

```
1 using System;
2 using System.Collections.Generic;
3 using System.ComponentModel;
4 using System.Data;
5 using System.Drawing;
6 using System.Linq;
7 using System.Text;
8 using System.Threading.Tasks;
9 using System.Windows.Forms;
10
11 namespace ADA1
12 {
13     public partial class Accueil : Form
14     {
15
16         public Accueil()
17         {
18             InitializeComponent();
19         }
20
21         private void Accueil_Load(object sender, EventArgs e)
22         {
23             // Paramètre IsMdiContainer propriété sur vrai.
24             IsMdiContainer = true;
25             if (tabformulaire.SelectedIndex == 0)
26             {
27                 frmmodif newFrm = new frmmodif
28                 {
29                     MdiParent = this,
30                     // définit le parent du formulaire appeler sur un onglet du ↗
31                     Parent = tabformulaire.TabPages[0]
32                 };
33                 newFrm.Show();
34             }
35
36             lblbonjour.Text = "Bienvenue personnel de l'agence " + ↗
37                 varGlobale.monagence.getnom();
38
39
40         }
41
42
43         private void tabformulaire_SelectedIndexChanged(object sender, ↗
44             EventArgs e)
45         {
46             if (tabformulaire.SelectedIndex == 0)
47             {
48                 frmmodif newFrm = new frmmodif
49                 {
50                     MdiParent = this,
51                     Parent = tabformulaire.TabPages[0]
52                 };
53                 newFrm.Show();
```

```
54     }
55
56     if (tabformulaire.SelectedIndex == 1)
57     {
58         vehiculeagence newFrm = new vehiculeagence
59         {
60             MdiParent = this,
61             Parent = tabformulaire.TabPages[1]
62         };
63         newFrm.Show();
64     }
65
66     if (tabformulaire.SelectedIndex == 2)
67     {
68         frmcontrat newFrm = new frmcontrat
69         {
70             MdiParent = this,
71             Parent = tabformulaire.TabPages[2]
72         };
73         newFrm.Show();
74     }
75 }
76
77 //Déconnecte l'utilisateur et le ramène à l'écran de connection
78 private void btndéco_Click(object sender, EventArgs e)
79 {
80
81     Frmconnecter frmco = new Frmconnecter();
82     frmco.Show();
83     this.Close();
84 }
85 }
86 }
87
```

```
1 using System;
2 using System.Collections.Generic;
3 using System.ComponentModel;
4 using System.Data;
5 using System.Drawing;
6 using System.Linq;
7 using System.Text;
8 using System.Threading.Tasks;
9 using System.Windows.Forms;
10 using System.Data.Odbc;
11 using System.Data.SqlClient;
12
13 namespace ADA1
14 {
15     public partial class Frmconnector : Form
16     {
17         public Frmconnector()
18         {
19             InitializeComponent();
20         }
21
22         private void btnconnector_Click(object sender, EventArgs e)
23         {
24
25             //permet de vérifier si la valeur saisi est bien un entié
26             int id = 0;
27             if(Int32.TryParse(txtid.Text,out id )==false)
28             {
29                 MessageBox.Show("Veuillez écrire un nombre pour l'ID");
30                 txtid.Text = "";
31                 txtmdp.Text = "";
32             }
33             else
34             {
35                 //permet de vérifier si le compte existe sinon un message d'erreur s'affiche
36                 try
37                 {
38                     varGlobale.monagence = PasserelleAgence.connecter
39                     (Convert.ToInt32(txtid.Text), txtmdp.Text);
40                 }
41                 catch (Exception exception)
42                 {
43                     Console.WriteLine(exception.Message);
44                 }
45                 if (varGlobale.monagence != null)
46                 {
47                     txtid.Text = "";
48                     txtmdp.Text = "";
49                 }
50
51                 Accueil frma = new Accueil();
52                 frma.Show();
53                 this.Hide();
54             }
55         }
56     }
57 }
```

```
55         }
56     }
57
58     else
59     {
60         MessageBox.Show("Erreur de connexion");
61         txtid.Text = "";
62         txtmdp.Text = "";
63     }
64
65
66 }
67
68 }
69
70
71 private void Frmconnecter_Load(object sender, EventArgs e)
72 {
73     //connexion à la base de données
74     varglob.strconnect = System.IO.File.ReadAllText      ↗
75     (Application.StartupPath + "\\connexion.txt");
76     //permet de se connecter à l'agence de Mâcon (bien sûr ses lignes  ↗
77     de code devront être supprimés en cas d'application réelle
78     txtid.Text = "1";
79     txtmdp.Text = "Mot2pa$$";
80 }
81 //Ferme l'application
82 private void Frmconnecter_FormClosed(object sender, FormClosedEventArgs ↗
83     e)
84 {
85     Application.Exit();
86 }
87 }
```

```
1 using System;
2 using System.Collections.Generic;
3 using System.ComponentModel;
4 using System.Data;
5 using System.Drawing;
6 using System.Linq;
7 using System.Text;
8 using System.Threading.Tasks;
9 using System.Windows.Forms;
10
11 namespace ADA1
12 {
13     public partial class frmcontrat : Form
14     {
15         List<vehicule> lesV = new List<vehicule>();
16         public frmcontrat()
17         {
18             InitializeComponent();
19         }
20
21         //Définit les différents éléments afficher lors de l'ouverture du form
22         private void frmcontrat_Load(object sender, EventArgs e)
23         {
24
25             visible(false);
26
27             btnmodifier.Visible = false;
28             btncreation.Visible = false;
29             btnfermer.Visible = false;
30             btnverifclient.Visible = false;
31             lblexiste.Visible = false;
32             cboimmat.Visible = false;
33             // chargement des client dans une liste afin qu'elle soit réutilisable
34             foreach (var item in PasserelleClient.ChargerClients())
35             {
36                 varGlobale.monagence.getlesclient().Add(item);
37             }
38
39         }
40
41         //permet de faire une selection d'élément afficher à l'écran
42         private void visible(bool param)
43         {
44
45             txtEDL.Visible = param;
46             txtimmat.Visible = param;
47             txtmarque.Visible = param;
48             txtmodele.Visible = param;
49             txtnom.Visible = param;
50             txtprenom.Visible = param;
51             txtkm.Visible = param;
52             txtmontant.Visible = param;
53             lblclient.Visible = param;
54             lblvehic.Visible = param;
55             lblkmeffectue.Visible = param;
```

```
56         lblheuredeb.Visible = param;
57         lblheurefin.Visible = param;
58         lbldatesign.Visible = param;
59         lblmontant.Visible = param;
60         lbldatefacture.Visible = param;
61         dtpheuredeb.Visible = param;
62         dtpheurefin.Visible = param;
63         dtpfacture.Visible = param;
64         dtpsignature.Visible = param;
65
66
67     }
68
69     //permet de selectionner les element en lecture seul ou non
70     private void lecture(bool param)
71     {
72         txtnom.ReadOnly = param;
73         txtmarque.ReadOnly = param;
74         txtmodele.ReadOnly = param;
75         txtprenom.ReadOnly = param;
76         txtimmat.ReadOnly = param;
77
78     }
79
80     //affichage du formulaire de création avec tous les paramètre utile au ↗
81     //différent combobox, textbox et label
82     private void btnnouv_Click(object sender, EventArgs e)
83     {
84         lecture(false);
85         dtpheuredeb.Enabled = true;
86         dtpsignature.Enabled = true;
87
88         visible(false);
89         txtnom.Visible = true;
90         txtprenom.Visible = true;
91         lblexiste.Visible = false;
92         lblclient.Visible = true;
93         btnfermer.Visible = true;
94         btnnouv.Visible = false;
95         btnmodifier.Visible = false;
96         btnverifclient.Visible = true;
97         cbocontrat.SelectedIndex = -1;
98         txtimmat.Visible = false;
99         cboimmat.Visible = false;
100
101         txtEDL.Text = "";
102         cboimmat.SelectedIndex = -1;
103         txtmarque.Text = "";
104         txtmodele.Text = "";
105         txtnom.Text = "";
106         txtprenom.Text = "";
107         txtkm.Text = "";
108         txtkm.ReadOnly = true;
109         dtpheurefin.Enabled = false;
110         txtmontant.Text = "";
```

```
111         txtmontant.ReadOnly = true;
112         dtpfacture.Enabled = false;
113
114     }
115
116     //cache le formulaire de création
117     private void btnfermer_Click(object sender, EventArgs e)
118     {
119         //permet de définir ce qui est modifiable ou non et la valeur des ↗
120         //   textbox et de gérer ce qui est visible ou non
121         visible(false);
122         lecture(true);
123         dtpheuredeb.Enabled = false;
124         dtpsignature.Enabled = false;
125         btncreation.Visible = false;
126         btnfermer.Visible = false;
127         btnnouv.Visible = true;
128         btnverifclient.Visible = false;
129         lbllexiste.Visible = false;
130         cboimmat.Visible = false;
131         dtpfacture.Enabled = true;
132         txtmontant.ReadOnly = false;
133         txtkm.ReadOnly = false;
134         dtpheurefin.Enabled = true;
135     }
136
137     //Permet de remplir automatiquement toutes les valeur d'un contrats ↗
138     //   dans les zones dédiées
139     private void cbocontrat_SelectedIndexChanged(object sender, EventArgs ↗
140     e)
141     {
142         if(cbocontrat.SelectedIndex!=-1)
143         {
144             //permet de définir ce qui est modifiable ou non et la valeur ↗
145             //   des textbox et de gérer ce qui est visible ou non
146             dtpfacture.Enabled = true;
147             dtpfacture.Enabled = true;
148             txtmontant.ReadOnly = false;
149             txtkm.ReadOnly = false;
150             dtpheurefin.Enabled = true;
151             dtpheurefin.Enabled = true;
152             cboimmat.Items.Clear();
153             btncreation.Visible = false;
154             btnfermer.Visible = false;
155             btnnouv.Visible = true;
156             visible(true);
157             lecture(true);
158             dtpheuredeb.Enabled = false;
159             dtpsignature.Enabled = false;
160             btnmodifier.Visible = true;
161             btncreation.Visible = false;
162             btnverifclient.Visible = false;
163             lbllexiste.Visible = false;
164
165             //remplissage des zones de text par leur valeur
```

```

C:\Users\Admin\Desktop\Projet ADA\ADA1\ADA1\frmcontrat.cs 4
163 txtnom.Text = varGlobale.monagence.getlescontrats() ↗
    [cbocontrat.SelectedIndex].GetClient().getnom().ToString();
164 txtprenom.Text = varGlobale.monagence.getlescontrats() ↗
    [cbocontrat.SelectedIndex].GetClient().getprenom().ToString() ↗
    ();
165 txtmarque.Text = varGlobale.monagence.getlescontrats() ↗
    [cbocontrat.SelectedIndex].GetVehicule().getmarque ↗
    ().ToString();
166 txtimmat.Text=varGlobale.monagence.getlescontrats() ↗
    [cbocontrat.SelectedIndex].GetVehicule().getimmat().ToString ↗
    ();
167 txtmodele.Text = varGlobale.monagence.getlescontrats() ↗
    [cbocontrat.SelectedIndex].GetVehicule().getmodele ↗
    ().ToString();
168 txtEDL.Text = varGlobale.monagence.getlescontrats() ↗
    [cbocontrat.SelectedIndex].GetVehicule().getetatdeslieux ↗
    ().ToString();
169 dtpheuredeb.Value= varGlobale.monagence.getlescontrats() ↗
    [cbocontrat.SelectedIndex].getdateheuredeb();
170 dtpheurefin.Value = varGlobale.monagence.getlescontrats() ↗
    [cbocontrat.SelectedIndex].getdateheurefin();
171 dtpsignature.Value = varGlobale.monagence.getlescontrats() ↗
    [cbocontrat.SelectedIndex].getdatesignature();
172 dtpfacture.Value = varGlobale.monagence.getlescontrats() ↗
    [cbocontrat.SelectedIndex].getdatefacture();
173 txtkm.Text = varGlobale.monagence.getlescontrats() ↗
    [cbocontrat.SelectedIndex].getnbkmeffectue().ToString();
174 txtmontant.Text = varGlobale.monagence.getlescontrats() ↗
    [cbocontrat.SelectedIndex].getmontantfacture().ToString();
175
176
177
178     }
179
180 }
181
182 //permet de rechercher un contrat à la date selectionner et de remplir ↗
    le combobox des contrats
183 private void btnrecherche_Click(object sender, EventArgs e)
184 {
185     // me permet de vider la liste de contrat d'une journée à l'autre
186     btnfermer_Click(sender, e);
187     btnmodifier.Visible = false;
188     cbocontrat.SelectedIndex = -1;
189     varGlobale.monagence.getlescontrats().Clear();
190     //permet de définir ce qui est modifiable ou non et la valeur des ↗
        textbox
191     dtpfacture.Enabled = true;
192     txtmontant.ReadOnly = false;
193     txtkm.ReadOnly = false;
194     dtpheurefin.Enabled = true;
195     cbocontrat.Items.Clear();
196     lecture(true);
197     dtpheuredeb.Enabled = false;
198     dtpsignature.Enabled = false;
199     try

