

Programsko inženjerstvo

Ak.god. 2024./2025.

Easy Flat

Dokumentacija, Rev. 2.0

Grupa: TG 10.4

Voditelj: Tadija Škugor

Datum predaje: 24.1.2025.

Nastavnik: Vlado Sruk

Sadržaj

Sadržaj.....	2
1. Opis projektnog zadatka.....	4
2. Analiza zahtjeva	8
2.1 Funkcionalni zahtjevi.....	8
2.2 Ostali zahtjevi	10
2.2.1 Zahtjevi programske potpore.....	10
2.2.2 Organizacijski zahtjevi.....	10
2.2.3 Vanjski zahtjevi	11
2.3 Dionici	11
3. Specifikacija zahtjeva sustava.....	13
3.1 Obrasci uporabe	13
3.1.1 Opis obrazaca uporabe	13
3.1.2 Dijagrami obrazaca uporabe	21
3.2 Sekvencijski dijagrami	23
3.3 Provjera uključenosti ključnih funkcionalnosti u obrasce uporabe.....	27
4 Arhitektura i dizajn sustava	28
4.1 Opis arhitekture	28
Stil arhitekture.....	28
Podsistavi	29
Spremišta podataka	29
Mrežni protokoli	29
Globalni upravljački tok	29
Obrazloženje odabira arhitekture	30
Fleksibilnost.....	30
Visoka kohezija i niska povezanost.....	30
Skalabilnost	30
Organizacija sustava na visokoj razini.....	30
Organizacija aplikacije	31
4.2 Baza podataka	31
4.2.1 Opis tablica	31
4.2.2 Dijagram baze podataka	36
4.3 Dijagram razreda	36
4.4 Dijagram stanja	37
4.5 Dijagram aktivnosti	38
5 Arhitektura komponenata i razmještaja	40
5.1 Dijagram komponenata	40

5.2 Dijagram razmještaja.....	41
6 Ispitivanje programskog rješenja.....	42
6.1 Ispitivanje komponenti.....	42
6.2 Ispitivanje sustava	46
7 Tehnologije za implementaciju aplikacije	50
7.1 Korištene tehnologije i alati	50
8 Upute za puštanje u pogon.....	51
8.1 Instalacija.....	51
8.2 Postavke.....	52
8.3 Pokretanje aplikacije	52
8.4 Opis pristupa aplikaciji na javnom poslužitelju	53
9 Zaključak i budući rad	53
Popis literature.....	54
Dnevnik promjena dokumentacije.....	54
Prikaz aktivnosti grupe.....	55
Dnevnik sastajanja	55
Tablica aktivnosti	57
Dijagram pregleda promjena	58

1. Opis projektnog zadatka

Cilj ovog projekta je razviti web aplikaciju koja će omogućiti poboljšanje komunikacije i olakšanu koordinaciju inicijativa suvlasnika u zgradama. Često postoji potreba za međusobnom koordinacijom suvlasnika vezano za dijeljenje troškova, izvođenje građevinskih radova i ostalih poslova vezanih uz zgradu. Za ovu svrhu suvlasnici mogu koristiti razne aplikacije općenitije namjene kao što su Discord serveri i WhatsApp grupe ili pak koristiti samo fizičke poruke na oglasnoj ploči zgrade. Ovakva komunikacija se može pokazati izazovna zbog velikog broja suvlasnika. Prednost ove aplikacije je prilagođenost toj specifičnoj potrebi povezivanja suvlasnika pojedine zgrade. Glavni dio, a ujedno i početna stranica aplikacije je oglasna ploča. Na oglasnoj ploči korisnici imaju pristup diskusijama koje su stvorili ostali suvlasnici u zgradama, kao i mogućnost iniciranja vlastite diskusije.

Korisnici aplikacije

Uloge koje korisnici aplikacije mogu imati su suvlasnik zgrade, predstavnik suvlasnika te administrator. Funkcionalnosti kojima može pristupiti suvlasnik zgrade i predstavnik suvlasnika su jednake. Predstavnik suvlasnika nema nikakve dodatne ovlasti unutar same aplikacije osim što zbog svoje pozicije ima uvid u procedure, povijest problema i finansijski kontekst zgrade. Korisnici s ulogom administrator imaju pristup dodatnim funkcionalnostima.

Prijava i registracija

Novi korisnik se ne može sam registrirati u sustav već može samo poslati zahtjev za registraciju. Kada otvori aplikaciju, prikazat će mu se stranica Prijava gdje se prijavljuje u sustav koristeći Google korisnički račun. Ako korisnik nije registriran, aplikacija ga preusmjerava na stranicu za slanje zahtjeva za registraciju. Zahtjev za registraciju sadrži sljedeća polja: ime, prezime, email adresu i stan. Sustav popunjava polja s imenom, prezimenom i email adresom na temelju podataka dobivenih prijavom u Google korisnički račun. Korisnik može promijeniti ime i prezime, ali ne i email adresu. Nakon što korisnik odabere svoj stan iz padajućeg izbornika svih dostupnih stanova u zgradama, može poslati zahtjev za registraciju. Zahtjev za registraciju se može poslati samo jednom s iste email adrese. Nakon slanja zahtjeva za registraciju, korisnik mora čekati da administrator aplikacije potvrdi zahtjev. Registrirani korisnik se nakon uspješne prijave preusmjerava na početnu stranicu aplikacije.

Administrator

Administrator ima mogućnost upravljanja korisničkim računima ostalih korisnika aplikacije. Za razliku od ostalih korisnika, administrator može dodati novog korisnika, izbrisati korisnički račun te prihvati/odbiti zahtjev za registraciju. Nakon prijave, na stranici Osobni podaci ima pristup listi svih korisnika aplikacije gdje može dodavati i brisati korisnike. Pri registriranju novog korisnika vidljivo je ime, prezime, adresa elektroničke pošte, te odabran stan s liste svih stanova u zgradu. Nakon pregleda zahtjeva, administrator odlučuje hoće li ga prihvati ili odbiti. Prihvaćanjem zahtjeva za registraciju korisnik koji ga je poslao dobiva pristup funkcionalnostima aplikacija. Osim dodavanja korisničkih računa, administrator ih može i obrisati. Prilikom pregleda liste korisnika aplikacije, odabirom pojedinog korisničkog računa prikazuje se opcija Deaktiviraj. Ako administrator odabere opciju Deaktiviraj korisnički račun se vraća na listu korisnika koji su zatražili registraciju i navedeni korisnik gubi pristup aplikaciji.

Izbornik

U gornjem desnom kutu aplikacije nalazi se izbornik klikom na koji se prikazuju sljedeće opcije: Osobni podaci, Poruke, Home, Glasanje, Kontakt, Obavijesti i Odjava. Odabirom opcije Odjava korisnik se odjavljuje iz sustava, gubi pristup funkcionalnostima aplikacije i biva preusmјeren na stranicu Prijava.

Opcija Osobni podaci preusmjerava korisnika na stranicu na kojoj vidi svoje osobne podatke: ime, prezime, adresu elektroničke pošte, stan i profilnu sliku. Klikom na gumb Uredi korisnik može mijenjati ime i prezime. Ako korisnik ima ulogu Administratora dostupna je još i lista korisnika koja mu omogućuje upravljanje korisničkim računima.

Opcija Poruke preusmjerava korisnika na stranicu gdje su vidljive obavijesti koje su postavili stanari. Obavijesti može obrisati samo administrator i odabirom opcije Obriš obavijest.

Opcija Glasanje preusmjerava korisnika na stranicu gdje su prikazane sve aktivne forme za glasanje. Korisnik može glasati odabirom jedne od opcija i klikom na Pošaljite glas. Glasati se može samo jednom te se nakon slanja glasa korisniku prikazuje trenutni rezultat glasanja.

Opcija Kontakt preusmjerava korisnika na stranicu gdje su dostupne kontakt informacije vezane za zgradu i aplikaciju.

Opcija Obavijesti preusmjerava korisnika na stranicu s formom za obavijesti. Korisnik može ispuniti formu te će se nakon slanja navedena forma prikazati u centru za poruke gdje će biti vidljiva svim ostalim stanašima.

Oglasna ploča

Na početnoj stranici prikazana je oglasna ploča s aktivnim diskusijama kojima korisnik ima pristup. Korisnik ima pristup svim javnim diskusijama i privatnim diskusijama kojima je na listi sudionika. Na oglasnoj ploči vidljivi su naslov i tekst diskusije, inicijator te datum kreiranja diskusije i glasanje ako postoji.

Diskusija

Diskusije se nalaze na oglasnoj ploči, osim podataka vidljivih na oglasnoj ploči dostupni su još i komentari. Korisnik može vidjeti svoje i tuđe komentare na odabranu diskusiju. Pisanjem komentara i pritiskom gumba Pošaljite odgovor korisnik može objaviti odgovor na diskusiju koji će biti vidljiv svim sudionicima diskusije. Broj dopuštenih odgovora na diskusiju je ograničen te ako korisnik pokuša komentirati nakon što je dosegnut maksimalan broj odgovora prikazat će se prozor s upozorenjem i komentar neće biti objavljen.

Ako je u diskusiji otvoreno glasanje korisnik može glasati odabirom pozitivnog ili negativnog odgovora na postavljeno pitanje i pritiskom gumba Pošaljite glas. Rezultati glasanja vidljivi su svim sudionicima diskusije i ažuriraju se svakim novim glasom. Svaki korisnik može glasati samo jedanput na pojedino pitanje.

Osim pristupa aktivnim diskusijama, korisnik ima pristup i arhivi diskusija gdje se nalaze neaktivne diskusije. Neaktivnim diskusijama se smatraju one u kojima nije objavljen novi komentar ili pokrenuto glasanje više od 30 dana. Na dnu početne stranice odabirom opcije Arhiva korisnik se preusmjerava na stranicu gdje vidi sve neaktivne diskusije kojima ima pristup. Neaktivne diskusije funkcioniraju jednako kao i aktivne.

U zaglavlju početne stranice nalazi se tražilica s pomoću koje se mogu pretražiti sve diskusije. Korisnik može pretražiti diskusije koristeći ključne riječi iz naslova. Pretraživanjem se na oglasnoj ploči prikazuju samo one diskusije koje sadrže navedene ključne riječi u naslovu.

Stvaranje diskusije

Osim pretraživanja i sudjelovanja u diskusijama korisnik može stvoriti vlastitu diskusiju. Klikom na gumb + koji se nalazi u donjem desnom kutu stranice korisnik je preusmjeren na stranicu za kreiranje diskusije. Pri kreiranju nove diskusije upisuju se naslov, tekst i maksimalan broj odgovora, odabire se vrsta diskusije: javna ili privatna te se može dodati glasanje. Ako je odabrana privatna diskusija korisnik mora iz liste korisnika odabrati željene sudionike koji će imati pristup diskusiji. Diskusija neće biti vidljiva korisnicima koji nisu na listi sudionika. Dodavanjem glasanja potrebno je upisati pitanje glasanja. Nakon što korisnik ispuni obrazac za kreiranje diskusije, klikom na gumb Objavi diskusija se postavlja na oglasnu ploču.

Sastanak

U diskusijama u kojima postoji glasanje ako broj pozitivnih glasova premaši četvrtinu svih sudionika iniciator diskusije ima mogućnost kreirati sastanak. Pri kreiranju sastanka navodi naslov i termin sastanka te tekst poziva na sastanak koji uključuje dnevni red i ciljeve sastanka. Svi sudionici diskusije primaju poziv na sastanak te mu mogu prisustvovati korištenjem sučelja StanPlan aplikacije.

Mogućnosti proširenja

Postoje razne mogućnosti proširenja koje bi poboljšale aplikaciju, ali nisu nužne za njezinu funkcionalnost, a neke od njih su:

- Implementacija mobilne aplikacije
- Slanje zahtjeva administratoru za promjenom e-mail adrese
- Mogućnost da svaki korisnik može pregledati osobne podatke ostalih korisnika
- Mogućnost slanja privatnih poruka između dva korisnika

2. Analiza zahtjeva

2.1 Funkcionalni zahtjevi

ID zahtjeva	Opis	Prioritet	Izvor	Kriterij prihvaćanja
F-001	Sustav omogućuje korisnicima kreiranje računa s pomoću email adrese	Visok	Zahtjev dionika	Korisnik se može registrirati e-mailom, te nakon potvrde administratora uspješno se prijaviti.
F-002	Sustav omogućuje korisnicima prijavu preko Google korisničkog računa.	Visok	Zahtjev dionika	Korisnik se može uspješno prijaviti u sustav Google korisničkim računom.
F-003	Sustav omogućuje korisnicima iniciranje javne diskusije.	Visok	Dokument zahtjeva	Korisnik može na početnoj stranici aplikacije kreirati javnu diskusiju o određenoj temi.
F-004	Sustav omogućuje korisnicima iniciranje privatne diskusije.	Visok	Dokument zahtjeva	Korisnik može na početnoj stranici aplikacije kreirati privatnu diskusiju o određenoj temi te dodati željene sudionike.
F-006	Sustav omogućuje inicijatoru diskusije određivanje maksimalnog broja odgovora unutar diskusije.	Srednji	Zahtjev dionika	Inicijator diskusije može odrediti maksimalan broj odgovora koje korisnici mogu postaviti u diskusiji.
F-007	Sustav omogućuje korisnicima pregled liste dostupnih diskusija.	Visok	Zahtjev dionika	Korisnik može vidjeti ažuriranu listu svih aktivnih diskusija.
F-008	Sustav omogućuje korisnicima pregled arhive diskusija.	Visok	Zahtjev dionika	Korisnik ima pristup arhivi diskusija gdje se nalaze sve diskusije koje su neaktivne više od 30 dana.
F-009	Sustav omogućuje korisnicima pretraživanje liste diskusija.	Srednji	Zahtjev dionika	Korisnik može pretražiti postojeće diskusije s pomoću ključnih riječi iz naslova.
F-010	Sustav omogućuje korisnicima	Srednji	Zahtjev dionika	Korisnik može urediti postojeće osobne

	promjenu osobnih podataka.			podatke i uspješno spremiti promjene
F-011	Sustav omogućuje inicijatoru diskusije pokretanje glasanja.	Visok	Zahtjev dionika	Inicijator diskusije može pokrenuti glasanje u kontekstu određene diskusije prilikom kreiranja ili tokom diskusije.
F-012	Sustav omogućuje pristup glasanju sudionicima diskusije.	Visok	Zahtjev dionika	Korisnici s pristupom diskusiji imaju mogućnost glasanja odabirom pozitivnog ili negativnog odgovora.
F-013	Sustav omogućuje korisnicima s pristupom diskusiji uvid u rezultate glasanja.	Visok	Zahtjev dionika	Korisnici s pristupom diskusiji mogu vidjeti prikaz trenutačnog rezultata glasanja koji se osvježuje svakim novim glasom.
F-014	Sustav omogućuje inicijatoru diskusije kreiranje poziva na sastanak	Visok	Zahtjev dionika	Inicijator diskusije može kreirati poziv na sastanak ako broj pozitivnih glasova premašuje četvrtinu svih sudionika diskusije.
F-015	Sustav omogućuje odjavu iz aplikacije.	Visok	Zahtjev dionika	Korisnici se mogu odjaviti iz aplikacije
F-016	Sustav omogućuje administratoru kreiranje novih korisnika.	Visok	Zahtjev dionika	Administrator može kreirati nove korisnike koji se žele registrirati. Prilikom registracije koriste se ime, prezime, lozinka, adresa elektroničke pošte te odabran stan s liste svih stanova u zgradama.
F-017	Sustav omogućuje administratoru prihvatanje zahtjeva za registraciju.	Visok	Zahtjev dionika	Administrator može prihvati zaprimljene zahtjeve za registraciju.
F-018	Sustav omogućuje administratoru brisanje korisničkih računa.	Visok	Zahtjev dionika	Administrator može obrisati korisničke račune ostalih korisnika. Administrator ne može obrisati svoj račun.

