**Ministerul Educației, Culturii și Cercetării al Republicii Moldova**

**Centrul de Excelență în Informatică și Tehnologii Informaționale**

**Catedra Informatica aplicată**

**RAPORT**

pentru practica tehnologică

Tema: **Abonamente**

Realizat **Damanciuc Xenia**

Verificat **Damaschin Ion**

Nota **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Cuprins**

[**Introducere** 3](#_Toc514746827)

[**1.** **Condiția sarcinii individuale** 3](#_Toc514746828)

[**2.** **Proiectarea bazei de date** 4](#_Toc514746829)

[**2.1** **Crearea modelului obiect** 4](#_Toc514746830)

[**2.2** **Crearea și descrierea modelului relațional** 4](#_Toc514746831)

[**2.3** **Normalizarea și crearea bazei de date** 7](#_Toc514746832)

[**3.** **Elaborarea aplicației** 8](#_Toc514746833)

[**3.1 Descrierea informației de intrare** 8](#_Toc514746834)

[**3.2** **Descrierea datelor de ieșire** 10](#_Toc514746835)

[**4.** **Ghid de utilizare a aplicației** 11](#_Toc514746836)

[**4.1 Produsul cu care este construită baza de date** 11](#_Toc514746837)

[**4.2** **Cerințe minime hardware și software pentru buna funcționare a aplicației** 11](#_Toc514746838)

[**4.3** **Descrierea aplicației în baza sarcinilor individuale** 12](#_Toc514746839)

# **Introducere**

* **Scopul practicii**

Stagiul de practică reprezintă o modalitate de prosperare care dezvoltă abilitățile  
profesioniste. În cadrul acesteia se realizează un schimb de experiență care oferă noi viziuni legate de viitorul loc de muncă. Stagiul de practică servește la stabilirea dorințelor referitor la locul viitor de muncă. De asemenea, ne face cunoștință cu posibilile dificultăți care le vom putea întîlni pe parcursul activării în domeniul dat, precum și îmbunătățește capacitățile de colaborare în grup.

Stagiul de practică oferă șansa de a îmbina teoria cu practica. Este un avantaj foarte mare,  
întrucît pe parcursul anului de învățămînt nu se reușește să se ofere mai multă atenție practicii.

Îmbinarea dintre limbajul de programare C# și crearea bazelor de date este utilizată frecvent  
pe piață. De aceea, practica tehnologică are un impact favorabil asupra studenților datorită necesității dezvoltarea unor aplicații cerute pe piață.

* **Obiective generale**
* Consolidarea cunoștințelor acumulate legate de limbajul de programare C# și crearea bazelor de date prin SQL
* Dezvoltarea abilităților muncii independente
* Deprinderea cu regimul de lucru

# **Condiția sarcinii individuale**

Informația despre ziarele editate de mai multe redacții și abonații la aceste ziare este înregistrată într-o bază de date. Se elaborează o bază de date, gestiunea căreia se va realiza dintr-o aplicație vizuală. Informațiile din baza de date se vor prelucra la solicitarea utilizatorului folosind meniuri, formulare și control-ere, ținând cont de următoarele funcționalități:

1. Înregistrează un nou abonat;
2. Exclude din baza de date înregistrarea ce corespunde abonatului introdus de la tastatură;
3. Determină ziarele abonatului cu numele și prenumele introduse de la tastatură;
4. Afișează la ecran atributele abonatului cu numele, prenumele și adresa, care corespunde unui anumit titlu de ziar introdus de la tastatură;
5. Determină numărul de ziare, ale unei redacții selectate din lista propusă;
6. Afișează la ecran abonații, în ordinea alfabetică a numelor acestora;
7. Determină prețul total al ziarelor editate de o anumită redacție selectată din lista propusă;
8. Exportă într-un fișier MS Word sau MS Excel lista ziarelor;
9. Afișează lista ziarelor editate în limba aleasă;
10. Returnează lista abonaților care revine pentru o lună indicată.

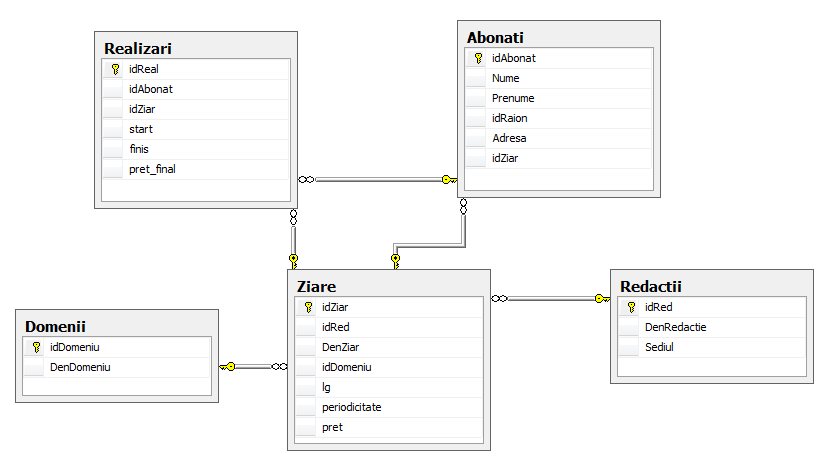
# **Proiectarea bazei de date**

## **Crearea modelului obiect**

Baza de date este intitulată ca fiind *„Abonamente”*. Este formată din 5 tabele pentru a oferi informații depline utilizatorului privind lucrul cu înregistrarea abonamentelor, abonaților și ziarele cu care se lucrează la înterprindere. Astfel, acestea sunt:

* *Realizări,* care conține date despre abonamente;
* *Abonați,* care conține date despre abonați;
* *Ziare,* conține date despre ziarele înregistrate;
* *Redacții,* conține date referitor la redacțiile care editează ziarele înregistrate;
* *Domenii,* este o tabelă auxiliară care oferă detalii despre domeniile ce se referă la ziarele înregistrate.

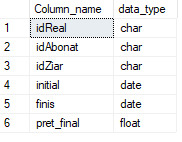
Între acestea se stabilesc următoarele relații:



Figură 1 Diagrama Entitate-Relație

## **Crearea și descrierea modelului relațional**

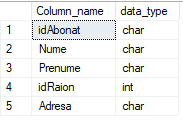
* Structura tabelului *Realizări* este următoareași conține date despre abonamentele înregistrate în baza de date și care vor urma să fie înregistrate. Respectiv, avem următoarele componente:



Figură 2 Structura tabelei Realizări

1. *idReal –* identificatorul realizării, care servește ca legătură pentru alte tabele;
2. *idAbonat –* identificatorul abonatului, care servește ca legătură pentru tabelul *Abonați;*
3. *idZiar -* identificatorul ziarului, care servește ca legătură pentru tabelul *Ziare;*
4. *initial –* este utilizat pentru specificarea datei din care începe abonamentul;
5. *finis -* este utilizat pentru specificarea datei în care se termină abonamentul;
6. *pret\_final –* este utilizat pentru a indica suma care revine aboantului pentru abonamentul respectiv;

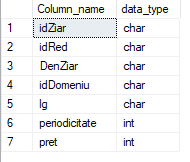
* Structura tabelului *Abonați* este următoareași conține date despre abonații actuali în baza de date și care urmează să fie înregistrați. Respectiv, avem următoarele componente:



Figură 3 Structura tabelei Abonați

1. *idAbonat –* identificatorul abonatului, care servește ca cheie primară;
2. *Nume –* numele abonatului;
3. *Prenume –* prenumele abonatului*;*
4. *idRaion –* identificatorul raionului;
5. *Adresa –* adresa abonatului unde se va livra ziarul;

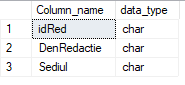
* Structura tabelului *Ziare* este următoareași conține date despre ziarele actuale în baza de date și care urmează să fie înregistrate. Respectiv, avem următoarele componente:



Figură 4 Structura tabelei Ziare

1. *idZiar –* identificatorul ziarului, care servește ca cheie primară;
2. *idRed –* identificatorul Redacției, care servește ca legătură pentru tabelul *Redacții;*
3. *DenZiar –* denumirea ziarului;
4. *idDomeniu -* identificatorul Domeniului, care servește ca legătură pentru tabelul *Domenii;*
5. *lg –* semnifică limba de editare a ziarului;
6. *periodicitate –* semnifică apariția ziarului (variante posibile o dată/pe lună sau o dată la 3 luni)
7. *pret –* suma pentru un ziar

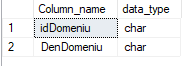
* Structura tabelului *Redacții* este următoareași conține date despre ziarele actuale în baza de date și care urmează să fie înregistrate. Respectiv, avem următoarele componente:



Figură 5 Structura tabelei Redacții

1. *idRed –* identificatorul Redacției, care servește ca legătură pentru tabelul *Redacții;*
2. *DenRedactie –* denumirea Redacției;
3. *Sediul –* unde poate fi găsită Redacția.

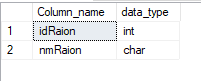
* Structura tabelului *Domenii* este următoareași conține date despre domeniile care cuprind ziarele înregistrate. Respectiv, avem următoarele componente:



Figură 6 Structura tabelei Domenii

1. *idDomeniu –* identificatorul Domeniului, care servește ca legătură pentru tabelul *Domenii;*
2. *DenDomeniu –* denumirea domeniului

* Structura tabelului *Raion* este următoarea și conține date despre raioanele Republicii Moldova. Respectiv, avem următoarele componente:



Figură 7 Structura tabelei Raion

1. *idRaion –* identificatorul raionului;
2. *nmRaion –* denumirea raionului.
   1. **Normalizarea și crearea bazei de date**

Baza de date conține date despre abonamentele la ziare. Respectiv , inițial tabela care oferă informații despre aceste date putea fi astfel:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| IdAbonat | Nume | Prenume | Raion | Adresa | Ziar | Redacție | Periodiciate | Start | Stop | Preț |

Acest lucru nu este optim pentru lucru, de aceea am realizat următoarele tabele. Astfel, tabelul *Raion* oferă detalii despre Raion *(fig. 7)* și legătură se realizează prin cheia primară *idRaion.* Tabelul *Ziare(fig. 4 )* conține detalii despre ziare, periodicitate, limba și legătură prin cheia *idRed* care face legătură cu tabelul *Redacții(fig. 5).* Apoi am creat un tabel aparte care oferă detalii despre domeniile care le cuprinde ziarul, deoarece acestea se repetă și pentru a crea ușurință pentru introducerea datelor creez *Domenii(fig. 6).* Pentru a face mai optimă găsirea abonamentelor înregistrate, am creat *Realizări(fig. 2),* care conține un identificator al realizării, identificatorul abonatului, momentul în care începe abonamentul și atunci când expiră abonamentul. De asemenea, prețul final pentru abonamentul încheiat și identificatorul ziarului la care se abonează utilizatorul.

# **Elaborarea aplicației**

## **3.1 Descrierea informației de intrare**

set dateformat dmy

use master

GO

if exists (select 'true' from sys.databases where name='Abonamente')

begin

alter database Abonamente set single\_user with rollback immediate

drop database Abonamente

end

GO

CREATE database Abonamente

GO

alter authorization on database::Abonamente to sa

use Abonamente

GO

CREATE TABLE Domenii

(idDomeniu char(2) PRIMARY KEY,

DenDomeniu char(50))

CREATE TABLE Redactii

(idRed char(5) PRIMARY KEY,

DenRedactie char(50),

Sediul char(50))

CREATE TABLE Ziare

(idZiar char(8) PRIMARY KEY,

idRed char(5),

DenZiar char(50),

idDomeniu char(2),

lg char(2),

periodicitate int,

pret int

)

CREATE TABLE Abonatii

(idAbonat char(7) PRIMARY KEY,

Nume char(50),

Prenume char(50),

idRaion int,

Adresa char(50))

CREATE TABLE Realizari

(idReal char(6) PRIMARY KEY,

idAbonat char(7),

idZiar char(8),

initial date,

finis date,

pret\_final float)

CREATE TABLE Raion

(idRaion int PRIMARY KEY,

nmRaion char(50))

GO

ALTER TABLE Ziare

ADD CONSTRAINT FK\_Ziare\_idRed

FOREIGN KEY (idRed) REFERENCES Redactii(idRed)

ALTER TABLE Ziare

ADD CONSTRAINT FK\_Ziare\_idDomeniu

FOREIGN KEY (idDomeniu) REFERENCES Domenii(idDomeniu)

ALTER TABLE Realizari

ADD CONSTRAINT FK\_Realizari\_idAbonat

FOREIGN KEY (idAbonat) REFERENCES Abonatii(idAbonat)

ALTER TABLE Realizari

ADD CONSTRAINT FK\_Realizari\_idZiar

FOREIGN KEY (idZiar) REFERENCES Ziare(idZiar)

ALTER TABLE Abonatii

ADD CONSTRAINT FK\_Abonatii\_idRaion

FOREIGN KEY (idRaion) REFERENCES Raion(idRaion)

GO

insert into Domenii(idDomeniu, DenDomeniu) values

('C1', 'Culinarie'),

('C2', 'Cultura'),

('E1', 'Educatie'),

('S1', 'Social'),

('S2', 'Sanatate'),

('G1', 'General')

insert into Redactii(idRed, DenRedactie, Sediul) values

('I1401', 'Adevarul', 'Chisinau'),

('I1402', 'Adverb', 'Cahul'),

('I1403', 'Curier','Cimislia'),

('I1404', 'Duminica', 'Falesti'),

('I1405','Luceafarul', 'Hancesti'),

('I1406', 'Timpul', 'Hancesti'),

('I1407', 'Vatra', 'Cantemir'),

('I1501', 'Вести', 'Balti'),

('I1502', 'Свет', 'Chisinau'),

('I1503', 'Маклер', 'Chisinau')

insert into Raion(idRaion, nmRaion) values

(1, 'Anenii-Noi'),

(2, 'Basarabeasca'),

(3, 'Briceni'),

(4, 'Cahul'),

(5, 'Cantemir'),

(6, 'Calarasi'),

(7, 'Causeni'),

(8, 'Cimislia'),

(9, 'Criuleni'),

(10, 'Donduseni'),

(11, 'Drochia'),

(12, 'Dubasari'),

(13, 'Edinet'),

(14, 'Falesti'),

(15, 'Floresti'),

(16, 'Glodeni'),

(17, 'Hincesti'),

(18, 'Ialoveni'),

(19, 'Leova'),

(20, 'Nisporeni'),

(21, 'Ocnita'),

(22, 'Orhei'),

(23, 'Rezina'),

(24, 'Riscani'),

(25, 'Singerei'),

(26, 'Soroca'),

(27, 'Straseni'),

(28, 'Soldanesti'),

(29, 'Stefan-Voda'),

(30, 'Taraclia'),

(31, 'Telenesti'),

(32, 'Ungheni'),

(33, 'Chisinau'),

(34, 'Balti')

insert into Ziare(idZiar, idRed, DenZiar, idDomeniu, lg, periodicitate, pret) values