```



```
200     {
201         //Charge tous les contrats existant pour une date selectionnée
202         foreach (var unC in PasserelleContrat.chargerlescontrats ↗
                (dtpdebut.Value, varGlobale.monagence))
203         {
204             varGlobale.monagence.getlescontrats().Add(unC);
205             cbocontrat.Items.Add(unC.getid() + " - " + unC.GetClient ↗
                ().getid().ToString() + " - " + unC.GetClient().getnom() + ↗
                " - " + unC.GetClient().getprenom());
206         }
207     }
208     catch (Exception exception)
209     {
210         Console.WriteLine(exception.Message);
211     }
212
213     //Indique à l'utilisateur qu'il n'y a pas de contrat à cette date et ↗
214     //Empêche une erreur lié à l'inexistence de contrat à la date ↗
215     //recherché
216     if (varGlobale.monagence.getlescontrats().Count == 0)
217     {
218         MessageBox.Show("Pas de contrat à cette date, vous pouvez en ↗
219         creer un");
220     }
221     else
222     {
223         cbocontrat.SelectedIndex = 0;
224     }
225 }
226
227 //modifie les valeur d'un contrat
228 private void btnmodifier_Click(object sender, EventArgs e)
229 {
230     varGlobale.monagence.getlescontrats() ↗
231     [cbocontrat.SelectedIndex].setdateheurefin(dtpheurefin.Value); ↗
232     varGlobale.monagence.getlescontrats() ↗
233     [cbocontrat.SelectedIndex].setdatefacture(dtpfacture.Value); ↗
234     varGlobale.monagence.getlescontrats() ↗
235     [cbocontrat.SelectedIndex].setnbkmeffectue(Convert.ToInt32 ↗
236     (txtkm.Text)); ↗
237     varGlobale.monagence.getlescontrats() ↗
238     [cbocontrat.SelectedIndex].setmontantfacture(Convert.ToSingle ↗
239     (txtmontant.Text)); ↗
240     varGlobale.monagence.getlescontrats() ↗
241     [cbocontrat.SelectedIndex].GetVehicule().setetatdeslieux ↗
242     (txtEDL.Text);
243     try
244     {
245         MessageBox.Show(PasserelleContrat.modifcontrat ↗
246         (varGlobale.monagence.getlescontrats() ↗
247         [cbocontrat.SelectedIndex]).ToString() + "modification ↗
248         effectué");
249     }
250 }
```

```
239         catch (Exception exception)
240         {
241             Console.WriteLine(exception.Message);
242         }
243     }
244 }
245 }
246
247 //permet de vérifier si le client existe déjà si ce n'est pas le cas ➤
    un formulaire de création de client sera à compléter
248 private void btnverifclient_Click(object sender, EventArgs e)
249 {
250     //permet de remplir le cboimmat avec la date du dtbheuredebut par ➤
    défaut
251     dtpheuredeb_ValueChanged(sender, e);
252     //vérifie que l'utilisateur a bien renseigné une valeur
253     if (txtnom.Text!="" && txtprenom.Text!="")
254     {
255         try
256         {
257             if (PasserelleClient.QuelleNumeroClient(txtnom.Text, ➤
    txtprenom.Text) == 0)
258             {
259                 //permet de donner des valeurs au textbox nom et ➤
    prenom du formulaire Frmcreationclient
260                 Frmcreationclient frmcc = new Frmcreationclient ➤
    (txtnom.Text, txtprenom.Text);
261                 frmcc.Show();
262             }
263         }
264     }
265     else
266     {
267         visible(true);
268         btncreation.Visible = true;
269         lblexiste.Visible = true;
270         btnverifclient.Visible = false;
271         cboimmat.Visible = true;
272         txtimmat.Visible = false;
273     }
274 }
275 }
276 catch (Exception exception)
277 {
278     Console.WriteLine(exception.Message);
279 }
280 }
281 }
282 }
283 }
284 }
285 }
286 private void cboimmat_SelectedIndexChanged(object sender, EventArgs e)
287 {
288 }
289     if (cboimmat.SelectedIndex != -1)
```

```
290     {
291         txtmarque.Text = lesV[cboimmat.SelectedIndex].getmarque
292             ().ToString();
293         txtmodele.Text = lesV[cboimmat.SelectedIndex].getmodele
294             ().ToString();
295         txtEDL.Text = lesV[cboimmat.SelectedIndex].getetatdeslieux
296             ().ToString();
297     }
298     private void txtkm_TextChanged(object sender, EventArgs e)
299     {
300         if (txtkm.Text != "")
301             txtmontant.Text = Convert.ToString(Convert.ToInt32(txtkm.Text)
302                 * 0.75);
303         else
304             txtmontant.Text = "";
305     }
306     private void btncreation_Click(object sender, EventArgs e)
307     {
308         //remplissage de la variable uneA avant d'être utilisé dans le
309         //constructeur de unCo
310         agence uneA = varGlobale.monagence;
311         //remplissage de la variable unV avant d'être utilisé dans le
312         //constructeur de unCo
313         vehicule unV = lesV[Convert.ToInt32(cboimmat.SelectedIndex)];
314         //remplissage de la variable unC avant d'être utilisé dans le
315         //constructeur de unCo
316         client unC = varGlobale.monagence.getleclientrecherche
317             (PasserelleClient.QuelleNumeroClient(txtnom.Text.Replace("'",
318                 "'"), txtprenom.Text));
319         // Remplissage de la variable unCo
320         contratlocation unCo = new contratlocation(dtpsignature.Value,
321             dtpheuredeb.Value, dtpheurefin.Value, 0, dtpfacture.Value, 0, uneA,
322             unV, unC);
323         try
324         {
325             //ajoute le contrat unCo à la base de données
326             PasserelleContrat.Ajouter_Contrat(unCo);
327         }
328         catch (Exception exception)
329         {
330             Console.WriteLine(exception.Message);
331         }
332     }
333     //permet de définir ce qui est modifiable ou non et la valeur des
334     //textbox
335     txtEDL.Text = "";
336     cboimmat.SelectedIndex = -1;
337     txtmarque.Text = "";
338     txtmodele.Text = "";
339     txtnom.Text = "";
340     txtprenom.Text = "";
```

```
335         txtkm.Text = "";
336         txtkm.ReadOnly = true;
337         dtpheurefin.Enabled = false;
338         txtmontant.Text = "";
339         txtmontant.ReadOnly = true;
340         dtpfacture.Enabled = false;
341         btnfermer_Click(sender,e);
342     }
343
344     //permet d'afficher dans le cbo les véhicules disponible
345     private void dtpheuredeb_ValueChanged(object sender, EventArgs e)
346     {
347
348         try
349         {
350             //remplissage de la liste les V à partir des véhicule de la ↗
351             //base de donnée disponible à la date selectionné ↗
352             lesV = PasserelleContrat.Chargervoiturecbocontrat
353                 (varGlobale.monagence,dtpheuredeb.Value);
354         }
355         catch (Exception exception)
356         {
357             Console.WriteLine(exception.Message);
358         }
359
360         //chargement des véhicule de l'agence dans le combobox a partir de ↗
361         //la liste lesV
362         cboimmat.Items.Clear();
363         foreach (var item in lesV)
364         {
365             cboimmat.Items.Add(item.getimmat());
366         }
367     }
368 }
```

```
1 using System;
2 using System.Collections.Generic;
3 using System.ComponentModel;
4 using System.Data;
5 using System.Drawing;
6 using System.Linq;
7 using System.Text;
8 using System.Threading.Tasks;
9 using System.Windows.Forms;
10 using System.Data.SqlClient;
11
12 namespace ADA1
13 {
14     public partial class frmmodif : Form
15     {
16         public frmmodif()
17         {
18             InitializeComponent();
19         }
20
21         private void frmmodif_Load(object sender, EventArgs e)
22         {
23
24             txtnom.ReadOnly = true;
25             grdhorraire.AllowUserToAddRows = false;
26             //Charge les horraires de l'agence dans la variable      ↗
