Описание на проект

**1.Тема:** „Приложение за авто-услуги и сервизи“

**1.1.Цели:**Приложението ще има за цел да улеснява работата на сервизите. Също ще намалило нуждата от излишно използване на хартия. Улеснение както за фирмата така и за клиента при следене и записване на направените услуги по даден автомобил.То ще улесни счетоводството в дадена фирма.

**2.**Използвани технологии:

За разработка използвах програмния език C# и по-точно “WindowsForms Apps” с помощта на който направих графичния дизайн. В приложението също са използвани различни nugget пакети от които:

-Microsoft.EntityFrameworkCore

-Microsoft.EntityFrameworkCore.Design

-Microsoft.EntityFrameworkCore.SqlServer

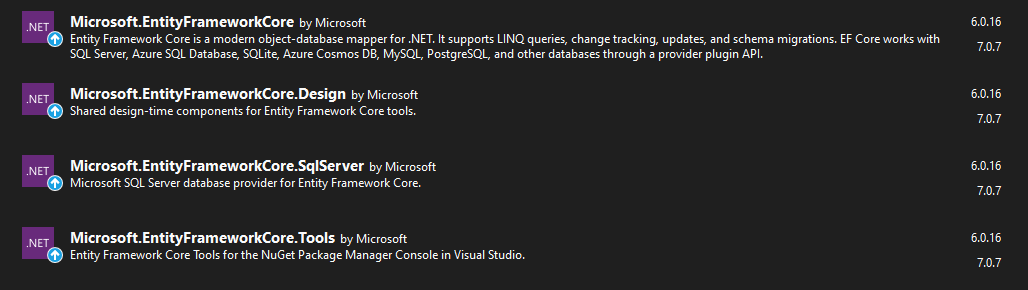
-Microsoft.EntityFrameworkCore.Tools

Тези пакети ни предоставят:

-Връзката между База Данните и С# кода

-използването на LinQ заявки при CRUD операции

-работа с Code-First и DB-First



**3.Описание на приложението:**

**3.1 Кратко описание:**

Приложението ще улеснява работата на сервизите то е улеснение за следене и записване на направените услуги по даден автомобил и също улеснение при работа с клиентите което би довело до улеснение на счетоводството в дадена фирма.

**3.2Основни функции:**

Приложението се стартира чрез Visual Studio. Трябва да има инсталиран SSMS иначе е невъзможно неговото употребяване.

**3.3 Основни модули и врзъки между тях:**

Основни модули са DataLayer и BussinessLayer. DataLayer-а отговаря за entity-тата, dbContext. BussinesLayer-а отговаря за основните методи за CRUD операциите. Връзките са между тях са:

DataLayer към BussinesLayer



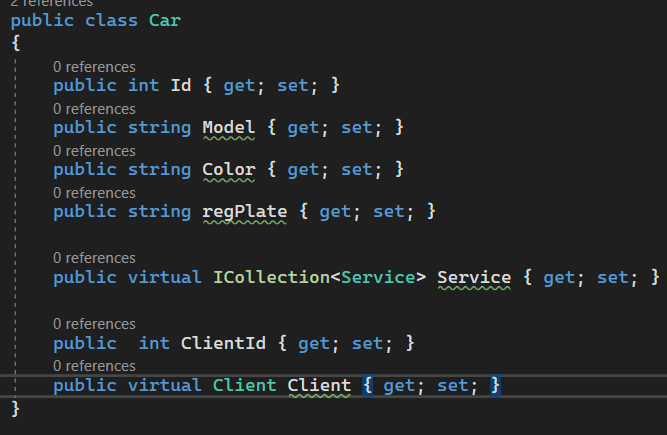
BussineLayer и DataLayer към формата



**4.Архитектура на проекта. Реализация:**

**4.1База Данни:** Приложението за авто-услуги и сервизи е съставено от 3 свързани компонента:Data която е class Library. В него са класовете(entity-тата):

