

# Oblikovanje programske podpore

Ak. god. 2019./2020.

## Moj AutoServis

Dokumentacija, Rev. 1.0

Grupa: *InfiniTeam*

Voditelj: *Patrik Matošević*

Datum predaje: *15. studenoga. 2019.*

Nastavnik: *Nikolina Frid*

# Sadržaj

<b>1</b>	<b>Dnevnik promjena dokumentacije</b>	<b>3</b>
<b>2</b>	<b>Opis projektnog zadatka</b>	<b>5</b>
2.1	Primjeri sličnih rješenja . . . . .	7
2.2	Moguće nadogradnje projektnog zadatka . . . . .	8
<b>3</b>	<b>Specifikacija programske potpore</b>	<b>10</b>
3.1	Funkcionalni zahtjevi . . . . .	10
3.1.1	Obrasci uporabe . . . . .	13
3.1.2	Sekvencijski dijagrami . . . . .	29
3.2	Ostali zahtjevi . . . . .	35
<b>4</b>	<b>Arhitektura i dizajn sustava</b>	<b>36</b>
4.1	Baza podataka . . . . .	39
4.1.1	Opis tablica . . . . .	39
4.1.2	Dijagram baze podataka . . . . .	43
4.2	Dijagram razreda . . . . .	45
4.3	Dijagram stanja . . . . .	52
4.4	Dijagram aktivnosti . . . . .	54
4.5	Dijagram komponenti . . . . .	56
<b>5</b>	<b>Implementacija i korisničko sučelje</b>	<b>58</b>
5.1	Korištene tehnologije i alati . . . . .	58
5.2	Ispitivanje programskog rješenja . . . . .	60
5.2.1	Ispitivanje komponenti . . . . .	60
5.2.2	Ispitivanje sustava . . . . .	63
5.3	Dijagram razmještaja . . . . .	69
5.4	Upute za puštanje u pogon . . . . .	70
5.4.1	Instalacija i konfiguracija baze podataka . . . . .	70
5.4.2	Konfiguracija i pokretanje poslužitelja . . . . .	70

<b>6 Zaključak i budući rad</b>	<b>73</b>
<b>Popis literature</b>	<b>75</b>
<b>Indeks slika i dijagrama</b>	<b>76</b>
<b>Dodatak: Prikaz aktivnosti grupe</b>	<b>77</b>

# 1. Dnevnik promjena dokumentacije

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Napravljen predložak.	Matošević	17.10.2019.
0.2	Napisan opis projekta te neka postojeća slična rješenja.	Boras	22.10.2019.
0.2.1	Napisani funkcionalni i ostali zahtjevi	Has, Prpić	23.10.2019.
0.3	Dopunjen opis projektnog zadatka.	Cvitković, Facković	24.10.2019.
0.3.1	Uređeni i dopunjeni funkcionalni zahtjevi	Prpić	25.10.2019.
0.3.2	Ispravljene greške u opisu projektnog zadatka. Dodan header i footer.	Boras	26.10.2019.
0.4	Dodani <i>Use Case</i> dijagrami.	Matošević	26.10.2019.
0.5	Dodani opisi svih obrazaca uporabe.	Matošević, Kolarec, Has	29.10.2019.
0.5.1	Ispravljane nepravilnosti u opisima obrazaca uporabe	Kolarec, Matošević	3.11.2019.
0.5.2	Funkcionalni zahtjevi uređeni u skladu s obrascima uporabe	Has, Prpić	4.11.2019.
0.6	Dodana 3 sekvencijska dijagrama	Boras, Facković, Cvitković, Has, Prpić	6.11.2019.
0.6.1	Nadopunjeni obrasci upotrebe	Matošević	7.11.2019.
0.6.2	Dodan relacijski dijagram baze podataka	Matošević	11.11.2019.
0.6.3	Dodan dijagram i opis arhitekture na najvišoj razini	Matošević	12.11.2019.
0.6.4	Dodani opisi relacija baze podataka	Boras, Facković	12.11.2019.
0.6.5	Dodani dijagram razreda	Matošević	12.11.2019.
0.7	Nadopunjen dijagram razreda, opis arhitektura i manji ispravci	Matošević	14.11.2019.

Rev.	Opis promjene/dodatka	Autori	Datum
0.7.1	Dodan sekvencijski dijagram tijeka izvanrednog servisa	Cvitković	14.11.2019.
1.0	Verzija samo s bitnim dijelovima za 1. ciklus	Matošević	15.11.2019.
1.1	Dodana 3 dijagrama stanja i dijagram aktivnosti	Facković, Kolarec, Cvitković	18.12.2019.
1.2	Dodani opisi dijagrama	Boras	12.01.2020.
1.3	Dodan dijagram razmještaja i opis korištenih tehnologija	Matošević	13.01.2020.
1.4	Nadopunjen i izmijenjen dijagram razreda i opis	Prpić	15.01.2020.
1.4.1	Dodane poveznice za korištene tehnologije i alate	Prpić	15.01.2020.
1.5	Dodan opis ispitivanja sustava i ispitni slučajevi	Prpić	15.01.2020.
1.6	Dodan zaključak	Boras	15.01.2020.
1.6.1	Dodana slika rezultata Selenium ispitnih slučajeva	Prpić	15.01.2020.
1.6.2	Dodan opis ispitivanja komponenti i slike rezultata	Boras	15.01.2020.
1.7	Prepravljeni dijagrami razreda prema promjenama u kodu i izmijenjen dio opisa	Prpić	16.01.2020.
1.8	Dodane upute za puštanje u pogon	Matošević	16.01.2020.
1.8.1	Ažuriran dijagram razreda s opisom	Matošević, Boras, Prpić	16.01.2020.

## 2. Opis projektnog zadatka

Cilj ovog projektnog zadatka je osmisliti i razviti programsku potporu za web aplikaciju *Moj AutoServis*, koja će registriranim korisnicima aplikacije — vlasnicima automobila s jedne i auto servisima sa druge strane — omogućiti praćenje redovitog i izvanrednog servisiranja automobila. Aplikacija će brzo i efikasno povezati vlasnike automobila sa auto servisima te eliminirati potrebu za dugim i nepotrebnim čekanjem, odnosno skratiti vrijeme od narudžbe do realizacije.

Aplikaciju će moći koristiti samo registrirani korisnici. Registrirani korisnik prijavljuje se sa svojim jedinstvenim korisničkim imenom i lozinkom. Postoje tri vrste korisnika:

- vlasnik automobila,
- zaposlenik autoservisa: vlasnik autoservisa ili serviser, i
- administrator.

Kako bi se registrirao, vlasnik automobila mora aplikaciji dati sljedeće podatke:

- svoje ime,
- svoje prezime,
- OIB,
- adresu svoje e-pošte,
- željeno korisničko ime za prijavu, te
- lozinku

Auto servis registrira njegov vlasnik te pri tome upisuje podatke o auto servisu:

- naziv,
- adresa, i
- OIB tvrtke.

i podatke novog korisničkog računa vlasnika autoservisa:

- ime,
- prezime,
- korisničko ime, i

- lozinku

Nakon registracije, korisnik može dodavati nove automobile unoseći njihove podatke iz prometne dozvole, među kojima su obavezno:

- registracijska oznaka, te
- marka automobila.

Prilikom dodavanja novog automobila, uneseni podaci se provjeravaju u HUO registru, iz kojeg se dobiva marka, model, i broj šasije automobila. U slučaju nepodudaranja podataka vozila sa podacima u HUO registru, odbija se dodavanje vozila u aplikaciji. Korisnik može u bilo kojem trenutku dodati još automobila, te može unijeti podatke o neograničeno mnogo vozila. Ukoliko to želi, korisnik također ima mogućnost izbrisati podatke o automobilu.

Nakon registracije auto servisa, njegov vlasnik može nakon prijave u sustav upravljati svojim zaposlenicima, odnosno dodavati ih definiranjem korisničkog imena i inicijalne lozinke korisničkog računa novog zaposlenika, te ih brisati. Vlasnik auto servisa također unosi i cjenik usluga (radova na servisu) sa podacima:

- vrsta usluge, i
- cijena usluge,

te podatke o auto-dijelovima, koji se ugrađuju u servisu:

- naziv dijela,
- rok trajanja,
- predviđena kilometraža, i
- cijena dijela,

Kada korisnik (vlasnik automobila) želi poslati svoj automobil na servis, mora otvoriti radni nalog, prilikom čega na karti odabire auto servis u kojem želi obaviti servis uz mogućnost filtriranja po lokaciji. Potom odabire redoviti koji bi periodički trebao obavljati na automobilu, ili izvanredni koji se otvara u slučaju uočenog kvara. Nakon što je servis obavljen, korisniku dolaze detalji servisa s cijenom, te moguće preporuke o slanju na izvanredni servis.

Aplikacija nudi vlasniku automobila različite statističke podatke o troškovima, zamijenjenim dijelovima i drugim aspektima svih prijašnjih obavljenih servisa. Te informacije su dostupne sve dok se automobil ne obriše iz aplikacije.

Serviseri primaju radne naloge poslane njihovoj tvrtci. Ako ih prihvate, obavljaju servis, te u radni nalog dopisuju dodatne podatke o servisiranju, kao što su

kilometraža, i općenito stanje vozila, i, ako je na automobilu primijećena potreba za dodatnim popravcima, serviser daje preporuku za izvanredni servis. U slučaju izvanrednog servisa unose se podaci o kvarovima, popravcima (uslugama) kao i ugrađenim rezervnim dijelovima.

Aplikacija nudi auto servisu različite statičke podatke o troškovima servisa, stanju rezervnih dijelova, te finansijskim aspektima poslovanja. Također, nudi i statičke podatke o serviserima.

Glavni administrator ima upravu nad svim korisnicima, može brisati račune i ima pristup svim statističkim podacima koji se izračunavaju. Njegov korisnički račun nije moguće registrirati već se on direktno postavlja u bazi podataka.

## 2.1 Primjeri sličnih rješenja

Dvije aplikacije koje su najbližije ovoj su ShopBoss i GetAFix.

GetAFix je aplikacija za upravljanje auto servisom. Za razliku od projektne aplikacije, korisnika se traži da na slici automobila označi gdje postoje ogrebotine, te da procijeni stanje auta u raznim kategorijama (ulje, ...).



Slika 2.1: GetAFix

ShopBoss je software dizajniran za auto-servise. Ima više mogućnosti od aplikacije *Moj AutoServis*, te uz to ima mogućnost pristupa korisničkom računu sa bilo kojeg uređaja. U nju su također integrirani razni alati i web-stranice poput Carfax-a itd.





The screenshot shows the ShopBoss application interface. On the left is a 'Main Menu' sidebar with options like 'Work In Process', 'Create RO', 'Manage Customers', 'Schedule', 'Quotes', 'Dispatch', 'Find an RO', 'History', 'Follow Up', 'Receive PO', 'Inventory', 'Reports', 'Accounting', 'Settings', and 'Support'. The main area displays a table of vehicle inspections with columns: RO, STATUS, DATE, CUSTOMER, PHONE, VEHICLE, TOTAL, LICENSE, W+TK, and TYPE. The table contains 20 rows of inspection data.

RO	STATUS	DATE	CUSTOMER	PHONE	VEHICLE	TOTAL	LICENSE	W+TK	TYPE
2987	INSPECTION	10/22/2019		H: (860) 829-7377	2015 CHEVROLET SILVERADO 2500H4	1,128.67			ANOTHER TEST
2986	INSPECTION	10/17/2019	TEST 123 TEST123		#377 2013 HYUNDAI SANTA FE SPORT 2.	95.22			ANOTHER TEST
2985	INSPECTION	10/17/2019		H: (860) 829-7377	2011 JAGUAR XF PREMIUM	14.22	BKKA 198		ANOTHER TEST
2983	INSPECTION	10/15/2019	RODIO SALAZAR	C: (480) 524-2361	2001 GMC SIERRA 1500 C3 EXT. C	358.41		JCJC	ANOTHER TEST
2982	INSPECTION	10/15/2019		H: (860) 829-7377	2013 HYUNDAI ELANTRA GLS/UMIT	135.01	BJLM 182		ANOTHER TEST
2981	INSPECTION	10/14/2019	KONRAD CURZE		2004 VOLKSWAGEN PASSAT GL SEDA	443.98			ANOTHER TEST
2979	INSPECTION	10/10/2019	123 123		11 TOYOTA 4RUNNER	14.22			ANOTHER TEST
2978	INSPECTION	10/10/2019	RICKY BOBBY	C: (410) 687-5309	2001 SATURN L200	645.05	GHL9219		ANOTHER TEST
2977	INSPECTION	10/10/2019	JOHN BETA	C: (098) 765-4321	2009 FORD TAUROS SE AWD BLACK	90.30			ANOTHER TEST
2976	INSPECTION	10/09/2019	ROBOUTE GULLIMAN	C: (860) 555-1066	1965 VOLKSWAGEN BEETLE BLUE	14.22			ANOTHER TEST
2974	INSPECTION	10/09/2019	JOHN BETA	C: (098) 765-4321	2009 FORD TAUROS SE AWD BLACK	14.22			ANOTHER TEST
2973	INSPECTION	10/08/2019	ME		2006 VOLKSWAGEN NEW BEETLE 2.5	14.22			ANOTHER TEST
2972	INSPECTION	10/08/2019	KONRAD CURZE		2004 VOLKSWAGEN PASSAT GL SEDA	60.09			ANOTHER TEST
2971	INSPECTION	10/08/2019		H: (860) 829-7377	1993 OLDSMOBILE 88	14.22			ANOTHER TEST
2970	INSPECTION	10/08/2019	ROGAL DORN	C: (888) 555-1941	2006 AUDI A3 2.0T HATCHBACK 4-	223.53			ANOTHER TEST
2968	INSPECTION	10/07/2019	CORVUS CORAX	H: (555) 555-5555	2004 VOLKSWAGEN PASSAT GL SEDA	182.22			ANOTHER TEST
2964	INSPECTION	10/05/2019		H: (860) 829-7377	2013 HYUNDAI ELANTRA GLS/UMIT	14.22	BJLM 182		ANOTHER TEST
2963	INSPECTION	10/04/2019	BRIAN BARTLETT	H: (727) 417-3318	2012 GMC SIERRA 1500 SLT EXT.	211.78	AWBE24		ANOTHER TEST
2962	INSPECTION	10/03/2019	FIRST TRY	H: (555) 555-555 V	2004 ACURA RSX	310.42			ANOTHER TEST
2961	INSPECTION	10/03/2019	JOE MCNARSON	W: (917) 876-3211	2011 CHRYSLER TOWN & COUNTRY T	523.02	PLK-457U		ANOTHER TEST
2959	INSPECTION	09/27/2019	JEFF CHAMBERS	C: (443) 000-0000	2003 FORD EXPLORER BLUE	42.48	3719CB8		ANOTHER TEST
2958	INSPECTION	09/26/2019	SKYLINE			738.16			ANOTHER TEST
2956	INSPECTION	09/25/2019	123			14.22			ANOTHER TEST
2955	INSPECTION	09/25/2019	CELIO ARIAS	H: (317) 395-8571	2013 HONDA CIVIC SI SEDAN 6-SP	194.12			ANOTHER TEST

Slika 2.2: ShopBoss

Uz ove aplikacije, postoje mnoge koje su usmjerene samo na vlasnike automobila (npr. SimplyAuto i Fuelio), pružajući im jednostavno i efikasno praćenje stanja njihovih vozila, te one koje su usmjerene samo na auto-servise i pomažu im u organizaciji i provođenju servisa. Ova aplikacija ta će dva smjera spojiti u jedan, omogućavajući i vlasnicima automobila i auto servisima da komuniciraju i organiziraju servise automobila.

## 2.2 Moguće nadogradnje projektnog zadatka

Jedna od mogućih nadogradnji koje bi učinile ovu aplikaciju pristupačnijom je prilagodba iste za korištenje na mobilnim uređajima. Ovo bi bilo izrazito pogodno za slučajeve kada je korisniku potreban izvanredni servis zbog kvara koji se dogodio tijekom vožnje, npr. nakon sudara. Aplikacija bi mogla biti dalje nadograđena da nudi uslugu vučne službe, tj. da omogućuje korisniku da na karti označi gdje se nalazi sa svojim vozilom, koje bi potom bilo odvučeno na servis.

Još jedan potencijalni dodatak projektnom zadatku je omogućavanje korisniku da, nakon što doda novi automobil, ima pristup podacima o svim servisima nad tim automobilom obavljenim dok nije bio u njegovom vlasništvu.

Naknadno, moguće je nadograditi aplikaciju na način da svaki korisnik ima priliku ocijeniti kvalitetu usluge nakon obavljenog servisa. Tada bi se mogla izračunati prosječna ocjena dodijeljena pojedinom serviseru, te pojedinom auto servisu u cijelosti. Te bi ocjene bile vidljive korisniku dok odabire gdje želi da je njegov automobil servisiran.

Naposljetku, još jedna ideja za nadogradnju je mogućnost da vlasnik automobila može platiti za uslugu elektronički kroz aplikaciju, što bi uštedilo još vremena kod podizanja automobila iz auto servisa.

