

Oblikovanje programske podpore

Ak. god. 2018./2019.

Sustav za praćenje rada auto servisa

Dokumentacija, Rev. 2

Grupa: *BrainStorm*

Voditelj: *Tin Komerički*

Datum predaje: *17. siječnja 2019.*

Nastavnik: *Dr. sc. Miljenko Krhen*

Contents

1.	Dnevnik promjena dokumentacije	1
2.	Opis projektnog zadatka.....	2
3.	Pojmovnik.....	4
4.	Funkcionalni zahtjevi	5
4.1.	Dionici.....	5
4.2.	Aktori	5
4.2.1.	Administrator	5
4.2.2.	Ovlašteni serviser	5
4.2.3.	Registrirani korisnik.....	5
4.2.4.	Neregistrirani korisnik	5
4.2.5.	Baza podataka	5
4.3.	Opisi obrazaca uporabe.....	6
4.4.	Dijagrami obrazaca uporabe	11
4.4.1.	Dijagram obrazaca uporabe za cijeli auto servis	11
4.4.2.	Dijagram obrazaca uporabe za administratora	12
4.4.3.	Dijagram obrazaca uporabe za korisnika.....	12
4.4.4.	Dijagram obrazaca uporabe za prijavu	13
4.4.5.	Dijagram obrazaca uporabe za serviseru.....	13
4.4.6.	Dijagram obrazaca uporabe za obradu prijave vozila	14
4.5.	Sekvencijski dijagrami.....	15
4.5.1.	Osnovni upis podataka	15
4.5.2.	Potvrda prijave vozila na popravak	15
4.5.3.	Pregled vlastitih korisnika.....	16
4.5.4.	Prijava na sustav	16
4.5.5.	Prijava vozila na popravak	17
4.5.6.	Registracija	18
4.5.7.	Upravljanje informacijama registriranih korisnika	18
5.	Ostali zahtjevi	19
6.	Arhitektura i dizajn sustava	20
6.1.	Arhitekturni obrazac MVC	20
6.2.	ER model baze podataka.....	21
6.2.1.	Opisi entiteta ER modela baze podataka	22
6.2.2.	CREATE naredbe relacijskog modela baze podataka	25
6.3.	Dijagram razreda	28
6.4.	Dijagram objekata	29

6.5.	Ostali UML dijagrami	30
6.5.1.	Dijagram stanja – prijava na sustav	30
6.5.2.	Dijagram stanja – izmjena podataka o korisniku.....	31
6.5.3.	Dijagram aktivnosti – registracija korisnika.....	32
6.5.4.	Dijagram aktivnosti – pregled korisnika	33
6.5.5.	Komunikacijski dijagram – prijava registriranog korisnika na sustav	34
6.5.6.	Komunikacijski dijagram – prijava vozila na popravak	35
6.5.7.	Dijagram komponenti.....	36
7.	Implementacija i korisničko sučelje.....	37
7.1.	Dijagram razmještaja.....	37
7.2.	Korišteni alati.....	38
7.3.	Korisničke upute	39
7.4.	Isječak programskog koda vezan za temeljnu funkcionalnost sustava	50
7.5.	Ispitivanje programskog rješenja	55
7.5.1.	Registracija korisnika	55
7.5.2.	Prijava na korisnika na sustav.....	57
7.5.3.	Izmjena podataka o serviseru.....	59
7.5.4.	Izmjena prijave vozila na popravak	61
8.	Zaključak i budući rad	64
9.	Popis literature	65
Dodatak A:	Indeks (slika, dijagrama, tablica, ispisa kôda).....	66
Dodatak B:	Dnevnik sastajanja	68
Dodatak C:	Prikaz aktivnosti grupe.....	70
Dodatak D:	Plan rada.....	73

1. Dnevnik promjena dokumentacije

Rev.	Opis promjene	Autor(i)	Datum
0.2	Dodan dijagrami obrazaca uporabe i njihovi opisi	Tea Kalenski	07.11.2018.
0.3	Dodan opis projektnog zadatka i prvi dio šestog poglavlja	Dino Avdić Filip Jandžel	11.11.2018.
0.6	Dodani sekvencijski dijagrami i njihovi opisi, dodan opis pojmova	Filip Jandžel Tea Kalenski	16.11.2018.
0.7	Dodan kratki opis arhitekture i tehnologije	Filip Jandžel	18.11.2018.
0.8	Dodan dijagram razreda i objekata	Dino Avdić	27.11.2018.
0.9	Dodan ER model i opisi relacija baze podataka Dodana naslovna stranica i sadržaj	Dino Avdić Filip Jandžel Filip Kovačević Marko Prnić	27.11.2018.
1.0	Dodani dodaci A-D Dodan dnevnik promjena dokumentacije i ostali zahtjevi	Tin Komerički	28.11.2018.
1.3	Dodani svi preostali dijagrami	Dino Avdić	8.1.2019.
1.6	Promijenjeni informacije o bazi podataka, dodan relacijski model baze podataka	Marko Prnić	12.1.2019.
1.9	Dodani isječci koda, dodane upute za korištenje aplikacije	Filip Jandžel Filip Kovačević	15.1.2019.
2.0	Ažurirani svi dodaci i dnevnik dokumentacije, dovršeno sedmo poglavlje, napisano osmo poglavlje	Dino Avdić Filip Jandžel Tin Komerički Filip Kovačević Marko Prnić	17.1.2019.

2. Opis projektnog zadatka

Cilj ovog projekta jest izrada informacijskog sustava za auto kuću koja u svome sklopu ima i auto servis „Najbolji mehaničar“. Prema pravilima poslovanja auto kuće, na servis se primaju svi modeli samo jedne određene marke automobila. Cilj i potencijalna korist ovog sustava je olakšavanje međusobne komunikacije između serviseri i njegovih klijenata te sveukupno povećanje kvalitete rada auto servisa. Strukturiranim pristupom prijave i odjave automobila sa servisa postiže se efikasniji rad servisa. U tom pristupu definira se točno vrijeme kada klijenti dostavljaju i podižu vozila te se samim time serviseri oslobađaju mogućnosti prekida za vrijeme rada zbog neočekivanih dolazaka potencijalnih klijenata. Ovaj informacijski sustav bit će dostupan na web stranici auto kuće kao web aplikacija kojoj će se moći pristupiti koristeći internetski preglednik po izboru. Aplikacija će omogućiti prijavu korisnika na servis kao i praćenje statusa popravka vozila koji su na servisu duže od jednog dana. Radno vrijeme servisa je radnim danom od 7:00 do 18:00 sati. Korisnicima aplikacije prijava vozila biti će omogućena cijelo vrijeme 24/7, no vozila se zaprimaju na popravak svakog radnog dana samo do 10:00 sati ujutro. Vlasnici svoje vozilo mogu preuzeti svakog radnog dana poslije 14:00 sati.

Postoje četiri vrste korisnika ovog informacijskog sustava: administrator, ovlašteni serviser, registrirani korisnik i neregistrirani korisnik.

Administrator sustava nakon puštanja u produkciju, upisuje podatke o servisu i kontakt podatke. Također, upisuje i podatke o ovlaštenim serviserima, od kojih svaki mora imati svoje korisničko ime i lozinku. Administrator sustava može mijenjati, brisati i/ili dodavati informacije o sebi i svim registriranim korisnicima.

Ovlašteni serviser nakon spajanja na sustav ima pregled prijavljenih korisnika koji su odabrali njega, ili im ih je sustav sam odabrao, za taj radni dan. Prikazuju mu se svi podaci koje je registrirani korisnik upisao. Sve te podatke on može mijenjati, ali i dodavati nove (posebne napomene i sl.). Nakon završetka ovlašteni servis ispisuje obrazac koji daje na potpis korisniku, a koji korisnik zadržava za sebe kao potvrdu predaje vozila.

Registrirani korisnik upisuje prilikom registracije podatke o sebi, pri čemu su obavezni podaci njegovo ime, prezime, adresa elektroničke pošte, broj telefona, tip vozila, godina proizvodnje i registarska oznaka.

Neregistrirani korisnik može vidjeti samo osnovne informacije o uslugama servisa.

Procedura registracije korisnika predviđa unos podataka korisnika (obaveznih i neobaveznih). Ti podaci obavezno uključuju korisničko ime koje je identično njegovoj adresi elektroničke pošte, i lozinka. Nakon spajanja na sustav registriranom korisniku se najprije nudi mogućnost izbora ovlaštenog serviseru kod kojega se želi prijaviti. Odabir nije obavezan, i ukoliko korisnik ne odabere željenog serviseru, sustav će mu sam dodijeliti slobodnog, u ovisnosti o vremenu dolaska na servis. Nakon odabira serviseru, korisniku se prikazuje prikaz slobodnih termina odabranog serviseru u slijedećih 10 dana, ili općenito prikaz slobodnih termina. Nakon toga mu se nude standardne usluge na odabir, s time da korisnik može i sam dodati neku novu uslugu ili opis problema koji ima. Također se pita korisnika želi li zamjensko vozilo dok je njegovo na popravku. Vremenski intervali za prijavu na servis su u koracima od 20 minuta. Nakon uspješnog završetka prijave, registriranom korisniku se na njegovu adresu elektroničke pošte šalje detalja obavijest o prijavi. Ukoliko korisnik prijavljuje vozilo za popravak u periodu dužem od 3 dana, sustav ga automatski podsjeća na prijavu zadnji dan prije dana koji je odabrao kao željeni za dolazak na servis. Servis ima ukupno 10 zamjenskih vozila na raspolaganju za njihove korisnike. Sustav mora kontrolirati zauzeće vozila, i na zahtjev ih rezervirati za pojedinog korisnika. Sustav mora omogućiti istovremeni rad administratora, ovlaštenih serviseru i neograničenog broja registriranih korisnika.

Pri razvoju informacijskog sustava odabrana je objektno orijentirana arhitektura pri čemu se za izgradnju koristi programski jezik Java. Za razvoj web aplikacije koristi se Java Spring radni okvir. Kako bi aplikacija bila interaktivna i sadržavala moderan izgled, koriste se tehnologije: HTML, CSS, Javascript, JQuery i Bootstrap. Za ispravno funkcioniranje aplikacije, podatci o svim korisnicima i statusi popravaka vozila pohranjuju se u predviđenu bazu podataka. Sustav koristi relacijsku bazu podataka PostgreSQL.

3. Pojmovnik

Bootstrap – vrlo popularan CSS, HTML i Javascript radni okvir za izradu web aplikacija i responsive web stranica

CSS – (eng. Cascading Style Sheets) jezik za definiranje stila web aplikacija, npr. boja, razmaka, fontova)

HTML – (eng. HyperText Markup Language) označni jezik za definiranje strukture i sadržaja web aplikacije

Java – objektno orijentirani programski jezik kojeg je razvila tvrtka Sun Microsystems

Javascript – skriptni programski jezik koji se izvršava u korisnikovom pregledniku. Najčešće se koristi za izradu dinamičnih web aplikacija

Java Spring – radni okvir koji služi za kreiranje web aplikacija koristeći programski jezik Java

Spring Boot – Java Spring rješenje za jednostavno stvaranje samostalnih aplikacija koje su prethodno podešene kako bi se što lakše stvarale

Thymeleaf – java predložak za efektivno povezivanje frontenda i backenda u MVC aplikacijama

MySQL – relacijska baza podataka

4. Funkcionalni zahtjevi

4.1. Dionici

administrator, ovlašteni serviser, registrirani korisnik, neregistrirani korisnik i programeri.

4.2. Aktori

- Administrator
- Ovlašteni serviser
- Registrirani korisnik
- Neregistrirani korisnik
- Baza podataka

4.2.1. Administrator

- upisuje podatke o servisu
- upisuje kontakt podatke
- upisuje podatke o ovlaštenim serviserima
- može mijenjati, brisati i/ili dodavati informacije o sebi i svim registriranim korisnicima
- puštanje sustava u produkciju

4.2.2. Ovlašteni serviser

- Pregled vlastitih prijavljenih korisnika za taj radni dan
- može mijenjati podatke o vlastitim prijavljenim korisnicima
- može dodavati nove podatke o vlastitim prijavljenim korisnicima
- ispisuje obrazac koji daje na potpis korisniku

4.2.3. Registrirani korisnik

- upisuje prilikom registracije podatke o sebi
- Mogućnost prijave vozila na popravak
- Mogućnost izbora ovlaštenog serviser
- prikaz slobodnih termina
- Odabir standardnih usluga
- Mogućnost dodavanja nove usluge ili opisa problema koji ima
- Mogućnost odabira zamjenskog vozila dok je njegovo na popravku

4.2.4. Neregistrirani korisnik

- može vidjeti samo osnovne informacije o uslugama servisa.
- Zahtjev za registraciju

4.2.5. Baza podataka

- Pohranjuje osnovne podatke o stranici (podatke o servisu, kontakt podatke, podatke o ovlaštenim serviserima)
- Pohranjuje podatke o registriranim korisnicima
- Pohranjuje prijave vozila na popravak

4.3. Opisi obrazaca uporabe

UC1

- Naziv: Osnovni upis podataka
- Glavni sudionici: Administrator
- Cilj: upisuje podatke o servisu i upisuje kontakt podatke
- Sudionici: Baza Podataka
- Preduvjeti: -
- Željeni scenarij: Administrator upisuje podatke o servisu, upisuje kontakt podatke, sustav provjerava informacije i pohranjuje u bazu podataka
- Opis mogućih odstupanja: -

UC1.1

- Naziv: Unos ovlaštenih serviser
- Glavni sudionici: Administrator
- Cilj: upisuje podatke o ovlaštenim serviserima
- Sudionici: Baza Podataka
- Preduvjeti: -
- Željeni scenarij: Administrator upisuje podatke o ovlaštenim serviserima, sustav provjerava informacije i pohranjuje u bazu podataka
- Opis mogućih odstupanja: -

UC2

- Naziv: Upravljanje informacijama registriranih korisnika
- Glavni sudionici: Administrator
- Cilj: izmjena, brisanje i/ili dodavanje informacija o sebi i svim registriranim korisnicima
- Sudionici: Baza Podataka
- Preduvjeti: -
- Željeni scenarij: Administrator može mijenjati, brisati i/ili dodavati informacije o sebi i svim registriranim korisnicima, sustav provjerava informacije i pohranjuje u bazu podataka
- Opis mogućih odstupanja: -

UC3

- Naziv: Pregled osnovnih informacija
- Glavni sudionici: Neregistrirani korisnik
- Cilj: Neregistrirani korisnik može vidjeti osnovne informacije o uslugama servisa
- Sudionici: Baza podataka
- Preduvjeti: -
- Željeni scenarij: Neregistrirani korisnik pregledava osnovne informacije o uslugama servisa
- Opis mogućih odstupanja: -

UC4

- Naziv: Registracija
- Glavni sudionici: Neregistrirani korisnik
- Cilj: Registracija novih korisnika
- Sudionici: Baza podataka
- Preduvjeti: -
- Željeni scenarij: Neregistrirani korisnik pokreće registraciju, sustav dohvaća polja koja Neregistrirani korisnik mora ispuniti
- Opis mogućih odstupanja: -

UC4.1

- Naziv: Dodatni podaci
- Glavni sudionici: Neregistrirani korisnik
- Cilj: Unos dodatnih podataka pri registraciji novih korisnika
- Sudionici: Baza podataka
- Preduvjeti: -
- Željeni scenarij: Neregistrirani korisnik pri registraciji unosi dodatne podatke
- Opis mogućih odstupanja: -

UC5

- Naziv: Prijava na sustav
- Glavni sudionici: Registrirani korisnik
- Cilj: Prijava Registriranog korisnika na sustav
- Sudionici: Baza podataka
- Preduvjeti: Registracija
- Željeni scenarij: Registrirani korisnik pokreće prijavu u sustav te unosi svoje podatke potrebne za prijavu, sustav ga propušta ako su ti podaci ispravni
- Opis mogućih odstupanja: unesena pogrešna lozinka, zaboravljena lozinka

UC5.1

- Naziv: Pogrešni podaci
- Glavni sudionici: Registrirani korisnik
- Cilj: Obrada unosa pogrešnih podataka
- Sudionici: Baza podataka, sustav
- Preduvjeti: -
- Željeni scenarij: Registrirani korisnik pokreće prijavu u sustav te unosi svoje podatke potrebne za prijavu, sustav ga ne propušta ako su ti podaci neispravni
- Opis mogućih odstupanja: -

UC5.2

- Naziv: Zaboravljena lozinka
- Glavni sudionici: Registrirani korisnik
- Cilj: Pružanje mogućnosti promjene lozinke registriranim korisnicima koji su je zaboravili
- Sudionici: Baza podataka, sustav
- Preduvjeti: Registracija
- Željeni scenarij: Registrirani korisnik pokreće zahtjev za novom lozinkom
- Opis mogućih odstupanja: -

UC6

- Naziv: Prijava vozila na popravak
- Glavni sudionici: Registrirani korisnik
- Cilj: Prijavljivanje vozila na popravak
- Sudionici: Baza podataka, Ovlašteni serviser
- Preduvjeti: Registracija, Prijava na sustav
- Željeni scenarij: Registrirani korisnik pokreće prijavu vozila na popravak, sustav dohvaća potrebna polja koja Registrirani korisnik mora ispuniti
- Opis mogućih odstupanja: -

UC6.1

- Naziv: Odabir serviser
- Glavni sudionici: Registrirani korisnik
- Cilj: Registrirani korisnik odabire serviser
- Preduvjeti: Registracija, Prijava na sustav, Započeta prijava vozila na popravak
- Željeni scenarij: Registriranom korisniku sustav daje neobaveznu opciju izbora ovlaštenog serviser, ako on to ne učini serviser mu se dodjeljuje slobodnog, u ovisnosti o vremenu dolaska na servis
- Opis mogućih odstupanja: -

UC6.2

- Naziv: Odabir rezervnog vozila
- Glavni sudionici: Registrirani korisnik
- Cilj: Mogućnost korištenja rezervnog vozila
- Preduvjeti: Registracija, Prijava na sustav, Započeta prijava vozila na popravak
- Željeni scenarij: Registrirani korisnik ima mogućnost odabira zamjenskog vozila dok je njegovo na popravku
- Opis mogućih odstupanja: -

UC6.3

- Naziv: Podsjetnik na prijavu
- Glavni sudionici: Registrirani korisnik
- Cilj: Podsjećanje Registriranog korisnika na prijavu
- Sudionici: Sustav
- Preduvjeti: Registracija, Prijava na sustav, Započeta prijava vozila na popravak
- Željeni scenarij: Ukoliko korisnik prijavljuje vozilo za popravak u periodu dužem od 3 dana, sustav ga automatski podsjeća na prijavu zadnji dan prije dana koji je odabrao kao željeni za dolazak na servis
- Opis mogućih odstupanja: -

UC6.4

- Naziv: Obavijest o prijavi
- Glavni sudionici: Registrirani korisnik
- Cilj: slanje detalja o obavijest o prijavi korisniku
- Sudionici: Sustav
- Preduvjeti: Registracija, Prijava na sustav, Uspješna prijava vozila na popravak
- Željeni scenarij: Nakon uspješnog završetka prijave, registriranom korisniku se na njegovu adresu elektroničke pošte šalje detalja obavijest o prijavi
- Opis mogućih odstupanja: -

UC7

- Naziv: Pregled vlastitih korisnika
- Glavni sudionici: Ovlašteni serviser
- Cilj: Pregled vlastitih prijavljenih korisnika za taj radni dan
- Sudionici: Baza podataka
- Preduvjeti: Prijava na sustav
- Željeni scenarij: Ovlašteni serviser nakon spajanja na sustav ima pregled prijavljenih korisnika koji su odabrali njega, ili im ih je sustav sam odabrao, za taj radni dan. Prikazuju mu se svi podaci koje je registrirani korisnik upisao
- Opis mogućih odstupanja: -

UC7.1

- Naziv: Upravljanje dobivenim podacima
- Glavni sudionici: Ovlašteni serviser
- Cilj: može mijenjati podatke o vlastitim prijavljenim korisnicima te može dodavati nove podatke o vlastitim prijavljenim korisnicima
- Sudionici: Baza podataka
- Preduvjeti: Prijava na sustav
- Željeni scenarij: Ovlašteni serviser nakon spajanja na sustav sve podatke o vlastitim prijavljenim korisnicima može mijenjati, ali i dodavati nove (posebne napomene i sl.)
- Opis mogućih odstupanja: -

