

Programsko inženjerstvo ak.god 2024./2025

Sveučilište u Zagrebu

Fakultet elektrotehnike i računarstva

Auto Servis

Tim: <TG05.3>

REST-API

Nastavnik: Vlado Sruk

1.1 Uvod

Projekt **Aplikacija za auto servis** razvija se u sklopu kolegija *Programsko inženjerstvo* s ciljem digitalizacije poslovanja i komunikacije između korisnika (vlasnika vozila) i servisera.

Tradicionalno, proces prijave vozila na servis uključuje telefonske pozive, ručne evidencije i nejasne rokove, što dovodi do čestih nesporazuma, kašnjenja i neučinkovitosti.

Ovaj projekt predviđa razvoj web aplikacije koja omogućuje jednostavnu prijavu, pregled statusa popravka i elektroničku komunikaciju između korisnika i osoblja servisa.

1.2 Cilj projekta

Glavni cilj projekta je **automatizirati i pojednostaviti proces prijave i obrade vozila u autoservisima**.

Sustav će omogućiti:

- prijavu vozila putem internetskog sučelja
- odabir termina servisa
- automatsko obavještavanje korisnika o promjenama statusa prijave
- administrativno praćenje podataka o vozilima, terminima i servisima

Time se postiže transparentnost procesa i smanjuje potreba za ručnom komunikacijom i papirologijom.

1.3 Problematika postojećeg sustava

Trenutni način poslovanja u većini autoservisa temelji se na:

- ručnoj evidenciji vozila
- telefonskim dogovorima s korisnicima
- nedostatku uvida u zauzetost servisera
- nepostojanju sustava za automatske obavijesti

Ovakav sustav uzrokuje česte pogreške, preklapanje termina, gubitak informacija i nezadovoljstvo korisnika.

Osim toga, korisnici nemaju mogućnost praćenja statusa servisa u stvarnom vremenu niti elektroničke evidencije svojih prijava.

1.4 Opis predloženog rješenja

Predloženo rješenje je centralizirana web aplikacija koja povezuje sve sudionike – korisnike, servisere i administratore.

Sustav omogućuje:

- **registraciju i autentifikaciju korisnika** putem OAuth2 (Google prijava)
- **unos i uređivanje podataka o vozilu**
- **prijavu vozila na servis i odabir termina**
- **automatsko obavještavanje e-mailom** o promjenama statusa prijave
- **uvid u povijest servisa** i mogućnost preuzimanja izvještaja
- **administratorima** generiranje izvještaja (PDF, XML, XLSX) i praćenje statistike servisa

1.5 Potencijalna korist projekta

Implementacija ovakvog sustava donosi višestruke koristi:

Korisnik	Dobiva
Vlasnik vozila	Brzu prijavu i uvid u status servisa bez potrebe za pozivima.
Serviser	Lakšu organizaciju posla, pregled svih prijava i termina, mogućnost dodavanja napomena.
Administrator / vlasnik servisa	Digitalne evidencije, praćenje opterećenosti servisera i generiranje izvještaja.
Cjelokupna organizacija	Veća učinkovitost, manje pogrešaka i veće zadovoljstvo korisnika.

Projekt donosi i širu društvenu korist – modernizaciju poslovanja i smanjenje nepotrebne papirologije u autoindustriji.

1.6 Postojeća slična rješenja

Na tržištu postoje aplikacije koje korisnicima omogućuju praćenje održavanja vozila i servisnih aktivnosti. Primjeri takvih rješenja su:

Naziv rješenja	Opis	Razlika u odnosu na ovaj projekt
CARFAX	Aplikacija za praćenje povijesti vozila, servisnih intervala i podsjetnika na održavanje.	Namijenjena samo korisnicima; nema prijavu vozila na servis ni ulogu servisera.
Car Care	Aplikacija za vođenje evidencije troškova, goriva i održavanja više vozila.	Ne podržava rezervacije termina ni komunikaciju s autoservisom.
Drivvo	Aplikacija za vođenje evidencije troškova, goriva i održavanja više vozila.	Ne podržava rezervacije termina ni komunikaciju s autoservisom.

1.7 Skup korisnika

Aplikaciju bi koristili:

- **privatni korisnici** – vlasnici osobnih automobila koji žele jednostavno prijaviti vozilo na servis
- **serviseri i tehničari** – koji vode i obrađuju prijave, planiraju radni raspored
- **administratori / vlasnici servisa** – za nadzor rada, izvještavanje i analitiku

Potencijalno bi se sustav mogao prilagoditi i za:

- **autopraonice**
- **vulkanizerske radionice**
- **sisteme najma vozila** – gdje bi se koristila funkcionalnost rezervacije zamjenskog vozila

1.8 Mogućnost prilagodbe i proširenja

Zbog modularne arhitekture sustava, moguće su sljedeće nadogradnje:

- dodavanje **notifikacija putem SMS-a ili mobilne aplikacije**
 - integracija s **vanjskim ERP sustavima** za fakturiranje
 - statističke analize i praćenje učinkovitosti servisera
 - proširenje sustava na **mobilnu aplikaciju (Android/iOS)**
-

1.9 Opseg projektnog zadatka

Trenutna verzija projekta uključuje sljedeće funkcionalnosti:

- prijava i autentifikacija korisnika (OAuth2)
 - unos i upravljanje vozilima
 - prijava vozila na servis i upravljanje terminima
 - ažuriranje statusa servisa i dodavanje napomena
 - automatsko obavještavanje korisnika putem e-maila
 - generiranje i preuzimanje izvještaja
-

1.10 Moguće nadogradnje

Projekt se može proširiti u sljedećim smjerovima:

- **mobilna aplikacija** za korisnike i servisere
- **sustav za ocjenjivanje** kvalitete servisa i servisera
- **automatsko prepoznavanje vozila** putem registrske oznake
- **AI prediktivna analiza** za planiranje kapaciteta servisa
- **integracija s IoT uređajima** (npr. pametni senzori vozila za dijagnostiku)

U ovom poglavlju definirani su svi **funkcionalni** i **nefunkcionalni zahtjevi** sustava *Aplikacija za auto servis*, kao i **dionici (akteri)** koji sudjeluju u njegovu radu.

Ova analiza predstavlja temelj za daljnju specifikaciju sustava i izradu obrazaca uporabe.

Funkcionalni zahtjevi

ID zahtjeva	Opis	Prioritet	Izvor	Kriterij prihvaćanja
F-001	Sustav omogućuje korisniku prijavu u sustav.	Visok	Zahtjev dionika	Korisnik se može prijaviti i pristupiti svom profilu.
F-002	Sustav omogućuje korisniku dodavanje vozila (marka, model, registracija, godina).	Visok	Zahtjev dionika	Vozilo se sprema u bazu i vidljivo je korisniku.
F-003	Korisnik može prijaviti vozilo na servis te odabrati termin.	Visok	Zahtjev dionika	Prijava se pohranjuje i korisnik dobiva potvrdu e-mailom.
F-004	Sustav omogućuje korisniku pregled statusa prijave.	Srednji	Zahtjev dionika	Korisnik vidi trenutni status svake svoje prijave.
F-005	Serviser može pregledati zaprimljene prijave.	Visok	Zahtjev dionika	Prijave su vidljive u listi i filtrirane po statusu.
F-006	Serviser može ažurirati status prijave (zaprimljeno, u obradi, završeno, odgođeno).	Visok	Zahtjev dionika	Status se promjeni i korisniku se automatski pošalje e-mail.
F-007	Serviser može dodati napomenu na prijavu.	Srednji	Zahtjev dionika	Napomena se prikazuje unutar detalja prijave.
F-008	Sustav omogućuje korisniku rezervaciju zamjenskog vozila.	Srednji	Povratne informacije korisnika	Zamjensko vozilo se rezervira i označava kao nedostupno u sustavu.
F-009	Administrator može generirati izvještaje o servisima u PDF, XML ili XLSX formatu.	Srednji	Dokument projektnog zadatka	Izvještaj se uspješno generira i može se preuzeti.
F-010	Sustav omogućuje korisnicima pregled osnovnih informacija o servisu.	Srednji	Zahtjev dionika	Svi korisnici mogu otvoriti javnu stranicu s prikazom informacija o servisu.
F-011	Sustav generira i pohranjuje PDF obrazac pri predaji i preuzimanju vozila (s potpisima korisnika i servisera).	Visok	Dokument projektnog zadatka	Prilikom predaje ili preuzimanja vozila generira se PDF s podacima o korisniku, vozilu i serviseru.

ID zahtjeva	Opis	Prioritet	Izvor	Kriterij prihvaćanja
F-012	Sustav automatski šalje podsjetnik korisniku ako je termin odgođen za više od 3 dana.	Srednji	Dokument projektnog zadatka	Nakon promjene termina s pomakom većim od 3 dana, korisnik automatski prima e-mail obavijest.
F-013	Administrator može upravljati podacima o servisima (naziv, lokacija, radno vrijeme).	Srednji	Zahtjev dionika	Promjene se pohranjuju i odmah vidljive na korisničkom sučelju.
F-014	Administrator može dodavati, uređivati i brisati servisere te ih povezivati s određenim servisima.	Visok	Zahtjev dionika	Novi serviseri se uspješno dodaju i prikazuju u sustavu.
F-015	Administrator može pregledavati i deaktivirati korisničke račune.	Srednji	Zahtjev dionika	Korisnik se nakon deaktivacije više ne može prijaviti.

Nefunkcionalni zahtjevi

ID zahtjeva	Opis	Prioritet
NF-001	Sustav mora koristiti siguran OAuth2 protokol za autentikaciju korisnika.	Visok
NF-002	Sustav mora osigurati komunikaciju putem HTTPS protokola.	Visok
NF-003	Sustav mora omogućiti odziv stranice unutar 2 sekunde.	Srednji
NF-004	Sustav mora omogućiti rad na mobilnim uređajima (responsivni dizajn).	Srednji
NF-005	Sustav mora biti implementiran s relacijskom bazom podataka (PostgreSQL).	Visok
NF-006	Sustav mora imati jednostavno održavanje i dobro dokumentiran kôd.	Srednji
NF-007	Sustav mora automatski slati e-mail obavijesti korisnicima kod promjena statusa prijave i odgoda termina duljih od tri dana.	Visok
NF-008	Sustav mora biti dostupan 24/7 uz pouzdanost veću od 99%.	Srednji
NF-009	Sustav mora omogućiti prikaz lokacije auto servisa korištenjem Google Maps servisa.	Srednji

Dionici i aktori

U sustavu *Aplikacija za auto servis* sudjeluju sljedeći dionici (akteri):

Oznaka	Naziv aktera	Uloga	Funkcionalnosti (povezani zahtjevi)
--------	--------------	-------	-------------------------------------

Oznaka	Naziv aktera	Uloga	Funkcionalnosti (povezani zahtjevi)
A-1	Neregistrirani korisnik	Inicijator	Može pregledati osnovne informacije o servisu (F-011). Može se registrirati (F-001).
A-2	Registrirani korisnik	Inicijator	Može dodavati vozila (F-003), prijaviti ih na servis (F-004), pregledati statuse (F-005), rezervirati zamjensko vozilo (F-009) te preuzeti PDF obrazac pri predaji i preuzimanju vozila (F-012).
A-3	Serviser	Sudionik	Može pregledavati prijave (F-006), mijenjati statuse (F-007), dodavati napomene (F-008) i generirati PDF obrasce pri predaji i preuzimanju vozila (F-012).
A-4	Administrator	Sudionik	Može upravljati servisima (F-014), serviserima (F-015), korisnicima (F-016) i generirati izvještaje (F-010).
A-5	E-mail sustav	Vanjski sustav	Automatski šalje potvrde i obavijesti korisnicima (F-004, F-007, F-013, NF-007).
A-6	OAuth2 servis (Google)	Vanjski servis	Omogućuje autentikaciju korisnika (F-001, NF-001).
A-7	Google Maps servis	Vanjski servis	Omogućuje prikaz lokacije servisa (NF-009).

