
286 .addGroup(javax.swing.GroupLayout.Alignment.LEADING, jPanel1Layout.createSequentialGroup()
287 .addComponent(lblMotdepasse)
288 .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
289 .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
290 .addComponent(jSeparator6, javax.swing.GroupLayout.DEFAULT_SIZE, 100, Short.MAX_VALUE)
291 .addComponent(pfMotdepasse, javax.swing.GroupLayout.DEFAULT_SIZE, 1, Short.MAX_VALUE)))
292 .addGroup(jPanel1Layout.createSequentialGroup()
293 .addComponent(lblNumsocial)
294 .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 26, Short.MAX_VALUE)
295 .addComponent(tfNumsocial, javax.swing.GroupLayout.PREFERRED_SIZE, 100,
javax.swing.GroupLayout.PREFERRED_SIZE))))
296 .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
297 .addGroup(jPanel1Layout.createSequentialGroup()
298 .addGap(18, 18, 18)
299 .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
300 .addComponent(jLabel1)
301 .addGroup(jPanel1Layout.createSequentialGroup()
302 .addComponent(lblVille)
303 .addGap(18, 18, 18)
304 .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
305 .addComponent(jSeparator9, javax.swing.GroupLayout.PREFERRED_SIZE, 50,
javax.swing.GroupLayout.PREFERRED_SIZE)
306 .addComponent(tfVille, javax.swing.GroupLayout.PREFERRED_SIZE, 60,
javax.swing.GroupLayout.PREFERRED_SIZE))))
307 .addGroup(jPanel1Layout.createSequentialGroup()

```

```

308         .addGap(10, 10, 10)
309         .addComponent(lbErreur, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))))
310         .addContainerGap()
311     );
312     jPanel1Layout.setVerticalGroup(
313     jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
314     .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup())
315     .addGap(39, 39, 39)
316     .addComponent(lbInscription, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)
317     .addGap(46, 46, 46)
318     .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
319     .addGroup(jPanel1Layout.createSequentialGroup())
320     .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
321     .addComponent(lbNom)
322     .addComponent(tfNom, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
323     .addComponent(lbPrenom)
324     .addComponent(tfPrenom, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
325     .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
326     .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
327     .addComponent(jSeparator2, javax.swing.GroupLayout.PREFERRED_SIZE, 10,
javax.swing.GroupLayout.PREFERRED_SIZE)
328     .addComponent(jSeparator1, javax.swing.GroupLayout.PREFERRED_SIZE, 10,
javax.swing.GroupLayout.PREFERRED_SIZE))
329     .addGap(18, 18, 18)
330     .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
331     .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
332     .addComponent(lbDatedenaissance)
333     .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 28,
javax.swing.GroupLayout.PREFERRED_SIZE))
334     .addGroup(jPanel1Layout.createSequentialGroup())
335     .addComponent(tfNumtel, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
336     .addGap(5, 5, 5)
337     .addComponent(jSeparator3, javax.swing.GroupLayout.PREFERRED_SIZE, 10,
javax.swing.GroupLayout.PREFERRED_SIZE))
338     .addGap(18, 18, 18)
339     .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
340     .addComponent(lbNumtel)
341     .addComponent(tfDatedenaissance, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
342     .addComponent(lbVille)
343     .addComponent(tfVille, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
344     .addGap(2, 2, 2)
345     .addComponent(jSeparator4, javax.swing.GroupLayout.PREFERRED_SIZE, 10,
javax.swing.GroupLayout.PREFERRED_SIZE)
346     .addComponent(jSeparator9, javax.swing.GroupLayout.PREFERRED_SIZE, 10,
javax.swing.GroupLayout.PREFERRED_SIZE)
347     .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
348     .addGroup(jPanel1Layout.createSequentialGroup())
349     .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
350     .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
351     .addComponent(lbPseudo)
352     .addComponent(tfPseudo, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
353     .addGap(5, 5, 5)
354     .addComponent(jSeparator5, javax.swing.GroupLayout.PREFERRED_SIZE, 10,
javax.swing.GroupLayout.PREFERRED_SIZE)
355     .addGap(18, 18, 18)
356     .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
357     .addComponent(lbMotdepasse)
358     .addComponent(pfMotdepasse, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
359     .addGap(2, 2, 2)
360     .addComponent(jSeparator6, javax.swing.GroupLayout.PREFERRED_SIZE, 10,
javax.swing.GroupLayout.PREFERRED_SIZE)
361     .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup())
362     .addGap(28, 28, 28)
363     .addComponent(lbErreur, javax.swing.GroupLayout.PREFERRED_SIZE, 62,
javax.swing.GroupLayout.PREFERRED_SIZE))

```

```

364     .addGap(18, 18, 18)
365     .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
366         .addComponent(lbNumsocial)
367         .addComponent(tfNumsocial, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
368     .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
369         .addGroup(jPanel1Layout.createSequentialGroup()
370             .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
371                 .addComponent(rbPatient)
372                 .addComponent(rbPraticien))
373             .addGap(0, 0, Short.MAX_VALUE))
374         .addGroup(jPanel1Layout.createSequentialGroup()
375             .addGap(2, 2, 2)
376             .addComponent(jSeparator7, javax.swing.GroupLayout.PREFERRED_SIZE, 10,
javax.swing.GroupLayout.PREFERRED_SIZE)
377             .addGap(18, 18, 18)
378             .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
379                 .addComponent(lbSpecialite)
380                 .addComponent(tfSpecialite, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
381             .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
382             .addComponent(jSeparator8, javax.swing.GroupLayout.PREFERRED_SIZE, 10,
javax.swing.GroupLayout.PREFERRED_SIZE)
383             .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 8, Short.MAX_VALUE)
384             .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
385                 .addComponent(bRetour)
386                 .addComponent(bContinuer))))
387     .addContainerGap()
388 );
389
390 jMenuBar1.setBackground(new java.awt.Color(51, 51, 51));
391
392 jMenu1.setBackground(new java.awt.Color(51, 51, 51));
393 jMenu1.setText("Help");
394
395 jMenuItem1.setText("Aide");
396 jMenuItem1.addActionListener(new java.awt.event.ActionListener() {
397     public void actionPerformed(java.awt.event.ActionEvent evt) {
398         jMenuItem1ActionPerformed(evt);
399     }
400 });
401 jMenu1.add(jMenuItem1);
402
403 jMenuItem2.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_F4,
java.awt.event.InputEvent.ALT_MASK));
404 jMenuItem2.setText("Quitter");
405 jMenuItem2.addActionListener(new java.awt.event.ActionListener() {
406     public void actionPerformed(java.awt.event.ActionEvent evt) {
407         jMenuItem2ActionPerformed(evt);
408     }
409 });
410 jMenu1.add(jMenuItem2);
411
412 jMenuBar1.add(jMenu1);
413
414 setJMenuBar(jMenuBar1);
415
416 javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
417 getContentPane().setLayout(layout);
418 layout.setHorizontalGroup(
419     layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
420     .addGroup(layout.createSequentialGroup()
421         .addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
422         .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
423         .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
424 );
425 layout.setVerticalGroup(
426     layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
427     .addComponent(jPanel2, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
428     .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)

```

```

429     };
430
431     pack();
432 } // </editor-fold>
433
434 private void tfNomActionPerformed(java.awt.event.ActionEvent evt) {
435     // TODO add your handling code here:
436 }
437
438 private void tfPrenomActionPerformed(java.awt.event.ActionEvent evt) {
439     // TODO add your handling code here:
440 }
441
442 private void tfPseudoActionPerformed(java.awt.event.ActionEvent evt) {
443     // TODO add your handling code here:
444 }
445
446 private void jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt) {
447     JOptionPane.showMessageDialog(null, "Saisissez tous les champs requis afin de procéder à l'inscription ou bien retournez au
menu principal", "Message d'information", JOptionPane.INFORMATION_MESSAGE);
448 }
449
450 private void jMenuItem2ActionPerformed(java.awt.event.ActionEvent evt) {
451     this.dispose();
452 }
453
454 private void bRetourActionPerformed(java.awt.event.ActionEvent evt) {
455     this.setVisible(false);
456     this.getParent().setVisible(true);
457 }
458
459 private void bContinuerActionPerformed(java.awt.event.ActionEvent evt) {
460     int indication = 0;
461     int indicPraticien = -1;
462     String datedenaissance = tfDatedenaissance.getText();
463     String numtel = tfNumtel.getText();
464     String num = tfNumsocial.getText();
465     String spec = tfSpecialite.getText();
466     if (rbPatient.isSelected()){
467         indication=1;
468         indicPraticien = 1;
469     }
470     else if (rbPraticien.isSelected()){
471         indication=2;
472         indicPraticien = 0;
473     }
474
475     if(tfNom.getText().length()==0 || tfPrenom.getText().length()==0 || tfNumtel.getText().length()==0 ||
datedenaissance.length()==0 || tfPseudo.getText().length()==0 || pfMotdepasse.getText().length()==0 || num.length()==0 ||
spec.length()==0 || num.length()==0 || tfVille.getText().length()==0){
476         lbErreur.setVisible(true);
477     }
478     if(tfNom.getText().length()!=0 && tfPrenom.getText().length()!=0 && tfNumtel.getText().length()!=0 &&
datedenaissance.length()!=0 && tfPseudo.getText().length()!=0 && pfMotdepasse.getText().length()!=0 && num.length()!=0 &&
spec.length()!=0 && tfVille.getText().length()!=0){
479         if(User.inscription(indication, tfNom.getText(), tfPrenom.getText(), Integer.parseInt(datedenaissance),
Integer.parseInt(numtel), tfPseudo.getText(), pfMotdepasse.getText(), Integer.parseInt(num), Integer.parseInt(spec),
tfVille.getText())==1)
480         {
481             JOptionPane.showMessageDialog(null, "TROLOLOLOLOLOL");
482             fichselection = new Fselection(this, false, indicPraticien, User);
483             this.setVisible(false);
484             fichselection.setVisible(true);
485         }
486         else if(User.inscription(indication, tfNom.getText(), tfPrenom.getText(), Integer.parseInt(datedenaissance),
Integer.parseInt(numtel), tfPseudo.getText(), pfMotdepasse.getText(), Integer.parseInt(num), Integer.parseInt(spec),
tfVille.getText())==2 || User.inscription(indication, tfNom.getText(), tfPrenom.getText(), Integer.parseInt(datedenaissance),
Integer.parseInt(numtel), tfPseudo.getText(), pfMotdepasse.getText(), Integer.parseInt(num), Integer.parseInt(spec),
tfVille.getText())==0){
487             JOptionPane.showMessageDialog(null, "Vous ne vous etes pas inscrit correctement");
488         }
489     }
490
491

```