UC8

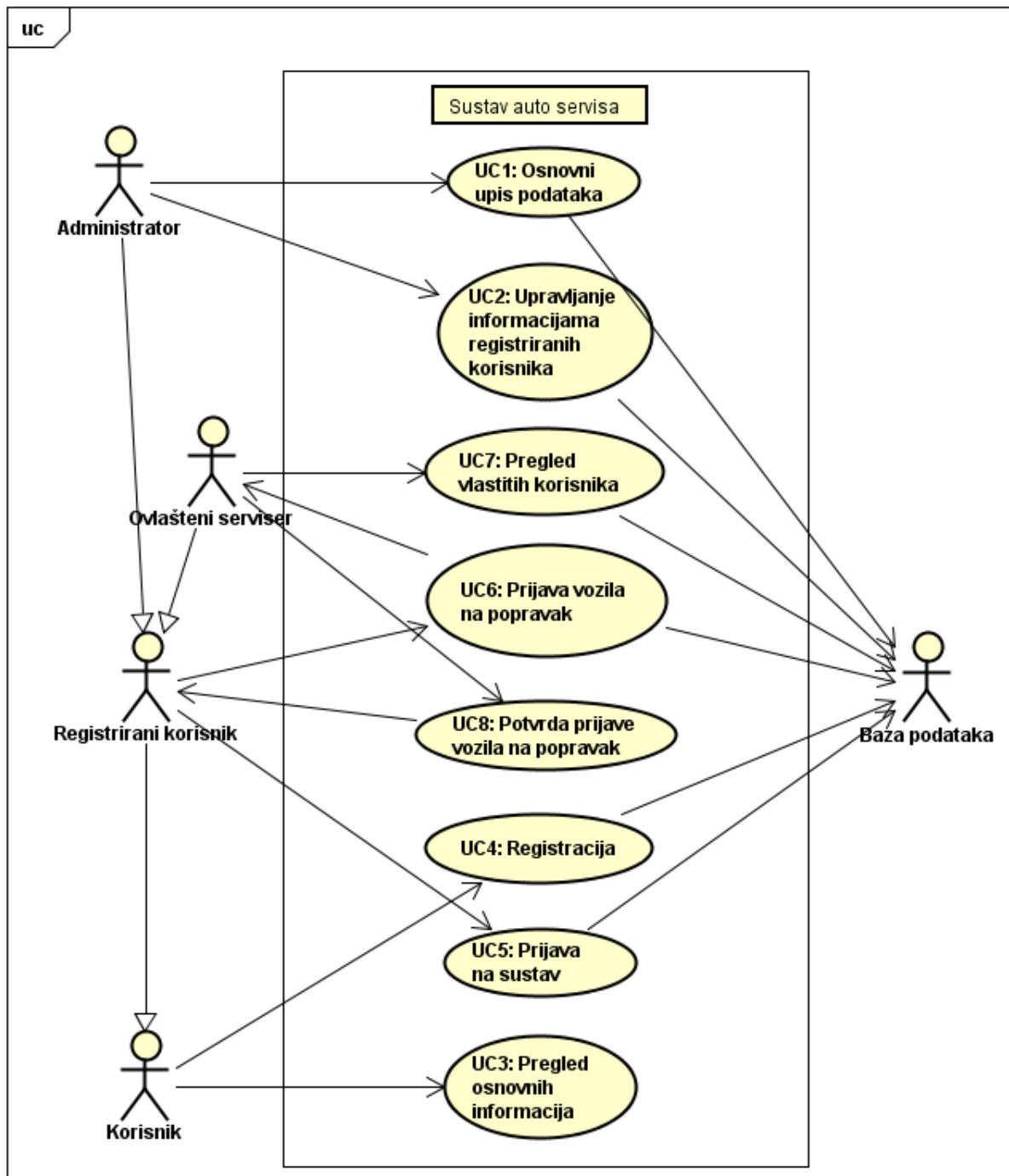
- Naziv: Potvrda prijave vozila na popravak
- Glavni sudionici: Ovlašteni serviser
- Cilj: Potvrda prijave vozila na popravak
- Preduvjeti: Prijava na sustav
- Željeni scenarij: Ovlašteni serviser potvrđuje prijavu vozila
- Opis mogućih odstupanja: -

UC8.1

- Naziv: Ispis obrasca
- Glavni sudionici: Ovlašteni serviser
- Cilj: Potvrda prijave vozila na popravak ispisom obrasca
- Sudionici: Registrirani korisnik
- Preduvjeti: Prijava na sustav
- Željeni scenarij: Nakon završetka ovlašteni servis ispisuje obrazac koji daje na potpis korisniku, a koji korisnik zadržava za sebe kao potvrdu predaje vozila.
- Opis mogućih odstupanja: -

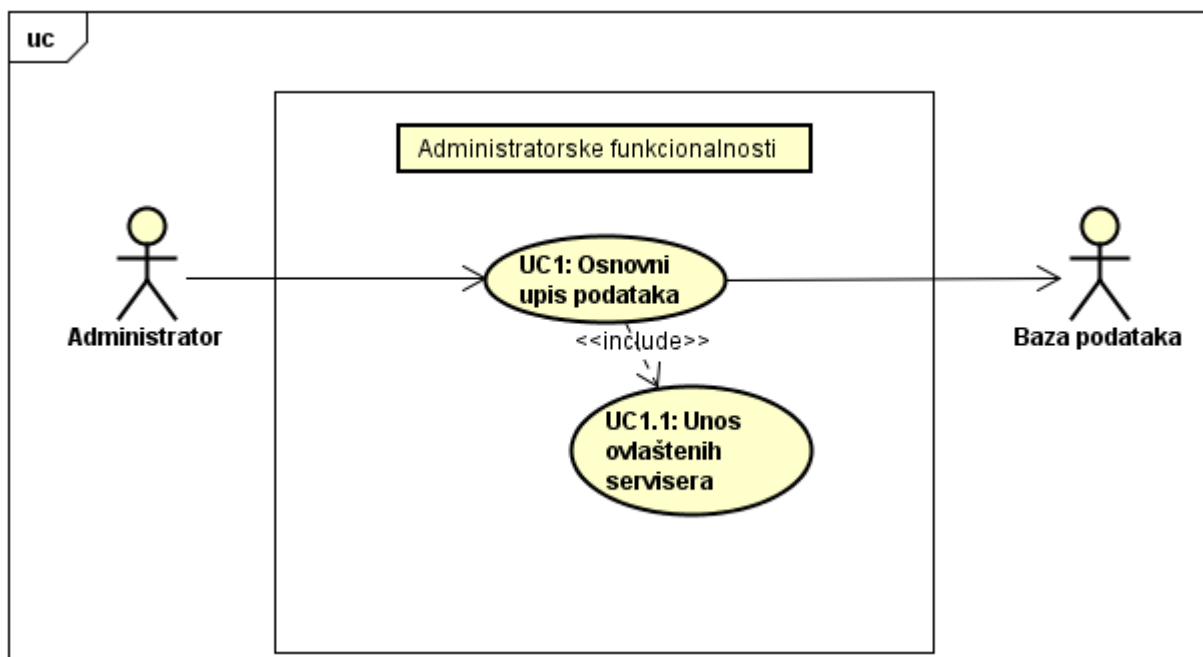
4.4. Dijagrami obrazaca uporabe

4.4.1. Dijagram obrazaca uporabe za cijeli auto servis



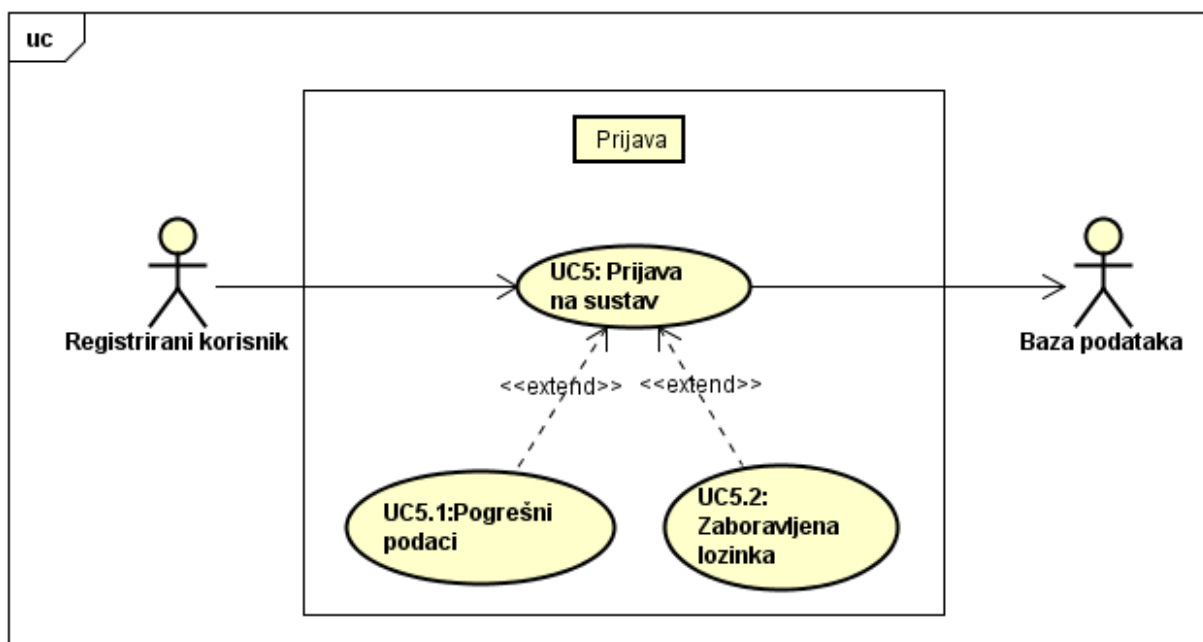
Slika 1. Dijagram obrazaca uporabe za cijeli auto servis

4.4.2. Dijagram obrazaca uporabe za administratora



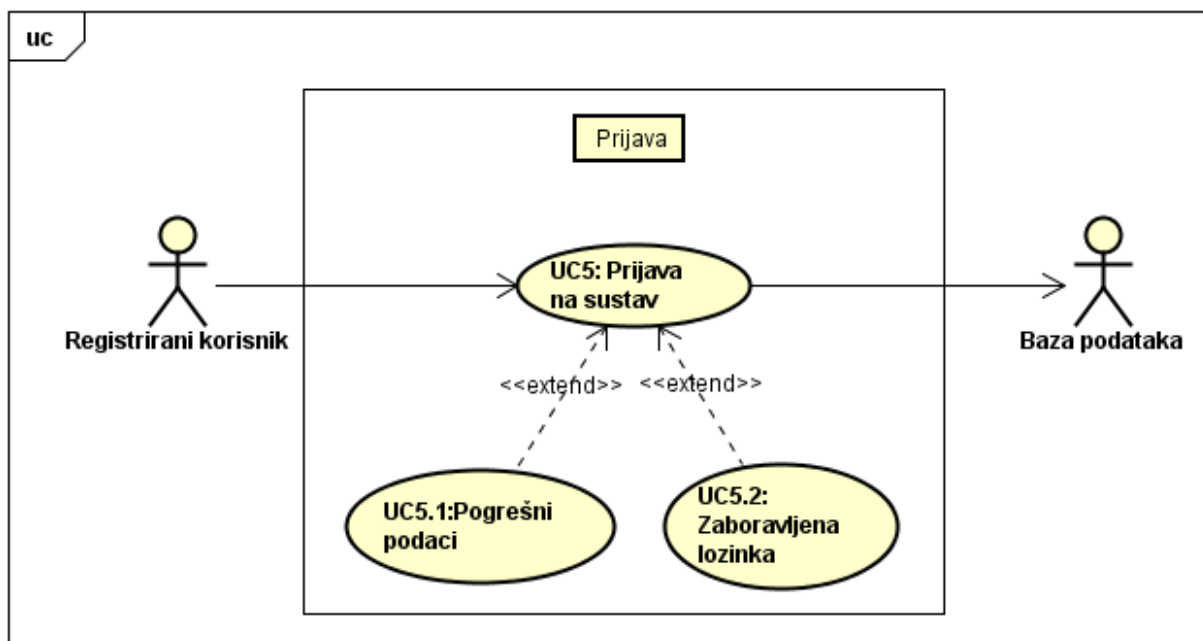
Slika 2. Dijagram obrazaca uporabe za administratora

4.4.3. Dijagram obrazaca uporabe za korisnika



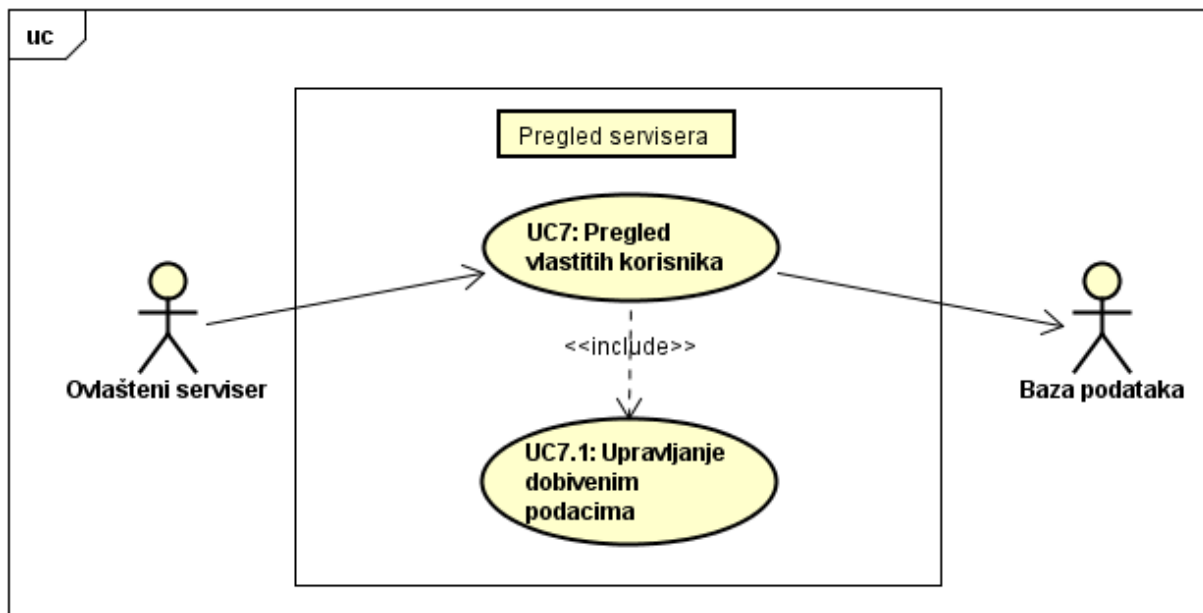
Slika 3. Dijagram obrazaca uporabe za korisnika

4.4.4. Dijagram obrazaca uporabe za prijavu



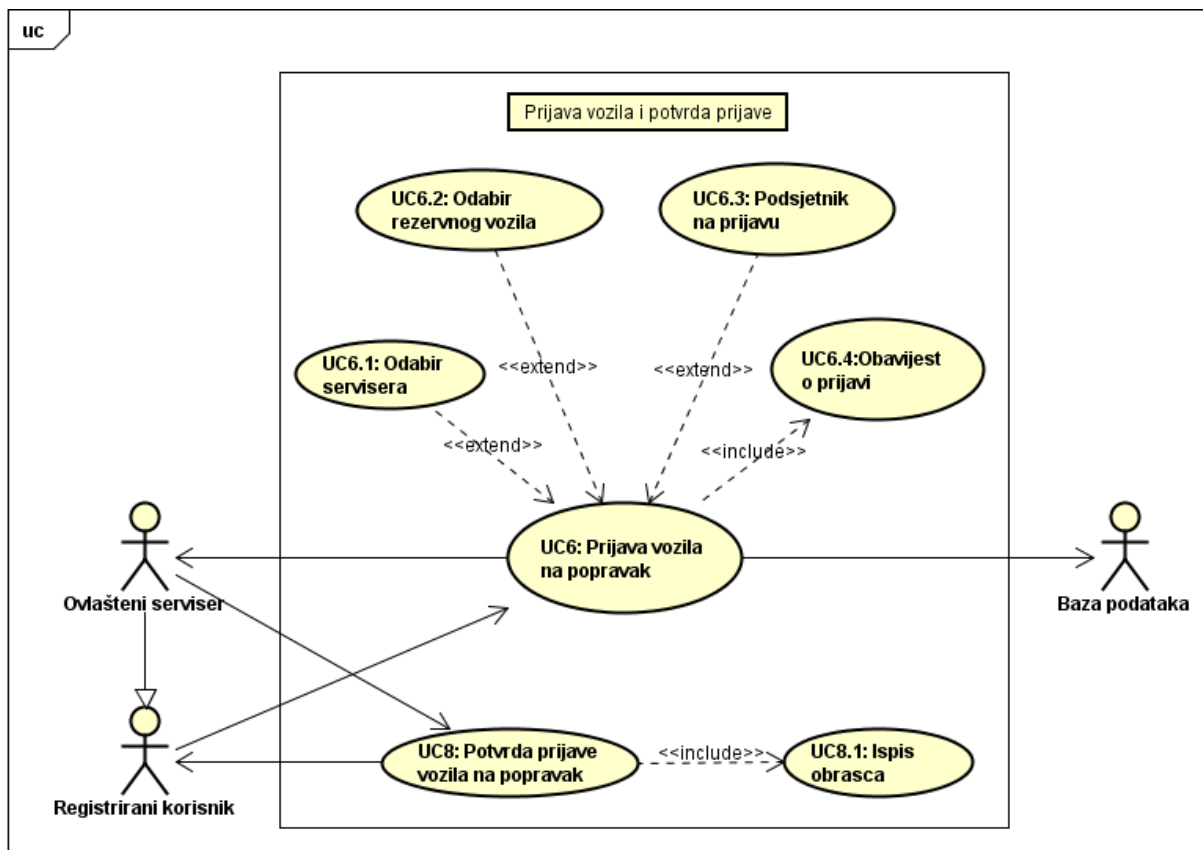
Slika 4. Dijagram obrazaca uporabe za prijavu

4.4.5. Dijagram obrazaca uporabe za serviseru



Slika 5. Dijagram obrazaca uporabe za serviseru

4.4.6. Dijagram obrazaca uporabe za obradu prijave vozila

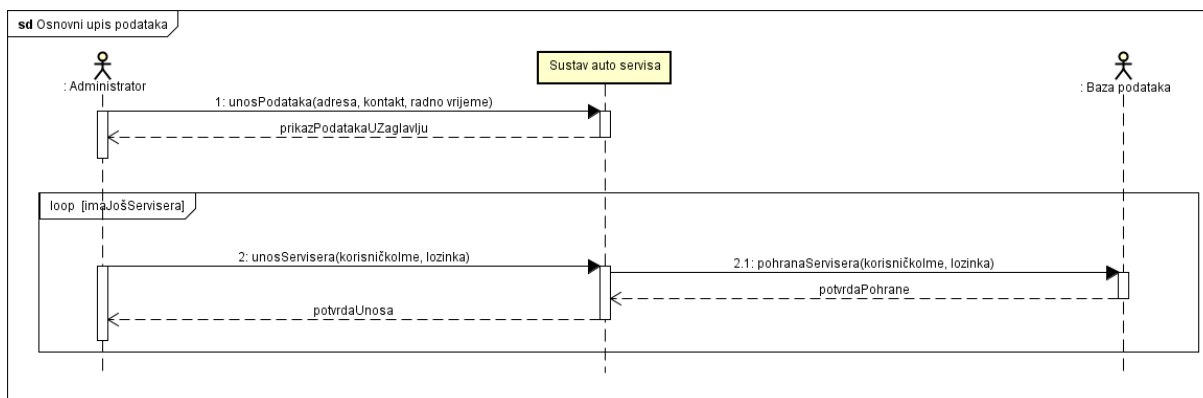


Slika 6. Dijagram obrazaca uporabe za obradu prijave vozila

4.5. Sekvencijski dijagrami

4.5.1. Osnovni upis podataka

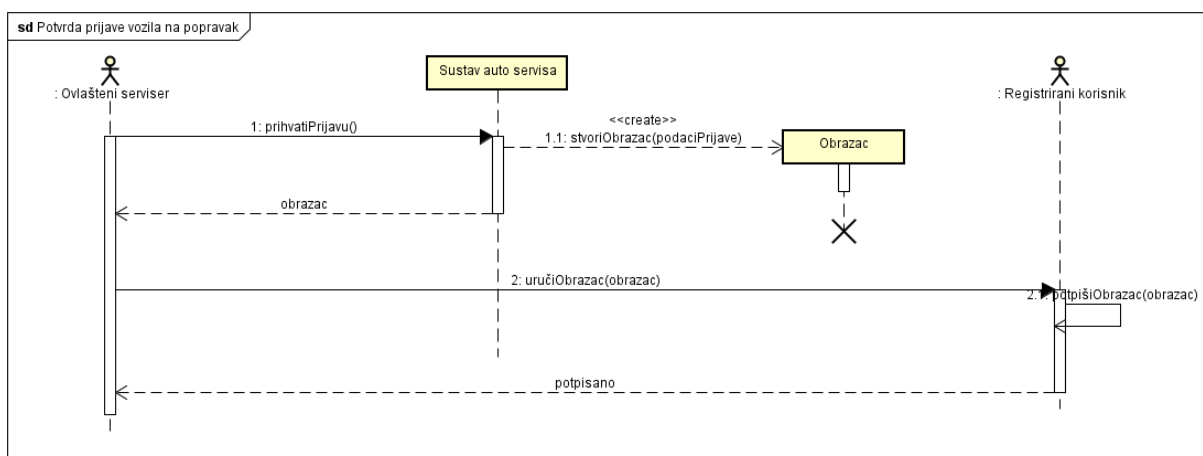
Administrator sustava upisuje podatke o servisu i kontakt podatke koje čine adresa, kontakt i radno vrijeme koje sustav prikazuje u zaglavlju. Administrator upisuje i podatke o ovlaštenim serviserima, od kojih svaki mora imati svoje korisničko ime i lozinku te se ti podaci pohranjuju u bazu podataka i unos se potvrđuje.



Slika 7. Sekvencijski dijagram - Osnovni upis podataka

4.5.2. Potvrda prijave vozila na popravak

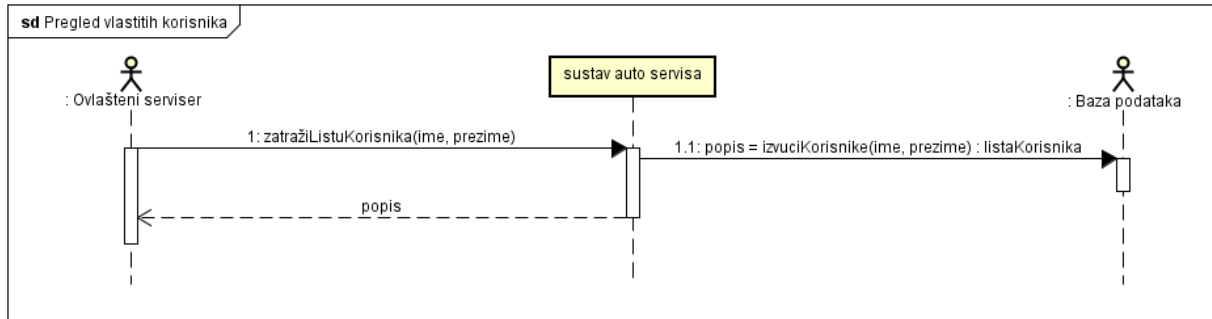
Ovlašteni serviser prihvaća prijavu vozila na popravak, sustav stvara obrazac koji vraća ovlaštenom serviseru te ga ovlašteni serviser uručuje korisniku na potpis i vraća ga.



