# **Veb aplikacija za agenciju za izdavanje putničkih vozila**

**Profesor:** prof. dr Vladislav Miškovic

**Asistent:** Milan Tair

**Student:** Stefan

Sadržaj

[1. Uvod 2](#_Toc42722912)

[1.1 Cilj razvoja 2](#_Toc42722913)

[1.2 Obim sistema 3](#_Toc42722914)

[1.3 Prikaz proizvoda 3](#_Toc42722915)

[1.3.1 Perspektiva proizvoda 3](#_Toc42722916)

[1.3.2 Funkcije proizvoda 3](#_Toc42722917)

[1.3.3 Karakteristike korisnika 5](#_Toc42722918)

[1.3.4 Ograničenja 5](#_Toc42722919)

[1.3.5 Definicije 5](#_Toc42722920)

[2. Reference 5](#_Toc42722921)

[3. Specifikacija zahteva 6](#_Toc42722922)

[3.1 Spoljašnji interfejsi 6](#_Toc42722923)

[3.2 Dijagram elemanata organizacije aplikacije - MVC 9](#_Toc42722925)

[3.3 Pogodnost za upotrebu 10](#_Toc42722926)

[3.4 Zahtevane performanse 10](#_Toc42722927)

[3.5 Zahtevi baze podataka 10](#_Toc42722928)

[3.6. Spisak svih kontrolera I njihovih metoda 13](#_Toc42722929)

[3.7 Projektna ograničenja 14](#_Toc42722930)

[3.8 Sistemske karakteristike softvera 15](#_Toc42722931)

# Uvod

Prateća dokumentacija detaljno opisuje cilj razvoja same aplikacije, obim sistema, prikaz okruženja u kojem sistem funkcioniše, način na koji sistem funkcioniše. Takođe su prikazani određeni elementi same aplkacije, dijagrami, model baze, popis svih kontrolera i njihovih metoda koji obavljaju potrebnu programsku i poslovnu logiku sa ciljem izvršavanja svih predviđenih aktivnosti aplikacije, a nevedena su i određena ograničenja koja je neophodno ispoštovati.

## Cilj razvoja

Cilj razvoja ovog sistema jeste da se stvori pouzdana, brza I jednostavna veb aplikacija namenjena agencijama za izdavanje vozila svojim klijentima.

## Obim sistema

Sistem omogućava korisnicima, tj. Zaposlenima u agenciji, da izvršavaju kreiranje novih vozila, kreiranje novih klijenata, izdavanje vozila svojim klijentima, registrovanje vozila, evidentiranje troškova nastalih pri održavanju vozila, mogu se pregledati detalji vozila, kao i zaključivate pojedinačna izdavanja vozila, pri čemu je potrebno evidentirati stanje vozila. Korisnici realizovanog sistema su isključivo zaposleni u agenciji.

Dakle, realizacijom ovog Sistema obuhvaćen je skup operacija potrebnih za osnovno funkcionisanje jedne agencije za izdavanje vozila.

## Prikaz proizvoda

Naziv ovog proizvoda je “Veb aplikacija za agenciju za izdavanje putničkih vozila”.

Osnovne funkcije ovog sistema čine:

* Pregled voznog parka i trenutnog statusa svih vozila
* Izvršavanje izdavanja vozila klijentima agencije
* Funkcionalnosti osnovnog održavanja vozila (registracija, popravke…)

