

## Documentație proiect 1 CarService

Subsemnata Bîrsan Ioana, declar pe proprie răspundere că acest cod nu a fost copiat din Internet sau din alte surse.

### Bibliografie:

- <https://blog.devart.com/set-identity-and-computed-properties-in-entity-framework-without-trigger-s.html>
- <https://stackoverflow.com/questions/25894587/how-to-update-record-using-entity-framework-6>
- <https://stackoverflow.com/questions/35799017/adding-attributes-to-ef6-generated-entity-classes>
- <http://alexwolfthoughts.com/adding-validation-metadata-to-entity-framework-generated-classes/>
- <https://docs.microsoft.com/en-us/aspnet/mvc/overview/getting-started/database-first-development/enhancing-data-validation>
- <https://dev.to/kylegalbraith/getting-familiar-with-the-awesome-repository-pattern--1ao3>
- <https://deviq.com/repository-pattern/>

### Noutăți:

Deoarece aveam nevoie să adaug reguli de validare folosind atrbute la nivelul proprietăților din clasele parțiale generate, iar acestea s-ar fi rescris la următoarea regenerare, am folosit următoarea abordare:

- pentru fiecare clasă parțială generată (e.g. Auto), am creat o clasă nouă (e.g. AutoMetadata), în care am adăugat reguli de validare la nivel de proprietate
- pentru fiecare clasă parțială generată am creat o nouă clasă parțială (e.g. Auto) și am utilizat atributul MetadataType (e.g. [MetadataType (typeof(AutoMetadata))])

Aceste reguli sunt acum impuse în proiect și nu vor fi sterse în momentul în care se regenerează clasele. Clasele create se află în folderul **Models**.

Am implementat şablonul Repository pentru a oferi un nivel de abstractizare între nivelul de persistență (baza de date AUTO) și nivelul de business (ICarService, CarService). Am creat o interfață generică IRepository în care expun următoarele metode:

```
public interface IRepository<T>
{
    T GetById(int id);
    IReadOnlyList<T> GetAll();
    void Create(T item);
    void Delete(int id);
    void Update(T item);
    bool Exists(int id);
    void SaveChanges();
}
```

Pentru fiecare model, am implementat această interfață, folosind ca dependență contextul generat CarServiceModelContainer. Unde a fost cazul am expus și alte metode necesare pentru logica aplicației. Nivelul acesta se regăsește în folderul **Repository**.

Nivelul de logică al aplicației, care expune API-ul cerut, se află în folderul **Service**. Am creat o interfață: ICarService și o clasă care o implementează: CarService. Dependentele folosite sunt cele din nivelul de repository.

Pentru testare, am creat în cadrul aceleiași soluții un proiect de tip Console App (TestCarService) căruia i-am adăugat referință către proiectul de tip Class Library (CarService). În arhivă voi adăuga doar fișierul numit Program.cs.

## **Actualizare Proiect 1**

1. Am eliminat folderul **Repository**, eliminând astfel redundanță (Entity Framework deja implementează Repository pattern, aşa cum ați menționat și dumneavoastră).
2. Am eliminat folderul **Service**, în locul lui am introdus folderul **Api** în care am adăugat doar clasa care expune metodele publice necesare (**CarServiceApi**)
3. Am corectat erorile de logică din modelul creat cu ADO.Net, am actualizat și relațiile de asociere dintre modele, am adăugat o nouă valoare enum-ului **StareComandă**.
4. Am adăugat atributele: `[DataContract(IsReference = true)]` la nivel de clasa pentru fiecare model, respectiv `[DataMember]` la nivel de proprietate pentru fiecare model.
5. Am înlocuit `ICollection<Model>` cu `List<Model>` pentru a rezolva o problema la serializare.