F-019	Sustav omogućuje korisnicima objavu komentara na diskusiju.	Visok	Zahtjev dionika	Korisnik može poslati komentar na pojedinu diskusiju čiji je sudionik
-------	---	-------	-----------------	---

2.2 Ostali zahtjevi

2.2.1 Zahtjevi programske potpore

ID zahtjeva	Opis	Prioritet
NF-101	Sustav treba biti oblikovan tako da omogućuje jednostavno održavanje.	Visok
NF-102	Sustavu se pristupa iz javne mreže korištenjem protokola HTTPS.	Visok
NF-103	Sustav omogućava istovremeni rad više korisnika.	Visok
NF-104	Korisnikovo neispravno korištenje ne smije poremetiti rad sustava.	Visok
NF-105	Sustav mora biti jednostavan za korištenje tako da ga je moguće koristiti bez opširnih uputa	Visok
NF-106	Svi privatni podaci korisnika su zaštićeni.	Visok
NF-107	Nadogradnja sustava ne smije narušavati postojeće funkcionalnosti sustava.	Visok
NF-108	Sustav će automatski arhivirati diskusije koje su neaktivne više od 30 dana.	Srednji
NF-109	Pristup diskusijama treba biti ograničen na ovjерene korisnike	Visok
NF-110	Aplikacija treba biti funkcionalna i održavati dosljedno korisničko iskustvo na različitim mobilnim uređajima, uključujući različite veličine zaslona i razlučivosti.	Visok
NF-111	Sustav treba moći obraditi pogreške i oporaviti se od njih bez gubitka podataka ili netočne obrade podataka.	Visok
NF-112	Najčešće korištene značajke moraju biti lako dostupne i pristupljene uz minimalnu navigaciju.	Visok

2.2.2 Organizacijski zahtjevi

ID zahtjeva	Opis	Prioritet
NF-201	Sustav treba imati dovoljnu dokumentaciju	Visok
NF-202	Sustav treba biti opisan putem dokumenta oblikovanja /SRS/.	Visok

NF-203	Sustav treba imati "Plan implementacije" za pravilno postavljanje sustava.	Visok
NF-204	Kôd sustava treba biti napisan objektno orijentiranim pristupom.	Visok
NF-205	Sustav treba koristiti hrvatski jezik.	Visok
NF-206	Sustav treba podržavati hrvatsku abecedu.	Visok

2.2.3 Vanjski zahtjevi

ID zahtjeva	Opis	Prioritet
NF-301	Softver treba raditi na više operativnih sustava, uključujući Windows, macOS i Linux, bez ikakvih izmjena.	Visok
NF-302	Web-aplikacija bi trebala biti potpuno funkcionalna na svim glavnim web-preglednicima, uključujući Chrome, Firefox, Safari i Edge, s dosljednom izvedbom i izgledom.	Visok
NF-303	Sustav treba biti interoperabilan sa sustavom StanPlan aplikacije radi sigurne i učinkovite razmjene podataka.	Visok
NF-304	Sustav mora raditi bez greške u 95 posto slučajeva korištenja tijekom mjeseca.	Visok
NF-305	Aplikacija mora raditi dosljedno na različitim uređajima i operativnim sustavima sa stopom pouzdanosti od 99%.	Visok

2.3 Dionici

Dionici:

- Vlasnik (naručitelj)
- Suvlasnici zgrade
- Običan suvlasnik
- Predstavnik suvlasnika
- Administrator
- Razvojni tim

Aktori i njihovi funkcionalni zahtjevi:

1. Neregistrirani korisnik (inicijator) može:

- se registrirati u sustav, za što su mu potrebni ime, prezime, adresa elektroničke pošte, lozinka, odabran stan s liste svih stanova u zgradi i potvrda administratora. (F-001)

2. Neprijavljeni korisnik (inicijator) može:

- se prijaviti u sustav, za što su mu potrebni adresa elektroničke pošte i lozinka. (F-002)

3. Suvlasnik zgrade (inicijator) može:

- sudjelovati u svim javnim diskusijama
- sudjelovati u privatnim diskusijama kojima je na listi sudionika
- kao sudionik ili inicijator diskusije pristupiti sadržaju, odgovarati svojim porukama i glasati
- kreirati vlastitu javnu ili privatnu diskusiju
- kao inicijator diskusije može odrediti maksimalan broj odgovora unutar diskusije i listu sudionika
- kao inicijator diskusije pokrenuti glasanje
- kao inicijator diskusije kreirati poziv na sastanak
- sudjelovati u sastanku za koji je primio poziv
- promijeniti osobne podatke
- odjaviti se iz sustava

4. Administrator (inicijator) može:

- kreirati nove korisnike pri čemu im dodjeljuje korisničko ime, lozinku i adresu elektroničke pošte
- prihvati nove korisnike koji su poslali zahtjev za registraciju
- odbiti zahtjev za registraciju
- vidjeti popis svih registriranih korisnika i njihove osobne podatke
- deaktivirati profil korisnika

5. Baza podataka (sudionik) može:

- pohraniti sve podatke o korisnicima
- pohraniti sve podatke o diskusijama, rezultatima glasanja i sastancima

6. StanPlan aplikacija (sudionik) može:

- kreirati sastanke što uključuje navođenje naslova i termina sastanka te teksta poziva na sastanak koji uključuje dnevni red i ciljeve sastanka
- preuzeti listu diskusija s pozitivnim ishodom glasanja

7. OAuth 2.0 (sudionik) može:

- autentificirati postojećeg korisnika

3. Specifikacija zahtjeva sustava

3.1 Obrasci uporabe

3.1.1 Opis obrazaca uporabe

UC1 – Prijava

- Glavni sudionik: Korisnik
- Cilj: Dobiti pristup korisničkom sučelju
- Sudionici: Baza podataka, Vanjski servis za autentikaciju (OAuth 2.0)
- Preduvjet:
 1. Postojanje korisničkog računa u bazi podataka
- Opis osnovnog tijeka:
 1. Korisnik odabire opciju Prijava
 2. Sustav preusmjerava korisnika na vanjski servis za autentifikaciju
 3. Korisnik unosi adresu elektroničke pošte s koje je trenutačno spojen ili alternativno unosi email i lozinku
 4. Vanjski sustav potvrđuje ispravnost unesenih podataka i preusmjerava ga natrag u aplikaciju
 5. Korisnik dobiva pristup korisničkom dijelu sustava
 6. Sustav preusmjerava korisnika na početnu stranicu
- Opis mogućih odstupanja:
 - 3.a Korisnik unosi neispravnu adresu elektroničke pošte i/ili lozinku
 - a. Korisnik dobiva poruku o neuspjeloj prijavi
 - b. Korisnik mijenja podatke i ponovno pokušava prijavu ili odustaje

UC2 – Zahtjev za registraciju

- Glavni sudionik: Korisnik
- Cilj: Slanje zahtjeva za registraciju
- Sudionici: Baza podataka
- Preduvjet: -
- Opis osnovnog tijeka:
 1. Korisnik odabire opciju Registracija

2. Korisnik unosi tražene podatke: ime, prezime, adresa elektroničke pošte, lozinka i odabire stan s liste stanova
 3. Korisnik odabire opciju Pošalji
 4. Korisnik dobiva obavijest o uspješno posланом zahtjevu
- Opis mogućih odstupanja:
 - 3.a Korisnik unosi e-mail adresu koju već koristi neki drugi korisnik ili unosi podatke u krivom formatu
 - a. Sustav provjerava jedinstvenost unesenih podataka i vraća poruku o pogrešci
 - b. Korisnik unosi nove podatke i ponovno šalje zahtjev ili odustaje

UC3 – Pregled Korisnika

- Glavni sudionik: Administrator
- Cilj: Pregledati stranicu Korisnici gdje je moguće upravljati korisnicima aplikacije
- Sudionici: Baza podataka
- Preduvjet:
 1. Korisnik je prijavljen u sustav
 2. Korisnik ima ulogu administratora
- Opis osnovnog tijeka:
 1. Administrator odabire opciju Korisnici
 2. Sustav preusmjerava administratora na stranicu gdje je moguće upravljanje korisnicima aplikacije

UC4 – Odjava

- Glavni sudionik: Korisnik
- Cilj: Odjaviti se iz sustava
- Sudionici: -
- Preduvjet:
 1. Korisnik je prijavljen u sustav
- Opis osnovnog tijeka:
 2. Korisnik odabire opciju Odjava
 3. Korisnik gubi pristup korisničkom dijelu sustava
 4. Sustav preusmjerava korisnika na stranicu za prijavu

UC5 – Pregled osobnih podataka

- Glavni sudionik: Korisnik
- Cilj: Pregledati osobne podatke
- Sudionici: Baza podataka
- Preduvjet:
 1. Korisnik je prijavljen u sustav
- Opis osnovnog tijeka:
 1. Korisnik odabire opciju Osobni podaci
 2. Sustav preusmjerava korisnika na stranicu gdje su prikazani osobni podaci: ime, prezime, adresa elektroničke pošte, lozinka, stan, broj telefona i profilna slika

UC6 – Promjena osobnih podataka

- Glavni sudionik: Korisnik
- Cilj: Promijeniti osobne podatke
- Sudionici: Baza podataka
- Preduvjet:
 1. Korisnik je prijavljen u sustav
 2. Korisnik pregledava svoje podatke (UC5)
- Opis osnovnog tijeka:
 1. Korisnik odabire opciju Uredi
 2. Korisnik mijenja podatke
 3. Korisnik odabire opciju Spremi
 4. Sustav ažurira bazu podataka
- Opis mogućih odstupanja:
 - 3.a Korisnik mijenja osobne podatke, ali ne odabere opciju Spremi
 - a. Aplikaciju upozorava korisnika da nije spremio promjene
 - b. Korisnik sprema promjene ili odustaje od njih
 - 3.b Korisnik pokuša promijeniti osobne podatke u nedozvoljeni format
 - a. Aplikacija upozorava korisnika o pogrešci
 - b. Korisnik mijenja osobne podatke u ispravni format ili odustaje od promjena

UC7 – Pregled diskusije

- Glavni sudionik: Korisnik
- Cilj: Pregledati sadržaj pojedine diskusije

- Sudionici: Baza podatka
- Preduvjet:
 1. Korisnik je prijavljen u sustav
- Opis osnovnog tijeka:
 1. Korisnik odabire prozor pojedine diskusije
 2. Aplikacija preusmjerava korisnika na stranicu gdje je prikazan sadržaj odabrane diskusije: naslov, tekst, inicijator, datum objave, komentari i glasanje ako postoji

UC8 – Sudjelovanje u diskusiji

- Glavni sudionik: Korisnik
- Cilj: Objaviti komentar na pojedinu diskusiju
- Sudionici: Baza podataka
- Preduvjet:
 1. Korisnik je prijavljen u sustav
 2. Korisnik pregledava diskusiju (UC7)
- Opis osnovnog tijeka:
 1. Korisnik unosi tekst komentara
 2. Korisnik odabire opciju Komentiraj
 3. Sustav sprema komentar u bazu podataka
 4. Aplikacija prikazuje komentar na stranici diskusije
- Opis mogućih odstupanja:
 - 1.a Korisnik unosi tekst komentara, ali ne odabire opciju Komentiraj
 - a. Aplikacija obavještava korisnika da nije objavio komentar
 - b. Korisnik odabire opciju Komentiraj ili odustane od objave
 - 2.a Diskusija je dosegnula maksimalan broj odgovora
 - a. Aplikacija obavještava korisnika da je dosegnut maksimalan broj odgovora
 - b. Korisnik odustaje od objave komentara

UC9 – Glasanje

- Glavni sudionik: Korisnik
- Cilj: Glasati na postavljeno pitanje u diskusiji
- Sudionici: Baza podataka
- Preduvjet:

1. Korisnik je prijavljen u sustav
 2. Korisnik pregledava diskusiju (UC7)
 3. Postoji glasanje u diskusiji
- Opis osnovnog tijeka:
 1. Korisnik odabire jedan od dva ponuđena odgovora
 2. Korisnik odabire opciju Glasaj
 3. Sustav ažurira rezultate glasanja u bazi podatka i prikazuje ih
 - Opis mogućih odstupanja:
 - 2.a Korisnik odabire opciju Glasaj bez odabranog odgovora
 - a. Aplikacija upozorava korisnika o pogrešci
 - b. Korisnik odabire odgovor i opciju Glasaj ili odustaje od glasanja

UC10 – Pregled arhive diskusija

- Glavni sudionik: Korisnik
- Cilj: Pregledati neaktivne diskusije
- Sudionici: Baza podataka
- Preduvjet:
 1. Korisnik je prijavljen u sustav
- Opis osnovnog tijeka:
 1. Korisnik odabire opciju Arhiva
 2. Sustav preusmjerava korisnika na stranicu Arhiva gdje su prikazane neaktivne diskusije

UC11 – Pretraživanje diskusija

- Glavni sudionik: Korisnik
- Cilj: Pretraživanje diskusija s pomoću ključnih riječi iz naslova
- Sudionici: Baza podataka
- Preduvjet:
 1. Korisnik je prijavljen u sustav
- Opis osnovnog tijeka:
 1. Korisnik u tražilicu upisuje željeni pojam
 2. Korisnik odabire opciju pretraži
 3. Aplikacija na oglasnoj ploči prikazuje diskusije čiji naslov sadrži traženi pojam
- Opis mogućih odstupanja:
 - 1.a Ne postoji diskusija čiji naslov odgovara traženom pojmu
 - a. Sustav prikazuje poruku o nepostojanju tražene diskusije
 - b. Korisnik pretražuje novi pojam ili odustaje od pretraživanja

UC12 – Iniciranje diskusije

- Glavni sudionik: Korisnik
- Cilj: Stvoriti novu diskusiju na oglasnoj ploči
- Sudionici: Baza podataka
- Preduvjet:
 1. Korisnik je prijavljen u sustav
- Opis osnovnog tijeka:
 1. Korisnik odabire opciju +
 2. Sustav korisniku nudi obrazac za kreiranje diskusije
 3. Korisnik odabire tip diskusije: javna ili privatna
 4. Ako je odabrana privatna diskusija, korisnik odabire listu sudionika
 5. Korisnik odabire hoće li diskusija sadržavati glasanje
 6. Ako diskusija sadrži glasanje, korisnik unosi pitanje glasanja
 7. Korisnik unosi maksimalan broj odgovora u diskusiji, naslov i tekst diskusije
 8. Korisnik odabire opciju Objavi
 9. Sustav sprema diskusiju u bazu podataka i prikazuje ju na oglasnoj ploči
- Opis mogućih odstupanja:
 - 5.a Korisnik nije unio sve potrebne podatke o diskusiji
 - a. Aplikacija obavještava korisnika o pogrešci
 - b. Korisnik unosi sve potrebne podatke i objavljuje diskusiju ili odustaje od objave
 - 5.b Korisnik unosi podatke u obrazac, ali ne odabire opciju Objavi
 - a. Aplikacija obavještava korisnika da nije objavio diskusiju prije izlaska iz prozora
 - b. Korisnik odabire opciju Objavi ili odustaje od objave

UC13 – Otvaranje glasanja

- Glavni sudionik: Korisnik
- Cilj: Otvoriti glasanje u diskusiji
- Sudionici: Baza podataka
- Preduvjet:
 1. Korisnik je prijavljen u sustav
 2. Korisnik je vlasnik diskusije
 3. Korisnik pregledava diskusiju (UC7)

