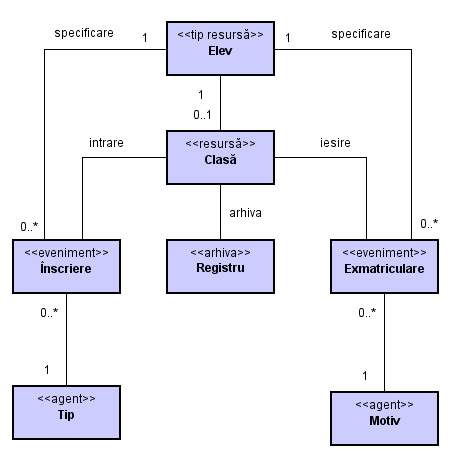
# Medii de Proiectare și Programare

## Tema: Aplicație pentru gestiunea elevilor unui liceu

### Realizat de: Bobu Natalia

#### Diagrama



#### Pagina de start

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void motiveExmatriculareToolStripMenuItem\_Click(object sender, EventArgs e)

{

AgentMotiveExmatriculare agent\_motiv = new AgentMotiveExmatriculare();

agent\_motiv.ShowDialog();

}

private void tipuriInscriereToolStripMenuItem\_Click(object sender, EventArgs e)

{

AgentTipuriInscrieri agent\_tip = new AgentTipuriInscrieri();

agent\_tip.ShowDialog();

}

private void inscriereToolStripMenuItem\_Click(object sender, EventArgs e)

{

EvenimentInscriere evenimentInscriere = new EvenimentInscriere();

evenimentInscriere.ShowDialog();

}

private void exmatriculareToolStripMenuItem\_Click(object sender, EventArgs e)

{

EvenimentExmatriculare evenimentExmatriculare = new EvenimentExmatriculare();

evenimentExmatriculare.ShowDialog();

}

private void claseToolStripMenuItem\_Click(object sender, EventArgs e)

{

ResursaClase resursaClasa = new ResursaClase();

resursaClasa.ShowDialog();

}

private void eleviToolStripMenuItem\_Click(object sender, EventArgs e)

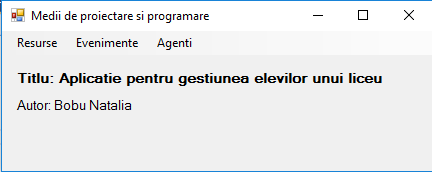
{

ResursaElevi resursaElevi = new ResursaElevi();

resursaElevi.ShowDialog();

}

}



#### Resurse

##### Elevi

public partial class ResursaElevi : Form

{

public ResursaElevi()

{

InitializeComponent();

}

private void ResursaElevi\_Load(object sender, EventArgs e)

{

A1();

}

private void btnAdaugare\_Click(object sender, EventArgs e)

{

A2();

}

private void btnModificare\_Click(object sender, EventArgs e)

{

A5();

}

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;

A6();

}

private void btnConfirmare\_Click(object sender, EventArgs e)

{

A4();

}

private void btnRenuntare\_Click(object sender, EventArgs e)

{

A3();

}

private void A1()

{

//Umple cu date obiectele DataTable

eleviDetaliiTableAdapter.Fill(dataSet1.EleviDetalii);

//Configurare comboBox-uri

cmbSex.DropDownStyle = ComboBoxStyle.Simple;

//Protectie grid

dataGridView1.AllowUserToAddRows = false;

dataGridView1.AllowUserToDeleteRows = false;

dataGridView1.ReadOnly = true;

//Protectie txtIdElev

txtId.ReadOnly = true;

txtId.Visible = false;

A3();

}

private void A2()

{

//Configurare butoane

configureazaButoane(false);

//Dezlegare controale Panel

legareControale(false);

//Ridicare protectie controale Panel

protectiePanel(false);

//Configurare(ComboBox - Uri)

cmbSex.DropDownStyle = ComboBoxStyle.DropDownList;

//Modifcare lblOp

lblOp.Text = "ADAUGARE";

//Pozitionare cursor pe primul camp

txtNume.Focus();

// Golire campuri

golireCampuri();

}

private void A3()

{

//Configurare comboBox-uri

cmbSex.DropDownStyle = ComboBoxStyle.Simple;

//Initializare lblOp

lblOp.Text = "";

//Çonfigurare(butoane)

configureazaButoane(true);

//Protectie componente Panel

protectiePanel(true);

//Legare controale

legareControale(true);

}

private void A4()

{

if (!validareCampuri())

return;

if (lblOp.Text == "ADAUGARE")

{

adauga\_inregistrare();

golireCampuri();

//Pune cursor pe primul camp

txtNume.Focus();

refresh\_grid(eleviDetaliiBindingSource.Position);

}

else if (lblOp.Text == "MODIFICARE")

{

modifica\_inregistrare();

refresh\_grid(eleviDetaliiBindingSource.Position);

A3();

}

else

{

MessageBox.Show("Operatie actualizare neefectuata");

}

}

private void A5()

{

//Configurare butoane

configureazaButoane(false);

//Dezlegare controale Panel

legareControale(true);

//Ridicare protectie controale Panel

protectiePanel(false);

//Configurare(ComboBox - Uri)

cmbSex.DropDownStyle = ComboBoxStyle.DropDownList;

//Modifcare lblOp

lblOp.Text = "MODIFICARE";

//Pozitionare cursor pe primul camp

txtNume.Focus();

}

private void A6()

{

OleDbConnection con = new OleDbConnection();

OleDbCommand cmd = new OleDbCommand();

con.ConnectionString = eleviDetaliiTableAdapter.Connection.ConnectionString;

cmd.Connection = con;

cmd.CommandText = "Delete From Elevi Where IdElev = " + txtId.Text;

con.Open();

try

{

cmd.ExecuteNonQuery();

}catch(Exception e)

{

if (e.HResult == (-2147467259))

{

MessageBox.Show("Nu puteti sterge acest elev !");

return;

}

}

//MessageBox.Show(cmd.CommandText);

con.Close();

refresh\_grid(eleviDetaliiBindingSource.Position);

}

private void configureazaButoane(bool v)