## **Documentație proiect 2 CarService**

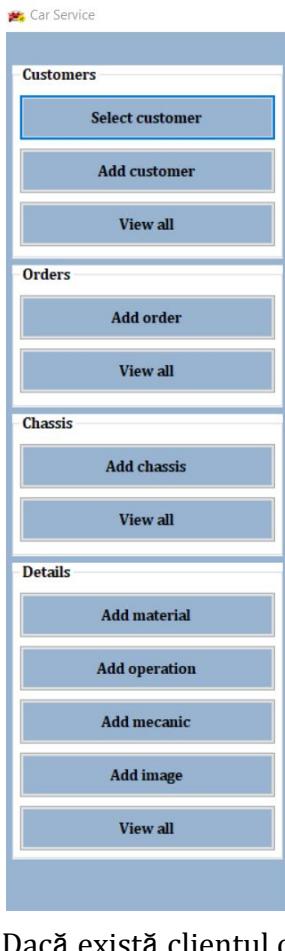
### **Bibliografie:**

- <https://docs.microsoft.com/en-us/dotnet/framework/winforms/>
- <https://docs.microsoft.com/en-us/dotnet/api/system.drawing.image?view=netframework-4.7.2>
- <https://www.youtube.com/watch?v=sGP6u68k2hc>
- <https://stackoverflow.com/questions/3801275/how-to-convert-image-to-byte-array>
- <https://www.youtube.com/watch?v=C9s0H6yeFLQ>

### **Interfață:**

Pentru această secțiune voi adăuga niște screenshot-uri, însorite de explicații.

În prima etapă, se caută existența unui client. În cazul în care acesta nu există se va afișa un mesaj corespunzător. Pentru fiecare câmp ce trebuie completat există validări corespunzătoare, iar butonul de *Search* nu este activat decât în momentul în care toate validările trec și sunt completeate toate câmpurile obligatorii.



