Oblikovanje programske potpore

Ak. god. 2018./2019.

***Sustav za praćenje rada auto servisa***

Dokumentacija, Rev. *1*

Grupa: *BrainStorm*

Voditelj: *Tin Komerički*

Datum predaje: *28. studenog 2018.*

Nastavnik: *Dr. sc. Miljenko Krhen*

Contents

[1. Dnevnik promjena dokumentacije 1](#_Toc534977733)

[2. Opis projektnog zadatka 2](#_Toc534977734)

[3. Pojmovnik 4](#_Toc534977735)

[4. Funkcionalni zahtjevi 5](#_Toc534977736)

[4.1. Dionici 5](#_Toc534977737)

[4.2. Aktori 5](#_Toc534977738)

[4.2.1. Administrator 5](#_Toc534977739)

[4.2.2. Ovlašteni serviser 5](#_Toc534977740)

[4.2.3. Registrirani korisnik 5](#_Toc534977741)

[4.2.4. Neregistrirani korisnik 5](#_Toc534977742)

[4.2.5. Baza podataka 5](#_Toc534977743)

[4.3. Opisi obrazaca uporabe 6](#_Toc534977744)

[4.4. Dijagrami obrazaca uporabe 11](#_Toc534977745)

[4.4.1. Dijagram obrazaca uporabe za cijeli auto servis 11](#_Toc534977746)

[4.4.2. Dijagram obrazaca uporabe za administratora 12](#_Toc534977747)

[4.4.3. Dijagram obrazaca uporabe za korisnika 12](#_Toc534977748)

[4.4.4. Dijagram obrazaca uporabe za prijavu 13](#_Toc534977749)

[4.4.5. Dijagram obrazaca uporabe za servisera 13](#_Toc534977750)

[4.4.6. Dijagram obrazaca uporabe za obradu prijave vozila 14](#_Toc534977751)

[4.5. Sekvencijski dijagrami 15](#_Toc534977752)

[4.5.1. Osnovni upis podataka 15](#_Toc534977753)

[4.5.2. Potvrda prijave vozila na popravak 15](#_Toc534977754)

[4.5.3. Pregled vlastitih korisnika 16](#_Toc534977755)

[4.5.4. Prijava na sustav 16](#_Toc534977756)

[4.5.5. Prijava vozila na popravak 17](#_Toc534977757)

[4.5.6. Registracija 18](#_Toc534977758)

[4.5.7. Upravljanje informacijama registriranih korisnika 18](#_Toc534977759)

[5. Ostali zahtjevi 19](#_Toc534977760)

[6. Arhitektura i dizajn sustava 20](#_Toc534977761)

[6.1. Arhitekturni obrazac MVC 20](#_Toc534977762)

[6.2. ER model baze podataka 21](#_Toc534977763)

[6.2.1. Opisi entiteta ER modela baze podataka 22](#_Toc534977764)

[6.3. Dijagram razreda 25](#_Toc534977765)

[6.4. Dijagram objekata 26](#_Toc534977766)

[7. Implementacija i korisničko sučelje 27](#_Toc534977767)

[7.1. Dijagram razmještaja 27](#_Toc534977768)

[7.2. Dijagram stanja – prijava na sustav 28](#_Toc534977769)

[7.3. Dijagram stanja – izmjena podataka o korisniku 29](#_Toc534977770)

[7.4. Dijagram aktivnosti – registracija korisnika 30](#_Toc534977771)

[7.5. Dijagram aktivnosti – pregled korisnika 31](#_Toc534977772)

[8. Zaključak i budući rad 32](#_Toc534977773)

[9. Popis literature 33](#_Toc534977774)

[Dodatak A: Indeks (slika, dijagrama, tablica, ispisa kôda) 34](#_Toc534977775)

[Dodatak B: Dnevnik sastajanja 35](#_Toc534977776)

[Dodatak C: Prikaz aktivnosti grupe 36](#_Toc534977777)

[Dodatak D: Plan rada / Pregled rada i stanje ostvarenja 38](#_Toc534977778)

