



Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

1. Opis projektnog zadatka

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

Ivan-Josip-Kardum edited this page on Nov 13, 2025 · [4 revisions](#)

Opis i cilj projektnog zadatka

Budući da svaki organizator umjetničke (keramičke) radionice neovisno o drugima organizira radionice i objavljuje informacije, zainteresirani polaznici često moraju pratiti obavijesti na različitim web stranicama te društvenim mrežama organizatora kako bi otkrili sve informacije koje ih zanimaju. Dodatni problem predstavlja postojeći komplikiran proces kupovine i prodaje umjetničkih djela te manjak jedinstvenog rješenja gdje se mogu lako pregledati velike količine proizvoda te usporediti s drugim ponudama.

Cilj projektnog zadatka je izraditi web aplikaciju koja bi omogućila organizatorima objavljivanje radionica na jednom mjestu, a polaznicima jednostavno pregledavanje svih dostupnih opcija, te filtriranje s obzirom na vlastite preferencije i mogućnost prijave. Također bi se mogle organizirati izložbe radova polaznika te prodaja izloženih radova. Bila bi dostupna i online trgovina za prodaju jedinstvenih radova umjetnika (instruktora).

Pomoću aplikacije korisnici bi mogli:

- jednostavno pregledavati, filtrirati i rezervirati termine radionica keramike
- obaviti plaćanje putem sigurnog vanjskog servisa
- organizirati radionice i izložbe
- prodavati i kupovati proizvode u online trgovini
- ostavljati svoje dojmove u obliku komentara, recenzija i odgovora
- primati novosti o nadolazećim događanjima

Cilj je na jednom mjestu objediniti sva događanja vezana uz keramiku kako bi se zainteresirani lakše povezali i otkrili najbitnije informacije.

Potencijalna korist projekta

Korist od razvoja aplikacije imali bi i organizatori i polaznici.

Polaznicima bi aplikacija omogućila:

- pregled svih dostupnih radionica i izložbi u blizini
- rezervaciju termina i sigurno plaćanje
- jednostavnu prijavu na izložbe
- filtriranje prikaza po vrsti događanja, datumu, lokaciji, cijeni i ocjeni
- pregled recenzija i ocjena drugih polaznika

- prikaz rezerviranih termina u kalendaru
- pregled i kupovinu izloženih proizvoda
- dobivanje obavijesti o događanjima koja bi ih mogla zanimati

Organizatorima bi aplikacija omogućila:

- objavljivanje događanja koja organiziraju
- vidljivost široj publici
- izgradnju profesionalnog online identiteta
- jednostavnu i sigurnu naplatu
- lakši pregled rada konkurenčije
- odobravanje prijava za izložbe
- pregled prijavljenih polaznika za radionice i izložbe
- dobivanje povratnih informacija od polaznika

Postojeća slična rješenja

Na tržištu postoje različite platforme za organizaciju događanja, pregled umjetničkih radova ili online prodaju, no nijedna ne pokriva sve funkcionalnosti obuhvaćene ovim projektnim zadatkom. U nastavku su prikazana dva najrelevantnija primjera i njihove ključne razlike u odnosu na predloženo rješenje.

Eventbrite

Online tickets events

Filters

1 filter applied

Online X

Clear All

Category

 Business

 Food & Drink

 Health

 Music

[View more](#)

Date

 Today

 Tomorrow

 This weekend

 Pick a date...


Going fast

AI Executive Lab: Scaling Agencies & Closing Deals Faster with AI Agents

Today • 8:00 PM GMT+1

Free

Promoted ⓘ



Almost full

Beyond Command & Control: A Governance Shift for 21st-Century Organizations

Today • 9:00 PM GMT+1

Free

Promoted ⓘ

Eventbrite je aplikacija koja nudi pregled i objavu događanja. Nije specijalizirana za umjetničke radionice nego za sve vrste događanja. Nudi jednostavnu naplatu ulaznica te pregled statistike za organizatora. Ne nudi sustav prijave polaznika na kreativne radionice niti mogućnost objave fotografija radova i prodajnih proizvoda, što su ključne funkcionalnosti koje nedostaju.

Artsy

[All Filters](#)[Rarity ▾](#)[Medium ▾](#)[Price Range ▾](#)

Martin Whatson

[Unique](#) [Painting](#)

US\$7,800



Kunstrasen



Can solo, 2016

APC Gallery

US\$1,200

[Increased Interest](#)

Hiroshi Sato

In News, 2025

Marrow Gallery

US\$2,500



Artsy je aplikacija za pregled i kupovinu radova umjetnika. Nudi širok spektar umjetnina te lagano filtriranje po željama (umjetnik, cijena, veličina...). Također prikazuje određene izložbe i radionice, međutim to se odnosi gotovo isključivo na veća svjetska događanja. Nema opciju pregleda i prijave na radionice gdje polaznici mogu sami nešto izraditi. Zbog toga Artsy nije prikladan alat za lokalne radionice keramike.

Skup korisnika zainteresiranih za rješenje

- organizatori (umjetničke organizacije i centri, galerije, samostalni umjetnici i instruktori)
- polaznici (amateri, učenici, škole i udruge)
- ljubitelji izložbi
- kupci i kolezionari ručno izrađenih radova

Mogućnost prilagodbe rješenja

S obzirom na modularnu arhitekturu, aplikacija se može jednostavno širiti dodatnim funkcionalnostima i prilagođavati novim potrebama korisnika.

Moguće prilagodbe:

- razvoj mobilne aplikacije za Android i iOS
- podrška za više jezika
- prilagodba za druge vrste umjetnosti (slikarstvo, kiparstvo)
- loyalty sistem, skupljanje bodova i kupona
- dodatne metode plaćanja (KEKS Pay, Aircash)
- integracija s društvenim mrežama (TikTok, Instagram, Facebook)

Opseg projektnog zadatka

Opseg projektnog zadatka obuhvaća izradu web aplikacije koja omogućava:

- jednostavnu registraciju i autentifikaciju korisnika
- odabir korištenja kao polaznik ili organizator te izradu profila
- stvaranje radionica i izložbi
- pregled i filtriranje dostupnog sadržaja
- rezervacije i otkazivanja termina
- prijave na izložbe
- online trgovinu keramičkih radova
- ostavljanje komentara nakon izložbi
- ostavljanje recenzija za proizvode
- integraciju s vanjskim sustavima za kalendar, autentifikaciju i plaćanje
- sigurnu obradu plaćanja
- bazu podataka koja čuva podatke za veliki broj korisnika
- enkriptirano čuvanje lozinki profila
- administraciju sustava

Moguće nadogradnje projektnog zadatka

Moguće nadogradnje:

- razvoj mobilne aplikacije za Android i iOS
- dodatne metode plaćanja (KEKS Pay, Aircash)
- loyalty sistem, skupljanje bodova i kupona
- podrška za veliki broj jezika
- AI asistent koji daje personalizirane preporuke
- 3D prikaz radova za prodaju
- mogućnosti dogovora za grupne radionice (rođendani, škole)

Zaključak

Projektni zadatak izrade web aplikacije predstavlja jednostavno, skalabilno rješenje za polaznike i organizatore, stvara nove mogućnosti za promociju sadržaja te rješava stvaran problem manjka sličnog rješenja za umjetničku zajednicu. Podržava umjetnike te pomaže polaznicima steći nova znanja i ojačati kreativnost. Platforma postavlja temelje za daljnji razvoj digitalne infrastrukture kreativne zajednice i omogućuje dugoročno širenje funkcionalnosti prema potrebama korisnika.

Stanari 

▼ Pages 13

Find a page or section...

- ▶ [Home](#)
- ▼ [1. Opis projektnog zadatka](#)
 - Opis i cilj projektnog zadatka
 - Potencijalna korist projekta
 - Postojeća slična rješenja
 - Eventbrite
 - Artsy
 - Skup korisnika zainteresiranih za rješenje
 - Mogućnost prilagodbe rješenja
 - Opseg projektnog zadatka
 - Moguće nadogradnje projektnog zadatka
 - Zaključak
- ▶ [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](#)
- ▶ [B. Dnevnik promjena dokumentacije](#)
- ▶ [C. Prikaz aktivnosti grupe](#)

+ Add a custom sidebar

Clone this wiki locally

<https://github.com/Pavle-Stanarevic/Stanari.wiki.git>





Pavle-Stanarevic / Stanari



Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

Home

Edit

New page

Jump to bottom

Bruno-Komocar edited this page on Oct 23, 2025 · [13 revisions](#)

Programsko inženjerstvo ak.god 2025./2026

Sveučilište u Zagrebu

Fakultet elektrotehnike i računarstva

ClayPlay

Tim: <TG15.2>

Stanari

Nastavnik: Vlado Sruk

Stanari -pencil icon

▼ Pages 13

Find a page or section...

▼ Home

- Programsko inženjerstvo ak.god 2025./2026
- Sveučilište u Zagrebu
- Fakultet elektrotehnike i računarstva
- ClayPlay

Tim: <TG15.2>

Stanari

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](#)
- ▶ [B. Dnevnik promjena dokumentacije](#)
- ▶ [C. Prikaz aktivnosti grupe](#)

+ Add a custom sidebar

Clone this wiki locally

<https://github.com/Pavle-Stanarevic/Stanari.wiki.git>





Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

2. Analiza zahtjeva

[Edit](#)[New page](#)[Jump to bottom](#)Pavle-Stanarevic edited this page on Nov 13, 2025 · [10 revisions](#)

Funkcionalni zahtjevi

ID zahtjeva	Opis	Prioritet	Izvor	Kriteriji prihvaćanja
F-001	Sustav omogućuje registraciju polaznika i organizatora radionica putem vanjskog autentifikacijskog servisa.	Visok	Zahtjev naručitelja	Korisnik se može registrirati putem vanjskog autentifikacijskog servisa, aplikacija kreira profil s e-mailom i imenom korisnika.
F-002	Sustav omogućuje prijavu korisnika putem vanjskog autentifikacijskog servisa.	Visok	Zahtjev naručitelja	Korisnik se može uspješno prijaviti i pristupiti svojim funkcionalnostima ovisno o ulozi.
F-003	Sustav omogućuje organizatorima da kreiraju i uređuju profile s osnovnim podatcima, fotografijama i kontaktima.	Visok	Zahtjev dionika	Profil organizatora sadrži ime, fotografiju, adresu, kontakt telefon i popis radionica.
F-004	Sustav omogućuje organizatorima da kreiraju radionice s nazivom, opisom, trajanjem, datumom i vremenom, lokacijom, brojem mjesta i cijenom.	Visok	Zahtjev dionika	Radionica je vidljiva polaznicima s kompletним informacijama, a organizator može ažurirati podatke.
F-005	Sustav omogućuje polaznicima da pregledavaju dostupne radionice s detaljima i filtriraju po datumu, lokaciji i cijeni.	Visok	Zahtjev dionika	Polaznici vide točne podatke radionica i mogu koristiti filtere za pretragu.
F-006	Sustav omogućuje polaznicima da rezerviraju radionicu i odaberu način plaćanja. Sustav vanjskom uslugom kalendarja prikazuje dostupne termine i omogućuje evidentiranje rezervacija.	Visok	Zahtjev naručitelja	Rezervacija je potvrđena, plaćanje procesirano, a termin evidentiran u sustavu i u korisnikovom kalendaru.

ID zahtjeva	Opis	Prioritet	Izvor	Kriteriji prihvatanja
F-007	Sustav omogućuje polaznicima da otkažu rezervaciju.	Srednji	Zahtjev naručitelja	Sustav omogućuje otkaživanje najkasnije 48 sati prije početka radionice i ažurira dostupnost mjesta.
F-008	Sustav omogućuje organizatorima da dodaju fotografije i radove polaznika u svoje profile i radionice.	Srednji	Zahtjev dionika	Dodane slike i radovi vidljivi su polaznicima i ostalim korisnicima.
F-009	Sustav omogućuje online trgovinu keramičkih proizvoda s opisima, cijenama i slikama.	Visok	Zahtjev naručitelja	Kupci mogu pregledavati proizvode, dodavati u košaricu i izvršiti plaćanje online.
F-010	Sustav omogućuje kupcima da pretražuju proizvode prema kategoriji i cijeni.	Srednji	Zahtjev naručitelja	Sustav omogućuje filtriranje i sortiranje proizvoda prema parametrima.
F-011	Sustav omogućuje kupcima da ostavljaju recenzije i ocjene kupljenih proizvoda.	Srednji	Povratne informacije korisnika	Recenzije se prikazuju uz proizvod, a prosječna ocjena se računa automatski.
F-012	Sustav omogućuje organizaciju izložbi s nazivom, lokacijom, datumom i prikazom radova.	Srednji	Zahtjev dionika	Polaznici mogu pregledati izložbu, prijaviti se i organizator može odobriti prijave.
F-013	Sustav omogućuje polaznicima da se prijavljuju na izložbu putem platforme.	Srednji	Zahtjev dionika	Polaznik može poslati prijavu za izložbu; organizator može odobriti ili odbiti prijavu.
F-014	Sustav omogućuje sudionicima izložbi da postavljaju komentare i slike nakon događaja.	Nizak	Povratne informacije korisnika	Komentari i slike se prikazuju u odjeljku izložbe.
F-015	Sustav omogućuje administratorima da upravljaju korisnicima i odobravaju nove profile.	Visok	Zahtjev naručitelja	Administrator može pregledavati, odobravati i blokirati korisničke profile.
F-016	Sustav omogućuje administratorima da određuju cijene članarine za organizatore radionica.	Visok	Zahtjev naručitelja	Sustav omogućuje unos, izmjenu i spremanje cijena članarine.
F-017	Sustav omogućuje plaćanje članarine organizatora putem	Visok	Zahtjev naručitelja	Organizator može odabrati način plaćanja, sustav

ID zahtjeva	Opis	Prioritet	Izvor	Kriteriji prihvatanja
	vanjskog servisa za obradu plaćanja.			procesira uplatu i potvrđuje plaćanje.

Nefunkcionalni zahtjevi

Zahtjevi za performanse i dostupnost

ID zahtjeva	Opis	Prioritet
NF-1.1	Stranica se mora učitavati unutar 3 sekunde.	Visok
NF-1.2	Sustav mora biti dostupan barem 95% vremena.	Visok
NF-1.3	Sustav mora podržavati veći broj istovremenih korisnika bez značajnog pada performansi.	Visok

Zahtjevi za sigurnost

ID zahtjeva	Opis	Prioritet
NF-2.1	Registracija i prijava moraju koristiti sigurnosne protokole (HTTPS, OAuth 2.0).	Visok
NF-2.2	Lozinke korisnika moraju biti pohranjene enkriptirane.	Visok
NF-2.3	Sva plaćanja moraju se obavljati preko sigurnog vanjskog servisa.	Visok

Zahtjevi za održavanje i dokumentaciju

ID zahtjeva	Opis	Prioritet
NF-3.1	Sustav treba biti oblikovan tako da omogućuje jednostavno održavanje.	Visok
NF-3.1.7	Sustav treba imati dovoljnu dokumentaciju.	Visok
NF-3.1.7.1	Kôd sustava treba biti dokumentiran prema "Code Conventions" dostupnim za korištene programske jezike.	Visok
NF-3.1.3	Sustav treba imati plan implementacije za pravilno postavljanje sustava i dokument koji opisuje funkcionalne i nefunkcionalne zahtjeve.	Visok
NF-3.1.7.3	Sustav treba biti popraćen "Priručnikom za rad" koji opisuje pravilnu upotrebu sustava.	Visok

Zahtjevi za korisničko iskustvo

ID zahtjeva	Opis	Prioritet
NF-4.1	Korisničko sučelje treba biti responzivno i prilagođeno uređajima s različitim rezolucijama ekrana.	Visok
NF-4.2	Sustav treba biti intuitivan i jednostavan za korištenje s jasnim uputama i navigacijom.	Visok

Dionici i njihovi funkcionalni zahtjevi

- Polaznici
- Organizatori
- Administratori sustava

A-1 Polaznik (inicijator) može:

- Pregledavati radionice (F-005)
- Rezervirati radionicu (F-006)
- Otkazati rezervaciju (F-007)
- Kupovati keramičke proizvode (F-009, F-010)
- Ostavljati recenzije (F-011)
- Prijavljivati se na izložbe (F-012, F-013)
- Ostavljati komentare (F-014)

A-2 Organizator (inicijator) može:

- Kreirati i uređivati profil (F-003)
- Kreirati i uređivati radionice (F-004)
- Dodavati fotografije i radove (F-008)
- Odobravati prijave za izložbe (F-012)
- Plaćati članarinu i koristiti funkcionalnosti sustava (F-016, F-017)

A-3 Administrator (inicijator) može:

- Upravljati korisnicima (F-015)
- Određivati cijene članarina (F-016)

A-4 Baza podataka (sudionik):

- Baza podataka pohranjuje, dohvaća i ažurira podatke potrebne za izvođenje funkcionalnosti sustava.

Stanari 

▼ Pages 13

Find a page or section...

- ▶ [Home](#)
- ▶ [1. Opis projektnog zadatka](#)
- ▼ [2. Analiza zahtjeva](#)
 - Funkcionalni zahtjevi
 - Nefunkcionalni zahtjevi
 - Zahtjevi za performanse i dostupnost
 - Zahtjevi za sigurnost
 - Zahtjevi za održavanje i dokumentaciju
 - Zahtjevi za korisničko iskustvo
 - Dionici i njihovi funkcionalni zahtjevi
 - A-1 Polaznik (inicijator) može:
 - A-2 Organizator (inicijator) može:
 - A-3 Administrator (inicijator) može:
 - A-4 Baza podataka (sudionik):
- ▶ [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](#)
- ▶ [B. Dnevnik promjena dokumentacije](#)
- ▶ [C. Prikaz aktivnosti grupe](#)

+ Add a custom sidebar

Clone this wiki locally

<https://github.com/Pavle-Stanarevic/Stanari.wiki.git>





Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

3. Specifikacija zahtjeva sustava

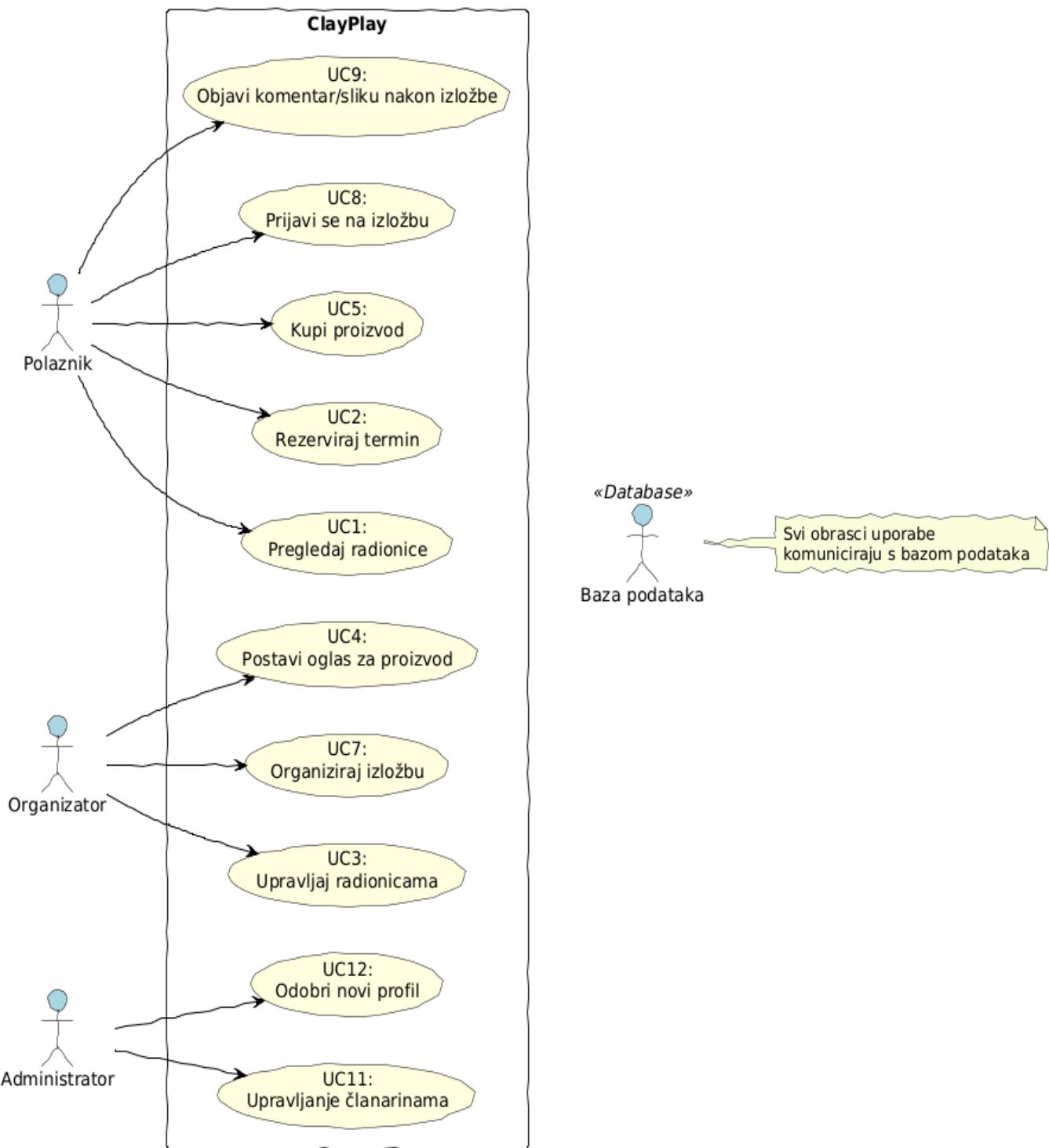
[Edit](#)[New page](#)[Jump to bottom](#)

Marko-Malenica edited this page on Nov 14, 2025 · [40 revisions](#)

