

# Oblikovanje programske potpore

Ak. god. 2019./2020.

## Moj AutoServis

Dokumentacija, Rev. 1.0

Grupa: *InfiniTeam*

Voditelj: *Patrik Matošević*

Datum predaje: *15. studenog. 2019.*

Nastavnik: *Nikolina Frid*

# Sadržaj

<b>1 Dnevnik promjena dokumentacije</b>	<b>3</b>
<b>2 Opis projektnog zadatka</b>	<b>5</b>
<b>3 Specifikacija programske potpore</b>	<b>10</b>
3.1 Funkcionalni zahtjevi . . . . .	10
3.1.1 Obrasci uporabe . . . . .	13
3.1.2 Sekvencijski dijagrami . . . . .	28
3.2 Ostali zahtjevi . . . . .	32
<b>4 Arhitektura i dizajn sustava</b>	<b>33</b>
4.1 Baza podataka . . . . .	35
4.1.1 Opis tablica . . . . .	35
4.1.2 Dijagram baze podataka . . . . .	39
4.2 Dijagram razreda . . . . .	40
4.3 Dijagram stanja . . . . .	43
4.4 Dijagram aktivnosti . . . . .	44
4.5 Dijagram komponenti . . . . .	45
<b>5 Implementacija i korisničko sučelje</b>	<b>46</b>
5.1 Korištene tehnologije i alati . . . . .	46
5.2 Ispitivanje programskog rješenja . . . . .	47
5.2.1 Ispitivanje komponenti . . . . .	47
5.2.2 Ispitivanje sustava . . . . .	47
5.3 Dijagram razmještaja . . . . .	48
5.4 Upute za puštanje u pogon . . . . .	49
<b>6 Zaključak i budući rad</b>	<b>50</b>
<b>Popis literature</b>	<b>51</b>
<b>Indeks slika i dijagrama</b>	<b>52</b>

**Dodatak: Prikaz aktivnosti grupe**

**53**

# 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.

## 2. Opis projektnog zadatka

Cilj ovog projektnog zadatka je osmisлити 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,
- auto-servis, i
- serviser.

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

- svoje ime,
- svoje prezime,
- OIB, te
- adresu svoje e-pošte.

U slučaju da se radi o tvrtci, podatci koji su potrebni su:

- naziv,
- adresa, i
- OIB tvrtke.

Nakon registracije, korisnik unosi podatke o automobilu iz prometne dozvole, među kojima su obavezno:

- registracijska oznaka, te
- marka automobila.

Kako bi novi automobil bio dodan, uneseni podaci se provjeravaju u HUUO registru, iz kojeg se dobiva marka, model, i broj šasije automobila. U slučaju nepodudaranja podataka vozila sa podacima u HUUO registru, odbija se registracija 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.

Auto servisi također moraju stvoriti svoje profile u aplikaciji, odnosno moraju se registrirati kao korisnici sa podacima:

- naziv,
- adresa, i
- OIB.

Nakon registracije, auto servisi trebaju definirati korisnički račun za administratora auto servisa i za servisere. Administrator auto servisa potom unosi cjenik usluga sa podacima:

- vrsta usluge, i
- cijena usluge,

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

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

kao i listu serviseri sa podacima, preko koje mogu upravljati svojim zaposlenicima.

Kada korisnik (vlasnik automobila) želi poslati svoj automobil na servis, mora otvoriti radni nalog, prilikom čega odabire auto servis u kojem želi obaviti servis. Odabire redoviti ili izvanredni servis. Nakon što je rad na automobilu obavljen, korisniku dolaze detalji servisa sa cijenom.

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, zamjena ulja, filtera i sl. Naknadno, 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 kao i ugrađenim rezervnim dijelovima.

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

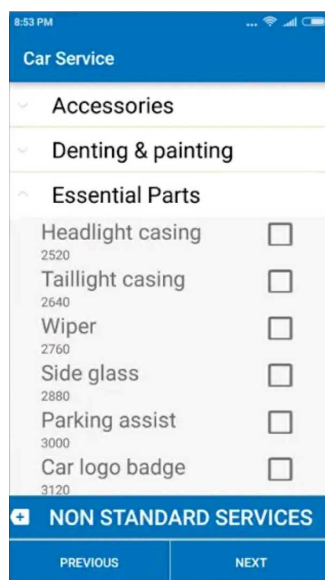
Glavni administrator odobrava registracije auto servisa, ima upravu nad svim korisnicima, može brisati račune i ima pristup svim statističkim podacima koji se izračunavaju.



## Primjeri sličnih rješenja

Dvije aplikacije koje su najbližnije 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.

RO	STATUS	DATE	CUSTOMER	PHONE	VEHICLE	TOTAL	LICENSE	W+TK	TYPE
2987	INSPECTION	10/22/2019		H: (860) 829-7377	2015 CHEVROLET SILVERADO 2500H	1,128.69		DEMOJC	ANOTHER TEST
2986	INSPECTION	10/17/2019	TEST 123 TEST123	H: (860) 829-7377	#977 2013 HYUNDAI SANTA FE SPORT 2.	95.22		DEMOJC	ANOTHER TEST
2985	INSPECTION	10/17/2019		H: (860) 829-7377	2011 JAGUAR XF PREMIUM	14.22	BKKA 198	DEMO	ANOTHER TEST
2983	INSPECTION	10/15/2019	ROCIO SALAZAR	C: (480) 524-3361	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	DEMOJC	ANOTHER TEST
2981	INSPECTION	10/14/2019	KONRAD CURZE		2004 VOLKSWAGEN PASSAT GL SEDA	443.98		DEMOFM	ANOTHER TEST
2979	INSPECTION	10/10/2019	123 123		11 TOYOTA 4RUNNER	14.22		DEMO	ANOTHER TEST
2978	INSPECTION	10/10/2019	RICKY BOBBY	C: (410) 687-5309	2001 SATURN L200	645.05	GHL9219	DEMOJC	ANOTHER TEST
2977	INSPECTION	10/10/2019	JOHN BETA	C: (098) 765-4321	2009 FORDI TAURUS SE AWD BLACK	90.30		DEMO	ANOTHER TEST
2976	INSPECTION	10/09/2019	ROBOUTE GULLIMAN	C: (860) 555-1066	1985 VOLKSWAGEN BEETLE BLUE	14.22		DEMO	ANOTHER TEST
2974	INSPECTION	10/09/2019	JOHN BETA	C: (098) 765-4321	2009 FORDI TAURUS SE AWD BLACK	14.22		DEMO	ANOTHER TEST
2973	INSPECTION	10/08/2019	ME		2006 VOLKSWAGEN NEW BEETLE 2.5	14.22		DEMOJD	ANOTHER TEST
2972	INSPECTION	10/08/2019	KONRAD CURZE		2004 VOLKSWAGEN PASSAT GL SEDA	60.09		DEMO	ANOTHER TEST
2971	INSPECTION	10/08/2019		H: (860) 829-7377	1993 OLDSMOBILE 88	14.22		DEMO	ANOTHER TEST
2970	INSPECTION	10/08/2019	ROGAL DORN	C: (888) 555-1941	2006 AUDI A3 2.0T HATCHBACK 4-	223.53		DEMOHM	ANOTHER TEST
2968	INSPECTION	10/07/2019	CORVUS CORAX	H: (555) 555-5555	2004 VOLKSWAGEN PASSAT GL SEDA	182.22		DEMO	ANOTHER TEST
2964	INSPECTION	10/05/2019		H: (860) 829-7377	2013 HYUNDAI ELANTRA GLS/UMIT	14.22	BJLM 182	DEMO	ANOTHER TEST
2963	INSPECTION	10/04/2019	BRIAN BARTLETT	H: (727) 417-3318	2012 GMC SIERRA 1500 SLT EXT.	211.78	AWBE24	DEMO	ANOTHER TEST
2962	INSPECTION	10/03/2019	FIRST TRY	H: (555) 555-555 V	2004 ACURA RSX	310.42		DEMOJC	ANOTHER TEST
2961	INSPECTION	10/03/2019	JOE MCNARSON	W: (917) 876-3211	2011 CHRYSLER TOWN & COUNTRY T	523.02	PLK-457U	DEMO	ANOTHER TEST
2959	INSPECTION	09/27/2019	JEFF CHAMBERS	C: (443) 000-0000	2003 FORD EXPLORER BLUE	42.48	379CBB	DEMO	ANOTHER TEST
2958	INSPECTION	09/26/2019	DAYLINE			738.16		DEMOJC	ANOTHER TEST
2956	INSPECTION	09/25/2019	123			14.22		DEMO	ANOTHER TEST
2955	INSPECTION	09/25/2019	CELIO ARIAS	H: (317) 395-8571	2013 HONDA CIVIC SI SEDAN 6-SP	194.12		DEMOJC	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.

