



A GitHub navigation bar showing the repository 'dgrabant/Auto-servis'. It includes links for Code, Issues, Pull requests, Actions, Projects (1), Wiki, Security, Insights, and Settings. There are also icons for search, download, and profile.

Home

[Edit](#) [New page](#)[Jump to bottom](#)

David edited this page last week · [9 revisions](#)

Dobrodošli na Auto-servis wiki!

Programsko inženjerstvo ak.god 2024./2025

Sveučilište u Zagrebu

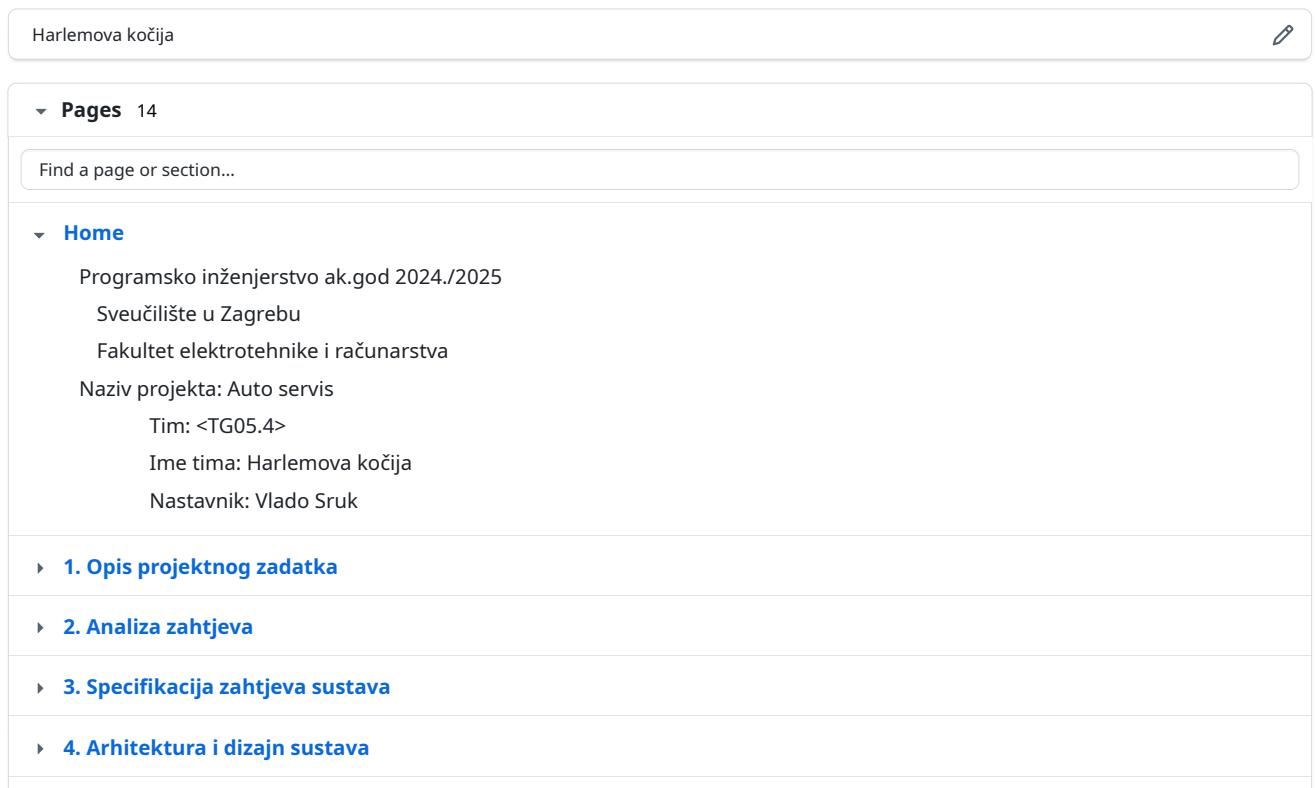
Fakultet elektrotehnike i računarstva

Naziv projekta: Auto servis

Tim: <TG05.4>

Ime tima: Harlemova kočija

Nastavnik: Vlado Sruk



Harlemova kočija

Pages 14

Find a page or section...

Home

- Programsko inženjerstvo ak.god 2024./2025
- Sveučilište u Zagrebu
- Fakultet elektrotehnike i računarstva
- Naziv projekta: Auto servis
- Tim: <TG05.4>
- Ime tima: Harlemova kočija
- Nastavnik: Vlado Sruk

1. Opis projektnog zadatka

2. Analiza zahtjeva

3. Specifikacija zahtjeva sustava

4. Arhitektura i dizajn sustava

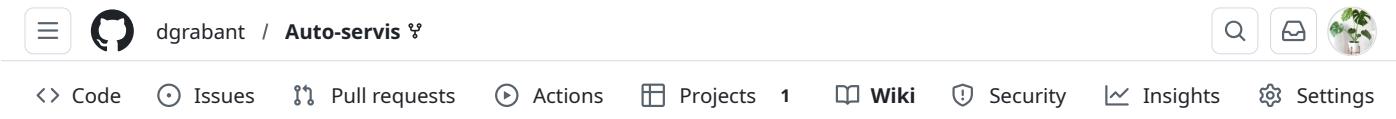
- ▶ [5. Arhitektura komponenata i razmještaja](#)
- ▶ [6. Ispitivanje programskog rješenja](#)
- ▶ [7. Tehnologije za implementaciju aplikacije](#)
- ▶ [8. Upute za puštanje u pogon](#)
- ▶ [9. Zaključak i budući rad](#)
- ▶ [A. Popis literature](#)
- ▶ [A. Dnevnik promjena dokumentacije](#)
- ▶ [B. Prikaz aktivnosti grupe](#)
- ▶ [D. Stranica za sastanke](#)

+ Add a custom sidebar

Clone this wiki locally

<https://github.com/dgrabant/Auto-servis.wiki.git>





A GitHub repository header for the 'Auto-servis' wiki page. It includes a sidebar icon, the repository name 'dgrabant / Auto-servis', a search bar, and several navigation links: Code, Issues, Pull requests, Actions, Projects (1), Wiki, Security, Insights, and Settings.

1. Opis projektnog zadatka

[Edit](#) [New page](#)

[Jump to bottom](#)

David edited this page 5 days ago · [4 revisions](#)

Opis projektnog zadatka

Ovaj projekt nastao je u sklopu kolegija *Programsko inženjerstvo* na Fakultetu elektrotehnike i računarstva Sveučilišta u Zagrebu.

Cilj tima je razvoj web aplikacije koja će olakšati organizaciju rada i komunikaciju između autoservisa i njihovih korisnika.

Aplikacija će omogućiti jednostavno upravljanje terminima popravaka, evidenciju vozila i pregled statistike učinkovitosti servisera.

Opis sustava

Projektni zadatak usmjeren je na razvoj web aplikacije za upravljanje radom autoservisa, s naglaskom na digitalizaciju procesa zaprimanja, praćenja i evidencije vozila.

Sustav omogućava korisnicima online prijavu i praćenje statusa popravka, dok serviserima i administratorima nudi alate za učinkovito planiranje, pregled zauzeća resursa i komunikaciju s klijentima.

Poseban naglasak stavljen je na preglednost, jednostavnost korištenja i mogućnost proširenja funkcionalnosti.

Korisnički zahtjevi

Korisnici očekuju jednostavan i responzivan sustav koji omogućava potpunu kontrolu nad procesom servisa.

Glavne funkcionalnosti uključuju:

- prijavu i registraciju putem OAuth2 autentifikacije
- online zakazivanje termina i rezervaciju zamjenskog vozila
- praćenje statusa popravka u stvarnom vremenu
- izvoz podataka u PDF, XML i XLSX formatu
- pregled statistike učinkovitosti servisera i autoservisa

Aplikacija će biti prilagođena za korištenje na stolnim i mobilnim uređajima.

Potencijalna korist projekta

Primjenom sustava smanjuje se administrativno opterećenje zaposlenika, a korisnicima se omogućuje veća transparentnost i brža komunikacija.

Servis može učinkovitije planirati resurse, pratiti rezultate rada i analizirati podatke o vremenu popravaka, iskorištenosti resursa i zadovoljstvu korisnika.

Na taj način aplikacija doprinosi digitalizaciji poslovanja i povećanju učinkovitosti autoservisa.

Postojeća rješenja i prednosti sustava

Na tržištu postoje neki alati, no većina ih je namijenjena velikim servisnim lancima i ne nudi lokalizirane funkcionalnosti.

Naš projekt razvija rješenje prilagođeno manjim i srednjim servisima, s fokusom na jednostavnost, mogućnost integracije s Google Maps servisom te jasnim i lokaliziranim sučeljem.

Skup korisnika

Primarni korisnici sustava su:

- serviseri i administratori koji upravljaju radom servisa
 - krajnji korisnici (vlasnici vozila) koji prijavljuju kvarove i prate status popravka
- Sekundarni korisnici mogu uključivati upravu tvrtke koja koristi podatke sustava za analizu učinkovitosti poslovanja.

Mogućnost nadogradnje

Sustav je zamišljen modularno, što omogućava jednostavno proširenje funkcionalnosti u budućnosti.
Potencijalne nadogradnje uključuju:

- integraciju s IoT uređajima za dijagnostiku vozila
- razvoj mobilne aplikacije
- primjenu AI modela za preporuke termina ili optimizaciju zaliha dijelova
- generiranje naprednih izvještaja potrebnih za porezni sustav i slično

Opseg projekta

Projekt obuhvaća razvoj baze podataka, web sučelja, autentifikacije, modula za rezervacije i izvještavanje.
Cilj je stvoriti stabilan, siguran i intuitivan sustav koji poboljšava poslovanje autoservisa i komunikaciju s korisnicima.

| napomena: dijelovi teksta izmjenjeni, formatirani uporabom alata ChatGPT

The screenshot shows a sidebar on the left with the following structure:

- Pages 14 (with a pencil icon)
- Find a page or section...
- Home
- 1. Opis projektnog zadatka
 - Opis projektnog zadatka
 - Opis sustava
 - Korisnički zahtjevi
 - Potencijalna korist projekta
 - Postojeća rješenja i prednosti sustava
 - Skup korisnika
 - Mogućnost nadogradnje
 - Opseg projekta
- 2. Analiza zahtjeva
- 3. Specifikacija zahtjeva sustava

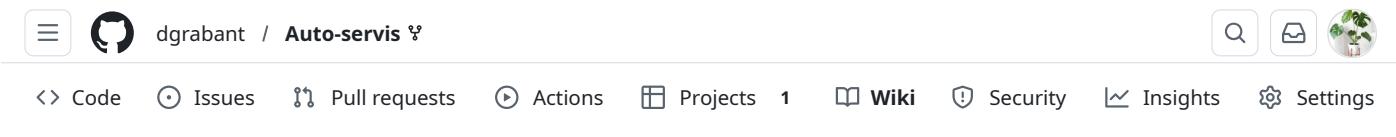
- ▶ [4. Arhitektura i dizajn sustava](#)
- ▶ [5. Arhitektura komponenata i razmještaja](#)
- ▶ [6. Ispitivanje programskog rješenja](#)
- ▶ [7. Tehnologije za implementaciju aplikacije](#)
- ▶ [8. Upute za puštanje u pogon](#)
- ▶ [9. Zaključak i budući rad](#)
- ▶ [A. Popis literature](#)
- ▶ [A. Dnevnik promjena dokumentacije](#)
- ▶ [B. Prikaz aktivnosti grupe](#)
- ▶ [D. Stranica za sastanke](#)

+ Add a custom sidebar

Clone this wiki locally

<https://github.com/dgrabant/Auto-servis.wiki.git>





A screenshot of a GitHub repository page. At the top left is the repository name "dgrabant / Auto-servis". To the right are several navigation links: "Code", "Issues", "Pull requests", "Actions", "Projects 1", "Wiki", "Security", "Insights", and "Settings". On the far right are icons for search, notifications, and profile. Below the header is a horizontal menu with "Edit" and "New page" buttons, followed by a link to "Jump to bottom".

2. Analiza zahtjeva

David edited this page on Oct 26, 2025 · [8 revisions](#)

Funkcionalni zahtjevi

ID zahtjeva	Opis	Prioritet	Izvor	Kriteriji prihvatanja
F-01	Sustav omogućuje korisnicima registraciju putem vanjskog OAuth2 servisa (npr. Google, Microsoft).	Visok	Zahtjev naručitelja	Korisnik se može uspješno prijaviti putem OAuth2 servisa bez potrebe za zasebnom lozinkom.
F-02	Sustav omogućuje korisnicima prijavu putem OAuth2 servisa.	Visok	Zahtjev naručitelja	Korisnik se može prijaviti i pristupiti svom korisničkom sučelju.
F-03	Registrirani korisnik može unijeti osnovne podatke o vozilu (marka, model, godina, registracija, opis problema).	Visok	Zahtjev korisnika	Nakon unosa podataka, sustav pohranjuje vozilo u bazu i povezuje ga s korisnikom.
F-04	Sustav omogućuje pregled dostupnih termina za servis.	Visok	Zahtjev korisnika	Prilikom odabira servisa korisnik vidi popis slobodnih termina.
F-05	Korisnik može rezervirati termin za servis i po potrebi rezervirati zamjensko vozilo.	Visok	Zahtjev korisnika	Sustav potvrđuje rezervaciju i šalje potvrdu e-mailom.
F-06	Sustav šalje automatsku potvrdu prijave servisa putem e-maila.	Visok	Zahtjev korisnika	Korisnik prima e-mail s potvrdom i detaljima o terminu.
F-07	Sustav automatski šalje podsjetnik ako je termin pomaknut više od 3 dana.	Srednji	Zahtjev korisnika	Korisnik prima obavijest e-mailom u roku od 1 minute nakon promjene termina.
F-08	Sustav generira obrazac u PDF formatu prilikom predaje i preuzimanja vozila.	Visok	Poslovno pravilo	PDF se generira automatski i uključuje potpis korisnika i servisera.
F-09	Serviser može pregledati i potvrđivati prijave koje su mu dodijeljene.	Visok	Zahtjev servisera	Serviser vidi sve dodijeljene prijave i može ažurirati status popravka.
F-10	Serviser može dodavati napomene o radu i statusu vozila.	Srednji	Zahtjev servisera	Serviser može unositi napomene koje su vidljive korisniku i administratoru.
F-11	Administrator može dodavati, uređivati i brisati podatke o servisu, serviserima i korisnicima.	Visok	Zahtjev voditelja servisa	Administrator vidi sve entitete sustava i ima ovlasti izmjene.
F-12	Sustav omogućuje pregled statistike o broju zaprimljenih vozila, trajanju popravaka i zauzeću zamjenskih vozila.	Visok	Zahtjev voditelja servisa	Statistika se prikazuje grafički i tabično.
F-13	Sustav omogućuje izvoz svih podataka u PDF, XML i XLSX formatu.	Visok	Zahtjev naručitelja	Korisnik može izvesti izvještaje u odabranom formatu.
F-14	Sustav prikazuje lokaciju servisa putem Google Maps servisa.	Srednji	Zahtjev korisnika	Na korisničkom sučelju se prikazuje interaktivna karta.
F-15	Sustav jednoznačno označava vozila koja su trenutno u obradi.	Visok	Zahtjev servisera	Svako vozilo ima status "U tijeku", "Završeno" ili "Čeka dijelove".

Ostali (nefunkcionalni) zahtjevi

Zahtjevi za održavanje

ID zahtjeva	Opis	Prioritet
NO-1	Sustav treba biti popraćen "Priručnikom za rad" koji opisuje pravilnu upotrebu sustava.	Visok
NO-2	Sustav treba imati detaljnu tehničku i korisničku dokumentaciju koja opisuje procese instalacije, konfiguracije i održavanja.	Visok
NO-3	Kod sustava treba biti dokumentiran prema "Code Conventions" smjernicama i sadržavati komentare u ključnim dijelovima programskog koda.	Visok
NO-4	Aplikacija mora biti modularna, kako bi omogućila jednostavno dodavanje novih funkcionalnosti i integraciju s vanjskim servisima (npr. Google Maps, OAuth2).	Srednje
NO-5	Sustav treba omogućiti jednostavno ažuriranje i nadogradnju bez potrebe za ručnom intervencijom administratora.	Srednje

Zahtjevi skalabilnosti

ID zahtjeva	Opis	Prioritet
NS-1	Sustav mora podržavati istovremeni rad najmanje 1000 korisnika bez značajnog pada performansi.	Visok
NS-2	Sustav mora omogućiti obradu složenih zadataka (npr. generiranje izvještaja, slanje e-mail obavijesti).	Srednje

Zahtjevi pristupačnosti i prilagodljivosti

ID zahtjeva	Opis	Prioritet
NPP-1	Sučelje aplikacije mora biti responzivno i prilagođeno različitim uređajima (računala, tableti, mobiteli).	Visok
NPP-2	Implementirati mehanizme visoke dostupnosti: redundanciju baze podataka, failover poslužitelje i sustav za automatski oporavak.	Visok
NPP-3	Sigurnost podataka: svi osjetljivi podaci moraju biti šifrirani i zaštićeni od neovlaštenog pristupa.	Visok
NPP-4	Sustav treba sadržavati podršku za korisnike putem e-maila i integriranog obrasca za povratne informacije.	Srednje
NPP-5	Sustav mora biti vizualno jasan i dosljedan u svim dijelovima korisničkog sučelja.	Srednje

Zahtjevi performansi

ID zahtjeva	Opis	Prioritet
NP-1	Sustav treba reagirati na korisničke akcije unutar 1-2 sekunde u 95% slučajeva.	Visok
NP-2	Sustav mora generirati izvještaje (PDF, XML, XLSX) u roku od 5 sekundi.	Visok

NP-3	Sustav mora omogućiti dohvat podataka o vozilima i terminima iz baze unutar 1 sekunde.	Visok
NP-4	Aplikacija mora biti optimizirana za efikasno korištenje procesora, memorije i mrežnih resursa.	Srednje
NP-5	Sustav mora automatski brisati zastarjele privremene podatke radi očuvanja performansi baze.	Nisko

Zahtjevi sigurnosti

ID zahtjeva	Opis	Prioritet
NSI-1	Sustav mora koristiti HTTPS (TLS 1.3) za enkripciju svih komunikacija između korisnika i poslužitelja.	Visok
NSI-2	Autentifikacija korisnika mora biti izvedena putem vanjskog OAuth2 servisa.	Visok
NSI-3	Sustav mora automatski odjaviti korisnika nakon 15 minuta neaktivnosti.	Srednje
NSI-4	Sustav mora bilježiti sve administrativne radnje i čuvati logove najmanje 90 dana.	Srednje
NSI-5	Prava pristupa moraju biti definirana po korisničkim ulogama (korisnik, serviser, administrator, voditelj).	Visok

Zahtjevi ograničenja

ID zahtjeva	Opis	Prioritet
NOG-1	Maksimalna veličina pojedinačne PDF datoteke ne smije prelaziti 10 MB.	Srednje
NOG-2	Sustav mora poštovati GDPR regulativu i omogućiti korisnicima brisanje podataka na zahtjev.	Visok

Dionici

Dionik	Opis	Interes u sustavu
Neregistrirani korisnik	Korisnik koji samo posjećuje stranicu i vidi osnovne informacije o njoj, ali ne sudjeluje u servisu.	Pregled javno dostupne prezentacije servisa.
Klijent / registrirani korisnik	Vlasnik vozila koji koristi aplikaciju za prijavu i praćenje servisa.	Brza i transparentna prijava i praćenje statusa popravka.
Serviser	Osoba koja obavlja popravke i ažurira statuse.	Jednostavan pregled dodijeljenih poslova i evidencija izvršenih radova.
Voditelj servisa	Koristi aplikaciju za pregled poslovnih statistika i generiranje izvještaja.	Uvid u učinkovitost i iskorištenost resursa.
Administrator	Osoba zadužena za upravljanje korisnicima, servisima i podacima.	Potpuna kontrola nad sustavom i održavanje podataka.
OAuth2 servis	Vanjski servis za autentifikaciju korisnika.	Sigurna prijava bez lokalnog čuvanja lozinki.

Aktori i njihovi funkcionalni zahtjevi

A-1 Registrirani korisnik (inicijator)

Može:

- Registrirati se putem OAuth2 servisa (F-01, F-02)
- Unijeti podatke o vozilu (F-03)
- Pregledati slobodne termine (F-04)
- Rezervirati termin i zamjensko vozilo (F-05)
- Primiti potvrdu i podsjetnik putem e-maila (F-06, F-07)
- Pregledati status vozila (F-15)
- Preuzeti obrazac u PDF formatu (F-08)

A-2 Serviser (sudionik)

Može:

- Pregledati dodijeljene prijave (F-09)
- Dodavati napomene i ažurirati status (F-10)
- Generirati obrazac prilikom primopredaje vozila (F-08)

A-3 Administrator

Može:

- Upravljati podacima o serviserima, servisima i korisnicima (F-11)
- Nadzirati sve rezervacije i izvještaje (F-12, F-13)
- Osigurati sigurnost i dostupnost sustava (NF-1, NF-4)

A-4 Voditelj servisa

Može:

- Pregledavati poslovnu statistiku i izvještaje (F-12, F-13)
- Pratiti zauzeće zamjenskih vozila (F-12)
- Donositi odluke na temelju izvještaja (F-13)

| napomena: dijelovi teksta formatirani te nadopunjeni uporabom alata ChatGPT

The screenshot shows a sidebar with the following structure:

- Pages 14
- Find a page or section...
- Home
- 1. Opis projektnog zadatka
- 2. Analiza zahtjeva
 - Funkcionalni zahtjevi
 - Ostali (nefunkcionalni) zahtjevi
 - Zahtjevi za održavanje

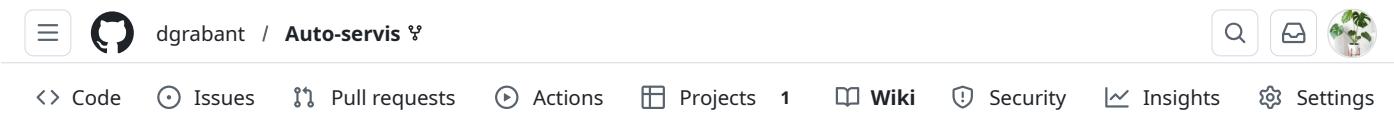
- ▶ [3. Specifikacija zahtjeva sustava](#)
- ▶ [4. Arhitektura i dizajn sustava](#)
- ▶ [5. Arhitektura komponenata i razmještaja](#)
- ▶ [6. Ispitivanje programskog rješenja](#)
- ▶ [7. Tehnologije za implementaciju aplikacije](#)
- ▶ [8. Upute za puštanje u pogon](#)
- ▶ [9. Zaključak i budući rad](#)
- ▶ [A. Popis literature](#)
- ▶ [A. Dnevnik promjena dokumentacije](#)
- ▶ [B. Prikaz aktivnosti grupe](#)
- ▶ [D. Stranica za sastanke](#)

+ Add a custom sidebar

Clone this wiki locally

<https://github.com/dgrabant/Auto-servis.wiki.git>





3. Specifikacija zahtjeva sustava

[Edit](#) [New page](#)[Jump to bottom](#)

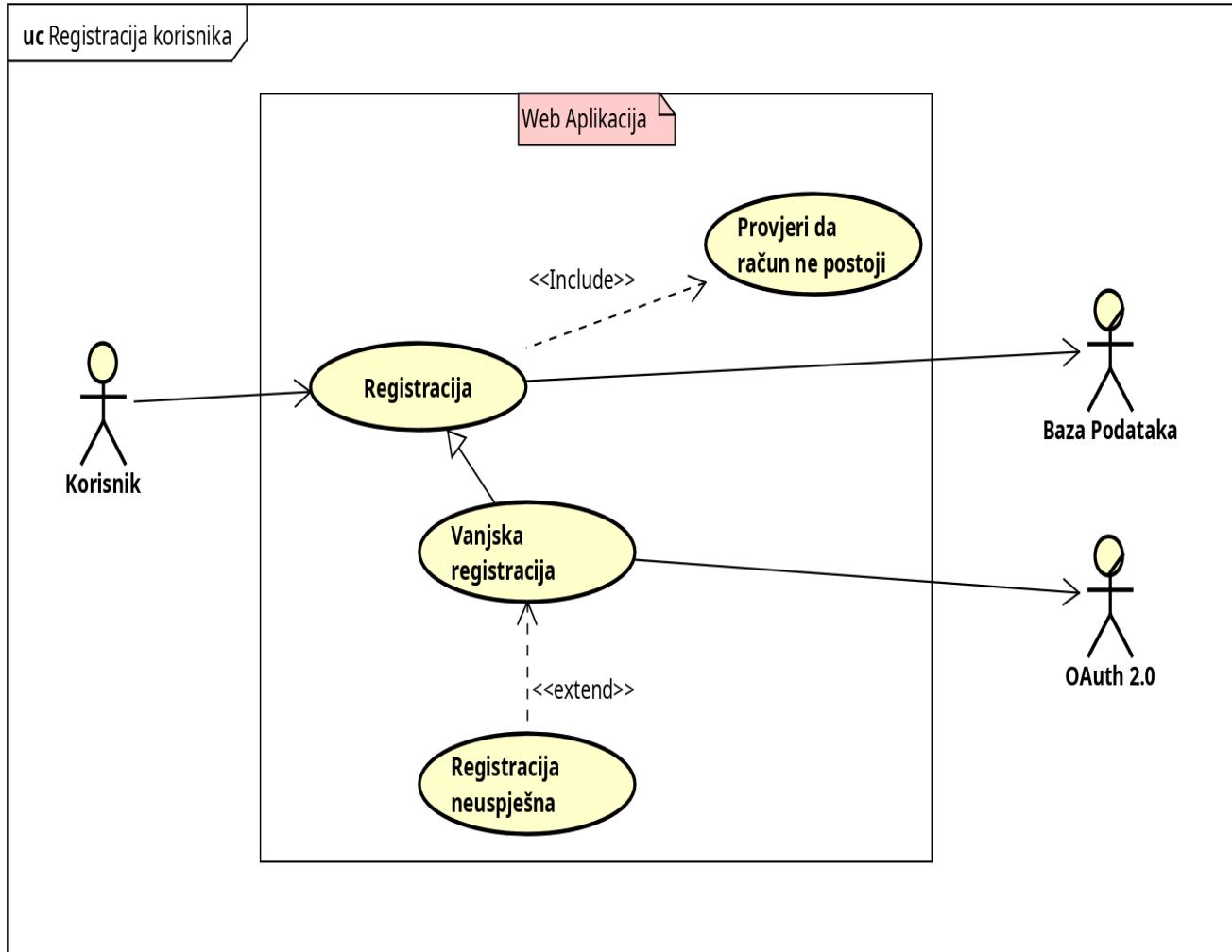
David edited this page on Oct 23, 2025 · [6 revisions](#)

Obrasci uporabe i Dijagrami obrazaca uporabe

Opis obrazaca uporabe

UC-01 Registracija korisnika

- Glavni sudionik:** Korisnik
- Cilj:** Omogućiti korisniku registraciju putem vanjskog OAuth2 servisa (Google, Microsoft).
- Sudionici:** Sustav, OAuth2 servis
- Preduvjet:** Korisnik nije već registriran.