Slika 8. Sekvencijski dijagram - Potvrda prijave vozila na popravak

4.5.3. Pregled vlastitih korisnika

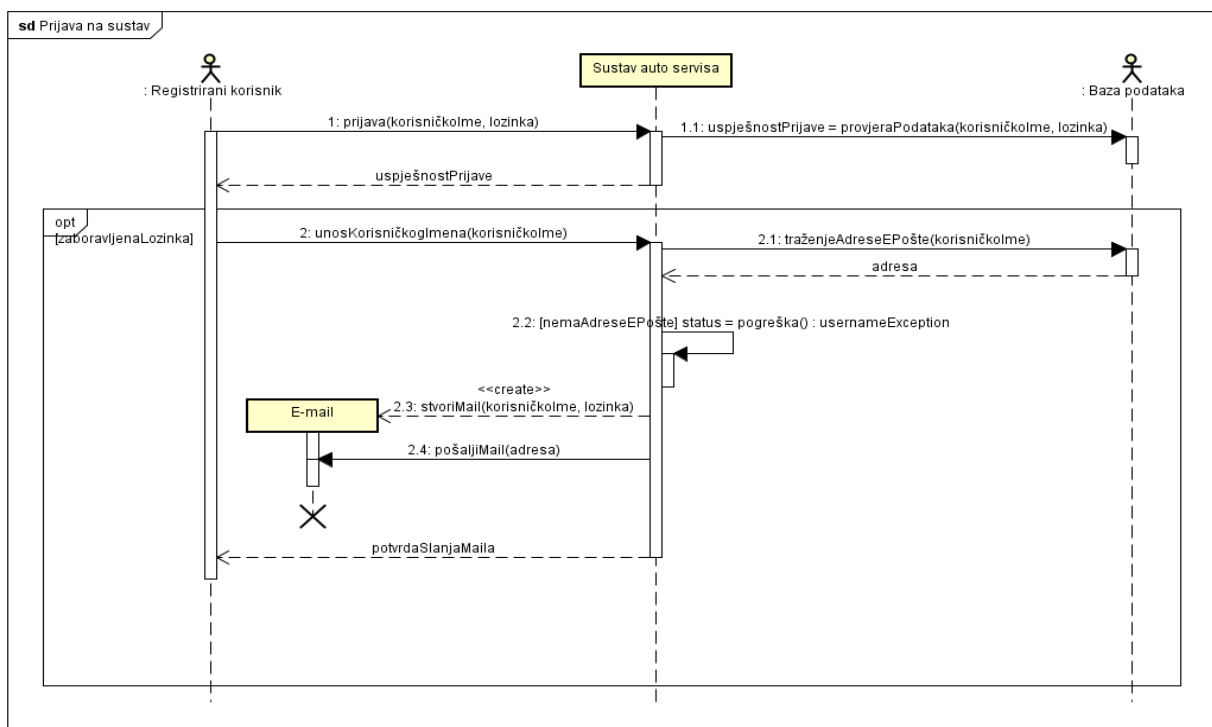
Ovlašteni serviser može u sustavu pregledavati vlastite prijavljene korisnike za taj radni dan, sustav te podatke uzima iz baze podataka i daje ih ovlaštenom serviseru.



Slika 9. Sekvencijski dijagram - Pregled vlastitih korisnika

4.5.4. Prijava na sustav

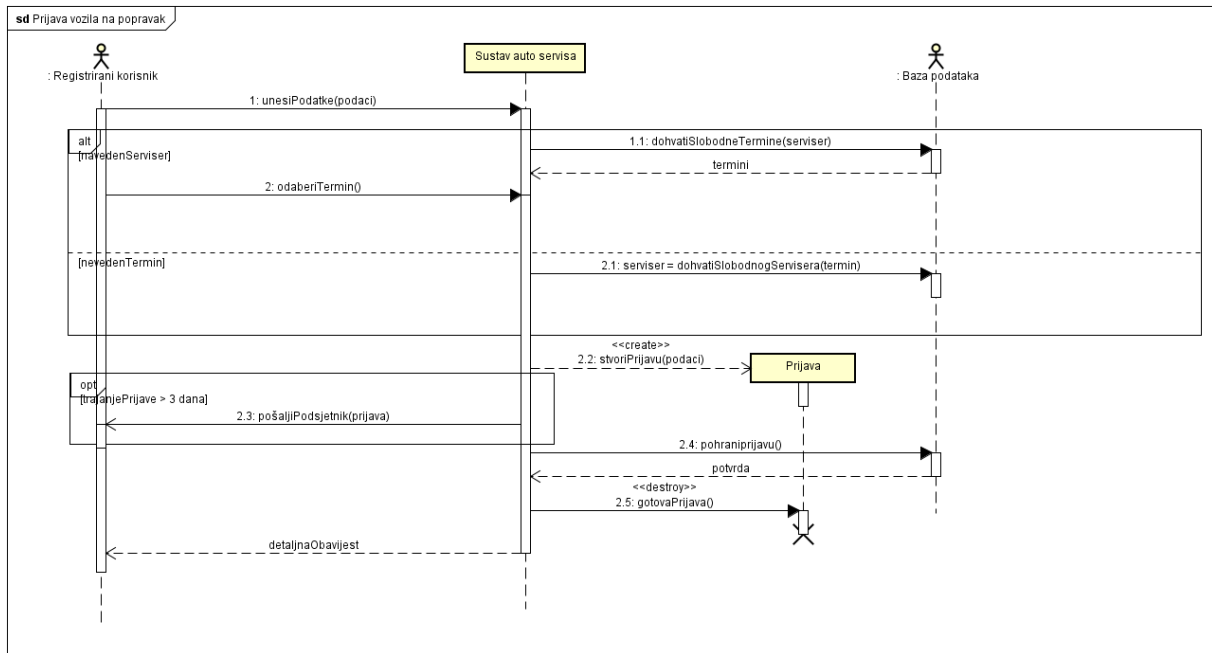
Registrirani korisnik se na sustav prijavljuje korisničkim imenom i lozinkom, sustav provjerava unesene podatke te ako su ispravni uspješno prijavljuje korisnika. Ako korisnik zaboravi lozinku za prijavu može unijeti samo svoje korisničko ime, sustav će preko korisničkog imena pokušati dohvatiti adresu elektroničke pošte i javit će pogrešku ako to nije mogao uspješno obaviti. U slučaju uspješnog dohvaćanja adrese elektroničke pošte sustav će stvoriti elektroničku poruku i poslati ju na dohvaćenu adresu, te potvrditi njeno slanje.



Slika 10. Sekvencijski dijagram - Prijava na sustav

4.5.5. Prijava vozila na popravak

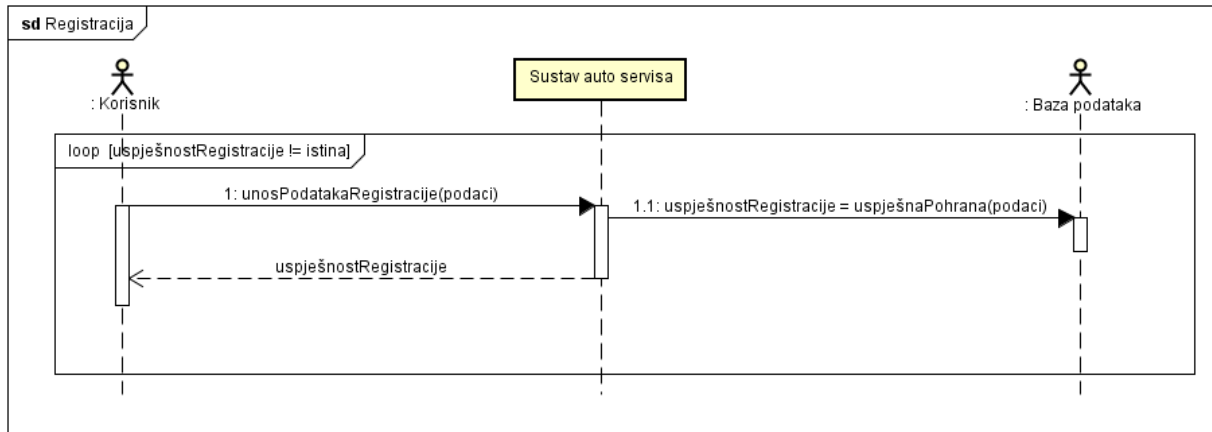
Registrirani korisnik pokreće prijavu vozila na popravak, unosi podatke o vozilu. Registriranom korisniku sustav daje neobaveznu opciju izbora ovlaštenog serviser, ako on to ne učini serviser mu se dodjeljuje slobodnog, u ovisnosti o vremenu dolaska na servis. Ukoliko korisnik prijavljuje vozilo za popravak u periodu dužem od 3 dana, sustav ga automatski podsjeća na prijavu zadnji dan prije dana koji je odabrao kao željeni za dolazak na servis. Nakon toga sustav stvara prijavu, pohranjuje ju u bazu podataka te dobiva potvrdu o spremanju. Objekt prijava se uništava, registriranom korisniku se na šalje detaljna obavijest o prijavi.



Slika 11. Sekvencijski dijagram - Prijava vozila na popravak

4.5.6. Registracija

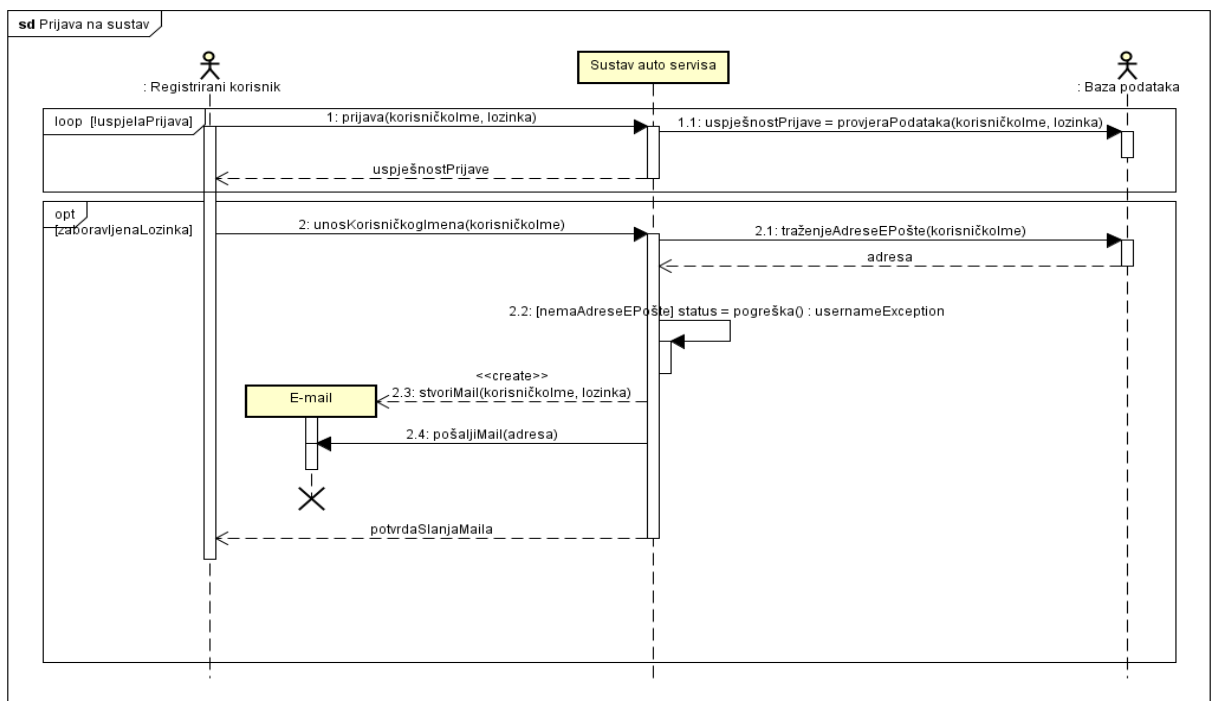
Korisnik unosi podatke potrebne za registraciju, ako sustav uspješno pohrani te podatke u bazu podataka registracija je uspješna, u suprotnom je neuspješna. Korisnik unosi podatke dok uspješno ne obavi registraciju.



Slika 12. Sekvencijski dijagram – Registracija

4.5.7. Upravljanje informacijama registriranih korisnika

Administrator preko sustava može pristupiti podacima o korisnicima pohranjenim u bazi podataka, te podatke može brisati ili mijenjati te ih pohraniti natrag u bazu podataka. Sustav dobiva potvrdu pohrane a Administrator potvrdu izmjene.



Slika 13. Sekvencijski dijagram – Upravljanje informacijama registriranih korisnika

5. Ostali zahtjevi

Od ostalih zahtjeva valja spomenuti vremenski intervali za prijavu na servis u koracima od 20 minuta. Sljedeći važan i često podrazumijevan zahtjev je odgovor web aplikacije na bilo kakav upit u realnom vremenu. Odgovor uključuje pretragu po bazi podataka, obradu informacija i prikaz web stranice na korisnikovom uređaju.

Aplikacija treba biti izrađena tako da se može koristiti na bilo kojem uređaju neovisno o tehnologiji. To znači da web stranice moraju biti kvalitetno prikazane kako na osobnim računalima tako i na mobilnim uređajima.

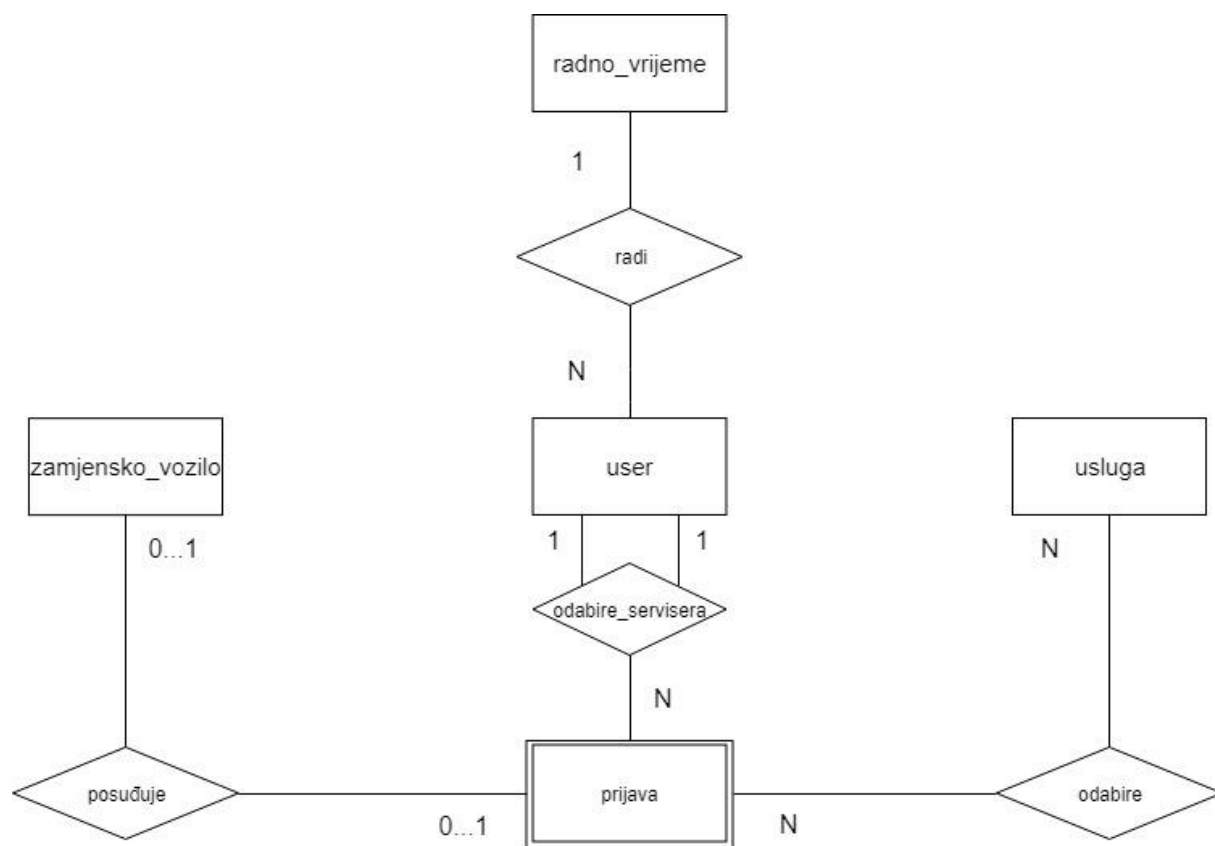
6. Arhitektura i dizajn sustava

6.1. Arhitekturni obrazac MVC

Izgrađena web aplikacija bazira se na jednom od najčešće korištenih arhitekturnih obrazaca, MVC (Model-View-Controller). MVC je arhitekturni obrazac koji razdvaja aplikaciju na tri odvojena dijela: model (eng. model), pogled (eng. view) i kontroler (eng. controller). Arhitekturni obrazac MVC je u današnje vrijeme gotovo standard u razvoju razne programske podrške. Glavna prednost MVC obrasca je jednostavno testiranje aplikacije te odvajanje zaduženja pojedinog dijela (eng. Separation of Concerns). Jednostavno testiranje aplikacije kreirane MVC obrascem znači kako su dijelovi aplikacije velikim dijelom nezavisni i odvojeni te ih je kao takve i jednostavnije testirati. Kao primjer možemo navesti test u kojem nas zanima da li naša aplikacija prikazuje popis ovlaštenih serviseri. U tome testu nas ne zanima od kud podaci dolaze, već nas samo zanima, da li će za dane podatke aplikacija prikazati krajnjem korisniku listu dohvaćenih korisnika kako je i specificirano samim zahtjevima na rad sustava. Odvajanje zaduženja (eng. Separation of Concerns) zapravo prikazuje da je svaki od pojedinih dijelova aplikacije zadužen za jednu ili više operacija, koje su između pojedinih dijelova nezavisne. Samim time možemo i objasniti za što je zadužen pojedini dio samog arhitekturnog obrasca MVC:

- Model (eng. model) – model upravlja ponašanjem i podacima same aplikacije. U njega su ugrađena poslovna pravila te logika (business logic). Model je najčešće reprezentacija samih relacija baze podataka.
- Pogled (eng. view) – pogled možemo promatrati kao samo korisničko sučelje programa. Glavna zadaća pogleda je korisniku prikazati sve relevantne podatke kroz strukturiran i jednostavan format. Kroz pogled se također vrši sva interakcija korisnika sa samim sustavom, no bilo bi pogrešno reći kako je pogled odgovoran za interakciju korisnika sa sustavom, s obzirom da sav posao obavlja sam kontroler, dok pogled samo omogućava korisniku korištenje sustava bez ikakve tehničke naobrazbe.
- Kontroler (eng. controller) – glavna zadaća kontrolera odgovaranje na akcije koje inicira korisnik preko pogleda te prijenos podataka modela u pogled i obrnuto. Kontroler je također zadužen i za svu drugu interakciju između ovih dvije komponente.

6.2. ER model baze podataka



Slika 14. ER model baze podataka

6.2.1. Opisi entiteta ER modela baze podataka

user – osoba koja koristi usluge sustava za praćenje rada auto servisa, korisnik može biti registrirani korisnik, serviser ili administrator

user_id – jedinstveni identifikacijski broj korisnika

broj_telefona – broj telefona serviser

email – e-mail adresa korisnika

godina_proizvodnje – godina proizvodnje vozila

id_radnog_vremena – identifikacijski broj radnog vremena serviser

last_name – prezime određenog korisnika

name – ime određenog korisnika

password – šifrirana lozinka korisnika

original_password – originalna lozinka korisnika

registracija – registracijska oznaka vozila registriranog korisnika kojeg je isti doveo na popravak

role – vrsta korisnika, može biti registrirani korisnik, administrator ili serviser

tip_vozila – model vozila

zamjensko_vozilo – relacija koja sadrži fiksnih 10 vozila koje registrirani korisnik može posuditi u slučaju kvara svojeg vozila

reg_oznaka_vozila – registracijska oznaka zamjenskog vozila

godina_proizvodnje – godina proizvodnje vozila

id_korisnika – identifikacijski broj korisnika koji je rezervirao zamjensko vozilo

tip_vozila – model vozila

radno_vrijeme – relacija s popisom radnih vremena pojedinog serviseru u sustavu

id_radnog_vremena – identifikacijski broj radnog vremena serviseru

cet_kraj – kraj radnog vremena četvrtkom

cet_pocetak – početak radnog vremena četvrtkom

pet_kraj – kraj radnog vremena petkom

pet_pocetak – početak radnog vremena petkom

pon_kraj – kraj radnog vremena ponedjeljkom

pon_pocetak – početak radnog vremena ponedjeljkom

sri_kraj – kraj radnog vremena srijedom

sri_pocetak – početak radnog vremena srijedom

uto_kraj – kraj radnog vremena utorkom

uto_pocetak – početak radnog vremena utorkom

odabire_servisera – relacija veze koja opisuje odabir serviseru od strane registriranog korisnika

id_servisera – identifikacijski broj izabranog serviseru

user_id – identifikacijski broj korisnika koji je odabrao određenog serviseru

vrijeme_prijave – vrijeme kad je prijava podnesena

radi – relacija veze koja govori koje dane u tjednu radi određeni serviser

id_radnog_vremena – identifikacijski broj radnog vremena serviseru

user_id – identifikacijski broj korisnika koji je odabrao određenog serviseru

odabire – relacija veze koja opisuje odabir usluge korisnika u trenutku podnošenja prijave

id_korisnika – identifikacijski broj korisnika koji podnosi prijavu

id_usluge – identifikacijski broj usluge koju korisnik želi dobiti

vrijeme_prijave – vrijeme početka zahtjeva za uslugom

posuđuje – relacija veze koja opisuje posudbu zamjenskog vozila od strane registriranog korisnika

id_korisnika – identifikacijski broj korisnika koji podnosi zahtjev za posudbom

vrijeme_prijave – vrijeme podnošenja zahtjeva za posudbom zahtjeva za zamjenskim autom

reg_oznaka_vozila – registracijska oznaka rezerviranog zamjenskog vozila

usluga – relacija s popisom usluga koje sustav nudi

id_usluge – identifikacijski broj usluge

ime_usluge – naziv usluge

cijena_usluge – cijena pojedine usluge

prijava – relacija koja opisuje podnošenje zahtjeva korisnika na popravak auta

id_korisnika – identifikacijski broj korisnika koji podnosi prijavu

vrijeme_prijave – vrijeme kad je prijava podnesena

dodatni_zahitjevi – dodatni zahtjevi koje korisnik zahtijeva

id_servisera – identifikacijski broj izabranog servisera

reg_zamjensko – registracija zamjenskog vozila

vrijeme_dolaska – vrijeme dolaska na servis

vrijeme_zavrsetka – vrijeme zavrsetka popravka vozila

6.2.2. CREATE naredbe relacijskog modela baze podataka

```
CREATE TABLE user (  
    user_id INT NOT NULL,  
    broj_telefona VARCHAR(255) NOT NULL,  
    email VARCHAR(255) NOT NULL,  
    godina_proizvodnje VARCHAR(255),  
    id_radnog_vremena VARCHAR(255),  
    last_name VARCHAR(255) NOT NULL,  
    name VARCHAR(255) NOT NULL,  
    password VARCHAR(255) NOT NULL,  
    original_password VARCHAR(255) NOT NULL,  
    registracija VARCHAR(255),  
    role VARCHAR(255) NOT NULL,  
    tip_vozila VARCHAR(255),  
    PRIMARY KEY (user_id),  
    FOREIGN KEY (id_radnog_vremena) REFERENCES radno_vrijeme (id_radnog_vremena));
```

```
CREATE TABLE zamjensko_vozilo (  
    reg_oznaka_vozila VARCHAR(20) NOT NULL,  
    godina_proizvodnje INT NOT NULL,  
    id_korisnika VARCHAR(255),  
    tip_vozila VARCHAR(255) NOT NULL,  
    PRIMARY KEY (reg_oznaka_vozila),  
    FOREIGN KEY (id_korisnika) REFERENCES prijava (id_korisnika));
```