Obrasci uporabe

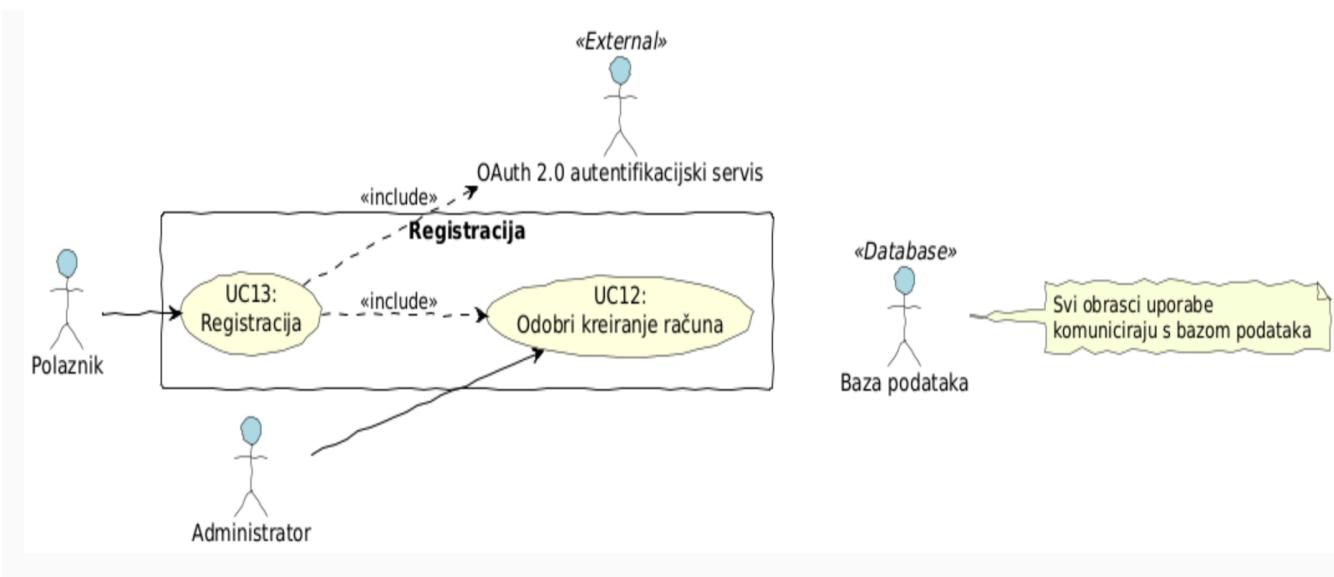
Dijagrami obrazaca uporabe

1. Visokorazinski dijagram obrazaca uporabe cijelog sustava

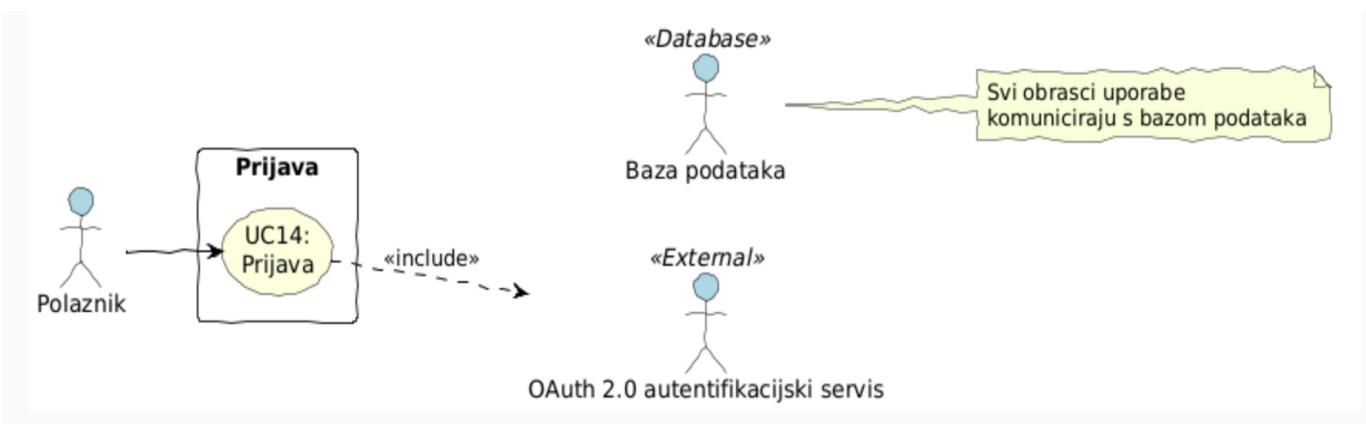


2. Dijagram obrazaca uporabe za ključne značajke

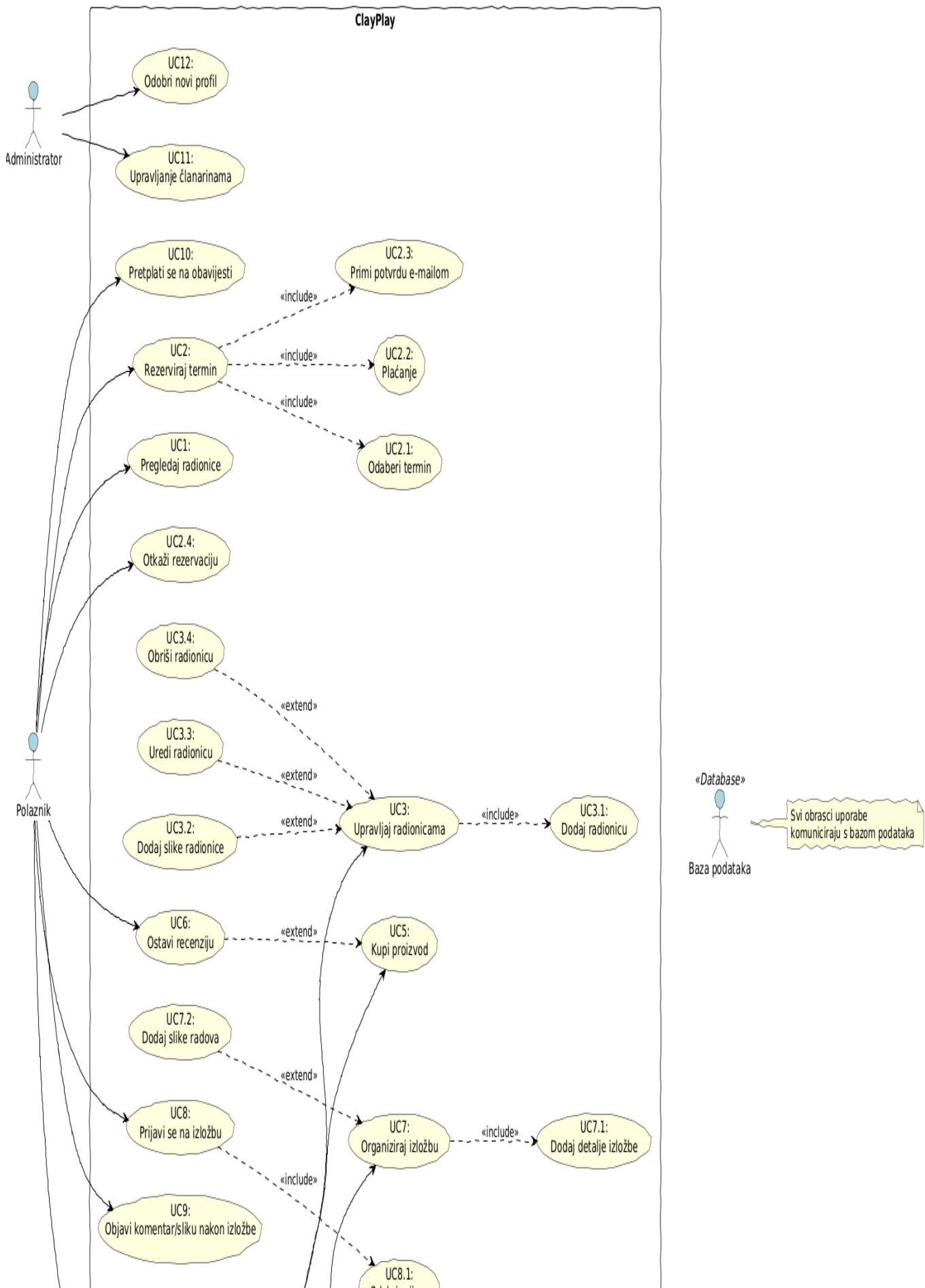
Dijagram obrasca uporabe za registraciju:

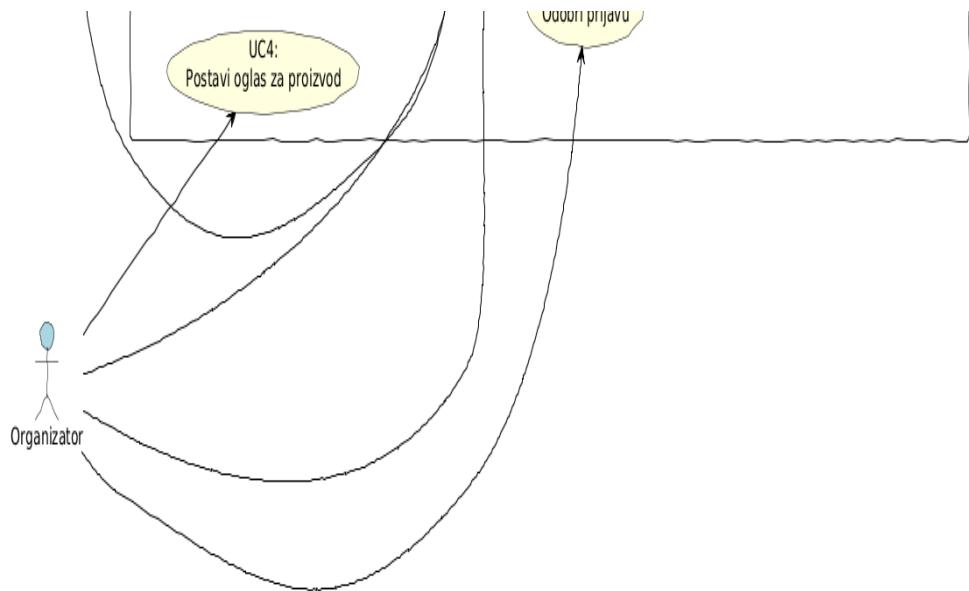


Dijagram obrasca uporabe za prijavu:

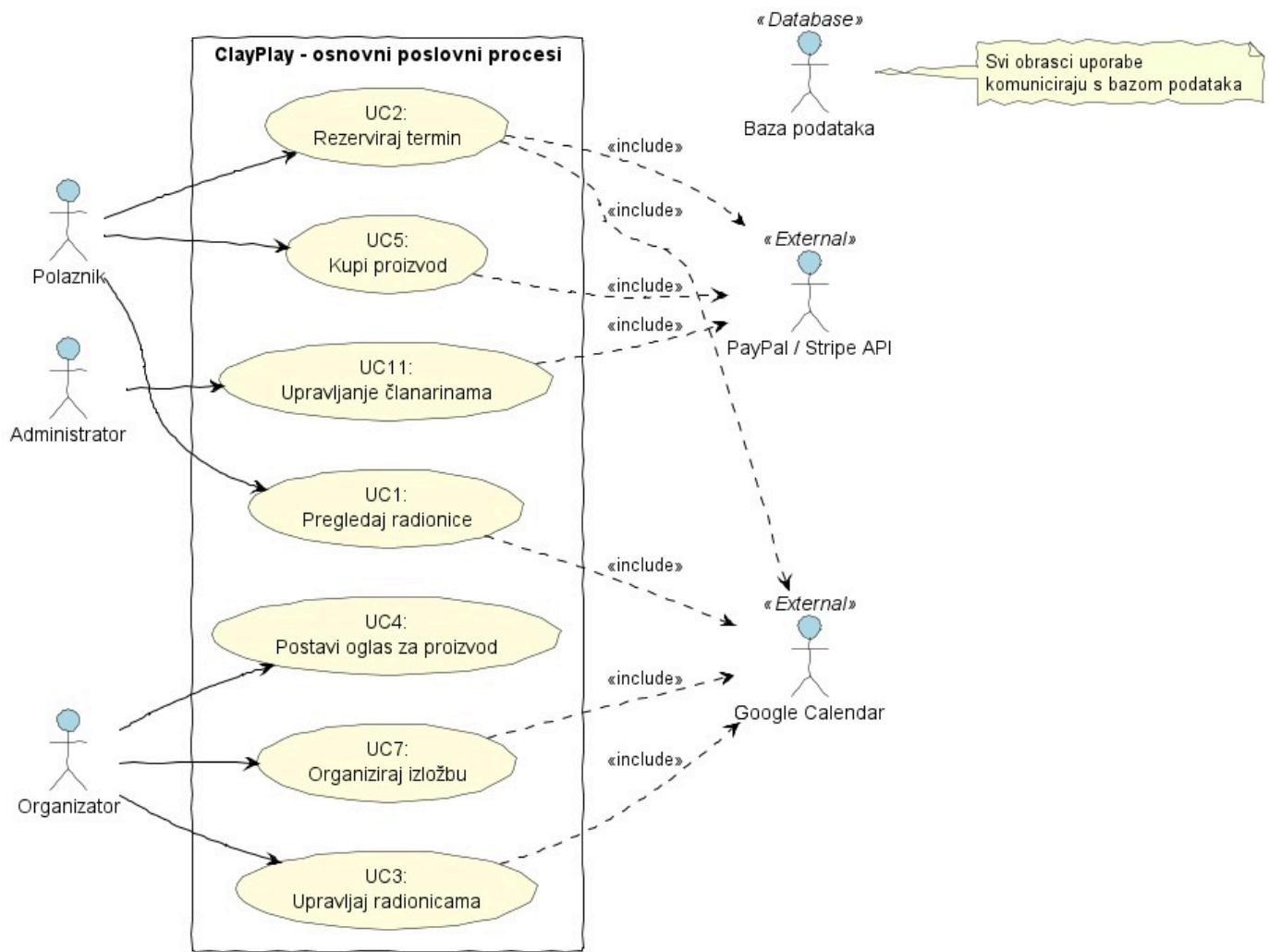


3. Dijagram obrazaca uporabe za korisničke uloge



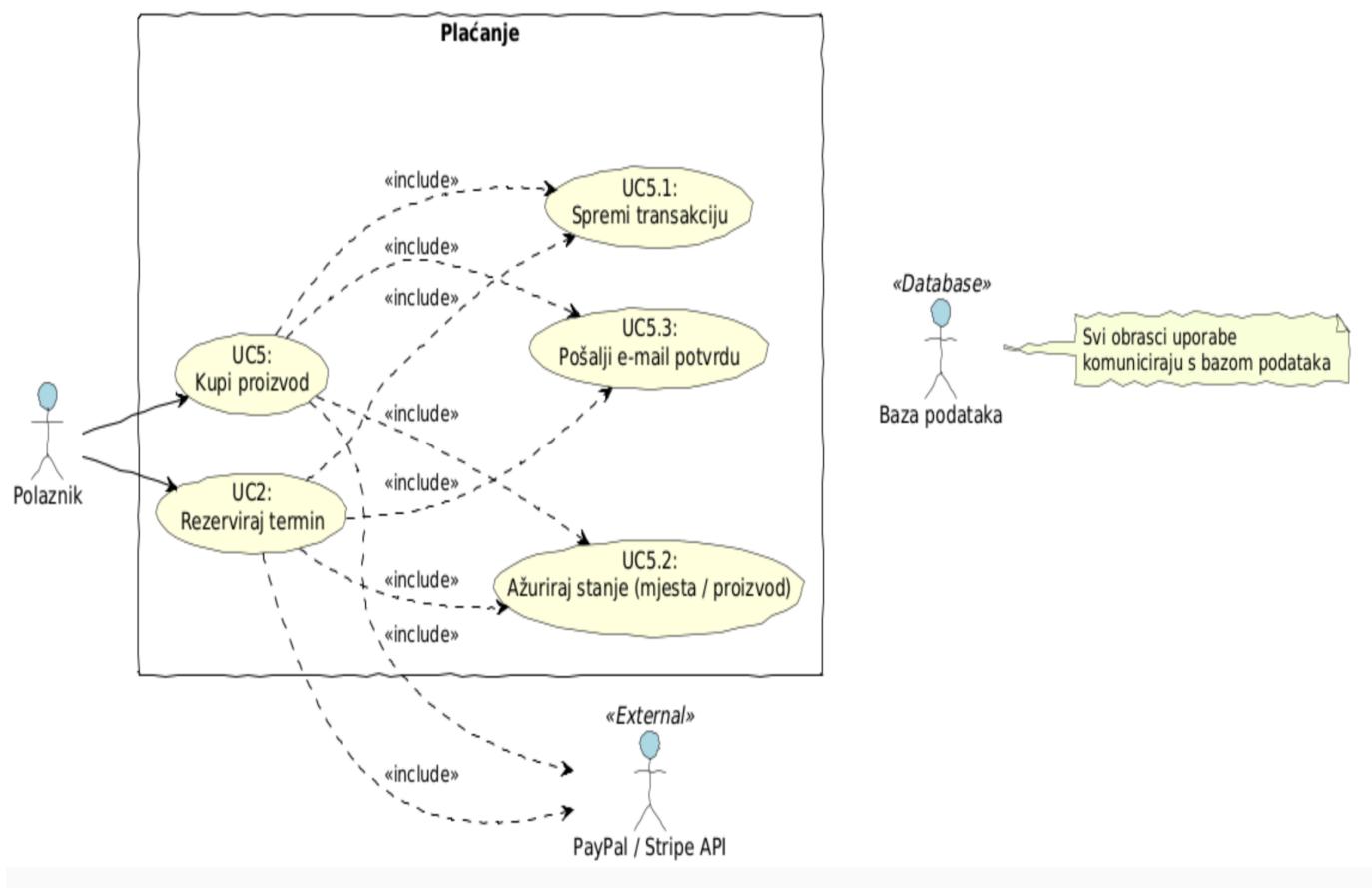


4. Dijagram obrazaca uporabe za osnovne poslovne procese

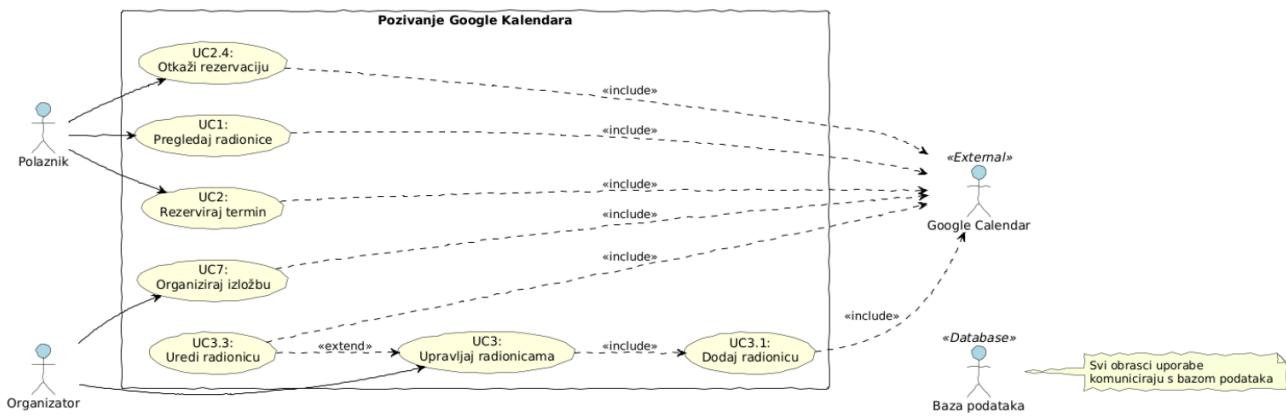


5. dijagram obrazaca uporabe za kritične sustave i integracije

Dijagram obrasca uporabe za plaćanje:



Dijagram obrasca uporabe za korištenje kalendara:



Opis obrazaca uporabe

UC1 – Pregledaj radionice

- Glavni sudionik: Polaznik
- Cilj: Pronaći i pregledati sve nadolazeće keramičke radionice.

- Sudionici: Baza podataka
- Preduvjet: Nije potrebna prijava.
- Opis osnovnog tijeka:
 1. Polaznik otvara stranicu „Radionice“.
 2. Sustav dohvaća listu aktivnih radionica (naziv, datum, cijena, broj slobodnih mjesta).
 3. Polaznik može koristiti filtere (datum, cijena, lokacija).
 4. Sustav prikazuje rezultate.
- Opis mogućih odstupanja:
 1. Nema aktivnih radionica → poruka „Trenutno nema radionica“.