Search customer

Name	First name	Phone Number
Birsan	Ioana	1234561234567

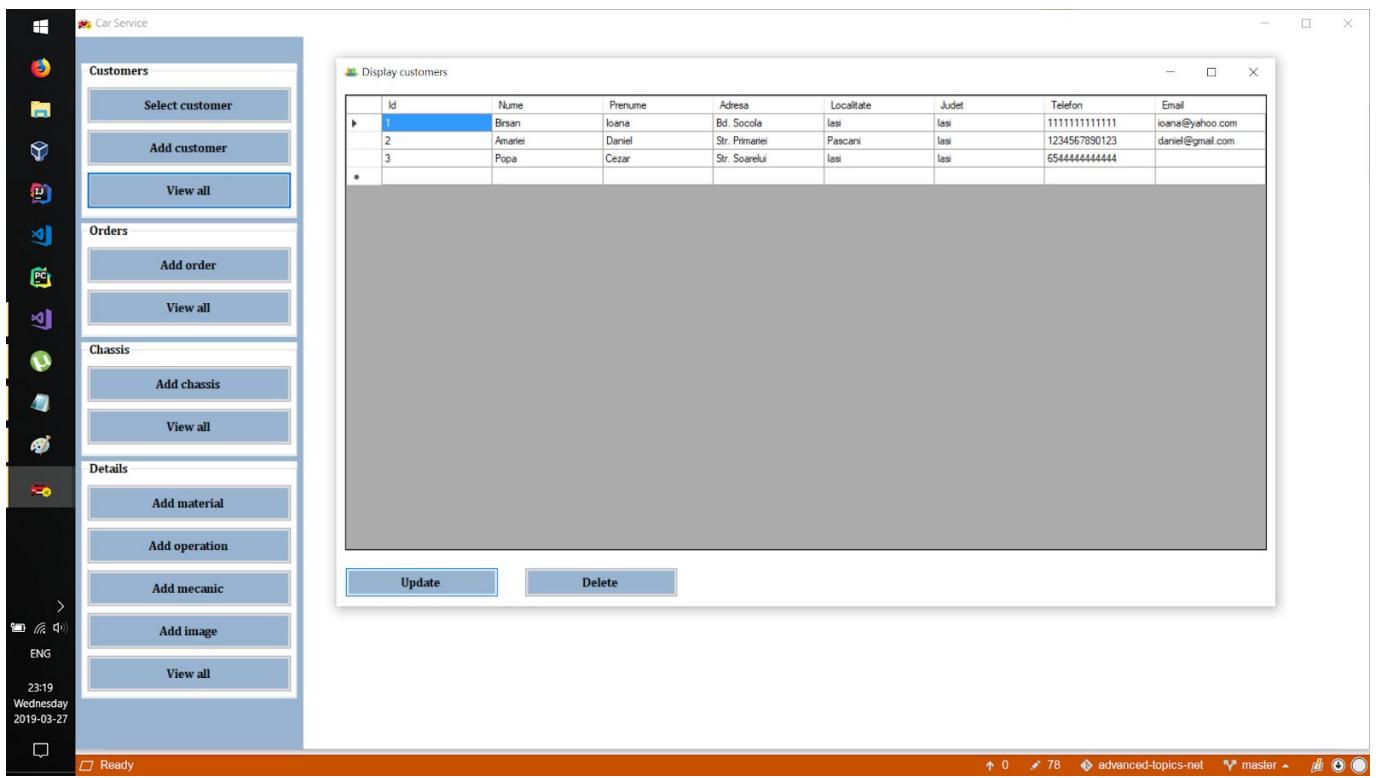
Email

There is no customer with these specification. Please add new customer.

Dacă există clientul căutat, atunci se vor afișa mașinile sale, precum și opțiunea de a adăuga o nouă mașină. La fel ca în cazul anterior, are loc validarea câmpurilor necesare.

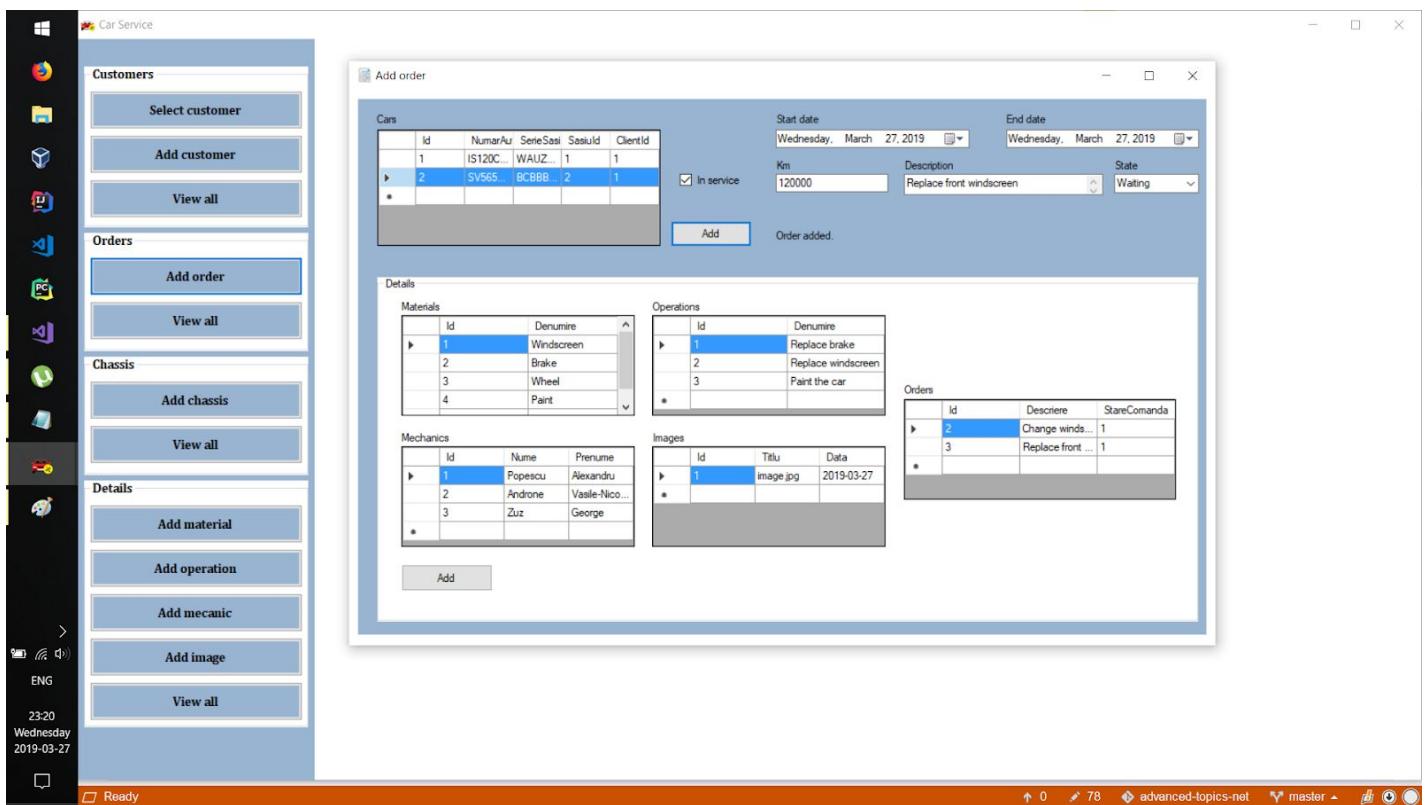
ID	Nume Auto	Serie Sasiu	Sasiul	Cliențid
1	IS120CPP	WAUZZZ4FZ6A111222	1	1
2	SV565MNB	BCBBBN5GZ6A233344	2	1
*				