# 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 Janđel | 11.11.2018. |
| 0.6 | Dodani sekvencijski dijagrami i njihovi opisi, dodan opis pojmova | Filip Janđel  Tea Kalenski | 16.11.2018. |
| 0.7 | Dodan kratki opis arhitekture i tehnologije | Filip Janđel | 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 Janđel  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 Janđel  Filip Kovačević | 15.1.2019. |
| 2.0 | Ažurirani svi dodaci i dnevnik dokumentacije, dovršeno sedmo poglavlje, napisano osmo poglavlje | Dino Avdić  Filip Janđel  Tin Komerički  Filip Kovačević  Marko Prnić | 17.1.2019. |

# 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 servisera 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 servisera kod kojega se želi prijaviti. Odabir nije obavezan, i ukoliko korisnik ne odabere željenog servisera, sustav će mu sam dodijeliti slobodnog, u ovisnosti o vremenu dolaska na servis. Nakon odabira servisera, korisniku se prikazuje prikaz slobodnih termina odabranog servisera 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 servisera 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.

# 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 Microsytems

**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

**Jquery** – Javascript biblioteka dizajnirana da pojednostavi pisanje javascript frontend skripti

**Postgresql** – relacijska baza podataka

# Funkcionalni zahtjevi

## Dionici

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

## Aktori

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

### 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

### 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

### Registrirani korisnik

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

### Neregistrirani korisnik

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

### 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

## 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 servisera
* 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 pokreče 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 pokreče 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 pokreče 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 pokreče 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 pokreče prijavu vozila na popravak, sustav dohvaća potrebna polja koja Registrirani korisnik mora ispuniti
* Opis mogućih odstupanja: -

UC6.1

* Naziv: Odabir servisera
* Glavni sudionici: Registrirani korisnik
* Cilj: Registrirani korisnik odabire servisera
* Preduvjeti: Registracija, Prijava na sustav, Započeta prijava vozila na popravak
* Željeni scenarij: Registriranom korisniku sustav daje neobaveznu opciju izbora ovlaštenog servisera, 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: -

## Dijagrami obrazaca uporabe

### Dijagram obrazaca uporabe za cijeli auto servis



Slika 1. Dijagram obrazaca uporabe za cijeli auto servis

### Dijagram obrazaca uporabe za administratora



Slika 2. Dijagram obrazaca uporabe za administratora

### Dijagram obrazaca uporabe za korisnika



Slika 3. Dijagram obrazaca uporabe za korisnika

### Dijagram obrazaca uporabe za prijavu



Slika 4. Dijagram obrazaca uporabe za prijavu

### Dijagram obrazaca uporabe za servisera



Slika 5. Dijagram obrazaca uporabe za servisera

### Dijagram obrazaca uporabe za obradu prijave vozila



Slika 6. Dijagram obrazaca uporabe za obradu prijave vozila

## Sekvencijski dijagrami

### 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

### 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

### 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

### 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

### Prijava vozila na popravak

Registrirani korisnik pokreče prijavu vozila na popravak, unosi podatke o vozilu. Registriranom korisniku sustav daje neobaveznu opciju izbora ovlaštenog servisera, 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

### 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

### 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

# 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.

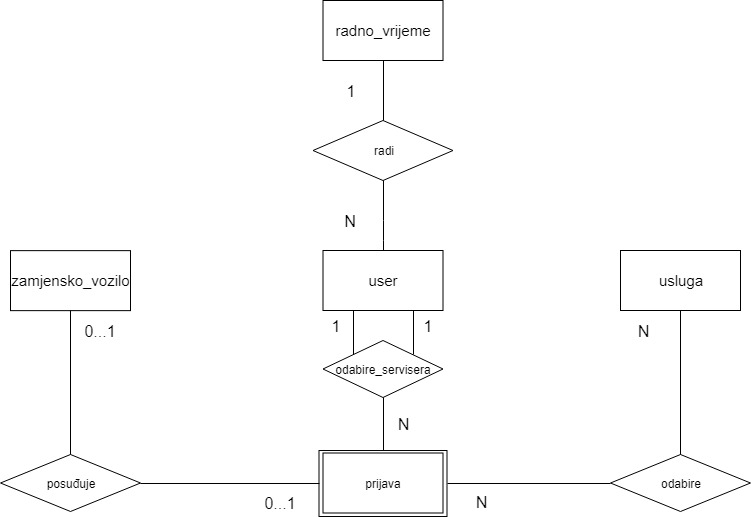
# Arhitektura i dizajn sustava

## 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 servisera. 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 arhutekturnog 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 komponenata.