Osnovni tijek:

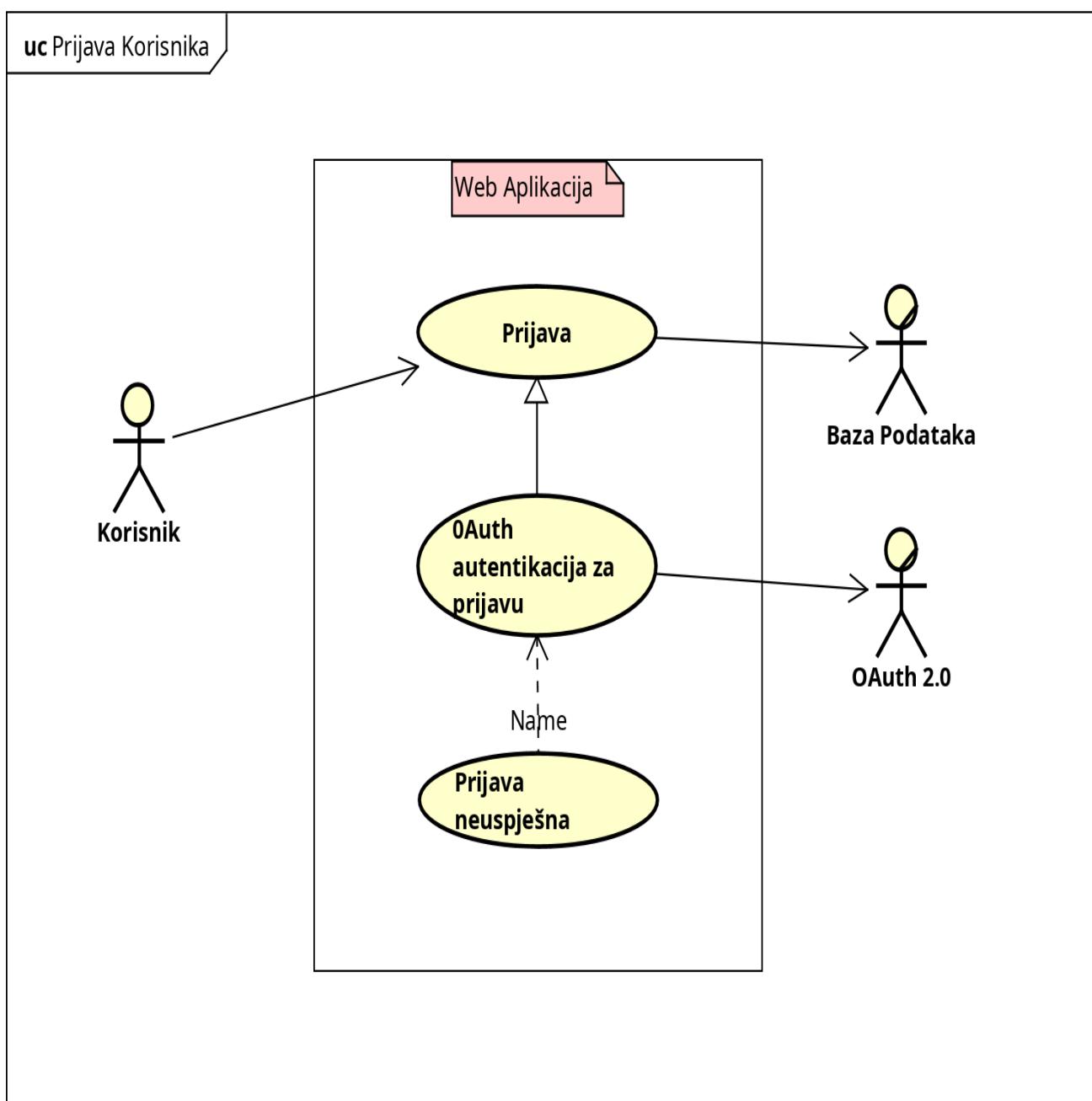
1. Korisnik odabire opciju "Registracija putem Google/Microsoft računa". (F-001)
2. Sustav preusmjerava korisnika na OAuth2 servis.
3. OAuth2 servis provjerava identitet korisnika.
4. Nakon potvrde, sustav kreira novi korisnički račun.
5. Korisnik prima potvrdu e-mailom o uspješnoj registraciji.

Moguća odstupanja:

- 2a: OAuth2 servis nije dostupan → sustav prikazuje poruku o nedostupnosti i nudi ponovni pokušaj.
- 3a: Korisnik odbije pristup → registracija se prekida.

UC-02 Prijava korisnika

- **Glavni sudionik:** Korisnik
- **Cilj:** Omogućiti prijavu korisnika putem OAuth2 servisa.
- **Sudionici:** Sustav, OAuth2 servis
- **Preduvjet:** Korisnik mora biti prethodno registriran.



Osnovni tijek:

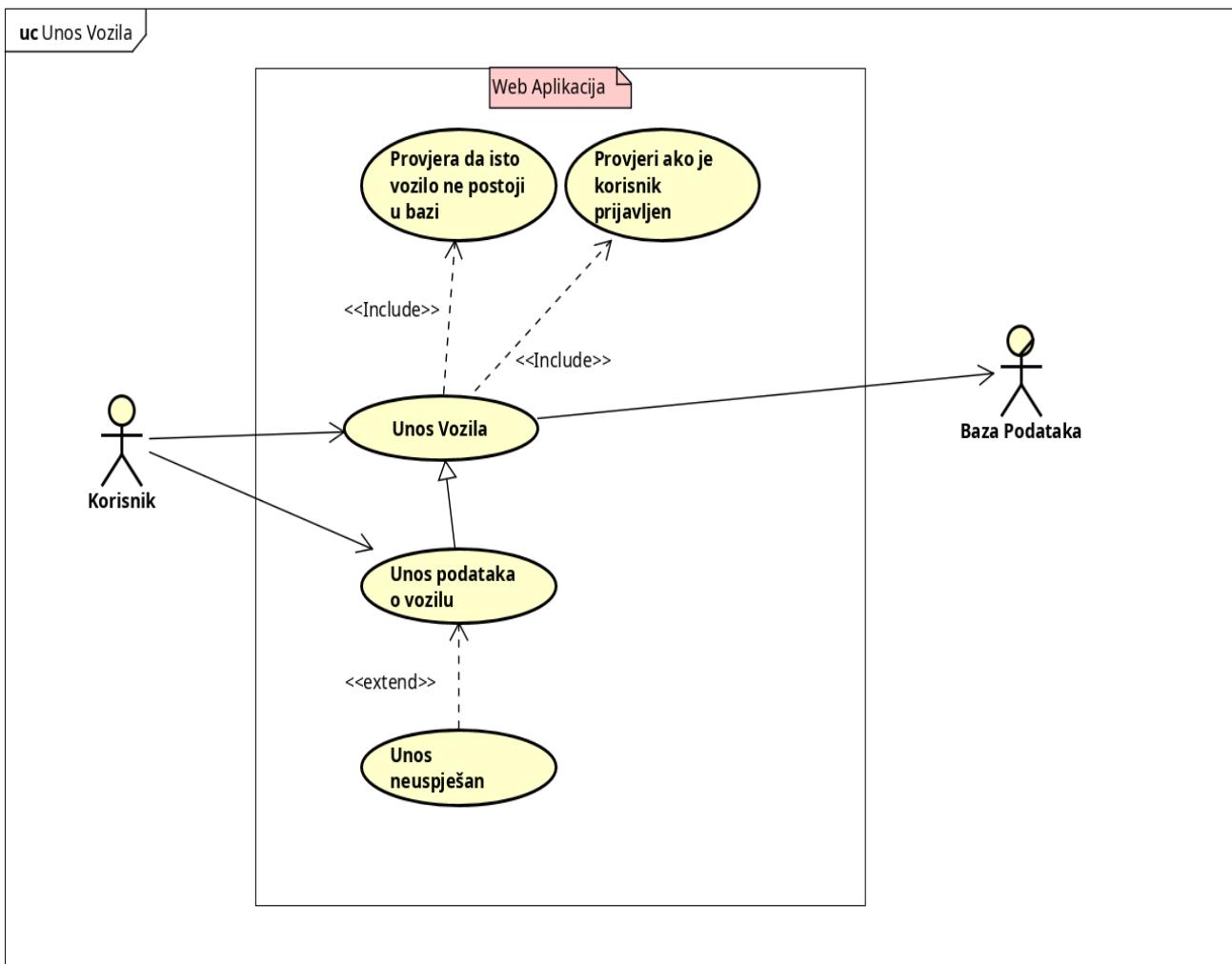
1. Korisnik odabire opciju "Prijava putem Google/Microsoft". (F-002)
2. Sustav preusmjerava korisnika na OAuth2 servis.
3. OAuth2 servis potvrđuje identitet i vraća token.
4. Sustav validira token i otvara korisnički profil.

Moguća odstupanja:

- 3a: Token nije valjan → sustav prikazuje poruku "Prijava nije uspjela".
- 4a: Sustav ne može pristupiti servisu → korisniku se nudi ponovni pokušaj.

UC-03 Unos vozila

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Omogućiti korisniku unos podataka o vozilu.
- **Sudionici:** Sustav, baza podataka
- **Preduvjet:** Korisnik je prijavljen.



Osnovni tijek:

1. Korisnik otvara izbornik "Moja vozila" i odabire "Dodaj novo vozilo". (F-003)
2. Unosi podatke: marka, model, godina, registracija, opis problema.
3. Sustav provjerava valjanost unosa.
4. Sustav pohranjuje vozilo u bazu i povezuje ga s korisničkim računom.
5. Korisnik dobiva potvrdu o uspješnom unosu.

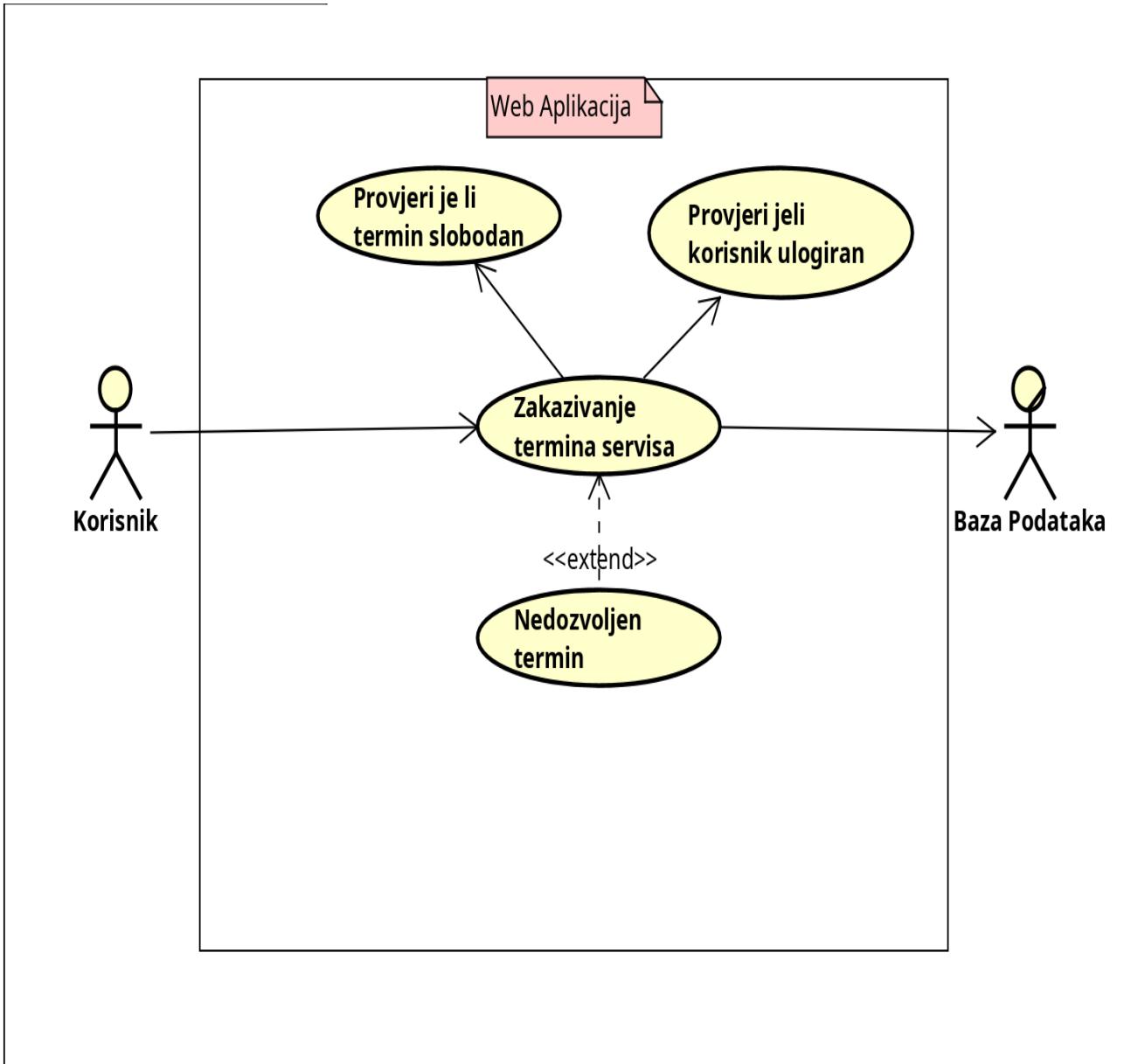
Moguća odstupanja:

- 3a: Podaci nisu valjni → sustav prikazuje poruku o grešci i traži ispravak.
- 4a: Neuspjelo spremanje → sustav nudi ponovni pokušaj.

UC-04 Zakazivanje termina servisa

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Omogućiti korisniku zakazivanje termina i servisera.
- **Sudionici:** Sustav, baza podataka, servisni kalendar
- **Preduvjet:** Korisnik ima registrirano vozilo.

uc Zakazivanje Termina Servisa



Osnovni tijek:

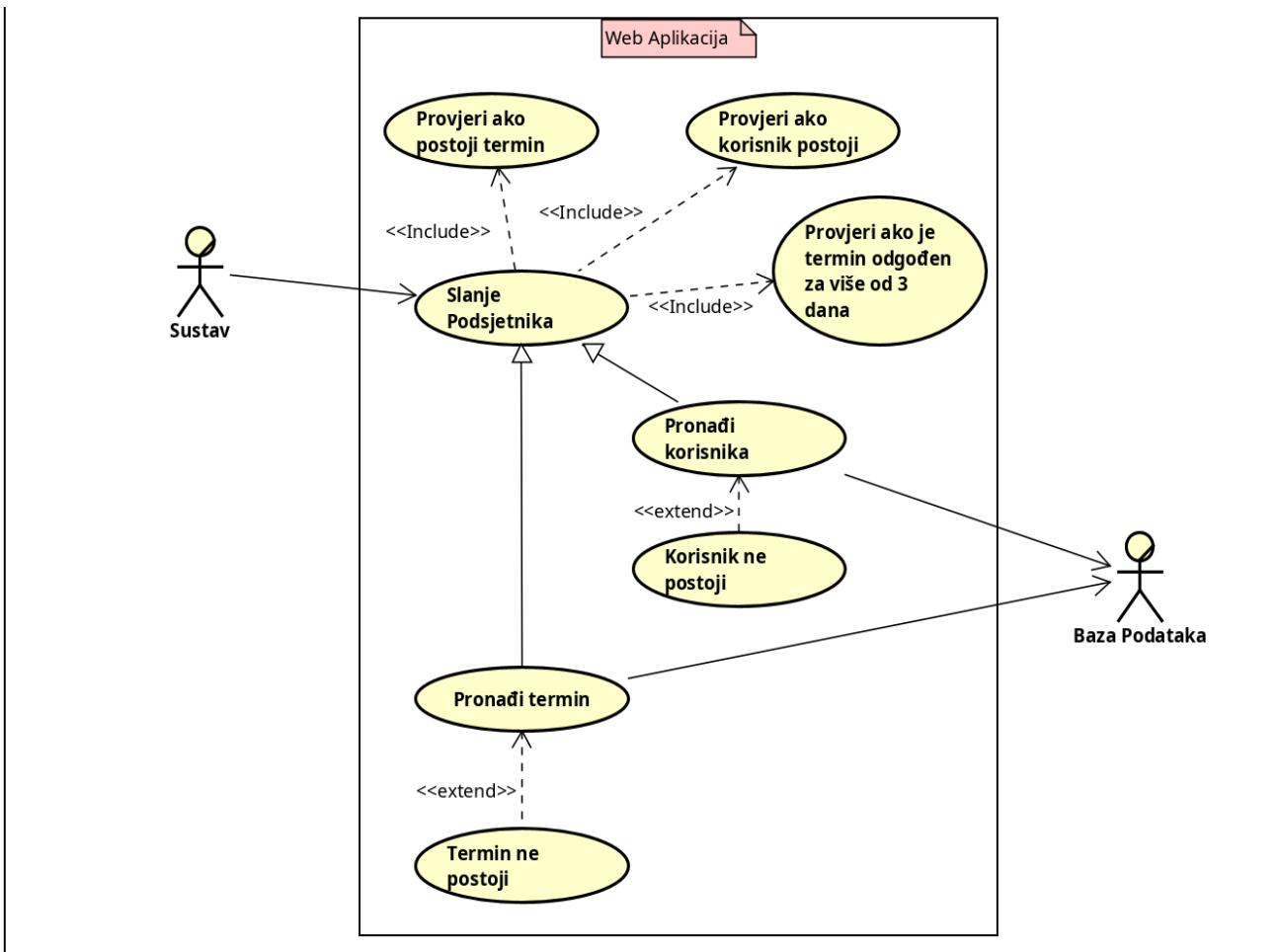
1. Korisnik odabire vozilo za servis. (F-004)
2. Prikazuju se dostupni termini i serviseri.
3. Korisnik odabire željeni termin i servisera.
4. Sustav potvrđuje rezervaciju i šalje e-mail potvrdu.
5. Termin se bilježi u kalendar sustava.

Moguća odstupanja:

- 2a: Nema slobodnih termina → sustav nudi alternativne datume.
- 4a: Slanje e-maila neuspješno → sustav bilježi pokušaj i nudi ručno slanje.

UC-005 Automatsko slanje podsjetnika

- **Glavni sudionik:** Sustav
- **Cilj:** Automatski obavijestiti korisnika o odgođenom terminu.
- **Sudionici:** Sustav, e-mail servis
- **Preduvjet:** Postoji zakazan termin koji je odgođen za više od 3 dana.



Osnovni tijek:

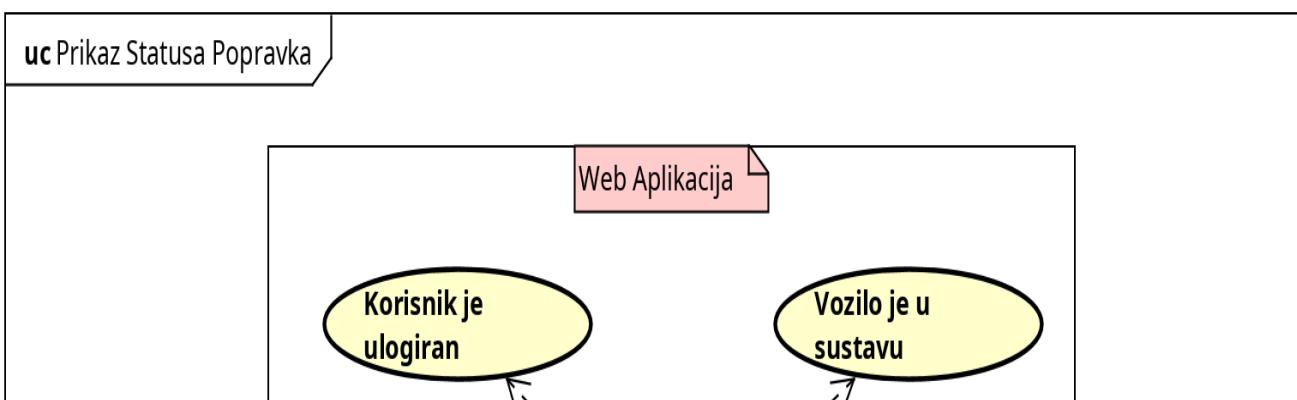
1. Sustav periodički provjerava termine. (F-005)
2. Ako je termin odgođen > 3 dana, sustav generira e-mail.
3. E-mail se automatski šalje korisniku.
4. Sustav bilježi datum i vrijeme obavijesti.

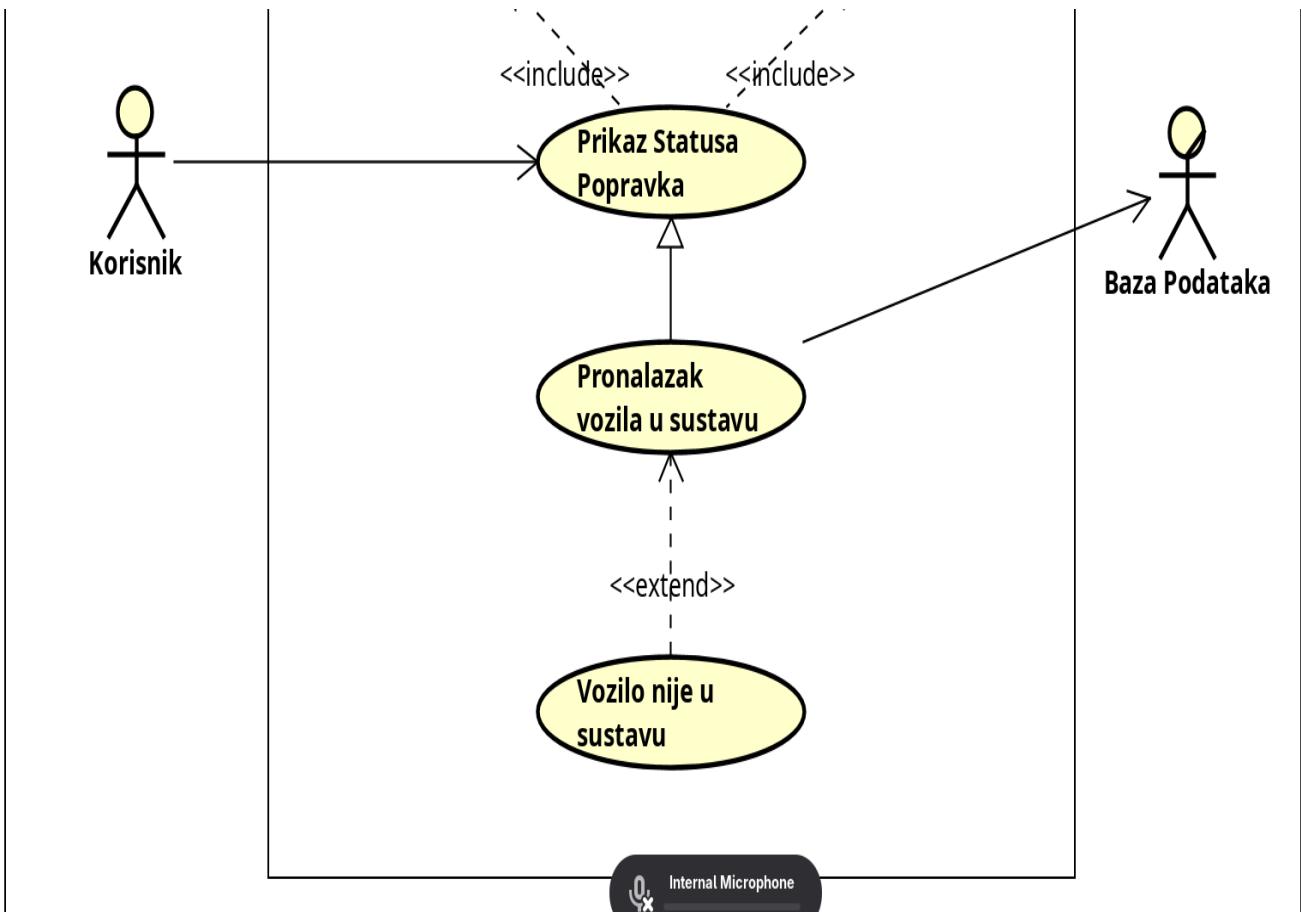
Moguća odstupanja:

- 3a: Neuspjelo slanje e-maila → sustav ponovno pokušava nakon 15 minuta.

UC-06 Prikaz statusa popravka

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Omogućiti korisniku praćenje statusa popravka.
- **Sudionici:** Sustav, serviser
- **Preduvjet:** Vozilo je zaprimljeno u servis.





Osnovni tijek:

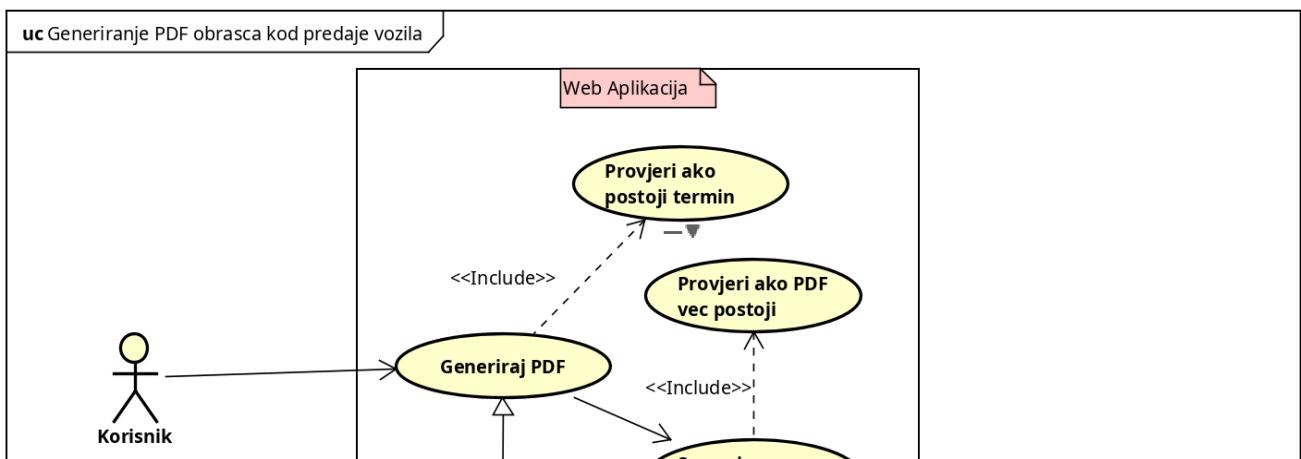
1. Korisnik otvara svoj profil i odabire vozilo. (F-006)
2. Sustav dohvata status popravka.
3. Status se prikazuje (npr. "U tijeku", "Završeno", "Čeka dijelove").
4. Korisnik može pregledati bilješke servisera.

