

Rezervare online pentru biletele la cinema



Documentație proiect – Ingineria Programării

Studenti:

Furcoi G.T. Tase-Isidor, grupa 1310A

Mărmureanu L. Elena-Adeline, grupa 1310A

Petrescu G. Ambra-Costina, grupa 1310A

Pintilie V. Vasile-Emilian, grupa 1310A

*Facultatea de Automatică și Calculatoare, Iași
Anul universitar 2021-2022*

Cuprins

I. Documentația SRS

1. Introducere

1.1 Scopul.....	1
1.2 Convențiile documentului	1
1.3 Publicul vizat	1
1.4 Domeniul de aplicare.....	1
1.5 Referințe.....	1

2. Descriere generală

2.1 Perspectiva produsului	1
2.2 Funcțiile produsului.....	2
2.3 Clase de utilizatori și caracteristici.....	2
2.4 Mediul de operare.....	2
2.5 Constrângeri de proiectare și punere în aplicare.....	2
2.6 Documentația utilizatorului	2
2.7 Ipoteze și dependențe	2

3. Cerințe privind interfața externă

3.1 Interfața utilizatorului.....	3
3.2 Interfața hardware	3
3.3 Interfața software.....	3
3.4 Interfața de comunicații.....	3

4. Caracteristici de sistem.....3

5. Alte cerințe nefuncționale

5.1 Cerințe de performanță.....	3
5.2 Cerințe de securitate	4
5. Atributele calității software-ului.....	4
5.5 Reguli de afaceri	4

II. Diagrame UML.....5

III. Modul de utilizare a proiectului.....7

IV. Capturi de ecran cu programul în execuție.....8

V. Anexă cod semnificativ.....12

I. Documentația SRS

1. Introducere

1.1 Scop

Scopul acestui proiect este dezvoltarea unei aplicații care permite utilizatorilor rezervarea online de bilete la cinema. Astfel, dacă rezervările realizate personal prin deplasarea la cinema cu ceva timp înainte de începerea filmului necesită un consum de timp și energie, rezervările online ne oferă un confort sporit, prin simplul fapt că presupun doar folosirea unui calculator cu acces la internet, lucru ce este destul de comun în momentul de față. Aplicația oferă facilitatea de a păstra biletul sau de a modifica și anula rezervarea respectivă.

1.2 Convenții de documente

Acest document urmează formatul standard IEEE pentru dezvoltare software. Standardele urmăresc să definească un format ușor de citit, rapid și cu informațiile importante, cât și detaliile din proiectul nostru.

1.3 Publicul vizat

Aplicația este destinată tuturor persoanelor ce doresc să vizioneze un film și vor să aibă siguranța că vor găsi un loc fix acolo unde doresc să-l vizioneze fără a se deplasa fizic la cinema, acest lucru fiind posibil prin rezervarea online a билетelor.

1.4 Domeniul de aplicare

Aplicația de rezervare online pentru bilete la cinema reprezintă un beneficiu deoarece ajută la păstrarea datelor, precum a clienților dar și a filmelor disponibile pe care le avem, și ținerea evidenței acestora. În plus, aceasta poate fi pusă la dispoziția angajaților, dar și a altor persoane din diferite departamente. De aceea, există două tipuri de utilizatori: un utilizator simplu, ce dorește să vizualizeze site-ul și posibil să își facă o rezervare și adminul care este responsabil cu managementul datelor. Acesta trebuie să poată vizualiza, insera, edita și șterge datele din orice tabel al bazei de date.

1.5 Referințe

Ca și referințe am utilizat materia de la cursurile și laboratoarele materiei de Ingineria Programării.

2. Descriere generală

2.1 Perspectiva produsului

Aplicația dezvoltată de noi nu este unică pe piață dar îmbină niște elemente ce ușurează semnificativ munca utilizatorului.

Sistemul trebuie să permită căutarea și vizualizarea informațiilor într-un mod eficient, în timp cât mai rapid.