27             varGlobale.monagence
28             try
29             {
30                 foreach (var unH in PasserelleAgence.chargerleshorraire      ↗
31                     (varGlobale.monagence))
32                 {
33                     varGlobale.monagence.getleshorraires().Add(unH);
34                 }
35             }
36             catch (Exception exception)
37             {
38                 Console.WriteLine(exception.Message);
39             }
40
41             //chargement des jours et horraire dans le grd
42             foreach (var unH in varGlobale.monagence.getleshorraires())
43             {
44
45                 grdhorraire.Rows.Add(unH.GetJour().getnom(),
46                     Convert.ToString(unH.getmatideb()),
47                     Convert.ToString(unH.getmatifin()),
48                     Convert.ToString(unH.getapresmidideb()),
49                     Convert.ToString(unH.getapresmidifin()));
50             }
51
52         }
53
54     }
```

```
55
56
57
58
59
60     //préremplissage des information de l'agence
61
62     txtnom.Text = varGlobale.monagence.getnom();
63     txtadresse.Text = varGlobale.monagence.getadresse();
64     txtcp.Text = varGlobale.monagence.getcp();
65     txtville.Text = varGlobale.monagence.getville();
66     txttel.Text = varGlobale.monagence.gettel();
67 }
68 //modification de l'adresse de l'agence
69 private void btnmodifadrs_Click(object sender, EventArgs e)
70 {
71     varGlobale.monagence.setadresse(txtadresse.Text.Replace("'",
72     "''"));
73     varGlobale.monagence.setcp(txtcp.Text);
74     varGlobale.monagence.setville(txtville.Text.Replace("'", "''"));
75     varGlobale.monagence.settel(txttel.Text);
76     try
77     {
78         //Dis à l'utilisateur que les modifications ont bien eu lieu
79         MessageBox.Show(Convert.ToString(PasserelleAgence.modifagence
80         (varGlobale.monagence) + " modification(s) réussite(s)"));
81     }
82     catch (Exception exception)
83     {
84         Console.WriteLine(exception.Message);
85     }
86 }
87
88 //modification des horaires
89 private void btnmodifhor_Click(object sender, EventArgs e)
90 {
91     int i;
92     //Modifie les valeur des horraire de l'agence dans la variable
93     varGlobale.monagence
94     for (i = 0; i < grdhorraire.Rows.Count; i = i + 1)
95     {
96         varGlobale.monagence.getleshoraires()[i].setmatindeb
97         (grdhorraire[1, i].Value.ToString());
98         varGlobale.monagence.getleshoraires()[i].setmatinfin
99         (grdhorraire[2, i].Value.ToString());
100        varGlobale.monagence.getleshoraires()[i].setapresmidideb
101        (grdhorraire[3, i].Value.ToString());
102        varGlobale.monagence.getleshoraires()[i].setapresmidifin
103        (grdhorraire[4, i].Value.ToString());
104    }
105    try
106        //informe l'utilisateur que les changements ont bien été
```

```
    effectués
104     {
105     MessageBox.Show(Convert.ToString
    (PasserelleAgence.modifhorraires(varGlobale.monagence) + "
    modification(s) réussite(s)"));
106     }
107     catch (Exception exception)
108     {
109     Console.WriteLine(exception.Message);
110     }
111
112     }
113
114
115     }
116 }
117
```

```
1 using System;
2 using System.Collections.Generic;
3 using System.ComponentModel;
4 using System.Data;
5 using System.Drawing;
6 using System.Linq;
7 using System.Text;
8 using System.Threading.Tasks;
9 using System.Windows.Forms;
10
11 namespace ADA1
12 {
13     public partial class Frmcreationclient : Form
14     {
15
16         //création de deux variable permettant l'appelle de valeur d'un autre form
17         private string pourStockernom;
18         private string pourStockerprenom;
19
20         //initialisation des textbox avec les valeur saisie dans frmcontrat
21         public Frmcreationclient(string unNom, string unPrenom)
22         {
23             pourStockernom = unNom;
24             pourStockerprenom = unPrenom;
25             InitializeComponent();
26             txtnom.Text = pourStockernom;
27             txtprenom.Text = pourStockerprenom;
28         }
29         public Frmcreationclient()
30         {
31             InitializeComponent();
32         }
33
34
35
36         private void btncreation_Click(object sender, EventArgs e)
37         {
38
39             client unC = new client(0, txtnom.Text.Replace("'", "''"),
40                 txtprenom.Text, dtpnaiss.Value
41                 , txtmail.Text, txtmdp.Text, txtpermis.Text, txtadresse.Text,
42                 txtcp.Text, txtville.Text);
43
44             try
45             {
46                 //envoi un nouveau client à la base de données
47                 PasserelleClient.Ajouter_Client(unC);
48             }
49             catch (Exception exception)
50             {
51                 Console.WriteLine(exception.Message);
52             }
53
54             //retour de l'idclient par la base de données afin de le mettre à
55             jour la variable unC
```



```
53     try
54     {
55         unC.Setid(PasserelleClient.QuelleNumeroClient(unC.getnom(),
56             unC.getprenom()));
57     }
58     catch (Exception exception)
59     {
60         Console.WriteLine(exception.Message);
61     }
62     //ajout de unC dans la liste clients de la variable
63     varGlobale.monagence
64     varGlobale.monagence.getlesclient().Add(unC);
65     Frmcreationclient.ActiveForm.Close();
66 }
67 }
68 }
69 }
```

```
1 using System;
2 using System.Collections.Generic;
3 using System.ComponentModel;
4 using System.Data;
5 using System.Drawing;
6 using System.Linq;
7 using System.Text;
8 using System.Threading.Tasks;
9 using System.Windows.Forms;
10
11 namespace ADA1
12 {
13     public partial class vehiculeagence : Form
14     {
15         //initialisation de la list comportant des catégories
16         List<categorie> LesCategories;
17
18         public vehiculeagence()
19         {
20             InitializeComponent();
21         }
22
23         private void vehiculeagence_Load(object sender, EventArgs e)
24         {
25             //instantiation d'une variable contenant une liste d'objets
                //catégories se trouvant dans la base de données
26             try
27             {
28                 LesCategories = PasserelleVehicule.ChargerCategorie();
29             }
30             catch (Exception exception)
31             {
32                 Console.WriteLine(exception.Message);
33             }
34             //parcours de la liste
35             foreach (categorie unC in LesCategories)
36             {
37                 //Ajoute des objets categories dans les combo box
38                 cbocat.Items.Add(unC.getid() + "-" + unC.getlibelle1());
39                 cbonewcat.Items.Add(unC.getlibelle1());
40             }
41             //remplit automatiquement le textbox txtagence selon l'agence
                //sélectionné à la connection
42             txtagence.Text = Convert.ToString(varGlobale.monagence.getid());
43             //supprime la dernière ligne du grd
44             grdvehicule.AllowUserToAddRows = false;
45
46         }
47
48         private void cbocat_SelectedIndexChanged(object sender, EventArgs e)
49         {
50             //évite une erreur index outrange
51             if(cbocat.SelectedIndex!=-1)
52             {
53
54                 txtimmat.Text = "";
```

```
55     txtmarque.Text = "";
56     txtmodele.Text = "";
57     txtetatlieux.Text = "";
58     cbonewcat.SelectedIndex = -1;
59     txtagence.Text = varGlobale.monagence.getid().ToString();
60     //efface les objets véhicule se trouvant dans la liste
        getlesvehics