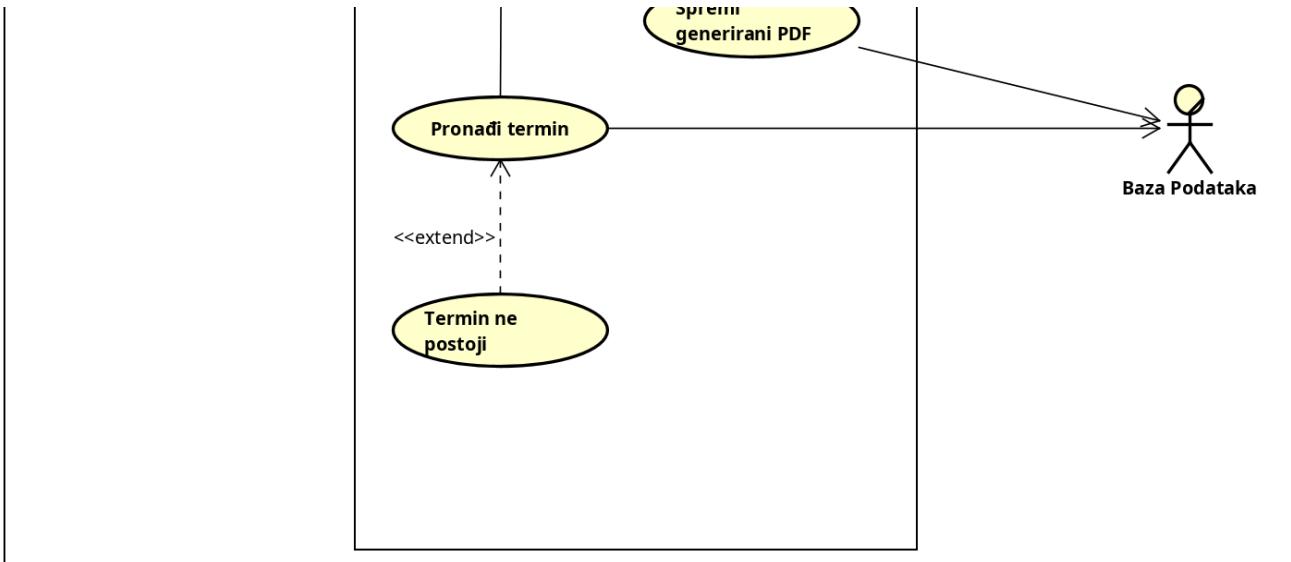
Moguća odstupanja:

- 2a: Povezivanje s bazom neuspješno → prikazuje se posljednji poznati status.

UC-07 Generiranje PDF obrasca kod predaje vozila

- **Glavni sudionik:** Serviser, korisnik
- **Cilj:** Generirati obrazac kod predaje vozila u servis.
- **Sudionici:** Sustav, PDF generator
- **Preduvjet:** Termin servisa postoji.





Osnovni tijek:

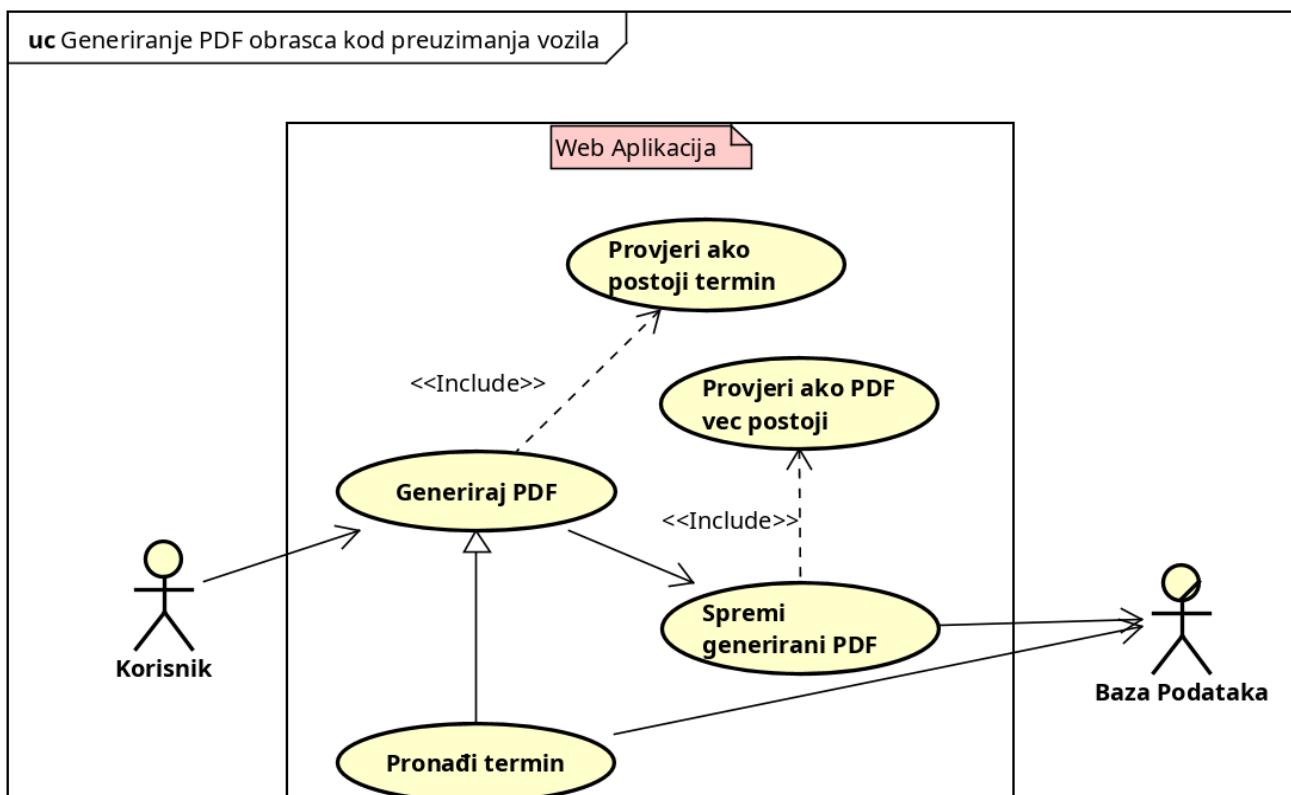
1. Serviser označava vozilo kao "Primljeno". (F-007)
2. Sustav generira PDF obrazac s podacima o korisniku, vozilu i serviseru.
3. Korisnik i serviser potpisuju obrazac.
4. Sustav pohranjuje obrazac u bazu i šalje kopiju korisniku.

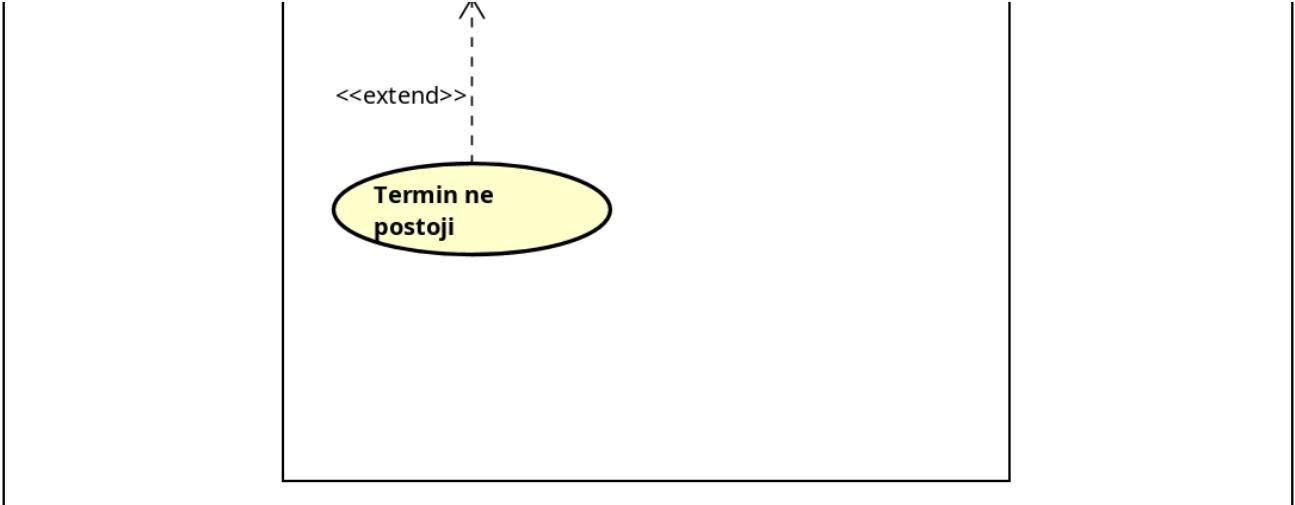
Moguća odstupanja:

- 3a: Elektronički potpis neuspješan → omogućeno ručno potpisivanje.

UC-08 Generiranje PDF obrasca kod preuzimanja vozila

- **Glavni sudionik:** Serviser, korisnik
- **Cilj:** Generirati obrazac kod preuzimanja vozila.
- **Sudionici:** Sustav, PDF generator
- **Preduvjet:** Popravak završen.





Osnovni tijek:

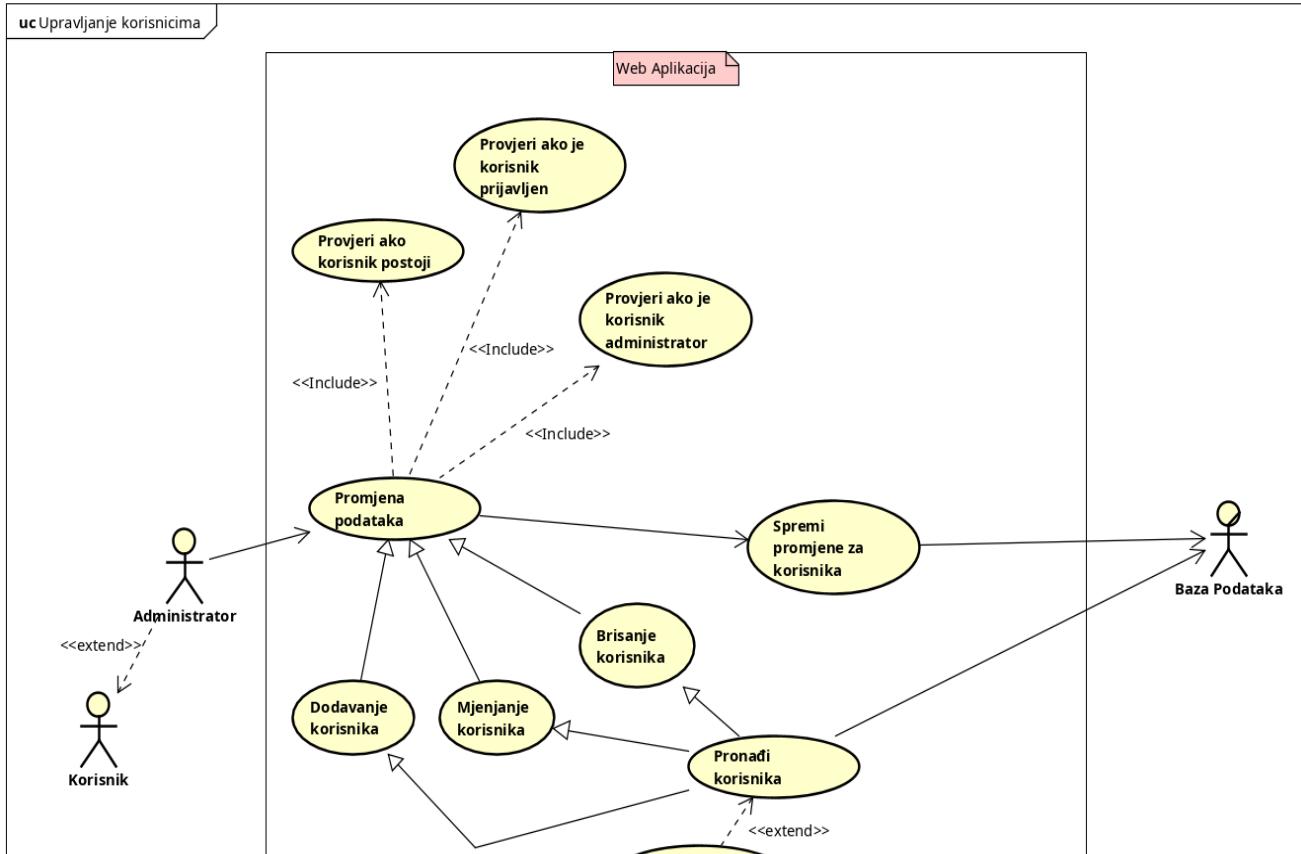
1. Serviser označava vozilo kao "Završeno". (F-008)
2. Sustav generira obrazac o preuzimanju.
3. Korisnik potvrđuje preuzimanje i potpisuje dokument.
4. Sustav spremi obrazac i šalje e-mail potvrdu.

Moguća odstupanja:

- 4a: E-mail nije poslan → sustav ponavlja slanje.

UC-09 Upravljanje korisnicima

- **Glavni sudionik:** Administrator
- **Cilj:** Dodavanje, uređivanje i brisanje korisnika.
- **Sudionici:** Sustav, baza podataka
- **Preduvjet:** Administrator je prijavljen.





Osnovni tijek:

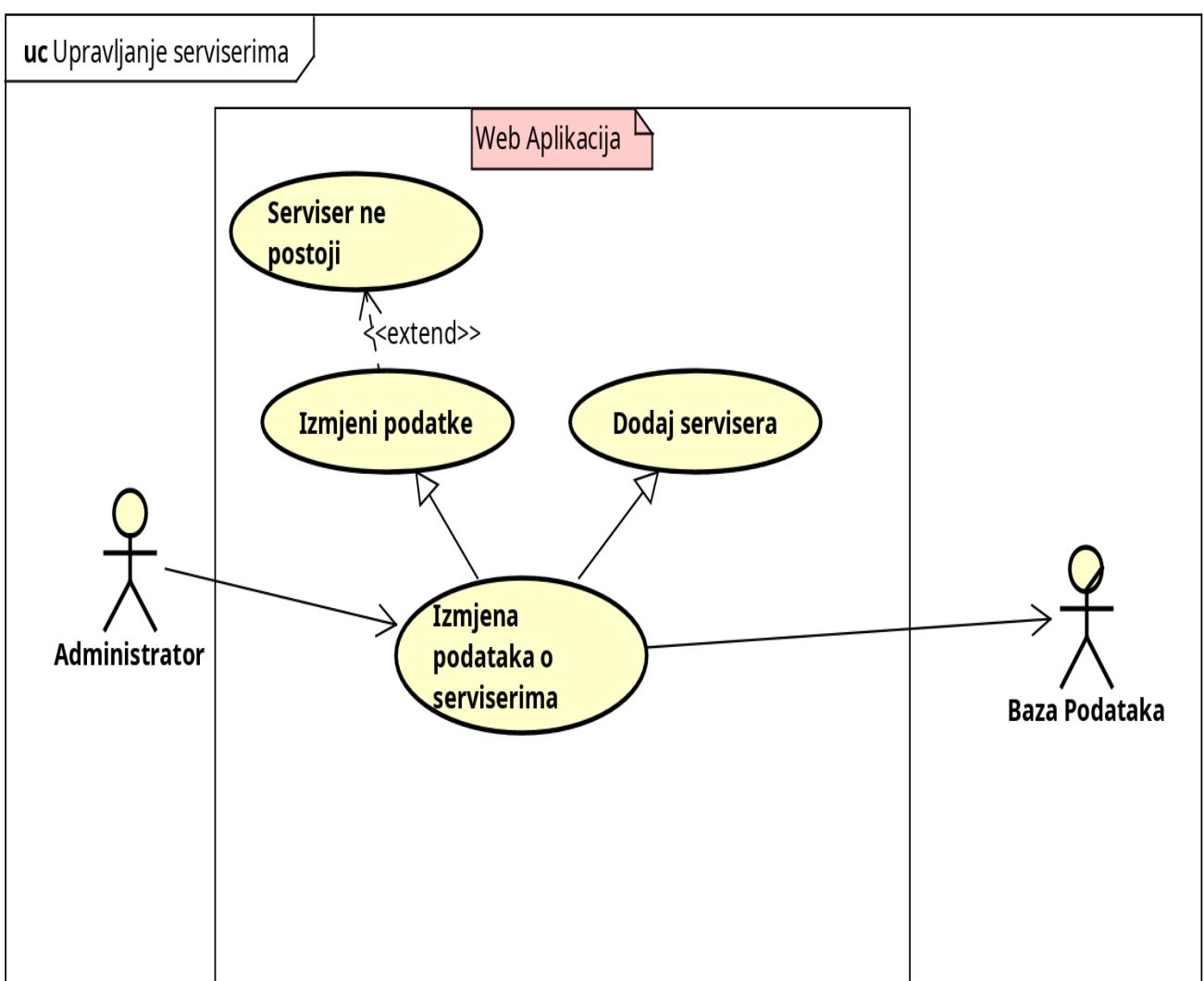
1. Administrator otvara popis korisnika. (F-009)
2. Dodaje novog korisnika ili uređuje postojeći zapis.
3. Sustav validira podatke i sprema promjene.
4. Prikazuje se poruka o uspjehu.

Moguća odstupanja:

- 3a: Neispravni podaci → sustav traži ispravak.

UC-10 Upravljanje serviserima

- **Glavni sudionik:** Administrator
- **Cilj:** Dodavanje i ažuriranje podataka o serviserima.
- **Sudionici:** Sustav, baza podataka
- **Preduvjet:** Administrator je prijavljen.





Osnovni tijek:

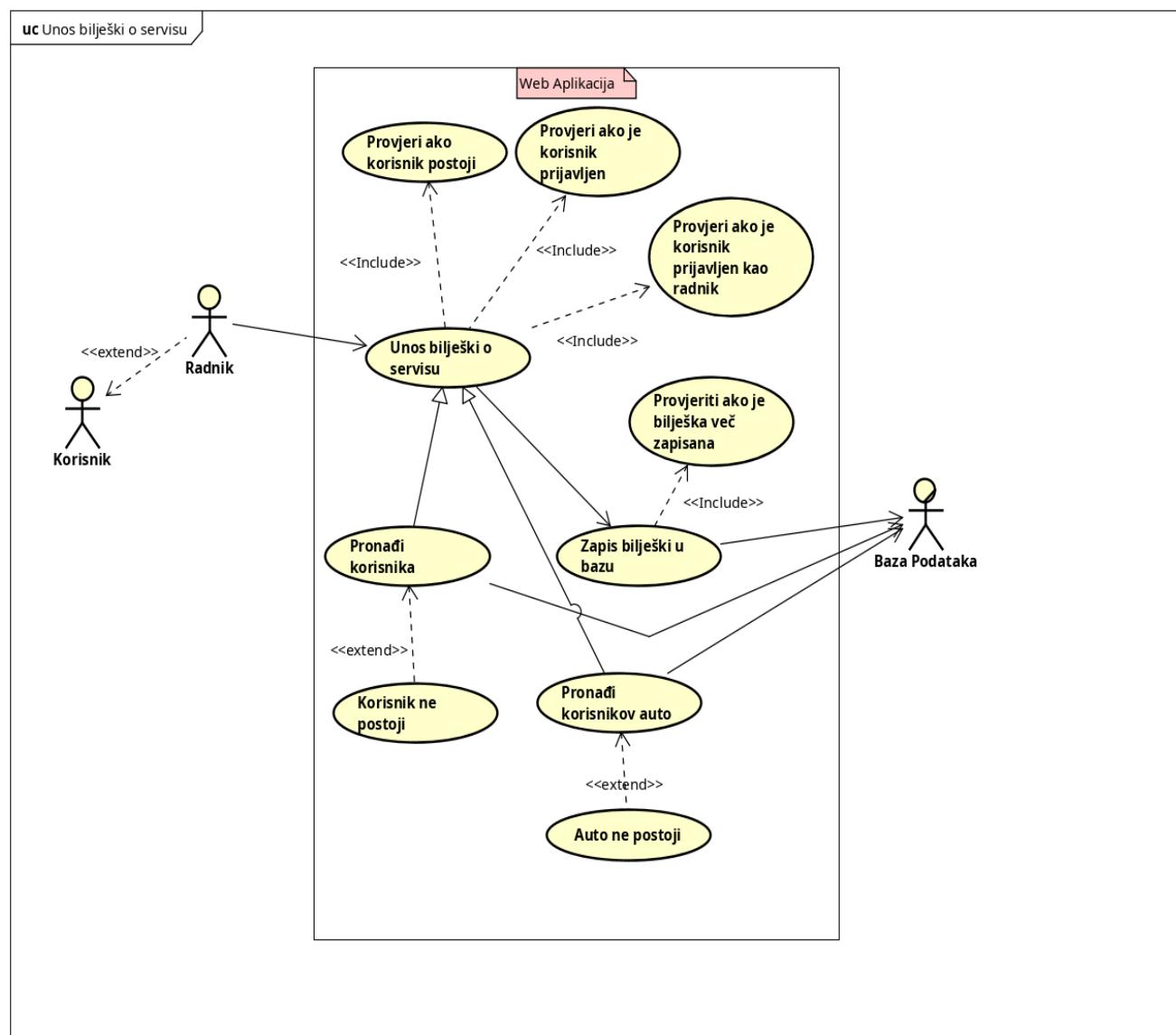
1. Administrator odabire modul "Serviseri". (F-010)
2. Dodaje ili uređuje podatke o serviseru.
3. Sustav pohranjuje izmjene.

Moguća odstupanja:

- 3a: Serviser već postoji → sustav upozorava korisnika.

UC-11 Unos bilješki o servisu

- **Glavni sudionik:** Serviser
- **Cilj:** Zabilježiti napomene i aktivnosti vezane uz servis.
- **Sudionici:** Sustav, baza podataka
- **Preduvjet:** Vozilo je u servisu.



Osnovni tijek:

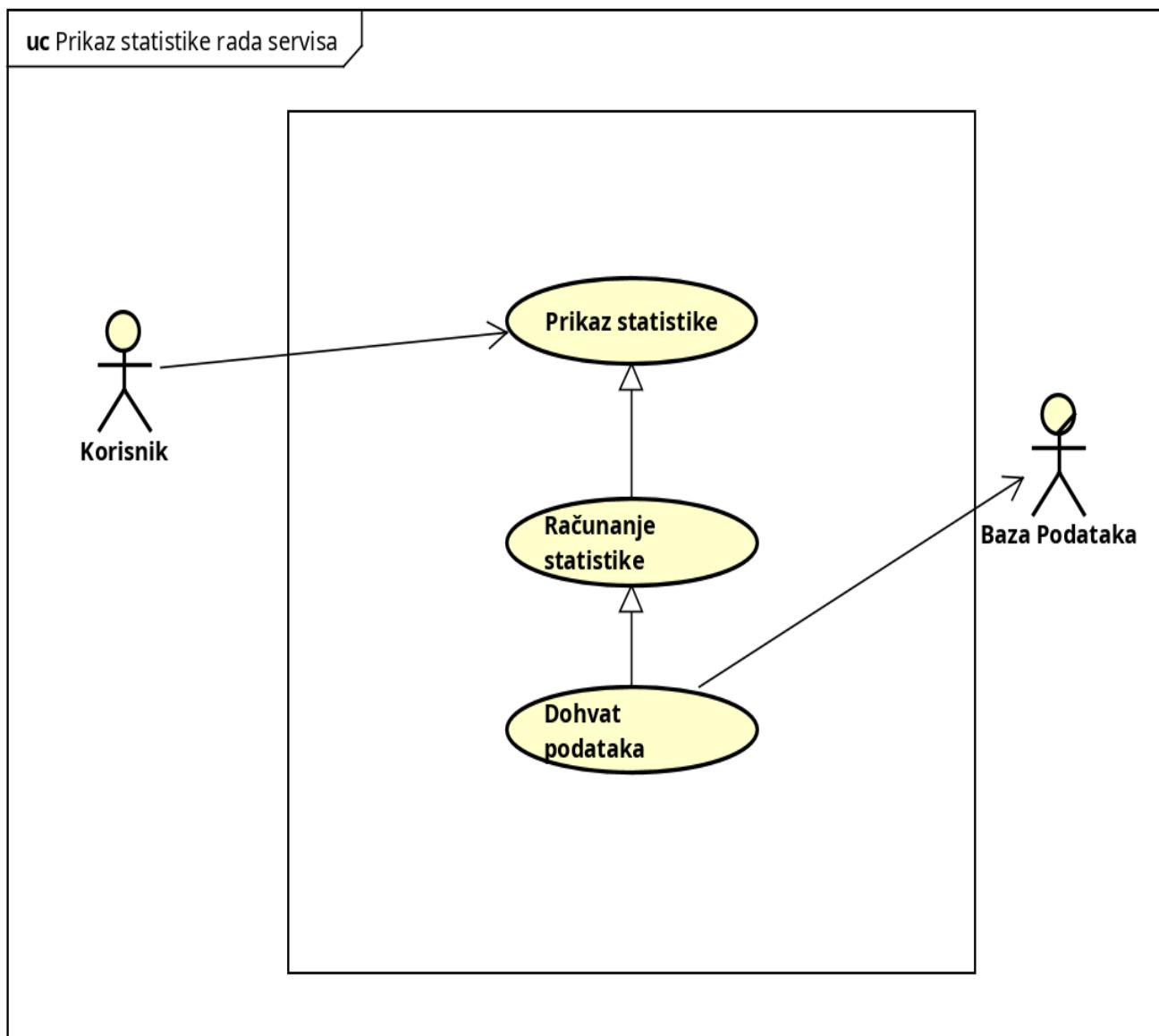
1. Serviser otvara prijavu vozila. (F-011)
2. Dodaje bilješku o radu.
3. Sustav pohranjuje bilješku i povezuje s vozilom.

Moguća odstupanja:

- 3a: Veza s bazom prekinuta → bilješka se sprema lokalno i sinkronizira kasnije.

UC-12 Prikaz statistike rada servisa

- **Glavni sudionik:** Voditelj servisa
- **Cilj:** Pregled poslovne statistike.
- **Sudionici:** Sustav, baza podataka
- **Preduvjet:** Postoje pohranjeni podaci o servisima.

