

```
1 using System;
2 using System.Collections.Generic;
3 using System.ComponentModel;
4 using System.Data;
5 using System.Data.SQLite;
6 using System.Drawing;
7 using System.Linq;
8 using System.Text;
9 using System.Threading.Tasks;
10 using System.Windows.Forms;
11
12 namespace PimPomBro
13 {
14     public partial class gestionPompiers : Form
15     {
16         // mettre true pour bypass le test de login (login = "vrichard",
17         mdp = "mdpVero")
18         bool admin = false;
19         private int matricule = -1;
20         string requete = "";
21         SQLiteCommand cmd = null;
22         SQLiteDataReader reader = null;
23
24         public gestionPompiers()
25         {
26             InitializeComponent();
27         }
28
29         private void gestionPompiers_Load(object sender, EventArgs e)
30         {
31             picLogo.Image = Image.FromFile(@"../Images/logo.png");
32             picLogo.SizeMode = PictureBoxSizeMode.Zoom;
33             picNouveau.Image = Image.FromFile(@"../Images/jsp.jpg");
34
35             cboCaserne.DisplayMember = "nom";
36             cboCaserne.ValueMember = "id";
37             cboCaserne.DataSource = MesDatas.DsGlobal.Tables["Caserne"];
38             // cboCaserne.SelectedIndex = -1;
39
40             cboCaserneDeRattachement.DisplayMember = "nom";
41             cboCaserneDeRattachement.ValueMember = "id";
42             cboCaserneDeRattachement.DataSource =
43             MesDatas.DsGlobal.Tables["Caserne"].Copy();
44
45             cboGrade.DisplayMember = "libelle";
46             cboGrade.ValueMember = "code";
47             cboGrade.DataSource = MesDatas.DsGlobal.Tables["Grade"];
48
49             cboPompier.Text = "choisir pompier";
50             cboPompier.SelectedIndex = -1;
51         }
52     }
53 }
```

```
52     private void cboPompier_Click(object sender, EventArgs e)
53     {
54         if (cboCaserne.SelectedItem == null)
55         {
56             MessageBox.Show("Veuillez selectionner une caserne pour
pouvoir choisir un pompier.");
57         }
58     }
59
60     private void cboCaserne_SelectedIndexChanged(object sender,
EventArgs e)
61     {
62         lblConsigne.Text = "Veuillez selectionner un pompier.";
63         cboPompier.Text = "choisir pompier";
64
65         if (cboCaserne.SelectedItem == null) { return; }
66         int idCaserne = Convert.ToInt32(cboCaserne.SelectedValue);
67
68         DataTable tablePompier = MesDats.DsGlobal.Tables["Pompier"];
69         DataTable tableAffectation =
MesDats.DsGlobal.Tables["Affectation"];
70
71         List<int> listeMatricules = new List<int>();
72         foreach (DataRow rowAff in tableAffectation.Rows)
73         {
74             if (rowAff["DateFin"].ToString() == "" &&
Convert.ToInt32(rowAff["idCaserne"]) == idCaserne)
75             {
76                 listeMatricules.Add(Convert.ToInt32(rowAff["matriculePompier"]));
77             }
78         }
79
80         string filter = "matricule IN (" + string.Join(",",
listeMatricules) + ")";
81         DataView dv = new DataView(tablePompier, filter, "nom ASC,
prenom ASC", DataViewRowState.CurrentRows);
82
83         cboPompier.ValueMember = "matricule";
84         cboPompier.DataSource = dv;
85
86         cboPompier.Format += (s, e2) =>
87         {
88             DataRowView drv = e2.ListItem as DataRowView;
89             if (drv != null)
90             {
91                 e2.Value = drv["nom"] + " " + drv["prenom"];
92             }
93         };
94
95         pnlPompier.Visible = false;
96         //cboPompier.Text = "";
97         cboPompier.SelectedIndex = -1;
98         cboPompier.Select();
99     }
```

```
100
101     private void cboPompier_SelectedIndexChanged(object sender,
EventArgs e)
102     {
103         if (cboPompier.SelectedItem == null ||
cboPompier.SelectedIndex == -1 || cboPompier.Text == "")
104         {
105             // MessageBox.Show("null pompier");
106             return;
107         }
108         // MessageBox.Show("good pompier");
109