2.2 Funcțiile produsului

Funcțiile pe care aplicația trebuie să le îndeplinească sunt:

- Aplicația trebuie să ofere posibilitatea de a stoca datele introduce în formulare;
- Aplicația trebuie să verifice dacă fiecare informație introdusă este corectă;
- Aplicația poate oferi utilizatorului posibilitatea de a rezerva un bilet la cinema sau a anula o rezervare prin accesarea diferitelor formulare și apăsarea diferitelor butoane.
- Aplicația să fie stabilă: să asigure coerența datelor și să nu genereze întreruperi cauzate de potențiale erori de proiectare.

2.3 Clase și caracteristici ale utilizatorilor

Aplicația noastră este concepută pentru a fi ușor de utilizat, astfel încât orice clasă de utilizatori care știe să utilizeze minimal programul Microsoft Visual Studio ar trebui să se descurce să realizeze rezervarea biletelor pentru cinema. De aceea, interfețele sunt destul de simple, descriptive și ușor de navigat.

2.4 Mediul de operare

Proiectul de față este o aplicație dezvoltată folosind tehnologii Microsoft dar și a unei baze de date. Sistemul a fost dezvoltat în mediul de programare Microsoft Visual Studio, iar limbajul de programare folosit este C#. Baza de date a sistemului a fost realizată folosind Microsoft SQL Server. Deoarece rularea programului depinde de IDE-ul Microsoft Visual Studio, IDE disponibil pe majoritatea sistemelor de operare, atunci aplicația de rezervare online pentru biletele de cinema va putea rula pe toate sistemele de operare pe care este disponibil Microsoft Visual Studio.

2.5 Constrângeri de proiectare și implementare

Deoarece aplicația realizată este proiectată și implementată într-un singur semestru ca proiect pentru disciplina Ingineria Programării, este posibil ca doar timpul să fie cel mai limitativ factor în acest sens al ciclului de dezvoltare.

2.6 Documentația utilizatorului

Utilizatorul va avea în fiecare interfață un buton de help care îl va ajuta pe acesta să înțeleagă funcționalitatea aplicației.

2.7 Ipoteze și dependențe

Interfața este dependentă de baza de date cu care s-a realizat conexiunea, iar dacă aceasta nu mai funcționează ceva timp, atunci aplicația nu mai poate fi utilizată.

3. Cerințe de interfață externă

3.1 Interfața utilizatorului

Aplicația se deschide cu interfața principală în care apare o imagine de introducere unde se observă încărcarea programului prin o bară verde de progres. După încărcare ,utilizatorul va fi redirectionat în interfața de logare unde i se va cere să introducă userid și parola.Dacă datele au fost introduse în mod corect se va permite accesul la interfața home menu unde se va permite navigarea back and forth (înainte și înapoi) între restul interfețelor și interfața home menu prin apăsarea diferitelor butoane ce au nume sugestive corespunzând cu numele interfețelor unde se pot introduce de noi informații(cum ar fi interfețele AddMovie și AddUser) , precum și vizualizarea și actualizarea informațiilor (cum ar fi interfețele ViewMovies și ViewUsers) cât și a efectuării rezervarilor de bilete sau anularea acestora(cum ar fi interfețele Tickets și Cancellation) .

3.2 Interfața hardware

Interfața hardware este una simplă deoarece aplicația necesită doar un mouse și o tastatură pentru ca utilizatorul să își creeze un cont la noi în aplicație și apoi să își poată realiza rezervarea biletului la filmul pe care dorște să îl vizioneze, bineînțeles pe rândul și locul dorit,cât și anularea rezervării.

3.3 Interfața software

Aplicația poate rula pe orice versiune de Microsoft Visual Studio. Codul aplicației este scris în limbajul de programare C#.

3.4 Interfața de comunicații

Aplicația noastră necesită doar folosirea unui calculator cu un sistem de operare ce poate rula programul Microsoft Visual Studio.