**Osnovni tijek:**

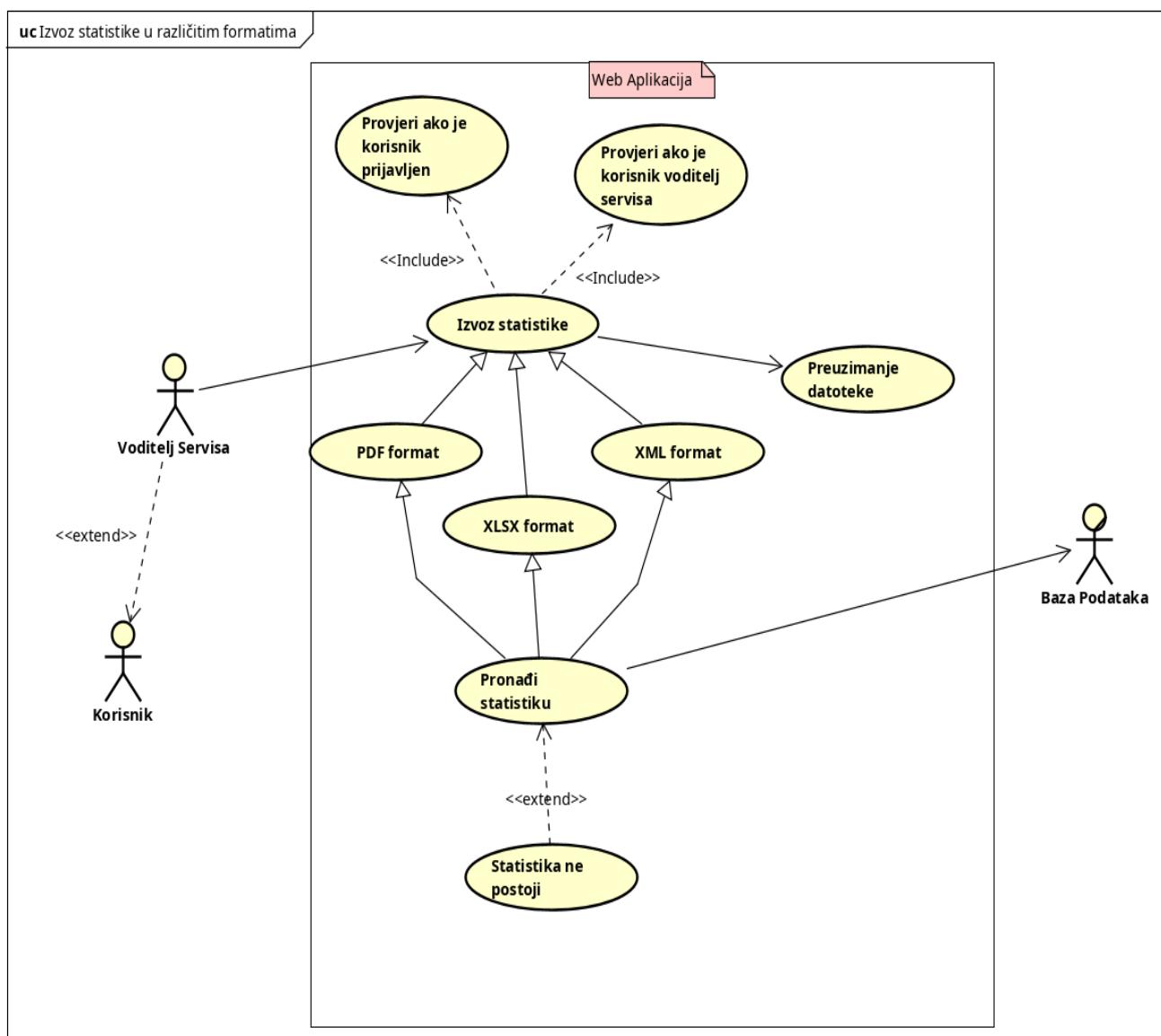
1. Voditelj otvara modul "Statistika". (F-012)
2. Odabire parametre: period, serviser, trajanje.
3. Sustav generira grafički prikaz.
4. Statistika je prikazana na ekranu.

Moguća odstupanja:

- 3a: Nema dostupnih podataka → prikazuje se poruka "Nema rezultata".

UC-13 Izvoz statistike u različitim formatima

- **Glavni sudionik:** Voditelj servisa
- **Cilj:** Izvoz podataka u PDF, XLSX ili XML.
- **Sudionici:** Sustav, generator izvještaja
- **Preduvjet:** Postoji generirana statistika.

**Osnovni tijek:**

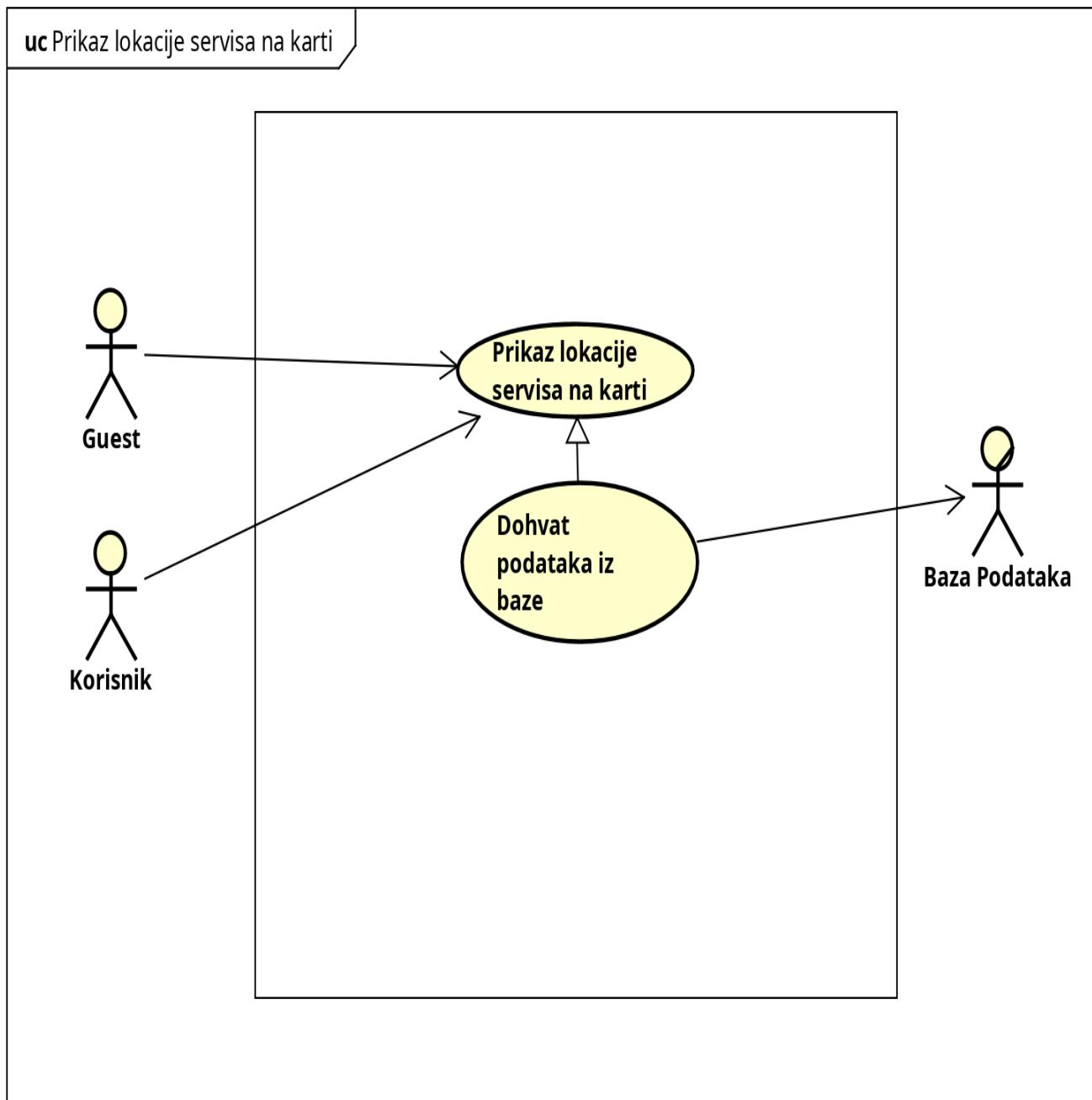
1. Voditelj odabire opciju "Izvoz". (F-013)
2. Bira format (PDF, XML, XLSX).
3. Sustav generira datoteku i omogućuje preuzimanje.

Moguća odstupanja:

- 3a: Generiranje neuspješno → prikazuje se poruka o grešci.

UC-14 Prikaz lokacije servisa na karti

- **Glavni sudionik:** Svi korisnici
- **Cilj:** Prikazati lokaciju servisa putem Google Maps servisa.
- **Sudionici:** Sustav, Google Maps API
- **Preduvjet:** Internet veza dostupna.

**Osnovni tijek:**

1. Korisnik otvara stranicu "Lokacija servisa". (F-014)
2. Sustav učitava kartu pomoću Google Maps API-ja.
3. Prikazuje se marker s adresom servisa.

Moguća odstupanja:

- 2a: Google API nije dostupan → prikazuje se statična karta s adresom.

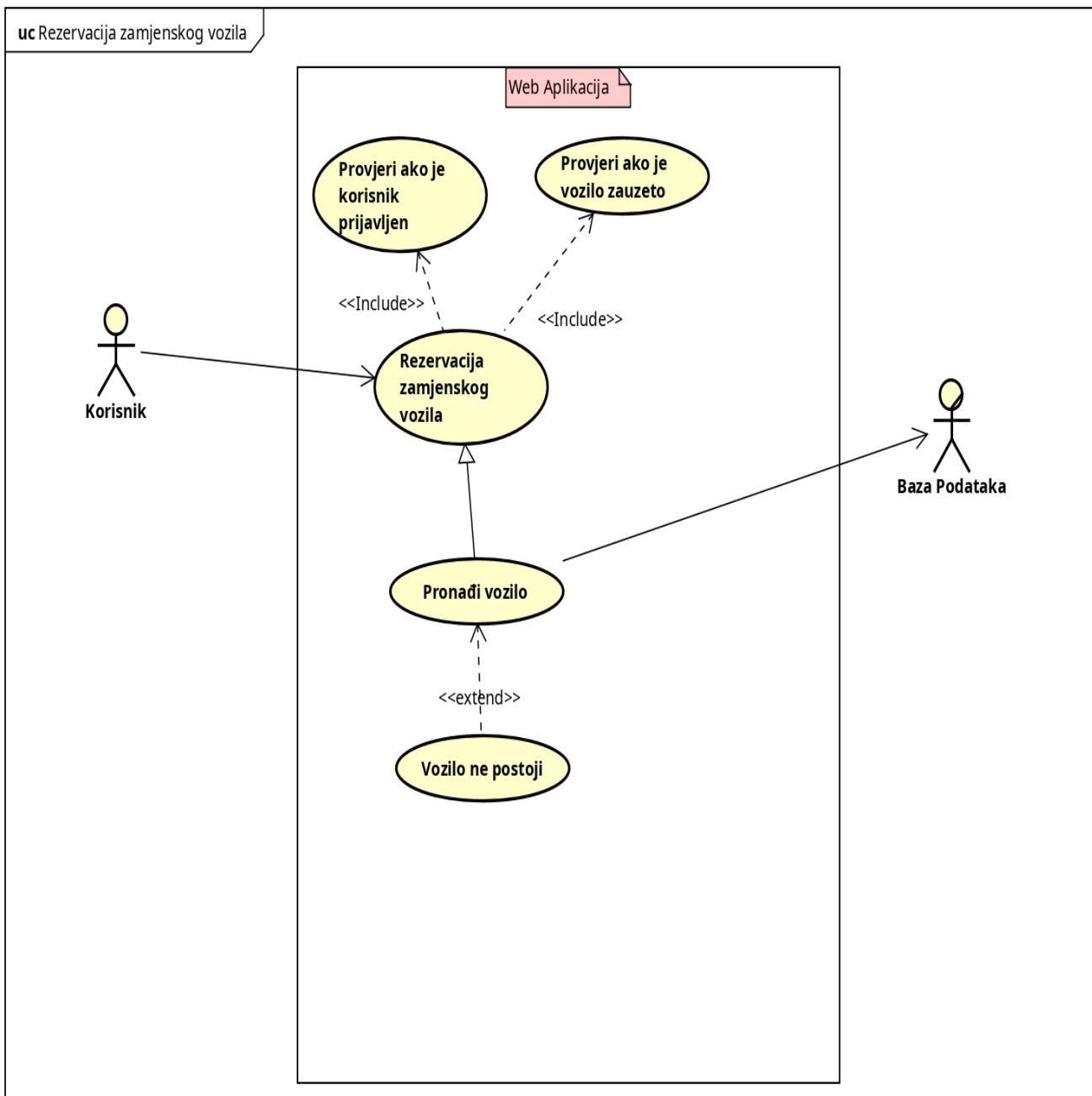
UC-15 Rezervacija zamjenskog vozila

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Omogućiti rezervaciju zamjenskog vozila uz termin servisa

• Upravljači i rezervacija zamjenskog vozila uz termini koji vise.

- **Sudionici:** Sustav, baza podataka

- **Preduvjet:** Postoji slobodno zamjensko vozilo.



Osnovni tijek:

1. Korisnik odabire opciju "Rezerviraj zamjensko vozilo". (F-015)
2. Sustav prikazuje popis dostupnih vozila.
3. Korisnik odabire vozilo i potvrđuje rezervaciju.
4. Sustav pohranjuje rezervaciju i povezuje je s terminom.

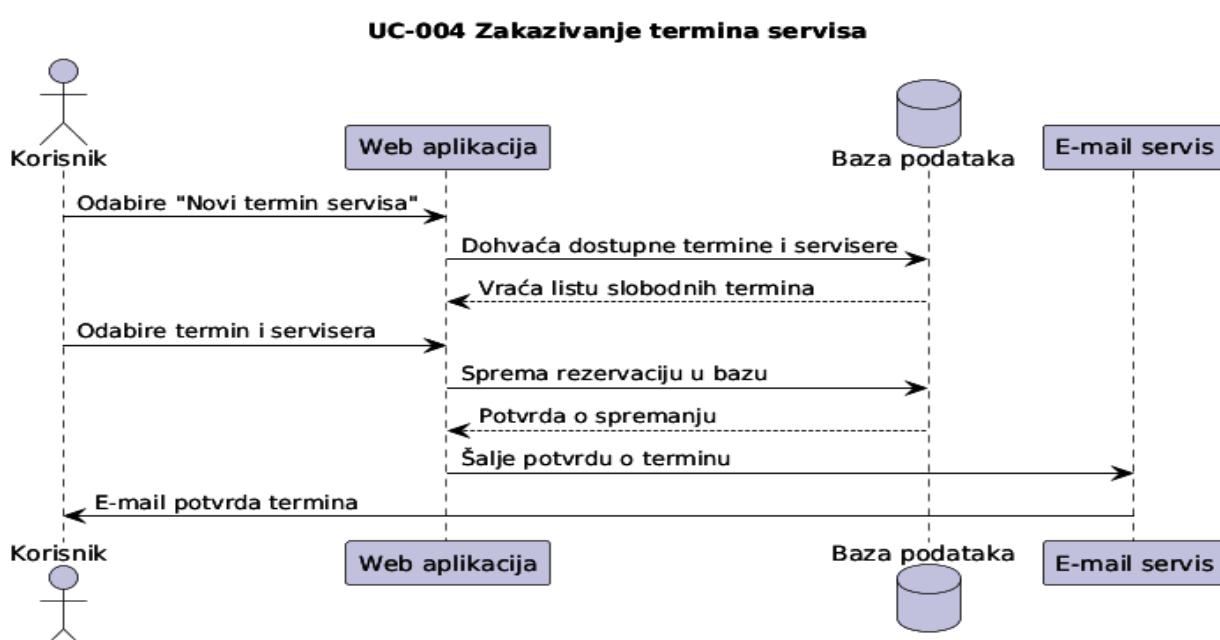
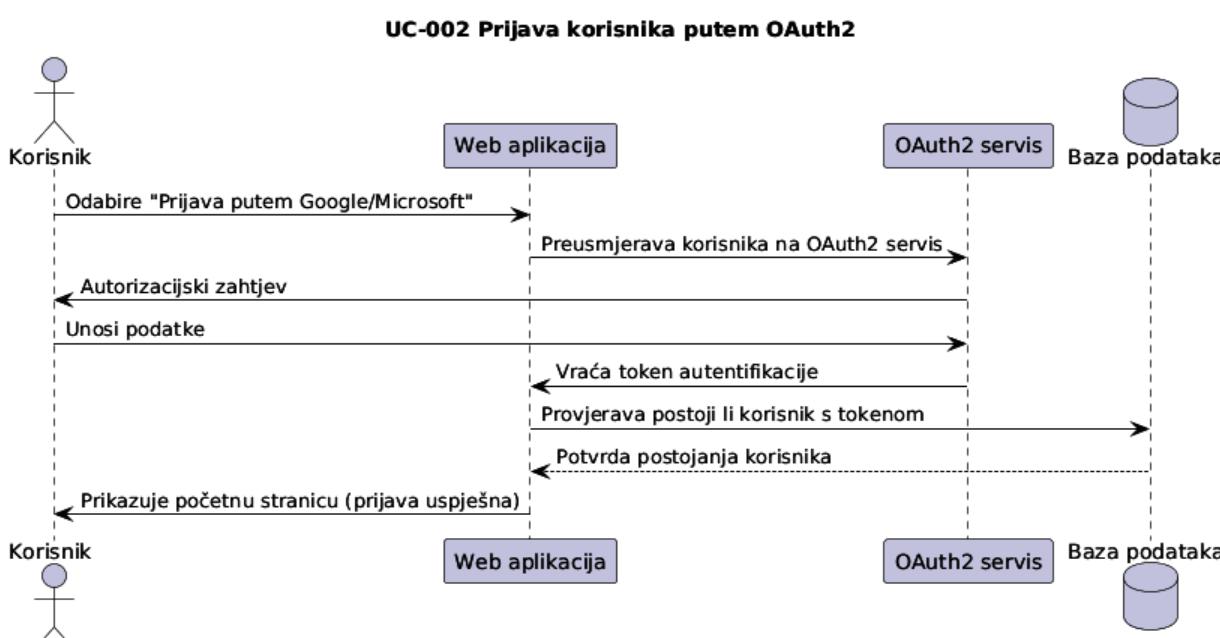
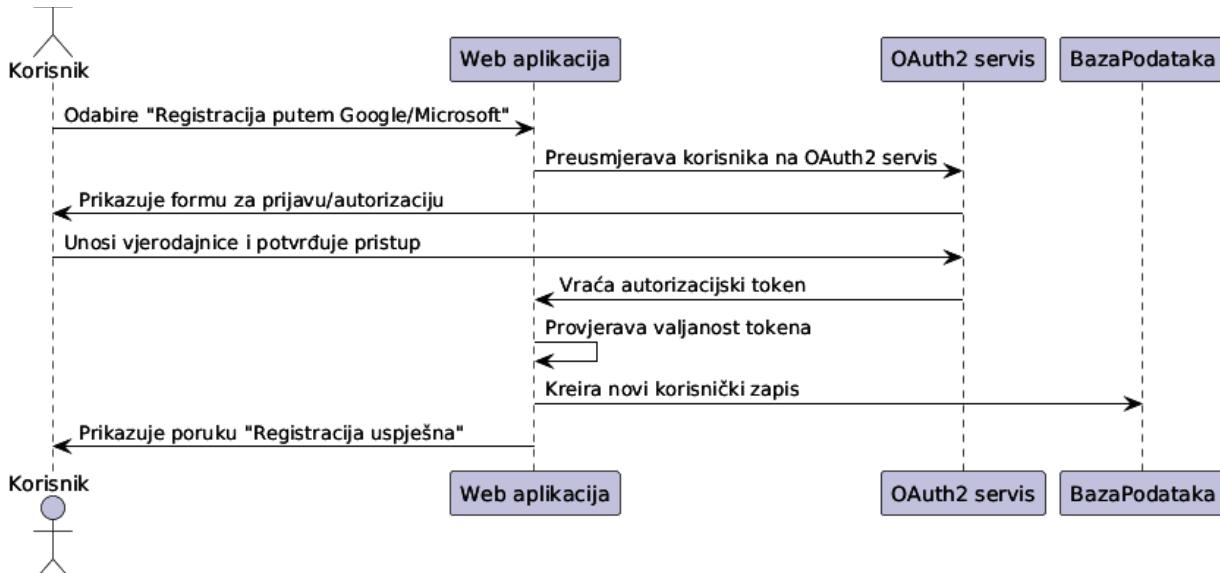
Moguća odstupanja:

- 2a: Nema slobodnih vozila → korisniku se prikazuje poruka "Trenutno nema dostupnih vozila".

Sekvencijski dijagrami

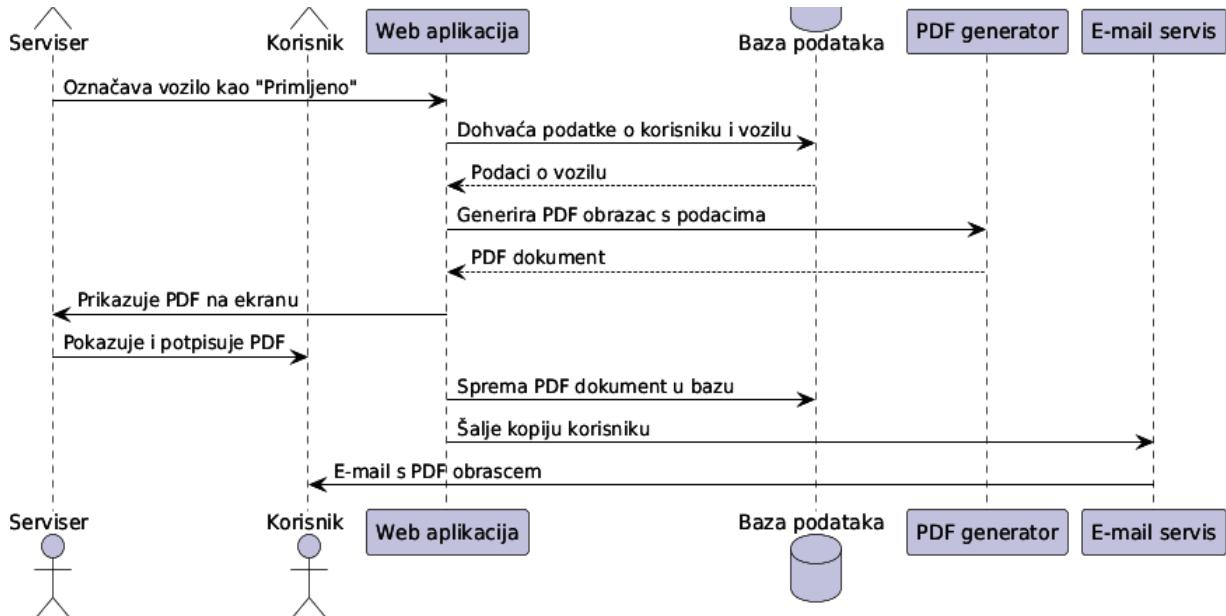
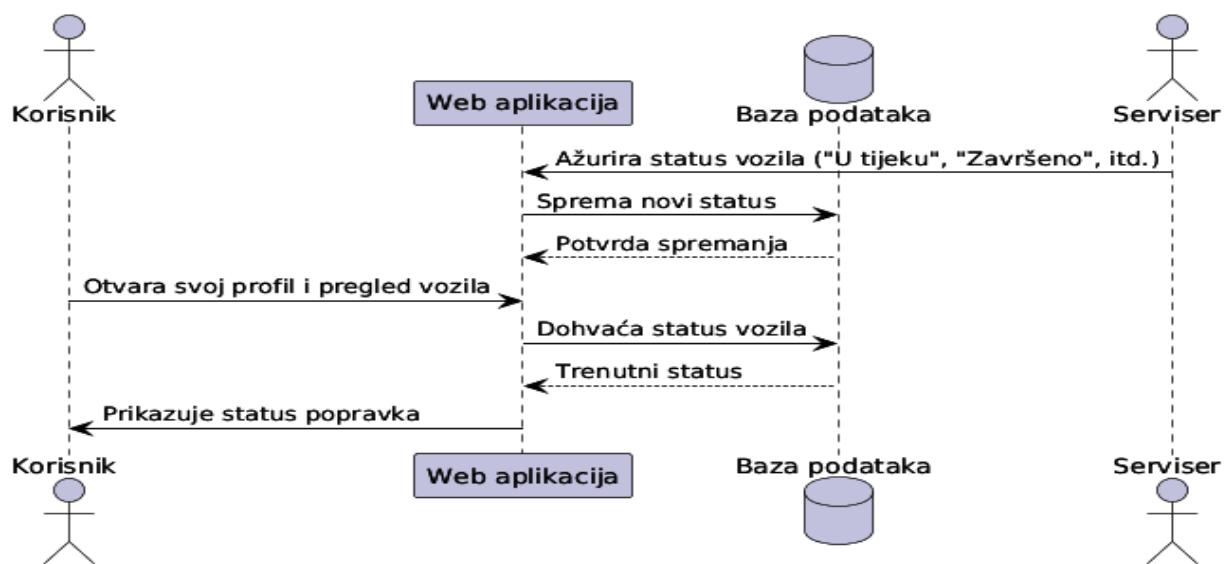
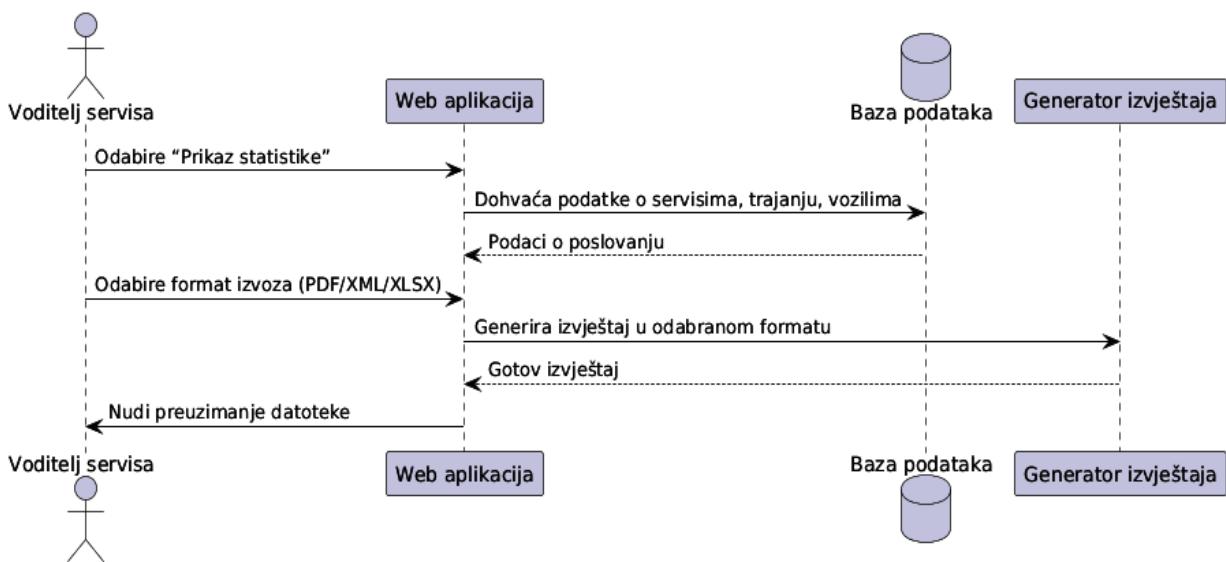
UC-001 Registracija korisnika putem OAuth2