## ER model baze podataka



Slika 14 ER model baze podataka

### 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 servisera

email – e-mail adresa korisnika

godina\_proizvodnje – godina proizvodnje vozila

id\_radnog\_vremena – identifikacijski broj radnog vremena servisera

last\_name – prezime određenog korisnika

name – ime određenog korisnika

password – 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 servisera u sustavu

id\_radnog\_vremena – identifikacijski broj radnog vremena servisera

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 servisera od strane registriranog korisnika

id\_servisera – identifikacijski broj izabranog servisera

user\_id – identifikacijski broj korisnika koji je odabrao određenog servisera

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 servisera

user\_id – identifikacijski broj korisnika koji je odabrao određenog servisera

**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\_zahtjevi – 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

### CREATE naredbe relacijskog modela baze podataka

CREATE TABLE user (

user\_id INT NOT NULL,

broj\_telefona VARCHAR(255),

email VARCHAR(255),

godina\_proizvodnje VARCHAR(255),

id\_radnog\_vremena VARCHAR(255),

last\_name VARCHAR(255),

name VARCHAR(255),

password VARCHAR(255),

registracija VARCHAR(255),

role VARCHAR(255),

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,

id\_korisnika VARCHAR(255),

tip\_vozila VARCHAR(255),

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\_zahtjevi VARCHAR(255),

id\_servisera INT,

reg\_zamjensko VARCHAR(255),

vrijeme\_dolaska DATE,

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),

cijena\_usluge INT,

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));

## Dijagram razreda



Slika 15. Dijagram razreda

## Dijagram objekata



Slika 16. Dijagram objekata

# Implementacija i korisničko sučelje

## Dijagram razmještaja



Slika 17. Dijagram razmještaja

## Dijagram stanja – prijava na sustav



Slika 18. Dijagram stanja – prijava na sustav

## Dijagram stanja – izmjena podataka o korisniku



Slika 19. Dijagram stanja – izmjena podataka o korisniku

## Dijagram aktivnosti – registracija korisnika



Slika 20. Dijagram aktivnosti – registracija korisnika

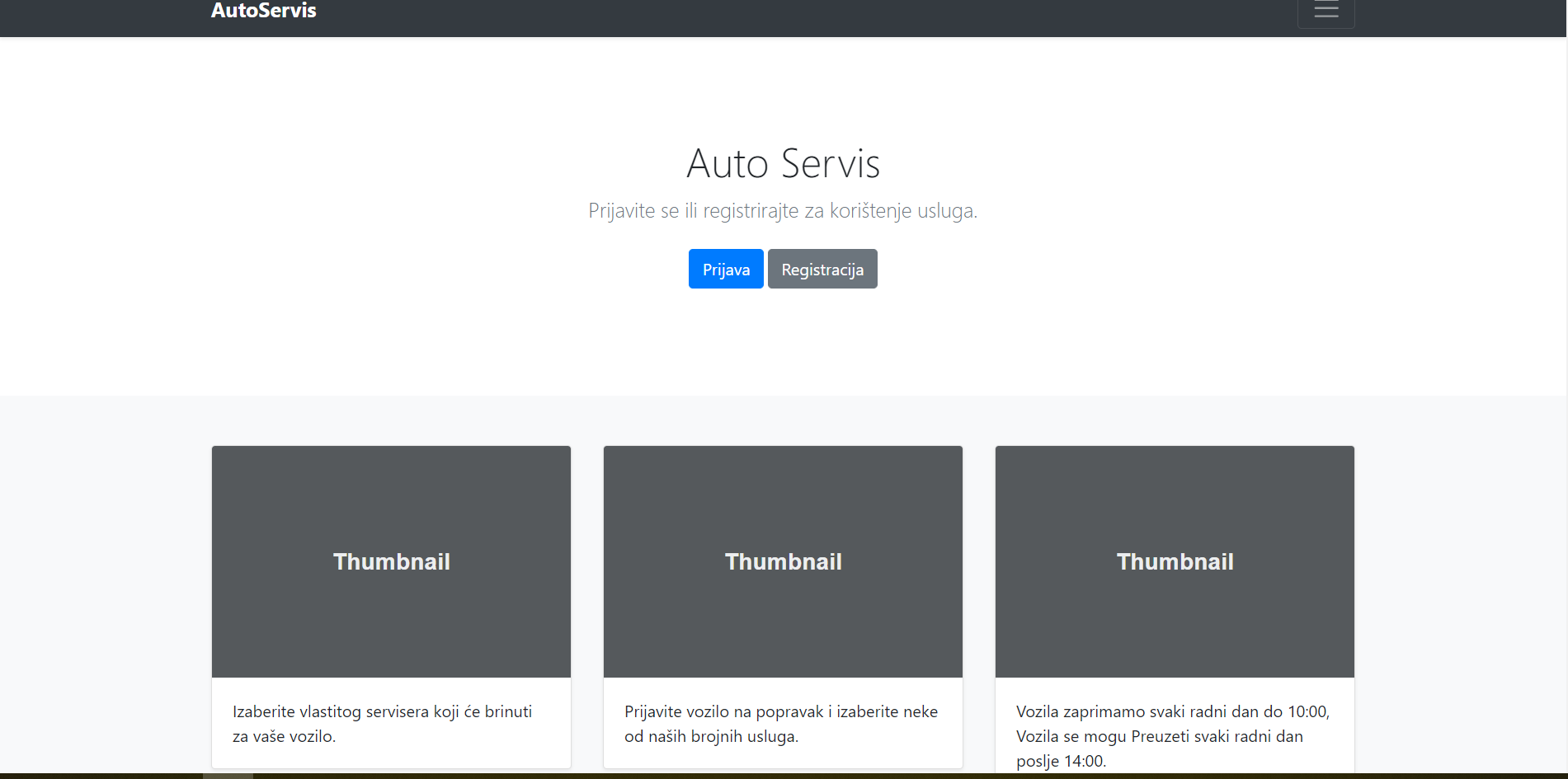
## Dijagram aktivnosti – pregled korisnika



