-secțiunea utilitar-

Aplicatie creata de: **Vlad-Nicolae Badii** – **Elev cls a XII-a A**

**Colegiul Național B. P. Hasdeu Buzău**

**mai 2021**

**Cuprins**

[1. Prezentarea generală a temei 3](#_Toc73135261)

[1.1. Avantaje utilizare AllinOneTaxiApp 4](#_Toc73135262)

[2. Realizarea aplicației 5](#_Toc73135263)

[2.1. Scurtă descriere a programării orientate pe obiecte 8](#_Toc73135264)

[2.2. Scurtă prezentarea a serverului de baze de date MSSql Server 10](#_Toc73135265)

[2.3. Descrierea interfeței 12](#_Toc73135266)

[3. Bibliografie 21](#_Toc73135267)

# Prezentarea generală a temei

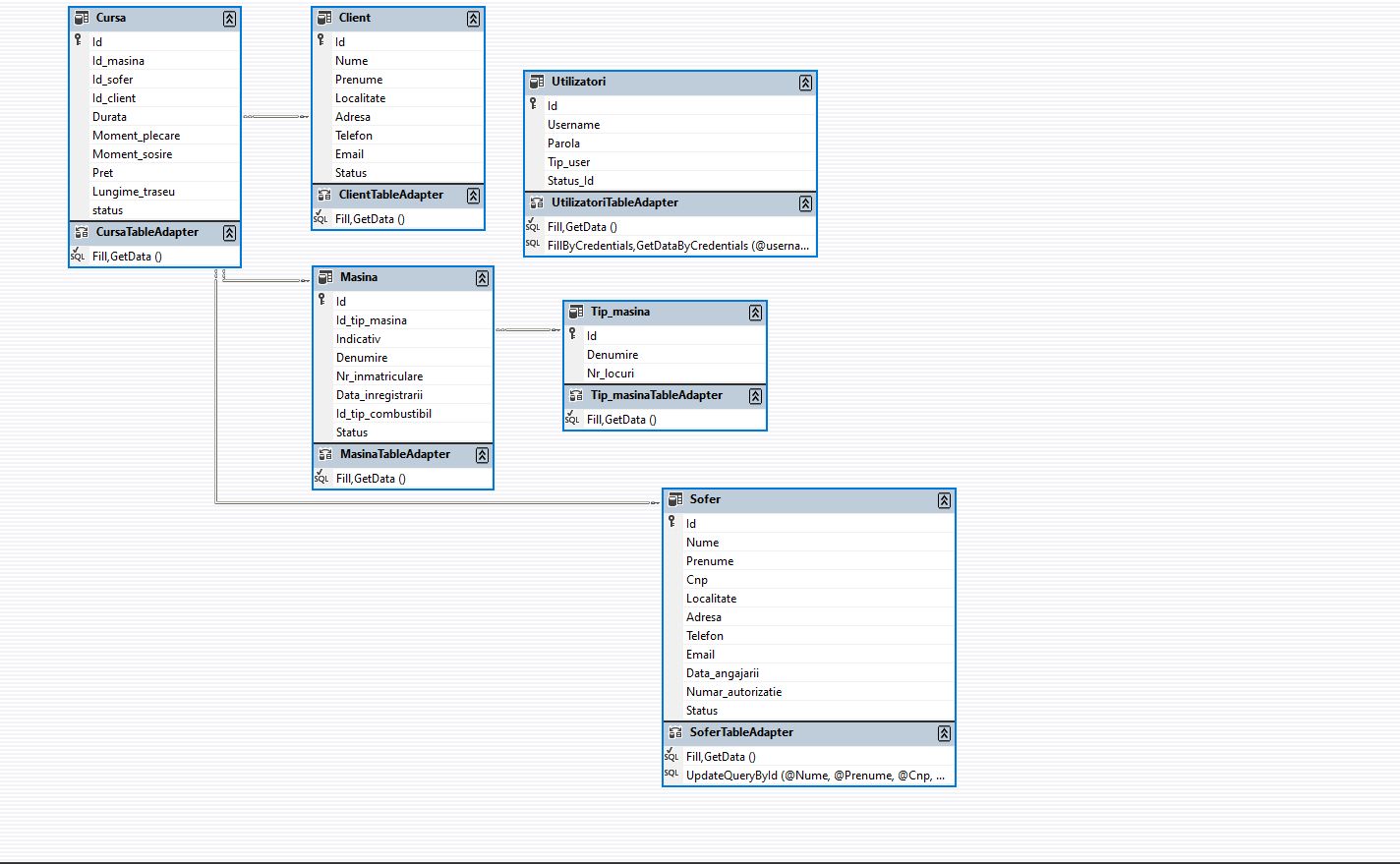
Gestiunea computerizată a unei firme este foarte importantă în era tehnologiei de astăzi. Din ce în ce mai mulți deținători de firme mici și mijlocii apelează la asemenea modalități de gestiune digitală pentru a le ușura activitatea zilnică. În funcție de cerințele individului sau a firmei ce concesionează o altfel de aplicație, ea poate implementa mai multe condiții specifice firmei: salarii, termene, plăti, etc. Astfel de aplicații pot funcționa în relație cu programe de software deja cunoscute precum Excel, Word, Acrobat pentru a face prelucrarea datelor mai facilă pentru un utilizator.