('PM223050', 'I1501', 'Литературная газета', 'C2', 'RU', 3, 50.4),

('PM301255', 'I1501','Музыкальное обозрение', 'E1', 'RU', 1, 100),

('PM201821', 'I1406', 'Faclia', 'E1','MD', 1, 45),

('PM251020', 'I1503','Makler','G1', 'MD', 1, 15),

('PM130450', 'I1407', 'Ziarul de garda', 'S1', 'MD', 3, 20),

('PM219080', 'I1502', 'Vocea poporului', 'S1', 'MD', 1, 18),

('PM305047', 'I1502', 'Universul Sanatatii', 'S2', 'MD', 3, 45)

insert into Abonatii(idAbonat, Nume, Prenume, idRaion, Adresa) values

('A123456', 'Petrescu', 'Ion',3, 'str. Alba-Iulia 18'),

('B123456', 'Cartescu', 'Maria',8, 'str. Paris'),

('C123456', 'Sima','Felix',33, 'str. Columna 10'),

('D123456', 'Popescu', 'Gheorghe', 33, 'str. Florilor'),

('E123456', 'Trisca', 'Valeria', 33, 'str. Cantemir 14'),

('F123456', 'Tomulescu', 'Anatol', 25, 'str. Sangera'),

('G123456', 'Zurga', 'Ana', 30, 'str. Frumoasa 18'),

('H123456', 'Zvezdenko', 'Alexandru', 33, 'str. Toma Ciorba 10'),

('I123456', 'Cusnir', 'Dumitru', 30, 'str. Haiducului'),

('J123456', 'Enescu', 'Vitalina', 12, 'str. Armatei 2'),

('K123456', 'Dalcenco', 'Valeria', 14, 'str. Noul Neamt'),

('L123456', 'Turta', 'Andrei', 10, 'str. Mihai Eminescu'),

('M123456', 'Cristea', 'Cristina', 33, 'str. Veronica Micle'),

('N123456', 'Cecan','Valeria', 32, 'str. Eugen Doga 12'),

('O123456', 'Rusu', 'Alexandru', 30, 'str. Vlaicu Parcalab 10'),

('P123456', 'Lungu', 'Cristian', 31, 'str. Alecu Russo 5'),

('Q123456', 'Ravcenco', 'Maxim', 28, 'str. Armatei 20')

insert into Realizari(idReal, idAbonat, idZiar, initial, finis, pret\_final) values

('ABC123', 'A123456', 'PM223050', '10.01.2018', '10.04.2018', 50.4),

('DEF123', 'A123456', 'PM301255', '05.02.2018', '05.04.2018', 400),

('ABC456', 'B123456', 'PM251020', '01.01.2018', '01.06.2018', 75),

('ABC987', 'N123456', 'PM305047', '02.02.2018', '02.05.2018', 45),

('KLM123', 'D123456', 'PM201821', '01.01.2018', '01.08.2018', 315)

insert into Ziare(idZiar, idRed, DenZiar, idDomeniu, lg, periodicitate, pret) values

('PM240461', 'I1401', 'Adevarul', 'G1', 'MD', 1, 18),

('PM205682', 'I1402', 'Кулинария', 'C1', 'RU', 1, 53),

('PM213018', 'I1502','Рецепты', 'C1', 'RU', 3, 30),

('PM182525', 'I1404', 'Literatura si Arta', 'C2', 'MD', 1, 15.30)

## **Descrierea datelor de ieșire**

* Se afișează ziarele unui abonat indicat

select Ziare.DenZiar from ((Abonatii inner join Realizari on Abonatii.idAbonat = Realizari.idAbonat and Abonatii.idAbonat = 'A123456') inner join Ziare on Realizari.idZiar = Ziare.idZiar )

* Se afișează lista abonaților în ordine crescătoare după numele acestora

select \* from dbo.Abonatii order by Nume asc

* Se afișează datele abonatului în dependență de ziarul selectat

select Abonatii.Nume, Abonatii.Prenume, Abonatii.Adresa from ((Realizari inner join Ziare on Realizari.idZiar = Ziare.idZiar and Realizari.idZiar = 'PM223050') inner join Abonatii on Realizari.idAbonat = Abonatii.idAbonat)

* Se afișează numărul de ziare cu redacția indicată

select count( Ziare.idZiar) from Ziare where Ziare.idRed = 'I1501'

* Se afișează suma prețurilor al ziarelor editate de o anumită editură

select ROUND( SUM(Ziare.pret),2) from Ziare where Ziare.idRed='I1501'

* Se afișează toate ziarele în limba română

select \* from Ziare where Ziare.lg = 'MD'

* Se afișează toate ziarele în limba rusă

select \* from Ziare where Ziare.lg = 'RU'

* Se afișează abonamentele care corespund perioadei indicate

declare @find date = '01.01.2018'

select Abonatii.idAbonat, Abonatii.Nume, Abonatii.Prenume,Ziare.DenZiar, Realizari.initial , Realizari.finis ,Realizari.pret\_final

from Abonatii inner join Realizari on( Abonatii.idAbonat = Realizari.idAbonat) and month(@find) between MONTH(initial ) and MONTH(finis)

inner join Ziare on Realizari.idZiar = Ziare.idZiar

1. **Ghid de utilizare a aplicației**

**4.1 Produsul cu care este construită baza de date**

Baza de date este creată cu ajutorul limbajului T-SQL și SGBD-ului Microsoft ServerManagement Studio , iar aplicația este create cu ajutorul limbajului C# și aplicației Visual Studio.