## 3. Specifikacija programske potpore

### 3.1 Funkcionalni zahtjevi

Dionici:

1. Naručitelj (FER)
2. Razvojni tim
3. Vlasnici automobila
4. Vlasnici autoservisa
5. Serviseri
6. Administrator

Aktori i njihovi funkcionalni zahtjevi:

1. Posjetitelj (inicijator) može:
  - (a) Registracija vlasnika automobila
  - (b) Registracija vlasnika autoservisa
  - (c) Pregled autoservisa
2. Registrirani korisnik (inicijator) može:
  - (a) Upravljanje korisničkim računom
    - i. Pregled korisničkih podataka
    - ii. Promjena korisničkih podataka
  - (b) Prijava u sustav
  - (c) Odjava
3. Vlasnik automobila (inicijator) može:
  - (a) Pregled autoservisa
  - (b) Upravljanje automobilima
    - i. Dodavanje automobila
    - ii. Pregled statističkih podataka za automobil
    - iii. Brisanje automobila

- (c) Upravljanje servisnim nalogima
  - i. Otvaranje novog servisnog naloga
  - ii. Pregled statusa servisnog naloga

4. Vlasnik autoservisa (inicijator) može:

- (a) Upravljanje rezervnim dijelovima
  - i. Dodavanje rezervnog dijela
  - ii. Uređivanje rezervnog dijela
  - iii. Brisanje rezervnog dijela
- (b) Upravljanje cjenikom usluga
  - i. Dodavanje usluge
  - ii. Uređivanje usluge
  - iii. Brisanje usluge
- (c) Uređivanje podataka autoservisa
- (d) Upravljanje serviserima
  - i. Dodavanje serviser
  - ii. Uklanjanje serviser

5. Serviser (inicijator) može:

- (a) Prihvatanje radnog naloga
- (b) Zatvaranje radnog naloga
- (c) Rad na radnom nalogu
  - i. Uređivanje naloga redovnog servisa
  - ii. Rad na nalogu izvanrednog servisa
    - A. Dodavanje ugrađenog rezervnog dijela
    - B. Brisanje ugrađenog rezervnog dijela
    - C. Dodavanje usluge
    - D. Brisanje usluge

6. Administrator (inicijator) može:

- (a) Administracija korisnika
  - i. Pregled podataka korisničkog računa
  - ii. Brisanje korisničkog računa
- (b) Administracija autoservisa
  - i. Pregled podataka autoservisa

ii. Brisanje autoservisa

7. HUO registar (sudionik) može:

(a) Provjera registracije i dohvat broja šasije

8. Baza podataka (sudionik) može:

(a) Komunikacija sa svim dijelovima sustava

(b) Pohrana podataka i ovlasti

### 3.1.1 Obrasci uporabe

#### UC1 -Registracija vlasnika automobila

- **Glavni sudionik:** Posjetitelj
- **Cilj:** Stvaranje novog korisničkog računa vlasnika automobila
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Posjetitelj odabire opciju registracije
  2. Posjetitelj unosi osobne podatke i nastavlja registraciju
  3. Sustav provjerava ispravnost te ih sprema u bazu podataka ako su ispravni
  4. Posjetitelju se prikazuje poruka o uspješnoj registraciji
- **Opis mogućih odstupanja:**
  - 3.a Posjetitelj je unio neispravan OIB, korisničko ime ili e-mail, ili postoji registrirani korisnik s nekim od tih podataka
    1. Posjetitelju se prikazuje poruka o neispravnosti/zauzetosti podataka

#### UC2 -Registracija vlasnika autoservisa

- **Glavni sudionik:** Posjetitelj
- **Cilj:** Stvaranje novog korisničkog računa vlasnika autoservisa
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Posjetitelj odabire opciju registracije autoservisa
  2. Posjetitelj unosi podatke o tvrtki te svoje korisničke podatke za vlasnika servisa
  3. Sustav provjerava upisane podatke te ih sprema u bazu podataka ako su ispravni
  4. Posjetitelju se prikazuje poruka o uspješnoj registraciji
- **Opis mogućih odstupanja:**
  - 3.a Posjetitelj je unio neispravan OIB ili korisničko ime ili već postoji tvrtka s nekim od tih podataka
    1. Posjetitelju se prikazuje poruka o neispravnosti/zauzetosti podataka

#### UC3 -Prijava u sustav

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Dobiti pristup korisničkom sučelju
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je registriran kao vlasnik automobila ili serviser
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju prijave u aplikaciju
  2. Korisnik upisuje korisničko ime i lozinku i nastavlja prijavu
  3. Sustav provjerava upisane podatke u bazi podataka
  4. Korisnika se preusmjerava na korisničko sučelje ovisno o tome je li vlasnik automobila ili serviser
- **Opis mogućih odstupanja:**
  - 3.a Korisnik je unio neispravno korisničko ime ili lozinku
    1. Korisniku se prikazuje poruka o neispravnosti, a korisnik može pokušati ispraviti podatke

#### UC4 -Odjava

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Odjaviti se iz korisničkog sučelja
- **Sudionici:** -
- **Preduvjet:** Korisnik je prijavljen u sustav
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju odjave iz aplikacije
  2. Sustav odjavljuje korisnika te ga preusmjeri na početnu stranicu

#### UC5 -Pregled autoservisa

- **Glavni sudionik:** Posjetitelj, vlasnik automobila
- **Cilj:** Pregled svih dostupnih autoservisa u aplikaciji
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Vlasnik automobila/posjetitelj odabire opciju za pregled svih autoservisa na početnom sučelju aplikacije
  2. Sustav dohvati iz baze listu svih registriranih autoservisa i prikaže ih

#### UC6 -Pregled korisničkih podataka

- **Glavni sudionik:** Registrirani korisnik

- **Cilj:** Pregled korisničkih podataka
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav
- **Opis osnovnog tijeka:**
  1. Korisnik klikne odabire opciju upravljanja korisničkim računom
  2. Sustav prikaže osnovne podatke korisnika, nakon što ih dohvati iz baze

#### **UC7 -Promjena korisničkih podataka**

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Promjena korisničkih podataka
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju upravljanja korisničkim računom
  2. Sustav dozvoli promjenu i prikaže sučelje za promjene korisničkih podataka
  3. Korisnik promijeni podatke i potvrđuje promjene lozinkom
  4. Sustav sprema nove podatke u bazu i prikazuje poruku za uspješnu promjenu podataka

#### **UC8 -Dodavanje automobila**

- **Glavni sudionik:** Vlasnik automobila
- **Cilj:** Dodavanje automobila
- **Sudionici:** Baza podataka, HUO registar
- **Preduvjet:** Korisnik je prijavljen u sustav kao vlasnik automobila
- **Opis osnovnog tijeka:**
  1. Vlasnik automobila odabire opciju upravljanja svojim automobila
  2. Sustav prikaže sučelje za dodavanje novog automobila
  3. Vlasnik automobila unosi registracijsku oznaku i marku automobila
  4. Sustav provjerava unesene podatke pomoću HUO registra i dohvaća broj šasije, te sprema podatke u bazu
  5. Vlasniku automobila se prikazuje poruka o uspješnom dodavanju automobila
- **Opis mogućih odstupanja:**
  - 4.a Vlasnik automobila je unio neispravnu registracijsku oznaku



1. Vlasniku automobila se prikazuje poruka o neispravnosti i može pokušati ispraviti podatke

#### **UC9 -Pregled statističkih podataka za automobil**

- **Glavni sudionik:** Vlasnik automobila
- **Cilj:** Pregled podataka o odabranom automobilu
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav kao vlasnik automobila
- **Opis osnovnog tijeka:**
  1. Vlasnik automobila odabire opciju upravljanja svojim automobilima
  2. Sustav prikaže sučelje s listom automobila
  3. Vlasnik automobila zatim odabere automobil i sustav prikaže podatke o tom automobilu, nakon što ih dohvati iz baze podataka

#### **UC10 -Brisanje automobila**

- **Glavni sudionik:** Vlasnik automobila
- **Cilj:** Brisanje automobila
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav kao vlasnik automobila
- **Opis osnovnog tijeka:**
  1. Vlasnik automobila odabire opciju upravljanja svojim automobilima
  2. Sustav prikaže sučelje s listom automobila
  3. Vlasnik automobila zatim odabere automobil koji će izbrisati
  4. U bazu podataka se pohranjuje promjena
  5. Sustav preusmjerava vlasnika automobila u prijašnje sučelje za upravljanje automobilima i prikazuje poruku za uspješno brisanje odabranog automobila
- **Opis mogućih odstupanja:**
  - 3.a Vlasnik automobila je odustao od brisanja
    1. Vlasniku automobila se prikazuje izbornik hoće li izbrisati odabrani automobil i preusmjerava ga u sučelje ovisno o odabiru
  - 3.b Postoji otvoren servisni nalog za automobil koji se želi obrisati
    1. Vlasniku automobila se prikazuje poruka o nemogućnosti brisanja odabranog automobila

#### **UC11 -Otvaranje novog servisnog naloga**

- **Glavni sudionik:** Vlasnik automobila
- **Cilj:** Otvaranje novog radnog naloga za servis automobila
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav kao vlasnik automobila
- **Opis osnovnog tijeka:**
  1. Vlasnik automobila odabire opciju upravljanja servisnim nalogima
  2. Sustav prikaže sučelje s listom radnih naloga
  3. Vlasnik automobila zatim odabire opciju otvaranja novog naloga i pri tom odabere za koji autoservis
  4. Nakon što je stvoren radni nalog, sprema se u bazu podataka i vlasniku se prikazuje poruka za uspješno kreiranje radnog naloga te se osvježava lista radnih naloga

#### UC12 -Pregled statusa servisnog naloga

- **Glavni sudionik:** Vlasnik automobila
- **Cilj:** Pregled statusa radnog naloga
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav kao vlasnik automobila
- **Opis osnovnog tijeka:**
  1. Vlasnik automobila odabire opciju upravljanja servisnim nalogima
  2. Sustav prikaže sučelje s listom radnih naloga dohvaćenih iz baze podataka i njihov status (otvoren/zatvoren/u obradi)

#### UC13 -Prihvaćanje radnog naloga

- **Glavni sudionik:** Serviser
- **Cilj:** Prihvaćanje radnog naloga
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav kao vlasnik autoservisa i radni nalog je otvoren
- **Opis osnovnog tijeka:**
  1. Vlasnik autoservisa odabire opciju prihvaćanja radnog naloga
  2. Sustav prikaže listu otvorenih radnih naloga dohvaćenih iz baze podataka
  3. Vlasnik prihvaća odabrani nalog
  4. Status radnog naloga se mijenja i sprema se u bazu, te postaje dostupan svim serviserima tog autoservisa

**UC14 -Zatvaranje radnog naloga**

- **Glavni sudionik:** Serviser
- **Cilj:** Zatvaranje radnog naloga
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav kao serviser i radni nalog je prihvaćen
- **Opis osnovnog tijeka:**
  1. Serviser odabire opciju rada na radnom nalogu
  2. Sustav prikaže listu dostupnih radnih naloga
  3. Serviser odabire opciju zatvaranja odabranog radnog naloga
  4. Status radnog naloga se promijeni u "zatvoren", sprema se u bazu i serviseru se prikaže poruka o uspješnom zatvaranju

**UC15 -Uređivanje naloga redovnog servisa**

- **Glavni sudionik:** Serviser
- **Cilj:** Uređivanje prihvaćenog radnog naloga
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav kao serviser, radni nalog je za redovan servis i prihvaćen
- **Opis osnovnog tijeka:**
  1. Serviser odabire opciju rada na radnom nalogu
  2. Sustav prikaže listu dostupnih radnih naloga
  3. Serviser klikne na uređivanje odabranog radnog naloga za redovan servis
  4. U radnom nalogu serviser unosi dodatne podatke (očitana kilometraža vozila te preporuka za izvanredni servis ukoliko je potrebno)
  5. Promjene se zatim spremaju i pohranjuju u bazu podataka

**UC16 -Dodavanje ugrađenog rezervnog dijela**

- **Glavni sudionik:** Serviser
- **Cilj:** Dodavanje ugrađenog rezervnog dijela u radni nalog
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav kao serviser, radni nalog je za izvanredan servis i prihvaćen
- **Opis osnovnog tijeka:**

1. Serviser odabire opciju za rad na radnom nalogu izvanrednog servisa
2. Na odabranom radnom nalogu, serviser odabire opciju dodavanja ugrađenog rezervnog dijela
3. Serviser odabere koji će rezervni dio unijeti
4. Promjene se zatim spremaju i pohranjuju u bazu podataka

#### **UC17 -Brisanje ugrađenog rezervnog dijela**

- **Glavni sudionik:** Serviser
- **Cilj:** Izbrisati odabrani ugrađeni rezervni dio u radnom nalogu
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav kao serviser, radni nalog je za izvanredan servis i prihvaćen
- **Opis osnovnog tijeka:**
  1. Serviser odabire opciju rada na radnom nalogu izvanrednog servisa
  2. Na odabranom radnom nalogu, serviser odabire opciju brisanja ugrađenog rezervnog dijela
  3. Serviser odabere koji će rezervni dio izbrisati
  4. Promjene se zatim spremaju i pohranjuju u bazu podataka

#### **UC18 -Brisanje usluge**

- **Glavni sudionik:** Serviser
- **Cilj:** Izbrisati uslugu
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav kao serviser, radni nalog je za izvanredan servis i prihvaćen
- **Opis osnovnog tijeka:**
  1. Serviser odabire opciju rada na radnom nalogu izvanrednog servisa
  2. Na odabranom radnom nalogu, serviser odabire opciju brisanja usluge
  3. Serviser odabere koju će uslugu obrisati
  4. Promjene se zatim spremaju i pohranjuju u bazu podataka

#### **UC19 -Dodavanje usluge**

- **Glavni sudionik:** Serviser
- **Cilj:** Dodati uslugu
- **Sudionici:** Baza podataka

- **Preduvjet:** Korisnik je prijavljen u sustav kao serviser, radni nalog je za izvanredan servis i prihvaćen
- **Opis osnovnog tijeka:**
  1. Serviser odabire opciju rada na radnom nalogu izvanrednog servisa
  2. Na odabranom radnom nalogu, serviser odabire opciju dodavanja usluge
  3. Serviser odabere uslugu za radni nalog
  4. Promjene se zatim spremaju i pohranjuju u bazu podataka

#### UC20 -Dodavanje rezervnog dijela

- **Glavni sudionik:** Vlasnik autoservisa
- **Cilj:** Dodavanje novog rezervnog dijela za ugradnju
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav i dodijeljeno mu je pravo vlasnika autoservisa
- **Opis osnovnog tijeka:**
  1. Vlasnik autoservisa odabire opciju upravljanja rezervnim dijelovima
  2. Vlasnik autoservisa odabire opciju dodavanja novog rezervnog dijela
  3. Vlasnik upisuje naziv i cijenu rezervnog dijela i potvrđuje upis
  4. U bazu podataka se pohrani promjena
- **Opis mogućih odstupanja:**
  - 3.a Vlasnik je unio neispravnu cijenu rezervnog dijela
    1. Vlasniku se prikazuje poruka o neispravnosti, a vlasnik je ispravlja i nastavlja s potvrdom

#### UC21 -Uređivanje rezervnog dijela

- **Glavni sudionik:** Vlasnik autoservisa
- **Cilj:** Uređivanje postojećeg rezervnog dijela za ugradnju
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav i dodijeljeno mu je pravo vlasnika autoservisa
- **Opis osnovnog tijeka:**
  1. Vlasnik autoservisa odabire opciju upravljanja rezervnim dijelovima
  2. Vlasnik autoservisa iz popisa rezervnih dijelova odabere opciju uređivanja
  3. Vlasniku se prikazuje prozor s upisanim postojećim podacima o odabranom dijelu
  4. Vlasnik mijenja podatke i potvrđuje promjene

5. U bazu podataka se pohrani promjena

- **Opis mogućih odstupanja:**

4.a Vlasnik je unio neispravnu cijenu rezervnog dijela

1. Vlasniku se prikazuje poruka o neispravnosti, a vlasnik je ispravlja i nastavlja s potvrdom

### **UC22 -Brisanje rezervnog dijela**

- **Glavni sudionik:** Vlasnik autoservisa
- **Cilj:** Brisanje rezervnog dijela
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav i dodijeljeno mu je pravo vlasnika autoservisa
- **Opis osnovnog tijeka:**
  1. Vlasnik autoservisa odabire opciju upravljanja rezervnim dijelovima
  2. Vlasnik autoservisa iz popisa rezervnih dijelova odabere rezervni dio za brisanje
  3. U bazi podataka se odabrani dio označava kao nedostupan za daljnju ugradnju

### **UC23 -Dodavanje usluge u cjenik**

- **Glavni sudionik:** Vlasnik autoservisa
- **Cilj:** Dodavanje novog usluge u cjenik
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav i dodijeljeno mu je pravo vlasnika autoservisa
- **Opis osnovnog tijeka:**
  1. Vlasnik autoservisa odabire opciju upravljanja cjenikom usluga
  2. Vlasnik autoservisa odabire opciju dodavanja nove usluge
  3. Vlasnik upisuje naziv i cijenu usluge i potvrđuje upis
  4. U bazu podataka se pohrani promjena
- **Opis mogućih odstupanja:**
  - 3.a Vlasnik je unio neispravnu cijenu usluge
    1. Vlasniku se prikazuje poruka o neispravnosti, a vlasnik je ispravlja i nastavlja s potvrdom

### **UC24 -Uređivanje usluge u cjeniku**

- **Glavni sudionik:** Vlasnik autoservisa
- **Cilj:** Uređivanje postojećeg usluge u cjeniku
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav i dodijeljeno mu je pravo vlasnika autoservisa
- **Opis osnovnog tijeka:**
  1. Vlasnik autoservisa odabire opciju upravljanja cjenikom usluga
  2. Vlasnik autoservisa iz popisa usluga u cjeniku odabere opciju uređivanja
  3. Vlasniku se prikazuje prozor s upisanim postojećim podacima o odabranoj usluzi
  4. Vlasnik mijenja podatke i potvrđuje promjene
  5. U bazu podataka se pohrani promjena
- **Opis mogućih odstupanja:**
  - 4.a Vlasnik je unio neispravnu cijenu usluge
    1. Vlasniku se prikazuje poruka o neispravnosti, a vlasnik je ispravlja i nastavlja s potvrdom

#### UC25 -Brisanje usluge iz cjenika

- **Glavni sudionik:** Vlasnik autoservisa
- **Cilj:** Brisanje usluge iz cjenika
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav i dodijeljeno mu je pravo vlasnika autoservisa
- **Opis osnovnog tijeka:**
  1. Vlasnik autoservisa odabire opciju cjenikom usluga
  2. Vlasnik autoservisa iz cjenika usluga odabire uslugu za brisanje
  3. U bazi podataka se odabrani usluga označava kao nedostupan za daljnju dodavanje na radne naloge