Acest proiect a fost realizat cu scop practic. Programul conține, la momentul actual, comenzi de adăugare, ștergere și modificare a elementelor constituiente unei firme de taximetrie (șoferi, mașini, clienți, curse), precum și funcții de prelucrare a datelor din baza de date (venit, kilometrii parcurși, număr de curse). Pe baza valorilor introduse de către utilizator, programul poate face gestiunea unei firme de taximetrie. Astfel, el poate fi utilizat în situații reale.

## Avantaje utilizare AllinOneTaxiApp

1. Interfata prietenoasa si usor de folosit de catre utilizator.
2. Fata de aplicatiile clasice de management, se adreseaza unor utilizatori care doresc implementari simple ale datelor companiei, fara a fi nevoiti de a folosi multiple fisiere- totul este realizat in cadrul aplicatiei.
3. Poate fi utilizat de catre orice individ fara cunostinte in aplicatii de management.

# Realizarea aplicației



Aceasta imagine prezintă data set-ul bazei de date a proiectului. Un set de date este un set de obiecte care stochează date dintr-o bază de date în memorie și acceptă urmărirea modificărilor pentru a permite operațiunile de creare, citire, actualizare și ștergere (CRUD) pe acele date fără a fi necesară conectarea întotdeauna la baza de date. Seturile de date au fost concepute pentru formulare simple peste aplicații de afaceri de date.

Aplicatia este creata in Visual Studio 2019, scrisa in limbajul C#.Baza de date a fost creată prin Microsoft SQL Server , un sistem de gestionare a bazelor de date relaționale (RDBMS) produs de compania Microsoft. Limbajul de interogare este SQL, iar extensia procedurală este T-SQL.

Pentru realizarea interfeței, am utilizat formularele de design din Visual Studio 2019, autogenerate prin xml scris de program. Logo-ul acestui site a fost creat integral in Logo Maker & Adobe Photoshop CC.

Una dintre cele mai mari probleme care a apărut odată cu dezvoltarea tehnologiei și introducerea unei varietăți din ce în ce mai mare de platforme și sisteme de operare diferite este portabilitatea codului. Limbajul C# a fost creat pentru a ajuta dezvoltatorii de aplicații să depășească această problemă. Cu toate că limbajul Java a fost primul care a oferit o soluție, acesta are câteva limitări. Una dintre acestea este interoperabilitatea limbajelor diferite, sau programarea în limbaj mixt (posibilitatea codului scris într-un limbaj de a lucra în mod natural cu codul scris în alt limbaj). Interoperabilitatea limbajelor diferite este esentială la realizarea sistemelor software de dimensiuni mari.

Ca parte a ansamblului strategiei .NET, dezvoltată de Microsoft, în jurul anului 2000 a fost creat limbajul C#. C# este direct înrudit cu C, C++, dar și cu Java. Precursorul limbajului C# este C-ul. De la C, C# moștenește sintaxa, multe din cuvintele cheie și operatorii. De asemenea, C# construiește peste modelul de obiecte definit în C++. Relația dintre C# și Java este mai complexă. Java derivă la rândul său din C și C++. Ca și Java, C# a fost proiectat pentru a produce cod portabil. Limbajul C# nu derivă din Java. Relația dintre cele două provine din faptul că ele derivă din același strămoș, dar se deosebesc prin multe caracteristici importante.

Limbajul C# conține mai multe facilități inovatoare, dintre care cele mai importante se referă la suportul incorporat pentru componente software. C# dispune de facilități care implementeaza direct elementele care alcătuiesc componentele software, cum ar fi proprietățile, metodele și evenimentele. Poate cea mai importantă facilitate de care dispune C# este posibilitatea de a lucra într-un mediu cu limbaj mixt.