61     varGlobale.monagence.getlesvehics().Clear();
62     //efface les lignes du grdvehicule
63     grdvehicule.Rows.Clear();
64     //defini l'indice selon la ligne sélectionnée
65     int indice = cbocat.SelectedIndex;
66     //innitialise un objet categorie et affecte un objet pris dans
        la liste LesCategories selon l'indice
67     categorie choixcat = LesCategories[indice];
68     //obtient l'id d'une categorie selon un indice et l'affecte à
        un palier
69     int palier = LesCategories[indice].getid();
70     //permet d'afficher une image selon la caegorie sélectionnée
71     try
72     {
73         pictureBox1.Image = Image.FromFile(Application.StartupPath
        + "\\images\\" + PasserelleVehicule.getidcategorie
        (palier).getphoto());
74     }
75     catch (Exception exception)
76     {
77         Console.WriteLine(exception.Message);
78     }
79     pictureBox1.SizeMode = PictureBoxSizeMode.StretchImage;
80     //parcours la liste des voitures selon agence et la categorie
        choisie
81     try
82     {
83         foreach (vehicule uneV in
        PasserelleVehicule.ChargerVoitures(varGlobale.monagence,
        choixcat))
84         {
85             //ajoute les voitures sélectionnées dans le grd
86             grdvehicule.Rows.Add(uneV.getimmat(), uneV.getmarque
        (), uneV.getmodele(), uneV.getetatdeslieux(),
        uneV.getidagence());
87             //les ajoutes aussi dans une liste
88             varGlobale.monagence.getlesvehics().Add(uneV);
89         }
90     }
91     catch (Exception exception)
92     {
93         Console.WriteLine(exception.Message);
94     }
95 }
96 //si le grd n'est pas vide les zone de textes se remplisse
        automatique avec les valeurs du premier véhicules du grd
97 if(grdvehicule.RowCount!=0)
98 {
99
```

```
100         try
101         {
102             vehicule unV = varGlobale.monagence.getlesvehics()
103             [grdvehicule.SelectedRows.Count];
104             txtimmat.Text = unV.getimmat();
105             txtimmat.ReadOnly = true;
106             txtmarque.Text = unV.getmarque();
107             txtmodele.Text = unV.getmodele();
108             txtetatlieux.Text = unV.getetatdeslieux();
109             categorie uneC = PasserelleVehicule.getidcategorie
110             (unV.getidcat());
111             cbonewcat.SelectedItem = uneC.getlibelle1();
112             txtagence.Text = unV.getidagence().ToString();
113             txtagence.ReadOnly = true;
114         }
115         catch (Exception exception)
116         {
117             Console.WriteLine(exception.Message);
118         }
119     else
120         txtimmat.ReadOnly = false;
121
122
123     }
124
125     private void btncreervehicule_Click(object sender, EventArgs e)
126     {
127         //empêche la création du véhicule si les valeurs essentielles ne
128         //sont pas renseignés
129         if (txtimmat.Text != "" && cbonewcat.SelectedIndex!=-1)
130         {
131             //initialise et affecte un objet véhicule
132             vehicule unV = new vehicule(txtimmat.Text, txtmarque.Text,
133             txtmodele.Text, txtetatlieux.Text.Replace("'", ""),
134             LesCategories[cbonewcat.SelectedIndex].getid(),
135             Convert.ToInt16(txtagence.Text));
136
137             //ajoute un véhicule dans la BDD
138             PasserelleVehicule.Ajouter_Vehicule(unV);
139             MessageBox.Show("Véhicule Créé");
140         }
141         catch (Exception exception)
142         {
143             //si l'immatriculation est déjà affecté à un véhicule
144             //renvoie un message
145             Console.WriteLine(exception.Message);
146             MessageBox.Show("Le vehicule existe déjà");
147         }
148
149         //effectue le changement dans le grd en temps réel
150         cbocat_SelectedIndexChanged(sender, e);
```

```
149     }
150
151   }
152
153   private void btnsup_Click(object sender, EventArgs e)
154   {
155       //Empêche l'utilisateur de supprimer quand rien n'est dans le grd ↗
156       //    empechant une erreur outofrange
157       if (grdvehicule.RowCount != 0)
158       {
159           //affecte le véhicule se trouvant dans la BDD en cliquant sur ↗
160           //    une ligne du grd
161           vehicule unV = varGlobale.monagence.getlesvehics() ↗
162           [grdvehicule.CurrentRow.Index];
163
164           try
165           {
166               //supprime le véhicule
167               PasserelleVehicule.Supprimer_Vehicule(unV);
168           }
169           catch (Exception exception)
170           {
171               Console.WriteLine(exception.Message);
172               MessageBox.Show("le véhicule ne peut pas être supprimer car ↗
173               il est associé à un contrat");
174           }
175
176           varGlobale.monagence.getlesvehics().Remove(unV);
177           //met à jour le grd
178           cbocat_SelectedIndexChanged(sender, e);
179       }
180   }
181
182   private void btnupdate_Click(object sender, EventArgs e)
183   {
184       //Empêche l'utilisateur de modifier quand rien n'est dans le grd ↗
185       //    empechant une erreur outofrange
186       if (grdvehicule.RowCount!=0)
187       {
188           //prend l'objet véhicule sélectionné dans le grd
189           vehicule unV = varGlobale.monagence.getlesvehics() ↗
190           [grdvehicule.CurrentRow.Index];
191           //change la valeur modele,marque,etatdeslieux,setidcat de ↗
192           //    l'objet précédant
193           unV.setmodele(txtmodele.Text);
194           unV.setmarque(txtmarque.Text);
195           unV.setetatdeslieux(txtetatlieux.Text);
196           unV.setidcat(LesCategories[cbonewcat.SelectedIndex].getid());
197           try
198           {
199               //le met à jour dans la BDD
200               PasserelleVehicule.UpdateVehicule(unV);
201           }
202           catch (Exception exception)
203           {
```

```
198         Console.WriteLine(exception.Message);
199     }
200
201
202
203     cbocat_SelectedIndexChanged(sender, e);
204     MessageBox.Show("Voiture modifiée");
205 }
206
207
208 }
209
210 private void grdvehicule_CellClick(object sender,           ↗
    DataGridViewCellEventArgs e)
211 {
212     //permet de mettre dans les text box les parametres de l'objet   ↗
    //sélectionné dans le grd
213     try
214     {
215         vehicule unV = varGlobale.monagence.getlesvehics()           ↗
            [grdvehicule.CurrentRow.Index];
216         txtimmat.Text = unV.getimmat();
217         txtimmat.ReadOnly = true;
218         txtmarque.Text = unV.getmarque();
219         txtmodele.Text = unV.getmodele();
220         txtetatlieux.Text = unV.getetatdeslieux();
221         categorie uneC = PasserelleVehicule.getidcategorie           ↗
            (unV.getidcat());
222         cbonewcat.SelectedItem = uneC.getlibelle1();
223         txtagence.Text = unV.getidagence().ToString();
224         txtagence.ReadOnly = true;
225
226     }
227     catch (Exception exception)
228     {
229         Console.WriteLine(exception.Message);
230     }
231
232
233
234 }
235
236 private void btnretourcrea_Click(object sender, EventArgs e)
237 {
238     //le but de ce bouton est de diminuer le nombre d'interaction que   ↗
    //l'utilisateur doit effectuer pour creer une voiture après un   ↗
    //update
239     txtimmat.Text = "";
240     txtimmat.ReadOnly = false;
241     txtmarque.Text = "";
242     txtmodele.Text = "";
243     txtetatlieux.Text = "";
244     cbonewcat.SelectedIndex = -1;
245     txtagence.Text = varGlobale.monagence.getid().ToString();
246     txtagence.ReadOnly = false;
247 }
```