## **Cerințe minime hardware și software pentru buna funcționare a aplicației**

* Procesor – 300 MHz
* Memorie operativă – 128 MB
* HDD – 50 MB
* Microsoft Server
* Versiunea Windows 7 și mai sus

## **Descrierea aplicației în baza sarcinilor individuale**

* **Forma inițitală care o vede utilizatorul. În partea de sus este plasat un meniu, ce oferă diverse posibilități. Imaginile indică ziarele actuale. În partea stângă este un meniu auxiliar, care permite vizualizarea ziarelor( imagini ale acestora ) conform domeniului.**

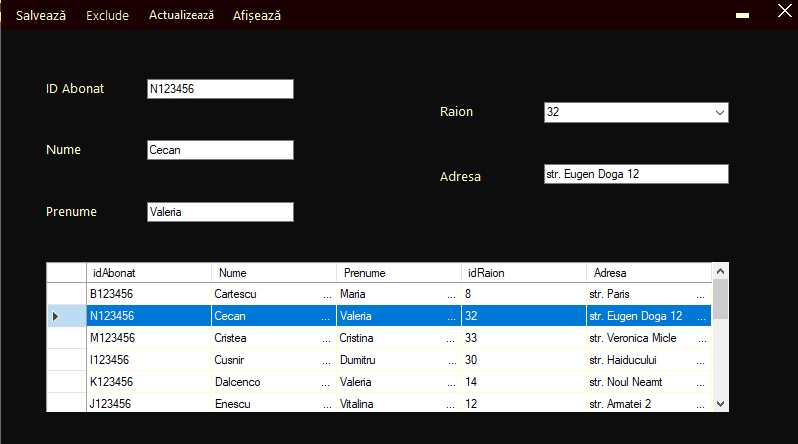


Figură 8 Forma inițială

* **La alegerea *Fișier – Adaugă – Abonat* se deschide următoarea formă , care permite înregistrarea, excluderea, actualizarea ( prin intermediul unui meniu) a abonaților.**

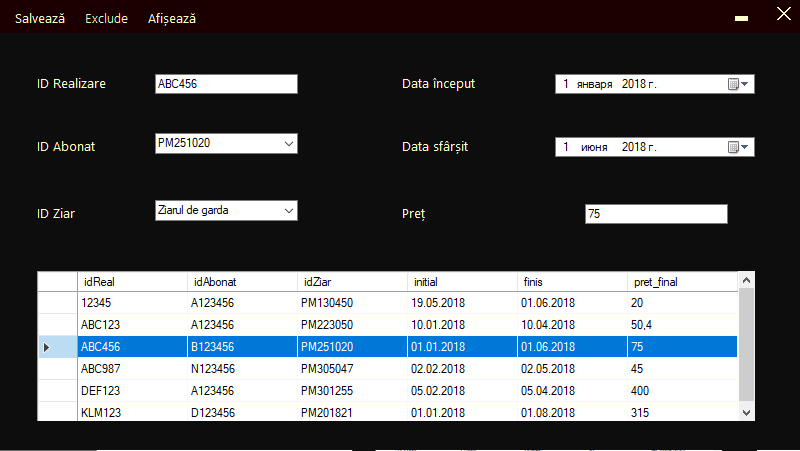


Figură 9 Meniu ( Adaugă)



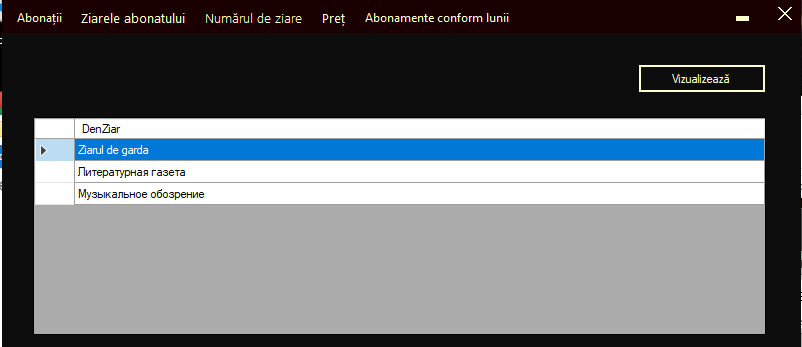
Figură 10 Forma 2

* **La alegerea *Fișier – Adaugă – Abonament* se deschide următoarea formă , care permite înregistrarea, excluderea, actualizarea ( prin intermediul unui meniu) a abonamentelor.**

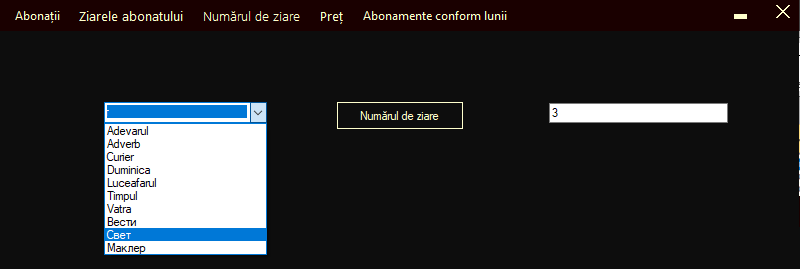


Figură 11 Forma 3

* **La alegerea *Fișier – Determină* se deschide următoarea formă , care permite următoarele opțiuni:**
* ***Ziarele abonatului* –** afișează ziarele abonatului selectat (fig. 12)
* ***Numărul de ziare* –** în baza redacției alese din lista propusă se determină numărul de ziare a redacției date (fig. 13)
* ***Preț* –** determină prețul total al ziarelor editate de o anumită editură din lista propusă (fig. 14)
* ***Abonamente conform lunii* –** permite utilizatorul să aleagă o lună și să vizualizeze care sunt abonamente ce conțin luna data (fig. 15)



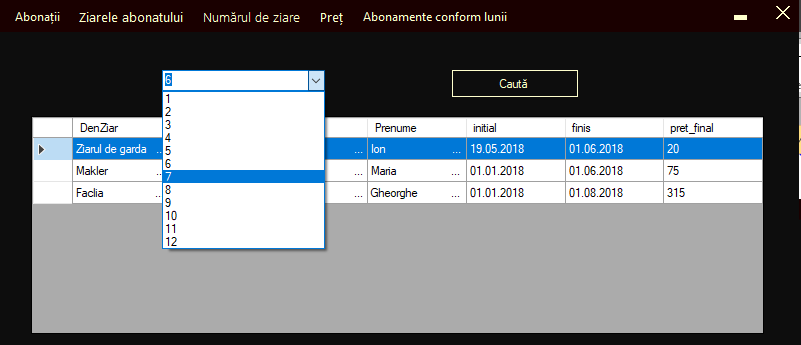
Figură 12 Forma 4 ( opțiunea Ziarele abonatului – date de ieșire)



Figură 13 Forma 4 (opțiunea Numărul de ziare )



Figură 14 Forma 4 (opțiunea Preț)



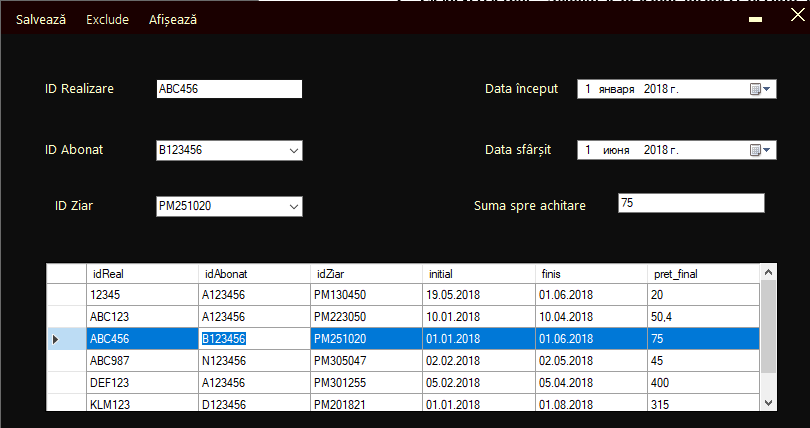
Figură 15 Forma 4 ( opțiunea Abonamente conform lunii)