110         pnlInformationsDetaillees.Visible = false;
111         cboGrade.Enabled = false;
112
113
114         matricule = Convert.ToInt32(cboPompier.SelectedValue);
115         DataTable tablePompier = MesDatas.DsGlobal.Tables["Pompier"];
116         DataRow rowPompier = tablePompier.Rows.Find(matricule);
117
118         lblMatricule.Text = matricule.ToString();
119         lblNom.Text = rowPompier["nom"].ToString();
120         lblPrenom.Text = rowPompier["prenom"].ToString();
121         lblSexe.Text = rowPompier["sexe"].ToString().ToUpper();
122         lblDateEmbauche.Text = rowPompier["dateEmbauche"].ToString();
123         txtGrade.Text = rowPompier["codeGrade"].ToString();
124         if (rowPompier["type"].ToString() == "p") {
rdbProfessionnel.Checked = true; } else { rdbVolontaire.Checked = true; }
125         try
126         {
127             lblTelephone.Text = rowPompier["portable"].ToString();
128             lblBip.Text = rowPompier["bip"].ToString();
129         }
130         catch (Exception ex) { }
131
132         try
133         {
134             requete = "SELECT libelle FROM Grade WHERE code =
@codeGrade";
135             cmd = new SQLiteCommand(requete, Connexion.Connec);
136             cmd.Parameters.AddWithValue("@codeGrade", txtGrade.Text);
137             reader = cmd.ExecuteReader();
138             if (reader.Read())
139             {
140                 cboGrade.Text = reader["libelle"].ToString();
141             }
142         } catch (Exception ex) {
143             cboGrade.Visible = false;
144         }
145
146         pnlPompier.Visible = true;
147     }
148     private void btnModifications_Click(object sender, EventArgs e)
149     {
150         if (!admin) { connexionAdmin(); }
```

```
151         if (!admin) { return; }
152
153         cboCaserneDeRattachement.Text = cboCaserne.Text;
154
155
156         lstHabilitations.Items.Clear();
157         requete = "SELECT h.libelle,p.dateObtention FROM Passer p JOIN
Habilitation h ON p.idHabilitation = h.id WHERE p.matriculePompier =
@matricule";
158         cmd = new SQLiteCommand(requete, Connexion.Connec);
159         cmd.Parameters.AddWithValue("@matricule", matricule);
160         reader = cmd.ExecuteReader();
161         while (reader.Read()) {
162             DateTime date =
DateTime.Parse(reader["dateObtention"].ToString());
163             lstHabilitations.Items.Add(date.ToString("dd/MM/yyyy") + "
- " + reader["libelle"]);
164         }
165
166         lstAffectations.Items.Clear();
167         requete = "SELECT a.dateA,a.dateFin,c.nom FROM Affectation a
JOIN Caserne c ON a.idCaserne = c.id WHERE a.matriculePompier = @matricule
AND a.dateFin IS NOT NULL";
168         cmd = new SQLiteCommand(requete, Connexion.Connec);
169         cmd.Parameters.AddWithValue("@matricule", matricule);
170         reader = cmd.ExecuteReader();
171         while (reader.Read())
172         {
173             string caserne = reader["nom"].ToString();
174             DateTime dA = DateTime.Parse(reader["dateA"].ToString());
175             string dateA = dA.ToString("dd/MM/yyyy");
176             DateTime dF =
DateTime.Parse(reader["dateFin"].ToString());
177             string dateFin = dF.ToString("dd/MM/yyyy");
178             lstAffectations.Items.Add(caserne + " : " + dateA + " - "
+ dateFin);
179         }
180
181
182         requete = "SELECT enConge FROM Pompier WHERE matricule =
@matricule";
183         cmd = new SQLiteCommand(requete, Connexion.Connec);
184         cmd.Parameters.AddWithValue("@matricule", matricule);
185         reader = cmd.ExecuteReader();
186         reader.Read();
187         //
MessageBox.Show(Convert.ToInt32(reader["enConge"]).ToString());
188         if (Convert.ToInt32(reader["enConge"]) == 1)
189         {
190             chkConge.Checked = true;
191         } else
192         {
193             chkConge.Checked = false;
194         }
195
```

```
196         reader.Close();
197         cboGrade.Enabled = true;
198         pnlInformationsDetaillees.Visible = true;
199     }
200
