**LICEUL DE INFORMATICA “TIBERIU POPOVICIU”**

**CLUJ-NAPOCA**

PROIECT PENTRU OBŢINEREA ATESTĂRII  
PROFESIONALE IN INFORMATICĂ

TITLUL LUCRĂRII:

**Sistem de gestiune a comenzilor în restaurante, baruri şi cafenele**

MUNTEAN CRISTIAN

CLASA A XII-A A

**CLUJ-NAPOCA**

IUNIE, 2020

**CUPRINS**

[PREZENTAREA GENERALA A TEMEI 3](#_Toc42424700)

[PROBLEMATICA ABORDATA 3](#_Toc42424701)

[DESCRIEREA APLICATIEI 3](#_Toc42424702)

[BENEFICII 3](#_Toc42424703)

[PUBLICUL-TINTA 3](#_Toc42424704)

[RESURSE HARDWARE SI SOFTWARE NECESARE 4](#_Toc42424705)

[CERINTE HARDWARE 4](#_Toc42424706)

[CERINTE SOFTWARE 4](#_Toc42424707)

[REALIZAREA APLICATIEI 5](#_Toc42424708)

[MENIURI 5](#_Toc42424709)

[BAZA DE DATE 9](#_Toc42424710)

[SUBPROGRAME 9](#_Toc42424711)

[POSIBILE EXTINDERI ALE APLICATIEI 12](#_Toc42424712)

# **PREZENTAREA GENERALĂ A TEMEI**

## **PROBLEMATICA ABORDATĂ**

Aplicația a fost realizată cu scopul de a contribui la optimizarea si digitalizarea activității restaurantelor, barurilor şi cafenelelor, reducând timpul necesar pentru gestiunea comenzilor.

## **DESCRIEREA APLICAŢIEI**

Aplicația se deschide cu o pagina de logare, in care chelnerii trebuie sa îşi introducă datele (nume de utilizator + parolă) la începutul turei. Odată intrați in cont, chelnerii pot selecta mesele pentru a introduce comenzi; la încheierea unei comenzi, masa respectiva se eliberează, iar încasările vor fi înregistrate pentru totalizare pentru fiecare tura.

## **BENEFICII**

Cu ajutorul aceste aplicații, prelucrarea comenzilor devine mult mai ușoara. Chelnerii nu mai trebuie sa retina prețul fiecărui produs, acestea fiind introduse intr-o baza de date. De asemenea, se pot realiza statistici, precum numărul de comenzi preluate de către fiecare chelner sau încasările totale în decursul unei zile.

## **PUBLICUL-ŢINTĂ**

Aceasta aplicație este destinată persoanelor care administrează sau lucrează la un restaurant, cafenea sau bar.

# **RESURSE HARDWARE ȘI SOFTWARE NECESARE**

Pentru ca aplicația sa ruleze in condiții optime sunt necesare următoarele cerințe hardware si software.

## **CERINŢE HARDWARE**

* Procesor de 2GHz
* 1GB RAM

## **CERINŢE SOFTWARE**

* Sistem de operare: Windows 10 (32-bit sau 64-bit)
* 500MB spațiu liber
* Microsoft SQL Server 2019

