```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SQLite;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace PimPomBro
{
    public partial class gestionPompiers : Form
    {
        // mettre true pour bypass le test de login (login = "vrichard", mdp = "mdpVero")
        bool admin = false;
        private int matricule = -1;
        string requete = "";
        SQLiteCommand cmd = null;
        SQLiteDataReader reader = null;

        public gestionPompiers()
        {
            InitializeComponent();
        }

        private void gestionPompiers_Load(object sender, EventArgs e)
        {

            picLogo.Image = Image.FromFile(@"../../Images/logo.png");
            picLogo.SizeMode = PictureBoxSizeMode.Zoom;
            picNouveau.Image = Image.FromFile(@"../../Images/jsp.jpg");

            cboCaserne.DisplayMember = "nom";
            cboCaserne.ValueMember = "id";
            cboCaserne.DataSource = MesDatas.DsGlobal.Tables["Caserne"];
            // cboCaserne.SelectedIndex = -1;

            cboCaserneDeRattachement.DisplayMember = "nom";
            cboCaserneDeRattachement.ValueMember = "id";
            cboCaserneDeRattachement.DataSource = MesDatas.DsGlobal.Tables["Caserne"].Copy();

            cboGrade.DisplayMember = "libelle";
            cboGrade.ValueMember = "code";
            cboGrade.DataSource = MesDatas.DsGlobal.Tables["Grade"];

            cboPompier.Text = "choisir pompier";
            cboPompier.SelectedIndex = -1;
        }

        private void cboPompier_Click(object sender, EventArgs e)
        {
            if (cboCaserne.SelectedItem == null)
            {
                MessageBox.Show("Veuillez selectionner une caserne pour pouvoir choisir un pompier.");
            }
        }

        private void cboCaserne_SelectedIndexChanged(object sender, EventArgs e)
        {
            lblConsigne.Text = "Veuillez selectionner un pompier.";
            cboPompier.Text = "choisir pompier";

            if (cboCaserne.SelectedItem == null) { return; }
            int idCaserne = Convert.ToInt32(cboCaserne.SelectedValue);

            DataTable tablePompiers = MesDatas.DsGlobal.Tables["Pompier"];
            DataTable tableAffectation = MesDatas.DsGlobal.Tables["Affectation"];

            List<int> listeMatricules = new List<int>();
            foreach (DataRow rowAff in tableAffectation.Rows)
            {
                if (rowAff["DateFin"].ToString() == "" && Convert.ToInt32(rowAff["idCaserne"]) == idCaserne)
                {
                    listeMatricules.Add(Convert.ToInt32(rowAff["matriculePompier"]));
                }
            }

            string filter = "matricule IN (" + string.Join(",", listeMatricules) + ")";
            DataView dv = new DataView(tablePompiers, filter, "nom ASC, prenom ASC", DataViewRowState.CurrentRows);

            cboPompier.ValueMember = "matricule";
            cboPompier.DataSource = dv;

            cboPompier.Format += (s, e2) =>
            {
                DataRowView drv = e2.ListItem as DataRowView;
                if (drv != null)
                {
                    e2.Value = drv["nom"] + " " + drv["prenom"];
                }
            };

            pnlPompier.Visible = false;
            //cboPompier.Text = "";
            cboPompier.SelectedIndex = -1;
            cboPompier.Select();
        }
        private void cboPompier_SelectedIndexChanged(object sender, EventArgs e)
        {
            if (cboPompier.SelectedItem == null || cboPompier.SelectedIndex == -1 || cboPompier.Text == "")
            {
                // MessageBox.Show("null pompier");
                return;
            }
            // MessageBox.Show("good pompier");

            pnlInformationsDetaillees.Visible = false;
            cboGrade.Enabled = false;


            matricule = Convert.ToInt32(cboPompier.SelectedValue);
            DataTable tablePompier = MesDatas.DsGlobal.Tables["Pompier"];
            DataRow rowPompier = tablePompier.Rows.Find(matricule);

            lblMatricule.Text = matricule.ToString();
            lblNom.Text = rowPompier["nom"].ToString();
            lblPrenom.Text = rowPompier["prenom"].ToString();
            lblSexe.Text = rowPompier["sexe"].ToString().ToUpper();
            lblDateEmbauche.Text = rowPompier["dateEmbauche"].ToString();
            txtGrade.Text = rowPompier["codeGrade"].ToString();
            if (rowPompier["type"].ToString() == "p") { rdbProfessionnel.Checked = true; } else { rdbVolontaire.Checked = true; }
            try
            {
                lblTelephone.Text = rowPompier["portable"].ToString();
                lblBip.Text = rowPompier["bip"].ToString();
            }
            catch (Exception ex) { }

            try
            {
                requete = "SELECT libelle FROM Grade WHERE code = @codeGrade";
                cmd = new SQLiteCommand(requete, Connexion.Connec);
                cmd.Parameters.AddWithValue("@codeGrade", txtGrade.Text);
                reader = cmd.ExecuteReader();
                if (reader.Read())
                {
                    cboGrade.Text = reader["libelle"].ToString();
                }
            } catch (Exception ex) {
                cboGrade.Visible = false;
            }

            pnlPompier.Visible = true;
        }
        private void btnModifications_Click(object sender, EventArgs e)
        {
            if (!admin) { connexionAdmin(); }
            if (!admin) { return; }

            cboCaserneDeRattachement.Text = cboCaserne.Text;


            lstHabilitations.Items.Clear();
            requete = "SELECT h.libelle,p.dateObtention FROM Passer p JOIN Habilitation h ON p.idHabilitation = h.id WHERE p.matriculePompier = @matricule";