UC-007 Generiranje PDF obrasca kod predaje vozila



**UC-006 Praćenje statusa popravka****UC-013 Izvoz statistike u PDF/XML/XLSX**

Nacrtati sekvencijske dijagrame koji modeliraju najvažnije dijelove sustava. Ukoliko postoji nedoumica oko odabira, razjasniti s asistentom. Uz svaki dijagram napisati detaljni opis dijagrama.

| izvorne kodove dijagrama pohranjujite u svom GIT-u

Provjera uključenosti ključnih funkcionalnosti u obrasce uporabe

Funkcionalnost	Opis	Obuhvaćeni UC
Registracija korisnika	Autentifikacija putem OAuth2	UC-01
Prijava korisnika	Prijava u sustav putem OAuth2	UC-02
Upravljanje vozilima	Unos i pregled podataka o vozilu	UC-03
Zakazivanje servisa	Odabir termina i servisera	UC-04
Automatski podsjetnici	E-mail podsjetnik kod odgode	UC-05
Praćenje statusa	Uvid u status popravka	UC-06
Predaja vozila	Generiranje PDF obrasca kod prijema	UC-07
Preuzimanje vozila	Generiranje PDF obrasca kod završetka	UC-08
Upravljanje korisnicima	Administracija korisnika	UC-09
Upravljanje serviserima	Administracija servisera	UC-10
Bilješke servisera	Evidencija aktivnosti	UC-11
Statistika rada	Analiza poslovanja	UC-12
Izvoz statistike	Izvoz u PDF, XML, XLSX	UC-13
Lokacija servisa	Prikaz Google Maps karte	UC-14
Zamjensko vozilo	Rezervacija dostupnog vozila	UC-15

upotreba alata ChatGPT za formatiranje te nadopunu određenih dijelova teksta

The screenshot shows a Microsoft Word document with the following structure:

- Pages: 14
- Find a page or section...
- Home
- 1. Opis projektnog zadatka
- 2. Analiza zahtjeva
- 3. Specifikacija zahtjeva sustava
 - Obrasci uporabe i Dijagrami obrazaca uporabe
 - Opis obrazaca uporabe
 - UC-01 Registracija korisnika
 - UC-02 Prijava korisnika
 - UC-03 Unos vozila
 - UC-04 Zakazivanje termina servisa
 - UC-005 Automatsko slanje podsjetnika
 - UC-06 Prikaz statusa popravka
 - UC-07 Generiranje PDF obrasca kod predaje vozila
 - UC-08 Generiranje PDF obrasca kod preuzimanja vozila
 - UC-09 Upravljanje korisnicima
 - UC-10 Upravljanje serviserima

UC-14 Prikaz lokacije servisa na karti
UC-15 Rezervacija zamjenskog vozila
Sekvencijski dijagrami
Provjera uključenosti ključnih funkcionalnosti u obrasce uporabe

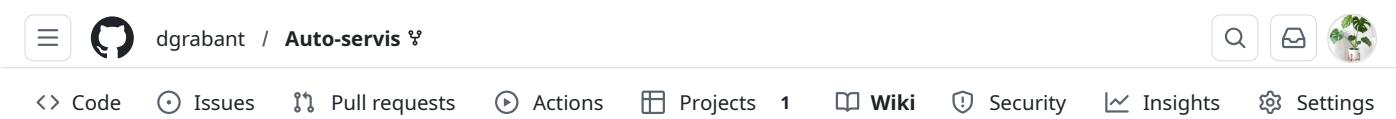
- ▶ [4. Arhitektura i dizajn sustava](#)
- ▶ [5. Arhitektura komponenata i razmještaja](#)
- ▶ [6. Ispitivanje programskog rješenja](#)
- ▶ [7. Tehnologije za implementaciju aplikacije](#)
- ▶ [8. Upute za puštanje u pogon](#)
- ▶ [9. Zaključak i budući rad](#)
- ▶ [A. Popis literature](#)
- ▶ [A. Dnevnik promjena dokumentacije](#)
- ▶ [B. Prikaz aktivnosti grupe](#)
- ▶ [D. Stranica za sastanke](#)

+ Add a custom sidebar

Clone this wiki locally

<https://github.com/dgrabant/Auto-servis.wiki.git>





4. Arhitektura i dizajn sustava

[Edit](#) [New page](#)

[Jump to bottom](#)

David edited this page 4 hours ago · [16 revisions](#)

Arhitektura sustava

Opis arhitekture

Aplikacija za upravljanje auto servisom razvijena je kao web sustav s arhitekturom klijent–poslužitelj. Backend je izrađen u Spring Boot (Java) tehnologiji, a frontend koristi Node.js okruženje te Three.js biblioteku za 3D prikaz modela vozila.

Sustav omogućuje korisnicima prijavu vozila, praćenje statusa popravka i rezervaciju zamjenskih vozila.

Zaposlenici servisa imaju pristup alatima za organizaciju posla i evidenciju zaprimljenih vozila.

Aplikacija je postavljena u oblaku putem render.com servisa, što omogućuje stalnu dostupnost i skalabilnost.

Stil arhitekture

Odabrana je slojovita arhitektura koja razdvaja frontend i backend dijelove.

Unutar backenda primijenjen je MVC (Model–View–Controller) pristup, čime je omogućeno jasno razdvajanje odgovornosti i lakše održavanje sustava.

Docker okruženje koristi se za jednostavno pokretanje i izolaciju aplikacijskih servisa.

Baza podataka je implementirana u Postgres-u i povezana je sa Spring Boot backendom putem Spring Data JPA.

Glavne komponente sustava

- **Frontend (Node.js + Three.js)**

Klijentska aplikacija izrađena u JavaScriptu, pokretana na Node.js poslužitelju.

Komunicira s backendom putem REST API poziva i koristi Three.js za prikaz i uređivanje 3D modela vozila.

Sučelje je responzivno i prilagođeno stolnim i mobilnim uređajima.

- **Backend (Spring Boot)**

Implementira poslovnu logiku, autentifikaciju korisnika, upravljanje vozilima, terminima i zamjenskim vozilima te generiranje izvještaja.

Organiziran je prema MVC principu:

- Model – definira entitete (vozilo, korisnik, serviser, termin)
- View - ne sadrži elemente jer backend služi kao API prema frontendu
- Controller – obrađuje HTTP zahtjeve i vraća JSON odgovore

- **Baza podataka (Postgres)**

Pohranjuje sve podatke o korisnicima, vozilima, servisnim prijavama, zamjenskim vozilima i terminima.

- **Autentifikacija (OAuth2)**

Koristi se vanjski OAuth2 servis za registraciju i prijavu korisnika.

Za provjeru identiteta i pristup API-ju koristi se JWT token.

- **Izvoz podataka**

Sustav omogućuje izvoz podataka u PDF, XML i XLSX formatima.

Voditelj servisa može generirati izvještaje o popravcima, zauzeću vozila i resursima.

- **Infrastruktura (render.com + Docker)**

Aplikacija je postavljena na render.com poslužitelj, a Docker omogućuje kontejnerizaciju i jednostavno upravljanje implementacijom.

Mrežni protokoli

Komunikacija između frontenda i backenda odvija se putem HTTPS protokola i REST API-ja.

Podaci se razmjenjuju u JSON formatu, dok OAuth2 autentifikacija također koristi siguran HTTPS kanal.

Globalni upravljački tok

1. Prijava i registracija korisnika – putem OAuth2 servisa; korisnik dobiva JWT token.
2. Unos i obrada podataka – korisnik unosi podatke o vozilu i bira termin; backend ih pohranjuje u bazu.

3. Rezervacija i obavijesti – sustav provjerava dostupnost zamjenskog vozila i šalje potvrdu e-mailom.
4. Praćenje statusa – korisnik može pratiti napredak popravka putem web sučelja.
5. Generiranje obrazaca – prilikom preuzimanja vozila generira se PDF dokument.
6. Analitika i izvještaji – administrator i voditelj servisa imaju uvid u statističke podatke.

Obrazloženje odabira arhitekture

Odabrana arhitektura klijent–poslužitelj sa slojevitim MVC modelom omogućuje modularnost, sigurnost i jednostavno održavanje.

Spring Boot je odabran zbog stabilnosti i podrške za integraciju s vanjskim servisima, dok Node.js i Three.js omogućuju dinamičan frontend s 3D prikazima.

Render.com i Docker omogućuju jednostavnu skalabilnost, a Postgres osigurava pouzdanu pohranu podataka. OAuth2 autentifikacija povećava sigurnost i omogućuje povezivanje s vanjskim korisničkim servisima.

Alternativna monolitna arhitektura bila bi jednostavnija, ali bi otežala održavanje i nadogradnju.

Slojni pristup omogućuje buduće proširenje sustava, poput izdvajanja modula za rezervacije u zaseban servis.

Organizacija sustava na visokoj razini

- **Frontend (klijent):**

Web aplikacija izrađena u Node.js i Three.js, responzivna i prilagođena različitim uređajima. Omogućuje prijavu, unos vozila, praćenje statusa i pregled resursa.

- **Backend (poslužitelj):**

Spring Boot aplikacija implementira REST API, obrađuje zahtjeve i komunicira s bazom podataka. Upravljanje autentifikacijom se odvija putem OAuth2.

- **Baza podataka:**

Postgres relacijska baza povezana sa Spring Data JPA slojem.

- **Vanjski servisi:**

- OAuth2 za prijavu korisnika
- Google Maps API za prikaz lokacije servisa

Baza podataka

U projektu je korištena relacijska baza podataka implementirana pomoću sustava za upravljanje bazama podataka Postgres. Relacijski model omogućuje jednostavno povezivanje tablica putem primarnih i stranih ključeva te osigurava integritet i dosljednost podataka.

Baza se sastoji od više međusobno povezanih entiteta: korisnik, vozilo, vrstavozila, usluga i dijelovi. Svaka tablica sadrži svoje atribute, a veze između njih definirane su preko stranih ključeva, čime je omogućena normalizacija podataka i izbjegavanje redundantnosti.

Opis tablica

Tablica: korisnik

Atribut	Tip podatka	Opis varijable
idkorisnik	bigint (PK)	Primarni ključ, jedinstveni identifikator korisnika
ime	character varying(50)	Ime korisnika
prezime	character varying(50)	Prezime korisnika
email	character varying(100)	Email adresa korisnika
dovoljeni_usluga	character varying(20)	Status korisnika (npr da li pruža usluge)

Uvodljivost u tablicu	Opis tablice	Opis korisnika (npr. učni ili pruža usluge)
uloga	character varying(10)	Uloga korisnika u sustavu (npr. korisnik, admin)

Tablica: vozilo

Atribut	Tip podatka	Opis varijable
idvozila	serial (PK)	Primarni ključ, jedinstveni identifikator vozila
idkorisnik	integer (FK)	Strani ključ koji povezuje vozilo s korisnikom
idvrsta	integer (FK)	Strani ključ koji povezuje vozilo s vrstom vozila
regoznaka	character varying(10)	Registracijska oznaka vozila
godinaproizvodnje	smallint	Godina proizvodnje vozila
serijskibr	character varying(100)	Serijski broj vozila
jezamjensko	boolean	Označava je li vozilo zamjensko

Tablica: vrstavozila

Atribut	Tip podatka	Opis varijable
idvrsta	serial (PK)	Primarni ključ, jedinstveni identifikator vrste vozila
nazivmodela	character varying(50)	Naziv modela vozila
opisvrste	character varying(100)	Opis vrste vozila

Tablica: usluga

Atribut	Tip podatka	Opis varijable
idusluge	bigserial (PK)	Primarni ključ, jedinstveni identifikator usluge
naziv	character varying(50)	Naziv usluge
cijena	double precision	Cijena usluge
opis	character varying(500)	Detaljan opis usluge
slikaurl	character varying(200)	URL slike koja prikazuje uslugu

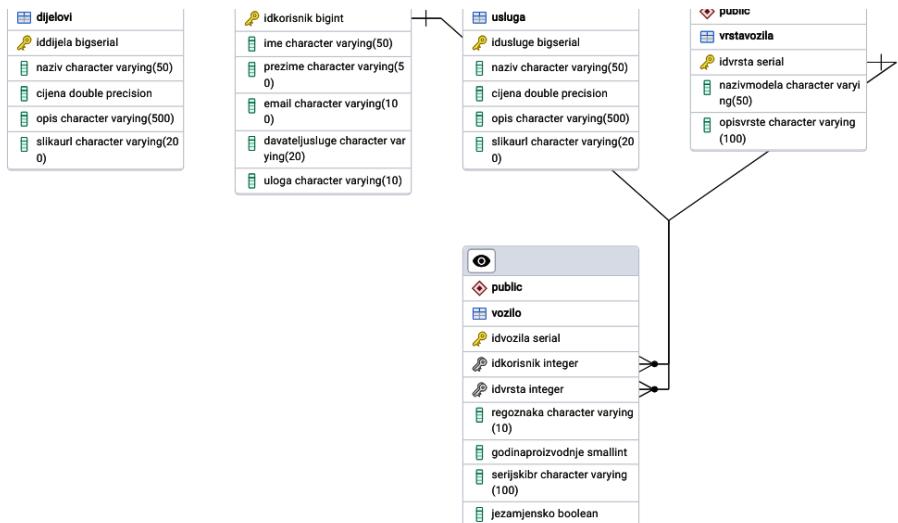
Tablica: dijelovi

Atribut	Tip podatka	Opis varijable
iddjela	bigserial (PK)	Primarni ključ, jedinstveni identifikator dijela
naziv	character varying(50)	Naziv dijela
cijena	double precision	Cijena dijela
opis	character varying(500)	Opis dijela
slikaurl	character varying(200)	URL slike dijela

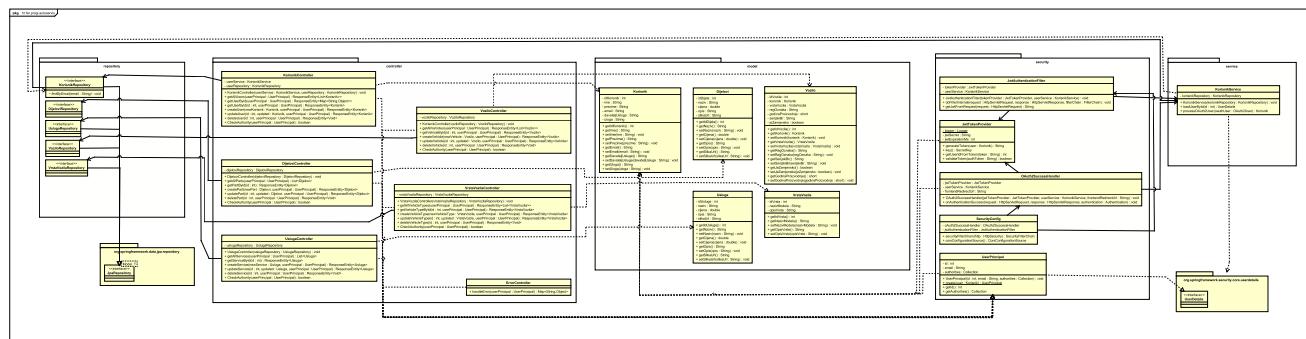
Dijagram baze podataka

Na slici ispod prikazan je ER dijagram baze podataka. Dijagram prikazuje odnose između tablica, uključujući primarne i strane ključeve.