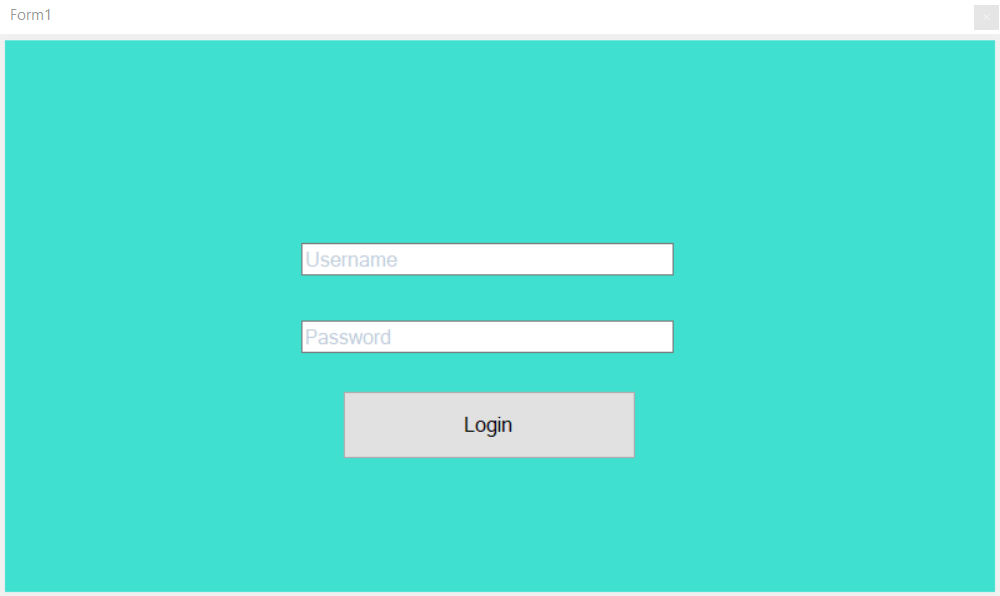
# **REALIZAREA APLICAŢIEI**

## **MENIURI**

##### **Pagina de login**

Pagina de login are următoarele funcții:

* asigură logarea personalului autorizat;
* avertizează daca numele de utilizator şi/ sau parola sunt invalide;
* recunoaște şi atenționează dacă funcția „Caps lock” este activă.

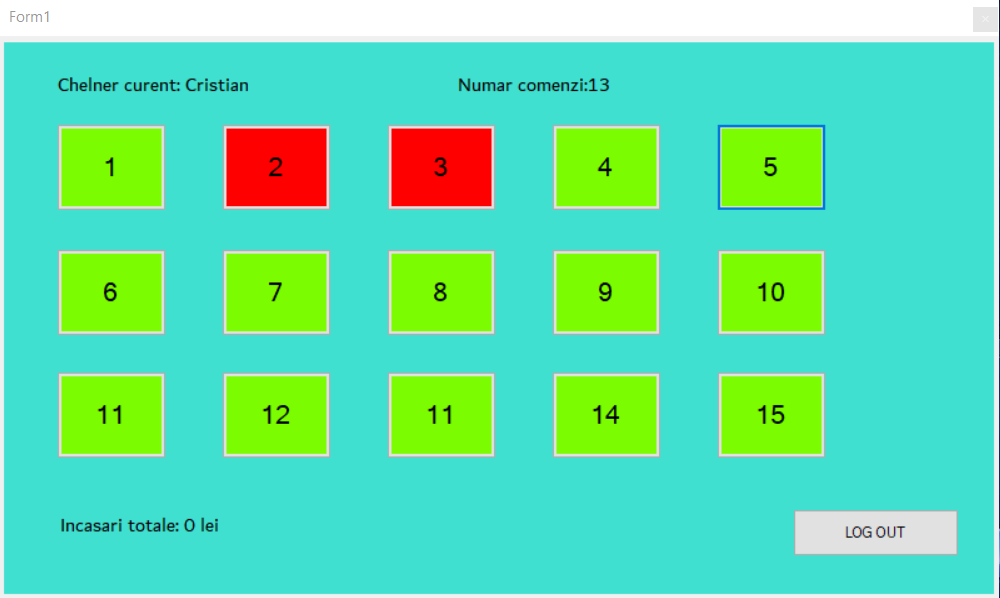


##### **Meniul principal**

Intr-o formă grafică intuitivă, mesele sunt reprezentate de butoanele inscripționate cu numere de la 1 la 15: dacă un buton este verde, înseamnă ca masa asociata acestuia este liberă, iar daca acesta este roșu, înseamnă ca exista o comandă asociata mesei respective, deci masa este ocupata.

Meniu asigură:

* accesarea fiecărei mese şi comanda asociată acesteia;
* schimbarea contului activ prin apăsarea butonului de logout;
* vizualizarea contului activ;
* vizualizarea numărului de comenzi finalizate de către chelner;
* vizualizarea încasărilor totale din tura respectivă.



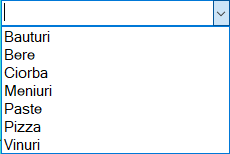
##### **Meniul de adăugare a unei comenzi**

Acest meniu reprezintă interfața bazei de date pe care o gestionează aplicația, asigurȃnd selecția produselor comandate de clienți, totalizarea preturilor acestora şi atribuirea lor unei anumite mese

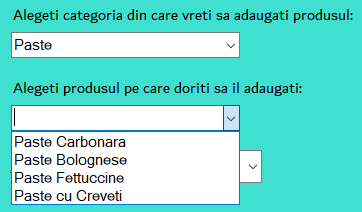
****

După selecția unei mese, aplicația deschide meniul în care se pot adăuga sau scoate produse dintr-o comandă. De exemplu, în captura de ecran de mai sus se poate observa comanda curentă de la masa 3.

