

# **Documentație tehnică**

**Numele Proiectului: Il Motors Auto Sale**

**Numele echipei: Pro AutoTraders**

**Disciplina: Informatică industrială**

Profesor îndrumător :

Goța Dan

Membrii echipei:

Leșan Alexandru (team leader)

Mocan Angela Isabela (developer)

Lazea Radu-Nicușor (developer)

Toadere Diana-Sara (tester)

Mai 2021

# Cuprins

|   |    |
|---|----|
| Introducere .....                               | 3  |
| Enunțul Problemei.....                          | 3  |
| Prezentarea modului de lucru al echipei .....   | 4  |
| Obiectivele proiectului .....                   | 5  |
| Funcțiile aplicației .....                      | 5  |
| Implementarea aplicației .....                  | 19 |
| Models .....                                    | 20 |
| Controllers.....                                | 24 |
| Views.....                                      | 29 |
| Testarea aplicației.....                        | 31 |
| Arhitectura Aplicației .....                    | 32 |
| Reprezentarea bazei de date in DBDesigner ..... | 32 |
| Diagrama UML.....                               | 33 |
| Diagrama Use Case.....                          | 35 |

## Introducere

### Enunțul Problemei

Echipa noastră a realizat o aplicație web, care în practica poate fi folosit ca aplicația unui dealer auto. Ceea ce noi ne-am propus să realizăm este:

- a) pagina principală a magazinului
- b) pagină pentru afișarea tuturor automobilelor, dar și alte pagini pentru afișarea mașinilor după categoria căreia îi aparține
- c) posibilitatea de adăugare și ștergere a mașinilor dintr-un coș de cumpărături
- d) înregistrarea comenzii după ce utilizatorul completează un formular cu datele sale personale și urmează a fi contactat ulterior

## Prezentarea modului de lucru al echipei

După alegerea temei proiectului, am discutat împreună și am stabilit task-urile săptămânale pe care ar trebui să le îndeplinim pentru a reuși să terminăm la timp tot proiectul.

Team leader-ul împreună cu developerii echipei s-au ocupat de implementarea codului și rularea acestuia. Testerul după fiecare implementare a acestora s-a asigurat că proiectul, dar și task-urile săptămânale propuse sunt îndeplinite, acest lucru realizându-se prin corectarea erorilor survenite la nivelul codului.

## Obiectivele proiectului

În acest proiect se urmărește crearea unei aplicații cu ajutorul căreia să se administreze activitatea într-un magazin de mașini.

## Funcțiile aplicației

Aplicația are ca scop prezentarea mai multor mașini, împărțite în 2 categorii principale: „Electric Cars” și „Gasoline Cars”. Mașinile (ce au stocul pozitiv) sunt afișate în diferite moduri: pe pagina principală sunt afișate mașinile favorite (au proprietatea `IsFavourite` egală cu „TRUE”), pe pagina „All Cars” sunt afișate toate mașinile, iar pe paginile „Electric Cars” și „Gasoline Cars” sunt afișate mașinile în funcție de categoria din care face parte.

Mașinile, la randul lor, au mai multe proprietăți (stocate în baza de date). Fiecare mașină este prezentată într-un „card” în care este afișată imaginea sa, titlul, o descriere scurtă însoțită de categoria din care face parte, o descriere lungă, stocul, prețul și butonul „Add to Cart” (ne permite să adăugăm mașina respectivă în coșul de cumpărături).

Pe următoarea pagină este prezentată o captură de ecran a paginii principale în care sunt prezentate mașinile favorite, cu roșu sunt încercuite opțiunile din meniu pentru accesarea paginilor de afișare descrise anterior, iar cu portocaliu este încercuit „card-ul” de prezentare a mașinii în sine.



## About us:

The company was founded in 2010, thanks to the ideas of people passionate about cars, who wanted to bring in one place famous, verified brands with a 100% known history.

Most cars are new, but there are a few that have been driven less than 20,000 kilometers, which makes them look and work like new, but at a much more attractive price for customers.

The diversity of cars we have is very wide, trying to cover the needs of each customer, from electric cars at a very affordable price, such as Nissan Leaf, to Tesla Model S or Porsche Taycan, for those passionate about their famous acceleration, speed and all the technological capabilities they have. We have family cars like the Volvo S60. And the one we are most proud of can only be the symbol of masculinity, of power, a car that stands out regardless of the situation, the emblematic Mercedes-Benz G-Class. Among the largest companies we have connections with are the VAG group, Tesla and Mercedes-Benz.

You can cross our threshold in *Calea Turzii 172/174, Cluj-Napoca 400495, România*

Phone: +40 264 438 121

## These are our best sellers:



### Tesla Model S

Luxury Large [Electric Car](#)

The most recognizable models on the road and part of history for making electric cars viable.

**AVAILABLE:** 15 cars **Price:** 69.420,00 €

Add to Cart



### Porsche Taycan

Luxury Mid-Size [Electric Car](#)

It's the performance-car brand's first four-door sedan and first fully electric car.

**AVAILABLE:** 10 cars **Price:** 79.900,00 €

Add to Cart



### Tesla Model X

Large Hatchback [Electric Car](#)

A fully electric vehicle that's more of a crossover than a true SUV and based on the Tesla Model S.

**AVAILABLE:** 9 cars **Price:** 83.700,00 €

Add to Cart

## All cars

II Motors Auto Sale

[Home](#) [All Cars](#) [Electric Cars](#) [Gasoline Cars](#) [Shop Cart](#)



### All available cars:



#### Tesla Model S

Luxury Large [Electric Car](#)

The most recognizable models on the road and part of history for making electric cars viable.

**AVAILABLE:** 15 cars    **Price:** 69.420,00 €

[Add to Cart](#)



#### Nissan Leaf

Small Hatchback [Electric Car](#)

It was the first mass-market electric car to be sold in the United States.

**AVAILABLE:** 10 cars    **Price:** 14.000,00 €

[Add to Cart](#)



#### Porsche Taycan

Luxury Mid-Size [Electric Car](#)

It's the performance-car brand's first four-door sedan and first fully electric car.

**AVAILABLE:** 10 cars    **Price:** 79.900,00 €

[Add to Cart](#)



#### Jaguar I-Pace

Svelte Crossover [Electric Car](#)

Merges the powertrain technology of the future with everything expected in a prestige luxury.

**AVAILABLE:** 15 cars    **Price:** 71.490,00 €

[Add to Cart](#)



#### Audi E-Tron

SUV and Sportback [Electric Car](#)

All the comfort, luxury, and features of Audi's other premium models, with a fully electric twist.

**AVAILABLE:** 10 cars    **Price:** 65.900,00 €

[Add to Cart](#)



#### Tesla Model X

Large Hatchback [Electric Car](#)

A fully electric vehicle that's more of a crossover than a true SUV and based on the Tesla Model S.

**AVAILABLE:** 9 cars    **Price:** 83.700,00 €

[Add to Cart](#)



#### Ford Fiesta

Small Hatchback [Gasoline Car](#)



#### BMW 3-Series

Sport Small Sedan [Gasoline Car](#)



#### Mercedes C-Class

Luxury Small Sedan [Gasoline Car](#)



# Electric Cars

II Motors Auto Sale

[Home](#) [All Cars](#) [Electric Cars](#) [Gasoline Cars](#) [Shop Cart](#)



## All available electric cars:



### Tesla Model S

Luxury Large [Electric Car](#)

The most recognizable models on the road and part of history for making electric cars viable.

**AVAILABLE:** 15 cars **Price:** 69.420,00 €

[Add to Cart](#)



### Nissan Leaf

Small Hatchback [Electric Car](#)

It was the first mass-market electric car to be sold in the United States.

**AVAILABLE:** 10 cars **Price:** 14.000,00 €

[Add to Cart](#)



### Porsche Taycan

Luxury Mid-Size [Electric Car](#)

It's the performance-car brand's first four-door sedan and first fully electric car.

**AVAILABLE:** 10 cars **Price:** 79.900,00 €

[Add to Cart](#)



### Jaguar I-Pace

Svelte Crossover [Electric Car](#)

Merges the powertrain technology of the future with everything expected in a prestige luxury.

**AVAILABLE:** 15 cars **Price:** 71.490,00 €

[Add to Cart](#)



### Audi E-Tron

SUV and Sportback [Electric Car](#)

All the comfort, luxury, and features of Audi's other premium models, with a fully electric twist.

**AVAILABLE:** 10 cars **Price:** 65.900,00 €

[Add to Cart](#)



### Tesla Model X

Large Hatchback [Electric Car](#)

A fully electric vehicle that's more of a crossover than a true SUV and based on the Tesla Model S.

**AVAILABLE:** 9 cars **Price:** 83.700,00 €

[Add to Cart](#)

[Back to top](#)

© 2021 Company, Inc. · [Privacy](#) · [Terms](#)



# Gasoline Cars

II Motors Auto Sale

[Home](#) [All Cars](#) [Electric Cars](#) [Gasoline Cars](#) [Shop Cart](#)



## All available gasoline cars:



### Ford Fiesta

Small Hatchback [Gasoline Car](#)

A small car that struggled to find a crowd with low gas prices and rising competition.

**AVAILABLE:** 19 cars **Price:** 11.000,00 €

[Add to Cart](#)



### BMW 3-Series

Sport Small Sedan [Gasoline Car](#)

It's a compact luxury sedan that's been part of the automaker's lineup for more than 30 years.

**AVAILABLE:** 16 cars **Price:** 41.250,00 €

[Add to Cart](#)



### Mercedes C-Class

Luxury Small Sedan [Gasoline Car](#)

The 2021 C-Class should be at the top of your compact luxury sedan shopping list.

**AVAILABLE:** 13 cars **Price:** 41.600,00 €

[Add to Cart](#)



### Volvo S60

Mid-Size [Gasoline Car](#)

It's a mid-size sporty luxury sedan that directly competes with a trio of German models.

**AVAILABLE:** 12 cars **Price:** 38.950,00 €

[Add to Cart](#)



### Kia K5

Mid-Size [Gasoline Car](#)

Has all we'd want in a mid-size sedan, and it comes with a higher grade of materials.

**AVAILABLE:** 8 cars **Price:** 23.590,00 €

[Add to Cart](#)



### Porsche 911

Best Coupe [Gasoline Car](#)

The Porsche 911 is a two-seat sports car that's instantly recognizable as one of the world's best.

**AVAILABLE:** 9 cars **Price:** 99.200,00 €

[Add to Cart](#)




După alegerea modelului pe care ne-am dori sa-l achiziționăm, îl putem adăuga în coș prin apăsarea butonului „Add to Cart”.

Exemplu: Ne dorim să cumpărăm mașina „Porsche Taycan”

II Motors Auto Sale
Home AllCars Electric Cars Gasoline Cars Shop Cart
Search


All cars:



**Tesla Model S**  
Luxury Large Electric Car  
The most recognizable models on the road and part of history for making electric cars viable.

AVAILABLE: 19 cars Price: 69.420,00 €


Add to Cart



**Nissan Leaf**  
Small Hatchback Electric Car  
It was the first mass-market electric car to be sold in the United States.