## 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 izvandredni 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 dodjeljena 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) Prihvatanje radnog naloga
- (b) Upravljanje rezervnim dijelovima
  - i. Dodavanje rezervnog dijela
  - ii. Uređivanje rezervnog dijela
  - iii. Brisanje rezervnog dijela
- (c) Uređivanje podataka autoservisa
- (d) Upravljanje serviserima
  - i. Dodavanje serviseri
  - ii. Uklanjanje serviseri

5. Serviser (inicijator) može:

- (a) Rad na radnom nalogu
  - i. Uređivanje naloga redovnog servisa
  - ii. Zatvaranje radnog naloga
  - iii. 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 -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

#### UC28 -Uklanjanje serviser

- **Glavni sudionik:** Vlasnik autoservisa
- **Cilj:** Uklanjanje serviser 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 serviser kojeg želi ukloniti
  3. Korisnički račun serviser se briše iz baze podataka

#### UC29 -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

#### UC30 -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

#### UC31 - Brisanje autoservisa

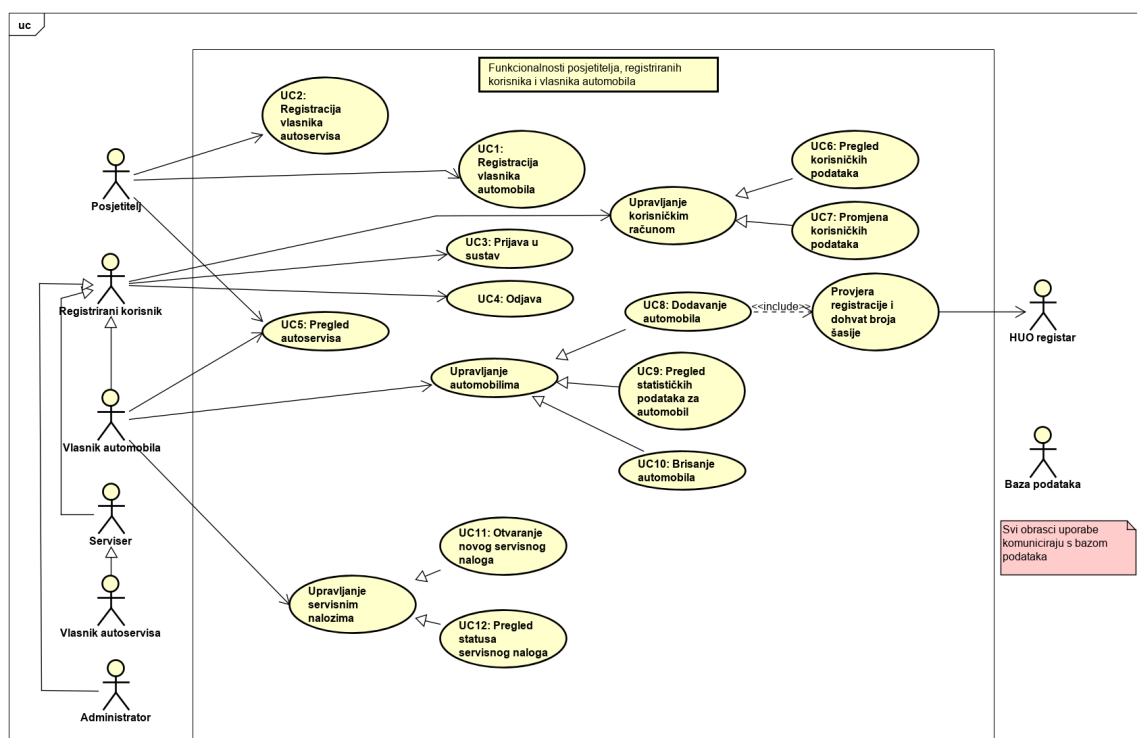
- **Glavni sudionik:** Administrator
- **Cilj:** Brisanje odabranog autoservisa
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav i ima ulogu administratora
- **Opis osnovnog tijeka:**
  1. Administrator odabere opciju upravljanja autoservisima
  2. Sustav prikaže listu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