C# are o legatură deosebită cu mediul său de rulare, arhitectura .NET. Arhitectura .NET definește un mediu de programare care permite dezvoltarea și execuția aplicațiilor, indiferent de platformă. Aceasta permite programarea în limbaj mixt și oferă facilități de securitate și portabilitate a programelor. Deși Microsoft nu a implementat-o în totalitate decât pentru sistemele de operare Windows, arhitectura a fost concepută astfel încât să fie cross-platform. Astfel, ea poate fi implementată cu succes și pe alte sisteme de operare, păstrând compatibilitatea.

Legat de C#, arhitectura .NET definește două entități importante și anume biblioteca de clase .NET sau Framework Class Library (FCL) și motorul comun de programare sau Common Language Runtime (CLR).

C# nu are o bibliotecă de clase proprie, ci utilizează direct biblioteca de clase .NET. De exemplu, când se ruleaza un program care efectuează operații de intrare-ieșire, cum ar fi afișarea unui text pe ecran, se utilizează biblioteca de clase

.NET.

Motorul comun de programare (CLR) se ocupă de execuția programelor C#. El asigură, de asemenea, programarea în limbaj mixt, securitatea și portabilitatea programelor. Atunci când este compilat un program C# sau un program în limbaj mixt, rezultatul compilării nu este un cod executabil. În locul acestuia, se produce un fișier care conține un tip de pseudocod numit limbaj intermediar sau pe scurt, IL (Intermediate Language). Acest fișier IL poate fi copiat în orice calculator care dispune de .NET CLR. Prin intermediul unui compilator denumit JIT (Just In Time), motorul comun de pogramare transformă codul intermediar în cod executabil. Procesul de conversie decurge astfel: atunci când un program .NET este executat, CLR activează compilatorul JIT. Compilatorul JIT convertește IL în cod executabil pe masură ce fiecare parte a programului este neceasră. In concluzie, orice program compilat pana in format IL poate rula in orice mediu pentru care CLR este implementat, asigurându-se portabilitatea.

## Scurtă descriere a programării orientate pe obiecte

Metodologiile de programare s-au modificat continuu de la apariția calculatoarelor pentru a ține pasul cu mărirea complexității programelor. Pentru primele calculatoare, programarea se făcea introducând instrucțiunile scrise în binar. Pe masură ce programele au crescut, s-a inventat limbajul de asamblare, în care se puteau gestiona programe mai mari prin utilizarea unor reprezentări simbolice ale instrucțiunilor mașină. Cum programele continuau să crească, s-au introdus limbaje de nivel înalt, precum FORTRAN și COBOL, iar apoi s-a inventat programarea structurată

POO a preluat cele mai bune idei de la programarea structurată, combinându-le cu concepte noi. A rezultat o modalitate diferită de a organiza un program. În fapt, un program poate fi organizat în două moduri: în jurul codului (mod de lucru descris de sintagma „codul acționeaza asupra datelor”, valabil în cazul programării structurate) sau în jurul datelor (abordare descrisă de sintagma

„datele controlează accesul la cod”, valabila in cazul programarii orientate obiect).

Toate limbajele POO au patru caracteristici comune: incapsularea,

polimorfismul, moștenirea și reutilizarea.

Incapsularea este un mecanism care combină codul și datele pe care le manipulează, menținând integritatea acestora față de interferența cu lumea exterioară. Când codul și datele sunt incapsulate, se creează un obiect. În cadrul unui obiect, codul și datele pot fi publice sau private. Codul și datele private sunt accesibile doar în cadrul aceluiași obiect, în timp ce codul și datele publice pot fi utilizate și din părți ale programului care există în afara acelui obiect. Unitatea fundamentală de incapsulare este clasa. Clasa specifică datele și codul care operează asupra datelor. O clasă definește forma unui obiect.

Polimorfismul este calitatea care permite unei interfețe să aibă acces la un grup generic de acțiuni. Termenul este derivat dintr-un cuvânt grecesc având semnificația “cu mai multe forme”. Spre exemplu, să presupunem că avem nevoie

de o rutină care să returneze aria unei forme geometrice, care poate fi un triunghi, cerc sau trapez. Întrucât ariile celor trei forme se calculeaza diferit, rutina trebuie să fie adaptată la datele pe care le primește, astfel încât să distingă despre ce fel de formă este vorba și să returneze rezultatul corect.

Moștenirea este mecanismul prin care o clasă preia structura (datele membre) și comportamentul (metodele) unei alte clase, la care adaugă elemente specifice. Se definesc conceptele de clasă de bază (clasa de la care se preiau structura și comportamentul) și clasă derivată (clasa care preia structura și comportamentul). Avantajele oferite de acest mecanism sunt posibilitatea reutilizării codului și obținerea extensiei unei clase fără a fi necesară o recompilare a clasei inițiale.