Pentru fiecare grup (Customers, Orders, Chassis, Details) există posibilitatea de a vizualiza toate înregistrările existente. La selectarea unui rând, avem posibilitatea de a actualiza informațiile existente, dar și de ștergere a unei înregistrări. Am exemplificat mai jos doar pentru Customers.

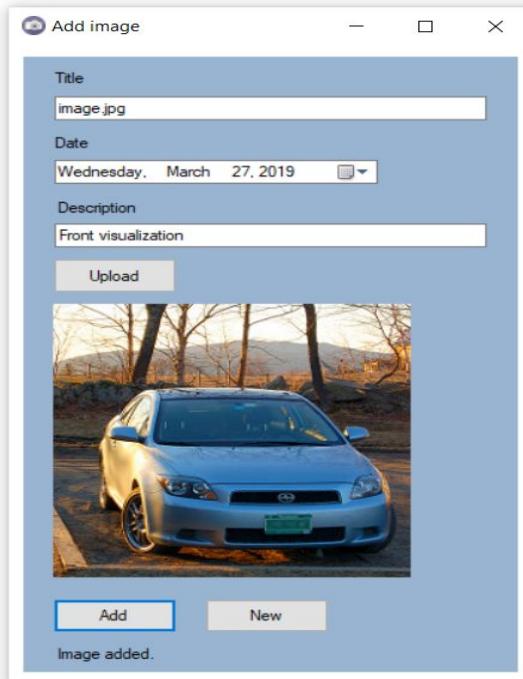
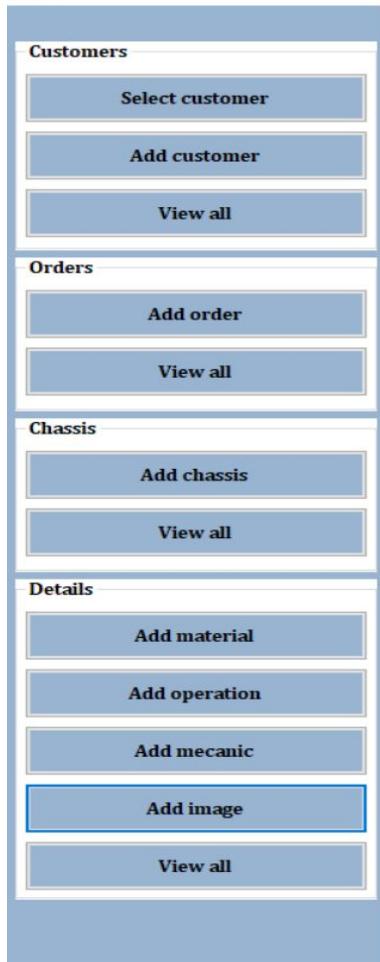


Pentru plasarea unei comenzi se alege o mașină, împreună cu toate datele necesare creării unei comenzi. De asemenea, datele sunt validate corespunzător. În funcție de opțiunea de a lăsa/nu mașina în service, se va activa/dezactiva posibilitatea de completare a câmpului Km.

