

*Meniu Intelligent*

Colegiul Național „Alexandru Lahovari”

Râmnicu Vâlcea

# **Meniu Intelligent**

Lucrare de atestat Informatică

Profesor coordonator:

Cataragă Issabella

Absolvent:

Vîlculescu Mihai-Bogdan

**Mai 2017**

## CUPRINS

<b>1. Despre aplicație</b>	3
<b>2. Documentație de utilizare</b>	4
<b>3. Documentație de proiectare (Client)</b>	5
3.1 Meniu initial	6
3.1.1 Particularități	7
3.1.2 Forma rezultată	8
3.2 Creare_cont_client	9
3.2.1 Particularități	10
3.2.2 Forma rezultată	11
3.3 Login	12
3.3.1 Particularități	13
3.3.2 Forma rezultată	14
3.4 Opțiuni	15
3.4.1 Calculator Kcal	16
3.4.1.1 Particularități	17
3.4.1.2 Forma rezultată	18
3.4.2 Comanda	19
3.4.2.1 Particularități	20
3.4.2.2 Forma rezultată	21
3.4.3 Grafic Kcal	22
3.4.3.1 Particularități	23
3.4.3.2 Forma rezultată	24
<b>4. Documentație de proiectare ( Administrator)</b>	25
4.1 Admin	26
4.1.1 Particularități	27
4.1.2 Forma rezultată	28
<b>5. Tabele</b>	29
<b>6. Bibliografie</b>	30

*Meniu Intelligent***1. Despre Aplicație**

În ultimii ani, lumea a început să devină din ce în ce mai interesată de nutriție, în special mâncatul sănătos. Din păcate, din ce în ce mai multe persoane în zilele noastre suferă de obezitate, în mare parte din pricina unei alimentații inadecvate. Astfel, cuvântul „calorie” are mai mult sens pentru majoritatea persoanelor care încearcă să ducă un stil de viață cât mai sănătos.

Am creat această aplicație pentru a propune utilizatorilor un meniu structurat în funcție de necesitățile calorice ale fiecăruia.

*Configurația sistemului care ar permite o rulare optimă este:*

( System requirements )

- Sistem de operare: *Windows Vista/7/8/8.1/10*
- Procesor: *1 GHz*
- Hard disk: *50MB* free ( se menționează că pentru rularea optimă a sistemului de operare vor fi necesari 200 MB liberi pe partiția acestuia).
- Memorie RAM: *1 GB*
- Domeniu: *Microsoft Visual Studio 2010 Express, c#*

*Configurația sistemului care garantează rularea programului este:*

( Minimum System requirements )

- Sistem de operare: *Windows Vista*
- Procesor: *512 MHz*
- Hard disk: *25MB* free ( se menționează că pentru rularea optimă a sistemului de operare vor fi necesari 100 MB liberi pe partiția acestuia).
- Memorie RAM: *512 GB*
- Domeniu: *Microsoft Visual Studio 2010 Express, c#*

## **2. Documentație de utilizare**

### **Administratorul**

Contul administratorului nu poate fi modificat prin intermediul aplicației, pentru a spori securitatea acesteia. Astfel, contul este creat chiar de către administrator modificând bazele de date, fiind singurul care are acces la acestea.

### **Autentificarea**

La rularea aplicației, utilizatorul trebuie să introducă emailul și parola contului, verificându-se în baza de date validitatea acestora. În același timp, se va verifica și statusul utilizatorului: administrator sau client.

### **Forma admin**

Forma Admin este creată pentru a realiza modificarea cu ușurință a meniului restaurantului, și anume:

- Adăugarea unui nou produs: se introduc datele ce urmează să fie adăugate în căsuțele text corespunzătoare.
- Ștergerea unui produs: se selectează produsul dorit din listă și se acționează butonul de ștergere.

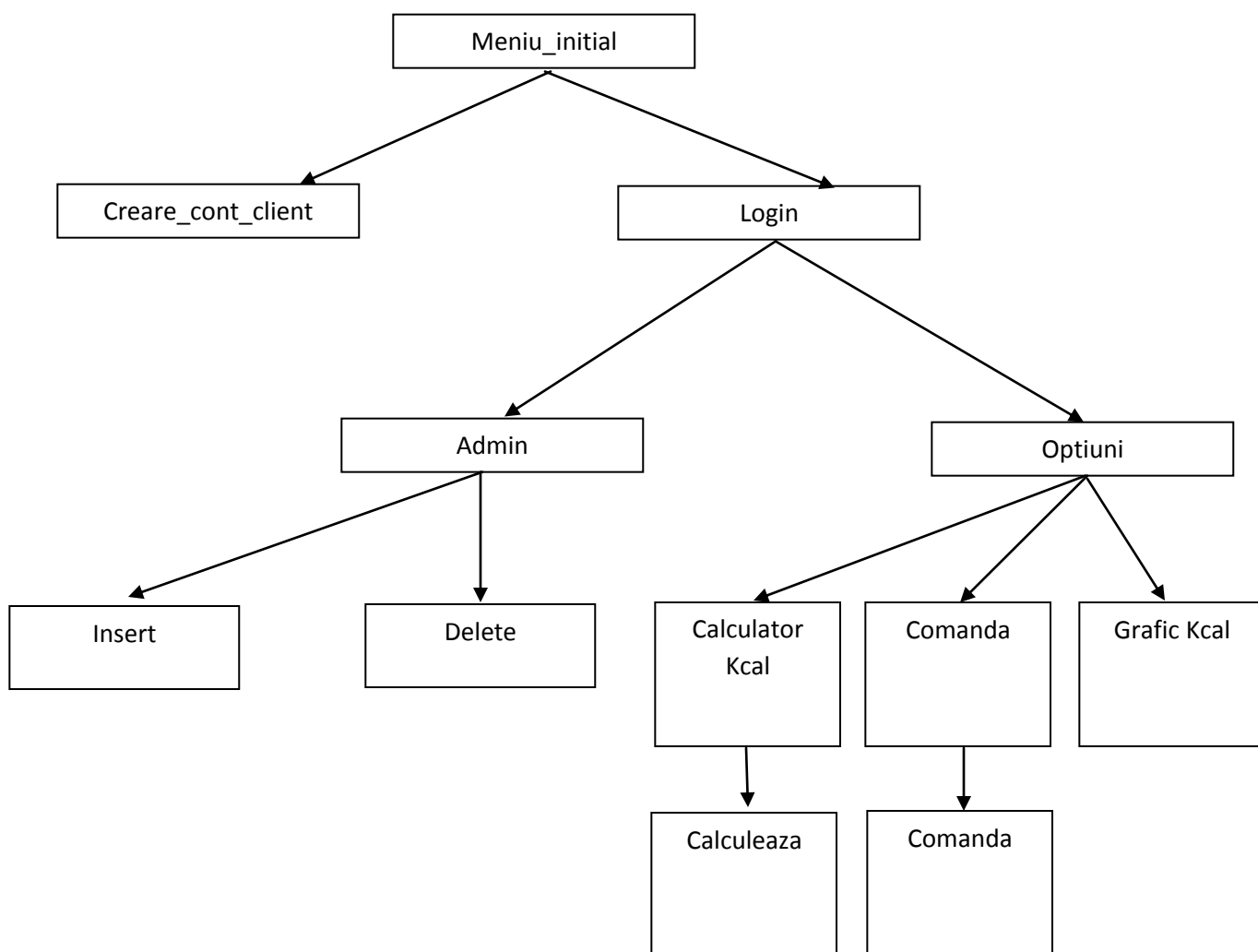
### **Forma Opțiuni**

Forma Opțiuni permite clientului să realizeze 3 operațiuni:

- Modificarea numărului de kcal necesar prin introducerea greutateii, înălțimii și a vârstei.
- Selectarea produselor dorite și finalizarea comenzii.
- Afișarea unui grafic realizat în funcție de comenzile utilizatorului.