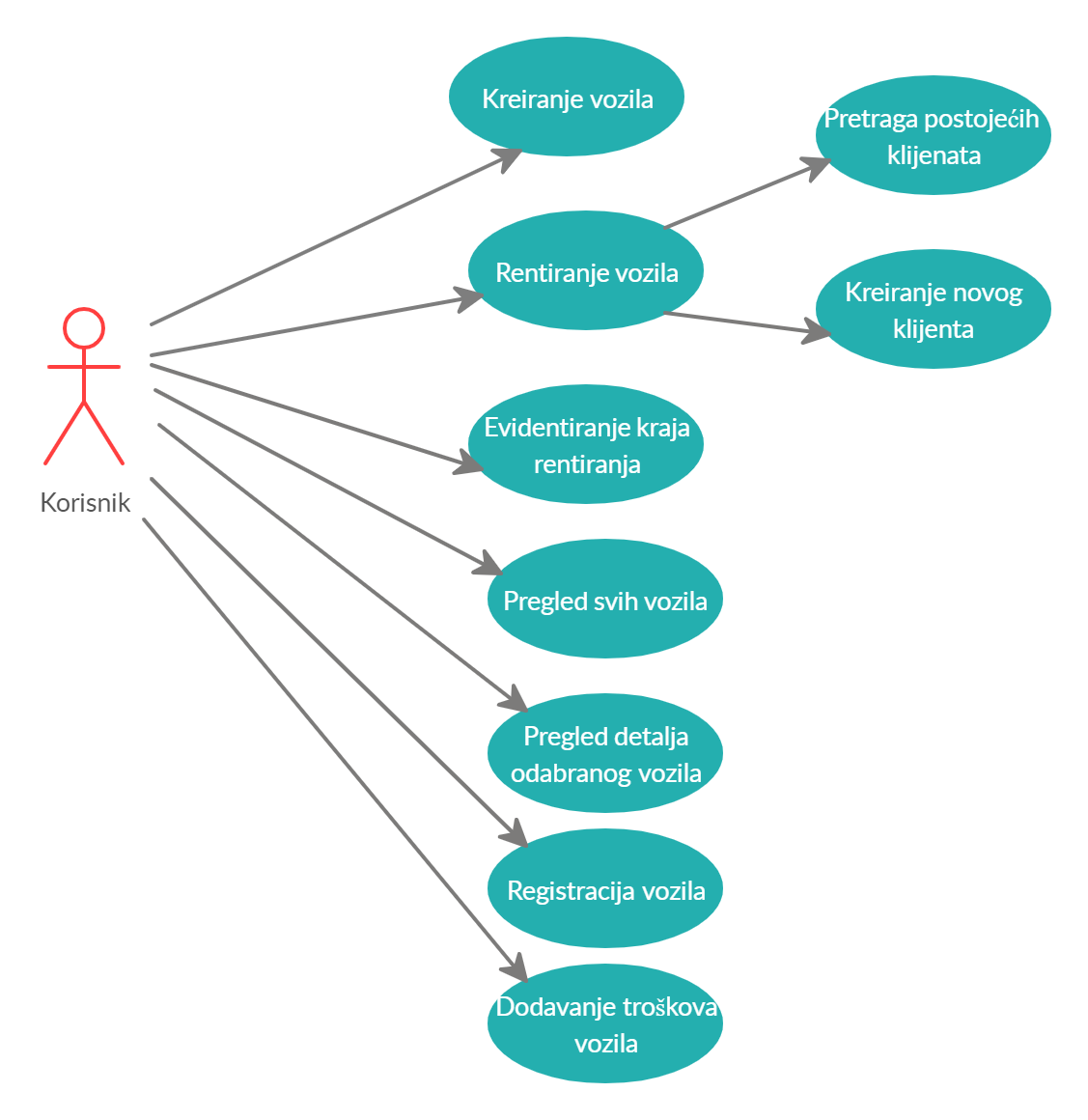
Cilj realizovanog Sistema je da se, na bezbedan i efikasan način, agencijama ponudi aplikacija kojom se obuhvata realizacija I održavanje sistema za izdavanje vozila. Da se obezbedi *GUI* koji će biti jednostavan I intuitivan za korišćenje svim korisnicima bez potrebe za bilo kakvim predznanjem ili iskustvom u radu sa ovakvim vrstama sistema. Takođe, cilj je kreirati aplikaciju koja se može sa podjednakom lakoćom koristiti na različitim uređajima sa različitim dimenzijama ekrana (*responsiveness*).

### Perspektiva proizvoda

Sistem je nezavisan od drugih, spoljašnjih sistema i funkcioniše potpuno samostalno.

Bitno je napomenuti da je sistem realizovan na takav način da, ukoliko bi se u budućnosti javila potreba za povezivanjem sistema sa drugim sistemima, to bi bilo veoma lako izvodljivo. Primera radi, ovakvo povezivanje bi imalo smisla ostvariti sa nekim od sistema za prodaju vozila, odakle bi eventualno naš sistem kupovao vozila.

### Funkcije proizvoda



*Slika 1.3.2. use-case dijagram*

Napomena: sve funkcije prikazane na UML dijagramu sa slike su dostupne nakon prijave korisnika na sistem.

Funkcije koje mogu obavljati korisnici aplikacije su:

* Kreiranje novog vozila
* Izdavanje odabranog vozila
  + Već postojećem klijentu (pretraga klijenata po broju lične karte)
  + Novom klijentu (kreiranje novog klijenta)
* Završetak izdavanja odabranog vozila I evidentiranje trenutnog stanja vozila
* Pregled svih vozila i njihovih statusa (status: registracija, slobodan/zauzet…)
* Pregled detalja za odabrano vozilo
* Registracija odabranog vozila koje je neregistrovano
* Zapisivanje dodatnih troškova odabranog vozila

### 1.3.3 Karakteristike korisnika

Aplikacija je namenjena za sve zaposlene u agenciji koji se bave izdavanjem ili održavanjem vozila. Interfejs je dizajniran da bude lak za korišćenje i nije neophodno nikakvo prethodno iskustvo ili obrazovanje za njegovo korišćenje. Jedino što je korisniku potrebno jeste pristup Internetu

### 1.3.4 Ograničenja

Kada su u pitanju ograničenja, s obzirom da je aplikacija za sada projektovana isključivo za korišćenje od strane zaposlenih, neophodno je prijaviti se na sistem kako bi se ostvarili mogućnost za korišćenje njenih pogodnosti.

## 1.3.5 Definicije

Definicije manje poznatih pojmova i skraćenica:

1. *Browser* – veb pregledač (*Google Chrome, Mozzilla Firefox*…)
2. *Client / FrontEnd* – klijentska strana, deo veb aplikacije koji se prikazuje korisniku u *browser*-u
3. *Server / BackEnd –* serverska strana veb aplikacije zadužena za komunikaciju sa bazom podataka I dopremanje podataka, koji se prikazuju na klijentskoj strain
4. *HTTP Request* – zahtev za podacima, osnovni element komunikacije između klijenta i servera
5. *GUI* , “*Graphic user interface” –* grafički korisnički interfejs
6. *Responsiveness* – prilagodljivost prikaza sadržaja različitim dimenzijama ekrana
7. *ORM, “Object relational mapper”* – objektno relacioni mapper, koristan alat pri modeliranju entiteta I relacija baze podataka
8. *MVC, Model-View-Controller* – arhitektura organizacije koda koja se zasniva na 3 celine sa uočljivo razdvojenim ulogama
9. *Separation of concerns* – podela odgovornosti