După plasarea unei comenzi, se poate opta pentru adăugarea detaliilor necesare procesării comenzi selectate. Are loc calculul costului total pentru reparații (am considerat și un cost de service), împreună cu actualizarea stocului de materiale.



Aici este prezentată opțiunea de adăugare a unei imagini.



Vizualizare înregistrări detalii: materiale, mecanici, operațiuni și imagini. Pentru fiecare avem posibilitatea de editare sau ștergere.

This screenshot shows a detailed view of the application. On the left is the same main menu as before. The right side features a window titled 'Display details' with tabs for Mechanics, Operations, Materials, and Images. The Mechanics tab is selected, displaying a table with the following data:

ID	Denumire	Cantitate	Pret	DataAprovisionare
1	Windscreen	54.00	210.00	2019-03-27
2	Brake	120.00	65.00	2019-03-27
3	Wheel	456.00	50.00	2019-03-27
4	Paint	200.00	35.00	2019-03-27

At the bottom of this window are 'Update' and 'Delete' buttons. The status bar at the bottom of the screen shows 'Ready'.

## Documentație proiect 3 CarService

### Bibliografie:

- <https://profs.info.uaic.ro/~iasimin/>
- <https://docs.microsoft.com/en-us/dotnet/framework/wpf/>
- <https://stackoverflow.com/questions/20169765/wpf-datagrid-how-to-determine-the-current-row-index>
- <https://www.godo.dev/tutorials/wpf-load-external-image/>
- <https://stackoverflow.com/questions/553611/wpf-image-to-byte>

Am creat 3 proiecte: **HostWCF** (refera ObjectWCF) de tipul Console App, **ObjectWCF** (refera CarService) de tipul Class Library si **WpfAppCarService** (refera HostWCF) de tipul WPF App. Modificările pe care le-am facut primul proiect, **Car Service**, le-am mentionat mai sus.

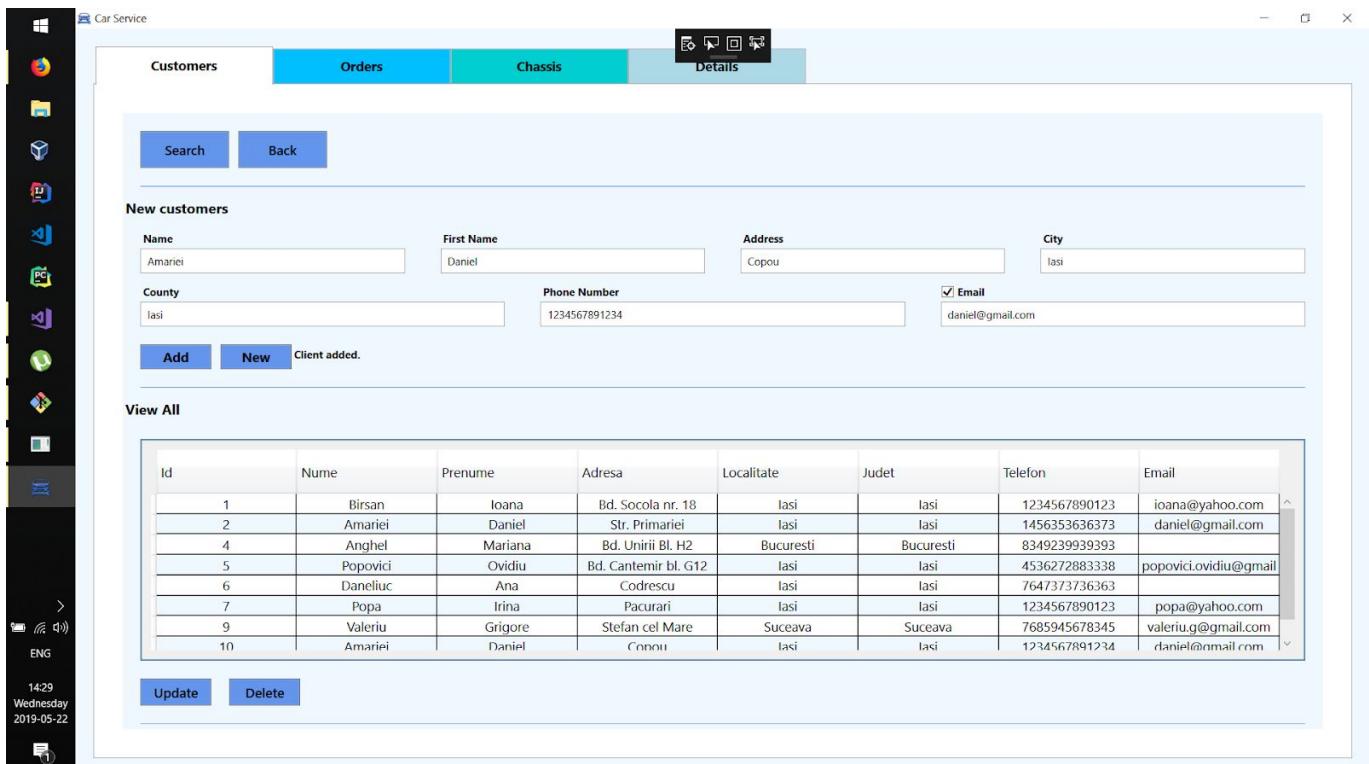
Pentru HostWCF am folosit codul oferit de dumneavoastra in exemplul cu Post si Comment.