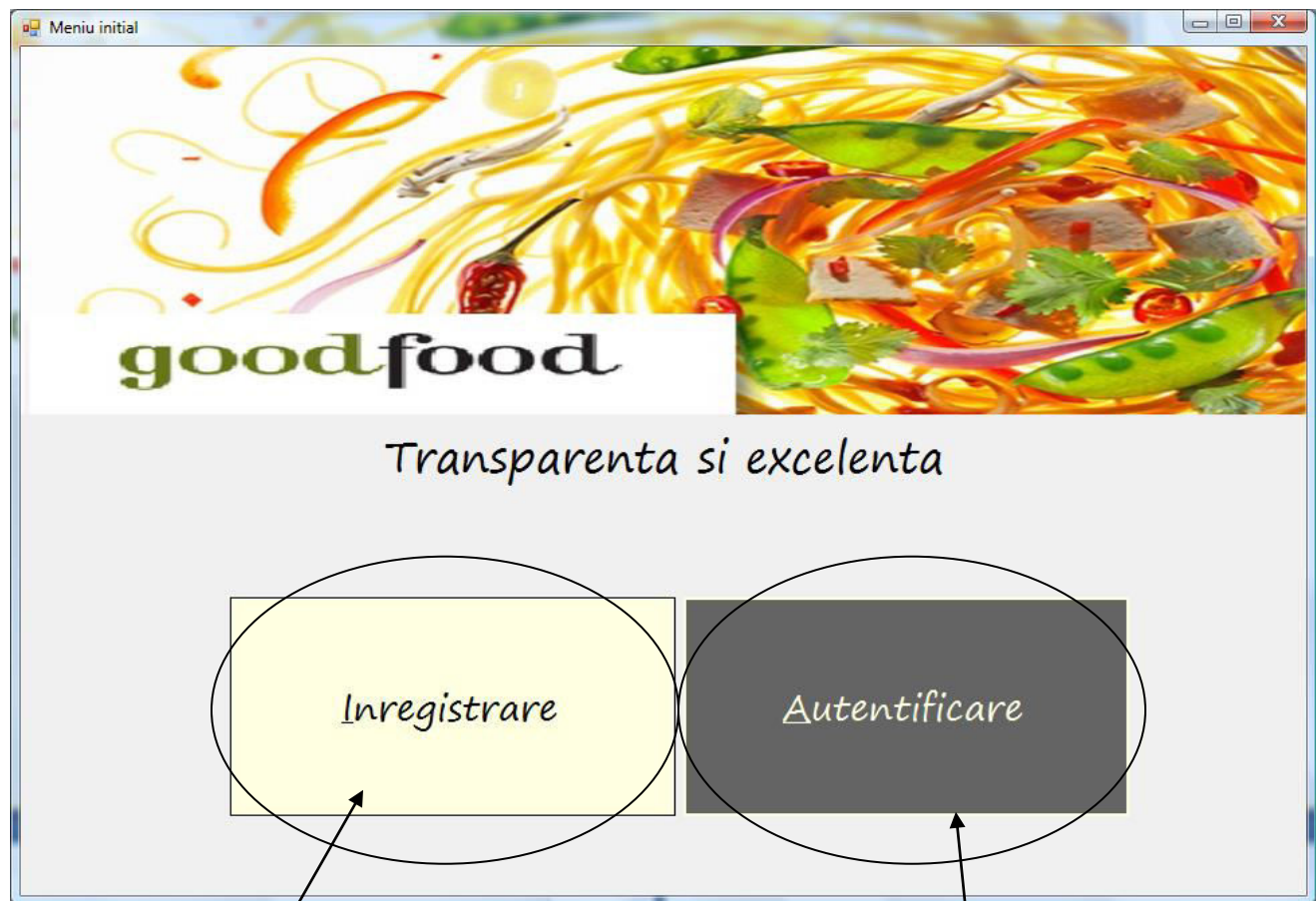
### 3. Documentatie de proiectare (Client)

Următoarea schemă prezintă relațiile între formele aplicației:



*Meniu Intelligent***3.1 Meniu initial**

Forma Meniu\_initial reprezintă prima formă a programului care apare de fiecare dată când programul este lansat în execuție. A fost concepută interactiv folosind new form și adăugând elementele după cum urmează:



Butonul a fost adăugat din Form Controls. La apăsarea acestuia, va fi deschisă forma Creare\_cont\_client.

Butonul a fost adăugat din Form Controls. La apăsarea acestuia, va fi deschisă forma Login.

*Meniu Inteligent***3.1.1 Particularități**

- Imaginea de fundal a fost setată cu ajutorul unui control de tip PictureBox;
- Pentru PictureBox a fost setată proprietatea: Dock = Top;
- La etichete am setat background transparent folosind proprietatea: BackColor = Transparent;
- Pentru butonul ”Inregistrare” am setat culoarea de fundal: BackColor = Info;
- Pentru butonul ”Autentificare” am setat culoarea de fundal: BackColor = WindowFrame;
- Fontul folosit pentru ambele butoane este: Segoe Print; 20,25pt;
- Pentru butonul ”Inregistrare” am setat culoarea textului: ForeColor = ControlText;
- Pentru butonul ”Autentificare” am setat culoarea textului: ForeColor = Info;
- La proprietățile formei am setat proprietatea: StartPosition = CenterScreen. Astfel, forma este poziționată în centrul ecranului la fiecare rulare;
- Mărimea formei a fost setată la: 974 (lățime) și 670 (înălțime);
- La apăsarea butonului ”Inregistrare” este executată următoarea secvență de cod:

```
this.Hide();  
Create_cont_client login = new Create_cont_client();  
login.Closed += (s, args) => this.Show();  
  
login.Show();
```

- La apăsarea butonului ”Autentificare” este executată următoarea secvență de cod:

```
this.Hide();  
Login login = new Login();  
login.Closed += (s, args) => this.Show();  
  
login.Show();
```

*Meniu Inteligent*

### 3.1.2 Forma rezultată

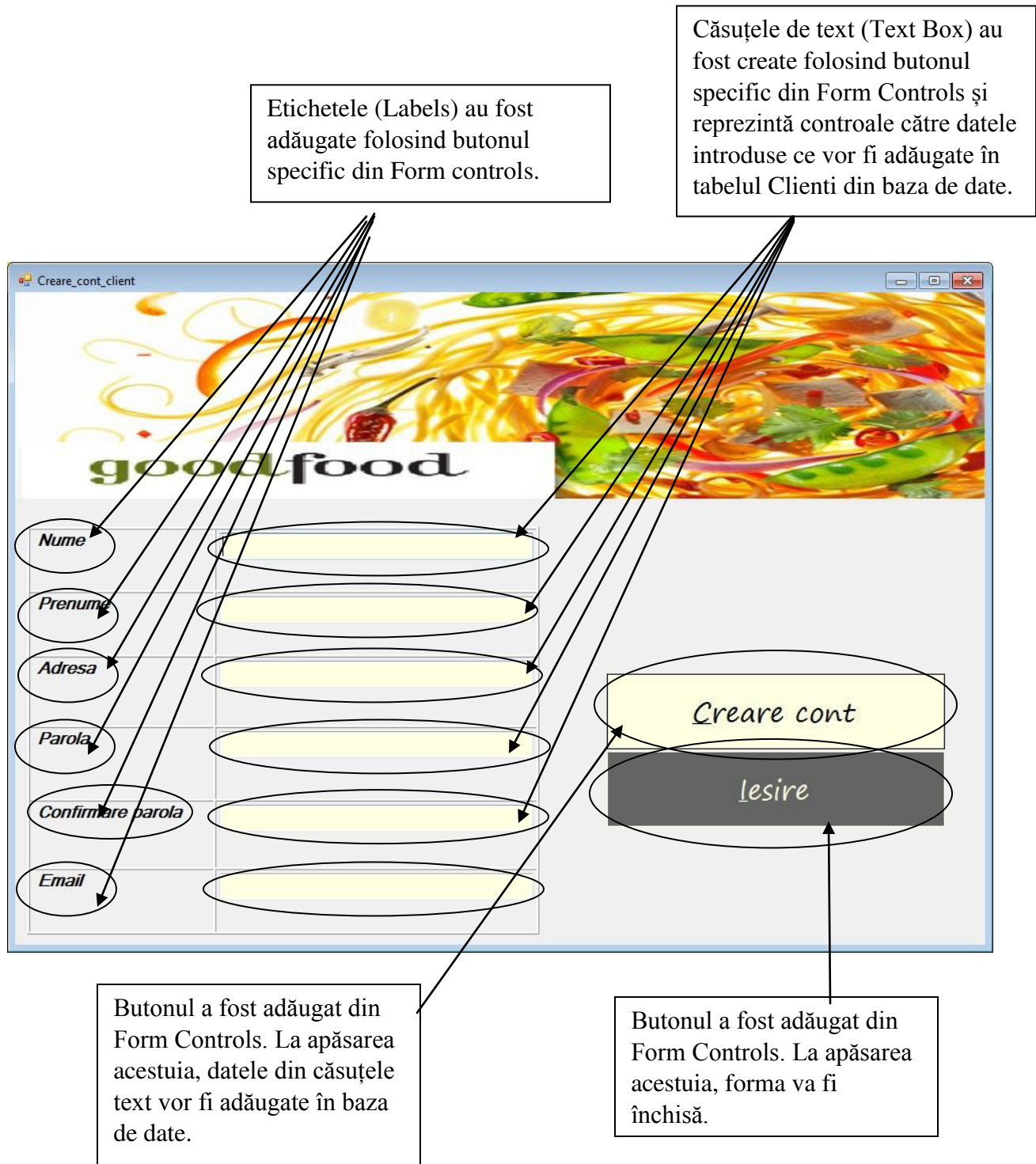


### 3.2 Creare\_cont\_client

Forma Creare\_cont\_client apare de fiecare dată prin selectarea opțiunii "Inregistrare" din meniul inițial. A fost concepută interactiv folosind new form și adăugând elementele după cum urmează:



*Meniu Inteligent*



*Meniu Intelligent***3.2.1 Particularități**

- Imaginea de fundal a fost setată cu ajutorul unui control de tip PictureBox;
- Pentru PictureBox a fost setată proprietatea: Dock = Top;
- La etichete am setat background transparent folosind proprietatea: BackColor = Transparent;
- Pentru butonul "Creare cont" am setat culoarea de fundal: BackColor = Info;
- Pentru butonul "Iesire" am setat culoarea de fundal: BackColor = WindowFrame;
- Fontul folosit pentru ambele butoane este: Segoe Print; 20,25pt;
- Pentru o mai bună organizare a controalelor de tip TextBox și Label, am utilizat un control de tip TableLayoutPanel cu 2 coloane și 6 rânduri;
- Pentru butonul "Iesire" am setat culoarea textului: ForeColor = Info;
- Pentru butonul "Creare cont" am setat culoarea textului: ForeColor = ControlText;
- La proprietățile formei am setat proprietatea: StartPosition = CenterScreen. Astfel, forma este poziționată în centrul ecranului la fiecare rulare;
- Mărimea formei a fost setată la: 509 (lățime) și 405 (înălțime);
- Pentru controlul de tip TableLayoutPanel, am setat marginile cu proprietatea: CellBorderStyle = OutsetDouble;
- La apăsarea butonului "Creare cont" este executată următoarea secvență de cod:

```
string name = nameTxt.Text;
string surname = surnameTxt.Text;
string address = addressTxt.Text;
string password = pwdTxt.Text;
string email = mailTxt.Text;

if (functions.validMail(email))
{
    if (!functions.duplicateMail(email))
    {
        if (pwdTxt.Text == confPwdTxt.Text)
        {
            string connString = @"Data Source=.\SQLEXPRESS;AttachDbFilename=" +
                functions.getDBPath("GOOD_FOOD.mdf") + ";Integrated Security=True;User
                Instance=True";
            using (SqlConnection con = new SqlConnection(connString))
            {
                try
                {
                    con.Open();
                    SqlCommand cmd = new SqlCommand(@"INSERT INTO Clienti (parola, nume, prenume,
                    adresa, email, kcal_zilnice) VALUES (@pass, @name, @surname, @address, @mail, 0)",
                    con);

                    cmd.Parameters.AddWithValue("@pass", pwdTxt.Text);
                    cmd.Parameters.AddWithValue("@name", nameTxt.Text);
                    cmd.Parameters.AddWithValue("@surname", surnameTxt.Text);
```

*Meniu Intelligent*

```
cmd.Parameters.AddWithValue("@address", addressTxt.Text);
cmd.Parameters.AddWithValue("@mail", mailTxt.Text);

cmd.ExecuteNonQuery();
}

catch (Exception ex)
{
    //MessageBox.Show(this, "A avut loc o eroare!", "Eroare",
    MessageBoxButtons.AbortRetryIgnore, MessageBoxIcon.Error);
    functions.mesajEroare(this, "A avut loc o eroare!", "Eroare");
}
}

this.Close();
}

else
{
    MessageBox.Show("Parolele nu se potrivesc!");
    pwdTxt.Text = "";
    confPwdTxt.Text = "";
}
}

else
{
    MessageBox.Show("Emailul exista deja in baza de date!");
    mailTxt.Text = "";
}
}

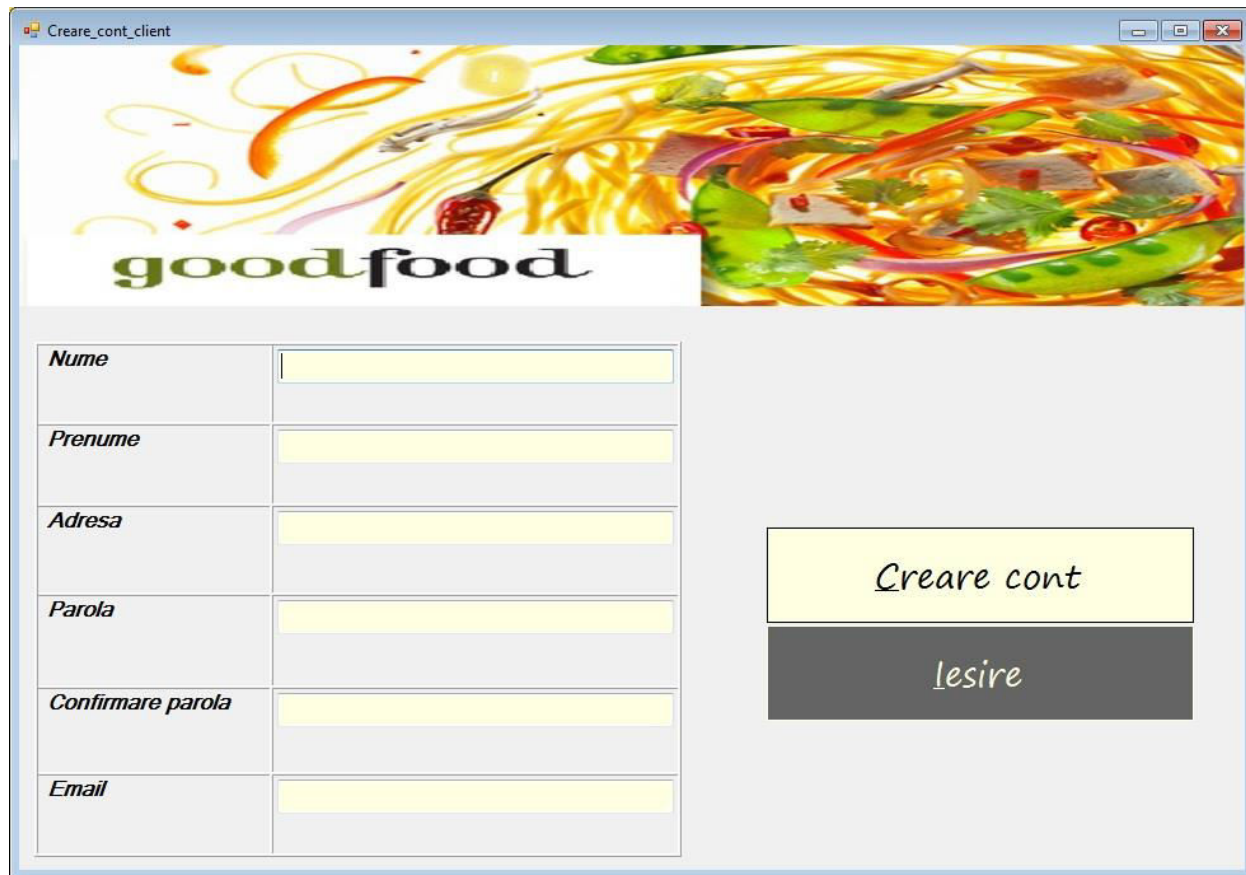
else
{
    MessageBox.Show("Emailul nu este valid!");
    mailTxt.Text = "";
}
}
```

- La apăsarea butonului "Iesire" este executată următoarea secvență de cod:

```
this.Close();
```

## *Meniu Intelligent*

### 3.2.2 Forma rezultată



The screenshot shows a web application window titled "Creare\_cont\_client". The header area contains a vibrant image of spaghetti with various toppings and the text "good food". Below the header, there is a registration form with the following fields:

<i>Nume</i>	<input type="text"/>
<i>Prenume</i>	<input type="text"/>
<i>Adresa</i>	<input type="text"/>
<i>Parola</i>	<input type="password"/>
<i>Confirmare parola</i>	<input type="password"/>
<i>Email</i>	<input type="text"/>