Zaključak

Definirani funkcionalni i nefunkcionalni zahtjevi predstavljaju temelj za daljnju izradu specifikacije sustava. Njima su jasno obuhvaćene sve osnovne i napredne funkcionalnosti aplikacije — od prijave korisnika, prijave vozila i upravljanja servisima do generiranja izvještaja i automatskog obavještavanja korisnika.

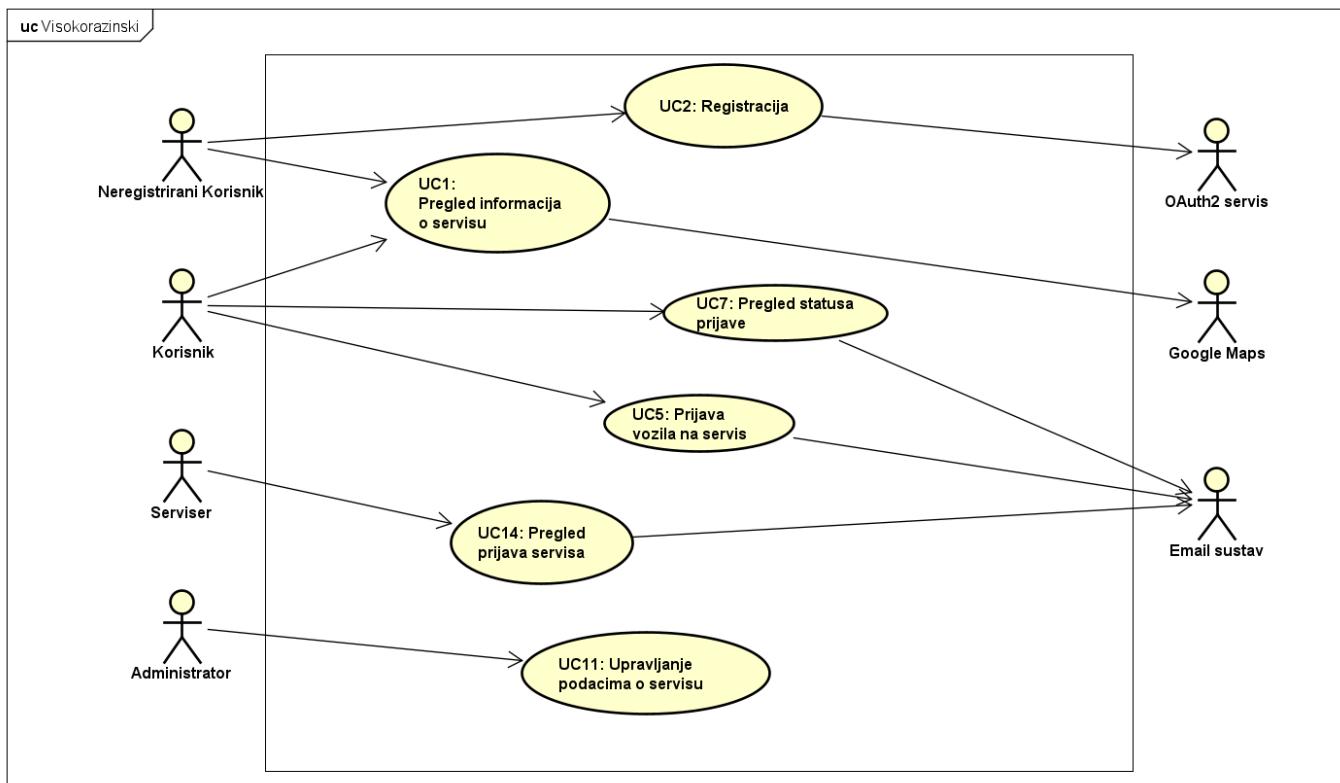
U ovom poglavlju prikazani su obrasci uporabe (Use Case) i njihovi opisi, koji detaljno prikazuju način na koji korisnici, serviseri, administratori i vanjski sustavi međusobno komuniciraju sa sustavom.

Napomena: Use-case dijagrami ne prikazuju internu arhitekturu sustava, no svi obrasci uporabe komuniciraju s bazom podataka radi spremanja i dohvaćanja podataka.

3.1 Dijagrami obrazaca uporabe

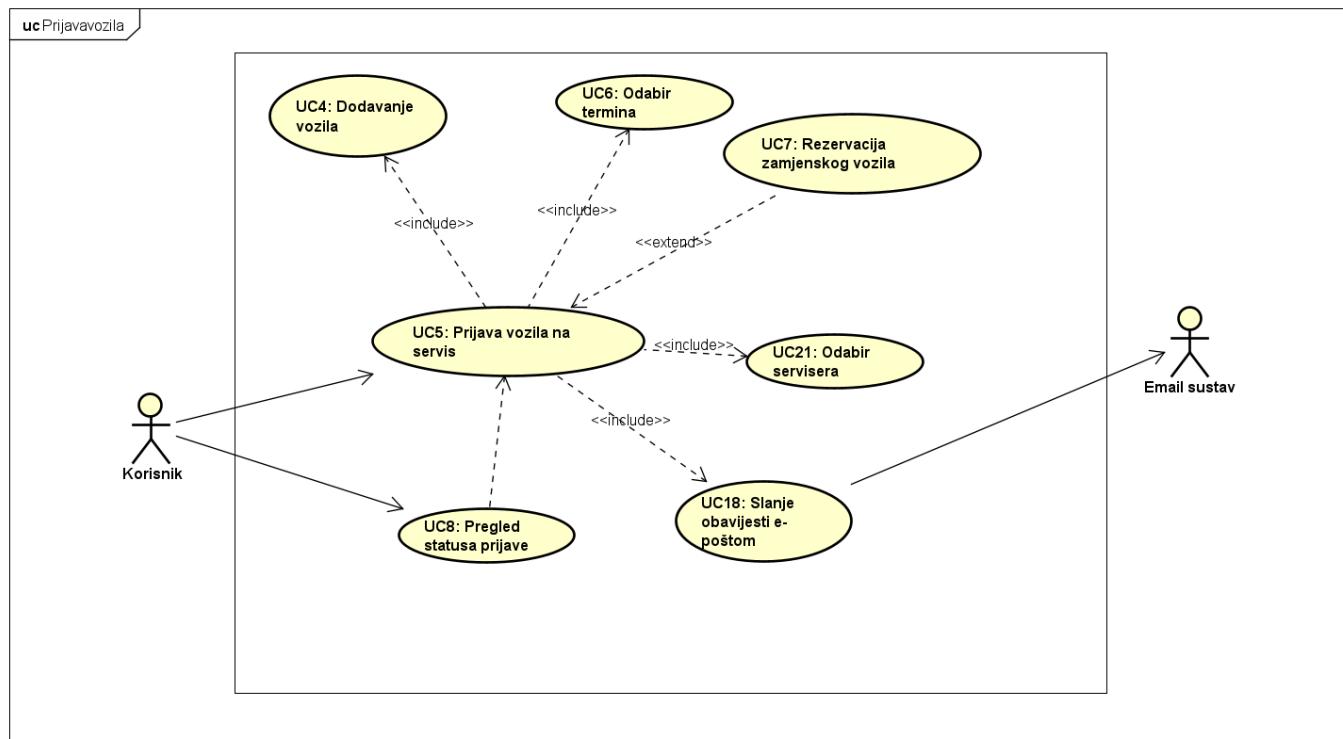
Visokorazinski dijagram

Prikazuje najvažnije funkcionalnosti sustava i sve glavne aktere na jednoj visokoj razini.



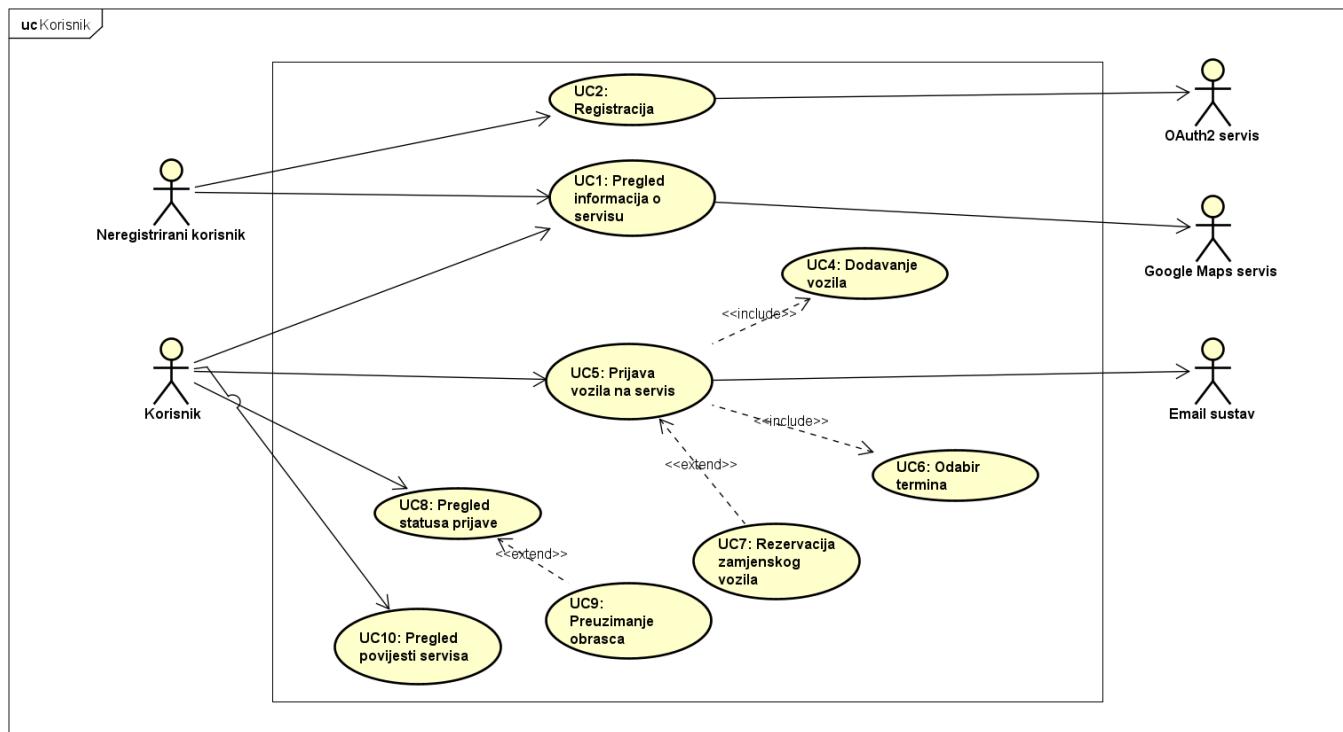
Dijagram ključne značajke – Prijava vozila

Prikazuje detaljan tijek prijave vozila i sve povezane obrasce uporabe.