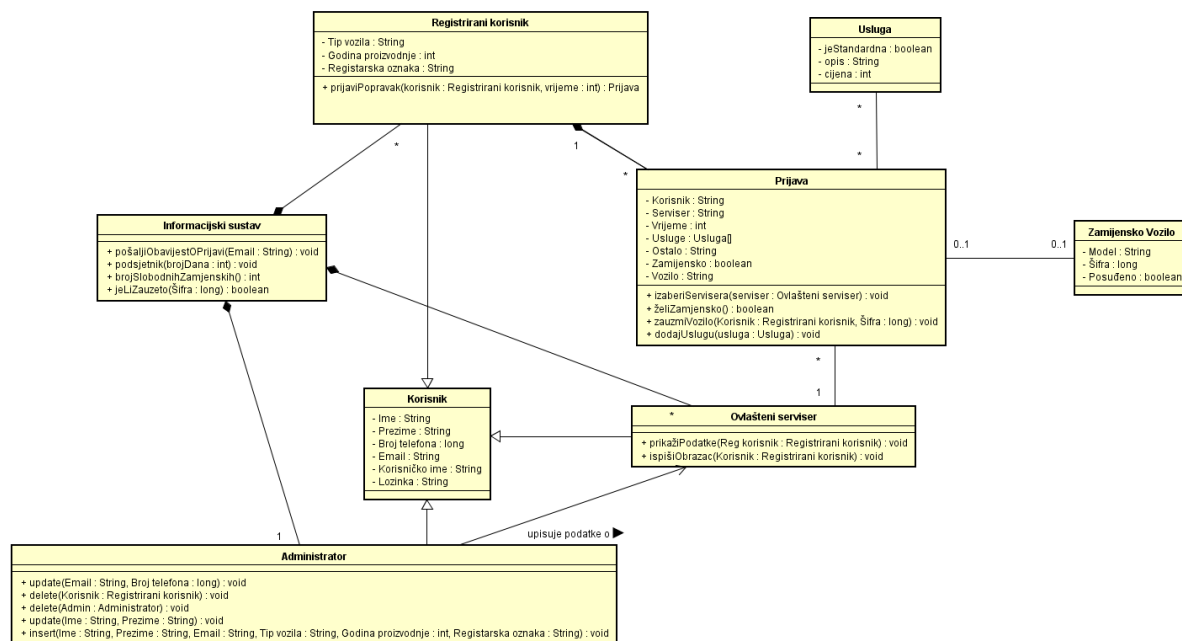
```
CREATE TABLE prijava (  
    id_korisnika INT NOT NULL,  
    vrijeme_prijave DATETIME NOT NULL,  
    dodatni_zahitjevi VARCHAR(255),  
    id_servisera INT NOT NULL,  
    reg_zamjensko VARCHAR(255),  
    vrijeme_dolaska DATE NOT NULL,  
    preuzeto BOOLEAN,  
    završeno BOOLEAN,  
    vrijeme_zavrsetka DATETIME,  
    PRIMARY KEY (id_korisnika, vrijeme_prijave),  
    FOREIGN KEY (id_korisnika) REFERENCES user (user_id),  
    FOREIGN KEY (id_servisera) REFERENCES user (user_id),  
    FOREIGN KEY (reg_zamjensko) REFERENCES zamjensko_vozilo(reg_oznaka_vozila));
```

```
CREATE TABLE usluga (  
    id_usluge INT NOT NULL,  
    ime_usluge VARCHAR(255) NOT NULL,  
    cijena_usluge INT NOT NULL,  
    PRIMARY KEY (id_usluge));
```

```
CREATE TABLE odabire (  
    id_usluge INT NOT NULL,  
    id_korisnika INT NOT NULL,  
    vrijeme_prijave DATETIME NOT NULL,  
    PRIMARY KEY (id_korisnika, vrijeme_prijave, id_usluge),  
    FOREIGN KEY (id_usluge) REFERENCES usluga (id_usluge),  
    FOREIGN KEY (id_korisnika) REFERENCES prijava (id_korisnika),  
    FOREIGN KEY (vrijeme_prijave) REFERENCES prijava (vrijeme_prijave));
```

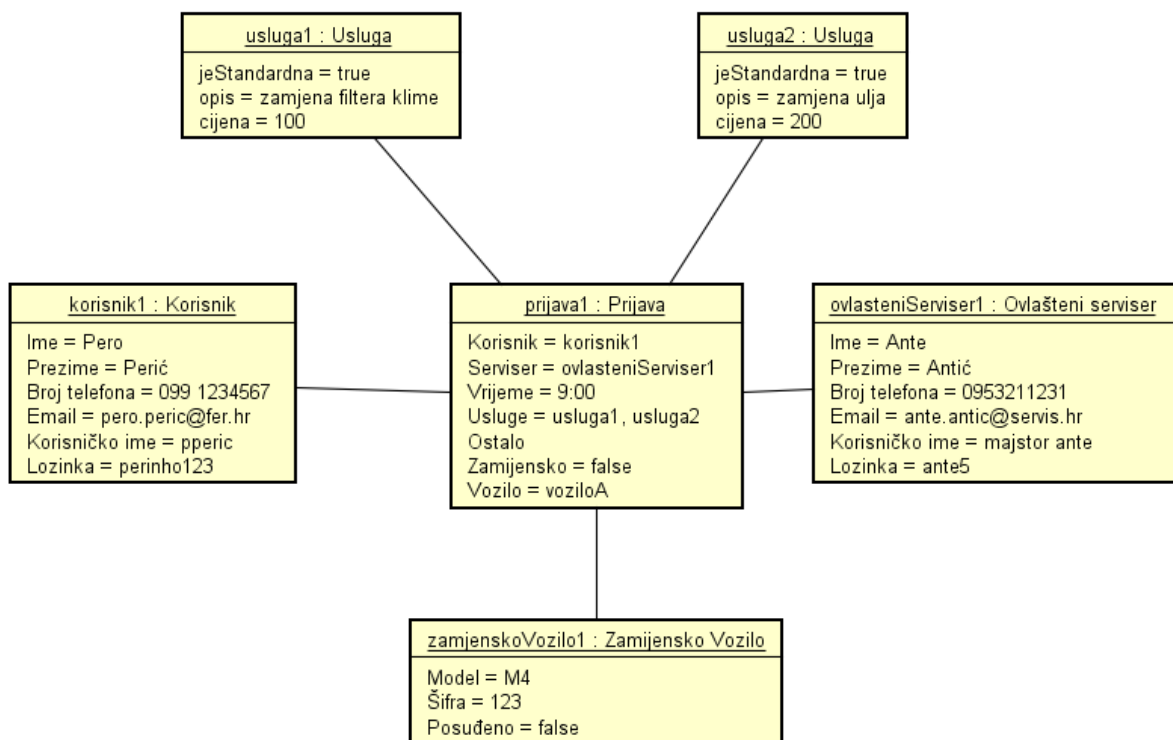
```
CREATE TABLE radno_vrijeme (  
    id_radnog_vremena INT NOT NULL,  
    cet_kraj TIME,  
    cet_pocetak TIME,  
    pet_kraj TIME,  
    pet_pocetak TIME,  
    pon_kraj TIME,  
    pon_pocetak TIME,  
    sri_kraj TIME,  
    sri_pocetak TIME,  
    uto_kraj TIME,  
    uto_pocetak TIME,  
    PRIMARY KEY (id_radnog_vremena));
```

6.3. Dijagram razreda



Slika 15 Dijagram razreda

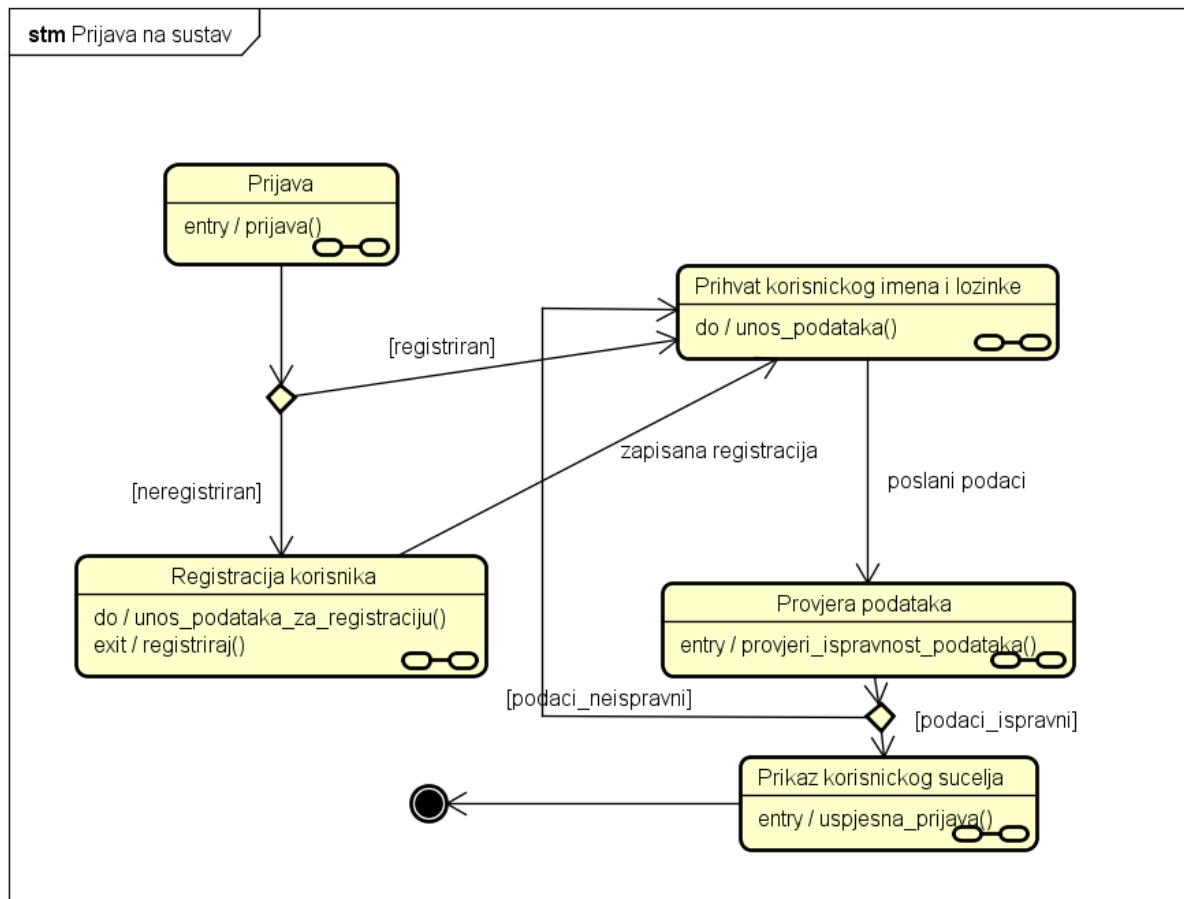
6.4. Dijagram objekata



Slika 16. Dijagram objekata

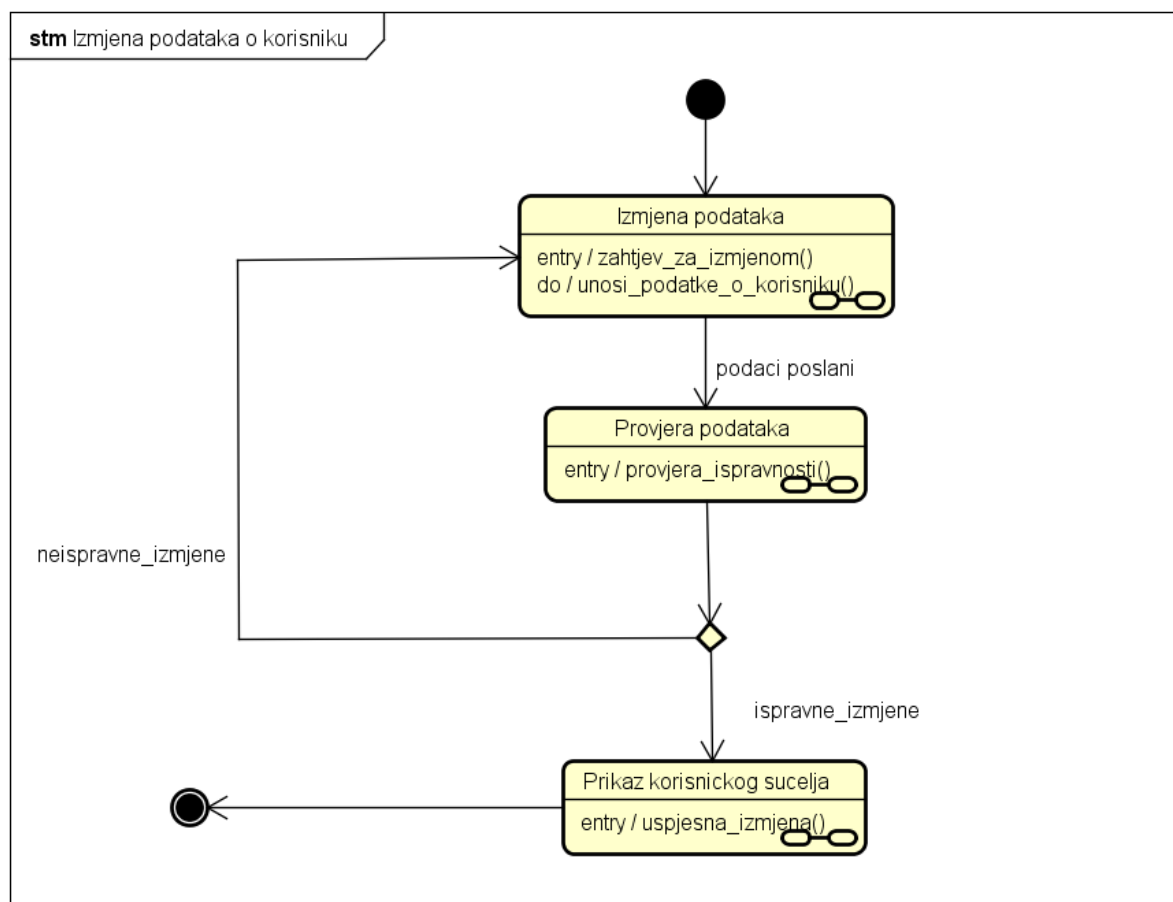
6.5. Ostali UML dijagrami

6.5.1. Dijagram stanja – prijava na sustav



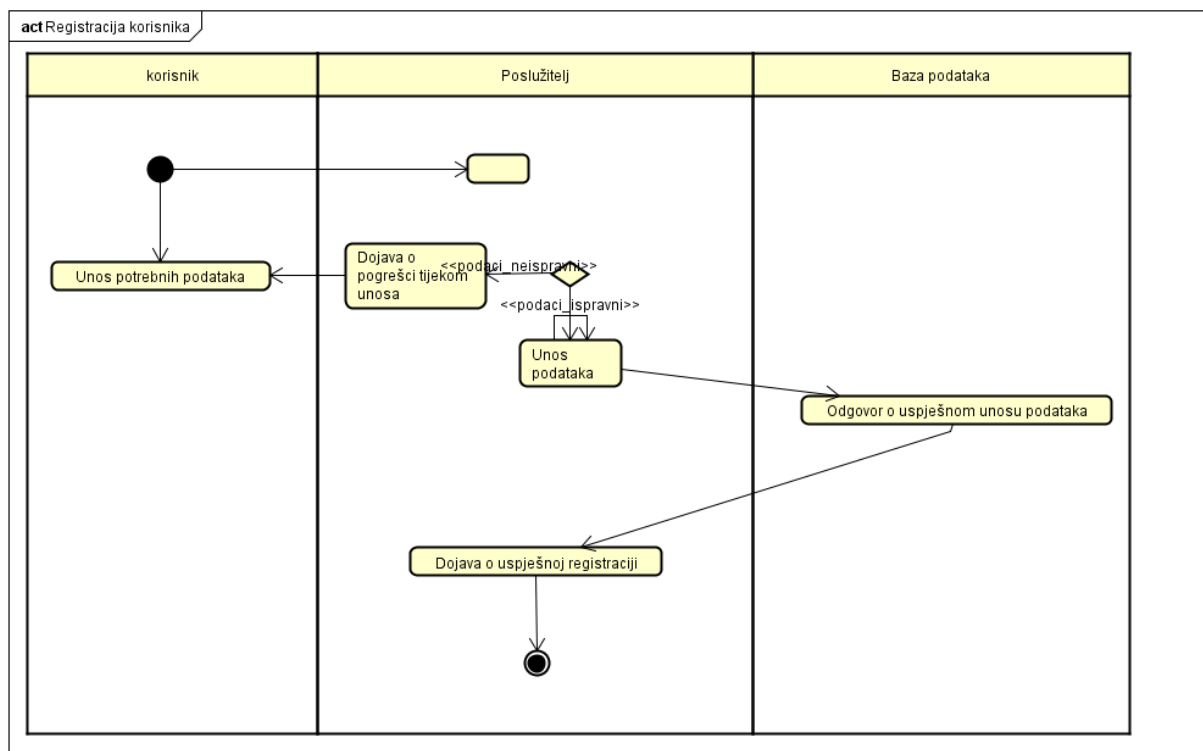
Slika 17. Dijagram stanja – prijava na sustav

6.5.2. Dijagram stanja – izmjena podataka o korisniku



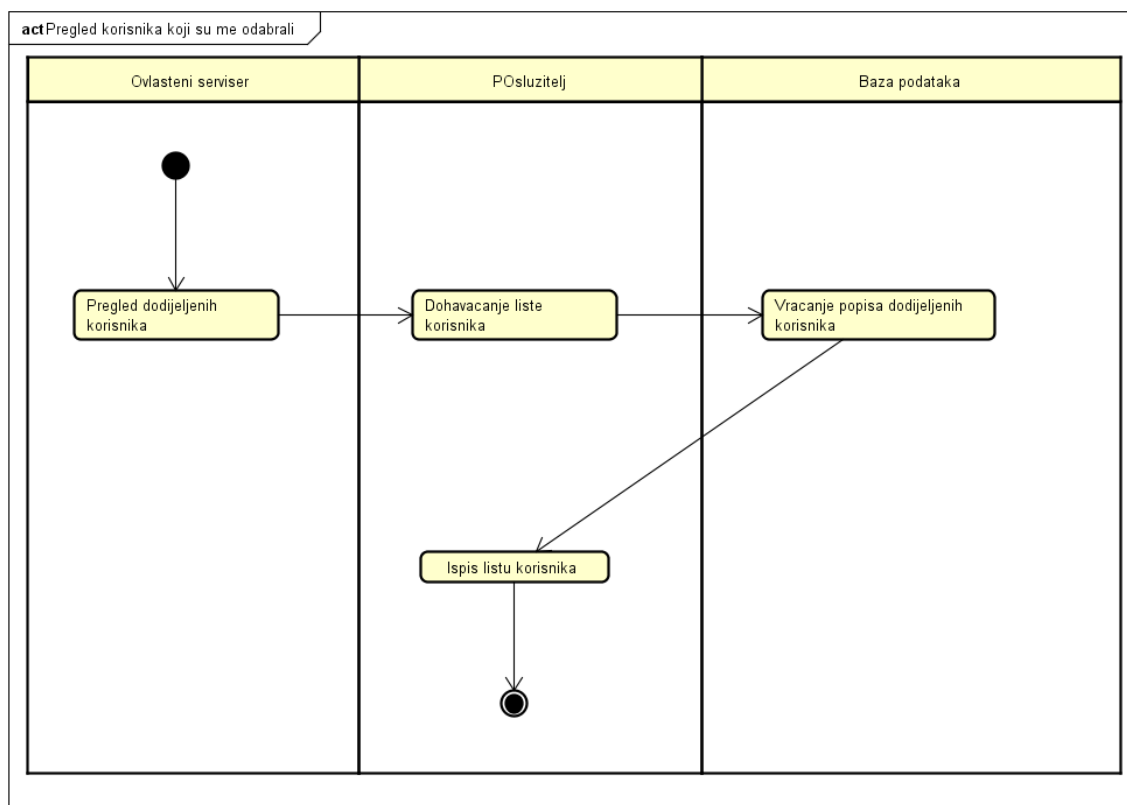
Slika 18. Dijagram stanja – izmjena podataka o korisniku