# 2. Reference

[1] “Aplikacija za agenciju za izdavanje putničkih vozila”

<http://zadatak.singidunum.ac.rs/app/piivt-biranje-tema/?action=topic&id=7&fbclid=IwAR3CaPvOu4N0TK737q7EqBHw0RE2M-ZvIxedHIkJ9k-Eco0F09is4fKYCC4>

[2] <https://docs.nestjs.com/> dokumentacija rayvojnog okvira *NestJS*

[3] <https://typeorm.io/#/> dokumentacija *Type* *ORM*-a

[4] <https://reactjs.org/> dokumentacija *React* biblioteke

[5] <https://react-bootstrap.github.io/> dokumentacija *ReactBootstrap* biblioteke

[6] <https://getbootstrap.com/> *Bootstrap* dokumentacija

[7] <https://github.com/MilanTairYouTube/React2020> *GitHub React* projekat, Milan Tair

[8] <https://developer.mozilla.org/en-US/docs/Web/JavaScript> *JavaScript* dokumentacija

[9] <https://www.w3schools.com/> *JavaScript/CSS/HTML/NodeJS* dokumentacija

[11] <https://www.restapitutorial.com/httpstatuscodes.html> *HTTP* status kodovi

[10] <https://www.youtube.com/channel/UCoabUK8jyHYwuxY3DFIbWcQ> *Youtube* kanal, Milan Tair

# Specifikacija zahteva

## 3.1 Spoljašnji interfejsi

Nakon uspešne prijave, korisniku se prikazuje glavni ekran aplikacije koji čini prikaz svih vozila u okviru voznog parka agencije (Slika 3.1.1). U tabeli su prikazani redom: registracioni broj, model, trenutna dostupnost automobila, kategorija, tip goriva, pređena kilometraža, stanje rezervoara, vreme isteka registracije, kao i 4 opcije: iznajmljivanje vozila, završetak iznajmljivanja, evidentiranje novih troškova i detalji o vozilu.

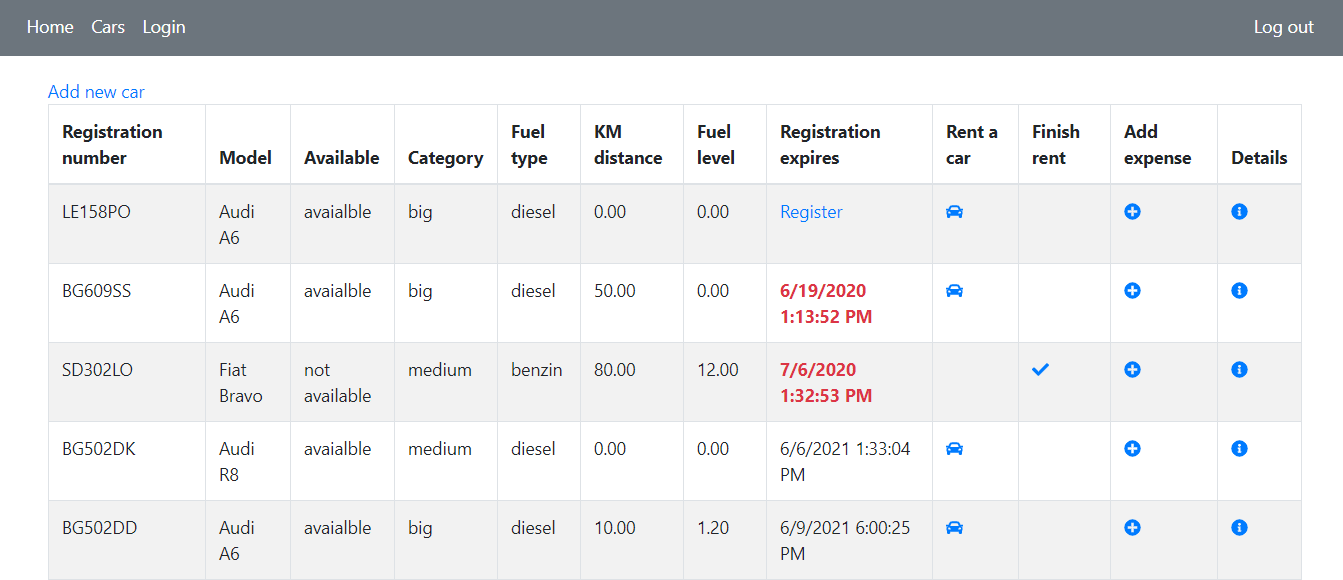
Prikazani automobili su sortirani po vremenu isteka registracije rastuće, što znači da su pri vrhu automobili čija registracija ističe u najkraćem vremenskom intervalu ili koji su trenutno neregistrovani

U okviru kolone za vreme isteka registracije, crvenom bojom su istaknuti automobili kojima registracija ističe u roku od mesec dana. Registrovanje vozila ostvaruje se klikom na link za registraciju čime vozilo automatski biva registrovano.

Opcija za iznajmljivanje vozila (*Rent a car*) je blokirana za vozila koja su trenutno iznajmljenja ili neregistrovana.