- Opis osnovnog tijeka:
 1. Korisnik odabire opciju Uredi
 2. Aplikacija otvara prozor s obrascem za popunjavanje
 3. Korisnik označava da na pitanje sadrži li diskusija glasanje
 4. Korisnik unosi pitanje glasanja
 5. Korisnik odabire opciju Spremi
 6. Sustav spremi promjene u bazu podataka i prikazuje glasanje na stranici diskusije
- Opis mogućih odstupanja:
 - 5.a Korisnik ne unoše pitanje glasanja, a odabere opciju Spremi
 - a. Aplikacija obavještava korisnika o pogrešci
 - b. Korisnik unosi pitanje glasanja ili odustaje od otvaranja glasanja

UC14 – Dodavanje novog korisnika

- Glavni sudionik: Administrator
- Cilj: Kreiranje i dodavanje novog korisnika u sustav
- Sudionici: Baza podataka
- Preduvjet:
 1. Korisnik je prijavljen kao administrator
- Opis osnovnog tijeka:
 1. Administrator odabire opciju Dodaj
 2. Aplikacija administratoru nudi formu s osnovnim podacima o korisniku
 3. Administrator unosi podatke o novom korisniku: ime, prezime, adresu električne pošte, lozinku, stan i ulogu
 4. Administrator odabire opciju Stvori
 5. Sustav spremi podatke o novom korisniku u bazu podataka
 6. Administrator dobiva obavijest o uspješnom stvaranju novog korisnika
- Opis mogućih odstupanja:
 - 3.a Administrator unosi e-mail adresu koju već koristi neki drugi korisnik ili unosi podatke u krivom formatu
 - a. Sustav provjerava jedinstvenost unesenih podataka i vraća poruku o pogrešci
 - b. Administrator unosi nove podatke i ponovno odabire opciju Stvori ili odustaje

UC15 – Prihvatanje zahtjeva za registraciju

- Glavni sudionik: Administrator
- Cilj: Prihvati zahtjev za registraciju koji je posao novi korisnik
- Sudionici: Baza podataka
- Preduvjet:
 1. Korisnik je prijavljen u sustav
 2. Korisnik ima ulogu administratora
 3. Korisnik pregledava stranicu Korisnici
- Opis osnovnog tijeka:
 1. Administrator odabire opciju Zahtjevi
 2. Aplikacija prikazuje prozor sa svim prispjelim zahtjevima za registraciju
 3. Nakon provjere zahtjeva, administrator odabire opciju Prihvati
 4. Sustav sprema podatke o novom korisniku u bazu podataka
 5. Administrator dobiva obavijest o uspješnom stvaranju novog korisnika

UC16 – Brisanje korisničkog računa

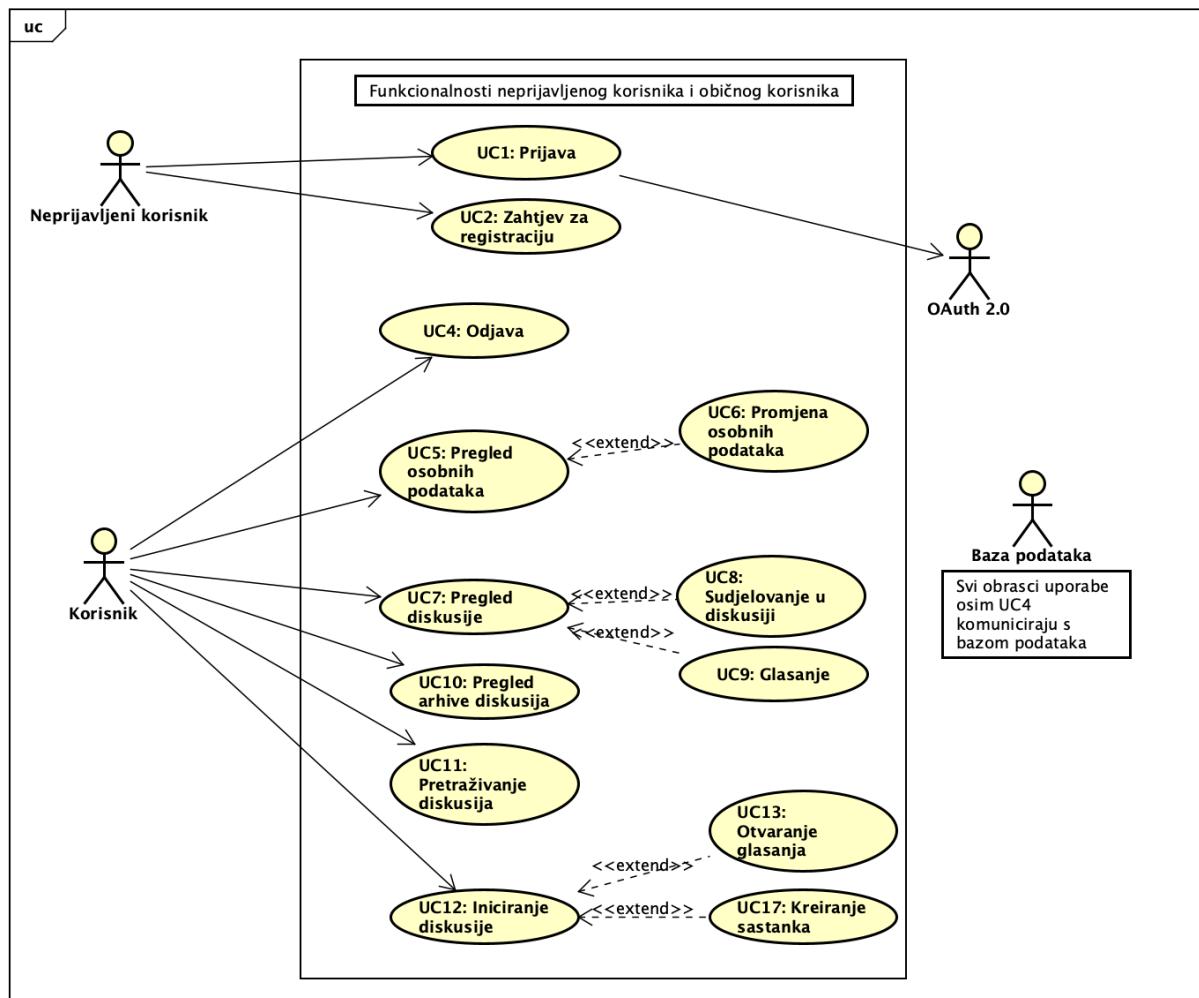
- Glavni sudionik: Administrator
- Cilj: Obrisati korisnički račun
- Sudionici: Baza podataka
- Preduvjet:
 1. Korisnik je prijavljen u sustav
 2. Korisnik ima ulogu administratora
 3. Korisnik pregledava stranicu Korisnici
- Opis osnovnog tijeka:
 1. Administrator odabire opciju Obriši
 2. Aplikacija prikazuje prozor sa svim korisničkim računima
 3. Administrator odabire korisnički račun koji želi obrisati
 4. Administrator odabire opciju Obriši
 5. Sustav ažurira bazu podataka
 6. Administrator dobiva obavijest o uspješnom brisanju korisnika
- Opis mogućih odstupanja:
 - 3.a Administrator odabire svoj korisnički račun
 - a. Aplikacija obavještava korisnika o pogrešci
 - b. Korisnik odabire drugi račun ili odustaje

UC17 – Kreiranje sastanka

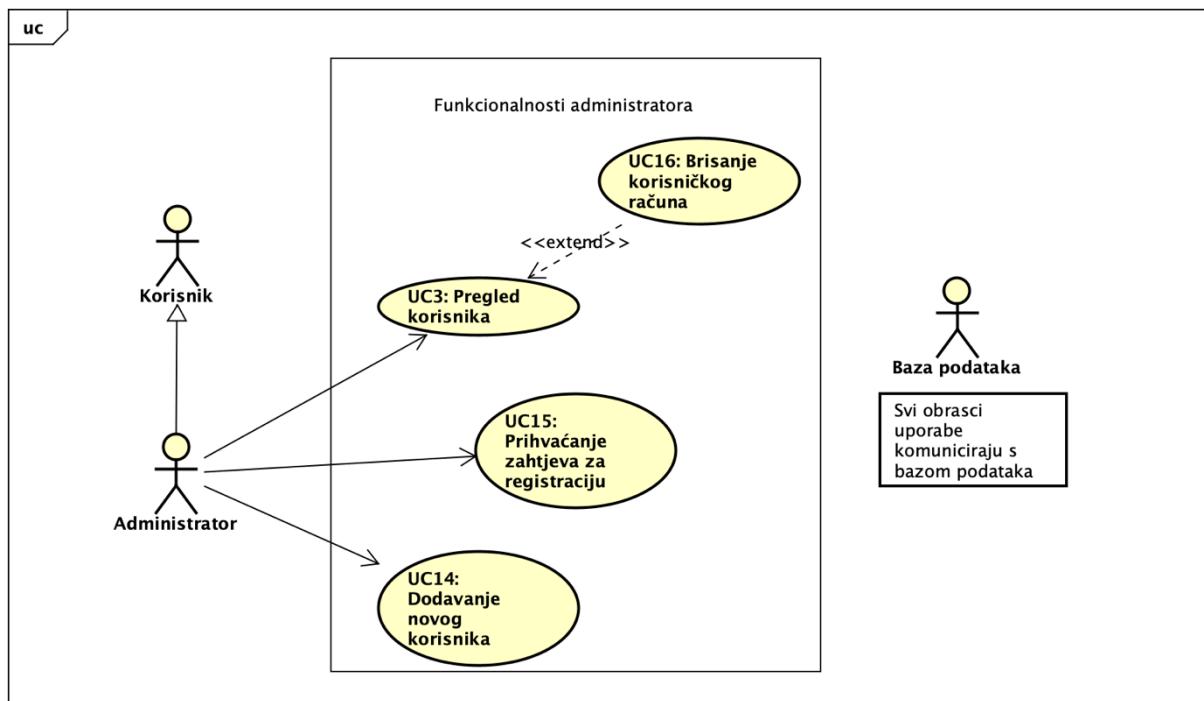
- Glavni sudionik: Korisnik
- Cilj: Kreiranja sastanka na temu diskusije
- Sudionici: Baza podataka, StanPlan aplikacija
- Preduvjet:

1. Korisnik je prijavljen u sustav
 2. Korisnik je vlasnik diskusije
 3. Korisnik pregledava diskusiju (UC7)
 4. U diskusiji je otvoreno glasanje (UC13)
 5. Broj pozitivnih glasova premašuje četvrtinu sudionika diskusije
- Opis osnovnog tijeka:
 1. Korisnik odabire opciju Sastanak
 2. Sustav otvara prozor s formom za popunjavanja
 3. Korisnik unosi naslov, termin i tekst poziva na sastanak
 4. Korisnik odabire opciju Stvori
 5. Sustav sprema promjene u bazu podataka
 6. Sustav razmjenjuje podatke s StanPlan aplikacijom
 7. Korisnik dobiva obavijest o uspješnom stvaranju sastanka
 - Opis mogućih odstupanja:
 - 3.a Korisnik ne unese sve potrebne podatke ili ih unese u krivom formatu
 - a. Aplikacija upozorava korisnika o pogrešci
 - b. Korisnik unosi ispravne podatke ili odustaje od kreiranja sastanka

3.1.2 Dijagrami obrazaca uporabe



Slika 3. 1 Dijagram obrasca uporabe, funkcionalnost neprijavljenog korisnika i običnog korisnika