Atunci când este creată o clasă, aceasta poate fi utilizată pentru a crea o multitudine de obiecte. Prin utilizarea moștenirii și a incapsulării, clasa amintită poate fi reutilizată. Nu mai este necesară testarea codul respectiv, ci doar utlizarea sa corectă.

## Scurtă prezentarea a serverului de baze de date MSSql Server

Microsoft SQL Server este un sistem de gestionare a bazelor de date relaționale (RDBMS) produs de compania Microsoft. Limbajul de interogare este SQL, iar extensia procedurală este T-SQL.

Acesta suportă versiunea companiei Microsoft de SQL (Structured Query Language), cel mai răspândit limbaj de interogare pentru bazele de date. Este considerat un sistem de gestionare a bazelor de date pentru întreprinderi. El se poate aplica bazelor de date de dimensiuni foarte mari. Cu toate aceste, ultimele versiuni au venit cu mai multe tipuri de licențiere, de la cele free la unele avansate, astfel că în prezent se poate considera că SQL Server este potrivit pentru baze de date de orice dimensiune.

Codul de bază pentru Microsoft SQL Server își are originile în Sybase SQL Server și a reprezentat intrarea Microsoft pe piața bazelor de date relaționale, concurând cu Oracle, IBM și Sybase. Microsoft, Sybase si Ashton-Tate s-au unit pentru a crea și a scoate pe piață prima versiune numită SQL Server 4.2. Mai tarziu Microsoft a negociat pentru drepturi de exclusivitate la toate versiunile de SQL Server concepute pentru sistemele de operare Microsoft. Sybase și-a schimbat ulterior numele în Adaptive Server Enterprise, pentru a evita confuzia cu Microsoft SQL Server.

SQL Server 7.0 a fost primul server de baze de date bazat pe o interfață grafică (GUI). O variantă de SQL Server 2000 a fost prima variantă comercială pentru arhitectura Intel. În anii următori s-au lansat și alte versiuni, ce au adus îmbunătățiri de performanță, unelte IDE și ETL, suport pentru date XML, geometrice sau spațiale, căutare în text, self-tuning, clustering (un cluster este un grup de 2 sau mai multe servere fizice, cu același nivel de acces la stocare, care au rolul de a asigura resursele necesare stocării datelor din baza de date și care poartă denumirea de noduri) și HADR (high availability and disaster recovery).

Microsoft SQL Server folosește o variantă de SQL numită T-SQL, sau Transact-SQL, o implementare de SQL-92 (standardul ISO pentru SQL) cu unele extensii. T-SQL adaugă sintaxă adițională pentru procedurile stocate și pentru tranzacții. Standardele SQL necesită ACID: cele 4 condiții ce trebuie îndeplinite de orice tranzacție: atomicitate, consistență, izolare, durabilitate. MS SQL Server suportă ODBC (Open Database Connectivity).

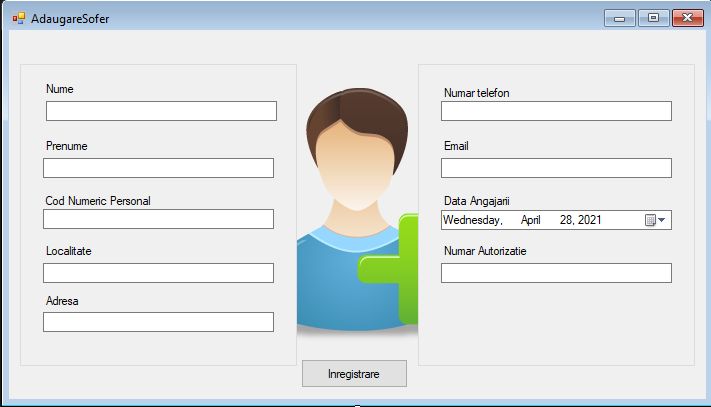
Stocarea datelor se face pe modelul relațional, sub forma unei colecții de tabele cu rânduri și coloane. Fiecare coloană este de un anumit tip: SQL Server suportă atât tipurile primare, precum întreg, zecimal, caracter, dată calendaristică, dar și tipuri mai complexe precum text, date binare, date geometrice, date spațiale, XML, etc.