#### UC26 -Uređivanje podataka autoservisa

- **Glavni sudionik:** Vlasnik autoservisa
- **Cilj:** Promjena naziva i ostalih podataka autoservisa
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav i dodijeljeno mu je pravo vlasnika autoservisa
- **Opis osnovnog tijeka:**

1. Vlasnik autoservisa odabire opciju uređivanja podataka autoservisa
2. Vlasnik autoservisa se prikazuju trenutni podaci o njegovom autoservisu (naziv)
3. Vlasnik autoservisa mijenja podatke i potvrđuje promjenu
4. Promjena se pohranjuje u bazu podataka

#### UC27 -Pregled statistike autoservisa

- **Glavni sudionik:** Vlasnik autoservisa
- **Cilj:** Pregled statističkih podataka autoservisa
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav i dodijeljeno mu je pravo vlasnika autoservisa
- **Opis osnovnog tijeka:**
  1. Vlasnik autoservisa odabire opciju pregleda statistike
  2. Vlasnik autoservisa se prikazuju različiti statistički podaci o radu servisa (zarada, broj popravljenih automobila i ostalo)

#### UC28 -Dodavanje serviser

- **Glavni sudionik:** Vlasnik autoservisa
- **Cilj:** Dodavanje novog serviser u vlastiti autoservis
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav i dodijeljeno mu je pravo vlasnika autoservisa
- **Opis osnovnog tijeka:**
  1. Vlasnik autoservisa odabere opciju upravljanja serviserima
  2. Vlasnik autoservisa odabere opciju dodavanja novog serviser
  3. Vlasnik autoservisa upisuje ime i prezime te korisničko ime i inicijalnu lozinku za korisnički račun novog serviser
  4. Novi račun se pohranjuje u bazu podataka
- **Opis mogućih odstupanja:**
  - 4.a Postoji korisnički račun s istim korisničkim imenom
    1. Vlasniku se prikazuje poruka o zauzetosti korisničkog imena, a vlasnik bira drugo korisničko ime

#### UC29 -Uklanjanje serviser

- **Glavni sudionik:** Vlasnik autoservisa



- **Cilj:** Uklanjanje serviserima iz vlastitog autoservisa
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav i dodijeljeno mu je pravo vlasnika autoservisa
- **Opis osnovnog tijeka:**
  1. Vlasnik autoservisa odabere opciju upravljanja serviserima
  2. Vlasnik autoservisa odabere iz popisa odabire serviserima kojeg želi ukloniti
  3. Korisnički račun serviserima se briše iz baze podataka

#### UC30 -Pregled podataka korisničkog računa

- **Glavni sudionik:** Administrator
- **Cilj:** Pregled podataka odabranog korisničkog računa
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav i ima ulogu administratora
- **Opis osnovnog tijeka:**
  1. Administrator odabere opciju upravljanja računima
  2. Sustav prikaže listu korisničkih računa
  3. Administrator odabere korisnički račun čije podatke želi pregledati
  4. Administratoru se prikažu korisnički podaci dohvaćeni iz baze podataka

#### UC31 -Brisanje korisničkog računa

- **Glavni sudionik:** Administrator
- **Cilj:** Brisanje odabranog korisničkog računa
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav i ima ulogu administratora
- **Opis osnovnog tijeka:**
  1. Administrator odabere opciju upravljanja serviserima
  2. Sustav prikaže listu korisničkih računa
  3. Administrator iz popisa odabere korisnički račun kojeg želi ukloniti
  4. Korisnički račun briše se iz baze podataka

#### UC32 - Brisanje autoservisa

- **Glavni sudionik:** Administrator
- **Cilj:** Brisanje odabranog autoservisa
- **Sudionici:** Baza podataka

- **Preduvjet:** Korisnik je prijavljen u sustav i ima ulogu administratora
- **Opis osnovnog tijeka:**
  1. Administrator odabere opciju upravljanja autoservisima
  2. Sustav prikaže listu autoservisa
  3. Administrator iz popisa odabere autoservis kojeg želi ukloniti
  4. Autoservis se briše iz baze podataka

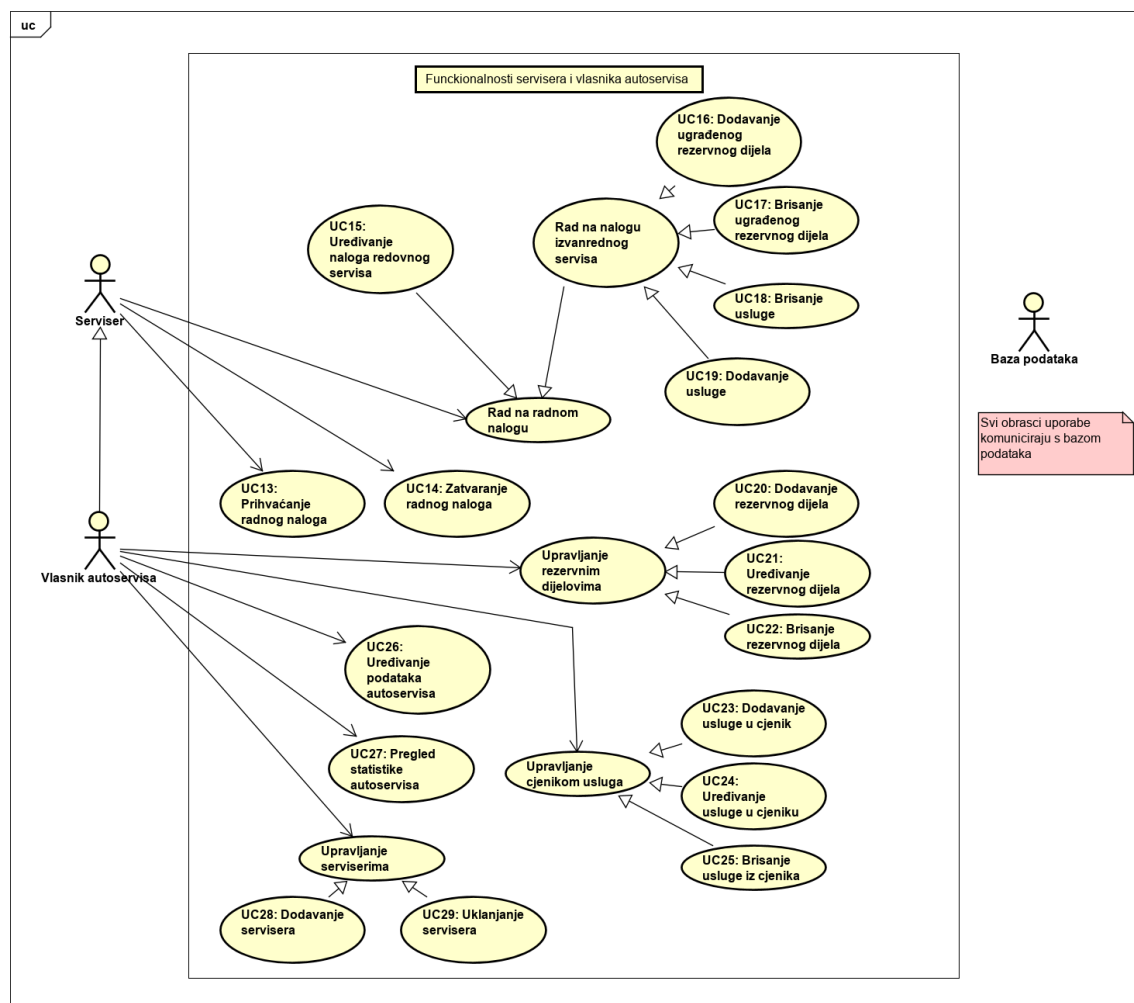
### **UC33 -Pregled podataka i statistike autoservisa**

- **Glavni sudionik:** Administrator
- **Cilj:** Pregled podataka odabranog autoservisa
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav i ima ulogu administratora
- **Opis osnovnog tijeka:**
  1. Administrator odabere opciju upravljanja autoservisima
  2. Sustav prikaže listu autoservisa
  3. Administrator odabire autoservis čije podatke želi pregledati
  4. Administratoru se prikazuju osnovni i statistički podaci o autoservisu dohvaćeni iz baze podataka

## Dijagrami obrazaca uporabe



Slika 3.1: Dijagram obrasca uporabe, funkcionalnost posjetitelja, registriranog korisnika i vlasnika automobila



Slika 3.2: Dijagram obrasca uporabe, funkcionalnost servisera i vlasnika autoservisa

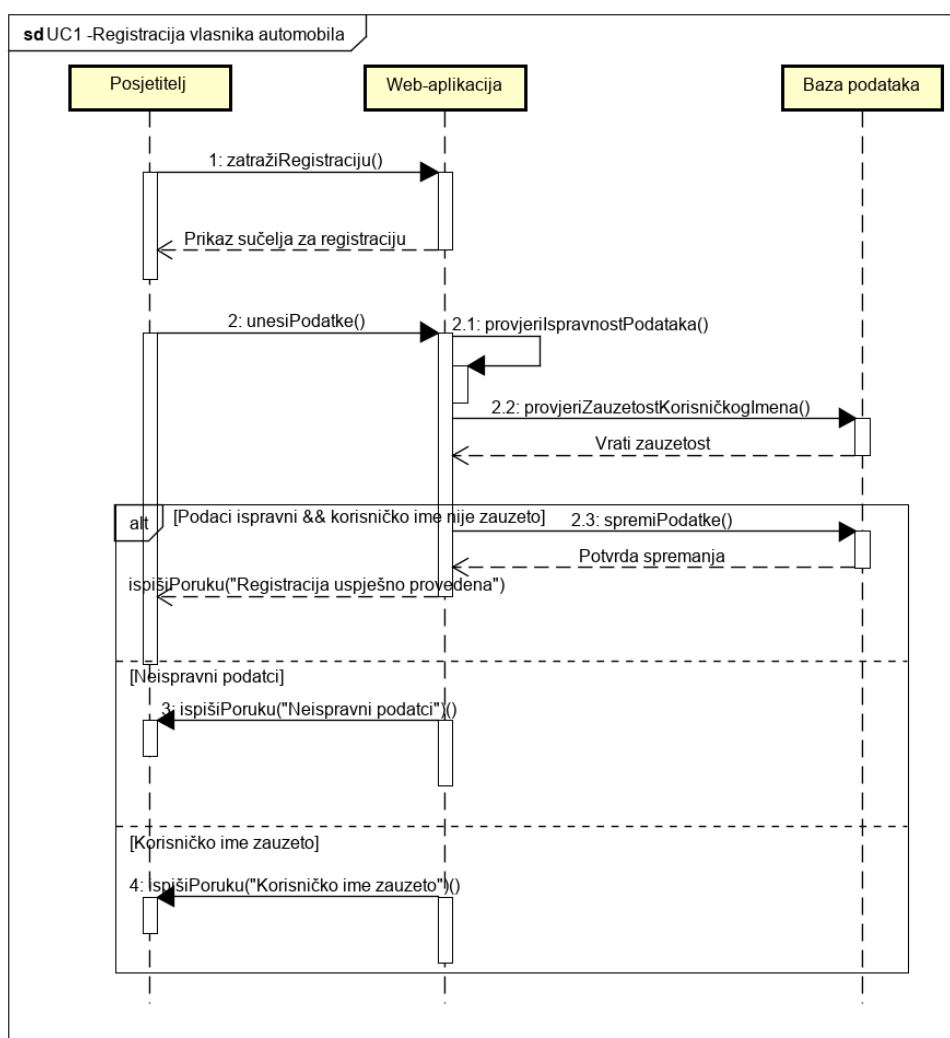


Slika 3.3: Dijagram obrasca uporabe, funkcionalnost administratora

### 3.1.2 Sekvencijski dijagrami

#### Obrazac uporabe UC1: Registracija vlasnika automobila

Posjetitelj na početnoj stranici odabire opciju registracije. Poslužitelj posjetitelju prikazuje sučelje za registraciju. Posjetitelj zatim unosi osobne podatke i nastavlja registraciju. Sustav provjerava ispravnost podataka te ih sprema u bazu podataka ako su ispravni. Posjetitelju se na kraju prikazuje poruka o uspješnoj registraciji. Ukoliko je posjetitelj unio neispravan OIB, korisničko ime ili e-mail, ili postoji registrirani korisnik s nekim od tih podataka, posjetitelju se prikazuje poruka o neispravnosti/zauzetosti podataka.



Slika 3.4: Sekvencijski dijagram za UC1

**Obrazac uporabe UC7: Promjena korisničkih podataka**

Korisnik odabire opciju upravljanja korisničkim računom, na što mu aplikacija dozvoljava promjenu, iz baze podataka dohvaća korisničke podatke te korisniku prikazuje sučelje za promjenu korisničkih podataka. Korisnik mijenja podatke i potom promjene potvrđuje unosom lozinke. Aplikacija provjerava točnost lozinke te ukoliko je ona ispravna i ako su novi podatci ispravni, novi podatci se spremaju u bazu podataka i korisniku se prikazuje poruka o uspješnoj promjeni podataka. Ukoliko je unesena lozinka neispravna ili ukoliko su novi podatci neispravni, korisniku se prikazuje poruka o neispravnosti unesene lozinke ili unesениh podataka.



Slika 3.5: Sekvencijski dijagram za promjenu korisničkih podataka, UC7

**Obrasci uporabe UC12–UC15: Upravljanje radnim nalogima**

Vlasnik automobila (korisnik) odabire opciju upravljanja radnim nalogima, na što mu aplikacija vraća popis radnih naloga te njihov status. Tada korisnik otvara novi radni nalog koji se sprema u bazu podataka, te se uspješno spremanje javlja korisniku. Web aplikacija osvježava listu radnih naloga koja se prikazuje korisniku. Serviser javlja aplikaciji da može raditi na radnom nalogu, te se iz baze podataka tada dohvaća lista otvorenih radnih naloga, od kojih on odabire jedan. Tom radnom nalogu mijenja se status u *u obradi*, te se novi status potom ažurira u bazi podataka. Serviser tijekom servisiranja upisuje kilometražu te daje preporuku za izvanredan servis. Te podatke aplikacija ažurira u bazi podataka. Nakon što serviser u aplikaciji zatvori radni nalog, status naloga se još jednom ažurira na *zatvoren* u bazi podataka.





Slika 3.6: Sekvencijski dijagram za tijek redovnog servisa, UC12–UC15

**Obrasci uporabe UC11-14 i UC16: Upravljanje izvanrednim servisima**

Korisnik odabire opciju upravljanja radnim nalogima, na što mu aplikacija vraća popis radnih naloga te njihov status. Tada korisnik otvara novi radni nalog koji se sprema u bazu podataka, te se uspješno spremanje javlja korisniku. Web aplikacija osvježava listu radnih naloga koja se prikazuje korisniku. Serviser zatraži listu otvorenih naloga izvanrednog servisa te prihvća jednog od njih. Tom se radnom nalogu mijenja status u *u obradi*, te se novi status potom ažurira u bazi podataka. Serviser zatim unese opis kvarova. Ako se automobilu ugradio rezervni dio serviser ih dodaje jedan po jedan i sprema promjene u bazu, a ako je obavljena neka usluga dodaje se opis popravka. Serviser ovo radi dok god još ima popravaka za obaviti nad automobilom. Promjene se ažuriraju. Nakon što serviser u aplikaciji zatvori radni nalog, status naloga se još jednom ažurira na *zatvoren* u bazi podataka.



Slika 3.7: Sekvencijski dijagrami tijeka izvanrednog servisa

## 3.2 Ostali zahtjevi

- Sustav treba biti implementiran u objektno-orijentiranom jeziku te omogućiti laku nadogradnju u slučaju potrebe
- Sustav treba biti implementiran kao web-aplikacija
- Korisničko sučelje mora biti responzivno te se ispravno prikazivati u preglednicima na desktop računalima i mobilnim uređajima
- Sučelje za vlasnike automobila treba biti jednostavno za korištenje osobama svih dobni skupina
- Sučelje za servisere treba biti prilagođeno za uporabu u okruženju servisa i osigurati laku dostupnost svih funkcionalnosti
- Sustavu mogu istovremeno pristupiti više različitih vrsta korisnika
- Aplikacija mora podržavati hrvatsku abecedu pri unosu podataka
- Sustav koristi HRK kao valutu za izračun i prikaz cijena
- Pristup sustavu mora biti osiguran putem protokola HTTPS
- Podaci o automobilu se na temelju registracije trebaju provjeravati i dohvaćati iz HUO registra
- Izvršavanje dijelova aplikacije (uključujući i pristup HUO registru) mora se odvijati u roku manjem od nekoliko sekundi

## 4. Arhitektura i dizajn sustava

Jedan od najvažnijih koraka pri razvoju sustava je izbor i oblikovanje arhitekture sustava. Kako će našu aplikaciju koristiti više različitih korisnika istovremeno (vlasnici automobila, autoservisi i administratori), te stoga što autoservisi koriste različite uređaje kojima će pristupati sustavu, odlučen je razvoj web aplikacije. Time će se sustavu moći pristupiti sa svih uređaja (mobilni uređaji, tableti, desktop računala) bez obzira na platformu uz uvjet da na uređaju postoji odgovarajući web preglednik čime se želi i privući što veći broj vlasnika automobila kao korisnika aplikacije.

Na slici 4.1 prikazana je klijent-poslužitelj arhitektura na kojoj će se temeljiti sustav. Korisnici aplikacije putem svojih web preglednika (putem *HTTP* ili *HTTPS* protokola) pristupaju poslužitelju na kojem je pokrenuta web aplikacija koja obrađuje njihove zahtjeve i komunicira s bazom podataka (pokrenutoj na istom ili nekom drugom poslužitelju) putem *JDBC* protokola.