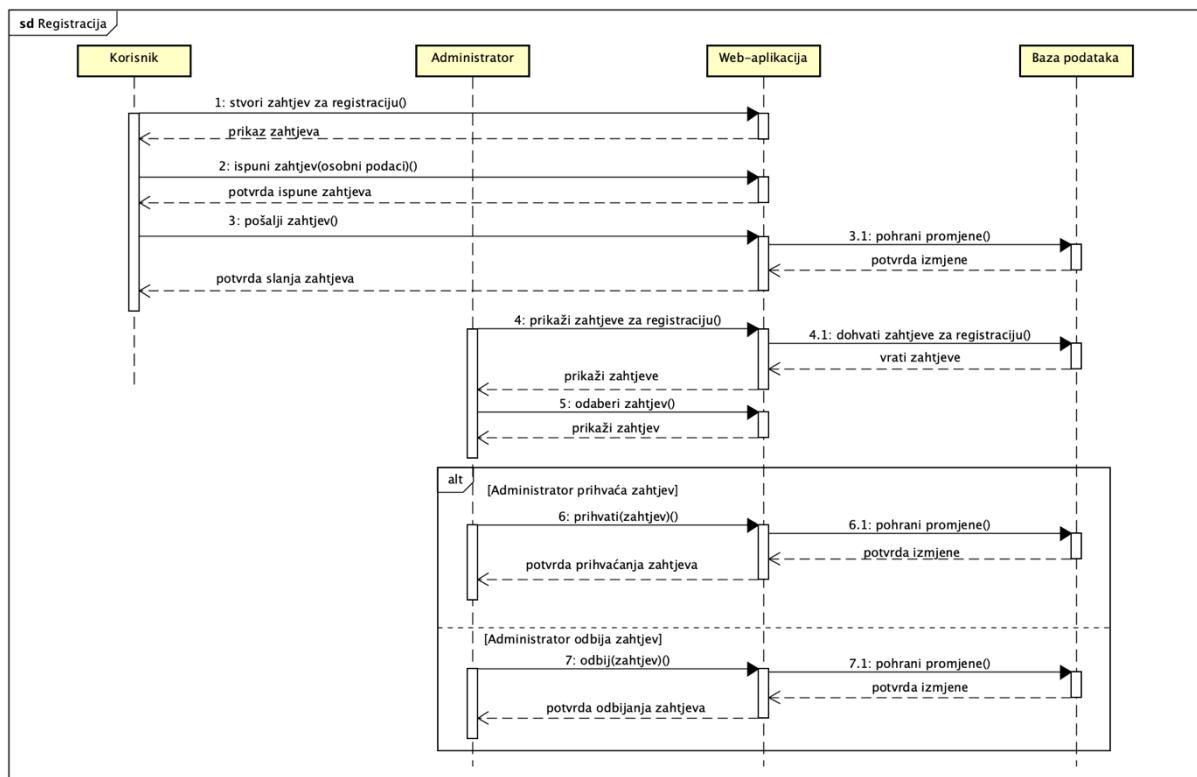


Slika 3. 2 Dijagram obrasca uporabe, funkcionalnost administratora

3.2 Sekvencijski dijagrami

Registracija - Obrasci uporabe UC2 i UC15

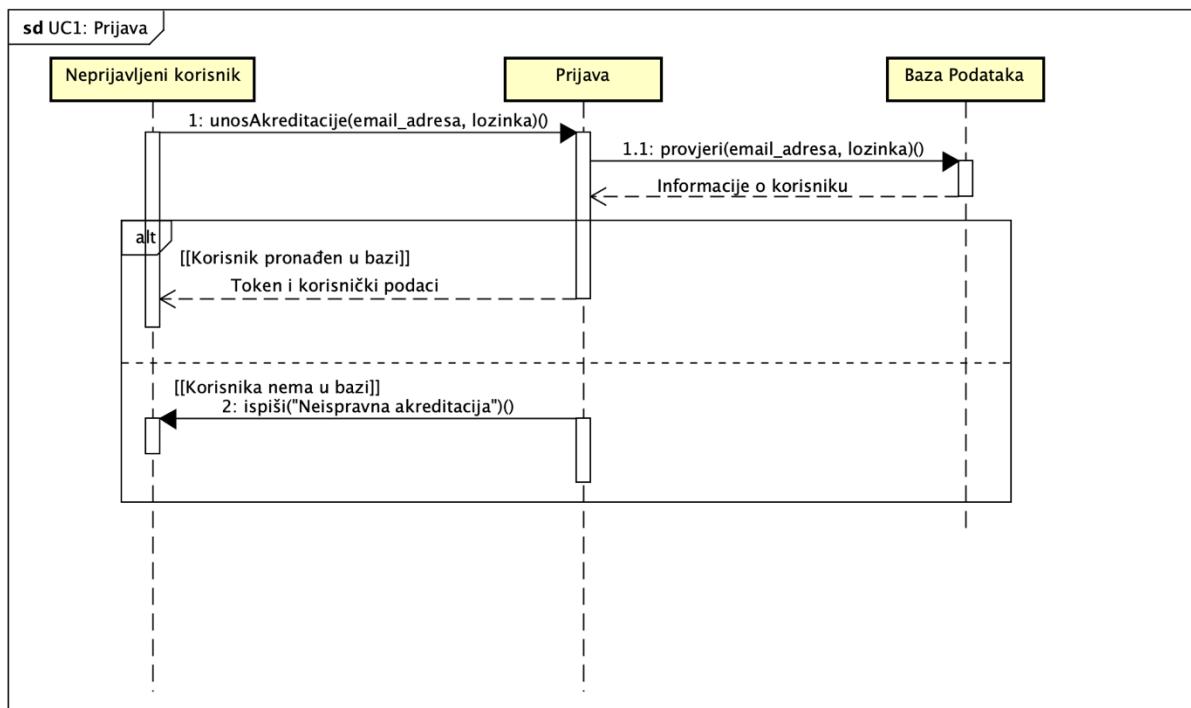
Dijagram prikazuje proces slanja zahtjeva za registraciju i prihvatanje ili odbijanje navedenog zahtjeva od strane administratora. Korisnik šalje poslužitelju zahtjev za registraciju, poslužitelj vraća prikaz zahtjeva. Zatim korisnik unosi osobne podatke: ime, prezime, stan i adresu elektroničke pošte. Nakon ispunjenja zahtjeva poslužitelj traži potvrdu ispunjenosti zahtjeva te korisnik šalje zahtjev za registraciju. Poslužitelj pohranjuje promjene u bazu podataka i nakon što baza potvrdi izmjene, šalje potvrdu slanja zahtjeva korisniku. Administrator šalje zahtjev za prikaz zahtjeva za registraciju. Poslužitelj dohvaća zahtjeve iz baze podataka te ih prikazuje administratoru. Administrator odabire željeni zahtjev te ga poslužitelj prikazuje. Administrator prihvata odabrani zahtjev, poslužitelj pohranjuje promjene u bazu podataka i nakon što baza potvrdi izmjene, šalje administratoru potvrdu o prihvatanju zahtjeva za registraciju. Ako administrator odbije zahtjev, poslužitelj pohranjuje promjene u bazu podataka te vraća potvrdu odbijanja zahtjeva administratoru.



Slika 3. 3 Sekvenčni dijagram za UC2 i UC15

Prijava - Obrazac uporabe UC1

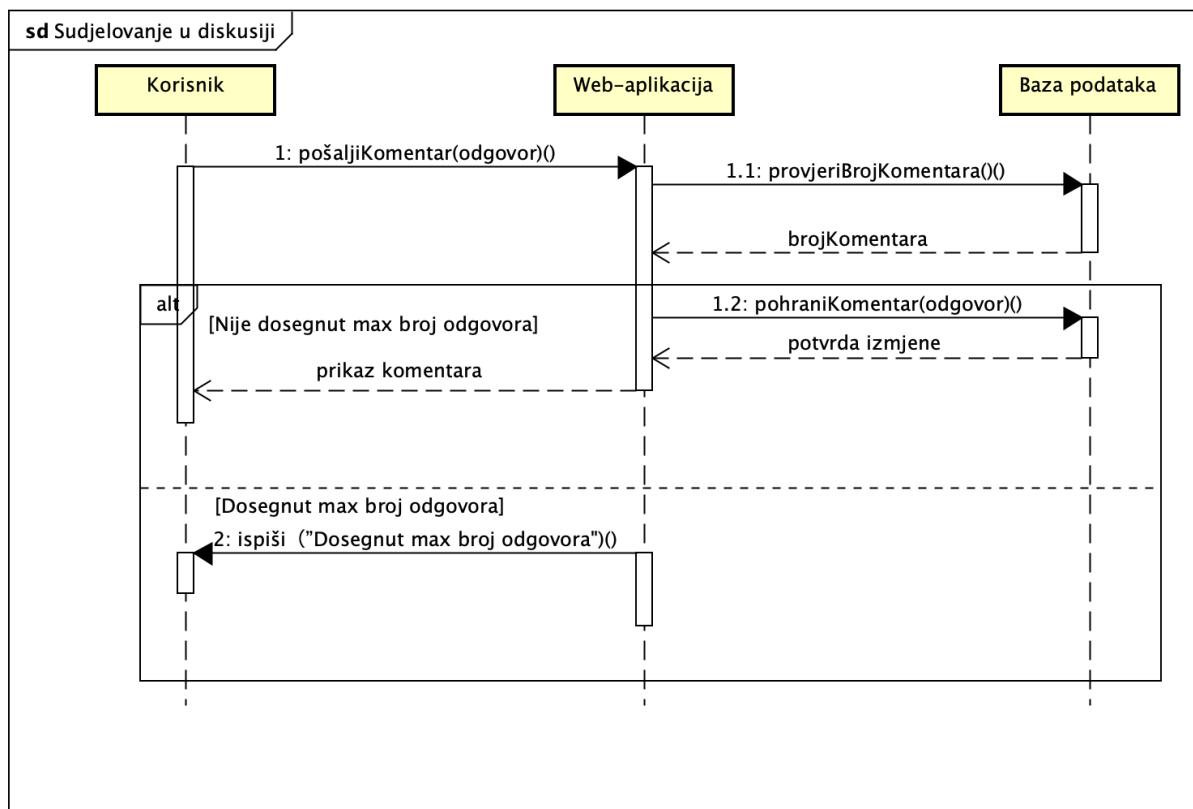
Dijagram prikazuje proces prijave korisnika u sustav. Korisnik upisuje email i lozinku te se odabirom opcije prijava uneseni podaci šalju na validaciju. Ako su uneseni podaci pronađeni u bazi, aplikacija generira token sesije. U suprotnom, šalje se obavijest o neispravnoj akreditaciji.



Slika 3. 4 Sekvencijski dijagram za UC1

Sudjelovanje u diskusiji - Obrazac uporabe UC8

Dijagram prikazuje proces slanja komentara na pojedinu diskusiju. Korisnik unosi odgovor na diskusiju te se odabirom opcije Komentiraj uneseni odgovor šalje na poslužitelj. Poslužitelj šalje zahtjev bazi podataka za provjeru broja komentara u diskusiji, baza odgovara brojem komentara u diskusiji. U slučaju da maksimalan broj odgovora nije dosegnut, promjene se pohranjuju u bazu podataka te se nakon primljene potvrde izmjene prikazuje komentar. U suprotnom, korisnik prima poruku o nemogućnosti komentiranja na odabranu diskusiju zbog dosegnutog maksimalnog broja odgovora.

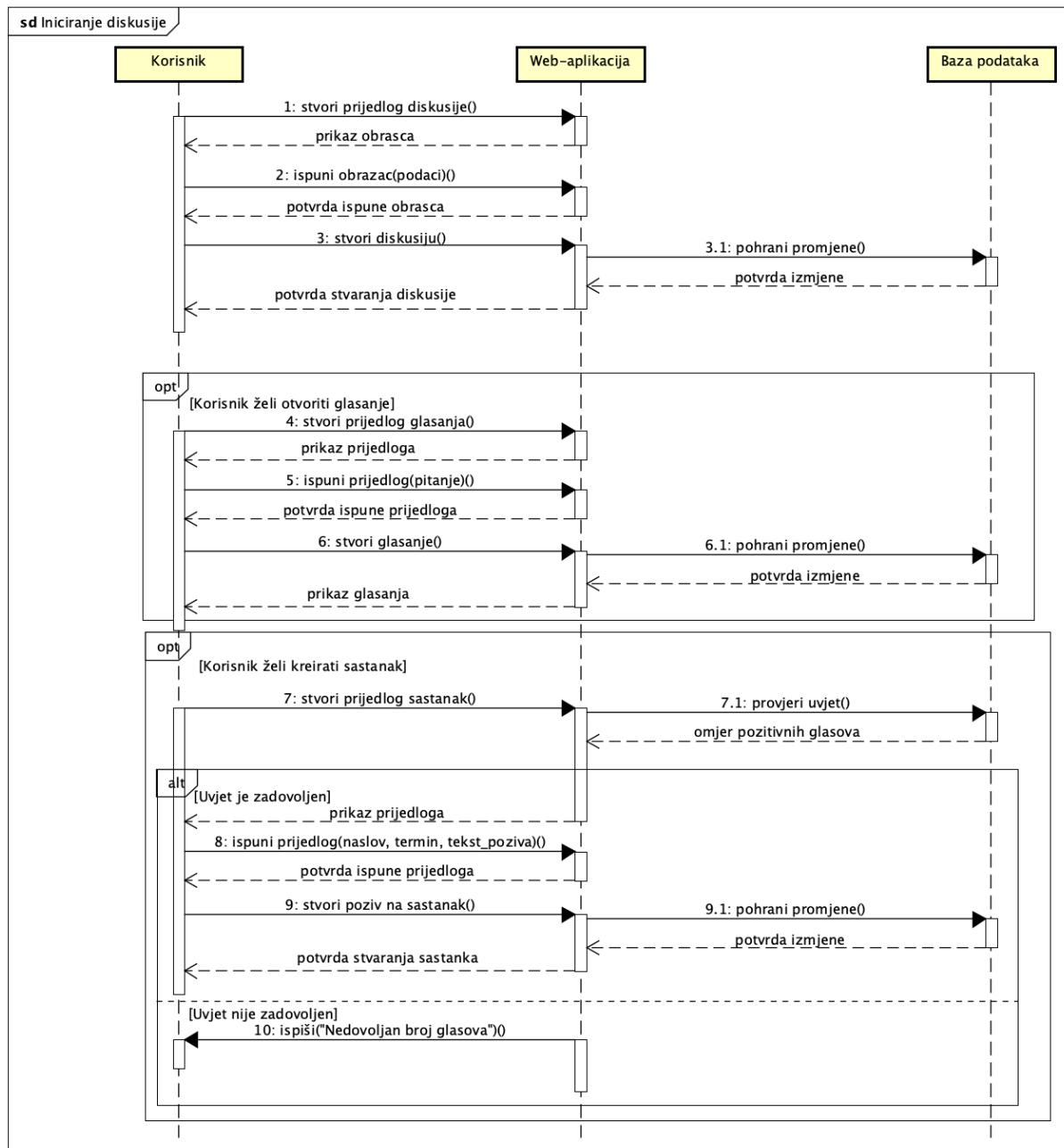


Slika 3. 5 Sekvencijski dijagram za UC8

Iiniciranje diskusije - Obrasci uporabe UC12, UC13 i UC17

Dijagram prikazuje proces iniciranja diskusije. Korisnik šalje zahtjev za stvaranje diskusije, poslužitelj vraća prikaz obrasca za stvaranje diskusije. Korisnik unosi tražene podatke: naslov, sadržaj i listu sudionika. Nakon ispune obrasca sustav traži potvrdu ispune obrasca te korisnik šalje zahtjev za stvaranje diskusije. Sustav pohranjuje promjene u bazu podataka te nakon dobivene potvrde o uspješnoj izmjeni podataka, potvrđuje stvaranje diskusije. Nakon iniciranja diskusije, korisnik ima mogućnost otvaranja glasanja. Korisnik šalje zahtjev za stvaranje glasanja te poslužitelj vraća prikaz prijedloga za glasanje. Nakon ispune zahtjeva sustav traži potvrdu ispune zahtjeva nakon čega se šalje zahtjev za otvaranje glasanja. Sustav pohranjuje izmjene u bazu podataka te nakon što primi potvrdu, prikazuje glasanje. Stvaranjem glasanja korisnik dobiva mogućnost kreiranja sastanka. Uvjet za kreiranje sastanka jest da broj pozitivnih glasova premašuje četvrtinu svih sudionika diskusije. Zato nakon što korisnik pošalje zahtjev za kreiranje sastanka, sustav provjera podatke o glasanju pohranjene u bazi podataka. Baza vraća omjer pozitivnih glasova. Ako je uvjet zadovoljen, prikazuje se prijedlog za kreiranje sastanka. Korisnik unosi tražene podatke (naslov, termin i tekst poziva na sastanak) nakon čega sustav traži potvrdu ispune prijedloga. Promjene se pohranjuju u bazu podataka te nakon što baza potvrdi izmjene, korisnik prima poruku o uspješnom stvaranju sastanka. U suprotnom, ako uvjet za stvaranje sastanka nije

zadovoljen korisnik prima poruku o nedovoljnem broju pozitivnih glasova potrebnih za kreiranje sastanka.



Slika 3. 6 Sekvencijski dijagram za UC12, UC13 i UC17

3.3 Provjera uključenosti ključnih funkcionalnosti u obrasce uporabe

ID obrasca uporabe	ID funkcionalnih zahtjeva
UC1	F-002

UC2	F-001
UC3	F-016
UC4	F-015
UC5	F-010
UC6	F-010
UC7	F-007
UC8	F-019
UC9	F-012 F-013
UC10	F-008
UC11	F-009
UC12	F-003 F-004 F-006 F-011
UC13	F-011
UC14	F-016
UC15	F-017
UC16	F-018
UC17	F-014

4 Arhitektura i dizajn sustava

4.1 Opis arhitekture

Stil arhitekture

Sustav je dizajniran prema arhitekturi klijent-poslužitelj (eng. Client-server). Kod ovog arhitekturnog stila, neka računala služe kao centri za razmjenu informacija i pružanje usluga. Takva računala nazivaju se poslužitelji (eng. Server). Druga računala služe za pristup informacijama i dobivanje usluga te se ona nazivaju klijenti (eng. Client). Prednosti arhitekture klijent-poslužitelj su:

- Klijenti udaljeno pristupaju funkcionalnostima poslužitelja
- Klijent i poslužitelj se mogu oblikovati odvojeno
- Oba entiteta mogu biti jednostavnija
- Svi podaci mogu se držati na jednom mjestu (na poslužitelju)
- Poslužitelju može istovremeno pristupiti više klijenata

Podsistemi

Frontend je zaslužan za interakciju s korisnicima. Omogućuje dohvaćanje i prikaz resursa kao i slanje zahtjeva i podataka na poslužitelj. Za izradu frontend-a korišten je jezik JavaScript i biblioteka React.

Backend obrađuje zahtjeve korisnika, pohranjuje podatke u bazu, dohvaća podatke iz baze, upravlja poslovnom logikom sustava i omogućava komunikaciju s frontend-om. Za izradu backend-a korišten je radni okvir Express.

Baza podataka služi za pohranu podataka sustava. Koristili smo PostgreSQL relacijsku bazu podataka.

Autentifikacija i autorizacija temelji se na OAuth 2.0 standardu koji omogućuje provjeru autentičnosti korisnika i pristup zaštićenim podacima. Za autentificirani pristup zaštićenim resursima koristi se JSON Web Token (JWT).

Spremišta podataka

Za pohranu podataka koristi se PostgreSQL relacijska baza podataka.

Mrežni protokoli

HTTP služi za komunikaciju između klijenta i poslužitelja. Ovaj protokol koristi razne metode kao što su POST, GET, PUT, DELETE i slično kako bi omogućio upravljanje podacima i pristupanje resursima sustava.

HTTPS je sigurna verzija HTTP protokola. Koristi TLS protokol za enkripciju podataka u prijenosu pomoću asimetričnog javnog ključa.

Globalni upravljački tok

1. Početna interakcija korisnika (klijent)

- Korisnik pristupa aplikaciji putem web preglednika.
- Prikazuje se korisničko sučelje i omogućava se interakcija s aplikacijom.
- Korisnik se može prijaviti ili poslati zahtjev za registraciju ako nema korisnički račun.
- Akcija koju korisnik poduzme inicira HTTP zahtjev, koji sadrži podatke potrebne za daljnju obradu, i šalje se na poslužitelj.