6.5.3. Dijagram aktivnosti – registracija korisnika



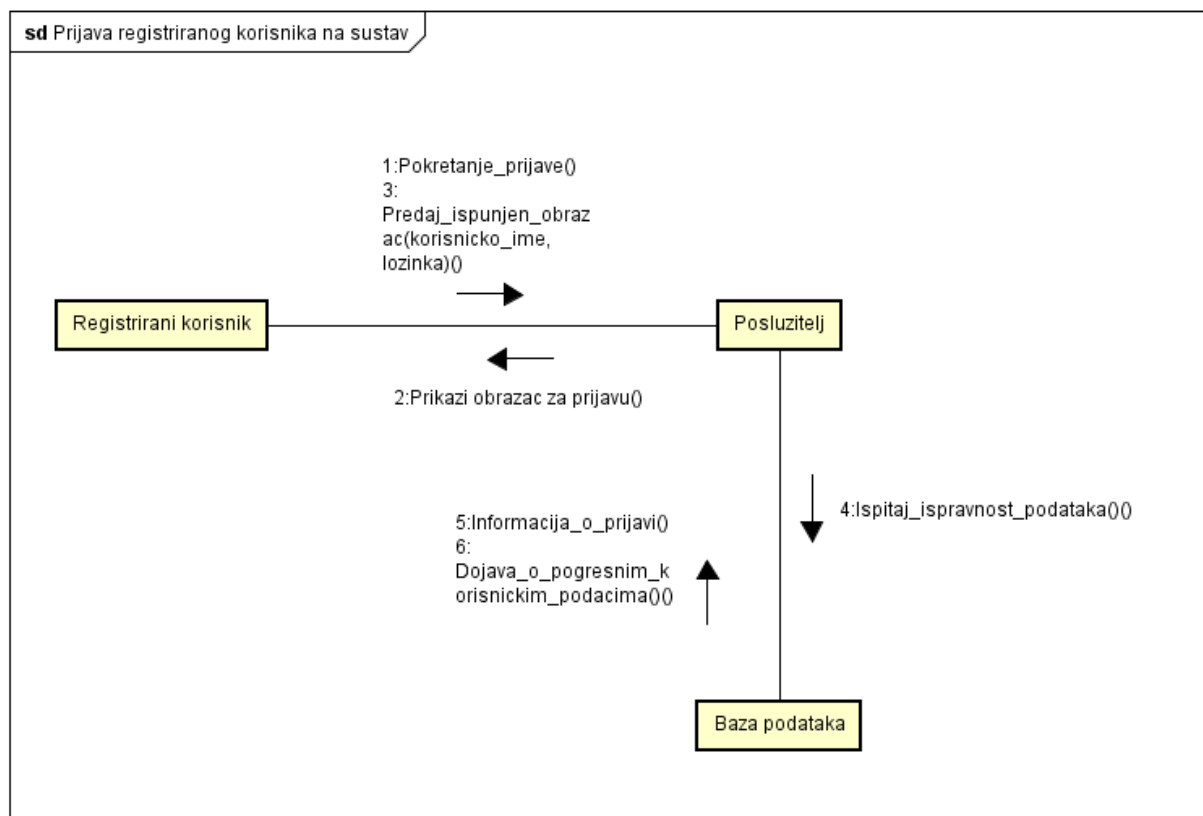
Slika 19. Dijagram aktivnosti – registracija korisnika

6.5.4. Dijagram aktivnosti – pregled korisnika



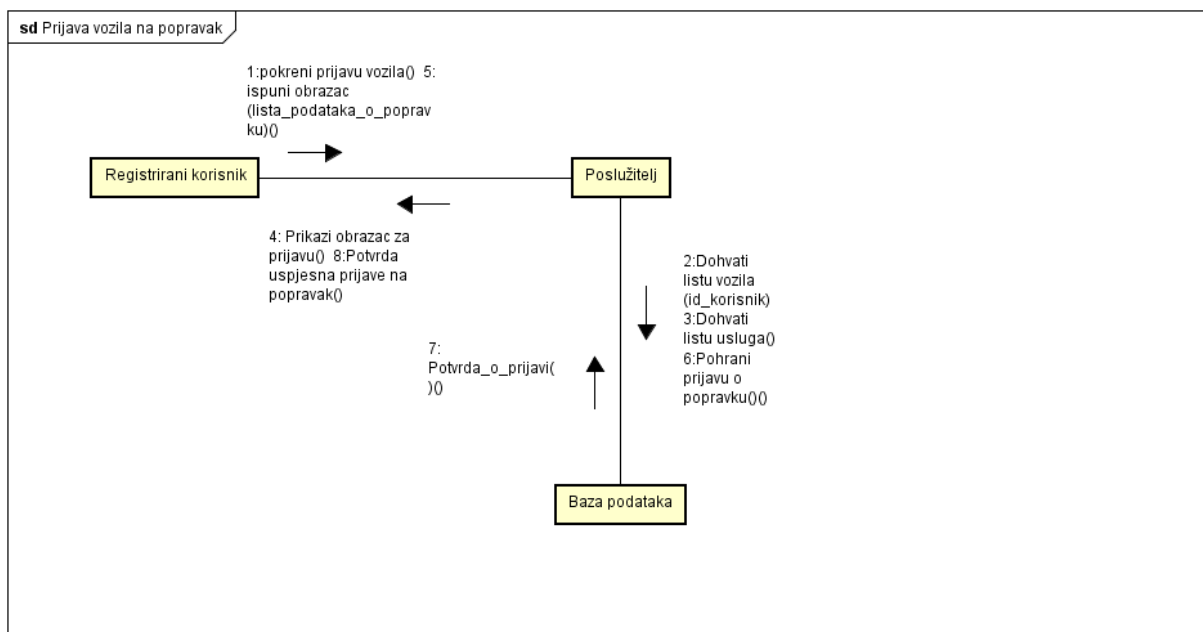
Slika 20. Dijagram aktivnosti – pregled korisnika

6.5.5. Komunikacijski dijagram – prijava registriranog korisnika na sustav



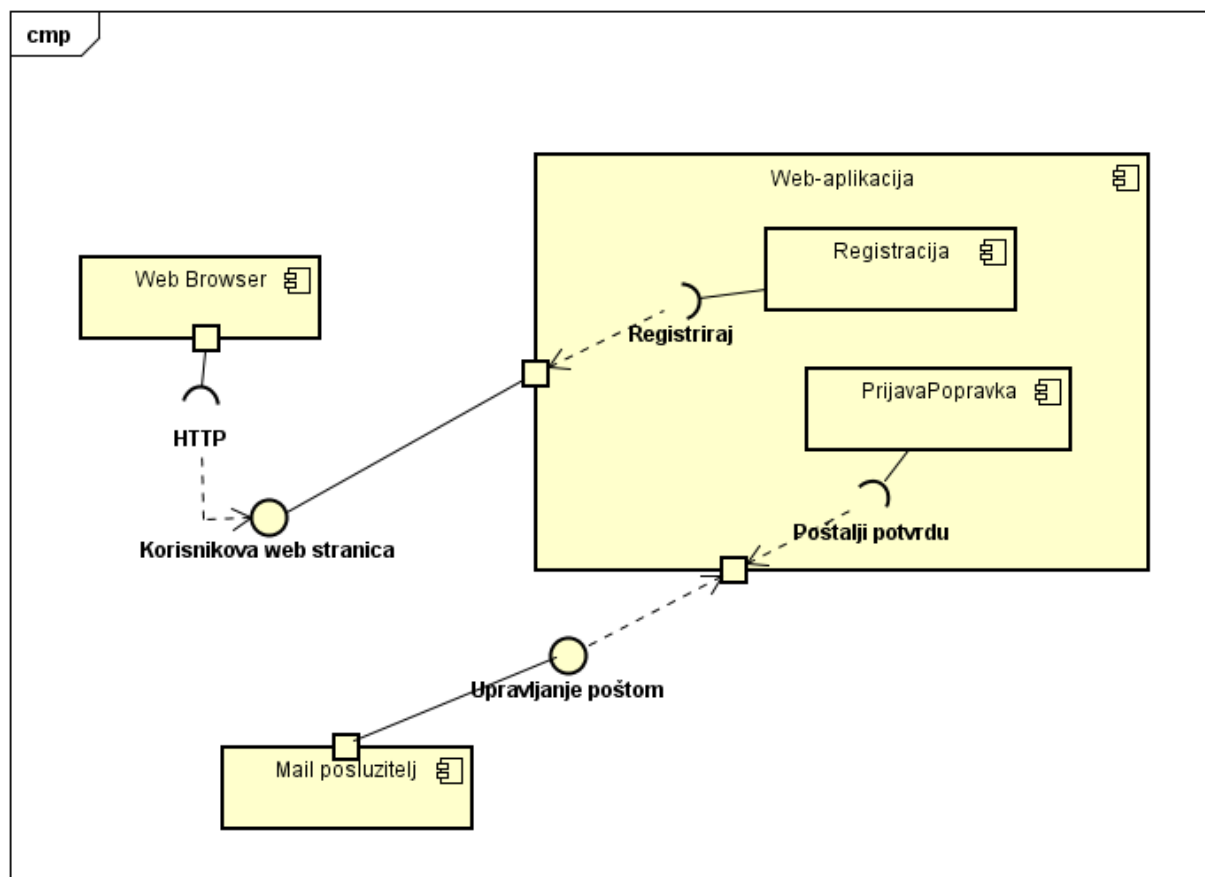
Slika 21. Komunikacijski dijagram – prijava registriranog korisnika na sustav

6.5.6. Komunikacijski dijagram – prijava vozila na popravak



Slika 22. Komunikacijski dijagram - prijava vozila na popravak

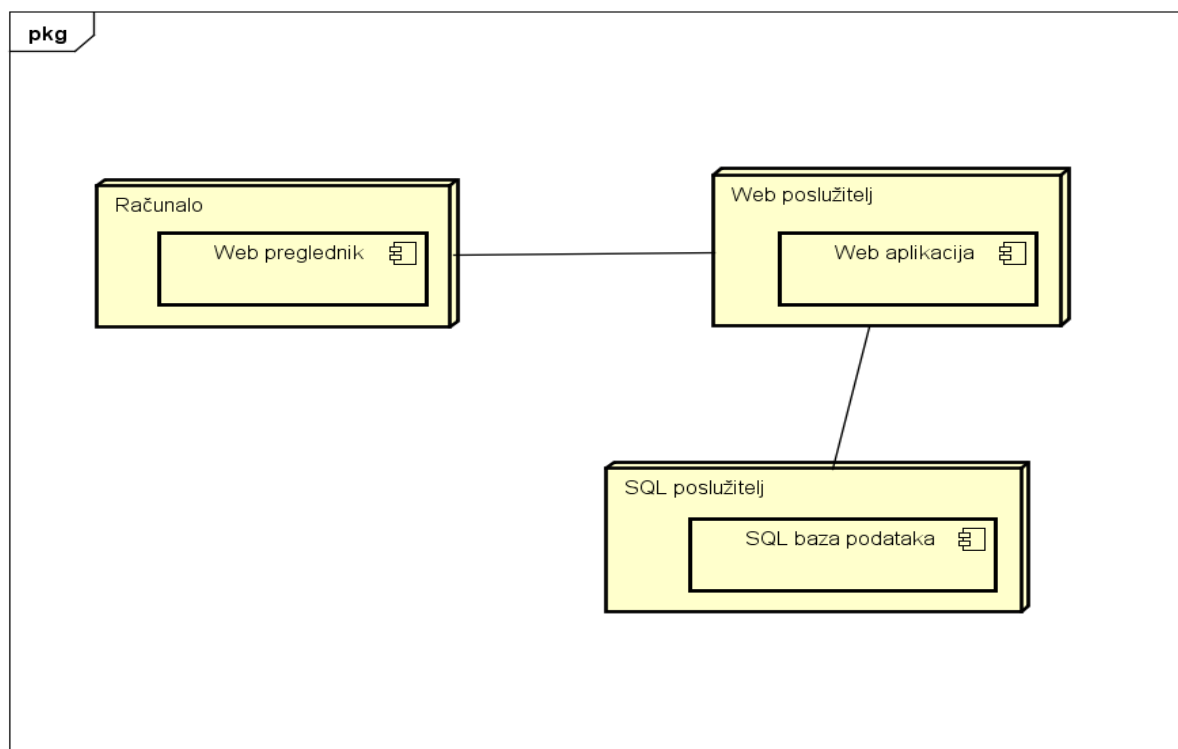
6.5.7. Dijagram komponenti



Slika 23. Dijagram komponenti

7. Implementacija i korisničko sučelje

7.1. Dijagram razmještaja



Slika 24. Dijagram razmještaja

7.2. Korišteni alati

Za izradu ove aplikacije korištena su 3 alata: Eclipse, Visual Studio Code i MySQL Workbench/Server

Eclipse je programska razvojna okolina (IDE), a može se koristiti za razvoj aplikacija u raznim programskim jezicima kao što su ADA, C, C++, C#, Cobol, Fortran, Javascript i druge. To je ujedno i najkorišteniji IDE za javu. Okolina posjeduje brojne nadogradnje (eng. plug-in) koje donose brojne funkcionalnosti i mogućnosti korisniku.

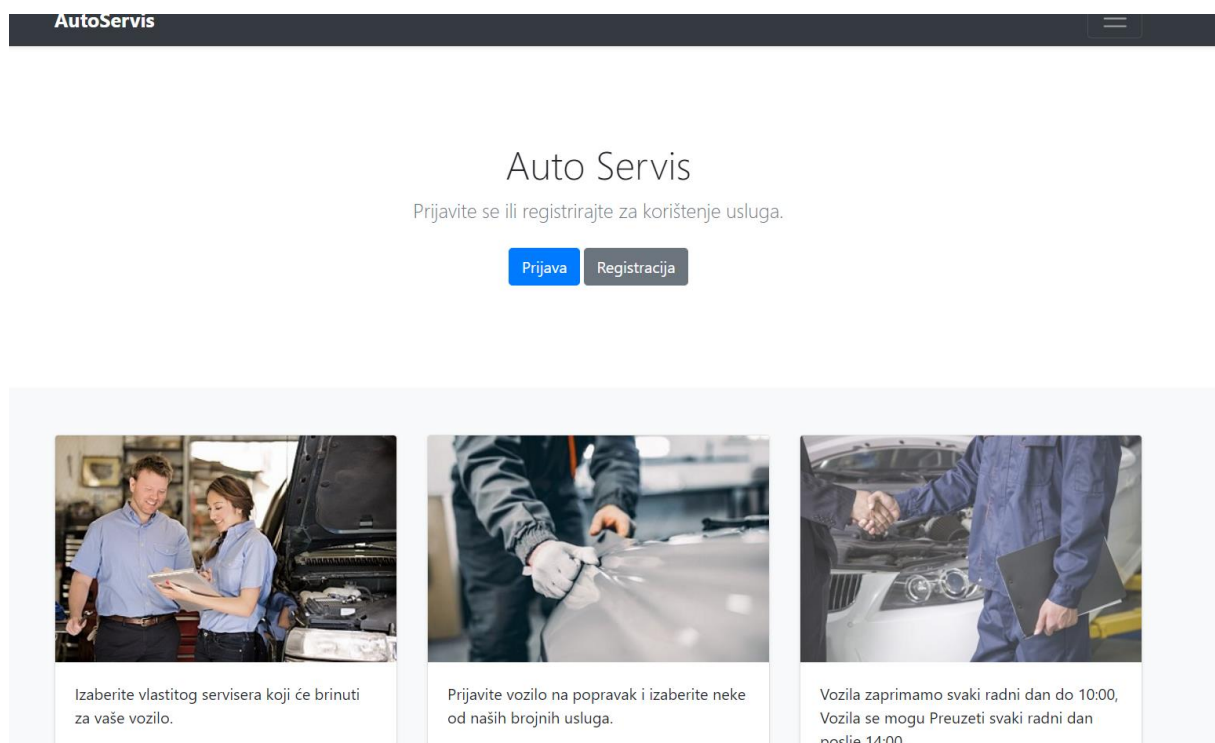
Visual Studio Code je uređivač izvornih kodova kojeg je napravila tvrtka Microsoft. Uređivač podržava i debugging te korištenje Git-a te je 2018. na Stack Overflow-u proglašen najkorištenijim uređivačem sa više od trećine korisnika. Podržava brojne programske jezike od kojih ima potpunu podršku za C#, F#, Java, Javascript, TypeScript, HTML i druge.

MySQL Workbench/Server su upravitelj i server na kojem živi lokalno baza podataka. Razvijeni su od tvrtke Microsoft sa ciljem pojednostavljenja i apstrakcije upravljanja bazama podataka. Primarna svrha im je spremanje i dohvaćanje podataka koristeći user-friendly sučelje.

7.3. Korisničke upute

Prije početka opisa korisničkih uputa važno je napomenuti da je ovo web aplikacija te kao takva za njezino korištenje nije potrebna nikakva instalacija programa, već je potrebno otići na sljedeći link: <https://nameless-meadow-70620.herokuapp.com/>

Početna stranica :



Slika 25. Početna stranica

Otvaranjem linka <https://nameless-meadow-70620.herokuapp.com/>, otvara se početna stranica. Klikom na gornje lijevi gumb „AutoServis“, korisnik se može s bilo koje stranice vratiti na početnu stranicu ili je jednostavno osvježiti ako je već na njoj. Klikom na gornje desni padajući izbornik otvaraju se kontakt informacije o autoservisu :



Slika 26. Podaci o auto servisu

U sredini početne stranice, nalaze gumbi „Prijava“ i Registracija.

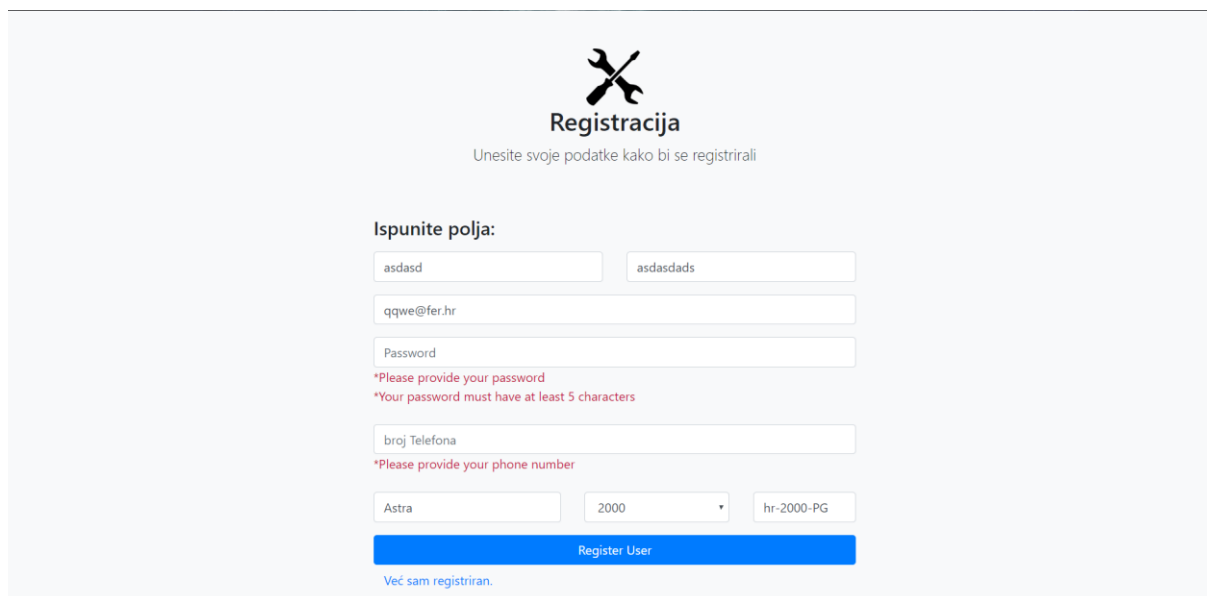
Klikom na gumb „Registracija“, korisnika će se preusmjeriti na stranicu za registraciju korisnika, a klikom na gumb „Prijava“ na stranicu za prijavu korisnika.

Stranica za registraciju korisnika :

Slika 27. Registracija korisnika

Stranica za registraciju korisnika omogućuje registraciju korisnika u sustav Autoservisa, nakon čega korisnik može koristiti usluge aplikacije. Korisnik je obavezan ispuniti sva ponuđena polja kako bi postigao uspješnu registraciju. Klikom na logo iznad natpisa „Registracija“ stranica nudi opciju vraćanja na početnu stranicu autoservisa.

Ukoliko se neka od polja za unos teksta ne ispune te se pritisne gumb „Register User“, aplikacija baca upozorenje kod onih polja koja nisu ispunjena :



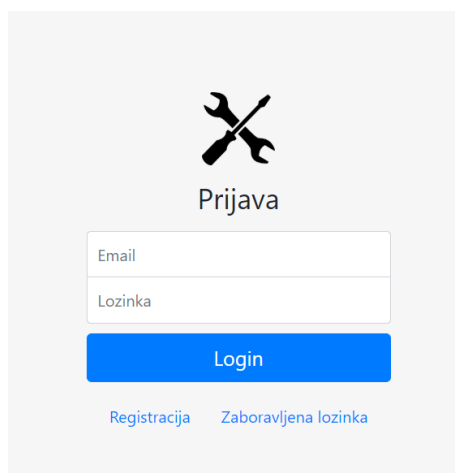
The screenshot shows a registration form titled "Registracija" with a wrench and screwdriver icon. Below the title is the instruction "Unesite svoje podatke kako bi se registrirali". The form is labeled "Ispunite polja:" and contains several input fields: two text boxes at the top (one with "asdasd", one with "asdasdads"), an email field with "qqwe@fer.hr", a password field with "Password", a phone number field with "broj Telefona", and a dropdown menu with "Astra" and "2000". There are also two small text boxes with "hr-2000-PG". A blue "Register User" button is at the bottom. Below the button is a link "Već sam registriran.". Error messages in red text are visible: "*Please provide your password" and "*Your password must have at least 5 characters" near the password field, and "*Please provide your phone number" near the phone number field.

Slika 28. Pogreške prilikom registracije

Klikom na gumb „Već sam registriran“, aplikacija preusmjerava korisnika na stranicu prijave.

Stranica za prijavu korisnika :

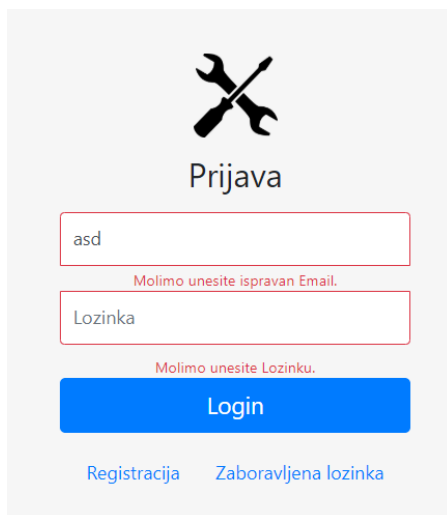
Stranica za prijavu korisnika služi za verifikaciju korisnika kao već registriranoga člana sustava.