AVAILABLE: 10 cars Price: 14.000,00 €

Add to Cart



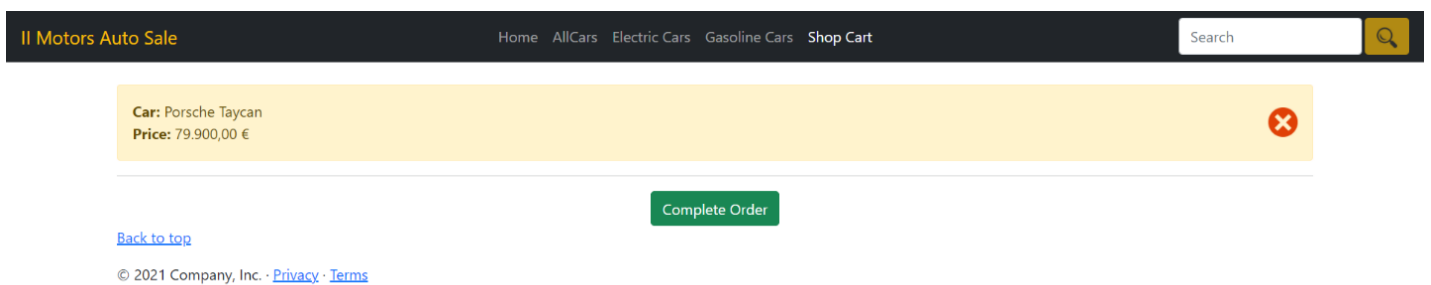
**Porsche Taycan**  
Luxury Mid-Size Electric Car  
It's the performance-car brand's first four-door sedan and first fully electric car.

AVAILABLE: 16 cars Price: 79.900,00 €

Add to Cart

|  | Id | Name                  | ShortDesc          | LongDesc             | Img                  | Price  | IsFavour... | Stock | Category... |
|--|----|-----------------------|--------------------|----------------------|----------------------|--------|-------------|-------|-------------|
|  | 1  | Tesla Model S         | Luxury Large       | The most recog...    | /img/ModelS.jpg      | 69420  | True        | 15    | 1           |
|  | 2  | Nissan Leaf           | Small Hatchback    | It was the first ... | /img/Leaf.jpg        | 14000  | False       | 10    | 1           |
|  | 3  | Porsche Taycan        | Luxury Mid-Size    | It's the perform...  | /img/Taycan.jpg      | 79900  | True        | 16    | 1           |
|  | 4  | Jaguar I-Pace         | Svelte Crossover   | Merges the po...     | /img/IPace.jpg       | 71490  | False       | 15    | 1           |
|  | 5  | Audi E-Tron           | SUV and Sportb...  | All the comfort...   | /img/ETron.jpg       | 65900  | False       | 10    | 1           |
|  | 6  | Tesla Model X         | Large Hatchback    | A fully electric ... | /img/ModelX.jpg      | 83700  | True        | 9     | 1           |
|  | 7  | Ford Fiesta           | Small Hatchback    | A small car that...  | /img/Fiesta.jpg      | 11000  | False       | 19    | 2           |
|  | 8  | BMW 3-Series          | Sport Small Sed... | It's a compact l...  | /img/M3.jpg          | 41250  | True        | 16    | 2           |
|  | 9  | Mercedes C-Class      | Luxury Small Se... | The 2021 C-Class...  | /img/CClass.jpg      | 41600  | False       | 13    | 2           |
|  | 10 | Volvo S60             | Mid-Size           | It's a mid-size s... | /img/S60.jpg         | 38950  | False       | 12    | 2           |
|  | 11 | Kia K5                | Mid-Size           | Has all we'd wa...   | /img/K5.jpg          | 23590  | False       | 8     | 2           |
|  | 12 | Porsche 911           | Best Coupe         | The Porsche 91...    | /img/911.jpg         | 99200  | True        | 9     | 2           |
|  | 13 | Audi A7               | Mid-Size Hatch...  | A car with a be...   | /img/A7.jpg          | 69200  | True        | 3     | 2           |
|  | 14 | Mercedes-Benz G-Class | Luxury Large SUV   | The G-Class SU...    | /img/GClass.jpg      | 131400 | True        | 14    | 2           |
|  | 15 | Toyota Land Cruiser   | Luxury Large SUV   | A luxury four-w...   | /img/LandCruiser.jpg | 85665  | True        | 6     | 2           |
|  | 16 | NULL                  | NULL               | NULL                 | NULL                 | NULL   | NULL        | NULL  | NULL        |

După cum putem observa, stocul afișat pe site corespunde cu cel din tabela Car a bazei de date. Avem un stoc disponibil de 16 mașini la modelul Porsche Taycan. După ce apăsăm butonul “Add to Cart” vom fi redirecționați în coșul de cumpărături în care a fost adăugată mașina respectivă.



În același timp, în baza de date, în tabela ShopCartItem se va înregistra mașina adăugată în coșul de cumpărături:

|   | Id   | carId | price | ShopCartId                           |
|---|------|-------|-------|--------------------------------------|
| ► | 5    | 3     | 79900 | 9032d233-6eb4-41c6-853c-f678eecada91 |
| * | NULL | NULL  | NULL  | NULL                                 |

După ce am adăugat în coșul de cumpărături toate mașinile pe care am dori să le achiziționăm, pentru a trece la pasul următor trebuie să apăsăm pe butonul „Complete Order”.



## Complete Order

Please enter your details in the given form:

Name:

► Validation rules

Surname:

► Validation rules

Complete address:

► Recommended format and validation rules

Phone number:

► Requested format: +(country code)-(phone)

Email:

Example: firstname.surname@domain.tld

Place Order

[Back to top](#)

Clientul trebuie să completeze formularul cu datele personale și să plaseze comanda prin apăsarea butonului “Place Order”.

Dacă completăm formularul respectând cerințele fiecărui câmp, vom fi redirecționați pe o pagina ce ne indică printr-un mesaj că formularul și comanda a fost procesată cu succes:



Your order has been successfully processed. We will contact you soon to establish the billing data, payment method & delivery process of your order.

[Back to top](#)

© 2021 Company, Inc. · [Privacy](#) · [Terms](#)

Tot odată observam faptul că comanda a fost plasată și înregistrată cu succes în tabela „Order” din baza de date.

|   | id   | name           | surname | adress           | phone         | email         | orderTime                   |
|---|------|----------------|---------|------------------|---------------|---------------|-----------------------------|
| ► | 1    | Angela-Isab... | Mocan   | str. Fantanel... | +40-747034... | isabelamoc... | 2021-05-29 12:50:47.7864349 |
| * | NULL | NULL           | NULL    | NULL             | NULL          | NULL          | NULL                        |

De asemenea, în tabela „OrderDetailau fost înregistrate detaliile comenzii, id-ul comenzii respective din tabela „Order”, id-ul mașinii comandate si prețul acesteia, extrase din tabela „Car”.

|    | id   | orderId | CarId | price |
|----|------|---------|-------|-------|
|    | 1    | 1       | 3     | 79900 |
| ▶* | NULL | NULL    | NULL  | NULL  |

Dacă ne întoarcem pe pagina principală și verificăm stocul disponibil al mașinii „Porsche Taycan”, observăm că acum avem ramase 15 mașini. De asemenea s-a făcut update pentru Stock și în baza de date, în tabela „Car”.

## All cars:



### Tesla Model S

Luxury Large [Electric Car](#)

The most recognizable models on the road and part of history for making electric cars viable.

AVAILABLE: 19 cars Price: 69.420,00 €

Add to Cart



### Nissan Leaf

Small Hatchback [Electric Car](#)

It was the first mass-market electric car to be sold in the United States.

AVAILABLE: 10 cars Price: 14.000,00 €

Add to Cart



### Porsche Taycan

Luxury Mid-Size [Electric Car](#)

It's the performance-car brand's first four-door sedan and first fully electric car.

AVAILABLE: 15 cars Price: 79.900,00 €

Add to Cart



|  | Id | Name           | ShortDesc       | LongDesc             | Img             | Price | IsFavour... | Stock | Category... |
|--|----|----------------|-----------------|----------------------|-----------------|-------|-------------|-------|-------------|
|  | 1  | Tesla Model S  | Luxury Large    | The most recog...    | /img/ModelS.jpg | 69420 | True        | 15    | 1           |
|  | 2  | Nissan Leaf    | Small Hatchback | It was the first ... | /img/Leaf.jpg   | 14000 | False       | 10    | 1           |
|  | 3  | Porsche Taycan | Luxury Mid-Size | It's the perform...  | /img/Taycan.jpg | 79900 | True        | 15    | 1           |

## Funcția Search

În bara de navigare a aplicației web, avem implementată funcția de căutare printre mașinile disponibile pe site. Partea de front-end a funcției este alcătuită dintr-un ansamblu(html form) ce conține un input de tip text și un buton cu imaginea unei lupe.

Cât timp în input-ul este gol, butonul este dezactivat, iar starea lui se modifică de fiecare dată când introducem sau ștergem un caracter din input, această funcționalitate este implementată în JavaScript:

```
const searchInput = document.querySelector("#searchInput");
const searchBtn = document.querySelector("#searchBtn");

searchBtn.disabled = true; //setting button state to disabled


searchInput.addEventListener("input", stateHandle);

function stateHandle() {
  if (searchInput.value === "") {
    searchBtn.disabled = true; //button remains disabled
  } else {
    searchBtn.disabled = false; //button is enabled
  }
}
```

Odată introdus numele mașinii pe care dorim să o căutam, programul caută printre mașinile prezente în tabela Car și le afișează pe toate (chiar și pe cele ce au stocul egal cu 0, ca clientul să ne poată contacta pentru a afla când va fi din nou disponibilă acea mașină) ce conțin șirul de caractere introdus în input în numele său.




### Cars available based on your search:



**Tesla Model S**  
Luxury Large [Electric Car](#)  
The most recognizable models on the road and part of history for making electric cars viable.

AVAILABLE: 19 cars Price: 69.420,00 €

Add to Cart



**Tesla Model X**  
Large Hatchback [Electric Car](#)  
A fully electric vehicle that's more of a crossover than a true SUV and based on the Tesla Model S.

AVAILABLE: 11 cars Price: 83.700,00 €

Add to Cart

[Back to top](#)

© 2021 Company, Inc. · [Privacy](#) · [Terms](#)

Dacă nu există o astfel de mașină în magazin va apărea o pagina ce afișează următorul mesaj:

### No cars were found based on your search.

[Back to top](#)

© 2021 Company, Inc. · [Privacy](#) · [Terms](#)

```

public IActionResult Search(string searchWord) {
    return View("Index", _allCars.Cars.Where(c => c.Name.Contains(searchWord, StringComparison.InvariantCultureIgnoreCase)));
}

```

\*Metoda prin intermediul căreia sunt returnate mașinile rezultate în urma căutării după un anumit șir de caractere.

Magazinul nostru furnizează o serie de termeni și condiții acestea pot fi descoperite prin apăsarea cuvântului „Terms” din partea de jos a paginii.

### Luxury Large SUV Gasoline Car

A luxury four-wheel drive vehicle produced by the Japanese automobile manufacturer Toyota.

**AVAILABLE:** 6 cars

**Price:** 85.665,00 €

Add to Cart

[Back to top](#)

© 2021 Company, Inc. · [Privacy](#) · [Terms](#)

## Va apărea următoarea interfață:

II Motors Auto Sale

[Home](#) [AllCars](#) [Electric Cars](#) [Gasoline Cars](#) [Shop Cart](#)

Search



### Terms and Conditions

Welcome to II Motors Auto Sale!

These terms and conditions outline the rules and regulations for the use of II Motors Auto Sale's Website, located at <https://localhost:44371/>.

By accessing this website we assume you accept these terms and conditions. Do not continue to use II Motors Auto Sale if you do not agree to take all of the terms and conditions stated on this page.