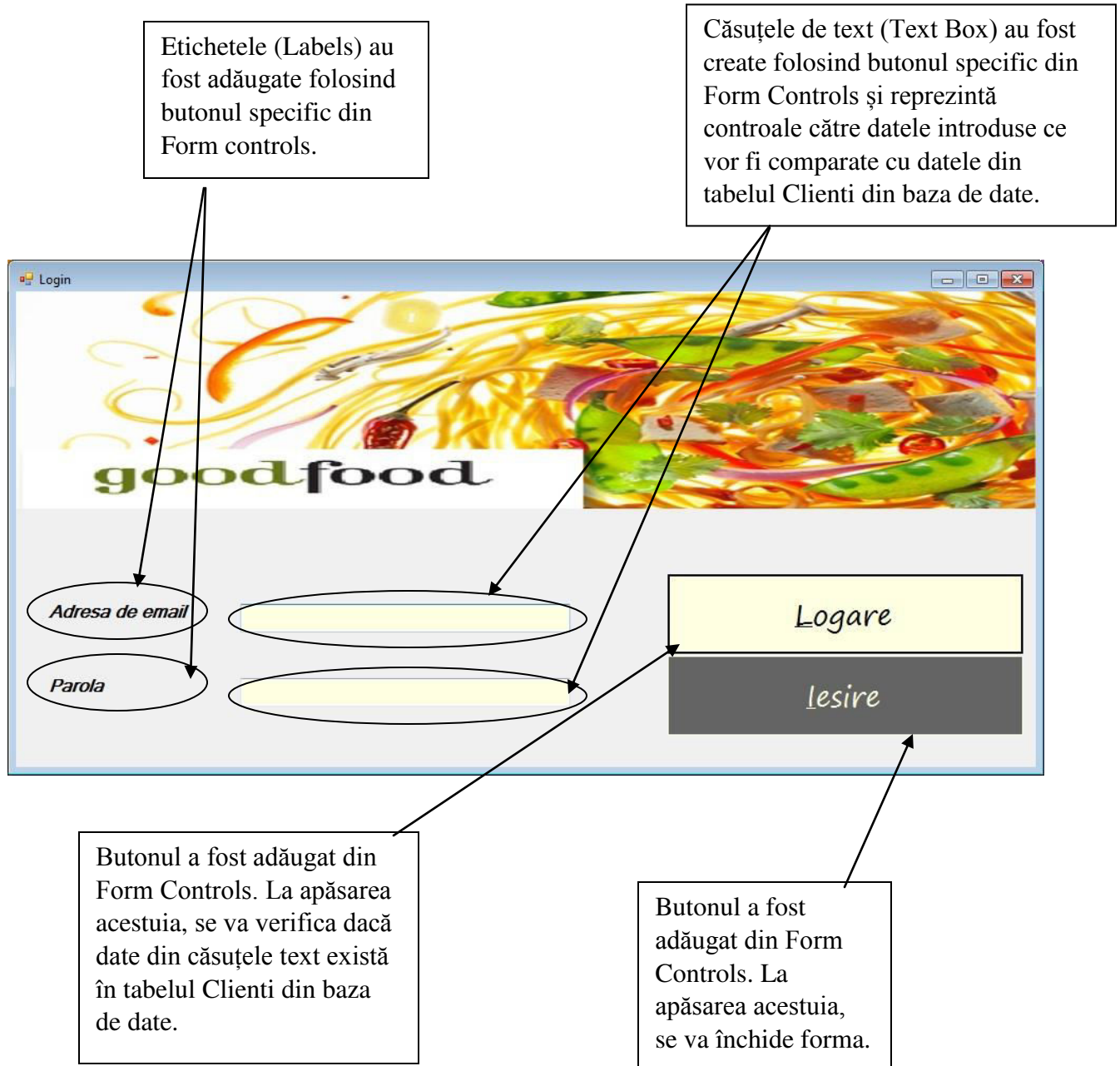
To the right of the form, there are two buttons:

- A yellow button labeled "Creare cont".
- A dark grey button labeled "Intrare".

### 3.3 Login

Forma Login apare de fiecare dată prin selectarea opțiunii "Autentificare" din meniul inițial. A fost concepută interactiv folosind new form și adăugând elementele după cum urmează:

*Meniu Intelligent*



*Meniu Intelligent***3.3.1 Particularități**

- Imaginea de fundal a fost setată cu ajutorul unui control de tip PictureBox;
- Pentru PictureBox a fost setată proprietatea: Dock = Top;
- La etichete am setat background transparent folosind proprietatea: BackColor = Transparent;
- Pentru butonul "Logare" am setat culoarea de fundal: BackColor = Info;
- Pentru butonul "Iesire" am setat culoarea de fundal: BackColor = WindowFrame;
- Fontul folosit pentru ambele butoane este: Segoe Print; 20,25pt;
- Pentru butonul "Iesire" am setat culoarea textului: ForeColor = Info;
- Pentru butonul "Logare" am setat culoarea textului: ForeColor = ControlText;
- La proprietățile formei am setat proprietatea: StartPosition = CenterScreen. Astfel, forma este poziționată în centrul ecranului la fiecare rulare;
- Mărimea formei a fost setată la: 980 (lățime) și 487 (înălțime);
- La apăsarea butonului "Ieșire" este executată următoare secvență de cod:

```
this.Close();
```

- La apăsarea butonului "Logare" este executată următoarea secvență de cod:

```
Functions functions = new Functions();
string connString = @"Data Source=.\SQLEXPRESS;AttachDbFilename=" +
functions.getDBPath("GOOD_FOOD.mdf") + @";Integrated Security=True;User
Instance=True";

using (SqlConnection con = new SqlConnection(connString))
{
    try
    {
        con.Open();
        SqlCommand cmd = new SqlCommand(@"SELECT * FROM Clienti WHERE parola = @pass AND
email = @mail", con);

        cmd.Parameters.AddWithValue("@pass", pwdTxt.Text);
        cmd.Parameters.AddWithValue("@mail", mailTxt.Text);

        SqlDataReader reader = cmd.ExecuteReader();

        if (reader.Read())
        {
            if (reader["admin"].ToString().Trim().Equals("NU"))
            {
                id_client = int.Parse(reader["id_client"].ToString());
                daily_kcal = reader["kcal_zilnice"].ToString();

                this.Hide();
```

*Meniu Intelligent*

```
Optiuni login = new Optiuni();  
login.Closed += (s, args) => this.Show();  
  
login.Show();  
}  
  
else  
{  
    this.Hide();  
    Admin login = new Admin();  
    login.Closed += (s, args) => this.Show();  
  
    login.Show();  
}  
  
mailTxt.Text = "";  
pwdTxt.Text = "";  
}  
  
else  
{  
    functions.mesajEroare(this, "Datele introduse sunt gresite!", "Date gresite");  
}  
}  
  
catch (Exception ex)  
{  
    functions.mesajEroare(this, "A avut loc o eroare!", "Eroare");  
}  
}
```

**3.3.2 Forma rezultată**

### 3.4 Optiuni

Forma Optiuni apare de fiecare dată după ce utilizatorul intra în cont. Această formă conține un control de tip TabControl, fiind alcătuită din trei tab-uri:

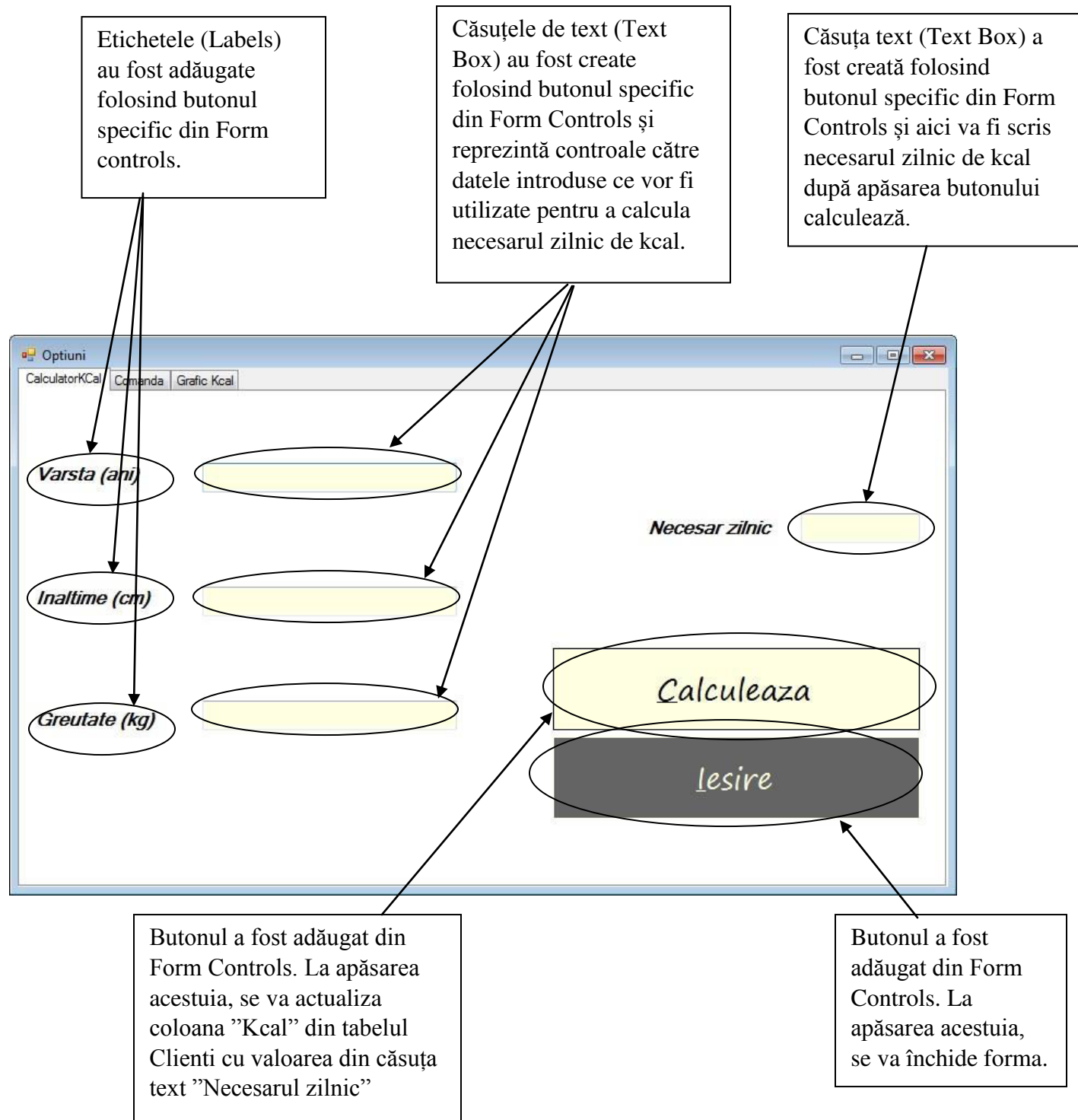
- CalculatorKcal
- Comanda
- GraficKcal

The screenshot shows a Windows-style application window titled "Optiuni". It features a TabControl with three tabs: "CalculatorKcal", "Comanda", and "Grafic Kcal". The "CalculatorKcal" tab is currently selected. The interface includes three input fields on the left for "Varsta (ani)", "Inaltime (cm)", and "Greutate (kg)". On the right, there is an input field for "Necesar zilnic". At the bottom right, there are two buttons: a yellow button labeled "Calculeaza" and a dark gray button labeled "Iesire".