UC2 – Rezerviraj termin

- Glavni sudionik: Polaznik
- Cilj: Rezervirati i platiti mjesto na radionici
- Sudionici: Baza podataka, vanjski servis plaćanja
- Preduvjet: Polaznik je prijavljen
- Opis osnovnog tijeka:
 1. Polaznik odabere radionicu (UC1)
 2. Pritiskom na „Rezerviraj“ pokreće se UC2.1
 3. Sustav provjerava slobodna mjesta (F-006)
 4. Pokreće se UC2.2 (plaćanje)
 5. Nakon uspješnog plaćanja sustav smanjuje slobodna mjesta, spremi rezervaciju u bazu i šalje e-mail potvrdu (UC2.3) (F-007)
- Opis mogućih odstupanja:
 1. Nema slobodnih mjesta → poruka „Radionica je puna“
 2. Plaćanje neuspješno → poruka „Transakcija odbijena“; rezervacija se poništava

UC2.1 – Odaberi termin

- Glavni sudionik: Polaznik
- Cilj: Odabrati točan datum i vrijeme unutar ponuđenih termina radionice
- Sudionici: Baza podataka
- Preduvjet: Radionica ima više termina
- Opis osnovnog tijeka:
 1. Sustav prikazuje kalendar dostupnih termina
 2. Polaznik bira termin
- Opis mogućih odstupanja:
 1. Nema dostupnih termina → poruka „Nema slobodnih termina“

UC2.2 – Plaćanje

- Glavni sudionik: Polaznik
- Cilj: Izvršiti plaćanje rezervacije
- Sudionici: Vanjski servis za obradu plaćanja
- Preduvjet: Odabran termin
- Opis osnovnog tijeka:
 1. Sustav prosljeđuje iznos API-u vanjskog servisa za obradu plaćanja
 2. Vanjski servis vraća token uspješnosti
- Opis mogućih odstupanja:
 1. Plaćanje neuspješno → poruka „Transakcija odbijena“

UC2.3 – Primi potvrdu e-mailom

- Glavni sudionik: Polaznik
- Cilj: Primiti e-mail potvrdu s detaljima rezervacije
- Sudionici: Baza podataka, e-mail servis
- Preduvjet: Plaćanje uspješno
- Opis osnovnog tijeka:
 1. Sustav generira potvrdu
 2. Šalje potvrdu na e-mail polaznika
- Opis mogućih odstupanja:
 1. Slanje e-maila neuspješno → poruka „Došlo je do greške pri slanju potvrde, pokušajte kasnije“

UC2.4 – Otkaži rezervaciju

- Glavni sudionik: Polaznik
- Cilj: Poništiti rezervaciju i dobiti povrat novca
- Sudionici: Baza podataka, vanjski servis plaćanja
- Preduvjet: Do početka radionice ≥ 48 h
- Opis osnovnog tijeka:
 1. Polaznik otvara „Moje rezervacije“
 2. Bira rezervaciju i klikna „Otkaži“
 3. Sustav pokreće proces vraćanja novca
 4. Mjesto postaje slobodno; rezervacija se briše
- Opis mogućih odstupanja:
 1. Rok < 48 h → poruka „Otkazivanje nije moguće“

UC3 – Upravljanje radionicama

- Glavni sudionik: Organizator
- Cilj: Izvršiti CRUD operacije nad vlastitim radionicama
- Sudionici: Baza podataka
- Preduvjet: Organizator je prijavljen i odobren
- Opis osnovnog tijeka:
 1. Otvara „Moje radionice“
 2. Sustav prikazuje listu
 3. Bira: dodaj (UC3.1), uredi (UC3.3), obriši (UC3.4), dodaj slike (UC3.2)
- Opis mogućih odstupanja:
 1. Lista prazna → poruka „Nemate kreiranih radionica“

UC3.1 – Dodaj radionicu

- Glavni sudionik: Organizator
- Cilj: Kreirati novu radionicu
- Sudionici: Baza podataka
- Preduvjet: /
- Opis osnovnog tijeka:
 1. Unose se naziv, opis, datum, vrijeme, lokacija, cijena, broj mjesta
 2. Sustav zapisuje u bazu
- Opis mogućih odstupanja:
 1. Podaci nepotpuni → poruka „Sva polja su obavezna“

UC3.2 – Dodaj sliku radionice

- Glavni sudionik: Organizator
- Cilj: Uploadati fotografije radionice
- Sudionici: Baza podataka
- Preduvjet: Radionica je kreirana
- Opis osnovnog tijeka:
 1. Upload JPG/PNG
 2. Putanja slike se sprema u bazu
- Opis mogućih odstupanja:
 1. Datoteka veća od 5 MB → poruka „Datoteka je prevelika“
 2. Format nije JPG/PNG → poruka „Nepodržani format slike“

UC3.3 – Uredi radionicu

- Glavni sudionik: Organizator

- Cilj: Izmijeniti podatke o radionicici
- Sudionici: Baza podataka
- Preduvjet: /
- Opis osnovnog tijeka:
 1. Organizator mijenja bilo koji podatak osim ID-a
 2. Promjene se trenutno prikazuju polaznicima
- Opis mogućih odstupanja:
 1. Podaci nepotpuni → poruka „Sva polja su obavezna“

UC3.4 – Obriši radionicu

- Glavni sudionik: Organizator
- Cilj: Ukloniti radionicu iz sustava
- Sudionici: Baza podataka
- Preduvjet: Nema aktivnih rezervacija
- Opis osnovnog tijeka:
 1. Brišu se svi podaci vezani uz radionicu
- Opis mogućih odstupanja:
 1. Postoje rezervacije → sustav traži da se prvo otkažu

UC4 – Postavi oglas za proizvod

- Glavni sudionik: Organizator
- Cilj: Objaviti keramički proizvod za prodaju
- Sudionici: Baza podataka
- Preduvjet: Prijavljen i odobren profil
- Opis osnovnog tijeka:
 1. Otvara „Moja trgovina“ → „Novi proizvod“
 2. Unosi naziv, opis, cijenu, količinu, uploada slike
 3. Sustav objavljuje proizvod u javnoj trgovini
- Opis mogućih odstupanja:
 1. Cijena ≤ 0 → poruka „Cijena mora biti veća od 0“

UC5 – Kupi proizvod

- Glavni sudionik: Polaznik
- Cilj: Kupiti gotov keramički proizvod
- Sudionici: Baza podataka, PayPal/Stripe
- Preduvjet: Proizvod objavljen i količina ≥ 1

- Opis osnovnog tijeka:

1. Dodaje proizvod u košaricu
2. Odabire adresu dostave
3. Pokreće se plaćanje (UC5.1)
4. Nakon uspjeha: količina se smanjuje, kreira narudžba, šalje e-mail potvrda
5. Polaznik može ostaviti recenziju (UC6)

- Opis mogućih odstupanja:

1. Proizvod nije dostupan → poruka „Proizvod trenutno nije dostupan“
2. Plaćanje neuspješno → poruka „Transakcija odbijena“

UC5.1 – Plaćanje

- Glavni sudionik: Polaznik

- Cilj: Izvršiti online plaćanje

- Sudionici: Vanjski servis

- Preduvjet: /

- Opis osnovnog tijeka:

1. Sustav proslijeđuje iznos vanjskom servisu
2. Servis vraća token uspješnosti

- Opis mogućih odstupanja:

1. Plaćanje neuspješno → poruka „Transakcija odbijena“

UC6 – Ostavi recenziju

- Glavni sudionik: Polaznik

- Cilj: Ocijeniti i komentirati kupljeni proizvod

- Sudionici: Baza podataka

- Preduvjet: Status narudžbe = „dostavljeno“

- Opis osnovnog tijeka:

1. Polaznik unosi ocjenu (1–5) i tekst
2. Sustav objavljuje recenziju

- Opis mogućih odstupanja:

1. Ocjena izvan raspona 1–5 → poruka „Nevažeća ocjena“
2. Tekst dulji od 1000 znakova → poruka „Recenzija je predugačka“

UC7 – Organiziraj izložbu

- Glavni sudionik: Organizator

- Cilj: Objaviti događaj za izlaganje i prodaju radova

- Sudionici: Baza podataka
- Preduvjet: ≥ 1 završena radionica
- Opis osnovnog tijeka:
 1. Otvara „Izložbe“ → „Nova izložba“
 2. Unosi naziv, lokaciju, datum, vrijeme (UC7.1)
 3. Uploada slike radova (UC7.2)
 4. Sustav objavljuje izložbu i omogućuje prijave (UC8)
- Opis mogućih odstupanja:
 1. Podaci nepotpuni → poruka „Sva polja su obavezna“

UC7.1 – Dodaj detalje izložbe

- Glavni sudionik: Organizator
- Cilj: Unos osnovnih podataka o izložbi
- Sudionici: Baza podataka
- Preduvjet: /
- Opis osnovnog tijeka:
 1. Unose se naziv, adresa, datum, vrijeme, opis, cijena ulaznice
 2. Sustav sprema podatke
- Opis mogućih odstupanja:
 1. Podaci nepotpuni → poruka „Sva polja su obavezna“

UC7.2 – Dodaj slike radova

- Glavni sudionik: Organizator
- Cilj: Uploadati fotografije radova koji će se izlagati
- Sudionici: Baza podataka
- Preduvjet: Izložba je kreirana
- Opis osnovnog tijeka:
 1. Multi-upload JPG/PNG uz opis i cijenu rada
 2. Putanja slike se sprema u bazu
- Opis mogućih odstupanja:
 1. Datoteka veća od 5 MB → poruka „Datoteka je prevelika“
 2. Format nije JPG/PNG → poruka „Nepodržani format slike“

UC8 – Prijavi se na izložbu

- Glavni sudionik: Polaznik
- Cilj: Prijaviti vlastiti rad ili posjetiti izložbu

- Sudionici: Baza podataka
- Preduvjet: Izložba objavljena i rok nije istekao
- Opis osnovnog tijeka:
 1. Polaznik bira izložbu → „Prijavi se“
 2. Ispunjava podatke (naslov rada, opis, upload slike)
 3. Prijava se čeka na odobrenje (UC8.1)
- Opis mogućih odstupanja:
 1. Rok istekao → poruka „Prijave više nisu moguće“

UC8.1 – Odobri prijavu

- Glavni sudionik: Organizator
- Cilj: Prihvati ili odbiti prijavljene radove
- Sudionici: Baza podataka
- Preduvjet: Postoje nove prijave
- Opis osnovnog tijeka:
 1. Organizator pregledava prijavu
 2. Odobrava/odbija s napomenom
 3. Ako odobri, rad se dodaje u izložbu
- Opis mogućih odstupanja:
 1. Prijava neispravna → poruka „Prijava odbijena zbog nedostataka“

UC9 – Objavi komentar/sliku nakon izložbe

- Glavni sudionik: Polaznik
- Cilj: Podijeliti dojmove i fotografije s događaja
- Sudionici: Baza podataka
- Preduvjet: Izložba završena i korisnik bio prihvaćen
- Opis osnovnog tijeka:
 1. Otvara stranicu izložbe
 2. Piše komentar (≤ 300 znakova) i/ili uploada sliku
 3. Sustav objavljuje odmah
- Opis mogućih odstupanja:
 1. Komentar dulji od 300 znakova → poruka „Komentar je predugačak“
 2. Slika veća od 5 MB → poruka „Datoteka je prevelika“

UC10 – Pretplati se na obavijesti

- Glavni sudionik: Polaznik

- Cilj: Primati e-mail obavijesti o novim radionicama/proizvodima
- Sudionici: Baza podataka
- Preduvjet: /
- Opis osnovnog tijeka:

1. Klik na „Prati”
2. Odabir kategorija (radionice, proizvodi, izložbe)
3. Sustav sprema pretplatu i šalje batch-mail 1× tjedno

- Opis mogućih odstupanja:

1. Pretplata već postoji → poruka „Već ste pretplaćeni na odabrane kategorije”

UC11 – Upravljanje članarinama

- Glavni sudionik: Administrator
- Cilj: Postaviti cijene mjesečne/godišnje članarine za organizatore
- Sudionici: Baza podataka
- Preduvjet: /
- Opis osnovnog tijeka:

1. Unosi iznos, valutu, trajanje
2. Promjene se primjenjuju odmah na nove članove

- Opis mogućih odstupanja:

1. Nevažeći iznos ili valuta → poruka „Neispravan unos, pokušajte ponovo”

UC12 – Odobri novi profil

- Glavni sudionik: Administrator
- Cilj: Onemogućiti neželjene korisnike i osigurati kvalitetu sadržaja
- Sudionici: Baza podataka
- Preduvjet: Organizator se registrirao
- Opis osnovnog tijeka:

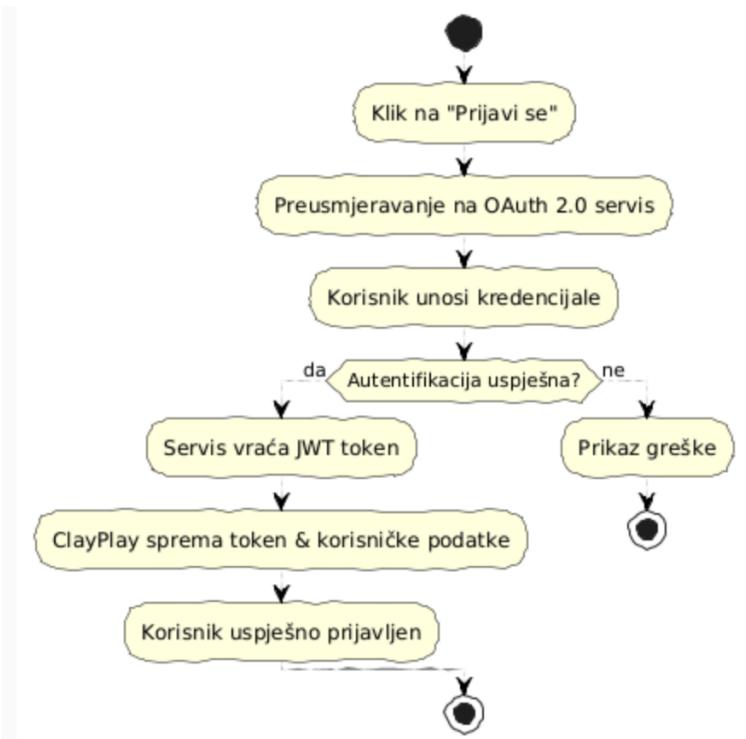
1. Pregledava podatke (broj mobitela, fotografije, e-mail...)
2. Odobrava/odbija s napomenom
3. Ako odobri, račun postaje aktivan

- Opis mogućih odstupanja:

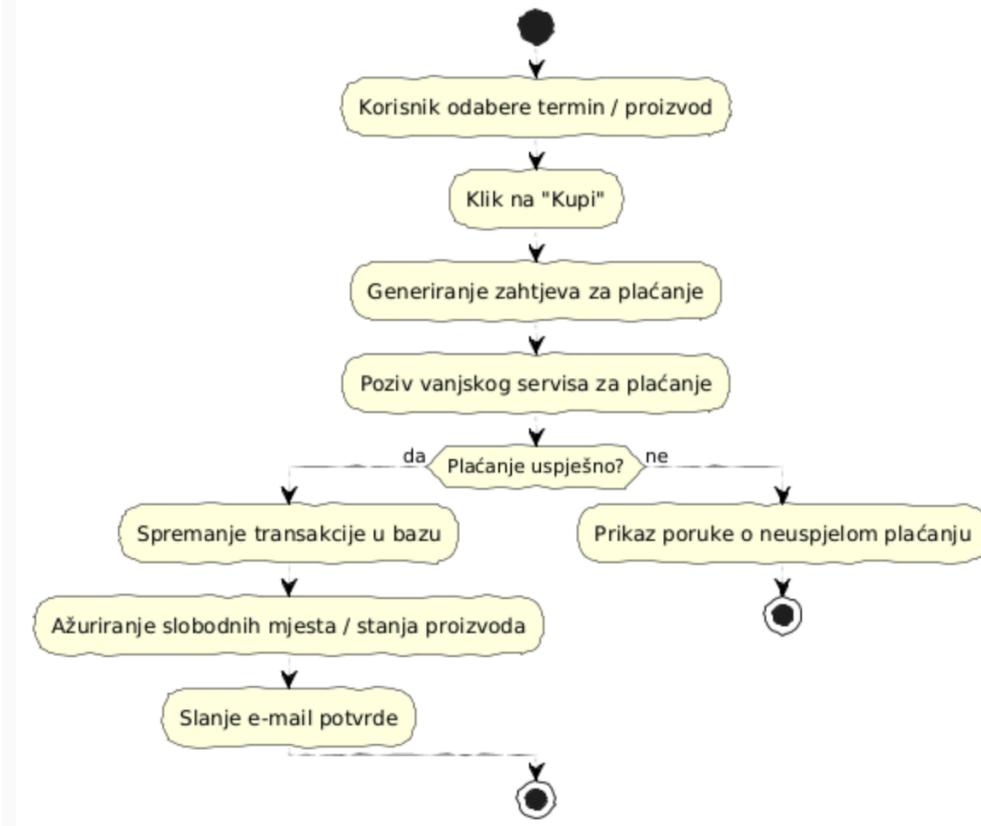
1. Podaci neispravni → poruka „Podaci u neispravnom formatu”

Sekvencijski dijagrami

Sekvencijski dijagram za prijavu korisnika:



Sekvencijski dijagram za plaćanje:



Provjera uključenosti ključnih funkcionalnosti u obrascu uporabe

Obrasci uporobe	Obuhvaćeni funkcionalni zahtjevi.
UC1 – Pregledaj radionice	F-005
UC2 – Rezerviraj termin	F-006
UC2.1 – Odaberij termin	F-006
UC2.2 – Plaćanje	F-006
UC2.3 – Primi potvrdu e-mailom	F-006
UC2.4 – Otkaži rezervaciju	F-007
UC3 – Upravlja radionicama	F-004, F-008
UC3.1 – Dodaj radionicu	F-004
UC3.2 – Dodaj slike radionice	F-008
UC3.3 – Uredi radionicu	F-004
UC3.4 – Obriši radionicu	F-004
UC4 – Postavi oglas za proizvod	F-009
UC5 – Kupi proizvod	F-009, F-010
UC5.1 – Plaćanje	F-017
UC6 – Ostavi recenziju	F-011
UC7 – Organiziraj izložbu	F-012
UC7.1 – Dodaj detalje izložbe	F-012
UC7.2 – Dodaj slike radova	F-008
UC8 – Prijavi se na izložbu	F-013
UC8.1 – Odobri prijavu	F-013
UC9 – Objavi komentar/sliku nakon izložbe	F-014
UC10 – Pretplati se na obavijesti	F-005, F-009
UC11 – Upravljanje članarinama	F-016
UC12 – Odobri novi profil	F-015

Stanari



▼ Pages 13

Find a page or section...

▶ [Home](#)

- ▶ [1. Opis projektnog zadatka](#)
- ▶ [2. Analiza zahtjeva](#)
- ▼ [3. Specifikacija zahtjeva sustava](#)

Obrasci uporabe

Dijagrami obrazaca uporabe

1. Visokorazinski dijagram obrazaca uporabe cijelog sustava
 2. Dijagram obrazaca uporabe za ključne značajke
- Dijagram obrasca uporabe za registraciju:
- Dijagram obrasca uporabe za prijavu:
3. Dijagram obrazaca uporabe za korisničke uloge
 4. Dijagram obrazaca uporabe za osnovne poslovne procese
 5. dijagram obrazaca uporabe za kritične sustave i integracije

Dijagram obrasca uporabe za plaćanje:

Dijagram obrasca uporabe za korištenje kalendara:

Opis obrazaca uporabe

UC1 – Pregledaj radionice

UC2 – Rezerviraj termin

UC2.1 – Odaberi termin

UC2.2 – Plaćanje

UC2.3 – Primi potvrdu e-mailom

UC2.4 – Otkaži rezervaciju

UC3 – Upravljam radionicama

UC3.1 – Dodaj radionicu

UC3.2 – Dodaj slike radionice

UC3.3 – Uredi radionicu

UC3.4 – Obriši radionicu

UC4 – Postavi oglas za proizvod

UC5 – Kupi proizvod

UC5.1 – Plaćanje

UC6 – Ostavi recenziju

UC7 – Organiziraj izložbu

UC7.1 – Dodaj detalje izložbe

UC7.2 – Dodaj slike radova

UC8 – Prijavi se na izložbu

UC8.1 – Odobri prijavu

UC9 – Objavi komentar/sliku nakon izložbe

UC10 – Preplati se na obavijesti

UC11 – Upravljanje članarinama

UC12 – Odobri novi profil

Sekvencijski dijagrami

Sekvencijski dijagram za prijavu korisnika:

Sekvencijski dijagram za plaćanje:

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](#)
- ▶ [B. Dnevnik promjena dokumentacije](#)
- ▶ [C. Prikaz aktivnosti grupe](#)

+ Add a custom sidebar

Clone this wiki locally

<https://github.com/Pavle-Stanarevic/Stanari.wiki.git>





Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

4. Arhitektura i dizajn sustava

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

Ivan Josip Kardum edited this page 8 hours ago · [14 revisions](#)

Arhitektura sustava

Opis arhitekture

Stil arhitekture:

Sustav je razvijen kao klijent–poslužitelj web aplikacija s odvojenim frontendom i backendom koji komuniciraju putem REST API-ja. Arhitektura je višeslojna, s jasnom podjelom na prezentacijski, aplikacijski i podatkovni sloj. Ovakav pristup omogućuje jednostavnije održavanje, modularnost i mogućnost neovisnog razvoja pojedinih dijelova sustava.

Podsustavi

Frontend (React + Vite)

- rukuje prikazom korisničkog sučelja
- poziva REST API putem Axios-a
- prikazuje karte i lokacije (Leaflet)

Backend (Spring Boot)

- autentifikacija i autorizacija (Spring Security + Google OAuth)
- poslovna logika za korisnike, radionice, izložbe, trgovinu i recenzije
- komunikacija s bazom preko Spring Data JPA
- upravljanje datotekama (slike)

Sustav pohrane

- PostgreSQL relacijska baza podataka
- datotečni sustav za pohranu slika (URL-ovi)

Preslikavanje na radnu platformu

- lokalno okruženje: Docker Compose (PostgreSQL + backend + frontend)
- produkcija: Render (backend, frontend i PostgreSQL baza)
- odabранo zbog jednostavnosti deploymenta manjeg projekta

Spremišta podataka

Relacijska baza PostgreSQL

- koristi se zbog jasnih odnosa među entitetima (korisnici, radionice, rezervacije...)