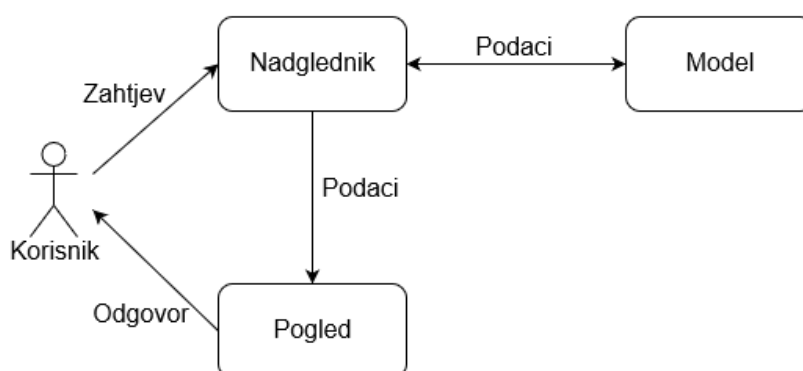


Slika 4.1: Osnovna arhitektura cijelog sustava

Kao tip arhitekture uzet će se objektno usmjerena arhitektura budući da najbolje odgovara razvoju složene višekorisničke aplikacije.

Pri razvoju će se pratiti Model-pogled-nadglednik obrazac (engl. *MVC: Model-View-Controller*) prikazan na slici 4.2 koji dijeli sustav na tri komponente:

- Model - komponenta koja je zadužena za upravljanje podacima te uključuje sve razrede čiji se podaci obrađuju
- Pogled - komponenta koja služi za prikaz podataka (modela) korisniku
- Nadglednik - komponenta koja prima i obrađuje korisničke zahtjeve na način da dohvaća, provjerava i mijenja podatke u modelu te ih šalje pogledu za prikaz



Slika 4.2: Oblikovni obrazac Model-pogled-nadglednik

Korištenjem Model-pogled-nadglednik oblikovnog obrasca smanjit će se ovisnost između korisničkog sučelja i ostatka sustava, a zbog podjele odgovornosti između komponenti i smanjene međuovisnosti olakšat će se paralelni razvoj, testiranje i moguće buduće promjene u sustavu.

Arhitektura aplikacije se može podijeliti i na *backend* i *frontend* sloj. *Backend* sloj izvršavat će se na poslužitelju (i biti ravnjen po *MVC* obrascu) te će rezultat obrade klijentskih zahtjeva biti grafički prikaz stranice. Takav prikaz u jezicima *HTML*, *CSS* i *JavaScript* se šalje klijentu i predstavlja *frontend* sloj.

Prema pogledu korisnika (aktora) koji će koristiti sustav, on se može podijeliti na tri manja podsustava što je vidljivo na slici 4.1:

- Aplikacija za vlasnike automobila
- Aplikacija za autoservise
- Aplikacija za administraciju

Sustav će biti implementiran u programskom jeziku *Java* korištenjem *Spring* okvira uz potrebna proširenja. To je okvir koji nudi dio poslužiteljske implementacije web aplikacije slijedeći obrazac Model-pogled-nadglednik. Uz to, slijedi princip

inverzije ovisnosti koji će olakšati kasnije nadogradnje i promjene smanjivanjem ovisnosti te nudi gotovu implementaciju različitih protokola i automatsku postavljanje autentikacije. Time će se ubrzati razvoj, a dodatno će se uz njega koristiti i okvir za pristup podacima *Spring JPA* te okvir *Thymeleaf* namijenjen generiranju *HTML* koda iz definiranih predložaka.

## 4.1 Baza podataka

Sustav tijekom svojeg rada sprema i dohvaća različite podatke, koji će biti pohranjeni u bazu podataka. U bazi podataka se ti podaci spremaju u relacije (tablice) s definiranim atributima. Baza treba osigurati konzistentnost podataka kroz definirana ograničenja te primarne i strane ključeve, mogućnost istovremenog pristupa podacima te izdržljivost i mogućnost oporavka podataka u slučaju kvara.

Za preslikavanje modela razreda u relacijski model podataka koristi se biblioteka Hibernate unutar okvira *Spring JPA*. To je ORM (*Object-Relational Mapping*) okvir koji automatski preslikava razrede i njihove veze u relacijski model te generira i provodi SQL upite koje sam obrađuje.

U razvojnoj okolini se koristi *H2* baza podataka koja se čuva u memoriji za vrijeme pokretanja aplikacije te omogućava lako testiranje i dodavanje podataka. Ona je odabrana jer za nju već postoji podrška u *Spring* okviru koji se koristi.

Sustav će u produkciji koristiti *PostgreSQL* bazu podataka, koja je besplatna i otvorenog koda. Ona je također dobro podržana u *Spring* okviru, ima različite mogućnosti replikacije podataka koje bi se kasnije mogle implementirati i podržava većinu operativnih sustava. Ona može biti pokrenuta na istom poslužitelju na kojem je pokrenut i sustav ili na nekom drugom, a sustav se na nju povezuje putem JDBC protokola.

### 4.1.1 Opis tablica

**Napomena** Atributi koji su **podebljani** označavaju primarne ključeve, a oni koji su podcrtani označavaju strane.

**app\_user** Ovaj entitet sadrži sve podatke o registriranom korisniku aplikacije te predstavlja vlasnika automobila, serviseru i administratora ovisno o sadržaju atributa *dtype*. Atributi koje sadrži su: tip, jedinstveni identifikacijski broj, e-mail adresa, lozinka, korisničko ime, ime, prezime i OIB korisnika, te tip zaposlenika u autoservisu i jedinstveni identifikacijski broj autoservisa. Ovisno o tipu korisnika, neki atributi će imati vrijednost *NULL* ako se na njih ne odnosi taj atribut. Entitet je u *One-to-Many* vezi s entitetom *vehicle* preko atributa identifikacijski broj korisnika. Također je u vezi *Many-to-One* sa entitetom *auto\_service* preko jedinstvenog identifikatora autoservisa.



app_user (Korisnik)		
dtype	VARCHAR	Tip korisnika: vlasnik vozila, zaposlenik autoservisa ili administrator.
user_id	BIGINT	Jedinstveni identifikacijski broj korisnika.
email	VARCHAR	E-mail adresa korisnika.
first_name	VARCHAR	Ime korisnika.
last_name	VARCHAR	Prezime korisnika.
email	VARCHAR	E-mail adresa korisnika.
password_hash	VARCHAR	Lozinka korisnika.
username	VARCHAR	Korisničko ime korisnika.
oib	VARCHAR	OIB korisnika.
employee_type	VARCHAR	Tip zaposlenika u autoservisu - serviser ili vlasnik autoservisa.
auto_service_id	BIGINT	Jedinstveni identifikacijski broj autoservisa.

**vehicle** Ovo je entitet koji sadrži podatke o vozilu. Atributi su mu: jedinstveni identifikator, registarska oznaka, model i broj šasije vozila, te identifikacijski broj vlasnika vozila. Ovaj entitet je u vezi *Many-to-One* s entitetom app\_user preko atributa identifikacijski broj vlasnika vozila. Također je u *One-to-Many* vezi s entitetom repair\_order preko identifikacijakog broja vozila.

vehicle (Vozilo)		
vehicle_id	BIGINT	Jedinstveni identifikator vozila.
license_plate	VARCHAR	Registarska oznaka vozila.
vehicle_model	VARCHAR	Model vozila.
vin_number	VARCHAR	Broj šasije vozila.
owner_user_id	BIGINT	Identifikacijski broj vlasnika vozila.

**repair\_order** Ovo je entitet koji sadrži sve važne podatke o pojedinom radnom nalogu. Sadrži attribute: jedinstveni identifikacijski broj radnog naloga, identifikacijski broj autoservisa i vozila, te ukupna cijena i status radnog naloga (otvoren, zatvoren ili u obradi). Entitet je u vezi *Many-to-One* s entitetom vehicle preko atributa identifikacijski broj vozila. Također je u *Many-to-One* vezi s entitetom auto\_service preko identifikacijskog broja autoservisa. Ovaj entitet je generalizacija entiteta regular\_repair\_order i entiteta repairing\_repair\_order.

<b>repair_order (Radni nalog)</b>		
<b>repair_order_id</b>	BIGINT	Jedinstveni identifikacijski broj radnog naloga.
price	DOUBLE PRECISION	Ukupna cijena.
service_job_ status	VARCHAR	Status radnog naloga: otvoren, zatvoren ili u obradi.
<u>auto_service_id</u>	BIGINT	Identifikacijski broj autoservisa.
<u>vehicle_id</u>	BIGINT	Identifikacijski broj vozila.

**auto\_service** Ovaj entitet sadrži sve važne informacije o autoservisu. Atributi koje sadrži su: jedinstveni identifikator, ime, adresa i OIB autoservisa, te cijena redovitog servisa. Ovaj entitet povezan je vezom *One-to-Many* preko atributa jedinstveni identifikator autoservisa s entitetima: repair\_order, app\_user, service\_labor i vehicle\_part.

<b>auto_service (Autoservis)</b>		
<b>auto_service_id</b>	BIGINT	Jedinstveni identifikator autoservisa.
address	VARCHAR	Adresa autoservisa.
oib	VARCHAR	OIB autoservisa.
regular_service_ _price	DOUBLE PRECISION	Cijena redovitog servisa.
shop_name	VARCHAR	Naziv autoservisa.

**regular\_repair\_order** Ovaj entitet sadržava sve važne informacije o radnom nalogu koji se provodi za redoviti servis. Sadrži attribute: kilometražu vozila, primijećene kvarove, preporuku za izvanredni servis te ID radnog naloga kojem pripada. Ovaj entitet je specijalizacija entiteta repair\_order.

<b>regular_repair_order (Radni nalog za redoviti servis)</b>		
kilometers	INTEGER	Kilometraža vozila zabilježena na servisu.
observed_ malfunctions	VARCHAR	Primijećeni kvarovi.
repair_re- commended	BOOLEAN	Je li potreban izvanredni servis ili ne?
<u>id</u>	BIGINT	Identifikacijski broj radnog naloga.

**repairing\_repair\_order** Ovaj entitet sadrži sve važne informacije za izvanredni ser-

vis vozila. Sadrži attribute: kvarove na vozilu i ID radnog naloga kojem pripada. Ovaj entitet je specijalizacija entiteta `repair_order`.

<b>repairing_repair_order (Radni nalog za izvanredni servis)</b>		
<code>malfunctions</code>	VARCHAR	Kvarovi na vozilu.
<u><code>id</code></u>	BIGINT	Identifikacijski broj radnog naloga.

**repair\_order\_item** Ovaj entitet sadrži sve važne informacije za pojedinu stavku na radnom nalogu. Atributi ovog entiteta su: identifikator stavke na radnom nalogu, njen naziv, cijena te ID izvanrednog radnog naloga kojem stavka pripada. Ovaj entitet je u vezi *Many-to-One* sa entitetom `repair_order` preko ID-a tog radnog naloga.

<b>repair_order_item (Stavka na radnom nalogu)</b>		
<u><code>item_id</code></u>	BIGINT	Identifikator stavke.
<code>name</code>	VARCHAR	Naziv stavke.
<code>price</code>	DOUBLE PRECISION	Cijena stavke.
<u><code>repair_order_id</code></u>	BIGINT	Identifikator izvanrednog radnog naloga kojem pripada popravak.

**service\_labor** Ovaj entitet sadrži sve informacije važne za servisnu uslugu. Atributi ovog entiteta su: ID usluge, cijena usluge, naziv usluge i ID autoservisa koji pruža uslugu. Ovaj entitet je u vezi *Many-to-One* sa entitetom `auto_service` preko ID-a autoservisa.

<b>service_labor (Servisna usluga)</b>		
<u><code>service_labor_id</code></u>	BIGINT	Identifikator usluge.
<code>price</code>	DOUBLE PRECISION	Cijena usluge.
<code>service_name</code>	VARCHAR	Naziv usluge.
<u><code>auto_service_id</code></u>	BIGINT	Identifikator autoservisa koji pruža uslugu.

**vehicle\_part** Ovaj entitet sadrži sve važne informacije za rezervni dio automobila. Ovaj entitet sadrži attribute: ID dijela, kilometraža za koju je dio predviđen, naziv dijela, njegova cijena te ID autoservisa koji ima rezervni dio. Ovaj entitet je u vezi *Many-to-One* sa entitetom `auto_service` preko ID-a tog autoservisa.

<b>vehicle_part (Rezervni dio)</b>		
<b>part_id</b>	BIGINT	Identifikator rezervnog dijela.
estimated_ dura- tion in km	INTEGER	Kilometraža za koju je predviđen rezervni dio.
part_name	VARCHAR	Naziv rezervnog dijela.
price	DOUBLE PRECISION	Cijena rezervnog dijela.
<u>auto_service_id</u>	BIGINT	Identifikator autoservisa koji ima rezervni dio.

#### 4.1.2 Dijagram baze podataka



Powered by yFiles

Slika 4.3: Relacijski model baze podataka

## 4.2 Dijagram razreda

Na slici 4.4 je prikazan dijagram razreda koji pripadaju sloju modela. Razred *AppUser* predstavlja registriranog korisnika koji nakon unosa svojih podataka može koristiti osnovne funkcionalnosti aplikacije. Njega nasljeđuje razred *Administrator* koji predstavlja administratora koji ima najveće ovlasti, razred *VehicleOwner* koji predstavlja registriranog vlasnika automobila, razred *ServiceEmployee* predstavlja osobu zaposlenu u auto servisu, koja može biti vlasnik tog auto servisa ili serviser ovisno ovisno o tipu (enumeracija *ServiceEmployeeType*) te razred *CurrentUser* u koji se pohranjuju podatci o trenutnom korisniku.

Razred *Vehicle* predstavlja vozilo koje pripada nekom korisniku. Razred *AutoService* označava jedan auto servis. Razred *RepairOrder* predstavlja radni nalog za servis, ima trenutni status koji prikazuje je li nalog u čekanju, odbijen, prihvaćen ili gotov (enumeracija *ServiceJobStatus*), te ima dva podrazreda, razred *RegularRepairOrder* koji predstavlja radni nalog za redovni servis, te razred *RepairingRepairOrder* koji predstavlja radni nalog za izvanredni servis. Razred *RepairOrderItem* predstavlja konkretnu stavku (ugrađeni rezervni dio ili obavljenju uslugu) s cijenom na izvanrednom servisu. Razredi *VehiclePart* i *ServiceLabor* implementiraju sučelje *Product* koji predstavlja proizvod ili uslugu. Razred *VehiclePart* predstavlja rezervni dio za automobil, a razred *ServiceLabor* predstavlja uslugu koju serviser može obaviti na automobilu te oni pripadaju određenom autoservisu te predstavljaju stavke koje se mogu dodati na radne naloge izvanrednog servisa.



Razred *MainController* nudi operacije povezane s prikazom početne stranice, registracijom, te općenitom promjenom korisničkih podataka i lozinke. Za te operacije ovisi o sučeljima *VehicleOwnerService*, *AutoServiceService* i *UserService*.

Razred *VehicleOwnerController* nudi operacije povezane s vlasnikom vozila. On koristi operacije koje nude sučelja *AutoServiceService*, *VehicleService* i *RepairOrder-*

*Service.*

Razred *AutoServiceController* nudi operacije povezane sa serviserom. On koristi sučelja *RepairOrderService*, *VehiclePartService*, *ServiceLaborService* i *ServiceEmployeeService*.

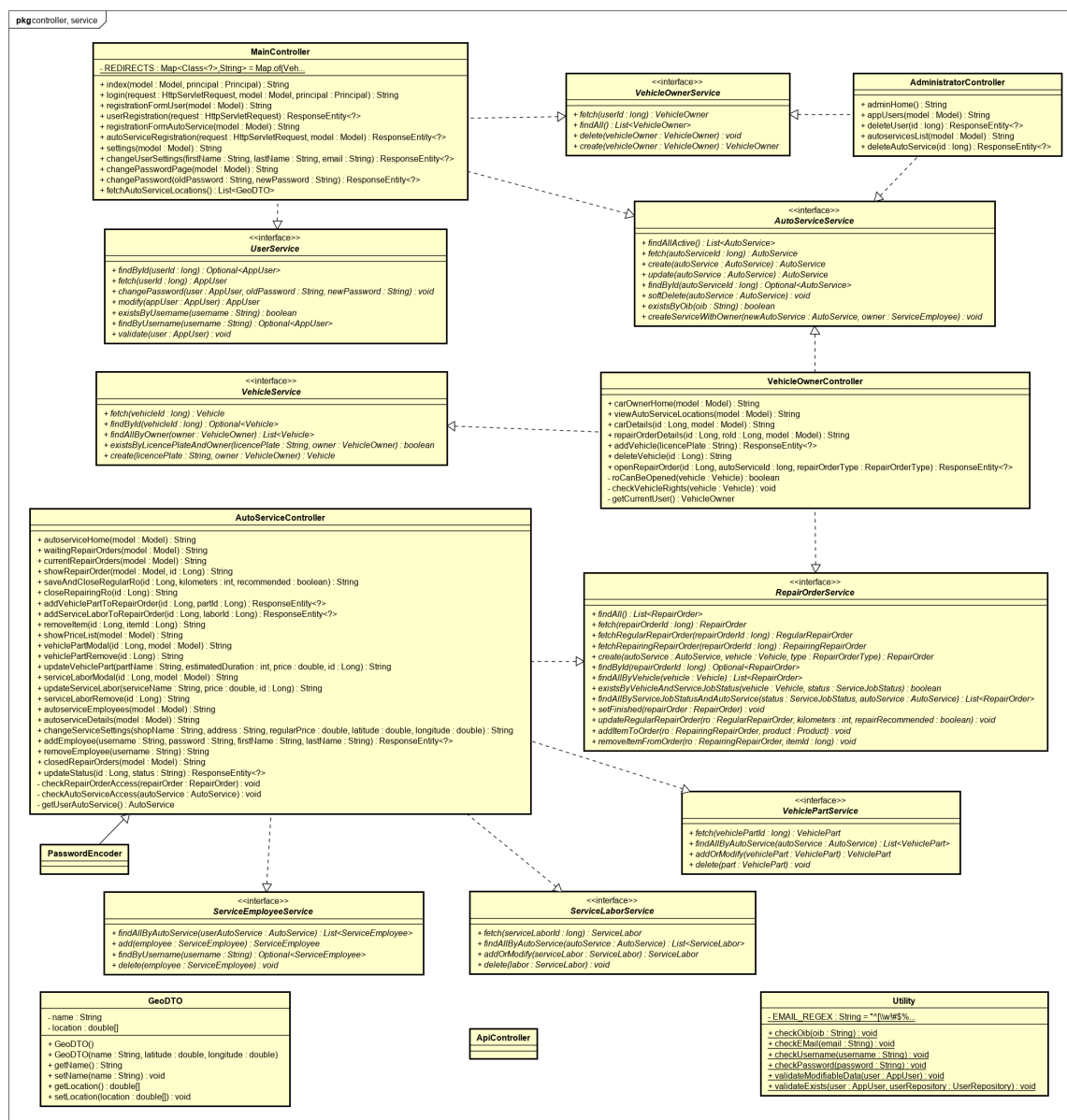
Razred *GeoDTO* služi za dinamički prijenos podataka o lokacijama servisa na *frontend*.