Operatiile din ObjectWCF fac redirectare catre operatiile din CarService.

Pentru partea de client, am pornit host-ul si am executat comanda: svcutil.exe http://localhost:8000/CarService -config:App.config -out:proxy.cs

Fisierele generate le-am copiat in proiectul WPF. Am implementat UI-ul direct in fisierul **MainWindow.xaml** (este mai usor decat folosind designer-ul), iar functionalitatea in **MainWindow.xaml.cs**.

In continuare voi atasa screenshot-uri insotite de scurte explicatii.



Spre deosebire de proiectul realizat folosind Windows Forms, acum folosesc doar o singura fereastra: MainWindow in care am adaugat un TabControl ce cuprinde urmatoarele sectiuni: Customers, Orders, Chassis, si Details.

In sectiunea **Customers** avem posibilitatea de a adauga noi clienti, de a vizualiza, sterge sau modifica clientii existenti. Toate campurile sunt validate in timp ce se tasteaza pentru a nu introduce date neconforme.

De asemenea, putem cauta un client, folosind butonul Search, in functie de anumite criterii: daca este gasit, putem vizualiza masinile pe care le detine, eventual se poate adauga o noua masina. In caz ca nu exista un asemenea client, ne intoarcem apasand butonul Back si adaugam un nou client.

Id	NumarAuto	SerieSasiu	Sasiuld	ClientId
1	IS200ABN	WAUZZZ4FZ6A11122	1	1
2	SV786ZXZ	WAUZZZ4FZ6A8569494	1	1
3	IS222AD	WAUZZZ4FZ6A1113	1	1

In sectiunea **Orders** avem posibilitatea de a adauga o noua comanda sau de a vizualiza si opera asupra comenzilor inregistrate deja. In imaginea de mai jos este prezentata subsectiunea in care se poate adauga o comanda noua impreuna cu detaliiile atasate acesteia. Fiind destul de multe informatii: am folosit un ScrollViewer. De asemenea datele sunt validate.

Id	NumarAuto	SerieSasiu	Sasiuld	ClientId	Start Date	End Date
1	IS200ABN	WAUZZZ4FZ6A1	1	1	2019-05-23	2019-06-07
2	SV786ZXZ	WAUZZZ4FZ6A8	1	1		
3	IS222AD	WAUZZZ4FZ6A1	1	1		

Apasand butonul View all, se pot vizualiza, modifica sau sterge comenziile inregistrate.