Datotečni sustav

- pohranjuje slike radova i galerija, u bazi se čuvaju URL-ovi

Mrežni protokoli

- HTTP/HTTPS za komunikaciju s backendom
- REST API za dohvati i izmjenu podataka
- OAuth 2.0 za Google prijavu

Globalni upravljački tok

- korisnik koristi frontend u pregledniku
- frontend šalje zahtjeve na backend (login, rezervacije, izložbe...)
- backend obrađuje zahtjev, pristupa bazi i datotekama
- backend vraća odgovor u JSON formatu
- frontend prikazuje ažurirane podatke korisniku

Sklopovskoprogramski zahtjevi

- server: 2–4 GB RAM, 2 CPU jezgre, 30 GB disk, Linux ili Windows; Docker i Docker Compose
- baza: PostgreSQL (lokalno preko Docker-a ili u cloudu - Render)
- frontend/backend build alati: Node.js, Maven
- klijent: moderan web preglednik

Obrazloženje odabira arhitekture

Odabrana arhitektura temelji se na modelu klijent–poslužitelj s jasnim razdvajanjem frontend i backend sloja. Takav pristup izabran je zbog jednostavnosti održavanja, modularnosti i mogućnosti neovisnog razvoja.

Razlozi odabira arhitektonskog stila

- frontend (React)** upravlja korisničkim sučeljem i komunikacijom s backendom kroz REST API
- backend (Spring Boot)** obrađuje poslovnu logiku, autentifikaciju, autorizaciju i rad s bazom
- REST API** omogućuje malu ovisnost između slojeva i lakše buduće nadogradnje (npr. mobilna aplikacija)

Ovaj stil omogućuje:

- brži razvoj i jednostavnije testiranje
- visoku modularnost
- neovisne nadogradnje frontend i backend dijela
- skaliranje samo onih dijelova sustava koji to zahtijevaju

Primjenjeni principi dizajna

- visoka kohezija - svaki sloj sustava ima jasno definiranu ulogu
- niska povezanost - frontend i backend komuniciraju isključivo kroz REST API
- modularnost - backend strukturiran u kontrolere, servise i repozitorije
- sigurnost - Spring Security i Google OAuth za pouzdanu autentifikaciju
- održivost - jednostavno proširivanje funkcionalnosti (radionice, izložbe, trgovina)

Razlozi za odabir PostgreSQL baze

- projekt uključuje mnogo povezanih entiteta (korisnici, radionice, rezervacije, proizvodi)
- podaci o slikama spremaju se u datotečni sustav, a URL-ovi u bazu zbog boljih performansi i manje potrošnje memorije

Razmatrane alternative i zašto su odbačene

Monolitna aplikacija

- veća složenost koda
- lošije korisničko iskustvo
- otežano održavanje

NoSQL baze

- slabija podrška za složene relacije i integritetska pravila

Organizacija sustava na visokoj razini

Klijent–poslužitelj

Aplikacija radi u klijent–poslužitelj modelu:

- klijent (frontend) pokreće se u web pregledniku i zadužen je za prikaz korisničkog sučelja te komunikaciju s poslužiteljem putem HTTP/HTTPS zahtjeva
- poslužitelj (backend) obrađuje zahtjeve, izvršava poslovnu logiku, komunicira s bazom podataka i upravlja autentifikacijom korisnika

Frontend i backend su potpuno odvojeni sustavi koji međusobno komuniciraju kroz REST API, što omogućuje neovisne nadogradnje i razvoj.

Baza podataka

Sustav koristi relacijsku bazu podataka PostgreSQL koja pohranjuje sve ključne informacije:

- korisnike (polaznike, organizatore, administratore)
- radionice i izložbe
- rezervacije i prijave
- proizvode i recenzije

- detalje o članarinama

Zbog velikog broja povezanih entiteta i pravila integriteta koristi se relacijski model. ORM sloj (Spring Data JPA) omogućuje jednostavnu komunikaciju između backenda i baze podataka.

Datotečni sustav

Uz bazu podataka koristi se i datotečni sustav poslužitelja:

- slike radova, profila i galerija spremaju se direktno na server
- u bazi se pohranjuje samo URL slike
- backend koristi servis (FileStorageService) za spremanje i dohvata datoteka

Ovaj pristup smanjuje opterećenje baze i omogućava brže učitavanje sadržaja.

Grafičko sučelje

Korisničko sučelje implementirano je kao web aplikacija izrađena u Reactu i renderira se u pregledniku. Ono omogućava:

- pregled radionica, izložbi i proizvoda
- prijave i rezervacije
- administraciju sadržaja
- navigaciju putem React Routera
- prikaz lokacija pomoću Leafleta

Frontend koristi Axios za dohvata podataka s backend API-ja, te prikazuje informacije ovisno o dobivenim odgovorima poslužitelja.

Organizacija aplikacije

Frontend (React)

- prikazuje korisničko sučelje i omogućuje interakciju korisnika sa sustavom
- koristi React Router za navigaciju između stranica
- šalje HTTP/HTTPS zahtjeve prema backendu putem Axios biblioteke
- dohvaćene podatke prikazuje u obliku komponenti
- obrađuje lokalne validacije podataka i prilagodbu prikaza

Backend (Spring Boot)

- implementira poslovnu logiku sustava
- provodi autentifikaciju i autorizaciju (Spring Security + Google OAuth)
- obrađuje podatke, upravlja rezervacijama, izložbama, proizvodima, korisnicima i recenzijama
- koristi Spring Data JPA za komunikaciju s PostgreSQL bazom
- upravlja datotekama (spremanje i dohvat slika)

Frontend i backend povezani su preko REST API-ja, što omogućuje neovisno razvijanje i nadogradnju svakog sloja.

MVC arhitektura

Backend je organiziran prema MVC (Model–View–Controller) principu kroz ove komponente:

Model

- predstavlja entitete iz baze
- definiran pomoću JPA anotacija
- održava podatkovne strukture i pravila relacija

Controller

- prima HTTP zahtjeve klijenta
- poziva odgovarajuće servise
- vraća JSON odgovore frontendu
- ne sadrži poslovnu logiku

Service

- sadrži poslovnu logiku aplikacije
- validira podatke, provodi izračune, upravlja rezervacijama i prijavama
- zove repozitorije za pristup podacima

Repository

- služi kao sloj za komunikaciju s bazom
- koristi Spring Data JPA
- omogućuje CRUD operacije bez pisanja SQL-a

Ovakva podjela omogućuje:

- visoku koheziju (svaki sloj radi samo svoj posao)
- nisku povezanost (kontroleri ne ovise o bazama, modeli ne ovise o sučelju)
- lakše održavanje i proširivanje sustava

Dijagram visoke razine

Baza podataka

Opis tablica

FOTOGRAFIJA

Atribut	Tip podatka	Opis varijable
fotold	BIGINT	PK. Jedinstveni identifikator fotografije.
fotoURL	TEXT	URL fotografije na serveru. (NOT NULL, UNIQUE)

CLANARINA

Atribut	Tip podatka	Opis varijable
idClanarina	BIGINT	PK. Identifikator članarine.
iznosEUR	NUMERIC(10,2)	Iznos članarine. (NOT NULL)
tipClanarine	VARCHAR(50)	Vrsta članarine. (NOT NULL)

KORISNIK

Atribut	Tip podatka	Opis varijable
idKorisnik	BIGINT	PK. Identifikator korisnika.
email	CITEXT	Email korisnika. (NOT NULL, UNIQUE)
password	VARCHAR(100)	Lozinka (hash). (NOT NULL)
ime	VARCHAR(50)	Ime korisnika.
prezime	VARCHAR(50)	Prezime korisnika.
adresa	VARCHAR(100)	Adresa korisnika.
brojTelefona	VARCHAR(15)	Broj telefona. (UNIQUE)
fotold	BIGINT	FK → FOTOGRAFIJA.fotold. Profilna fotografija.

ORGANIZATOR

Atribut	Tip podatka	Opis varijable
idKorisnik	BIGINT	PK, FK → KORISNIK.idKorisnik. Identifikator organizatora.
imeStudija	VARCHAR(50)	Naziv studija organizatora.

POLAZNIK

Atribut	Tip podatka	Opis varijable
idKorisnik	BIGINT	PK, FK → KORISNIK.idKorisnik. Identifikator polaznika.
zeliObavijesti	BOOLEAN	Preferira li primati obavijesti. (NOT NULL)

ADMINISTRATOR

Atribut	Tip podatka	Opis varijable
idKorisnik	BIGINT	PK, FK → KORISNIK.idKorisnik. Identifikator administratora.

RADIONICA

Atribut	Tip podatka	Opis varijable
idRadionica	BIGINT	PK. Identifikator radionice.
nazivRadionica	VARCHAR(50)	Naziv radionice. (NOT NULL)
opisRadionica	VARCHAR(1000)	Opis radionice. (NOT NULL)
trajanje	INTERVAL	Trajanje radionice. (NOT NULL)
datVrRadionica	TIMESTAMPTZ	Datum i vrijeme. (NOT NULL)
lokacijaRadionica	VARCHAR(100)	Lokacija. (NOT NULL)
brSlobMjesta	INT	Broj slobodnih mjesta. (NOT NULL)
cijenaRadionica	NUMERIC(10,2)	Cijena radionice. (NOT NULL)
idKorisnik	BIGINT	FK → ORGANIZATOR.idKorisnik. Organizator. (NOT NULL)

PROIZVOD

Atribut	Tip podatka	Opis varijable
proizvodId	BIGINT	PK. Identifikator proizvoda.
opisProizvod	VARCHAR(1000)	Opis proizvoda.
cijenaProizvod	NUMERIC(10,2)	Cijena proizvoda. (NOT NULL)
kategorijaProizvod	VARCHAR(50)	Kategorija. (NOT NULL)
idKorisnik	BIGINT	FK → ORGANIZATOR.idKorisnik. (NOT NULL)

RECENZIJA

Atribut	Tip podatka	Opis varijable
idRecenzija	BIGINT	PK. Identifikator recenzije.
ocjena	INT	Ocjena 1–5. (NOT NULL)
textRecenzija	VARCHAR(1000)	Tekst recenzije.
proizvodId	BIGINT	FK → PROIZVOD.proizvodId. (NOT NULL)
idKorisnik	BIGINT	FK → POLAZNIK.idKorisnik. (NOT NULL)

IZLOZBA

Atribut	Tip podatka	Opis varijable
idIzlozba	BIGINT	PK. Identifikator izložbe.
nazivIzlozba	VARCHAR(50)	Naziv izložbe. (NOT NULL)
lokacijaIzlozba	VARCHAR(100)	Lokacija. (NOT NULL)
datVriIzlozba	TIMESTAMPTZ	Datum i vrijeme. (NOT NULL)
idKorisnik	BIGINT	FK → ORGANIZATOR.idKorisnik. (NOT NULL)

PRIJAVA

Atribut	Tip podatka	Opis varijable
idPrijava	BIGINT	PK. Identifikator prijave.
statusIzlozba	TEXT	Status: pending/accepted/rejected. (NOT NULL)
idKorisnik	BIGINT	FK → POLAZNIK.idKorisnik. (NOT NULL)
idIzlozba	BIGINT	FK → IZLOZBA.idIzlozba. (NOT NULL)

REZERVACIJA

Atribut	Tip podatka	Opis varijable
idRezervacija	BIGINT	PK. Identifikator rezervacije.
idKorisnik	BIGINT	FK → POLAZNIK.idKorisnik. (NOT NULL)
idRadionica	BIGINT	FK → RADIONICA.idRadionica. (NOT NULL)
statusRez	TEXT	Status: reserved/canceled. (NOT NULL)

fotoRad

Atribut	Tip podatka	Opis varijable
fotold	BIGINT	PK, FK → FOTOGRAFIJA.fotold.
idRadionica	BIGINT	PK, FK → RADIONICA.idRadionica.

izlozeni

Atribut	Tip podatka	Opis varijable
fotold	BIGINT	PK, FK → FOTOGRAFIJA.fotold.
idIzlozba	BIGINT	PK, FK → IZLOZBA.idIzlozba.

placa

Atribut	Tip podatka	Opis varijable
idPlacanje	BIGINT	PK. Identifikator plaćanja.
datVrPocetakClanarine	TIMESTAMPTZ	Početak članarine. (NOT NULL)
datVrKrajClanarine	TIMESTAMPTZ	Kraj članarine.
idKorisnik	BIGINT	FK → ORGANIZATOR.idKorisnik. (NOT NULL)
idClanarina	BIGINT	FK → CLANARINA.idClanarina. (NOT NULL)

fotoProizvod

Atribut	Tip podatka	Opis varijable
proizvodId	BIGINT	PK, FK → PROIZVOD.proizvodId.
fotoid	BIGINT	PK, FK → FOTOGRAFIJA.fotoid.

KOMENTAR

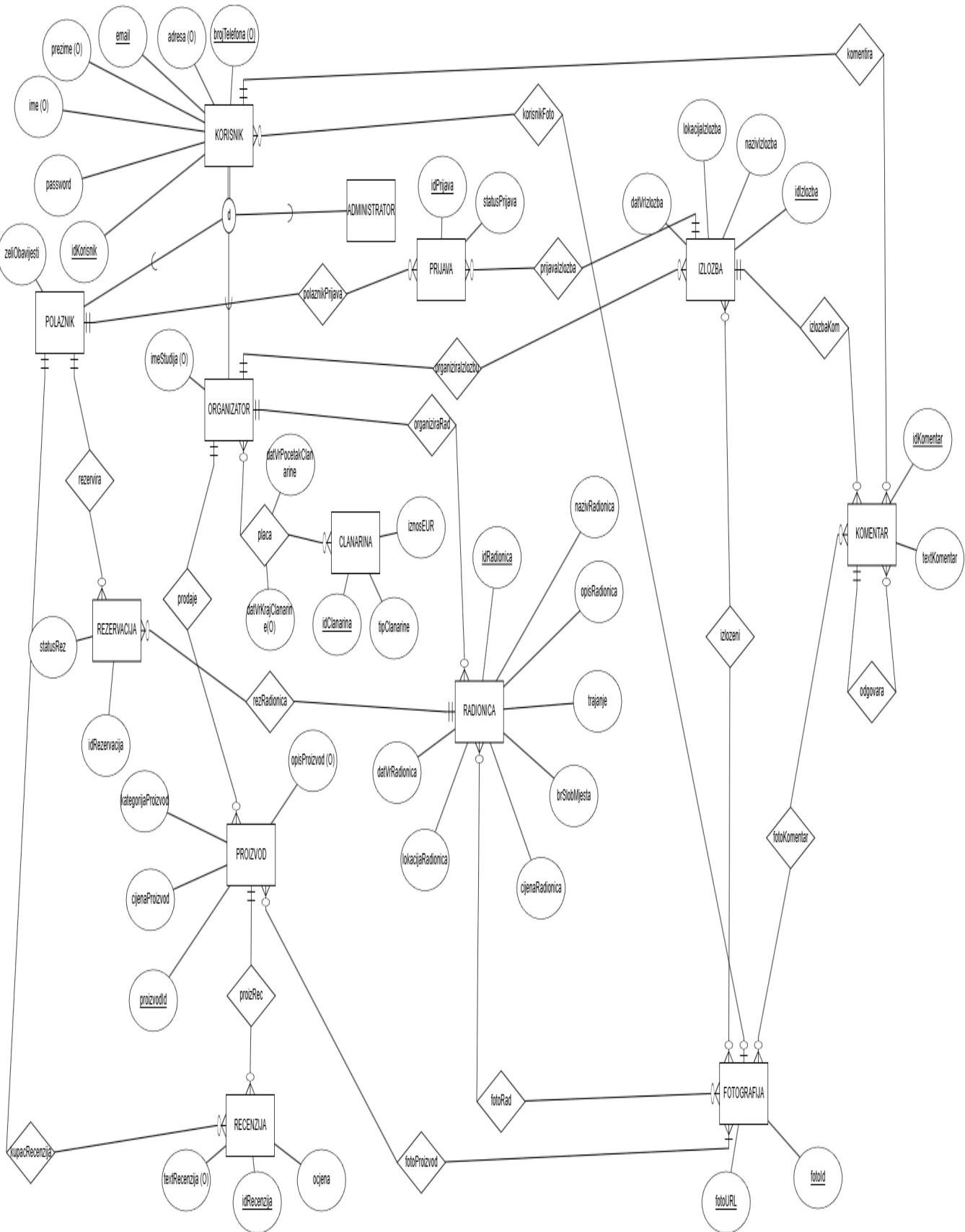
Atribut	Tip podatka	Opis varijable
idKomentar	BIGINT	PK. Identifikator komentara.
textKomentar	VARCHAR(1000)	Tekst komentara. (NOT NULL)
idlzlozba	BIGINT	FK → IZLOZBA.idLzlozba. (NOT NULL)
idKorisnik	BIGINT	FK → KORISNIK.idKorisnik. (NOT NULL)
odgovara_idKomentar	BIGINT	FK → KOMENTAR.idKomentar.

fotoKomentar

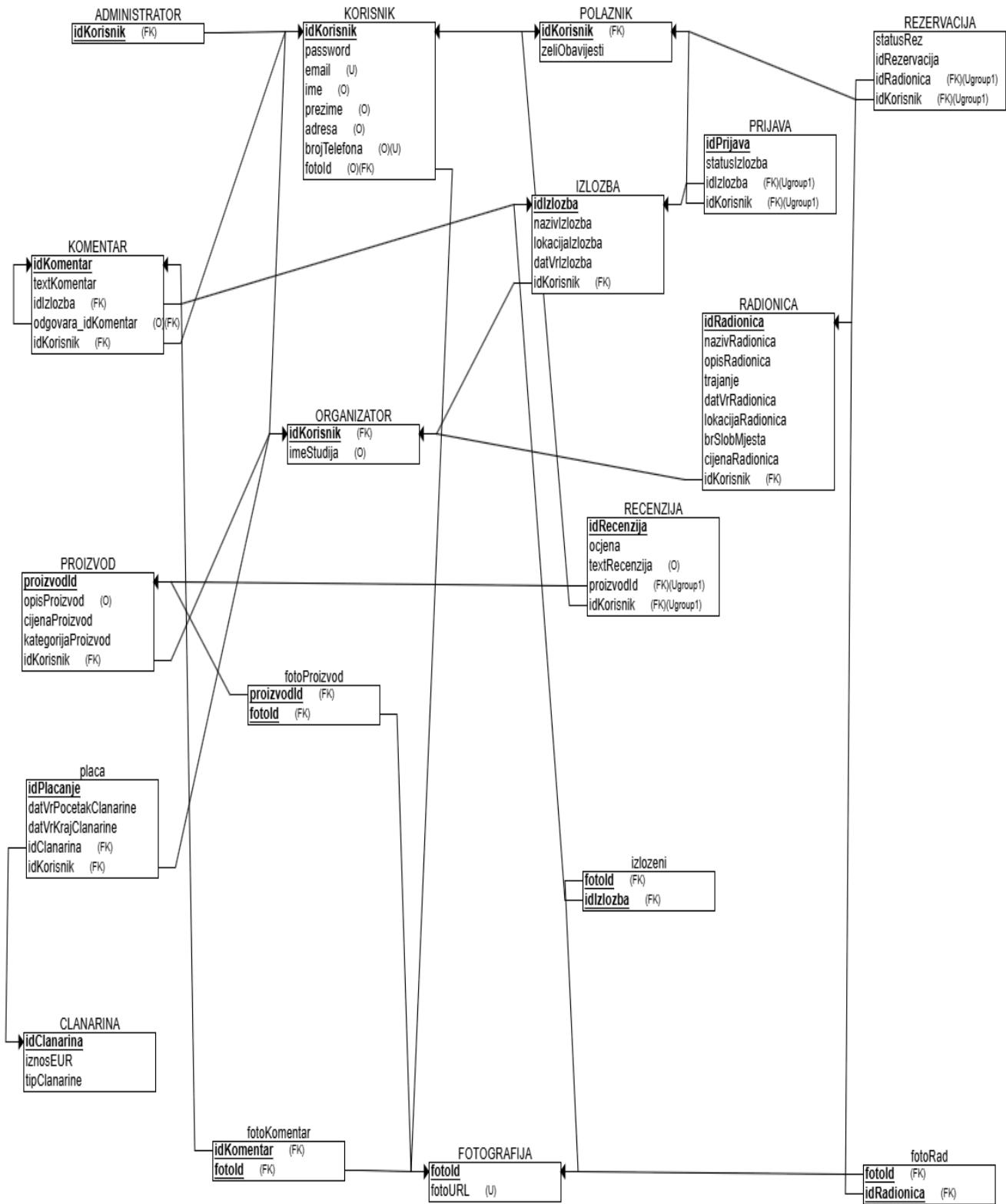
Atribut	Tip podatka	Opis varijable
idKomentar	BIGINT	PK, FK → KOMENTAR.idKomentar.
fotoid	BIGINT	PK, FK → FOTOGRAFIJA.fotoid.