The following terminology applies to these Terms and Conditions, Privacy Statement and Disclaimer Notice and all Agreements: "Client", "You" and "Your" refers to you, the person log on this website and compliant to the Company's terms and conditions. "The Company", "Ourselves", "We", "Our" and "Us", refers to our Company. "Party", "Parties", or "Us", refers to both the Client and ourselves. All terms refer to the offer, acceptance and consideration of payment necessary to undertake the process of our assistance to the Client in the most appropriate manner for the express purpose of meeting the Client's needs in respect of provision of the Company's stated services, in accordance with and subject to, prevailing law of Netherlands. Any use of the above terminology or other words in the singular, plural, capitalization and/or he/she or they, are taken as interchangeable and therefore as referring to same.

### Cookies

We employ the use of cookies. By accessing II Motors Auto Sale, you agreed to use cookies in agreement with the II Motors Auto Sale's Privacy Policy.

Most interactive websites use cookies to let us retrieve the user's details for each visit. Cookies are used by our website to enable the functionality of certain areas to make it easier for people visiting our website. Some of our affiliate/advertising partners may also use cookies.

### License

Unless otherwise stated, II Motors Auto Sale and/or its licensors own the intellectual property rights for all material on II Motors Auto Sale. All intellectual property rights are reserved. You may access this from II Motors Auto Sale for your own personal use subjected to restrictions set in these terms and conditions.

You must not:

- Republish material from II Motors Auto Sale

Magazinul are și o politică de confidențialitate disponibilă de asemenea printr-un click pe cuvântul „Privacy” din partea de jos a paginii.

[Back to top](#)

© 2021 Company, Inc. · [Privacy](#) · [Terms](#)

La apăsarea butonului vom fi redirecționați pe următoarea pagină:

**II Motors Auto Sale**[Home](#)[AllCars](#)[Electric Cars](#)[Gasoline Cars](#)[Shop Cart](#)

## Website Privacy, Policies & Notices

II Motors Auto Sale Privacy Policy

This Privacy Policy describes how your personal information is collected, used, and shared when you visit or make a purchase from <https://localhost:44371/>.

### PERSONAL INFORMATION WE COLLECT

When you visit the Site, we automatically collect certain information about your device, including information about your web browser, IP address, time zone, and some of the cookies that are installed on your device. Additionally, as you browse the Site, we collect information about the individual web pages or products that you view, what websites or search terms referred you to the Site, and information about how you interact with the Site. We refer to this automatically-collected information as “Device Information.”

We collect Device Information using the following technologies:

- “Cookies” are data files that are placed on your device or computer and often include an anonymous unique identifier. For more information about cookies, and how to disable cookies, visit <http://www.allaboutcookies.org>.
- “Log files” track actions occurring on the Site, and collect data including your IP address, browser type, Internet service provider, referring/exit pages, and date/time stamps.
- “Web beacons,” “tags,” and “pixels” are electronic files used to record information about how you browse the Site.

Additionally when you make a purchase or attempt to make a purchase through the Site, we collect certain information from you, including your name, surname, billing address, shipping address, payment information including credit card numbers etc., email address, and phone number. We refer to this information as “Order Information.”

When we talk about “Personal Information” in this Privacy Policy, we are talking both about Device Information and Order Information.

### HOW DO WE USE YOUR PERSONAL INFORMATION?

We use the Order Information that we collect generally to fulfill any orders placed through the Site (including processing your payment information, arranging for shipping, and providing you with invoices and/or order confirmations). Additionally, we use this Order Information to: Communicate with you; Screen our orders for potential risk or fraud; and When in line with the preferences you have shared with us, provide you with information or advertising relating to our products or services. We use the Device Information that we collect to help us screen for potential risk and fraud (in particular, your IP address), and more generally to improve and optimize our Site (for example, by generating analytics about how our customers browse and interact with the Site, and to assess the success of our marketing and advertising campaigns).

Dacă ne dorim să ajungem doar printr-un “click” la partea superioară a paginii apăsăm butonul “Back to top”.

A luxury four-wheel drive vehicle produced by the Japanese automobile manufacturer Toyota.

**AVAILABLE:** 6 cars      **Price:** 85.665,00 €

Add to Cart

[Back to top](#)

© 2021 Company, Inc. · [Privacy](#) · [Terms](#)

Această funcționalitate este realizată prin intermediul HTML, folosind atributul href="#" ce permite navigarea în partea de sus a paginii curente.

## Implementarea aplicației

În primă instanță am ales tema proiectului am proiectat baza de date de comun acord cu echipa. La implementare au participat Isabela cu Alexandru, Diana a realizat documentația tehnică și a interfețelor grafice, testarea întregii aplicații și Radu a realizat documentația pentru diagramele UML, diagrama bazei de date și diagrama Use Case. De asemenea, team leaderul a participat la redactarea finală și îmbunătățirea documentației tehnice și crearea prezentării.

După ce am ales tema proiectului ne-am documentat și am căutat surse care ne-ar putea ajuta să realizăm proiectul, astfel primele săptămâni le-am dedicat învățării și studierii.

În urma studiilor de caz realizate am decis că vom lucra cu baza de date prin intermediul EntityFrameworkCore și o vom crea cu ajutorul migrațiilor.

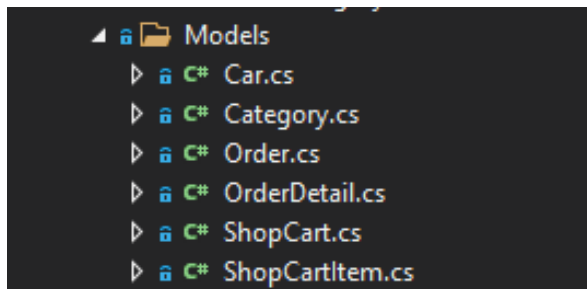
Pentru asta am instalat următoarele plugin-uri în proiect:

- Microsoft.AspNetCore.StaticFiles => pentru a lucra cu fișierele statice(imagini...)
- Microsoft.AspNetCore.Mvc => pentru a lucra cu MVC
- (Model–view–controller)
- Microsoft.EntityFrameworkCore => pentru baza de date
- Microsoft.EntityFrameworkCore.SqlServer => furnizorul de baze de date Microsoft SQL Server pentru EF
- Microsoft.EntityFrameworkCore.Tools => instrumentele necesare pentru a ne putea folosi de comenzile EF-ului
- JSON => pentru a lucra cu datele primite din baza de date

Implementarea am inceput-o prin crearea modelelor necesare pentru crearea și generarea bazei de date, tabelelor și informațiilor necesare pentru EntityFramework.

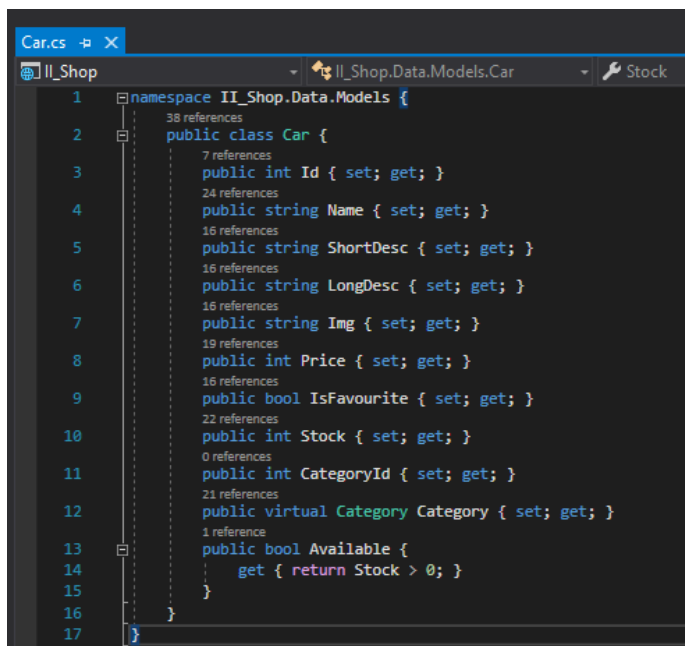
## Models

În dosarul “Models” avem declarate toate tabelele necesare bazei de date.



Iar în fiecare clasă avem declarate pe rand toate campurile necesare fiecărei tabele, fiecare avand un tip și proprietățile necesare.

### Car.cs:





Avem mai multe campuri (Id, Name, ShortDesc, LongDesc, Ing, Price, IsFavourite, Stock, CategoryId, Category ), respectiv avem și settere si gettere.

Category din interiorul Car.cs este setată ca fiind cheie străină.

Stock reprezintă numărul de mașini disponibile spre a fi cumpărate, în momentul în care voi comanda mașina se va realiza o modificare în baza de date.

Available este folosit pentru a ști dacă mașina trebuie afișată pe site sau nu, în momentul în care stocul este mai mare ca 0, acesta are atribuita valoarea booleana TRUE, însemnând că mașina este disponibilă. În cazul în care stocul este 0, campului Available i se atribuie valoarea FALSE.

\*!Celelalte clase sunt create pe baza acelorași principii.

### ShopCart.cs:

În această clasă realizăm toate operațiile ce le putem face la nivelul coșului de cumpărături: crearea coșului cu un id generat automat, afișarea coșului corespunzător pentru fiecare utilizator în funcție de sesiunea activă, adăugare și ștergerea elementelor (mașinilor) din coș, golirea coșului, și totodata funcțiile responsabile de decrementarea stocului mașinilor ce au fost cumpărate și actualizarea stocului în baza de date.

```
11 references  
public List<ShopCartItem> listShopItems { get; set; }
```

Declararea listei necesare pentru toate produsele care sunt adăugate în coș.

```

1 reference
public void AddToCart(Car car) {
    appDbContent.ShopCartItem.Add(new ShopCartItem {
        ShopCartId = ShopCartId,
        car = car,
        price = car.Price
    });
    appDbContent.SaveChanges();
}

1 reference
public void DeleteFromCart(ShopCartItem car) {
    appDbContent.ShopCartItem.Remove(getCarFromShopCart(car.Id));
    appDbContent.SaveChanges();
}

1 reference
public void DeleteAllFromCart() {
    appDbContent.ShopCartItem.RemoveRange(appDbContent.ShopCartItem.ToList());
    appDbContent.SaveChanges();
}

2 references
public void DecreaseStock() {
    var shopList = getShopItems();
    foreach(var item in shopList)
        item.car.Stock--;
}

1 reference
public void UpdateStock() {

```

Metodele prin care am reușit să realizăm operațiunile de adăugare, ștergere și actualizare a coșului. Prin aceste metode noi efectuăm modificări la nivelul bazei de date.