#### UC32 -Pregled podataka autoservisa

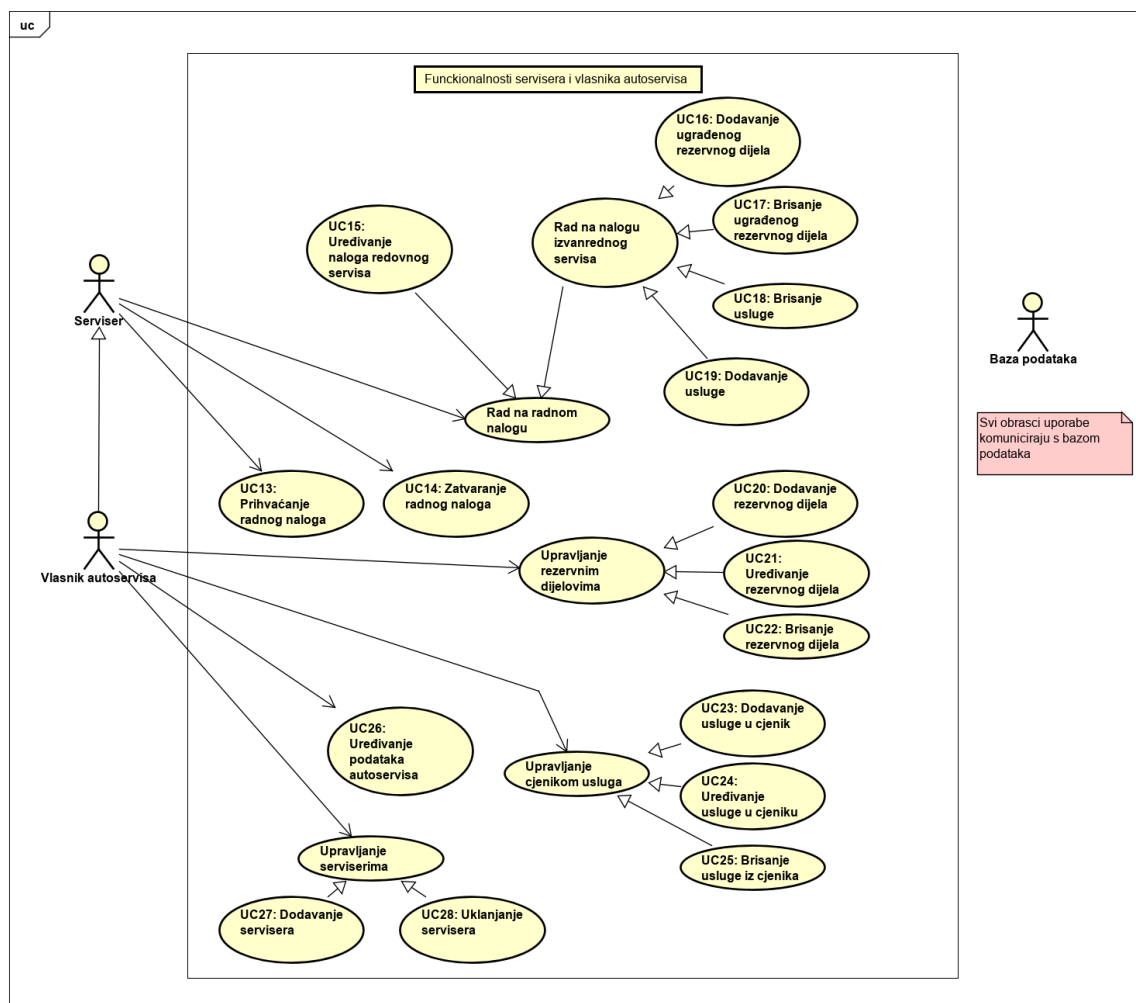
- **Glavni sudionik:** Administrator
- **Cilj:** Pregled podataka odabranog autoservisa
- **Sudionici:** Baza podataka

- **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 podaci o autoservisu dohvaćeni iz baze podataka