Dijagram baze podataka

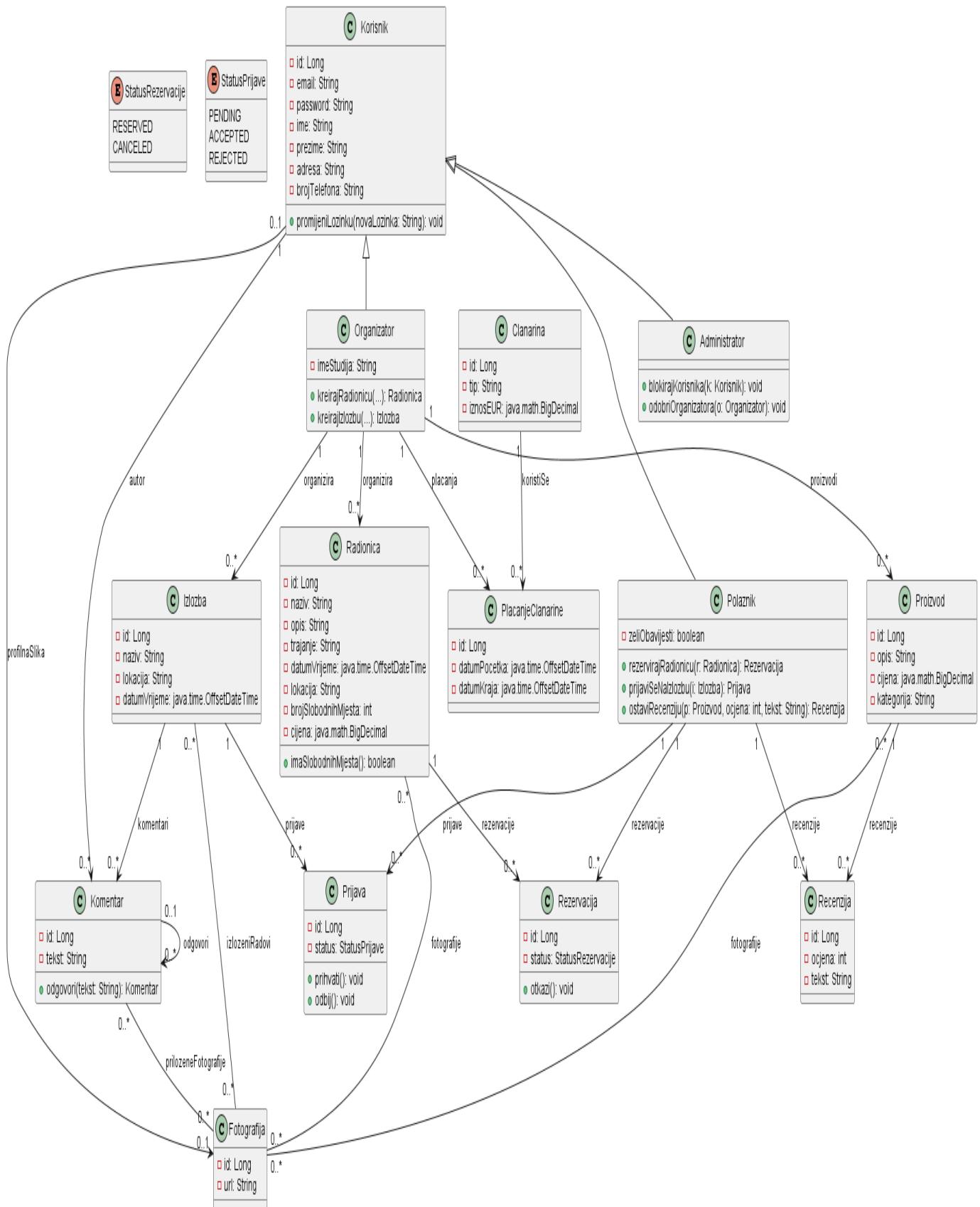
ER dijagram



Relacijska shema baze

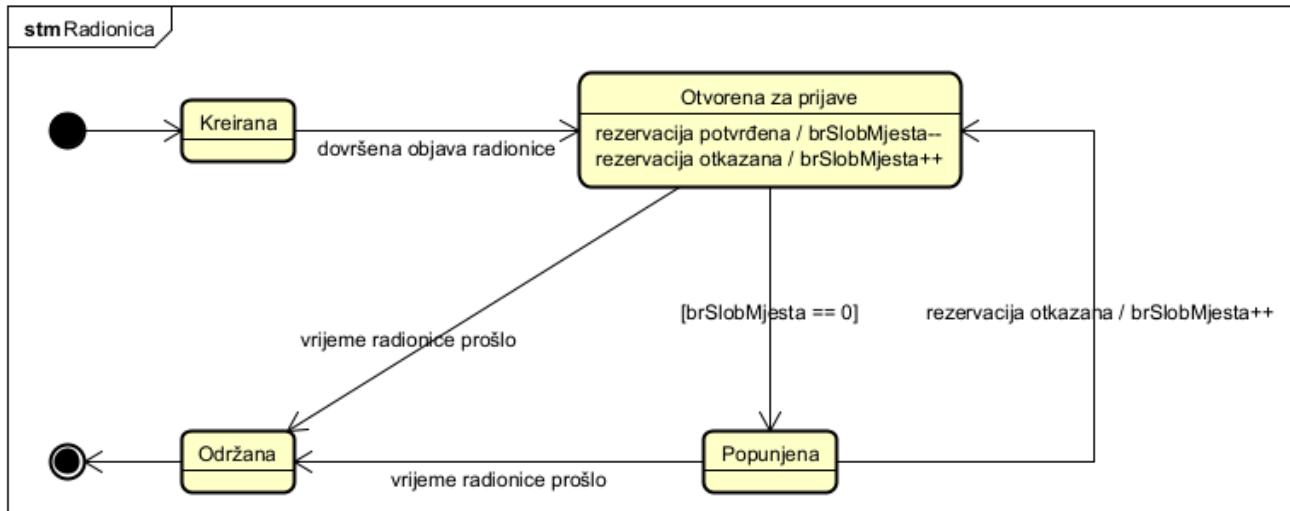


Dijagram razreda



Dinamičko ponašanje aplikacije

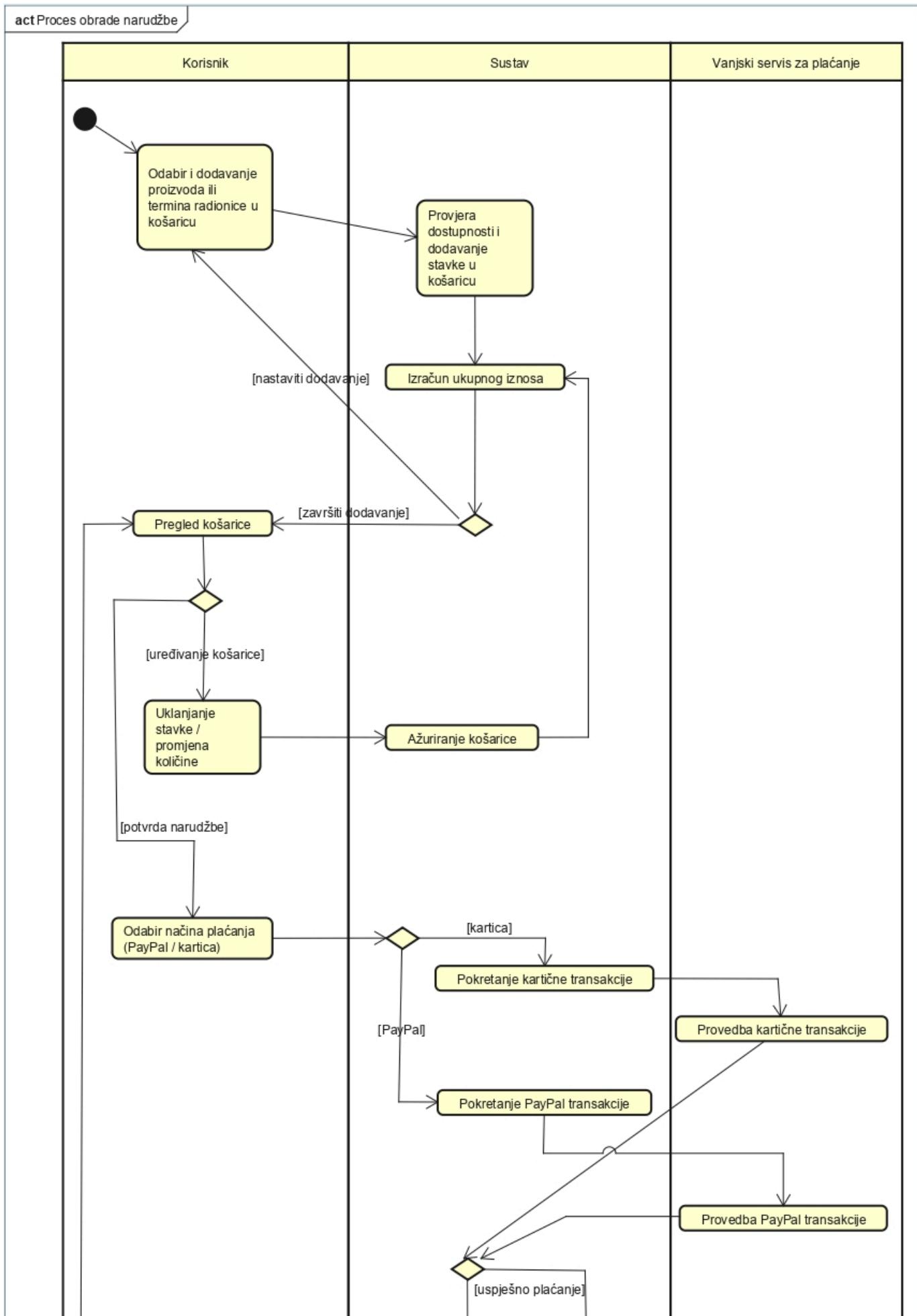
Dijagram stanja - Radionica

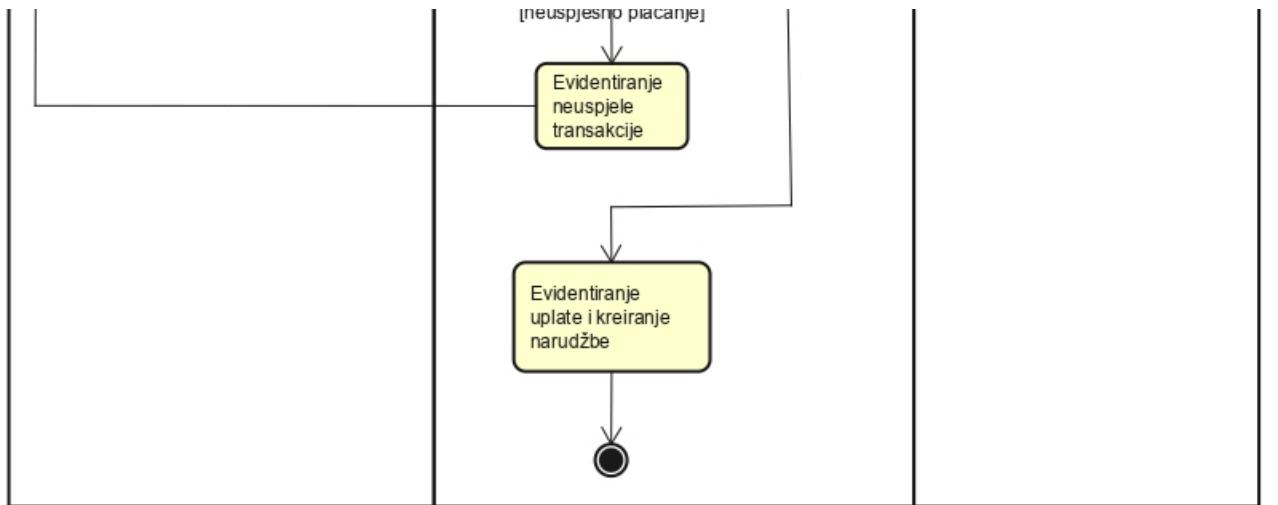


Dijagram stanja prikazuje životni ciklus objekta Radionica, od trenutka kreiranja do održavanja radionice. Nakon što organizator objavi radionicu, ona prelazi u stanje "Otvorena za prijave", u kojem je omogućena registracija polaznika sve dok postoje slobodna mjesta. U slučaju da se sva mjesta popune, radionica prelazi u stanje "Popunjena", a eventualnim otkazivanjem rezervacije može se ponovno vratiti u stanje otvoreno za prijave.

Bez obzira na stanje popunjenoosti, po isteku predviđenog datuma i vremena radionice objekt prelazi u stanje "Održana". Dijagram se fokusira isključivo na promjene stanja objekta Radionica, dok se pravila vezana uz rezervacije i otkazivanje (npr. pravilo 48 sati) primjenjuju nad objektom Rezervacija i nisu dio ovog dijagrama.

Dijagram aktivnosti - Proces obrade narudžbe





Dijagram aktivnosti prikazuje tijek obrade objekta Narudžba, koji predstavlja objedinjeni proces kupnje proizvoda iz trgovine i rezervacije termina radionica. Proces započinje dodavanjem proizvoda ili termina radionice u košaricu, nakon čega sustav provjerava dostupnost stavki te izračunava ukupni iznos narudžbe. Korisniku je omogućeno višekratno dodavanje i uređivanje sadržaja košarice prije pokretanja naplate.

Nakon pregleda košarice korisnik može pokrenuti postupak naplate, koji je dopušten samo ako narudžba sadrži barem jednu stavku. Sustav tada inicira transakciju putem vanjskog servisa za plaćanje. Dijagram obuhvaća ishode uspješne i neuspješne naplate: u slučaju neuspjeha transakcija se evidentira kao neuspješna te se narudžba vraća u stanje pregleda košarice, dok se u slučaju uspješne naplate evidentira uplata i kreira završna narudžba.

Dijagram se ne bavi prikazom korisničkog sučelja, već promjenama i tijekom obrade domenskog objekta Narudžba te interakcijom između korisnika, sustava i vanjskog servisa za plaćanje.

Stanari ✎

▾ Pages 13 ✎

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:

Podsustavi

Preslikavanje na radnu platformu

Spremišta podataka

Mrežni protokoli

Globalni upravljački tok
Sklopoškoprogramske zahtjevi
Obrazloženje odabira arhitekture
Razlozi odabira arhitektonskog stila
Primjenjeni principi dizajna
Razlozi za odabir PostgreSQL baze
Razmatrane alternative i zašto su odbačene
Organizacija sustava na visokoj razini
Klijent–poslužitelj
Baza podataka
Datotečni sustav
Grafičko sučelje
Organizacija aplikacije
Frontend (React)
Backend (Spring Boot)
MVC arhitektura
Dijagram visoke razine
Baza podataka
Opis tablica
FOTOGRAFIJA
CLANARINA
KORISNIK
ORGANIZATOR
POLAZNIK
ADMINISTRATOR
RADIONICA
PROIZVOD
RECENZIJA
IZLOZBA
PRIJAVA
REZERVACIJA
fotoRad
izlozeni
placa
fotoProizvod
KOMENTAR
fotoKomentar
Dijagram baze podataka
Dijagram razreda
Dinamičko ponašanje aplikacije
Dijagram stanja - Radionica
Dijagram aktivnosti - Proces obrade narudžbe

▶ [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](#)
- ▶ [B. Dnevnik promjena dokumentacije](#)
- ▶ [C. Prikaz aktivnosti grupe](#)

+ Add a custom sidebar

Clone this wiki locally

<https://github.com/Pavle-Stanarevic/Stanari.wiki.git>





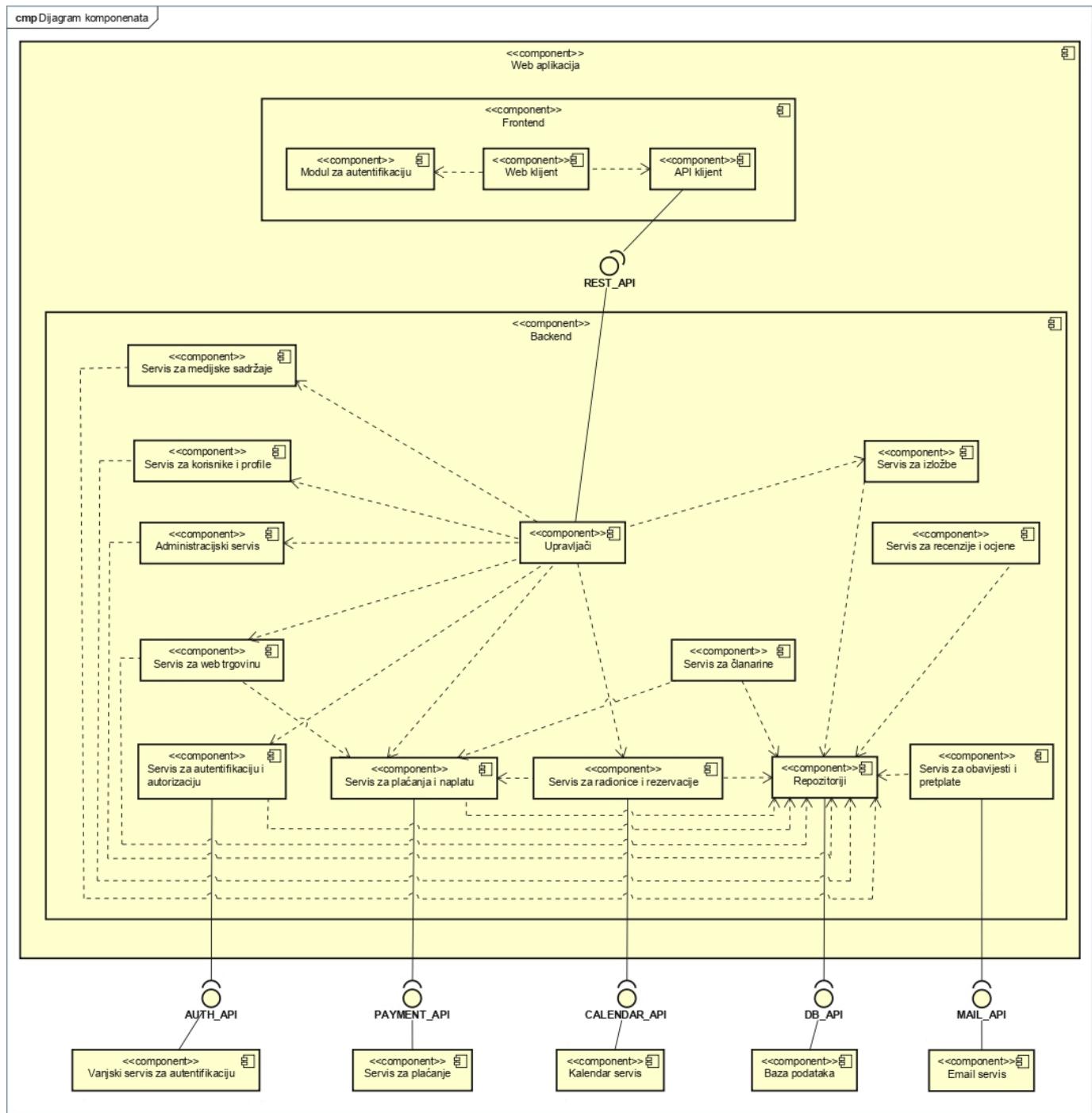
5. Arhitektura komponenata i razmještaja

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

[Jump to bottom](#)

Ivan Josip Kardum edited this page 7 hours ago · [7 revisions](#)

Dijagram komponenata

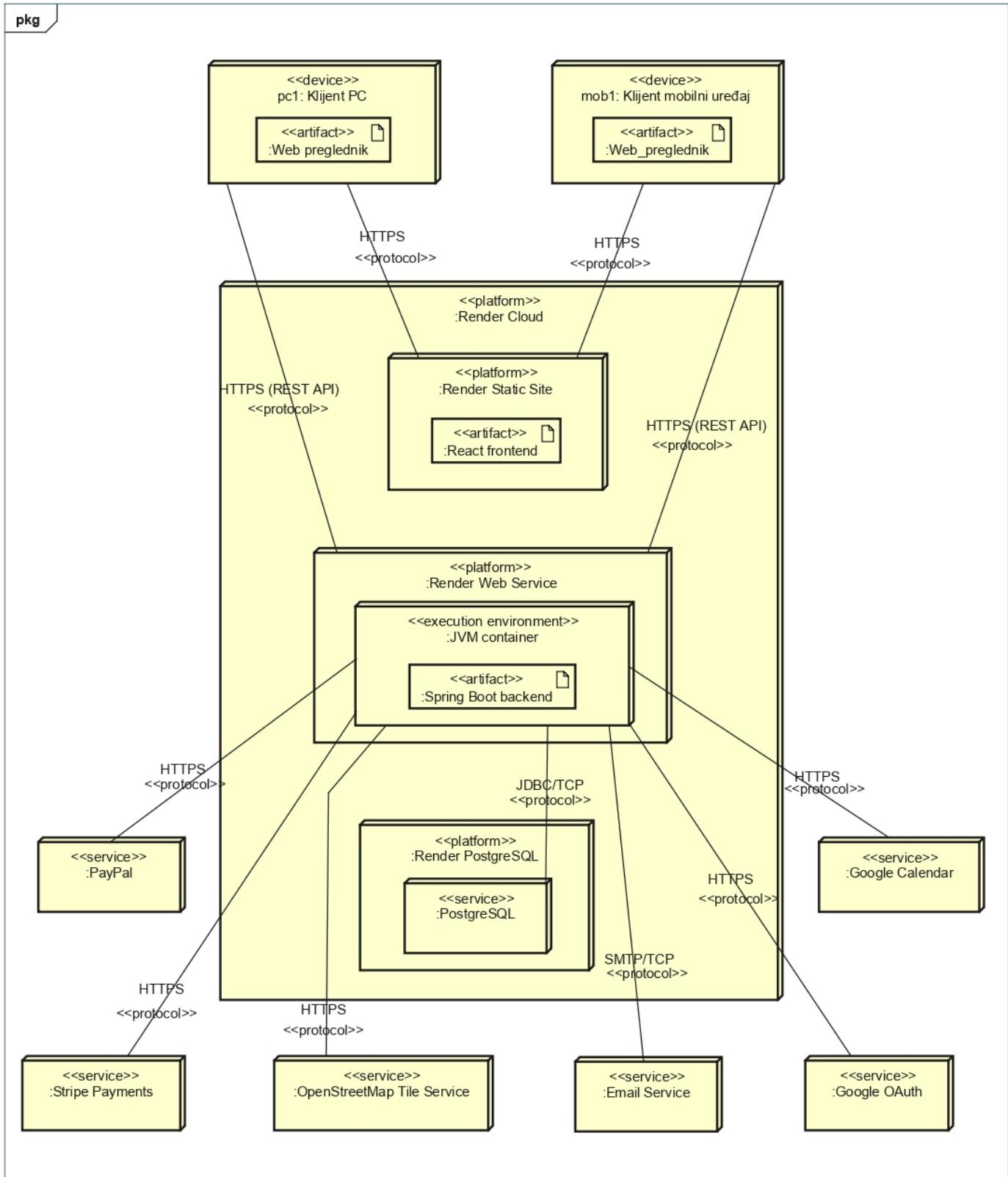


Dijagram komponenata prikazuje logičku strukturu sustava web aplikacije za organizaciju keramičkih radionica, izložbi i online trgovine. Dijagram razdvaja klijentsku i poslužiteljsku stranu aplikacije te prikazuje njihove međusobne odnose, kao i integraciju s vanjskim servisima.