{

btnRenuntare.Visible = !v;

btnConfirmare.Visible = !v;

btnAdaugare.Enabled = v;

btnModificare.Enabled = v;

btnStergere.Enabled = v;

}

private void protectiePanel(bool v)

{

txtNume.Enabled = !v;

txtCNP.Enabled = !v;

cmbSex.Enabled =!v;

dateTimePicker1.Enabled = !v;

}

private void golireCampuri()

{

cmbSex.SelectedIndex = -1;

txtNume.Text = "";

txtId.Text = "";

txtCNP.Text = "";

dateTimePicker1.Value = DateTime.Now;

}

public void clearDataBindings()

{

txtNume.DataBindings.Clear();

txtCNP.DataBindings.Clear();

dateTimePicker1.DataBindings.Clear();

cmbSex.DataBindings.Clear();

txtId.DataBindings.Clear();

}

private void refresh\_grid(int p)

{

dataSet1.Inscrieri.Clear();

eleviDetaliiTableAdapter.Fill(dataSet1.EleviDetalii);

eleviDetaliiBindingSource.Position = p;

}

private void legareControale(bool v)

{

clearDataBindings();

if (v)

{

cmbSex.DataBindings.Add("Text", eleviDetaliiBindingSource, "Sex");

txtNume.DataBindings.Add("Text", eleviDetaliiBindingSource, "Nume");

txtCNP.DataBindings.Add("Text", eleviDetaliiBindingSource, "CNP");

txtId.DataBindings.Add("Text", eleviDetaliiBindingSource, "IdElev");

dateTimePicker1.DataBindings.Add("Text", eleviDetaliiBindingSource, "DataNastere");

}

}

private bool validareCampuri()

{

//Validare de completare obligatorie campurile

if (txtNume.Text == "")

{

MessageBox.Show("Introduceti numele elevului !");

txtNume.Focus();

return false;

}

if (txtCNP.Text == "")

{

MessageBox.Show("Introduceti CNP !");

txtCNP.Focus();

return false;

}

if (cmbSex.Text == "")

{

MessageBox.Show("Selectati sex-ul !");

cmbSex.Focus();

return false;

}

try

{

int a = Int32.Parse(txtCNP.Text);

}

catch (Exception)

{

MessageBox.Show("CNP contine caractere invalide!");

return false;

}

System.String cnp = txtCNP.Text;

foreach(DataRow r in dataSet1.EleviDetalii.Rows)

{

if (String.Equals(cnp, r["CNP"]))

{

MessageBox.Show("CNP duplicat !!!");

txtCNP.Focus();

return false;

}

}

return true;

}

private void modifica\_inregistrare()

{

string listaSet;

OleDbConnection con = new OleDbConnection();

OleDbCommand cmd = new OleDbCommand();

con.ConnectionString = eleviDetaliiTableAdapter.Connection.ConnectionString;

cmd.Connection = con;

listaSet = "Set DataNastere = #" + dateTimePicker1.Value.Month + "/" + dateTimePicker1.Value.Day + "/" + dateTimePicker1.Value.Year + "#," + "Nume = '" + txtNume.Text + "'," + "CNP = '" + txtCNP.Text + "'," + "Sex = '" + cmbSex.Text+"'";

cmd.CommandText = "Update EleviDetalii " + listaSet + " Where IdElev=" + txtId.Text;

MessageBox.Show(cmd.CommandText);

con.Open();

cmd.ExecuteNonQuery();

con.Close();

}

private void adauga\_inregistrare()

{

string listaCampuri;

string listaValori;

OleDbConnection con = new OleDbConnection();

OleDbCommand cmd = new OleDbCommand();

con.ConnectionString = eleviDetaliiTableAdapter.Connection.ConnectionString;

cmd.Connection = con;

listaCampuri = "Nume, CNP, DataNastere, Sex";

listaValori = "'"+txtNume.Text+"','"+txtCNP.Text+"',"+"#" + dateTimePicker1.Value.Month + "/" + dateTimePicker1.Value.Day + "/" + dateTimePicker1.Value.Year + "#,'" + cmbSex.Text+"'";

cmd.CommandText = "Insert into Elevi(" + listaCampuri + ") Select " + listaValori;

MessageBox.Show(cmd.CommandText);

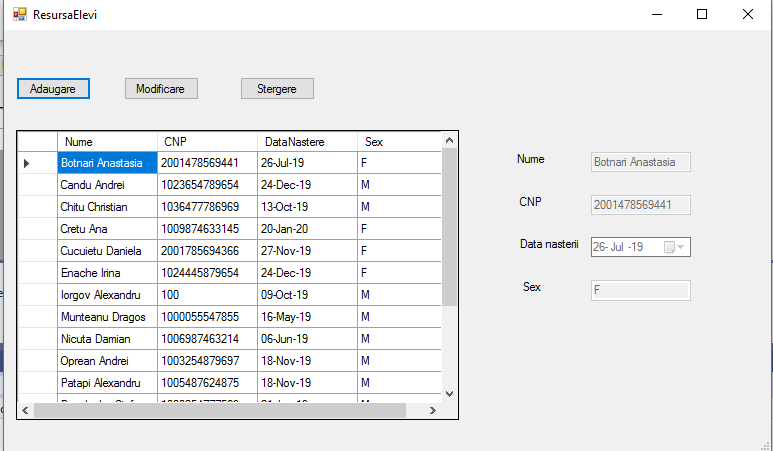
con.Open();

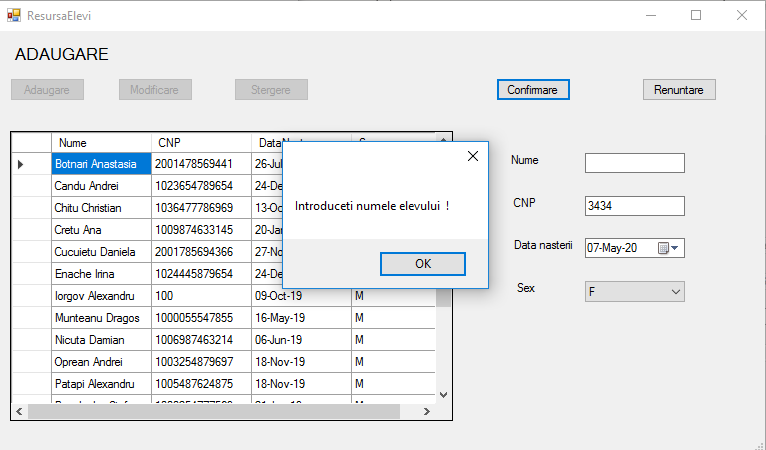
cmd.ExecuteNonQuery();