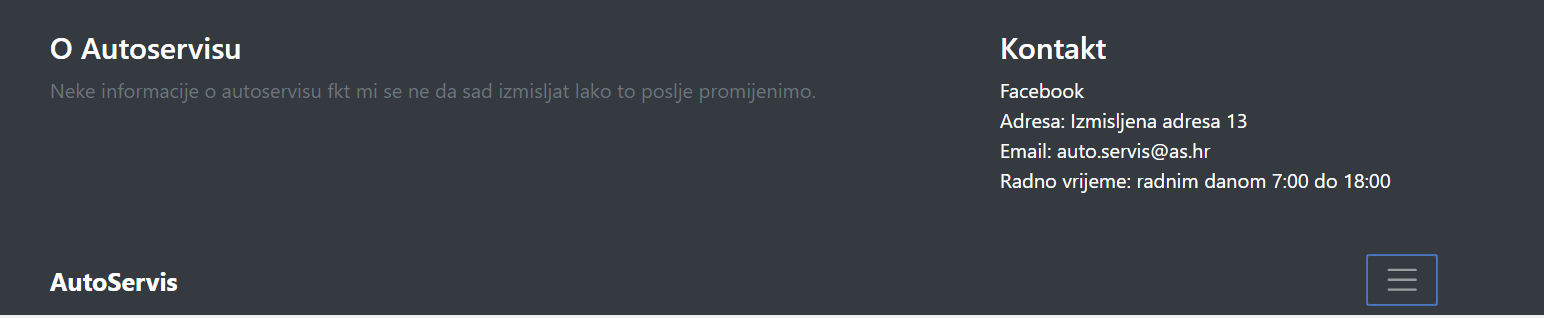
Slika 21. Dijagram aktivnosti – pregled korisnika

## Korisničke upute

**Početna stranica :**



Otvaranjem linka <https://gentle-river-64840.herokuapp.com/pocetna> , 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 :