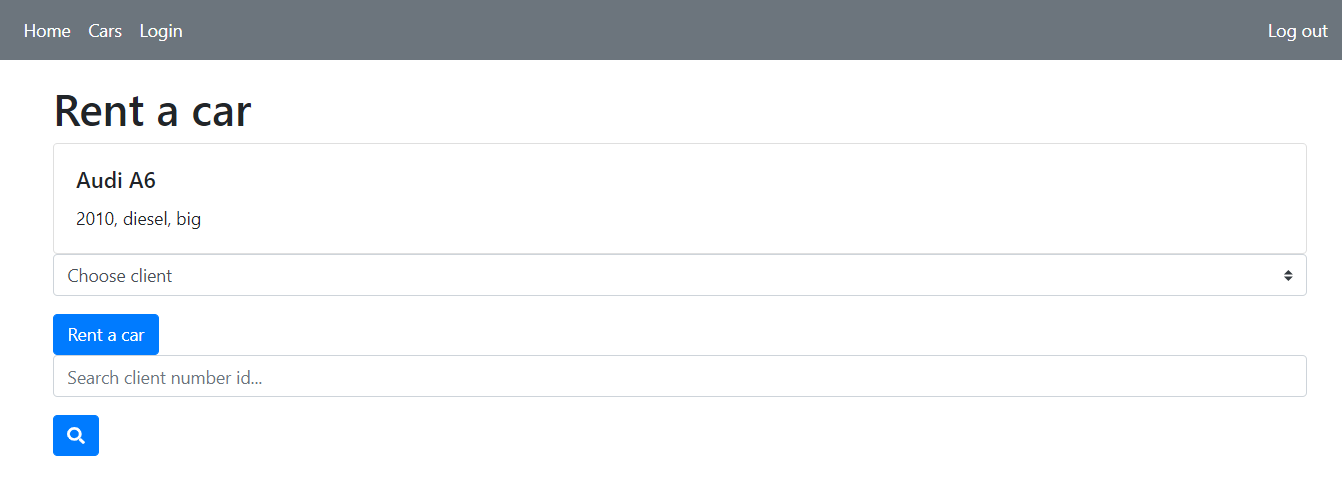
Opcija za završetak iznajmljivanja (*Finish rent*) dostupna je samo za vozila koja su trenutno iznajmljenja.

Opcije za dodavanje troškova (*Add expense*) i pregled detalja (*Details*) dostupna je za sva vozila.



*Slika 3.1.1 Prikaz svih automobila voznog parka I njihov trenutni status*

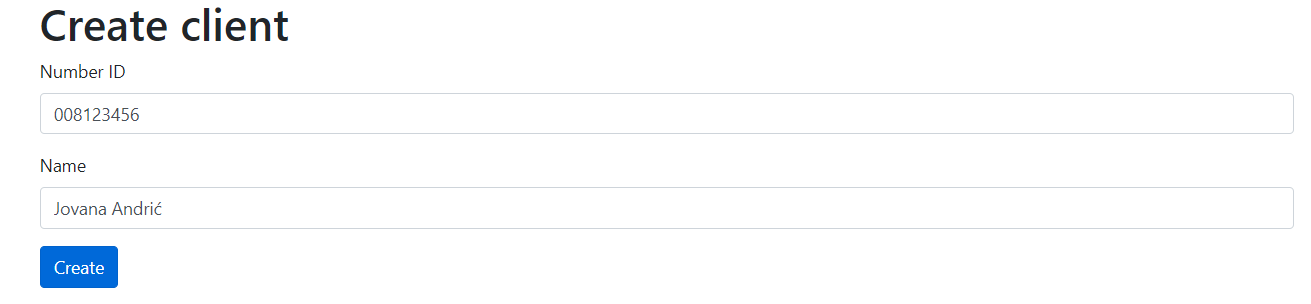
Klikom na opciju za iznajmljivanje vozila (*Rent a car*) prikazuje se forma za izbor klijenta (Slika 3.1.2). U okviru forme prikazani su osnovni detalji izabranog vozila, kao i dve opcije za izbor klijenta za kojeg se vrši iznajmljivanje odabranog vozila, a to su: pretraga klijenata iz padajuće liste i pretraga po broju lične karte. Ukoliko klijent sa datim brojem lične karte ne postoji u bazi podataka, korisniku se signalizira odgovarajućom poruku i u dnu stranice mu se prikazuje forma za kreiranje klijenta (Slika 3.1.3. i Slika 3.1.4). Kada korisnik unese podatke potrebne za kreiranje klijenta i klikne na opciju *Create* prikazuje mu se odgovarajuća poruka I ukoliko je klijent uspešno kreiran, biva automatski selektovan i potom je potrebno još samo potvrditi iznajmljivanje automobile novo kreiranom klijentu (Slika 3.1.5).