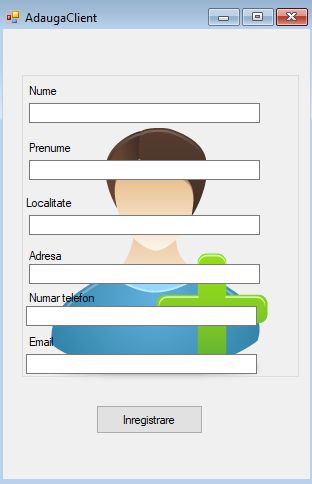
## Descrierea interfeței



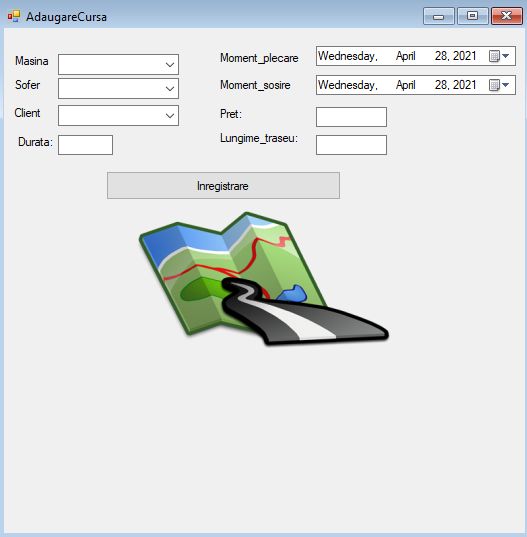
Meniul principal al aplicației. Există butoane pentru fiecare tip de operație pe care un administrator al unei firme de taxi ar dori să le întreprindă, de la adaugari de șoferi, client, mașini sau curse până la modificări ale acestora sau ștergeri. Există și butoane ce întreprind comenzi special, precum generarea unui total de venituri, total de kilometrii sau total de curse într-o zi aleasă de către utilizator.

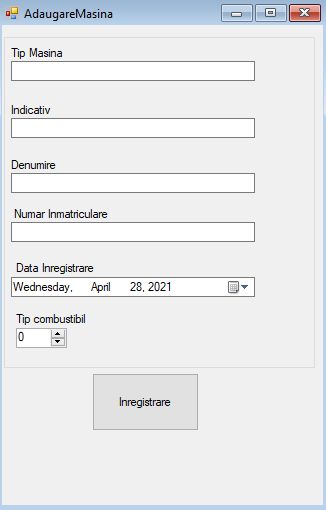
Fiecare dintre butoanele cu pictograma de plus deschid o interfață de adăugare în baza de date a elementului.

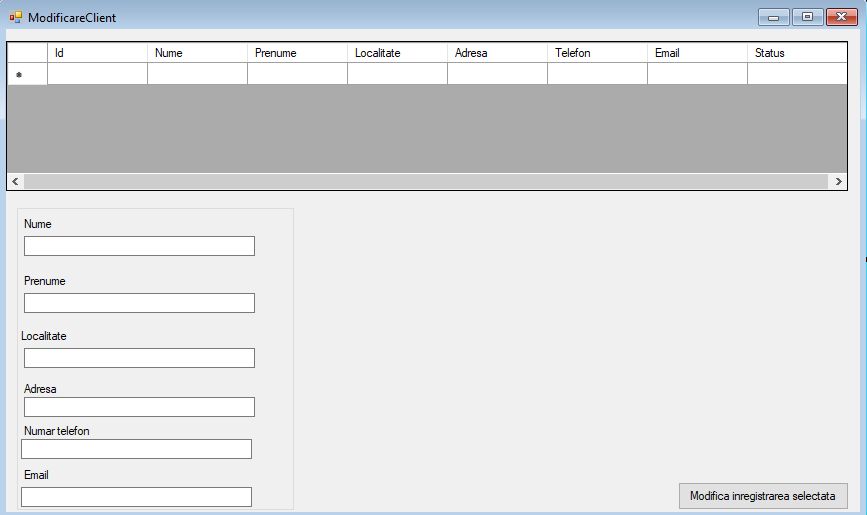
Formularul de adaugare a unui sofer. În stânga și în dreapta se completează valorile pentru fiecare parametru. Butonul „Înregistrare” declanșează algoritmul de înregistrare în baza de date a noii intrări în tabela Sofer.