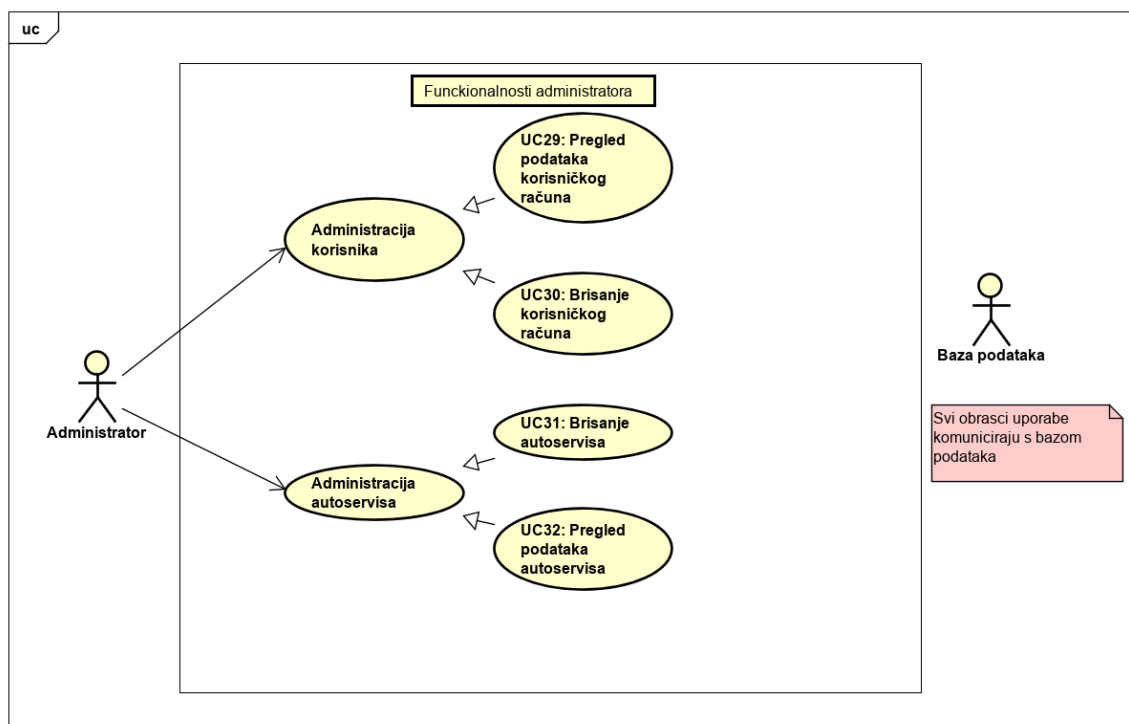
## Dijagrami obrazaca uporabe



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



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

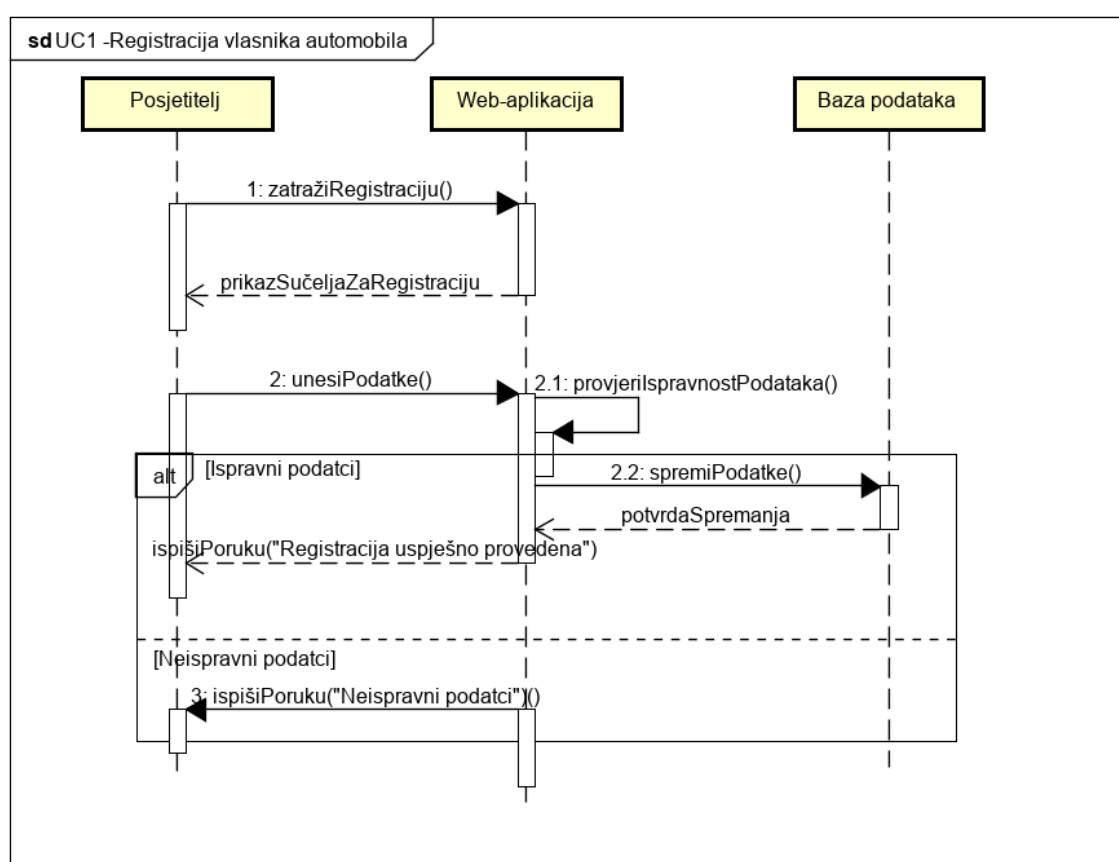


Slika 3.3: Dijagram obrasca uporabe, funkcionalnost administratora

### 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.

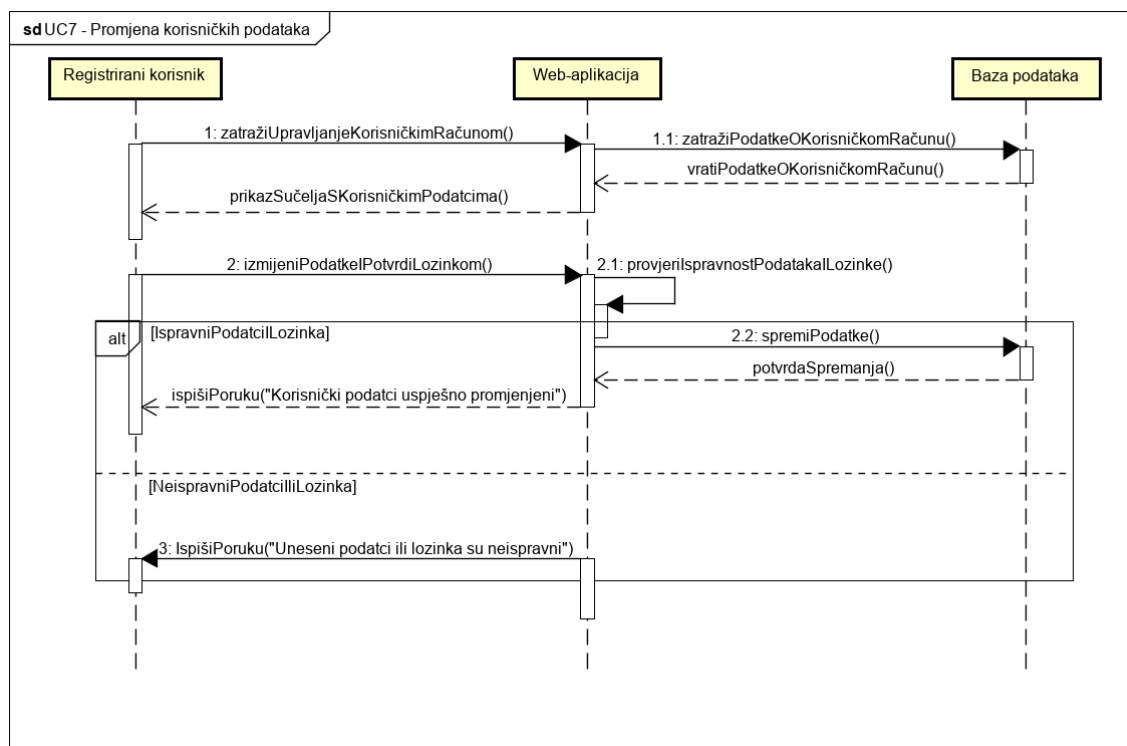


Slika 3.4: Sekvencijski dijagram za UC1

#### 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

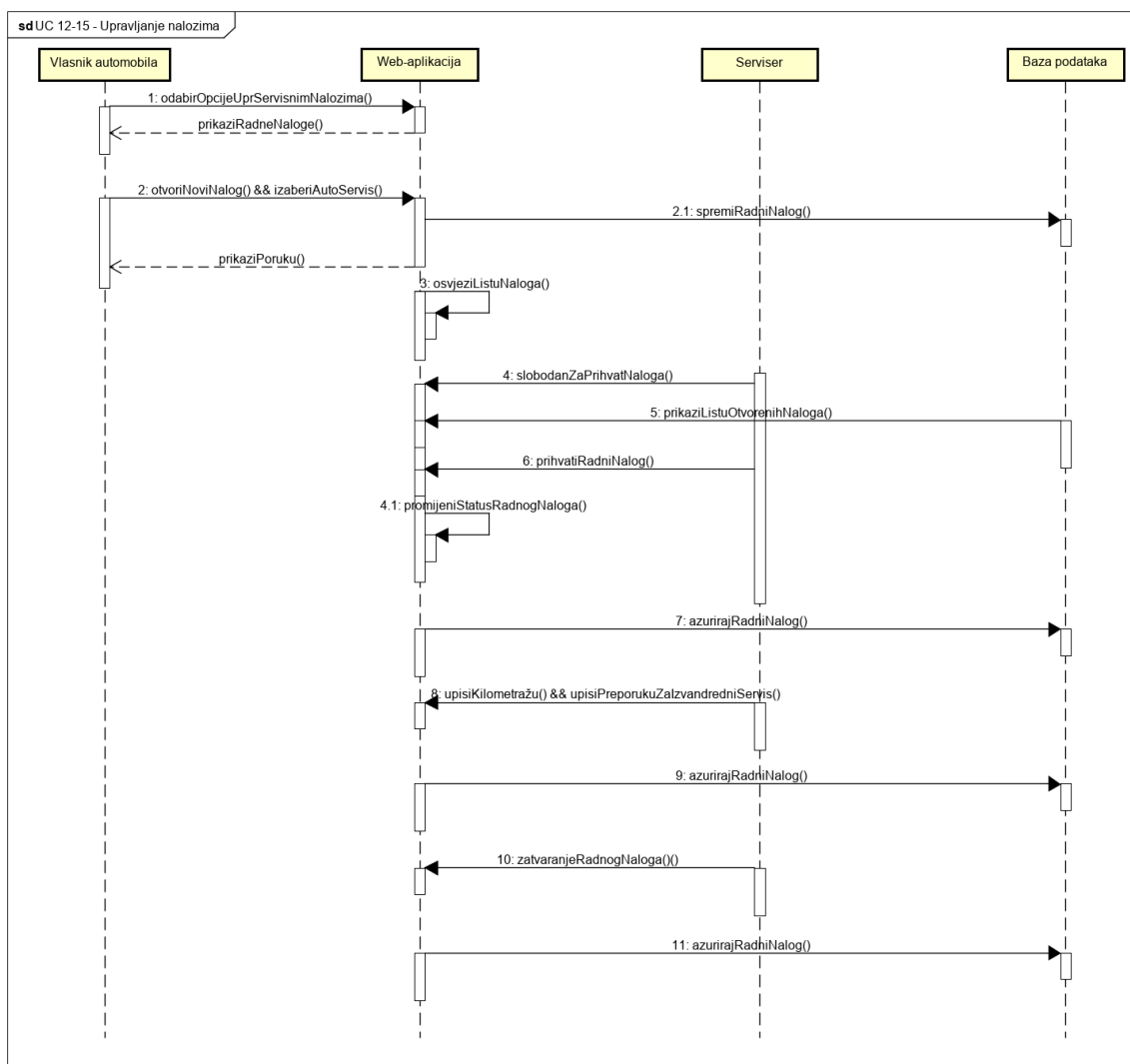
te ukoliko je ona ispravna i ako su novi podatci ispravni, novi podatci se spremaju u bazu podataka i korisniku se prikazuje poruka o uspješnoj promjeni podataka. Ukoliko je unesena lozinka neispravna ili ukoliko su novi podatci neispravni, korisniku se prikazuje poruka o neispravnosti unesene lozinke ili unesenih podataka.