2. Obrada zahtjeva (poslužitelj)

- Poslužitelj prima HTTP zahtjev, obrađuje ga i pokreće odgovarajuće procese.
- Provjerava se autentičnost korisnika s pomoću JWT tokena.
- Ovisno o zahtjevu, poslužitelj komunicira s bazom podataka kako bi dohvatio ili pohranio podatke.
- Nakon obrade zahtjeva, poslužitelj šalje odgovor u JSON formatu, koji može sadržavati potrebne podatke ili statusnu informaciju.

3. Konzistentnost podataka

- Poslužitelj komunicira s bazom podataka kako bi dohvatio ili pohranio podatke.
- Sve promjene u bazi podataka odražavaju stvarni status podataka u aplikaciji.

4. Povratne informacije korisnicima (klijent)

- Poslužitelj šalje odgovor u JSON formatu koji se prikazuje korisniku na odgovarajući način.

5. Sigurnost

- Tijekom cijelog procesa osigurava se zaštita podataka korištenjem JWT-a za autorizaciju.

Obrazloženje odabira arhitekture

Fleksibilnost

Klijent i poslužitelj su odvojene komponente što omogućuje njihovo odvojeno oblikovanje. Ovaj pristup pruža fleksibilnost u odabiru tehnologija budući da se mogu koristiti različite tehnologije za frontend i backend.

Visoka kohezija i niska povezanost

Ova arhitektura minimizira povezanost između klijenta i servera dok istovremeno osigurava jasno definiranu komunikaciju.

Skalabilnost

Arhitektura klijent-poslužitelj omogućava istovremeni pristup više korisnika s mogućnosti daljnog rasta. Klijent i poslužitelj se mogu oblikovati odvojeno što omogućuje jednostavno skaliranje poslužitelja kako bi se mogao obraditi veći broj korisnika i podataka.

Organizacija sustava na visokoj razini

Web preglednik je program preko kojeg korisnik pristupa sustavu. Korisnik putem web preglednika šalje zahtjeve za resursima, koje web preglednik dohvaća od web poslužitelja, i onda ih interpretira i prikazuje na ekranu korisnika. Korisnik također može i slati podatke preko web aplikacije (npr. Korištenjem formi) koji se pohranjuju na poslužitelju.

Web poslužitelj je centralni dio web aplikacije. Zaslužan je za pohranjivanje podataka u bazu, obrađivanje poslanih podataka i dostavljanje resursa klijentima web stranice. Komunikacija između web poslužitelja i web klijenta odvija se korištenje protokola HTTP. Web stranice koje web poslužitelj isporučuje klijentu su HTML dokumenti, koji uključuju tekst, slike, stilove, skripte i drugi sadržaj.

Baza podataka služi za pohranu podataka sustava. Web aplikacija tijekom svog rada komunicira s bazom podataka te iz nje dohvata podatke ili ih u nju pohranjuje.

Organizacija aplikacije

Aplikacija je oblikovana korištenjem arhitekturnog obrasca MVC arhitekture.

Aplikacija se dijeli na tri komponente:

- **Model** je glavna komponenta sustava. Sadrži razrede čiji se objekti obrađuju.
- **Pogled** (eng. View) sadrži razrede čiji objekti služe za prikaz podatka.
- **Nadglednik** (eng. Controller) sadrži razrede koji upravljaju i rukuju korisničkom interakcijom s pogledom i modelom.

4.2 Baza podataka

Sustav koristi relacijsku bazu podataka koja se sastoji od entiteta definiranih imenom i skupom atributa. Zadaća baze podataka je pohrana podataka o korisnicima, stanovima, diskusijama i glasanju. Entiteti od kojih se sastoji baza podataka su:

- Korisnik
- Stan
- Zgrade
- Diskusija
- Glasanje
- Glasanje_forma
- Upit
- Sastanak
- Arhiva

4.2.1 Opis tablica

Korisnik

Entitet Korisnik sadrži podatke o korisniku aplikacije. Sadrži atribute: Email (PRIMARY KEY), Ime, Prezime, Lozinka, Aktivan, Suvlasnik, Zgrada_id (FOREIGN KEY) i Stan_id (FOREIGN KEY). Ovaj entitet je u vezi One-to-One s entitetom Stan preko atributa Stan_id i u vezi _Many-to-Many_s entitetom Diskusija preko atributa Email. Također, je u vezi One-to-One sa slabim entitetom Glasanje preko atributa Email i id te u vezi Many-to-One s entitetom Zgrade preko atributa Zgrada_id.

Atribut	Tip podatka	Opis varijable
Email	VARCHAR	Jedinstvena email adresa
Ime	VARCHAR	Ime korisnika
Prezime	VARCHAR	Prezime korisnika
Lozinka	VARCHAR	Šifrirana lozinka
Aktivan	BOOLEAN	Oznaka je li korisnik registriran
Suvlasnik	BOOLEAN	Oznaka je li korisnik suvlasnik zgrade
Stan_id	INT	Identifikacijski ključ korisnikovog stana
Zgrada_id	INT	Identifikacijski ključ korisnikove zgrade

Stan

Entitet Stan sadrži podatke o stanovima u zgradama. Sadrži atribute: Stan_id (PRIMARY KEY) i Zauzet. Ovaj entitet je u vezi *One-to-One* s entitetom Korisnik preko atributa Stan_id.

Atribut	Tip podatka	Opis varijable
Stan_id	INT	Jedinstveni identifikator stana
Zauzet	BOOLEAN	Oznaka je li stan zauzet

Zgrade

Entitet Zgrade sadrži podatke o zgradama čiji stanari mogu imaju pristup aplikaciji. Sadrži atribute: id (PRIMARY KEY), naziv_zgrade i slika_link. Ovaj entitet je u vezama *One-to-Many* s entitetima Korisnik, Upit, Diskusija i Glasanje_forma, preko atributa id.

Atribut	Tip podatka	Opis varijable
id	INT	Jedinstveni identifikator zgrade
naziv_zgrade	VARCHAR	Ime zgrade
slika_link	text	link na sliku zgrade

Diskusija

Entitet Diskusija sadrži podatke o diskusijama koje su korisnici inicirali. Sadrži atribute: Id (PRIMARY KEY), Naslov, Odgovori, Kreator, Datum_stvorena, Zadnji_pristup, Br_odgovora, Opis, Id_forme(FOREIGN KEY) i Zgrada_id (FOREIGN KEY). Ovaj entitet je u vezi *One-to-One* s entitetom Glasanje_forma preko atributa Id_forme te u vezi *Many-to-One* s entitetom Zgrade preko atributa Zgrada_id. Također, je u vezi *Many-to-Many* s entitetom Korisnik preko atributa Id i Email. Povezan je sa slabim entitetom Sastanak vezom *One-to-One* preko atributa Id.

Atribut	Tip podatka	Opis varijable
Id	INT	Jedinstveni identifikator diskusije
Naslov	VARCHAR	Naslov diskusije
Odgovori	TEXT	Odgovori korisnika u diskusiji
Kreator	VARCHAR	Identifikator inicijatora diskusije
Datum_stvorena	DATE	Datum stvaranja diskusije
Zadnji_pristup	DATE	Datum zadnjeg pristupa diskusiji
Br_odgovora	INT	Broj odgovora
Opis	TEXT	Tekst opisa diskusije
Id_forme	INT	Jedinstveni identifikator forme glasanja
Zgrada_id	INT	Identifikacijski ključ korisnikove zgrade

Glasanje_forma

Entitet Glasanje_forma sadrži podatke o glasanju unutar pojedine diskusije. Sadrži atribute: Id (PRIMARY KEY), Datum_stvorenje, Datume_istece, Glasovanje_da, Glasovanje_ne, Kreator, Naslov i Zgrada_id (FOREIGN KEY). Ovaj entitet je u vezi *One-to-One* s entitetom Diskusija preko atributa id te u vezi *Many-to-One* s entitetom Zgrade preko atributa Zgrada_id. Također, je u vezi *One-to-Many* sa slabim entitetom Glasanje preko atributa Id.

Atribut	Tip podatka	Opis varijable
Id	INT	Jedinstveni identifikator forme glasanja
Datum_stvoreno	DATE	Datum stvaranja forme
Datum_istec	DATE	Datum isteka forme
Glasovanje_da	INT	Broj pozitivnih glasova
Glasovanje_ne	INT	Broj negativnih glasova
Kreator	VARCHAR	Identifikator inicijatora diskusije
Naslov	CHAR	Naslov forme
Zgrada_id	INT	Identifikacijski ključ korisnikove zgrade

Glasanje

Entitet Glasanje je slabi entitet koji sadrži podatke o glasanju korisnika. Sadrži atribute: Odgovor, Id_forme (FOREIGN KEY) i Mail_glasaca (FOREIGN KEY). Ovaj entitet je u vezi *One-to-One* s entitetom Korisnik preko atributa Mail_glasaca. Također, je u vezi *Many-to-One* s entitetom Glasanje_forma preko atributa Id_forme.

Atribut	Tip podatka	Opis varijable
Odgovor	BOOLEAN	Odgovor korisnika
Id_forme	INT	Jedinstveni identifikator forme glasanja
Mail_glasaca	VARCHAR	Jedinstvena email adresa korisnika

Upit

Entitet Upit je slabi entitet koji sadrži podatke o upitima koje su poslali korisnici aplikacije. Sadrži atribute: emailosobe (FOREIGN KEY), tekst, rjeseno i Zgrada_id (FOREIGN KEY). Ovaj je u vezama *Many-to-One* s entitetima Korisnik i Zgrade preko atributa emailosobe, odnosno Zgrada_id.

Atribut	Tip podatka	Opis varijable
emailosobe	VARCHAR	Jedinstveni identifikator korisnika
tekst	VARCHAR	sadržaj upita
rjeseno	BOOLEAN	Oznaka je upit rjesen
Zgrada_id	INT	Identifikacijski ključ korisnikove zgrade

Sastanak

Entitet Sastanak je slabi entitet koji sadrži podatke o sastancima. Sadrži atribute: id_diskusije (FOREIGN KEY) i link. Ovaj entitet je u vezi One-to-One s entitetom diskusija preko atributa id_diskusije.

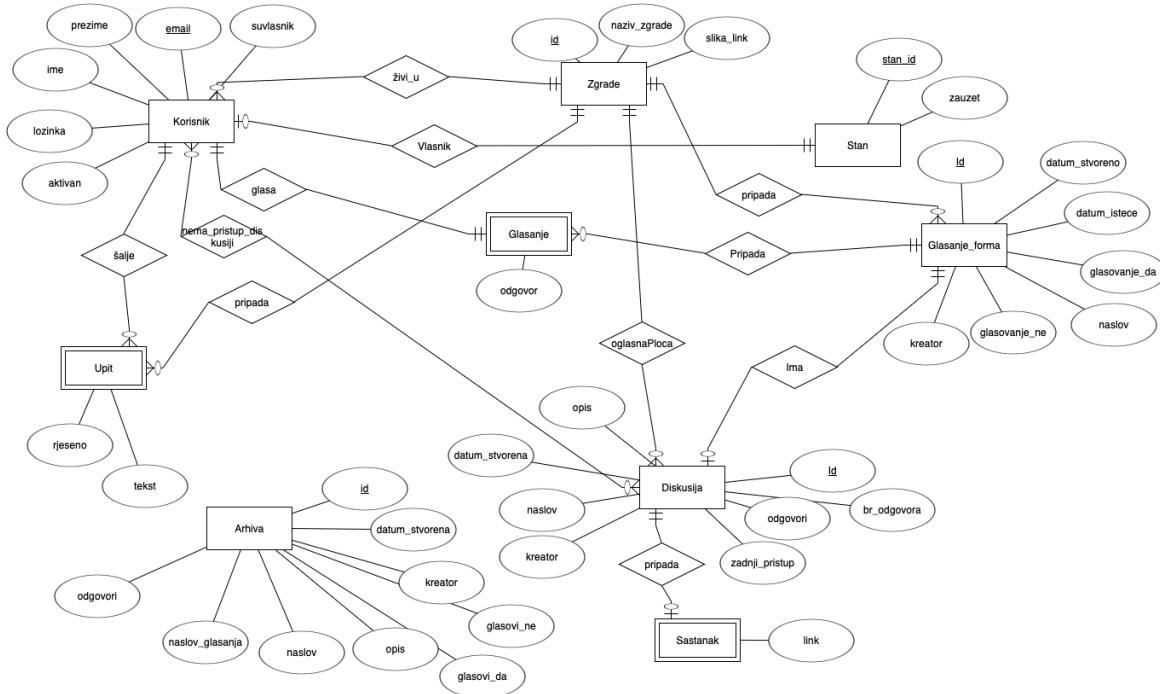
Atribut	Tip podatka	Opis varijable
id_diskusije	INT	Jedinstveni identifikator diskusije
link	VARCHAR	link na sastanak

Arhiva

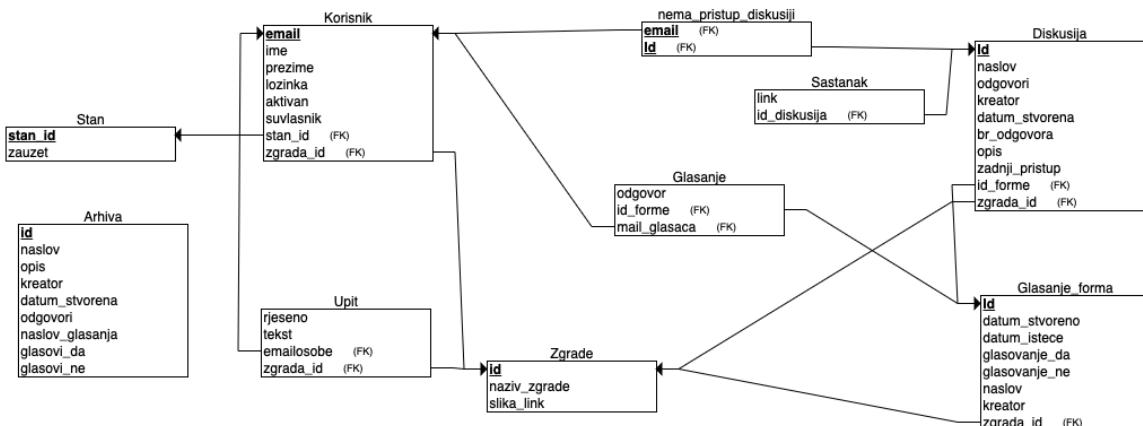
Entitet Arhiva sadrži podatke o neaktivnim diskusijama. Sadrži atribute: Id, Naslov, Odgovori, Kreator, Datum_stvorena, Opis, Glasova_da i Glasova_ne

Atribut	Tip podatka	Opis varijable
Id	INT	Jedinstveni identifikator diskusije
Naslov	VARCHAR	Naslov diskusije
Odgovori	TEXT	Odgovori korisnika u diskusiji
Kreator	VARCHAR	Identifikator inicijatora diskusije
Datum_stvorena	DATE	Datum stvaranja diskusije
Opis	TEXT	Tekst opisa diskusije
Glasova_da	INT	broj pozitivnih glasova
Glasova_ne	INT	broj negativnih glasova