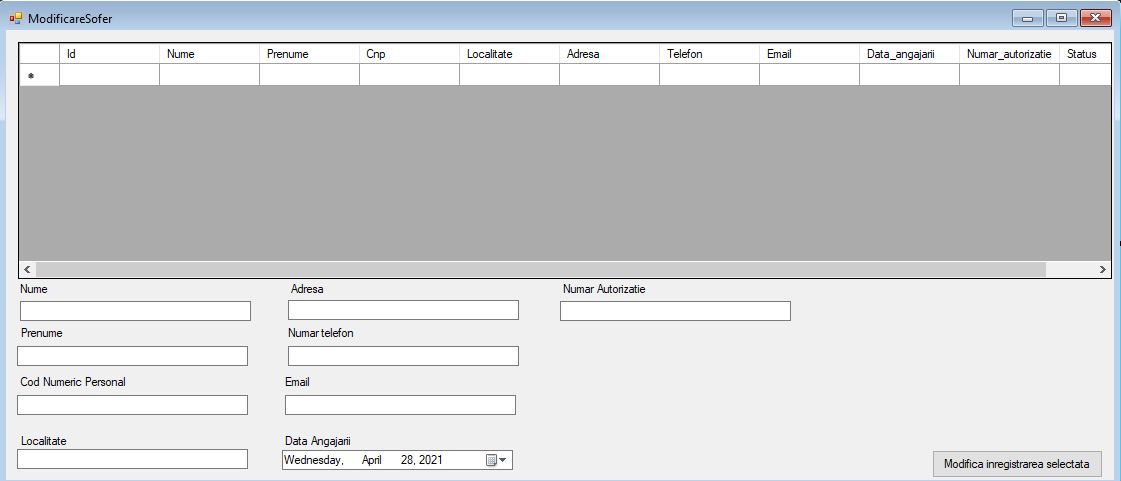
Formularul de adaugare a unui client. În stânga și în dreapta se completează valorile pentru fiecare parametru. Butonul „Înregistrare” declanșează algoritmul de înregistrare în baza de date a noii intrări în tabela Client.

Formularul de adaugare a unei curse În stânga și în dreapta se completează valorile pentru fiecare parametru. Butonul „Înregistrare” declanșează algoritmul de înregistrare în baza de date a noii intrări în tabela Cursa.

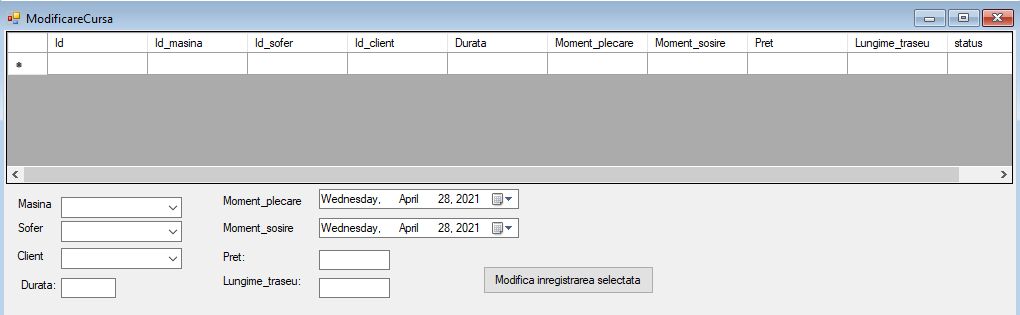


Formularul de adăugare a unei mașini. În stânga și în dreapta se completează valorile pentru fiecare parametru. Butonul „Înregistrare” declanșează algoritmul de înregistrare în baza de date a noii intrări în tabela Masina.

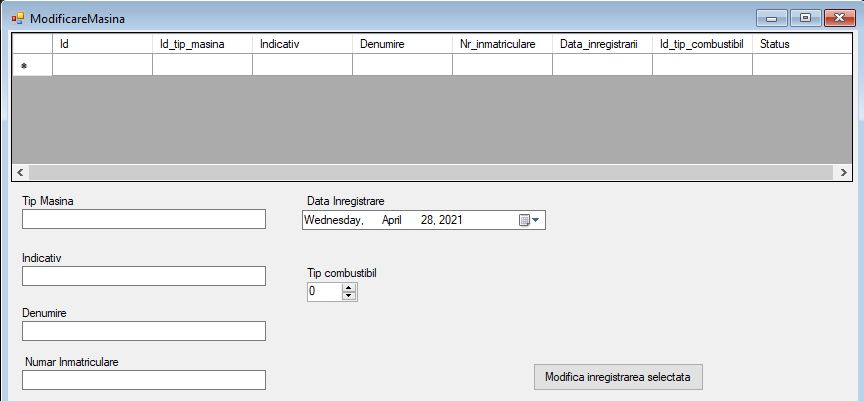
Formularul de modificare a unei intrari in tabela “Client”. În stânga se completează valorile pentru fiecare parametru. Elementul se selectează cu ajutorul elemental de tip DataGridView. Butonul „Modifică înregistrarea selectată”declanșează algoritmul de modificării în baza de date a intrării in tabela “Client”.