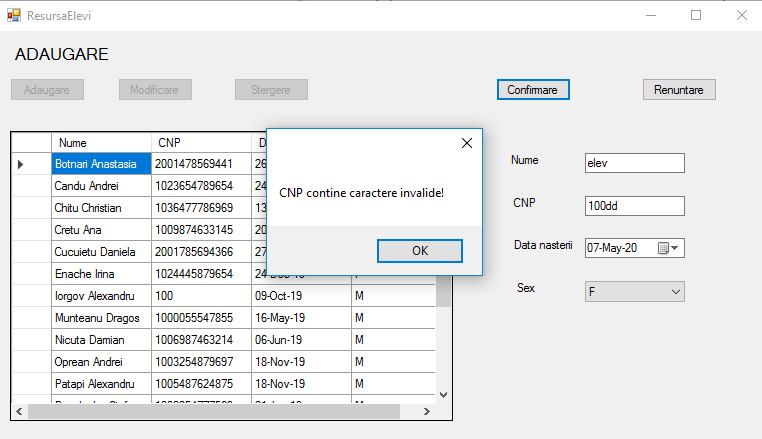
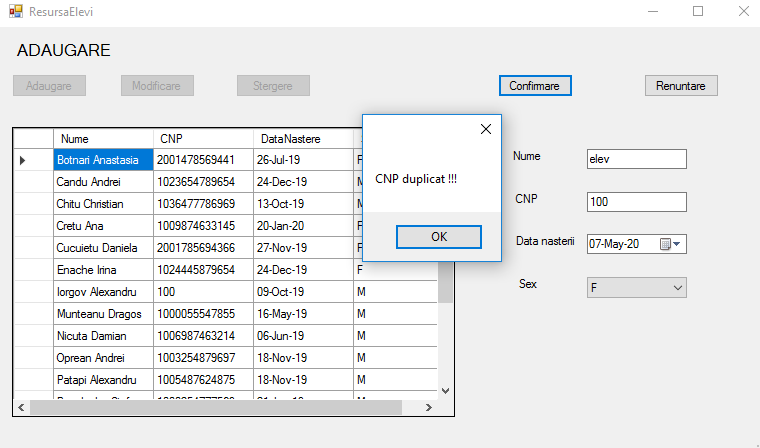
con.Close();

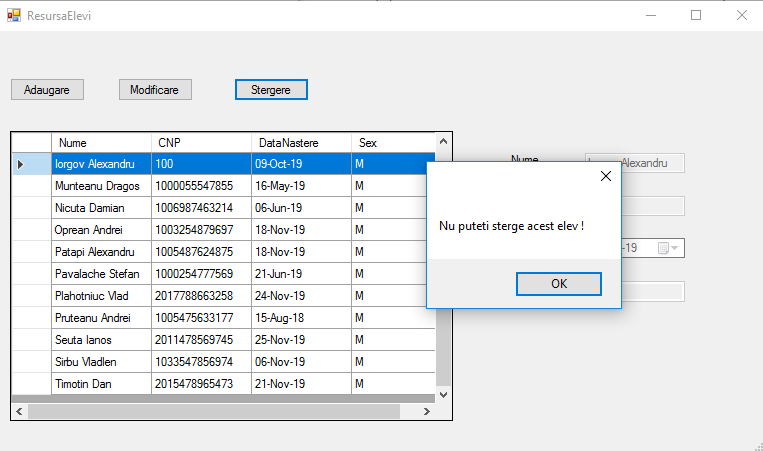
}

}









##### Clase

public partial class ResursaClase : Form

{

public ResursaClase()

{

InitializeComponent();

}

private void ResursaClase\_Load(object sender, EventArgs e)

{

//A1

config(true);

refresh();

}

private void config(bool v)

{

dataGridView1.AllowUserToAddRows = !v;

dataGridView1.AllowUserToDeleteRows = !v;

dataGridView1.ReadOnly = v;

btnActualizare.Enabled = v;

btnSalvare.Visible = !v;

btnRenuntare.Visible = !v;

}

private void refresh()

{

this.claseTableAdapter.Fill(this.dataSet1.Clase);

}

private void dataGridView1\_DataError(object sender, DataGridViewDataErrorEventArgs e)

{

MessageBox.Show("Format eronat");

}

private void dataGridView1\_UserDeletingRow(object sender, 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;

}

private void btnRenuntare\_Click(object sender, EventArgs e)

{

//A3

config(true);

refresh();

}

private void btnSalvare\_Click(object sender, EventArgs e)

{

//A4

try

{

this.claseTableAdapter.Update(this.dataSet1.Clase);

config(true);

refresh();

}

catch (Exception exc)

{

string s = exc.Message;

if (s.IndexOf("duplicate values") > 0)

MessageBox.Show("Inregistrare deja existenta !");

else if (s.IndexOf("cannot be deleted") > 0)

MessageBox.Show("Ati sters inregistrari referite in alte tabele !");

}

}

private void btnActualizare\_Click(object sender, EventArgs e)

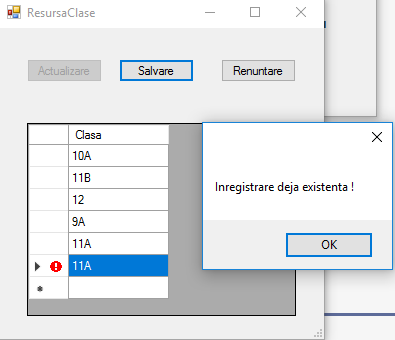
{

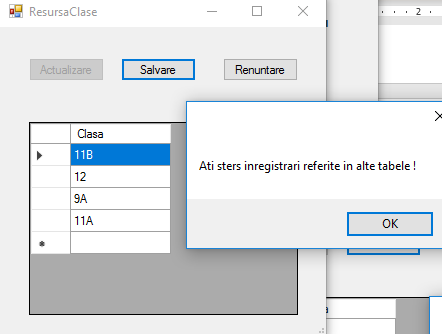
//A2

config(false);