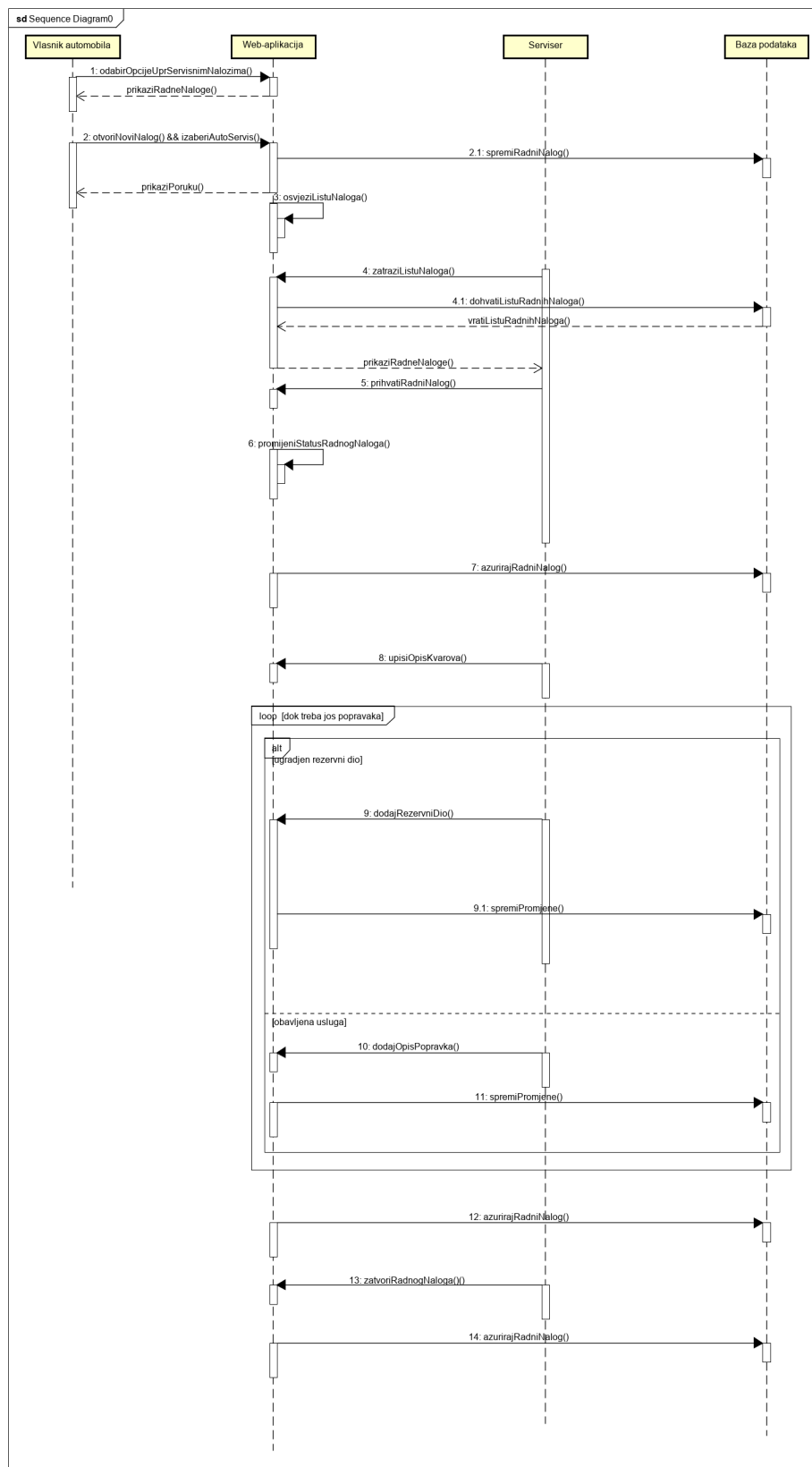
Slika 3.5: Sekvencijski dijagram za promjenu korisničkih podataka, UC7

### 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



Slika 3.7: Sekvencijski dijagrami za tijek izvanrednog servisa



## 3.2 Ostali zahtjevi

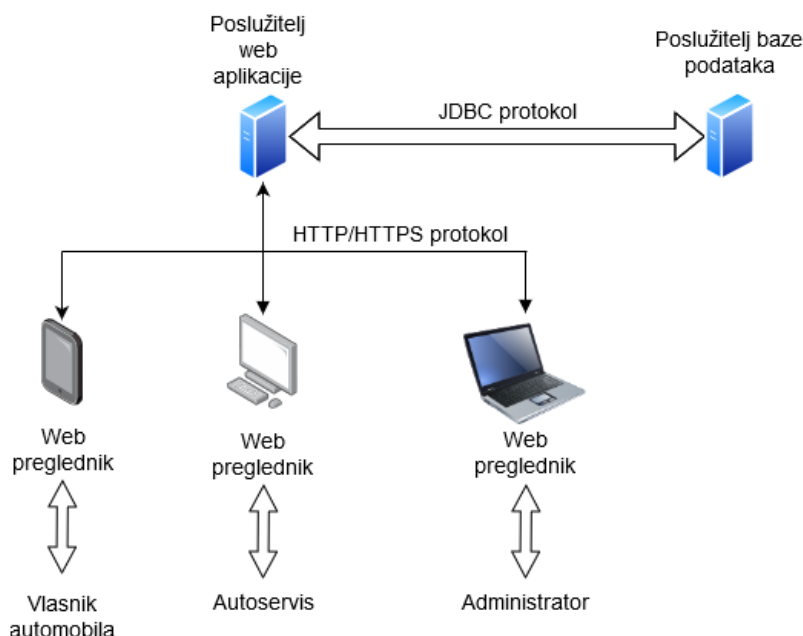
### *dio 1. revizije*

- 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 HUU registra
- Izvršavanje dijelova aplikacije (uključujući i pristup HUU 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 pristupati 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.