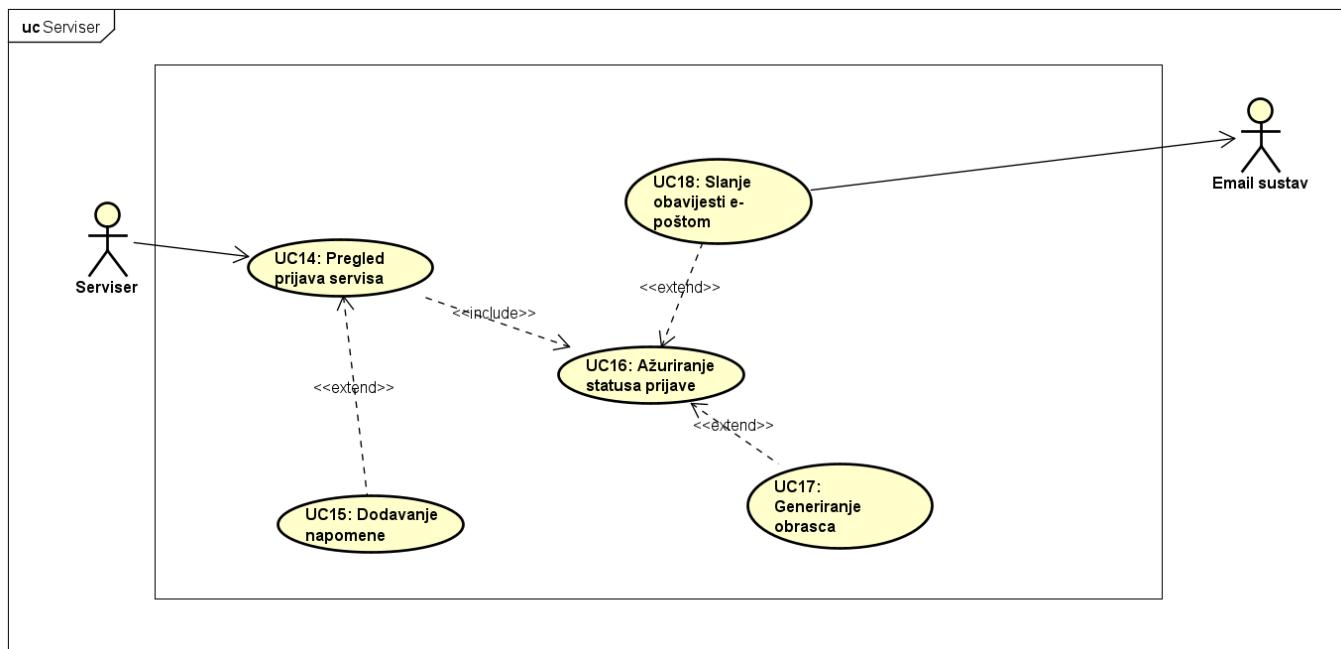
Dijagram po ulozi – Korisnik

Prikazuje sve funkcionalnosti dostupne registriranom i neregistriranom korisniku.



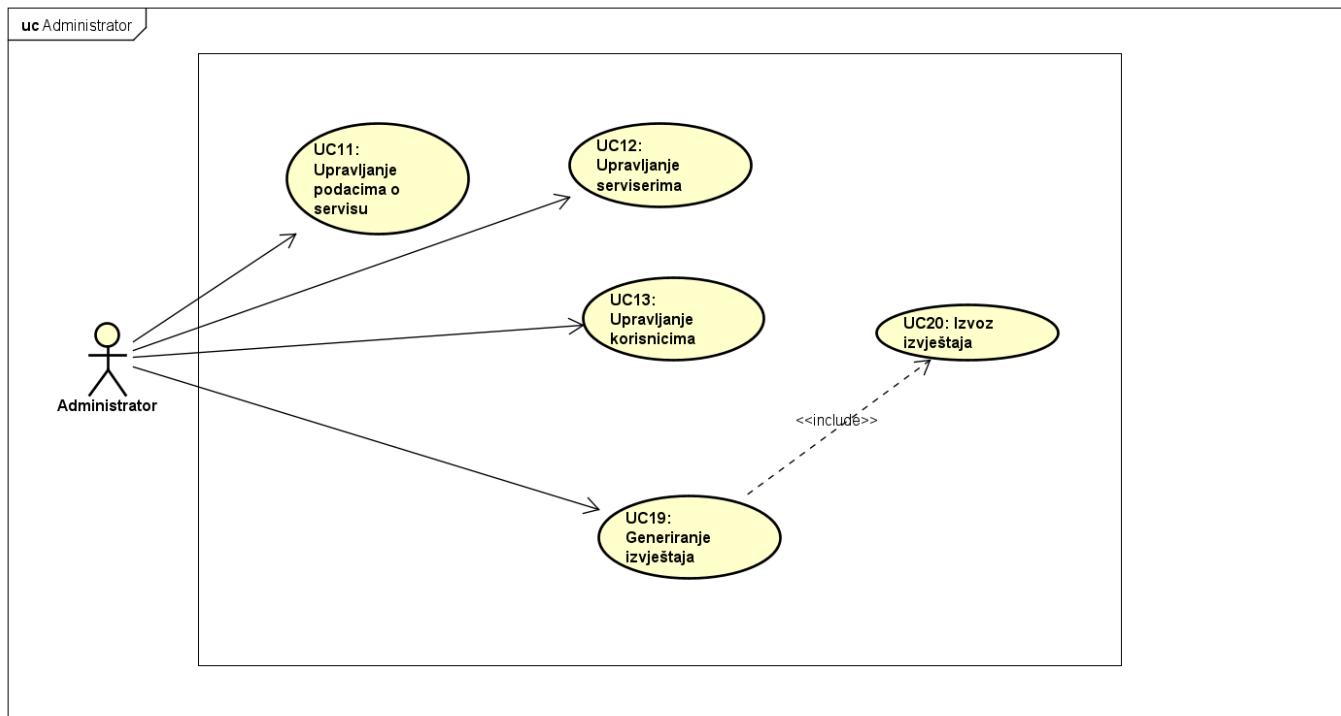
Dijagram po ulozi – Serviser

Prikazuje funkcionalnosti servisera u procesu obrade prijava.



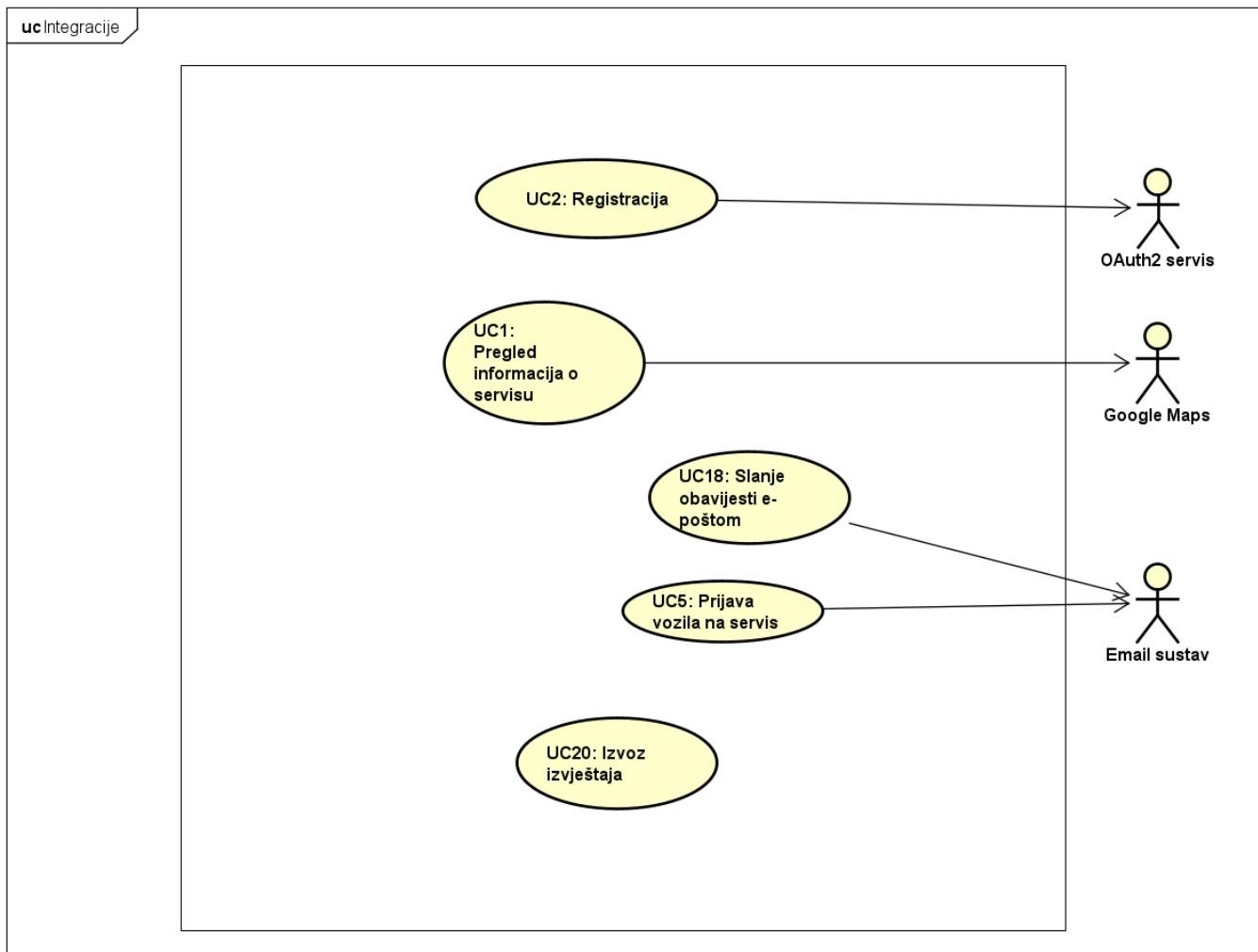
Dijagram po ulozi – Administrator

Prikazuje administrativne funkcionalnosti i upravljanje sustavom.



Dijagram integracija

Prikazuje interakcije sustava s vanjskim servisima poput OAuth2, e-mail sustava i Google Mapsa.



3.2 Opis obrazaca uporabe

U nastavku su prikazani detaljni opisi svih obrazaca uporabe.

Svaki opis sadrži glavnog sudionika, cilj, sudionike, preduvjete, osnovni tijek, moguća odstupanja i poveznice na funkcionalne zahtjeve (F-ID).

UC1 – Pregled osnovnih informacija o servisu

Glavni sudionik: Neregistrirani korisnik

Cilj: Pregled osnovnih informacija o servisu (radno vrijeme, lokacija, kontakt).

Sudionici: Google Maps servis

Preduvjet: Nema (korisnik nije prijavljen).