4. Caracteristici de sistem

Pentru ca aplicația să poată fi utilizată, trebuie să funcționeze conexiunea cu baza de date. Fără baza de date, interfața nu funcționează, ceea ce înseamnă că sistemul nu poate fi folosit.

5. Alte cerințe nefuncționale

5.1 Cerințe de performanță

Aplicația poate rula pe orice sistem de operare. Se poate rula aplicația chiar dacă este pe un sistem de operare mai vechi precum Windows 7 sau pe un sistem de operare nou precum Windows 11. De asemenea se poate rula și pe sistemul de operare Linux, chiar dacă poate nu este un sistem de operare foarte bine cunoscut de toată lumea. Pentru a rula aplicația, calculatorul pe care îl folosim nu necesită să aibă o placă grafică sau memorie RAM mai mare de 2GB.

5.2 Cerințe de securitate

Această aplicație nu va aduna nicio altă informație privată de la clienții care doresc să se logheze sau de la utilizatorii deja înregistrați, în afară de id-ul de utilizator și parola aleasă de ei.

5.3 Atribute de calitate a software-ului

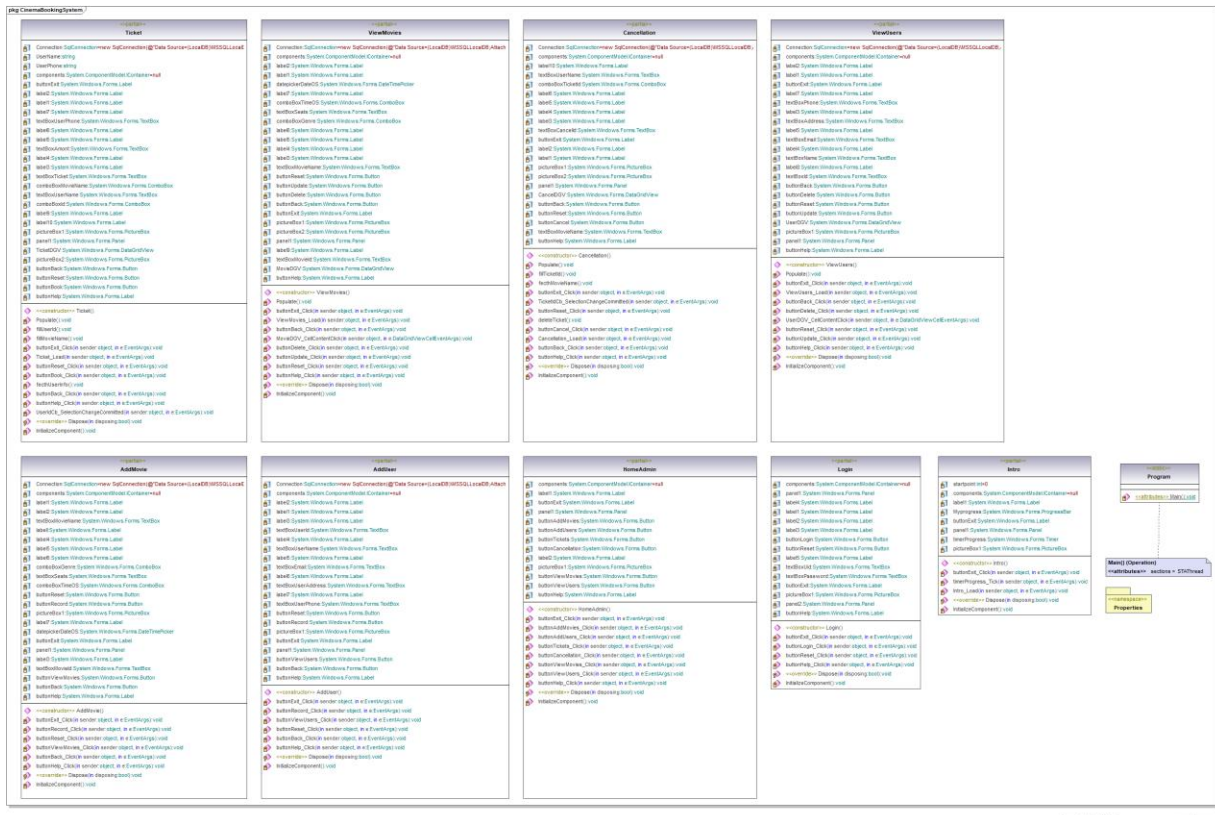
Aplicația este ușor de utilizat și oferă o interfață prietenoasă. Nu are cerințe soft foarte mari.