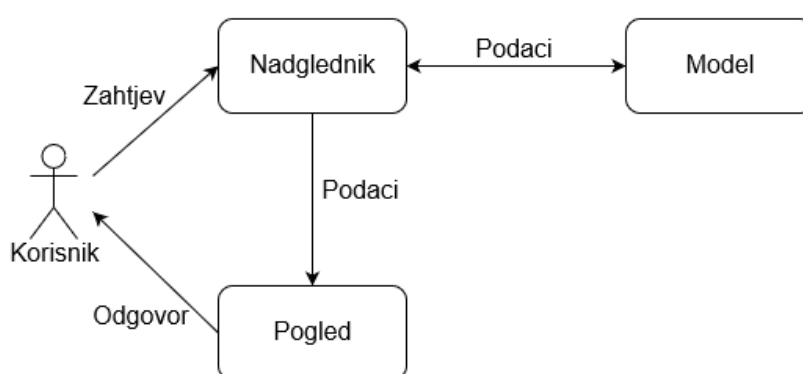


Slika 4.1: Osnovna arhitektura cijelog sustava

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 ravijen 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

## 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. 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

**app.user** Ovaj entitet sadrži sve podatke o registriranom korisniku aplikacije te predstavlja vlasnika automobila, servisera 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.
userid	BIGINT	Jedinstveni identifikacijski broj korisnika.

Korisnik		
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`.

repair_order (Radni nalog)		
repair_order_id	BIGINT	Jedinstveni identifikacijski broj radnog naloga.

Radni nalog		
price	DOUBLE PRECISION	Ukupna cijena.
service_job_status	VARCHAR	Status radnog naloga: otvoren, zatvoren ili u obradi.
auto_service_id	BIGINT	Identifikacijski broj autoservisa.
vehicle_id	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.

auto_service (Autoservis)		
auto_service_id	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.

regular_repair_order (Radni nalog za redoviti servis)		
kilometers	INTEGER	Kilometraža vozila zabilježena na servisu.
observed_malfunctions	VARCHAR	Primijećeni kvarovi.
repair_recommended	BOOLEAN	Je li potreban izvanredni servis ili ne?
id	BIGINT	Identifikacijski broj radnog naloga.

**repairing\_repair\_order** Ovaj entitet sadrži sve važne informacije za izvanredni servis 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.
<code>id</code>	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>		
<code>item_id</code>	BIGINT	Identifikator stavke.
<code>name</code>	VARCHAR	Naziv stavke.
<code>price</code>	DOUBLE PRECISION	Cijena stavke.
<code>repair_order_id</code>	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>		
<code>service_labor_id</code>	BIGINT	Identifikator usluge.
<code>price</code>	DOUBLE PRECISION	Cijena usluge.
<code>service_name</code>	VARCHAR	Naziv usluge.
<code>auto_service_id</code>	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>		
part_id	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.
auto_service_id	BIGINT	Identifikator autoservisa koji ima rezervni dio.