The screenshot shows a login page titled "Prijava" with a wrench and screwdriver icon. Below the title is a form with two input fields: "Email" and "Lozinka". A blue "Login" button is below the fields. At the bottom, there are two links: "Registracija" and "Zaboravljena lozinka".

Slika 29. Prijava korisnika u sustav

Od korisnika se traži da upiše email i lozinku kojom je već registriran u sustav. Ukoliko pogriješi email ili lozinku ili jednostavno korisnik s tom kombinacijom emaila i lozinke ne postoji u bazi podataka, aplikacija izbacuje upozorenje za nepostojanje takve kombinacije :



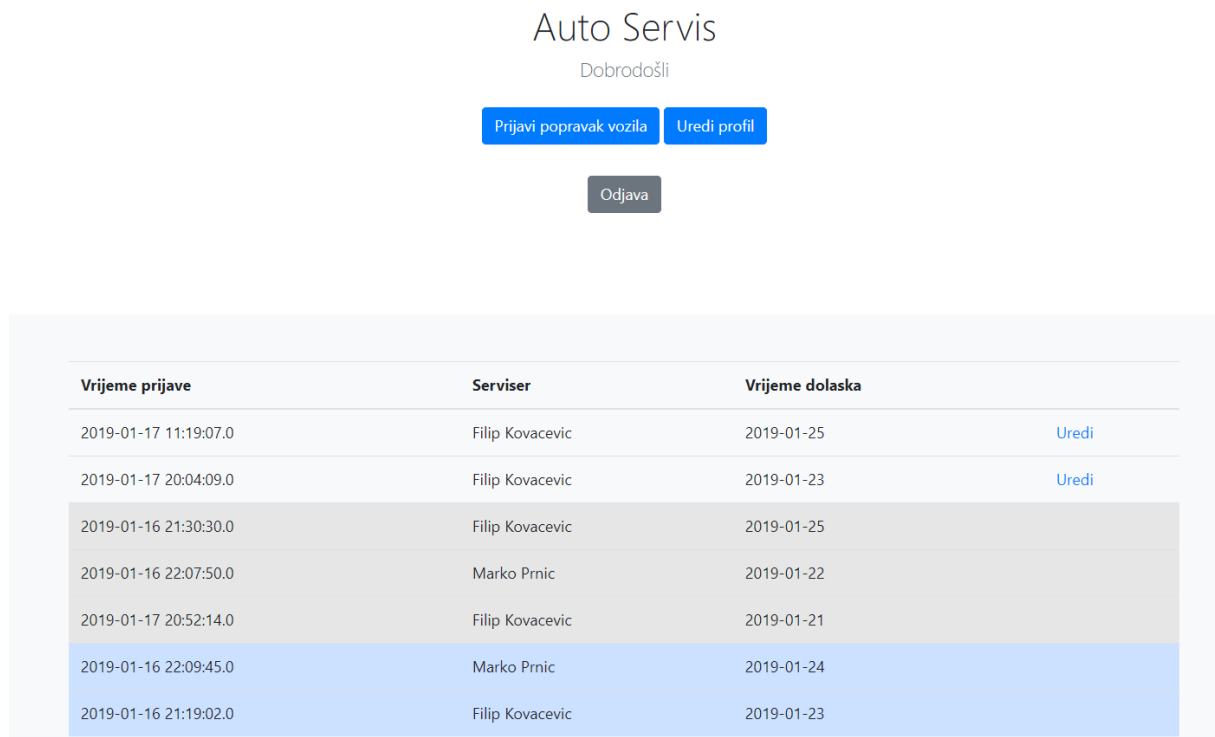
The screenshot shows a login interface with the title "Prijava" (Login) and a wrench and screwdriver icon. It contains two input fields: the first is labeled "Email" and contains the text "asd", with a red error message "Molimo unesite ispravan Email." below it; the second is labeled "Lozinka" (Password) and is empty, with a red error message "Molimo unesite Lozinku." below it. A blue "Login" button is positioned below the password field. At the bottom, there are two links: "Registracija" (Registration) and "Zaboravljena lozinka" (Forgot password).

Slika 30. Pogrešni podaci prijave

Ovisno o vrsti korisnika (je li on „običan“ korisnik, serviser ili administrator), usmjeruje ga se na odgovarajuću stranicu za njegovu ulogu.

Početna stranica za „običnog“ korisnika :

Ukoliko se korisnik aplikacije uspješno ulogirao te je u bazi podataka obilježen kao „običan“ korisnik, korisniku se prikazuje stranica s popisom svih njegovih zahtjeva za popravak :



Slika 31. Glavna stranica "običnog" korisnika

Tri boje predstavljaju tri stanja u kojem se određeni zahtjev može nalaziti.

Plava boja predstavlja zahtjeve koji su završeni, odnosno aute koji su uspješno popravljeni.

Tamno siva boja predstavlja zahtjeve koji su preuzeti od strane serviser, ali su još u procesu popravka.

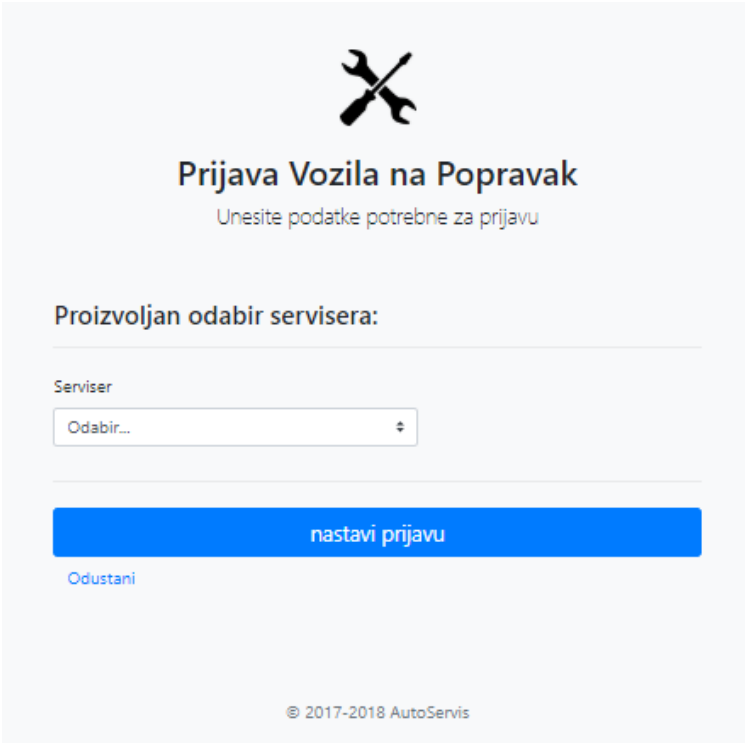
Siva boja predstavlja da zahtjev za popravkom još nije preuzet od strane serviser, a samim time ni završen. Pored takvih prijavi nalazi se gumb za uređivanje. Budući da prijava još nije potvrđena, uređivanje je omogućeno. Klikom na taj gumb, otvara se nova stranica :


Stranica za izmjenu podataka prijave :

Na ovoj stranici omogućuje se promjena dolaska u narednih 10 dana kada postoji serviser koji radi ujutro, dodavanje novih usluga ili ukidanje prethodno označenih, dodavanje dodatnih usluga te slobodni unos nekih dodatnih usluga.

Slika 32. Izmjena podataka prijave

Početna stranica za „običnog korisnika“ također sadrži i dva gumba : „Prijavi vozilo“ i „Odjava“. Pritiskom na gumb „Odjava“, korisnik se odjavljuje i vraća na početnu stranicu, a pritiskom na gumb „Prijavi vozilo“ otvara se stranica za odabir serviseru tog popravka.

Stranica za odabir serviseru novog zahtjeva :



Prijava Vozila na Popravak

Unesite podatke potrebne za prijavu

Proizvoljan odabir serviseru:

Serviser

Odabir...

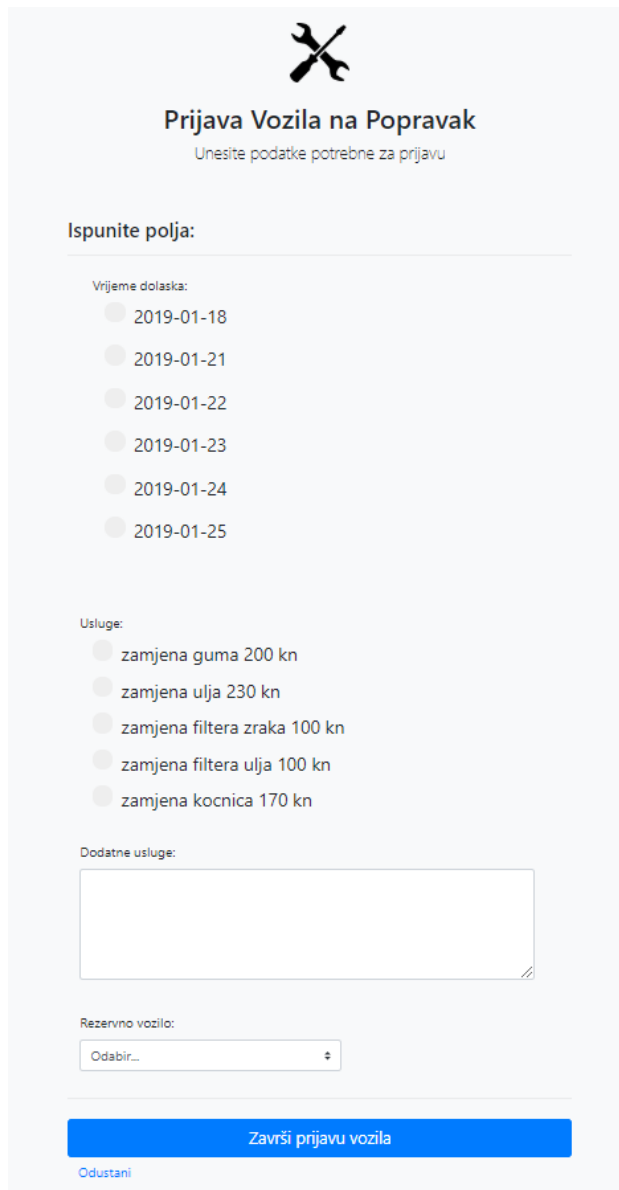
nastavi prijavu


[Odustani](#)

© 2017-2018 AutoServis

Slika 33. Odabir serviseru

Korisnik na ovoj stranici može odabrati određenog serviseru, a ukoliko to ne odluči učiniti, sustav će mu nasumično sam to odrediti. Klikom na gumb „nastavi prijavu“, korisnika se preusmjerava na stranicu za dodatne specifikacije zahtjeva.

Stranica za specifikaciju novog zahtjeva :



Prijava Vozila na Popravak

Unesite podatke potrebne za prijavu

Ispunite polja:

Vrijeme dolaska:

☐ 2019-01-18

☐ 2019-01-21

☐ 2019-01-22

☐ 2019-01-23

☐ 2019-01-24

☐ 2019-01-25

Usluge:

☐ zamjena guma 200 kn

☐ zamjena ulja 230 kn

☐ zamjena filtera zraka 100 kn

☐ zamjena filtera ulja 100 kn

☐ zamjena kocnica 170 kn

Dodatne usluge:

Rezervno vozilo:

[Završi prijavu vozila](#)

[Odustani](#)

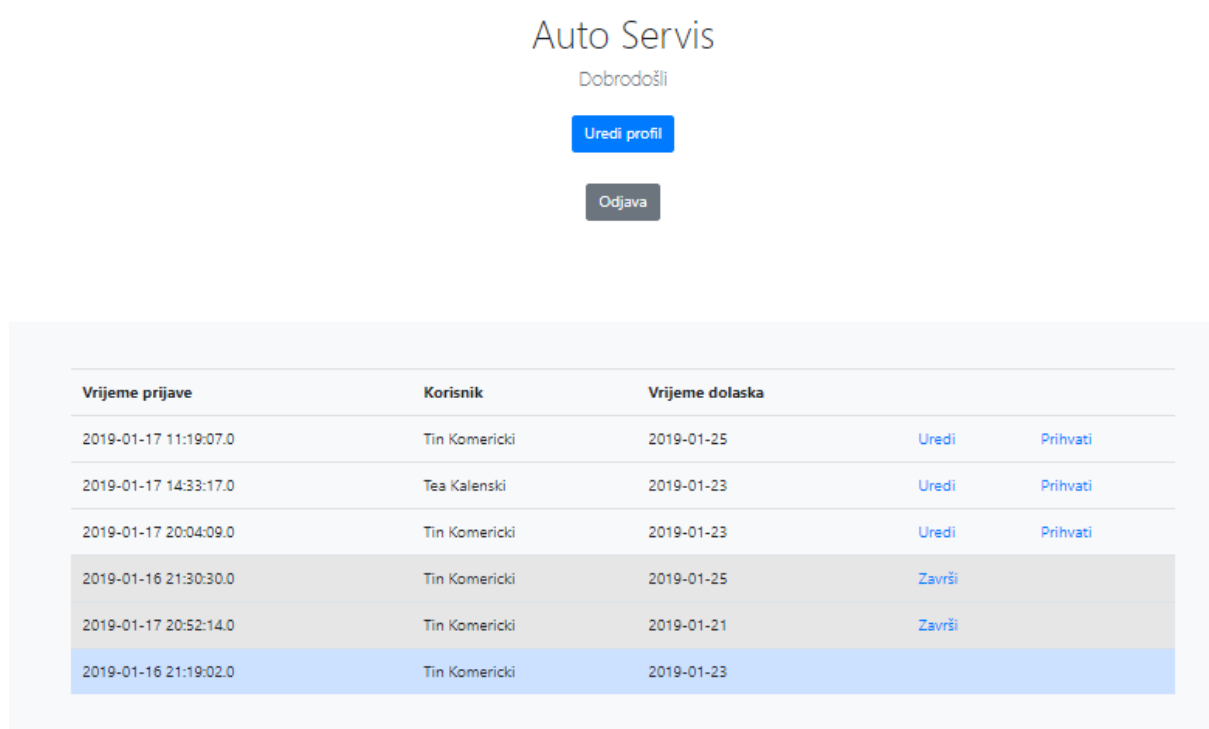
Slika 34. Odabir specifikacija popravka

Na ovoj stranici potrebno je navesti vrijeme dolaska te odabir usluge, dok se opcionalne usluge pišu u tekstualni okvir „Dodatne usluge“. Također, moguće je odabrati rezervno vozilo ukoliko korisnik želi jedno.

Klikom na gumb „Završi prijavu vozila“, prijava se pohranjuje te će se prikazati na stranici serviseru koji je zadužen za istu.

Ukoliko je ulogirani korisnik serviser, nakon prijavljivanja, otvara mu se početna stranica serviser.

Početna stranica serviser :



Slika 35. Glavna stranica serviser

Slično kao kod „običnog“ korisnika, serviseru su prikazane prijave kojima su korisnici zatražili da točno taj serviser njima popravlja auto.

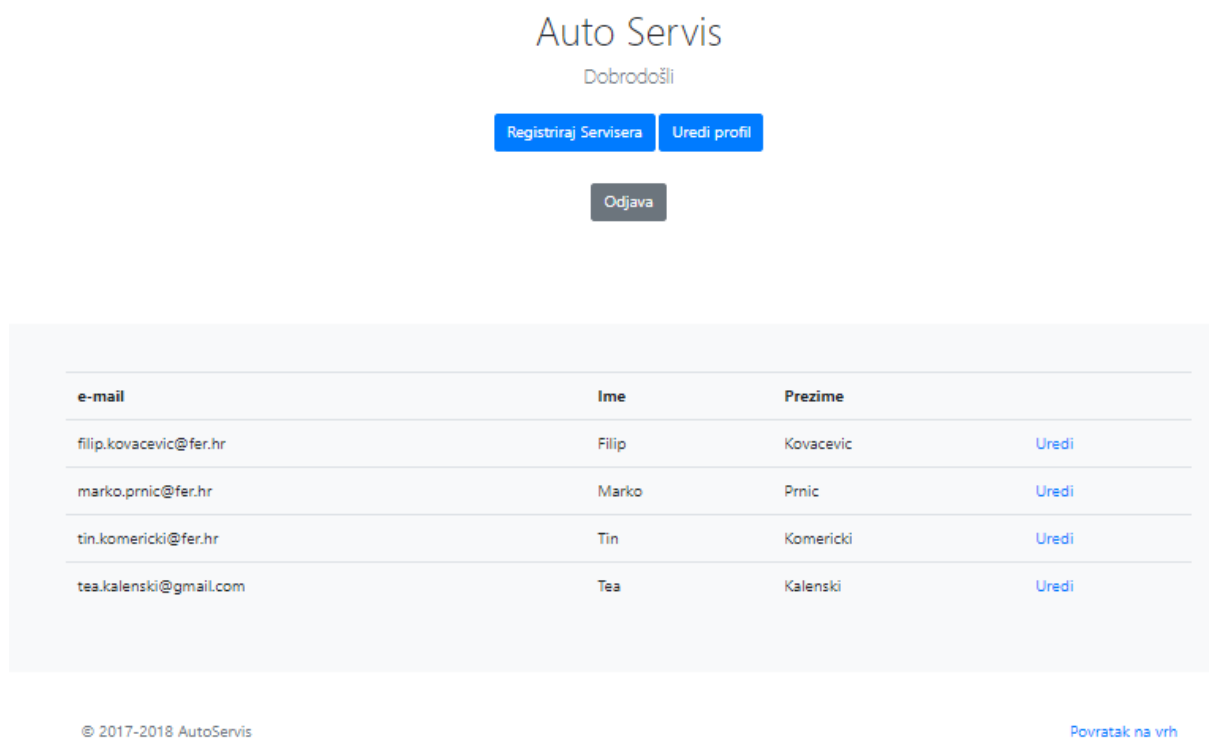
Plava boja predstavlja zahtjeve koje su obradili.

Tamno siva boja predstavlja zahtjeve koji su u procesu popravljavanja te nudi opciju „Završi“, kojom će se korisnik koji je zatražio popravak obavijestiti o gotovom popravku.

Svijetlo siva boja predstavlja zahtjeve koje su korisnici poslali prema tom serviseru, a koje serviser još nije prihvatio što može ostvariti pritiskom na gumb „Prihvati“, čime će se taj zahtjev obojati u tamno sivu boju. Također, takve zahtjeve serviser može i mijenjati pritiskom na gumb „Uredi“, čime se otvara serviserova stranica za uređivanje zahtjeva, koja se razlikuje od korisnikove samo po manjku opcije za izmjenu datuma dolaska.

Ukoliko je ulogirani korisnik administrator, nakon prijavljivanja, otvara mu se početna stranica administratora.


Početna stranica administratora :



Slika 36. Glavna stranica administratora

Administrator preko svoje stranice može uređivati servisere. Klikom na gumb „Registriraj Servisera“, preusmjerava ga se na stranicu za dodavanje serviser, koja je zapravo jednaka stranici za uređivanje serviser, koja se otvara klikom na gumb „Uredi“.

Stranica za uređivanje serviser :


Serviser
[Izmjeni podatke o serviseru](#)

Ispunite polja:

Ime

Prezime

Marko

Pronic

Email

marko.pronic@fer.hr

Lozinka

Lozinka

Broj Telefona

01254861

☐ Smjena 1

Ponedjeljak	Utorak	Srijeda	Četvrtak	Petak
07:00-13:00	12:00-18:00	07:00-13:00	12:00-18:00	07:00-13:00

☒ Smjena 2

Ponedjeljak	Utorak	Srijeda	Četvrtak	Petak
12:00-18:00	07:00-13:00	12:00-18:00	07:00-13:00	12:00-18:00

Spremi izmjene

[Odustani od izmjena.](#)

Slika 37. Registracija serviser

Na ovoj stranici, administrator može promijeniti osnovne podatke serviser te njegovu smjenu.

Klikom na gumb „Spremi izmjene“, novonastale izmjene se spremaju.

Klikom na gumb „Odustani od izmjena“, preusmjeruje ga se natrag na administratorovu stranicu.

7.4. Isječak programskog koda vezan za temeljnu funkcionalnost sustava

```

@RequestMapping(value={"/ispis"}, method = RequestMethod.GET)
public ModelAndView ispis(){
    ModelAndView modelAndView = new ModelAndView();
    Authentication auth = SecurityContextHolder.getContext().getAuthentication();
    User user = userService.findUserByEmail(auth.getName());
    List<User> users = new ArrayList<>();
    List<Prijava> prijave = new ArrayList<>();
    List<Prijava> preuzetePrijave = new ArrayList<>();
    List<Prijava> gotovePrijave = new ArrayList<>();
    String role = "";

    for(GrantedAuthority authority : auth.getAuthorities()) {
        role = authority.getAuthority();
        if(authority.getAuthority().equals("ADMIN")) users.addAll(userService.getAllUsers());
        else if(authority.getAuthority().equals("KORISNIK")) {
            prijave.addAll(userService.getUserPrijave(user.getId()));
            preuzetePrijave.addAll(userService.getUserPreuzetePrijave(user.getId()));
            gotovePrijave.addAll(userService.getUserZavrsenePrijave(user.getId()));
        }
        else if(authority.getAuthority().equals("SERVISER")) {
            prijave.addAll(userService.getServiserPrijave(user.getId()));
            preuzetePrijave.addAll(userService.getServiserPreuzetePrijave(user.getId()));
            gotovePrijave.addAll(userService.getServiserZavrsenePrijave(user.getId()));
        }
    }

    modelAndView.addObject("users", users);
    modelAndView.addObject("prijave", prijave);
    modelAndView.addObject("preuzetePrijave", preuzetePrijave);
    modelAndView.addObject("gotovePrijave", gotovePrijave);
    modelAndView.addObject("role", role);

    modelAndView.setViewName("pocetnaLogin");
    return modelAndView;
}

```

Slika 38. Metoda kontrolera za prikaz glavne stranice korisnika

Metoda razreda Controller koja je zadužena za prikaz početne stranice korisniku koji se tek prijavio na sustav ovisno o njegovoj ulozi. Ukoliko se radi o adminu, iz baze podataka će se dohvatiti svi korisnici koji postoje na sustavu, a ukoliko se radi o serviseru ili korisniku iz baze će se dohvatiti prijave koje su vezane uz tog korisnika podijeljene u 3 kategorije: podnesene, preuzete i gotove.