### 3.4.1 CalculatorKcal

Tabul CalculatorKcal este primul dintre cele trei taburi ale formei Optiuni. A fost concepută interactiv folosind new form și adăugând elementele după cum urmează:



*Meniu Intelligent***3.4.1.1 Particularități**

- La etichete am setat background transparent folosind proprietatea: BackColor = Transparent;
- Pentru butonul "Calculeaza" am setat culoarea de fundal: BackColor = Info;
- Pentru butonul "Iesire" am setat culoarea de fundal: BackColor = WindowFrame;
- Fontul folosit pentru ambele butoane este: Segoe Print; 20,25pt;
- Pentru butonul "Iesire" am setat culoarea textului: ForeColor = Info;
- Pentru butonul "Calculeaza" am setat culoarea textului: ForeColor = ControlText;
- La proprietățile formei am setat proprietatea: StartPosition = CenterScreen. Astfel, forma este poziționată în centrul ecranului la fiecare rulare;
- Mărimea formei a fost setată la: 868 (lățime) și 512 (înălțime);
- La apăsarea butonului "Calculeaza" este executată următoarea secvență de cod:

```
if (!varstaTxt.Text.Equals(""))
{
    if (!heightTxt.Text.Equals(""))
    {
        if (!weightTxt.Text.Equals(""))
        {
            int age = int.Parse(varstaTxt.Text);
            int height = int.Parse(heightTxt.Text);
            int weight = int.Parse(weightTxt.Text);

            suma = age + height + weight;

            if (suma < 250)
            {
                Login.daily_kcal = "1800";
            }

            else if (suma >= 250 && suma <= 275)
            {
                Login.daily_kcal = "2200";
            }

            else
            {
                Login.daily_kcal = "2500";
            }

            string connString = @"Data Source=.\SQLEXPRESS;AttachDbFilename=" +
                functions.getDBPath("GOOD_FOOD.mdf") + ";Integrated Security=True;User
                Instance=True";
            using (SqlConnection con = new SqlConnection(connString))
            {
                try
                {
                    con.Open();
                    SqlCommand cmd = new SqlCommand(@"UPDATE Clienti SET kcal_zilnice = @kcal WHERE
                    id_client = @id", con);
```

*Meniu Inteligent*

```
cmd.Parameters.AddWithValue("@kcal", Login.daily_kcal);
cmd.Parameters.AddWithValue("@id", Login.id_client);

cmd.ExecuteNonQuery();

MessageBox.Show("Necesarul zilnic calculat cu succes!");
}

catch (Exception ex)
{
    //MessageBox.Show(this, "A avut loc o eroare!", "Eroare",
    MessageBoxButtons.AbortRetryIgnore, MessageBoxIcon.Error);
    functions.mesajEroare(this, "A avut loc o eroare!", "Eroare");
}

dailyTxt.Text = Login.daily_kcal.ToString();
varstaTxt.Text = "";
heightTxt.Text = "";
weightTxt.Text = "";
}

else
{
    functions.mesajEroare(this, "Completati toate campurile!", "Date incomplete");
}

else
{
    functions.mesajEroare(this, "Completati toate campurile!", "Date incomplete");
}

else
{
    functions.mesajEroare(this, "Completati toate campurile!", "Date incomplete");
}
```

- La apăsarea butonului ”Ieșire” este executată următoare secvență de cod:

```
this.Close();
```

*Meniu Intelligent***3.4.1.2 Forma rezultată**

The screenshot shows a Windows-style application window titled 'Optiuni'. It has three tabs: 'CalculatorKCal', 'Comanda' (which is selected), and 'Grafic Kcal'. The 'Comanda' tab contains a form with five input fields and two buttons. On the left, there are three vertically stacked labels: 'Varsta (ani)', 'Inaltime (cm)', and 'Greutate (kg)', each followed by a yellow rectangular input field. On the right, there is a label 'Necesar zilnic' followed by a yellow rectangular input field. At the bottom right, there are two stacked buttons: a yellow one labeled 'Calculeaza' and a dark gray one labeled 'Iesire'.

**3.4.2 Comanda**

Tabul Comanda este al doilea dintre cele trei taburi ale formei Optiuni. A fost concepută interactiv folosind new form și adăugând elementele după cum urmează:

*Meniu Inteligent*

Controlul de tip DataGridView a fost creat folosind butonul specific din Form Controls și aici sunt afișate datele din tabelul Meniu din baza de date.

Butoanele au fost adăugate din proprietățile controlului dataGridView și prin apăsarea unuia dintre acestea se adaugă produsul corespunzător liniei în coș

The screenshot shows a Windows application window titled "Optiuni" with three tabs: "CalculatorKCal", "Comanda", and "Grafic KCal". The "Comanda" tab is active, displaying a DataGridView table with the following data:

	id_produs	denumire_produs	descriere	pret	kcal	felul	Cantitate	Adauga
1		Ciorba de burta ...	Burta de vita, sm...	9	400	1	1	Adauga
2		Ciorba radautean...	Piept de pui, sm...	12	450	1	1	Adauga
3		Ciorba de vacuta...	Came de vita, m...	10	300	1	1	Adauga
4		Ciorba de fasole ...	Ciolan afumat de ...	12	400	1	1	Adauga
5		Ciorba de legume...	Morcov, patrunjel...	9	300	1	1	Adauga
6		Pizza - La Strada ...	sos de roșii, mozz...	18	550	2	1	Adauga
7		Pizza - La Campa...	sos de roșii, mozz...	16	400	2	1	Adauga
8		Pizza - Margherit...	sos de roșii, mozz...	16	350	2	1	Adauga

Below the table, there are three labels and text boxes on the left:

- Necesar zilnic** (label) and **2500** (text box)
- Total kcal** (label) and **0** (text box)
- Pret Total** (label) and **0** (text box)

On the right, there are two buttons:

- Comanda** (yellow button)
- Iesire** (dark grey button)

Etichetele (Labels) au fost adăugate folosind butonul specific din Form controls.

Căsuțele de text (Text Box) au fost create folosind butonul specific din Form Controls și reprezintă controale care vor fi actualizate la fiecare apăsare a unui buton "Adauga"

Butonul a fost adăugat din Form Controls. La apăsarea acestuia, vor fi inserate în baza de date produsele comandate.

Butonul a fost adăugat din Form Controls. La apăsarea acestuia, se va închide forma.

*Meniu Intelligent***3.4.2.1 Particularități**

- La etichete am setat background transparent folosind proprietatea: BackColor = Transparent;
- Pentru butonul "Comanda" am setat culoarea de fundal: BackColor = Info;
- Pentru butonul "Iesire" am setat culoarea de fundal: BackColor = WindowFrame;
- Fontul folosit pentru ambele butoane este: Segoe Print; 20,25pt;
- Pentru butonul "Iesire" am setat culoarea textului: ForeColor = Info;
- Pentru butonul "Comanda" am setat culoarea textului: ForeColor = ControlText;
- La proprietățile formei am setat proprietatea: StartPosition = CenterScreen. Astfel, forma este poziționată în centrul ecranului la fiecare rulare;
- Mărimea formei a fost setată la: 868 (lățime) și 512 (înălțime);
- La apăsare unuia dintre butoanele "Adauga" este executată următoarea secvență de cod:

```
int kcal_necesar = int.Parse(textBox1.Text);
int total_kcal = int.Parse(textBox2.Text);
int total_pret = int.Parse(textBox3.Text);

int cantitate = int.Parse(dataGridView1.Rows[e.RowIndex].Cells[6].Value.ToString());
int kcals = int.Parse(dataGridView1.Rows[e.RowIndex].Cells[4].Value.ToString());
int pret = int.Parse(dataGridView1.Rows[e.RowIndex].Cells[3].Value.ToString());

total_kcal += kcals * cantitate;
total_pret += pret * cantitate;

if (total_kcal <= int.Parse(Login.daily_kcal))
{
    ids.Add(int.Parse(dataGridView1.Rows[e.RowIndex].Cells[0].Value.ToString()));
    quantities.Add(cantitate);

    MessageBox.Show("Produsul a fost adaugat in cos!");

    textBox1.Text = (kcal_necesar - total_kcal).ToString();
    textBox2.Text = total_kcal.ToString();
    textBox3.Text = total_pret.ToString();
}

else
{
    functions.mesajEroare(this, "Ati depasit numarul de calorii necesar!", "Numar
    calorii");
}
```