Osnovni tijek:

1. Korisnik odabire opciju „O servisu“. (F-011)
2. Sustav prikazuje osnovne informacije o servisu.
3. Sustav prikazuje točnu lokaciju servisa putem Google Maps servisa. (NF-009)

Odstupanja:

- Neuspješno dohvaćanje podataka s Google Mapsa → prikazuje se poruka o pogrešci.

UC2 – Registracija korisnika putem OAuth2

Glavni sudionik: Neregistrirani korisnik

Cilj: Stvaranje korisničkog računa pomoću vanjskog OAuth2 servisa.

Sudionici: OAuth2 servis

Preduvjet: Korisnik nije registriran.

Osnovni tijek:

1. Korisnik odabire opciju „Registracija“. (F-001)
2. Sustav preusmjerava korisnika na OAuth2 servis (Google).
3. Korisnik se autentificira i autorizira pristup.
4. Sustav pohranjuje korisničke podatke i potvrđuje registraciju.

Odstupanja:

- Autentifikacija neuspješna → prikazuje se poruka o neuspjeloj registraciji.
-

UC4 – Dodavanje vozila

Glavni sudionik: Registrirani korisnik

Cilj: Dodavanje novog vozila u sustav.

Preduvjet: Korisnik mora biti prijavljen.

Osnovni tijek:

1. Korisnik odabire „Dodaj vozilo“. (F-003)
2. Unosi podatke (marka, model, registracija, godina).
3. Sustav provjerava jedinstvenost registracije.
4. Sustav spremi vozilo u bazu i prikazuje potvrdu.

Odstupanja:

- Unesena registracija već postoji → prikazuje se poruka o pogrešci.
-

UC5 – Prijava vozila na servis

Glavni sudionik: Registrirani korisnik

Cilj: Prijaviti vozilo na servis i odabrati termin.

Sudionici: E-mail sustav

Preduvjet: Korisnik mora imati registrirano vozilo.

Osnovni tijek:

1. Korisnik odabire „Prijava vozila“. (F-004)
2. Sustav prikazuje popis vozila korisnika.
3. Korisnik odabire termin (UC6).
4. Po potrebi rezervira zamjensko vozilo (UC7).
5. Sustav spremi prijavu i šalje potvrdu e-mailom. (F-004, NF-007)

Odstupanja:

- Nema slobodnih termina → sustav nudi druge datume.
- Nema dostupnih zamjenskih vozila → prikazuje se obavijest.

UC6 – Odabir termina

Glavni sudionik: Korisnik

Cilj: Odabrati slobodan termin za servis.

Preduvjet: U tijeku je prijava vozila (UC5).

Osnovni tijek:

1. Sustav prikazuje sve slobodne termine. (F-004)
2. Korisnik odabire željeni termin.
3. Sustav sprema termin u prijavu.

Odstupanja:

- Termin više nije dostupan → sustav nudi drugi.
-

UC7 – Rezervacija zamjenskog vozila

Glavni sudionik: Korisnik

Cilj: Rezervirati zamjensko vozilo tijekom servisa.

Preduvjet: U tijeku je prijava vozila (UC5).

Osnovni tijek:

1. Korisnik odabire opciju „Rezerviraj zamjensko vozilo“. (F-009)
2. Sustav prikazuje dostupna zamjenska vozila.
3. Korisnik odabire vozilo i potvrđuje rezervaciju.

Odstupanja:

- Nema dostupnih vozila → prikazuje se poruka.
-

UC8 – Pregled statusa prijave

Glavni sudionik: Korisnik

Cilj: Pregled trenutnog statusa svih prijava.

Preduvjet: Korisnik mora imati barem jednu prijavu.

Osnovni tijek:

1. Korisnik otvara „Moje prijave“. (F-005)
 2. Sustav prikazuje statuse (zaprimljeno, u obradi, završeno, odgođeno).
 3. Ako je status „završeno“, korisnik može preuzeti PDF (UC9).
-

UC9 – Preuzimanje obrasca (PDF)

Glavni sudionik: Korisnik

Cilj: Preuzeti PDF obrazac o predaji ili preuzimanju vozila.

Preduvjet: Servis mora biti završen.

Osnovni tijek:

1. Korisnik odabire završenu prijavu. (F-012)

2. Sustav dohvaća već generirani PDF iz prethodnog procesa (UC17).

3. Korisnik preuzima dokument.

Odstupanja:

- Greška pri generiranju → prikazuje se poruka o neuspjelom preuzimanju.
-

UC10 – Povijest servisa

Glavni sudionik: Korisnik

Cilj: Pregled povijesti prijava i popravaka.

Preduvjet: Postoji barem jedna završena prijava.

Osnovni tijek:

1. Korisnik otvara „Povijest servisa“. (F-005)

2. Sustav prikazuje sve završene prijave.

UC11 – Upravljanje podacima o servisu

Glavni sudionik: Administrator

Cilj: Dodavanje, uređivanje i brisanje podataka o servisu.

Preduvjet: Administrator prijavljen.

Osnovni tijek:

1. Administrator otvara „Postavke servisa“. (F-014)

2. Uređuje ili dodaje podatke.

3. Sustav spremi promjene u bazu.

UC12 – Upravljanje serviserima

Glavni sudionik: Administrator

Cilj: Dodavanje, uređivanje i brisanje servisera te njihovo povezivanje sa servisima.

Preduvjet: Administrator prijavljen.

Osnovni tijek:

1. Administrator otvara „Serviseri“. (F-015)

2. Pregledava, dodaje i povezuje servisere.

3. Sustav spremi promjene.

UC13 – Upravljanje korisnicima

Glavni sudionik: Administrator

Cilj: Pregled i deaktivacija korisničkih računa.

Preduvjet: Administrator prijavljen.

Osnovni tijek:

1. Administrator otvara „Korisnici“. (F-016)

2. Pregledava popis korisnika.

3. Deaktivira odabranog korisnika.

Odstupanja:

- Korisnik ima aktivne prijave → prikazuje se upozorenje.
-

UC14 – Pregled prijava (serviser)

Glavni sudionik: Serviser

Cilj: Pregled svih prijava dodijeljenih serviseru. (F-006)

Osnovni tijek:

1. Serviser otvara „Prijave servisa“.
 2. Sustav prikazuje dodijeljene prijave.
 3. Serviser pregledava detalje i statuse.
-

UC15 – Dodavanje napomene

Glavni sudionik: Serviser

Cilj: Dodavanje napomene na prijavu. (F-008)

Preduvjet: Otvorena prijava (UC14).

Osnovni tijek:

1. Serviser odabire opciju „Dodaj napomenu“.
 2. Unosi opis radnje.
 3. Sustav spremi napomenu.
-

UC16 – Ažuriranje statusa prijave

Glavni sudionik: Serviser

Cilj: Promjena statusa prijave. (F-007)

Osnovni tijek:

1. Serviser otvara prijavu.
 2. Odabire novi status (zaprimljeno, u obradi, završeno, odgođeno).
 3. Sustav spremi promjenu i pokreće UC17 i UC18.
-

UC17 – Generiranje obrasca (PDF)

Glavni sudionik: Sustav

Sekundarni sudionik: Serviser

Cilj: Automatsko generiranje PDF obrasca o predaji/preuzimanju vozila. (F-012)

Osnovni tijek:

1. Sustav prepoznaje završenu prijavu.
 2. Generira PDF obrazac s podacima o korisniku, vozilu i serviseru.
 3. Sprema ga u tablicu obrazac.
-

UC18 – Slanje obavijesti e-poštom

Glavni sudionik: Sustav

Sekundarni sudionik: E-mail sustav

Cilj: Obavijestiti korisnika o promjeni statusa prijave ili odgodi. (F-013, NF-007)

Osnovni tijek:

1. Sustav prepoznae promjenu statusa.
 2. Ako je odgoda veća od 3 dana, šalje e-mail podsjetnik.
 3. E-mail sustav isporučuje poruku.
-

UC19 – Generiranje izvještaja

Glavni sudionik: Administrator

Cilj: Izrada izvještaja o poslovanju i prijavama. (F-010)

Osnovni tijek:

1. Administrator odabire parametre (vremenski raspon, tip).
 2. Sustav dohvaća podatke i generira izvještaj.
-

UC20 – Izvoz izvještaja

Glavni sudionik: Administrator

Cilj: Izvoz izvještaja u PDF, XML ili XLSX format. (F-010)

Preduvjet: Izvještaj mora biti generiran.

Osnovni tijek:

1. Administrator odabire format izvoza.
 2. Sustav generira datoteku i omogućuje preuzimanje.
-

UC21 – Odabir servisera

Glavni sudionik: Korisnik

Cilj: Odabir dostupnog servisera tijekom prijave vozila.

Preduvjet: U tijeku je proces prijave vozila (UC5).

Osnovni tijek:

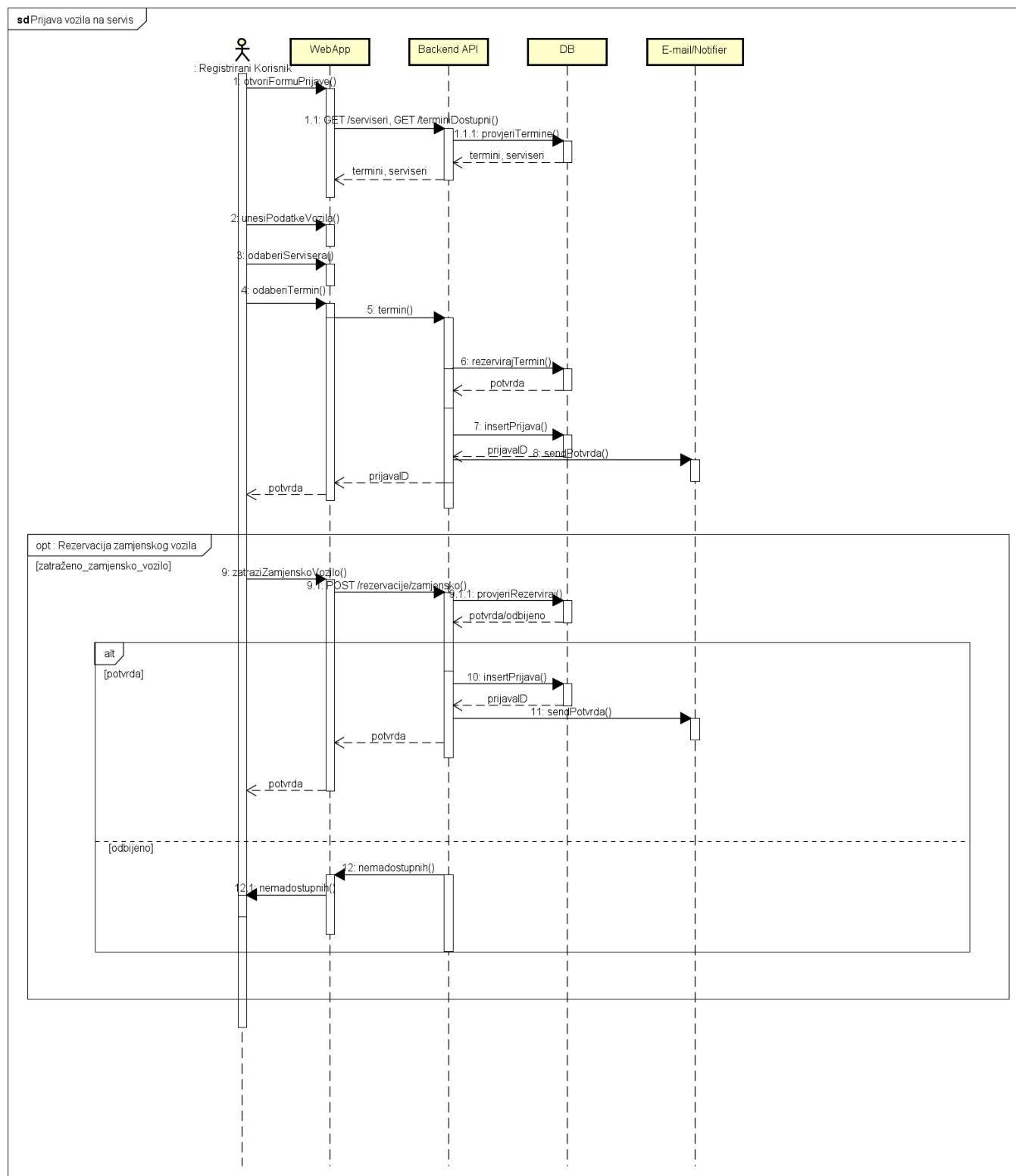
1. Sustav prikazuje listu dostupnih servisera za odabrani termin.
2. Korisnik odabire servisera.
3. Sustav sprema odabir servisera u prijavu.