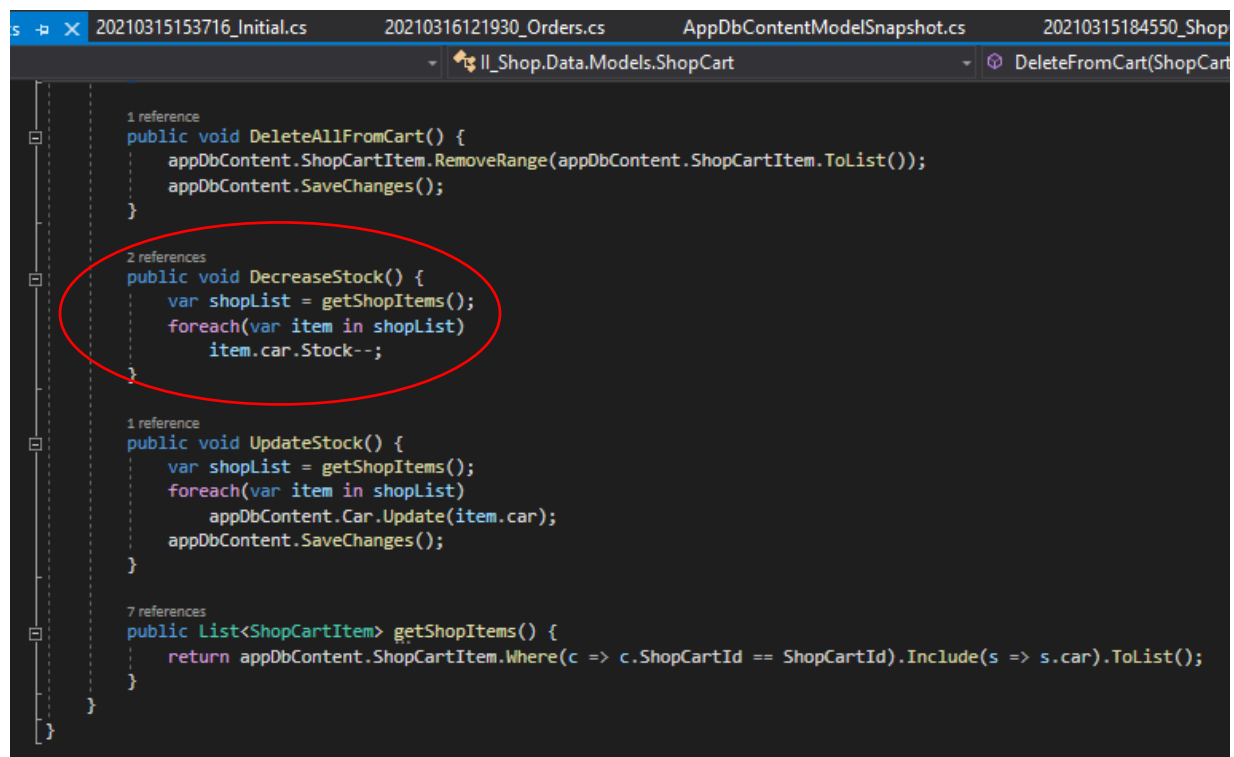
Prin metoda “.SaveChanges()” se salvează modificările în appDbcontent. În AppDbContent.cs moștenim proprietățile clasei DbContext, cu ajutorul căreia putem efectua diferite operații asupra bazei de date.

```

7 references
public List<ShopCartItem> getShopItems() {
    return appDbContent.ShopCartItem.Where(c => c.ShopCartId == ShopCartId).Include(s => s.car).ToList();
}

```

Aici avem o metoda de tip List<ShopCartItem> care ne va returna toate înregistrările din tabela ShopCartItem.



```

1 reference
public void DeleteAllFromCart() {
    appDbContent.ShopCartItem.RemoveRange(appDbContent.ShopCartItem.ToList());
    appDbContent.SaveChanges();
}

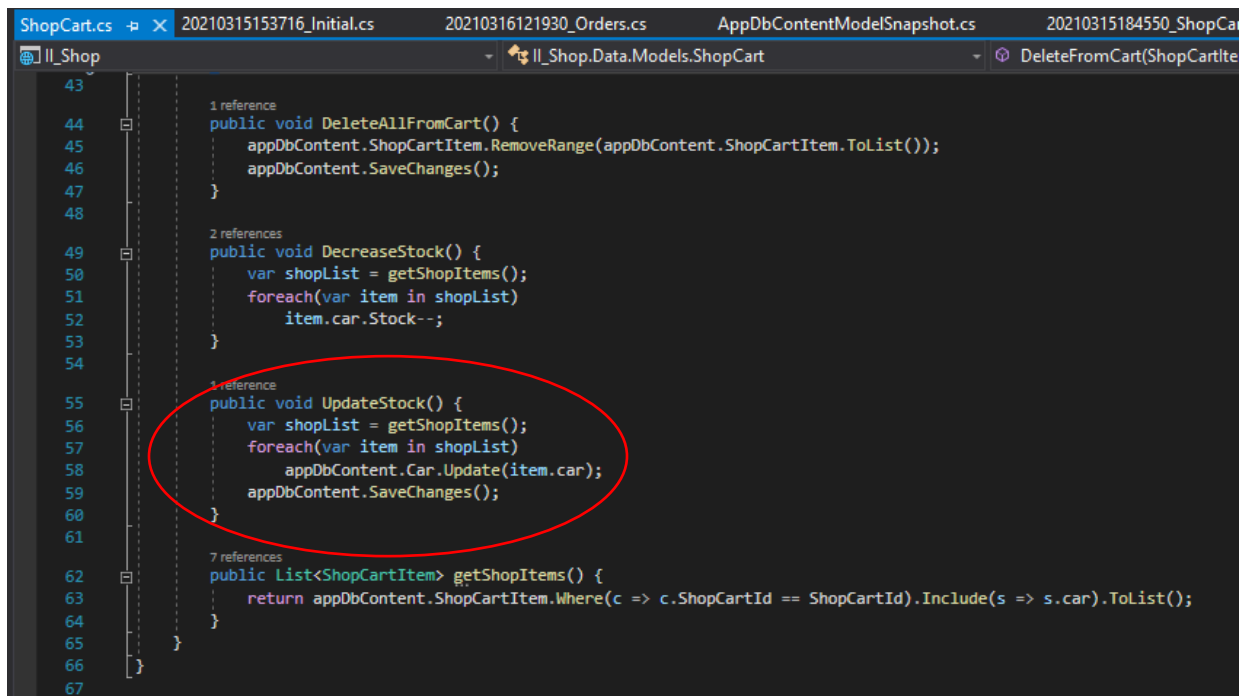
2 references
public void DecreaseStock() {
    var shopList = getShopItems();
    foreach(var item in shopList)
        item.car.Stock--;
}

1 reference
public void UpdateStock() {
    var shopList = getShopItems();
    foreach(var item in shopList)
        appDbContent.Car.Update(item.car);
    appDbContent.SaveChanges();
}

7 references
public List<ShopCartItem> getShopItems() {
    return appDbContent.ShopCartItem.Where(c => c.ShopCartId == ShopCartId).Include(s => s.car).ToList();
}

```

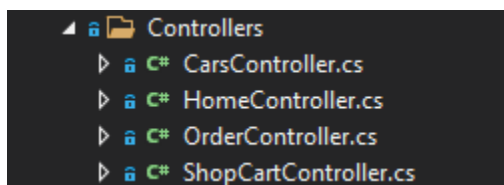
Aceasta metodă se va apela de fiecare dată când a fost adăugat un produs în coșul de cumpărături, având scopul de a duce evidența la stocul mașinii, pentru a nu se putea comanda mai multe mașini decât sunt disponibile.



```
43  
44 1 reference  
45 public void DeleteAllFromCart() {  
46     appDbContent.ShopCartItem.RemoveRange(appDbContent.ShopCartItem.ToList());  
47     appDbContent.SaveChanges();  
48 }  
49  
50 2 references  
51 public void DecreaseStock() {  
52     var shopList = getShopItems();  
53     foreach(var item in shopList)  
54         item.car.Stock--;  
55 }  
56  
57 1 reference  
58 public void UpdateStock() {  
59     var shopList = getShopItems();  
60     foreach(var item in shopList)  
61         appDbContent.Car.Update(item.car);  
62     appDbContent.SaveChanges();  
63 }  
64  
65 7 references  
66 public List<ShopCartItem> getShopItems() {  
67     return appDbContent.ShopCartItem.Where(c => c.ShopCartId == ShopCartId).Include(s => s.car).ToList();  
68 }  
69  
70 }
```

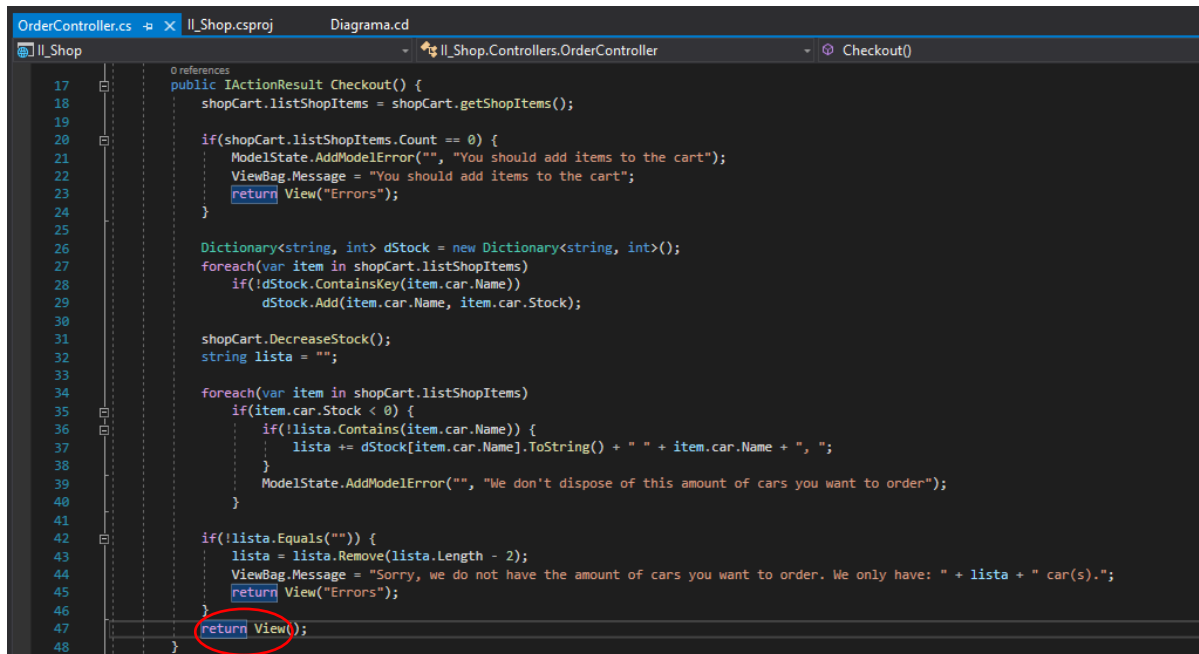
Se va apela doar după ce comanda a fost plasată astfel modificările efectuate asupra stocului vor fi salvate.

## Controllers



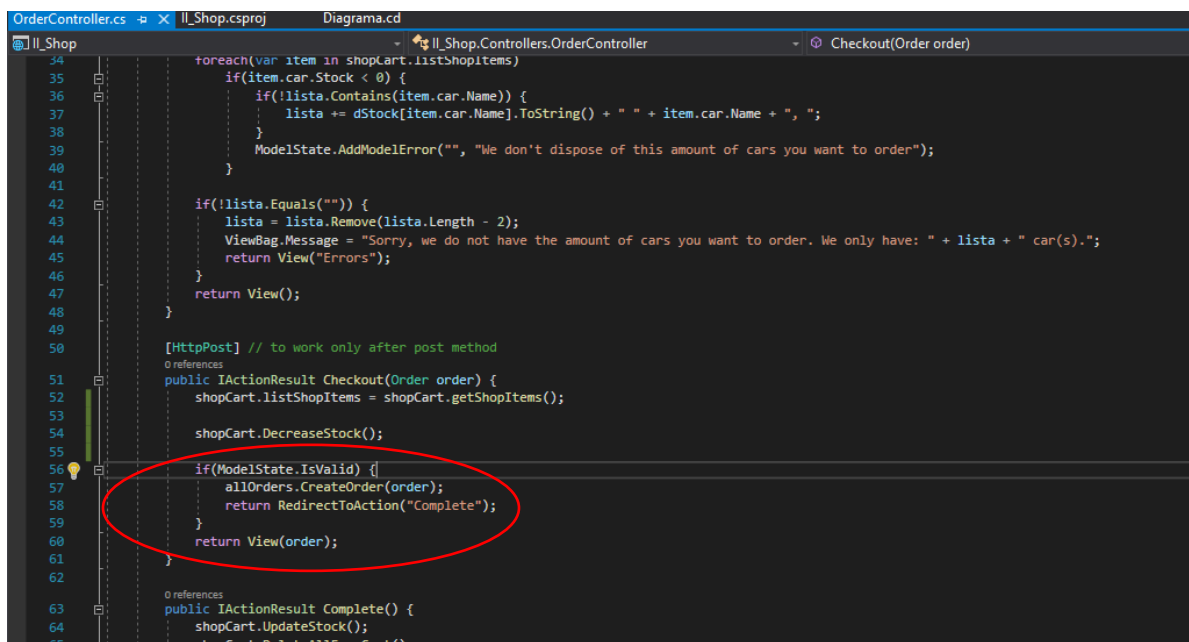
În dosarul „Controllers” avem create clasele ce sunt responsabile de returnarea unui View (fișier .cshtml ce returnează o pagină cu care se poate interactiona) către utilizator în funcție de requestul efectuat de acesta. Un controller poate avea una sau mai multe metode care la randul lor pot returna diferite views-uri. Pe scurt, un controller poate randa unul sau mai multe Views-uri.