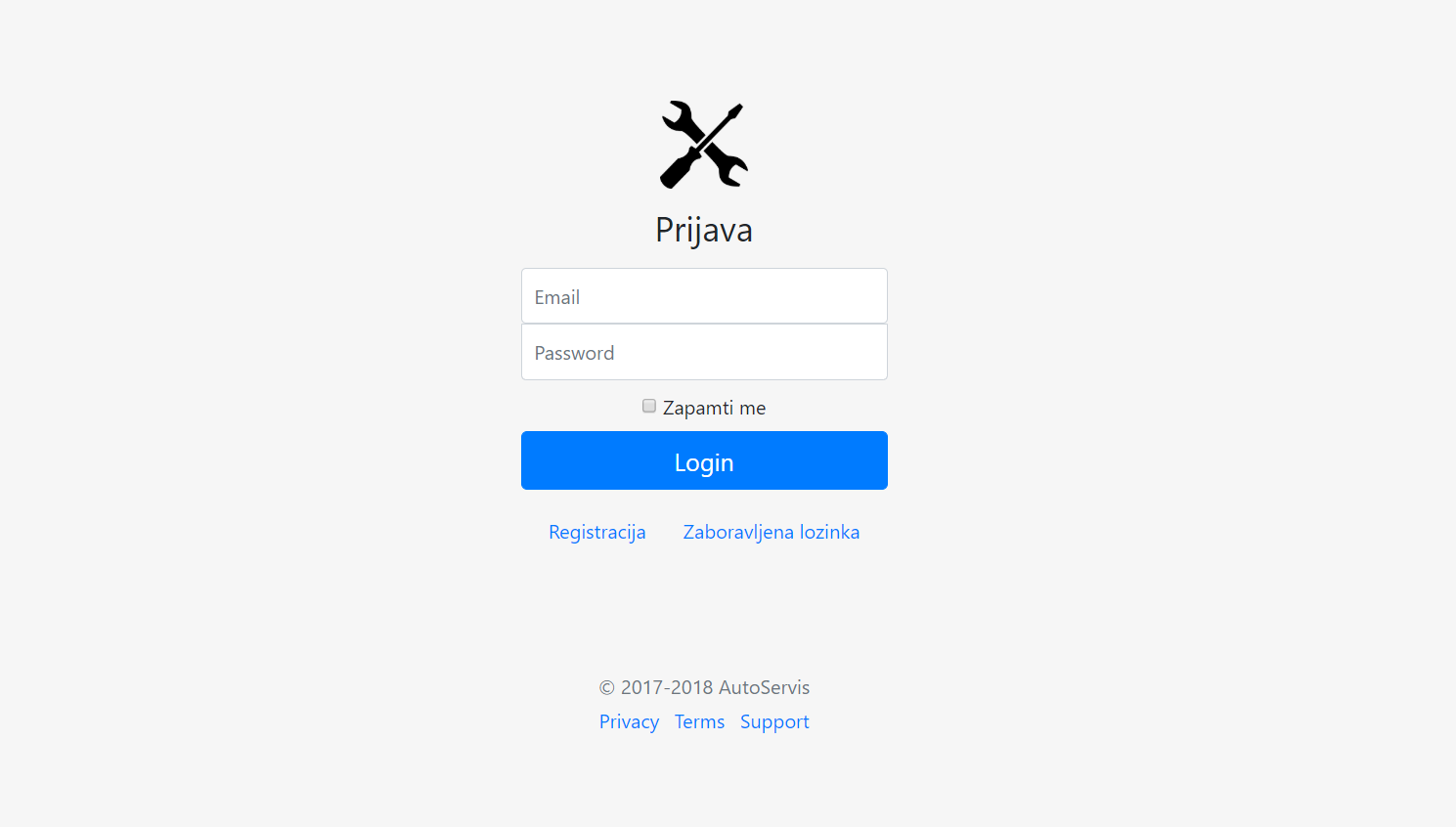


U sredini početne stranice, nalaze gumbi „Prijava“ i Registracija.

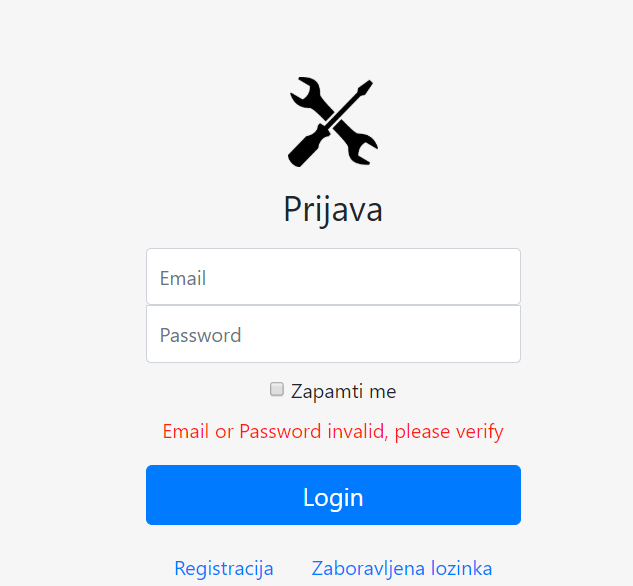
Klikom na gumb „Registracija“, korisnika će se preusmjeriti na stranicu za registraciju korisnika, a klikom na gumb „Prijava“ na stranicu za prijavu korisnika.