248

249 }

250 }

251

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6 using System.Data.SqlClient;
7
8 namespace ADA1
9 {
10     class agence
11     {
12         private int id;
13         private string nom;
14         private string adresse;
15         private string cp;
16         private string ville;
17         private string tel;
18         private string email;
19         private string mdp;
20         List<horraires> leshorraires;
21         List<contratlocation> lescontrats;
22         List<client> lesclients;
23         List<vehicule> lesvehics;
24
25         public agence(int unId, string unNom, string uneAdresse, string unCp, ↗
                string uneVille, string unTel, string unEmail, string unMdp)
26         {
27             id = unId;
28             nom = unNom;
29             adresse = uneAdresse;
30             cp = unCp;
31             ville = uneVille;
32             tel = unTel;
33             email = unEmail;
34             mdp = unMdp;
35             leshorraires = new List<horraires>();
36             lescontrats = new List<contratlocation>();
37             lesclients = new List<client>();
38             lesvehics = new List<vehicule>();
39
40         }
41
42         //methode pour rechercher un client par rapport à son id dans une ↗
            liste
43         public client getleclientrecherche(int unID)
44         {
45             bool Trouve = false;
46             int i = 0;
47             while(Trouve == false && i <= lesclients.Count() )
48         {
49                 if(lesclients[i].getid() == unID)
50                     Trouve = true;
51                 else
52                     i++;
53             }
54
```



```
55         if(Trouve == true)
56             return(lesclients[i]);
57         else
58             return null;
59     }
60 }
61 public List<horraires> getleshorraires()
62 {
63     return leshorraires;
64 }
65
66 public List<contratlocation> getlescontrats()
67 {
68     return lescontrats;
69 }
70
71 public List<client> getlesclient()
72 {
73     return lesclients;
74 }
75
76 public List<vehicule> getlesvehics()
77 {
78     return lesvehics;
79 }
80 public int getid()
81 {
82     return id;
83 }
84 public string getnom()
85 {
86     return nom;
87 }
88 public string getadresse()
89 {
90     return adresse;
91 }
92
93 public string getcp()
94 {
95     return cp;
96 }
97 public string getville()
98 {
99     return ville;
100 }
101 public string gettel()
102 {
103     return tel;
104 }
105 public string getemail()
106 {
107     return email;
108 }
109 public string getmdp()
110 {
```

```
111         return mdp;
112     }
113
114     public void setid(int unId)
115     {
116         id = unId;
117     }
118
119     public void setnom(string unNom)
120     {
121         nom = unNom;
122     }
123
124     public void setadresse(string uneAdresse)
125     {
126         adresse = uneAdresse;
127     }
128
129     public void setcp(string unCp)
130     {
131         cp = unCp;
132     }
133
134     public void setville(string uneVille)
135     {
136         ville = uneVille;
137     }
138     public void settel(string unTel)
139     {
140         tel = unTel;
141     }
142     public void setemail(string unEmail)
143     {
144         email = unEmail;
145     }
146     public void setmdp(string unMdp)
147     {
148         mdp = unMdp;
149     }
150 }
151
152 }
153
```

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6 using System.Data.SqlClient;
7
8 namespace ADA1
9 {
10     class categorie
11     {
12         private int id;
13         private string libelle1;
14         private string libelle2;
15         private string description;
16         private string typemodele;
17         private double tarifjournee;
18         private double tarifkm;
19         private string photo;
20
21         public categorie(int unid,string lib1,string lib2,string unedescription,string untypemod, double untarifj, double untarifkm,string unephoto)
22         {
23             id = unid;
24             libelle1 = lib1;
25             libelle2 = lib2;
26             description = unedescription;
27             typemodele = untypemod;
28             tarifjournee = untarifj;
29             tarifkm = untarifkm;
30             photo = unephoto;
31         }
32         public int getid()
33         {
34             return id;
35         }
36
37         public string getlibelle1()
38         {
39             return libelle1;
40         }
41
42         public string getlibelle2()
43         {
44             return libelle2;
45         }
46
47         public string getdescription()
48         {
49             return description;
50         }
51
52         public string gettypemodele()
53         {
54             return typemodele;
```

```
55     }
56
57     public double gettarifjournee()
58     {
59         return tarifjournee;
60     }
61
62     public double gettarifkm()
63     {
64         return tarifkm;
65     }
66
67     public string getphoto()
68     {
69         return photo;
70     }
71
72     public void ToString(string unString)
73     {
74         Convert.ToString(unString);
75     }
76 }
77 }
78
```

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace ADA1
8 {
9     class client
10    {
11        private int id;
12        private string nom;
13        private string prenom;
14        private DateTime datenais;
15        private string email;
16        private string mdp;
17        private string nopermis;
18        private string adresse;
19        private string cp;
20        private string ville;
21
22        public client(int unId,string unN,string unP,DateTime uneDN,string ↗
23            unE, string unMdp, string unNoP,string uneA,string unCp,string uneV)
24        {
25            id = unId;
26            nom = unN;
27            prenom = unP;
28            datenais = uneDN;
29            email = unE;
30            mdp = unMdp;
31            nopermis = unNoP;
32            adresse = uneA;
33            cp = unCp;
34            ville = uneV;
35        }
36
37        public int getid()
38        {
39            return id;
40        }
41
42        public string getnom()
43        {
44            return nom;
45        }
46
47        public string getprenom()
48        {
49            return prenom;
50        }
51
52        public DateTime getdatenais()
53        {
54            return datenais;
55        }
56    }
57 }
```

```
56
57     public string getemail()
58     {
59         return email;
60     }
61
62     public string getmdp()
63     {
64         return mdp;
65     }
66
67     public string getnopermis()
68     {
69         return nopermis;
70     }
71
72     public string getadresse()
73     {
74         return adresse;
75     }
76
77     public string getcp()
78     {
79         return cp;
80     }
81
82     public string getville()
83     {
84         return ville;
85     }
86
87     public void Setid(int unId)
88     {
89         id = unId;
90     }
91
92     public void Setnom(string unN)
93     {
94         nom = unN;
95     }
96
97     public void Setprenom(string unP)
98     {
99         prenom = unP;
100    }
101
102    public void Setdatenais(DateTime uneD)
103    {
104        datenais = uneD;
105    }
106
107    public void Setemail(string unE)
108    {
109        email = unE;
110    }
111
```