5.4 Reguli de afaceri

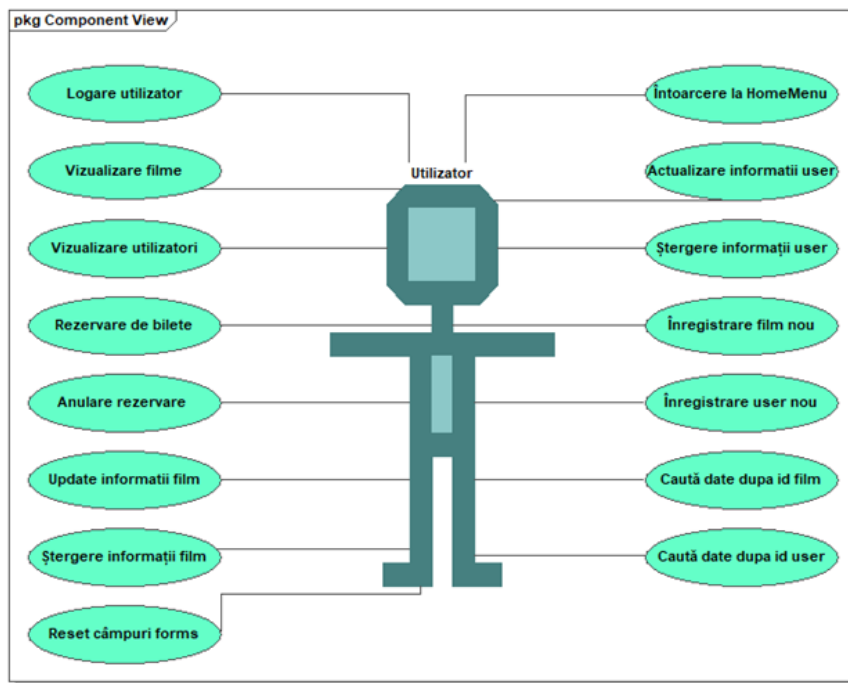
Echipa noastră de dezvoltare a acestei aplicații a urmărit să respecte toate codurile de conduită stabilite de către Universitatea Tehnică “Gheorghe Asachi” din Iași.