#### 4.1.2 Dijagram baze podataka





Slika 4.3: Relacijski model baze podataka

## 4.2 Dijagram razreda

Razred *AppUser* predstavlja registriranog korisnika koji nakon unosa svojih podataka može koristiti osnovne funkcionalnosti aplikacije. Razred *Administrator* predstavlja administratora koji nakon registracije ima najveće ovlasti. Razred *VehicleOwner* predstavlja registriranog vlasnika automobila. Razred *Vehicle* predstavlja vozilo koje pripada nekom korisniku. Razred *AutoService* označava jedan auto servis. Razred *ServiceEmployee* predstavlja osobu zaposlenu u auto servisu, koja može biti vlasnik tog auto servisa ili serviser koji radi na automobilima. Razred *RepairOrder* predstavlja radni nalog za servis. On ima dva podrazreda, razred *Re-*

```

classDiagram
    class AppUser {
        -userId: Long
        -username: String
        -password: String
        -email: String
        -firstName: String
        -lastName: String
        +getUserId(): Long
        +setUserId(userId: Long): void
        +getUsername(): String
        +setUsername(username: String): void
        +getPassword(): String
        +setPassword(password: String): void
        +getEmail(): String
        +setEmail(email: String): void
        +getFirstName(): String
        +setFirstName(firstName: String): void
        +getLastName(): String
        +setLastName(lastName: String): void
    }

    class Administrator {
    }

    class ServiceEmployeeType {
        <<enum>>
        + <<enum constant>> REGULAR_EMPLOYEE: ServiceEmployeeType
        + <<enum constant>> SERVICE_ADMINISTRATOR: ServiceEmployeeType
        - employeeType
    }

    class ServiceEmployee {
        +getAutoService(): AutoService
        +setAutoService(autoService: AutoService): void
        +getServiceEmployeeType(): ServiceEmployeeType
        +setEmployeeType(employeeType: ServiceEmployeeType): void
    }

    class ServiceLabor {
        -serviceLaborId: Long
        -serviceName: String
        -price: double
        +getServiceLaborId(): Long
        +setServiceLaborId(serviceLaborId: Long): void
        +getService(): AutoService
        +setAutoService(autoService: AutoService): void
        +getServiceName(): String
        +setServiceName(serviceName: String): void
        +getPrice(): double
        +setPrice(price: double): void
    }

    class VehicleOwner {
        -obj: String
        +getObj(): String
        +setObj(obj: String): void
        +getVehicles(): List<Vehicle>
        +setVehicles(vehicles: List<Vehicle>): void
    }

    class AutoService {
        -autoServiceId: Long
        -shopName: String
        -address: String
        -obj: String
        -regularServicePrice: double
        +getAutoServiceId(): Long
        +setAutoServiceId(autoServiceId: Long): void
        +getShopName(): String
        +setShopName(shopName: String): void
        +getAddress(): String
        +setAddress(address: String): void
        +getObj(): String
        +setObj(obj: String): void
        +getRegularServicePrice(): double
        +setRegularServicePrice(regularServicePrice: double): void
        +getServiceLabor(): List<ServiceLabor>
        +setServiceLabors(serviceLabors: List<ServiceLabor>): void
        +getVehicleParts(): List<VehiclePart>
        +setVehicleParts(vehicleParts: List<VehiclePart>): void
        +getRepairOrders(): List<RepairOrder>
        +setRepairOrders(repairOrders: List<RepairOrder>): void
        +getEmployees(): List<ServiceEmployee>
        +setEmployees(employees: List<ServiceEmployee>): void
    }

    class VehiclePart {
        -partId: Long
        -partName: String
        -estimatedDurationInKm: int
        -price: double
        +getPartId(): Long
        +setPartId(partId: Long): void
        +getAutoService(): AutoService
        +setAutoService(autoService: AutoService): void
        +getPartName(): String
        +setPartName(partName: String): void
        +getEstimatedDurationInKm(): int
        +setEstimatedDurationInKm(estimatedDurationInKm: int): void
        +getPrice(): double
        +setPrice(price: double): void
    }

    class Vehicle {
        -vehicleId: Long
        -licensePlate: String
        -vinNumber: String
        -vehicleModel: String
        +getVehicleId(): Long
        +setVehicleId(vehicleId: Long): void
        +getLicensePlate(): String
        +setLicensePlate(licensePlate: String): void
        +getOwner(): VehicleOwner
        +setOwner(owner: VehicleOwner): void
        +getVinNumber(): String
        +setVinNumber(vinNumber: String): void
        +getVehicleModel(): String
        +setVehicleModel(vehicleModel: String): void
    }

    class RepairOrder {
        -repairOrderId: Long
        -price: double
        +getRepairOrderId(): Long
        +setRepairOrderId(repairOrderId: Long): void
        +getVehicle(): Vehicle
        +setVehicle(vehicle: Vehicle): void
        +getAutoService(): AutoService
        +setAutoService(autoService: AutoService): void
        +getServiceJobStatus(): ServiceJobStatus
        +setServiceJobStatus(serviceJobStatus: ServiceJobStatus): void
        +getPrice(): double
        +setPrice(price: double): void
    }

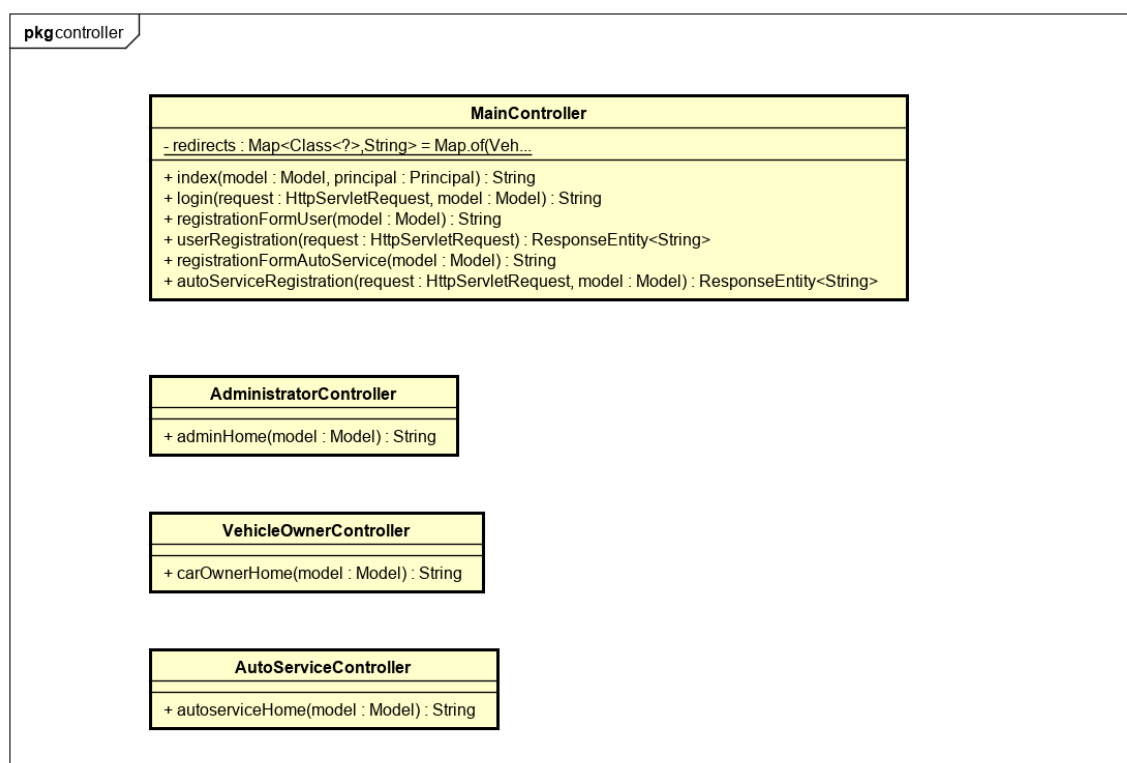
    class RegularRepairOrder {
        -kilometers: int
        -repairRecommended: boolean
        -observedMalfunctions: String
        +getKilometers(): int
        +setKilometers(kilometers: int): void
        +isRepairRecommended(): boolean
        +setRepairRecommended(repairRecommended: boolean): void
        +getObservedMalfunctions(): String
        +setObservedMalfunctions(observedMalfunctions: String): void
    }

    class RepairingRepairOrder {
        -malfunctions: String
        +getMalfunctions(): String
        +setMalfunctions(malfunctions: String): void
        +getItem(): List<RepairOrderItem>
        +setItems(items: List<RepairOrderItem>): void
    }

    class RepairOrderItem {
        -itemId: Long
        -name: String
        -price: double
        +getItemId(): Long
        +setItemId(itemId: Long): void
        +getRepairOrder(): RepairingRepairOrder
        +setRepairOrder(repairOrder: RepairingRepairOrder): void
        +getName(): String
        +setName(name: String): void
        +getPrice(): double
        +setPrice(price: double): void
    }

    AppUser <|-- Administrator
    AppUser <|-- ServiceEmployee
    ServiceEmployeeType <|-- ServiceEmployee
    ServiceEmployee "1" -- "*" ServiceLabor : - serviceLabor
    ServiceEmployee "1" -- "*" VehiclePart : - vehiclePart
    VehicleOwner "1" -- "*" Vehicle : - vehicles
    AutoService "1" -- "*" ServiceLabor : - serviceLabor
    AutoService "1" -- "*" VehiclePart : - vehiclePart
    AutoService "1" -- "*" RepairOrder : - repairOrders
    Vehicle "1" -- "*" RepairOrder : - vehicle
    RepairOrder "1" -- "*" RegularRepairOrder
    RepairOrder "1" -- "*" RepairingRepairOrder
    RepairingRepairOrder "1" -- "*" RepairOrderItem : - items
    ServiceJobStatus <|-- RegularRepairOrder
    ServiceJobStatus <|-- RepairingRepairOrder
    
```

Slika 4.4: Dijagram razreda modela



Slika 4.5: Dijagram razreda u sloju nadglednika

## 4.3 Dijagram stanja

### *dio 2. revizije*

*Potrebno je priložiti dijagram stanja i opisati ga. Dovoljan je jedan dijagram stanja koji prikazuje **značajan dio funkcionalnosti** sustava. Na primjer, stanja korisničkog sučelja i tijekom korištenja neke ključne funkcionalnosti jesu značajan dio sustava, a registracija i prijava nisu.*

## 4.4 Dijagram aktivnosti

### *dio 2. revizije*

*Potrebno je priložiti dijagram aktivnosti s pripadajućim opisom. Dijagram aktivnosti treba prikazivati značajan dio sustava.*

## 4.5 Dijagram komponenti

### *dio 2. revizije*

*Potrebno je priložiti dijagram komponenti s pripadajućim opisom. Dijagram komponenti treba prikazivati strukturu cijele aplikacije.*