**Car** за автомобила който ще се извърши дадена услуга:



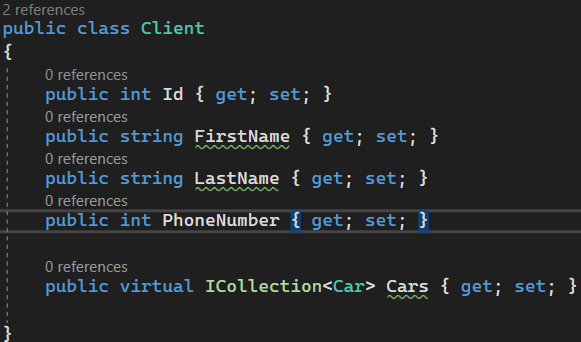
-Id от тип int, уникалния записа на автомобила

-Model от тип string, модел на автомобила

-Color от тип string, цвят на автомобила

-regPlate от тип string, регистрационен номер на автомобила

В **Client** въвеждаме информация за клиента на когото е автомобила:



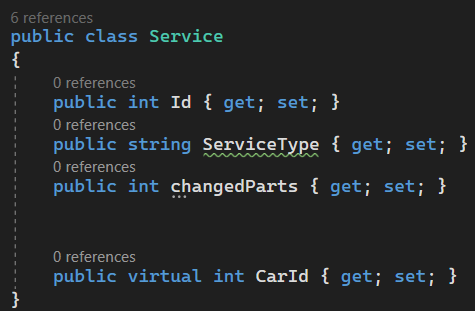
-Id от тип int, уникалния запис на клиента

-FirstName от тип string, Първото име на клиента

-lastName от тип string, Фамилия на клиента

-PhoneNumber от тип int, Телефония номер на клиент за връзка с него

И клас **Service** за услугите на дейноста:



-Id от тип int, уникалния запис на услугата

-ServiceType от тип string, описание на услуга

-changedParts от тип int, брой на сменените части

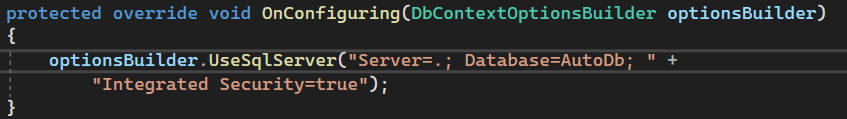
**Всеки клас отговаря за определена таблица**

**Car(entity)** към **Cars(таблица)**

**Client(entity)** към **Clients(таблица)**

**Service(entity)** към **Services (таблица)**

**Името на базата данние е AutoDb:**



**Използван подход**  Code-First при които

1)- създаваме моделите.Това включва

-класове на данните(EntityClass)

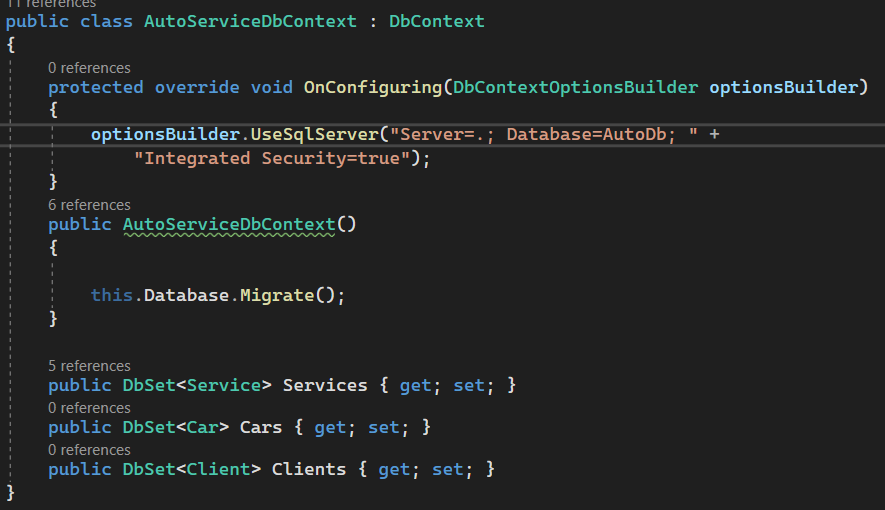
-DbContext класа(с DbSet)

2)-инсталиране на провайдъри(nugget packget)