```

492 }
493
494 private void rbPatientActionPerformed(java.awt.event.ActionEvent evt) {
495 // if(rbPatient.isSelected()==true){
496 //     tfSpecialite.setVisible(false);
497 //     lbSpecialite.setVisible(false);
498 // }
499 }
500
501 private void rbPraticienActionPerformed(java.awt.event.ActionEvent evt) {
502 // if(rbPraticien.isSelected()==true){
503 //     tfSpecialite.setVisible(true);
504 //     lbSpecialite.setVisible(true);
505 // }
506 }
507
508 private void tfSpecialiteActionPerformed(java.awt.event.ActionEvent evt) {
509 // TODO add your handling code here:
510 }
511
512 private void tfNumsocialActionPerformed(java.awt.event.ActionEvent evt) {
513 // TODO add your handling code here:
514 }
515
516 /**
517  * @param args the command line arguments
518  */
519 public static void main(String args[]) {
520     /* Set the Nimbus look and feel */
521     //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
522     /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
523      * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
524      */
525     try {
526         for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
527             if ("Nimbus".equals(info.getName())) {
528                 javax.swing.UIManager.setLookAndFeel(info.getClassName());
529                 break;
530             }
531         }
532     } catch (ClassNotFoundException ex) {
533         java.util.logging.Logger.getLogger(Finscription.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
534     } catch (InstantiationException ex) {
535         java.util.logging.Logger.getLogger(Finscription.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
536     } catch (IllegalAccessException ex) {
537         java.util.logging.Logger.getLogger(Finscription.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
538     } catch (javax.swing.UnsupportedLookAndFeelException ex) {
539         java.util.logging.Logger.getLogger(Finscription.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
540     }
541     //</editor-fold>
542
543     /* Create and display the dialog */
544     java.awt.EventQueue.invokeLater(new Runnable() {
545         public void run() {
546             Finscription dialog = new Finscription(new javax.swing.JFrame(), true);
547             dialog.addWindowListener(new java.awt.event.WindowAdapter() {
548                 @Override
549                 public void windowClosing(java.awt.event.WindowEvent e) {
550                     System.exit(0);
551                 }
552             });
553             dialog.setVisible(true);
554         }
555     });
556 }
557
558 // Variables declaration - do not modify
559 private javax.swing.JButton bContinuer;
560 private javax.swing.JButton bRetour;
561 private javax.swing.ButtonGroup buttonGroup1;
562 private javax.swing.JLabel jLabel1;
563 private javax.swing.JMenu jMenu1;
564 private javax.swing.JMenuBar jMenuBar1;
565 private javax.swing.JMenuItem jMenuItem1;

```

```

566 private javax.swing.JMenuItem jMenuItem2;
567 private javax.swing.JPanel jPanel1;
568 private javax.swing.JPanel jPanel2;
569 private javax.swing.JSeparator jSeparator1;
570 private javax.swing.JSeparator jSeparator2;
571 private javax.swing.JSeparator jSeparator3;
572 private javax.swing.JSeparator jSeparator4;
573 private javax.swing.JSeparator jSeparator5;
574 private javax.swing.JSeparator jSeparator6;
575 private javax.swing.JSeparator jSeparator7;
576 private javax.swing.JSeparator jSeparator8;
577 private javax.swing.JSeparator jSeparator9;
578 private javax.swing.JLabel lbDatedenaissance;
579 private javax.swing.JLabel lbErreur;
580 private javax.swing.JLabel lbInscription;
581 private javax.swing.JLabel lbMotdepasse;
582 private javax.swing.JLabel lbNom;
583 private javax.swing.JLabel lbNumsocial;
584 private javax.swing.JLabel lbNumtel;
585 private javax.swing.JLabel lbPrenom;
586 private javax.swing.JLabel lbPseudo;
587 private javax.swing.JLabel lbSpecialite;
588 private javax.swing.JLabel lbVille;
589 private javax.swing.JPasswordField pfMotdepasse;
590 private javax.swing.JRadioButton rbPatient;
591 private javax.swing.JRadioButton rbPraticien;
592 private java.awt.ScrollPane scrollPane1;
593 private javax.swing.JTextField tfDatedenaissance;
594 private javax.swing.JTextField tfNom;
595 private javax.swing.JTextField tfNumsocial;
596 private javax.swing.JTextField tfNumtel;
597 private javax.swing.JTextField tfPrenom;
598 private javax.swing.JTextField tfPseudo;
599 private javax.swing.JTextField tfSpecialite;
600 private javax.swing.JTextField tfVille;
601 // End of variables declaration
602 }
603

```

D:\Documents\EPF\2A\Projet Java\RDV\_MEDIC\_SUTH\_Ranjit\RDV\_MEDIC\_SUTH\_Ranjit\inscription\JAVA\_FINAL - Copie - Copie\JAVA\_FINAL\RDV\_MEDICAUX\_SUTHARSAN\_RANJIT\src\pFiches\Fselection.java

```

1 /*
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6 package pFiches;
7
8 import javax.swing.JOptionPane;
9 import rdv_medicaux_sutharsan_ranjit.Utilisateurs;
10
11 /**
12  *
13  * @author Vithusan
14  */
15 public class Fselection extends javax.swing.JDialog {
16
17     /**
18      * Creates new form Fselection
19      */
20     private int type=-1;
21     private Utilisateurs User;
22     public Fselection(java.awt.Frame parent, boolean modal) {
23         super(parent, modal);
24         initComponents();
25     }
26
27     public Fselection(javax.swing.JDialog parent, boolean modal,int type, Utilisateurs u) {
28         super(parent, modal);
29         initComponents();
30         User = u;
31         this.type=type;
32     }
33     public Fselection(javax.swing.JDialog parent, boolean modal, Utilisateurs u) {
34         super(parent, modal);
35     }
36

```

```

35     initComponents();
36     User = u;
37
38 }
39 public Fselection(java.awt.Frame parent, boolean modal, Utilisateurs u) {
40     super(parent, modal);
41     initComponents();
42     User = u;
43
44 }
45
46 /**
47  * This method is called from within the constructor to initialize the form.
48  * WARNING: Do NOT modify this code. The content of this method is always
49  * regenerated by the Form Editor.
50  */
51 @SuppressWarnings("unchecked")
52 // <editor-fold defaultstate="collapsed" desc="Generated Code">
53 private void initComponents() {
54
55     buttonGroup1 = new javax.swing.ButtonGroup();
56     buttonGroup2 = new javax.swing.ButtonGroup();
57     buttonGroup3 = new javax.swing.ButtonGroup();
58     buttonGroup4 = new javax.swing.ButtonGroup();
59     buttonGroup5 = new javax.swing.ButtonGroup();
60     buttonGroup6 = new javax.swing.ButtonGroup();
61     jPanel1 = new javax.swing.JPanel();
62     rbListprat = new javax.swing.JRadioButton();
63     rbPriserdv = new javax.swing.JRadioButton();
64     rbProposerrdv = new javax.swing.JRadioButton();
65     jRetour = new javax.swing.JButton();
66     jContinuer = new javax.swing.JButton();
67     jLabel1 = new javax.swing.JLabel();
68     rbGeneraliste = new javax.swing.JRadioButton();
69     rbSpecialiste = new javax.swing.JRadioButton();
70     rbGeneralistebis = new javax.swing.JRadioButton();
71     rbSpecialistebis = new javax.swing.JRadioButton();
72     rbConsulter = new javax.swing.JRadioButton();
73     rbPrise = new javax.swing.JRadioButton();
74     rbAnnuler = new javax.swing.JRadioButton();
75     rbConsulterRDV = new javax.swing.JRadioButton();
76     rbAjouter = new javax.swing.JRadioButton();
77     rbModif = new javax.swing.JRadioButton();
78     jScrollPane1 = new javax.swing.JScrollPane();
79     jList1 = new javax.swing.JList<>();
80     tfName = new javax.swing.JTextField();
81     lbName = new javax.swing.JLabel();
82     tfNum = new javax.swing.JTextField();
83     lbNum = new javax.swing.JLabel();
84     lbVille = new javax.swing.JLabel();
85     tfVille = new javax.swing.JTextField();
86     rbAffiner = new javax.swing.JRadioButton();
87     rbVoir = new javax.swing.JRadioButton();
88     rbAnnulerRDV = new javax.swing.JRadioButton();
89     tfSaisieDate = new javax.swing.JTextField();
90     lbSaisieDate = new javax.swing.JLabel();
91     tfSaisieNum = new javax.swing.JTextField();
92     lbSaisieNum = new javax.swing.JLabel();
93     jMenuBar1 = new javax.swing.JMenuBar();
94     jMenu1 = new javax.swing.JMenu();
95     jMenuItem1 = new javax.swing.JMenuItem();
96     jMenuItem2 = new javax.swing.JMenuItem();
97
98     setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
99     setBackground(new java.awt.Color(0, 102, 102));
100    addWindowListener(new java.awt.event.WindowAdapter() {
101        public void windowActivated(java.awt.event.WindowEvent evt) {
102            formWindowActivated(evt);
103        }
104    });
105
106    jPanel1.setBackground(new java.awt.Color(0, 102, 102));
107
108    rbListprat.setBackground(new java.awt.Color(0, 102, 102));

```



```

109 buttonGroup1.add(rbListprat);
110 rbListprat.setText("Consulter la liste de praticiens");
111 rbListprat.addActionListener(new java.awt.event.ActionListener() {
112     public void actionPerformed(java.awt.event.ActionEvent evt) {
113         rbListpratActionPerformed(evt);
114     }
115 });
116
117 rbPriserdv.setBackground(new java.awt.Color(0, 102, 102));
118 buttonGroup1.add(rbPriserdv);
119 rbPriserdv.setText("Prendre un rendez-vous");
120 rbPriserdv.addActionListener(new java.awt.event.ActionListener() {
121     public void actionPerformed(java.awt.event.ActionEvent evt) {
122         rbPriserdvActionPerformed(evt);
123     }
124 });
125
126 rbProposerrdv.setBackground(new java.awt.Color(0, 102, 102));
127 buttonGroup1.add(rbProposerrdv);
128 rbProposerrdv.setText("Gérer vos rendez-vous");
129 rbProposerrdv.addActionListener(new java.awt.event.ActionListener() {
130     public void actionPerformed(java.awt.event.ActionEvent evt) {
131         rbProposerrdvActionPerformed(evt);
132     }
133 });
134
135 jRetour.setBackground(new java.awt.Color(0, 102, 102));
136 jRetour.setText("Retour");
137 jRetour.addActionListener(new java.awt.event.ActionListener() {
138     public void actionPerformed(java.awt.event.ActionEvent evt) {
139         jRetourActionPerformed(evt);
140     }
141 });
142
143 jContinuer.setBackground(new java.awt.Color(0, 102, 102));
144 jContinuer.setText("Continuer");
145 jContinuer.addActionListener(new java.awt.event.ActionListener() {
146     public void actionPerformed(java.awt.event.ActionEvent evt) {
147         jContinuerActionPerformed(evt);
148     }
149 });
150
151 jLabel1.setFont(new java.awt.Font("Tahoma", 0, 24)); // NOI18N
152 jLabel1.setText("Que souhaitez- vous faire ?");
153
154 rbGeneraliste.setBackground(new java.awt.Color(0, 102, 102));
155 buttonGroup2.add(rbGeneraliste);
156 rbGeneraliste.setText("Généraliste");
157
158 rbSpecialiste.setBackground(new java.awt.Color(0, 102, 102));
159 buttonGroup2.add(rbSpecialiste);
160 rbSpecialiste.setText("Spécialiste");
161
162 rbGeneralistebis.setBackground(new java.awt.Color(0, 102, 102));
163 buttonGroup3.add(rbGeneralistebis);
164 rbGeneralistebis.setText("Généraliste");
165 rbGeneralistebis.addActionListener(new java.awt.event.ActionListener() {
166     public void actionPerformed(java.awt.event.ActionEvent evt) {
167         rbGeneralistebisActionPerformed(evt);
168     }
169 });
170
171 rbSpecialistebis.setBackground(new java.awt.Color(0, 102, 102));
172 buttonGroup3.add(rbSpecialistebis);
173 rbSpecialistebis.setText("Spécialiste");
174 rbSpecialistebis.addActionListener(new java.awt.event.ActionListener() {
175     public void actionPerformed(java.awt.event.ActionEvent evt) {
176         rbSpecialistebisActionPerformed(evt);
177     }
178 });
179
180 rbConsulter.setBackground(new java.awt.Color(0, 102, 102));
181 buttonGroup4.add(rbConsulter);
182 rbConsulter.setText("Consulter les RDV");

```