## OrderController.cs



```
17 public IActionResult Checkout() {
18     shopCart.listShopItems = shopCart.getShopItems();
19
20     if(shopCart.listShopItems.Count == 0) {
21         ModelState.AddModelError("", "You should add items to the cart");
22         ViewBag.Message = "You should add items to the cart";
23         return View("Errors");
24     }
25
26     Dictionary<string, int> dStock = new Dictionary<string, int>();
27     foreach(var item in shopCart.listShopItems)
28         if(!dStock.ContainsKey(item.car.Name))
29             dStock.Add(item.car.Name, item.car.Stock);
30
31     shopCart.DecreaseStock();
32     string lista = "";
33
34     foreach(var item in shopCart.listShopItems)
35         if(item.car.Stock < 0) {
36             if(!lista.Contains(item.car.Name)) {
37                 lista += dStock[item.car.Name].ToString() + " " + item.car.Name + ", ";
38             }
39             ModelState.AddModelError("", "We don't dispose of this amount of cars you want to order");
40         }
41
42     if(!lista.Equals("")) {
43         lista = lista.Remove(lista.Length - 2);
44         ViewBag.Message = "Sorry, we do not have the amount of cars you want to order. We only have: " + lista + " car(s).";
45         return View("Errors");
46     }
47     return View();
48 }
```

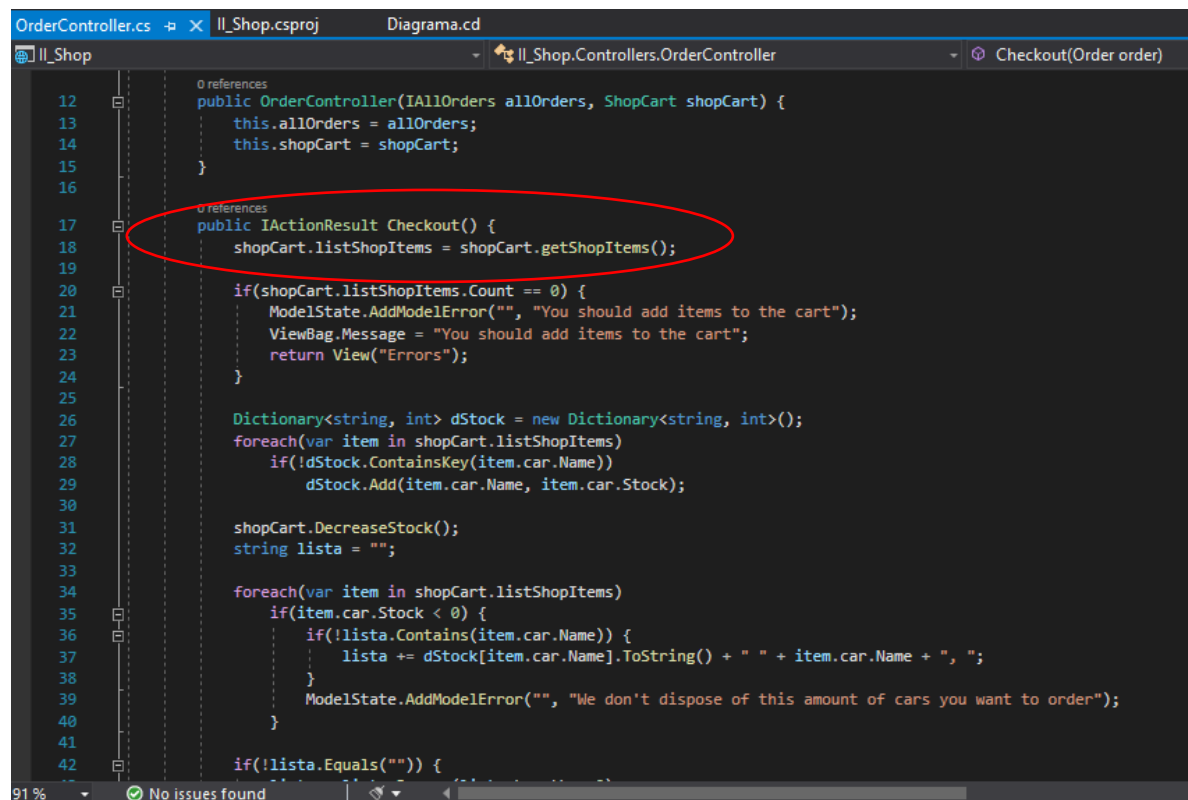
Metoda Checkout() trebuie sa returneze un View(). Am folosit IActionResult care returneaza o actiune, te redirectioneaza catre un View().



```
34 foreach(var item in shopCart.listShopItems)
35     if(item.car.Stock < 0) {
36         if(!lista.Contains(item.car.Name)) {
37             lista += dStock[item.car.Name].ToString() + " " + item.car.Name + ", ";
38         }
39         ModelState.AddModelError("", "We don't dispose of this amount of cars you want to order");
40     }
41
42     if(!lista.Equals("")) {
43         lista = lista.Remove(lista.Length - 2);
44         ViewBag.Message = "Sorry, we do not have the amount of cars you want to order. We only have: " + lista + " car(s).";
45         return View("Errors");
46     }
47     return View();
48
49 [HttpPost] // to work only after post method
50 public IActionResult Checkout(Order order) {
51     shopCart.listShopItems = shopCart.getShopItems();
52
53     shopCart.DecreaseStock();
54
55     if(ModelState.IsValid) {
56         allOrders.CreateOrder(order);
57         return RedirectToAction("Complete");
58     }
59     return View(order);
60 }
61
62 public IActionResult Complete() {
63     shopCart.UpdateStock();
64     shopCart.DeleteAllFromCart();
65 }
```

Daca modelul va fi valid mi se va crea comanda.

In caz contrar, modelul nu va mai fi valid, si voi primi mesajul din ViewBag.Message.



```
12 public OrderController(IAIOrders allOrders, ShopCart shopCart) {
13     this.allOrders = allOrders;
14     this.shopCart = shopCart;
15 }
16
17 public IActionResult Checkout() {
18     shopCart.listShopItems = shopCart.getShopItems();
19
20     if(shopCart.listShopItems.Count == 0) {
21         ModelState.AddModelError("", "You should add items to the cart");
22         ViewBag.Message = "You should add items to the cart";
23         return View("Errors");
24     }
25
26     Dictionary<string, int> dStock = new Dictionary<string, int>();
27     foreach(var item in shopCart.listShopItems)
28         if(!dStock.ContainsKey(item.car.Name))
29             dStock.Add(item.car.Name, item.car.Stock);
30
31     shopCart.DecreaseStock();
32     string lista = "";
33
34     foreach(var item in shopCart.listShopItems)
35         if(item.car.Stock < 0) {
36             if(!lista.Contains(item.car.Name)) {
37                 lista += dStock[item.car.Name].ToString() + " " + item.car.Name + ", ";
38             }
39             ModelState.AddModelError("", "We don't dispose of this amount of cars you want to order");
40         }
41
42     if(!lista.Equals("")) {
```

Dictionarul l-am facut pentru cazurile in care un client doreste sa comande un numar superior stocului nostru de masini, daca acesta dorește să achiziționeze mai multe modele de mașini, dintre care unele au stocul suficient iar altele nu, să se afișeze un mesaj în care este specificat care mașină nu are stocul suficient și numărul de elemente disponibile.



Exemplu: Vream să cumpărăm 5 Audi A7, dar in stock exista doar 4 masini.

## Audi A7

Mid-Size Hatchback Gasoline Car

A car with a beautiful sloped profile, that heaps luxury into a svelte shape like a great tuxedo.

**AVAILABLE: 4 cars** **Price: 69.200,00 €**

Add to Cart

II Motors Auto Sale

Home AllCars Electric Cars Gasoline Cars Shop Cart

Search

Car: Audi A7  
Price: 69.200,00 €

×

Car: Audi A7  
Price: 69.200,00 €

×

Car: Audi A7  
Price: 69.200,00 €

×

Car: Audi A7  
Price: 69.200,00 €

×

Car: Audi A7  
Price: 69.200,00 €

×

Complete Order

Activați Windows  
Accesați Setări pentru a activa Windows.

Acesta este mesajul pe care-l vom primi:

II Motors Auto Sale

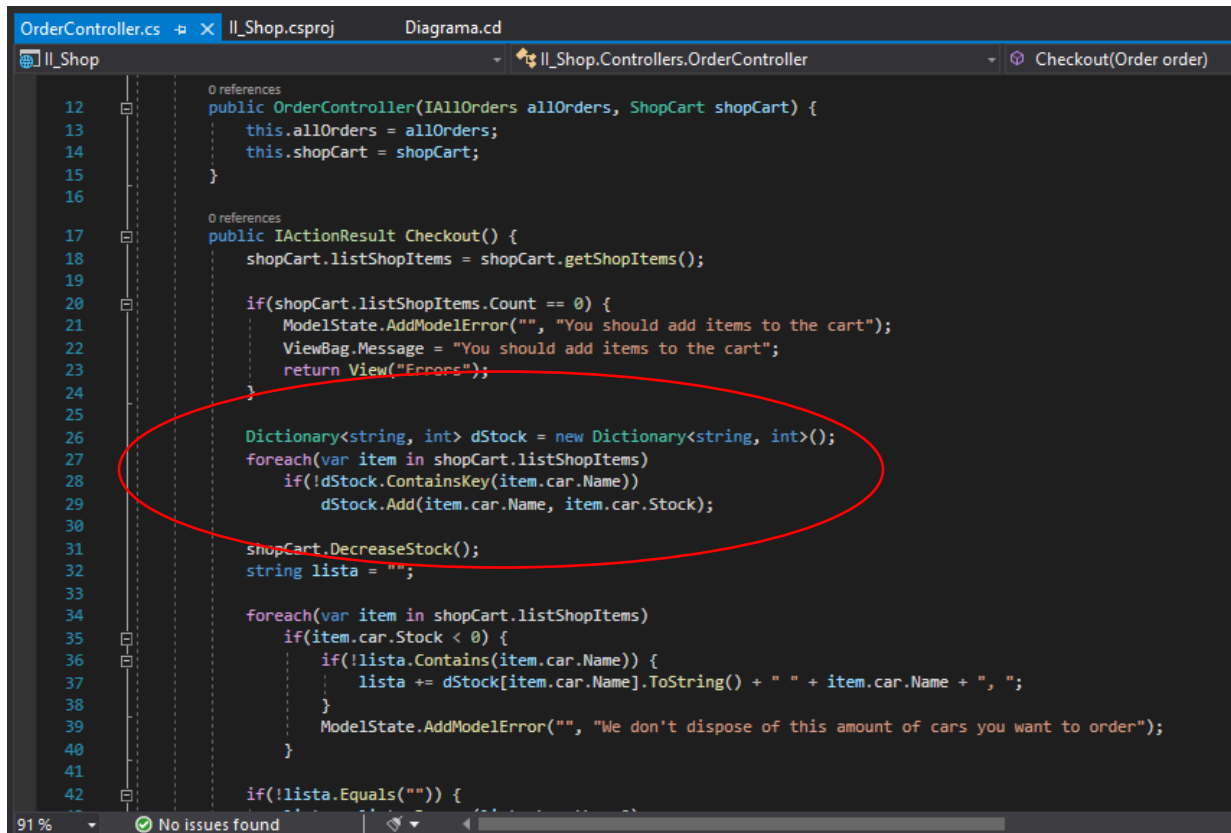
Home AllCars Electric Cars Gasoline Cars Shop Cart

Search

Sorry, we do not have the amount of cars you want to order. We only have: 4 Audi A7 car(s).