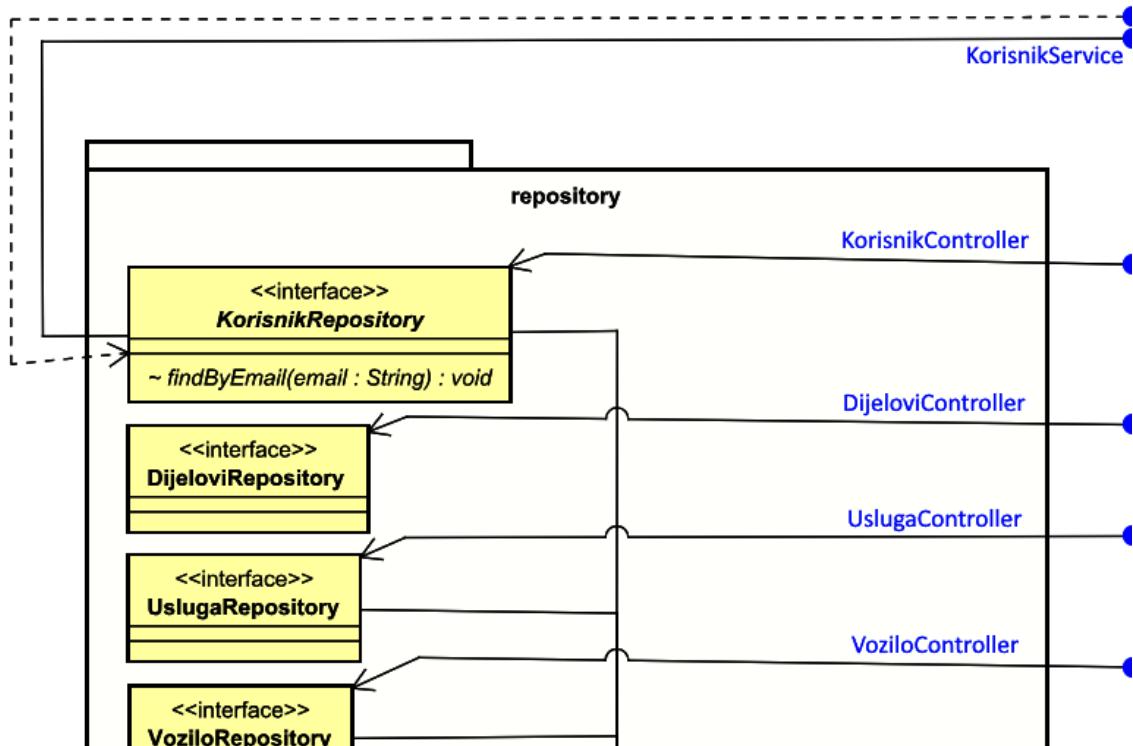


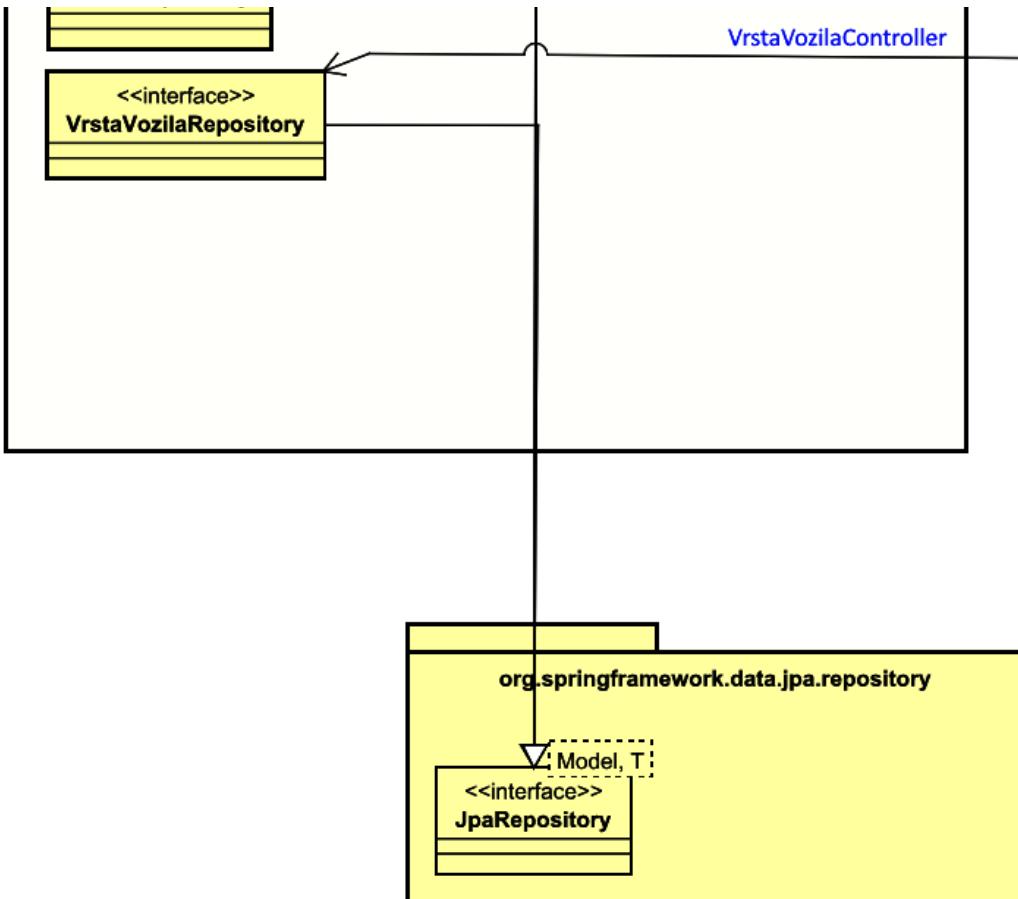


Dijagram razreda

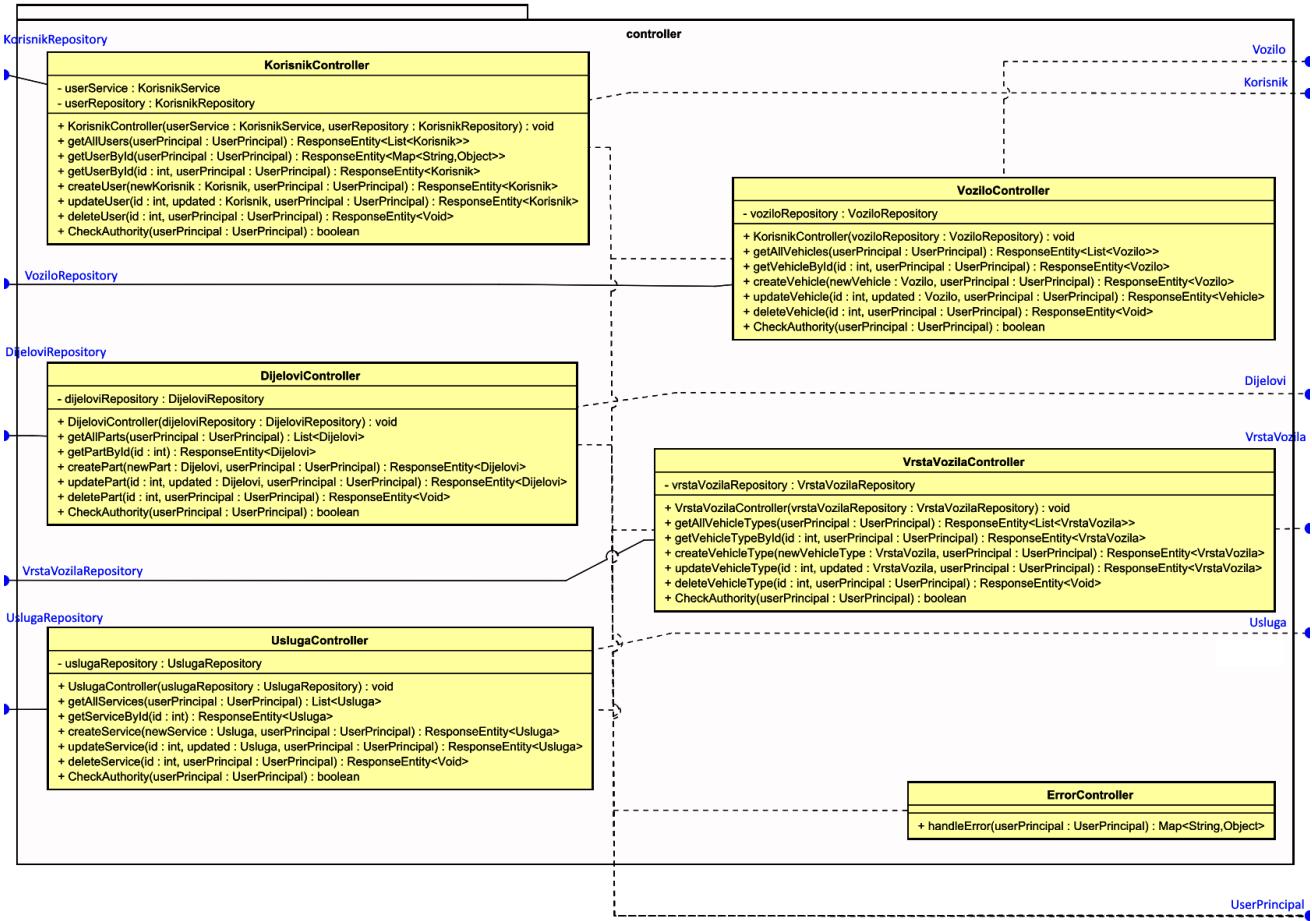


Kategorija Repository

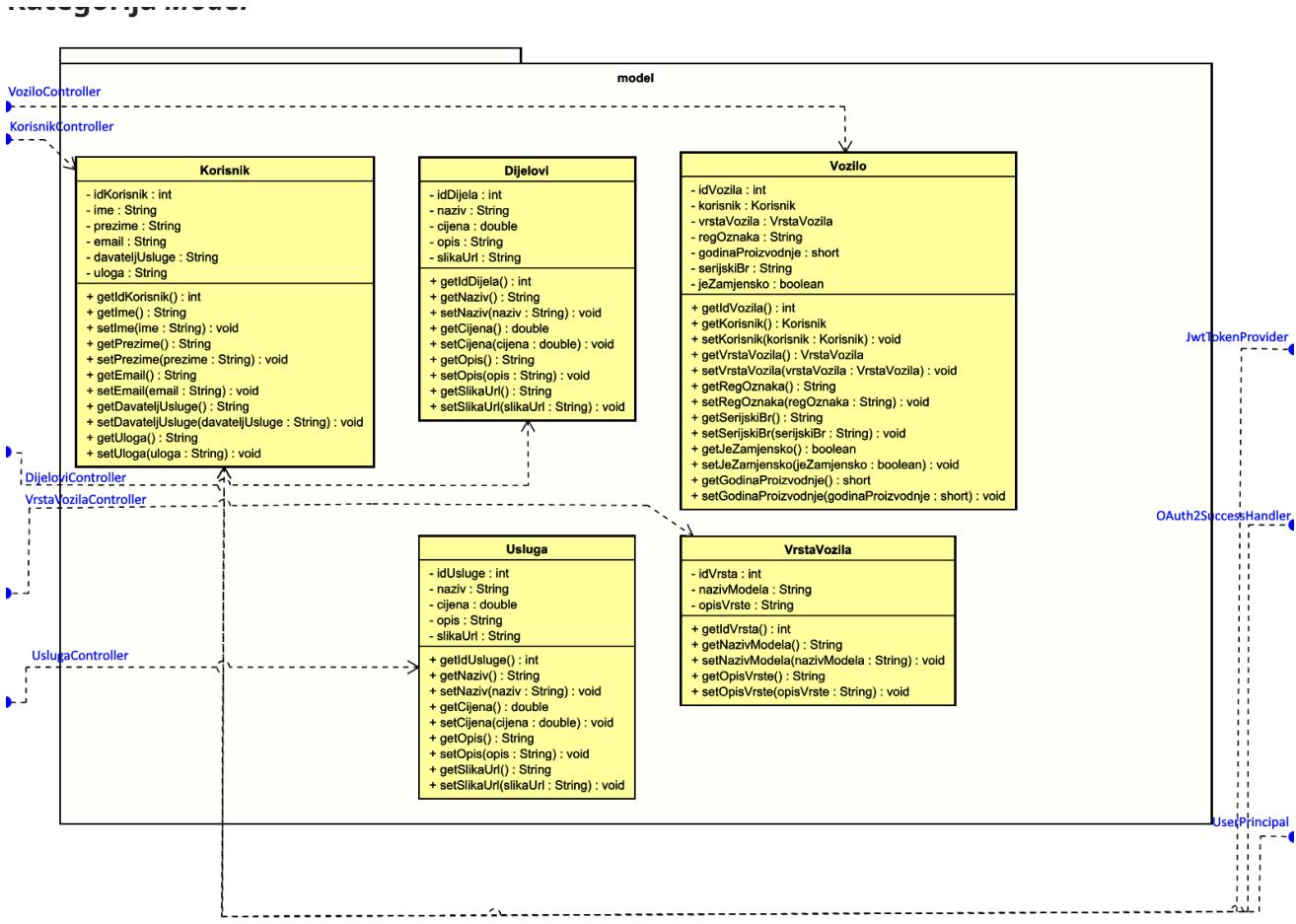




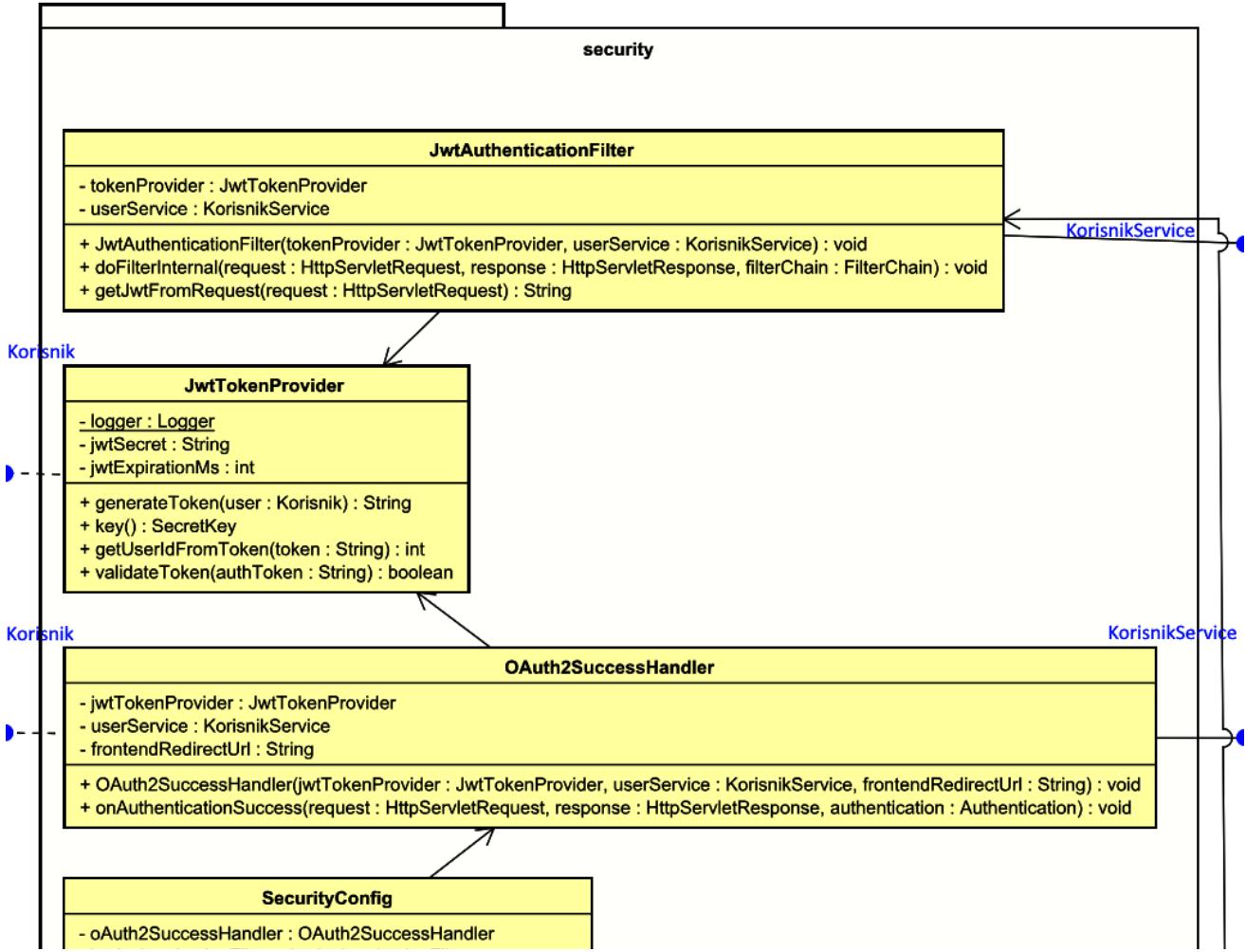
Kategorija Controller

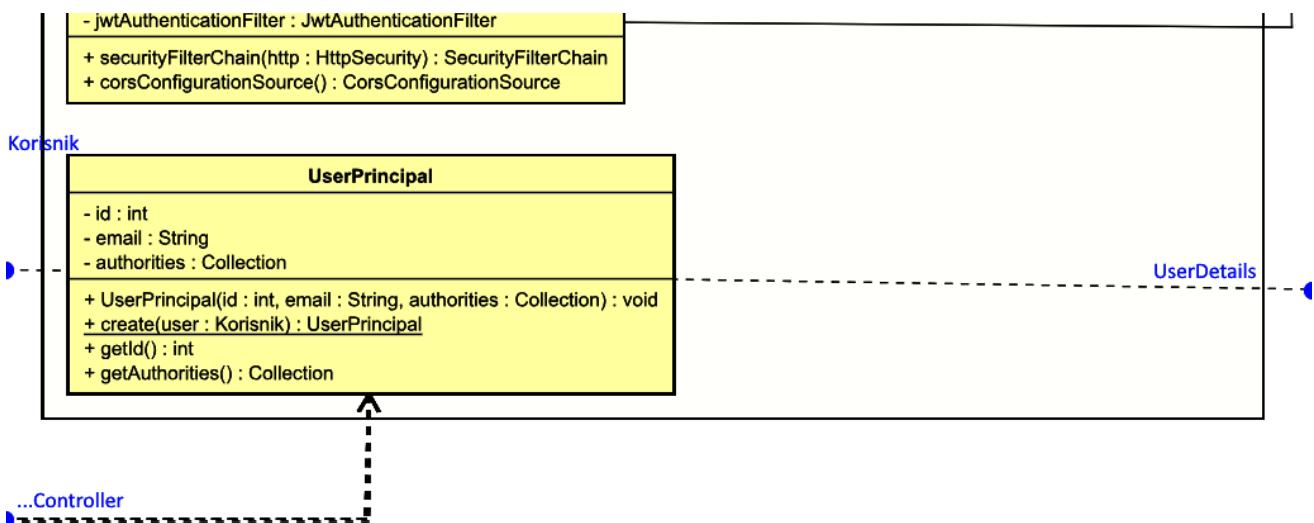


Kategorija Model

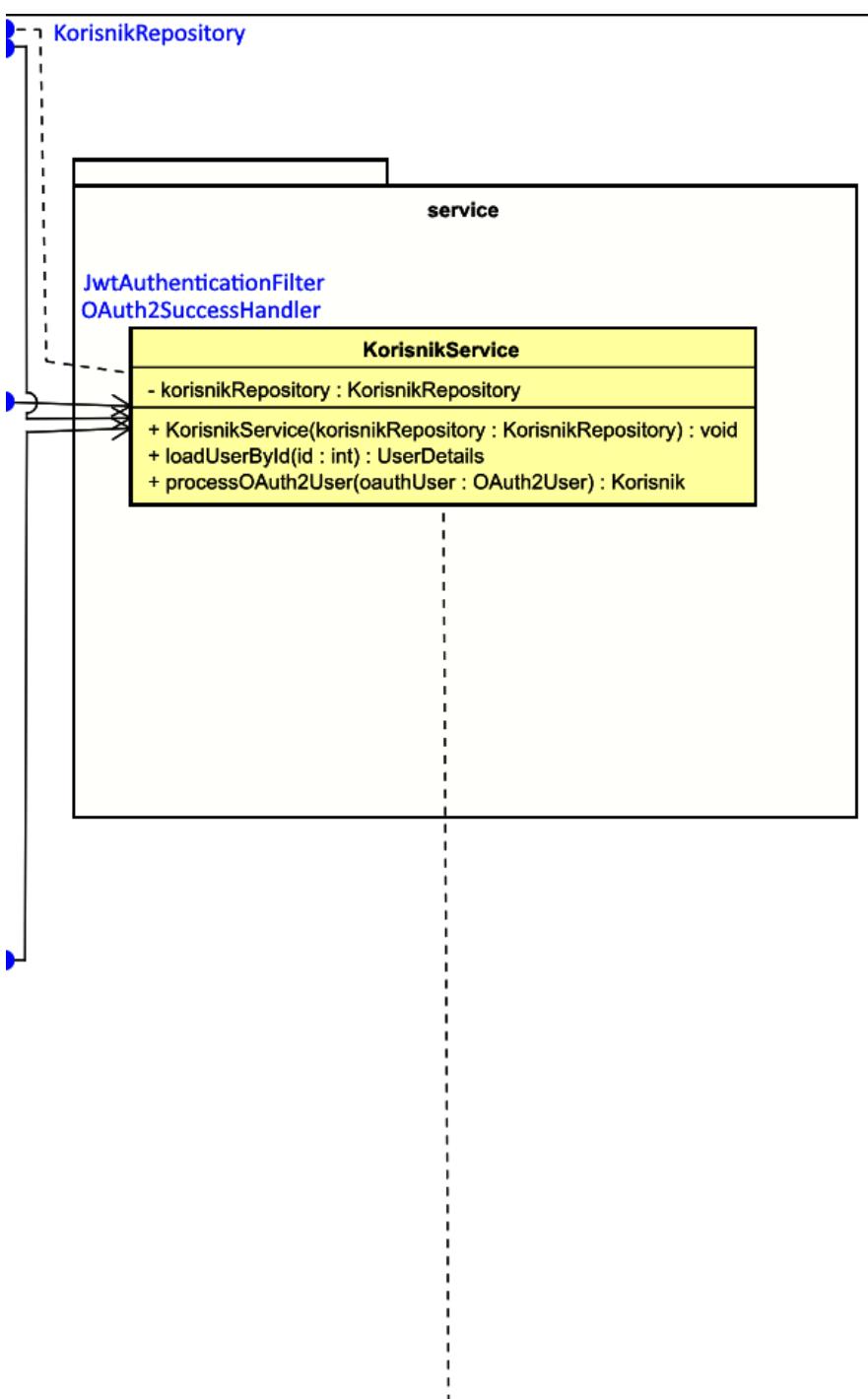


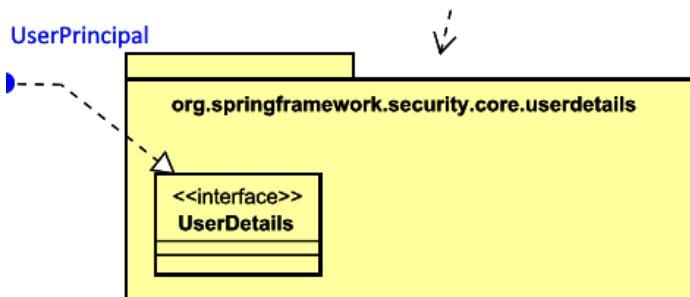
Kategorija Security





Kategorija Service





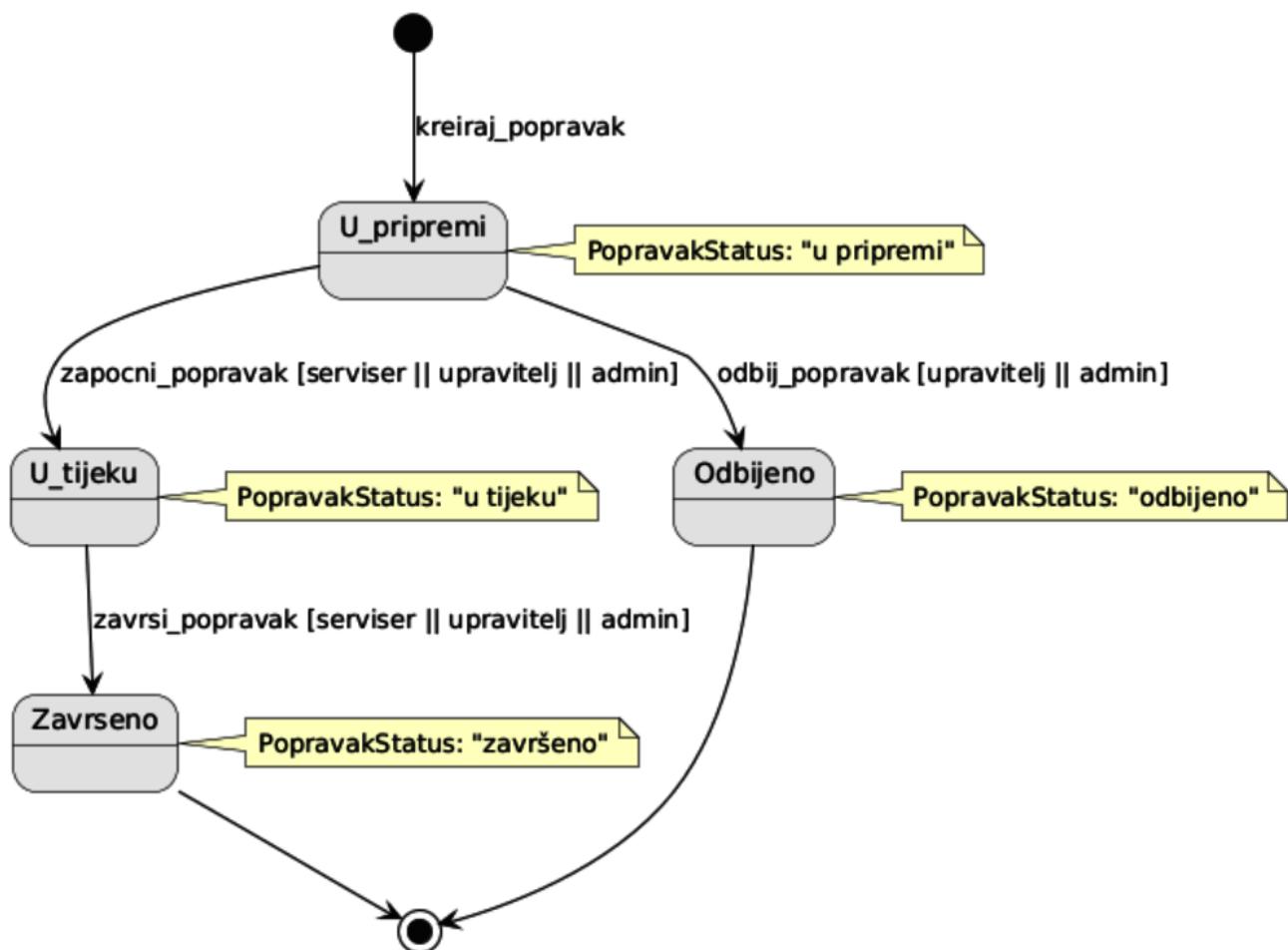
Dinamičko ponašanje aplikacije

Dinamičko ponašanje aplikacije odnosi se na način na koji objekti u sustavu evoluiraju kroz vrijeme, uključujući prijelaze između različitih stanja. To uključuje aktivnosti, događaje, odluke i interakcije unutar aplikacije. UML dijagrami stanja omogućuju vizualizaciju tih promjena i olakšavaju razumijevanje dinamike sustava.

Razumijevanje promjena stanja neophodno je za pravilno funkcioniranje aplikacije jer pruža uvid u interakcije među objektima, komponentama i korisnicima tijekom rada sustava. Korištenjem UML dijagrama stanja i aktivnosti moguće je vizualizirati prijelaze i stanja objekata, identificirati potencijalne probleme, osigurati točnu implementaciju te poboljšati komunikaciju među članovima tima.

UML dijagrami stanja

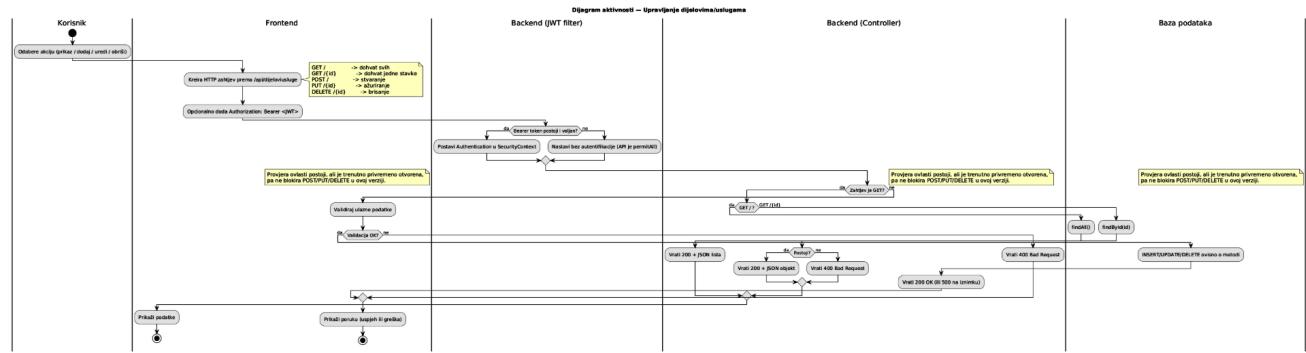
Dijagram stanja — Popravak



Dijagram stanja prikazuje životni ciklus objekta Popravak u sustavu Auto-servis. Popravak se inicijalno nalazi u stanju U pripremi nakon kreiranja, a zatim može prijeći u stanje U tijeku kada serviser započne rad. Alternativno, popravak se može označiti kao Odbijen (npr. od strane upravitelja ili administratora), čime proces završava. Nakon završetka servisiranja, popravak prelazi u stanje Završeno, što predstavlja završno stanje procesa. Prijelazi su dodatno ograničeni

ulogama korisnika (serviser/upravitelj/admin), a bilješke uz stanja pojašnjavaju mapiranje na vrijednosti PopravakStatus u implementaciji.

UML dijagrami aktivnosti



Dijagram aktivnosti prikazuje tijek upravljanja resursom dijelovi/usluge putem REST rute /api/dijeloviusluge iz korisničkog sučelja. Frontend šalje zahtjev (GET/POST/PUT/DELETE), opcionalno s JWT tokenom u Authorization headeru. Na backendu se token (ako postoji) pokušava obraditi kroz JWT filter, ali API pozivi nisu blokirani na razinu Spring Security (permitAll) te se zahtjev nastavlja obrađivati i bez autentifikacije. Controller zatim provodi dohvati ili izmjene nad bazom te vraća odgovor: uspjeh (200) ili grešku (npr. 400 za nepostojeći ID/validaciju, 500 na iznimku). Time je prikazan kompletan dinamički tok od UI akcije do baze i povratnog odgovora.

▼ Pages 14 ✎

Find a page or section...

- ▶ [Home](#)
- ▶ [1. Opis projektnog zadatka](#)
- ▶ [2. Analiza zahtjeva](#)
- ▶ [3. Specifikacija zahtjeva sustava](#)
- ▼ [4. Arhitektura i dizajn sustava](#)
 - Arhitektura sustava
 - Opis arhitekture
 - Stil arhitekture
 - Glavne komponente sustava
 - Mrežni protokoli
 - Globalni upravljački tok
 - Obrazloženje odabira arhitekture
 - Organizacija sustava na visokoj razini
 - Baza podataka
 - Opis tablica
 - Tablica: korisnik
 - Tablica: vozilo
 - Tablica: vrstavozila
 - Tablica: usluga
 - Tablica: dijelovi
 - Dijagram baze podataka

[Dijagram razreda](#)[Kategorija Repository](#)[Kategorija Controller](#)[Kategorija Model](#)[Kategorija Security](#)[Kategorija Service](#)[Dinamičko ponašanje aplikacije](#)[UML dijagrami stanja](#)[UML dijagrami aktivnosti](#)[▶ 5. Arhitektura komponenata i razmještaja](#)[▶ 6. Ispitivanje programskog rješenja](#)[▶ 7. Tehnologije za implementaciju aplikacije](#)[▶ 8. Upute za puštanje u pogon](#)[▶ 9. Zaključak i budući rad](#)[▶ A. Popis literature](#)[▶ A. Dnevnik promjena dokumentacije](#)[▶ B. Prikaz aktivnosti grupe](#)[▶ D. Stranica za sastanke](#)[+ Add a custom sidebar](#)**Clone this wiki locally**<https://github.com/dgrabant/Auto-servis.wiki.git>



5. Arhitektura komponenata i razmještaja

[Edit](#) [New page](#)

[Jump to bottom](#)

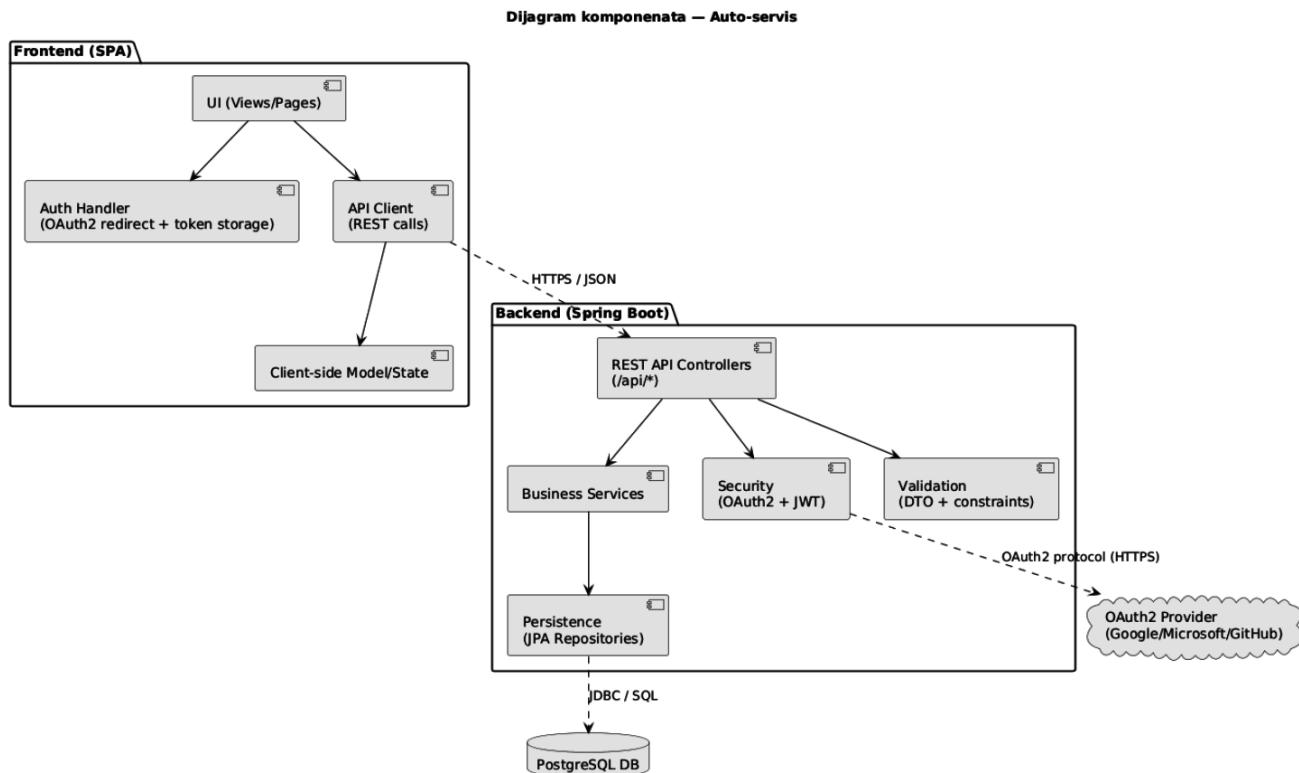
David edited this page 4 hours ago · [6 revisions](#)

Arhitektura sustava predstavlja temeljni okvir za razumijevanje i implementaciju svih njegovih funkcionalnosti. U kontekstu razvojne dokumentacije aplikacija, dijagrami komponenata i razmještaja odlučujući su za prikaz povezanosti i rasporeda različitih komponenata sustava. Ovi dijagrami omogućuju sudionicima projekta razumijevanje i vizualizaciju fizičkog i logičkog dizajna sustava, uključujući interakcije između dijelova aplikacije, što je odlučujuće za efikasnu implementaciju i dugoročnu održivost sustava.

Arhitektura sustava, u kontekstu dijagrama komponenata i razmještaja, pruža uvid u strukturu i raspored ključnih dijelova aplikacije. Ovi dijagrami nisu korisni samo tijekom faza oblikovanja i implementacije, već služe i kao alati za održavanje i optimizaciju sustava u budućnosti.

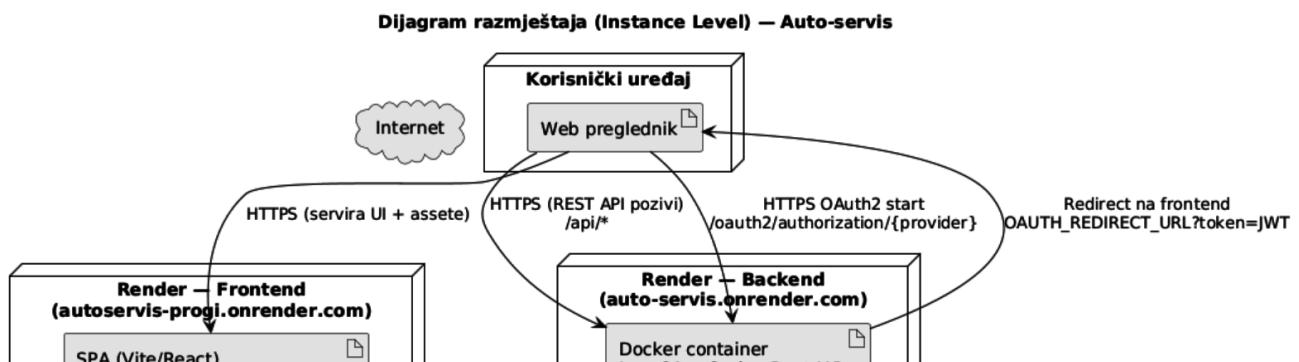
Kao dio razvoja aplikacije, važno je osmisliti i dokumentirati arhitekturu sustava s naglaskom na dijagrame komponenata i razmještaja. Vaš zadatak je izraditi **dijagram komponenata** koji će jasno prikazivati ključne funkcionalne komponente aplikacije, njihovu međusobnu povezanost te sučelja za komunikaciju. Također, trebate izraditi **dijagram razmještaja** komponenata, koji treba detaljno prikazivati kako su te komponente raspoređene u infrastrukturi sustava, uključujući fizičke i virtualne resurse poput poslužitelja ili uređaja krajnjih korisnika.