II. Diagramme UML

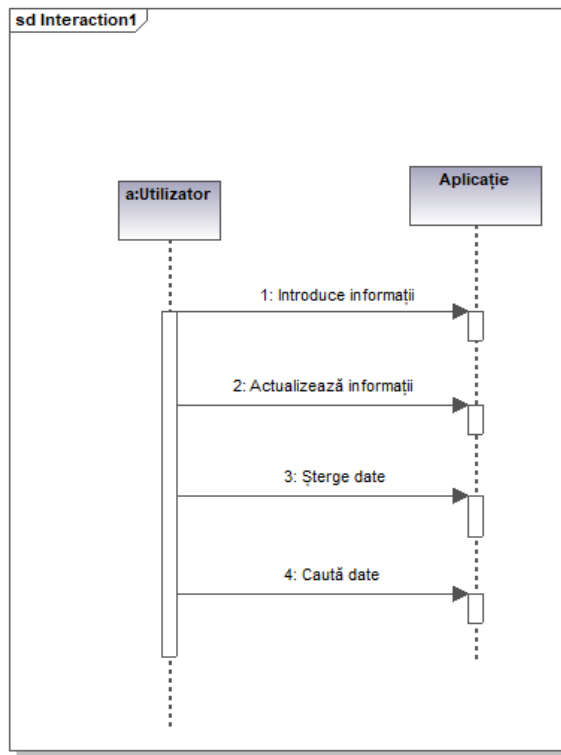
➤ Diagrama de clase



► Diagrama cazuri de utilizare



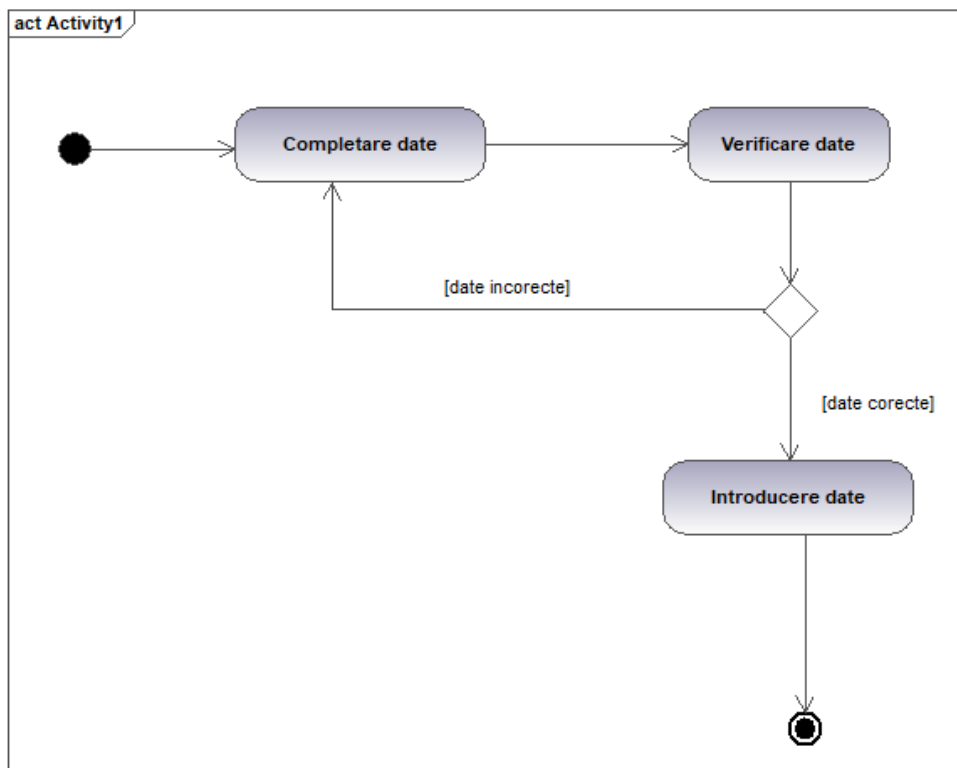
➤ Diagrama de secvențe



Generated by UModel

www.altova.com

➤ Diagrama de activități



Generated by UModel

www.altova.com

III. Modul de utilizare a proiectului

Acest proiect reprezintă o aplicație prin care se pot crea rezervări de bilete la cinema .Aceasta se ocupă cu gestiunea bazei de date folosită pentru a stoca informații cu privire la : date despre utilizator și filme , rezervări efectuate cât și anulări ale acestora.

Diferitele butoane din interfața de Home Menu ne pot duce pe diferite formulare în care putem înregistra , actualiza sau șterge folosind diferite butoane cu nume sugestive.

Atunci când dorim să vizualizăm filmele sau utilizatorii din baza de date cât și informații despre aceștia vom putea accesa formulare din interfețele View Movies sau View Users.

Sus, în partea stângă se află butonul de help cu același nume ,care oferă specificații generale asupra interfețelor și cu privire la modul de utilizare a aplicației.

Pentru închiderea ferestrei se apasă butonul “X”.

IV. Capturi de ecran cu programul în execuție

Interfața principală



Interfața de logare

HelpX

Cinema Booking System

It's showtime!