```

183 rbConsulter.addActionListener(new java.awt.event.ActionListener() {
184     public void actionPerformed(java.awt.event.ActionEvent evt) {
185         rbConsulterActionPerformed(evt);
186     }
187 });
188
189 rbPrise.setBackground(new java.awt.Color(0, 102, 102));
190 buttonGroup4.add(rbPrise);
191 rbPrise.setText("Prendre un RDV");
192 rbPrise.addActionListener(new java.awt.event.ActionListener() {
193     public void actionPerformed(java.awt.event.ActionEvent evt) {
194         rbPriseActionPerformed(evt);
195     }
196 });
197
198 rbAnnuler.setBackground(new java.awt.Color(0, 102, 102));
199 buttonGroup4.add(rbAnnuler);
200 rbAnnuler.setText("Annuler un RDV");
201 rbAnnuler.addActionListener(new java.awt.event.ActionListener() {
202     public void actionPerformed(java.awt.event.ActionEvent evt) {
203         rbAnnulerActionPerformed(evt);
204     }
205 });
206
207 rbConsulterRDV.setBackground(new java.awt.Color(0, 102, 102));
208 buttonGroup5.add(rbConsulterRDV);
209 rbConsulterRDV.setText("Consulter vos RDV(s)");
210
211 rbAjouter.setBackground(new java.awt.Color(0, 102, 102));
212 buttonGroup5.add(rbAjouter);
213 rbAjouter.setText("Ajouter un RDV");
214
215 rbModif.setBackground(new java.awt.Color(0, 102, 102));
216 buttonGroup5.add(rbModif);
217 rbModif.setText("Modifier un RDV");
218
219 jList1.setBackground(new java.awt.Color(0, 153, 153));
220 jScrollPane1.setViewportView(jList1);
221
222 lbName.setText("Saisissez le nom du praticien en question");
223
224 lbNum.setText("Saisissez l'indice de la date ou la date");
225
226 lbVille.setText("Saisissez la ville");
227
228 tfVille.addActionListener(new java.awt.event.ActionListener() {
229     public void actionPerformed(java.awt.event.ActionEvent evt) {
230         tfVilleActionPerformed(evt);
231     }
232 });
233
234 rbAffiner.setBackground(new java.awt.Color(0, 102, 102));
235 buttonGroup6.add(rbAffiner);
236 rbAffiner.setText("Affiner par ville");
237
238 rbVoir.setBackground(new java.awt.Color(0, 102, 102));
239 buttonGroup6.add(rbVoir);
240 rbVoir.setText("Voir tous les praticiens");
241
242 rbAnnulerRDV.setBackground(new java.awt.Color(0, 102, 102));
243 buttonGroup5.add(rbAnnulerRDV);
244 rbAnnulerRDV.setText("Annuler un RDV");
245
246 lbSaisieDate.setText("Saisissez la date");
247
248 lbSaisieNum.setText("Saisissez l'indice de la date");
249
250 javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
251 jPanel1.setLayout(jPanel1Layout);
252 jPanel1Layout.setHorizontalGroup(
253     jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
254         .addGroup(jPanel1Layout.createSequentialGroup()
255             .addContainerGap()
256             .addComponent(jRetour)

```

```

257 .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
258 .addComponent(jContinuer)
259 .addContainerGap()
260 .addGroup(jPanel1Layout.createSequentialGroup())
261 .addGap(18, 18, 18)
262 .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
263 .addComponent(rbProposerrdv)
264 .addGroup(jPanel1Layout.createSequentialGroup())
265 .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
266 .addComponent(rbPriserdv)
267 .addComponent(rbListprat))
268 .addGap(50, 50, 50)
269 .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
270 .addGroup(jPanel1Layout.createSequentialGroup())
271 .addComponent(rbAffiner)
272 .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
273 .addComponent(rbVoir))
274 .addGroup(jPanel1Layout.createSequentialGroup())
275 .addComponent(rbGeneralistebis)
276 .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
277 .addComponent(rbSpecialistebis))
278 .addGroup(jPanel1Layout.createSequentialGroup())
279 .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)
280 .addComponent(tfNum, javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.DEFAULT_SIZE, 49, Short.MAX_VALUE)
281 .addComponent(tfName, javax.swing.GroupLayout.Alignment.LEADING))
282 .addGap(18, 18, 18)
283 .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
284 .addComponent(lbName)
285 .addComponent(lbNum)))
286 .addGroup(jPanel1Layout.createSequentialGroup())
287 .addComponent(rbGeneraliste)
288 .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
289 .addComponent(rbSpecialiste))
290 .addGroup(jPanel1Layout.createSequentialGroup())
291 .addComponent(tfVille, javax.swing.GroupLayout.PREFERRED_SIZE, 65,
javax.swing.GroupLayout.PREFERRED_SIZE)
292 .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
293 .addComponent(lbVille))
294 .addGroup(jPanel1Layout.createSequentialGroup())
295 .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
296 .addGroup(jPanel1Layout.createSequentialGroup())
297 .addComponent(rbConsulterRDV)
298 .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
299 .addComponent(rbAjouter))
300 .addGroup(jPanel1Layout.createSequentialGroup())
301 .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)
302 .addComponent(tfSaisieNum, javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.DEFAULT_SIZE, 50, Short.MAX_VALUE)
303 .addComponent(tfSaisieDate, javax.swing.GroupLayout.Alignment.LEADING))
304 .addGap(23, 23, 23)
305 .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
306 .addComponent(lbSaisieDate)
307 .addComponent(lbSaisieNum)))
308 .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
309 .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
310 .addComponent(rbAnnulerRDV)
311 .addComponent(rbModif))
312 .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
313 .addComponent(jLabel1)
314 .addGroup(jPanel1Layout.createSequentialGroup())
315 .addComponent(rbConsulter)
316 .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
317 .addComponent(rbPrise)
318 .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
319 .addComponent(rbAnnuler))))))
320 .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 75, Short.MAX_VALUE)
321 .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 227,
javax.swing.GroupLayout.PREFERRED_SIZE)
322 .addGap(24, 24, 24))
323 );
324 jPanel1Layout.setVerticalGroup(
325 jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

```

```

326 .addGroup(jPanel1Layout.createSequentialGroup())
327 .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
328 .addGroup(jPanel1Layout.createSequentialGroup())
329 .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
330 .addGroup(jPanel1Layout.createSequentialGroup())
331 .addGap(24, 24, 24)
332 .addComponent(jLabel1)
333 .addGap(18, 18, 18)
334 .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
335 .addComponent(rbListprat)
336 .addComponent(rbAffiner)
337 .addComponent(rbVoir))
338 .addGap(18, 18, 18)
339 .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
340 .addComponent(rbGeneraliste)
341 .addComponent(rbSpecialiste))
342 .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
343 .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
344 .addComponent(tfVille, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
345 .addComponent(lbVille))
346 .addGap(63, 63, 63)
347 .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
348 .addComponent(rbPriserdv)
349 .addComponent(rbGeneralistebis)
350 .addComponent(rbSpecialistebis))
351 .addGap(18, 18, 18)
352 .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
353 .addComponent(rbConsulter)
354 .addComponent(rbPrise)
355 .addComponent(rbAnnuler))
356 .addGap(18, 18, 18)
357 .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
358 .addComponent(tfName, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
359 .addComponent(lbName))
360 .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
361 .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
362 .addComponent(tfNum, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
363 .addComponent(lbNum))
364 .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 10, Short.MAX_VALUE)
365 .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
366 .addComponent(rbProposerrdv)
367 .addComponent(rbConsulterRDV)
368 .addComponent(rbAjouter)
369 .addComponent(rbModifier))
370 .addGroup(jPanel1Layout.createSequentialGroup())
371 .addGap(85, 85, 85)
372 .addComponent(jScrollPane1))
373 .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
374 .addComponent(rbAnnulerRDV)
375 .addGap(17, 17, 17))
376 .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup())
377 .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
378 .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
379 .addComponent(tfSaisieDate, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
380 .addComponent(lbSaisieDate))
381 .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)))
382 .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
383 .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
384 .addComponent(jRetour)
385 .addComponent(jContinuer)
386 .addComponent(lbSaisieNum))
387 .addComponent(tfSaisieNum, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
388 .addContainerGap()
389 );
390
391 jMenuBar1.setBackground(new java.awt.Color(51, 51, 51));
392
393 jMenuItem1.setBackground(new java.awt.Color(51, 51, 51));
394 jMenuItem1.setText("Help");

```

```

395
396 jMenuItem1.setText("Aide");
397 jMenuItem1.addActionListener(new java.awt.event.ActionListener() {
398     public void actionPerformed(java.awt.event.ActionEvent evt) {
399         jMenuItem1ActionPerformed(evt);
400     }
401 });
402 jMenuItem1.add(jMenuItem1);
403
404 jMenuItem2.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_F4,
java.awt.event.InputEvent.ALT_MASK));
405 jMenuItem2.setText("Quitter");
406 jMenuItem2.addActionListener(new java.awt.event.ActionListener() {
407     public void actionPerformed(java.awt.event.ActionEvent evt) {
408         jMenuItem2ActionPerformed(evt);
409     }
410 });
411 jMenuItem2.add(jMenuItem2);
412
413 jMenuItemBar1.add(jMenuItem);
414
415 setJMenuBar(jMenuBar1);
416
417 javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
418 getContentPane().setLayout(layout);
419 layout.setHorizontalGroup(
420     layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
421         .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
422 );
423 layout.setVerticalGroup(
424     layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
425         .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
426 );
427
428 pack();
429 } // </editor-fold>
430
431 private void jMenuItem2ActionPerformed(java.awt.event.ActionEvent evt) {
432     this.dispose();
433 }
434
435 private void jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt) {
436     JOptionPane.showMessageDialog(null, "Sélectionnez votre choix pour accéder aux différentes fonctionnalités du site. Les
visiteurs auront un accès limité du site, sinon quitter le site en allant dans Aide puis Quitter", "Message d'information",
JOptionPane.INFORMATION_MESSAGE);
437 }
438
439 private void rbPriserdvActionPerformed(java.awt.event.ActionEvent evt) {
440     rbGeneraliste.setVisible(false);
441     rbSpecialiste.setVisible(false);
442     rbConsulterRDV.setVisible(false);
443     rbAjouter.setVisible(false);
444     rbModif.setVisible(false);
445     rbAnnulerRDV.setVisible(false);
446     lbSaisieDate.setVisible(false);
447     tfSaisieDate.setVisible(false);
448     lbSaisieNum.setVisible(false);
449     tfSaisieNum.setVisible(false);
450     rbGeneralistebis.setVisible(true);
451     rbSpecialistebis.setVisible(true);
452     tfName.setVisible(true);
453     tfNum.setVisible(true);
454     lbNum.setVisible(true);
455     lbName.setVisible(true);
456     rbAffiner.setVisible(false);
457     rbVoir.setVisible(false);
458     tfVille.setVisible(false);
459     lbVille.setVisible(false);
460 }
461
462 private void jRetourActionPerformed(java.awt.event.ActionEvent evt) {
463     this.setVisible(false);

```