Odstupanja:

- Nema dostupnih servisera → sustav prikazuje obavijest i predlaže alternativne termine.
-

Sekvencijski dijagrami

Prijava vozila na servis



Kratki opis sekvencijskog dijagrama: Prijava vozila na servis

Dijagram prikazuje proces u kojem korisnik putem web aplikacije prijavljuje vozilo na servis te po potrebi rezervira zamjensko vozilo.

1. Dohvaćanje podataka za formu

Korisnik otvara formu za prijavu vozila.

Web aplikacija (WebApp) od Backend API-ja dohvaca listu servisa i dostupne termine.

Backend API preuzima podatke iz baze podataka i vraća ih WebApp-u.

2. Unos podataka korisnika

Korisnik unosi podatke o vozilu, odabire servis i termin.

WebApp šalje Backend API-ju zahtjev za spremanje prijave.

3. Rezervacija termina i spremanje prijave

Backend API provjerava dostupnost odabranog termina u bazi podataka.

Ako je termin slobodan, rezervira se i sprema se prijava u bazu.

Backend API zatim vraća potvrdu WebApp-u.

4. Opcionalno: Rezervacija zamjenskog vozila

Ako korisnik zatraži zamjensko vozilo, WebApp šalje zahtjev Backend API-ju.

Backend API provjerava dostupnost zamjenskog vozila u bazi.

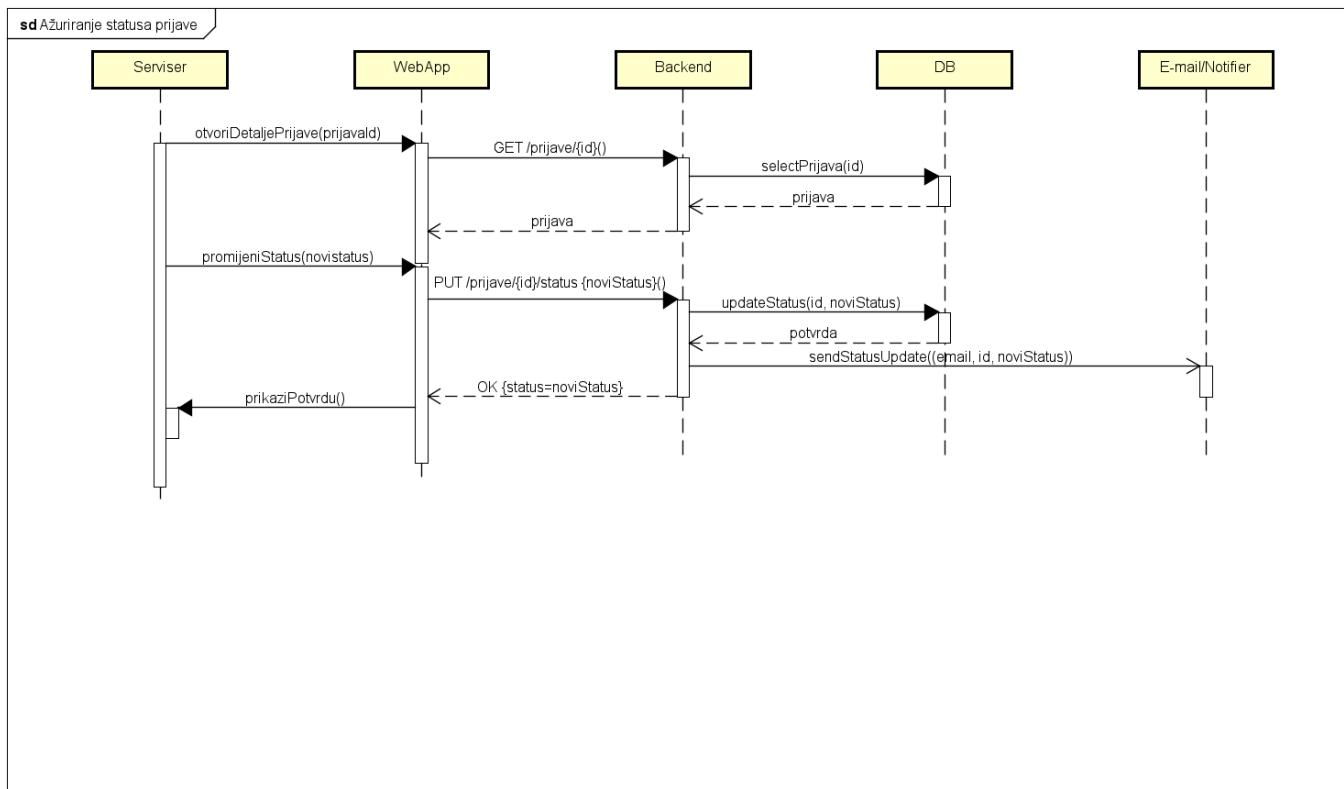
Mogući ishodi:

- Ako je zamjensko vozilo dostupno, rezervacija se sprema u bazu i korisniku se vraća potvrda.
- Ako nije dostupno, WebApp korisniku prikazuje obavijest o nedostupnosti.

5. Slanje završne potvrde

E-mail Notifier šalje korisniku e-mail s potvrdom prijave servisa i eventualne rezervacije zamjenskog vozila.

Ažuriranje statusa prijave



Kratki opis sekvencijskog dijagrama: Ažuriranje statusa prijave

Dijagram prikazuje kako serviser ažurira status postojeće prijave putem web aplikacije, pri čemu se novi status sprema u bazu podataka, a korisniku se šalje obavijest o promjeni.

1. Otvaranje detalja prijave

Serviser preko WebApp-a otvara detalje određene prijave.

WebApp traži od Backend sustava podatke o toj prijavi.

Backend dohvaća podatke iz baze i proslijeđuje ih WebApp-u.

2. Promjena statusa

Serviser u sučelju odabire novi status prijave.

WebApp zatim šalje Backend sustavu informaciju o promjeni statusa.

3. Spremanje promjene

Backend zapisuje novi status prijave u bazu podataka.

Baza potvrđuje da je promjena uspješno spremljena.

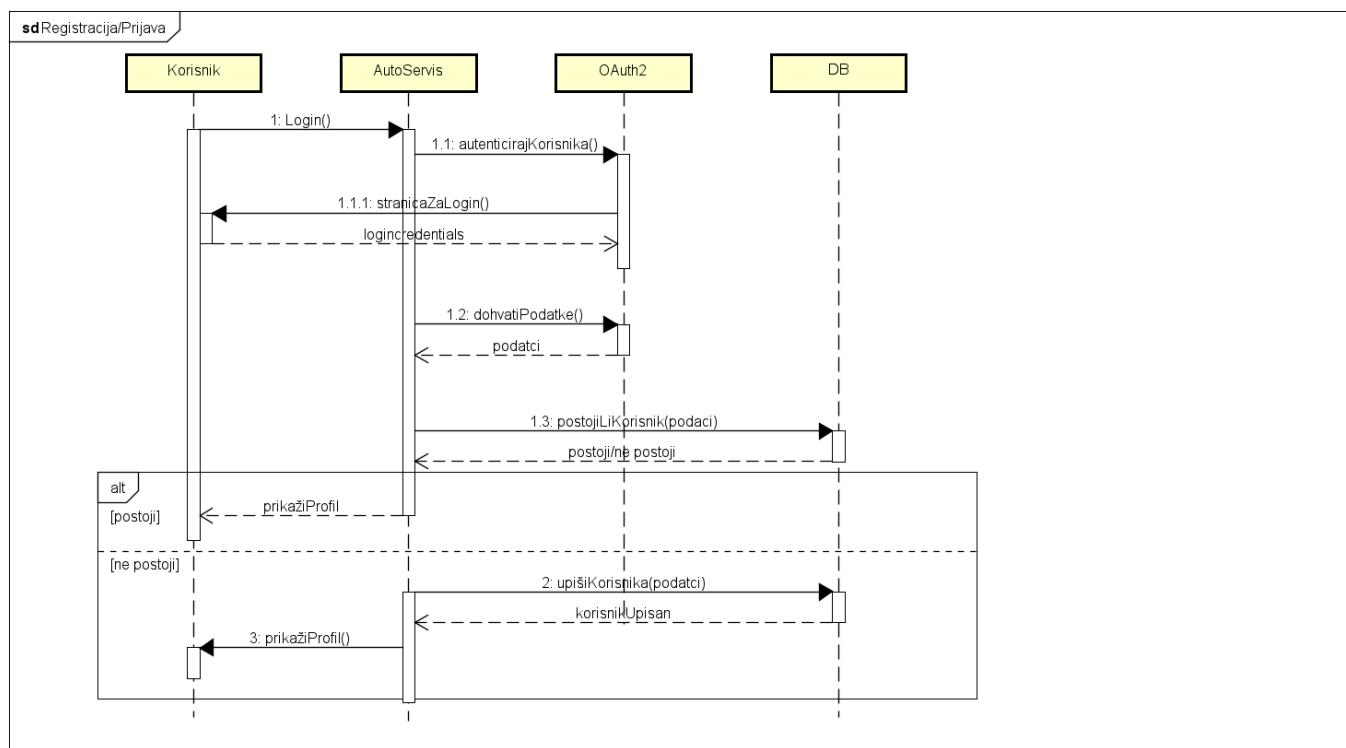
4. Slanje obavijesti korisniku

Nakon što je status ažuriran, Backend obavještava sustav za slanje e-mailova, koji korisniku šalje obavijest o novom statusu prijave.

5. Povratna informacija serviseru

Backend šalje WebApp-u potvrdu da je promjena uspješno provedena, a WebApp prikazuje serviseru poruku o uspješnom ažuriranju statusa.

Registracija/Prijava



Kratki opis sekvencijskog dijagrama: Registracija/Prijava

Dijagram prikazuje proces prijave korisnika putem vanjskog OAuth sustava, kao i automatsko kreiranje profila ako korisnik još ne postoji u aplikacijskoj bazi.