* **La alegerea *Fișier – Ieșire* se închide aplicația**
* **La alegerea *Date - Abonați* se deschide forma ce permite vizualizarea abonaților curenți și permite manipularea datelor**



Figură 16 Meniu , opțiunea Date

* **La alegerea *Date – Abonamente* se deschide forma ce permite vizualizarea abonamentelor curente și permite manipularea datelor**



Figură 17 Forma 5

* **La alegerea *Date – Ziare și Redacții* se deschide forma ce permite vizualizarea ziarelor și redacțiilor curente și permite exportul listei ziarelor în Excel *(Meniu – Export)***



Figură 18 Forma 6

* ***MD*  -** afișează lista ziarelor editate în limba română
* ***RU*** – afișează lista ziarelor editate în limba rusă
* ***Ambele –*** afișeză lista tuturor ziarelor

# **Concluzie**

În urma realizării acestei practici am închegat cunoștințele legate de limbajele de programare SQL și C#. Practica dată a fost de folos, deoarece au fost realizate obiectivele generale a acesteia. Pe parcursul lucrării nu am întâlnit dificultăți. Am tins să elaborez o aplicație care va fi comodă pentru lucru, incluzând eventuale erori ce pot apărea în urma lansării aplicației.

Sunt de părerea că elaborarea aplicației date a fost un avantaj, datorită faptului că astfel de lucruri se cer pe piață și este o experiență destul de bună. Rezultatul acesteia poate fi atașată la CV.

Pe lângă aceasta a dus la deprinderea cu regimul de lucru și a fost realizată într-un mediu favorabil.

# **Bibliografie**

1. Gălățan Constantin „Curs de programare în Visual C# ”
2. <https://www.w3schools.com/sql/default.asp>

# **Anexe**

**Foma 1**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Ziare

{

public partial class Form1 : Form

{

int mov, movX, movY;

public void menuStrip1\_MouseDown(object sender, MouseEventArgs e)

{

mov = 1;

movX = e.X;

movY = e.Y;

}

public void menuStrip1\_MouseMove(object sender, MouseEventArgs e)

{

if (mov == 1)

{

this.SetDesktopLocation(MousePosition.X - movX, MousePosition.Y - movY);

}

}

public void menuStrip1\_MouseUp(object sender, MouseEventArgs e)

{

mov = 0;

}

public void Min\_Paint(object sender, PaintEventArgs e)

{

Graphics z = e.Graphics;

Color myColor = Color.FromArgb(255, 255, 204);

SolidBrush myBrush = new SolidBrush(myColor);

Pen pen = new Pen(Color.FromArgb(255, 255, 204));

z.DrawRectangle(pen, 7, 16, 12, 4);

z.FillRectangle(myBrush, 7, 16, 12, 4);

}

public void CloseMenu\_Paint(object sender, PaintEventArgs e)

{

Graphics z = e.Graphics;

Pen pen = new Pen(Color.FromArgb(255, 255, 204));

z.DrawLine(pen, 7, 7, 19, 19);

z.DrawLine(pen, 7, 19, 19, 7);

z.DrawLine(pen, 8, 7, 20, 19);

z.DrawLine(pen, 8, 19, 20, 7);

}

public void CloseMenu\_MouseClick(object sender, MouseEventArgs e)

{

Application.Exit();

}

public void Min\_MouseClick(object sender, MouseEventArgs e)

{

WindowState = FormWindowState.Minimized;

}

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

{

}

private void Culinarie\_MouseHover(object sender, EventArgs e)

{

Culinarie.BackColor = Color.FromArgb(26, 0, 0);

}

private void Culinarie\_MouseLeave(object sender, EventArgs e)

{

Culinarie.BackColor = Color.FromArgb(13, 13, 13);

}

private void Cultura\_MouseHover(object sender, EventArgs e)

{

Cultura.BackColor = Color.FromArgb(26, 0, 0);

}

private void Cultura\_MouseLeave(object sender, EventArgs e)

{

Cultura.BackColor = Color.FromArgb(13, 13, 13);

}

private void Educatie\_MouseHover(object sender, EventArgs e)

{

Educatie.BackColor = Color.FromArgb(26, 0, 0);

}

private void Educatie\_MouseLeave(object sender, EventArgs e)

{

Educatie.BackColor = Color.FromArgb(13, 13, 13);

}

private void Social\_MouseHover(object sender, EventArgs e)

{

Social.BackColor = Color.FromArgb(26, 0, 0);

}

private void Social\_MouseLeave(object sender, EventArgs e)

{

Social.BackColor = Color.FromArgb(13, 13, 13);

}

private void Sanatate\_MouseHover(object sender, EventArgs e)

{

Sanatate.BackColor = Color.FromArgb(26, 0, 0);

}

private void sleave(object sender, EventArgs e)

{

Sanatate.BackColor = Color.FromArgb(13, 13, 13);

}

private void General\_MouseHover(object sender, EventArgs e)

{

General.BackColor = Color.FromArgb(26, 0, 0);

}

private void General\_MouseLeave(object sender, EventArgs e)

{

General.BackColor = Color.FromArgb(13, 13, 13);

}

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

{

Form2 f = new Form2();

f.Show();

}

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

{

Form3 f = new Form3();

f.Show();

}

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

{

Application.Exit();

}

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

{

Form4 f4 = new Form4();

f4.Show();

}

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

{

Form4 f4 = new Form4();

f4.Show();

}

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

{

Form5 f5 = new Form5();

f5.Show();

}

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

{

Form6 f6 = new Form6();

f6.Show();

}

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

menuStrip1.ForeColor = Color.FromArgb(255, 255, 204);

}

}

}

**Forma 2**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace Ziare

{

public partial class Form2 : Form

{

SqlConnection conn = new SqlConnection(@"Data Source=.\SQLEXPRESS;Initial Catalog=Abonamente;Integrated Security=True;Connect Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWrite;MultiSubnetFailover=False");

int mov, movX, movY;

public Form2()

{

InitializeComponent();

}

private void CloseMenu\_Paint(object sender, PaintEventArgs e)

{

Graphics z = e.Graphics;

Pen pen = new Pen(Color.FromArgb(255, 255, 204));

z.DrawLine(pen, 7, 7, 19, 19);

z.DrawLine(pen, 7, 19, 19, 7);

z.DrawLine(pen, 8, 7, 20, 19);

z.DrawLine(pen, 8, 19, 20, 7);

}

private void MinMax\_Paint(object sender, PaintEventArgs e)

{

Graphics z = e.Graphics;

Color myColor = Color.FromArgb(255, 255, 204);

SolidBrush myBrush = new SolidBrush(myColor);

Pen pen = new Pen(Color.FromArgb(255, 255, 204));

z.DrawRectangle(pen, 7, 7, 12, 12);

z.FillRectangle(myBrush, 7, 7, 12, 12);

z.DrawRectangle(pen, 9, 7, 7, 10);

z.DrawRectangle(pen, 12, 12, 10, 10);

}

private void Min\_Paint(object sender, PaintEventArgs e)

{

Graphics z = e.Graphics;

Color myColor = Color.FromArgb(255, 255, 204);

SolidBrush myBrush = new SolidBrush(myColor);

Pen pen = new Pen(Color.FromArgb(255, 255, 204));

z.DrawRectangle(pen, 7, 16, 12, 4);

z.FillRectangle(myBrush, 7, 16, 12, 4);