            cmd = new SQLiteCommand(requete, Connexion.Connec);
            cmd.Parameters.AddWithValue("@matricule", matricule);
            reader = cmd.ExecuteReader();
            while (reader.Read()) {
                DateTime date = DateTime.Parse(reader["dateObtention"].ToString());
                lstHabilitations.Items.Add(date.ToString("dd/MM/yyyy") + " - " + reader["libelle"]);
            }

            lstAffectations.Items.Clear();
            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";
            cmd = new SQLiteCommand(requete, Connexion.Connec);
            cmd.Parameters.AddWithValue("@matricule", matricule);
            reader = cmd.ExecuteReader();
            while (reader.Read())
            {
                string caserne = reader["nom"].ToString();
                DateTime dA = DateTime.Parse(reader["dateA"].ToString());
                string dateA = dA.ToString("dd/MM/yyyy");
                DateTime dF = DateTime.Parse(reader["dateFin"].ToString());
                string dateFin = dF.ToString("dd/MM/yyyy");
                lstAffectations.Items.Add(caserne + " : " + dateA + " - " + dateFin);
            }


            requete = "SELECT enConge FROM Pompier WHERE matricule = @matricule";
            cmd = new SQLiteCommand(requete, Connexion.Connec);
            cmd.Parameters.AddWithValue("@matricule", matricule);
            reader = cmd.ExecuteReader();
            reader.Read();
            // MessageBox.Show(Convert.ToInt32(reader["enConge"]).ToString());
            if (Convert.ToInt32(reader["enConge"]) == 1)
            {
                chkConge.Checked = true;
            } else
            {
                chkConge.Checked = false;
            }

            reader.Close();
            cboGrade.Enabled = true;
            pnlInformationsDetaillees.Visible = true;
        }

        private void btnAppliquerModif_Click(object sender, EventArgs e)
        {
            using (SQLiteTransaction tran = Connexion.Connec.BeginTransaction())
            {
                try
                {
                    requete = "UPDATE Pompier SET codeGrade = @codeGrade, enConge = @enConge WHERE matricule = @matricule";
                    cmd = new SQLiteCommand(requete, Connexion.Connec, tran);
                    cmd.Parameters.AddWithValue("@codeGrade", txtGrade.Text);
                    cmd.Parameters.AddWithValue("@matricule", matricule);
                    cmd.Parameters.AddWithValue("@enConge", (chkConge.Checked) ? 1 : 0);
                    cmd.ExecuteNonQuery();

                    requete = "UPDATE Affectation SET dateFin = @dateFin WHERE matriculePompier = @matricule AND (dateFin IS NULL or dateFin = '')";
                    cmd = new SQLiteCommand(requete, Connexion.Connec, tran);
                    cmd.Parameters.AddWithValue("@matricule", matricule);
                    cmd.Parameters.AddWithValue("@dateFin", DateTime.Now);
                    cmd.ExecuteNonQuery();

                    requete = "INSERT INTO Affectation VALUES (@matricule, @dateA, NULL,@caserne)";
                    cmd = new SQLiteCommand(requete, Connexion.Connec, tran);
                    cmd.Parameters.AddWithValue("@matricule", matricule);
                    cmd.Parameters.AddWithValue("@dateA", DateTime.Now);
                    cmd.Parameters.AddWithValue("@caserne", cboCaserneDeRattachement.SelectedValue);
                    cmd.ExecuteNonQuery();

                    tran.Commit();
                }
                catch (Exception ex)
                {
                    tran.Rollback();
                    MessageBox.Show(ex.Message);
                    return;
                }
            }
            MesDatas.refreshTable("Pompier");
            MesDatas.refreshTable("Affectation");

            this.gestionPompiers_Load(sender, e);
            this.cboCaserne_SelectedIndexChanged(sender, e);
        }

        private void cboGrade_SelectedIndexChanged(object sender, EventArgs e)
        {
            requete = "SELECT code FROM Grade WHERE libelle = @libelle";
            cmd = new SQLiteCommand(requete, Connexion.Connec);
            cmd.Parameters.AddWithValue("@libelle", cboGrade.Text);
            reader = cmd.ExecuteReader();
            if (reader.Read())
            {
                txtGrade.Text = reader["code"].ToString();
            }

            reader.Close();
        }
        private void connexionAdmin()
        {
            ConnexionAdmin connexionAdmin = new ConnexionAdmin();
            var result = connexionAdmin.ShowDialog();
            if (result == DialogResult.OK) {
                admin = true;
            } else
            {
                admin = false;
            }

        }

        private void btnNouveau_Click(object sender, EventArgs e)
        {
            // si l'utilisateur n'est pas admin on lui demande de ce connecter
            if (!admin) { connexionAdmin(); }
            // s'il n'est toujours pas connecté on quitte la fonction
            if (!admin) { return; }

            CreationPompier creationPompier = new CreationPompier();
            creationPompier.ShowDialog();

            this.gestionPompiers_Load(sender, e);
            this.cboCaserne_SelectedIndexChanged(sender, e);
        }

        private void btnAnnuler_Click(object sender, EventArgs e)
        {
            cboPompier_SelectedIndexChanged(sender, e);
            btnModifications_Click(sender, e);
        }

        private void button1_Click(object sender, EventArgs e)
        {
            this.Close();
        }
```

```
    }
}

```