1. Pokretanje prijave

Korisnik pokreće prijavu kroz AutoServis aplikaciju.

Aplikacija preusmjerava korisnika na OAuth sustav za autentikaciju.

2. Autentikacija i dohvaćanje podataka

Korisnik se prijavljuje putem OAuth sustava.

OAuth vraća aplikaciji osnovne podatke o korisniku nakon uspješne autentikacije.

3. Provjera postojanja korisnika

Aplikacija provjerava u bazi podataka postoji li korisnik s dobivenim podacima.

Baza može vratiti jedan od dva rezultata:

- korisnik postoji
- korisnik ne postoji

4. Alternativni tokovi

Ako korisnik postoji

Aplikacija korisniku prikazuje njegov profil i omogućuje mu nastavak rada.

Ako korisnik ne postoji

Aplikacija sprema korisnikove podatke u bazu (automatska registracija).

Nakon uspješnog upisa korisnika, aplikacija prikazuje korisnički profil.

3.3 Tablica povezanosti obrazaca uporabe i funkcionalnih zahtjeva

UC	Naziv	Povezani F-ID
UC1	Pregled informacija o servisu	F-010
UC2	Registracija korisnika (OAuth2)	F-001
UC4	Dodavanje vozila	F-002
UC5	Prijava vozila na servis	F-003, F-008
UC6	Odabir termina	F-003
UC7	Rezervacija zamjenskog vozila	F-008
UC8	Pregled statusa prijave	F-004
UC9	Preuzimanje PDF obrasca	F-011
UC10	Povijest servisa	F-004
UC11	Upravljanje podacima o servisu	F-013
UC12	Upravljanje serviserima	F-014
UC13	Upravljanje korisnicima	F-015
UC14	Pregled prijava (serviser)	F-005
UC15	Dodavanje napomene	F-007
UC16	Ažuriranje statusa prijave	F-006
UC17	Generiranje obrasca	F-011
UC18	Slanje e-mail obavijesti	F-006, F-012
UC19	Generiranje izvještaja	F-009
UC20	Izvoz izvještaja	F-009
UC21	Odabir servisera	F-003

Arhitektura i dizajn sustava

Stil arhitekture

Sustav je izrađen prema **klijent–poslužiteljskom** arhitektonskom stilu.

Korisničko sučelje (frontend) je **React SPA**, dok je poslužiteljski dio (backend) **Java Spring Boot** koji izlaže **REST API**.

Ovaj stil je odabran jer:

- pruža **jasnu separaciju odgovornosti** između prikaza (React) i poslovne logike/podataka (Spring),
 - omogućuje **neovisno skaliranje** frontenda i backenda,
 - olakšava **CI/CD i testiranje** zasebnih dijelova,
 - REST API omogućuje kasniju **reupotrebu** (npr. mobilna aplikacija).
-

Podsistavi

1. Autentikacija i autorizacija (Spring Security)

Upravljanje korisničkim identitetom i pristupnim pravima; OAuth2 integracija.

2. Upravljanje korisnicima

Registracija, ažuriranje profila i uloge (administrator, serviser, korisnik).

3. Servisni nalozi i vozila

Evidencija vozila, otvaranje/praćenje naloga, poslovna pravila (statusi, napomene).

4. Rasporedi i termini

Rezervacije termina.

5. Zamjenska vozila

Evidencija i rezervacija zamjenskih vozila.

6. Izvještavanje i dokumenti (PDF)

Generiranje zapisnika/potvrda u **PDF** formatu (npr. iText/OpenPDF/PDFBox).

7. Integracije

- **Google Maps API** za prikaz lokacije i ruta (na frontendu).
 - Email obavijesti/podsjetnici (SMTP ili servis).
-

Preslikavanje na radnu platformu

- **Razvoj**: lokalno (Spring Boot aplikacija + PostgreSQL baza; optionalno Docker Compose).
- **Producija (cloud)**: **GitHub Actions** za build/test/deploy pipeline.

Prednosti:

- jednostavan lokalni setup i **ponovljiv build**,

- brza i automatizirana isporuka u oblak,
 - mogućnost horizontalnog skaliranja backenda.
-

Spremišta podataka

- **Relacijska baza podataka: PostgreSQL** (ACID, relacije, indeksi).
 - Pristup bazi preko **Spring Data JPA (Hibernate)**; mogućnost pisanja prilagođenih upita (JPQL/SQL) gdje je potrebno.
 - Pohrana generiranih PDF-ova: datotečni sustav ili objektna pohrana (ovisno o okruženju).
-

Mrežni protokoli

- **HTTP(S) + REST** za komunikaciju između React SPA i Spring Boot API-ja (JSON).
 - **SMTP/Email API** za slanje obavijesti.
-

Obrazloženje odabira arhitekture

Izbor arhitekture temeljen na principima oblikovanja

Arhitektura sustava temelji se na **klijent–poslužiteljskom modelu** s odvojenim frontendom i backendom.

Frontend je izrađen u **Reactu** kao Single Page Application (SPA), dok je backend implementiran pomoću **Spring Boot** frameworka.

Komunikacija između slojeva odvija se putem **REST API-ja**, pri čemu su svi zahtjevi autentificirani i autorizirani putem **OAuth2 protokola**.

Ovakav pristup omogućuje jasan razdvajanje odgovornosti između prikaza i poslovne logike te slijedi ključne principe suvremenog arhitektonskog dizajna:

- **Visoka kohezija** – svaki modul (korisnici, vozila, nalozi, PDF generiranje, izvještavanje) ima jasno definiranu odgovornost.
 - **Niska povezanost** – frontend komunicira s backendom isključivo putem REST API sučelja, čime se osigurava neovisnost slojeva.
 - **Modularnost** – backend je podijeljen u slojeve (kontroleri, servisi, rezervorije), što omogućuje jednostavno testiranje i održavanje.
 - **Sigurnost** – autentikacija i autorizacija implementirane su pomoću **Spring Security + OAuth2**, čime se postiže pouzdana zaštita korisničkih podataka.
 - **Skalabilnost** – frontend i backend mogu se neovisno skalirati, ovisno o opterećenju sustava.
 - **Održavanje i proširivost** – standardizirani slojevi i REST komunikacija omogućuju jednostavno dodavanje novih funkcionalnosti.
-

Prednosti odabrane arhitekture

- **Paralelan razvoj** – frontend i backend razvijaju se i testiraju neovisno.
- **Jednostavno održavanje** – svaka promjena u jednom sloju ne utječe izravno na drugi.
- **Reusabilnost** – REST API se može koristiti i za buduće mobilne aplikacije ili vanjske integracije.

- **Sigurna autentikacija** – korisnički pristup kontrolira se pomoću **OAuth2**, koji omogućuje pouzdanu prijavu preko vanjskih identitetskih pružatelja (Google, GitHub, Auth0...).
 - **Integracija s vanjskim servisima** – jednostavno povezivanje s **Google Maps API-jem** i servisima za slanje email obavijesti.
-

Organizacija sustava na visokoj razini

Klijent–poslužitelj arhitektura

Aplikacija je organizirana prema **klijent–poslužiteljskom modelu**, pri čemu je korisničko sučelje (frontend) odvojeno od poslužiteljske logike (backend).

- **Frontend:** Implementiran pomoću **React** frameworka kao **Single Page Application (SPA)**.
Korisničko sučelje omogućuje interaktivno iskustvo, dinamičko učitavanje podataka i prikaz statusa vozila, naloga i termina u stvarnom vremenu.
React aplikacija komunicira s poslužiteljem putem **REST API-ja** i koristi **OAuth2** autentikaciju za pristup zaštićenim resursima.
 - **Backend:** Implementiran pomoću **Spring Boot** frameworka u programskom jeziku **Java**.
Backend je odgovoran za obradu zahtjeva, provedbu poslovne logike, upravljanje korisnicima, generiranje PDF dokumenata i komunikaciju s bazom podataka.
Komunikacija između slojeva unutar backenda organizirana je prema principima **MVC (Model–View–Controller)**, uz dodatne slojeve za servise i rezervorije.
-

Baza podataka

- **Tip:** Relacijska baza podataka
- **Tehnologija:** **PostgreSQL**

Služi za pohranu podataka o korisnicima, vozilima, servisnim nalozima, terminima i zamjenskim vozilima.

Pristup bazi implementiran je putem **Spring Data JPA (Hibernate)**, koji omogućuje:

- automatsko mapiranje Java entiteta na relacijske tablice
- upotrebu **JPQL** i **native SQL** upita po potrebi
- jednostavno dohvaćanje i spremanje podataka kroz rezervorije

Prednosti PostgreSQL-a:

- podrška za složene upite i relacije
- transakcijska sigurnost (ACID)
- visoka stabilnost i mogućnost skaliranja

Opis tablica

osoba

Atribut	Tip podataka	Opis varijable
---------	--------------	----------------

Atribut	Tip podataka	Opis varijable
id_osoba	Primarni ključ	(PK) Jedinstveni identifikator osobe.
ime	VARCHAR(100)	Ime osobe.
prezime	VARCHAR(100)	Prezime osobe.
email	VARCHAR(100), UNIQUE	Valjana email adresa.
telefon	VARCHAR(20)	Kontakt broj telefona.
uloga	VARCHAR(50)	Uloga: korisnik, serviser, administrator.
oauth_id	VARCHAR(255), UNIQUE	Identifikator vanjske autentifikacije (OAuth).

marka

Atribut	Tip podataka	Opis varijable
id_marka	Primarni ključ	(PK) Identifikator marke vozila.
naziv	VARCHAR(50), UNIQUE	Naziv marke (npr. Audi, BMW).

model

Atribut	Tip podataka	Opis varijable
id_model	Primarni ključ	(PK) Identifikator modela vozila.
id_marka	Vanjski ključ	(FK) Povezuje model s tablicom marka.
naziv	VARCHAR(50)	Naziv modela vozila.

vozilo

Atribut	Tip podataka	Opis varijable
id_vozilo	Primarni ključ	(PK) Identifikator vozila.
id_osoba	Vanjski ključ	(FK) Vlasnik vozila → tablica osoba.
id_model	Vanjski ključ	(FK) Model vozila → tablica model.
registracija	VARCHAR(20), UNIQUE	Registarska oznaka vozila.
godina_proizvodnje	INT	Godina proizvodnje (\leq trenutna godina).

servis

Atribut	Tip podataka	Opis varijable
id_servis	Primarni ključ	(PK) Identifikator servisnog centra.
ime_servisa	VARCHAR(100)	Naziv servisa.

