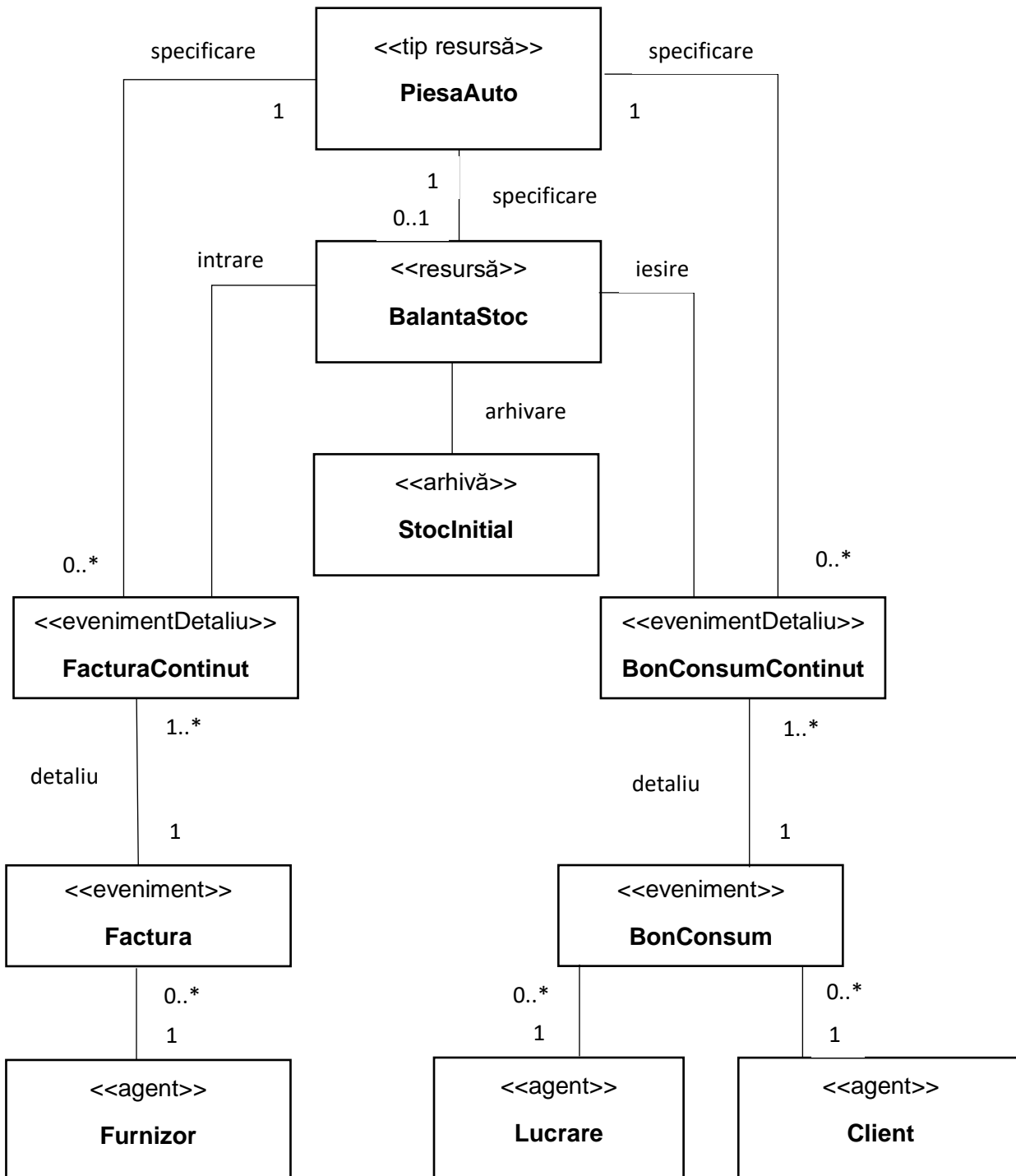


## Proiect MPP

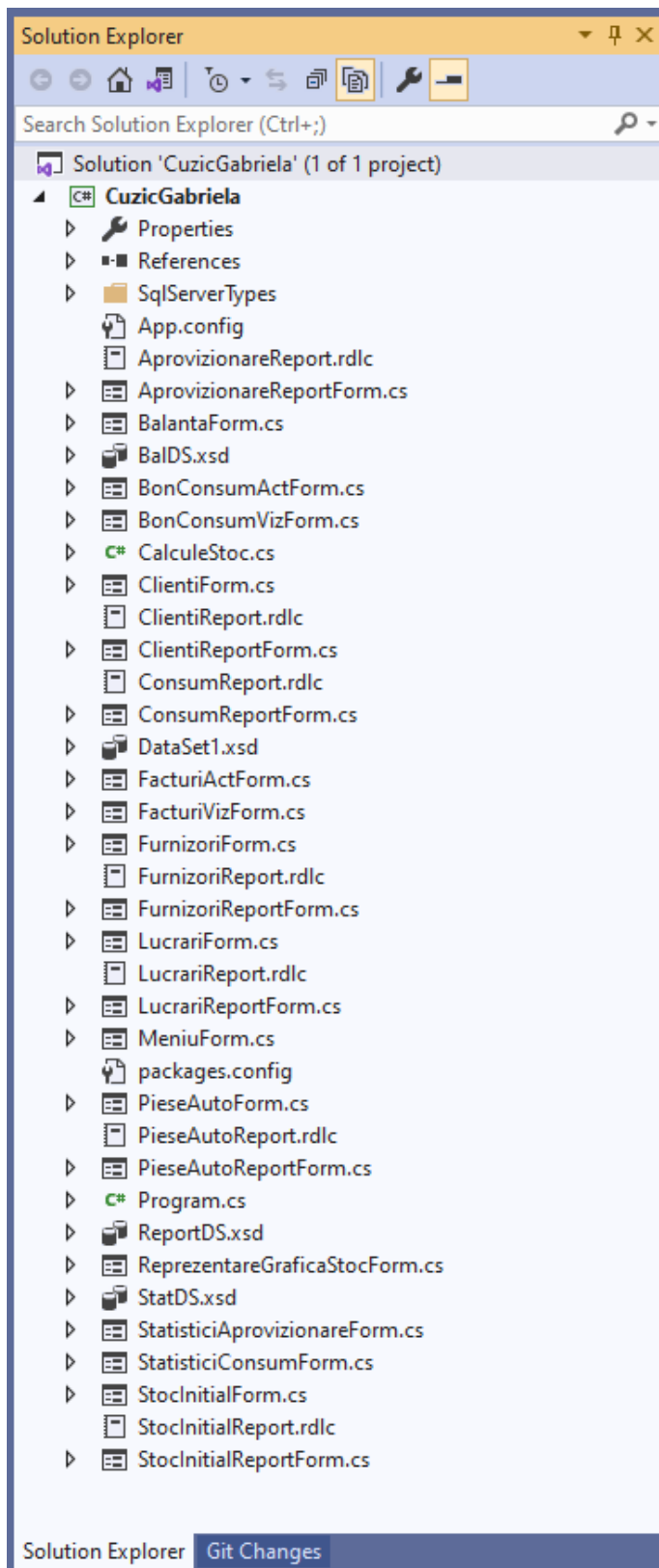
Aplicatie pentru evidenta pieselor dintr-un service auto

Cuzic Gabriela info3 – grupa 2

# 1. Diagrama REA



## 2. Structura aplicației



### 3. Form-uri

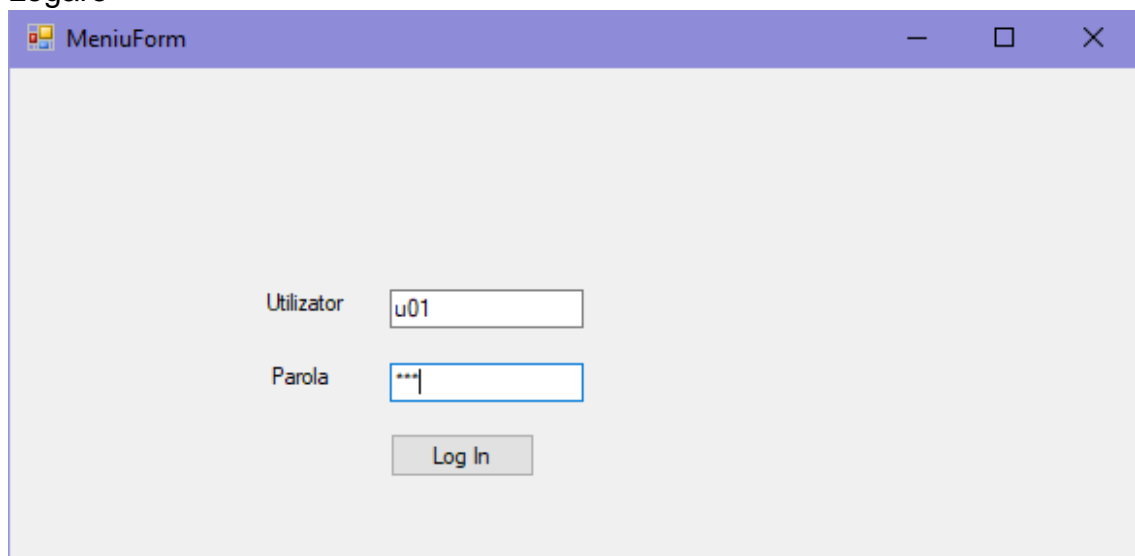
Am folosit form-uri pentru vizualizarea, actualizarea și ștergerea înregistrărilor modelelor folosite, utilizând dataSet-ul **DataSet1.xsd**.

De asemenea, am folosit form-uri pentru vizualizarea acestora în **rapoarte** cu tableAdaptare parametrizate, folosind dataSet-ul **ReportDS.xsd**, dar și form-uri pentru a **reprezenta grafic** statistici legate de evaluarea resurselor și evenimente de aprovizionare și consum, prin intermediul dataSet-ului **StatDS.xsd**.

#### 3.0. Meniu

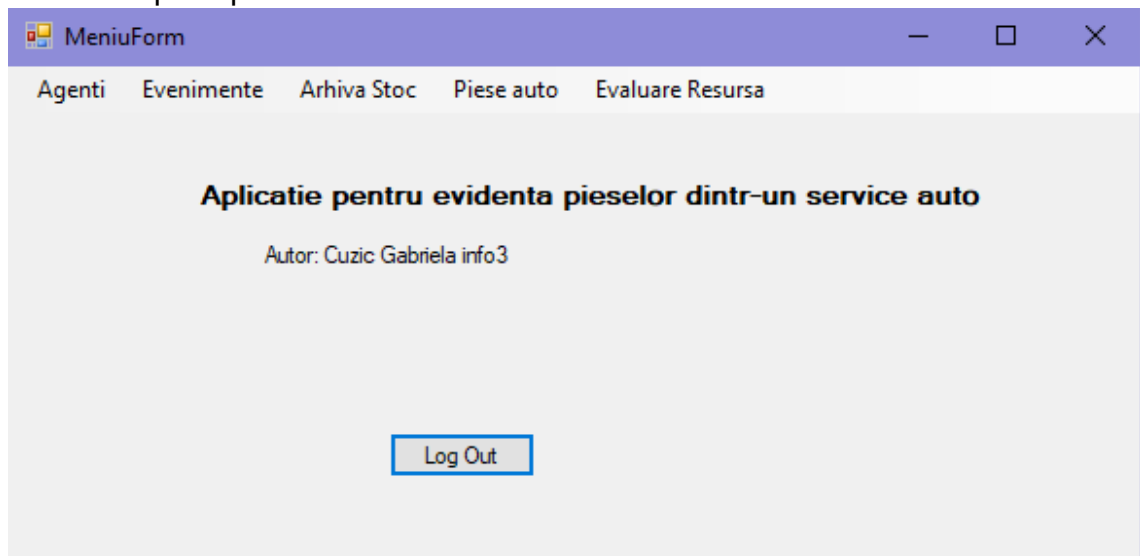
##### MeniuForm.cs

Logare



The screenshot shows a Windows application window titled "MeniuForm". Inside the window, there is a login form with two text boxes: "Utilizator" (User) containing the text "u01" and "Parola" (Password) containing three asterisks "\*\*\*". Below the password box is a "Log In" button.

Fereastra principala



The screenshot shows the main application window titled "MeniuForm". It features a menu bar with five items: "Agenti", "Evenimente", "Arhiva Stoc", "Piese auto", and "Evaluare Resursa". Below the menu bar, the text "Aplicatie pentru evidenta pieselor dintr-un service auto" is displayed, followed by "Autor: Cuzic Gabriela info3". At the bottom center, there is a "Log Out" button.

## Codul:

```
using System;
using System.Windows.Forms;
using System.Data.OleDb;

namespace CuzicGabriela
{
    public partial class MeniuForm : Form
    {
        private OleDbConnection con = new OleDbConnection();
        private OleDbCommand cmd = new OleDbCommand();
        private OleDbDataReader rdr;
        public MeniuForm()
        {
            InitializeComponent();

            private void A1(Boolean v)
            {
                menuStrip1.Visible = !v;
                lblTitlu.Visible = !v;
                lblAutor.Visible = !v;
                lblParola.Visible = v;
                txtParola.Visible = v;
                lblUtilizator.Visible = v;
                txtUtilizator.Visible = v;
                if (v)
                {
                    btnStart.Text = "Log In";
                    txtUtilizator.Text = "";
                    txtParola.Text = "";
                }
                else btnStart.Text = "Log Out";
            }

            private Boolean Logare_OK()
            {
                if (txtUtilizator.Text == "")
                {
                    MessageBox.Show("Completati utilizator !");
                    txtUtilizator.Focus();
                    return false;
                }
                if (txtParola.Text == "")
                {
                    MessageBox.Show("Completati parola !");
                    txtParola.Focus();
                    return false;
                }
                con.ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;" +
                    "Data Source=D:\\Medii de proiectare\\Proiect sem2\\BD-ServiceAuto.mdb";
                cmd.Connection = con;
                cmd.CommandText = "Select username,parola from Conturi " +
                    "where username='" + txtUtilizator.Text + "'";
                con.Open();
                rdr = cmd.ExecuteReader();
            }
        }
    }
}
```

```

        if (rdr.Read())
        {
            if (txtParola.Text != rdr.GetString(1))
            {
                MessageBox.Show("Parola eronata");
                txtParola.Focus();
                con.Close();
                return false;
            }
            con.Close();
            return true;
        }
        else
        {
            MessageBox.Show("Utilizator eronat");
            txtUtilizator.Focus();
            con.Close();
            return false;
        }
    }
}

```

e) `private void aprovizionareToolStripMenuItem_Click(object sender, EventArgs`

```

{
    FacturiVizForm f = new FacturiVizForm();
    f.ShowDialog();
}

```

```

private void consumToolStripMenuItem_Click(object sender, EventArgs e)
{
    BonConsumVizForm f = new BonConsumVizForm();
    f.ShowDialog();
}

```

```

private void clientiToolStripMenuItem_Click(object sender, EventArgs e)
{
    ClientiForm f = new ClientiForm();
    f.ShowDialog();
}

```

```

private void furnizoriToolStripMenuItem_Click(object sender, EventArgs e)
{
    FurnizoriForm f = new FurnizoriForm();
    f.ShowDialog();
}

```

```

private void lucrariToolStripMenuItem1_Click(object sender, EventArgs e)
{
    LucrariForm f = new LucrariForm();
    f.ShowDialog();
}

```

e) `private void tipuriResursaToolStripMenuItem_Click(object sender, EventArgs`

```

{
    PieseAutoForm f = new PieseAutoForm();
    f.ShowDialog();
}

```

```

private void arhivaStocToolStripMenuItem_Click(object sender, EventArgs e)
{
    StocInitialForm f = new StocInitialForm();
    f.ShowDialog();
}

private void MeniuForm_Load(object sender, EventArgs e)
{
    A1(true);
}

e) private void CalculResursaToolStripMenuItem_Click(object sender, EventArgs
{
    BalantaForm f = new BalantaForm();
    f.ShowDialog();
}

private void btnStart_Click(object sender, EventArgs e)
{
    if (btnStart.Text == "Log In")
    {
        if (Logare_OK()) A1(false);
    }
    else A1(true);
}
}
}

```

### 3.1. Agenti

#### ClientiForm.cs

#### Vizualizare

The screenshot shows a Windows application window titled "Agenti". At the top, there are three buttons: "Adaugare", "Actualizare", and "Stergere". Below these, the section "Clienti" is displayed with a "Raport Clienti" button. A table lists clients with columns "Nume Client" and "Nr Telefon". The first row, "Costea Marian" with phone number "0763363043", is selected. To the right of the table, there are three input fields for "Nume", "Nr Telefon", and "ID", with values "Costea Marian", "0763363043", and "2" respectively.

| Nume Client      | Nr Telefon |
|------------------|------------|
| Costea Marian    | 0763363043 |
| Popescu Sorin    | 0761168904 |
| Arsenie Ruxandra | 0738324674 |
| Mirea Daniel     | 0739024674 |
| Luca Eugen       | 0738321234 |
| Craiu Lucian     | 0735678674 |
| Dinescu Adrian   | 0734024674 |
| Manea Iulia      | 0736923674 |
| Oltan Emanuel    | 0738231674 |

Nume: Costea Marian

Nr Telefon: 0763363043

ID: 2

## Actualizare

Agents

Adaugare

Actualizare

Stergere

Salvare

Renuntare

Cienti

Raport Cienti

|   | Nume Client      | Nr Telefon |
|---|------------------|------------|
|   | Costea Marian    | 0763363043 |
|   | Popescu Sorin    | 0761168904 |
| ▶ | Arsenie Ruxandra | 0738324674 |
|   | Mirea Daniel     | 0739024674 |
|   | Luca Eugen       | 0738321234 |
|   | Craiu Lucian     | 0735678674 |
|   | Dinescu Adrian   | 0734024674 |
|   | Manea Iulia      | 0736923674 |
|   | Oltean Emanuel   | 0738231674 |

ACTUALIZARE

Nume

Arsenie Ruxandra

Nr Telefon

0738324674

ID

4

## Adăugare

Agents

Adaugare

Actualizare

Stergere

Salvare

Renuntare

Cienti

Raport Cienti

|   | Nume Client      | Nr Telefon |
|---|------------------|------------|
|   | Costea Marian    | 0763363043 |
|   | Popescu Sorin    | 0761168904 |
| ▶ | Arsenie Ruxandra | 0738324674 |
|   | Mirea Daniel     | 0739024674 |
|   | Luca Eugen       | 0738321234 |
|   | Craiu Lucian     | 0735678674 |
|   | Dinescu Adrian   | 0734024674 |
|   | Manea Iulia      | 0736923674 |
|   | Oltean Emanuel   | 0738231674 |

ADAugARE

Nume

Nr Telefon

ID



Codul:

```
using System;
using System.Data.OleDb;
using System.Windows.Forms;

namespace CuzicGabriela
{
    public partial class ClientiForm : Form
    {
        public ClientiForm()
        {
            InitializeComponent();

            private void ClientiForm_Load(object sender, EventArgs e)
            {
                A1();
            }

            private void A1()
            {
                //Umple cu date obiectele DataTable: Clienti (din DataSet):
                clientiTableAdapter.Fill(dataSet1.Clienti);
                //Protectie grid
                dataGridView1.AllowUserToAddRows = false;
                dataGridView1.AllowUserToDeleteRows = false;
                dataGridView1.ReadOnly = true;
                //Protectie txtIdClient
                txtIdClient.ReadOnly = true;
                A3();
            }

            private void A2()
            {
                //Configurare butoane
                configureazaButoane(false);
                //Dezlegare controale Panel
                legareControale(false);
                //Ridicare protectie controale Panel
                protectiePanel(false);
                //Modificare lblOp
                lblOp.Text = "ADAUGARE";
                //Pozitionare cursor pe primul camp
                txtDClient.Focus();
                // Golire campuri
                golireCampuri();
            }

            private void A3()
            {
                //Initializare lblOp
                lblOp.Text = "";
                //Configurare(butoane)
                configureazaButoane(true);
                //Protectie componente Panel
                protectiePanel(true);
                //Legare controale
                legareControale(true);
            }
        }
    }
}
```

```

}
private void A4()
{
    if (lblOp.Text == "ADAUGARE")
    {
        if (!validareCampuriObligatorii()) return;
        adauga_inregistrare();
        golireCampuri();
        //Pune cursor pe primul camp
        txtDClient.Focus();
        refresh_grid(clientiBindingSource.Position);
    }
    else if (lblOp.Text == "ACTUALIZARE")
    {
        if (!validareCampuriObligatorii()) return;
        modifica_inregistrare();
        refresh_grid(clientiBindingSource.Position);
        A3();
    }
    else
        MessageBox.Show("Operatie actualizare neefectuata");
}

private void A5(TextBox txtB)
{
    decimal p;
    OleDbConnection con = new OleDbConnection();
    OleDbCommand cmd = new OleDbCommand();
    OleDbDataReader r;
    if (lblOp.Text == "") return;
    if (txtB.Text == "") return;
    if (btnRenuntare.Focused) return;

    if (txtB == txtNrTelefon)
    {
        // Validare numericitate
        try { p = Convert.ToDecimal(txtB.Text); }
        catch { MessageBox.Show("Format eronat"); txtB.Focus(); }
        con.ConnectionString =
clientiTableAdapter.Connection.ConnectionString;
        cmd.Connection = con;
        if (lblOp.Text == "ADAUGARE")
        {
            cmd.CommandText = "Select DClient From Clienti where
NrTelefon='" + txtNrTelefon.Text + "'";
            con.Open();
            r = cmd.ExecuteReader();
            if (r.Read())
            {
                MessageBox.Show("Număr de telefon deja existent");
                txtNrTelefon.Focus();
            }
            con.Close();
        }
        else if (lblOp.Text == "ACTUALIZARE")
        {
            cmd.CommandText = "Select DClient From Clienti where
NrTelefon='" + txtNrTelefon.Text +