```

464     this.getParent().setVisible(true);
465 }
466
467 private void jContinuerActionPerformed(java.awt.event.ActionEvent evt) {
468
469     String num = tfNum.getText();
470     String date = tfSaisieDate.getText();
471     String indice = tfSaisieNum.getText();
472
473     if(rbGeneraliste.isSelected()==true){
474         if(rbAffiner.isSelected()==false && rbVoir.isSelected()==false){
475             JOptionPane.showMessageDialog(null, "Veuillez faire un choix", "Message d'erreur", JOptionPane.ERROR_MESSAGE);
476         }
477         else{
478             if(rbAffiner.isSelected()==true){
479                 User.consulterPraticienVille(1, jList1, tfVille.getText());
480             }
481             if(rbVoir.isSelected()==true){
482                 User.consulterPraticien(1, jList1);
483             }
484         }
485     }
486 }
487
488 else if(rbSpecialiste.isSelected()==true){
489     if(rbAffiner.isSelected()==false && rbVoir.isSelected()==false){
490         JOptionPane.showMessageDialog(null, "Veuillez faire un choix", "Message d'erreur", JOptionPane.ERROR_MESSAGE);
491     }
492     else{
493         if(rbAffiner.isSelected()==true){
494             User.consulterPraticienVille(2, jList1, tfVille.getText());
495         }
496         if(rbVoir.isSelected()==true){
497             User.consulterPraticien(2, jList1);
498         }
499     }
500 }
501
502 else if(rbPriserdv.isSelected()==true){
503     if(rbSpecialistebis.isSelected()==false && rbGeneralistebis.isSelected()==false){
504         JOptionPane.showMessageDialog(null, "Veuillez faire un choix", "Message d'erreur", JOptionPane.ERROR_MESSAGE);
505     }
506 }
507
508 if(rbGeneralistebis.isSelected()==true){
509     if(rbConsulter.isSelected()==false && rbPrise.isSelected()==false && rbAnnuler.isSelected()==false){
510         JOptionPane.showMessageDialog(null, "Veuillez faire un choix", "Message d'erreur", JOptionPane.ERROR_MESSAGE);
511     }
512     else{
513         if(rbConsulter.isSelected()==true){
514             User.lookrdv(1, tfName.getText(), jList1);
515         }
516         if(rbPrise.isSelected()==true){
517             if(num.length()==0 || tfName.getText().length()==0){
518                 JOptionPane.showMessageDialog(null, "Saisissez l'indice de la date et le praticien en question", "Message d'avertissement", JOptionPane.ERROR_MESSAGE);
519             }
520             if(num.length()!=0 && tfName.getText().length()!=0){
521                 User.prendrerdv(1, tfName.getText(), jList1, Integer.parseInt(num));
522             }
523         }
524         if(rbAnnuler.isSelected()==true){
525             if(num.length()==0 || tfName.getText().length()==0){
526                 JOptionPane.showMessageDialog(null, "Saisissez l'indice de la date", "Message d'avertissement", JOptionPane.ERROR_MESSAGE);
527             }
528             if(num.length()!=0 && tfName.getText().length()!=0){
529                 User.annulerRDV(1, tfName.getText(), jList1, Integer.parseInt(num));
530             }
531         }
532     }
533 }
534 }
535

```



```

536 if(rbSpecialistebis.isSelected()==true){
537     if(rbConsulter.isSelected()==false && rbPrise.isSelected()==false && rbAnnuler.isSelected()==false){
538         JOptionPane.showMessageDialog(null, "Veuillez faire un choix", "Message d'erreur", JOptionPane.ERROR_MESSAGE);
539     }
540     else{
541         if(rbConsulter.isSelected()==true){
542             User.lookrdv(2, tfName.getText(), jList1);
543         }
544         if(rbPrise.isSelected()==true){
545             if(num.length()==0){
546                 JOptionPane.showMessageDialog(null, "Saisissez l'indice de la date", "Message d'avertissement",
JOptionPane.ERROR_MESSAGE);
547             }
548             if(num.length()!=0){
549                 User.prendrerdv(2, tfName.getText(), jList1, Integer.parseInt(num));
550             }
551         }
552         if(rbAnnuler.isSelected()==true){
553             if(num.length()==0){
554                 JOptionPane.showMessageDialog(null, "Saisissez l'indice de la date", "Message d'avertissement",
JOptionPane.ERROR_MESSAGE);
555             }
556             if(num.length()!=0){
557                 User.annulerRDV(2, tfName.getText(), jList1, Integer.parseInt(num));
558             }
559         }
560     }
561 }
562 }
563
564 else if(rbProposerrdv.isSelected()==true){
565     if(rbConsulterRDV.isSelected()==false && rbAjouter.isSelected()==false && rbModif.isSelected()==false &&
rbAnnulerRDV.isSelected()==false){
566         JOptionPane.showMessageDialog(null, "Veuillez faire un choix", "Message d'erreur", JOptionPane.ERROR_MESSAGE);
567     }
568 }
569
570 if(rbConsulterRDV.isSelected()==true){
571     User.consulterSesRDV(jList1);
572 }
573
574 if(rbAjouter.isSelected()==true){
575     if(date.length()==0){
576         JOptionPane.showMessageDialog(null, "Saisissez la date souhaitée", "Message d'avertissement",
JOptionPane.ERROR_MESSAGE);
577     }
578     if(date.length()!=0){
579         User.ajouterRDV(jList1, Integer.parseInt(date));
580     }
581 }
582
583 if(rbModif.isSelected()==true){
584     if(date.length()==0 || indice.length()==0){
585         JOptionPane.showMessageDialog(null, "Saisissez l'indice de la date et la date souhaitée", "Message d'avertissement",
JOptionPane.ERROR_MESSAGE);
586     }
587     if(date.length()!=0 || indice.length()!=0){
588         User.modifierRDV(jList1, Integer.parseInt(indice), Integer.parseInt(date));
589     }
590 }
591
592 if(rbAnnulerRDV.isSelected()==true){
593     if(indice.length()==0){
594         JOptionPane.showMessageDialog(null, "Saisissez l'indice de la date", "Message d'avertissement",
JOptionPane.ERROR_MESSAGE);
595     }
596     if(indice.length()!=0){
597         User.annulerRDV(jList1, Integer.parseInt(indice));
598     }
599 }
600
601 else if(rbListprat.isSelected()==false && rbPriserdv.isSelected()==false && rbProposerrdv.isSelected()==false){
602     JOptionPane.showMessageDialog(null, "Veuillez faire un choix", "Message d'information",
JOptionPane.INFORMATION_MESSAGE);

```

```

603     }
604 }
605
606 private void formWindowActivated(java.awt.event.WindowEvent evt) {
607
608     rbGeneraliste.setVisible(false);
609     rbSpecialiste.setVisible(false);
610     rbGeneralistebis.setVisible(false);
611     rbSpecialistebis.setVisible(false);
612     rbConsulter.setVisible(false);
613     rbPrise.setVisible(false);
614     rbAnnuler.setVisible(false);
615     rbConsulterRDV.setVisible(false);
616     rbAjouter.setVisible(false);
617     rbModif.setVisible(false);
618     rbAnnulerRDV.setVisible(false);
619     lbSaisieDate.setVisible(false);
620     tfSaisieDate.setVisible(false);
621     lbSaisieNum.setVisible(false);
622     tfSaisieNum.setVisible(false);
623     tfName.setVisible(false);
624     tfNum.setVisible(false);
625     lbNum.setVisible(false);
626     lbName.setVisible(false);
627     rbAffiner.setVisible(false);
628     rbVoir.setVisible(false);
629     tfVille.setVisible(false);
630     lbVille.setVisible(false);
631     if (type== -1){ //si c'est un visiteur
632         rbPriserdv.setVisible(false);
633         rbProposerrdv.setVisible(false);
634     }
635     else if(type ==1){ //si c'est un patient
636         rbProposerrdv.setVisible(false);
637     }
638     if(rbPriserdv.isSelected()==true){
639         rbGeneralistebis.setVisible(true);
640         rbSpecialistebis.setVisible(true);
641         tfName.setVisible(true);
642         tfNum.setVisible(true);
643         lbNum.setVisible(true);
644         lbName.setVisible(true);
645     }
646 }
647
648 private void rbListpratActionPerformed(java.awt.event.ActionEvent evt) {
649     rbGeneraliste.setVisible(true);
650     rbSpecialiste.setVisible(true);
651     rbAffiner.setVisible(true);
652     rbVoir.setVisible(true);
653     tfVille.setVisible(true);
654     lbVille.setVisible(true);
655     rbGeneralistebis.setVisible(false);
656     rbSpecialistebis.setVisible(false);
657     rbConsulter.setVisible(false);
658     rbPrise.setVisible(false);
659     rbAnnuler.setVisible(false);
660     rbConsulterRDV.setVisible(false);
661     rbAjouter.setVisible(false);
662     rbModif.setVisible(false);
663     rbAnnulerRDV.setVisible(false);
664     lbSaisieDate.setVisible(false);
665     tfSaisieDate.setVisible(false);
666     lbSaisieNum.setVisible(false);
667     tfSaisieNum.setVisible(false);
668     tfName.setVisible(false);
669     tfNum.setVisible(false);
670     lbNum.setVisible(false);
671     lbName.setVisible(false);
672 }
673
674 private void rbProposerrdvActionPerformed(java.awt.event.ActionEvent evt) {
675     rbGeneraliste.setVisible(false);
676     rbSpecialiste.setVisible(false);

```



```

677 rbGeneralistebis.setVisible(false);
678 rbSpecialistebis.setVisible(false);
679 rbConsulter.setVisible(false);
680 rbPrise.setVisible(false);
681 rbAnnuler.setVisible(false);
682 rbConsulterRDV.setVisible(true);
683 rbAjouter.setVisible(true);
684 rbModif.setVisible(true);
685 rbAnnulerRDV.setVisible(true);
686 lbSaisieDate.setVisible(true);
687 tfSaisieDate.setVisible(true);
688 lbSaisieNum.setVisible(true);
689 tfSaisieNum.setVisible(true);
690 tfName.setVisible(false);
691 tfNum.setVisible(false);
692 lbNum.setVisible(false);
693 lbName.setVisible(false);
694 rbAffiner.setVisible(false);
695 rbVoir.setVisible(false);
696 tfVille.setVisible(false);
697 lbVille.setVisible(false);
698 }
699
700 private void rbGeneralistebisActionPerformed(java.awt.event.ActionEvent evt) {
701     if(rbGeneralistebis.isSelected()==true){
702         rbConsulter.setVisible(true);
703         rbPrise.setVisible(true);
704         rbAnnuler.setVisible(true);
705     }
706 }
707
708 private void rbSpecialistebisActionPerformed(java.awt.event.ActionEvent evt) {
709     if(rbSpecialistebis.isSelected()==true){
710         rbConsulter.setVisible(true);
711         rbPrise.setVisible(true);
712         rbAnnuler.setVisible(true);
713     }
714 }
715
716 private void rbConsulterActionPerformed(java.awt.event.ActionEvent evt) {
717
718 }
719
720 private void rbPriseActionPerformed(java.awt.event.ActionEvent evt) {
721
722 }
723
724 private void rbAnnulerActionPerformed(java.awt.event.ActionEvent evt) {
725
726 }
727
728 private void tfVilleActionPerformed(java.awt.event.ActionEvent evt) {
729     // TODO add your handling code here:
730 }
731
732 /**
733  * @param args the command line arguments
734  */
735 public static void main(String args[]) {
736     /* Set the Nimbus look and feel */
737     /*<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
738     /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
739     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
740     */
741     try {
742         for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
743             if ("Nimbus".equals(info.getName())) {
744                 javax.swing.UIManager.setLookAndFeel(info.getClassName());
745                 break;
746             }
747         }
748     } catch (ClassNotFoundException ex) {
749         java.util.logging.Logger.getLogger(Fselection.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
750     } catch (InstantiationException ex) {

```

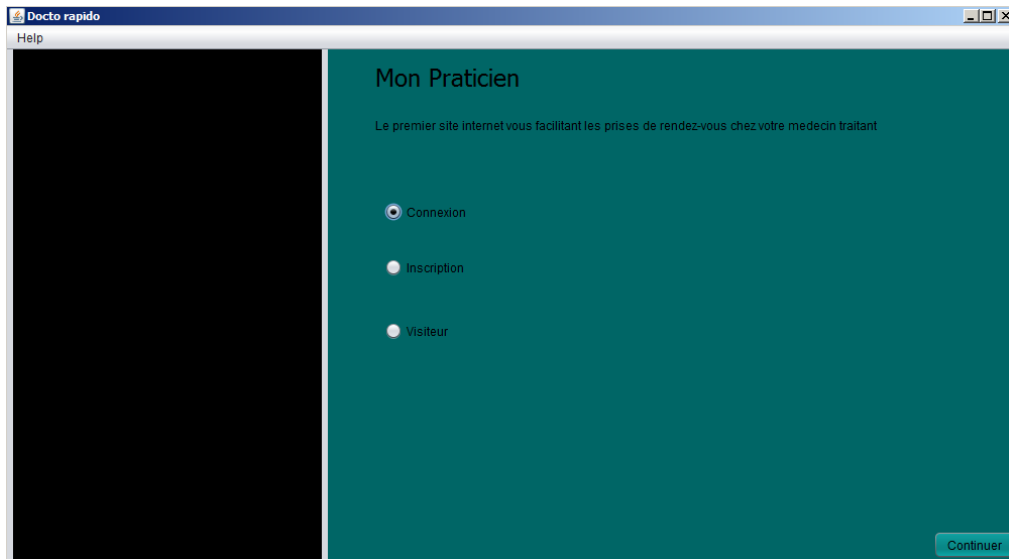
```