**Stranica za prijavu korisnika :**

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



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 :



## Isječak programskog koda vezan za temeljnu funkcionalnost sustava



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.

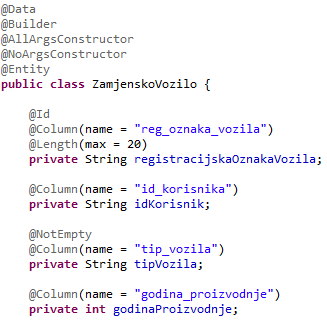
U donjem dijelu metode (**1**) se svi potrebni podaci prenose na frontend koji te podatke može pokazati u određenom formatu.



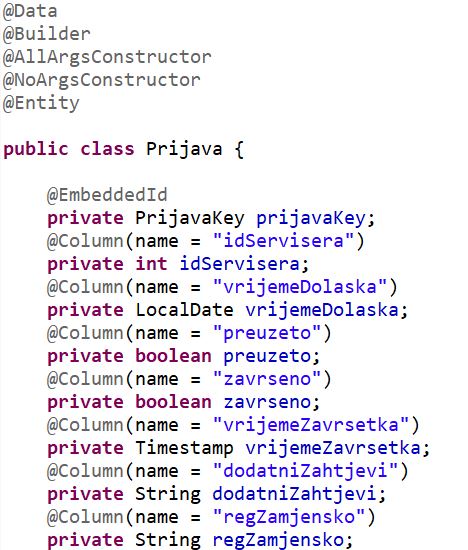
Ovdje je prikazana djelomična implementacija razreda koji je odgovoran za dohvaćanje, izmjenu i spremanje podataka u sustavu. Razred u sebi sadrži sve repozitorije podataka(**1**) i ovisno o zahtijevanoj akciji i ulozi korisnika, obavlja pojedinu operaciju. Također, razred sadrži objekt koji služi za kriptiranje lozinke (bCryptPasswordEncoder). Na primjer, prilikom registracije servisera 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 servisera u sustavu koji će se korisniku prikazati na izbor prilikom stvaranja prijave popravka svog automobila.



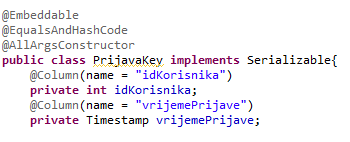
Jedna od ključnih funkcionalnosti ove aplikacije jest ograničenje pristupa pojedinim stranicama na temelju uloge korisnika u sustavu. Ovdje je prikazana implementacija tih ograničenja u skladu sa ulogama pojedinih korisnika sustava(admin, serviser, registrirani korisnik). Metoda permitAll() omogućuje bilo kome da pristupi danim resursima. Medota 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. Methoda 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.



Slika 1



Slika 2



Slika 3

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.

# 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 servisera 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. Stoga bi zadatak za budući rad bio počistiti redundantan kod, promijeniti imena varijablama 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.

# 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](#_Toc531593947)

[Slika 2. Dijagram obrazaca uporabe za administratora 12](#_Toc531593948)

[Slika 3. Dijagram obrazaca uporabe za korisnika 12](#_Toc531593949)

[Slika 4. Dijagram obrazaca uporabe za prijavu 13](#_Toc531593950)

[Slika 5. Dijagram obrazaca uporabe za servisera 13](#_Toc531593951)

[Slika 6. Dijagram obrazaca uporabe za obradu prijave vozila 14](#_Toc531593952)

[Slika 7. Sekvencijski dijagram - Osnovni upis podataka 15](#_Toc531593953)

[Slika 8. Sekvencijski dijagram - Potvrda prijave vozila na popravak 15](#_Toc531593954)