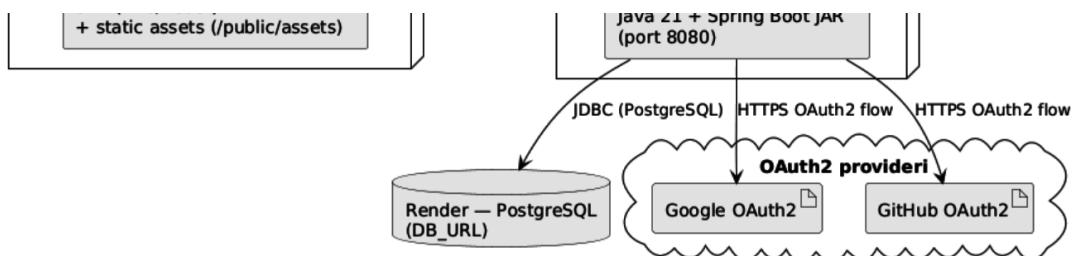
Dijagram komponenata



Dijagram komponenata prikazuje glavne dijelove sustava Auto-servis i njihovu komunikaciju. Frontend (SPA) sadrži UI, obradu autentifikacije i API klijent koji preko HTTPS/JSON poziva komunicira s backendom. Backend (Spring Boot) je podijeljen na REST kontrolere, servisni sloj, sigurnost (OAuth2/JWT), validaciju i perzistenciju (repozitorije) koji pristupaju PostgreSQL bazi. OAuth2 provideri su vanjska komponenta za prijavu korisnika.

Dijagram razmještaja





Sustav je razmješten kao web aplikacija s odvojenim frontend i backend dijelom u oblaku. Frontend je SPA aplikacija (Vite/React) raspoređena na Renderu i dostupna korisnicima putem HTTPS-a, uključujući statičke resurse (slike i ostali asseti). Backend je Spring Boot aplikacija pokrenuta unutar Docker kontejnera (Java 21) također na Renderu te izlaže REST API rute. Backend komunicira s PostgreSQL bazom podataka putem JDBC veze koristeći konfiguraciju kroz varijable okruženja (DB_URL, DB_USERNAME, DB_PASSWORD). Autentifikacija korisnika provodi se preko vanjskih OAuth2 providera (Google i GitHub). Korisnik inicira OAuth2 prijavu iz preglednika, backend odraduje OAuth2 tok i nakon uspjeha preusmjerava korisnika natrag na frontend URL (konfiguiran OAUTH_REDIRECT_URL) uz proslijeđeni JWT token.

UML dijagram razmještaja prikazuje fizičku ili virtualnu raspodjelu komponenata sustava unutar infrastrukture. Cilj je prikazati kako su komponente raspoređene (npr. na poslužiteljima, u okruženjima oblaka ili na uređajima krajnjih korisnika) te način komunikacije, API-ja ili drugih komunikacijskih protokola.

▼ Pages 14 ✎

Find a page or section...

- ▶ [Home](#)
- ▶ [1. Opis projektnog zadatka](#)
- ▶ [2. Analiza zahtjeva](#)
- ▶ [3. Specifikacija zahtjeva sustava](#)
- ▶ [4. Arhitektura i dizajn sustava](#)
- ▶ [5. Arhitektura komponenata i razmještaja](#)
 - Dijagram komponenata
 - Dijagram razmještaja

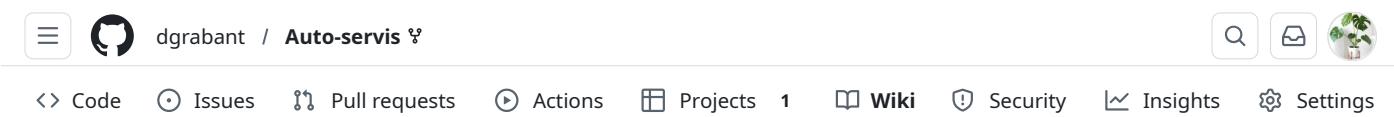
- ▶ [6. Ispitivanje programskog rješenja](#)
- ▶ [7. Tehnologije za implementaciju aplikacije](#)
- ▶ [8. Upute za puštanje u pogon](#)
- ▶ [9. Zaključak i budući rad](#)
- ▶ [A. Popis literature](#)
- ▶ [A. Dnevnik promjena dokumentacije](#)
- ▶ [B. Prikaz aktivnosti grupe](#)
- ▶ [D. Stranica za sastanke](#)

+ Add a custom sidebar

Clone this wiki locally

<https://github.com/dgrabant/Auto-servis.wiki.git>





A screenshot of a GitHub repository page. At the top, there's a header with the repository name "dgrabant / Auto-servis" and a "Wiki" section. Below the header are several navigation links: Code, Issues, Pull requests, Actions, Projects (with a count of 1), Wiki, Security, Insights, and Settings. There are also icons for search, download, and profile.

6. Ispitivanje programskog rješenja

[Edit](#) [New page](#)

[Jump to bottom](#)

David edited this page 4 hours ago · [2 revisions](#)

napomena: članovi su aplikaciju testirali paralelno uz njezin razvoj. Uz to, određene komponente su testirane više puta, i to od više članova. Ispitivanje programskog rješenja nije formalno dokumentirano. U nastavku je opis kako bi testiranje trebalo izgledati te kako bi trebalo biti dokumentirano

Ovo poglavlje treba opisati provedena ispitivanja implementiranih funkcionalnosti na razini komponenti i sustava. Fokus je na odabiru i izvedbi ispitnih slučajeva koji obuhvaćaju redovne, rubne uvjete i testiranje grešaka, kao i upotrebu odgovarajućih alata za provedbu testiranja.

Ispitivanje komponenti

Cilj ispitivanja komponenti je provjera osnovnih funkcionalnosti implementiranih u razredima sustava. Ovdje je potrebno izolirati svaku komponentu kako bi se testirala njezina ispravnost i reakcija na različite scenarije.

Zadaci:

1. Razviti minimalno **6 ispitnih slučajeva** koji obuhvaćaju:

- **Redovne slučajeve:** testiranje uobičajenog ponašanja funkcionalnosti.
- **Rubne uvjete:** provjera ulaznih podataka na granici valjanosti.
- **Izazivanje pogreške (exception throwing):** testiranje reakcije na iznimke.
- **Nepostojeće funkcionalnosti:** provjera reakcije na poziv neimplementirane funkcionalnosti.

Struktura ispitivanja:

Za svaki ispitni slučaj potrebno je:

1. Opisite funkcionalnost koju testirate (npr. dodavanje korisnika, validacija podataka).

2. Navedite ispitni slučaj:

- Ulazne podatke.
- Očekivane rezultate.
- Dobivene rezultate (prolaz/pad ispitivanja).

3. Opisite postupak prvođenja ispitivanja

4. U Gitu moraju biti dostupni izvorni kodovi ispitnih slučajeva.

**Ispitivanje sustava **

Cilj ispitivanja sustava je testiranje ponašanja cijelog sustava u uvjetima stvarnog korištenja, uz posebnu pažnju na međusobnu povezanost svih komponenti. Ispitivanje treba obuhvatiti sve aspekte sustava i njegovu interakciju s korisnicima.

Zadaci:

Razviti minimalno **4 ispitna slučaja** koji obuhvaćaju:

- **Redovne slučajeve:** očekivano ponašanje sustava.
- **Rubne uvjete:** reakcija sustava na granične ulaze.
- **Poziv nepostojećih funkcionalnosti:** testiranje kako sustav reagira na neimplementirane ili neispravne funkcije.

Struktura ispitnih slučajeva za Selenium:

1. **Ulazi:**

- Konkretni podaci koji se unose u sustav (npr. korisničko ime i lozinka za prijavu).

- Simulacija korisničkih akcija (klikanje, unos teksta, navigacija).

2. Koraci ispitivanja:

- Npr. Detaljan opis koraka koje Selenium izvršava:
 - a. Otvoriti aplikaciju u pregledniku.
 - b. Unijeti podatke u formu.
 - c. Kliknuti na gumb za potvrdu.
 - d. Verificirati očekivane rezultate (npr. prijava uspješna).

3. Očekivani izlaz:

- Očekivani rezultat ispitivanja (npr. korisnik preusmjeren na početnu stranicu nakon prijave).

4. Dobiveni izlaz:

- Priložiti logove, screenshotove ili izvještaje s generiranim rezultatima (npr. iz JUnit-a ili Selenija).

Alati za ispitivanje sustava:

- **Selenium IDE** ili prikladan obzirom na vaše razvojno okruženje:

- Jednostavan alat za snimanje korisničkih akcija u pregledniku i automatsko ponavljanje testova.
- Preporučuje se za osnovne ispitne slučajeve.

- **Selenium WebDriver**:

- Omogućuje pisanje naprednih testova u različitim programskim jezicima (Java, Python, C#).
- Preporučuje se za složenije testove, koji zahtijevaju detaljno prilagodbu i automatizaciju.

Primjeri ispitivanja Seleniumom:

1. Formular za prijavu:

u dokumentaciji obavezna su specifičnija ispitivanja vaše aplikacije! - **Ulaz:** Korisničko ime = "user@fer.ugnz.hr", Lozinka = "password123".

- ****Koraci**:**

1. Otvoriti aplikaciju.
2. Unijeti korisničko ime i lozinku.
3. Kliknuti na "Prijava".
4. Provjeriti je li korisnik preusmjeren na početnu stranicu.



- ****Očekivani izlaz**:** "Prijava uspješna."

2. Rubni uvjet - nevažeća lozinka:

- **Ulaz:** Korisničko ime = "user@example.com", Lozinka = "malimedo".

- **Koraci:**

- a. Unijeti podatke u formu za prijavu.
- b. Kliknuti na "Prijava".
- c. Provjeriti je li prikazana poruka o grešci.

- **Očekivani izlaz:** "Pogrešno korisničko ime ili lozinka."

Prezentacija rezultata

Za oba tipa ispitivanja (komponenti i sustava) potrebno je:

- Jasno dokumentirati ulaze, korake, očekivane i dobivene rezultate.
- Priložiti slike ekrana, logove ili izvješća generirana alatima (npr. JUnit ili Selenium).
- Detaljno opisati ponašanje sustava, posebno u rubnim uvjetima.
- Navesti broj otkrivenih grešaka!

▼ Pages 14 

Find a page or section...

- ▶ [Home](#)
- ▶ [1. Opis projektnog zadatka](#)
- ▶ [2. Analiza zahtjeva](#)
- ▶ [3. Specifikacija zahtjeva sustava](#)
- ▶ [4. Arhitektura i dizajn sustava](#)
- ▶ [5. Arhitektura komponenata i razmještaja](#)
- ▼ [6. Ispitivanje programskog rješenja](#)
 - Ispitivanje komponenti
 - **Ispitivanje sustava **
- ▶ [7. Tehnologije za implementaciju aplikacije](#)
- ▶ [8. Upute za puštanje u pogon](#)
- ▶ [9. Zaključak i budući rad](#)
- ▶ [A. Popis literature](#)
- ▶ [A. Dnevnik promjena dokumentacije](#)
- ▶ [B. Prikaz aktivnosti grupe](#)
- ▶ [D. Stranica za sastanke](#)

+ Add a custom sidebar

Clone this wiki locally

<https://github.com/dgrabant/Auto-servis.wiki.git> 



The GitHub repository header for the 'Auto-servis' project by user 'dgrabant'. It shows the repository name, a fork icon, and a star icon. Below the repository name are links for 'Code', 'Issues', 'Pull requests', 'Actions', 'Projects' (with 1 item), 'Wiki', 'Security', 'Insights', and 'Settings'. There are also icons for search, download, and profile.

7. Tehnologije za implementaciju aplikacije

[Edit](#) [New page](#)

[Jump to bottom](#)

David edited this page 3 hours ago · [3 revisions](#)

Korištene tehnologije i alati

Programski jezici

- **Java 8** – koristi se za razvoj backend aplikacije i implementaciju poslovne logike te komunikaciju s bazom podataka.
- **Node.js 18** – koristi se kao runtime okruženje za razvoj i pokretanje frontend aplikacije.

Radni okviri i biblioteke

- **Three.js** – JavaScript biblioteka za prikaz 3D grafike u web pregledniku. Koristi se za vizualizaciju 3D modela vozila u korisničkom sučelju.
- **Node.js 18** – koristi se i za upravljanje frontend ovisnostima i razvojnim alatima (npm).

Baza podataka

- **PostgreSQL 18** – relacijska baza podataka za pohranu podataka o korisnicima, vozilima, terminima i servisnim zapisima. Backend pristupa bazi putem standardne JDBC veze.

Razvojni alati

- **Visual Studio Code** – razvoj frontend dijela aplikacije.
- **IntelliJ IDEA 2024.1.2** – razvoj backend dijela aplikacije u Javi.

Alati za razmještaj

- **Docker** – aplikacija se pokreće unutar kontejnera radi jednostavnijeg razmještaja i konzistentnog okruženja.

Cloud platforma

- **Render.com** – koristi se za hosting aplikacije i omogućuje javnu dostupnost sustava putem interneta.

Ova kombinacija tehnologija omogućuje razvoj modularne, stabilne i lako održive web aplikacije.

Primjer:

Za razvoj klijentskog dijela aplikacije korišten je **React** (verzija 18) [ref.], popularna JavaScript biblioteka za izgradnju interaktivnih korisničkih sučelja. React omogućuje stvaranje samostalnih komponenti koje se mogu ponovno koristiti i lako ažurirati, čime se poboljšava učinkovitost razvoja. Za stiliziranje su upotrijebljene **styled-components** (verzija 5.3), što omogućava integraciju stilova izravno unutar Ručat komponenti koristeći **CSS-in-JS** pristup.

napomena: dio ovog teksta proširen odnosno formatiran pomoću alata ChatGPT.

Pages 14

Find a page or section...

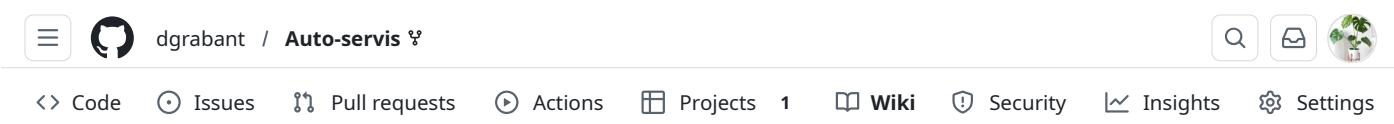
- ▶ [Home](#)
- ▶ [1. Opis projektnog zadatka](#)
- ▶ [2. Analiza zahtjeva](#)
- ▶ [3. Specifikacija zahtjeva sustava](#)
- ▶ [4. Arhitektura i dizajn sustava](#)
- ▶ [5. Arhitektura komponenata i razmještaja](#)
- ▶ [6. Ispitivanje programskog rješenja](#)
- ▼ [7. Tehnologije za implementaciju aplikacije](#)
 - Korištene tehnologije i alati
 - Programski jezici
 - Radni okviri i biblioteke
 - Baza podataka
 - Razvojni alati
 - Alati za razmještaj
 - Cloud platforma
 - ▶ [8. Upute za puštanje u pogon](#)
 - ▶ [9. Zaključak i budući rad](#)
- ▶ [A. Popis literature](#)
- ▶ [A. Dnevnik promjena dokumentacije](#)
- ▶ [B. Prikaz aktivnosti grupe](#)
- ▶ [D. Stranica za sastanke](#)

+ Add a custom sidebar

Clone this wiki locally

<https://github.com/dgrabant/Auto-servis.wiki.git>





8. Upute za puštanje u pogon

[Edit](#) [New page](#)

[Jump to bottom](#)

Antonio Šantak edited this page 10 hours ago · [3 revisions](#)

1. Instalacija

- **Preduvjeti:** Node.js 18, Spring Boot, registracija na servis Mailjet (email klijent), registracija na servis ImgBB (prijenos slika)
- **Preuzimanje:** kloniranje GitHub repozitorija

Primjer:

```
git clone https://github.com/dgrabant/Auto-servis.git
```

Instalacija ovisnosti:

```
cd Auto-servis/source/frontend  
npm install
```

2. Postavke

Detaljne upute za konfiguraciju aplikacije:

- **Konfiguracijske datoteke:** .env -> Auto-servis/source/backend

- Primjer .env datoteke:

```
OAUTH_REDIRECT_URL=http://localhost:5173/oauth-redirect
```

```
OAUTH_LOGOUT_URL=http://localhost:5173/logout
```

```
JWT_SECRET_KEY=slozeni_tajni_string
```

```
OAUTH_GOOGLE_CLIENT_ID=google_client_id
```

```
OAUTH_GOOGLE_CLIENT_SECRET=google_secret
```

```
OAUTH_GITHUB_CLIENT_ID=github_client_id
```

```
OAUTH_GITHUB_CLIENT_SECRET=github_secret
```

```
DB_URL=database_url
```

```
DB_USERNAME=database_username
```

```
DB_PASSWORD=database_password
```

```
IMGBB_API_KEY=imgbb_api_key
```

```
MAILJET_API_KEY=mailjet_key
```

```
MAILJET_API_SECRET=mailjet_secret
```

```
MAILJET_API_SENDER=sender_email_address
```

- **Postavke baze podataka:** Postgres DB, model prema dijagramu baze podataka, baza može biti smještena bilo gdje (ako je lokalno, potrebno ju je pokrenuti), vjerodajnice se spremaju u .env datoteku

3. Pokretanje aplikacije

Upute za pokretanje aplikacije u različitim okruženjima:

- **Razvojno okruženje:**

```
npm run dev
```

- **Producčijsko okruženje:**

- Prevođenje aplikacije:

```
npm run build
```

- Pokretanje poslužitelja:

```
./gradlew clean bootJar --no-daemon java -jar /app/build/libs/app.jar
```

- **Provjera rada:** URL (npr. <http://localhost:8080>) .

4. Upute za administratore

Smjernice za administratore aplikacije nakon puštanja u pogon:

- **Pristup administratorskom sučelju:**

- URL za admin panel (npr. /admin.html).

- **Redovito održavanje:**

- Arhiviranje baze podataka (pgAdmin).

- Pregled logova na poslužitelju (npr. Render.com).

- Ažuriranje aplikacije (primjer: povlačenje novih verzija iz Git repozitorija i ponovno pokretanje aplikacije se aktivira nakon commit-a).

5. Primjer za Render platformu (Cloud Deploy)

Render je popularna cloud platforma za jednostavno pokretanje aplikacija.

- **Priprema repozitorija:**

- Osigurajte da vaš projekt ima datoteku Dockerfile za konfiguraciju deploya.

- Primjer Dockerfile-a:

```
FROM gradle:8.5-jdk21 AS builder
WORKDIR /app
COPY ..
RUN chmod +x gradlew
RUN ./gradlew clean bootJar --no-daemon
FROM eclipse-temurin:21-jdk-alpine
WORKDIR /app
COPY --from=builder /app/build/libs/*.jar app.jar
ENTRYPOINT ["java","-jar","/app/app.jar"]
```

- **Postavljanje na Render:**

- Prijavite se na [Render](#).

- Kreirajte novi **Web Service** i povežite ga s vašim GitHub repozitorijem.

- Konfigurirajte postavke (npr. build i start naredbe).

- Dodajte environment varijable (npr. DATABASE_URL, API_KEY).

- **Pokretanje aplikacije:**

Render će automatski preuzeti repozitorij, instalirati ovisnosti i pokrenuti aplikaciju. Nakon deploya, aplikaciji možete pristupiti putem generiranog URL-a (npr. <https://autoservis-progi.onrender.com>).

Opis pristupa aplikaciji na javnom poslužitelju

Javni poslužitelj omogućava besplatno pružanje usluga, ali to nosi i svoje posljedice: web-servis se nakon neaktivnosti gasi i pri prvom pokretanju je potrebno pričekati dok se servis ponovno pokrene. Pokretanje je sporije (besplatna inačica) što može dovesti do 10 minuta pokretanja.

Pages 14

Find a page or section...

- ▶ [Home](#)
- ▶ [1. Opis projektnog zadatka](#)
- ▶ [2. Analiza zahtjeva](#)
- ▶ [3. Specifikacija zahtjeva sustava](#)
- ▶ [4. Arhitektura i dizajn sustava](#)
- ▶ [5. Arhitektura komponenata i razmještaja](#)
- ▶ [6. Ispitivanje programskog rješenja](#)
- ▶ [7. Tehnologije za implementaciju aplikacije](#)
- ▼ [8. Upute za puštanje u pogon](#)
 - Opis pristupa aplikaciji na javnom poslužitelju
- ▶ [9. Zaključak i budući rad](#)
- ▶ [A. Popis literature](#)
- ▶ [A. Dnevnik promjena dokumentacije](#)
- ▶ [B. Prikaz aktivnosti grupe](#)
- ▶ [D. Stranica za sastanke](#)

+ Add a custom sidebar

Clone this wiki locally

<https://github.com/dgrabant/Auto-servis.wiki.git>





Code Issues Pull requests Actions Projects 1 Wiki Security Insights Settings

9. Zaključak i budući rad

[Edit](#)[New page](#)[Jump to bottom](#)

David edited this page 49 minutes ago · [2 revisions](#)

Projekt je završen u skladu sa vremenskim ograničenjem. Sve su funkcionalnosti ostvarene, jedino u projektu fali formalno opisano i izvršeno testiranje. Nažalost, naš tim nije to uspio napraviti. Razlozi su: vremensko ograničenje, nepredvidljivost razvoja u timskom okruženju kao i individualni razlozi te individualne razlike. Također, bitno je naglasiti i radni kapacitet - koliko ljudi sudjeluje u razvoju projekta (6 ili 7, nekada jedna osoba može puno značiti) te kakvo je njihovo predznanje. Ipak, smatram da je svatko od članova naučio nešto više, razvio je pregled te primjenu programsko-inženjerskih principa te metoda. Za sam projekt, članovi tima radili su njima znanjem i sposobnostima prikladnim zadacima, ali i razvijali sposobnosti u širem okružju njihovih vlastitih interesa. Tokom razvoja, pokušao se slijediti princip optimizacije.

Što se tiče daljnog razvoja aplikacije, mogla bi se implementirati i direktna komunikacija između pojedinih korisnika same aplikacije, dodavati više informacija o servisu, unaprijediti dizajn te pristupačnost aplikacije, razmisliti koje bi još značajke u aplikacije dobro doše te odraditi jedan tkzv. "brainstorm" unutar tima kako bi pronašli najbolje i najkvalitetnije ideje.

Harlemona kočija 

▼ Pages 14

Find a page or section... 

- ▶ [Home](#)
- ▶ [1. Opis projektnog zadatka](#)
- ▶ [2. Analiza zahtjeva](#)
- ▶ [3. Specifikacija zahtjeva sustava](#)
- ▶ [4. Arhitektura i dizajn sustava](#)
- ▶ [5. Arhitektura komponenata i razmještaja](#)
- ▶ [6. Ispitivanje programskog rješenja](#)

▶ [7. Tehnologije za implementaciju aplikacije](#)

▶ [8. Upute za puštanje u pogon](#)

[9. Zaključak i budući rad](#)

▶ [A. Popis literature](#)

▶ [A. Dnevnik promjena dokumentacije](#)

▶ [B. Prikaz aktivnosti grupe](#)

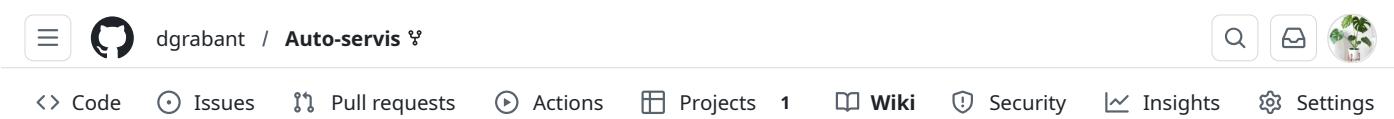
▶ [D. Stranica za sastanke](#)

+ Add a custom sidebar

Clone this wiki locally

<https://github.com/dgrabant/Auto-servis.wiki.git>





A. Popis literature

[Edit](#)[New page](#)[Jump to bottom](#)

David edited this page 5 days ago · [2 revisions](#)

Reference i literatura koja je pomogla pri ostvarivanju projekta:

1. Programsko inženjerstvo, FER ZEMRIS, <http://www.fer.hr/predmet/proinz>
2. I. Sommerville, "Software engineering", 8th ed, Addison Wesley, 2007.
3. T.C.Lethbridge, R.Langaniere, "Object-Oriented Software Engineering", 2nd ed. McGraw-Hill, 2005.
4. I. Marsic, Software engineering book", Department of Electrical and Computer Engineering, Rutgers University, <http://www.ece.rutgers.edu/~marsic/> books/SE
5. The Unified Modeling Language, <https://www.uml-diagrams.org/>
6. Astah Community, <http://astah.net/editions/uml-new>
7. Sun Tzu, Umijeće ratovanja (dvojezično izdanje), Planetopija, 2023.

Harlemova kočija 

▼ Pages 14

Find a page or section...