```

```

        "" and IdClient <> " + txtIdClient.Text;
        con.Open();
        r = cmd.ExecuteReader();
        if (r.Read())
        {
            MessageBox.Show("Număr de telefon deja existent");
            txtNrTelefon.Focus();
        }
        con.Close();
    }
}

private void A7()
{
    //Configurare butoane
    configureazaButoane(false);
    //Dezlegare controale Panel
    legareControale(false);
    //Ridicare protectie controale Panel
    protectiePanel(false);
    //Modificare lblOp
    lblOp.Text = "ACTUALIZARE";
    //Pozitionare cursor pe primul camp
    txtDClient.Focus();
}

private void A8()
{
    OleDbConnection con = new OleDbConnection();
    OleDbCommand cmd = new OleDbCommand();
    OleDbDataReader r;
    con.ConnectionString =
clientiTableAdapter.Connection.ConnectionString;
    cmd.Connection = con;
    cmd.CommandText = "Select IdBonConsum From BonuriConsum where
IdClient=" + txtIdClient.Text;
    con.Open();
    r = cmd.ExecuteReader();
    if (r.Read())
    {
        MessageBox.Show("Client referit in tabela BonuriConsum! Nu se
poate sterge!");
        con.Close();
        return;
    }
    con.Close();
    con.ConnectionString =
clientiTableAdapter.Connection.ConnectionString;
    cmd.Connection = con;
    cmd.CommandText = "Delete From Clienti Where IdClient = " +
txtIdClient.Text;
    //MessageBox.Show(cmd.CommandText);
    con.Open();
    cmd.ExecuteNonQuery();
    con.Close();
    refresh_grid(clientiBindingSource.Position);
}

```

```

private void configureazaButoane(bool v)
{
    btnRenuntare.Visible = !v;
    btnSalvare.Visible = !v;
    btnAdaugare.Enabled = v;
    btnActualizare.Enabled = v;
    btnStergere.Enabled = v;
}
private void legareControale(bool v)
{
    if (v)
    {
        txtDClient.DataBindings.Add("Text", clientiBindingSource,
"DClient");
        txtNrTelefon.DataBindings.Add("Text", clientiBindingSource,
"NrTelefon");
        txtIdClient.DataBindings.Add("Text", clientiBindingSource,
"IdClient");
    }
    else
    {
        txtDClient.DataBindings.Clear();
        txtNrTelefon.DataBindings.Clear();
        txtIdClient.DataBindings.Clear();
    }
}
private void protectiePanel(bool v)
{
    txtDClient.ReadOnly = v;
    txtNrTelefon.ReadOnly = v;
}
private void golireCampuri()
{
    txtDClient.Text = "";
    txtNrTelefon.Text = "";
    txtIdClient.Text = "";
}
private bool validareCampuriObligatorii()
{
    //Validare de completare obligatorie campurile: Nume, NrTelefon
    if (txtDClient.Text == "")
    {
        MessageBox.Show("Completati Numele clientului !");
        txtDClient.Focus();
        return false;
    }
    if (txtNrTelefon.Text == "")
    {
        MessageBox.Show("Completati numărul de telefon !");
        txtNrTelefon.Focus();
        return false;
    }
    return true;
}

private void adauga_inregistrare()
{
    string listaCampuri;

```

```

        string listaValori;
        OleDbConnection con = new OleDbConnection();
        OleDbCommand cmd = new OleDbCommand();
        con.ConnectionString =
clientiTableAdapter.Connection.ConnectionString;
        cmd.Connection = con;
        listaCampuri = "DClient, NrTelefon";
        listaValori = "'" + txtDClient.Text + "'" +
        ",'" + txtNrTelefon.Text + "'";
        cmd.CommandText = "Insert into Clienti(" + listaCampuri + ") " +
        "Select " + listaValori;
        //MessageBox.Show("Client adaugat cu succes!");
        con.Open();
        cmd.ExecuteNonQuery();
        con.Close();
    }
    private void refresh_grid(int p)
    {
        clientiTableAdapter.Fill(dataSet1.Clienti);
        clientiBindingSource.Position = p;
    }

    private void modifica_inregistrare()
    {
        string listaSet;
        OleDbConnection con = new OleDbConnection();
        OleDbCommand cmd = new OleDbCommand();
        con.ConnectionString =
clientiTableAdapter.Connection.ConnectionString;
        cmd.Connection = con;
        listaSet = "DClient = '" + txtDClient.Text + "'," +
        "NrTelefon = '" + txtNrTelefon.Text + "'";
        cmd.CommandText = "Update Clienti Set " + listaSet + " Where
IdClient=" +
        txtIdClient.Text;
        //MessageBox.Show("Actualizare realizata cu succes!");
        con.Open();
        cmd.ExecuteNonQuery();
        con.Close();
    }

    private void btnActualizare_Click(object sender, EventArgs e)
    {
        A7();
    }

    private void btnSalvare_Click(object sender, EventArgs e)
    {
        A4();
    }

    private void btnRenuntare_Click(object sender, EventArgs e)
    {
        A3();
    }

    private void btnAdaugare_Click(object sender, EventArgs e)
    {

```

```

        A2();
    }

    private void btnStergere_Click(object sender, EventArgs e)
    {
        const string mesaj = "Confirmati stergerea";
        const string titlu = "Stergere inregistrare";
        var rezultat = MessageBox.Show(mesaj, titlu, MessageBoxButtons.YesNo,
        MessageBoxIcon.Warning);
        if (rezultat == DialogResult.No) return;
        A8();
    }

    private void txtNrTelefon_Leave(object sender, EventArgs e)
    {
        A5(txtNrTelefon);
    }

    private void btnRaportClienti_Click(object sender, EventArgs e)
    {
        ClientiReportForm f = new ClientiReportForm();
        f.ShowDialog();
    }
}

```

## ClientiReportForm.cs

Nume Client

1 of 1

Lista Clientilor ce contin "escu" in nume

| Nume Client    | Nr Telefon |
|----------------|------------|
| Popescu Sorin  | 0761168904 |
| Dinescu Adrian | 0734024674 |

Codul:

```
using Microsoft.Reporting.WinForms;
using System;
using System.Windows.Forms;

namespace CuzicGabriela
{
    public partial class ClientiReportForm : Form
    {
        public ClientiReportForm()
        {
            InitializeComponent();

            private void ClientiReportForm_Load(object sender, EventArgs e)
            {
                // TODO: This line of code loads data into the 'ReportDS.Clienti'
                table. You can move, or remove it, as needed.
                //this.ClientiTableAdapter.Fill(this.ReportDS.Clienti);

                //this.reportViewer1.RefreshReport();
            }

            private void btnAfiseaza_Click(object sender, EventArgs e)
            {
                this.ClientiTableAdapter.Fill(this.ReportDS.Clienti,
                txtDClient.Text.ToString());
                //Pregatesc parametrii raportului
                ReportParameter[] parameters = new ReportParameter[1];
                if(txtDClient.Text!="")
                    parameters[0] = new ReportParameter("Titlu", "Lista Clientilor ce
                contin \"" + txtDClient.Text.ToString() + "\" in nume");
                else parameters[0] = new ReportParameter("Titlu", "Lista Clientilor "
                + txtDClient.Text.ToString());

                //Comunicare parametrii
                this.reportViewer1.LocalReport.SetParameters(parameters);

                //Emit raport
                this.reportViewer1.RefreshReport();
            }
        }
    }
}
```

## FurnizoriForm.cs

### Vizualizare

FurnizoriForm

Adaugare Actualizare Stergere

Furnizori Raport Furnizori

|   | Denumire Furnizor              | NrTelefon  |
|---|--------------------------------|------------|
| ▶ | AutoKama SRL                   | 0246286112 |
|   | SC Euro Parts Distribution SRL | 0246286745 |
|   | TecDoc SRL                     | 0246286926 |
|   | PieseAuto Marketplace SRL      | 0243876100 |
|   | SC E-Autoparts Europa SRL      | 0246992112 |

Denumire AutoKama SRL

Nr Telefon 0246286112

ID 1

### Actualizare

FurnizoriForm

Adaugare Actualizare Stergere Salvare Renuntare

Furnizori Raport Furnizori

|   | Denumire Furnizor              | NrTelefon  |
|---|--------------------------------|------------|
|   | AutoKama SRL                   | 0246286112 |
| ▶ | SC Euro Parts Distribution SRL | 0246286745 |
|   | TecDoc SRL                     | 0246286926 |
|   | PieseAuto Marketplace SRL      | 0243876100 |
|   | SC E-Autoparts Europa SRL      | 0246992112 |

ACTUALIZARE

Denumire SC Euro Parts Distribution SRL

Nr Telefon 0246286745

ID 2



## Adăugare

The screenshot shows a Windows application window titled "FurnizoriForm". At the top, there are five buttons: "Adaugare", "Actualizare", "Stergere", "Salvare", and "Renuntare". Below these, the window is divided into two main sections. The left section, titled "Furnizori", contains a button "Raport Furnizori" and a table with two columns: "Denumire Furnizor" and "NrTelefon". The table lists five suppliers, with "SC Euro Parts Distribution SRL" selected. The right section, titled "ADAUGARE", contains three input fields labeled "Denumire", "Nr Telefon", and "ID".

|   | Denumire Furnizor              | NrTelefon  |
|---|--------------------------------|------------|
|   | AutoKama SRL                   | 0246286112 |
| ▶ | SC Euro Parts Distribution SRL | 0246286745 |
|   | TecDoc SRL                     | 0246286926 |
|   | PieseAuto Marketplace SRL      | 0243876100 |
|   | SC E-Autoparts Europa SRL      | 0246992112 |

Codul:

```
using System;
using System.Windows.Forms;
using System.Data.OleDb;

namespace CuzicGabriela
{
    public partial class FurnizoriForm : Form
    {
        public FurnizoriForm()
        {
            InitializeComponent();
        }

        private void FurnizoriForm_Load(object sender, EventArgs e)
        {
            A1();
        }

        private void A1()
        {
            //Umple cu date obiectele DataTable: Furnizori (din DataSet):
            furnizoriTableAdapter.Fill(dataSet1.Furnizori);
            //Protectie grid
            dataGridView1.AllowUserToAddRows = false;
            dataGridView1.AllowUserToDeleteRows = false;
            dataGridView1.ReadOnly = true;
            //Protectie txtIdFurnizor
        }
    }
}
```

```

        txtIdFurnizor.ReadOnly = true;
        A3();
    }
    private void A2()
    {
        //Configurare butoane
        configureazaButoane(false);
        //Dezlegare controale Panel
        legareControale(false);
        //Ridicare protectie controale Panel
        protectiePanel(false);
        //Modificare lblOp
        lblOp.Text = "ADAUGARE";
        //Pozitionare cursor pe primul camp
        txtDFurnizor.Focus();
        // Golire campuri
        golireCampuri();
    }

    private void A3()
    {
        //Initializare lblOp
        lblOp.Text = "";
        //Configurare(butoane)
        configureazaButoane(true);
        //Protectie componente Panel
        protectiePanel(true);
        //Legare controale
        legareControale(true);
    }
    private void A4()
    {
        if (lblOp.Text == "ADAUGARE")
        {
            if (!validareCampuriObligatorii()) return;
            adauga_inregistrare();
            golireCampuri();
            //Pune cursor pe primul camp
            txtDFurnizor.Focus();
            refresh_grid(furnizoriBindingSource.Position);
        }
        else if (lblOp.Text == "ACTUALIZARE")
        {
            if (!validareCampuriObligatorii()) return;
            modifica_inregistrare();
            refresh_grid(furnizoriBindingSource.Position);
            A3();
        }
        else
            MessageBox.Show("Operatie actualizare neefectuata");
    }

    private void A5(TextBox txtB)
    {
        decimal p;
        OleDbConnection con = new OleDbConnection();
        OleDbCommand cmd = new OleDbCommand();
        OleDbDataReader r;
    }

```

```

        if (lblOp.Text == "") return;
        if (txtB.Text == "") return;
        if (btnRenuntare.Focused) return;

        if (txtB == txtNrTelefon)
        {
            // Validare numericitate
            try { p = Convert.ToDecimal(txtB.Text); }
            catch { MessageBox.Show("Format eronat"); txtB.Focus(); }
            con.ConnectionString =
furnizoriTableAdapter.Connection.ConnectionString;
            cmd.Connection = con;
            if (lblOp.Text == "ADAUGARE")
            {
                cmd.CommandText = "Select DFurnizor From Furnizori where
NrTelefon='" + txtNrTelefon.Text + "'";
                con.Open();
                r = cmd.ExecuteReader();
                if (r.Read())
                {
                    MessageBox.Show("Număr de telefon deja existent");
                    txtNrTelefon.Focus();
                }
                con.Close();
            }
            else if (lblOp.Text == "ACTUALIZARE")
            {
                cmd.CommandText = "Select DFurnizor From Furnizori where
NrTelefon='" + txtNrTelefon.Text +
                "' and IdFurnizor <> " + txtIdFurnizor.Text;
                con.Open();
                r = cmd.ExecuteReader();
                if (r.Read())
                {
                    MessageBox.Show("Număr de telefon deja existent");
                    txtNrTelefon.Focus();
                }
                con.Close();
            }
        }
    }

    private void A7()
    {
        //Configurare butoane
        configureazaButoane(false);
        //Dezlegare controale Panel
        legareControale(false);
        //Ridicare protectie controale Panel
        protectiePanel(false);
        //Modificare lblOp
        lblOp.Text = "ACTUALIZARE";
        //Pozitionare cursor pe primul camp
        txtDFurnizor.Focus();
    }

    private void A8()
    {
        OleDbConnection con = new OleDbConnection();

```

```

        OleDbCommand cmd = new OleDbCommand();
        OleDbDataReader r;
        con.ConnectionString =
furnizoriTableAdapter.Connection.ConnectionString;
        cmd.Connection = con;
        cmd.CommandText = "Select IdFactura From Facturi where IdFurnizor=" +
txtIdFurnizor.Text;
        con.Open();
        r = cmd.ExecuteReader();
        if (r.Read())
        {
            MessageBox.Show("Furnizor referit in tabela Facturi! Nu se poate
sterge!");
            con.Close();
            return;
        }
        con.Close();
        con.ConnectionString =
furnizoriTableAdapter.Connection.ConnectionString;
        cmd.Connection = con;
        cmd.CommandText = "Delete From Furnizori Where IdFurnizor = " +
txtIdFurnizor.Text;
        //MessageBox.Show("Inregistrare stearsa cu succes!");
        con.Open();
        cmd.ExecuteNonQuery();
        con.Close();
        refresh_grid(furnizoriBindingSource.Position);
    }

    private void configureazaButoane(bool v)
    {
        btnRenuntare.Visible = !v;
        btnSalvare.Visible = !v;
        btnAdaugare.Enabled = v;
        btnActualizare.Enabled = v;
        btnStergere.Enabled = v;
    }

    private void legareControale(bool v)
    {
        if (v)
        {
            txtDFurnizor.DataBindings.Add("Text", furnizoriBindingSource,
"Dfurnizor");
            txtNrTelefon.DataBindings.Add("Text", furnizoriBindingSource,
"NrTelefon");
            txtIdFurnizor.DataBindings.Add("Text", furnizoriBindingSource,
"IdFurnizor");
        }
        else
        {
            txtDFurnizor.DataBindings.Clear();
            txtNrTelefon.DataBindings.Clear();
            txtIdFurnizor.DataBindings.Clear();
        }
    }

    private void protectiePanel(bool v)
    {
        txtDFurnizor.ReadOnly = v;
    }

```

```

        txtNrTelefon.ReadOnly = v;
    }
    private void golireCampuri()
    {
        txtDFurnizor.Text = "";
        txtNrTelefon.Text = "";
        txtIdFurnizor.Text = "";
    }
    private bool validareCampuriObligatorii()
    {
        //Validare de completare obligatorie campurile: Nume, NrTelefon
        if (txtDFurnizor.Text == "")
        {
            MessageBox.Show("Completati Numele furnizorului !");
            txtDFurnizor.Focus();
            return false;
        }
        if (txtNrTelefon.Text == "")
        {
            MessageBox.Show("Completati numărul de telefon !");
            txtNrTelefon.Focus();
            return false;
        }
        return true;
    }

    private void adauga_inregistrare()
    {
        string listaCampuri;
        string listaValori;
        OleDbConnection con = new OleDbConnection();
        OleDbCommand cmd = new OleDbCommand();
        con.ConnectionString =
furnizoriTableAdapter.Connection.ConnectionString;
        cmd.Connection = con;
        listaCampuri = "DFurnizor, NrTelefon";
        listaValori = "'" + txtDFurnizor.Text + "'" +
        ", '" + txtNrTelefon.Text + "'";
        cmd.CommandText = "Insert into Furnizori(" + listaCampuri + ") " +
        "Select " + listaValori;
        //MessageBox.Show("Adaugare efectuata cu succes!");
        con.Open();
        cmd.ExecuteNonQuery();
        con.Close();
    }

    private void refresh_grid(int p)
    {
        furnizoriTableAdapter.Fill(dataSet1.Furnizori);
        furnizoriBindingSource.Position = p;
    }

    private void modifica_inregistrare()
    {
        string listaSet;
        OleDbConnection con = new OleDbConnection();
        OleDbCommand cmd = new OleDbCommand();
        con.ConnectionString =
furnizoriTableAdapter.Connection.ConnectionString;

```

```

        cmd.Connection = con;
        listaSet = "DFurnizor = '" + txtDFurnizor.Text + "'," +
        "NrTelefon = '" + txtNrTelefon.Text + "'";
        cmd.CommandText = "Update Furnizori Set " + listaSet + " Where
IdFurnizor=" +
        txtIdFurnizor.Text;
        //MessageBox.Show("Actualizare efectuata cu succes!");
        con.Open();
        cmd.ExecuteNonQuery();
        con.Close();
    }

    private void btnAdaugare_Click(object sender, EventArgs e)
    {
        A2();
    }

    private void btnActualizare_Click(object sender, EventArgs e)
    {
        A7();
    }

    private void btnStergere_Click(object sender, EventArgs e)
    {
        const string mesaj = "Confirmati stergerea";
        const string titlu = "Stergere inregistrare";
        var rezultat = MessageBox.Show(mesaj, titlu, MessageBoxButtons.YesNo,
        MessageBoxIcon.Warning);
        if (rezultat == DialogResult.No) return;
        A8();
    }

    private void btnSalvare_Click(object sender, EventArgs e)
    {
        A4();
    }

    private void btnRenuntare_Click(object sender, EventArgs e)
    {
        A3();
    }

    private void txtNrTelefon_Leave(object sender, EventArgs e)
    {
        A5(txtNrTelefon);
    }

    private void btnRaportFurnizori_Click(object sender, EventArgs e)
    {
        FurnizoriReportForm f = new FurnizoriReportForm();
        f.ShowDialog();
    }
}
}

```

## FurnizoriReportForm.cs

Denumire Furnizor SC Afiseaza raport

Lista Furnizorilor ce contin "SC" in nume

| Denumire Furnizor              | Nr Telefon |
|--------------------------------|------------|
| SC Euro Parts Distribution SRL | 0246286745 |
| SC E-Autoparts Europa SRL      | 0246992112 |