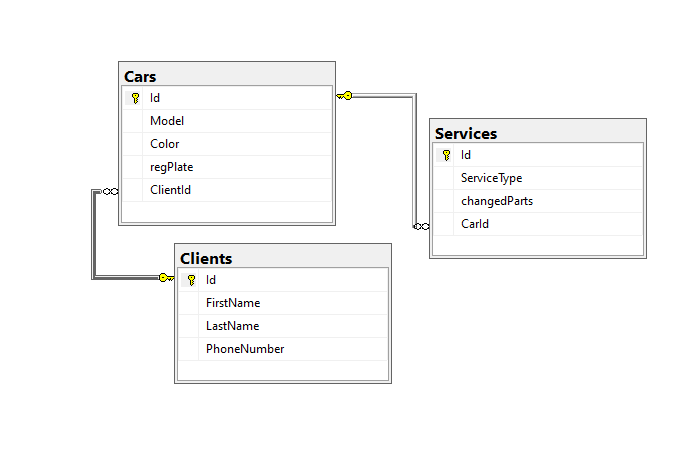
3)Създаване на миграция

4) В главната програма се създава обект db

5) Прилагат се CRUD операции за достъп и модифициране на БД

Снимка на **DbContext**-а **:**

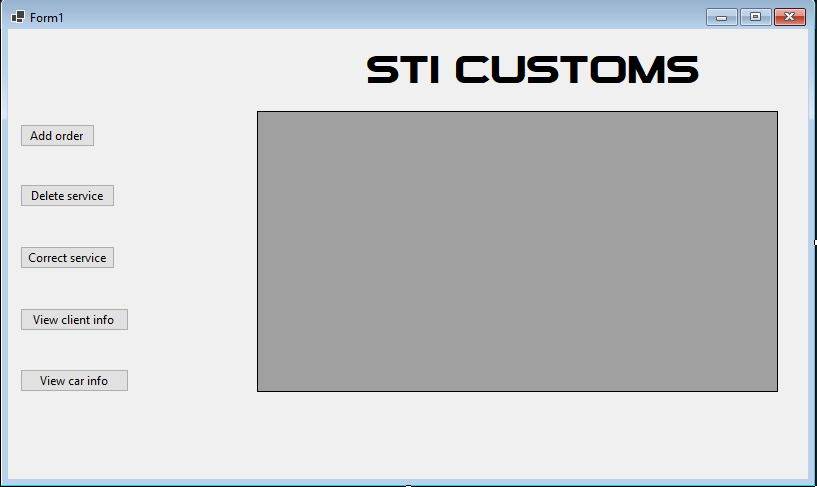
**Снимка на структура на БД с взръзките между таблиците:**



**1 кола към много услуги 1:M**

**1 кола има един клиент 1:1**

**4.2 Потребителски интерфейс:**

****

-бутон Add order

Отваряме нова форма в която въвеждаме данни за нова поръчка

-бутон Delete service

Изтриваме завършена услуга.

-бутон Correct service

Обновяваме вече съществуващ запис на услуга.

-бутон View client info

Четем вече съществуващи записи на клиент.

-бутон View car info

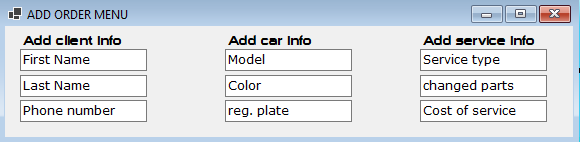
Четем вече съществуващи записи на кола.

-Data Grind View

За да гледаме постоянно записите в таблица Service

-Label “STI CUSTOMS”

Декорация към формичката



Това е формата където попълваме информация предоставена от клиента.

Под Label ”Add client info” добавяме първото, последното и тел. номер на клиента.

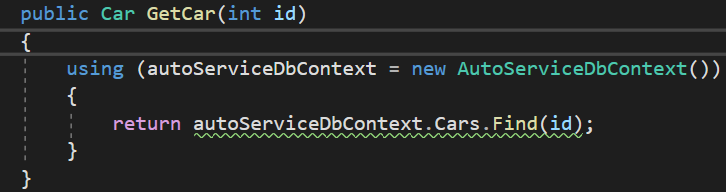
Под Label ”Add car info” добавяме модел, цвят и регистрационен номер на автомобила.