Cinema Booking System

Uid

Password

LoginReset

Interfața Home Menu

HelpX

Cinema Booking System

Home Menu

View Movies

Add Movies

Tickets

View Users


Add Users

Cancellation


CINEMA

Interfața de vizualizare/actualizare a informațiilor legate de filmele din baza de date (View Movies)

Help



Cinema Booking System
View Scheduled Movies



Movie Id

Movie Name

Genre

TimeOfShowing

DateOfShowing

Seats

Update

Delete


Reset

Back

	MovieId	MovieName	Genre	TimeOfShowing	DateOfShowing	Seats
▶	14	Movie	Horror	14pm	02 July 2022	45
	46	DrStrange	Fantasy	18pm	02/07/2022...	78
*						

Interfața de adăugare a informațiilor unui nou film in baza de date(Add Movies)

Help



Cinema Booking System
Record new movie

Movie Id

Movie Name

Genre

TimeOfShowing

DateOfShowing


Seats

Record

Reset

View Movies

Back



Interfața de vizualizare/actualizare a informațiilor legate de utilizatorii din baza de date (View Users)

Help


X

Cinema Booking System
View Users

Id

Email

Phone



Name

Address

Update

Delete

Reset

Back

Interfața de adăugare a informațiilor unui nou utilizator în baza de date (Add Users)

Help

X

Cinema Booking System
Record new user


User Id

User Name

User Email

User Address

User Phone



Record

Reset


View Users

Back

Interfața de adăugare rezervare a biletelor de film(Tickets)


Help

X



Cinema Booking System

Ticket Booking



Ticket Id

Movie Name

Movie

User Id

14

User Name

User Phone

Amount

Book

Reset

Back

Bookings


	TicketId	MovieName	UserId	UserName	UserPhone	Amount
▶	12	Cocoo	2	Elena	0798745632	2
	14	DrStrange	31	Mihai	0798563214	2
*						

< >

Interfața de anulare a rezervarii de bilete de film(Cancellation)


Help

X



Cinema Booking System

Cancel booking



Cancel Id

Ticket Id

12

Movie Name

User Name

Cancel

Reset

Back

Cancellations List

	CancelId	TicketId	MovieName	UserName
▶	45	21	DrStrange ...	Mihai
	47	13	Percy Jac...	Mihai
	122	12	Cocoo ...	Elena
*				

V. Anexă cod semnificativ

- Realizarea conexiunii la baza de date

```
/// <summary>
    ///realizam o noua conexiune la baza de date sql creata
    CinemaDb
    /// <summary>
    SqlConnection Connection = new SqlConnection(@"Data
    Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\User\Documents\
    CinemaDb.mdf;Integrated Security=True;Connect Timeout=30");
```

- Înregistrarea in baza de date , in exemplul de mai jos a unui film

```
private void buttonRecord_Click(object sender, EventArgs e)
{
    /// <summary>
    ///in aceasta metoda ni se permite prin apasarea
    butonului record
    ///sa inregistram un nou film si informatii despre
    acesta in baza de date
    /// <summary>

    /// <summary>
    ///in prim pas verificam daca avem campuri de informatii
    goale
    /// <summary>
    if ( textBoxMovieId.Text == "" || textBoxMovieName.Text
    == "" || comboBoxGenre.SelectedItem.ToString() == "" ||
    comboBoxTimeOS.SelectedItem.ToString() == "" ||
    datepickerDateOS.Text == "" || textBoxSeats.Text == "")
    {
        /// <summary>
        ///in cazul in care avem campuri de informatii
        goale/necompletate va aparea urmatorul mesaj ca avertizare catre
        utilizator

        /// <summary>
        MessageBox.Show("Missing information!Please complete
        the fields.");
    }
    else
    {
        /// <summary>
        ///error handling
        /// <summary>
```

```

try
{
    /// <summary>
    ///daca am completat informatiile corect in
    campurile interfetei se va crea o conexiune la baza de date
    ///ce ne va permite sa inseram in tabela
    MovieTbl date noi prin comanda ce o vom stoca in variabila cmd
    ///ca mai apoi sa fie executata prin apelul
    functiei ExecuteNonQuery
    /// <summary>
    Connection.Open();
    string query = "insert into MovieTbl values(" +
textBoxMovieId.Text + ", '" + textBoxMovieName.Text + "', '" +
comboBoxGenre.SelectedItem.ToString() + "', '" +
comboBoxTimeOS.SelectedItem.ToString() + "', '" +
datepickerDateOS.Value.ToString() + "', '" + textBoxSeats.Text +
"')";

    SqlCommand cmd = new SqlCommand(query,
Connection);

    cmd.ExecuteNonQuery();

    /// <summary>
    ///dupa inserarea noilor date in tabel urmatorul
    mesaj va aparea intr-un MessageBox
    ///ca mai apoi conexiunea cu baze de date sa fie
    inchisa prin apelarea functiei Close
    /// <summary>
    MessageBox.Show("Movie Recorded Successfully");
    Connection.Close();
}
catch (Exception Ex)
{
    /// <summary>
    ///in cazul in care va afea loc o exceptie va
    aparea un mesaj de eroare generat de Ex.Message ce explica motivul
    acesteia

    /// <summary>
    MessageBox.Show(Ex.Message);
}
}
}