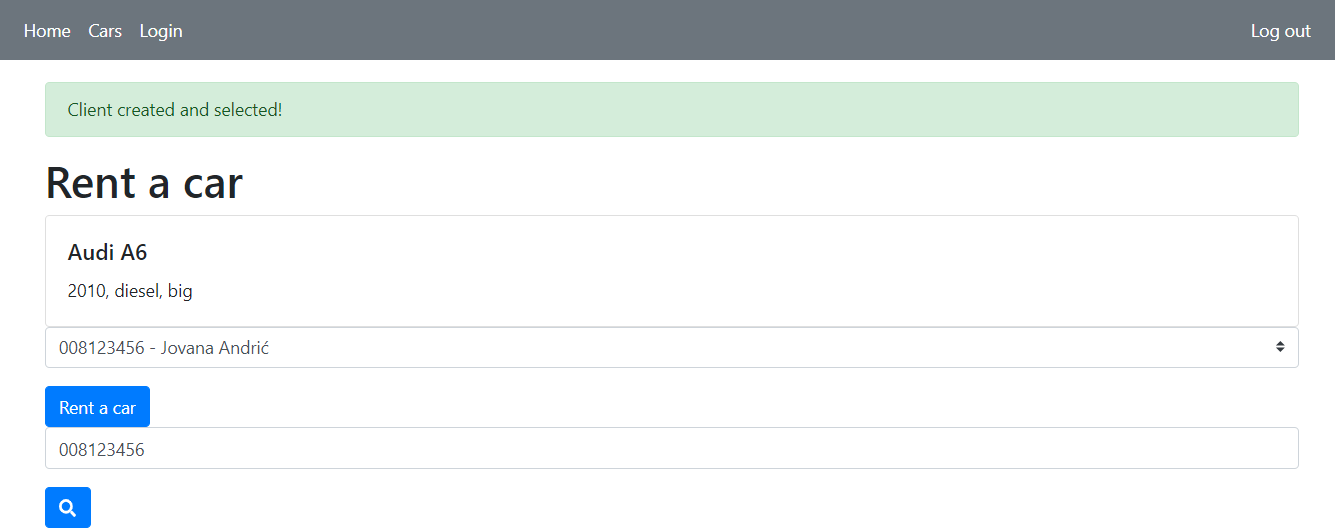
*Slika 3.1.2. Iznajmljivanje automobile – izbor klijenta*

## 

*Slika 3.1.3. Klijent nije pronađen*

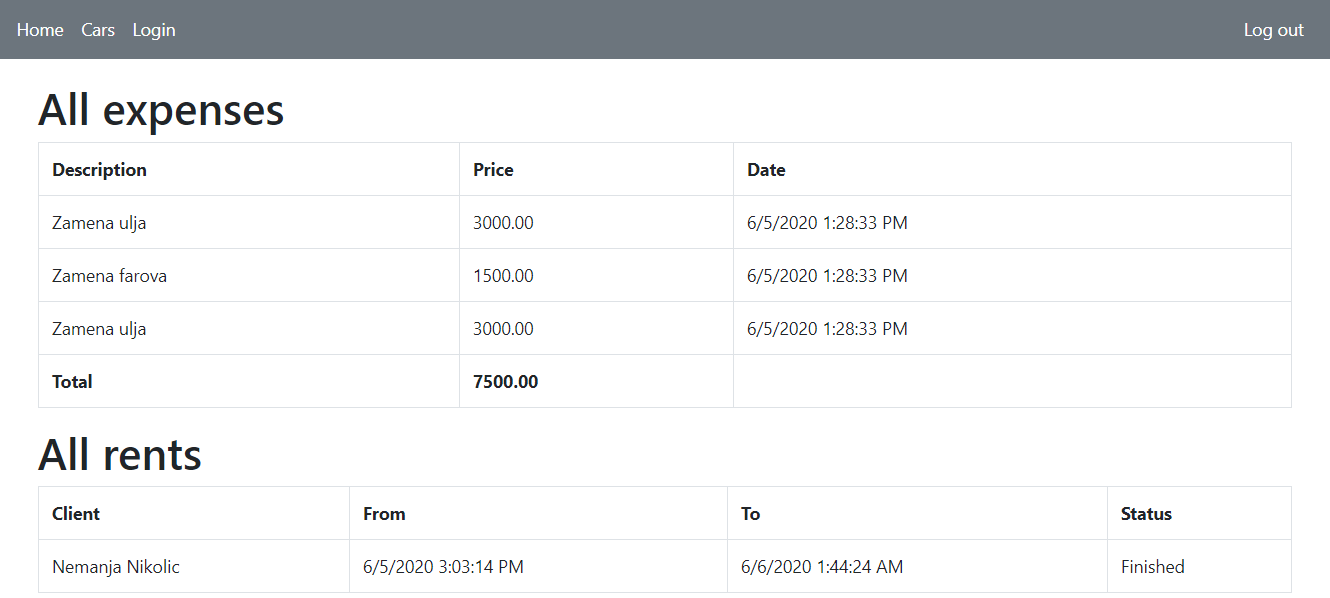


*Slika 3.1.4. Kreiranje novog klijenta*



*Slika 3.1.5. Potvrda iznajmljivanja*

Klikom na opciju za detalje o proizvodu prikazuje se lista svih troškova vozila, kao i spisak svih renti vozila, kao i lista korisnika koji su izvršavali prikazane rente (Slika 3.1.6)