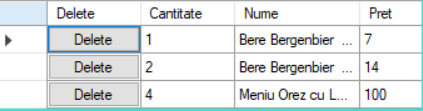
Cu ajutorul ComboBox-urilor se poate selecta atât tipul produselor, cat si numărul acestora:



În funcție de categoria selectată în primul ComboBox, se activează un al doilea, care va conține produsele din categoria selectată. În exemplul de mai jos, categoria selectată este cea a pastelor; al doilea ComboBox va conține sortimentele disponibile de paste şi se va actualiza automat:



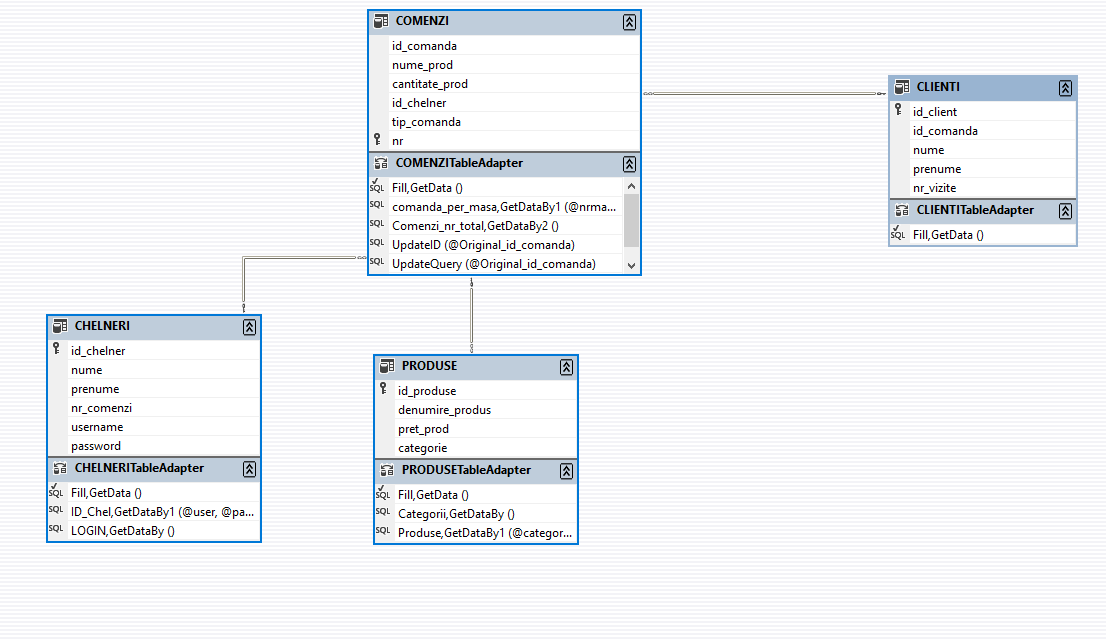
Daca se doreşte eliminarea unui produs din comandă se poate selecta comanda de ştergere (butonul „Delete” aflat lȃngă produs, în lista care afișează produsele din comandă):



## **BAZA DE DATE**

Baza de date relațională include 4 tabele. Pe baza diagramei entitate-relație evidențiază următoarele aspecte:

* fiecare produs are atât o categorie, cȃt şi un preț;
* fiecare comandă poate conține mai multe produse;
* în fiecare comandă este stocat numărul de produse;
* fiecare chelner are asociat un nume de utilizator, o parola şi numărul total de comenzi finalizate de acesta



## **SUBPROGRAME**

##### **Subprogramul folosit pentru logarea intr-unul din conturile chelnerilor**

private void Login\_butotn\_Click(object sender, EventArgs e)

{

bool ok = false;

int ch=0;

cHELNERITableAdapter.LOGIN(baza\_atestatDataSet.CHELNERI);

DataTable dt = baza\_atestatDataSet.CHELNERI;

string user = Username.Text, pass = Password.Text;

for (int i = 0; i < dt.Rows.Count; i++)