Codul:

```
using Microsoft.Reporting.WinForms;
using System;
using System.Windows.Forms;

namespace CuzicGabriela
{
    public partial class FurnizoriReportForm : Form
    {
        public FurnizoriReportForm()
        {
            InitializeComponent();

            private void FurnizoriReportForm_Load(object sender, EventArgs e)
            {
                // TODO: This line of code loads data into the 'ReportDS.Furnizori'
                table. You can move, or remove it, as needed.
                //this.FurnizoriTableAdapter.Fill(this.ReportDS.Furnizori);

                //this.reportViewer1.RefreshReport();
            }
        }
    }
}
```

```

private void btnAfiseaza_Click(object sender, EventArgs e)
{
    this.FurnizoriTableAdapter.Fill(this.ReportDS.Furnizori,
txtDFurnizor.Text.ToString());
    //Pregatesc parametrii raportului
    ReportParameter[] parameters = new ReportParameter[1];
    if (txtDFurnizor.Text != "")
        parameters[0] = new ReportParameter("Titlu", "Lista Furnizorilor
ce contin \" + txtDFurnizor.Text.ToString() + "\" in nume");
    else parameters[0] = new ReportParameter("Titlu", "Lista Furnizorilor
" + txtDFurnizor.Text.ToString());

    //Comunicare parametrii
    this.reportViewer1.LocalReport.SetParameters(parameters);

    //Emit raport
    this.reportViewer1.RefreshReport();
}
}
}

```

## LucrariForm.cs

### Vizualizare

The screenshot shows a Windows application window titled "LucrariForm". The interface includes three buttons at the top: "Adaugare", "Actualizare", and "Stergere". Below these is a section labeled "Lucrari" with a "Raport Lucrari" button. A table lists various work items, with "Reparatie aprindere, alimentare" selected. To the right of the table is a form with labels "Denumire" and "ID", and corresponding text and numeric input fields.

|   | Denumire Lucrare                 |
|---|----------------------------------|
| ▶ | Reparatie aprindere, alimentare  |
|   | Reparatie evacuare               |
|   | Reparatie optica si vizibilitate |
|   | Reparatie sistem franare         |
|   | Reparatie sistem racire motor    |
|   | Revizie                          |

Denumire:

ID:



## Actualizare

LucrariForm

Adaugare

Actualizare

Stergere

Salvare

Renuntare

Lucrari

Raport Lucrari

|   | Denumire Lucrare                 |
|---|----------------------------------|
| ► | Reparatie aprindere, alimentare  |
|   | Reparatie evacuare               |
|   | Reparatie optica si vizibilitate |
|   | Reparatie sistem franare         |
|   | Reparatie sistem racire motor    |
|   | Revizie                          |

ACTUALIZARE

Denumire

Reparatie aprindere, alimentare

ID

3

## Adăugare

LucrariForm

Adaugare

Actualizare

Stergere

Salvare

Renuntare

Lucrari

Raport Lucrari

|   | Denumire Lucrare                 |
|---|----------------------------------|
| ► | Reparatie aprindere, alimentare  |
|   | Reparatie evacuare               |
|   | Reparatie optica si vizibilitate |
|   | Reparatie sistem franare         |
|   | Reparatie sistem racire motor    |
|   | Revizie                          |

ADAUGARE

Denumire

ID

Codul:

```
using System;
using System.Windows.Forms;
using System.Data.OleDb;

namespace CuzicGabriela
{
    public partial class LucrariForm : Form
    {
        public LucrariForm()
        {
            InitializeComponent();

            private void LucrariForm_Load(object sender, EventArgs e)
            {
                A1();
            }

            private void A1()
            {
                //Umple cu date obiectele DataTable: Lucrari (din DataSet):
                lucrariTableAdapter.Fill(dataSet1.Lucrari);
                //Protectie grid
                dataGridView1.AllowUserToAddRows = false;
                dataGridView1.AllowUserToDeleteRows = false;
                dataGridView1.ReadOnly = true;
                //Protectie txtIdLucrari
                txtIdLucrare.ReadOnly = true;
                A3();
            }

            private void A2()
            {
                //Configurare butoane
                configureazaButoane(false);
                //Dezlegare controale Panel
                legareControale(false);
                //Ridicare protectie controale Panel
                protectiePanel(false);
                //Modificare lblOp
                lblOp.Text = "ADAUGARE";
                //Pozitionare cursor pe primul camp
                txtDLucrare.Focus();
                // Golire campuri
                golireCampuri();
            }

            private void A3()
            {
                //Initializare lblOp
                lblOp.Text = "";
                //Configurare(butoane)
                configureazaButoane(true);
                //Protectie componente Panel
                protectiePanel(true);
                //Legare controale
                legareControale(true);
            }
        }
    }
}
```

```

}
private void A4()
{
    if (lblOp.Text == "ADAUGARE")
    {
        if (!validareCampuriObligatorii()) return;
        adauga_inregistrare();
        golireCampuri();
        //Pune cursor pe primul camp
        txtDLucrare.Focus();
        refresh_grid(lucrariBindingSource.Position);
    }
    else if (lblOp.Text == "ACTUALIZARE")
    {
        if (!validareCampuriObligatorii()) return;
        modifica_inregistrare();
        refresh_grid(lucrariBindingSource.Position);
        A3();
    }
    else
        MessageBox.Show("Operatie actualizare neefectuata");
}

private void A5(TextBox txtB)
{
    OleDbConnection con = new OleDbConnection();
    OleDbCommand cmd = new OleDbCommand();
    OleDbDataReader r;
    if (lblOp.Text == "") return;
    if (txtB.Text == "") return;
    if (btnRenuntare.Focused) return;

    if (txtB == txtDLucrare)
    {
        con.ConnectionString =
lucrariTableAdapter.Connection.ConnectionString;
        cmd.Connection = con;
        if (lblOp.Text == "ADAUGARE")
        {
            cmd.CommandText = "Select IdLucrare From Lucrari where
DLucrare='" + txtDLucrare.Text + "'";
            con.Open();
            r = cmd.ExecuteReader();
            if (r.Read())
            {
                MessageBox.Show("Lucrare deja existentă!");
                txtDLucrare.Focus();
            }
            con.Close();
        }
        else if (lblOp.Text == "ACTUALIZARE")
        {
            cmd.CommandText = "Select IdLucrare From Lucrari where
DLucrare='" + txtDLucrare.Text +
            "' and IdLucrare<> " + txtIdLucrare.Text;
            con.Open();
            r = cmd.ExecuteReader();
            if (r.Read())

```

```

        {
            MessageBox.Show("Lucrare deja existentă!");
            txtDLucrare.Focus();
        }
        con.Close();
    }
}

private void A7()
{
    //Configurare butoane
    configureazaButoane(false);
    //Dezlegare controale Panel
    legareControale(false);
    //Ridicarea protectiei controale Panel
    protectiePanel(false);
    //Modificare lblOp
    lblOp.Text = "ACTUALIZARE";
    //Pozitionare cursor pe primul camp
    txtDLucrare.Focus();
}

private void A8()
{
    OleDbConnection con = new OleDbConnection();
    OleDbCommand cmd = new OleDbCommand();
    OleDbDataReader r;
    con.ConnectionString =
lucrariTableAdapter.Connection.ConnectionString;
    cmd.Connection = con;
    cmd.CommandText = "Select IdBonConsum From BonuriConsum where
IdLucrare=" + txtIdLucrare.Text;
    con.Open();
    r = cmd.ExecuteReader();
    if (r.Read())
    {
        MessageBox.Show("Lucrare referită in tabela BonuriConusum! Nu se
poate sterge!");
        con.Close();
        return;
    }
    con.Close();
    con.ConnectionString =
lucrariTableAdapter.Connection.ConnectionString;
    cmd.Connection = con;
    cmd.CommandText = "Delete From Lucrari Where IdLucrare = " +
txtIdLucrare.Text;
    //MessageBox.Show("Inregistrare stearsa cu succes!");
    con.Open();
    cmd.ExecuteNonQuery();
    con.Close();
    refresh_grid(lucrariBindingSource.Position);
}

private void configureazaButoane(bool v)
{
    btnRenuntare.Visible = !v;
    btnSalvare.Visible = !v;
}

```

```

        btnAdaugare.Enabled = v;
        btnActualizare.Enabled = v;
        btnStergere.Enabled = v;
    }
    private void legareControale(bool v)
    {
        if (v)
        {
            txtDLucrare.DataBindings.Add("Text", lucrariBindingSource,
"DLucrare");
            txtIdLucrare.DataBindings.Add("Text", lucrariBindingSource,
"IdLucrare");
        }
        else
        {
            txtDLucrare.DataBindings.Clear();
            txtIdLucrare.DataBindings.Clear();
        }
    }
    private void protectiePanel(bool v)
    {
        txtDLucrare.ReadOnly = v;
    }
    private void golireCampuri()
    {
        txtDLucrare.Text = "";
        txtIdLucrare.Text = "";
    }
    private bool validareCampuriObligatorii()
    {
        //Validare de completare obligatorie campurile: Nume, NrTelefon
        if (txtDLucrare.Text == "")
        {
            MessageBox.Show("Completati denumirea lucrării!");
            txtDLucrare.Focus();
            return false;
        }
        return true;
    }

    private void adauga_inregistrare()
    {
        string listaCampuri;
        string listaValori;
        OleDbConnection con = new OleDbConnection();
        OleDbCommand cmd = new OleDbCommand();
        con.ConnectionString =
lucrariTableAdapter.Connection.ConnectionString;
        cmd.Connection = con;
        listaCampuri = "DLucrare";
        listaValori = "" + txtDLucrare.Text + "";
        cmd.CommandText = "Insert into Lucrari(" + listaCampuri + ") " +
"Select " + listaValori;
        //MessageBox.Show("Inregistrare adaugata cu succes!");
        con.Open();
        cmd.ExecuteNonQuery();
        con.Close();
    }
}

```

```

private void refresh_grid(int p)
{
    lucrariTableAdapter.Fill(dataSet1.Lucrari);
    lucrariBindingSource.Position = p;
}

private void modifica_inregistrare()
{
    string listaSet;
    OleDbConnection con = new OleDbConnection();
    OleDbCommand cmd = new OleDbCommand();
    con.ConnectionString =
lucrariTableAdapter.Connection.ConnectionString;
    cmd.Connection = con;
    listaSet = "DLucrare = '" + txtDLucrare.Text + "'";
    cmd.CommandText = "Update Lucrari Set " + listaSet + " Where
IdLucrare=" +
    txtIdLucrare.Text;
    //MessageBox.Show("Actualizare efectuata cu succes!");
    con.Open();
    cmd.ExecuteNonQuery();
    con.Close();
}

private void btnAdaugare_Click(object sender, EventArgs e)
{
    A2();
}

private void btnActualizare_Click(object sender, EventArgs e)
{
    A7();
}

private void btnStergere_Click(object sender, EventArgs e)
{
    const string mesaj = "Confirmati stergerea";
    const string titlu = "Stergere inregistrare";
    var rezultat = MessageBox.Show(mesaj, titlu, MessageBoxButtons.YesNo,
    MessageBoxIcon.Warning);
    if (rezultat == DialogResult.No) return;
    A8();
}

private void btnSalvare_Click(object sender, EventArgs e)
{
    A4();
}

private void btnRenuntare_Click(object sender, EventArgs e)
{
    A3();
}

private void txtDLucrare_Leave(object sender, EventArgs e)
{
    A5(txtDLucrare);
}

```

```

        private void btnRaportLucrari_Click(object sender, EventArgs e)
        {
            LucrariReportForm f = new LucrariReportForm();
            f.ShowDialog();
        }
    }
}

```

## LucrariReportForm.cs

Denumire Lucrare: sistem

Afiseaza raport

Lista Lucrarilor ce contin "sistem" in denumire

| Denumire Lucrare              |
|-------------------------------|
| Reparatie sistem franare      |
| Reparatie sistem racire motor |

Codul:

```

using Microsoft.Reporting.WinForms;
using System;
using System.Windows.Forms;

namespace CuzicGabriela
{
    public partial class LucrariReportForm : Form
    {
        public LucrariReportForm()
        {
            InitializeComponent();
        }
    }
}

```

```

    }

    private void LucrariReportForm_Load(object sender, EventArgs e)
    {
        // TODO: This line of code loads data into the 'ReportDS.Lucrari'
        table. You can move, or remove it, as needed.
        //this.LucrariTableAdapter.Fill(this.ReportDS.Lucrari);
    }

    private void btnAfiseaza_Click(object sender, EventArgs e)
    {
        this.LucrariTableAdapter.Fill(this.ReportDS.Lucrari,
        txtDLucrare.Text.ToString());
        //Pregatesc parametrii raportului
        ReportParameter[] parameters = new ReportParameter[1];
        if (txtDLucrare.Text != "")
            parameters[0] = new ReportParameter("Titlu", "Lista Lucrarilor ce
        contin \"\" + txtDLucrare.Text.ToString() + "\" in denumire");
        else parameters[0] = new ReportParameter("Titlu", "Lista Lucrarilor "
        + txtDLucrare.Text.ToString());

        //Comunicare parametrii
        this.reportViewer1.LocalReport.SetParameters(parameters);

        //Emit raport
        this.reportViewer1.RefreshReport();
    }
}
}

```

### 3.2. Evenimente

#### FacturiVizForm.cs

The screenshot shows a Windows application window titled "Facturi". The interface is divided into two main sections: "Facturi" (Invoices) and "Detalii factura" (Invoice Details).

**Facturi Section:**

|   | Denumire Furnizor               | Nr Factura | Data Factura | Total |
|---|---------------------------------|------------|--------------|-------|
| ▶ | AutoKama SRL                    | 100        | 2/1/2021     | 1900  |
|   | SC Euro Parts Distribution S... | 200        | 2/3/2021     | 1000  |
|   | TecDoc SRL                      | 300        | 3/7/2021     | 1140  |
|   | PieseAuto Marketplace SRL       | 301        | 4/18/2021    | 995   |

**Detalii factura Section:**

|   | Nr | Denumire Piesa        | Canitate | Pret Achizitie | UM  |
|---|----|-----------------------|----------|----------------|-----|
| ▶ | 1  | Motor stergator spate | 10       | 100            | Buc |
|   | 2  | Far                   | 5        | 100            | Buc |
|   | 3  | Ulei motor            | 20       | 20             | l   |

**Right Panel:**

ID Factura:

Buttons:

- Factura noua
- Modificare factura
- Sterge factura
- Raport Aproximare
- Reprezentare grafica



## Codul:

```
using System;
using System.Windows.Forms;
using System.Data.OleDb;

namespace CuzicGabriela
{
    public partial class FacturiVizForm : Form
    {
        public FacturiVizForm()
        {
            InitializeComponent();

            private void refreshGrid()
            {
                facturiTableAdapter.Fill(dataSet1.Facturi);
                facturiContinutTableAdapter.Fill(dataSet1.FacturiContinut);
            }

            public void position_grid(int p)
            {
                facturiBindingSource.Position = p;
            }

            private void FacturiVizForm_Load(object sender, EventArgs e)
            {
                refreshGrid();
                try
                {
                    facturiContinutBindingSource.Filter = "IdFactura=" + txtIdFactura.Text;
                }
                catch { }
            }

            private void facturiBindingSource_PositionChanged(object sender, EventArgs e)
            {
                try
                {
                    facturiContinutBindingSource.Filter = "IdFactura=" + txtIdFactura.Text;
                }
                catch { }
            }

            private void btnFacturaNoua_Click(object sender, EventArgs e)
            {
                FacturiActForm f = new FacturiActForm();
                f.ShowDialog();
                refreshGrid();
            }

            private void btnStergeFactura_Click(object sender, EventArgs e)
            {
                const string mesaj = "Confirmati stergerea";
                const string titlu = "Stergere inregistrare";
                var rezultat = MessageBox.Show(mesaj, titlu, MessageBoxButtons.YesNo,
                    MessageBoxIcon.Warning);
            }
        }
    }
}
```

```

        if (rezultat == DialogResult.No) return;
        OleDbConnection con = new OleDbConnection();
        OleDbCommand cmd = new OleDbCommand();
        int p = facturiBindingSource.Position;
        con.ConnectionString = facturiTableAdapter.Connection.ConnectionString;
        cmd.Connection = con;
        // Sterg continut factura
        cmd.CommandText = "Delete From FacturiContinut Where IdFactura = " +
txtIdFactura.Text;
        MessageBox.Show(cmd.CommandText);
        con.Open();
        cmd.ExecuteNonQuery();
        // Sterg factura
        cmd.CommandText = "Delete From Facturi Where IdFactura = " +
txtIdFactura.Text;
        //MessageBox.Show("Stergere efectuata cu succes!");
        cmd.ExecuteNonQuery();
        con.Close();
        // Refresh grid-uri
        refreshGrid();
        position_grid(p + 1);
        try
        {
            facturiContinutBindingSource.Filter = "IdFactura=" + txtIdFactura.Text;
        }
        catch { }
    }

    private void btnModificaFactura_Click(object sender, EventArgs e)
    {
        int p = facturiBindingSource.Position;
        FacturiActForm f = new FacturiActForm();
        f.completeazaTitlu("MODIFICARE FACTURA");
        f.bs1 = facturiBindingSource;
        f.bs2 = facturiContinutBindingSource;
        f.ShowDialog();
        refreshGrid();
        position_grid(p);
        try
        {
            facturiContinutBindingSource.Filter = "IdFactura=" + txtIdFactura.Text;
        }
        catch { }
    }

    private void btnRaportAprovizionare_Click(object sender, EventArgs e)
    {
        AprovizionareReportForm f = new AprovizionareReportForm();
        f.ShowDialog();
    }

    private void btnGrafic_Click(object sender, EventArgs e)
    {
        StatisticiAprovizionareForm f = new StatisticiAprovizionareForm();
        f.ShowDialog();
    }
}

```

## FacturiActForm.cs

### Actualizare

FacturiActForm

Nr Factura

200

Data

2/ 3/2021

Furnizor

SC Euro Parts Distributio

MODIFICARE FACTURA

Total

1000

Confirmare

|   | Nr | Denumire Piesa     | Cantitate | Pret Achizitie | UM  | Valoare |
|---|----|--------------------|-----------|----------------|-----|---------|
| ▶ | 1  | Placute frana      | 30        | 25             | Set | 750     |
|   | 2  | Oglinda exterioara | 5         | 50             | Buc | 250     |
| * |    |                    |           |                |     |         |

### Adăugare

FacturiActForm

Nr Factura

Data

5/ 9/2021

Furnizor

FACTURA NOUA

Total

Confirmare

|    | Nr | Denumire Piesa | Cantitate | Pret Achizitie | UM | Valoare |
|----|----|----------------|-----------|----------------|----|---------|
| ▶▶ | 1  |                |           |                |    |         |

## Codul:

```
using System;
using System.Windows.Forms;
using System.Data.OleDb;

namespace CuzicGabriela
{
    public partial class FacturiActForm : Form
    {
        private OleDbConnection con = new OleDbConnection();
        private OleDbCommand cmd = new OleDbCommand();
        private OleDbDataReader rdr;
        private int idF;
        public BindingSource bs1;
        public BindingSource bs2;
        public FacturiActForm()
        {
            InitializeComponent();
        }

        private void FacturiActForm_Load(object sender, EventArgs e)
        {
            A1();
        }

        private void A1()
        {
            //Incarcare DataTable PieseAuto
            pieseAutoTableAdapter.Fill(dataSet1.PieseAuto);
            //Incarcare DataTable Furnizori
            furnizoriTableAdapter.Fill(dataSet1.Furnizori);
            // Protectie la modificare
            txtNrFactura.ReadOnly = true;
            txtTotal.ReadOnly = true;
            nrCrtDataGridViewTextBoxColumn.ReadOnly = true;
            uMDataGridViewTextBoxColumn.ReadOnly = true;
            //pretAchizitieDataGridViewTextBoxColumn.ReadOnly = true;
            //idPiesaAutoDataGridViewTextBoxColumn.ReadOnly = true;

            // Initializare cmbFurnizor
            if (lblOp.Text == "MODIFICARE FACTURA") completeazaFactura();
            else if (lblOp.Text == "FACTURA NOUA") cmbFurnizor.SelectedIndex = -1;
        }

        private void A2()
        {
            if (!validareCampuriObligatorii()) return;
            if (lblOp.Text == "MODIFICARE FACTURA")
            {
                modificaInregistrare();
                stergeContinut();
                adaugaInregistrariFacturiContinut();
                this.Close();
            }
            else if (lblOp.Text == "FACTURA NOUA")
            {
                generez_nrF();
                adaugaInregistrareFacturi();
            }
        }
    }
}
```

```

        cautaInregistrare();
        adaugaInregistrariFacturiContinut();
        golireCampuri();
    }
}

private void A3()
{
    // Generare NrCrt
    DataRowView current =
(DataRowView)facturaContinutManevraBindingSource.Current;
    try { current["NrCrt"] = facturaContinutManevraBindingSource.Position + 1; }
    catch { }
}

private void A4()
{
    try
    {
        if (dataGridView1.CurrentCell.ColumnIndex == 1)
        {
            DataRowView current = (DataRowView)pieseAutoBindingSource.Current;
            dataGridView1.CurrentRow.Cells[4].Value = current["UM"];
            //dataGridView1.CurrentRow.Cells[3].Value = current["Pret"];
            //dataGridView1.CurrentRow.Cells[6].Value = current["IdPiesaAuto"];
            calcTotal();
        }
        if (dataGridView1.CurrentCell.ColumnIndex == 3)
        {
            facturaContinutManevraBindingSource.EndEdit();
            calcTotal();
        }
        if (dataGridView1.CurrentCell.ColumnIndex == 2)
        {
            facturaContinutManevraBindingSource.EndEdit();
            calcTotal();
        }
    }
    catch { }
}

private void A5(DataGridViewRowCancelEventArgs e)
{
    const string mesaj = "Confirmati stergerea";
    const string titlu = "Stergere inregistrare";
    var rezultat = MessageBox.Show(mesaj, titlu,
    MessageBoxButtons.YesNo,
    MessageBoxIcon.Warning);
    if (rezultat == DialogResult.No) e.Cancel = true;
}

public void completeazaTitlu(String titlu)
{
    lblOp.Text = titlu;
}

private void completeazaFactura()
{