751     java.util.logging.Logger.getLogger(Fselection.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
752 } catch (IllegalAccessException ex) {
753     java.util.logging.Logger.getLogger(Fselection.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
754 } catch (javax.swing.UnsupportedLookAndFeelException ex) {
755     java.util.logging.Logger.getLogger(Fselection.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
756 }
757 //</editor-fold>
758
759 /* Create and display the dialog */
760 java.awt.EventQueue.invokeLater(new Runnable() {
761     public void run() {
762         Fselection dialog = new Fselection(new javax.swing.JFrame(), true);
763         dialog.addWindowListener(new java.awt.event.WindowAdapter() {
764             @Override
765             public void windowClosing(java.awt.event.WindowEvent e) {
766                 System.exit(0);
767             }
768         });
769         dialog.setVisible(true);
770     }
771 });
772 }
773
774 // Variables declaration - do not modify
775 private javax.swing.ButtonGroup buttonGroup1;
776 private javax.swing.ButtonGroup buttonGroup2;
777 private javax.swing.ButtonGroup buttonGroup3;
778 private javax.swing.ButtonGroup buttonGroup4;
779 private javax.swing.ButtonGroup buttonGroup5;
780 private javax.swing.ButtonGroup buttonGroup6;
781 private javax.swing.JButton jContinuer;
782 private javax.swing.JLabel jLabel1;
783 private javax.swing.JList<String> jList1;
784 private javax.swing.JMenu jMenu1;
785 private javax.swing.JMenuBar jMenuBar1;
786 private javax.swing.JMenuItem jMenuItem1;
787 private javax.swing.JMenuItem jMenuItem2;
788 private javax.swing.JPanel jPanel1;
789 private javax.swing.JButton jRetour;
790 private javax.swing.JScrollPane jScrollPane1;
791 private javax.swing.JLabel lbName;
792 private javax.swing.JLabel lbNum;
793 private javax.swing.JLabel lbSaisieDate;
794 private javax.swing.JLabel lbSaisieNum;
795 private javax.swing.JLabel lbVille;
796 private javax.swing.JRadioButton rbAffiner;
797 private javax.swing.JRadioButton rbAjouter;
798 private javax.swing.JRadioButton rbAnnuler;
799 private javax.swing.JRadioButton rbAnnulerRDV;
800 private javax.swing.JRadioButton rbConsulter;
801 private javax.swing.JRadioButton rbConsulterRDV;
802 private javax.swing.JRadioButton rbGeneraliste;
803 private javax.swing.JRadioButton rbGeneralistebis;
804 private javax.swing.JRadioButton rbListprat;
805 private javax.swing.JRadioButton rbModif;
806 private javax.swing.JRadioButton rbPrise;
807 private javax.swing.JRadioButton rbPriserdv;
808 private javax.swing.JRadioButton rbProposerrdv;
809 private javax.swing.JRadioButton rbSpecialiste;
810 private javax.swing.JRadioButton rbSpecialistebis;
811 private javax.swing.JRadioButton rbVoir;
812 private javax.swing.JTextField tfName;
813 private javax.swing.JTextField tfNum;
814 private javax.swing.JTextField tfSaisieDate;
815 private javax.swing.JTextField tfSaisieNum;
816 private javax.swing.JTextField tfVille;
817 // End of variables declaration
818 }

```

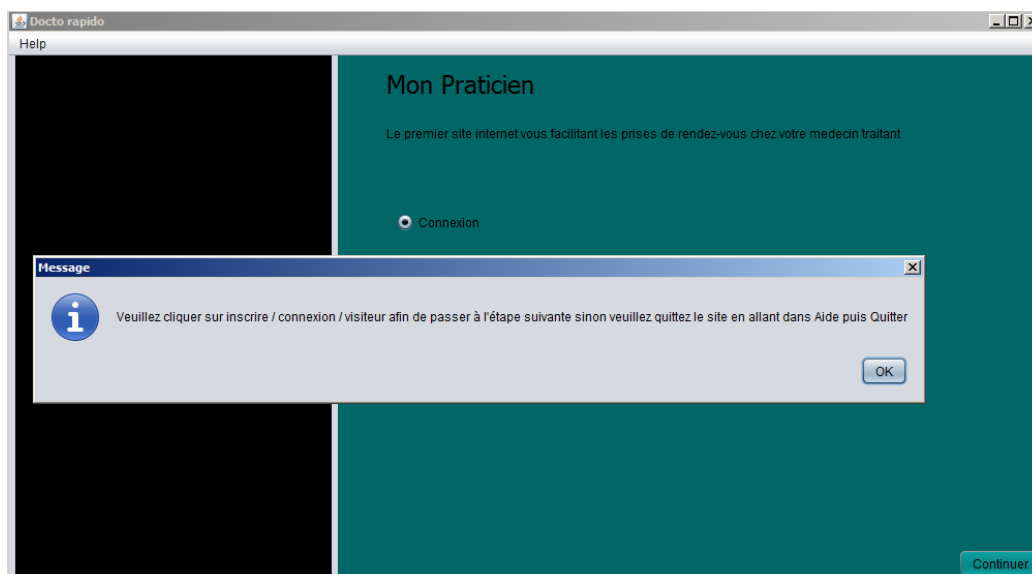
## Jeu d'essai

### ► Menu principal



Lorsqu'on lance le programme, une fenêtre invite l'utilisateur à choisir s'il souhaite se connecter, s'inscrire, ou s'il veut parcourir le site en tant que visiteur sans se connecter.

Il y a aussi un onglet « Help » : si on clique dessus, l'utilisateur a le choix entre quitter le programme et afficher un message lui expliquant la démarche à suivre pour utiliser le site et ce pour chacun des fenêtres existants :