{

string user1 = dt.Rows[i]["username"].ToString(),user2="",pass1=dt.Rows[i]["password"].ToString(),pass2="";

int j = 0;

while ((user1[j] >= 'A' && user1[j] <= 'Z') || (user1[j] >= 'a' && user1[j] <= 'z') || (user1[j] >= '0' && user1[j] <= '9'))

{

user2 = user2 + user1[j];

j++;

}

j = 0;

while ((pass1[j] >= 'A' && pass1[j] <= 'Z') || (pass1[j] >= 'a' && pass1[j] <= 'z') || (pass1[j] >= '0' && pass1[j] <= '9'))

{

pass2 = pass2 + pass1[j];

j++;

}

if (user.Equals(user2) == true && pass.Equals(pass2) == true)

{

ok = true;

ch = i;

}

}

if (ok == true)

{

tabControl1.SelectedTab = Main\_menu;

User = User + user;

Pass = Pass + pass;

string chel = dt.Rows[ch]["prenume"].ToString();

Chelner.Text +=chel;

Username.Text = "Username";

Username.ForeColor = Color.Silver;

Password.Text = "Password";

Password.ForeColor = Color.Silver;

cHELNERITableAdapter.ID\_Chel(baza\_atestatDataSet.CHELNERI, User, Pass);

DataTable ID = baza\_atestatDataSet.CHELNERI;

if (ID.Rows.Count > 0)

ID\_chelner = (int)ID.Rows[ID.Rows.Count - 1]["id\_chelner"];

cOMENZITableAdapter.Comenzi\_nr\_total(baza\_atestatDataSet.COMENZI);

DataTable comenzi = baza\_atestatDataSet.COMENZI;

if (comenzi.Rows.Count > 0)

KeyCom = Convert.ToInt32(comenzi.Rows[0]["MAX"]);

cHELNERITableAdapter.Fill(baza\_atestatDataSet.CHELNERI);

DataTable nr\_com = baza\_atestatDataSet.CHELNERI;

comenzi123.Text= "Numar comenzi:" + nr\_com.Rows[ID\_chelner - 1]["nr\_comenzi"];

}

else Error.Text = "Username-ul sau parola nu sunt corecte. Va rugam incercati din nou.";

}

Acesta retine numele de utilizator si parola introduse in TextBox-urile de pe pagina de login si le compara cu toate conturile stocate in baza de date. Daca se potrivesc cu unul din conturi, utilizatorul este redirectionat catre pagina principala, în caz contrar se afișează un meesaj.

##### **Subprogramul folosit pentru redirecționarea către meniul de comanda a unei mese**

private void Masa1\_Click(object sender, EventArgs e)

{

this.cOMENZITableAdapter.Update(baza\_atestatDataSet.COMENZI);

this.cOMENZITableAdapter.Fill(this.baza\_atestatDataSet.COMENZI);

tabControl1.SelectedTab = Menu;

mese[1] = 1;

Masa1.BackColor = Color.Red;

selectedTable = 1;

comanda.Text = "Comanda de la masa 1 este:";

cOMENZITableAdapter.comanda\_per\_masa(baza\_atestatDataSet.COMENZI, selectedTable);

DataTable dt = baza\_atestatDataSet.COMENZI;

plata = 0;

for (int i = 0; i < dt.Rows.Count; i++)

{

plata += Convert.ToInt32(dt.Rows[i]["Preț"]);

}

Total.Text = plata.ToString() + " lei";

disp\_data();

}

In momentul in care este apasata un buton masa devine ocupata si programul redirectionează către comanda asociata mesei respective. Daca nu este introdus niciun produs si se revine la meniul principal atunci subprogramul următor va elibera masa.