[Back to top](#)

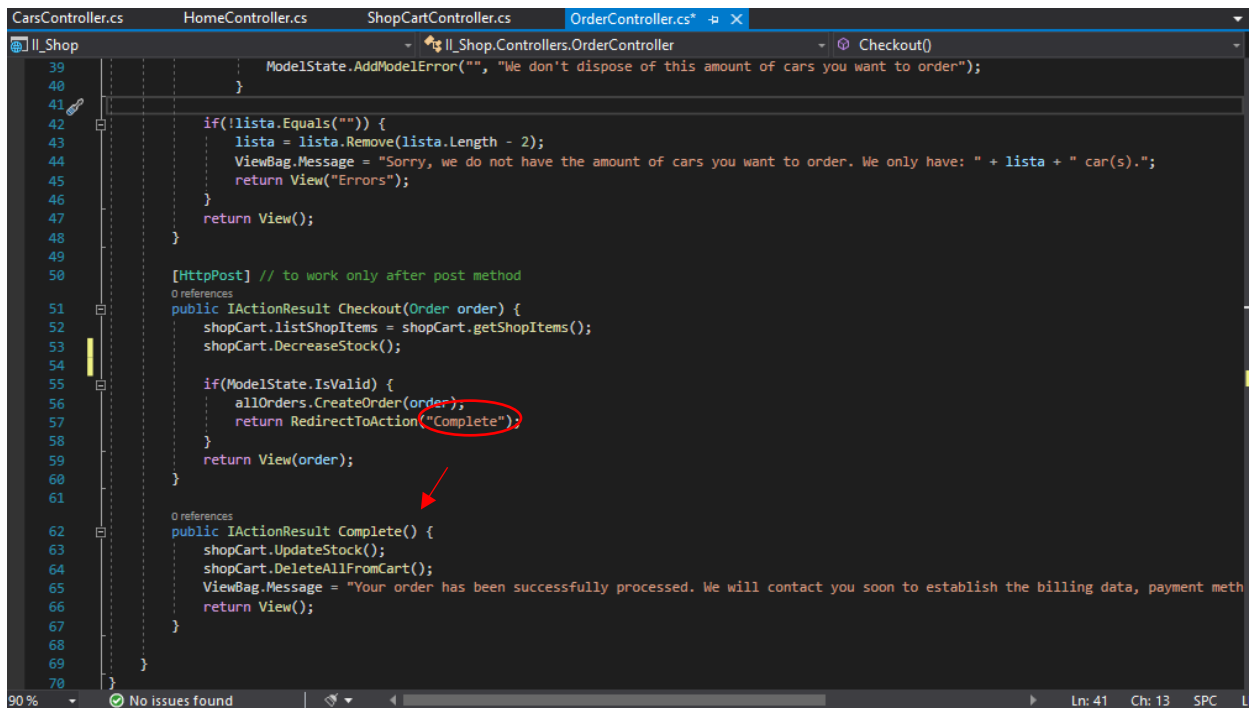
© 2021 Company, Inc. · [Privacy](#) · [Terms](#)



```
12 public OrderController(IAIAllOrders allOrders, ShopCart shopCart) {
13     this.allOrders = allOrders;
14     this.shopCart = shopCart;
15 }
16
17 public IActionResult Checkout() {
18     shopCart.listShopItems = shopCart.getShopItems();
19
20     if(shopCart.listShopItems.Count == 0) {
21         ModelState.AddModelError("", "You should add items to the cart");
22         ViewBag.Message = "You should add items to the cart";
23         return View("Errors");
24     }
25
26     Dictionary<string, int> dStock = new Dictionary<string, int>();
27     foreach(var item in shopCart.listShopItems)
28         if(!dStock.ContainsKey(item.car.Name))
29             dStock.Add(item.car.Name, item.car.Stock);
30
31     shopCart.DecreaseStock();
32     string lista = "";
33
34     foreach(var item in shopCart.listShopItems)
35         if(item.car.Stock < 0) {
36             if(!lista.Contains(item.car.Name)) {
37                 lista += dStock[item.car.Name].ToString() + " " + item.car.Name + ", ";
38             }
39             ModelState.AddModelError("", "We don't dispose of this amount of cars you want to order");
40         }
41
42     if(!lista.Equals("")) {
```

Am pus dicționarul înainte de `shopCart.DecreaseStock()` pentru ca în cazul în care l-am fi pus după, mi-ar fi scăzut deja din numărul de mașini de pe stoc și mi-ar fi afișat că am pe stoc -n mașini, n fiind egal cu numărul de mașini din stocul nostru – numărul de mașini dorite de cumpărător.

Dacă modelul e valid (avem item-uri în `shopCart` care au stocul suficient) ni se va crea comanda ce ne va redirecționa către acțiunea `Complete()`, unde stock-ul își va face update, `shopCart` se va goli pentru a se putea crea unul nou și ni se va afișa mesajul ce ne indică faptul că operațiunea s-a realizat cu succes.



```
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70

ModelState.AddModelError("", "We don't dispose of this amount of cars you want to order");
}

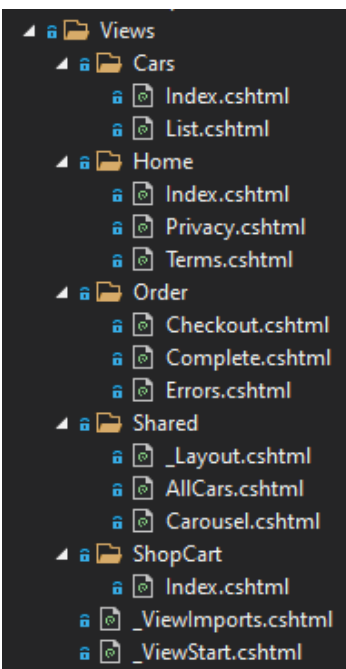
if(!lista.Equals("")) {
    lista = lista.Remove(lista.Length - 2);
    ViewBag.Message = "Sorry, we do not have the amount of cars you want to order. We only have: " + lista + " car(s).";
    return View("Errors");
}
return View();
}

[HttpPost] // to work only after post method
public IActionResult Checkout(Order order) {
    shopCart.ListShopItems = shopCart.getShopItems();
    shopCart.DecreaseStock();

    if(ModelState.IsValid) {
        allOrders.CreateOrder(order);
        return RedirectToAction("Complete");
    }
    return View(order);
}

public IActionResult Complete() {
    shopCart.UpdateStock();
    shopCart.DeleteAllFromCart();
    ViewBag.Message = "Your order has been successfully processed. We will contact you soon to establish the billing data, payment meth";
    return View();
}
```

## Views



În dosarul „Views” sunt reprezentate toate fişierile responsabile crearea interfeţei grafice a aplicaţiei (pe langa cod html acestea conţin şi cod C#). Fiecare dosar, poarta denumirea similară controllerului din care view-ul respectiv este returnat şi reprezinta o pagină sau derivatele acesteia, ce sunt returnate in diferite cazuri (în cazul AllCars, unde se returneaza masini in functie de cateforii). Un „View”(fişier .cshtml) este folosit pentru a afişa în browser datele prin intermediul obiectelor de tip model.

În fişierul „\_Layout.cshtml” din dosarul Shared e creat şablonul principal al aplicaţiei web, pe baza caruia se creează fiecare pagină afişată, modificandu-se doar conţinutul din interiorul paginii.

```

<!DOCTYPE html>
<html>
<head>
<title>@ViewBag.Title</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-wEmeIVmKuiNpC+IOBjI7aZPcEzediSyVh5f2yOq5SMMLwNGmvvx4Umlvsketj0" crossorigin="anonym
<link rel="stylesheet" href="/css/style.css" type="text/css">
</head>
<body>
<header>
<nav class="navbar navbar-expand-md navbar-dark bg-dark">
<a class="navbar-brand text-warning ms-3" href="/">II Motors Auto Sale</a>
<div class="collapse navbar-collapse mx-auto" id="navbarCollapse">
<ul class="navbar-nav mx-auto">
<li class="nav-item">
<a class="nav-link @ViewContext.HttpContext.Request.Path.ToString() == "/" ? "active" : """ asp-controller="Home" asp-action="Index">Home</a>
</li>
<li class="nav-item">
<a class="nav-link @ViewContext.HttpContext.Request.Path.ToString() == "/Cars/List" ? "active" : """ href="/Cars/List">All Cars</a>
</li>
<li class="nav-item">
<a class="nav-link @ViewContext.HttpContext.Request.Path.ToString() == "/Cars/List/electric" ? "active" : """ href="/Cars/List/electric">Electric Cars</a>
</li>
<li class="nav-item">
<a class="nav-link @ViewContext.HttpContext.Request.Path.ToString() == "/Cars/List/gasoline" ? "active" : """ href="/Cars/List/gasoline">Gasoline Cars</a>
</li>
<li class="nav-item">
<a class="nav-link @ViewContext.HttpContext.Request.Path.ToString() == "/ShopCart" ? "active" : """ href="/ShopCart">Shop Cart</a>
</li>
</ul>
</div>
<form style="width: 250px;" class="input-group me-3 form-inline" asp-controller="Cars" asp-action="Search">
<input type="text" id="searchInput" class="form-control rounded me-1" name="searchWord" placeholder="Search" aria-label="Search"
aria-describedby="search-addon" />
<button type="submit" id="searchBtn" class="btn btn-warning rounded"></button>
</form>
</nav>
</header>
<div class="text-center">
<@RenderBody()
</div>
<footer class="container">
<p class="float-right"><a href="#">Back to top</a></p>
<p>© 2021 Company, Inc. · <a asp-controller="Home" asp-action="Privacy">Privacy</a> · <a asp-controller="Home" asp-action="Terms">Terms</a></p>
</footer>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0/dist/js/bootstrap.bundle.min.js" integrity="sha384-p34f1UutsS3wqzfto5wAmdvj+osOnFyQfpp4Ua3gs/ZVhX6o0ypYoCJhGGScy+8" crossorigin="anonymous"></script>
<script src="/js/script.js"></script>
</body>
</html>

```

După cum se poate observa în el sunt incluse header-ul (alcatuit din bara de navigație ce conține titlul site-ului, meniul și ansamblul format dintre inputul și butonul destinate funcției de căutare.