4.2.2 Dijagram baze podataka



Slika 4. 1 ER dijagram baze podataka

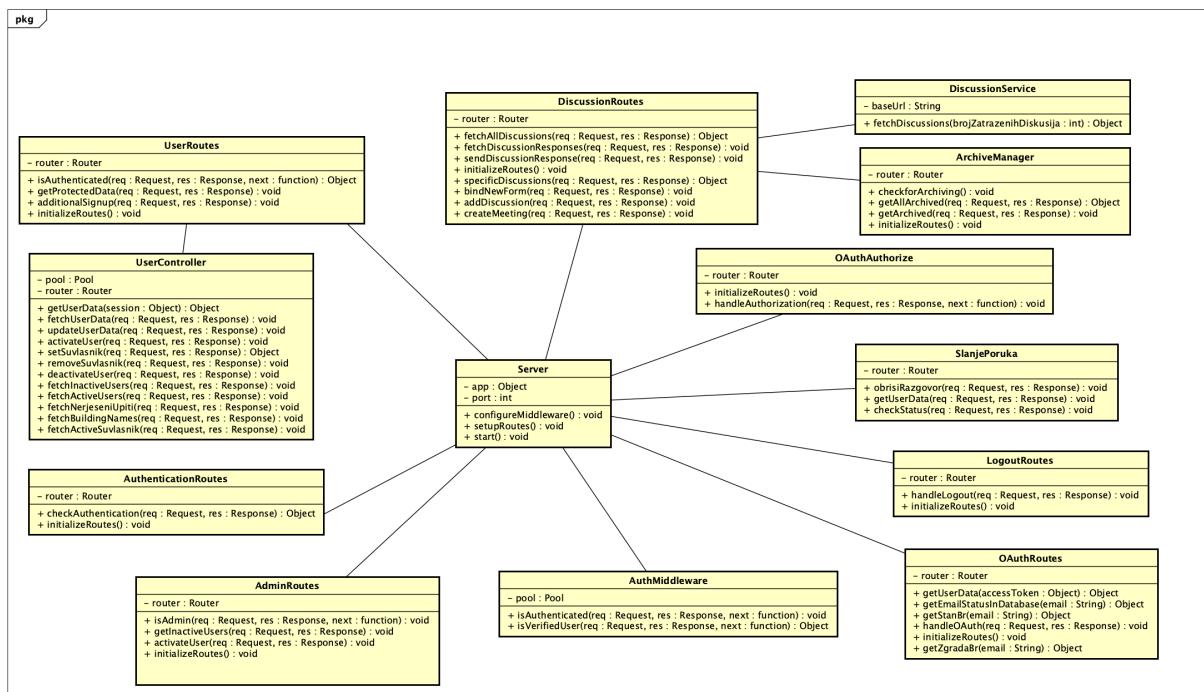


Slika 4. 2 Relacijski dijagram baze podataka

4.3 Dijagram razreda

Na slici 4.3 prikazani su razredi koji odgovaraju *backend* dijelu MVC arhitekture. Metode implementirane u tim razredima manipuliraju modelima i vraćaju zatražene podatke koji su reprezentirani modelima u listama. Razred Server upravlja svim

ostalim razredima. Razredi UserRoutes i UserController upravljaju podacima o korisniku i akcijama korisnika unutar aplikacije. Razredi DiscussionRoutes, DiscussionService i ArchiveManager upravljaju diskusija. Razred DiscussionService omogućuje dohvaćanje određenog broja diskusija iz baze podataka. Razred ArchiveManager služi za provjeru i premještanje neaktivnih diskusija u arhivu. Razredi OAuthAuthorize, OAuthRoutes, AuthMiddleware i AuthenticationRoutes služe za autentifikaciju i verifikaciju korisnika kako neautorizirani korisnici ne bi imali pristup funkcionalnostima aplikacije. Razred SlanjePoruka upravlja upitima koje su poslali korisnici. Razred LogoutRoutes omogućuje odjavu iz sustava.

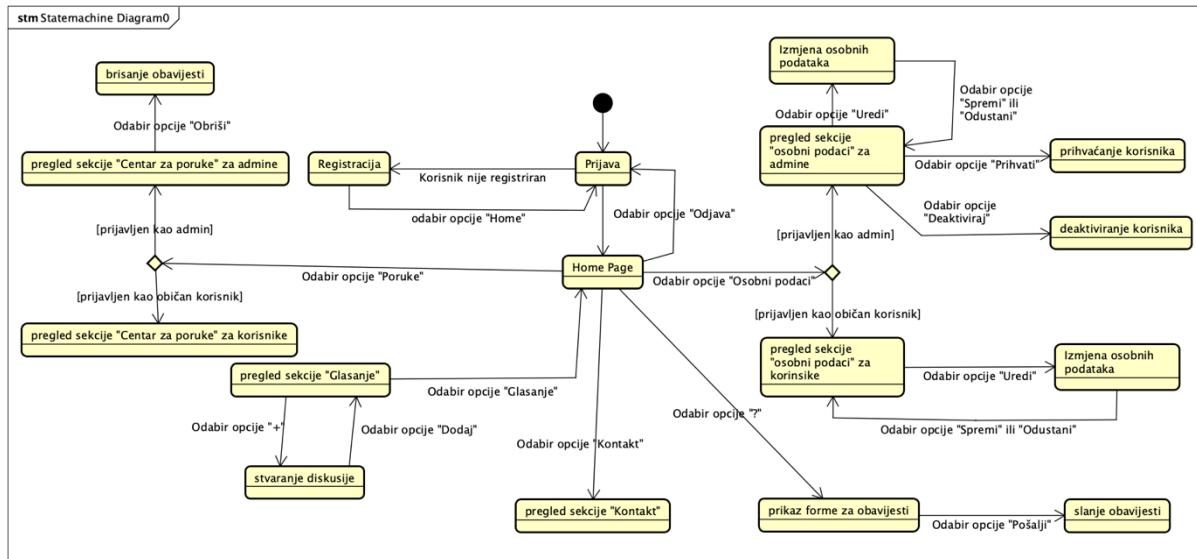


Slika 4. 3 Dijagram razreda

4.4 Dijagram stanja

Na slici 4.4 prikazan je dijagram stanja korisničkog sučelja. Stranica na koju se uvijek prvo dođe je Prijava. Ovisno o email adresi s kojom se korisnik prijavljuje može biti preusmjeren na stranicu za registraciju ili početnu stranicu. Korisnici s aktivnim korisničkim računima tj. korisnički računi koje je prihvatio administrator aplikacije preusmjereni su na početnu stranicu, a ostali na stranicu za registraciju. Korisnik s početne stranice može birati opcije Osobni podaci, Poruke, Glasanje, Kontakt, Obavijesti i Odjava. Pri pregledu sekcijske Osobni podaci korisnik može mijenjati osobne podatke odabirom opcije Uredi. Administrator na stranici Osobi podaci ima pristup listi korisnika te može upravljati korisničkim računima. Prilikom

pregleda sekcije Poruke administrator za razliku od običnog korisnika ima mogućnost brisanja obavijesti. Pri pregledu sekcije Glasanje moguće je stvoriti novo glasaje.

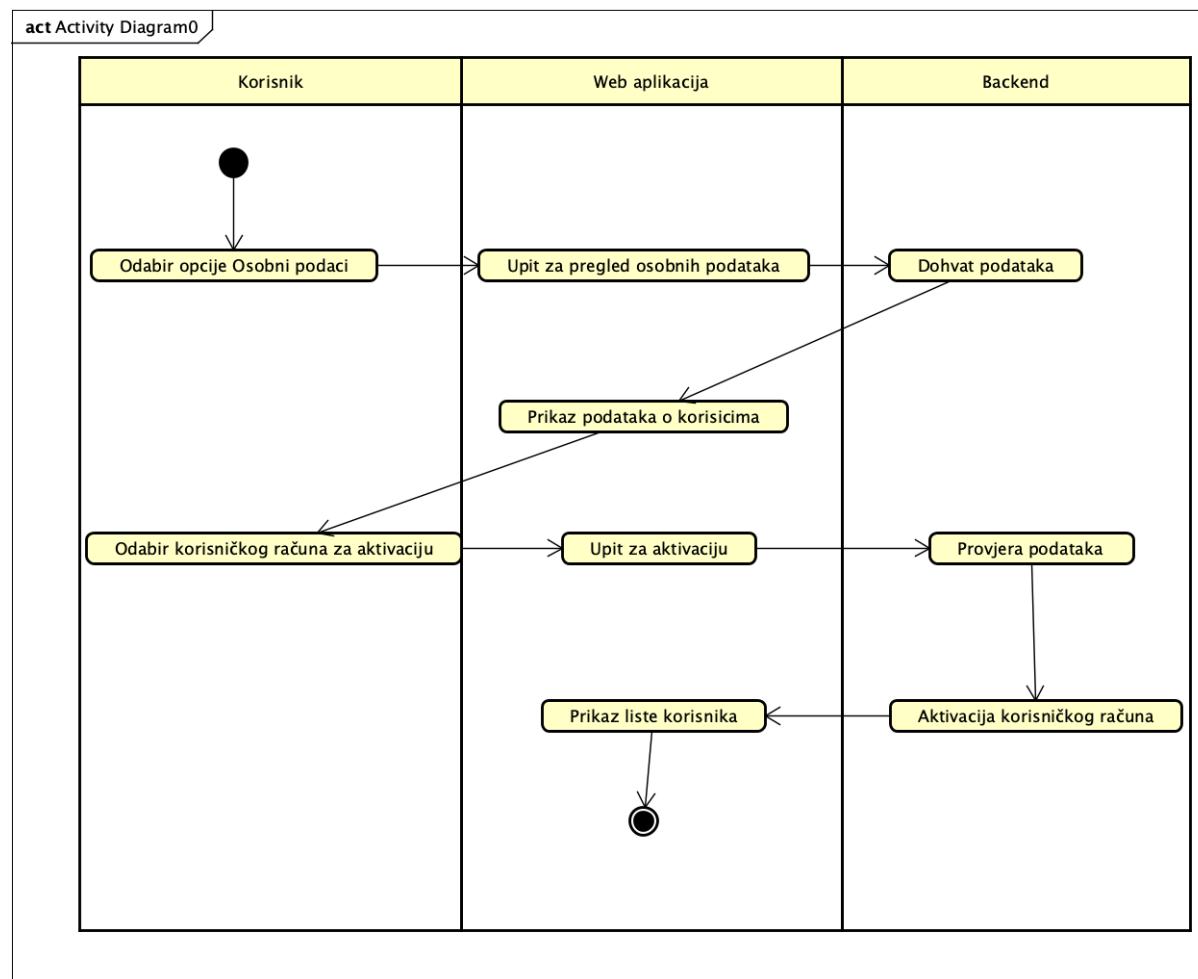


Slika 4. 4 Dijagram stanja

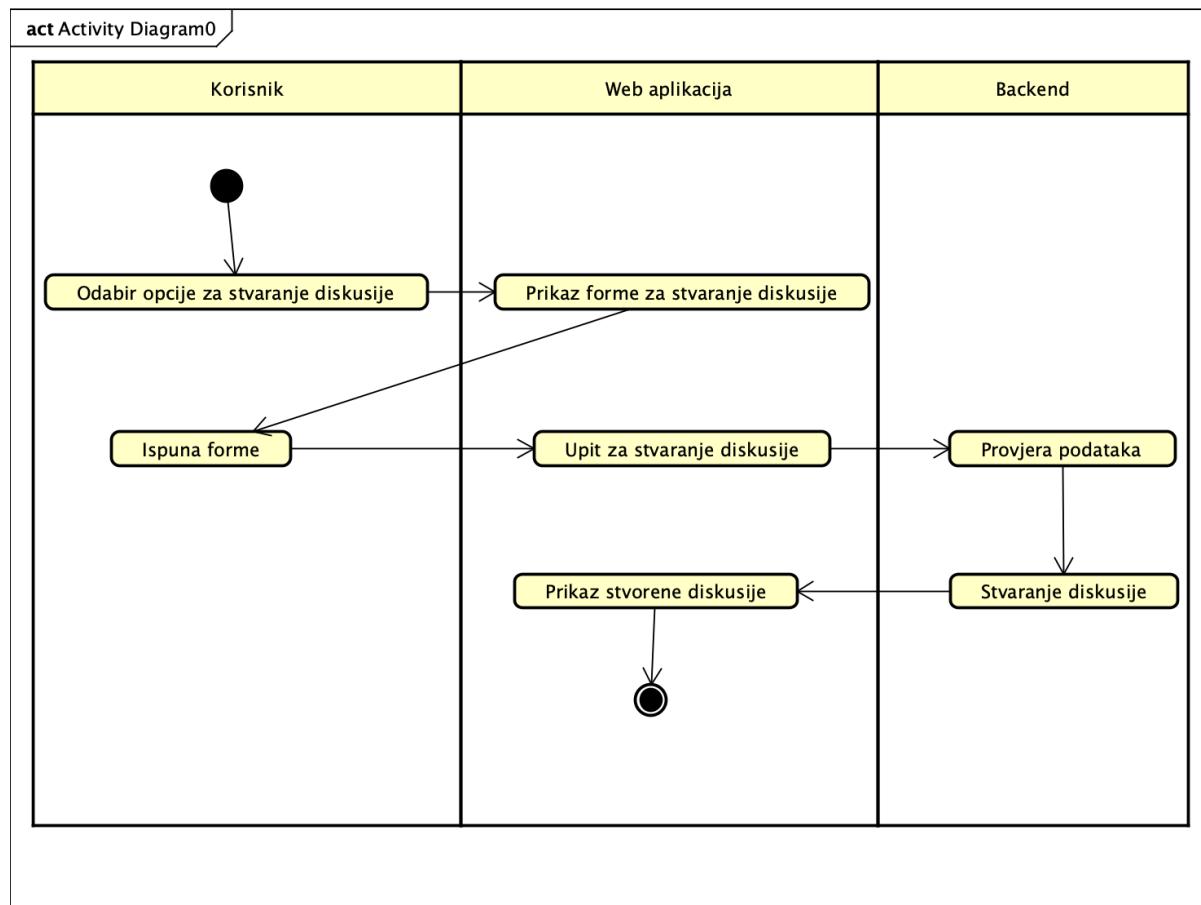
4.5 Dijagram aktivnosti

Na slici 4.5 prikazan je dijagram aktivnosti prilikom aktivacije/deaktivacije korisničkog računa. Prijavljeni korisnik s ulogom administratora na stranici Osobni podaci odabire korisnički račun koji želi aktivirati/deaktivirati. Nakon toga sustav obrađuje dobiveni zahtjev i prikazuje ažuriranu listu korisnika.

Na slici 4.6 prikazan je dijagram aktivnosti prilikom stvaranja nove diskusije. Prijavljeni korisnik odabire opciju za stvaranje diskusije nakon čega sustav prikazuje formu za stvaranje diskusije koju korisnik mora popuniti. Nakon predaje forme, sustav provjerava podatke, sprema ih u bazu i ažurira listu diskusija. Diskusija se prikazuje na oglašnoj ploči.