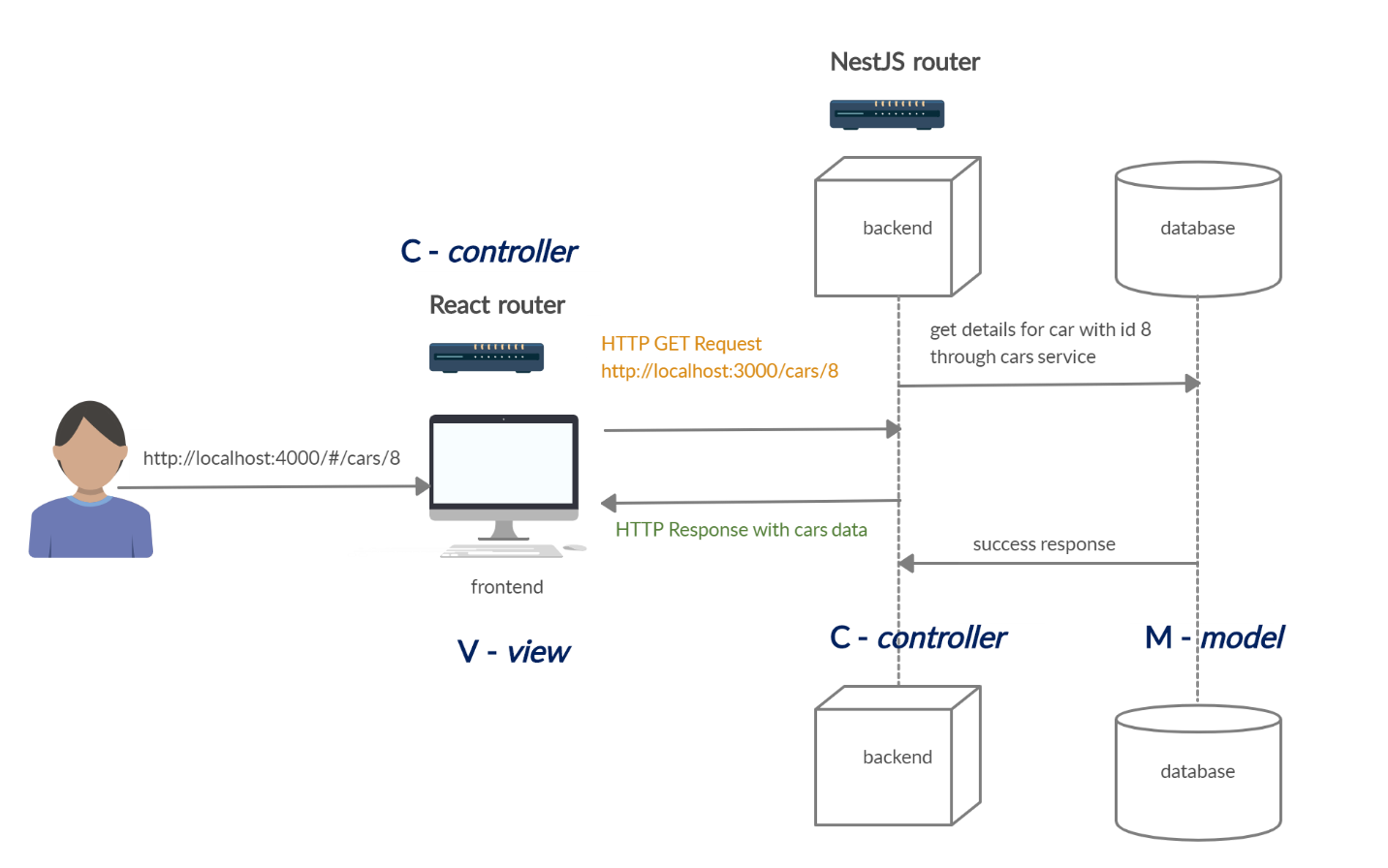


*Slika 3.1.6 – detalji o automobile*

## 3.2 Dijagram elemanata organizacije aplikacije - MVC

Serverski deo aplikacije je realizovan u *NestJS* razvojnom okviru, dok je klijentski deo realizovan u *ReactJS* biblioteci. Aplikacija funkcioniše tako što korisnik gađa određenu rutu aplikacije (*ReactJS* ruter) I korisniku se prikazuje komponenta definisana određenom rutom. Da bi se dohvatili potrebni podaci za prikaz određene stranice, u pozadini se šalje *HTTP* zahtev (*eng. request*) za tim podacima ka serveru, na definisanu rutu predviđenu za dohvatanje potrebnih podataka (*NestJS* ruter). Na Slici 3.2.1. prikazan je primer u kojem se korisniku prikazuje stranica sa detaljima o automobile sa prosleđenim id-em (u primeru je to id:8).

S obzirom da se sa servera isključivo dohvataju podaci u sirovom obliku, i kao takvi prosleđuju klijentu putem JSON formata, dok se na klijentskoj strani ti podaci prikazuju u željenom obliku, jasno je razdvojena uloga svakog od delova naše aplikacije (*eng. separation of concerns)*. Prema tome, moglo bi se reći da na serveru postoje kontroler (*NestJS* ruter koji definiše sve postojeće rute) i model (baza podataka kojoj se pristupa putem servisa koji komuniciraju sa kontrolerom), dok na klijentskoj strani postoje komponente koje su ujedno i kontroleri (kontrolišu dohvatanje podataka sa servera) i *view*-ovi (formiraju HTML kod zadužen za prikaz dobijenih podataka sa servera)