Aplikacija je organizirana unutar komponente Web aplikacija, koja se sastoji od Frontend i Backend dijela. Frontend sadrži web klijent, modul za autentifikaciju i API klijent te komunicira s poslužiteljskom stranom putem sučelja REST_API. Backend je realiziran kao skup funkcionalno odvojenih servisa (npr. autentifikacija i autorizacija, radionice i rezervacije, web trgovina, plaćanja, izložbe, administracija), kojima upravljaju Upravljači (REST kontroleri).

Sustav koristi više vanjskih servisa, uključujući vanjski servis za autentifikaciju, servis za plaćanje, kalendarski servis te email servis, s kojima komunicira putem jasno definiranih sučelja (AUTH_API, PAYMENT_API, CALENDAR_API, MAIL_API). Pristup bazi podataka ostvaren je preko komponente Repozitoriji, koja služi kao sloj pristupa podacima i komunicira s bazom podataka putem sučelja DB_API. Dijagram prikazuje samo ključne ovisnosti između komponenata kako bi se osigurala čitljivost i jasnoća arhitekture sustava.

Dijagram razmještaja



Dijagram razmještaja prikazuje producijski razmještaj web aplikacije za keramičke radionice, izložbe i online trgovinu u okruženju oblaka. Dijagram je izrađen kao dijagram razmještaja instanci te prikazuje konkretnе čvorove, servise i uređaje koji sudjeluju u izvođenju sustava, kao i njihove međusobne komunikacijske veze.

Korisnici pristupaju aplikaciji putem web-preglednika na osobnom računalu ili mobilnom uređaju. Frontend aplikacija implementirana u Reactu raspoređena je kao statički sadržaj na servisu Render Static Site, dok je backend aplikacija implementirana u Spring Boot okviru i izvršava se kao aplikacijski servis na platformi Render Web Service unutar JVM okruženja. Podaci sustava pohranjuju se u upravljanu bazu podataka Render PostgreSQL, kojoj backend pristupa putem JDBC protokola.

Komunikacija između klijentskog uređaja i frontenda te između klijenta i backend API-ja odvija se putem HTTPS protokola. Backend aplikacija integrirana je s vanjskim servisima, uključujući servis za autentifikaciju (Google OAuth), servise za plaćanje (Stripe i PayPal), kalendarski servis (Google Calendar), kartografski servis (OpenStreetMap Tile Service) te email servis, s kojima komunicira putem HTTPS protokola, odnosno SMTP/TCP protokola za slanje elektroničke pošte. Za potrebe obrade plaćanja koristi se webhook mehanizam kojim vanjski servisi za plaćanje obavještavaju backend aplikaciju o promjenama statusa transakcija.

Dijagram razdvaja klijentsku, aplikacijsku i podatkovnu razinu sustava te prikazuje način na koji su komponente fizički raspoređene u proizvodnjoskom okruženju, u skladu s planiranom arhitekturom sustava.

Stanari

Pages 13

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](#)
- ▶ [B. Dnevnik promjena dokumentacije](#)
- ▶ [C. Prikaz aktivnosti grupe](#)

+ Add a custom sidebar

Clone this wiki locally

<https://github.com/Pavle-Stanarevic/Stanari.wiki.git>





Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

6. Ispitivanje programskog rješenja

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

Ivan Josip Kardum edited this page 32 minutes ago · [12 revisions](#)

Ispitivanje komponenti

IS-01 — CheckoutStore: spremanje i dohvata payload-a (redovni slučaj)

Funkcionalnost koju testiram:

Spremanje checkout payload-a u `CheckoutStore.save()` i dohvata istog payload-a preko `CheckoutStore.get(id)`.

Ispitni slučaj:

- Ulazni podaci:

```
payload = {"amount": 120, "currency": "EUR", "userId": 5}
```

- Očekivani rezultat:

- `save(payload)` vraća ne-null ID
 - `get(id)` vraća isti Map payload

- Dobiveni rezultat:

PROLAZ

Postupak provođenja ispitivanja:

1. Pokrenuti test (čime se pokreću i svi ostali testovi):

```
cd Stanari/backend
```

```
mvn test
```

2. Provjeriti da test `CheckoutStoreSaveAndGetTest` prolazi.

IS-02 — CheckoutStore: dohvata i brisanja s null ID (rubni uvjet)

Funkcionalnost koju testiram:

Otpornost komponente `CheckoutStore` na rubne ulazne vrijednosti (null identifikator).

Ispitni slučaj:

- Ulazni podaci:

```
id = null
```

- **Očekivani rezultat:**

- `get(null)` vraća `null`
- `remove(null)` vraća `null`
- ne dolazi do iznimke

- **Dobiveni rezultat:**

PROLAZ

Postupak provođenja ispitivanja:

1. Pokrenuti test (čime se pokreću i svi ostali testovi):

```
cd Stanari/backend
```

```
mvn test
```

2. Provjeriti da test `CheckoutStoreNullIdTest` prolazi.

IS-03 — UserService: registracija polaznika (redovni slučaj)

Funkcionalnost koju testiram:

Registracija korisnika tipa *polaznik* putem metode `UserService.register()`.

Ispitni slučaj:

- **Ulazni podaci:**

Valjani `RegistrationRequest`:

- ime i prezime
- email: `ana.ivic@example.com`
- kontakt broj
- lozinka: `Test123`
- potvrda lozinke: `Test123`
- userType: `polaznik`

- **Očekivani rezultat:**

- metoda vraća `Korisnik` objekt
- poziva se `korisnikRepository.save()`
- poziva se `polaznikRepository.save()`
- ne poziva se `organizatorRepository.save()`

- **Dobiveni rezultat:**

PROLAZ

Postupak provođenja ispitivanja:

1. Pokrenuti test (čime se pokreću i svi ostali testovi):

```
cd Stanari/backend
```

```
mvn test
```

2. Provjeriti da test `UserServiceRegisterPolaznikTest` prolazi.

IS-04 — UserService: registracija s postojećim emailom (izazivanje pogreške)

Funkcionalnost koju testiram:

Validacija jedinstvenosti email adrese prilikom registracije korisnika.

Ispitni slučaj:

- **Ulazni podaci:**

`RegistrationRequest` s emailom koji već postoji u sustavu.

- **Očekivani rezultat:**

- baca se `IllegalArgumentException`
- poruka: "Email already registered"
- spremanje korisnika se ne izvršava

- **Dobiveni rezultat:**

PROLAZ

Postupak provođenja ispitivanja:

1. Pokrenuti test (čime se pokreću i svi ostali testovi):

```
cd Stanari/backend
```

```
mvn test
```

2. Provjeriti da test `UserServiceRegisterDuplicateEmailTest` prolazi.

IS-05 — UserService: neispravna lozinka (rubni uvjet + izazivanje pogreške)

Funkcionalnost koju testiram:

Validacija lozinke prilikom registracije korisnika.

Ispitni slučaj:

- **Ulazni podaci:**

Lozinka bez velikog slova: `test123`

- **Očekivani rezultat:**

- baca se `IllegalArgumentException`
- poruka: "Password must contain at least one uppercase letter"

- **Dobiveni rezultat:**

PROLAZ

Postupak provođenja ispitivanja:

1. Pokrenuti test (čime se pokreću i svi ostali testovi):

```
cd Stanari/backend
```

```
mvn test
```

2. Provjeriti da test `UserServiceRegisterInvalidPasswordTest` prolazi.

IS-06 — FileStorageService: spremanje prazne datoteke (izazivanje pogreške)

Funkcionalnost koju testiram:

Validacija ulaznih podataka pri spremanju datoteke.

Ispitni slučaj:

- Ulazni podaci:

`byte[0]` (prazna datoteka)

- Očekivani rezultat:

- baca se `IllegalArgumentException`
 - poruka: "Empty file"

- Dobiveni rezultat:

PROLAZ

Postupak provođenja ispitivanja:

1. Pokrenuti test (čime se pokreću i svi ostali testovi):

```
cd Stanari/backend
```

```
mvn test
```

2. Provjeriti da test `FileStorageServiceEmptyFileTest` prolazi.

IS-07 — Poziv nepostojećeg endpointa (poziv neimplementirane funkcionalnosti)

Funkcionalnost koju testiram:

Reakcija sustava na poziv neimplementirane (nepostojeće) funkcionalnosti.

Ispitni slučaj:

- Ulazni podaci:

HTTP GET zahtjev na `/api/does-not-exist`

- Očekivani rezultat:

HTTP status 404 Not Found

- Dobiveni rezultat:

PROLAZ

Postupak provođenja ispitivanja:

1. Pokrenuti test (čime se pokreću i svi ostali testovi):

```
cd Stanari/backend
```

```
mvn test
```

2. Provjeriti da test `NonExistingEndpointTest` prolazi.

Ispitivanje sustava

Cilj ispitivanja sustava

Cilj ispitivanja sustava je provjera ponašanja cijelog sustava u uvjetima stvarnog korištenja, s naglaskom na međusobnu povezanost funkcionalnih komponenti i interakciju korisnika s aplikacijom ClayPlay. Ispitivanje je provedeno automatiziranim putem korištenjem alata Selenium IDE.

Ispitni slučajevi fokusirani su na specifične funkcionalnosti aplikacije ClayPlay, poput prijave na radionice, rada s košaricom, validacije korisničkih unosa te ograničenja pristupa administracijskim funkcijama.

Alat i okruženje

- **Alat:** Selenium IDE (Firefox)
- **Okruženje:** lokalno pokretanje aplikacije
(`http://localhost:5173`)
- **Zajednički preuvjeti:**
 - postoji registrirani korisnik u ulozi *Polaznik*
 - postoji barem jedna aktivna radionica dostupna za prijavu (npr. *Modeliranje vaza*)

TC01 – Prijava korisnika (redovni slučaj)

Ulazi:

- Email: `ivankar@gmail.com`
- Lozinka: `Ab1234`

Koraci ispitivanja:

1. Otvoriti aplikaciju u pregledniku.
2. Kliknuti na opciju „Prijavi se“.
3. Unijeti korisnički email i lozinku.
4. Kliknuti gumb za potvrdu prijave.
5. Provjeriti da je korisnik uspješno prijavljen.

Očekivani izlaz:

- Korisnik je prijavljen u sustav.
- U zaglavlju aplikacije vidljiv je gumb „Logout“.

Dobiveni izlaz:

- Test je uspješno prošao (PASS).

Dokaz ispitivanja:

Project: SystemTests-ClayPlay*



Tests	+	Search tests...	REC																					
<table border="1"> <thead> <tr> <th>Command</th><th>Target</th><th>Value</th></tr> </thead> <tbody> <tr> <td>1 ✓ open</td><td>http://localhost:5173/</td><td></td></tr> <tr> <td>2 ✓ click</td><td>linkText=Prijavi se</td><td></td></tr> <tr> <td>3 ✓ type</td><td>css=input[type="email"]</td><td>ivankar@gmail.com</td></tr> <tr> <td>4 ✓ type</td><td>css=input[type="password"]</td><td>Ab1234</td></tr> <tr> <td>Untitled</td><td>5 ✓ click</td><td>css=button[type="submit"]</td></tr> <tr> <td></td><td>6 ✓ assert element present</td><td>xpath=//button[normalize-space()="Logout"]</td></tr> </tbody> </table>				Command	Target	Value	1 ✓ open	http://localhost:5173/		2 ✓ click	linkText=Prijavi se		3 ✓ type	css=input[type="email"]	ivankar@gmail.com	4 ✓ type	css=input[type="password"]	Ab1234	Untitled	5 ✓ click	css=button[type="submit"]		6 ✓ assert element present	xpath=//button[normalize-space()="Logout"]
Command	Target	Value																						
1 ✓ open	http://localhost:5173/																							
2 ✓ click	linkText=Prijavi se																							
3 ✓ type	css=input[type="email"]	ivankar@gmail.com																						
4 ✓ type	css=input[type="password"]	Ab1234																						
Untitled	5 ✓ click	css=button[type="submit"]																						
	6 ✓ assert element present	xpath=//button[normalize-space()="Logout"]																						

The screenshot shows a web browser window with the title "ClayPlay". The address bar displays "localhost:5173". The page content is the "Radionice izrade keramike" section of the website. It features a blue header with the ClayPlay logo, navigation links (Radionice, Shop, Izložbe, Naš tim, Košarica, Ivan, Logout), and a "Finish setup" button. The main content area has a light blue background and contains text about pottery classes and a "Pregled radionica →" button. Below this are two columns: "Online shop" and "Izložbe radova", each with their own text and a "Odi u shop →" or "Activate Windows" button.

Radionice izrade keramike

U svijetu koji juri, gлина nas uči strpljenju. Na radionicama keramike zastajemo, dišemo i dopuštamo rukama da prate svoju priču. Svaki zajednici, žlica ili figura postaje odraz trenutka, nesavršenog savršenstva, baš kao i mi. Dođi i otkrij koliko je lijepo kad se kreativnost pretvori u oblik.

Pregled radionica →

Online shop

Za sve one koji vole keramiku, ali više uživaju u ispijanju kave nego u miješanju gline, otvorili smo naš online shop! Tam možeš pronaći unikatne keramičke komade koje su izradili naši instruktori i talentirani umjetnici.

Odi u shop →

Izložbe radova

Naše izložbe radova polaznika dokaz su da se gлина ne radi samo žlicama i žlicama - nego i ponosom, strpljenjem i čistom radošću stvaranja. Svaki izloženi komad priča svoju malu priču o prvim pokušajima, neočekivanim oblečenjima, pobojama koje su postale čari, i trenucima kad se ruke i mašta savršeno razumiju.

Activate Windows
Go to Settings to activate Windows.

TC02 – Dodavanje radionice u košaricu (redovni slučaj)

Ulazi:

- Odabrana radionica: *Modeliranje vaza*

Koraci ispitivanja:

- Otvoriti aplikaciju.
- Kliknuti na gumb "Radionice".
- Odabratи radionicu i izvršiti prijavu na radionicu.
- Otvoriti košaricu.
- Provjeriti sadržaj košarice.

Očekivani izlaz:

- Odabrana radionica dodana je u košaricu.
- U košarici je prikazan naziv radionice i cijena.

Dobiveni izlaz:

- Test je uspješno prošao (PASS).

Dokaz ispitivanja:

Project: SystemTests-ClayPlay^



Executing	Command	Target	Value
Executing	✓ open	http://localhost:5173/	
	✓ click	linkText=Radionice	
	✓ click	css=.new-workshop-btn	
	✓ click	css=.wd-primary	
	✓ click	css=.sign-btn:nth-child(1)	
	✓ assert element present	xpath=//*[contains(normalize-space(), 'Modeliranje vaza')]	

The screenshot shows a web browser window with the URL `localhost:5173/kosarica`. The page is titled "Košarica". At the top right, there is a notification badge with the number "1" over the "Košarica" button. Below the header, there are links for "Radionice", "Shop", "Izložbe", "Naš tim", "Logout", and "Ivan". On the left, there is a logo for "CLAY PLAY" featuring a stylized vase icon.

The main content area displays the shopping cart items:

- Radionice**: One item listed: "Modeliranje vaza" at 15.00€, with a "Ukloni" (Remove) button.
- Proizvodi**: No items listed.

At the bottom, it shows "Ukupno 15.00€" and a "Idi na plaćanje" (Go to payment) button.

At the very bottom right of the page, there is a watermark: "Activate Windows Go to Settings to activate Windows."

TC03 – Prazna košarica (rubni uvjet)

Ulazi:

- Košarica bez stavki

Koraci ispitivanja:

1. Otvoriti aplikaciju.
2. Otvoriti košaricu.
3. Provjeriti dostupnost opcije za plaćanje.

Očekivani izlaz:

- Kada je košarica prazna, gumb „Idi na plaćanje“ nije prikidan.
- Korisniku nije omogućeno nastaviti proces naplate.

Dobiveni izlaz:

- Test je uspješno prošao (PASS).

Dokaz ispitivanja:

Project: SystemTests-ClayPlay*

Command	Target	Value
1 ✓ open	http://localhost:5173/	
2 ✓ click	linkText=Košarica	
3 ✓ assert element not present	xpath=//button[contains(normalize-space(), 'Idi na plaćanje')]	

The screenshot shows a web browser window with the following details:

- Title Bar:** ClayPlay
- Address Bar:** localhost:5173/kosarica
- User Interface:** The header includes the ClayPlay logo, navigation links (Radionice, Shop, Izložbe, Naš tim), and a user menu (Košarica, Ivan, Logout). A "Finish setup" button is also present.
- Main Content:** The page title is "Košarica". It displays the message "Košarica je prazna." (The cart is empty.) and two buttons: "Pregled radionica" (View workshops) and "Shop".
- Bottom Right Corner:** A watermark or advertisement for Windows activation: "Activate Windows" and "Go to Settings to activate Windows."

TC04 – Neovlašten pristup administraciji (poziv nedozvoljene funkcionalnosti)

Ulazi:

- URL: /admin
- Uloga korisnika: *Polaznik*

Koraci ispitivanja:

1. Otvoriti aplikaciju.
2. U adresnu traku preglednika upisati /admin .
3. Pričekati završetak preusmjeravanja.
4. Provjeriti dostupnost administracijskih opcija.

Očekivani izlaz:

- Pristup administraciji je onemogućen.
- Korisnik je preusmjeren na početnu stranicu.
- Administracijske opcije nisu vidljive.

Dobiveni izlaz:

- Test je uspješno prošao (PASS).

Dokaz ispitivanja:

Project: SystemTests-ClayPlay*



Test	Command	Target	Value
✓ TC01_Login_Success*	✓ open	http://localhost:5173/	
✓ TC02_Add_Workshop_To	✓ open	http://localhost:5173/admin	
✓ TC03_Empty_Cart_Check	✓ wait for element visible	linkText=Radionice	5000
✓ TC04_Admin_Access_De	✓ assert element not present	xpath=//*[self::a or self::button][contains(normalize-space(), 'Admin')]	
Untitled			

The screenshot shows a web browser window with the URL `localhost:5173` in the address bar. The page content is as follows:

- Header:** ClayPlay logo, navigation icons (back, forward, search), and a "Finish setup" button.
- Top Navigation Bar:** Radionice, Shop, Izložbe, Naš tim, Košarica (with a shopping cart icon), Ivan, and Logout.
- Section 1 (Radionice izrade keramike):**
 - Section Header:** Radionice izrade keramike
 - Text:** U svijetu koji juri, gлина nas uči strpljenju. Na radionicama keramike zastajemo, dišemo i dopuštamo rukama da prate svoju priču. Svaki zajednici, žlica ili figura postaje odraz trenutka, nesavršenog savršenstva, baš kao i mi. Dođi i otkrij koliko je lijepo kad se kreativnost pretvori u oblik.
 - Button:** Pregled radionica →
- Section 2 (Online shop):**
 - Section Header:** Online shop
 - Text:** Za sve one koji vole keramiku, ali više uživaju u ispijanju kave nego u miješanju gline, otvorili smo naš online shop! Tam možeš pronaći unikatne keramičke komade koje su izradili naši instruktori i talentirani umjetnici.
 - Button:** Odi u shop →
- Section 3 (Izložbe radova):**
 - Section Header:** Izložbe radova
 - Text:** Naše izložbe radova polaznika dokaz su da se gлина ne radi samo žlicama i žlicama - nego i ponosom, strpljenjem i čistom radošću stvaranja. Svaki izloženi komad priča svoju malu priču o prvim pokušajima, neočekivanim oblečenjima, pobojama koje su postale čari, i trenucima kad se ruke i mašta savršeno razumiju.

On the right side of the page, there is a watermark for Windows activation: "Activate Windows Go to Settings to activate Windows."

TC05 – Registracija s nevažećom lozinkom (grešni/rubni uvjet)

Ulazi:

- Lozinka bez velikog slova (npr. 123456a)

Koraci ispitivanja:

1. Otvoriti aplikaciju.
2. Odabratи opciju registracije korisnika.
3. Unijeti sve potrebne podatke.
4. Unijeti nevažeću lozinku koja ne sadrži veliko slovo.
5. Pokušati potvrditi registraciju.

Očekivani izlaz:

- Registracija nije dopuštena.
- Sustav prikazuje poruku o grešci vezanu uz pravila lozinke.

Dobiveni izlaz:

- Test je uspješno prošao (PASS).