}

}





#### Evenimente

##### Inscriere

public partial class EvenimentInscriere : Form

{

public EvenimentInscriere()

{

InitializeComponent();

}

private void EvenimentInscriere\_Load(object sender, EventArgs e)

{

A1();

}

private void btnAdaugare\_Click(object sender, EventArgs e)

{

A2();

}

private void btnModificare\_Click(object sender, EventArgs e)

{

A5();

}

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;

A6();

}

private void btnConfirmare\_Click(object sender, EventArgs e)

{

A4();

}

private void btnRenuntare\_Click(object sender, EventArgs e)

{

A3();

}

private void A1()

{

//Umple cu date obiectele DataTable

this.eleviTableAdapter.Fill(this.dataSet11.Elevi);

this.claseTableAdapter.Fill(this.dataSet1.Clase);

this.tipuriInscrieriTableAdapter.Fill(this.dataSet1.TipuriInscrieri);

this.inscrieriTableAdapter.Fill(this.dataSet1.Inscrieri);

//Configurare comboBox-uri

configurareComboBoxuri(false);

//Protectie grid

dataGridView1.AllowUserToAddRows = false;

dataGridView1.AllowUserToDeleteRows = false;

dataGridView1.ReadOnly = true;

//Protectie txtIdPersoana

txtNr.ReadOnly = true;

txtNr.Visible = false;

A3();

}

private void A2()

{

//Configurare butoane

configureazaButoane(false);

//Dezlegare controale Panel

legareControale(false);

//Ridicare protectie controale Panel

protectiePanel(false);

//Configurare(ComboBox - Uri)

configurareComboBoxuri(true);

//Legare comboBox-uri la sursa de date

legareComboBoxuri(true);

//Modifcare lblOp

lblOp.Text = "ADAUGARE";

//Pozitionare cursor pe primul camp

dateTimePicker1.Focus();

// Golire campuri

golireCampuri();

}

private void A3()

{

//Configurare comboBox-uri

configurareComboBoxuri(false);

//Initializare lblOp

lblOp.Text = "";

//Çonfigurare(butoane)

configureazaButoane(true);

//Protectie componente Panel

protectiePanel(true);

//Legare controale

legareControale(true);

}

private void A4()

{

if (lblOp.Text == "ADAUGARE")

{

if (!validareCampuri())

return;

adauga\_inregistrare();

golireCampuri();

//Pune cursor pe primul camp

dateTimePicker1.Focus();

refresh\_grid(inscrieriBindingSource.Position);

}

else if (lblOp.Text == "MODIFICARE")

{

modifica\_inregistrare();

refresh\_grid(inscrieriBindingSource.Position);

A3();

}

else

{

MessageBox.Show("Operatie actualizare neefectuata");

}

}

private void A5()

{

//Configurare butoane

configureazaButoane(false);

//Dezlegare controale Panel

legareControale(true);

//Ridicare protectie controale Panel

protectiePanel(false);

//Configurare(ComboBox - Uri)

configurareComboBoxuri(true);

//Legare comboBox-uri la sursa de date

legareComboBoxuri(true);

//Modifcare lblOp

lblOp.Text = "MODIFICARE";

//Pozitionare cursor pe primul camp

dateTimePicker1.Focus();

}

private void A6()

{

OleDbConnection con = new OleDbConnection();

OleDbCommand cmd = new OleDbCommand();

con.ConnectionString = inscrieriTableAdapter.Connection.ConnectionString;

cmd.Connection = con;

cmd.CommandText = "Delete From Inscrieri Where NrInscriere = " + txtNr.Text;

MessageBox.Show(cmd.CommandText);

con.Open();

cmd.ExecuteNonQuery();

con.Close();

refresh\_grid(inscrieriBindingSource.Position);

}

private void configureazaButoane(bool v)

{

btnRenuntare.Visible = !v;

btnConfirmare.Visible = !v;

btnAdaugare.Enabled = v;

btnModificare.Enabled = v;

btnStergere.Enabled = v;

}

private void legareControale(bool v)

{

clearDataBindings();

if (v)

{

cmbStudent.DataBindings.Add("Text", inscrieriBindingSource, "Nume");

cmbType.DataBindings.Add("Text", inscrieriBindingSource, "NumeTip");

cmbClass.DataBindings.Add("Text", inscrieriBindingSource, "Clasa");

txtNr.DataBindings.Add("Text", inscrieriBindingSource, "NrInscriere");

dateTimePicker1.DataBindings.Add("Text", inscrieriBindingSource, "DataInscriere");

}

}

private void protectiePanel(bool v)

{

cmbStudent.Enabled = !v;

cmbType.Enabled = !v;

cmbClass.Enabled = !v;

lblCurrentClass.Visible = !v;

txtCurrentClass.Text = "";

}

private void golireCampuri()

{

cmbStudent.SelectedIndex = -1;

cmbType.SelectedIndex = -1;

cmbClass.SelectedIndex = -1;

txtNr.Text = "";

txtCurrentClass.Text = "";

dateTimePicker1.DataBindings.Clear();

dateTimePicker1.Value = DateTime.Now;

}

private bool validareCampuri()

{

//Validare de completare obligatorie campurile

if (cmbStudent.Text == "")

{

MessageBox.Show("Selectati un elev !");

cmbStudent.Focus();

return false;

}

if (cmbType.Text == "")

{

MessageBox.Show("Selectati tipul inscrierii !");

cmbType.Focus();

return false;

}

if (cmbClass.Text == "") {

MessageBox.Show("Selectati clasa in care va fi inscris elevul !");

cmbClass.Focus();

return false;

}

if((cmbType.Text=="Inceput studii" ||cmbType.Text=="Transfer interscolar")&& txtCurrentClass.Text != "")

{

MessageBox.Show("Acest elev este deja inscris intr-o clasa, selectati un elev care nu are clasa curenta sau alt tip de inscriere !");

cmbType.Focus();

return false;