}

private void CloseMenu\_MouseClick(object sender, MouseEventArgs e)

{

Close();

}

private void Form2\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "abonamenteDataSet1.Raion". При необходимости она может быть перемещена или удалена.

this.raionTableAdapter.Fill(this.abonamenteDataSet1.Raion);

}

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

{

conn.Open();

String query = "INSERT INTO dbo.Abonatii(idAbonat, Nume, Prenume, idRaion, Adresa) values('" + textBox1.Text + "','" + textBox2.Text + "','" + textBox3.Text + "','" + comboBox1.SelectedValue + "','" + textBox4.Text + "')";

SqlDataAdapter SDA = new SqlDataAdapter(query, conn);

SDA.SelectCommand.ExecuteNonQuery();

conn.Close();

Form7 f7 = new Form7();

f7.Show();

conn.Close();

}

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

{

conn.Open();

String query = "DELETE FROM dbo.Abonatii where idAbonat = '" + textBox1.Text + "'";

SqlDataAdapter SDA = new SqlDataAdapter(query, conn);

SDA.SelectCommand.ExecuteNonQuery();

dataGridView1.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.Fill;

conn.Close();

Form8 f8 = new Form8();

f8.Show();

}

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

{

conn.Open();

string query = "UPDATE dbo.Abonatii SET IDAbonat = '" + textBox1.Text + "' Nume = '" + textBox2.Text + "', Prenume = '" + textBox3.Text + "',idRaion = '" + comboBox1.Text + "',Adresa = '" + textBox4.Text + "'";

SqlDataAdapter SDA = new SqlDataAdapter(query, conn);

conn.Close();

}

private void dateToolStripMenuItem\_DoubleClick(object sender, EventArgs e)

{

}

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

{

conn.Open();

string query = "SELECT \* from dbo.Abonatii order by Nume asc";

SqlDataAdapter SDA = new SqlDataAdapter(query, conn);

DataTable dt = new DataTable();

SDA.Fill(dt);

dataGridView1.DataSource = dt;

dataGridView1.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.Fill;

conn.Close();

}

private void dataGridView1\_MouseClick(object sender, MouseEventArgs e)

{

textBox1.Text = dataGridView1.SelectedRows[0].Cells[0].Value.ToString();

textBox2.Text = dataGridView1.SelectedRows[0].Cells[1].Value.ToString();

textBox3.Text = dataGridView1.SelectedRows[0].Cells[2].Value.ToString();

comboBox1.Text = dataGridView1.SelectedRows[0].Cells[3].Value.ToString();

textBox4.Text = dataGridView1.SelectedRows[0].Cells[4].Value.ToString();

}

private void menuStrip1\_MouseUp(object sender, MouseEventArgs e)

{

mov = 0;

}

private void menuStrip1\_MouseDown(object sender, MouseEventArgs e)

{

mov = 1;

movX = e.X;

movY = e.Y;

}

private void menuStrip1\_MouseMove(object sender, MouseEventArgs e)

{

if (mov == 1)

{

this.SetDesktopLocation(MousePosition.X - movX, MousePosition.Y - movY);

}

}

private void Min\_MouseClick(object sender, MouseEventArgs e)

{

WindowState = FormWindowState.Minimized;

}

}

}

**Forma 3**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace Ziare

{

public partial class Form2 : Form

{

SqlConnection conn = new SqlConnection(@"Data Source=.\SQLEXPRESS;Initial Catalog=Abonamente;Integrated Security=True;Connect Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWrite;MultiSubnetFailover=False");

int mov, movX, movY;

public Form2()

{

InitializeComponent();

}

private void CloseMenu\_Paint(object sender, PaintEventArgs e)

{

Graphics z = e.Graphics;

Pen pen = new Pen(Color.FromArgb(255, 255, 204));

z.DrawLine(pen, 7, 7, 19, 19);

z.DrawLine(pen, 7, 19, 19, 7);

z.DrawLine(pen, 8, 7, 20, 19);

z.DrawLine(pen, 8, 19, 20, 7);

}

private void MinMax\_Paint(object sender, PaintEventArgs e)

{

Graphics z = e.Graphics;

Color myColor = Color.FromArgb(255, 255, 204);

SolidBrush myBrush = new SolidBrush(myColor);

Pen pen = new Pen(Color.FromArgb(255, 255, 204));

z.DrawRectangle(pen, 7, 7, 12, 12);

z.FillRectangle(myBrush, 7, 7, 12, 12);

z.DrawRectangle(pen, 9, 7, 7, 10);

z.DrawRectangle(pen, 12, 12, 10, 10);

}

private void Min\_Paint(object sender, PaintEventArgs e)

{

Graphics z = e.Graphics;

Color myColor = Color.FromArgb(255, 255, 204);

SolidBrush myBrush = new SolidBrush(myColor);

Pen pen = new Pen(Color.FromArgb(255, 255, 204));

z.DrawRectangle(pen, 7, 16, 12, 4);

z.FillRectangle(myBrush, 7, 16, 12, 4);

}

private void CloseMenu\_MouseClick(object sender, MouseEventArgs e)

{

Close();

}

private void Form2\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "abonamenteDataSet1.Raion". При необходимости она может быть перемещена или удалена.

this.raionTableAdapter.Fill(this.abonamenteDataSet1.Raion);

}

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

{

conn.Open();

String query = "INSERT INTO dbo.Abonatii(idAbonat, Nume, Prenume, idRaion, Adresa) values('" + textBox1.Text + "','" + textBox2.Text + "','" + textBox3.Text + "','" + comboBox1.SelectedValue + "','" + textBox4.Text + "')";

SqlDataAdapter SDA = new SqlDataAdapter(query, conn);

SDA.SelectCommand.ExecuteNonQuery();

conn.Close();

Form7 f7 = new Form7();

f7.Show();

conn.Close();

}

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

{

conn.Open();

String query = "DELETE FROM dbo.Abonatii where idAbonat = '" + textBox1.Text + "'";

SqlDataAdapter SDA = new SqlDataAdapter(query, conn);

SDA.SelectCommand.ExecuteNonQuery();

dataGridView1.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.Fill;

conn.Close();

Form8 f8 = new Form8();

f8.Show();

}

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

{

conn.Open();

string query = "UPDATE dbo.Abonatii SET IDAbonat = '" + textBox1.Text + "' Nume = '" + textBox2.Text + "', Prenume = '" + textBox3.Text + "',idRaion = '" + comboBox1.Text + "',Adresa = '" + textBox4.Text + "'";

SqlDataAdapter SDA = new SqlDataAdapter(query, conn);

conn.Close();

}

private void dateToolStripMenuItem\_DoubleClick(object sender, EventArgs e)

{

}

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

{

conn.Open();

string query = "SELECT \* from dbo.Abonatii order by Nume asc";

SqlDataAdapter SDA = new SqlDataAdapter(query, conn);

DataTable dt = new DataTable();

SDA.Fill(dt);

dataGridView1.DataSource = dt;

dataGridView1.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.Fill;

conn.Close();

}

private void dataGridView1\_MouseClick(object sender, MouseEventArgs e)