```

```

        DataRowView current = (DataRowView)bs1.Current;
        idF = (int)current["IdFactura"];
        txtNrFactura.Text = Convert.ToString(current["NrFactura"]);
        cmbFurnizor.Text = current["DFurnizor"].ToString();
        txtTotal.Text = current["Total"].ToString();
        dateTimePicker1.Value = Convert.ToDateTime(current["DataFactura"]);
        bs2.MoveFirst();
        dataSet1.FacturaContinutManevra.Clear();
        for (int i = 1; i <= bs2.Count; i++)
        {
            current = (DataRowView)bs2.Current;
            DataSet1.FacturaContinutManevraRow r =
dataSet1.FacturaContinutManevra.NewFacturaContinutManevraRow();
            r.NrCrt = Convert.ToInt16(current["NrCrt"]);
            r.DPiesa = Convert.ToString(current["DPiesa"]);
            r.UM = Convert.ToString(current["UM"]);
            r.PretAchizitie = Convert.ToDecimal(current["PretAchizitie"]);
            r.Cantitate = Convert.ToDecimal(current["Cantitate"]);
            r.IdPiesaAuto = Convert.ToInt32(current["IdPiesaAuto"]);
            r.Valoare = Convert.ToDecimal(current["Valoare"]);
            dataSet1.FacturaContinutManevra.Rows.Add(r);
            bs2.MoveNext();
        }
    }

    private void calcTotal()
    { // Calcul total valori
        decimal t = 0;
        foreach (DataRow r in dataSet1.FacturaContinutManevra)
        {
            if (r["Valoare"] != DBNull.Value)
                t += (decimal)r["Valoare"];
        }
        txtTotal.Text = Convert.ToString(t);
    }

    private void generez_nrF()
    {
        con.ConnectionString = pieseAutoTableAdapter.Connection.ConnectionString;
        cmd.Connection = con;
        cmd.CommandText = "SELECT Max(Facturi.NrFactura) AS NrMax FROM Facturi";
        con.Open();
        rdr = cmd.ExecuteReader();
        rdr.Read();
        txtNrFactura.Text = Convert.ToString(rdr.GetInt32(0) + 1);
        rdr.Close();
        con.Close();
    }

    private bool validareCampuriObligatorii()
    {
        //Validare de completare obligatorie campul Furnizor
        if (cmbFurnizor.Text == "")
        {
            MessageBox.Show("Completati Furnizor !");
            cmbFurnizor.Focus();
            return false;
        }
    }

```

```

// Validare completare continut
if (facturaContinutManevraBindingSource.Count == 0)
{
    MessageBox.Show("Completati continut factura !");
    dataGridView1.Focus();
    return false;
}
if (dataGridView1.CurrentRow.Cells[2].Value.ToString() == "")
{
    MessageBox.Show("Completati pretul de achizitie !");
    dataGridView1.Focus();
    return false;
}
if (dataGridView1.CurrentRow.Cells[3].Value.ToString() == "")
{
    MessageBox.Show("Completati cantitatea !");
    dataGridView1.Focus();
    return false;
}
return true;
}

private void adaugaInregistrareFacturi()
{
    string listaCampuri;
    string listaValori;
    DateTime d = dateTimePicker1.Value;
    listaCampuri = "NrFactura, DataFactura, IdFurnizor";
    listaValori = txtNrFactura.Text +
    ",#" + Convert.ToString(d.Month) + "/" +
    + Convert.ToString(d.Day) + "/" +
    + Convert.ToString(d.Year) + "#," +
    + cmbFurnizor.SelectedValue;
    cmd.CommandText = "Insert into Facturi(" + listaCampuri + ") " +
    "Select " + listaValori;
    con.Open();
    cmd.ExecuteNonQuery();
    con.Close();
}

private void cautaInregistrare()
{
    cmd.CommandText = "SELECT idFactura From Facturi Where NrFactura = " +
    txtNrFactura.Text;
    con.Open();
    rdr = cmd.ExecuteReader();
    rdr.Read();
    idF = rdr.GetInt32(0);
    rdr.Close();
    con.Close();
}

private void adaugaInregistrariFacturiContinut()
{
    string listaCampuri = "IdFactura, NrCrt, IdPiesaAuto, Cantitate,
PretAchizitie";
    string listaValori;
    con.Open();

```

```

        foreach (DataRow r in dataSet1.FacturaContinutManevra)
        {
            //con.ConnectionString =
pieseAutoTableAdapter.Connection.ConnectionString;
            cmd.Connection = con;
            cmd.CommandText = "SELECT IdPiesaAuto FROM PieseAuto WHERE DPiesa=\"\" +
r["DPiesa"] + "\"";
            //con.Open();
            rdr = cmd.ExecuteReader();
            rdr.Read();
            String IdPiesaAuto = Convert.ToString(rdr.GetInt32(0));
            rdr.Close();
            //con.Close();

            listaValori = idF + "," + r["NrCrt"] + "," + IdPiesaAuto + "," +
r["Cantitate"] + "," + r["PretAchizitie"];
            cmd.CommandText = "Insert into FacturiContinut(" + listaCampuri + ") " +
"Select " + listaValori;
            cmd.ExecuteNonQuery();
        }
        con.Close();
    }
    private void golireCampuri()
    {
        txtNrFactura.Text = "";
        cmbFurnizor.SelectedIndex = -1;
        txtTotal.Text = "";
        dataSet1.FacturaContinutManevra.Clear();
    }

    private void stergeContinut()
    {
        cmd.CommandText = "Delete from FacturiContinut Where IdFactura = " + idF;
        con.Open();
        cmd.ExecuteNonQuery();
        con.Close();
    }

    private void modificaInregistrare()
    {
        DateTime d = dateTimePicker1.Value;
        con.ConnectionString = pieseAutoTableAdapter.Connection.ConnectionString;
        cmd.Connection = con;
        string clauzaWhere = " Where IdFactura = " + idF;
        string clauzaSet = "Set DataFactura = #" + d.Month + "/" + d.Day + "/" +
d.Year + "#," + "IdFurnizor = " + cmbFurnizor.SelectedValue;
        cmd.CommandText = "Update Facturi " + clauzaSet + clauzaWhere;
        con.Open();
        cmd.ExecuteNonQuery();
        con.Close();
    }

    private void facturaContinutManevraBindingSource_CurrentChanged(object sender,
EventArgs e)
    {
        A3();
    }

```



```

        private void dataGridView1_CellValueChanged(object sender,
DataGridViewCellEventArgs e)
        {
            A4();
        }

        private void dataGridView1_UserDeletingRow(object sender,
DataGridViewRowCancelEventArgs e)
        {
            A5(e);
        }

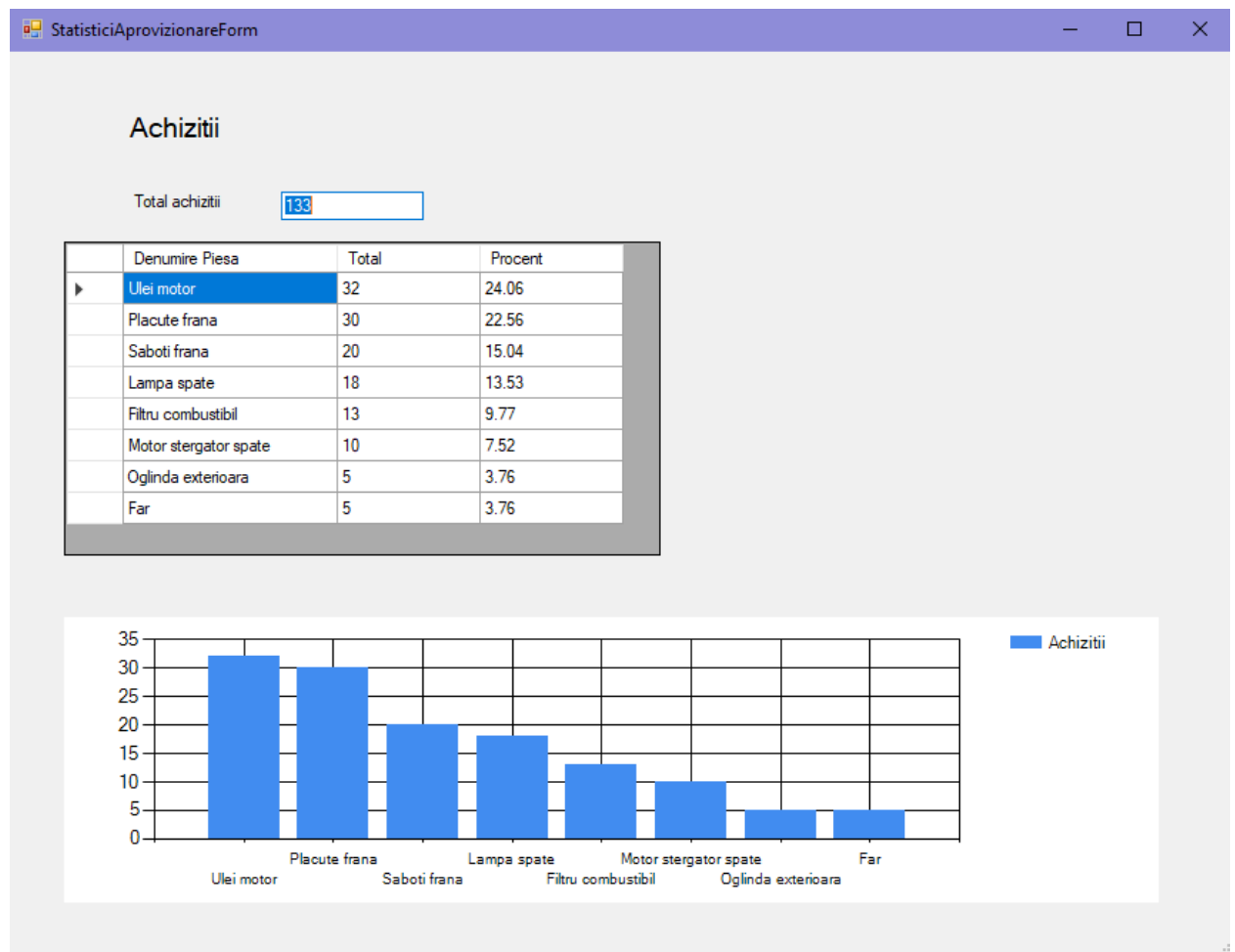
        private void dataGridView1_DataError(object sender,
DataGridViewDataErrorEventArgs e)
        {
            MessageBox.Show("Format eronat");
        }

        private void dataGridView1_UserDeletedRow(object sender, DataGridViewRowEventArgs
e)
        {
            calcTotal();
        }

        private void btnConfirmare_Click(object sender, EventArgs e)
        {
            A2();
        }
    }
}

```

## StatisticiAprovizionareForm.cs



### Codul:

```
using System;
using System.Data;
using System.Data.OleDb;
using System.Windows.Forms;

namespace CuzicGabriela
{
    public partial class StatisticiAprovizionareForm : Form
    {
        private OleDbConnection con = new OleDbConnection();
        private OleDbCommand cmd = new OleDbCommand();
        private OleDbDataReader rdr;
        private decimal totalAchizitii;
        public StatisticiAprovizionareForm()
        {
            InitializeComponent();
        }

        private void StatisticiAprovizionareForm_Load(object sender, EventArgs e)
        {

```

```

        // TODO: This line of code loads data into the 'statDS.AchizitiiPieseAuto'
        table. You can move, or remove it, as needed.
        this.achizitiiPieseAutoTableAdapter.Fill(this.statDS.AchizitiiPieseAuto);
        //Pregatire con si cmd
        con = achizitiiPieseAutoTableAdapter.Connection;
        cmd.Connection = con;
        //Total achizitii
        cmd.CommandText = "select sum(Cantitate) as Total from FacturiContinut";

        con.Open();
        rdr = cmd.ExecuteReader();
        rdr.Read();
        totalAchizitii = rdr.GetDecimal(0);
        con.Close();
        rdr.Close();

        txtTotalAchizitii.Text = "" + totalAchizitii;

        foreach (DataRow r in statDS.AchizitiiPieseAuto)
        {
            decimal x;
            x = (decimal)r["Total"] / totalAchizitii;
            x = Math.Round(x, 4) * 100;
            x = Math.Round(x, 2);
            r["Procent"] = x;
        }
    }
}

```

## AprovizionareReportForm.cs

The screenshot shows a Windows application window titled "AprovizionareReportForm". The interface includes two date pickers: "Data inceput" (3/ 2/2021) and "Data sfarsit" (5/ 9/2021), followed by an "Afiseaza raport" button. Below these is a report viewer showing a table of purchases. The table has columns: Data Factura, DFurnizor, DPiesa, Cantitate, Pret, and UM. The data is filtered for the period 3/2/2021 - 5/9/2021.

| Data Factura | DFurnizor                 | DPiesa             | Cantitate | Pret | UM  |
|--------------|---------------------------|--------------------|-----------|------|-----|
| 07-Mar-2021  | TecDoc SRL                | Ulei motor         | 12        | 50   | l   |
| 07-Mar-2021  | TecDoc SRL                | Lampa spate        | 18        | 200  | Buc |
| 18-Apr-2021  | PieseAuto Marketplace SRL | Saboti frana       | 20        | 65   | Buc |
| 18-Apr-2021  | PieseAuto Marketplace SRL | Filtru combustibil | 13        | 25   | Buc |

## Codul:

```
using System;
using System.Windows.Forms;
using Microsoft.Reporting.WinForms;

namespace CuzicGabriela
{
    public partial class AprovizionareReportForm : Form
    {
        public AprovizionareReportForm()
        {
            InitializeComponent();

            private void AprovizionareReportForm_Load(object sender, EventArgs e)
            {
                // TODO: This line of code loads data into the 'ReportDS.Aprovizionare'
                table. You can move, or remove it, as needed.
                //this.AprovizionareTableAdapter.Fill(this.ReportDS.Aprovizionare);

                //this.reportViewer1.RefreshReport();
            }

            private void btnAfiseaza_Click(object sender, EventArgs e)
            {
                DateTime d1 = new DateTime(dateTimePicker1.Value.Year,
                dateTimePicker1.Value.Month, dateTimePicker1.Value.Day);
                DateTime d2 = new DateTime(dateTimePicker2.Value.Year,
                dateTimePicker2.Value.Month, dateTimePicker2.Value.Day);
                this.AprovizionareTableAdapter.Fill(this.ReportDS.Aprovizionare, d1, d2);
                //Pregatesc parametrii raportului
                ReportParameter[] parameters = new ReportParameter[1];
                parameters[0] = new ReportParameter("Titlu", "Lista achizitiilor din perioada
                " + d1.ToShortDateString() +
                " - " +
                d2.ToShortDateString());

                //Comunicare parametrii
                this.reportViewer1.LocalReport.SetParameters(parameters);

                //Emit raport
                this.reportViewer1.RefreshReport();
            }
        }
    }
}
```

## BonConsumVizForm.cs

**Bonuri consum**

|   | Nr Bon | Data Bon  | Nume Client      | Denumire Lucrare                 | Total |
|---|--------|-----------|------------------|----------------------------------|-------|
|   | 1001   | 2/6/2021  | Costea Marian    | Reparatie sistem franare         | 580   |
| ▶ | 1002   | 2/7/2021  | Popescu Sorin    | Reparatie sistem racire motor    | 435   |
|   | 1003   | 2/7/2021  | Craiu Lucian     | Reparatie evacuare               | 1390  |
|   | 1004   | 2/12/2021 | Mirea Daniel     | Reparatie aprindere, alimentare  | 363   |
|   | 1005   | 2/13/2021 | Arsenie Ruxandra | Reparatie optica si vizibilitate | 1140  |
|   | 1006   | 2/14/2021 | Costea Marian    | Reparatie sistem franare         | 470   |

ID Bon

Bon nou  
Modificare bon  
Sterge Bonul  
Raport Consumuri  
Reprezentare grafica

**Detalii bonuri consum**

|   | Nr | Denumire Piesa            | Cantitate | Pret Vanzare | UM  | Valoare |
|---|----|---------------------------|-----------|--------------|-----|---------|
| ▶ | 1  | Toba esapament            | 3         | 230          | Buc | 690     |
|   | 2  | Teava flexibila esapament | 2         | 350          | Buc | 700     |

### Codul:

```
using System;
using System.Windows.Forms;
using System.Data.OleDb;

namespace CuzicGabriela
{
    public partial class BonConsumVizForm : Form
    {
        public BonConsumVizForm()
        {
            InitializeComponent();
        }

        private void refreshGrid()
        {
            bonuriConsumTableAdapter.Fill(dataSet1.BonuriConsum);
            bonuriConsumContinutTableAdapter.Fill(dataSet1.BonuriConsumContinut);
        }

        public void position_grid(int p)
        {
            bonuriConsumBindingSource.Position = p;
        }

        private void BonConsumVizForm_Load(object sender, EventArgs e)
        {
            refreshGrid();
            try
            {

```

```

        bonuriConsumContinutBindingSource.Filter = "IdBonConsum=" +
txtIdBonConsum.Text;
    }
    catch { }
}

e)
private void bonuriConsumBindingSource_PositionChanged(object sender, EventArgs
{
    try
    {
        bonuriConsumContinutBindingSource.Filter = "IdBonConsum=" +
txtIdBonConsum.Text;
    }
    catch { }
}

private void btnBonNou_Click(object sender, EventArgs e)
{
    BonConsumActForm f = new BonConsumActForm();
    f.ShowDialog();
    refreshGrid();
}

private void btnStergeBon_Click(object sender, EventArgs e)
{
    const string mesaj = "Confirmati stergerea";
    const string titlu = "Stergere inregistrare";
    var rezultat = MessageBox.Show(mesaj, titlu, MessageBoxButtons.YesNo,
    MessageBoxIcon.Warning);
    if (rezultat == DialogResult.No) return;
    OleDbConnection con = new OleDbConnection();
    OleDbCommand cmd = new OleDbCommand();
    int p = bonuriConsumBindingSource.Position;
    con.ConnectionString = bonuriConsumTableAdapter.Connection.ConnectionString;
    cmd.Connection = con;
    // Sterg continut BonConsum
    cmd.CommandText = "Delete From BonuriConsumContinut Where IdBonConsum = " +
txtIdBonConsum.Text;
    //MessageBox.Show("Stergere realizata cu succes!");
    con.Open();
    cmd.ExecuteNonQuery();
    // Sterg BonConsum
    cmd.CommandText = "Delete From BonuriConsum Where IdBonConsum = " +
txtIdBonConsum.Text;
    cmd.ExecuteNonQuery();
    con.Close();
    // Refresh grid-uri
    refreshGrid();
    position_grid(p + 1);
    try
    {
        bonuriConsumContinutBindingSource.Filter = "IdBonConsum=" +
txtIdBonConsum.Text;
    }
    catch { }
}

```

```

private void btnModificaBon_Click(object sender, EventArgs e)
{
    int p = bonuriConsumBindingSource.Position;
    BonConsumActForm f = new BonConsumActForm();
    f.completeazaTitlu("Modificare Bon consum");
    f.bs1 = bonuriConsumBindingSource;
    f.bs2 = bonuriConsumContinutBindingSource;
    f.ShowDialog();
    refreshGrid();
    position_grid(p);
    try
    {
        bonuriConsumContinutBindingSource.Filter = "IdBonConsum=" +
txtIdBonConsum.Text;
    }
    catch { }
}

private void btnRaportConsumuri_Click(object sender, EventArgs e)
{
    ConsumReportForm f = new ConsumReportForm();
    f.ShowDialog();
}

private void btnGrafic_Click(object sender, EventArgs e)
{
    StatisticiConsumForm f = new StatisticiConsumForm();
    f.ShowDialog();
}
}
}

```

## BonConsumActForm.cs

### Actualizare

The screenshot shows a Windows application window titled "BonConsumActForm". Inside, there is a form titled "Modificare Bon consum". The form contains several input fields and a table.