}

if(cmbType.Text=="Transfer clasa" && txtCurrentClass.Text == cmbClass.Text)

{

MessageBox.Show("Clasa curenta este egala cu clasa de transfer");

cmbClass.Focus();

return false;

}

return true;

}

private void refresh\_grid(int p)

{

dataSet1.Inscrieri.Clear();

inscrieriTableAdapter.Fill(dataSet1.Inscrieri);

inscrieriBindingSource.Position = p;

}

private void adauga\_inregistrare()

{

string listaCampuri;

string listaValori;

OleDbConnection con = new OleDbConnection();

OleDbCommand cmd = new OleDbCommand();

con.ConnectionString = inscrieriTableAdapter.Connection.ConnectionString;

cmd.Connection = con;

listaCampuri = "DataInscriere, IdTip, IdElev, IdClasa";

listaValori = "#" + dateTimePicker1.Value.Month + "/" + dateTimePicker1.Value.Day + "/" + dateTimePicker1.Value.Year + "#," + cmbType.SelectedValue + "," + cmbStudent.SelectedValue+ ","+cmbClass.SelectedValue;

cmd.CommandText = "Insert into Inscrieri(" + listaCampuri + ") Select " + listaValori;

MessageBox.Show(cmd.CommandText);

con.Open();

try

{

cmd.ExecuteNonQuery();

}

catch (Exception e)

{

if (e.HResult == -2147467259)

{

MessageBox.Show("Cheie semantica duplicat! Schimbati data sau selectati alt elev!");

return;

}

}

cmd.CommandText = "Update Registru Set IdClasa=" + cmbClass.SelectedValue + " Where IdElev=" + cmbStudent.SelectedValue;

MessageBox.Show(cmd.CommandText);

cmd.ExecuteNonQuery();

con.Close();

}

private void modifica\_inregistrare()

{

string listaSet;

OleDbConnection con = new OleDbConnection();

OleDbCommand cmd = new OleDbCommand();

con.ConnectionString = inscrieriTableAdapter.Connection.ConnectionString; cmd.Connection = con;

listaSet = "Set DataInscriere = #" + dateTimePicker1.Value.Month + "/" + dateTimePicker1.Value.Day + "/" + dateTimePicker1.Value.Year + "#," + "IdTip = " + cmbType.SelectedValue + "," + "IdClasa = " + cmbClass.SelectedValue + "," + "IdElev = " + cmbStudent.SelectedValue;

cmd.CommandText = "Update Inscrieri " + listaSet + " Where NrInscriere=" + txtNr.Text;

MessageBox.Show(cmd.CommandText);

con.Open();

try

{

cmd.ExecuteNonQuery();

}catch(Exception e)

{

if (e.HResult == -2147467259)

{

MessageBox.Show("Cheie semantica duplicat! Schimbati data sau selectati alt elev!");

return;

}

}

cmd.CommandText = "Update Registru Set IdClasa=" + cmbClass.SelectedValue + " Where IdElev=" + cmbStudent.SelectedValue;

MessageBox.Show(cmd.CommandText);

con.Open();

cmd.ExecuteNonQuery();

con.Close();

}

public void legareComboBoxuri(Boolean v)

{

clearDataBindings();

if (v)

{

cmbType.DataSource = tipuriInscrieriBindingSource;

cmbType.DisplayMember = "NumeTip";

cmbType.ValueMember = "IdTip";

cmbClass.DataSource = claseBindingSource;

cmbClass.DisplayMember = "Clasa";

cmbClass.ValueMember = "IdClasa";

cmbStudent.DataSource = eleviBindingSource;

cmbStudent.DisplayMember = "Nume";

cmbStudent.ValueMember = "IdElev";

txtCurrentClass.DataBindings.Add("Text", eleviBindingSource, "Clasa");

}

}

public void configurareComboBoxuri(Boolean v)

{

if (v)

{

cmbType.DropDownStyle = ComboBoxStyle.DropDownList;

cmbClass.DropDownStyle = ComboBoxStyle.DropDownList;

cmbStudent.DropDownStyle = ComboBoxStyle.DropDownList;

}

else

{

cmbType.DropDownStyle = ComboBoxStyle.Simple;

cmbClass.DropDownStyle = ComboBoxStyle.Simple;

cmbStudent.DropDownStyle = ComboBoxStyle.Simple;

}

}

public void clearDataBindings()

{

cmbStudent.DataBindings.Clear();

cmbType.DataBindings.Clear();

cmbClass.DataBindings.Clear();

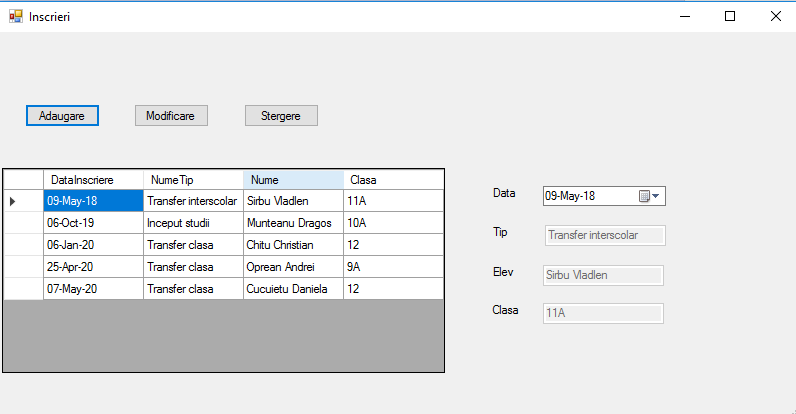
txtNr.DataBindings.Clear();