- ▶ [Home](#)
- ▶ [1. Opis projektnog zadatka](#)
- ▶ [2. Analiza zahtjeva](#)
- ▶ [3. Specifikacija zahtjeva sustava](#)
- ▶ [4. Arhitektura i dizajn sustava](#)
- ▶ [5. Arhitektura komponenata i razmještaja](#)
- ▶ [6. Ispitivanje programskog rješenja](#)

▶ [7. Tehnologije za implementaciju aplikacije](#)

▶ [8. Upute za puštanje u pogon](#)

▶ [9. Zaključak i budući rad](#)

A. Popis literature

▶ [A. Dnevnik promjena dokumentacije](#)

▶ [B. Prikaz aktivnosti grupe](#)

▶ [D. Stranica za sastanke](#)

+ Add a custom sidebar

Clone this wiki locally

<https://github.com/dgrabant/Auto-servis.wiki.git>



A. Dnevnik promjena dokumentacije

[Edit](#)
[New page](#)
[Jump to bottom](#)

David edited this page 48 minutes ago · [30 revisions](#)

Rev.	Opis promjene/dodatka	Autori	Datum
0.0	Inicijalizacija wiki	David Grabant	8.10.2025.
0.1	Napravljen predložak uz sitne česte brojne dopune	David Grabant	10-20.10.2025.
0.2	Dodana Anazila zahtjeva te Specifikacije sustava	David Grabant	20.10.2025.
0.3	Dodane slike za UC i sekvencijske diagrame	Mihael Car	21.10.2025.
0.4	Ažurirani dionici u poglavlju Analiza zahtjeva	Antonio Šantak	21.10.2025.
0.5	1. opis projekta	Luka Ivelić	23.10.2025.
0.6	Dodavanje i dokumentiranje izmjena	David Grabant	26.10.2025.
0.7	Ažurirani dionici u poglavlju Analiza zahtjeva	David Grabant	26.10.2025.
0.8	Obrisana str. Tijek projekta (bila u uporabi umjesto B. str.)	David Grabant	26.10.2025.
0.9	Dodan popis sastanaka	Luka Žlender	27.10.2025.
0.10	Ažurirana stranica Arhitekture sustava	David Grabant	11.11.2025.
0.11	Ažuriran popis sastanaka	Luka Žlender	11.11.2025.
0.12	Ažuriran, povećan i dopunjjen opis projekta	David Grabant	11.11.2025.
0.13	Dopunjjen README file	David Grabant	11.11.2025.
0.14	Ažurirane aktivnosti projekta	David Grabant	11.11.2025.
0.15	Ažurirana stranica Arhitekture sustava	Antonio Šantak	13.11.2025.
0.16	Dodana tablica aktivnosti	David Grabant	10.12.2025.
0.17	Ažuriran dijagram razreda	Antonio Šantak	14.12.2025.
0.18	Dodan plan rada (Gantt)	Luka Žlender	15.12.2025.
0.19	Dodani dijagrami stanja, aktivnosti i razmještaja	Luka Žlender	12.01.2025.
0.20	Ažuriran popis sastanaka	Luka Žlender	12.01.2025.
0.21	Uređivanje nekih wiki stranica	David Grabant	16.01.2025.
0.22	Detaljnije uređivanje	David Grabant	18.01.2025.
0.23	Dodan dijagram komponenata	LŽ	20.01.2025.
0.22	Krajnje uređivanje dokumentacije	David Grabant	23.01.2025.
N/A	N/A	N/A	N/A

▼ Pages 14



Find a page or section...

▶ Home

▶ 1. Opis projektnog zadatka

▶ 2. Analiza zahtjeva

▶ 3. Specifikacija zahtjeva sustava

▶ 4. Arhitektura i dizajn sustava

▶ 5. Arhitektura komponenata i razmještaja

▶ 6. Ispitivanje programskog rješenja

▶ 7. Tehnologije za implementaciju aplikacije

▶ 8. Upute za puštanje u pogon

▶ 9. Zaključak i budući rad

▶ A. Popis literature

A. Dnevnik promjena dokumentacije

▶ B. Prikaz aktivnosti grupe

▶ D. Stranica za sastanke

+ Add a custom sidebar

Clone this wiki locally

<https://github.com/dgrabant/Auto-servis.wiki.git>



Code Issues Pull requests Actions Projects 1 Wiki Security Insights Settings

B. Prikaz aktivnosti grupe

[Edit](#) [New page](#)[Jump to bottom](#)

David edited this page 4 minutes ago · [29 revisions](#)

#Dnevnik sastajanja

Tim: Harlemova kočija

Voditelj tima: David Grabant

Evidenciju sastanaka vodi: Luka Žlender

Razdoblje: listopad 2025.

1. sastanak

Datum: 7. listopada 2025.

Prisustvovali: D. Grabant, D. Nell, A. Šantak, M. Car, L. Žlender, L. Ivelić

Teme sastanka:

Stvorena WhatsApp grupa radi brže komunikacije i dogovora o terminima.

Dogovoren prvi susret uživo u sklopu laboratorijskih vježbi sljedeći dan.

2. sastanak

Datum: 8. listopada 2025.

Prisustvovali: D. Grabant, D. Nell, A. Šantak, M. Car, L. Žlender, L. Ivelić

Teme sastanka:

Prvi susret uživo u sklopu laboratorijskih vježbi.

Formiran tim te odabran projektni zadatak "Auto servis".

Rasprava o načinu rada i komunikacijskim kanalima.

Odlučeno koristiti Discord kao glavni kanal komunikacije i koordinacije — M. Car otvorio Discord server.

Dogovoreni osnovni termini i tjedni ritam sastanaka.

Nakon sastanka:

D. Grabant otvorio GitHub repozitorij i Wiki projekta.

3. sastanak

Datum: 14. listopada 2025.

Prisustvovali: D. Grabant, D. Nell, A. Šantak, M. Car, L. Žlender, L. Ivelić

Teme sastanka:

Analiza projektnog zadatka Auto servis i rasprava o funkcionalnostima aplikacije (prijava korisnika, zakazivanje servisa, rezervacija vozila).

Dogovor o pravilima komunikacije i podjeli početnih zaduženja.

Zaduženja:

svi članovi pročitati zadatak,

proučiti razliku između funkcijskih i nefunkcijskih zahtjeva,

istražiti alate za crtanje UML dijagrama.

Dogovoreno: sljedeći sastanak 16. listopada na Discordu.

4. sastanak

Datum: 15. listopada 2025.

Prisustvovali: D. Grabant, D. Nell, A. Šantak, M. Car, L. Ivelić

Teme sastanka:

Rad s asistentom i razrada specifikacije zadatka.

Zaključci koje je poslao D. Grabant nakon vježbi:

Još jednom detaljno pročitati zadatak i promisliti o zahtjevima.

Kad kolega postavi sustav, svaki tim dodaje svoj dio u dokument.

Pregledati sve druge dijelove dokumenta i podcrtatiti/dokumentirati nejasnoće.

Sljedeći sastanak: 19:15 na Discordu.

Proučiti primjere prethodnih projekata i načine pisanja specifikacija, funkcijskih i nefunkcijskih zahtjeva.

Proučiti programe za crtanje grafova i dijagrama.

5. sastanak

Datum: 16. listopada 2025.

Prisustvovali: D. Nell, M. Car, L. Žlender

Teme sastanka:

Proučavanje i komentiranje Use Case dijagrama i osnovnih UML modela.

Rasprava o primjerima iz prethodnih projekata i njihovoj primjeni na zadatak Auto servis.

Zaduženja: pripremiti prve skice Use Case dijagrama i definirati glavne aktere sustava.

Dogovoren: sljedeći sastanak 19. listopada (Discord).

6. sastanak

Datum: 19. listopada 2025.

Prisustvovali: D. Grabant, L. Ivelić, M. Car

Teme sastanka:

Voditelj tima D. Grabant izradio i objavio na GitHub Wiki:

funkcijiske zahtjeve,



Use Case dijagrame,

sekvensijske dijagrame (slike podijeljene na Discordu).

Dogovor o dodavanju UML dijagrama kolega u Wiki.

Zadaci: pregledati objavljene materijale, ukloniti višak zahtjeva, dopuniti gdje je potrebno.

Rasprava o potrebi vođenja detaljne evidencije sastanaka u Wiki dokumentaciji.

Napomena: sastanak nije održan sinkrono; većina rada odvila se samostalno i asinkrono.

7. sastanak

Datum: 21. listopada 2025.

Prisustvovali: D. Grabant, D. Nell, A. Šantak, M. Car, L. Žlender, L. Ivelić

Teme sastanka:

Pregled napretka i priprema za završetak prve faze projekta.

Voditelj pripremio agendu s glavnim točkama:

Provjera i dorada izrađenih dijagrama.

Rasprava o funkcijskim i nefunkcijskim zahtjevima.

Plan zadataka za idući tjedan.

Uređivanje evidencije sastanaka.

Izrada i objava opisa projektnog zadatka.

Zaduženja:

D. Grabant: pročistiti i dovršiti tekst na Wiki-ju, urediti evidenciju sastanaka.

Ostali: dodati dijagrame i grafove u Wiki, započeti pisanje opisa projekta.

Naglašena važnost potvrđivanja prisustva na sastancima i redovitog izvještavanja.

8. sastanak

Datum: 23. listopada 2025.

Prisustvovali: D. Grabant, D. Nell, A. Šantak, M. Car, L. Žlender, L. Ivelić

Teme sastanka:

Koordinacija i planiranje zadataka te dogovor o raspodjeli uloga unutar tima.

Voditelj (D. Grabant) naglasio važnost jasne komunikacije, redovite evidencije i praćenja napretka.

Podjela uloga:

Frontend tim: D. Nell i M. Car

Backend tim: A. Šantak i L. Ivelić

L. Žlender: izrada relacijskog modela baze podataka te vođenje evidencije sastanaka i dnevnika aktivnosti

D. Grabant: vođenje tima, koordinacija i održavanje Wiki dokumentacije

Zaduženja:

A. Šantak i M. Car: definirati plan rada do ponедјeljka i javiti napredak.

M. Car: postaviti admin ovlasti na Discordu i dodati kanale frontend i backend.

L. Ivelić: ažurirati Wiki i popisati izmjene.

D. Grabant: urediti tekstove na Wiki-ju i nadzirati dokumentaciju.

UML dorade:

ukloniti bazu iz dijagrama

maknuti nepotrebne provjere

9. sastanak

Datum: 27. listopada 2025.

Prisustvovali: D. Grabant, D. Nell, A. Šantak, M. Car, L. Žlender, L. Ivelić

Teme sastanka:

Tjedni statusni sastanak održan putem Discorda.

Svaki član iznio što je do sada napravio, što planira napraviti u idućim danima te koje dijelove drugih timova treba da bi mogao nastaviti svoj rad.

Rasprava o međusobnim ovisnostima između frontenda, backenda i baze podataka.

Dogovor da se do idućeg sastanka uskladi redoslijed implementacije funkcionalnosti kako bi svi dijelovi sustava mogli biti testirani zajedno.

Naglašena važnost međusobne komunikacije i redovitog ažuriranja napretka na Discordu i GitHubu.

10. sastanak

Datum: 30. listopada 2025.

Prisustvovali: D. Grabant, A. Šantak, M. Car, L. Žlender

Teme sastanka:

Kratki sastanak uživo uz laboratorijske vježbe.

Rasprava o načinu rada tima i inženjerskom pristupu projektu.

Dogovor o inkrementalnom modelu razvoja te važnosti preuzimanja odgovornosti i funkcionalnosti ispred estetike.

Backend tim radi na API-ju i proučava OAuth verifikaciju.

Frontend tim treba pojednostaviti korisničko sučelje i dodati početnu stranicu s objašnjnjem funkcionalnosti aplikacije.

L. Žlender izrađuje prvu verziju baze podataka.

Dogovoreno da svi članovi dostave popis korištenih tehnologija za potrebe dokumentacije arhitekture.

11. sastanak

Datum: 4. studenoga 2025.

Prisustvovali: D. Grabant, A. Šantak, M. Car, L. Ivelić

Teme sastanka:

Dogovor o dalnjem spajanju backend i frontend dijelova sustava.

Potrebno dovršiti osnovne funkcionalnosti API-ja i pojednostaviti bazu.

Cilj: do kraja tjedna izraditi radnu verziju aplikacije spremnu za zajedničko testiranje i prijavu bugova.

Dogovoreno prikupljanje povratnih informacija nakon testiranja radi planiranja sljedeće iteracije razvoja.

12. sastanak

Datum: 10. studenoga 2025.

Prisustvovali: D. Grabant, D. Nell, A. Šantak, M. Car, L. Žlender, L. Ivelić

Teme sastanka:

Pregled trenutnog stanja projekta i rješavanje bugova.

Dogovor o načinu upravljanja GitHub repozitorijem i organizaciji zadataka.

Raspodjela tema i uloga za prezentaciju na nadolazećim laboratorijskim vježbama.

Prolaz kroz dokumentaciju te dogovor o završnim doradama arhitekture.

Završni sastanak prije međuispita — osvrt na dosadašnji napredak i plan dalnjih koraka.

13. sastanak

...

Datum: 15. prosinca 2025.

Prisustvovali: D. Grabant, L. Žlender

Teme sastanka:

Kratki sastanak vezan uz uređivanje Wiki dokumentacije (dopune i formatiranje postojećih poglavlja).

Kratka rasprava o trenutnom stanju projekta i prioritetima za nastavak rada.

14. sastanak

Datum: 8. siječnja 2026.

Prisustvovali: D. Grabant, D. Nell, A. Šantak, M. Car, L. Žlender, L. Ivelić

Teme sastanka:

Sinkronizacija između backenda i frontenda (uskladijanje API-ja i testiranje integracije).

Završetak dokumentacije na Wiki-ju (pregled preostalih dijelova i raspodjela zadataka).

Zaduženja:

Backend (A. Šantak): dovršiti i uskladiti API prema potrebama integracije.

Frontend (D. Nell, M. Car): spojiti ključne funkcionalnosti s backend-om i prijaviti probleme.

Dokumentacija dijagrama (L. Žlender, L. Ivelić):

Dijagram razreda



Dokumentacija ponašanja: dijagram stanja i dijagram aktivnosti

Dokumentacija arhitekture sustava: komponentni dijagram i dijagram razmještaja

Koordinacija i završne dorade (D. Grabant): ujednačiti i provjeriti da su sve obavezne cjeline na Wiki-ju popunjene.

Plan rada

Ganttov dijagram

Aktivnost / Tjedan	T1	T2	T3	T4	T5	T6	T7	T8	T9
Formiranje tima i podjela uloga	DG								
Postavljanje GitHub repozitorija		DG							
Postavljanje Discord kanala i ovlasti			MC						
Analiza projektnog zadatka		Svi	Svi						
Specifikacija zahtjeva i UC dijagrami			DG	DG					

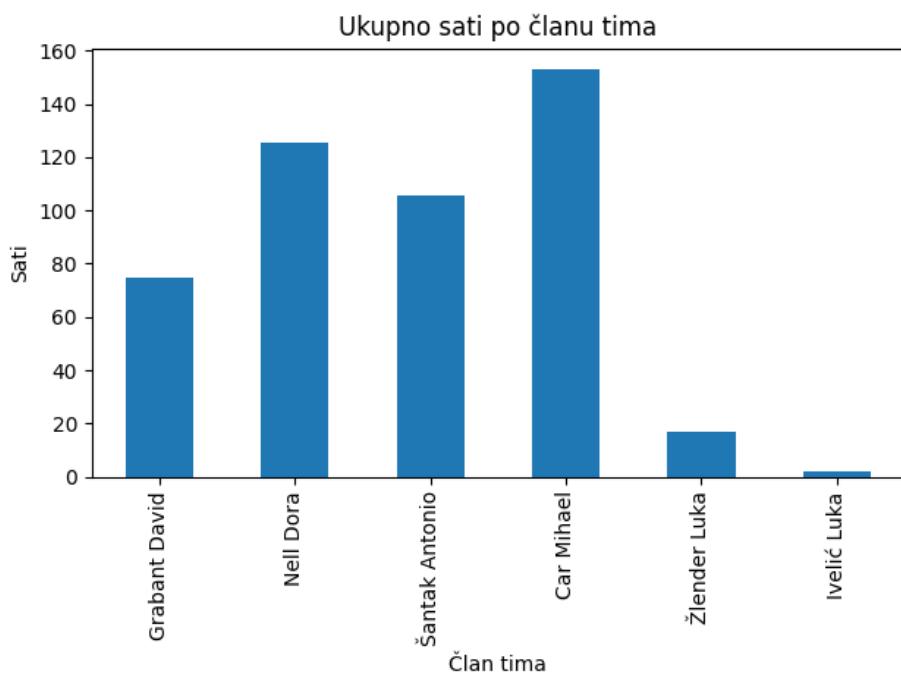
Relacijski model baze podataka			LŽ							
Oblikovanje arhitekture sustava			AŠ,LI	AŠ,LI						
Plan rada projekta			DG,LŽ							
Implementacija backend jezgre			AŠ,LI		AŠ,LI	AŠ,LI				
Implementacija frontend sučelja					DN,MC	DN,MC	DN,MC			
OAuth2 autentifikacija					AŠ,MC	AŠ,MC				
Integracija frontend-backend					AŠ,DN,MC,LI		AŠ,DN,MC,LI			
Ažuriranje Wiki-ja i popisa izmjena	Svi	Svi	Svi	Svi	Svi	Svi	Svi	Svi	Svi	S
Koordinacija tima i uređivanje Wiki-ja	DG	DG	DG	DG	DG	DG	DG	DG	DG	D
Osnovno testiranje i objava prve verzije					Svi					
Implementacija ključnih funkcionalnosti							AŠ,LI,DN,MC		A	
Završetak implementacije i dokumentacije										

Tablica aktivnosti

Aktivnost, u satima	Grabant David	Nell Dora	Šantak Antonio	Car Mihael	Žlender Lukáš	Ivelić Luka
Upravljanje projektom	43					
Opis projektnog zadatka	1					
Funkcionalni zahtjevi	1					
Opis pojedinih obrazaca	0.5					
Dijagram obrazaca	1			2		2
Sekvencijski dijagrami	0.5					

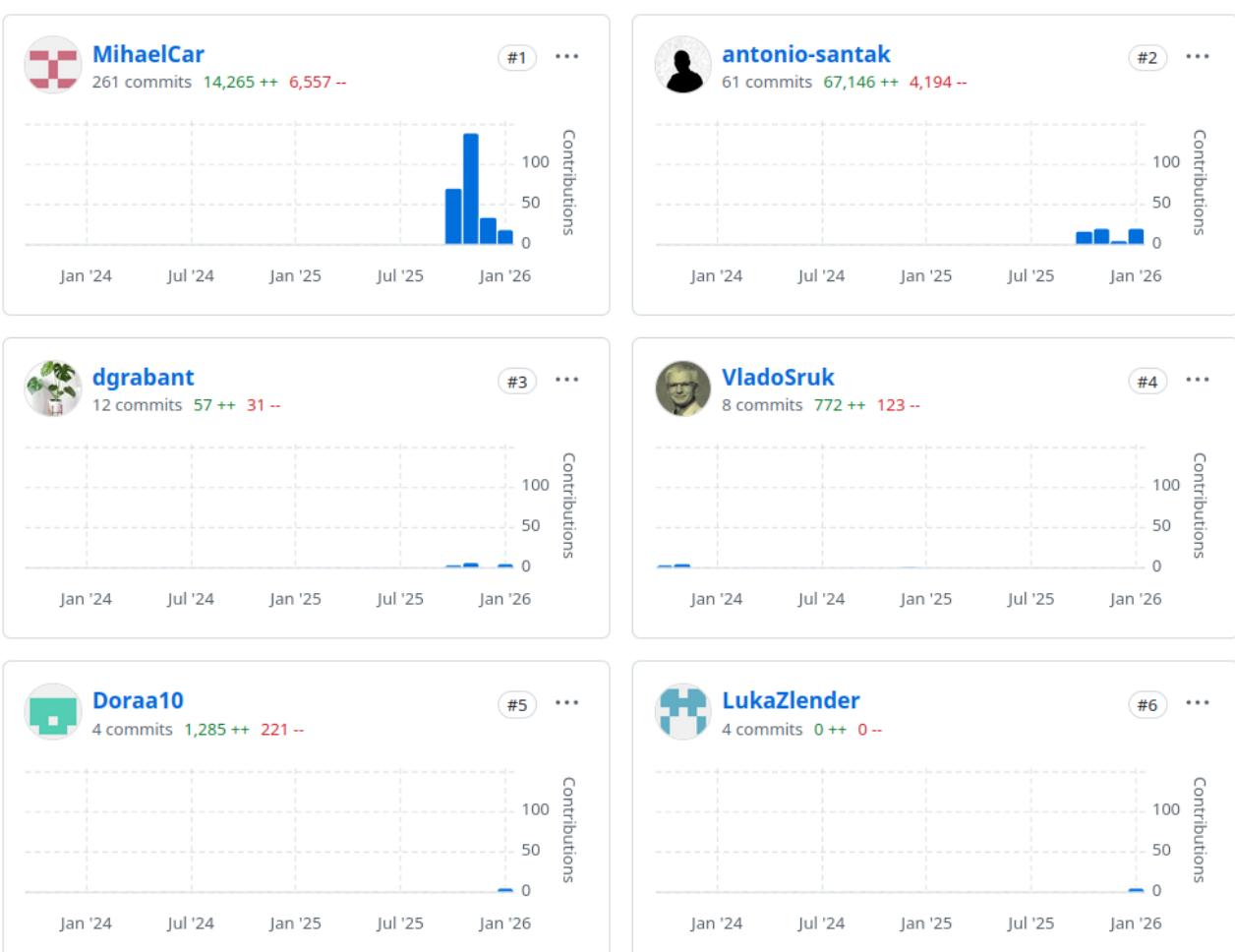
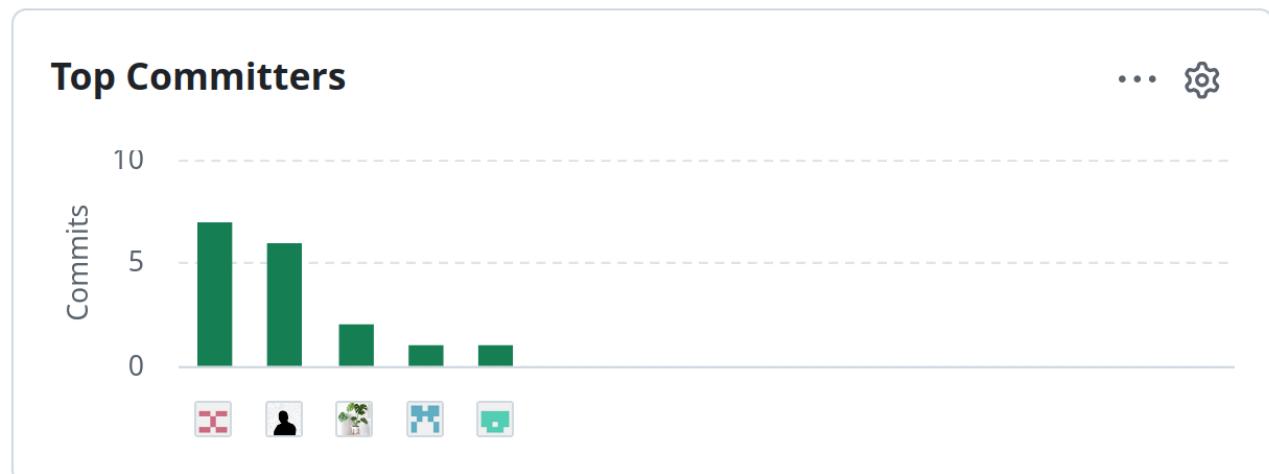
Opis ostalih zahtjeva	2					
Arhitektura i dizajn sustava	4		4			
Baza podataka	2		12			
Dijagram razreda	0.5		4			
Dijagram stanja					2.5	
Dijagram aktivnosti					2	
Dijagram komponenti					3	
Korištenje tehnologije i alati	2		2	0.2	2	
Ispitivanje programskog rješenja	3	25	4	30		
Dijagram razmještaja					2	
Upute za puštanje u pogon			0.5			
Dnevnik sastajanja	1				3.5	
Izrada plana rada					1.5	
Zaključak i budući rad	1					
Popis literature	0.5					
Izrada početne stranice		15		20		
Izrada baze podataka	0.5		4		0.5	
Spajanje s bazom podataka			2			
Izrada aplikacije – backend/frontend		85	72	100		
Uređivanje dokumentacije	6	0.5	1	1		
Izrada prezentacije	5					
Ukupno	74.5	125.5	105.5	153.2	17	2

Ukupan broj sati



napomena: dijagram generiran pomoću alata ChatGPT

Dijagram pregleda promjena



Kjučni izazovi i rješenja

Najveći izazovi tokom razvoja bili su: komunikacija, usklađivanje tima, bonton, individualne različitosti, različita predznanja te raznolik porov i motivacija svakog pojedinca kao i fluktuacije u angažmanu pojedinaca. Uz to, vremensko ograničenje. Na takve izazove, nije bilo lako utjecati, do kraja ih razriješiti jer se ljudi teško mijenjaju. Ipak, ostvaren je napredak, a izazovi ipak nisu bili preveliki da nadvladaju samo ostvarenje projekta. Naravno, cijeli projekt

ima pojedinih mana i manjkavosti, no cijeli se tim trudio da toga bude čim manje, kao i da projekt napravimo čim bolje. Rješenje tih izazova bilo je transparentno komunikacije kao i kolegijalnost.

▼ Pages 14

Find a page or section...

- ▶ [Home](#)
- ▶ [1. Opis projektnog zadatka](#)
- ▶ [2. Analiza zahtjeva](#)
- ▶ [3. Specifikacija zahtjeva sustava](#)
- ▶ [4. Arhitektura i dizajn sustava](#)
- ▶ [5. Arhitektura komponenata i razmještaja](#)
- ▶ [6. Ispitivanje programskog rješenja](#)
- ▶ [7. Tehnologije za implementaciju aplikacije](#)
- ▶ [8. Upute za puštanje u pogon](#)
- ▶ [9. Zaključak i budući rad](#)
- ▶ [A. Popis literature](#)
- ▶ [A. Dnevnik promjena dokumentacije](#)
- ▼ **B. Prikaz aktivnosti grupe**
 - Plan rada
 - Ganttov dijagram
 - Tablica aktivnosti
 - Ukupan broj sati
 - Dijagram pregleda promjena
 - Kjučni izazovi i rješenja
- ▶ [D. Stranica za sastanke](#)

+ Add a custom sidebar

Clone this wiki locally

<https://github.com/dgrabant/Auto-servis.wiki.git>



Code Issues Pull requests Actions Projects 1 Wiki Security Insights Settings

D. Stranica za sastanke

[Edit](#)[New page](#)[Jump to bottom](#)

David edited this page 5 days ago · [8 revisions](#)

Easter egg, stranica koja je nekada služila kao pomoć pri sastancima, na njoj su postojale točke reda sastanka.

Harlemonova kočija

Pages 14

Find a page or section...

- ▶ [Home](#)
- ▶ [1. Opis projektnog zadatka](#)
- ▶ [2. Analiza zahtjeva](#)
- ▶ [3. Specifikacija zahtjeva sustava](#)
- ▶ [4. Arhitektura i dizajn sustava](#)
- ▶ [5. Arhitektura komponenata i razmještaja](#)
- ▶ [6. Ispitivanje programskog rješenja](#)
- ▶ [7. Tehnologije za implementaciju aplikacije](#)
- ▶ [8. Upute za puštanje u pogon](#)
- ▶ [9. Zaključak i budući rad](#)
- ▶ [A. Popis literature](#)
- ▶ [A. Dnevnik promjena dokumentacije](#)
- ▶ [B. Prikaz aktivnosti grupe](#)
- ▶ [D. Stranica za sastanke](#)

+ Add a custom sidebar

Clone this wiki locally

<https://github.com/dgrabant/Auto-servis.wiki.git>