Dokaz ispitivanja:

Project: SystemTests-ClayPlay*



Tests	+	Search tests...	HTTP Request URL: http://localhost:5173/	Actions
		Command	Target	Value
✓ TC01_Login_Success*		1 ✓ open	http://localhost:5173/	
✓ TC02_Add_Workshop_To_Cart		2 ✓ click	linkText=Prijavi se	
✓ TC03_Empty_Cart_Check		3 ✓ click	linkText=Registiraj se	
✓ TC04_Admin_Access_Decline		4 ✓ type	name=firstName	Ivan
✓ TC05_Registration_Invalid_Password		5 ✓ type	name=lastName	Runjić
Untitled		6 ✓ type	name=address	Ozaljska 3
		7 ✓ type	name=contact	+386911234567
		8 ✓ type	name=email	ivanrunjic@gmail.com
		9 ✓ type	name=password	123456a
		10 ✓ type	name=confirmPassword	123456a



Project: SystemTests-ClayPlay*

Tests	+	▷	◁	⟳	∅	▼	REC
http://localhost:5173/							
		Command	Target	Value			
✓ TC01_Login_Success*		✓ type	name=lastName	Runjić			
✓ TC02_Add_Workshop_To_Cart		✓ type	name=address	Ozaljska 3			
✓ TC03_Empty_Cart_Check		✓ type	name=contact	+386911234567			
✓ TC04_Admin_Access_Decline		✓ type	name=email	ivanrunjic@gmail.com			
✓ TC05_Registration_Invalid		✓ type	name=password	123456a			
Untitled		✓ type	name=confirmPassword	123456a			
		✓ click	xpath=//button[contains(text(),'Registriraj se!')]				
		✓ assert element present	xpath=//*[contains(normalize-space(), 'barem jedno veliko slovo')]				

The screenshot shows a web browser window with the URL `localhost:5173/registerPolaznik`. The page title is "Polaznik". The main content is a "Kreiraj račun" (Create account) form. The form fields include:

- Prijava putem Googlea (Login via Google)
- Ivan Runjić (First name: Ivan, Last name: Runjić)
- Ozaljska 3 +386911234567 (Address: Ozaljska 3, Phone number: +386911234567)
- ivanrunjic@gmail.com (Email address: ivanrunjic@gmail.com)
- Two masked password fields (Redacted)
- A red error message: Lozinka mora sadržavati barem jedno veliko slovo. (Password must contain at least one uppercase letter.)
- A placeholder text: Priložite fotografiju ili logo (Upload a photograph or logo) with a dashed file input area.
- Activation watermark: Activate Windows Go to Settings to activate Windows.

Rezultati ispitivanja sustava

- Ukupan broj provedenih ispitnih slučajeva: 5
- Broj uspješno izvršenih ispitnih slučajeva: 5
- Broj otkrivenih grešaka tijekom ispitivanja sustava: 0

Automatiziranim ispitivanjem potvrđeno je da sustav ispravno reagira u redovnim, rubnim i grešnim scenarijima te da su ključne funkcionalnosti dostupne isključivo ovlaštenim korisnicima.

Stanari (edit)

▼ Pages 13

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

IS-01 — CheckoutStore: spremanje i dohvati payload-a (redovni slučaj)
IS-02 — CheckoutStore: dohvati i brisanje s null ID (rubni uvjet)
IS-03 — UserService: registracija polaznika (redovni slučaj)
IS-04 — UserService: registracija s postojećim emailom (izazivanje pogreške)
IS-05 — UserService: neispravna lozinka (rubni uvjet + izazivanje pogreške)
IS-06 — FileStorageService: spremanje prazne datoteke (izazivanje pogreške)
IS-07 — Poziv nepostojećeg endpointa (poziv neimplementirane funkcionalnosti)

Ispitivanje sustava

Cilj ispitivanja sustava

Alat i okruženje

TC01 – Prijava korisnika (redovni slučaj)
TC02 – Dodavanje radionice u košaricu (redovni slučaj)
TC03 – Prazna košarica (rubni uvjet)
TC04 – Neovlašten pristup administraciji (poziv nedozvoljene funkcionalnosti)
TC05 – Registracija s nevažećom lozinkom (grešni/rubni uvjet)

Rezultati ispitivanja sustava

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

- ▶ [8. Upute za puštanje u pogon](#)
- ▶ [9. Zaključak i budući rad](#)
- ▶ [A. Popis literature](#)
- ▶ [B. Dnevnik promjena dokumentacije](#)
- ▶ [C. Prikaz aktivnosti grupe](#)

+ Add a custom sidebar

Clone this wiki locally

<https://github.com/Pavle-Stanarevic/Stanari.wiki.git>





Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

7. Tehnologije za implementaciju aplikacije

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

Marko-Malenica edited this page 3 minutes ago · [9 revisions](#)

Korištene tehnologije i alati

Ovaj dio dokumentacije opisuje tehnologije korištene u razvoju aplikacije s ciljem jasnog razumijevanja tehnološke osnove projekta te olakšavanja budućeg održavanja, nadogradnje i timske suradnje. U nastavku su obuhvaćeni programski jezici, radni okviri, baza podataka, razvojni i deploy alati te cloud infrastruktura na kojoj je aplikacija postavljena.

Serverski dio aplikacije razvijen je u programskom jeziku Java (verzija 17). Odabrana verzija predstavlja dugoročno podržano izdanje koje pruža stabilnost, sigurnost i dobre performanse, što je posebno važno za aplikacije koje rade s trajnim podacima i većim brojem korisničkih zahtjeva. Java omogućuje jasan objektno-orientirani pristup i dobru čitljivost koda, što olakšava održavanje i daljnji razvoj aplikacije.

Za izradu backend dijela korišten je Spring Boot framework (verzija 3.5.6). Spring Boot pojednostavljuje razvoj web aplikacija kroz automatsku konfiguraciju i ugrađeni aplikacijski poslužitelj, čime se smanjuje količina potrebnog konfiguracijskog koda. Backend aplikacija izložena je kao REST API, a konfiguracija se definira putem datoteke `application.properties`. Osjetljive postavke, poput podataka za spajanje na bazu, nisu pohranjene u kodu već se učitavaju iz environment varijabli, što omogućuje sigurnije i fleksibilnije postavljanje aplikacije u različitim okruženjima.

Korisničko sučelje aplikacije razvijeno je korištenjem JavaScript jezika (ECMAScript 2020 ili noviji) uz React biblioteku (verzija 19.1.1). React je odabran zbog svog deklarativnog i komponentskog načina rada, koji omogućuje izgradnju složenog sučelja iz manjih, logički odvojenih cjelina. Takav pristup olakšava ponovnu upotrebu komponenti i prilagodbu korisničkog sučelja bez utjecaja na ostatak aplikacije. Za razvoj i izgradnju frontend dijela korišten je Vite (verzija 7.2.2), koji osigurava brzo pokretanje razvojnog okruženja i optimizirani build za produkcijsko izvođenje. Node.js (verzija 25.1.0) se koristi u razvoju frontend dijela aplikacije kao runtime okruženje za build alate i upravljanje JavaScript ovisnostima, dok backend aplikacija nije implementirana u Node.js okruženju. Povezivanje s backendom ostvaruje se putem HTTP zahtjeva, pri čemu se adresa API-ja konfigurira kroz environment varijablu.

Podaci aplikacije pohranjuju se u relacijskoj bazi podataka PostgreSQL (verzija 17). PostgreSQL je odabran zbog svoje pouzdanosti, dobre podrške za relacijske modele i mogućnosti rada u produkcijskim cloud okruženjima. Backend koristi Spring Data JPA (verzija 3.3) i Hibernate (verzija 6.4) za mapiranje objekata na relacijske tablice. Struktura baze podataka generira se automatski prilikom pokretanja aplikacije korištenjem Hibernate mehanizma za upravljanje shemom, bez korištenja dodatnih alata za migracije baze podataka.

Razvoj aplikacije proveden je uz pomoć razvojnog alata Visual Studio Code, koji omogućuje rad s Java i JavaScript projektima kroz dostupne dodatke, ugrađeno debuggiranje i integraciju s alatima za upravljanje verzijama. Za verzioniranje izvornog koda korišten je Git (verzija 2.39.3), dok GitHub služi kao centralno mjesto za pohranu repozitorija, suradnju članova tima i praćenje promjena u projektu.

Za testiranje korisničkog sučelja korišten je Selenium (verzija 4.11), koji omogućuje automatizirano testiranje web aplikacija kroz simulaciju stvarnih korisničkih interakcija u pregledniku. Selenium se koristi za provjeru ispravnosti rada ključnih funkcionalnosti aplikacije s krajne korisničke perspektive. Na taj način smanjuje se rizik od regresijskih grešaka prilikom nadogradnji i promjena u frontend dijelu aplikacije.

Za razmještaj aplikacije korišten je Docker (verzija 4.24.2), koji omogućuje jednostavno pakiranje aplikacije i njezinih ovisnosti u kontejnere. Tako postižemo konzistentno izvođenje aplikacije.

Aplikacija je postavljena u proizvodnji okruženje korištenjem cloud platforme Render. Backend aplikacija postavljena je kao Web Service, dok je frontend aplikacija postavljena kao Static Site. Render omogućuje automatski build i razmještaj aplikacije iz GitHub repozitorija te jednostavno upravljanje environment varijablama. PostgreSQL baza podataka također je hostana na Render platformi, čime je osigurana konzistentna i pouzdana infrastruktura za rad aplikacije.

Korištene tehnologije čine uravnoteženu i modernu tehnološku osnovu koja omogućuje stabilan rad aplikacije, jednostavno održavanje sustava i daljnje proširenje funkcionalnosti prema potrebama korisnika.

Stanari 

▼ Pages 13 

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
- ▶ [8. Upute za puštanje u pogon](#)
- ▶ [9. Zaključak i budući rad](#)
- ▶ [A. Popis literature](#)

▶ [B. Dnevnik promjena dokumentacije](#)

▶ [C. Prikaz aktivnosti grupe](#)

+ Add a custom sidebar

Clone this wiki locally

<https://github.com/Pavle-Stanarevic/Stanari.wiki.git>



[Code](#)[Issues](#)[Pull requests](#)[Actions](#)[Projects](#)[Wiki](#)[Security](#)[Insights](#)[Settings](#)

8. Upute za puštanje u pogon

[Edit](#)[New page](#)[Jump to bottom](#)Marko-Malenica edited this page 11 minutes ago · [29 revisions](#)

Ovaj odjeljak dokumentacije treba dati detaljne smjernice za instalaciju, konfiguraciju, pokretanje i administraciju aplikacije. Cilj je olakšati postavljanje aplikacije na razvojnom, ispitnom i produkcijskom okruženju.

1. Instalacija

- **Preduvjeti:**
 - Git
 - Java JDK 17
 - Maven 3.8+
 - Node.js 16+
 - Node Package Manager
 - GitHub račun
 - Render račun
 - Docker desktop aplikacija (Docker 4+)
- **Preuzimanje:** Izvorni kod aplikacije preuzima se kloniranjem Git repozitorija.

Primjer:

```
git clone https://github.com/Pavle-Stanarevic/Stanari.git
cd Stanari
```

Instalacija ovisnosti

Ovisnosti se instaliraju unutar Docker containera.

2. Postavke

- **Konfiguracijske datoteke:** Backend aplikacija koristi `application.properties`, dok frontend koristi environment varijable direktno. Osjetljivim podacima i environment varijablama upravljamo pomoću `.env` datoteke, koju ispunjavamo po uzoru na danu `.env.example` datoteku.

```
cp .env.example .env
```



Predložak prekopiramo te ga nakon ispunimo. Neke od environment varijabli su korištene u `application.properties`.

- Primjer varijabli u `application.properties` :

```
spring.datasource.url=${DATABASE_URL}
spring.datasource.username=${DB_USERNAME}
spring.datasource.password=${DB_PASSWORD}

spring.datasource.driver-class-name=org.postgresql.Driver
spring.jpa.hibernate.ddl-auto=update

server.port=${PORT:10000}

spring.servlet.multipart.max-file-size=5MB
spring.servlet.multipart.max-request-size=6MB
server.tomcat.max-swallow-size=6MB
```



Korištene varijable su automatski generirane pri postavljanju baze podataka putem Rendera, te ih trebamo ažurirati u .env datoteci kada generiramo iste.

- Primjer environment varijabli za frontend:

`VITE_API_URL=https://stanari-backend.onrender.com`

- **Postavke baze podataka:** Baza podataka postavlja se korištenjem Render PostgreSQL servisa. Render automatski generira potrebne podatke za spajanje na bazu podataka putem environment varijabli. Prijava na <https://render.com> putem GitHub računa je prepostavljena.

Kreiranje PostgreSQL baze

1. U Render dashboardu kliknuti na **New**
2. Odabratи opciju **PostgreSQL**
3. U obrascu za kreiranje baze podataka postaviti:
 - **Name:** stanari-db
 - **Database:** stanari
 - **User:** automatski generirano
 - **Plan:** Free
4. Kliknuti na **Create Database**

Dohvaćanje podataka za spajanje

Nakon uspješnog kreiranja baze:

1. Otvoriti PostgreSQL servis `stanari-db`
2. U sekciji **Connections / Environment** dostupne su sljedeće automatski generirane environment varijable:

`DATABASE_URL`

`DB_USERNAME`

`DB_PASSWORD`

`DB_HOST`

DB_PORT**DB_NAME**

Ove su ranije spomenute varijable koje se koriste za povezivanje baze s backend aplikacijom.

Povezivanje baze s backend servisom

1. U Render dashboardu otvoriti backend **Web Service**
2. Otvoriti karticu **Environment**
3. Dodati sljedeće environment varijable:
 - o **DATABASE_URL** – vrijednost iz PostgreSQL servisa
 - o **DB_USERNAME** – vrijednost iz PostgreSQL servisa
 - o **DB_PASSWORD** – vrijednost iz PostgreSQL servisa
4. Spremiti promjene

Render će automatski ponovno pokrenuti backend servis.

3. Pokretanje aplikacije

- **Lokalno:**

Nakon pokretanja aplikacije Docker Desktop, smješteni u folder Stanari u kojem je kloniran projekt u terminal upišite redom:

```
git checkout main  
git pull origin main  
docker compose up --build
```

- **Provjera rada:** Na svom web pregledniku pretražite adresu: localhost:5173.

4. Upute za administratore

- **Pristup administratorskom sučelju:**

- o Administratorske funkcionalnosti dostupne su kroz aplikaciju nakon prijave korisnika s administratorskom ulogom.
- o Poseban admin panel za sada nije implementiran.

- **Redovito održavanje:**

- o Praćenje stanja baze podataka putem Render dashboarda.
- o Pregled logova Spring Boot aplikacije unutar Render servisa.
- o Ažuriranje aplikacije vrši se automatski prilikom pushanja promjena u GitHub repozitorij.

Primjer lokalnog ažuriranja:

```
git pull origin main  
mvn clean package
```

- **Rješavanje problema:**

- Logovi su dostupni u Render dashboardu.
- Greške se analiziraju putem Spring Boot stack trace zapisa.

5. Primjer za Render platformu (Cloud Deploy)

- **Postavljanje baze podataka:**

- Kreirati PostgreSQL servis na Render platformi i spremiti generirane environment varijable kao što je opisano u sekciji označenoj brojem 2

- **Postavljanje backend servisa (Spring Boot):**

- Tip servisa: Web Service
- Root Directory: backend
- Environment: Java
- Build Command:

```
mvn clean package
```

- Start Command:

```
java -jar target/*.jar
```

- Environment varijable:

```
DATABASE_URL
```

```
DB_USERNAME
```

```
DB_PASSWORD
```

```
PORT
```

- **Postavljanje frontend servisa:**

- Tip servisa: Static Site
- Root Directory: frontend
- Build Command:

```
npm install && npm run build
```

- Publish Directory:

```
dist
```

- Environment varijable:

```
VITE_API_URL
```

- **Pokretanje aplikacije:**

Render će automatski preuzeti repozitorij, instalirati ovisnosti i pokrenuti aplikaciju. Nakon deploja, aplikaciji možete pristupiti putem generiranog URL-a <https://stanari-frontend.onrender.com>.

Opis prisutpa aplikaciji na javnom poslužitelju

Aplikaciji se pristupa putem internetskog preglednika koristeći javnu URL adresu <https://stanari-frontend.onrender.com>.

- Frontend aplikacija dostupna je svim korisnicima.
- Administratorske funkcionalnosti dostupne su samo korisnicima s administratorskom ulogom.

Ograničenja

- Backend servis na Render Free planu može imati kašnjenje pri prvom pokretanju, koje se mora provesti kada god nije bilo prometa više od 15 minuta.
- Baza podataka ima maksimalno 1 GB prostora.
- Po Render Free planu aplikacija može raditi do 750 sati mjesecno.

Pristup administratorskom sučelju

- Administrator se prijavljuje putem standardnog korisničkog sučelja.
- Administratorske funkcije su zaštićene autentifikacijom i autorizacijom.
- Administratorski račun 1:
 - E-pošta: admin1@progi.hr
 - Lozinka: admin1
- Administratorski račun 2:
 - E-pošta: admin2@progi.hr
 - Lozinka: admin2

Stanari ✎

▼ Pages 13

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 prisutpa aplikaciji na javnom poslužitelju

- ▶ [9. Zaključak i budući rad](#)
- ▶ [A. Popis literature](#)
- ▶ [B. Dnevnik promjena dokumentacije](#)
- ▶ [C. Prikaz aktivnosti grupe](#)

+ Add a custom sidebar

Clone this wiki locally

<https://github.com/Pavle-Stanarevic/Stanari.wiki.git>



[Code](#)[Issues](#)[Pull requests](#)[Actions](#)[Projects](#)[Wiki](#)[Security](#)[Insights](#)[Settings](#)

9. Zaključak i budući rad

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

Marko-Malenica edited this page 7 minutes ago · [5 revisions](#)

Izrada projektnog zadatka odvijala se u jasno definiranim fazama, započevši s analizom problema i definiranjem osnovnih funkcionalnih zahtjeva, zatim razradom arhitekture sustava i podjelom odgovornosti unutar projektnog tima, te naposljetku implementacijom backend i frontend dijela aplikacije. Vrijeme predviđeno za izradu projekta bilo je ograničeno, što je zahtijevalo fokusiranje na temeljne funkcionalnosti i donošenje odluka o tome koje će se značajke implementirati u početnoj verziji aplikacije, a koje će biti ostavljene za budući razvoj.

Tijekom rada na projektu identificirano je nekoliko tehničkih izazova. Jedan od ključnih izazova odnosio se na uspostavu pouzdane komunikacije između frontend i backend dijela aplikacije, uključujući pravilno definiranje REST API endpointa i obradu podataka. Poseban izazov predstavljalo je i modeliranje baze podataka na način koji omogućuje jednostavno proširenje sustava u budućnosti. Dodatno, izazov je bio i prvi susret nekih članova tima s razvojem aplikacije temeljene na Spring Boot okviru te rad s cloud platformom Render. Svi navedeni izazovi uspješno su riješeni kroz iterativni pristup razvoju, praćenje dostupne dokumentacije i međusobnu suradnju unutar tima.

Rad na projektu omogućio je stjecanje i produbljivanje znanja iz područja web razvoja, uključujući razvoj REST API-ja u Spring Boot okruženju, izradu korisničkog sučelja korištenjem React biblioteke te rad s relacijskom bazom podataka PostgreSQL. Osim tehničkih znanja, poseban naglasak bio je na razvoju vještina timskog rada, planiranja zadataka, verzioniranja izvornog koda te rješavanja problema kroz zajedničku analizu i komunikaciju. Također je stečeno praktično iskustvo u razmještaju aplikacije u produkcijsko okruženje korištenjem cloud platforme.

Za bržu i kvalitetniju realizaciju projekta u budućnosti bila bi korisna dodatna znanja iz područja testiranja softvera, posebno vezana uz jedinične i integracijske testove, kao i dublje razumijevanje sigurnosnih aspekata web aplikacija. Također, iskustvo u dizajnu korisničkog iskustva i optimizaciji performansi aplikacije doprinijelo bi kvalitetnijem konačnom rješenju.

Projekt ima potencijal za daljnji razvoj unutar projektne grupe. Perspektive nastavka rada uključuju proširenje postojećih funkcionalnosti, poboljšanje korisničkog sučelja, uvođenje naprednijih mehanizama upravljanja podacima te prilagodbu aplikacije potrebama većeg broja korisnika. Sustav je arhitekturno postavljen na način koji omogućuje postupno nadograđivanje bez većih izmjena temeljne strukture.

U ostvarenoj verziji aplikacije implementirane su sve željene funkcionalnosti.