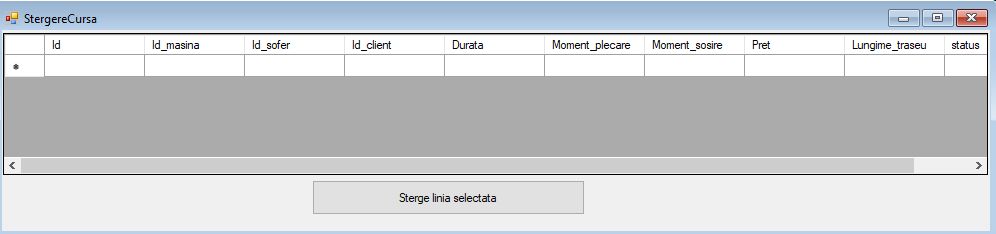


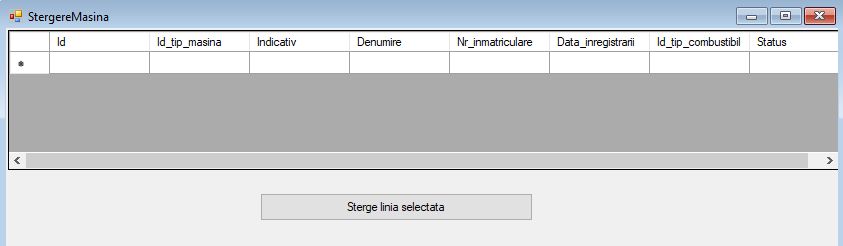
Formularul de modificare a unei intrari in tabela “Sofer”. În stânga se completează valorile pentru fiecare parametru. Elementul se selectează cu ajutorul elemental de tip DataGridView. Butonul „Modifică înregistrarea selectată” declanșează algoritmul de modificării în baza de date a intrării in tabela “Sofer”.



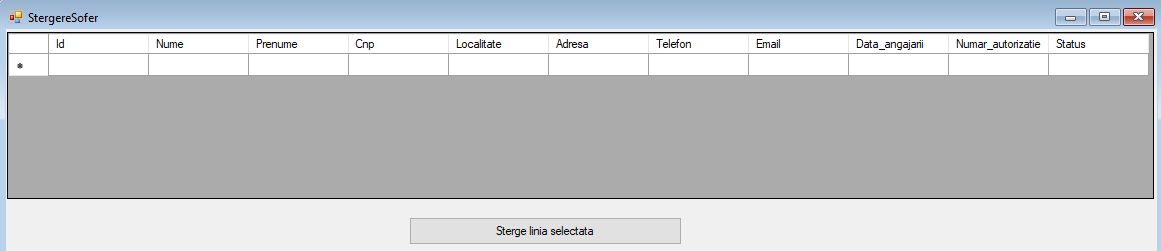
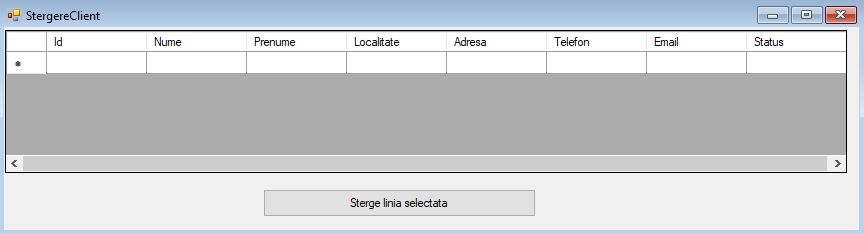
Formularul de modificare a unei intrari in tabela “Cursa”. În stânga se completează valorile pentru fiecare parametru. Elementul se selectează cu ajutorul elemental de tip DataGridView. Butonul „Modifică înregistrarea selectată” declanșează algoritmul de modificării în baza de date a intrării in tabela “Cursa”.

Formularul de modificare a unei intrări în tabela “Masina”. În stânga se completează valorile pentru fiecare parametru. Elementul se selectează cu ajutorul elemental de tip DataGridView. Butonul „Modifică înregistrarea selectată” declanșează algoritmul de modificării în baza de date a intrării in tabela “Masina”.