[Slika 9. Sekvencijski dijagram - Pregled vlastitih korisnika 16](#_Toc531593955)

[Slika 10. Sekvencijski dijagram - Prijava na sustav 16](#_Toc531593956)

[Slika 11. Sekvencijski dijagram - Prijava vozila na popravak 17](#_Toc531593957)

[Slika 12. Sekvencijski dijagram – Registracija 18](#_Toc531593958)

[Slika 13. Sekvencijski dijagram – Upravljanje informacijama registriranih korisnika 18](#_Toc531593959)

[Slika 14 ER model baze podataka 21](#_Toc531593960)

[Slika 15. Dijagram razreda 25](#_Toc531593961)

[Slika 16. Dijagram objekata 26](#_Toc531593962)

[Slika 17 Aktivnost grupe u grani master 33](#_Toc531593963)

[Slika 18 Aktivnost grupe u grani Doklumentacija 33](#_Toc531593964)

# Dodatak B: Dnevnik sastajanja

|  |  |  |
| --- | --- | --- |
| 21.10.2018. | Filip Janđel  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 Janđel  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 Janđel  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 Janđel  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 Janđel  Tea Kalenski  Tin Komerički  Filip Kovačević  Marko Prnić | Kratka rasprava o dosad napravljenoj 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 Janđel  Tea Kalenski  Tin Komerički  Filip Kovačević  Marko Prnić | Definirane faze projekta, definirana raspodjela poslova po članovima tima. |
| 20.12.2018. | Dino Avdić  Filip Janđel  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 Janđel  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 Janđel  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

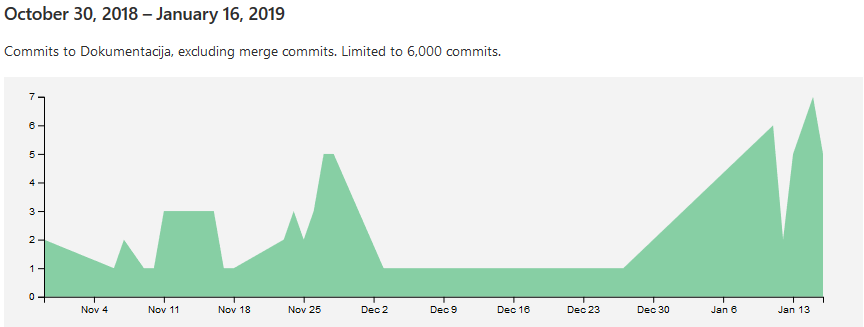
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Popis aktivnosti** | **Članovi grupe** | | | | | |
| Tin Komerički | Dino Avdić | Filip janđel | Tea Kalenski | Filip Kovačević | Marko Prnić |
| Opis zadatka i riječnik pojmova | 10% |  | 90% |  |  |  |
| Funkcionalni zahtjevi | 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 | 50% | 30% | 20% |  |  |  |
| 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% |  |  |  |



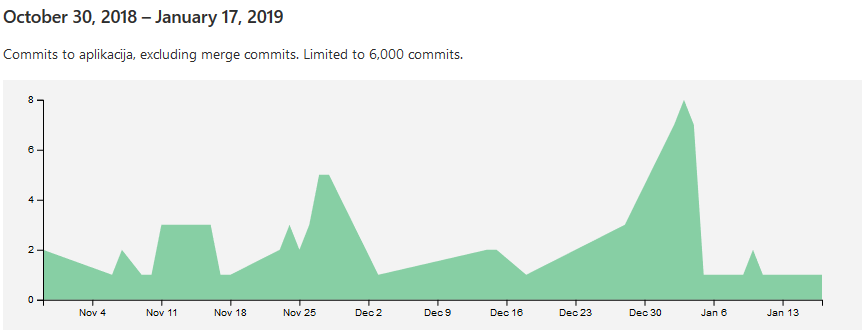
Slika 17 Aktivnost grupe u grani master



Slika 18 Aktivnost grupe u grani Doklumentacija za prvu verziju dokumentacije



Slika 19 Aktivnost grupe u grani Dokumentacija za drugu verziju dokumentacije



Slika 20 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 spremiti podatke o adminu
* Izrada stranice za registraciju servisera (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

* Staranje backend logike za edit prijava
  + 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 prijava
  + 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 prijava popravka od strane servisera 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.