Atribut	Tip podataka	Opis varijable
lokacija	TEXT	Lokacija servisa.
serviser		
Atribut	Tip podataka	Opis varijable
id_serviser	Primarni ključ	(PK) Identifikator servisera.
id_osoba	Vanjski ključ	(FK) Povezuje servisera s tablicom osoba.
id_servis	Vanjski ključ	(FK) Povezuje servisera sa servisom.
je_li_voditelj	BOOLEAN	Oznaka je li serviser voditelj servisa.
termin		
Atribut	Tip podataka	Objasnjenje
id_termin	Primarni ključ	(PK) Identifikator termina servisa.
datum_vrijeme	TIMESTAMP	Datum i vrijeme termina.
zauzet	BOOLEAN	Oznaka je li termin zauzet.
prijava_servisa		
Atribut	Tip podataka	Opis varijable
id_prijava	Primarni ključ	(PK) Identifikator prijave servisa.
id_vozilo	Vanjski ključ	(FK) Vozilo koje je prijavljeno.
id_serviser	Vanjski ključ	(FK) Dodijeljeni serviser.
id_termin	Vanjski ključ	(FK) Termin servisa.
status	VARCHAR(50)	Status: zaprimljeno, u obradi, završeno, odgođeno.
datum_prijave	TIMESTAMP	Kada je prijava kreirana.
datum_predaje	TIMESTAMP	Kada je vozilo predano servisu.
datum_preuzimanja	TIMESTAMP	Kada je vozilo preuzeto.
napomena_vlasnika	TEXT	Napomena vlasnika vozila.
zamjena_vozilo		
Atribut	Tip podataka	Opis varijable
id_zamjena	Primarni ključ	(PK) Identifikator zamjenskog vozila.
id_model	Vanjski ključ	(FK) Model vozila → tablica model.

Atribut	Tip podataka	Opis varijable
registracija	VARCHAR(20), UNIQUE	Registarska oznaka zamjenskog vozila.
dostupno	BOOLEAN	Trenutna dostupnost vozila.

rezervacija_zamjene

Atribut	Tip podataka	Opis varijable
id_rezervacija	Primarni ključ	(PK) Identifikator rezervacije.
id_prijava	Vanjski ključ	(FK) Povezuje rezervaciju s prijavom servisa.
id_zamjena	Vanjski ključ	(FK) Povezuje rezervaciju sa zamjenskim vozilom.
datum_od	DATE	Početni datum rezervacije.
datum_do	DATE	Završni datum rezervacije.

napomena_servisera

Atribut	Tip podataka	Opis varijable
id_napomena	Primarni ključ	(PK) Identifikator napomene.
id_prijava	Vanjski ključ	(FK) Povezana prijava servisa.
datum	TIMESTAMP	Vrijeme unosa napomene.
opis	TEXT	Sadržaj napomene.

obrazac

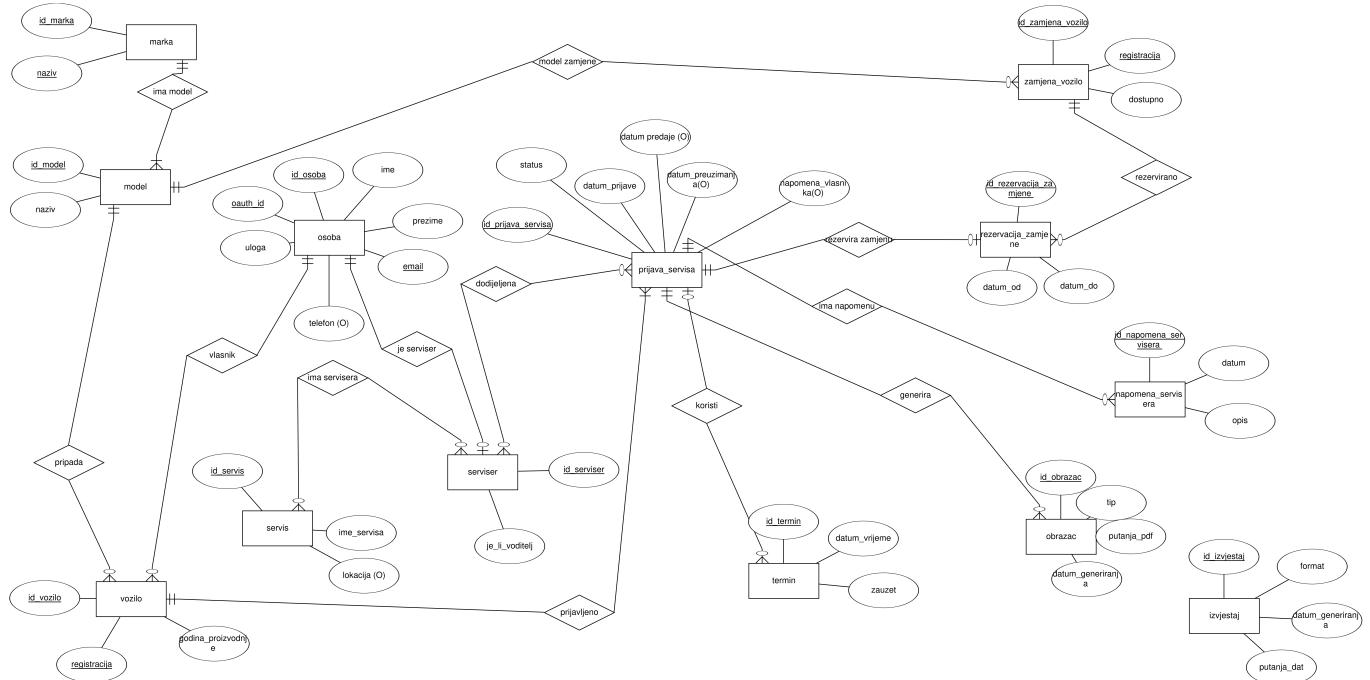
Atribut	Tip podataka	Opis varijable
id_obrazac	Primarni ključ	(PK) Identifikator obrasca.
id_prijava	Vanjski ključ	(FK) Povezana prijava servisa.
tip	VARCHAR(20)	Tip obrasca: predaja ili preuzimanje.
putanja_pdf	TEXT	Putanja do generiranog PDF dokumenta.
datum_generiranja	TIMESTAMP	Vrijeme generiranja dokumenta.

izvjestaj

Atribut	Tip podataka	Opis varijable
id_izvjestaj	Primarni ključ	(PK) Identifikator izvještaja.
format	VARCHAR(20)	Format izvještaja: pdf, xml, xlsx.
datum_generiranja	TIMESTAMP	Vrijeme generiranja izvještaja.

Atribut	Tip podataka	Opis varijable
putanja_dat	TEXT	Putanja do datoteke izvještaja.