Slika 4. 5 Dijagram aktivnosti 1

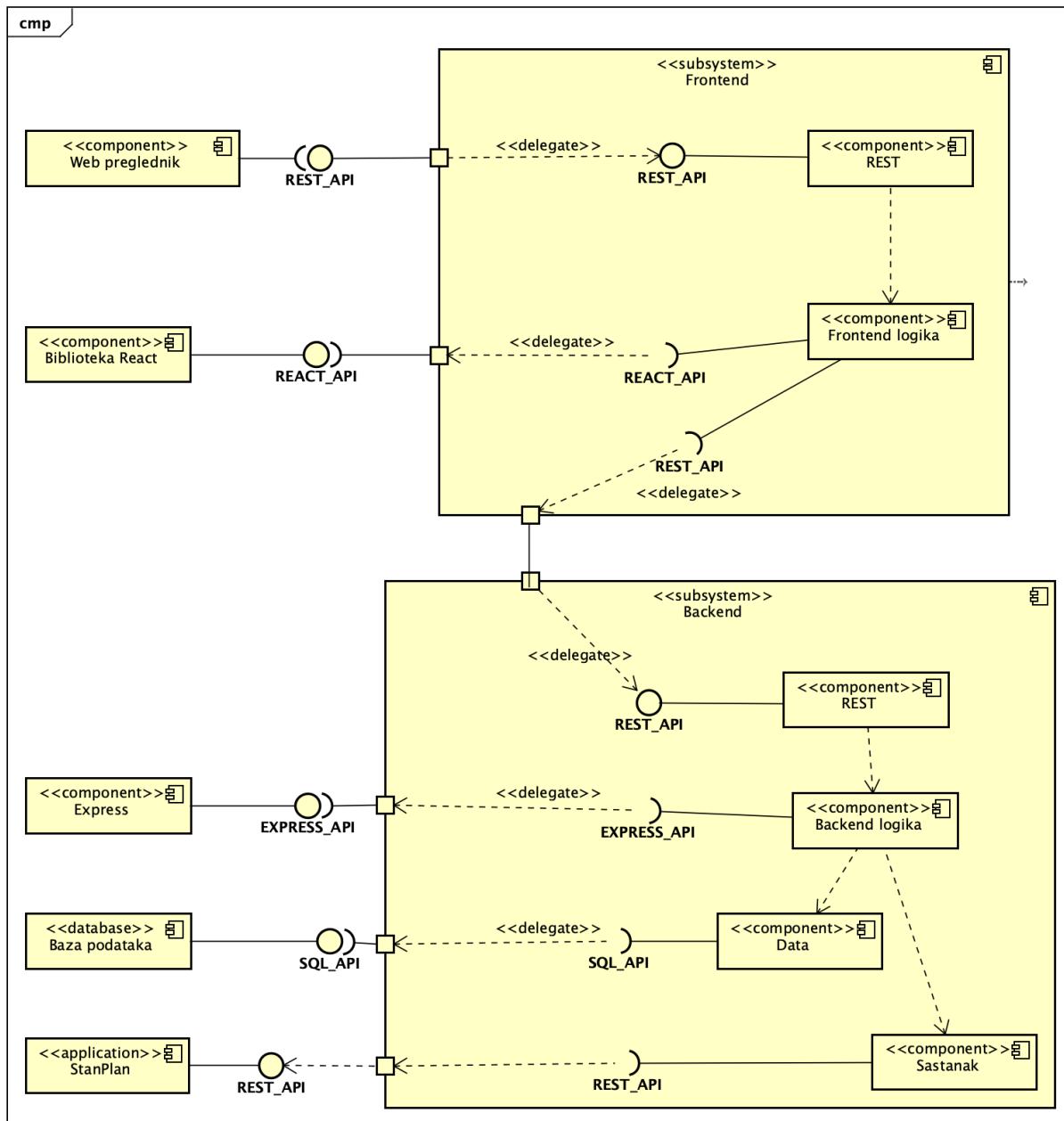


Slika 4. 6 Dijagram aktivnosti 2

5 Arhitektura komponenata i razmještaja

5.1 Dijagram komponenata

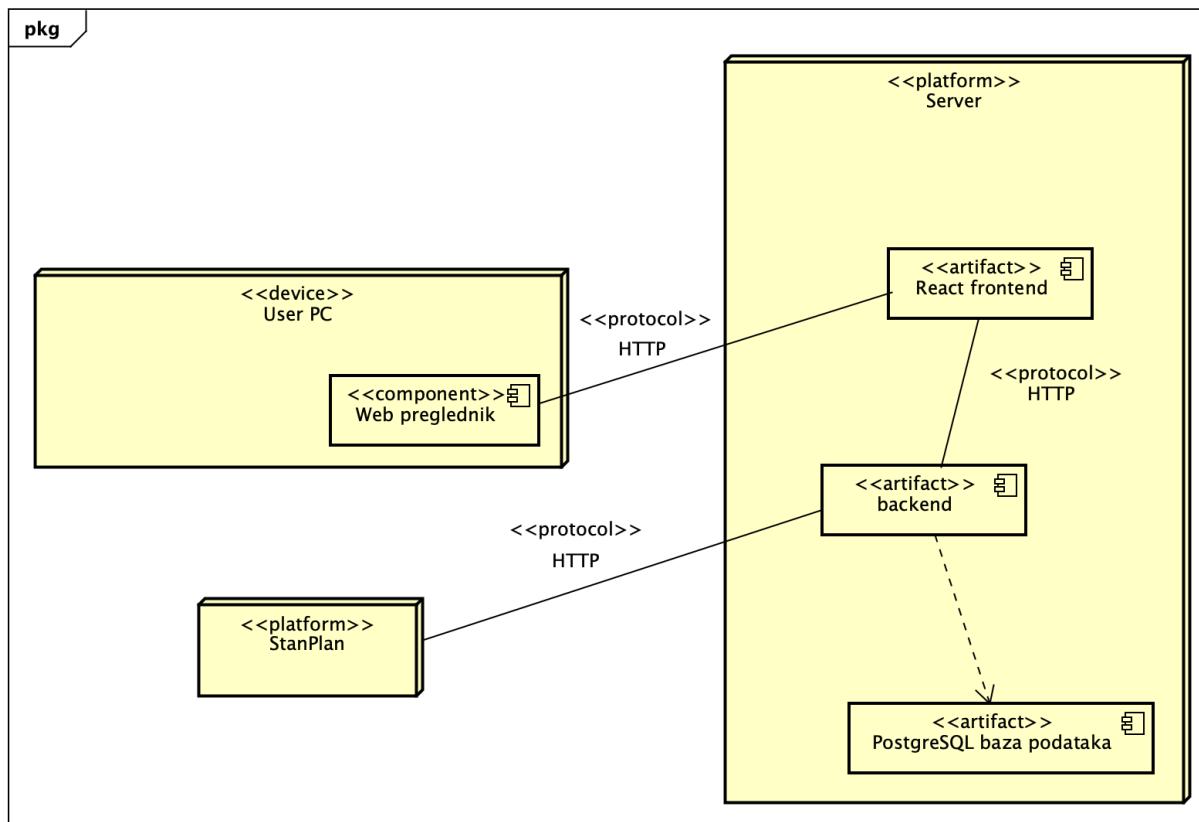
Dijagram komponenti prikazan na slići 5.1 opisuju međuvisnost i organizaciju i odnose u internoj strukturi. Korisnik iz web preglednika pristupa aplikaciji korištenjem REST API-ja. Sama aplikacija se sastoji od dvije komponente. Prva komponenta odgovara frontendu i izgrađena je korištenjem React biblioteke. Druga komponenta odgovara backendu. Frontend i backend komuniciraju korištenjem REST API-ja. Baza podataka je relacijska i backend joj pristupa slanjem SQL upita. Backend komunicira s aplikacijom StanPlan korištenjem REST API-ja.



Slika 5. 1 Dijagram komponenti

5.2 Dijagram razmještaja

Dijagram razmještaja prikazan je na skici 5.2. Sustav je baziran na arhitekturi klijent-poslužitelj. Korisnik pristupa aplikaciji korištenjem web preglednika. Na serveru se nalaze poslužitelji za frontend, backend i bazu podataka. Komunikacija između korisnika i poslužitelja za frontend te poslužitelja za frontend i poslužitelja za backend ostvaruje se korištenjem HTTP protokola.



Slika 5. 2 Dijagram razmještaja

6 Ispitivanje programskog rješenja

6.1 Ispitivanje komponenti

Ispitivanje komponenti ostvareno je pomoću Jest alata za ispitivanje. U nastavku su opisani provedeni testni slučajevi.

Ispitni slučajevi 1-4 testiraju ArchiveManager. Inicijalizacijska metoda za testove 1-4 izgleda ovako:

```

beforeAll(async () => {
    await pool.query("insert into diskusija values (16, 'Discussion 16', 'text 16', 1, '2024-08-01', '2024-09-09', 4, 'description')")
    await pool.query("insert into diskusija values (17, 'Discussion 17', 'text 17', 2, '2024-09-01', '2024-12-22', 4, 'description')")
    await pool.query("insert into diskusija values (18, 'Discussion 18', 'text 18', 2, '2024-09-01', '2024-12-24', 4, 'description')")
    await pool.query("insert into diskusija values (19, 'Discussion 19', 'text 19', 1, '2024-09-01', '2025-01-14', 4, 'description')")
});
  
```

U prvome ispitnom slučaju testira se arhiviranje diskusije neaktivne više od 30 dana. Prije provođenja testa diskusija se nalazi u tablici Diskusija s ostalim aktivnim diskusijama. Očekivani izlaz je da će diskusija biti prebačena iz aktivnih diskusija u arhivu.

```
test('diskusija neaktivna vise od 30 dana',async () => {
    ArchiveManager.checkforArchiving();
    const resultArhiva = await pool.query(
        'SELECT * FROM arhiva WHERE id=16'
    );
    const resultDiskusija = await pool.query(
        'SELECT * FROM diskusija WHERE id=16'
    );
    await expect(resultArhiva.rowCount).toBe(1)
    await expect(resultDiskusija.rowCount).toBe(0)
});
```

Slika 6. 1 Testiranje komponenti 1

U drugome ispitnom slučaju testira se diskusija neaktivna točno 31 dan. Očekivani rezultat je da će diskusija biti obrisana iz aktivnih diskusija i biti prebačena u arhivu.

```
test('diskusija neaktivna tocno 31 dan',async () => {
    ArchiveManager.checkforArchiving();
    const resultArhiva = await pool.query(
        'SELECT * FROM arhiva WHERE id=17'
    );
    const resultDiskusija = await pool.query(
        'SELECT * FROM diskusija WHERE id=17'
    );
    await expect(resultArhiva.rowCount).toBe(1)
    await expect(resultDiskusija.rowCount).toBe(0)
});
```

Slika 6. 2 Testiranje komponenti 2

U trećem ispitnom slučaju testira se arhiviranje diskusije neaktivne točno 30 dana. Očekivani rezultat je da će diskusija ostati među aktivnim diskusijama i neće biti premještena u arhivu.

```
test('diskusija neaktivna točno 30 dana',async () => {
  ArchiveManager.checkforArchiving();
  const resultArhiva = await pool.query(
    | 'SELECT * FROM arhiva WHERE id=18'
  );
  const resultDiskusija = await pool.query(
    | 'SELECT * FROM diskusija WHERE id=18'
  );
  await expect(resultArhiva.rowCount).toBe(0)
  await expect(resultDiskusija.rowCount).toBe(1)
});
```

Slika 6. 3 Testiranje komponenti 3

U četvrtom ispitnom slučaju testira se arhiviranje diskusije neaktivne manje od 30 dana. Očekivani rezultat je da diskusija neće biti premještena u arhivu i ostati će među aktivnim diskusijama.

```
test('diskusija neaktivna manje od 30 dana',async () => {
  ArchiveManager.checkforArchiving();
  const resultArhiva = await pool.query(
    | 'SELECT * FROM arhiva WHERE id=19'
  );
  const resultDiskusija = await pool.query(
    | 'SELECT * FROM diskusija WHERE id=19'
  );
  await expect(resultArhiva.rowCount).toBe(0)
  await expect(resultDiskusija.rowCount).toBe(1)
});
```

Slika 6. 4 Testiranje komponenti 4

Rezultati:

```
Test Suites: 1 passed, 1 total
Tests:       4 passed, 4 total
Snapshots:   0 total
Time:        0.263 s, estimated 1 s
Ran all test suites.
```

Slika 6. 5 Rezultati ispitnih slučajeva 1-4

Ispitni slučajevi 5 i 6 testiraju UserController koji upravlja korisničkim računima.

Initializacijska metoda za testove 5 i 6 izgleda ovako:

```
beforeAll(async () => {
  await pool.query("insert into korisnik values ('firstName', 'lastName', 'password',
'example@gmail.com', 3, false)")
  await pool.query("insert into korisnik values ('firstName', 'lastName', 'password',
'example1@gmail.com', 4, true)")
});
```

U petom ispitnom slučaju testira se aktivacija korisničkog računa. Na temelju poslanih podataka o korisničkom računu potrebno je ako u bazi postoji odgovarajući korisnik, aktivirati njegov korisnički račun. Očekivani izlaz je da se u bazi nakon provedbe testa nalazi aktiviran primljeni korisnički račun.

```
test("dodavanje neaktivnog korisnika", async () => {
  mock.onPost("/activate-user", {body: [{email: 'example@gmail.com',
    ime: 'firstName', prezime: 'lastName', stan_id: 3}]});
  const result = await pool.query(
    "SELECT * FROM korisnik WHERE stan_id=3 AND aktivan=false"
  );
  expect(result.rowCount).toBe(1)
});
```

Slika 6. 6 Testiranje komponenti 5

U šestom ispitnom slučaju testira se deaktivacija korisničkog računa. Na temelju poslanih podataka o korisničkom računu potrebno je ako u bazi postoji odgovarajući korisnik, deaktivirati njegov korisnički račun. Očekivani izlaz je da se u bazi nakon provedbe testa nalazi deaktiviran primljeni korisnički račun.

```
test("deaktiviranje korisničkog računa", async () => {
  mock.onPost("/deactivate-user", {body: [{email: 'example1@gmail.com',
    |   | ime: 'firstName', prezime: 'lastName', stan_id: 4}]});
  const result = await pool.query(
    "SELECT * FROM korisnik WHERE stan_id=4 and aktivan=false"
  );
  expect(result.rowCount).toBe(1)
});
```

Slika 6. 7 Testiranje komponenti 6

Rezultati:

```
Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:   0 total
Time:        0.253 s, estimated 1 s
Ran all test suites.
```

Slika 6. 8 Rezultati ispitnih slučajeva 5 i 6

6.2 Ispitivanje sustava

Ispitivanje sustava provedeno je pomoću dodatka za preglednik Selenium IDE. U nastavku su opisani provedeni testni slučajevi.

1. Ispitni slučaj

Prvi ispitni slučaj ispitiva pokušaj prijave registriranog korisnika čiji podaci postoje u bazi podataka.

1. Ulazi:

- Klik na gumb Prijavite se putem Google-a
- Odabir korisničkog računa
- Klik na gumb Nastavi

2. Koraci ispitivanja:

- Otvoriti aplikaciju u pregledniku
- Kliknuti na gumb Prijavite se putem Google-a
- Kliknuti na željeni korisnički račun
- Kliknuti na Nastavi
- Verificirati očekivane rezultate

3. Očekivani izlaz:

- a. Korisnik je preusmjeren na početnu stranicu nakon prijave

4. Dobiveni izlaz:

	Command	Target	Value
1	✓ open	/signup	
2	✓ set window size	616x848	
3	✓ click	css=.google-auth-button	
4	✓ click	css=.aZvCDf:nth-child(2) > .VV3oRb	
5	✓ click	css=.VfPpkd-ksKsZd-mWPk3d-OWXExe-Tv8l5d-IJfZMc > .VfPpkd-RLmnJb	
6	✓ verify element not present	css=.Ekjuhf	

Slika 6. 9 Rezultat prvog ispitnog slučaja

2. Ispitni slučaj

Drugi ispitni slučaj ispitiva pokušaj prijave korištenjem neispravne email adrese.

1. Ulazi:

- a. Klik na gumb Prijavite se putem Google-a
- b. Klik na gumb Upotrijebite neki drugi račun
- c. Unos email adrese
- d. Klik na gumb Dalje\

2. Koraci ispitivanja:

- a. Otvoriti aplikaciju u pregledniku
- b. Kliknuti na gumb Prijavite se putem Google-a
- c. Kliknuti na gumb Upotrijebite neki drugi račun
- d. Unijeti podatke u formu
- e. Klik na gumb Dalje
- f. Verificirati očekivane rezultate

3. Očekivani izlaz:

- a. Neuspješna prijava
- b. Prikaz poruke pogreške

4. Dobiveni izlaz:

	Command	Target	Value
1	✓ <i>open</i>	/signup	
2	✓ <i>set window size</i>	1470x832	
3	✓ <i>click</i>	css=.google-auth-button	
4	✓ <i>click</i>	css=.B682ne > .VV3oRb	
5	✓ <i>type</i>	id=identifierId	4ab
6	✓ <i>click</i>	css=.VfPpkd-LgbsSe-OWXEXe-k8QpJ > .VfPpkd-vQzf8d	
7	✓ <i>verify element present</i>	css=.LXRPh	

Slika 6. 10 Rezultat drugog ispitnog slučaja

3. Ispitni slučaj

Treći ispitni slučaj ispitiva pokušaj prijave neregistriranim korisničkim računom za koji je već zaprimljen zahtjev za registraciju.