## 5. Implementacija i korisničko sučelje

### 5.1 Korištene tehnologije i alati

#### *dio 2. revizije*

*Detaljno navesti sve tehnologije i alate koji su primijenjeni pri izradi dokumentacije i aplikacije. Ukratko ih opisati, te navesti njihovo značenje i mjesto primjene. Za svaki navedeni alat i tehnologiju je potrebno **navesti internet poveznicu** gdje se mogu preuzeti ili više saznati o njima.*

## 5.2 Ispitivanje programskog rješenja

### dio 2. revizije

*U ovom poglavlju je potrebno opisati provedbu ispitivanja implementiranih funkcionalnosti na razini komponenti i na razini cijelog sustava s prikazom odabranih ispitnih slučajeva. Studenti trebaju ispitati temeljnu funkcionalnost i rubne uvjete.*

### 5.2.1 Ispitivanje komponenti

*Potrebno je provesti ispitivanje jedinica (engl. unit testing) nad razredima koji implementiraju temeljne funkcionalnosti. Razraditi **minimalno 6 ispitnih slučajeva** u kojima će se ispitati redovni slučajevi, rubni uvjeti te izazivanje pogreške (engl. exception throwing). Poželjno je stvoriti i ispitni slučaj koji koristi funkcionalnosti koje nisu implementirane. Potrebno je priložiti izvorni kôd svih ispitnih slučajeva te prikaz rezultata izvođenja ispita u razvojnom okruženju (prolaz/pad ispita).*

### 5.2.2 Ispitivanje sustava

*Potrebno je provesti i opisati ispitivanje sustava koristeći radni okvir Selenium<sup>1</sup>. Razraditi **minimalno 4 ispitna slučaja** u kojima će se ispitati redovni slučajevi, rubni uvjeti te poziv funkcionalnosti koja nije implementirana/izaziva pogrešku kako bi se vidjelo na koji način sustav reagira kada nešto nije u potpunosti ostvareno. Ispitni slučaj se treba sastojati od ulaza (npr. korisničko ime i lozinka), očekivanog izlaza ili rezultata, koraka ispitivanja i dobivenog izlaza ili rezultata.*

*Izradu ispitnih slučajeva pomoću radnog okvira Selenium moguće je provesti pomoću jednog od sljedeća dva alata:*

- *dodatak za preglednik **Selenium IDE** - snimanje korisnikovih akcija radi automatskog ponavljanja ispita*
- ***Selenium WebDriver** - podrška za pisanje ispita u jezicima Java, C#, PHP koristeći posebno programsko sučelje.*

*Detalji o korištenju alata Selenium bit će prikazani na posebnom predavanju tijekom semestra.*

---

<sup>1</sup><https://www.seleniumhq.org/>



## 5.3 Dijagram razmještaja

### *dio 2. revizije*

*Potrebno je umetnuti **specifikacijski** dijagram razmještaja i opisati ga. Moguće je umjesto specifikacijskog dijagrama razmještaja umetnuti dijagram razmještaja instanci, pod uvjetom da taj dijagram bolje opisuje neki važniji dio sustava.*

## 5.4 Upute za puštanje u pogon

### *dio 2. revizije*

*U ovom poglavlju potrebno je dati upute za puštanje u pogon (engl. deployment) ostvarene aplikacije. Na primjer, za web aplikacije, opisati postupak kojim se od izvornog kôda dolazi do potpuno postavljene baze podataka i poslužitelja koji odgovara na upite korisnika. Za mobilnu aplikaciju, postupak kojim se aplikacija izgradi, te postavi na neku od trgovina. Za stolnu (engl. desktop) aplikaciju, postupak kojim se aplikacija instalira na računalo. Ukoliko mobilne i stolne aplikacije komuniciraju s poslužiteljem i/ili bazom podataka, opisati i postupak njihovog postavljanja. Pri izradi uputa preporučuje se **naglasiti korake instalacije uporabom natuknica** te koristiti što je više moguće **slike ekrana** (engl. screenshots) kako bi upute bile jasne i jednostavne za slijediti.*

*Dovršenu aplikaciju potrebno je pokrenuti na javno dostupnom poslužitelju. Studentima se preporuča korištenje neke od sljedećih besplatnih usluga: Amazon AWS, Microsoft Azure ili Heroku. Mobilne aplikacije trebaju biti objavljene na F-Droid, Google Play ili Amazon App trgovini.*

## 6. Zaključak i budući rad

### *dio 2. revizije*

*U ovom poglavlju potrebno je napisati osvrt na vrijeme izrade projektnog zadatka, koji su tehnički izazovi prepoznati, jesu li riješeni ili kako bi mogli biti riješeni, koja su znanja stečena pri izradi projekta, koja bi znanja bila posebno potrebna za brže i kvalitetnije ostvarenje projekta i koje bi bile perspektive za nastavak rada u projektnoj grupi.*

*Potrebno je točno popisati funkcionalnosti koje nisu implementirane u ostvarenoj aplikaciji.*

# 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 . . . . .	8
2.2	ShopBoss . . . . .	8
3.1	Dijagram obrasca uporabe, funkcionalnost posjetitelja, registriranog korisnika i vlasnika automobila . . . . .	25
3.2	Dijagram obrasca uporabe, funkcionalnost servisera i vlasnika auto-servisa . . . . .	26
3.3	Dijagram obrasca uporabe, funkcionalnost administratora . . . . .	27
3.4	Sekvencijski dijagram za UC1 . . . . .	28
3.5	Sekvencijski dijagram za promjenu korisničkih podataka, UC7 . . . . .	29
3.6	Sekvencijski dijagram za tijek redovnog servisa, UC12–UC15 . . . . .	30
3.7	Sekvencijski dijagrami za tijek izvanrednog servisa . . . . .	31
4.1	Osnovna arhitektura cijelog sustava . . . . .	33
4.2	Oblikovni obrazac Model-pogled-nadglednik . . . . .	34
4.3	Relacijski model baze podataka . . . . .	40
4.4	Dijagram razreda modela . . . . .	41
4.5	Dijagram razreda u sloju nadglednika . . . . .	42

# 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: u ovom formatu: 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

## Tablica aktivnosti

	Patrik Matošević	Katarina Boras	Daria Vanesa Cvitković	Dora Facković	Mislav Has	Nina Kolarec	Juraj Prpić
Upravljanje projektom							
Opis projektnog zadatka							
Funkcionalni zahtjevi							
Opis pojedinih obrazaca							
Dijagram obrazaca							
Sekvencijski dijagrami							
Opis ostalih zahtjeva							
Arhitektura i dizajn sustava							
Baza podataka							
Dijagram razreda							
Dijagram stanja							
Dijagram aktivnosti							
Dijagram komponenti							
Korištene tehnologije i alati							
Ispitivanje programskog rješenja							
Dijagram razmještaja							
Upute za puštanje u pogon							
Dnevnik sastajanja							
Zaključak i budući rad							
Popis literature							
<i>Dodatne stavke kako ste podijelili izradu aplikacije</i>							
<i>npr. izrada početne stranice</i>							



	<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>
<i>izrada baze podataka</i>							
<i>spajanje s bazom podataka</i>							
<i>back end</i>							

## 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.*