*Slika 3.2.1. organizacija aplikacije sa akcentom na MVC arhitekturu*

## Pogodnost za upotrebu

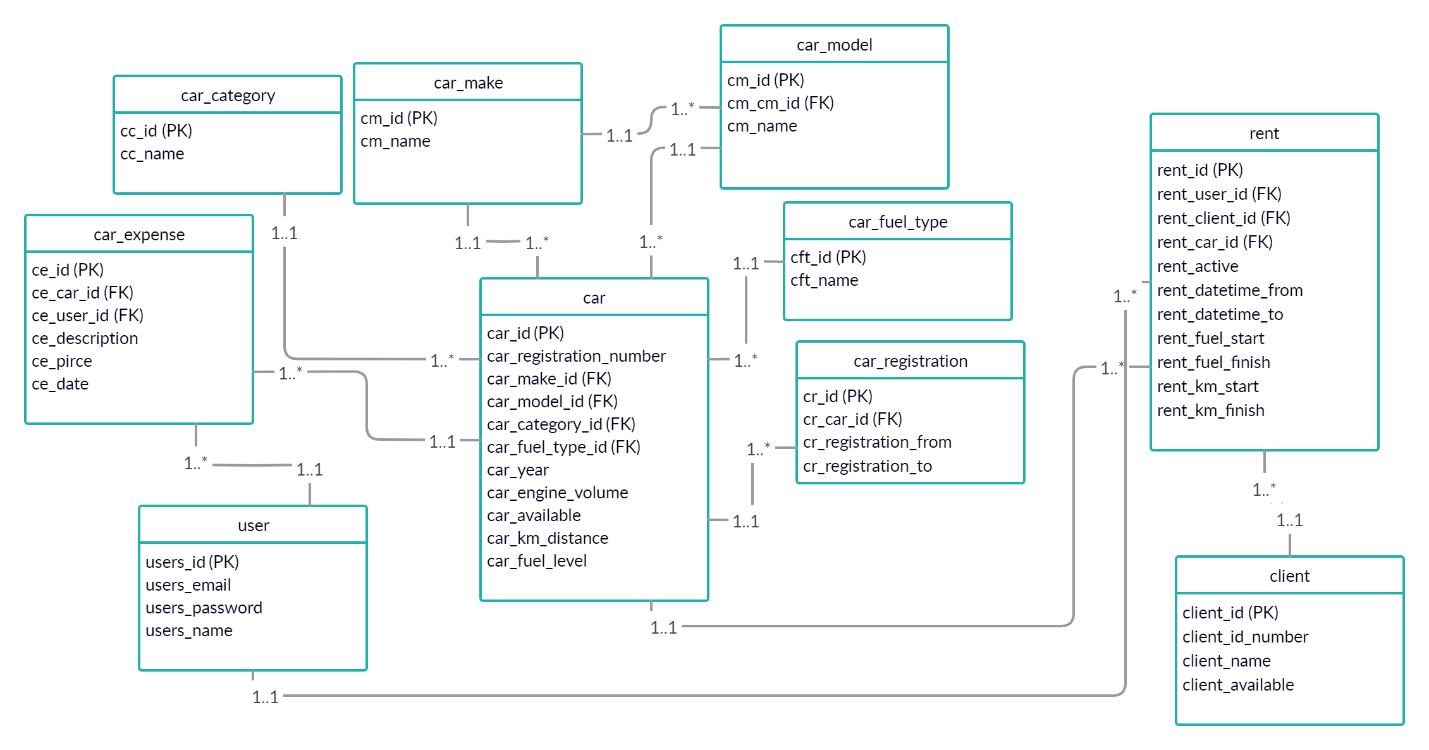
Aplikacija zadovoljava efektivnost i efkasnost korišćenja i predstavlja sistem koji pozitivno utiče na zadovoljstvo krajnjih korisnika.

## 3.4 Zahtevane performanse

Vreme odziva Web aplikacije je minimalno, a broj istovremenih korisnika ne remeti rad same aplikacije i ne predstavlja problem samom sistemu niti usporava njegov rad.

## 3.5 Zahtevi baze podataka

Za realizaciju ove veb aplikacije korišćenja je MySQL relaciona baza podataka. Na Slici 3.5.1. prikazan je model baze podataka sa svim relacijama među tabelama.



*Slika 3.5. model baze podataka*

U nastavku teksta dat je tabelarni prikaz svih tabela iz baze podataka, sa opisom svih polja, njihovih tipova, kao i ograničenja.

* user

|  |  |  |
| --- | --- | --- |
| **Naziv kolone** | **Tip podataka** | **Ograničenja** |
| users\_id | int(11) | PK, AI |
| users\_email | varchar(64) | UNIQUE |
| users\_password | varchar(250) |  |
| users\_name | varchar(32) |  |

* car\_make

|  |  |  |
| --- | --- | --- |
| **Naziv kolone** | **Tip podataka** | **Ograničenja** |
| cm\_id | int(11) | PK, AI |
| cm\_name | varchar(64) | UNIQUE |

* car\_model

|  |  |  |
| --- | --- | --- |
| **Naziv kolone** | **Tip podataka** | **Ograničenja** |
| cm\_id | int(11) | PK, AI |
| cm\_cm\_id | int(11) | FK references car\_make (cm\_id) |
| Cm\_name | varchar(32) |  |

* car\_category

|  |  |  |
| --- | --- | --- |
| **Naziv kolone** | **Tip podataka** | **Ograničenja** |
| cc\_id | int(11) | PK, AI |
| cc\_name | varchar(32) | UNIQUE |