The screenshot shows the 'Orders' section of the application. At the top, there are tabs for 'Customers', 'Orders', 'Chassis', and 'Details'. Below the tabs, there are two buttons: 'View all' (highlighted in blue) and 'Back'. A large table displays 10 rows of order data. The columns are: Id, DataSystem, DataProgramare, DataFinalizare, KmBord, Descriere, ValoarePiese, StareComanda, and AutId. The data in the table includes various dates, kilometers, descriptions like 'Change brakes and', and values like 134.00. At the bottom of the table area are 'Update' and 'Delete' buttons.

In sectiunea Chassis avem posibilitatea de a adauga, sau de modifica si sterge sasiurile inregistrate.

The screenshot shows the 'Chassis' section of the application. At the top, there are tabs for 'Customers', 'Orders', 'Chassis', and 'Details'. Below the tabs, there are input fields for 'Code' (containing '7S') and 'Name' (containing 'Dacia Duster'). There are also 'Add' and 'New' buttons. A message 'Chassis added.' is displayed. Below this, there is a table titled 'Chassis' with columns: Id, CodSasiu, and Denumire. The table contains 8 rows of chassis information, such as Audi A5 2010-2015 and Ford Raptor 2018. At the bottom of the table area are 'Update' and 'Delete' buttons.

In sectiunea **Details** sunt prezente toate functionalitatile necesare pentru procesarea informatiilor ce tin de detalii unei comenzi. In acesta sectiune am folosit un nou TabControl cu sectiunile: Images, Materials, Mechanics, si Operations. Cand selectam sectiunea Details, implicit se poate vizualiza ce este in sectiunea Images.

Am folosit aceeasi structurare a informatiilor pentru aceste sectiuni. Voi prezenta in continuare doar imaginile. Precizez ca datele de intrare sunt validate.

Car Service

Customers Orders Chassis Details

Images Materials Mechanics Operations

Title: accident4.jpg Description: Lateral and front view Date: 2019-05-21

Upload Add New

Image added.

Images

ID	Titlu	Descriere
4	image1.jpg	Front view
5	test.jpg	Try WPF
6	test2.jpg	Meaa
7	accident4.jpg	Lateral and front view

Update Delete

Car Service

Customers Orders Chassis Details

Images Materials Mechanics Operations

Name: Mirror Quantity: 50 Price: 45 Date Supply: 2019-05-21

Add New Material added.

Materials

ID	Denumire	Cantitate	Pret	DataAprovisionare
1	Paint	-3.00	34.00	3/28/2019 12:00:00 AM
2	Wheel	191.00	55.00	3/15/2019 12:00:00 AM
3	Windscreen	22.00	150.00	3/15/2019 12:00:00 AM
4	Brake	97.00	50.00	3/12/2019 12:00:00 AM
5	Back door	22.00	125.00	4/3/2019 12:00:00 AM
7	Upholstery	4.00	75.00	5/24/2019 12:00:00 AM
10	Mirror	50.00	45.00	5/21/2019 12:00:00 AM

Update Delete

Car Service

Customers Orders Chassis Details

Images Materials Mechanics Operations

Name First Name

Add New

Mechanics

ID	Nume	Prenume
1	Andrici	Cezar
2	Popescu	Viorel
3	Voda	Petru-Marian
5	Vasiliu	Constantin
6	Manoliu	Vasile

Update Delete

18:23 Wednesday 2019-05-22 ENG

ID	Nume	Prenume
1	Andrici	Cezar
2	Popescu	Viorel
3	Voda	Petru-Marian
5	Vasiliu	Constantin
6	Manoliu	Vasile

Car Service

Customers Orders Chassis Details

Images Materials Mechanics Operations

Name Execution Time

Add New

Operations

ID	Denumire	Timp Executie
1	Change wheel	1.00
2	Replace brakes	1.00
3	Replace windscreen	2.00
4	Replace windscreen	3.00
5	Change wheel	1.00
6	Schimba placute frana fata	12.00

Update Delete

18:24 Wednesday 2019-05-22 ENG

ID	Denumire	Timp Executie
1	Change wheel	1.00
2	Replace brakes	1.00
3	Replace windscreen	2.00
4	Replace windscreen	3.00
5	Change wheel	1.00
6	Schimba placute frana fata	12.00