Elementele ce se schimbă în restul Views-urilor sunt titlul (@ViewBag.Title) și conținutul (@RenderBody()) paginii respective:

```

<title>@ViewBag.Title</title>

```

```

<div class="text-center">
<@RenderBody()
</div>

```

Toată stilizarea site-ului a fost creată prin intermediul framework-ului Bootstrap, ce conține șabloane de design bazate pe CSS, care pot fi aplicate prin intermediul atributului HTML „class” ce va conține denumirea claselor asupra carora sunt aplicate stilizărilor CSS necesare nouă.

## Testarea aplicației

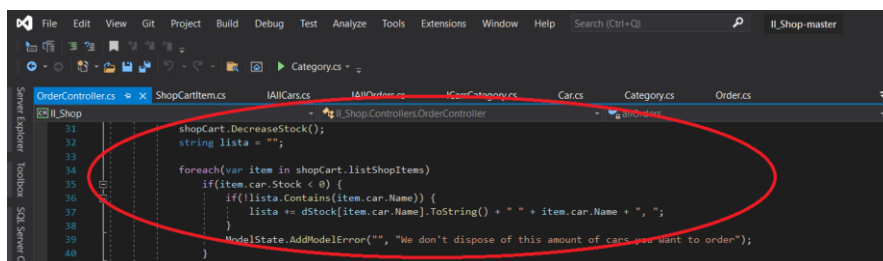
Testarea am realizat-o manual, folosind test cases(Excel atașat)

Una dintre erorile întâlnite în decursul testării aplicației este:

Am avut o problemă la nivelul bazei de date, în momentul în care doream să plasăm o comandă de o mașină care avea un stoc disponibil de 10 mașini și adăugam în coș 11, după ce realizam comanda, în baza de date ne apărea -1.

Rezolvarea erorilor:

Acest lucru l-am rezolvat impunând condiții, observandu-se în următorul printscreen

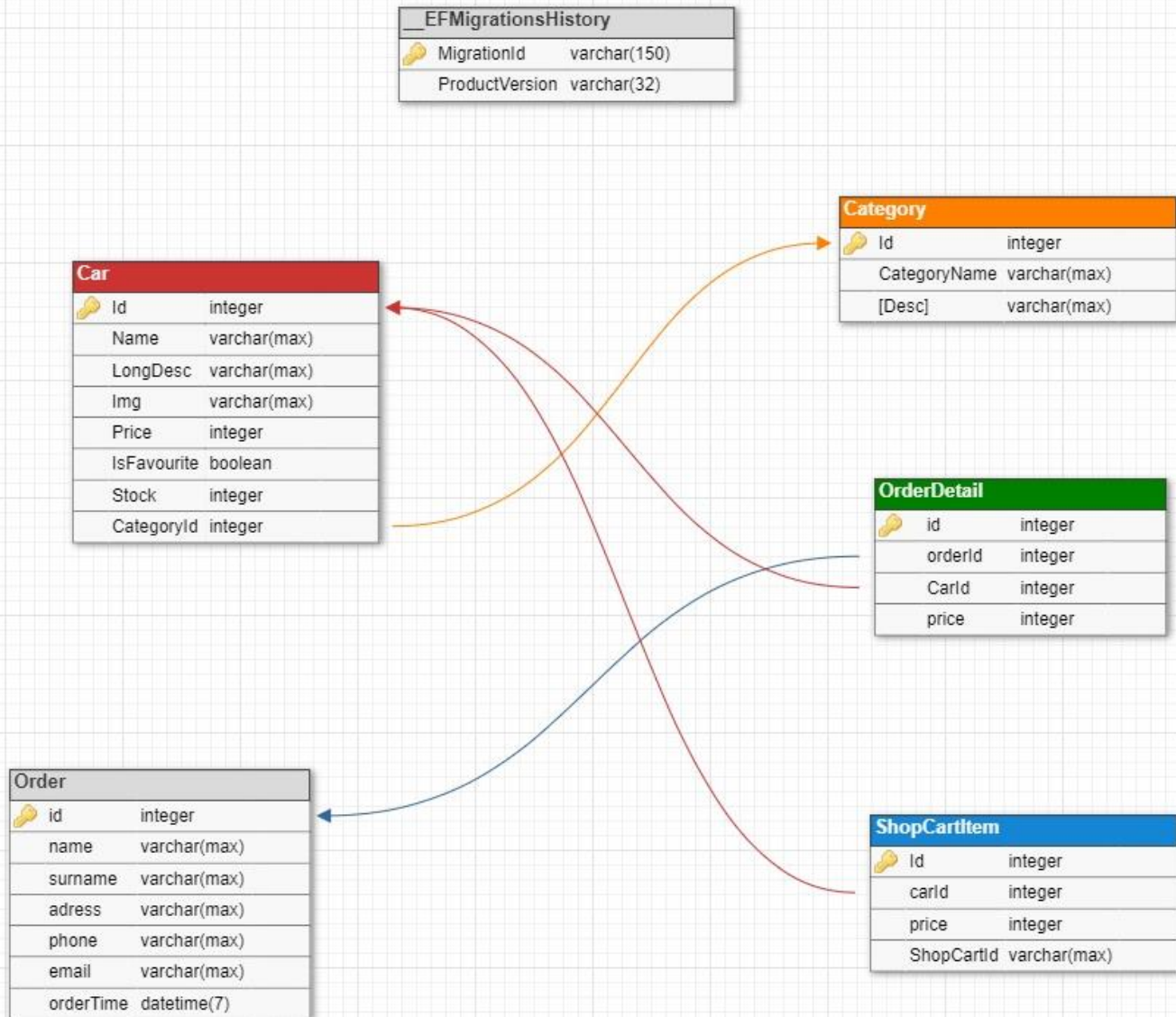


Am creat o listă, iar în momentul în care stocul mașinilor a fost mai mic decât 0, atunci mi s-a afișat mesajul următor în momentul în care am realizat plata: "We don't dispose of this amount of cars you want to order", iar în baza de date îmi va apărea un stoc = 0.

# Arhitectura Aplicatiei

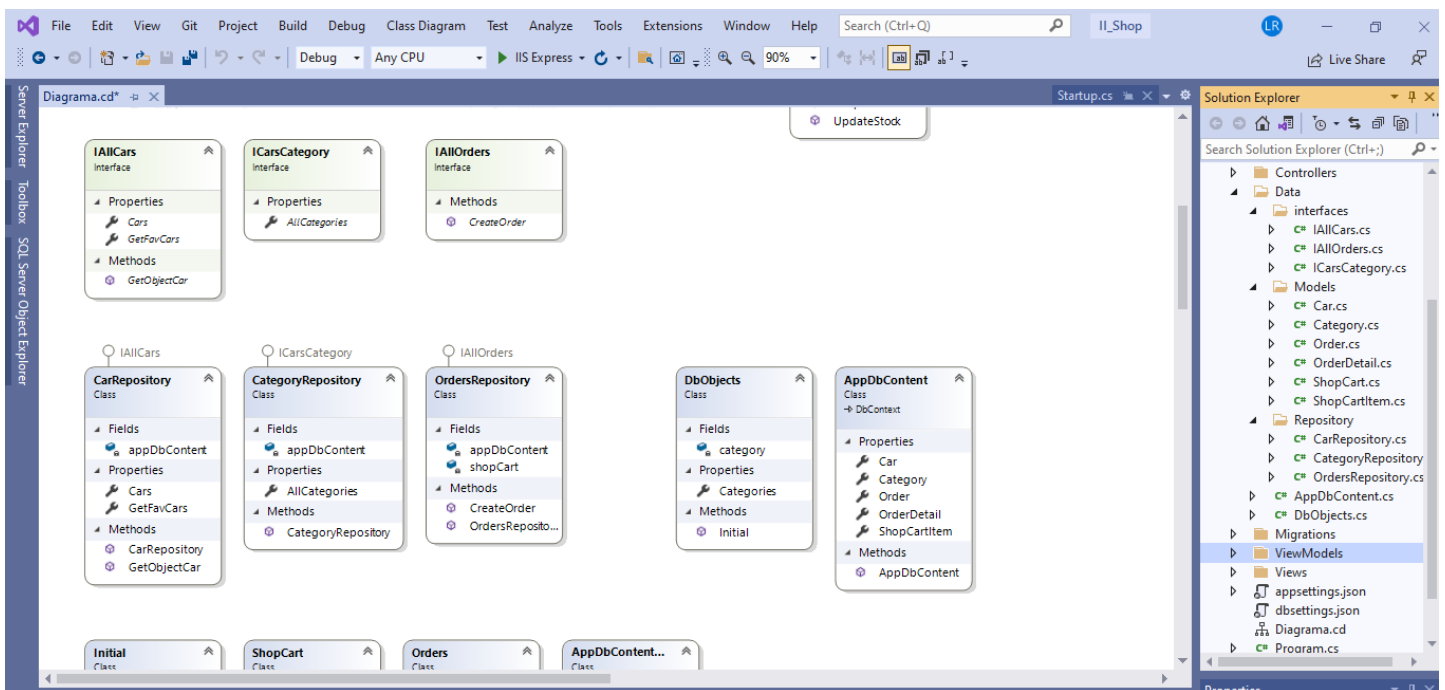
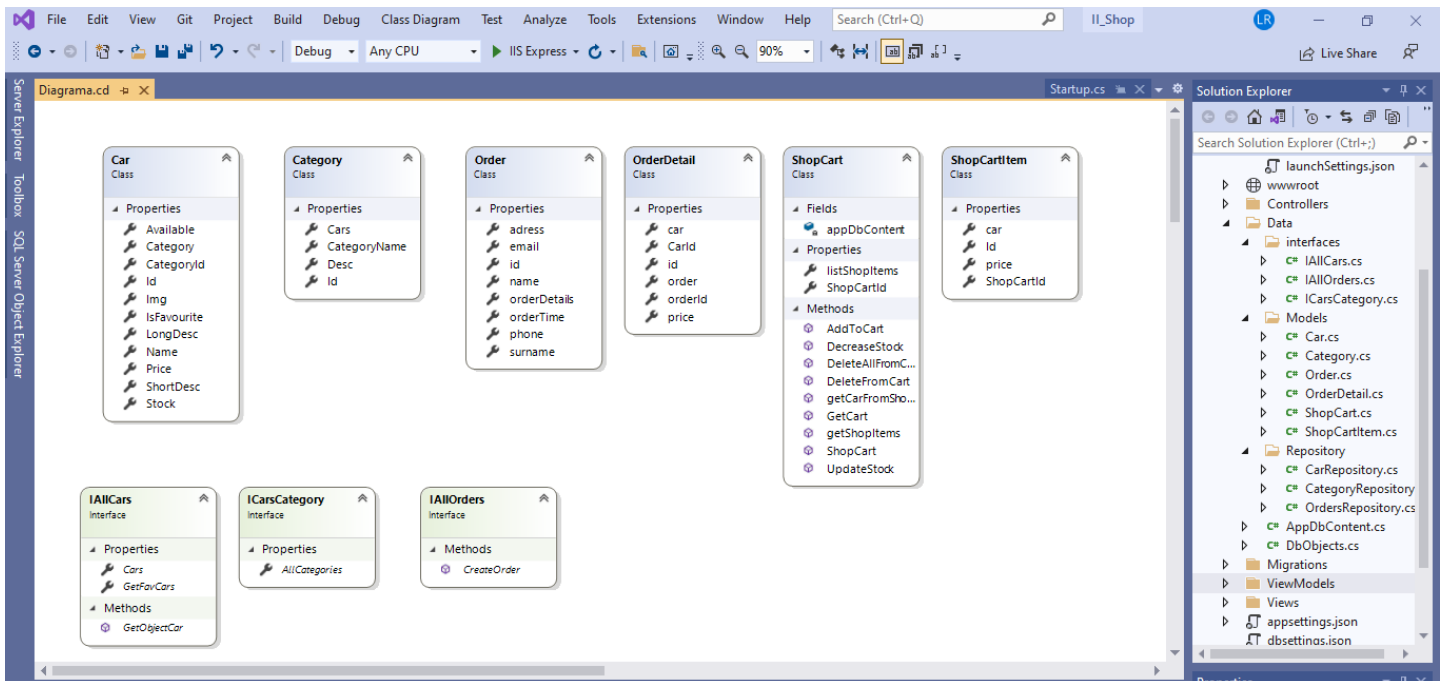
## Reprezentarea bazei de date in DBDesigner