* car\_fuel\_type

|  |  |  |
| --- | --- | --- |
| **Naziv kolone** | **Tip podataka** | **Ograničenja** |
| cft\_id | int(11) | PK, AI |
| cft\_name | varchar(32) | UNIQUE |

* client

|  |  |  |
| --- | --- | --- |
| **Naziv kolone** | **Tip podataka** | **Ograničenja** |
| client\_id | int(11) | PK, AI |
| client\_id\_number | varchar(9) | UNIQUE |
| client\_name | varchar(32) |  |
| client\_available | tinyint(1) | DEFAULT: 1 |

* car

|  |  |  |
| --- | --- | --- |
| **Naziv kolone** | **Tip podataka** | **Ograničenja** |
| car\_id | int(11) | PK, AI |
| car\_registration\_number | varchar(10) | UNIQUE |
| car\_make\_id | int(11) | FK references car\_make (cm\_id) |
| car\_model\_id | int(11) | FK references car\_model (cm\_id) |
| car\_fuel\_type\_id | int(11) | FK references car\_fuel\_type (cft\_id) |
| car\_category\_id | int(11) | FK references car\_category (cc\_id) |
| car\_make\_id | int(11) | FK references car\_make (cm\_id) |
| car\_year | int (4) |  |
| car\_engine\_volume | decimal(10, 2) |  |
| car\_available | Tinyint(1) | DEFAULT: 1 |
| car\_km\_dist | decimal(10, 2) | DEFAULT: 0.00 |
| car\_fuel\_level | decimal(10, 2) | DEFAULT: 0.00 |

* car\_expense

|  |  |  |
| --- | --- | --- |
| **Naziv kolone** | **Tip podataka** | **Ograničenja** |
| ce\_id | int(11) | PK, AI |
| ce\_user\_id | int(11) | FK references users (user\_id) |
| ce\_car\_id | int(11) | FK references car (car\_id) |
| ce\_description | varchar(64) |  |
| ce\_price | decimal(10,2) |  |
| ce\_date | datetime | DEFAULT: CURRENT\_TIMESTAMP |

## 3.6. Spisak svih kontrolera I njihovih metoda

U ovom potpoglavlju prikazan je spisak svih kontrolera i njihovih metoda koji obavljaju potrebnu programsku logiku sa ciljem izvršavanja svih predviđenih aktivnosti aplikacije. U nastavku su tabelarno prikazani svi serverski kontroleri ruta, realizovani u *NestJS* razvojnom okviru.

* users.controller.ts

|  |  |  |
| --- | --- | --- |
| **Ruta** | **HTTP Metod** | **Metoda kontrolera** |
| /users | GET | findAll() |

* clients.controller.ts

|  |  |  |
| --- | --- | --- |
| **Ruta** | **HTTP Metod** | **Metoda kontrolera** |
| /clients | GET | findAll() |
| /clients/:num\_id | GET | findOne(@Param('num\_id')numId: string) |
| /clients | POST | createClient(@Body()createClientDto: CreateClientDto) |

* cars.controller.ts

|  |  |  |
| --- | --- | --- |
| **Ruta** | **HTTP Metod** | **Metoda kontrolera** |
| /cars | GET | findAll() |
| /cars/models | GET | findAllModels(@Query('makeId') makeId: number) |
| /cars/models/:id | GET | findModel(@Param('id') id: string) |
| /cars/makes | GET | findAllMakes() |
| /cars/categories | GET | findAllCategories() |
| /cars/fueltypes | GET | findAllFuelTypes() |
| /cars/available | GET | findAllAvailableCars() |
| /cars/:id | GET | findOne(@Param('id') id: string) |
| /cars/ | POST | createCar(@Body()createCarDto: CreateCarDto) |
| /cars/expenses/:carId | POST | addExpenses(@Body()addExpensesDto: AddExpensesDto,@Param('carId')carId: number) |
| /cars/register | POST | @Body() carRegisterDto: CarRegisterDto |

* rents.controller.ts

|  |  |  |
| --- | --- | --- |
| **Ruta** | **HTTP Metod** | **Metoda kontrolera** |
| /rent | GET | findAll() |
| /rent/finish | POST | finsihRent(@Body()finishRentDto: FinishRentDto) |
| /rent/active/carId | GET | findActiveRent @Param('carId') carId: number) |
| /rent/:carId/:clientId | GET | rentACar(@Param('carId') carId: number, @Param('clientId') clientId: number) |

## 3.7 Projektna ograničenja

Serverski deo projekta je realizovan na *NodeJS* platformi korišćenjem *NestJS* razvojnog okvira. Sam kod je organizovan prema pravilima *MVC* arhitekture. Baza podataka je relaciona I korišćena je *MySQL* baza odataka. Klijentska strana je realizovana u *ReactJS* biblioteci, dok je za potrebe responsive dizajna korišten *Bootstrap*, tj. *ReactBootstrap*. Izrada projekta je sprovedena korišćenjem alata za verziranje koda *Git* i celokupan kod projekta nalazi se na *GitHub*-u na linku: <https://github.com/stex123/rent-a-car>.

## 3.8 Sistemske karakteristike softvera

Posebna pažnja je obraćena na realizaciju zahteva pouzdanosti, responsive dizajna, bezbednosti sistema, pogodnosti za održavanje aplikaije, prenostivosti i raspoloživosti.