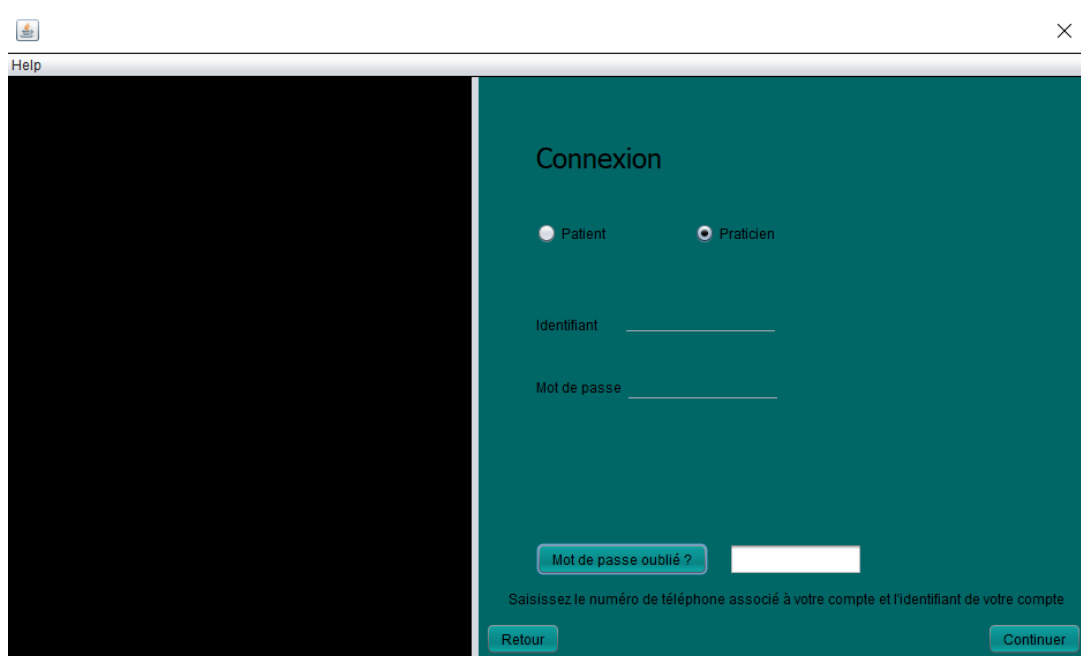
## ► Connexion

Sur cette fenêtre et sur les suivantes, le programme propose toujours à l'utilisateur de retourner à la fenêtre précédente si on clique sur le bouton **Retour**.

Si l'utilisateur souhaite se connecter, il doit d'abord cocher une case en fonction de s'il est patient ou praticien. Il doit ensuite entrer son identifiant et son mot de passe dans chacune des cases correspondantes.

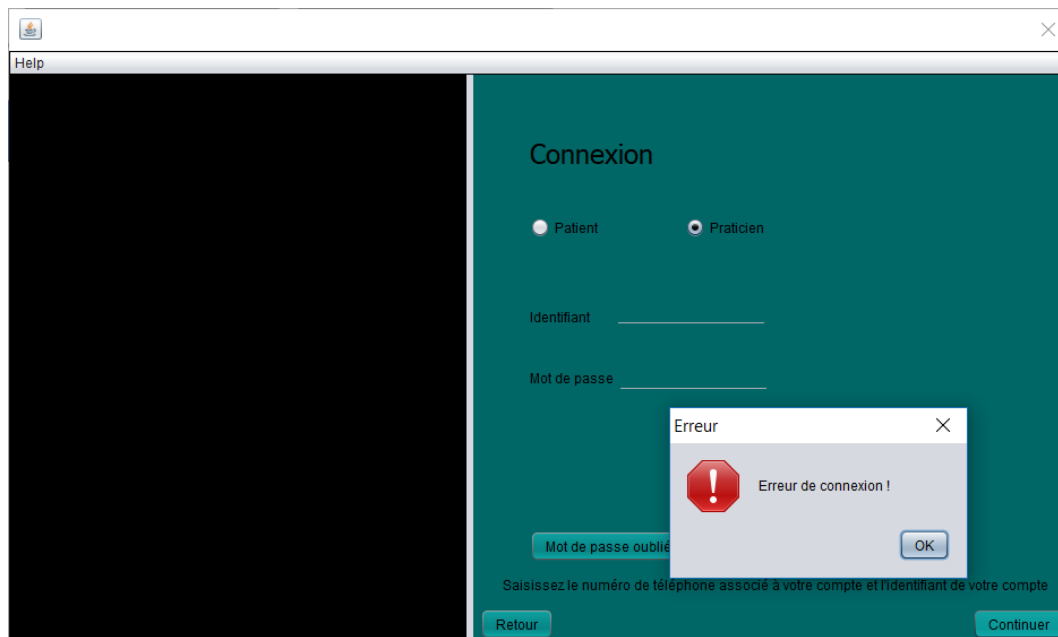
Une fois ces étapes réalisées, l'utilisateur appuie sur le bouton **Continuer**.

Voici un exemple de connexion avec un praticien qu'on gardera pour la suite (le procédé est identique pour un patient mais il ne peut pas proposer de rendez-vous comme le peut un praticien) :

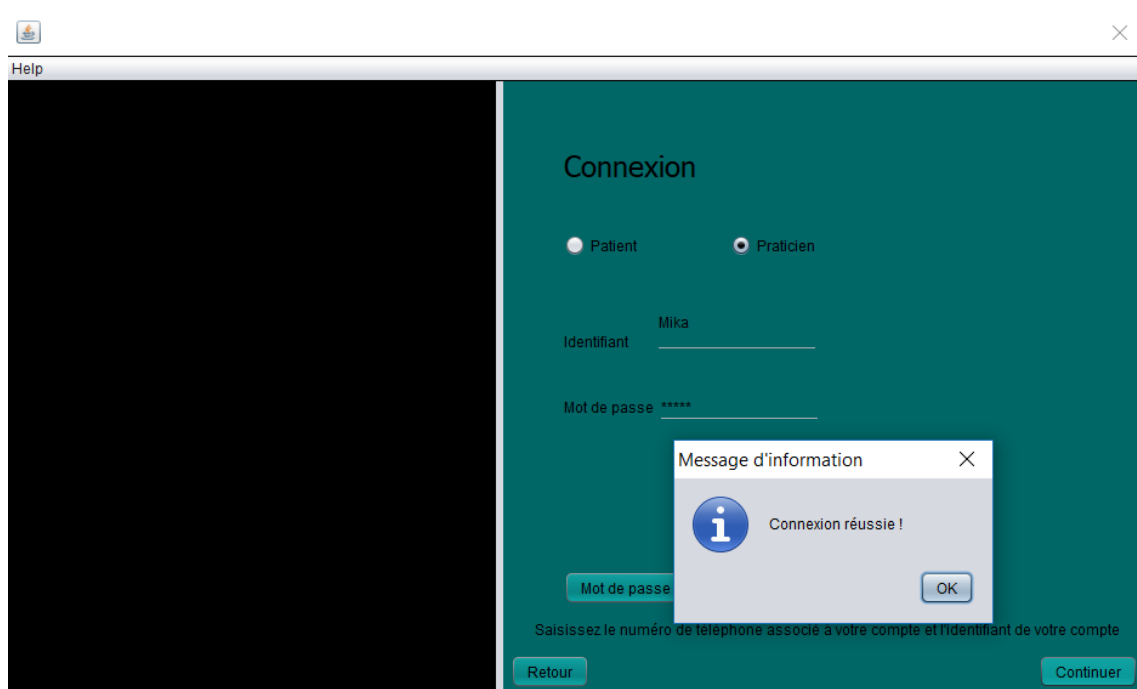


The screenshot shows a window titled 'Connexion' with a teal background. On the left is a black sidebar with a 'Help' button. The main area contains two radio buttons: 'Patient' (unselected) and 'Praticien' (selected). Below these are input fields for 'Identifiant' and 'Mot de passe'. A link 'Mot de passe oublié ?' is next to a small white input field. At the bottom, a text prompt says 'Saisissez le numéro de téléphone associé à votre compte et l'identifiant de votre compte'. Two buttons, 'Retour' and 'Continuer', are at the bottom right.

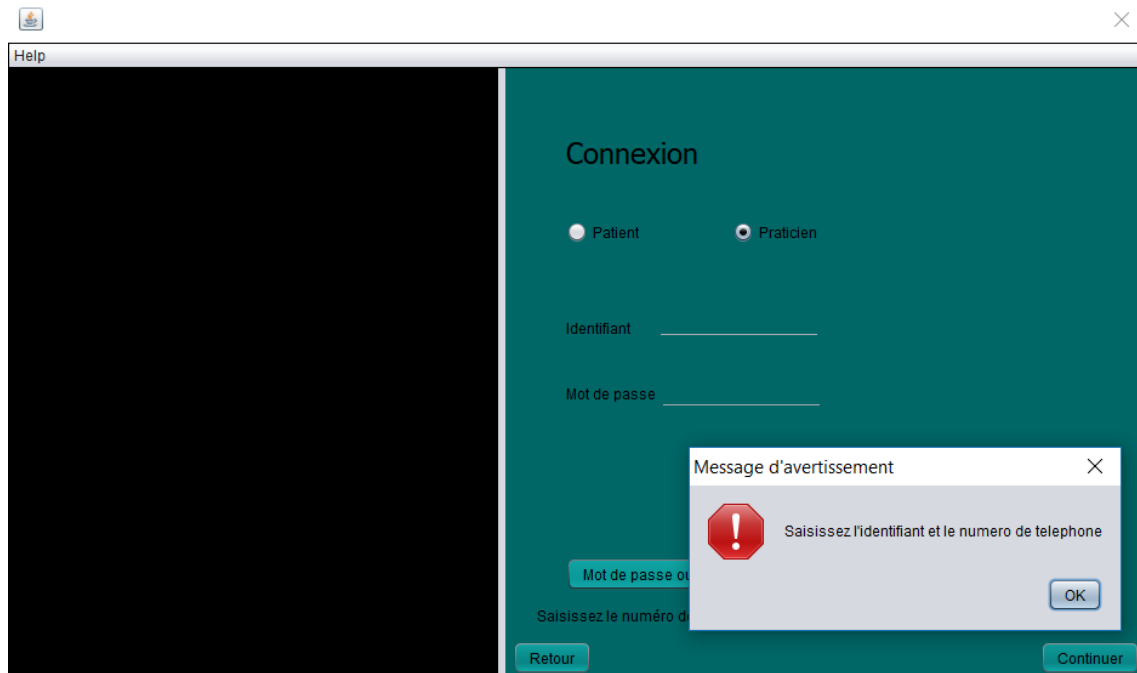
Bien entendu, l'utilisateur se connectant doit être référencé par le site internet, dans le cas contraire, une fenêtre de dialogue affichera un message informant de l'échec de la connexion. De même si l'utilisateur ne choisit pas entre patient et praticien :



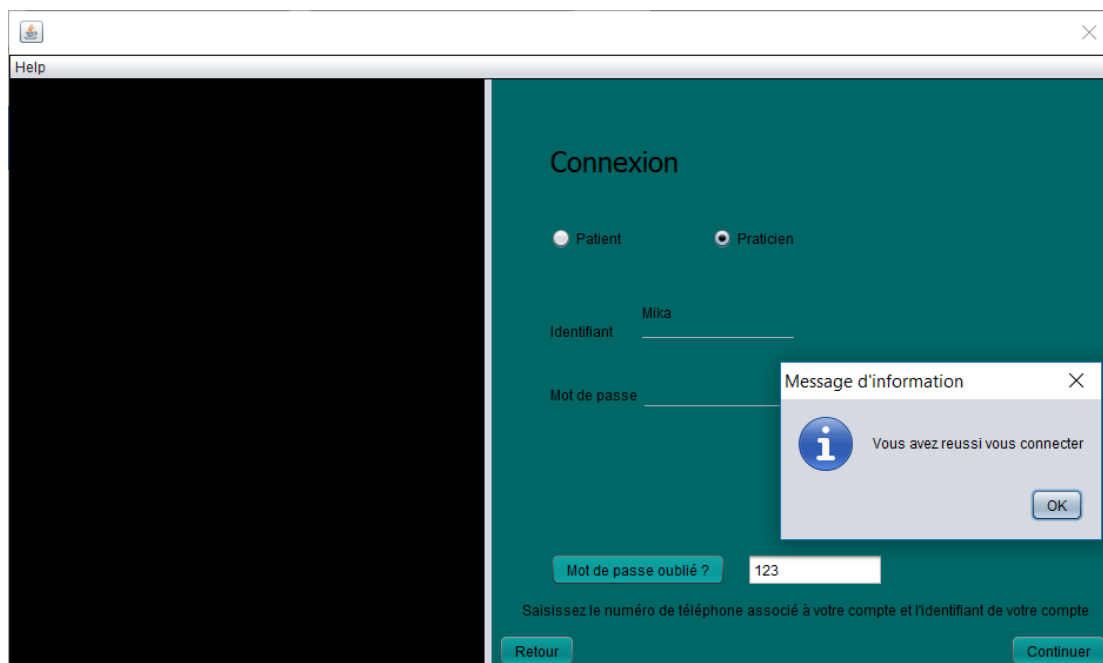
En revanche, si l'utilisateur est bien référencé par le site, alors une fenêtre confirmant la connexion de l'utilisateur s'affiche :



Une aide existe pour les utilisateurs ayant perdu leur mot de passe : En cliquant sur **Mot de passe oublié ?**, une fenêtre de dialogue s'affiche :

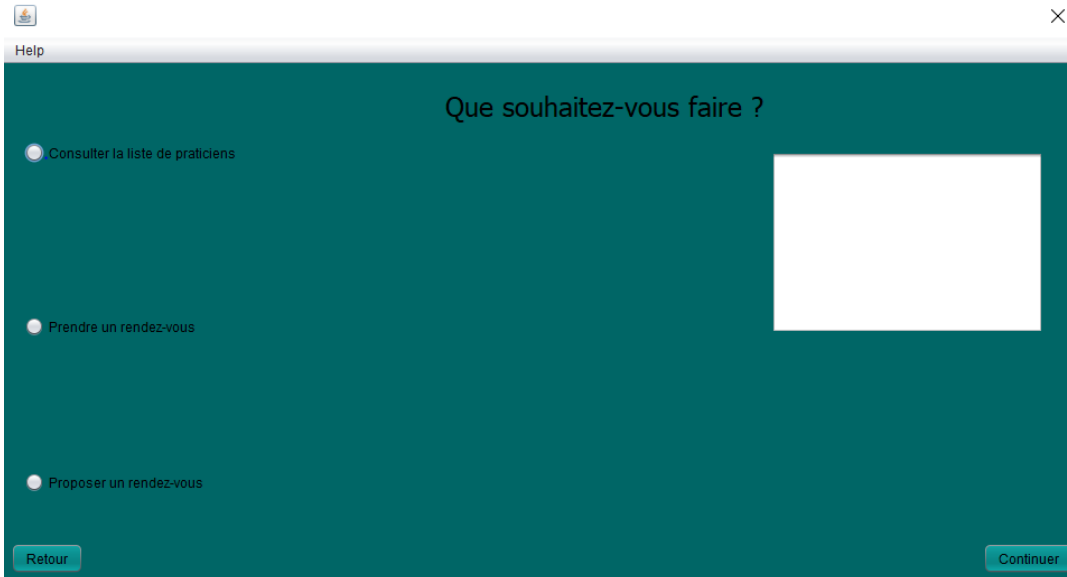


La fenêtre indique comment l'utilisateur peut se connecter : il doit rentrer son identifiant ainsi que son numéro de téléphone pour les praticiens ou son numéro de sécurité sociale pour les patients dans la zone de texte à droite de **Mot de passe oublié ?** On traitera, ici, le cas du praticien ; le cas du patient répondant à la même logique. Une fois ces instructions réalisées, il clique sur **Mot de passe oublié ?** et une fenêtre de dialogue s'affiche pour confirmer la connexion est réussie :




## ► Utilisation du site

Une fois connecté, l'utilisateur se voit proposer différentes fonctionnalités : il peut consulter la liste des praticiens recensés, prendre un rendez-vous et proposer un rendez-vous (cette dernière fonctionnalité est présente seulement chez les praticiens). Par ailleurs, si l'utilisateur est un visiteur, il ne sera capable que de consulter la liste de praticiens.



### 1- Consulter la liste des praticiens

Pour consulter la liste des praticiens, l'utilisateur a la possibilité de filtrer les praticiens selon la ville où ils exercent (comme c'est le cas ci-dessous), mais il peut aussi tous les afficher. Il doit choisir s'il veut consulter un praticien généraliste ou spécialiste.



Lorsque l'utilisateur appuie sur **Continuer**, s'il existe un praticien référencé correspondant à sa recherche, alors la liste des praticiens s'affiche dans le rectangle situé à droite :



Help

Que souhaitez-vous faire ?