*Meniu Intelligent*

- La apăsarea butonului ”Calculeaza” este executată următoarea secvență de cod:

```
string connString = @"Data Source=.\SQLEXPRESS;AttachDbFilename=" +  
functions.getDBPath("GOOD_FOOD.mdf") + ";Integrated Security=True;User  
Instance=True";  
using (SqlConnection con = new SqlConnection(connString))  
{  
    try  
    {  
        con.Open();  
        SqlCommand cmd = new SqlCommand(@"INSERT INTO Comenzi (id_client, data_comanda)  
        OUTPUT INSERTED.id_comanda VALUES (@idc, @data)", con);  
  
        cmd.Parameters.AddWithValue("@idc", Login.id_client);  
        cmd.Parameters.AddWithValue("@data", DateTime.Now);  
  
        int id_comanda = (int)cmd.ExecuteScalar();  
  
        cmd.CommandText = @"INSERT INTO Subcomenzi (id_comanda, id_produc, cantitate) VALUES  
        (@idcmd, @idp, @quant)";  
  
        for (int index = 0; index < ids.Count; index++)  
        {  
            cmd.Parameters.AddWithValue("@idcmd", id_comanda);  
            cmd.Parameters.AddWithValue("@idp", ids[index]);  
            cmd.Parameters.AddWithValue("@quant", quantities[index]);  
  
            cmd.ExecuteNonQuery();  
            cmd.Parameters.Clear();  
        }  
  
        ids.Clear();  
        quantities.Clear();  
  
        MessageBox.Show("Comanda a fost efectuata cu succes!");  
  
        this.Close();  
    }  
  
    catch (Exception ex)  
    {  
        functions.mesajEroare(this, "A avut loc o eroare!", "Eroare");  
    }  
}
```

- La apăsarea butonului ”Ieșire” este executată următoare secvență de cod:

```
this.Close();
```

*Meniu Intelligent***3.4.2.2 Forma rezultată**

The screenshot shows a window titled 'Optiuni' with three tabs: 'CalculatoareKcal', 'Comanda', and 'Grafic Kcal'. The 'Comanda' tab is active, displaying a table with the following data:

	id_produs	denumire_produs	descriere	pret	kcal	felul	Canitate	Adauga
▶	1	Ciorba de burta ...	Burta de vita, sm...	9	400	1	1	Adauga
	2	Ciorba radautean...	Piept de pui, sm...	12	450	1	1	Adauga
	3	Ciorba de vacuta...	Came de vita, m...	10	300	1	1	Adauga
	4	Ciorba de fasole ...	Ciolan afumat de ...	12	400	1	1	Adauga
	5	Ciorba de legume...	Morcov, patrunjel...	9	300	1	1	Adauga
	6	Pizza - La Strada ...	sos de ro?ii, mozz...	18	550	2	1	Adauga
	7	Pizza - La Campa...	sos de ro?ii, mozz...	16	400	2	1	Adauga
	8	Pizza - Margherit...	sos de ro?ii, mozz...	16	350	2	1	Adauga
	9	Pizza - Vegetara...	sos de ro?ii, mozz...	10	350	2	1	Adauga

Below the table, there are three summary fields and two buttons:

- Necesar zilnic**: 2500
- Total kcal**: 0
- Pret Total**: 0
- Comanda** (yellow button)
- lesire** (dark grey button)

**3.4.3 Grafic Kcal**

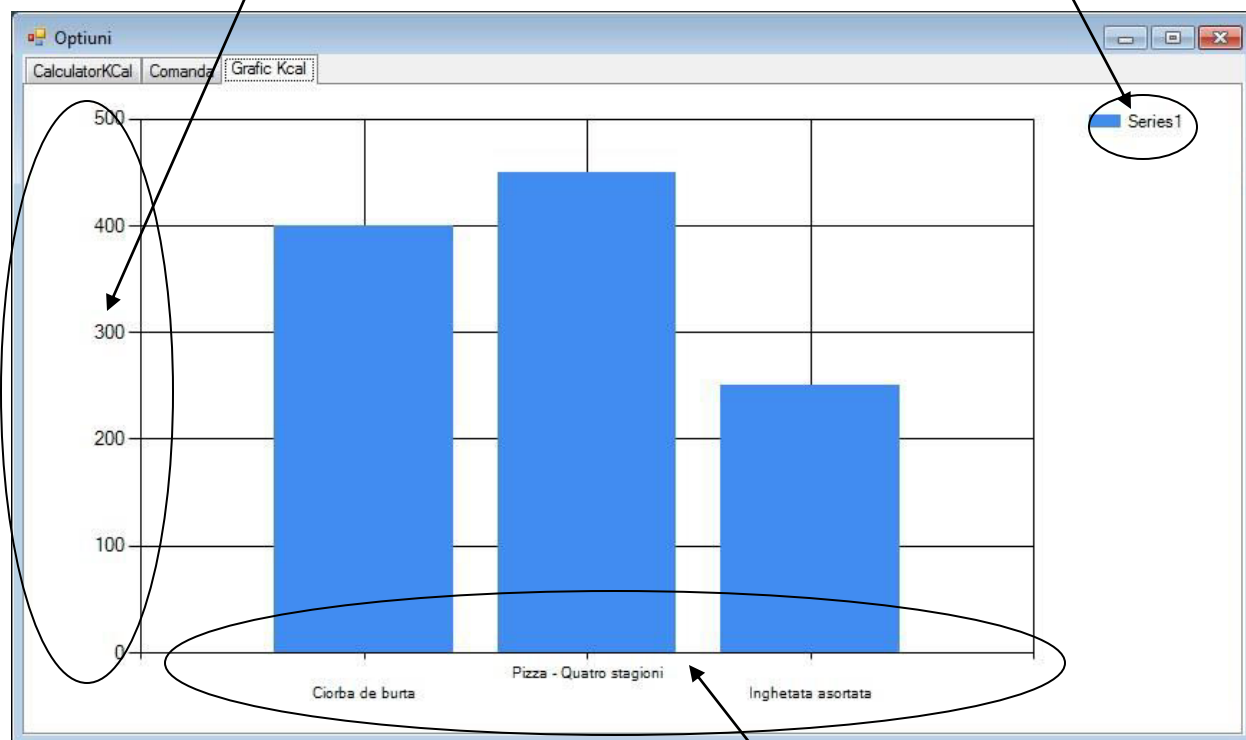
Tabul GraficKcal este ultimul dintre cele trei taburi ale formei Optiuni. Aceasta constă într-un control de tip Chart. A fost concepută interactiv folosind new form și adăugând elementele după cum urmează:



*Meniu Intelligent*

Valorile de pe axa Oy reprezintă numărul de kcal conținut de fiecare produs comandat în tabul Comanda în ziua logării.

Graficul prezentat conține numărul total de kcalorii pentru fiecare produs comandat în tabul Comanda în ziua logării.



Valorile de pe axa Ox reprezintă denumirea fiecărui produs comandat în tabul Comanda în ziua logării.

*Meniu Intelligent***3.4.3.1 Particularități**

- La proprietățile formei am setat proprietatea: StartPosition = CenterScreen. Astfel, forma este poziționată în centrul ecranului la fiecare rulare;
- Mărimea formei a fost setată la: 868 (lățime) și 512 (înălțime);
- Am setat tipul graficului prin: ChartType = Column;
- Culorile graficului au fost setate prin: Palette = BrightPastel;
- A fost creată funcția extractProducts() astfel:

```
private void extractProducts(SqlDataReader reader, int id_comanda, string connString)
{
    using (SqlConnection conSubcomenzi = new SqlConnection(connString))
    {
        conSubcomenzi.Open();

        SqlCommand cmdSubcomenzi = new SqlCommand(@"SELECT * FROM Subcomenzi WHERE id_comanda
        = @idcomanda", conSubcomenzi);
        cmdSubcomenzi.Parameters.AddWithValue(@"idcomanda", id_comanda);

        SqlDataReader readerSubcomenzi = cmdSubcomenzi.ExecuteReader();

        while (readerSubcomenzi.Read())
        {
            int produs = int.Parse(readerSubcomenzi["id_produs"].ToString());
            int cantitate = int.Parse(readerSubcomenzi["cantitate"].ToString());

            using (SqlConnection conMeniu = new SqlConnection(connString))
            {
                conMeniu.Open();

                SqlCommand cmdMeniu = new SqlCommand(@"SELECT * FROM Meniu WHERE id_produs = @idp",
                conMeniu);

                cmdMeniu.Parameters.AddWithValue(@"idp", produs);

                SqlDataReader readerMeniu = cmdMeniu.ExecuteReader();

                readerMeniu.Read();

                int kcals = int.Parse(readerMeniu["kcal"].ToString());
                string nume = readerMeniu["denumire_produs"].ToString();

                chart1.Series[0].Points.AddXY(nume, kcals * cantitate);
            }
        }
    }
}
```