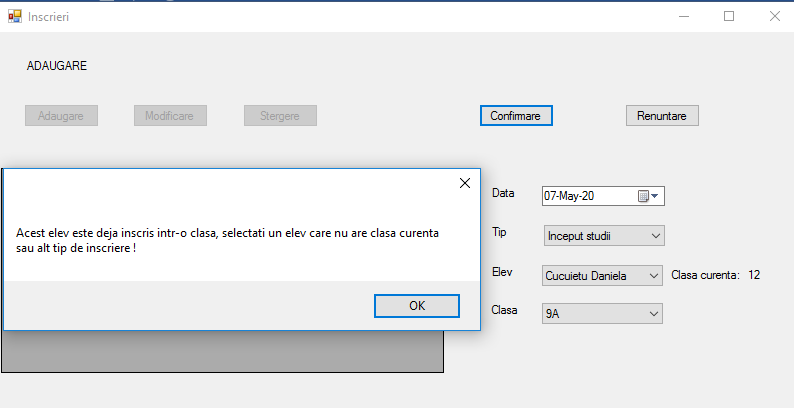
dateTimePicker1.DataBindings.Clear();

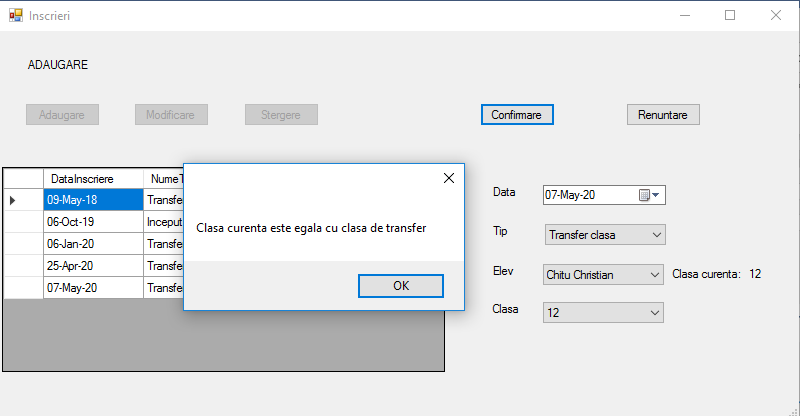
txtCurrentClass.DataBindings.Clear();

}

}







##### Exmatriculare

public partial class EvenimentExmatriculare : Form

{

int idElevVechi;

String idClasaVeche;

public EvenimentExmatriculare()

{

InitializeComponent();

}

private void EvenimentExmatriculare\_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'dataSet1.Elevi' table. You can move, or remove it, as needed.

this.eleviTableAdapter.Fill(this.dataSet1.Elevi);

A1();

}

private void btnAdaugare\_Click(object sender, EventArgs e)

{

A2();

}

private void btnModificare\_Click(object sender, EventArgs e)

{

idElevVechi = (int)cmbStudent.SelectedValue;

DataTable t;

t = dataSet1.Elevi;

foreach (DataRow y in t.Rows)

{

if ((int)y["IdElev"] == idElevVechi)

{

idClasaVeche = y["IdClasa"].ToString();

}

}

A5();

}

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;

A6();

}

private void btnConfirmare\_Click(object sender, EventArgs e)

{

A4();

}

private void btnRenuntare\_Click(object sender, EventArgs e)

{

A3();

}

private void A1()

{

//Umple cu date obiectele DataTable

exmatriculariTableAdapter.Fill(dataSet1.Exmatriculari);

motiveExmatriculariTableAdapter.Fill(dataSet1.MotiveExmatriculari);

eleviTableAdapter.Fill(this.dataSet1.Elevi);

//Configurare comboBox-uri

configurareComboBoxuri(false);

//Protectie grid

dataGridView1.AllowUserToAddRows = false;

dataGridView1.AllowUserToDeleteRows = false;

dataGridView1.ReadOnly = true;

//Protectie txtIdPersoana

txtClasa.ReadOnly = true;

txtNr.Visible = false;

A3();

}

private void A2()

{

//Configurare butoane

configureazaButoane(false);

//Dezlegare controale Panel

legareControale(false);

//Ridicare protectie controale Panel

protectiePanel(false);

//Configurare(ComboBox - Uri)

configurareComboBoxuri(true);

//Legare comboBox-uri la sursa de date

legareComboBoxuri(true);

//Modifcare lblOp

lblOp.Text = "ADAUGARE";

//Pozitionare cursor pe primul camp

dateTimePicker1.Focus();

// Golire campuri

golireCampuri();

}

private void A3()

{

//Configurare comboBox-uri

configurareComboBoxuri(false);

//Initializare lblOp

lblOp.Text = "";

//Çonfigurare(butoane)

configureazaButoane(true);

//Protectie componente Panel

protectiePanel(true);

//Legare controale

legareControale(true);

}

private void A4()

{

if (!validareCampuri())

return;

if (lblOp.Text == "ADAUGARE")

{

adauga\_inregistrare();

golireCampuri();

//Pune cursor pe primul camp

dateTimePicker1.Focus();

refresh\_grid(exmatriculariBindingSource.Position);

}

else if (lblOp.Text == "MODIFICARE")

{

modifica\_inregistrare();

refresh\_grid(exmatriculariBindingSource.Position);

A3();

}

else

{

MessageBox.Show("Operatie actualizare neefectuata");

}

}

private void A5()

{

//Configurare butoane

configureazaButoane(false);

//Dezlegare controale Panel

legareControale(true);

//Ridicare protectie controale Panel

protectiePanel(false);

//Configurare(ComboBox - Uri)

configurareComboBoxuri(true);

//Legare comboBox-uri la sursa de date

legareComboBoxuri(true);

//Modifcare lblOp

lblOp.Text = "MODIFICARE";

//Pozitionare cursor pe primul camp

dateTimePicker1.Focus();

txtClasa.Text = "";

}

private void A6()

{

OleDbConnection con = new OleDbConnection();

OleDbCommand cmd = new OleDbCommand();

con.ConnectionString = exmatriculariTableAdapter.Connection.ConnectionString;

cmd.Connection = con;

cmd.CommandText = "Delete From Exmatriculari Where NrExmatriculare = " + txtNr.Text;

MessageBox.Show(cmd.CommandText);

con.Open();

cmd.ExecuteNonQuery();

con.Close();

refresh\_grid(exmatriculariBindingSource.Position);

}

private void configureazaButoane(bool v)

{

btnRenuntare.Visible = !v;

btnConfirmare.Visible = !v;

btnAdaugare.Enabled = v;

btnModificare.Enabled = v;

btnStergere.Enabled = v;

}

private void legareControale(bool v)