Razred *Utility* nudi statičke metode za provjeru OIB-a, e-maila, korisničkog imena, lozinke te ostalih izmijenjivih podataka i za provjeru postojanja podataka.

Sučelje *UserService* definira operacije za uređivanje računa svih korisnika. Sučelje *VehicleOwnerService* definira operacije za uređivanje računa svih korisnika vlasnika vozila. Sučelje *AutoServiceService* definira operacije povezane s autoservisima. Sučelje *VehicleService* omogućava upravljanje vozilima. Sučelje *RepairOrderService* omogućava upravljanje servisnim nalogima. Sučelje *VehiclePartService* omogućava upravljanje zamjenskim komadima za vozila. Sučelje *ServiceLaborService* omogućava upravljanje uslugama koje se nude. Sučelje *ServiceEmployeeService* omogućava upravljanje radnicima (serviserima).

Sve ta sučelja imaju i konkretne implementacije koje za pristup podacima koriste *JPA* repozitorije, a oni su smješteni u paket *impl*.





Slika 4.5: Dijagram razreda kontrolerskog i sučelja servisnog sloja

Na slici 4.6 prikazan je razred *impl*. Razred *AutoServiceServiceJpa* pruža uslugu vezanu za autoservis – stvaranje, izmjenu, pronalazak, brisanje itd. Također pruža mogućnost stvaranja autoservisa s vlasnikom. Ovaj razred koristi sučelja *UserService*, *ServiceEmployeeRepository*, *ServiceLaborRepository*, *VehiclePartRepository*, *AutoServiceRepository* i *AutoServiceService*.

Razred *UserServiceJpa* pruža uslugu pronalaska, dohvata, ažuriranja podataka, promjenu lozinke korisnika itd. Ovaj razred koristi sučelja *UserRepository* i *UserService*.

Razred *PasswordEncoder* je prazan, te ga nasljeđuju *VehicleOwnerServiceJpa*, *UserServiceJpa* i *AutoServiceServiceJpa*.

Razred *VehicleOwnerServiceJpa* pruža uslugu vezanu za vlasnike automobila: dohvat, pronalazak, brisanje i stvaranje. Ovaj razred koristi sučelja *VehicleOwnerRepository*, *UserRepository*, *VehicleOwnerService* i *VehicleRepository*.

Razred *VehicleServiceJpa* pruža uslugu dohvata, pronalaska, stvaranja i provjere postojanja automobila. Koristi sučelja *HuoConnector*, *VehicleRepository* i *VehicleService*.

Razred *ServiceLaborServiceJpa* pruža uslugu za dohvat, pronalazak, dodavanje, mijenjanje i brisanje usluge autoservisa. Koristi sučelja *ServiceLaborService* i *ServiceLaborRepository*.

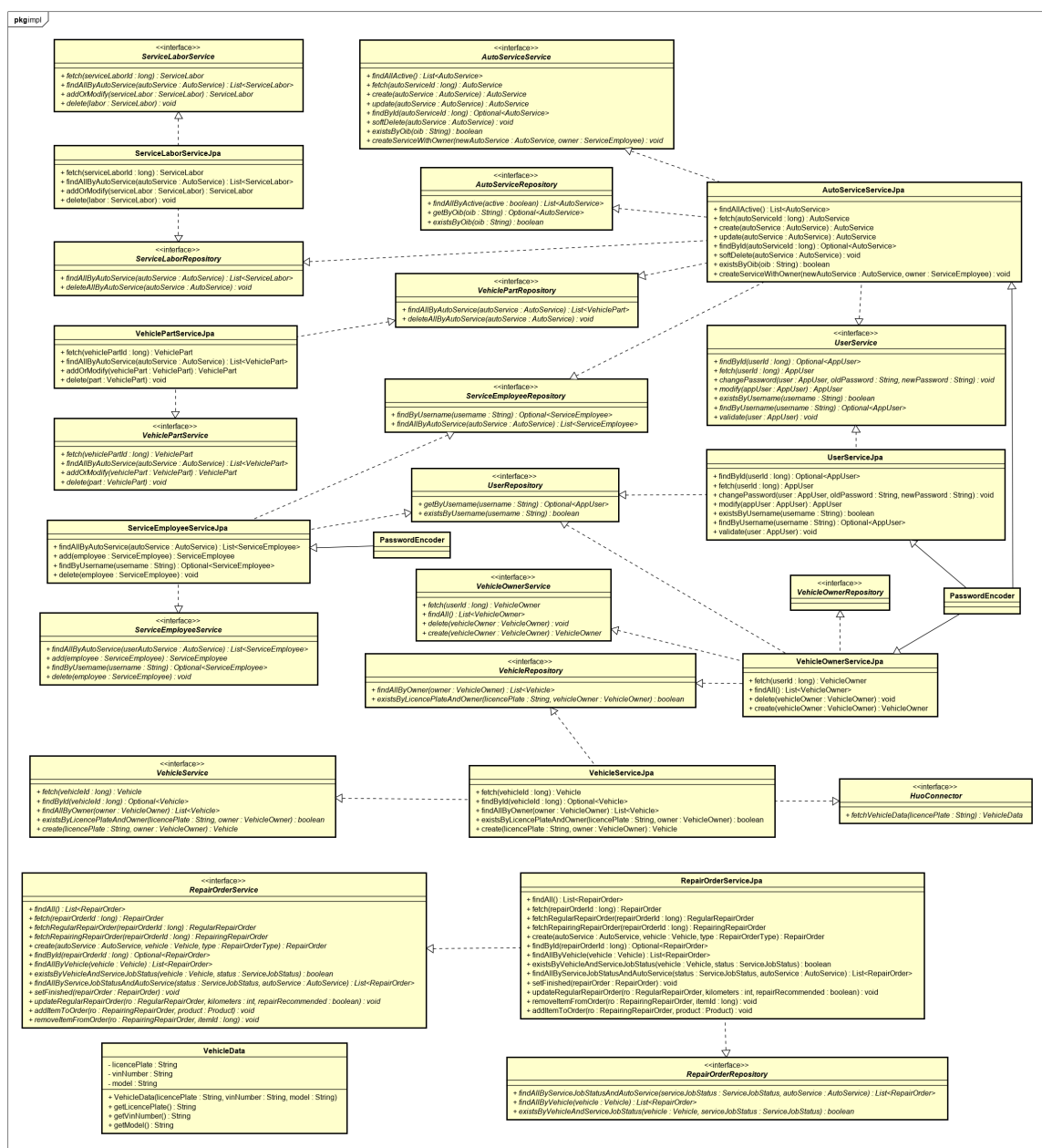
Razred *VehiclePartServiceJpa* pruža uslugu pronalaska, dohvata, dodavanja, mijenjanja i brisanja rezervnog dijela auta. Koristi sučelja *VehiclePartService* i *VehiclePartRepository*.

Razred *ServiceEmployeeServiceJpa* pruža uslugu pronalaska, dodavanja i brisanja zaposlenika određenog autoservisa. Koristi sučelje *ServiceEmployeeService*, *ServiceEmployeeRepository* i *UserRepository*, a nasljeđuje *PasswordEncoder*.

Razred *VehicleServiceJpa* pruža uslugu dohvata, pronalaska, provjere postojanja i stvaranje automobila koristeći sučelja *HuoConnector*, *Vehicle Repository* i *VehicleService*.

Razred *RepairOrderServiceJpa* pruža uslugu dohvata, pronalaska, stvaranja i ažuriranja radnog naloga, kao i dodavanje i micanje stavki. Koristi sučelja *RepairOrderRepository* i *RepairOrderService*.

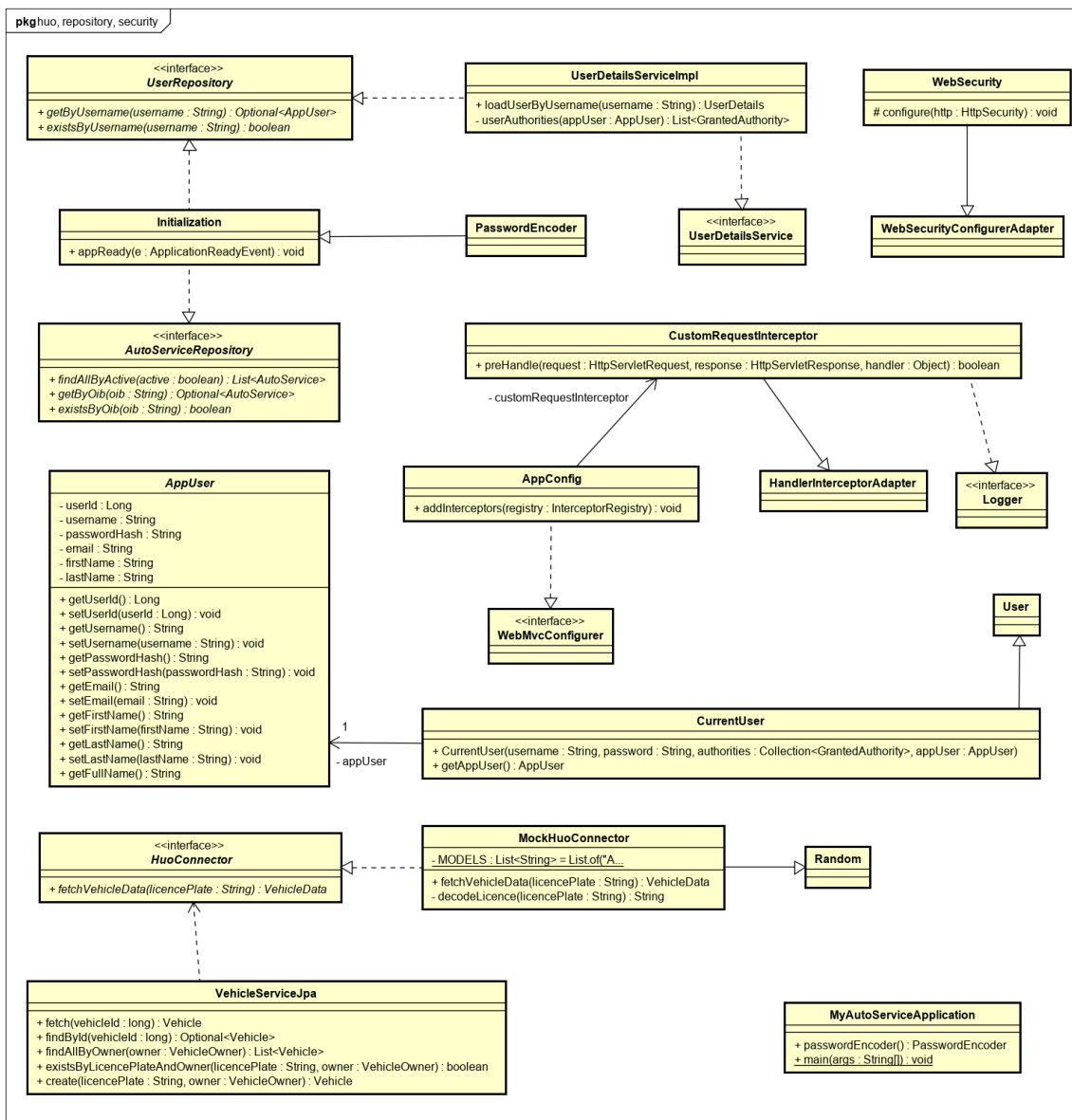
Razred *VehicleData* predstavlja podatke o automobilu: registracijske oznake, broj šasije i model.



Slika 4.6: Dijagram razreda servisnog sloja

Na slici 4.7 dijagramom su prikazani svi preostali razredi, preciznije paketi *huo*, *repository*, i *security*. Razred *MyAutoServiceApplication* predstavlja glavni razred od kojeg počinje izvršavanje poslužiteljske aplikacije. Razred *Initialization* izvršava upis testnih podataka u bazu prilikom pokretanja aplikacije u razvojnom okruženju. Sučelja *UserRepository* i *AutoServiceRepository* pripadaju sloju pristupa podacima. Razredi *UserDetailsServiceImpl* (čije je sučelje *UserDetailsService*) i *WebSecurity* služe za dohvaćanje podataka o korisniku prilikom autentikacije odnosno

za konfiguraciju autentifikacije i autorizacije, i podatke o trenutnom korisniku pohranjuju kao razred *CurrentUser*.



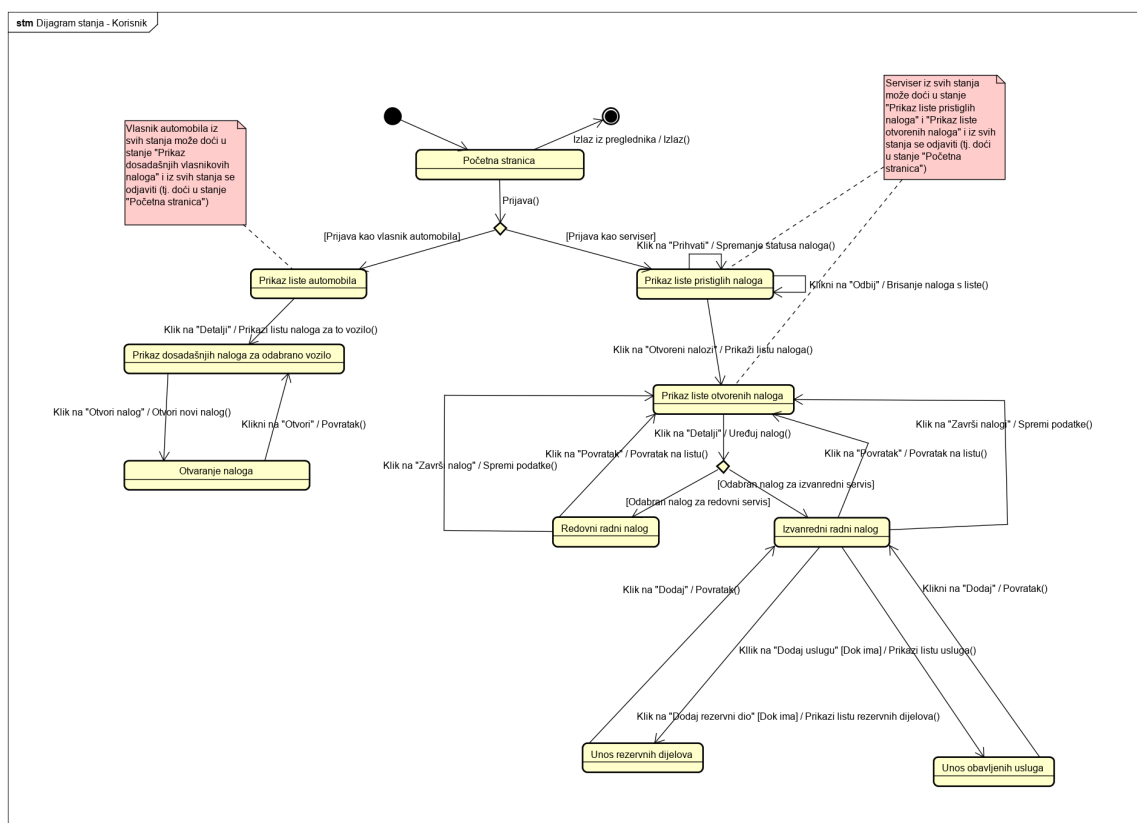
Slika 4.7: Dijagram razreda huo, repository i security

### 4.3 Dijagram stanja

Slika 4.8 prikazuje dijagram stanja korisnika koji pristupa aplikaciji. On se na početnoj stranici može prijaviti kao serviser ili kao vlasnik automobila, ili može izaći iz preglednika.

Ukoliko se prijavi kao vlasnik automobila, prikazuje mu se popis automobila. Klikom na gumb "Detalji" prikazuju mu se radni nalozi za određeni automobil. Klikom na "Otvori nalog" on može otvoriti novi radni nalog, te ga nakon toga aplikacija vraća na popis radnih naloga za taj automobil. Također, vlasnik iz svih stanja može doći na prikaz dosadašnjih naloga, ili se odjaviti (tj. doći na početnu stranicu).

Ako se korisnik prijavi kao serviser, prikazuje mu se lista pristiglih naloga koje on može prihvatiti klikom na gumb Prihvati (tada se nalog premješta u otvorene naloge) ili odbiti klikom na gumb Odbij (tada se nalog briše s liste). Klikom na "Otvoreni nalozi", serviser se preusmjerava na prikaz liste otvorenih naloga. Klikom na "Detalji" serviser može uređivati određeni nalog. Ukoliko je nalog za redovan servis, serviser klikom na "Završi" može završiti radni nalog, a klikom na "Povratak" vratiti se na listu. Ukoliko je nalog pak izvanredan, serviser može dodati novu uslugu klikom na "Dodaj uslugu" (tada mu se prikaže lista usluga) ili dodati novi rezervni dio klikom na "Dodaj rezervni dio" (tada mu se prikaže lista rezervnih dijelova). U oba slučaja, klik na "Dodaj" vraća serviseru na stranicu tog naloga, te kao i kod redovitog radnog naloga, klikom na "Završi" može završiti radni nalog ili se vratiti na listu klikom na "Povratak". Također, serviser iz svih stanja može doći do prikaza liste pristiglih naloga, te se iz svih stanja može odjaviti.