*Meniu Intelligent*

➤ La încărcarea tabului este executată următoarea secvență de cod:

```
string connString = @"Data Source=.\SQLEXPRESS;AttachDbFilename=" +  
functions.getDBPath("GOOD_FOOD.mdf") + ";Integrated Security=True;User Instance=True";  
using (SqlConnection con = new SqlConnection(connString))  
{  
    con.Open();  
    SqlCommand cmd = new SqlCommand(@"SELECT * FROM Comenzi WHERE id_client = @idc", con);  
    cmd.Parameters.AddWithValue("@idc", Login.id_client);  
  
    SqlDataReader reader = cmd.ExecuteReader();  
  
    if (reader.Read())  
    {  
  
        DateTime data_comanda = DateTime.Parse(reader["data_comanda"].ToString());  
        DateTime data_azi = DateTime.Now;  
  
        if (data_comanda.Day == data_azi.Day && data_comanda.Month == data_azi.Month &&  
            data_comanda.Year == data_azi.Year)  
        {  
  
            noCommandLbl.Visible = false;  
            chart1.Visible = true;  
  
            int id_comanda = int.Parse(reader["id_comanda"].ToString());  
  
            extractProducts(reader, id_comanda, connString);  
  
            while (reader.Read())  
            {  
                id_comanda = int.Parse(reader["id_comanda"].ToString());  
                extractProducts(reader, id_comanda, connString);  
            }  
        }  
        else  
        {  
            int k = 0;  
  
            while (reader.Read())  
            {  
                data_comanda = DateTime.Parse(reader["data_comanda"].ToString());  
                data_azi = DateTime.Now;  
  
                if (data_comanda.Day == data_azi.Day && data_comanda.Month == data_azi.Month &&  
                    data_comanda.Year == data_azi.Year)  
                {  
  
                    noCommandLbl.Visible = false;  
                    chart1.Visible = true;  
  
                    int id_comanda = int.Parse(reader["id_comanda"].ToString());
```

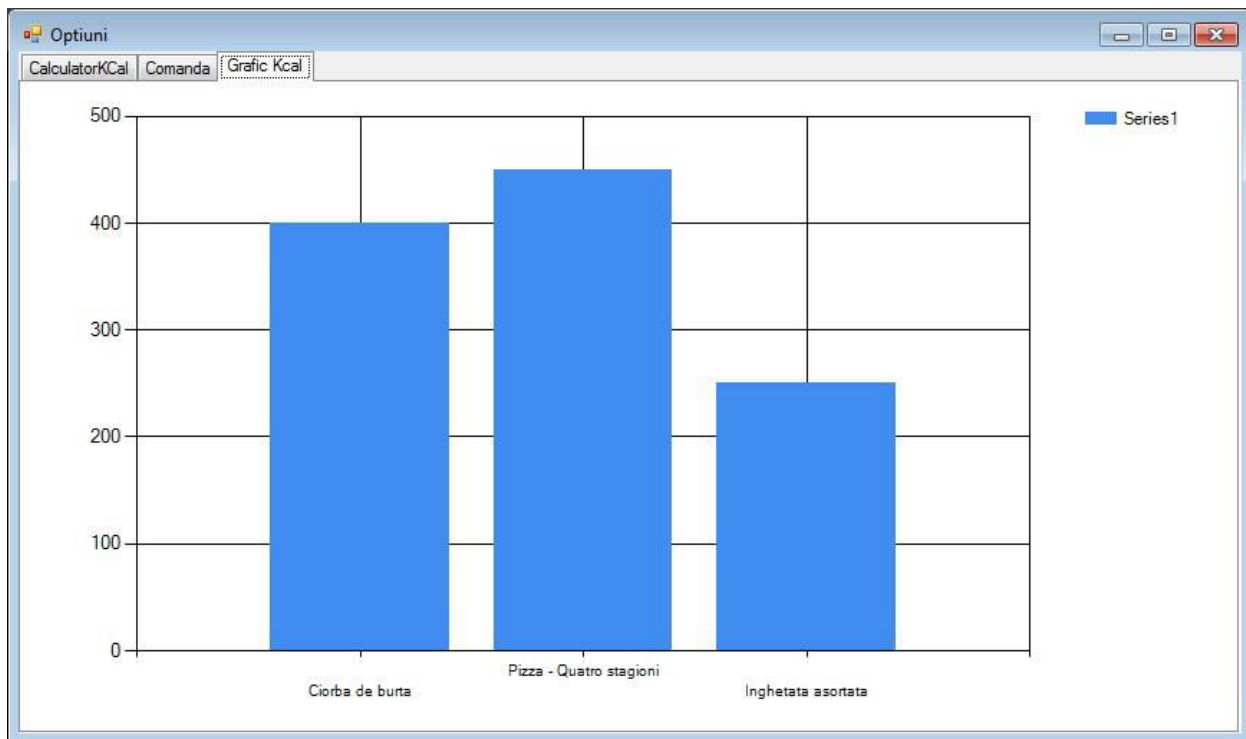
*Meniu Intelligent*

```
extractProducts(reader, id_comanda, connString);
```

```
k++;  
}  
}
```

```
if (k == 0)  
{  
    noCommandLbl.Visible = true;  
    chart1.Visible = false;  
}  
}  
}
```

```
else  
{  
    noCommandLbl.Visible = true;  
    chart1.Visible = false;  
}  
}  
}
```

**3.4.3.2 Forma rezultată**

*Meniu Inteligent***4. Documentație de proiectare ( Administrator)****4.1 Admin**

Forma corespunzătoare utilizatorului de tip administrator a fost creată interactiv de la new form și adăugând elemente după cum urmează:

**Butonul a fost adăugat din Form Controls. La apăsarea acestuia, datele din căsuțele text vor fi adăugate în baza de date.**

**Butonul a fost adăugat din Form Controls. La apăsarea acestuia, rândul selectat mai jos va fi șters din baza de date.**

**Etichetele (Labels) au fost adăugate folosind butonul specific din Form controls.**

**Căsuțele de text (Text Box) au fost create folosind butonul specific din Form Controls și reprezintă controale către datele introduse ce vor coincide.**

**Controlul de tip DataGridView a fost creat folosind butonul specific din Form Controls și aici sunt afișate datele din tabelul Meniu din baza de date.**

id_produc	denumire_produc	descriere	pret	kcal	felul
1	Ciorba de burta ...	Burta de vita, sm...	9	400	1
2	Ciorba radautean...	Piept de pui, sm...	12	450	1
3	Ciorba de vacuta...	Came de vita, m...	10	300	1
4	Ciorba de fasole ...	Ciolan afumat de ...	12	400	1
5	Ciorba de legume ...	Morcov,patrunjel,...	9	300	1
6	Pizza - La Strada ...	sos de ro'ii, mozz...	18	550	2
7	Pizza - La Campa...	sos de ro'ii, mozz...	16	400	2
8	Pizza - Margherit...	sos de ro'ii, mozz...	16	350	2
9	Pizza - Vegetaria...	sos de ro'ii, mozz...	19	350	2
10	Pizza - Quatro sta...	sos de ro'ii, mozz...	20	450	2
11	Pizza - Quatro for...	sos de ro'ii, mozz...	20	500	2
12	Pizza - Napoli ...	sos de ro'ii, mozz...	21	450	2
13	Pui Shanghai cu ...	piept de pui, pros...	20	450	2
14	Piept de pui ceda...	pui, brânza cedar...	21	500	2
15	Pui cu smantana ...	piept de pui, ciup...	15	500	2
16	Muschi de vita în...	muschi de vita, vi...	22	500	2
17	Muschi de vita în...	muschi de vita, c...	23	600	2

*Meniu Intelligent***4.1.1 Particularități**

- La etichetele care denumesc câmpurile ce urmează să fie introduse am setat background transparent folosind proprietatea: BackColor = Transparent.
- La proprietățile formei am setat proprietatea: StartPosition = CenterScreen. Astfel, forma este poziționată în centrul ecranului la fiecare rulare;
- Mărimea formei a fost setată la: 664 (lățime) și 724 (înălțime);
- Pentru butonul "Insert" am setat culoarea de fundal: BackColor = Info;
- Pentru butonul "Delete" am setat culoarea de fundal: BackColor = WindowFrame;
- Fontul folosit pentru ambele butoane este: Segoe Print; 20,25pt;
- Pentru butonul "Delete" am setat culoarea textului: ForeColor = Info;
- Pentru butonul "Insert" am setat culoarea textului: ForeColor = ControlText;
- Pentru afișarea datelor din tabelul Meniu am utilizat un control de tip DataGridView. Pentru acesta am setat proprietatea: Dock = Bottom;
- Pentru selectarea datelor din tabelul Meniu am creat funcția "selectMeniu()":

```
Functions functions = new Functions();

string connString = @"Data Source=.\SQLEXPRESS;AttachDbFilename=" +
functions.getDBPath("GOOD_FOOD.mdf") + @";Integrated Security=True;User
Instance=True";

using (SqlConnection con = new SqlConnection(connString))
{
    try
    {
        con.Open();
        SqlCommand cmd = new SqlCommand(@"SELECT * FROM Meniu", con);

        SqlDataAdapter adapter = new SqlDataAdapter(cmd);

        DataTable table = new DataTable();
        adapter.Fill(table);

        dataGridView1.DataSource = table;
    }

    catch (Exception ex)
    {
        functions.mesajEroare(this, "A aparut o eroare!", "Eroare");
    }
}
```