{

textBox1.Text = dataGridView1.SelectedRows[0].Cells[0].Value.ToString();

textBox2.Text = dataGridView1.SelectedRows[0].Cells[1].Value.ToString();

textBox3.Text = dataGridView1.SelectedRows[0].Cells[2].Value.ToString();

comboBox1.Text = dataGridView1.SelectedRows[0].Cells[3].Value.ToString();

textBox4.Text = dataGridView1.SelectedRows[0].Cells[4].Value.ToString();

}

private void menuStrip1\_MouseUp(object sender, MouseEventArgs e)

{

mov = 0;

}

private void menuStrip1\_MouseDown(object sender, MouseEventArgs e)

{

mov = 1;

movX = e.X;

movY = e.Y;

}

private void menuStrip1\_MouseMove(object sender, MouseEventArgs e)

{

if (mov == 1)

{

this.SetDesktopLocation(MousePosition.X - movX, MousePosition.Y - movY);

}

}

private void Min\_MouseClick(object sender, MouseEventArgs e)

{

WindowState = FormWindowState.Minimized;

}

}

}

**Forma 4**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace Ziare

{

public partial class Form2 : Form

{

SqlConnection conn = new SqlConnection(@"Data Source=.\SQLEXPRESS;Initial Catalog=Abonamente;Integrated Security=True;Connect Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWrite;MultiSubnetFailover=False");

int mov, movX, movY;

public Form2()

{

InitializeComponent();

}

private void CloseMenu\_Paint(object sender, PaintEventArgs e)

{

Graphics z = e.Graphics;

Pen pen = new Pen(Color.FromArgb(255, 255, 204));

z.DrawLine(pen, 7, 7, 19, 19);

z.DrawLine(pen, 7, 19, 19, 7);

z.DrawLine(pen, 8, 7, 20, 19);

z.DrawLine(pen, 8, 19, 20, 7);

}

private void MinMax\_Paint(object sender, PaintEventArgs e)

{

Graphics z = e.Graphics;

Color myColor = Color.FromArgb(255, 255, 204);

SolidBrush myBrush = new SolidBrush(myColor);

Pen pen = new Pen(Color.FromArgb(255, 255, 204));

z.DrawRectangle(pen, 7, 7, 12, 12);

z.FillRectangle(myBrush, 7, 7, 12, 12);

z.DrawRectangle(pen, 9, 7, 7, 10);

z.DrawRectangle(pen, 12, 12, 10, 10);

}

private void Min\_Paint(object sender, PaintEventArgs e)

{

Graphics z = e.Graphics;

Color myColor = Color.FromArgb(255, 255, 204);

SolidBrush myBrush = new SolidBrush(myColor);

Pen pen = new Pen(Color.FromArgb(255, 255, 204));

z.DrawRectangle(pen, 7, 16, 12, 4);

z.FillRectangle(myBrush, 7, 16, 12, 4);

}

private void CloseMenu\_MouseClick(object sender, MouseEventArgs e)

{

Close();

}

private void Form2\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "abonamenteDataSet1.Raion". При необходимости она может быть перемещена или удалена.

this.raionTableAdapter.Fill(this.abonamenteDataSet1.Raion);

}

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

{

conn.Open();

String query = "INSERT INTO dbo.Abonatii(idAbonat, Nume, Prenume, idRaion, Adresa) values('" + textBox1.Text + "','" + textBox2.Text + "','" + textBox3.Text + "','" + comboBox1.SelectedValue + "','" + textBox4.Text + "')";

SqlDataAdapter SDA = new SqlDataAdapter(query, conn);

SDA.SelectCommand.ExecuteNonQuery();

conn.Close();

Form7 f7 = new Form7();

f7.Show();

conn.Close();

}

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

{

conn.Open();

String query = "DELETE FROM dbo.Abonatii where idAbonat = '" + textBox1.Text + "'";

SqlDataAdapter SDA = new SqlDataAdapter(query, conn);

SDA.SelectCommand.ExecuteNonQuery();

dataGridView1.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.Fill;

conn.Close();

Form8 f8 = new Form8();

f8.Show();

}

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

{

conn.Open();

string query = "UPDATE dbo.Abonatii SET IDAbonat = '" + textBox1.Text + "' Nume = '" + textBox2.Text + "', Prenume = '" + textBox3.Text + "',idRaion = '" + comboBox1.Text + "',Adresa = '" + textBox4.Text + "'";

SqlDataAdapter SDA = new SqlDataAdapter(query, conn);

conn.Close();

}

private void dateToolStripMenuItem\_DoubleClick(object sender, EventArgs e)

{

}

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

{

conn.Open();

string query = "SELECT \* from dbo.Abonatii order by Nume asc";

SqlDataAdapter SDA = new SqlDataAdapter(query, conn);

DataTable dt = new DataTable();

SDA.Fill(dt);

dataGridView1.DataSource = dt;

dataGridView1.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.Fill;

conn.Close();

}

private void dataGridView1\_MouseClick(object sender, MouseEventArgs e)

{

textBox1.Text = dataGridView1.SelectedRows[0].Cells[0].Value.ToString();

textBox2.Text = dataGridView1.SelectedRows[0].Cells[1].Value.ToString();

textBox3.Text = dataGridView1.SelectedRows[0].Cells[2].Value.ToString();

comboBox1.Text = dataGridView1.SelectedRows[0].Cells[3].Value.ToString();

textBox4.Text = dataGridView1.SelectedRows[0].Cells[4].Value.ToString();

}

private void menuStrip1\_MouseUp(object sender, MouseEventArgs e)

{

mov = 0;

}

private void menuStrip1\_MouseDown(object sender, MouseEventArgs e)

{

mov = 1;

movX = e.X;

movY = e.Y;

}

private void menuStrip1\_MouseMove(object sender, MouseEventArgs e)

{

if (mov == 1)

{

this.SetDesktopLocation(MousePosition.X - movX, MousePosition.Y - movY);

}

}

private void Min\_MouseClick(object sender, MouseEventArgs e)

{

WindowState = FormWindowState.Minimized;

}

}

}

**Forma 5**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace Ziare

{

public partial class Form5 : Form

{

public Form5()

{

InitializeComponent();

}

Form1 f = new Form1();

int mov, movX, movY;

SqlConnection conn = new SqlConnection(@"Data Source=.\SQLEXPRESS;Initial Catalog=Abonamente;Integrated Security=True;Connect Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWrite;MultiSubnetFailover=False");

private void Form5\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "abonamenteDataSet7.Ziare". При необходимости она может быть перемещена или удалена.

this.ziareTableAdapter.Fill(this.abonamenteDataSet7.Ziare);

// TODO: данная строка кода позволяет загрузить данные в таблицу "abonamenteDataSet6.Abonatii". При необходимости она может быть перемещена или удалена.

this.abonatiiTableAdapter.Fill(this.abonamenteDataSet6.Abonatii);

}

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

{

conn.Open();

String query = "INSERT INTO dbo.Realizari(idReal, idAbonat, idZiar, initial, finis, pret\_final) values('" + textBox1.Text + "','" + abBox.SelectedValue + "','" + ziarBox.SelectedValue + "','" + dateTimePicker1.Value.Date.ToString() + "','" + dateTimePicker2.Value.Date.ToString() + "', '" + priceBox.Text + "')";

SqlDataAdapter SDA = new SqlDataAdapter(query, conn);