```
112     public void Setmdp(string unMdp)
113     {
114         mdp = unMdp;
115     }
116
117     public void Setnopermis(string unNoP)
118     {
119         nopermis = unNoP;
120     }
121
122     public void Setadresse(string uneA)
123     {
124         adresse = uneA;
125     }
126
127     public void Setcp(string unCp)
128     {
129         cp = unCp;
130     }
131
132     public void Setville(string uneV)
133     {
134         ville = uneV;
135     }
136     }
137 }
138
```

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6 using System.Data.SqlClient;
7
8 namespace ADA1
9 {
10     class contratlocation
11     {
12         private int id;
13         private DateTime datesignature;
14         private DateTime dateheuredeb;
15         private DateTime dateheurefin;
16         private int nbkmeffectue;
17         private DateTime datefacture;
18         private float montantfacture;
19         private agence Lagence;
20         private vehicule Levehic;
21         private client Leclient;
22
23         //constructeur globale de la classe
24         public contratlocation(int unId,DateTime uneDS,DateTime           ↗
25             uneDHD,DateTime uneDHF,int unNBK,DateTime uneDF,float unM,agence ↗
26             uneAG,vehicule unV,client unC)
27         {
28             id = unId;
29             datesignature = uneDS;
30             dateheuredeb = uneDHD;
31             dateheurefin = uneDHF;
32             nbkmeffectue = unNBK;
33             datefacture = uneDF;
34             montantfacture = unM;
35             Lagence = uneAG;
36             Levehic = unV;
37             Leclient = unC;
38         }
39
40         //Constructeur permettant la création d'un nouveau contrat dans la ↗
41         base de donnée
42         public contratlocation( DateTime uneDS, DateTime uneDHD, DateTime ↗
43             uneDHF, int unNBK, DateTime uneDF, float unM, agence uneAG, ↗
44             vehicule unV, client unC)
45         {
46             datesignature = uneDS;
47             dateheuredeb = uneDHD;
48             dateheurefin = uneDHF;
49             nbkmeffectue = unNBK;
50             datefacture = uneDF;
51             montantfacture = unM;
52             Lagence = uneAG;
53             Levehic = unV;
54             Leclient = unC;
```



```
52
53     }
54     public int getid()
55     {
56         return id;
57     }
58
59     public DateTime getdatesignature()
60     {
61         return datesignature;
62     }
63
64     public DateTime getdateheuredeb()
65     {
66         return dateheuredeb;
67     }
68
69     public DateTime getdateheurefin()
70     {
71         return dateheurefin;
72     }
73     public int getnbkmeffectue()
74     {
75         return nbkmeffectue;
76     }
77
78     public DateTime getdatefacture()
79     {
80         return datefacture;
81     }
82
83     public float getmontantfacture()
84     {
85         return montantfacture;
86     }
87
88     public agence GetAgence()
89     {
90         return Lagence;
91     }
92
93     public vehicule GetVehicule()
94     {
95         return Levehic;
96     }
97
98     public client GetClient()
99     {
100         return Leclient;
101     }
102     public void setid(int unId)
103     {
104         id = unId;
105     }
106
107     public void setdatesignature(DateTime uneDS)
```

```
108     {
109         datesignature = uneDS;
110     }
111
112     public void setdateheuredeb(DateTime uneDHD)
113     {
114         dateheuredeb = uneDHD;
115     }
116
117     public void setdateheurefin(DateTime uneDHF)
118     {
119         dateheurefin = uneDHF;
120     }
121
122     public void setnbkmeffectue(int unNBK)
123     {
124         nbkmeffectue = unNBK;
125     }
126
127     public void setdatefacture(DateTime uneDF)
128     {
129         datefacture = uneDF;
130     }
131
132     public void setmontantfacture(float unM)
133     {
134         montantfacture = unM;
135     }
136
137     public void SetAgence(agence uneAg)
138     {
139         Lagence = uneAg;
140     }
141
142     public void SetVehicule(vehicule unV)
143     {
144         Levehic = unV;
145     }
146
147     public void SetClient(client unC)
148     {
149         Leclient = unC;
150     }
151 }
152 }
153
154
155
```

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6 using System.Data.SqlClient;
7
8 namespace ADA1
9 {
10     class horraires
11     {
12         jour unjour;
13         string matindeb;
14         string matinfin;
15         string apresmidideb;
16         string apresmidifin;
17
18         public horraires(jour unJ,string unmd,string unmf, string unamd, string ↗
19             unamf)
20         {
21             unjour = unJ;
22             matindeb = unmd;
23             matinfin = unmf;
24             apresmidideb = unamd;
25             apresmidifin = unamf;
26         }
27         public jour GetJour()
28         {
29             return unjour;
30         }
31
32         public string getmatindeb()
33         {
34             return matindeb;
35         }
36
37         public string getmatinfin()
38         {
39             return matinfin;
40         }
41
42         public string getapresmidideb()
43         {
44             return apresmidideb;
45         }
46
47         public string getapresmidifin()
48         {
49             return apresmidifin;
50         }
51
52         public void setmatindeb(string unmd)
53         {
54             matindeb = unmd;
55         }
56     }
57 }
```

```
56
57     public void setmatinfin(string unmf)
58     {
59         matinfin = unmf;
60     }
61
62     public void setapresmidideb(string unamd)
63     {
64         apresmidideb = unamd;
65     }
66     public void setapresmidifin(string unamf)
67     {
68         apresmidifin = unamf;
69     }
70 }
71
72 }
73
```

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace ADA1
8 {
9     class jour
10    {
11        private int idjour;
12        private string nom;
13
14        public jour(int unidjour, string unjour )
15        {
16            idjour = unidjour;
17            nom = unjour;
18        }
19
20
21        public int getid()
22        {
23            return idjour;
24        }
25        public string getnom()
26        {
27            return nom;
28        }
29
30        public void setid(int unidjour)
31        {
32            idjour = unidjour;
33        }
34
35        public void setnom(string unjour)
36        {
37            nom = unjour;
38        }
39    }
40 }
41
```

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6 using System.Data.SqlClient;
7
8 namespace ADA1
9 {
10     class vehicule
11     {
12         private string immat;
13         private string modele;
14         private string marque;
15         private string etatdeslieux;
16         private int idcat;
17         private int idagence;
18         public vehicule(string uneimmat, string unemarque, string unmodele,
19             string unetatdeslieux, int unidcat, int unidagence)
20         {
21             immat = uneimmat;
22             marque = unemarque;
23             modele = unmodele;
24             etatdeslieux = unetatdeslieux;
25             idcat = unidcat;
26             idagence = unidagence;
27         }
28         public string getimmat()
29         {
30             return immat;
31         }
32         public string getmodele()
33         {
34             return modele;
35         }
36         public string getmarque()
37         {
38             return marque;
39         }
40         public string getetatdeslieux()
41         {
42             return etatdeslieux;
43         }
44         public int getidcat()
45         {
46             return idcat;
47         }
48         public int getidagence()
49         {
50             return idagence;
51         }
52         public void setmodele(string unM)
```

```
56     {
57         modele = unM;
58     }
59
60     public void setmarque(string uneM)
61     {
62
63         marque = uneM;
64     }
65
66     public void setetatdeslieux(string unE)
67     {
68         etatdeslieux = unE;
69     }
70
71     public void setidcat(int unidC)
72     {
73         idcat = unidC;
74     }
75     }
76 }
77
```

```
1 using System;
2 using System.Collections.Generic;
3 using System.Data.SqlClient;
4 using System.Data.Odbc;
5
6 namespace ADA1
7 {
8     class PasserelleAgence
9     {
10
11
12         // permet de connecter l'agence
13         public static agence connecter(int idagence, string mdpagence)
14         {
15             SqlConnection cnn;
16             SqlCommand cmd;
17             SqlDataReader drr;
18             agence uneAgence;
19             string req;
20             req = "SELECT * from agence where idagence =" + idagence + "'and
                mdp=" + mdpagence + "'";
21             cnn = new SqlConnection(varglob.strconnect);
22             cnn.Open();
23             cmd = new SqlCommand(req, cnn);
24             drr = cmd.ExecuteReader();
25             Boolean existanreg = drr.Read();
26             if (existanreg == true)
27                 uneAgence = new agence(drr.GetInt32(0), drr.GetString(1),
                drr.GetString(2), drr.GetString(3), drr.GetString(4),
                drr.GetString(5), drr.GetString(6), drr.GetString(7));
28             else
29                 uneAgence = null;
30             drr.Close();
31             cnn.Close();
32             return uneAgence;
33         }
34     }
35
36
37
38
39
40
41
42         //modifie l'adresse de l'agence
43         public static Int32 modifagence(agence Uneagence)
44         {
45             SqlConnection cnn;
46             SqlCommand cmd;
47             string req;
48             req = "UPDATE agence set adresse=" + Uneagence.getadresse() +
                "',cp=" + Uneagence.getcp() + "',ville=" + Uneagence.getville
                () + "',Tel=" + Uneagence.gettel() + "'where idagence =" +
                Uneagence.getid();
49             cnn = new SqlConnection(varglob.strconnect);
50             cnn.Open();
```



```
51     cmd = new SqlCommand(req, cnn);
52     int nbenreg = cmd.ExecuteNonQuery();
53     cnn.Close();
54     return nbenreg;
55
56 }
57
58 //modifie les horaires de l'agence
59 public static Int32 modifhorraires(agence Uneagence)
60 {
61     SqlConnection cnn;
62     SqlCommand cmd;
63     string req;
64     int nbenreg = 0;
65     cnn = new SqlConnection(varglob.strconnect);
66     cnn.Open();
67     foreach (var unH in Uneagence.getleshorraires())
68     {
69         req = "UPDATE ouvrir set matindeb=" + unH.getmatindeb() +
70             "',matinfin=" + unH.getmatinfin() + "',apresmidideb=" +
71             unH.getapresmidideb() + "',apresmidifin=" +
72             unH.getapresmidifin() + "'where idagence=" + Uneagence.getid
73             () + "and idjour=" + unH.GetJour().getid();
74         cmd = new SqlCommand(req, cnn);
75         nbenreg = nbenreg + cmd.ExecuteNonQuery();
76     }
77
78     cnn.Close();
79     return nbenreg;
80
81 //charge les horaires de l'agence
82 public static List<horraires> chargerleshorraire(agence uneA)
83 {
84     horraires unH;
85     SqlConnection cnn;
86     SqlCommand cmd;
87     SqlDataReader drr;
88     string req;
89     List<horraires> leshorraires = new List<horraires>();
90     req = "SELECT
91         j.idjour,libelle,matindeb,matinfin,apresmidideb,apresmidifin
92         from ouvrir o inner join joursemaine j on o.idjour = j.idjour
93         where idagence=" + uneA.getid();
94     cnn = new SqlConnection(varglob.strconnect);
95     cnn.Open();
96     cmd = new SqlCommand(req, cnn);
97     drr = cmd.ExecuteReader();
98     while (drr.Read())
99     {
100         unH = new horraires(new jour(drr.GetInt32(0), drr.GetString
101             (1), drr.GetString(2), drr.GetString(3), drr.GetString(4),
102             drr.GetString(5));
103         leshorraires.Add(unH);
104     }
105 }
```