Slika 4.8: Dijagram stanja za korisnika

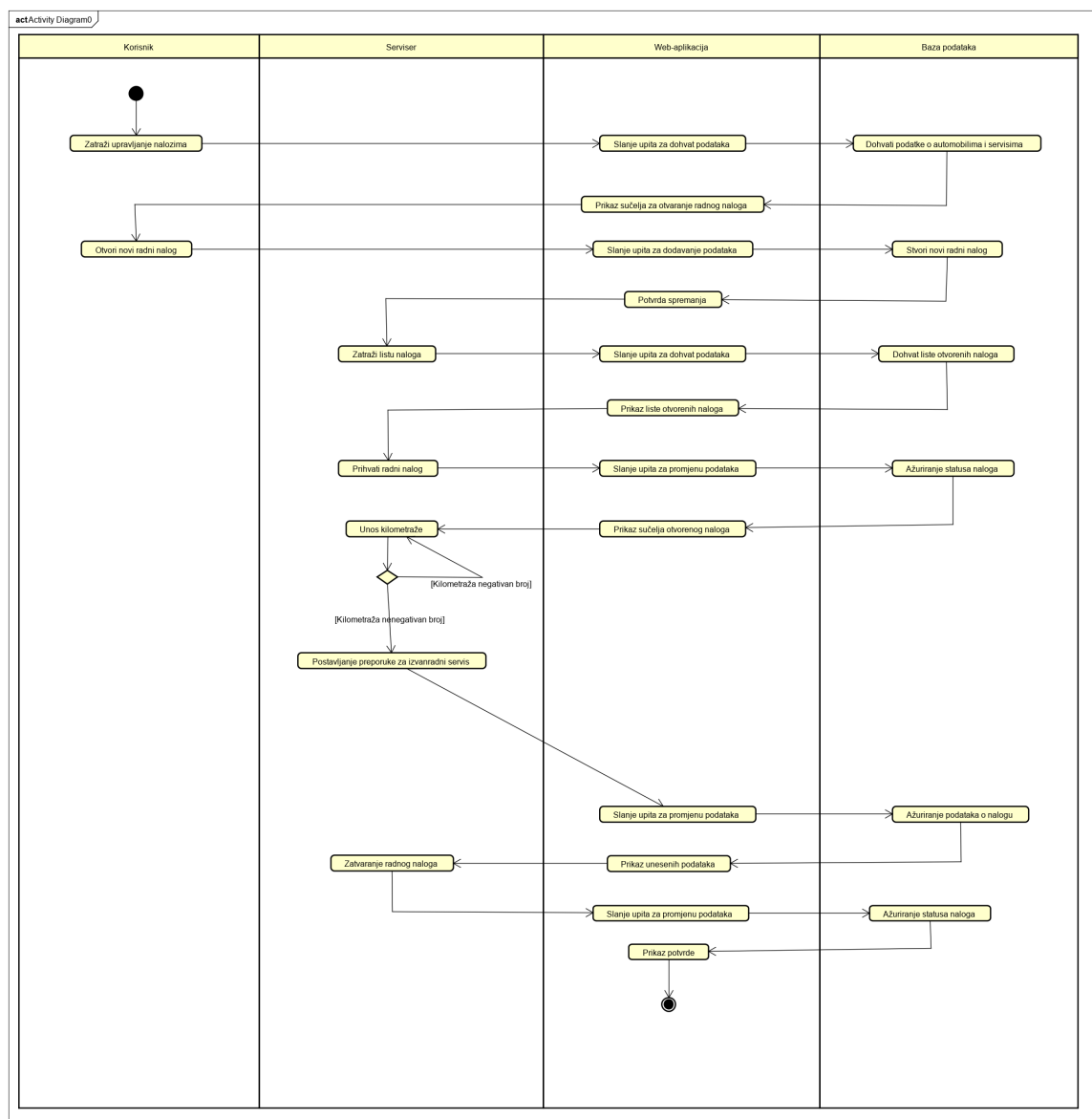
## 4.4 Dijagram aktivnosti

Slika 4.9 prikazuje dijagram aktivnosti za životni tijek jednog radnog naloga, od otvaranja koje vrši korisnik do zatvaranja.

Korisnik zatraži upravljanje nalogima web-aplikaciji, koja zatim šalje upit za dohvat podataka bazi podataka. Nakon što baza vrati podatke o automobilima i servisima, web-aplikacija prikazuje sučelje za otvaranje radnog naloga. Korisnik tada otvara novi radni nalog, a web-aplikacija šalje bazi podataka upit za dodavanjem podataka. Kada baza stvori novi radni nalog, web-aplikacija potvrđuje spremanje.

Nakon toga, serviser zatraži listu naloga te ju web-aplikacija dohvati iz baze. Serviser prihvća radni nalog te ponovno preko web-aplikacije i baze ažurira status naloga. Nakon što web-aplikacija prikaže sučelje otvorenog naloga, serviser unosi kilometražu (postupak se ponavlja dok serviser ne unese ispravan format kilometraže, odnosno nenegativan broj), te odlučuje je li potreban izvanredni servis. Web-aplikacija šalje upit bazi koja ažurira podatke, nakon čega web-aplikacija prikazuje ažurirane podatke serviseru.

Naposljetku, serviser zatvara radni nalog, te web-aplikacija šalje upit za promjenom podataka bazi. Ona ažurira podatke, nakon čega web-aplikacija prikazuje potvrdu serviseru.



Slika 4.9: Dijagram aktivnosti



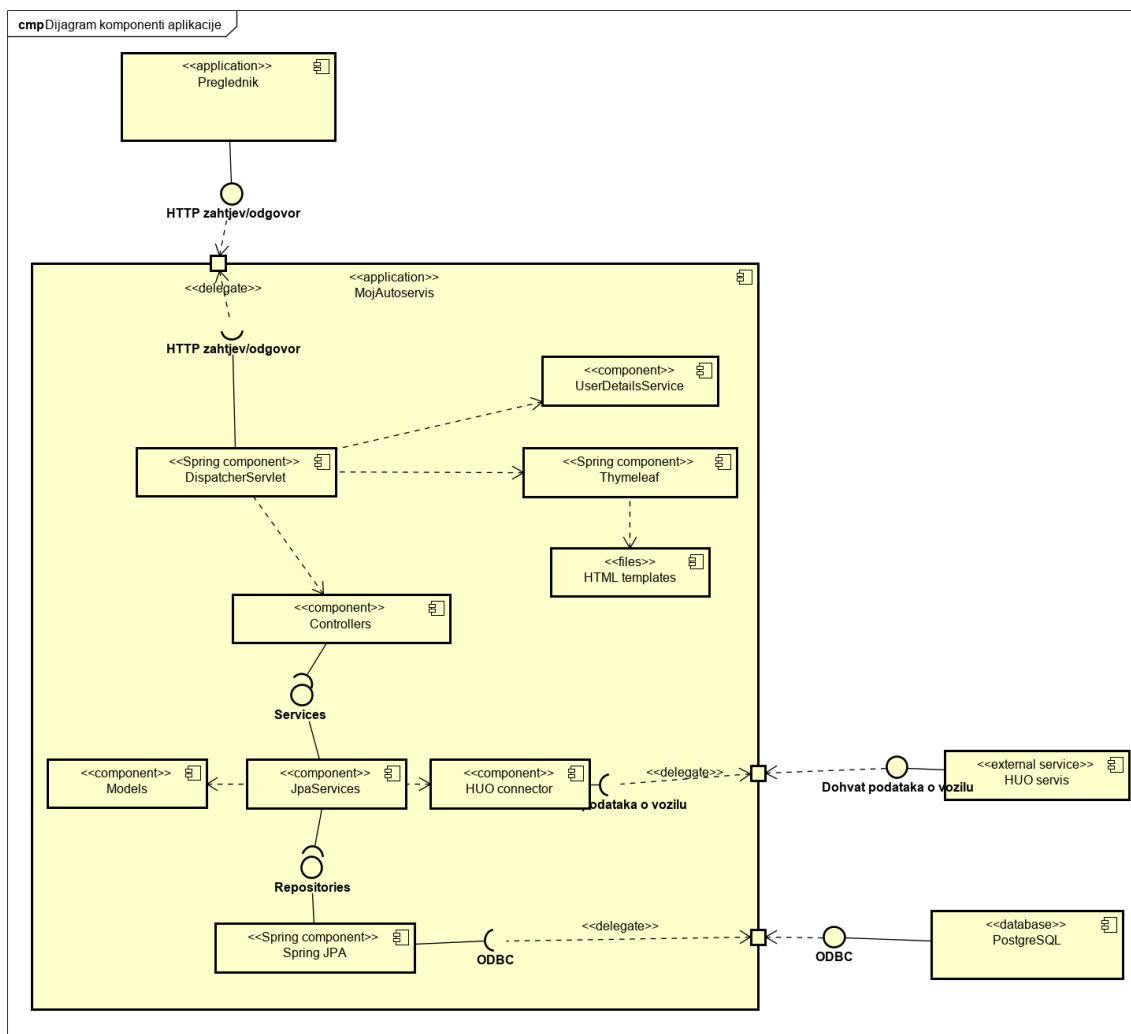
## 4.5 Dijagram komponenti

Na slici 4.10 su je na dijagramu komponenti prikazana interna struktura i odnosi glavnih komponenti aplikacije te njihova međuovisnost s okolinom.

Preglednik putem *HTTP* zahtjeva i odgovora komunicira s aplikacijom. Ti zahtjevi u aplikaciji pristižu u interni Springov *DispatcherServlet* koji pomoću komponente *UserDetailsService* saznaje trenutnog korisnika i njegove ovlasti te prosljeđuje obradu komponenti *Controllers* koja obuhvaća sve komponente u kontrolerskom sloju aplikacije. Pomoću vraćenih podataka i komponente *Thymeleaf* iz predložaka se stvara *HTML* kod koji *DispatcherServlet* šalje natrag klijentu.

Komponenta *Controllers* sve operacije obavlja oslanjajući se na komponentu *JpaServices* koja implementira tražene funkcionalnosti. Ta komponenta dohvat podataka vrši iz repozitorija čije implementacije stvara komponenta *Spring JPA* koja sama piše i parsira *SQL* upite i odgovore te se spaja na bazu podataka.

Postoji i mogućnost dohvata podataka o vozilu iz vanjskog HUO servisa koju interno vrši *HUOConnector*, a koriste je ostali servisi. Funkcionalnost spajanja na vanjski HUO servis trenutno nije implementirana, pa *HUO Connector* samo glumi tu funkcionalnost.



Slika 4.10: Dijagram komponenti

## 5. Implementacija i korisničko sučelje

### 5.1 Korištene tehnologije i alati

Prilikom rada na projektu komunikacija je ostvarena putem aplikacije *Slack*<sup>1</sup>, a za praćenje promjena na izvornom kodu aplikacije i dokumentacije je korišten alat *Git*<sup>2</sup> s udaljenim repozitorijem na servisu *GitLab*<sup>3</sup>. Za pisanje koda je korišteno razvojno okruženje *IntelliJ IDEA*<sup>4</sup>. Dokumentacija pisana u *LaTeX*<sup>5</sup>-u korištenjem aplikacija *TexStudio*<sup>6</sup>, a dijagrami su izrađeni alatom *Astah UML*<sup>7</sup>.

Aplikacija, odnosno njezin *backend* je napisan u programskom jeziku *Java* korištenjem radnog okvira *Spring*<sup>8</sup>. To je okvir koji nudi dio poslužiteljske implementacije web aplikacije slijedeći obrazac model-pogled-nadglednik, koristi princip inverzije ovisnosti koji će olakšati kasnije promjene te nudi gotovu implementaciju različitih protokola. Korištena je i biblioteka *Thymeleaf*<sup>9</sup> koja olakšava generiranje *HTML* koda koji se šalje klijentovom pregledniku.

Uz njega je korištena i biblioteka *Spring JPA*<sup>10</sup> (koja uključuje *ORM* biblioteku *Hibernate*) koja automatski preslikava razrede i njihove veze u relacijski model te generira i provodi SQL upite koje sam obrađuje. U razvojnoj okolini se koristi *H2*<sup>11</sup> baza podataka koja se čuva u memoriji za vrijeme pokretanja aplikacije te omogućava lako testiranje i dodavanje podataka. U produkcijskoj okolini je korištena *PostgreSQL*<sup>12</sup> baza podataka, koja je besplatna i otvorenog koda te dobro podržana u *Spring* okviru.

*Frontend* je ostvaren u *HTML*-u, *CSS*-u i *JavaScript*<sup>13</sup>-u. Korištena je biblioteka

---

<sup>1</sup><https://slack.com/>

<sup>2</sup><https://git-scm.com/>

<sup>3</sup><https://gitlab.com/>

<sup>4</sup><https://www.jetbrains.com/idea/>

<sup>5</sup><https://www.latex-project.org/>

<sup>6</sup><https://www.texstudio.org/>

<sup>7</sup><http://astah.net/>

<sup>8</sup><https://spring.io/>

<sup>9</sup><https://www.thymeleaf.org/>

<sup>10</sup><https://spring.io/projects/spring-data-jpa>

<sup>11</sup><https://www.h2database.com>

<sup>12</sup><https://www.postgresql.org/>

<sup>13</sup><https://www.javascript.com/>

*Bootstrap*<sup>14</sup> koja omogućuje lakšu i bržu izradu responzivnih stranica korištenjem gotovih komponenti. Biblioteka *jQuery*<sup>15</sup> je korištena zbog nadogradnje osnovnih mogućnosti *JavaScripta*.

---

<sup>14</sup><https://getbootstrap.com/>

<sup>15</sup><https://jquery.com/>

## 5.2 Ispitivanje programskog rješenja

### 5.2.1 Ispitivanje komponenti

#### Ispitni slučaj 1: Dodavanje stavke na račun

Izvorni kod:

```
@Test
public void addItemToOrderAddsItemAndUpdatesPrice() {
    repairOrderService.addItemToOrder(repRo, product);

    assertEquals(100.00, (double) repRo.getPrice());
    assertEquals(1, repRo.getItems().size());
    assertEquals("Item", repRo.getItems().get(0).getName());
}
```

Izvođenje ispita: Ispit je prošao.

#### Ispitni slučaj 2: Dodavanje stavke na račun koji je zatvoren

Izvorni kod:

```
@Test
public void addItemToOrderShouldThrowOnClosedRo() {
    repRo.setServiceJobStatus(ServiceJobStatus.FINISHED);

    assertEquals(RuntimeException.class, () ->
        repairOrderService.addItemToOrder(repRo, product));
}
```

Izvođenje ispita: Ispit je prošao.

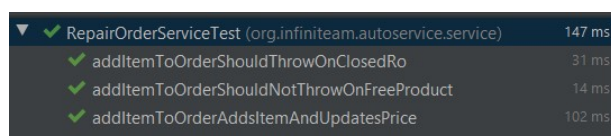
#### Ispitni slučaj 3: Dodavanje besplatne stavke na račun

Izvorni kod:

```
@Test
public void addItemToOrderShouldNotThrowOnFreeProduct() {
    product = new Product() {
        public String getName() {
            return "Free";
        }
    };
}
```

```
    }  
    public double getPrice() {  
        return 0.0;  
    }  
};  
assertDoesNotThrow(() ->  
    repairOrderService.addItemToOrder(repRo, product));  
}
```

Izvođenje ispita: Ispit je prošao.



A screenshot of a test runner interface showing the results of three test cases for the RepairOrderServiceTest class. Each test case is marked with a green checkmark, indicating it passed. The test cases and their execution times are listed in a table.

Test Case	Execution Time
addItemToOrderShouldThrowOnClosedRo	31 ms
addItemToOrderShouldNotThrowOnFreeProduct	14 ms
addItemToOrderAddItemAndUpdatesPrice	102 ms

Slika 5.1: Rezultat ispitnih slučajeva 1, 2 i 3

#### Ispitni slučaj 4: Dodavanje automobila i dohvat podataka o njemu

Izvorni kod:

```
@Test  
public void  
    createVehicleReturnsTheVehicleWithCorrectDataFromHuoServiceAndOwner()  
        throws HuoServiceException {  
    when(huoService.fetchVehicleData(any())).thenReturn(  
        new VehicleData("ZG1234AA", "MOCK-VIN", "Audi A1")  
    );  
    when(vehicleRepository.save(any())).thenReturn(invocation ->  
        invocation.getArgument(0));  
    VehicleOwner owner = new VehicleOwner();  
  
    Vehicle vehicle = vehicleService.create("ZG1234AA", owner);  
  
    assertEquals("ZG1234AA", vehicle.getLicencePlate());  
    assertEquals("MOCK-VIN", vehicle.getVinNumber());  
    assertEquals("Audi A1", vehicle.getVehicleModel());  
    assertEquals(owner, vehicle.getOwner());  
}
```

Izvođenje ispita: Ispit je prošao.

### Ispitni slučaj 5: Dodavanje automobila i dohvat podataka o njemu za automobil s neispravnim podacima

Izvorni kod:

```
@Test
public void createVehicleThrowsWhenHuoServiceThrows() throws
    HuoServiceException {
    when(huoService.fetchVehicleData(any())).thenThrow(new
        HuoServiceException());
    VehicleOwner owner = new VehicleOwner();

    Assertions.assertThrows(RuntimeException.class, () ->
        vehicleService.create("ZG1234AA", owner));
}
```

Izvođenje ispita: Ispit je prošao.

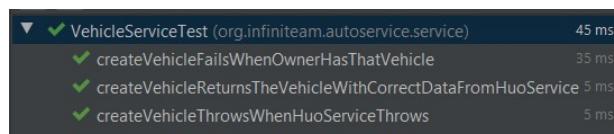
### Ispitni slučaj 6: Dodavanje automobila i dohvat podataka o njemu za automobil koji je već pridijeljen tom vlasniku

Izvorni kod:

```
@Test
public void createVehicleFailsWhenOwnerHasThatVehicle() throws
    HuoServiceException {
    VehicleData data = new VehicleData("ZG1234AA", "MOCK-VIN",
        "Audi A1");
    //when(huoService.fetchVehicleData(any())).thenReturn(data);
    VehicleOwner owner = new VehicleOwner();
    Vehicle vehicle = new Vehicle(data, owner);
    owner.setVehicles(List.of(vehicle));
    when(vehicleRepository.existsByLicencePlateAndOwner(any(),
        any())).thenReturn(true);
    //when(vehicleRepository.findAllByOwner(any())).thenReturn(List.of(vehicle));

    Assertions.assertThrows(AlreadyExistsException.class, () ->
        vehicleService.create("ZG1234AA", owner));
}
```

Izvođenje ispita: Ispit je prošao.