Form Fields:

- Nr Bon Consum: 1003
- Data: 7/2021 (with a calendar icon)
- Client: Craiu Lucian (dropdown menu)
- Lucrare: Reparatie evacuare (dropdown menu)
- Total: 580
- Confirmare button

Table:

| Nr | Denumire Pieasa           | Cantitate | Pret Vanzare | UM  | Valoare |
|----|---------------------------|-----------|--------------|-----|---------|
| 1  | Toba esapament            | 1         | 230          | Buc | 230     |
| 2  | Teava flexibila esapament | 1         | 350          | Buc | 350     |
| *  |                           |           |              |     |         |

## Adăugare

Nr Bon Consum

Data

Client

Lucrare

Total

|   | Nr | Denumire Piesa | Cantitate | Pret Vanzare | UM | Valoare |
|---|----|----------------|-----------|--------------|----|---------|
| * |    |                |           |              |    |         |

### Codul:

```
using System;
using System.Windows.Forms;
using System.Data.OleDb;

namespace CuzicGabriela
{
    public partial class BonConsumActForm : Form
    {
        private OleDbConnection con = new OleDbConnection();
        private OleDbCommand cmd = new OleDbCommand();
        private OleDbDataReader rdr;
        private int idB;
        public BindingSource bs1;
        public BindingSource bs2;
        public BonConsumActForm()
        {
            InitializeComponent();
        }

        private void BonConsumActForm_Load(object sender, EventArgs e)
        {
            // TODO: This line of code loads data into the 'dataSet1.PieseAuto' table.
            You can move, or remove it, as needed.
            this.pieseAutoTableAdapter.Fill(this.dataSet1.PieseAuto);
            A1();
        }
    }
}
```



```

private void A1()
{
    //Incarcare DataTable PieseAuto
    pieseAutoTableAdapter.Fill(dataSet1.PieseAuto);
    //Incarcare DataTable Furnizori
    clientiTableAdapter.Fill(dataSet1.Clienti);
    lucrariTableAdapter.Fill(dataSet1.Lucrari);
    // Protectie la modificare
    txtNrBonConsum.ReadOnly = true;
    txtTotal.ReadOnly = true;
    nrCrtDataGridViewTextBoxColumn.ReadOnly = true;
    uMDataGridViewTextBoxColumn.ReadOnly = true;
    pretVanzareDataGridViewTextBoxColumn.ReadOnly = true;
    //idPiesaAutoDataGridViewTextBoxColumn.ReadOnly = true;

    // Initializare cmbClient si cmbLucrare
    if (lblOp.Text == "Modificare Bon consum") completeazaBonConsum();
    else if (lblOp.Text == "Bon consum nou")
    {
        cmbClient.SelectedIndex = -1;
        cmbLucrare.SelectedIndex = -1;
    }
}

private void A2()
{
    if (!validareCampuriObligatorii()) return;
    if (!verificaStoc()) return;
    if (lblOp.Text == "Modificare Bon consum")
    {
        modificaInregistrare();
        stergeContinut();
        adaugaInregistrariBonuriConsumContinut();
        this.Close();
    }
    else if (lblOp.Text == "Bon consum nou")
    {
        generez_nrB();
        adaugaInregistrareBonuriConsum();
        cautaInregistrare();
        adaugaInregistrariBonuriConsumContinut();
        golineCampuri();
    }
}

private void A3()
{
    // Generare NrCrt
    DataRowView current =
(DataRowView)bonConsumContinutManevraBindingSource.Current;
    try { current["NrCrt"] = bonConsumContinutManevraBindingSource.Position + 1;
}
    catch { }
}

private void A4()
{
    try

```

```

    {
        if (dataGridView1.CurrentCell.ColumnIndex == 1)
        {
            DataRowView current = (DataRowView)pieseAutoBindingSource.Current;
            dataGridView1.CurrentRow.Cells[4].Value = current["UM"];
            dataGridView1.CurrentRow.Cells[3].Value = current["Pret"];
            dataGridView1.CurrentRow.Cells[6].Value = current["IdPiesaAuto"];
            calcTotal();
        }
        if (dataGridView1.CurrentCell.ColumnIndex == 3)
        {
            bonConsumContinutManevraBindingSource.EndEdit();
            calcTotal();
        }
        if (dataGridView1.CurrentCell.ColumnIndex == 4)
        {
            bonConsumContinutManevraBindingSource.EndEdit();
            calcTotal();
        }
        bonConsumContinutManevraBindingSource.EndEdit();
        calcTotal();
    }
    catch { }
}

private void A5(DataGridViewRowCancelEventArgs e)
{
    const string mesaj = "Confirmati stergerea";
    const string titlu = "Stergere inregistrare";
    var rezultat = MessageBox.Show(mesaj, titlu,
        MessageBoxButtons.YesNo,
        MessageBoxIcon.Warning);
    if (rezultat == DialogResult.No) e.Cancel = true;
}

public void completeazaTitlu(String titlu)
{
    lblOp.Text = titlu;
}

private void completeazaBonConsum()
{
    DataRowView current = (DataRowView)bs1.Current;
    idB = (int)current["IdBonConsum"];
    txtNrBonConsum.Text = Convert.ToString(current["NrBonConsum"]);
    cmbClient.Text = current["DClient"].ToString();
    cmbLucrare.Text = current["DLucrare"].ToString();
    txtTotal.Text = current["Total"].ToString();
    dateTimePicker1.Value = Convert.ToDateTime(current["DataBonConsum"]);
    bs2.MoveFirst();
    dataSet1.BonConsumContinutManevra.Clear();
    for (int i = 1; i <= bs2.Count; i++)
    {
        current = (DataRowView)bs2.Current;
        DataSet1.BonConsumContinutManevraRow r =
dataSet1.BonConsumContinutManevra.NewBonConsumContinutManevraRow();
        r.NrCrt = Convert.ToInt16(current["NrCrt"]);
    }
}

```

```

        r.DPiesa = Convert.ToString(current["DPiesa"]);
        r.UM = Convert.ToString(current["UM"]);
        r.PretVanzare = Convert.ToDecimal(current["PretVanzare"]);
        r.Cantitate = Convert.ToDecimal(current["Cantitate"]);
        r.IdPiesaAuto = Convert.ToInt32(current["IdPiesaAuto"]);
        r.Valoare = Convert.ToDecimal(current["Valoare"]);
        dataSet1.BonConsumContinutManevra.Rows.Add(r);
        bs2.MoveNext();
    }
}

private void calcTotal()
{ // Calcul total valori
    decimal t = 0;
    foreach (DataRow r in dataSet1.BonConsumContinutManevra)
    {
        if (r["Valoare"] != DBNull.Value)
            t += (decimal)r["Valoare"];
    }
    txtTotal.Text = Convert.ToString(t);
}

private Boolean verificaStoc()
{
    foreach (DataRow r in dataSet1.BonConsumContinutManevra)
    {
        con.ConnectionString = pieseAutoTableAdapter.Connection.ConnectionString;
        cmd.Connection = con;
        cmd.CommandText = "SELECT StocFinal FROM BalantaStoc bs LEFT OUTER JOIN
PieseAuto pa ON bs.IdPiesaAuto=pa.IdPiesaAuto WHERE pa.DPiesa=\"\" + r["DPiesa"]+"\"";
        con.Open();
        rdr = cmd.ExecuteReader();
        rdr.Read();
        int stoc = rdr.GetInt32(0);
        //MessageBox.Show("Cantitate in stoc: "+stoc);
        rdr.Close();
        con.Close();
        if (r["Cantitate"] != DBNull.Value)
        {
            if (Convert.ToInt32(r["Cantitate"]) > stoc)
            {
                MessageBox.Show("Cantitate insuficienta in stoc pentru " +
r["DPiesa"]+ ". Cantitate in stoc: " + stoc);
                dataGridView1.Focus();
                return false;
            }
        }
    }
    return true;
}

private void generez_nrB()
{
    con.ConnectionString = pieseAutoTableAdapter.Connection.ConnectionString;
    cmd.Connection = con;
    cmd.CommandText = "SELECT Max(BonuriConsum.NrBonConsum) AS NrMax FROM
BonuriConsum";
    con.Open();

```

```

        rdr = cmd.ExecuteReader();
        rdr.Read();
        txtNrBonConsum.Text = Convert.ToString(rdr.GetInt32(0) + 1);
        rdr.Close();
        con.Close();
    }

    private bool validareCampuriObligatorii()
    {
        //Validare de completare obligatorie campul Client
        if (cmbClient.Text == "")
        {
            MessageBox.Show("Completati Client !");
            cmbClient.Focus();
            return false;
        }
        //Validare de completare obligatorie campul Lucrare
        if (cmbLucrare.Text == "")
        {
            MessageBox.Show("Completati Lucrarea !");
            cmbLucrare.Focus();
            return false;
        }
        // Validare completare continuut
        if (bonConsumContinutManevraBindingSource.Count == 0)
        {
            MessageBox.Show("Completati continuut bon consum !");
            dataGridView1.Focus();
            return false;
        }
        if (dataGridView1.CurrentRow.Cells[4].Value.ToString() == "")
        {
            MessageBox.Show("Completati cantitatea !");
            dataGridView1.Focus();
            return false;
        }
        return true;
    }

    private void adaugaInregistrareBonuriConsum()
    {
        string listaCampuri;
        string listaValori;
        DateTime d = dateTimePicker1.Value;
        listaCampuri = "NrBonConsum, DataBonConsum, IdClient, IdLucrare";
        listaValori = txtNrBonConsum.Text +
            ",#" + Convert.ToString(d.Month) + "/" +
            Convert.ToString(d.Day) + "/" +
            Convert.ToString(d.Year) + "#," +
            cmbClient.SelectedValue + "," + cmbLucrare.SelectedValue;
        cmd.CommandText = "Insert into BonuriConsum(" + listaCampuri + ") " +
            "Select " + listaValori;
        con.Open();
        cmd.ExecuteNonQuery();
        con.Close();
    }

    private void cautaInregistrare()

```

```

        {
            cmd.CommandText = "SELECT idBonConsum From BonuriConsum Where NrBonConsum = "
+
            txtNrBonConsum.Text;
            con.Open();
            rdr = cmd.ExecuteReader();
            rdr.Read();
            idB = rdr.GetInt32(0);
            rdr.Close();
            con.Close();
        }

        private void adaugaInregistrariBonuriConsumContinut()
        {
            string listaCampuri = "IdBonConsum, NrCrt, IdPiesaAuto, Cantitate,
PretVanzare";
            string listaValori;
            con.Open();
            foreach (DataRow r in dataSet1.BonConsumContinutManevra)
            {
                //con.ConnectionString =
pieseAutoTableAdapter.Connection.ConnectionString;
                cmd.Connection = con;
                cmd.CommandText = "SELECT IdPiesaAuto FROM PieseAuto WHERE DPiesa=\"\" +
r["DPiesa"] + "\"\"";
                //con.Open();
                rdr = cmd.ExecuteReader();
                rdr.Read();
                String IdPiesaAuto =Convert.ToString(rdr.GetInt32(0));
                rdr.Close();
                //con.Close();

                listaValori = idB + "," + r["NrCrt"] + "," + IdPiesaAuto + "," +
r["Cantitate"] + "," + r["PretVanzare"];
                cmd.CommandText = "Insert into BonuriConsumContinut(" + listaCampuri + ")
" +
                "Select " + listaValori;
                //MessageBox.Show(cmd.CommandText);
                cmd.ExecuteNonQuery();
            }
            con.Close();
        }
        private void golireCampuri()
        {
            txtNrBonConsum.Text = "";
            cmbClient.SelectedIndex = -1;
            cmbLucrare.SelectedIndex = -1;
            txtTotal.Text = "";
            dataSet1.BonConsumContinutManevra.Clear();
        }

        private void stergeContinut()
        {
            cmd.CommandText = "Delete from BonuriConsumContinut Where IdBonConsum = " +
idB;
            con.Open();
            cmd.ExecuteNonQuery();
            con.Close();
        }
    }

```

```

    }

    private void modificaInregistrare()
    {
        DateTime d1 = dateTimePicker1.Value;
        con.ConnectionString = pieseAutoTableAdapter.Connection.ConnectionString;
        cmd.Connection = con;
        string clauzaWhere = " Where IdBonConsum = " + idB;
        string clauzaSet = "Set DataBonConsum = #" + d1.Month + "/" + d1.Day + "/" +
        d1.Year + "#," + "IdClient = " + cmbClient.SelectedValue + "," + "IdLucrare = " +
        cmbLucrare.SelectedValue;
        cmd.CommandText = "Update BonuriConsum " + clauzaSet + clauzaWhere;
        con.Open();
        cmd.ExecuteNonQuery();
        con.Close();
    }

    private void bonConsumContinutManevraBindingSource_CurrentChanged(object sender,
    EventArgs e)
    {
        A3();
    }

    private void dataGridView1_CellValueChanged(object sender,
    DataGridViewCellEventArgs e)
    {
        A4();
        calcTotal();
    }

    private void dataGridView1_UserDeletingRow(object sender,
    DataGridViewRowCancelEventArgs e)
    {
        A5(e);
    }

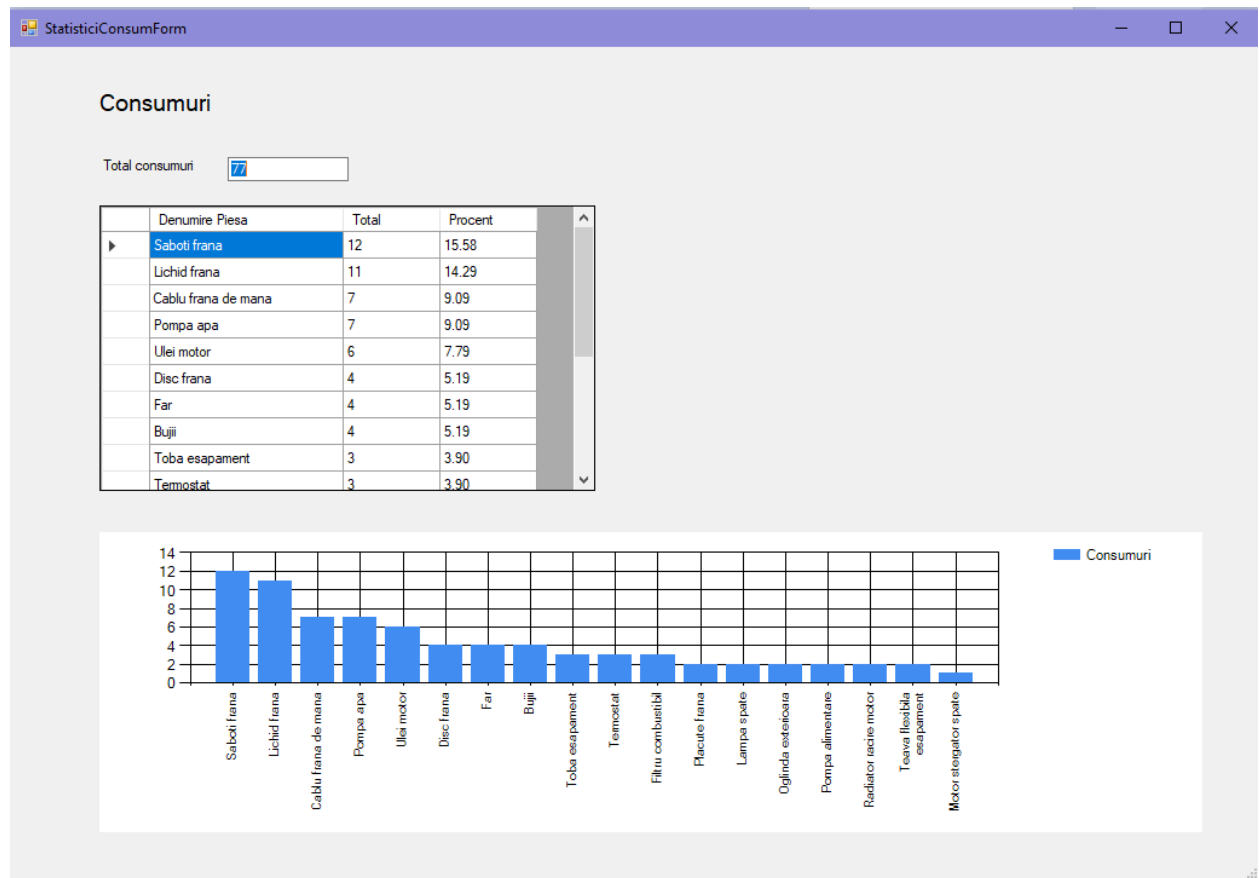
    private void dataGridView1_UserDeletedRow(object sender, DataGridViewRowEventArgs
    e)
    {
        calcTotal();
    }

    private void dataGridView1_DataError(object sender,
    DataGridViewDataErrorEventArgs e)
    {
        MessageBox.Show("Format eronat");
    }

    private void btnConfirmare_Click(object sender, EventArgs e)
    {
        A2();
    }
}

```

## StatisticiConsumForm.cs



### Codul:

```
using System;
using System.Data;
using System.Data.OleDb;
using System.Windows.Forms;

namespace CuzicGabriela
{
    public partial class StatisticiConsumForm : Form
    {
        private OleDbConnection con = new OleDbConnection();
        private OleDbCommand cmd = new OleDbCommand();
        private OleDbDataReader rdr;
        private decimal totalConsumuri;

        public StatisticiConsumForm()
        {
            InitializeComponent();
        }

        private void StatisticiConsumForm_Load(object sender, EventArgs e)
        {
            // TODO: This line of code loads data into the 'statDS.ConsumPieseAuto'
            table. You can move, or remove it, as needed.
        }
    }
}
```

```

this.consumPieseAutoTableAdapter.Fill(this.statDS.ConsumPieseAuto);
//Pregatire con si cmd
con = consumPieseAutoTableAdapter.Connection;
cmd.Connection = con;
//Total consumuri
cmd.CommandText = "select sum(Cantitate) as Total from BonuriConsumContinut";

con.Open();
rdr = cmd.ExecuteReader();
rdr.Read();
totalConsumuri = rdr.GetDecimal(0);
con.Close();
rdr.Close();

txtTotalConsumuri.Text = "" + totalConsumuri;

foreach (DataRow r in statDS.ConsumPieseAuto)
{
    decimal x;
    x = (decimal)r["Total"] / totalConsumuri;
    x = Math.Round(x, 4) * 100;
    x = Math.Round(x, 2);
    r["Procent"] = x;
}
}
}
}

```

## ConsumReportForm.cs

The screenshot shows a Windows application window titled "ConsumReportForm". It features a date range selector with "Data Inceput" (3/ 9/2021) and "Data Sfarait" (5/ 9/2021), and a button labeled "Afiseaza Raport". Below the selector is a table viewer displaying a list of consumptions for the period 3/9/2021 - 5/9/2021. The table has 7 columns: Data Bon Consum, Denumire Lucrare, Nume Client, Denumire Piesa, Pret, Cantitate, and UM. The data is as follows:

| Data Bon Consum | Denumire Lucrare              | Nume Client    | Denumire Piesa        | Pret | Cantitate | UM  |
|-----------------|-------------------------------|----------------|-----------------------|------|-----------|-----|
| 01-May-2021     | Reparatie sistem franare      | Dinescu Adrian | Cablu frana de mana   | 45   | 5         | Buc |
| 01-May-2021     | Reparatie sistem franare      | Dinescu Adrian | Saboti frana          | 65   | 4         | Buc |
| 01-May-2021     | Reparatie sistem franare      | Dinescu Adrian | Lichid frana          | 40   | 2         | l   |
| 09-May-2021     | Reparatie sistem racire motor | Craiu Lucian   | Radiator racire motor | 185  | 1         | Buc |
| 09-May-2021     | Reparatie sistem racire motor | Craiu Lucian   | Pompa apa             | 100  | 5         | Buc |
| 09-May-2021     | Reparatie sistem racire motor | Craiu Lucian   | Pompa apa             | 100  | 5         | Buc |



## Codul:

```
using Microsoft.Reporting.WinForms;
using System;
using System.Windows.Forms;

namespace CuzicGabriela
{
    public partial class ConsumReportForm : Form
    {
        public ConsumReportForm()
        {
            InitializeComponent();

            private void ConsumReportForm_Load(object sender, EventArgs e)
            {
                // TODO: This line of code loads data into the 'ReportDS.Consum' table. You
                can move, or remove it, as needed.
                //this.ConsumTableAdapter.Fill(this.ReportDS.Consum);
                //this.reportViewer1.RefreshReport();
            }

            private void btnAfiseaza_Click(object sender, EventArgs e)
            {
                DateTime d1 = new DateTime(dateTimePicker1.Value.Year,
                dateTimePicker1.Value.Month, dateTimePicker1.Value.Day);
                DateTime d2 = new DateTime(dateTimePicker2.Value.Year,
                dateTimePicker2.Value.Month, dateTimePicker2.Value.Day);
                this.ConsumTableAdapter.Fill(this.ReportDS.Consum, d1, d2);
                //Pregatesc parametrii raportului
                ReportParameter[] parameters = new ReportParameter[1];
                parameters[0] = new ReportParameter("Title", "Lista consumurilor din perioada
                " + d1.ToShortDateString() + " - " +
                d2.ToShortDateString());

                //Comunicare parametrii
                this.reportViewer1.LocalReport.SetParameters(parameters);

                //Emit raport
                this.reportViewer1.RefreshReport();
                this.reportViewer1.DataBindings.Clear();
            }
        }
    }
}
```