201     private void btnAppliquerModif_Click(object sender, EventArgs e)
202     {
203         using (SQLiteTransaction tran =
204         Connexion.Connec.BeginTransaction())
205         {
206             try
207             {
208                 requete = "UPDATE Pompier SET codeGrade = @codeGrade,
209                 enConge = @enConge WHERE matricule = @matricule";
210                 cmd = new SQLiteCommand(requete, Connexion.Connec,
211                 tran);
212                 cmd.Parameters.AddWithValue("@codeGrade",
213                 txtGrade.Text);
214                 cmd.Parameters.AddWithValue("@matricule", matricule);
215                 cmd.Parameters.AddWithValue("@enConge",
216                 (chkConge.Checked) ? 1 : 0);
217                 cmd.ExecuteNonQuery();
218
219                 requete = "UPDATE Affectation SET dateFin = @dateFin
220                 WHERE matriculePompier = @matricule AND (dateFin IS NULL or dateFin =
221                 '')";
222                 cmd = new SQLiteCommand(requete, Connexion.Connec,
223                 tran);
224                 cmd.Parameters.AddWithValue("@matricule", matricule);
225                 cmd.Parameters.AddWithValue("@dateFin", DateTime.Now);
226                 cmd.ExecuteNonQuery();
227
228                 requete = "INSERT INTO Affectation VALUES (@matricule,
229                 @dateA, NULL,@caserne)";
230                 cmd = new SQLiteCommand(requete, Connexion.Connec,
231                 tran);
232                 cmd.Parameters.AddWithValue("@matricule", matricule);
233                 cmd.Parameters.AddWithValue("@dateA", DateTime.Now);
234                 cmd.Parameters.AddWithValue("@caserne",
235                 cboCaserneDeRattachement.Selected.Value);
236                 cmd.ExecuteNonQuery();
237
238                 tran.Commit();
239             }
240             catch (Exception ex)
241             {
242                 tran.Rollback();
243                 MessageBox.Show(ex.Message);
244                 return;
245             }
246         }
247         MesDatas.refreshTable("Pompier");
248         MesDatas.refreshTable("Affectation");
249
250         this.gestionPompiers_Load(sender, e);
251     }
```

```
240         this.cboCaserne_SelectedIndexChanged(sender, e);
241     }
242
243     private void cboGrade_SelectedIndexChanged(object sender,
EventArgs e)
244     {
245         requete = "SELECT code FROM Grade WHERE libelle = @libelle";
246         cmd = new SQLiteCommand(requete, Connexion.Connec);
247         cmd.Parameters.AddWithValue("@libelle", cboGrade.Text);
248         reader = cmd.ExecuteReader();
249         if (reader.Read())
250         {
251             txtGrade.Text = reader["code"].ToString();
252         }
253
254         reader.Close();
255     }
256     private void connexionAdmin()
257     {
258         ConnexionAdmin connexionAdmin = new ConnexionAdmin();
259         var result = connexionAdmin.ShowDialog();
260         if (result == DialogResult.OK) {
261             admin = true;
262         } else
263         {
264             admin = false;
265         }
266     }
267
268
269     private void btnNouveau_Click(object sender, EventArgs e)
270     {
271         // si l'utilisateur n'est pas admin on lui demande de se
connecter
272         if (!admin) { connexionAdmin(); }
273         // s'il n'est toujours pas connecté on quitte la fonction
274         if (!admin) { return; }
275
276         CreationPompier creationPompier = new CreationPompier();
277         creationPompier.ShowDialog();
278
279         this.gestionPompiers_Load(sender, e);
280         this.cboCaserne_SelectedIndexChanged(sender, e);
281     }
282
283     private void btnAnnuler_Click(object sender, EventArgs e)
284     {
285         cboPompier_SelectedIndexChanged(sender, e);
286         btnModifications_Click(sender, e);
287     }
288
289     private void button1_Click(object sender, EventArgs e)
290     {
291         this.Close();
292     }
```

```
293     }  
294 }  
295
```