Под Label ”Add service info” добавяме вид на услуга, сменени части и цена на услугата.

**4.3Бизнес слой:**

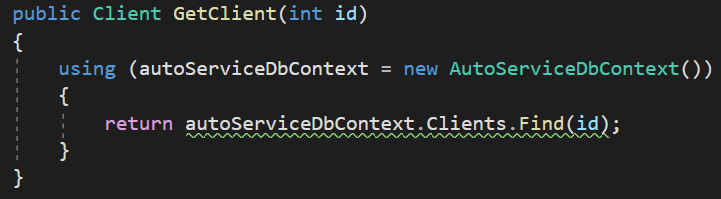
В бизнес слоя се намират методите за реализация на CRUD операции, които са:

-GetCar



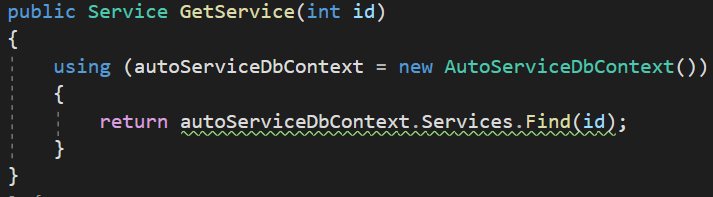
Прочитаме(Read) един запис от таблицата Cars.

-GetClient



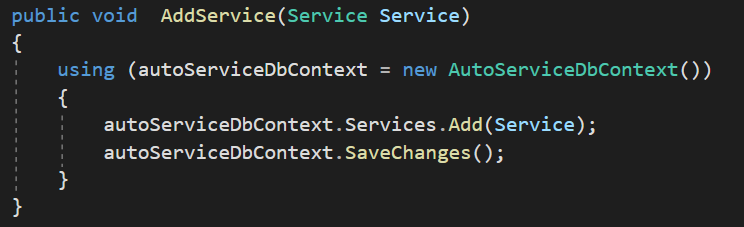
Прочитаме(Reaд) един запис от таблицата Clients.

-GetService



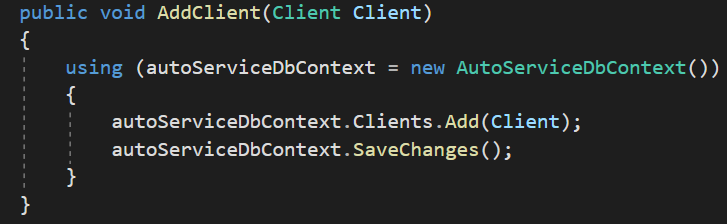
Прочитаме(Read) само един запис от таблицата Services.

-AddService



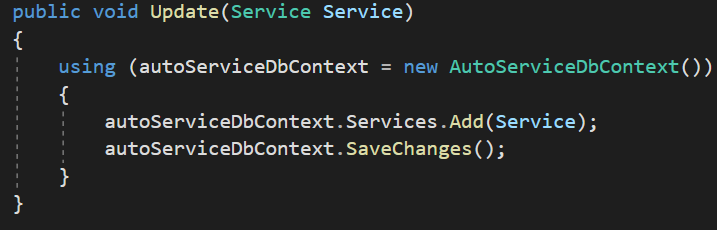
Добавяме(Create) услуга в таблицата Services.

-AddClient



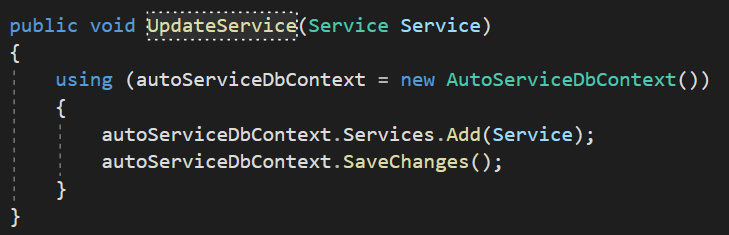
Добавяме(Create) клиент в таблицата Clients.