### 3.3. Arhivă

#### StocInitialForm.cs

StocInitialForm

Adaugare Actualizare Stergere

Stoc Initial Raport Arhiva Stoc

|   | Data inregistrare | Denumire Piesa auto   | Canitate | Pret | UM  |
|---|-------------------|-----------------------|----------|------|-----|
| ▶ | 2/1/2021          | Lichid frana          | 6        | 40   | I   |
|   | 2/1/2021          | Filtru aer            | 20       | 55   | Buc |
|   | 2/1/2021          | Bujii                 | 25       | 22   | Set |
|   | 5/4/2021          | Pompa apa             | 15       | 100  | Buc |
|   | 5/5/2021          | Radiator              | 23       | 150  | Buc |
|   | 5/5/2021          | Motor stergator spate | 10       | 200  | Buc |
|   | 5/5/2021          | Disc frana            | 30       | 55   | Buc |
|   | 5/5/2021          | Toba esapament        | 18       | 230  | Buc |
|   | 5/5/2021          | Filtru combustibil    | 15       | 25   | Buc |
|   | 5/5/2021          | Ulei motor            | 20       | 50   | I   |

Data 2/ 1/2021

Piesa Auto Lichid frana

Canitate 6

Pret 40

UM I

ID 1

#### Actualizare

StocInitialForm

Adaugare Actualizare Stergere Salvare Renuntare

Stoc Initial Raport Arhiva Stoc

|   | Data inregistrare | Denumire Piesa auto   | Canitate | Pret | UM  |
|---|-------------------|-----------------------|----------|------|-----|
| ▶ | 2/1/2021          | Lichid frana          | 6        | 40   | I   |
|   | 2/1/2021          | Filtru aer            | 20       | 55   | Buc |
|   | 2/1/2021          | Bujii                 | 25       | 22   | Set |
|   | 5/4/2021          | Pompa apa             | 15       | 100  | Buc |
|   | 5/5/2021          | Radiator              | 23       | 150  | Buc |
|   | 5/5/2021          | Motor stergator spate | 10       | 200  | Buc |
|   | 5/5/2021          | Disc frana            | 30       | 55   | Buc |
|   | 5/5/2021          | Toba esapament        | 18       | 230  | Buc |
|   | 5/5/2021          | Filtru combustibil    | 15       | 25   | Buc |
|   | 5/5/2021          | Ulei motor            | 20       | 50   | I   |

Data 2/ 1/2021

Piesa Auto Lichid frana

Canitate 6

Pret 40

UM I

ID 1

ACTUALIZARE

## Adăugare

StocInitialForm

Adaugare Actualizare Stergere Salvare Renuntare

ADAUGARE

Stoc Initial Raport Arhiva Stoc

|   | Data inregistrare | Denumire Piesa auto   | Cantitate | Pret | UM  |
|---|-------------------|-----------------------|-----------|------|-----|
| ▶ | 2/1/2021          | Lichid frana          | 6         | 40   | l   |
|   | 2/1/2021          | Filtru aer            | 20        | 55   | Buc |
|   | 2/1/2021          | Bujii                 | 25        | 22   | Set |
|   | 5/4/2021          | Pompa apa             | 15        | 100  | Buc |
|   | 5/5/2021          | Radiator              | 23        | 150  | Buc |
|   | 5/5/2021          | Motor stergator spate | 10        | 200  | Buc |
|   | 5/5/2021          | Disc frana            | 30        | 55   | Buc |
|   | 5/5/2021          | Toba esapament        | 18        | 230  | Buc |
|   | 5/5/2021          | Filtru combustibil    | 15        | 25   | Buc |
|   | 5/5/2021          | Ulei motor            | 20        | 50   | l   |

Data: 9/2021

Piesa Auto:

Cantitate:

Pret:

UM:

ID:

## Codul:

```
using System;
using System.Windows.Forms;
using System.Data.OleDb;

namespace CuzicGabriela
{
    public partial class StocInitialForm : Form
    {
        public StocInitialForm()
        {
            InitializeComponent();
        }

        private void StocInitialForm_Load(object sender, EventArgs e)
        {
            A1();
        }

        private void A1()
        {
            //Umple cu date obiectele DataTable: StocInitial si PieseAuto (din DataSet):
            pieseAutoTableAdapter.Fill(dataSet1.PieseAuto);
            stocInitialTableAdapter.Fill(dataSet1.StocInitial);
            //Protectie grid
            dataGridView1.AllowUserToAddRows = false;
            dataGridView1.AllowUserToDeleteRows = false;
            dataGridView1.ReadOnly = true;
            //Protectie txtPret
            txtPret.ReadOnly = true;
            txtUM.ReadOnly = true;
            txtId.ReadOnly = true;
            A3();
        }
    }
}
```

```

private void A2()
{
    //Configurare butoane
    configureazaButoane(false);
    //Dezlegare controale Panel
    legareControale(false);
    //Ridicare protectie controale Panel
    protectiePanel(false);
    //Modificare lblOp
    lblOp.Text = "ADAUGARE";
    //Pozitionare cursor pe primul camp
    dateTimePicker1.Focus();
    // Golire campuri
    golireCampuri();
}

private void A3()
{
    //Initializare lblOp
    lblOp.Text = "";
    //Configurare(butoane)
    configureazaButoane(true);
    //Protectie componente Panel
    protectiePanel(true);
    //Legare controale
    legareControale(true);
}

private void A4()
{
    if (lblOp.Text == "ADAUGARE")
    {
        if (!validareCampuriObligatorii()) return;
        adauga_inregistrare();
        golireCampuri();
        //Pune cursor pe primul camp
        dateTimePicker1.Focus();
        refresh_grid(stocInitialBindingSource.Position);
        A2();
    }
    else if (lblOp.Text == "ACTUALIZARE")
    {
        if (!validareCampuriObligatorii()) return;
        modifica_inregistrare();
        refresh_grid(stocInitialBindingSource.Position);
        A3();
    }
    else
        MessageBox.Show("Operatie actualizare neefectuata");
}

private void A5(ComboBox cmbB)
{
    OleDbConnection con = new OleDbConnection();
    OleDbCommand cmd = new OleDbCommand();
    OleDbDataReader r;
    if (lblOp.Text == "") return;
    if (cmbB.Text == "") return;
    if (btnRenuntare.Focused) return;

```

```

        if (cmbB == cmbPiesaAuto)
        {
            DateTime d = dateTimePicker1.Value;
            con.ConnectionString =
stocInitialTableAdapter.Connection.ConnectionString;
            cmd.Connection = con;
            if (lblOp.Text == "ADUGARE")
            {
                //cmd.CommandText = "Select IdStocInitial From StocInitial where
IdPiesaAuto=" + cmbPiesaAuto.SelectedValue + " and Data= #" + d.Month + "/" + d.Day + "/"
+ d.Year + "#";
                cmd.CommandText = "Select IdStocInitial From StocInitial where
IdPiesaAuto=" + cmbPiesaAuto.SelectedValue;
                con.Open();
                r = cmd.ExecuteReader();
                if (r.Read())
                {
                    MessageBox.Show("Situatie existantă deja pentru piesa auto
selectată!");
                    cmbPiesaAuto.Focus();
                }
                con.Close();
            }
            else if (lblOp.Text == "ACTUALIZARE")
            {
                //cmd.CommandText = "Select IdStocInitial From StocInitial where
IdPiesaAuto=" + cmbPiesaAuto.SelectedValue + " and Data= #" + d.Month + "/" + d.Day + "/"
+ d.Year + "# and IdStocInitial<>" + txtId.Text;
                cmd.CommandText = "Select IdStocInitial From StocInitial where
IdPiesaAuto=" + cmbPiesaAuto.SelectedValue + " and IdStocInitial<>" + txtId.Text;
                con.Open();
                r = cmd.ExecuteReader();
                if (r.Read())
                {
                    MessageBox.Show("Situatie existantă deja pentru piesa auto
selectată!");
                    cmbPiesaAuto.Focus();
                }
                con.Close();
            }
        }
    }

private void A5(TextBox txtB)
{
    decimal p;
    if (lblOp.Text == "") return;
    if (txtB.Text == "") return;
    if (btnRenuntare.Focused) return;

    if (txtB == txtCantitate)
    {
        // Validare numericitate
        try { p = Convert.ToDecimal(txtB.Text); }
        catch { MessageBox.Show("Format eronat"); txtB.Focus(); }
    }
}

```

```

private void A7()
{
    //Configurare butoane
    configureazaButoane(false);
    //Dezlegare controale Panel
    legareControale(false);
    //Ridicare protectie controale Panel
    protectiePanel(false);
    //Modificare lblOp
    lblOp.Text = "ACTUALIZARE";
    //Pozitionare cursor pe primul camp
    dateTimePicker1.Focus();
}
private void A8()
{
    OleDbConnection con = new OleDbConnection();
    OleDbCommand cmd = new OleDbCommand();
    con.ConnectionString = pieseAutoTableAdapter.Connection.ConnectionString;
    cmd.Connection = con;
    cmd.CommandText = "Delete From StocInitial Where IdStocInitial = " +
txtId.Text;
    //MessageBox.Show("Inregistrare stearsa cu succes!");
    con.Open();
    cmd.ExecuteNonQuery();
    con.Close();
    refresh_grid(stocInitialBindingSource.Position);
}

private void configureazaButoane(bool v)
{
    btnRenuntare.Visible = !v;
    btnSalvare.Visible = !v;
    btnAdaugare.Enabled = v;
    btnActualizare.Enabled = v;
    btnStergere.Enabled = v;
}
private void legareControale(bool v)
{
    if (v)
    {
        dateTimePicker1.DataBindings.Add("Text", stocInitialBindingSource,
>Data");
        cmbPiesaAuto.DataBindings.Add("Text", stocInitialBindingSource,
>DPiesa");
        //txtPret.DataBindings.Add("Text", pieseAutoBindingSource, "Pret");
        txtId.DataBindings.Add("Text", stocInitialBindingSource,
>IdStocInitial");
        txtCantitate.DataBindings.Add("Text", stocInitialBindingSource,
>Cantitate1");
    }
    else
    {
        dateTimePicker1.DataBindings.Clear();
        cmbPiesaAuto.DataBindings.Clear();
        //txtPret.DataBindings.Clear();
        txtId.DataBindings.Clear();
        txtCantitate.DataBindings.Clear();
    }
}

```

```

    }
}
private void protectiePanel(bool v)
{
    dateTimePicker1.Enabled = !v;
    txtCantitate.ReadOnly = v;
    cmbPiesaAuto.Enabled = !v;
}
private void golireCampuri()
{
    dateTimePicker1.Text = "";
    cmbPiesaAuto.SelectedIndex = -1;
    cmbPiesaAuto.Text = "";
    txtPret.Text = "";
    txtUM.Text = "";
    txtId.Text = "";
    txtCantitate.Text = "";
}
private bool validareCampuriObligatorii()
{
    //Validare de completare obligatorie campurile: Cantitate, PiesaAuto
    if (cmbPiesaAuto.Text == "")
    {
        MessageBox.Show("Alegeti piesa auto!");
        cmbPiesaAuto.Focus();
        return false;
    }
    if (txtCantitate.Text == "")
    {
        MessageBox.Show("Completati cantitatea!");
        txtCantitate.Focus();
        return false;
    }
    if (dateTimePicker1.Text == "")
    {
        MessageBox.Show("Alegeti data!");
        dateTimePicker1.Focus();
        return false;
    }
    return true;
}

private void adauga_inregistrare()
{
    string listaCampuri;
    string listaValori;
    OleDbConnection con = new OleDbConnection();
    OleDbCommand cmd = new OleDbCommand();
    con.ConnectionString = stocInitialTableAdapter.Connection.ConnectionString;
    cmd.Connection = con;
    DateTime d = dateTimePicker1.Value;
    listaCampuri = "Data, IdPiesaAuto, Cantitate, Pret ";
    listaValori = "#" + Convert.ToString(d.Month) + "/"
+ Convert.ToString(d.Day) + "/"
+ Convert.ToString(d.Year) + "#"
+ "," + cmbPiesaAuto.SelectedValue +
+ "," + txtCantitate.Text + ""
+ "," + txtPret.Text + """;

```

```

        cmd.CommandText = "Insert into StocInitial(" + listaCampuri + ") " +
        "Select " + listaValori;
        //MessageBox.Show("Inregistrare adaugata cu succes!");
        con.Open();
        cmd.ExecuteNonQuery();
        con.Close();
        golireCampuri();
        legareControale(false);
    }
    private void refresh_grid(int p)
    {
        pieseAutoTableAdapter.Fill(dataSet1.PieseAuto);
        stocInitialTableAdapter.Fill(dataSet1.StocInitial);
        stocInitialBindingSource.Position = p;
    }

    private void modifica_inregistrare()
    {
        string listaSet;
        OleDbConnection con = new OleDbConnection();
        OleDbCommand cmd = new OleDbCommand();
        DateTime d = dateTimePicker1.Value;
        con.ConnectionString = stocInitialTableAdapter.Connection.ConnectionString;
        cmd.Connection = con;
        listaSet = "Data = #" + d.Month + "/" + d.Day + "/" + d.Year + "#," +
        "IdPiesaAuto = " + cmbPiesaAuto.SelectedValue + "," +
        "Cantitate = '" + txtCantitate.Text + "'," +
        "Pret = '" + txtPret.Text + "'";
        cmd.CommandText = "Update StocInitial Set " + listaSet + " Where
IdStocInitial=" +
        txtId.Text;
        //MessageBox.Show("Actualizare efectuata cu succes!");
        con.Open();
        cmd.ExecuteNonQuery();
        con.Close();
    }

    private void btnAdaugare_Click(object sender, EventArgs e)
    {
        A2();
    }

    private void btnActualizare_Click(object sender, EventArgs e)
    {
        A7();
    }

    private void btnStergere_Click(object sender, EventArgs e)
    {
        const string mesaj = "Confirmati stergerea";
        const string titlu = "Stergere inregistrare";
        var rezultat = MessageBox.Show(mesaj, titlu, MessageBoxButtons.YesNo,
        MessageBoxIcon.Warning);
        if (rezultat == DialogResult.No) return;
        A8();
    }

    private void btnSalvare_Click(object sender, EventArgs e)

```



```

    {
        A4();
    }

    private void btnRenuntare_Click(object sender, EventArgs e)
    {
        A3();
    }

    private void txtCantitate_Leave(object sender, EventArgs e)
    {
        A5(txtCantitate);
    }

    private void cmbPiesaAuto_Leave(object sender, EventArgs e)
    {
        A5(cmbPiesaAuto);
    }

    private void btnRaportStocInitial_Click(object sender, EventArgs e)
    {
        StocInitialReportForm f = new StocInitialReportForm();
        f.ShowDialog();
    }
}
}

```

## StocInitialReportForm.cs

The screenshot shows a Windows application window titled "StocInitialReportForm". The interface includes a header section with input fields for "Denumire piesa", "Data inceput" (3/ 5/2021), and "Data sfarsit" (5/ 7/2021), along with an "Afiseaza raport" button. Below this is a report viewer displaying a table of inventory registrations for the period 3/5/2021 to 5/7/2021.

| Data inregistrare | Denumire Piesa        | Cantitate | Pret | UM  |
|-------------------|-----------------------|-----------|------|-----|
| 04-May-2021       | Pompa apa             | 15        | 100  | Buc |
| 05-May-2021       | Radiator              | 23        | 150  | Buc |
| 05-May-2021       | Motor stergator spate | 10        | 200  | Buc |
| 05-May-2021       | Disc frana            | 30        | 55   | Buc |
| 05-May-2021       | Toba esapament        | 18        | 230  | Buc |
| 05-May-2021       | Filter combustibil    | 15        | 25   | Buc |

StocInitialReportForm

Denumire piesa:  Data inceput:  Data sfarsit:

1 of 1 | 100% | Find | Next

Lista inregistrarilor ce contin "filtru" in denumire, din perioada 1/11/2021-5/9/2021

| Data inregistrare | Denumire Piesa     | Cantitate | Pret | UM  |
|-------------------|--------------------|-----------|------|-----|
| 01-Feb-2021       | Filtru aer         | 20        | 55   | Buc |
| 05-May-2021       | Filtru combustibil | 15        | 25   | Buc |

### Codul:

```
using Microsoft.Reporting.WinForms;
using System;
using System.Windows.Forms;

namespace CuzicGabriela
{
    public partial class StocInitialReportForm : Form
    {
        public StocInitialReportForm()
        {
            InitializeComponent();
        }

        private void StocInitialReportForm_Load(object sender, EventArgs e)
        {
            // TODO: This line of code loads data into the 'ReportDS.StocInitial' table.
            You can move, or remove it, as needed.
            //this.StocInitialTableAdapter.Fill(this.ReportDS.StocInitial);

            //this.reportViewer1.RefreshReport();
        }

        private void btnAfiseaza_Click(object sender, EventArgs e)
        {
            DateTime d1 = new DateTime(dateTimePicker1.Value.Year,
            dateTimePicker1.Value.Month, dateTimePicker1.Value.Day);
            DateTime d2 = new DateTime(dateTimePicker2.Value.Year,
            dateTimePicker2.Value.Month, dateTimePicker2.Value.Day);
            this.StocInitialTableAdapter.Fill(this.ReportDS.StocInitial, d1, d2,
            txtDPiesa.Text);
        }
    }
}
```

```

        //Pregatesc parametrii raportului
        ReportParameter[] parameters = new ReportParameter[1];
        if (txtDPiesa.Text != "")
            parameters[0] = new ReportParameter("Titlu", "Lista inregistrarilor ce
contin \" + txtDPiesa.Text.ToString() + "\" in denumire, din perioada " +
d1.ToShortDateString() + "-" + d2.ToShortDateString());
        else parameters[0] = new ReportParameter("Titlu", "Lista inregistrarilor din
perioada " + d1.ToShortDateString() + "-" + d2.ToShortDateString());

        //Comunicare parametrii
        this.reportViewer1.LocalReport.SetParameters(parameters);

        //Emit raport
        this.reportViewer1.RefreshReport();
    }
}
}

```

### 3.4. Tip resursă

#### PieseAutoForm.cs

#### Vizualizare

The screenshot shows the PieseAutoForm application window. At the top, there are three buttons: "Adaugare", "Actualizare", and "Stergere". Below these, there is a section titled "Piese Auto" with a button "Raport Piese Auto". The main part of the window contains a table with the following data:

|   | Denumire Piesa               | UM  | Pret |
|---|------------------------------|-----|------|
| ▶ | Antigel motor                | l   | 35   |
|   | Axa planetara cu articulatie | Buc | 230  |
|   | Bujii                        | Set | 22   |
|   | Cablu frana de mana          | Buc | 45   |
|   | Catalizator                  | Buc | 550  |
|   | Cilindru receptor            | Buc | 75   |
|   | Disc frana                   | Buc | 60   |
|   | Far                          | Buc | 170  |

To the right of the table, there is a form with the following fields:

- Denumire:
- UM:
- Pret:
- ID:

## Actualizare

PieseAutoForm

Adaugare Actualizare Stergere Salvare Renuntare

Piese Auto Raport Piese Auto

|   | Denumire Piesa               | UM  | Pret |
|---|------------------------------|-----|------|
| ► | Antigel motor                | I   | 35   |
|   | Axa planetara cu articulatie | Buc | 230  |
|   | Bujii                        | Set | 22   |
|   | Cablu frana de mana          | Buc | 45   |
|   | Catalizator                  | Buc | 550  |
|   | Cilindru receptor            | Buc | 75   |
|   | Disc frana                   | Buc | 60   |
|   | Far                          | Buc | 170  |

ACTUALIZARE

Denumire

UM

Pret

ID

## Adăugare

PieseAutoForm

Adaugare Actualizare Stergere Salvare Renuntare

Piese Auto Raport Piese Auto

|   | Denumire Piesa               | UM  | Pret |
|---|------------------------------|-----|------|
| ► | Antigel motor                | I   | 35   |
|   | Axa planetara cu articulatie | Buc | 230  |
|   | Bujii                        | Set | 22   |
|   | Cablu frana de mana          | Buc | 45   |
|   | Catalizator                  | Buc | 550  |
|   | Cilindru receptor            | Buc | 75   |
|   | Disc frana                   | Buc | 60   |
|   | Far                          | Buc | 170  |