```
98         }
99         drr.Close();
100        cnn.Close();
101        return leshorraires;
102
103    }
104
105
106
107
108
109    }
110
111 }
112
113
```

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6 using System.Data.SqlClient;
7
8 namespace ADA1
9 {
10     class PasserelleClient
11     {
12
13
14         //permet d'ajouter un client
15         public static void Ajouter_Client(client unC)
16         {
17             SqlConnection cnn;
18             SqlCommand cmd;
19             string req;
20             req = "insert into client
                (nom,prenom,datenais,email,mdp,nopermis,adresse,cp,ville) values
                ('" + unC.getnom() + "','" + unC.getprenom() + "','" +
                unC.getdatenais() + "','" + unC.getemail() + "','" + unC.getmdp()
                + "','" + unC.getnopermis() + "','" + unC.getadresse() + "','" +
                unC.getcp() + "','" + unC.getville() + "')";
21             cnn = new SqlConnection(varglob.strconnect);
22             cnn.Open();
23             cmd = new SqlCommand(req, cnn);
24
25             cmd.ExecuteNonQuery();
26             cnn.Close();
27         }
28
29         //permet de charger les client
30         public static List<client> ChargerClients()
31         {
32             client unC;
33             SqlConnection cnn;
34             SqlCommand cmd;
35             SqlDataReader drr;
36             string req;
37             List<client> LesClients = new List<client>();
38             req = "SELECT * from client";
39
40
41             cnn = new SqlConnection(varglob.strconnect);
42             cnn.Open();
43             cmd = new SqlCommand(req, cnn);
44             drr = cmd.ExecuteReader();
45             while (drr.Read())
46             {
47
48                 unC = new client(Convert.ToInt32(drr[0]), Convert.ToString(drr
                    [1]), Convert.ToString(drr[2]), Convert.ToDateTime(drr[3]),
                    Convert.ToString(drr[4]), Convert.ToString(drr[5]),
                    Convert.ToString(drr[6]), Convert.ToString(drr[7]),
```

```
Convert.ToString(drr[8]), Convert.ToString(drr[9]));
49     LesClients.Add(unC);
50     }
51     drr.Close();
52     cnn.Close();
53     return (LesClients);
54
55 }
56
57 //permet de trouver le numéro d'un client par rapport à son nom et son  ↗
58 //prenom
59 public static int QuelleNumeroClient(string unNom, string unPrenom)
60 {
61
62     SqlConnection cnn;
63     SqlCommand cmd;
64     SqlDataReader drr;
65     string req;
66     req = "select idclient from client where nom like '" + unNom + "'  ↗
67           and prenom like " +
68           "' '" + unPrenom + "' ";
69
70     cnn = new SqlConnection(varglob.strconnect);
71     cnn.Open();
72     cmd = new SqlCommand(req, cnn);
73     drr = cmd.ExecuteReader();
74     drr.Read();
75     //permet de vérifier l'existence ou non d'une valeur
76     if (drr.HasRows)
77     {
78         return (Convert.ToInt32(drr["idclient"]));
79     }
80     else
81     {
82         return 0;
83     }
84 }
85
86
87
88 }
89 }
90 }
91
```

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6 using System.Data.SqlClient;
7
8 namespace ADA1
9 {
10     class PasserelleContrat
11     {
12
13
14         //permet de charger les contrat d'une journée de l'agence
15         public static List<contratlocation> chargerlescontrats(DateTime unedate, agence uneAg)
16         {
17
18             contratlocation unC;
19             SqlConnection cnn;
20             SqlCommand cmd;
21             SqlDataReader drr;
22             string req;
23             List<contratlocation> lescontrats = new List<contratlocation>();
24             req = "SELECT idcontrat,datesignature,dateheuredeb,dateheurefin,
25                 nbkmeffectue" +
26                 ",datefature,montantfature,cl.idagence" +
27                 ",cl.idvehic,v.marque,v.modele,v.etatdeslieux," +
28                 "cl.idclient,c.nom,c.prenom,c.datenais,c.email,c.nopermis,c.adresse,c.cp,c.vile" +
29                 " from contratlocation cl " +
30                 " inner join vehicule v on cl.idvehic=v.immat" +
31                 " inner join client c on cl.idclient=c.idclient " +
32                 " where dateheuredeb between '" + unedate.ToString("dd/MM/yyyy") + "' and '" + unedate.AddDays(1).ToString("dd/MM/yyyy") + "' and cl.idagence=" + uneAg.getid();
33             cnn = new SqlConnection(varglob.strconnect);
34             cnn.Open();
35             cmd = new SqlCommand(req, cnn);
36             drr = cmd.ExecuteReader();
37
38             while (drr.Read())
39             {
40
41                 if(Convert.IsDBNull(drr["nbkmeffectue"]))
42                 {
43                     unC = new contratlocation(Convert.ToInt32(drr["idcontrat"]), Convert.ToDateTime(drr["datesignature"].ToString()),
44                     , Convert.ToDateTime(drr["dateheuredeb"]), Convert.ToDateTime(drr["dateheurefin"]),
45                     0, DateTime.Now, 0,
46                     uneAg,
47                     new vehicule(Convert.ToString(drr["idvehic"]), Convert.ToString
```

```
(drr["marque"])
48     , Convert.ToString(drr["modele"]), Convert.ToString(drr      ↗
    ["etatdeslieux"]), 0, uneAg.getid()),
49     new client(Convert.ToInt32(drr["idclient"]), Convert.ToString      ↗
    (drr["nom"]), Convert.ToString(drr["prenom"]))
50     , Convert.ToDateTime(drr["datenais"]), Convert.ToString(drr      ↗
    ["email"]), null, Convert.ToString(drr["nopermis"]),
51     Convert.ToString(drr["adresse"]), Convert.ToString(drr["cp"]),      ↗
    Convert.ToString(drr["ville"])); ;
52     lescontrats.Add(unC);
53     }
54     else
55     {
56         unC = new contratlocation(Convert.ToInt32(drr      ↗
    ["idcontrat"]), Convert.ToDateTime(drr      ↗
    ["datesignature"].ToString())
57     , Convert.ToDateTime(drr["dateheuredeb"]), Convert.ToDateTime(drr      ↗
    ["dateheurefin"]),
58
59     Convert.ToInt32(drr["nbkmeffectue"]), Convert.ToDateTime(drr      ↗
    ["datefacture"]),
60
61     Convert.ToSingle(drr["montantfacture"]),
62     uneAg,
63     new vehicule(Convert.ToString(drr["idvehic"]), Convert.ToString      ↗
    (drr["marque"])
64     , Convert.ToString(drr["modele"]), Convert.ToString(drr      ↗
    ["etatdeslieux"]), 0, uneAg.getid()),
65     new client(Convert.ToInt32(drr["idclient"]), Convert.ToString(drr      ↗
    ["nom"]), Convert.ToString(drr["prenom"])
66     , Convert.ToDateTime(drr["datenais"]), Convert.ToString(drr      ↗
    ["email"]), null, Convert.ToString(drr["nopermis"]),
67     Convert.ToString(drr["adresse"]), Convert.ToString(drr["cp"]),      ↗
    Convert.ToString(drr["ville"])); ;
68     lescontrats.Add(unC);
69     }
70
71
72
73
74
75
76
77     }
78
79
80
81
82
83     drr.Close();
84     cnn.Close();
85     return lescontrats;
86
87     }
88
89     //modifie les valeurs du contrats
```