☒ Consulter la liste de praticiens
 ☐ Affiner par ville
 ☐ Voir tous les praticiens

☒ Généraliste
 ☐ Spécialiste

Saisissez la ville

☐ Prendre un rendez-vous

☐ Proposer un rendez-vous

Retour Continuer

Dr.Dupont  
Dr.Eric

Voici le cas où l'utilisateur veut afficher tous les praticiens (généralistes) référencés par le site, sans filtrer par ville :



Help

Que souhaitez-vous faire ?

☐ Consulter la liste de praticiens
 ☐ Affiner par ville
 ☒ Voir tous les praticiens

☒ Généraliste
 ☐ Spécialiste

Saisissez la ville

☐ Prendre un rendez-vous

☐ Proposer un rendez-vous

Retour Continuer

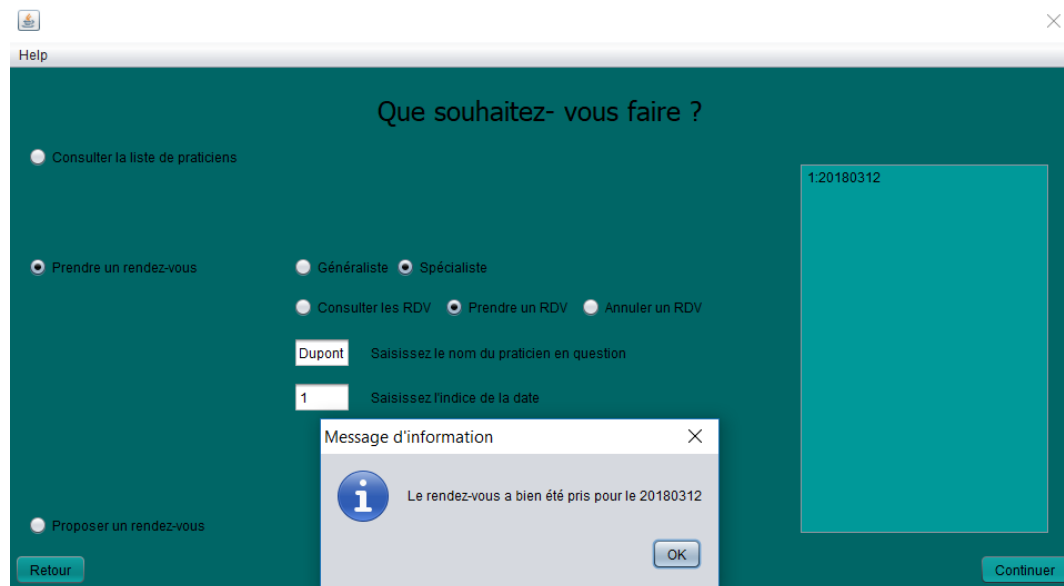
Dr.Dupont  
Dr.Eric  
Dr.Pierre

## 2- Prendre un rendez-vous

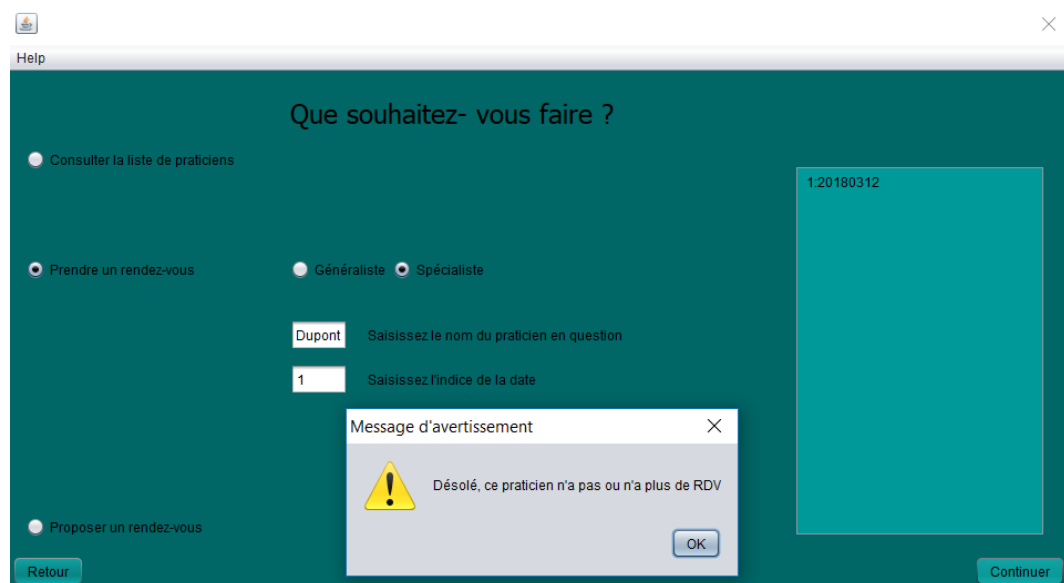
Si l'utilisateur veut prendre un rendez-vous, alors de même il clique sur la case correspondante. Il doit alors choisir s'il veut consulter un spécialiste ou un généraliste, puis rentrer le nom du praticien qu'il veut consulter. Dans le rectangle à droite, la liste



des dates disponibles avec ce praticien s'affiche et l'utilisateur rentre l'indice de la date à laquelle il veut consulter. Une fois que l'utilisateur clique sur **Continuer**, une fenêtre s'affiche lui confirmant que son rendez-vous est bien réservé :



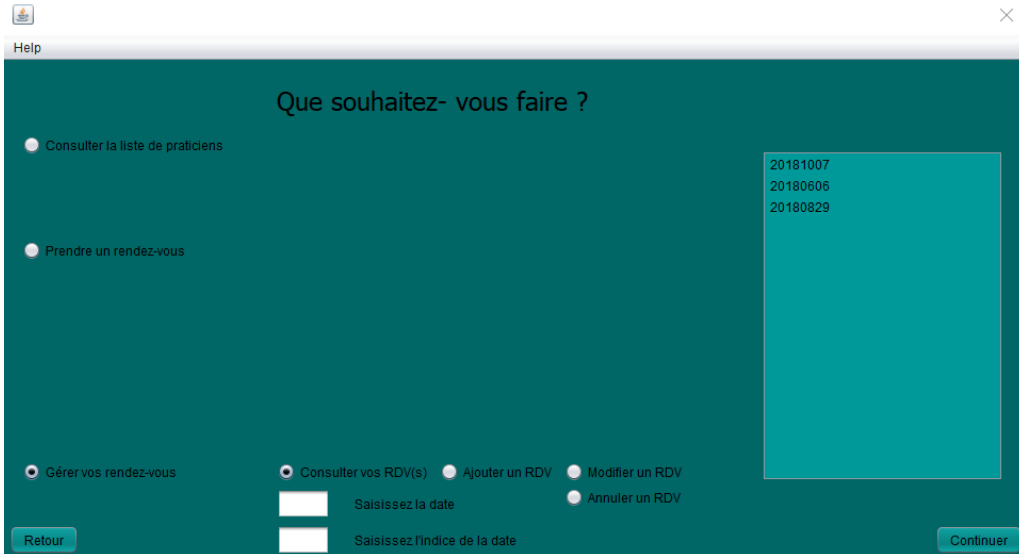
Il se peut que la date souhaitée soit déjà réservée. Dans ce cas, une fenêtre informe que le rendez-vous ne peut pas être pris à cette date :



### 3 – Gérer vos rendez-vous (pour les praticiens uniquement)

S'il est connecté en tant que praticien, l'utilisateur peut gérer ses rendez-vous : consulter sa liste de rendez-vous déjà enregistrés, ajouter une date de rendez-vous, modifier (reporter) une date de rendez-vous et annuler un rendez-vous.

· S'il souhaite consulter ses rendez-vous, l'utilisateur n'a qu'à cliquer sur la case **Consulter vos RDV(s)** puis sur le bouton **Continuer**. La liste des rendez-vous sauvegardés s'affiche alors dans le rectangle de droite.



Help

Que souhaitez-vous faire ?

☐ Consulter la liste de praticiens

☐ Prendre un rendez-vous

☒ Gérer vos rendez-vous

☒ Consulter vos RDV(s)

☐ Ajouter un RDV

☐ Modifier un RDV

☐ Annuler un RDV

Saisissez la date

Saisissez l'indice de la date

Retour

Continuer

20181007  
20180606  
20180829

· S'il souhaite ajouter un rendez-vous, l'utilisateur clique sur **Ajouter un RDV**. Il entre ensuite la date du rendez-vous (12/12/2018) pour que celui-ci soit enregistré. Une fenêtre de dialogue confirme que l'ajout a bien été effectué.



Help

Que souhaitez-vous faire ?

☐ Consulter la liste de praticiens

☐ Prendre un rendez-vous

☒ Gérer vos rendez-vous

☒ Consulter vos RDV(s)

☒ Ajouter un RDV

☐ Modifier un RDV

☐ Annuler un RDV

81212 Saisissez la date

4 Saisissez l'indice de la date

Retour

Message d'information

la date 20181212 a été ajoutée

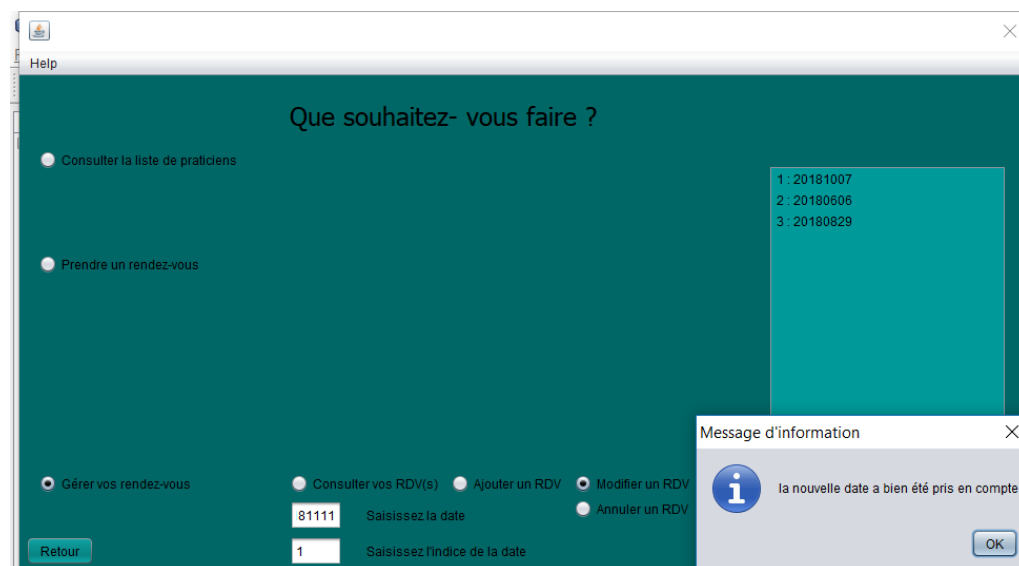
OK

20181007  
20180606  
20180829

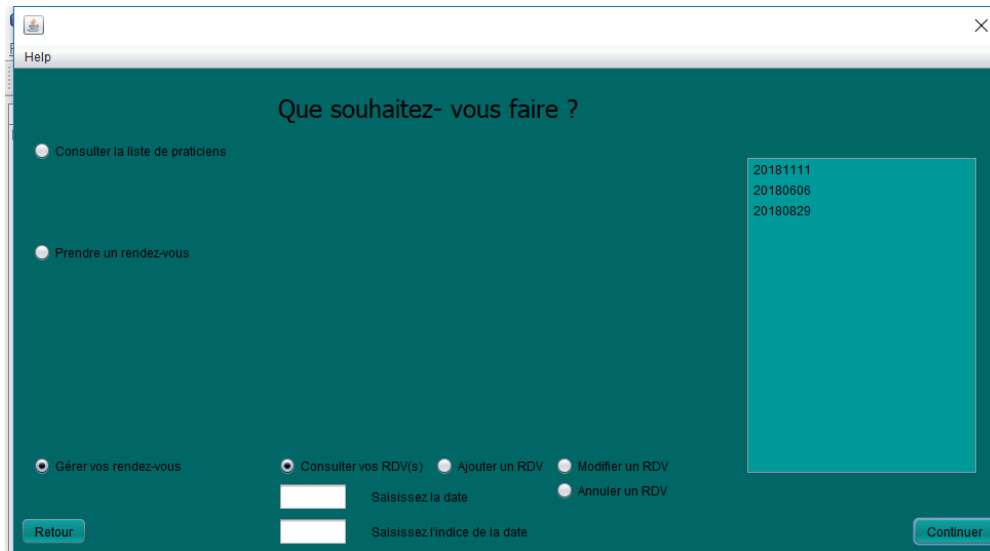
Lorsqu'on refait **Consulter**, le rendez-vous (12/12/2018) a bien été ajouté à la liste :



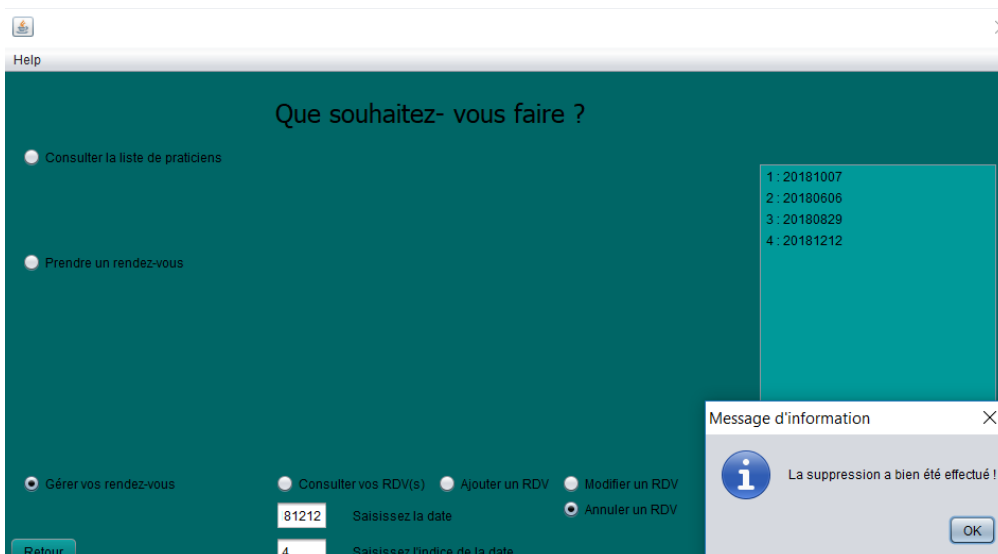
· S'il souhaite modifier un rendez-vous alors il lui suffira tout simplement de saisir l'indice de la date sur sa liste de rendez-vous puis de saisir la nouvelle date. Une fenêtre de dialogue confirme la modification survenue.



Dans cet exemple, on modifie la première date (indiquée donc à 1) par la nouvelle date, ici, 20181111, une fenêtre nous confirme que la modification est effective et lorsque l'on consulte à nouveau la liste de rendez-vous, on s'aperçoit bien que les modifications ont bien été prises en compte.