U donjem dijelu metode (1) se svi potrebni podaci prenose na frontend koji te podatke može pokazati u određenom formatu.

```

@Service("userService")
public class UserService {

    private UserRepository userRepository;
    private PrijavaRepository prijavaRepository;
    private UslugaRepository uslugaRepository;
    private ZamjenskoVoziloRepository zamjenskoVoziloRepository;
    private RadnoVrijemeRepository radnoVrijemeRepository;
    private VezaUslugaPrijavaRepository vezaUslugaPrijavaRepository;
    private BCryptPasswordEncoder bCryptPasswordEncoder;

    @Autowired
    public UserService(UserRepository userRepository, PrijavaRepository prijavaRepository,
        UslugaRepository uslugaRepository, RadnoVrijemeRepository radnoVrijemeRepository,
        ZamjenskoVoziloRepository zamjenskoVoziloRepository,
        VezaUslugaPrijavaRepository vezaUslugaPrijavaRepository,
        BCryptPasswordEncoder bCryptPasswordEncoder) {
        this.userRepository = userRepository;
        this.prijavaRepository = prijavaRepository;
        this.uslugaRepository = uslugaRepository;
        ① this.zamjenskoVoziloRepository = zamjenskoVoziloRepository;
        this.radnoVrijemeRepository = radnoVrijemeRepository;
        this.vezaUslugaPrijavaRepository = vezaUslugaPrijavaRepository;
        this.bCryptPasswordEncoder = bCryptPasswordEncoder;
    }
}

```

Slika 39. Konstruktor razreda UserService

```

public User replaceUser(User newUser) {
    System.out.println(newUser.getId());
    User oldUser = userRepository.findById(newUser.getId()).get();
    if(newUser.getPassword().isEmpty()) newUser.setPassword(oldUser.getPassword());
    else newUser.setPassword(bCryptPasswordEncoder.encode(newUser.getPassword()));
    //userRepository.delete(oldUser);
    userRepository.save(newUser);

    return newUser;
}

public User saveServiser(User serviser) {
    serviser.setPassword(bCryptPasswordEncoder.encode(serviser.getPassword()));
    serviser.setId(this.getAllUsers().size());
    serviser.setActive(1);
    serviser.setRole("SERVISER"); ②
    return userRepository.save(serviser);
}

public List<User> getAllUsers(){
    List<User> users = new ArrayList<>();
    users.addAll(userRepository.findAll());
    users.removeIf(r -> r.getRole().equals("ADMIN"));
    return users;
}

public Set<User> getAllServiseri(){
    Set<User> serviseri = new HashSet<>(); ③
    serviseri.addAll(userRepository.findAll());
    serviseri.removeIf(r -> !r.getRole().equals("SERVISER"));
    return serviseri;
}

```

Slika 40. Neke od metoda razreda UserService

Ovdje je prikazana djelomična implementacija razreda koji je odgovoran za dohvaćanje, izmjenu i spremanje podataka u sustavu. Razred u sebi sadrži sve repozitorije podataka⁽¹⁾ i ovisno o zahtijevanoj akciji i ulozi korisnika, obavlja pojedinu operaciju. Također, razred sadrži objekt koji služi za kriptiranje lozinke (bCryptPasswordEncoder). Na primjer, prilikom registracije serviseru poziva se metoda saveServiser(...) ⁽²⁾ koja kriptira serviserovu lozinku, postavlja mu ulogu u sustavu i šalje ga na pohranu u repozitorij korisnika sustava. Metoda getAllServiseri() ⁽³⁾ zadužena je za dohvat svih serviseru u sustavu koji će se korisniku prikazati na izbor prilikom stvaranja prijave popravka svog automobila.

```
@Override
protected void configure(HttpSecurity http) throws Exception {

    http.
        authorizeRequests()
        .antMatchers("/").permitAll()
        .antMatchers("/pocetna").permitAll()
        .antMatchers("/login").permitAll()
        .antMatchers("/registracija").permitAll()
        .antMatchers("/editKorisnik").hasAuthority("ADMIN")
        .antMatchers("/editPrijava").hasAnyAuthority("KORISNIK", "SERVISER")
        // .antMatchers("/editPrijava").hasAuthority("SERVISER")
        .antMatchers("/ispis").authenticated()
        .antMatchers("/registracijaServiseru").hasAuthority("ADMIN")
        .antMatchers("/popravak").hasAuthority("KORISNIK")
        .antMatchers("/popravak2").hasAuthority("KORISNIK")
        .antMatchers("/prijavaPopravka").hasAuthority("KORISNIK")
        .antMatchers("/admin/**").hasAuthority("ADMIN").anyRequest()
        .authenticated().and().csrf().disable().formLogin()
        ① .loginPage("/login").failureUrl("/login?error=true")
        .defaultSuccessUrl("/ispis")
        .usernameParameter("email")
        .passwordParameter("password")
        .and().logout()
        .logoutRequestMatcher(new AntPathRequestMatcher("/logout"))
        .logoutSuccessUrl("/").and().exceptionHandling()
        .accessDeniedPage("/access-denied");
}
```

Slika 41. Ograničenja pristupa prema vrsti autentikacije

Jedna od ključnih funkcionalnosti ove aplikacije jest ograničenje pristupa pojedinim stranicama na temelju uloge korisnika u sustavu. Ovdje je prikazana implementacija tih ograničenja u skladu sa ulogama pojedinih korisnika sustava (admin, serviser, registrirani korisnik). Metoda permitAll() omogućuje bilo kome da pristupi danim resursima. Metoda hasAuthority(String role) omogućava isključivo ulogiranim korisnicima koji imaju ulogu u sustavu „role“ da pristupe tom resursu, dok za pristup resursu ograničenom metodom hasAnyAuthority(niz Stringova) mora imati jednu od navedenih uloga u nizu Stringova. Metoda authenticated() zahtjeva da je korisnik ulogiran, neovisno o njegovoj ulozi u sustavu. Dio koda ⁽¹⁾ definira koja stranica služi za login te gdje će se korisnika preusmjeriti u ovisnosti o uspješnosti logiranja te koji se podaci koriste za login. Nakon toga definira se i koja stranica služi za logout te gdje se korisnika preusmjerava u ovisnosti o uspješnosti odlogiranja.

```

@Data
@Builder
@AllArgsConstructor
@NoArgsConstructor
@Entity
public class ZamjenskoVozilo {

    @Id
    @Column(name = "reg_oznaka_vozila")
    @Length(max = 20)
    private String registracijskaOznakaVozila;

    @Column(name = "id_korisnika")
    private String idKorisnik;

    @NotEmpty
    @Column(name = "tip_vozila")
    private String tipVozila;

    @Column(name = "godina_proizvodnje")
    private int godinaProizvodnje;
}

```

Slika 42. Razred ZamjenskoVozilo koje stvara istoimenu tablicu

```

@Data
@Builder
@AllArgsConstructor
@NoArgsConstructor
@Entity
public class Prijava {

    @EmbeddedId
    private PrijavaKey prijavaKey;
    @Column(name = "idServisera")
    private int idServisera;
    @Column(name = "vrijemeDolaska")
    private LocalDate vrijemeDolaska;
    @Column(name = "preuzeto")
    private boolean preuzeto;
    @Column(name = "zavrsono")
    private boolean zavrsono;
    @Column(name = "vrijemeZavrsetka")
    private Timestamp vrijemeZavrsetka;
    @Column(name = "dodatniZahtjevi")
    private String dodatniZahtjevi;
    @Column(name = "regZamjensko")
    private String regZamjensko;
}

```

Slika 43. Razred Prijava koja stvara istoimenu tablicu


```
@Embeddable
@EqualsAndHashCode
@AllArgsConstructor
public class PrijavaKey implements Serializable{
    @Column(name = "idKorisnika")
    private int idKorisnika;
    @Column(name = "vrijemePrijave")
    private Timestamp vrijemePrijave;
```

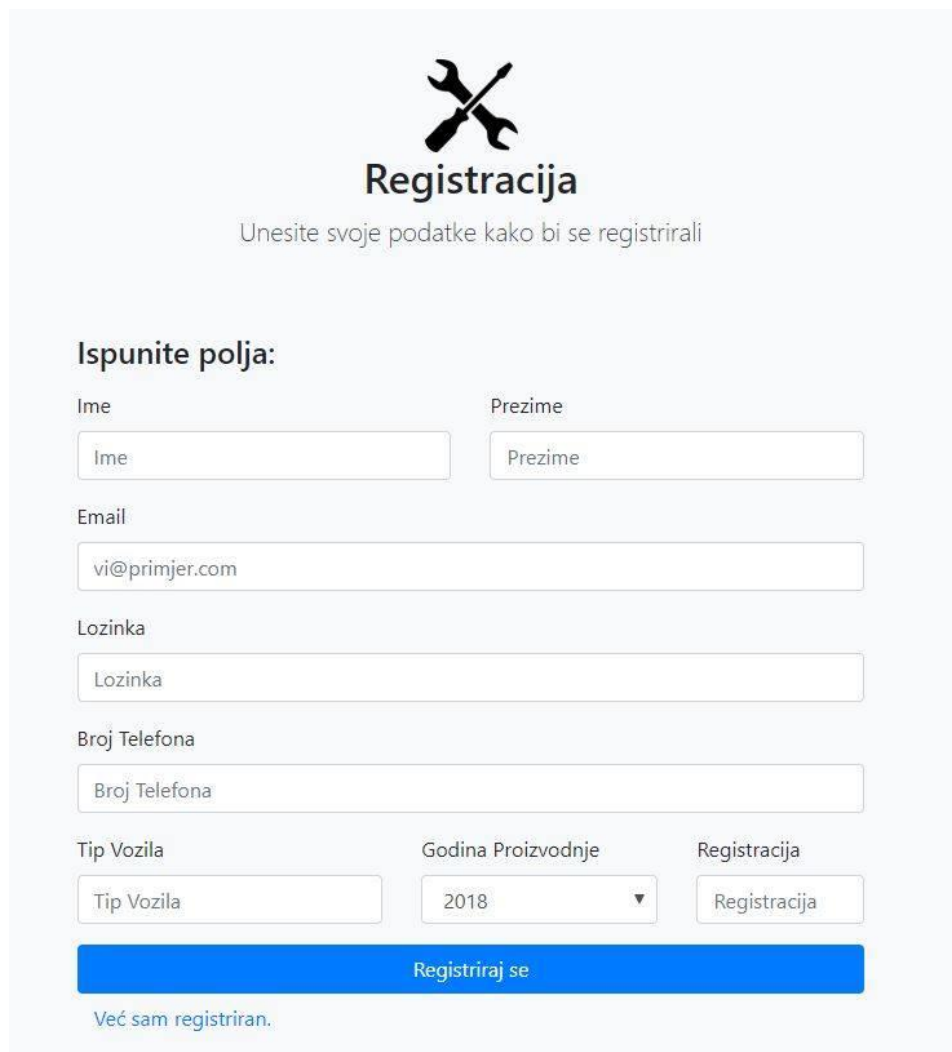
Slika 44. Razred PrijavaKey, ključ (eng. Composite Key) tablice Prijava

Implementacija jednog modela naše aplikacije, prijave. Prilikom pokretanja aplikacije, automatski se stvara tablica prijava u bazi podataka na temelju ovog razreda i u tu tablicu se pohranjuju podatci o svakoj prijavi vozila na servis u sustavu. Na prvoj slici prvi atribut klase označen je anotacijom @Id što znači da će taj atribut biti ključ tablice. Na drugoj slici je prvi atribut označen oznakom @EmbeddedId, što znači da ova tablica ima više atributa (stupaca u tablici) kao ključ tablice (eng. Composite Key). Takav ključ se mora pohraniti u zasebni razred koji je vidljiv na slici 3 i u kojem su navedeni svi atributi koji čine ključ. Anotacija @Column govori da će taj atribut biti stupac u tablici u bazi podataka.

7.5. Ispitivanje programskog rješenja

7.5.1. Registracija korisnika

- **Opis ispitnog slučaja:**
Neregistrirani korisnik pritiskom na gumb „Registracija“ pokreće registraciju u sustav. Upisuje svoje podatke, prijavljuje informacije o vozilu te potvrđuje podatke pritiskom na gumb „Registriraj se“
- **Očekivani rezultat:**
U slučaju ispravno unesenih podataka, korisnik se uspješno registrira i postaje registrirani korisnik. U slučaju neispravno unesenih podataka, korisnika se obavještava prikladnom porukom
- **Dobiveni rezultat:**
U slučaju točno upisanih podataka, korisnik je uspješno registriran u sustav, dok je pri upisivanju krivih podataka korisniku ispisana poruka o grešci



The screenshot shows a web form for user registration. At the top, there is a logo consisting of two crossed wrenches and the title "Registracija" in a bold, sans-serif font. Below the title, a subtitle reads "Unesite svoje podatke kako bi se registrirali". The form is organized into several sections with labels: "Ispunite polja:". The input fields include: "Ime" (First Name) and "Prezime" (Last Name), both with placeholder text; "Email" with a placeholder "vi@primjer.com"; "Lozinka" (Password) with a placeholder "Lozinka"; "Broj Telefona" (Phone Number) with a placeholder "Broj Telefona"; "Tip Vozila" (Vehicle Type) with a placeholder "Tip Vozila"; "Godina Proizvodnje" (Year of Production) with a dropdown menu showing "2018"; and a "Registracija" button. At the bottom of the form, there is a large blue button labeled "Registriraj se" and a link that says "Već sam registriran.".

Slika 45. Obrazac za registraciju


Registracija
Unesite svoje podatke kako bi se registrirali

Ispunite polja:

Ime

Unos imena je obavezan.

Prezime

Unos prezimena je obavezan.

Email

Unesite ispravan e-mail.

Lozinka

Unos lozinke je obavezan.

Broj Telefona

Unos broja telefona je obavezan.

Tip Vozila

Molimo unesite tip vozila.

Godina Proizvodnje

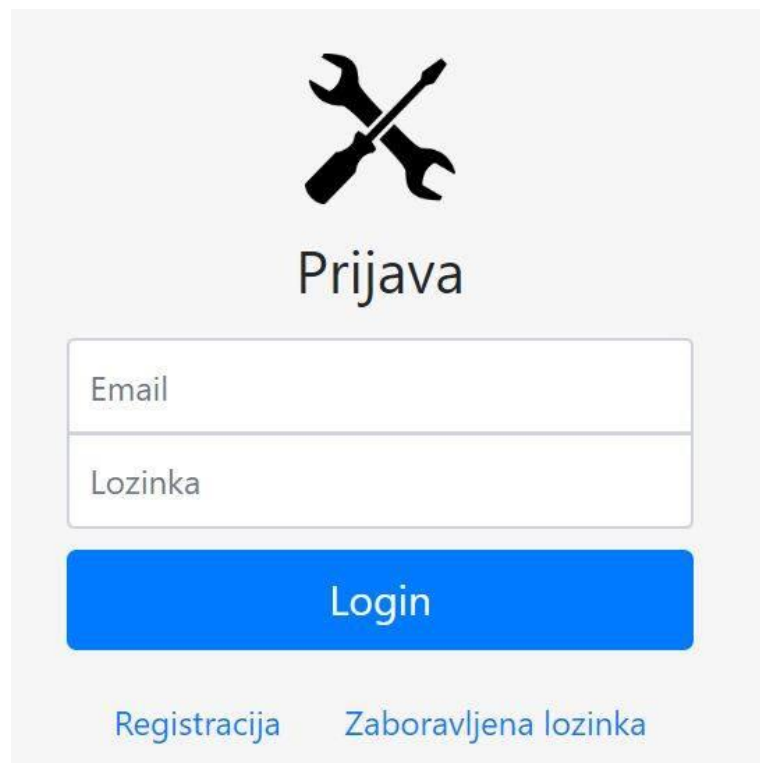
Registracija
☐
Unos registracije je obavezan.

Registriraj se
Već sam registriran.

Slika 46. Neuspjela registracija

7.5.2. Prijava na korisnika na sustav

- **Opis ispitnog slučaja:**
Neregistrirani korisnik pritiskom na gumb "Prijava" pokreće prijavu na sustav. Upisuje svoj e-mail i lozinku koji služe kao identifikacijski elementi
- **Očekivani rezultat:**
U slučaju ispravno unesenih podataka, korisnik se uspješno prijavljuje na sustav, te dobiva pregled nad svojim zatraženim popravcima i dr. U slučaju neispravno unesenih podataka, korisnika se obavještava prikladnom porukom. Također, u slučaju krivo unesene lozinke ili e-maila, korisnika se obavještava o neuspješnoj prijavi
- **Dobiveni rezultat:**
U slučaju točno upisanih podataka, korisnik je uspješno prijavljen na sustav, dok je pri upisivanju krivih podataka korisniku ispisana poruka o grešci



The image shows a login form titled "Prijava" (Login). At the top, there is an icon of two crossed wrenches. Below the title, there are two input fields: "Email" and "Lozinka" (Password). Below these fields is a blue button labeled "Login". At the bottom of the form, there are two links: "Registracija" (Registration) and "Zaboravljena lozinka" (Forgot password).

Slika 47. Obrazac za prijavu



Prijava

Molimo unesite ispravan Email.

Molimo unesite Lozinku.


Login

[Registracija](#) [Zaboravljena lozinka](#)

Slika 48. Neuspjela prijava

7.5.3. Izmjena podataka o serviseru

- **Opis ispitnog slučaja:**
Administrator pritiskom na gumb „Edit“ kod serviseru započinje izmjenu podataka o serviseru
- **Očekivani rezultat:**
Prilikom izmjene podataka očekuje se unošenje svih obaveznih podataka o serviseru. Polje lozinka ostaje prazno ako se ne želi mijenjati
- **Dobiveni rezultat:**
U slučaju ispunjenja svih obaveznih podataka, izmijenjeni serviser sprema se u sustav, dok je pri neispunjavanju bilo kojeg obaveznog podataka administratoru ispisana poruka o grešci



Serviser

Izmjeni podatke o serviseru

Ispunite polja:

Ime

Prezime

Email

Lozinka

Broj Telefona

☒ Smjena 1


Ponedjeljak	Utorak	Srijeda	Četvrtak	Petak
07:00-13:00	12:00-18:00	07:00-13:00	12:00-18:00	07:00-13:00

☐ Smjena 2

Ponedjeljak	Utorak	Srijeda	Četvrtak	Petak
12:00-18:00	07:00-13:00	12:00-18:00	07:00-13:00	12:00-18:00

[Odustani od izmjena.](#)

Slika 49. Obrazac za izmjenu podataka serviseru


Serviser
Izmjeni podatke o serviseru

Ispunite polja:

Ime

Prezime

Email

Lozinka

Broj Telefona

Unos broja telefona je obavezan.

☒ Smjena 1

Ponedjeljak	Utorak	Srijeda	Četvrtak	Petak
07:00-13:00	12:00-18:00	07:00-13:00	12:00-18:00	07:00-13:00

☐ Smjena 2

Ponedjeljak	Utorak	Srijeda	Četvrtak	Petak
12:00-18:00	07:00-13:00	12:00-18:00	07:00-13:00	12:00-18:00

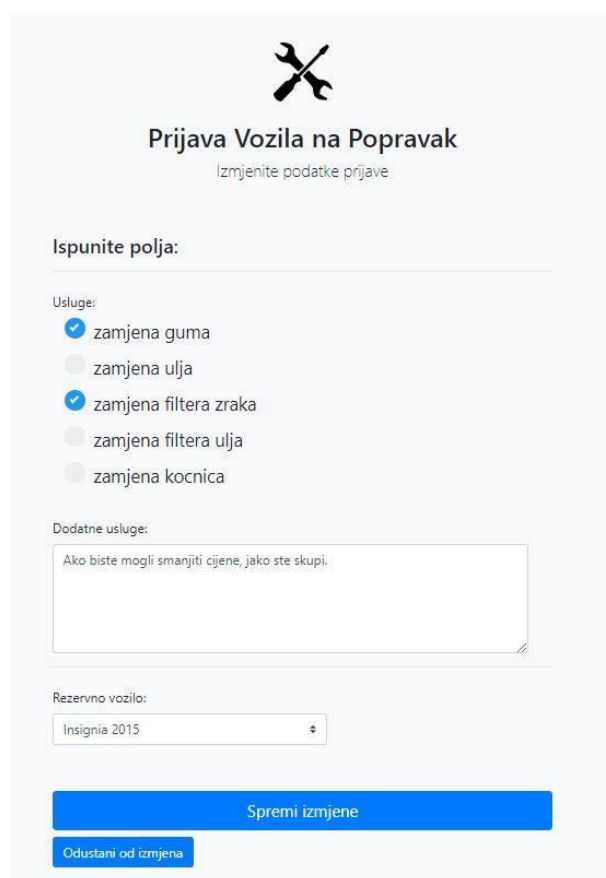
Spremi izmjene


Odustani od izmjena.

Slika 50. Neuspjela izmjena podataka serviseru

7.5.4. Izmjena prijave vozila na popravak

- **Opis ispitnog slučaja:**
Serviser može mijenjati prijave vozila za koje je odabran (ne može mijenjati vrijeme dolaska)
- **Očekivani rezultat:**
U prijavi mora ostati barem jedna usluga kako bi bila valjana, u slučaju da serviser makne sve usluge treba mu se ispisati pogreška i poništiti promjena
- **Dobiveni rezultat:**
Uspješna izmjena prijave pohranjuje se u sustavu, u suprotnome se serviseru ispisuje poruka o grešci i vraćaju mu se početni podaci o prijavi





Prijava Vozila na Popravak

Izmjenite podatke prijave

Ispunite polja:

Usluge:

- ☒ zamjena guma
- ☐ zamjena ulja
- ☒ zamjena filtera zraka
- ☐ zamjena filtera ulja
- ☐ zamjena kocnica

Dodatne usluge:

Ako biste mogli smanjiti cijene, jako ste skupi.