-AddCar



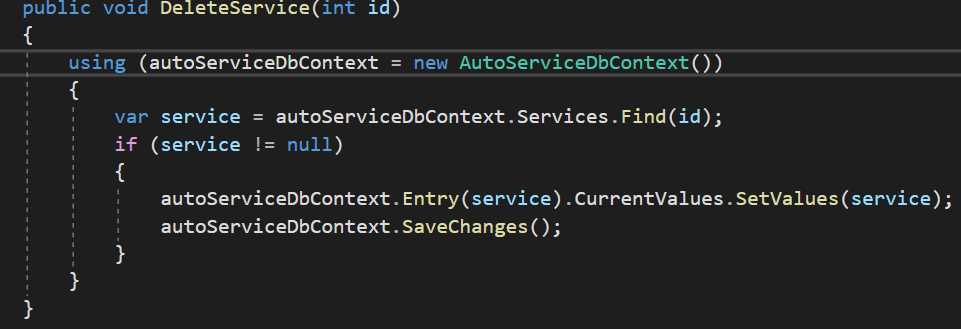
Добавяме(Create) кола в таблицата Cars.

-UpdateService



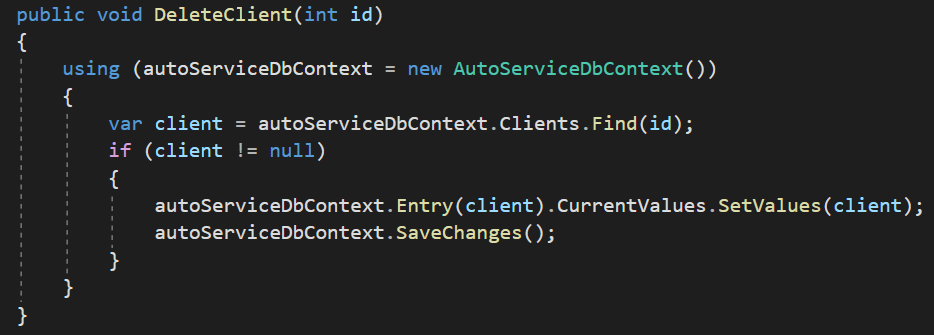
Обновяване(Update) на траблица Service.

-DeleteService



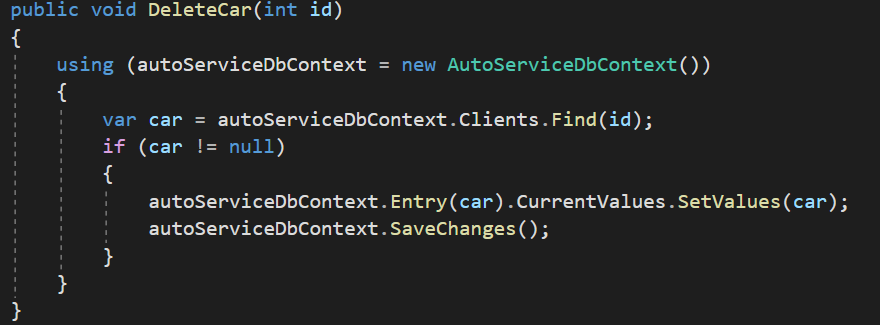
Изтриване(Delete) на запис от таблицата Services.

-DeleteClient



Изтриване(Delete) на запис от таблицата Clients.

-DeleteCar



Изтриване(Delete) на запис от таблицата Cars.

**5.Заключение:**

Приложението има за цел да подобри ефективността по счетоводната част на сервизите. Улеснение както за фирмата така и за клиента при следене и записване на направените услуги по даден автомобил.

**5.1 Развитието на приложението и неговото усъвършенстване**

Следващата цел на приложението би била да се разпространи в интернет под формата на web приложение и чрез „уникален код” или чрез направата на профил да следиш услугата която се извършва върху автомобила или до къде е работата по него. След като бъде доразработено клиента да следи как се развива услугата извършена по своя автомобил следващата цел ще бъде плащането онлайн.