· S'il souhaite annuler un rendez-vous, alors l'utilisateur rentre la date en l'indice du rendez-vous qu'il veut supprimer (12/12/2018). Une fois qu'il a cliqué sur le bouton **Continuer**, une fenêtre de dialogue confirme que la suppression est effective :



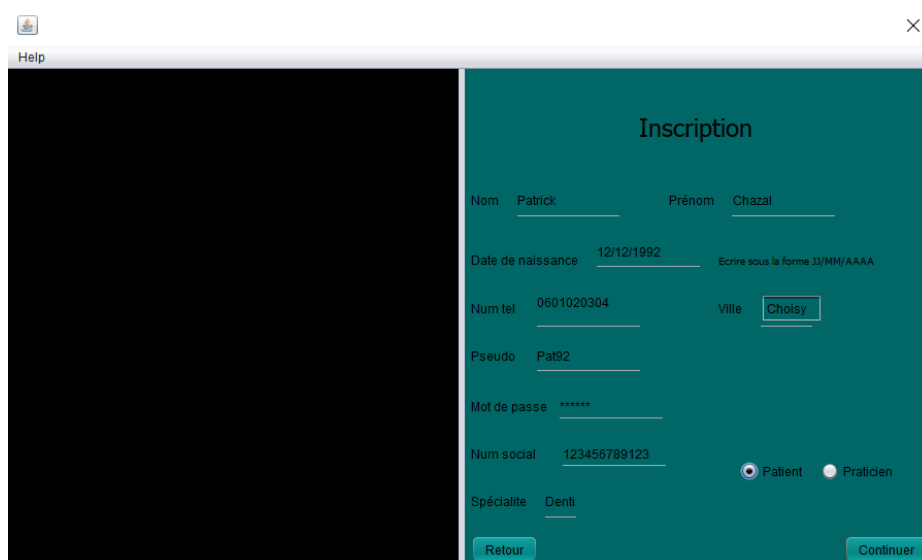
Lorsqu'on réaffiche la liste avec **Consulter vos RDV(s)**, la date du 12/12/2018 a bien été supprimée de la liste :



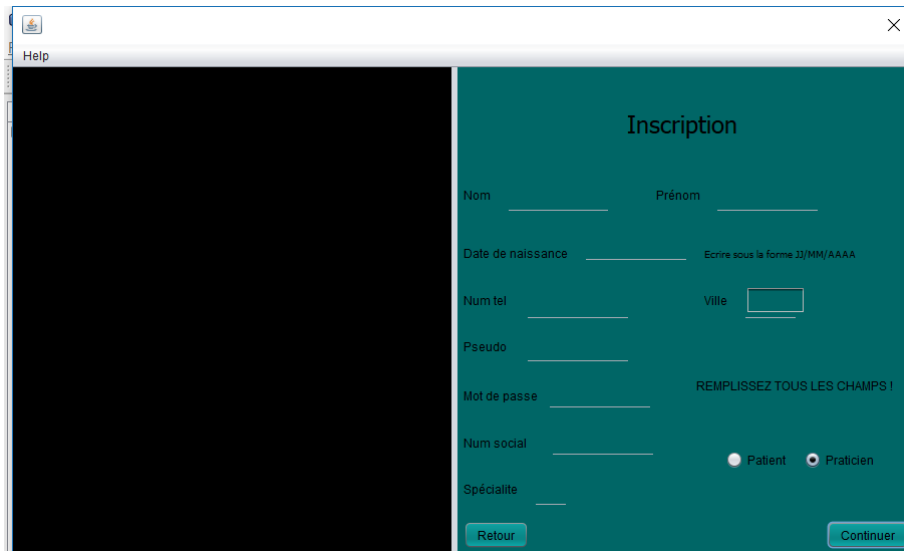
## ► Inscription

S'il souhaite s'inscrire, alors un formulaire d'inscription s'affiche en distinguant toujours le patient du praticien. Ici, on traitera l'exemple avec un patient mais le procédé est toujours identique pour le praticien à l'exception qu'il devra saisir une spécialité (généraliste ou spécialiste).

L'utilisateur doit remplir plusieurs informations afin d'être référencé sur le site :



Une fois que tout a été rempli, l'utilisateur appuie sur **Continuer** pour finaliser son inscription. Une fenêtre apparaît alors si l'inscription a bien été effectuée. S'il n'a pas rempli tous les champs, un JLabel apparaît et notifie l'utilisateur qu'il n'a pas saisi tous les champs.



→ Cependant, même si l'utilisateur s'inscrit correctement, l'interprétation graphique nous empêche de poursuivre sur le menu de sélection. En effet, la méthode utilisée pour l'inscription d'un utilisateur est fonctionnelle dans la partie console, mais pêche lorsque l'on combine à la partie graphique pour notre cas.

## ► Visiteur

S'il s'agit d'un simple visiteur (pas de compte), alors il ne peut que consulter la liste des praticiens déjà inscrits sur le site :



Il s'agit exactement des mêmes fonctionnalités que pour un utilisateur qui se connecte et accède à ce choix. Sauf que le visiteur ne dispose que de cette option.

## Conclusion

Lors de ce projet de Java de second semestre, nous avons pu renforcer nos acquis du premier semestre et apprendre de nouvelles compétences (implémentation d'une interface graphique). Ce sujet nous a intéressé car il nous a permis de mieux comprendre le fonctionnement des sites internet et plus particulièrement des sites de prise de rendez-vous comme *Doctolib.fr*.

Le fait d'avoir premièrement réalisé le « gros » du code avant l'implémentation graphique nous a permis de nous concentrer sur cette dernière qui nous était encore inconnue il y a quelques mois, bien que pas mal de changements ont dû être opérés entre les deux parties de notre projet. Par ailleurs, nous avons codé la partie console avec les saisies de l'utilisateur pour que notre compréhension du code soit plus simple et nous pensons que le fait d'avoir également traité les saisies de l'utilisateur nous a permis de traiter plus facilement la partie graphique. En effet, s'il y avait des exceptions ou des conditions à traiter lors des différentes saisies de l'utilisateur sur la partie graphique, nous les avons déjà traités en amont avec la partie console et c'était donc une partie en moins à traiter.

Comme lors du rendu du TP de l'agence de location de voiture, le site pourrait être amélioré de plusieurs façons pour qu'il soit réellement exploitable : pouvoir stocker plusieurs rendez-vous par jour en les classant par heure et minute, et en particulier l'utilisation d'un fichier dont la logique a été saisie mais dont nous n'avons pas réussi à transposer cette logique dans notre code.