[Stanari](#)[Pages 13](#)

- ▶ [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](#)
 - ▶ [B. Dnevnik promjena dokumentacije](#)
 - ▶ [C. Prikaz aktivnosti grupe](#)

+ Add a custom sidebar

Clone this wiki locally

<https://github.com/Pavle-Stanarevic/Stanari.wiki.git>





Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

A. Popis literature

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

Ivan Josip Kardum edited this page yesterday · [3 revisions](#)

1. Programsko inženjerstvo, FER ZEMRIS, <http://www.fer.hr/predmet/proinz>
2. Objektno orijentirano programiranje, FER, <https://www.fer.unizg.hr/predmet/oop>
3. Baze podataka, FER, <https://www.fer.unizg.hr/predmet/bazpod>
4. Razvoj programske potpore za web, FER, <https://www.fer.unizg.hr/predmet/web1>
5. The Unified Modeling Language, <https://www.uml-diagrams.org/>
6. Astah Community, <http://astah.net/editions/uml-new>
7. Figma, <https://www.figma.com/>
8. React documentation, <https://react.dev/learn>
9. Spring Boot documentation, <https://docs.spring.io/spring-boot>
10. PostgreSQL documentation, <https://www.postgresql.org/docs/>
11. Docker documentation, <https://docs.docker.com/>
12. Google OAuth 2.0 documentation, <https://oauth.net/2/>

Stanari

[Pages 13](#)

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](#)

▶ B. Dnevnik promjena dokumentacije

▶ C. Prikaz aktivnosti grupe

+ Add a custom sidebar

Clone this wiki locally

<https://github.com/Pavle-Stanarevic/Stanari.wiki.git>





B. Dnevnik promjena dokumentacije

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

Ivan Josip Kardum edited this page 40 minutes ago · [1 revision](#)

Rev.	Opis promjene/dodataka	Autori	Datum
0.1	Inicijalni commit - stvoreni README.md i LICENCE	Pavle Stanarević	10.10.2025
0.2	Stvorena inicijalna struktura projekta i kopiran predložak za Wiki od profesora Sruka	Bruno Komočar	19.10.2025
0.3	Dodana analiza zahtjeva	Toni Brajko & Bruno Komočar	23.10.2025
0.4	Ažurirana analiza zahtjeva	Ivan Josip Kardum & Marko Malenica	12.11.2025
0.5	Dodan opis projektnog zadatka	Ivan Josip Kardum	13.11.2025
0.6	Ažurirani funkcionalni zahtjevi	Pavle Stanarević	13.11.2025
0.7	Dodani podaci o arhitekturi i dizajnu sustava	Ivan Josip Kardum	13.11.2025
0.8	Dodane specifikacije zahtjeva sustava	Marko Malenica	13.11.2025
0.9	Ažurirani README.md i dodan CODE_OF_CONDUCT.md	Ivan Josip Kardum	14.11.2025
0.10	Dodane upute za puštanje u pogon	Marko Malenica	18.1.2026
0.11	Dodan dijagram stanja	Ivan Josip Kardum	19.1.2026
0.12	Dodan popis tehnologija za implementaciju aplikacije	Marko Malenica	19.1.2026
0.13	Dodan dijagram aktivnosti	Ivan Josip Kardum	20.1.2026
0.14	Dodan zaključak i plan budućeg rada	Marko Malenica	20.1.2026
0.15	Dodani podaci o arhitekturi komponenata i razmještaja	Ivan Josip Kardum	22.1.2026
0.16	Ažurirani dijagrami stanja, aktivnosti, komponenti i razmještaja	Ivan Josip Kardum	23.1.2026
0.17	Dodani podaci o ispitivanju komponenti	Marko Malenica	23.1.2026
0.18	Dodani podaci o ispitivanju sustava	Ivan Josip Kardum	23.1.2026
0.19	Ažurirane upute za puštanje u pogon	Marko Malenica	23.1.2026
1.0	Dokumentacija finalizirana	Ivan Josip Kardum	23.1.2026

Stanari



▼ Pages 13

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](#)
- B. Dnevnik promjena dokumentacije**
- ▶ [C. Prikaz aktivnosti grupe](#)

+ Add a custom sidebar

Clone this wiki locally<https://github.com/Pavle-Stanarevic/Stanari.wiki.git>



Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

C. Prikaz aktivnosti grupe

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

Bruno Komočar edited this page now · [4 revisions](#)

Dnevnik sastajanja

1. sastanak

- Datum: 10. listopada 2025.
- Prisustvovali: T.Brajko, I.J.Kardum, B.Komočar, M.Malenica, P.Stanarević, F.Mihalj
- Teme sastanka:
 - Napravljena WhatsApp grupa i smišljeno ime tima

2. sastanak

- Datum: 15. listopada 2025.
- Prisustvovali: T.Brajko, I.J.Kardum, B.Komočar, M.Malenica, P.Stanarević, F.Mihalj
- Teme sastanka:
 - Napravljen GitHub repozitorij za tim
 - Postavljen Discord server za komunikaciju članova tima
 - Dogovorena osnovna struktura projekta
 - Određene su uloge na projektu:
 - Frontend: P. Stanarević (voditelj tima), F. Mihalj
 - Backend: B.Komočar, T. Brajko
 - Baze i dokumentacija: I.J.Kardum, M. Malenica

3. sastanak

- Datum: 28. listopada 2025.
- Prisustvovali: T.Brajko, I.J.Kardum, B.Komočar, M.Malenica, P.Stanarević, F.Mihalj
- Teme sastanka:
 - Napravljen dizajn svih stranica u Figma
 - Dogovor o korištenim alatima u projektu
 - Spring Boot, React i PostgreSQL kao glavne tehnologije razvoja
 - Figma za dizajn korisničkog sučelja
 - Discord za komunikaciju tima

4. sastanak

- Datum: 5. studenoga 2025.
- Prisustvovali: T.Brajko, I.J.Kardum, B.Komočar, M.Malenica, P.Stanarević, F.Mihalj
- Teme sastanka:
 - Završen fronted landing page
 - poveznica na issue: [#2](#)
 - Završavanje implementacija stranica za prijavu i registraciju
 - Frontend login i register forme dovršene
 - Dovršavane baze podataka
 - poveznica na issue: [#3](#) [#5](#) [#8](#)
 - Backend kreće s implementacijom

5. sastanak

- Datum: 9. studenoga 2025.
- Prisustvovali: T.Brajko, I.J.Kardum, B.Komočar, M.Malenica, P.Stanarević, F.Mihalj
- Teme sastanka:
 - Izrađivanje fronted djela /profile stranice
 - poveznica na issue: [#10](#)
 - Izrađivanje frontend dijela stranice za popisa radionica i organiziranje radionica
 - poveznica na issue: [#14](#)
 - Dovršavanje backenda za funkcionalnosti prijave i registracije
 - Započinje razvoj nove funkcionalnosti: kreiranje radionica

6. sastanak

- Datum: 12. studenoga 2025.
- Prisustvovali: T.Brajko, I.J.Kardum, B.Komočar, M.Malenica, P.Stanarević, F.Mihalj
- Teme sastanka:
 - Potrebno je dovršiti implementaciju OAuth prijave
 - poveznica na issue: [#11](#)
 - Provjera funkcionalnosti baze podataka
 - poveznica na issue: [#18](#)
 - Popravljanje funkcionalnosti stranice
 - poveznica na issue: [#21](#)

- poveznica na issue: [#13](#)

7. sastanak

- Datum: 19. studenoga 2025.
- Prisustvovali: T.Brajko, I.J.Kardum, B.Komočar, M.Malenica, P.Stanarević, F.Mihalj
- Teme sastanka:

- Napraviti page o biranju plana plaćanja za organizatore

- poveznica na issue: [#26](#)

- Izrada shopa

- poveznica na issue: [#27](#)

- Page za naplaćivanje

- poveznica na issue: [#28](#)

8. sastanak

- Datum: 26. studenoga 2025.
- Prisustvovali: T.Brajko, I.J.Kardum, B.Komočar, M.Malenica, P.Stanarević, F.Mihalj
- Teme sastanka:

- Objasniti errore pri registraciji

- poveznica na issue: [#32](#)

- Promijeniti kategorije u shopu

- poveznica na issue: [#33](#)

- Osposobiti plaćanje

- poveznica na issue: [#35](#)

9. sastanak

- Datum: 3. prosinca 2025.
- Prisustvovali: T.Brajko, I.J.Kardum, B.Komočar, M.Malenica, P.Stanarević, F.Mihalj
- Teme sastanka:

- Izrada stranice tima organizatora

- poveznica na issue: [#38](#)

- Prikaz fotografija radionice

- poveznica na issue: [#40](#)

- Mijenjanje slike na profilu

- poveznica na issue: [#45](#)

10. sastanak

- Datum: 10. prosinca 2025.
- Prisustvovali: T.Brajko, I.J.Kardum, B.Komočar, M.Malenica, P.Stanarević, F.Mihalj
- Teme sastanka:

- Ne mogu nastaviti na plaćanje ako pretplata nije spremljena u bazu
 - poveznica na issue: [#49](#)
- PayPal Sandbox – backend endpointi za plaćanje
 - poveznica na issue: [#50](#)
- Backend podrška za plaćanje pretplate (Stripe + PayPal)
 - poveznica na issue: [#51](#)

11. sastanak

- Datum: 17. prosinca 2025.
- Prisustvovali: T.Brajko, I.J.Kardum, B.Komočar, M.Malenica, P.Stanarević, F.Mihalj
- Teme sastanka:

- Dodat mogućnost mijenjanja slike i šifre
 - poveznica na issue: [#55](#)
- Backend – dodati adresu, promjenu lozinke i profilnu sliku na profilu korisnika
 - poveznica na issue: [#56](#)
- Backend – evidentiranje rezervacija (kalendar integracija ili potvrda)
 - poveznica na issue: [#58](#)

12. sastanak

- Datum: 21. prosinca 2025.
 - Prisustvovali: T.Brajko, I.J.Kardum, B.Komočar, M.Malenica, P.Stanarević, F.Mihalj
 - Teme sastanka:
- Promjena pozadine stranica
 - poveznica na issue: [#61](#)
 - Organizator se ne može prijaviti na radionicu
 - poveznica na issue: [#63](#)
 - Backend – dodati ulogu admin u sustav

- poveznica na issue: [#65](#)

13. sastanak

- Datum: 7. siječnja 2026.
- Prisustvovali: T.Brajko, I.J.Kardum, B.Komočar, M.Malenica, P.Stanarević, F.Mihalj
- Teme sastanka:

- Provjera je li adresa radionice/izložbe valjana adresa
 - poveznica na issue: [#71](#)
- Prikaz opisa izložbe na stranici detalja izložbe
 - poveznica na issue: [#80](#)

Plan rada

Aktivnost	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12
Organizacija tima i komunikacije	PS											
Kreiran GitHub repozitorij	BK											
Postavljen Discord server	BK											
Definirane uloge	PS											
Dizajn svih stranica (Figma)		PS, FM	PS, FM									
Funkcionalni zahtjevi	TB, BK	TB, BK		IJK, MM								
Arhitektura i dizajn sustava			BK	IJK, MM	IJK, MM							
Izrada baze podataka		IJK, MM	IJK, MM						IJK, MM			
Backend: Login/Register			BK	TB								
Frontend: Login/Register			FM, PS	FM, PS								
Backend: OAuth (Google prijava)				BK	BK							

Aktivnost	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12
Frontend: OAuth (Google prijava)				PS	PS							
Backend: Kreiranje radionica				BK	TB							
Frontend: Prijava na radionicu				PS	FM							
Backend: Prijava na radionicu				BK	TB							
Frontend: Uređivanje profila					PS	FM						
Backend: Uređivanje profila					BK	TB						
Integracija front- back					BK	PS						
Implementacija karte (OSM / Google Maps)					FM							
Testiranje API-ja				TB	BK							
Testiranje frontenda					PS, FM				PS, FM	PS, FM	PS, FM	
Frontend: Izrada shopa i košarice									PS, FM			
Backend: Shop i košarica (API)									TB, BK			
Frontend: Prikaz izložbi									FM	PS		
Backend: Izložbe (lista)									BK	TB		
Frontend: Detalji izložbe									PS, FM			
Backend: Izložbe (detalji)										BK, TB		
Frontend: Pretraživanje i									PS	PS		

Aktivnost	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12
filtriranje												
Backend: Pretraživanje/filtriranje izložbi								BK, TB	TB, BK			
Frontend: Plaćanje (UI)										FM	FM	
Backend: Plaćanje (API / integracija)												TB, BK
Frontend: Potvrda plaćanja										FM	FM	
Backend: Potvrda plaćanja (status/webhook)												TB, BK
Frontend: Admin										FM	PS	
Backend: Admin										BK, TB	TB, BK	
Testiranje admin sučelja												FM, PS
Testiranje admin API-ja												TB
Dokumentacija (Dnevnik sastajanja)	FM	FM	FM	FM	FM	FM		FM	FM	FM	FM	FM
Dokumentacija (tehnička)		IJK, MM	IJK, MM	IJK, MM					IJK, MM	IJK, MM	IJK, MM	IJK, MM
Ispitivanje programskog rješenja												IJK, MM
Deploy / priprema prezentacije												BK

Tablica aktivnosti

Upravljanje projektom

Aktivnost	Toni Brajko	Ivan Josip Kardum	Bruno Komočar	Marko Malenica	Filip Mihalj	Pavle Stanarević
Opis projektnog zadatka		1h				1h
Funkcionalni zahtjevi	2h	2h	2h	2h		
Opis pojedinih obrazaca				2h		
Dijagram obrazaca				1h		
Sekvencijski dijagrami				2h		
Opis ostalih zahtjeva				1h		
Arhitektura i dizajn sustava		3h				
Baza podataka		1h				
Dijagram razreda		3h				
Dijagram stanja		3h				
Dijagram aktivnosti		3h		2h		
Dijagram komponenti		4h				
Korištene tehnologije i alati				4h		
Ispitivanje programskog rješenja		5h		6h		
Dijagram razmještaja		2h				
Upute za puštanje u pogon				5h		
Dnevnik sastajanja					1h	
Zaključak i budući rad				1h		
Popis literature		1h				

Dodatne stavke

Aktivnost	Toni Brajko	Ivan Josip Kardum	Bruno Komočar	Marko Malenica	Filip Mihalj	Pavle Stanarević
Inicijalizacija projekta			2h			
Postavljanje Docker-a			2h			
Dizajn i modeliranje baze podataka		6h		5h		
Implementacija baze (SQL tablice, ograničenja, relacije)		2h	1h	2h		
Nadogradnje baze (trigeri, inicijalni podaci, administratori)		4h	1h			
Implementacija autentifikacije (Login + Register)	3h		4h			
Implementacija Google servisa OAuth 2.0	4h					
Implementacija backend API-ja za radionice	1h		5h			
Implementacija backend API-ja za rezervacije	1h		3h			
Implementacija backend API-ja za izložbe						
Implementacija backend API-ja za trgovinu i recenzije						
Upload i upravljanje datotekama (slike)	2h					
Frontend dizajn glavnih stranica (layout, navigacija)				5h	8h	
Frontend funkcionalnosti: registracija i prijava				4h	2h	
Frontend funkcionalnosti: radionice i filtriranje					10h	
Frontend funkcionalnosti: izložbe i prijave				2h	8h	
Frontend funkcionalnosti: trgovina i košarica					4h	

Aktivnost	Toni Brajko	Ivan Josip Kardum	Bruno Komočar	Marko Malenica	Filip Mihalj	Pavle Stanarević
Frontend funkcionalnosti: pregled termina putem Google kalendarja					4h	
Frontend funkcionalnosti: pregled i uređivanje profila					8h	
Frontend funkcionalnosti: odabir plana					4h	
Frontend funkcionalnosti: plaćanje					3h	
Frontend funkcionalnosti: admin - odobravanje organizatora i određivanje članarina					3h	
Frontend funkcionalnosti: prikaz lokacije putem karte					6h	
Izrada dizajna u Figmi					5h	5h
Backend funkcionalnosti: dodavanje proizvoda i radionica u košaricu	4h		4h			
Backend funkcionalnosti: plaćanje (Stripe i PayPal)	6h		2h			
Backend funkcionalnosti: odobravanje organizatora na admin panelu	1h		5h			
Backend funkcionalnosti: blokiranje korisnika na admin panelu	1h		4h			
Backend funkcionalnosti: odjavljivanje s radionica	2h		2h			
Backend funkcionalnosti: mogućnost pretplate i slanje mailova	5h		2h			
Postavljanje aplikacije na poslužitelj			3h	5h		

Dijagram pregleda promjena

...

Commits over time

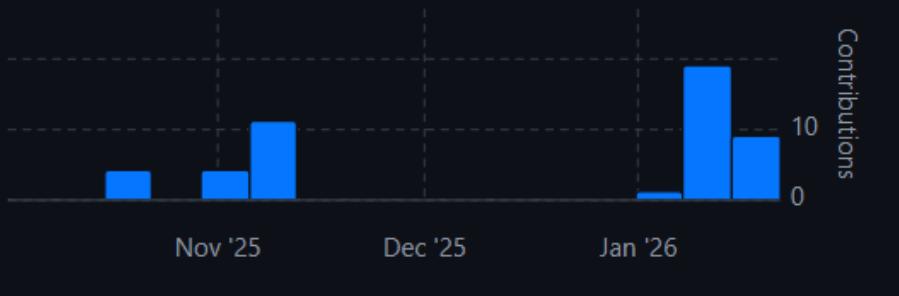
Weekly from Oct 5, 2025 to Jan 18, 2026

**Filip-Mihalj**

48 commits 534,137 ++ 2,626 --

#1

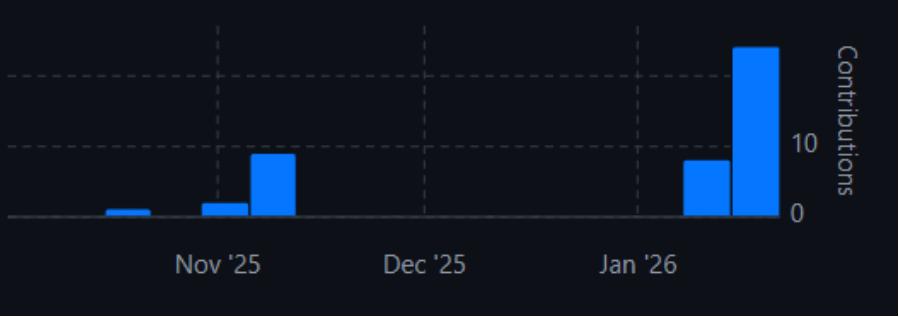
...

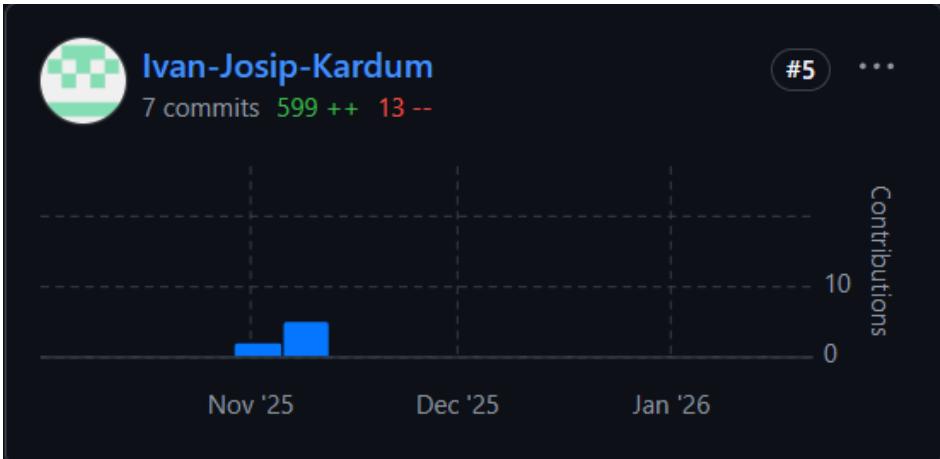
**Bruno-Komocar**

44 commits 9,543 ++ 1,630 --

#2

...





Kjučni izazovi i rješenja

Deploy aplikacije na Render

Izazov:

Prilikom deployanja aplikacije na platformu Render pojavili su se problemi vezani uz konfiguraciju okruženja, povezivanje backend servisa s bazom podataka te pravilno postavljanje produkcijskih varijabli.

Rješenje:

Problem je riješen postupnim testiranjem konfiguracije, jasnim razdvajanjem razvojne i produkcijske okoline te prilagodbom postavki za cloud bazu podataka. Time je omogućeno stabilno i dostupno produkcijsko okruženje aplikacije.

Ispravno modeliranje baze podataka

Izazov:

Tijekom razvoja pokazalo se da početni model baze podataka nije u potpunosti pokriva sve relacije između korisnika, radionica, narudžbi i plaćanja, što je uzrokovalo poteškoće u dalnjoj implementaciji.

Rješenje:

Model baze podataka je revidiran i prilagođen stvarnim poslovnim pravilima aplikacije. Jasnije su definirane relacije i odgovornosti entiteta, što je rezultiralo stabilnjim backendom i lakšim proširenjem funkcionalnosti.

Integracija sustava za plaćanje

Izazov:

Dodavanje online plaćanja zahtjevalo je povezivanje vanjskog servisa s postojećim sustavom narudžbi te osiguravanje ispravnog tijeka procesa naplate.

Rješenje:

Integracija je provedena korištenjem vanjskog servisa za plaćanje uz jasno definirane korake naplate i obradu povratnih informacija. Time je omogućena sigurna i pouzdana realizacija plaćanja unutar aplikacije.

Uvođenje OAuth autentikacije

Izazov:

Implementacija OAuth prijave (npr. putem Google računa) zahtjevala je dodatnu konfiguraciju, rukovanje tokenima i usklađivanje s postojećim sustavom korisnika.

Rješenje:

OAuth autentikacija je uspješno integrirana u backend, uz povezivanje vanjskih korisničkih podataka s internim modelom korisnika. Time je korisnicima omogućena brža i jednostavnija prijava u sustav.

Povezivanje svih komponenti u backendu

Izazov:

Jedan od izazova bio je osigurati pravilnu komunikaciju između frontend dijela aplikacije, backend servisa, baze podataka i vanjskih servisa (plaćanja, OAuth).

Rješenje:

Kroz jasno definirane API rute, servisni sloj i dosljednu strukturu backenda ostvarena je stabilna povezanost svih komponenti sustava. Time je poboljšana održivost i čitljivost koda.

Zaključno

Tijekom razvoja aplikacije ClayPlay tim se susreo s nizom tehničkih i organizacijskih izazova, od modeliranja baze podataka do integracije vanjskih servisa i deployanja aplikacije u produkcijsko okruženje. Sustavnim rješavanjem problema, iterativnim razvojem i timskom suradnjom postignuto je stabilno i funkcionalno programsко rješenje.

Stečena iskustva doprinijela su boljem razumijevanju arhitekture web aplikacija, rada s cloud okruženjima te važnosti kvalitetnog planiranja i testiranja sustava.

Stanari edit

▼ Pages 13

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](#)
- ▶ [B. Dnevnik promjena dokumentacije](#)
- ▼ [C. Prikaz aktivnosti grupe](#)
 - Dnevnik sastajanja
 - Plan rada
 - Tablica aktivnosti
 - Upravljanje projektom
 - Dodatne stavke
 - Dijagram pregleda promjena

Kjučni izazovi i rješenja

- Deploy aplikacije na Render
 - Ispravno modeliranje baze podataka
 - Integracija sustava za plaćanje
 - Uvođenje OAuth autentikacije
 - Povezivanje svih komponenti u backendu
- Zaključno

+ Add a custom sidebar

Clone this wiki locally

<https://github.com/Pavle-Stanarevic/Stanari.wiki.git>