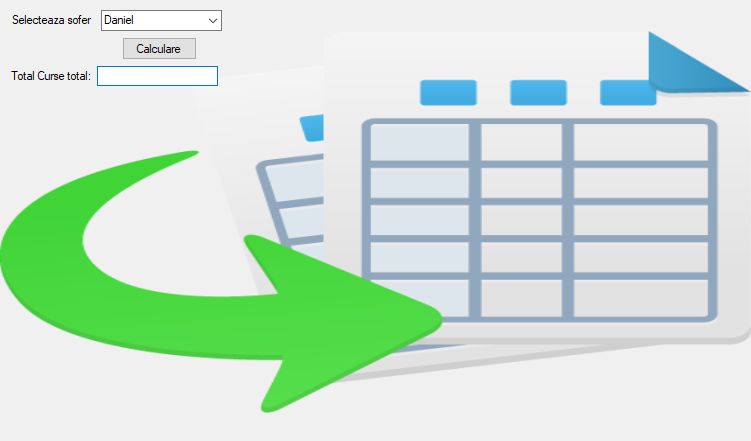
Formularul de ștergere a unei intrări în tabela “Cursa”. Elementul se selectează cu ajutorul elemental de tip DataGridView. De reținut că ștergerea unui element nu poate fi realizată din cauza condițiilor de arhivare a datelor unei companii și a activitatiilor sale. Butonul „Șterge linia selectată” modifică intrarea câmpului Status din cadrul tabelei din 1 în 2, astfel marcandu-se “ștergerea”, ce este defapt o marca ce arată dacă data din tabel mai este activă sau nu.



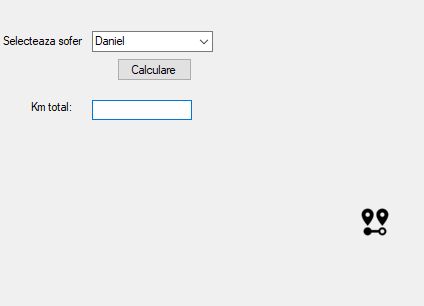
Formularul de ștergere a unei intrări în tabela “Masina”. Elementul se selectează cu ajutorul elemental de tip DataGridView. De reținut că ștergerea unui element nu poate fi realizată din cauza condițiilor de arhivare a datelor unei companii și a activitatiilor sale. Butonul „Șterge linia selectată” modifică intrarea câmpului Status din cadrul tabelei din 1 în 2, astfel marcandu-se “ștergerea”, ce este defapt o marca ce arată dacă data din tabel mai este activă sau nu.

Formularul de ștergere a unei intrări în tabela “Sofer”. Elementul se selectează cu ajutorul elemental de tip DataGridView. De reținut că ștergerea unui element nu poate fi realizată din cauza condițiilor de arhivare a datelor unei companii și a activitatiilor sale. Butonul „Șterge linia selectată” modifică intrarea câmpului Status din cadrul tabelei din 1 în 2, astfel marcandu-se “ștergerea”, ce este defapt o marca ce arată dacă data din tabel mai este activă sau nu.

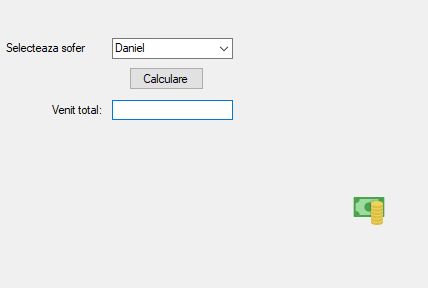
Formularul de ștergere a unei intrări în tabela “Client”. Elementul se selectează cu ajutorul elemental de tip DataGridView. De reținut că ștergerea unui element nu poate fi realizată din cauza condițiilor de arhivare a datelor unei companii și a activitatiilor sale. Butonul „Șterge linia selectată” modifică intrarea câmpului Status din cadrul tabelei din 1 în 2, astfel marcandu-se “ștergerea”, ce este defapt o marca ce arată dacă data din tabel mai este activă sau nu.



Formularul de număr de curse. Acest formular are un parametru de sofer ales, și pe baza acestui parametru se generează numărul total de curse al acelui sofer din baza de date.



Formularul de număr de kilometrii. Acest formular are un parametru de un parametru de sofer ales, și pe baza acestui parametru se generează numărul total de kilometrii al acelui sofer din baza de date.



Formularul de venit. Acest formular are un parametru de un parametru de sofer ales, și pe baza acestui parametru se generează venitul total al acelui sofer din baza de dat

# Bibliografie

1. [Crișan Daniela Alexandra](https://www.prouniversitaria.ro/autor/daniela-alexandra-crisan), Programarea aplicatiilor folosind limbajul C# si platforma .NET, Ed Pro-Universitaria (2015);
2. Herbert Schildt, C#: A Beginner’s Guide, (2001);
3. Karli Watson et al., Beginning Visual C#, Wrox Press Ltd. (2002);
4. <https://www.c-sharpcorner.com/>
5. <http://www.functionx.com/csharp/>