A screenshot of a test runner interface showing the results of a test suite. The suite is 'VehicleServiceTest (org.infiniteam.autoservice.service)' and it passed. It contains four individual tests, all of which also passed. The execution times for each test are listed in milliseconds.

VehicleServiceTest (org.infiniteam.autoservice.service)	45 ms
createVehicleFailsWhenOwnerHasThatVehicle	35 ms
createVehicleReturnsTheVehicleWithCorrectDataFromHuoService	5 ms
createVehicleThrowsWhenHuoServiceThrows	5 ms

Slika 5.2: Rezultat ispitnih slučajeva 4, 5 i 6

### 5.2.2 Ispitivanje sustava

Svi testovi izvršeni su uz pomoć alata Selenium. Svrha ispitivanja je provjera osnovnih funkcionalnosti sustava te pronalazak mogućih pogrešaka ili neočekivanih ponašanja u izvođenju koda.

#### Ispitni slučaj 1: Korisnik dodaje automobil s ispravnom registarskom oznakom

##### Ulaz:

1. Otvaranje početne web stranice u pregledniku
2. Klik na gumb za otvaranje stranice za prijavu *Prijava*
3. Unos korisničkog imena korisnika vlasnika vozila i lozinke u polja za unos
4. Klik na gumb *Prijava*
5. Klik na gumb *Novi automobil*
6. Unos registracijske tablice 'ZG1234AB' u polje unosa
7. Klik na gumb *Dodaj*
8. Zatvaranje prozora preglednika

##### Očekivani rezultat:

1. Prikazuje se početna web stranica
2. Prikazuje se stranica za prijavu
3. U polja za unos unešeni su korisničko ime korisnika vlasnika vozila i lozinka
4. Prikazuje se početna web stranica prijavljenog korisnika
5. Pojavljuje se prozor za dodavanje vozila
6. U polje za unos unešena je registracijska tablica 'ZG1234AB'
7. Na stranici postoji kartica ili neki tekst s navedenom registarskom oznakom
8. Prozor preglednika zatvoren



**Rezultat:** Sva očekivanja su zadovoljena. Selenium test je prošao.

## **Ispitni slučaj 2: Korisnik dodaje automobil s registarskom oznakom koja već postoji**

### **Ulaz:**

1. Otvaranje početne web stranice u pregledniku
2. Klik na gumb za otvaranje stranice za prijavu *Prijava*
3. Unos korisničkog imena korisnika vlasnika vozila i lozinke u polja za unos
4. Klik na gumb *Prijava*
5. Klik na gumb *Novi automobil*
6. Unos registracijske tablice 'ZG1234AB' u polje unosa
7. Klik na gumb *Dodaj*
8. Klik na gumb *Novi automobil*
9. Unos registracijske tablice 'ZG1234AB' u polje unosa
10. Klik na gumb *Dodaj*
11. Zatvaranje prozora preglednika

### **Očekivani rezultat:**

1. Prikazuje se početna web stranica
2. Prikazuje se stranica za prijavu
3. U polja za unos unešeni su korisničko ime korisnika vlasnika vozila i lozinka
4. Prikazuje se početna web stranica prijavljenog korisnika
5. Pojavljuje se prozor za dodavanje vozila
6. U polje za unos unešena je registracijska tablica 'ZG1234AB'
7. Zatvara se prozor za dodavanje vozila i na stranici postoji kartica ili neki tekst s navedenom registarskom oznakom
8. Pojavljuje se prozor za dodavanje vozila
9. U polje za unos unešena je registracijska tablica 'ZG1234AB'
10. Pojavljuje se poruka pogreške: *Vozilo već postoji!*
11. Prozor preglednika zatvoren

**Rezultat:** Sva očekivanja su zadovoljena. Selenium test je prošao.

## **Ispitni slučaj 3: Administrator servisa dodaje novu uslugu**

**Ulaz:**

1. Otvaranje početne web stranice u pregledniku
2. Klik na gumb za otvaranje stranice za prijavu *Prijava*
3. Unos korisničkog imena administratora i lozinke u polja za unos
4. Klik na gumb *Prijava*
5. Brisanje svih dijelova i usluga
6. Klik na gumb *Dodaj* za novi rezervni dio
7. Unos 'Guma' u polje *Naziv*, '200' u polje *Cijena u kn* i '100' u polje *Predviđeno trajanje u km*
8. Klik na gumb *Spremi*
9. Zatvaranje prozora preglednika

**Očekivani rezultat:**

1. Prikazuje se početna web stranica
2. Prikazuje se stranica za prijavu
3. U polja za unos unešeni su korisničko ime administratora i lozinka
4. Prikazuje se početna web stranica prijavljenog administratora
5. Nema ni jednog dijela ni usluge
6. Otvara se prozor za unos rezervnog dijela
7. U polje *Naziv* unesen je naziv 'Guma', u polje *Cijena u kn* unesena je cijena '200', a u polje *Predviđeno trajanje u km* uneseno je '100'
8. Zatvara se prozor unosa novog rezervnog dijela te se rezervni dio i njegovi detalji pojavljuju na popisu rezervnih dijelova
9. Prozor preglednika zatvoren

**Rezultat:** Sva očekivanja su zadovoljena. Selenium test je prošao.

**Ispitni slučaj 4: Administrator servisa registrira novog serviser a serviser se prijavljuje****Ulaz:**

1. Otvaranje početne web stranice u pregledniku
2. Klik na gumb za otvaranje stranice za prijavu *Prijava*
3. Unos korisničkog imena korisnika vlasnika vozila i lozinke u polja za unos

4. Klik na gumb *Prijava*
5. Klik na gumb *Zaposlenici*
6. Brisanje svih ostalih zaposlenika
7. Klik na gumb *Dodaj zaposlenika*
8. Unos 'user2' u polja *Korisničko ime* i *Lozinka*, 'Darian' u polje *Ime* i 'Horvat' u polje *Prezime*
9. Klik na gumb *Spremi*
10. Klik na gumb za dropdown opcija
11. Klik na gumb *Odjava*
12. Klik na gumb *Prijava*
13. Unos 'user2' u polje *Korisničko ime* i *Lozinka*
14. Klik na gumb *Prijava*
15. Zatvaranje prozora preglednika

**Očekivani rezultat:**

1. Prikazuje se početna web stranica
2. Prikazuje se stranica za prijavu
3. U polja za unos unešeni su korisničko ime administratora i lozinka
4. Prikazuje se početna web stranica prijavljenog administratora
5. Prikazuje se stranica sa popisom zaposlenika
6. Svi ostali zaposlenici izbrisani
7. Otvara se prozor za dodavanje novog zaposlenika
8. U polja *Korisničko ime* i *Lozinka* unešeno je 'user2', u polje *Ime* 'Darian' te u polje *Prezime* 'Horvat'
9. Zatvara se prozor za dodavanje novog zaposlenika i novi se zaposlenik nalazi na popisu zaposlenika
10. Prikazuju se opcije u dropdownu
11. Korisnik je odjavljen i prikazuje se početna web stranica
12. Prikazuje se stranica za prijavu
13. U polja *Korisničko ime* i *Lozinka* unešeno je 'user2'
14. Korisnik je uspješno prijavljen i prikazuje se početna stranica prijavljenog korisnika
15. Prozor preglednika zatvoren

**Rezultat:** Sva očekivanja su zadovoljena. Selenium test je prošao.

**Ispitni slučaj 5: Administrator servisa registrira novog servisera s već postojećim korisničkim imenom****Ulaz:**

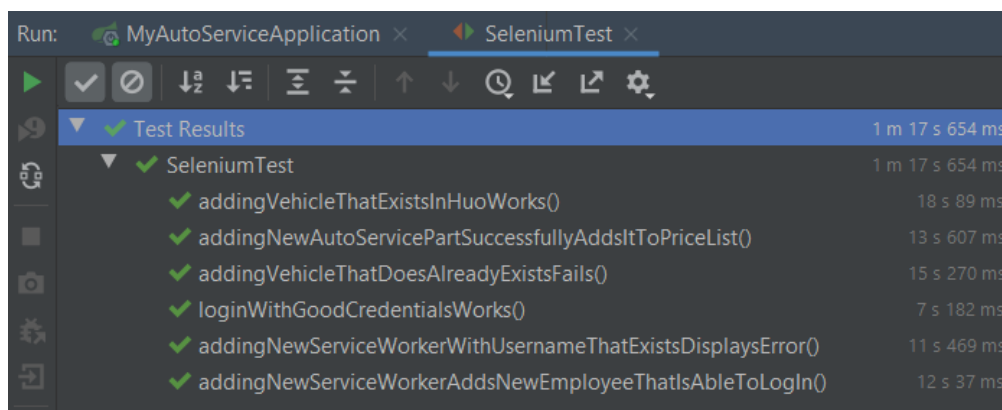
1. Otvaranje početne web stranice u pregledniku
2. Klik na gumb za otvaranje stranice za prijavu *Prijava*
3. Unos korisničkog imena korisnika vlasnika vozila i lozinke u polja za unos
4. Klik na gumb *Prijava*
5. Klik na gumb *Zaposlenici*
6. Brisanje svih ostalih zaposlenika
7. Klik na gumb *Dodaj zaposlenika*
8. Unos 'user2' u polja *Korisničko ime* i *Lozinka*, 'Darian' u polje *Ime* i 'Horvat' u polje *Prezime*
9. Klik na gumb *Spremi*
10. Klik na gumb *Dodaj zaposlenika*
11. Unos 'user2' u polja *Korisničko ime* i *Lozinka*, 'Neki' u polje *Ime* i 'Novi' u polje *Prezime*
12. Klik na gumb *Spremi*
13. Zatvaranje prozora preglednika

**Očekivani rezultat:**

1. Prikazuje se početna web stranica
2. Prikazuje se stranica za prijavu
3. U polja za unos unešeni su korisničko ime administratora i lozinka
4. Prikazuje se početna web stranica ulogiranog administratora
5. Prikazuje se stranica sa popisom zaposlenika
6. Svi ostali zaposlenici izbrisani
7. Otvara se prozor za dodavanje novog zaposlenika
8. U polja *Korisničko ime* i *Lozinka* unešeno je 'user2', u polje *Ime* 'Darian' te u polje *Prezime* 'Horvat'
9. Zatvara se prozor za dodavanje novog zaposlenika i novi se zaposlenik nalazi na popisu zaposlenika
10. Otvara se prozor za dodavanje novog zaposlenika

11. U polja *Korisničko ime* i *Lozinka* unešeno je 'user2', u polje *Ime* 'Neki' te u polje *Prezime* 'Novi'
12. Poljavljuje se poruka pogreške: *Korisničko ime je zauzeto*
13. Prozor preglednika zatvoren

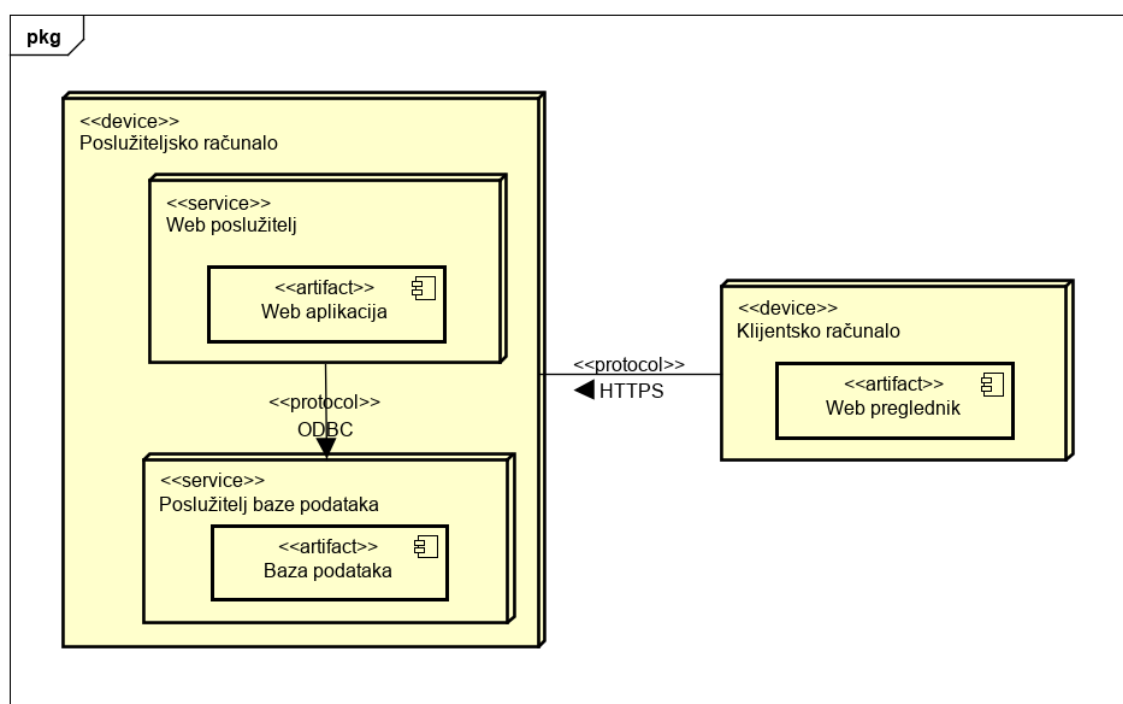
**Rezultat:** Sva očekivanja su zadovoljena. Selenium test je prošao.



Slika 5.3: Rezultat svih Selenium ispitnih slučajeva

## 5.3 Dijagram razmještaja

Cijeli sustav je organiziran na temelju klijent-poslužitelj arhitekture. Na dijagramu se nalazi klijent (klijentsko računalo) s web preglednikom koji putem *HTTPS* protokola razmjenjuje podatke s web poslužiteljem na poslužiteljskom računalu. Na poslužiteljskom računalu se nalaze web poslužitelj u kojem je pokrenuta aplikacija koji *ODBC* protokolom s poslužiteljem baze podataka dohvaća i mijenja podatke u bazi podataka aplikacije.



Slika 5.4: Dijagram razmještaja

## 5.4 Upute za puštanje u pogon

U ovom poljaviću će biti opisana instalacija potrebnih softverskih paketa, stvaranje izvršnog koda, konfiguracija i pokretanje web aplikacije na *Debian* baziranom *Linux* poslužitelju. Ova web aplikacije je prilagođena i testirana za rad s *PostgreSQL* bazom podataka u produkcijskom okruženju te se zbog toga preporuča njezina uporaba iako je aplikaciju moguće konfigurirati i za korištenje ostalih baza podataka. Moguće je koristiti već unaprijed stvorenu bazu podataka, a u nastavku su dane upute za instalaciju *PostgreSQL* baze u slučaju da se ručno želi postaviti baza.

### 5.4.1 Instalacija i konfiguracija baze podataka

*PostgreSQL* baza može se instalirati koristeći *apt* sustava izvođenjem sljedećih naredbi:

```
$ sudo apt update
$ sudo apt install postgresql postgresql-contrib
```

Time je instalirana baza podataka te je potrebno stvoriti novog korisnika i njemu pripadajuću bazu podataka koju će aplikacija koristiti:

```
$ su - postgres
$ createuser korisnik
$ createdb naziv_baze_podataka
$ psql
$ ALTER USER korisnik WITH ENCRYPTED PASSWORD 'lozinka';
$ GRANT ALL PRIVILEGES ON DATABASE naziv_baze_podataka TO korisnik;
```

### 5.4.2 Konfiguracija i pokretanje poslužitelja

Za pokretanje konfiguriranje, prevođenje i pokretanje web aplikacije potrebno je putem *apt* sustava instalirati pakete:

```
$ sudo apt update
$ sudo apt install openjdk-11-jdk-headless maven screen
```

Nakon preuzimanja izvornog koda na poslužitelj (npr. putem *Git*-a ili preko sustava za automatizirani razmjestaj) potrebno je otići u direktorij izvornog koda te nakon otvaranja konfiguracijske datoteke naredbom:

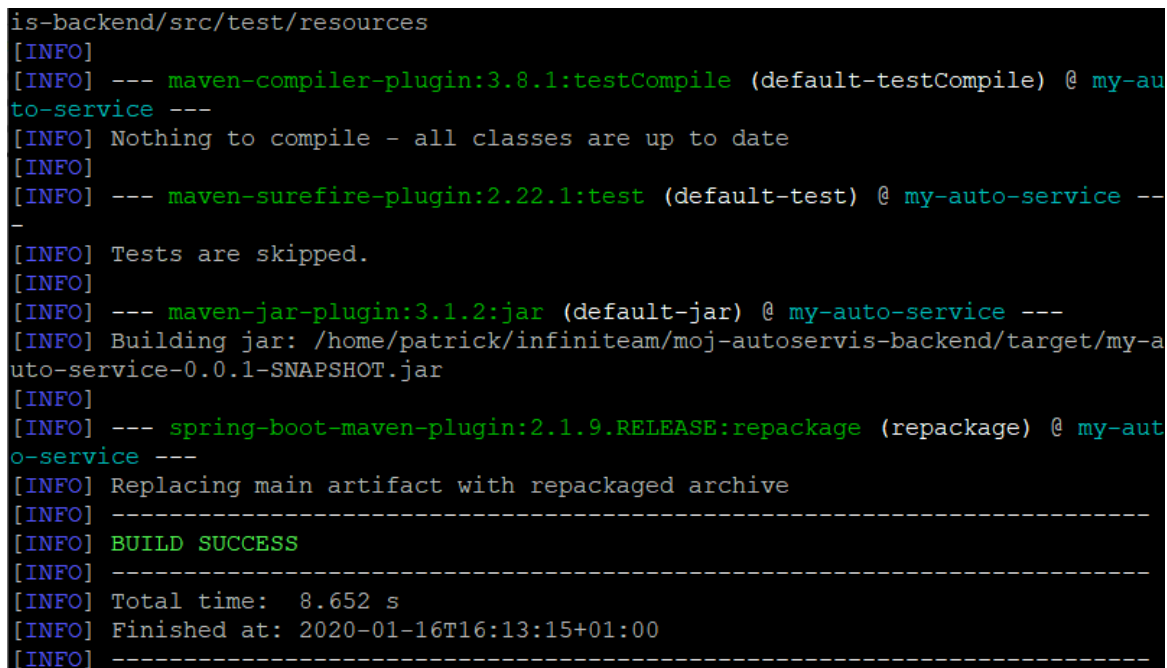
```
$ nano src/main/resources/application.properties
```

postaviti parametre pristupa prema bazi podataka (novostvorenoj u prethodnom koraku ili nekoj već kreiranoj):

```
spring.datasource.url=jdbc:postgresql://localhost:5432/naziv_baze_podataka
spring.datasource.username=korisnik
spring.datasource.password=lozinka
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.PostgreSQLDialect
```