{

clearDataBindings();

if (v)

{

cmbStudent.DataBindings.Add("Text", exmatriculariBindingSource, "Nume");

cmbMotiv.DataBindings.Add("Text", exmatriculariBindingSource, "NumeMotiv");

txtClasa.DataBindings.Add("Text", exmatriculariBindingSource, "Clasa");

txtNr.DataBindings.Add("Text", exmatriculariBindingSource, "NrExmatriculare");

dateTimePicker1.DataBindings.Add("Text", exmatriculariBindingSource,

"DataExmatriculare");

}

}

private void protectiePanel(bool v)

{

cmbStudent.Enabled = !v;

cmbMotiv.Enabled = !v;

}

private void golireCampuri()

{

cmbStudent.SelectedIndex = -1;

cmbMotiv.SelectedIndex = -1;

txtNr.Text = "";

txtClasa.Text = " ";

dateTimePicker1.Value = DateTime.Now;

}

private bool validareCampuri()

{

//Validare de completare obligatorie campurile

if (cmbStudent.Text == "")

{

MessageBox.Show("Selectati un elev !");

cmbStudent.Focus();

return false;

}

if (cmbMotiv.Text == "")

{

MessageBox.Show("Selectati motivul exmatricularii !");

cmbMotiv.Focus();

return false;

}

if (txtClasa.Text == "")

{

MessageBox.Show("Selectati un elev inregistrat intr-o clasa !");

txtClasa.Focus();

return false;

}

return true;

}

private void refresh\_grid(int p)

{

dataSet1.Exmatriculari.Clear();

exmatriculariTableAdapter.Fill(dataSet1.Exmatriculari);

exmatriculariBindingSource.Position = p;

}

private void adauga\_inregistrare()

{

string listaCampuri;

string listaValori;

OleDbConnection con = new OleDbConnection();

OleDbCommand cmd = new OleDbCommand();

con.ConnectionString = exmatriculariTableAdapter.Connection.ConnectionString;

cmd.Connection = con;

listaCampuri = "DataExmatriculare, IdMotiv, IdElev, IdClasa";

DataTable t;

int x;

x = (int)cmbStudent.SelectedValue;

t = dataSet1.Elevi;

int IdClasa=0;

foreach (DataRow y in t.Rows)

{

if ((int)y["IdElev"] == x)

{

IdClasa = (int)y["IdClasa"];

}

}

listaValori = "#" + dateTimePicker1.Value.Month + "/" + dateTimePicker1.Value.Day + "/" + dateTimePicker1.Value.Year + "#," + cmbMotiv.SelectedValue + "," + cmbStudent.SelectedValue+","+IdClasa;

cmd.CommandText = "Insert into Exmatriculari(" + listaCampuri + ") Select " + listaValori;

MessageBox.Show(cmd.CommandText);

con.Open();

try

{

cmd.ExecuteNonQuery();

}

catch (Exception e)

{

if (e.HResult == -2147467259)

{

MessageBox.Show("Cheie semantica duplicat! Schimbati data sau selectati alt elev!");

return;

}

}

cmd.CommandText = "Update Registru Set IdClasa=NULL Where IdElev=" + cmbStudent.SelectedValue;

MessageBox.Show(cmd.CommandText);

cmd.ExecuteNonQuery();

con.Close();

}

private void modifica\_inregistrare()

{

string listaSet;

OleDbConnection con = new OleDbConnection();

OleDbCommand cmd = new OleDbCommand();

con.ConnectionString = exmatriculariTableAdapter.Connection.ConnectionString;

cmd.Connection = con;

DataTable t;

int x;

x = (int)cmbStudent.SelectedValue;

t = dataSet1.Elevi;

int IdClasa=0;

foreach (DataRow y in t.Rows)

{

if ((int)y["IdElev"] == x)

{

IdClasa = (int)y[2];

}

}

listaSet = "Set DataExmatriculare = #" + dateTimePicker1.Value.Month + "/" + dateTimePicker1.Value.Day + "/" + dateTimePicker1.Value.Year + "#," + "IdMotiv = " + cmbMotiv.SelectedValue + "," + "IdClasa = " + IdClasa + "," + "IdElev = " + cmbStudent.SelectedValue;

cmd.CommandText = "Update Exmatriculari " + listaSet + " Where NrExmatriculare=" + txtNr.Text;

MessageBox.Show(cmd.CommandText);

con.Open();

try

{

cmd.ExecuteNonQuery();

}

catch (Exception e)

{

if (e.HResult == -2147467259)

{

MessageBox.Show("Cheie semantica duplicat! Schimbati data sau selectati alt elev!");

return;

}

}

cmd.CommandText = "Update Registru Set IdClasa=" + idClasaVeche + " Where IdElev=" + idElevVechi;

MessageBox.Show(cmd.CommandText);

cmd.ExecuteNonQuery();

cmd.CommandText = "Update Registru Set IdClasa=null Where IdElev=" + cmbStudent.SelectedValue;

MessageBox.Show(cmd.CommandText);

cmd.ExecuteNonQuery();

con.Close();

}

public void legareComboBoxuri(Boolean v)

{

clearDataBindings();

if (v)

{

cmbMotiv.DataSource = motiveExmatriculariBindingSource;

cmbMotiv.DisplayMember = "NumeMotiv";

cmbMotiv.ValueMember = "IdMotiv";

cmbStudent.DataSource = eleviBindingSource;

cmbStudent.DisplayMember = "Nume";

cmbStudent.ValueMember = "IdElev";

}

}

public void configurareComboBoxuri(Boolean v)

{

if (v)

{

cmbMotiv.DropDownStyle = ComboBoxStyle.DropDownList;

cmbStudent.DropDownStyle = ComboBoxStyle.DropDownList;

}

else

{

cmbMotiv.DropDownStyle = ComboBoxStyle.Simple;

cmbStudent.DropDownStyle = ComboBoxStyle.Simple;

}

}

public void clearDataBindings()

{

cmbStudent.DataBindings.Clear();

cmbMotiv.DataBindings.Clear();

txtClasa.DataBindings.Clear();

txtNr.DataBindings.Clear();

dateTimePicker1.DataBindings.Clear();