Rezervno vozilo:

Insignia 2015

Spremi izmjene

Odustani od izmjena

Slika 51. Serviserov obrazac za izmjenu prijave popravka



Prijava Vozila na Popravak

Izmijenite podatke prijave

Ispunite polja:

Usluge:

- ☐ zamjena guma
- ☐ zamjena ulja
- ☐ zamjena filtera zraka
- ☐ zamjena filtera ulja
- ☐ zamjena kocnica

Dodatne usluge:

Ako biste mogli smanjiti cijene, jako ste skupi.


Rezervno vozilo:

Insignia, 2015

Spremi izmjene

Odustani od izmjena

Slika 52. Serviser je uklonio sve usluge iz prijave popravka



Prijava Vozila na Popravak

Izmijenite podatke prijave

Ispunite polja:

Usluge:

- ☒ zamjena guma
- ☐ zamjena ulja
- ☒ zamjena filtera zraka
- ☐ zamjena filtera ulja
- ☐ zamjena kocnica

Molimo odaberite barem jednu uslugu za popravak

Dodatne usluge:

Ako biste mogli smanjiti cijene, jako ste skupi.

Rezervno vozilo:

Insignia 2015

Spremi izmjene

Odustani od izmjena

Slika 53. Sustav izbacuje grešku i vraća početne podatke u obrazac

8. Zaključak i budući rad

Cilj projekta je implementirati web aplikaciju putem koje će korisnici svoje automobile prijaviti na popravak u samo par koraka. Taj zadatak mora biti implementiran na način da korisnik može jednostavno koristiti aplikaciju što je i uspješno odrađeno.

Kao i sve u svijetu, i ova aplikacija se može poboljšati i/ili nadograditi što je zapravo temelj budućeg rada.

Sigurnost je jedna od najvažnijih dijelova svake aplikacije. Kako je ovo prvi timski projekt koji je moj tim radio, uvjeren sam da postoje propusti na ovom području koje bi trebalo poboljšati. Također, razvojem tehnologije, raste i opasnost od probijanja sigurnosti, stoga je potrebno, bez obzira na dosad napravljen posao, konstantno nadograđivati sigurnost aplikacije dodatnim mogućnostima koje se pružaju razvojem algoritama zaštite, mogućnosti programskog jezika java i radnog okvira Spring Boot.

Sljedeća točka budućeg rada je nadogradnja funkcionalnosti/podataka. Možda će korisnici htjeti znati izgled svog serviseru stoga bi se u budućnosti mogla implementirati slika u serviserove podatke. Razvojem auto industrije, moglo bi se tražiti više podataka o automobilu koji se šalje na popravak, stoga bi se i oni trebali pohraniti i obrađivati. Dodatno, korisnika bi moglo zanimati više podataka od samo tipa i godine proizvodnje zamjenskog vozila što je još jedan zadatak za budući rad. Naravno ima još mnoštvo mogućnosti koje bi s vremenom mogle biti razmatrane, al to je, naravno, zadatak za budući rad.

Čistoća koda koji je napisan je isto vrlo važan kako bi osoba ili tim koji preuzima zadatak održavanja i nadogradnje aplikacije mogao jasno razumjeti čemu pojedini dio koda služi i na što sve treba pripaziti prilikom promjene koda. Stoga bi zadatak za budući rad bio počistiti redundantan kod, promijeniti imena varijabla čije ime ne odražava njihovu ulogu, bolje organizirati datoteke u direktorije ovisno o njihovoj ulozi i sl.

Brzina izvođenja je još jedna važna stavka aplikacije. Čistoća koda može utjecati na brzinu aplikacije, no za brzinu je važnije koristiti odgovarajuće algoritme i tehnike ovisno o situaciji te dohvaćati samo podatke iz baze podataka koji su doista potrebni. Uvjeren sam da postoji poneki dio koda u kojem se mogao iskoristiti brži algoritam ili upotrijebiti bolja tehnika pisanja. Stoga bi zadatak budućeg rada bio takve propuste popraviti.

9. Popis literature

- ¹ Oblikovanje programske potpore, FER ZEMRIS, <http://www.fer.hr/predmet/opp>
- ² Astah Community, <http://astah.net/editions/community/>
- ³ SourceTree tool, <https://www.sourcetreeapp.com/>

Dodatak A: Indeks (slika, dijagrama, tablica, ispisa kôda)

Slika 1. Dijagram obrazaca uporabe za cijeli auto servis	11
Slika 2. Dijagram obrazaca uporabe za administratora	12
Slika 3. Dijagram obrazaca uporabe za korisnika	12
Slika 4. Dijagram obrazaca uporabe za prijavu	13
Slika 5. Dijagram obrazaca uporabe za serviseru	13
Slika 6. Dijagram obrazaca uporabe za obradu prijave vozila	14
Slika 7. Sekvencijski dijagram - Osnovni upis podataka	15
Slika 8. Sekvencijski dijagram - Potvrda prijave vozila na popravak	15
Slika 9. Sekvencijski dijagram - Pregled vlastitih korisnika	16
Slika 10. Sekvencijski dijagram - Prijava na sustav	16
Slika 11. Sekvencijski dijagram - Prijava vozila na popravak	17
Slika 12. Sekvencijski dijagram – Registracija	18
Slika 13. Sekvencijski dijagram – Upravljanje informacijama registriranih korisnika	18
Slika 14. ER model baze podataka	21
Slika 15. Dijagram razreda	28
Slika 16. Dijagram objekata	29
Slika 17. Dijagram stanja – prijava na sustav	30
Slika 18. Dijagram stanja – izmjena podataka o korisniku	31
Slika 19. Dijagram aktivnosti – registracija korisnika	32
Slika 20. Dijagram aktivnosti – pregled korisnika	33
Slika 21. Komunikacijski dijagram – prijava registriranog korisnika na sustav	34
Slika 22. Komunikacijski dijagram - prijava vozila na popravak	35
Slika 23. Dijagram komponenti	36
Slika 24. Dijagram razmještaja	37
Slika 25. Početna stranica	39
Slika 26. Podaci o auto servisu	39
Slika 27. Registracija korisnika	40
Slika 28. Pogreške prilikom registracije	41
Slika 29. Prijava korisnika u sustav	41
Slika 30. Pogrešni podaci prijave	42
Slika 31. Glavna stranica "običnog" korisnika	43
Slika 32. Izmjena podataka prijave	44
Slika 33. Odabir serviseru	45
Slika 34. Odabir specifikacija popravka	46
Slika 35. Glavna stranica serviseru	47
Slika 36. Glavna stranica administratora	48
Slika 37. Registracija serviseru	49
Slika 38. Metoda kontrolera za prikaz glavne stranice korisnika	50
Slika 39. Konstruktor razreda UserService	51
Slika 40. Neke od metoda razreda UserService	51
Slika 41. Ograničenja pristupa prema vrsti autentifikacije	52
Slika 42. Razred ZamjenskoVozilo koje stvara istoimenu tablicu	53
Slika 43. Razred Prijava koja stvara istoimenu tablicu	53

Slika 44. Razred PrijavaKey, ključ (eng. Composite Key) tablice Prijava.....	54
Slika 45. Obrazac za registraciju	55
Slika 46. Neuspjela registracija.....	56
Slika 47. Obrazac za prijavu	57
Slika 48. Neuspjela prijava.....	58
Slika 49. Obrazac za izmjenu podataka serviseru.....	59
Slika 50. Neuspjela izmjena podataka serviseru.....	60
Slika 51. Serviserov obrazac za izmjenu prijave popravka	61
Slika 52. Serviser je uklonio sve usluge iz prijave popravka	62
Slika 53. Sustav izbacuje grešku i vraća početne podatke u obrazac	63
Slika 54. Aktivnost grupe u grani master za prvu verziju dokumentacije	71
Slika 55. Aktivnost grupe u grani master za drugu verziju dokumentacije	71
Slika 56. Aktivnost grupe u grani Dokumentacija za prvu verziju dokumentacije	71
Slika 57. Aktivnost grupe u grani Dokumentacija za drugu verziju dokumentacije	72
Slika 58. Aktivnost grupe u grani aplikacija	72

Dodatak B: Dnevnik sastajanja

21.10.2018.	Filip Jandžel Tea Kalenski Tin Komerički Filip Kovačević Marko Prnić	Međusobno upoznavanje svih članova tima, razgovor članova o njihovim vještinama i područjima na kojima bi htjeli raditi, podjela tima na backend i frontend, dodijelen zadatak izrade dijagrama obrazaca uporabe.
03.11.2018.	Dino Avdić Filip Jandžel Tea Kalenski Tin Komerički Filip Kovačević Marko Prnić	Raspored članova tima po poglavljima dokumentacije, maksimalno 2 poglavlja po članu tima. Dodjela izrade sekvencijskih dijagrama.
08.11.2018.	Dino Avdić Filip Jandžel Tea Kalenski Tin Komerički Filip Kovačević Marko Prnić	Upoznavanje tima sa alatom SourceTree za lakši rad sa Gitlabom. Dodjeljivanje zadatka izrade ER modela baze podataka.
18.11.2018.	Dino Avdić Filip Jandžel Tea Kalenski Tin Komerički Filip Kovačević Marko Prnić	Dodjela završnih poslova članovima tima za dovršetak prve verzije dokumentacije. Dogovor o izradi dijagrama razreda i objekata.
01.12.2018.	Dino Avdić Filip Jandžel Tea Kalenski Tin Komerički Filip Kovačević Marko Prnić	Kratka rasprava o dosad napravljenj dokumentaciji, pojašnjenje nejasnoća pojedinih područja drugim članovima. Zakazan sljedeći sastanak za 8.12. gdje će se krenuti za planom izrade aplikacije.
8.12.2018.	Dino Avdić Filip Jandžel Tea Kalenski Tin Komerički Filip Kovačević Marko Prnić	Definirane faze projekta, definirana raspodjela poslova po članovima tima.
20.12.2018.	Dino Avdić Filip Jandžel Tea Kalenski Tin Komerički Filip Kovačević Marko Prnić	Iznenadni sastanak zbog zimskih praznika, dogovor oko početka rada između Božića i Nove Godine.
3.1.2019.	Filip Jandžel Tea Kalenski Tin Komerički	Analiza dosad napravljenog dijela aplikacije, nastavak rada i planiranje za raspodjelu poslova vezanih za dokumentaciju

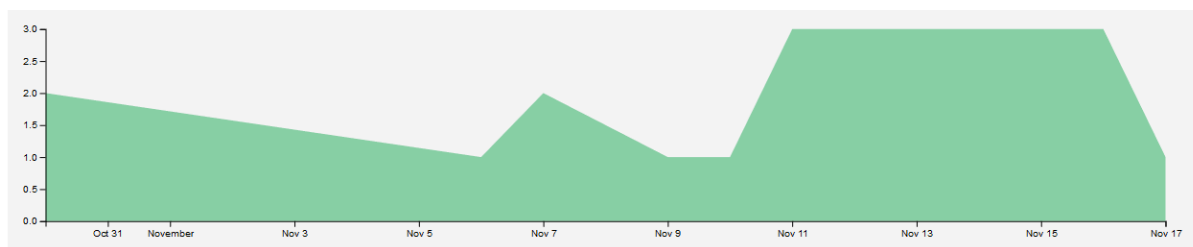
12.1.2019.	Dino Avdić Filip Jandžel Tea Kalenski Tin Komerički Filip Kovačević Marko Prnić	Završni sastanak, posljednja podijela poslova i završavanje kompletne dokumentacije, zakazane završne konzultacije prije prezentacije aplikacije za 21.1.2019.
------------	------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------

Dodatak C: Prikaz aktivnosti grupe

Popis aktivnosti	Članovi grupe					
	Tin Komerički	Dino Avdić	Filip Janđel	Tea Kalenski	Filip Kovačević	Marko Prnić
Opis zadatka i riječnik pojmova	10%		90%			
Funkcionalni zahtjevi	70%			30%		
Dijagram razreda	10%	50%		40%		
Dijagram objekata			100%			
ER model baze podataka					40%	60%
Opis relacija baze podataka					40%	60%
Relacijski model baze podataka						100%
Stilsko uređivanje dokumentacije	60%	25%	15%			
Izrada HTML stranica	20%			80%		
Stilsko uređivanje stranica				100%		
Izrada backend logike	80%		20%			
Dijagrami stanja, komponenti i razmještaja		100%				
Dijagrami aktivnosti, komunikacijski dijagram		100%				
Izrada korisničkih uputa					100%	
Dokumentiranje koda	20%		80%			

October 30, 2018 – November 17, 2018

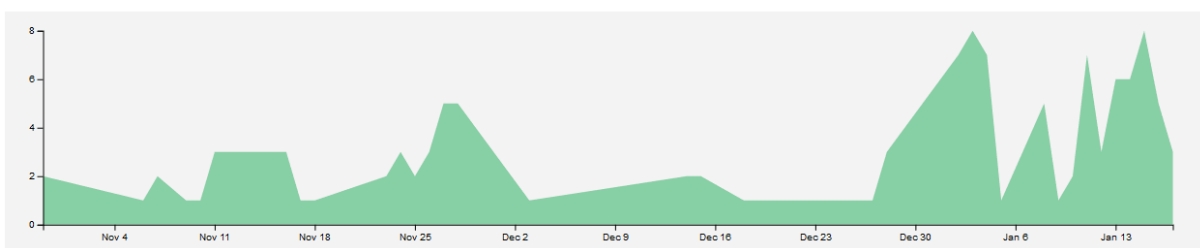
Commits to master, excluding merge commits. Limited to 6,000 commits.



Slika 54. Aktivnost grupe u grani master za prvu verziju dokumentacije

October 30, 2018 – January 17, 2019

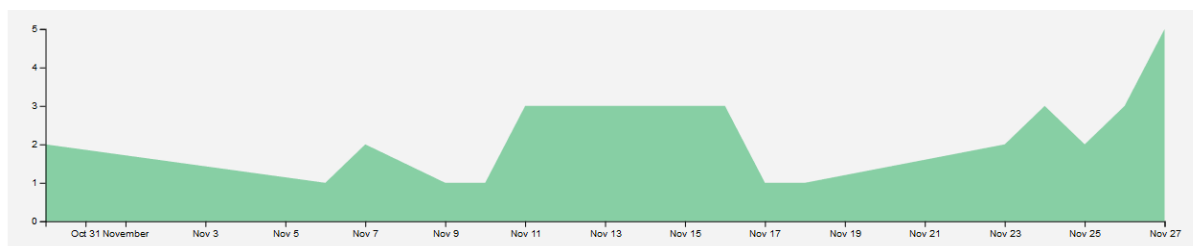
Commits to master, excluding merge commits. Limited to 6,000 commits.



Slika 55. Aktivnost grupe u grani master za drugu verziju dokumentacije

October 30, 2018 – November 27, 2018

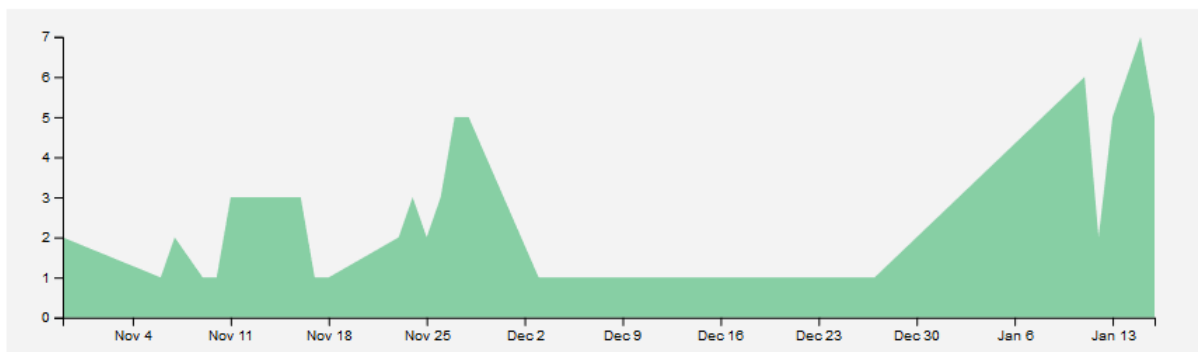
Commits to Dokumentacija, excluding merge commits. Limited to 6,000 commits.



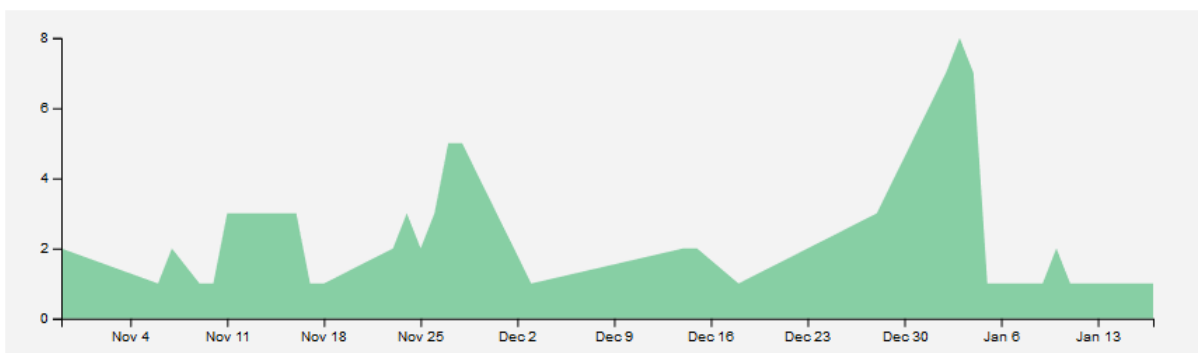
Slika 56. Aktivnost grupe u grani Doklumentacija za prvu verziju dokumentacije

October 30, 2018 – January 16, 2019

Commits to Dokumentacija, excluding merge commits. Limited to 6,000 commits.

*Slika 57. Aktivnost grupe u grani Dokumentacija za drugu verziju dokumentacije***October 30, 2018 – January 17, 2019**

Commits to aplikacija, excluding merge commits. Limited to 6,000 commits.

*Slika 58. Aktivnost grupe u grani aplikacija*

Dodatak D: Plan rada

Izrada aplikacije podijeljena je u 7 faza. Svaka faza djelomično ili potpuno ovisi o prethodnim fazama.

1.FAZA

- Izrada baze podataka
- Povezivanje baze podataka sa praznom Spring Boot aplikacijom
- Stvaranje stranica za registraciju i prijavu

2.FAZA

- Povezivanje stvorenih stranica sa aplikacijom (aplikacija pristupa tim stranicama)
- Izrada metoda za dohvat podataka o korisnicima
- Izrada metoda u kontroleru koje će registrirati i ulogirati korisnika na osnovu ispunjene forme na stranici, ispis odgovarajućih poruka u slučaju krivih podataka
- Prva verzija skripte koja će se izvršiti prilikom pokretanja aplikacije i koja će u bazu podataka spremi podatke o adminu
- Izrada stranice za registraciju serviseru (odvojena stranica od one za registraciju korisnika)

3.FAZA

- Izrada backend logike koja će dohvatiti drugačije podatke ovisno o tome koji se tip korisnika ulogira:
 - Adminu se dohvaćaju svi korisnici
 - Korisniku prijave popravka koje je on stvorio
 - Serviseru prijave popravka koje su upućene njemu, odnosno koje on mora popraviti
- Izrada stranice na kojoj će se dohvaćeni podaci ispisati i na koju će se otići nakon uspješnog logina
 - Dodavanje gumbi edit pored svakog korisnika/prijave kako bi se mogla uređivati
 - Dodavanje gumba „prihvati prijavu“ kako bi serviser mogao prihvatiti prijavu
 - Dodavanje gumba „završi prijavu“ kako bi serviser mogao označiti popravak kao popravljen
 - Dodavanje gumba „prijavi popravak“ kako bi korisnik mogao popravak popraviti

4.FAZA

- Stvaranje dvije stranice za izradu prijave popravka
 - Na prvoj stranici se bira samo serviser
 - Na drugoj stranici se biraju ostali podaci: vrijeme dolaska, usluge, dodatne usluge i odabir želi li korisnik rezervno vozilo ili ne
- Na backendu stvoriti logiku koja će ovisno o tome je li serviser odabran ili ne, na drugoj stranici prikazati samo datume kada je odabran serviser slobodan ili kada auto servis radi ukoliko serviser nije odabran, u kojem će se slučaju na osnovu odabranog datuma nasumično odabrati serviser koji radi ujutro na taj datum
- Stvoriti metode za dohvat usluga i zamjenskih vozila koja bi se mogla prikazivati korisniku na odabir za prijavu popravka
- Pohrana odabranih metoda jedne prijave i ostalih podataka prijave u bazu podataka

5.FAZA

- Stvaranje backend logike za edit prijave
 - Dohvat odgovarajućih podataka iz baze podataka
 - Postaviti ograničenja da se datum dolaska i barem jedna usluga moraju odabrati i na stvaranju prijave i na editu prijave
 - Pohrana izmjena u bazu
- Stvaranje backend logike za edit korisnika
 - Svaki korisnik ima drugačiji skup podataka (neki su zajednički) stoga se ovisno o vrsti korisnika prikazuje drugačija stranica
 - Osigurati da se svi podaci moraju popuniti, osim lozinke koja ukoliko je prazna ostaje nepromijenjena
- Omogućavanje prihvaćanja prijave popravka od strane serviser i označavanja popravka da je popravljen
- Stvaranje stranice koja će ovisno o akciji koja se izvela ispisati poruku da je ta akcija uspješno izvedena i ponuditi korisniku mogućnost povratka na profil ili na početnu stranicu aplikacije

6.FAZA

- Izrada stranice za zaboravljenu lozinku
 - Obavezan unos pravilno formatirane lozinke
 - Backend provjera postoji li korisnik s danom lozinkom
- Izrada email podrške za aplikaciju
 - Email za zaboravljenu lozinku
 - Email za pregled podataka prijave popravka koju je serviser potvrdio
- Izrada opcije „uredi svoj profil“ koja dopušta promjenu vlastitih podataka

7.FAZA

- Objavljivanje aplikacije na server
 - Odabrati server koji podržava Spring Boot aplikaciju sa MySQL bazom podataka
 - Konfigurirati aplikaciju i bazu na serveru da funkcioniraju jednako dobro kao i u development fazi
- Dokumentirati ostvarene funkcionalnosti i aplikaciju općenito

Nakon gore definiranih faza stvorena je aplikacija koja zadovoljava funkcionalnosti navedene u zahtjevima te koja uspješno živi i radi na serveru te je spremna za korištenje.