U slučaju korištenja neke druge baze podataka potrebno je na promijeniti *spring.datasource.url* ovisno o uputama te baze podataka. Nakon toga je potrebno stvoriti izvršni kod izvršavanjem naredbe:

```
$ mvn compile
$ mvn build
$ mvn package
```



```
is-backend/src/test/resources
[INFO] --- maven-compiler-plugin:3.8.1:testCompile (default-testCompile) @ my-auto-service ---
[INFO] Nothing to compile - all classes are up to date
[INFO] --- maven-surefire-plugin:2.22.1:test (default-test) @ my-auto-service ---
[INFO] Tests are skipped.
[INFO] --- maven-jar-plugin:3.1.2:jar (default-jar) @ my-auto-service ---
[INFO] Building jar: /home/patrick/infiniteam/moj-autoservis-backend/target/my-auto-service-0.0.1-SNAPSHOT.jar
[INFO] --- spring-boot-maven-plugin:2.1.9.RELEASE:repackage (repackage) @ my-auto-service ---
[INFO] Replacing main artifact with repackaged archive
[INFO] BUILD SUCCESS
[INFO] Total time: 8.652 s
[INFO] Finished at: 2020-01-16T16:13:15+01:00
```

Slika 5.5: Prikaz nakon uspješnog stvaranja izvršnog koda

Preporuča se naredbom

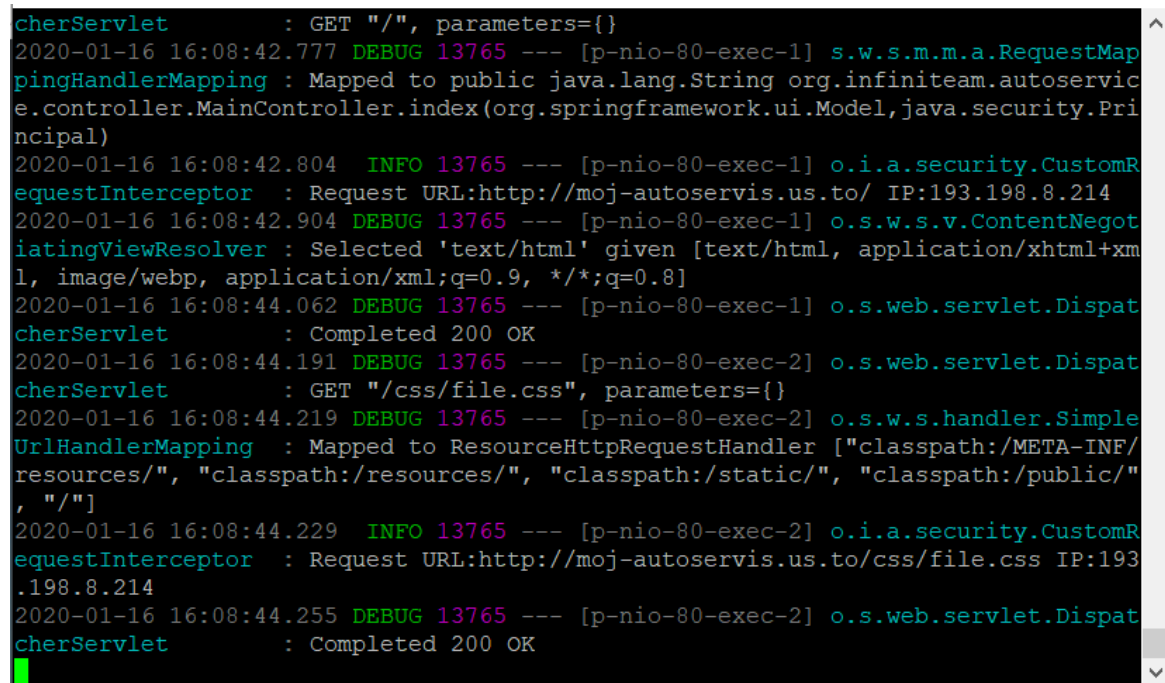
```
$ screen
```

stvoriti virtualni terminal te u njemu pokrenuti aplikaciju izvođenjem naredbe:



```
$ ./run.sh
```

Ta će skripta pokrenuti Springov ugrađeni *Tomcat* poslužitelj na portu standardnom portu 80.



```
cherServlet      : GET "/", parameters={}
2020-01-16 16:08:42.777 DEBUG 13765 --- [p-nio-80-exec-1] s.w.s.m.a.RequestMappingHandlerMapping : Mapped to public java.lang.String org.infiniteam.autoservice.controller.MainController.index(org.springframework.ui.Model,java.security.Principal)
2020-01-16 16:08:42.804 INFO 13765 --- [p-nio-80-exec-1] o.i.a.security.CustomRequestInterceptor : Request URL:http://moj-autoservis.us.to/ IP:193.198.8.214
2020-01-16 16:08:42.904 DEBUG 13765 --- [p-nio-80-exec-1] o.s.w.s.v.ContentNegotiatingViewResolver : Selected 'text/html' given [text/html, application/xhtml+xml, image/webp, application/xml;q=0.9, */*;q=0.8]
2020-01-16 16:08:44.062 DEBUG 13765 --- [p-nio-80-exec-1] o.s.web.servlet.DispatcherServlet : Completed 200 OK
2020-01-16 16:08:44.191 DEBUG 13765 --- [p-nio-80-exec-2] o.s.web.servlet.DispatcherServlet : GET "/css/file.css", parameters={}
2020-01-16 16:08:44.219 DEBUG 13765 --- [p-nio-80-exec-2] o.s.w.s.handler.SimpleUrlHandlerMapping : Mapped to ResourceHttpRequestHandler ["classpath:/META-INF/resources/", "classpath:/resources/", "classpath:/static/", "classpath:/public/", "/"]
2020-01-16 16:08:44.229 INFO 13765 --- [p-nio-80-exec-2] o.i.a.security.CustomRequestInterceptor : Request URL:http://moj-autoservis.us.to/css/file.css IP:193.198.8.214
2020-01-16 16:08:44.255 DEBUG 13765 --- [p-nio-80-exec-2] o.s.web.servlet.DispatcherServlet : Completed 200 OK
```

Slika 5.6: Prikaz nakon pokretanja poslužitelja

## 6. Zaključak i budući rad

Zadatak koji je naša grupa dobila sastojao se od izrade programske potpore za web-aplikaciju *Moj AutoServis*. Ona bi vlasnicima automobila služila kao alat za praćenje stanja svojih vozila i izradu radnih naloga, a autoservisima omogućavala prihvrat i obradu radnih naloga za vozila svojih klijenata, kao i upravljanje podacima autoservisa. Na projektu smo radili nešto manje od četiri mjeseca, tijekom kojih smo se s obzirom na ciljeve projekta fokusirali više na dokumentaciju (prvi dio) ili na sam kod (drugi dio).

Prvi dio izrade projekta sastojao se većinom od strukturiranja kako će sama aplikacija izgledati i prenošenja dogovorenih rješenja u dokumentaciju. Složili smo popis funkcionalnih zahtjeva, opisali sve potrebne obrasce uporabe te po njima izradili *use-case* dijagrame, koji će kasnije biti ključni u izradi same aplikacije. Sekvencijski dijagrami pomogli su nam u shvaćanju procesa koji se moraju odvititi u aplikaciji, a relacijski dijagram baze podataka te dijagram razreda pojasnio je *backend*. Sklonost detaljima i vrijeme utrošeno na izradu ovih dijagrama uvelike nam je pomoglo kasnije, pogotovo u koordinaciji raznih dijelova aplikacije te izradi dizajna.

Sa jasno napisanim ciljevima krenuli smo na samu izradu aplikacije. Iako su u timu postojale povećane razlike u programerskim vještinama, one su suradnjom i timskim radom smanjene, te je svaki član tima pridonio aplikaciji svojim sposobnostima. Članovi su stekli znanje o jezicima poput HTML-a, CSS-a i JavaScript-a, izradi testova u okviru Selenium, te naravno o strukturiranju aplikacije koja prati MVC-obrazac. U ovom dijelu izrade projekta dokumentirali smo i dijagrame stanja, aktivnosti, komponenti te razmještaja. Naposljetku smo dodali i upute za korištenje.

Sva komunikacija tima odvijala se preko Slack-a, koji je omogućio jasno odvajanje dretvi razgovora za određen dio aplikacije te lako slanje koda ili slika, kao i obavješćivanje o napravljenim promjenama u aplikaciji na GitLab-u.

Iako smo zadovoljni napravljenim, svaki rad uvijek može biti bolji. Neke od mogućih nadogradnji uključuju:

1. Prilagodbu aplikacije za korištenje na mobilnim uređajima,

2. Omogućavanje pregleda podataka o servisima automobila dok nije bio u vlasništvu trenutnog vlasnika,
3. Omogućavanje ocjenjivanja kvalitete usluge nakon obavljenog servisa, i
4. Elektroničko plaćanje usluge kroz aplikaciju.

Iako ponekad naporan i frustrirajuć – bilo zbog vlastitog neznanja i neiskustva, bilo zbog nekoordinacije članova – ovaj je projekt bio vrijedno iskustvo svim članovima *InfiniTeam*-a. Kroz sve svoje izazove pružio nam je nešto novog iskustva, vještina, i malo bolju spremnost za sve nadolazeće projekte.

# Popis literature

1. Oblikovanje programske potpore, FER ZEMRIS, <http://www.fer.hr/predmet/opp>
2. I. Sommerville, "Software engineering", 8th ed, Addison Wesley, 2007.
3. T.C.Lethbridge, R.Langaniere, "Object-Oriented Software Engineering", 2nd ed. McGraw-Hill, 2005.
4. I. Marsic, Software engineering book", Department of Electrical and Computer Engineering, Rutgers University, <http://www.ece.rutgers.edu/~marsic/books/SE>
5. The Unified Modeling Language, <https://www.uml-diagrams.org/>
6. Astah Community, <http://astah.net/editions/uml-new>

# Indeks slika i dijagrama

2.1	GetAFix . . . . .	7
2.2	ShopBoss . . . . .	8
3.1	Dijagram obrasca uporabe, funkcionalnost posjetitelja, registriranog korisnika i vlasnika automobila . . . . .	26
3.2	Dijagram obrasca uporabe, funkcionalnost servisera i vlasnika auto-servisa . . . . .	27
3.3	Dijagram obrasca uporabe, funkcionalnost administratora . . . . .	28
3.4	Sekvencijski dijagram za UC1 . . . . .	29
3.5	Sekvencijski dijagram za promjenu korisničkih podataka, UC7 . . . . .	30
3.6	Sekvencijski dijagram za tijek redovnog servisa, UC12–UC15 . . . . .	32
3.7	Sekvencijski dijagrami tijekom izvanrednog servisa . . . . .	34
4.1	Osnovna arhitektura cijelog sustava . . . . .	36
4.2	Oblikovni obrazac Model-pogled-nadglednik . . . . .	37
4.3	Relacijski model baze podataka . . . . .	44
4.4	Dijagram razreda modela . . . . .	46
4.5	Dijagram razreda kontrolerskog i sučelja servisnog sloja . . . . .	48
4.6	Dijagram razreda servisnog sloja . . . . .	50
4.7	Dijagram razreda huo, repository i security . . . . .	51
4.8	Dijagram stanja za korisnika . . . . .	53
4.9	Dijagram aktivnosti . . . . .	55
4.10	Dijagram komponenti . . . . .	57
5.1	Rezultat ispitnih slučajeva 1, 2 i 3 . . . . .	61
5.2	Rezultat ispitnih slučajeva 4, 5 i 6 . . . . .	63
5.3	Rezultat svih Selenium ispitnih slučajeva . . . . .	68
5.4	Dijagram razmještaja . . . . .	69
5.5	Prikaz nakon uspješnog stvaranja izvršnog koda . . . . .	71
5.6	Prikaz nakon pokretanja poslužitelja . . . . .	72

# Dodatak: Prikaz aktivnosti grupe

## Dnevnik sastajanja

### 1. sastanak

- Datum: 09.10.2019.
- Prisustvovali: P.Matošević, K.Boras, D.V.Cvitković, D.Facković, M.Has, N.Kolarec, J.Prpić
- Teme sastanka:
  - sastanak s asistentom
  - analiza zadanog zadatka i mogućih alata i tehnologija
  - rasprava potencijalnih ostalih tema

### 2. sastanak

- Datum: 17.10.2019.
- Prisustvovali: P.Matošević, K.Boras, D.V.Cvitković, D.Facković, N.Kolarec, J.Prpić
- Teme sastanka:
  - postavljanje okoline potrebne za nastavak rada na projektu
  - upoznavanje sa Gitom i kontrolerima u Spring Boot-u
  - rasprava oko organizacije posla
  - diskusija o iskustvu članova tima u pojedinim tehnologijama

### 3. sastanak

- Datum: 22.10.2019.
- Prisustvovali: P.Matošević, K.Boras, D.V.Cvitković, D.Facković, M.Has, N.Kolarec, J.Prpić
- Teme sastanka:
  - detaljna analiza zadatka
  - pregled sličnih aplikacija
  - razrada funkcionalnosti u aplikaciji po aktorima
  - podjela posla pisanja uvoda, funkcionalnosti i dijagrama

### 4. sastanak

- Datum: 24.10.2019.
- Prisustvovali: P.Matošević, D.Facković, N.Kolarec
- Teme sastanka:
  - sastanak s asistentom - razrješavanje dilema oko procesa servisiranja automobila i komentiranje dosadašnjeg rada

## 5. sastanak

- Datum: 07.11.2019.
- Prisustvovali: P.Matošević, K.Boras, D.Facković, N.Kolarec, D.V.Cvitković
- Teme sastanka:
  - sastanak s asistentom - pregled dosadašnjeg rada
  - provjera sekvencijskih dijagrama
  - diskusija oko razreda i entiteta u bazi podataka

## 6. sastanak

- Datum: 14.11.2019.
- Prisustvovali: P.Matošević, K.Boras, D.V.Cvitković, D.Facković, M.Has, N.Kolarec, J.Prpić
- Teme sastanka:
  - sastanak s demonstratorom – pregled i diskusija trenutnog stanja dokumentacije prije službene predaje

## 7. sastanak

- Datum: 11.12.2019.
- Prisustvovali: P.Matošević, D.V.Cvitković, D.Facković, N.Kolarec, J.Prpić
- Teme sastanka:
  - upoznavanje s radnim okvirima i bibliotekama kod razvoja
  - okvirni plan nastavka rada

## 8. sastanak

- Datum: 09.01.2020.
- Prisustvovali: P.Matošević, D.V.Cvitković, D.Facković, M.Has
- Teme sastanka:
  - sastanak s demonstratorom - pregled i diskusija trenutnog stanja aplikacije prije službene predaje

## Tablica aktivnosti

	Patrik Matošević	Katarina Boras	Daria Vanesa Cvitković	Dora Facković	Mislav Has	Nina Kolarec	Juraj Prpić
Upravljanje projektom	15						
Opis projektnog zadatka		4	5	5			
Funkcionalni zahtjevi					3		6
Opis pojedinih obrazaca	3				1	4	
Dijagram obrazaca	3						
Sekvencijski dijagrami		6	6	6	2		2
Opis ostalih zahtjeva					3		1
Arhitektura i dizajn sustava	4						
Baza podataka	4	3		2			
Dijagram razreda	4	1		2			11
Dijagram stanja		1	7	7		5	
Dijagram aktivnosti		0.5	6	6		5	
Dijagram komponenti	2					3	
Korištene tehnologije i alati	1						0.5
Ispitivanje programskog rješenja		1.5					5
Dijagram razmještaja	0.5						
Upute za puštanje u pogon	1						
Dnevnik sastajanja	1						
Zaključak i budući rad		2					
Popis literature							
Postavljanje Spring okvira i izrada predložaka web sučelja	3						



	<b>Patrik Matošević</b>	<b>Katarina Boras</b>	<b>Daria Vanesa Cvitković</b>	<b>Dora Facković</b>	<b>Mislav Has</b>	<b>Nina Kolarec</b>	<b>Juraj Prpić</b>
Izrada modela i spajanje baze podataka	4						
Implementacija backenda i frontenda prijave	6						
Implementacija backenda i frontenda registracije	1					9	
Konfiguriranje poslužitelja i puštanje aplikacije u pogon	4						
Implementacija backenda i frontenda temeljnih funkcionalnosti	50		6	6		7	
Rad na frontendu i poboljšanje dizajna		20					
Unit testiranje	1						
Selenium testiranje	4		6	6		5	

## Dijagrami pregleda promjena

### *dio 2. revizije*

*Prenijeti dijagram pregleda promjena nad datotekama projekta. Potrebno je na kraju projekta generirane grafove s gitlaba prenijeti u ovo poglavlje dokumentacije. Dijagrami za vlastiti projekt se mogu preuzeti s [gitlab.com](https://gitlab.com) stranice, u izborniku Repository, pritiskom na stavku Contributors.*