```

- Vizualizarea datelor dintr-un tabel din baza de date sub forma unui tabel, in exemplul de mai jos afisam datele din tabela MoviesTbl

```
private void Populate()
{
    /// <summary>
    ///prin aceasta metoda se popoleaza data grid view-ul cu
    datele din tabela de interes MovieTbl , modificate sau nu
    /// <summary>
    Connection.Open();
    string query = "select * from MovieTbl";
    SqlDataAdapter sda = new SqlDataAdapter(query, Connection);
    SqlCommandBuilder builder = new SqlCommandBuilder(sda);
    var dataset = new DataSet();
    sda.Fill(dataset);
    MovieDGV.DataSource = dataset.Tables[0];
    Connection.Close();
}
```

- Ștergerea din baza de date , in exemplul de mai jos a unui film

```
private void buttonDelete_Click(object sender, EventArgs e)
{
    /// <summary>
    ///in aceasta metoda ni se permite prin apasarea butonului
    Delete sa stergem informatiile unui
    ///film din tabel introducand id-ul filmului in campul MovieId
    /// <summary>

    /// <summary>
    ///in prim pas verificam daca avem campuri de informatii goale
    /// <summary>
    if (textBoxMovieId.Text == "")
    {
        /// <summary>
        ///in cazul in care avem campuri de informatii
        goale/necompletat va aparea urmatorul mesaj ca avertizare catre utilizator
        /// <summary>
        MessageBox.Show("Enter The Movie to Delete");
    }
    else
    {
        /// <summary>
        ///error handling
        /// <summary>
        try
        {
            /// <summary>
        }
    }
}
```



```

        ///daca am completat corect informatiile in campurile
interfetei se va crea o conexiune la baza de date
        ///ce ne va permite sa inseram in tabela MovieTbl date
noi prin comanda ce o vom stoca in variabila cmd
        ///ca mai apoi sa fie executata prin apelul functiei
ExecuteNonQuery
        /// <summary>
Connection.Open();
string query = "delete from MovieTbl where MovieId=" +
textBoxMovieId.Text + ";";
SqlCommand cmd = new SqlCommand(query, Connection);
cmd.ExecuteNonQuery();

        /// <summary>
        ///dupa inserarea noilor date in tabel urmatorul mesaj
va aparea intr-un MessageBox
        ///ca mai apoi conexiunea cu baze de date sa fie
inchisa prin apelarea functiei Close
        /// <summary>
MessageBox.Show("Movie Deleted Successfully");
Connection.Close();
Populate();
    }
    catch (Exception Ex)
    {
        /// <summary>
        ///in cazul in care va afea loc o exceptie va aparea
un mesaj de eroare generat de Ex.Message ce explica motivul acesteia
        /// <summary>
MessageBox.Show(Ex.Message);
    }
}
}

```