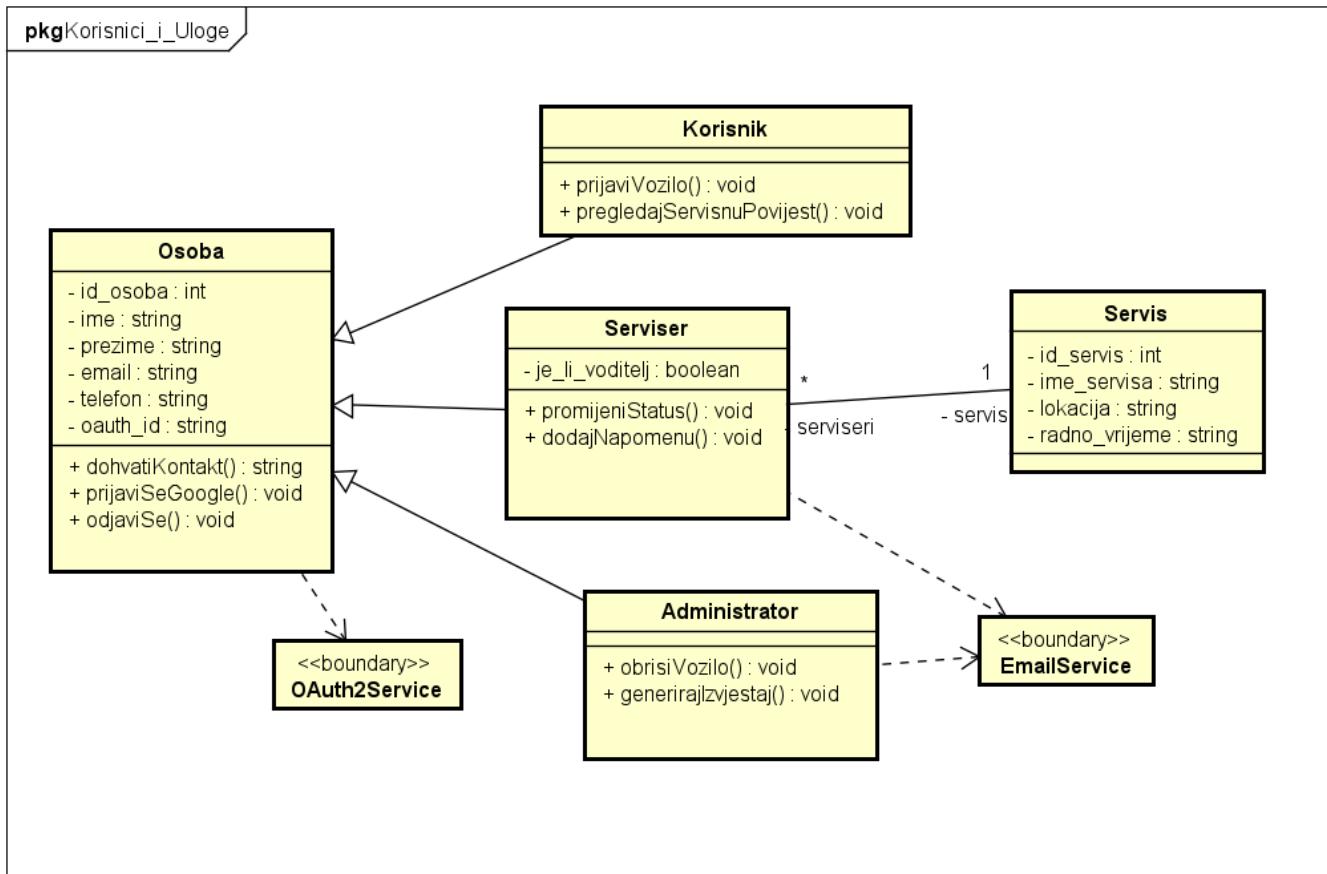
ER dijagram baze podataka



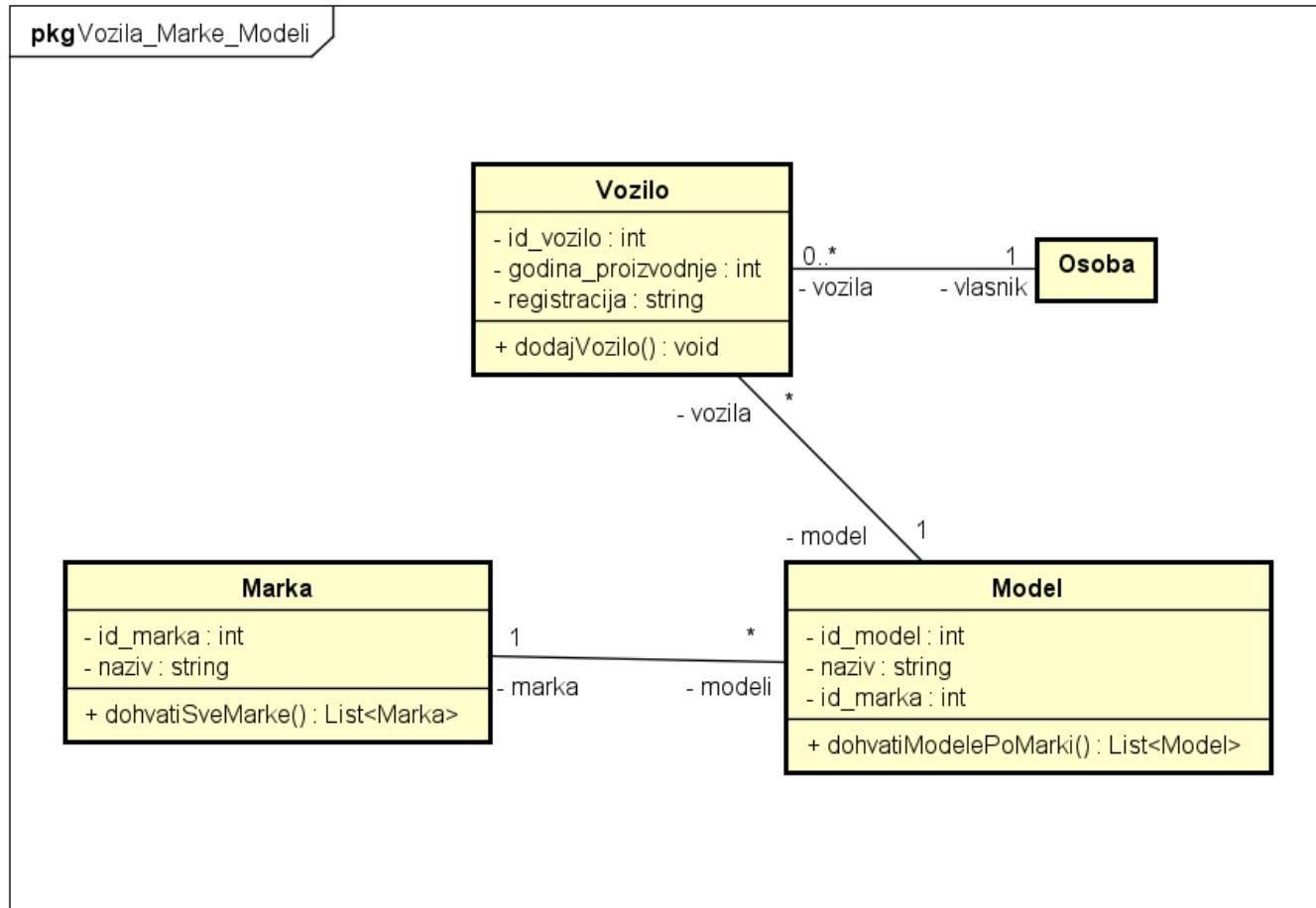
Relacijska shema baze podataka

Class diagram - relacijska_shema

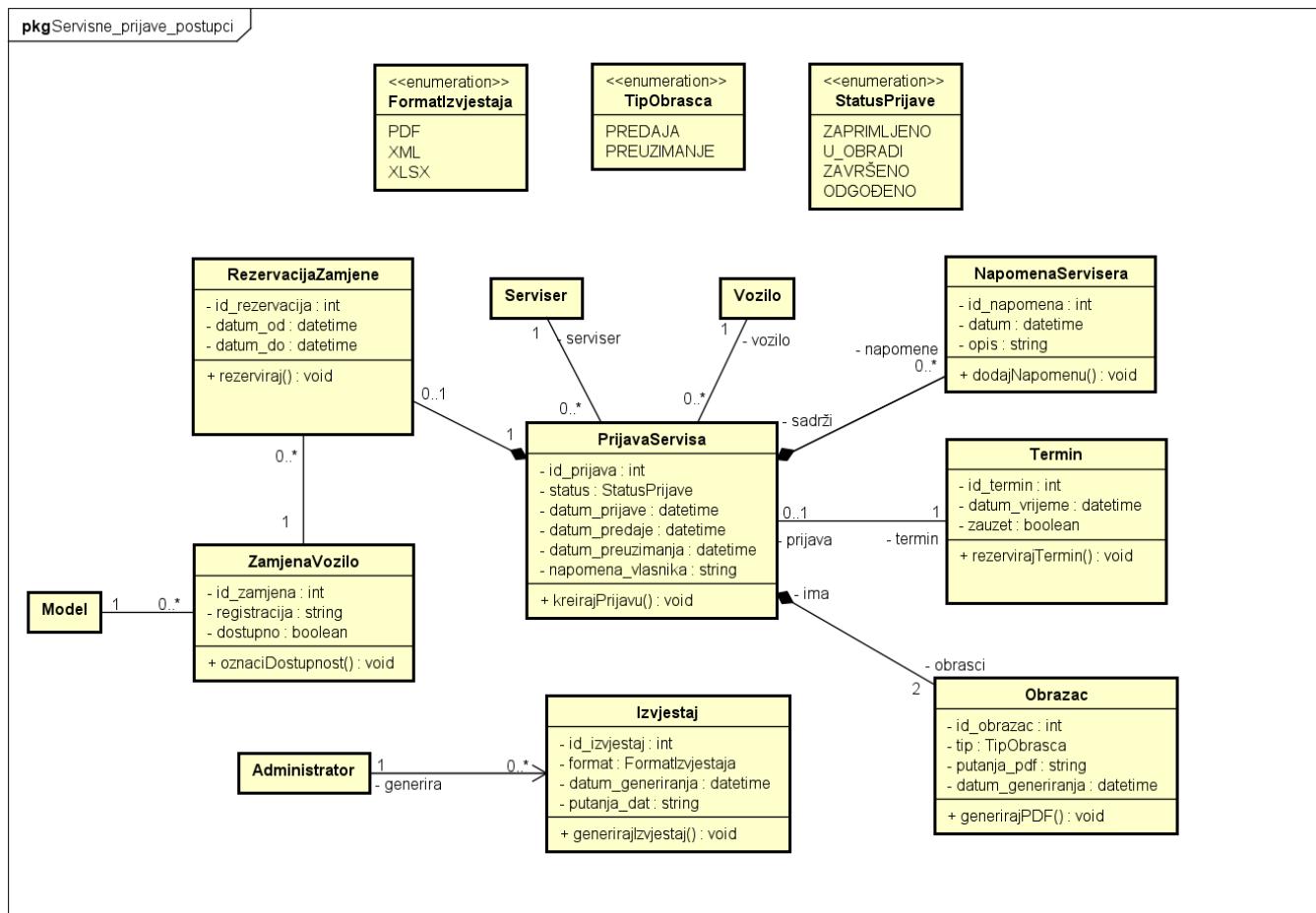
Dijagram razreda – Korisnici i uloge



Dijagram razreda – Vozila, marke i modeli



Dijagram razreda – Servisne prijave i postupci



Grafičko korisničko sučelje (GUI)

Korisničko sučelje implementirano u **Reactu** omogućuje:

- responzivni dizajn (prilagođen desktop i mobilnim uređajima),
- prikaz liste vozila i servisnih naloga,
- prikaz statusa popravaka i zamjenskih vozila,
- prikaz lokacije servisa putem **Google Maps API-ja**,
- autentikaciju korisnika putem **OAuth2** prijave.

1. Programsко инженерство, FER ZEMRIS, <http://www.fer.hr/predmet/proinz>
2. Astah Community, <http://astah.net/editions/uml-new>
3. The Unified Modeling Language, <https://www.uml-diagrams.org/>
4. React, <https://react.dev/learn>
5. Spring Boot – <https://spring.io/projects/spring-boot>
6. PostgreSQL dokumentacija – <https://www.postgresql.org/docs/>

Rev.	Opis promjene/dodataka	Autori	Datum
0.1	Kreiran osnovni kostur dokumentacije i početno strukturiranje sadržaja	Ivano Sorić	20.10.2024
0.2	Dodana uvodna sekcija i opis poslovnog problema	Ivan Petrović	22.10.2024
0.3	Provedena analiza korisničkih potreba i izrađen početni popis funkcionalnosti	Ivano Sorić	25.10.2024
0.4	Izrađena specifikacija korisničkih uloga i definiran tok interakcija	Ivan Petrović	27.10.2024
0.5	Dodani dijagrami slučajeva korištenja (Use Case)	Ivan Petrović	30.10.2024
0.6	Kreirana struktura baze podataka i dodani ER dijagrami	Katarina Bencun	02.11.2024
0.7	Opisana logika glavnih poslovnih procesa i definirani tokovi podataka	Ivano Sorić	05.11.2024
0.8	Dodana poglavlja o arhitekturi sustava i tehničkim prepostavkama	Ivan Petrović	07.11.2024
0.9	Provedena revizija dokumentacije, ispravci i usklađivanje terminologije	Ivano Sorić	10.11.2024
1.0	Dovršena finalna verzija dokumentacije i priprema za predaju	Ivan Petrović	12.11.2024

Dnevnik sastajanja

1. sastanak

Datum: 15. listopada 2024.

Prisustvovali: Ivan Petrović, Ivano Sorić, Katarina Bencun, Adrijan Lazanja, Mia Bagarić, Domagoj Vitković

Teme sastanka

Podjela uloga u timu

Frontend: Mia Bagarić, Domagoj Vitković

Backend: Adrijan Lazanja, Katarina Bencun

Dokumentacija: Ivan Petrović, Ivano Sorić

Uspostava komunikacije

Otvoren Discord server

Kreirana WhatsApp grupa

2. sastanak

Datum: 22. listopada 2024.

Prisustvovali: Ivan Petrović, Ivano Sorić, Katarina Bencun, Adrijan Lazanja, Mia Bagarić, Domagoj Vitković

Teme sastanka

Definiranje funkcionalnih i nefunkcionalnih zahtjeva

Raspravljene osnovne funkcionalnosti sustava

Kreiran inicijalni popis zahtjeva

Raspodjela zadataka

Frontend: izrada početne stranice i UI strukture

Backend: priprema temeljnog API sloja i konfiguracija OAuth2

Dokumentacija: postavljanje strukture wiki stranica

3. sastanak

Datum: 29. listopada 2024.

Prisustvovali: Ivan Petrović, Ivano Sorić, Katarina Bencun, Adrijan Lazanja, Mia Bagarić, Domagoj Vitković

Teme sastanka

Pregled napretka članova tima

Planiranje sljedeće faze razvoja

Frontend: priprema stranice za unos podataka o vozilu
Backend: izrada API logike za dodavanje vozila i spremanje u bazu
Dokumentacija: ažuriranje specifikacije zahtjeva i dijagrama

4. sastanak

Datum: 5. studenog 2024.

Prisustvovali: Ivan Petrović, Ivano Sorić, Katarina Bencun, Adrijan Lazanja, Mia Bagarić, Domagoj Vitković

Teme sastanka

Revizija postojećeg koda i dokumentacije Dogovor oko implementacije ključnih funkcionalnosti

Frontend: UI za pregled i dodavanje vozila
Backend: dovršavanje funkcionalnosti za spremanje vozila
Backend: integracija OAuth2 prijave
Dokumentacija: izrada Use Case dijagrama i ažuriranje arhitekture

5. sastanak

Datum: 12. studenog 2024.

Prisustvovali: Ivan Petrović, Ivano Sorić, Katarina Bencun, Adrijan Lazanja, Mia Bagarić, Domagoj Vitković

Teme sastanka

Stanje projekta

Uspješno implementirane funkcionalnosti:
– dodavanje vozila na stranicu autoservisa
– autentikacija putem OAuth2

Planovi za sljedeće korake

Daljnje proširenje funkcionalnosti servisa
Dorade na sučelju i optimizacija korisničkog iskustva
Ažuriranje i finalizacija dokumentacije