1. Ulazi:

- a. Klik na gumb Prijavite se putem Google-a
- b. Odabir korisničkog računa
- c. Klik na gumb Nastavi

2. Koraci ispitivanja:

- a. Otvoriti aplikaciju u pregledniku
- b. Kliknuti na gumb Prijavite se putem Google-a
- c. Kliknuti na željeni korisnički račun
- d. Kliknuti na Nastavi
- e. Verificirati očekivane rezultate

3. Očekivani izlaz:

- a. Neuspješna prijava
- b. Prikaz poruke pogreške

4. Dobiveni izlaz:

	Command	Target	Value
1	✓ open	/signup	
2	✓ set window size	1200x817	
3	✓ click	css=.google-auth-button	
4	✓ click	css=.aZvCDf:nth-child(1) .LbOduc	
5	✓ click	css=.VfPpkd-ksKsZd-mWPk3d-OWXEX e-Tv8l5d-IJfZMc > .VfPpkd-RLmnJb	
6	✓ verify element not present	css=a:nth-child(1) > .nav-icon	

Slika 6. 11 Rezultat trećeg ispitnog slučaja

4. Ispitni slučaj

Četvrti ispitni slučaj ispitiva pokušaj prijave neregistriranog korisnika koji šalje zahtjev za registraciju.

1. Ulazi:

- a. Klik na gumb Prijavite se putem Google-a
- b. Odabir korisničkog računa
- c. Klik na gumb Nastavi
- d. Odabir stana iz izbornika
- e. Klik na gumb Complete Signup

2. Koraci ispitivanja:

- a. Otvoriti aplikaciju u pregledniku
- b. Kliknuti na gumb Prijavite se putem Google-a
- c. Kliknuti na željeni korisnički račun
- d. Kliknuti na Nastavi
- e. Odabrati stan iz padajućeg izbornika
- f. Kliknuti na dumb Complete Signup g. Verificirati očekivane rezultate

3. Očekivani izlaz:

- a. Uspješno poslani podaci
- b. Povratak na početnu stranicu

4. Dobiveni izlaz:

	Command	Target	Value
1	✓ open	/signup	
2	✓ set window size	1200x817	
3	✓ click	css=.google-auth-button	
4	✓ click	css=.aZvCDf:nth-child(1) .pGzURd	
5	✓ click	css=.VfPpkd-ksKsZd-mWPk3d-OWXEXe-Tv8l5d-lJfZMc > .VfPpkd-RLmnJb	
6	✓ click	css=select	
7	✓ select	css=select	label=1
8	✓ click	css=.submit-button	
9	✓ verify element present	css=.google-auth-button	

Slika 6. 12 Rezultat četvrtog ispitnog slučaja

7 Tehnologije za implementaciju aplikacije

7.1 Korištene tehnologije i alati

1. Programski jezici

JavaScript ES2023

JavaScript je objektno orijentirani programski jezik koji se može koristiti na klijentskoj i na poslužiteljskoj strani. JavaScript čini web stranice dinamičnijima i lakšima za korištenje.

2. Radni okviri i biblioteke

Node.js v20.18

Node.js je open-source i cross-platform runtime okruženje za izvršavanje JavaScript koda izvan preglednika. Služi za programiranje na strani poslužitelja i prvenstveno se koristi za poslužitelje vođene događajima, kao što su tradicionalne web stranice i pozadinske API usluge.

React 19

Za razvoj klijentskog dijela aplikacije korištena je JavaScript biblioteka **React**. React se ističe zbog svoje jednostavnosti, fleksibilnosti i mogućnosti ponovne upotrebe komponenti. Sjajan je izbor za početnike jer se može naučiti brzo i jednostavno.

Express.js v5

Za razvoj backend dijela aplikacije korišten je **Express.js**, web framework za izgradnju RESTful API-ja s Node.js-om. Express pomaže organizirati web aplikaciju na strani poslužitelja u organiziranju MVC arhitekturu. JavaScript biblioteka Expressa pomaže u izradi učinkovite i brze web aplikacije

3. Baza podataka

PostgreSQL 17

PostgreSQL relacijska baza koja se ističe svojom dokazanom arhitekturom, pouzdanošću i integritetom podataka. Radi na svim glavnim operativnim sustavima te ispunjava ACID zahtjeve.

4. Razvojni alati:

Visual Studio Code 1.96

Visual Studion Code omogučava jednostavno pisanje, uređivanje i testiranje koda. Radi na svim glavnim operativnim sustavima te podržava brojne programske jezike i ekstenzije.

GitHub

GitHub omogućuje suradnju u stvarnom vremenu. Potiče timove da rade zajedno u razvoju koda, izgradnji web stranica i ažuriranju sadržaja. Uz GitHub mogu se jednostavno pratiti promjene i upravljati revizijama.

5. Alati za ispitivanje:

Jest 29

Jest je framework za testiranje JavaScripta dizajniran da osigura ispravnost bilo koje baze JavaScript koda. Omogućuje jednostavno pisanje testova koji brzo daju rezultate. Jest je dobro dokumentiran, zahtijeva malo konfiguracije i može se proširiti kako bi odgovarao traženim zahtjevima.

Selenium IDE

Selenium IDE je open-source alat koji se široko koristi u provođenju automatiziranog web testiranja i automatizacije preglednika. Selenium IDE bilježi ono što radite na web stranici. Naknadno se te snimljene interakcije mogu reproducirati kao automatizirani testovi.

8 Upute za puštanje u pogon

8.1 Instalacija

Preduvjeti:

- Node.js v18 ili noviji
- NPM

- PostgreSQL 16 ili noviji
- Git

Preuzimanje:

```
git clone https://github.com/Tadija-Skugor/EasyFlat
```

```
cd EasyFlat
```

8.2 Postavke

Konfiguracijske datoteke:

Potrebno je kreirati .env datoteku u backend direktoriju sa sljedećim sadržajem:

DB_USER = <database user>

DB_HOST = <host>

DB_NAME = EasyFlat

DB_PASSWORD = <password>

CLIENT_ID = <client id>

CLIENT_SECRET = <client secret>

Postavke baze podataka:

Kreirati PostgreSQL bazu podataka koja odgovara shemi iz odjeljka Baze podataka. Datoteke za stvaranje baze podataka pod imenom EasyFlat se nalaze u direktoriju dbBackups

8.3 Pokretanje aplikacije

Pokretanje poslužiteljskog dijela aplikacije:

```
cd backend
```

```
npm i
```

```
npm start
```

Pokretanje klijentskog dijela aplikacije:

```
cd frontend
```

```
npm i
```

```
npm run dev
```

- Provjera rada: <http://localhost:5000>

8.4 Opis pristupa aplikaciji na javnom poslužitelju

Ograničenja:

- Potrebna je registracija za pristup svim funkcionalnostima aplikacije
- OAuth prijava putem Google računa je podržana
- Optimizirano za moderne web preglednike (Chrome, Firefox, Safari, Edge)

Aplikaciji je moguće pristupiti putem bilo kojeg web preglednika upisivanje poveznice: <https://easyflat.eu.ngrok.io/main> u polje za pretragu. Administratori pristupaju aplikaciji isto kao i normalni korisnici.

9 Zaključak i budući rad

Zadatak naše grupe bio je razvoj web aplikacije pod imenom "Easy Flat". Ideja aplikacije je da bude verzija online oglasne ploče zgrade gdje će stanari moći objavljivati diskusije, glasati i organizirati sastanke. Nakon 12 tjedana, ostvarili smo zadani cilj i projekt je završen. Projekt se sastojao od tri faze.

Prva faza je najopširnija te je uključivala upoznavanje te okupljanje tima, raspravljanje o idejama te konačno podjelu poslova i početak rada. Formirala su se tri podtima. Prvi podtim je bio zadužen za dizajn i implementaciju funkcionalnosti na frontendu. Drugi podtim je radio na razvoju baze podataka, implementaciji funkcionalnosti na backendu i deploymentu. Treći podtim je bio zadužen za izradu dokumentacije. Prva faza trajala je do predaje prve verzije projekta.

Druga faza se većinom sastojala od rada na implementaciji svih ključnih funkcionalnosti aplikacije. Tijekom ove faze bilo potrebno surađivati s timom koji je radio na razvoju aplikacije "StanPlan", preko koje se održavaju sastanci stanara. Cilj

ove faze bila je implementacija svih predviđenih funkcionalnosti aplikacije. Druga faza trajala je do demonstracije alfa inačice aplikacije.

Treću fazu karakterizira testiranje aplikacije, ispravljanje uočenih pogrešaka i izrada UML dijagrama. U ovoj fazi prvi i drugi podtim su radili na ispravljanju pogrešaka i završnim detaljima u dizajnu i implementaciji. Treći podtim se fokusirao na ispitivanje aplikacije i izradu UML dijagrama. Ovo je završna faza projekta i trajala je do konačne predaje.

Izgrađenu aplikaciju moguće je proširiti na mnogo načina. Jedna od ideja je dodati više funkcionalnosti na oglasnu ploču tako da korisnici mogu pregledati profile ostalih korisnika te vidjeti njihove objave. Također, bi se moglo dodat slanje obavijesti korisnicima kada netko odgovori na njihovu diskusiju ili su pozvani da sudjeluju u sastanku. Drugo proširenje koje bi doprinijelo kvaliteti aplikacije je mogućnost da korisnici šalju izravno poruke drugim korisnicima.

Sudjelovanje u ovom projektu je bilo vrijedno iskustvo za sve članove tima. Većini nas je ovo bio prvi ozbiljniji grupni projekt, i snašli smo se jako dobro. Konflikata u timu gotovo nije bilo, a suradnja i komunikacija su bili na iznimno zadovoljavajućoj razini. Većini nas je ovaj projekt bio prvi ozbiljniji dodir s tehnologijama poput Gita. Naučili smo koristiti neke moderne radne okvire pri izradi web aplikacija. Iznimno smo zadovoljni postignutim rezultatima i timskim radom koji je do tih rezultata doveo.

Popis literature

1. Programsко inženjerstvo, FER
ZEMRIS, <https://www.fer.unizg.hr/predmet/proinz>
2. I. Sommerville, "Software engineering", 8th ed, Addison Wesley, 2007
3. T.C.Lethbridge, R.Langaniere, "Object-Oriented Software Engineering", 2nd ed. McGraw-Hill, 2005.
4. The Unified Modeling Language, <https://www.uml-diagrams.org/>
5. Astah Community, <http://astah.net/editions/uml-new>

Dnevnik promjena dokumentacije

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Napravljen predložak	Đana Pavić	26.10.2024.
0.2	Dodani funkcionalni zahtjevi	Đana Pavić	27.10.2024.

0.3	Dodani dionici i ostali zahtjevi	Đana Pavić	28.10.2024.
0.4	Dodan opis projektnog zadatka	Đana Pavić	2.11.2024.
0.5	Dodani obrasci uporabe	Đana Pavić	3.11.2024.
0.6	Dorađeni obrasci uporabe. Dodani dijagrami obrazaca uporabe, sekvenčni dijagrami i dijagram baze podataka	Đana Pavić	5.11.2024.
0.7	Dodani dijagrami razreda.	Đana Pavić	11.11.2024.
0.8	Dodana arhitektura sustava	Đana Pavić	12.11.2024.
0.9	Dorađeni dijagrami obrazaca uporabe. Dorađeni sekvenčni dijagrami	Đana Pavić	14.11.2024.
0.9.1	Dorađeni dijagrami razreda. Osvježen opis projektnog zadatka. Dodan ER dijagram baze.	Đana Pavić	15.11.2024.
1.0	Verzija prve predaje.	Đana Pavić	15.11.2024.
1.1	Osvježen predložak. Dodano ispitivanje sustava	Đana Pavić	14.01.2025.
1.2	Dodan dijagram stanja	Đana Pavić	17.01.2025.
1.3	Dodan dijagram komponenti	Đana Pavić	18. 01.2025.
1.4	Dodane korištene tehnologije i alati	Đana Pavić	19. 01.2025.
1.5	Dodan dijagram razmještaja. Dodano ispitivanje komponenti.	Đana Pavić	20. 01.2025.
1.6	Osvježena baza podataka i dijagram razreda.	Đana Pavić	22. 01.2025.
1.7	Dodan zaključak	Đana Pavić	23. 01.2025.
1.8	Dodane upute za puštanje u pogon	Đana Pavić	24. 01.2025.
2.0	Verzija druge predaje	Đana Pavić	24. 01.2025.

Prikaz aktivnosti grupe

Dnevnik sastajanja

1. sastanak

- Datum: 11. listopada 2024.
- Prisustvovali: F.Cvjetićanin, T.Livaić, Đ.Pavić, M.Krajačić, M.Šarić, T.Škugor, D.Weiner
- Teme sastanka:
- formiranje tima
- odabir voditelja tima
- upoznavanje s zadatkom

2. sastanak

- Datum: 16. listopada 2024.
- Prisustvovali: F.Cvjetićanin, T.Livaić, Đ.Pavić, M.Krajačić, M.Šarić, T.Škugor, D.Weiner
- Teme sastanka:
- odabrane bazne tehnologije
- okvirna podjela zadataka na projektu u prvo ciklusu predaje

3. sastanak

- Datum: 19. listopada 2024.
- Prisustvovali: F.Cvjetićanin, T.Livaić, Đ.Pavić, M.Krajačić, M.Šarić, T.Škugor, D.Weiner
- Teme sastanka:
- definiran općenit dizajn web stranice
- definirani funkcionalni i nefunkcionalni zahtjevi

4. sastanak

- Datum: 7. studenog 2024.
- Prisustvovali: F.Cvjetićanin, T.Livaić, M.Krajačić, T.Škugor
- Teme sastanka:
- planiranje daljnog rada backenda

5. sastanak

- Datum: 11. studenog 2024.
- Prisustvovali: F.Cvjetićanin, T.Livaić, Đ.Pavić, M.Krajačić, M.Šarić, T.Škugor, D.Weiner
- Teme sastanka:
- testiranje deployane aplikacije

Tablica aktivnosti

	Filip Cvjetićanin	Tihomir Livaić	Đana Pavić	Mirta Krajačić	Marko Šarić	Tadija Škugor	Dona Weiner
Upravljanje projektom						5	
Opis projektnog zadatka			4				
Funkcionalni zahtjevi	2	2	3	2	2	2	2
Opis pojedinih obrazaca			5				
Dijagram obrazaca			3				
Sekvencijski dijagrami			3				
Opis ostalih zahtjeva			1				
Arhitektura i dizajn sustava			5				
Baza podataka			3	6			
Dijagram razreda			4				
Dijagram stanja			3				
Dijagram aktivnosti			3				
Dijagram komponenti			2				
Korištene tehnologije i alati			3			7	
Ispitivanje programskog rješenja			6			7	
Dijagram razmještaja			2				

Upute za puštanje u pogon			1				
Dnevnik sastajanja			1				
Zaključak i budući rad			1			2	
Popis literature			1				
Backend	39	42		36		20	3
Frontend					20	6	40
Deployment						6	
Izrada baze podataka	3	2		6		3	
Izrada dizajna aplikacije					7	3	5

Dijagram pregleda promjena