ADAUGARE

Denumire

UM

Pret

ID

## Codul:

```
using System;
using System.Windows.Forms;
using System.Data.OleDb;

namespace CuzicGabriela
{
    public partial class PieseAutoForm : Form
    {
        public PieseAutoForm()
        {
            InitializeComponent();

            private void PieseAutoForm_Load(object sender, EventArgs e)
            {
                A1();
            }

            private void A1()
            {
                //Umple cu date obiectele DataTable: PieseAuto (din DataSet):
                pieseAutoTableAdapter.Fill(dataSet1.PieseAuto);
                //Protectie grid
                dataGridView1.AllowUserToAddRows = false;
                dataGridView1.AllowUserToDeleteRows = false;
                dataGridView1.ReadOnly = true;
                //Protectie txtIdPiesaAuto
                txtIdPiesaAuto.ReadOnly = true;
                A3();
            }

            private void A2()
            {
                //Configurare butoane
                configureazaButoane(false);
                //Dezlegare controale Panel
                legareControale(false);
                //Ridicare protectie controale Panel
                protectiePanel(false);
                //Modificare lblOp
                lblOp.Text = "ADAUGARE";
                //Pozitionare cursor pe primul camp
                txtDPiesaAuto.Focus();
                // Golire campuri
                golireCampuri();
            }

            private void A3()
            {
                //Initializare lblOp
                lblOp.Text = "";
                //Configurare(butoane)
                configureazaButoane(true);
                //Protectie componente Panel
                protectiePanel(true);
                //Legare controale
                legareControale(true);
            }
        }
    }
}
```

```

}
private void A4()
{
    if (lblOp.Text == "ADAUGARE")
    {
        if (!validareCampuriObligatorii()) return;
        adauga_inregistrare();
        golireCampuri();
        //Pune cursor pe primul camp
        txtDPiesaAuto.Focus();
        refresh_grid(pieseAutoBindingSource.Position);
    }
    else if (lblOp.Text == "ACTUALIZARE")
    {
        if (!validareCampuriObligatorii()) return;
        modifica_inregistrare();
        refresh_grid(pieseAutoBindingSource.Position);
        A3();
    }
    else
        MessageBox.Show("Operatie actualizare neefectuata");
}

private void A5(TextBox txtB)
{
    decimal p;
    OleDbConnection con = new OleDbConnection();
    OleDbCommand cmd = new OleDbCommand();
    OleDbDataReader r;
    if (lblOp.Text == "") return;
    if (txtB.Text == "") return;
    if (btnRenuntare.Focused) return;

    if (txtB == txtPret)
    {
        // Validare numericitate
        try { p = Convert.ToDecimal(txtB.Text); }
        catch { MessageBox.Show("Format eronat"); txtB.Focus(); }
    }
    else if (txtB == txtDPiesaAuto)
    {
        con.ConnectionString = pieseAutoTableAdapter.Connection.ConnectionString;
        cmd.Connection = con;
        if (lblOp.Text == "ADAUGARE")
        {
            cmd.CommandText = "Select IdPiesaAuto From PieseAuto where DPiesa='"
+ txtDPiesaAuto.Text + "'";
            con.Open();
            r = cmd.ExecuteReader();
            if (r.Read())
            {
                MessageBox.Show("Piesa auto deja existentă!");
                txtDPiesaAuto.Focus();
            }
            con.Close();
        }
        else if (lblOp.Text == "ACTUALIZARE")
        {

```

```

        cmd.CommandText = "Select IdPiesaAuto From PieseAuto where DPiesa='"
+ txtDPiesaAuto.Text +
        "' and IdPiesaAuto<> " + txtIdPiesaAuto.Text;
        con.Open();
        r = cmd.ExecuteReader();
        if (r.Read())
        {
            MessageBox.Show("Piesa auto deja existentă!");
            txtDPiesaAuto.Focus();
        }
        con.Close();
    }
}

private void A7()
{
    //Configurare butoane
    configureazaButoane(false);
    //Dezlegare controale Panel
    legareControale(false);
    //Ridicare protectie controale Panel
    protectiePanel(false);
    //Modificare lblOp
    lblOp.Text = "ACTUALIZARE";
    //Pozitionare cursor pe primul camp
    txtDPiesaAuto.Focus();
}

private void A8()
{
    OleDbConnection con = new OleDbConnection();
    OleDbCommand cmd = new OleDbCommand();
    OleDbDataReader r;
    con.ConnectionString = pieseAutoTableAdapter.Connection.ConnectionString;
    cmd.Connection = con;
    cmd.CommandText = "Select IdBonConsum,NrCrt From BonuriConsumContinut where
IdPiesaAuto=" + txtIdPiesaAuto.Text;
    con.Open();
    r = cmd.ExecuteReader();
    if (r.Read())
    {
        MessageBox.Show("Piesa auto referită in tabela BonuriConusumContinut! Nu
se poate sterge!");
        con.Close();
        return;
    }
    con.Close();
    con.ConnectionString = pieseAutoTableAdapter.Connection.ConnectionString;
    cmd.Connection = con;
    cmd.CommandText = "Select IdFactura,NrCrt From FacturiContinut where
IdPiesaAuto=" + txtIdPiesaAuto.Text;
    con.Open();
    r = cmd.ExecuteReader();
    if (r.Read())
    {
        MessageBox.Show("Piesa auto referită in tabela FacturiContinut! Nu se
poate sterge!");
        con.Close();
    }
}

```

```

        return;
    }
    con.Close();
    con.ConnectionString = pieseAutoTableAdapter.Connection.ConnectionString;
    cmd.Connection = con;
    cmd.CommandText = "Delete From PieseAuto Where IdPiesaAuto = " +
txtIdPiesaAuto.Text;
    //MessageBox.Show("Inregistrare stearsa cu succes!");
    con.Open();
    cmd.ExecuteNonQuery();
    con.Close();
    refresh_grid(pieseAutoBindingSource.Position);
}

private void configureazaButoane(bool v)
{
    btnRenuntare.Visible = !v;
    btnSalvare.Visible = !v;
    btnAdaugare.Enabled = v;
    btnActualizare.Enabled = v;
    btnStergere.Enabled = v;
}
private void legareControale(bool v)
{
    if (v)
    {
        txtDPiesaAuto.DataBindings.Add("Text", pieseAutoBindingSource, "DPiesa");
        cmbUM.DataBindings.Add("Text", pieseAutoBindingSource, "UM");
        txtPret.DataBindings.Add("Text", pieseAutoBindingSource, "Pret");
        txtIdPiesaAuto.DataBindings.Add("Text", pieseAutoBindingSource,
"IdPiesaAuto");
    }
    else
    {
        txtDPiesaAuto.DataBindings.Clear();
        cmbUM.DataBindings.Clear();
        txtPret.DataBindings.Clear();
        txtIdPiesaAuto.DataBindings.Clear();
    }
}
private void protectiePanel(bool v)
{
    txtDPiesaAuto.ReadOnly = v;
    txtPret.ReadOnly = v;
    cmbUM.Enabled = !v;
}
private void golireCampuri()
{
    txtDPiesaAuto.Text = "";
    cmbUM.SelectedIndex = -1;
    txtPret.Text = "";
    txtIdPiesaAuto.Text = "";
}
private bool validareCampuriObligatorii()
{
    //Validare de completare obligatorie campurile: Denumire, UM, Pret
    if (txtDPiesaAuto.Text == "")
    {

```



```

        MessageBox.Show("Completati denumirea piesei auto!");
        txtDPiesaAuto.Focus();
        return false;
    }
    if (txtPret.Text == "")
    {
        MessageBox.Show("Completati pretul piesei auto!");
        txtPret.Focus();
        return false;
    }
    if (cmbUM.Text == "")
    {
        MessageBox.Show("Completati unitatea de masura!");
        cmbUM.Focus();
        return false;
    }
    return true;
}

private void adauga_inregistrare()
{
    string listaCampuri;
    string listaValori;
    OleDbConnection con = new OleDbConnection();
    OleDbCommand cmd = new OleDbCommand();
    con.ConnectionString = pieseAutoTableAdapter.Connection.ConnectionString;
    cmd.Connection = con;
    listaCampuri = "DPiesa, Pret, UM";
    listaValori = "'" + txtDPiesaAuto.Text + "'" +
        ", '" + txtPret.Text + "'" +
        ", '" + cmbUM.Text + "'";
    cmd.CommandText = "Insert into PieseAuto(" + listaCampuri + ") " +
        "Select " + listaValori;
    //MessageBox.Show("Inregistrare adaugata cu succes!");
    con.Open();
    cmd.ExecuteNonQuery();
    con.Close();
}

private void refresh_grid(int p)
{
    pieseAutoTableAdapter.Fill(dataSet1.PieseAuto);
    pieseAutoBindingSource.Position = p;
}

private void modifica_inregistrare()
{
    string listaSet;
    OleDbConnection con = new OleDbConnection();
    OleDbCommand cmd = new OleDbCommand();
    con.ConnectionString = pieseAutoTableAdapter.Connection.ConnectionString;
    cmd.Connection = con;
    listaSet = "DPiesa = '" + txtDPiesaAuto.Text + "', " +
        "Pret = '" + txtPret.Text + "', " +
        "UM = '" + cmbUM.Text + "'";
    cmd.CommandText = "Update PieseAuto Set " + listaSet + " Where IdPiesaAuto="
+
    txtIdPiesaAuto.Text;
    //MessageBox.Show("Actualizare efectuata cu succes!");

```

```

        con.Open();
        cmd.ExecuteNonQuery();
        con.Close();
    }

    private void btnAdaugare_Click(object sender, EventArgs e)
    {
        A2();
    }

    private void btnActualizare_Click(object sender, EventArgs e)
    {
        A7();
    }

    private void btnStergere_Click(object sender, EventArgs e)
    {
        const string mesaj = "Confirmati stergerea";
        const string titlu = "Stergere inregistrare";
        var rezultat = MessageBox.Show(mesaj, titlu, MessageBoxButtons.YesNo,
        MessageBoxIcon.Warning);
        if (rezultat == DialogResult.No) return;
        A8();
    }

    private void btnSalvare_Click(object sender, EventArgs e)
    {
        A4();
    }

    private void btnRenuntare_Click(object sender, EventArgs e)
    {
        A3();
    }

    private void txtPret_Leave(object sender, EventArgs e)
    {
        A5(txtPret);
    }

    private void txtDPiesaAuto_Leave(object sender, EventArgs e)
    {
        A5(txtDPiesaAuto);
    }

    private void btnRaportPieseAuto_Click(object sender, EventArgs e)
    {
        PieseAutoReportForm f = new PieseAutoReportForm();
        f.ShowDialog();
    }
}
}

```

## PieseAutoReportForm.cs

The application window is titled "PieseAutoReportForm". It features three input fields at the top: "Denumire Piesa auto" (empty), "Pret minim" (50), and "Pret maxim" (70). A blue button labeled "Afiseaza raport" is positioned to the right of these fields.

The main content area displays a report titled "Lista Pieselor auto la pretul de 50-70 lei". Below the title is a table with three columns: "Denumire Piesa", "Pret", and "UM".

| Denumire Piesa     | Pret | UM  |
|--------------------|------|-----|
| Filtru ulei        | 60   | Buc |
| Filtru aer         | 55   | Buc |
| Ulei motor         | 50   | l   |
| Disc frana         | 60   | Buc |
| Termostat          | 50   | Buc |
| Saboti frana       | 65   | Buc |
| Ulei servodirectie | 55   | l   |

The bottom screenshot shows the same application after the "Denumire Piesa auto" field has been filled with "filtru". The report title is now "Lista Pieselor auto ce contin 'filtru' in denumire, la pretul de 50-70 lei". The table below shows only the parts containing "filtru" in their names.

| Denumire Piesa | Pret | UM  |
|----------------|------|-----|
| Filtru ulei    | 60   | Buc |
| Filtru aer     | 55   | Buc |

### Codul:

```
using Microsoft.Reporting.WinForms;  
using System;  
using System.Windows.Forms;
```

```

namespace CuzicGabriela
{
    public partial class PieseAutoReportForm : Form
    {
        public PieseAutoReportForm()
        {
            InitializeComponent();
        }

        private void PieseAutoReportForm_Load(object sender, EventArgs e)
        {
            // TODO: This line of code loads data into the 'ReportDS.PieseAuto' table.
            You can move, or remove it, as needed.
            //this.PieseAutoTableAdapter.Fill(this.ReportDS.PieseAuto);
        }

        private void btnAfiseaza_Click(object sender, EventArgs e)
        {
            // Validare numericitate
            decimal p1,p2;
            try {
                p1 = Convert.ToDecimal(txtPretMinim.Text);
                p2 = Convert.ToDecimal(txtPretMaxim.Text);
                this.PieseAutoTableAdapter.Fill(this.ReportDS.PieseAuto, txtDPiesa.Text,
p1, p2);

                //Pregatesc parametrii raportului
                ReportParameter[] parameters = new ReportParameter[1];
                if (txtDPiesa.Text != "")
                    parameters[0] = new ReportParameter("Titlu", "Lista Pieselor auto ce
contin \"\" + txtDPiesa.Text.ToString() + "\" in denumire, la pretul de \" +
txtPretMinim.Text + \"-\" + txtPretMaxim.Text+\" lei\");
                else parameters[0] = new ReportParameter("Titlu", "Lista Pieselor auto la
pretul de \" + txtPretMinim.Text + \"-\" + txtPretMaxim.Text+ \" lei\");

                //Comunicare parametruii
                this.reportViewer1.LocalReport.SetParameters(parameters);

                //Emit raport
                this.reportViewer1.RefreshReport();
            }
            catch { MessageBox.Show("Completati pretul minim si maxim in format
numeric!"); txtPretMinim.Focus(); }
        }
    }
}

```

### 3.5. Resursă

Am folosit form-uri pentru prezentarea evaluării resursei printr-o balanță, utilizând dataSet-ul **BalDS.xsd** și clasa **CalculStoc.cs**.

## Codul clasei CalculeStoc.cs

```
using System;
using System.Data.OleDb;
using System.Windows.Forms;

namespace CuzicGabriela
{
    class CalculeStoc
    {
        private static OleDbConnection con = new OleDbConnection();
        private static OleDbCommand cmd = new OleDbCommand();

        public static void calculStoc(string cs)
        {
            con.ConnectionString = cs;
            cmd.Connection = con;
            con.Open();
            //Sterg continut tabelei CalculStoc
            cmd.CommandText = "Delete * from CalculStoc";
            cmd.ExecuteNonQuery();

            stocInitial(); intrari(); iesiri();stocFinal();
            con.Close();
        }

        private static void stocInitial()
        {
            cmd.CommandText = "Insert into CalculStoc(IdOperatie, Data, IdPiesaAuto, Cantitate) " +
                               "Select 1,Data, IdPiesaAuto, Cantitate " +
                               "From StocInitial";

            try { cmd.ExecuteNonQuery(); }
            catch (Exception ex) { MessageBox.Show(ex.Message); }
        }

        private static void intrari()
        {
            cmd.CommandText = "Insert into CalculStoc(IdOperatie, Data, IdPiesaAuto, Cantitate) " +
                               "Select 2,Facturi.DataFactura as Data, FacturiContinut.IdPiesaAuto, FacturiContinut.Cantitate " +
                               "From FacturiContinut LEFT OUTER JOIN Facturi ON FacturiContinut.IdFactura=Facturi.IdFactura";
            try { cmd.ExecuteNonQuery(); }
            catch (Exception ex) { MessageBox.Show(ex.Message); }
        }

        private static void iesiri()
        {
            cmd.CommandText = "Insert into CalculStoc(IdOperatie, Data, IdPiesaAuto, Cantitate) " +
                               "Select 3,BonuriConsum.DataBonConsum as Data, BonuriConsumContinut.IdPiesaAuto, -BonuriConsumContinut.Cantitate " +
                               "From BonuriConsumContinut LEFT OUTER JOIN BonuriConsum ON BonuriConsumContinut.IdBonConsum=BonuriConsum.IdBonConsum";
            try { cmd.ExecuteNonQuery(); }
            catch (Exception ex) { MessageBox.Show(ex.Message); }
        }
    }
}
```

```

    }
    private static void stocFinal()
    {

        // Calculez totaluri pe Cantitate si grupez pe IdPiesaAuto

        cmd.CommandText = "INSERT INTO CalculStoc ( IdOperatie, IdPiesaAuto,
Cantitate, Data ) " +
                        "SELECT 4,CalculStoc.IdPiesaAuto,
Sum(CalculStoc.Cantitate) AS Cantitate, now() " +
                        "FROM CalculStoc GROUP BY CalculStoc.IdPiesaAuto";
        cmd.ExecuteNonQuery();
    }
}
}

```

## BalantaForm.cs

The screenshot shows a Windows Forms application titled 'BalantaForm'. It contains two data grids. The left grid, titled 'Calcul Stoc', shows a list of stock calculations with columns for Date, Denumire Piesa, Operatie, and Cantitate. The right grid, titled 'Balanta Stoc', shows a detailed stock balance with columns for Denumire Piesa, StocInitial, Achizitii, Consumuri, StocFinal, Pret, and UM. A 'Reprezentare grafica' button is visible above the right grid.

| Data             | Denumire Piesa | Operatie     | Cantitate |
|------------------|----------------|--------------|-----------|
| 5/5/2021         | Ulei motor     | Stoc Initial | 20        |
| 2/1/2021         | Ulei motor     | Achizitii    | 20        |
| 3/7/2021         | Ulei motor     | Achizitii    | 12        |
| 2/10/2021        | Ulei motor     | Consumuri    | -2        |
| 5/9/2021         | Ulei motor     | Consumuri    | -4        |
| 5/9/2021 4:47 PM | Ulei motor     | Stoc Final   | 46        |

| Denumire Piesa        | StocInitial | Achizitii | Consumuri | StocFinal | Pret | UM  |
|-----------------------|-------------|-----------|-----------|-----------|------|-----|
| Filtru aer            | 20          |           |           | 20        | 55   | Buc |
| Bujii                 | 25          |           | -4        | 21        | 22   | Set |
| Ulei motor            | 20          | 32        | -6        | 46        | 50   | I   |
| Placuta frana         | 0           | 30        | -2        | 28        | 165  | Set |
| Lichid frana          | 15          |           | -11       | 4         | 40   | I   |
| Antigel motor         | 30          |           |           | 30        | 35   | I   |
| Pompa alimentara      | 14          |           | -2        | 12        | 120  | Buc |
| Radiator racire motor | 3           |           | -2        | 1         | 185  | Buc |
| Disc frana            | 30          |           | -4        | 26        | 60   | Buc |
| Radiator              | 23          |           |           | 23        | 150  | Buc |
| Pompa apa             | 15          |           | -7        | 8         | 100  | Buc |

## Codul:

```

using System;
using System.Windows.Forms;
using System.Data.OleDb;

namespace CuzicGabriela
{
    public partial class BalantaForm : Form
    {
        private static OleDbConnection con = new OleDbConnection();
        private static OleDbCommand cmd = new OleDbCommand();
        public BindingSource bs1;
        public BalantaForm()
        {
            InitializeComponent();
        }

        private void BalantaForm_Load(object sender, EventArgs e)
        {
            con.ConnectionString = balantaTableAdapter.Connection.ConnectionString;
            cmd.Connection = con;
        }
    }
}

```

```

con.Open();

//Sterg continut tabela BalantaStoc
cmd.CommandText = "Delete * from BalantaStoc";
cmd.ExecuteNonQuery();
con.Close();
//Calcul stoc
CalculStoc.calculStoc(balantaTableAdapter.Connection.ConnectionString);
calculStocTableAdapter.Fill(balDS.CalculStoc);

con.Open();
cmd.CommandText = "INSERT INTO BalantaStoc ( IdPiesaAuto, StocFinal )" +
    " SELECT IdPiesaAuto, Cantitate FROM CalculStoc Where
IdOperatie=4";
cmd.ExecuteNonQuery();
con.Close();
//Afisez balanta
balantaTableAdapter.Fill(balDS.Balanta);
con.Open();
cmd.CommandText = "UPDATE BalantaStoc LEFT JOIN StocInitial ON
BalantaStoc.IdPiesaAuto = StocInitial.IdPiesaAuto " +
    "SET BalantaStoc.StocInitial = StocInitial.Cantitate";
cmd.ExecuteNonQuery();
con.Close();
//Afisez balanta
balantaTableAdapter.Fill(balDS.Balanta);
BalDSTableAdapters.AchizitiiTableAdapter taA = new
BalDSTableAdapters.AchizitiiTableAdapter();
taA.Fill(balDS.Achizitii);

DataRelation balAchizitii =
balDS.Relations.Add(balDS.Balanta.Columns["IdPiesaAuto"],
balDS.Achizitii.Columns["IdPiesaAuto"]);

foreach (DataRow rb in balDS.Balanta)
{
    foreach (DataRow ri in rb.GetChildRows(balAchizitii))
    {
        rb["Achizitii"] = ri["Total"];
    }
}
BalDSTableAdapters.ConsumuriTableAdapter taC = new
BalDSTableAdapters.ConsumuriTableAdapter();
taC.Fill(balDS.Consumuri);

DataRelation balConsumuri =
balDS.Relations.Add(balDS.Balanta.Columns["IdPiesaAuto"],
balDS.Consumuri.Columns["IdPiesaAuto"]);

foreach (DataRow rb in balDS.Balanta)
{
    foreach (DataRow ri in rb.GetChildRows(balConsumuri))
    {
        rb["Consumuri"] = ri["Total"];
    }
}
bs1 = balantaBindingSource;
DataRowView current = (DataRowView)bs1.Current;