SDA.SelectCommand.ExecuteNonQuery();

conn.Close();

Form7 f7 = new Form7();

f7.Show();

}

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

{

conn.Open();

string query = "SELECT \* from dbo.Realizari";

SqlDataAdapter SDA = new SqlDataAdapter(query, conn);

DataTable dt = new DataTable();

SDA.Fill(dt);

dataGridView1.DataSource = dt;

dataGridView1.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.Fill;

conn.Close();

}

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

{

conn.Open();

String query = "DELETE FROM dbo.Realizari where idReal = '" + textBox1.Text + "'";

SqlDataAdapter SDA = new SqlDataAdapter(query, conn);

SDA.SelectCommand.ExecuteNonQuery();

dataGridView1.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.Fill;

conn.Close();

Form8 f8 = new Form8();

f8.Show();

}

private void dataGridView1\_Click(object sender, EventArgs e)

{

textBox1.Text = dataGridView1.SelectedRows[0].Cells[0].Value.ToString();

abBox.Text = dataGridView1.SelectedRows[0].Cells[1].Value.ToString();

ziarBox.Text = dataGridView1.SelectedRows[0].Cells[2].Value.ToString();

dateTimePicker1.Text = dataGridView1.SelectedRows[0].Cells[3].Value.ToString();

dateTimePicker2.Text = dataGridView1.SelectedRows[0].Cells[4].Value.ToString();

priceBox.Text = dataGridView1.SelectedRows[0].Cells[5].Value.ToString();

}

private void menuStrip1\_MouseUp(object sender, MouseEventArgs e)

{

mov = 0;

}

private void menuStrip1\_MouseDown(object sender, MouseEventArgs e)

{

mov = 1;

movX = e.X;

movY = e.Y;

}

private void menuStrip1\_MouseMove(object sender, MouseEventArgs e)

{

if (mov == 1)

{

this.SetDesktopLocation(MousePosition.X - movX, MousePosition.Y - movY);

}

}

private void CloseMenu\_Paint(object sender, PaintEventArgs e)

{

f.CloseMenu\_Paint(sender, e);

}

private void Min\_Paint(object sender, PaintEventArgs e)

{

f.Min\_Paint(sender, e);

}

}

}

**Forma 6**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace Ziare

{

public partial class Form6 : Form

{

public Form6()

{

InitializeComponent();

}

Form1 f = new Form1();

SqlConnection conn = new SqlConnection(@"Data Source=.\SQLEXPRESS;Initial Catalog=Abonamente;Integrated Security=True;Connect Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWrite;MultiSubnetFailover=False");

int mov, movX, movY;

public void fill()

{

conn.Open();

string query3 = "SELECT \* from dbo.Ziare ";

SqlDataAdapter SDA3 = new SqlDataAdapter(query3, conn);

DataTable dt3 = new DataTable();

SDA3.Fill(dt3);

dataGridView1.DataSource = dt3;

dataGridView1.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.Fill;

conn.Close();

}

private void Form6\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "abonamenteDataSet8.Redactii". При необходимости она может быть перемещена или удалена.

this.redactiiTableAdapter.Fill(this.abonamenteDataSet8.Redactii);

}

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

{

Close();

}

private void Min\_Paint(object sender, PaintEventArgs e)

{

f.Min\_Paint(sender, e);

}

private void CloseMenu\_Paint(object sender, PaintEventArgs e)

{

f.CloseMenu\_Paint(sender, e);

}

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

{

WindowState = FormWindowState.Minimized;

}

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

{

panel1.Visible = true;

fill();

}

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

{

conn.Open();

string query1 = "SELECT \* from Ziare where Ziare.lg = 'MD'";

SqlDataAdapter SDA1 = new SqlDataAdapter(query1, conn);

DataTable dt1 = new DataTable();

SDA1.Fill(dt1);

dataGridView1.DataSource = dt1;

dataGridView1.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.Fill;

conn.Close();

}

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

{

conn.Open();

string query2 = "SELECT \* from Ziare where Ziare.lg = 'RU'";

SqlDataAdapter SDA2 = new SqlDataAdapter(query2, conn);

DataTable dt2 = new DataTable();

SDA2.Fill(dt2);

dataGridView1.DataSource = dt2;

dataGridView1.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.Fill;

conn.Close();

}

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

{

fill();

}

private void menuStrip1\_MouseUp(object sender, MouseEventArgs e)

{

mov = 0;

}

private void menuStrip1\_MouseMove(object sender, MouseEventArgs e)

{

if (mov == 1)

{

this.SetDesktopLocation(MousePosition.X - movX, MousePosition.Y - movY);

}

}

private void menuStrip1\_MouseDown(object sender, MouseEventArgs e)

{

mov = 1;

movX = e.X;

movY = e.Y;

}

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

{

panel1.Visible = false;

conn.Open();

string query3 = "SELECT \* from dbo.Redactii ";

SqlDataAdapter SDA3 = new SqlDataAdapter(query3, conn);

DataTable dt3 = new DataTable();

SDA3.Fill(dt3);

dataGridView1.DataSource = dt3;

dataGridView1.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.Fill;

conn.Close();

}

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

{

Microsoft.Office.Interop.Excel.Application app = new Microsoft.Office.Interop.Excel.Application();

Microsoft.Office.Interop.Excel.Workbook workbook = app.Workbooks.Add(Type.Missing);

Microsoft.Office.Interop.Excel.Worksheet worksheet = null;

worksheet = workbook.Sheets["Лист1"];

worksheet = workbook.ActiveSheet;

worksheet.Name = "Detalii despre ziare";

for (int i = 1; i < dataGridView1.Columns.Count + 1; i++)

{

worksheet.Cells[1, i] = dataGridView1.Columns[i - 1].HeaderText;

}

for (int i = 0; i < dataGridView1.Rows.Count; i++)

{

for (int j = 0; j < dataGridView1.Columns.Count; j++)

{

worksheet.Cells[i + 2, j + 1] = dataGridView1.Rows[i].Cells[j].Value.ToString();

}

}

var saveFileDialog = new SaveFileDialog();

saveFileDialog.FileName = "output";

saveFileDialog.DefaultExt = ".xlsx";

if (saveFileDialog.ShowDialog() == DialogResult.OK)

{

workbook.SaveAs(saveFileDialog.FileName, Type.Missing, Type.Missing, Type.Missing, Type.Missing, Type.Missing, Microsoft.Office.Interop.Excel.XlSaveAsAccessMode.xlExclusive, Type.Missing, Type.Missing, Type.Missing, Type.Missing, Type.Missing);

}

app.Quit();

}

}

}

**Forma 7**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Ziare

{

public partial class Form7 : Form

{

public Form7()

{

InitializeComponent();

}

Form1 f1 = new Form1();

private void CloseMenu\_Paint(object sender, PaintEventArgs e)

{

f1.CloseMenu\_Paint(sender, e);

}

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

{

Close();

}

}

}

**Forma 8**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Ziare

{

public partial class Form8 : Form

{

public Form8()

{

InitializeComponent();

}

Form1 f1 = new Form1();

private void CloseMenu\_Paint(object sender, PaintEventArgs e)

{

f1.CloseMenu\_Paint(sender, e);

}

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

{

Close();

}

}

}