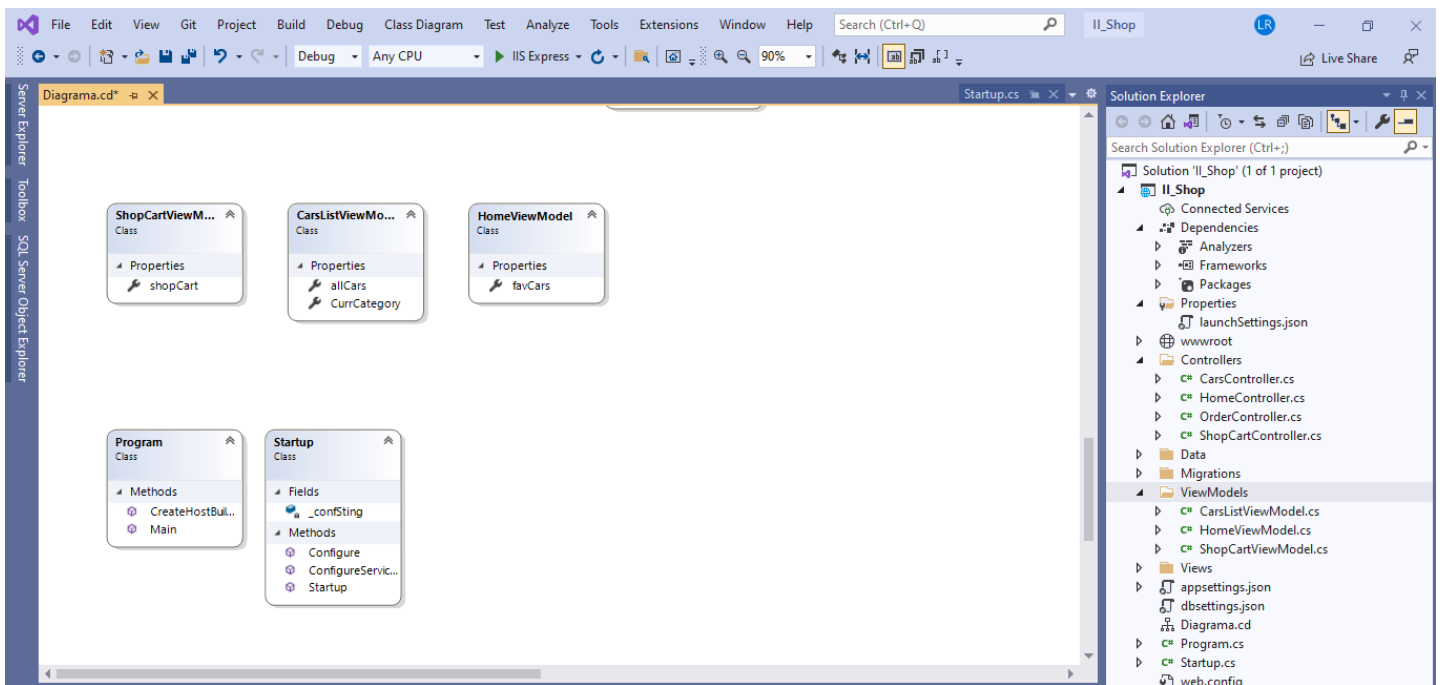
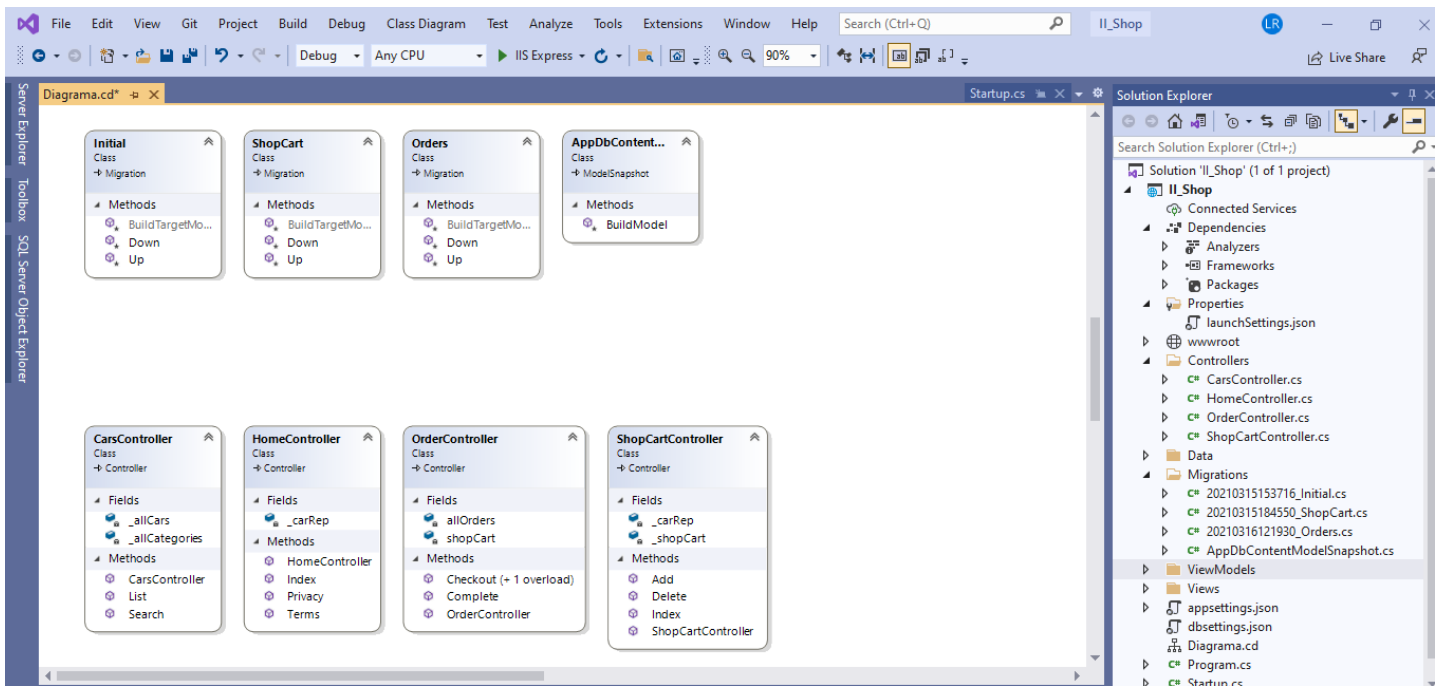
net



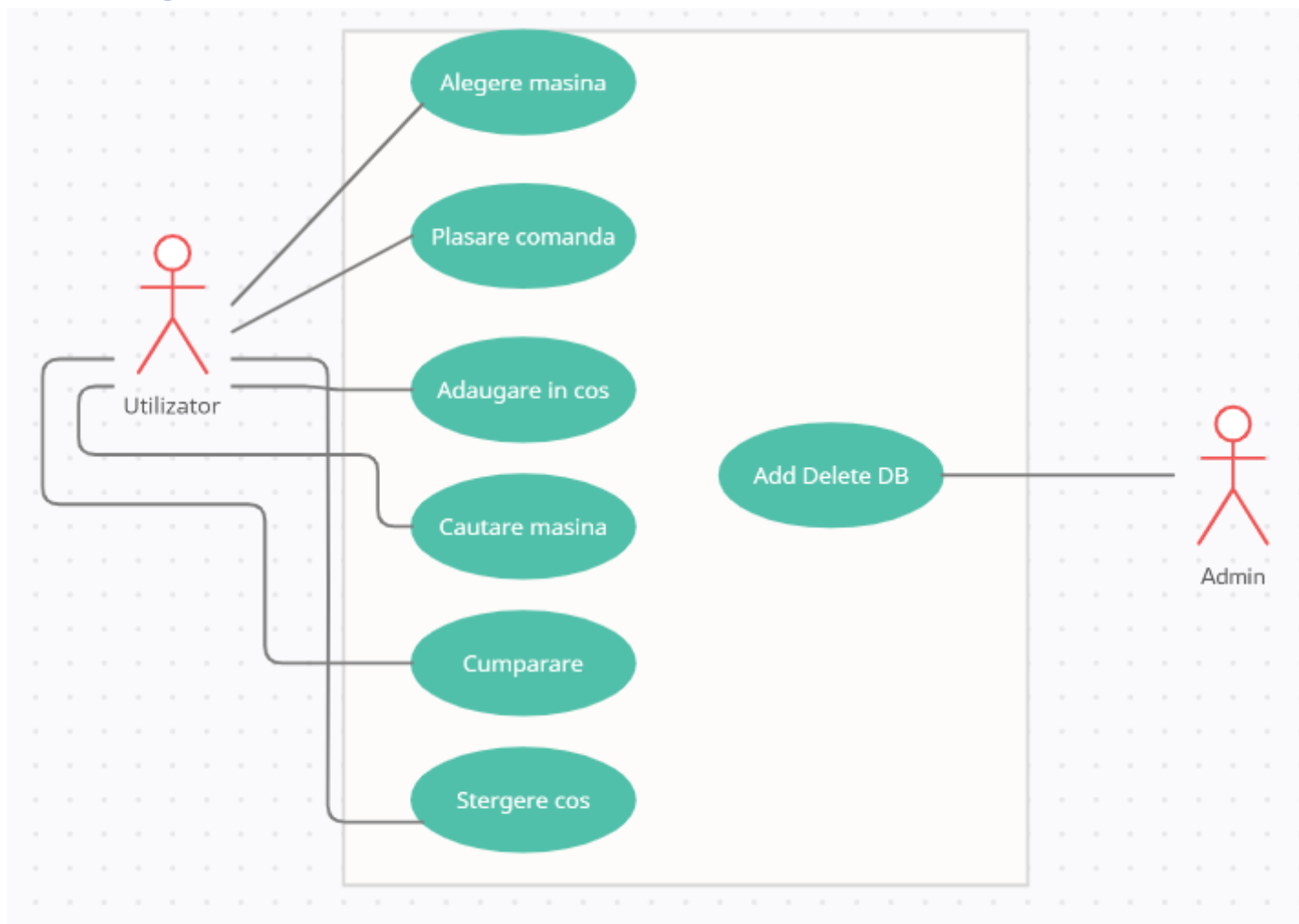


# Diagrama UML





## Diagrama Use Case



\*Cu toate ca nu avem metoda de logare si pagina pentru admin, acesta poate face stergeri si inregistrari, poate edita tabelele, crea inregistrari noi, vizualiza inregistrarile, toate acestea accesand baza de date.

Observație: Nu am anexat tot codul sursă al proiectului din cauză ca conținea prea multe fișiere și linii de cod. De asemenea nu am anexat nici tabelul cu testările efectuate, pentru că nu l puteam incadra calumea in pagina, fiind de dimensiuni mai mari, și am decis sa le trimitem separat.