```

```

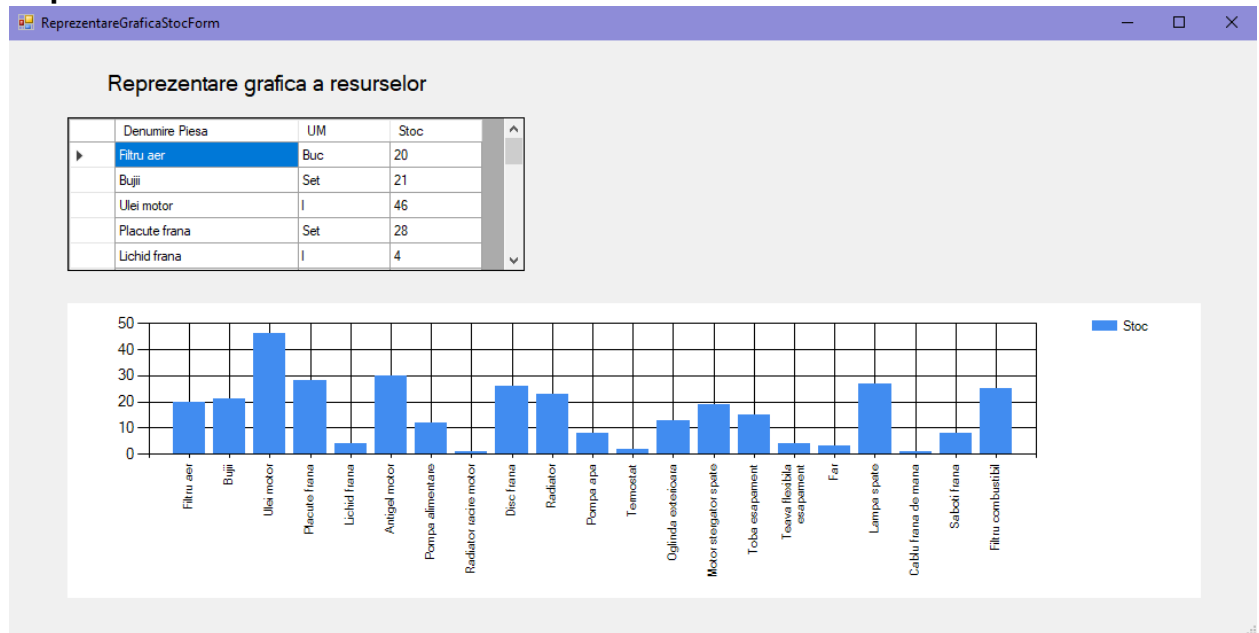
        try
        {
            calculStocBindingSource.Filter = "IdPiesaAuto=" +
current["IdPiesaAuto"].ToString();
        }
        catch { }
    }

    private void balantaBindingSource_PositionChanged(object sender, EventArgs e)
    {
        bs1 = balantaBindingSource;
        DataRowView current = (DataRowView)bs1.Current;
        try
        {
            calculStocBindingSource.Filter = "IdPiesaAuto=" +
current["IdPiesaAuto"].ToString();
        }
        catch { }
    }

    private void btnGrafic_Click(object sender, EventArgs e)
    {
        ReprezentareGraficaStocForm f = new ReprezentareGraficaStocForm();
        f.ShowDialog();
    }
}

```

## ReprezentareGraficaStocForm.cs



Codul:

```

using System;
using System.Windows.Forms;

```



```

namespace CuzicGabriela
{
    public partial class ReprezentareGraficaStocForm : Form
    {
        public ReprezentareGraficaStocForm()
        {
            InitializeComponent();
        }

        private void ReprezentareGraficaStocForm_Load(object sender, EventArgs e)
        {
            this.stocTableAdapter.Fill(this.dataSet1.Stoc);
        }
    }
}

```

## 4. Baza de date

| Conturi    |            |  |
|------------|------------|--|
| Field Name | Data Type  |  |
| IdCont     | AutoNumber |  |
| username   | Short Text |  |
| parola     | Short Text |  |

| IdCont | username | parola |
|--------|----------|--------|
| 1      | u01      | p01    |
| 2      | u02      | p02    |
| *      | (New)    |        |

| Clienti    |            |  |
|------------|------------|--|
| Field Name | Data Type  |  |
| IdClient   | AutoNumber |  |
| DClient    | Short Text |  |
| NrTelefon  | Short Text |  |

| Clienti  |                  |            |              |
|----------|------------------|------------|--------------|
| IdClient | DClient          | NrTelefon  | Click to Add |
| 2        | Costea Marian    | 0763363043 |              |
| 3        | Popescu Sorin    | 0761168904 |              |
| 4        | Arsenie Ruxandra | 0738324674 |              |
| 5        | Mirea Daniel     | 0739024674 |              |
| 6        | Luca Eugen       | 0738321234 |              |
| 7        | Craiu Lucian     | 0735678674 |              |
| 8        | Dinescu Adrian   | 0734024674 |              |
| 9        | Manea Iulia      | 0736923674 |              |
| 10       | Oltean Emanuel   | 0738231674 |              |
| *        | (New)            |            |              |

| Furnizori  |            |
|------------|------------|
| Field Name | Data Type  |
| IdFurnizor | AutoNumber |
| DFurnizor  | Short Text |
| NrTelefon  | Short Text |

| Furnizori  |                                |            |              |
|------------|--------------------------------|------------|--------------|
| IdFurnizor | DFurnizor                      | NrTelefon  | Click to Add |
| 1          | AutoKarma SRL                  | 0246286112 |              |
| 2          | SC Euro Parts Distribution SRL | 0246286745 |              |
| 3          | TecDoc SRL                     | 0246286926 |              |
| 4          | PieseAuto Marketplace SRL      | 0243876100 |              |
| 5          | SC E-Autoparts Europa SRL      | 0246992112 |              |
| *          | (New)                          |            |              |

| Lucrari    |            |
|------------|------------|
| Field Name | Data Type  |
| IdLucrare  | AutoNumber |
| DLucrare   | Short Text |

| Lucrari   |                                  |              |
|-----------|----------------------------------|--------------|
| IdLucrare | DLucrare                         | Click to Add |
| 1         | Reparatie sistem franare         |              |
| 2         | Revizie                          |              |
| 3         | Reparatie aprindere, alimentare  |              |
| 4         | Reparatie sistem racire motor    |              |
| 5         | Reparatie optica si vizibilitate |              |
| 6         | Reparatie evacuare               |              |
| *         | (New)                            |              |

| BonuriConsum  |            |
|---------------|------------|
| Field Name    | Data Type  |
| IdBonConsum   | AutoNumber |
| NrBonConsum   | Number     |
| DataBonConsum | Date/Time  |
| IdLucrare     | Number     |
| IdClient      | Number     |

| BonuriConsum |             |               |           |          |       |
|--------------|-------------|---------------|-----------|----------|-------|
| IdBonConsum  | NrBonConsum | DataBonConsum | IdLucrare | IdClient | Click |
| 22           | 1001        | 2/6/2021      | 1         | 2        |       |
| 23           | 1002        | 2/7/2021      | 4         | 3        |       |
| 24           | 1003        | 2/7/2021      | 6         | 7        |       |
| 25           | 1004        | 2/12/2021     | 3         | 5        |       |
| 26           | 1005        | 2/13/2021     | 5         | 4        |       |
| 27           | 1006        | 2/14/2021     | 1         | 2        |       |
| 28           | 1007        | 2/10/2021     | 2         | 2        |       |
| 31           | 1010        | 5/1/2021      | 1         | 8        |       |
| 34           | 1011        | 5/9/2021      | 4         | 7        |       |
| *(New)       |             |               |           | 0        |       |

| BonuriConsumContinut |           |
|----------------------|-----------|
| Field Name           | Data Type |
| IdBonConsum          | Number    |
| NrCrt                | Number    |
| IdPiesaAuto          | Number    |
| Cantitate            | Currency  |
| PretVanzare          | Currency  |

| BonuriConsumContinut |       |            |           |            |  |
|----------------------|-------|------------|-----------|------------|--|
| IdBonConsum          | NrCrt | IdPiesaAut | Cantitate | PretVanz   |  |
| 22                   | 1     | 11         | 4.00      | 60.00 lei  |  |
| 22                   | 2     | 27         | 4.00      | 65.00 lei  |  |
| 22                   | 3     | 6          | 2.00      | 40.00 lei  |  |
| 23                   | 1     | 14         | 1.00      | 50.00 lei  |  |
| 23                   | 2     | 10         | 1.00      | 185.00 lei |  |
| 23                   | 3     | 13         | 2.00      | 100.00 lei |  |
| 24                   | 1     | 19         | 3.00      | 230.00 lei |  |
| 24                   | 2     | 20         | 2.00      | 350.00 lei |  |
| 25                   | 1     | 29         | 3.00      | 25.00 lei  |  |
| 25                   | 2     | 9          | 2.00      | 100.00 lei |  |
| 25                   | 3     | 3          | 4.00      | 22.00 lei  |  |
| 26                   | 1     | 22         | 2.00      | 170.00 lei |  |
| 26                   | 2     | 23         | 2.00      | 200.00 lei |  |
| 26                   | 3     | 15         | 2.00      | 100.00 lei |  |
| 26                   | 4     | 17         | 1.00      | 200.00 lei |  |
| 27                   | 1     | 27         | 4.00      | 65.00 lei  |  |
| 27                   | 2     | 26         | 2.00      | 45.00 lei  |  |
| 27                   | 3     | 6          | 3.00      | 40.00 lei  |  |
| 28                   | 1     | 4          | 2.00      | 50.00 lei  |  |
| 28                   | 2     | 5          | 2.00      | 165.00 lei |  |
| 28                   | 3     | 22         | 2.00      | 170.00 lei |  |
| 28                   | 4     | 6          | 4.00      | 40.00 lei  |  |
| 31                   | 1     | 26         | 5.00      | 45.00 lei  |  |
| 31                   | 2     | 27         | 4.00      | 65.00 lei  |  |
| 31                   | 3     | 6          | 2.00      | 40.00 lei  |  |
| 34                   | 1     | 10         | 1.00      | 185.00 lei |  |

| Facturi     |            |
|-------------|------------|
| Field Name  | Data Type  |
| IdFactura   | AutoNumber |
| NrFactura   | Number     |
| DataFactura | Date/Time  |
| IdFurnizor  | Number     |

| Facturi |          |             |            |              |
|---------|----------|-------------|------------|--------------|
| IdFactu | NrFactor | DataFactura | IdFurnizor | Click to Add |
| 1       | 100      | 2/1/2021    | 1          |              |
| 2       | 101      | 2/3/2021    | 2          |              |
| 3       | 102      | 3/7/2021    | 3          |              |
| 4       | 103      | 4/18/2021   | 4          |              |
| * (New) |          |             |            |              |

| FacturiContinut |           |
|-----------------|-----------|
| Field Name      | Data Type |
| IdFactura       | Number    |
| NrCrt           | Number    |
| IdPiesaAuto     | Number    |
| Cantitate       | Currency  |
| PretAchizitie   | Currency  |

| FacturiContinut |       |            |           |               |              |  |
|-----------------|-------|------------|-----------|---------------|--------------|--|
| IdFactura       | NrCrt | IdPiesaAut | Cantitate | PretAchizitie | Click to Add |  |
| 1               | 1     | 17         | 10.00     | 100.00 lei    |              |  |
| 1               | 2     | 22         | 5.00      | 100.00 lei    |              |  |
| 1               | 3     | 4          | 20.00     | 20.00 lei     |              |  |
| 2               | 1     | 5          | 30.00     | 25.00 lei     |              |  |
| 2               | 2     | 15         | 5.00      | 50.00 lei     |              |  |
| 3               | 1     | 4          | 12.00     | 20.00 lei     |              |  |
| 3               | 2     | 23         | 18.00     | 50.00 lei     |              |  |
| 4               | 1     | 27         | 20.00     | 40.00 lei     |              |  |
| 4               | 2     | 29         | 13.00     | 15.00 lei     |              |  |
| *               |       |            |           |               |              |  |

| StocInitial X |            |
|---------------|------------|
| Field Name    | Data Type  |
| IdStocInitial | AutoNumber |
| Data          | Date/Time  |
| IdPiesaAuto   | Number     |
| Cantitate     | Number     |
| Pret          | Currency   |

| StocInitial X |          |             |           |            |
|---------------|----------|-------------|-----------|------------|
| IdStocInitial | Data     | IdPiesaAuto | Cantitate | Pret       |
| 1             | 2/1/2021 | 6           | 15.00     | 40.00 lei  |
| 2             | 2/1/2021 | 2           | 20.00     | 55.00 lei  |
| 3             | 2/1/2021 | 3           | 25.00     | 22.00 lei  |
| 4             | 5/4/2021 | 13          | 15.00     | 100.00 lei |
| 5             | 5/5/2021 | 12          | 23.00     | 150.00 lei |
| 7             | 5/5/2021 | 17          | 10.00     | 200.00 lei |
| 8             | 5/5/2021 | 11          | 30.00     | 55.00 lei  |
| 9             | 5/5/2021 | 19          | 18.00     | 230.00 lei |
| 10            | 5/5/2021 | 29          | 15.00     | 25.00 lei  |
| 12            | 5/5/2021 | 4           | 20.00     | 50.00 lei  |
| 13            | 5/8/2021 | 7           | 30.00     | 35.00 lei  |
| 14            | 5/8/2021 | 14          | 5.00      | 50.00 lei  |
| 15            | 5/8/2021 | 15          | 10.00     | 100.00 lei |
| 16            | 5/8/2021 | 26          | 8.00      | 45.00 lei  |
| 17            | 5/8/2021 | 23          | 11.00     | 200.00 lei |
| 18            | 5/8/2021 | 22          | 2.00      | 170.00 lei |
| 19            | 5/8/2021 | 20          | 6.00      | 350.00 lei |
| 20            | 5/8/2021 | 9           | 14.00     | 120.00 lei |
| 21            | 5/8/2021 | 10          | 3.00      | 185.00 lei |
| 24            | 5/9/2021 | 5           | 0.00      | 165.00 lei |
| * (New)       |          |             |           |            |

| PieseAuto   |            |
|-------------|------------|
| Field Name  | Data Type  |
| IdPiesaAuto | AutoNumber |
| DPiesa      | Short Text |
| UM          | Short Text |
| Pret        | Currency   |

| PieseAuto   |                              |     |            |
|-------------|------------------------------|-----|------------|
| IdPiesaAuto | DPiesa                       | UM  | Pret       |
| 1           | Filtru ulei                  | Buc | 60.00 lei  |
| 2           | Filtru aer                   | Buc | 55.00 lei  |
| 3           | Bujii                        | Set | 22.00 lei  |
| 4           | Ulei motor                   | l   | 50.00 lei  |
| 5           | Placuta frana                | Set | 165.00 lei |
| 6           | Lichid frana                 | l   | 40.00 lei  |
| 7           | Antigel motor                | l   | 35.00 lei  |
| 8           | Axa planetara cu articulatie | Buc | 230.00 lei |
| 9           | Pompa alimentare             | Buc | 120.00 lei |
| 10          | Radiator racire motor        | Buc | 185.00 lei |
| 11          | Disc frana                   | Buc | 60.00 lei  |
| 12          | Radiator                     | Buc | 150.00 lei |
| 13          | Pompa apa                    | Buc | 100.00 lei |
| 14          | Termostat                    | Buc | 50.00 lei  |
| 15          | Oglinda exterioara           | Buc | 100.00 lei |
| 16          | Ulei compresor               | l   | 45.00 lei  |
| 17          | Motor stergator spate        | Buc | 200.00 lei |
| 18          | Ventilator radiator          | Buc | 125.00 lei |
| 19          | Toba esapament               | Buc | 230.00 lei |
| 20          | Teava flexibila esapament    | Buc | 350.00 lei |
| 21          | Catalizator                  | Buc | 550.00 lei |
| 22          | Far                          | Buc | 170.00 lei |
| 23          | Lampa spate                  | Buc | 200.00 lei |
| 24          | Cilindru receptor            | Buc | 75.00 lei  |
| 25          | Tambur                       | Buc | 105.00 lei |
| 26          | Cablu frana de mana          | Buc | 45.00 lei  |

| BalantaStoc |           |
|-------------|-----------|
| Field Name  | Data Type |
| IdPiesaAuto | Number    |
| StocInitial | Number    |
| Achizitii   | Number    |
| Consumuri   | Number    |
| StocFinal   | Number    |

| BalantaStoc |             |           |           |           |
|-------------|-------------|-----------|-----------|-----------|
| IdPiesaAuto | StocInitial | Achizitii | Consumuri | StocFinal |
| 2           | 20.00       |           |           | 20.00     |
| 3           | 25.00       |           |           | 21.00     |
| 4           | 20.00       |           |           | 46.00     |
| 5           | 0.00        |           |           | 28.00     |
| 6           | 15.00       |           |           | 4.00      |
| 7           | 30.00       |           |           | 30.00     |
| 9           | 14.00       |           |           | 12.00     |
| 10          | 3.00        |           |           | 1.00      |
| 11          | 30.00       |           |           | 26.00     |
| 12          | 23.00       |           |           | 23.00     |
| 13          | 15.00       |           |           | 8.00      |
| 14          | 5.00        |           |           | 2.00      |
| 15          | 10.00       |           |           | 13.00     |
| 17          | 10.00       |           |           | 19.00     |
| 19          | 18.00       |           |           | 15.00     |
| 20          | 6.00        |           |           | 4.00      |
| 22          | 2.00        |           |           | 3.00      |
| 23          | 11.00       |           |           | 27.00     |
| 26          | 8.00        |           |           | 1.00      |
| 27          |             |           |           | 8.00      |
| 29          | 15.00       |           |           | 25.00     |

| CalculStoc  |           |
|-------------|-----------|
| Field Name  | Data Type |
| IdOperatie  | Number    |
| Data        | Date/Time |
| IdPiesaAuto | Number    |
| Cantitate   | Currency  |

| CalculStoc |           |             |           |  |
|------------|-----------|-------------|-----------|--|
| IdOperatie | Data      | IdPiesaAuto | Cantitate |  |
| 1          | 2/1/2021  | 2           | 20.00     |  |
| 4          | #####     | 2           | 20.00     |  |
| 1          | 2/1/2021  | 3           | 25.00     |  |
| 3          | 2/12/2021 | 3           | -4.00     |  |
| 4          | #####     | 3           | 21.00     |  |
| 1          | 5/5/2021  | 4           | 20.00     |  |
| 2          | 2/1/2021  | 4           | 20.00     |  |
| 2          | 3/7/2021  | 4           | 12.00     |  |
| 3          | 5/9/2021  | 4           | -4.00     |  |
| 3          | 2/10/2021 | 4           | -2.00     |  |
| 4          | #####     | 4           | 46.00     |  |
| 1          | 5/9/2021  | 5           | 0.00      |  |
| 2          | 2/3/2021  | 5           | 30.00     |  |
| 3          | 2/10/2021 | 5           | -2.00     |  |
| 4          | #####     | 5           | 28.00     |  |
| 1          | 2/1/2021  | 6           | 15.00     |  |
| 3          | 5/1/2021  | 6           | -2.00     |  |
| 3          | 2/14/2021 | 6           | -3.00     |  |
| 3          | 2/6/2021  | 6           | -2.00     |  |
| 3          | 2/10/2021 | 6           | -4.00     |  |
| 4          | #####     | 6           | 4.00      |  |
| 1          | 5/8/2021  | 7           | 30.00     |  |
| 4          | #####     | 7           | 30.00     |  |
| 1          | 5/8/2021  | 9           | 14.00     |  |
| 3          | 2/12/2021 | 9           | -2.00     |  |

| Operatii   |            |
|------------|------------|
| Field Name | Data Type  |
| IdOperatie | Number     |
| DOperatie  | Short Text |

| Operatii       |           |
|----------------|-----------|
| IdOperatie     | DOperatie |
| 1 Stoc Initial |           |
| 2 Achizitii    |           |
| 3 Consumuri    |           |
| 4 Stoc Final   |           |

Tabelele bazei de date:

| Tables               |  |
|----------------------|--|
| Search...            |  |
| BalantaStoc          |  |
| BonuriConsum         |  |
| BonuriConsumContinut |  |
| CalculStoc           |  |
| Cienti               |  |
| Conturi              |  |
| Facturi              |  |
| FacturiContinut      |  |
| Furnizori            |  |
| Lucrari              |  |
| Operatii             |  |
| PieseAuto            |  |
| StocInitial          |  |