- La apăsarea butonului "Insert" este executată următoarea secvență de cod:

*Meniu Intelligent*

```

Functions functions = new Functions();

string connString = @"Data Source=.\SQLEXPRESS;AttachDbFilename="+
functions.getDBPath("GOOD_FOOD.mdf") + @";Integrated Security=True;User
Instance=True";

using (SqlConnection con = new SqlConnection(connString))
{
    try
    {
        con.Open();
        SqlCommand cmd = new SqlCommand(@"INSERT INTO Meniu (id_produș, denumire_produș,
descriere, pret, kcal, felul) VALUES (@id, @den, @desc, @pret, @kcal, @fel)",
con);

        cmd.Parameters.AddWithValue("@id", dataGridView1.Rows.Count);
        cmd.Parameters.AddWithValue("@den", nameTxt.Text);
        cmd.Parameters.AddWithValue("@desc", textBox1.Text);
        cmd.Parameters.AddWithValue("@pret", textBox2.Text);
        cmd.Parameters.AddWithValue("@kcal", textBox3.Text);
        cmd.Parameters.AddWithValue("@fel", textBox4.Text);

        cmd.ExecuteNonQuery();

        MessageBox.Show("Inserarea a avut loc cu succes!");

        nameTxt.Text = "";
        textBox1.Text = "";
        textBox2.Text = "";
        textBox3.Text = "";
        textBox4.Text = "";

        selectMeniu();
    }
    catch (Exception ex)
    {
        functions.mesajEroare(this, "A aparut o eroare!", "Eroare");
    }
}

```

- La apăsarea butonului "Delete" este executată următoarea secvență de cod:

```

Functions functions = new Functions();

string connString = @"Data Source=.\SQLEXPRESS;AttachDbFilename=" +
functions.getDBPath("GOOD_FOOD.mdf") + @";Integrated Security=True;User
Instance=True";

using (SqlConnection con = new SqlConnection(connString))
{
    try
    {
        con.Open();

```

*Meniu Intelligent*

```

SqlCommand cmd = new SqlCommand(@"DELETE FROM Meniu WHERE id_produs = @id",
con);

cmd.Parameters.AddWithValue("@id", id.ToString());

cmd.ExecuteNonQuery();

MessageBox.Show("Stergerea a avut loc cu succes!");

selectMeniu();
}

catch (Exception ex)
{
functions.mesajEroare(this, "A aparut o eroare!", "Eroare");
}
}

```

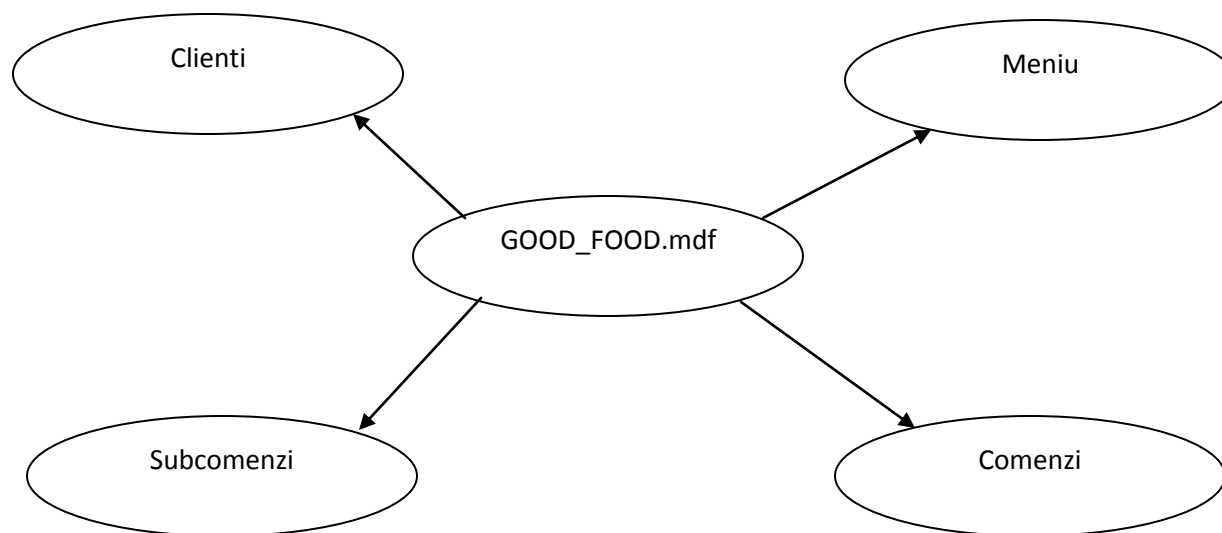
**4.1.2 Forma rezultată**

	id_produs	denumire_produs	descriere	pret	kcal	felul
▶	1	Giorba de burta ...	Burta de vita, sm...	9	400	1
	2	Giorba radautean...	Piept de pui, sm...	12	450	1
	3	Giorba de vacuta...	Came de vita, m...	10	300	1
	4	Giorba de fasole ...	Ciolan afumat de ...	12	400	1
	5	Giorba de legume...	Morcov, patrunjel,...	9	300	1
	6	Pizza - La Strada ...	sos de ro?ii, mozz...	18	550	2
	7	Pizza - La Campa...	sos de ro?ii, mozz...	16	400	2
	8	Pizza - Margherit...	sos de ro?ii, mozz...	16	350	2
	9	Pizza - Vegetaria...	sos de ro?ii, mozz...	19	350	2
	10	Pizza - Quatro sta...	sos de ro?ii, mozz...	20	450	2
	11	Pizza - Quatro for...	sos de ro?ii, mozz...	20	500	2
	12	Pizza - Napoli ...	sos de ro?ii, mozz...	21	450	2
	13	Pui Shanghai cu ...	piept de pui, pros...	20	450	2
	14	Piept de pui ceda...	pui, brânza cedar...	21	500	2
	15	Pui cu smantana ...	piept de pui, ciup...	15	500	2
	16	Muschi de vita în...	muschi de vita, vi...	22	500	2
	17	Muschi de vita în...	muschi de vita, c...	23	600	2



## 5. Tabele

### 5.1. Schemă



### 5.2 Tabele

#### Clienti

dbo.Clienti: Tab...T\GOOD_FOOD.MDF			
	Column Name	Data Type	Allow Nulls
	id_client	int	<input type="checkbox"/>
	parola	char(50)	<input type="checkbox"/>
	nume	char(50)	<input type="checkbox"/>
	prenume	char(50)	<input type="checkbox"/>
	adresa	char(100)	<input type="checkbox"/>
	email	char(100)	<input type="checkbox"/>
	kcal_zilnice	int	<input type="checkbox"/>
	admin	char(10)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

*Meniu Intelligent***Meniu**

dbo.Meniu: Tabl...T\GOOD_FOOD.MDF) X			
	Column Name	Data Type	Allow Nulls
PK	id_produș	int	<input type="checkbox"/>
	denumire_produș	char(50)	<input type="checkbox"/>
	descriere	text	<input type="checkbox"/>
	pret	int	<input type="checkbox"/>
	kcal	int	<input type="checkbox"/>
	felul	int	<input type="checkbox"/>
			<input type="checkbox"/>

**Comenzi**

dbo.Comenzi: Ta...T\GOOD_FOOD.MDF) X			
	Column Name	Data Type	Allow Nulls
PK	id_comanda	int	<input type="checkbox"/>
	id_client	int	<input type="checkbox"/>
	data_comanda	datetime	<input type="checkbox"/>
			<input type="checkbox"/>

**Subcomenzi**

dbo.Subcomenzi:...GOOD_FOOD.MDF) X			
	Column Name	Data Type	Allow Nulls
PK	id_subcomanda	int	<input type="checkbox"/>
	id_comanda	char(100)	<input type="checkbox"/>
	id_produș	int	<input type="checkbox"/>
	cantitate	int	<input type="checkbox"/>
			<input type="checkbox"/>

## **6. Bibliografie**

- Gălăţan, Constantin şi Gălăţan, Susana. *Curs de C# - Programare în Visual C# 2008 Express Edition*. Editura L&S INFOMAT
- <https://msdn.microsoft.com>
- <https://stackoverflow.com/>