}

private void cmbStudent\_SelectedIndexChanged(object sender, EventArgs e)

{

DataTable t;

int x;

if (lblOp.Text == "")

return;

if (cmbStudent.Text == "")

return;

x = (int)cmbStudent.SelectedValue;

t = dataSet1.Elevi;

foreach(DataRow y in t.Rows){

if ((int)y["IdElev"] == x)

{

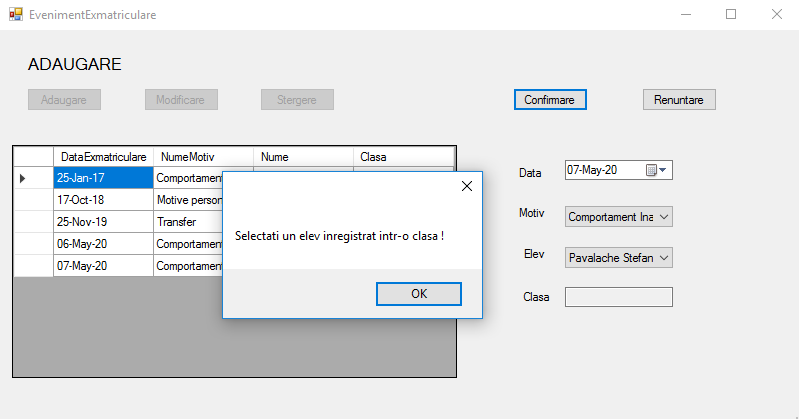
txtClasa.Text = y[3].ToString();

}

}

}

}



#### Agenti

##### Tipuri inscriere

public AgentTipuriInscrieri()

{

InitializeComponent();

}

private void Agent\_tipuri\_inscrieri\_Load(object sender, EventArgs e)

{

//A1

config(true);

refresh();

}

private void config(bool v)

{

dataGridView1.AllowUserToAddRows = !v;

dataGridView1.AllowUserToDeleteRows = !v;

dataGridView1.ReadOnly = v;

btnActualizare.Enabled = v;

btnSalvare.Visible = !v;

btnRenuntare.Visible = !v;

}

private void refresh()

{

this.tipuriInscrieriTableAdapter.Fill(this.dataSet1.TipuriInscrieri);

}

private void btnActualizare\_Click(object sender, EventArgs e)

{

//A2

config(false);

}

private void btnSalvare\_Click(object sender, EventArgs e)

{

//A4

try

{

this.tipuriInscrieriTableAdapter.Update(this.dataSet1.TipuriInscrieri);

config(true);

refresh();

}

catch (Exception exc)

{

string s = exc.Message;

if (s.IndexOf("duplicate values") > 0)

MessageBox.Show("Inregistrare deja existenta !");

else if (s.IndexOf("cannot be deleted") > 0)

MessageBox.Show("Ati sters inregistrari referite in alte tabele !");

}

}

private void btnRenuntare\_Click(object sender, EventArgs e)

{

//A3

config(true);

refresh();

}

private void dataGridView1\_DataError(object sender, DataGridViewDataErrorEventArgs e)

{

MessageBox.Show("Format eronat");

}

private void dataGridView1\_UserDeletingRow(object sender, DataGridViewRowCancelEventArgs e)

{

const string mesaj = "Confirmati stergerea";

const string titlu = "Stergere inregistrare";

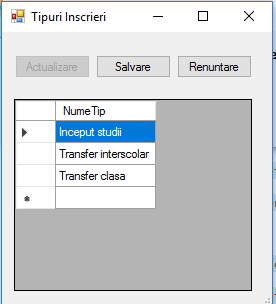
var rezultat = MessageBox.Show(mesaj, titlu, MessageBoxButtons.YesNo,

MessageBoxIcon.Warning);

if (rezultat == DialogResult.No) e.Cancel = true;

}

}



##### Motive exmatriulare

public partial class AgentMotiveExmatriculare : Form

{

public AgentMotiveExmatriculare()

{

InitializeComponent();

}

private void Agent\_Motive\_Exmatriculare\_Load(object sender, EventArgs e)

{

//A1

config(true);

refresh();

}

private void config(bool v)

{

dataGridView1.AllowUserToAddRows = !v;

dataGridView1.AllowUserToDeleteRows = !v;

dataGridView1.ReadOnly = v;

btnActualizare.Enabled = v;

btnSalvare.Visible = !v;

btnRenuntare.Visible = !v;

}

private void refresh()

{

this.motiveExmatriculariTableAdapter.Fill(this.dataSet1.MotiveExmatriculari);

}

private void btnActualizare\_Click(object sender, EventArgs e)

{

//A2

config(false);

}

private void btnSalvare\_Click(object sender, EventArgs e)

{

//A4

try

{

this.motiveExmatriculariTableAdapter.Update(this.dataSet1.MotiveExmatriculari);

config(true);

refresh();

}

catch (Exception exc)

{

string s = exc.Message;

if (s.IndexOf("duplicate values") > 0)

MessageBox.Show("Inregistrare deja existenta !");

else if (s.IndexOf("cannot be deleted") > 0)

MessageBox.Show("Ati sters inregistrari referite in alte tabele !");

}

}

private void btnRenuntare\_Click(object sender, EventArgs e)

{

//A3

config(true);

refresh();

}

private void dataGridView1\_DataError(object sender, DataGridViewDataErrorEventArgs e)

{

MessageBox.Show("Format eronat");

}

private void dataGridView1\_UserDeletingRow(object sender, DataGridViewRowCancelEventArgs e)

{

const string mesaj = "Confirmati stergerea";

const string titlu = "Stergere inregistrare";

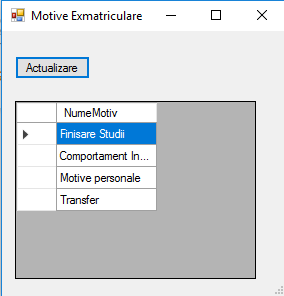
var rezultat = MessageBox.Show(mesaj, titlu, MessageBoxButtons.YesNo,

MessageBoxIcon.Warning);

if (rezultat == DialogResult.No) e.Cancel = true;

}

}



#### DataSet