```
90     public static Int32 modifcontrat(contratlocation unContrat)
91     {
92         SqlConnection cnn;
93         SqlCommand cmd;
94         string req;
95         req = "UPDATE contratlocation set dateheurefin='" +
96             unContrat.getdateheurefin() +
97             "',nbkmeffectue='" + unContrat.getnbkmeffectue() +
98             "',datefacture='" + unContrat.getdatefacture() +
99             "',montantfacture='" + unContrat.getmontantfacture() + "'where
100             idcontrat =" + unContrat.getid();
101         cnn = new SqlConnection(varglob.strconnect);
102         cnn.Open();
103         cmd = new SqlCommand(req, cnn);
104         int nbenreg = cmd.ExecuteNonQuery();
105         //modifie la valeur de EDL de l'état des lieu
106         req = "UPDATE vehicule set etatdeslieux='" + unContrat.GetVehicule
107             ().getetatdeslieux() + "' where immat = '" +
108             unContrat.GetVehicule().getimmat() + "'";
109         cmd = new SqlCommand(req, cnn);
110         nbenreg = nbenreg + cmd.ExecuteNonQuery();
111         cnn.Close();
112         return nbenreg;
113     }
114
115     //permet l'ajout d'un nouveau contrat
116     public static void Ajouter_Contrat(contratlocation unCo)
117     {
118         SqlConnection cnn;
119         SqlCommand cmd;
120         string req;
121         req = "insert into contratlocation
122             (nbkmeffectue,montantfacture,dateheuredeb,datesignature,idagence
123             ,idvehic,idclient,dateheurefin,datefacture) " +
124             "values(" + unCo.getnbkmeffectue() + "," +
125             unCo.getmontantfacture() + "," + unCo.getdateheuredeb() +
126             "','" + unCo.getdatesignature() + "','" + unCo.GetAgence
127             ().getid() + "','" + unCo.GetVehicule().getimmat() + "','" +
128             unCo.GetClient().getid() + "','" + unCo.getdateheurefin() +
129             "','" + unCo.getdatefacture() + "')";
130         cnn = new SqlConnection(varglob.strconnect);
131         cnn.Open();
132         cmd = new SqlCommand(req, cnn);
133
134         cmd.ExecuteNonQuery();
135         cnn.Close();
136     }
137
138     //permet de charger les voiture afin de les selectionner dans le
139     frmcontrat ayant pour but d'en créer un nouveau
140     public static List<vehicule> Chargerveiturecbocontrat(agence
141         uneagence, DateTime unedate)
142     {
143         vehicule uneV;
144         SqlConnection cnn;
```

```
133     SqlCommand cmd;
134     SqlDataReader drr;
135     string req;
136     List<vehicule> lesVoitures = new List<vehicule>();
137     //req = "SELECT                                ↗
138     immat,marque,modele,etatdeslieux,idcategorie,idagence from ↗
139     vehicule where idagence=" + uneagence.getid() ; ↗
140     req = "SELECT                                ↗
141     immat,marque,modele,etatdeslieux,idcategorie,idagence from ↗
142     vehicule where idagence=" + uneagence.getid() + " and immat ↗
143     not in(select idvehic from contratlocation where dateheuredeb ↗
144     between '" + unedate.ToString("dd/MM/yyyy") + "' and '" + ↗
145     unedate.AddDays(1).ToString("dd/MM/yyyy") + "')";
146     cnn = new SqlConnection(varglob.strconnect);
147     cnn.Open();
148     cmd = new SqlCommand(req, cnn);
149     drr = cmd.ExecuteReader();
150     while (drr.Read())
151     {
152         uneV = new vehicule(Convert.ToString(drr["immat"]), ↗
153         Convert.ToString(drr["marque"]), Convert.ToString(drr ↗
154         ["modele"]), Convert.ToString(drr["etatdeslieux"]), ↗
155         Convert.ToInt32(drr["idcategorie"]), Convert.ToInt32(drr ↗
156         ["idagence"]));
157         lesVoitures.Add(uneV);
158     }
159     drr.Close();
160     cnn.Close();
161     return (lesVoitures);
162 }
163 }
164 }
165 }
```



```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6 using System.Data.SqlClient;
7
8 namespace ADA1
9 {
10     class PasserelleVehicule
11     {
12
13         //charge une liste de voitures de l'agence par catégorie
14         public static List<vehicule> ChargerVoitures(agence uneagence,
15             categorie unecategorie)
16         {
17             vehicule uneV;
18             SqlConnection cnn;
19             SqlCommand cmd;
20             SqlDataReader drr;
21             string req;
22             List<vehicule> lesVoitures = new List<vehicule>();
23             req = "SELECT
24                 immat,marque,modele,etatdeslieux,idcategorie,idagence from
25                 vehicule where idagence=" + uneagence.getid() + " and
26                 idcategorie=" + unecategorie.getid();
27             cnn = new SqlConnection(varglob.strconnect);
28             cnn.Open();
29             cmd = new SqlCommand(req, cnn);
30             drr = cmd.ExecuteReader();
31             while (drr.Read())
32             {
33                 uneV = new vehicule(Convert.ToString(drr["immat"]),
34                     Convert.ToString(drr["marque"]), Convert.ToString(drr
35                     ["modele"]), Convert.ToString(drr["etatdeslieux"]),
36                     Convert.ToInt32(drr["idcategorie"]), Convert.ToInt32(drr
37                     ["idagence"]));
38                 lesVoitures.Add(uneV);
39             }
40             drr.Close();
41             cnn.Close();
42             return (lesVoitures);
43         }
44
45         //permet de charger les catégorie de véhicule
46         public static List<categorie> ChargerCategorie()
47         {
48             categorie uneCategorie;
49             SqlConnection cnn;
50             SqlCommand cmd;
51             SqlDataReader drr;
52             string req;
53             List<categorie> LesCategories = new List<categorie>();
```

```
49         req = "SELECT
           idcategorie,libelle1,libelle2,description,typemodele,tarifjourne
           e,tarifkm,photo,idtype from categorie";
50
51
52         cnn = new SqlConnection(varglob.strconnect);
53         cnn.Open();
54         cmd = new SqlCommand(req, cnn);
55         drr = cmd.ExecuteReader();
56         while (drr.Read())
57         {
58
59             uneCategorie = new categorie(Convert.ToInt32(drr[0]),
           Convert.ToString(drr[1]), Convert.ToString(drr[2]),
           Convert.ToString(drr[3]), Convert.ToString(drr[4]),
           Convert.ToDouble(drr[5]), Convert.ToDouble(drr[6]),
           Convert.ToString(drr[7]));
60             LesCategories.Add(uneCategorie);
61         }
62         drr.Close();
63         cnn.Close();
64         return (LesCategories);
65
66     }
67
68     //Ajoute un vehicule dans la base de données
69     public static void Ajouter_Vehicule(vehicule uneV)
70     {
71         SqlConnection cnn;
72         SqlCommand cmd;
73         string req;
74         req = "Insert into Vehicule
           (immat,marque,modele,etatdeslieux,idcategorie,idagence) values
           ('" + uneV.getimmat() + "','" + uneV.getmarque() + "','" +
           uneV.getmodele() + "','" + uneV.getetatdeslieux() + "','" +
           uneV.getidcat() + "','" + uneV.getidagence() + "')";
75         cnn = new SqlConnection(varglob.strconnect);
76         cnn.Open();
77         cmd = new SqlCommand(req, cnn);
78         cmd.ExecuteNonQuery();
79         cnn.Close();
80     }
81
82     //permet de supprimer un vehicule
83     public static void Supprimer_Vehicule(vehicule uneV)
84     {
85         SqlConnection cnn;
86         SqlCommand cmd;
87         string req;
88         req = "Delete from vehicule where immat='" + uneV.getimmat() +
           "'";
89         cnn = new SqlConnection(varglob.strconnect);
90         cnn.Open();
91         cmd = new SqlCommand(req, cnn);
92
93         cmd.ExecuteNonQuery();
```

```
94         cnn.Close();
95     }
96
97     //permet d'attribuer un index a l'idcatégorie dans le cbo
98     public static categorie getidcategorie(int unindex)
99     {
100         categorie uneCat;
101         SqlConnection cnn;
102         SqlCommand cmd;
103         SqlDataReader drr;
104         string req;
105         req = "SELECT * from categorie where idcategorie='" + unindex +
106             "'";
107         cnn = new SqlConnection(varglob.strconnect);
108         cnn.Open();
109         cmd = new SqlCommand(req, cnn);
110         drr = cmd.ExecuteReader();
111         //cmd.ExecuteNonQuery();
112         drr.Read();
113         uneCat = new categorie(Convert.ToInt32(drr["idcategorie"]),
114             Convert.ToString(drr["libelle1"]), Convert.ToString(drr
115             ["libelle2"]), Convert.ToString(drr["description"]),
116             Convert.ToString(drr["typemodele"]), Convert.ToSingle(drr
117             ["tarifjournee"]), Convert.ToSingle(drr["tarifkm"]),
118             Convert.ToString(drr["photo"]));
119         cnn.Close();
120         return uneCat;
121     }
122
123     //permet de modifier un vehicule
124     public static void UpdateVehicule(vehicule uneV)
125     {
126         SqlConnection cnn;
127         SqlCommand cmd;
128         string req;
129         req = "Update Vehicule SET marque='" + uneV.getmarque() +
130             "',modele= '" + uneV.getmodele() + "',etatdeslieux= '" +
131             uneV.getetatdeslieux() + "',idcategorie= '" + uneV.getidcat() +
132             "' where immat='" + uneV.getimmat() + "'";
133         cnn = new SqlConnection(varglob.strconnect);
134         cnn.Open();
135         cmd = new SqlCommand(req, cnn);
136         cmd.ExecuteNonQuery();
137         cnn.Close();
138     }
139 }
```

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace ADA1
8 {
9     class varglob
10    {
11
12        public static string strconnect;
13
14    }
15 }
16
```

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6 using System.Data.SqlClient;
7
8 namespace ADA1
9 {
10     class varGlobale
11     {
12         public static agence monagence;
13     }
14 }
15
```