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

{

tabControl1.SelectedTab = Main\_menu;

cOMENZITableAdapter.Fill(baza\_atestatDataSet.COMENZI);

DataTable dt = baza\_atestatDataSet.COMENZI;

bool ok = true;

for(int i=0;i<grid.Rows.Count;i++)

{

if (Convert.ToInt32(grid.Rows[i]["id\_comanda"]) == selectedTable) ok = false;

}

if(ok==true)

{

mese[selectedTable] = 0;

if (selectedTable == 1) Masa1.BackColor = Color.LawnGreen;

if (selectedTable == 2) Masa2.BackColor = Color.LawnGreen;

if (selectedTable == 3) Masa3.BackColor = Color.LawnGreen;

if (selectedTable == 4) Masa4.BackColor = Color.LawnGreen;

if (selectedTable == 5) Masa5.BackColor = Color.LawnGreen;

if (selectedTable == 6) Masa6.BackColor = Color.LawnGreen;

if (selectedTable == 7) Masa7.BackColor = Color.LawnGreen;

if (selectedTable == 8) Masa8.BackColor = Color.LawnGreen;

if (selectedTable == 9) Masa9.BackColor = Color.LawnGreen;

if (selectedTable == 10) Masa10.BackColor = Color.LawnGreen;

if (selectedTable == 11) Masa11.BackColor = Color.LawnGreen;

if (selectedTable == 12) Masa12.BackColor = Color.LawnGreen;

if (selectedTable == 13) Masa13.BackColor = Color.LawnGreen;

if (selectedTable == 14) Masa14.BackColor = Color.LawnGreen;

if (selectedTable == 15) Masa15.BackColor = Color.LawnGreen;

}

grid.Clear();

this.cHELNERITableAdapter.Update(baza\_atestatDataSet.CHELNERI);

this.cHELNERITableAdapter.Fill(this.baza\_atestatDataSet.CHELNERI);

DataTable dti = baza\_atestatDataSet.CHELNERI;

comenzi123.Text = "Numar comenzi: " + dti.Rows[ID\_chelner - 1]["nr\_comenzi"];

}

##### **Subprogramul pentru adăugarea un produs intr-o comandă**

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

{

if (Produse.Text != "" && Nr\_prod.Text != "")

{

KeyCom++;

con.Open(); ///De aici incepe inserarea unei linii noi in tabel

SqlCommand cmd = con.CreateCommand();

cmd.CommandType = CommandType.Text;

cmd.CommandText = "insert into COMENZI (id\_comanda, nume\_prod, cantitate\_prod, id\_chelner, tip\_comanda, nr) values (@id\_comanda,@nume\_prod,@cantitate\_prod,@id\_chelner,@tip\_comanda,@nr)";

cmd.Parameters.AddWithValue("@id\_comanda", selectedTable);

cmd.Parameters.AddWithValue("@nume\_prod", Produse.SelectedItem.ToString());

cmd.Parameters.AddWithValue("@cantitate\_prod", int.Parse(Nr\_prod.SelectedItem.ToString()));

cmd.Parameters.AddWithValue("@id\_chelner", ID\_chelner);

cmd.Parameters.AddWithValue("@tip\_comanda", "Aici");

cmd.Parameters.AddWithValue("@nr", KeyCom);

cmd.ExecuteNonQuery();

this.cOMENZITableAdapter.Update(baza\_atestatDataSet.COMENZI);

this.cOMENZITableAdapter.Fill(this.baza\_atestatDataSet.COMENZI);

con.Close(); ///Aici se termina inserarea

disp\_data();

Categorie.Text = "";

Produse.Items.Clear();

Produse.Text = "";

Nr\_prod.Text = "";

cOMENZITableAdapter.comanda\_per\_masa(baza\_atestatDataSet.COMENZI, selectedTable);

DataTable dt = baza\_atestatDataSet.COMENZI;

plata = 0;

for (int i = 0; i < grid.Rows.Count; i++)

{

plata += Convert.ToInt32(grid.Rows[i]["Preț"]);

}

Total.Text = plata.ToString() + " lei";

}

# **}POSIBILE EXTINDERI ALE APLICAŢIEI**

Aplicația poate fi extinsă cu minime adaptări în cel puțin două direcții:

* fidelizarea clienților - tabelul “Clienți” din baza de date nu este folosit la nimic in prezent, însă ar putea fi folosit la contorizarea clienților după nume, prenume si numărul de vizite ale acestora, putȃndu-se introduce un discount daca un client are un număr stabilit de vizite(de exemplu tot la a treia vizita a unui client acesta sa primească un discount de 10%);
* înregistrare COVID 19 a clienților şi rezervarea rapidă a meselor pentru aceştia – date fiind noile reglementări, folosind acelaşi tabel „Clienți” şi modificȃnd în mod corespunzător meniul de modificare a comenzilor.